

Personalized Adaptive Math Tutoring with Large Language Models

Hai Huang, Ziheng Xin

1. Introduction

Mathematics is foundational in early education, yet delivering effective instruction to a diverse student population remains a significant challenge. In middle school classrooms, students differ widely in prior knowledge, learning pace, and confidence. Traditional educational approaches typically employ uniform teaching methods, offering the same problem sets and standardized feedback to all students, regardless of individual needs. This one-size-fits-all model can lead to disengagement among advanced learners and frustration among those who struggle to keep up.

To address this gap, we introduce an intelligent math tutoring platform powered by a large language model (LLM), with GPT-4o at its core. The system dynamically adjusts problem difficulty based on student performance, evaluates open-ended responses, and provides personalized feedback. By modulating task complexity in real time and offering on-demand hints or full explanations, the platform replicates many of a human tutor's adaptive and supportive qualities.

2. Literature Review and Motivation

Adaptive learning has long been a central theme in educational technology research. Early systems like the Cognitive Tutor showed measurable gains in student outcomes by personalizing problem-solving experiences. These rule-based frameworks linked common student errors to targeted feedback, offering structured support. However, they often lacked the flexibility and linguistic nuance to function effectively in open-ended domains such as natural language or complex mathematical reasoning.

Advances in natural language processing (NLP) and deep learning have enabled a new generation of educational tools. Large language models (LLMs), such as OpenAI's GPT series, have demonstrated strong capabilities in understanding context, generating human-like text, and solving multi-step arithmetic problems. Recent research underscores their potential to interpret diverse student inputs, generate tailored hints, and assess open-ended responses with minimal human oversight.

However, the use of LLMs in real-time adaptive tutoring remains nascent. This project advances that frontier by using GPT-4o not only as a solver but also as a feedback engine. By drawing on its generative and inferential capabilities, our platform delivers personalized instruction that dynamically adjusts to each learner's progress, emulating the adaptability and responsiveness of a human tutor.

3. Methodology

3.1 System Architecture

The adaptive math tutoring platform has four core components: a question bank, an adaptive decision engine, an answer evaluation module, and an interactive user interface. The question bank includes problems organized into three tiers of difficulty—easy, medium, and hard—sourced from the GSM8K dataset and manually labeled based on complexity, reasoning depth, and number of steps required.

The decision engine determines the next question's difficulty by analyzing the correctness of the student's most recent response. A correct answer increases the challenge level, while an incorrect one maintains or reduces it, ensuring learners operate within their optimal cognitive zone.

To evaluate responses, GPT-4o is instructed to append the final numeric answer after the token "####". This formatting enables automatic comparison with the correct answer. The interactive front end, built with HTML and IPyWidgets, allows students to submit answers, request assistance, and receive feedback through a streamlined, browser-based interface.

3.2 Dataset

All problems are drawn from GSM8K, a standard benchmark for grade-school mathematical reasoning. Before deployment, questions were reformatted for consistency, ambiguities were resolved, and answer strings were aligned with a structured evaluation format. Each problem was classified into one of three difficulty levels using heuristics like sentence complexity, numerical range, and step count.

3.3 Answer Evaluation Pipeline

To ensure reliable answer validation, the system follows a two-step evaluation process. GPT-4o is first prompted to provide answers using the "####" format. If the format is not followed, a fallback mechanism uses regular expressions to extract the final numeric value from the response.

Currently, the system evaluates answers through exact string matching, which can misclassify mathematically equivalent values (e.g., "325.00" vs. "325"). Future improvements will include numerical parsing with tolerance thresholds to reduce false negatives.

Despite this limitation, the fallback significantly improved performance. Without it, strict formatting yielded a 71% accuracy rate. With the fallback enabled, accuracy rose

to 96%, indicating that most evaluation errors were due to format deviations rather than incorrect reasoning.

3.4 Adaptive Feedback Logic

The platform enhances personalization by offering interactive support tailored to student needs. Performance-based progression keeps learners appropriately challenged, while two optional features provide on-demand guidance: the "Need a Hint?" button generates a context-specific scaffolded hint, and the "Show Full Steps" button displays a complete solution walkthrough.

These tools are designed to emulate the support of a human tutor while preserving learner autonomy. The system fosters engagement and encourages active problem-solving by allowing students to choose when and how to receive assistance.

