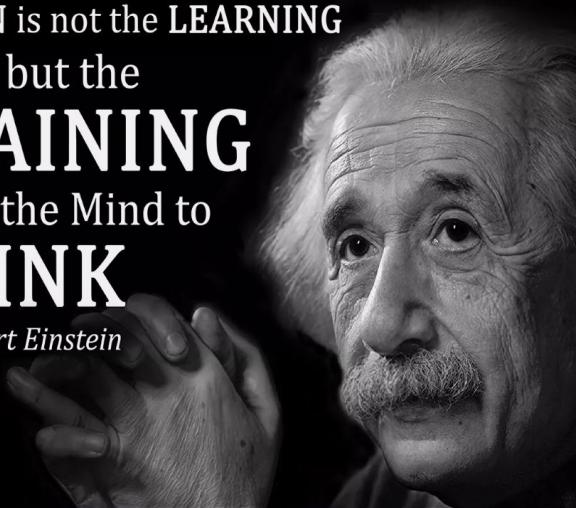


Teaching LLMs to Think

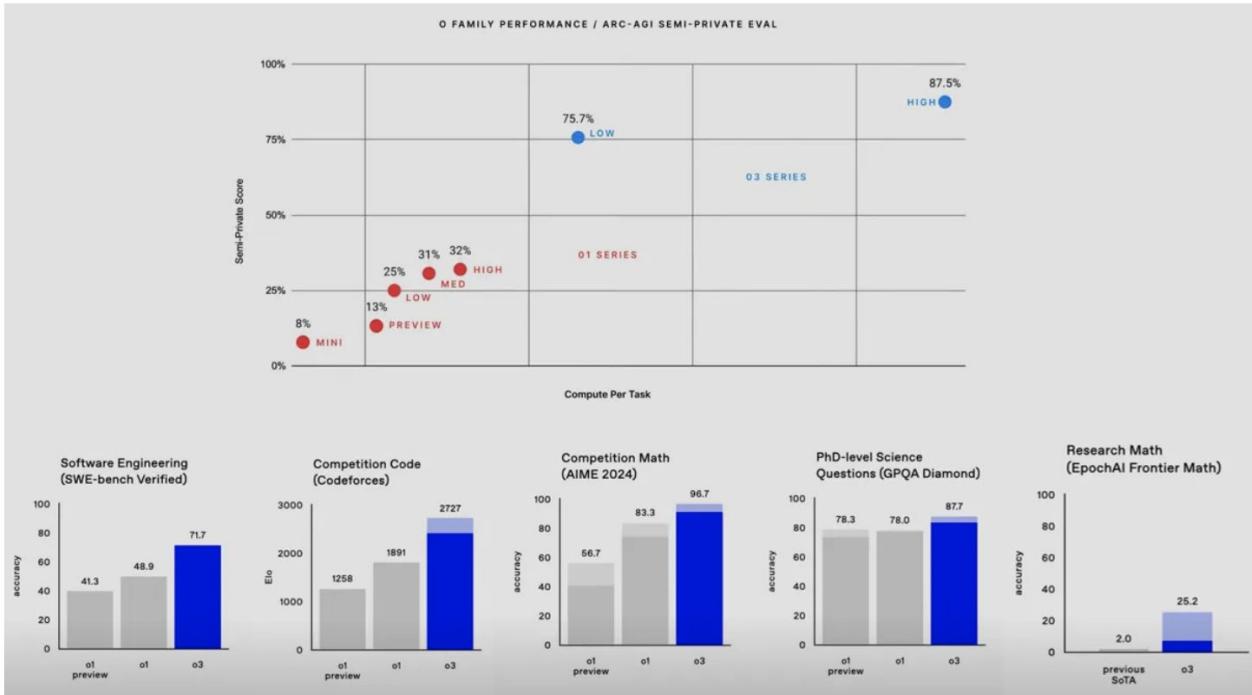
Tianbing Xu Jan. 2025

EDUCATION is not the LEARNING
of Facts but the
TRAINING
of the Mind to
THINK

Albert Einstein



O3 Results in Math, Coding, Science, SWE, AGI (OpenAI 2024)



Source: OpenAI, "[OpenAI o3 and o3-mini—12 Days of OpenAI: Day 12](#)" (12/20/24)

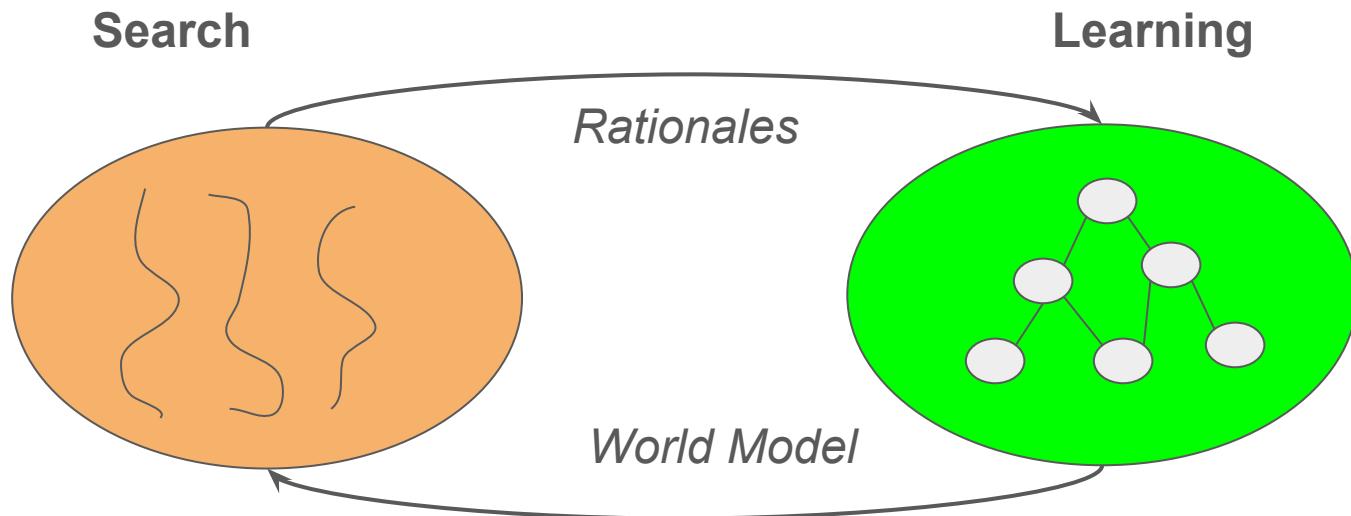
What is Reasoning for AI?



