技术发展全景报告 · 2026

大语言模型应用发展历程

从 Token 补全 → 对话助手 → Tool-Use Agent → Multi-Agent → 自主 Agent

追踪 LLM 从"预测下一个词"到能够独立规划、使用工具、自主完成复杂任务的完整技术演进脉络，剖析每一代架构的核心突破与里程碑产品。

-- by Winfred Chen @202602

Era 1
Token 补全

Era 2
对话助手

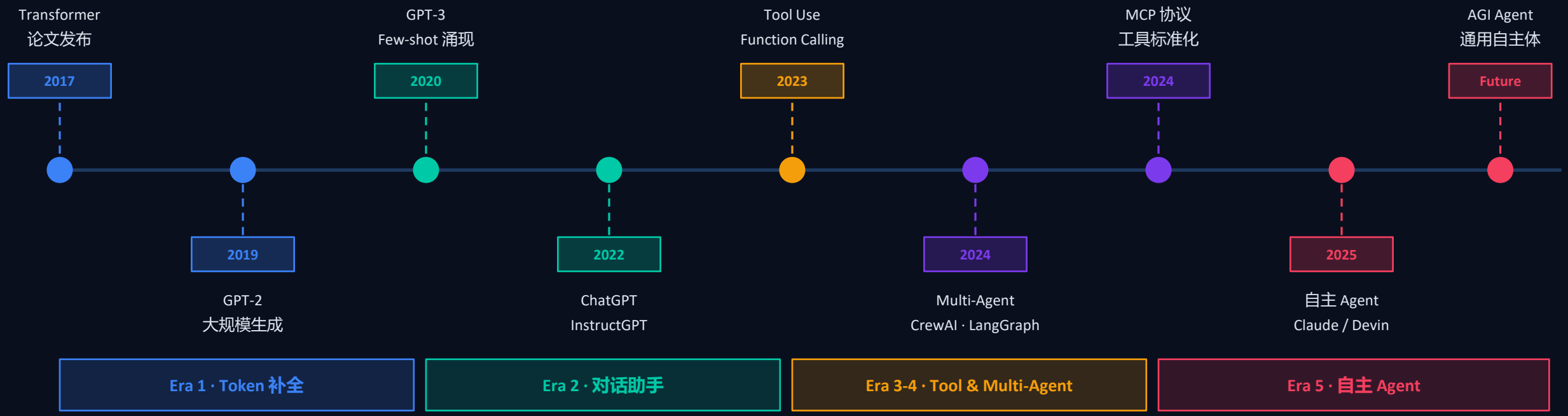
Era 3
Tool-Use Agent

Era 4
Multi-Agent

Era 5
自主 Agent

技术演进总览时间线

2017 — 2025+



每个 Era 代表一次架构范式跃迁，能力边界随之指数级扩展

Era 1

Token 补全时代

2017 — 2020

核心理念

预测序列中的下一个 Token，
无监督自回归训练，
规模即能力。

里程碑产品

2017 Attention Is All You Need（Transformer）

2018 GPT-1：首个大规模预训练语言模型

2019 GPT-2：惊艳的零样本生成能力

2020 GPT-3：175B 参数，少样本涌现

技术原理

Transformer 自注意力机制，并行建模长距离依赖

自回归语言建模（CLM），预测 P(token_t | token_1…t-1)

Byte-Pair Encoding（BPE）分词，词表约 50K

规模定律（Scaling Law）：参数 × 数据 × 算力 → 能力

无监督预训练 + 有监督微调（Fine-tuning）范式

能力边界

[✓] 流畅文本生成，代码补全，摘要翻译

[✓] Few-shot 学习（GPT-3 涌现）

[X] 无法执行动作，只能输出文字

[X] 无记忆，单次上下文窗口约 2K-4K token

[X] 事实幻觉严重，缺乏对齐机制

>> 核心洞察："规模即能力"——这一时代的 LLM 本质是一个极其复杂的自动完成器，不具备任何主动性。

核心概念：Transformer · 规模定律 · 自回归建模

自注意力机制（Self-Attention）

Transformer 的核心创新。序列中每个 Token 同时与所有其他 Token 计算关联权重，权重由 Query × Key 点积得出，再对 Value 加权求和：

