

LLM Reasoning: From Learning to Giving Up

Liyun Zhu
2025 Jan 22



Content

1. How to Train a GPT?
2. Chain of Thought (CoT)
3. Reinforcement Learning from Human Feedback (RLHF)
4. OpenAI o1 / Deepseek R1 for Reasoning

Since the launch of OpenAI O1 Model



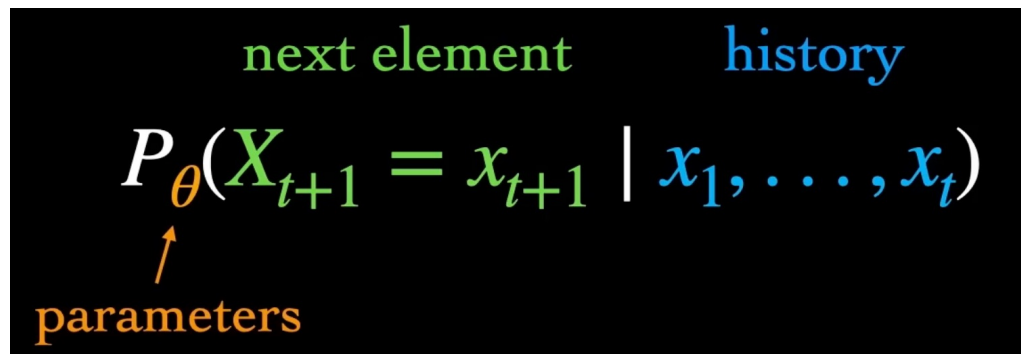
1.1 How to Train a GPT?

- Generative pretraining
- Supervised fine-tuning
- Reinforcement learning from Human Feedback

“Pre-training as we know it will unquestionably end...because we have but one internet”

- Ilya Sutskever, OpenAI co-founder, at the NeurIPS 2024

1.1 How to Train a GPT - Pretraining



The diagram shows the equation $P_{\theta}(X_{t+1} = x_{t+1} \mid x_1, \dots, x_t)$ on a black background. The text "next element" is in green above X_{t+1} and x_{t+1} . The text "history" is in blue above x_1, \dots, x_t . The text "parameters" is in orange below θ , with an orange arrow pointing up to θ .

$$P_{\theta}(X_{t+1} = x_{t+1} \mid x_1, \dots, x_t)$$

Why the Generative pretraining is not enough?

The alignment problem: Specific task

1.1 How to Train a GPT - SFT

- **Supervised Fine-Tuning (SFT):** Behavior clone expert
- Distribution shift: Bad approximation / limited training data / partial observability of the environment
- May cause overconfident assertions or output complete nonsense
- RLHF: treating the reward model as a binary classifier

2.1 Chain of Thought – Core Idea

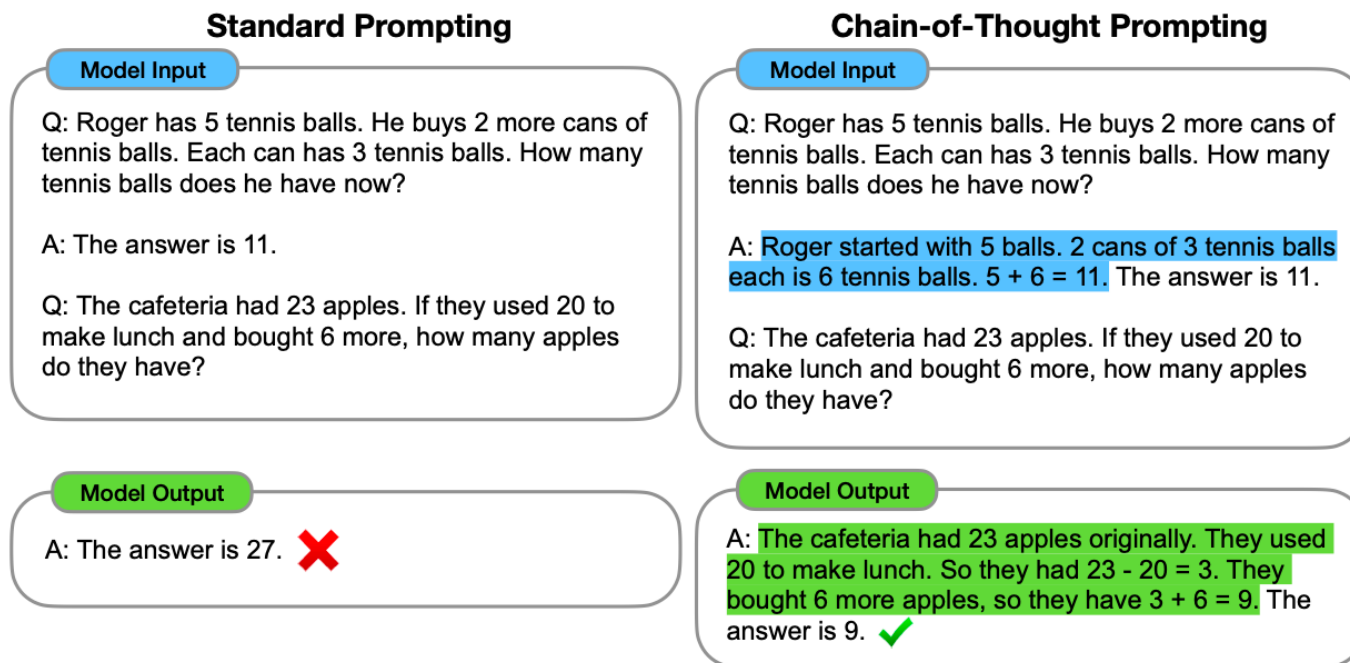


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Large Language Models offer the exciting prospect of in-context **few-shot learning** via *prompting*.

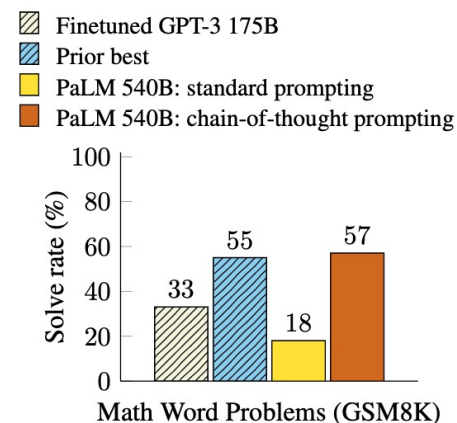


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).

