

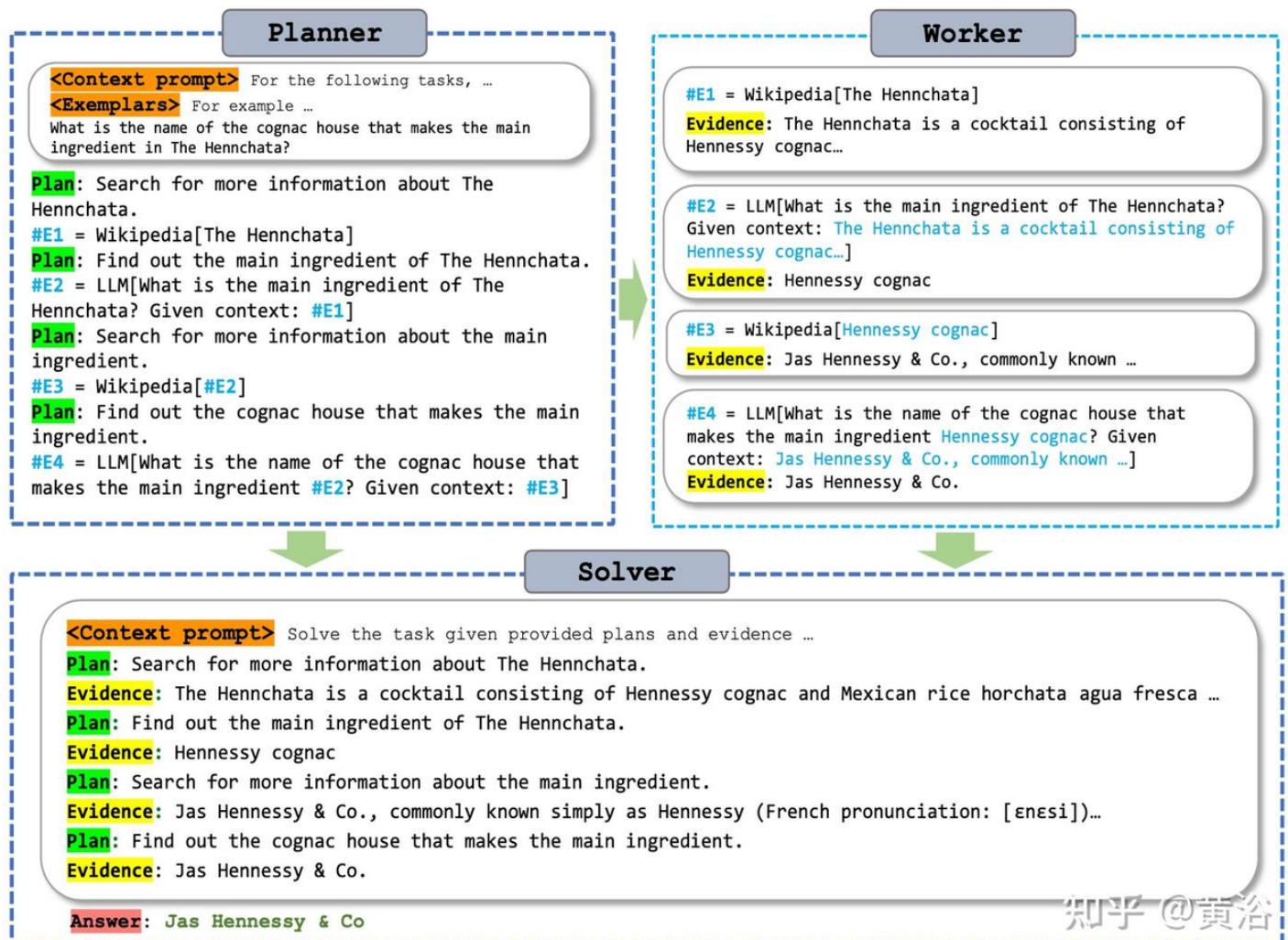
# Agent Planning2 规划

## 规划

### ReWoo

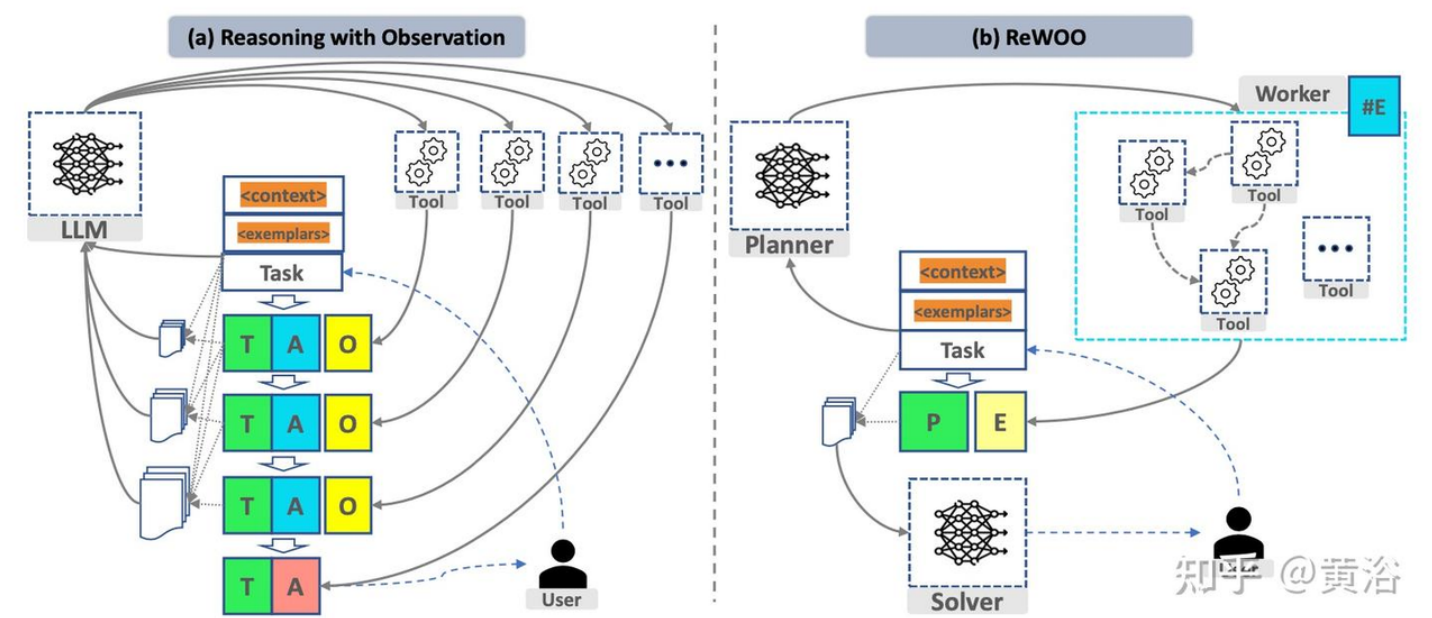
ReAct 提示词结构是 Thought→ Action→ Observation, React每一轮思考都要将之前所有的响应加入 prompt, 会消耗大量的token。ReWoo将推理过程与外部观察分离, 将 Observation 隐式地嵌入到下一步的执行单元中了, 即由下一步骤的执行器自动去 observe 上一步执行器的输出, 从而显著减少了 token消耗。

ReWoo包含三个部分: Planner负责将用户问题分解为子任务并确定执行顺序, 每个子任务都分配给 Worker; Worker利用工具检索外部知识提供证据; Solver负责综合所有任务和证据, 生成最终答案。



如下图所示, React每一轮的都要讲上下文、示例和之前轮次的相应输入到LLM中, 带来大量的冗余, 并且可能需要调用LLM很多次; ReWoo中Planner负责生成一个子任务列表, 并调用Worker从工具中获取证据, 根据列表循环执行完成任务, 避免了将prompt中一样的内容反复交给LLM, 这个过程只调

用了两次LLM（Planner和Solver各一次）。ReWoo的另一个优点是简化微调过程，由于Planner不依赖于工具的输出，因此可以在不实际调用工具的情况下对Planner进行微调。