$$\text{Attention}(Q,K,V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

相比 RNN 逐步传递，自注意力一次并行捕获任意距离的语义依赖，彻底解决了长程依赖消失问题。

自回归语言建模（Autoregressive LM）

GPT 系列的训练目标：给定前 t 个 Token，预测第 t+1 个 Token 的概率分布，最大化语料似然：

$$L = -\Sigma \log P(\text{token}_t \mid \text{token}_{1 \ldots t-1}; \theta)$$

看似简单，但随着规模增大，翻译、推理、编程等复杂能力自发"涌现"——这是规模定律的核心发现。

规模定律（Scaling Law）

Kaplan et al. 2020：模型能力与参数量 N、数据量 D、算力 C 成幂律关系，三者需协同增长。

Chinchilla 定律（2022）修正：最优训练满足 D ≈ 20 × N——LLaMA 系列据此以更少参数达到更强效果。

BPE 分词（Byte-Pair Encoding）

LLM 不处理字符，而是将文本分割为子词 Token。BPE 通过贪心合并高频字符对构建词表（约 50K-100K）。

英文约 1-2 token / 词；中文约 1-2 token / 汉字。上下文窗口大小与 API 计费均以 Token 为单位。

多头注意力 & 位置编码

多头注意力（Multi-Head）：并行运行 8-32 个独立注意力头，每个头学习不同语义维度（句法、指代、相似性），输出拼接后线性变换，大幅提升表达能力。

位置编码（Positional Encoding）：为每个 Token 注入位置信息，使模型能区分词序差异。

Era 1 架构关系总结

Era 2

对话助手时代

2021 — 2022

核心理念

通过 RLHF 对齐人类意图，
从"补全"转向"理解指令"，
ChatGPT 引爆全民 AI 时代。

里程碑产品

2021 InstructGPT：RLHF 对齐突破

2022 ChatGPT：1亿用户两个月

2023 GPT-4：多模态，律考98%

2023 Claude 1/2：宪法AI，长上下文

核心技术突破

RLHF（人类反馈强化学习）：SFT → Reward Model → PPO

InstructGPT 范式：无需改架构，仅对齐训练

System Prompt：角色设定与行为约束成为产品设计工具

上下文窗口扩展：4K → 8K → 32K → 128K token

宪法 AI（Constitutional AI）：Claude 的价值观锚定

产品形态与局限

[✓] 多轮对话，指令遵循大幅提升

[✓] 写作、翻译、编程辅助已接近专业水平

[✓] 基础问答达到知识工作者替代门槛

[X] 仍为被动响应，无法主动触发行动

[X] 知识截止日期，无法访问实时信息

>> 核心洞察：RLHF 将 LLM 从"语言预测器"变为"听得懂人话的助手"，ChatGPT 是第一个大众化 AI 产品。

核心概念：RLHF · 宪法 AI · 指令微调 · 上下文窗口

RLHF 三阶段流程（InstructGPT 范式）

阶段一 · SFT 有监督微调

收集（指令→期望回答）配对数据，人工标注高质量示例，对预训练模型做初步微调——"教会模型按指令写作"。

→

阶段二 · 训练奖励模型 (RM)

对同一 Prompt 生成多个候选回答，人工进行偏好排序 (A>B>C)。用排序数据训练奖励模型，学会为回答打分——"量化人类偏好"。

→

阶段三 · PPO 强化学习

以奖励模型打分为信号，用 PPO 算法对 SFT 模型迭代优化，最大化期望奖励；KL 散度惩罚防止模型偏离预训练分布过远。

宪法 AI（Constitutional AI，CAI）

Anthropic 用于训练 Claude 的对齐方法：用一套明确书面原则（"宪法"）代替人工偏好标注。

① 让模型生成潜在有害回答 → ② 按宪法原则自我批判并修改 → ③ 用 AI 反馈（RLAIF）替代人工排序。

效果：对齐成本更低、价值观更可解释，同时保持有益性。

上下文窗口演进

GPT-3 (2020)	4K	~3,000词，无法处理长文档
GPT-4 Turbo	128K	~10万词，完整书籍章节
Claude 3	200K	~15万词，400+页PDF
Gemini 1.5 Pro	1M	~75万词，完整代码库