2.1 Chain of Thought – Core Idea

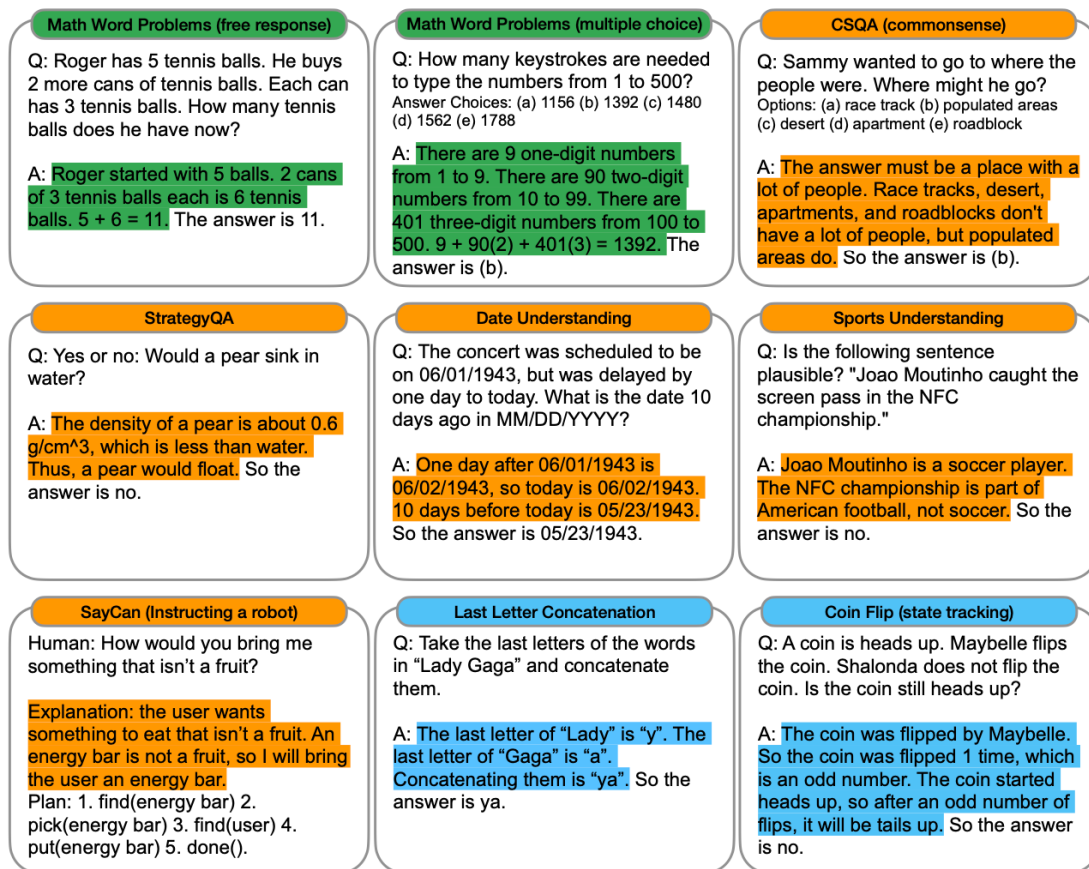


Figure 3: Examples of (input, chain of thought, output) triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

- Example of responses which breaks problems into simple steps.

Table 1: Chain of thought prompting outperforms standard prompting for various large language models on five arithmetic reasoning benchmarks. All metrics are accuracy (%). Ext. calc.: post-hoc external calculator for arithmetic computations only. Prior best numbers are from the following. a: Cobbe et al. (2021). b & e: Pi et al. (2022), c: Lan et al. (2021), d: Piękos et al. (2021).

	Prompting	GSM8K	SVAMP	ASDiv	AQuA	MAWPS
Prior best	N/A (finetuning)	55 ^a	57.4 ^b	75.3 ^c	37.9 ^d	88.4 ^e
UL2 20B	Standard	4.1	10.1	16.0	20.5	16.6
	Chain of thought + ext. calc	4.4 (+0.3) 6.9	12.5 (+2.4) 28.3	16.9 (+0.9) 34.3	23.6 (+3.1) 23.6	19.1 (+2.5) 42.7
LaMDA 137B	Standard	6.5	29.5	40.1	25.5	43.2
	Chain of thought + ext. calc	14.3 (+7.8) 17.8	37.5 (+8.0) 42.1	46.6 (+6.5) 53.4	20.6 (-4.9) 20.6	57.9 (+14.7) 69.3
GPT-3 175B (text-davinci-002)	Standard	15.6	65.7	70.3	24.8	72.7
	Chain of thought + ext. calc	46.9 (+31.3) 49.6	68.9 (+3.2) 70.3	71.3 (+1.0) 71.1	35.8 (+11.0) 35.8	87.1 (+14.4) 87.5
Codex (code-davinci-002)	Standard	19.7	69.9	74.0	29.5	78.7
	Chain of thought + ext. calc	63.1 (+43.4) 65.4	76.4 (+6.5) 77.0	80.4 (+6.4) 80.0	45.3 (+15.8) 45.3	92.6 (+13.9) 93.3
PaLM 540B	Standard	17.9	69.4	72.1	25.2	79.2
	Chain of thought + ext. calc	56.9 (+39.0) 58.6	79.0 (+9.6) 79.8	73.9 (+1.8) 72.6	35.8 (+10.6) 35.8	93.3 (+14.2) 93.5

2.1 CoT - Few-shot / Zero-shot

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) *The answer is 8.* ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) *The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4.* ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) *8* ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

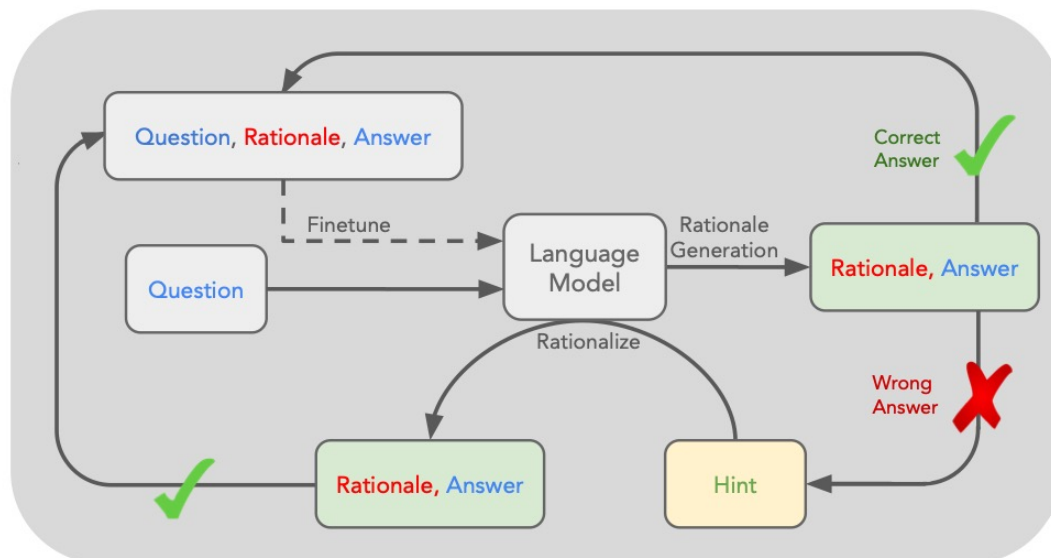
A: **Let's think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

2.1 CoT - Comparison

	Standard Prompting	Chain of Thought
Input Token	Less (Question + Answer)	More (Question + Rationale + Answer)
Inference Time	Shorter	Longer
Interpretability	Worse	Better
Accuracy	Low	High

