# UCO: A Multi-Turn Interactive Reinforcement Learning Method for Adaptive Teaching with Large Language Models


1st Shouang Wei
*East China Normal University*
*School of Computer Science and Technology*
Shanghai, China
52285901019@stu.ecnu.edu.cn

2nd Min Zhang
*East China Normal University*
*Shanghai Institute of AI for Education*
Shanghai, China
mzhang@cs.ecnu.edu.cn

3rd Xin Lin
*East China Normal University*
*Shanghai Institute of AI for Education*
Shanghai, China
xlin@cs.ecnu.edu.cn

4th Bo Jiang
*East China Normal University*
*Shanghai Institute of AI for Education*
Shanghai, China
bjiang@deit.ecnu.edu.cn

5th Kun Kuang
*Zhejiang University*
*College of Computer Science and Technology*
Hangzhou, China
kunkuang@zju.edu.cn

6th Zhongxiang Dai
*The Chinese University of Hong Kong, Shenzhen*
*School of Data Science*
Shenzhen, China
daizhongxiang@cuhk.edu.cn



***Abstract*—Large language models (LLMs) are shifting from answer providers to intelligent tutors in educational settings, yet current supervised fine-tuning methods only learn surface teaching patterns without dynamic adaptation capabilities. Recent reinforcement learning approaches address this limitation but face two critical challenges. First, they evaluate teaching effectiveness solely based on whether students produce correct outputs, unable to distinguish whether students genuinely understand or echo teacher-provided answers during interaction. Second, they cannot perceive students' evolving cognitive states in real time through interactive dialogue, thus failing to adapt teaching strategies to match students' cognitive levels dynamically. We propose the Unidirectional Cognitive Optimization (UCO) method to address these challenges. UCO uses a multi-turn interactive reinforcement learning paradigm where the innovation lies in two synergistic reward functions: the Progress Reward captures students' cognitive advancement, evaluating whether students truly transition from confusion to comprehension, while the Scaffold Reward dynamically identifies each student's Zone of Proximal Development (ZPD), encouraging teachers to maintain productive teaching within this zone. We evaluate UCO by comparing it against 11 baseline models on BigMath and MathTutorBench benchmarks. Experimental results demonstrate that our UCO model outperforms all models of equivalent scale and achieves performance comparable to advanced closed-source models. The code and data are available at https://github.com/Mind-Lab-ECNU/UCO.**



***Index Terms*—Large Language Models; Reinforcement Learning; Interactive Dialogue; Zone of Proximal Development; Unidirectional Cognitive Optimization.**


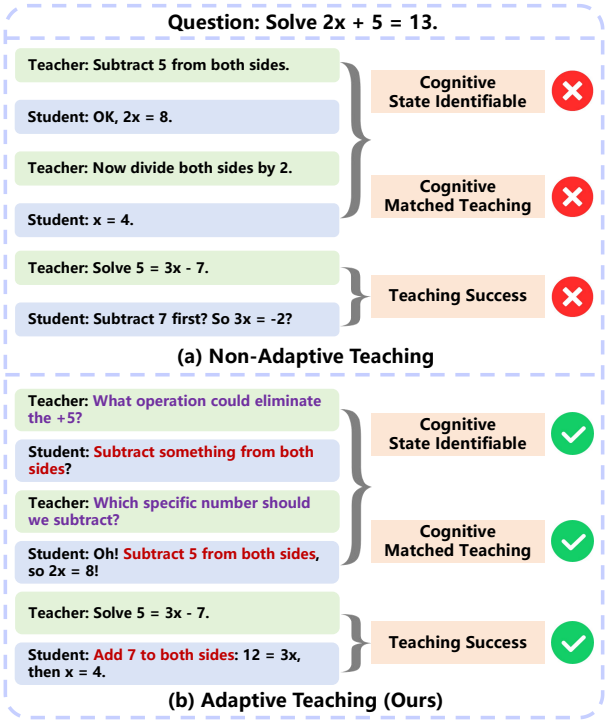


Fig. 1. Example of Two Teaching Methods: Comparison Between Non-Adaptive (a) and Adaptive (b) approaches.

## I. Introduction

Large language models are shifting from "answer providers" to "intelligent tutors" in educational applications. Current mainstream methods rely on supervised fine-tuning (SFT) with human-annotated or model-generated teaching dialogues [1]–[3]. These dialogues show how teachers guide students through questioning, prompting, and feedback. However, this data-driven imitation approach faces a key challenge. SFT-trained models only repeat fixed teaching patterns from training data. They cannot dynamically adjust teaching behaviors like real teachers do in actual practice.

Prior research explores reinforcement learning (RL) methods to break through this bottleneck [4]–[6]. The core advantage lies in explicitly defining optimization objectives through reward signals. Early work primarily adopts single-turn reinforcement learning paradigms [7]–[9]. After students give one answer, the teacher models provide hints or feedback. However, real teaching is inherently a multi-turn dynamic process. Single-turn paradigms cannot model continuous interactions between teachers and students. Therefore, recent research shifts to multi-turn RL methods [6], [10], [11], enabling multi-turn teacher-student interactions to optimize the model. Nevertheless, existing approaches primarily rely on the formal correctness of student outputs as a criterion for evaluation. They cannot accurately perceive the evolution of students' internal cognitive states in real time. This consequently leads to two critical deficiencies. First, when students output correct steps, models cannot distinguish whether students truly transition from confusion to comprehension. This prevents identifying which strategies truly promote cognitive improvement. Second, teacher models cannot select appropriate teaching strategies that match students' understanding levels. As long as students answer correctly, the system provides rewards. This causes models to favor the shortest path by directly providing answers for students to repeat. As illustrated in Figure 1(a), this non-adaptive teaching fails to identify student cognitive states. It also cannot provide cognitive-matched teaching strategies. This leads to teaching failure.

To effectively address these limitations, we propose the **Unidirectional Cognitive Optimization (UCO)** method. UCO specifically adopts a multi-turn interactive reinforcement learning paradigm. The teacher and student models interact continuously to dynamically generate online training rollouts. The method computes rollout returns based on reward mechanisms. We use Group Relative Policy Optimization (GRPO) [12] to update the teacher policy. As shown in Figure 1(b), our adaptive teaching approach can identify student cognitive states. It can also provide cognitive-matched teaching strategies. This ultimately leads to teaching success. The method's core innovation lies in specifically designing two reward functions:

- **Progress Reward.** From an information theory perspective, we formalize student cognitive progress as an entropy reduction process [13], [14]. This means students transition from high-entropy states with high uncertainty to low-entropy states with high certainty. However, directly calculating the entropy of student cognitive states is impractical. We argue that increases in student model probability for outputting correct answers essentially equal cognitive state transitions. These transitions manifest as shifts from high entropy to low entropy and from uncertainty to certainty. Based on this insight, we design dual-dimensional proxy indicators to quantify this entropy reduction process.
  ***1) Potential Capability Score:*** For each teacher output, we first use an oracle model to systematically generate multiple candidate correct student responses. We then accurately compute the student model's log-probability for generating each candidate response. We take the maximum value as the peak performance indicator. Higher values clearly indicate the student model shows high confidence in at least one correct response. This effectively corresponds to cognitive entropy reduction. Responses with insufficient confidence still directly reflect student confusion states. We apply the hyperbolic tangent function to map log-probabilities to the $[-1, 1]$ interval. This penalizes low-confidence outputs appropriately.
  ***2) Semantic Quality Score:*** We use text embedding models to compute maximum cosine similarity between student outputs and candidate correct answers. This measures the semantic alignment degree of their external understanding expressions. We combine these two dimensions with weighted aggregation to provide dense reward signals for each interaction turn.
- **Scaffold Reward.** Teaching practice shows that teaching difficulty cannot be too low, causing student boredom. It also cannot be too high, triggering frustration. Difficulty should remain within the student's "productive struggle" range. Vygotsky calls this range the Zone of Proximal Development (ZPD) [15]. To effectively translate this theory into a computable mechanism, we divide teacher behaviors by cognitive load into five ordered levels. These levels range from high to low: Metacognitive Hints, Strategic Hints, Conceptual Hints, Step-by-Step Hints, and Example Demonstration Hints. Based on this hierarchical system, we design a strategy to dynamically locate the ZPD. We first systematically evaluate the student model's success probability under each scaffold level. We identify the level with the highest success rate as their mastery ability. We then reduce scaffolding intensity by one level. This moves tasks into the ZPD region, requiring moderate productive struggle. Finally, we design a corresponding reward mechanism. When the teacher selects a scaffold level that falls exactly within the ZPD, the system provides positive rewards. When deviating from the ZPD, the system applies proportionally increasing penalties based on deviation magnitude.

