Prompting Guide – 进阶篇

伊辉

目标

“Prompt”（提示）是给预训练的生成式模型（AIGC）的输入文本，它的目的是指导模型生成特定的输出。在使用AIGC生成性式模型时，Prompt充当了人与机器间交流的媒介，提出问题、设置任务或引导对话。

随着当前人工智能技术的不断普及与进步，Prompt Engineering成为最直观的方式，让算法人员、开发人员以及非技术用户能够访问和利用AI的强大能力，帮助人类与AI更加紧密和高效地协作。

Prompt是现代AI应用的基石，它的发展和研究将会深刻影响我们使用和与AI交互的方式。随着模型的进步，Prompt Engineering将会变得更加复杂和精细化，它的重要性随时间不断增长。

课程目标：

1、通过构建有效的Prompt，更有效的与大语言模型沟通

2、使用Reasoning策略，协调大语言模型不同角色交互迭代工作，解决复杂问题

3、通过Prompt Engineering，提供个性化与专业化的服务

4、使用Prompt Engineering进行AI助手（Copilot）及AI智能体（Agent）的开发

目标受众：有初步的Prompt编写经验，对ChatGPT、文心一言等对话助手有了解与实际应用；有大语言模型、NLP相关基础（可选）。

课程大纲

LONGFOR 龙湖

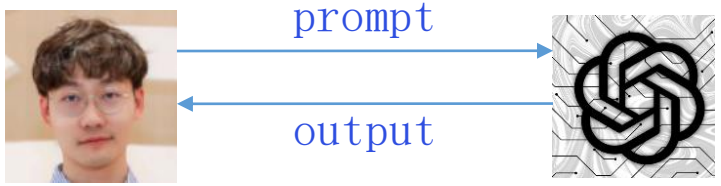
1. 基础概念

理解Prompt Engineering的相关基础概念

LLM | Transformer | Token | In-Context Learning

2. 入门篇

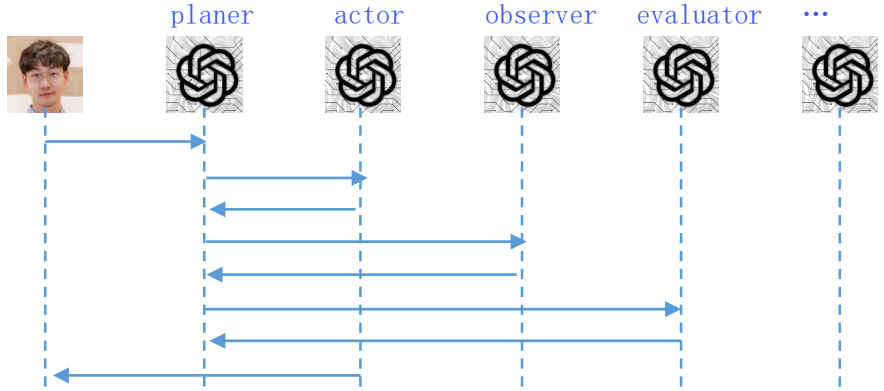
构造单个有效的Prompt，与大语言模型进行有效沟通



3. 进阶篇

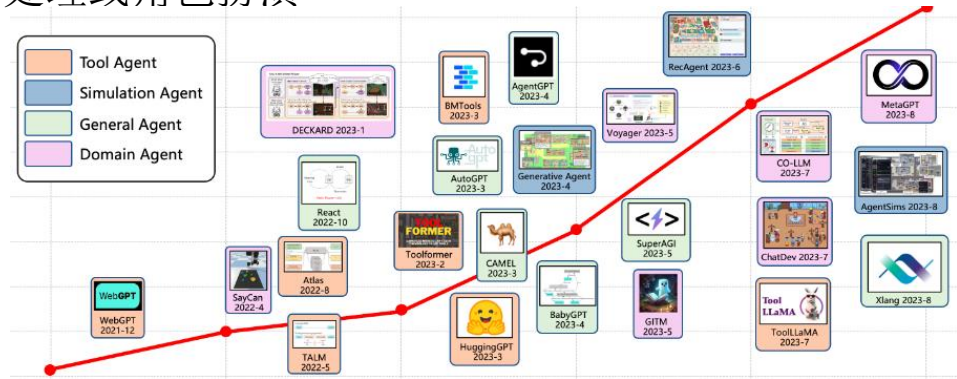
本次课程

协调大语言模型迭代工作，提升结果质量或解决复杂问题



4. 智能体篇

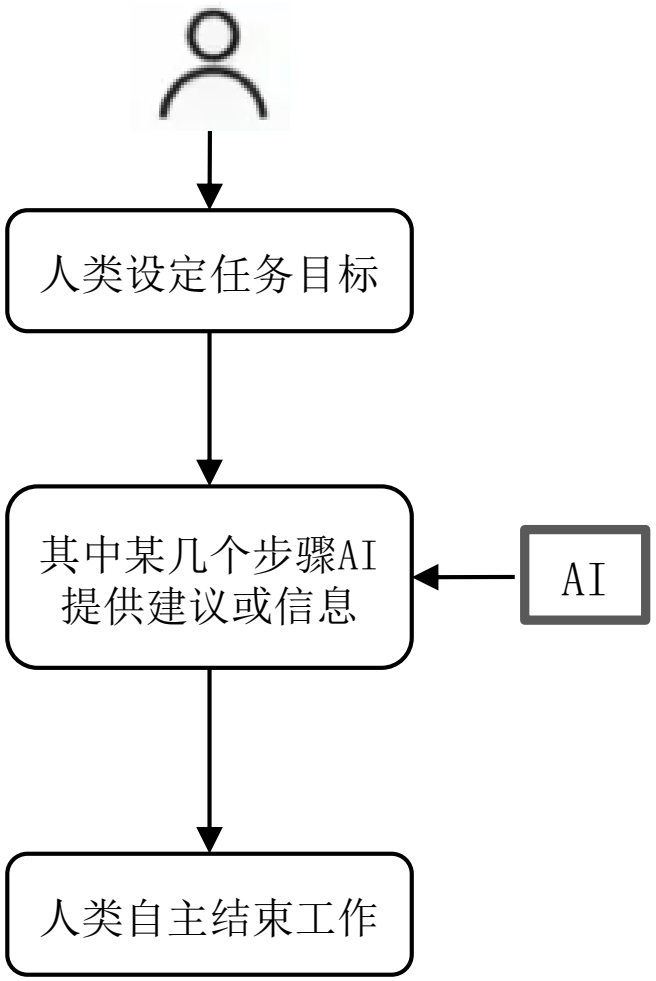
利用Prompt Engineering构建AI智能体，进行专业化任务处理或角色扮演



LONGFOR 龙湖

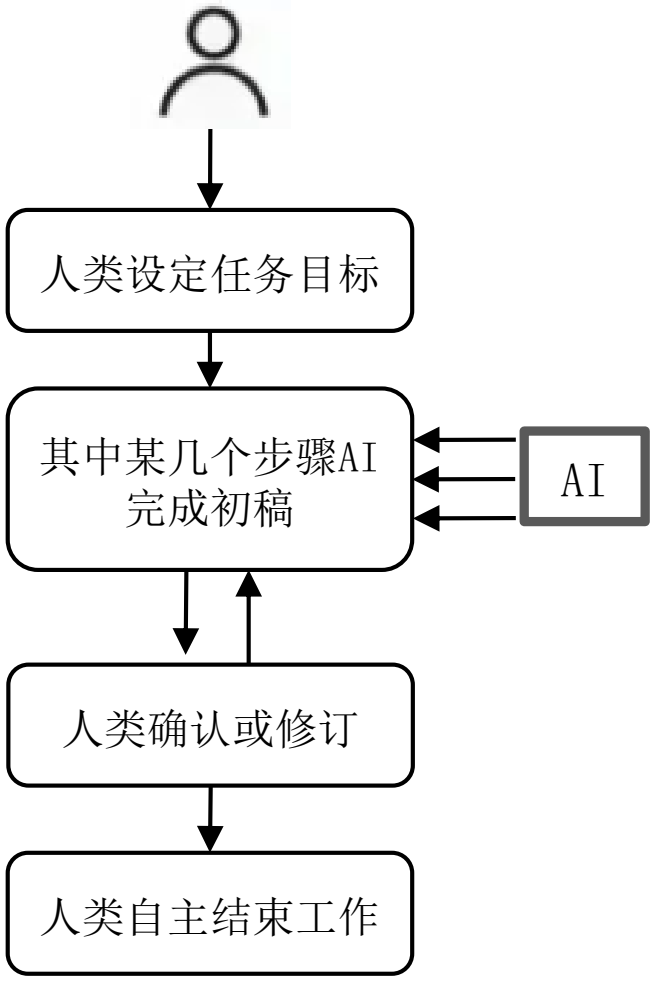
AI Chat

人类完成大部分任务



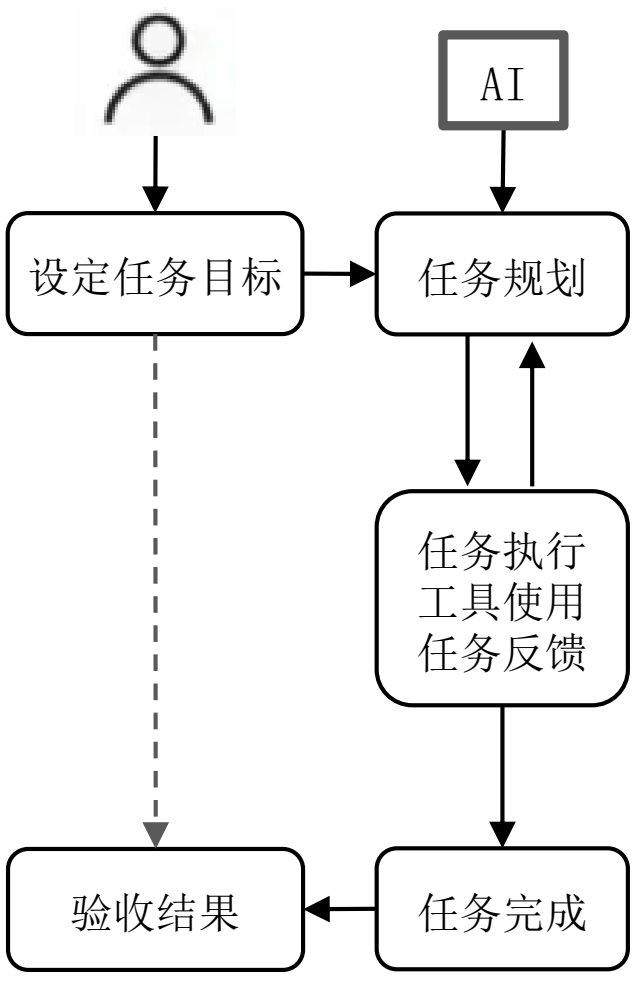
AI Copilot

AI辅助人类协同完成任务



AI Agent

AI主动思考与行动，完成绝大部分任务



1、Prompt入门篇（复习）

LONGFOR 龙湖

- 构建有效Prompt的前提
- Prompt任务目标的明确
- Prompt的语境设置
- 少样本学习的威力
- 理解上下文学习
- 思维链入门
- 结构化提示
- 伪代码提示
- 长度外推与大海捞针
- 无关内容的影响
- Prompt的自动评估
- Prompt的自动优化

AI Agent要解决的是什么问题?

LONGFOR 龙湖

李飞飞

我认为我们必须小心代替工作（job）和代替任务（task）的区别。每一个人类工作（job），实际上是涉及到多个任务（task）的集合。比如我研究医疗保健，一个护士 8 小时的工作包含数百个任务。我确实看到人工智能代理在许多任务（task）上提供帮助、具有辅助和增强功能，但在讨论工作（job）时我们应该非常小心。

……

吴恩达

我想回到任务（task）的话题，因为我认为这很重要。

我的团队为许多企业工作，有时我会接到一位CEO的电话。他们说：「嘿，我在阅读关于人工智能代理的文章，我该怎么办？」

我们的朋友Avery Browne经常给出的建议，就是观察你的团队，找出你所有员工实际在做哪些任务（task），并分析不是在工作（job）层面而是在任务（task）层面上，这些任务对于人工智能增强或自动化来说有多容易，并且其业务回报率是多少。每次我在企业中进行这项工作时，我们总是想出非常多的想法。因此，人工智能增强或自动化有很多机会。

第二件事是，往往具有最高回报率的任务（task）并不是人们最初认为的那样。例如，当你想到放射科医生时，人们往往认为，哦，放射科医生要看 X 光片。这是你脑海中有关于这份工作（job）的认知，但实际上我们将这份工作（job）分解为许多不同的任务（task），比如收集患者病史之类的事情，结果可能更容易，而且回报率更高。

……

中文语料库： COIG-CQIA: Quality is All You Need for Chinese Instruction Fine-tuning

LONGFOR 龙湖

中科院为了搞出更适合中文的 AI，搜集了各大社区平台的语料，精心整理出一份数据集。

他们拿这些数据去对大语言模型（Yi-34B）做Fine-tuning，然后测了下各家数据炼出来的性能：

Source	Open QA	Brainstorming	Classification	Generation	Summarization	Rewrite	Closed QA	Extract	Math	Code	Average
SegmentFault	51.3	70.7	43.8	66.8	57.1	75.7	41.4	47.1	75.5	65.0	60.7
COIG PC	19.1	38.0	27.8	43.9	37.8	63.7	40.6	25.8	43.6	19.1	37.2
Douban	57.4	81.7	63.6	76.2	54.7	60.1	47.5	50.4	73.8	50.6	63.2
Zhihu	66.5	90.4	52.6	82.6	71.2	78.2	46.4	39.6	**76.0**	62.2	69.1
Logi QA	51.4	76.4	64.9	75.6	60.0	71.4	61.6	52.7	47.0	45.3	62.0
Ruozhiba	**75.9**	**92.3**	**76.5**	**92.1**	**77.3**	70.9	**67.2**	**68.5**	72.6	**65.2**	**76.9**
Wiki	63.0	75.7	44.0	80.6	47.9	66.6	47.9	50.0	56.8	55.6	60.5
Finance	46.8	71.1	17.1	60.1	27.4	23.6	17.2	29.4	28.5	24.8	36.7
Exam	49.4	79.7	64.7	79.9	61.5	**79.8**	66.2	61.0	52.8	56.3	66.2
Xhs	51.3	76.1	38.5	68.0	25.8	46.0	28.4	32.1	74.6	36.3	50.3
Wikihow	54.7	75.2	32.1	68.2	45.3	55.9	40.9	55.8	41.0	44.4	52.7
CQIA-Subset	56.2	84.5	48.1	72.9	60.5	70.9	54.6	50.8	52.5	49.5	61.9

第一名：弱智吧，76.9分。10项指标中，问答、脑暴、分类、总结等8项排在首位，数学任务排在第3位，润色任务排在第5位。

第二名：知乎，69.1分

第八名：小红书，50.3分

Table 3: The performance of Yi-34B trained on various datasets evaluated on BELLE-EVAL using GPT4.

