Introduction to Agentic AI

-- AI Agent Reasoning Basics

Instructor: Guangjing Wang

guangjingwang@usf.edu

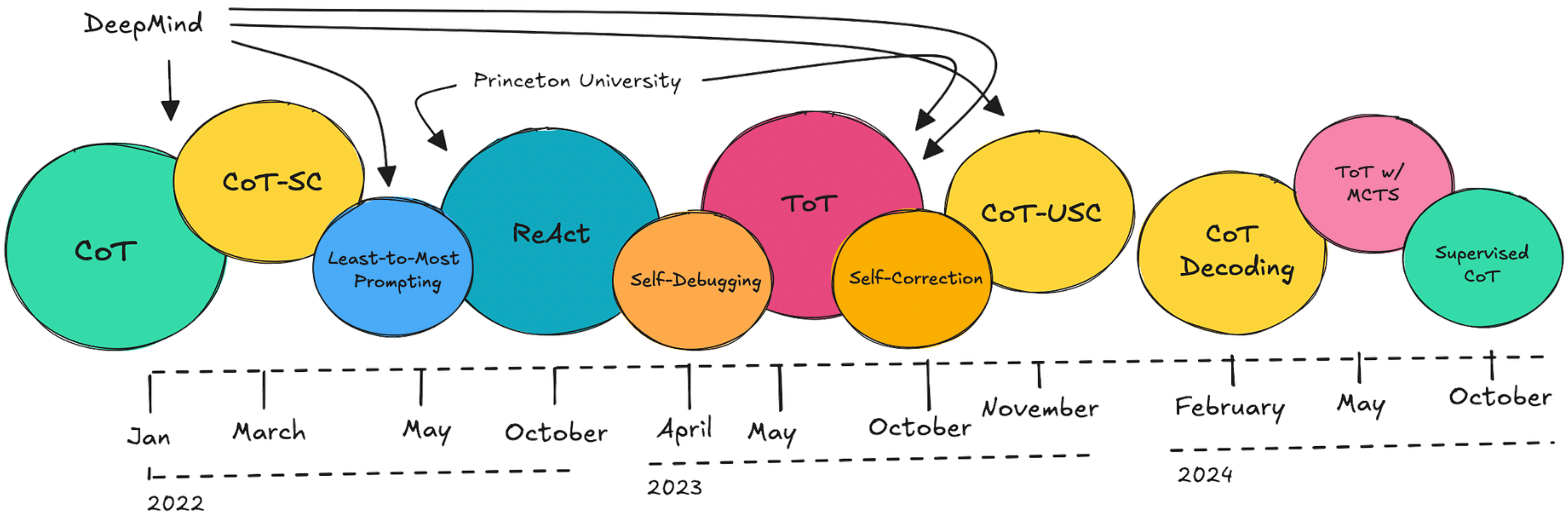
Last Lecture

- Short-term and Working Memory
  - Context Window
- Long-term Memory
  - Retrieval Augmented Generation
- Memory Design in Research

This Lecture

- Chain-of-Thought and ReAct
- Post-training (Fine-tuning) Reasoning
  - DeekSeek R1
- In-context Reasoning Basics
  - Reflexion

In-context Reasoning Mechanism Examples



https://medium.com/@ilsilfverskiold/8d8b090bf699

Review: Chain-of-Thought (CoT)

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is 6.

Q: A coin is heads up. Jamey flips the coin. Teressa flips the coin. Is the coin still heads up?

A: The coin was flipped by Jamey and Teressa. So the coin was flipped 2 times, which is an even number. The coin started heads up, so after an even number of flips, it will still be heads up. So the answer is yes.

- CoT: A series of intermediate reasoning steps.
- Through CoT, additional computation can be allocated to problems that require more reasoning steps.

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models https://arxiv.org/abs/2201.11903

Why is the Idea of CoT Important?

