

# Lecture 20: LLM Test-Time Inference and Reasoning

## CS486/686 Intro to Artificial Intelligence

2026-3-19

Pascal Poupart  
David R. Cheriton School of Computer Science  
CIFAR AI Chair at Vector Institute



# Outline

Improvements to RL from Human Feedback:

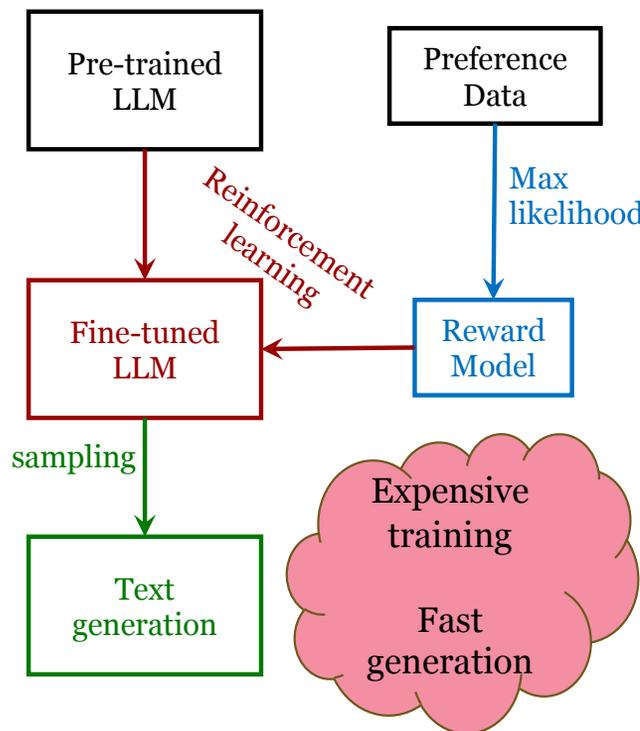
- Direct Preference Optimization
- Reward Guided Text Generation

LLM Reasoning (test time inference):

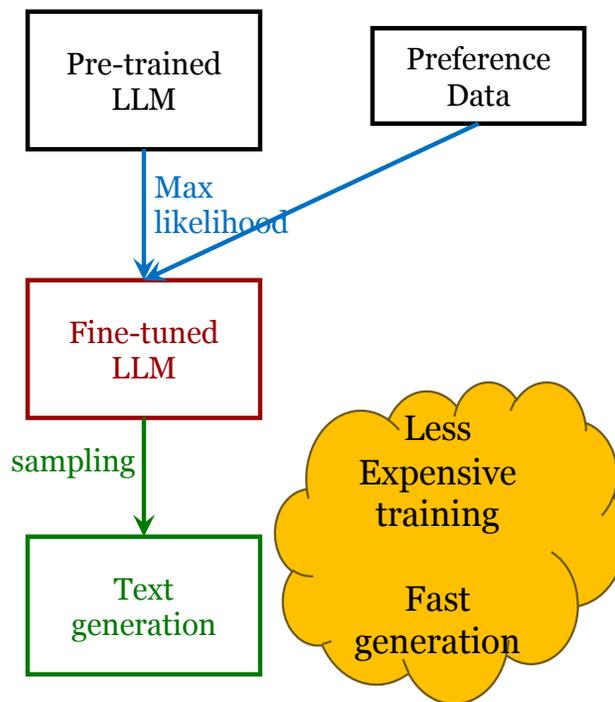
- Chain of thought reasoning
- Reasoning by searching
- Self-reflection

# RLHF Improvements

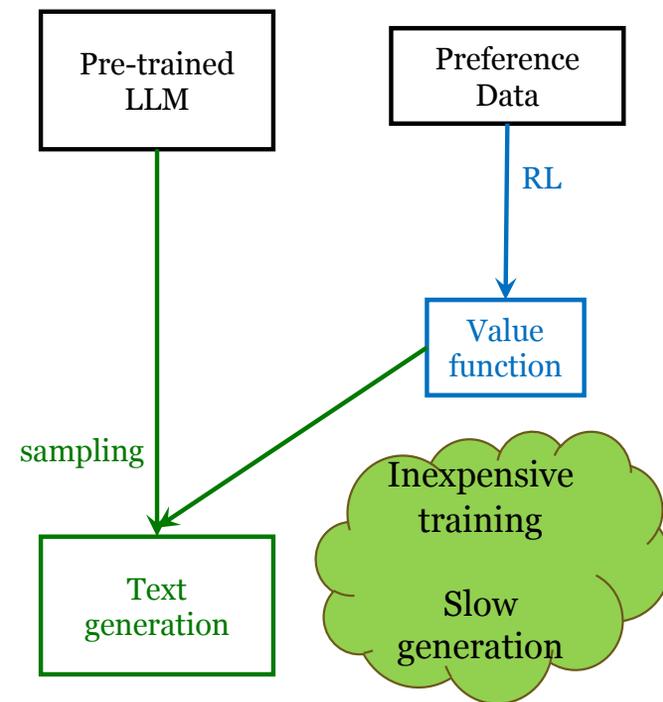
**Proximal Policy Optimization (PPO)**  
Ouyang et al., 2022



**Direct Preference Optimization (DPO)**  
Rafailov et al., 2023



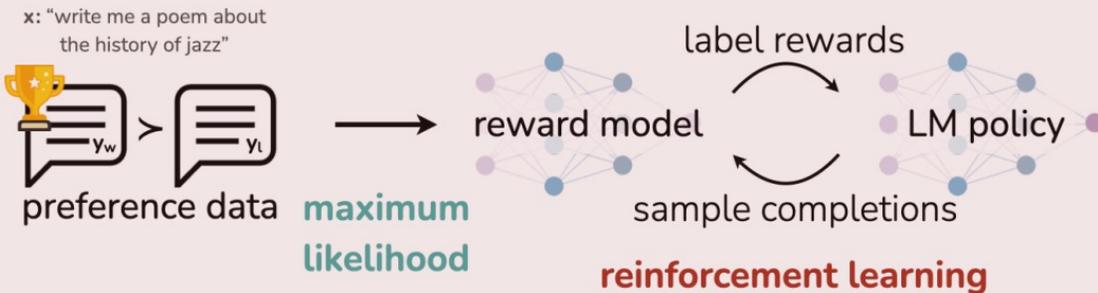
**Reward Guided Text Generation (RGTG)**  
Khanov et al., 2024  
Rashid et al., 2025



# Direct Preference Optimization

Rafailov, Sharma, Mitchell, Ermon, Manning, Finn (2023) **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**, *NeurIPS*.