By carefully combining these two reward functions with weighted aggregation, the method efficiently computes comprehensive rewards after each interaction turn. We subsequently update the teacher model's policy parameters accordingly to achieve cognition-oriented adaptive teaching optimization. Our key contributions are precisely the following:

- We propose a novel unidirectional cognitive optimization multi-turn reinforcement learning method. This method efficiently generates training trajectories online through iterative multi-turn teacher-student interactions, where the teacher model dynamically adjusts teaching strategies based on the continuous dynamic evolution of student cognitive states, achieving truly adaptive teaching.
- We design a cognition-oriented reward function comprising two core components: a progress reward that

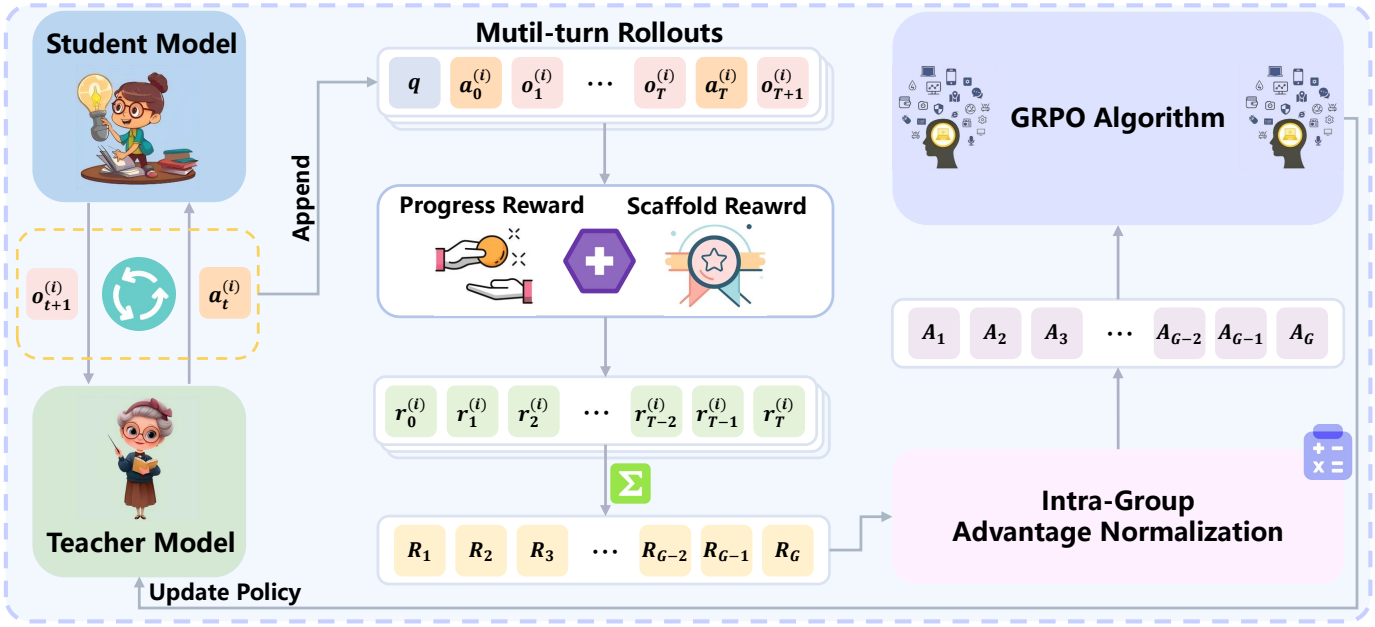


Fig. 2. Overview of the Unidirectional Cognitive Optimization (UCO) method, which optimizes the teacher model through multi-turn interactions with the student model, designing progress reward and scaffold reward to evaluate teacher outputs at each turn, and updating the teacher model via GRPO.

quantifies the degree of student cognitive improvement, and a scaffolding reward based on ZPD theory that dynamically matches teaching difficulty. These two rewards work synergistically to ensure the teacher model provides guidance aligned with students' cognitive levels.

- We systematically compare our method against 11 baseline models on BigMath and MathTutorBench benchmarks. Our UCO model consistently surpasses all models of comparable size and successfully achieves performance comparable to advanced closed-source models.

## II. Related Work

### A. Math Tutoring with Large Language Models

Effective teaching builds student cognition through scaffolding guidance rather than providing direct answers [16]–[18]. However, Large Language Models (LLMs) center their training paradigm on answer generation. This fundamentally conflicts with pedagogical principles [19]–[22]. Transforming LLMs into qualified teaching partners becomes an active research direction. Mathematics serves as an ideal testbed for exploring advanced tutoring strategies [23], [24]. Its logical rigor and quantifiable nature enable systematic evaluation. Early research focuses on constructing structured educational dialogue data. Initial work creates synthetic math teaching dialogues through multi-agent simulation [3], [25]. These datasets provide valuable structured examples of pedagogical interactions. They establish the foundation for models to learn teaching skills. Researchers then build benchmarks such as Big-Math [26] and MathTutorBench [27]. These benchmarks provide RL-specific training sets and evaluation systems. They significantly advance research standardization and reproducibility. To enable dynamic optimization of teaching strategies, researchers adopt RL as the mainstream framework. Different implementation paths emerge from this foundation. One approach introduces offline preference learning to improve the accuracy of teaching content [10], [28]. It optimizes the logic and quality of entire problem-solving trajectories. Yet this offline nature prevents real-time strategy adjustment based on student feedback. The method lacks dynamic interaction capabilities. Subsequent research builds online interactive environments with simulated students [6]. These methods directly reward specific teaching behaviors to address interactivity issues. However, they rely on simple rewards for "good teaching behaviors". They fail to measure the long-term goal of student understanding. We design an online reinforcement learning framework to overcome these limitations. Our teacher model optimizes by real-time perceiving student cognitive states through interaction with student models. It achieves truly adaptive teaching in authentic interactive contexts.

### B. Reward Mechanisms for Teaching Dialogues

Reinforcement learning formulates teaching dialogues as sequential decision-making processes [29], [30]. It optimizes pedagogical strategies through reward signals. Within this framework, the reward function architecture serves as the primary determinant of system performance. Early work employs outcome-oriented sparse rewards [31]–[33]. These rewards provide feedback solely at dialogue termination based on response accuracy. This evaluation mechanism faces severe credit assignment problems. Multi-turn dialogues make it difficult to identify critical teaching actions. This leads to inefficient policy learning. Subsequent research shifts to process-oriented dense rewards to address this problem [34]–[36]. These designs evaluate teaching effectiveness after each

dialogue turn. However, these mechanisms suffer from goal alienation issues. Reward functions focus only on student output correctness. They ignore whether students understand the content. This evaluation causes policy degradation. Teacher models receive high rewards whenever student models output correct steps. This mechanism encourages teacher models to take shortcuts. We propose a new reward function to address these limitations. Our approach analyzes student model cognitive state changes to evaluate cognitive progress. Further, we measure how teaching interventions match the student's proximal development zone. It achieves a paradigm shift from answer correctness to understanding depth.

## III. Method

### A. Problem Formalization

We formalize adaptive teaching dialogue generation as a Partially Observable Markov Decision Process (POMDP). We define $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{R}, \gamma)$. The state space $\mathcal{S}$ contains problem descriptions and dialogue history. The teacher action space $\mathcal{A}$ represents pedagogical strategies. The observation space $\mathcal{O}$ corresponds to student text responses. The state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ models environment dynamics. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ evaluates teaching effectiveness. The discount factor $\gamma \in [0, 1]$ balances immediate and future rewards.

As clearly illustrated in Figure 2, we propose the innovative Unidirectional Cognitive Optimization (UCO) method. Our primary goal is to train an optimal teacher policy that can adaptively guide students to effectively solve mathematical problems. Specifically, we aim to maximize the expected cumulative reward of teacher responses across dialogue interactions. The objective function is formally formulated as:

$$\pi^*_{\text{teacher}} = \arg\max_{\pi_{\text{teacher}}} \mathbb{E}_{q\sim\mathcal{D},\tau\sim\rho(\pi_{\text{teacher}},\pi^{\text{fixed}}_{\text{student}})}\left[\sum_{t=0}^{T}\gamma^t r_t^{(i)}\right], \quad (1)$$

where $\pi^{\text{fixed}}_{\text{student}}$ denotes the fixed-parameter student model, $\pi_{\text{teacher}}$ represents the teacher model to be optimized, $q$ denotes a problem sampled from the problem distribution $\mathcal{D}$, $\tau$ represents a dialogue rollout generated by the interaction between teacher and student policies, $\rho(\pi_{\text{teacher}}, \pi^{\text{fixed}}_{\text{student}})$ denotes the rollout distribution, $T$ is the maximum number of dialogue turns. The state at turn $t$ for the $i$-th rollout is $s_t^{(i)} = \{q, a_0^{(i)}, o_1^{(i)}, \ldots, a_{t-1}^{(i)}, o_t^{(i)}\}$, which consists of the problem $q$ and the dialogue history up to turn $t$. The reward $r_t^{(i)} = \mathcal{R}(s_t^{(i)}, a_t^{(i)})$ is computed by the reward function.

### B. Dual-Objective Reward Function

We carefully design a comprehensive dual-objective reward function to thoroughly evaluate the overall teaching effectiveness of the teacher model, which specifically includes progress reward and scaffold reward components.

*1) **Progress Reward**:* Drawing from Shannon's information theory, we argue that effective teaching should maximize information gain, helping the student's cognitive state transition from high uncertainty to high certainty. This shift corresponds to a move from high entropy to low entropy, a process we term **Cognitive Progress**. We formalize this concept as follows: given a student's cognitive state $\psi_t$ at time $t$, the cognitive progress after a teacher's action $a_t$ is ideally measured by the reduction in entropy:

$$\Delta H_t = H(\psi_t) - H(\psi_{t+1}|a_t), \quad (2)$$

where $H(\cdot)$ represents information entropy.