- CoT allows models to decompose multi-step problems into intermediate steps.
- CoT provides an interpretable window into the behavior of the model, suggesting how it might have arrived at a particular answer.
- Usage of CoT in **Prompting**:
  - Chain-of-thought examples in prompts
  - An example that consists of triples: **<input, chain of thought, output>**
- Usage of CoT in **Post-training** (SFT and Reinforcement Learning)
  - Compose CoT Training Dataset

Scaling Instruction-Finetuned Language Models

Explore instruction finetuning with a particular focus on

- scaling the number of tasks
- scaling the model size
- finetuning on chain-of-thought data (training dataset)

- Instruction finetuning with the above aspects dramatically improves performance on a variety of model classes.

- Some Timeline:
  - **Google: Transformer**: 06/2017; **BERT**: 10/2018; **CoT**: 01/2022; **Scaling Instruction-tuning**: 10/2022.
  - **OpenAI: GPT**: 06/2018; **GPT-3**: 06/2020; **GPT-3.5**: 11/2022.

Scaling Instruction-Finetuned Language Models https://arxiv.org/abs/2210.11416

CoT and ReAct Prompting-based Reasoning

- CoT reasoning is a static black box
  - Model uses its own internal representations to generate thoughts and is not grounded in the external world.
  - Hallucination and error propagation.
- ReAct: <reasoning-action> examples in prompting
  - Prompting LLMs to generate both verbal reasoning traces and actions pertaining to a task in an interleaved manner.
  - Dynamic reasoning to create, maintain, and adjust high-level plans for acting.
  - Acting interacts with the external environments to incorporate additional information into reasoning.

ReAct Prompting: few-shot examples in prompts

Example 2491 (gt: REFUTES)

```
ReAct
Claim:          Bermuda Triangle is a loosely-defined region in the Pacific Ocean.
Thought 1:      I need to search Bermuda Triangle and find if it is in the Pacific Ocean.
Action 1:       Search[Bermuda Triangle]
Observation 1:  Reportedly haunted locations...
Thought 2:      The observation says that it is in the western part of the North Atlantic
                Ocean, so it is not in the Pacific Ocean.
Action 2:       Finish[REFUTES]
Observation 2:  Episode finished
```

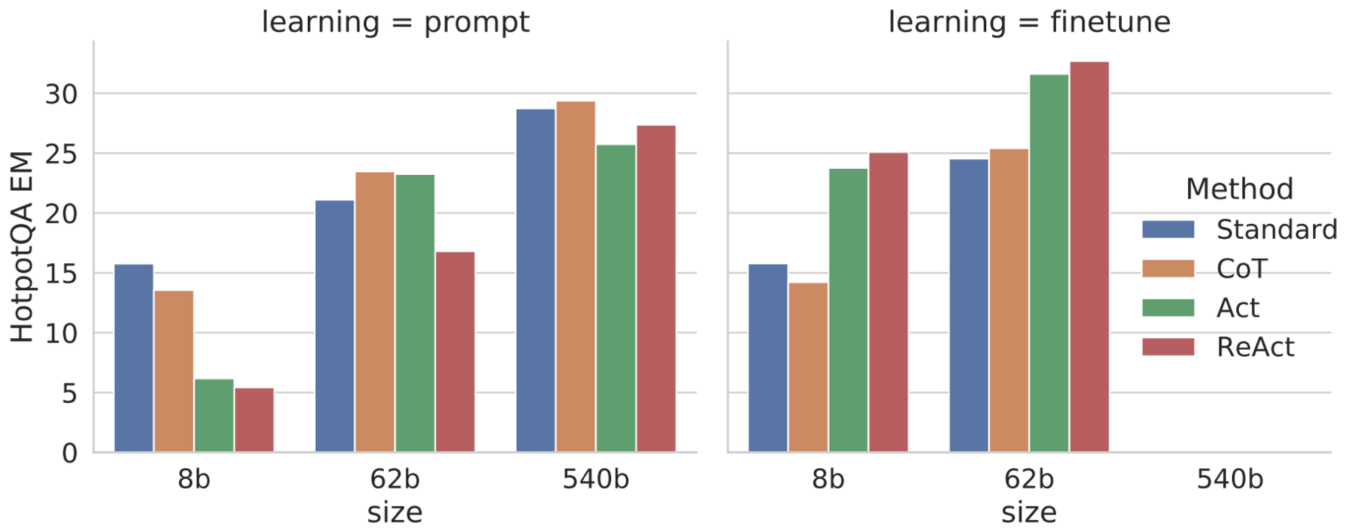
```
Act
Claim:          Bermuda Triangle is a loosely-defined region in the Pacific Ocean.
Action 1        :  Search[Bermuda Triangle]
Observation 1   :  Reportedly haunted locations:...
Action 2:       Finish[REFUTES]
Observation 2:  Episode finished
```

```
CoT
Claim:          Bermuda Triangle is a loosely-defined region in the Pacific Ocean.
Thought:        Let's think step by step.  The Bermuda Triangle is a loosely-defined region
                in the Atlantic Ocean.  The Pacific Ocean is not the Atlantic Ocean.
Answer:         REFUTES
```

