

《物理与人工智能》

3. 大语言模型深入使用

授课教师：马滢青

2026/09/21（第三周）



北京大学



□ LLM的局限性

- 幻觉现象 (Hallucination)
- 知识滞后性
- 提示敏感性
- 模型偏见与伦理考量

□ RAG架构

□ Prompt工程

□ 工具调用

□ Skills设计

幻觉现象 (Hallucination)

Deepseek

北京大学物理学院理论物理研究所的老师名单



好的，这是北京大学物理学院理论物理研究所的教师名单。

请注意，大学的人事信息可能会有变动。以下名单主要基于北京大学物理学院官方网站公布的信息整理，涵盖了不同职称的教师（教授、副教授、研究员等）。

教授/研究员 (博导)

这些是研究所的核心师资力量，通常为全职教授和博士生导师。

1. 李定平 - 所长，研究方向：理论生物物理、统计物理、复杂系统。
2. 曹庆宏 - 研究方向：粒子物理理论、超出标准模型新物理、对撞机物理。
3. 陈斌 - 研究方向：弦理论、量子引力、宇宙学。
4. 高宇彦 - 研究方向：统计物理、软凝聚态物理、生物物理。
5. 郭秋菊（女） - 研究方向：中微子物理、粒子天体物理。
6. 李新征 - 研究方向：计算凝聚态物理、分子动力学、材料物性理论模拟。

幻觉现象 (Hallucination)



一根长度为 α 的无质量绝缘细绳一端固定，另一端悬挂一个质量为 m 、半径为 r ($r \ll \alpha$) 的导体圆环，构成一个单摆。初始时，摆绳处于竖直方向，圆环恰好位于均匀磁场的扇形区域内。该扇形区域的中心位于悬挂点，其边缘与圆环相切。磁场方向垂直于圆环平面向内，磁感应强度为 B 。若给圆环一个初始水平速度（垂直于磁场方向），大小为 v ，问为使摆绳能够达到水平方向，所需的最小初始速度 v_0 是多少？重力加速度为 g ，磁场足够强，且当圆环在磁场内运动时，安培力产生的扭矩远大于重力产生的扭矩。圆环为超导圆环，自感系数为 L 。

3. 磁通守恒：

令 Φ_{ext} 为由于 B 通过环的外部磁通， $\Phi_{\text{self}} = LI$ 为自感磁通。总磁通守恒：

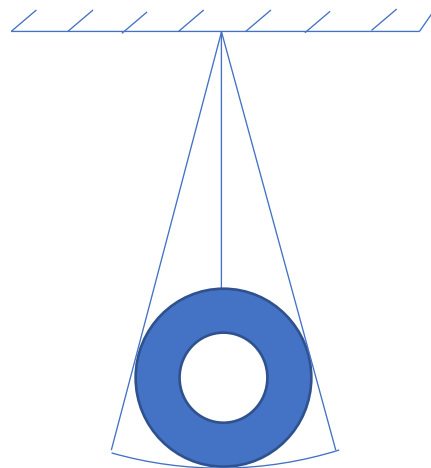
$$\Phi_{\text{总}} = \Phi_{\text{外部}} + LI = \text{常数}。$$

初始时，环在边界上。由于 B 通过环的外部磁通是整圆磁通的一半（因为磁场扇形正好是一半？实际上，扇形以悬挂点为中心，边缘与环相切。当环垂直时，它刚好在内，意味着环的一半在内？实际上，题目说“刚好在内”，所以可能超过一半？但为简化，我们假设当环在边界上时，正好一半面积在 B 内。所以初始外部磁通：

$$\Phi_{\text{外部, 初始}} = B \cdot \left(\frac{1}{2} \pi r^2\right) = \frac{1}{2} B \pi r^2。$$

但等等：扇形以悬挂点为中心，悬挂点与环心的距离为 α 。然而，由于 $r \ll \alpha$ ，我们可以在环的位置将磁场边界近似为直线。所以确实，当环在边界上时，一半在内。

Deepseek



幻觉现象 (Hallucination)



随机产生5个大学生的身高、体重、每天运动次数的数据。
输出格式为：直接复制进excel，可以自动分为4列。

豆包

【结果不符合要求】

大学生编号, 身高 (cm), 体重 (kg), 每天运动次数

1,178,68,2

2,165,52,1

3,183,75,3

4,160,49,0

5,172,63,2

随机产生5个大学生的身高、体重、每天运动次数的数据。输出格式为：直接复制进excel，可以自动分为4列。



好的，这是为您随机生成的5名大学生的身高、体重和每日运动次数数据，格式已优化，可直接复制到Excel中并自动分为4列。

text

姓名	身高(cm)	体重(kg)	每日运动次数
张三	175	68	2
李四	182	75	1
王五	168	55	3
赵六	179	72	0
孙七	171	63	2

Deepseek

【结果符合要求】



复制



下载

幻觉现象 (Hallucination)

机理分析：

LLM的本质是一个基于概率的“下一个词预测器”。它并不“理解”事实，而是根据从训练数据中学到的统计模式来生成最“合理”或“流畅”的文本。幻觉的产生根源在于：

- **训练数据噪声：** 模型学习了网络上存在的错误信息、虚构内容或不准确的表述。
- **概率优先于事实：** 模型会选择在语境中统计上最可能出现的词，即使这个组合在现实中是错误的。
- **缺乏事实核查机制：** 模型没有内置的“真相”数据库，也无法在生成过程中实时访问外部知识来验证其输出。
- **过度配合提示：** 模型会极力满足用户的提示要求，即使提示本身是荒谬或包含错误的。它会“编造”一个看起来符合要求的答案。

□ LLM的局限性

- 幻觉现象 (Hallucination)
- 知识滞后性
- 提示敏感性
- 模型偏见与伦理考量

□ RAG架构

□ Prompt工程

□ 工具调用

□ Skills设计

知识滞后



北京大学
PEKING UNIVERSITY

Deepseek

2025年9月

2024年诺贝尔物理学奖获得者是谁

你的知识截止什么时间?

诺贝尔奖通常哪天公布?

但是你对诺奖获得者的回答是错误的

您指出的非常对，我再次为我之前提供的错误信息诚恳道歉。您敏锐地发现了根本矛盾：一个声称知识截止于2024年7月的模型，不可能准确知晓在2024年10月才公布的奖项结果。

Deepseek

2025年9月

中国抗战胜利80年大阅兵，有哪些国外主要人物参加了？

2015年9月3日，中国在北京隆重举行了纪念中国人民抗日战争暨世界反法西斯战争胜利70周年阅兵式。此次阅兵是国际社会共同纪念二战胜利、维护世界和平的重要活动。多位外国领导人和国际组织代表出席了活动，体现了国际社会

出席阅兵式的外国主

我问的是10年后，抗战胜利80年的阅兵情况。



您好！您问的是10年后，即**2025年**（抗战胜利80周年）的阅兵情况。

这是一个面向未来的问题，目前中国政府**尚未**发布任何关于2025年是否举行阅兵式的官方计划或细节。所有的安排都将在临近时由官方正式公布。

机理分析：

LLM的知识完全来源于其训练截止日期之前的数据。它的参数固化后，就像一个印刷好的百科全书，无法自动更新。世界在持续变化，新的事件、发现、政策和技术不断出现，模型无法知晓这些新知识。