However, directly calculating the entropy $H(\psi_t)$ of student cognitive states is impractical. We argue that increases in the student model's probability of outputting correct answers essentially equal cognitive state transitions. These transitions shift from high-entropy uncertain states to low-entropy certain states. Based on this insight, we design a composite reward function $r_{\text{progress}}$ to approximate student cognitive progress. This function quantifies progress through two dimensions. The first dimension is the student model's **potential capacity**. The second dimension is the **semantic quality** of its actual outputs.
**Potential Capability Score.** This term directly measures the confidence of the fixed student model $\pi_S^{\text{fixed}}$ in correct paths. We effectively use it as a core proxy for cognitive progress. We calculate the score accurately as:

$$f_{\text{potential}}(s_t, a_t) = \tanh\left(\alpha \cdot \max_{k=1}^{n} \sum_{j=1}^{m_k} \log \pi_S^{\text{fixed}}(c_{t,j}^k | s_t, a_t, c_{t,1:j-1}^k)\right), \quad (3)$$

where $\{c_t^1, c_t^2, ..., c_t^n\}$ is the complete set of candidate correct reasoning responses generated by the oracle model for the student model in the $t$-th interaction round. Each candidate $c_t^k$ is a token sequence of length $m_k$, where $c_{t,j}^k$ denotes the $j$-th token in candidate $k$. The term $\max_{k=1}^n$ effectively captures the highest log probability that the student model assigns across all oracle-generated candidates, computed autoregressively over the entire token sequence. This directly reflects the student model's peak performance potential at the current step. A higher probability value clearly indicates that the student model has greater confidence in generating correct reasoning paths. This corresponds to a cognitive shift from uncertainty (high entropy) to certainty (low entropy), serving as a key signal for measuring cognitive progress.

We introduce the $\tanh$ function and temperature coefficient $\alpha$ to convert log-likelihoods. The range of log-likelihoods is $(-\infty, 0]$. The $\tanh$ function maps values to the bounded range $[-1, 1]$. This ensures numerical stability and penalizes outputs with insufficient confidence. These outputs still reflect the student's confused state. The temperature coefficient $\alpha > 0$ controls reward sensitivity. A larger $\alpha$ strongly rewards high-confidence paths. A smaller $\alpha$ rewards even minor cognitive

improvements. Through this design, $f_{\text{potential}}$ quantifies the student model's cognitive progress.

**Semantic Quality Score.** Besides potential ability assessment, we also need to carefully measure whether the student's actual output is semantically close to the correct answer. We formally define this important component as:

$$f_{\text{semantic}}(s_t, a_t, o_{t+1}) = \max_{c_j \in \mathcal{C}} \text{sim}(\mathbf{e}(o_{t+1}), \mathbf{e}(c_j)) - \delta, \quad (4)$$

where $\mathbf{e}(\cdot)$ denotes a text embedding model (e.g., BAAI General Embedding [37]) that maps text into a vector space, $\text{sim}(\cdot, \cdot)$ accurately measures cosine similarity, and $\mathcal{C}$ is the set of candidate correct answers. The $\max$ operator effectively captures that semantic alignment with any correct answer indicates concept mastery. The bias term $\delta$ filters out semantically ambiguous or low-quality outputs.

We integrate potential ability and semantic quality through the tunable hyperparameter $\lambda \in [0, 1]$. This effectively captures both internal ability improvement (via confidence change) and external semantic performance (via output quality). It directly provides the teacher agent with a comprehensive cognitive progress signal, defined as:

$$r_{\text{progress}} = \lambda \cdot f_{\text{potential}}(s_t, a_t) + (1 - \lambda) \cdot f_{\text{semantic}}(s_t, a_t, o_{t+1}). \quad (5)$$

*2) **Scaffold Reward**:* The scaffold reward $r_{\text{scaffold}}$ ensures teaching difficulty matches students' current ability levels. It complements $r_{\text{progress}}$, which maximizes cognitive advancement. Teaching practice shows tasks that are too easy cause students to lose interest. Tasks that are too hard lead to frustration. Ideal teaching maintains task difficulty within a range where students need moderate effort to succeed. Vygotsky calls this range the Zone of Proximal Development [15]. We transform this educational theory into a computable mechanism. We discretize teaching actions into five **scaffold levels** $\mathcal{L} = \{\ell_0, ..., \ell_4\}$ based on cognitive load from high to low: $\ell_0$ (**Metacognitive Hints**) uses open-ended questions to guide student thinking without providing substantive hints; $\ell_1$ (**Strategic Hints**) provides macro-level problem-solving directions without specific methods or steps; $\ell_2$ (**Conceptual Hints**) explicitly identifies key knowledge points or theorems needed for the problem; $\ell_3$ (**Step-by-Step Hints**) provides specific operational instructions to guide students through the solution process; and $\ell_4$ (**Example Demonstration Hints**) offers reference through complete solutions to similar problems, imposing minimal cognitive load.

We design a systematic strategy to dynamically locate the ZPD based on this hierarchy. In each interaction round, the oracle model generates five hints at different levels $\{\text{hint}_{\ell_0}, ..., \text{hint}_{\ell_4}\}$ for the student model. We then carefully evaluate the probability that the student model generates correct candidate responses under each level's hint. For each scaffold level $\ell$, we calculate this probability as:

$$P(\ell) = \log \pi_S(c_t | s_t, \text{hint}_\ell), \quad (6)$$

where $c_t$ specifically represents a single sample from the candidate response set, $s_t$ clearly denotes the current student state, and $\text{hint}_\ell$ is the corresponding hint at scaffold level $\ell$.

We identify the optimal level that maximizes $P(\ell)$ to determine the student's current ability range. We then lower the target scaffold level by exactly one grade. This effectively shifts teaching difficulty from the student's comfort zone to the ZPD region. We formalize this process as:

$$\ell_{\text{ZPD}} = \max(0, \arg\max_{\ell \in \mathcal{L}} P(\ell) - 1), \quad (7)$$

where $\arg\max_{\ell \in \mathcal{L}} P(\ell)$ specifically identifies the optimal level that maximizes success probability. The minus-one operation strategically sets the target level exactly one grade below this ability range, thereby creating moderate challenge.

Based on $\ell_{\text{ZPD}}$, we define the scaffolding reward $r_{\text{scaffold},t}$:

$$r_{\text{scaffold}} = \begin{cases} \sigma(P(\ell(a_t))) + 0.5 & \text{if } \ell(a_t) = \ell_{\text{ZPD}} \\ -c \cdot |\text{index}(\ell(a_t)) - \ell_{\text{ZPD}}| & \text{otherwise,} \end{cases} \quad (8)$$

where the function $\text{index}(\cdot)$ specifically converts levels to integer indices (0 to 4), $P(\ell(a_t))$ is the corresponding log probability of the student model outputting the correct candidate response, $\sigma$ is the standard Sigmoid function, and $c > 0$ is a penalty coefficient. When $\ell(a_t) = \ell_{\text{ZPD}}$, the model receives a positive reward with two parts: a fixed base reward of $0.5$ and a dynamic reward $\sigma(P(\ell(a_t)))$ based on the log probability. When deviating from the target, the model receives a negative reward directly proportional to the index distance.

This reward mechanism effectively combines both positive and negative incentives. It guides the teacher agent to provide appropriate instructional scaffolding within the student's ZPD. This avoids both excessive support that weakens learning and insufficient support that hinders learning.

For each teacher response within each complete rollout, our designed reward function systematically combines both reward components into the final total reward, with the progress reward driving cognitive advancement and the scaffold reward ensuring pedagogical appropriateness:

$$r_t^{(i)} = r_{\text{progress}} + r_{\text{scaffold}}, \quad (9)$$

where $r_t^{(i)}$ denotes the reward for the teacher model's response at round $t$ in the $i$-th rollout.

### C. Policy Optimization via GRPO

We use GRPO to optimize the teacher model $\pi_{\text{teacher}}$. Algorithm 1 presents the complete optimization procedure. For each training iteration, the algorithm first generates $G$ rollouts per problem through teacher-student interactions. It then computes the dual-objective rewards across dialogue turns, normalizes advantages within each problem group to balance reward scales, and finally updates the teacher policy parameters via the policy gradient objective. This grouped normalization approach enhances training stability when optimizing for both progress and scaffolding objectives.

**Algorithm 1** UCO-GRPO Algorithm

**Input:** Teacher model $\pi_{\text{teacher}}$, reference model $\pi^{\text{ref}}_{\text{teacher}}$, student model $\pi^{\text{fixed}}_{\text{student}}$, problem distribution $\mathcal{D}$, oracle model.
**Hyperparameters:** Batch size $|\mathcal{B}|$, rollouts per problem $G$, dialogue turns $T$, discount factor $\gamma$.
**for** each training iteration **do**
  Sample problem batch $\mathcal{B} \sim \mathcal{D}$.
  ▷ *Rollout Generation*
  **for all** problem $q \in \mathcal{B}$ **do**
    **for** $i = 1$ to $G$ **do**
      Initialize $s_0^{(i)} \leftarrow q$.
      **for** $t = 0$ to $T-1$ **do**
        Teacher acts: $a_t^{(i)} \sim \pi_{\text{teacher}}(\cdot|s_t^{(i)})$.
        Student responds: $o_{t+1}^{(i)} \sim \pi^{\text{fixed}}_{\text{student}}(\cdot|s_t^{(i)}, a_t^{(i)})$.
        Compute $r_t^{(i)} = r_{\text{progress}} + r_{\text{scaffold}}$ ▷ Eq. (9)
        Update $s_{t+1}^{(i)} \leftarrow \{s_t^{(i)}, a_t^{(i)}, o_{t+1}^{(i)}\}$.
      Compute cumulative reward: $\mathcal{R}_i = \sum_{t=0}^{T-1} \gamma^t r_t^{(i)}$ ▷ Eq. (10)
  ▷ *Advantage Normalization*
  **for all** problem $q \in \mathcal{B}$ **do**
    Compute $\mu_q, \sigma_q$ over $\{\mathcal{R}_1, ..., \mathcal{R}_G\}$ ▷ Eq. (11)
    Compute advantages: $A_i = \frac{\mathcal{R}_i - \mu_q}{\sigma_q + \epsilon}$ for $i = 1, ..., G$ ▷ Eq. (12)
  ▷ *Policy Update*
  Compute GRPO objective $\mathcal{J}(\theta_{\text{teacher}})$ ▷ Eq. (13)
  Update: $\theta_{\text{teacher}} \leftarrow \theta_{\text{teacher}} + \eta \cdot \nabla_{\theta_{\text{teacher}}} \mathcal{J}(\theta_{\text{teacher}})$.
**Return:** Optimized teacher model $\pi_{\text{teacher}}$.

