ZOMI

大模型系列之智能体

AI Agent 规划手段

关于大模型系列

- **内容背景**
  - **LLM + AI Agent**：大模型遇到智能体
- **具体内容**

1. **AI Agent 组成介绍**：LLM + 记忆 + 规划 + 工具
2. **AI Agent 规划手段**： Task Decomposition 与 Self Reflection
3. **AI Agent 热门应用**：交互式 Agent、自动化 Agent 与多模态 Agent
4. **AI Agent 问题与挑战**： Agent 的问题、Agent 的局限性

大模型业务全流程

数据 & 模型算法

2. 数据处理

- 开源数据
- 数据预处理
- 向量数据库

3. 模型算法

- LLM模型架构
- 多模态一切皆Tokens

模型训练 & 微调

4. 模型训练

- 混合精度
- 梯度检查
- 梯度累积
- …
- 5. 分布式并行
- 训练集群稳定性

6. 模型微调

- 全参微调
- 低参微调
- 指令微调

模型验证 & 推理部署

7. 模型验证

- 下游任务
- 测评标准
- benchmark

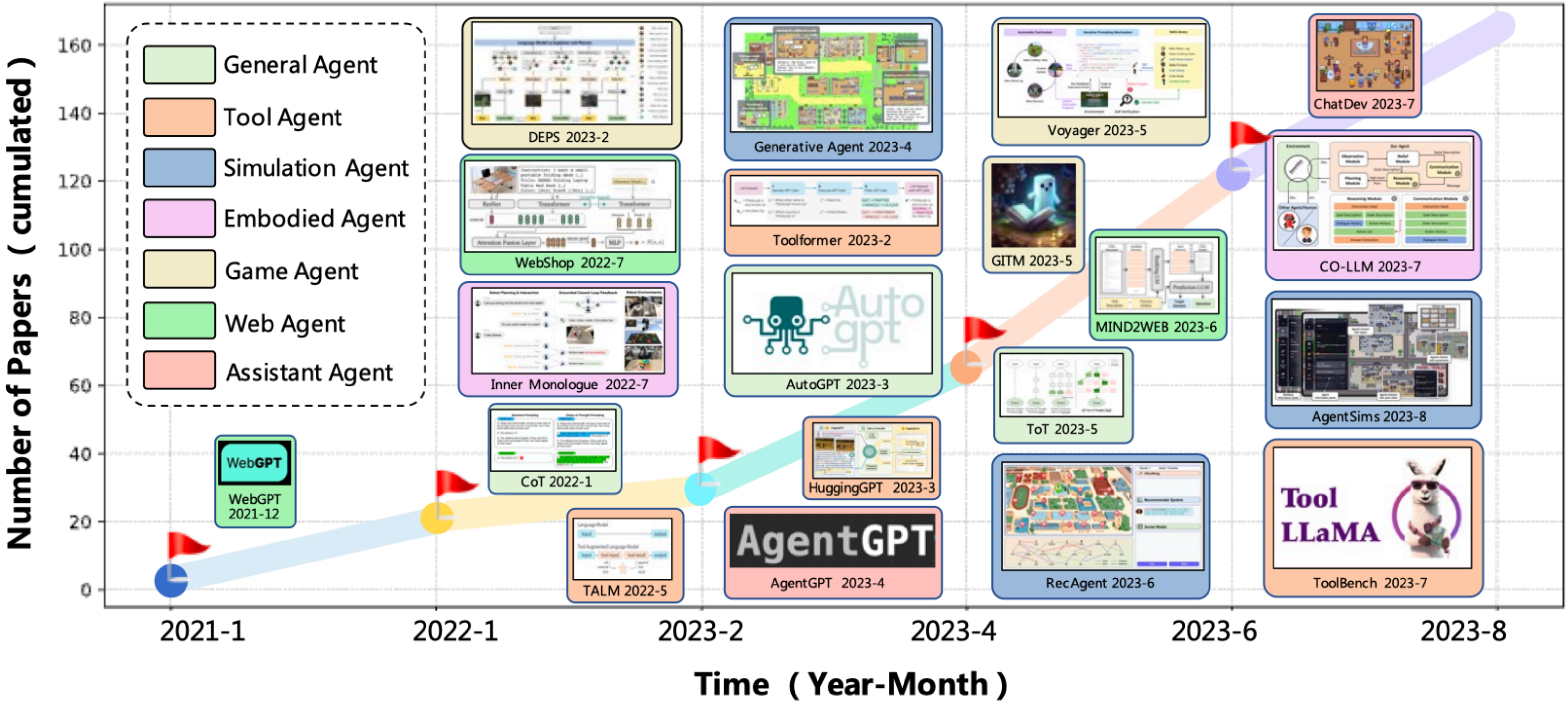
8. 推理与智能体

- 量化压缩
- 推理加速
- 9. Agent 智能体

1. AI 集群建设

- 计算、存储、网络
- AI 集群机房建设
- AI 集群上线与运维

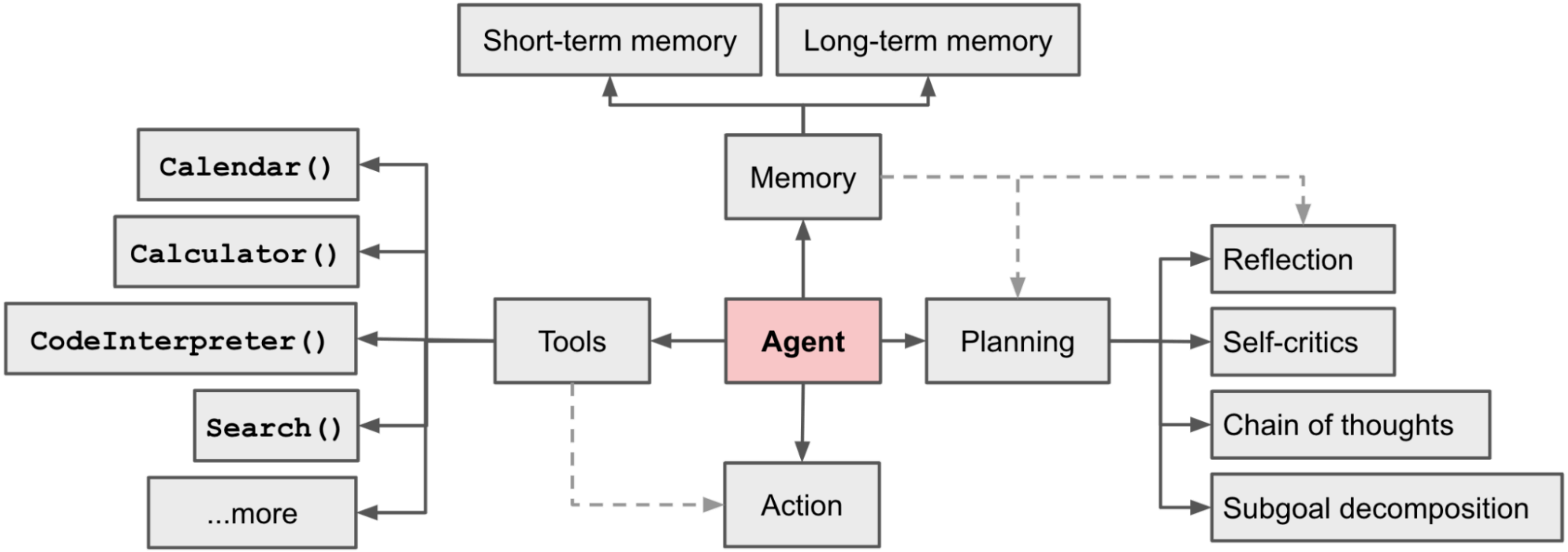
集群算力准备



Number of Papers (cumulated)
160
140
120
100
80
60
40
20
0
General Agent
Tool Agent
