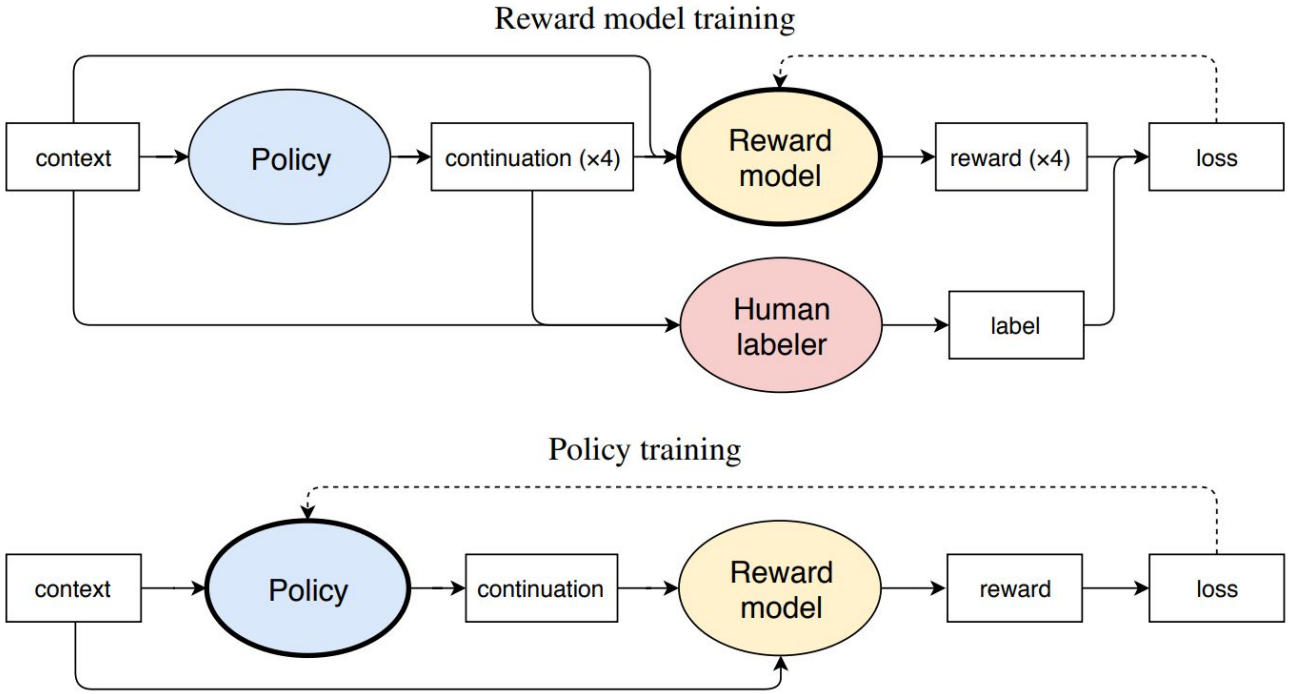
22.10.2025

# Zastosowania algorytmu MCTS w generowaniu i optymalizacji procesów rozumowania wielkich modeli językowych

Szymon Pawlonka




$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right],$$

Ziegler et al. (2019). Fine-Tuning Language Models from Human Preferences. OpenAI. https://arxiv.org/abs/2305.18290

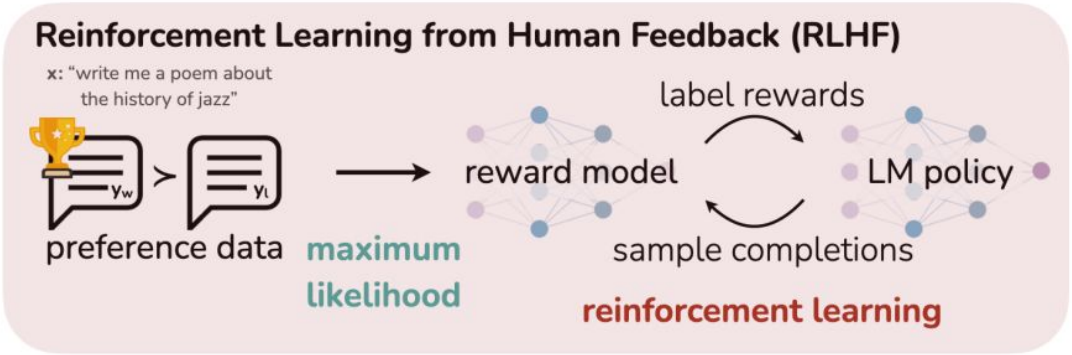



Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right].$$

Rafailov et al. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. NeurIPS 2023. https://arxiv.org/abs/2305.18290

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}}\Bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Bigg]\Bigg],$$

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

Rafailov et al. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. NeurIPS 2023. https://arxiv.org/abs/2305.18290

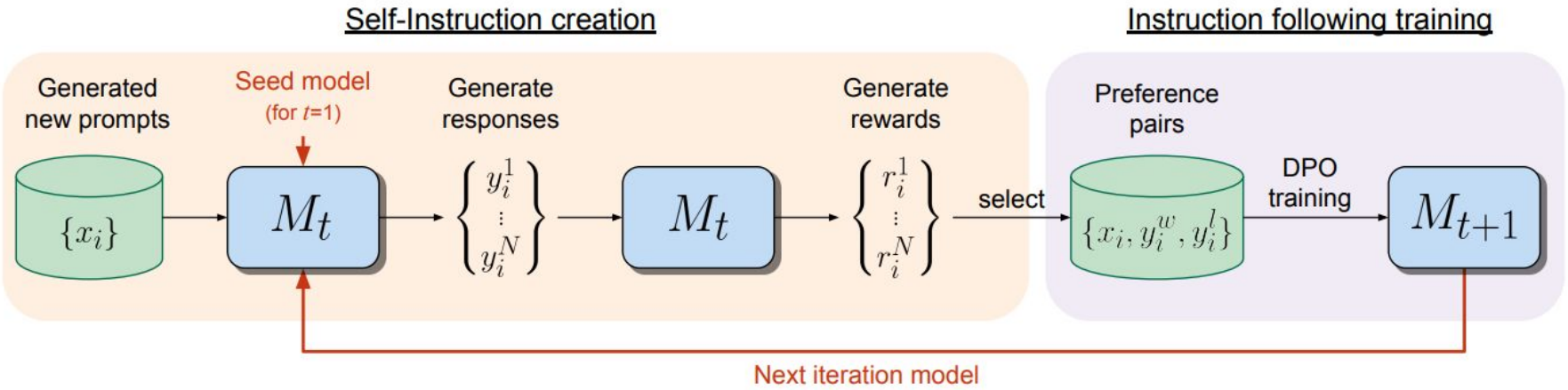



Figure 1: **Self-Rewarding Language Models.** Our self-alignment method consists of two steps: (i) *Self-Instruction creation*: newly created prompts are used to generate candidate responses from model $M_t$, which also predicts its own rewards via LLM-as-a-Judge prompting. (ii) *Instruction following training*: preference pairs are selected from the generated data, which are used for training via DPO, resulting in model $M_{t+1}$. This whole procedure can then be iterated resulting in both improved instruction following and reward modeling ability.

