



AI智能体与思维链

清华大学电子工程系 徐丰力

• 高等机器学习课程, 2025.05.10



智能体工作流 (AI Agentic Workflow)

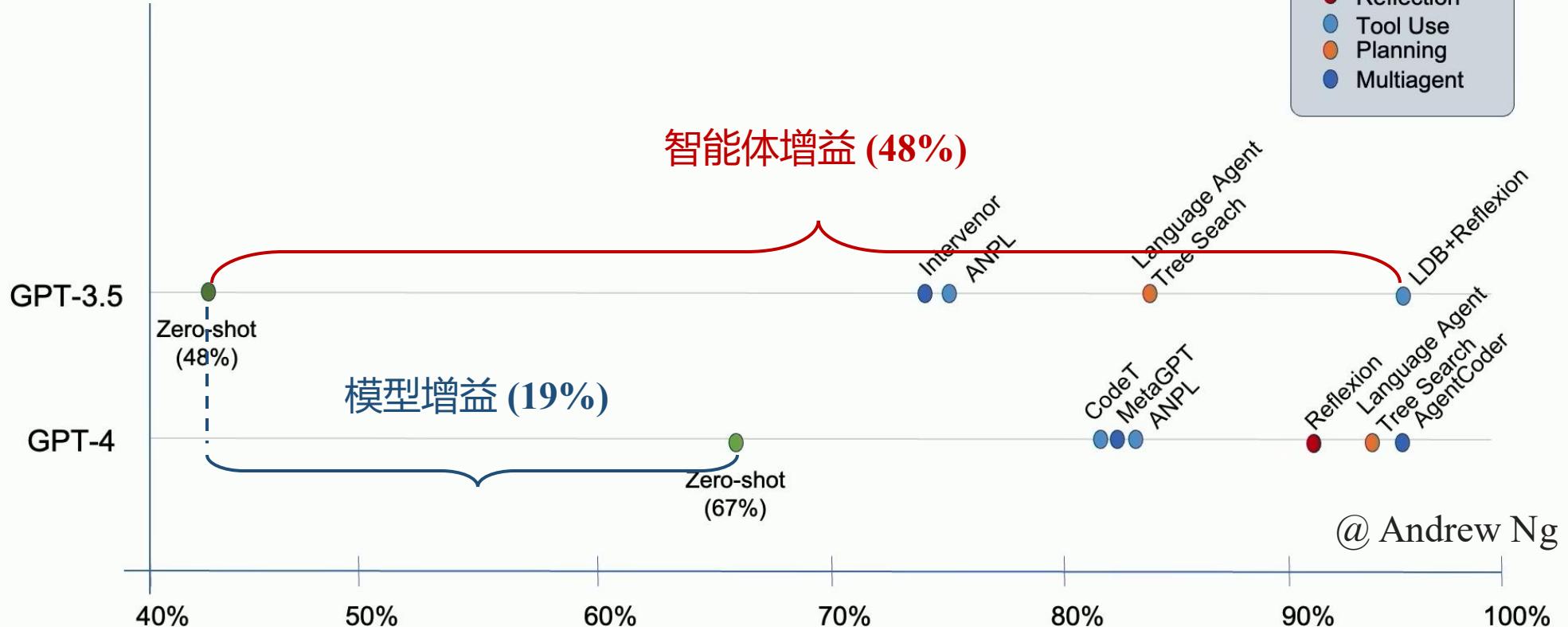


智能模型



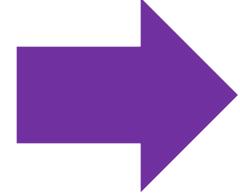
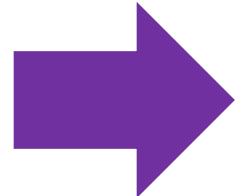
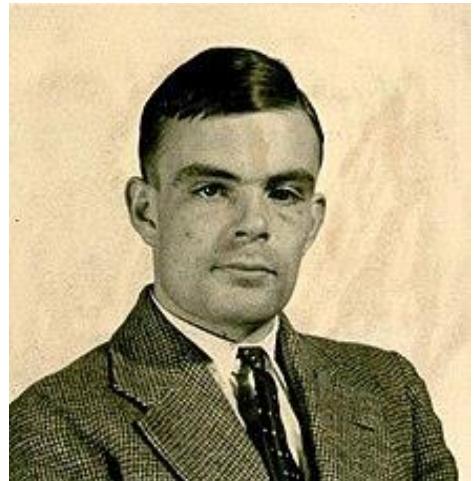
智能体

Coding benchmark (HumanEval)



用自然语言编程大模型智能体 (LLM Agent) 的潜力巨大。

机器如何思考?



Alan Turing
纸带+电子机械装置

Geoffrey Hinton
参数化神经网络

LLM Agent
自然语言编程智能?

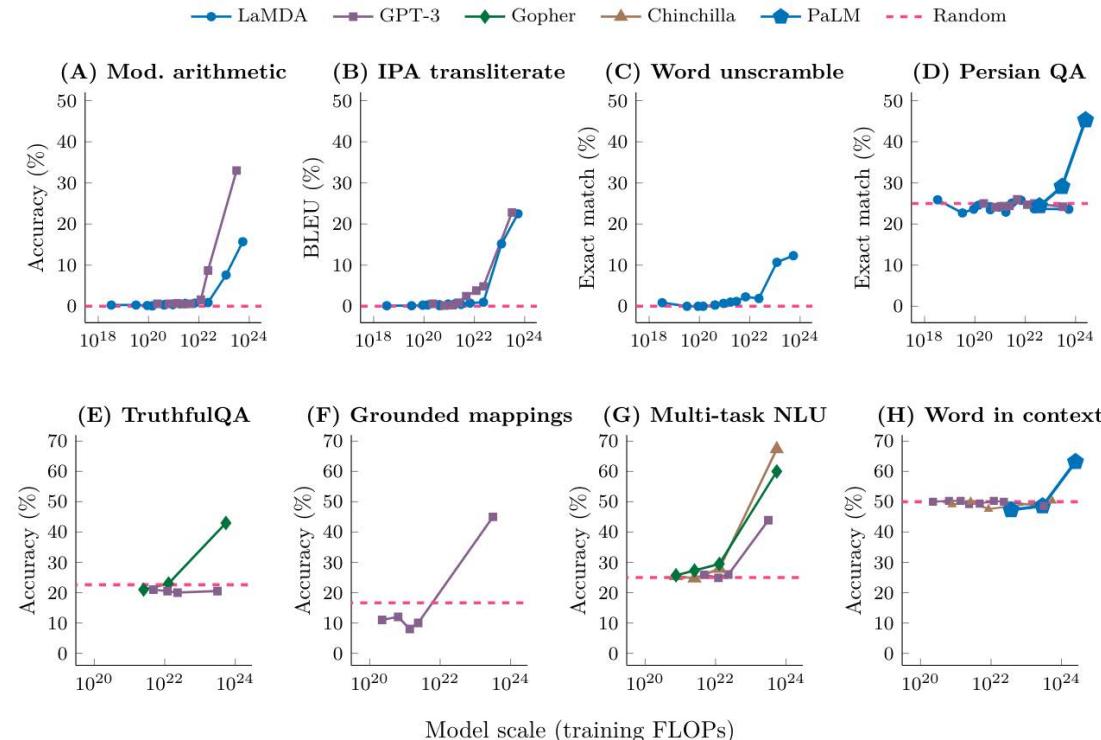
思维链推理

大模型涌现的推理时能力

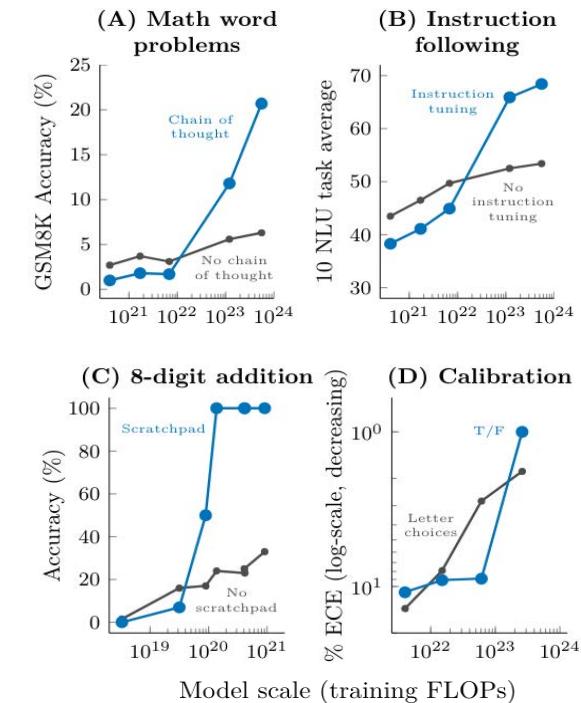
The first work discovering and defining the emergence reasoning ability of LLMs

Definition of Emergent Abilities:

- An ability is emergent if it is **not present in smaller models but is present in larger models**



Emergence in the few-shot prompting setting



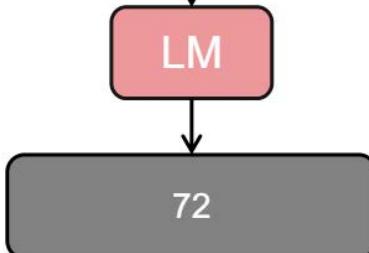
Emergence of some specialized prompting or finetuning methods

大模型涌现的推理时能力

standard prompting

$$p(\mathcal{A} | \mathcal{T}, \mathcal{Q}) = \prod_{i=1}^{|\mathcal{A}|} p_{\text{LM}}(a_i | \mathcal{T}, \mathcal{Q}, a_{<i})$$

Q: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
T: The answer is :



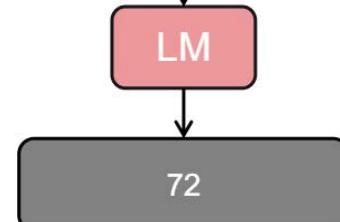
few-shot prompting

$$\mathcal{T} = \{(\mathcal{Q}_i, \mathcal{A}_i)\}_{i=1}^{\mathcal{K}}$$

$$p(\mathcal{A} | \mathcal{T}, \mathcal{Q}) = \prod_{i=1}^{|\mathcal{A}|} p_{\text{LM}}(a_i | \mathcal{T}, \mathcal{Q}, a_{<i})$$

Q: There are 3 cars in the parking lot and 2 more cars arrive. How many cars are in the parking lot?
A: The answer is 5.
.....

Q: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
A: The answer is :



CoT prompting

$$p(\mathcal{A} | \mathcal{T}, \mathcal{Q}) = p(\mathcal{A} | \mathcal{T}, \mathcal{Q}, \mathcal{C}) p(\mathcal{C} | \mathcal{T}, \mathcal{Q})$$

$$p(\mathcal{C} | \mathcal{T}, \mathcal{Q}) = \prod_{i=1}^{|\mathcal{C}|} p_{\text{LM}}(c_i | \mathcal{T}, \mathcal{Q}, c_{<i})$$

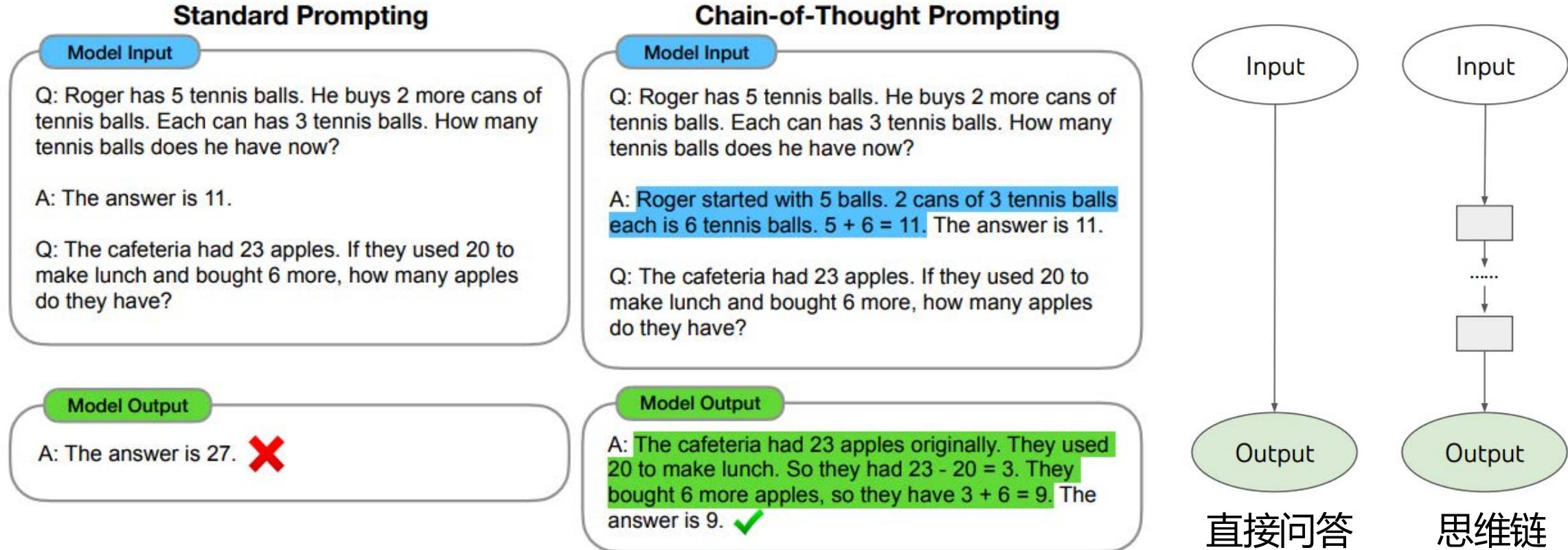
$$p(\mathcal{A} | \mathcal{T}, \mathcal{Q}, \mathcal{C}) = \prod_{j=1}^{|\mathcal{A}|} p_{\text{LM}}(a_j | \mathcal{T}, \mathcal{Q}, \mathcal{C}, a_{<j})$$

Q: There are 3 cars in the parking lot and 2 more cars arrive. How many cars are in the parking lot?
C: There are 3 cars in the parking lot already. 2 more arrive. Now there are $3 + 2 = 5$ cars.
A: The answer is 5.
.....

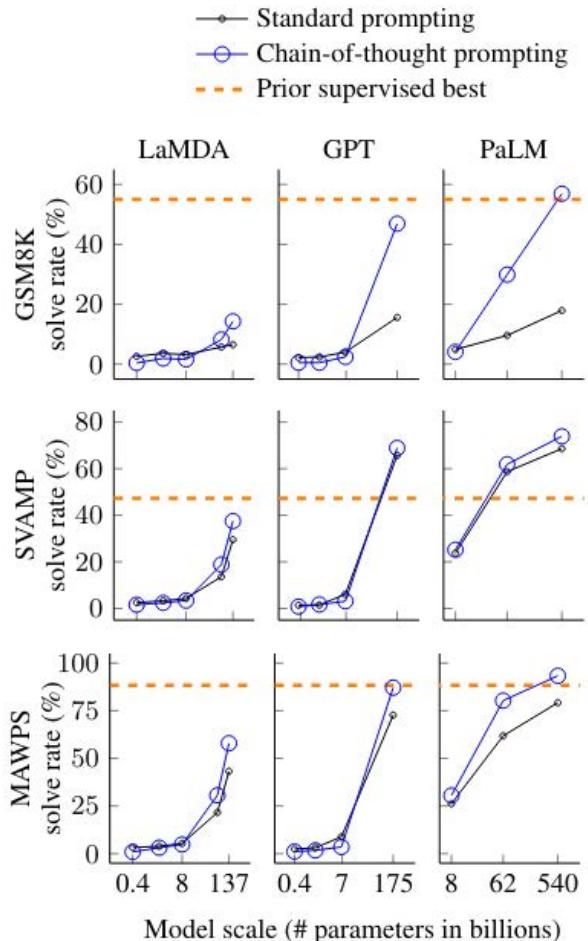
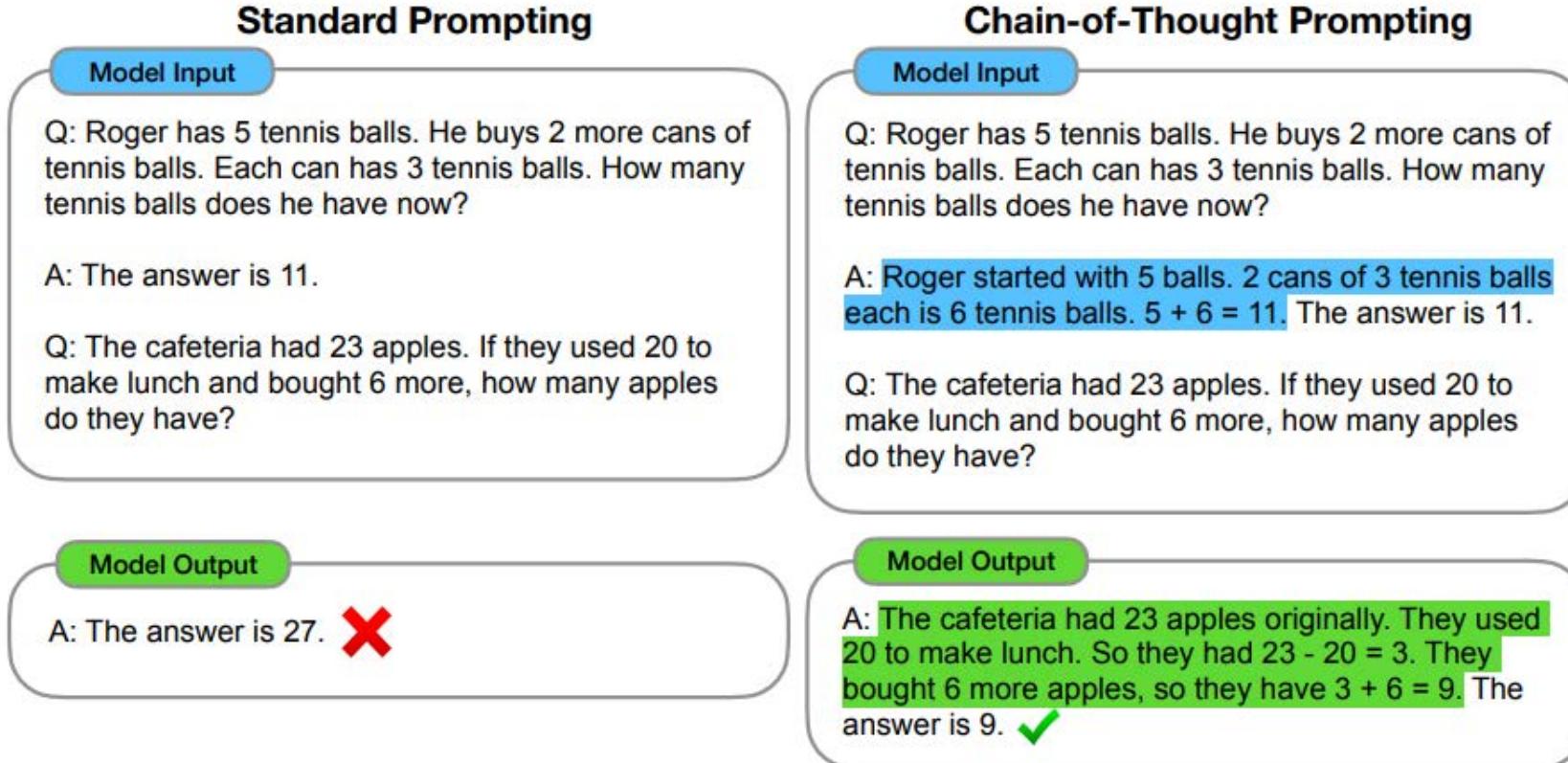
Q: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?

C: Natalia sold $48 / 2 = 24$ clips in May. Altogether, Natalia sold $48 + 24 = 72$ clips in April and May.
A: The answer is 72.

思维链技术



思维链技术



以Chain-of-thought (CoT) 为代表的大模型智能体工作流 (Agentic Workflows)

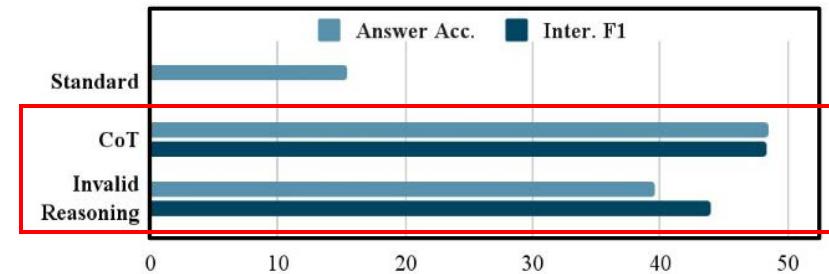
Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." Advances in neural information processing systems 35 (2022): 24824-24837.

思维链推理的经验分析

Main findings:

- Prompting with invalid reasoning steps can achieve over 80-90% of CoT's performance
- Being **relevant to the query** and **correctly ordering the reasoning steps (coherence)** are the key for the effectiveness of CoT

| | In-context Demonstration | Inference by LLM |
|-------------------|--|--|
| Query | Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total? | Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read? |
| Standard | The answer is 39. | The answer is 18. ✗ |
| CoT | Originally, Leah had 32 chocolates and her sister had 42. So in total they had $32 + 42 = 74$. After eating 35, they had $74 - 35 = 39$ pieces left in total. The answer is 39. | Julie is reading a 120-page book. Yesterday, she read 12 pages and today, she read 24 pages. So she read a total of $12 + 24 = 36$ pages. Now she has $120 - 36 = 84$ pages left. Since she wants to read half of the remaining pages, she should read $84 / 2 = 42$ pages. The answer is 42. ✓ |
| Invalid Reasoning | Originally, Leah had 32 chocolates and her sister had 42. So her sister had $42 - 32 = 10$ chocolates more than Leah has. After eating 35, since $10 + 35 = 45$, they had $45 - 6 = 39$ pieces left in total. The answer is 39. | Yesterday, Julie read 12 pages. Today, she read $12 * 2 = 24$ pages. So she read a total of $12 + 24 = 36$ pages. Now she needs to read $120 - 36 = 84$ more pages. She wants to read half of the remaining pages tomorrow, so she needs to read $84 / 2 = 42$ pages tomorrow. The answer is 42. ✓ |



| | GSM8K | | | Bamboogle | |
|---------------------------------------|---------------|-----------|-------------|---------------|-----------|
| | Inter. Recall | Inter. F1 | Answer Acc. | Inter. Recall | Answer F1 |
| STD (Standard prompting) | N/A | N/A | 15.4 | N/A | 20.6 |
| CoT (Chain-of-Thought prompting) | 43.9 | 48.3 | 48.5 | 45.2 | 45.2 |
| ① Invalid Reasoning | 39.8 | 43.9 | 39.5 | 44.4 | 39.4 |
| ② No coherence for bridging objects | 35.3 | 39.2 | 35.8 | 40.8 | 37.4 |
| ③ No relevance for bridging objects | 21.4 | 26.2 | 27.5 | 39.6 | 34.0 |
| ④ No coherence for language templates | 24.1 | 28.3 | 25.8 | 35.2 | 32.1 |
| ⑤ No relevance for language templates | 29.5 | 34.0 | 32.8 | 40.4 | 29.4 |
| ⑥ No coherence | 25.2 | 29.4 | 23.1 | 39.6 | 33.8 |
| ⑦ No relevance | 9.6 | 11.9 | 11.0 | 36.8 | 23.9 |

Keeping relevance is the most crucial

Zero-shot CoT

- ❑ “Let’s think step by step” is all you need
- ❑ Outperforms few-shot settings but underperforms few-shot-CoT

(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. **X**

(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 **X**

(d) Zero-shot-CoT (Ours)

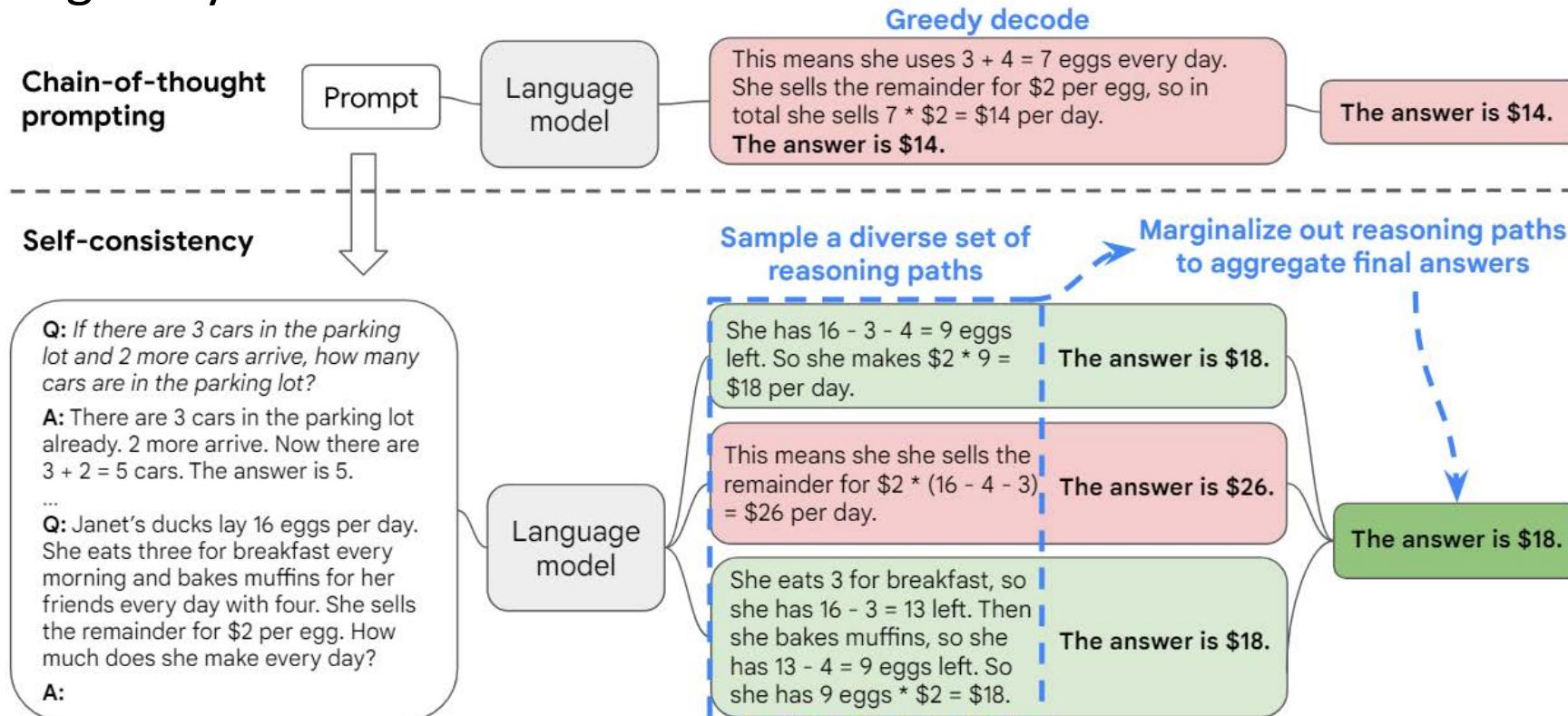
Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

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.* ✓

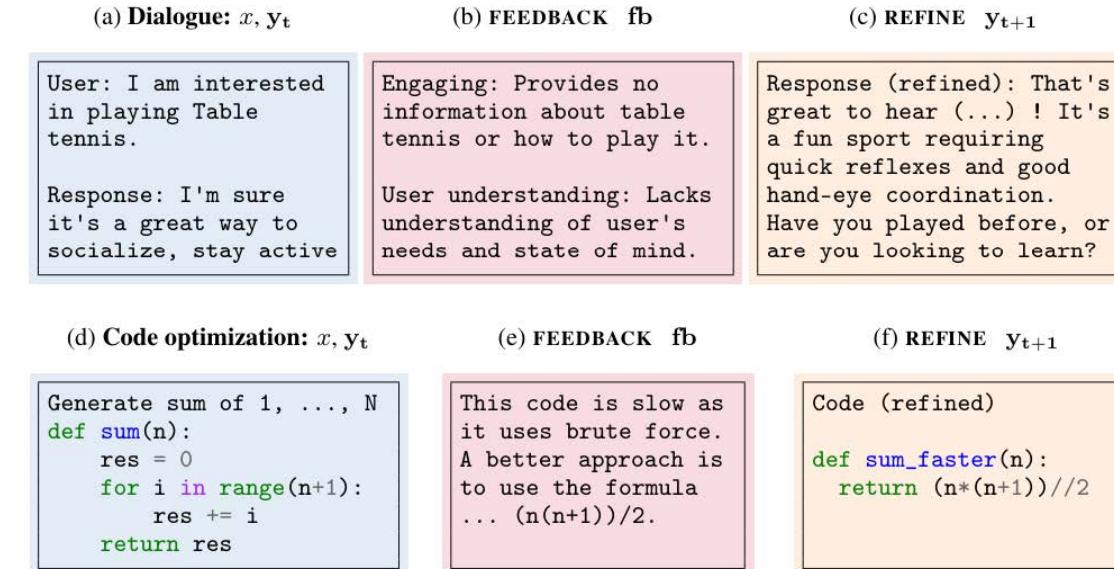
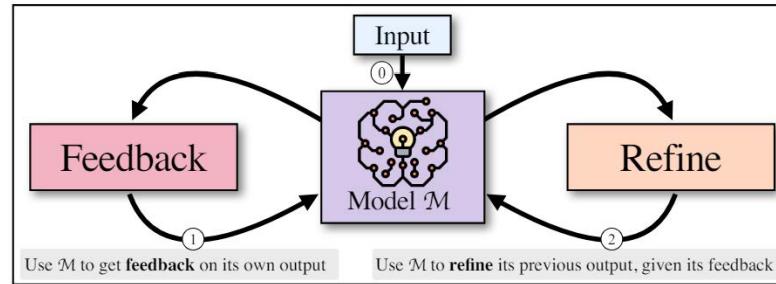
Self-consistent CoT