Simulation Agent
Embodied Agent
Game Agent
Web Agent
Assistant Agent
WebGPT
WebGPT 2021-12
DEPS 2023-2
WebShop 2022-7
Inner Monologue 2022-7
CoT 2022-1
TALM 2022-5
Generative Agent 2023-4
Toolformer 2023-2
AutoGPT 2023-3
HuggingGPT 2023-3
AgentGPT
AgentGPT 2023-4
Voyager 2023-5
GITM 2023-5
MIND2WEB 2023-6
ToT 2023-5
RecAgent 2023-6
ChatDev 2023-7
CO-LLM 2023-7
AgentSims 2023-8
Tool LLaMA
ToolBench 2023-7
2021-1
2022-1
2023-2
2023-4
2023-6
2023-8
Time (Year-Month)

关键组成

- 规划 Planning + 记忆 Memory + 工具 Tools



规划 Planning

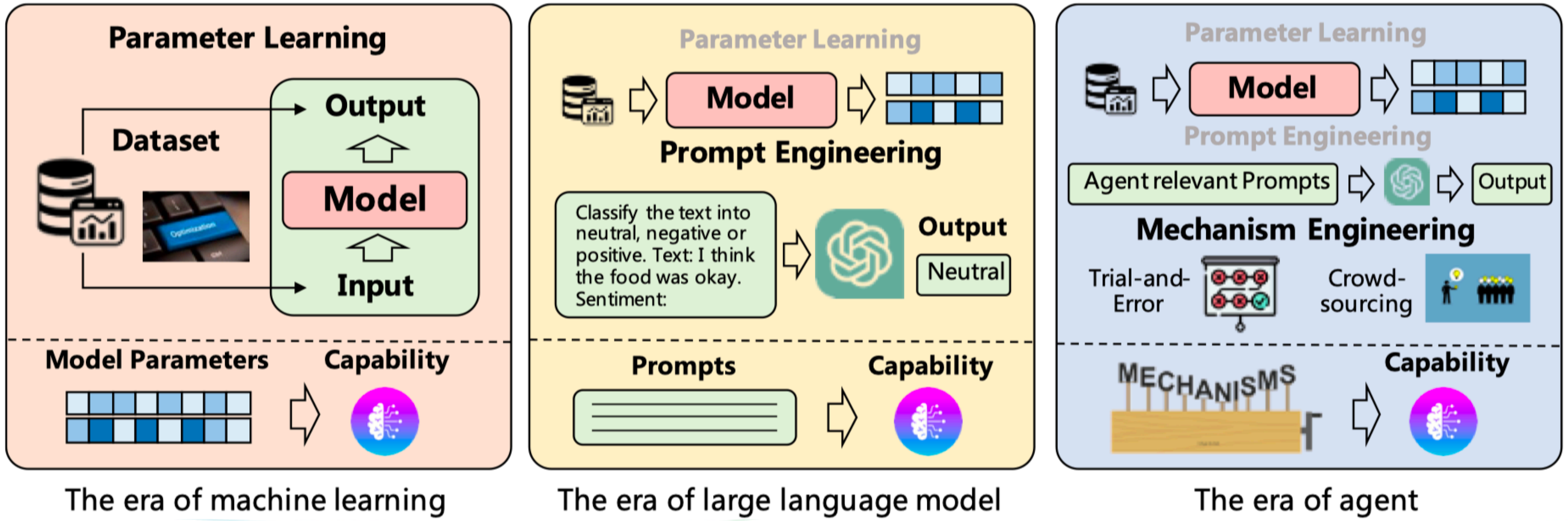
- **规划**：一项复杂任务通常包括多个子步骤，Agent 需要提前将一项任务分解为多个子任务。
  - **子目标与分解（Subgoal and decomposition）**：Agent 将复杂任务分解为更小、更易于处理的子目标，从而实现对复杂任务的高效处理。
  - **反思与完善（Reflection and refinement）**：Agent 可以对历史的动作进行自我批评和自我反思，从错误中吸取教训，并为未来的步骤进行改进，从而提高最终结果的质量。

