EDUMCTS: IMPROVING LARGE LANGUAGE MODEL REASONING THROUGH ADAPTIVE PEDAGOGICAL INTERACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality reasoning trajectories serve as critical post-training resources for enhancing the reasoning abilities of large language models (LLMs). However, existing reasoning data augmentation methods mainly focus on exploring diverse candidate solutions and selecting among reasoning trajectories, while overlooking how to actively improve intermediate reasoning during trajectory generation. To mitigate this limitation, we propose EduMCTS, a reasoning data augmentation framework that constructs reasoning trajectories by selectively introducing pedagogical interventions during problem solving. Specifically, EduMCTS formalizes pedagogical strategies as executable search actions for intermediate reasoning states, evaluates candidate reasoning paths using local reasoning progress and final answer correctness, and leverages tree search to construct verified interactive reasoning trajectories. Extensive experiments across three LLMs and eleven reasoning benchmarks, spanning seven mathematical and four scientific reasoning tasks, demonstrate that EduMCTS consistently outperforms existing data augmentation methods and generalizes well to out-of-domain scientific reasoning. Further analyses support the role of pedagogical strategy search and show that models trained with EduMCTS exhibit adaptive pedagogical reasoning behaviors during inference. Our code and dataset are anonymously available at `https://anonymous.4open.science/r/EduMCTS-B568/`.

1 INTRODUCTION

In recent years, high-quality reasoning trajectories have become important post-training resources for improving the reasoning abilities of large language models (LLMs) (Yue et al., 2024; Toshniwal et al., 2025). However, constructing such supervision through human annotation is costly, while directly fine-tuning on solutions from existing datasets does not necessarily improve reasoning performance (Rrv et al., 2025). As shown in Figure 1, supervised fine-tuning (SFT) on 1,080 original solutions from NuminaMath-1.5(LI et al., 2024) reduces the average performance of all three backbones across seven mathematical reasoning benchmarks. This observation further highlights the importance of constructing high-quality reasoning supervision.

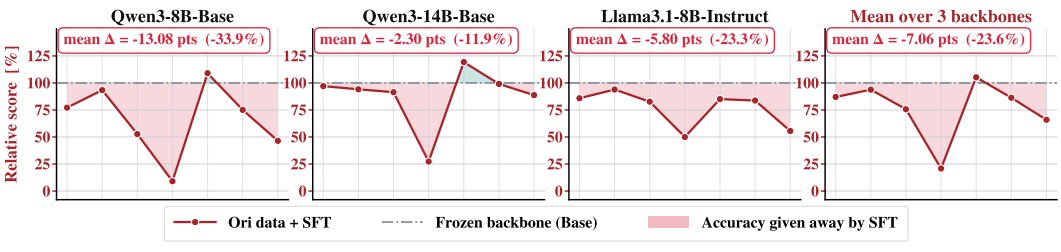


Figure 1: **Fine-tuning on original solutions can degrade reasoning performance.** Scores are normalized per benchmark to the corresponding frozen backbone (100%) on seven reasoning benchmarks. Red curves denote models fine-tuned on the original solutions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

To construct such supervision, prior work has used strong LLMs to generate step-by-step solutions or rewrite problems, and then filter candidate trajectories using correctness or quality signals (Mukherjee et al., 2023; Zelikman et al., 2022; Yu et al., 2024). Recent studies have further introduced structured search into data construction, expanding candidate branches from intermediate reasoning states and selecting among them using process or outcome rewards (Zhang et al., 2024; Xie et al., 2024; Guan et al., 2025). However, these methods mainly use process-level signals to evaluate candidate reasoning paths, rather than to **actively improve intermediate reasoning** during generation. How to incorporate such adaptive guidance into trajectory construction remains underexplored.

To mitigate this limitation, we propose **EduMCTS**, a reasoning data augmentation framework that constructs reasoning trajectories by selectively introducing pedagogical interventions during problem solving. EduMCTS organizes intermediate reasoning states into a search tree and formalizes **pedagogical strategies**, including HINT, CRITIQUE, and SOCRATIC, as executable search actions. EduMCTS scores candidate reasoning paths using **local reasoning progress and final answer correctness**, then uses **Monte Carlo Tree Search (MCTS)** to explore pedagogical actions and construct verified interactive reasoning trajectories for SFT.

Using EduMCTS-synthesized data, we fine-tune Qwen3-8B-Base, Qwen3-14B-Base, and Llama3.1-8B-Instruct, and evaluate them on eleven reasoning benchmarks spanning seven mathematical and four scientific reasoning tasks. EduMCTS consistently outperforms existing augmentation methods across model scales and families, and generalizes well to out-of-domain reasoning tasks. Ablation studies support the contribution of each key component, while behavioral analyses show that the trained models exhibit adaptive pedagogical reasoning behaviors during inference.

Our main contributions are summarized as follows:

- We propose EduMCTS, a reasoning data augmentation framework that incorporates adaptive pedagogical interaction into trajectory construction. EduMCTS treats pedagogical strategies as searchable actions at intermediate reasoning states and uses tree search to construct interactive reasoning trajectories for supervised fine-tuning.
- We evaluate EduMCTS across three LLMs and eleven reasoning benchmarks, spanning seven mathematical and four scientific reasoning tasks, and show that it consistently outperforms existing data augmentation baselines while generalizing to out-of-domain reasoning tasks.
- We analyze the main design choices and learned reasoning behaviors of EduMCTS, showing that the resulting models exhibit adaptive pedagogical reasoning behaviors during inference.

2 RELATED WORK

Synthetic Reasoning Data. Synthetic post-training data has been widely used to improve instruction following and reasoning capabilities (Gou et al., 2024; Xu et al., 2025). Early work expands instruction and problem distributions through methods such as Self-Instruct, WizardLM, and MetaMath (Wang et al., 2023b; Xu et al., 2024; Yu et al., 2024). Later studies focus more on reasoning supervision. Orca distills explanation traces from stronger models, STaR trains on correct self-generated rationales, MAmmoTH combines CoT and PoT supervision, and OpenMathInstruct-2 studies how teacher strength, solution format, and problem diversity affect mathematical SFT (Mukherjee et al., 2023; Zelikman et al., 2022; Yue et al., 2024; Toshniwal et al., 2025). V-STaR further trains a verifier using both correct and incorrect generations (Hosseini et al., 2024). Overall, these methods mainly construct training data by generating and filtering complete reasoning trajectories, with limited attention to finer-grained reasoning steps.

Search-Augmented Reasoning. Structured search has been widely used to explore multiple reasoning paths beyond single-path generation. Self-Consistency aggregates sampled chains, while Tree of Thoughts and Graph of Thoughts organize intermediate states for evaluation and backtracking (Wang et al., 2023a; Yao et al., 2023; Besta et al., 2024). RAP and LATS further combine tree search with model-based planning, feedback, or reflection (Hao et al., 2023; Zhou et al., 2024). Recent work extends search to training data construction: ReST-MCTS* collects high-quality trajectories with process values, MCTS-DPO derives step-level preferences from search trees, and rStar-Math uses MCTS with process rewards to generate verifiable reasoning data (Zhang et al., 2024; Xie et al., 2024; Guan et al., 2025). Overall, these methods expand the search space at the

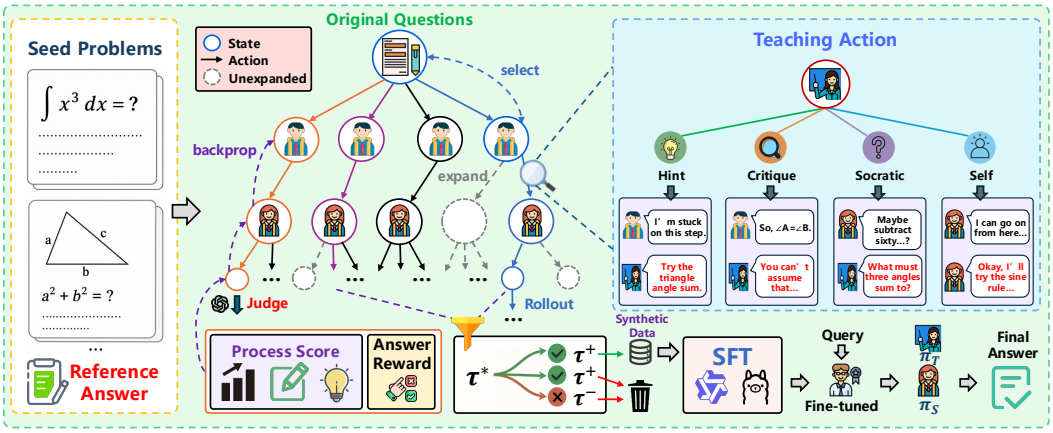


Figure 2: **Overview of EduMCTS.** EduMCTS searches over pedagogical actions at intermediate reasoning states with MCTS, evaluates their induced reasoning using process and outcome signals, and converts verified interaction trajectories into supervision for downstream fine-tuning.

step level, but still focus on selecting among alternative reasoning continuations rather than actively improving intermediate reasoning during generation.

Simulated Pedagogical Interaction. Teacher-student interaction provides adaptive supervision beyond static solution distillation. MathDial, Book2Dial, and Tutor CoPilot explore pedagogical interaction through tutoring dialogues, teacher-student simulation, and contextual teaching support (Macina et al., 2023; Wang et al., 2024a;b). Recent work further models adaptive teaching through classroom simulation, learner states, and pedagogical planning (Zhang et al., 2025; Sanyal et al., 2025; Li et al., 2026). MASTER shows that simulated teaching interactions can also be converted into SFT data to improve reasoning (Yue et al., 2025). These studies motivate us to treat pedagogical strategies as searchable actions and compare their effects on subsequent reasoning.

3 EduMCTS

Given a seed problem and its reference answer $(x, y^\star) \in \mathcal{D}_{\text{seed}}$, EduMCTS uses MCTS to construct reasoning trajectories through **pedagogical interaction**. As shown in Figure 2, EduMCTS searches over **pedagogical actions** at intermediate reasoning states, evaluates the resulting branches using **local reasoning progress and final answer correctness**, and retains verified high-value trajectories as supervision for downstream fine-tuning.

We use the same frozen language model M_ϕ with different role prompts to instantiate a teacher T_ϕ, a student S_ϕ, a progress evaluator E_ϕ, and a terminal verifier V_ϕ. T_ϕ provides pedagogical feedback, S_ϕ generates reasoning steps, and E_ϕ and V_ϕ evaluate local progress and final solution correctness, respectively. The prompts used to instantiate these four roles are provided in Appendix A.1.

The reference answer $y^\star$ is available only to the evaluators and is never exposed to the teacher or student. For a problem x, the student first generates an initial reasoning step u_0, forming the root history $H_0 = [u_0]$. Each node v represents a reasoning state $s(v) = (x, H(v))$, where $H(v)$ contains all student reasoning and teacher feedback along the path to v. Each edge represents a pedagogical action and the student response it induces. EduMCTS explores these transitions under a fixed search budget and retains a terminally verified trajectory as augmented supervision. The complete EduMCTS data augmentation procedure is summarized in the pseudocode in Appendix A.2, Table 5.

3.1 Pedagogical Actions and State Transitions

Pedagogical action space. Prior educational research supports instructional hints, corrective feedback, and Socratic questioning as effective forms of pedagogical guidance (Pardos & Bhandari, 2024; Wisniewski et al., 2020; Nuraeni et al., 2026). Inspired by these studies, we define Hint,

CRITIQUE, and SOCRATIC as three pedagogical actions, together with autonomous SELF, forming the discrete action space:

$$\mathcal{A} = \{\textsc{Self}, \textsc{Hint}, \textsc{Critique}, \textsc{Socratic}\}. \tag{1}$$

SELF lets the student continue without teacher feedback. HINT provides directional guidance, CRITIQUE identifies a concrete error or questionable assumption, and SOCRATIC asks a local question that prompts the student to derive the next step. Teacher feedback is restricted to the current reasoning state and does not reveal the final answer or a complete solution. Including SELF allows the search to decide both whether to intervene and which intervention to use.