- Complex reasoning tasks typically admit multiple reasoning paths that reach a correct answer
- Sample a diverse set of reasoning paths instead of only taking the greedy one

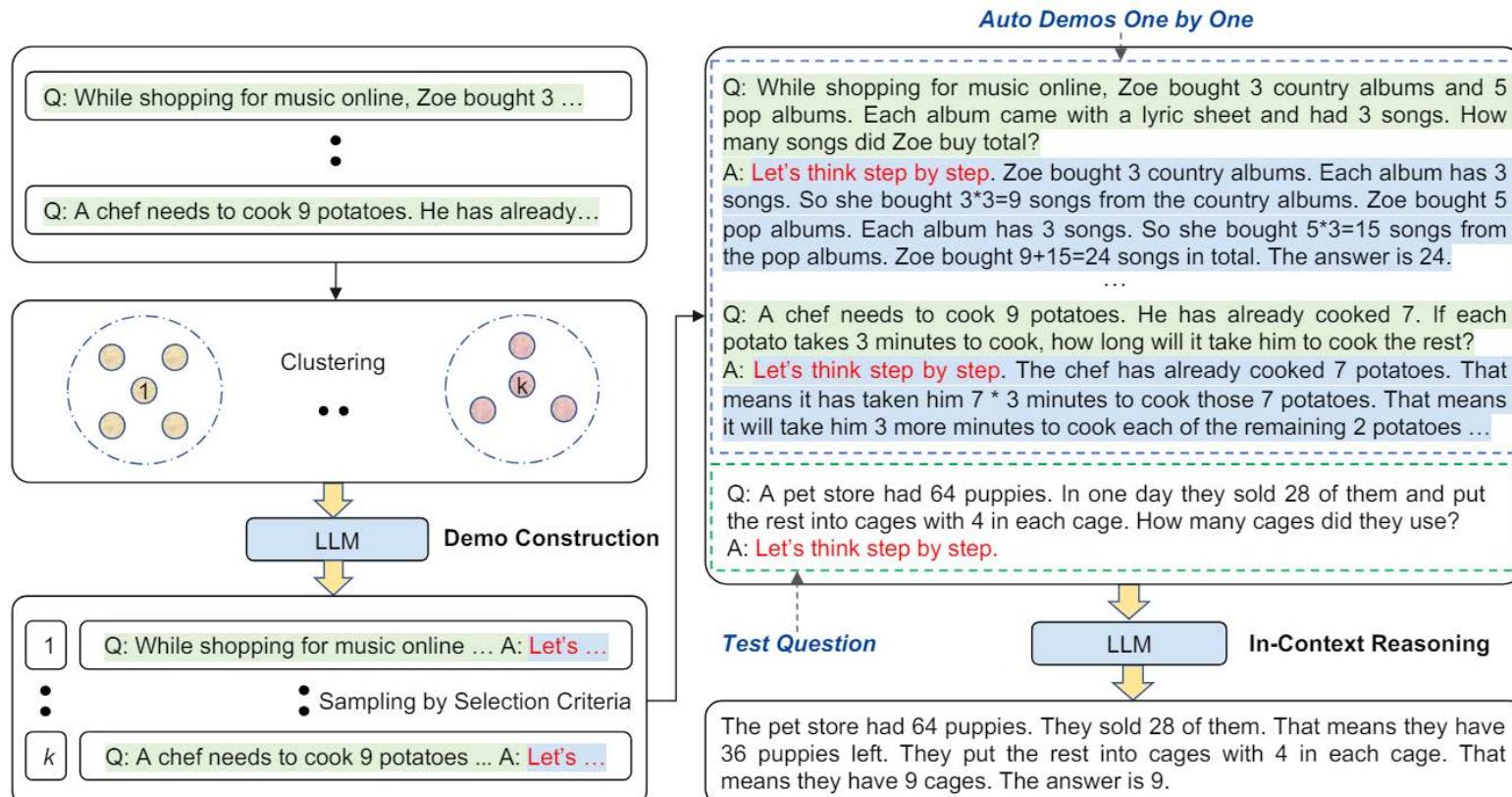


Self-refine

- ☐ Iteratively provide self-feedback and refine the model's own outputs (within a single LLM)
- ☐ Smaller LMs (e.g., Vicuna-13B) can not obtain much benefit with Self-refine, due to the low feedback quality or hallucination

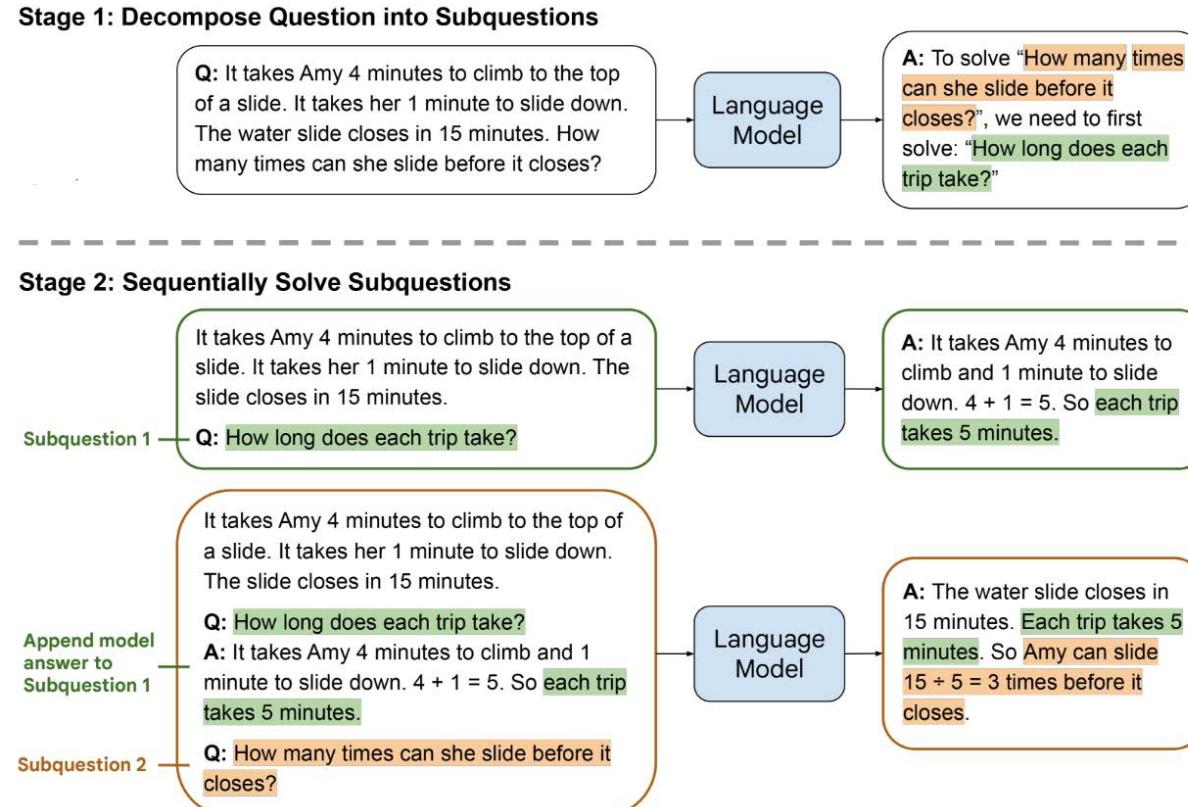


- Zero-shot CoT: low performance, Manual-CoT: high efforts
- Construct **diverse demonstrations** with question clustering and sampling
- Near the performance of Manual-CoT



Least-to-most prompting

- LLM tends to perform poorly on tasks which requires solving problems harder than the exemplars in prompts
- **Decompose** a complex problem into simpler sub-problems and reason stage by stage

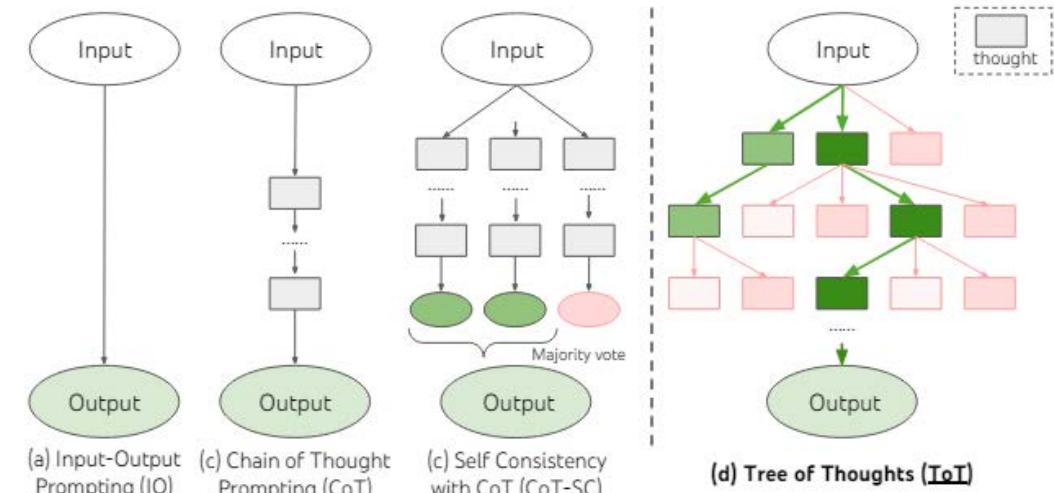
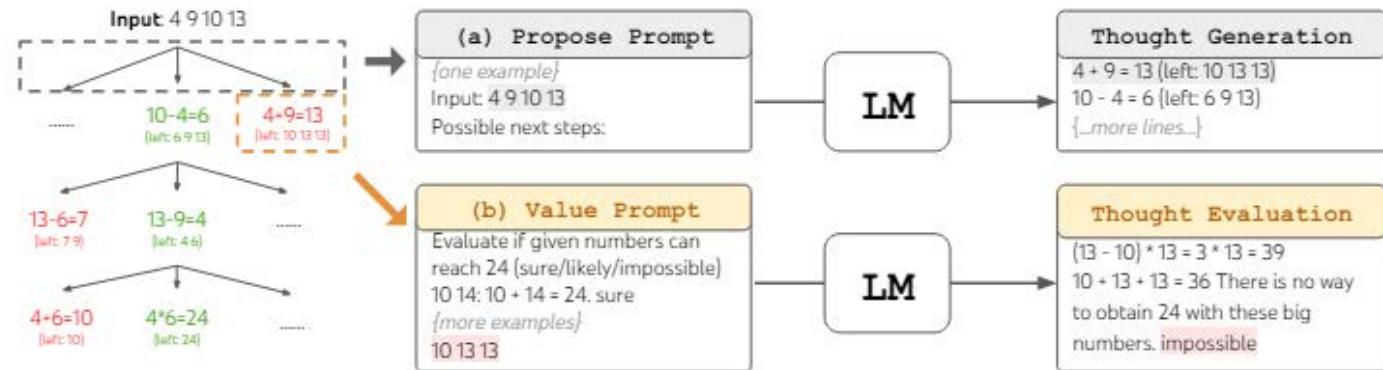


Tree-of-Thoughts (ToT)

- Perform deliberate decision making by considering **multiple different reasoning paths** and **self-evaluating** multiple choices to decide the next action
- Achieved with tree-searching methods (BFS, DFS)

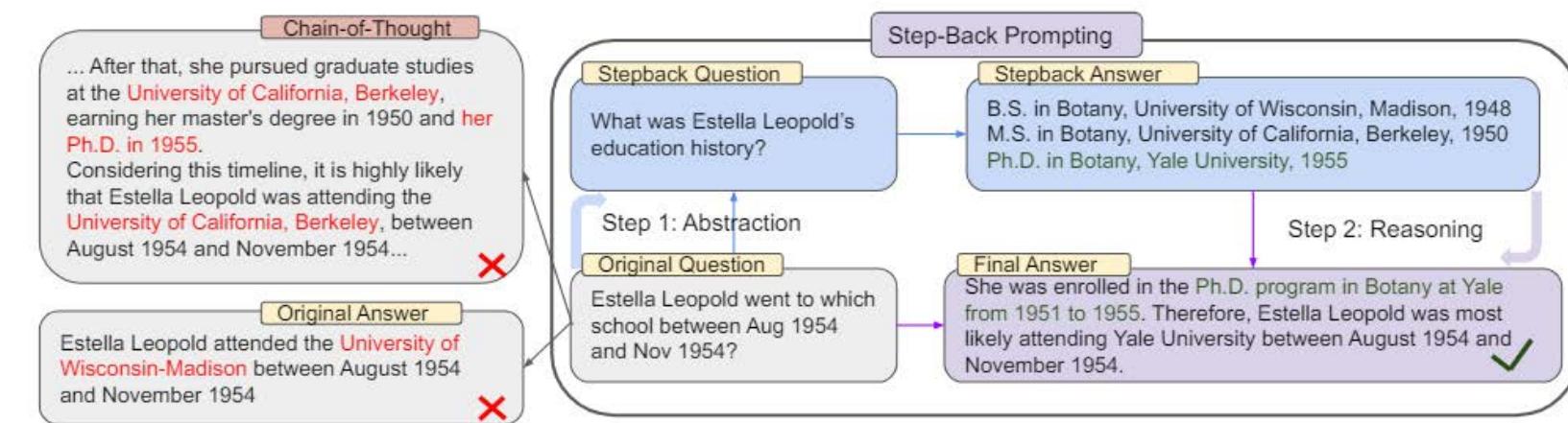
| Method | Success |
|------------------------|------------|
| IO prompt | 7.3% |
| CoT prompt | 4.0% |
| CoT-SC ($k=100$) | 9.0% |
| ToT (ours) ($b=1$) | 45% |
| ToT (ours) ($b=5$) | 74% |
| IO + Refine ($k=10$) | 27% |
| IO (best of 100) | 33% |
| CoT (best of 100) | 49% |

Table 2: Game of 24 Results.



Step-back prompting

☐ Step back and do abstractions to arrive at high-level principles to guide the reasoning process

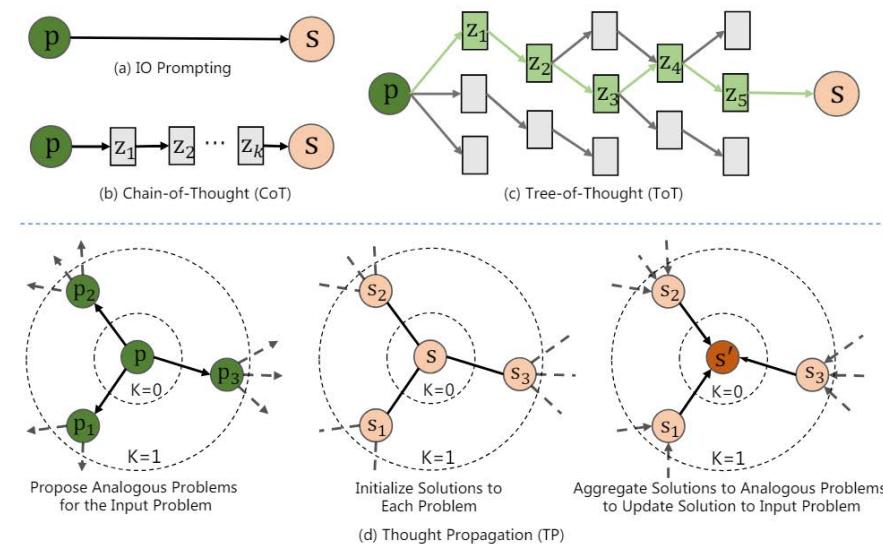
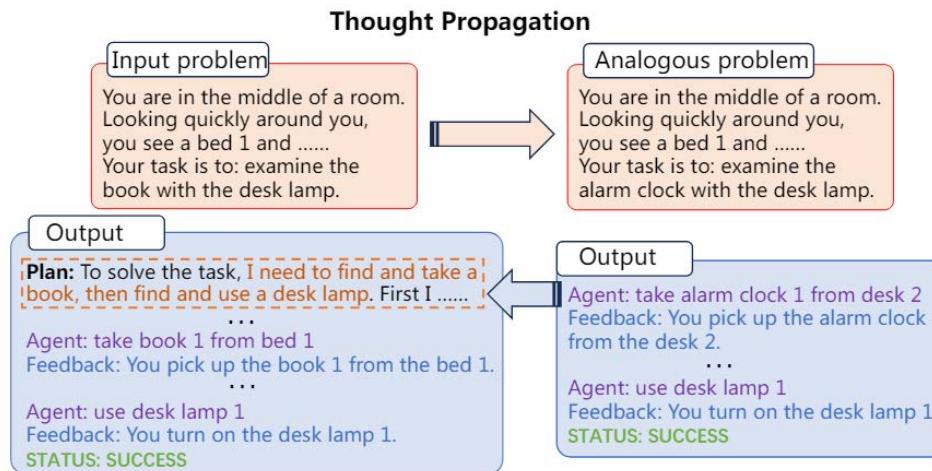


| Method | MMLU Physics | MMLU Chemistry |
|----------------------------|---------------------|---------------------|
| PaLM-2L | 66.4% (0.8%) | 70.9% (0.9%) |
| PaLM-2L 1-shot | 64% (1.6%) | 75.6% (0.4%) |
| PaLM-2L + CoT | 65% (2%) | 75.3% (1.5%) |
| PaLM-2L + CoT 1-shot | 61.5% (1.8%) | 76.6% (1%) |
| PaLM-2L + TDB | 65.7% (0.7%) | 73.8% (1.1%) |
| PaLM-2L + Step-Back (ours) | 73.2% (1.9%) | 81.8% (1.4%) |
| GPT-4 | 69.4% (2.0%) | 80.9% (0.7%) |
| GPT-4 1-shot | 78.4% (2.4%) | 80.5% (1.6%) |
| GPT-4 + CoT | 82.9% (0.5%) | 85.3% (1.0%) |
| GPT-4 + CoT 1-shot | 79.3% (1.0%) | 82.8% (0.5%) |
| GPT-4 + TDB | 74.4% (4.0%) | 81.5% (1.3%) |
| GPT-4 + Step-Back (ours) | 84.5% (1.2%) | 85.6% (1.4%) |

| Method | TimeQA | TQA Easy | TQA Hard | SituatedQA |
|----------------------------------|--------------|--------------|--------------|---------------------|
| PaLM-2L | 41.5% | 42.6% | 40.4% | 54.3% (0.3%) |
| PaLM-2L 1-shot | 40.7% | 41.7% | 39.1% | 51.8% (0.6%) |
| PaLM-2L + CoT | 40.8% | 41.8% | 39.8% | 56.4% (0.2%) |
| PaLM-2L + CoT 1-shot | 38.1% | 39.3% | 36.8% | 54% (0.8%) |
| PaLM-2L + TDB | 40.9% | 42.6% | 39.1% | 54% (0.5%) |
| PaLM-2L + RAG | 57.4% | 67.8% | 46.8% | 59.3% (0.4%) |
| PaLM-2L + Step-Back (ours) | 66% | 70.4% | 61.6% | 57.5% (0.3%) |
| PaLM-2L + Step-Back + RAG (ours) | 68.7% | 75.2% | 62.3% | 61% (0.4%) |
| GPT-4 | 45.6% | 48.9% | 42.6% | 63.2% (0.4%) |

Thought propagation

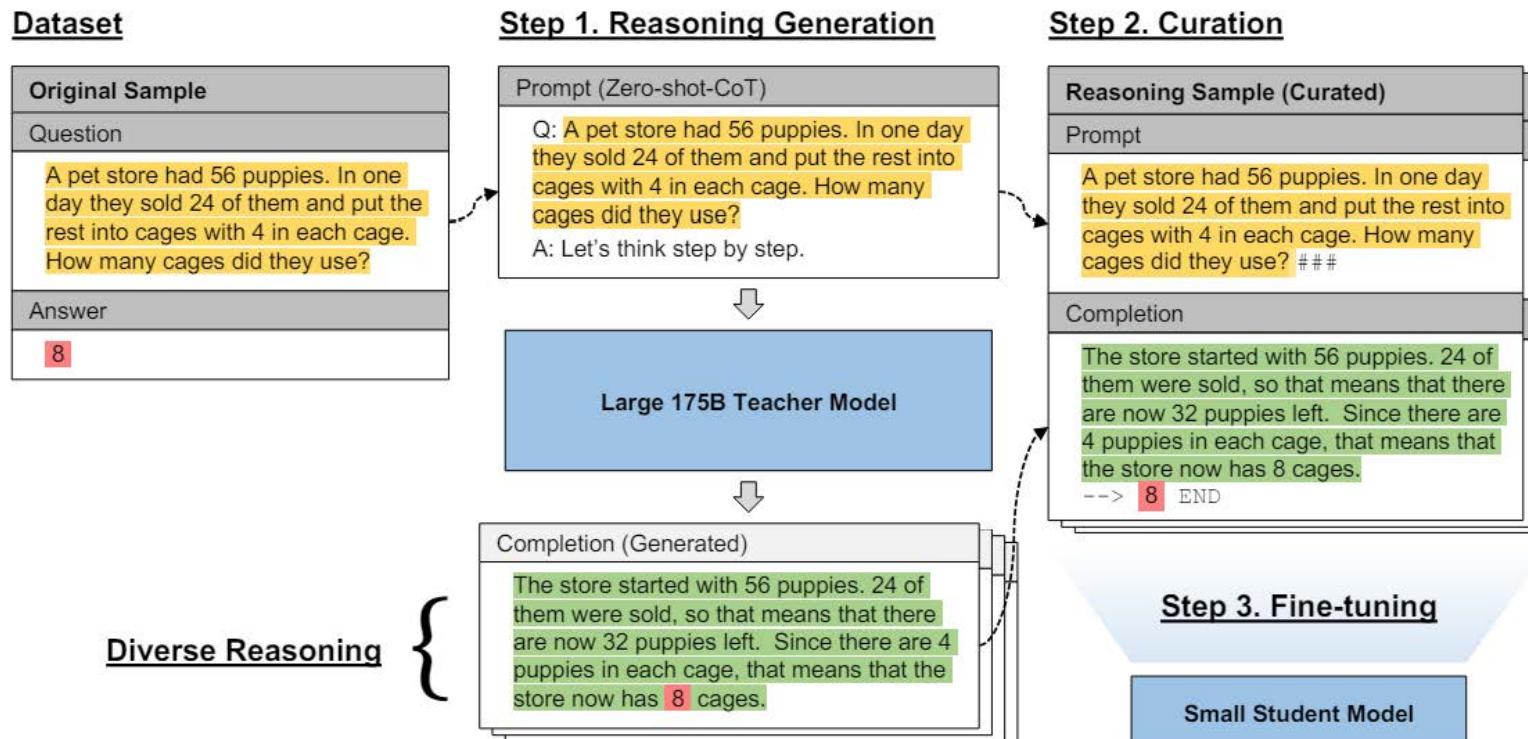
- ❑ Explore the **analogous problems** and leverage their solutions to enhance the complex reasoning ability of LLM
- ❑ Similar to the message propagation in GNN



Yu, Junchi, Ran He, and Rex Ying. "Thought propagation: An analogical approach to complex reasoning with large language models." ICLR 2024.

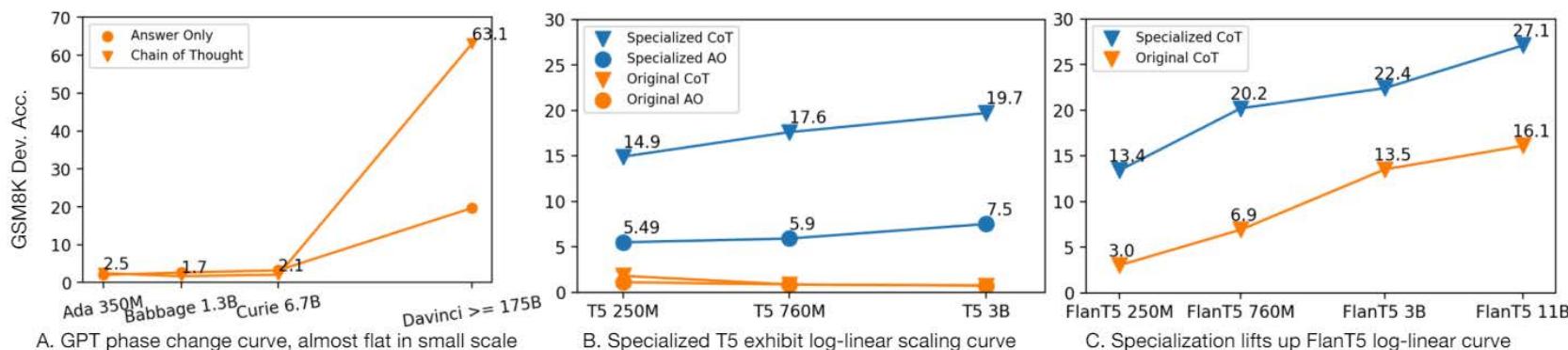
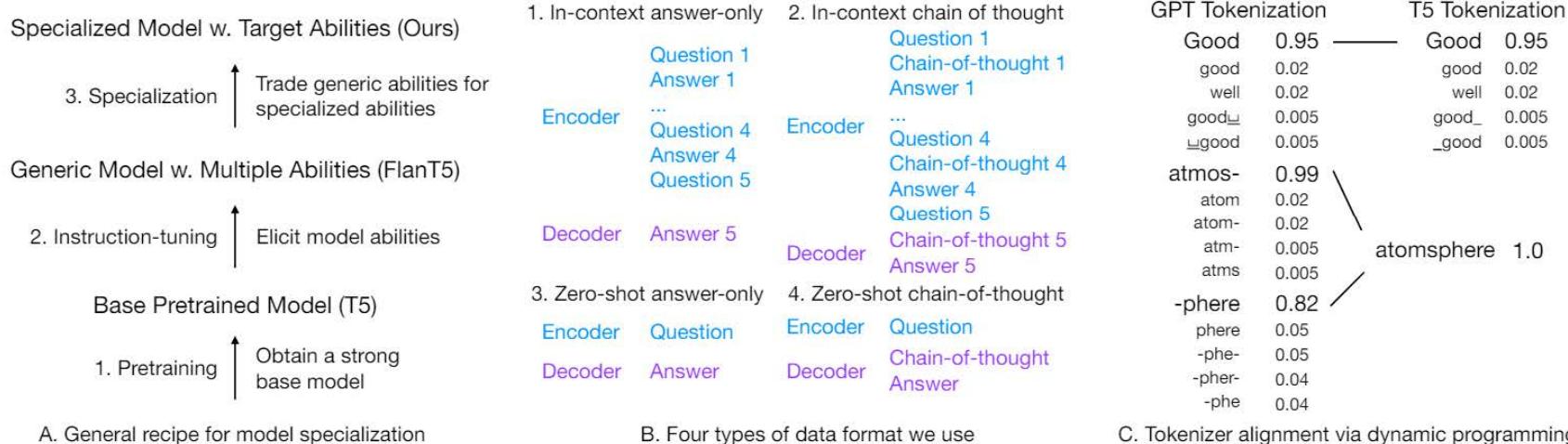
Fine-tune-CoT

- Prompt-based CoT methods are dependent on very large models which are **prohibitive to deploy** at scale
- Generate reasoning samples from very large teacher models to fine-tune smaller models



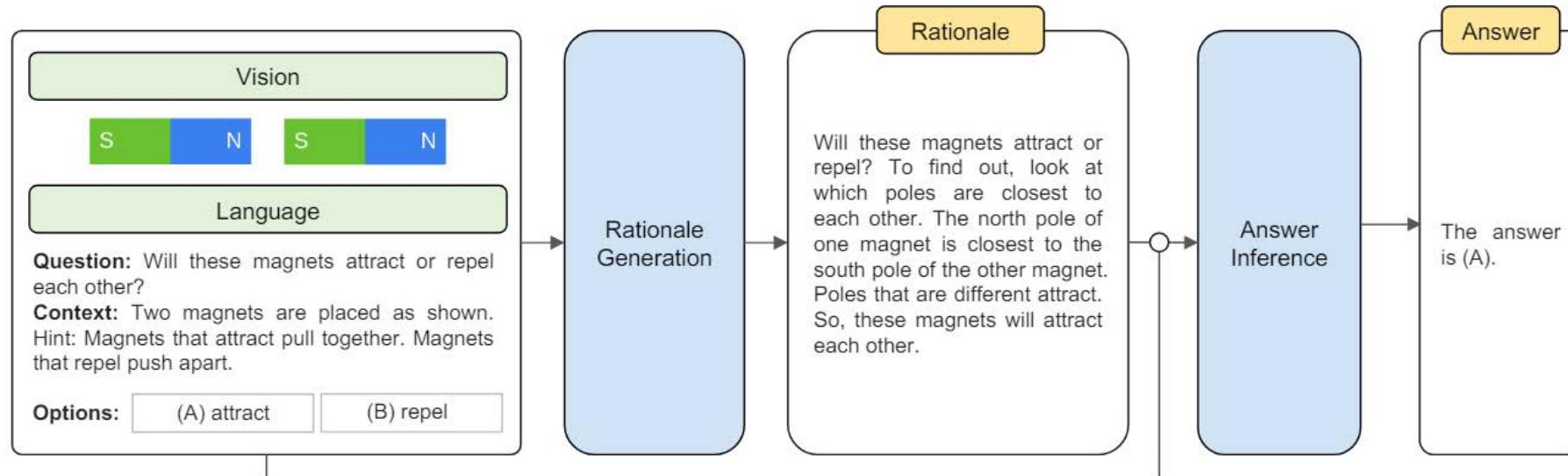
Specialized Reasoning

□ By paying the price of decreased generic ability, we can clearly lift up the specialized multi-step math reasoning ability



Multimodal CoT

- ❑ Incorporate CoT with vision features to generate more informative rationales to facilitate inferring final answers (via fine-tuning)
- ❑ Step1: Fuse questions and vision features and decode to obtain the gold rationale
- ❑ Step2: Add the generated rationale to prompts and decode the answer



Video-of-Thoughts (VoT)

- Implement the CoT technique for achieving human-level video reasoning
- Reason step-by-step from a **low-level pixel perception** to **high-level cognitive interpretation**

Question: What will happen to *the red oil tanker truck*?



