

deeppresearch 研究方向

# 目录

Contents



01. deep research技术框架现状及其突破点
02. deep research核心架构
03. 未来改进方向
04. 应用场景

在语言上，我们定义“深度研究代理”为：由大型语言模型驱动的人工智能代理，集成动态推理、自适应规划和迭代工具使用，用于获取、汇总和分析外部信息，最终产出完成开放式信息研究任务的综合成果。

# deep research技术框架 及其突破点

01

# 01. 技术框架现状及其突破点

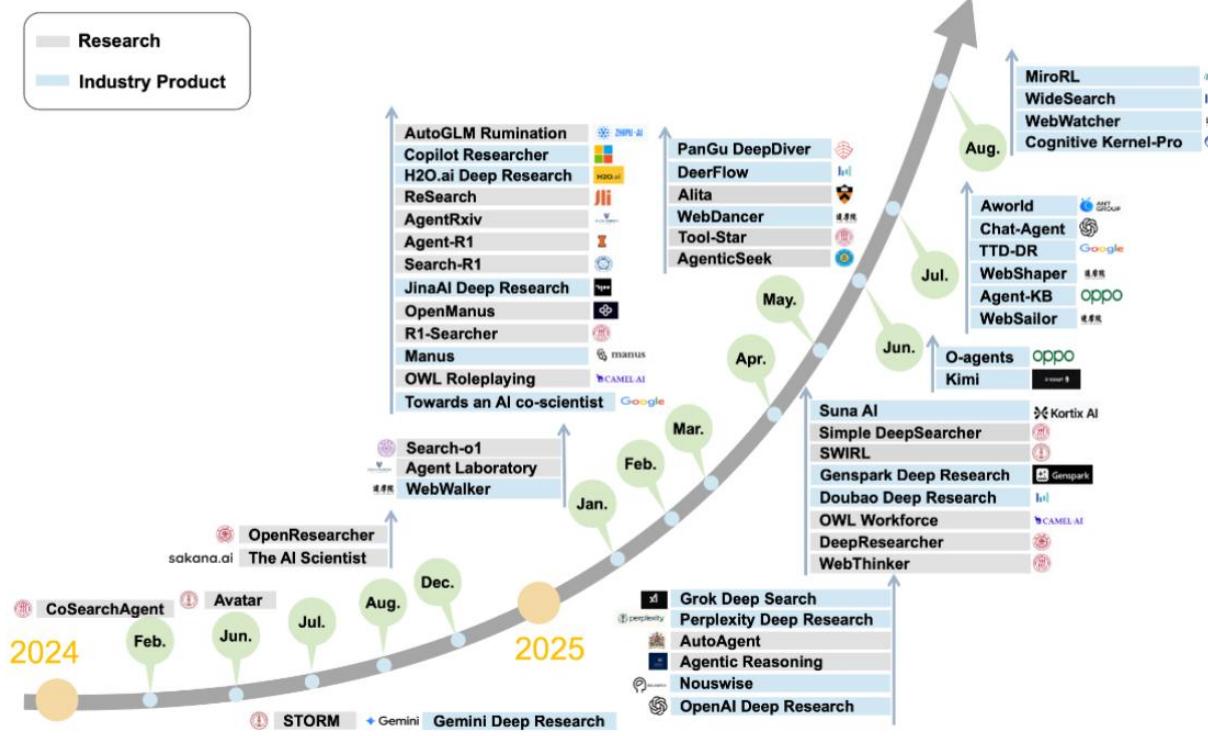
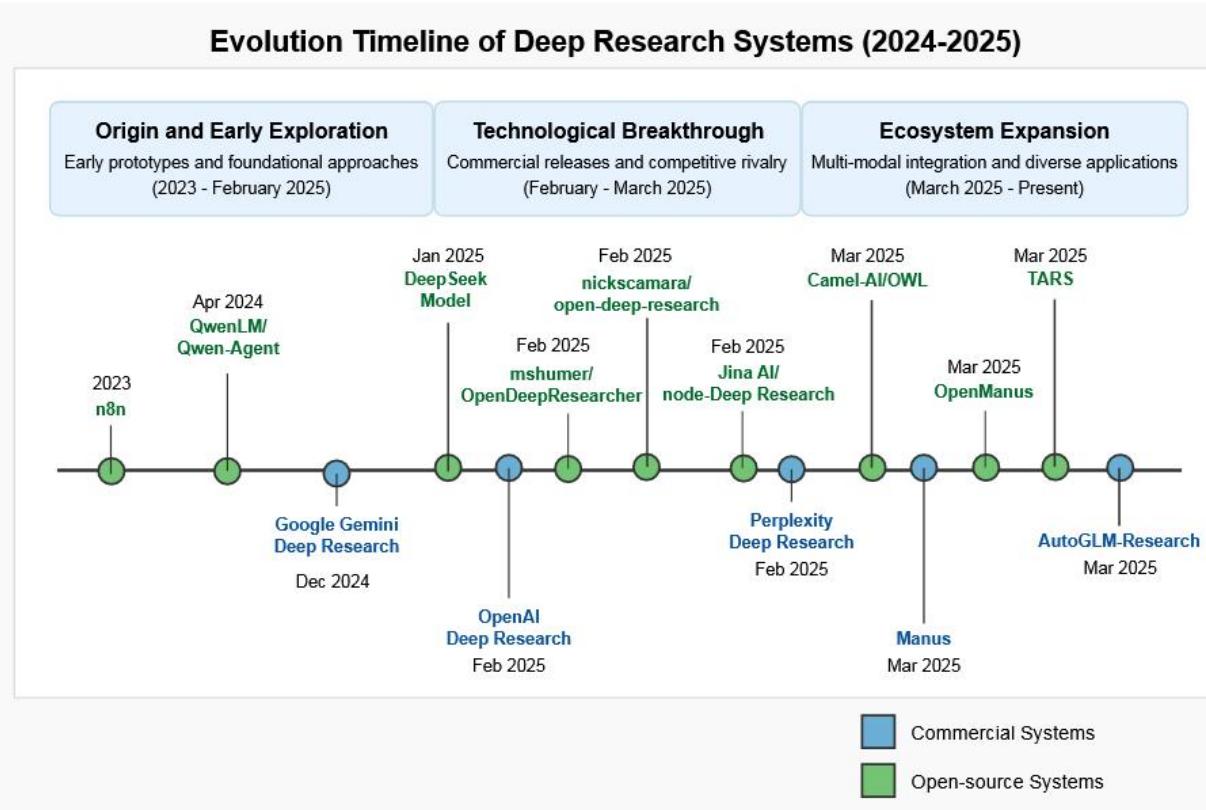


Figure 5: An overview of DR agents' evolution over years.

Fig. 1. Evolution Timeline of Deep Research Systems

## 起源与早期探索（2023 – 2025年2月）

这个阶段主要是基础原型和底层技术的探索，包括2023年的n8n、2024年4月的QwenLM/Qwen-Agent，以及2024年底谷歌推出的Gemini Deep Research。2025年1月DeepSeek Model的发布，标志着技术从实验室走向更成熟的原型。

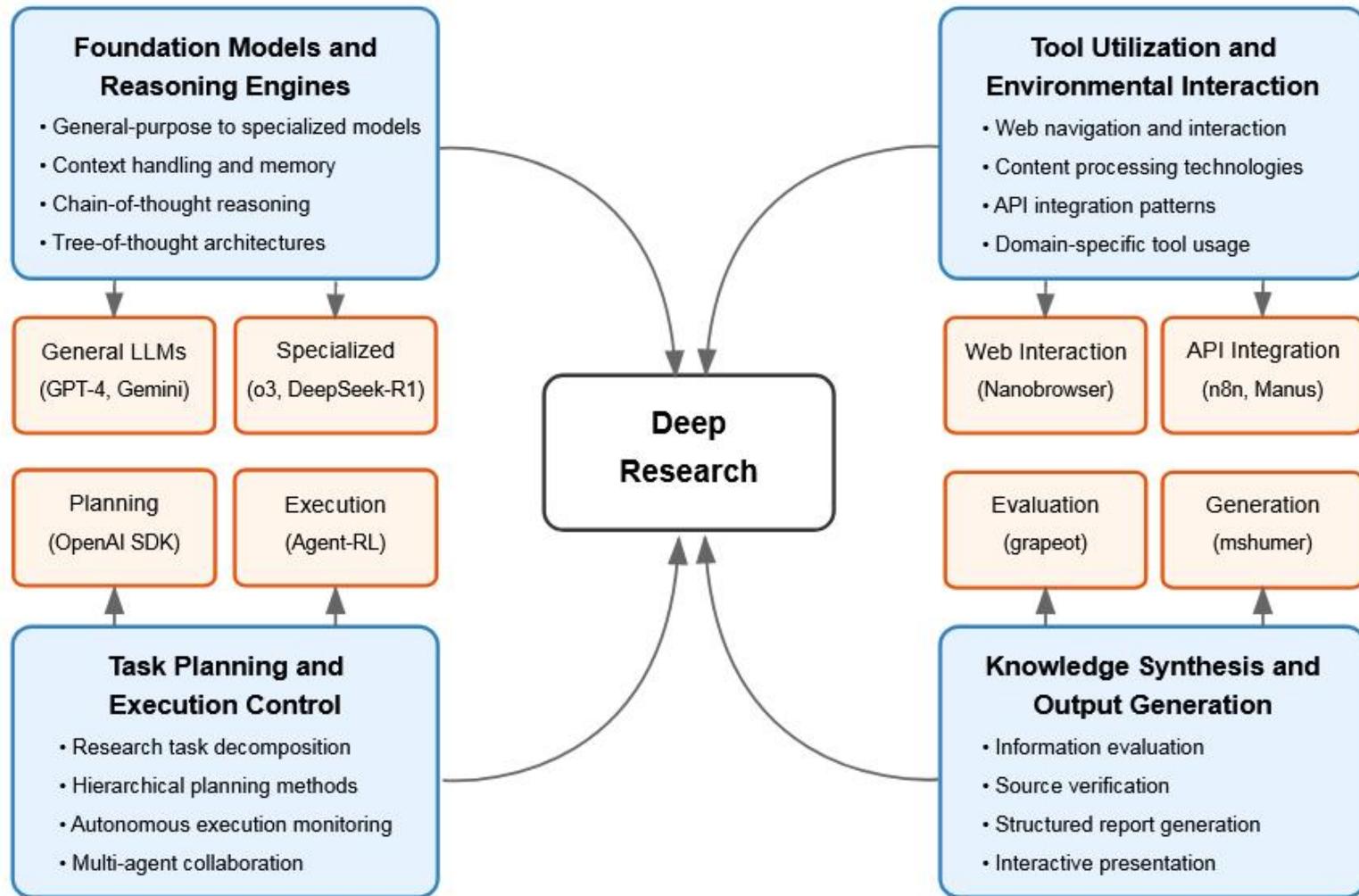
## 技术突破期（2025年2月 – 3月）

这是技术迭代加速，竞争最激烈的阶段，商业公司和开源项目集中爆发。2024年还处于零星探索阶段，到2025年随着Grok Deep Search、PanGu DeepDiver等产品的出现，进入爆发期，新工具的发布频率明显加快。

**场景多元化（2025年3月 – 至今）：**从早期的文献搜索，扩展到多模态分析、智能协作等更复杂的研究场景，例如MiroRL、WideSearch等工具开始支持认知推理、跨模态检索等能力。

# 01. deepricerca技术架构

Hierarchical Technical Framework of Deep Research Systems



# 01. Deep Research 的技术框架发展现状

主要类别	核心定位	架构特点与关键突破点	代表系统
基础模型与推理引擎 (DR 的“大脑”)	是 DR 系统的核心驱动力，经历“通用 LLM→专用研究模型”的进化	<p>模型 specialization: 从 GPT-4 等通用模型，进化为 OpenAI o3、Gemini 2.5 Pro 等研究优化模型，通过架构修改（如增强推理层）、专用训练语料（如学术文献、科研方法论）提升分析能力。</p>	OpenAI/DeepResearch (o3 模型，支持多步推理)、Google Gemini/DeepResearch (Gemini 2.5 Pro, 100 万 token 上下文窗口)、AutoGLM-Research (基于 ChatGLM 微调，优化科研推理)。
		<p>上下文与记忆机制(长上下文处理优化——开发长上下文优化模型，如 Google Gemini 2.0 支持 100 万 token 窗口，)：</p> <ul style="list-style-type: none"><li>从“有限上下文窗口”（早期 LLM 仅 4k-16k tokens）进化为“分层记忆架构”，区分“工作记忆”（实时推理上下文）和“长期记忆”（知识仓库），通过情景缓冲、注意力检索突破上下文限制。</li><li>提出位置插值 (Positional Interpolation) 与稀疏注意力 (Sparse Attention) 机制（如 Bai et al., 2024），在不显著增加计算成本的前提下，将有效上下文窗口扩展至百万 token 级别，支持整本书籍、长周期实验数据的连贯分析；</li><li>结合 RAG、知识图谱技术可同时处理多模态长文档（如科研论文、实验记录）。</li><li>信息压缩与优先级排序：开发“语义保留压缩”技术；通过分层 summarization 技术，维持多级别信息表征。</li></ul>	代表系统：Perplexity/DeepResearch (基于 DeepSeek-R1, 支持数百源信息连贯分析)、Camel-AI/OWL (动态注意力分配，优先处理高相关信息)。
		<ul style="list-style-type: none"><li>增强推理能力：从“零样本提示推理”进化为“结构化推理框架”，如链思维 (Chain-of-Thought)、树思维 (Tree-of-Thought)、辩论驱动推理 (Debate-driven Reasoning)，模拟人类科研中的逻辑验证过程。</li><li>增强多轮对话推理：增强上下文连贯性与意图对齐。针对多轮研究中“推理断裂”“意图偏移”问题，最新技术通过对话状态建模与结构化推理链提升连贯性</li><li>并行推理合成，突破长序列和复杂推理限制。</li></ul>	代表系统：QwenLM/Qwen-Agent (多阶段推理，含规划 - 采集 - 分析 - 合成闭环)、CycleResearcher (集成自动化同行评审，通过反馈修正推理偏差)。Dialogue CoT：将对话历史与上下文状态嵌入推理链；Structured CoT (SCoT)：通过结构化模板（如“问题拆解→子任务规划→证据记录→结论整合”）规范推理步骤。