- **实现**：通过prompt engine来引导 LLM 实现规划（即步骤分解）。

1. Agent & Prompt

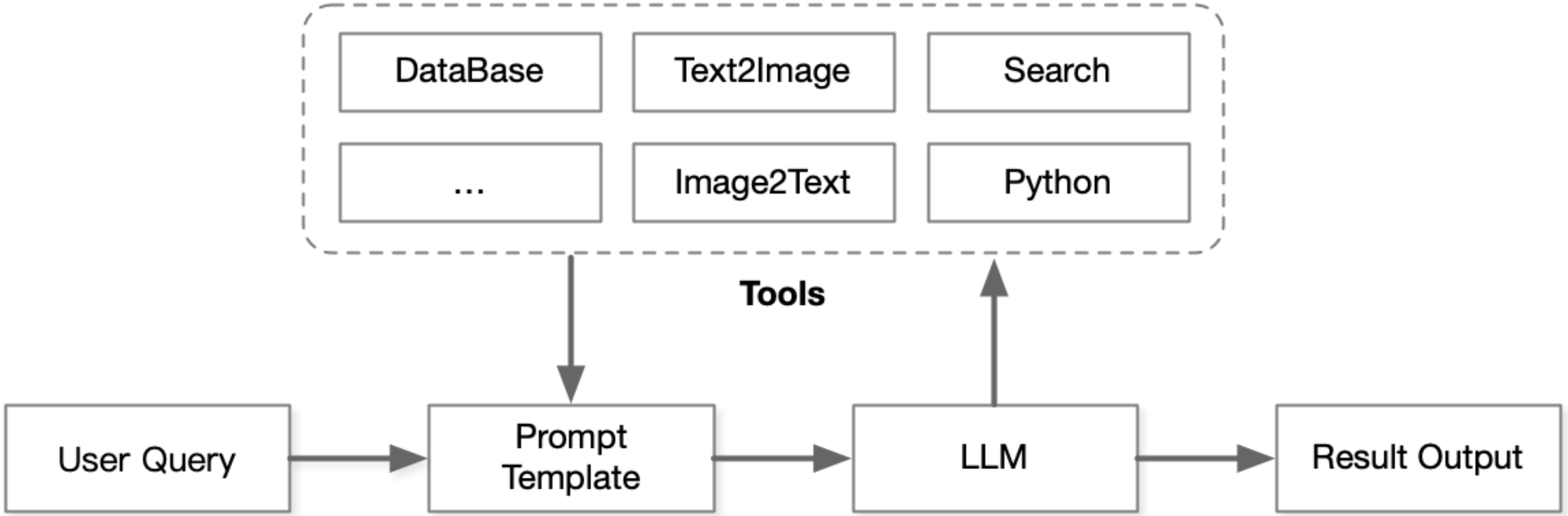
Agent 智能体与 Prompt 提示词的关系

- AI Agent 绝大多数创新点还是在 Prompt 层面，即：通过更好的 Prompt 提示词来激发 LLM 模型的能力。通过 Prompt 提示词，让 LLM 大模型仿照 Prompt 给出的方式来执行的一种应用范式。
- Prompt 里面包含关于 Tools 的描述，最后 AI Agent 智能体就可以根据模型的输出使用外部 Tools（例如计算器，搜索API，数据库，程序接口，各种模型的API） 能使用外部 API 或者知识库。



Agent 智能体与 Prompt 提示词的关系

- 图中展示了一个包含 AI Agents 智能体系统，通过 Tools让 LLM 大模型获取外部输入，通过 Prompts 来驱动 LLM 大模型得到用户想要的输出。