## Reinforcement Learning from Human Feedback (RLHF)



## Direct Preference Optimization (DPO)



# Bypassing RL

- Recall RL objective:

$$\max_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} [r_{\theta}(s, a)] - \beta KL(\pi_{\phi}(\cdot | s) | \pi_{ref}(\cdot | s)) \right]$$

- Closed form solution (proof on next slide):

$$\pi_{\phi}(a|s) = \frac{1}{Z(s)} \pi_{ref}(a|s) \exp\left(\frac{r_{\theta}(s, a)}{\beta}\right)$$

- Isolate reward:  $r_{\theta}(s, a) = \beta \log \frac{\pi_{\phi}(a|s)}{\pi_{ref}(a|s)} + \beta \log Z(s)$

- Plug into preference objective:

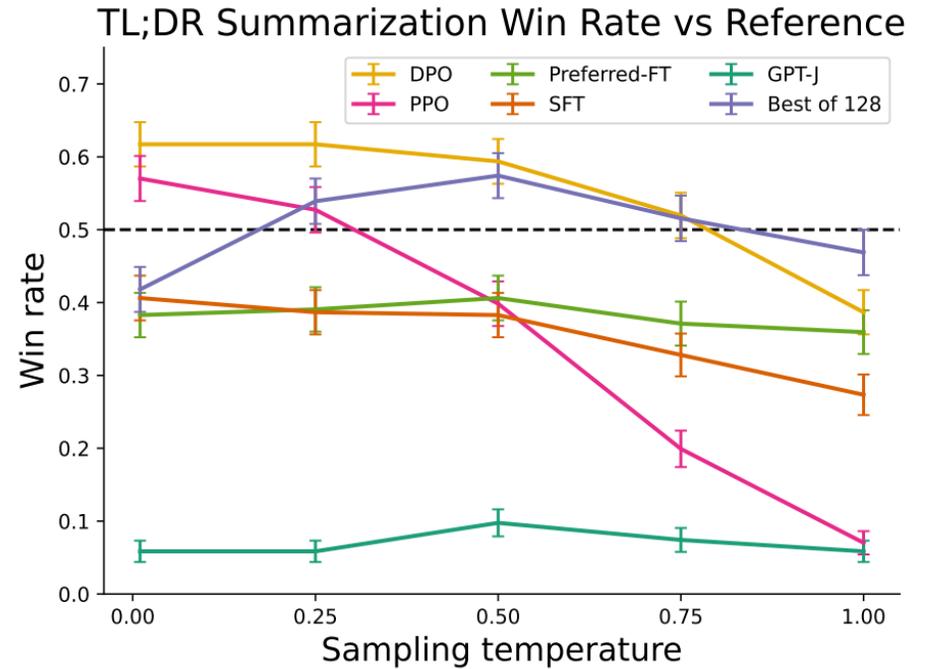
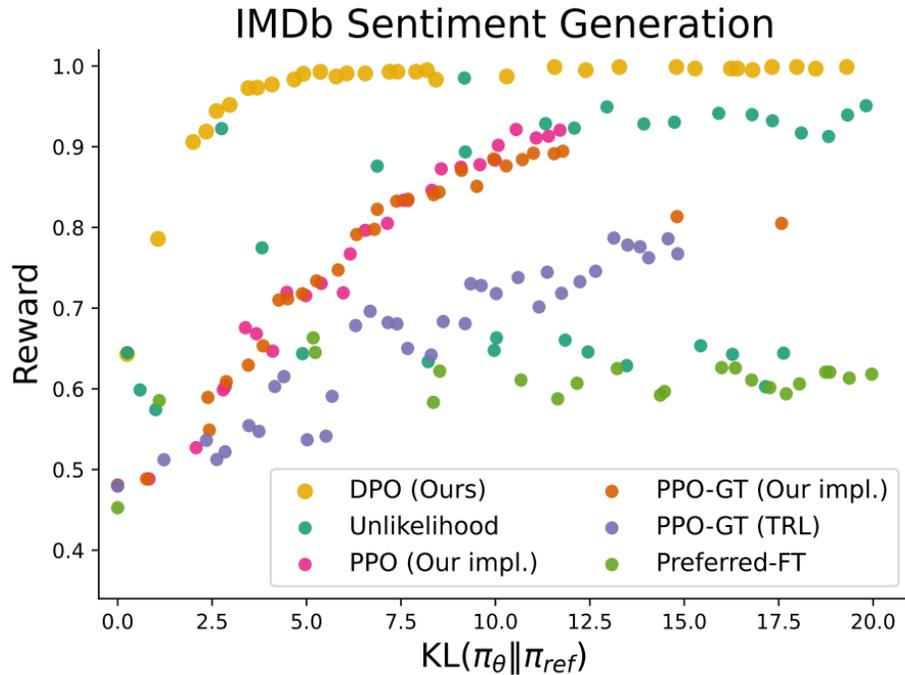
$$\begin{aligned} \text{Loss}(\theta) &= -\frac{1}{\binom{k}{2}} E_{(s, a_i, a_j) \in \text{Dataset}} \log \sigma\left(r_{\theta}(s, a_i) - r_{\theta}(s, a_j)\right) \\ &= -\frac{1}{\binom{k}{2}} E_{(s, a_i, a_j) \in \text{Dataset}} \log \sigma\left(\beta \log \frac{\pi_{\phi}(a_i|s)}{\pi_{ref}(a_i|s)} - \beta \log \frac{\pi_{\phi}(a_j|s)}{\pi_{ref}(a_j|s)}\right) \end{aligned}$$