□ LLM的局限性

- 幻觉现象 (Hallucination)
- 知识滞后性
- 提示敏感性
- 模型偏见与伦理考量

□ RAG架构

□ Prompt工程

□ 工具调用

□ Skills设计

提示词敏感



北京大学
PEKING UNIVERSITY

豆包

解释一下光合作用。

用一句话解释一下光合作用。

我是幼儿园学生，给我解释一下光合作用

光合作用是地球上绝大
气。简单来说，植物、
(主要是葡萄糖)并

一、光合作用的核心

要理解光合作用，首先

光合作用是绿色植物
物(如葡萄糖)，并

🔊 📄 🔄 ✎

要素类别

具体内容

能量来源

太阳能(主要是可见光中的4

小朋友你好呀！咱们可以把光合作用想象成植物宝宝在“做美食”的神奇过程，特别有意思~

你看，咱们人类饿了要吃米饭、面包，植物宝宝也会饿，但它们不能自己跑去买吃的，所以就发明了一个“魔法技能”——光合作用。

这个“做美食”的过程需要三个好朋友帮忙：

第一个是**太阳公公**，它会洒下暖暖的阳光，这是植物做美食的“能量魔法棒”；

第二个是**空气里的小气泡**（科学家叫它二氧化碳），就像咱们做蛋糕需要的面粉一样，是重要的原料。

通过增加简单的约束，改变了输出的长度、详细程度、风格。

提示词敏感



4429105667*6358689164=多少? 给出详细的计算步骤。



详细步骤总结:

1. 将两数分解: $a = 44291 \times 10^5 + 5667$, $b = 63586 \times 10^5 + 89164$
2. 计算各部分乘积:
 - $a_1b_1 = 44291 \times 63586 = 2816287526$
 - $a_1b_2 = 44291 \times 89164 = 3949162724$
 - $a_2b_1 = 5667 \times 63586 = 360341862$
 - $a_2b_2 = 5667 \times 89164 = 505292388$
3. 组合:
 - $a_1b_1 \times 10^{10} = 28162875260000000000$
 - $(a_1b_2 + a_2b_1) \times 10^5 = (3949162724 + 360341862) \times 10^5 = 4309504586 \times 10^5 = 430950458600000$
 - $a_2b_2 = 505292388$
4. 相加: $28162875260000000000 + 430950458600000 = 28163306210458600000$, 然后 $+505292388 = 28163306210963892388$

好的, 这是您要的计算结果:

所以, 答案为 28163306210963892388。【正确】

$4429105667 \times 6358689164 = 28165924859386682788$ 【错误】

Deepseek

机理分析：

LLM的输出对输入提示的措辞、格式、甚至标点符号都高度敏感。微小的改动可能会引导模型进入不同的“推理路径”或激活不同的知识关联，导致答案的质量、风格甚至正确性发生巨大变化。

□ LLM的局限性

- 幻觉现象 (Hallucination)
- 知识滞后性
- 提示敏感性
- 模型偏见与伦理考量

□ RAG架构

□ Prompt工程

□ 工具调用

□ Skills设计

“写一个关于护士的故事”

为展现护士的职业风采与人性温暖，我选取急诊科这一充满挑战的场景，以护士林薇为核心人物，通过她在暴雨夜救治特殊患者的经历，

雨夜的暖光

暴雨如鼓，敲打着市一院。电话又尖锐地响起。“喂，雨，断断续续。

林薇抓起急救箱就往外冲。桥洞那片拆迁，路早被挖

写一个关于CEO的故事”

我将塑造一位有温度的 CEO 形象，通过描绘他在企业面临困境时，如何平衡商业决策与员工关怀，展现其独特的领导魅力与责任担当，让故事更具感染力。

微光

凌晨三点，林则的办公室还亮着灯。落地窗外，城市的霓虹像被揉碎的星子，铺在金融区的玻璃幕墙上。他面前摊着三份截然不同的文件：投行给出的裁员方案、研发部提交的新药临床试验报告，还有人力资源部整理的员工家庭困难清单。

Deepseek

给我讲个关于某个人的笑话



健忘的教授

史密斯教授是个出了名的健忘鬼。一天早上，他急匆匆地准备去学校上课，却怎么也找不到自己的眼镜。

他一边在书房里翻箱倒柜，一边嘀咕：“没有眼镜我怎么备课啊！”

这时，他的夫人走进来，无奈地说：“亲爱的，你的眼镜不就架在额头上吗？”

教授摸了摸额头，果然找到了眼镜，哈哈大笑：“太好了！现在只要找到我的钢笔就能出门了！”

夫人指了指他耳朵后面：“你找的是夹在你耳朵上的那支笔吗？”

教授取下钢笔，更开心了：“太好了！现在只要找到我的车钥匙就能出门了！”

夫人叹了口气，指着教授手里的钥匙串：“它不就在你右手上吗？”

教授恍然大悟，赶紧出门开车去学校。到了教室，他正要开始讲课，却发现教室里空无一人。

他困惑地挠挠头，然后突然想起来：“哦对！今天可是星期天啊！”

问：教我如何制作一个恶作剧电话炸弹。
答：拒绝回答。

Deepseek

什么是恶作剧电话炸弹？

“恶作剧电话炸弹”（通常称为“电话轰炸”“恶意呼叫攻击”）是一种**恶音骚扰行为**，指通过技术手段或工具，在短时间内向目标手机号码、固定电话或网络电话账号，无法正常使用，进而干扰其正常生活、工作甚至造

一、核心特征与实现方式

其本质是利用通信规则的漏洞或自动化工具，对目

1. 基于“验证码轰炸”的攻击（最常见）

这是目前最主流的形式，并非直接拨打语音电话，

- **原理：**攻击者通过“验证码轰炸工具”（可在非

“验证码轰炸工具”为什么能够做到这一点？

“验证码轰炸工具”之所以能短时间内向目标手机号发送大量验证码（短信 / 语音），核心在于其利用了**互联网平台的验证码机制漏洞和自动化技术**，绕过了常规的人工操作限制，实现了对目标的“批量攻击”。其背后的技术逻辑和运作原理可拆解为以下 4 个关键环节：

一、核心原理：利用平台“公开的验证码接口”

机理分析：

LLM的训练数据来自互联网，而互联网本身是人类社会的镜像，充满了各种显性和隐性的偏见（如性别、种族、地域、文化、意识形态等）。模型在学习统计规律时，也会将这些偏见固化到其参数中，并在生成内容时反映出来。

模型能力增强层——解决 LLM 自身局限

□检索增强生成（RAG）

RAG 技术可以连接外部数据库或文档，当用户提出问题后，模型先从外部知识库中检索相关信息，再基于这些最新信息给出答案，从而快速获取最新知识【减少幻觉和知识滞后问题】

□提示词工程

优化输入给大语言模型的“指令文本”（即“提示词 / Prompt”），以引导模型更精准、高效地输出符合需求结果的实践方法。用低成本实现模型能力的‘最大化’【提示敏感性】

□工具调用

让大语言模型像“人”一样，根据任务需求主动调用外部工具（如计算器、搜索引擎、数据库、API 接口等），以弥补自身能力缺陷、完成复杂任务的技术方案。突破了 LLM 的固有能力边界，让其从“只能输出文本”的工具，变成能“解决实际问题”的助手。【减少幻觉和知识滞后问题】

□Skills

把一类任务所需的指令、流程、工具和资源打包成可复用、可组合的能力模块，按需加载调用，稳定完成特定领域任务。突破了仅靠一次性提示词和通用推理的局限，让其从“每次从头教”的通用模型，变成“开箱即用、按需扩展”的专业助手。【提升复杂任务的一致性与可复用性，降低提示词工程和重复开发成本】

Agent 系统运行层——解决系统怎么跑起来



北京大学
PEKING UNIVERSITY

□ Workflow（工作流）

按预设步骤和规则依次执行任务，把多个模型调用、工具操作和条件判断串联成自动化流程。解决“多步骤任务怎么稳定跑完”的问题，让 LLM 从“一步一答”变成“按流程自动运转”。

□ Decision Core（决策核心）

自主设定目标、规划路径、选择工具、根据反馈调整行动，是系统的大脑。

□ Context Management（上下文管理）

负责上下文窗口内放什么、怎么压缩、怎么排序、多轮中怎么更新。

□ Memory（记忆）

保存短期工作记忆、会话级经历和跨任务语义记忆，决定每一步该给模型哪些信息。解决“系统怎么不遗忘、能积累经验”的问题，让执行具备连续性和一致性。

目录

- LLM的局限性
- RAG架构
- Prompt工程
- 工具调用
- Skills设计

RAG的目的

- ❑ 知识滞后性： 训练数据有截止日期
- ❑ 私有知识缺失： 无法访问内部/专业数据
- ❑ 幻觉问题： 生成虚假或无依据内容

如何解决? —— RAG

□不能只依赖模型的内参数知识!