# 01. deepresearch技术架构

## 基础模型增强

纯提示词方法受限于基础 LLM 的固有能力，复杂任务易达性能上限，因此需通过监督微调（SFT）、强化学习（RL）等优化技术扩展模型能力。

### 监督微调（SFT）目标

优化 DR 关键组件：搜索查询生成、结构化报告、外部工具使用，提升检索质量、减少幻觉、支持证据 *grounded* 的长文本生成。

### 典型方法

**Open-RAG**：加入检索 / 相关性 /grounding/ utility 等多样监督信号，通过对抗训练提升信息过滤与检索精度。

**AUTO-RAG**：构建基于推理的指令数据集，让模型自主规划查询、多轮与检索器交互，动态 refine 检索策略。

**DeepRAG**：二叉树搜索机制，递归生成子查询，构建多轮检索轨迹，平衡内部知识与外部检索，提升效率、减少冗余查询。

**基于拒绝采样的 SFT**（如 CoRAG）：从 QA 数据中提取中间检索链，支持逐步检索增强与子查询动态重写，减少对人工 SFT 数据的依赖。

**hint-infer 机制**：监测生成时的 token 模式，触发外部计算工具（Python 执行器等），再通过拒绝采样微调，让模型自主生成提示与调用工具。

**ATLAS**：仅在专家轨迹的关键步骤上训练，显著提升泛化。

### 局限

SFT 仍局限于离线、静态检索管道，在线自适应能力不足，因此需要 RL。

■ = Yes, ■ = Yes but details unknown, □ = Not present

DR Agent	SFT	RL	Base Model	Data	Reward Design	Release
Gemini DR [33]	■	■	Gemini-2.0-Flash	□	■	Dec-2024
WebWalker	□	□	GPT-4o, Qwen-2.5 7-72B	WebWalkerQA	□	Jan-2025
Grok DeepSearch [124]	□	■	Grok3	□	■	Feb-2025
OpenAI DR [78]	□	■	GPT-o3	□	■	Feb-2025
Agentic Reasoning [122]	■	□	DeepSeek-R1, Qwen2.5	GPQA	Rule-Outcome	Feb-2025
AutoAgent [101]	□	■	Claude-Sonnet-3.5	□	□	Feb-2025
Towards an AI co-scientist	□	□	Gemini 2.0	□	□	Feb-2025
Agent-R1 [80]	□	PPO [90], Reinforce++ [43], GRPO [92]	Qwen2.5-1.5B-Inst	HotpotQA	Rule-Outcome	Mar-2025
AutoGLM Romination [137]	■	■	GLM-Z1-Air	□	■	Mar-2025
H2O.ai DR [39]	■	■	h2ogpt-oasst1-512-12b	□	■	Mar-2025
Copilot Researcher [69]	■	■	o3-mini	□	□	Mar-2025

Continued on next page

# 01. deepresearch技术架构

## 基础模型增强

### 强化学习 (RL)

#### 核心优势

通过实时奖励信号学习，实现在线查询生成、工具调用的自适应优化，解决合成演示数据与分布偏移问题，在开放环境中更鲁棒。

#### 发展脉络

早期：DeepRetrieval 优化查询生成，提升检索质量。

中期：ReSearch、R1-Searcher 将 RL 扩展到检索后的自适应推理，动态 refine 搜索策略。

近期：Search-R1 结构整合复杂搜索交互与推理；Agent-R1 实现端到端 RL 训练的全流程 DR 框架，支持多工具协调与自适应规划。

其他：WebThinker 用 Iterative Online DPO 交织搜索 / 浏览 / 报告；Pangu DeepDiver 用 SFT+RL 两阶段课程，实现搜索强度自适应。

#### 奖励与策略优化

多数开源 RL 采用规则 - based 奖励（检索相关性、信息准确性、工具调用成功等）。

主流算法：PPO、GRPO (Group Relative Policy Optimisation)。

Table 3 continued from previous page

DR Agent	SFT	RL	Base Model	Data	Reward Design	Release
ReSearch [16]	□	GRPO [92]	Qwen2.5-7B-Inst, Qwen2.5-32B-Inst	2WikiMultiHopQA Rule-Outcome		Mar-2025
R1-Searcher [96]	■	Reinforce++ [43], GRPO [92]	Qwen2.5-7B-Inst, LLaMA-3.1-8B-Inst	2WikiMultiHopHotpotQA	Rule-Outcome	Mar-2025
Search-R1 [52]	■	PPO [90], GRPO [92]	Qwen2.5-3B/7B, LLaMA3.2-3B-Inst	NQ, HotpotQA	Rule-Outcome	Mar-2025
Nouswise [75]	■	■	Nouswise	□	■	Mar-2025
DeepResearcher [135]	□	GRPO [92]	Qwen2.5-7B-Inst	NQ, HotpotQA	Rule-Outcome	Apr-2025
Genspark Super Agent [106]	□	■	Mixture of Agents	□	■	Apr-2025
WebThinker [59]	■	Iterative Online DPO [86]	QwQ-32B	Expert Dataset	Rule-Outcome	Apr-2025
SWIRL [30]	□	Offline-RL	Gemma-2-27B	HotPotQA	□	Apr-2025
SimpleDeepSearcher [100]	■	PPO [90]	Qwen-2.5-7B-In, Qwen-2.5-32B-In, Deepseek-Distilled-Qwen-32B, QwQ-32B	NQ, HotpotQA, 2WikiMultiHopQA, Musique, SimpleQA, MultiHop-RAG	Process-based reward	Apr-2025
PANGU DEEP-DIVER [94]	■	GRPO [92]	Pangu-7B-Reasoner	WebPuzzle	Rule-Outcome	May-2025
Tool-Star [22]	■	GRPO [92]	Qwen-2.5	NuminaMath, HotpotQA, 2WikiMultiHopQA	Rule-Outcome	May-2025
WebDancer [120]	■	DAPO [130]	Qwen-2.5-7B/32B, QwQ-32B, DeepSeek-R1, GPT-4o	CRAWLQA, E2HQA	Rule-Outcome	May-2025
O-agents [140]	□	□	GPT-4o, GPT-4.1, Claude-3.7-Sonnet, DeepSeek-R1, Gemini-2.5	□	□	Jun-2025
Kimi-Researcher [70]	□	REINFORCE	Kimi k1.5/k2	□	Rule-Outcome	Jun-2025
WebSailor [57]	■	DUPPO	Qwen-2.5-3B/7B/32B/72B	SailorFog-QA	Rule-Outcome	Jul-2025
Agent-KB [103]	□	□	GPT-4o, GPT-4.1, Claude-3.7-Sonnet, o3-mini, Qwen-3 32B, DeepSeek-R1	□	□	Jul-2025
WebShaper [104]	■	GRPO [92]	Qwen-2.5-3B/7B/32B/72B, QwQ-32B	WebShaper	Rule-Outcome	Jul-2025

Continued on next page

# 01. deepresearch技术架构

## ——基础模型增强

参数化方法 (SFT/RL) 面临参数扩展、海量结构化经验数据管理、训练算法复杂等问题；非参数化方法在运行时通过与环境交互优化，数据与计算开销更低。

### 非参数化持续学习

**核心：**不更新模型参数，通过优化外部记忆、工作流、工具配置实现能力提升，轻量化、适合复杂 DR 架构。

**主流范式：**Case-Based Reasoning (CBR，基于案例推理)

**思想：**从外部案例库动态检索、适配、复用结构化问题解决轨迹，实现在线上线下文自适应与任务级泛化。

与 RAG 区别：RAG 多是静态非结构化文本检索；CBR 是轨迹级、以推理为中心的记忆组织。

### 典型应用：

DS-Agent、LAM：将 CBR 用于数据科学、功能测试生成；

Agent K：基于奖励驱动的记忆策略，实现真正自进化；

AgentRxiv：构建类 arXiv 共享预印本库，智能体共享研究成果，不修改参数即可增强能力

Table 3 continued from previous page

DR Agent	SFT	RL	Base Model	Data	Reward Design	Release
Cognitive Kernel-Pro [109]	■	□	Claude-3.7-Sonnet, CK-Pro-8B	OpenWebVoyager, Multi-hop URLQA, AgentWebQA, WebWalkerQA, DocBench, TableBench, BrowseComp	□	Aug-2025
WebWatcher [29]	□	GRPO [92]	Qwen-2.5-VL-32B	VL, Long-tail VQA, Hard VQA	Rule-Outcome	Aug-2025
MiroRL [107]	■	GRPO [92]	Qwen3-14B	MiroRL-GenQA	Rule-Outcome	Aug-2025

# 01. Deep Research 的技术框架——推理能力增强 检索增强生成 (RAG) 的进化：从“静态检索”到“智能体化检索 (Agentic RAG) ”

RAG 是 DR Agents 获取外部知识的核心技术，近年来从“单一检索 - 生成”流水线，演进为具备动态推理、多步检索、实时适配的智能体化范式，核心进展如下：

## 1. 检索机制：从“静态单轮”到“迭代多步与动态优化”

早期 RAG 依赖单轮检索（如从 Wikipedia 获取固定文档），难以应对复杂多跳问题，最新研究通过多轮检索规划与策略优化突破局限：

### · 迭代检索与查询精炼：

FLARE (Zhang et al., 2024) 、Self-RAG (Asai et al., 2023) 等框架让模型在生成过程中自主判断是否需要补充检索；  
RAG-RL (Huang et al., 2025) 引入强化学习与课程学习，让模型学习“何时检索”“检索什么”。

### · 多源与跨模态检索扩展：

检索源从结构化数据库 (Wikipedia) 扩展到大规模网页语料、多模态数据（如医学影像、图表）；  
DeepRAG (2025) 提出二叉树检索机制，递归生成子查询并构建检索轨迹，平衡“内部知识”与“外部检索”。

## 2. Agentic RAG：融合推理与决策，突破传统 RAG 局限

为解决传统 RAG “无法动态适配任务需求”“复杂推理能力弱”问题，最新研究将 RAG 升级为智能体化架构，核心特征包括：

### · 推理层与检索层深度耦合：

在检索前增加“任务分析与查询规划”步骤，再针对性检索各维度数据；

GeAR (2024) 通过知识图谱建模实体关联，在多跳检索中，显著提升信息关联度与检索精度。

### · 动态 workflow 调整：

模型可根据检索结果实时调整策略，例如检索到“某数据来源不可靠”时，自动切换到其他权威数据源（如从普通网页转向政府统计 API）。

# 03.最新研究方法——

## “模型 - 上下文 - 交互”三维缩放框架

### 1. 交互缩放 (Interactive Scaling)

把「交互深度」提升到和 模型参数量、上下文长度同等地位的 scaling 维度

·通过强化学习让模型与环境进行更深、更频繁的交互。

支持单任务最多 600 次工具调用，远超此前开源模型的 100 次上限，实现持续多轮推理和复杂研究流程。

借助环境反馈和外部信息获取修正错误、优化轨迹，解决了传统长推理链性能退化的问题。

·工具内部嵌入轻量 LLM

### 2. 上下文管理新范式

保留完整 Thought + Action

·只保留最近 K 个 Tool Observation

早期工具结果直接 mask 掉 (不是删除 reasoning 轨迹)