# Optimal Policy Derivation

$$\begin{aligned}
 & \operatorname{argmax}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} [r_{\theta}(s, a)] - \beta \text{KL}(\pi_{\phi}(\cdot | s) | \pi_{\text{ref}}(\cdot | s)) \right] \\
 &= \operatorname{argmax}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} \left[ r_{\theta}(s, a) - \beta \log \frac{\pi_{\phi}(a|s)}{\pi_{\text{ref}}(a|s)} \right] \right] && \text{by KL definition} \\
 &= \operatorname{argmin}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} \left[ \log \frac{\pi_{\phi}(a|s)}{\pi_{\text{ref}}(a|s)} - \frac{1}{\beta} r_{\theta}(s, a) \right] \right] && \text{since max} = - \text{min} \\
 &= \operatorname{argmin}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} \left[ \log \frac{\pi_{\phi}(a|s)}{\frac{1}{Z(s)} \pi_{\text{ref}}(a|s) \exp\left(\frac{r_{\theta}(s, a)}{\beta}\right)} - \log Z(s) \right] \right] && \text{where } Z(s) = \sum_a \pi_{\text{ref}}(a|s) \exp\left(\frac{r_{\theta}(s, a)}{\beta}\right) \\
 &= \operatorname{argmin}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} \left[ \log \frac{\pi_{\phi}(a|s)}{\frac{1}{Z(s)} \pi_{\text{ref}}(a|s) \exp\left(\frac{r_{\theta}(s, a)}{\beta}\right)} \right] \right] && \text{since } \log Z(s) \text{ is independent of } \phi \\
 &= \operatorname{argmin}_{\phi} E_{s \in \text{Dataset}} \left[ E_{a \sim \pi_{\phi}(a|s)} \left[ \log \frac{\pi_{\phi}(a|s)}{\pi_{\phi^*}(a|s)} \right] \right] && \text{where } \pi_{\phi^*}(a|s) = \frac{1}{Z(s)} \pi_{\text{ref}}(a|s) \exp\left(\frac{r_{\theta}(s, a)}{\beta}\right) \\
 &= \operatorname{argmin}_{\phi} E_{s \in \text{Dataset}} \left[ \text{KL}(\pi_{\phi}(\cdot | s) | \pi_{\phi^*}(\cdot | s)) \right] && \text{by KL definition} \\
 &= \phi^* && \text{since KL is minimized when both arguments are equal}
 \end{aligned}$$