2.2 STaR - Bootstrap Reasoning with Reasoning



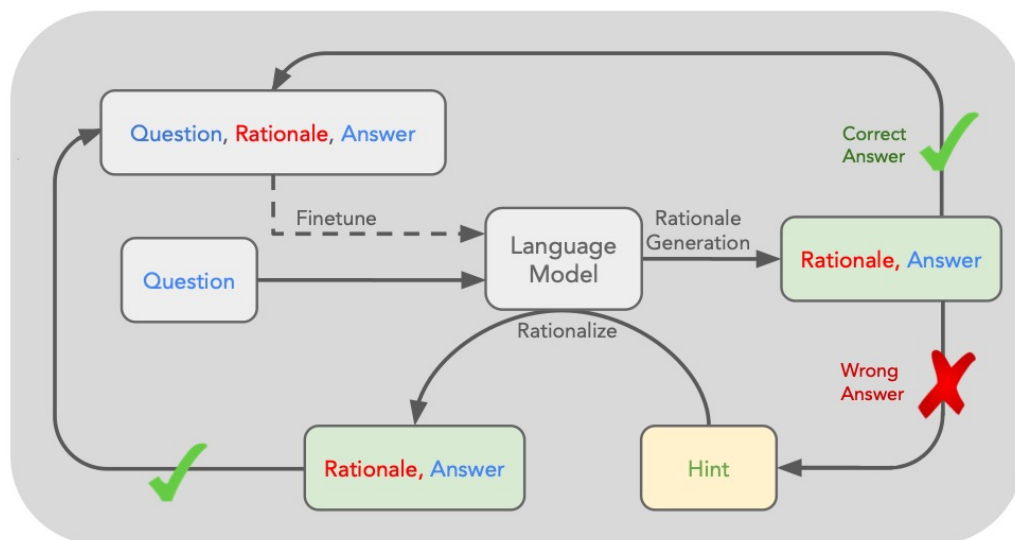
Q: What can be used to carry a small dog?
Answer Choices:
(a) swimming pool
(b) basket
(c) dog show
(d) backyard
(e) own home
A: The answer must be something that can be used to carry a small dog. Baskets are designed to hold things. Therefore, the answer is basket (b).

2.2 The Idea of STaR

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: $(\hat{r}_i, \hat{y}_i) \leftarrow M_{n-1}(x_i) \quad \forall i \in [1, D]$ # Perform rationale generation
- 4: $(\hat{r}_i^{\text{rat}}, \hat{y}_i^{\text{rat}}) \leftarrow M_{n-1}(\text{add_hint}(x_i, y_i)) \quad \forall i \in [1, D]$ # 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**



1. 初始训练模型

复制预训练模型 M 为 M_0 。

2. 外部送入循环

对每次外部送入 (共 N 次) 执行:

• 步骤 2.1: 推理生成 (Rationale Generation)

使用当前模型 M_{n-1} 对所有问题 x_i 生成推理路径 r_i 和答案 \hat{y}_i 。

- 若生成的答案 \hat{y}_i 正确 (即 $\hat{y}_i = y_i$)，将 (x_i, r_i, y_i) 加入到正确推理数据集中。

• 步骤 2.2: 逆向推理 (Rationalization)

对于那些生成错误答案的问题 ($y_i \neq \hat{y}_i$)，提供正确答案 y_i ，引导模型生成逆向推理路径 r_i^{rat} 。

- 若逆向推理路径成功生成正确答案，加入到正确逆向推理数据集中。

• 步骤 2.3: 数据集合并

合并推理生成数据集和逆向推理数据集，形成新的训练数据集 D_n 。

• 步骤 2.4: 微调模型 (Fine-tuning)

使用新生成的训练数据集 D_n 微调模型 M_{n-1} 得到 M_n 。

3. 检查收敛

如果模型性能不再显著提高或达到最大迭代次数 N ，停止循环。

4. 返回最终模型

输出最终微调模型 M^* 。

2.2 STaR – An example

- Assume we have 10,000 questions.
- The question set is different from the training set $(x, r, y) \rightarrow$ (question, rationale, answer).
- The rationale is different in every inference step. The same question may generate different (x, r, y) samples.
- Assume the failure rate of given the wrong answer is 30%, and the failure rate of the rationalization is 50%. Therefore, the overall failure rate is 15%.
- N depends on the size of the task.

Number of iterations (N)	Cumulative New Samples Added
1	8500
2	17000
3	25500
4	34000
5	42500
6	51000
7	59500
8	68000
9	76500
10	85000

2.2 STaR – A Detailed Failure Case



2.2 STaR – A Detailed Failure Case

User: If the video is anomaly, please extract the key anomaly object from your answer, 1 or 2 words

Assistant: I apologize for any confusion, but the **video does not show any anomalies**. The scene is typical of a retail store with no visible signs of unusual or dangerous activities.

User: **Actually, the video is a robbery**. Please find the reason of the anomaly and anomaly signs.

Assistant: I apologize for the oversight. Upon closer inspection, the video does show a robbery. Here are the key anomaly signs:

1. **Individual in Distress**: A person appears to be on the floor, possibly indicating a struggle or being incapacitated.
2. **Staff Interaction**: Store staff are interacting with individuals in a way that suggests confrontation or restraint.

These signs indicate a robbery is taking place.

1. 人员遇险：有一人似乎倒在地上，这可能表明其正在挣扎或失去行动能力。
2. 员工互动：商店员工与一些人发生互动，其方式显示出双方可能处于对峙或员工在实施制止行为。这些迹象表明正在发生抢劫事件。

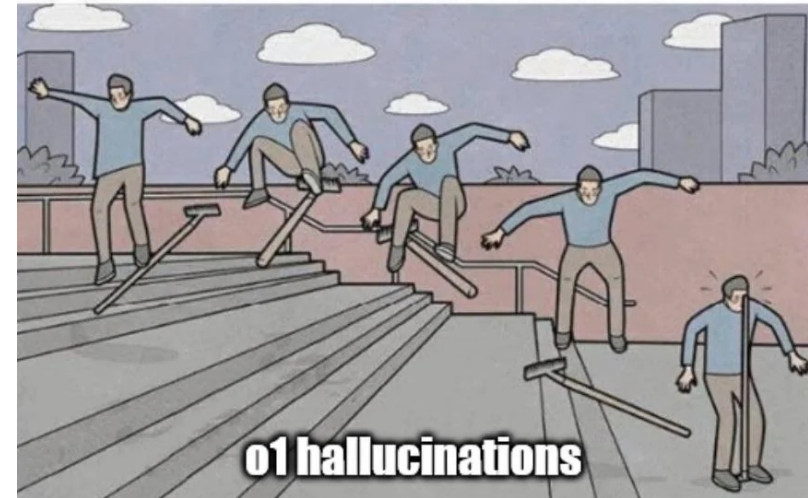
User: If the video is anomaly, please extract the key anomaly object from your answer, 1 or 2 words

Assistant: **Staff member**.