*1) **Grouped Rollout Sampling**:* We consider each individual problem $q$ in the training batch $\mathcal{B}$. We sample $G$ separate interaction rollouts in parallel for each problem. We use the current teacher model $\pi_{\text{teacher}}$ and the student model $\pi^{\text{fixed}}_{\text{student}}$. Each rollout contains state sequences, teacher actions, and student observations. We then compute the cumulative reward for the $i$-th rollout ($i \in \{1, 2, \ldots, G\}$) on problem $q$. We sum the discounted immediate rewards:

$$\mathcal{R}_t^{(i)} = \sum_{t=0}^{T} \gamma^t r_t^{(i)}, \tag{10}$$

where $\gamma \in [0, 1]$ is the discount factor, $r_t^{(i)}$ is the immediate reward at turn $t$ in the $i$-th rollout.

*2) **Intra-Group Advantage Normalization**:* We compute the group-level mean and standard deviation for each problem $q$. We calculate them across its $G$ rollouts:

$$\mu_q = \frac{1}{G}\sum_{i=1}^{G} \mathcal{R}_i, \quad \sigma_q = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(\mathcal{R}_i - \mu_q)^2}. \tag{11}$$

We then calculate the standardized advantage for each rollout among these $G$ rollouts:

$$A_i = \frac{\mathcal{R}_i - \mu_q}{\sigma_q + \epsilon}, \tag{12}$$

where $\epsilon$ is a small constant. It prevents division by zero.

*3) **Policy Gradient Update**:* We maximize an objective function to update the parameters $\theta_{\text{teacher}}$ of the teacher model $\pi_{\text{teacher}}$. This objective combines policy gradient and KL divergence regularization. The KL divergence term constrains the magnitude of policy updates. It prevents the policy from deviating too far from a reference policy $\pi^{\text{ref}}_{\text{teacher}}$. It also ensures fluency and coherence of the generated teaching content. We define the optimization objective $\mathcal{J}(\theta_{\text{teacher}})$ as:

$$\mathcal{J}(\theta_{\text{teacher}}) = \mathbb{E}_{q\sim\mathcal{B}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(A_i \cdot \sum_{t=0}^{T} \log \pi_{\text{teacher}}(a_t^{(i)}|s_t^{(i)}) - \xi \cdot D_{\text{KL}}[\pi_{\text{teacher}}(\cdot|s_t^{(i)}) \| \pi^{\text{ref}}_{\text{teacher}}(\cdot|s_t^{(i)})]\right)\right], \tag{13}$$

where $\xi > 0$ is the weight coefficient for KL regularization. We compute the gradient of this objective using the policy gradient theorem. We then update the model parameters accordingly. This optimization process guides the teacher's policy to generate high-reward teaching actions. It maintains stability at the same time.

## IV. Experiments

### A. Experimental Setup

**Dataset.** We use the BigMath dataset [26] as our training data, following [6]. We randomly sample 500 samples from the 10,000 available samples constructed by [6]. This training set serves two important purposes. First, it better validates the effectiveness of our method. Second, it significantly reduces the computational cost of RL training.

**Metrics.** We comprehensively evaluate tutoring performance on two separate datasets. Following [6], we evaluate on 500 held-out test samples. The evaluation metrics include: ΔSolve Rate, which quantifies improvements in student problem-solving ability before and after dialogue. Leak Solution leverages an LLM-based adjudicator to quantify the prevalence of dialogues in which instructors explicitly divulge solutions. Ped-RM micro/macro applies the pedagogical reward model from MathTutorBench [27] to assess overall teaching quality.

We evaluate the benchmark established by [27], comprising nine distinct test sets spanning three main categorical domains. Math Expertise evaluates mathematical competence through Problem Solving (accuracy on generating correct numerical answers) and Socratic Questioning (BLEU score on generating guiding questions for correct steps). Student Understanding assesses comprehension of student work through Solution Correctness (F1 micro on judging answer correctness), Mistake Location (F1 on identifying the first student error), and Mistake Correction (accuracy on generating correct solutions after errors). Pedagogy measures teaching ability across four different tasks using the win rate. Win rate specifically denotes the frequency with which the reward model favors model-generated responses over teacher responses. Tasks include Scaffolding Generation (generating teacher responses to student errors), Pedagogical IF (following detailed pedagogical

instructions), and their respective Hard versions (evaluated in extended dialogues averaging 5.78 turns).

**Baselines.** We evaluate 11 mainstream large language models (LLMs). It includes three closed-source models, four open-source models, and four education-specific models. (a) The closed-source models include GPT-4o-2024-11-20 [38], LearnLM 1.5 Pro Experimental [39], and LearnLM 2.0 Flash Experimental [39]. (b) We select open-source models based on two factors. First, we group them by parameter size: small (7B), medium (14B), and large (72B and 671B). Second, we categorize them by functionality. We include general instruction-following models (Qwen2.5-7B-Instruct [40], Qwen2.5-14B-Instruct [40], Qwen2.5-72B-Instruct [40]). We also include a general reasoning model, DeepSeek-V3-0324 [41]. (c) We evaluate four education-specific models. These include Qwen2.5-7B-SFT [1], Qwen2.5-7B-MDPO [10], SocraticLM [3], and Qwen2.5-7B-RL [6].

**Implementation Details.** We conduct all experiments on a server equipped with 8 NVIDIA H200 GPUs. The complete experimental environment runs Python 3.11, CUDA 12.8, and PyTorch 2.7. We use Qwen2.5-7B-Instruct as the primary teacher model $\pi_{teacher}$. We employ Qwen2.5-14B-Instruct as the student model $\pi^{\text{fixed}}_{student}$. We specifically utilize Gemini-2.5-Pro-Exp-03-25 [42] as the sampling model to generate candidate responses. At each interaction turn, we generate one single teacher response per scaffolding level. We also sample $K = 5$ diverse candidate student responses.

We determine all hyperparameters through grid search. We set the learning rate to $\eta = 5 \times 10^{-6}$ and the batch size to $|\mathcal{B}| = 16$. We sample $G = 4$ interaction rollouts per problem. We set the maximum dialogue length to $T = 10$ and the discount factor to $\gamma = 0.95$. Within the cognitive progress reward $r_{\text{progress}}$, we set the balance coefficient to $\lambda = 0.5$. This coefficient balances potential capacity and semantic quality. We set the temperature coefficient for potential score to $\alpha = 2.0$. We set the bias term for the semantic quality score to $\delta = 0.7$. We employ the BAAI General Embedding (BGE [37]) model to compute text embeddings and similarity scores.

For the reward $r_{\text{scaffold}}$, we set the deviation penalty coefficient to $c = 0.2$. In GRPO optimization, we set the KL divergence regularization weight to $\xi = 0.01$. We set the stabilization term for advantage normalization to $\epsilon = 10^{-8}$. We use the AdamW optimizer with a weight decay of $10^{-4}$. The complete training process takes approximately 28 hours.

### *B. Main Results*

*1)* ***Performance Comparison on BigMath Benchmark****:* Table I shows that existing models struggle to balance three dimensions: teaching effectiveness ($\Delta$ Solve Rate), answer leakage (Leak Solution), and teaching quality (Ped-RM micro/macro). General-purpose models prioritize answers over teaching. Closed-source GPT-4o achieves $33.1\%$ $\Delta$ Solve Rate but has $35.2\%$ Leak Solution and only $1.5/-0.3$ Ped-RM scores. Among open-source models, larger parameters worsen the imbalance. The Qwen2.5 series increases $\Delta$ Solve Rate from $11.3\%$ to $38.7\%$ as parameters grow from 7B to 72B.

However, Leak Solution rises simultaneously from $29.3\%$ to $61.0\%$, while Ped-RM remains negative. DeepSeek-V3 has 671B parameters and the highest $\Delta$ Solve Rate of $39.3\%$. Yet it exhibits $46.6\%$ Leak Solution with Ped-RM Micro/Macro scores of $-1.5/-0.8$. It reveals that large-scale models fundamentally function as "solvers" rather than effective teachers, lacking pedagogical optimization.

Education-specialized models exhibit a different pattern. SFT and MDPO achieve lower Leak Solution rates of $36.0\%$ and $35.6\%$, respectively. However, their $\Delta$ Solve Rates drop to $8.9\%$ and $16.4\%$, trailing behind base models. SocraticLM and Qwen2.5-7B-RL achieve a better balance between the two objectives. Specifically, Qwen2.5-7B-RL reaches $29.1\%$ $\Delta$ Solve Rate with $15.1\%$ Leak Solution. But its Ped-RM scores of $3.6/3.1$ still leave room for improvement. Closed-source LearnLM 2.0 reaches the best Ped-RM scores of $6.8/6.4$ and the lowest Leak Solution of $0.9\%$. Yet its $\Delta$ Solve Rate is only $4.3\%$. It demonstrates the inherent trade-off: strict adherence to pedagogical principles can compromise student learning outcomes. **Our UCO achieves synergistic optimization with only 7B parameters.** It reaches $30.2\%$ $\Delta$ Solve Rate, surpassing similar-sized models and approaching larger models. UCO achieves $12.9\%$ Leak Solution, representing a $14.6$ percentage point improvement over the strongest teaching baseline, Qwen2.5-7B-RL. Its Ped-RM Micro/Macro scores reach $4.6/4.5$, improving by $28\%$. This three-way balance validates our dual-objective reward mechanism.