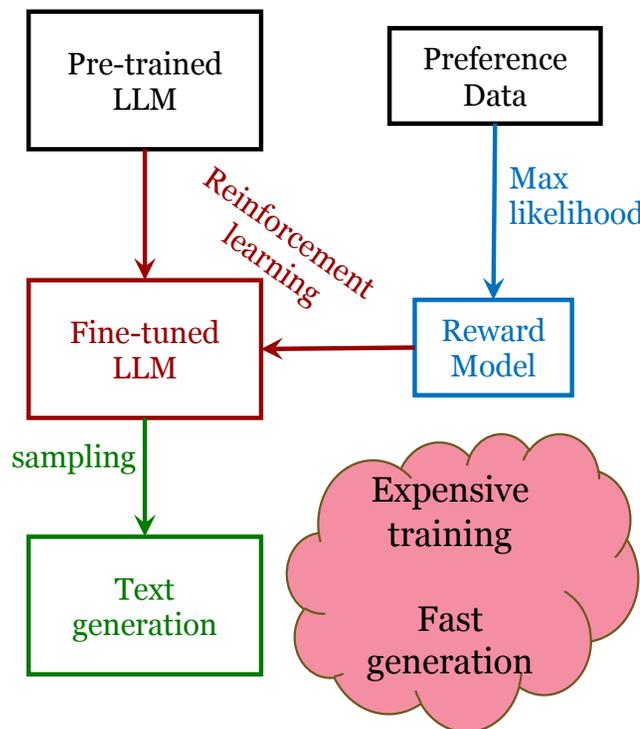
# Empirical Results

Rafailov et al. 2023

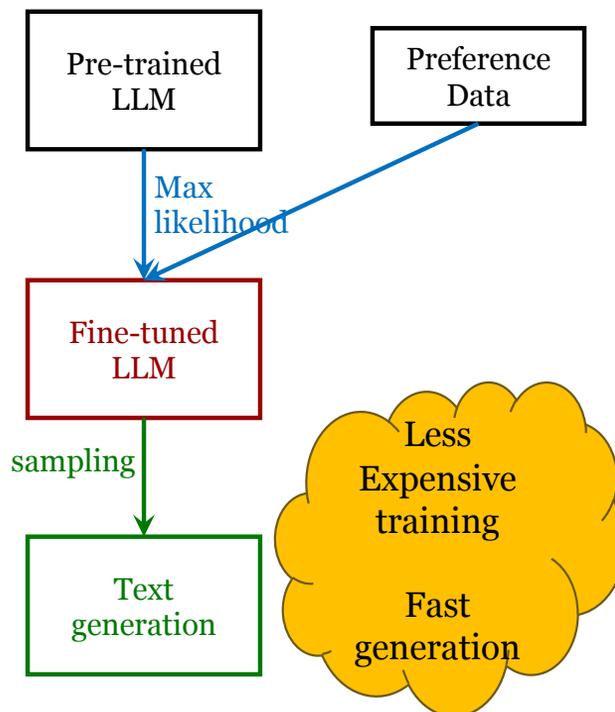


# RLHF Improvements

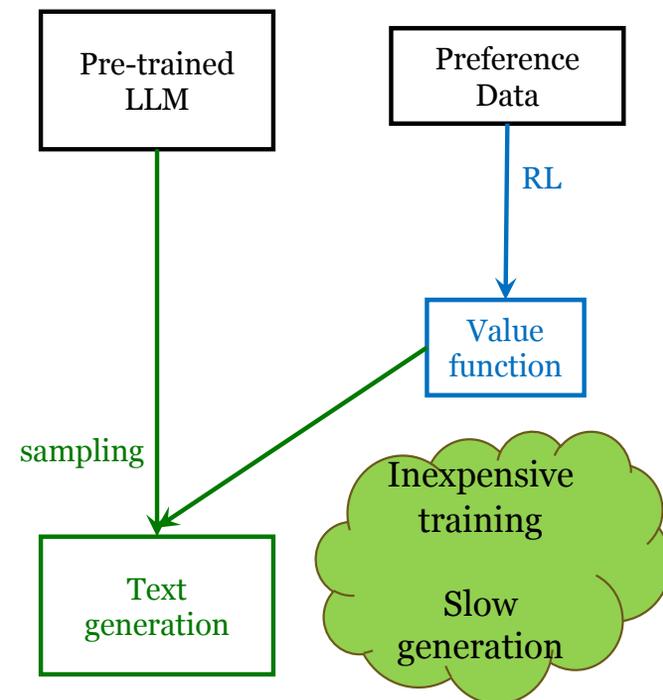
**Proximal Policy Optimization (PPO)**  
Ouyang et al., 2022



**Direct Preference Optimization (DPO)**  
Rafailov et al., 2023



**Reward Guided Text Generation (RGTG)**  
Khanov et al., 2024  
Rashid et al., 2025



# Text Generation

- **Utterance-level RL** (learn reward function)

- State: prompt
- **Action: response**
- Reward: score complete response
- Horizon: 1 time step (no transition function)

- **Token-level RL** (learn Q-function)

- State: [prompt, partial response]
- **Action: next token**
- Reward: score complete response (0 for partial response)
- Horizon: length of response

- **Deterministic transition function:**

$T([\text{prompt, partial response}], \text{next token}) \rightarrow [\text{prompt, partial response, next token}]$

- **$Q(s,a)$ : score next token**

state

action

state'

# Sequence Generation

$$\text{softmax}(x) = \frac{\exp(x)}{\sum_{s'} \exp(x')}$$

- Recall closed form solution

$$\begin{aligned}\pi_{\phi}(\mathbf{a}|\mathbf{s}) &= \frac{1}{Z(\mathbf{s})} \pi_{ref}(\mathbf{a}|\mathbf{s}) \exp\left(\frac{r_{\theta}(\mathbf{s}, \mathbf{a})}{\beta}\right) \\ &= \text{softmax}\left(\log \pi_{ref}(\mathbf{a}|\mathbf{s}) + \frac{r_{\theta}(\mathbf{s}, \mathbf{a})}{\beta}\right)\end{aligned}$$

- Text generation:

$$\mathbf{a} \sim \text{softmax}\left(\log \begin{pmatrix} \pi_{ref}(\mathbf{a}_1|\mathbf{s}) \\ \pi_{ref}(\mathbf{a}_2|\mathbf{s}) \\ \pi_{ref}(\mathbf{a}_3|\mathbf{s}) \\ \dots \\ \pi_{ref}(\mathbf{a}_n|\mathbf{s}) \end{pmatrix} + \begin{pmatrix} r_{\theta}(\mathbf{s}, \mathbf{a}_1) \\ r_{\theta}(\mathbf{s}, \mathbf{a}_2) \\ r_{\theta}(\mathbf{s}, \mathbf{a}_3) \\ \dots \\ r_{\theta}(\mathbf{s}, \mathbf{a}_n) \end{pmatrix} / \beta\right)$$

# Token Generation

- Token-wise LLM modeling

$$\begin{aligned}\pi_{\phi}(a^i | \mathbf{s}, \mathbf{a}^{1:i-1}) &= \frac{1}{Z(\mathbf{s})} \pi_{ref}(a^i | \mathbf{s}, \mathbf{a}^{1:i-1}) \exp\left(\frac{Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i})}{\beta}\right) \\ &= \text{softmax}\left(\log \pi_{ref}(a^i | \mathbf{s}, \mathbf{a}^{1:i-1}) + \frac{Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i})}{\beta}\right)\end{aligned}$$

- Token generation:

$$a^i \sim \text{softmax}\left(\log \begin{pmatrix} \pi_{ref}(a_1^i | \mathbf{s}, \mathbf{a}^{1:i-1}) \\ \pi_{ref}(a_2^i | \mathbf{s}, \mathbf{a}^{1:i-1}) \\ \pi_{ref}(a_3^i | \mathbf{s}, \mathbf{a}^{1:i-1}) \\ \dots \\ \pi_{ref}(a_n^i | \mathbf{s}, \mathbf{a}^{1:i-1}) \end{pmatrix} + \begin{pmatrix} Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i-1}, a_1^i) \\ Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i-1}, a_2^i) \\ Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i-1}, a_3^i) \\ \dots \\ Q_{\theta}(\mathbf{s}, \mathbf{a}^{1:i-1}, a_n^i) \end{pmatrix} / \beta\right)$$

# FaRMA: Faster Reward Model for Alignment

- Rashid, Wu, Fan, Li, Kristiadi, Poupart (2025) **Towards Cost-Effective Reward Guided Text Generation**, *ICML*.
- Optimization problem:

$$\begin{aligned} & \max_{\theta} E_{(s, a_+, a_-) \in Dataset} \log \sigma(Q_{\theta}(s, a_+) - Q_{\theta}(s, a_-)) \\ & \text{Subject to } \forall_{\theta} (s, a^{1:i}) = \max_{a^{i+1:|a|}} \forall_{\theta} (s, [a^{1:i}, a^{i+1:|a|}]) \quad \forall s, a, i \end{aligned}$$

- In practice: alternate between minimizing two loss functions

- $L_1(\theta) = -E_{(s, a_+, a_-) \in Dataset} \log \sigma(Q_{\theta}(s, a_+) - Q_{\theta}(s, a_-))$

- $L_2(\theta) = \frac{1}{2} E_{(s, a) \in Dataset, i \leq |a|} \left( Q_{\theta}(s, a^{1:i}) - \max_{a^{i+1:|a|}} Q_{\theta}(s, [a^{1:i}, a^{i+1:|a|}]) \right)^2$