Reasoning = Search (Rationale Space) + Learning (World Model)

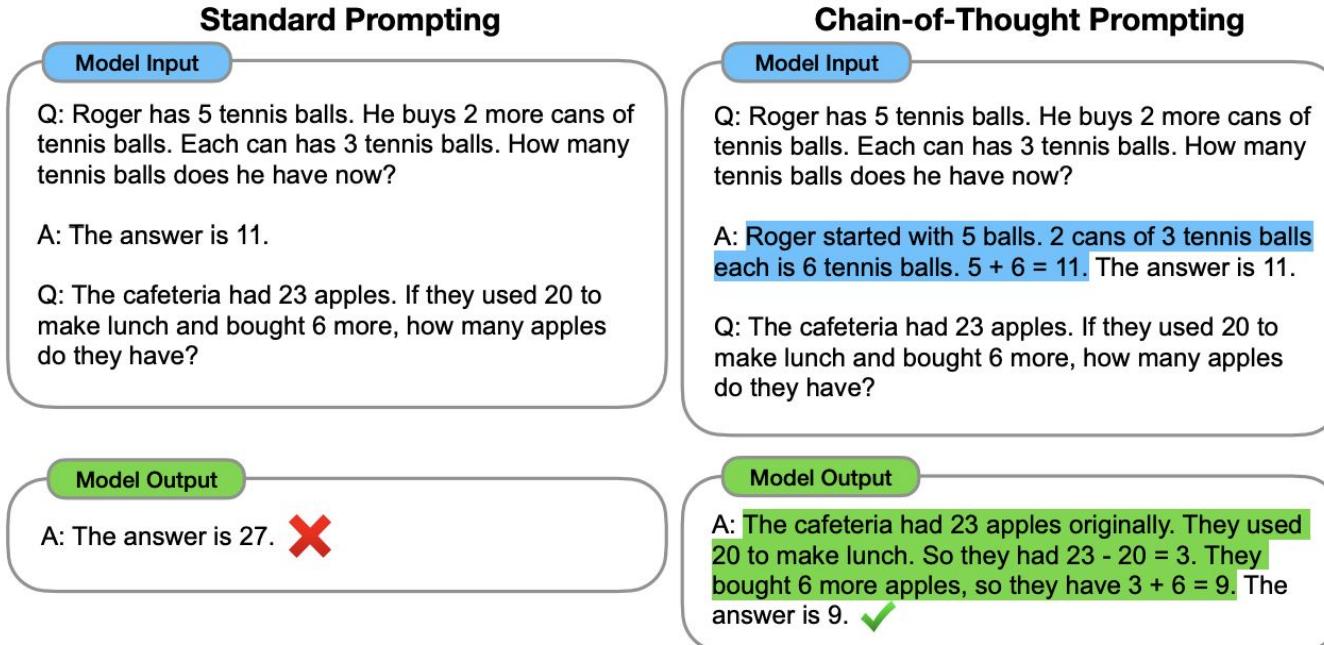
- **Search**
Guided by the World Model, search ***explores*** rationales by navigating the Chain-of-Thoughts (CoT) space or sampling from the probabilistic distribution of rationales to generate plausible explanations.
- **Learning**
The World Model ***evolves*** by refining its understanding based on the generated rationales, improving its ability to guide future searches effectively.

Inter-play of Search and Learning



Rationales (Chain-of-Thoughts)

Chain-of-Thoughts (Wei et. al. NeurIPS 2022)



O1 Chain of Thoughts (OpenAI 2024)

Through reinforcement learning, o1 learns to hone its chain of thought and refine the strategies it uses. It learns to recognize and **correct its mistakes**. It learns to **break down tricky steps into simpler ones**. It learns to **try a different approach** when the current one isn't working.

O1 Chain of Thoughts (OpenAI 2024)



Chain of thought

User

oyfjdnisdr rtqwainr acxz mynzbhhx -> Think step by step

Use the example above to decode:

oyekajzdf aaptcg suaokybhai ouow aqht mynznaatzacdfoulxxz

Thought for 5 seconds ^

First, what is going on here?

We are given:

First, an example:

"oyfjdnisdr rtqwainr acxz mynzbhhx -> Think step by step"

We can see that when "oyfjdnisdr rtqwainr acxz mynzbhhx" is transformed to "Think step by step"

Identifying problem and solution space...

Wait a minute.

I think maybe there is an anagram or substitution cipher here.

Testing of hypotheses...

Idea: Maybe we need to take every other letter or rebuild the plaintext from the ciphertext accordingly.

Let's test this theory.

Identifying most promising path...

Alternatively, maybe the average of the two letters corresponds to 'T'

$(15 + 25)/2 = 20$, which is 'T'

Wait a minute, that seems promising.



Development of hypotheses...

Interesting.

It seems that the ciphertext words are exactly twice as long as the plaintext words.

Rejecting ideas and backtracking...