Our core advantage lies in explicitly modeling student cognitive states. Learning progress reward quantifies student cognitive changes. It provides rewards only when students truly understand and correctly express reasoning steps. This design directly incentivizes the teacher model to promote genuine cognitive progress. It thus improves the $\Delta$ Solve Rate effectively. Scaffolding reward operationalizes ZPD theory. It estimates student ability distributions across five scaffolding levels. It precisely identifies the student's zone of proximal development. The reward mechanism encourages strategies within this zone while discouraging both overly directive and insufficiently supportive interventions. It enables adaptive teaching that avoids direct answers and prevents inefficient over-questioning. In summary, UCO leverages differentiable cognitive reward signals to empower compact models. These models surpass similarly sized baselines across all three dimensions. They achieve comprehensive performance that rivals the closed-source models.

*2)* ***Comprehensive Evaluation on MathTutorBench****:* Table II shows the evaluation results on MathTutorBench. We compare UCO with education-specific models. This benchmark covers nine tasks across three categories: Math Expertise, Student Understanding, and Pedagogy. Qwen2.5-7B-SFT performs well in Math Expertise. However, it lags significantly in Student Understanding and Pedagogy. Qwen2.5-7B-MDPO improves math solving and student understanding through direct preference optimization. However, this causes severe degradation in Pedagogy. SocraticLM excels in Pedagogy but shows weaknesses in Math Expertise and Student Understand-

TABLE I
PERFORMANCE COMPARISON ON BIGMATH BENCHMARK ACROSS TEACHING EFFECTIVENESS (ΔSOLVE RATE), SOLUTION LEAKAGE (LEAK SOLUTION), AND PEDAGOGICAL QUALITY (PED-RM).

| Models | ΔSolve Rate (%) ↑ | Leak Solution (%) ↓ | Ped-RM micro/macro ↑ |
|---|---|---|---|
| *Closed-source Models* | | | |
| GPT-4o-2024-11-20 [38] | 33.1 | 35.2 | 1.5/-0.3 |
| LearnLM 1.5 Pro Exp. [39] | 1.5 | 2.6 | 5.9/5.3 |
| LearnLM 2.0 Flash Exp. [39] | 4.3 | **0.9** | **6.8/6.4** |
| *Open-source Models* | | | |
| Qwen2.5-7B-Instruct [40] | 11.3 | 29.3 | -0.2/-0.5 |
| Qwen2.5-14B-Instruct [40] | 29.3 | 41.9 | -0.6/-1.2 |
| Qwen2.5-72B-Instruct [40] | 38.7 | 61.0 | 1.8/-0.4 |
| DeepSeek-V3-0324 [41] | **39.3** | 46.6 | -1.5/-0.8 |
| *Education-specific Models* | | | |
| Qwen2.5-7B-SFT [1] | 8.9 | 36.0 | -0.3/-0.7 |
| Qwen2.5-7B-MDPO [10] | 16.4 | 35.6 | 0.2/-0.3 |
| SocraticLM [3] | 15.9 | 40.4 | 1.7/1.7 |
| Qwen2.5-7B-RL [6] | 29.1 | 15.1 | 3.6/3.1 |
| **UCO** | 30.2 | 12.9 | 4.6/4.5 |

TABLE II
COMPREHENSIVE EVALUATION ON MATHTUTORBENCH ACROSS MATH EXPERTISE, STUDENT UNDERSTANDING, AND PEDAGOGICAL CAPABILITIES.

| Model | Math Expertise | | Student Understanding | | | Pedagogy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Problem solving | Socratic questioning | Solution correctness | Mistake location | Mistake correction | Teacher response generation | | | |
| | | | | | | scaff. | ped.IF | scaff. [hard] | ped.IF [hard] |
| | acc. | BLEU | F1 | micro F1 | acc. | win rate over human teacher | | | |
| *Education-specific Models* | | | | | | | | | |
| Qwen2.5-7B-SFT | 0.77 | 0.24 | 0.27 | **0.45** | 0.10 | 0.64 | 0.58 | 0.57 | 0.59 |
| Qwen2.5-7B-MDPO | 0.86 | 0.23 | 0.62 | 0.39 | 0.03 | 0.37 | 0.60 | 0.47 | 0.56 |
| SocraticLM | 0.73 | **0.32** | 0.05 | 0.39 | **0.23** | 0.39 | 0.39 | 0.28 | 0.28 |
| Qwen2.5-7B-RL | 0.83 | 0.23 | **0.67** | 0.35 | 0.05 | 0.57 | 0.72 | 0.61 | 0.69 |
| **UCO** | **0.89** | 0.24 | 0.65 | **0.45** | 0.08 | **0.68** | **0.74** | **0.72** | **0.77** |

ing. Qwen2.5-7B-RL achieves balance across three dimensions but lacks overall strength.

In contrast, our model UCO performs strongly in Math Expertise and Student Understanding. It achieves optimal performance in Pedagogy. This advantage stems from UCO's cognitive state modeling mechanism. Learning progress rewards track student cognitive changes. Scaffolding rewards dynamically adjust teaching strategies. The model maintains high-quality teaching even in complex multi-turn interactions.

### C. Ablation Study

*1) **Ablation Study on BigMath Dataset**:* Table III validates the necessity of our reward components through ablation studies on BigMath. The entire UCO model achieves 30.2% Δ Solve Rate, 12.9% Leak Solution, and 4.6/4.5 Ped-RM micro/macro. Removing scaffolding rewards causes the most severe degradation. Compared to the entire setup, Δ Solve Rate decreases by 6.7 percentage points to 23.5%, Leak Solution rises by 7.3 percentage points to 20.2%, and Ped-RM micro/macro drops to 2.2/2.3. Without scaffolding guidance, the model fails to adapt to students' zones of proximal development, resulting in inefficient teaching. Removing progress rewards leads to smaller but notable declines. Compared to the entire setup, Δ Solve Rate drops by 1.6 percentage points to 28.6%, Leak Solution increases by 2.7 percentage points to 15.6%, and Ped-RM micro/macro decreases to 4.1/3.9. The model struggles to track student understanding and cannot adjust teaching strategies based on learning states. All reward configurations outperform the backbone, confirming the effectiveness of our reward design.

TABLE III
ABLATION STUDY ON BIGMATH DEMONSTRATING THE CONTRIBUTION OF PROGRESS AND SCAFFOLDING REWARD COMPONENTS.

| Models | ΔSolve Rate (%) ↑ | Leak Solution (%) ↓ | Ped-RM micro/macro ↑ |
|---|---|---|---|
| *Backbone Model* | | | |
| Qwen2.5-7B-Instruct [40] | 11.3 | 29.3 | -0.2/-0.5 |
| **UCO (full)** | **30.2** | **12.9** | **4.6/4.5** |
| **UCO w/o scaffold reward** | 23.5 | 20.2 | 2.2/2.3 |
| **UCO w/o progress reward** | 28.6 | 15.6 | 4.1/3.9 |

TABLE IV
ABLATION STUDY ON MATHTUTORBENCH SHOWING THE IMPACT OF DIFFERENT REWARD COMPONENTS ACROSS ALL EVALUATION DIMENSIONS.

| Model | Math Expertise | | Student Understanding | | | Pedagogy | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Problem solving | Socratic questioning | Solution correctness | Mistake location | Mistake correction | Teacher response generation | | | |
| | | | | | | scaff. | ped.IF | scaff. [hard] | ped.IF [hard] |
| | acc. | BLEU | F1 | micro F1 | acc. | win rate over human teacher | | | |
| *Backbone Model* | | | | | | | | | |
| Qwen2.5-7B-Instruct [40] | 0.87 | 0.23 | 0.63 | 0.39 | 0.04 | 0.37 | 0.60 | 0.45 | 0.56 |
| **UCO (full)** | **0.89** | **0.24** | **0.65** | **0.45** | **0.08** | **0.68** | **0.74** | **0.72** | **0.77** |
| **UCO w/o scaffold reward** | 0.86 | 0.23 | 0.61 | 0.40 | 0.05 | 0.56 | 0.70 | 0.67 | 0.71 |
| **UCO w/o progress reward** | 0.88 | **0.24** | 0.63 | 0.42 | 0.07 | 0.61 | 0.71 | 0.69 | 0.74 |

*2) **Ablation Study on MathTutorBench Dataset**:* Table IV further validates our reward function through ablation studies on MathTutorBench. The entire UCO model performs well across all three dimensions. We remove scaffolding rewards and observe significant declines: $3.4\%$ in Math Expertise, $9.0\%$ in Student Understanding, and $10.6\%$ in Pedagogy. Without scaffolding guidance, the model fails to break down content into progressive steps. It causes a mismatch between teaching strategies and student capabilities. Removing progress rewards has a greater impact on recognizing student understanding states (Student Understanding) and dynamically adjusting teaching strategies (Pedagogy), but less impact on accurately delivering mathematical knowledge (Math Expertise). It indicates that progress rewards primarily regulate teaching pace based on student feedback.

