Learning to Better Search with Language Models via Guided Reinforced Self-Training

Seungyong Moon [1]   Bumsoo Park [2]   Hyun Oh Song [1]

[1]Seoul National University   [2]KRAFTON

TL;DR

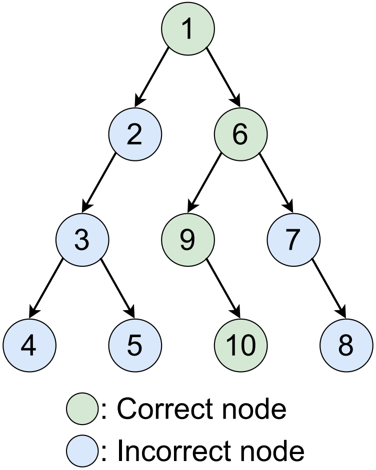
We propose a novel fine-tuning algorithm that enhances the search capability of language models through guided data generation.

Stream of search (SoS)

- An optimal solution $S = (s_1, \ldots, s_n)$, where s_i is a single reasoning step.
- A search trace $Z = (z_1, \ldots, z_m)$, where z_i is a tree search operation.
- Train the model π_θ to imitate Z via SFT.

$$\max_\theta \mathbb{E}_{(q,Z)\sim\mathcal{D}} \left[\log \pi_\theta(Z \mid q)\right]$$

- Show better generalization than S.



Training with self-generated data

- Reinforced self-training (ReST)
  - Generate, filter, and fine-tune via SFT over multiple iterations.

$$\max_{\theta} \mathbb{E}_{q \sim \mathcal{D}, Z \sim \pi_\theta(\cdot|q)} \left[\mathbb{1}_{R(Z|q) > \tau} \cdot \log \pi_\theta(Z \mid q) \right]$$

- Reinforcement fine-tuning (RFT)
  - Directly maximize rewards via RL (e.g., PPO or GRPO).

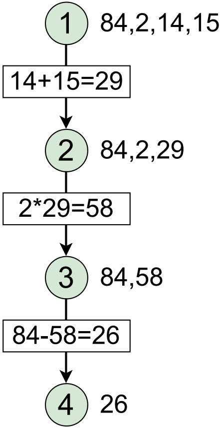
$$\max_{\theta} \mathbb{E}_{q \sim \mathcal{D}, Z \sim \pi_\theta(\cdot|q)} \left[R(Z \mid q) - \beta \cdot D_{\mathrm{KL}}(\pi_\theta(\cdot \mid q) \parallel \pi_{\mathrm{ref}}(\cdot \mid q)) \right]$$

Countdown

- Goal: Combine the input numbers using the four basic arithmetic operations to reach the target number.

- Simple yet challenging: even GPT-4 struggles.

Input: 84,2,14,15 Target: 26
Solution: 84-2*(14+15)=26

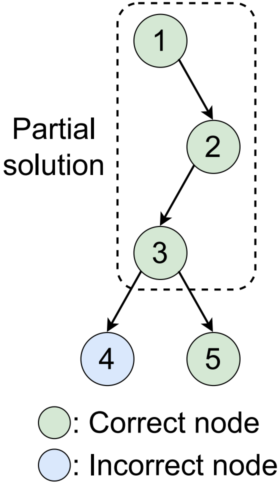


Motivation

- Search traces generalize well but are noisy and suboptimal.
- Consequently, models trained on such traces suffer from inefficient search.
- Can fine-tuning methods like ReST or RFT fundamentally improve search efficiency?
- To address this, we leverage optimal solutions as guidance.

Motivation

- First attempt: Provide the model with partial optimal solutions as hints.
- This significantly improves performance by effectively reducing the search space.
- This motivates us to utilize such high-quality, self-generated traces for fine-tuning.
- However, these traces often have low likelihood under the model distribution.



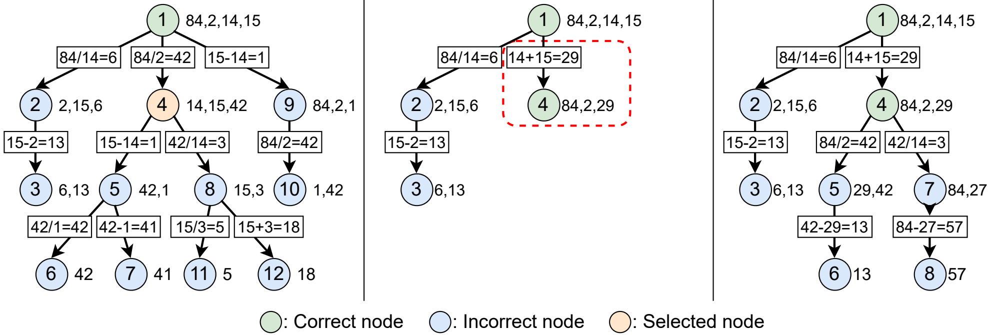
Guided reinforced self-training (Guided-ReST)

Step 1. Select a random child node from the last correct reasoning step.

