

大模型智能体构建平台与应用

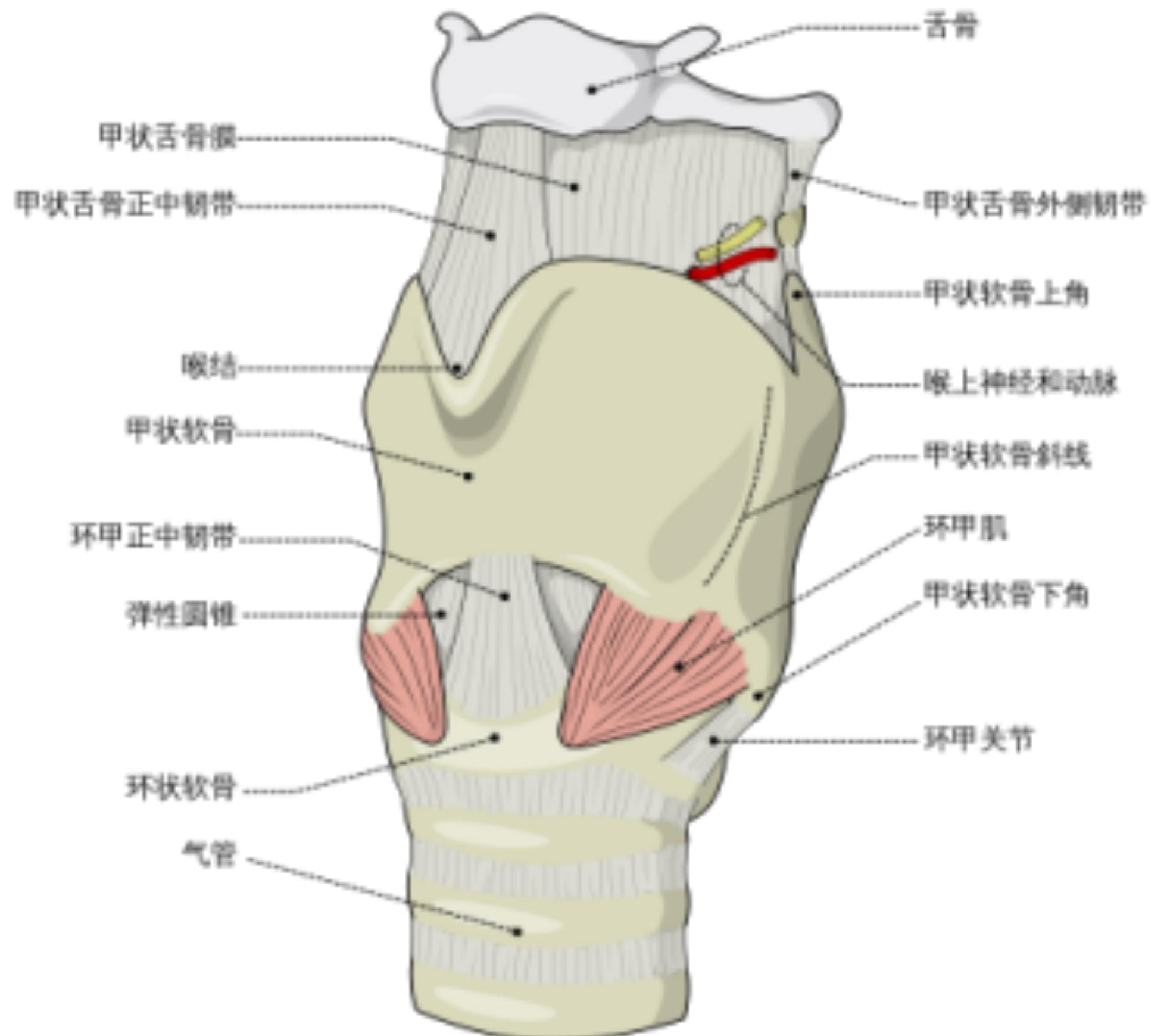
Outline

- 从LLM到智能体的演化
- 智能体的基本概念
- 智能体的构建
- 应用案例
- 展望

一、从LLM到智能体的演化

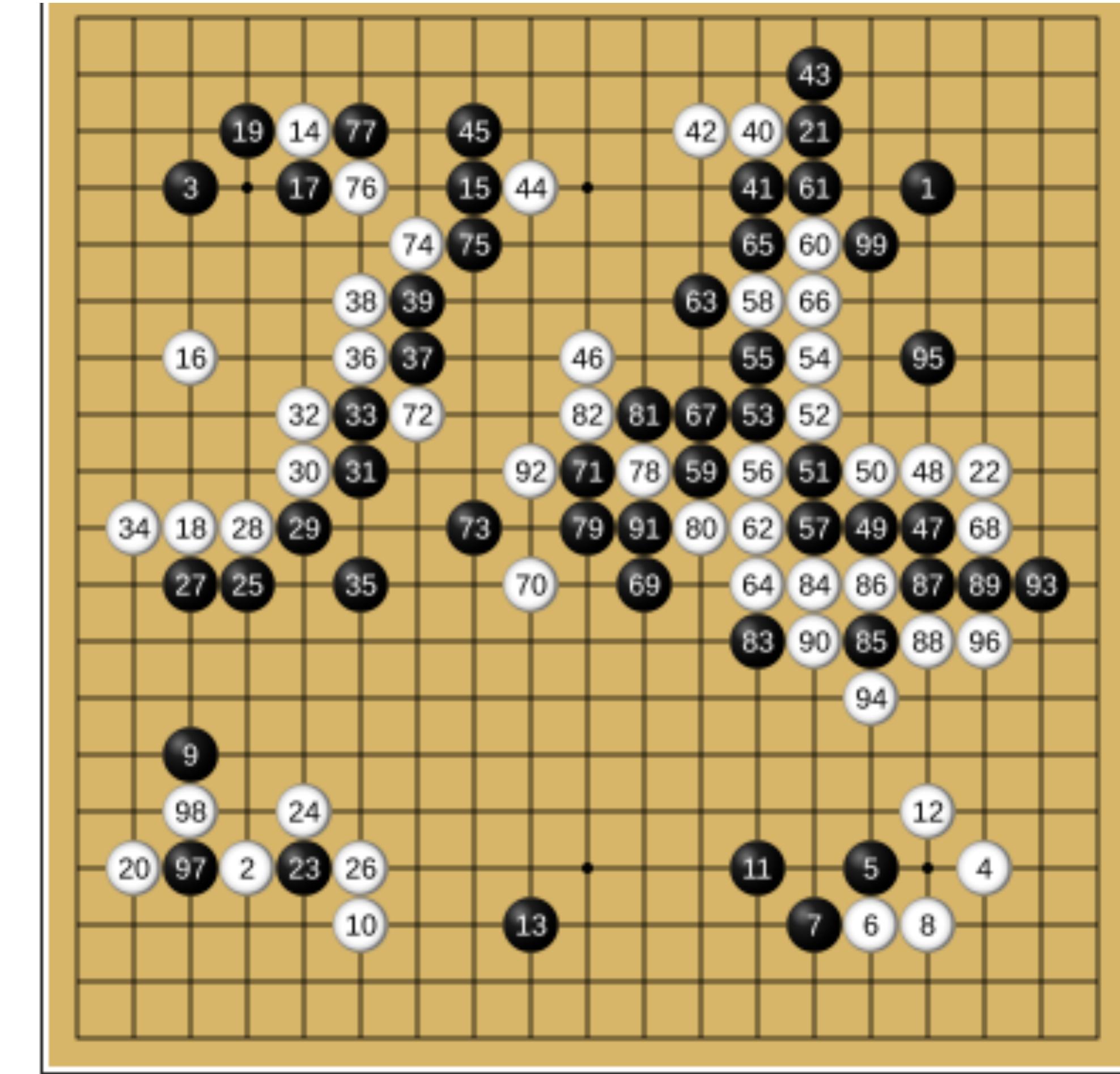
语言的起源

- 基因的突变：
- 抛物状牙弓
- 喉部位置降低
- 音：词法
- 组：语法
- 意：语义
- 用：语用
- 智人胜在团队
- 语言承载了知识和逻辑



AlphaGO 神之一手

- 古力将李世乭78手称为“神之一手”，
- AlphaGo大约在87手才发现它79手算错了。
-



ChatGPT

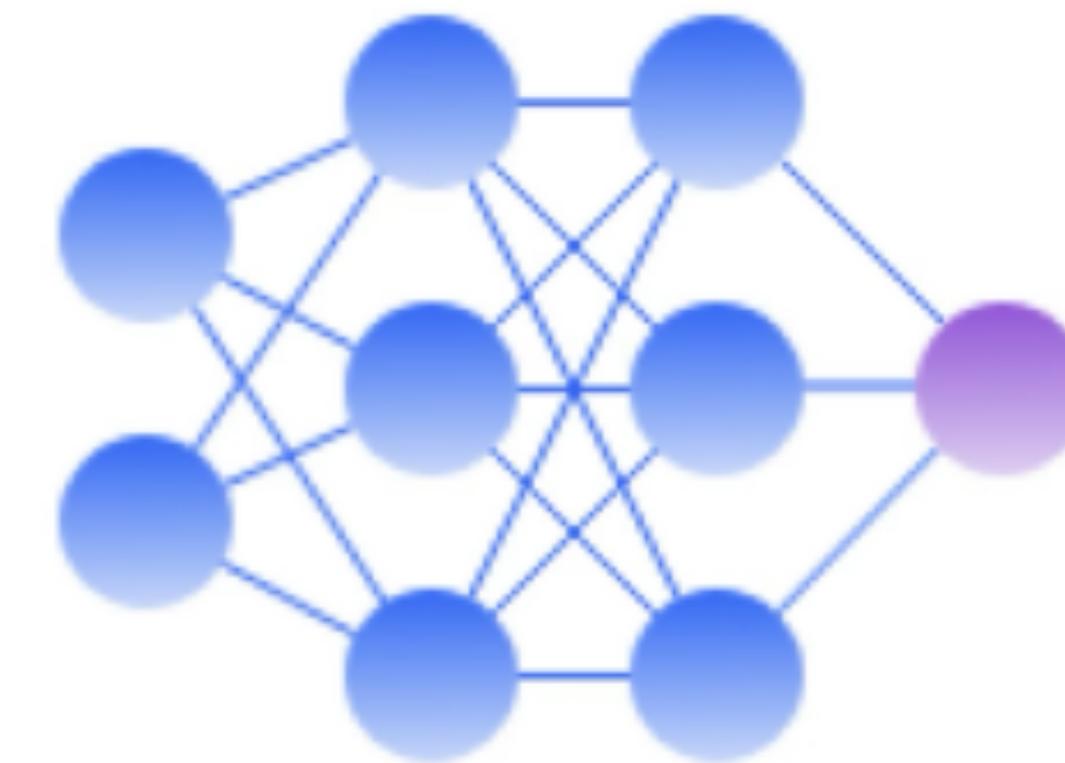
- a generative pre-trained transformer

ChatGPT		
 Examples	 Capabilities	 Limitations
"Explain quantum computing in simple terms" →	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?" →	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?" →	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

Image

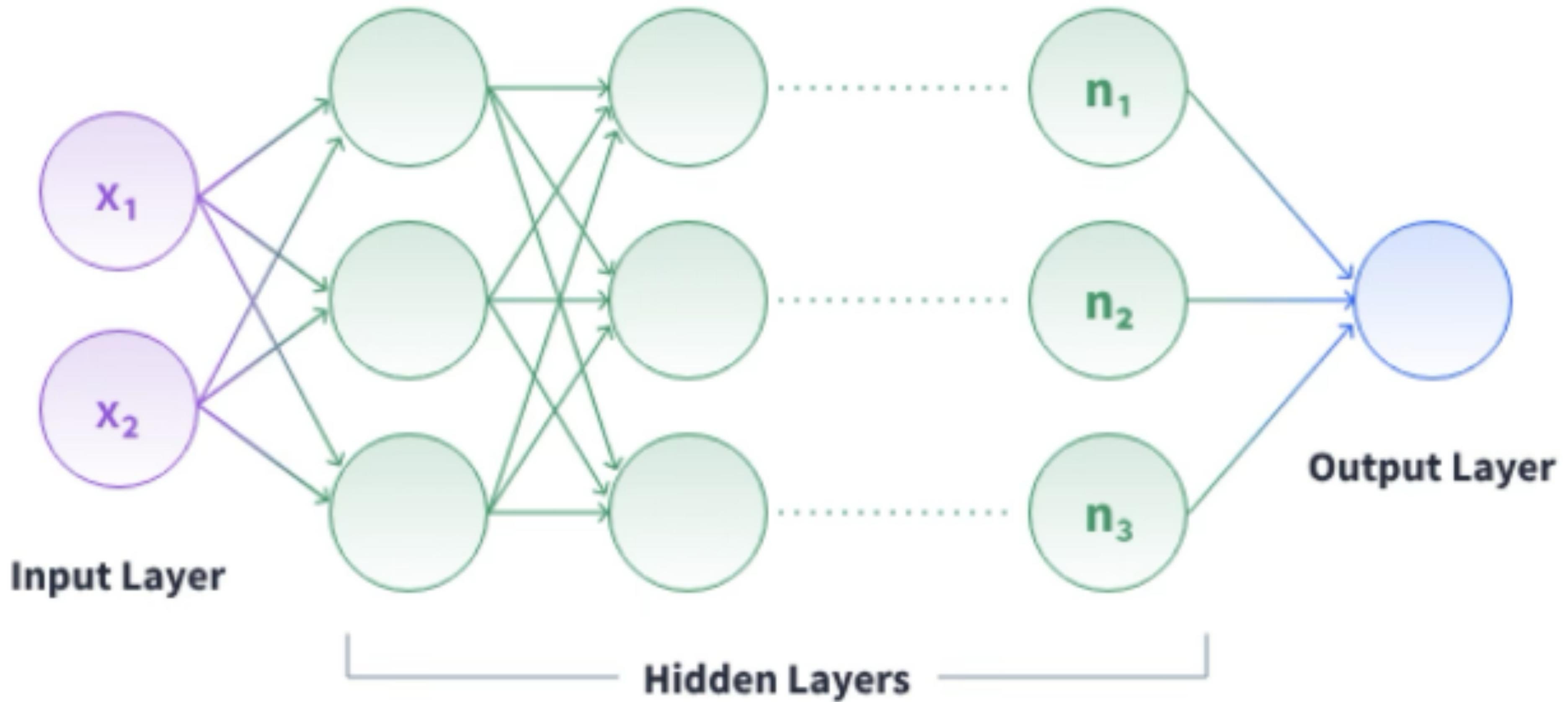


Neural network

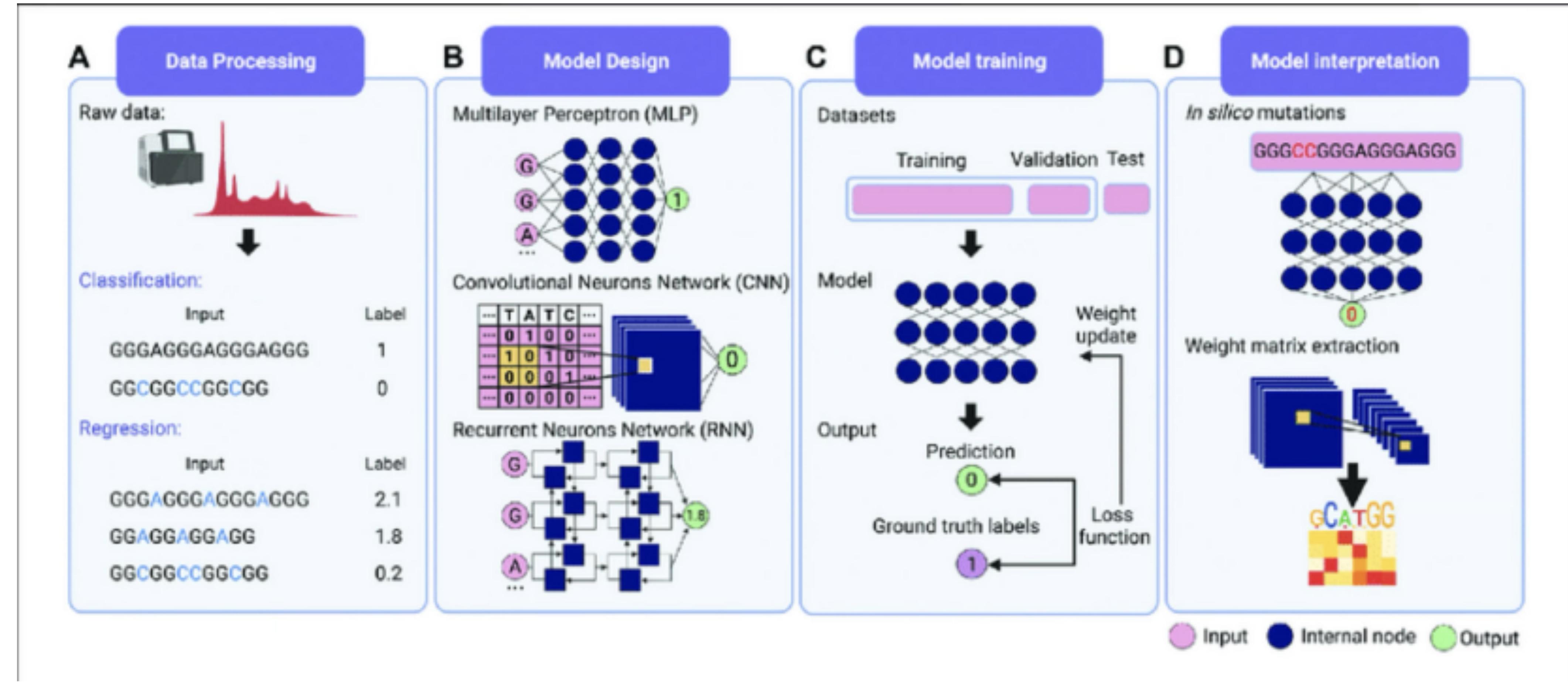


BIRD

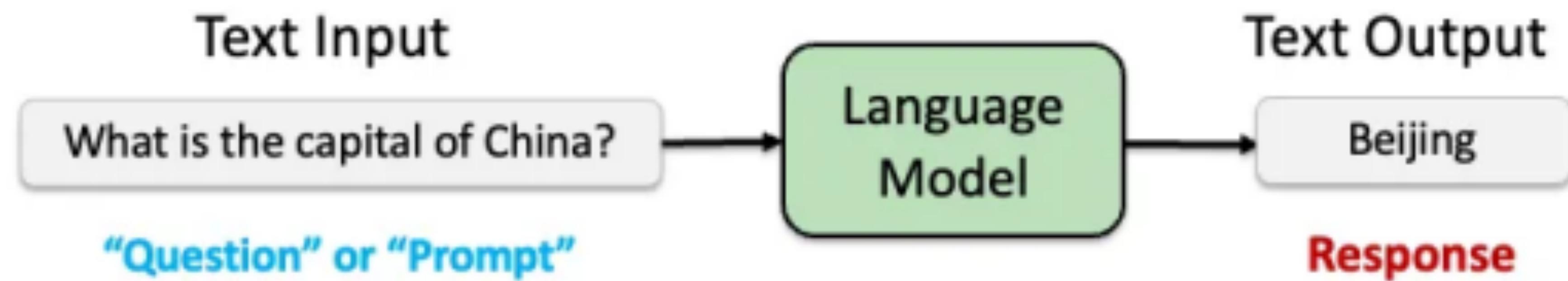
神经网络时代



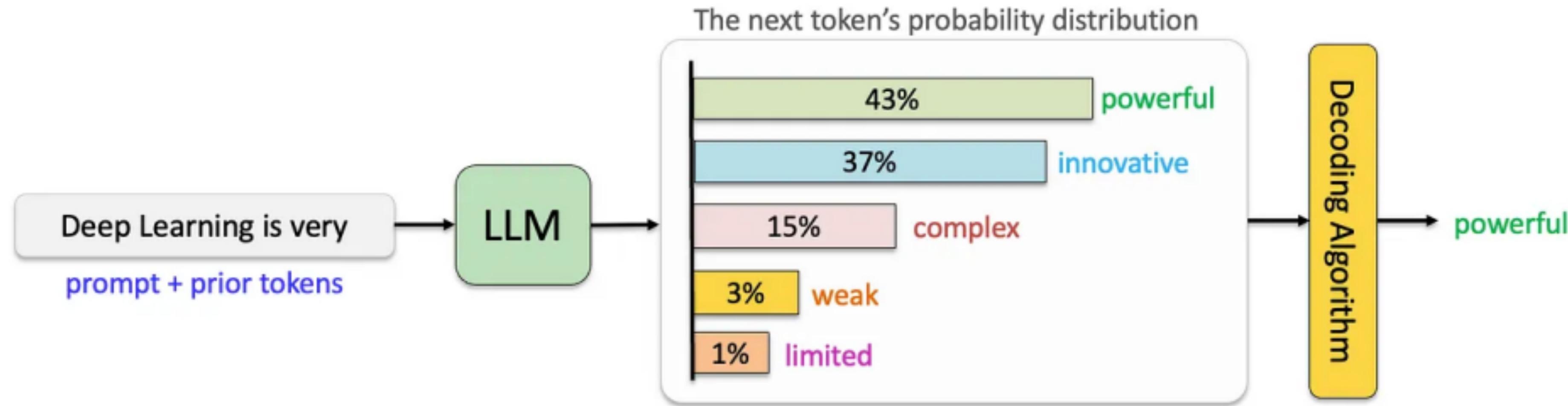
深度学习时代



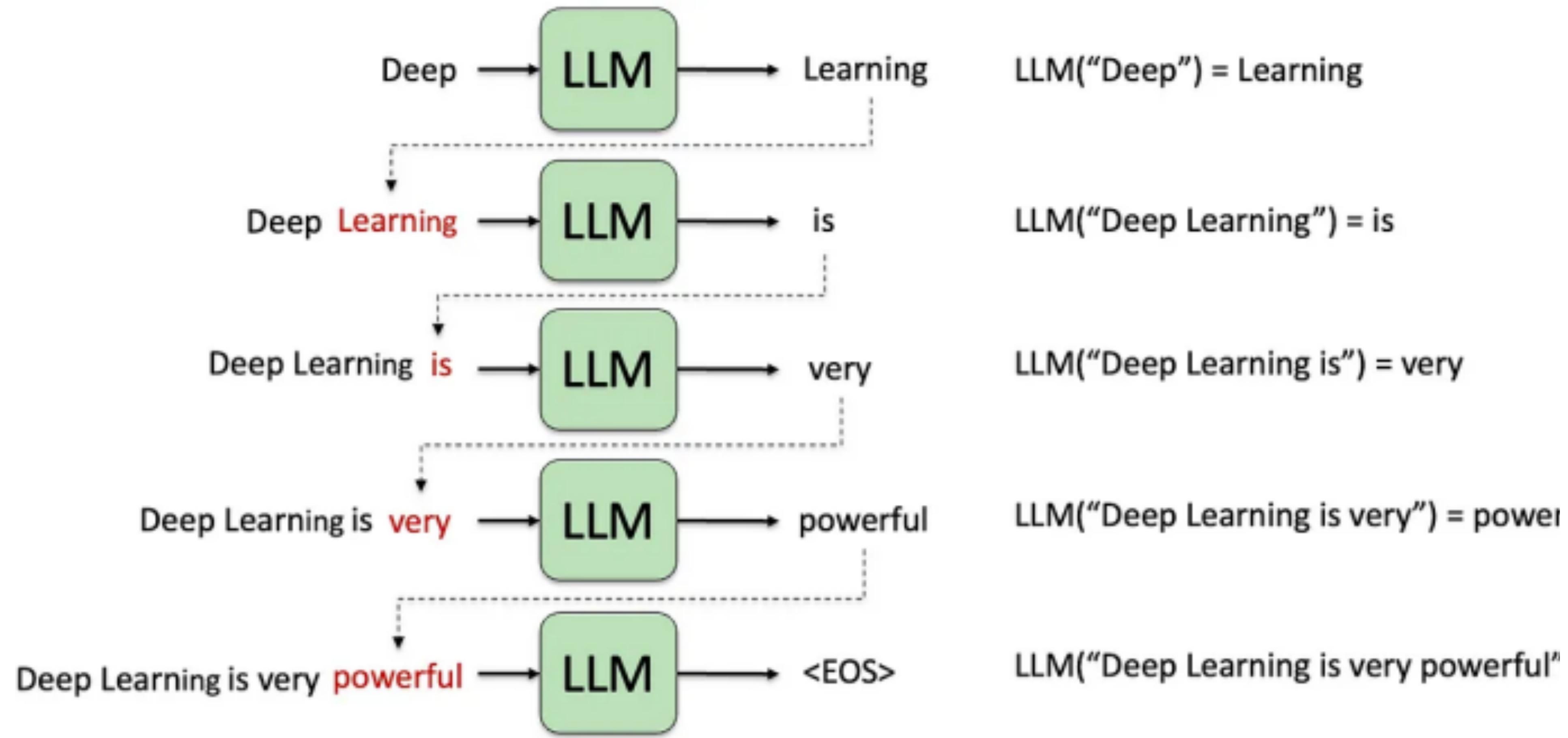
深度学习流程



语言模型时代

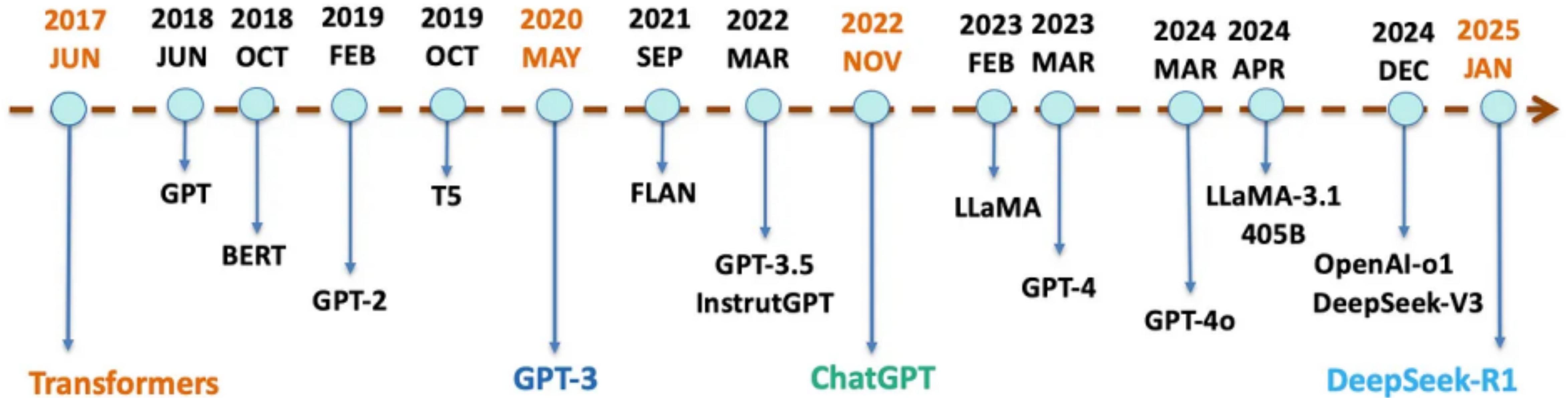


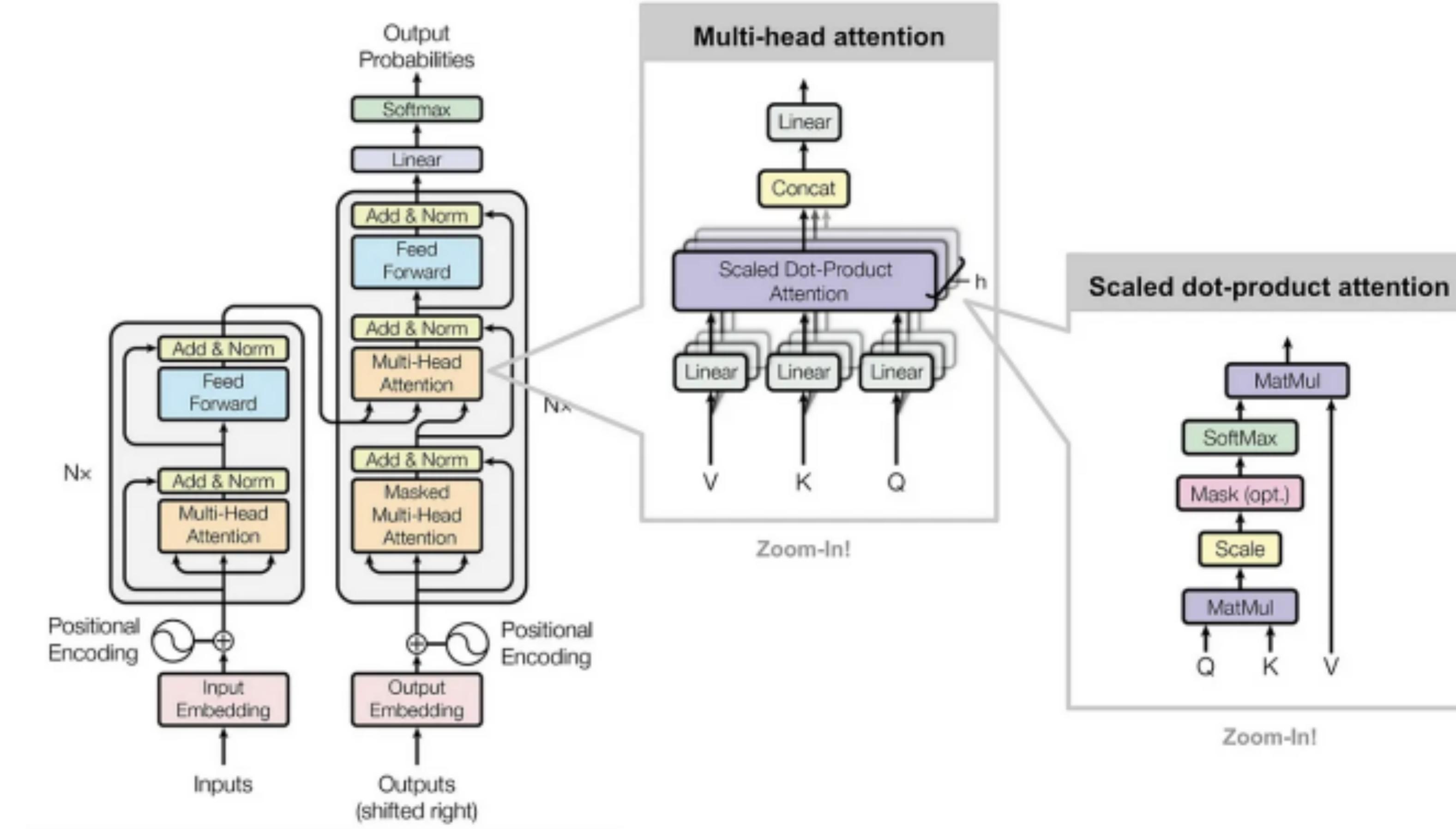
自回归语言模型时代 (Autoregressive Language Models)



文本生成

大语言模型简史

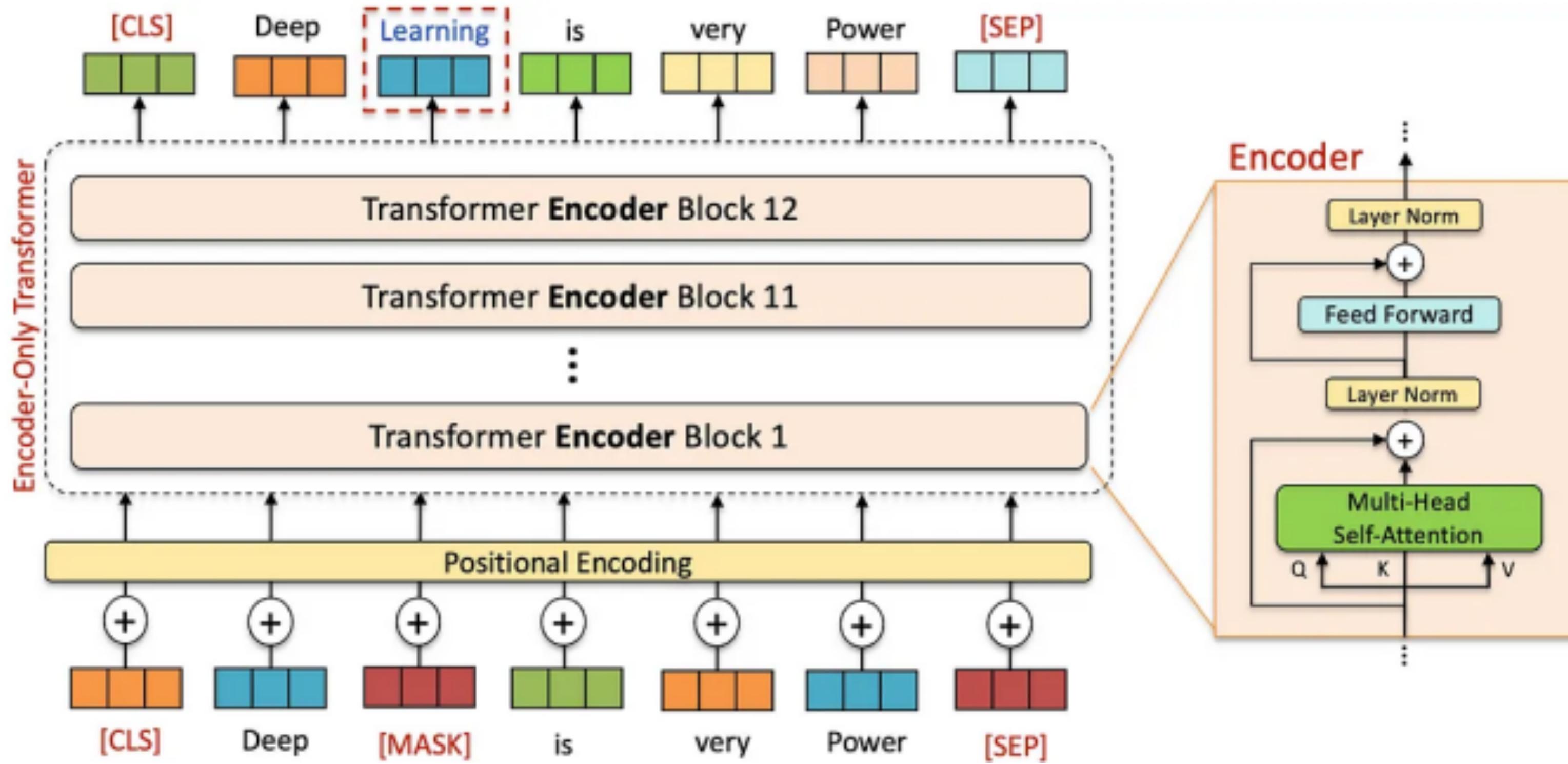




Transformer 革命 (2017)



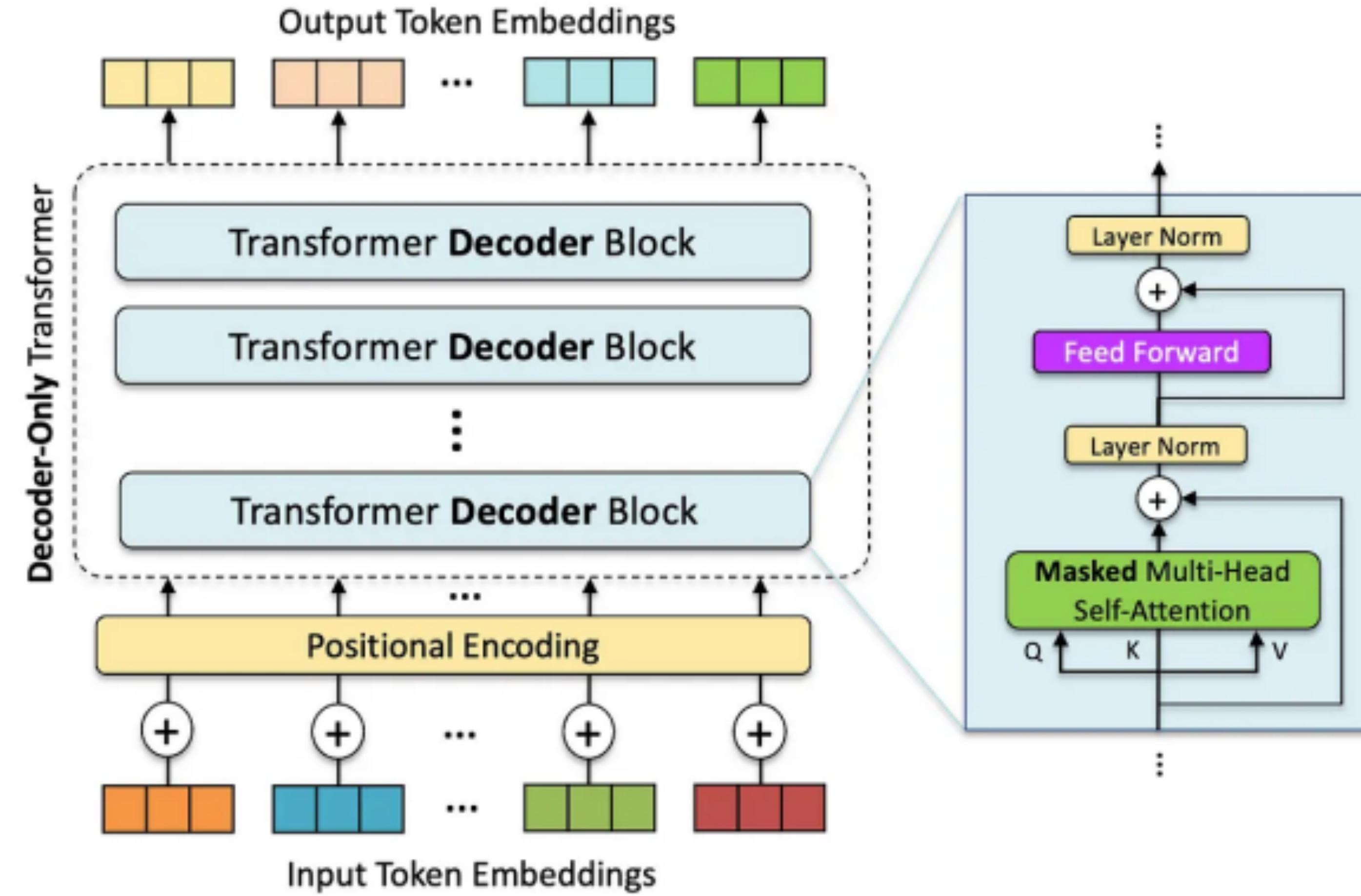
BERT



预训练Transformer模型时代 (2018–2020)
BERT：双向上下文理解 (2018)