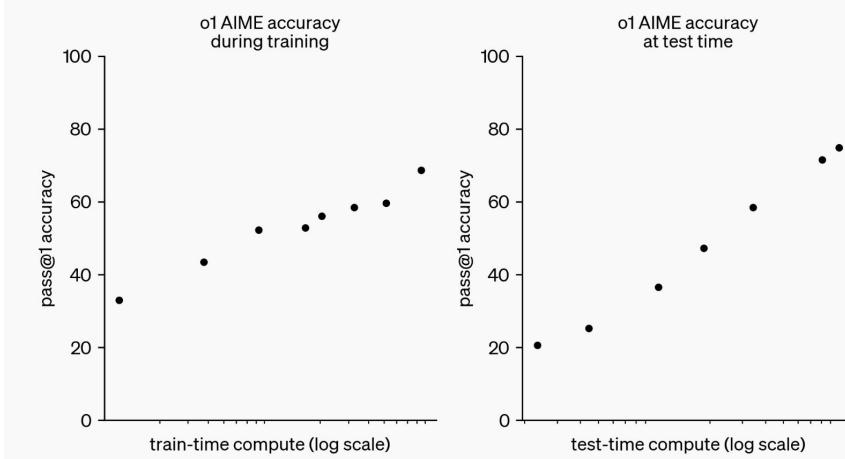
Wait, that's 21 letters.

But in total, we have 22 letters; I think I have a mismatch.

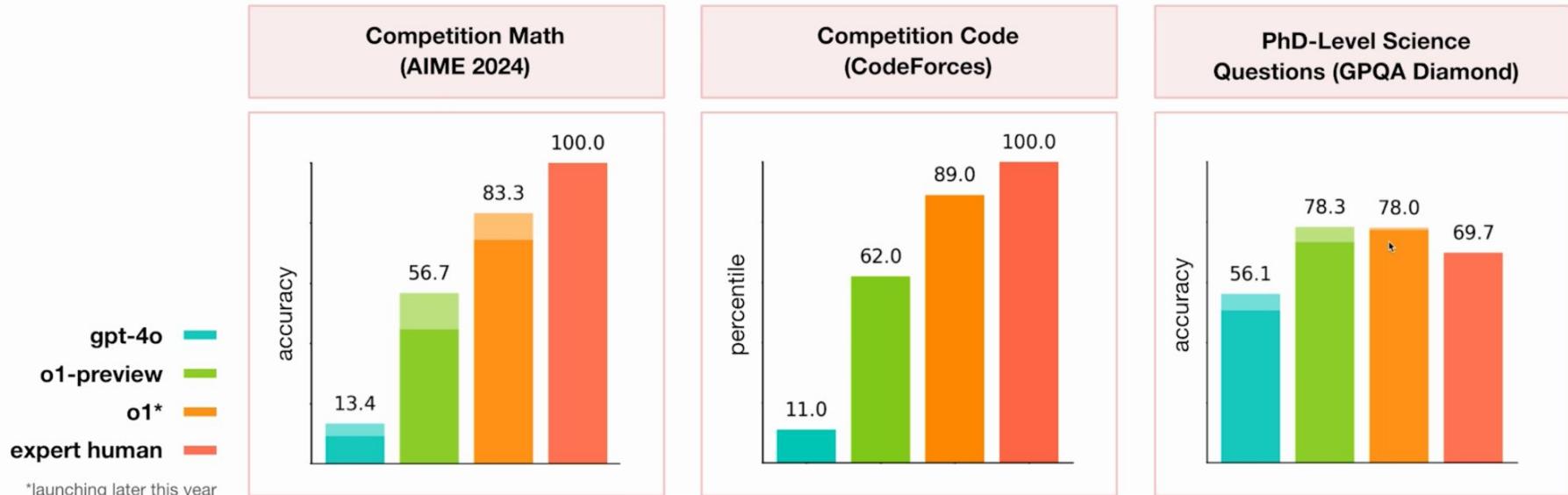
Learning to Reason

Learning to reason with LLMs (OpenAI 2024)

Our large-scale **reinforcement learning** algorithm teaches the model how to think productively using its **chain of thought** in a highly *data-efficient training* process.



O1 Results in Math, Coding, Science (OpenAI 2024)



Self-Taught Reasoner (Zelikman et. al. NeurIPS 2022)

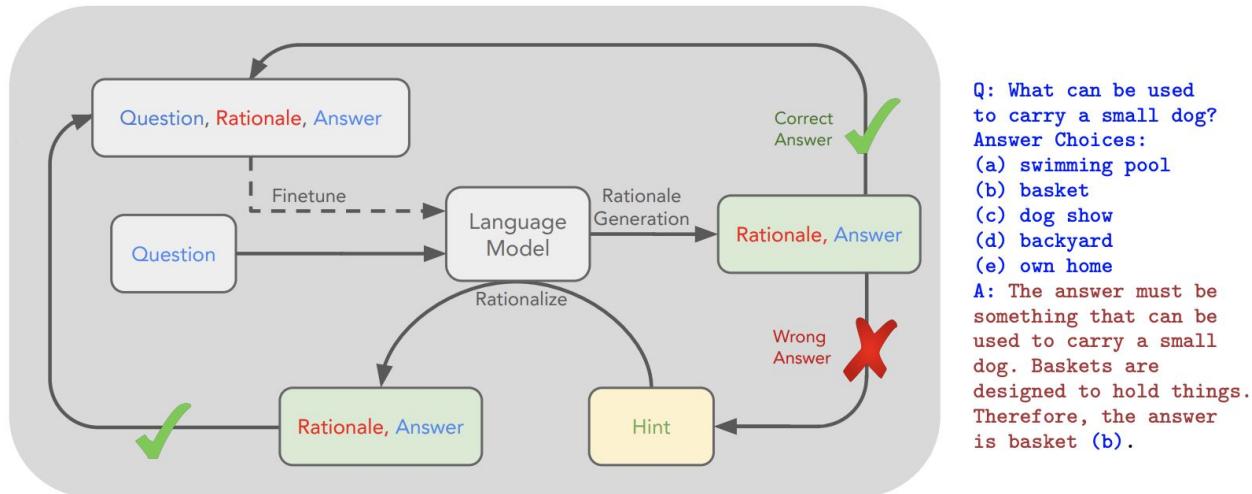


Figure 1: An overview of STaR and a STaR-generated rationale on CommonsenseQA. We indicate the fine-tuning outer loop with a dashed line. The **questions** and ground truth **answers** are expected to be present in the dataset, while the **rationales** are generated using STaR.