Step-1: Task Definition and Target Identification

You are an expert ... answer a question based on the given video. For the question, several candidate answers ...

Input: Given the question: [What is the relationship between the white truck and this neighborhood? A. Transportation B. Buildings C. Clean Services D. Entertainment Facilities]

Output: What are the possible targets of the mainly mentioned or involved?

The involved targets are [the white truck], [the neighborhood]

Step-2: Object Tracking

Input: Provide the tracklet of involved [the neighborhood] and [the white truck] by outputting the corresponding partial expression in the .

Output: The partial in tracking [the neighborhood] and [the white truck] is [Frame 1: {Objects: [{"car-1": [0.0, 13.4, 7.0, 8.1]}, ...}, Triplets: [{"car-1", "on the left", "street"}, ...]}...].

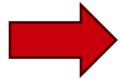
Step-3: Action Analyzing

Input: Combining all possible related commonsense, analyze the motion behavior based on the [the white truck] and [the neighborhood] and the neighbor scenes within . Describing the action and the implication.

Output: [The two men are driving the white truck into a neighborhood, and pour the garbage from the roadside trash cans into the white truck. According to commonsense, the white car is used for collecting rubbish...]

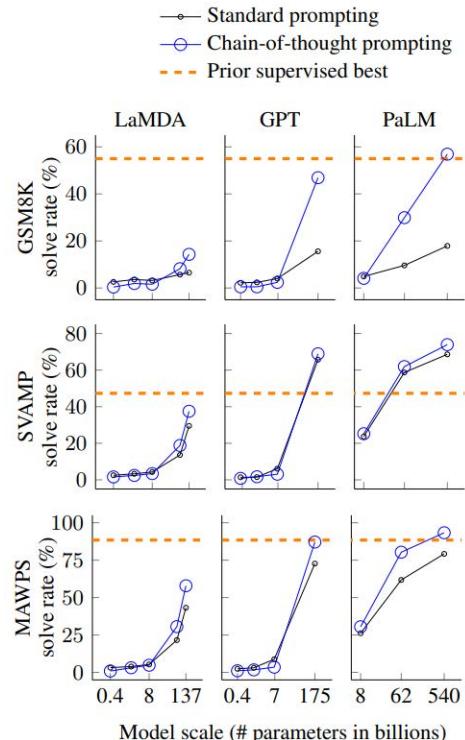
Background

How to improve reasoning ability of LLMs?



Additional costs

□ Model scaling, fine-tuning, in-context learning...



Scaling law of reasoning ability under CoT

| Model | Input /1M tokens | Output /1M tokens |
|------------|------------------|-------------------|
| GPT-3.5 | \$1.5 | \$2 |
| GPT-4 | \$30 | \$60 |
| Yi-34B | \$0.35 | \$0.35 |
| Gemini1pro | \$0.5 | \$1.5 |
| Mistral-7B | \$0.25 | \$0.25 |
| LLaMA-13B | \$0.28 | \$0.28 |
| PaLM2 | 0 | 0 |

Expensive token price of commercial LLMs

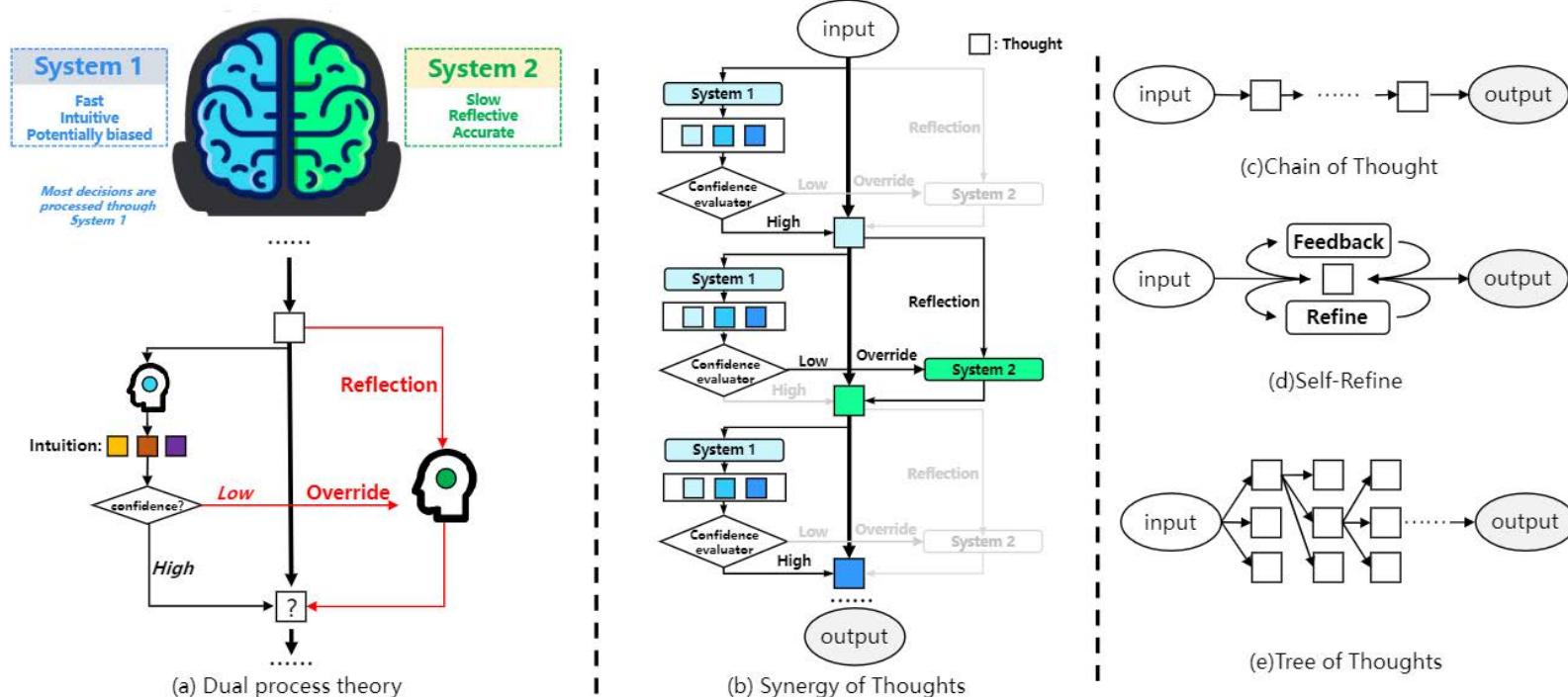
| Game of 24 | Generate/Prompt tokens | Cost per case |
|-------------------|------------------------|---------------|
| IO (best of 100) | 1.8k / 1.0k | \$0.13 |
| CoT (best of 100) | 6.7k / 2.2k | \$0.47 |
| ToT | 5.5k / 1.4k | \$0.74 |

ToT has 100-times token costs than CoT

□ Dual process theory of human cognition

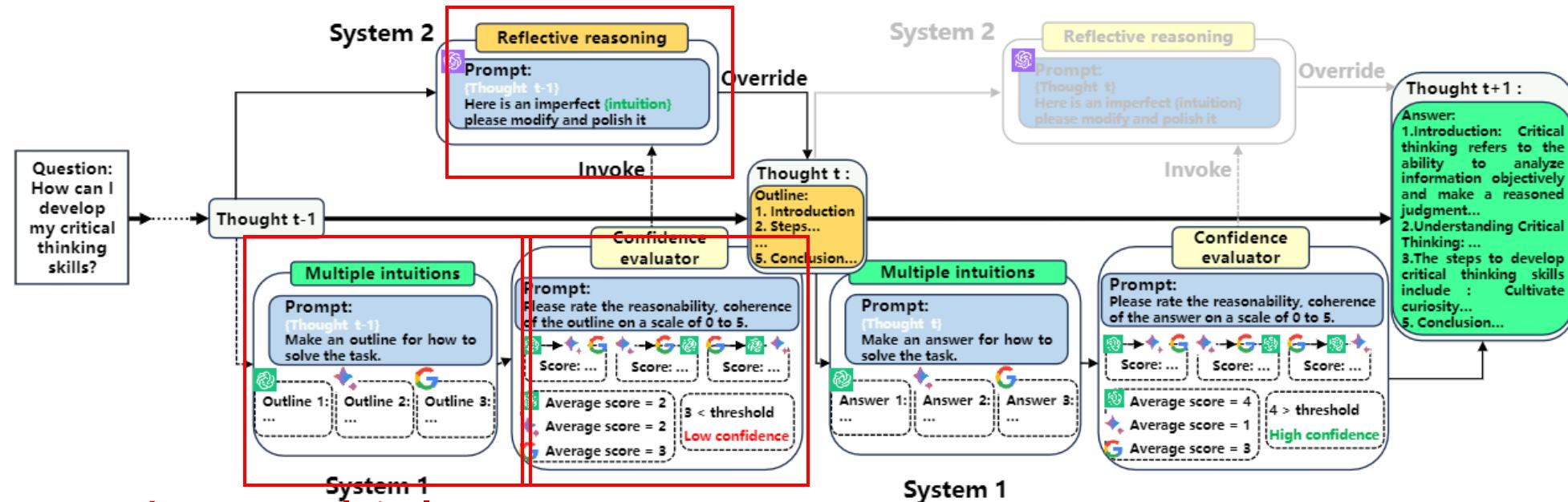
- System 1: fast, intuitive, potentially biased
- System 2: slow, reflective, accurate

Accurate and efficient reasoning



SoT: a model-agnostic and training-free reasoning framework with hybrid LLMs

Invoke System 2 to override intuitions

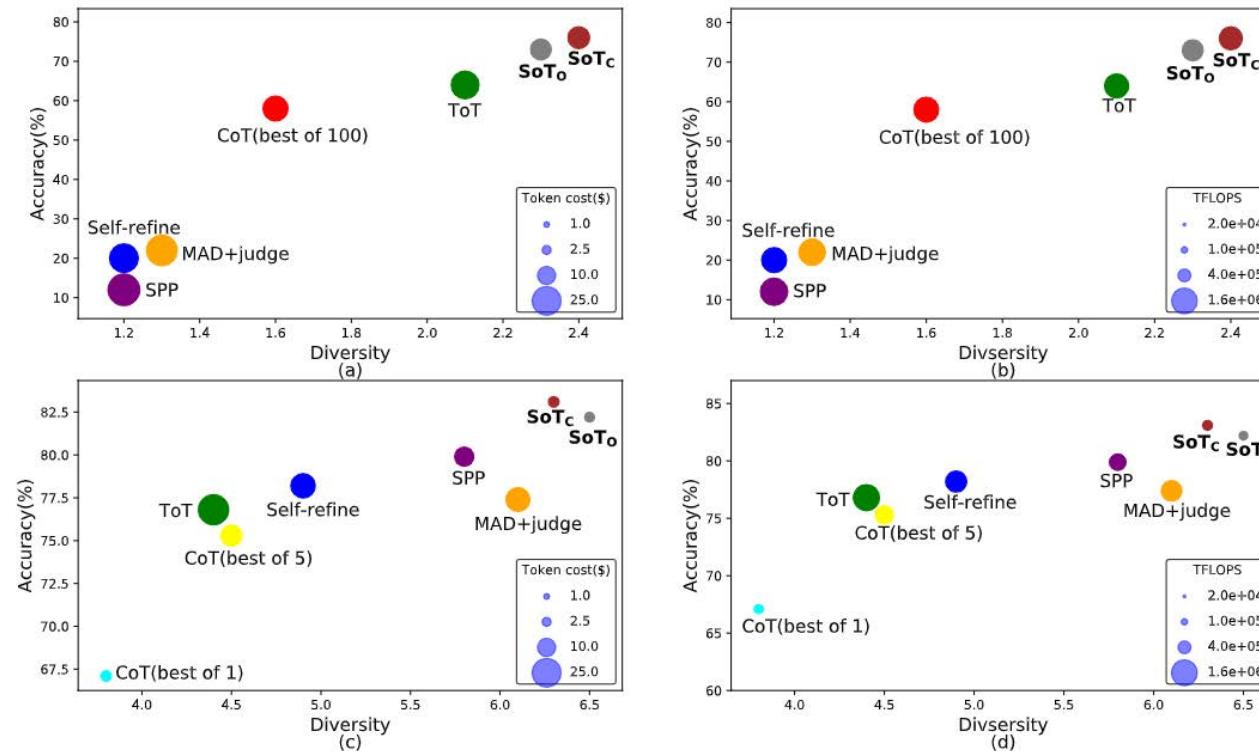


Produce multiple intuitions
(System 1)

Conflict-based confidence evaluation

- SoT achieves **SOTA reasoning accuracy** on six typical tasks, **sufficiently reducing token costs** at the same time
- The average Game of 24 reaches **69.**

Creative Writing



Analysis of cost efficiency about SoT

Input and output per-token cost of System 1: C_{Ii}, C_{Io}

Input and output per-token cost of System 2: C_{Ri}, C_{Ro}

→ Intervention rate of System 2: $r = \frac{7C_{Ii} + 6C_{Io}}{C_{Ri} + C_{Ro}}$

Cost-efficient reasoning $r < 1 - \frac{7C_{Ii} + 6C_{Io}}{C_{Ri} + C_{Ro}}$

with SoT:

| LLM combinations | Game of 24 | Logic Grid Puzzle | GSM8K | Creative Writing | OpenQA | Constrained Generation |
|-----------------------------|------------|-------------------|-------|------------------|--------|------------------------|
| 3 GPT-3.5 | 28% | 49% | 24% | 51% | 55% | 56% |
| 3 PaLM2 | 33% | 55% | 27% | 57% | 57% | 61% |
| 3 Gemini1pro | 30% | 53% | 26% | 50% | 54% | 58% |
| GPT-3.5/PaLM2/Gemini1pro | 26% | 44% | 23% | 42% | 52% | 53% |
| 3 LLaMA-13B | 39% | 68% | 41% | 65% | 67% | 65% |
| 3 Mistral-7B | 36% | 61% | 39% | 60% | 61% | 63% |
| 3 Yi-34B | 36% | 59% | 36% | 61% | 59% | 59% |
| LLaMA-13B/Mistral-7B/Yi-34B | 33% | 54% | 35% | 49% | 54% | 57% |

Empirical intervention rate of SoT implemented with different LLM combinations

- 使用外部工具验证和纠正LLM的输出。
- 将不同的外部工具（如搜索引擎、代码编译器和各种API）转换为文本到文本的函数，并使用上下文学习使LLM能够利用这些工具。
- “验证然后纠正”的过程不断迭代，直到满足停止标准，满足验证条件、接收环境反馈、达到最大迭代次数等。

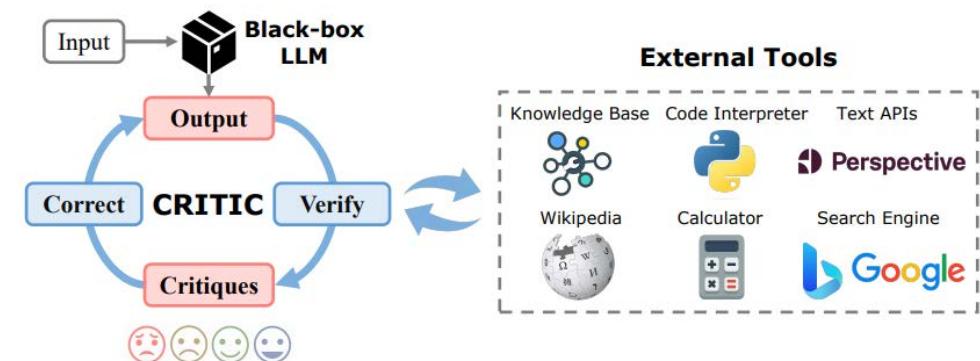


Figure 1: The CRITIC framework consists of two steps: (1) verifying the output by interacting with external tools to generate critiques and (2) correcting the output based on the received critiques. We can iterate such verify-then-correct process to enable continuous improvements.

- 目的：

- 利用大型语言模型（LLMs）作为优化器来解决各种优化问题。
通过逐步生成新的提示优化任务，并指导LLMs迭代生成新的解决方案。

- 方法：

- LLM根据优化问题描述和先前评估的解决方案生成优化任务的候选解决方案。对新的解决方案进行评估，并将其添加到元提示中，以进行后续的优化过程。
- 元提示设计：包含优化问题描述、优化轨迹（先前解决方案及其分数）。
- 解决方案生成：LLM根据元提示生成新解决方案，同时优化稳定性和权衡探索与利用。

- 结论

- 在GSM8K和BBH等推理基准测试中，OPRO优化的提示在zero-shot的情况下，与人类设计的提示相比，在GSM8K上性能提升

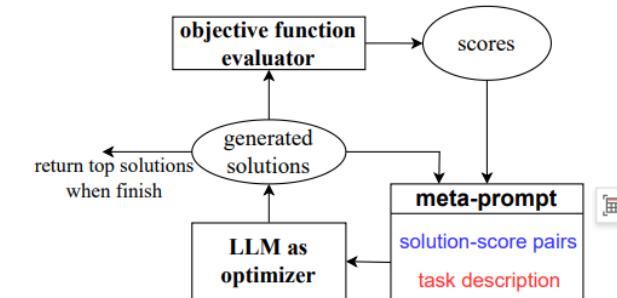
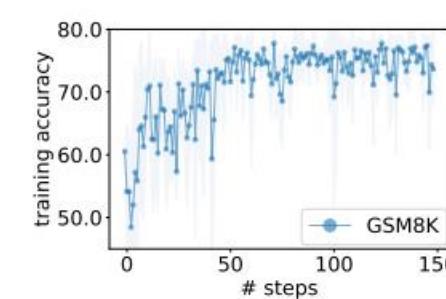
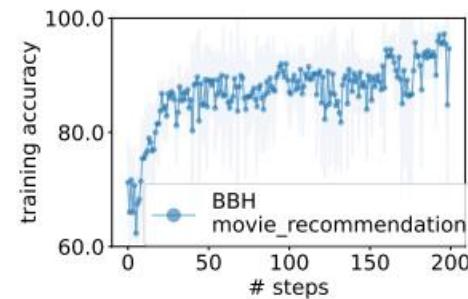


Figure 2: An overview of the OPRO framework. Given the meta-prompt as the input, the LLM generates new solutions to the objective function, then the new solutions and their scores are added into the meta-prompt for the next optimization step. The meta-prompt contains the solution-score pairs obtained throughout optimization, a natural language description of the task, and (in prompt optimization) a few task exemplars. Figure 3 shows a sample meta-prompt for prompt optimization.

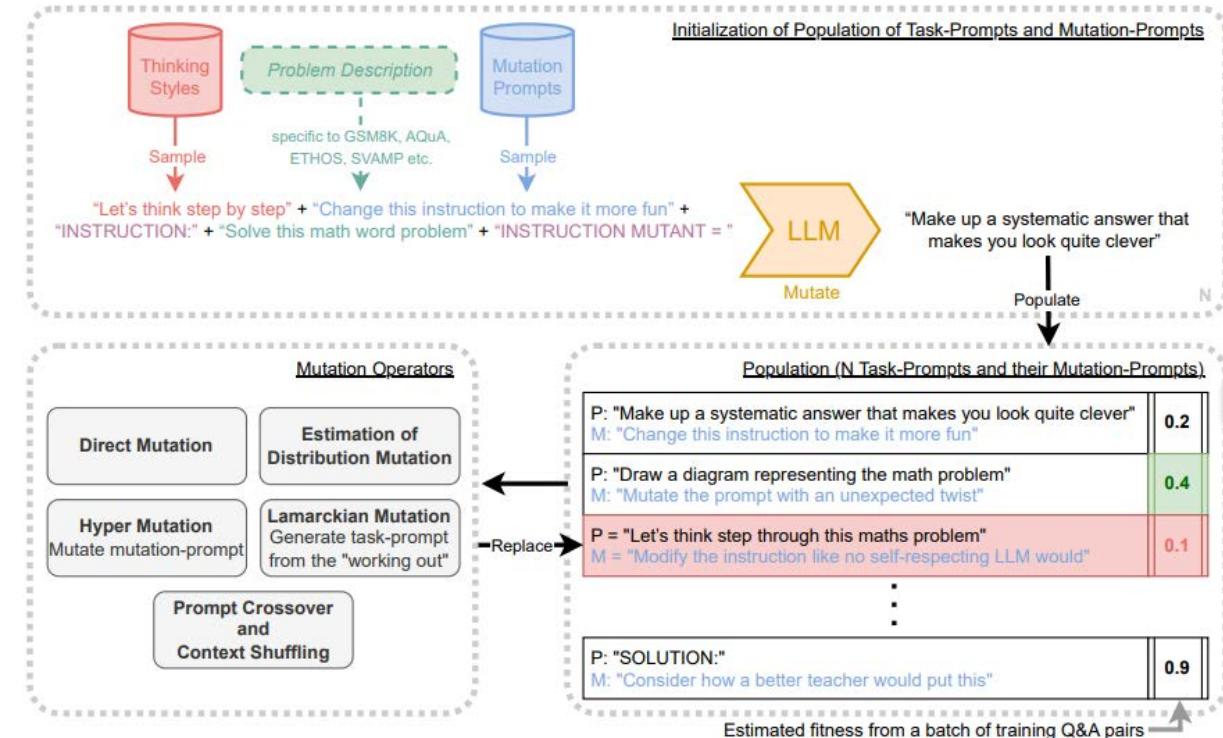


(a) GSM8K



(b) BBH movie_recommendation

- 目标:
 - 手动设计的提示策略往往不是最优的提示词，设计一个能够自我参照、自我改进的系统，可以自动为特定领域的任务制定有效的提示。
- 方法:
 - 对一组任务提示进行变异（通过LLM生成），评估它们在训练集上的表现（通过监督训练的模型），并在重复这一过程以进化任务提示。
 - 不仅改进任务提示，还改进用于生成这些提示的思考风格。



| | Method | LLM | MultiArith* | SingleEq* | AddSub* | SVAMP* | SQA | CSQA | AQuA-RAT | GSM8K |
|-----------|------------|------------------|--------------|-------------|---------------|-------------|-------------|-------------|-------------|-------------|
| Zero-shot | CoT | text-davinci-003 | (83.8) | (88.1) | (85.3) | (69.9) | (63.8) | (65.2) | (38.9) | (56.4) |
| | PoT | text-davinci-003 | (92.2) | (91.7) | (85.1) | (70.8) | — | — | (43.9) | (57.0) |
| | PS | text-davinci-003 | (87.2) | (89.2) | (88.1) | (72.0) | — | — | (42.5) | (58.2) |
| | PS+ | text-davinci-003 | (91.8) | (94.7) | (92.2) | (75.7) | (65.4) | (71.9) | (46.0) | (59.3) |
| | PS | PaLM 2-L | 97.7 | 90.6 | 72.4 | 83.8 | 50.0 | 77.9 | 40.2 | 59.0 |
| | PS+ | PaLM 2-L | 92.5 | 94.7 | 74.4 | 86.3 | 50.1 | 73.3 | 39.4 | 60.5 |
| | APE | PaLM 2-L | 95.8 | 82.2 | 72.2 | 73.0 | 38.4 | 67.3 | 45.7 | 77.9 |
| | OPRO | PaLM 2-L | — | — | — | — | — | — | — | 80.2 |
| | PB (ours) | PaLM 2-L | 99.7 | 96.4 | 87.8 | 90.2 | 71.8 | 85.4 | 62.2 | 83.9 |
| Few-shot | Manual-CoT | text-davinci-003 | (93.6) | (93.5) | (91.6) | (80.3) | (71.2) | (78.3) | (48.4) | (58.4) |
| | Auto-CoT | text-davinci-003 | (95.5) | (92.1) | (90.8) | (78.1) | — | — | (41.7) | (57.1) |
| | PB (ours) | PaLM 2-L | 100.0 | 98.9 | 87.1 | 93.7 | 80.2 | 85.9 | 64.6 | 83.5 |

AFLOW: AUTOMATING AGENTIC WORKFLOW GENERATION arxiv,2024

• 目的:

- 通过自动化的方式，减少对人类干预的依赖，提高LLMs在多样化任务中的性能和应用范围。
- 自动化生成和优化LLMs的智能体工作流。

• 方法:

- AFLOW将工作流优化问题重新定义为代码表示的工作流的搜索问题，其中LLM调用节点通过边连接。
- 引入 operator的概念。通过将N和E组合成统一的接口封装了常见的智能体工作模块（例如集成、审查、修订）。
- AFLOW使用蒙特卡洛树搜索（Monte Carlo Tree Search, MCTS）来系统地探索和发现最优的智能体工作流。通过修改代码、树状结构的经验和执行反馈来迭代优化工作流。

- 结果：在六个基准数据集（HumanEval、MBPP、MATH、GSM8K、HotPotQA和DROP）上的实证评估表明AFLOW的有效性，其性能平均高于ADAS, Self-refine等sota方法的5. 7%。

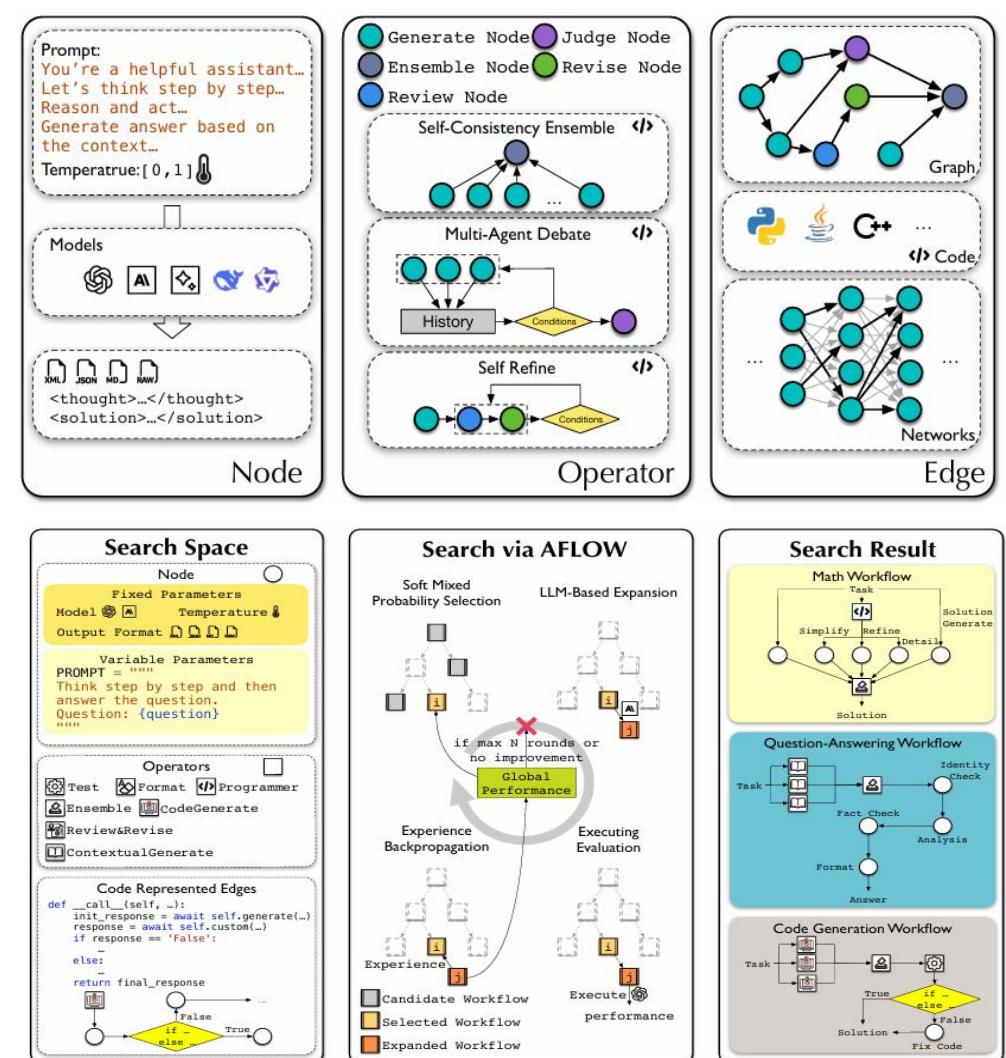


Figure 3: **Overall AFLOW framework:** By setting a search space composed of nodes with only prompt parameters flexible, a given operator set, and a code representing edge, AFLOW performs an MCTS-based search within this space. Through a variant of MCTS designed for workflow optimization, AFLOW iteratively executes a cycle of Soft Mixed Probability Selection, LLM-Based Expansion, Execution Evaluation, and Experience Backpropagation until reaching the maximum number of iterations or meeting convergence criteria.