### 3. Multi-Document QA + 知识图驱动的数据合成

数据构建流程极其先进：

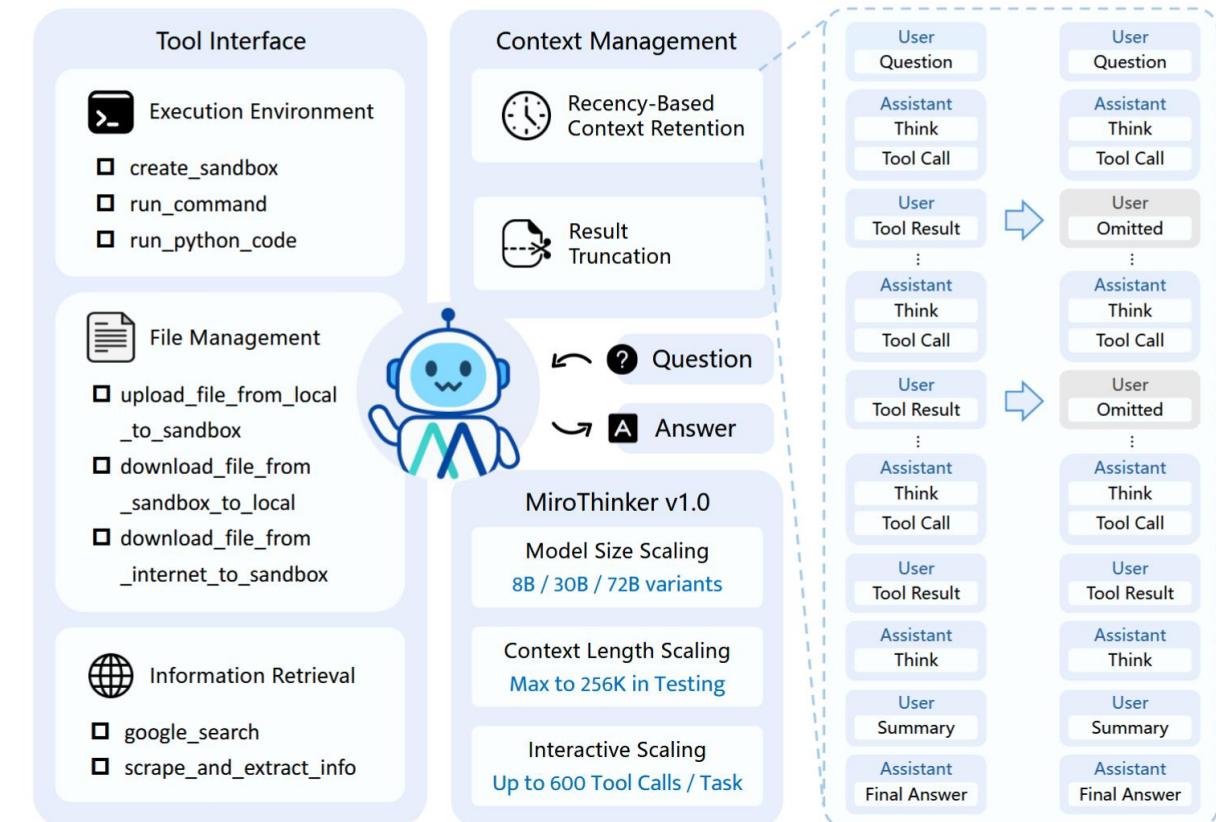
多源文档 (Wiki / Web)

构建 文档知识图

抽取跨文档事实

事实约束混淆 (Constraint Obfuscation)

生成必须 multi-hop 才能回答的问题



**Figure 2:** Overview of the MiroThinker v1.0 agent architecture. The framework integrates a structured tool interface, i.e., execution environment, file management, and information retrieval, with a simple recency-aware context management to support interactive scaling. On the right, an agentic trajectory example illustrates the recency-based context retention mechanism, where tool outputs from earlier turns are omitted to maintain context efficiency.

# 01. Deep Research 的技术框架现状及其突破点

主要类别	核心定位	架构特点与关键突破点	代表系统
工具利用与环境交互（DR的“手脚”）	DR 需主动与外部环境交互以获取信息，核心能力覆盖“网页 - 文档 - 专用工具”三类交互场景	网页交互技术：从“简单 API 搜索”进化为“语义化网页导航”，支持动态内容渲染（JavaScript 执行）、登录态管理、多页跳转，甚至模拟人类浏览行为（如滚动、点击）。(如下图是两者对比)	代表系统：Nanobrowser（AI 专用浏览器，优化研究型网页提取）、AutoGLM（全浏览器自动化，支持表单填写、GUI 操作）。
		内容处理能力：从“纯文本提取”进化为“多模态内容解析”，支持 PDF（含公式 / 表格）、结构化数据（JSON/CSV）、图像（如科研图表）的语义提取与整合。	代表系统：dzhng/deep-research（学术论文专用解析，提取实验方法、结果图表）、nickscamara/open-deep-research（多格式内容归一化，转化为统一知识表征）。
		专用工具整合：从“通用 API 调用”进化为“领域工具链编排”，MCP调用等，通过 ToolLLM（支持 1.6 万 + API）、AssistGPT 等框架，动态选择并调用专业工具（如统计分析软件、金融数据库）。	代表系统：Manus（150 + 服务集成，支持科研工具链自动化）、n8n（工作流平台，可配置“数据采集 → 分析 → 报告”工具流水线）。

# 01. deepresearch技术架构

## ——深度研究 (DR) 智能体的工具使用能力

### 三大核心工具模块详解

#### 1. 代码解释器 (Code Interpreter)

- 核心功能：支持智能体在推理阶段执行脚本，完成数据处理、算法验证、模型仿真等任务。多数 DR 智能体（除 CoSearchAgent 外）均嵌入脚本执行环境，主要依赖 Python（如 Aider）、Java 等工具，实现动态脚本编排、文献驱动分析与实时计算推理。

#### 2. 数据分析 (Data Analytics)

- 核心功能：将原始检索信息转化为结构化洞察，通过统计分析、交互式可视化、定量模型评估等，加速假设验证与决策。

#### 3. 多模态处理与生成 (Multimodal Processing and Generation)

- 核心功能：统一处理文本、图像、音视频等异构数据，丰富上下文理解，拓展输出形式。

### 技术突破：DR 智能体与计算机使用 (Computer Use) 融合

DR 智能体边界进一步扩展，集成计算机辅助任务执行能力，实现更自主的复杂任务处理。

#### 典型案例：AutoGLM Rummation

基于强化学习 + 自我反思迭代机制，强化多步推理与高级函数调用能力；

可自主交互网页环境、执行代码、调用外部 API，完成数据检索、分析及结构化报告生成；

对比 OpenAI DR：更侧重实际执行自主性，能将抽象分析转化为网页交互、实时数据处理等具体操作；

解决模拟浏览器局限：融合真实浏览器交互，可访问 CNKI、小红书、微信公众号等需用户认证的资源，大幅提升信息获取与真实任务执行的适配性。

Table 2: Comparison of DR Agents with Tool Use Capabilities

■ = Involved, ■ = Non Disclosure, □ = Not present

DR Agent	Code Interpreter	Data Analytics	Multimodal	Release
CoSearchAgent [31]	□	■	□	Feb-2024
Storm [91]	■	□	□	Jul-2024
The AI Scientist [63]	■	□	□	Aug-2024
Agent Laboratory [89]	■	□	□	Jan-2025
Agentic Reasoning [122]	■	□	□	Feb-2025
AutoAgent [101]	■	□	■	Feb-2025
Genspark DR [106]	■	■	■	Feb-2025
Grok DeepSearch [124]	■	■	■	Feb-2025
OpenAI DR [78]	■	■	■	Feb-2025
Perplexity DR [81]	■	■	■	Feb-2025
Towards an AI co-scientist [34]	□	■	■	Feb-2025
Agent-R1 [80]	■	□	■	Mar-2025
AutoGLM Rummation [137]	■	□	■	Mar-2025
Copilot Researcher [69]	■	■	■	Mar-2025
Manus [66]	■	■	■	Mar-2025
OpenManus [60]	■	■	■	Mar-2025
OWL [12]	■	■	■	Mar-2025
H2O.ai DR [39]	■	■	■	Mar-2025
Genspark Super Agent [106]	■	■	■	Apr-2025
WebThinker [59]	■	■	■	Apr-2025
Suna Ai [4]	■	■	■	Apr-2025
Tool-Star [22]	■	■	■	May-2025
AgenticSeek [67]	■	■	■	May-2025
Alita [84]	■	■	■	May-2025
DeerFlow [20]	■	■	■	May-2025
O-agents [140]	■	■	■	Jun-2025
Kimi-Researcher [70]	■	■	■	Jun-2025
Agent-KB [103]	■	■	■	Jul-2025
AWorld [8]	■	■	■	Jul-2025
Cognitive Kernel-Pro [109]	■	■	■	Aug-2025
WebWatcher [29]	■	■	■	Aug-2025
MiroRL [107]	■	■	■	Aug-2025

# 01. deepricerca技术架构——基于 API 的检索与基于浏览器的检索

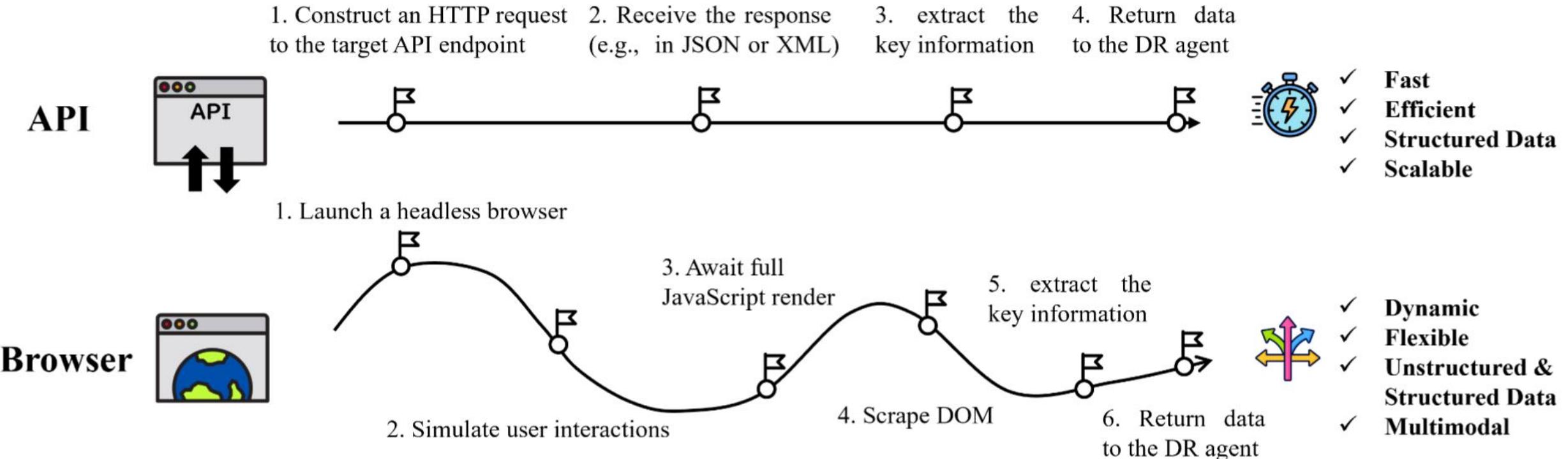


Figure 2: General Comparison of API-Based and Browser-Based Retrieval Workflow.

- 1) 基于API的SE，与结构化数据源交互，如搜索引擎API或科学数据库API，实现高效检索有组织信息。
- 2) 基于浏览器的软件开发系统，模拟网页的类人交互，便于动态或非结构化内容的实时提取，提升外部知识的全面性。

Table 1: Comparison of DR Agents with Search Engine Details

# 01. deepresearch技术架构

## ——基于 API 的检索

### 基于 API 的检索 (API-based retrieval)

- 核心优势：速度快、效率高、可扩展性强，延迟与计算开销较低，适合结构化、高吞吐量的知识获取。
- 典型应用与实现：
  - Gemini DR、Grok DeepSearch 等通过 Google Search、arXiv、Wikipedia 等 API 实现大规模、实时或持续更新的检索；
  - Cognitive Kernel-Pro 借助 DuckDuckGo 构建开源低成本 DR 流程；
  - Agentic Reasoning、R1-Search 等明确教会模型“何时搜、搜什么、如何整合检索证据”；
  - PANGU DeepDiver 用强化学习适配任务难度调整搜索强度；
  - Agent Laboratory、AI Scientist 分别调用 arXiv、Semantic Scholar API 用于文献综述、创意新颖性验证；
  - DeepRetrieval、Search-o1 等针对生物医学、通用推理场景，结合 PubMed、Bing、Jina Reader 等 API 优化查询与内容提取。
- 局限性：难以处理深度嵌套、客户端 JS 渲染、交互组件、需认证的内容，覆盖范围受限。

