POLITECNICO DI TORINO
Repository ISTITUZIONALE

Towards Step-Aware ITSs: Generation and Evaluation of Synthetic Step-by-Step Exercise Solutions

Original
Towards Step-Aware ITSs: Generation and Evaluation of Synthetic Step-by-Step Exercise Solutions / Russo, F., Calo, T., De Russis, L.. - (2025), pp. 281-285. (L@S '25: Twelfth ACM Conference on Learning @ Scale Palermo (ITA) 21-23 July 2025) [10.1145/3698205.3733940].

Availability:
This version is available at: 11583/3000468 since: 2025-08-04T16:09:36Z

Publisher:
ACM

Published
DOI:10.1145/3698205.3733940

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
ACM postprint/Author's Accepted Manuscript

(Article begins on next page)

26 November 2026

Towards Step-Aware ITSs: Generation and Evaluation of Synthetic Step-by-Step Exercise Solutions

Francesca Russo
francesca.russo@polito.it
Politecnico di Torino
Torino, Italy

Tommaso Calò
tommaso.calo@polito.it
Politecnico di Torino
Torino, Italy

Luigi De Russis
luigi.derussis@polito.it
Politecnico di Torino
Torino, Italy

ABSTRACT

Intelligent Tutoring Systems (ITSs) have shown great potential in enhancing how education is delivered. Many existing ITSs leverage Reinforcement Learning (RL) to optimize the sequence of exercises proposed to the learner. These systems adapt content based on the student's performance on previous exercises, addressing knowledge gaps while advancing through mastered concepts. However, they typically operate at the whole-exercise level, without visibility into the intermediate steps. In reality, learners may fail to solve an exercise because they encounter difficulties with specific substeps. Existing ITSs rely on datasets that do not include exercise decomposition in steps.

To overcome this limitation, in this paper, we employ GPT-o3-mini to generate synthetic step-by-step solutions for mathematics exercises from the Junyi Academy dataset. To evaluate if these synthetic steps are useful in reaching the final solution, we use three models of varying size from the Llama family to simulate students of different knowledge levels (i.e., low, medium, high) and verify if the step-by-step guidance increases their problem-solving capabilities.

By comparing direct answers for exercises to answers that leverage an incremental step guidance strategy, models successfully solve up to 42% more exercises. This evaluation serves as a foundation for creating synthetic step-by-step solutions that can be employed to develop next-generation step-aware ITSs tailored to students' specific knowledge gaps.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; • **Applied computing** → **E-learning**; *Interactive learning environments.*

KEYWORDS

Intelligent Tutoring Systems, Large Language Models, Synthetic Data Generation

ACM Reference Format:
Francesca Russo, Tommaso Calò, and Luigi De Russis. 2025. Towards Step-Aware ITSs: Generation and Evaluation of Synthetic Step-by-Step Exercise Solutions. In *Proceedings of the Twelfth ACM Conference on Learning @ Scale (L@S '25), July 21–23, 2025, Palermo, Italy*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3698205.3733940

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
L@S '25, July 21–23, 2025, Palermo, Italy
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1291-3/2025/07
https://doi.org/10.1145/3698205.3733940

1 INTRODUCTION

Intelligent Tutoring Systems (ITSs) are Artificial Intelligence (AI)-based systems that act as human tutors, providing personalized feedback and dynamically adapting the sequence of exercises based on the learner progression, enhancing learning outcomes [9].

Previous studies have explored the development of ITSs that leverage Reinforcement Learning (RL) algorithms to adapt the sequence of exercises [5, 7, 13] based on the learner's performance on previous exercises.

However, these systems are limited because they rely on datasets, such as Junyi Academy [12], which provide the text of exercises and whether the student has been able to solve them, without offering step-by-step solutions. Consequently, it is not possible to create algorithms that adapt to the learner's specific difficulties within individual exercises.

In particular, for mathematics exercises, since learners may not be able to directly solve an exercise because they may struggle with its intermediate steps [17], it is crucial to reinforce the learner knowledge relative to challenging steps, adapting the sequence of exercises proposed by an ITS to repair those knowledge building blocks.