ReAct: Synergizing Reasoning and Acting in Language Models https://arxiv.org/pdf/2210.03629

ReAct in Prompting and Fine-tuning

Prompt Method[a]	**HotpotQA** (EM)	**Fever** (Acc)
Standard	28.7	57.1
CoT (Wei et al., 2022)	29.4	56.3
CoT−SC (Wang et al., 2022a)	33.4	60.4
Act	25.7	58.9
ReAct	27.4	60.9
CoT−SC → ReAct	34.2	**64.6**
ReAct→ CoT−SC	**35.1**	62.0
Supervised SoTA[b]	67.5	89.5



- ReAct + CoT-SC (self-consistency) perform best for prompting LLMs
- ReAct performs best for fine-tuning

Is human-annotated demonstrations necessary?

Human-annotated reasoning like CoT:

- Traces hinders scalability and introduces cognitive biases.
- Performance is inherently capped by the human-provided exemplars, which prevents the exploration of superior, non-human-like reasoning pathways.

DeepSeek R1-Zero:

- The reasoning abilities of LLMs can be incentivized through pure reinforcement learning (RL).
- No need for human-labeled reasoning trajectories.
- The proposed RL framework facilitates the emergent development of advanced reasoning patterns, such as self-reflection, verification, and dynamic strategy adaptation.

Group Relative Policy Optimization in DeepSeek

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

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

Question q; **Output** o; **Policy (Model)** pi_theta; **Reward** r

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}$$

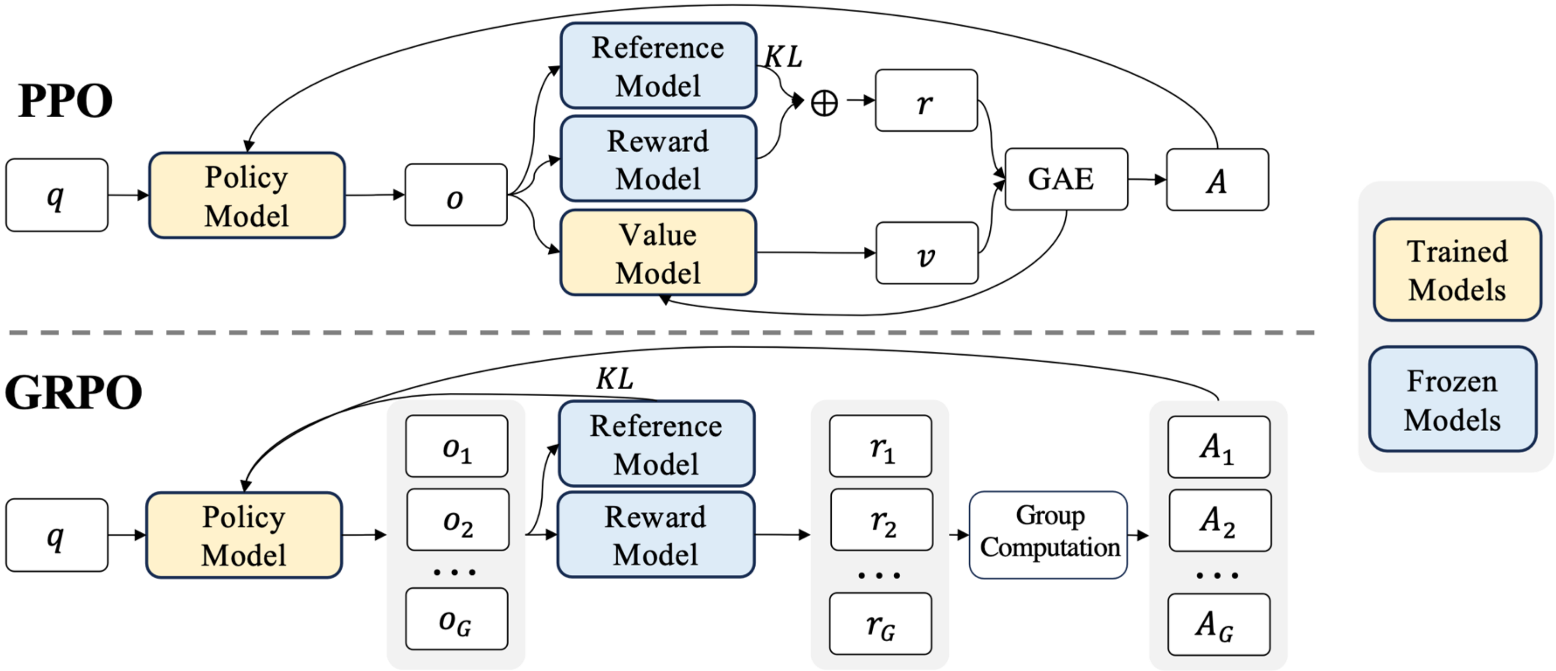
A_i: **Advantage** (z-score); A positive z-score indicates a value above the mean, while a negative z-score indicates a value below the mean.

DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models https://arxiv.org/abs/2402.03300

Key Concepts in GRPO

- **Reference Model (Frozen):**
  - This is usually a copy of the model before RL training (e.g., the SFT model).
  - It ensures the new policy does not diverge too much from the original, maintaining coherence and preventing "reward hacking". (**KL Divergence**)
- **Reward Model (Frozen):**
  - This model takes a prompt and the generated completion and outputs a scalar score representing quality or human preference.
  - It is used to calculate the reward signal that guides the actor (policy) model.
- **Value Model (Learned/Trainable, similar size as policy model):**
  - It predicts the expected cumulative reward from the current position onward, based on the tokens generated from the beginning up to the current position.
  - It is used to compute the advantage function, which determines how much better a specific action is compared to the average, reducing variance in training.
  - Note: Value model is NOT used in GRPO.

Why Value Model is Not Designed in GRPO?



- Value model introduces memory and computational overhead;
- Not suitable for the setting: only the outcome reward is available;
- In long CoT process, the content generated can be revised or contradicted, making it hard to predict final reward based on partial response;

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning https://arxiv.org/abs/2501.12948

DeepSeek R1-Zero: Aha Moment

- The reward signal is solely based on the correctness of final predictions against ground-truth answers, without imposing constraints on the reasoning process itself.

- Pros: Although we do not explicitly teach the model how to reason, it learns improved reasoning strategies through reinforcement learning.