Plan and solve

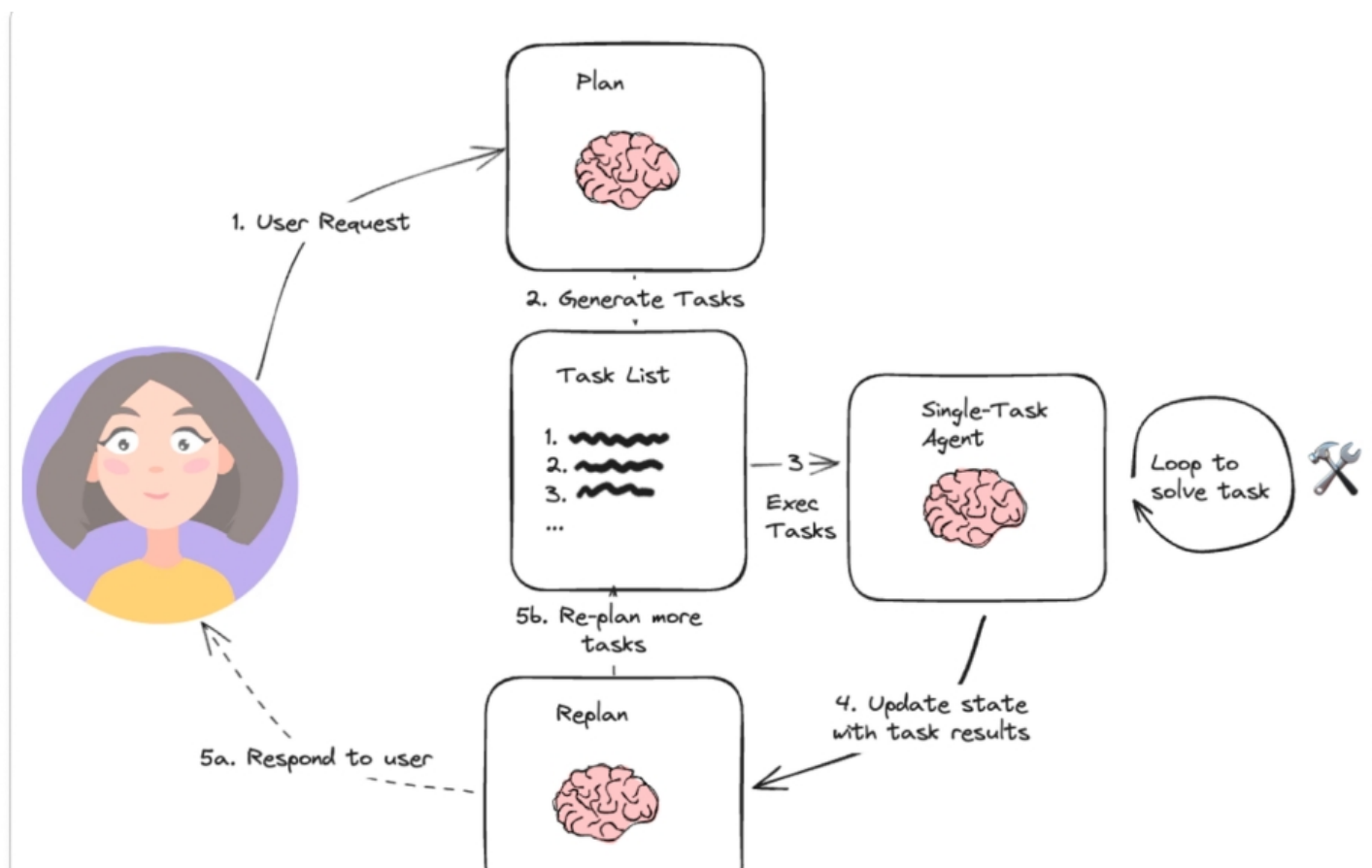
这个方法在零样本思维链的基础上进行优化。Zero-shot-CoT知识简单的在prompt中加入“让我们逐步思考”，面临着计算错误、缺失步骤错误和语义误解错误等三个问题。Plan and solve解决了缺失步骤错误，先指定计划将任务分解为子任务，再按照计划执行子任务。