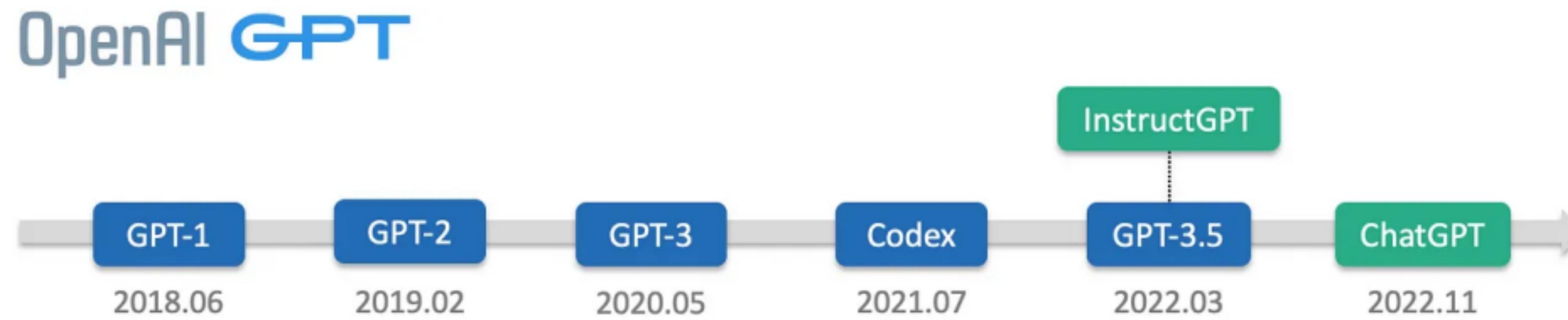
GPT



GPT: 生成式预训练和自回归文本生成 (2018–2020)

GPT (2018): 单向解码器Transformer建模

- 2018年6月，GPT系列的首个版本问世，首次引入了仅解码器的Transformer架构，参数规模达1.17亿，成为自然语言处理领域的重要突破。该模型通过单向自回归方法在4.6GB高质量文本数据上进行训练，能够基于先前的标记预测下一个标记。凭借对自然语言序列依赖关系的精准捕捉，它在文本补全、摘要生成和对话生成等生成任务中表现优异。尽管如此，GPT的设计初衷主要是通过监督微调（SFT）来适应不同的目标任务。



GPT-2 (2019): 扩展语言模型以实现广义零样本学习

- GPT-2于2019年2月发布，相较于前代实现了重大升级，参数量达到15亿，并在40GB多样化的互联网文本上进行训练。GPT-2引入了突破性的零样本学习能力，能够在不进行任务特定微调的情况下完成诸如撰写文章、回答问题、摘要生成和翻译等任务。

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



GPT-3 (2020): 大规模语言模型的崛起

- GPT-3于2020年6月发布，凭借其惊人的1750亿参数彻底改变了人工智能领域，成为当时规模最大的语言模型。它在庞大且多样化的文本语料库上进行训练，展现了卓越的少量样本和零样本学习能力，能够以极少甚至无需示例完成诸如撰写文章、生成代码、解决推理问题以及创意写作等任务。

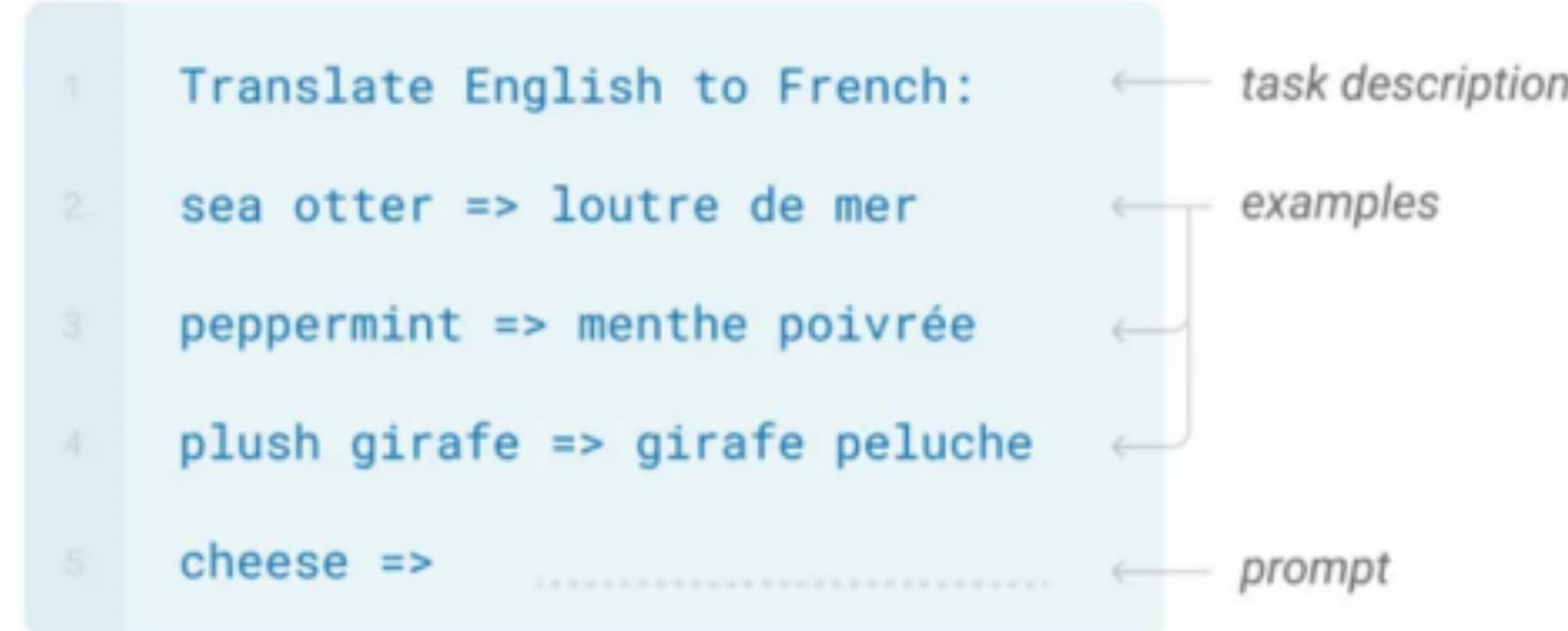
One-shot

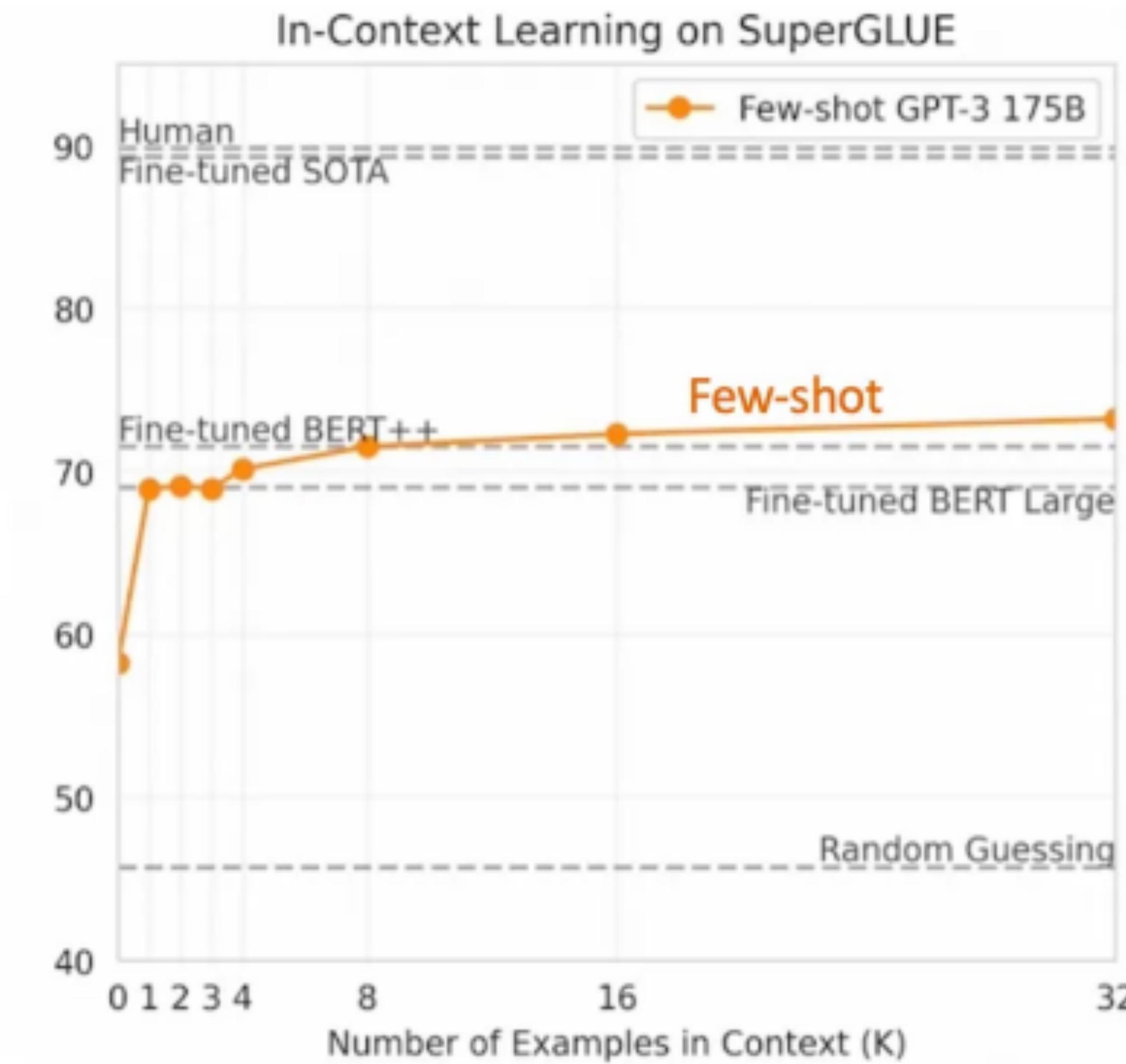
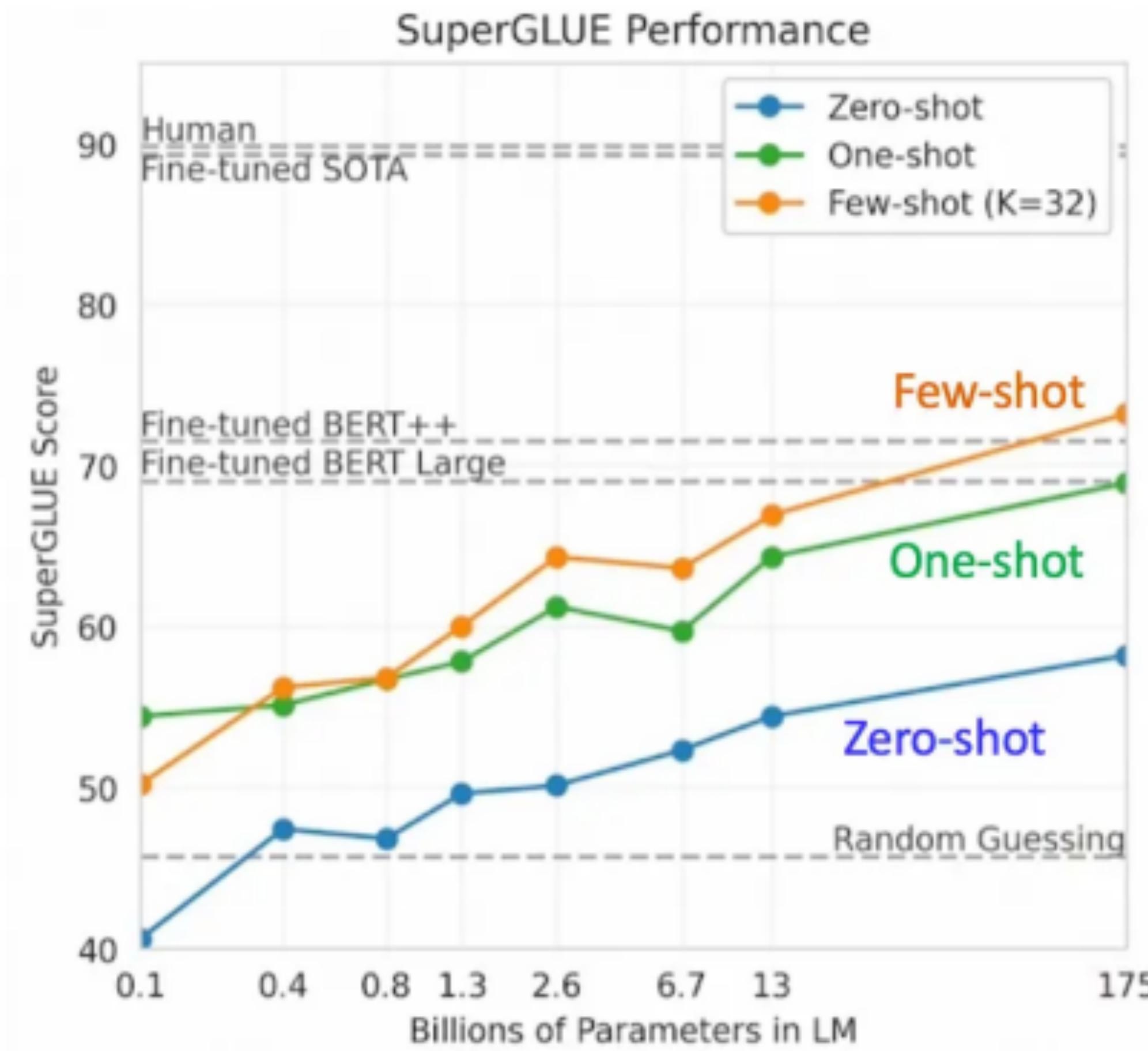
In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



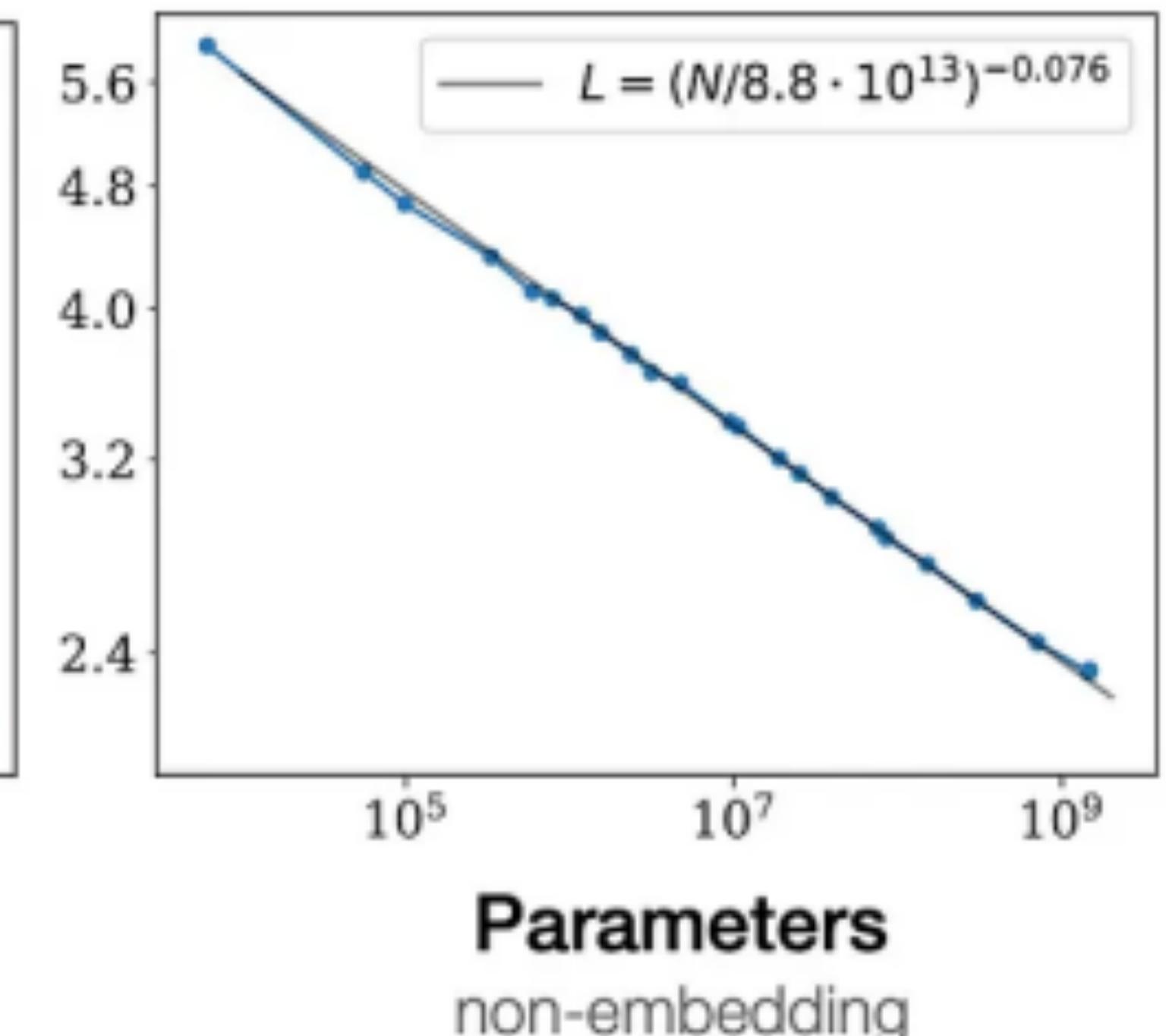
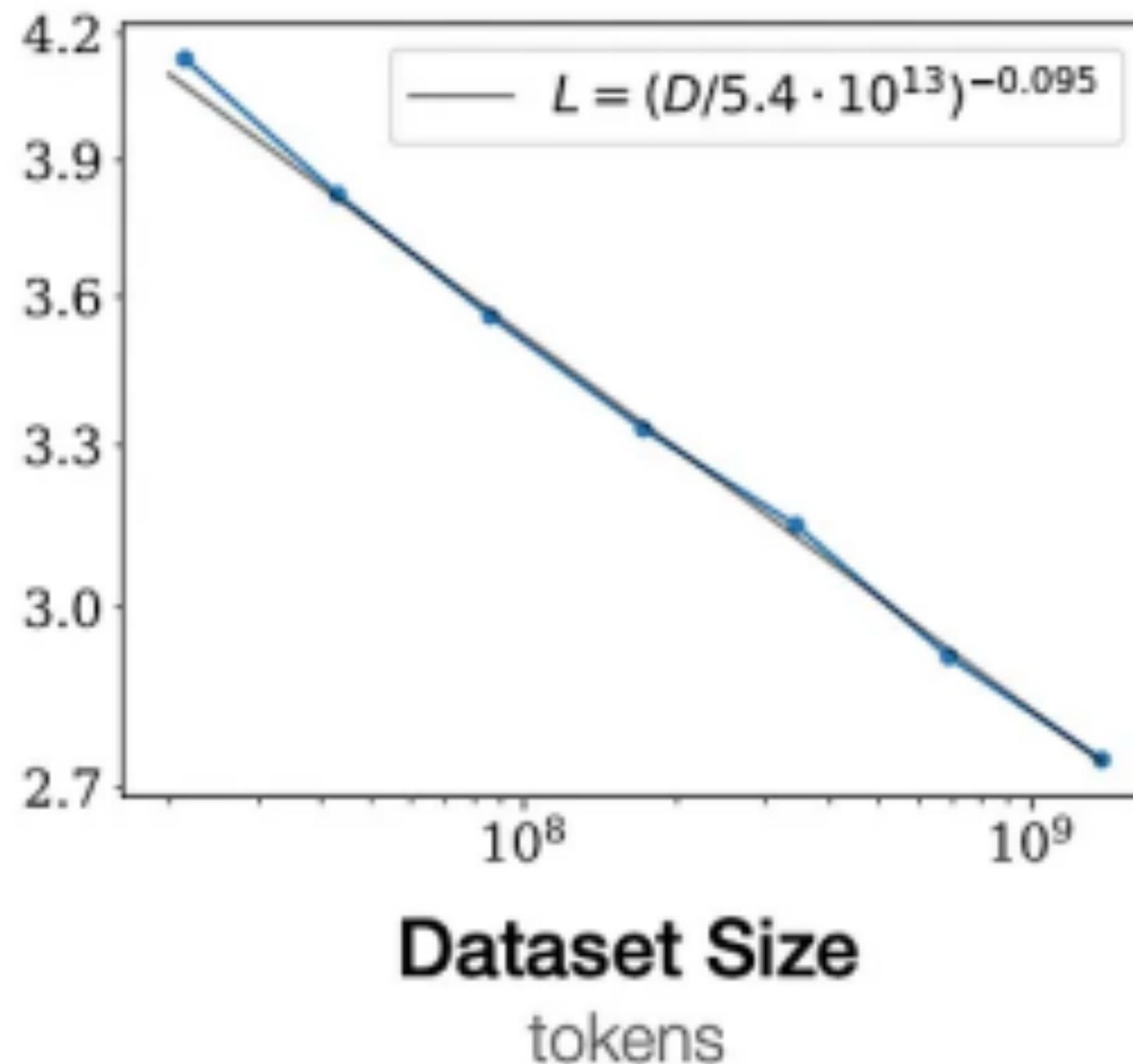
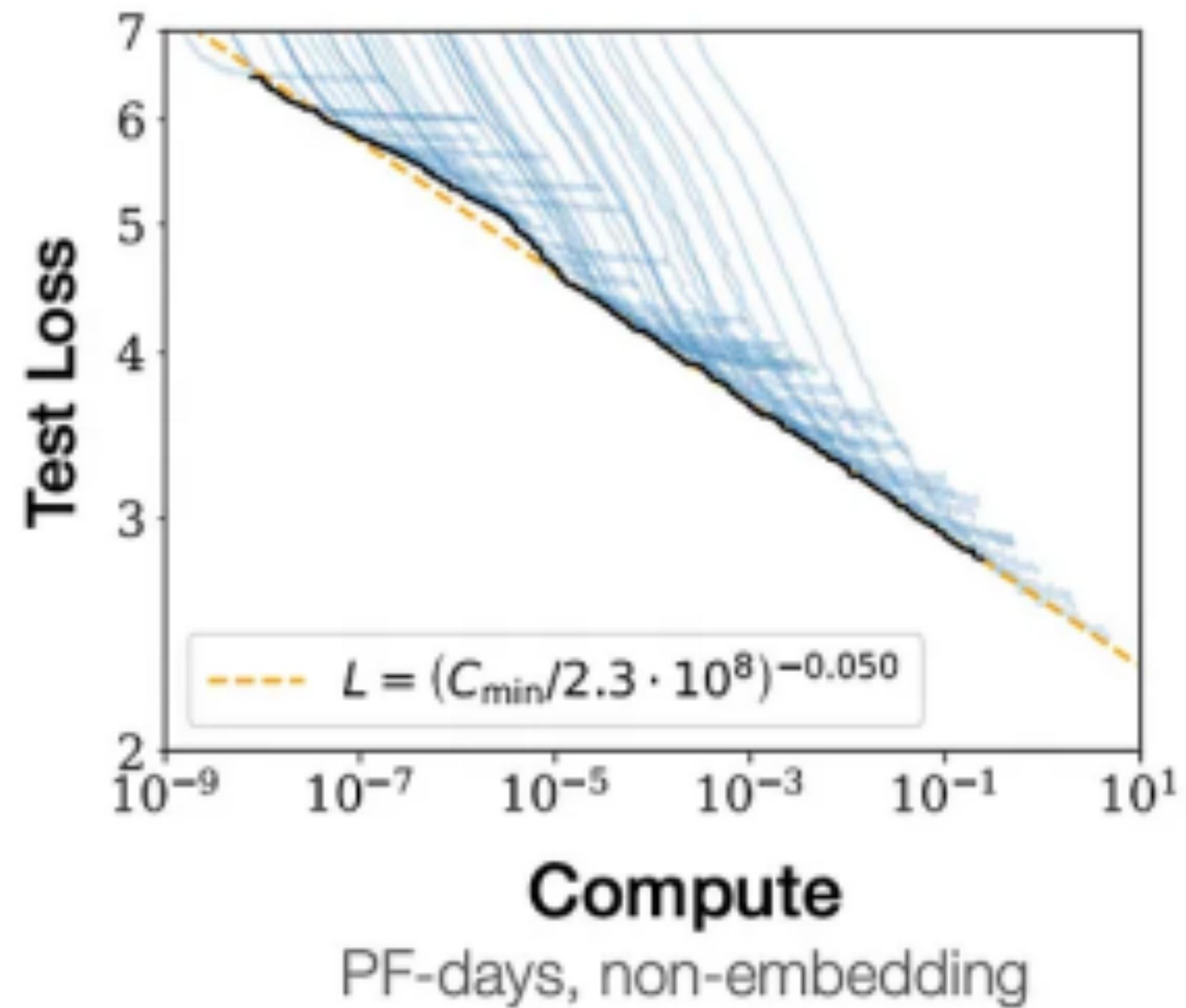
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

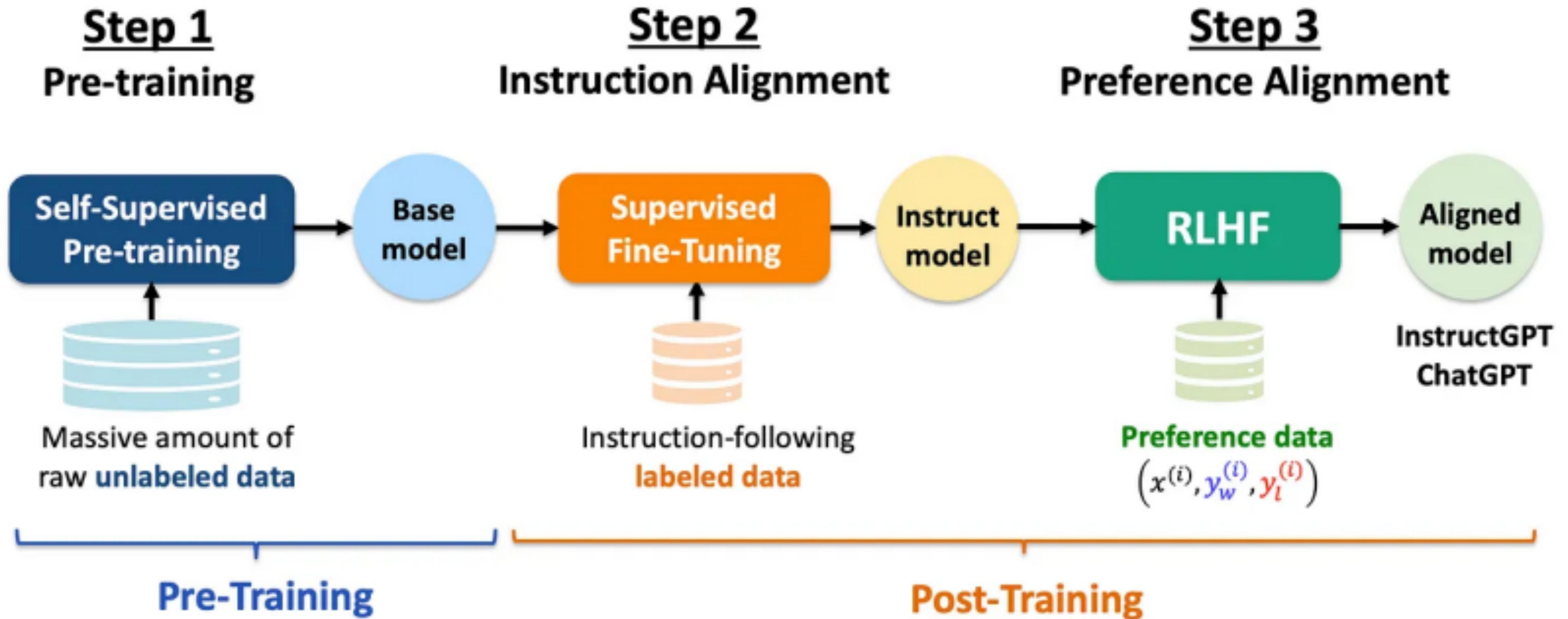




GPT-3重新定义了语言模型的可能性，为高级通用人工智能的发展铺平了道路



语言建模性能随着模型大小、数据集大小和训练使用的计算量的增加而平稳提升。



后训练对齐：弥合AI与人类价值观之间的差距 (2021–2022)

Instruction

Translate English into Simplified Chinese

Summarize in just 10 words to make the message even more brief and easier to remember.

Input Context (Optional)

Welcome to Hong Kong

The AAAI Conference on Artificial Intelligence, or AAAI, is a highly prestigious event organized by the Association for the Advancement of Artificial Intelligence. It gathers researchers, academics, and industry professionals globally to present and discuss the latest advancements, innovations, and applications in AI.

Response

欢迎来到香港

AAAI is a prestigious conference on artificial intelligence.

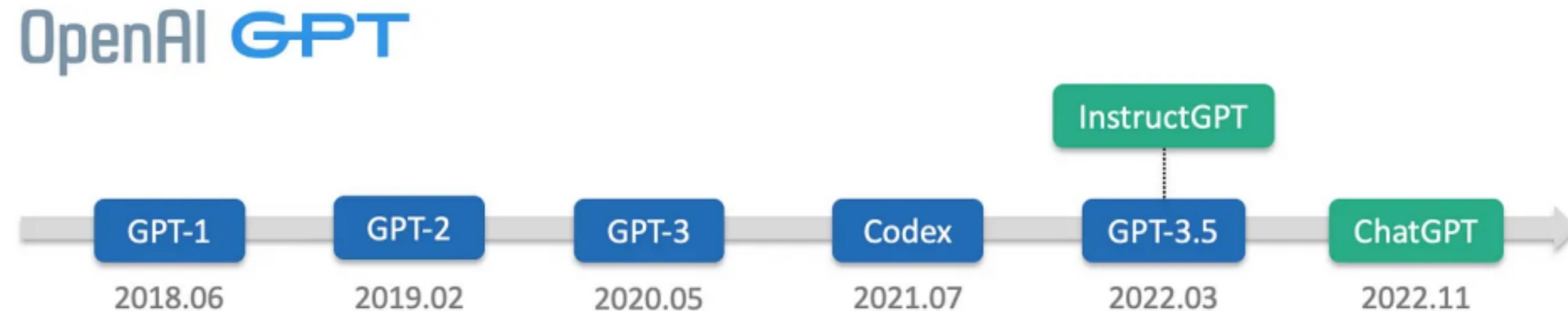
监督微调 (SFT)

基于人类反馈的强化学习 (RLHF)

- OpenAI在2022年引入的RLHF解决了SFT的可扩展性和性能限制。与需要人类编写完整输出的SFT不同，RLHF涉及根据质量对多个模型生成的输出进行排名。这种方法允许更高效的数据收集和标注，显著增强了可扩展性。
- RLHF过程包括两个关键阶段：
 - 训练奖励模型：人类注释者对模型生成的多个输出进行排名，创建一个偏好数据集。这些数据用于训练一个奖励模型，该模型学习根据人类反馈评估输出的质量。
 - 使用强化学习微调LLM：奖励模型使用近端策略优化（Proximal Policy Optimization - PPO）（一种强化学习算法）指导LLM的微调。通过迭代更新，模型学会了生成更符合人类偏好和期望的输出。
- 这个两阶段过程 — — 结合SFT和RLHF — — 使模型不仅能够准确遵循指令，还能适应新任务并持续改进。通过将人类反馈整合到训练循环中，RLHF显著增强了模型生成可靠、符合人类输出的能力，为AI对齐和性能设定了新标准。

ChatGPT：推进对话式AI (2022)

- 基于GPT-3.5和InstructGPT，OpenAI于2022年11月推出了ChatGPT，这是一种突破性的对话式AI模型，专门为自然的多轮对话进行了微调。ChatGPT的关键改进包括：
- 对话聚焦的微调：在大量对话数据集上进行训练，ChatGPT擅长维持对话的上下文和连贯性，实现更引人入胜和类似人类的互动。
- RLHF：通过整合RLHF，ChatGPT学会了生成不仅有用而且诚实和无害的响应。人类培训师根据质量对响应进行排名，使模型能够逐步改进其表现。
- ChatGPT的推出标志着AI的一个关键时刻，通常被称为「ChatGPT时刻」(ChatGPT moment)，因为它展示了对话式AI改变人机交互的潜力。



	预训练阶段	指令微调	奖励函数	强化学习
数据集合	原始数据 数 千 亿单词：图书、百科、网页等	标注用户指令 数万 用户指令和对应的答案	标注对比对 百 万标注对比对	用户指令 十 万用户指令
算法	语言模型预训练	语言模型预训练	二分类模型	强化学习方法
模型	基础模型	SFT 模型	RM 模型	RL 模型
资源需求	1000+GPU 月级别训练时间 GPT 3.0、LLaMa、PaLM	1-100GPU 天级别训练时间 MOSS、ChatGLM6b、Vicuna-13B等	1-100GPU 天级别训练时间	1-100GPU 天级别训练时间 ChatGPT、Claude

State of GPT, Microsoft Build 2023, Andrej Karpathy

27

LLM流程

LLM技术简介 -- 相关工程数据

- 训练
 - 训练费用：GPT-3 175B 单次训练 460 万美元 [3]。
 - 能耗：训练 PaLM 两个月左右耗费约了 3.4 Gwh [6]。
 - 数据集大小：GPT-3 175B 训练了 4990 亿个 token [16]。
 - OpenAI 训练集群：285k CPU, 10k high-end GPU。
- 推理
 - 推理时间
 - 最大 token 分别为 2、8 和 32 时，GPT-J 6B 模型的推理时间为 0.077s、0.203s 和 0.707s。
 - 最大 token 固定为 32 时，InstructGPT 模型（davinci v2）的推理时间为 1.969s。
 - API 延迟：OpenAI API 的平均延迟时间从几百毫秒到几秒不等。
 - InstructGPT davinci v2 (175B) 的理想去噪推理时间 0.21s/request。

Prompt:

Describir la imagen en 20 idiomas diferentes



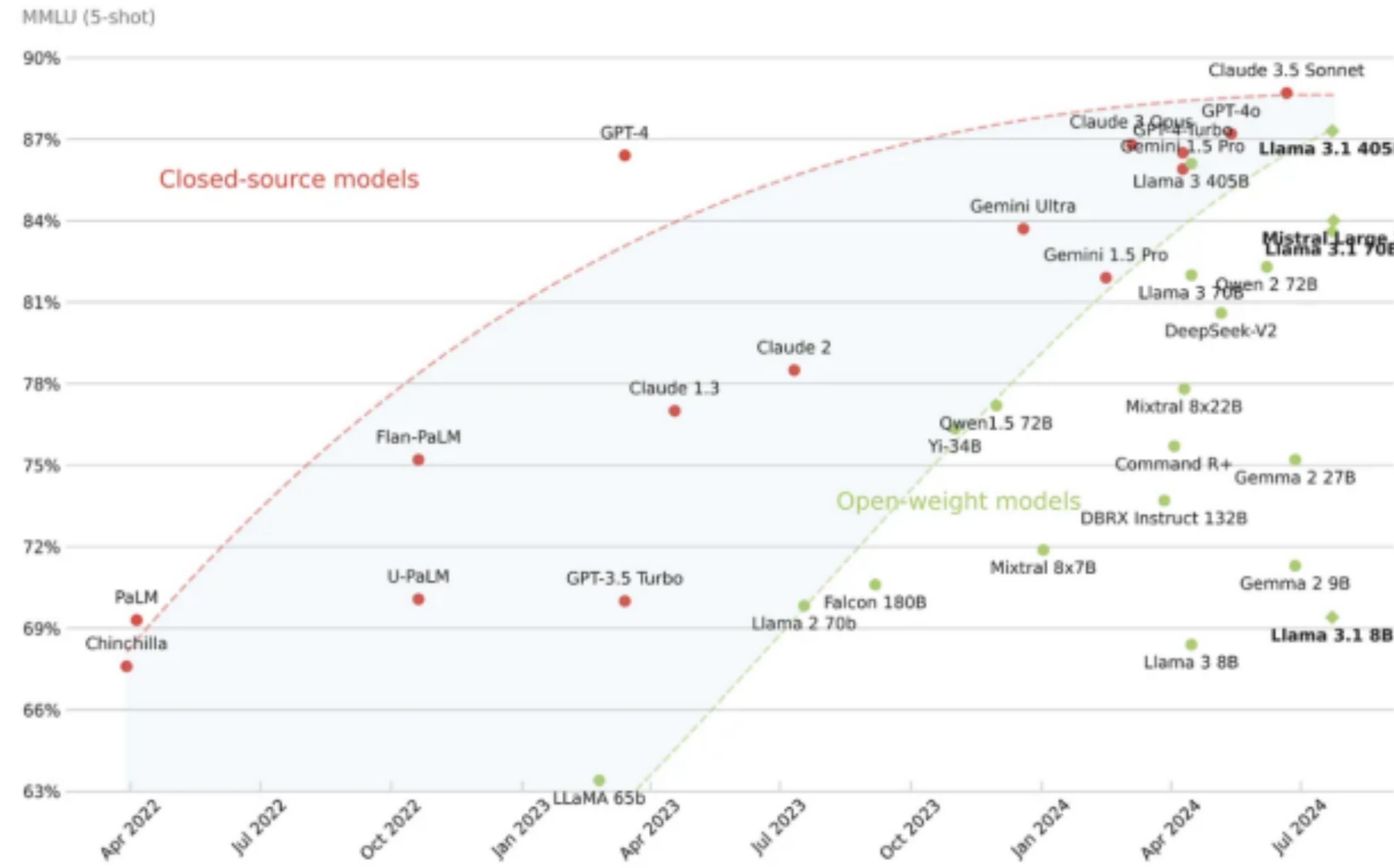
GPT-4V:

1. Inglés: This is a beautiful view of Machu Picchu in Peru.
2. Español: Esta es una hermosa vista de Machu Picchu en Perú.
3. Francés: C'est une belle vue de Machu Picchu au Pérou.