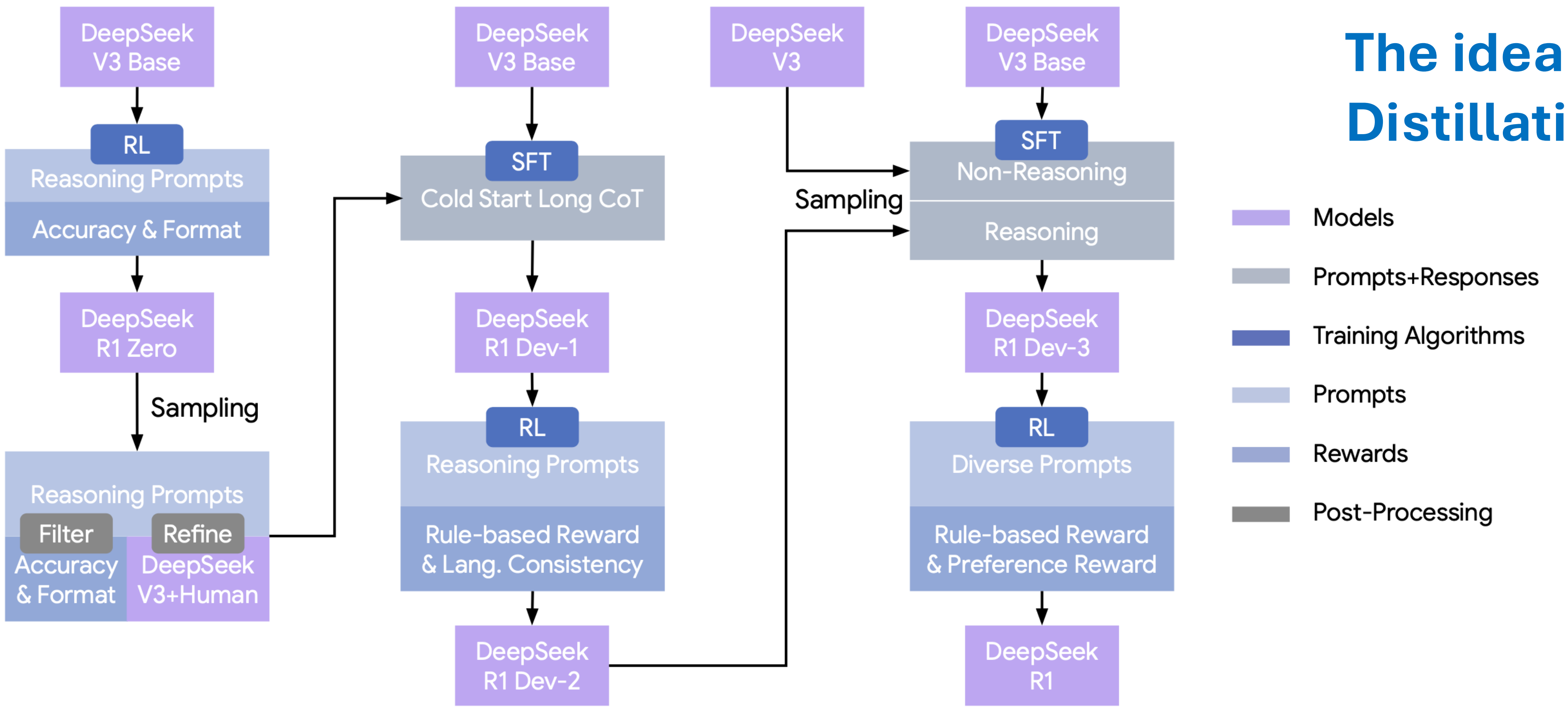
Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

---

Response: <think>
To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$
$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
$\ldots$
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 $\cdots$
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: $\cdots$
$\ldots$

- Cons: DeepSeek R1-Zero has limited performance in broader areas such as writing and open-domain question answering.

The multi-stage pipeline of DeepSeek-R1



The idea of Distillation!

- Non-Reasoning Data: <problem, original response>
- Reasoning Data: <system prompt, problem, R1response>

Rule-based Rewards

- Rule-based Model: For questions that can be validated using specific rules, they adopt a rule-based reward system to determine the feedback.
- **Accuracy rewards** evaluate whether the response is correct.
  - For instance, certain math problems have deterministic results, and they require the model to provide the final answer within a designated format.
- **Format rewards** complement the accuracy reward model by enforcing specific formatting requirements.
  - In particular, the model is incentivized to encapsulate its reasoning process within designated tags, specifically <think> and </think> .

Model-based Rewards: LLM as A Judge

- Helpful Reward Model: The architecture of the reward model is consistent with that of DeepSeek-R1, with the addition of a reward head designed to predict scalar preference scores.

$$Reward_{helpful} = RM_{helpful}(Response_A, Response_B)$$

- Safety Reward Model: The safety reward model was trained to distinguish between safe and unsafe responses.

$$Reward_{safety} = RM_{safety}(Response)$$

Insights from DeepSeek R1

- The RL component enables the model to explore and discover optimal reasoning trajectories for tasks capabilities that cannot be fully realized through human-annotated reasoning traces alone.

- The SFT stage plays a crucial role in tasks where reliable reward signals are difficult to define or model, such as open-ended question answering and creative writing.

- Exclusive reliance on RL can lead to reward hacking and suboptimal behavior in ill-posed tasks, while depending solely on SFT may prevent the model from optimizing its reasoning capabilities through exploration.