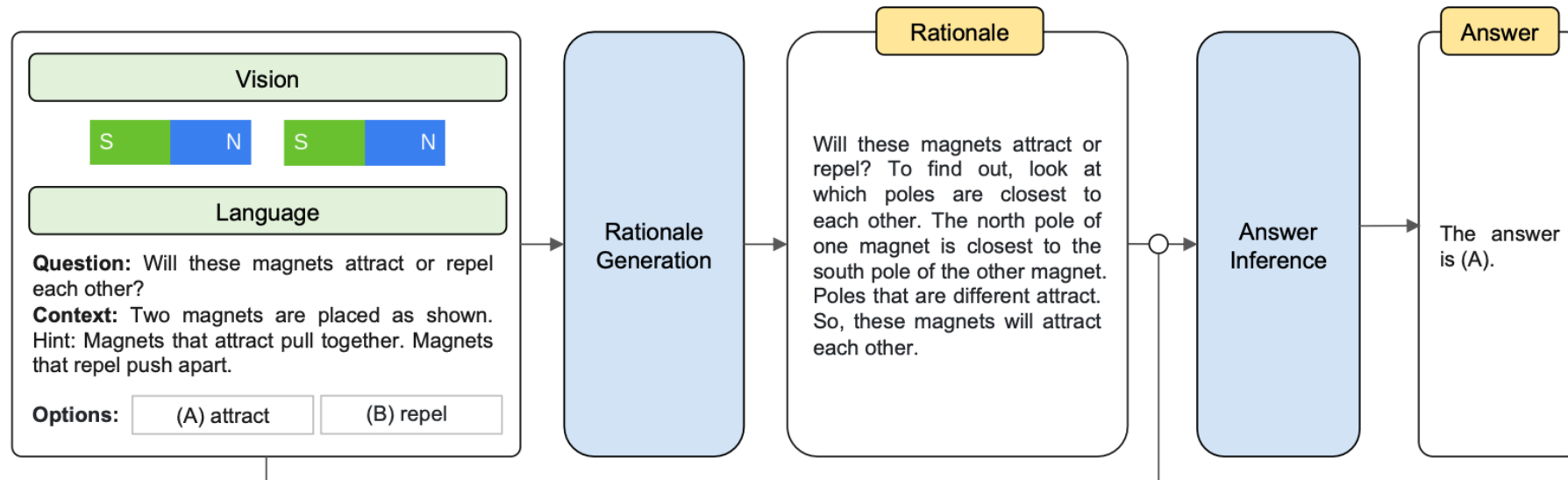
2.2 STaR – Summary

Disadvantage:

- Potential Failure case – Wrong reasoning process given the correct answer
- High leverage - With 100 questions, I can generate 100,000 data points. (Just like borrowing a large amount of money to buy a house 🏠) – May cause overfitting, limited computational resource, etc.



2.3 Multimodal CoT



We propose Multimodal-CoT that incorporates language (text) and vision (images) modalities into a two-stage framework that separates rationale generation and answer inference.

2.4 Summary of CoT

Chain of Thought contains behavior like:

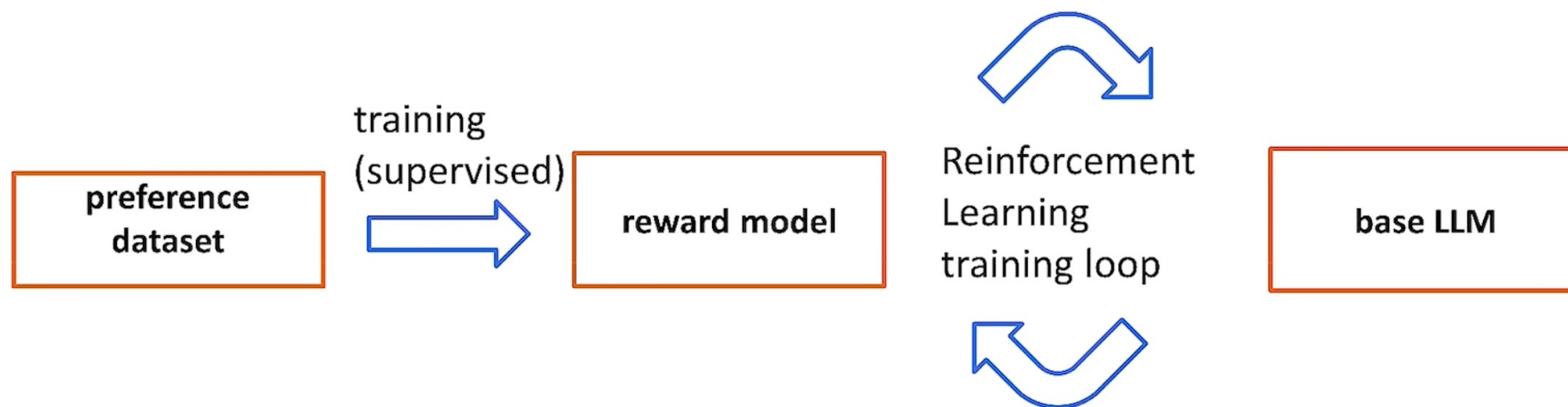
- Error correction
- Trying multiple strategies
- Breaking down problems into smaller steps

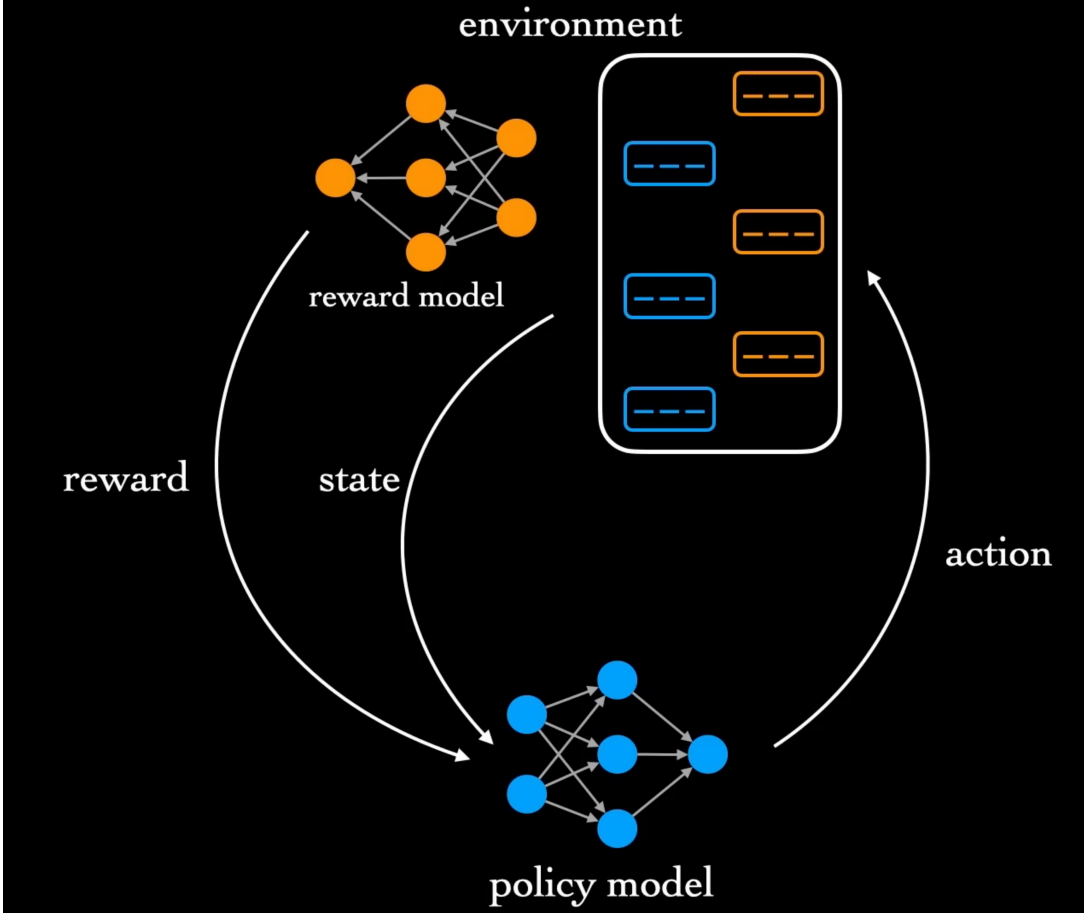
3.1 Reinforcement Learning from Human Feedback

- In classical reinforcement learning, an intelligent agent's goal is to learn a function that guides its behavior, called a policy. This function is iteratively updated to maximize rewards based on the agent's task performance.
- However, explicitly defining a reward function that accurately approximates human preferences is challenging. Therefore, RLHF seeks to train a "reward model" directly from human feedback.