GPT-4V output continued

4. Alemán: Dies ist eine schöne Aussicht auf Machu Picchu in Peru.
5. Italiano: Questa è una bella vista di Machu Picchu in Perù.
6. Portugués: Esta é uma bela vista de Machu Picchu no Peru.
7. Russo: Это красивый вид на Мачу-Пикчу в Перу.
8. Chino: 这是秘鲁马丘比丘的美丽景色。
9. Japonés: これはペルーのマチュピチュの美しい景色です。
10. Coreano: 이것은 페루의 마추 피추의 아름다운 경치입니다.
11. Hindi: यह पेरु के माचू पिच्चू का सुंदर दृश्य है।
12. Árabe: هنا منظر جميل لمعابد بيتشو في بيرو .
13. Turco: Bu, Peru'daki Machu Picchu'nun güzel bir manzarasıdır.
14. Griego: Αυτή είναι μια όμορφη θέα του Μάτσου Πίτσου στο Περού.
15. Hebreo: זה נוף יפה של מאצ'י פיצ'ו בפרו.
16. Sueco: Det här är en vacker vy av Machu Picchu i Peru.
17. Noruego: Dette er en vakker utsikt over Machu Picchu i Peru.
18. Danés: Dette er en smuk udsigt over Machu Picchu i Peru.
19. Holandés: Dit is een prachtig uitzicht op Machu Picchu in Peru.
20. Polaco: To piękny widok na Machu Picchu w Peru.

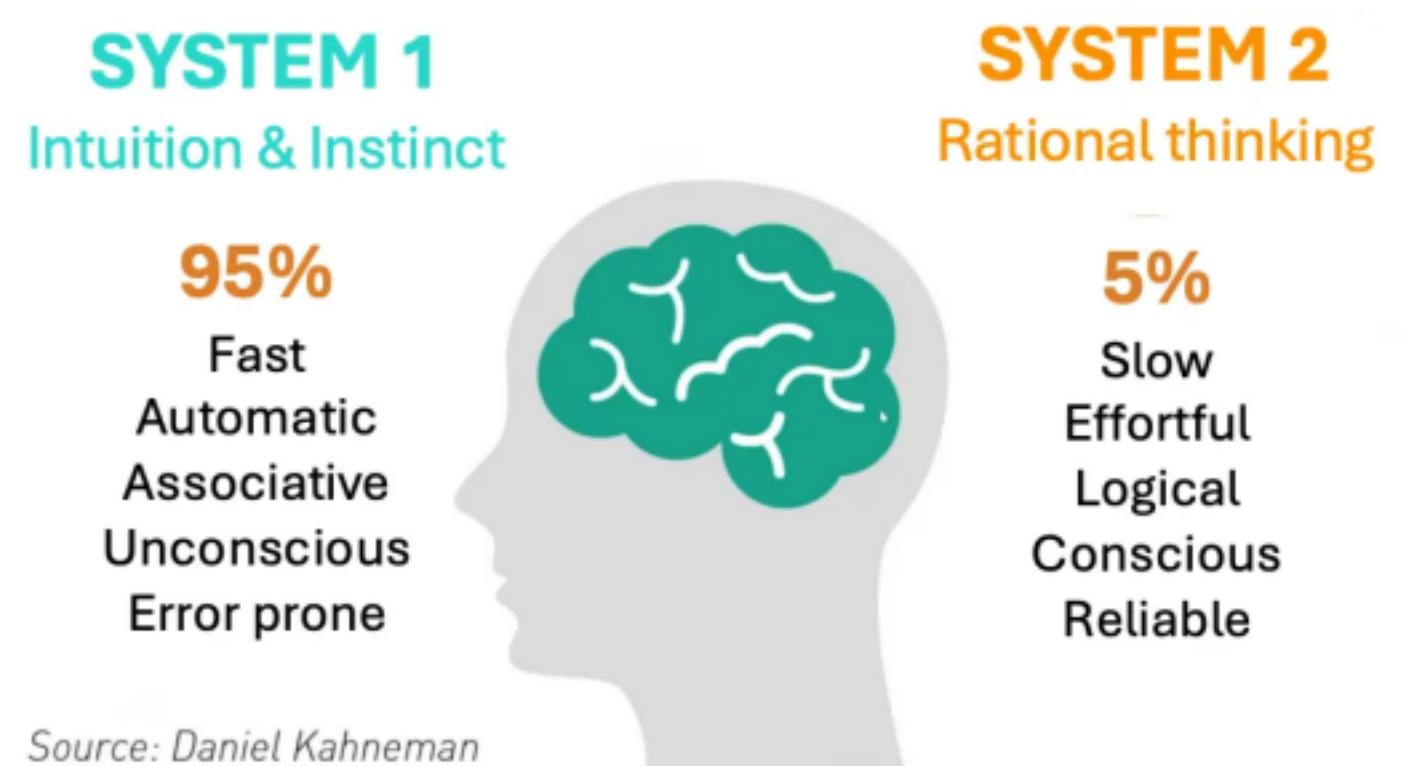
多模态模型：连接文本、图像及其他 (2023–2024)



开源和开放权重模型 (2023–2024)

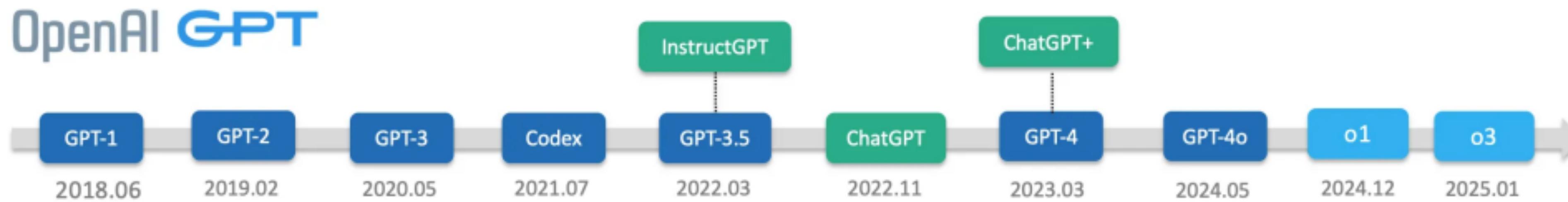
推理模型：从「系统1」到「系统2」思维的转变 (2024)

- 2024年，AI开发开始强调增强「推理」(Reasoning)，从简单的模式识别转向更逻辑化和结构化的思维过程。这一转变受到认知心理学双重过程理论的影响，区分了「系统1思维」（快速、直觉）和「系统2思维」（缓慢、分析）思维。虽然像GPT-3和GPT-4这样的早期模型在生成文本等「系统1」任务上表现出色，但在深度推理论和问题解决方面却有所欠缺。



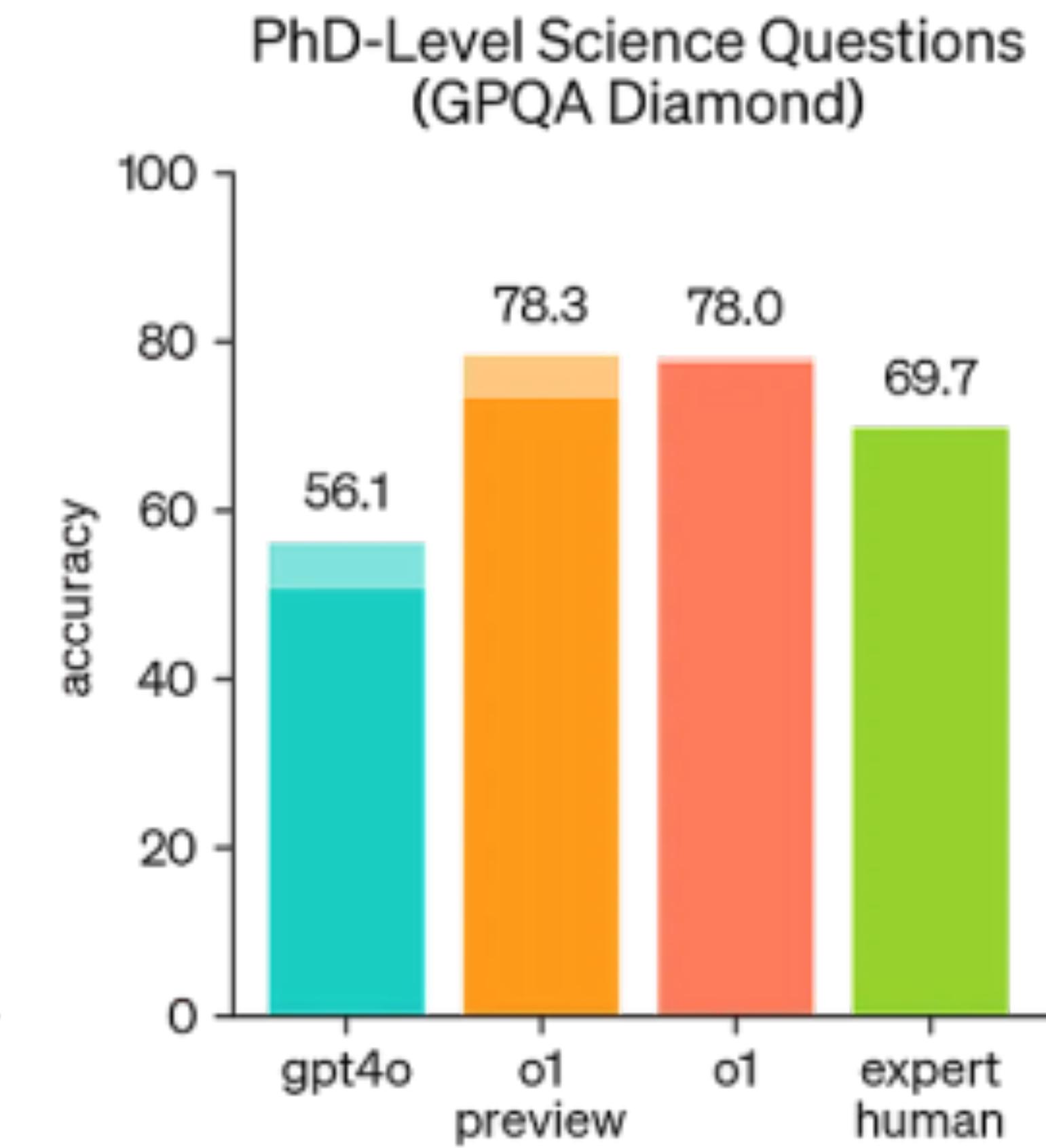
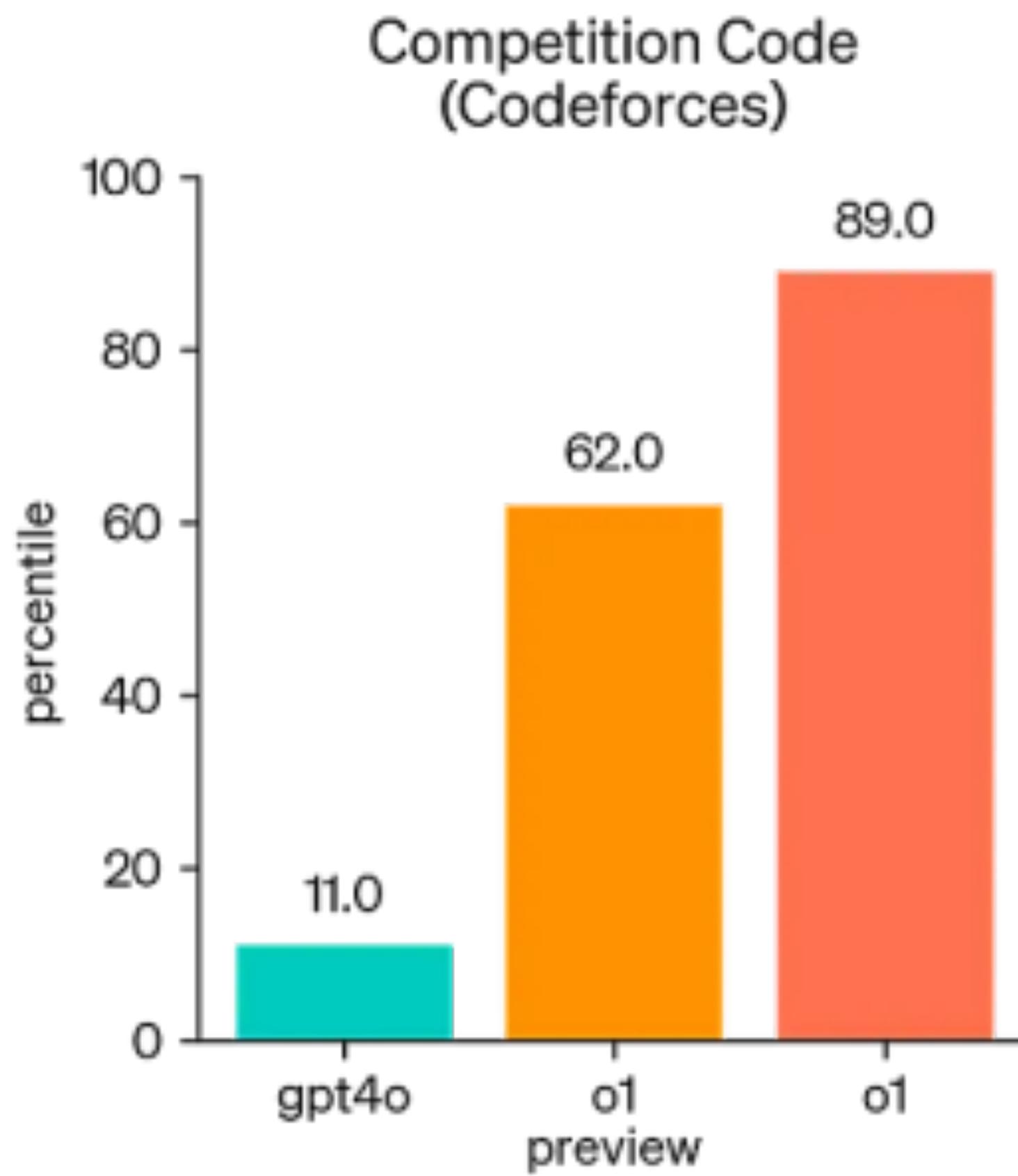
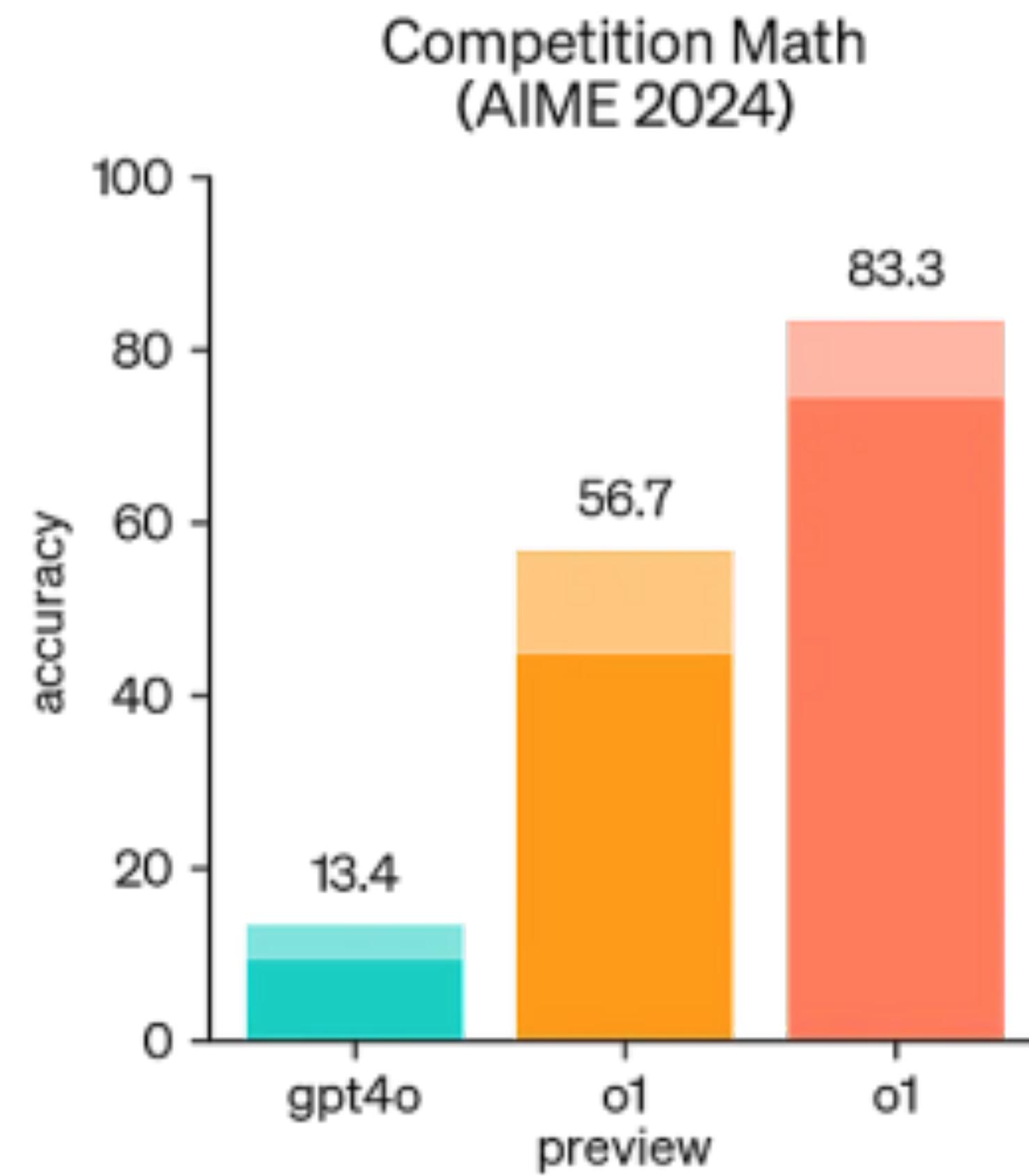
OpenAI-o1：推理能力的一大飞跃(2024)

- 2024年9月12日，OpenAI发布的o1-preview标志着人工智能能力的重大飞跃，尤其是在解决复杂推理任务（如数学和编程）方面。与传统LLMs不同，推理模型采用了「长链思维」（Long CoT）——即内部的推理轨迹，使模型能够通过分解问题、批判自己的解决方案并探索替代方案来“思考”问题。这些CoTs对用户是隐藏的，用户看到的是一个总结性的输出。

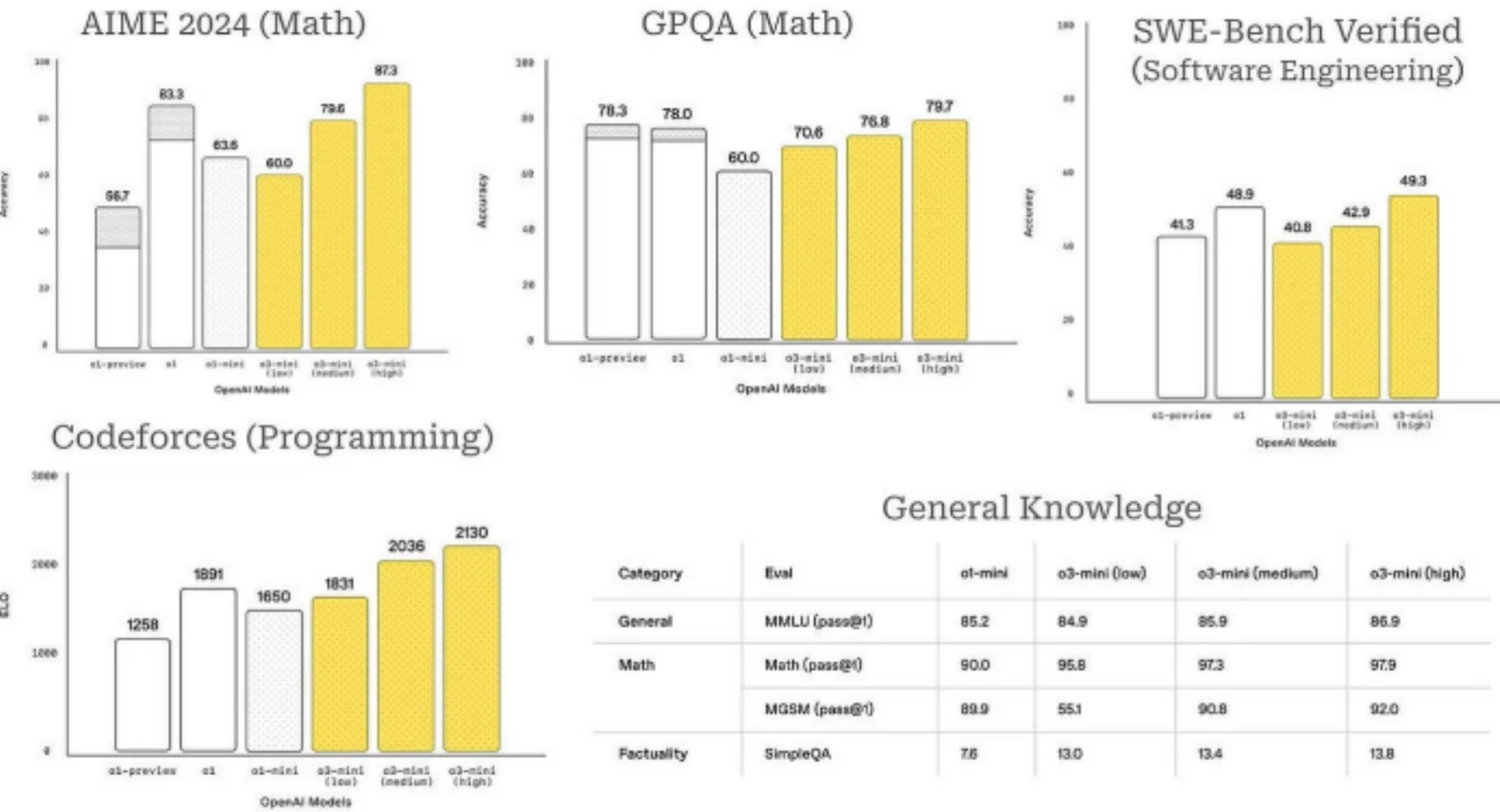


推理模型

- 推理模型的关键特性包括：
- 长链思维（Long CoT）：使模型能够将复杂问题分解为更小的部分，批判性地评估其解决方案，并探索多种方法，类似于搜索算法。
- 推理时计算控制：对于更复杂的问题，可以生成更长的CoTs；而对于较简单的问题，则使用较短的CoTs以节省计算资源。
- 增强的推理能力：尽管像o1-preview这样的初始推理模型在某些领域的表现不如标准LLMs，但在推理任务中，它们的表现远远超越了后者，常常能与人类专家媲美。例如，o1-preview在数学（AIME 2024）、编程（CodeForces）和博士级别的科学问题上均超越了GPT-4o。



人机对比



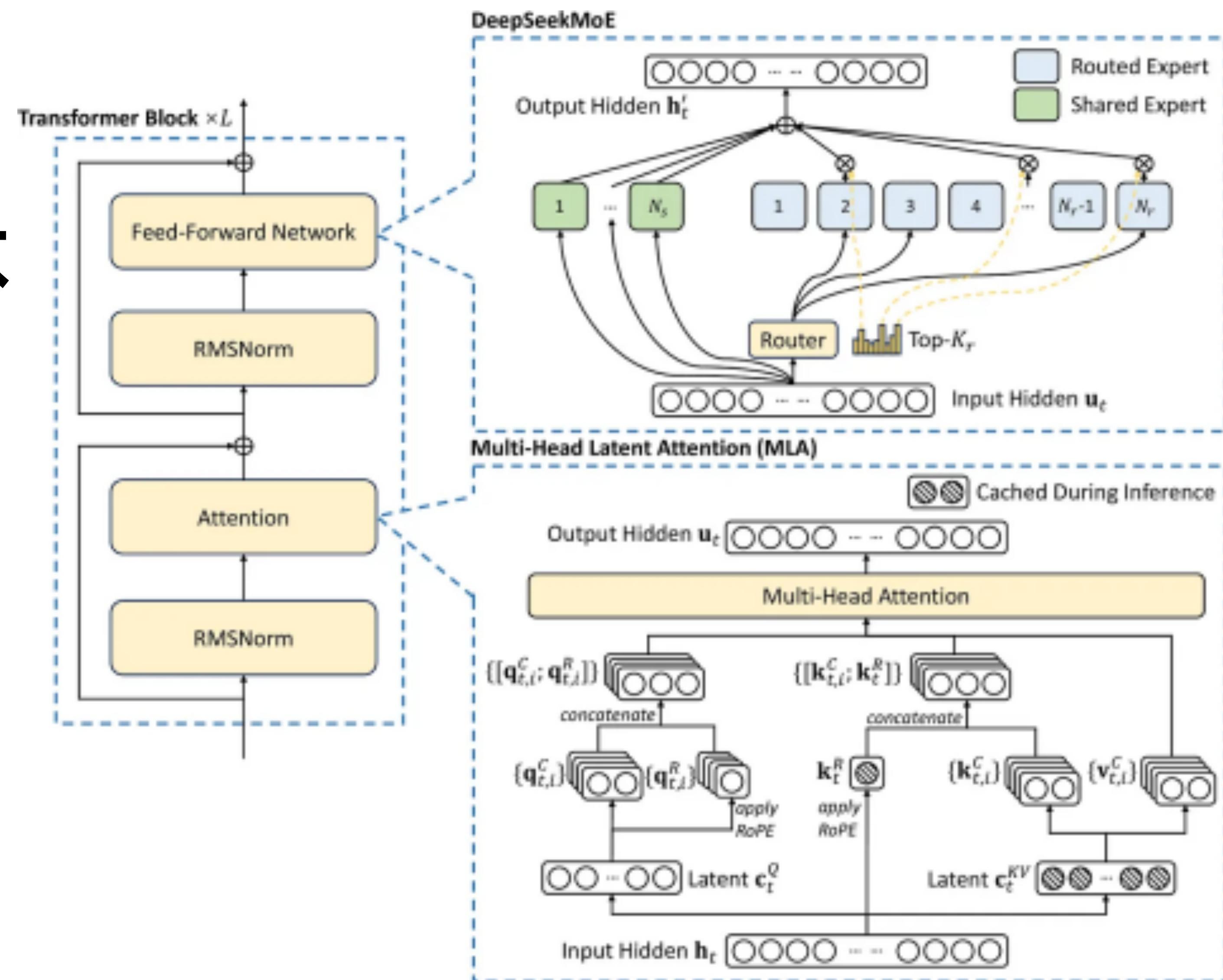
人机对比

成本高效的推理模型：DeepSeek-R1 (2025)

DeepSeek-V3 (2024-12)

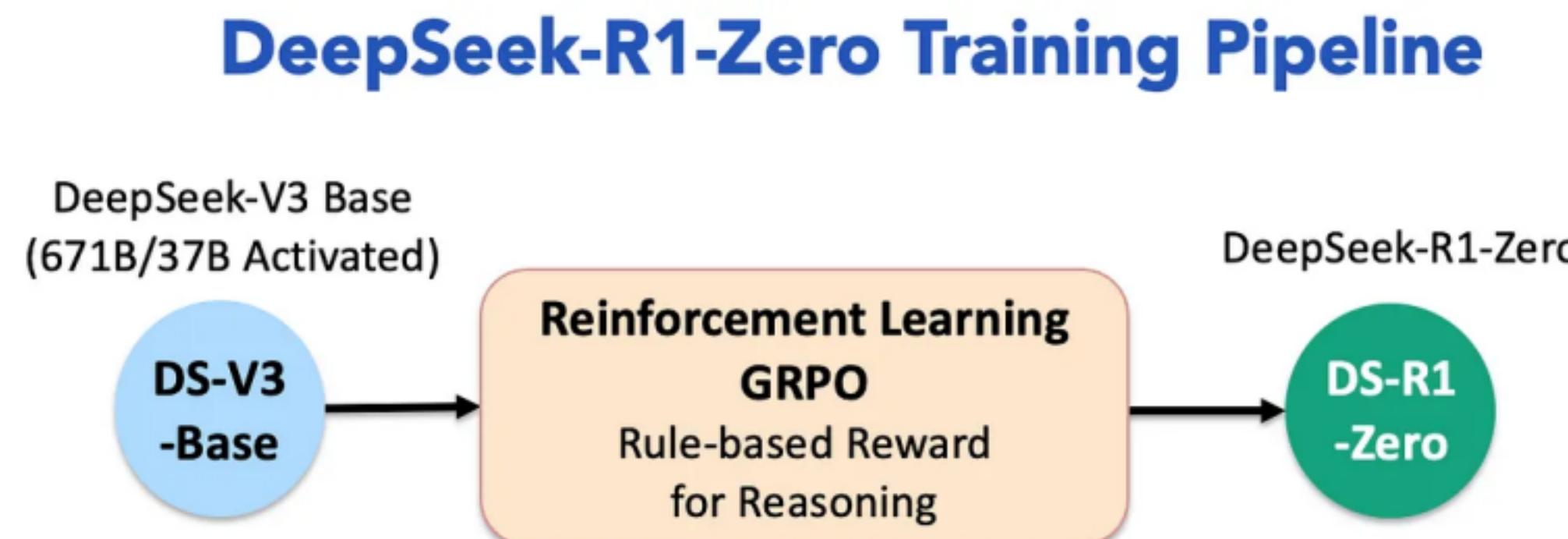
- 2024年12月下旬，「深度求索-V3」(DeepSeek-V3)作为一种成本高效的开放权重LLM出现，为AI的可访问性设定了新标准。DeepSeek-V3与OpenAI的ChatGPT等顶级解决方案相媲美，但开发成本显著降低，估计约为560万美元，仅为西方公司投资的一小部分。该模型最多包含6710亿个参数，其中370亿个活跃参数，并采用专家混合(MoE)架构，将模型划分为专门处理数学和编码等任务的组件，以减轻训练负担。DeepSeek-V3采用了工程效率，例如改进Key-Value缓存管理和进一步推动专家混合方法。该模型引入了三个关键架构：
 - 多头潜在注意力 (Multi-head Latent Attention – MLA)：通过压缩注意力键和值来减少内存使用，同时保持性能，并通过旋转位置嵌入 (RoPE) 增强位置信息。
 - DeepSeek专家混合 (DeepSeekMoE)：在前馈网络 (FFNs) 中采用共享和路由专家的混合，以提高效率并平衡专家利用率。
 - 多标记预测 (Multi-Token Prediction – MTP)：增强模型生成连贯且上下文相关的输出的能力，特别是对于需要复杂序列生成的任务。

DeepSeek-V3技术 架构



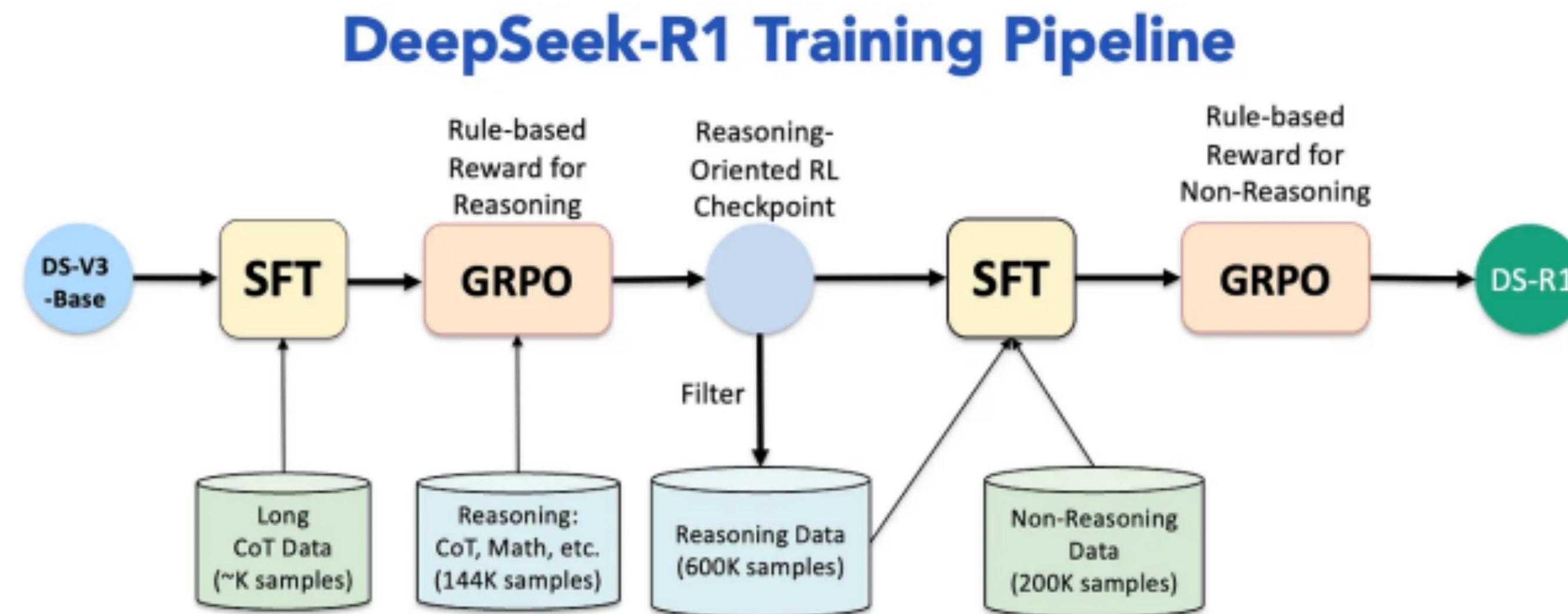
DeepSeek-R1-Zero 和 DeepSeek-R1 (2025-01)

- DeepSeek-R1-Zero：一种基于DeepSeek-V3的推理模型，通过强化学习 (RL) 增强其推理能力。它完全消除了「监督微调」(SFT)阶段，直接从名为DeepSeek-V3-Base的预训练模型开始。它采用了一种基于「规则的强化学习方法」(Rule-based Reinforcement Learning)，称为「组相对策略优化」(Group Relative Policy Optimization – GRPO)，根据预定义规则计算奖励，使训练过程更简单且更具可扩展性。



DeepSeek-R1

- DeepSeek-R1：为了解决DeepSeek-R1-Zero的局限性，如低可读性和语言混杂，DeepSeek-R1纳入了一组有限的高质量冷启动数据和额外的RL训练。该模型经历了多个微调和RL阶段，包括拒绝采样和第二轮RL训练，以提高其通用能力和与人类偏好的一致性。