STaR Algorithm

Algorithm 1 STaR

Input M : a pretrained LLM; dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^D$ (w/ few-shot prompts)

- 1: $M_0 \leftarrow M$ # Copy the original model
- 2: **for** n **in** $1\dots N$ **do** # Outer loop
- 3: $(\hat{r}_i, \hat{y}_i) \leftarrow M_{n-1}(x_i)$ $\forall i \in [1, D]$ # Perform rationale generation
- 4: $(\hat{r}_i^{\text{rat}}, \hat{y}_i^{\text{rat}}) \leftarrow M_{n-1}(\text{add_hint}(x_i, y_i))$ $\forall i \in [1, D]$ # Perform rationalization
- 5: $\mathcal{D}_n \leftarrow \{(x_i, \hat{r}_i, y_i) \mid i \in [1, D] \wedge \hat{y}_i = y_i\}$ # Filter rationales using ground truth answers
- 6: $\mathcal{D}_n^{\text{rat}} \leftarrow \{(x_i, \hat{r}_i^{\text{rat}}, y_i) \mid i \in [1, D] \wedge \hat{y}_i \neq y_i \wedge \hat{y}_i^{\text{rat}} = y_i\}$ # Filter rationalized rationales
- 7: $M_n \leftarrow \text{train}(M, \mathcal{D}_n \cup \mathcal{D}_n^{\text{rat}})$ # Finetune the original model on correct solutions - inner loop
- 8: **end for**

RL Policy Gradient

STaR can be seen as an approximation to an RL-style policy gradient objective. To see this, note that M can be viewed as a discrete latent variable model $p_M(y \mid x) = \sum_r p(r \mid x)p(y \mid x, r)$; in other words, M first samples a latent rationale r before predicting y . Now, given the indicator reward function $\mathbb{1}(\hat{y} = y)$, the total expected reward across the dataset is

$$J(M, X, Y) = \sum_i \mathbb{E}_{\hat{r}_i, \hat{y}_i \sim p_M(\cdot \mid x_i)} \mathbb{1}(\hat{y}_i = y_i), \quad (1)$$

$$\nabla J(M, X, Y) = \sum_i \mathbb{E}_{\hat{r}_i, \hat{y}_i \sim p_M(\cdot \mid x_i)} [\mathbb{1}(\hat{y}_i = y_i) \cdot \nabla \log p_M(\hat{y}_i, \hat{r}_i \mid x_i)], \quad (2)$$

Expert Iteration Framework (Anthony et. al. NIPS 2017)

The Imitation Learning step is analogous to a human improving their intuition for the task by studying example problems, while the expert improvement step is analogous to a human using their improved intuition to guide future analysis.

Algorithm 1 Expert Iteration

```
1:  $\hat{\pi}_0 = \text{initial\_policy}()$ 
2:  $\pi_0^* = \text{build\_expert}(\hat{\pi}_0)$ 
3: for  $i = 1$ ;  $i \leq \text{max\_iterations}$ ;  $i++$  do
4:    $S_i = \text{sample\_self\_play}(\hat{\pi}_{i-1})$ 
5:    $D_i = \{(s, \text{imitation\_learning\_target}(\pi_{i-1}^*(s))) | s \in S_i\}$ 
6:    $\hat{\pi}_i = \text{train\_policy}(D_i)$ 
7:    $\pi_i^* = \text{build\_expert}(\hat{\pi}_i)$ 
8: end for
```

Reinforced Self-Training (Gulcehre et. al. Deepmind 2023)

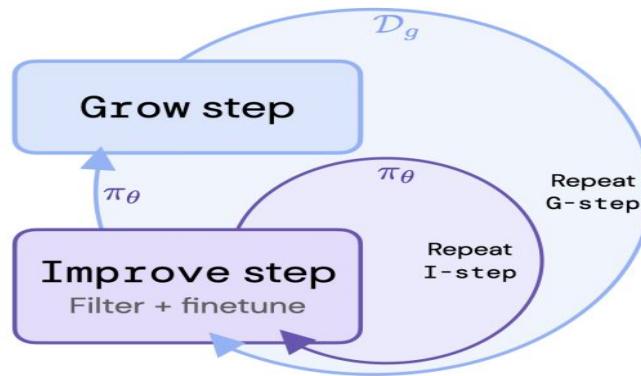


Figure 1 | ReST method. During Grow step, a policy generates a dataset. At Improve step, the filtered dataset is used to fine-tune the policy. Both steps are repeated, Improve step is repeated more frequently to amortise the dataset creation cost.

Learning to Search

Tree-of-Thoughts (Yao et. al. NeurIPS 2023)

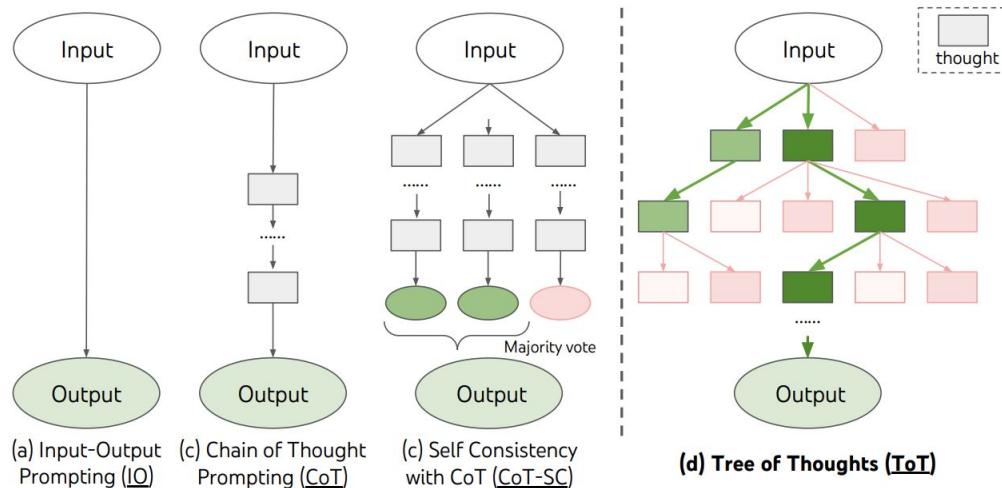


Figure 1: Schematic illustrating various approaches to problem solving with LLMs. Each rectangle box represents a *thought*, which is a coherent language sequence that serves as an intermediate step toward problem solving. See concrete examples of how thoughts are generated, evaluated, and searched in Figures 2,4,6.