- Prioritizing the use of sufficiently large and expressive models when aiming to validate the efficacy of RL from scratch.

How do thinking and reasoning models work?

https://www.youtube.com/watch?v=xCRvOUykOX0

Agentic Reasoning

- LLMs show strong reasoning capabilities in closed-world settings.
- Agentic reasoning bridging thought and action by reframing LLMs as autonomous agents that plan, act, and learn through continual interaction.
- Agentic AI systems examine how agents refine above capabilities through feedback, memory, and adaptation in evolving settings.
- Agentic AI systems **develop in-context reasoning mechanisms** for feedback integration and memory-driven adaptation to navigate evolving environments.

Agentic Reasoning: Post-training and In-context

- Post-training Reasoning targets capability **internalization**: it **consolidates** successful reasoning patterns or tool-use strategies **into the model's weights** via reinforcement learning and fine-tuning.
  - Substantial amounts of compute and time.

- In-context Reasoning focuses on scaling inference-time compute: through structured orchestration, search-based planning, and adaptive workflow design, it enables agents to navigate complex problem spaces dynamically **without modifying model parameters.**

From LLM Reasoning to Agentic Reasoning

Dimension	LLM Reasoning	↔	Agentic Reasoning
Paradigm	passive	↔	interactive
	static input	↔	dynamic context
Computation	single pass	↔	multi step
	internal compute	↔	with feedback
Statefulness	context window	↔	external memory
	no persistence	↔	state tracking
Learning	offline pretraining	↔	continual improvement
	fixed knowledge	↔	self evolving
Goal Orientation	prompt based	↔	explicit goal
	reactive	↔	planning