Model Name	Base Model	Total Parameters
DeepSeek-R1-Distill-Qwen-1.5B	Qwen2.5-Math-1.5B	1.5 billion
DeepSeek-R1-Distill-Qwen-7B	Qwen2.5-Math-7B	7 billion
DeepSeek-R1-Distill-Llama-8B	Llama-3.1-8B	8 billion
DeepSeek-R1-Distill-Qwen-14B	Qwen2.5-14B	14 billion
DeepSeek-R1-Distill-Qwen-32B	Qwen2.5-32B	32 billion
DeepSeek-R1-Distill-Llama-70B	Llama-3.3-70B-Instruct	70 billion

DeepSeek 蒸馏 DeepSeek 模型

Qwen3-Max 2025年9月

- 据阿里介绍，Qwen3-Max 模型总参数超过 1T，预训练使用了 36T tokens。模型结构沿用了 Qwen3 系列的模型结构设计范式，使用了 global-batch load balancing loss。
 - 训练稳定性：得益于 Qwen3 的 MoE 模型结构设计，Qwen3-Max 的预训练 loss 稳定平滑。训练过程一气呵成，没有任何 loss 尖刺，也没有使用训练回退、改变数据分布等调整策略。
 - 训练高效性：在 PAI-FlashMoE 的高效多级流水并行策略优化下，Qwen3-Max-Base 训练效率显著提升，其 MFU 相比 Qwen2.5-Max-Base 相对提升 30%。在长序列训练场景中，进一步使用 ChunkFlow 策略获得了相比序列并行方案提升 3 倍的吞吐收益，支持 Qwen3-Max 1M 长上下文的训练。同时，通过 SanityCheck、EasyCheckpoint、调度链路优化等多种手段，Qwen3-Max 在超大规模集群上因硬件故障造成的时间损失下降为 Qwen2.5-Max 的五分之一。
- 据介绍，Qwen3-Max-Instruct 的预览版已在 LMArena 文本排行榜上稳居全球前三（超越 GPT-5-Chat）。正式发布版本进一步提升了其能力，尤其在代码生成与智能体表现方面表现卓越。
 - 在专注于解决现实编程挑战的基准测试 SWE-Bench Verified 上，Qwen3-Max-Instruct 取得了高达 69.6 分的优异成绩，稳居全球顶尖模型之列。
 - 此外，在评估智能体工具调用能力的严苛基准 Tau2-Bench 上，Qwen3-Max-Instruct 更是实现了突破性表现，以 74.8 分超越 Claude Opus 4 与 DeepSeek-V3.1。

二、智能体的基本概念

智能体 — Google

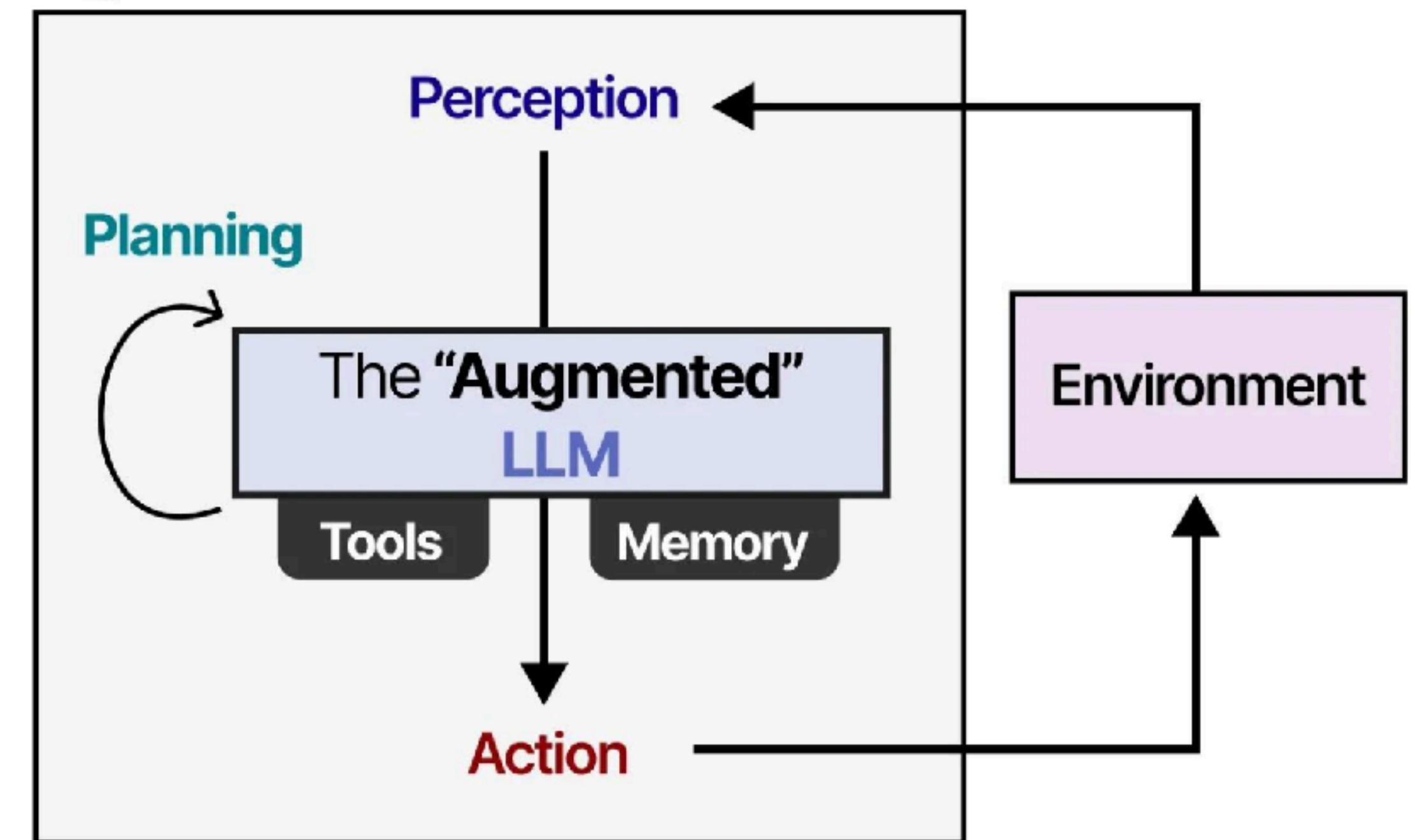
AI 智能体是使用 AI 来实现目标并代表用户完成任务的软件系统。其表现出了推理、规划和记忆能力，并且具有一定的自主性，能够自主学习、适应和做出决定。

这些功能在很大程度上得益于生成式 AI 和 AI 基础模型的多模态功能。AI 智能体可以同时处理文本、语音、视频、音频、代码等多模态信息；可以进行对话、推理、学习和决策。它们可以随着时间的推移不断学习，并简化事务和业务流程。智能体可以与其他智能体协作，来协调和执行更复杂的工作流。

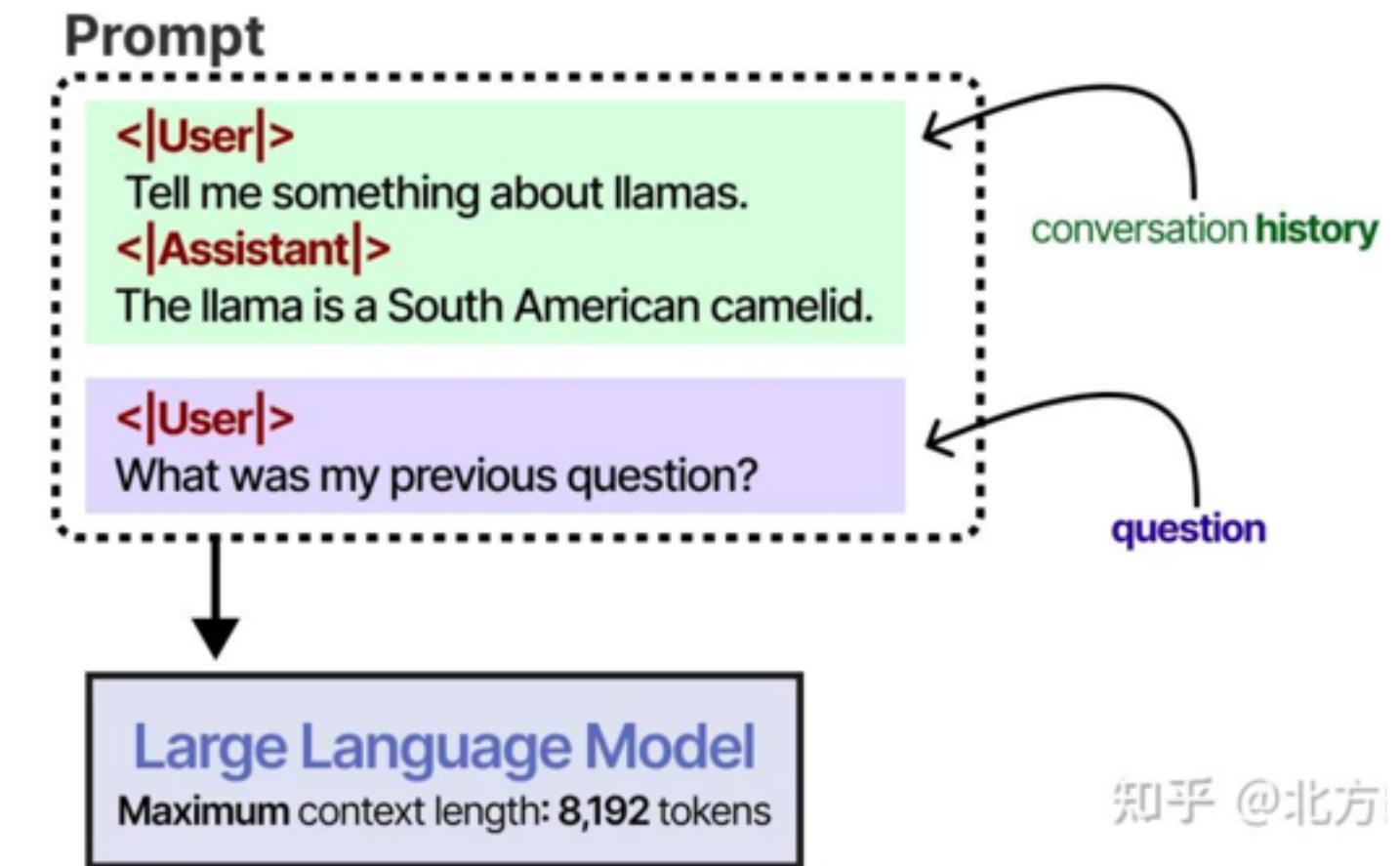
如上所述，虽然 AI 智能体的主要功能是推理和执行（如 [ReAct Framework](#) 中所述），但随着时间的推移，更多功能也随之发展起来。

- **推理：**此核心认知过程涉及使用逻辑和可用信息来得出结论、进行推断及解决问题。具有强大推理能力的 AI 智能体可以分析数据、识别模式，并根据证据和上下文做出明智的决策。
- **行动：**根据决策、计划或外部输入采取行动或执行任务的能力对于 AI 智能体与其环境进行互动和实现目标至关重要。这可能包括具身 AI 的物理动作，或发送消息、更新数据或触发其他流程等数字操作。
- **观察：**通过感知或感应收集有关环境或情况的信息，对于 AI 智能体了解上下文并做出明智的决策至关重要。这可能涉及各种形式的感知，例如计算机视觉、自然语言处理或传感器数据分析。
- **规划：**制定实现目标的战略计划是智能行为的关键方面。具有规划能力的 AI 智能体可以确定必要的步骤、评估潜在行动，并根据可用信息和预期结果选择最佳行动方案。这通常涉及预测未来状态和考虑潜在障碍。
- **协作：**在复杂且动态的环境中，与他人（无论是人类还是其他 AI 智能体）有效协作来实现共同目标变得越来越重要。协作需要沟通、协调以及理解和尊重他人观点的能力。
- **自我完善：**自我改进和自适应能力是高级 AI 系统的标志。具有自我完善能力的 AI 智能体可以从经验中学习，根据反馈调整行为，并随着时间的推移不断提升性能和能力。这可能涉及机器学习技术、优化算法或其他形式的自行修改。

Agent

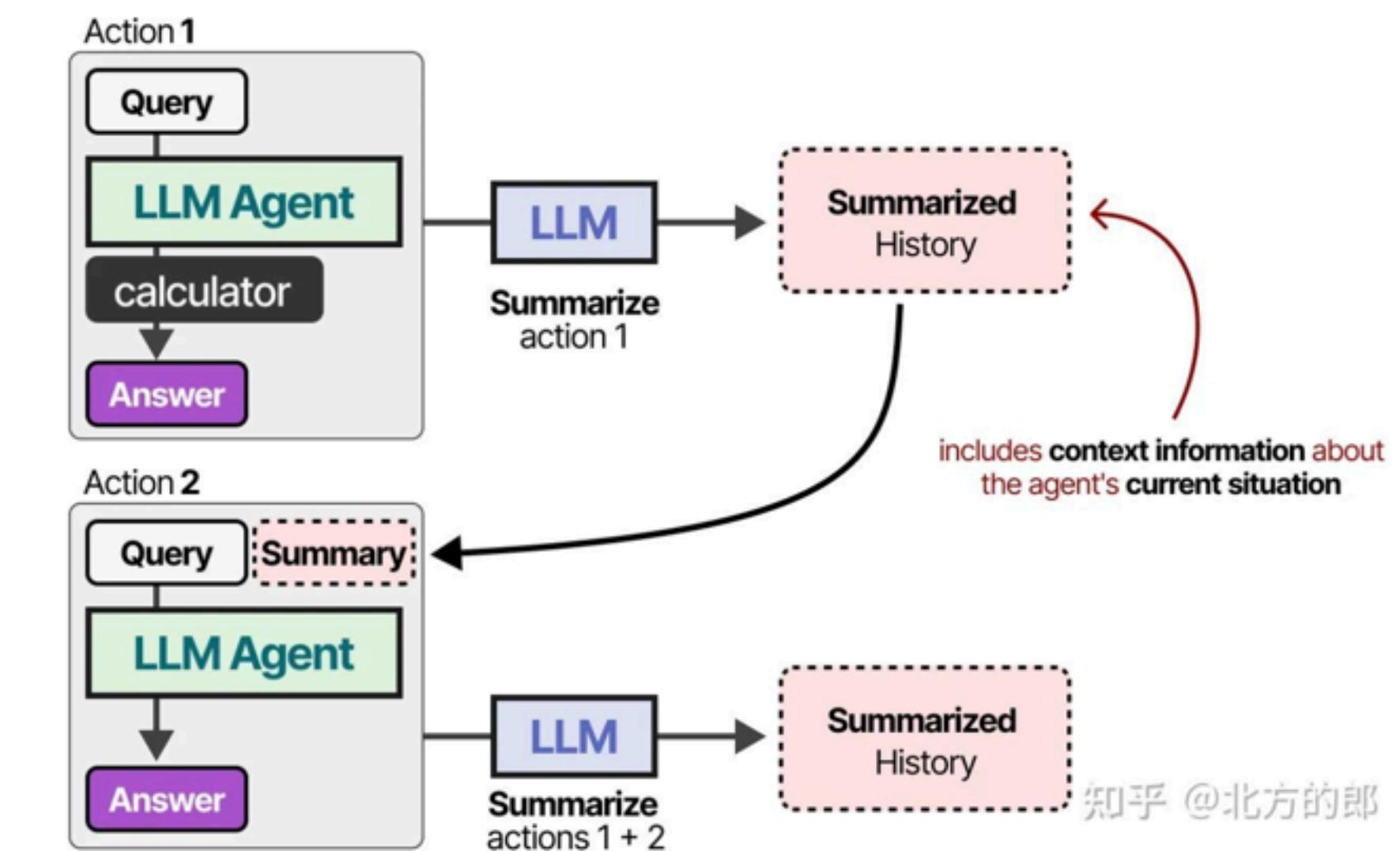


Memory



知乎 @北方

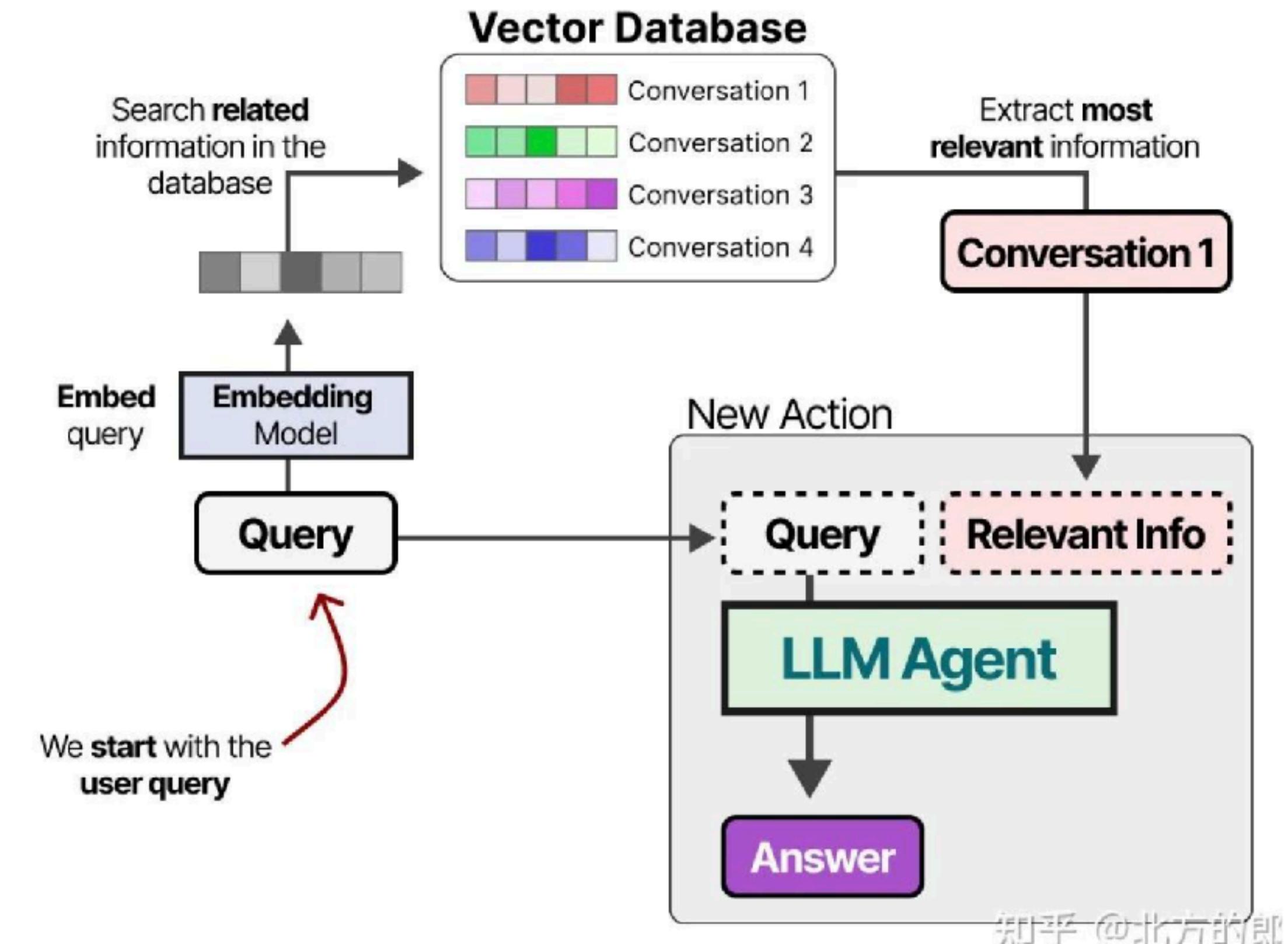
- Short-term memory
- Long-term memory



知乎 @北方的郎

Memory

- Short-term memory
- Long-term memory



Tools

- Search
- API
- MCP

Few-shot prompt per tool ([Calculator](#))

You can add calls to a calculator API. You can call the API by writing “[Calculator(formula)]”.

Input: 5 times 3 is 15.

Output: 5 times 3 is [\[Calculator\(5*3\)\]](#).

Input: 12 divided by 6 is 2.

Output: 12 divided by 6 is [\[Calculator\(6/2\)\]](#).

Input: 5 times 3 is 15.

LLM

Output: 5 times 3 is [\[Multiply\(5-3\)\]](#). ←

Output: 5 times 3 is [\[Calculator\(5*3\)\]](#).

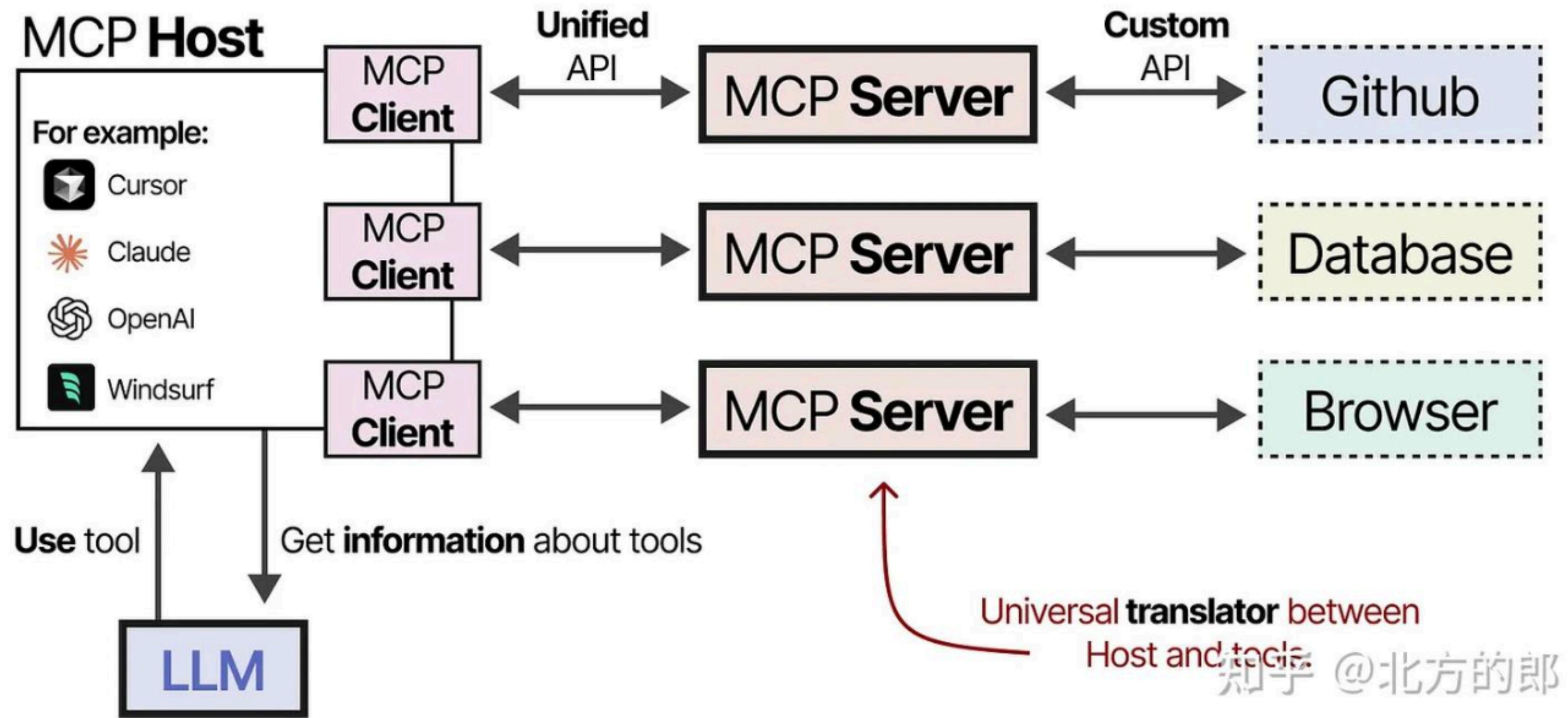
Output: 5 times 3 is [\[Calculator\(5/12\)\]](#).

examples
(manually created)

data
(questions from
large datasets)

filtered
(the best outputs are selected
based on correctness of tool use,
output, and loss decrease)

知乎 @北方的郎

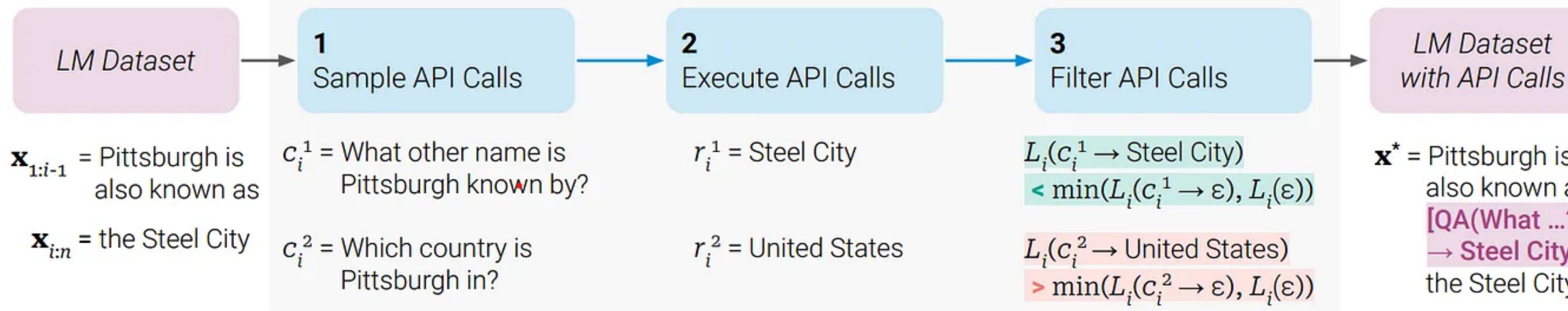


MCP

Universal translator between
Host and tools
知乎 @北方的郎

Toolformer

Toolformer: Language Models Can Teach Themselves to Use Tools



The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

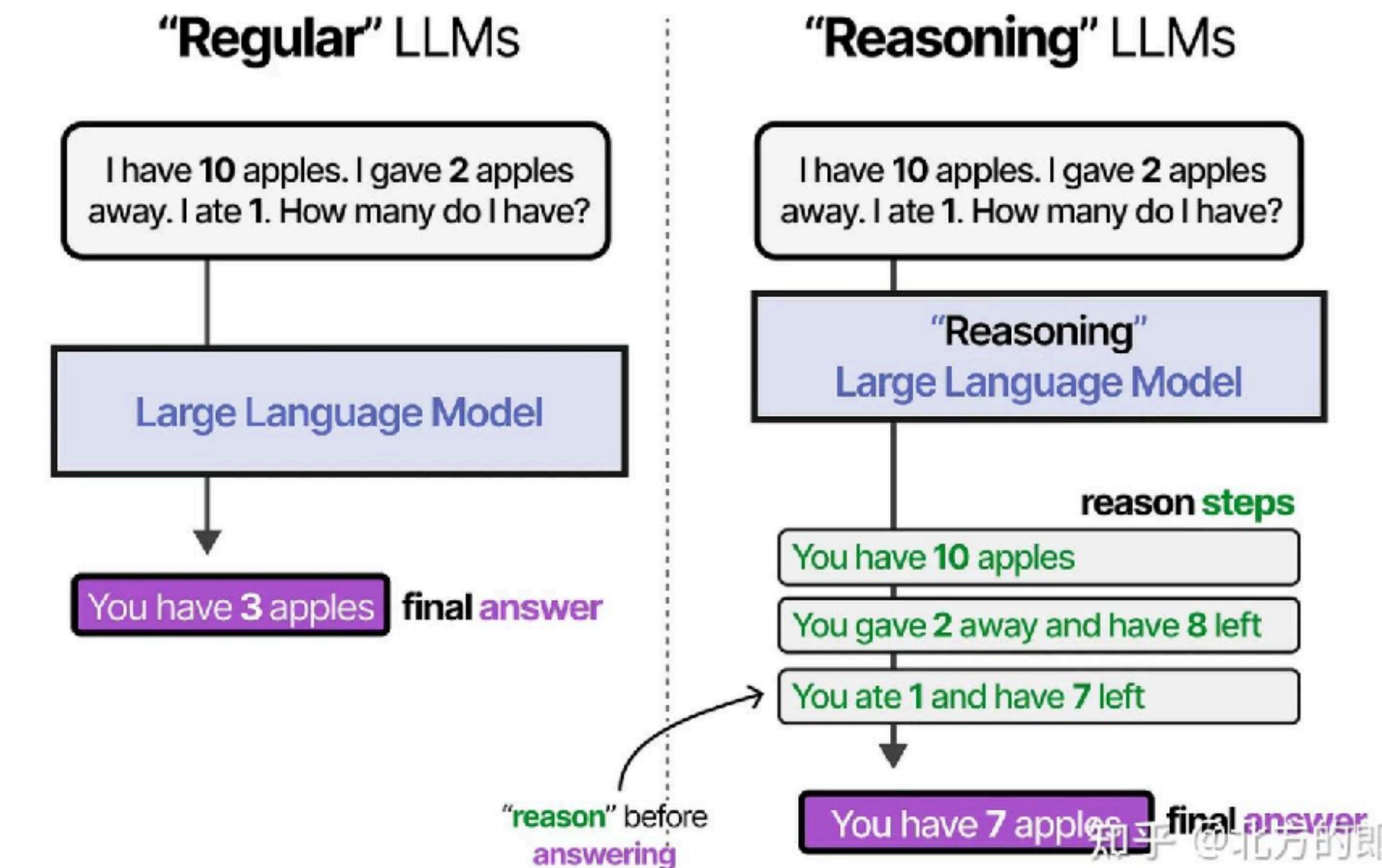
Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

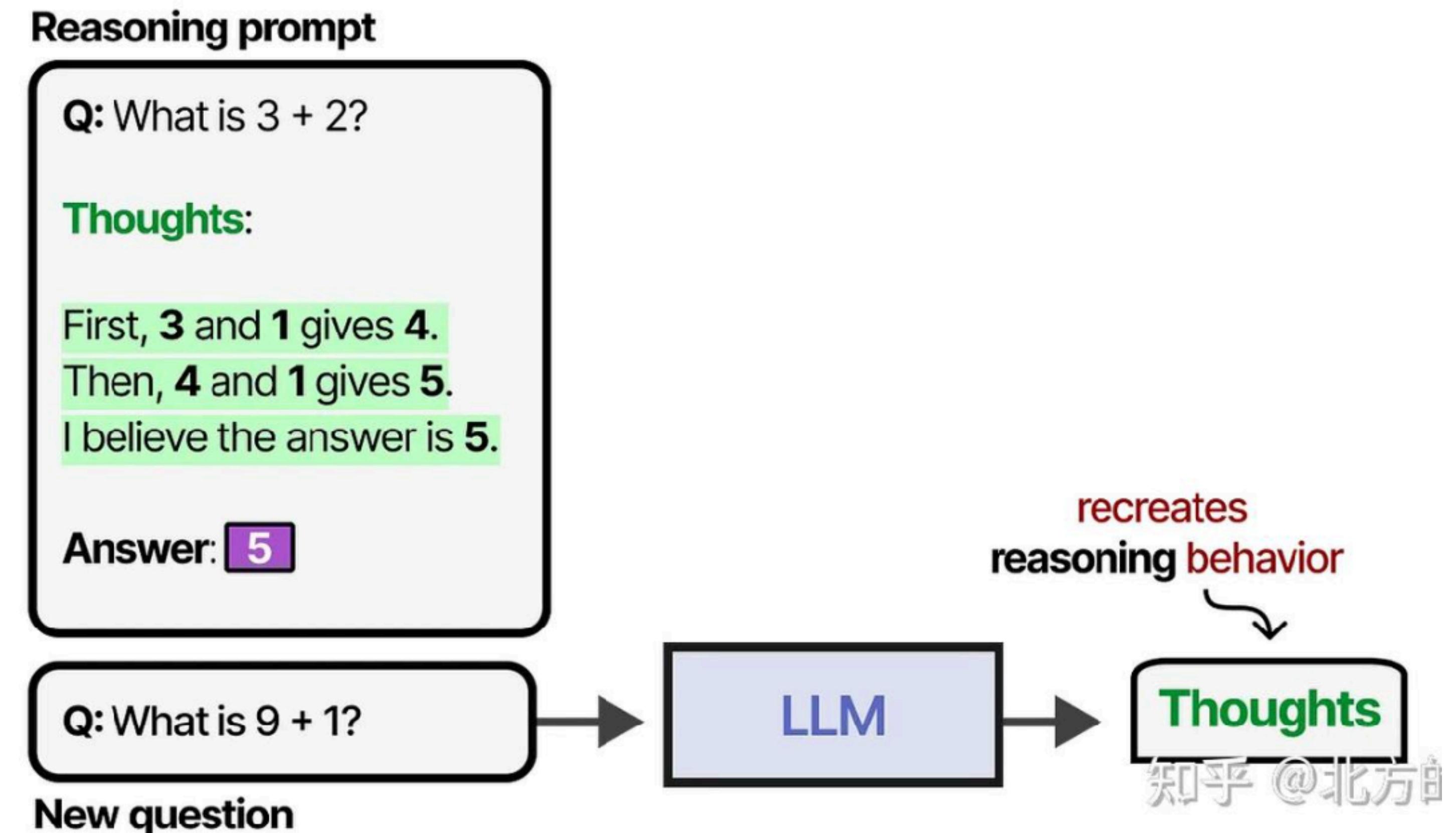
Planning

- Chain-of-Thoughts
- ReAct
- Reflection
- Self-Refine
- Subgoal decomposition



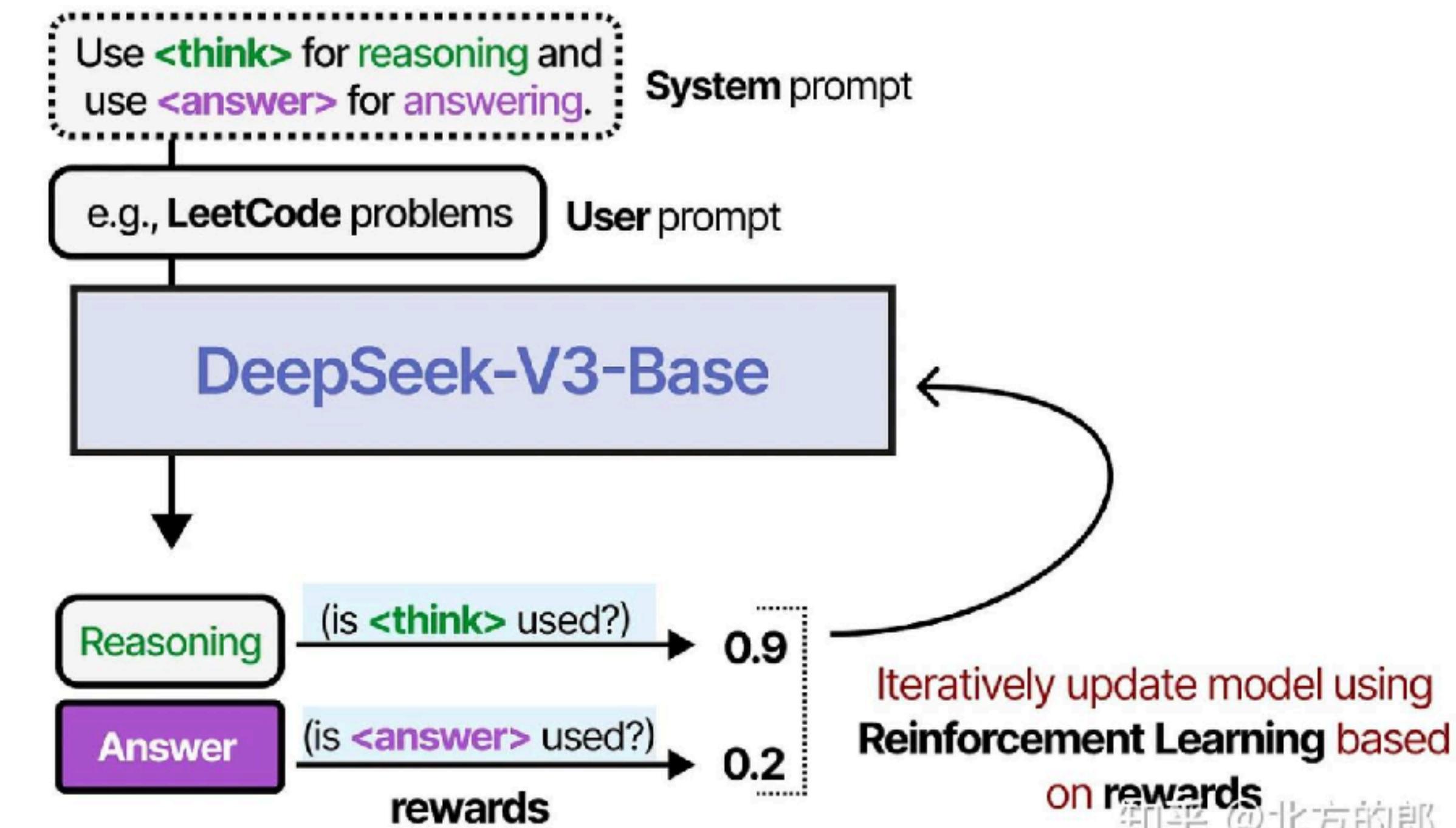
Planning

- Chain-of-Thoughts
- ReAct
- Reflection
- Self-Refine
- Subgoal decomposition



Planning

- Chain-of-Thoughts
- ReAct
- Reflection
- Self-Refine
- Subgoal decomposition



知乎 @北方的郎

Profile

Role - Identity of the Agent. Also stores information about available **memory** and **actions**.

coder, teacher, etc.