Stream of Search (Gandhi et. al. COLM 2024)

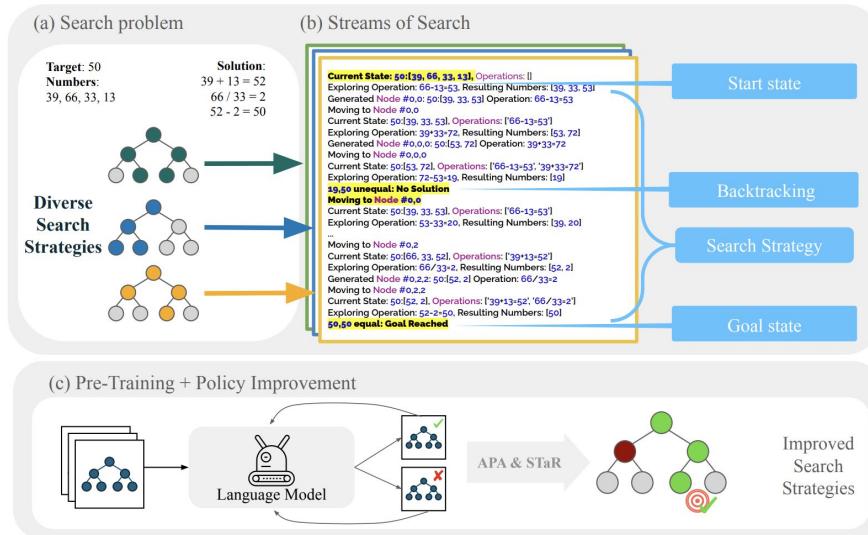


Figure 1: Overview of the Stream of Search (SoS) framework. (a) A search problem in Countdown is instantiated with input numbers and a target number. The input numbers need to be combined with simple arithmetic operations to get to the target. (b) The Stream of Search dataset contains search trajectories generated by diverse search strategies, including exploration and backtracking. (c) The language model is first trained on the SoS dataset and then iteratively improved using policy improvement techniques such as APA and STaR.

Learning to Search

(Exploration, Expansion, Prune, Backtracking, Learning from Mistakes)

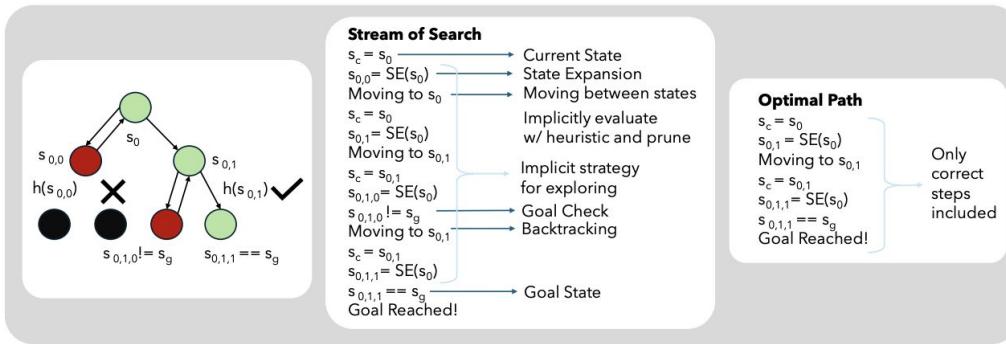
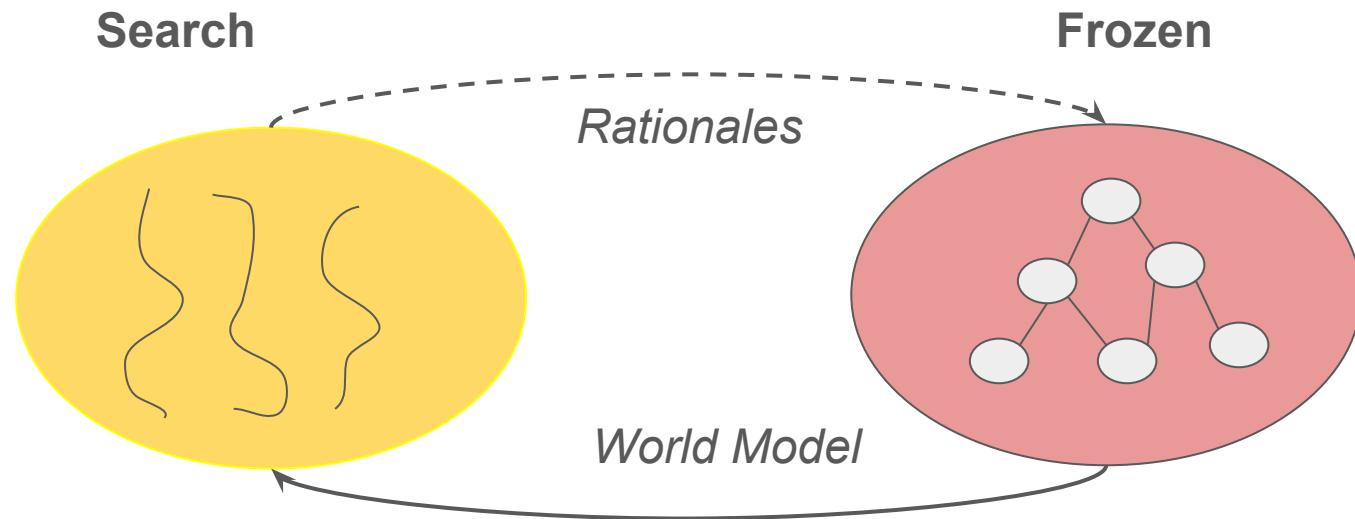


Figure 2: A visualization of how a search process is translated into a stream of search with a language for search. (left) The search process represented as a tree, with different states and operations. The colored states represent the search trajectory \mathcal{T} , the green states represent the correct path to the goal \mathcal{P} and the arrows represent transitions between the states. The black circles represent unexplored states. (center) The search process serialized as text to create a stream of search. The labels specify the different components of the process. See Fig. 1b for how this is realized in Countdown. (right) The optimal path, \mathcal{P} , to the goal state. Backtracking, exploration and the messy process of search are excluded.