Ruozhiba is a sub-forum of Baidu Tieba, an interests-based community forum. Its posts often contain puns, polysemous terms, causal reversals, and homophones, many of which are designed with logical traps, posing challenges even for humans…We conjecture this is because it may enhance the model's logic reasoning ability, thereby benefiting most of the instruct-following tasks.

弱智吧是百度贴吧的一个子论坛，是一个基于兴趣的社区论坛。帖子经常包含双关语、多义词、因果颠倒和同音词，其中许多设计都有逻辑陷阱，甚至对人类也构成挑战……我们推测这是因为它可以增强模型的逻辑推理能力，从而有利于大多数指令跟踪任务。

《COIG-CQIA: Quality is All You Need for Chinese Instruction Fine-tuning》，中科院

- 执行死刑时本人不去，委托律师去可以吗？根据民法典规定当事人可以委托一至二人作为自己的代理人啊
- 冥婚算红事还是白事
- 据数据表明，异性间的离婚率远大于同性，而中国同性离婚率至今为0这是否说明同性间结婚比异性更有优势
- 我吃了狗拉的屎后拉出来的屎还是狗屎吗？
- 孩子得了小儿多动症，再得一个小儿麻痹症是不是就好了
- 鸡柳是鸡身上哪个部位啊？
- 手机软件有点良心的也只剩淘宝京东和拼多多了就它们没有跳转广告，不愧是大厂
- 坏人为什么不趁警察上班的时候作案？这样的话不就没空逮捕他们了吗？
- 过马路要看车，但是马路上没车怎么办？
- 爸爸再婚，我是不是就有了个新娘？
- 为什么我买的生鱼片是死鱼片？
- 明明是等绿灯，为什么叫做等红灯？
- 为什么灭火叫做救火？
- 有哪些违法行为是合法的
- 既然我们有联合国的一票否决权那为什么不能提出“不能用核弹轰炸美国”，然后一票否决，这样大家都能轰炸美国了
- 假如现在去做冻卵，那以后生的孩子算不算预制菜
- 我用的是晨光的笔，那么你们都是啥笔

《COIG-CQIA: Quality is All You Need for Chinese Instruction Fine-tuning》，中科院

2、Prompt进阶篇

- Agentic Workflow
- Least to Most
- Self-Consistency
- Self-Refine
- Tree of Thought
- Graph of Thought
- Reflexion
- Reasoning and Acting
- Lanuage Agent Tree Search

2、Prompt进阶篇

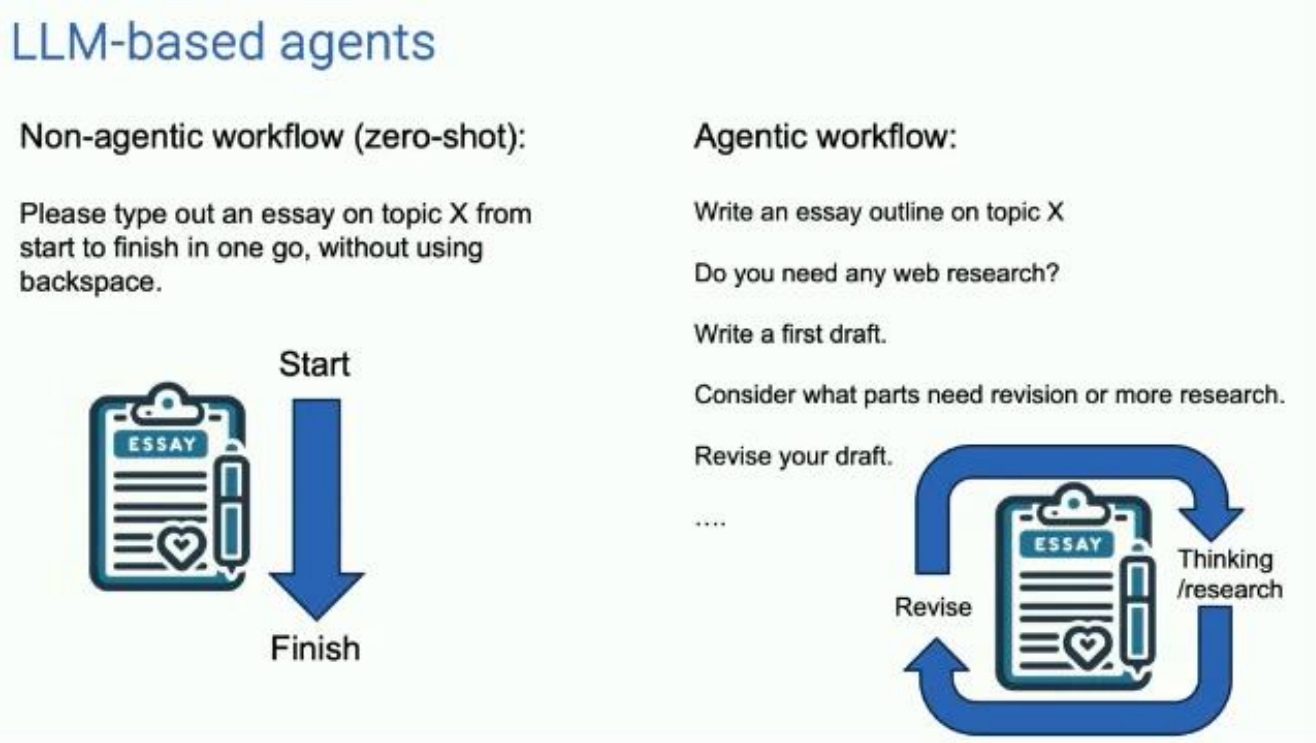
Path to AGI feels like a journey rather than a destination, but I think agentic workflow could help us take a small step forward on this very long journey.

—— by Andrew Ng

迈向人工通用智能之路，宛若一场旅程而非终点，然而我深信，Agentic Workflow能助我们在这条漫长道途上迈出微小而坚实的一步。

—— by Translator Agent （huiyi）

LLM-based Agents



Andrew：你跟LLM说，先写一个大纲，如果需要的话去网上查点资料，再写一个草稿，然后思考你的草稿该怎么改，最后再修改，多次如此迭代。很多人没有意识到这会带来多大的优化，事实是我经常这样做，得到的效果非常惊艳。

今年4月份，Andrew Ng（吴恩达）在红杉资本的人工智能峰会（AI Ascent）上，发表了《Agentic Reasoning》的演讲，提出了Agentic Workflow以及几种主要的模式。主要结论观点：

- 主动型工作流中的AI代理可以产生比传统工作流更好的效果。
- 多代理协作是提高AI性能的有效策略。
- AI代理的运用将扩大人工智能可执行任务的范围。
- ~~快速生成Token非常重要。在低质量的LLM上生成更多的Token可以获取好的效果。~~
- 人们需要耐心等待AI代理完成任务的方式。

《Agentic Reasoning》，Andrew Ng

Agentic Workflow Design Patterns

LONGFOR 龙湖

1. Reflection
2. Tool use

robust technology

3. Planning
4. Multi-agent collaboration

emerging technology

Reflection 反思，LLM 检查自己的工作，以提出改进方法

Tool Use 工具使用

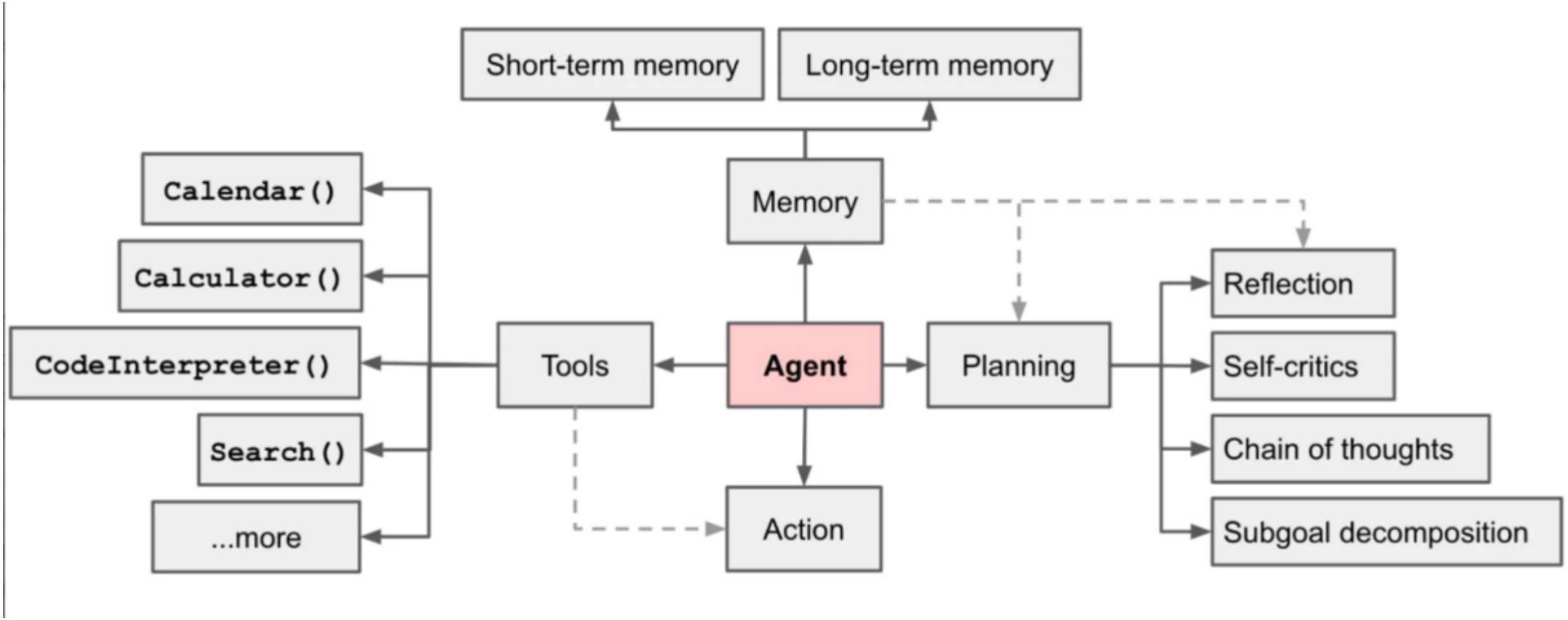
Planning 规划，LLM 提出并执行一个多步骤计划来实现目标

Multi-Agent 多个 AI 智能代理一起工作，分配任务并讨论和辩论想法，以提出比单个智能体更好的解决方案

《Agentic Reasoning》，Andrew Ng

LONGFOR 龙湖

AI Agent Overview



AI Agent = LLM + Planning + Action + Memory + Tools

Lilian Weng, OpenAI

推理（Reasoning）

LONGFOR 龙湖

提示生成
Instruction Generation

Zero-Shot CoT
Let's think step by step

Plan-and-Solve
Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.

APE
Let's work this out in a step by step way to be sure we have the right answer.

OPRO
Take a deep breath and work on the problem step by step.

示例生成
Examplar Generation

Manual CoT

Auto CoT

Active Prompt

推理执行
Reasoning

推理增强 (Reasoning Enhancement)

- Self-Consistency
- Self-Refine
- Tab-Cot
- Skeleton-of-Thought
- Tree-of-Thoughts
- Graph-of-Thought
- Program-of-Thought
- Recursion-of-Thought
- Algorithm-of-Thought
- Everything-of-Thought
- Meta-Prompting
- Least-to-Most

推理结构 (Cot Formulation)

推理验证 (Verification)

- RPM
- Self-Verification
- CRITIC
- External Knowledge（CoK）

任务规划 (Planing)

- ReAct
- Reflexion
- Chain of Hindsight

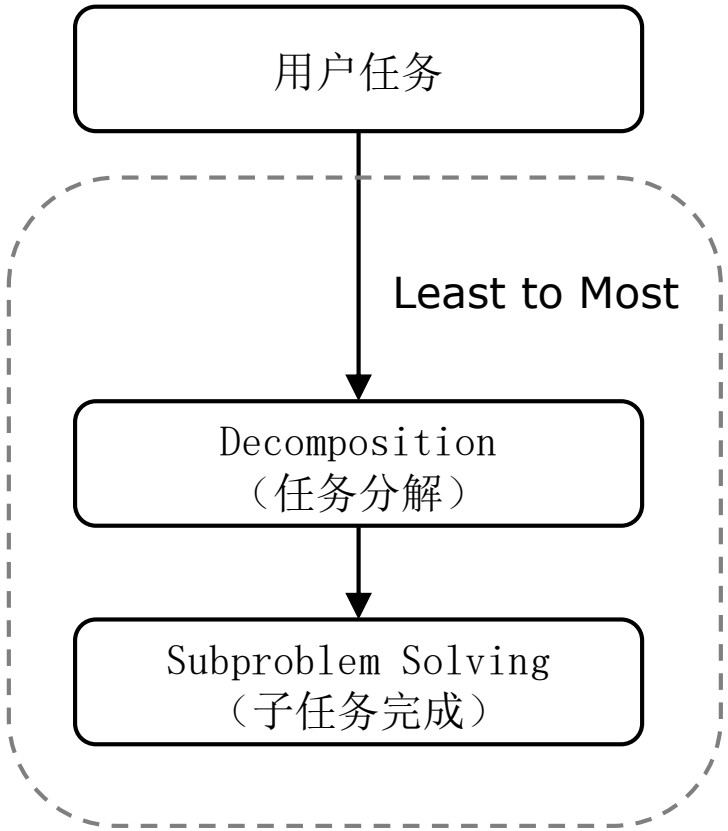
《Igniting Language Intelligence: The Hitchhiker's Guide From Chain-of-Thought Reasoning to Language Agents》，上海交大

从少到多：L2M（Least to Most Prompting）

LONGFOR 龙湖

《Complexity-Based Prompting for Multi-Step Reasoning》发现，思维链的步骤越详尽、推理步骤越多时，模型的性能越好。更复杂思维链作为提示时，模型生成的结果会更加置信。

核心逻辑



Q: “jump left”
A: The output of “jump left” concatenates: the output of “turn left”, the output of “jump”. “turn left” outputs “TURN_LEFT”. “jump” outputs “JUMP”. So concatenating the output of “turn left” and the output of “jump” leads to “TURN_LEFT” + “JUMP”. So the output of “jump left” is “TURN_LEFT” + “JUMP”.