# FaRMA Pseudocode

Repeat

Repeat for each  $(s, \mathbf{a}_+, \mathbf{a}_-)$  in minibatch

$$L_1(\theta) = \log \sigma(Q_\theta(\mathbf{s}, \mathbf{a}_+) - Q_\theta(\mathbf{s}, \mathbf{a}_-))$$

$$\theta \leftarrow \theta - \alpha \nabla L_1(\theta)$$

Repeat for each  $(\mathbf{s}, \mathbf{a}, i)$  in minibatch

$$L_2(\theta) = \frac{1}{2} \left( Q_\theta(\mathbf{s}, \mathbf{a}^{1:i}) - \max_{\mathbf{a}^{i+1}} Q_\theta(\mathbf{s}, \mathbf{a}^{1:i+1}) \right)^2$$

$$\theta \leftarrow \theta - \alpha \nabla L_2(\theta)$$

# Empirical Results

TL;DR Summarization			
Method	LLM	$r \pm SE$	Time(min)
$\pi_{ref}$	frozen	$0.98 \pm 0.18$	2
ARGS	frozen	$1.46 \pm 0.16$	32
PARGS	frozen	$1.56 \pm 0.19$	31
CD	frozen	$1.15 \pm 0.16$	29
FaRMA	frozen	$2.05 \pm 0.15$	5
CARDS	frozen	$1.73 \pm 0.16$	17
DPO	trained	$2.08 \pm 0.18$	2
PPO	trained	$2.05 \pm 0.14$	2

Table 2. Avg. reward (over 100 samples)  $\pm$  standard error total generation time for the TL;DR summarization task.