■ = Primary focus, ■ = Secondary/minor focus, □ = Not present

DR Agent	Search Engine API	Browser	GAIA HLE	Benchmark Other QA	Base Model	Release
Avatar [123]	■	□	□ □	Stark	Claude-3-Opus, GPT-4	Feb-2024
CoSearch-Agent [31]	■	□	□ □	□	GPT-3.5-turbo	Feb-2024
MMAC-Copilot [97]	■	□	■ □	□	GPT-3.5, GPT-4	Mar-2024
Storm [91]	■	□	□ □	FreshWiki	GPT-3.5-turbo	Jul-2024
OpenResearcher [136]	■	□	□ □	Privately Collected QA Data	DeepSeek-V2-Chat	Aug-2024
The AI Scientist [63]	■	□	□ □	MLE-Bench	GPT-4o, o1-mini, o1-preview	Aug-2024
Gemini DR [33]	■	■	□ ■	GPQA	Gemini-2.0-Flash	Dec-2024
Agent Laboratory [89]	■	□	□ □	MLE-Bench	GPT-4o, o1-preview	Jan-2025
Search-o1 [58]	■	□	□ □	GPQA, NQ, TriviaQA	QwQ-32B-preview	Jan-2025
WebWalker [121]	□	□	□ □	WebWalkerQA	GPT-4o, Qwen-2.5	Jan-2025
Agentic Reasoning [122]	■	□	□ □	GPQA	DeepSeek-R1, Qwen2.5	Feb-2025
AutoAgent [101]	□	■	■ □	□	Claude-Sonnet-3.5	Feb-2025
Grok DeepSearch [124]	■	■	□ □	GPQA	Grok3	Feb-2025
OpenAI DR [78]	□	■	■ ■	■	GPT-o3	Feb-2025
Perplexity DR [81]	■	■	□ ■	SimoleQA	Flexible	Feb-2025
Towards an AI co-scientist [34]	■	□	□ □	GPQA	Gemini 2.0	Feb-2025
Nouswise [75]	□	□	□ □	□	—	Mar-2025
AgentRxiv [88]	■	□	□ □	GPQA, MedQA	GPT-4o-mini	Mar-2025
Agent-R1 [80]	■	□	□ □	HotpotQA	Qwen2.5-1.5B-Inst	Mar-2025
AutoGLM Rumination[137]	□	■	□ □	GPQA	GLM-Z1-Air	Mar-2025
Copilot Researcher [69]	□	■	□ □	□	o3-mini	Mar-2025
H2O.ai DR [39]	■	■	■ □	□	h2ogpt-oasst1-512-12b	Mar-2025
Manus [66]	■	■	□ □	□	Claude3.5, GPT-4o	Mar-2025
Openmanus [60]	■	■	□ □	□	Claude3.5, GPT-4o	Mar-2025

# 01. deepresearch技术架构

## ——基于浏览器的检索

### 基于浏览器的检索 (Browser-based retrieval)

- 核心优势：通过模拟人类浏览器交互，实现对动态、多模态、非结构化、深层嵌套网页内容的灵活、交互式访问，弥补 API 检索的覆盖短板。
- 典型应用与实现：
  - 闭源方案（如 Gemini 2.5 DR、Grok DeepSearch、OpenAI DR）虽未公开细节，但从交互能力推测采用无头浏览器框架；
  - 开源方案中，AutoAgent、DeepResearcher 等在 BrowserGym 等环境中实现滚动、交互、分块读取与增量信息聚合；
  - AutoGLM、Genspark Super Agent 等通过“计划 - 执行”循环、多智能体协作，结合浏览与反思 / 写作流程；
  - Tool-Star、AgenticSeek 等明确区分搜索工具与浏览器代理，结合元搜索、隐身浏览器实现点击、表单提交等操作；
  - WebThinker、WebDancer 等以“自主搜索 + 页面导航”为最小工具集，通过 React 闭环实现“思考 - 搜索 - 撰写”实时协同策略（突破“先搜索后撰写”的割裂模式）；
  - WebWatcher 结合 SerpApi 与 OCR 图像处理模拟浏览器交互，支持多模态检索。
- 局限性：延迟更高、资源消耗更大，处理网页多样性与错误的复杂度显著提升。

Table 1 continued from previous page

DR Agent	Search Engine API	Search Engine Browser	GAIA HLE	Benchmark Other QA	Base Model	Release
OWL [12]	■	■	■ □	□	Deepeek-R1, Gemini2.5-Pro, GPT-4o	Mar-2025
R1-Searcher [96]	■	□	□ □	2WikiMultiHopQA, HotpotQA	Llama3.1-8B-Inst, Qwen2.5-7B	Mar-2025
ReSearch [16]	■	□	□ □	2WikiMultiHopQA, HotpotQA	Qwen2.5-7B, Qwen2.5-7B-Inst	Mar-2025
Search-R1 [52]	■	■	□ □	2WikiMultiHopQA, HotpotQA, NQ, TriviaQA	Llama3.2-3B, Qwen2.5-3B/7B	Mar-2025
DeepResearcher [135]	□	■	■ □	HotpotQA, NQ, TriviaQA	Qwen2.5-7B-Inst	Apr-2025
Genspark Super Agent [106]	■	■	■ □	□	Mixture of Agents*	Apr-2025
WebThinker [59]	■	■	■ ■	GPQA, WebWalkerQA	QwQ-32B	Apr-2025
SWIRL [30]	■	□	□ □	HotQA, BeerQA	Gemma 2-27b	Apr-2025
SimpleDeepSearcher [100]	□	■	■ □	2WikiMultiHopQA	Qwen-2.5-7B-In, Qwen-2.5-32B-In, DeepSeek-Distilled-Qwen-2.5-32B, QwQ-32B	Apr-2025
Suna AI [4]	■	■	□ □	□	GPT-4o, Claude	Apr-2025
Tool-Star [22]	■	■	■ ■	WebWalker, HotpotQA, 2WikiMultiHopQA	Qwen-2.5	May-2025
WebDancer [120]	■	■	■ □	WebWalkerQA	Qwen-2.5, QwQ-32B, DeepSeek-R1, GPT-4o, GPT-4o, DeepSeek-R1, Claude	May-2025
AgenticSeek [67]	□	■	□ □	□	DeepSeek-R1, Claude	May-2025
Alita [84]	■	■	■ □	PathVQA	GPT-4o, Claude-Sonnet-4	May-2025
DeerFlow [20]	■	□	□ □	□	Douba-1.5-Pro, 32k, DeepSeek-R1, GPT-4o, Qwen	May-2025
PANGU DEEP-DIVER [94]	■	□	□ □	C-SimpleQA, HotpotQA, ProxyQA	Pangu-7B-Reasoner	May-2025
O-agents [140]	■	□	■ □	□	GPT-4o, GPT-4.1, Claude-3.7-Sonnet, DeepSeek-R1, Gemini-2.5	Jun-2025
Kimi-Researcher [70]	■	■	□ ■	SimpleQA	Kimi k1.5/k2	Jun-2025
WebSailor [57]	■	■	■ □	SimpleQA	Qwen-2.5	Jul-2025
Agent-KB [103]	■	□	■ □	SWE-bench	GPT-4o, GPT-4.1, Claude-3.7-Sonnet, o3-mini, Qwen-3, DeepSeek-R1	Jul-2025

Continued on next page

# 01. deepresearch技术架构

## ——API 与浏览器的深度融合与 AI 原生浏览器崛起

### ·混合检索架构成为主流

2025-2026 年主流 DR 系统（如 OpenAI DR、Gemini DR V2）均采用“API + 浏览器”双引擎架构：

API 负责结构化数据的高效获取（如 Google Search API、arXiv API）；浏览器处理动态 / 认证场景（如登录 CNKI、小红书等平台）。例如 AutoGLM Rumination 通过真实浏览器交互，突破模拟环境限制，实现对用户认证资源的可靠访问，信息覆盖范围较传统方案提升 40%。

### ·AI 原生浏览器解决传统瓶颈

针对传统浏览器渲染慢、反爬限制等问题，2025 年出现 Browserbase、Comet (Perplexity)、Fellou 等 AI 原生浏览器，具备三大核心优化：

- 结构化 DOM 视图：支持程序级页面遍历，避免坐标依赖的脆弱操作；
- 异步无头执行：减少视觉渲染开销，页面加载速度提升 3 倍；
- 内置视觉 - 语言模型 (VLM)：自动突破登录墙、验证码等反爬机制，成功率超 85%。

Table I continued from previous page

DR Agent	Search Engine API	Search Engine Browser	GAIA HLE	Benchmark Other QA	Base Model	Release
WebShaper [104]	■	■	■ □	WebWalkerQA	Qwen-2.5, QwQ-32B	Jul-2025
Deep Researcher with Test-Time Diffusion [40]	■	□	■ ■	□	Gemini-2.5-Pro	Jul-2025
ChatGPT-Agent [77]	□	□	□ □	□	—	Jul-2025
AWorld [8]	■	■	■ □	HotpotQA	Gemini-2.5-Pro, GPT-4o	Jul-2025
Cognitive Kernel-Pro [109]	■	□	■ □	AgentWebQA, WebWalkerQA, Multi-hop URLQA, DocBench, TableBench	Claude-3.7-Sonnet, CK-Pro-8B	Aug-2025
WebWatcher [29]	■	■	□ ■	Browsercom-VL, LiveVQA, MMSearch	Qwen-2.5-VL-32B	Aug-2025
WideSearch [119]	■	□	□ □	WideSearch	DeepSeek-R1, Doubao-Seed-1.6, Claude Sonnet 4, Gemini-2.5-Pro	Aug-2025
MiroRL [107]	■	■	■ □	□	Qwen3-14B	Aug-2025

# 01.deeppresearch技术架构

## ——“模型 - 上下文 - 交互”三维缩放框架----单任务 600 次工具调用

四大核心支柱

1. 强化学习驱动的交互策略优化（核心动力）

采用**Group Relative Policy Optimization (GRPO)** 算法：

通过“群体相对优势”计算优化 agent 的交互策略，避免单一轨迹优化的局部最优。

2. 高效上下文管理技术（交互频次突破的关键）

要支撑单任务 600 次工具调用（远超传统模型的 100 次），必须解决上下文窗口溢出问题：

·近期优先上下文保留：仅保留最新 K 条工具响应，但完整保留“思考 - 行动”轨迹。

·结果截断机制：对代码执行、命令运行等工具的超长输出进行截断，既防止上下文溢出，又不影响模型对工具输出的核心理解。

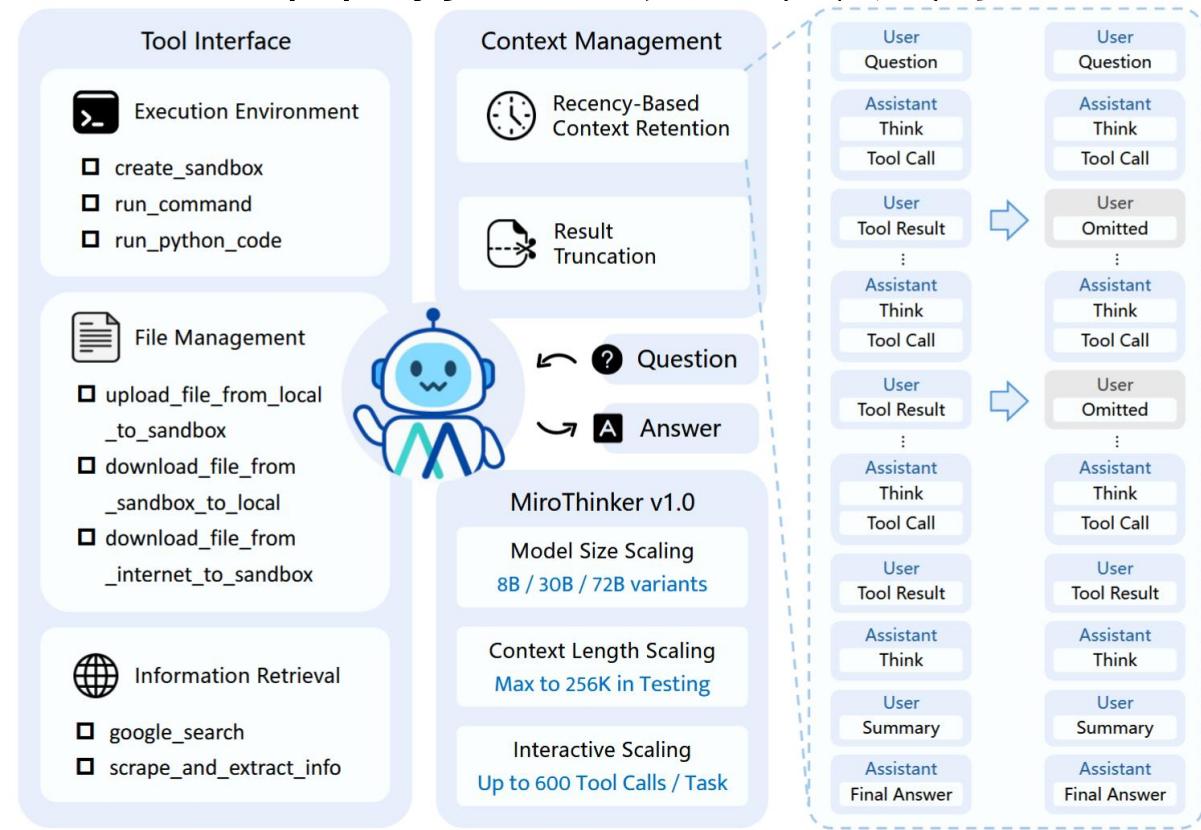
3. 模块化工具接口与交互环境（交互能力基础）

构建了支撑多场景研究交互的工具链，让模型能通过工具与真实环境深度联动：执行环境工具、文件管理工具、智能信息检索工具

4. 三阶段训练流水线（交互能力的习得路径）

通过分层训练让模型逐步掌握“深度交互”能力，避免直接训练导致的交互无序。

三个阶段分别是：代理监督微调、代理偏好优化（、代理强化学习



**Figure 2:** Overview of the MiroThinker v1.0 agent architecture. The framework integrates a structured tool interface, i.e., execution environment, file management, and information retrieval, with a simple recency-aware context management to support interactive scaling. On the right, an agentic trajectory example illustrates the recency-based context retention mechanism, where tool outputs from earlier turns are omitted to maintain context efficiency.