Q: “run and look twice”
A: The output of “run and look twice” concatenates: the output of “run”, the output of “look twice”. “run” outputs “RUN”. “look twice” outputs “LOOK” * 2. So concatenating the output of “run” and the output of “look twice” leads to “RUN” + “LOOK” * 2. So the output of “run and look twice” is “RUN” + “LOOK” * 2.

Method	Standard prompting	Chain-of-Thought	Least-to-Most
code-davinci-002	16.7	16.2	**99.7**
text-davinci-002	6.0	0.0	**76.0**
code-davinci-001	0.4	0.0	**60.7**

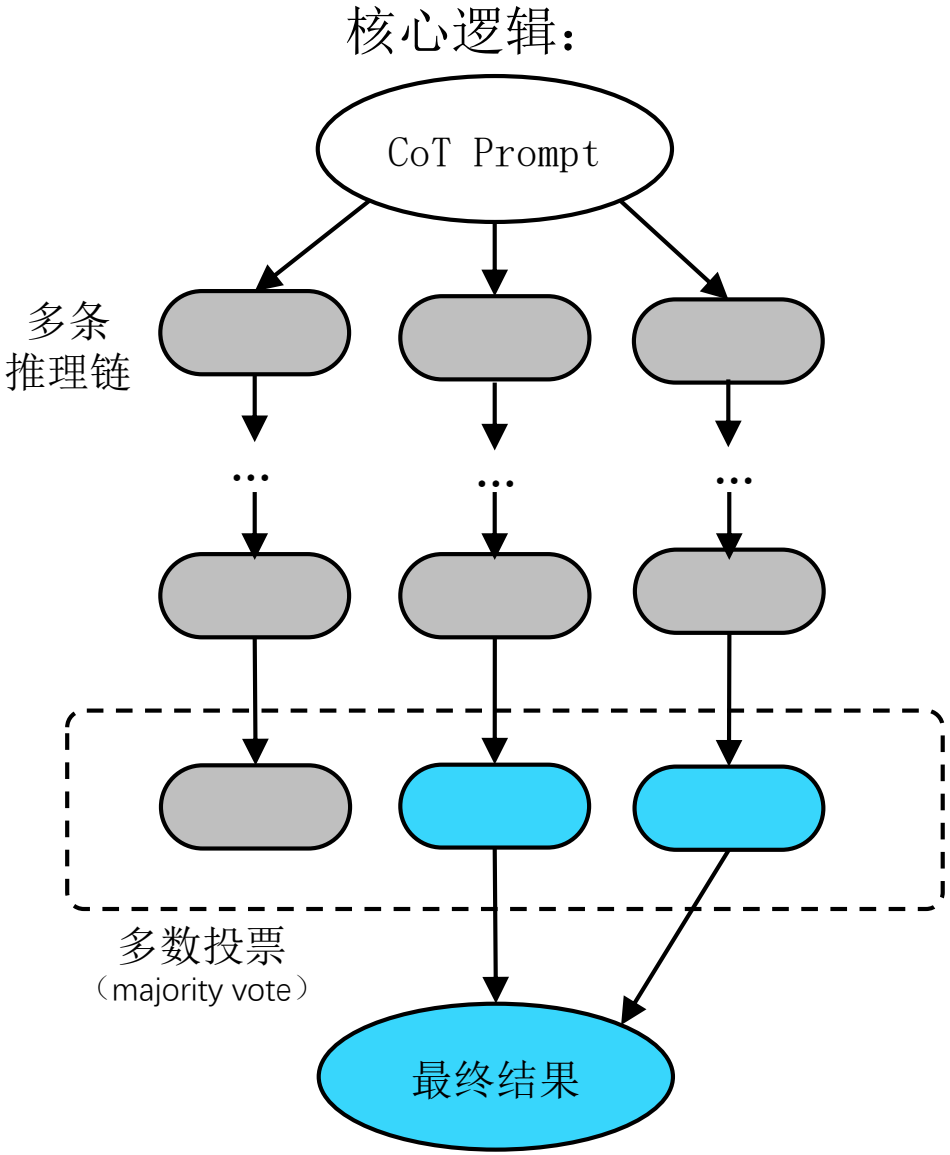
示例：least-to-most

Least-to-Most 在某种意义上拥有了AI Agent的雏形。

该模式当前往往用来有效的增强RAG应用。

《LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS》，Google Brain

自我一致性：Self-Consistency



自我一致性（CoT-SC，Self-Consistency）利用CoT生成多个推理路径和答案，选择答案出现最多的作为最终答案输出。

自一致性（Self-Consistency）方法包括三个步骤：

1 使用思维链（CoT）提示语言模型；
2 从语言模型的解码器中采样，取代CoT提示中的“贪婪解码”，生成一组不同的推理路径；
3 通过在最终答案集中选择最一致的答案拿到聚合结果。

特点：

1 思路简单、容易实现；
2 是一种集成方法，容易与现有的其他方法进行集成。

CoT-SC在本质上是一种Monte Carlo method（蒙特卡洛方法），利用随机化seed控制大语言模型生成Next Token过程。CoT-SC的上限是模型平均能力的表现。

与Zero-shot CoT一样，被后续的研究作为基准的Benchmark。

《SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS》，Google

自我一致性：Self-Consistency

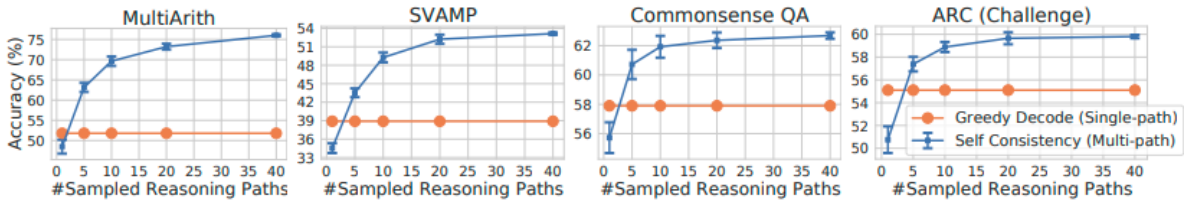
LONGFOR 龙湖

结论1 数学推理上有比较大提升，常识与符号推理也有提升

	Method	AddSub	MultiArith	ASDiv	AQuA	SVAMP	GSM8K
	Previous SoTA	**94.9**[a]	60.5[a]	75.3[b]	37.9[c]	57.4[d]	35[e] / 55[g]
UL2-20B	CoT-prompting	18.2	10.7	16.9	23.6	12.6	4.1
	Self-consistency	24.8 (+6.6)	15.0 (+4.3)	21.5 (+4.6)	26.9 (+3.3)	19.4 (+6.8)	7.3 (+3.2)
LaMDA-137B	CoT-prompting	52.9	51.8	49.0	17.7	38.9	17.1
	Self-consistency	63.5 (+10.6)	75.7 (+23.9)	58.2 (+9.2)	26.8 (+9.1)	53.3 (+14.4)	27.7 (+10.6)
PaLM-540B	CoT-prompting	91.9	94.7	74.0	35.8	79.0	56.5
	Self-consistency	93.7 (+1.8)	99.3 (+4.6)	81.9 (+7.9)	48.3 (+12.5)	86.6 (+7.6)	74.4 (+17.9)
GPT-3 Code-davinci-001	CoT-prompting	57.2	59.5	52.7	18.9	39.8	14.6
	Self-consistency	67.8 (+10.6)	82.7 (+23.2)	61.9 (+9.2)	25.6 (+6.7)	54.5 (+14.7)	23.4 (+8.8)
GPT-3 Code-davinci-002	CoT-prompting	89.4	96.2	80.1	39.8	75.8	60.1
	Self-consistency	91.6 (+2.2)	**100.0** (+3.8)	**87.8** (+7.6)	**52.0** (+12.2)	**86.8** (+11.0)	**78.0** (+17.9)

	Method	CSQA	StrategyQA	ARC-e	ARC-c	Letter (4)	Coinflip (4)
	Previous SoTA	**91.2**[a]	73.9[b]	86.4[c]	75.0[c]	N/A	N/A
UL2-20B	CoT-prompting	51.4	53.3	61.6	42.9	0.0	50.4
	Self-consistency	55.7 (+4.3)	54.9 (+1.6)	69.8 (+8.2)	49.5 (+6.8)	0.0 (+0.0)	50.5 (+0.1)
LaMDA-137B	CoT-prompting	57.9	65.4	75.3	55.1	8.2	72.4
	Self-consistency	63.1 (+5.2)	67.8 (+2.4)	79.3 (+4.0)	59.8 (+4.7)	8.2 (+0.0)	73.5 (+1.1)
PaLM-540B	CoT-prompting	79.0	75.3	95.3	85.2	65.8	88.2
	Self-consistency	80.7 (+1.7)	**81.6** (+6.3)	**96.4** (+1.1)	**88.7** (+3.5)	70.8 (+5.0)	91.2 (+3.0)
GPT-3 Code-davinci-001	CoT-prompting	46.6	56.7	63.1	43.1	7.8	71.4
	Self-consistency	54.9 (+8.3)	61.7 (+5.0)	72.1 (+9.0)	53.7 (+10.6)	10.0 (+2.2)	75.9 (+4.5)
GPT-3 Code-davinci-002	CoT-prompting	79.0	73.4	94.0	83.6	70.4	99.0
	Self-consistency	81.5 (+2.5)	79.8 (+6.4)	96.0 (+2.0)	87.5 (+3.9)	**73.4** (+3.0)	**99.5** (+0.5)

结论2 随着采样的推理路径个数增加，精度越来越高



采样到20左右，Self-Consitency效果提升最明显。

结论3 在Zero-shot CoT表现比较差的NLP任务上，也有提升

	ANLI R1 / R2 / R3	e-SNLI	RTE	BoolQ	HotpotQA (EM/F1)
Standard-prompting (no-rationale)	69.1 / 55.8 / 55.8	85.8	84.8	71.3	27.1 / 36.8
CoT-prompting (Wei et al., 2022)	68.8 / 58.9 / 60.6	81.0	79.1	74.2	28.9 / 39.8
Self-consistency	**78.5 / 64.5 / 63.4**	**88.4**	**86.3**	**78.4**	**33.8 / 44.6**

Ye & Durrett (2022) show that sometimes chain-of-thought prompting could hurt performance compared to standard prompting in few-shot in-context learning

《SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS》, Google

自我一致性：Self-Consistency

LONGFOR 龙湖

问题1： 可否通过调整temperature、top_p参数来控制概率，降低采样推理路径的采样数量，节省token？

问题2： 论文中给出的都是封闭域的评估。对于开放域（文案撰写、代码编写、任务规划、内容润色…）等场景，如何进一步提升质量？

问题3： Cot-SC基于大部分推理链是正确的情况。如果问题过于复杂，大部分推理链都不对怎么办？

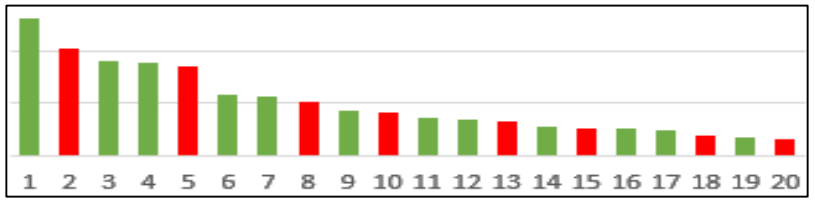
《SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS》，Google