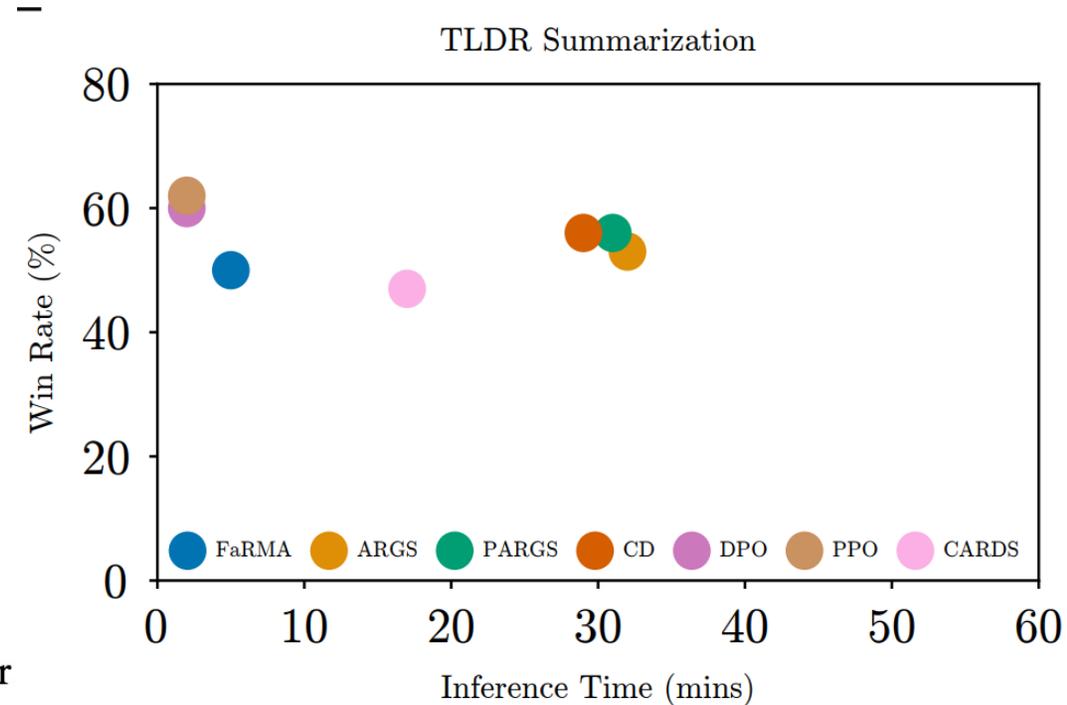


Figure 2. GPT4 evaluation on TLDR

# Reasoning LLMs

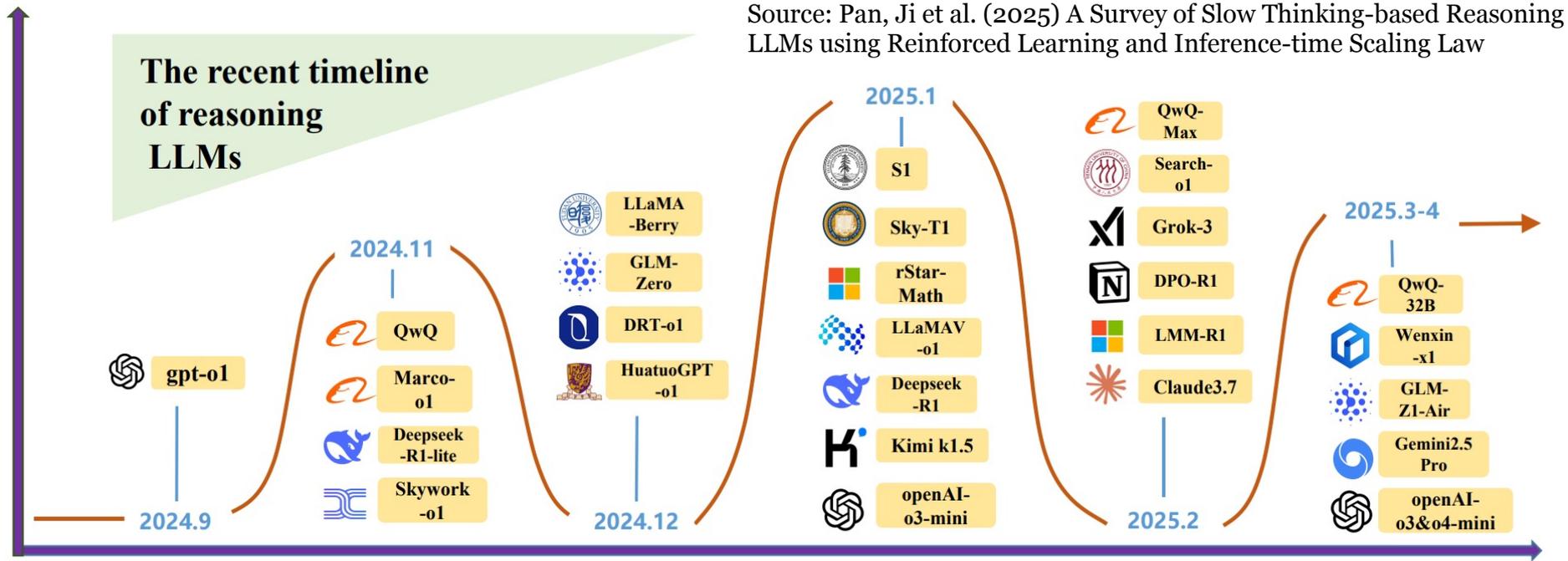


Fig. 1. The timeline of main reasoning LLMs.

# Chain of Thought (CoT) Prompting

## Standard Prompting

### Model Input

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: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain-of-Thought Prompting

### Model Input

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: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅

Credit: <https://cameronwolfe.substack.com/p/chain-of-thought-prompting-for-llms>

# Impact of CoT

Source: Wei et al. (2022)  
Chain of thought prompting  
elicits reasoning in large  
language models.

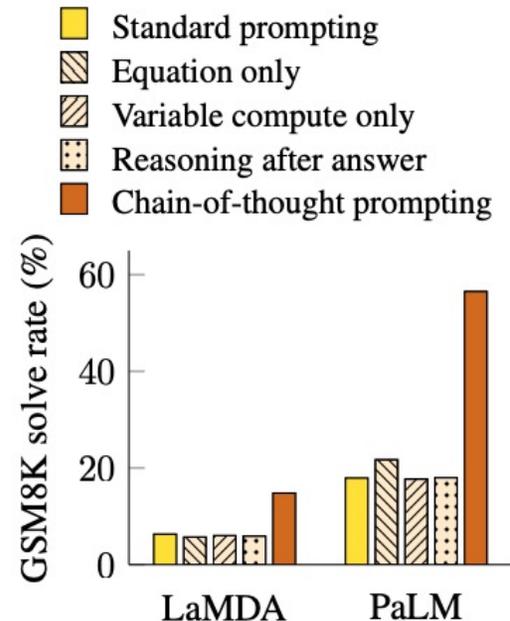
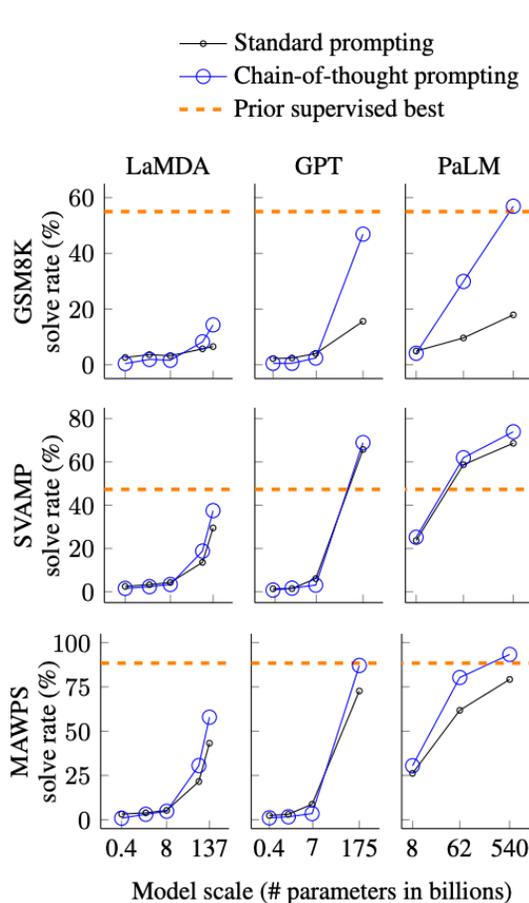
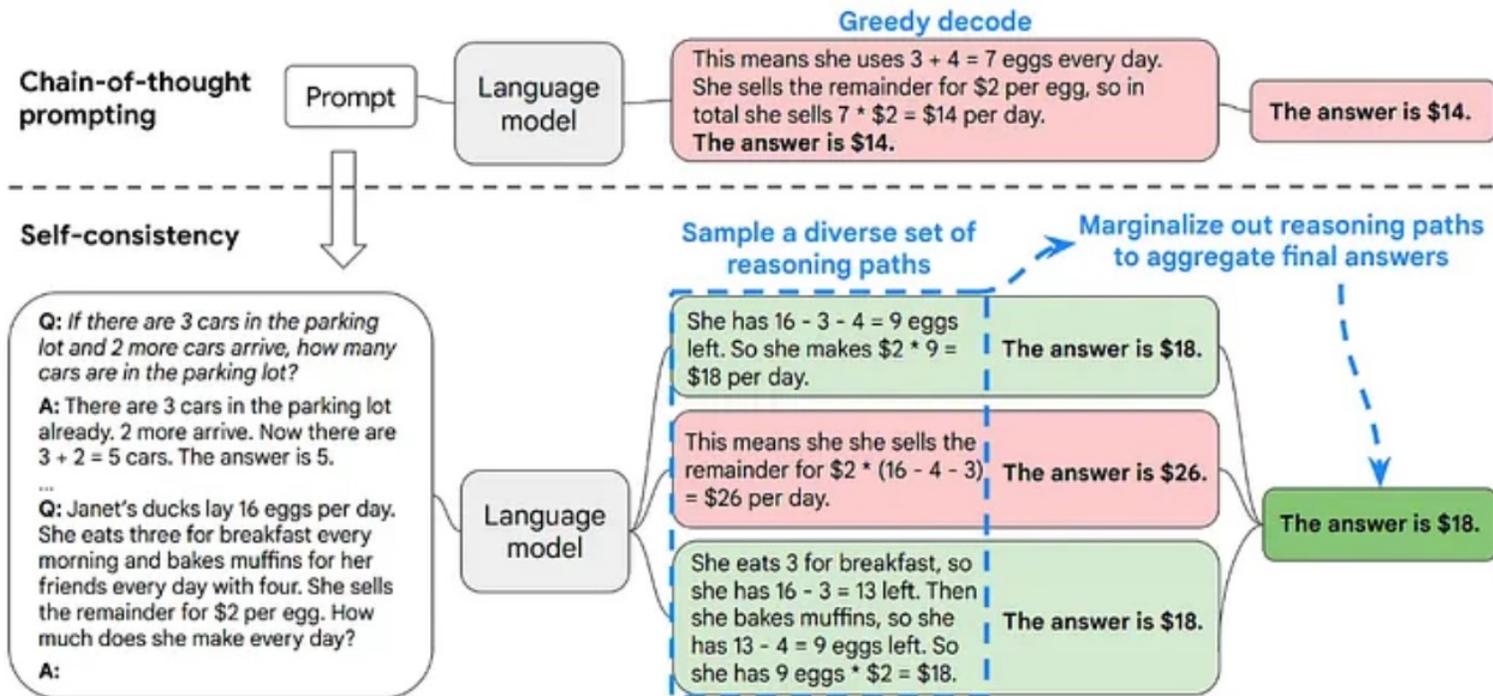