Inference

Inference = Search (Rationale Space) + Frozen (World Model)



Verifier: PRM

(Lightman et. al. OpenAI, ICLR 2024)

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x) \cdots p_k(x),$$

where each non-constant polynomial $p_i(x)$ is monic with integer coefficients, and cannot be factored further over the integers. Compute $p_1(1) + p_2(1) + \cdots + p_k(1)$.

I notice that the given polynomial has even degree and only even powers of x , so I can try to make a substitution to simplify it.

Let $y = x^4$, then the polynomial becomes $y^2 + 3y - 4$, which is a quadratic equation.

I can factor this quadratic equation as $(y+4)(y-1)$, so the original polynomial is $(x^4 + 4)(x^4 - 1)$.

Now I need to factor each of these factors further, if possible, over the integers.

For the first factor, $x^4 + 4$, I recall the Sophie Germain identity, which says that $a^4 + 4b^4 = (a^2 + 2b^2 + 2ab)(a^2 + 2b^2 - 2ab)$.

Applying this identity with $a = x$ and $b = 1$, I get $x^4 + 4 = (x^2 + 2x + 2)(x^2 - 2x + 2)$.

For the second factor, $x^4 - 1$, I can use the difference of squares formula, which says that $a^2 - b^2 = (a+b)(a-b)$.

Applying this formula with $a = x^2$ and $b = 1$, I get $x^4 - 1 = (x^2 + 1)(x^2 - 1)$.

Then I can apply the difference of squares formula again with $a = x$ and $b = 1$, to get $x^2 - 1 = (x + 1)(x - 1)$.

So the original polynomial is $(x^4 + 4)(x^4 - 1) = (x^2 + 2x + 2)(x^2 - 2x + 2)(x^2 + 1)(x + 1)(x - 1)$.

These are all the irreducible factors over the integers, so $p_1(x) = x^2 + 2x + 2$, $p_2(x) = x^2 - 2x + 2$, $p_3(x) = x^2 + 1$, $p_4(x) = x + 1$, and $p_5(x) = x - 1$.

To find the sum of their values at $x = 1$, I just need to plug in $x = 1$ into each factor and add them up.

I get $p_1(1) = 1^2 + 2 \cdot 1 + 2 = 5$, $p_2(1) = 1^2 - 2 \cdot 1 + 2 = 1$, $p_3(1) = 1^2 + 1 = 2$, $p_4(1) = 1 + 1 = 2$, and $p_5(1) = 1 - 1 = 0$.

So the final answer is $5 + 1 + 2 + 2 + 0 = 10$.

Answer: 10

Let

$$x^8 + 3x^4 - 4 = p_1(x)p_2(x) \cdots p_k(x),$$

where each non-constant polynomial $p_i(x)$ is monic with integer coefficients, and cannot be factored further over the integers. Compute $p_1(1) + p_2(1) + \cdots + p_k(1)$.

This problem is about factoring a polynomial over the integers, which can be a tricky task.

One possible strategy is to look for rational roots of the polynomial, using the rational root theorem.

However, that would involve checking a lot of possible factors of -4 , and there is no guarantee that the polynomial has any rational roots at all.

Another possible strategy is to use some algebraic tricks to simplify the polynomial or reduce its degree.

I notice that the polynomial has only even powers of x , so I can try to substitute $y = x^2$ and see what happens.

Then the polynomial becomes $y^4 + 3y^2 - 4$, which is a quadratic equation in y^2 .

I can try to factor this quadratic equation by looking for two numbers that multiply to -4 and add to 3 .

Those numbers are 4 and -1 , so I can write $y^4 + 3y^2 - 4 = (y^2 + 4)(y^2 - 1)$.

Now I can substitute back $x^2 = y$ and get $x^8 + 3x^4 - 4 = (x^4 + 4)(x^4 - 1)$.

This looks promising, because I have reduced the degree of the factors from 8 to 4 .

I can try to factor each of these factors further, using the difference of squares formula.

I get $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x^2 - 1)$.

I can apply the difference of squares formula again to the last factor and get $x^8 + 3x^4 - 4 = (x^2 + 2)(x^2 - 2)(x^2 + 1)(x + 1)(x - 1)$. Now I have factored the polynomial completely into monic linear and quadratic factors with integer coefficients.

These are the $p_i(x)$'s that the problem is asking for.

To find the sum of their values at $x = 1$, I just need to plug in $x = 1$ into each factor and add them up.

I get $p_1(1) + p_2(1) + \cdots + p_k(1) = (1^2 + 2)(1^2 - 2)(1^2 + 1)(1 + 1)(1 - 1)$.

Simplifying, I get $p_1(1) + p_2(1) + \cdots + p_k(1) = (3)(-1)(2)(2)(0)$.

Multiplying, I get $p_1(1) + p_2(1) + \cdots + p_k(1) = 0$.

Answer: 0

Figure 2: Two solutions to the same problem, graded by the PRM. The solution on the left is correct while the solution on the right is incorrect. A green background indicates a high PRM score, and a red background indicates a low score. The PRM correctly identifies the mistake in the incorrect solution.