指令微调（SFT）与 RLHF 的关系

SFT（监督微调）：给出正确示例让模型模仿，解决"会不会"的问题。

RLHF（强化学习+人类反馈）：通过偏好排序给模型反馈，解决"好不好"的问题——哪种回答更有帮助、更安全、更诚实。

两者结合是现代对话助手的标准训练流程：预训练 → SFT → 奖励模型训练 → PPO 微调。

Era 3

Tool-Use Agent 时代

2023

核心理念

LLM 获得"手"：

通过 Function Calling

调用工具，

突破纯文字边界。

里程碑

2023.3 OpenAI 发布 Function Calling

2023.4 AutoGPT 走红，Agent 概念爆炸

2023.6 ReAct 论文被工程化落地

2023.8 LangChain Agent 生态成熟

2023.10 OpenAI GPT-4 Turbo + Plugins

ReAct 核心循环：Think → Act → Observe

Thought 思考
CoT 推理
决定下一步

Action 行动
调用工具
(搜索/代码/API)

Observe 观察
读取工具
返回结果

Answer 输出
生成最终
回复

↺ *直到任务完成或达到最大循环次数*

典型工具类型与应用场景

搜索工具 Tavily、Bing API · 实时信息获取

API 调用 Slack、GitHub、日历 · 系统集成

代码执行 Python Sandbox、E2B · 数据分析、计算

浏览器 Playwright、Selenium · Web 自动化

数据库 SQL、向量检索 · 知识访问

文件操作 PDF、Office、CSV · 文档处理

>> 核心洞察：Function Calling 赋予 LLM "手脚"，ReAct 给了它"思维回路"——Agent 时代正式开启。

核心概念：Function Calling · ReAct · CoT · RAG

Function Calling（函数调用）

开发者在 API 请求中声明工具的 JSON Schema（名称、描述、参数类型）。LLM 根据上下文决定是否调用，若需要则在回复中输出结构化 JSON 调用意图，由客户端执行并将结果回传。

示例：用户问"今天北京天气？" → LLM 输出 {"tool":"get_weather","args":{"city":"北京"}} → 客户端调用 API 返回数据 → LLM 生成最终回答。

MCP（Model Context Protocol）是其标准化进化：一套协议让同一工具服务器对接所有

思维链推理（CoT / ToT）

CoT（Chain-of-Thought）：Prompt 中加入逐步推理示例，或直接用 "让我们一步步思考" 触发内部推理，使 LLM 在给出答案前生成可见的推导步骤，大幅提升数学、逻辑类任务准确率。

ToT（Tree-of-Thought）：将 CoT 从线性扩展为树状，LLM 并行探索多条推理路径并回溯，适合需要搜索与规划的复杂问题（如解谜、策略规划）。

两者都是提示工程而非架构改动，无需重新训练。

检索增强生成（RAG）——解决知识截止与幻觉问题

① 离线索引

文档→切分 Chunk→Embedding 向量→存入向量数据库

→

② 在线检索

用户查询→向量化→余弦相似度→取 Top-K 相关片段

→

③ 注入生成

检索片段拼入 Prompt→LLM 基于参考内容生成有据可查的回答

工具使用的本质是"感知-行动循环"的实现：LLM 从自然语言生成器升级为可以感知外部环境（观察工具结果）并采取行动（调用工具）的智能体。
RAG 解决知识边界，Function Calling 解决动作边界，CoT 解决推理深度——三者共同构成 Era 3 的核心技术栈。

Era 4

Multi-Agent 协作时代

2024

核心理念

分而治之：

Orchestrator 分发任务

给专业 Sub-Agent，

并行协作突破单体上限。

代表框架

LangGraph — 状态图流程编排

CrewAI — 角色驱动协作

AutoGen — 多轮对话代理

Dify / n8n — 低代码部署

Google A2A — 跨 Agent 协议

Multi-Agent 拓扑结构

2024 关键里程碑：MCP 协议 · 工具生态标准化

- Anthropic MCP（Model Context Protocol）：Agent ↔ 工具服务器标准化接口，"插件一次开发，处处可用"
- Google A2A 协议：Agent 之间跨系统通信标准，2024 年多家厂商联合推进
- OpenAI Swarm / Assistants API：首个官方 Multi-Agent 编排框架，Thread + Run + Tool 架构
- 企业落地：Salesforce、ServiceNow、SAP 均宣布 Agent 战略，数字员工概念正式成型