- 目的:

- 解决使用大型语言模型（LLMs）进行长篇故事创作时面临的挑战，如保持叙述的一致性、准确性以及展示独特性或幽默感。

- 方法:

- 将写作任务分解为多个子任务，并由专门的智能体协作完成。
- 中央协调器是“AGENTS’ ROOM”框架中的控制中心，负责决定智能体的调用顺序和是否需要调用某个智能体。根据当前草稿本的状态和任务需求来确定下一步应该执行的操作。
- 自动化评估提供了即时反馈，有助于进一步训练和调整智能体，以生成更符合预期的故事。

- 结论

- 人类评估者及LLM评估器对故事的多个维度（如情节、创

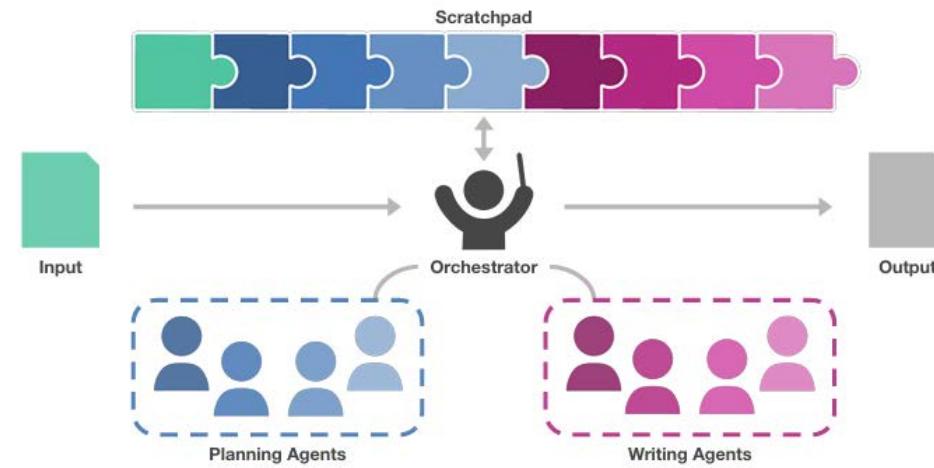


Figure 1: AGENTS’ ROOM, a multi-agent framework for collaborative writing. A central orchestrator calls the individual agents and consolidates their contributions into the scratchpad. We color-code each piece of the scratchpad with the contributing agent’s color.

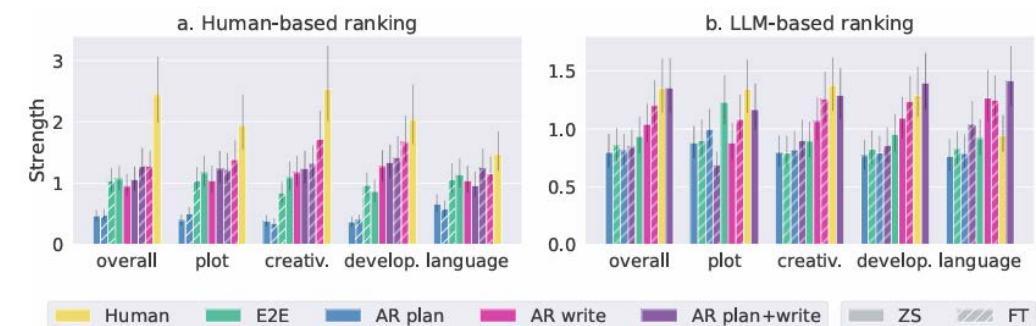


Figure 3: Overall system ranking across dimensions of plot, creativity, development, and language, according to human ratings (a) and a LLM-based evaluator (b).

Self-improvement at Multi-agent Level:

GPTSwarm: Language Agents as Optimizable Graphs ICML,2024

• 目的：

- 已经提出了各种人为设计的提示工程技术来改进基于大型语言模型（LLM）的问题求解器，从而产生了许多不同的代码库。
 - LLM智能体系统的高效开发、集成和自动改进。

• 方法：

- 节点优化：优化节点级LLM提示词。迭代地更新节点的历史记录和提示来改进节点的性能。
 - 边优化：通过REINFORCE算法改变图连通性并应用梯度上升优化。

• 结论

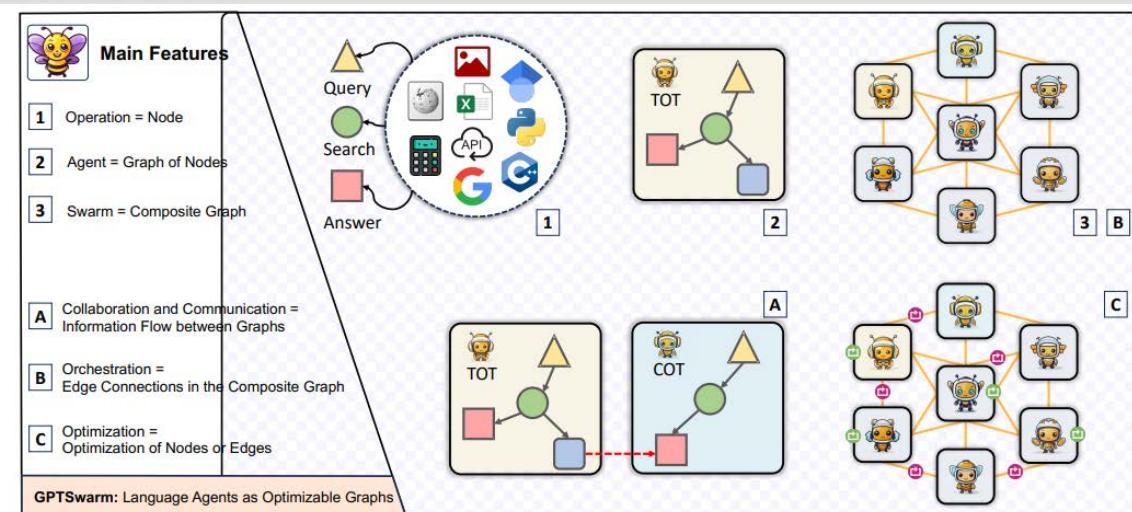
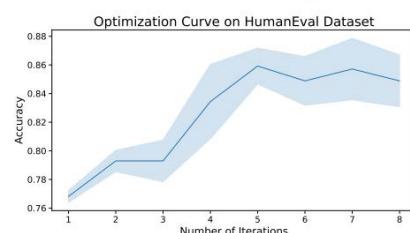
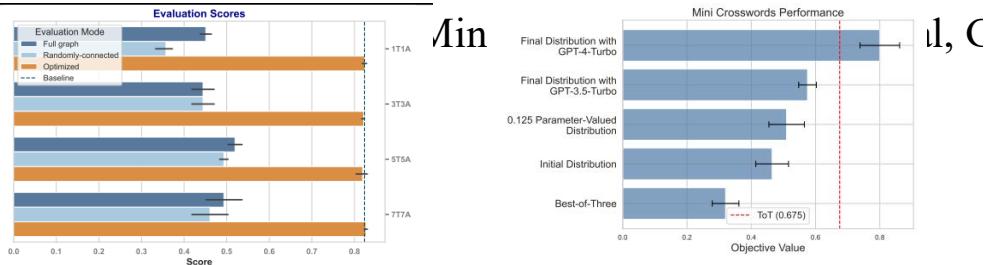


Figure 1. GPTSwarm is a framework that represents agents as graphs. In this framework, each node represents an operation (e.g., LLM inference or tool use). An agent is a graph composed of these nodes. An edge between two agent graphs characterizes a communication channel; each agent collaborates with others through different channels. When connected, multiple agents form a composite graph with a certain orchestration topology. This graph representation lends itself to optimization of nodes and edges via prompting and evolutionary or reinforcement learning techniques.

Algorithm 2 Edge Optimization with REINFORCE

Require: A parameterized probabilistic distribution over computation graphs D_θ , an unbiased utility estimator $\hat{u}_\tau(\cdot)$, and a learning rate α .
Initialize: $\theta \in \mathbb{R}^d$

while terminate condition not met **do**

While terminate condition not met do
 Sample C_i , σ_i , D_i for $i = 1, 2, \dots, M$

Update $\theta \leftarrow \theta + \frac{\alpha}{M} \sum_{i=1}^M \hat{y}_i(G_i) \nabla_\theta \log(p_\theta(G_i))$

Update end while

END WHILE

hEval Dataset

10

Algorithm 3 Node Optimization

Require: A parameterized graph $G^P = (N, E, F^P, o)$, natural language function descriptions $D = \{d_n\}_{n \in N}$, and a distribution of inputs D_X .

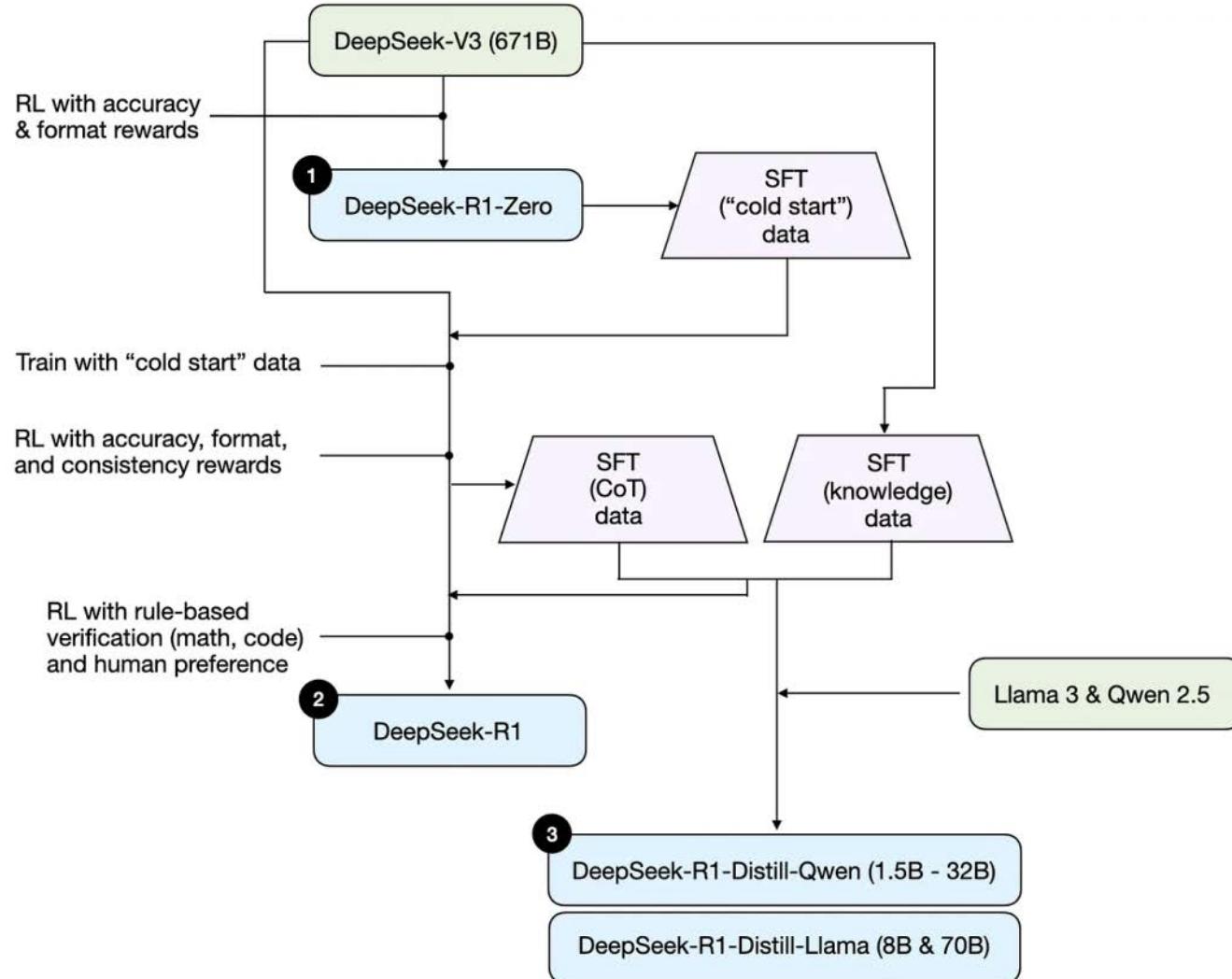
Initialize p_n for all $n \in N$.

Initialize $h_n \leftarrow \emptyset$ for all $n \in N$.

while terminate condition not met **do**
 Sample input $x \sim D_X$.
 $y \leftarrow G^P(x)$ following Algorithm 1.
 $h_n \leftarrow h_n \cup \{(z_n, x), f_n^{p_n}(z_n, x)\}$ for all $n \in N$.
 $p_n \leftarrow I(h_n, p_n, d_n)$, for all $n \in N$.
end while.

通向DeepSeek-R1之路

通向DeepSeek-R1之路



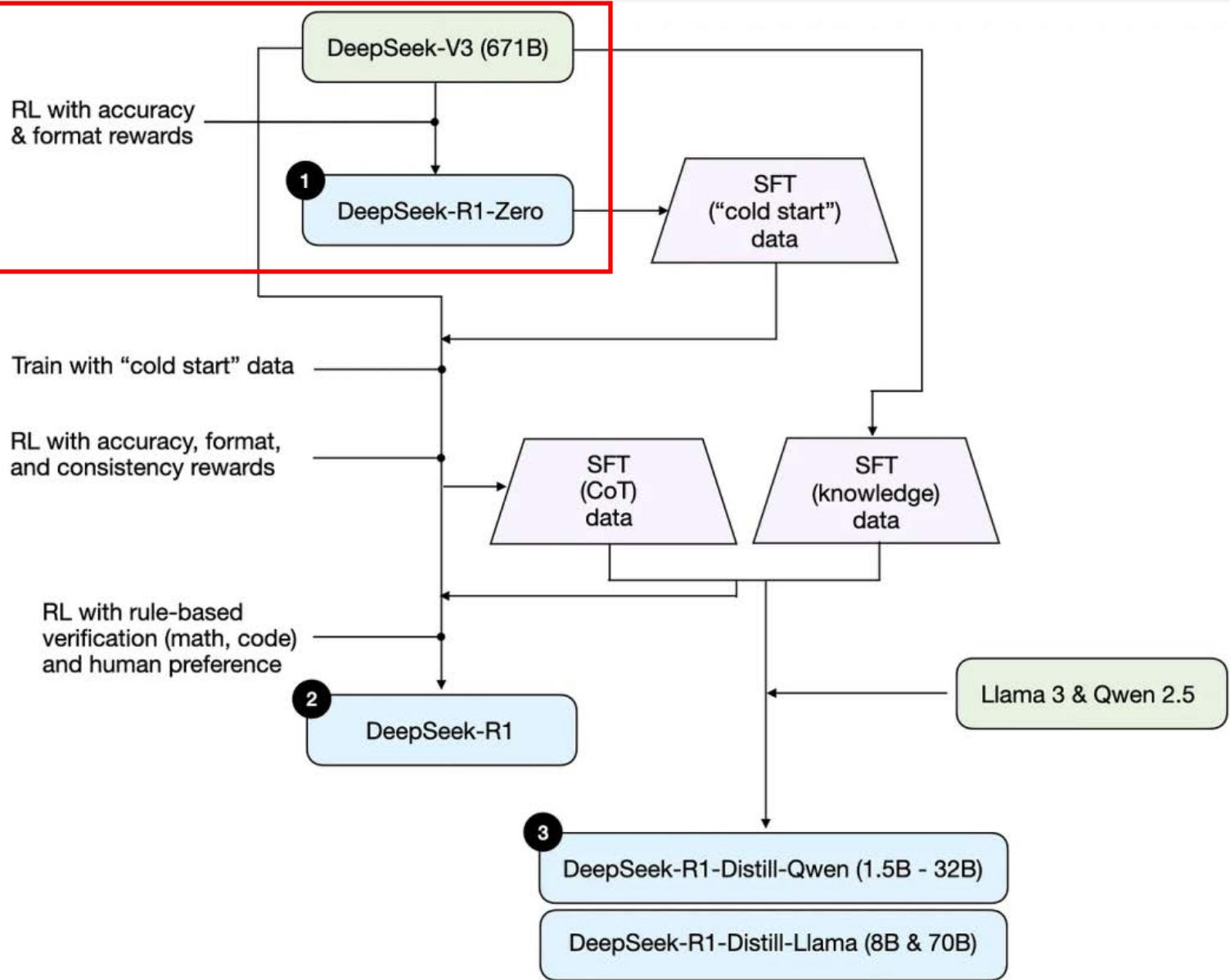
R1-Zero

- 基于V3进行强化学习训练
- 意义：探索LLM在无标签数据下的推理潜能

R1

- Step1. 冷启动 (Cold-start)
- Step2. 第一次RL强化学习训练
- Step3. SFT监督微调
- Step4. 第二次RL强化学习训练
- 意义：通用任务对话模型

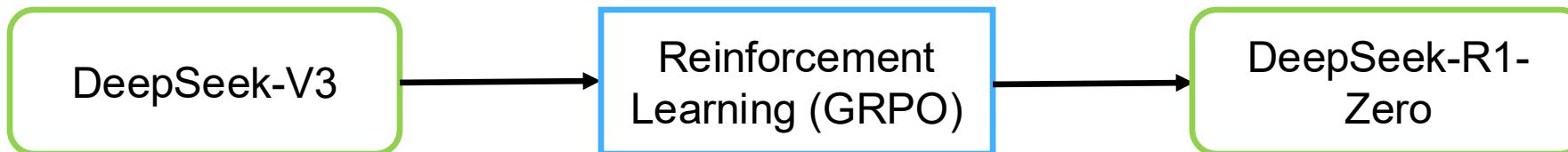
DeepSeek-R1-Zero



- **RL路径验证**

训练方式

- 以V3为base model应用GRPO进行强化学习，其中：
 - State**: 完整的数学问题描述和已生成的token序列
 - Action**: 模型每一步生成的Token
 - Reward**: 预定义的Reward，要求模型回答正确且符合输出格式要求



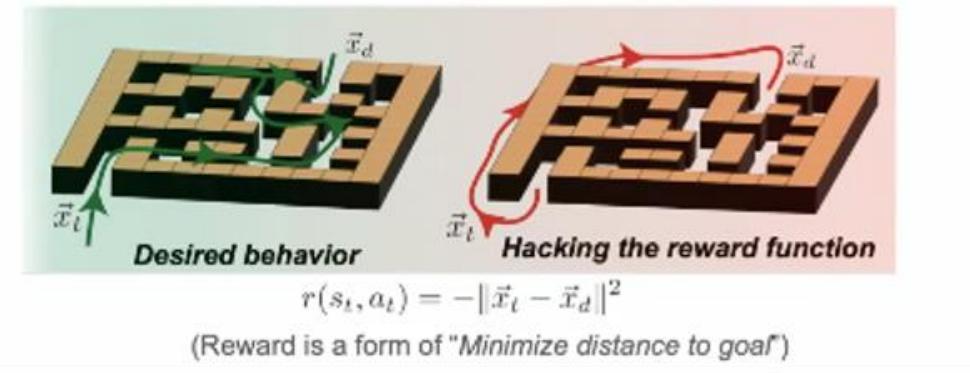
基于规则的预定义Reward

- **Accuracy rewards:** 评估回答是否正确
 - Math problems: 存在确定性答案，直接评估准确性
 - LeetCode problems: 编译代码，在预定义好的评测例上运行
- **Format rewards:** 评估格式是否符合要求
 - 要求模型把它的思考过程放在 '<think>' 和 '</think>' 标签之间

Why not neural reward model?

- 大规模强化学习中的Reward hacking问题

- 智能体“钻了规则的空子”，找到一种偏离设计初衷的方法最大化奖励
- 例如：“每捡起1个垃圾得1分” \rightarrow 捡起垃圾后丢下再捡一次，不断重复



- 资源需求与复杂性

- 训练neural reward model需要额外的训练数据及训练时间
- 使整个pipeline更为复杂

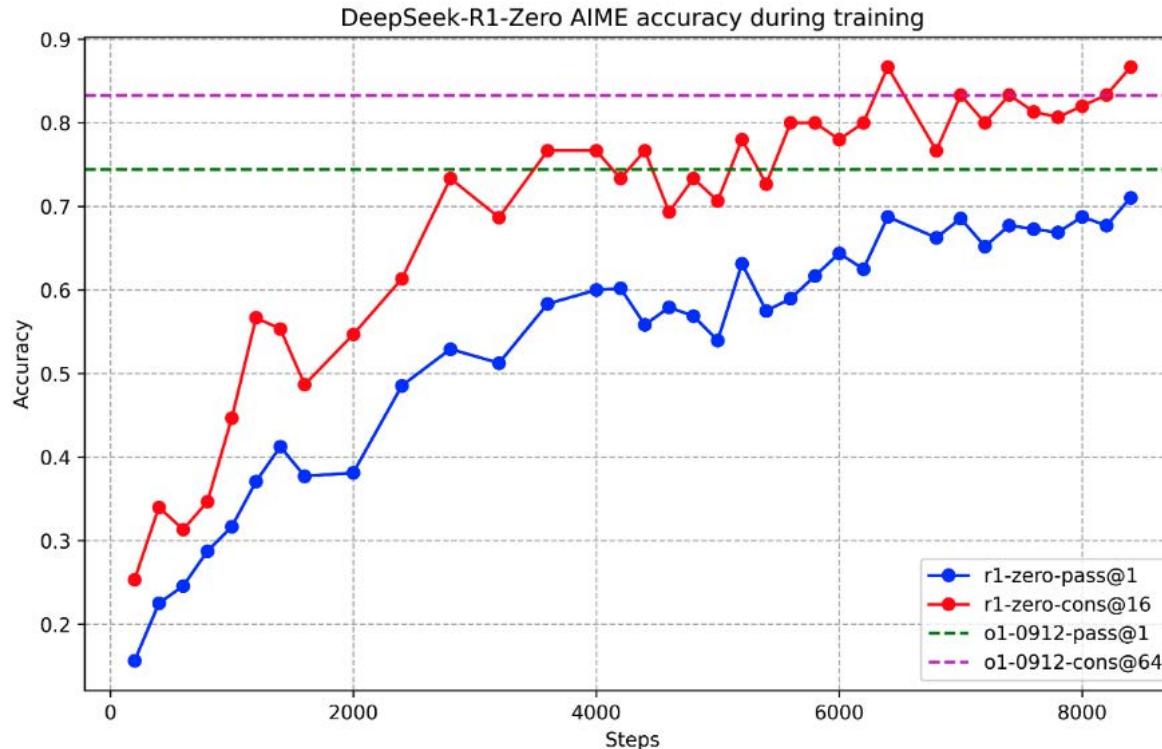
训练模板

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:

- 要求模型首先输出(`<think>`推理过程`</think>`), 之后输出(`<answer>`回答`</answer>`)
 - 不强制模型在此过程中进行反思推理
 - 不引入特定的问题解决策略
- 
- 观察模型在RL训练过程中的自然进展

DeepSeek-R1-Zero

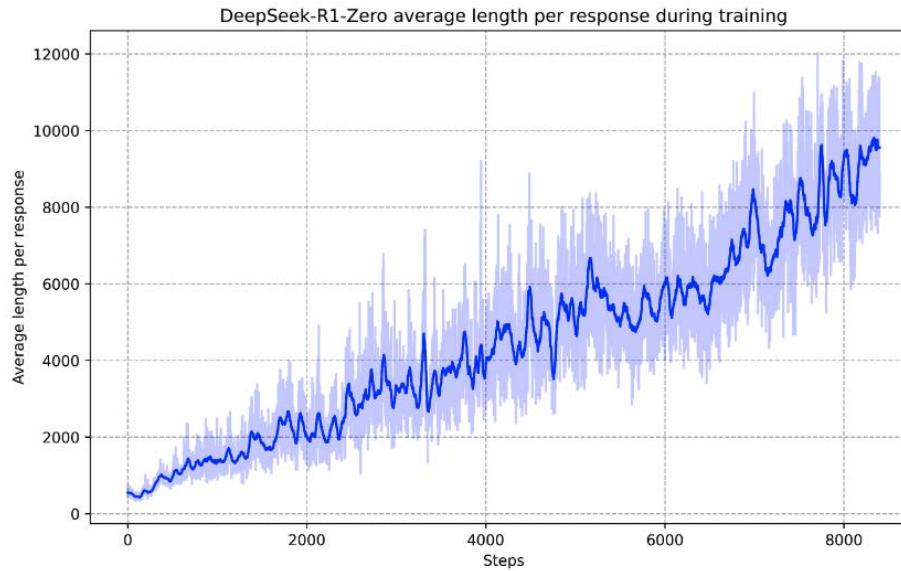
性能表现



- R1-Zero随RL训练过程，在AIME24数据集上的准确率稳步提升，接近OpenAI-o1水平
- 通过使用majority voting可以获得更高的推理能力
- 在不同领域、不同问题的评估数据集中均取得了与OpenAI-o1相当的性能水平

| Model | AIME 2024 | | MATH-500 | GPQA Diamond | LiveCode Bench | CodeForces |
|------------------|-----------|---------|----------|-----------------|-------------------|------------|
| | pass@1 | cons@64 | pass@1 | pass@1 | pass@1 | rating |
| OpenAI-o1-mini | 63.6 | 80.0 | 90.0 | 60.0 | 53.8 | 1820 |
| OpenAI-o1-0912 | 74.4 | 83.3 | 94.8 | 77.3 | 63.4 | 1843 |
| DeepSeek-R1-Zero | 71.0 | 86.7 | 95.9 | 73.3 | 50.0 | 1444 |

“自我进化”过程与“Aha Moment”涌现



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: ...

...