Yuan et al. (2024). Self-Rewarding Language Models. ICML 2024.
https://arxiv.org/abs/2401.10020

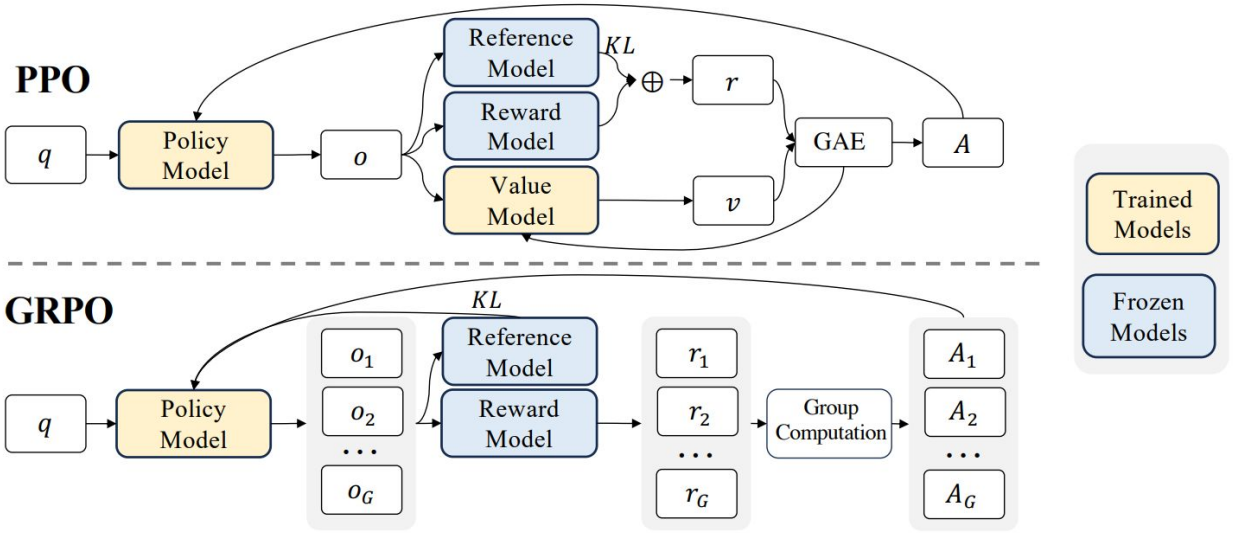



Figure 4 | Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\},$$

Shao, Z., et al. (2024). DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. https://arxiv.org/abs/2402.03300

1. Outcome supervision

$$\{o_1, o_2, \cdots, o_G\}$$

$$\mathbf{r} = \{r_1, r_2, \cdots, r_G\}$$

$$\hat{A}_{i,t} = \widetilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

2. Process supervision

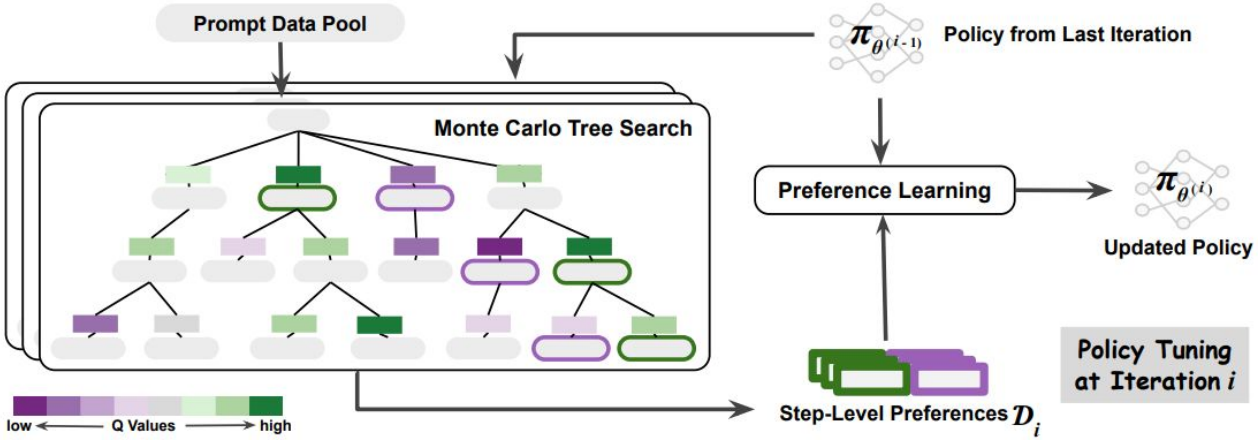
$$\mathbf{R} = \{\{r_1^{index(1)}, \cdots, r_1^{index(K_1)}\}, \cdots, \{r_G^{index(1)}, \cdots, r_G^{index(K_G)}\}\}$$

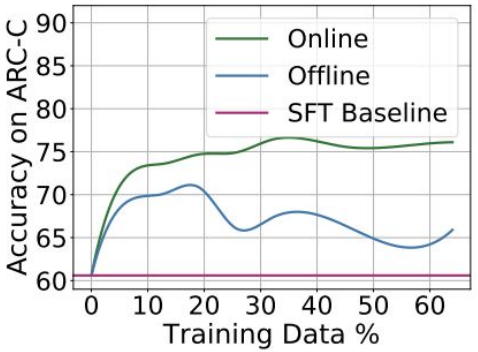
$$\widetilde{r}_i^{index(j)} = \frac{r_i^{index(j)} - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$$

$$\hat{A}_{i,t} = \sum_{index(j) \geq t} \widetilde{r}_i^{index(j)}$$

# Ograniczenia rozwiązań bazujących wyłącznie na uczeniu ze wzmocnieniem

1. Wymaga danych do nadzorowanego pretreningu
2. Ograniczona różnorodność generowanych ścieżek rozwiązań
3. Nieefektywność przeszukiwań




Xie et al. (2024). Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning. NeurIPS 2024. https://arxiv.org/abs/2405.00451

# 1. Expansion

$$R(s_t) = \mathcal{O}(s_t) + \mathcal{C}(s_t)$$

$$\mathcal{C}(s_t) = \pi_\theta(\mathrm{A} \mid \mathrm{prompt}_{\mathrm{eval}}, x, s_t)$$

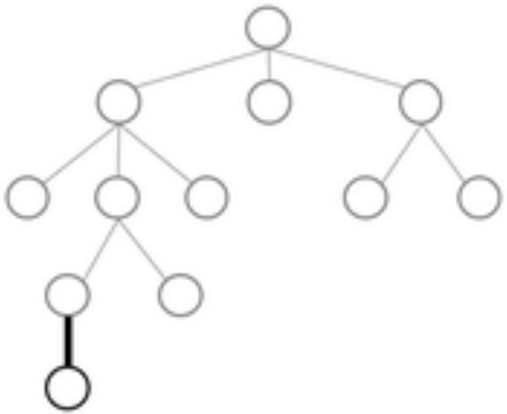


Table 6: Evaluation Prompt Template. The text underlined will be replaced with content from different examples.

**QUESTION:** Which of the following is an example of the formation of a mixture?
Answer Choices: (A) rust forming on an iron nail (B) sugar crystals dissolving in water (C) sodium and chlorine forming table salt (D) hydrogen and oxygen reacting to produce water

**EXAMPLE ANSWER:** The answer is (B) sugar crystals dissolving in water

**PROPOSED SOLUTION:** The formation of a mixture occurs when two or more substances are combined together without changing their individual properties. In the given options, rust forming on an iron nail is an example of the formation of a mixture. The iron nail and the oxygen in the air combine to form iron oxide, which is a mixture. The answer is A.

**QUESTION**: Evaluate if the proposed solution is logically heading in the correct direction.
Provide an answer of (A) correct or (B) incorrect.

**ANSWER**: The answer is

$$R(s_t) = \mathcal{O}(s_t) + \mathcal{C}(s_t)$$

$$\mathcal{C}(s_t) = \pi_\theta(\mathrm{A} \mid \mathrm{prompt}_{\mathrm{eval}}, x, s_t)$$

Table 3: Ablation of "EXAMPLE ANSWER" in self-evaluation on GSM8K, MATH, and ARC-C. We report AUC and accuracy (%) to compare the discriminative abilities of self-evaluation scores.

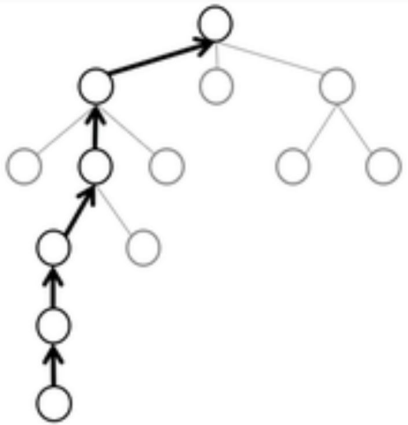
| **Approach** | **GSM8K** | | **MATH** | | **ARC-C** | |
|---|---|---|---|---|---|---|
| | **AUC** | **Accuracy** | **AUC** | **Accuracy** | **AUC** | **Accuracy** |
| w/ example answer | 74.7 | 72.5 | 76.6 | 48.8 | 65.2 | 57.5 |
| w/o example answer | 62.0 | 69.5 | 48.1 | 42.3 | 55.8 | 48.4 |

# 2. Backpropagation

$$Q(s_t, a) \leftarrow r(s_t, a) + \gamma V(s_{t+1})$$

$$V(s_t) \leftarrow \sum_a N(s_{t+1}) Q(s_t, a) / \sum_a N(s_{t+1})$$

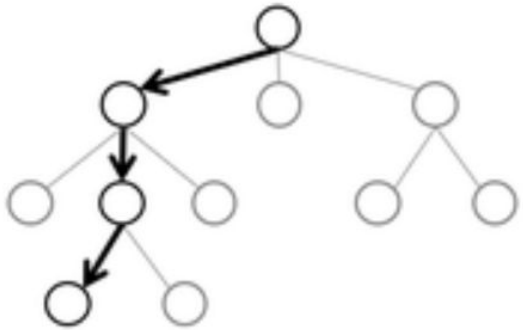
$$N(s_t) \leftarrow N(s_t) + 1$$



# 3. Selection

$$s_{t+1}{}^{*} = \arg\max_{s_t} \left[ Q(s_t, a) + c_{\text{puct}} \cdot p(a \mid s_t) \frac{\sqrt{N(s_t)}}{1 + N(s_{t+1})} \right]$$

$$p(a \mid s_t) = \pi_\theta(a \mid x, s_t) / |a|^{\lambda}$$



**Theorem 3.1** (Offline setting can fail with high probability). *Let $\pi$ be any distribution for which there exists $\bar{y} \in Y$ such that $\pi(\bar{y} \mid x), \pi(\bar{y} \mid x, y) \leq \epsilon$ for all $y \in (Y \setminus \bar{y}) \cup \{y^*\}$ and $\pi_{\theta^{(i-1)}}(\bar{y} \mid x) \geq c$ for some $i \in \{1, 2, \cdots, M\}$. Set $\pi^{(i)} = \pi$ for all $i \in \{1, 2, \cdots, M\}$. Then, there exists $\theta \in \Theta$ such that with probability at least $1 - 2\epsilon M$ (over the samples of $\pi^{(i)} = \pi$), the following holds: $\pi_\theta(y^* \mid x) \leq 1 - c$.*

If the current policy and the sampling policy differ too much, it is possible that $\epsilon = 0$ and $c \approx 1.0$, for which Theorem 3.1 can conclude $\pi_\theta(y^* \mid x) \approx 0$ with probability 1 for any number of steps $M$.

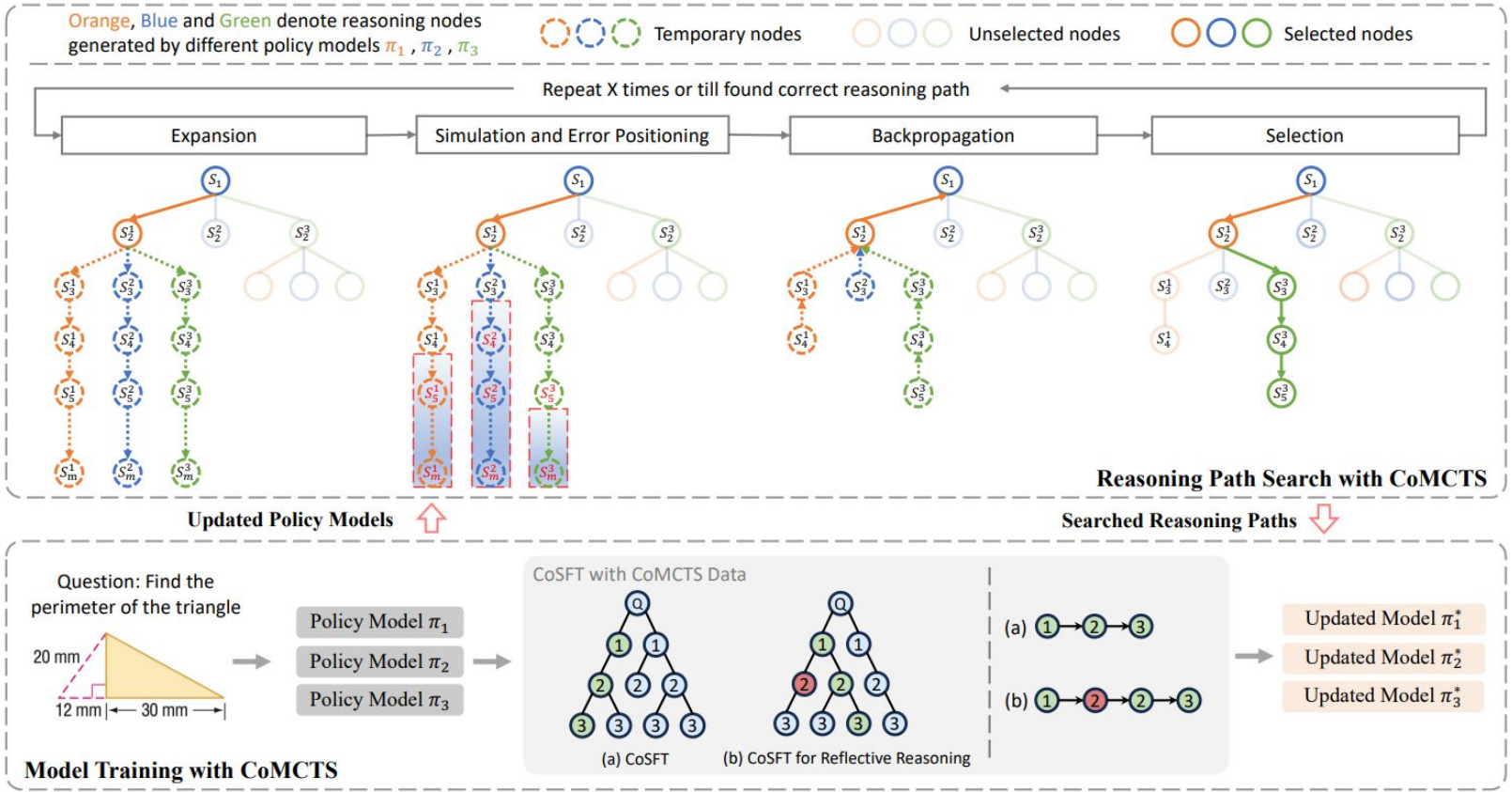
**Theorem 3.2** (Online setting can avoid offline failure case). *Let* $\pi^{(i)} = \pi_{\theta^{(i-1)}}$. *Then, for any* $\theta \in \Theta$, *it holds that* $\pi_\theta(y^* \mid x) = 1$ *if* $M \geq n + 1$.

Table 7: Pass Rates when Ablating MCTS Settings. SE represents the guidance from self-evaluation.

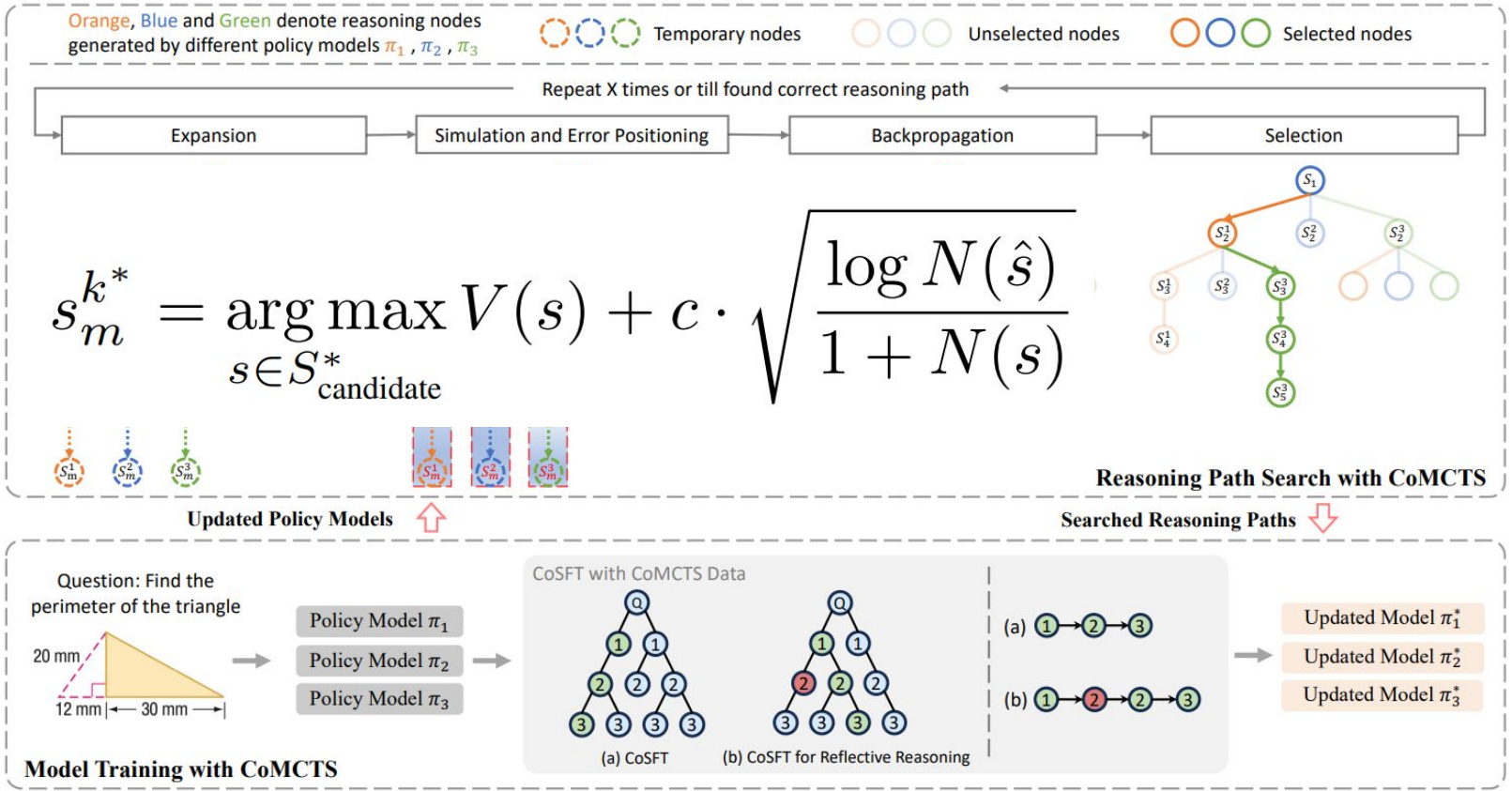
| Decoding Strategy | After Learning | ARC-C | AI2Sci-M | GSM8K |
|---|---|---|---|---|
| Greedy Decoding | ✗ | 60.6 | 70.9 | 75.9 |
| | ✓ | 76.4$\uparrow_{16.4}$ | 88.2$\uparrow_{17.3}$ | 80.7$\uparrow_{5.2}$ |
| MCTS w/o SE | ✗ | 82.5 | 87.3 | 84.4 |
| | ✓ | 91.0$\uparrow_{8.5}$ | 96.1$\uparrow_{9.8}$ | 89.0$\uparrow_{5.6}$ |
| MCTS | ✗ | 83.0 | 90.5 | 85.8 |
| | ✓ | 92.1$\uparrow_{9.1}$ | 97.3$\uparrow_{6.8}$ | 90.2$\uparrow_{4.4}$ |

Table 1: Result comparison (accuracy %) on arithmetic tasks. We supervised fine-tune the base model Mistral-7B on Arithmo data, while Math-Shepherd (Wang et al., 2023a) use MetaMATH (Yu et al., 2023b) for SFT. We highlight the advantages of our approach via conceptual comparison with other methods, where NR, OG, OF, and NS represent "w/o Reward Model", "On-policy Generation", "Online Feedback", and "w/ Negative Samples".

| **Approach** | **Base Model** | **Conceptual Comparison** | | | | **GSM8K** | **MATH** |
|---|---|---|---|---|---|---|---|
| | | NR | OG | OF | NS | | |
| LMSI | PaLM-540B | ✓ | ✓ | ✗ | ✗ | 73.5 | – |
| SFT (MetaMath) | Mistral-7B | – | – | – | – | 77.7 | 28.2 |
| Math-Shepherd | | ✗ | ✓ | ✗ | ✓ | **84.1** | 33.0 |
| SFT (Arithmo) | Mistral-7B | – | – | – | – | 75.9 | 28.9 |
| MCTS Offline-DPO | | ✓ | ✗ | ✗ | ✓ | 79.9 | 31.9 |
| Instance-level Online-DPO | | ✓ | ✓ | ✓ | ✓ | 79.7 | 32.9 |
| Ours | | ✓ | ✓ | ✓ | ✓ | 80.7 | 32.2 |
| Ours (w/ G.T.) | | ✓ | ✓ | ✓ | ✓ | 81.8 | **34.7** |

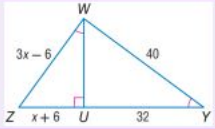



Yao et al. (2024). Mulberry: Empowering MLLM with o1-like Reasoning and Reflection via Collective Monte Carlo Tree Search. NeurIPS 2025.
https://arxiv.org/abs/2412.18319




Yao et al. (2024). Mulberry: Empowering MLLM with o1-like Reasoning and Reflection via Collective Monte Carlo Tree Search. NeurIPS 2025.
https://arxiv.org/abs/2412.18319

Orange, Blue and Green denote reasoning nodes generated by different policy models $\pi_1$ , $\pi_2$ , $\pi_3$




**Question:**
**Calculate the length of ZU**

**GT Answer: 18**

***Level 1***

The figure shows a triangle named WZY, with W at the top and Z and Y on a horizontal line at the bottom..

The diagram shows a right triangle ∠ WYZ, where WU is perpendicular to the base ZY. The hypotenuse WY…

***Level 2***

We have a triangle WZY with an altitude WU drawn to the base ZY. This altitude creates two right triangles: WUZ (on the left)…

By dropping a perpendicular WU to the base ZY, two right triangles.

The figure is a right triangle with the right angle at W…

The problem involves finding UZ, labeled as x + 8. This requires using the geometric properties of right triangles, specifically the…

***Level 3***

Step1: Apply Pythagorean theorem to ∠ WUZ : $(3x - 6)^2 = WU^2 + (x + 6)^2$

**Reflection**

Step1: Use the Pythagorean theorem in ∠ WUZ : $WU^2 = (3x - 6)^2 + (x + 6)^2$

Step1: Apply the geometric mean property of right triangles: In a right triangle, the altitude to …

Step 1: Label the known lengths:- WZ = 3x - 6 \n- WY = 40 \n- ZU = x + 6 \n- UY = 32 .

⋮ ⋮

***Level N***

The final answer is 22

The final answer is 24

The final answer is **18**

The final answer is 24

The final answer is 12

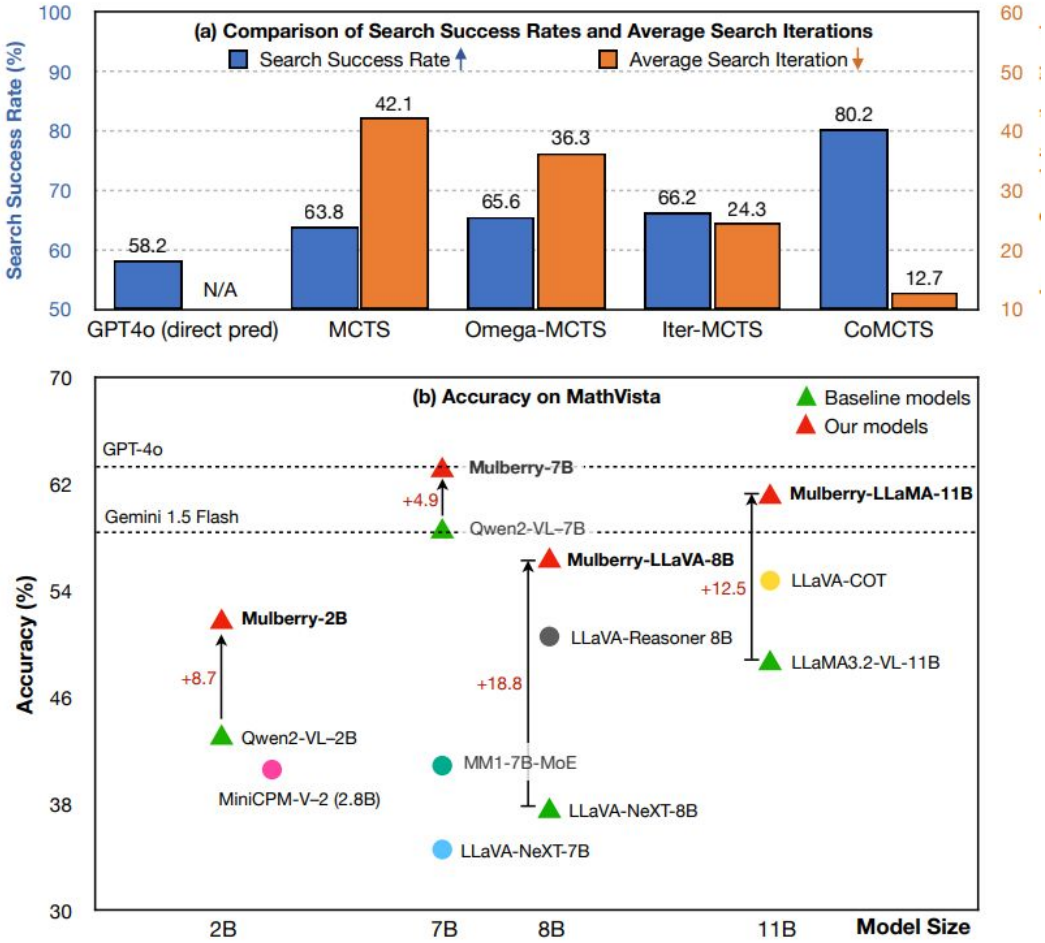
The final answer is 12

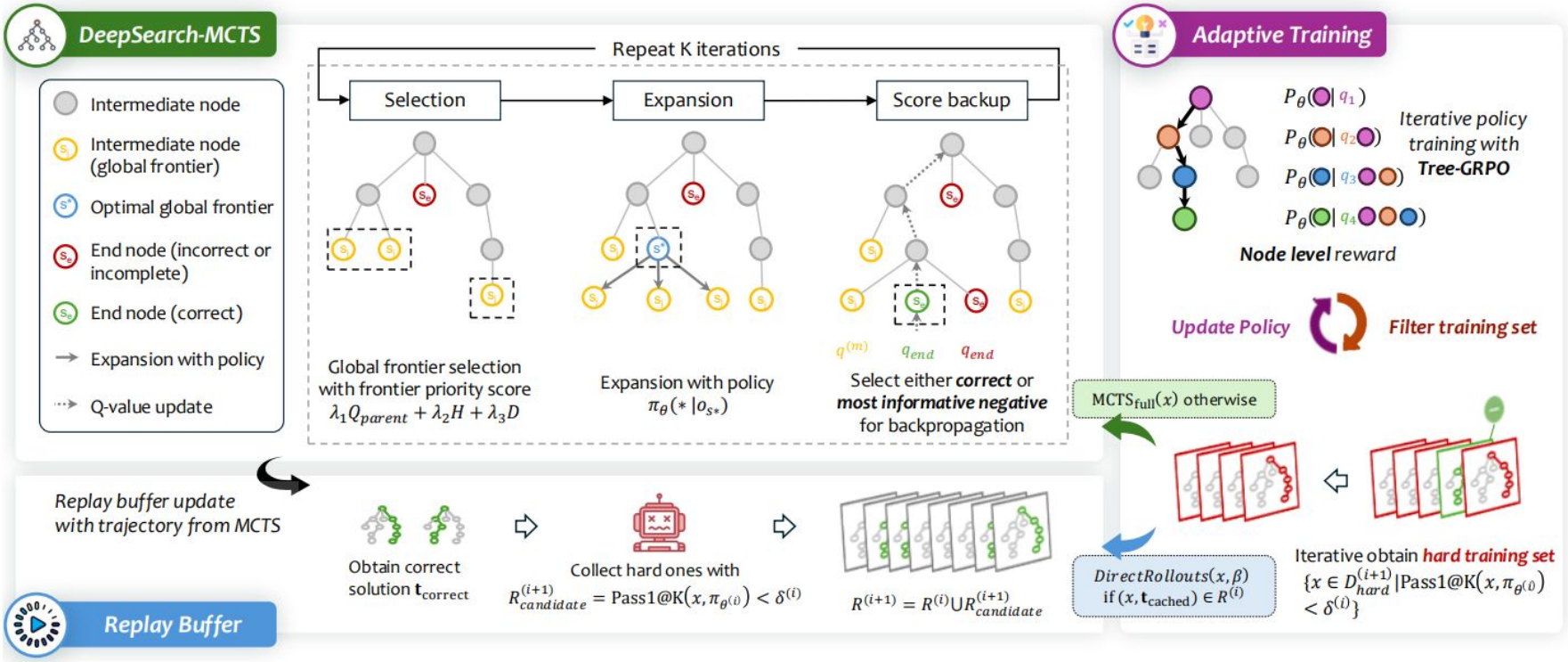
The final answer is 32

The final answer is 16

| Direct Pred | CoMCTS | | | | S.S.R. |
|---|---|---|---|---|---|
| GPT-4o | GPT-4o | Qwen2-VL-7B | LLama3.2-11B | Qwen2-VL-72B | |
| ✔ | | | | | 58.2 |
| | ✔ | | | | 63.8 |
| | ✔ | ✔ | | | 66.2 |
| | ✔ | ✔ | ✔ | | 69.7 |
| | ✔ | ✔ | ✔ | ✔ | 80.2 |

Table 2: **Ablation Study on CoMCTS.** We study how each model in CoMCTS collective learning contribute to overall tree search performance in Search Success Rate (S.S.R.).


(a) Comparison of Search Success Rates and Average Search Iterations
Search Success Rate
Average Search Iteration
Search Success Rate (%)
58.2
N/A
63.8
42.1
65.6
36.3
66.2
24.3
80.2
12.7
GPT4o (direct pred)
MCTS
Omega-MCTS
Iter-MCTS
CoMCTS
(b) Accuracy on MathVista
Baseline models
Our models
GPT-4o
Gemini 1.5 Flash
Mulberry-7B
+4.9
Qwen2-VL-7B
Mulberry-LLaMA-11B
Mulberry-LLaVA-8B
+12.5
LLaVA-COT
Mulberry-2B
LLaVA-Reasoner 8B
+8.7
+18.8
LLaMA3.2-VL-11B
Qwen2-VL-2B
MM1-7B-MoE
MiniCPM-V-2 (2.8B)
LLaVA-NeXT-8B
LLaVA-NeXT-7B
Accuracy (%)
2B
7B
8B
11B
Model Size




Wu et al. (2025). DeepSearch: Overcome the Bottleneck of Reinforcement Learning with Verifiable Rewards via Monte Carlo Tree Search.
https://www.arxiv.org/abs/2509.25454

The q-value initialization is $q^{(0)}(s_i) = 0$ for all $s_i \in \mathcal{T}$. Terminal node rewards are assigned according to the verification function's result:

$$q(s_{\text{end}}) = \begin{cases} +1 & \text{if } \mathcal{V}(s_{\text{end}}) = 1 \text{ (correct)}, \\ -1 & \text{if } \mathcal{V}(s_{\text{end}}) = 0 \text{ (incorrect)} \vee d(s_{\text{end}}) < d_{\mathcal{T}} \text{ (incomplete)}. \end{cases} \tag{6}$$

To ensure positive q-values (*e.g.*, $q_{\text{correct}} = 0.1$) for nodes on correct reasoning paths while penalizing nodes leading to incorrect or incomplete solutions, we enforce the constrained update rule:

$$q^{(m)}(s_i) = \begin{cases} q^{(m-1)}(s_i) + \gamma(i,l) \cdot q^{(m)}(s_{\text{end}}) & \text{if } q^{(m-1)}(s_i) \cdot q^{(m)}(s_{\text{end}}) \geq 0, \\ \gamma(i,l) \cdot q^{(m)}(s_{\text{end}}) & \text{elif } q^{(m)}(s_{\text{end}}) > 0, \\ q^{(m-1)}(s_i) & \text{elif } q^{(m-1)}(s_i) > 0. \end{cases} \tag{7}$$

Wu et al. (2025). DeepSearch: Overcome the Bottleneck of Reinforcement Learning with Verifiable Rewards via Monte Carlo Tree Search.
https://www.arxiv.org/abs/2509.25454

For each frontier node $s \in \mathcal{F}$, we compute a *frontier priority score*:

$$F(s) = \underbrace{\lambda_1 \times \tanh(Q_{\text{parent}}(s))}_{\text{Quality Potential}} + \underbrace{\lambda_2 \times H(\pi_\theta(s|o))}_{\text{Uncertainty Bonus}} + \underbrace{\lambda_3 \times D(d(s))}_{\text{Depth Bonus}}. \tag{10}$$

Wu et al. (2025). DeepSearch: Overcome the Bottleneck of Reinforcement Learning with Verifiable Rewards via Monte Carlo Tree Search.
https://www.arxiv.org/abs/2509.25454

# Tree-GRPO

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathbb{T}\sim\mathcal{T},\mathbf{t}^i\sim\mathbb{T},(s_j,o_j)\sim\mathbf{t}^i} \frac{1}{|s_j|} \sum_{k=1}^{|s_j|} \min\left(\rho_{j,k}(\theta)\hat{A}_{j,k}, \text{clip}\left(\rho_{j,k}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{A}_{j,k}\right)$$

$$\rho_{j,k}(\theta) = \frac{\pi_\theta(a_{j,k}|o_j, a_{j,<k})}{\pi_{\theta_{\text{old}}}(a_{j,k}|o_j, a_{j,<k})}$$

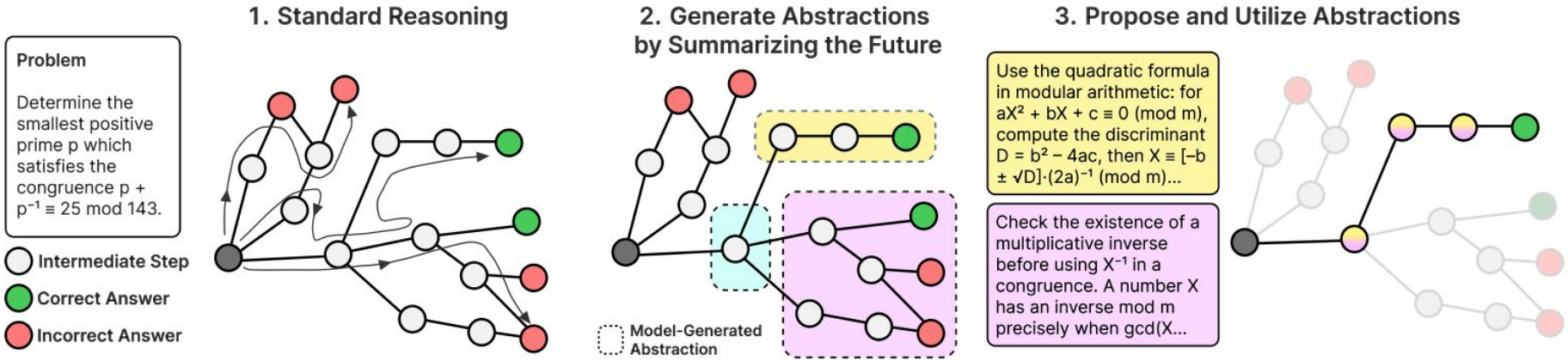
**Q-Value Soft Clipping.** To address the q-value explosion problem for intermediate nodes while preserving meaningful gradients, we first apply *soft clipping* using the hyperbolic tangent function:

$$q(s_j) = \tanh\left(q^{(k_{\max})}(s_j)/\epsilon_q\right) \cdot q_{\max} \quad \text{for all } s_j \in \mathcal{T} \setminus \mathcal{S}_{\text{end}} \tag{16}$$

where $k_{\max}$ is the maximum rollout iterations, $\epsilon_q = 1.0$ is the temperature parameter, and $q_{\max} = 1$ defines the maximum allowable q-value magnitude.

$$\hat{A}_{j,k} = q(s_j) - \mu_{\mathbf{t}},$$

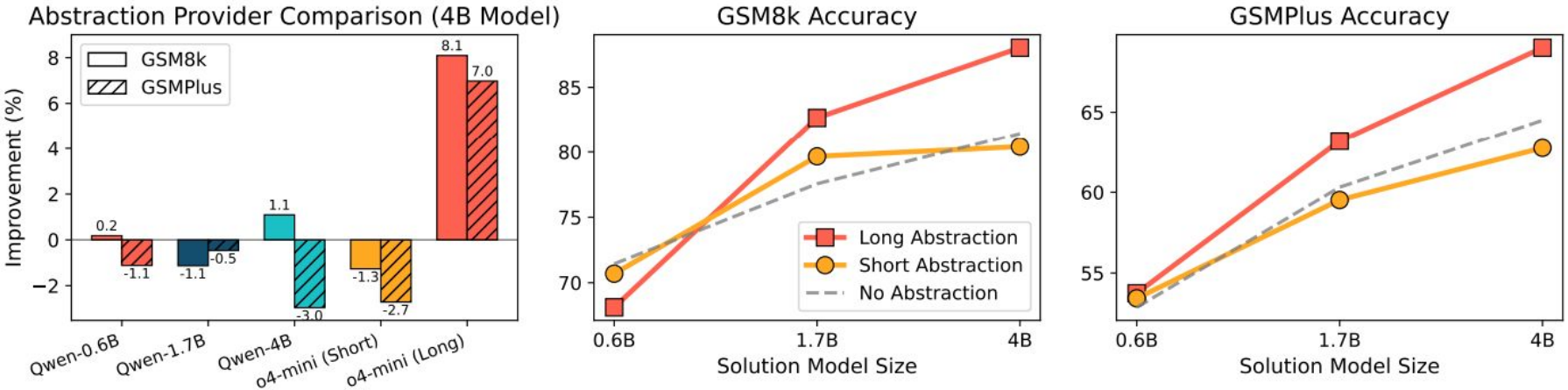
where $\mu_{\mathbf{t}}$ is the average reward of the terminal nodes $\mathcal{S}_{\text{end}}$ throughout the tree $\mathbb{T}$




Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263

$$\mathbb{E}_{\widetilde{\mathbf{y}} \sim \pi_\theta^{\text{sol}}(\cdot|\mathbf{x},\mathbf{z})} \left[ \text{Acc}(\widetilde{\mathbf{y}}, \mathbf{y}^*) \right] > \mathbb{E}_{\widetilde{\mathbf{y}} \sim \pi_\theta^{\text{sol}}(\cdot|\mathbf{x})} \left[ \text{Acc}(\widetilde{\mathbf{y}}, \mathbf{y}^*) \right].$$

Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263




Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263

| k | pass@k | | coverage (max@k) | |
|---|---|---|---|---|
| | w/o abs | w/ abs | w/o abs | w/ abs |
| 1 | 14.0% | **18.0%** | 19.5% | **22.5%** |
| 2 | 17.5% | **22.5%** | 24.0% | **28.5%** |
| 4 | 20.0% | **26.5%** | 28.5% | **34.0%** |
| 8 | 22.8% | **30.2%** | 31.8% | **38.8%** |
| 16 | 24.7% | **33.2%** | 35.0% | **42.9%** |

Table 1: ***pass@k accuracy and max@k coverage on the ARC-AGI benchmark.*** Abstractions yield consistent gains in both metrics.

“To ensure that the abstractions do not “leak” content of the solution, we verify post-hoc that prompting a model with only the abstraction and no problem yields zero accuracy when sampling 16 times from the base model. This makes these abstractions suitable for our study as they only provide useful information while not allowing the model to shortcut to the answer.”

Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263

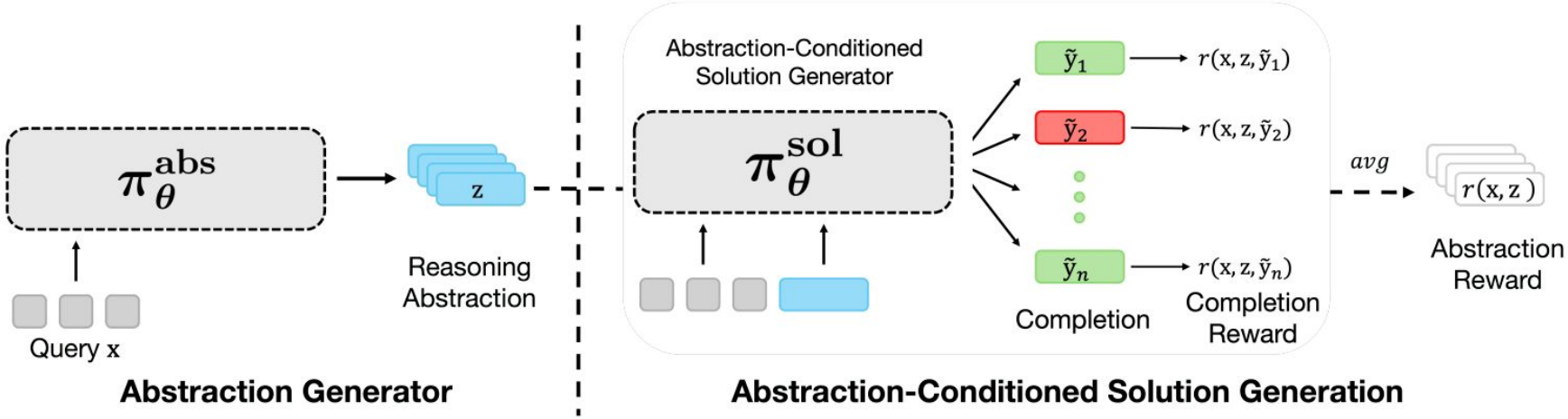
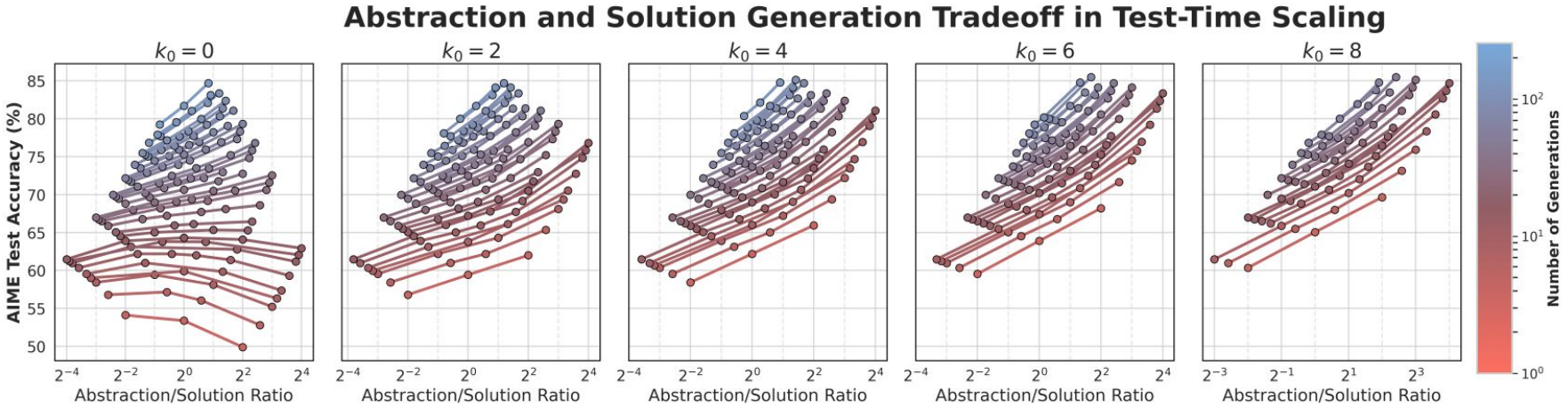
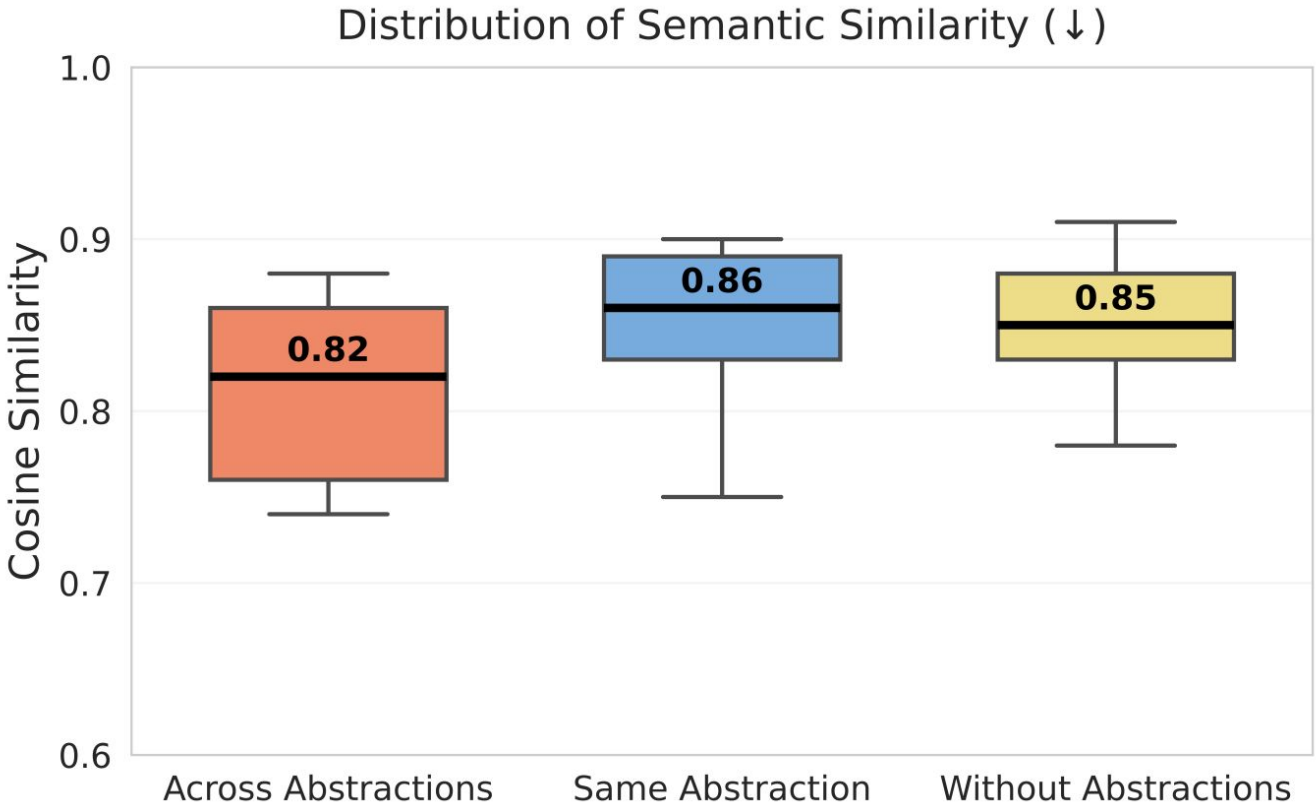



**Figure 3:** ***RLAD training paradigm.*** We train an abstraction generator, $\pi_\theta^{\text{abs}}$, that proposes some reasoning abstractions conditioned on the question $\mathbf{x}$, denoted as $\mathbf{z}$. Then, the solution generator, $\pi_\theta^{\text{sol}}$, is trained to produce a response, $\tilde{\mathbf{y}}$, conditioned on the generated abstraction $\mathbf{z}$. The reward used for training $\pi_\theta^{\text{abs}}$ corresponds to the average success rate of the solution generator conditioned on the proposed abstraction.

Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263




Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263




Qu et al. 2025. RLAD: Training LLMs to Discover Abstractions for Solving Reasoning Problems https://arxiv.org/abs/2510.02263