>> 核心洞察：单体 Agent 遇到任务复杂度天花板，专业化分工 + 协调机制是提升上限的关键架构范式。

核心概念：DAG 编排 · MCP 协议 · Reflexion · 通信模式

有向图任务编排（DAG / LangGraph）

Multi-Agent 用有向图（支持循环）作为流程模型：

节点（Node）：LLM 调用、工具执行、条件判断或子 Agent
边（Edge）：固定顺序或条件跳转（基于节点输出决定下一步）
状态（State）：节点间传递的共享数据结构（对话历史、中间结果）
循环边：允许节点在条件满足前反复执行（如重试失败的工具调用）

MCP 协议架构（Model Context Protocol）

Anthropic 2024 年发布的工具标准化协议，目标是成为 Agent 生态的"USB 接口"：

MCP Host：发起连接的 AI 应用（如 Claude Desktop、Cursor）
MCP Client：Host 内部的连接管理组件，每个 Server 一个实例
MCP Server：提供工具能力的轻量服务（本地 stdio 或远程 SSE/HTTP）

三类原语：Tools（可执行动作）· Resources（只读数据）· Prompts（模板）

五种 Multi-Agent 通信与编排模式

顺序链
A完成→传B→传C，串行执行

并行扇出
同时启动N个Sub-Agent，汇总结果

层级委派
Orchestrator递归分发给子Orchestrator

黑板系统
共享中央状态，各Agent读写协调

消息总线
异步队列通信，解耦生产与消费

Reflexion（反思机制）——Agent 的自我改进

Shinn et al. 2023：Agent 执行失败后不仅记录轨迹，还生成"语言反思"描述失败原因和改进策略，将反思文本存入长期记忆，下次执行前注入 System Prompt。
短期记忆：当前任务执行轨迹（所有 Thought-Action-Observation 步骤）
长期记忆：历史任务反思摘要（"上次因未验证网址格式而失败，下次先用正则校验"）
实测效果：在 HotpotQA 等基准上相比直接 ReAct 提升 10-20% 成功率。

Era 5

自主 Agent 时代

2024 — 2025+

核心理念

无需人工逐步干预，
Agent 自主规划、执行、
纠错、完成完整工作流，
"数字员工"成为现实。

代表产品

Claude 系列 — Computer Use / Code / 现象级自主体
OpenClaw — 开源，风格自主 Agent 生态
Devin — 首个 AI 软件工程师
OpenAI Operator — 浏览器自动化
Manus — 通用自主体（国产代表）

自主 Agent 的核心能力矩阵

长程规划

将复杂目标分解为 DAG 子任务，动态调整执行路径

Computer Use

直接操控桌面/浏览器 UI，不依赖 API 接口

自我纠错

执行失败后 Reflexion 反思，重新规划下一步

持久记忆

跨会话的长期记忆与知识积累，越用越聪明

安全护栏

权限最小化、沙箱隔离、Human-in-the-loop 确认

异步执行

后台长时运行任务，返回结果推送用户

Claude — 自主 Agent 旗舰案例

Claude Computer Use	2024.10 发布，截图→分析→点击→输入，操控整个 OS 界面，完成跨应用复杂工作流
Claude Code	2025 发布，命令行 AI 编程 Agent，自主读写代码库、运行测试、提交 PR，成为开发者标配工具。
OpenClaw 现象	2026年初, 以 OpenClaw 为代表的 Claw 系列掀起自主 Agent 浪潮,中国大陆云服务厂商纷纷上线 OpenClaw相关部署服务。 MCP, A2A, Skill生态三线并进，成为企业级 Agent 中互操作性的事实标准。
Projects & Memory	项目级长期记忆，跨会话保留用户偏好、工作背景，实现个性化自主助手