这个方法整体感觉偏向提示工程。prompt应该满足以下条件：

- 引导LLMs确定子任务并完成这些子任务,
- 指导LLMs更加关注计算和中间结果，并尽可能确保它们的正确执行。

最终的prompt格式为：“Q: [X]. A: Let’ s first understand the problem, extract relevant variables and their corresponding numerals, and devise a plan.Then let's carryout the plan, calculate intermediate results(pay attention to calculation and common sense), solve the problem step by step, and show the answer.”

Plan-and-Solve相比ReWOO，最大的不同就是加入了Replan机制，整体的思考流程如下图。Planner负责生成任务列表，replanner负责当完成一个子任务时进行重新思考，并将原有计划和已经完成的步骤加入prompt中，更新任务列表。

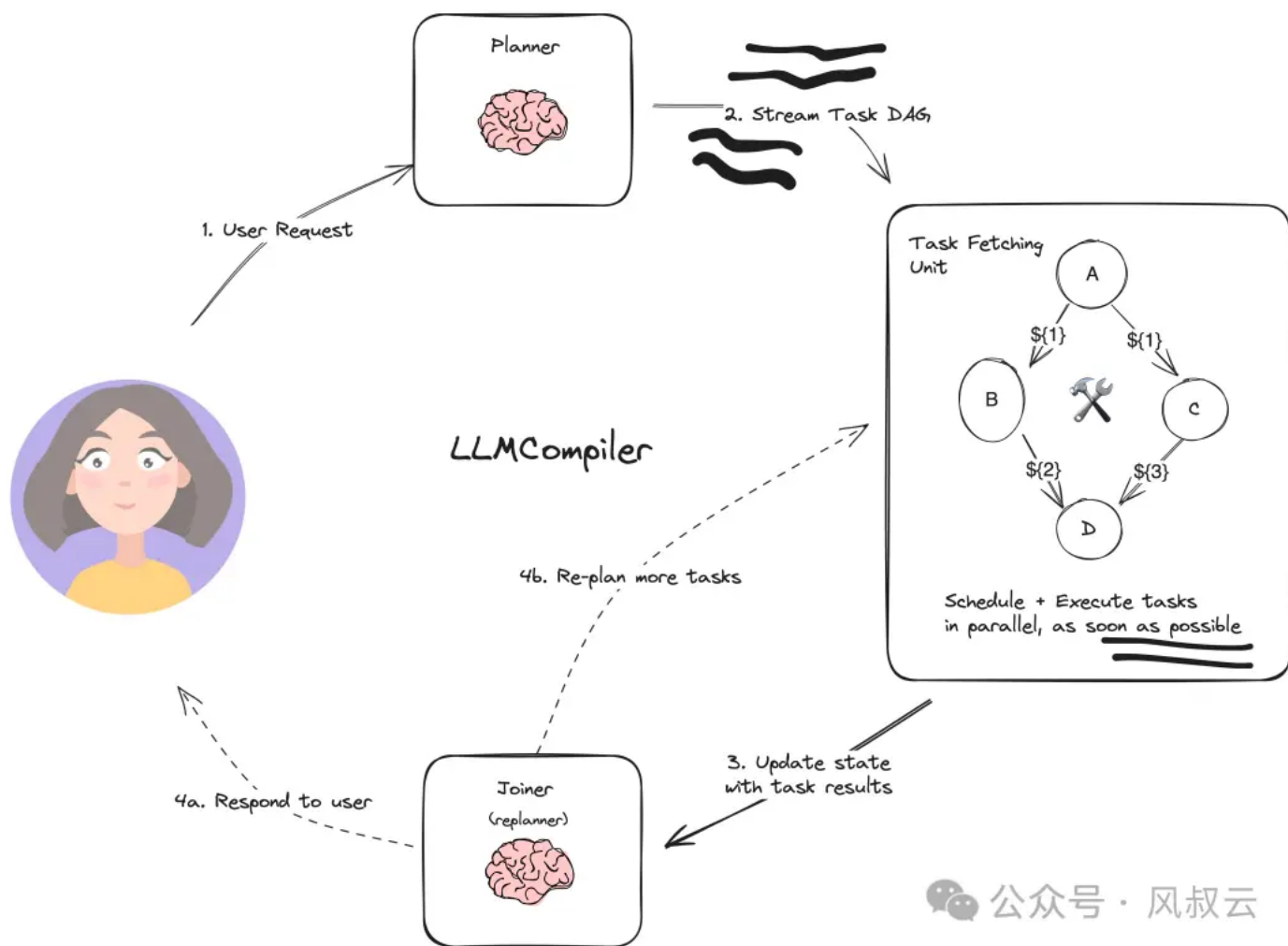
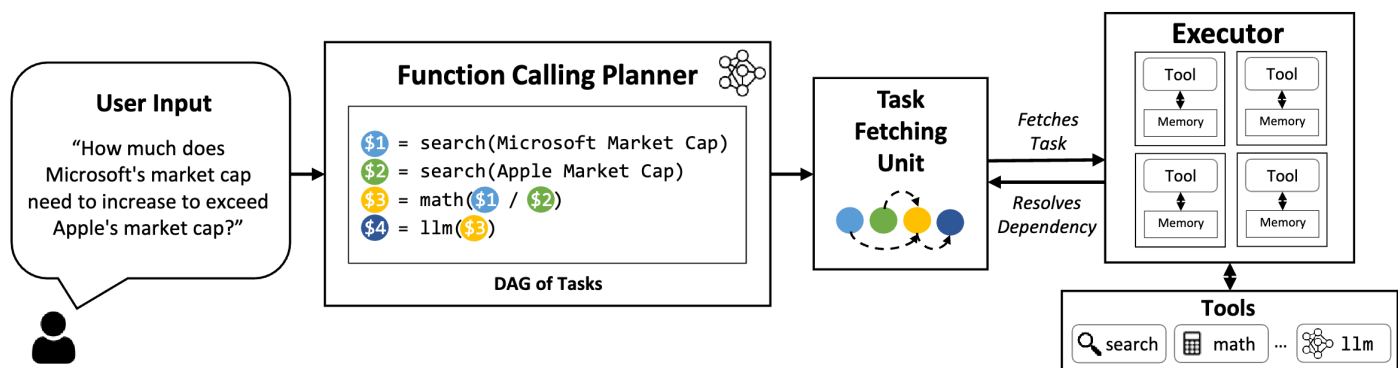