自我一致性：Self-Consistency

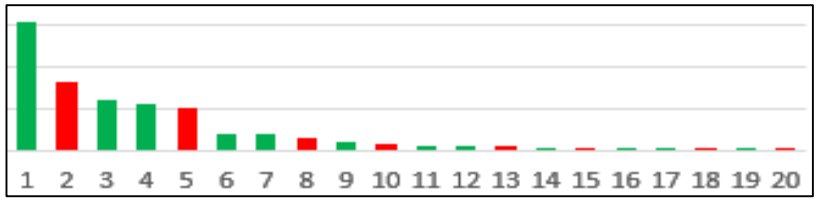
LONGFOR 龙湖

正确

错误

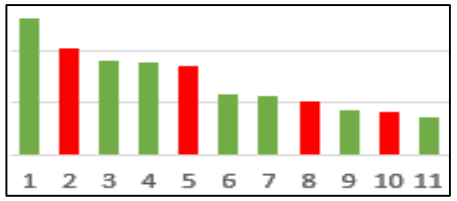


示例1：原始分布，正确70%，错误30%



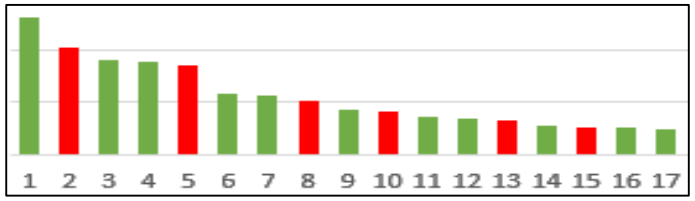
temperature = 0.5，82% ： 18%

√



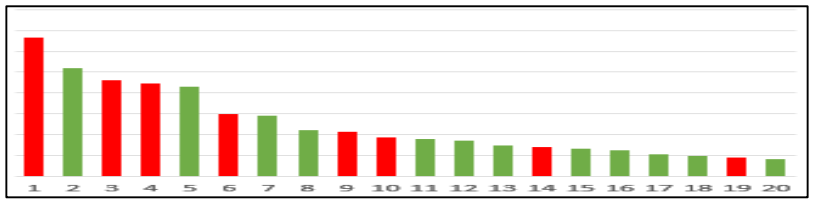
top_p = 0.5，73% ： 27%

√

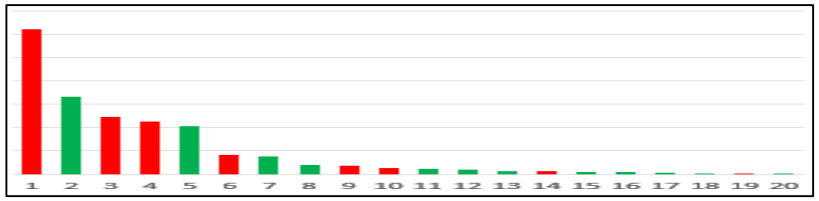


top_p = 0.9，72% ： 28%

√

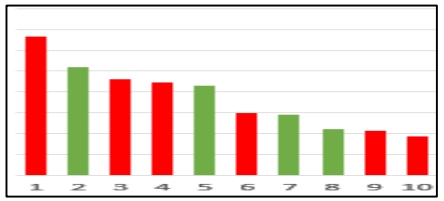


示例2：原始分布，正确60%，错误40%



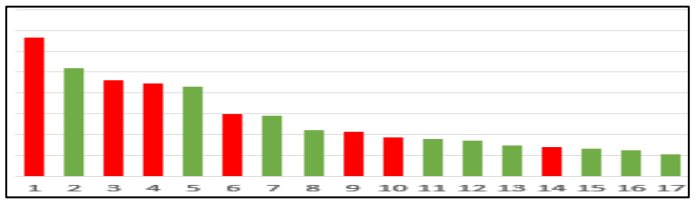
temperature = 0.5，37% ： 63%

X



top_p = 0.5，35% ： 65%

X



top_p = 0.9，57% ： 43%

√

结论： CoT-SC的适用性在于正确答案占比高于错误答案的情况。应保持分布接近于原始分布。

1、降低temperature对于原始分布的影响较大，与CoT-SC难以结合。

2、适当小幅降低top_p，在大多数情况下可调整推理链数量节省token。

自我一致性：Self-Consistency

问题1： 可否通过调整temperature、top_p参数来控制概率，降低采样推理路径的采样数量，节省token？

问题2： 论文中给出的都是封闭域的评估。对于开放域（文案撰写、代码编写、任务规划、内容润色…）等场景，如何进一步提升质量？

问题3： Cot-SC基于大部分推理链是正确的情况。如果问题过于复杂，几乎每条推理链都不对怎么办？

CoT-SC Demo演示 cot-sc

反馈精炼：Self-Refine

LONGFOR 龙湖

流程

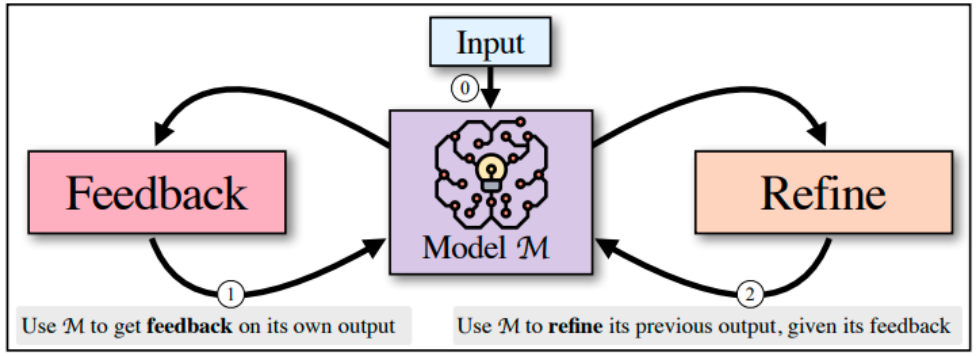


Figure 1: Given an input (⓪), SELF-REFINE starts by generating an output and passing it back to the same model $\mathcal{M}$ to get feedback (①). The feedback is passed back to $\mathcal{M}$, which refines the previously generated output (②). Steps (①) and (②) iterate until a stopping condition is met. SELF-REFINE is instantiated with a language model such as GPT-3.5 and does not involve human assistance.

示例

(a) **Dialogue:** x, y_t

```
User: I am interested
in playing Table
tennis.

Response: I'm sure
it's a great way to
socialize, stay active
```

(b) **FEEDBACK** fb

```
Engaging: Provides no
information about table
tennis or how to play it.

User understanding: Lacks
understanding of user's
needs and state of mind.
```

(c) **REFINE** y_{t+1}

```
Response (refined): That's
great to hear (...) ! It's
a fun sport requiring
quick reflexes and good
hand-eye coordination.
Have you played before, or
are you looking to learn?
```

(d) **Code optimization:** x, y_t

```
Generate sum of 1, ..., N
def sum(n):
    res = 0
    for i in range(n+1):
        res += i
    return res
```

(e) **FEEDBACK** fb

```
This code is slow as
it uses brute force.
A better approach is
to use the formula
... (n(n+1))/2.
```

(f) **REFINE** y_{t+1}

```
Code (refined)

def sum_faster(n):
  return (n*(n+1))//2
```

方法

Algorithm 1 SELF-REFINE algorithm

Require: input x, model $\mathcal{M}$, prompts $\{p_{\text{gen}}, p_{\text{fb}}, p_{\text{refine}}\}$, stop condition $\text{stop}(\cdot)$

$y_0 = \mathcal{M}(p_{\text{gen}} \| x)$ ▷ Initial generation (Eqn. 1)
for iteration $t \in 0, 1, \ldots$ **do**
  $fb_t = \mathcal{M}(p_{\text{fb}} \| x \| y_t)$ ▷ Feedback (Eqn. 2)
  if $\text{stop}(fb_t, t)$ **then** ▷ Stop condition
    break
  else
    $y_{t+1} = \mathcal{M}(p_{\text{refine}} \| x \| y_0 \| fb_0 \| \ldots \| y_t \| fb_t)$ ▷ Refine (Eqn. 4)
  end if
end for
return y_t

Figure 3: The SELF-REFINE algorithm. See (§2) for a discussion of each component.

实验结果

	GPT-3.5		ChatGPT		GPT-4	
Task	Base	+SELF-REFINE	Base	+SELF-REFINE	Base	+SELF-REFINE
Sentiment Reversal	8.8	**30.4** (↑21.6)	11.4	**43.2** (↑31.8)	3.8	**36.2** (↑32.4)
Dialogue Response	36.4	**63.6** (↑27.2)	40.1	**59.9** (↑19.8)	25.4	**74.6** (↑49.2)
Code Optimization	14.8	**23.0** (↑8.2)	23.9	**27.5** (↑3.6)	27.3	**36.0** (↑8.7)
Code Readability	37.4	**51.3** (↑13.9)	27.7	**63.1** (↑35.4)	27.4	**56.2** (↑28.8)
Math Reasoning	**64.1**	**64.1** (0)	74.8	**75.0** (↑0.2)	92.9	**93.1** (↑0.2)
Acronym Generation	41.6	**56.4** (↑14.8)	27.2	**37.2** (↑10.0)	30.4	**56.0** (↑25.6)
Constrained Generation	28.0	**37.0** (↑9.0)	44.0	**67.0** (↑23.0)	15.0	**45.0** (↑30.0)

Table 1: SELF-REFINE results on various tasks using GPT-3.5, ChatGPT, and GPT-4 as base LLM. SELF-REFINE consistently improves LLM. Metrics used for these tasks are defined in Section 3.2.

《Self Refine: Iterative Refinement with Self-Feedback》, CMU, Nvidia

结论

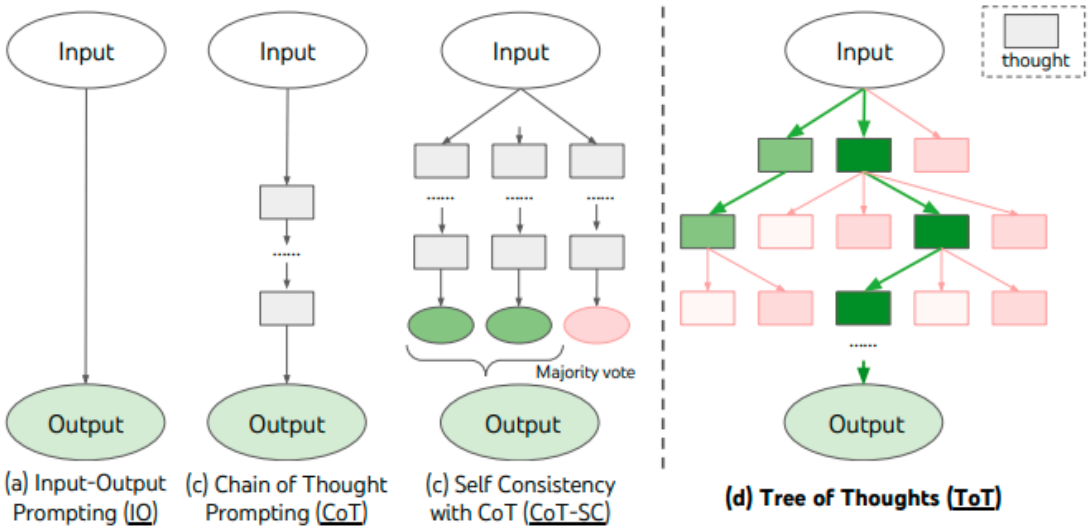
- 在情感逆转、交互对话、代码可读性提升、约束生成等开放域（文科）问题上，Self-Refine提升非常明显；在数学、物理等问题上，几乎没有提升。
- 在大模型上提升明显；在小的模型上（Vicuna-13B），refinement process表现很差，不能提供好的反馈意见。

用LLM本身，或者外部环境对于LLM进行反馈预评价，是提升LLM Reasoning能力，进行Agent规划和行动能力增强的手段。

示例：self-refine

《Self Refine: Iterative Refinement with Self-Feedback》，CMU，Nvidia

思维树：ToT（Tree of Thought）



Algorithm 1 ToT-BFS$(x, p_\theta, G, k, V, T, b)$

Require: Input x, LM p_θ, thought generator $G()$ & size limit k, states evaluator $V()$, step limit T, breadth limit b.

$S_0 \leftarrow \{x\}$

for $t = 1, \cdots, T$ **do**

$S'_t \leftarrow \{[s, z] \mid s \in S_{t-1}, z_t \in \mathrm{G}(p_\theta, s, k)\}$

$V_t \leftarrow V(p_\theta, S'_t)$

$S_t \leftarrow \arg\max_{S \subset S'_t, |S|=b} \sum_{s \in S} V_t(s)$

end for

return $G(p_\theta, \arg\max_{s \in S_T} V_T(s), 1)$

Algorithm 2 ToT-DFS$(s, t, p_\theta, G, k, V, T, v_{th})$

Require: Current state s, step t, LM p_θ, thought generator $G()$ and size limit k, states evaluator $V()$, step limit T, threshold v_{th}

if $t > T$ **then** record output $G(p_\theta, s, 1)$

end if

for $s' \in G(p_\theta, s, k)$ **do** ▷ sorted candidates

if $V(p_\theta, \{s'\})(s) > v_{thres}$ **then** ▷ pruning

DFS$(s', t+1)$

end if

end for

BFS：广度优先

DFS：深度优先

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.

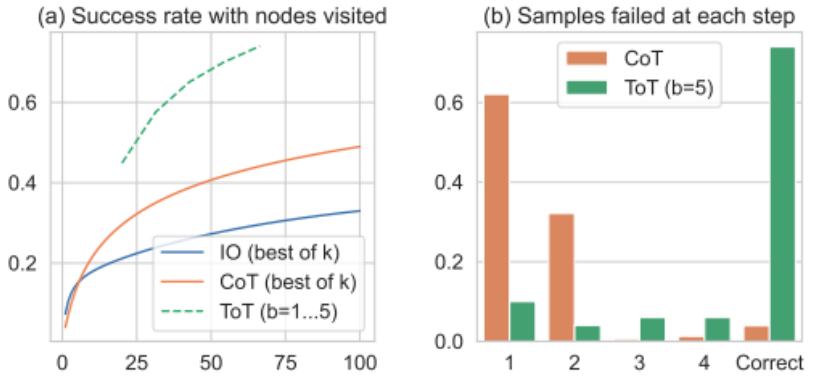


Figure 3: Game of 24 (a) scale analysis & (b) error analysis.

GitHub - princeton-nlp/tree-of-thought-llm

Tree of Thought 主要由4个部分组成：

- **Thought Decomposition**：思维拆分，设计中间步骤。需要因具体问题而定。
- **Thought Generator**：给定当前树的状态，从k个候选项中决定下一个step。使用投票的方法。
- **State Evaluator**：给定不同状态的边界，状态评估器评估它们在解决问题方面取得的进展，并作为搜索算法的启发式算法，以确定要继续探索哪些状态以及以何种顺序进行探索。State Evaluator可以是Value类型的分类或者打分，也可以是Vote形式。
- **Search Algorithm**：BFS 或者 DFS

《Tree of Thoughts: Deliberate Problem Solving with Large Language Models》, Google, Princeton

结论

- 在24点、数独、填字游戏中，ToT表现非常好，得益于对于Thought Decomposition的精确设计。但正因为此，ToT泛化很难，要针对不同场景进行细颗粒度的推理示例设计，才能使ICL+CoT工作的很好。

- Dave Hulbert提出了ToT Prompting提示法，将ToT 框架的主要概念概括成了一段简短的提示词，指导 LLM 在一次提示中对中间思维做出评估。这种做法以及简单的变形，更方便应用于AI Agent的subtasks中用于提升reasoning的性能。

示例：Tree-of-Thought

《Tree of Thoughts: Deliberate Problem Solving with Large Language Models》, Google, Princeton

思维图：GoT（Graph of Thought）

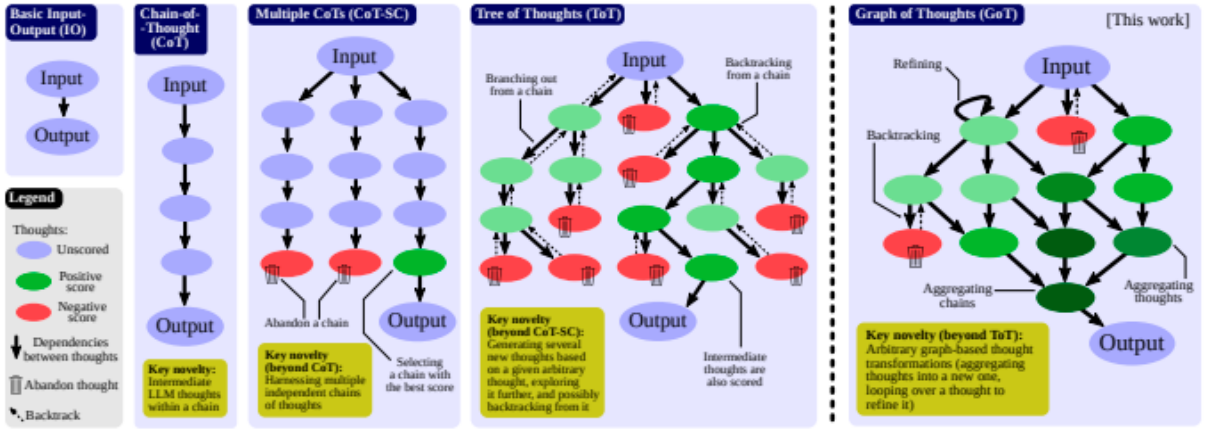


Figure 1: Comparison of Graph of Thoughts (GoT) to other prompting strategies.