Building a step-aware adaptive ITS requires a step-by-step decomposition of a set of exercises and their associated knowledge graph (i.e., a graph representing the dependencies among topic exercises). This data would allow the training a RL algorithm, simulating students that interact with the system, and embedding the RL model into an ITS [13].

In this work, we leverage GPT-o3-mini[1], a model optimized for STEM reasoning [11], to generate step-by-step solutions for the exercises from the Junyi Academy dataset. For each exercise, we prompt GPT-o3-mini to decompose the problem-solving process into a sequence of intermediate steps. Additionally, for steps identified as particularly challenging, we instruct the model to further break them down into more granular sub-steps, creating a two-tier guidance structure that can adapt to different levels of difficulty within the same problem.

To investigate whether the steps can effectively provide guidance to the final solution, we simulate students using three distinct models of varying size from the Llama family: Llama2-7b-chat [16], Llama3-8b-instruct, and Llama3-70b-instruct [2], representing three different knowledge levels (i.e., low, medium, high). We assume that larger models (in terms of parameter count) are capable of more advanced reasoning and problem-solving capabilities and can

[1]version: 2025-01-31

represent students with higher levels of domain knowledge, and vice-versa [4].

We evaluate each model's performance with and without step guidance. Interestingly, while larger models solved more exercises overall, smaller models showed greater relative improvement from step-by-step guidance. Findings show that models exposed to incremental step guidance with additional sub-step decomposition successfully solve up to 42% more exercises than when asked to generate direct answers. These results provide empirical demonstration of the effectiveness of language models in generating exercise-specific decomposition steps that can serve as the foundation for developing next-generation RL-based adaptive ITSs capable of fine-grained personalization.

2 BACKGROUND

Researchers have been investigating how to adapt the sequence of learning items presented to the learner based on the knowledge state and goals.

Liu et al. [7] developed CSEAL, a framework that combines the actual knowledge level of the learner and the concept dependency structures to personalize learning pathways. Similarly, Zhang et al. [19] created a framework aware of the difficulty of items, reordering their sequence in an incremental manner based on item difficulty to achieve the learner's goals. Li et al. [5] have enhanced the learning path recommendation by employing a Hierarchical RL algorithm that plans the path to follow to master a learning element. The learning elements sequence is continuously re-adapted based on the difference between the acquired knowledge of the learner and the goal to achieve.

However, if the learner is not able to solve an exercise, these ITSs may understand that the learner lacks the requisite knowledge for the entire exercise and inaccurately adapt the sequence of learning items. The educational datasets these works rely on [12, 14] lack of step-by-step resolutions that can be leveraged to create a step-aware ITS. By decomposing the exercises into steps and letting the learners solve each step, it is possible to identify the exact point where they struggle and dynamically adapt the learning path, targeting the strengthening of the knowledge behind it.

Recently, LLMs have been leveraged to provide Socratic guidance to students, demonstrating improved performance by decomposing problems into steps [6, 15]. However, these abilities have primarily been utilized for tutoring evaluation, not for the generation of synthetic data or enhancement of existing datasets.

In this work, we leverage the LLMs training knowledge in mathematics [1, 11] to generate step-by-step solutions for mathematics exercises of the Junyi Academy dataset [12], and we evaluate whether these generated decompositions are helpful in solving the whole exercise. Building on recent studies that have demonstrated the potential of LLMs to simulate diverse student profiles and learning behaviors [8, 18], we use varying-sized models to assess the effectiveness of our synthetic step decompositions across different knowledge levels.

We hope our contribution could enhance existing educational datasets with fine-grained solution paths and lay the groundwork for next-generation step-aware RL algorithms that can adapt to students' specific difficulties within individual exercises rather than treating exercises as the units.

3 METHODOLOGY

The methodology is divided in three steps detailed below. Figure 1 illustrates our approach to generate step-by-step solutions and investigates their efficacy in guiding in the resolution of mathematical exercises using the Llama models.