Input: 84,2,14,15 Target: 26
Solution: 84-2*(14+15)=26

Step 2. Replace the selected node to the next reasoning step and truncate the search trace up to it.

Step 3. Continue the search from the modified node.



It progressively integrates each step of the optimal solution into the search trace, yielding high-quality, high-likelihood traces.

Guided reinforced self-training (Guided-ReST)

- Fine-tune the model on the resulting search traces via SFT.
- Repeat this procedure for multiple iterations.

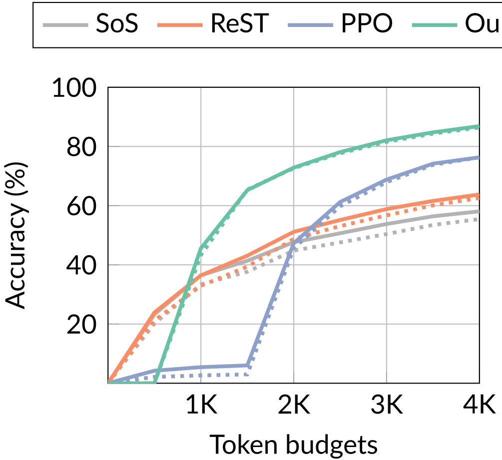
RL fine-tuning

- Apply RFT using PPO on top of Guided-ReST.
- Use operation-level MDP instead of token-level MDP.
    - The log importance ratio is computed as the sum over tokens.
- Remove the KL penalty term, following recent practice.

Application to code self-repair

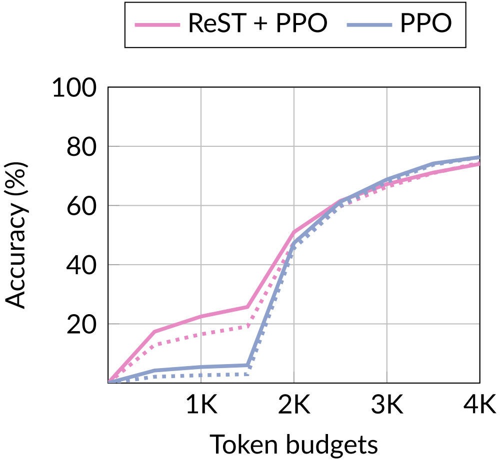
- We extend our method to a more realistic domain: code self-repair.
- Consider episode-level search rather than stepwise search.

Countdown results (Llama-3.2-1B)

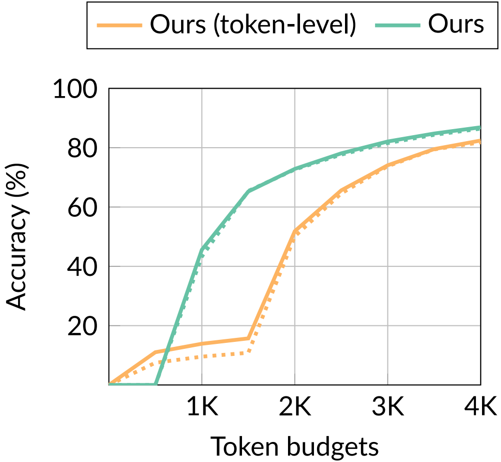


Our method improves the upper bound and achieves 2× token efficiency.

Countdown results (Llama-3.2-1B)



ReST does not benefit from PPO.



Operation-level MDP is essential.

Countdown results (Llama-3.2-1B)

Method	Seen target						Unseen target					
	1	2	4	8	16	32	1	2	4	8	16	32
SoS	55.3	63.8	69.6	73.3	75.5	77.1	53.2	62.3	68.7	73.0	75.6	77.5
ReST	62.3	67.7	71.3	73.6	75.3	76.6	60.8	66.7	70.8	73.5	75.5	77.0
Guided-ReST	**62.7**	**75.3**	**84.6**	**90.8**	**94.7**	**96.8**	**61.0**	**74.1**	**83.8**	**90.3**	**94.3**	**96.8**

Our method achieves higher pass@k accuracy.

Code self-repair results (Qwen2.5-7B)

Method	CodeContests						CodeForces					
	1	2	4	8	16	32	1	2	4	8	16	32
Base	4.5	7.5	11.3	15.7	20.6	25.8	5.5	8.5	12.5	17.2	22.4	27.7
ReST	9.4	13.1	17.0	21.3	25.9	30.4	**9.7**	14.2	19.3	24.8	30.5	35.9
Guided-ReST	**10.5**	**14.8**	**19.4**	**24.1**	**28.9**	**33.9**	**9.7**	**14.5**	**20.2**	**26.2**	**32.0**	**37.6**

Our method generalizes well beyond Countdown.

Conclusion

- Our method significantly improves the search efficiency of language models by leveraging optimal solutions as guidance.
- Extending the approach to broader tasks would be a promising future direction.

Code: https://github.com/snu-mllab/guided-rest