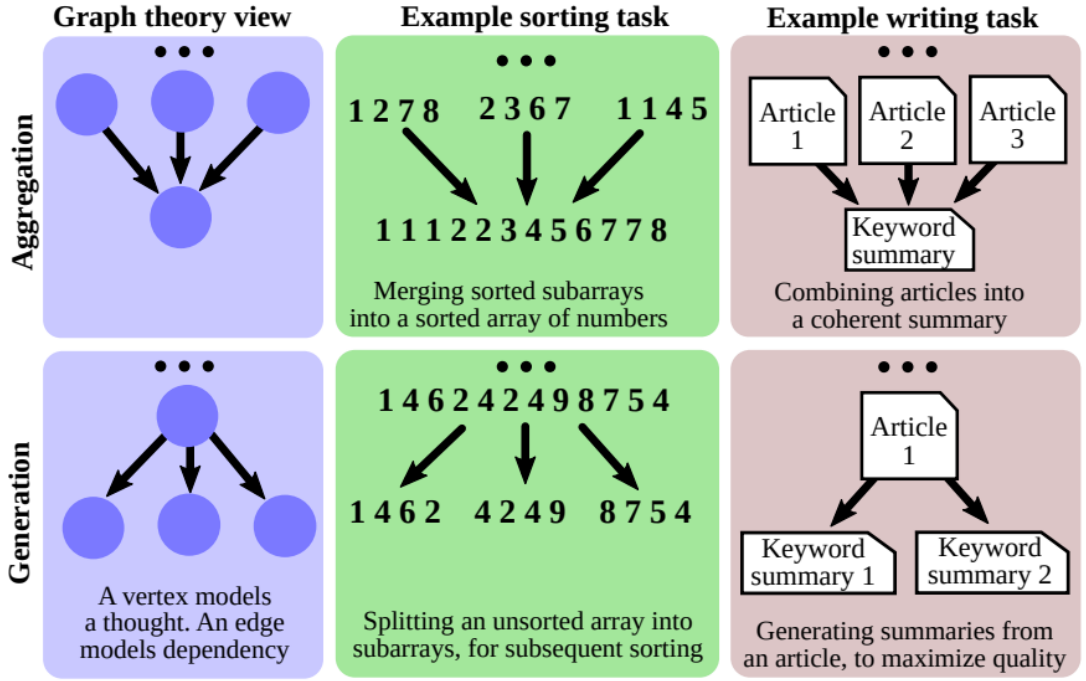
推理过程

推理过程被建模为一个有向图 G = (V, E)，其中 V 是一组顶点，表示LLM的Thinking，E ⊆ V × V 是一组边，表示依赖关系。

G 是有向的，因此边是有序顶点对 E ⊆ V × V 的子集。一个顶点包含对当前问题的一个解答，不管这个问题是最初的问题、还是中间问题或最后的问题。这种思维的具体形式取决于用例；其可能是一段文本（在写作任务中），也可能是一个数值序列（在排序任务中）。有向边 (t_1, t_2) 表示思维 t_2 的构建方式是将 t_1 用作「直接输入」，即通过明确指示 LLM 使用 t_1 来生成 t_2。

思维变换

将图模型用于推理，GoT 能实现全新的思维变换。研究者称之为图的使能变换（graph-enabled transformation）。比如，在写作任务中可以将多篇输入文章组合成一篇连贯一致的摘要。在排序时，可将多个已排序的数值子数组合并为一个最终已排序数组。



《Graph of Thoughts: Solving Elaborate Problems with Large Language Models》, warsaw university of technology

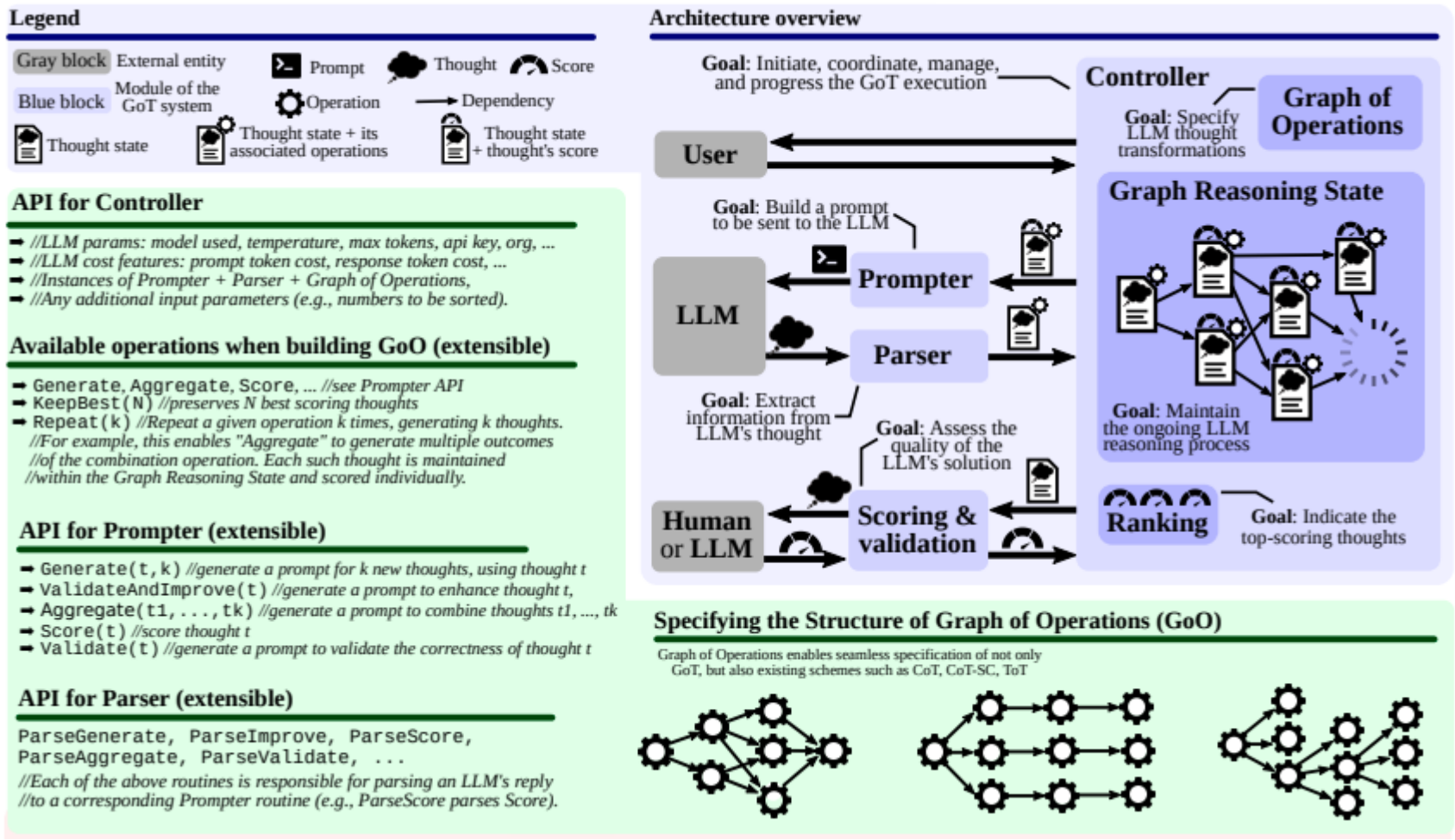
思维容量

研究团队还有另一项贡献，即提出一种新的评估指标 —— **思维容量（the volume of a thought）**，可用于评估 prompt 设计策略。使用这一指标的目标是更好地理解 prompt 设计方案之间的差异。
对于一个给定的思维 v，v 的容量是指 LLM 思维的数量，用户可以基于此使用有向边得到 v。直观上说，这些就是有望对 v 做出贡献的所有 LLM 思维。

研究团队表明，通过整合聚合等思维变换技术，GoT 能让思维容量比其它方案显著更大。

《Graph of Thoughts: Solving Elaborate Problems with Large Language Models》，warsaw university of technology

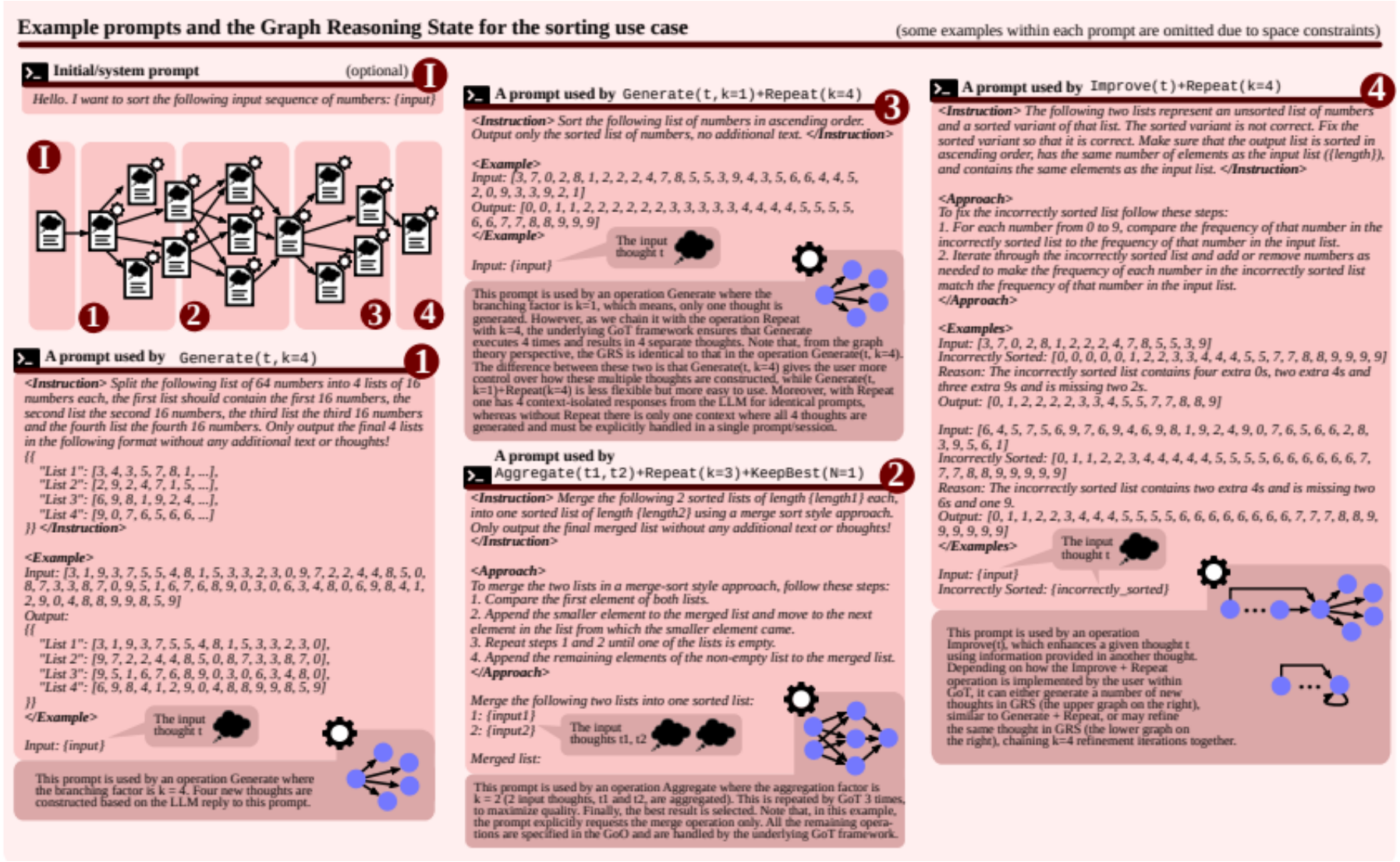
思维图：GoT（Graph of Thought）



GoT系统结构包含一系列的交互模块，这些模块包括Prompter（为LLM准备消息）、Parser（从LLM的回复中提取信息）、Scoring module（验证LLM回复并对其进行评分）和Controller（协调整个推理过程，并决定如何进行推理）。控制器包含另外两个重要元素：Graph of Operations（GoO）和Graph Reasoning State（GRS）。GoO是一种静态结构，它指定了给定任务的图分解，它规定了应用于LLM思想的转换，以及它们的顺序和依赖关系。GRS是一个动态结构，它保持正在进行的LLM推理过程的状态（其思想及其状态的历史）。

《Graph of Thoughts: Solving Elaborate Problems with Large Language Models》, warsaw university of technology

思维图：GoT（Graph of Thought）



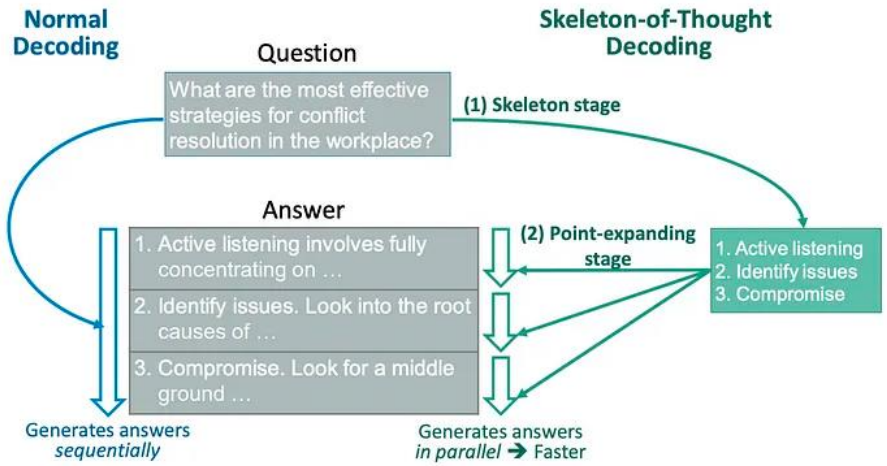
Graph Reasoning State（GRS）推理示例。

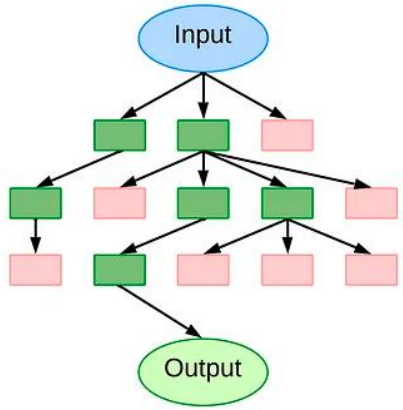
除图中数组排序的示例外，论文中给出了文档聚合，统计词频，求集合的示例。

对于每种不同的场景，需要单独针对性设计计算图。

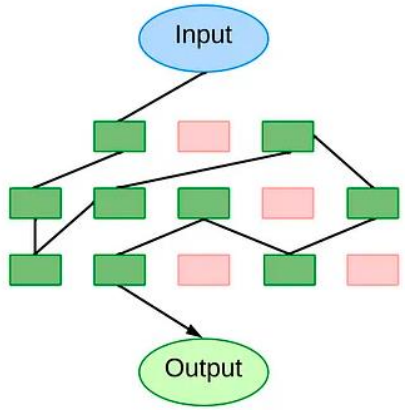
《Graph of Thoughts: Solving Elaborate Problems with Large Language Models》, warsaw university of technology