# 03.最新研究方法

## —— 从“文本域”扩展到“视觉 - 语言多模态域”

聚焦视觉 - 语言多模态深度推理，核心创新集中在数据构建、工具与轨迹训练、基准测试、模型优化四大维度

### 1.多模态训练数据构建

从权威来源递归爬取超链接，构建多跳知识图谱，生成含复杂推理链的文本 QA，Level 2 数据通过“实体模糊化”提升推理难度。

### 2.工具集成与轨迹训练

- 整合 Web 图像搜索、文本搜索、网页访问、代码解释器、OCR 五大类工具，覆盖多模态推理全场景。

- 基于 ReAct 框架，用 GPT-4o 自动生成“思考 - 动作 - 反馈”轨迹，锚定真实工具行为，经“答案匹配、步骤一致、最少 3 次工具调用”自动化生成高质量轨迹。

- 采用“SFT 冷启动 + GRPO 强化学习”范式，SFT 教模型工具使用逻辑，GRPO 通过组内相对优势优化决策，平衡格式合规与语义准确性。

### 3.基准测试创新

提出 BrowseComp-VL 基准，分为两级难度，包含 5 大领域 17 个子领域，要求跨模态推理 + 外部工具使用，填补多模态 DeepResearch 评估空白。

### 4.模型性能优化

工具调用动态适配任务场景，模型能根据不同基准的任务特性动态调整工具使用比例：如 BrowseComp-VL 以文本搜索为主（62%），LiveVQA 以图像搜索为主（38.2%）。

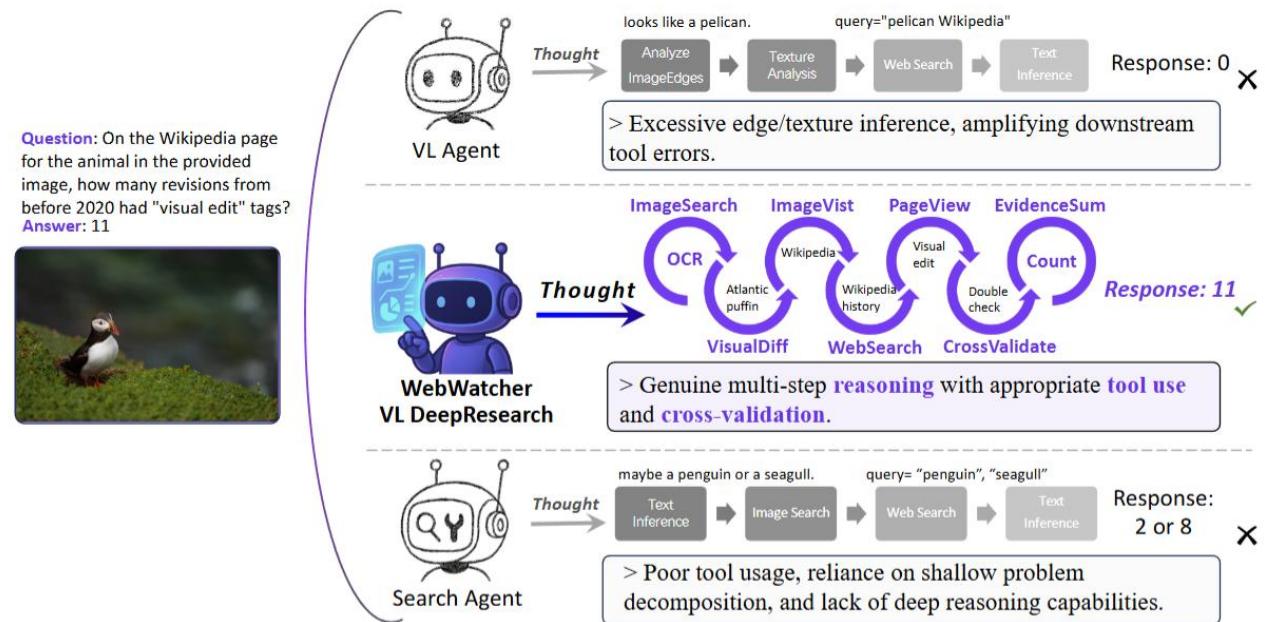


Figure 2: Comparison of VL reasoning agents. WebWatcher resolves the GAIA case that defeats either vision-only reasoning or search-based agents, demonstrating the strength of multi-tool integration and in-depth reasoning generalization.

# 01. Deep Research 的技术框架现状及其突破点

主要类别	核心定位	架构特点与关键突破（进化）点	代表系统
任务规划与执行控制（DR 的“指挥官”）	复杂研究需拆解为多步骤任务并可靠执行，核心突破体现在“规划粒度”与“执行韧性”	任务规划机制：从“线性任务分解”（如 MetaGPT 的简单步骤拆分）进化为“分层动态规划”，支持基于中间结果调整策略，甚至引入“树搜索算法”探索最优研究路径。	代表系统：OpenAI/AgentsSDK（目标分解 + 执行跟踪 + 自适应 refinement）、Flowith/OracleMode（优先评估信息质量，动态调整搜索优先级）。
		自主执行与监控：从“无错误处理的 sequential 执行”进化为“并发执行 + 故障恢复”，支持实时知识驱动的工具选择， <b>多路径并行、失败检测、自动重试</b> ，甚至通过强化学习优化执行策略。如 MultiTool-CoT (Inaba et al., 2023) 让 LLM 在推理过程中自主判断是否需要调用外部工具	代表系统：Agent-RL/ReSearch（强化学习驱动，从经验中优化研究流程）、TARS（状态持久化，支持任务中断后恢复）。
		多智能体协作：从“单智能体全流程处理”进化为“专用智能体分工”，如“搜索 Agent + 分析 Agent + 验证 Agent”协同，通过标准化协议（如 A2A）实现信息共享与冲突解决。	代表系统：smolagents/open_deep_research（模块化智能体架构，支持角色定制）、Camel-AI/OWL（开放权重多智能体，动态分配任务）。
知识合成与输出生成（DR 的“成果输出端”）	DR 的终极价值是生成“可信、结构化、可交互”的研究成果，核心能力体现在“信息核验”与“输出适配”	信息评估技术：从“依赖源声誉”进化为“多维度质量评分”，评估维度包括源可信度、内容一致性、证据强度，甚至通过“矛盾检测”识别冲突信息。	代表系统：grapeot/deep_research_agent（证据分类 + 矛盾标记，量化信息可靠性）、Perplexity/DeepResearch（源多样性过滤 + 出版日期权重，优先最新权威信息）。
		报告生成能力：从“简单文本摘要”进化为“结构化研究报告”，支持学术格式（如 IEEE/APA 引文）、逻辑分层（如“背景 - 方法 - 结果 - 结论”）、证据嵌入（实时链接源文档）。	代表系统：mshumer/OpenDeepResearcher（模板化报告生成，含明确引文与论据）、OpenAI/DeepResearch（万字级深度报告，支持章节折叠与源验证）。
		交互式呈现：从“静态文本输出”进化为“动态探索界面”，支持用户钻取细节、验证来源、调整分析视角，甚至通过反馈迭代优化报告。	代表系统：HKUDS/Auto-Deep-Research（动态结果探索，支持引文导航）、OpenManus（交互式可视化，将研究结论转化为图表）。

# 01. deepresearch技术架构

## 静态VS动态工作流程

**静态工作流**: 基于预定义的任务管道，将研究流程拆解为顺序子任务，适合定义清晰、结构化的研究场景（如 AI Scientist、Agent Laboratory）。优点是实现简单、结构清晰；缺点是泛化性差，不同任务需定制化管道。

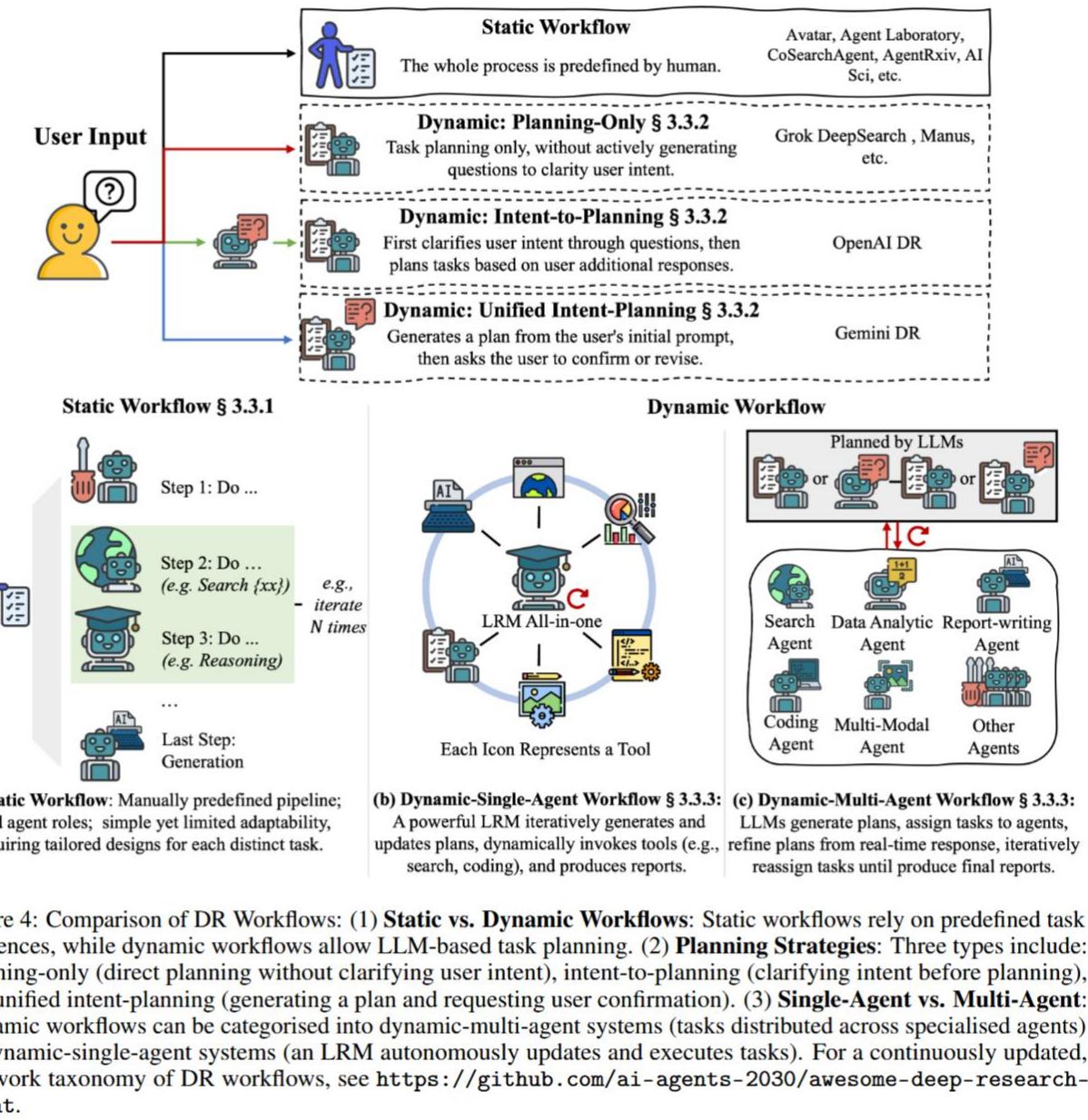
**动态工作流**: 支持自适应任务规划，基于迭代反馈和动态上下文重构任务结构，具备更强的通用性和适应性，适合复杂、知识密集型研究任务。

### 动态工作流的规划策略（基于 LLM 的三种交互方式）

**1.仅规划（Planning-Only）**：基于初始用户提示直接生成任务计划，无额外澄清（如 Grok、H2O、Manus）。

**2.意图到规划（Intent-to-Planning）**：先通过问题澄清用户意图，再生成定制化任务序列（如 OpenAI DR）。

**3.统一意图 - 规划（Unified Intent-Planning）**：先基于初始提示生成初步计划，再与用户交互确认 / 修订（如 Gemini DR），兼顾效率与用户引导。



# 01. deepresearch技术架构

## ——动态工作流的智能体架构

### 动态工作流的智能体架构

**单智能体系统：**将规划、工具调用、执行集成于统一 LLM，形成连贯认知循环，支持端到端强化学习优化（如 Search-o1、WebDancer、Kimi-Researcher）。优点是推理与执行一体化；缺点是对基础模型能力要求高，模块化灵活性受限。