*3) **Impact of Rollout Numbers on Training Efficiency**:* We conduct comprehensive ablation studies on rollout numbers using a smaller training set of 100 samples. Figure 3 shows the impact of different rollouts on three key metrics. As rollout numbers increase progressively from 2 to 8, performance improves consistently. Figure 3(a) shows that ΔSolve Rate increases from $14.6\%$ to $20.2\%$. Figure 3(b) demonstrates that Leak Solution decreases from $25.2\%$ to $18.8\%$. Figure 3(c) illustrates that Ped-RM scores also improve steadily, with micro scores rising from $0.6$ to $2.3$ and macro scores from $0.5$ to $2.1$. However, training cost grows substantially with rollout numbers. Figure 3(d) shows the detailed training time comparison. With 100 training samples, rollout=2 takes approximately 2.9 hours. Rollout=8 requires 10.8 hours, a 3.7-fold increase. We choose rollout=4 as our default setting, which achieves comparable performance to rollout=8 while reducing training time by $52\%$ (5.2 hours vs 10.8 hours). This provides the optimal trade-off between performance and efficiency.

*4) **Impact of Balance Coefficient** $\lambda$:* Table V examines how $\lambda$ balances potential capacity and semantic quality in progress rewards. At low values ($\lambda = 0.1$ and $0.3$), semantic quality dominates the reward signal. Rewards depend only on output correctness. Even when the student internally understands correct reasoning, rewards stay low if the output is inaccurate. It prevents the teacher from adjusting teaching strategies to guide the student. This imbalance restricts the model's ability to recognize internal cognitive improvements effectively. It limits improvements in ΔSolve Rate and teaching quality scores. At high values ($\lambda = 0.7$ and $0.9$), potential capacity dominates the reward signal. Rewards mainly depend on increases in internal confidence and ignore output accuracy. To quickly boost student confidence, the teacher may directly provide answer-related information rather than guide autonomous reasoning. It increases answer leakage rates while decreasing ΔSolve Rate and Ped-RM scores. At $\lambda = 0.5$, performance reaches the optimum. The potential capacity component measures student confidence in correct reasoning. The semantic quality component verifies output correctness. Together, they enable the teacher to optimize both reasoning capability and expression quality. It achieves the best performance across all metrics.

TABLE V
ABLATION STUDY ON BALANCE COEFFICIENT $\lambda$ BETWEEN POTENTIAL CAPACITY AND SEMANTIC QUALITY IN PROGRESS REWARD.

| $\lambda$ | **ΔSolve Rate (%) ↑** | **Leak Solution (%) ↓** | **Ped-RM micro/macro ↑** |
|---|---|---|---|
| 0.1 | 26.5 | 16.8 | 3.5/3.6 |
| 0.3 | 27.8 | 15.3 | 3.8/4.0 |
| 0.5 | **30.2** | **12.9** | **4.6/4.5** |
| 0.7 | 29.1 | 13.5 | 4.3/4.2 |
| 0.9 | 27.4 | 14.9 | 3.9/3.7 |

*5) **Impact of Temperature Coefficient** $\alpha$:* Table VI examines how the temperature coefficient $\alpha$ affects reward signal strength in potential ability scoring. At lower values ($\alpha = 1.0, 1.5$), the tanh function produces weak reward signals. Consequently, these signals cannot effectively distinguish different confidence levels in student responses. This insufficient discrimination prevents the teacher model from identifying meaningful cognitive progress. As a result, without clear learning signals, the model cannot effectively guide students toward correct reasoning paths. Ultimately, this leads to poor teaching performance across all evaluation metrics. At higher values ($\alpha = 2.5, 3.0$), reward signals become overly sensitive to small confidence fluctuations. They amplify minor probability changes into large reward differences. This excessive sensitivity introduces noise into the training signals, which disrupts learning stability and leads to inconsistent teaching behavior. Such instability consequently compromises the model's overall performance. At the optimal value ($\alpha = 2.0$), the temperature coefficient achieves an ideal balance. Reward signals maintain sufficient discriminative strength while preserving stability against noise amplification. This balance enables the teacher model to reliably identify genuine cognitive improvements and deliver consistent, effective teaching strategies.

TABLE VI
ABLATION STUDY ON TEMPERATURE COEFFICIENT $\alpha$ CONTROLLING REWARD SENSITIVITY IN POTENTIAL CAPACITY SCORE.

| $\alpha$ | **ΔSolve Rate (%) ↑** | **Leak Solution (%) ↓** | **Ped-RM micro/macro ↑** |
|---|---|---|---|
| 1.0 | 27.4 | 15.1 | 3.8/4.0 |
| 1.5 | 28.9 | 13.8 | 4.1/4.2 |
| 2.0 | **30.2** | **12.9** | **4.5/4.4** |
| 2.5 | 28.6 | 14.3 | 4.2/4.0 |
| 3.0 | 27.1 | 16.1 | 3.3/2.8 |

*6) **Impact of Semantic Quality Threshold** $\delta$:* Table VII examines how the bias term $\delta$ affects output filtering in semantic quality scoring. At lower values ($\delta = 0.5, 0.6$), the threshold is too lenient. Consequently, the model cannot effectively distinguish genuine understanding from shallow pattern matching. This causes the reward function to inappropriately reward superficial matches lacking true reasoning, resulting in teaching strategies that lack precision and effectiveness. At higher values ($\delta = 0.8, 0.9$), the threshold becomes overly strict. Consequently, many responses demonstrating valid understanding are incorrectly filtered out. This causes the reward signal to inadequately capture the actual progression in students' cognitive development. As a result, the teacher model tends to provide excessive scaffolding, which undermines students' autonomous reasoning capacity and increases answer leakage risk. At the optimal value $\delta = 0.7$, the threshold achieves an ideal balance. Unlike lower values that allow noisy signals to mislead the optimization process, or higher values that discard valuable learning signals, this optimal setting enables the model to accurately filter ambiguous outputs while reliably identifying genuine understanding. Consequently, the reward function precisely evaluates students' cognitive states, and performance is optimized across all metrics.

TABLE VII
ABLATION STUDY ON SEMANTIC QUALITY THRESHOLD $\delta$ FOR FILTERING AMBIGUOUS OUTPUTS IN SEMANTIC QUALITY SCORE.

| $\delta$ | **ΔSolve Rate (%) ↑** | **Leak Solution (%) ↓** | **Ped-RM micro/macro ↑** |
|---|---|---|---|
| 0.5 | 27.2 | 16.1 | 3.7/3.9 |
| 0.6 | 28.6 | 14.5 | 4.0/4.2 |
| 0.7 | **30.2** | **12.9** | **4.6/4.5** |
| 0.8 | 29.3 | 13.7 | 4.2/4.3 |
| 0.9 | 26.8 | 17.3 | 3.6/3.8 |

TABLE VIII
ABLATION STUDY ON DEVIATION PENALTY COEFFICIENT $c$ REGULATING SCAFFOLDING PRECISION IN SCAFFOLDING REWARD.

| $c$ | **ΔSolve Rate (%) ↑** | **Leak Solution (%) ↓** | **Ped-RM micro/macro ↑** |
|---|---|---|---|
| 0.1 | 27.6 | 15.8 | 3.8/3.9 |
| 0.2 | **30.2** | **12.9** | **4.6/4.5** |
| 0.3 | 28.7 | 14.1 | 4.2/4.1 |
| 0.4 | 27.3 | 15.6 | 3.9/4.0 |
| 0.5 | 26.1 | 17.2 | 3.5/3.7 |

*7) **Impact of Deviation Penalty Coefficient** $c$:* Table VIII examines how the deviation penalty coefficient $c$ regulates scaffolding precision. This coefficient penalizes teaching strategies that deviate from students' Zone of Proximal Development (ZPD). At lower values ($c = 0.1$), the penalty strength is insufficient. The model receives weak feedback signals for scaffolding deviations. The teacher model struggles to effectively adjust teaching difficulty to appropriate levels. All evaluation metrics consequently show suboptimal performance. At higher values ($c = 0.3, 0.4, 0.5$), penalty strength becomes overly strict. The model's teaching strategies are consequently over-constrained. The teacher model's ability to respond to individual student needs declines accordingly. Limited teaching flexibility leads to overall performance degradation. At the optimal value ($c = 0.2$), penalty strength achieves an ideal balance. The model receives sufficient incentive to maintain scaffolding precisely within the ZPD range. All evaluation metrics achieve optimal performance under this configuration.

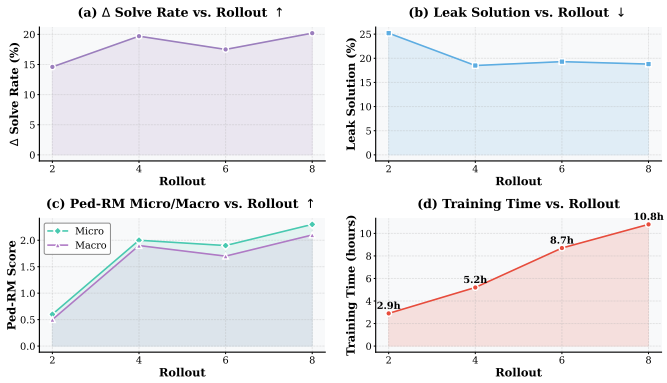


Fig. 3. Impact of rollout numbers on model performance and training efficiency.

*8) **General Mathematical Reasoning Ability Preservation**:* Table IX shows UCO's performance on general math reasoning benchmarks. UCO achieves 85.9% and 75.0% accuracy on GSM8K and MATH500. This represents only 0.9 and 0.4 percentage point drops from the backbone model. Other education-specific models show larger performance declines. Qwen2.5-7B-SFT drops by 7.3 and 9.4 percentage points. Qwen2.5-7B-RL drops by 0.7 and 1.8 percentage points. Qwen2.5-7B-MDPO performs best on general ability. However, it shows very limited improvement in teaching quality. This demonstrates that UCO achieves the best trade-off between teaching specialization and general ability.