- 一项复杂的任务通常涉及许多步骤
- Agent 必须了解任务是什么并提前进行规划

Prompt 提示词

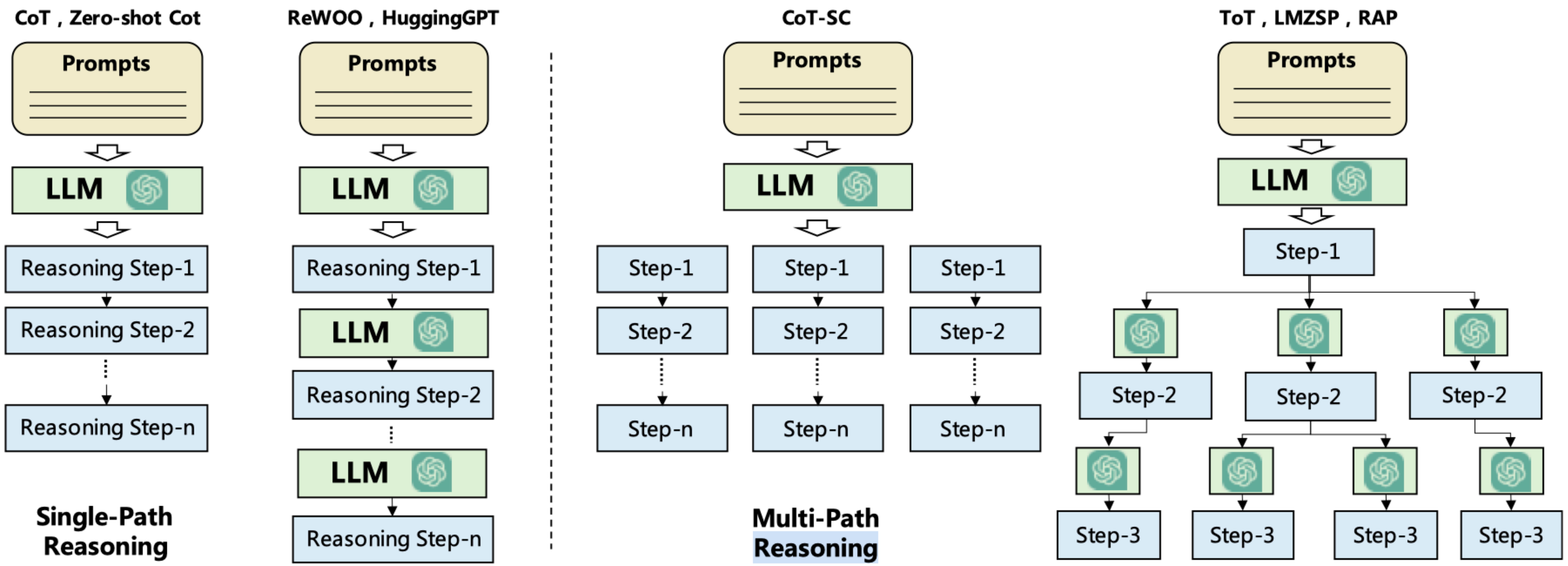


Figure 3: Comparison between the strategies of single-path and multi-path reasoning. LMZSP represents the model proposed in [70].

2. 规划 Planning

—— 任务分解

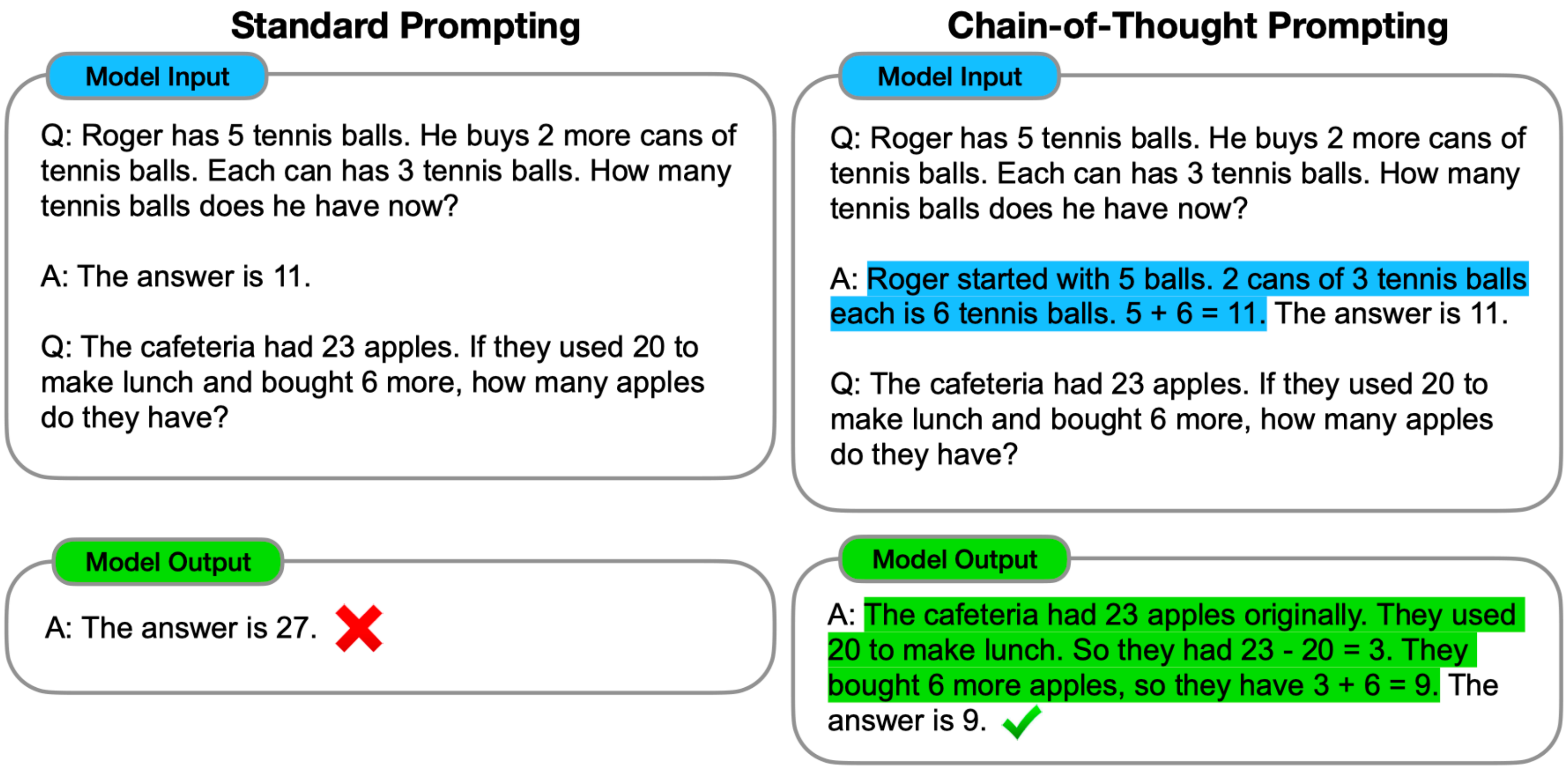
任务分解1：思维链 CoT

- **思维链（Chain-of-thought，CoT）**一种改进的提示策略，用于提高 LLM 在复杂推理任务中的性能，如算术、常识和符号推理。**CoT 已经成为增强复杂任务上模型性能的标准提示技术。**
- 实现过程中，模型被指示「一步一步思考」，从而将困难任务分解为更小、更简单的步骤。在 AI Agent 中 CoT 将大型任务转化为多个可管理的小任务，并解释清楚模型的思维过程。