Figure 5: Ablation study for different variations of prompting using LaMDA 137B and PaLM 540B. Results for other datasets are given in Appendix Table 6 and Table 7.

# CoT with Self-Consistency



Credit: <https://cameronrwolfe.substack.com/p/chain-of-thought-prompting-for-llms>

# Reasoning by Searching

Source: Pan, Ji et al. (2025) A Survey of Slow Thinking-based Reasoning LLMs using Reinforced Learning and Inference-time Scaling Law

→ DFS, BFS, A\*, etc.

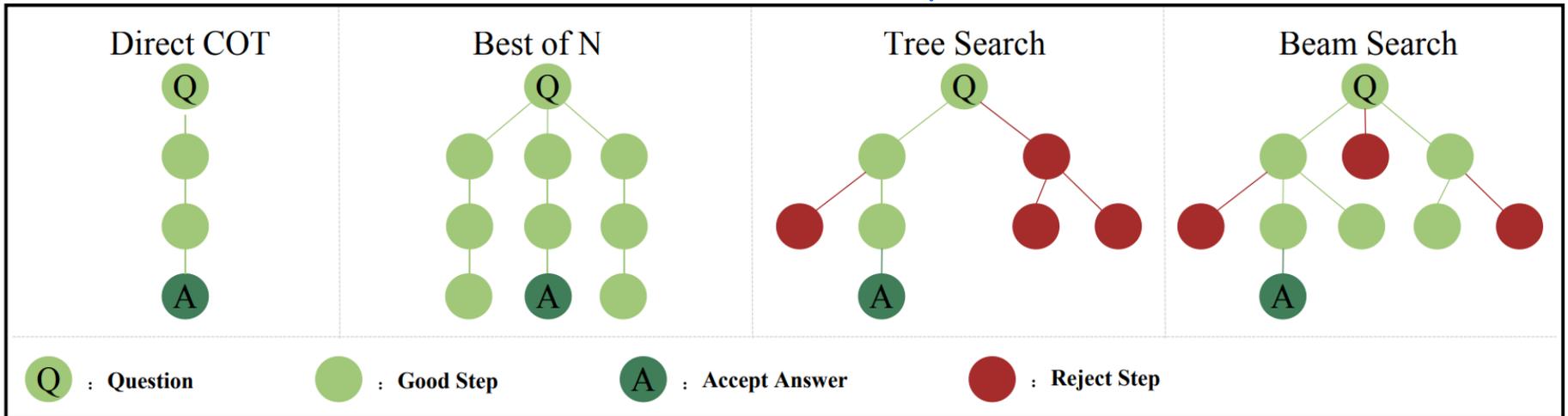


Fig. 3. The search algorithms for test-time scaling

# Learning to Reason

Source: Pan, Ji et al. (2025) A Survey of Slow Thinking-based Reasoning LLMs using Reinforced Learning and Inference-time Scaling Law