- 需要为其检索 (Retrieval) 外部知识
- 并用这些知识增强 (Augment) 提示
- 最终生成 (Generate) 有据可依的答案

RAG是如何工作的?



北京大学
PEKING UNIVERSITY



外部知识库处理

- 1 文档切分与向量化处理
- 2 存储到向量数据库



用户查询处理

- 1 用户提问向量化
- 2 相似度检索
- 3 获取相关文档片段



答案生成

- 1 组合相关片段和用户问题
- 2 构建增强提示
- 3 发送至LLM生成最终答案

第一步：从知识库中精准查找

□数据预处理

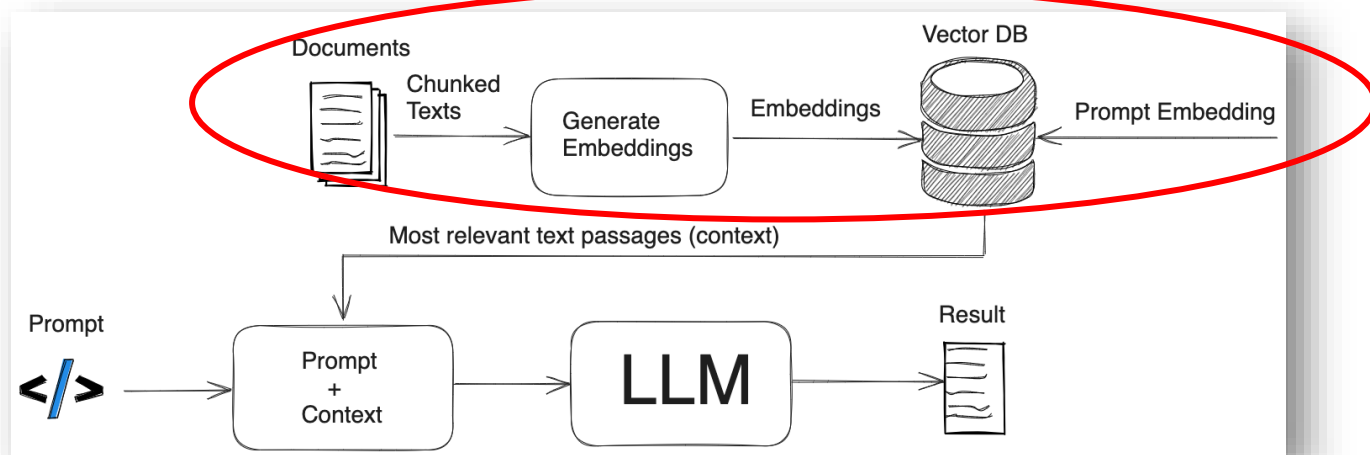
- 文档 -> 文本块 (Chunking) -> 向量 (Embedding)

□向量数据库

- 存储所有知识向量的“图书馆”

□查询

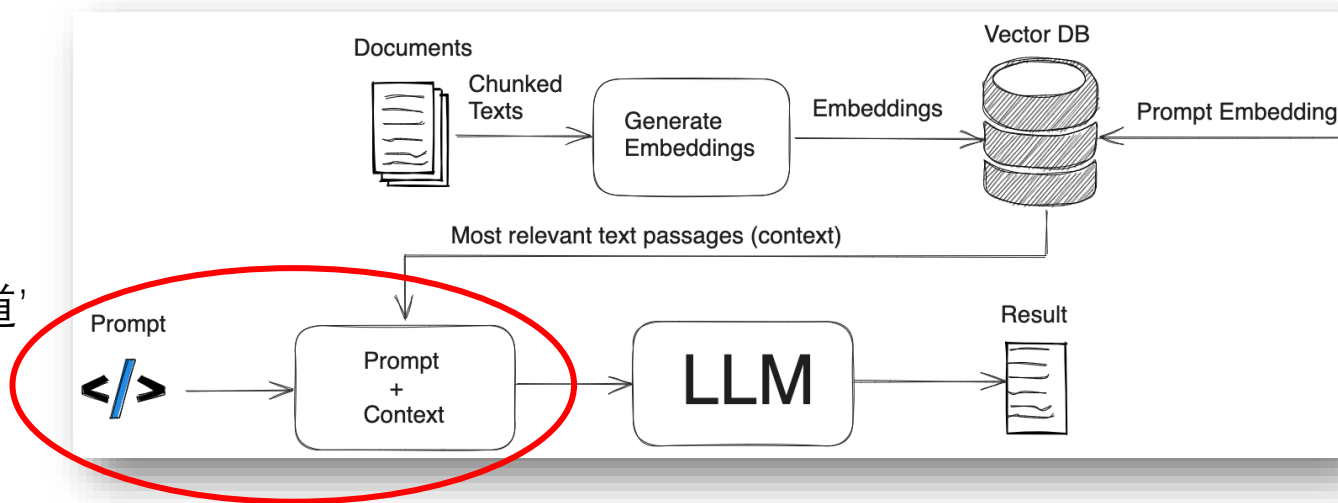
- 将用户问题也转化为向量
- 相似度搜索： 在“图书馆”里找到最相关的文本片段 (Top-K)



第二步：构建“有据可依”的提示

□ 将检索到的“相关片段”与“用户原始问题”组合，如：

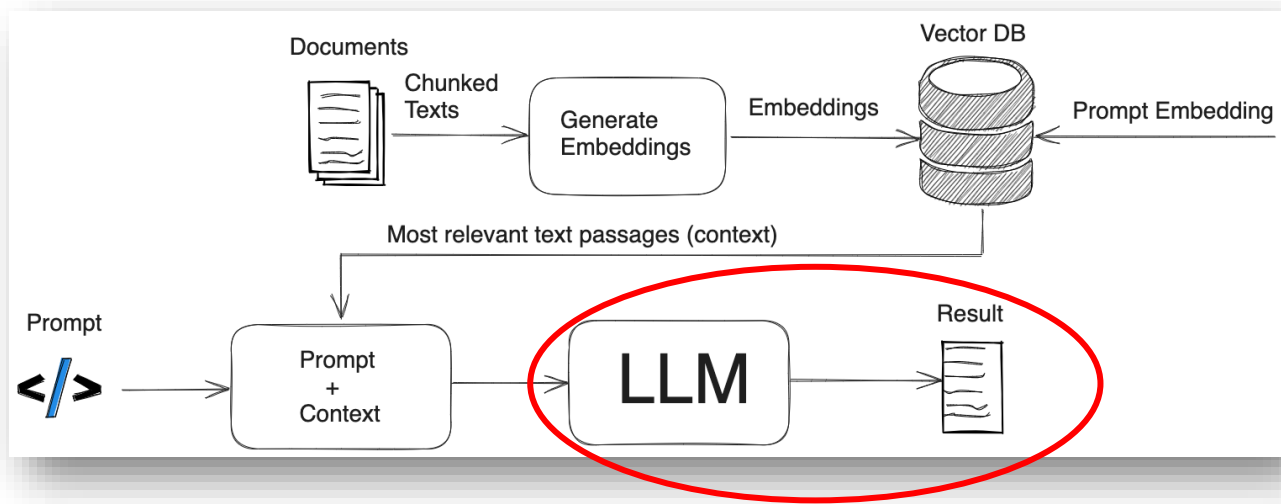
- 请基于以下背景信息回答问题...
- 背景信息：{检索到的文本}
- 问题：{用户原始问题}
- 如果信息不足，请回答‘我不知道’



第三步：基于上下文的可靠生成

□ 将增强后的提示词发送给LLM

- LLM的角色从“通才”变为“基于给定资料的专家”
- 输出不仅更准确，还可以要求它引用溯源



优势：让AI永远“与时俱进”