Memory

Store information - Information database related to Perception, Planning and Actions

short-term memory, long-term memory, RAG, etc.

Action

Tools - Methods for interacting with the environment

tools, function calling, etc. [google](#) [ArXiv](#)

Planning

Reasoning - Methods for reasoning, planning, and Reflection

ReAct, Reflexion, etc.

Percept

Sensory Information - Inputs to the Agent

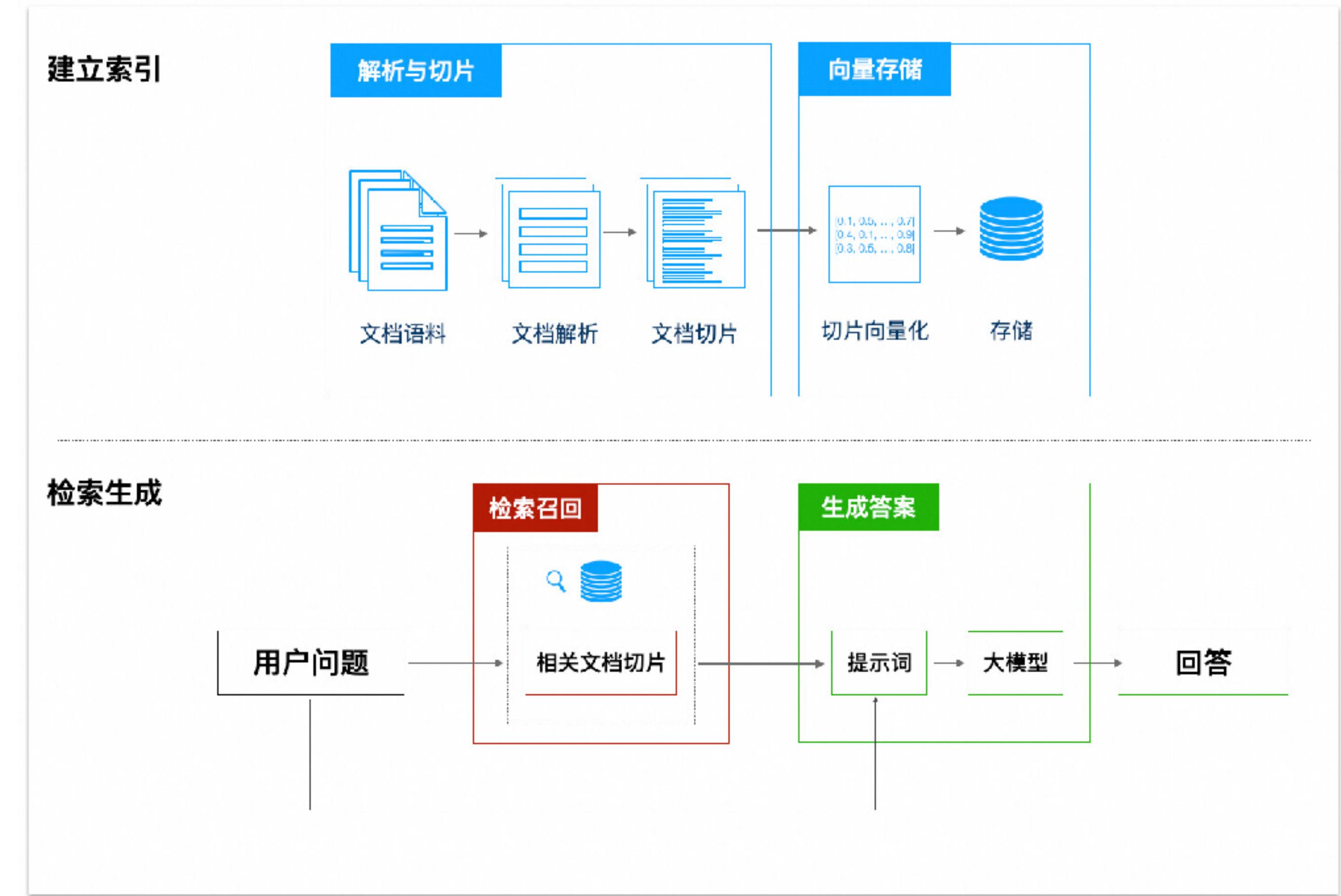
text, images, sound, etc.



模块化框架

第一层次：智能体应用（Assistant）— 阿里百炼

- 大语言模型（Large Language Model, LLM）无法直接访问专有知识库或获取实时动态信息。针对这一瓶颈，智能体支持以零代码方式，将大模型与外部工具进行集成，从而扩展模型的能力边界。
- 基本原理
 - 智能体（Agent）由提示词（Prompt）驱动，通过协同多种外部能力来完成复杂任务。在接收请求后，大模型进行意图理解和任务规划，自主决策并调用一个或多个外部能力来执行任务，最终整合信息生成响应。
- 智能体支持核心能力包括：
 - 检索增强生成（RAG）：通过连接外部知识库，使应用能基于私有数据回答问题，解决大模型无法访问特定信息的问题。
 - 插件（Plug-in）：能调用平台预置的效率工具（如代码执行、图像生成、天气查询等）。
 - 模型上下文协议（MCP）：允许将第三方服务封装并接入智能体，智能体可调用这些外部服务来完成特定工作。
 - 组件（Component）：允许将其他智能体或工作流应用封装为模块化组件，实现复杂功能的复用。
 - 记忆（Memory）：使应用能够跨会话存储和回忆关键信息，实现个性化、连贯的对话体验。



RAG

插件概述

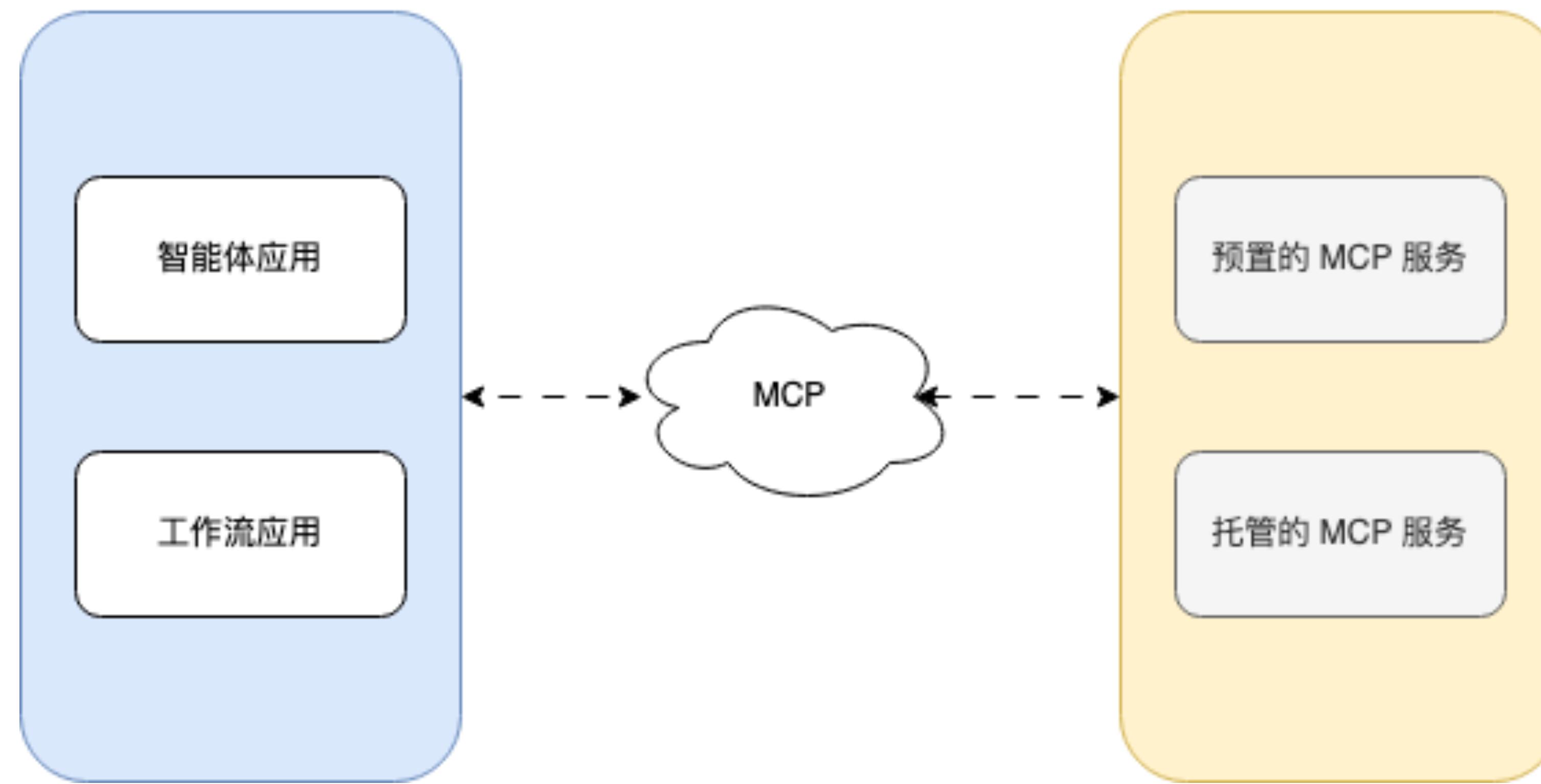
- 尽管大模型在处理复杂任务时展现出了卓越的性能，但它仍然存在一些局限性，例如无法获取最新信息、容易出现幻觉、难以精确计算等。为了解决这些问题，您可以将插件集成到大模型应用中进一步拓展大模型能力，执行更复杂的任务。
- 插件介绍
 - 插件是一个工具的集合。一个插件下可以包含多个工具（API），每个工具实现特定的功能。百炼支持官方插件、三方插件及自定义插件。

官方插件名称	工具 ID	说明	计费方案
Python 代码解释器	code_interpreter	使大模型可以执行 Python 代码片段，例如数学计算、数据分析与可视化、数据处理。	免费
计算器	calculator	使大模型可以进行复杂的数学计算，例如计算“12313x13232”。	免费
图片生成	text_to_image	使大模型可以基于文本生成图片，例如“请画一只在笑的小狗”。	限时免费，需申请开通
夸克搜索	quark_search	使大模型可以搜索实时信息，查找公开的网络知识和信息，例如“杭州今天天气如何”。 夸克搜索插件目前支持检索出网页标题、关键词和摘要，但不支持直接访问网页详情。	限时免费，需申请开通
生成二维码	generate_qrcode	使大模型可以根据网站链接地址生成二维码，例如“请给百炼简介文档生成二维码： https://help.aliyun.com/zh/model-studio/getting-started/what-is-model-studio ”。	免费
GitHub 搜索	github_search	使大模型可以在 GitHub 中搜索相关项目列表，例如“GitHub 搜索：通义千问”。	免费

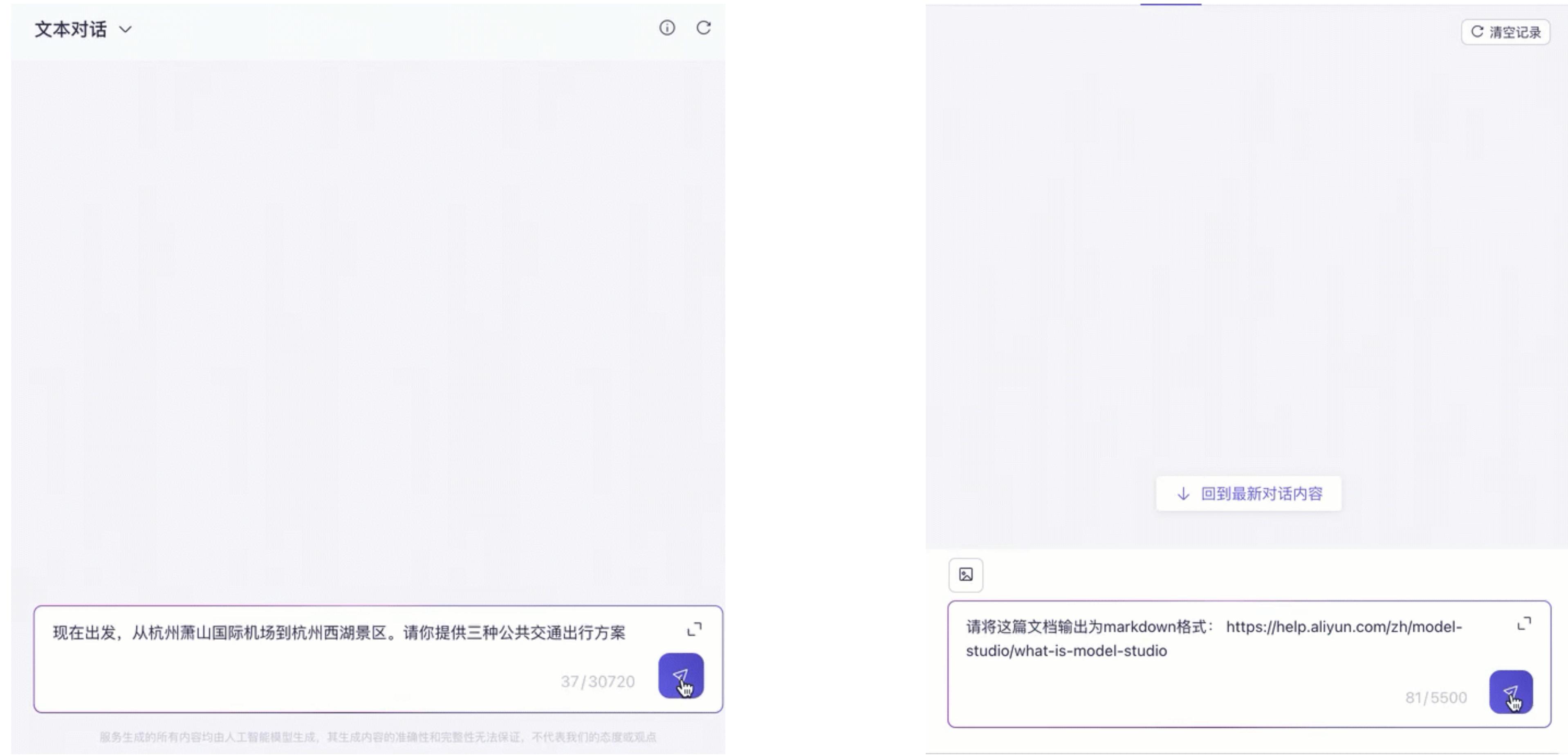
百炼官方插件

MCP 客户端

MCP 服务



MCP



MCP案例

长期记忆

功能介绍

在智能体应用中启用长期记忆功能后，将自动创建长期记忆体（Memory）。长期记忆体通过记忆片段和记忆变量实现记忆管理。在调用智能体应用时，系统会根据传入的长期记忆体ID（Memory ID），自动召回记忆体内容（记忆片段和记忆变量），并将其与当前用户提问一起传递给模型生成答案。

- **记忆片段（即MemoryNode）：**

阿里云百炼会自动提取对话中的用户个性化信息存储到记忆片段中（如用户说“我喜欢吃辣的”会存储为“用户喜辣”），用于后续对话推荐。但其易受临时性信息干扰（如用户临时提及“尝试素食”），需通过记忆变量校准。

- **记忆变量：**

记忆变量是以键值对形式存在的，需手动定义字段，值可通过模型推理（基于用户提问生成）或人工输入（如录入外部用户行为数据）获取。其权重高于记忆片段，可弥补记忆片段的不完整性和即时偏差，确保推荐准确性。

场景示例

- 智能导购助手：收集并记住顾客的预算范围、品牌偏好、关注的具体功能等信息，提供个性化的购买建议。
- 健身教练助手：收集并记住用户的身体状况、健身目标和可用设备等信息，提供个性化的锻炼计划和饮食建议。
- 旅游规划助手：收集并记住用户的预算、兴趣爱好和旅行时间等信息，推荐合适的目的地、景点和行程安排。
- 学习辅导助手：收集并记住学生的年级、知识水平和薄弱科目等信息，提供有针对性的学习资料和练习题。

您也可以手动将特定信息添加到长期记忆中，例如添加“语言：中文”的键值对，这样智能体应用在后续对话中就会持续用中文回复。

第二层次：工作流应用（Workflow）

- 在您上手智能体应用后，如果觉得不能满足您的需求，还希望设计一个固定的 AI 工作流程，建议使用工作流应用。工作流应用是一种将复杂任务拆解为若干子任务，以提高工作流程可控性的流程式 AI 应用。用户可以通过拖拽节点来创建自定义的任务流程。详情参见工作流应用。

对比项	智能体应用	工作流应用
创建方法	0 代码	0 代码
应用形式	对话式	流程式
对提示词的依赖	较高	较低
知识库	支持	支持
插件	支持 (官方插件、自定义插件)	支持 (插件节点、API 节点)
长期记忆	支持	不支持
文本转换	部分支持 (依赖提示词)	支持 (变量处理节点、脚本节点)
接入外部服务	支持 (自定义插件、MCP 服务)	支持 (函数计算节点 、 AppFlow 节点、MCP 节点)
分支	部分支持 (依赖提示词)	支持 (条件判断节点、意图分类节点)
自动规划	部分支持 (依赖提示词)	支持 (智能体群组节点)

区别

第三个层次：Agentic RAG

- while (Lv2-智能体无法根据其记忆回答问题)
 - {
 - Lv2-智能体提出一个新的子问题待解答。
 - Lv2-智能体向Lv1-RAG提问这个子问题。
 - 将Lv1-RAG的回应添加到Lv2-智能体的记忆中。
 - }
 - Lv2-智能体提供原始问题的最终答案。

终极形态

- 智能体模式具备自主决策、环境感知、工具使用等能力，并能够自主调用 MCP 工具，可以让你通过自然语言描述，完成复杂的编码任务。

三、智能体的构建

LLM技术路线

- 大模型的参数调整
- 提示词工程
- 检索增强RAG
- 知识库挂载
- 指令微调
- 强化学习
- 流程编排

0. 调参

大模型参数

- **Temperature:** 简单来说，temperature 的参数值越小，模型就会返回越确定的一个结果。如果调高该参数值，大语言模型可能会返回更随机的结果，也就是说这可能会带来更多多样化或更具创造性的产出。（调小temperature）实质上，你是在增加其他可能的 token 的权重。在实际应用方面，对于质量保障（QA）等任务，我们可以设置更低的 temperature 值，以促使模型基于事实返回更真实和简洁的结果。对于诗歌生成或其他创造性任务，适度地调高 temperature 参数值可能会更好。
- **Top_p:** 同样，使用 top_p（与 temperature 一起称为核采样（nucleus sampling）的技术），可以用来控制模型返回结果的确定性。如果你需要准确和事实的答案，就把参数值调低。如果你在寻找更多样化的响应，可以将其值调高点。
- 使用Top P意味着只有词元集合（tokens）中包含top_p概率质量的才会被考虑用于响应，因此较低的top_p值会选择最有信心的响应。这意味着较高的top_p值将使模型考虑更多可能的词语，包括不太可能的词语，从而导致更多样化的输出。
- 一般建议是改变 Temperature 和 Top P 其中一个参数就行，不用两个都调整。
- **Max Length:** 您可以通过调整 max length 来控制大模型生成的 token 数。指定 Max Length 有助于防止大模型生成冗长或不相关的响应并控制成本。
- **Stop Sequences:** stop sequence 是一个字符串，可以阻止模型生成 token，指定 stop sequences 是控制大模型响应长度和结构的另一种方法。例如，您可以通过添加“11”作为 stop sequence 来告诉模型生成不超过 10 个项的列表。
- **Frequency Penalty:** frequency penalty 是对下一个生成的 token 进行惩罚，这个惩罚和 token 在响应和提示中已出现的次数成比例，frequency penalty 越高，某个词再次出现的可能性就越小，这个设置通过给重复数量多的 Token 设置更高的惩罚来减少响应中单词的重复。
- **Presence Penalty:** presence penalty 也是对重复的 token 施加惩罚，但与 frequency penalty 不同的是，惩罚对于所有重复 token 都是相同的。出现两次的 token 和出现 10 次的 token 会受到相同的惩罚。此设置可防止模型在响应中过于频繁地生成重复的词。如果您希望模型生成多样化或创造性的文本，您可以设置更高的 presence penalty，如果您希望模型生成更专注的内容，您可以设置更低的 presence penalty。
- 与 temperature 和 top_p 一样，一般建议是改变 frequency penalty 和 presence penalty 其中一个参数就行，不要同时调整两个。

1. Prompt

预置场景	介绍	推荐场景	预置 Prompt	生成效果文案
营销标题生成	快速生成各流量渠道的吸引眼球的标题文案	营销推广	<p>你是一名专业的【小红书】爆款标题专家。请你以【杭州亚运会】为主题，以【YYDS、建议收藏、揭秘】为关键词，提供【10】个吸引人眼球的标题。每个标题字数在【20】个字以内。</p>	<ol style="list-style-type: none">1. 【YYDS】揭秘杭州亚运会幕后故事！2. 【建议收藏】杭州亚运会精彩瞬间不容错过！3. 杭州亚运会【YYDS】金牌选手独家采访！4. 【揭秘】杭州亚运会比赛场馆全揭秘！5. 杭州亚运会【YYDS】金牌赛事精彩回顾！6. 【建议收藏】杭州亚运会志愿者感人故事！7. 杭州亚运会【YYDS】金牌教练独家分享！8. 【揭秘】杭州亚运会美食探秘之旅！9. 杭州亚运会【YYDS】金牌赛事精彩瞬间！

Prompt模版

Prompt样例库

- 大量实践表明，FewShot方法在提升大模型的推理效果和性能方面具有显著的助益。Prompt样例库功能作为百炼的FewShot能力，通过录入用户输入（Query）和期望的模型回复（Answer）作为样例信息源，在模型调用时根据用户输入，检索召回相关样例信息，以此作为输出参考，从而提高大模型的回答准确性，适用于客服及问答等场景。

用户输入	模型输出	操作
API-KEY的上限个数是多少?	当前每个主账号可以同时有3个生效的API-KEY	删除
什么是API-KEY?	DashScope通过API-KEY进行调用鉴权和计量计费，目前支持主账号进行API-KEY的管理。	删除
API-KEY被删除会有什么影响?	目前支持主账号对API-KEY的删除，子账号无法进行操作。删除的API-KEY无法找回也无法使用，但您在调用统计页面仍能查询被删除的API-KEY以往的调用统计结果。	删除

下面汽车相关的文章属于下面哪一种类别：“产品解析”，“车商卖车”，“经典怀旧”，“质量投诉”，“销量表现”，“其他”。最终的结果请按照(“type”:“<类别结果>”的json格式进行输出

原始Prompt

以下是一些参考示例：

输入：

文章标题：ID3不墨迹 直接点下面有惊喜！！！\n文章正文:\n我承认，我一直都放不下你，就算我打电话你不接、发信息你不回，约你见面你也不来我依然放不下你只因为之前你说过 等你有需要买车，会主动联系我，而且还会让你的同学、同事和七大姑八大姨找我买车 ID3月底了一台没卖，经理又过来找我 10.xci这个价只卖一台，不开玩笑，这关乎我的职业生涯来扣一，直接报高价 #ID3掀屋顶了#适合女生开的车#上汽大众# ID #上汽大众ID3 #ID3 #上汽大众ID4 #上汽大众id4x #ID4 #id4x

输出：

{"type": "车商卖车"}

输入：

文章标题：大众朗逸，中保研安全指数全优成绩，让你的安全无忧。大众朗逸在中保研安全指数评测中取得全优成绩，证明了其卓越的安全性能。大众朗逸更是凭借其出色的安全配置和稳定性能，成为驾驶者的放心之选。选择大众朗逸，让安全成为你的座驾标配！ #朗逸 #大众朗逸中保研安全指数全优成绩 #大众朗逸\n这是一段视频，视频摘要是：【舆情分析摘要】近期，大众朗逸在中保研的严格安全测评中荣获了全面优秀的评价。这一成绩充分体现了朗逸车型在安全性能方面的领先地位，其丰富的安全配置和稳定的机械性能赢得了消费者的高度认可。作为一款可靠的车型，大众朗逸为驾驶者提供了全方位的安全保障，使其在市场上树立了良好的安全口碑。此次评测结果进一步增强了消费者对朗逸的信任，提升了品牌的市场竞争力。

输出：

{"type": "产品解析"}

添加样例

输入：

文章标题：大众帕萨特底盘异响车主竟说车上有鸭子叫\n文章正文: 大众帕萨特底盘异响车主竟说车上有鸭子叫\n这是一段视频，视频摘要是：\n一位大众车主反映其车辆在行驶中出现疑似鸭子胶的异响。经过检查，问题源于下巴臂胶套移位。车主之前在其修理厂更换了非原厂的总成，但因孔径不匹配导致问题持续存在。鉴于此，维修团队建议车主为确保修复效果和避免反复维修成本，应考虑更换原厂总成。他们强调，修车时不应盲目节约，以免造成时间和金钱的浪费，提倡杜绝不必要的反复修车行为。

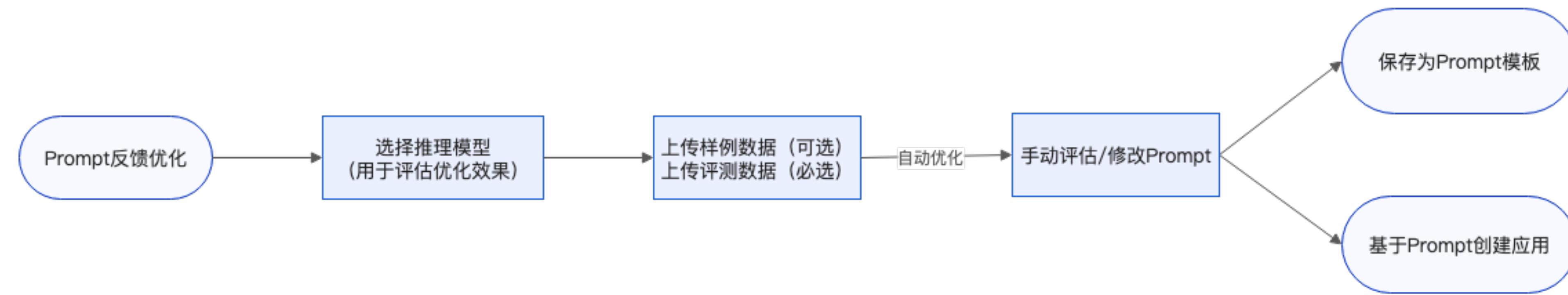
输出：

{"type": "质量投诉"}

- 当文章内容主要描述了车主遇到的具体车辆问题（如异响）及其解决过程时，建议归类为“质量投诉”。
- 当文章内容主要介绍车辆的具体性能、配置或评测结果时，建议归类为“产品解析”。
- 当文章提供的信息非常有限，仅包含车型名称和链接而没有其他具体内容或描述时，建议归类为“其他”。
- 当文章内容主要是个人对某款车型的看法或感受，且不侧重于产品具体解析、销售推广、历史回顾、质量投诉或销量数据时，建议归类为“其他”。
- 当文章内容主要涉及销售促销信息、价格优惠等直接促进购买行为的信息时，建议归类为“车商卖车”。
- 当文章内容主要提出问题或讨论个人偏好，而不侧重于产品具体解析、销售推广、历史回顾、质量投诉或销量数据时，建议归类为“其他”。

根据输入输出样例总结提示词
- 当文章提供的信息非常有限，仅包含车型名称和标签而没有其他具体内容或描述时，建议归类为“其他”。
- 当文章内容主要回顾某一车型的历史地位及其对市场的影响时，建议归类为“经典怀旧”。即使提及车辆销售情况或促销活动，仍需重点考虑文章的核心主题。
- 当文章内容涉及品牌历史、车型发展背景等信息，并且这些信息占据了文章的主要部分时，建议归类为“经典怀旧”。
</tips>

Prompt优化



Prompt自动优化

少样本提示

- 根据 Touvron et al. 2023等人的在 2023 年的论文，当模型规模足够大时，少样本提示特性开始出现 (Kaplan et al., 2020) 让我们通过Brown等人2020年提出的一个例子来演示少样本提示。在这个例子中，任务是在句子中正确使用一个新词。
 - 提示：
 - “whatpu”是坦桑尼亚的一种小型毛茸茸的动物。一个使用whatpu这个词的句子的例子是：
 - 我们在非洲旅行时看到了这些非常可爱的whatpus。
 - “farduddle”是指快速跳上跳下。一个使用farduddle这个词的句子的例子是：
 - 输出：
 - 当我们赢得比赛时，我们都开始庆祝跳跃。
- 我们可以观察到，模型通过提供一个示例（即1-shot）已经学会了如何执行任务。对于更困难的任务，我们可以尝试增加演示（例如3-shot、5-shot、10-shot等）。
- 根据Min等人（2022）的研究结果，以下是在进行少样本学习时关于演示/范例的一些额外提示：
 - “标签空间和演示指定的输入文本的分布都很重要（无论标签是否对单个输入正确）”
 - 使用的格式也对性能起着关键作用，即使只是使用随机标签，这也比没有标签好得多。
 - 其他结果表明，从真实标签分布（而不是均匀分布）中选择随机标签也有帮助。

(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?

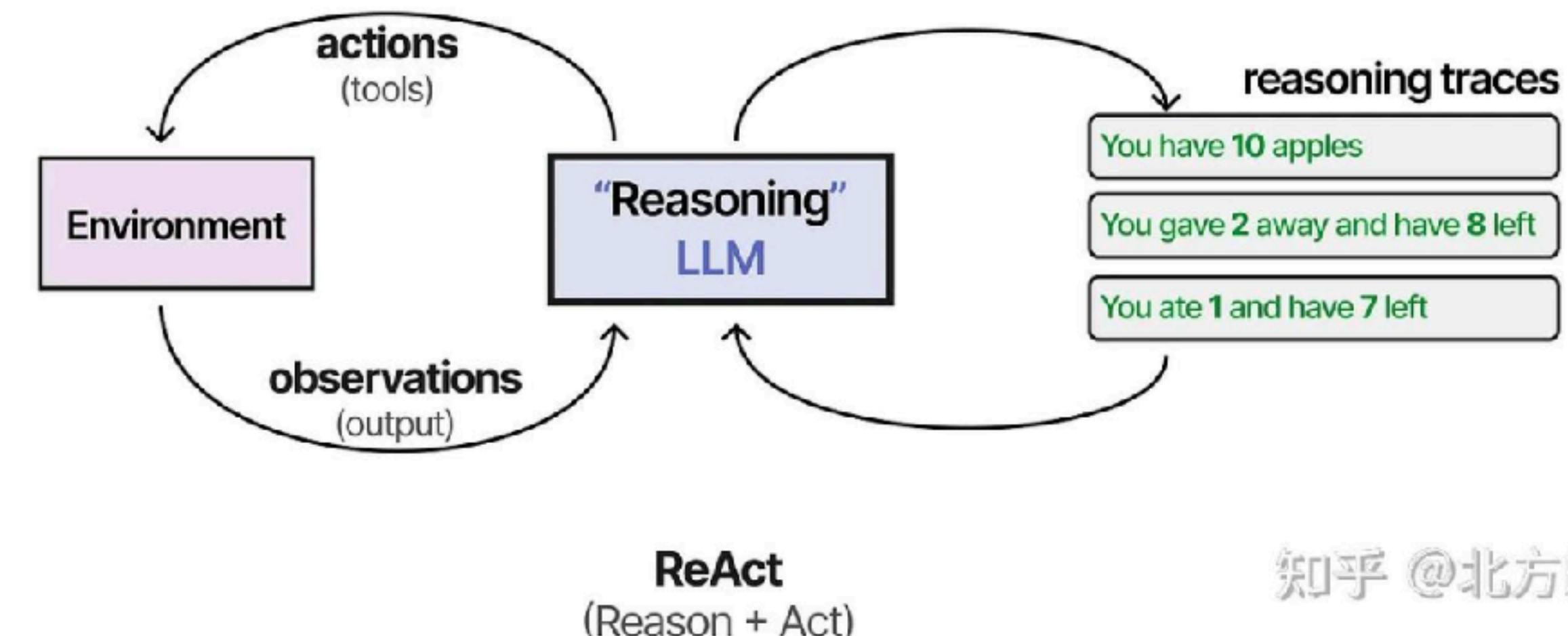
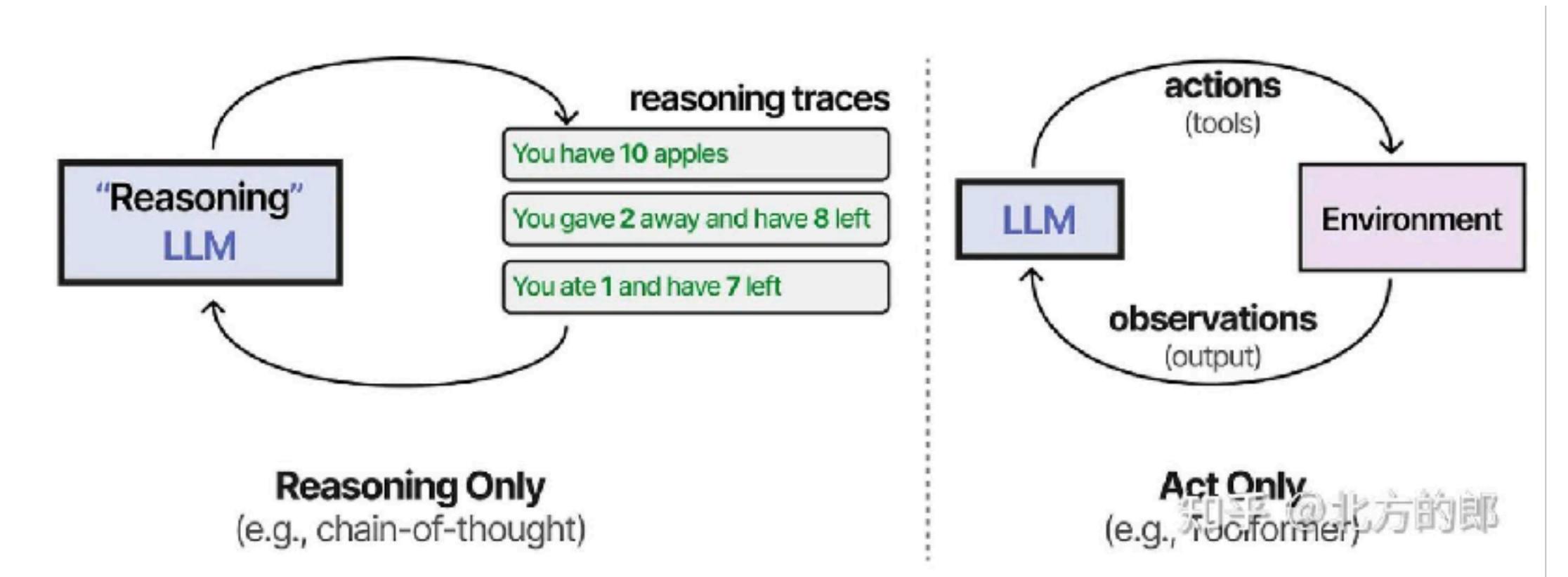
A: Let's think step by step.