Feedback conditioned state transition. Let H_{t-1} denote the reasoning history before step t. Given a selected action $a_t \in \mathcal{A}$, the teacher generates feedback m_t when intervention is required, and the student then produces one local reasoning step u_t from the updated context. For SELF, we set $m_t = \varnothing$ and omit the teacher call:

$$\begin{aligned} m_t &\sim \begin{cases} T_\phi(\cdot \mid x, H_{t-1}, a_t), & a_t \neq \textsc{Self}, \\ \varnothing, & a_t = \textsc{Self}, \end{cases} \\ u_t &\sim S_\phi^{\text{step}}(\cdot \mid x, H_{t-1} \oplus m_t), \\ H_t &= H_{t-1} \oplus m_t \oplus u_t, \end{aligned} \tag{2}$$

where $\oplus$ denotes chronological concatenation. Each student response is limited to one local derivation, calculation, or verification, so each edge represents a distinct reasoning update. At each node, every action is instantiated at most once, and the resulting transition is cached for later visits. This yields a finite search space over pedagogical actions while generating feedback and student responses from the current reasoning state.

3.2 REASONING PROGRESS AND TERMINAL EVALUATION

A pedagogical action may improve local reasoning without leading to a correct solution, or appear ineffective locally while still supporting eventual success. EduMCTS therefore evaluates each branch using both local reasoning progress and final answer correctness.

Local reasoning progress. For a transition (H_{t-1}, a_t, m_t, u_t), the progress evaluator E_ϕ scores the reasoning change introduced by the new step:

$$p_t = E_\phi(x, y^\star, H_{t-1}, a_t, m_t, u_t) \in \{-2, -1, 0, 1, 2\}, \qquad r_t = \frac{p_t}{2}. \tag{3}$$

Positive scores indicate meaningful progress or error correction, zero indicates no substantive change, and negative scores indicate degraded reasoning, new errors, or feedback that reveals the solution. SELF is evaluated under the same criterion as the intervention actions.

Combined process and outcome return. Local progress alone does not determine final success. At the bth search iteration, suppose the search reaches node v_b. The student completes the remaining reasoning autonomously:

$$c_b \sim S_\phi^{\text{roll}}(\cdot \mid x, H(v_b)), \tag{4}$$

which forms the complete trajectory $\tau_b = H(v_b) \oplus c_b$. The terminal verifier then evaluates its correctness:

$$R_b = V_\phi(x, y^\star, \tau_b) \in \{0, 1\}. \tag{5}$$

If v_b already has a verified correct completion, we reuse the cached result. Let $\mathcal{P}(v)$ denote the edges from the root to node v, and let $d(v) = |\mathcal{P}(v)|$ be its depth. We define the average progress along the path as

$$\bar{r}(v) = \begin{cases} \dfrac{1}{d(v)} \displaystyle\sum_{e \in \mathcal{P}(v)} r(e), & d(v) > 0, \\ 0, & d(v) = 0. \end{cases} \tag{6}$$

The simulation return is

$$G_b = R_b + \lambda \bar{r}(v_b), \tag{7}$$

where λ controls the contribution of process evaluation. Averaging progress avoids favoring deeper paths simply because they contain more steps. Since $\bar{r}(v) \in [-1, 1]$, setting $0 \leq \lambda < 1/2$ ensures any correct trajectory receives a larger return than any incorrect one, while process scores still distinguish branches with the same terminal outcome. We use $\lambda = 0.25$ in all experiments.

3.3 Monte Carlo Tree Search over Pedagogical Actions

EduMCTS performs selection, expansion, simulation, and backup under a fixed budget B, using the return in Eq. equation 7. Each node v stores a visit count $N(v)$ and cumulative return $W(v)$, with empirical value $Q(v) = W(v)/N(v)$ defined after the node has been visited.

Selection and expansion. Starting from the root, EduMCTS applies UCT at each fully expanded nonterminal node:

$$v' = \arg \max_{w \in \mathrm{Ch}(v)} \left[Q(w) + \kappa \sqrt{\frac{\log(N(v)+1)}{N(w)}} \right], \tag{8}$$

where $\mathrm{Ch}(v)$ denotes the children of v, and κ controls exploration. Unvisited children receive the highest exploration priority.

Selection stops at an expandable node, a successful terminal node, or the maximum search depth. For an expandable node, EduMCTS samples one untried pedagogical action, generates the resulting student state with Eq. equation 2, and scores its local progress using Eq. equation 3. No prior preference is assigned to the pedagogical actions; search effort is allocated according to the reasoning outcomes observed after each action.

Simulation and backup. After expansion, the student completes the remaining reasoning autonomously, and EduMCTS computes the simulation return G_b using Eq. equation 7. The return is propagated to the simulation node and all its ancestors:

$$N(v) \leftarrow N(v) + 1, \qquad W(v) \leftarrow W(v) + G_b, \qquad v \in \mathrm{Anc}^+(v_b), \tag{9}$$

where $\mathrm{Anc}^+(v_b)$ includes v_b. Repeated simulations update $Q(v)$ with the observed final outcomes and reasoning progress, which then guide subsequent UCT selection.

Termination. A node becomes terminal when the student produces a final answer that passes verification. If verification fails, the node remains expandable so that later pedagogical actions can revise the reasoning. A maximum depth D limits the number of student reasoning steps. Nodes at depth D are not expanded further, but one autonomous completion is still used to estimate the terminal outcome. If the initial autonomous reasoning is already correct, EduMCTS retains the trajectory directly and skips tree search. The budget B counts MCTS iterations, not LLM calls. EduMCTS caches generated transitions and verified correct completions to avoid redundant inference.

3.4 Trajectory Extraction and Supervised Fine-Tuning

After search, EduMCTS extracts one training trajectory from the multi-branch tree. Instead of choosing the best single simulation, we use the visit statistics accumulated during search to identify the main path.

Trajectory extraction and filtering. Starting from the root, we repeatedly select the visited child with the largest visit count, breaking ties by the empirical value Q:

$$v^\star_{t+1} = \operatorname*{lexargmax}_{w \in \mathrm{Ch}_+(v^\star_t)} \big(N(w), Q(w) \big), \tag{10}$$

where $\mathrm{Ch}_+(v)$ denotes the children visited at least once. Extraction stops at a successful terminal node, the maximum depth, or a node with no visited child.

Each node caches any verified correct completion found from that state. At the end of the selected path, EduMCTS reuses a cached completion when available; otherwise, the student performs one

autonomous completion. If it still fails verification, we use a cached successful completion from another branch, prioritizing larger visit counts and empirical values. Each retained problem therefore produces one complete trajectory $\tau_i^\star$ that passes terminal verification.

We further compute the empirical success rate

$$\hat{\rho}_i = \frac{1}{B} \sum_{b=1}^{B} R_{i,b}, \tag{11}$$

and retain only samples with $\hat{\rho}_i \geq \eta$. We use $\eta = 0.10$ by default.

Interaction trajectory serialization. For each retained trajectory, we preserve the student reasoning and place teacher feedback with its action type before the corresponding student response. Let $\mathrm{Reflect}_{a_t}(m_t)$ denote feedback annotated by action type, with $\mathrm{Reflect}_{\textsc{Self}}(\varnothing) = \varnothing$. The supervision target is

$$z_i = u_0 \oplus \bigoplus_{t=1}^{L_i} [\mathrm{Reflect}_{a_t}(m_t) \oplus u_t] \oplus c_i, \tag{12}$$

where L_i is the number of edges on the selected path and c_i is the autonomous completion from its final node. If the solution is completed within the search tree, then $c_i = \varnothing$. This serialization preserves each pedagogical intervention together with the student reasoning that follows it.

Supervised fine-tuning. The retained samples form the augmented dataset

$$\mathcal{D}_{\mathrm{syn}} = \{(x_i, z_i)\}_{i \in \mathcal{I}}, \tag{13}$$

where $\mathcal{I}$ indexes the retained examples. We use the problem with a fixed solving instruction as input and z_i as the target. The downstream model f_θ is trained with the standard autoregressive objective

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{k=1}^{|z_i|} \log f_\theta \left(z_{i,k} \mid x_i, z_{i,<k} \right). \tag{14}$$

Tree search is used only to construct and filter supervision trajectories; the downstream parameters θ are optimized entirely through supervised fine-tuning.

4 Experiments

We evaluate EduMCTS across three downstream LLMs on mathematical reasoning and cross-domain scientific reasoning, followed by controlled analyses of data composition, key components, and learned reasoning behaviors.

4.1 Experimental Setup

Data and synthesis. We sample 1,800 question-answer pairs from the competition-level split of NuminaMath-1.5(LI et al., 2024). After filtering invalid or incomplete entries, 1,331 problems remain for synthesis, from which EduMCTS produces 1,080 terminally verified training trajectories. We use a search budget of $B = 20$, maximum depth $D = 4$, UCT coefficient $\kappa = 1.4$, and process weight $\lambda = 0.25$. Data filtering and generation details are provided in Appendix B.1.

Baselines. We compare EduMCTS with six data construction baselines: Ori, DART, MetaMath, CoT, MASTER, and CRPS(Tong et al., 2024; Yu et al., 2024; Kim et al., 2023; Yue et al., 2025; Liu et al., 2026). Ori uses the original solutions, while DART, MetaMath, and CoT represent common reasoning data augmentation strategies. MASTER introduces simulated pedagogical interaction, and CRPS provides a recent MCTS-based reasoning data synthesis baseline. For methods requiring generation, we use qwen3-235b-a22b-instruct-2507 to keep the synthesis model consistent. Implementation details for all baselines are provided in Appendix B.2.

Training setup. We fine-tune Qwen3-8B-Base, Qwen3-14B-Base, and Llama3.1-8B-Instruct on the data produced by each method using the same LoRA SFT setup (Team, 2025; Grattafiori et al.,

Table 1: Reasoning accuracy (%) across three backbone models. Per-benchmark scores are averaged over two independent runs. Bold and underline indicate the best and second-best mean within each backbone, respectively, including ties. Full results are provided in Appendix C.1.