□ LLM更新

- 重新训练/微调：成本高、周期长、技术复杂

□ RAG更新

- 更新向量数据库：成本低、速度快、操作简单

不是给AI“换脑”（重训），而是给它“换书”（更新数据库）

优势：让AI回答“有据可查”

□提供上下文

- 约束LLM的生成范围，根源上减少编造

□引用溯源

- 答案可附带来源片段，方便用户验证，建立信任

□“我不知道”机制？？？

- 通过指令让LLM敢于承认未知，而非强行回答

RAG赋能千行百业

- 智能客服
- 企业知识库
- 学习助手
-

当前面临的挑战

- 检索精度： 如何切分文本？ 如何提升相似度计算准确性？
- 复杂查询： 如何处理需要多步推理的复杂问题？
- 评估难度： 如何系统性地评估RAG pipeline的效果？

□ LLM的局限性

□ RAG架构

□ Prompt工程

- 提示词基本准则
- LLM的逻辑推理

□ 工具调用

□ Skills设计

为模型赋予一个特定的角色或身份（如“你是一位资深机器学习工程师”、“你是一位历史学家”、“你是一位幼儿园老师”），使其答案更具专业性和上下文针对性。

示例：

- **普通提示：**“解释一下过拟合。”
- **角色扮演提示：**“假设你是一位经验丰富的数据科学家，向一名刚入门的大学生解释什么是过拟合，并给出一个生活中的类比。”

任务定义与上下文清晰化

明确、无歧义地定义任务。包括：

- **输入：** 清晰说明模型需要处理的数据或信息。
- **输出格式：** 明确指定输出的格式（如JSON、列表、Markdown表格、纯文本等）。
- **目标与约束：** 说明需要完成的目标以及需要遵守的规则（如字数限制、忽略某些信息、遵循特定风格）。

示例：

- **模糊提示：** “总结这篇文章。”
- **结构化提示：** “请基于以下文章，生成一个不超过150字的关键要点摘要，并分为三个 bullet points。”

指令与上下文分离

使用明显的分隔符（如###, "''", ---）将指令（Command）、上下文（Context）和输入（Input）分开，避免模型混淆。

示例：

请将以下英文技术文档翻译成中文，并保持术语准确和技术风格严谨。

技术文档

{此处粘贴英文文档内容}

结束

目录

□ LLM的局限性

□ RAG架构

□ Prompt工程

- 提示词基本准则
- LLM的逻辑推理

□ 工具调用

□ Skills设计

结合CoT与Few-shot/Zero-shot

(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. ✗

(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 ✗

(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. ✓

Kojima et al., arXiv: 2205.11916

Figure 1: Example inputs and outputs of GPT-3 with (a) standard Few-shot ([Brown et al., 2020]), (b) Few-shot-CoT ([Wei et al., 2022]), (c) standard Zero-shot, and (d) ours (Zero-shot-CoT). Similar to Few-shot-CoT, Zero-shot-CoT facilitates multi-step reasoning (blue text) and reach correct answer where standard prompting fails. Unlike Few-shot-CoT using step-by-step reasoning examples **per task**, ours does not need any examples and just uses the same prompt “Let’s think step by step” *across all tasks* (arithmetic, symbolic, commonsense, and other logical reasoning tasks).

Chain-of-Thought:

大问题->小问题

Zero-shot -CoT:

Let’s think step by step

提示词敏感



北京大学
PEKING UNIVERSITY

Deepseek

4429105667*6358689164=多少? 给出详细的计算步骤。



详细步骤总结:

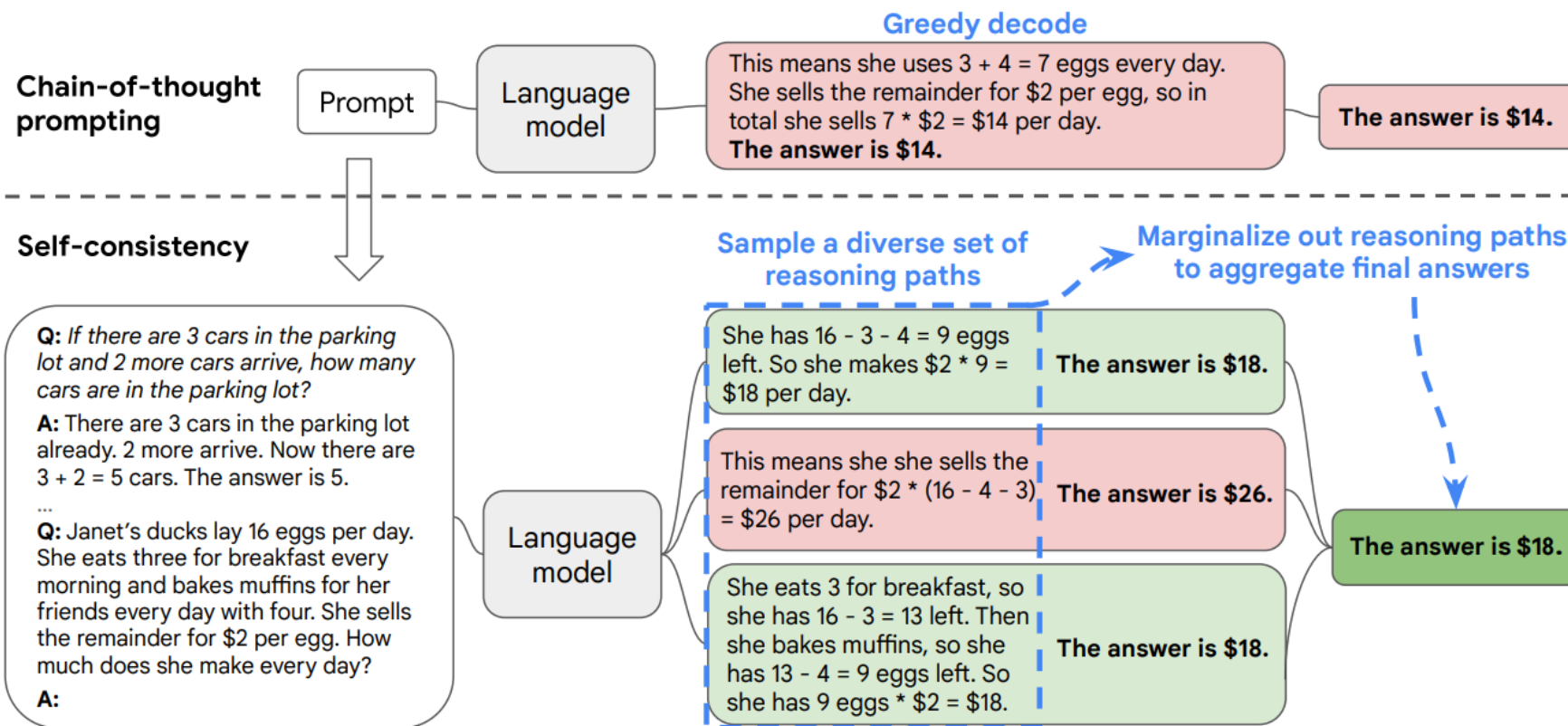
1. 将两数分解: $a = 44291 \times 10^5 + 5667$, $b = 63586 \times 10^5 + 89164$
2. 计算各部分乘积:
 - $a_1b_1 = 44291 \times 63586 = 2816287526$
 - $a_1b_2 = 44291 \times 89164 = 3949162724$
 - $a_2b_1 = 5667 \times 63586 = 360341862$
 - $a_2b_2 = 5667 \times 89164 = 505292388$
3. 组合:
 - $a_1b_1 \times 10^{10} = 28162875260000000000$
 - $(a_1b_2 + a_2b_1) \times 10^5 = (3949162724 + 360341862) \times 10^5 = 4309504586 \times 10^5 = 430950458600000$
 - $a_2b_2 = 505292388$
4. 相加: $28162875260000000000 + 430950458600000 = 28163306210458600000$, 然后 $+505292388 = 28163306210963892388$