- 模型思考时间和回复长度随着RL训练过程增加，而非监督学习训练等外部人为干预结果

更多Inference-time computation

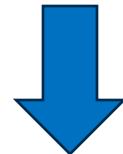


复杂能力如反思、多路径问题探索的标示

- “Aha moment”**：模型学会为反思当前求解问题的方法分配更多的思考时间
- 启示：**不需要教模型如何解决问题，而是给它提供正确的激励（reward），可以自动发展出解决问题的能力

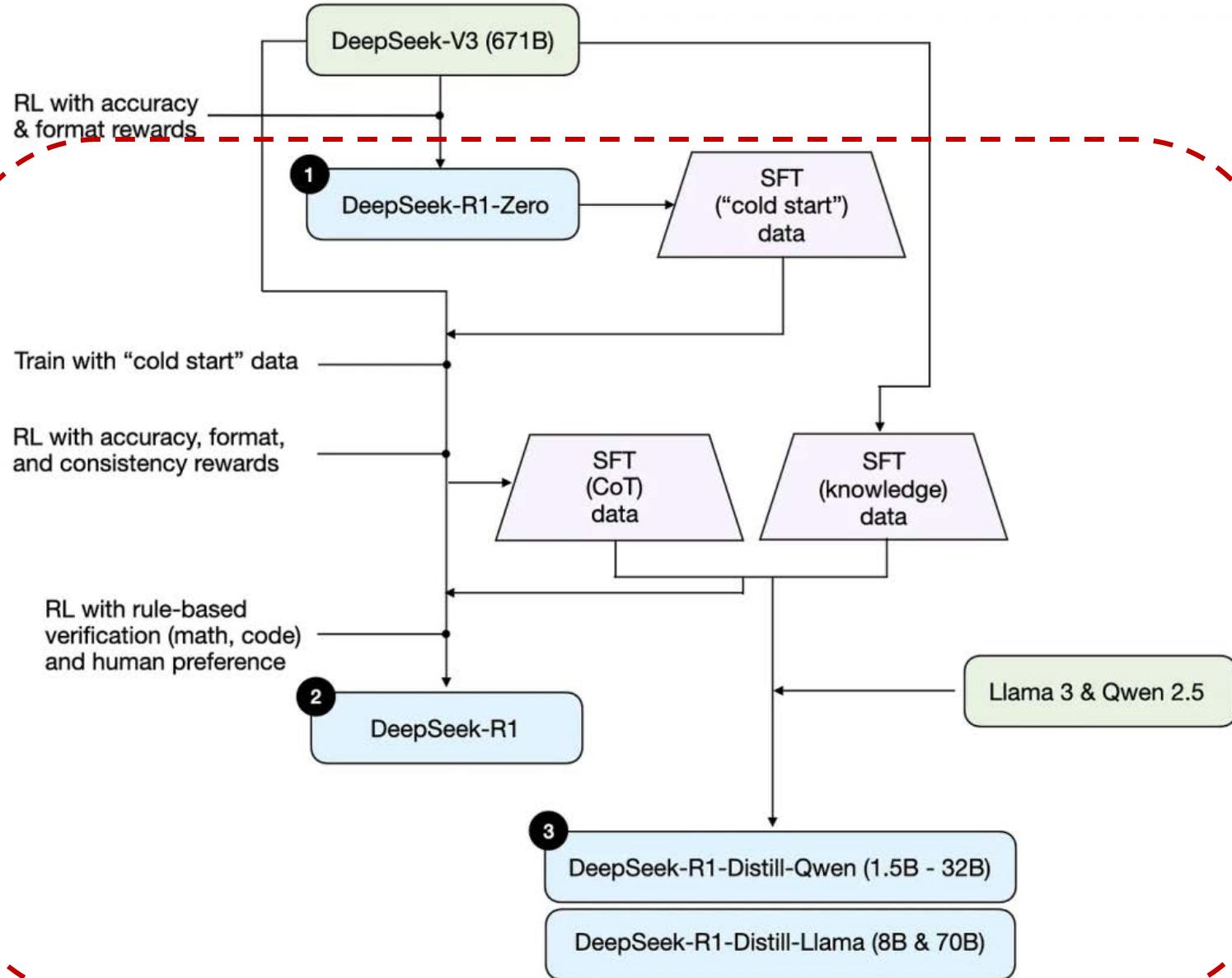
R1-Zero的缺陷

- 输出结果可读性差
- 语言混合
 - 例如回答结果中英文夹杂等



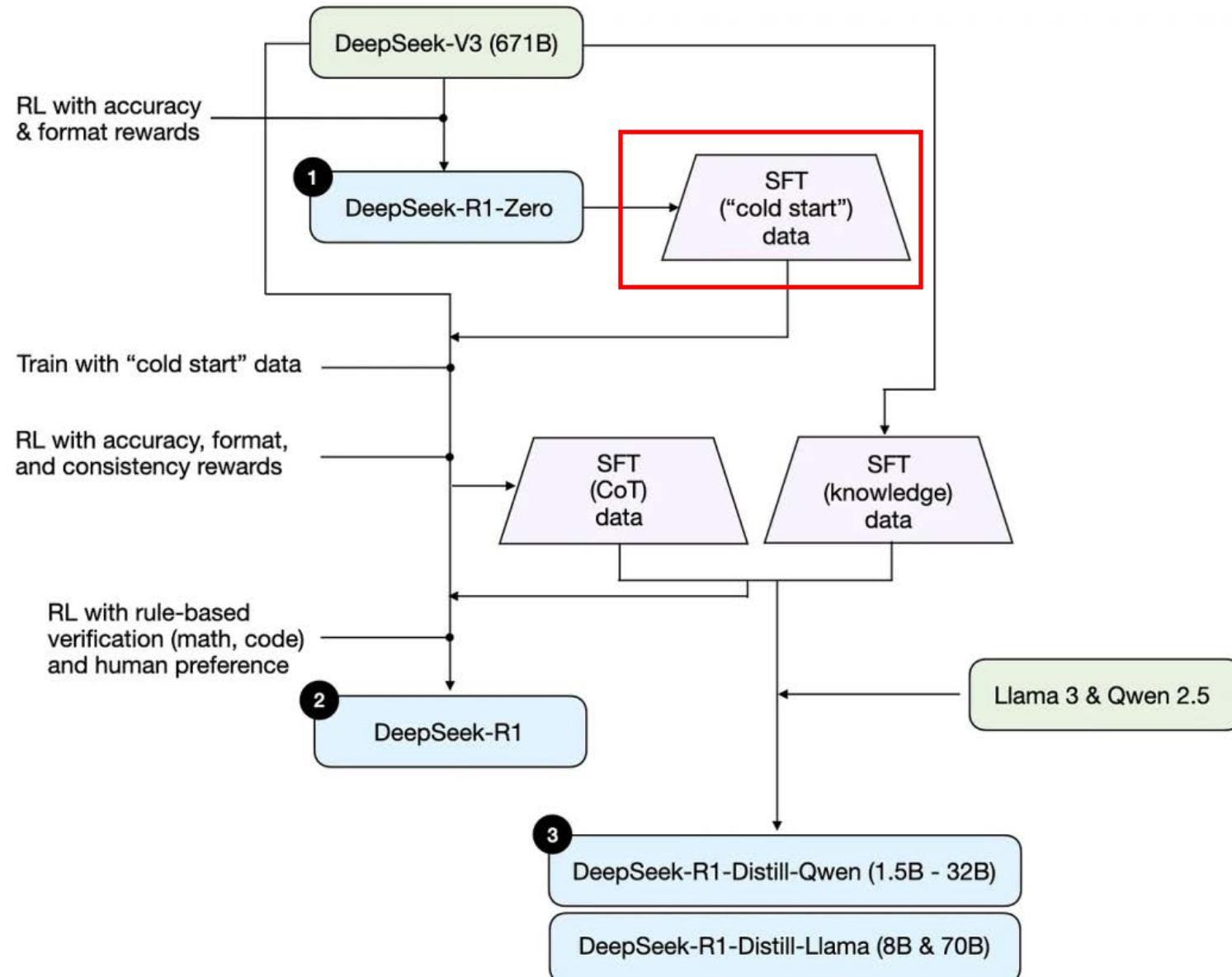
探索引入RL条件下的推理能力潜力，未将回答结果对齐人类偏好

DeepSeek-R1



两次SFT、两次RL

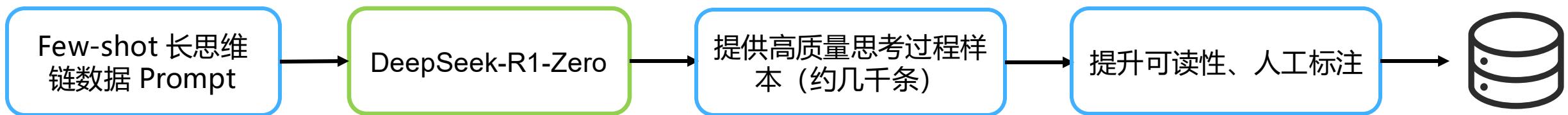
DeepSeek-R1



第一步：Cold start Supervised Fine-Tuning

目的：初步提升模型推理能力，使后续RL训练过程更快收敛

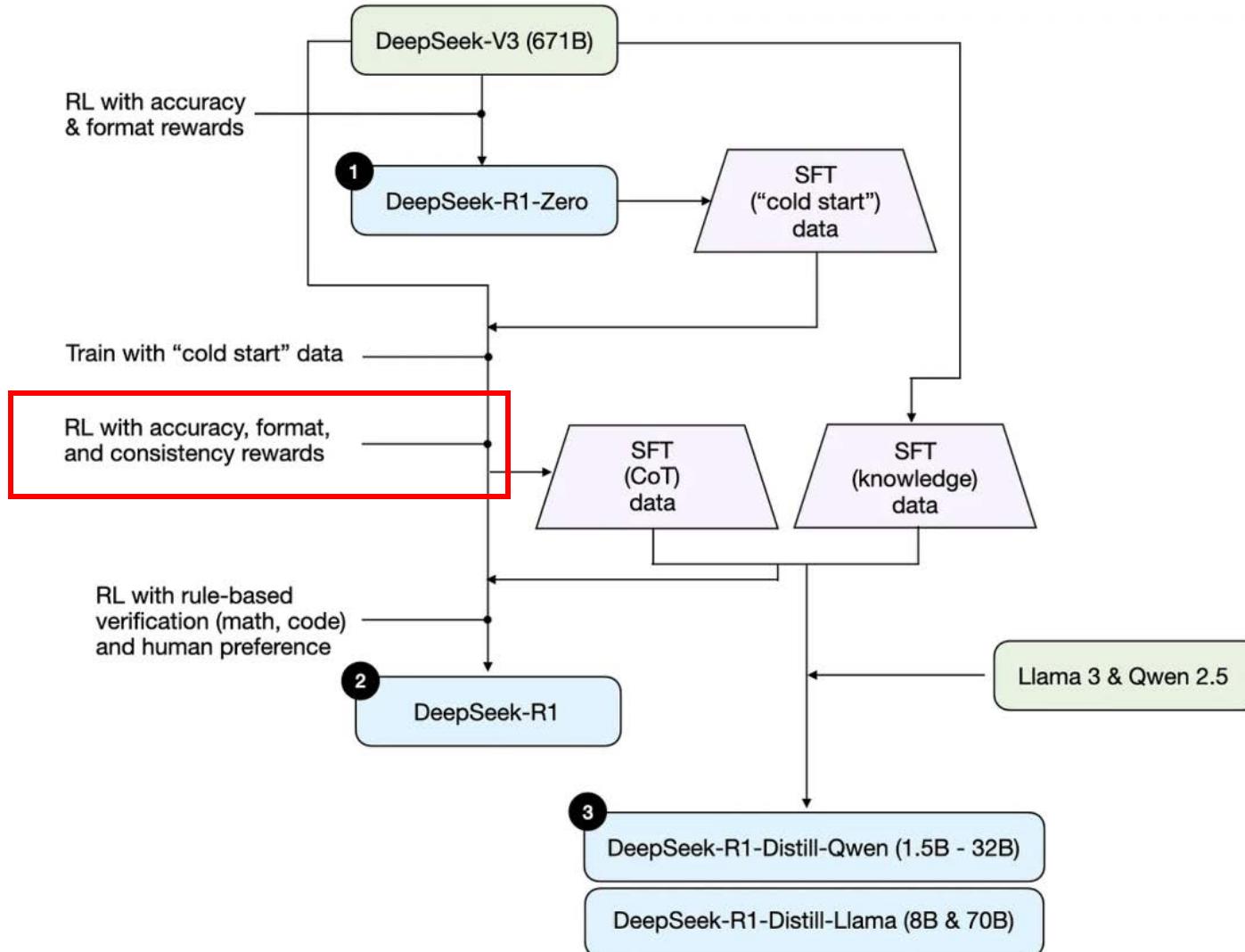
Step 1：推理过程数据收集（From R1-Zero）



Step 2：冷启动训练



第二步：Reasoning-oriented Reinforcement Learning

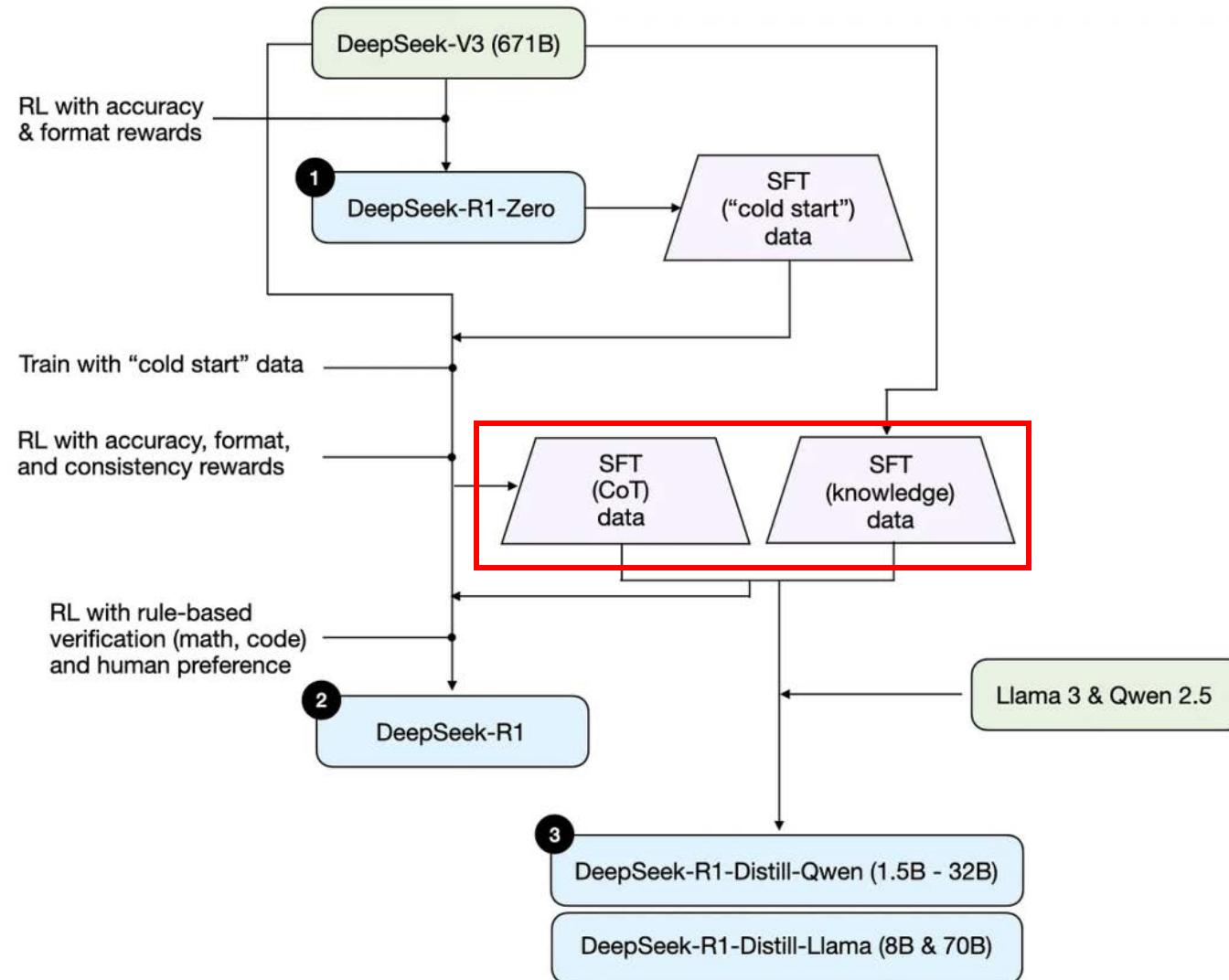


第二步：Reasoning-oriented Reinforcement Learning

- 与DeepSeek-R1-Zero一致的RL训练算法
 - 进一步提升模型推理能力
- + language consistency reward
 - 定义：输出CoT中目标语言单词的所占比例
 - 解决Language mixing问题

Total Reward = accuracy reward + format reward + language consistency reward

第三步：第二次Supervised Fine-tuning (SFT)



第三步：第二次Supervised Fine-tuning (SFT)

Cold start SFT

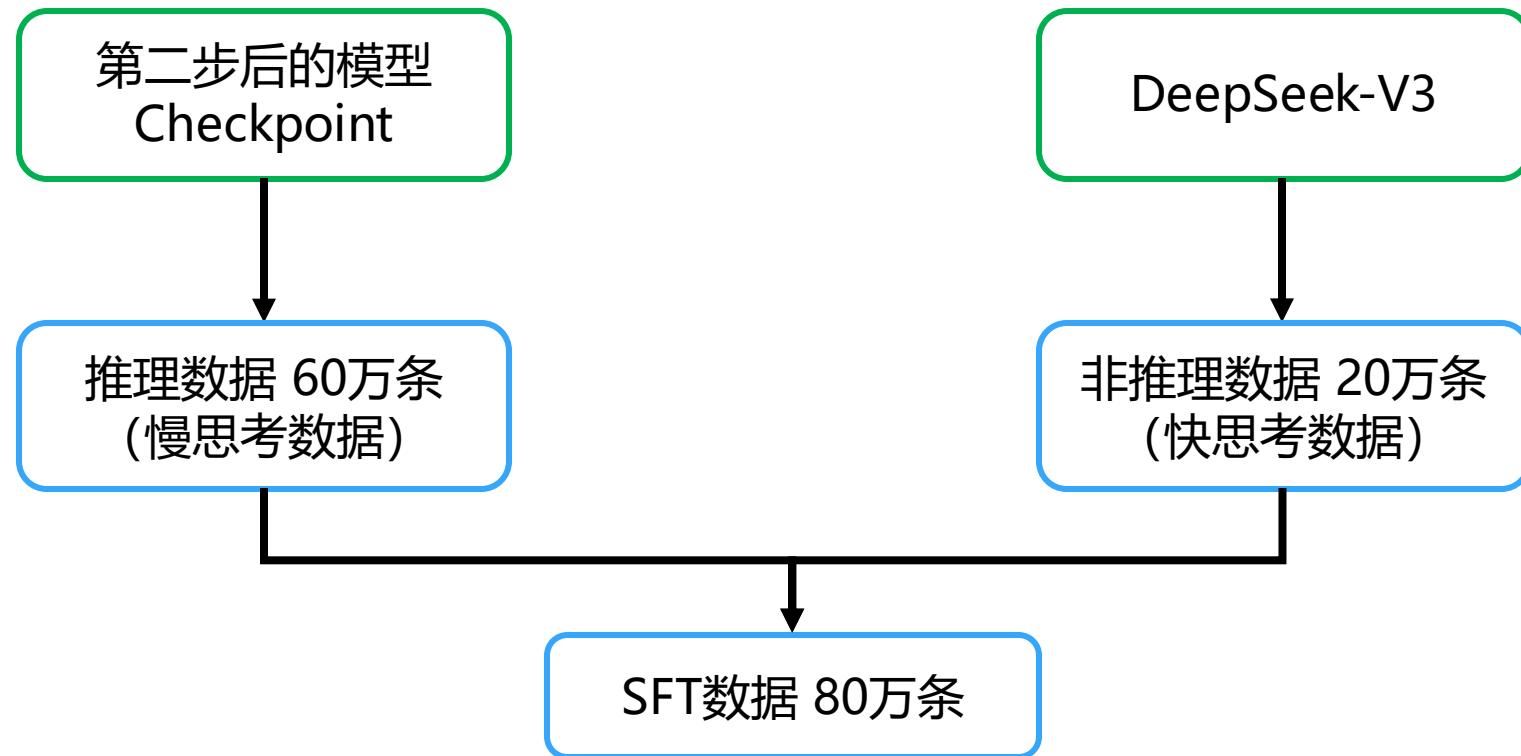
- 使用Long CoT数据
- 增强模型推理能力



第二次SFT

- 使用多领域、多问题数据
- 增强模型写作等通用任务能力

第三步：第二次Supervised Fine-tuning (SFT)



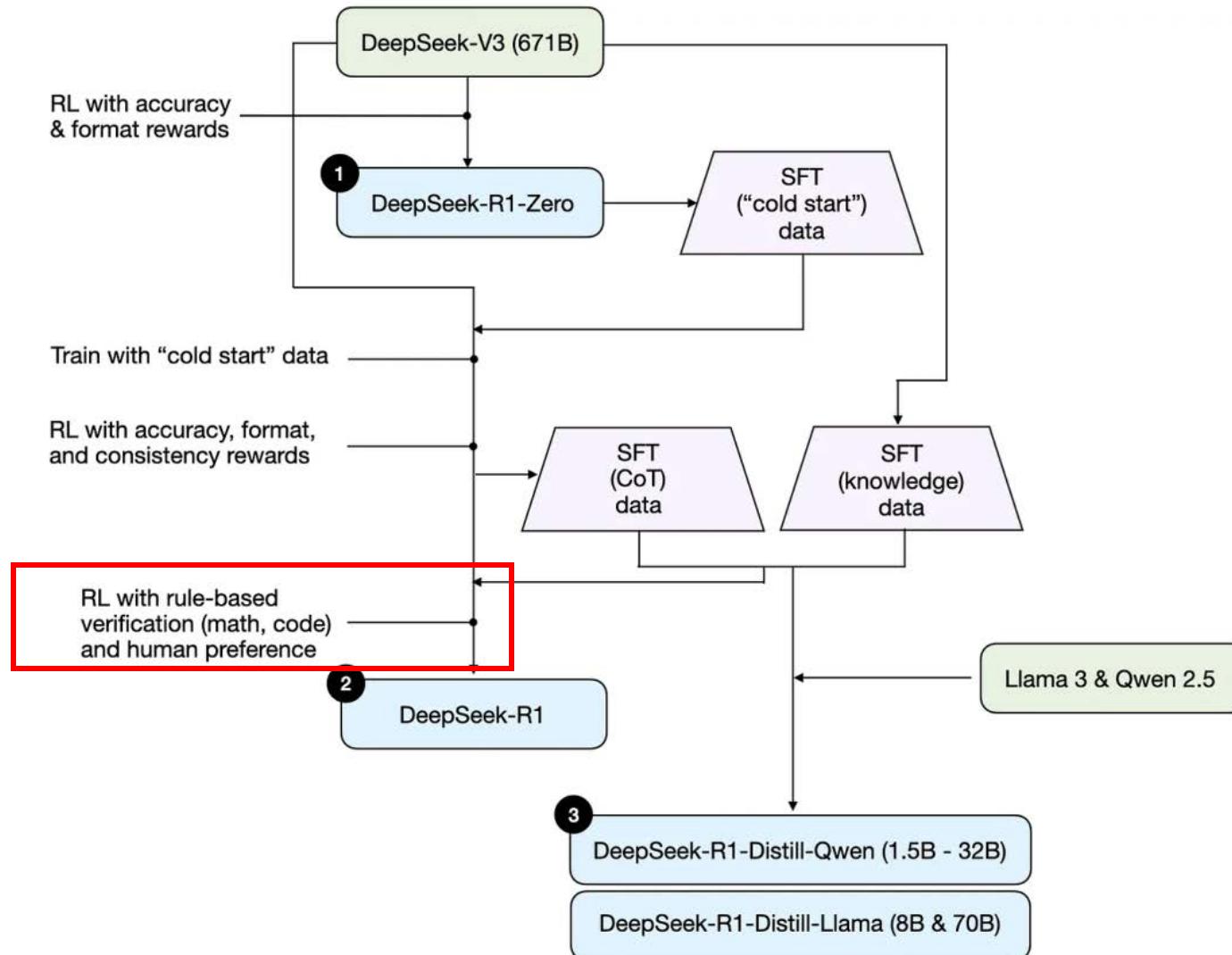
推理数据：

- 已有推理问题的推理过程轨迹：Prompt第二步后的Checkpoint模型，并收集其推理轨迹

非推理数据：

- 源于DeepSeek-V3 SFT过程的数据；通用NLP任务QA数据

第四步：Reinforcement learning for all scenarios



第四步：Reinforcement learning for all scenarios

第一次RL训练

- 与R1-Zero一致的训练过程
- 增强模型的推理能力



第二次RL训练

- 人类偏好对齐、有益性、无害性；
- 推理（次要地位）

第四步：Reinforcement learning for all scenarios

第二次RL训练

- 人类偏好对齐、有益性、无害性
- 推理（次要地位）

Reasoning data

- GRPO算法训练（与R1-Zero一致）

General data

- 基于已有人类偏好奖励模型，让模型的输出更符合人类期望

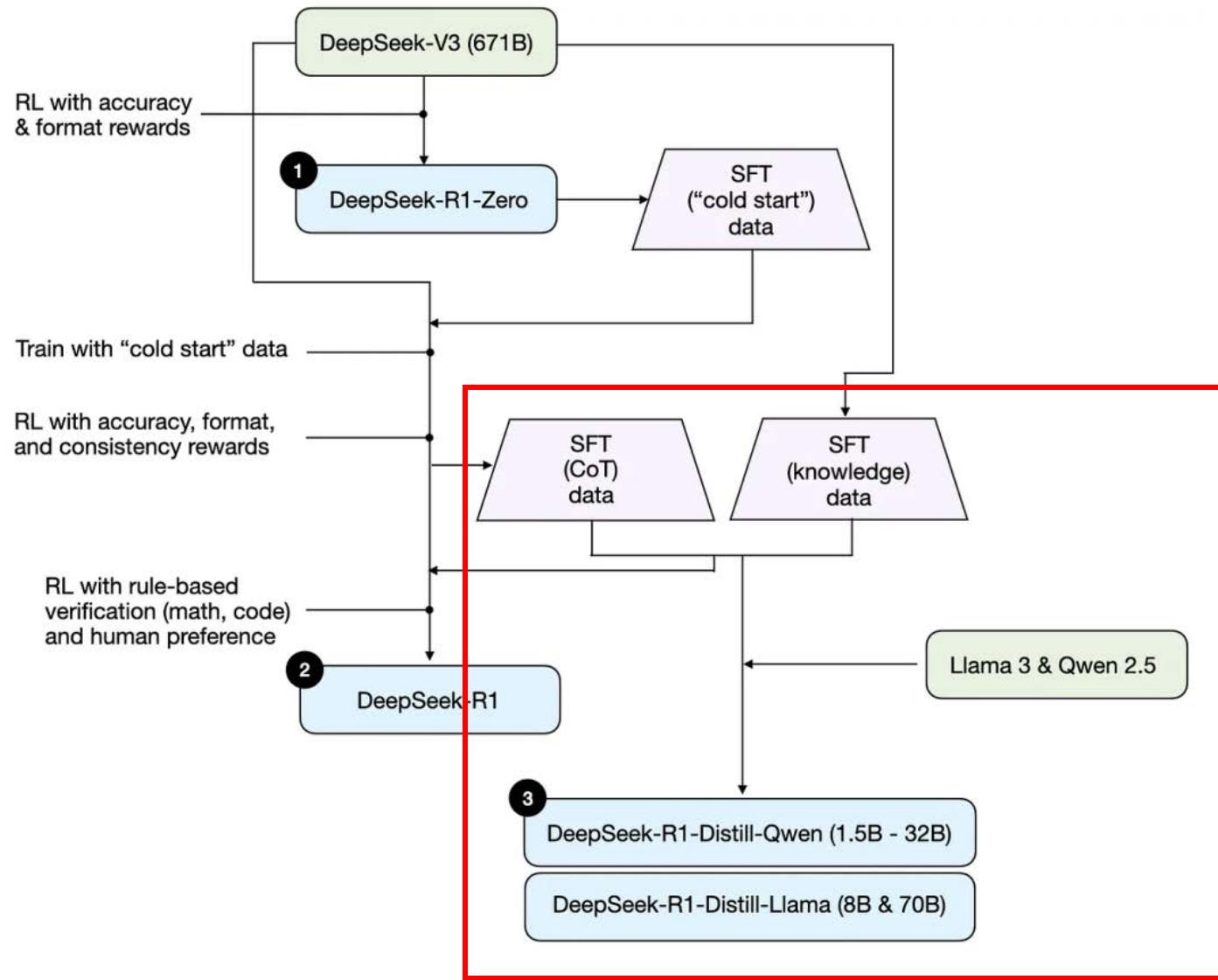
- **有益性**: 回复结果对用户的实用性、与prompt的相关性
- **无害性**: 回复内容无害

R1性能表现

| Benchmark (Metric) | Claude-3.5-Sonnet-1022 | GPT-4o-0513 | DeepSeek-V3 | OpenAI o1-mini | OpenAI o1-1217 | DeepSeek R1 |
|----------------------------|------------------------|-------------|-------------|----------------|----------------|-------------|
| Architecture | - | - | MoE | - | - | MoE |
| # Activated Params | - | - | 37B | - | - | 37B |
| # Total Params | - | - | 671B | - | - | 671B |
| MMLU (Pass@1) | 88.3 | 87.2 | 88.5 | 85.2 | 91.8 | 90.8 |
| MMLU-Redux (EM) | 88.9 | 88.0 | 89.1 | 86.7 | - | 92.9 |
| MMLU-Pro (EM) | 78.0 | 72.6 | 75.9 | 80.3 | - | 84.0 |
| DROP (3-shot F1) | 88.3 | 83.7 | 91.6 | 83.9 | 90.2 | 92.2 |
| IF-Eval (Prompt Strict) | 86.5 | 84.3 | 86.1 | 84.8 | - | 83.3 |
| GPQA Diamond (Pass@1) | 65.0 | 49.9 | 59.1 | 60.0 | 75.7 | 71.5 |
| SimpleQA (Correct) | 28.4 | 38.2 | 24.9 | 7.0 | 47.0 | 30.1 |
| FRAMES (Acc.) | 72.5 | 80.5 | 73.3 | 76.9 | - | 82.5 |
| AlpacaEval2.0 (LC-winrate) | 52.0 | 51.1 | 70.0 | 57.8 | - | 87.6 |
| ArenaHard (GPT-4-1106) | 85.2 | 80.4 | 85.5 | 92.0 | - | 92.3 |
| LiveCodeBench (Pass@1-COT) | 38.9 | 32.9 | 36.2 | 53.8 | 63.4 | 65.9 |
| Codeforces (Percentile) | 20.3 | 23.6 | 58.7 | 93.4 | 96.6 | 96.3 |
| Codeforces (Rating) | 717 | 759 | 1134 | 1820 | 2061 | 2029 |
| SWE Verified (Resolved) | 50.8 | 38.8 | 42.0 | 41.6 | 48.9 | 49.2 |
| Aider-Polyglot (Acc.) | 45.3 | 16.0 | 49.6 | 32.9 | 61.7 | 53.3 |
| AIME 2024 (Pass@1) | 16.0 | 9.3 | 39.2 | 63.6 | 79.2 | 79.8 |
| MATH-500 (Pass@1) | 78.3 | 74.6 | 90.2 | 90.0 | 96.4 | 97.3 |
| CNMO 2024 (Pass@1) | 13.1 | 10.8 | 43.2 | 67.6 | - | 78.8 |
| CLUEWSC (EM) | 85.4 | 87.9 | 90.9 | 89.9 | - | 92.8 |
| C-Eval (EM) | 76.7 | 76.0 | 86.5 | 68.9 | - | 91.8 |
| C-SimpleQA (Correct) | 55.4 | 58.7 | 68.0 | 40.3 | - | 63.7 |