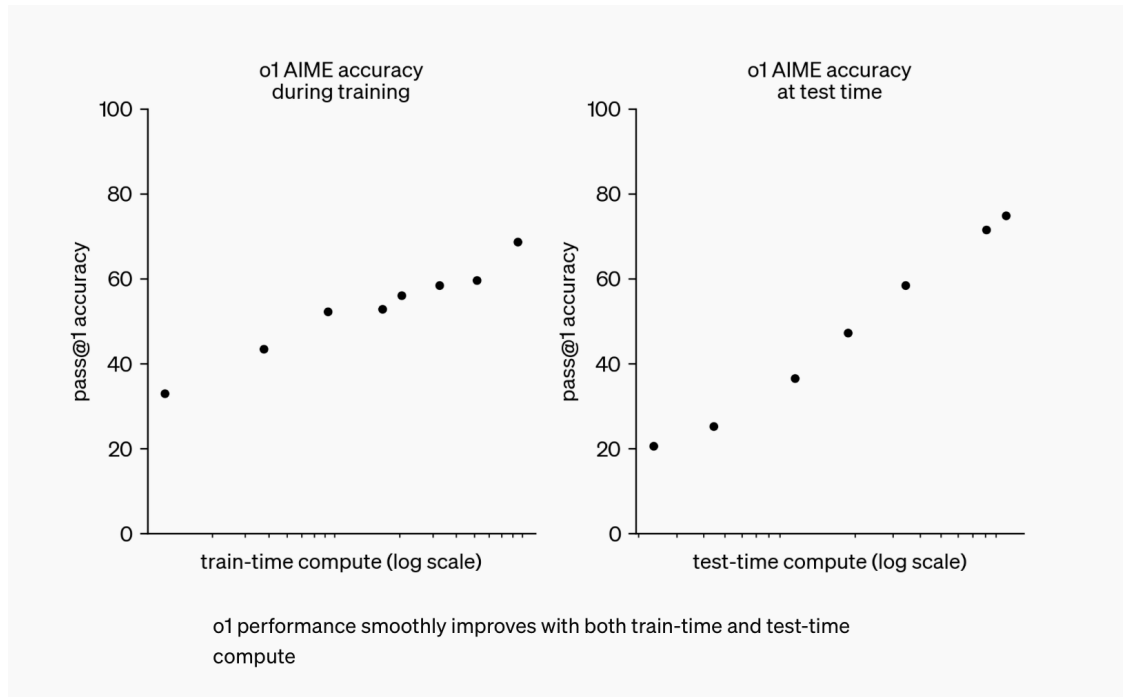
3.1 Reinforcement Learning from Human Feedback

- Create a Preference Dataset (helpfulness / truthfulness / harmlessness)
- Use the preference dataset to train a reward model.
- Use the reward model in a RL loop to finetune the LLM





4.1 OpenAI o1 – Ragnarök? (Twilight of the Gods?)



We have found that the performance of o1 consistently improves with more reinforcement learning (train-time compute) and with more time spent thinking (test-time compute).

post-training process

Thinking for longer time gets better result! 🤔

- For some problems, verifying a good solution is easier than generating one
 - Many puzzles (Sudoku, for example)
 - Math
 - Programming
- Examples where verification isn't much easier
 - Information retrieval (What's the capital of Bhutan?)
 - Image recognition
- When a generator-verifier gap exists *and we have a good verifier*, we can **spend more compute on inference to achieve better performance**

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

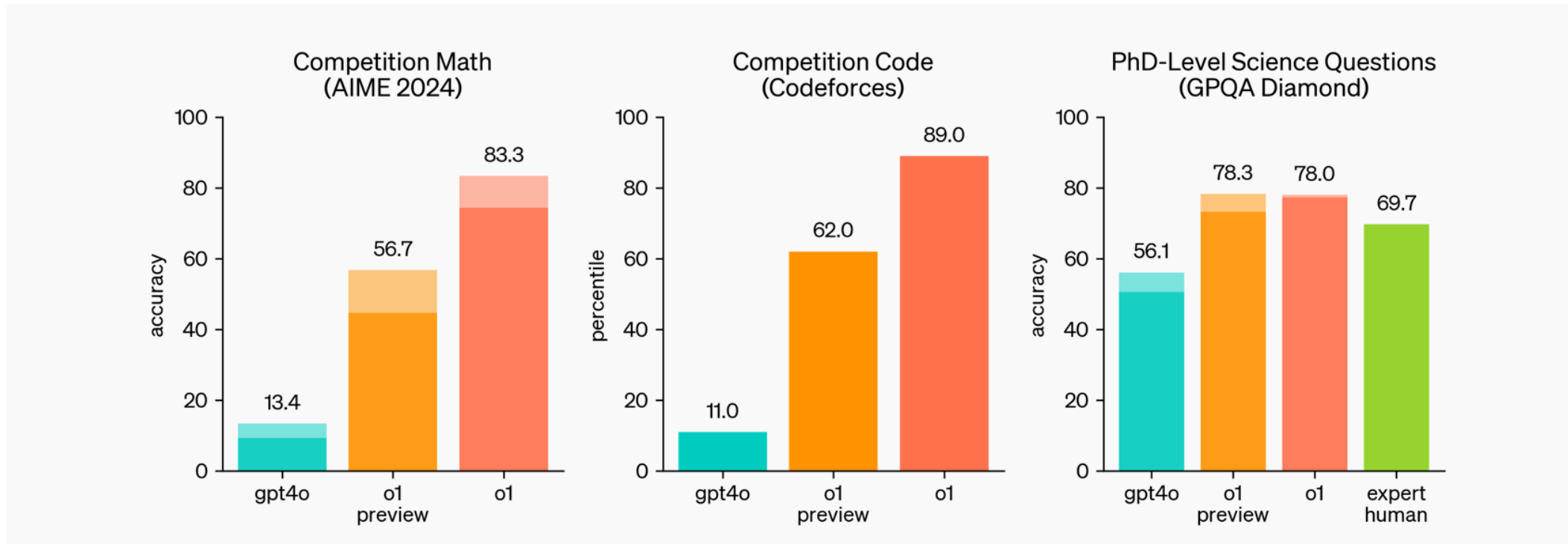
Extension— inference-time scaling for diffusion models

Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps

Nanye Ma^{†, *},¹, Shangyuan Tong^{†, *},², Haolin Jia³, Hexiang Hu³, Yu-Chuan Su³, Mingda Zhang³, Xuan Yang³,
Yandong Li³, Tommi Jaakkola², Xuhui Jia³ and Saining Xie^{1,3}

[†]Equal contribution, ¹NYU, ²MIT, ³Google, ^{*}Work done during an internship at Google