**多智能体系统：**由多个专业化智能体协作执行子任务，通过协调器动态分配 / 重分配任务（如 OpenManus、OWL、AWorld）。优点是适配复杂、可并行的研究任务，灵活性与可扩展性强；缺点是智能体协调复杂，难以实现端到端强化学习优化。

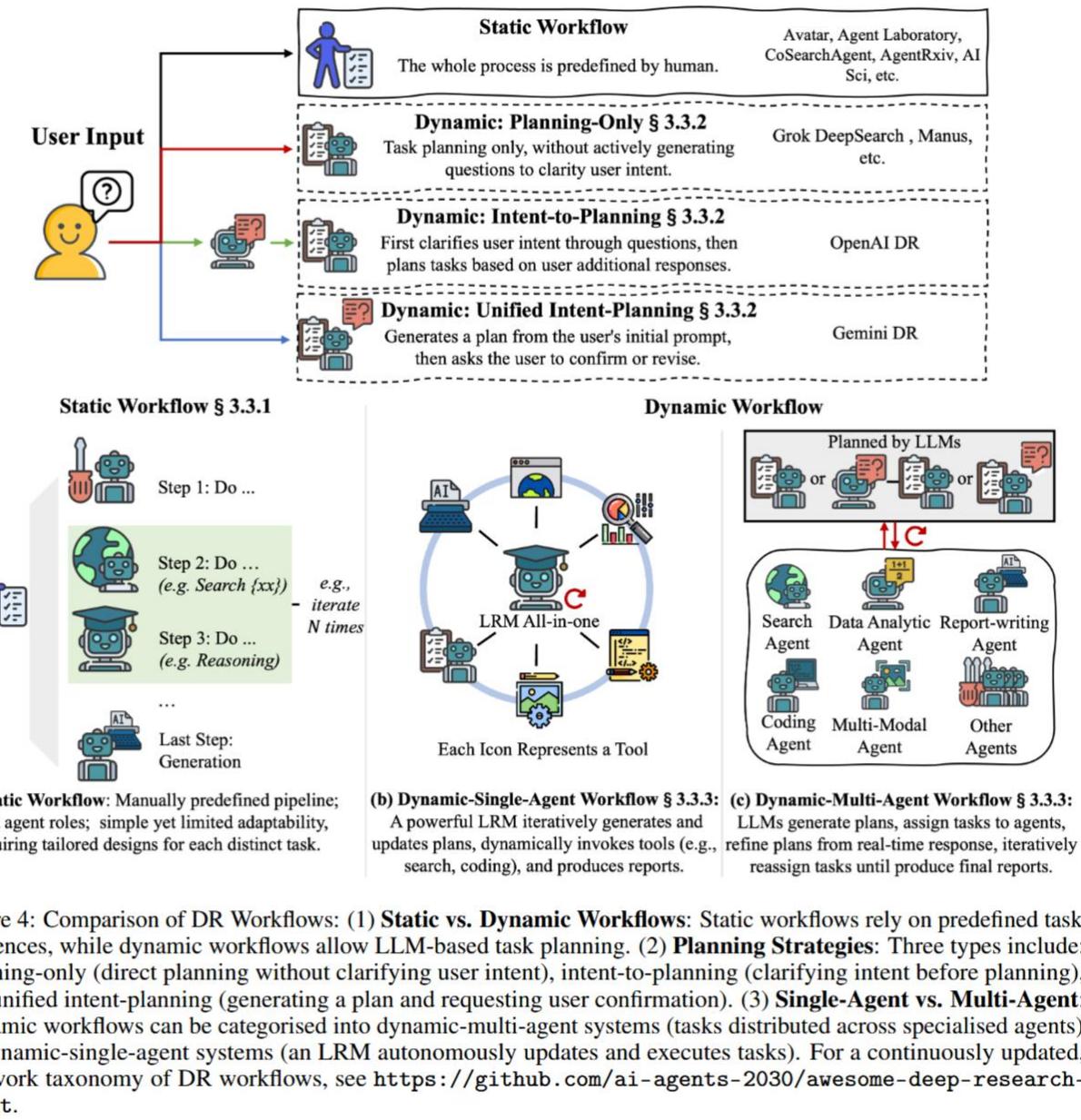
例如：智能体轨迹级数据合成

#### (1) ReAct 单智能体轨迹

思考 (Think)、决策 (Act: 调用搜索 / 代码 / 抽取工具)、观察 (Observe: 工具返回)，强调个人研究闭环，非常适合文献检索，多跳 QA，分析性研究问题

#### (2) MiroFlow 多智能体协作轨迹 (前沿)

多个角色化 Agent 协同完成研究：搜索 Agent，编程 Agent，验证 Agent，总结 Agent

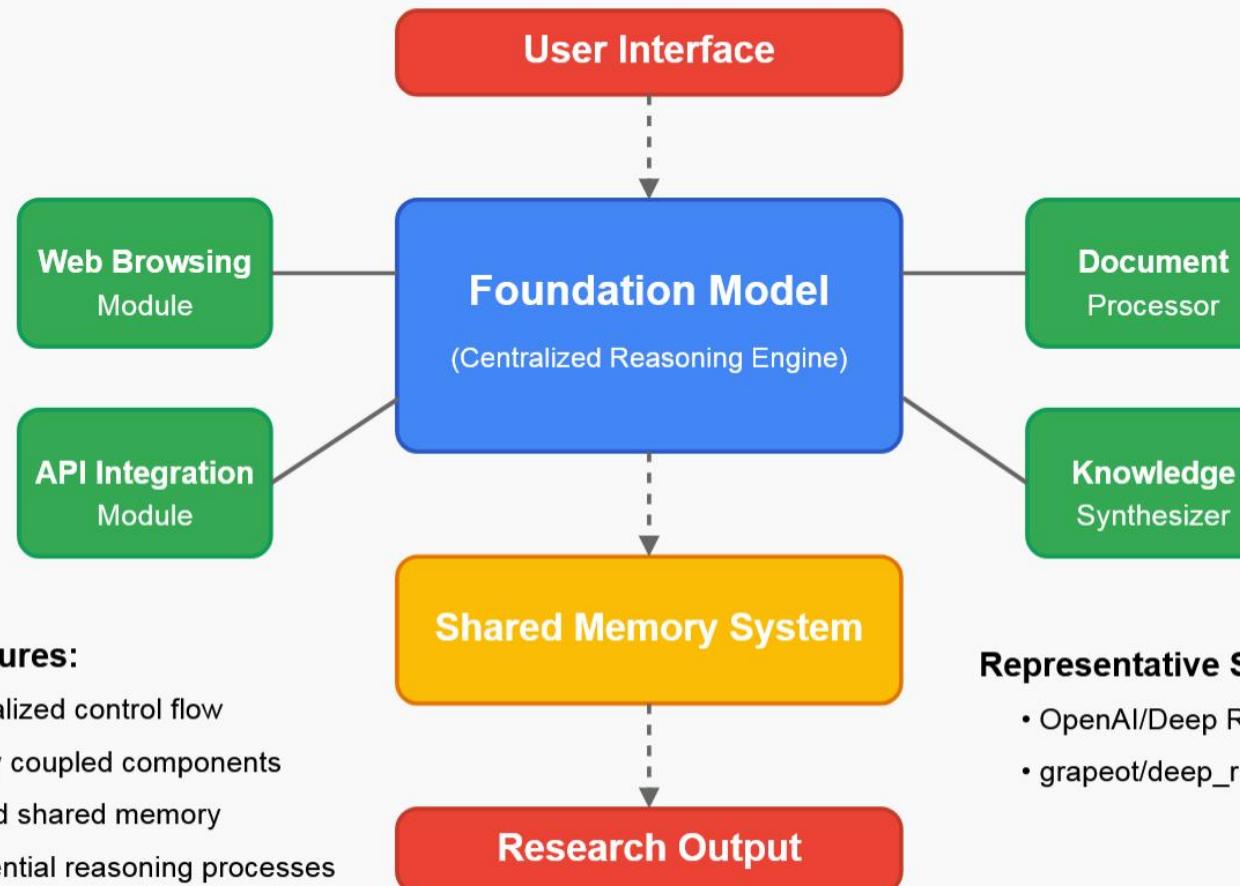


deep research核心架构

02

## 02. 单体架构

### Monolithic Deep Research Architecture



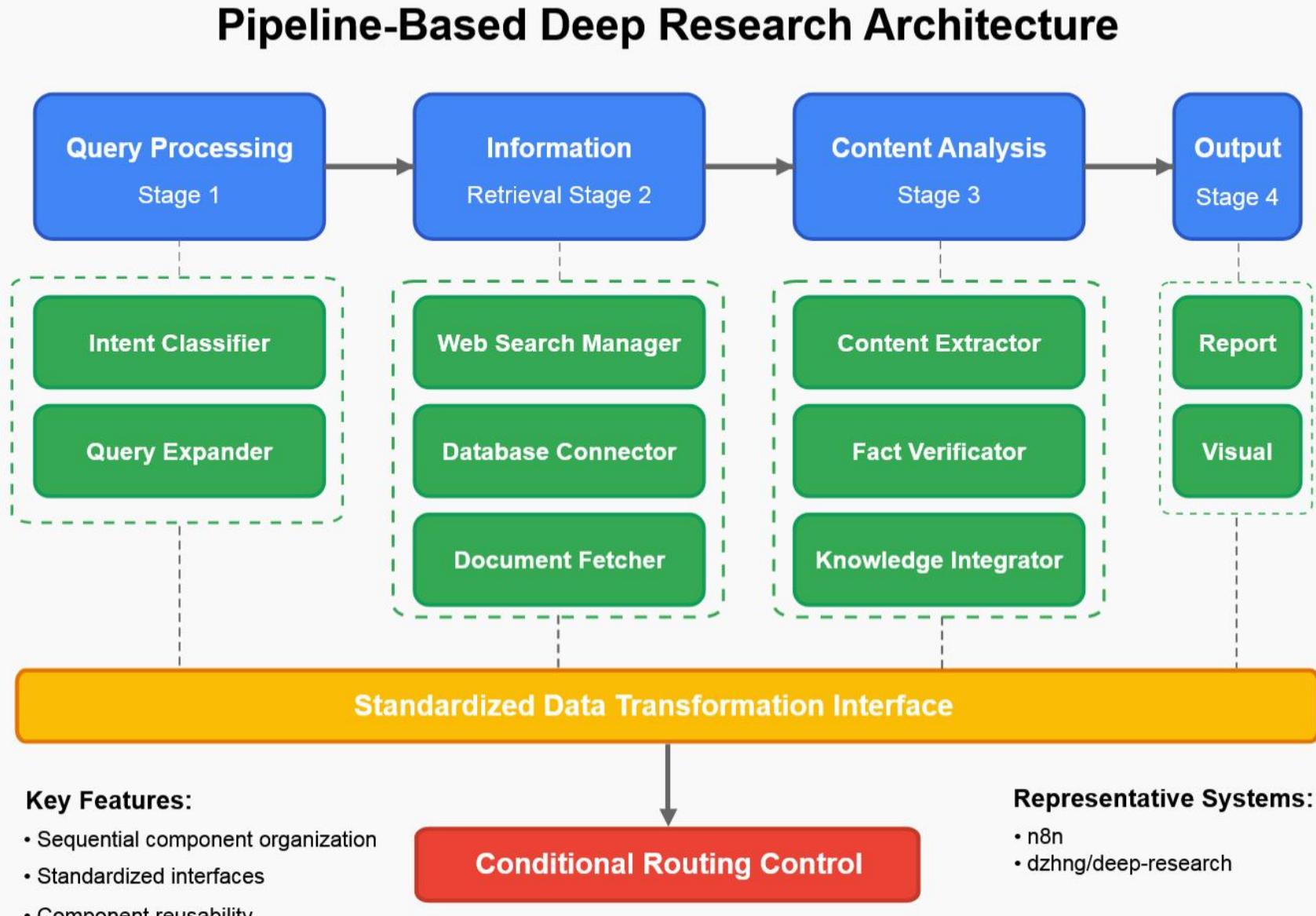
核心推理引擎集中控制所有模块，  
模块耦合紧密，共享统一内存，  
推理连贯性强，代表系统有  
OpenAI/DeepResearch、  
grapeot/deep\_research\_agent。