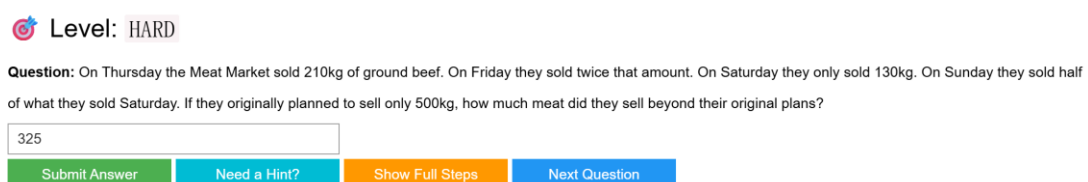
4. Results and Analysis

4.1 Interface Behavior

The tutoring platform enables real-time interaction through a streamlined interface where students can solve problems, receive instant feedback, and access on-demand support. Users input responses directly and receive immediate validation of correctness along with performance tracking. When needed, students can request a hint or a complete solution breakdown by clicking "Need a Hint?" or "Show Full Steps," respectively. These features replicate the responsive guidance of a human tutor while allowing students to control the level of assistance they receive, supporting both autonomy and reflective learning.

4.2 Sample Interaction: Hard-Level Question

Consider the hard-level problem presented in the figure below to illustrate the platform's dynamic support features. The student is asked to calculate the total ground beef sold over four days and determine the excess relative to an initial sales plan of 500 kg.



The screenshot shows a user interface for a math problem. At the top, there is a red icon followed by the text "Level: HARD". Below this, the question text reads: "Question: On Thursday the Meat Market sold 210kg of ground beef. On Friday they sold twice that amount. On Saturday they only sold 130kg. On Sunday they sold half of what they sold Saturday. If they originally planned to sell only 500kg, how much meat did they sell beyond their original plans?". Below the question is a text input field containing the number "325". At the bottom, there are four buttons: "Submit Answer" (green), "Need a Hint?" (teal), "Show Full Steps" (orange), and "Next Question" (blue).

When the student submits their answer, the platform evaluates it using GPT-4o and displays immediate feedback. In this case, the answer "325" is assessed as correct, and the platform presents an expected solution string showing the step-by-step arithmetic reasoning. The feedback includes a breakdown of daily sales (e.g., Thursday: 210 kg,

Friday: $2 \times 210 = 420$ kg, etc.), a cumulative total of 825 kg, and the final subtraction to determine the excess sold ($825 - 500 = 325$ kg).

✔ Correct!

Expected Answer: On Friday they sold $210 \times 2 = 420$ kg. On Sunday they sold $130 \div 2 = 65$ kg. They sold a total of $210 + 420 + 130 + 65 = 825$ kg of meat. Meat Market sold an extra $825 - 500 = 325$ kg of meat beyond their original 500kg plans.

325

Streak: 1 | Difficulty: HARD

If the student is uncertain and clicks "Need a Hint?", the platform generates a context-aware prompt encouraging them to compute individual day totals before summing and comparing.

💡 **Hint:** First, calculate the amount of meat sold on Friday, then on Sunday. Add up the total amount of meat sold over the four days and compare it to the original plan of 500kg to find out how much meat they sold beyond their original plans.

If they request "Show Full Steps," a complete explanation is revealed, including annotated arithmetic operations and logic in a clear, educational format.

📘 **Full Explanation:** To solve this problem, we need to calculate the total amount of ground beef sold over the four days (Thursday, Friday, Saturday, and Sunday) and then compare it to the original plan of selling 500 kg. Here's a step-by-step breakdown:

1. **Calculate the amount sold on each day:**

- **Thursday:** The Meat Market sold 210 kg of ground beef.
- **Friday:** They sold twice the amount sold on Thursday. Therefore, the amount sold on Friday is: $[2 \times 210 , \text{kg} = 420 , \text{kg}]$
- **Saturday:** They sold 130 kg of ground beef.
- **Sunday:** They sold half of what they sold on Saturday. Therefore, the amount sold on Sunday is: $[\frac{130}{2} , \text{kg} = 65 , \text{kg}]$

2. **Calculate the total amount sold over the four days:**

Add up the amounts sold on each day: $[210 , \text{kg (Thursday)}] + 420 , \text{kg (Friday)} + 130 , \text{kg (Saturday)} + 65 , \text{kg (Sunday)}]$

Perform the addition step-by-step: $[210 + 420 = 630] [630 + 130 = 760] [760 + 65 = 825]$

So, the total amount of ground beef sold over the four days is 825 kg.

3. **Determine how much meat was sold beyond the original plans:**

The original plan was to sell 500 kg. To find out how much was sold beyond this plan, subtract the planned amount from the actual amount sold: $[825 , \text{kg (actual)}] - 500 , \text{kg (planned)} = 325 , \text{kg}]$

Therefore, the Meat Market sold 325 kg of ground beef beyond their original plans.

This interaction, visualized in the figures, demonstrates the layered support system—ranging from subtle hints to detailed walkthroughs—designed to scaffold student understanding.