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

任务分解1：思维链 CoT

- 相比于上下文学习，思维链多了中间的中间的推导提示：



Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

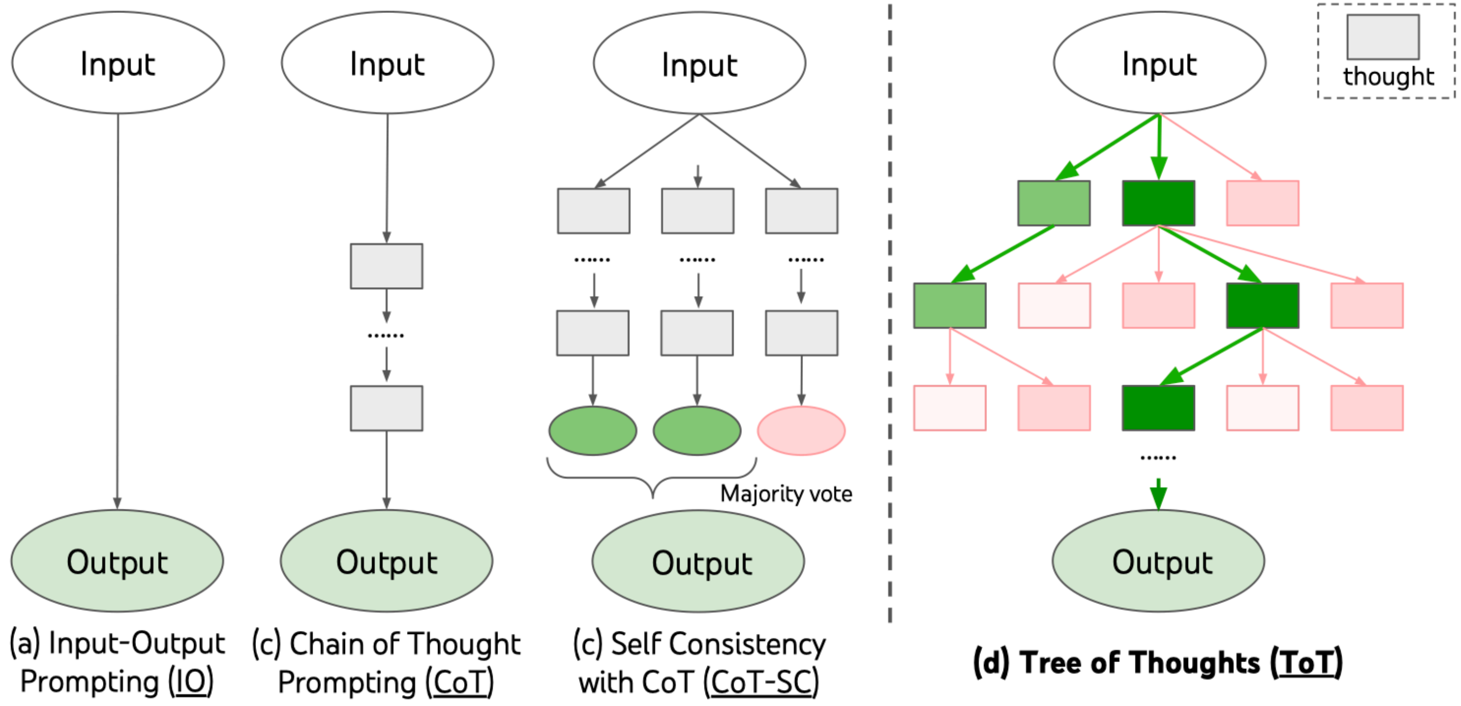
任务分解II：思维树 ToT

- 思维树（Tree of Thoughts，ToT），通过同时考虑多个可能的 Plan，并利用价值反馈机制进行决策，扩展了现有的规划方法。**即通过每一步探索多种推理可能性来扩展 CoT。**
- 首先将问题分解为多个思考步骤，并在每个步骤中生成多个思考，创建一种树结构。搜索过程可以是广度优先搜索（BFS）或深度优先搜索（DFS），其中每个状态由分类器或多数 vote 进行评估。具体任务分解过程可以通过以下三种方式完成：
  1. 基于 LLM 提示，如「XXX 的步骤是什么？」、「实现 XXX 的子目标是什么？」
  2. 使用特定于任务的提示，比如「写一个XXX故事大纲」
  3. 人工输入。

Tree of Thoughts: Deliberate Problem Solving with Large Language Models

任务分解II：思维树 ToT

- 思维树（Tree of Thoughts，ToT），通过同时考虑多个潜在的可行计划，并利用价值反馈机制进行决策，扩展了现有的规划方法。即通过每一步探索多种推理可能性来扩展 CoT。



Tree of Thoughts: Deliberate Problem Solving with Large Language Models

3. 规划 Planning

—— 自我反省

自我反思 Self Reflection

- 自我反思（Self-reflection）允许 Agent 通过完善以往行动决策和纠正以往错误来迭代改进。
  1. **ReAct**：通过将动作空间扩展为一个任务特定的「离散动作和语言空间的组合」；
  2. **Self-ask**：是一种follow-up的使用范式，仅仅包含follow-up, immediate answer步骤；

自我反思 I：ReAct

- **ReAct**：将动作空间扩展为任务特定的「离散动作和语言空间的组合」，将推理和动作集成在 LLM 中。离散动作使 LLM 能够与环境交互，而语言空间促使 LLM 以自然语言生成推理轨迹。
- Thought: ... Action: ... Observation: ... ... (Repeated many times)

```
(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the
program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in
October 2005 by Apple … originally designed to control the
Front Row media center program …

Thought 2: Apple Remote was originally designed to control
the Front Row media center program. I need to search Front
Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row
Seat to Earth', 'Front Row Motorsports',' Front Row
(software)', …]

Thought 3: Front Row is not found. I need to search Front
Row (software) .
Act 3: Search[Front Row (software) ]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple
Remote or the keyboard function keys. So the answer is
keyboard function keys.
Act 4: Finish[keyboard function keys ]
```

```
(2b) ReAct (Reason + Act)

Act 1: Think[First I need to find a pepper shaker...more
likely to appear in cabinets (1-6), countertops (1-3), ... ]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.
(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)
Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1,
...... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

Act 8: Think[Now I find a pepper shaker 1. Next, I need to
put it in/on drawer 1. ]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.
```