缺点：

可能存在调用混乱 输出不稳定

Fig. 4. Monolithic Deep Research Architecture

## 02. 流水线架构

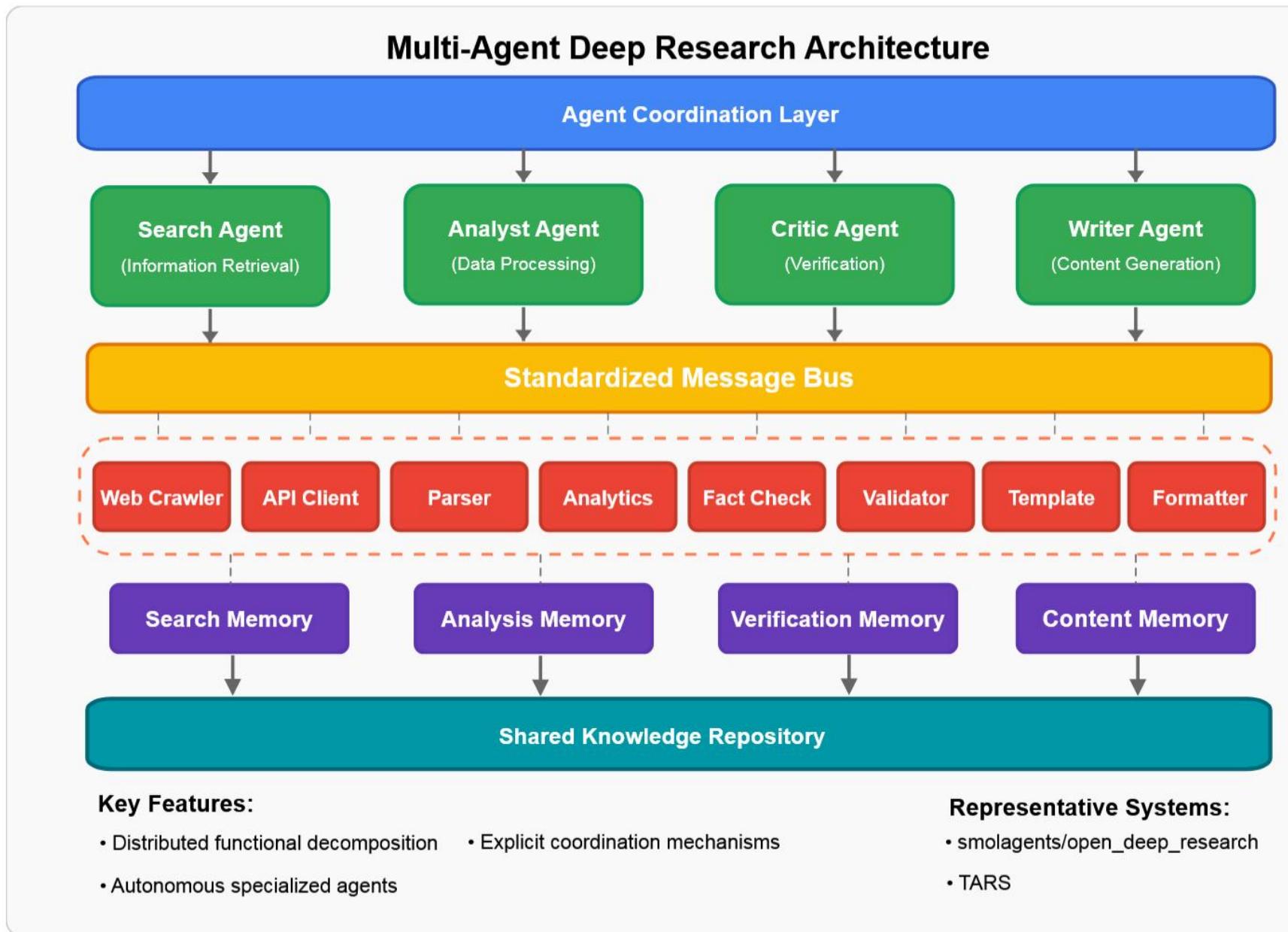


按研究流程拆分为离散处理阶段，通过标准化接口连接，支持组件复用与条件路由，适合定制化 workflow，代表系统有 n8n、dzhng/deep-research。

缺点：

缺乏灵活性 流程固定性

## 02. 多智能体架构



将功能分配给专用智能体（搜索、分析、验证等），通过标准化协议协作，支持并行处理与动态任务分配，代表系统有smolagents/open\_deep\_research、TARS。

缺点：

## 02. 多智能体架构

整个流程体现了  
“用户驱动→LLM 规划→多智能体协作→  
工具增强→成果交付”  
的闭环

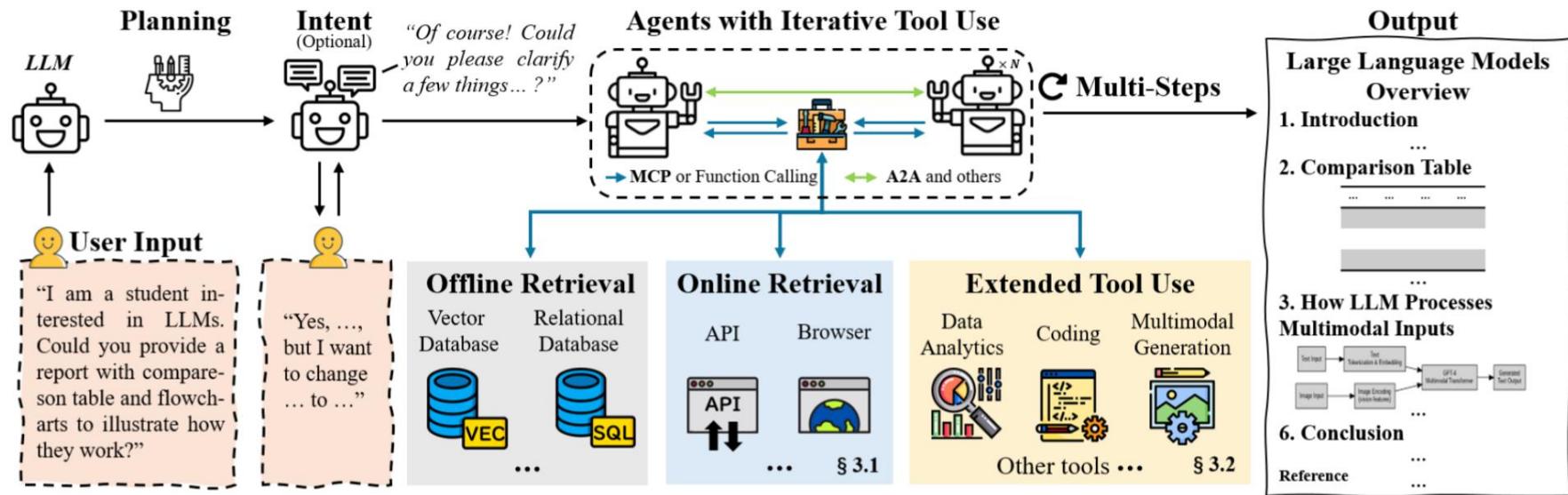
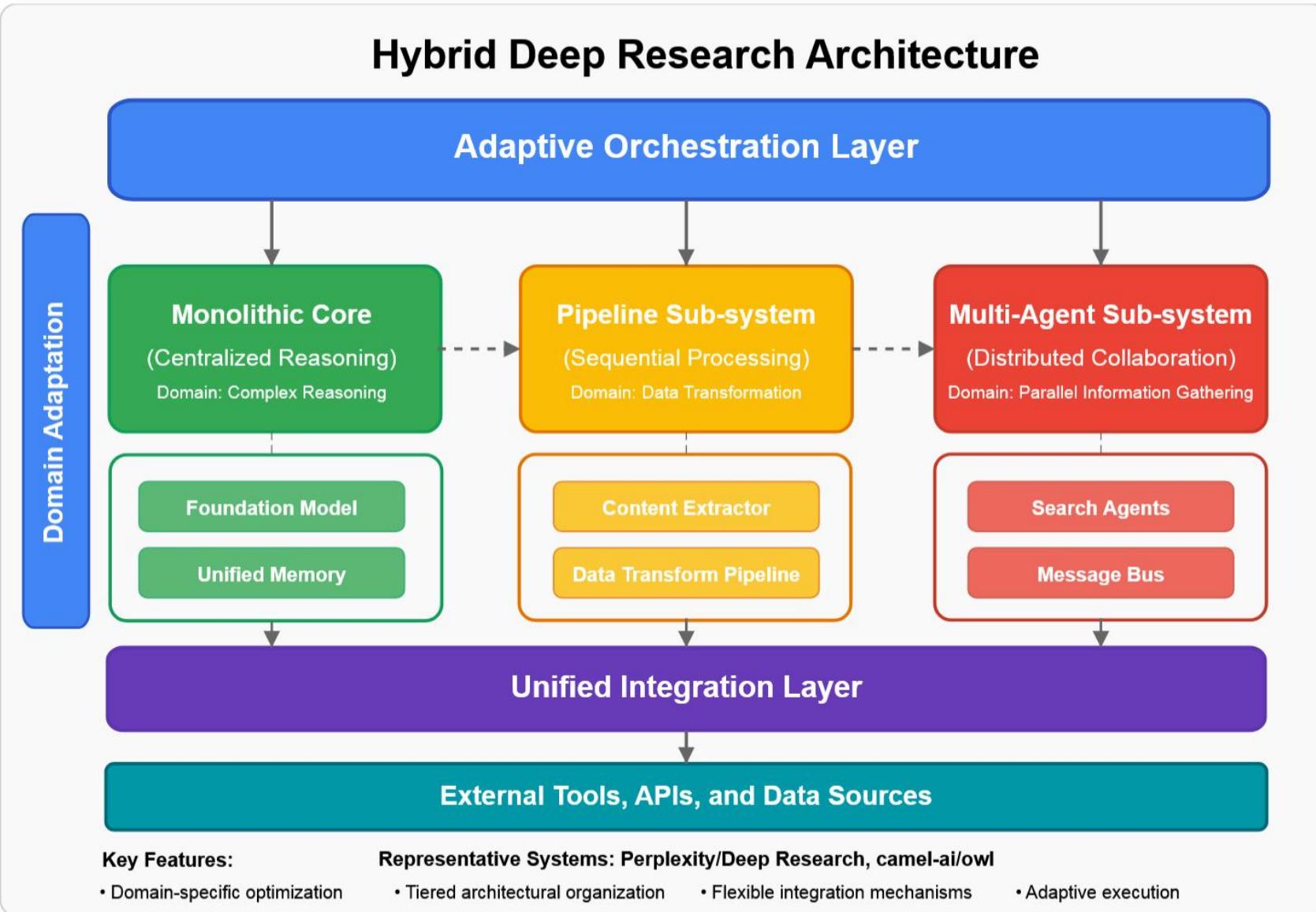


Figure 1: A structural overview of a DR agent in a multi-agent architecture for ease of illustration.

1. 用户输入与初始规划 用户提交需求（如 “生成一份关于 LLM 的报告，包含对比表和流程图”），大语言模型（LLM）首先对需求进行解析和规划，确定任务目标与执行路径。
2. 意图澄清若需求存在模糊点，系统会主动与用户交互，进一步明确意图，确保任务方向精准。
3. 多智能体迭代工具调用 系统启动多智能体协作，通过多轮步骤完成任务：
  - 离线检索：从向量数据库、关系型数据库中调取本地已有数据。
  - 在线检索：通过 API、浏览器获取实时网络信息。
  - 扩展工具使用：调用数据分析、代码执行、多模态生成等工具，处理复杂任务。
4. 输出 最终成果所有信息与处理结果整合后，生成包含引言、对比表、流程图、结论等完整结构的最终报告，交付给用户。

## 02. 混合架构



融合多种模式优势，按领域或任务动态调整处理方式，兼顾灵活性与性能，代表系统有 Perplexity/DeepResearch、Camel-AI/OWL。

Fig. 7. Hybrid Deep Research Architecture

未来改进方向

03

### 03. 未来改进方向

主分类	子方法/技术	具体技术方案
高级推理架构：突破现有推理能力边界	上下文窗口优化与管理	<b>信息压缩与优先级排序：</b> 开发“语义保留压缩”技术；通过分层 summarization 技术，维持多级别信息表征。 <b>外部记忆架构：</b> 构建“工作记忆 + 长期知识仓库”的分层外部记忆库，采用可微分检索机制无缝集成外部知识。
	混合符号 - 神经推理融合	神经 - 符号整合：将符号逻辑（规则引擎、知识图谱）与神经模型结合，增设逻辑验证层验证神经推理结论。 高级知识图谱整合：构建动态、上下文感知的知识图谱。
	因果推理增强	因果推断机制：开发领域专用因果关系图构建技术。 干预建模技术：构建反事实生成工具，设计领域专用干预模板。
	不确定性表征与推理	多维度不确定性建模：区分认知不确定性（知识缺口）、随机不确定性（数据噪声）、模型不确定性（表征局限），将贝叶斯概率框架融入 LLM 推理。
多模态 Deep Research：整合全维度信息	视觉信息整合	科学图像专用解析：开发领域定制化图像识别模型。 视觉证据整合：建立“文本 - 视觉”证据对齐技术，匹配同一问题的跨模态信息。
	多模态源分析与处理	视频内容处理和音频内容整合。
	跨模态推理技术	<ul style="list-style-type: none"><li>• 多模态思维链推理：构建“文本分析→视觉处理→数值计算→空间推理”的跨模态推理流程，跨模态一致性验证，多模态解释生成。</li><li>• 多模态与跨模态推理：扩展至音频（如访谈录音转录）、视频（如实验演示解析）、3D 模型（如工程设计图），实现“文本 + 视觉 + 结构化数据”的统一理解；</li></ul>
领域专用优化：提升垂直场景适配性	科学、法律与监管、医疗与健康等领域定制化适配	领域模型微调，科研工作流集成