- 在多种Benchmark数据集上取得与OpenAI-o1相当的性能

DeepSeek-R1-Distill

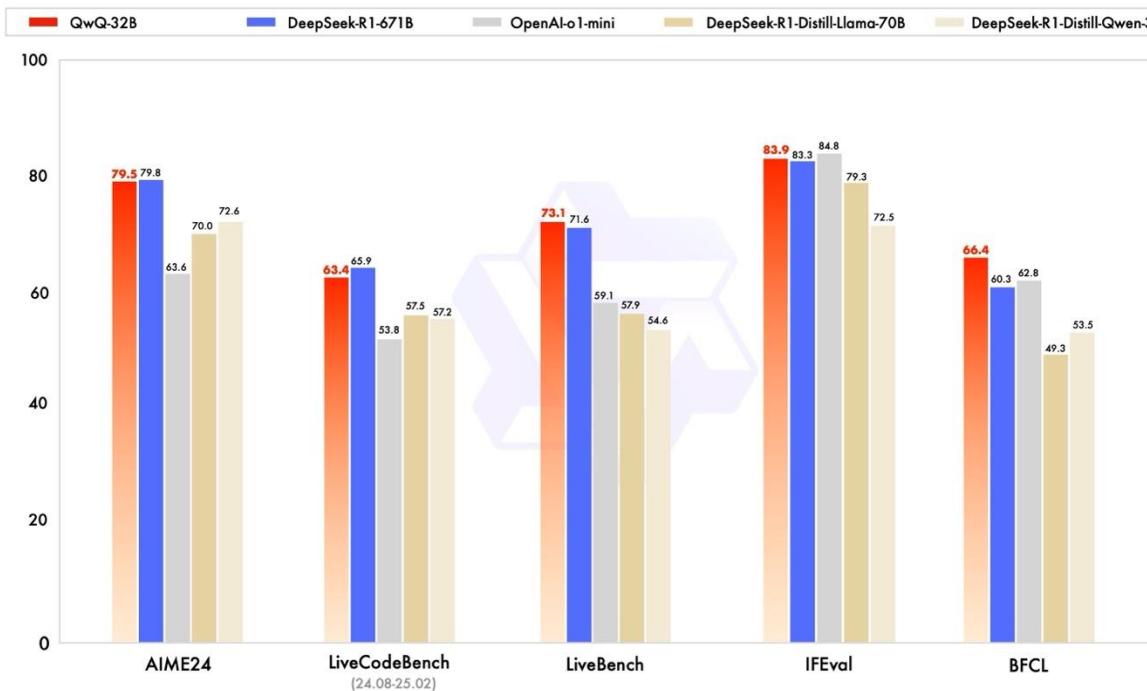


蒸馏

- R1的训练过程积累了60万条推理数据和20万条非推理数据
- 将这些数据用于微调Llama3 & Qwen 2.5等小模型，经过简单的SFT即可直接使用

DeepSeek-R1-Distill

| Model | AIME 2024 | | MATH-500 | GPQA Diamond | LiveCodeBench |
|------------------------------|-----------|---------|----------|--------------|---------------|
| | pass@1 | cons@64 | pass@1 | pass@1 | pass@1 |
| QwQ-32B-Preview | 50.0 | 60.0 | 90.6 | 54.5 | 41.9 |
| DeepSeek-R1-Zero-Qwen-32B | 47.0 | 60.0 | 91.6 | 55.0 | 40.2 |
| DeepSeek-R1-Distill-Qwen-32B | 72.6 | 83.3 | 94.3 | 62.1 | 57.2 |



- **R1论文中结论：**对小模型进行与R1-Zero一致的RL训练，性能与使用R1 SFT阶段的数据进行蒸馏存在较大差距
- 但近期QwQ-32B通过RL也取得了与R1-671B相当的性能^[1]
 - 同样遵循冷启动SFT + 两次RL的过程，但无技术细节报告

[1] <https://qwenlm.github.io/blog/qwq-32b/>

Unsuccessful Attempts of Developing DeepSeek-R1

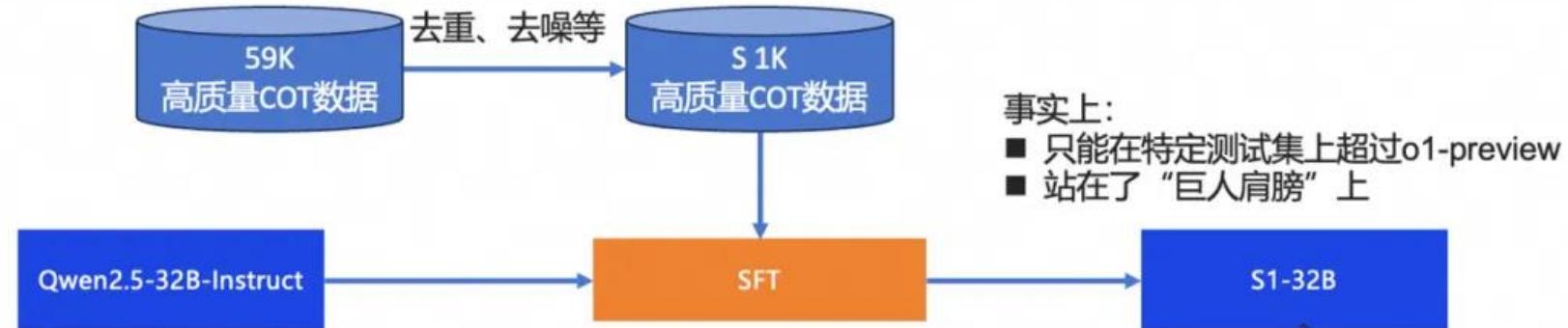
Process Reward Model (PRM)

- 难以问题定义fine-grain的解决步骤
- 对于复杂问题，难确定中间推理过程是否合理、正确
- PRM容易导致reward hacking问题
- 训练PRM需要额外资源

Monte Carlo Tree Search (MCTS)

- 推理过程中的搜索：与象棋不同，语言token的搜索空间指数增长
- 用于指导搜索过程的value model直接影响生成质量，但训练基于语言的精细化value model十分困难

“50美元复现R1（的性能）”



s1 by Stanford (Feifei Li)^[1]

- 不依赖大规模强化学习，仅用少量高质量数据对基座模型进行监督微调 (SFT)
- **预算强制 (Budget Forcing)**
 - 用时间换效果：通过强制延长生成过程或强制终止，控制模型思考时间或进行多少步操作
 - 如果模型过早结束推理，系统会鼓励模型延长思考时间，确保其充分考虑问题
- **高质量小数据集 s1K**
 - 过滤低质量或格式错误的样本；排除模型能轻易解决的问题；覆盖 50 个数学和科学领域
 - 后续被发现使用其他基座模型效果不好：站在了Qwen这个“巨人”的肩膀上

[1] Muennighoff, Niklas, et al. "s1: Simple test-time scaling." arXiv preprint arXiv:2501.19393 (2025).

高效智能体设计

Reasoning Big Model: OpenAI o1-preview

September 12, 2024

Introducing OpenAI o1-preview

A new series of reasoning models for solving hard problems. Available now.

Update on September 17, 2024: Rate limits are now 50 queries per week for o1-preview and 50 queries per day for o1-mini.

We've developed a new series of AI models designed to spend more time thinking before they respond. They can reason through complex tasks and solve harder problems than previous models in science, coding, and math.



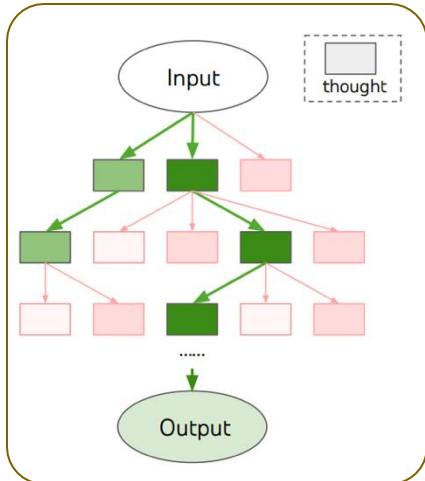
@DrJimFan

| Model | Pricing |
|------------|-----------------------------------|
| gpt-4o | \$2.50 / 1M input tokens |
| | \$1.25 / 1M cached** input tokens |
| | \$10.00 / 1M output tokens |
| o1-preview | \$15.00 / 1M input tokens |
| | \$7.50 / 1M cached* input tokens |
| | \$60.00 / 1M output** tokens |

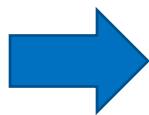
*Output tokens include **internal reasoning tokens** generated by the model that are not visible in API responses.

Introduce more thought searching in inference stage. **Token cost is 6X!**

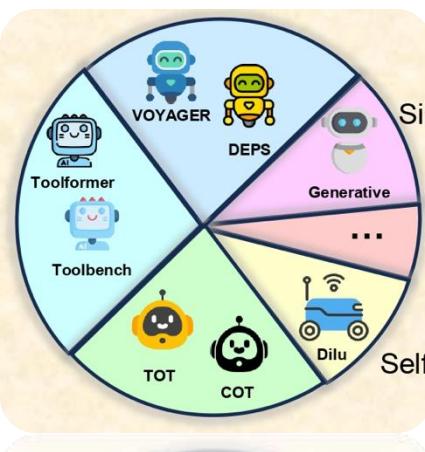
Research Challenges



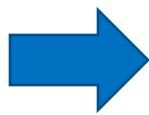
Computation Cost



Human cognition-inspired
efficient reasoning framework

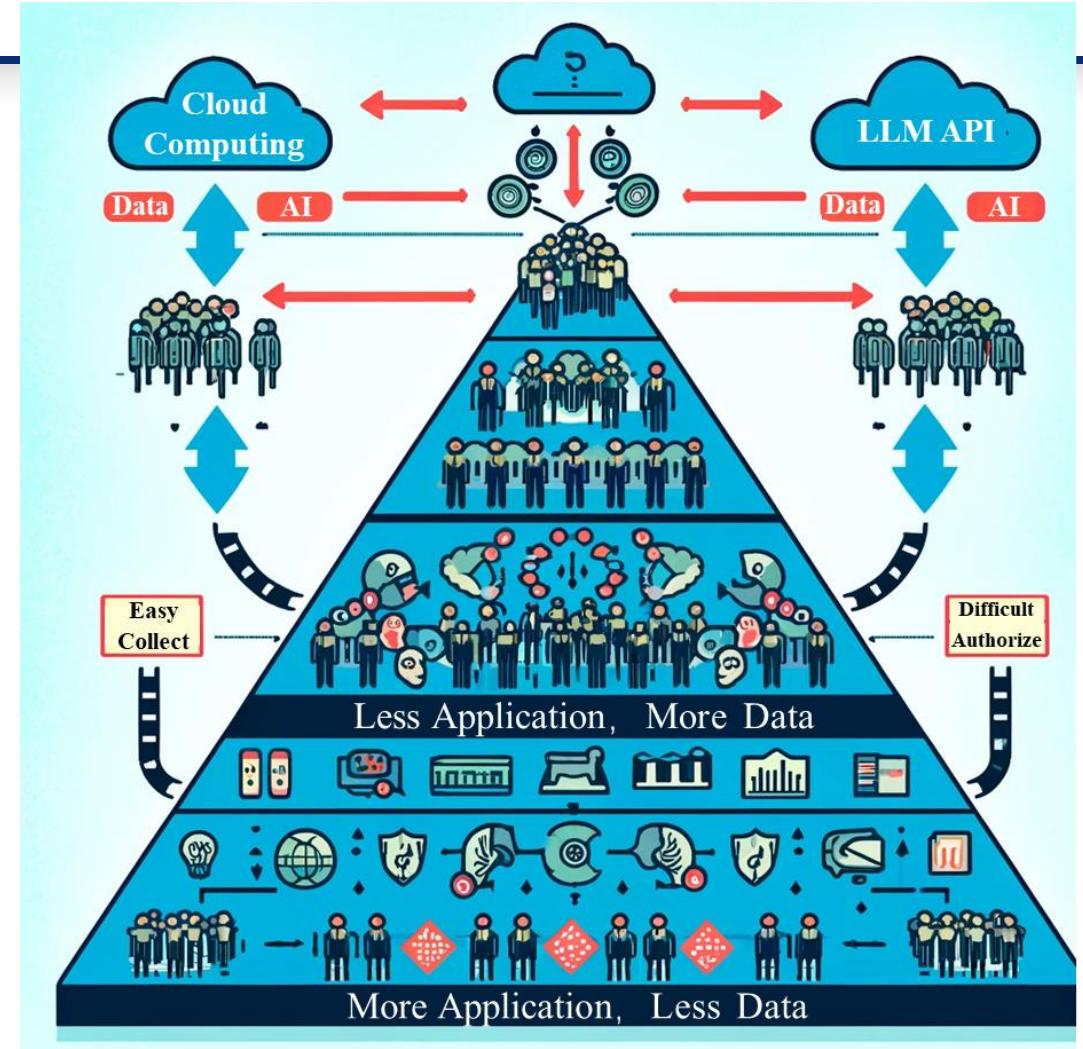
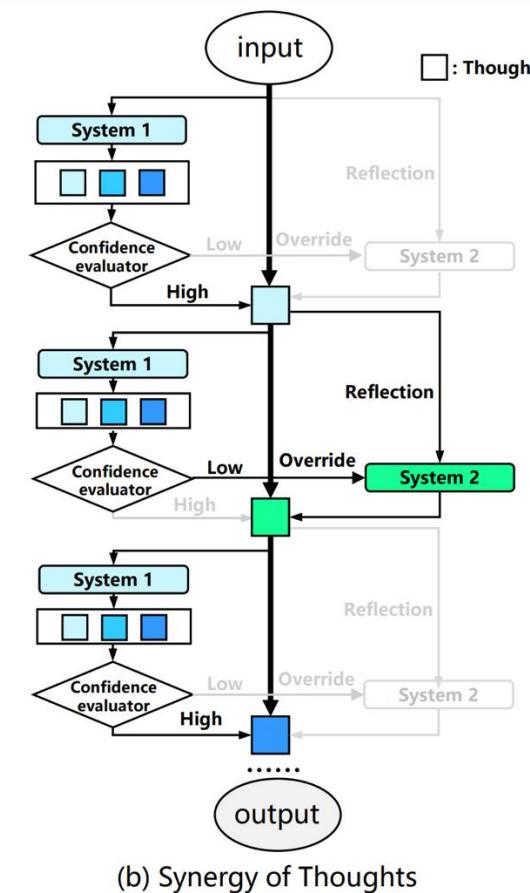
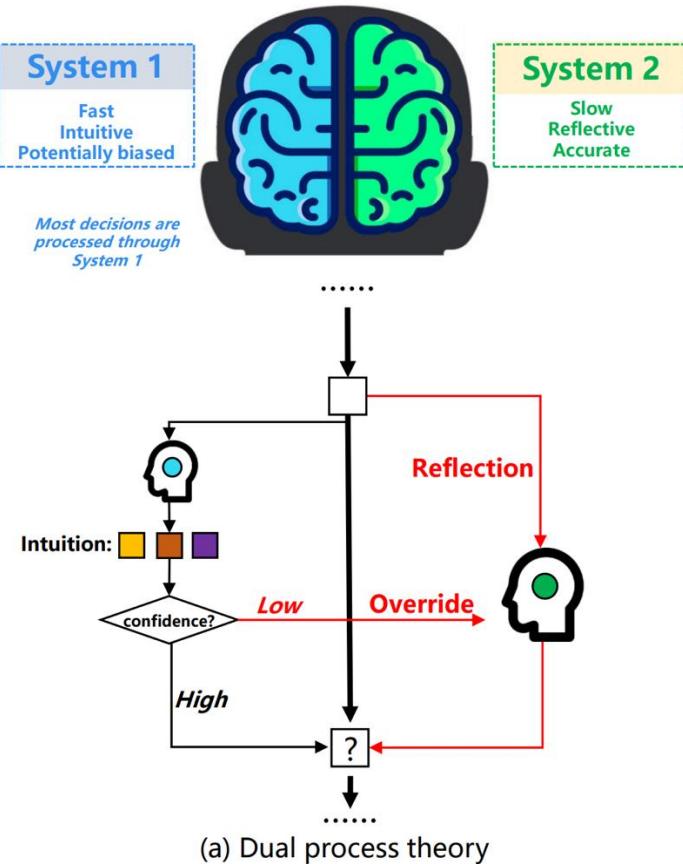


Agent Design



Automatic optimization
framework for new tasks

Edge-Cloud Synergy

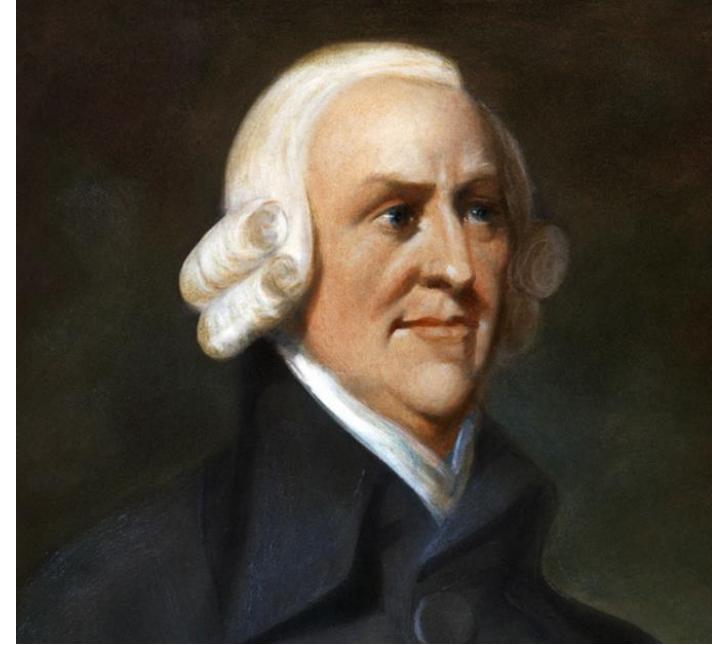


Synergy of Small-Large Models

Synergy of Edge-Cloud

- Low cost
- Privacy
- Personalized

Inspiration from Division-of-Labor

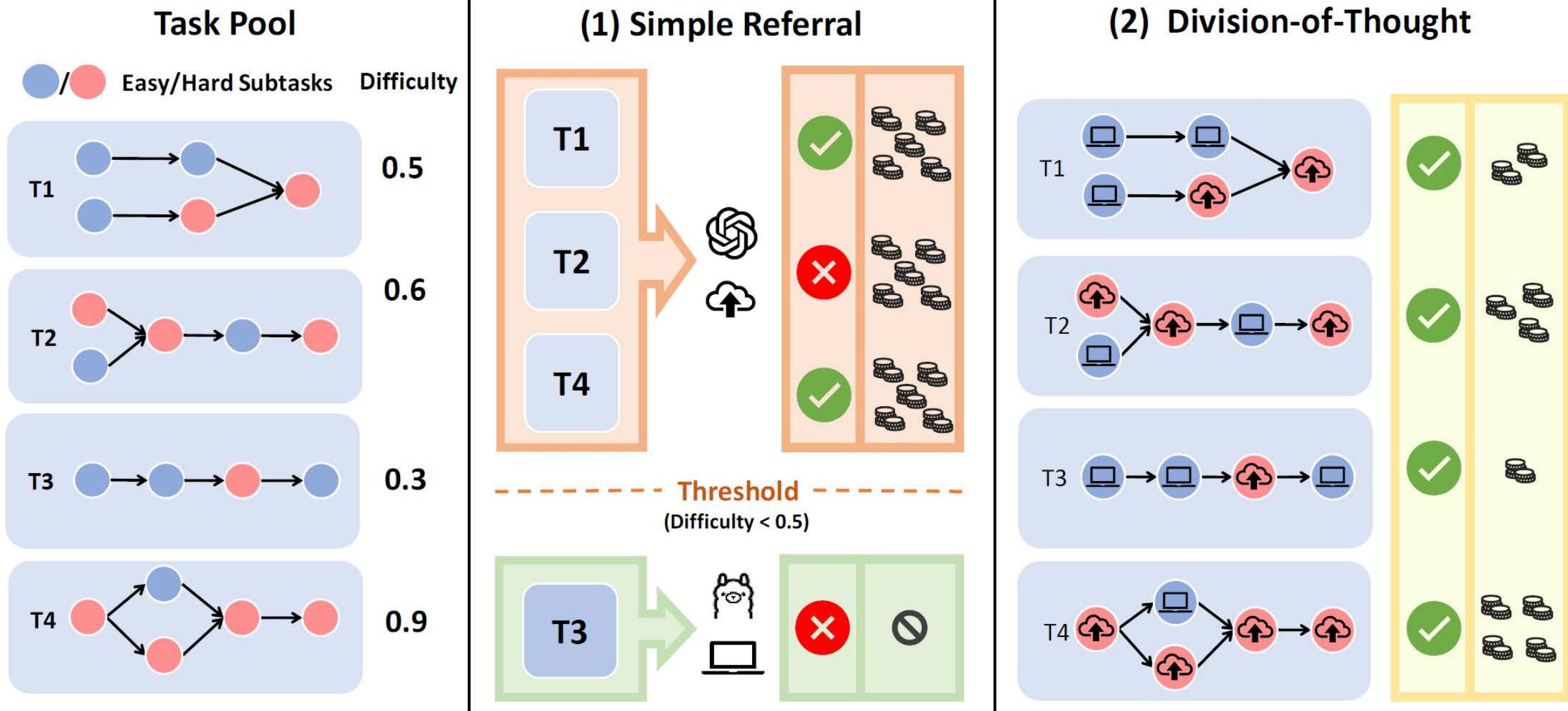


Adam Smith

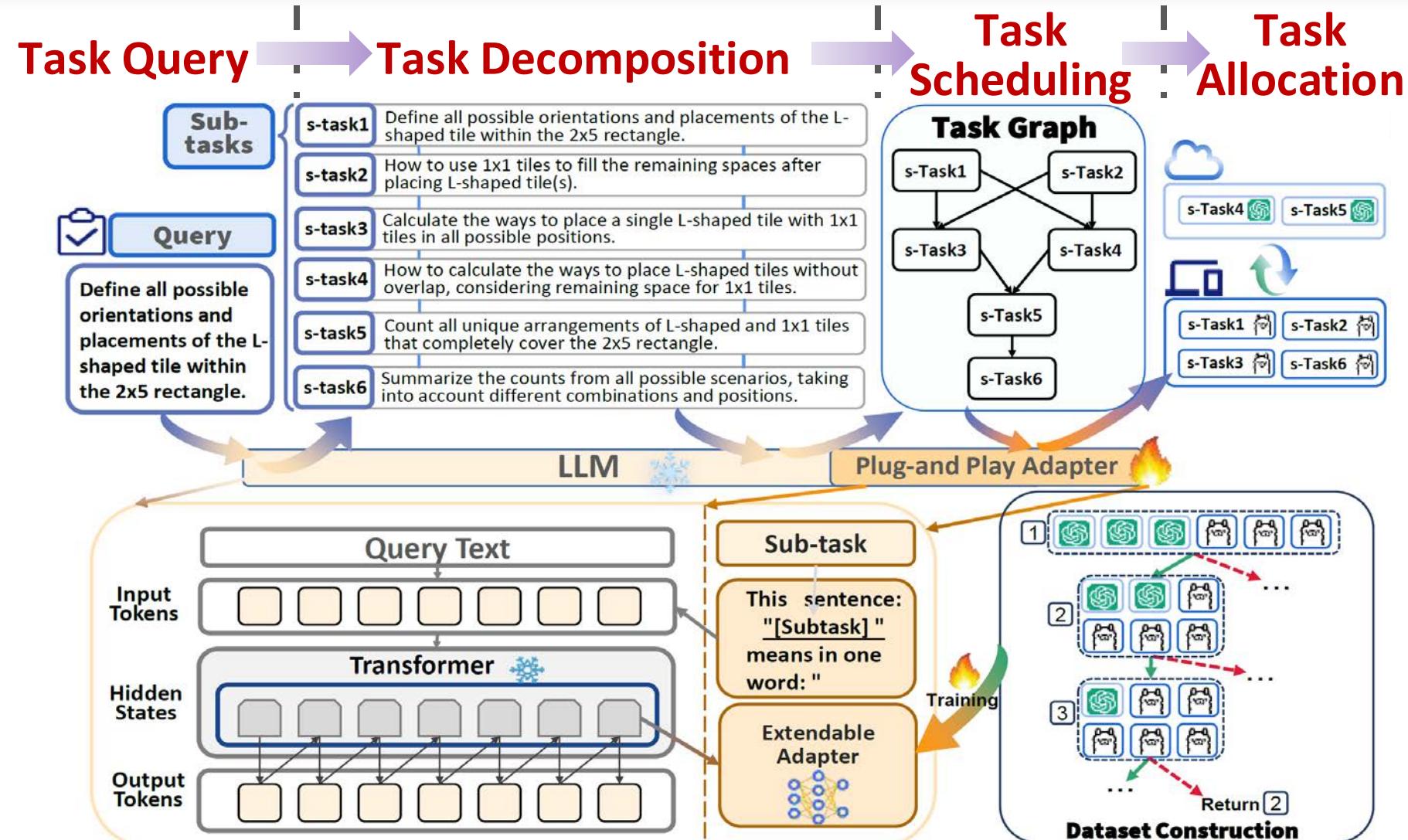
*Prosperity is a consequence
of increasing division of labor.*

Task decomposition facilitate efficient collaboration at finer levels.

Division-of-Thoughts V.S. Simple Referral



Division-of-Thoughts Framework

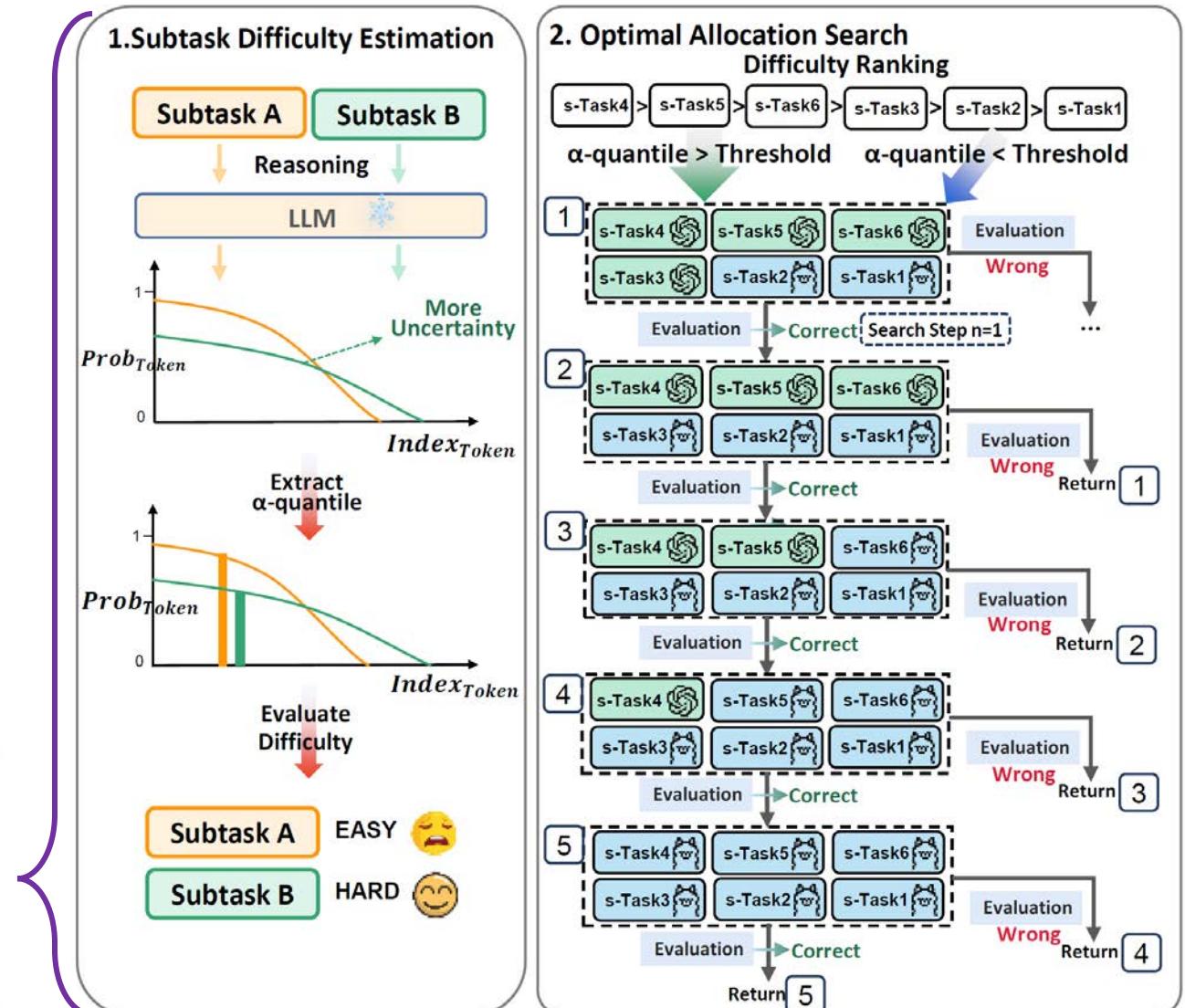
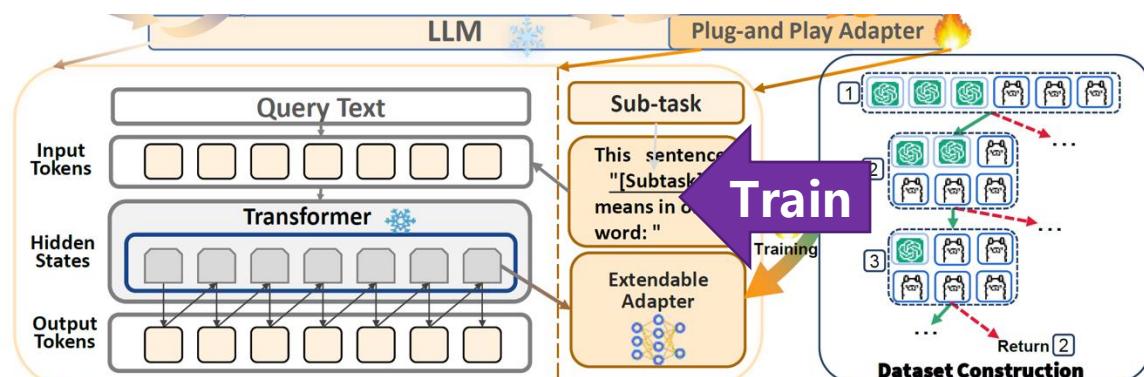


Shao, Chenyang, et al. "Division-of-thoughts: Harnessing hybrid language model synergy for efficient on-device agents." *Proceedings of the ACM on Web Conference 2025*. 2025.