3.1 Dataset Preparation

For our work, we used mathematics exercises from the Junyi Academy dataset[2], a K-12 mathematics exercise dataset and their relative knowledge graph. Exercises were originally stored in an HTML format with embedded solutions; therefore, we developed a custom JavaScript script to extract the text and the solution.

To ensure dataset consistency, we implemented two preprocessing steps. First, we excluded exercises with screen elements that require the user interaction (e.g., drag and drop an element), as these would have been complex to manage within our LLM-based evaluation. Second, we removed exercises from topics with fewer than 10 sample to evaluate the models in mathematical areas with sufficient representation in the dataset.

3.2 Step-by-Step Solution Generation Process

We prompted an instance of GPT-o3-mini to provide the final solution of the exercises from the Junyi dataset while decomposing the problem-solving process into steps, and, if a step was considered difficult, to further break it down into sub-steps. Given the original Chinese language of the Junyi exercises, we simultaneously requested their English translation. Then, we employed a separate GPT-o3-mini instance to validate whether the generated final answers matched our extracted ground truth.

3.3 Experimental Design

We conducted three distinct experiments using three different LLMs of varying size, Llama2-7b-chat, Llama3-8b-instruct, and Llama3-70b-instruct, to simulate students of three capability levels: low, medium, and high, respectively.

The simulation is conducted under three different conditions: (i) the models are asked to directly solve an exercise, (ii) if they fail, they are incrementally provided with the steps generated by GTP-o3-mini, and each time asked to solve both the new step and the whole exercise, (iii) if the models fail to solve a step and the failed step contains sub-steps, the model incrementally receives these additional sub-steps, and is asked for the exercise, step, and relative sub-step solutions.

By comparing the performance of these three approaches, we aim to determine whether stepwise guidance leads to improved outcomes.

Experiment 1: Zero-shot Resolution

We initially prompted each Llama model to provide a direct answer to the exercise without any guidance.

Experiment 2: Step-by-Step Guidance

We define N as the number of first-level steps previously generated

[2]https://github.com/junyiacademy/junyiexercise, last-accessed 2025-04-05

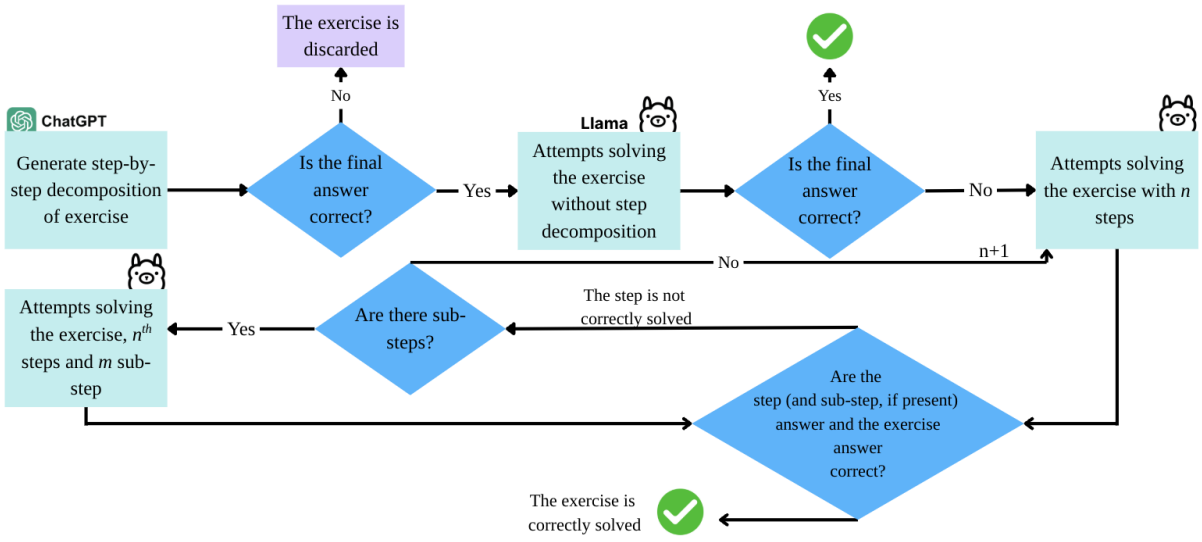

Figure 1: Overview of our approach. We generate step-by-step solutions and verify the correctness of the final answer. Then, we ask a Llama model to solve the exercise, verify the correctness of the answer, eventually iterating over steps and sub-steps if the exercise is not solved.