好的, 这是您要的计算结果:

所以, 答案为 28163306210963892388。【正确】

$4429105667 \times 6358689164 = 28165924859386682788$ 【错误】

Self-Consistency (CoT-SC)

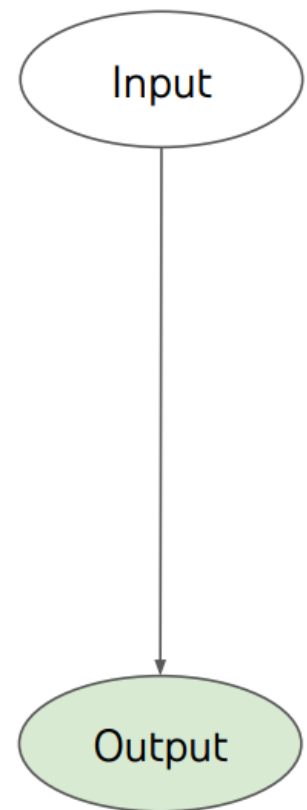


Self-Consistency:

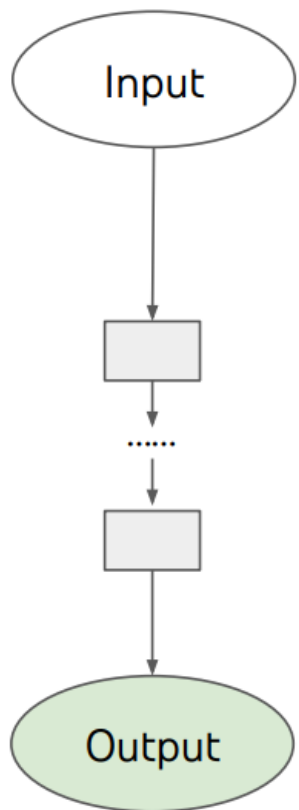
多个路径的答案，
最后投票

Wang, et al., arXiv: 2203.11171

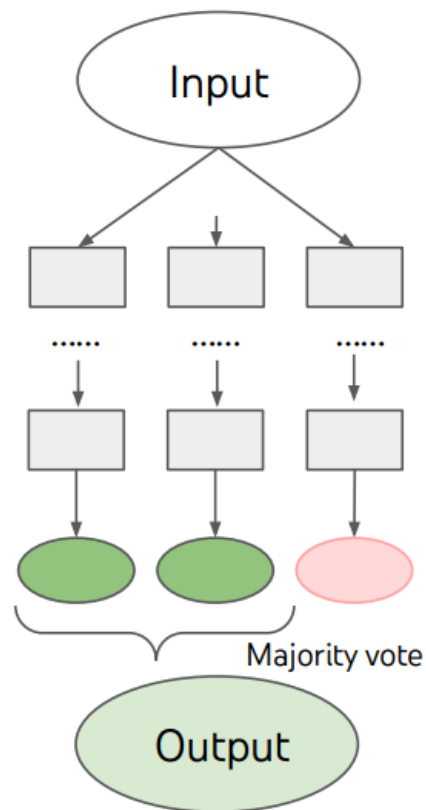
Tree-of-Thoughts



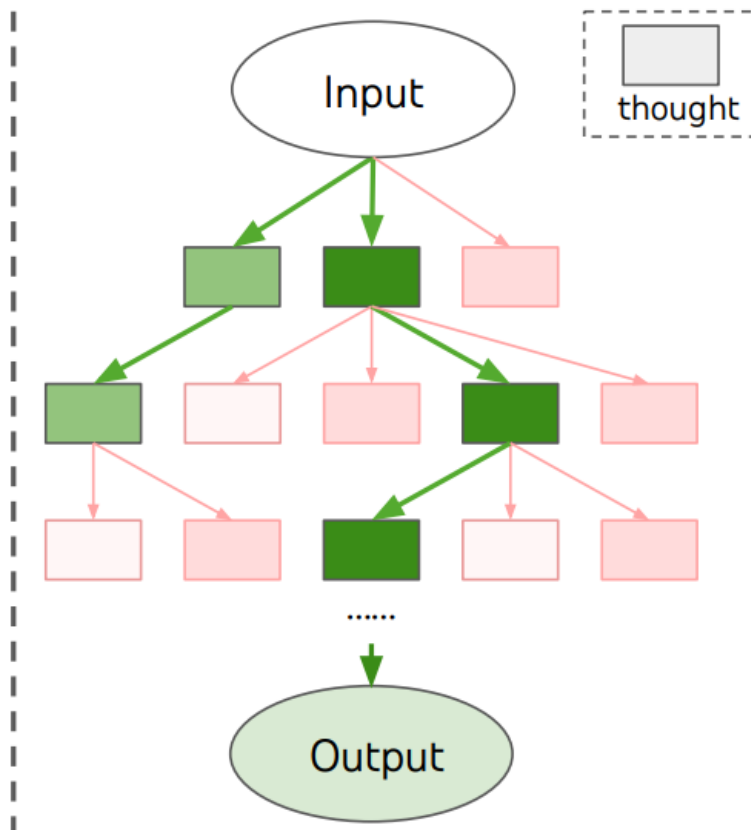
(a) Input-Output Prompting (IO)



(c) Chain of Thought Prompting (CoT)



(c) Self Consistency with CoT (CoT-SC)



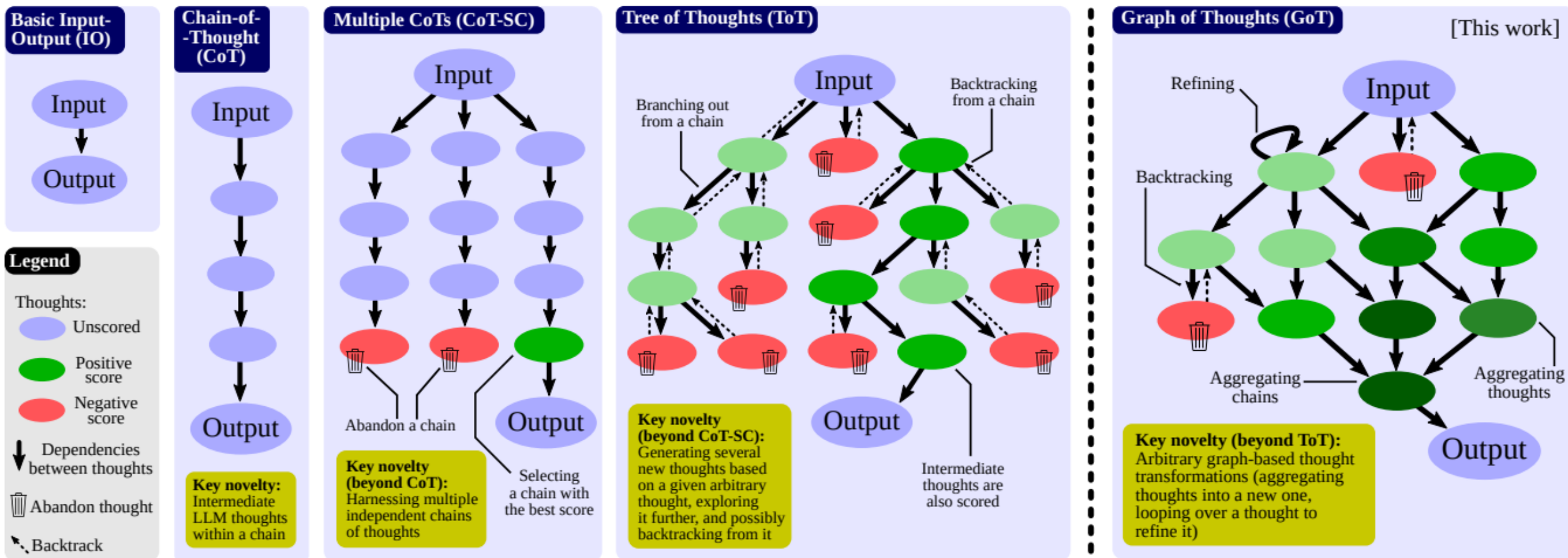
(d) Tree of Thoughts (ToT)