Reinforced Tree Search for Data Construction

| Methods | SLM Ratio | SR | # Evaluation | API Cost |
|----------------------|-----------|--------|--------------|----------|
| Zero-Shot LLM | 53.11% | 92.78% | 1 | \$2.56 |
| Binary-Search | 69.46% | 88.45% | 8.456 | \$16.45 |
| α -tree (n=1) | 85.53% | 99.44% | 3.4589 | \$7.34 |
| α -tree (n=2) | 86.45% | 96.34% | 2.3445 | \$5.12 |

- Bootstrapping task allocation quality @ *SLM ratio* and *success rate*.
- Require far less #evaluation compared to binary search.



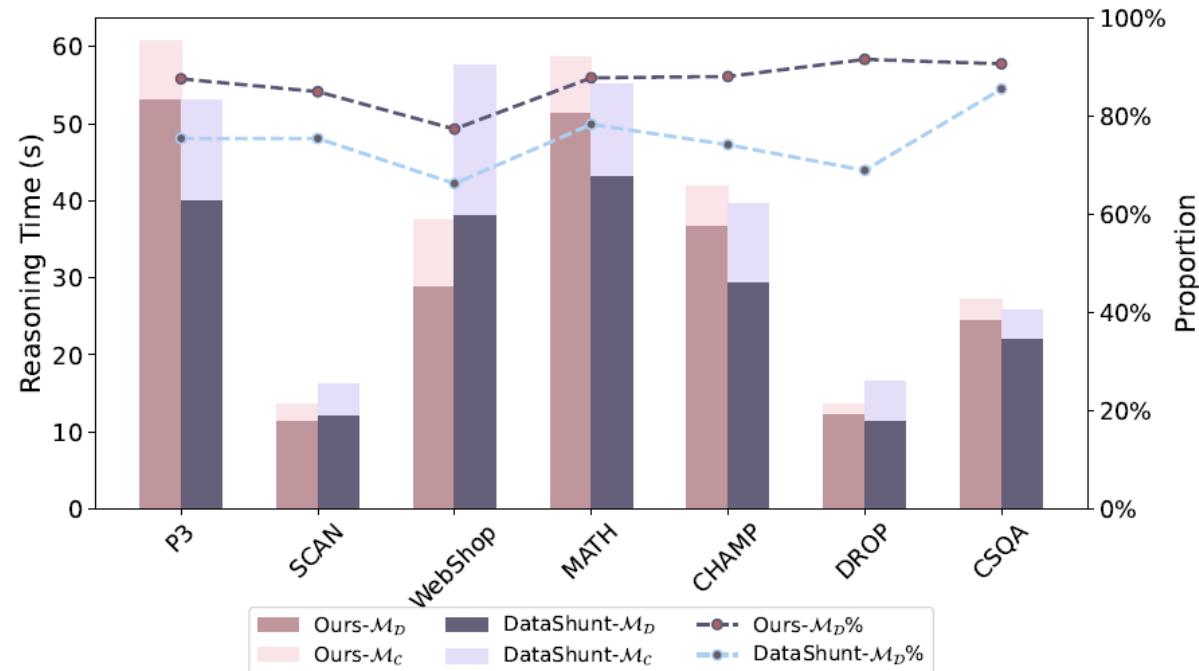
α – Tree Search

Experiment

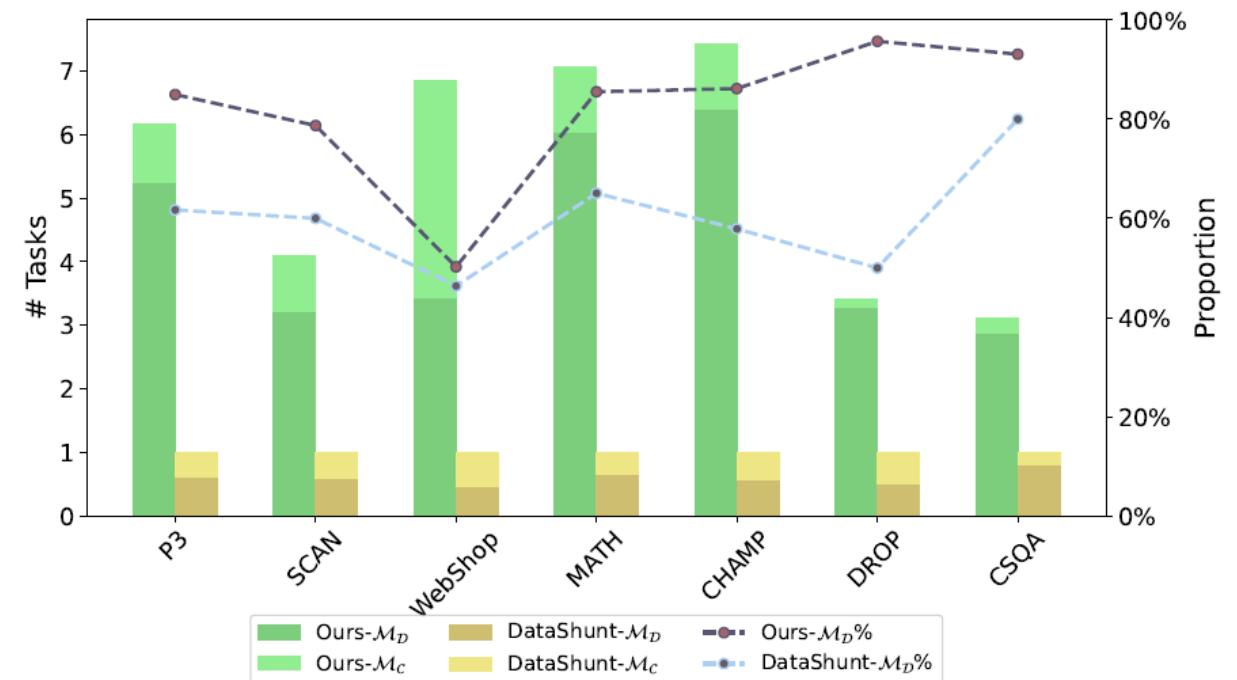
| Model | Logical Reasoning | | | | | | Web Browsing | | | Solving Math Problems | | | | | | Commonsense Reasoning | | | | | |
|------------------|-------------------|-------------------|------------------|--------|-------------------|------------------|--------------|-------------------|------------------|-----------------------|-------------------|------------------|--------|-------------------|------------------|-----------------------|-------------------|------------------|-------|-------------------|------------------|
| | P3 | | | SCAN | | | WebShop | | | MATH | | | CHAMP | | | DROP | | | CSQA | | |
| | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} | Acc | C _{Time} | C _{Api} |
| COT (GPT-4o) | 42% | 35.8 | 4.45¢ | 68% | 9.21 | 2.75¢ | 35% | 30.9 | 10.65¢ | 51.5% | 34.5 | 5.34¢ | 55.5% | 26.4 | 4.45¢ | 80% | 11.6 | 1.30¢ | 80% | 17.0 | 3.60¢ |
| TOT (GPT-4o) | 38% | 93.1 | 14.55¢ | 52% | 32.5 | 9.82¢ | 36% | 62.4 | 47.34¢ | 63% | 60.5 | 9.97¢ | 57% | 64.2 | 11.65¢ | 80.5% | 40.2 | 5.41¢ | 82% | 98.8 | 20.50¢ |
| COT (Llama 3-8B) | 5.5% | 18.1 | N/A | 17% | 5.0 | N/A | 0.0% | 10.5 | N/A | 10% | 21.1 | N/A | 19% | 13.1 | N/A | 72% | 3.8 | N/A | 70% | 8.4 | N/A |
| TOT (Llama 3-8B) | 5.5% | 58.3 | N/A | 13% | 21.8 | N/A | 1.4% | 22.5 | N/A | 29.5% | 49.0 | N/A | 25% | 68.1 | N/A | 65% | 27.8 | N/A | 68.5% | 89.4 | N/A |
| DataShunt | 14% | 25.1 | 2.45¢ | 23.5% | 7.6 | 1.72¢ | 34% | 30.9 | 8.35¢ | 16% | 24.9 | 1.66¢ | 34% | 19.1 | 2.98¢ | 74% | 8.6 | 0.60¢ | 73% | 10.4 | 1.28¢ |
| DoT (ours) | 41% | 23.5 | 1.58¢ | 63% | 5.5 | 1.20¢ | 31% | 17.2 | 4.97¢ | 59% | 22.6 | 1.02¢ | 58% | 16.1 | 0.84¢ | 85% | 4.9 | 0.32¢ | 82% | 9.9 | 0.49¢ |
| Improvement | ↓2.38% | ↓34.36% | ↓64.49% | ↓7.35% | ↓40.28% | ↓56.36% | ↓13.89% | ↓72.43% | ↓89.95 | ↓6.35% | ↓62.72% | ↓89.50% | ↑1.75% | ↓74.92% | ↓92.79% | ↑5.59% | ↓87.81% | ↓94.09% | 0% | ↓89.98% | ↓97.60% |

- Maintaining comparable reasoning accuracy as GPT-4o, while reducing average **inference time** and **API costs** by **66.12%** and **83.57%**, respectively.
- Achieve substantial performance gain in reasoning accuracy compared to **pure SLM** or **simple referral** solutions.

Experiment

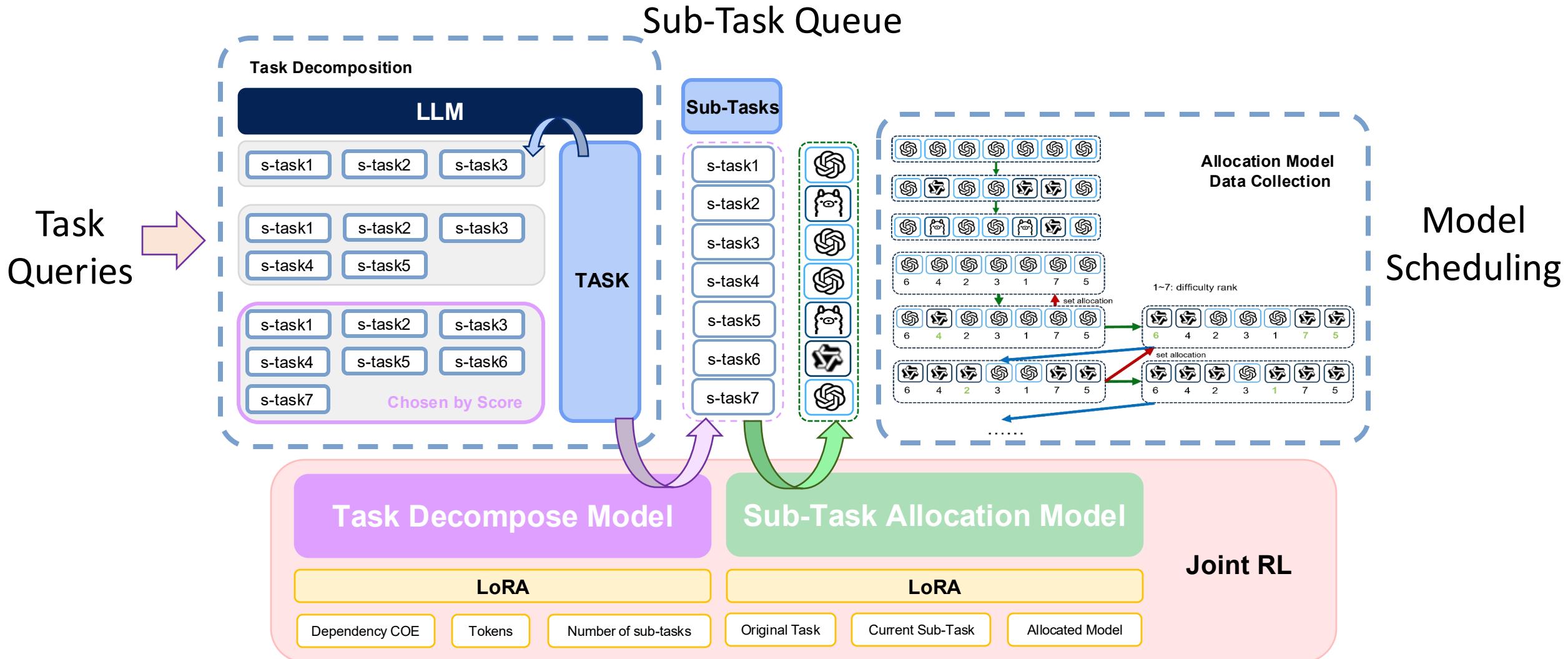


The proportion of SLMs in reasoning time.



The proportion of SLMs in the number of assigned tasks.

Future Work – Intelligence As A Service

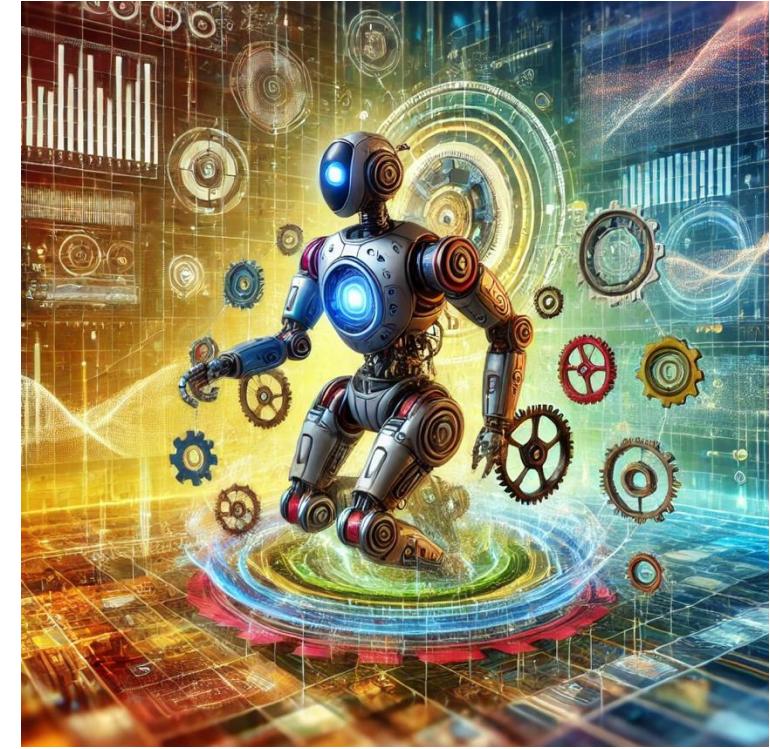
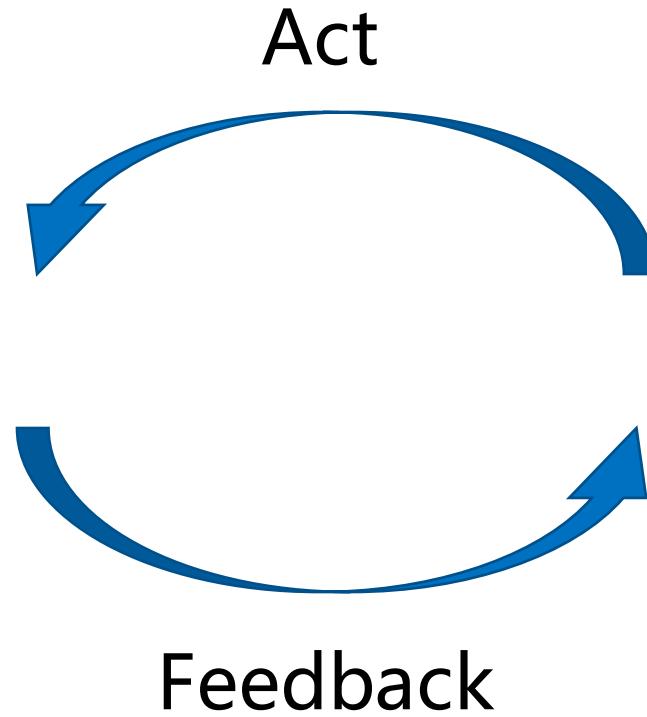


Scale up Edge-cloud Collaboration via RL

Self-adaptive LLM Agent Search

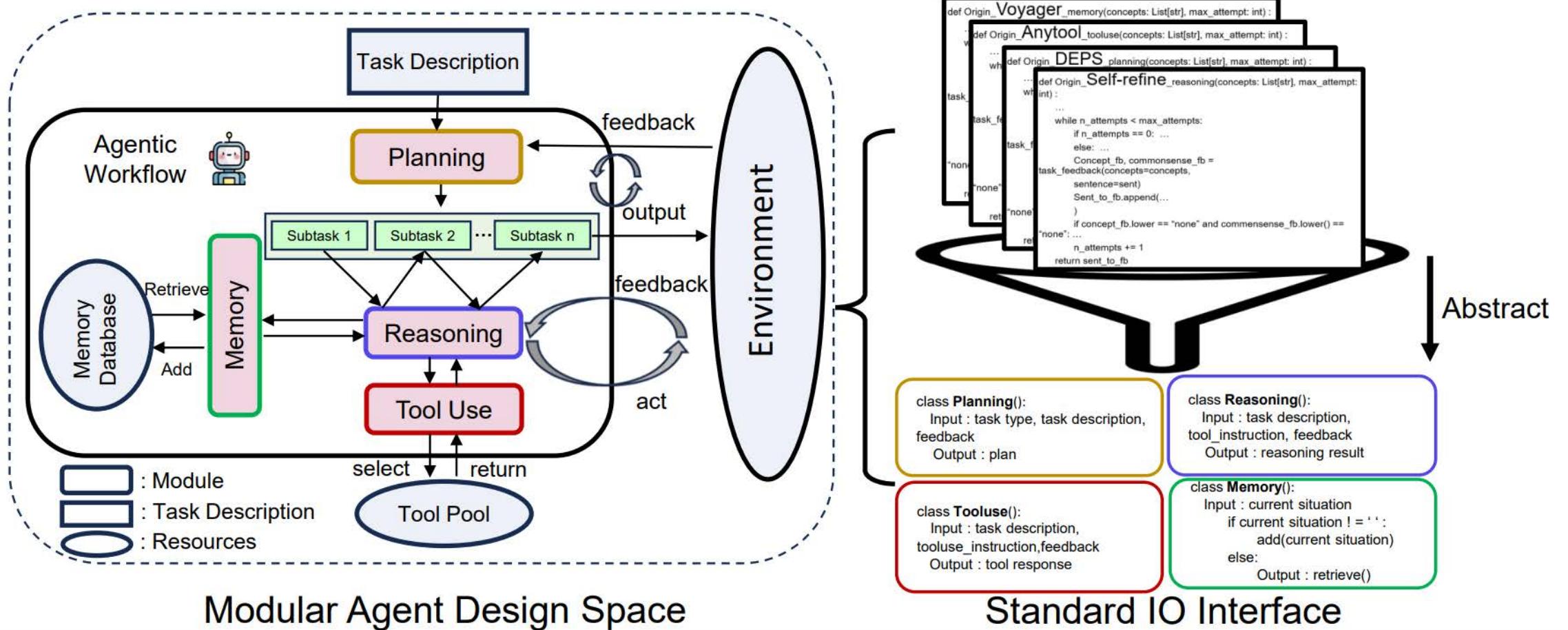


Diverse Task
Scenarios



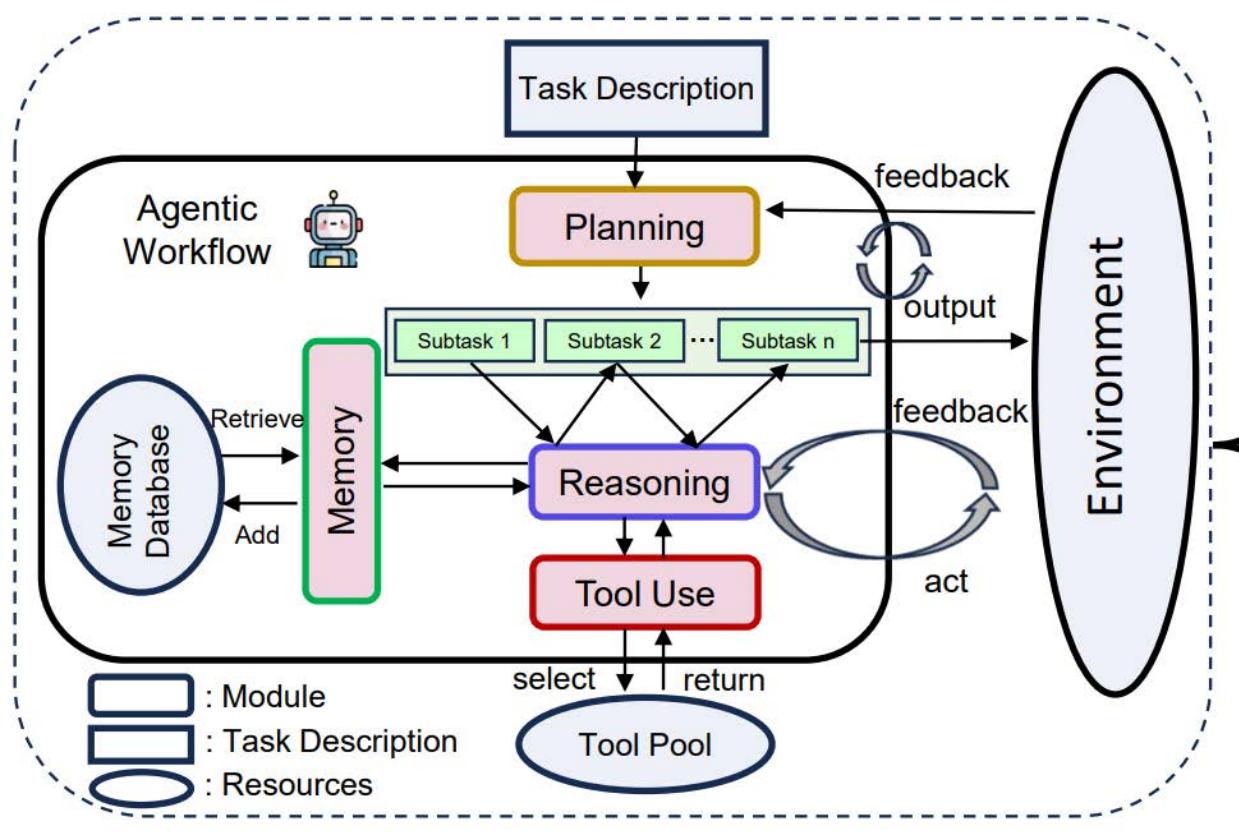
Self-adapt LLM Agent

Modular Design Space



We analyze 16 high-impact LLM Agents, abstract a design space of 4 modules: **Planning(6x)**, **Reasoning (7x)**, **Tool Use (5x)**, **Memory(5x)**, leading to **1050 possible agent designs**.

Modular Design Space



Modular Agent Design Space

We analyze 16 high-impact LLM Agents, abstract a design space of 4 modules: **Planning(6x)**, **Reasoning (7x)**, **Tool Use (5x)**, **Memory(5x)**, leading to **1050 possible agent designs**.

class Planning():
Input : task type, task description, feedback
Output : plan

class Reasoning():
Input : task description, tool_instruction, feedback
Output : reasoning result

Decompose complex queries into sub-tasks

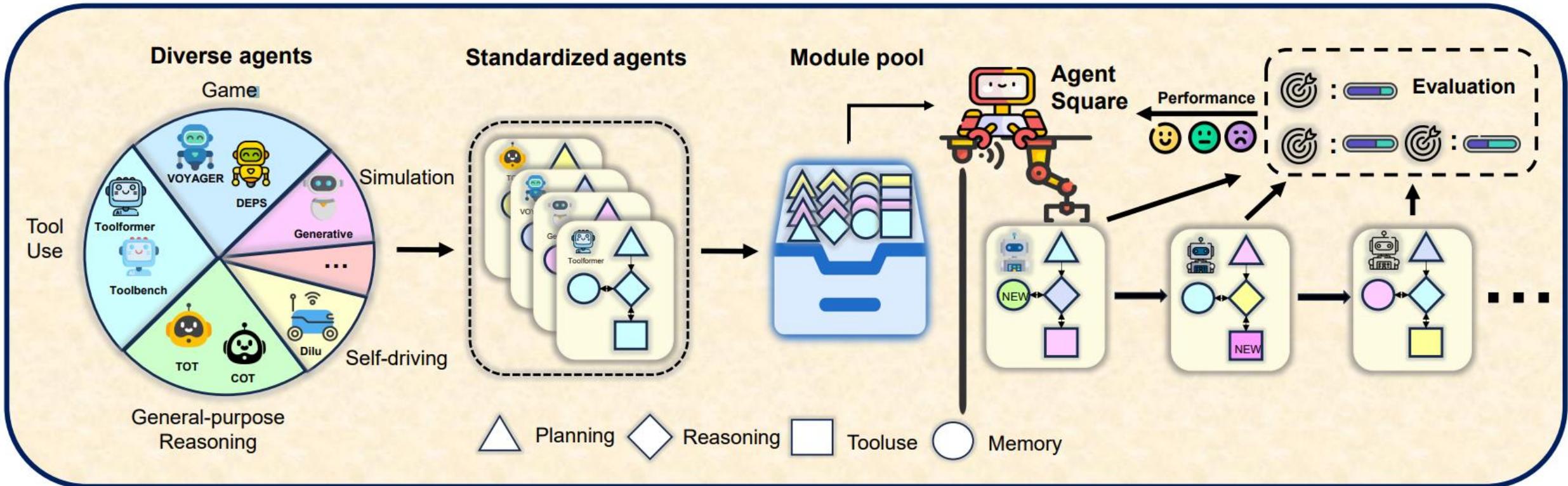
class Tooluse():
Input : task description, tooluse_instruction, feedback
Output : tool response

class Memory():
Input : current situation
if current situation != '' :
add(current situation)
else:
Output : retrieve()

Whether and **How** to use **Which** tool

Augmenting decision through memory read & write

Modular LLM Agent Search

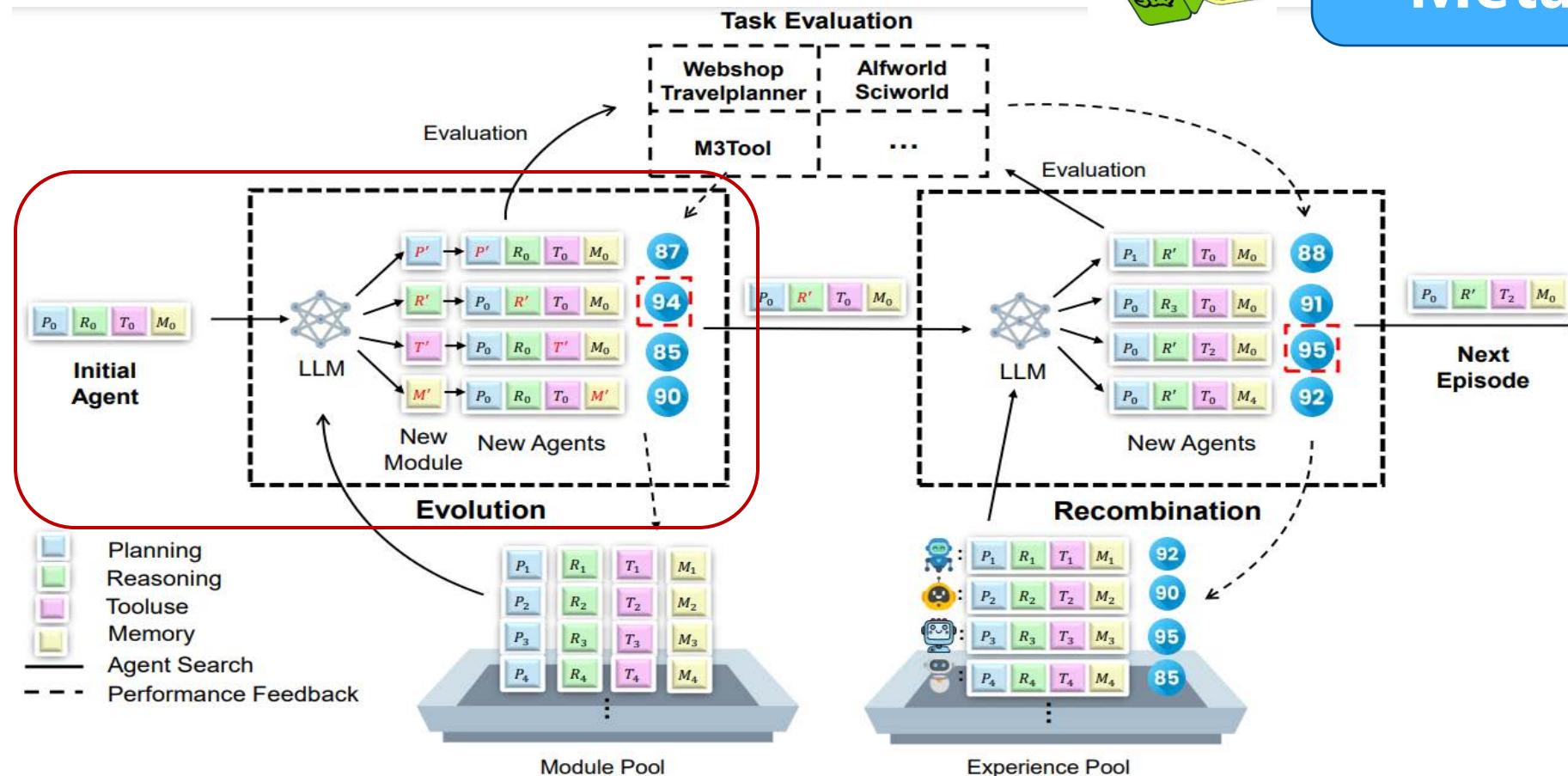


Modular LLM Agent Search (MoLAS):

- How can we efficiently search for optimal module combinations?
- How can we discover novel, better modules?