Method	MATH-500	GSM8K	AMC	AIME	MMLU-Pro	AGIEval	Olympiad	ARC-C	GPQA	MMLU-STEM	MMLU-Pro (Sci.)	Avg.
Qwen3-8B-Base												
w/o SFT	77.60	92.31	42.91	11.11	55.70	73.65	**44.52**	71.33	31.56	73.69	50.16	56.78 ± 0.53
Ori	59.30	85.86	21.40	0.56	58.33	55.55	20.70	82.51	36.37	75.75	53.64	49.99 ± 0.28
Dart	62.70	88.29	30.97	1.67	69.21	62.40	39.39	93.69	32.83	84.83	62.29	57.11 ± 0.71
MetaMath	74.80	90.34	32.09	12.22	71.83	71.00	17.88	92.11	34.09	84.84	60.10	58.30 ± 0.19
CoT	75.60	91.85	42.44	12.78	73.06	73.30	34.65	92.92	32.32	85.34	60.93	61.38 ± 0.56
MASTER	72.70	91.93	36.94	13.89	78.94	72.25	30.71	90.15	42.17	85.29	**64.67**	61.78 ± 0.23
CRPS	78.00	93.14	42.16	15.56	**79.61**	75.85	43.03	93.35	40.15	85.29	63.39	64.50 ± 0.07
EduMCTS	**82.20**	**94.13**	**48.13**	**20.56**	78.50	**78.32**	42.81	**94.67**	**43.18**	**86.09**	64.54	$\mathbf{66.65 \pm 0.19}$
Qwen3-14B-Base												
w/o SFT	80.30	93.82	51.87	12.22	57.18	75.80	46.44	93.43	40.15	84.15	43.67	61.73 ± 0.08
Ori	79.00	88.63	45.90	3.89	68.65	74.65	40.06	93.13	45.96	87.25	64.00	62.83 ± 0.31
Dart	82.30	90.64	46.64	2.78	78.20	78.80	50.00	94.36	45.71	87.96	66.96	65.85 ± 0.26
MetaMath	78.00	**95.72**	42.91	12.78	81.09	75.90	43.25	94.26	39.14	87.41	65.71	65.11 ± 0.09
CoT	80.40	95.49	47.01	13.89	79.46	77.65	42.95	94.71	43.18	88.21	69.50	66.59 ± 0.37
MASTER	82.70	93.67	49.63	16.12	78.43	77.45	42.14	92.58	38.13	89.00	70.05	66.35 ± 0.31
CRPS	83.10	95.11	51.12	17.23	84.05	**81.35**	49.93	94.93	50.00	**89.50**	70.75	69.73 ± 0.04
EduMCTS	**86.00**	95.15	**52.24**	**18.34**	**84.42**	**81.35**	**51.93**	**95.56**	**52.53**	89.42	**71.44**	$\mathbf{70.76 \pm 0.07}$
Llama3.1-8B-Instruct												
w/o SFT	49.50	86.28	17.16	**2.22**	50.19	50.35	18.55	87.97	26.77	69.09	42.70	45.52 ± 0.17
Ori	43.10	80.78	13.81	0.56	42.71	42.95	10.69	87.29	**30.30**	67.95	39.30	41.77 ± 0.05
Dart	49.30	82.87	14.93	1.67	42.42	45.15	11.80	85.84	25.51	65.91	37.80	42.11 ± 0.03
MetaMath	51.20	84.08	15.68	0.56	47.93	50.25	17.29	86.86	23.74	66.85	39.71	44.01 ± 0.40
CoT	51.60	83.51	16.05	**2.22**	50.19	50.40	15.28	86.44	19.70	67.10	39.71	43.84 ± 0.10
MASTER	51.80	85.03	18.28	**2.22**	50.67	48.85	17.07	87.55	23.49	70.32	41.94	45.20 ± 0.21
CRPS	52.60	85.07	17.54	**2.22**	49.08	50.70	18.55	87.46	23.49	69.33	42.50	45.32 ± 0.01
EduMCTS	**56.20**	**86.66**	**18.66**	2.22	**51.18**	**51.00**	**19.44**	**88.57**	25.25	**71.76**	**44.40**	$\mathbf{46.85 \pm 0.23}$

2024). The corresponding pretrained or instruction-tuned model without additional SFT is denoted as w/o SFT. Full training configurations are provided in Appendix B.2.

Evaluation. We use EvalScope (Team, 2024) for all evaluations. Mathematical reasoning is evaluated on seven benchmarks: MATH-500 (Hendrycks et al., 2021b), GSM8K (Cobbe et al., 2021), AMC, AIME, OlympiadBench (He et al., 2024), and the mathematics subsets of MMLU-Pro (Wang et al., 2024c) and AGIEval (Zhong et al., 2024). For cross-domain generalization, we evaluate all three backbones on ARC-C (Clark et al., 2018), GPQA-Diamond (Rein et al., 2024), MMLU-STEM (Hendrycks et al., 2021a), and the science subset of MMLU-Pro. We use greedy decoding with temperature 0 and a maximum generation length of 2,048 tokens, and report per-benchmark accuracy and the mean. Further details are provided in Appendix B.3.

4.2 Main Results

Overall performance. As shown in Table 1, EduMCTS consistently achieves the best overall performance across all three backbones and eleven reasoning benchmarks. Using only 1,080 synthesized trajectories for LoRA fine-tuning, it outperforms diverse augmentation baselines, including MASTER, which augments the same amount of original data, the data-expanding MetaMath, and the recent MCTS-based CRPS. This suggests that searching over pedagogical actions provides more effective reasoning supervision than trajectory generation or search alone.

The gains also extend to four out-of-domain scientific reasoning benchmarks, demonstrating cross-domain generalization. EduMCTS further improves Llama3.1-8B-Instruct despite its existing instruction tuning, showing that the framework remains effective beyond base models.

4.3 Effect of Augmentation Ratio

We vary the EduMCTS proportion $\alpha \in \{0, 0.25, 0.50, 0.75, 1\}$ on Qwen3-8B-Base, with the remaining training data drawn from Ori. As shown in Table 2, the average accuracy across eleven benchmarks increases monotonically with the EduMCTS share, showing a strong positive Pearson correlation ($r = 0.971$, $p = 0.006$). Increasing the share from 0% to 100% improves the overall average by 16.66 percentage points. The gains are also observed in both mathematical and scientific reasoning, although individual benchmarks do not always improve monotonically. Even 25% EduMCTS data clearly outperforms Ori, while full EduMCTS achieves the best overall performance.

Table 2: Effect of the EduMCTS data share on Qwen3-8B-Base. Scores are averaged over two runs; Avg. reports mean ± standard deviation across runs. Full results are provided in Appendix C.1.

EduMCTS share	MATH-500	GSM8K	AMC	AIME	MMLU-Pro	AGIEval	Olympiad	ARC-C	GPQA	MMLU-STEM	MMLU-Pro (Sci.)	Avg.
Qwen3-8B-Base												
0% (Ori)	59.30	85.86	21.40	0.56	58.33	55.55	20.70	82.51	36.37	75.75	53.64	49.99 ± 0.28
25%	40.90	87.61	35.45	13.33	73.06	66.20	41.39	90.52	40.66	79.54	63.35	57.45 ± 0.31
50%	53.00	89.58	40.55	17.78	76.91	72.55	42.95	92.32	41.41	84.20	63.32	61.33 ± 0.63
75%	65.50	91.51	47.76	17.23	78.28	76.45	**43.85**	93.05	**45.45**	85.73	64.23	64.46 ± 0.34
100% (EduMCTS)	**82.20**	**94.13**	**48.13**	**20.56**	**78.50**	**78.32**	42.81	**94.67**	43.18	**86.09**	**64.54**	$\mathbf{66.65 \pm 0.19}$

Table 3: Component ablations on Qwen3-8B-Base. Scores are averaged over two runs; Avg. reports mean ± standard deviation across runs. Full results are provided in Appendix C.1.

Method	MATH-500	GSM8K	AMC	AIME	MMLU-Pro	AGIEval	Olympiad	ARC-C	GPQA	MMLU-STEM	MMLU-Pro (Sci.)	Avg.
Qwen3-8B-Base												
w/o Teach	79.70	92.76	47.76	15.56	77.50	76.85	**43.77**	93.69	39.39	85.59	63.15	65.06 ± 0.37
w/o Search	80.30	92.76	45.90	11.67	**79.28**	77.10	43.10	93.60	41.67	85.69	**64.56**	65.06 ± 0.54
w/o Process Reward	80.30	92.27	45.90	16.12	78.39	77.10	42.58	93.56	39.39	85.51	64.31	65.04 ± 0.15
EduMCTS	**82.20**	**94.13**	**48.13**	**20.56**	78.50	**78.32**	42.81	**94.67**	**43.18**	**86.09**	64.54	$\mathbf{66.65 \pm 0.19}$

These results show that increasing pedagogical interaction supervision consistently strengthens overall reasoning performance.

4.4 Ablation Study

We ablate pedagogical intervention, tree search, and process reward on Qwen3-8B-Base, starting from the same seed problems and using the same LoRA setup; implementation and data-construction details are provided in Appendix B.2. As shown in Table 3, removing any component reduces the average performance across eleven benchmarks. The three ablations yield similar average scores of about 65.0, approximately 1.6 points below the full EduMCTS model. Removing pedagogical intervention degrades performance despite retaining the same branching process, suggesting that the gains cannot be explained by additional reasoning branches alone. Removing tree search similarly reduces performance, supporting the value of structured exploration and comparison among candidate reasoning paths. Finally, removing the process reward while retaining terminal correctness rewards also leads to a comparable drop, indicating that local reasoning progress provides useful supervision beyond final answer correctness.

4.5 Further Analysis

Gains are not explained by longer reasoning traces. To examine whether the gains of EduMCTS simply arise from longer reasoning traces, we sample 100 examples from the synthetic data produced by EduMCTS and each augmentation baseline, and collect the corresponding MATH-500 inference traces from the resulting fine-tuned models. Using a unified Data-Juicer protocol (Chen et al., 2024; 2025), we measure four surface statistics: token count, character count, dependency edges, and verb count. As shown in Figure 3, EduMCTS is not the largest method on either the training or inference side of these statistics, suggesting that its gains cannot be explained simply by longer or more elaborate reasoning traces.

Pedagogical behaviors emerge with problem difficulty. We further analyze 500 model-generated trajectories on MATH-500 across five difficulty levels. We distinguish explicit reflections that use the reflection tags seen during training from implicit HINT, SOCRATIC, and CRITIQUE behaviors expressed directly in natural language. GPT-4.1 serves as an independent judge for implicit behaviors, with detailed counts and judging prompts provided in Appendix B.3. As shown in Figure 4, implicit pedagogical behavior coverage increases from 39.5% at Level 1 to 82.8% at Level 5, with all three action types showing an overall upward trend with difficulty. In contrast, explicit reflections remain sparse across difficulty levels, suggesting that the model does more than reproduce the reflection format seen during training and increasingly invokes learned pedagogical strategies when harder reasoning is required.

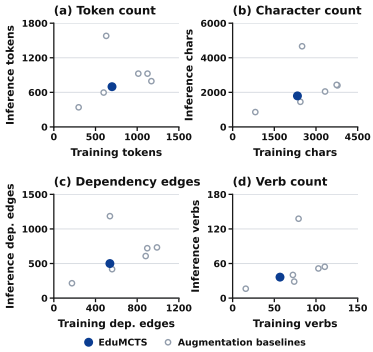


Figure 3: Training–inference surface statistics. Each point denotes one method.

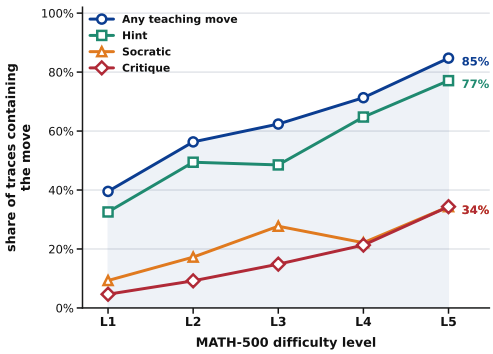


Figure 4: Prevalence of pedagogical behaviors across MATH-500 difficulty levels.

Table 4: **Examples of learned pedagogical reasoning.** EduMCTS exhibits distinct pedagogical behaviors during inference.

Component	Content
Case 1: Critique redirects ineffective reasoning	
State	Attempts $\frac{1}{F_n F_{n+2}} = \frac{A}{F_n} + \frac{B}{F_{n+2}}$.
CRITIQUE	The decomposition is invalid and the model should instead seek a telescoping difference.
Effect	The model later derives $\frac{F_{n+1}}{F_n F_{n+2}} = \frac{1}{F_n} - \frac{1}{F_{n+2}}$ and obtains $\boxed{2}$.
Case 2: Hint simplifies the next step	
State	$p(x) = a(x - (4 - \sqrt{11}))(x - (4 + \sqrt{11}))$.
HINT	Expand the two factors and simplify the expression.
Effect	The model obtains $p(x) = a(x^2 - 8x + 5)$.
Case 3: Socratic reflection focuses the goal	
State	$p(x) = a(x^2 - 8x + 5)$ with target $\frac{p(3)}{p(4)}$.
SOCRATIC	How do $p(3)$ and $p(4)$ depend on a, and what does this imply for their ratio?
Effect	Model finds $p(3) = -10a$ and $p(4) = -11a$, so the common factor cancels and yields $\boxed{10/11}$.