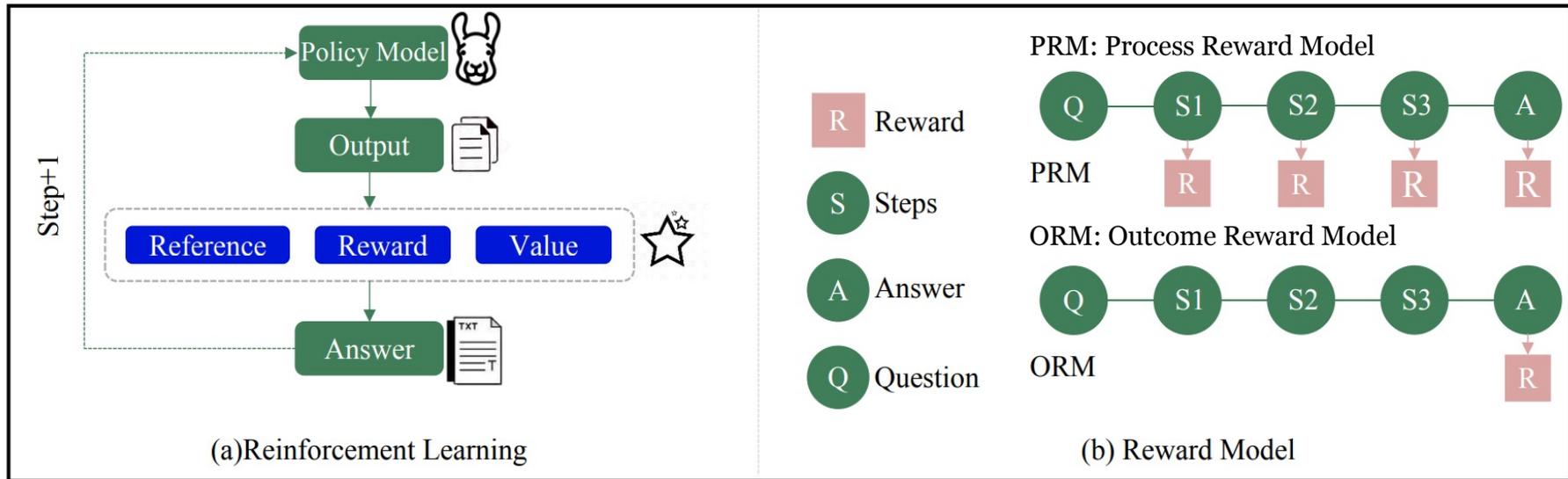


Fig. 4. The reinforcement learning framework and reward model

# Reflexion: Verbalized Reinforcement Learning

Source: Shinn, Cassano et al. (2023) Reflexion: Language Agents with Verbal Reinforcement Learning

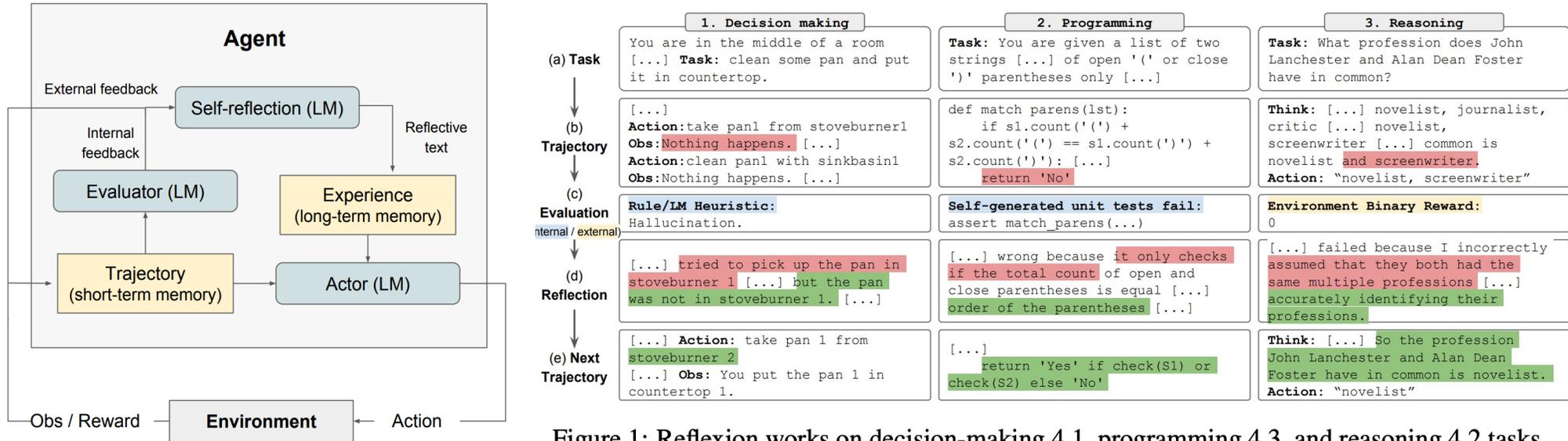


Figure 1: Reflexion works on decision-making 4.1, programming 4.3, and reasoning 4.2 tasks.

# Improved Reasoning by Self-Reflection

Source: Shinn, Cassano et al. (2023) Reflexion: Language Agents with Verbal Reinforcement Learning

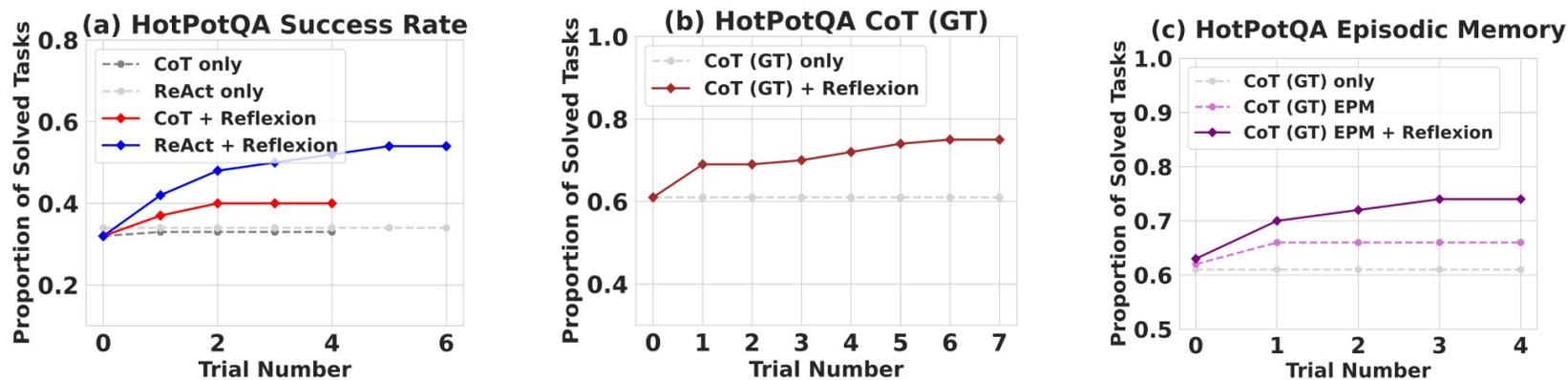


Figure 4: Chain-of-Thought (CoT) and ReAct. Reflexion improves search, information retrieval, and reasoning capabilities on 100 HotPotQA questions. (a) Reflexion ReAct vs Reflexion CoT (b) Reflexion CoT (GT) for reasoning only (c) Reflexion vs episodic memory ablation.

# Inference Time Reasoning

Source: Ke, Jiao et al. (2025) A Survey of Frontiers in LLM Reasoning