自我反思 I：ReAct

- Thought: ... Action: ... Observation: ... ... (Repeated many times)

知识密集型任务（HotpotQA、FEVER）

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.
Act 1: Search[Apple Remote]
Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple … originally designed to control the Front Row media center program …

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.
Act 2: Search[Front Row]
Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports',' Front Row (software) ', …]

Thought 3: Front Row is not found. I need to search Front Row (software) .
Act 3: Search[Front Row (software)]
Obs 3: Front Row is a discontinued media center software …

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.
Act 4: Finish[keyboard function keys]

决策型任务（AlfWorld Env、WebShop）

(2b) ReAct (Reason + Act)

Act 1: Think[First I need to find a pepper shaker...more likely to appear in cabinets (1-6), countertops (1-3), ...]
Act 2: Go to cabinet 1
Obs 2: On the cabinet 1, you see a vase 2.

(Here the agent go to cabinet 1, then cabinet 2, 3, then countertop 1 and 2)

Act 6: Go to countertop 3
Obs 6: On the countertop 3, you see a apple 1, a bread 1, ... a pepper shaker 1, and a vase 3.

Act 7: Take pepper shaker 1 from countertop 3
Obs 7: You pick up the pepper shaker 1 from the countertop 3.

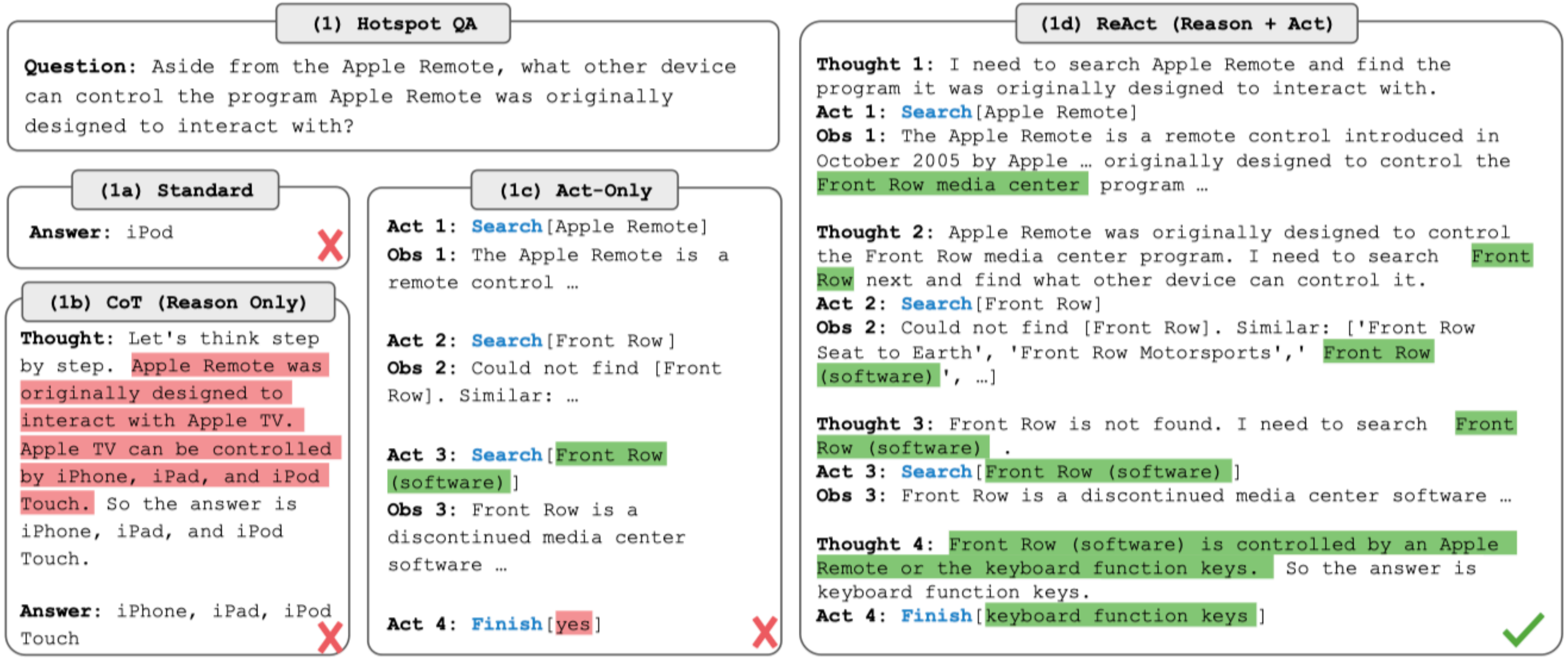
Act 8: Think[Now I find a pepper shaker 1. Next, I need to put it in/on drawer 1.]
Act 9: Go to drawer 1
Obs 9: Drawer 1 is closed.

Act 10: Open drawer 1
Obs 10: You open Drawer 1 …

Act 11: Put pepper shaker 1 in/on drawer 1
Obs 11: You put pepper shaker 1 in/on the drawer 1.