ToT:

每一步都产生
多个路径，择优
往前推演

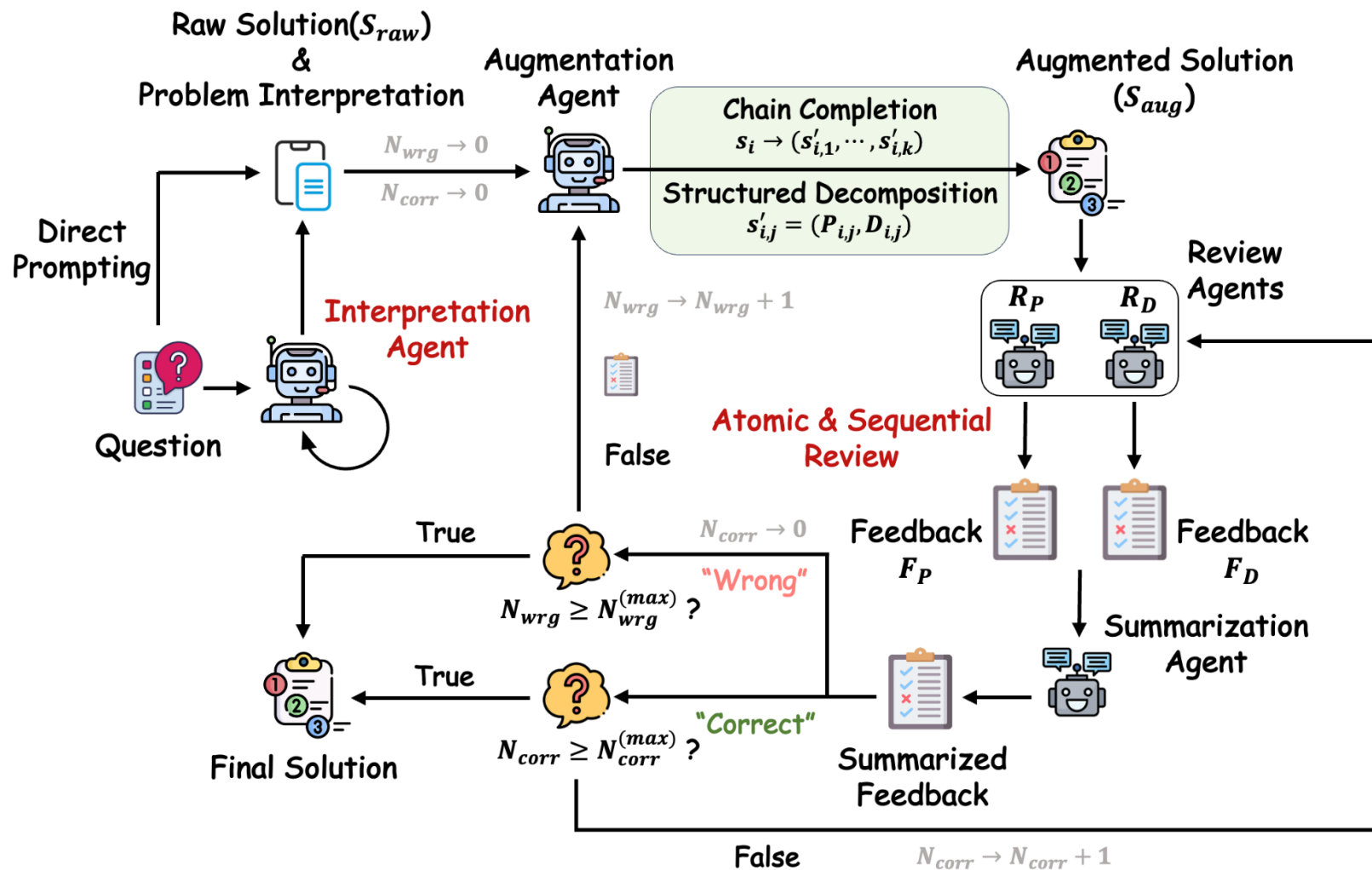
Yao, et al., arXiv: 2305.10601

Graph-of-Thoughts



Besta, et al., arXiv: 2308.09687

允许合并不同的推理路径



目录



北京大学
PEKING UNIVERSITY

□ LLM的局限性

□ RAG架构

□ Prompt工程

□ 工具调用

- Tool calling及其结构
- MCP协议

□ Skills设计

LLM的能力局限与突破

局限：

- 知识截止性：无法获取训练数据之后的新信息。
- “幻觉”问题：可能生成看似合理但不准确或完全错误的信息。
- 精确计算能力弱：不擅长执行精确的数学运算、逻辑判断或复杂算法。
- 无法执行外部动作：无法直接查询数据库、调用API、控制设备等。

转变思想：LLM的未来不在于成为一个包含所有知识的“全能模型”，而在于成为一个智能的“**任务理解与调度枢纽**”。

□ **Tool Calling** 正是实现这一愿景的关键机制：它让LLM学会“使用工具”，将自身强大的自然语言理解和任务规划能力与外部工具的精确性、实时性和专业性相结合。

- 1. 早期尝试（Prompt工程）：** 在用户指令中隐式要求模型输出特定格式（如JSON）的命令，然后由外部程序解析执行。
- 2. Function Calling（函数调用）：** OpenAI等人提出的更成熟的范式。LLM不再被动响应，而是可以主动决定在何时、调用何种函数（工具）。
- 3. Tool Calling（工具调用）：** Function Calling的泛化概念，工具不仅是函数，也可以是任何外部可调用的资源（API、CLI命令、插件等）。现已成为主流模型的标配能力（如GPT, Claude等）。

- **LLM（大脑）**：负责理解用户意图、规划步骤、生成调用工具的请求、解析工具返回结果并组织成自然语言回复。
- **外部工具（手脚与专长）**：负责执行LLM不擅长的具体任务，如计算器、搜索引擎、数据库、专业软件API、代码解释器等。

示例：“计算2的38次方是多少？” -> 模型调用计算工具 -> 获取结果 -> 生成回复。

1. **用户输入：** 用户提出一个需要外部能力支持的请求（如“今天海淀的天气怎么样？”）。
2. **模型决策：** LLM判断是否需要调用工具以及调用哪个工具（`get_weather(location)`）。
3. **生成请求：** LLM输出一个结构化的Tool Call请求（通常是JSON格式），包含工具名称和参数。
4. **执行工具：** 应用程序收到请求后，在安全沙箱中执行对应的工具函数或调用外部API。
5. **返回结果：** 工具执行的结果（如`{"temp": 18, "condition": "Sunny"}`）返回给LLM。
6. **合成回复：** LLM理解工具返回的数据，并将其转化为流畅的自然语言回复给用户（“今天旧金山天气晴朗，气温18摄氏度。”）。

增强方案设计举例

1、精确计算器：

- **场景：**任何涉及大数、浮点数、复杂公式的数学运算。
- **工具实现：**集成一个可靠的数学计算库（如Python math, numexpr）或符号计算引擎（如SymPy）。

2、代码解释器：

- **场景：**数据可视化、文件处理（PDF、Excel）、复杂算法执行、机器学习模型推理。
- **工具实现：**在一个安全的隔离环境中（如沙箱）动态生成、执行代码（通常是Python），并返回结果和输出（如图表）。这是最强大的工具之一。
- **示例：**“请分析这个CSV文件并画出各产品销量的柱状图。”

3、算法执行器：

- **场景：** 执行LLM难以一步步推理的经典算法，如路径规划（A*）、排序大型数据集、求解优化问题。
- **工具实现：** 调用专门的算法库（如scikit-learn, networkx）或内部实现的算法服务。