Representative cases. To provide an intuitive view of how EduMCTS influences reasoning during inference, we present two complete trajectories in Appendix C.2 and summarize three representative excerpts in Table 4. The table focuses on the pedagogical actions triggered at critical intermediate reasoning states. As shown in Case 1, when the model encounters an invalid decomposition, it invokes a learned `<reflection type="critique">` action to identify the issue and redirect the reasoning toward a telescoping strategy. In Cases 2 and 3, the model similarly activates `<reflection type="hint">` to provide local guidance and `<reflection type="socratic">` to elicit the key relation needed for the next step. These examples illustrate how EduMCTS learns to invoke different pedagogical actions based on the current reasoning state.

5 CONCLUSION

We introduced **EduMCTS**, a reasoning data augmentation framework that uses Monte Carlo Tree Search to explore pedagogical strategies at intermediate reasoning states. EduMCTS evaluates pedagogical paths via local reasoning progress and final answer correctness, and converts successful interactive trajectories into supervised fine-tuning data. Experiments across three downstream models and eleven reasoning benchmarks spanning seven mathematical and four scientific reasoning tasks show consistent improvements over existing augmentation methods and strong out-of-domain generalization. Further analyses validate the effectiveness of the proposed design and provide additional evidence for the benefits of pedagogically guided reasoning supervision. These results highlight searching over how to guide reasoning as a promising direction for reasoning data synthesis.

AI USE STATEMENT

In this work, we used generative AI tools for generating synthetic training data as part of the proposed EduMCTS pipeline. Additionally, we used generative AI tools to assist with manuscript writing and polishing. All AI-assisted content was reviewed by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects or personally identifiable information. Our experiments use publicly available research datasets and synthetic reasoning trajectories generated from them. As with other LLM-generated data, the synthesized trajectories may inherit errors or biases from the underlying language models; we therefore apply explicit verification and filtering before using them for training. The proposed method is intended for research on reasoning and data augmentation, and we do not claim its suitability for high-stakes applications.

REPRODUCIBILITY STATEMENT

We provide the implementation, synthesized training data, and experiment configurations in an anonymous repository at `https://anonymous.4open.science/r/EduMCTS-B568/`. For redundancy, the same materials are also included as a compressed archive in the Supplementary Material in case the anonymous repository becomes temporarily unavailable. The complete EduMCTS procedure and pseudocode are provided in Appendix A.2, including Table 5. Prompts for all model roles are given in Appendix A.1, data construction and filtering details in Appendix B.1, and benchmark configurations, prompts, and grading protocols in Appendix B.3. These materials specify the model, search, training, and evaluation settings used to reproduce our experiments.

REFERENCES

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In Proceedings of the AAAI conference on artificial intelligence, volume 38, pp. 17682–17690, 2024.

Daoyuan Chen, Yilun Huang, Zhijian Ma, Hesen Chen, Xuchen Pan, Ce Ge, Dawei Gao, Yuexiang Xie, Zhaoyang Liu, Jinyang Gao, Yaliang Li, Bolin Ding, and Jingren Zhou. Data-juicer: A one-stop data processing system for large language models. In SIGMOD, 2024.

Daoyuan Chen, Yilun Huang, Xuchen Pan, Jiang Nana, Haibin Wang, Yilei Zhang, Ce Ge, Yushuo Chen, Wenhao Zhang, Zhijian Ma, Jun Huang, Wei Lin, Yaliang Li, Bolin Ding, and Jingren Zhou. Data-juicer 2.0: Cloud-scale adaptive data processing for and with foundation models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), Advances in Neural Information Processing Systems, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-2768. URL `https://proceedings.neurips.cc/paper_files/paper/2025/file/76dbd7e897be2eb883ede598b91270fb-Paper-Datasets_and_Benchmarks_Track.pdf`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL `https://arxiv.org/abs/1803.05457`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and

Y. Sun (eds.), International Conference on Learning Representations, volume 2024, pp. 48362–48395, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d3cf1559a8795eb1ed2b3ad52409ac7d-Paper-Conference.pdf.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-math: Small LLMs can master math reasoning with self-evolved deep thinking. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), Proceedings of the 42nd International Conference on Machine Learning, volume 267 of Proceedings of Machine Learning Research, pp. 20640–20661. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/guan25f.html.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In International Conference on Learning Representations, 2021a. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), 2021b. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners, 2024. URL https://arxiv.org/abs/2402.06457.

Seungone Kim, Se Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The CoT collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 12685–12708, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.782. URL https://aclanthology.org/2023.emnlp-main.782/.

Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/datasets/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Zechen Li, Qiannan Zhu, Mei Wang, Jia Li, and Hua Huang. Planning-guided tutoring with assessment-driven memory for pedagogical LLM tutors. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 7165–7188, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.325. URL https://aclanthology.org/2026.acl-long.325/.

Peiyang Liu, Zhirui Chen, Xi Wang, Di Liang, Youru Li, Zhi Cai, and Wei Ye. Learning from contrasts: Synthesizing reasoning paths from diverse search trajectories. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 10947–10969, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.501. URL https://aclanthology.org/2026.acl-long.501/.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 5602–5621, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.372. URL https://aclanthology.org/2023.findings-emnlp.372/.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023. URL https://arxiv.org/abs/2306.02707.

Reni Nuraeni, Rizky Rosjanuardi, Al Jupri, and Elah Nurlaelah. The socratic method in mathematics learning: a systematic literature review. Frontiers in Education, Volume 11 - 2026, 2026. ISSN 2504-284X. doi: 10.3389/feduc.2026.1831645.

Zachary A. Pardos and Shreya Bhandari. Chatgpt-generated help produces learning gains equivalent to human tutor-authored help on mathematics skills. PLOS ONE, 19(5):1–18, 05 2024. doi: 10.1371/journal.pone.0304013. URL https://doi.org/10.1371/journal.pone.0304013.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In First Conference on Language Modeling, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Aswin Rrv, Jacob Dineen, Divij Handa, Md Nayem Uddin, Mihir Parmar, Chitta Baral, and Ben Zhou. ThinkTuning: Instilling cognitive reflections without distillation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 31248–31262, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1592. URL https://aclanthology.org/2025.emnlp-main.1592/.

Debdeep Sanyal, Agniva Maiti, Umakanta Maharana, Dhruv Kumar, Ankur Mali, C. Lee Giles, and Murari Mandal. Investigating pedagogical teacher and student LLM agents: Genetic adaptation meets retrieval-augmented generation across learning styles. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pp. 13348–13389, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.675. URL https://aclanthology.org/2025.emnlp-main.675/.

ModelScope Team. EvalScope: Evaluation framework for large models, 2024. URL https://github.com/modelscope/evalscope.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 7821–7846. Curran Associates, Inc., 2024. doi: 10.52202/079017-0251. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/0ef1afa0daa888d695dcd5e9513bafa3-Paper-Conference.pdf.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. Openmathinstruct-2: Accelerating AI for math with massive open-source instruction

data. In The Thirteenth International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=mTCbq2QssD.

Junling Wang, Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, and Mrinmaya Sachan. Book2Dial: Generating teacher student interactions from textbooks for cost-effective development of educational chatbots. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), Findings of the Association for Computational Linguistics: ACL 2024, pp. 9707–9731, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.578. URL https://aclanthology.org/2024.findings-acl.578/.

Rose E Wang, Ana T Ribeiro, Carly D Robinson, Susanna Loeb, and Dora Demszky. Tutor copilot: A human-ai approach for scaling real-time expertise. arXiv preprint arXiv:2410.03017, 2024b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations, 2023a. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 13484–13508, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024c. URL https://openreview.net/forum?id=y10DM6R2r3.

Benedikt Wisniewski, Klaus Zierer, and John Hattie. The power of feedback revisited: A meta-analysis of educational feedback research. Frontiers in Psychology, Volume 10 - 2019, 2020. ISSN 1664-1078. doi: 10.3389/fpsyg.2019.03087.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. In The First Workshop on System-2 Reasoning at Scale, NeurIPS'24, 2024. URL https://openreview.net/forum?id=s004OmYP2P.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. WizardLM: Empowering large pre-trained language models to follow complex instructions. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=CfXh93NDgH.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), International Conference on Learning Representations, volume 2025, pp. 76346–76382, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/be06e3802e9411381feece79b4d960c1-Paper-Conference.pdf.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), Advances in Neural Information Processing Systems, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions

for large language models. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Liang Yue, Yihong Tang, Kehai Chen, Jie Liu, and Min Zhang. MASTER: Enhancing large language model via multi-agent simulated teaching. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2025. URL https://openreview.net/forum?id=5GaDcRVgBw.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MAmmoTH: Building math generalist models through hybrid instruction tuning. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=yLClGs770I.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts$*$: Llm self-training via process reward guided tree search. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), Advances in Neural Information Processing Systems, volume 37, pp. 64735–64772. Curran Associates, Inc., 2024. doi: 10.52202/079017-2066. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/76ec4dc30e9faaf0e4b6093eaa377218-Paper-Conference.pdf.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, Lei Hou, and Juanzi Li. Simulating classroom education with LLM-empowered agents. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pp. 10364–10379, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.520. URL https://aclanthology.org/2025.naacl-long.520/.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), Findings of the Association for Computational Linguistics: NAACL 2024, pp. 2299–2314, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.149. URL https://aclanthology.org/2024.findings-naacl.149/.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 62138–62160. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhou24r.html.

APPENDIX

A METHOD DETAILS

A.1 PROMPT TEMPLATES

EduMCTS instantiates the teacher T_ϕ, student S_ϕ, progress evaluator E_ϕ, and terminal verifier V_ϕ from the same frozen synthesis model using role-specific prompts. The reference answer $y^\star$ is accessible only to E_ϕ and V_ϕ, and is never exposed to the teacher or student. The student uses separate prompts for one-step expansion and autonomous rollout. For SELF, the teacher call is skipped and the student directly generates the next reasoning step. We reproduce the prompts used in our implementation below. Variables enclosed in braces denote runtime inputs.

Teacher and pedagogical actions. The teacher observes the problem and current reasoning history and generates a local intervention according to the selected pedagogical action.

Teacher system prompt

```
You are an expert mathematics/reasoning tutor.
You supervise an already-started student solution.

Your intervention must be LOCAL, concise, and useful for the CURRENT reasoning state.
Never provide the final answer or a full replacement solution.
Do not praise the student. Do not add motivational filler.
Do not invent an error that is not actually present.
Return only the intervention text.
```

The selected pedagogical action further specifies the intervention:

Action-specific instructions

```
[HINT]
Identify the student's current bottleneck and give ONE directional clue.
Do not state the next full derivation, final formula, or final answer.
If the student is already progressing well, point to the next useful consideration rather
    than restating prior work.

[CRITIQUE]
Inspect the existing reasoning for a CONCRETE mathematical/logical error, unsupported
    assumption, contradiction, or invalid inference.
- If a concrete error exists: identify exactly ONE most important error and explain what
    must be reconsidered, without giving the corrected full solution.
- If NO definite error exists: do NOT invent one and do NOT say the reasoning is wrong.
    Instead identify ONE genuinely error-prone point, hidden assumption, boundary case,
    or calculation that should be checked.

[SOCRATIC]
Ask exactly ONE targeted question that makes the student inspect the current bottleneck,
    derive a missing relation, or verify a risky assumption.
The question must be answerable from the problem/current reasoning.
Do not answer the question yourself and do not reveal the final answer.
```

Teacher input template

```
PROBLEM:
{problem}

CURRENT REASONING:
{reasoning_history}

REQUIRED ACTION:
```

```
{action_instruction}
```

Student step generation. During tree expansion, the student generates exactly one local reasoning step conditioned on the current interaction history.