名称		机构	特点
ToT	Tree of Thoughts: Deliberate Problem Solving with Large Language Models	谷歌，普林斯顿	思维树。探索一个基本想法的多个推理分支。
GoT	Graph of Thoughts: Solving Elaborate Problems with Large Language Models	华沙理工	思维图。对思维树的想法进行总结与转换。
XoT	Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation	佐治亚大学	使用强化学习和蒙特卡洛搜索注入外部知识。
AoT	Algorithm of Thoughts: Enhancing Exploration of Ideas in Large Language Models	微软	思维算法。只有一条推理路径。推理路径动态可变，通过不断推理和反思过程，提高了推理效率。
PoT	Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks	滑铁卢大学，UCSB	思维程序。要求创建一段程序，来进行执行。
SoT	Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation	微软，清华	为了解决端到端生成延迟的问题，而不是增强推理能力。采用两阶段：生成答案的骨架模板；扩展答案。





Tree of Thoughts



Algorithm of Thoughts

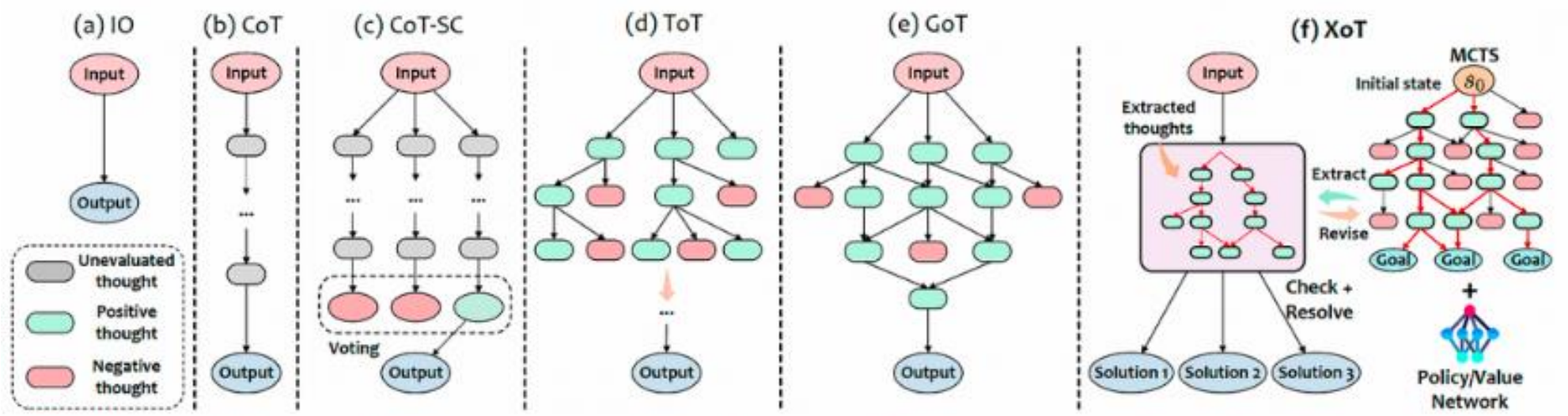
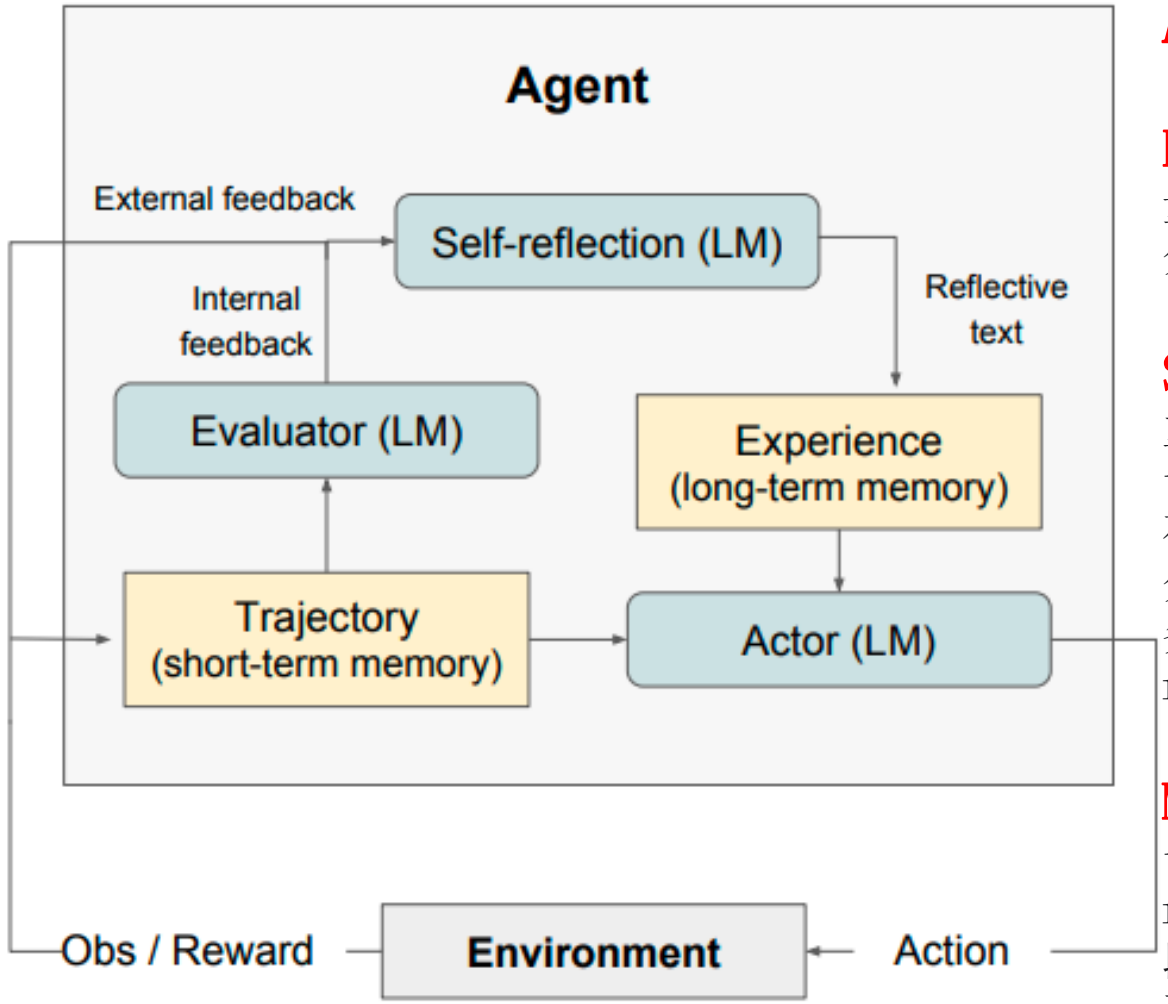


Figure 1: Comparison of XoT versus other prompting paradigms.

从错误经验中学习：Reflexion



Actor：主要工作是基于当前环境生成下一步的动作。

Evaluator：Evlauator主要工作是衡量Actor生成结果的质量。就像强化学习中的Reward函数对Actor的执行结果进行打分。

Self-reflection：Self-reflection是Reflexion框架中最重要的部分。它能结合离散的reward信号(如success/fail)、trajectory等生成具体且详细语言反馈信号，这种反馈信号会储存在Memory中，启发下一次实验的Actor执行动作。相比reward分数，这种语言反馈信号储存更丰富的信息，例如在代码生成任务中，Reward只会告诉你任务是失败还是成功，但是Self-reflection会告诉你哪一步错了，错误的原因是什么等。

Memory：分为短期记忆(short-term)和长期记忆(long-term)。在一次实验中的上下文称为短期记忆，多次试验中Self-reflection的结果称为长期记忆。类比人类思考过程，在推理阶段Actor会不仅会利用短期记忆，还会结合长期记忆中存储的重要细节，这是Reflexion框架能取得效果的关键。

《Reflexion: Language Agents with Verbal Reinforcement Learning》, NEU, MIT, Princeton

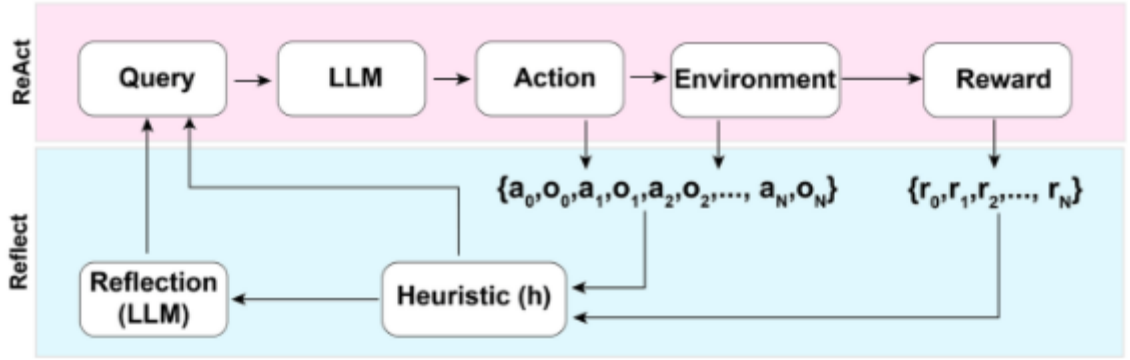
从错误经验中学习：Reflexion

LONGFOR 龙湖

Algorithm 1 Reinforcement via self-reflection

Initialize Actor, Evaluator, Self-Reflection: M_a, M_e, M_{sr}
Initialize policy $\pi_\theta(a_i|s_i), \theta = \{M_a, mem\}$
Generate initial trajectory using π_θ
Evaluate τ_0 using M_e
Generate initial self-reflection sr_0 using M_{sr}
Set $mem \leftarrow [sr_0]$
Set $t = 0$
while M_e not pass or $t <$ max trials **do**
    Generate $\tau_t = [a_0, o_0, \ldots a_i, o_i]$ using π_θ
    Evaluate τ_t using M_e
    Generate self-reflection sr_t using M_{sr}
    Append sr_t to mem
    Increment t
end while
return

Reflexion是一个迭代过程，Actor产生行动，Evaluator对Actor的行动做出评价，Self-Reflection基于行动和评价形成反思，并将反思结果存储到长期记忆中，直到Actor执行的结果达到目标效果。



Reflexion & ReAct

《Reflexion: Language Agents with Verbal Reinforcement Learning》, NEU, MIT, Princeton

数据集介绍：ALFWorld

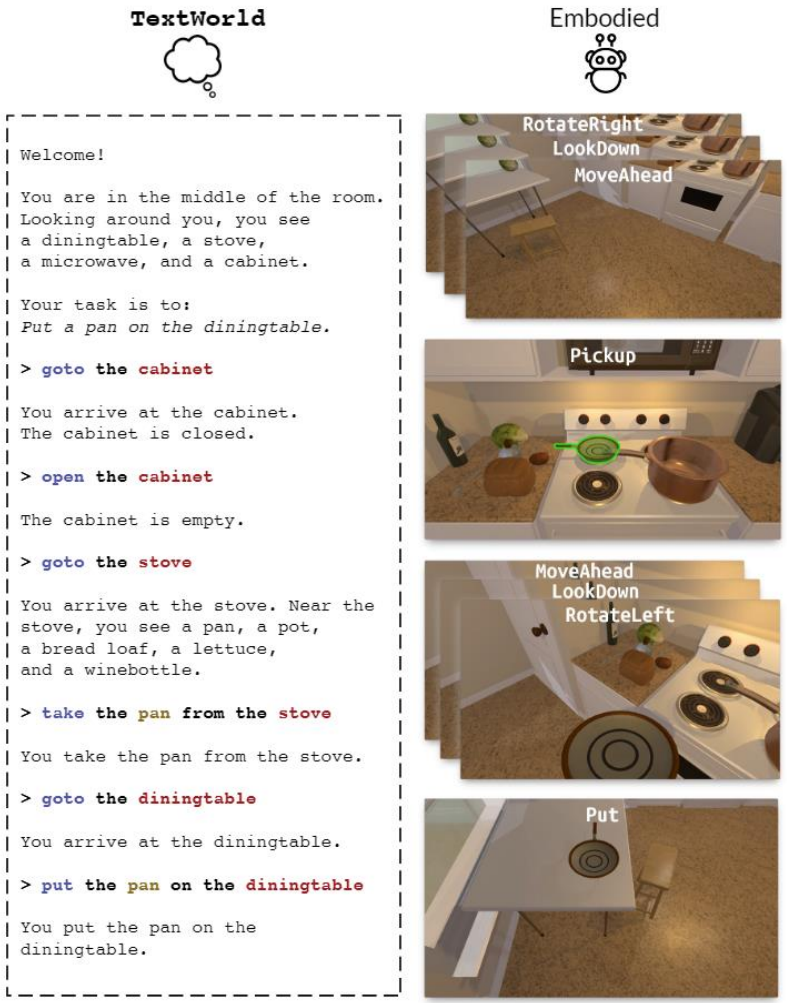


Figure 1: ALFWorld: Interactive aligned text and embodied worlds. An example with high-level text actions (left) and low-level physical actions (right).