## LLMCompiler

这个方法的主要思想是通过并行function call来提高效率，比如询问微软的市值需要增长多少才能超过苹果的市值，可以并行的查询微软市值和苹果市值。

函数调用规划器负责生成一个包含任务及其相互依赖关系的 DAG(有向无环图)。然后，任务获取单元根据任务的依赖关系将这些任务并行调度到执行器。在本例中，任务 \$1 和 \$2 被同时获取，以并行执行两个独立的搜索任务。每个任务执行完成后，结果将被转发给任务获取单元，用实际值替换其占位符变量，同时解除被依赖任务的阻塞（例如，任务 \$3 中依赖 \$1 和 \$2）。所有任务执行完成后，最终答案将被传递给用户。

- 函数调用规划器：负责理解用户输入，拆分成可执行的子任务，并确定它们之间的依赖关系，形成任务依赖的有向无环图（DAG）。该部分需要用到大模型，最好用户为规划器提供一些上下文示例。
- 任务获取单元：根据贪婪策略，将可以执行的任务发给执行器，并用执行后的输出替换后续任务的占位符。无需LLM
- 执行器：多个执行器并发执行，可以调用用户提供的工具。
- 动态重规划：对于复杂的任务，可能需要根据中间结果进行重新规划，由函数调用规划器生成新的子任务和它们之间的依赖关系。



公众号 · 风叔云

引用:

1. ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models
2. AI大模型实战篇: AI Agent设计模式 – Plan and Execute
3. AI大模型实战篇: AI Agent设计模式 – LLM Compiler