### 03. 未来改进方向

主分类	子方法/技术	核心原理与机制
解决“信息获取受限”的核心路径	拓宽信息来源	通过模型上下文协议（MCP）集成更精细、更广泛的模块化工具，让DR智能体能够动态访问专有API、私有数据库、付费应用等传统方式无法触及的资源，突破信息获取的边界
	强化事实核查能力	<p>构建“结构化验证+自我反思”的双重核查机制，结合强化学习优化。</p> <ul style="list-style-type: none"><li><b>结构化交叉验证：</b>智能体生成初步答案后，不直接输出，而是主动检索多个独立来源，核查事实一致性、评估来源可信度；</li><li><b>自我反思机制：</b>智能体在推理过程中，主动检查中间结果的合理性，类似人类研究者的反思过程；</li><li><b>强化学习激励：</b>在RL中加入“正确性导向”奖励，若检测到矛盾或不确定性，自动重新规划检索策略，必要时回溯修正前期推理（如OpenAI DR的回溯机制）。</li></ul>
执行效率与鲁棒性不足	异步并行执行	<ul style="list-style-type: none"><li>基于有向无环图（DAG）的异步并行架构：通过DAG等高级任务建模结构，实现子任务的并行执行和动态优先级排序；</li><li>强化学习驱动的调度智能体：训练专门的调度智能体，基于运行时性能信号（如执行延迟），动态分配子任务、调整执行顺序。</li></ul>
	工具集成推理	<ul style="list-style-type: none"><li>超越简单工具调用：要求智能体不仅能按逻辑顺序调用合适的工具，还能根据中间结果自适应调整推理路径；</li><li>强化学习优化：设计精细化的奖励结构，不仅评估最终答案的正确性，还评估工具选择的合理性、参数设置的准确性和推理效率；</li><li>实证效果：经TIR优化的智能体，在多个基准测试中性能提升15-17%，对未见过的工具和任务具有更强的泛化能力，工具调用模式更合理，能更好平衡工具利用与自身知识</li></ul>
	多智能体架构的参数优化	<ul style="list-style-type: none"><li>分层强化学习（HRL）与任务定制化后训练：引入分层内部奖励机制，促进智能体间的协作学习和反馈传播；设计多阶段的后训练优化流程，迭代提升智能体间的交互效果；</li><li>RL驱动的专用调度智能体：训练专门的调度智能体，基于实时性能指标动态分配子任务、调整执行顺序，将调度决策建模为RL动作，学习平衡并行执行、资源利用率和任务优先级的自适应策略。</li></ul>

### 03. 未来改进方向

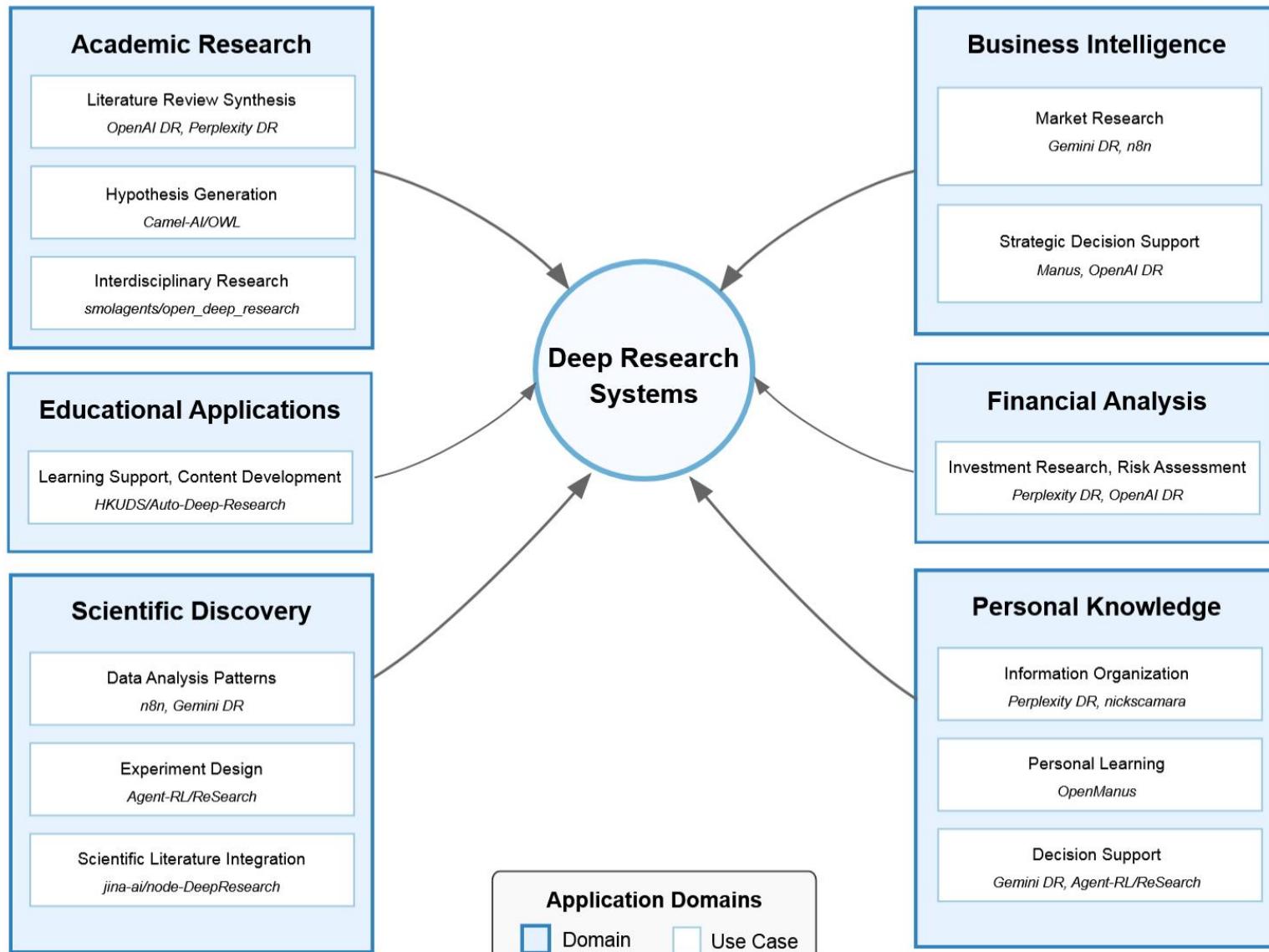
人机协作与标准化：构建高效协同生态	交互式研究工作流设计	自适应查询优化：开发意图澄清系统，识别研究问题的模糊点与替代方向；设计范围调整界面，基于初步结果动态扩展或缩小研究焦点；提供视角多样化工具，推荐相关替代观点。 交互式探索界面：开发研究导航专用信息可视化技术；实现自适应细节管理，根据用户兴趣扩展或折叠内容；设计“合成结果 - 原始来源”的无缝切换机制。
	用户专长适配机制	专长动态推断：通过用户交互行为（提问方式、反馈内容）评估领域知识水平，自适应调整解释深度与术语使用。 互补能力协同设计：明确人机分工边界，AI 聚焦信息处理，人类主导主观判断；设计置信度信号传递机制，高亮需人类评估的关键结论；搭建交互式批判框架，支持高效的人类反馈输入。
	框架与接口标准化	组件接口统一（智能体通信协议）：基于 MCP（Model Context Protocol）、A2A（Agent2Agent Protocol）等协议，定义检索、规划、生成等核心组件的交互标准；建立跨系统组件复用机制。 评估指标标准化：开发专用基准套件，针对特定研究能力设计测试任务；建立跨领域、多维度的评估框架，超越单一准确率指标。
	跨平台研究协议与协同创造	研究结果交换格式标准化：定义包含归因、置信度元数据的研究发现结构；建立证据表征与研究目标的共享 schema，支持分布式处理。 分布式研究协调：设计分布式搜索协调协议，高效调度专用搜索能力；开发跨系统结果验证技术，保障分布式研究的一致性。 联合知识创造：构建协同文档开发环境，支持人机内容无缝衔接；设计混合主动研究规划框架，动态平衡人类偏好与 AI 识别的研究机会。

应用场景

04

# 04. 应用场景

## Deep Research Application Domains and Use Cases



# deeppresearch可行点

·近期优先上下文保留：仅保留最新 K 条工具响应，但完整保留“思考 - 行动”轨迹。

从“文本域”扩展到“视觉 - 语言多模态域”

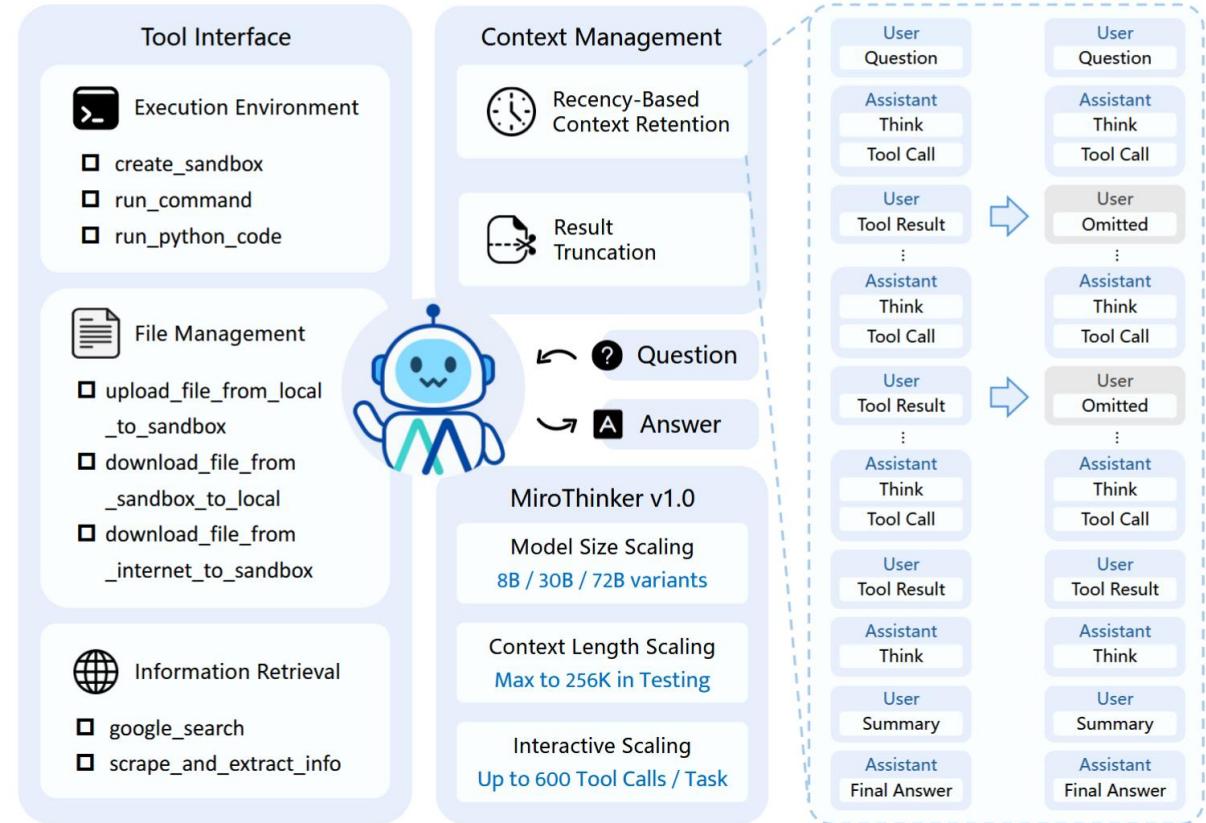
多智能体协作：从“单智能体全流程处理”进化为“专用智能体分工”，如“搜索 Agent + 分析 Agent + 验证 Agent”协同，通过标准化协议（如 A2A）实现信息共享与冲突解决

如何触发交叉验证和自我反思？

如何进行交叉验证和自我反思？

- **结构化交叉验证：**智能体生成初步答案后，不直接输出，而是主动检索多个独立来源，核查事实一致性、评估来源可信度。

- **自我反思机制：**智能体在推理过程中，主动检查中间结果的合理性，类似人类研究者的反思过程；



**Figure 2:** Overview of the MiroThinker v1.0 agent architecture. The framework integrates a structured tool interface, *i.e.*, execution environment, file management, and information retrieval, with a simple recency-aware context management to support interactive scaling. On the right, an agentic trajectory example illustrates the recency-based context retention mechanism, where tool outputs from earlier turns are omitted to maintain context efficiency.