Agentic Reasoning for Large Language Models https://arxiv.org/abs/2601.12538

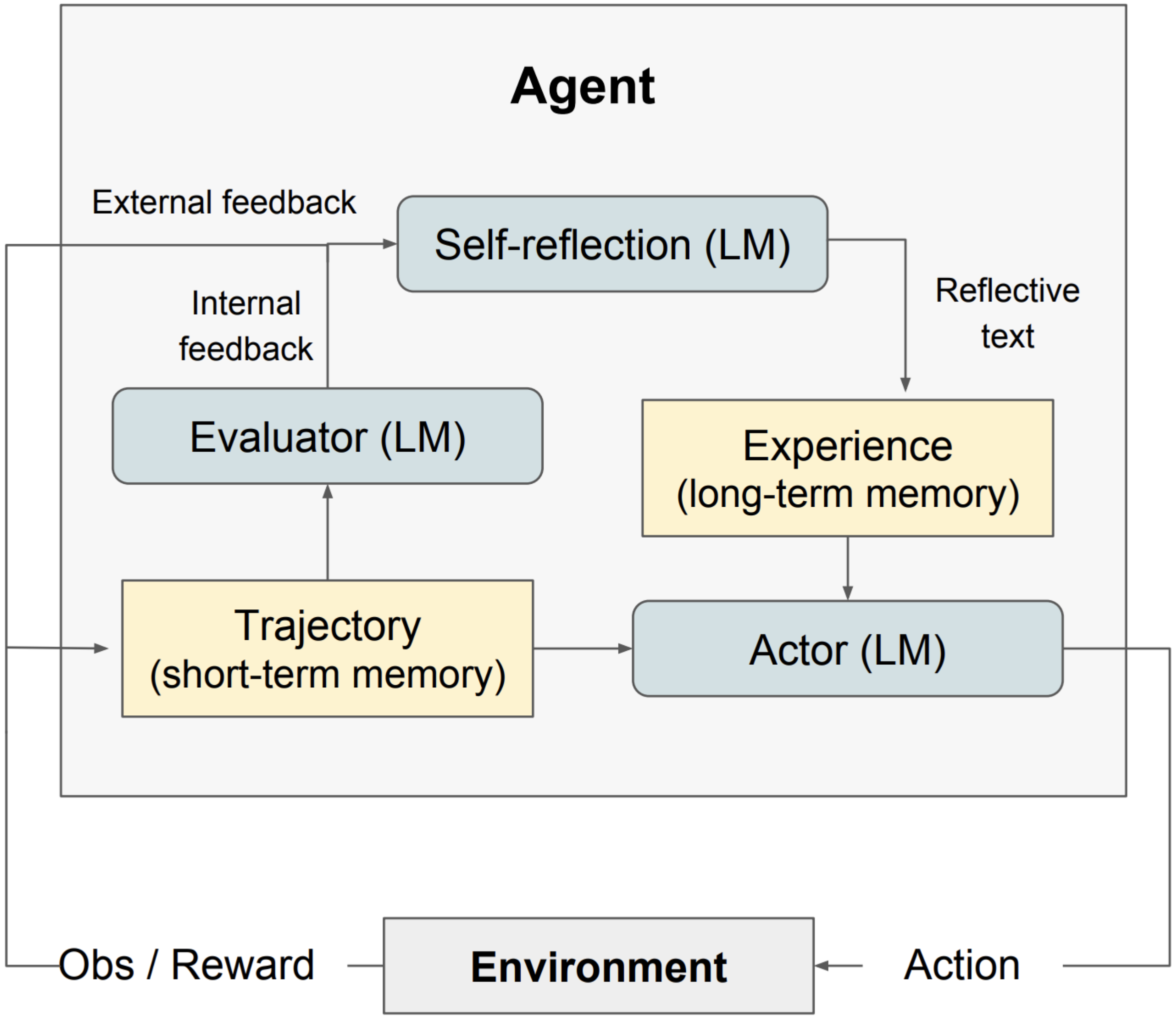
Reflexion

- Reflexion agents verbally reflect on task feedback signals, then maintain their own reflective text in an episodic memory buffer to induce better decision-making in subsequent trials.
  - Using LLMs to generate text and "actions" that can be used in API calls and executed in an environment.
  - Receiving feedback signal:
    - Simple binary environment feedback
    - Pre-defined heuristics for common failure cases
    - Self-evaluation such as binary classification using LLMs (decision-making)
    - Self-written unit tests (programming)
- Reflexion converts feedback signal into verbal feedback in the form of a textual summary, which is then added as additional context for the LLM agent in the next episode.

Reflexion: Language Agents with Verbal Reinforcement Learning https://arxiv.org/abs/2303.11366

Reflexion Mechanism Design

- Actor is built upon a LLM that is specifically prompted to generate the necessary text and actions conditioned on the state observations.
- Evaluator: proving internal feedback.
- Self-reflection: generating verbal reflection.



Reflexion Mechanism Design (Cont’d)