by GPT-o3-mini for a given exercise and available for step-by-step guidance. For exercises where the initial attempt failed, we implemented an iterative process. At each iteration n, with $n \in \{1, \ldots, N\}$, we provided the models with the previous $(n-1)$ steps and their solutions, and prompted the models to solve both the nth step and the complete exercise. This iterative process continued until either the model successfully solved the exercise or the N available steps were exhausted.

Experiment 3: Second-level Guidance

Building upon the same strategy used for the second experiment, we implemented a more granular guidance approach for cases where models struggled with specific steps. When a model failed to correctly solve a step for which sub-steps were available, we iteratively provided the model with these sub-steps. The models were then prompted to solve the original exercise, the challenging step, and the newly introduced sub-step, creating a two-tiered guidance structure for particularly challenging aspects of the problem-solving process.

3.4 Evaluation Methodology

For the three experiments, we employed GPT-o3-mini for the assessment of solution correctness across all granularity levels: complete exercise, intermediate step, and sub-step solutions.

We give GPT-o3-mini the answer generated by the Llama model and the corresponding exercise solution, asking to compare them and return a score, 1 if they are equal, 0 otherwise. The solutions comparison leverages GPT-o3-mini, rather than an equality direct comparison, as the solutions may have multiple valid representations and the LLM can be more robust against them.

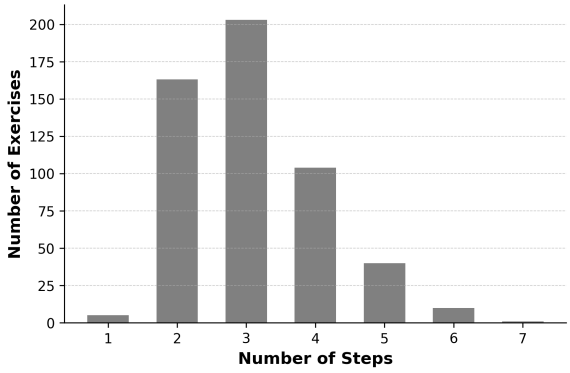

Figure 2: Distribution of exercises based on the number of steps generated by GPT-o3-mini.

4 RESULTS

After implementing the filtering process described in Section 3.1, the initial 968 exercises from the Junyi Academy dataset were reduced to 622 exercises. Subsequently, we generated step-by-step solutions for each exercise using GPT-o3-mini, which led to 526 correct solutions.

Figure 2 illustrates the distribution of the 526 exercises based on the number of steps generated by GPT-o3-mini. The majority of problems is decomposed into 2-4 steps, with fewer exercises requiring either a single step or more than 4 steps.

Figure 3 displays the percentage of correctly-solved exercises as a function of the number of first-level guidance steps provided to the Llama LLMs. All models exhibit substantial improvements when empowered with step guidance: Llama-3-70b-instruct solved

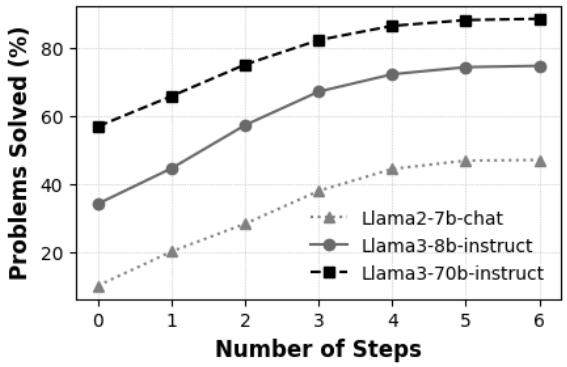

Figure 3: Percentage of correctly solved exercises in relation to the number of first-level guidance steps provided to three different Large Language Models.

Model	Without Guidance	With Steps	With Sub-steps	Δ
Llama2-7b-chat	53	248	266	213
Llama3-8b-instruct	180	394	402	222
Llama3-70b-instruct	300	467	477	177

Table 1: From left to right, number of correctly-solved exercises by the three Llama models when: asked to directly solve the exercise, provided with first-level steps guidance, and provided with additional sub-steps guidance. The rightmost column reports the difference between the number of exercises solved with sub-steps and the one without guidance.

32% more exercises, while Llama-3-8b-instruct and Llama-2-7b-chat solved 41% and 37% more exercises, respectively.

Additionally, Table 1 reports the number of correctly solved exercises by the three models under three conditions: directly solve the problem (i.e., without guidance), steps guidance, additional sub-steps guidance. In the end, the models correctly solve 177, 222, and 213, more exercises, compared to the scenario without any guidance, leading to a maximum improvement of 42% obtained using Llama3-8b-instruct.

The three model variants in our study effectively simulate students with different knowledge, as shown by their initial performance differences when solving exercises without any guidance. The largest model, Llama3-70b-instruct, reflecting a student with high knowledge, achieved the highest baseline accuracy, while the smallest model, Llama2-7b-chat, performed most poorly. All models exhibited significant and consistent performance gains when provided with step-level guidance, demonstrating the steps effectiveness to conduct the learner to the solution. The models' performance plateau beyond 4 steps due to the limited number of exercises requiring 5 or more solution steps in our dataset (see Figure 2). The gain was further amplified with the introduction of step decomposition into sub-steps, although the magnitude of improvement varied by model capacity. The largest model successfully solved more exercises in absolute terms, when provided with sub-steps, reflecting its enhanced capability to leverage the provided guidance.

Lastly, our analysis reveals an inverse relationship between model size and performance gain from guidance. While Llama-3-70b-instruct successfully solved more exercises, smaller models gained more substantial improvements (see Table 1). This finding suggests that guidance may be particularly valuable for addressing knowledge gaps in less-prepared learners.

5 CONCLUSIONS AND FUTURE WORK

In this work, we have generated step-by-step solutions using GPT-o3-mini for 526 mathematics exercises from the Junyi Academy dataset. Using three models of different sizes from the Llama family to simulate students of three different knowledge levels, we have demonstrated that the previously generated step-by-step solutions can be a valid support to provide incremental guidance, allowing Llama models to solve 42% more exercises in the best case scenario, compared to the zero-shot exercise resolution.

However, while our work has shown that guided resolutions help LLMs to solve mathematical problems, it comes with some limitations. The first one is that our approach relies on step-by-step solutions generated by another LLM (i.e., GPT-o3-mini), which can hallucinate and produce intermediate steps that may be incorrect [3], not useful for the solution of the overall problem [17] or revel the solution too early [10]. We verify the correctness of the solution generated by GPT-o3-mini by only comparing its final answer for the exercise to the ground-truth final solution, but we do not perform a verification on the generated steps. Future works can address these by manually inspecting a sample of the generated steps, to ensure they do not include the solution of the exercise.

Secondly, we only apply our methodology on the Junyi Academy dataset. Despite including dependencies between exercises, which can be valuable for future the development of a RL model, this dataset targets K-12 students only. Future works could extend experiments by including datasets like MathOdyssey [1] to assess the generalization of our approach across exercises difficulty levels.

Additionally, in the Solution Generation Process (see Section 3.2), we have generated a fixed exercise decomposition that is not dynamically adapted to the model's actual understanding at inference time. Our approach can be further refined to perform dynamic step generation, so that the step provided to the model is context-aware and adapted to the actual model comprehension.

Finally, the next step would be to use the generated step-by-step solutions to train and evaluate step-aware RL algorithms that could be embedded into ITSs to better support learners.

ACKNOWLEDGMENTS

This work was supported by the Cineca consortium.

REFERENCES

[1] Meng Fang, Xiangpeng Wan, Fei Lu, Fei Xing, and Kai Zou. 2024. MathOdyssey: Benchmarking Mathematical Problem-Solving Skills in Large Language Models Using Odyssey Math Data. arXiv:2406.18321 [cs.CL] https://arxiv.org/abs/2406.18321

[2] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[3] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A Survey on Hallucination in Large Language Models: Principles,

Taxonomy, Challenges, and Open Questions. *ACM Trans. Inf. Syst.* 43, 2, Article 42 (Jan. 2025), 55 pages. https://doi.org/10.1145/3703155

[4] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361 [cs.LG] https://arxiv.org/abs/2001.08361

[5] Qingyao Li, Wei Xia, Li'ang Yin, Jian Shen, Renting Rui, Weinan Zhang, Xianyu Chen, Ruiming Tang, and Yong Yu. 2023. Graph Enhanced Hierarchical Reinforcement Learning for Goal-oriented Learning Path Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (Birmingham, United Kingdom) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 1318–1327. https://doi.org/10.1145/3583780.3614897

[6] Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024. SocraticLM: exploring socratic personalized teaching with large language models. *Advances in Neural Information Processing Systems* 37 (2024), 85693–85721.

[7] Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting Cognitive Structure for Adaptive Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 627–635. https://doi.org/10.1145/3292500.3330922

[8] Zhengyuan Liu, Stella Xin Yin, Geyu Lin, and Nancy F. Chen. 2024. Personality-aware Student Simulation for Conversational Intelligent Tutoring Systems. *arXiv preprint arXiv:2404.06762* (2024).

[9] Wenting Ma, Olusola O. Adesope, John C. Nesbit, and Qing Liu. 2014. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology* (2014). https://doi.org/10.1037/a0037123

[10] Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. In *Findings of the Association for Computational Linguistics: EMNLP*. https://aclanthology.org/2023.findings-emnlp.372.pdf

[11] OpenAI. 2025. *OpenAI o3-mini - Pushing the frontier of cost-effective reasoning.* https://openai.com/index/openai-o3-mini/

[12] Chen Pojen, Hsieh Mingen, and Tsai Tzuyang. 2020. Junyi Academy Online Learning Activity Dataset: A large-scale public online learning activity dataset from elementary to senior high school students. *Dataset available from https://www.kaggle.com/junyiacademy/learning-activity-public-dataset-by-junyi-academy* (2020).

[13] Wafaa S. Sayed, Ahmed M. Noeman, Abdelrahman Abdellatif, Moemen Abdelrazek, Mostafa G. Badawy, Ahmed Hamed, and Samah El-Tantawy. 2023. AI-based adaptive personalized content presentation and exercises navigation for an effective and engaging E-learning platform. *Multimedia Tools and Applications* (2023). https://doi.org/10.1007/s11042-022-13076-8

[14] Douglas Selent, Thanaporn Patikorn, and Neil Heffernan. 2016. ASSISTments Dataset from Multiple Randomized Controlled Experiments. In *Proceedings of the Third (2016) ACM Conference on Learning @ Scale* (Edinburgh, Scotland, UK) *(L@S '16)*. Association for Computing Machinery, New York, NY, USA, 181–184. https://doi.org/10.1145/2876034.2893409

[15] Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, Tanmay Sinha, Manu Kapur, and Mrinmaya Sachan. 2022. Automatic Generation of Socratic Subquestions for Teaching Math Word Problems. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4136–4149. https://aclanthology.org/2022.emnlp-main.277

[16] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288 [cs.CL] https://arxiv.org/abs/2307.09288

[17] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating Mathematical Reasoning Beyond Accuracy. arXiv:2404.05692 [cs.CL] https://arxiv.org/abs/2404.05692

[18] Yicheng Xu and Jie Zhang. 2024. EduAgent: Simulating Student Behaviors with Cognitive Priors in Large Language Models. *arXiv preprint arXiv:2402.11678* (2024).

[19] Haotian Zhang, Shuanghong Shen, Bihan Xu, Zhenya Huang, Jinze Wu, Jing Sha, and Shijin Wang. 2024. Item-Difficulty-Aware Learning Path Recommendation: From a Real Walking Perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Barcelona, Spain) *(KDD '24)*. Association for Computing Machinery, New York, NY, USA, 4167–4178. https://doi.org/10.1145/3637528.3671947