TABLE IX
EVALUATION OF GENERAL MATHEMATICAL REASONING BENCHMARKS GSM8K AND MATH500.

| Model | GSM8K | MATH500 |
|---|---|---|
| *Backbone Model* | | |
| Qwen2.5-7B-Instruct [40] | 86.8 | 75.4 |
| *Education-specific Models* | | |
| Qwen2.5-7B-SFT [1] | 79.5 | 66.0 |
| Qwen2.5-7B-MDPO [10] | **87.0** | **76.4** |
| Qwen2.5-7B-RL [6] | 86.1 | 73.6 |
| **UCO** | 85.9 | 75.0 |

## D. Case Study

To verify the effectiveness of our proposed method, we conduct a detailed qualitative case study. We compare our model with GPT-4o-2024-11-20 [38], both serving as teacher models. The student model uses Qwen2.5-14B-Instruct [40]. We carefully select two representative problems from the BigMath [26] test set. The maximum dialogue turns are set to 15. The teacher model outputs `<end_of_conversation>` when it determines the conversation should end. Following prior research [6], the specific prompt designs for teacher and student models are shown in Table XII. We present two dialogue cases and conduct an in-depth comprehensive analysis to reveal the core advantages of our method:

TABLE X
CASE ONE DIALOGUE COMPARISON

| UCO | GPT-4o |
|---|---|
| *Problem: Gina can paint 6 rose cups/hour and 7 lily cups/hour. Her store receives an order for 6 rose cups and 14 lily cups. If she gets $90 total, what is her hourly pay?* | |
| **T:** To calculate a "dollars per hour" rate, what two core pieces of information do we need? | **T:** Let's solve this. What do you think we should calculate first? |
| **S:** The total dollars she made... and the total hours she worked? | **S:** Um... maybe how long it takes to finish the order? |
| **T:** Exactly. Total pay is given. So the real task is finding total hours. The order has two parts. How would you approach that? | **T:** Good thinking. To get total time, we look at two cup types separately. For 6 rose cups at 6/hour, that takes 1 hour. Clear? |
| **S:** I'd figure out time for roses, then lilies, and add them. | **S:** Clear, that's simple. |
| **T:** That's a solid plan. Go ahead, what's the time for roses? | **T:** The other part is 14 lily cups. She paints 7/hour. Can you calculate how long that takes? |
| **S:** She paints 6 at 6/hour... so 1 hour. | **S:** 14 divided by 7 is 2 hours. |
| **T:** Right. And the lilies? | **T:** Right. So total time is 1 + 2 = 3 hours. Now we know total time and pay, what's the final step? |
| **S:** 14 cups at 7/hour... 14÷7 = 2 hours. | **S:** Um... use $90 divided by 3 hours? |
| **T:** Correct. Now you have everything. What's the final step? | **T:** Yes, that gives her hourly rate. |
| **S:** Total time is 3 hours. So $90÷3 = $30/hour. | **S:** The result is $30/hour. |
| **T:** You've walked through the solution logically. `<end_of_conversation>` | **T:** Right, the final answer is $30. `<end_of_conversation>` |

For case one, this problem involves basic rate calculation. Table X shows the complete dialogue process of both models.

Our method demonstrates stronger adaptive guidance ability. First, our model activates students' metacognitive thinking through open-ended questions ("What two core pieces of information do we need?"). This heuristic questioning forces students to identify the problem's essential structure. Students do not passively accept solution steps. In contrast, GPT-4o directly states the solution direction ("we look at two cup types separately"). It deprives students of independent thinking opportunities. Second, our method exhibits more precise progressive scaffolding. When the student proposes a complete plan, the teacher model only gives affirmation and encourages execution ("That's a solid plan. Go ahead"). It transfers complete computational agency to the student. GPT-4o directly provides intermediate calculation results ("that takes 1 hour"). It interferes with students' autonomous problem-solving process. Such over-guidance diminishes students' cognitive engagement and independent reasoning. Finally, our method emphasizes students' thinking process when ending the dialogue ("You've walked through the solution logically"). It strengthens students' metacognitive reflection on solution strategies. GPT-4o only confirms answer correctness. It ignores the summary of the thinking process.

For case two, this problem examines students' comprehensive understanding of the periodicity and parity. Table XI shows different strategies of both models when handling

TABLE XI
CASE TWO DIALOGUE COMPARISON

| UCO | GPT-4o |
|---|---|
| *Problem: $f(x)$ on $\mathbb{R}$ is even, satisfies $f(x+1) = -f(x)$. When $x \in [-1, 0)$, $f(x) = (\frac{1}{2})^x$. Find $f(\log_2 8)$.* | |
| **T:** This gives three pieces of info about $f(x)$. First, it's "even." Recall its definition? | **T:** This looks complex, but the key is using given properties. We need $f(\log_2 8)$, which is $f(3)$. |
| **S:** I think it's $f(-x) = f(x)$. | **S:** But the formula's domain is $[-1, 0)$, and 3 isn't in it. |
| **T:** Correct. Second, $f(x + 1) = -f(x)$. This tells what happens when input increases by 1. What about increasing by 2? | **T:** Correct. We use properties to transform $f(3)$. Which helps? |
| **S:** Hmm... $f(x + 2) = -f(x + 1)$. Since $f(x + 1) = -f(x)$, then $f(x + 2) = f(x)$. | **S:** I'm not sure... the periodic one? |
| **T:** Your reasoning is correct. What does this property tell us? | **T:** Yes. We deduce $f(x + 2) = f(x)$, so period is 2. Using this, $f(3) = f(1)$. |
| **S:** The function is periodic with period 2. | **S:** Now we need $f(1)$. |
| **T:** Very good. Now what we need: $f(\log_2 8)$. What is $\log_2 8$? | **T:** Let's use $f(x + 1) = -f(x)$. When $x = 0$, this becomes $f(1) = -f(0)$. |
| **S:** 2 to power 3 is 8, so it's 3. We need $f(3)$. | **S:** Oh, so we just need $f(0)$. |
| **T:** Right. How can we use the period property? | **T:** Right. We use $f(x + 1) = -f(x)$ again. When $x = -1$, we get $f(0) = -f(-1)$. |
| **S:** $f(3) = f(1)$. | **S:** So $f(-1) = (1/2)^{-1} = 2$. |
| **T:** Good. Now the problem becomes finding $f(1)$. Which condition helps? | **T:** Correct. So $f(0) = -2$. |
| **S:** $f(x + 1) = -f(x)$? | **S:** Then $f(1) = -f(0) = 2$. |
| **T:** Yes, how would you use it? | **T:** Exactly. Since $f(3) = f(1)$, what's the final answer? |
| **S:** If I let $x = 0$, I get $f(1) = -f(0)$. | **S:** The final answer is 2. |
| **T:** Excellent. How can we find $f(0)$? | **T:** Correct. The answer is 2. <end_of_conversation> |
| **S:** If I let $x = -1$, I get $f(0) = -f(-1)$. | |
| **T:** Perfect! What is $f(-1)$? | |
| **S:** $f(-1) = (1/2)^{-1} = 2$. | |
| **T:** Correct. So what is $f(0)$? | |
| **S:** $f(0) = -2$. | |
| **T:** Can you now connect all results? | |
| **S:** Since $f(0) = -2$, then $f(1) = 2$. And $f(3) = 2$. | |
| **T:** Exactly right! You successfully applied multiple function properties to complete the derivation. <end_of_conversation> | |

TABLE XII
TEACHER AND STUDENT PROMPTS

| Teacher Prompt |
|---|
| You are tasked with being a teacher and helping a student with a math problem. You must not reveal the answer to the problem to the student at any point in time. Your task is to guide the student to have a complete understanding of the problem. Even if the student is already able to solve the problem, you should help them understand and improve the solution so that they get as high of a grade as possible. If possible, do not respond with overly long responses to the student. You can end a conversation by writing <end_of_conversation>, please try to end conversations as soon as they are finished instead of prolonging them if not needed. But do not end them prematurely either. Here is the math problem: {{ problem }}. |
| **Student Prompt** |
| You will act as a student in a conversation with a teacher in training. You will need to act as much like a student as possible. If possible do not respond with overly long messages. The conversation with the teacher will be about this math problem: {{ problem }}. You may or may not know how to solve it already, let the teacher guide you to the correct understanding. You will be tested at the end and scored thus it is best if you collaborate with the teacher as it has more experience in math than you. |

abstract mathematical concepts.

When handling complex abstract problems, our method shows finer cognitive scaffolding construction ability. First, our model verifies students' basic concept understanding ("Recall its definition?"). It guides students to independently derive periodicity ("What about increasing by 2?"). It promotes students to abstract derivation results into mathematical properties ("What does this property tell us?"). This structured guidance cultivates students' mathematical reasoning ability and concept transfer ability. GPT-4o skips the concept verification stage. It directly tells students "We deduce $f(x+2) = f(x)$" and "period is 2". While expedient, this approach limits students' opportunities for independent mathematical discovery. It hinders the formation of a deep understanding. Second, our method requires active student participation at each reasoning node ("Which condition helps?" "How would you use it?"). This continuous cognitive activation ensures students remain in the leading position of problem-solving. GPT-4o tends to provide explicit procedural directions ("Let's use $f(x+1) = -f(x)$"). Students only need to execute the given steps. Finally, our method strengthens students' overall understanding of multi-step reasoning through comprehensive questions ("Can you now connect all results?"). It facilitates the transfer of the application to similar problems.

Through comparative analysis of these two cases, our method demonstrates stronger adaptive teaching ability. It reflects the effectiveness of the proposed reward function.