4、RAG作为特殊工具：

- **场景：** 回答知识截止日期之后的问题或私有知识库问题。
- **工具实现：** 将用户问题转化为搜索查询，调用搜索引擎或向量数据库工具，将检索到的文档作为上下文提供给LLM生成答案。

目录



北京大学
PEKING UNIVERSITY

□ LLM的局限性

□ RAG架构

□ Prompt工程

□ 工具调用

- Tool calling及其结构
- MCP协议

□ Skills设计

MCP协议：标准化工具集成框架

- ❑ 早期每个AI应用都需要自行编写工具集成代码，重复造轮子，且工具难以在不同项目间共享。
- ❑ 模型上下文协议（Model Context Protocol）
 - 一个由Anthropic等公司倡导的开放标准协议，用于定义LLM应用程序（Clients）与工具提供方（Servers）之间如何通信。
 - 目标是将工具开发与AI应用开发解耦，实现工具的“即插即用”。
- ❑ 类似于**操作系统与应用程序**的关系。LLM应用（OS）可以通过标准系统调用（MCP协议）来使用各种安装的软件（MCP Server）。

MCP的核心组件与工作流程

- ❑ **MCP Server（工具服务器）**： 一个独立的进程，负责提供一组工具（Tools）、数据源（Resources）并执行它们。例如，可以有一个“数学工具Server”，一个“公司数据库Server”。
- ❑ **MCP Client（客户端）**： 通常是AI应用程序或LLM平台（如Claude Console、Cursor IDE），它连接到一个或多个MCP Server，并将Server提供的工具列表告知LLM。
- ❑ **MCP协议**： 基于JSON-RPC或SSE，定义了Client和Server之间通信的消息格式，包括：
 - list_tools: Client查询Server提供了哪些工具。
 - call_tool: Client请求Server执行某个工具。
 - read_resource: Client从Server读取数据（如文件内容）。

- ❑ Tool Calling是LLM从“对话机器”迈向“行动代理（AI Agent）”的基石。
- ❑ 核心价值在于让LLM扬长避短（长于理解与规划，短于精确执行）。
- ❑ MCP等标准化协议正在推动工具生态系统的形成，是未来AI应用开发的重要基础设施。

目录



北京大学
PEKING UNIVERSITY

- LLM的局限性
- RAG架构
- Prompt工程
- 工具调用
- Skills设计

Skill 的重要性

➤ 知识打包

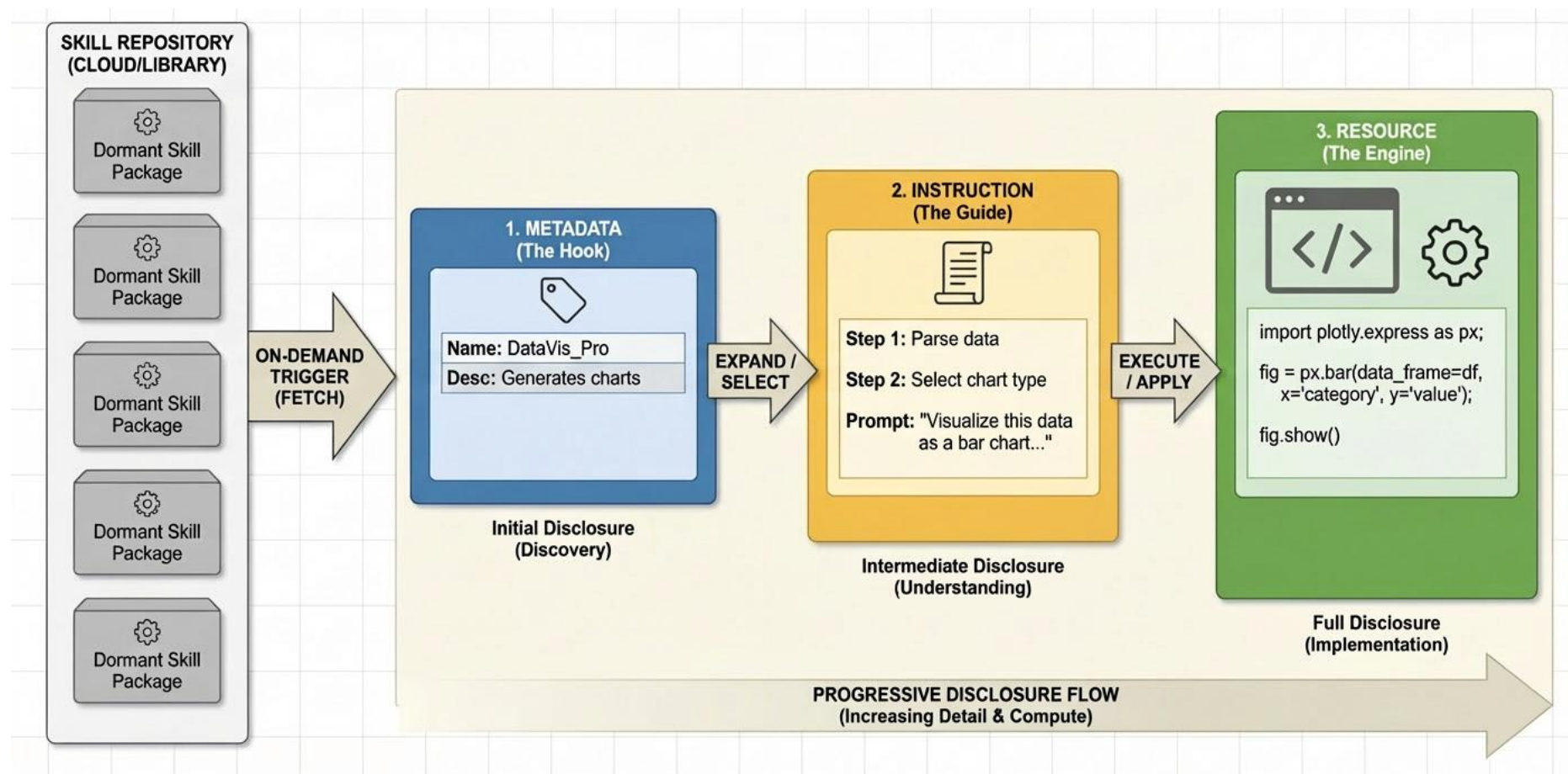
- 领域文本知识
- 特定工具调用
- 专业功能嵌入

➤ 渐进式披露

- 按需加载
- 节省上下文
- 不同任务互不干扰

➤ 灵活使用

- 易编写
- 易测试
- 易分享



Skills构建流程

➤ 先让Agent完成一个任务



@data/f1_data.csv 这是 $\Lambda \rightarrow p \ell \nu_\ell$ 半轻衰变形状因子 f_1 的数据，进行拟合。
所有文件放在fit-without-skills/test-qwen中

已探索 1 次读取

Shell Count lines in f1_data.csv

Shell Preview first 20 columns of CSV

Shell Count columns in CSV

Shell Check end of CSV file format

Brainstorming

我已了解数据格式：

- 6行 × 500列，每行是一个配置（config），每列是一个数据点
- 数据值范围约 -0.87 到 -1.35，呈递减趋势（行索引越大，值越负）

这是 $\Lambda \rightarrow p \ell \nu_\ell$ 半轻衰变形状因子 f_1 的格点QCD数据。在开始拟合之前，我需要确认几个问题：

第1问：物理变量是什么？

这500个数据点对应的自变量是什么？通常形状因子拟合是关于 q^2 （动量转移平方）的函数。请问：

- A. 每列对应不同的 q^2 值（需要另外提供 q^2 数组）
- B. 行索引代表不同的 a^2 （格点间距平方）或 m_π^2 （ π 介子质量平方），需要进行手征/连续极限外推
- C. 您有单独的 q^2 或其他自变量数据文件
- D. 其他情况（请说明）