自我反思 I：ReAct

- ReAct本质融合 Reasoning 和 Acting 的一种 Prompt 范式，推理过程是浅显易懂，仅仅包含thought-action-observation步骤，很容易判断推理的过程的正确性。四种 Prompt 比较：
  - （1）Native Prompt；（2）Chain-of-thought；（3）Acting Only ；（4）ReAct 。

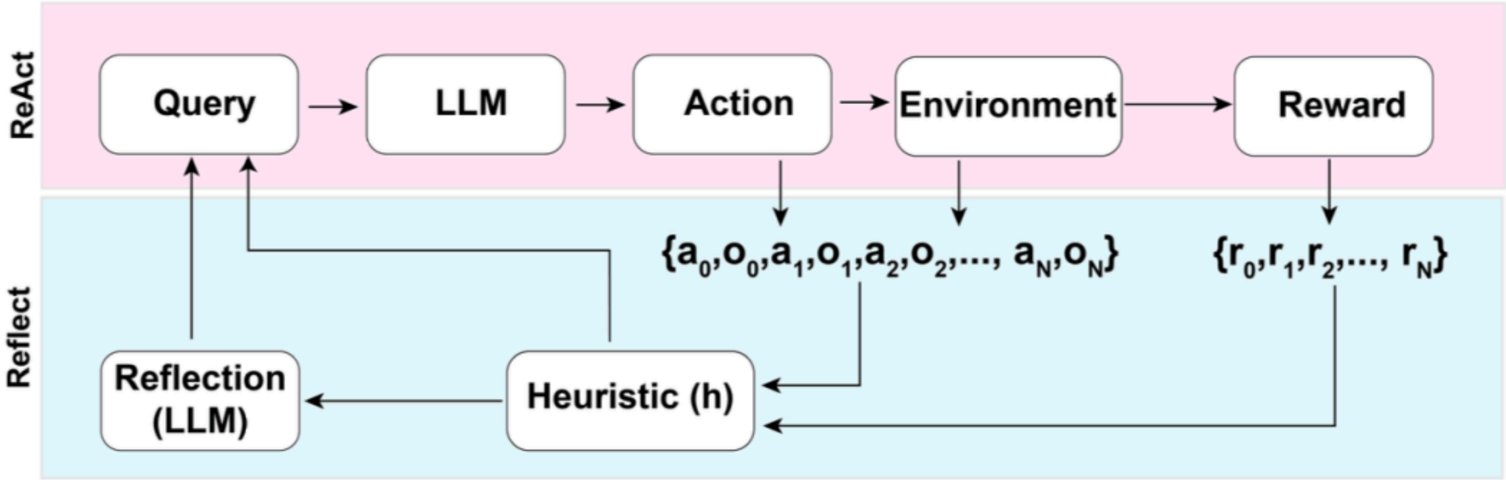


自我反思 I：ReAct vs CoT

- Chain-of-Thought 推理只是作为静态黑盒，它没有用外部知识，所以在推理过程中会出现事实幻想（fact hallucination）和错误传递（error propagation）的问题。
- ReAct 克服了在思维链推理中的幻觉和错误传播问题，通过与简单的维基百科API交互，生成类似于人的任务解决型轨迹，解释性进一步增强。

自我反思 I： ReAct 的扩展 Reflexion 框架

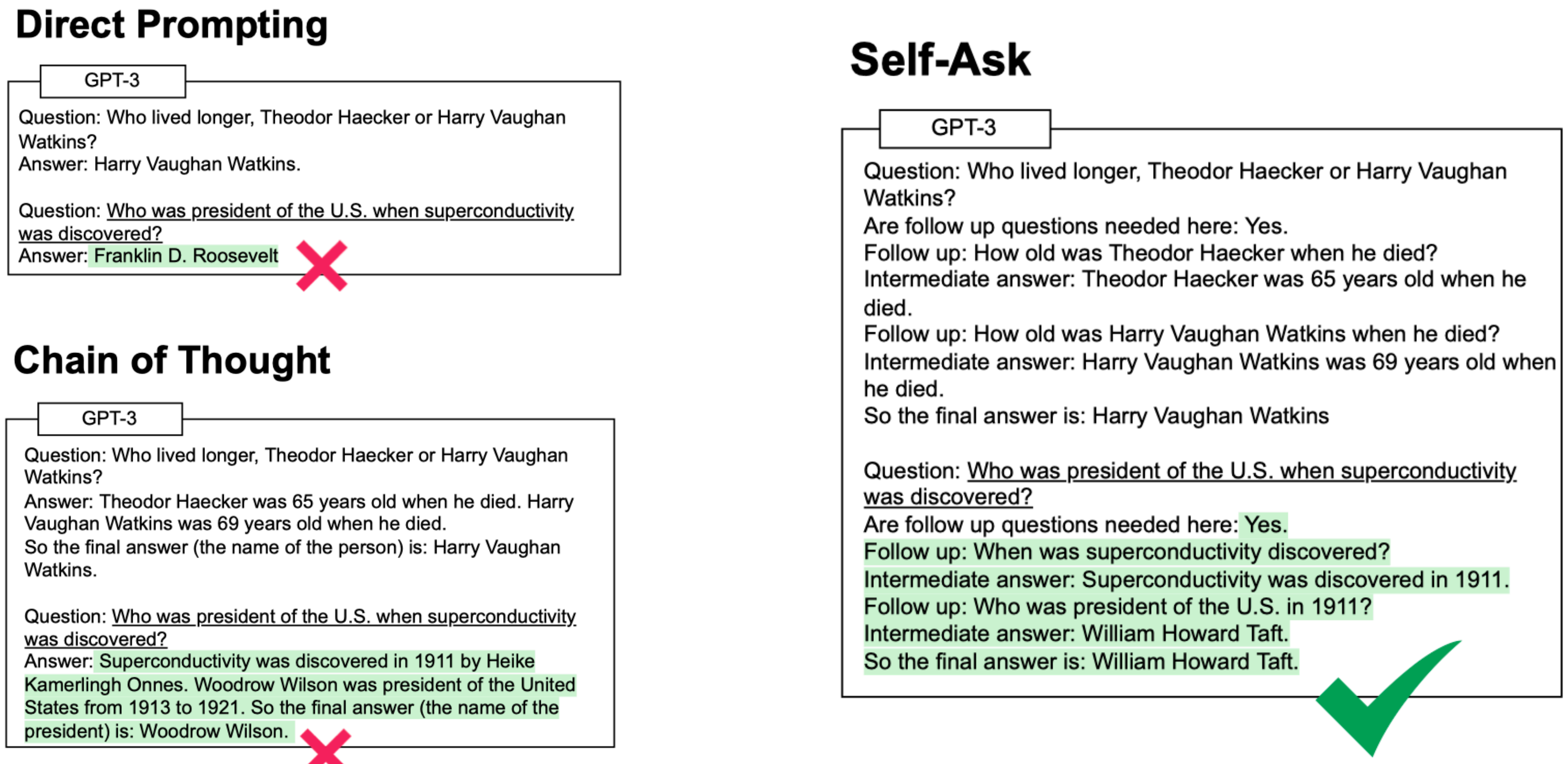
- Reflexion 框架则为 Agent 配备了动态记忆和自我反思能力，提高 Reasoning 能力。通过借鉴 RL 流程，奖励模型提供简单的二元奖励，特定于任务的动作空间通过 LLM 进行增强，实现复杂推理步骤，动作空间遵循 ReAct 中的配置（ Thought... Action ... Observation ）；在每个动作 a_t 之后，智能体计算启发式 h_t，并选择性地根据自我反思结果来决定重置环境，从而开始新的试验。