>> 核心洞察：自主 Agent 不只是"更强的助手"，而是第一次让 AI 成为能独立承担工作流的"数字同事"。

核心概念：Computer Use · Extended Thinking · 长期记忆

Computer Use（计算机使用能力）

Claude 2024.10 发布的实验性能力：无需 API，像人类一样操控计算机 GUI。

工作循环：

① 截取屏幕截图（视觉 Token 输入）

② Claude 分析界面，理解当前应用状态

③ 输出动作指令：{ "type": "click", "coordinate": [x, y] } 或 { "type": "type", "text": "..." }

④ 客户端执行动作，再次截图 → 回到 ①

意义：打破 Agent 只能通过 API 交互的限制，任何有 GUI 的软件均可被 Agent 操控。

扩展思维（Extended Thinking）

Claude 3.7 Sonnet 引入：生成最终答案前，模型在 <thinking> 标签内进行大量内部 CoT 推理（对用户可见但不计入输出限制）。

工作原理：分配更多计算预算用于深度推理→ 探索多条解题路径 → 生成简洁最终答案。

与 o1/o3 的关系：同源技术——将"让我们一步步思考"的提示工程能力内化为模型原生能力，推理深度由计算预算决定，而非 Prompt 设计技巧。

长期记忆实现策略——从"单次会话"到"数字同事"

向量记忆库

对话片段 Embedding 化存储，按语义相似度检索，适合非结构化知识

结构化摘要

将长对话压缩为 JSON/Markdown，存储关键事实、决策、偏好，适合精确查询

实体记忆

维护"实体字典"，记录人物、项目、概念及其属性，支持快速精确定位

记忆写入策略

判断信息的重要性、时效性、隐私敏感性，决定是否写入，防止记忆库质量下降

Human-in-the-Loop（人机协作节点）

自主性越高，风险控制越关键。对删除数据、财务交易、代码部署等高风险操作，强制设计人工确认节点（Human Approval Gate）。

这不是降低自主性，而是产品成熟度的体现：Agent 应知道"何时停下来问人"，而不是盲目执行到底。

核心概念：Skill 技能系统 · 工具即技能 · 技能复用与组合

什么是 Agent Skill（技能）

Skill 是 Agent 的"可复用能力单元"——将一项具体能力（搜索、写代码、分析数据）封装为标准化模块，可以被调用、组合、共享。

与工具（Tool）的区别：

- Tool：单一 API 调用，如 search(query)，无状态
- Skill：多步工作流 + 状态管理 + 错误处理，如"竞品分析 Skill"包含搜索→整理→对比→输出报告的完整流程

Skill Library（技能库）

Agent 将成功执行过的任务模式固化为可复用技能，存入 Skill Library，后续遇到相似任务直接调用，无需重新规划。

三类技能来源：

- 人工预定义：开发者手写的固定流程（如"发送周报 Skill"）
- 经验积累：Agent 从 Reflexion 反思中提炼的成功模式
- 社区共享：MCP Server 中开放的第三方技能包

Skill 分层架构——从原子能力到复合智能

L1 原子技能 *单步工具调用*	L2 复合技能 *多步工作流*	L3 领域技能 *业务场景封装*	L4 Agent 角色 *完整职能定义*
web_search(query)	研究报告 Skill	销售邮件撰写	Researcher Agent
execute_code(script)	代码审查 Skill	财报分析	Coder Agent
read_file(path)	数据分析 Skill	需求文档生成	PM Agent
api_call(endpoint)	竞品对比 Skill	客服工单处理	Analyst Agent

技能复用与迁移

跨任务复用：同一 Skill 被不同 Agent / 用户复用，通过参数化适配不同场景（如"报告撰写 Skill"可生成市场报告、技术文档、周报）。

迁移学习：在新领域任务中，Agent 优先检索 Skill Library 中的相似技能并微调，而非从零规划——降低 Token 消耗，提升稳定性。

Claude / OpenClaw 中的 Skill 实践

Claude Code Skill	自动识别编程任务类型（调试/重构/新建），调用对应 Skill 工作流
MCP Tool-as-Skill	MCP Server 中的 Tool 可直接作为 Skill 原子单元被 Agent 编排复用
Projects 技能沉淀	Claude Projects 将用户常用指令模式固化为项目级 Skill，跨会话一键复用