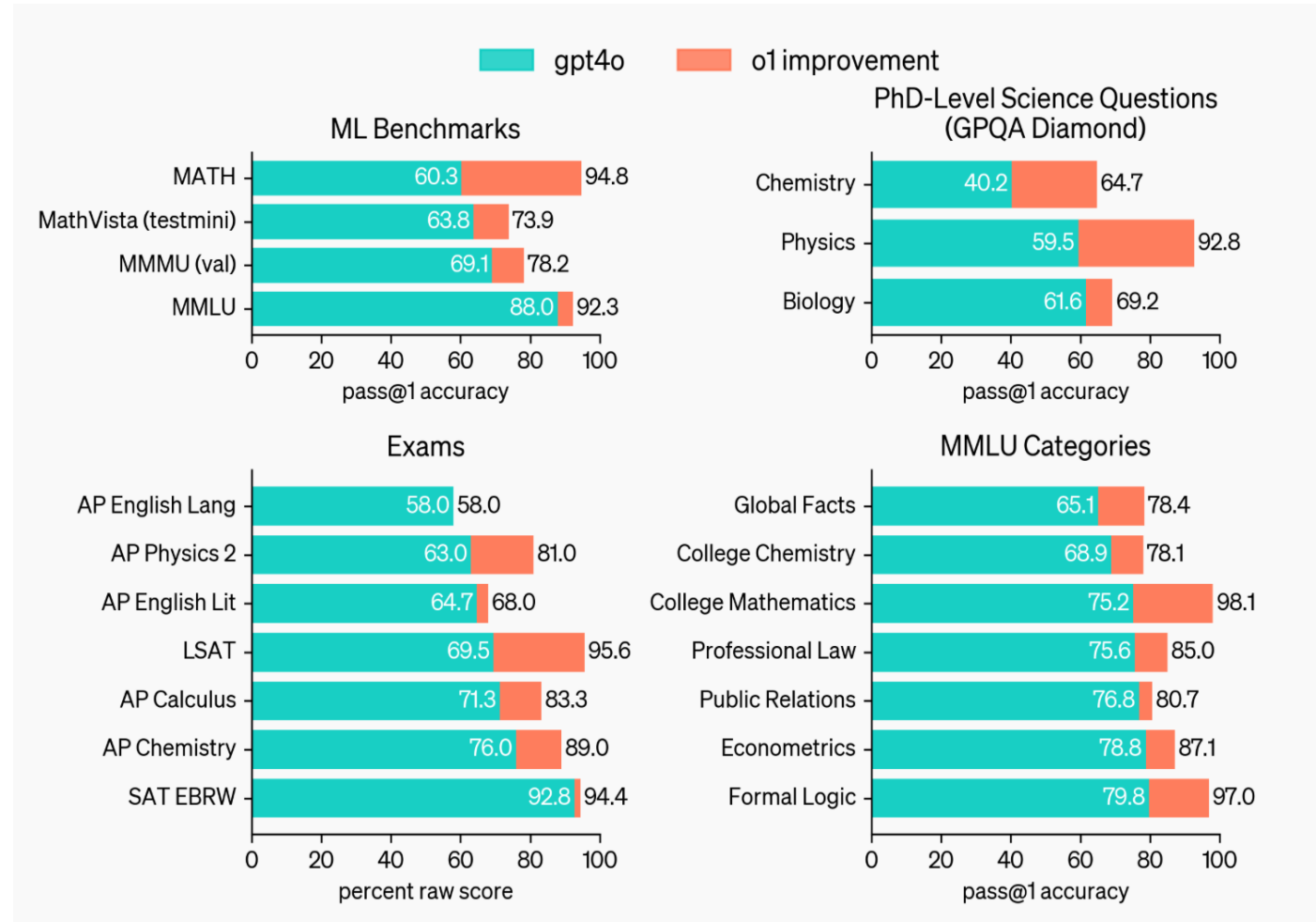
4.1 OpenAI o1 – Ragnarök? (Twilight of the Gods?)



o1 greatly improves over GPT-4o on challenging reasoning benchmarks.

4.1 OpenAI o1 – Ragnarök? (Twilight of the Gods?)

- o1 improves over GPT-4o on a wide range of benchmarks, including 54/57 MMLU subcategories. Seven are shown for illustration. o1 improves over GPT-4o on a wide range of benchmarks, including 54/57 MMLU subcategories. Seven are shown for illustration.



4.2 OpenAI o1 – No Reasoning tokens?

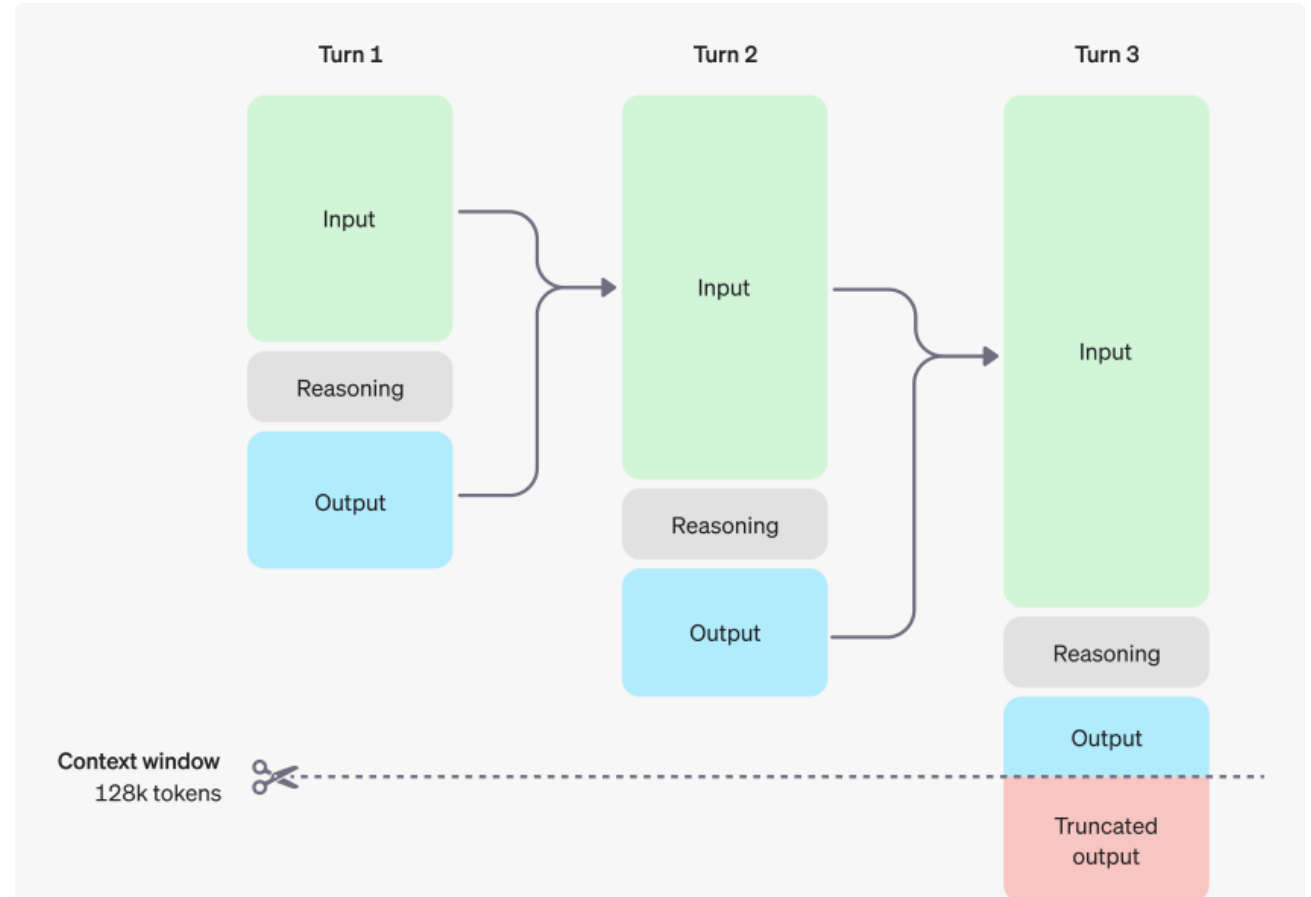
“Assuming it is faithful and legible, the hidden chain of thought allows us to “read the mind” of the model and understand its thought process ...