Student step system prompt

```
You are the student solving the problem.
Continue from the existing reasoning history with exactly ONE atomic reasoning step.

Rules:
- Advance only one local subgoal, inference, check, or computation.
- Do not restart or summarize the whole solution.
- Prefer <= 120 words unless equations require slightly more space.
- If a tutor intervention is present, use it as guidance but do the reasoning yourself.
- If this single step genuinely completes the whole problem, end with:
  FINAL ANSWER: <answer or proved statement>
- Otherwise do NOT output FINAL ANSWER.
Return only the next reasoning step.
```

Student step input template

```
PROBLEM:
{problem}

REASONING / TUTOR HISTORY:
{reasoning_history}

Produce the next atomic reasoning step.
```

Autonomous rollout. During simulation, the same student model autonomously completes the remaining solution from the current search state.

Rollout system prompt

```
You are the student completing a reasoning problem autonomously.
Continue from the supplied reasoning history and finish the remaining solution in ONE
     response.

Rules:
- Do not restart from the beginning or repeat already-established work.
- Resolve any visible unresolved issue before relying on it.
- Give enough reasoning for the final conclusion to be justified.
- Solve every requested subpart.
- End with exactly:
  FINAL ANSWER: <complete answer or proved statement>
Return only the continuation and final answer.
```

Rollout input template

```
PROBLEM:
{problem}

REASONING / TUTOR HISTORY SO FAR:
{reasoning_history}

Continue autonomously from exactly this point and finish the entire remaining solution.
```

Progress evaluator. The progress evaluator scores the realized transition from the previous reasoning state to the newly generated student step, rather than judging the teacher intervention in isolation. The reference answer is available only for evaluation.

Progress evaluator system prompt

```
You are a STRICT evaluator of one realized student-reasoning transition.
You are NOT evaluating how nice the teacher message sounds. You judge whether the student'
    s NEW reasoning is better than the BEFORE state after the chosen control action.

You will receive:
1. the original problem;
2. a reference answer/conclusion for checking correctness (never shown to the student);
3. the reasoning BEFORE the action;
4. the chosen action: self / hint / critique / socratic;
5. the teacher intervention, if any;
6. the student's NEW one-step response;
7. whether generation was truncated.

Return ONE integer progress score in {-2,-1,0,1,2}.

+2 = major positive change: fixes a key error, resolves a real bottleneck, or adds a
    decisive correct insight that materially improves the solution state.
+1 = modest but real positive change: a correct, useful next inference/check/computation
    that advances the solution.
 0 = essentially neutral: harmless bookkeeping, restatement, very weak progress, or an
    intervention that was unnecessary and produced no meaningful improvement.
-1 = meaningful deterioration: introduces a nontrivial gap, confusion, unsupported move,
    or follows a poorly targeted intervention in a way that weakens the reasoning state.
-2 = major deterioration: concrete mathematical/logical error, contradiction, compounding
    an existing key error, or a teacher intervention that effectively gives away the
    solution/final answer so the student merely copies rather than reasons.

Critical rules:
- Judge BEFORE -> NEW STUDENT RESPONSE, not the teacher text in isolation.
- A good-looking hint that leads to wrong reasoning must not receive a positive score.
- A critique that invents a nonexistent flaw should be non-positive.
- SELF can receive +1/+2 when autonomous reasoning genuinely progresses.
- Do not reward verbosity, confidence, or surface similarity to the reference answer.
- If truncated=true, score -2.
- Be conservative: +2 should be rare.

Return ONLY valid JSON:
{"progress": 0, "reason": "one short diagnostic sentence"}
```

Progress evaluator input template

```
PROBLEM:
{problem}

REFERENCE ANSWER / CONCLUSION (for evaluator only):
{reference_answer}

BEFORE REASONING:
{reasoning_history}

CONTROL ACTION:
{action}

TEACHER INTERVENTION:
{teacher_feedback}

NEW STUDENT RESPONSE:
```

```
{student_response}

TRUNCATED:
{true_or_false}
```

Only the integer progress score is used to compute the normalized process reward $r_t = p_t/2$ in the search procedure; the diagnostic sentence is retained for inspection.

Terminal verifier. The terminal verifier determines whether the complete trajectory correctly solves the original problem.

Terminal verifier system prompt

```
You are a strict verifier of a complete mathematical/logical solution.
Decide whether the supplied trajectory fully and correctly answers the ORIGINAL problem.

Rules:
- Check the reasoning, not only the text after FINAL ANSWER.
- For proof problems, a bare conclusion is not sufficient: the proof must be logically
    valid.
- For multi-part problems, every requested part must be correctly solved.
- Reject a trajectory with a material unresolved contradiction or invalid step.
- Equivalent mathematical forms are acceptable.
- Teacher/reflection text may guide reasoning but does not excuse a missing or invalid
    solution.
- Marked partial steps must be completed or repaired by later reasoning.
Keep reason to one plain-text sentence of at most 30 words, without LaTeX.

Return ONLY valid JSON:
{"correct": true, "reason": "brief reason"}
```

Terminal verifier input template

```
ORIGINAL PROBLEM:
{problem}

REFERENCE ANSWER / REFERENCE CONCLUSION:
{reference_answer}

CANDIDATE TRAJECTORY:
{complete_reasoning_history}
```

A.2 Search and Trajectory Construction Details

We summarize the complete EduMCTS synthesis procedure in Table 5. Each search node stores its interaction history, visit count $N(v)$, cumulative return $W(v)$, incoming pedagogical action, and local progress reward. Verified successful completions discovered during search are cached and reused in subsequent simulation and trajectory extraction. The implementation uses a student step counter initialized at the root and prevents further expansion once the maximum number of student steps D is reached.

Table 5: **EduMCTS data synthesis procedure.**

Input: problem x, reference answer $y^\star$, search budget B, maximum student steps D, UCT coefficient κ, process weight λ, and filtering threshold η
Output: verified interaction trajectory $\tau^\star$ or $\varnothing$

$u_0 \leftarrow S_\phi^{\text{step}}(x, \varnothing)$; $H_0 \leftarrow [u_0]$ ▷ Initial autonomous step
if u_0 contains a non-truncated final answer **and** $V_\phi(x, y^\star, H_0) = 1$ **then**
  return H_0 ▷ Skip tree search
end if
Initialize root v_0 with $H(v_0) = H_0$; $C \leftarrow 0$
for $b = 1, \ldots, B$ **do**
  $v \leftarrow \textsc{SelectUCT}(v_0, \kappa)$ ▷ Eq. equation 8
  if v is expandable and has fewer than D student steps **then**
    Sample one untried action $a \in \{\textsc{Self}, \textsc{Hint}, \textsc{Critique}, \textsc{Socratic}\}$
    if $a \neq \textsc{Self}$ **then** $m \leftarrow T_\phi(x, H(v), a)$; **else** $m \leftarrow \varnothing$
    $u \leftarrow S_\phi^{\text{step}}(x, H(v) \oplus m)$
    $p \leftarrow E_\phi(x, y^\star, H(v), a, m, u)$
    $v \leftarrow \textsc{AddChild}(v, a, m, u, p/2)$
  end if
  $(R_b, \tau_b) \leftarrow \textsc{Simulate}(v)$ ▷ Reuse cached completion
  $G_b \leftarrow R_b + \lambda \bar{r}(v)$ ▷ Eq. equation 7
  $\textsc{Backup}(v, G_b)$
  $C \leftarrow C + \mathbb{1}[R_b = 1]$
end for
$v^\star \leftarrow \textsc{PrincipalPath}(v_0)$ ▷ Maximize N, then Q
$\tau^\star \leftarrow \textsc{VerifiedCompletion}(v^\star)$
if $\tau^\star$ is not verified **then** $\tau^\star \leftarrow \textsc{SuccessfulFallback}(v_0)$
if $\tau^\star$ is verified and $C/B \geq \eta$ **then**
  return $\tau^\star$
else return $\varnothing$

Selection, expansion, and termination. Selection follows the UCT rule in Equation (8). At an expandable node, one untried action is sampled uniformly, with no fixed preference for SELF. Each action is instantiated at most once per node because children are indexed by their incoming action. For SELF, the teacher call is skipped and the student directly generates the next reasoning step. A node becomes terminal only when the student produces a non-truncated final answer that passes terminal verification. An incorrect final answer therefore remains expandable and can be revised by subsequent pedagogical actions.

Caching and simulation. Simulation performs an autonomous completion from the selected state and assigns terminal reward $R_b \in \{0, 1\}$. If a verified successful completion has already been found from that state, EduMCTS reuses the cached trajectory instead of issuing another rollout. The return $G_b = R_b + \lambda \bar{r}(v)$ is then backed up to the simulation node and all of its ancestors. The search budget B counts MCTS iterations rather than individual LLM calls.

Trajectory extraction and filtering. After search, EduMCTS constructs the principal path by repeatedly selecting the most visited child, breaking ties by empirical value Q. If the endpoint has no cached successful completion, one fresh autonomous rollout is attempted. If this trajectory still fails verification, the algorithm falls back to a verified successful branch elsewhere in the tree, prioritizing higher visit counts and values. Finally, the trajectory is retained only if it passes terminal verification and its empirical success rate

$$\hat{\rho} = \frac{1}{B} \sum_{b=1}^{B} R_b$$

satisfies $\hat{\rho} \geq \eta$.

Table 6: Search configuration used for EduMCTS data synthesis.

Parameter	Value
Search budget B	20
Maximum student steps D	4
UCT coefficient κ	1.4
Process weight λ	0.25
Success-rate threshold η	0.10
Terminal reward (correct / incorrect)	$1/0$

Hyperparameter selection. We select the search budget using a pilot study on 200 seed problems, varying $B \in \{4, 8, 12, 16, 20\}$. The verified-trajectory rate increases with the budget and begins to saturate at $B = 20$, which we therefore use in all experiments. Under this budget, successful searches typically reach depth 3–4, motivating the maximum depth $D = 4$. We set the UCT coefficient to $\kappa = 1.4$, close to the conventional $\sqrt{2}$ value. The process weight is fixed to $\lambda = 0.25 < 0.5$, ensuring that terminal correctness remains dominant over the normalized process reward $\bar{r} \in [-1, 1]$. Finally, we use $\eta = 0.10$, requiring at least two successful simulations under $B = 20$, as a mild stability filter.

B Experimental Setup Details

B.1 Data Construction and Synthesis Details

Seed pool and sampling. We sample 1,800 question–answer pairs from the competition-level portion of NuminaMath-1.5 to construct the synthesis seed pool.

Seed filtering. We further apply a GPT-4.1-based quality review to exclude seed problems that are unsuitable for reliable synthesis. The review considers four major sources of invalidity: missing figures, formulas, tables, or necessary conditions; ambiguous or underspecified problem statements; incorrect or unverifiable reference answers; and undefined variables or other forms of incompleteness. This process removes 469 examples, leaving 1,331 valid seed problems for EduMCTS synthesis.

Quality-review prompt. For reproducibility, we provide the prompt used for seed quality review. The reviewer is instructed to prefer conservatively flagging problematic examples rather than retaining uncertain ones.

Seed quality-review prompt

```
You are an expert quality reviewer for mathematical and reasoning problems.
Strictly inspect the following problem and identify as many potential issues as possible.

PROBLEM:
{problem}

REFERENCE ANSWER:
{reference_answer}

Strictly evaluate the following aspects:

1. Problem completeness (very strict):
- Does the problem contain phrases such as "as shown in the figure",
  "see figure", "in the figure", "above figure", "below figure", or
  similar expressions indicating a missing image?
- Does it contain phrases such as "as shown in the table", "see table",
  or "in the table", indicating a missing table?
- Does it mention a graph, geometric diagram, coordinate system, or
  function plot without actually providing the corresponding content?
- Does it contain complex mathematical symbols, formulas, matrices,
  integrals, summations, or other content that may be incomplete in
```

```
  plain text?
- Are there undefined variables or missing given conditions?
- Is there a case where some quantity is stated as known but its value
  or definition is never actually provided?

2. Problem clarity (strict):
- Is the statement ambiguous or open to multiple interpretations?
- Are there logically contradictory or mutually inconsistent conditions?
- Are any key concepts undefined or insufficiently specified?

3. Reference-answer correctness (strict):
- Even if the problem itself has issues, is the reference answer correct
  under the provided conditions?
- Is the answer complete in form, e.g., does a proof problem provide an
  actual proof rather than only a conclusion?
- Does the reference answer match what the problem asks?
- If the reference answer can clearly be determined to be wrong, mark
  reference_correct as false.
- If incomplete problem information makes verification impossible, mark
  reference_correct as "unknown".
- Mark reference_correct as true only when you are confident that the
  reference answer is correct.

4. Overall solvability:
- Can a human mathematics expert completely solve the problem without
  requiring any additional information?
- If not, set solvable to false.

Return valid JSON only. Prefer reporting a potential issue rather than
missing one:
{
  "solvable": true/false,
  "issue_type": "none" | "missing_figure" | "missing_table" |
                "missing_formula" | "ambiguous" | "wrong_answer" |
                "undefined_variables" | "incomplete",
  "issue_detail": "Detailed description of all identified issues.",
  "reference_correct": true/false/"unknown",
  "difficulty": "easy" | "medium" | "hard" | "very_hard",
  "confidence": "high" | "medium" | "low"
}
```

Trajectory synthesis and filtering. We run EduMCTS on all 1,331 retained seed problems using `qwen3-235b-a22b-instruct-2507` as the frozen synthesis model. After search and trajectory extraction, 1,081 problems yield a terminally verified candidate trajectory. We then apply the empirical success-rate criterion defined in Section A.2, together with structural validity checks requiring the selected trajectory to start from student reasoning and terminate with a non-truncated `FINAL ANSWER:`. One candidate fails the success-rate criterion, resulting in 1,080 final training trajectories.

Synthesis configuration. All synthesis is performed through an OpenAI-compatible interface with 30 parallel workers. The role-specific prompts are provided in Section A.1, while the complete search procedure and its hyperparameters are given in Section A.2. A summary of the data construction pipeline is provided in Table 7.

Table 7: Summary of the EduMCTS data construction pipeline.

Stage	Value
Sampled for synthesis	1,800
Excluded by quality review	469
Valid synthesis seeds	1,331
Terminally verified candidates	1,081
Final retained trajectories	1,080
Synthesis model	`qwen3-235b-a22b-instruct-2507`
Parallel workers	30

B.2 Baseline and Fine-Tuning Details

Baseline methods. We compare EduMCTS with six representative data construction baselines. **Ori** directly uses the original reference solutions without augmentation. **CoT** regenerates a step-by-step reasoning solution for each source problem. **DART** uses repeated sampling and correctness-based selection to obtain valid reasoning trajectories. **MetaMath** augments the data through problem rewriting, producing multiple variants of a source problem. **MASTER** constructs reasoning supervision through simulated teacher–student interactions under pedagogical scenarios. **CRPS** uses search trajectories to construct contrastive reasoning signals and synthesize improved solution paths. These baselines cover direct solution regeneration, problem rewriting, sampling-based selection, pedagogical interaction, and search-based reasoning synthesis.

Baseline data construction. All baselines are constructed from the same 1,080 source problems corresponding to the final EduMCTS training set, together with their original reference solutions. For methods requiring model generation, we use `qwen3-235b-a22b-instruct-2507` as the common synthesis model to minimize differences attributable to the generator, while otherwise following the default settings of the corresponding open-source implementations. To maintain a controlled training-data budget, CoT, DART, MASTER, CRPS, and EduMCTS each contribute one training instance per source problem. In our implementation, CoT and MASTER generate one augmented trajectory per seed, whereas DART and CRPS may explore multiple candidates internally, from which one valid synthesized trajectory is retained. If no valid augmentation is obtained, the corresponding original instance is retained instead. Consequently, these methods each contain 1,080 training examples, while Ori consists of the 1,080 original reference solutions. MetaMath is the only exception: because multi-variant problem rewriting is an integral part of its augmentation procedure, we retain all generated variants, resulting in 2,604 training examples.

Ablation data construction. All ablation variants start from the same 1,331 filtered seed problems used by EduMCTS and use the same synthesis model and generation settings unless the corresponding component is explicitly removed. For **w/o Teach**, we remove the pedagogical interventions and replace the HINT, CRITIQUE, and SOCRATIC branches with autonomous student continuations, while retaining the same branching and evaluation procedure. For **w/o Search**, we remove the MCTS-based tree search and construct trajectories without UCT-guided exploration, while retaining the pedagogical interaction and evaluation mechanisms. For **w/o Process Reward**, we retain the full search procedure but remove local progress from the search return, i.e., set $\lambda = 0$ so that branch evaluation depends only on terminal correctness. Starting from the common 1,331 seeds, these three variants yield 1,171, 1,161, and 1,056 valid training trajectories, respectively.

Downstream fine-tuning. We fine-tune Qwen3-8B-Base, Qwen3-14B-Base, and Llama3.1-8B-Instruct using LLaMA-Factory with the same LoRA SFT configuration across all data construction methods. We use the `default` conversation template for the two Qwen3 base models and the `llama3` template for Llama3.1-8B-Instruct. LoRA is applied to all attention projection and MLP projection layers, while all other training hyperparameters are kept fixed across methods. The complete configuration is summarized in Table 8.

Table 8: Downstream fine-tuning configuration used for all training datasets and backbone models.

Configuration	Value
Fine-tuning framework	LLaMA-Factory
Training stage	SFT
Fine-tuning method	LoRA
LoRA rank r	8
LoRA alpha	16
LoRA target modules	`q/k/v/o`, `gate/up/down` projections
Learning rate	5×10^{-5}
Epochs	3
LR scheduler	Cosine
Warmup ratio	0.1
Per-device batch size	1
Gradient accumulation	8
Maximum sequence length	4,096
Precision	BF16
Gradient checkpointing	Enabled
Distributed training	DeepSpeed ZeRO-3
Preprocessing workers	16
Dataloader workers	4
Random seed	42
Qwen3 template	`default`
Llama3.1 template	`llama3`

Computing environment. Experiments are conducted on a local Slurm-based computing cluster. Each compute node is equipped with a 48-core CPU, eight NVIDIA L20 GPUs with 48 GB of memory per GPU, and approximately 925 GB of system memory. Multi-GPU fine-tuning is performed with DeepSpeed ZeRO-3, and gradient checkpointing is enabled to reduce memory consumption.

B.3 Evaluation and Analysis Protocols

Benchmarks and evaluation setting. We evaluate all downstream models on eleven reasoning benchmarks, including seven mathematical reasoning benchmarks and four additional science benchmarks for cross-domain generalization. The mathematical benchmarks are MATH-500, GSM8K, AMC, AIME, MMLU-Pro Math, AGIEval Math, and OlympiadBench. The science benchmarks are ARC-Challenge (ARC-C), GPQA-Diamond, MMLU-STEM, and the science subsets of MMLU-Pro. The four science benchmarks are reported separately as a measure of cross-domain generalization. The evaluation sizes in Table 9 correspond to the examples actually processed in our evaluation runs.

Table 9: Benchmarks used for downstream evaluation. Sizes correspond to the examples actually evaluated in our experiments.

Benchmark	Domain	Subset / Split	Size	Format	Grading
MATH-500	Math	`math_500`, Levels 1–5	500	Open	Numeric / symbolic
GSM8K	Math	`gsm8k_default`, test	1,319	Open	Numeric
AMC	Math	AMC 2022–2024	134	Open	Numeric
AIME	Math	AIME 2024–2026	90	Open	Numeric
MMLU-Pro Math	Math	Mathematics subset	1,351	MCQ	Choice
AGIEval Math	Math	Mathematics subset	1,000	Open	Numeric / symbolic
OlympiadBench	Math	`OE_TO_maths_en_COMP`	674	Open	Numeric / symbolic
ARC-C	Science	`ARC-Challenge`	1,172	MCQ	Choice
GPQA-Diamond	Science	Diamond subset	198	MCQ	Choice
MMLU-STEM	Science	18 STEM subsets, 5-shot	3,018	MCQ	Choice
MMLU-Pro Sci	Science	Physics / Chemistry / Biology / CS / Engineering	4,527	MCQ	Choice

Inference and answer grading. After LoRA fine-tuning, each checkpoint is merged with its corresponding backbone and served through a local OpenAI-compatible inference endpoint. EvalScope queries the merged model through its `openai_api` backend. All methods are evaluated with the same benchmark-specific prompt and decoding configuration.

We use deterministic decoding with temperature 0, `top_p`=1, and random seed 42. The maximum generation length is 2,048 tokens for all benchmarks.

We use the benchmark-specific prompt templates provided by EvalScope without additional method-specific prompt engineering. For example, MATH-500 uses the following template:

MATH-500 evaluation prompt

```
{question}
Please reason step by step, and put your final answer within \boxed{}.
```

GSM8K, AMC, and OlympiadBench use the same general instruction to reason step by step and place the final answer in `\boxed{}`. AIME uses an equivalent step-by-step template with the same boxed-answer requirement. Multiple-choice benchmarks instead request an explicit final option in the form `ANSWER: [LETTER]` after the reasoning.

Answer extraction and grading are performed by the benchmark-specific EvalScope adapters. Open-ended mathematical benchmarks use their corresponding numerical or symbolic equivalence procedures, whereas multiple-choice benchmarks are graded by the extracted answer option.

Trajectory-statistics analysis. To examine whether the gains of EduMCTS can be explained by longer or more elaborate reasoning traces, we analyze both the synthesized training data and the corresponding MATH-500 inference trajectories using a unified Data-Juicer protocol (Chen et al., 2024; 2025). For the training-data analysis, we compare Ori, CoT, DART, MetaMath, MASTER, CRPS, and EduMCTS. For the inference analysis, we apply the same pipeline to the MATH-500 trajectories generated by the corresponding Qwen3-8B models under the same deterministic evaluation setting.

We measure four surface statistics: token count, character count, dependency edges, and verb count. Token count measures sequence length, character count provides a tokenizer-independent measure of textual length, dependency edges approximate syntactic structural complexity, and verb count provides a coarse measure of the number of expressed reasoning actions. English syntactic analysis is performed with spaCy, while token counting uses tiktoken. These statistics are used only as descriptive diagnostics of trajectory length and surface complexity, rather than as measures of reasoning quality.

Explicit pedagogical behavior matching. We distinguish pedagogical behaviors that are explicitly represented by the reflection format learned during training from analogous behaviors expressed implicitly in natural-language reasoning. An explicit pedagogical behavior is identified only when the generated trajectory contains one of the following properly closed typed reflection tags:

```
<reflection type="hint">...</reflection>
<reflection type="critique">...</reflection>
<reflection type="socratic">...</reflection>
```

We use strict literal regular-expression matching: the value of `type` must be exactly `hint`, `critique`, or `socratic`, and the corresponding reflection tag must be properly closed. We deliberately avoid keyword-based matching because ordinary reasoning statements may contain reflective language without constituting an explicit pedagogical action. Consequently, untagged natural-language behaviors are not counted by this explicit matching protocol.

Implicit pedagogical behavior judging. We further measure whether analogous pedagogical behaviors emerge without explicit reflection tags. We define the three behaviors operationally according to their function in the reasoning process.

A HINT redirects the current approach or supplies a missing direction, for example by suggesting a useful relation, theorem, method, or next consideration, without carrying out the subsequent derivation itself. A CRITIQUE identifies a concrete deficiency in reasoning that has already occurred, such as an error, unsupported assumption, contradiction, missing case, or inconsistency. A SOCRATIC move prompts the solver to inspect, justify, or recover part of its own reasoning through a targeted question or challenge rather than directly supplying the missing reasoning.

We use GPT-4.1 as an independent reader for this analysis, with temperature 0 and a maximum output length of 1,400 tokens. The judge is instructed to classify behaviors according to their functional role rather than particular lexical cues. The system prompt is:

Implicit behavior judging prompt

```
You read a worked mathematical solution and locate the places where the writer steps
outside the mathematics to manage its own reasoning.

You are looking for four kinds of move. Judge each by what it DOES, not by the words
it uses.

hint       redirects the approach or supplies a missing direction -- offering a
           different method, naming a relation or theorem to use, or pointing at
           the next thing to consider. It does NOT carry out the next derivation
           itself.

socratic   makes the solver inspect its own work -- poses a question, challenge, or
           prompt whose function is to force a check, a justification, or a
           recovery of a missing relation.

critique   asserts that something already done is wrong, unsupported, or
           contradictory -- flags an error, unjustified assumption, missing case,
           or inconsistency with an earlier result.

self_check verifies a completed result without asserting it was wrong -- for example,
           by substituting a value back, re-deriving by another route, or checking
           a boundary case.

Rules:
- Locate at most 8 substantial moves.
- A solution with no such move is common; return an empty list rather than inventing moves
    .
- Quote each move briefly.
- Do not judge whether the move was correct or helpful; only locate and classify it.

Return only a JSON object with the detected moves and their types.
```

Difficulty-stratified behavior analysis. We conduct the behavioral analysis on the 500 MATH-500 trajectories generated by the EduMCTS-trained Qwen3-8B model under the same deterministic evaluation setting described above. MATH-500 difficulty is taken directly from its native `Level 1`–`Level 5` annotations; no difficulty labels are assigned by EduMCTS or by an external judge.

For each difficulty level, we compute the proportion of trajectories containing at least one detected pedagogical behavior and separately count the occurrences of HINT, SOCRATIC, and CRITIQUE. The coverage curves in Figure 4 summarize the level-wise trend, while Table 10 reports the full statistics for both explicit reflections and implicit teaching actions. Because a single trajectory may contain multiple actions, action counts can exceed the number of traces.

Table 10: **Teaching actions in MATH-500 traces by difficulty.** Coverage denotes the percentage of traces containing at least one action in the corresponding channel.

		Explicit reflections				Implicit teaching actions			
Level	**# traces**	**Coverage**	**Hint**	**Socratic**	**Critique**	**Coverage**	**Hint**	**Socratic**	**Critique**
L1	43	0.0%	0	0	0	39.5%	24	6	2
L2	90	7.8%	3	3	2	54.4%	84	20	15
L3	105	12.4%	8	10	2	60.0%	103	41	18
L4	128	10.9%	10	5	1	68.0%	196	43	52
L5	134	11.2%	10	8	3	82.8%	284	84	113
All	**500**	**9.8%**	**31**	**26**	**8**	**65.4%**	**691**	**194**	**200**

The complete statistics reinforce the trend shown in the main paper: implicit pedagogical behaviors become increasingly prevalent as problem difficulty rises, while explicit reflection tags remain comparatively sparse. The increase is also distributed across all three pedagogical actions rather than being driven by a single action type.

C Experimental Result Details

C.1 Full Quantitative Results

This section provides the complete quantitative results corresponding to the condensed tables reported in the main paper. All results are reported as mean ± sample standard deviation over two independent runs. Unless otherwise specified, Avg. denotes the unweighted mean across all eleven evaluation benchmarks.

Main results. Table 11 reports the complete results of all data construction methods on the three downstream backbones. EduMCTS achieves the highest overall average on Qwen3-8B-Base, Qwen3-14B-Base, and Llama3.1-8B-Instruct, with average accuracies of 66.65%, 70.76%, and 46.85%, respectively. The full per-benchmark results also show that the improvements extend across both mathematical and scientific reasoning tasks.

Table 11: **Full reasoning results across two independent runs.**

Method	MATH	GSM8K	AMC	AIME	MMLU-P	AGIE	Olym.	ARC-C	GPQA	MMLU-S	MMLU-P(S)	Avg.
Qwen3-8B-Base												
w/o SFT	77.60±0.85	92.31±0.37	42.91±2.64	11.11±1.57	55.70±3.08	73.65±0.64	**44.52±0.63**	71.33±0.00	31.56±0.35	73.69±0.00	50.16±0.08	56.78±0.53
Ori	59.30±0.14	85.86±0.06	21.40±0.35	0.56±0.78	58.33±0.00	55.55±0.07	20.70±0.32	82.51±0.12	36.37±2.14	75.75±0.19	53.64±0.19	49.99±0.28
Dart	62.70±0.14	88.29±0.27	30.97±3.69	1.67±0.78	69.21±0.52	62.40±0.28	39.39±0.52	93.69±0.24	32.83±2.86	84.83±0.23	62.29±0.10	57.11±0.71
MetaMath	74.80±0.57	90.34±0.05	32.09±0.00	12.22±1.57	71.83±0.47	71.00±0.57	17.88±0.11	92.11±0.18	34.09±1.07	84.84±0.49	60.10±0.21	58.30±0.19
CoT	75.60±0.85	91.85±0.16	42.44±0.14	12.78±2.35	73.06±0.21	73.30±0.14	34.65±1.36	92.92±0.72	32.32±2.86	85.34±0.21	60.93±0.56	61.38±0.56
MASTER	72.70±1.56	91.93±0.49	36.94±0.52	13.89±2.36	78.94±0.99	72.25±0.49	30.71±0.21	90.15±0.30	42.17±1.07	85.29±0.00	**64.67±0.08**	61.78±0.23
CRPS	78.00±1.13	93.14±0.59	42.16±1.58	15.56±0.00	**79.61±0.37**	75.85±0.35	43.03±0.21	93.35±0.12	40.15±0.35	85.29±0.28	63.39±0.70	64.50±0.07
EduMCTS	**82.20±0.28**	**94.13±0.37**	**48.13±1.58**	**20.56±3.92**	78.50±0.68	**78.32±0.30**	42.81±0.11	**94.67±0.06**	**43.18±1.07**	**86.09±0.05**	64.54±0.05	**66.65±0.19**
Qwen3-14B-Base												
w/o SFT	80.30±1.56	93.82±0.16	51.87±0.53	12.22±0.00	57.18±0.37	75.80±0.28	46.44±0.63	93.43±0.48	40.15±0.35	84.15±0.49	43.67±0.25	61.73±0.08
Ori	79.00±0.00	88.63±0.33	45.90±2.64	3.89±0.78	68.65±0.16	74.65±0.35	40.06±1.05	93.13±0.06	45.96±0.71	87.25±0.19	64.00±0.12	62.83±0.31
Dart	82.30±0.99	90.64±0.05	46.64±0.52	2.78±0.78	78.20±0.06	78.80±1.13	50.00±0.42	94.36±0.00	45.71±1.79	87.96±0.07	66.96±0.06	65.85±0.26
MetaMath	78.00±0.85	**95.72±0.05**	42.91±0.52	12.78±0.78	81.09±0.88	75.90±0.57	43.25±0.74	94.26±0.57	39.14±1.07	87.41±0.14	65.71±2.55	65.11±0.09
CoT	80.40±0.57	95.49±0.06	47.01±0.00	13.89±2.36	79.46±0.16	77.65±0.07	42.95±0.95	94.71±0.36	43.18±1.07	88.21±1.12	69.50±1.56	66.59±0.37
MASTER	82.70±0.99	93.67±0.91	49.63±0.53	16.12±0.78	78.43±0.26	77.45±0.07	42.14±0.21	92.58±0.73	38.13±1.07	89.00±0.14	70.05±0.07	66.35±0.31
CRPS	83.10±0.71	95.11±0.06	51.12±0.52	17.23±0.78	84.05±0.47	**81.35±0.07**	49.93±0.11	94.93±0.30	50.00±2.86	**89.50±0.42**	70.75±0.21	69.73±0.04
EduMCTS	**86.00±0.28**	95.15±0.11	**52.24±0.00**	**18.34±0.78**	**84.42±0.16**	**81.35±0.49**	**51.93±0.21**	**95.56±0.24**	**52.53±0.71**	89.42±0.12	**71.44±0.18**	**70.76±0.07**
Llama3.1-8B-Instruct												
w/o SFT	49.50±1.84	86.28±0.21	17.16±0.00	**2.22±0.00**	50.19±0.21	50.35±0.92	18.55±0.21	87.97±0.12	26.77±0.00	69.09±0.28	42.70±0.28	45.52±0.17
Ori	43.10±0.71	80.78±0.16	13.81±0.53	0.56±0.78	42.71±0.21	42.95±0.35	10.69±0.42	87.29±0.12	**30.30±0.00**	67.95±0.06	39.30±0.57	41.77±0.05
Dart	49.30±1.27	82.87±0.32	14.93±1.05	1.67±0.78	42.42±0.11	45.15±1.34	11.80±0.53	85.84±0.00	25.51±1.07	65.91±0.42	37.80±0.28	42.11±0.03
MetaMath	51.20±0.85	84.08±0.42	15.68±1.05	0.56±0.78	47.93±0.89	50.25±0.64	17.29±0.94	86.86±0.13	23.74±0.71	66.85±0.49	39.71±0.14	44.01±0.40
CoT	51.60±0.85	83.51±0.16	16.05±0.53	**2.22±0.00**	50.19±0.00	50.40±1.98	15.28±0.21	86.44±0.12	19.70±2.14	67.10±0.28	39.71±0.14	43.84±0.10
MASTER	51.80±1.13	85.03±0.59	18.28±1.58	**2.22±1.57**	50.67±0.16	48.85±0.21	17.07±0.63	87.55±0.12	23.49±1.07	70.32±0.84	41.94±0.08	45.20±0.21
CRPS	52.60±0.00	85.07±0.32	17.54±1.58	**2.22±0.00**	49.08±0.21	50.70±1.41	18.55±1.05	87.46±1.09	23.49±0.36	69.33±0.07	42.50±0.57	45.32±0.01
EduMCTS	**56.20±1.70**	**86.66±0.54**	**18.66±0.00**	**2.22±0.00**	**51.18±0.47**	**51.00±0.42**	**19.44±0.21**	**88.57±0.48**	25.25±1.43	**71.76±0.77**	**44.40±0.00**	**46.85±0.23**

Augmentation ratio. Table 12 provides the full results for varying the proportion of EduMCTS trajectories in the training data on Qwen3-8B-Base. The overall average increases from 49.99% with only original data to 57.45%, 61.33%, 64.46%, and 66.65% as the EduMCTS share increases from 25% to 50%, 75%, and 100%, respectively. These results provide the complete benchmark-level statistics underlying the augmentation-ratio analysis in the main paper.

Table 12: **Full results for different EduMCTS data shares.**

Share	MATH	GSM8K	AMC	AIME	MMLU-P	AGIE	Olym.	ARC-C	GPQA	MMLU-S	MMLU-P-S	Avg.
Qwen3-8B-Base												
0% (Ori)	59.30±0.14	85.86±0.06	21.40±0.35	0.56±0.78	58.33±0.00	55.55±0.07	20.70±0.32	82.51±0.12	36.37±2.14	75.75±0.19	53.64±0.19	49.99±0.28
25%	40.90±0.99	87.61±0.80	35.45±0.53	13.33±0.00	73.06±0.32	66.20±0.28	41.39±0.00	90.52±0.02	40.66±3.22	79.54±0.21	63.35±0.36	57.45±0.31
50%	53.00±2.26	89.58±0.26	40.55±2.45	17.78±0.00	76.91±0.21	72.55±0.07	42.95±0.74	92.32±0.25	41.41±1.43	84.20±0.13	63.32±0.11	61.33±0.63
75%	65.50±0.14	91.51±0.42	47.76±3.17	17.23±2.35	78.28±0.06	76.45±0.64	**43.85±0.11**	93.05±0.18	**45.45±1.43**	85.73±0.05	64.23±0.08	64.46±0.34
100%	**82.20±0.28**	**94.13±0.37**	**48.13±1.58**	**20.56±3.92**	**78.50±0.68**	**78.32±0.30**	42.81±0.11	**94.67±0.06**	43.18±1.07	**86.09±0.05**	**64.54±0.05**	**66.65±0.19**

Component ablations. Table 13 reports the complete component ablation results on Qwen3-8B-Base. Removing pedagogical intervention, tree search, or process reward reduces the overall average from 66.65% to 65.06%, 65.06%, and 65.04%, respectively. The complete per-benchmark results support the contribution of all three components reported in the main paper.

Table 13: **Full component ablation results on Qwen3-8B-Base.**

Method	MATH	GSM8K	AMC	AIME	MMLU-P	AGIE	Olym.	ARC-C	GPQA	MMLU-S	MMLU-P-S	Avg.
Qwen3-8B-Base												
w/o Teach	79.70±0.99	92.76±0.16	47.76±2.11	15.56±0.00	77.50±0.42	76.85±1.06	**43.77±0.42**	93.69±0.60	39.39±0.00	85.59±0.28	63.15±0.08	65.06±0.37
w/o Search	80.30±0.71	92.76±0.27	45.90±0.53	11.67±2.35	**79.28±0.11**	77.10±0.71	43.10±0.52	93.60±0.48	41.67±0.35	85.69±0.14	**64.56±0.21**	65.06±0.54
w/o Process Reward	80.30±0.42	92.27±0.00	45.90±0.53	16.12±0.78	78.39±0.21	77.10±0.42	42.58±0.42	93.56±0.42	39.39±0.00	85.51±0.12	64.31±0.30	65.04±0.15
EduMCTS	**82.20±0.28**	**94.13±0.37**	**48.13±1.58**	**20.56±3.92**	78.50±0.68	**78.32±0.30**	42.81±0.11	**94.67±0.06**	**43.18±1.07**	**86.09±0.05**	64.54±0.05	**66.65±0.19**

C.2 Detailed Qualitative Cases

To complement the representative excerpts in Table 4, we provide two complete reasoning trajectories generated by the EduMCTS-trained Qwen3-8B model on MATH-500. The trajectories are reproduced without manual correction or deletion of intermediate reasoning. We only typeset the explicit pedagogical reflections separately for readability. Both examples receive full credit under the benchmark evaluator.

The first case illustrates how a CRITIQUE intervention is triggered after an unproductive decomposition attempt and redirects the solution toward a telescoping identity. The second case shows how HINT and SOCRATIC interventions can appear at different stages of the same trajectory, first guiding a local algebraic operation and later prompting the solver to connect the derived form to the target ratio.

Case 1: Critique after an invalid decomposition attempt.

MATH-500, Level 4: Fibonacci series

Problem. Compute

$$\sum_{n=1}^{\infty} \frac{F_{n+1}}{F_n F_{n+2}},$$

where F_n denotes the nth Fibonacci number, with $F_0 = 0$ and $F_1 = 1$.

EduMCTS trajectory.
We begin by expressing the given sum in terms of a telescoping series. Recall that the Fibonacci numbers satisfy the recurrence

$$F_{n+2} = F_{n+1} + F_n \qquad \text{for all } n \geq 0,$$

with initial conditions $F_0 = 0$, $F_1 = 1$.
Consider the term

$$\frac{F_{n+1}}{F_n F_{n+2}}.$$

Using $F_{n+2} = F_{n+1} + F_n$, the denominator becomes $F_n(F_{n+1} + F_n) = F_n F_{n+1} + F_n^2$. Thus,

$$\frac{F_{n+1}}{F_n F_{n+2}} = \frac{F_{n+1}}{F_n F_{n+1} + F_n^2}.$$

But this does not immediately suggest telescoping. Instead, we attempt partial fraction decomposition.
Let us write

$$\frac{1}{F_n F_{n+2}} = A\frac{1}{F_n} + B\frac{1}{F_{n+2}}.$$

Multiplying both sides by $F_n F_{n+2}$ gives

$$1 = AF_{n+2} + BF_n.$$

We aim to find constants A and B such that this holds for all n. Using the recurrence, try to relate Fibonacci terms.

Reflection: Critique

The attempted partial fraction decomposition assumption is incorrect: a decomposition of the form

$$\frac{1}{F_nF_{n+2}} = \frac{A}{F_n} + \frac{B}{F_{n+2}}$$

is not valid. This would imply that $1 = AF_{n+2} + BF_n$, which would require a constant identity in n, but the right-hand side scales with n, making it impossible for constants A and B to satisfy this equality for all $n \geq 0$. Fibonacci numbers grow exponentially, so such a linear combination cannot yield the constant 1 for all n.
The current approach must be revised—consider an alternative identity or expression that transforms the original term into a telescoping difference. Think about identities involving ratios or recurrence relations that simplify $\frac{F_{n+1}}{F_nF_{n+2}}$ without recourse to invalid decomposition.

We revisit the general term

$$\frac{F_{n+1}}{F_nF_{n+2}}.$$

Using the Fibonacci recurrence $F_{n+2} = F_{n+1} + F_n$, we write

$$\frac{F_{n+1}}{F_nF_{n+2}} = \frac{F_{n+1}}{F_n(F_{n+1} + F_n)}.$$

Now divide numerator and denominator by F_{n+1}:

$$= \frac{1}{\frac{F_n}{F_{n+1}}\left(1 + \frac{F_n}{F_{n+1}}\right)} = \frac{1}{\phi_n(1+\phi_n)}, \qquad \text{where } \phi_n = \frac{F_n}{F_{n+1}}.$$

But instead of introducing ratios, let's compute the reciprocal.
Let $a_n = F_nF_{n+1}$ and note $F_{n+2} = F_{n+1} + F_n$. Then

$$\frac{F_{n+1}}{F_nF_{n+2}} = \frac{F_{n+1}}{F_n(F_{n+1} + F_n)} = \frac{1}{F_n\phi_n(1+\phi_n)},$$

where $\phi_n = F_{n+1}/F_n$. This also seems indirect. Try instead expressing the difference

$$\frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}}.$$

Compute the difference:

$$\frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}} = \frac{F_{n+2} - F_n}{F_nF_{n+1}F_{n+2}}.$$

Now use $F_{n+2} = F_n + F_{n+1}$:

$$F_{n+2} - F_n = F_{n+1}.$$

So

$$\frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}} = \frac{F_{n+1}}{F_nF_{n+1}F_{n+2}} = \frac{1}{F_nF_{n+2}}.$$

Hence

$$\frac{1}{F_nF_{n+2}} = \frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}}.$$

Therefore, multiplying both sides by F_{n+1},

$$\frac{F_{n+1}}{F_nF_{n+2}} = \frac{1}{F_n} - \frac{1}{F_{n+1}}.$$

Wait—not quite.

$$\frac{F_{n+1}}{F_nF_{n+2}} = \frac{1}{F_nF_{n+2}} \cdot F_{n+1}$$
$$= \left(\frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}}\right) F_{n+1}.$$

Compute:

$$\left(\frac{1}{F_nF_{n+1}} - \frac{1}{F_{n+1}F_{n+2}}\right) F_{n+1} = \frac{1}{F_n} - \frac{1}{F_{n+2}}.$$

Yes! So

$$\frac{F_{n+1}}{F_nF_{n+2}} = \frac{1}{F_n} - \frac{1}{F_{n+2}}.$$

Now sum over $n = 1$ to ∞:

$$\sum_{n=1}^{\infty} \frac{F_{n+1}}{F_nF_{n+2}} = \sum_{n=1}^{\infty} \left(\frac{1}{F_n} - \frac{1}{F_{n+2}}\right).$$

This is a telescoping series.
Write out terms:

$$n = 1 : \frac{1}{F_1} - \frac{1}{F_3},$$
$$n = 2 : \frac{1}{F_2} - \frac{1}{F_4},$$
$$n = 3 : \frac{1}{F_3} - \frac{1}{F_5},$$
$$\vdots$$

Therefore,

$$S = \left(\frac{1}{F_1} + \frac{1}{F_2} + \cdots\right) - \left(\frac{1}{F_3} + \frac{1}{F_4} + \cdots\right)$$
$$= \frac{1}{F_1} + \frac{1}{F_2} = 1 + 1 = 2.$$

FINAL ANSWER: $\boxed{2}$

Case 2: Hint and Socratic guidance within one trajectory.

MATH-500, Level 3: Quadratic polynomial ratio

Problem. Let $p(x)$ be a quadratic polynomial with integer coefficients which has $4 - \sqrt{11}$ as a root. Compute

$$\frac{p(3)}{p(4)}.$$

EduMCTS trajectory.
Since $p(x)$ is a quadratic polynomial with integer coefficients and $4-\sqrt{11}$ is a root, its conjugate $4+\sqrt{11}$ must also be a root. This follows from the fact that irrational roots of polynomials with integer coefficients come in conjugate pairs.
Thus, the quadratic polynomial can be written as

$$p(x) = a(x - (4 - \sqrt{11}))(x - (4 + \sqrt{11}))$$

for some integer $a \neq 0$.

Reflection: Hint

Expand the product of the two factors and simplify the expression inside the parentheses.

Expanding the product of the factors:

$$(x-(4-\sqrt{11}))(x-(4+\sqrt{11}))$$
$$=(x-4+\sqrt{11})(x-4-\sqrt{11}).$$

This is a difference of squares:

$$(x-4)^2-(\sqrt{11})^2=(x^2-8x+16)-11=x^2-8x+5.$$

Thus, the polynomial is

$$p(x)=a(x^2-8x+5)$$

for some integer $a \neq 0$.

Reflection: Socratic

Since a is an integer and $p(x)$ has integer coefficients, what can you say about the value of $p(3)$ and $p(4)$ in terms of a, and how does this affect the ratio $\frac{p(3)}{p(4)}$?

Compute $p(3)$ and $p(4)$ using $p(x)=a(x^2-8x+5)$:

$$p(3)=a(3^2-8\cdot 3+5)=a(9-24+5)=-10a,$$

and

$$p(4)=a(4^2-8\cdot 4+5)=a(16-32+5)=-11a.$$

Now compute the ratio:

$$\frac{p(3)}{p(4)}=\frac{-10a}{-11a}=\frac{10}{11}.$$

FINAL ANSWER: $\boxed{\frac{10}{11}}$