(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. ✓

链式思考 (CoT) 提示

Planning

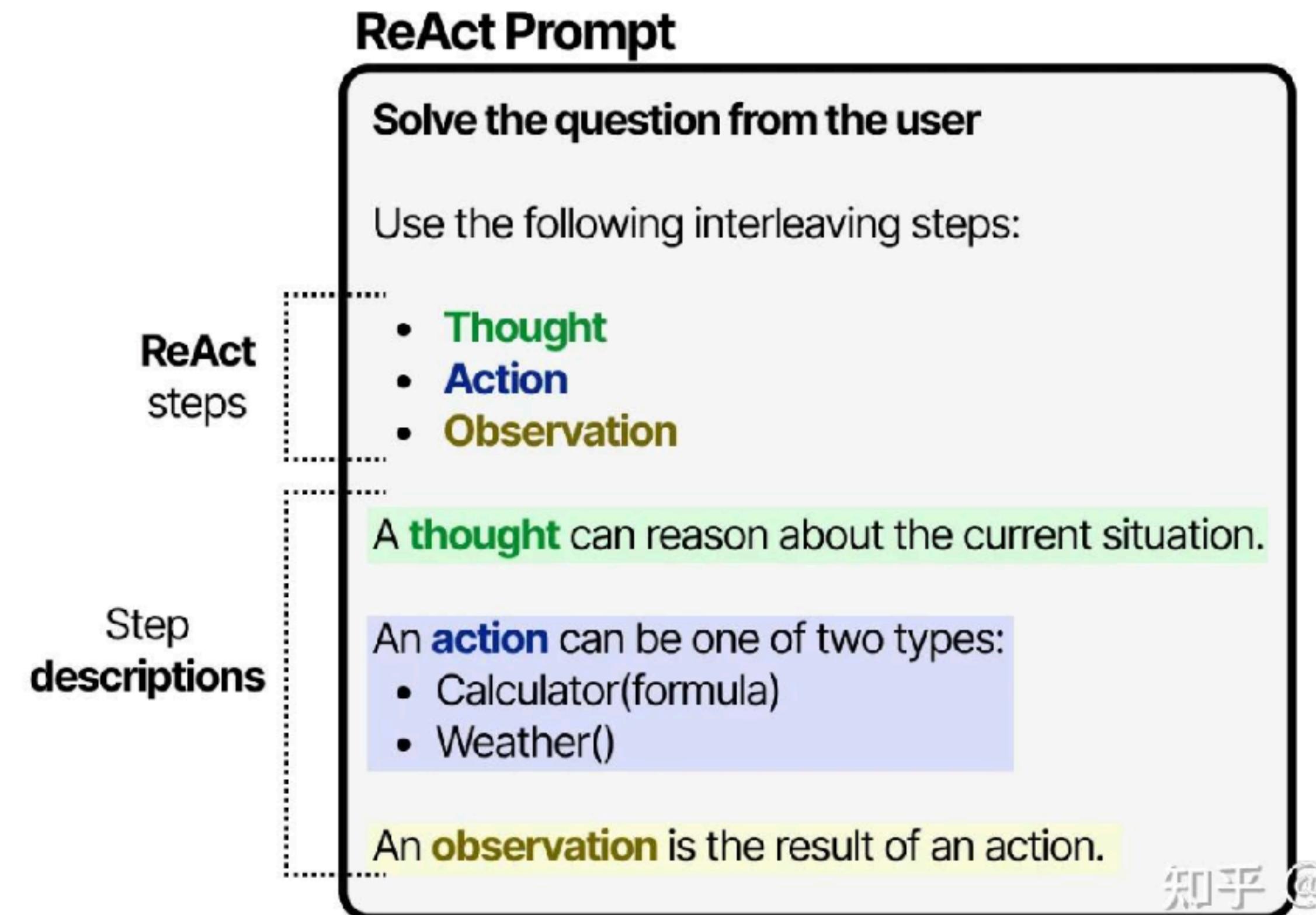
- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



知乎 @北方的郎

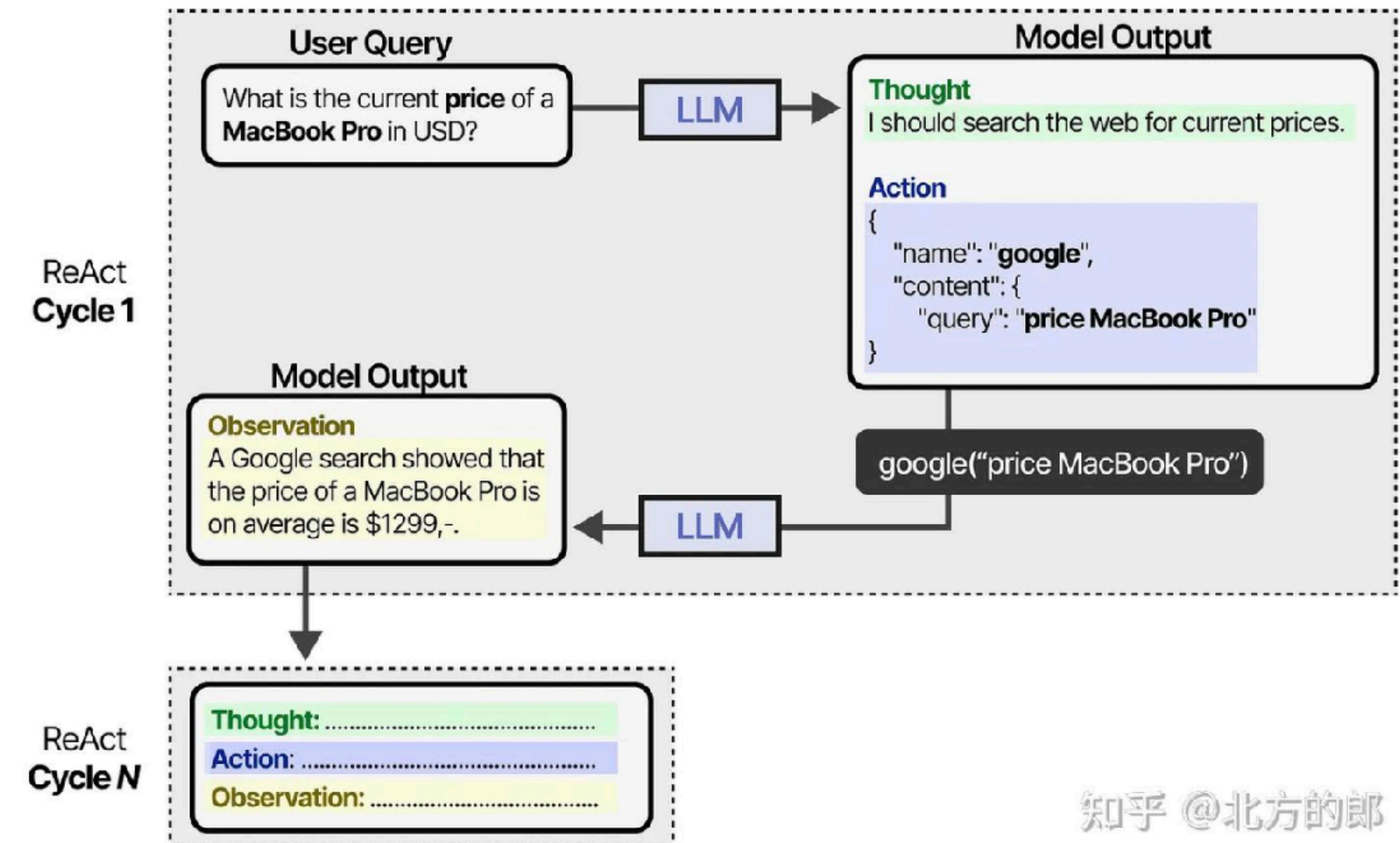
Planning

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



Planning

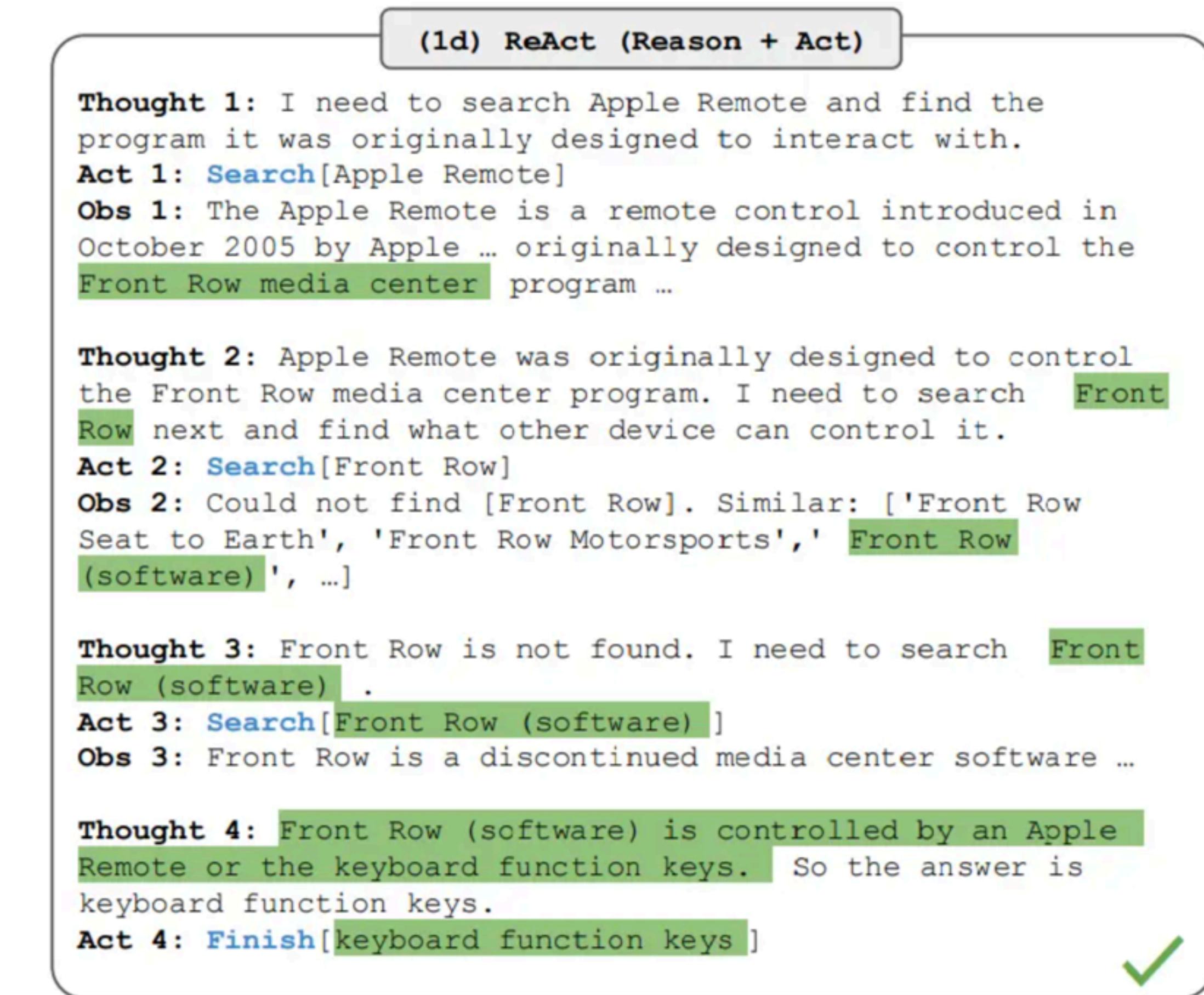
- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



知乎 @北方的郎

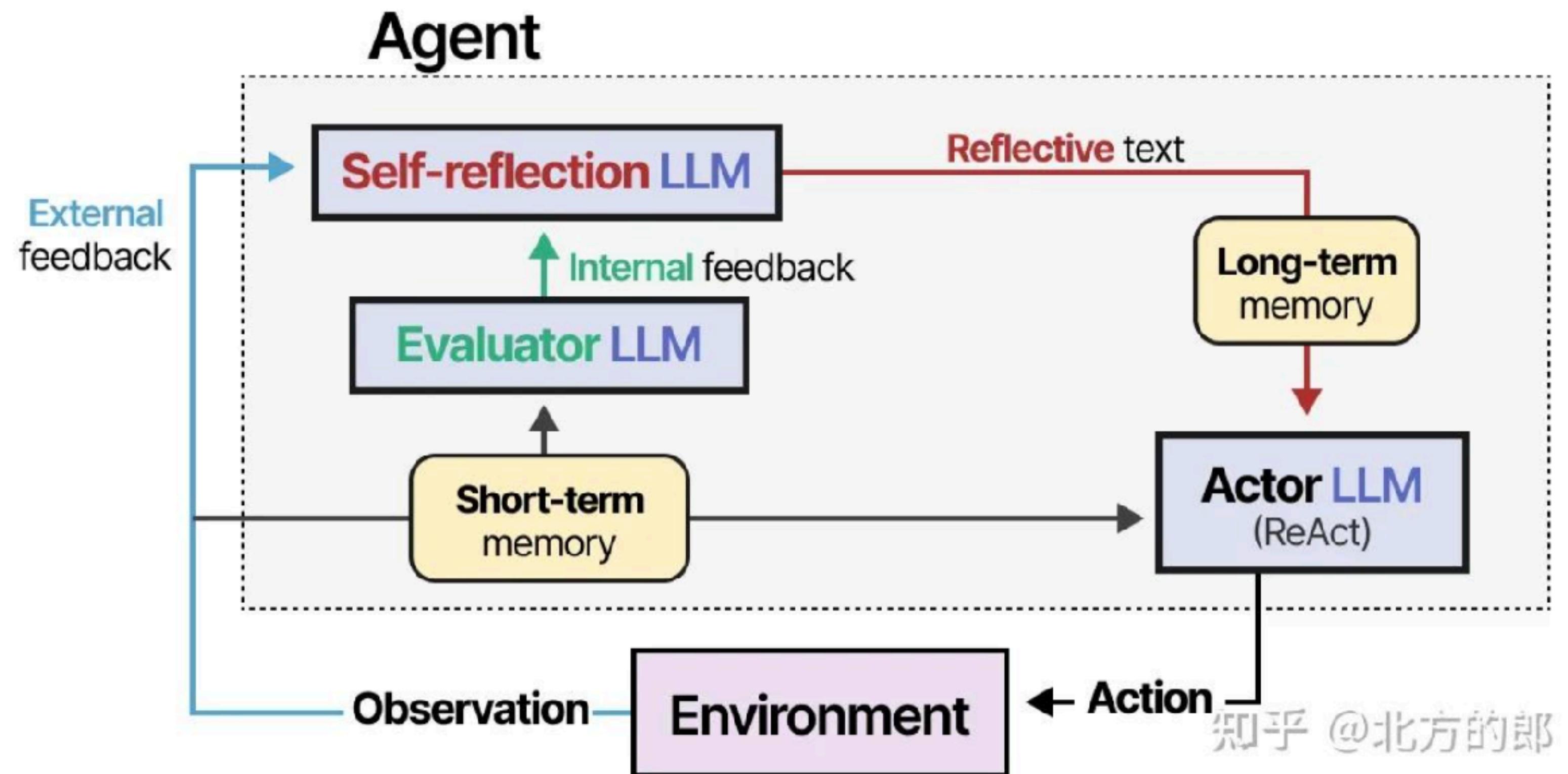
Plan

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



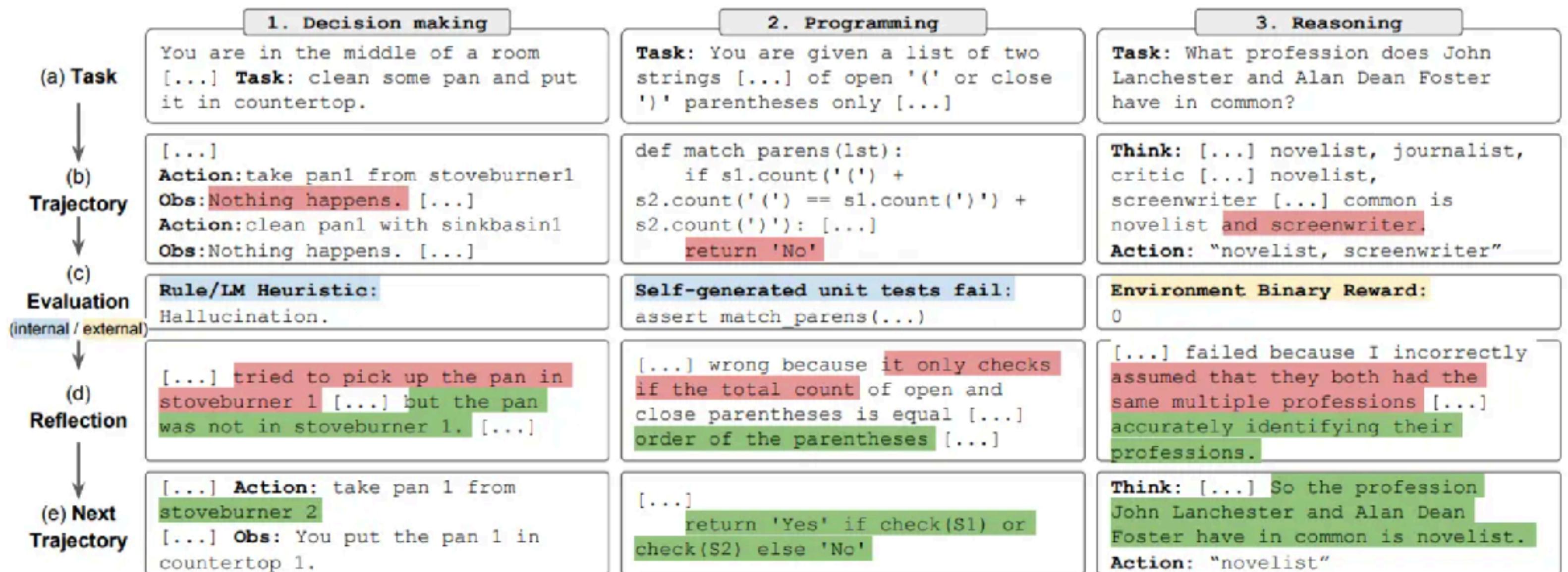
Planning

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposit



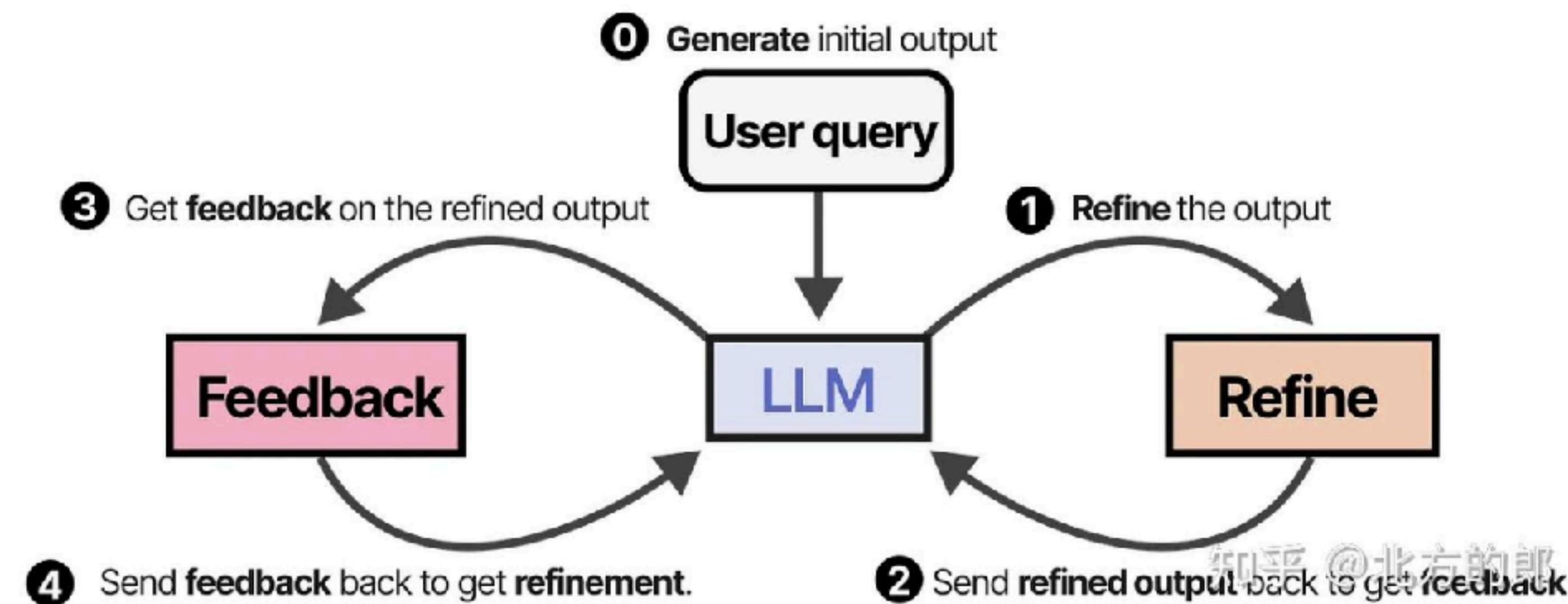
Planning

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



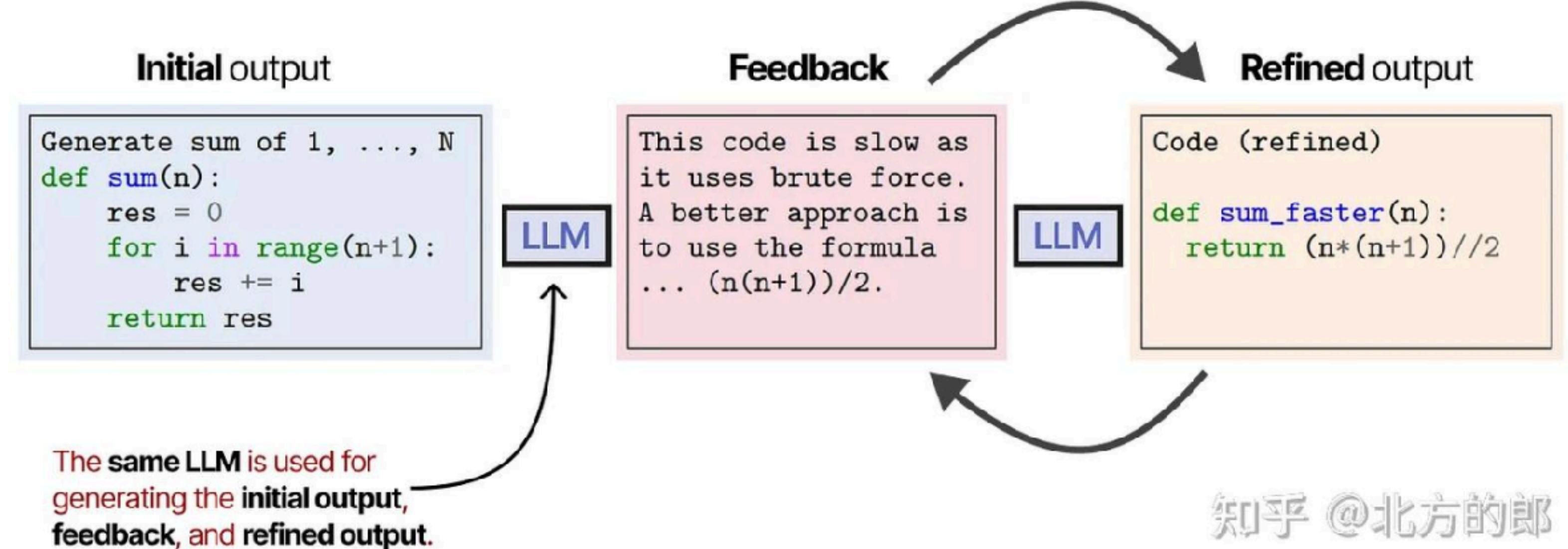
Planning

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



Planning

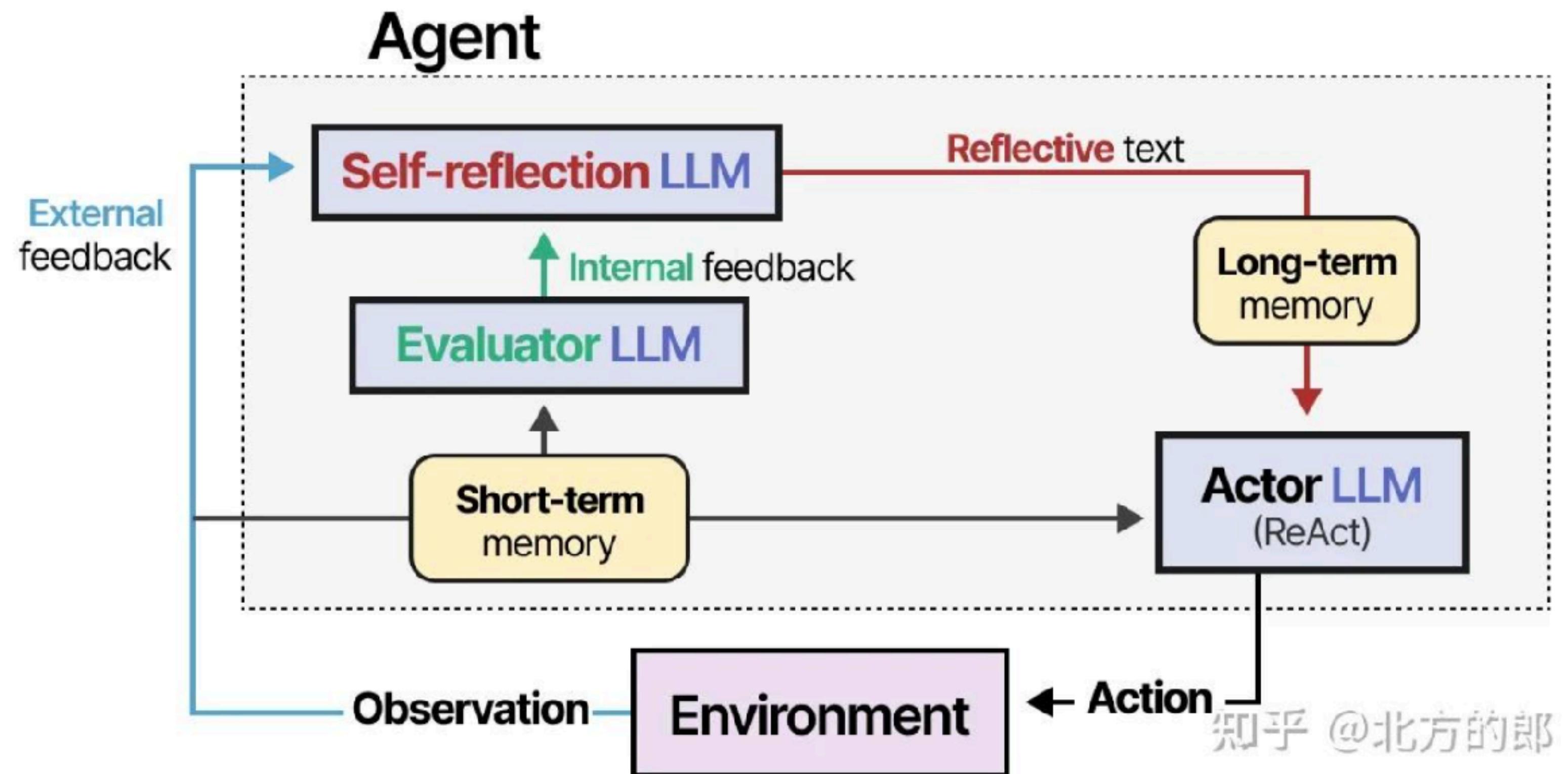
- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposition



知乎 @北方的郎

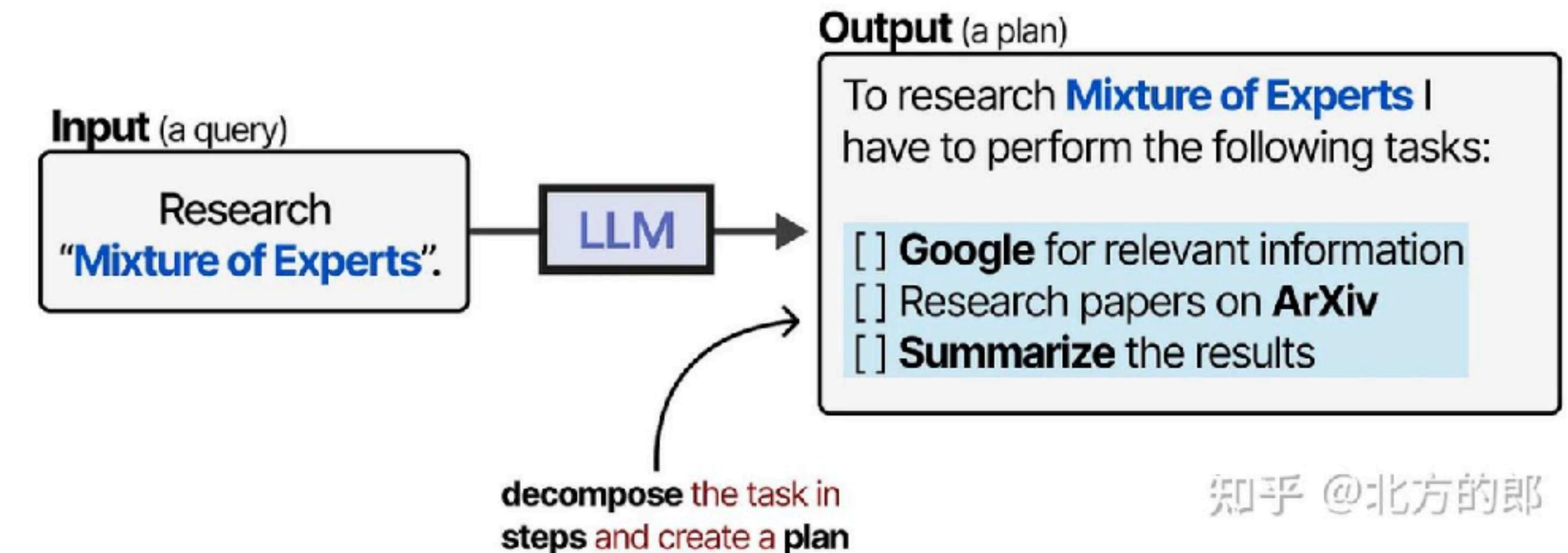
Planning

- Chain-of-Thoughts
- React
- Reflection
- Self-Refine
- Subgoal decomposit



Planning

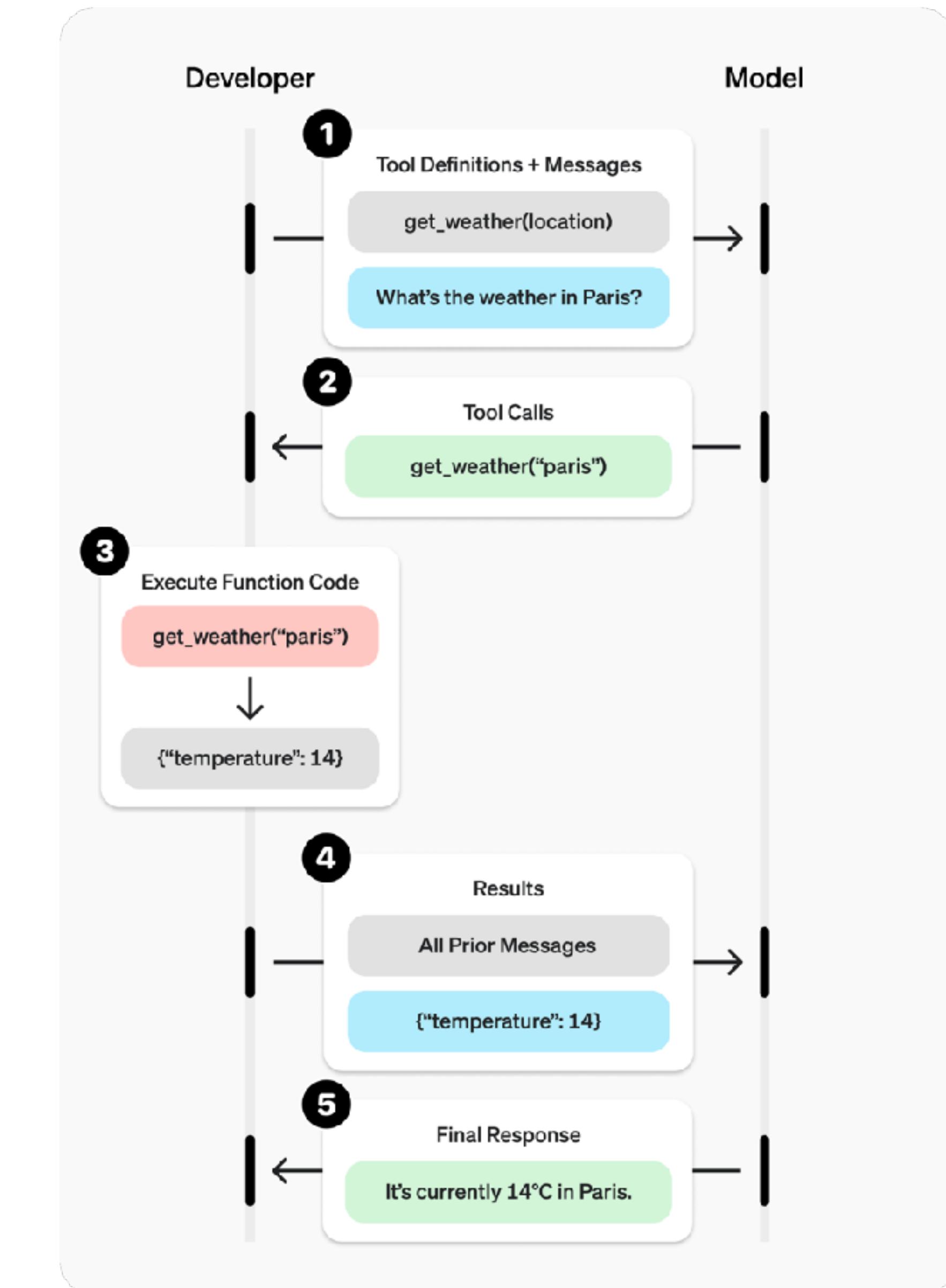
- Chain-of-Thoughts
- ReAct
- Reflection
- Self-Refine
- Subgoal decomposition



知乎 @北方的郎

2. 自定义插件

OpenAI Function Calling



Sample代码

```
from openai import OpenAI
import json

client = OpenAI()

# 1. Define a list of callable tools for the model
tools = [
    {
        "type": "function",
        "name": "get_horoscope",
        "description": "Get today's horoscope for an astrological sign.",
        "parameters": {
            "type": "object",
            "properties": {
                "sign": {
                    "type": "string",
                    "description": "An astrological sign like Taurus or Aquarius",
                },
            },
            "required": ["sign"],
        },
    },
]

def get_horoscope(sign):
    return f"{sign}: Next Tuesday you will befriend a baby otter."

# Create a running input list we will add to over time
input_list = [
    {"role": "user", "content": "What is my horoscope? I am an Aquarius."}
]

# 2. Prompt the model with tools defined
response = client.responses.create(
    model="gpt-5",
    tools=tools,
    input=input_list,
)

# 3. Save function call outputs for subsequent requests
input_list += response.output

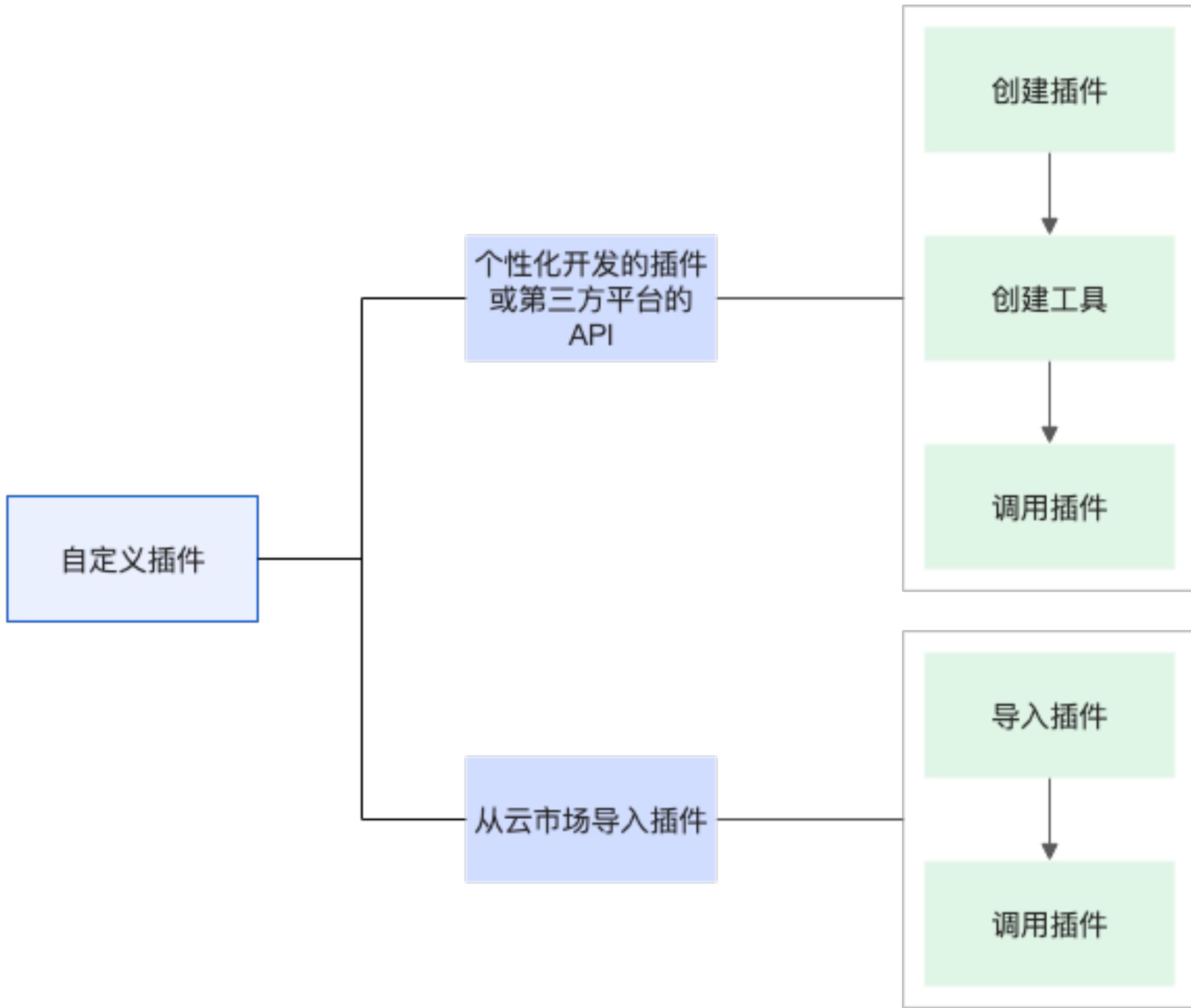
for item in response.output:
    if item.type == "function_call":
        if item.name == "get_horoscope":
            # 4. Execute the function logic for get_horoscope
            horoscope = get_horoscope(json.loads(item.arguments))

            # 5. Provide function call results to the model
            input_list.append({
                "type": "function_call_output",
                "call_id": item.call_id,
                "output": json.dumps({
                    "horoscope": horoscope
                })
            })

print("Final input:")
print(input_list)

response = client.responses.create(
    model="gpt-5",
    instructions="Respond only with a horoscope generated by a tool.",
    tools=tools,
    input=input_list,
)

# 5. The model should be able to give a response!
print("Final output:")
print(response.model_dump_json(indent=2))
print("\n" + response.output_text)
•
```



自定义插件工具 — 阿里百炼

3. 长期记忆

The screenshot shows two overlapping windows. The background window is the 'Smart Body Application - IFZa' configuration interface. In the foreground, a modal window titled '记忆变量' (Memory Variables) is displayed.