五代架构能力对比

维度	Era 1 Token 补全	Era 2 对话助手	Era 3 Tool Agent	Era 4 Multi-Agent	Era 5 自主 Agent
主动性	被动	被动	有限主动	协作主动	完全自主
工具使用	无	无	Function Call	MCP+多工具	任意工具+UI
规划能力	无	CoT初步	ReAct循环	DAG子任务	长程自主规划
记忆系统	单次上下文	多轮对话	工具结果	共享状态	持久跨会话
执行环境	纯文本	纯文本	API调用	多Agent并行	OS/浏览器
自我纠错	无	有限	部分	Critic审查	Reflexion全程
代表模型	GPT-3	ChatGPT	GPT-4+FC	CrewAI多模型	Claude/OpenClaw/Devin

能力增长轨迹

下一步：通往通用自主体的关键挑战

安全与可信

- Prompt Injection 与工具链攻击防护
- 权限最小化与沙箱隔离标准化
- 行为可审计、可解释、可回滚
- 红队测试与对抗性评估体系

长程规划精度

- 超长任务的错误累积与纠偏机制
- 动态环境下的计划重新调度
- 不确定性量化与风险感知
- 人机协作确认节点的优化设计

记忆与知识管理

- 个性化长期记忆的隐私保护
- 知识更新与遗忘的平衡策略
- 跨 Agent 知识共享协议
- 结构化知识 vs 向量检索的融合

生态与标准

- MCP / A2A 协议和 Skill 的普及与治理
- Agent 能力评测基准（GAIA/SWE-bench）
- 合规框架详见专页
- 开源 vs 闭源 Agent 能力差距

AGI Agent 地平线：当规划精度、记忆深度、工具广度、安全可信同时达到阈值，通用自主体将成为现实。

企业级 Agent 合规与监管框架

Enterprise AI Governance

数据与隐私合规

- GDPR / PIPL 个人数据处理合规
- Agent 调用工具时的数据最小化原则
- RAG 知识库的数据来源版权审查
- 用户对话数据的存储与删除权
- 跨境数据传输限制（尤其 CN/EU/US的对外传输）

AI 系统透明度与可解释性

- EU AI Act：高风险 AI 系统需提供决策说明
- Agent 执行链路的全程可审计（Trace 留存）
- 向用户明确披露：当前交互由 AI Agent 处理
- 模型偏见检测与公平性评估义务
- 可解释 AI（XAI）输出要求

操作安全与风险管理

- Agent 权限最小化：只授予任务必需权限
- 高风险操作强制 Human-in-the-Loop 审批
- 沙箱隔离防止横向扩散与供应链攻击
- 红队测试（Red-Teaming）定期安全评估
- AI 事故响应计划与回滚机制

全球主要监管框架

EU AI Act (2024)	按风险等级分类管控，高风险 AI 须合规审查、登记注册
中国 AIGC 管理办法	生成式 AI 服务需备案，内容安全审核，数据本地化
美国 EO 14110	联邦机构 AI 使用指引，安全评估要求，透明度标准
ISO/IEC 42001	AI 管理体系国际标准，企业 AI 治理的认证框架

企业部署上线检查清单

- [✓] 完成 AI 风险评估（Risk Assessment）
- [✓] 确认数据处理合法依据（同意/合同/合法利益）
- [✓] 部署可观测性基础设施（Trace / Log / Alert）
- [✓] 制定 AI 事故响应预案（Incident Response Plan）
- [✓] 用户知情告知与隐私政策更新

合规不是终点，而是持续过程——随着 AI 能力演进，监管框架也在动态更新，企业需建立常态化 AI 治理机制。

关键结论

01	范式跃迁而非渐进演化：每个 Era 都是架构层面的质变——从补全到对齐，从单轮到多轮，从文字到行动，从单体到群体，从受控到自主。
02	工具是边界，记忆是深度，规划是高度：Agent 的能力上限由这三个维度共同决定，缺一不可。
03	MCP 和 A2A 是基础设施革命：就像 TCP/IP 统一了网络，标准化工具协议将统一 Agent 生态，降低集成成本。
04	安全与自主性并行演进：更高的自主性必须配套更强的护栏——这不是矛盾，而是产品成熟度的双轮驱动。
05	机遇窗口：数字化转型进入 Agent 阶段，懂得设计和应用 Agent 的专业人员将成为稀缺价值。