ALFWorld是AI Agent序列决策任务的模拟测试环境。

任务：去切一个苹果
- 去客厅，水果篮找（没找到）
- 去厨房，开冰箱找
- 到水槽，清洗苹果
- 到刀架上找刀
- ……

ALFWorld是一个模拟器，用以训练和评估具身智能体（Embodied AI Agent）在Text World中学习基于文本的抽象策略，然后在丰富的环境中执行ALFRED基准中的目标。

ALFWorld允许在智能体遇到复杂的具身环境前，在抽象的语言环境中进行探索、互动和学习。

https://github.com/microsoft/TextWorld

《ALFWORLD: ALIGNING TEXT AND EMBODIED ENVIRONMENTS FOR INTERACTIVE LEARNING》, Microsoft

数据集介绍：HotpotQA

Reasoning Type	%	Example(s)
Inferring the *bridge entity* to complete the 2nd-hop question (Type I)	42	**Paragraph A:** The 2015 Diamond Head Classic was a college basketball tournament ... *Buddy Hield was named the tournament's MVP.* **Paragraph B:** *Chavano Rainier "Buddy" Hield* is a Bahamian professional basketball player for the **Sacramento Kings** of the NBA... **Q:** Which team does the player named 2015 Diamond Head Classic's MVP play for?
Comparing two entities (Comparison)	27	**Paragraph A:** LostAlone were a British rock band ... consisted of *Steven Battelle, Alan Williamson, and Mark Gibson*... **Paragraph B:** Guster is an American alternative rock band ... Founding members *Adam Gardner, Ryan Miller, and Brian Rosenworcel* began... **Q:** Did LostAlone and Guster have the same number of members? (**yes**)
Locating the **answer entity** by checking multiple properties (Type II)	15	**Paragraph A:** Several *current and former members of the Pittsburgh Pirates* – ... John Milner, **Dave Parker**, and Rod Scurry... **Paragraph B:** **David Gene Parker**, *nicknamed "The Cobra"*, is an American former player in Major League Baseball... **Q:** Which former member of the Pittsburgh Pirates was nicknamed "The Cobra"?
Inferring about the property of an entity in question through a *bridge entity* (Type III)	6	**Paragraph A:** *Marine Tactical Air Command Squadron 28* is a United States Marine Corps aviation command and control unit based at *Marine Corps Air Station Cherry Point*... **Paragraph B:** *Marine Corps Air Station Cherry Point* ... is a United States Marine Corps airfield located in **Havelock, North Carolina**, USA ... **Q:** What city is the Marine Air Control Group 28 located in?
Other types of reasoning that require more than two supporting facts (Other)	2	**Paragraph A:** ... the towns of Yodobashi, **Okubo, Totsuka, and Ochiai town** *were merged into Yodobashi ward.* ... *Yodobashi Camera is a store with its name taken from the town and ward.* **Paragraph B**: *Yodobashi Camera* Co., Ltd. is *a major Japanese retail chain specializing in electronics, PCs, cameras and photographic equipment.* **Q:** Aside from Yodobashi, what other towns were merged into the ward which gave the major Japanese retail chain specializing in electronics, PCs, cameras, and photographic equipment it's name?

词条“Return To Olympus”

[1]"Return To Olympus"是另类摇滚乐队Malfunkshun的唯一专辑。[2]这张专辑是在乐队解散后，以及主唱Andrew Wood（后来加入了Mother Love Bone乐队）于1990年死于药物过量之后发行的。[3]Pearl Jam乐队的成员Stone Gossard整理了这些歌曲，并在他自己的标签Loosegroove Records下发布了这张专辑。

词条“Mother Love Bone”

[4]Mother Love Bone是一支成立于1987年，来自华盛顿州西雅图的美国摇滚乐队。[5]这支乐队的活跃期是从1987年到1990年。[6]乐队主唱Dndrew Wood的个性和作品帮助乐队在20世纪80年代末至90年代初蓬勃发展的西雅图音乐界脱颖而出。[7]Wood去世的时间在乐队首张专辑"Apple"计划发行的几天前，从而终结了乐队对成功的希望。[8]这张专辑最终在几个月后才发布。

问：在专辑“Apple”发布前去世的Mother Love Bone乐队成员，他之前的乐队是什么？

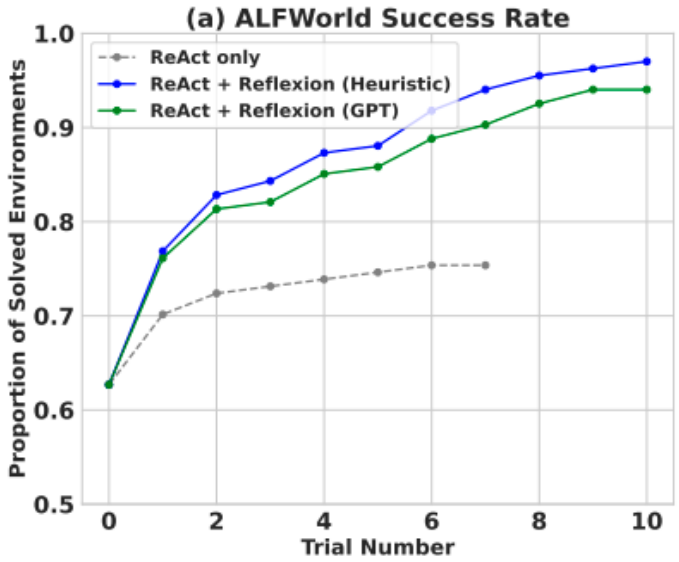
答： Malfunkshun 证据：1、2、4、6、7

HotpotQA 斯坦福、CMU和蒙特利尔大学联合推出的新型问答数据集。数据集由多跳复杂问题以及对应的答案组成，同时包含佐证证据用来解释答案的来源。

HotpotQA评测需要从整个wikipedia多篇文章中找出问题相关的隐藏的推理证据，通过这些证据结合常识进行推理，同时要求返回问题到答案的推理链。可以看做 RAG + Reasoning 构成的Agent 任务。

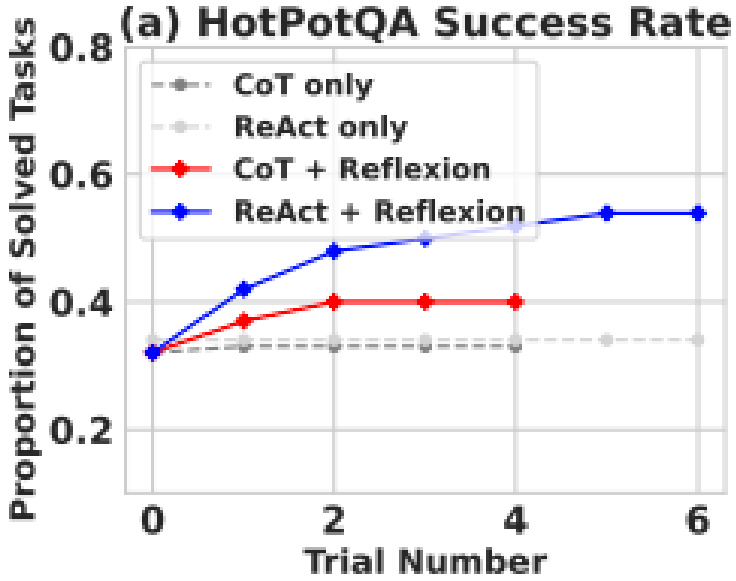
HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering - ACL Anthology

从错误经验中学习：Reflexion



ALFWorld序列决策任务

ReAct + Reflexion 用启发式和 GPT 的自我评估进行二元分类，完成了 130/134 项任务，显著优于 ReAct。



HotpotQA多跳推理任务

Reflexion+ReAct 显著优于CoT与ReAct。

Benchmark + Language	Prev SOTA Pass@1	SOTA Pass@1	Reflexion Pass@1
HumanEval (PY)	65.8 (CodeT [5] + GPT-3.5)	80.1 (GPT-4)	**91.0**
HumanEval (RS)	–	60.0 (GPT-4)	**68.0**
MBPP (PY)	67.7 (CodeT [5] + Codex [6])	**80.1** (GPT-4)	77.1
MBPP (RS)	–	70.9 (GPT-4)	**75.4**
Leetcode Hard (PY)	–	7.5 (GPT-4)	**15.0**

Table 1: Pass@1 accuracy for various model-strategy-language combinations. The base strategy is a single code generation sample. All instruction-based models follow zero-shot code generation.

《Reflexion: Language Agents with Verbal Reinforcement Learning》, NEU, MIT, Princeton

从错误经验中学习：Reflexion

LONGFOR 龙湖

Reflexion适合的情况

智能体需要从尝试和错误中学习：自我反思旨在通过反思过去的错误并将这些知识纳入未来的决策来帮助智能体提高表现。这非常适合智能体需要通过反复试验来学习的任务，例如决策、推理和编程。

传统的强化学习方法失效：传统的强化学习（RL）方法通常需要大量的训练数据和昂贵的模型微调。自我反思提供了一种轻量级替代方案，不需要微调底层语言模型，从而使其在数据和计算资源方面更加高效。

需要细致入微的反馈：自我反思利用语言反馈，这比传统强化学习中使用的标量奖励更加细致和具体。这让智能体能够更好地了解自己的错误，并在后续的试验中做出更有针对性的改进。

可解释性和直接记忆很重要：与传统的强化学习方法相比，自我反思提供了一种更可解释、更直接的情景记忆形式。智能体的自我反思存储在其记忆组件中，让分析和理解其学习过程变得更加简单。

Reflexion的有效任务类型

序列决策：自我反思提高了智能体在 AlfWorld 任务中的表现，涉及在各种环境中导航并完成多步目标。

多跳问答：自我反思提高了 HotPotQA 上智能体的性能，HotPotQA 是一个需要对多个文档进行推理的问答数据集。

编程：自我反思的智能体在 HumanEval 和 MBPP 等基准测试上编写出了更好的代码，在某些情况下实现 SOTA 结果。

Reflexion的限制

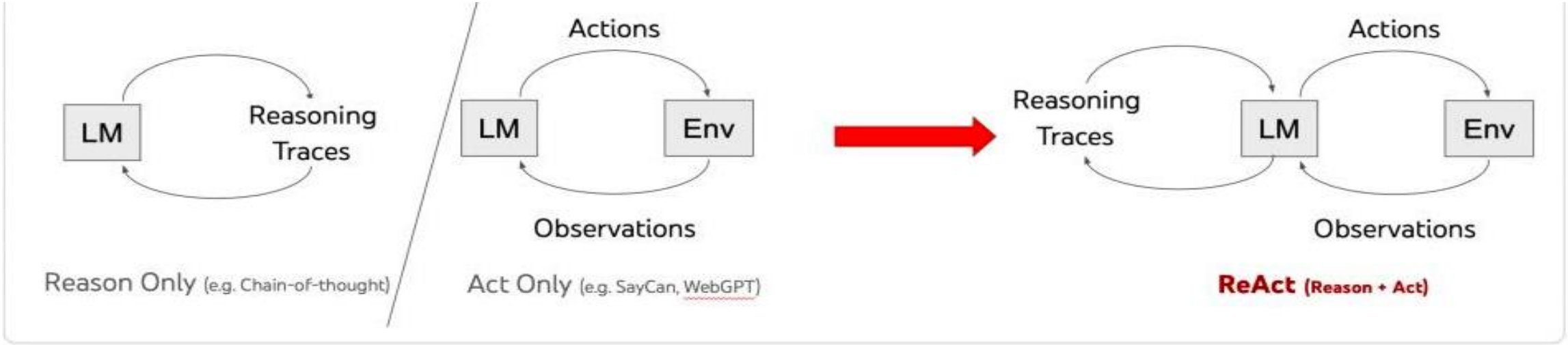
依赖自我评估能力：反思依赖于智能体准确评估其表现并产生有用反思的能力。这可能是具有挑战性的，尤其是对于复杂的任务，但随着模型功能的不断改进，预计自我反思会随着时间的推移而变得更好。

长期记忆限制：自我反思使用最大容量的滑动窗口，但对于更复杂的任务，使用向量嵌入或 SQL 数据库等高级结构可能会更有利。

示例：reflexion

《Reflexion: Language Agents with Verbal Reinforcement Learning》, NEU, MIT, Princeton

推理+行动+观察：ReAct（Thought -> Action -> Observation）

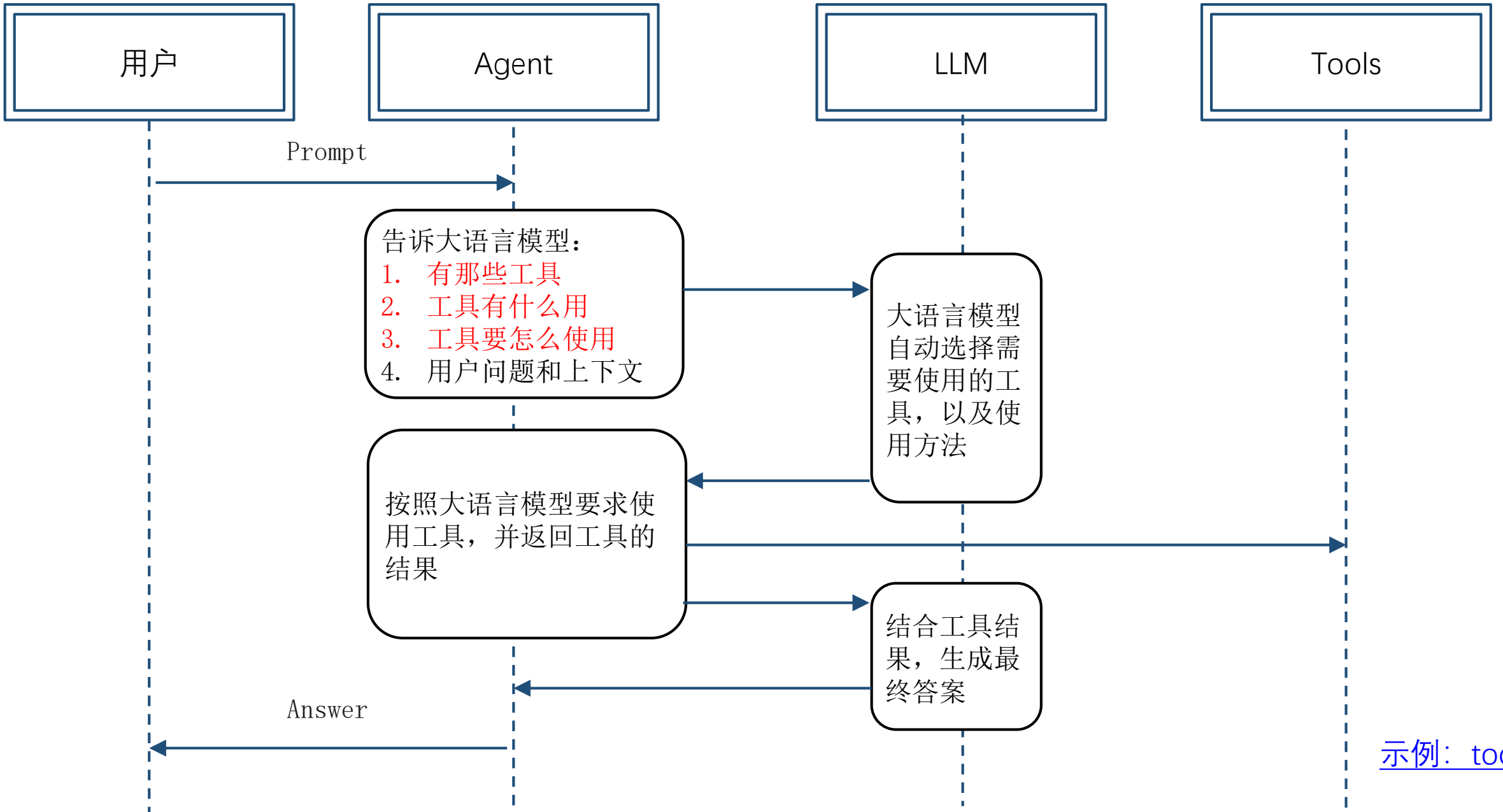


Answer the following questions as best you can. If it is in order, you can use some tools appropriately.
You have access to the following tools:
{tools}

Use the following format:
Question: the input question you must answer
Thought: you should always think about what to do and what tools to use.
Action: the action to take, should be one of [{tool_names}]
Action Input: the input to the action
Observation: the result of the action
... (this Thought/Action/Action Input/Observation can be repeated zero or more times)
Thought: I now know the final answer
Final Answer: the final answer to the original input question

《REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS》, Princeton, Google

扩展LLM能力，使用外部工具：Tool Use



[示例：tools]

推理+行动+观察：ReAct（Thought -> Action -> Observation）

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

Table 1: PaLM-540B prompting results on HotpotQA and Fever.