记忆变量

该功能用于配置变量字段，系统根据变量配置自动将变量信息。记忆变量抽取要在应用发布后生效。

操作

字段名称	字段描述	是否通过模型推理	操作
饮食偏好	用户偏好什么食物	<input checked="" type="checkbox"/>	删除
运动偏好	用户偏好什么运动	<input type="checkbox"/>	删除
年龄	用户的年龄	<input type="checkbox"/>	删除

步骤说明:

- 在左侧工具栏中点击 **+ 记忆变量** (1)
- 在弹出的对话框中点击 **添加字段 (V2)** (2)
- 选择要添加的变量 (3)
- 点击 **确定** (4) 完成添加

创建记忆槽

The screenshot shows a user interface for managing memory slots. A modal window titled '选择默认记忆体' (Select Default Memory) is open, listing five available memory slots:

ID	已绑定应用	描述
10c98 bb12e49e	智能体应用-IFZo	Automatically created when creating an agent
fe169 34f78a9	智能体应用-QVIR	Automatically created when creating an agent
e0e1 7713c60	文档智能百姓手机	Automatically created when creating an agent
4de0e2e53313-leaf-aec00001e 260095fa	智能体应用-语音视频	Automatically created when creating an agent
c7b1 39ubb35	智能体应用-WuCP	Automatically created when creating an agent

The first slot, '10c98 bb12e49e' (智能体应用-IFZo), is selected and highlighted with a blue background. The modal includes a text input field at the top right labeled '当前应用默认绑定的Memory ID: 10c98 bb12e49e' (Current application default bound Memory ID: 10c98 bb12e49e), a '创建 Memory ID' (Create Memory ID) button, and two buttons at the bottom right: '取消' (Cancel) and '确定' (Confirm). The background shows other tabs like '应用配置' (Application Configuration) and '发布渠道' (Release Channel), and a sidebar with sections like '指令', '知识', '动态文件解析', '联网搜索', '样例库', '技能', '插件', '智能体', '工作流', and '记忆'.

绑定记忆槽

请根据我的运动偏好，推荐周末的锻炼方式

长期记忆

“运动偏好”中不输入值，大模型回答时以记忆片段中的内容为准。

```
✓ 执行完成 ^  
⌚ 长期记忆检索 ^  
  
输入参数  
1 v {  
2   "memory_id": "10c98f140b7b42eab62bed8bbb12e49e",  
3   "memory_content": "当前时间是2025-06-20 11:03:48 周五。  
4   "[2025-6-20 10:58:35 周五] 用户习惯每周末都去打羽毛球。"  
  
输出参数 ^  
1 v [  
2   "当前时间是2025-06-20 11:03:48 周五。  
3   "[2025-6-20 10:58:35 周五] 用户习惯每周末都去打羽毛球。  
4 ]
```

根据您的运动偏好（每周末都去打羽毛球），我为您推荐以下周末锻炼方式：

1. 继续坚持打羽毛球

羽毛球是一项全身运动，能够有效提升心肺功能、增强肌肉力量和协调性。您可以继续安排时间与朋友或家人一起打球，享受运动的乐趣。

2. 结合有氧运动

如果想增加运动量，可以在打完羽毛球后进行一些轻松的有氧运动，例如慢跑或快走30分钟，进一步提高体能。

3. 尝试力量训练

为了提升羽毛球中的爆发力和稳定性，您可以在周末安排一次全身力量训练，重点锻炼腿部、核心肌群和上肢力量。

4. 瑜伽或拉伸放松

在运动后加入瑜伽或静态拉伸，有助于缓解肌肉紧张，预防运动损伤，同时帮助身体更快恢复。

5. 户外活动

如果天气允许，可以尝试爬山、骑行或徒步等户外活动，既能锻炼身体，又能亲近自然，放松心情。

请根据我的运动偏好，推荐周末的锻炼方式

长期记忆

“运动偏好”中手动输入“游泳”，在后续的对话中，将召回记忆变量，由模型判断并生成回答。

执行完成 ^

长期记忆检索 ^

输入参数

```
1 v {  
2   "memory_id": "10c98f140b7b42eab62bed8bbb12e49e",  
3   "memory_content": "平时喜欢游泳，周末打羽毛球。"  
4 }
```

输出参数

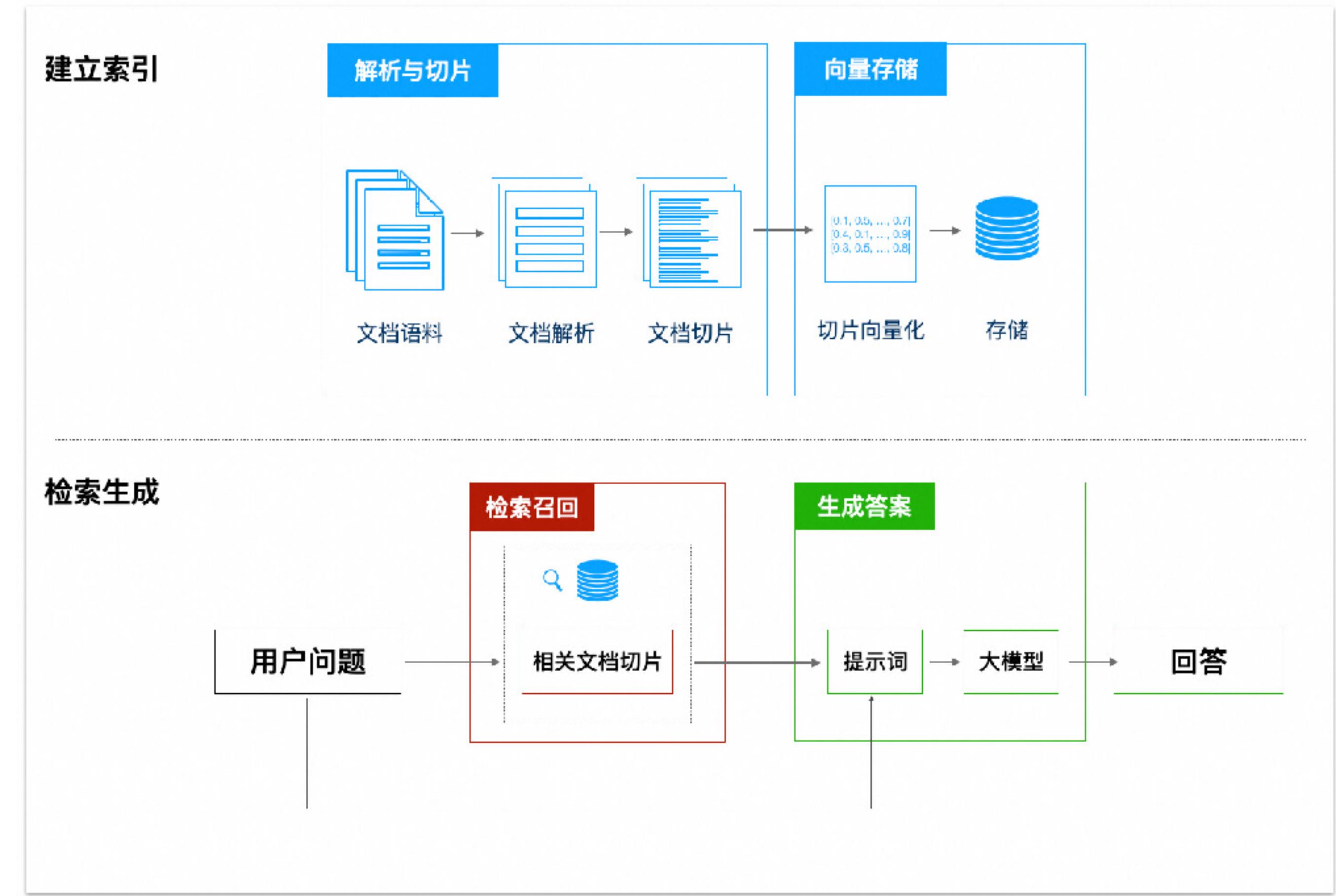
```
1 v [  
2   "当前时间是2025-06-20 11:05:00 周五。",  
3   "用户的运动偏好（用户偏好什么运动）：游泳",  
4   "[2025-6-20 10:58:35 周五] 用户习惯每周末都去打羽毛球。"  
5 ]
```

根据你的运动偏好和习惯，你平时喜欢游泳，并且每周末都有打羽毛球的习惯。针对这个周末（2025年6月24日是周二，离周末还有几天），我推荐以下锻炼方式：

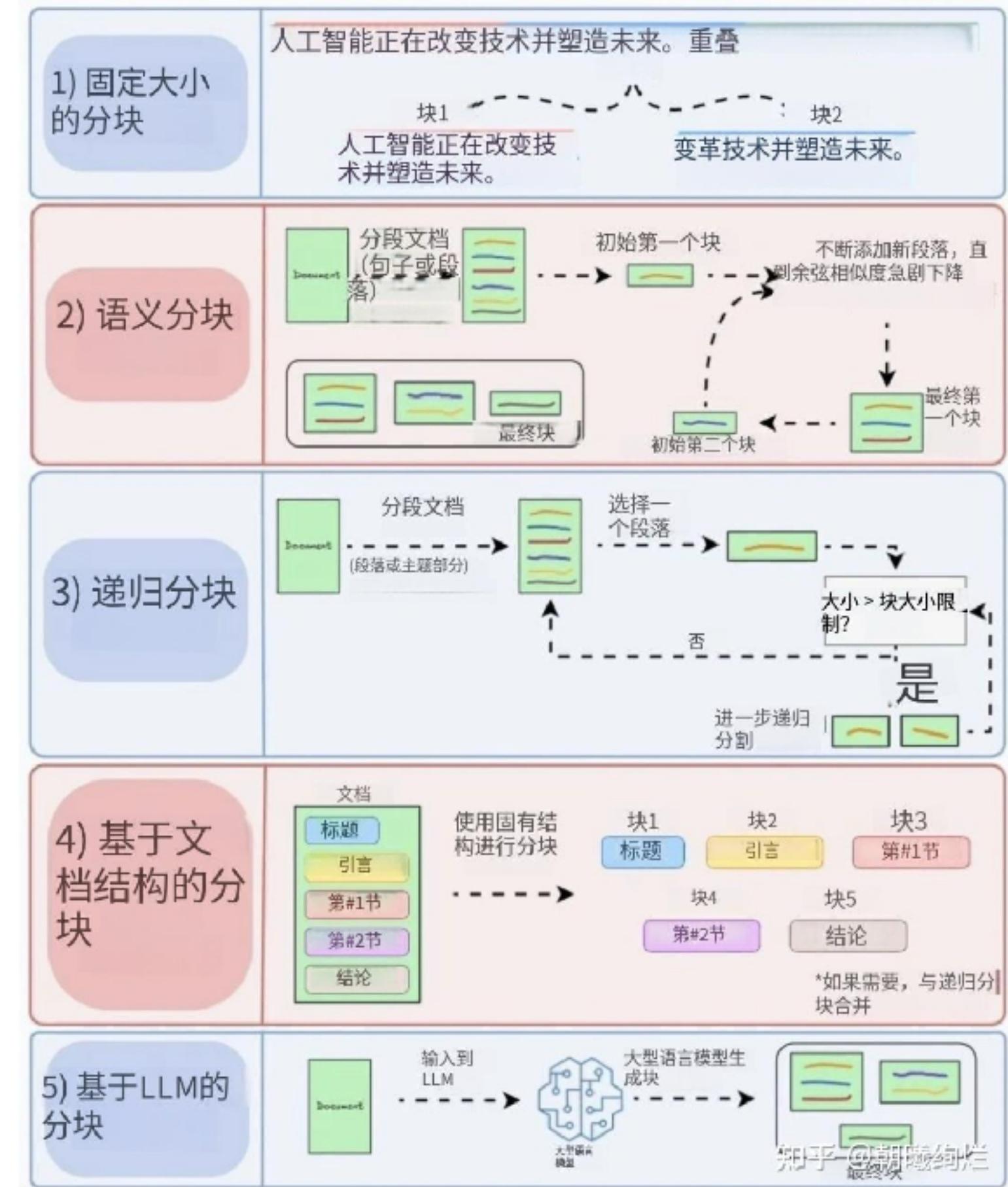
1. 游泳：既然你喜欢游泳，可以选择在周六或周日抽出1-2小时去游泳。游泳是一项全身运动，能有效锻炼心肺功能和肌肉力量，同时对关节的压力较小。
2. 羽毛球：继续保持你的周末羽毛球习惯。羽毛球不仅能提升反应速度和协调性，还能让你享受竞技的乐趣。如果可能，可以约上朋友一起打，增加互动性和趣味性。
3. 混合运动：如果时间允许，可以在一天选择游泳，另一天打羽毛球，或者在同一天内安排两种运动，例如上午游泳、下午打羽毛球，让身体得到全面锻炼。

记得运动前后做好热身和拉伸，避免受伤！

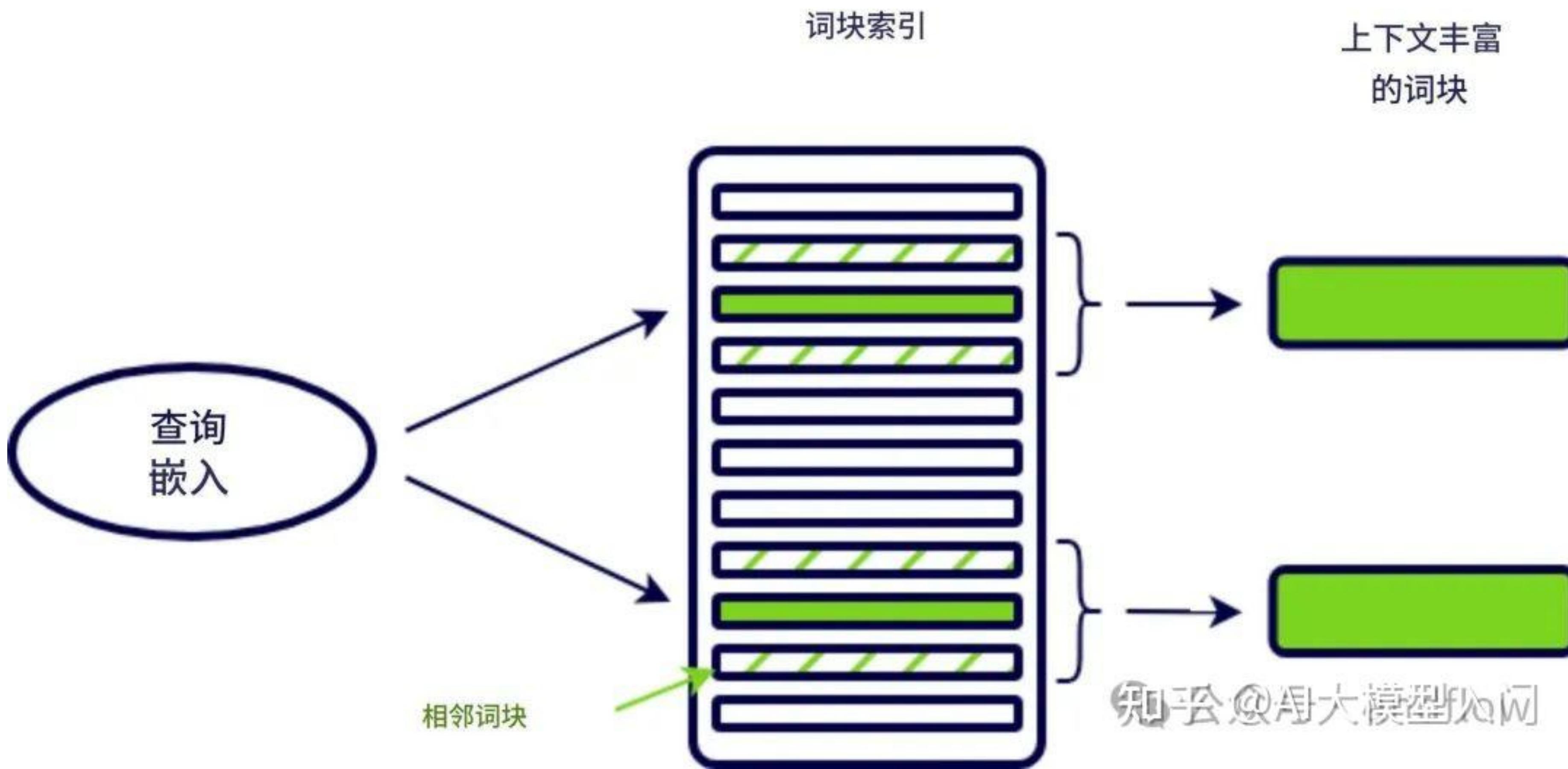
4 RAG



RAG 流程



分块/切片



上下文增强

索引模式抉择

- 经济模式特点
 - 经济模式通过每块10个关键词快速建索引，节省token消耗，但易丢失语义精度，导致长文档、多义词、口语化查询时出现漏检和误检。
- 高质量模式优势
 - 高质量模式调用嵌入模型，将文本转化为高维语义向量，实现精准匹配，尤其在医疗问诊、法律条款等严肃场景下，召回率与准确率显著提升。
- 长文本检索
 - 在长文档检索中，经济模式可能因关键词不足而漏掉关键信息；而高质量模式能完整捕捉文档语义，避免语义丢失，确保检索结果的精准性。

Embedding模型精解

- text-embedding-v4核心参数
 - text-embedding-v4模型输出维度为768，最大输入长度支持8192个token，支持批量推理。高维度向量空间能更精准地表示文本语义，提升检索精度。
 - 长文档采用‘段落+标题’滑动窗口分段，重叠20%，确保论点完整；短查询去掉停用词，保留动词和形容词，避免填充无效token，提升效率。

近义文本精准区分

- 长文本处理优化
 - 相比前代模型（如text-embedding-ada-002），text-embedding-v4在处理长文本时表现更优。它能更好地捕捉长文档中的“全局语义”（如一篇论文的核心论点），而非仅关注局部关键词，避免因文本过长导致的语义丢失。
- 细微文本区分能力
 - 对于语义相近但存在差异的文本，模型能更精准地区分。例如：
 - “如何预防感冒”（侧重预防方法）与“感冒了怎么办”（侧重治疗方法）；
 - “推荐一款便宜的手机”（价格优先）与“推荐一款耐用的手机”（质量优先）。

全文与向量双通道流程

- 双通道检索流程
 - 查询并行进入全文检索与向量检索通道，全文检索基于倒排索引匹配关键词，向量检索基于向量相似度匹配语义，两路结果在融合层按权重合并。
- 融合优势
 - 双通道检索兼顾字面匹配与语义泛化，相比单通道，Top10即返回的前十条文本命中率显著提升，确保检索结果既精准又全面，满足复杂查询需求。

语义与关键词权重调配

- 权重分配意义
 - 权重语义0.7/关键字0.3表示融合分数=0.7×向量余弦相似度+0.3×BM25得分，语义权重更高，确保检索结果更贴合语义需求。
- 权重调整方法
 - 针对同一问题多次查询，每次遍历语义权重步长0.1，选取查询结果的权重比例，确保检索效果最佳。

语义0.5/关键字0.5

关于如何缴纳学费的问题，以下是根据南京大学财务处知识库提供的信息回答：

【如何缴纳学费】

南京大学财务处于新学年开学前发布缴纳学费通知，学生需判断是否申请助学贷款：

- 若申请助学贷款，则暂不缴费，待助学贷款下发后足额缴纳学宿费。
- 若未申请助学贷款，学生在通知规定的日期前进行网上支付，支付方式有微信缴费和工行 APP 缴费：
 - 微信缴费：点击“南京大学财务处”公众号底部菜单栏中的“财务服务”，登录后在“学生缴费”模块进行缴费。
 - 工行 APP 缴费：登录中国工商银行手机 APP，点击“生活缴费”，定位“南京”，搜索“南京大学学杂费”进行缴费。

缴费完成后约一周后可以下载电子票据，下载方式有微信端和 PC 端：

- 微信端：点击“南京大学财务处”公众号底部菜单栏中的“财务服务”，登录后在“电子发票”模块下载票据。
- PC 端：校内 IP 登录财务处网站（ndcw.nju.edu.cn），登录后进入“财务查询系统”下载票据。

如需了解更详细信息，请在南京大学校园网环境下访问 <https://ndcw.nju.edu.cn/>。

引用

流程.docx

语义0.7/关键字0.3

您好，根据南京大学财务处提供的自助缴纳学宿费及取票流程，以下是缴纳学费的具体步骤：

- 判断是否申请助学贷款：
 - 若申请助学贷款，则暂不缴费，待助学贷款下发后足额缴纳学宿费。
- 若不申请助学贷款，网上支付：
 - 微信缴费：登录“南京大学财务处”公众号，点击“财务服务”模块进行缴费。
 - 工行 APP 缴费：登录中国工商银行手机 APP，选择“生活缴费”，定位“南京”，搜索“南京大学学杂费”进行缴费。
- 缴费完成后下载电子票据：
 - 微信端：登录“南京大学财务处”公众号，点击“财务服务”模块下载票据。
 - PC 端：校内 IP 登录财务处网站（ndcw.nju.edu.cn），登录后进入“财务查询系统”下载票据。

对于学宿费退费流程，具体如下：

- 学费退费：需准备学校关于学生退学的正式发文（复印件）、书面退费申请（由院系签字并盖章）、学费纸质收据原件或电子收据打印件。
- 住宿费退费：需准备资产管理处审核确认的退款单、住宿费纸质收据原件或电子收据打印件。

办理地点为行政北楼东财税科（211 室）或财务处苏州校区办公室，联系电话：025 - 89683149、025 - 83593149、0512 - 68768008。

请注意，以上信息来源于南京大学财务处官方指南，如需了解更详细信息，请在南京大学校园网环境下访问 <https://ndcw.nju.edu.cn/>。

引用

流程.docx

可以看到语义0.7的查询结果在分点排版上，划分步骤上明显优于语义0.5的查询结果，但是却多出学宿费退费的不相关信息。考虑引入score阈值筛除不相关信息

调配结果对比

Rerank模型二次精排

- Rerank模型作用
 - 对“初步检索得到的候选文档列表”，根据「候选文档与用户问题的语义匹配度」再次排序，进一步优化“语义层面的排序效果”，让最相关的文档排在更前的位置。

Top K与Score阈值联动

- TopK：用于筛选与用户问题相似度最高的文本片段。系统同时会根据选用模型上下文窗口大小动态调整片段数量。默认值为 3，数值越高，预期被召回的文本分段数量越多。
- Score 阈值：用于设置文本片段筛选的相似度阈值，只召回超过设置分数的文本片段，默认值为 0.5。数值越高说明对于文本与问题要求的相似度越高，预期被召回的文本数量也越少。
- 调参策略
 - 高价值小库可降K至5、阈值升至0.35，低价值大库可升K至20、阈值降至0.2，灵活调整以兼顾效率与效果。
- Top K与Score阈值协同
 - Top K=10确保重排阶段有足够的候选， $\text{Score} \geq 0.25$ 过滤低相关结果，减少大模型上下文噪声，提升最终结果质量。最终效果如右图所示



调整后结果

5. MCP

MCP案例 1

目标:

让 LLM 通过 MCP 接口访问一个“课程数据库”，查询课程名称、学分和任课教师。

一、安装依赖

在项目目录下执行:

```
pip install mcp fastapi uvicorn
```

二、编写 MCP 服务器 **course_server.py**

```
# course_server.py
from mcp.server.fastmcp import FastMCP
from mcp.types import TextContent

# 创建 MCP 服务实例
mcp = FastMCP("course-service")

# 模拟一个小型课程数据库
COURSES = {
    "SE101": {"name": "软件工程导论", "teacher": "刘钦", "credit": 3.0},
    "OOP202": {"name": "面向对象程序设计", "teacher": "王蕾", "credit": 4.0},
    "AI300": {"name": "人工智能基础", "teacher": "张伟", "credit": 3.5}
}
```

```
@mcp.tool()
def query_course(course_id: str) -> TextContent:
    """
    查询课程信息（通过课程编号）。
    """
    if course_id not in COURSES:
        return TextContent(text=f"未找到课程编号 {course_id}。")
    info = COURSES[course_id]
    return TextContent(
        text=f"{info['name']} (课程号: {course_id}) 由 {info['teacher']} 授课, 学分: {info['credit']}。"
    )

@mcp.tool()
def list_courses() -> TextContent:
    """
    列出所有课程。
    """
    lines = []
    for cid, info in COURSES.items():
        lines.append(f"{cid} - {info['name']} ({info['teacher']})")
    return TextContent(text="\n".join(lines))

if __name__ == "__main__":
    mcp.run()
```

MCP案例 1

三、创建 MCP 配置文件 `mcp.json`

如果你在 ChatGPT（支持 MCP 的版本）或你自己的 LLM 环境中运行，可配置如下：

```
{  
  "mcpServers": {  
    "course-service": {  
      "command": "python",  
      "args": ["course_server.py"]  
    }  
  }  
}
```

四、效果演示

启动服务后，模型就能调用 MCP 接口：

用户对话示例：

用户：有哪些课程可选？

→ MCP 调用 `list_courses()`
→ 返回：

SE101 - 软件工程导论 (刘钦)
OOP202 - 面向对象程序设计 (王蕾)
AI300 - 人工智能基础 (张伟)

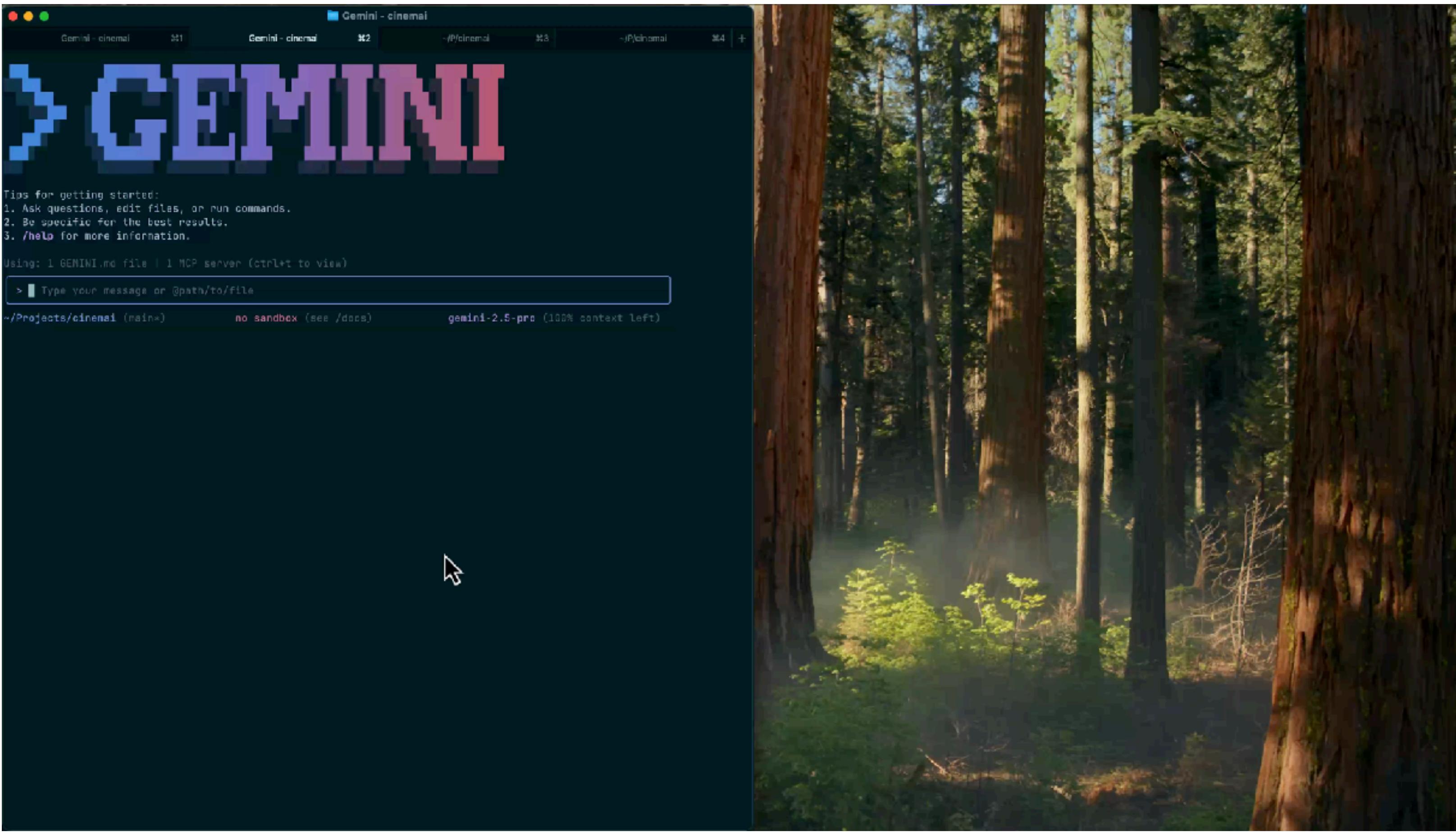
用户：AI300 这门课是谁教的?
→ MCP 调用 `query_course(course_id="AI300")`
→ 返回：
人工智能基础 (课程号：AI300) 由 张伟 授课，学分：3.5。

五、进阶说明

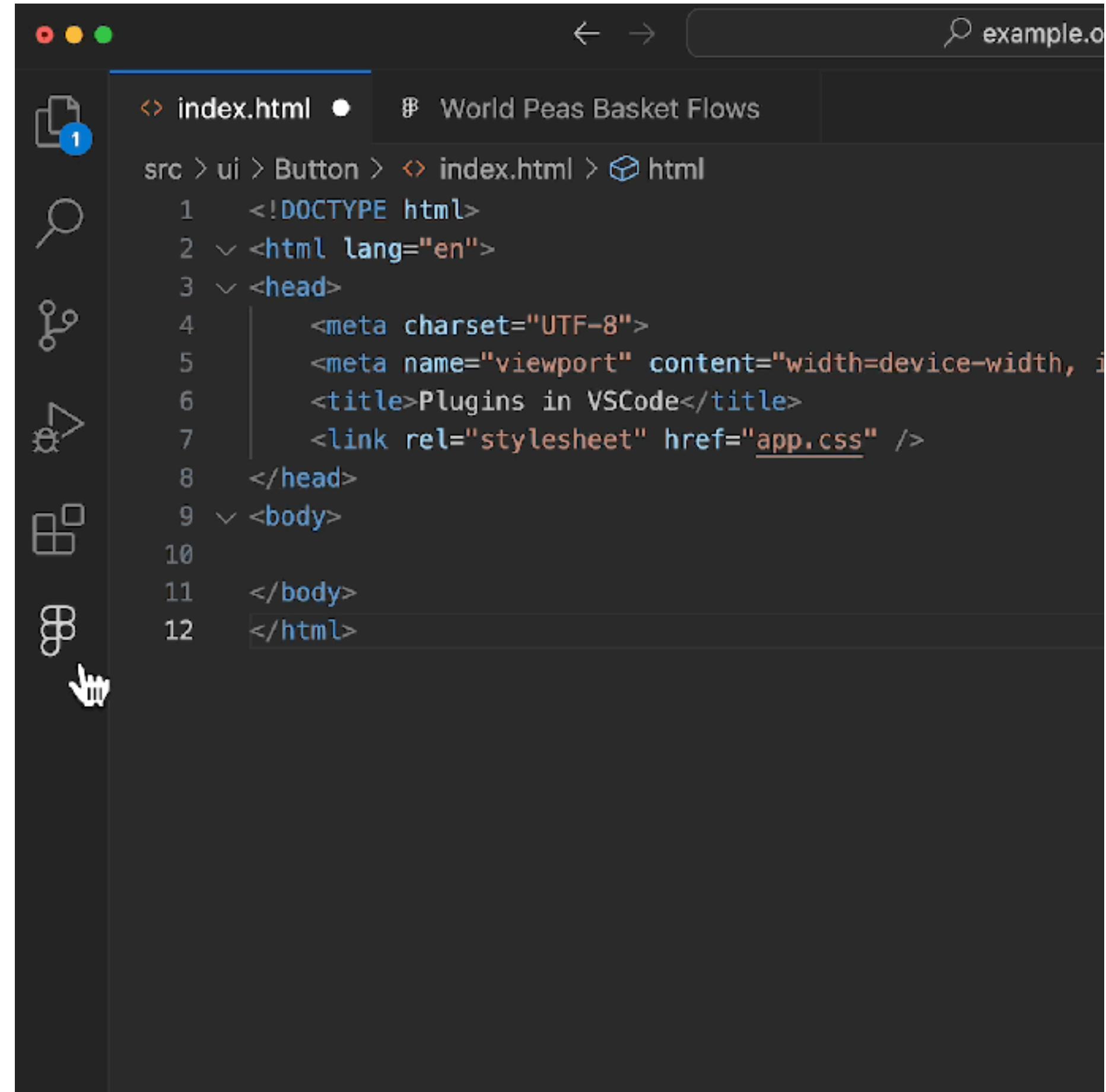
- `@mcp.tool()` 装饰器会自动将函数暴露给模型调用。
- 返回类型使用 `TextContent`，也可以返回 `JsonContent`（结构化结果）。
- MCP 的通信基于 JSON-RPC，可以通过标准输入输出进行通信，安全且高效。
- 你可以用它把数据库、文件系统、API、内部知识库等都接入 LLM。

MCP案例 2 – 服务器代码

- from fastmcp import FastMCP
- }
- # 2. 定义一个**工具（Tool）**，用于执行特定动作
- # 工具使用`@mcp_server.tool()`装饰器，并接收参数
- @mcp_server.tool()
- def get_weather(city: str, unit: str = "celsius") -> str:
- """
- 获取指定城市的天气。
- Args:
- city: 要查询天气的城市。
- unit: 温度单位，'celsius' 或 'fahrenheit'。默认为 'celsius'。
- Returns:
- 包含天气信息的字符串。
- """
- # 这里我们只返回一个模拟结果，实际应用中会调用天气API
- if city.lower() == "london":
- return f"Weather in London: 15°{unit[0].upper()}\nand rainy."
- elif city.lower() == "tokyo":
- return f"Weather in Tokyo: 22°{unit[0].upper()}\nand sunny."
- else:
- return "Weather data not available for this city."
- # 如果直接运行此脚本，将启动服务器
- if __name__ == "__main__":
- print("Starting Weather MCP Server on standard I/O...")
- mcp_server.run(transport="stdio")
-



Chrome DevTools (MCP) for your AI agent



```
index.html  ●  World Peas Basket Flows
src > ui > Button > index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, i
6      <title>Plugins in VSCode</title>
7      <link rel="stylesheet" href="app.css" />
8  </head>
9  <body>
10
11 </body>
12 </html>
```

Figma MCP server

6. 工作流

工作流

工作流应用将复杂的任务拆分成一系列有序执行的步骤，以降低系统复杂度。在阿里云百炼，通过工作流组合使用大模型、API和函数计算等节点，可有效降低编码成本。本文介绍如何创建工作流。

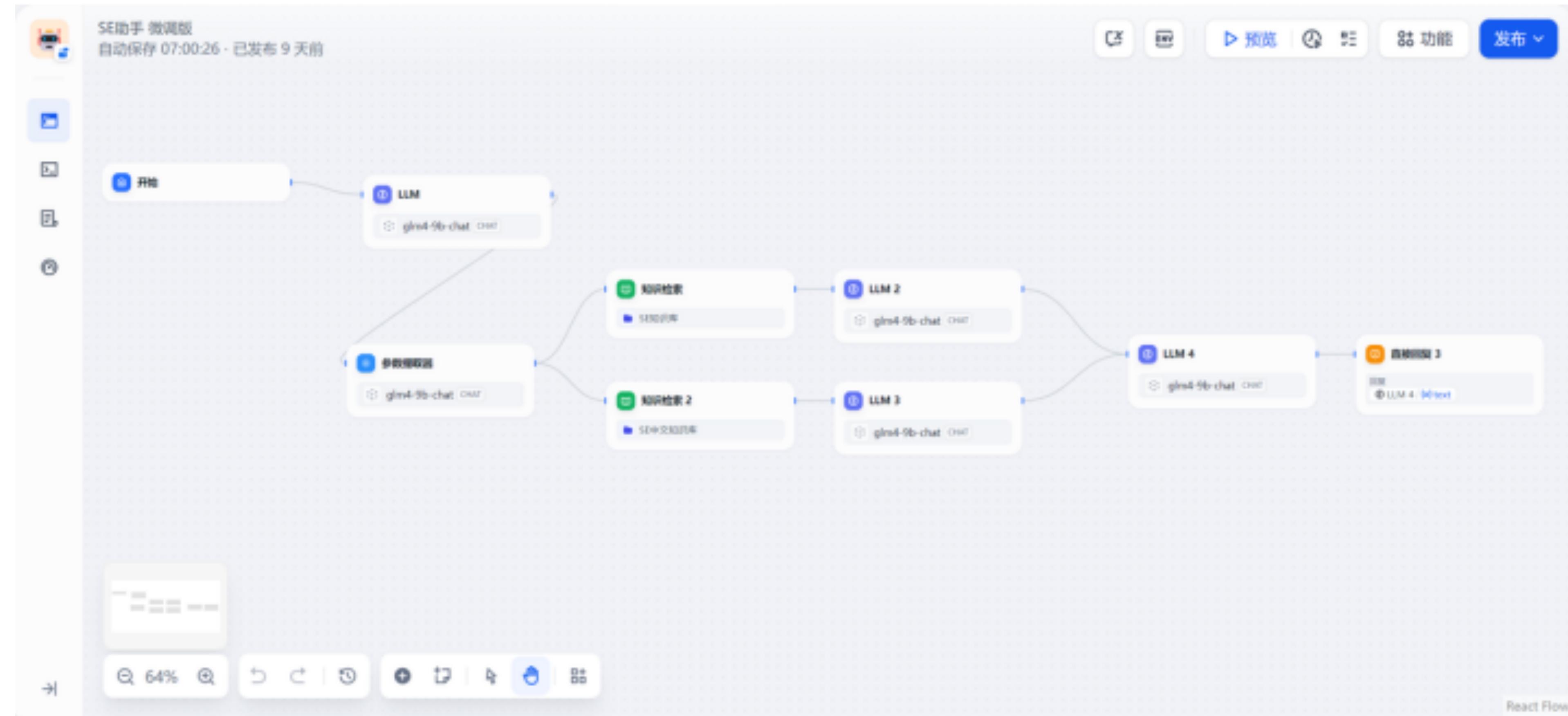
应用介绍

为什么使用工作流应用

工作流是一种将复杂任务拆分为一系列有序步骤的方法，旨在简化系统复杂度，提高工作效率。在现代软件开发和业务流程管理中，工作流应用变得尤为重要。通过在阿里云百炼平台上创建工作流应用，可以清晰地定义任务的执行顺序、责任分配以及各步骤之间的依赖关系，从而实现自动化和优化。

工作流应用有许多使用场景，如：

- 旅行规划**：用户可通过工作流插件选择目的地等参数，自动生成旅行计划，包括航班、住宿、景点推荐等。
- 报告分析**：针对复杂数据集，通过组合数据处理、分析和可视化插件，生成结构化和格式化的分析报告，满足不同业务需求。
- 客服支持**：通过自动化工作流处理客户咨询，包括问题分类等，提高客服响应速度和准确性。
- 内容创作**：实现文章、市场营销文案等内容的生成，用户只需输入主题和要求，系统自动生成符合要求的文稿。
- 教育培训**：通过工作流设计个性化学习方案，包括学习进度跟踪、测评等，实现学生的自主学习。
- 医疗问诊**：根据患者输入的症状，通过组合多种分析工具生成初步诊断或推荐相关检查，辅助医生进行进一步判断。



流程的编排

The screenshot shows a user interface for a knowledge management system. At the top, there is a navigation bar with tabs: 探索 (Explore), 工作室 (Studio), 知识库 (Knowledge Base) (which is highlighted in blue), and 工具 (Tools). On the far right of the top bar is a user profile icon with the name Qin LIU.