Test-Time Compute (Snell et. al. Deepmind 2024)

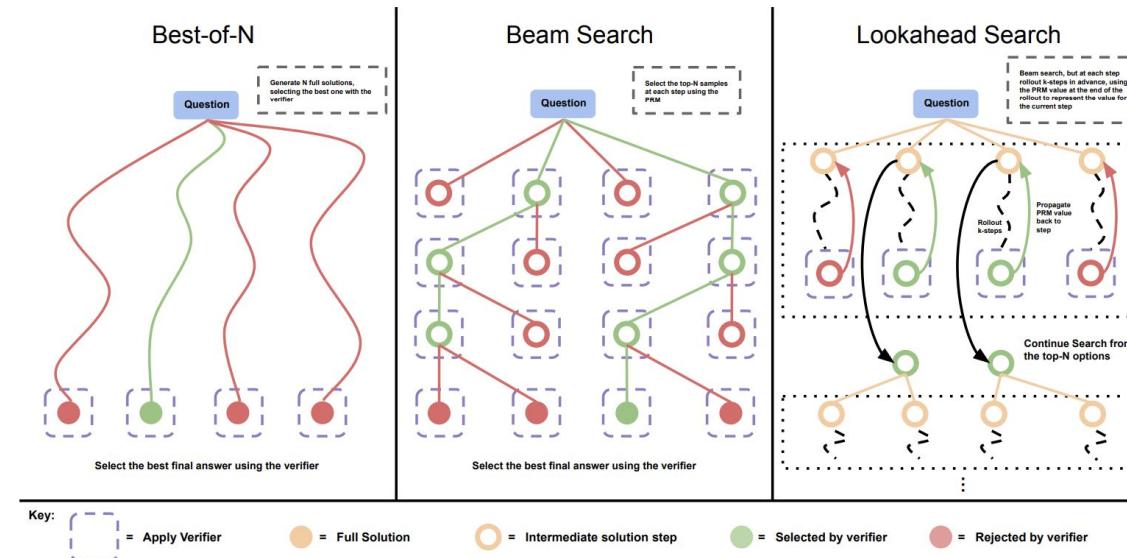


Figure 2 | Comparing different PRM search methods. Left: Best-of-N samples N full answers and then selects the best answer according to the PRM final score. Center: Beam search samples N candidates at each step, and selects the top M according to the PRM to continue the search from. Right: lookahead-search extends each step in beam-search to utilize a k-step lookahead while assessing which steps to retain and continue the search from. Thus lookahead-search needs more compute.

Deepseek-R1

Benchmark Result(DeepSeek-AI 2025)

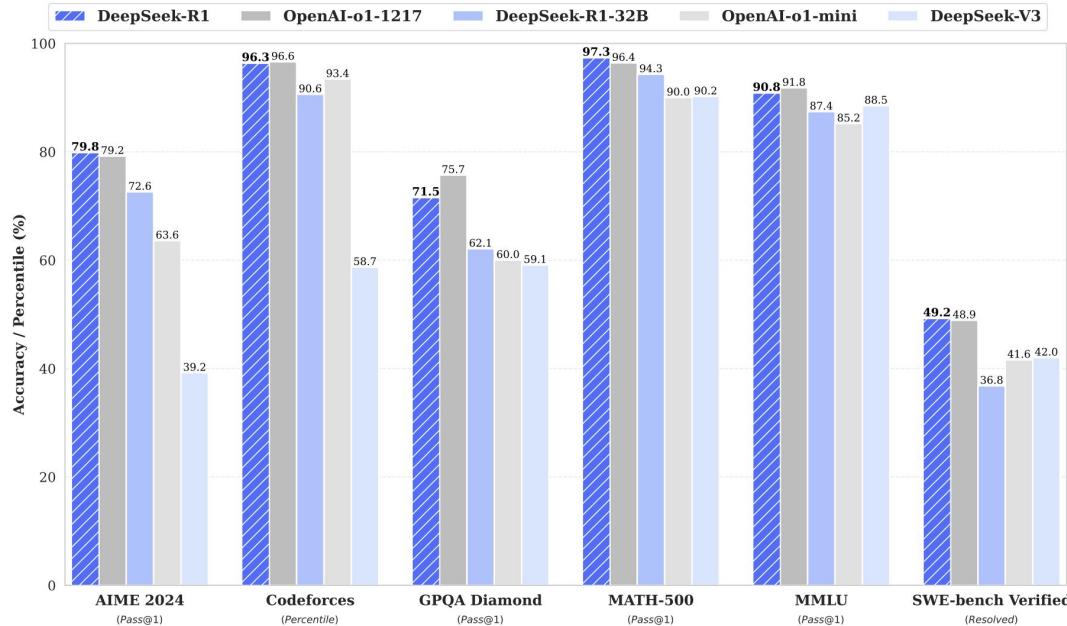


Figure 1 | Benchmark performance of DeepSeek-R1.

Large-Scale Reinforcement Learning

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)] \frac{1}{G} \sum_{i=1}^G \left(\min \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \text{clip} \left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1 - \varepsilon, 1 + \varepsilon \right) A_i \right) - \beta \mathbb{D}_{KL}(\pi_\theta || \pi_{ref}) \right), \quad (1)$$

$$\mathbb{D}_{KL}(\pi_\theta || \pi_{ref}) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \quad (2)$$