## V. CONCLUSION

In this paper, we propose UCO, a cognition-oriented multi-turn reinforcement learning method for adaptive teaching. To address the limitations in existing methods that lack true adaptive capability, we model the teaching process as continuous teacher-student interactions and design two synergistic reward functions. The Progress Reward quantifies cognitive progress through entropy reduction, evaluating whether students truly transition from uncertainty to understanding. The Scaffold Reward dynamically identifies each student's Zone of Proximal Development and encourages teaching strategies that maintain productive struggle within this zone. Extensive experiments on BigMath and MathTutorBench benchmarks demonstrate that our UCO model surpasses all models of equivalent scale and achieves performance comparable to advanced closed-source models. We argue that modeling the teaching process from the perspective of cognitive state evolution can be applied not only to mathematical tutoring, but also to other educational domains such as programming education and scientific reasoning. Therefore, in future work, we will explore extending UCO to broader educational scenarios.

## VI. AI-Generated Content Acknowledgement

This paper utilized LLMs in four distinct capacities:

- **As Core Research Objects:** We employed Qwen2.5-7B-Instruct as the teacher model and Qwen2.5-14B-Instruct as the student model throughout our adaptive teaching dialogue framework (Sections 3-4). These models were the primary experimental subjects for training, evaluation, and case studies.
- **As Auxiliary Tools:** We used Gemini-2.5-Pro-Exp-03-25 as an oracle model to generate candidate responses (Section 3). These tools operated under human supervision with prompts designed by the authors.
- **As Evaluation Baselines:** We systematically compared our method against 11 mainstream LLMs (Section 4.2).
- **During Manuscript Preparation:** LLMs were employed for minor language polishing in Sections 1-5, including grammar corrections, sentence structure improvements, and clarity enhancements. All ideas, methodologies, and conclusions are original contributions of the authors.

## References


[1] J. Macina, N. Daheim, S. Chowdhury, T. Sinha, M. Kapur, I. Gurevych, and M. Sachan, "Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 5602–5621.

[2] R. Wang, P. Wirawarn, K. Lam, O. Khattab, and D. Demszky, "Problem-oriented segmentation and retrieval: Case study on tutoring conversations," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 12 654–12 672.

[3] J. Liu, Z. Huang, T. Xiao, J. Sha, J. Wu, Q. Liu, S. Wang, and E. Chen, "Socraticlm: Exploring socratic personalized teaching with large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 85 693–85 721, 2024.

[4] F. M. Fahid, J. Rowe, Y. Kim, S. Srivastava, and J. Lester, "Online reinforcement learning-based pedagogical planning for narrative-centered learning environments," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 21, 2024, pp. 23 191–23 199.

[5] A. Riedmann, P. Schaper, and B. Lugrin, "Reinforcement learning in education: A systematic literature review," *International Journal of Artificial Intelligence in Education*, pp. 1–55, 2025.

[6] D. Dinucu-Jianu, J. Macina, N. Daheim, I. Hakimi, I. Gurevych, and M. Sachan, "From problem-solving to teaching problem-solving: Aligning llms with pedagogy using reinforcement learning," *arXiv preprint arXiv:2505.15607*, 2025.

[7] Z. Zhou, B. Hu, C. Zhao, P. Zhang, and B. Liu, "Large language model as a policy teacher for training reinforcement learning agents," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024, pp. 5671–5679.

[8] G. Poesia, D. Broman, N. Haber, and N. Goodman, "Learning formal mathematics from intrinsic motivation," *Advances in Neural Information Processing Systems*, vol. 37, pp. 43 032–43 057, 2024.

[9] Y. Sheng, L. Li, and D. D. Zeng, "Learning theorem rationale for improving the mathematical reasoning capability of large language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 14, 2025, pp. 15 151–15 159.

[10] W. Xiong, C. Shi, J. Shen, A. Rosenberg, Z. Qin, D. Calandriello, M. Khalman, R. Joshi, B. Piot, M. Saleh *et al.*, "Building math agents with multi-turn iterative preference learning," in *The Thirteenth International Conference on Learning Representations*.

[11] Y. Li, X. Shen, X. Yao, X. Ding, Y. Miao, R. Krishnan, and R. Padman, "Beyond single-turn: A survey on multi-turn interactions with large language models," *arXiv preprint arXiv:2504.04717*, 2025.

[12] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[13] O. Press, R. Shwartz-Ziv, Y. Lecun, and M. Bethge, "The entropy enigma: Success and failure of entropy minimization," in *International Conference on Machine Learning*. PMLR, 2024, pp. 41 064–41 085.

[14] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 209–216.

[15] K. Shabani, M. Khatib, and S. Ebadi, "Vygotsky's zone of proximal development: Instructional implications and teachers' professional development." *English language teaching*, vol. 3, no. 4, pp. 237–248, 2010.

[16] J. Van de Pol, M. Volman, and J. Beishuizen, "Scaffolding in teacher–student interaction: A decade of research," *Educational psychology review*, vol. 22, no. 3, pp. 271–296, 2010.

[17] J. Bliss, M. Askew, and S. Macrae, "Effective teaching and learning: Scaffolding revisited," *Oxford review of Education*, vol. 22, no. 1, pp. 37–61, 1996.

[18] B. Rosenshine, C. Meister *et al.*, "The use of scaffolds for teaching higher-level cognitive strategies," *Educational leadership*, vol. 49, no. 7, pp. 26–33, 1992.

[19] A. Tack and C. Piech, "The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues," in *Proceedings of the 15th International Conference on Educational Data Mining*, 2022, p. 522.

[20] A. Kucheria, N. Sawhney, and A. Hellas, "Comparing behavioral patterns of llm and human tutors: A population-level analysis with the cima dataset," in *Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2025.

[21] R. Puech, J. Macina, J. Chatain, M. Sachan, and M. Kapur, "Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure," *arXiv preprint arXiv:2410.03781*, 2024.

[22] J. Macina, N. Daheim, L. Wang, T. Sinha, M. Kapur, I. Gurevych, and M. Sachan, "Opportunities and challenges in neural dialog tutoring," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023, pp. 2357–2372.

[23] H. Luo, Q. Sun, C. Xu, P. Zhao, J.-G. Lou, C. Tao, X. Geng, Q. Lin, S. Chen, Y. Tang *et al.*, "Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct," in *The Thirteenth International Conference on Learning Representations*.

[24] H. Liu, Z. Zheng, Y. Qiao, H. Duan, Z. Fei, F. Zhou, W. Zhang, S. Zhang, D. Lin, and K. Chen, "Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 6884–6915.

[25] V. De Marco, F. Sciarrone, and M. Temperini, "Tutorchat: a chatbot for the support to dyslexic learner's activity through generative ai," in *2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*. IEEE, 2024, pp. 155–157.

[26] A. Albalak, D. Phung, N. Lile, R. Rafailov, K. Gandhi, L. Castricato, A. Singh, C. Blagden, V. Xiang, D. Mahan *et al.*, "Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models," *arXiv preprint arXiv:2502.17387*, 2025.

[27] J. Macina, N. Daheim, I. Hakimi, M. Kapur, I. Gurevych, and M. Sachan, "Mathtutorbench: A benchmark for measuring open-ended pedagogical capabilities of llm tutors," *arXiv preprint arXiv:2502.18940*, 2025.

[28] Z. Chu, S. Wang, J. Xie, T. Zhu, Y. Yan, J. Ye, A. Zhong, X. Hu, J. Liang, P. S. Yu *et al.*, "Llm agents for education: Advances and applications," *arXiv preprint arXiv:2503.11733*, 2025.

[29] M. Klissarov, R. D. Hjelm, A. T. Toshev, and B. Mazoure, "On the modeling capabilities of large language models for sequential decision making," in *The Thirteenth International Conference on Learning Representations*.

[30] J. Wang, J. Li, X. Han, D. Ye, and Z. Lu, "Language model adaption for reinforcement learning with natural language action space," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024, pp. 1620–1634.

[31] H. Peng, Y. Qi, X. Wang, Z. Yao, B. Xu, L. Hou, and J. Li, "Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems," *arXiv preprint arXiv:2502.19328*, 2025.

[32] M. Liao, C. Li, W. Luo, W. Jing, and K. Fan, "Mario: Math reasoning with code interpreter output-a reproducible pipeline," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 905–924.

[33] L. Zhong, Z. Wang, and J. Shang, "Debug like a human: A large language model debugger via verifying runtime execution step by step," in *Findings of the Association for Computational Linguistics ACL 2024*, 2024, pp. 851–870.

[34] Z. Zhang, C. Zheng, Y. Wu, B. Zhang, R. Lin, B. Yu, D. Liu, J. Zhou, and J. Lin, "The lessons of developing process reward models in mathematical reasoning," *arXiv preprint arXiv:2501.07301*, 2025.

[35] Z. Yin, Q. Sun, Z. Zeng, Q. Cheng, X. Qiu, and X.-J. Huang, "Dynamic and generalizable process reward modeling," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 4203–4233.

[36] W. Li and Y. Li, "Process reward model with q-value rankings," in *The Thirteenth International Conference on Learning Representations*.

[37] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation," *arXiv preprint arXiv:2402.03216*, 2024.

[38] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[39] L. Team, A. Modi, A. S. Veerubhotla, A. Rysbek, A. Huber, B. Wiltshire, B. Veprek, D. Gillick, D. Kasenberg, D. Ahmed *et al.*, "Learnlm: Improving gemini for learning," *arXiv preprint arXiv:2412.16429*, 2024.

[40] Q. Team, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, vol. 2, p. 3, 2024.

[41] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[42] G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, E. Rosen *et al.*, "Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities," *arXiv preprint arXiv:2507.06261*, 2025.