AgentSquare Framework

Agent² as a
Meta-Agent



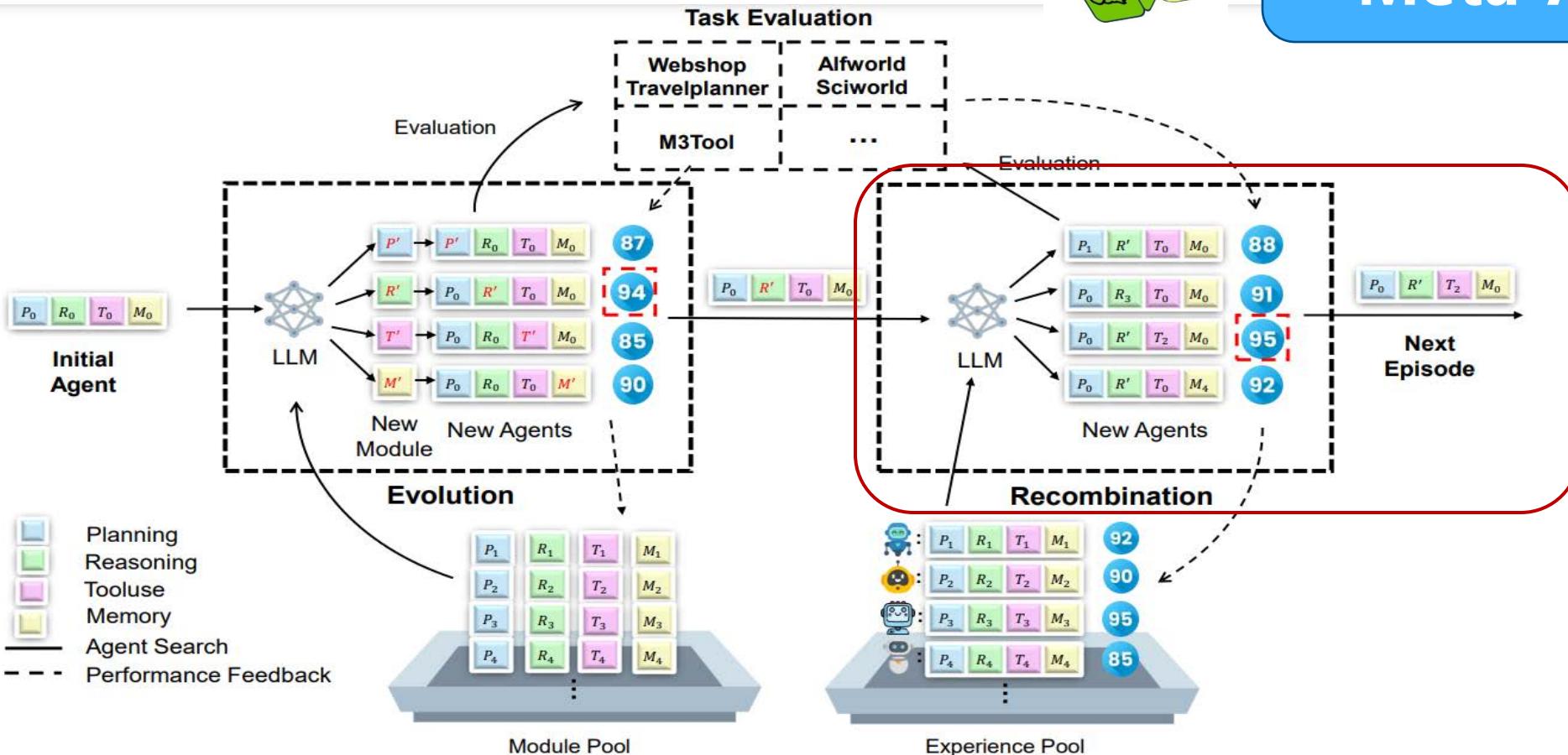
Module Evolution (Prompt-level Search):

- Analyze Task Description, Existing Module Designs, and Design Motivation to propose novel modules.
- Select beneficial new modules via task evaluation.

AgentSquare Framework



Agent² as a
Meta-Agent

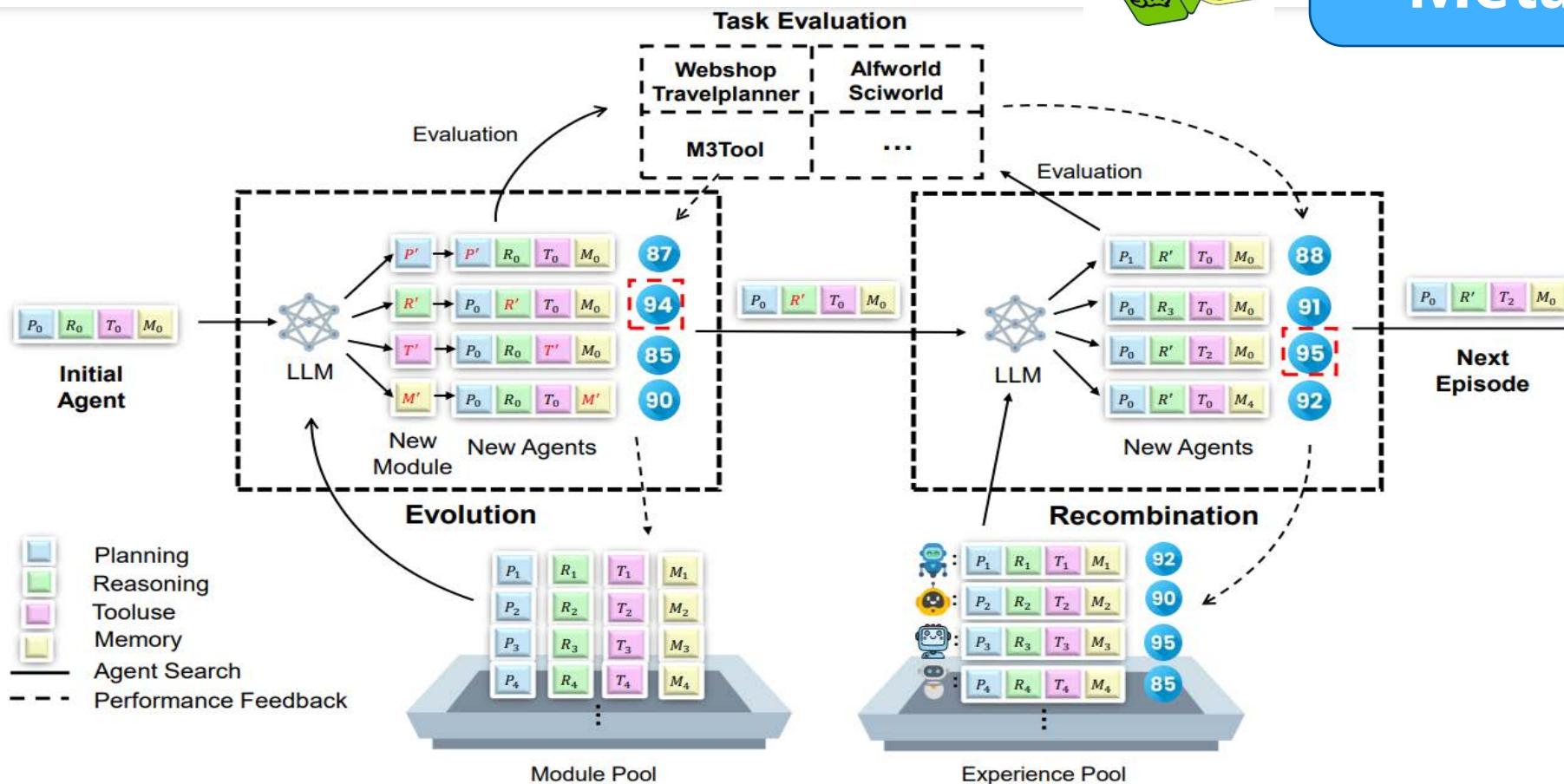


Module Recombination (Module-level Search):

- Analyze Task Description, Searched Combination, and Performance Feedback to explore new combinations.
- Select good performing combinations via task evaluation.

AgentSquare Framework

Agent² as a
Meta-Agent



In-context Surrogate Model (Performance Predictor):

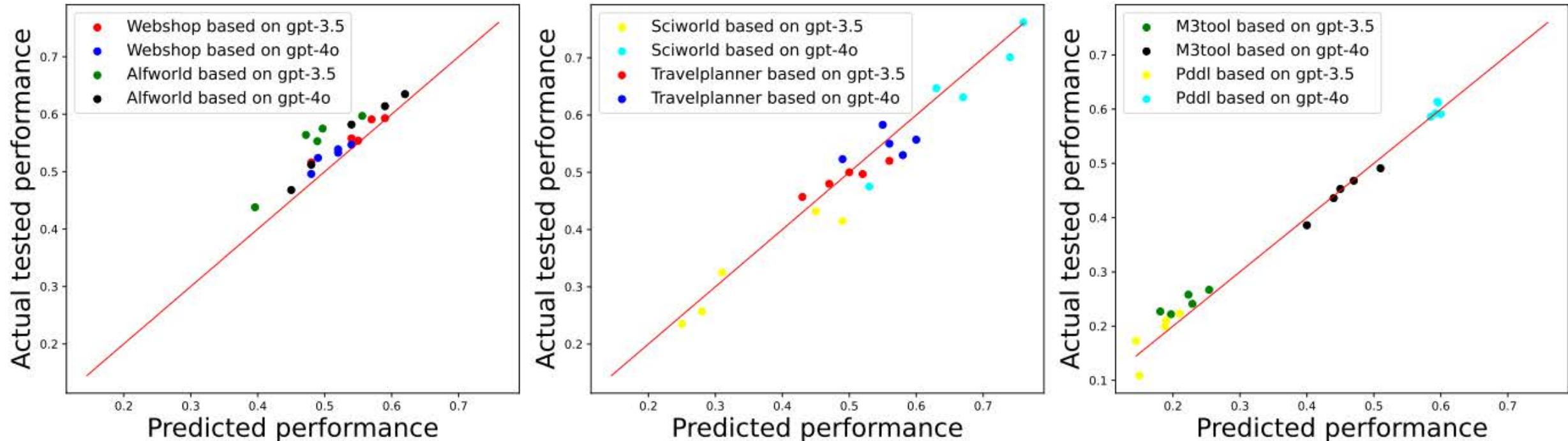
- Retrieve similar agent design, and perform in-context few-shot prediction for new LLM agents.
- Skip unpromising agent candidates, only evaluate those promising designs.

Experiment

| Baseline Type | Method | Web | Embodied | Tool | Game | | |
|---------------------|-------------------|--------------|--------------|--------------|--------------|---------------|--------------|
| | | Webshop | ALFWorld | SciWorld | M3Tool | TravelPlanner | PDDL |
| Hand-crafted Agents | CoT | 0.485 | 0.405 | 0.697 | 0.448 | 0.487 | 0.542 |
| | Cot-SC | 0.512 | 0.426 | 0.656 | 0.461 | 0.413 | 0.495 |
| | Self-refine | 0.461 | 0.567 | 0.654 | 0.442 | 0.000 | 0.514 |
| | ToT | 0.501 | 0.437 | 0.741 | 0.453 | 0.380 | 0.476 |
| | Step Back | 0.468 | 0.279 | 0.220 | 0.434 | 0.000 | 0.486 |
| | TP | 0.398 | 0.404 | 0.576 | 0.387 | 0.430 | 0.518 |
| | HuggingGPT | 0.519 | 0.481 | 0.680 | 0.354 | 0.510 | 0.584 |
| | Voyager | 0.366 | 0.425 | 0.776 | 0.247 | 0.523 | 0.412 |
| | Generative Agents | 0.499 | 0.477 | 0.663 | 0.402 | 0.480 | 0.553 |
| | DEPS | 0.481 | 0.459 | 0.740 | 0.278 | 0.540 | 0.591 |
| Module Search | OPENAGI | 0.506 | 0.510 | 0.718 | 0.322 | 0.533 | 0.616 |
| | Dilu | 0.451 | 0.433 | 0.682 | 0.475 | 0.360 | 0.463 |
| Prompt Search | Random | 0.533 | 0.620 | 0.704 | 0.438 | 0.563 | 0.660 |
| | Bayesian | 0.549 | 0.634 | 0.749 | 0.502 | 0.537 | 0.650 |
| Prompt Search | OPRO | 0.505 | 0.380 | 0.569 | 0.309 | 0.523 | 0.589 |
| | AgentSquare | 0.607 | 0.695 | 0.781 | 0.524 | 0.583 | 0.669 |

- *AgentSquare* beats best human designs by **17.2% on average** across diverse task scenarios.

Experiment



- In-context surrogate model predict agent performance with $R^2=0.95$.
- It reduces the token cost to 0.025% compared to actual evaluation of the LLM Agent designs.

Discovering Novel LLM Agents

| Task | Planning | Reasoning | Tooluse | Memory | Best Hand-crafted Agents |
|---------------|----------|-----------|---------|-------------------|--------------------------|
| Webshop | IO | HTSS | / | Dilu | HuggingGPT |
| ALFWORLD | TD | SF-ToT | / | Generative Agents | Self-refine |
| SciWorld | Voyager | CoT | / | Hier | Voyager |
| M3Tool | / | CoT-SC | ToolBF | / | Toolbench |
| TravelPlanner | DEPS | CoT | TH | / | DEPS |
| PDDL | IR | CASRC | / | Generative Agents | OPENAGI |

SF-TOT

Insights: Develop a module that not only generates multiple paths and evaluates them but also incorporates self-reflection and self-improvement strategies.

```
class REASONING_SF_TOT():
    def __call__(self, ...):
        prompt = f'''Interact with a
                    household to solve a task. Your
                    instructions must follow the
                    examples. Here are some
                    examples. ... '''
        responses = llm_response(...)
        response = self.get_votes(...)
        response = self.refine(response, ...)
        return response
```

TD

Insights: Use a timeline-based approach, where tasks are decomposed into sub-tasks with explicit temporal dependencies.

```
class PLANNING_TD():
    def __call__(self, ...):
        prompt = f'''You are a planner who
                    divides a {task_type} task into
                    several subtasks with explicit
                    temporal dependencies. Consider
                    the order of actions and their
                    dependencies to ensure logical
                    sequencing. ...'''
        sub_plans = llm_response(...)
        return sub_plans
```

Planning (6+2)
Reasoning (7+3)
Tool Use (5+2)
Memory (5+1)

Agent² = 3360



AgentSquare: Automatic LLM Agent Search In Modular Design Space (2024)

[Paper](#) [Code](#)

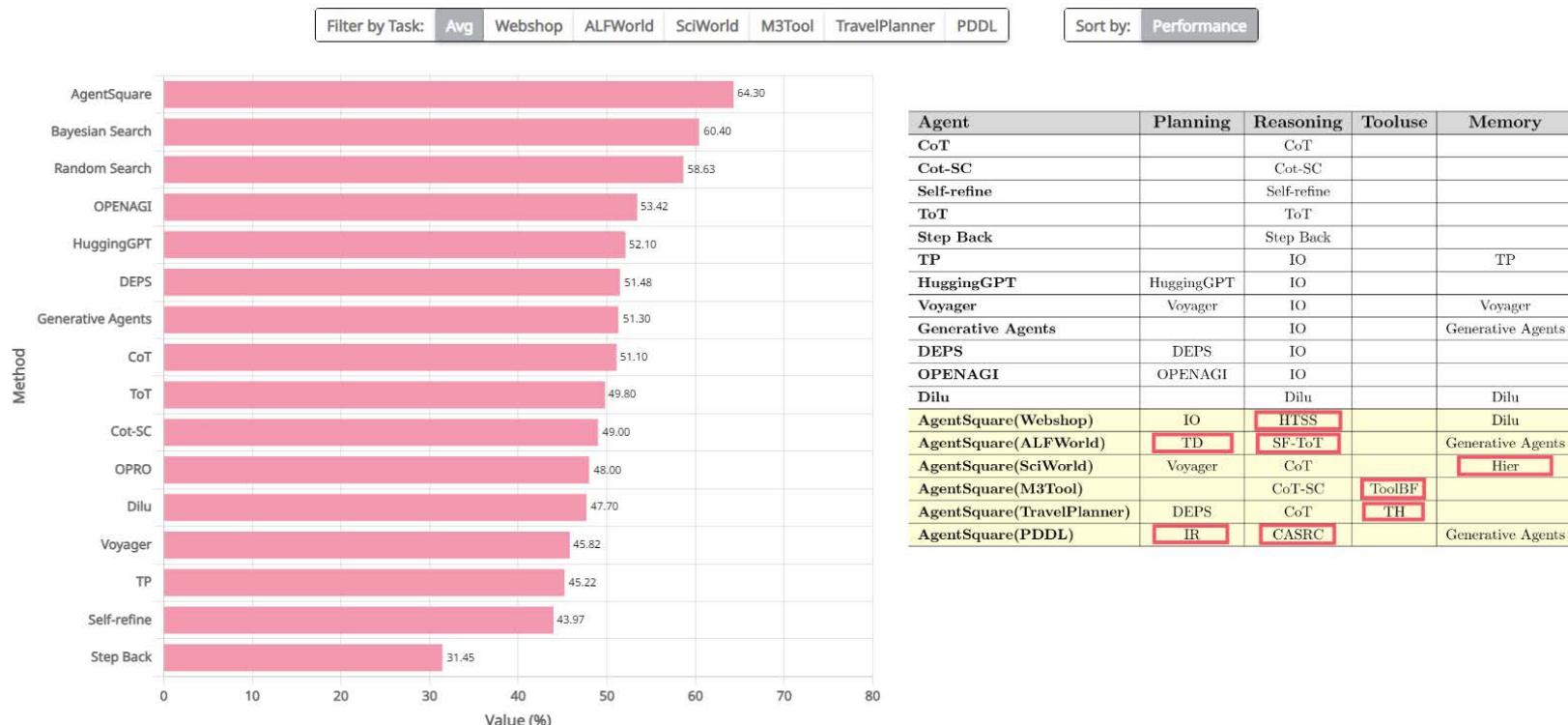
<Scan to Access>

We Challenge You to Contribute New Modules!

About AgentSquare

AgentSquare is a novel LLM agent search framework that utilizes module evolution and recombination to efficiently optimize agent designs across a modular design space consisting of Planning, Reasoning, Tool Use, and Memory modules.

Main Performance of different Methods across various tasks and the modules that different agents contain are shown below, please check our [New module](#) for more details about New Modules.

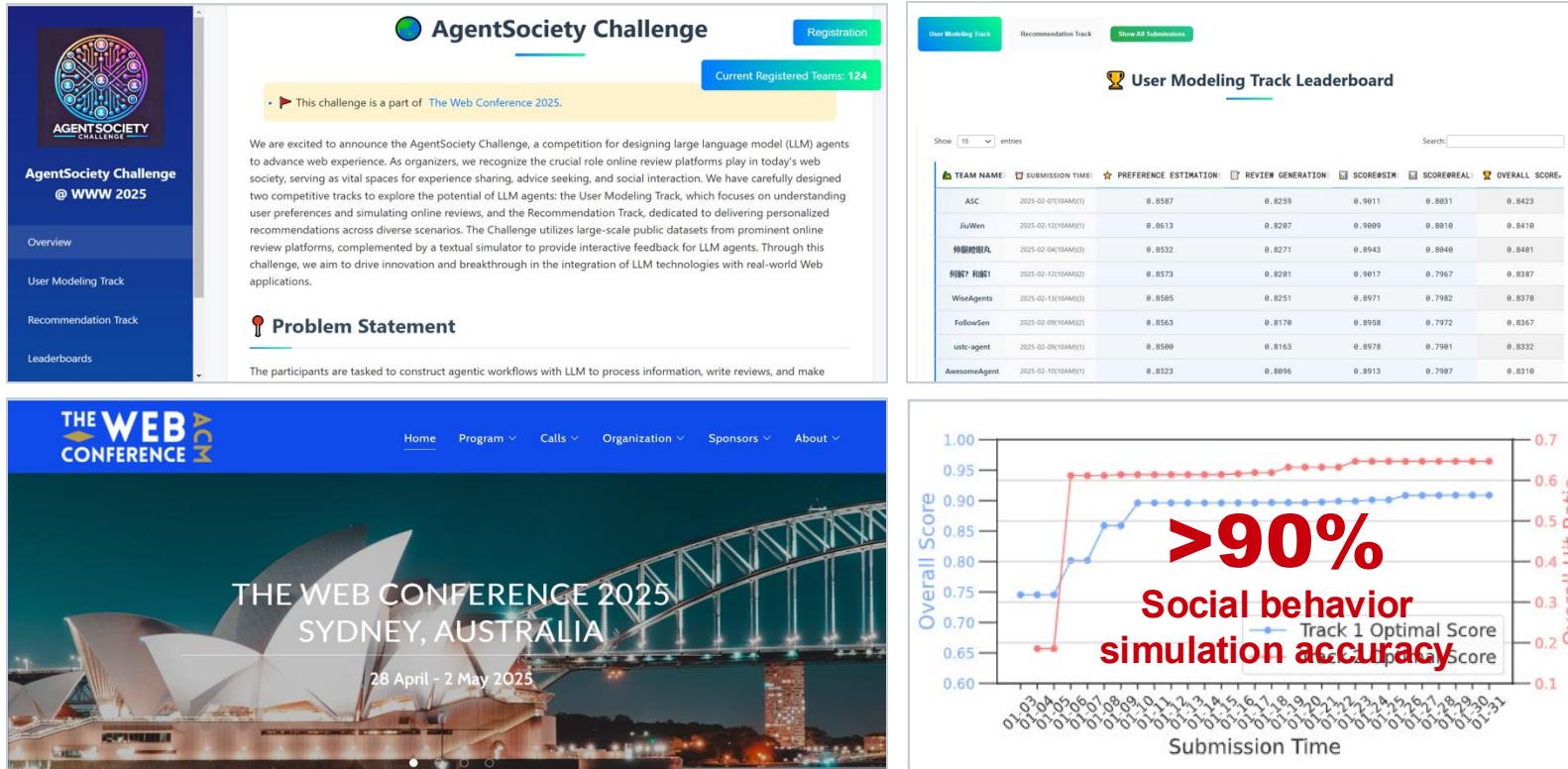


AgentSociety Challenge

We organize the first LLM agent challenge in WebConf.

295 top research teams, **1400** valid submissions, social behavior simulation accuracy up to **90%+**

Web Link: <https://tsinghua-fib-lab.github.io/AgentSocietyChallenge/workshop/index.html>



295 top research teams participate:

- Berkeley, Stanford, UChicago, NUS, UNSW, HKU, HKUST...
- THU, USTC, RUC, BUAA, FDU, ...
- Huawei, Honor, JD, Telecom, vivo...

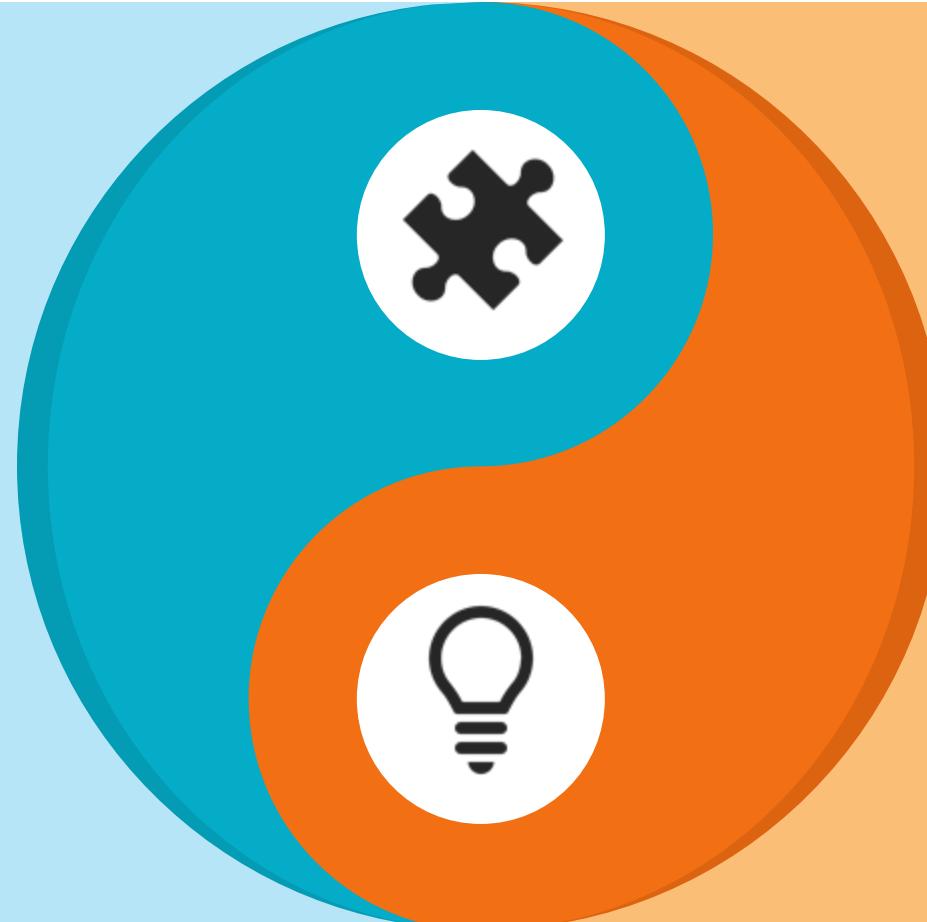
Yan, Yuwei, et al. "AgentSociety Challenge: Designing LLM Agents for User Modeling and Recommendation on Web Platforms." *WebConf 2025*.

协同高效推理

- ✓ 提升端侧推理能力
- ✓ 降低端侧推理时延
- ✓ 降低大模型调用成本
- 保护端侧数据隐私
- 提升个性化能力
- “通用-专用” 协同

自主演进框架

- ✓ 标准化设计框架
- ✓ 自动发现最优设计
- 搜索效率持续提升
- 面向新任务的冷启动
- 任务优化到连续优化



AI智能体，不只是一个模型。

感谢聆听！

fenglixu@tsinghua.edu.cn