Below the navigation bar, there is a search bar with a dropdown menu labeled '全部标签' (All Tags) and a search input field. To the right of the search bar is a link labeled '外部知识库 API' (External Knowledge Base API).

The main content area displays three knowledge base cards:

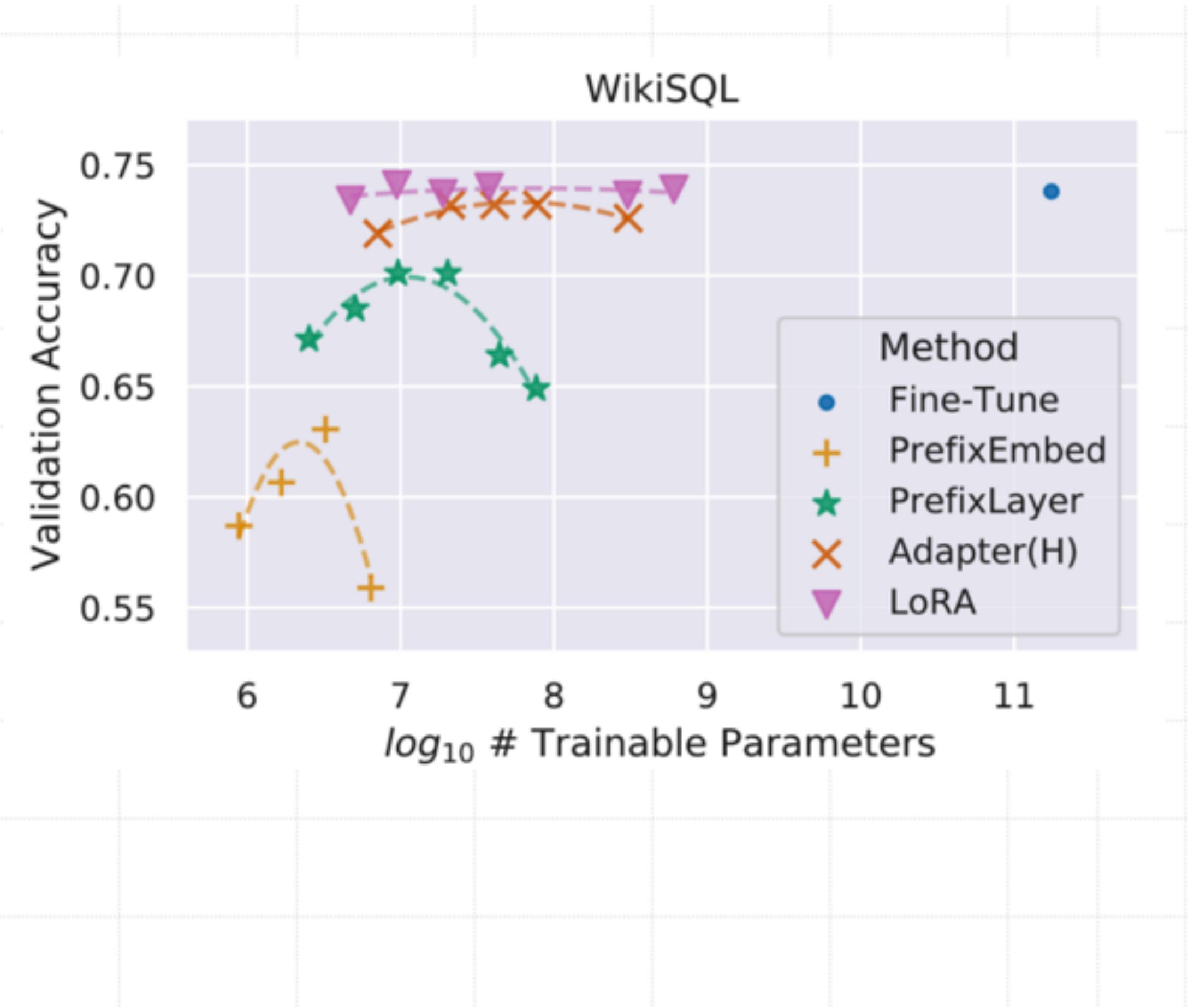
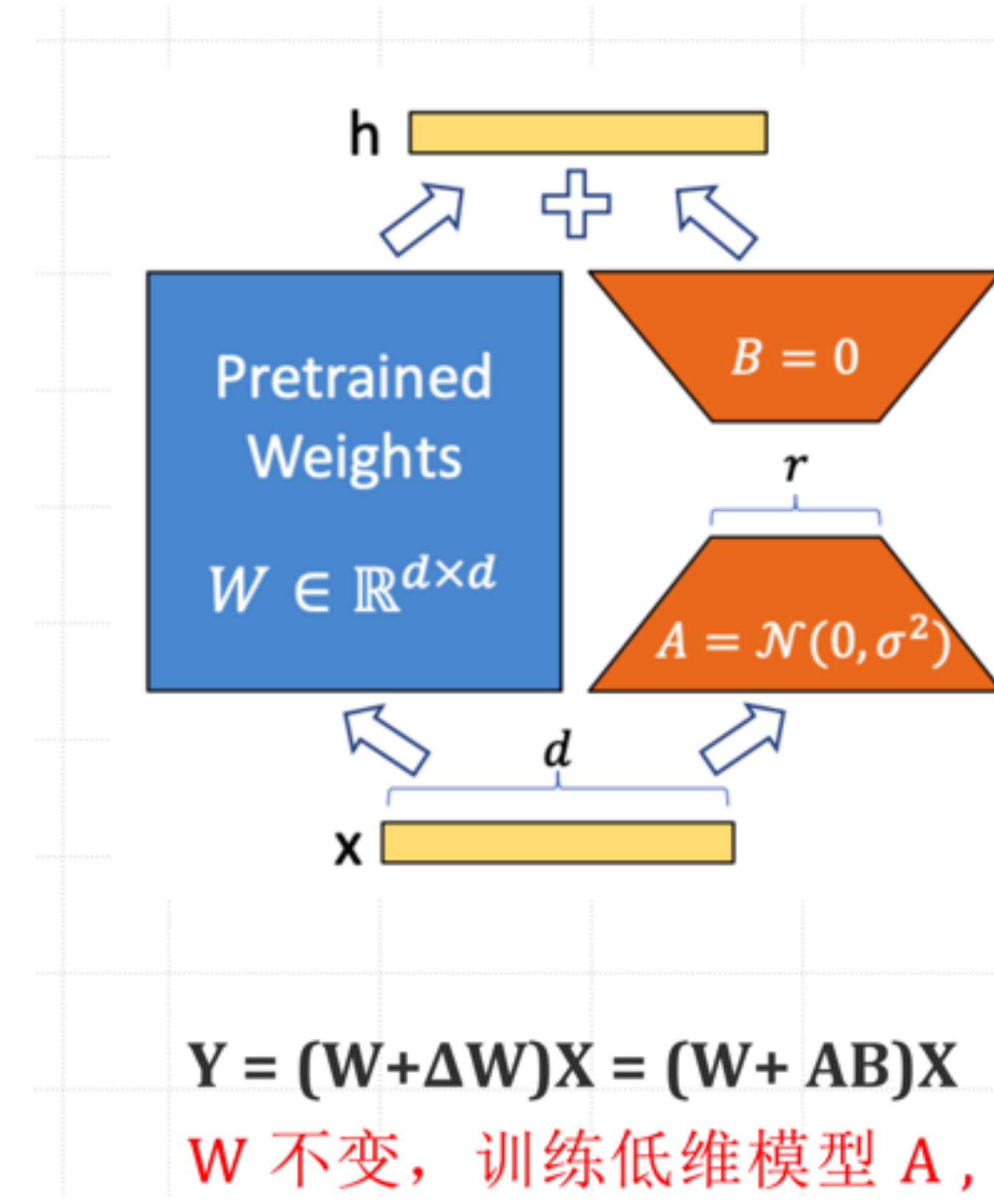
- SE中文知识库**: Shows 1 文档 (1 document), 425 千字符 (425 thousand characters), and 4 关联应用 (4 associated applications). It includes a brief description: "It includes documentation, tutorials, FAQs, and other relevant resources that are essential for understanding and utilizing HarmonyOS effectively. A detailed..." followed by a '鸿蒙' (HarmonyOS) tag.
- SE知识库**: Shows 256 文档 (256 documents), 162274 千字符 (162,274 thousand characters), and 4 关联应用 (4 associated applications).
- HarmonyOS 知识库**: Shows 15 文档 (15 documents), 82374 千字符 (82,374 thousand characters), and 3 关联应用 (3 associated applications).

知识库



知识检索

7. 微调



LoRA (Low-Rank Adaptation)

通过分解预训练模型中的部分权重矩阵为低秩矩阵，并仅微调这些低秩矩阵的少量参数来适应新任务。

LoRA参数

- LoRA参数主要包括秩 (lora_rank, 影响性能和训练时间)、缩放系数 (lora_alpha, 确保训练稳定) 和Dropout系数 (lora_dropout, 防止过拟合)，它们共同影响模型微调的效果和效率。

1. 秩 (Rank)

- 参数名称: lora_rank
- 描述: 秩是LoRA中最重要的参数之一，它决定了低秩矩阵的维度。
- 常用值: 对于小型数据集或简单任务，秩可以设置为1或2；对于更复杂的任务，秩可能需要设置为4、8或更高。

2. 缩放系数 (Alpha)

- 参数名称: lora_alpha
- 描述: 缩放系数用于在训练开始时对低秩矩阵的更新进行缩放。
- 常用值: 缩放系数的具体值取决于秩的大小和任务的复杂度。

3. Dropout系数

- 参数名称: lora_dropout
- 描述: Dropout系数决定了在训练过程中随机丢弃低秩矩阵中元素的概率。
- 常用值: Dropout系数的常用值范围在0到1之间。



微调的基底大模型的选择

- 1、首先选择那些在通用任务中表现优异且在各种基准测试中取得高分的预训练模型，同时并确保其参数量和计算资源需求在可承受范围内。
- 2、选择具有良好社区支持和文档丰富度的LLM也是关键，可以保证在微调中能够提供及时帮助和丰富资源。
- 3、模型应具备良好的扩展能力，并与现有工具和框架兼容，以便于集成和部署。
- 4、预训练数据与目标领域数据的匹配度和模型在训练与推理过程中的效率也需考虑。

微调的任务和数据集

- 教育模型目标任务
 - 1. 问答系统：学生问题的答案生成。
 - 2. 教学推荐：根据学习路径推荐课程或资源。
 - 3. 作业批改：自动评分、反馈生成。
 - 4. 知识点提取：帮助学生定位具体的知识点。
 - 5. 情感分析：监测学生的情绪和参与度。【无相关数据集】
- 数据集
 - 公开数据集
 - - 知识问答：如 SQuAD、TriviaQA。
 - - 考试题库：如 CLEF QA、MCTest 等。
 - 领域数据
 - - 自有数据：从教育平台、学习管理系统（LMS）或在线课程中采集数据。
 - - 合作伙伴数据：与学校、教育公司合作获取特定领域数据。

微调的数据要求

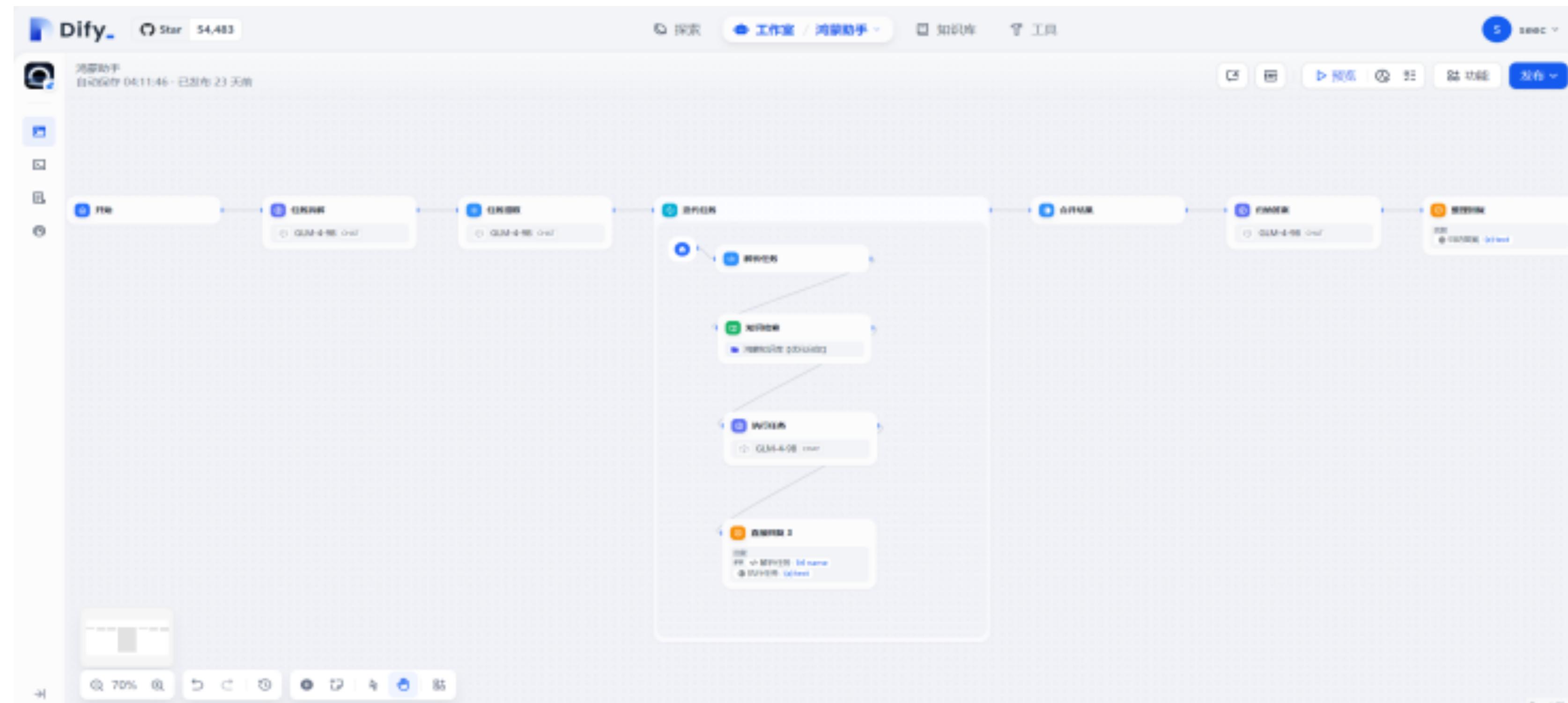
- 现在通过大模型微调实践，在具体使用数据进行大模型微调时目前通常达成的共识一般由以下几点：
 - 1. 在设计训练数据时数据的质量远比数据的数量要重要的多。
 - 2. 为防止灾难性遗忘，在微调时通常会考虑将专业领域知识和通用领域知识混合在一起进行训练。
 - 3. 如果涉及二次预训练，需要将一般添加5-10倍的原始训练的数据集中的数据。
 - 4. 在微调过程找出模型学习不好的那部分数据（对应模型PPL值变小）进行质量判断后进行针对性的加强训练。

四、应用案例

1. 教育大模型

AI项目--SEEC LLM 教育大模型

- SEEC LLM+EDU 智能教育平台深度整合检索增强生成（RAG）与数据微调技术，打造面向本科教育的AI学习助手。通过Dify平台的知识库构建与流程编排能力，集成开源鸿蒙生态技术资料，开发出“开源鸿蒙智能助手”，实现精准的课程资源匹配与实时技术答疑。
- 在软件工程专业教学中支撑开源鸿蒙系统开发实践，同时赋能南京大学跨学科通识课程的数字化升级。





RAG + 微调



鸿蒙o1

解释ArkTS

新对话

解释ArkTS

什么是ArkTS?



工作流

步骤1：定义ArkTS

ArkTS 是一种为构建高性能应用而设计的编程语言。它在 TypeScript 的基础上进行了优化，旨在提供更高的性能和开发效率。ArkTS 主要针对移动设备环境进行优化，以解决移动应用运行缓慢、低效、功耗大等问题。

以下是 ArkTS 的一些关键特性：

1. 静态类型：与 TypeScript 类似，ArkTS 也引入了静态类型的概念，但 ArkTS 会将编译时确定的类型应用到运行性能优化中，从而提高执行效率。
2. 性能优化：ArkTS 通过取消动态类型特性，使代码能更有效地被编译和优化，实现更快的应用启动和更低的功耗。
3. 兼容 TypeScript/JavaScript：ArkTS 兼容 TypeScript/JavaScript 生态系统，开发者可以使用 TypeScript/JavaScript 进行开发或复用已有代码。
4. 面向未来：ArkTS 会结合应用开发/运行的需求持续演进，引入包括并行和并发能力增强、类型系统增强等方面的语言特性。

总之，ArkTS 是一种专注于移动设备性能优化的编程语言，在 TypeScript 的基础上提供更高的性能和开发效率。

步骤2：ArkTS的功能和用途

0



模型部署和使用



课程广场 我的课程 个人资料

李老师



我教授的课程

查看和管理您教授的所有课程

创建课程

搜索课程名称

搜索

测试课程

教师: 数据科学 已选人数: 1

创建课程

创建于: 2025年3月26日

查看详情

编辑课程

软件工程与计算I

教师: 张三 软件工程 已选人数: 0

软工1

创建于: 2025年3月18日

查看详情

编辑课程

软件工程与计算II

教师: 张三 软件工程42 已选人数: 0

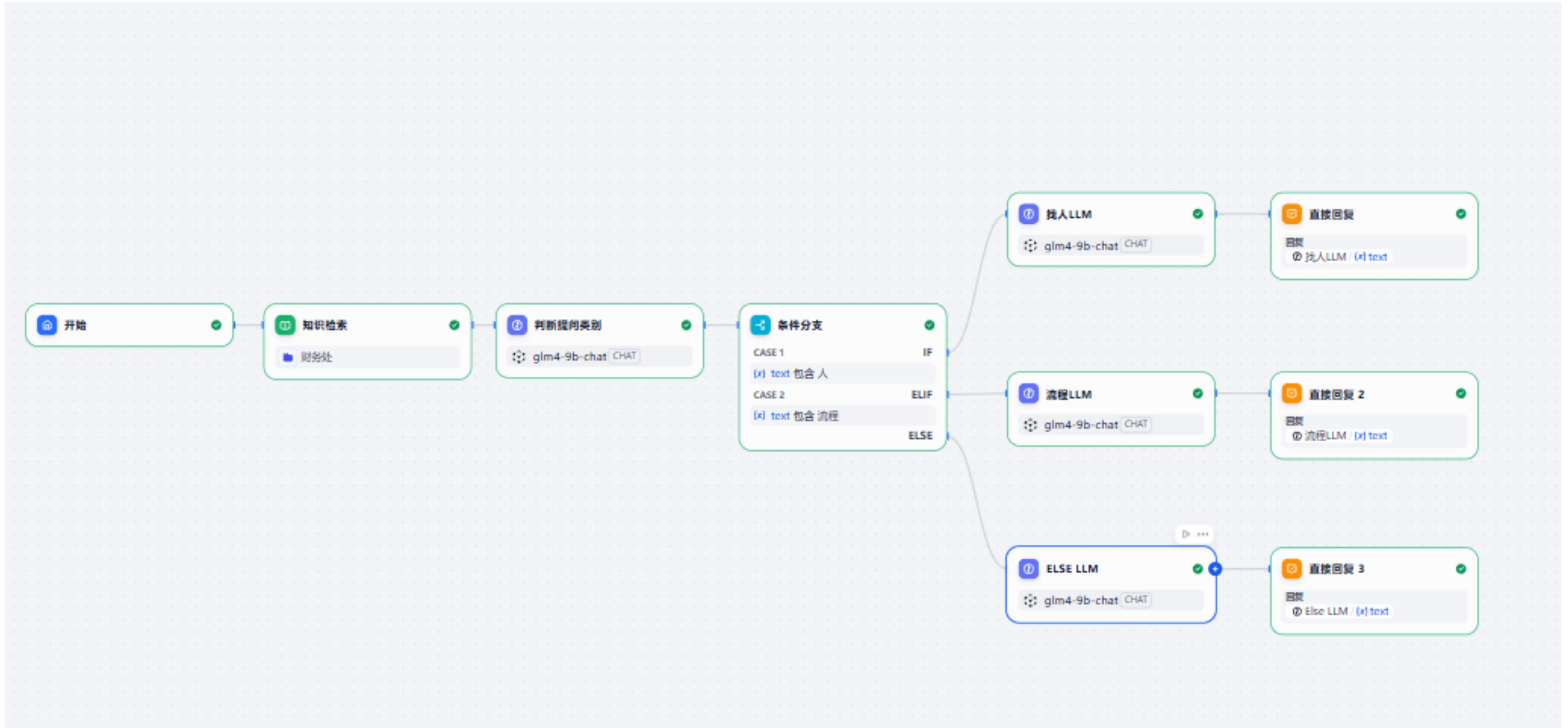
软件工程与计算II

创建于: 2025年3月18日

查看详情

编辑课程

2. 财务问答助手



dify流程

流程分支

- 在进入知识库进行检索之前，通过大模型将问题分为“找人”、“问流程”和“其他”三个问题种类。
- 在找人部分中，我们可以精确到具体的办公岗位和联系方式。
- 在问流程的部分中，我们可以提供具体的信息来源并且在涉及到数据的部分，我们会分布进行计算并提供免责声明。



当用户询问某个人（如老师、财务处工作人员）时，应尽可能根据知识库内容介绍该人的基本情况（姓名、职务、联系方式、职责范围等），并标明信息出处，可提示用户在南京大学校园网环境下访问“<https://ndcwc.nju.edu.cn/>”获得进一步信息。

找人

问流程

- 在询问流程的这个工作流中我们还细分为“一般的流程问题”、“询问可不可以、满不满足”的问题和“数据相关的问题三个部分”



嵌入模型

- 相比前代模型（如text-embedding-ada-002），text-embedding-v4在处理长文本时表现更优。它能更好地捕捉长文档中的“全局语义”（如一篇论文的核心论点），而非仅关注局部关键词，避免因文本过长导致的语义丢失。
- 对于语义相近但存在差异的文本，模型能更精准地区分。例如：
 - “如何预防感冒”（侧重预防方法）与“感冒了怎么办”（侧重治疗方法）；
 - “推荐一款便宜的手机”（价格优先）与“推荐一款耐用的手机”（质量优先）。

重排

- 对“初步检索得到的候选文档列表”，根据「候选文档与用户问题的语义匹配度」再次排序，进一步优化“语义层面的排序效果”，让最相关的文档排在更前的位置。
- 用权重分配的方式，通过score阈值筛选结果
- 测试了gte-rerank-v2以及bce-reranker-base两个rerank模型，效果并不符合预期，未来考虑进一步测试其他模型。

整体效果

- 对于收集到的79个问题，有超过70%的问题可以得到正确的答复，有超过50%的问题可以得到与反馈回答几乎一模一样的回答。
- 对于部分问题，会和之前的财务处问答系统一样回答错误。
- 还有一些问题，由于爬虫的局限性会反馈知识库中暂无此信息。

案例 1

- 1. 对于询问“如果差旅中仅报销住宿费，没有其他证明材料，可以领取差旅补贴吗？”
- 原回答：（错误）
 - 如果你仅报销住宿费，没有其他证明材料，仍然可以领取差旅补贴（伙食补助费和市内交通费），但需要提供详细的书面说明和住宿发票，并经过单位财务负责人的审批。”而改进之后我们可以正确回答
- 现回答：（正确）
 - 根据您提供的信息和知识库内容，以下是对您问题的回答：
 - 如果差旅中仅报销住宿费，没有其他证明材料，一般情况下是不能领取差旅补贴的。根据《南京大学国内差旅费管理办法》第三十条的规定，出差人员应在出差结束后一个月内完成报销手续。差旅费报销应提供差旅费报销单及相关报销要件，与出差无关的费用一律不得报销。原则上不单独报销外地住宿费。
 - 因此，若仅提供住宿费发票而没有其他证明材料，可能无法完成差旅费报销。具体报销政策可能会有变动，建议您在南京大学校园网环境下访问 <https://ndcwc.nju.edu.cn/> 获取最新信息。

案例 2

- 2. 对于询问“国库经费可以报销个人手机话费吗？”
- 原回答：（**错误**）
 - 根据南京大学的财务报销管理办法，国库经费可以报销手机话费，但需要满足以下条件：报销对象为课题相关人。提供手机通讯费月结账单。发票抬头应包含相关人员姓名及手机号。一次性报销金额一般不超过500元，特殊情况需提供说明。
- 现回答：（**正确**）
 - 国库经费不可以报销个人手机话费。根据南京大学财务处知识库的规定，电话费（含办公电话和专业通讯费）的报销对象原则上应为课题相关人，每人至多报销两部手机的通讯费用，且一次性报销手机充值发票单月不超过500元。个人家庭电话费、网络费、购买电话卡及滞纳金均不予报销。（来源：南京大学财务处知识库）”

案例 3

- 3. 老师可以报销市内常驻地地铁票吗?
- 原回答: **(错误)**
 - “老师可以报销常驻地市内的地铁票，但需要遵循以下步骤：将地铁票横向粘贴在专用的票据粘贴单（或A4纸）上，并确保排列顺序与所列支出明细一致。如果使用竞争性科研经费报销，单张发票金额200元以下可直接报销，200元以上的须提供清单或行程单。如果使用非竞争性科研经费报销，无论金额大小均须提供清单或行程单，并填写报销事由。项目负责人使用竞争性科研经费报销本人的市内交通费，需学院财务负责人审批。其他情况下，报销单需要项目负责人签字确认，并由单位财务负责人审批。”
- 现回答: **(错误)**
 - 根据《南京大学财务报销指南》的相关规定，常驻地市内交通费报销要求如下：
 - (十一) 常驻地市内交通费
 - 1. 报销要求
 - (1) 报销市内交通费时需将发票横向粘贴在专用的票据粘贴单(或A4纸)上，排列顺序要与所列支出明细一致
 - (2) 学生实习、实训发生的市内交通费（含地铁、公交票）根据实习、实训计划、任务，由学生所在学院统一报销
 - (3) 教职工地铁、公交充值定额发票不予报销。
 - (4) 外地市内交通费须与出差旅费一同报销，原则上不予单独报销。
 - 因此，老师可以报销在常驻地发生的市内交通费，包括地铁票，但教职工地铁、公交充值定额发票不予报销。外地市内交通费需与出差旅费一同报销。信息来源：《财务报销指南》第（十一）条。如需了解更详细信息，请在南京大学校园网环境下访问 <https://ndcwc.nju.edu.cn/>。

案例 4

- 国际差旅报销需要提供护照及签证复印件吗？
- 现回答：（[错误](#)）
- 国际差旅报销时，根据知识库内容，如果您需要提供护照及签证复印件，知识库中并未明确提及。但根据国际差旅费的相关规定，以下信息是必须提供的：
 - - 《因公出国（境）费用报销汇总表》（以下简称“汇总表”）缺少签字盖章。
 - - 缺少出国（境）任务批件。国际差旅费报销应提供出国（境）任务批件，确有特殊情况无法提供的，应提供情况说明并经国际处审批。
 - - 缺少《经费来源变更审批表》。若出国（境）任务批件中的经费来源与实际报销使用的经费号不一致，应提供《经费来源变更审批表》。
 - - 签证费未提供发票。美国签证费无法提供发票的，应提供面签材料“APPOINTMENT CONFIRMATION”；英国签证费无法提供发票的，应提供签证费的支付记录以及支付成功的邮件截图。
- 关于护照及签证复印件的具体要求，建议您参考《财务报销指南》或联系南京大学财务处获取更详细的信息。如需了解更详细信息，请在南京大学校园网环境下访问 [https://ndcwc.nju.edu.cn/。](https://ndcwc.nju.edu.cn/)”
- 这个回答答非所问且知识库中没有明确的信息，这是我们后续需要改进的。

后续工作

- 后续采取收集信息->微调大模型->更新知识库的自动化流程
 - 需要2000条及以上的问题以及回答)
- 后续可以给出多条建议输出，由老师来选择更符合预期的输出来进一步优化大模型
- 目前财务处官网中还有一些内容由于爬虫的限制没有成功爬取，后续我们会进一步爬取包括视频格式在内的网站内容

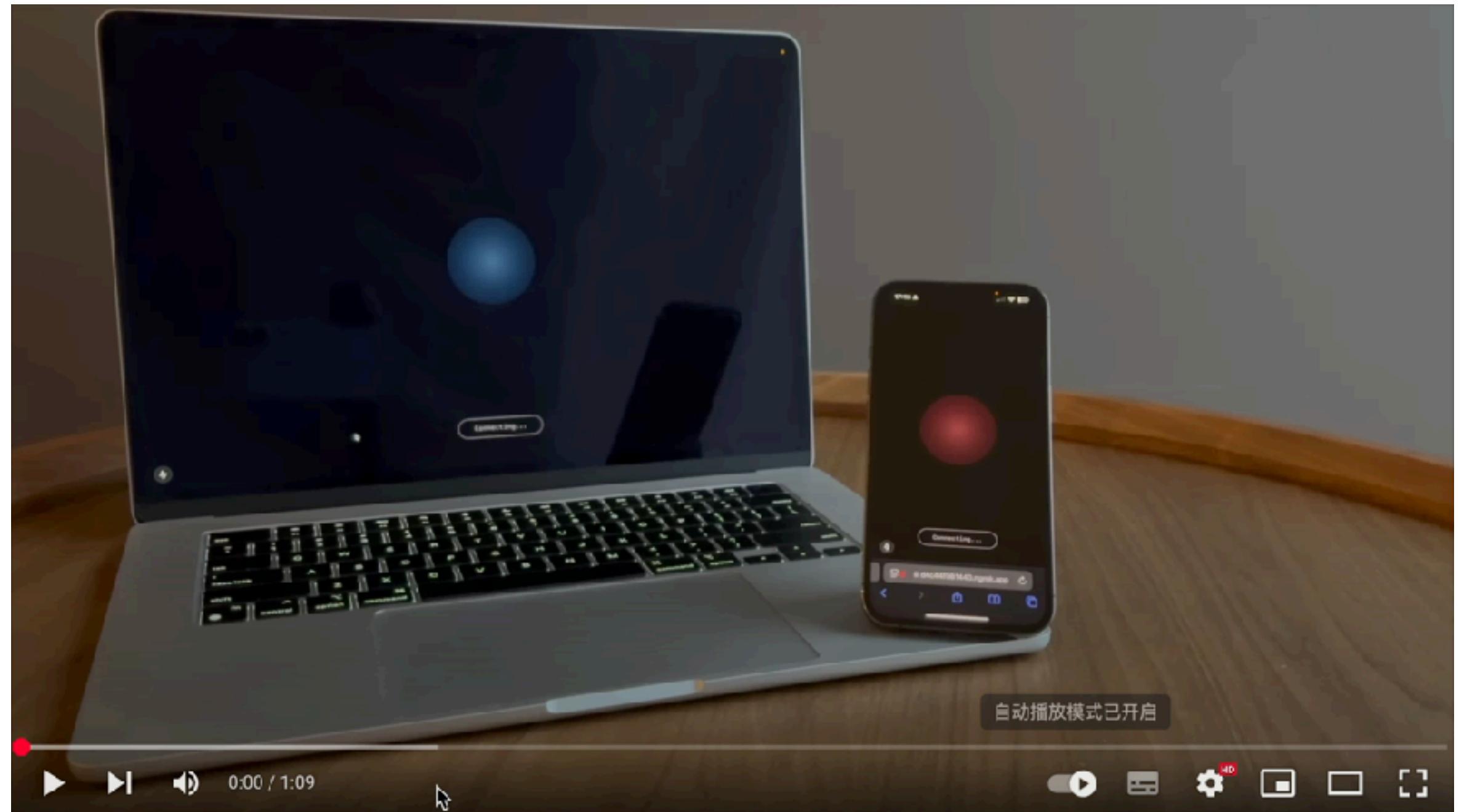
五、展望

思考的问题

- “Agent是否正在取代传统的软件工程流程？”
- “从SaaS到Agent-as-a-Service，开发范式将如何变化？”
- “推理模型（System 2）会不会催生‘自我评审型AI’？”

展望

- 单Agent
 - 自主性
 - 上下文
- 多Agent
 - 协同
 - 决策
- 统一框架
- 强化学习道路
- Timeless的东西：需求的涌现+现实的复杂度



实践作业

华为平台

- RAG构建专业知识问答机器人实验
 - https://lab.connect.huaweicloud.com/experiment-detail_20006902
- 基于DeepSeek和Dify实现智能拍照解题
 - https://lab.huaweicloud.com/experiment-detail_4238
- 基于DeepSeek和Dify构建旅游攻略ppt助手
 - https://lab.huaweicloud.com/experiment-detail_7377
- 基于华为云开发者空间开发Agent实验
 - https://lab.connect.huaweicloud.com/experiment-detail_20006943

阿里云平台

- 百炼平台
- 通义灵码