4.3 Quantitative Metrics

We tested the platform on 100 randomly selected math problems from the GSM8K dataset to evaluate system performance. The initial implementation, which relied exclusively on strict output formatting (i.e., requiring the "#####" suffix), achieved a 71% answer-matching accuracy. After incorporating a fallback extraction method that retrieves the final numeric value from GPT-4o's response via regular expressions, the overall accuracy increased substantially to 96%. This improvement indicates that most mismatches were due to formatting inconsistencies rather than errors in reasoning.

	question	gpt_final	reference_answer	gpt_answer
46	A 750 ml bottle of spirits costs \$30.00 and ha...	98.00	98	To determine how much money a restaurant makes...
47	Travis goes through 2 boxes of cereal a week. ...	312.00	312	To find out how much Travis spends on cereal i...
62	John has to get a new blanket. He decides to ...	240	2240	To find the cost of the quilt, we first need t...
66	Southton buries their time capsule 15 feet und...	72	48	To find the depth at which Northton's time cap...
75	Tonya has \$150.00 on her credit card. If she ...	100.00	120	Tonya currently has a balance of \$150.00 on he...
76	Kurt's old refrigerator cost \$0.85 a day in el...	12.00	12	To find out how much money Kurt saves in a 30-...
97	Henry took 9 pills a day for 14 days. Of these...	574.00	41	To find out how much Henry spent in total on t...

✅ GPT Accuracy: 93.00%

❌ Number of incorrect answers: 7

The accompanying figure summarizes these evaluation outcomes, highlighting the answer accuracy distribution and examples of model outputs. Notably, the GPT model achieved an overall accuracy of 93% across the trials after fallback parsing, with only seven incorrect outputs. These results demonstrate the system's reliability in answer validation and instructional assistance.

5. Discussion

This project showcases integrating a transformer-based language model into an interactive tutoring system designed for middle school mathematics. By leveraging GPT-4o's capabilities, the platform adjusts problem difficulty in response to student performance, interprets free-text answers, and generates natural-language feedback. These features collectively support a personalized and responsive learning experience, unlike static problem sets that lack real-time adaptability.

While the system performed well, certain limitations remain. One challenge is the model's occasional failure to follow the required output format (e.g., the "#####" suffix), which led to misclassification of correct answers. Although a fallback strategy using pattern matching improved accuracy significantly, this dependency underscores the fragility of format-based parsing and the need for more robust numerical validation.

Another limitation concerns the platform's deployment environment. Built using IPyWidgets within a Jupyter Notebook, the current interface is not optimized for broader accessibility. Adapting the system to a web- or mobile-based application would enhance usability and scalability across diverse educational settings.

A further area for improvement is the absence of persistent user modeling. The platform does not track student progress across sessions or generate personalized learning pathways over time. Incorporating longitudinal data would enable profound insights into learning behaviors and support more effective, long-term adaptation.

Despite these constraints, the modular architecture offers substantial room for future growth. The framework can be extended to other subjects or deployed in more scalable formats. The platform also encourages reflective practice by supporting learner autonomy through optional hints and detailed solution walkthroughs. These qualities make it a promising tool for delivering scalable, adaptive instruction using large language models.

6. Conclusion and Future Work

This project introduced an adaptive math tutoring platform powered by GPT-4o, designed to deliver personalized instruction to middle school students. The system adjusts question difficulty in response to performance, evaluates open-ended responses, and offers scaffolded feedback through hints and solution walkthroughs. With a fallback strategy, the platform achieved 96% answer validation accuracy, demonstrating technical robustness and instructional effectiveness.

Several improvements are envisioned to broaden its impact. First, migrating from the current notebook interface to a complete web application—using frameworks like Streamlit or React—would enhance usability and cross-device accessibility. Second, persistent user profiles would enable longitudinal progress tracking and support more tailored learning paths. Expanding the platform's capabilities to include multimodal content, such as diagrams or animations, could further enrich conceptual understanding.

Additional extensions may include educator-facing dashboards for monitoring student performance, personalizing instruction, and expanding content coverage beyond mathematics to other academic subjects. These developments position the platform as a flexible and scalable solution within the broader field of AI-powered education.

Overall, this work highlights the potential of large language models to deliver adaptive, engaging, and scalable tutoring experiences. By combining real-time evaluation, individualized progression, and interactive feedback, the platform represents a strong foundation for the next generation of intelligent learning systems.

Reference

- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, 30–43.
- Doignon, J. P., & Falzague, J. C. (1999). *Knowledge Spaces*. Springer.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., ... Song, D.

(2021). Measuring Mathematical Problem Solving With the MATH Dataset. NeurIPS.

Ling, W., Yogatama, D., Dyer, C., & Blunsom, P. (2017). Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. ACL.

Saxton, D., Grefenstette, E., Hill, F., & Kohli, P. (2019). Analysing Mathematical Reasoning Abilities of Neural Models. ICLR

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language Models are Few-Shot Learners. NeurIPS.

Liang, C., Norouzi, M., Berant, J., Le, Q., & Lao, N. (2019). Neural Symbolic Machines: Learning Semantic Parsers on Freebase With Weak Supervision. ACL.