*However, for this to work the model must have freedom to express its thoughts in unaltered form, so **we cannot train any policy compliance or user preferences onto the chain of thought. We also do not want to make an unaligned chain of thought directly visible to users.**”*

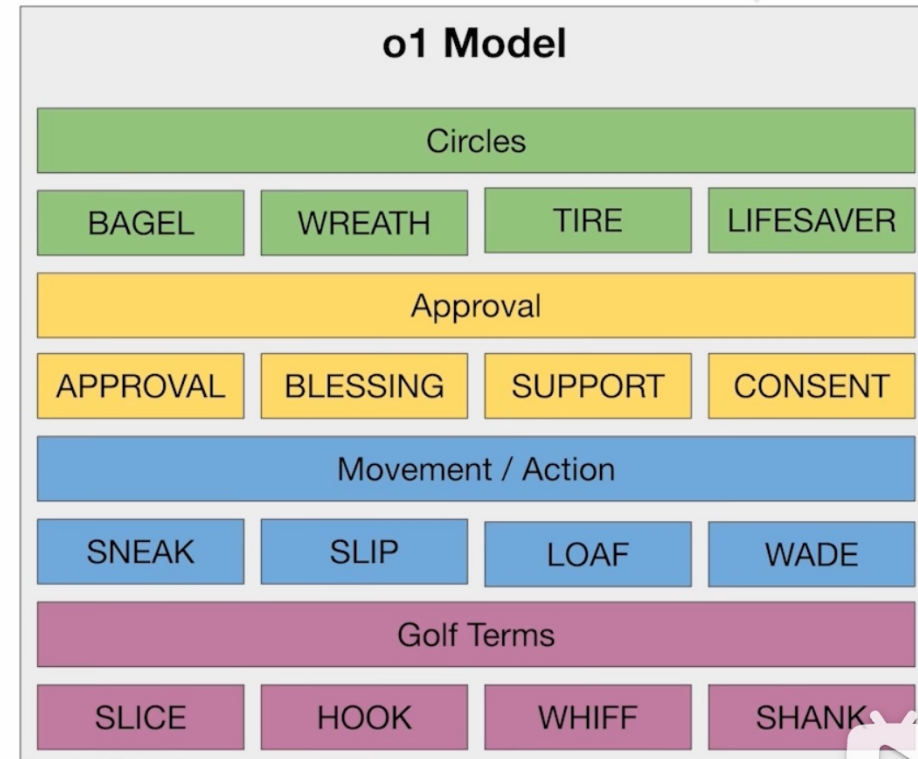
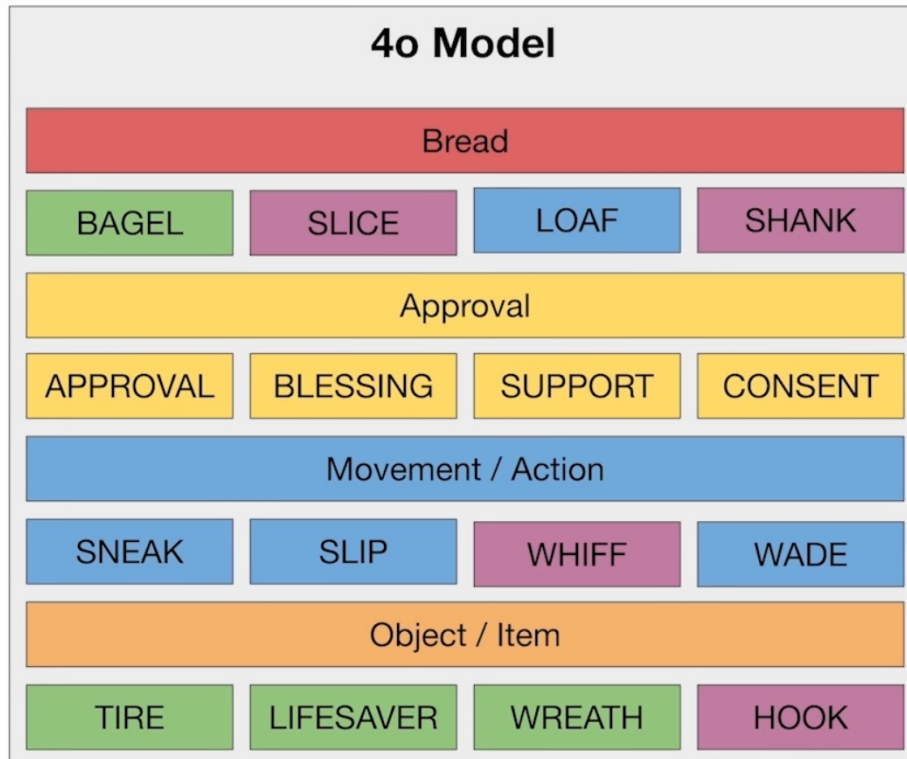
- Hiding the Chains of Thought, from OpenAI o1 Website

4.2 OpenAI o1 – No Reasoning tokens?

- Reasoning tokens are not passed from one turn to the next.