[a] HotpotQA EM is 27.1, 28.9, 33.8 for Standard, CoT, CoT-SC in Wang et al. (2022b).

[b] (Zhu et al., 2021; Lewis et al., 2020)

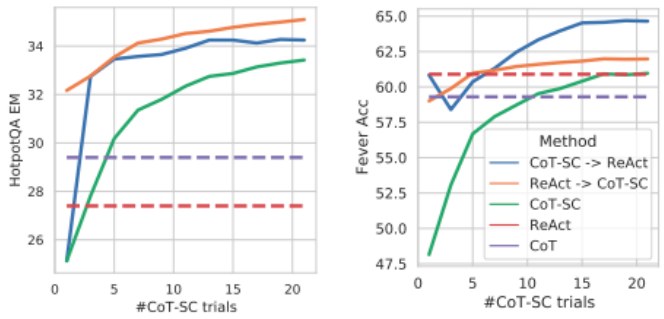
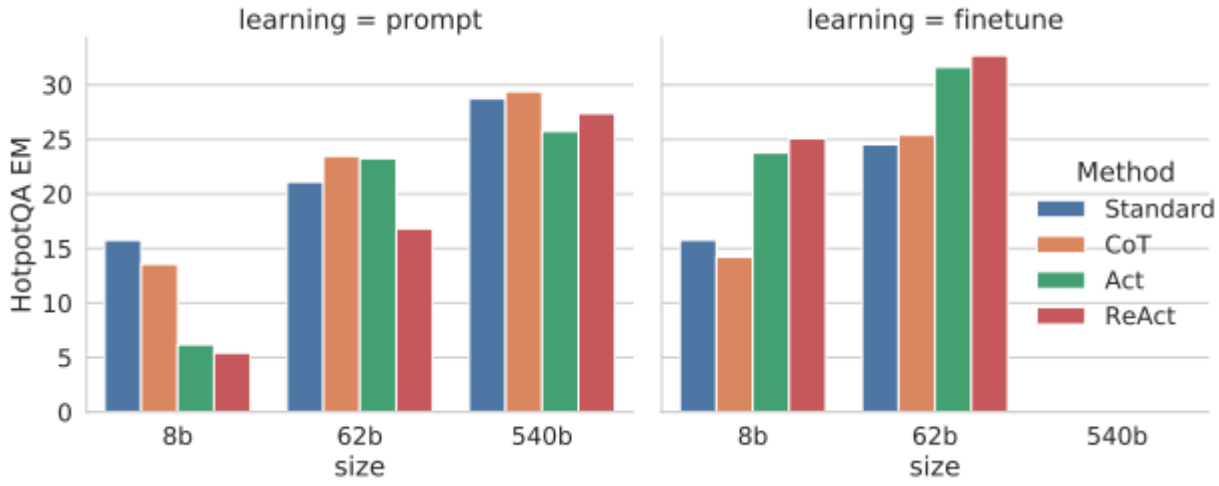


Figure 2: PaLM-540B prompting results with respect to number of CoT-SC samples used.



在Fine-Tune Model上，ReAct表现的更好

Conclusion

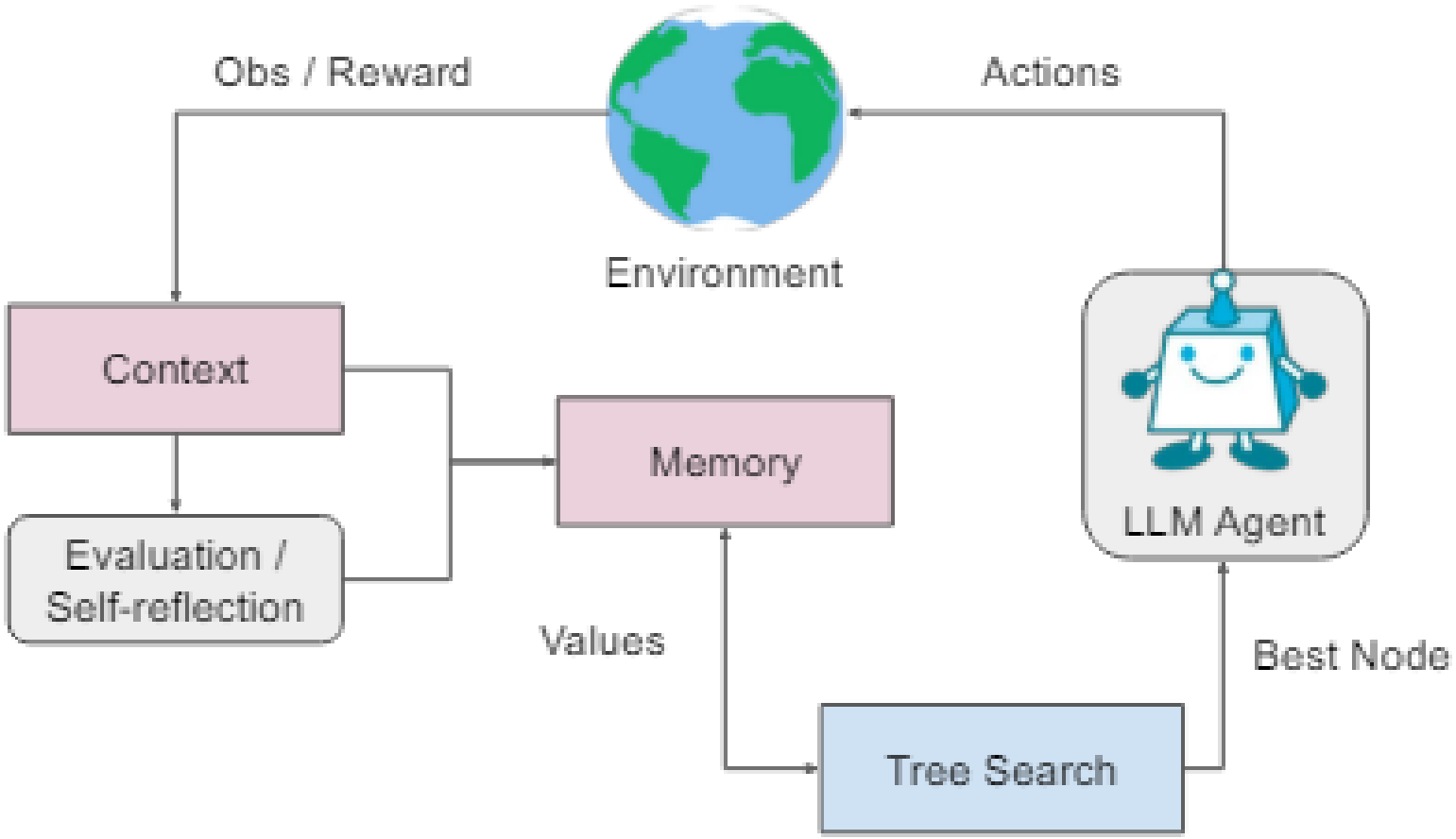
1. ReAct是一个使用LLM进行推理的推理框架，而非一项推理技术。
2. 与Plan & Solve不同，ReAct是动态Plan，根据Observation决定下一步的计划。
3. 作为推理框架，ReAct容易与Reflexion、CoT-SC、Self-Refine、Plan&Solve整合，提升整体效果。
4. 在知识推理的多跳问答中，ReAct在Standard LLM上表现一般，但在finetune LLM上工作的更好。

示例：ReAct

《REACT: SYNERGIZING REASONING AND ACTING IN LANGUAGE MODELS》, Princeton, Google

缝合怪：LATS（Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models）

LATS = ToT + ReAct + Plan & solve + Reflection + Reinforcement learning



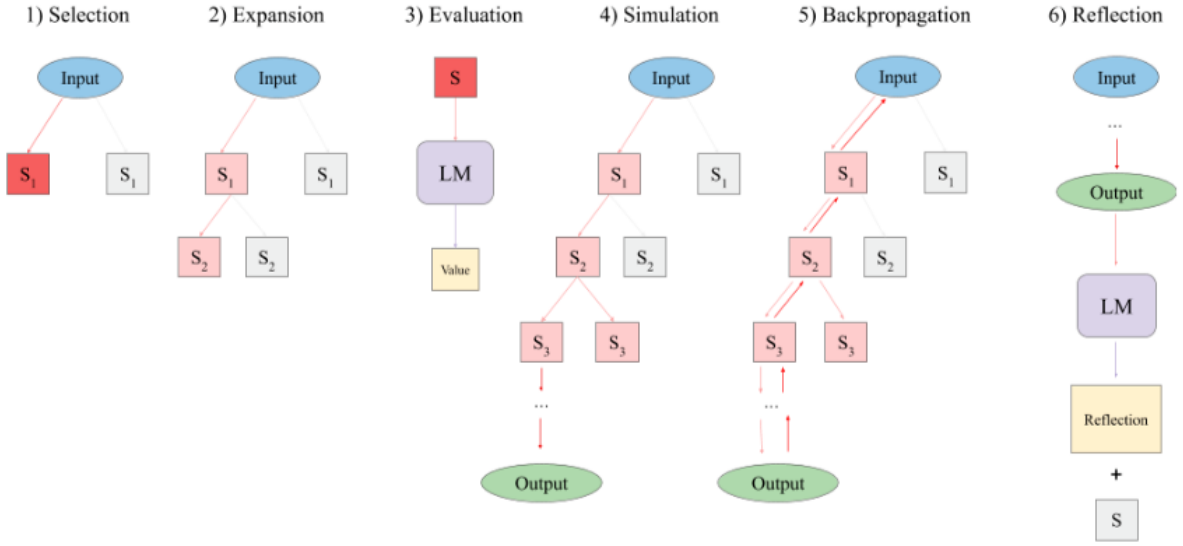
《Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models》, UIUC

缝合怪：LATS（Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models）

LONGFOR 龙湖

LATS = ToT + ReAct + Plan&solve + Reflection + Reinforcement learning

Approach	Reasoning	Acting	Planning	Self Reflection	External Memory
CoT (Wei et al., 2022)	✓	×	×	×	×
ReAct (Yao et al., 2023b)	✓	✓	×	×	×
ToT (Yao et al., 2023a)	✓	×	✓	✓	✓
RAP (Hao et al., 2023)	✓	×	✓	×	✓
Self-Refine (Madaan et al., 2023)	✓	×	×	✓	×
Beam Search (Xie et al., 2023)	✓	×	×	✓	×
Reflexion (Shinn et al., 2023)	✓	✓	×	✓	✓
LATS (Ours)	✓	✓	✓	✓	✓



Prompt Method	HotpotQA (EM)
I/O	0.32
CoT (Wei et al., 2022)	0.34
CoT - SC (Wang et al., 2022)	0.38
ToT (Yao et al., 2023a)	0.55
RAP (Hao et al., 2023)	0.60
RAP (n = 10)	0.60
LATS (CoT)	**0.60**

Prompt Method	HotpotQA (EM)
ReAct (Yao et al., 2023b)	0.32
ReAct (best of k)	0.38
Reflexion (Shinn et al., 2023)	0.51
LATS	0.61
LATS (n = 3)	0.56
LATS (n = 10)	0.64
LATS (CoT + ReAct)	**0.71**

Table 2: GPT-3.5 reasoning-based prompting (left) and acting-based prompting (right) results on HotpotQA. LATS achieves the highest exact match (EM) for acting and is competitive on reasoning. Unless otherwise specified, we sample $n = 5$ nodes during expansion and $k = 50$ trajectories.

Prompt Method	Model	Pass@1
CoT (Wei et al., 2022)	GPT-3.5	46.9
ReAct (Yao et al., 2023b)	GPT-3.5	56.9
Reflexion (Shinn et al., 2023)	GPT-3.5	68.1
ToT (Yao et al., 2023a)	GPT-3.5	54.4
RAP (Hao et al., 2023)	GPT-3.5	63.1
LATS (Ours)	GPT-3.5	**83.8**
I/O	GPT-4	80.1
Reflexion	GPT-4	91.0
LATS	GPT-4	**94.4**

Prompt Method	Pass@1
CoT (Wei et al., 2022)	54.9
ReAct (Wei et al., 2022)	67.0
Reflexion (Shinn et al., 2023)	70.0
ToT (Yao et al., 2023a)	65.8
RAP (Hao et al., 2023)	71.4
LATS (Ours)	**81.1**

Table 3: GPT-3.5 and GPT-4 Pass@1 accuracy on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). Prompting with LATS achieves the highest performance. We sample 5 solutions during expansion for 8 iterations.

《Language Agent Tree Search Unifies Reasoning Acting and Planning in Language Models》, UIUC

Prompt进阶篇（复习）

- 提升Reasoning性能的手段一般有两方面：将Task分解为详细的Plan；在执行过程中进行不断反思。
- 评估标准与自动化评估应该尽早建立，来评估Reasoning的整体性能。
- 清晰的Agentic Workflow是成功实现AI Agent的前提条件。
- 自动化的Prompt优化技术，在调试Agentic Workflow中是必要的。
- 没有银弹。根据具体的Case，需要针对场景定制ReAct, Reflexion, ToT, GoT, XoT, AoT，CoT-SC等具体策略。

课后练习

有1大杯水，期质量未知，以及3小杯等质量的水。将1小杯水倒入大杯后，大杯水温度上升了10度。然后将第2小杯水倒入大杯后，大杯水温又上升了6度。问：倒入第3小杯后，大杯水温度会上升几度？

设温差为DeltaT。大杯水质量为M大，小杯水质量为M小。根据热平衡方程，Q吸=Q放。

第一次：M大 * 10 * C水 = 1M小 * （DeltaT - 10）* C水

第二次：M大 * 16 * C水 = 2M小 * （DeltaT - 16）* C水

解得：DelaT = 40

M大 = 3M小

答案为4度。

预告：《Prompting Guide：智能体篇》

LONGFOR 龙湖

使用LLM构建AI智能体，解决特定领域任务。