RL: Learning from Scratch

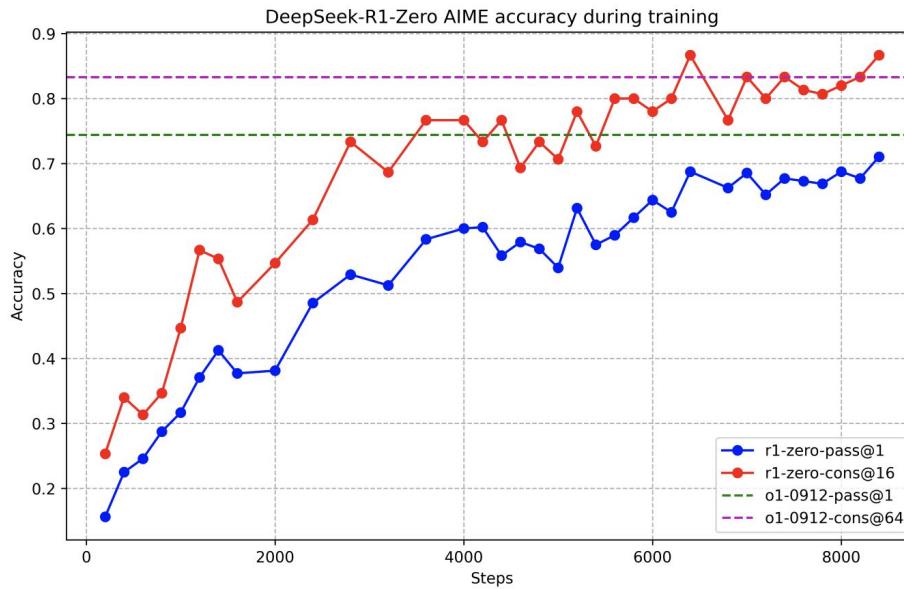


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

Learning to improve reasoning with more thinking

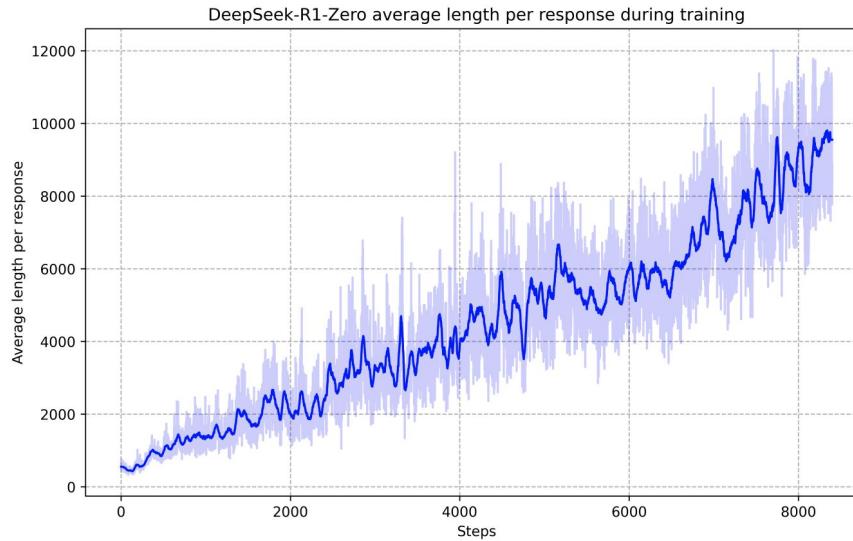


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

aha moment

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both ...

$$(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be ...

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

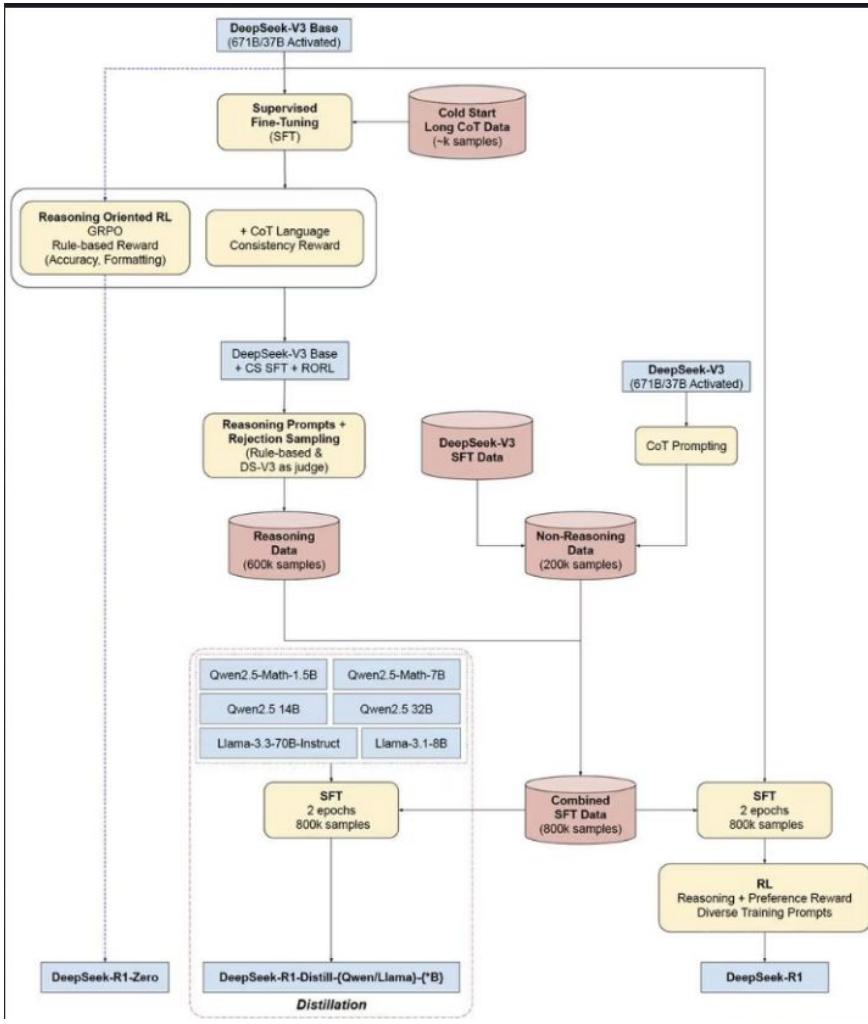
First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.



Conclusion and Discussion

*One thing that should be learned from the bitter lesson is the great power of **general purpose methods**, of methods that continue to **scale with increased computation** even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are **search** and **learning**.*

The Bitter Lesson – Rich Sutton

Reference

1. [Learning to reason with LLMs](#)
2. [Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#)
3. [LET'S VERIFY STEP BY STEP](#)
4. [Tree of Thoughts: Deliberate Problem Solving with Large Language Models](#)
5. [STaR: Self-Taught Reasoner Bootstrapping Reasoning With Reasoning](#)
6. [Thinking Fast and Slow with Deep Learning and Tree Search](#)
7. [Reinforced Self-Training \(ReST\) for Language Modeling](#)
8. [Stream of Search \(SoS\): Learning to Search in Language](#)
9. [Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters](#)
10. [DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning](#)