4.2 OpenAI o1 – Abstract Reasoning



4.2 DeepSeek R1

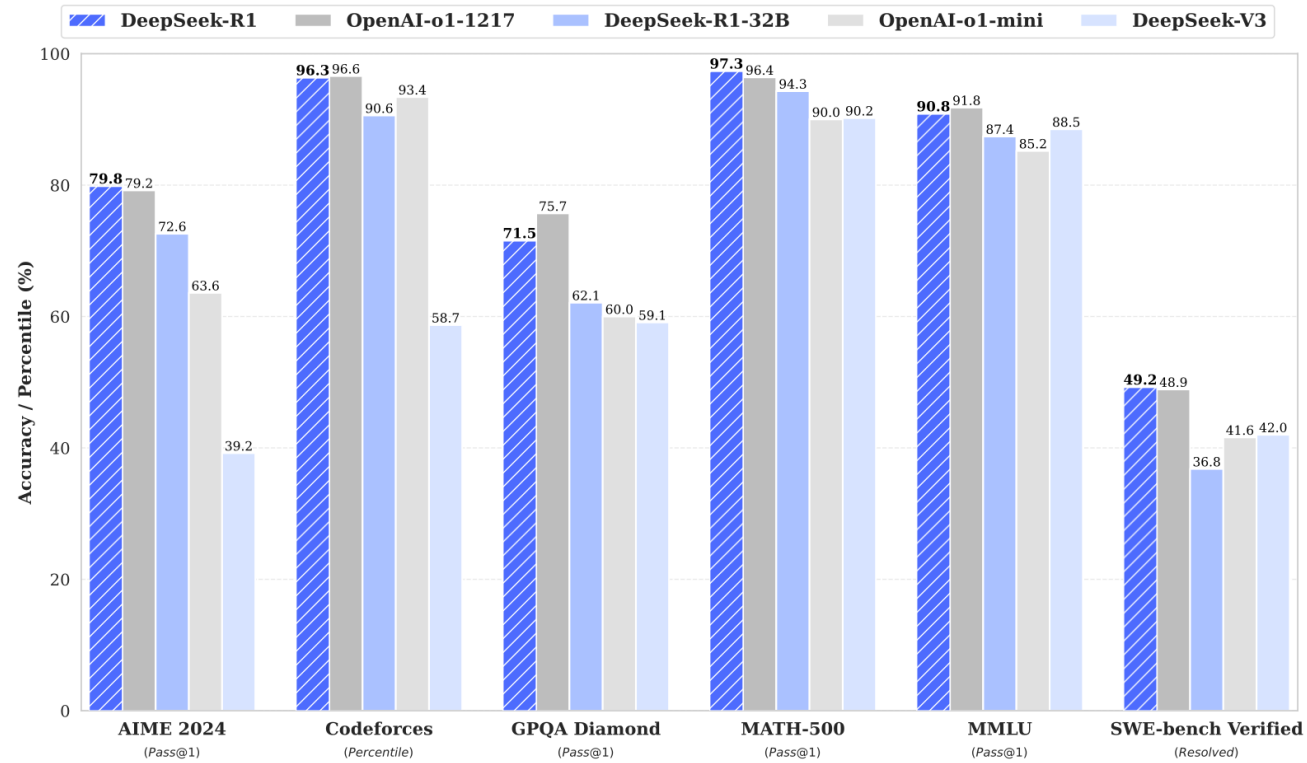


Figure 1 | Benchmark performance of DeepSeek-R1.

4.2 DeepSeek R1-Zero – The Beauty of RL

- Notably, it is the first open research to validate that reasoning capabilities of LLMs can be incentivized purely through RL, without the need for SFT.
- DeepSeek-R1-Zero to attain robust reasoning capabilities without the need for any supervised fine-tuning data.
- Base model: DeepSeek-V3

4.2 DeepSeek R1-Zero – The Beauty of RL

- Group Relative Policy Optimization
- Reward Modeling: Accuracy rewards + Format rewards

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **prompt**. Assistant:

Table 1 | Template for DeepSeek-R1-Zero. **prompt** will be replaced with the specific reasoning question during training.

4.2 DeepSeek R1-Zero – 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 ...

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

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:

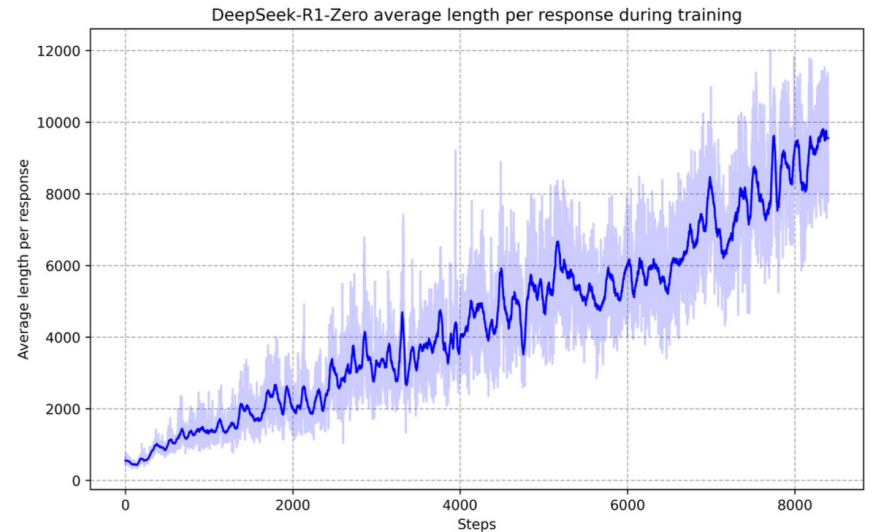
$$\sqrt{a - \sqrt{a + x}} = x$$

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: ...

...



Self-evolution Process of
DeepSeek-R1-Zero

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.

4.2 DeepSeek-R1: RL with Cold Start

- A key limitation of DeepSeek-R1-Zero is that its content is often not suitable for reading. Responses may mix multiple languages or lack markdown formatting to highlight answers for users.
- We collect thousands of cold-start data to fine-tune the DeepSeek-V3-Base as the starting point for RL.
- To mitigate the issue of language mixing, we introduce a language consistency reward during RL training.

4.2 DeepSeek-R1: Limitations

In the future, we plan to invest in research across the following directions for DeepSeek-R1.

- **General Capability:** Currently, the capabilities of DeepSeek-R1 fall short of DeepSeek-V3 in tasks such as function calling, multi-turn, complex role-playing, and json output. Moving forward, we plan to explore how leveraging long CoT to enhance tasks in these fields.
- **Language Mixing:** DeepSeek-R1 is currently optimized for Chinese and English, which may result in language mixing issues when handling queries in other languages. For instance, DeepSeek-R1 might use English for reasoning and responses, even if the query is in a language other than English or Chinese. We aim to address this limitation in future updates.
- **Prompting Engineering:** When evaluating DeepSeek-R1, we observe that it is sensitive to prompts. Few-shot prompting consistently degrades its performance. Therefore, we recommend users directly describe the problem and specify the output format using a zero-shot setting for optimal results.
- **Software Engineering Tasks:** Due to the long evaluation times, which impact the efficiency of the RL process, large-scale RL has not been applied extensively in software engineering tasks. As a result, DeepSeek-R1 has not demonstrated a huge improvement over DeepSeek-V3 on software engineering benchmarks. Future versions will address this by implementing reject sampling on software engineering data or incorporating asynchronous evaluations during the RL process to improve efficiency.

viv ⁺ 学习策略	优点	缺点	代表
Behaviour Clone Expert	<ol style="list-style-type: none"> 更像人、专家，并且有人的偏好 可以通过单agent的方式训练 当数据量无限多的时候可以取得完美表现 	<ol style="list-style-type: none"> 实际能力由于数据分布有偏 无法探索出人类行为之外的行为 无法利用错误数据 	各种游戏陪玩AI, LLM SFT
RLHF ⁺	<ol style="list-style-type: none"> 可以对齐人类偏好及价值观 能力利用错误数据 数据利用效率高 	<ol style="list-style-type: none"> 偏好建模困难，容易hacking 训练成本高 	ChatGPT
Self-play	<ol style="list-style-type: none"> 绝对强度更高，甚至超越最强人类、专家 可以实现双人零和博弈⁺的最优 	<ol style="list-style-type: none"> 有时候无法理解人类，行为不像人 训练及推理成本极高 	AlphaGo, OpenAI o1

Self-play in LLMs

- In games, we have a great verifier but a bad generator
 - Really easy to score chess
 - Unlimited reward data
- In LLMs, we have a great generator but a bad verifier (usually)
 - Trillions of tokens of human text
 - Far less reward data
 - Hard to score one poem versus another, especially a partial poem!
- But this may change with time!
 - Amount of reward data is increasing
 - Some domains are easier to score than others

Thank you!