Reflexion: Language Agents with Verbal Reinforcement Learning

自我反思 II：Self-ask

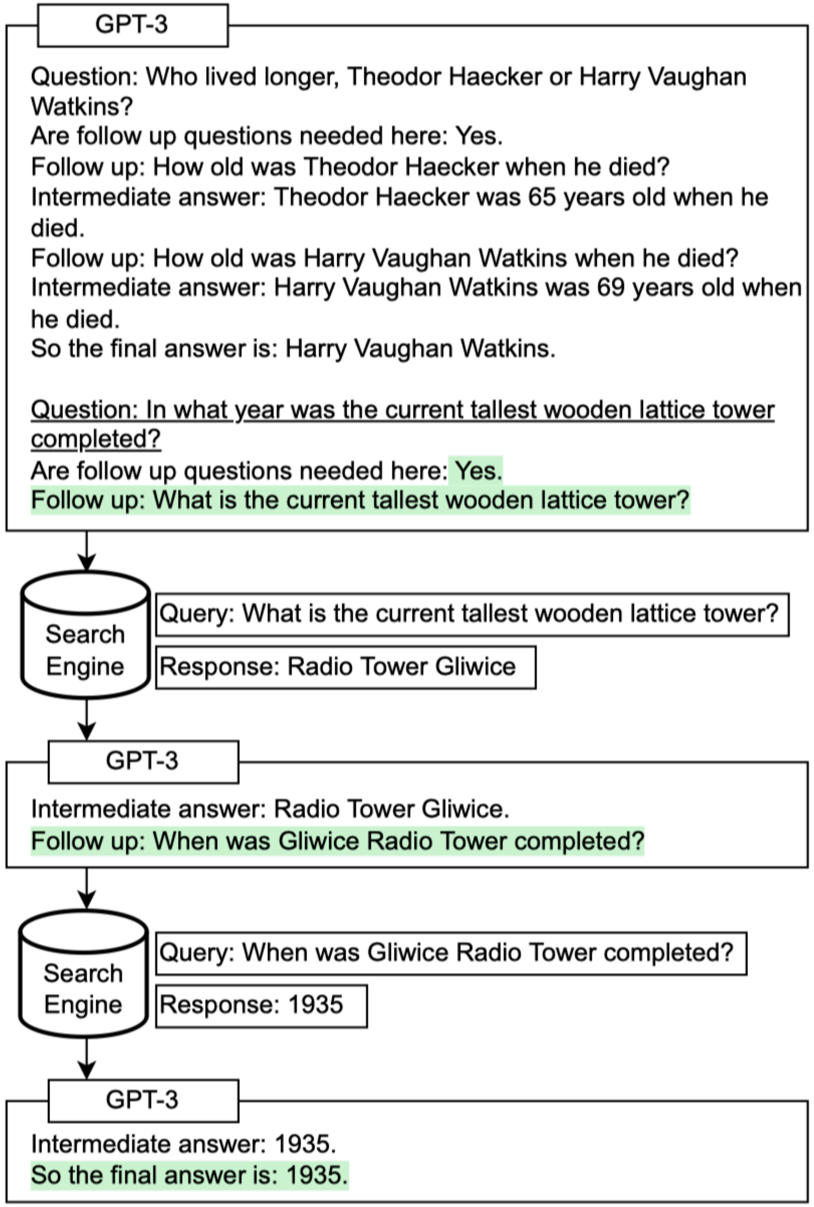
- Self-ask是一种follow-up的使用范式，仅仅包含 follow-up, immediate answer步骤，至于 follow-up 多少个 step，完全由 Agent 自己决定。



白色背景为 prompt，绿色背景为 LLM 输出，下划线为 inference-time；Self-ask 需要一个/少量 Prompt 来引导 LLM 如何回答 Prompt 问题。

自我反思 II：Self-ask

- 前面的是 inference-time question，Prompt 末尾插入短语 “Are follow up questions needed here:”，可以略微改善结果。
- LLM 输出一个 Response：Yes，这意味着后续行动问题是必要的。
- 然后 LLM 输出第一个 follow-up 问题，使用搜索引擎找到答案，作为 Prompt 继续 follow-up 问题，直到它决定有足够的信息为止。
- 最终引导 LLM 输出：“So the final answer is:”，使得最终答案可以很容易根据 Self ask 解析出来。



Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem