Large Language Models Are Students at Various Levels: Zero-shot Question Difficulty Estimation

Anonymous ACL submission

Abstract

Recent advancements in educational platforms have emphasized the importance of personalized education. Accurately estimating question difficulty based on the group level of a student is essential for personalized question recommendations. Several studies have focused on predicting question difficulty using student question-solving records or textual information about the questions. However, these approaches require a large amount of student question-solving records and fail to account for the subjective difficulties perceived by different student groups. To address these limitations, we propose the LLaSA framework that utilizes large language models to represent students at various levels. LLaSA estimates question difficulty using student abilities derived from their question-solving records. Furthermore, the zero-shot LLaSA can estimate question difficulty without any student question-solving records. In evaluations on the DBE-KT22 and ASSISTMents 2005–2006 benchmarks, the zero-shot LLaSA demonstrated a performance comparable to those of strong baseline models even without any training. When evaluated using the classification method, LLaSA outperformed the baseline models, achieving state-of-the-art performance. In addition, the zero-shot LLaSA achieved a high correlation compared with the question difficulty derived from the question-solving records of students, suggesting the potential of LLaSA for real world applications.

1 Introduction

The advancement of online learning platforms such as *Coursera*[1] and *Udemy*[2] has recently emphasized the importance of personalized education. These platforms utilize extensive educational question data to recommend questions with suitable difficulty levels to students. This enables students to effectively learn by solving questions that match their skill levels (Jafari et al., 2019). To provide questions that match the skill level of students, accurately estimating the difficulty of the questions before presenting them to the students is important (Boopathiraj and Chellamani, 2013).

The process of estimating question difficulty, also known as question difficulty estimation (QDE), was performed using manual estimation (Ning et al., 2023) or the item response theory (IRT) (Hambleton et al., 1991). Manual estimation was performed by educational experts, such as teachers and course instructors, who assigned difficulty labels to each question (Abdelrahman et al., 2023). However, manual estimation has the drawback of varying results based on the subjective judgment of experts (Huang et al., 2017). By contrast, QDE using the IRT predicts difficulty based on student question-solving records, thereby minimizing subjective bias. This method offers the advantages of explainability, and the ability to track changes in the abilities of students and difficulties of questions over time (Benedetto et al., 2020). However, a significant limitation lies in the need to collect vast amounts of student question-solving records.

To overcome these limitations, recent studies have explored new approaches using natural language processing (NLP) techniques to perform QDE based on textual information. For instance, the study (Huang et al., 2017) employed a TACNN, a CNN-based sentence classifier, and attention layers to estimate question difficulty from a text-based perspective. Leveraging the powerful language understanding capabilities of transformer-based pre-trained language models (PLMs), studies (Benedetto et al., 2021; Fang et al., 2019; Tong et al., 2020; Zhou and Tao, 2020) have utilized PLMs to comprehend the textual information of questions and answers in QDE.

NLP-based QDE methodologies have various advantages; however, they solely focus on the in-

[1] https://www.coursera.org/
[2] https://www.udemy.com/

formation of the questions themselves, not on the students solving them. The same question may have different difficulty levels depending on the proficiency level of the student group. Although it is possible to address this aspect using the difficulty of each question measured through the IRT based on the question-solving records of students for training, drawbacks are still present. These include the requirement for separate question-solving records and the need to train models for each student group.

To address these limitations, we focus on the general question-solving capabilities of large language models (LLMs). Noting the achievement of LLMs of human-level performance across diverse domains (OpenAI, 2023; Street et al., 2024), we hypothesize that LLMs can substitute for students at various levels. Based on this hypothesis, we propose a novel framework, LLMs are students at various levels (LLaSA). In LLaSA, we targeted the abilities of student groups to form LLM clusters with question-solving abilities similar to those of students. Considering LLMs as representatives of students, LLaSA can effectively predict the question difficulty perceived by student groups using the question-solving records of LLMs. In contrast to traditional QDE methods, our approach can easily adapt to changes in the perceived difficulty of questions among different student groups by modifying the composition of the LLMs.

In particular, LLaSA utilizes individual student ability levels derived from the IRT to form an LLM cluster that represents the student group. Typically, LLaSA requires student question-solving records to estimate these abilities. However, if alternative information is available (e.g., grades and levels), LLaSA can perform QDE without any question-solving records. To demonstrate this, we propose a zero-shot LLaSA that performs QDE using alternative information about student abilities without any question-solving records.

To validate the effectiveness of our approach, we evaluated on two QDE benchmarks: DBE-KT22 (Abdelrahman et al., 2023) and ASSISTMents 2005–2006 (Heffernan and Heffernan, 2014). Regarding the performance in regressing the question difficulty, LLaSA achieved a performance comparable to those of state-of-the-art (SOTA) QDE models, despite not being trained itself. Remarkably, in the classification setting, LLaSA achieved SOTA performance on both benchmarks. In correlation with question difficulty derived from student IRT results, the zero-shot LLaSA demonstrated over 74% relative performance compared with the strongest baseline. This result provides strong evidence of the ability of the method to substitute students using only approximate distributions and without any student question-solving records.

In summary, our contributions are three-fold:

- We propose a novel framework, LLaSA, in which LLMs solve the question and use the IRT to estimate the difficulty of the question even though students have not solved the question.
- We utilize various LLMs and prompting techniques to represent students at various levels, successfully simulating their distribution and demonstrating effectiveness on benchmarks.
- We perform a comprehensive analysis of the effectiveness of LLaSA in the QDE task, presenting an in-depth analysis of the efficacy of both LLaSA and zero-shot LLaSA compared to various baselines.

2 Method

Our framework, LLaSA, estimates question difficulty by performing the IRT on LLM-generated question-solving records. In Section 2.1, we describe the methods used to answer the questions using LLMs within the LLaSA framework. In Section 2.2, we describe LLaSA, which performs the IRT on the question-solving results of students to estimate their abilities and select similar LLM clusters. In Section 2.3, we describe the zero-shot LLaSA, which assigns student groups into low/middle/high ability categories based on teacher intuition, and selects the appropriate LLMs.

2.1 Question-solving with LLMs

Various Levels of LLMs We represent the abilities of students at various levels in LLMs by utilizing their structural diversity and training techniques. To reflect the range of innate and acquired abilities in the student population, we consider the model size of the LLMs, training techniques such as pre-training and alignment tuning (e.g., reinforcement learning from human feedback (Ouyang et al., 2022)), and the data used in pre-training. Based on these criteria, 65 LLMs are selected to reflect the student population diversity. As shown on the left side of Figure 1-a, these diverse LLMs resolve these questions.

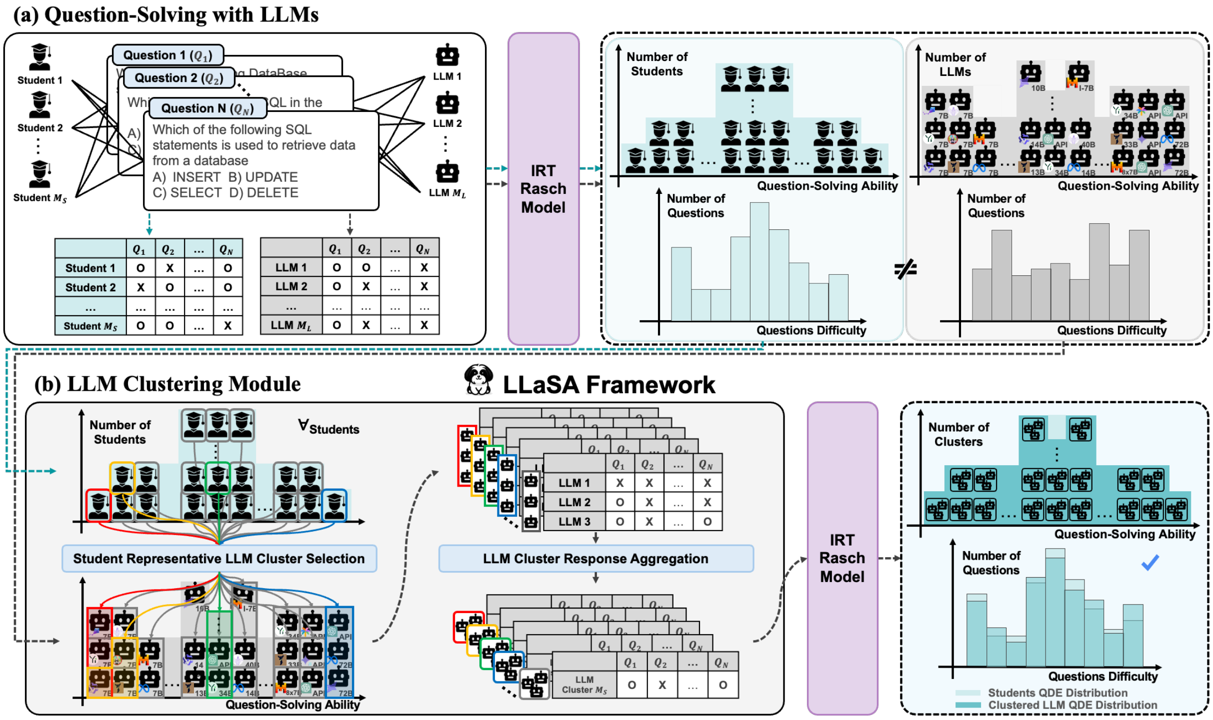

Figure 1: Overview of LLaSA. (a) Performing IRT to the question-solving records of students and LLMs to extract ability. (b) Using IRT results to select LLM clusters that substitute students, aggregate the question-solving results of LLM clusters, and re-perform IRT to estimate the question difficulty as perceived by the simulated students.

Question-solving Prompting Technique LLMs demonstrate in-context learning abilities that allow them to perform new tasks without additional training (Brown et al., 2020). Because LLMs are based on a causal language modeling architecture, various inference methods have been designed to solve multiple choice questions (MCQs) with LLMs (Zhao et al., 2021; Brown et al., 2020; Holtzman et al., 2021; Min et al., 2022). Considering the aspects of performance and inference efficiency, we follow the multiple choice prompt (MCP) method from the previous study (Robinson and Wingate, 2023).

To further leverage the question-solving ability of LLMs, we utilize prompting techniques in conjunction with MCP such as process of elimination (POE) (Ma and Du, 2023), chain-of-thought (CoT) (Wei et al., 2022), and plan-and-solve (PS) (Wang et al., 2023b). Across all question-solving methods, we experiment with zero-, 1-, 3-, and 5-shot prompting. For the models used via the OpenAI API[3], we conduct further experiments with 10-, 20-, and 30-shot prompting owing to its extended context length. In addition, we utilize GPT-4 (OpenAI, 2023) to generate hints for questions, use them to enhance the question-solving capabilities of LLMs.

[3] https://www.openai.com/api/

2.2 LLaSA

2.2.1 LLM Clustering Module

To effectively substitute for the question-solving abilities of students, we propose an LLM clustering module. This module includes three key parts: IRT for QDE, student representative LLM cluster selection, and LLM cluster response aggregation.

IRT for QDE In this study, we use the Rasch model (Rasch, 1960) for IRT to estimate question difficulty and extract abilities from LLM question-solving records. The Rasch model assigns an ability level α_m to each student m and a difficulty level β_n to each item (i.e., question) n, defined as follows:

$$p_{nm} = \frac{\exp(\alpha_m - \beta_n)}{1 + \exp(\alpha_m - \beta_n)}, \qquad (1)$$

where β_n denotes question difficulty and α_m denotes student ability. The question response function p_{nm} is defined as the probability that a student with ability α_m will correctly answer a question with difficulty β_n. The IRT allows for the simultaneous estimation of student ability and question difficulty using maximum likelihood estimation. Figure 1-a illustrates the process of solving questions, performing the IRT, and estimating the abilities and difficulties in LLaSA.

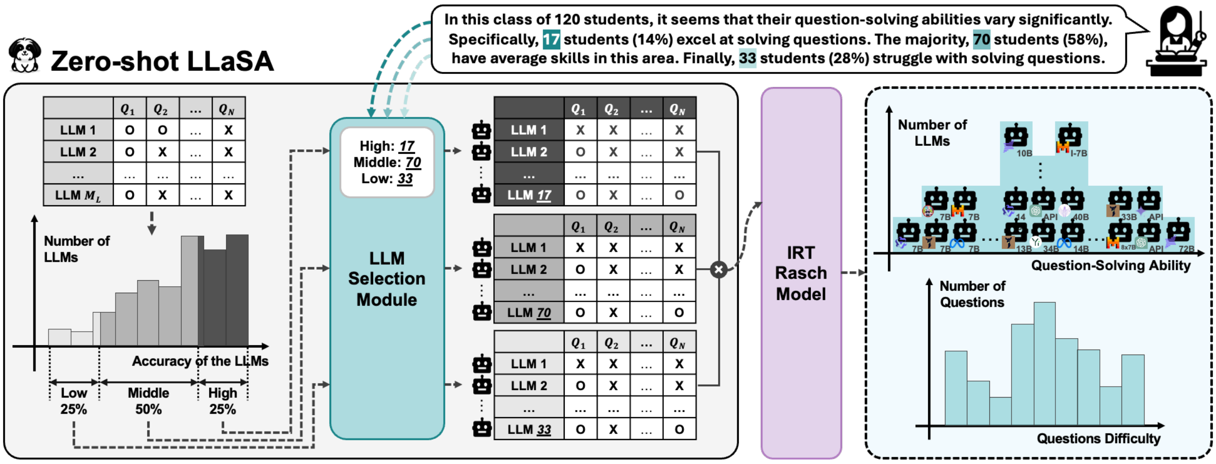

Figure 2: Overview of zero-shot LLaSA. The zero-shot LLaSA estimates student and LLM abilities as low, middle or high. The LLM selection module compose LLMs based on these groups and performs IRT to estimate student abilities and question difficulty.

Student Representative LLM Cluster Selection Based on the abilities of the students and LLMs obtained through the IRT, we select LLM clusters as substitutes for the students. We identify the top-k LLMs whose abilities closely match those of individual students. The process involves calculating the difference in ability between each student and each LLM, and thereafter selecting the top-k LLMs with the smallest difference for each student. These top-k LLMs collectively represent the question-solving capabilities of the students, ensuring accurate and reliable substitution. This process is illustrated on the left side of Figure 1-b.

LLM Cluster Response Aggregation During the course of our research, substituting each student with a single LLM has proven challenging to achieve the same question-solving performance. To overcome this, we utilize LLM clusters. As shown in the middle of Figure 1-b, we aggregate the LLM responses to substitute for student responses. If any LLM within a cluster correctly solves the question, the expected outcome of the LLM cluster is considered correct. Our method ensures that the response pattern of each student is closely represented by the LLM cluster, leveraging the ability of LLMs.

2.2.2 LLM Distribution Adjustment

To further enhance the LLM cluster selection performance, we introduce a selective methodology, the LLM distribution adjustment (LLMDA). LLMDA adjusts the composition of the LLM pool when sufficient student question-solving records are available, thereby forming more effective LLM clusters. The LLMDA method involves randomly removing 1–10 LLMs from the LLM pool, re-estimating the abilities of the remaining LLMs using the Rasch model, and iteratively evaluating their performance. LLMDA flexibly refines the LLM pool, ensuring that the LLM clusters are as representative and accurate as possible.

2.3 Zero-shot LLaSA

In the LLM cluster selection process, LLaSA utilizes the question-solving records of students to obtain information regarding their abilities. However, LLM cluster selection can proceed without the question-solving records of students if alternative information representing their abilities (e.g., grades and levels) is available. To demonstrate the effectiveness of LLaSA in scenarios without question-solving records, we propose a zero-shot LLaSA.

Figure 2 illustrates an example in which a teacher has an approximate understanding of the distribution of student levels. The LLM selection module of zero-shot LLaSA utilizes information such as the number of students at high, medium, and low proficiency levels. It then combines the information with the proficiency levels of the LLMs to configure an LLM cluster that represents a student group. In this study, we use the number of high-, medium-, and low-performing students within a group as approximate information. However, with slight modifications, various types of information such as grades or levels can be utilized.

LLM Selection Module To evaluate the proficiency level of LLMs, we divide the levels based on their question-solving accuracy. Rather than dividing by relative ranking, we categorize the proportion of performance they achieved relative to the highest-performing LLM. For instance, if the highest performing LLM has 0.8 accuracy, then LLMs with 0.6–0.8 accuracy (75%–100% of 0.8) are grouped into the high-level cluster. Those with 0.0–0.2 accuracy (0%–25% of 0.8) are grouped into the low-level cluster, and the remainder are placed into the medium-level cluster.

The importance of this approach lies in the fact that the distribution of question-solving abilities in LLMs does not mirror that of students. Generally, LLMs demonstrate question-solving abilities similar to those of students. However, unlike the normally distributed abilities of students, the abilities of LLMs exhibit significant polarization, with extremely few falling within the mid-range. Using this approach, LLaSA can effectively construct an LLM pool that substitutes for students, regardless of differences in question-solving ability distributions between LLMs and student groups.

3 Experiments

3.1 Datasets

To verify the effectiveness of LLaSA, we used two QDE benchmarks. DBE-KT22 was collected from a relational database course at the Australian National University and included MCQ data and responses from 131 students who answered 206 questions. ASSISTMents 2005–2006 features math questions solved by 8th-grade students. Images were converted to text, and short-answer questions were transformed into the MCQ format for LLMs. We used data from 1,194 students who answered more than the median number of 233 questions. For zero-shot LLaSA, we categorized students based on their question-solving accuracy, used as teacher intuition. DBE-KT22 and ASSISTMents are provided under licenses that allow for academic use, and we have used them for research purposes. In addition, both datasets have undergone de-identification to ensure privacy and safety. More details are described in Appendix B.3.

3.2 Models

In this study, we selected 65 LLMs by considering the model size, base architecture, and training techniques. We selected models with diverse architectures, including Llama 3 (AI@Meta, 2024), Mistral (Jiang et al., 2023), and Falcon (Almazrouei et al., 2023), along with similarly structured but differently trained models, such as Mistral and Solar (Kim et al., 2024). We also included variants of the same model with different sizes, such as Llama 3_{8B} and Llama 3_{70B}. Additionally, API-based models, such as GPT-3.5 and GPT-4 (OpenAI, 2023), were included to ensure that various LLMs participated in question-solving. The list of LLMs used in this study is provided in Appendix A.1.

3.3 Metrics

In this study, the root mean square error (RMSE) (Willmott and Matsuura, 2005) and Pearson correlation (P-Corr) (Freedman et al., 2007) were employed to evaluate the QDE regression effectiveness. In addition, we segmented the question difficulty into equal intervals and transformed it into a 6-class classification task, and F1-score (F1) (Chinchor, 1992) was used to evaluate performance.

3.4 Baselines

To demonstrate the efficacy of our methodology, we selected several baseline methods. We included the R2DE (Benedetto et al., 2020) model, which uses TF-IDF to extract features from question-related texts and employs random forest regression to predict the IRT difficulty. The TACNN model, which combines a CNN-based sentence classifier with attention layers, was also included. In addition, we considered recent QDE models utilizing PLMs such as $\text{BERT}_{\text{base/large}}$ and DistilBERT. We also included custom baselines like $\text{RoBERTa}_{\text{base/large}}$ (Liu et al., 2019) and $\text{DeBERTaV3}_{\text{base/large}}$ (He et al., 2023), and using low-rank adaptation (LoRA) (Hu et al., 2022) to tune the LLMs for QDE tasks. Specifically, we fine-tuned Llama 3_{8B} and Gemma 3_{7B} (Team et al., 2024) using LoRA.

3.5 Experimental Details

In our training process on baselines, we conducted experiments with various combinations of hyperparameters and reported the results averaged on five different random seeds. When conducting experiments on LLMs, the temperature was fixed at 0. All experiments were conducted with PyTorch[4] and HuggingFace Transformers (Wolf et al., 2020) on three NVIDIA A100 GPUs, with IRT performed using mirt (Chalmers, 2012). More experimental details are provided in the Appendix A.

3.6 QDE Results of LLaSA

Unlike baselines that train on the difficulty of each question derived from the IRT results using

[4]https://pytorch.org/

System	DBE-KT22				ASSISTMents			
	Full dataset		Sampled dataset		Full dataset		Sampled dataset	
	RMSE	F1	RMSE ($\Delta\delta$)	F1 ($\Delta\delta$)	RMSE	F1	RMSE ($\Delta\delta$)	F1 ($\Delta\delta$)
Published								
R2DE	$1.394_{0.04}$	$0.245_{0.02}$	**$1.556_{0.05}$** (-11.67%)	$0.253_{0.02}$ (3.27%)	$1.155_{0.04}$	$0.278_{0.04}$	$1.142_{0.04}$ (1.18%)	$0.223_{0.05}$ (-20.04%)
TACNN	$1.637_{0.02}$	$0.257_{<0.01}$	$1.787_{0.01}$ (-9.16%)	$0.256_{0.01}$ (-0.70%)	$1.139_{<0.01}$	$0.290_{0.01}$	$1.341_{0.03}$ (-17.77%)	$0.292_{0.02}$ (0.76%)
$\text{BERT}_{\text{base}}$	$1.482_{0.04}$	$0.213_{0.04}$	$1.867_{0.50}$ (-25.92%)	$0.229_{0.05}$ (7.50%)	$1.201_{0.08}$	$0.313_{0.03}$	$1.118_{0.01}$ (6.88%)	$0.181_{<0.01}$ (-42.25%)
$\text{BERT}_{\text{large}}$	$1.400_{0.04}$	$0.221_{0.03}$	$1.915_{0.56}$ (-36.78%)	$0.247_{0.01}$ (11.97%)	$1.135_{0.07}$	$0.273_{0.09}$	$1.185_{0.07}$ (-4.39%)	$0.192_{0.02}$ (-29.42%)
DistillBERT	$1.517_{0.03}$	$0.226_{0.03}$	$1.602_{0.13}$ (-5.61%)	$0.219_{0.02}$ (-3.45%)	**$1.091_{<0.01}$**	$0.211_{0.05}$	**$1.101_{<0.01}$** (-0.93%)	$0.181_{<0.01}$ (-14.38%)
Additional Systems								
$\text{RoBERTa}_{\text{base}}$	**$1.382_{0.08}$**	$0.261_{0.04}$	$1.684_{0.07}$ (-21.88%)	$0.261_{0.03}$ (0.23%)	$1.098_{<0.01}$	$0.214_{0.05}$	$1.197_{0.04}$ (-9.09%)	$0.183_{<0.01}$ (-14.50%)
$\text{RoBERTa}_{\text{large}}$	$1.465_{0.03}$	$0.226_{0.03}$	$1.595_{0.14}$ (-8.88%)	$0.204_{0.04}$ (-9.57%)	$1.094_{<0.01}$	$0.223_{0.05}$	$1.166_{0.04}$ (-6.53%)	**$0.335_{0.03}$** (49.78%)
$\text{DeBERTaV3}_{\text{base}}$	$1.499_{0.08}$	$0.242_{0.02}$	$1.621_{0.14}$ (-8.13%)	$0.224_{0.03}$ (-7.76%)	$1.111_{0.02}$	$0.180_{<0.01}$	$1.195_{0.02}$ (-7.58%)	$0.181_{<0.01}$ (0.11%)
$\text{DeBERTaV3}_{\text{large}}$	$1.518_{0.07}$	$0.239_{0.04}$	$1.660_{0.04}$ (-9.39%)	$0.233_{0.02}$ (-2.26%)	$1.112_{<0.01}$	$0.230_{0.05}$	$1.113_{0.01}$ (-0.09%)	$0.234_{0.05}$ (1.74%)
$\text{Llama3}_{\text{8B}}$ w/ LoRA	$2.025_{0.21}$	$0.241_{0.03}$	$2.228_{0.23}$ (-10.01%)	$0.241_{0.05}$ (-0.08%)	$2.328_{0.46}$	$0.253_{0.02}$	$2.215_{0.39}$ (4.83%)	$0.226_{0.08}$ (-10.51%)
Gemma_{7B} w/ LoRA	$2.771_{0.64}$	$0.180_{0.04}$	$4.001_{1.18}$ (-44.38%)	$0.186_{0.03}$ (3.22%)	$2.183_{0.73}$	$0.262_{0.03}$	$2.650_{0.40}$ (-21.38%)	$0.207_{0.06}$ (-21.11%)
Ours								
LLaSA w/o LLMDA	$1.858_{<0.01}$	$0.295_{<0.01}$	$1.764_{<0.01}$ (5.06%)	**$0.334_{<0.01}$** (13.22%)	$1.589_{<0.01}$	$0.183_{<0.01}$	$1.602_{<0.01}$ (-0.82%)	$0.246_{<0.01}$ (34.43%)
LLaSA w/ LLMDA	$1.640_{0.02}$	**$0.321_{0.02}$**	$1.668_{<0.01}$ (-1.66%)	$0.322_{0.03}$ (0.31%)	$1.611_{0.04}$	**$0.338_{0.02}$**	$1.614_{0.02}$ (-0.20%)	$0.298_{0.04}$ (-12.00%)
Zero-shot LLaSA	$2.360_{0.04}$	$0.150_{0.01}$	$2.360_{0.04}$ (=)	$0.150_{0.01}$ (=)	$1.323_{<0.01}$	$0.274_{0.01}$	$1.323_{<0.01}$ (=)	$0.274_{0.01}$ (=)

Table 1: Experimental results (with standard deviation) on DBE-KT22 and ASSISTMents, using full and sampled datasets. $\Delta\delta$ shows the improvement rate between full and sampled datasets. Zero-shot LLaSA shows no difference as it doesn't utilize student data. The best results are **boldfaced**, and the second-best results are underlined.

student question-solving records, LLaSA sets up LLM clusters. These clusters can substitute for students based on their abilities. It then estimates the question difficulty by performing the IRT on the question-solving results of the LLM clusters.

To verify the efficacy of our approach on small question-solving data, we experimented with both full and sampled datasets, using approximately 50% of the questions for the latter. In a sampled dataset, the baseline methods train on the question difficulty from the IRT results performed with fewer questions. The LLaSA adjusts the LLM clusters based on the student question-solving ability from these IRT results, which were also performed with fewer questions. Both approaches suffer from reduced IRT performance owing to the limited amount of question data in the sampled dataset, leading to a decline in the overall performance.

Full Dataset As summarized in Table 1, our evaluation results indicate that the LLaSA outperformed the baselines. In the classification setting on DBE-KT22, LLaSA with LLMDA achieved the best F1 of 0.321 among the baselines, reaching SOTA performance, followed by LLaSA without LLMDA. On ASSISTMents, LLaSA with LLMDA achieved the best F1 of 0.338, significantly outperforming the other baselines. In the regression setting, LLaSA exhibited a minimal RMSE difference of only 0.258 on DBE-KT22 and 0.531 on ASSISTMents, compared to the best performing baseline. Remarkably, the zero-shot LLaSA achieved an RMSE of 1.323 on ASSISTMents, outperforming LLaSA and exhibiting little difference from the baseline models. However, on DBE-KT22, the zero-shot LLaSA demonstrated poor performance.

System	DBE-KT22			
	Full dataset		Sampled dataset	
	P-Corr	P-value	P-Corr	P-value
Published				
R2DE	$0.436_{0.02}$	<0.05	$0.274_{0.03}$	<0.05
TACNN	$-0.212_{0.02}$	<0.05	$0.282_{<0.01}$	<0.05
$\text{BERT}_{\text{base}}$	$0.368_{0.03}$	<0.05	$0.316_{0.02}$	<0.05
$\text{BERT}_{\text{large}}$	$0.424_{0.02}$	<0.05	$0.293_{0.02}$	<0.05
DistillBERT	$0.371_{0.02}$	<0.05	**$0.374_{0.04}$**	<0.05
Additional Systems				
$\text{RoBERTa}_{\text{base}}$	**$0.470_{0.05}$**	<0.05	$0.337_{0.02}$	<0.05
$\text{RoBERTa}_{\text{large}}$	$0.403_{0.05}$	<0.05	$0.313_{0.02}$	<0.05
$\text{DeBERTaV3}_{\text{base}}$	$0.373_{0.03}$	<0.05	$0.297_{0.03}$	<0.05
$\text{DeBERTaV3}_{\text{large}}$	$0.370_{0.03}$	<0.05	$0.319_{0.05}$	<0.05
$\text{Llama3}_{\text{8B}}$ w/ LoRA	$0.225_{0.07}$	0.055	$0.210_{0.07}$	0.071
Gemma_{7B} w/ LoRA	$0.103_{0.11}$	0.419	$0.109_{0.10}$	0.444
Ours				
LLaSA w/o LLMDA	$0.143_{<0.01}$	0.149	$0.223_{<0.01}$	<0.05
LLaSA w/ LLMDA	$0.233_{0.02}$	<0.05	$0.283_{0.02}$	<0.05
Zero-shot LLaSA	$0.348_{<0.01}$	<0.05	$0.348_{<0.01}$	<0.05

Table 2: The comparison between the student IRT and the prediction of LLaSA, evaluated using P-Corr on the full and sampled DBE-KT22. The best results are **boldfaced**, and the second-best results are underlined.

For further analysis, we compared the P-Corr value between the question difficulty derived from the IRT using the question-solving records of students and the question difficulty predicted by LLaSA on DBE-KT22. As summarized in Table 2, LLaSA achieved a significant P-Corr value of 0.348 on DBE-KT22. Remarkably, zero-shot LLaSA exhibited over 74% relative performance compared with the best result. This result demonstrates its significance and potential in real world scenarios.

Sampled Dataset As summarized in Table 1, even with fewer questions to perform the IRT, LLaSA did not exhibit a significant performance decline. Similar to the full dataset, LLaSA outperformed the other baselines on the sampled dataset. In the classification setting experiments on DBE-KT22, LLaSA without LLMDA achieved the best F1 of 0.334 among the baselines, achieving SOTA performance with a large difference. On ASSISTMents, LLaSA with LLMDA achieved the second-best performance among the baselines, exhibiting little difference from the best-performing baseline. In the regression setting on DBE-KT22, LLaSA with LLMDA exhibited a 1.66% RMSE increase, whereas LLaSA without LLMDA improved by 5.06% and was the least affected by the reduced training dataset. On ASSISTMents, the RMSE changes compared with the full dataset setting for LLaSA with and without LLMDA were only 0.2% and 0.82%, respectively. Notably, the P-Corr for the zero-shot LLaSA on ASSISTMents achieved the second-best performance on the sampled dataset, as summarized in Table 2. This demonstrates that LLaSA maintains robust performance even with limited question-solving records.

4 Analysis of LLaSA

4.1 Question-Solving Based QDE of LLMs

Comparison of Question Difficulty Distribution We compared the QDE results of a strong baseline with those of LLaSA against the question difficulty derived from the question-solving records of students. Figure 3 presents the question difficulty histograms for each dataset. For DBE-KT22, the best-performing model, $\text{RoBERTa}_{\text{base}}$, rarely predicted difficulties above zero, likely because of the scarcity of such values in the training data. In contrast, the predictions of LLaSA closely matched the student IRT distribution, as shown by the kernel density estimation lines. In ASSISTMents, the best result model DistillBERT excessively predicts values at approximately 0. Conversely, LLaSA predicts a broader range of difficulties, similar to the distribution of the student IRT. This analysis highlights the robustness of LLaSA, avoiding the local minimum trap for predicting a single value to minimize the loss in the training process.

Effectiveness of top-k LLM Cluster Selection In Figure 4, we adjust the value of k, the number of LLMs used to substitute for a single student, from one to four. For DBE-KT22, increasing k improve

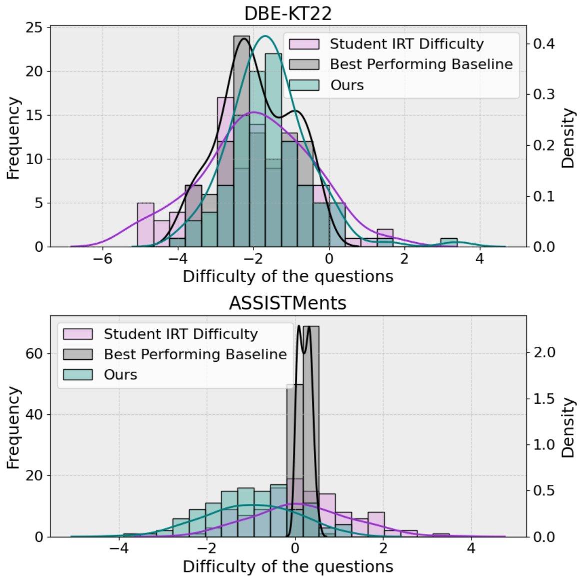

Figure 3: Predicted difficulty histograms for the DBE-KT22 and ASSISTMents comparing student IRT difficulty, the best resulting model, and LLaSA w/ LLMDA.

the RMSE and F1. In contrast, for ASSISTMents, the performance did not consistently improve with higher k. In ASSISTMents, not all students answered every question, limiting the IRT estimation. Therefore, we set k to a maximum of 4 for clustering experiments. The differences in the results across the two datasets are analyzed in Section 4.2.

Effectiveness of LLMDA To evaluate the effectiveness of LLMDA, we conducted experiments with and without LLMDA. As shown in Figure 4 and Table 1, applying LLMDA resulted in better performance for both DBE-KT22 and ASSISTMents. In the sampled DBE-KT22, the method with LLMDA exhibited an improvement over that without LLMDA, and the RMSE was improved by 5.45%. LLMDA enhances performance by allowing the model to more accurately simulate student distributions through a random selection of LLMs. This led to more precise IRT measurements and better functioning of the LLM clustering module.

4.2 LLM Cluster Representation

Representational Capability of LLM Clusters Experiments were conducted to evaluate the effectiveness of the LLM clustering module in representing students. To assess the extent to which the module selected the LLM clusters that represented student question responses, we measured the F1 by comparing the question responses of the LLM clusters with the actual answers of the students on

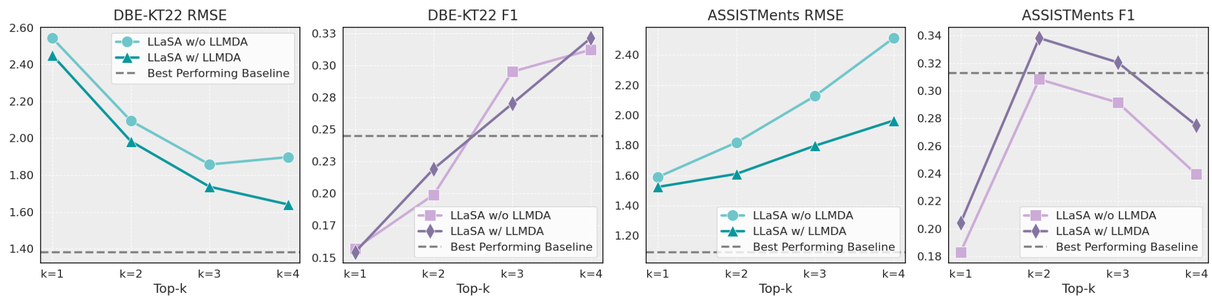

Figure 4: RMSE and F1 for each dataset, comparing the results of applying LLMDA and the top-k of LLM cluster.

DBE-KT22. The module achieved an F1 of 0.752 for the training set and 0.762 for the test set, indicating that it accurately represented student records. The detailed results of the question-solving records of the LLM clusters compared with the actual student responses are provided in Appendix C.1.

Is LLaSA Performing as Intended? In DBE-KT22, models such as Llama 3 and Falcon were frequently adopted, prompting methods used in the order of MCP, CoT, POE, and PS, and the number of few-shot examples in the order of 3-5-0-1. In ASSISTMents, models such as Amber (Street et al., 2024) and Openchat (Wang et al., 2023a) were frequently adopted, prompting methods used in the order of MCP, POE, PS, and CoT, and the number of few-shot examples in the order of 0-1-3-5. The relevant figures are provided in Appendix C.2.

In DBE-KT22, a larger number of LLMs representing students, high-performance models, and prompting methods with a large number of few-shot examples resulted in a better performance. By contrast, in ASSISTMents, a smaller number of LLMs representing students, relatively lower-performance models, and prompting methods with fewer few-shot examples yielded a better performance. Considering the characteristics of the datasets, DBE-KT22 comprises questions aimed at university undergraduates, whereas ASSISTMents comprises questions for 8th-grade students. Remarkably, appropriate LLMs and inference methodologies appear to be adopted according to the question levels and abilities of the student groups.

5 Related Works

5.1 Question-Solving Skills of LLM

Since the release of GPT-3 (Brown et al., 2020), LLMs have rapidly advanced. Notable models such as GPT-4 (OpenAI, 2023) and Llama 3 (AI@Meta, 2024) have emerged, exhibiting billions of parameters and excelling in various NLP tasks. Recently, MCQs have been used to evaluate the reasoning abilities of these models, with LLMs achieving human-like performances. Advanced studies (Robinson and Wingate, 2023; Ma and Du, 2023; Pezeshkpour and Hruschka, 2023) have improved MCQs by eliminating least probable options and reducing bias in answer positioning.

5.2 Question Difficulty Estimation

Traditionally, QDE relies on the IRT (Hambleton et al., 1991) method, which statistically measures question difficulty and learner ability based on responses. Prominent IRT models include the Rasch model (Rasch, 1960) and 2-parameter logistic model; however, they require substantial response data, posing challenges in data-scarce scenarios. To address this issue, recent studies have used text analysis to estimate the difficulty without response data. For instance, one study (Benedetto et al., 2020) used TF-IDF and a random forest regressor to infer difficulty, while another (Xue et al., 2020) utilized ELMo embeddings to predict response times and correct answer probabilities. In addition, research (Benedetto et al., 2021) using BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2020) has explored methods for analyzing question statements and choices to infer difficulty.

6 Conclusion

In this study, we proposed an LLaSA framework by leveraging LLMs to estimate question difficulty in personalized education. The LLaSA demonstrated a competitive performance with strong baseline models, even without extensive training data. The zero-shot LLaSA exhibited a high correlation with the student IRT, indicating its potential for effective real world applications. This study underscores the potential of LLMs in QDE, suggesting that they can substitute for human abilities in various domains.

Limitations

Our study, while introducing a novel framework for QDE, has several limitations. Firstly, due to the lack of publicly available datasets containing questions and student question-solving records, our experiments were limited to the fields of mathematics and computer science. However, institutions with proprietary datasets could adopt LLaSA to gain deeper insights and improve performance. Secondly, LLaSA requires significant storage and computational resources due to the use of multiple LLMs. As LLMs become more efficient and smaller while maintaining their question-solving capabilities, these limitations could be overcome, significantly enhancing the efficiency and effectiveness of the LLaSA framework. Lastly, this progress may pose a risk to jobs currently involved in the QDE domain.

References

Ghodai Abdelrahman, Sherif Abdelfattah, Qing Wang, and Yu Lin. 2023. Dbe-kt22: A knowledge tracing dataset based on online student evaluation. *Preprint*, arXiv:2208.12651.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

AI@Meta. 2024. Llama 3 model card.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *Preprint*, arXiv:2311.16867.

Luca Benedetto, Giovanni Aradelli, Paolo Cremonesi, Andrea Cappelli, Andrea Giussani, and Roberto Turrin. 2021. On the application of transformers for estimating the difficulty of multiple-choice questions from text. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 147–157, Online. Association for Computational Linguistics.

Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. 2020. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, LAK '20, page 412–421, New York, NY, USA. Association for Computing Machinery.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

C Boopathiraj and K Chellamani. 2013. Analysis of test items on difficulty level and discrimination index in the test for research in education. *International journal of social science & interdisciplinary research*, 2(2):189–193.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

R. Philip Chalmers. 2012. mirt: A multidimensional item response theory package for the R environment. *Journal of Statistical Software*, 48(6):1–29.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Nancy Chinchor. 1992. MUC-4 evaluation metrics. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiansheng Fang, Wei Zhao, and Dongya Jia. 2019. Exercise difficulty prediction in online education systems. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 311–317.

D. Freedman, R. Pisani, and R. Purves. 2007. *Statistics: Fourth International Student Edition*. Emersion: Emergent Village Resources for Communities of Faith Series. W.W. Norton & Company.

R.K. Hambleton, H. Swaminathan, and H.J. Rogers. 1991. *Fundamentals of Item Response Theory*. Number V. 2 in Fundamentals of Item Response Theory. SAGE Publications.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Neil T. Heffernan and Cristina Lindquist Heffernan. 2014. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question difficulty prediction for reading problems in standard tests. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

Haleh Jafari, Abbas Aghaei, and Alireza Khatony. 2019. Relationship between study habits and academic achievement in students of medical sciences in kermanshah-iran. *Advances in Medical Education and Practice*, 10:637–643.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Dahyun Kim, Chanjun Park, Sanghoon Kim, Wonsung Lee, Wonho Song, Yunsu Kim, Hyeonwoo Kim, Yungi Kim, Hyeonju Lee, Jihoo Kim, Changbae Ahn, Seonghoon Yang, Sukyung Lee, Hyunbyung Park, Gyoungjin Gim, Mikyoung Cha, Hwalsuk Lee, and Sunghun Kim. 2024. Solar 10.7b: Scaling large language models with simple yet effective depth upscaling. *Preprint*, arXiv:2312.15166.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, Richard Fan, Yi Gu, Victor Miller, Yonghao Zhuang, Guowei He, Haonan Li, Fajri Koto, Liping Tang, Nikhil Ranjan, Zhiqiang Shen, Xuguang Ren, Roberto Iriondo, Cun Mu, Zhiting Hu, Mark Schulze, Preslav Nakov, Tim Baldwin, and Eric P. Xing. 2023. Llm360: Towards fully transparent open-source llms. *Preprint*, arXiv:2312.06550.

Chenkai Ma and Xinya Du. 2023. POE: Process of elimination for multiple choice reasoning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4487–4496, Singapore. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *Preprint*, arXiv:2311.11045.

Yuting Ning, Zhenya Huang, Xin Lin, Enhong Chen, Shiwei Tong, Zheng Gong, and Shijin Wang. 2023. Towards a holistic understanding of mathematical questions with contrastive pre-training. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13409–13418.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *Preprint*, arXiv:2308.11483.

G. Rasch. 1960. *Probabilistic Models for Some Intelligence and Attainment Tests*. Studies in mathematical psychology. Danmarks Paedagogiske Institut.

Joshua Robinson and David Wingate. 2023. Leveraging large language models for multiple choice question answering. In *The Eleventh International Conference on Learning Representations*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Winnie Street, John Oliver Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Blaise Aguera y Arcas, and Robin I. M. Dunbar. 2024. Llms achieve adult human performance on higher-order theory of mind tasks. *Preprint*, arXiv:2405.18870.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Hanshuang Tong, Yun Zhou, and Zhen Wang. 2020. Exercise hierarchical feature enhanced knowledge tracing. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II*, page 324–328, Berlin, Heidelberg. Springer-Verlag.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Guan Wang, Sijie Cheng, Qiying Yu, and Changling Liu. 2023a. OpenLLMs: Less is More for Open-source Models.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023b. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

C. Willmott and K Matsuura. 2005. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research*, 30:79.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kang Xue, Victoria Yaneva, Christopher Runyon, and Peter Baldwin. 2020. Predicting the difficulty and response time of multiple choice questions using transfer learning. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 193–197, Seattle, WA, USA → Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Ya Zhou and Can Tao. 2020. Multi-task bert for problem difficulty prediction. In *2020 International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 213–216.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness harmlessness with rlaif.

Appendix

A Experimental Settings

A.1 List of LLMs Used in LLaSA

LLaSA utilizes various LLMs with comprehensive question-solving capabilities to substitute for students in answering questions. We employed 65 models ranging in size from 125M to 70B parameters, including various API-based models such as GPT-4. The LLMs used are Amber, Crystal (Liu et al., 2023), Falcon (Almazrouei et al., 2023), GPT-J (Wang and Komatsuzaki, 2021), GPT-Neo (Black et al., 2021), GPT-3.5, GPT-4 (OpenAI, 2023), Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024), OpenChat (Wang et al., 2023a), OPT (Zhang et al., 2022), Orca (Mitra et al., 2023), Pythia (Biderman et al., 2023), Solar (Kim et al., 2024), Starling (Zhu et al., 2023), Llama 1 (Touvron et al., 2023a), Llama 2 (Touvron et al., 2023b), Llama 3 (AI@Meta, 2024), Vicuna (Chiang et al., 2023), Yi (AI et al., 2024), and Zephyr (Tunstall et al., 2023). These models were sourced from the Huggingface Transformers library (Wolf et al., 2020) and the OpenAI API. A detailed list can be found in Table 4.

A.2 Question-Solving Prompts

In the DBE-KT22 and ASSISTments datasets, we utilized the MCP, POE, CoT, and PS prompting techniques for LLM question-solving. The specific prompts for each technique used in question-solving are detailed in Table 5 and Table 6.

A.3 Details of Baseline Experiments

Our baseline models included R2DE, TACNN, PLMs, and LLMs with LoRA. To comprehensively compare their performance with LLaSA, we first optimized the baseline models through extensive hyperparameter tuning.

For R2DE, we tuned the number of estimators {10, 25, 50, 100, 150, 200, 250} and the max depth {2, 5, 10, 15, 25, 50} in RandomForest. For TACNN, we tuned the learning rates {5e-5, 2e-5, 5e-6} and batch sizes {8, 16, 32}. For PLMs, we tuned the learning rates {2e-6, 5e-6, 2e-5, 5e-5} and batch sizes {16, 32}. For LLMs with LoRA, we tuned the learning rates {2e-6, 5e-6, 2e-5}, batch sizes {16, 32}, and LoRA parameters such as alpha {4, 8} and r as alpha * 2. Using these optimized hyperparameters, we trained and evaluated the models across five different seeds. We averaged the results and calculated the standard deviation to ensure a robust baseline experiment.

The R2DE model was implemented using publicly available code, TACNN was implemented manually, and PLM and LLM models with LoRA were implemented using the PyTorch-based Huggingface Transformers library. All experiments were conducted on three NVIDIA A100 GPUs.

B Implementation Details of LLaSA

B.1 IRT for LLaSA

LLaSA estimate question difficulty based on students' abilities derived from IRT. To achieve this, question-solving records are input into the IRT model. We used the R package mirt (Chalmers, 2012) to perform IRT analysis, estimating students' abilities and question difficulties. This allowed us to obtain each student's ability level and the perceived difficulty of questions based on their question-solving records.

B.2 LLM Clustering Module of LLaSA

LLaSA includes a LLM Clustering Module, which consists of LLM cluster selection and LLM Cluster Response Aggregation. In LLM cluster selection, question-solving records (transactions) are input into IRT to measure the question-solving ability of respondents and the difficulty of questions based on these respondents. Each student's ability is then used to select top-k LLMs with similar abilities, forming an LLM Cluster.

In the LLM Cluster Response Aggregation, the question-solving records of the selected LLM Cluster are aggregated using sum aggregation. This process of the LLM clustering module simulates the question-solving records of an individual student. Finally, the aggregated question-solving records of the LLM Cluster are input into IRT to measure the question-solving ability of the LLM Cluster and the difficulty of questions from their perspective. For more details, in Algorithm 1.

B.3 Zero-Shot LLaSA

Zero-shot LLaSA typically requires teacher intuition to categorize students. However, lacking this intuitive understanding, we categorized students by their question-solving accuracy. For DBE-KT22, we selected 31 students with accuracy ≤ 0.75 (low), 69 with accuracy between 0.75 and 0.85 (middle), and 31 with accuracy > 0.85 (high). For ASSISTMents, we sampled 20% from each group: 146 with

Algorithm 1 LLM Clustering Module

Input:
$T_{S_{\text{train}}}$: Student train questions transactions
$T_{L_{\text{train}}}$: LLM train questions transactions
$T_{S_{\text{test}}}$: Student test questions transactions
$T_{L_{\text{test}}}$: LLM test questions transactions
k: Number of top similar LLMs to identify
Initialize:
$LC \leftarrow \emptyset$: Dictionary of Students with LLM Clusters as Values
$T_{LC} \leftarrow \emptyset$: LLM Cluster's Aggregated responses
Rasch: Function returning ability α and difficulty β parameters for question transactions
LLM cluster selection:
$\alpha_S, \beta_S \leftarrow \text{Rasch}(T_{S_{\text{train}}})$
$\alpha_L, \beta_L \leftarrow \text{Rasch}(T_{L_{\text{train}}})$
for each student s and ability α_s in α_S **do**
  $\Delta\alpha_i = |\alpha_s - \alpha_l| \quad \forall l \in L$
  Sort LLMs by $\Delta\alpha_i$ in ascending order
  Select top k LLMs: $\{L_{(1)}, L_{(2)}, \ldots, L_{(k)}\}$
  $LC[s] \leftarrow \{L_{(1)}, L_{(2)}, \ldots, L_{(k)}\}$
LLM Cluster Response Aggregation:
for each student s and LLM Cluster l in $LC.items()$ **do**
  $t_{LC} \leftarrow \mathbf{0}$: Zero vector of length $|T_{L_{\text{test}}}[0]|$
  for each LLM l in L **do**
    $t_{LC} \leftarrow \text{sum}(t_{LC}, T_{L_{\text{test}}}[l], \text{axis} = 1)$
  $t_{LC} \leftarrow \text{clip}(t_{LC}, 0, 1)$
  Append t_{LC} to T_{LC}
$\alpha_{LC}, \beta_{LC} \leftarrow \text{Rasch}(T_{LC})$

accuracy ≤ 0.5, 61 with accuracy between 0.5 and 0.67, and 30 with accuracy > 0.67.

C Addtional Analysis

C.1 Evaluation Student Representation of LLM Clustering

We evaluated the effectiveness of the LLM Clustering module in LLaSA by assessing how well it represents students on the DBE-KT22 dataset. To accomplish this, we used the question-solving records of the students as the ground truth labels and the question-solving records of the LLM Clusters as the predictions. This comparison allowed us to evaluate the effectiveness of the LLM Clustering module in LLaSA. We measured this by calculating accuracy, precision, recall, and F1, and the results of this evaluation are presented in Table 3.

	F1	Accuracy	Recall	Precision
Training Set Responses	0.752	0.665	0.711	0.821
Test Set Responses	0.762	0.678	0.752	0.804

Table 3: Evaluating the LLM clusters prediction of students question-solving records, the table shows results for both training set questions and test set questions in the DBE-KT22 dataset.

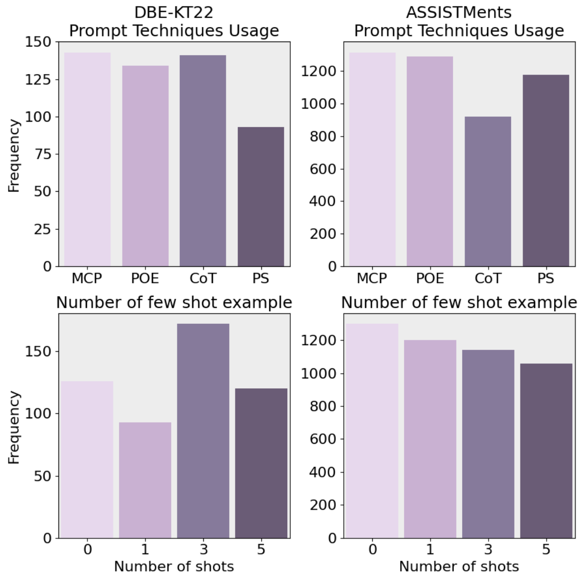

Figure 5: Histograms of prompting techniques and the number of few-shot examples used in LLM clusters for each dataset.

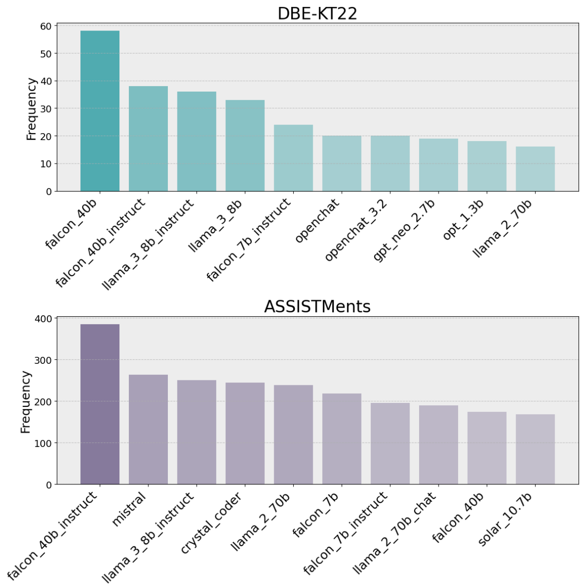

Figure 6: Histograms of models used in LLM clusters for each dataset.

C.2 Models and Prompting Techniques Used in the LLM Clusters

LLaSA uses various models with different prompting techniques and example counts to represent students. Each model used MCP, POE, PS, and CoT techniques to solve questions with zero-, 1-, 3-, or 5-shot examples. Additionally, LLaSA's clustering module selected LLMs most similar to each student's ability, constructing LLM clusters to represent students. We aimed to analyze the diversity of prompting techniques and models used in this process. Figures 5 and 6 illustrate the distribution of LLMs selected for the LLM clusters, as well as the number of shots for the adopted prompting techniques and model in the DBE-KT22 and ASSISTMents Full datasets. The analysis results are discussed in Section 4.2.

LLaSA employs various models with different prompting techniques and example counts to represent students. Each model utilized MCP, POE, PS, and CoT techniques to solve questions with zero-, 1-, 3-, or 5-shot examples. Additionally, LLaSA's clustering module selected LLMs most similar to each student's ability, constructing LLM clusters to represent students.

We aimed to analyze the diversity of prompting techniques and models used in this process. Figures 5 and 6 illustrate the histogram of LLMs selected for the LLM clusters, as well as the number of shots and models used in the DBE-KT22 and ASSISTments Full datasets. The analysis results are discussed in Section 4.2.

Model Name	HF Model Name	Model URL	Base Architecture	Model Size
Amber	amber	https://huggingface.co/LLM360/Amber	Llama	7B
Amber	amber_chat	https://huggingface.co/LLM360/AmberChat	Llama	7B
CrystalChat	crystal_chat	https://huggingface.co/LLM360/CrystalChat	Llama	7B
CrystalCoder	crystal_coder	https://huggingface.co/LLM360/CrystalCoder	Llama	7B
Falcon	falcon_40b	https://huggingface.co/tiiuae/falcon-40b	Llama	40B
Falcon	falcon_40b_instruct	https://huggingface.co/tiiuae/falcon-40b-instruct	Llama	40B
Falcon	falcon_7b	https://huggingface.co/tiiuae/falcon-7b	Llama	7B
Falcon	falcon_7b_instruct	https://huggingface.co/tiiuae/falcon-7b-instruct	Llama	7B
GPT-J	gpt_j_6b	https://huggingface.co/EleutherAI/gpt-j-6b	GPT2	6B
GPT-Neo	gpt_neo_1.3b	https://huggingface.co/EleutherAI/gpt-neo-1.3B	GPT2	1.3B
GPT-Neo	gpt_neo_125m	https://huggingface.co/EleutherAI/gpt-neo-125m	GPT2	125M
GPT-Neo	gpt_neo_2.7b	https://huggingface.co/EleutherAI/gpt-neo-2.7B	GPT2	2.7B
GPT-Neo	gpt_neox_20b	https://huggingface.co/EleutherAI/gpt-neox-20b	GPT2	20B
GPT 3.5	-	https://openai.com/index/openai-api/	OpenAI	unknown
GPT 4	-	https://openai.com/index/openai-api/	OpenAI	unknown
Llama 2	llama_2_13b	https://huggingface.co/meta-llama/Llama-2-13b	Llama	13B
Llama 2	llama_2_13b_chat	https://huggingface.co/meta-llama/Llama-2-13b-chat-hf	Llama	13B
Llama 2	llama_2_70b	https://huggingface.co/meta-llama/Llama-2-70b	Llama	70B
Llama 2	llama_2_70b_chat	https://huggingface.co/meta-llama/Llama-2-70b-chat-hf	Llama	70B
Llama 2	llama_2_7b	https://huggingface.co/meta-llama/Llama-2-7b	Llama	7B
Llama 2	llama_2_7b_chat	https://huggingface.co/meta-llama/Llama-2-7b-chat-hf	Llama	7B
Llama 3	llama_3_70b	https://huggingface.co/meta-llama/Meta-Llama-3-70B	Llama	70B
Llama 3	llama_3_70b_instruct	https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct	Llama	70B
Llama 3	llama_3_8b	https://huggingface.co/meta-llama/Meta-Llama-3-8B	Llama	8B
Llama 3	llama_3_8b_instruct	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct	Llama	8B
Mistral	mistral	https://huggingface.co/mistralai/Mistral-7B-v0.1	Llama	7B
Mistral	mistral_chat	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1	Llama	7B
Mixtral	mixtral	https://huggingface.co/mistralai/Mixtral-8x7B-v0.1	Llama	47B
Mixtral	mixtral_chat	https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1	Llama	47B
OpenChat	openchat	https://huggingface.co/openchat/openchat_8192	Llama	13B
OpenChat	openchat_2	https://huggingface.co/openchat/openchat_v2	Llama	13B
OpenChat	openchat_2_w	https://huggingface.co/openchat/openchat_v2_w	Llama	13B
OpenChat	openchat_3.2	https://huggingface.co/openchat/openchat_3.5	Llama	13B
OpenChat	openchat_3.2_super	https://huggingface.co/openchat/openchat_v3.2_super	Llama	13B
OPT	opt_1.3b	https://huggingface.co/facebook/opt-1.3b	GPT2	1.3B
OPT	opt_125m	https://huggingface.co/facebook/opt-125m	GPT2	125M
OPT	opt_2.7b	https://huggingface.co/facebook/opt-2.7b	GPT2	2.7B
OPT	opt_350m	https://huggingface.co/facebook/opt-350m	GPT2	350M
Orca	orca_2_13b	https://huggingface.co/microsoft/Orca-2-13b	Llama	13B
Orca	orca_2_7b	https://huggingface.co/microsoft/Orca-2-7b	Llama	7B
Pythia	pythia_1.4b	https://huggingface.co/EleutherAI/pythia-1.4b	GPT2	1.4B
Pythia	pythia_12b	https://huggingface.co/EleutherAI/pythia-12b	GPT2	12B
Pythia	pythia_1b	https://huggingface.co/EleutherAI/pythia-1b	GPT2	1B
Pythia	pythia_2.8b	https://huggingface.co/EleutherAI/pythia-2.8b	GPT2	2.8B
Pythia	pythia_410m	https://huggingface.co/EleutherAI/pythia-410m	GPT2	410M
Pythia	pythia_6.9b	https://huggingface.co/EleutherAI/pythia-6.9b	GPT2	6.9B
Solar	solar_10.7b	https://huggingface.co/upstage/SOLAR-10.7B-v1.0	Llama	10.7B
Solar	solar_10.7b_instruct	https://huggingface.co/upstage/SOLAR-10.7B-Instruct-v1.0	Llama	10.7B
Solar	solar_70b	https://huggingface.co/upstage/SOLAR-0-70b-16bit	Llama	70B
Solar	solar_orcadpo_solar_instruct_slerp	https://huggingface.co/kodonho/Solar-OrcaDPO-Solar-Instruct-SLERP	Llama	10.7B
Starling	starling	https://huggingface.co/berkeley-nest/Starling-LM-7B-alpha	Llama	7B
Llama 1	upstage_llama_1_30b	https://huggingface.co/upstage/llama-30b-instruct	Llama	30B
Llama 1	upstage_llama_1_65b	https://huggingface.co/upstage/llama-65b-instruct	Llama	65B
Llama 2	upstage_llama_2_70b	https://huggingface.co/upstage/Llama-2-70b-instruct	Llama	70B
Vicuna 1	vicuna_1_13b	https://huggingface.co/lmsys/vicuna-13b-v1.3	Llama	13B
Vicuna 1	vicuna_1_33b	https://huggingface.co/lmsys/vicuna-33b-v1.3	Llama	33B
Vicuna 1	vicuna_1_7b	https://huggingface.co/lmsys/vicuna-7b-v1.3	Llama	7B
Vicuna 2	vicuna_2_13b	https://huggingface.co/lmsys/vicuna-13b-v1.5-16k	Llama	13B
Vicuna 2	vicuna_2_7b	https://huggingface.co/lmsys/vicuna-7b-v1.5-16k	Llama	7B
Yi /w RLHF	yi_34b_chat	https://huggingface.co/01-ai/Yi-34B-Chat	Llama	34B
Yi	yi_6b	https://huggingface.co/01-ai/Yi-6B	Llama	6B
Yi /w RLHF	yi_6b_chat	https://huggingface.co/01-ai/Yi-6B-Chat	Llama	6B
Zephyr	zephyr_alpha	https://huggingface.co/HuggingFaceH4/zephyr-7b-alpha	Llama	7B
Zephyr	zephyr_beta	https://huggingface.co/HuggingFaceH4/zephyr-7b-beta	Llama	7B

Table 4: LLMs used in LLaSA with their corresponding model names, Huggingface model names, and model information.

Prompting Method	**Input Prompt**
MCP	Instruction: You are an intelligent agent specialized for database subject problem solving. The question below is about relational databases as taught at the Australian National University. The exam is intended for undergraduate and postgraduate students with a variety of majors, including computer science, engineering, arts, and business. Given the diversity of students' majors and learning experiences, the difficulty level of the exam will vary depending on the students' background and understanding of relational databases. The content is likely to be relatively familiar to computer science and engineering majors, but may be more challenging for arts or business majors. Therefore, the difficulty of the exam will vary depending on the student's major and relevant experience. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Answer:
CoT	You are an intelligent agent specialized for database subject problem solving. The question below is about relational databases as taught at the Australian National University. The exam is intended for undergraduate and postgraduate students with a variety of majors, including computer science, engineering, arts, and business. Given the diversity of students' majors and learning experiences, the difficulty level of the exam will vary depending on the students' background and understanding of relational databases. The content is likely to be relatively familiar to computer science and engineering majors, but may be more challenging for arts or business majors. Therefore, the difficulty of the exam will vary depending on the student's major and relevant experience. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Let's think step by step.
PS	You are an intelligent agent specialized for database subject solving. The question below is about relational databases as taught at the Australian National University. The exam is intended for undergraduate and postgraduate students with a variety of majors, including computer science, engineering, arts, and business. Given the diversity of students' majors and learning experiences, the difficulty level of the exam will vary depending on the students' background and understanding of relational databases. The content is likely to be relatively familiar to computer science and engineering majors, but may be more challenging for arts or business majors. Therefore, the difficulty of the exam will vary depending on the student's major and relevant experience. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan to solve the problem step by step.

Table 5: Prompts used for question-solving in DBE-KT22

prompting methods	**Input Prompt**
MCP	You are an intelligent agent specialized for various subject problem solving. The question below is a rich educational dataset derived from the ASSISTMents online tutoring system, which is used to help students with math and other subjects. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Answer:
CoT	You are an intelligent agent specialized for various subject problem solving. The question below is a rich educational dataset derived from the ASSISTMents online tutoring system, which is used to help students with math and other subjects. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Let's think step by step.
PS	You are an intelligent agent specialized for various subject problem solving. The question below is a rich educational dataset derived from the ASSISTMents online tutoring system, which is used to help students with math and other subjects. You'll need to step into the role of these students. Read the questions and options below, understand the question and select one answer from the choices. Use any hints provided to assist in solving the problems. **{Question}** A. **{Choice 1}** B. **{Choice 2}** **...** Hint: **{Hint}** Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan to solve the problem step by step.

Table 6: Prompts used for question-solving in ASSISTMents