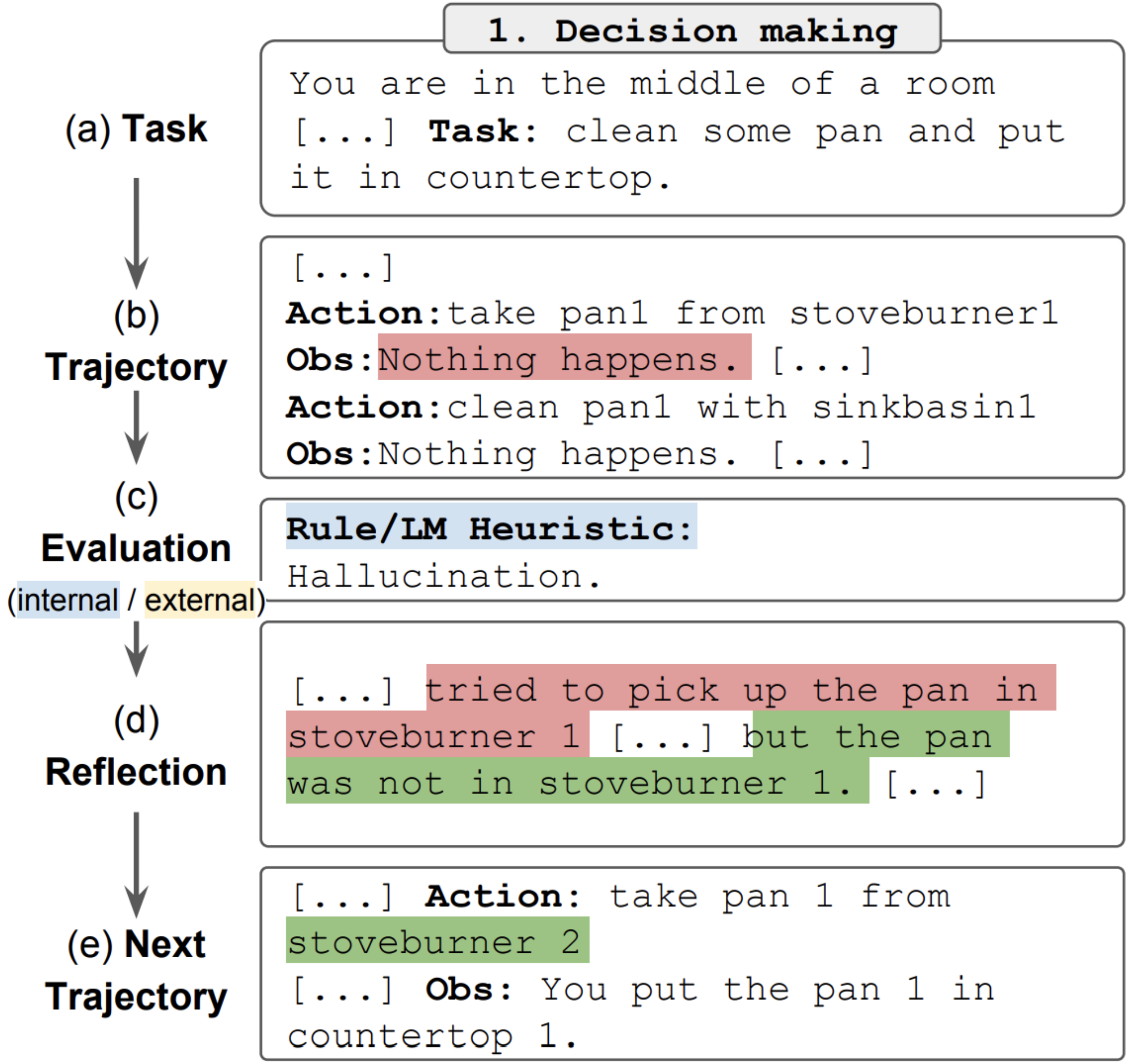
- The Evaluator takes as input a generated trajectory from Actor and computes a reward score that reflects its performance within the given task context.
- Reward functions
  - Exact match (EM) grading, ensuring that the generated output aligns closely with the expected solution.
  - A different instantiation of an LLM itself as an Evaluator, generating rewards for decision-making and programming task.
  - Pre-defined heuristic functions that are tailored to specific evaluation criteria.

Reflexion Mechanism Design (Cont’d)

- The Self-Reflection model instantiated as an LLM, generating verbal self-reflections to provide valuable feedback for future trials.
- Given a sparse reward signal, such as a binary success status (success/fail), the current trajectory, and its persistent memory, the self-reflection model generates nuanced and specific feedback.
- This feedback, which is more informative than scalar rewards, is then stored in the agent’s memory.

Reflexion Example

- The Actor conditions its decisions on short and long-term memory.
- In RL setup, the trajectory history serves as the short-term memory while outputs from the Self-Reflection model are stored in long-term memory.



Pros and Cons of Reflexion

Pros:

- It is lightweight and does not require finetuning the LLM.
  - This self-reflective feedback acts as a “semantic” gradient signal by providing the agent with a concrete direction to improve upon, helping it learn from prior mistakes to perform better on the task.
- It allows for a more explicit and interpretable form of episodic memory over prior experiences

Cons:

- It relies on the power of the LLM’s self-evaluation capabilities (or heuristics) and not having a formal guarantee for success.
- Generating useful reflective feedback is challenging since it requires a good understanding of where the model made mistakes as well as the ability to generate a summary containing actionable insights for improvement.

References

- DeepSeek R1: https://arxiv.org/pdf/2501.12948
- Reflexion: https://arxiv.org/abs/2303.11366
- CoT: https://arxiv.org/pdf/2201.11903
- Agentic Reasoning for Large Language Models
  https://arxiv.org/abs/2601.12538