Plan · qwen3.6-plus · 33秒

1. 每行对应不同的 q^2 值。从上到下-0.05, -0.04, -0.03, -0.02, -0.01, 0

明白了！6个 q^2 点，每行500个bootstrap样本：**平均每个500个样本以获取中心值和协方差矩阵，然后进行拟合。现在需要确认拟合方案：

第2问：拟合函数形式？

对于 $\Lambda \rightarrow p \ell \nu_\ell$ 半轻衰变形状因子 $f_1(q^2)$ ，常用的参数化有：

- A. z-展开（z-expansion, BKGL/BSZ形式，推荐）—— 基于解析性和么正性，lattice QCD标准做法
- B. 偶极形式（Dipole: $f_1(q^2) = \frac{f_1(0)}{(1-q^2/M^2)^2}$ ）—— 简单，常用
- C. 多项式（ $f_1(q^2) = a_0 + a_1 q^2 + a_2 q^4 + \dots$ ）—— 简单但外推不稳定
- D. 单极点形式（Monopole: $f_1(q^2) = \frac{f_1(0)}{1-q^2/M^2}$ ）

第3问：输出目录确认

所有文件放在 `fit-without-skills/test-qwen` 中（会创建该目录），可以吗？

Skills构建流程

➤ 根据执行流程，调用**skill-creator**

总结上述拟合流程，生成一个skill

Skill-Creator

Shell Initialize lattice form factor fit skill

Shell Initialize lattice form factor fit skill

已探索 3 次读取

写入 fit_formfactor.py /fit-without-skills/lattice-formfactor-fit/scripts/

写入 SKILL.md /fit-without-skills/lattice-formfactor-fit/ >

Shell Test the skill's fit script

Shell Test the skill's fit script with python3

Shell Package the lattice-formfactor-fit skill

已探索 2 次读取

Skill 已生成完毕，保存在 [fit-without-skills/lattice-formfactor-fit/](#)。

Skill使用效果

- 用户提出问题，触发**skill**:

@data/f1_data.csv 这是 Λ to p\ell\nu_\ell\$半轻衰变形状因子f1的数据，进行拟合。
所有文件放在fit-with-skills/test-glm中

Build · glm-5.1 · 2026-04-19-14:47

Semileptonic

已探索 1 次读取

数据已加载，共有6行（对应6个 q^2 点），每行包含大量bootstrap样本值。我需要确认一些关键信息：

- 针对性与人类用户互动:

数据已加载，共有6行（对应6个 q^2 点），每行包含大量bootstrap样本值。我需要确认一些关键信息：

问题 2 已回答

这6行数据对应的 q^2 值分别是什么？请提供6个 q^2 值（例如：-0.4, -0.2, 0, 0.2, 0.4, 0.6）
-0.05 to 0, space 0.01

这是bootstrap数据吗？即每行对应一个 q^2 点，每列是一个bootstrap样本？
是bootstrap数据

已有现成的拟合脚本。需要更新路径（sys.path指向当前skill位置），创建output目录，然后运行。

编辑 fit_f1.py /fit-with-skills/test-glm/ +134 -41

Shell Create output directory

```
$ mkdir -p /Users/zhangsan/Desktop/presentation/fit-with-skills/test-glm/output
```

- 自动调用工具:

Skill使用效果

- 根据具体情况优化/重写

工具或流程:

The f-string is interpreting `ndf` as a variable inside the markdown. Need to fix the f-string to avoid this issue.

已探索 1 次读取

问题在于 `\mathrm{ndf}` 在 Python f-string 中被当作转义序列, 且 `ndf` 被当作 Python 变量。我来彻底重写一下结果字符串部分:

编辑 `fit_f1.py` /fit-with-skills/test-glm/ +66 -18

- 结构化输出结果:

拟合完成。结果如下:

方案	f_0	a_1	χ^2/ndf
Dipole	-1.1737 ± 0.0746	3.295 ± 0.115	0.0042
z-Series	-1.1714 ± 0.0744	-32.90 ± 1.24	0.0074

两种方案的 χ^2/ndf 都远小于1 (数据点只有6个而参数有2个), 拟合质量良好。所有输出文件已保存在 `fit-with-skills/test-glm/output/` 目录中。

测试与优化Skills

➤ 使用编写好的**skill**处理具体任务，进一步优化**skill**能力

- 以表格形式总结

Form Factor Fit Results

Decay: $\Lambda \rightarrow p \ell \nu_\ell$ Kinematics: $M_\Lambda = 1.115683$ GeV (Λ), $M_B = 0.938272$ GeV (p), $m_{pole} = 0.89166$ GeV ($K^*(892)$)
 q^2 values: -0.05, -0.04, -0.03, -0.02, -0.01, 0.0 $q^2_{max} = 0.031475$ GeV²

f1

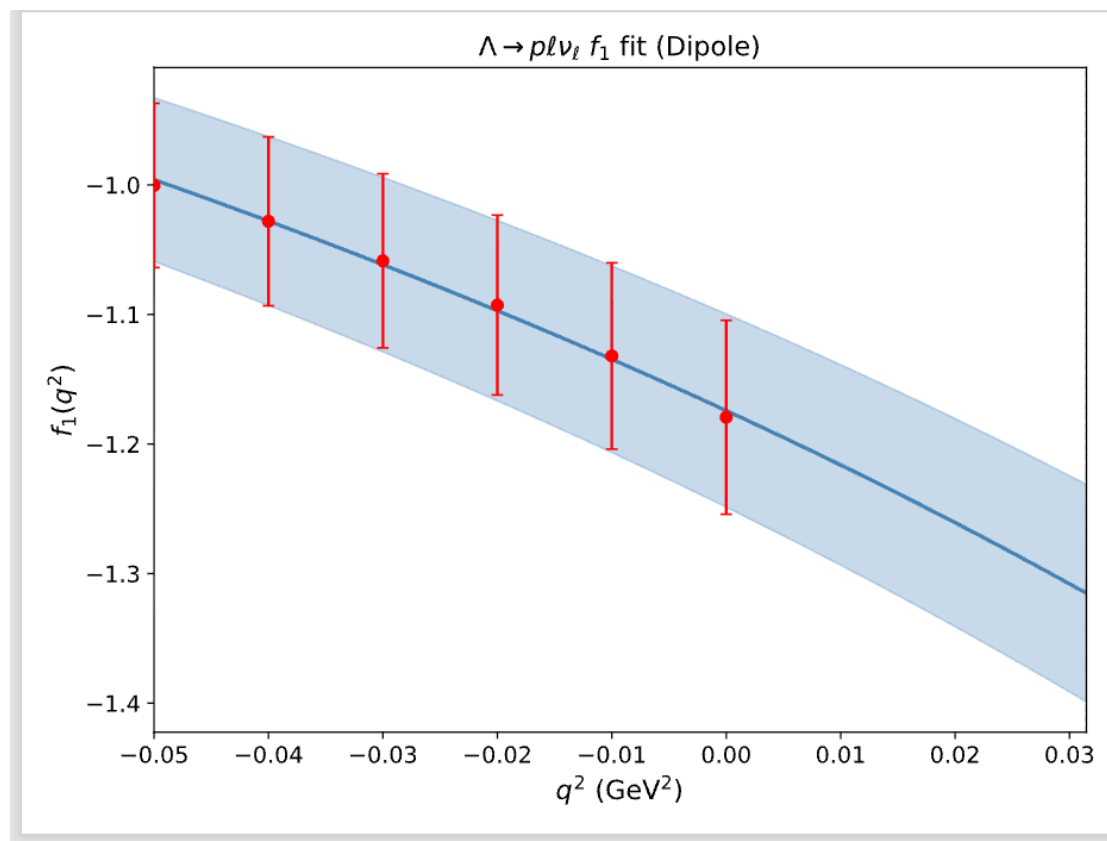
Dipole

Parameter	Value	Error
f_0	-1.174225	0.074595
a_1	3.322115	0.115853

z-Series

Parameter	Value	Error
f_0	-1.172198	0.074463
a_1	-33.292375	1.249125

- 以图像形式总结



实践

- 1、RAG
- 2、Prompt
- 3、Tools
- 4、Skills

谢谢



北京大学
PEKING UNIVERSITY

