

AI 编程与智能体开发

林子雨 副教授 厦门大学
E-mail: ziyulin@xmu.edu.cn
2026年9月



主讲教师



- ◆ 2013年度厦门大学奖教金获得者
- ◆ 2017年度厦门大学奖教金获得者
- ◆ 2020年度厦门大学奖教金获得者
- ◆ 2023年度厦门大学奖教金获得者
- ◆ 2026年度厦门大学奖教金获得者
- ◆ 2018年国家精品在线开放课程
- ◆ 2020年国家级线上一流本科课程
- ◆ 2023年国家级线上一流本科课程
- ◆ 2020年福建省线上一流本科课程
- ◆ 2021年福建省线上线下混合一流本科课程
- ◆ 2018年福建省高等教育教学成果奖二等奖
- ◆ 2022年福建省高等教育教学成果奖特等奖
- ◆ 2024年福建省高等教育教学成果奖特等奖
- ◆ 2023年教育部国家智慧教育公共服务平台应用典型案例





配套教材



《AI编程与智能体开发》

林子雨 编著

人民邮电出版社 预计2026年11月上市销售

教材官网提供详细信息和样书申请

官网: <https://dbllab.xmu.edu.cn/post/ai-programming-agent/>

教材目录

- 第1章 大模型
- 第2章 AI编程概述
- 第3章 AI编程环境搭建
- 第4章 基于提示词的AI编程
- 第5章 基于规范的AI编程
- 第6章 流程驱动的AI编程
- 第7章 智能体概述
- 第8章 智能体开发框架LangChain
- 第9章 智能体开发框架LangGraph

教材特色

- 教材内容来自互联网大厂真实实践
- 以实践教学为依托
- 理论实践贯通
- 覆盖主流技术



扫码访问教材官网
了解详情、获取资源、申请样书

第7章

智能体概述



目录

01 从大语言模型到智能体

03 智能体的核心组成

05 常见的智能体架构

07 智能体工程化要点

02 智能体的定义与特征

04 智能体的工作流程

06 智能体的能力边界



Part 01

从大语言模型到智能体



7.1 从大语言模型到智能体



什么是
大语言模型

大语言模型的
优势

大语言模型的
局限

智能体解决
什么问题

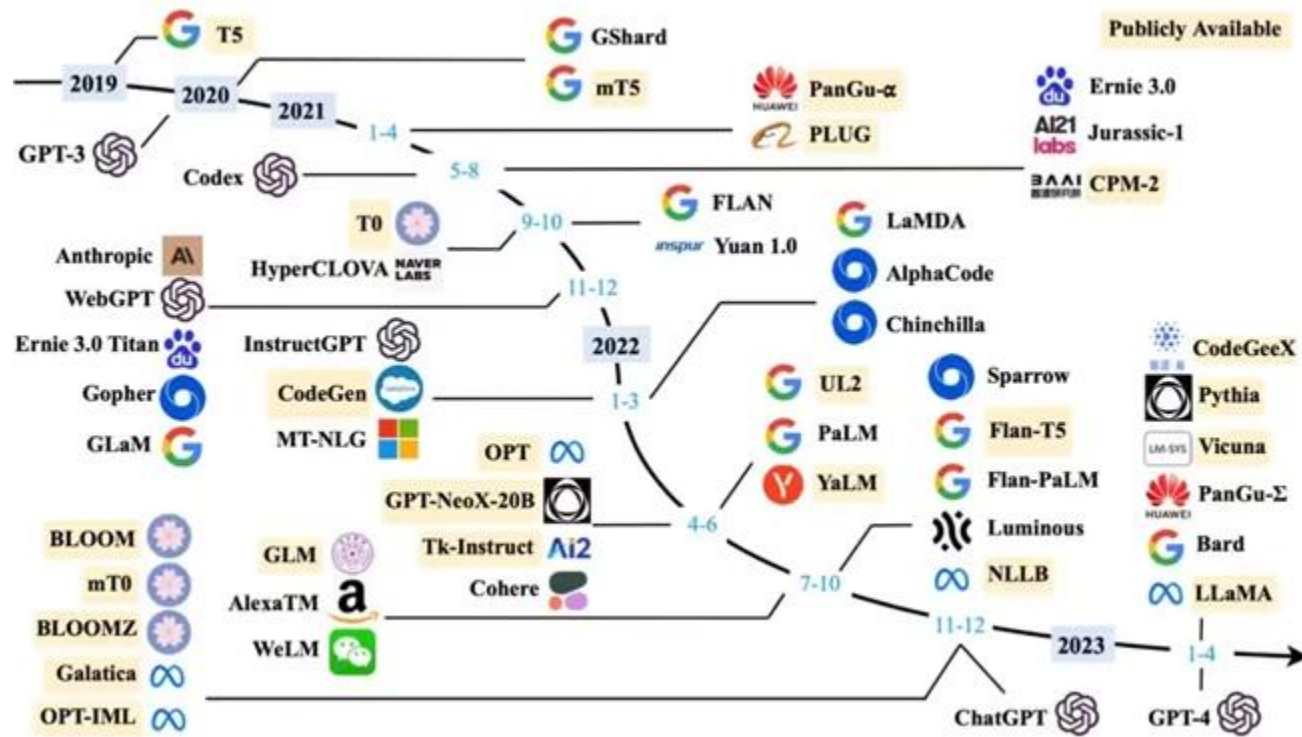


7.1.1 什么是大语言模型

■ 定义与范畴

大语言模型（LLM）是依托海量文本数据训练的 AI 模型，为自然语言处理核心技术、当下 AI 应用基础载体，代表产品有 DeepSeek、豆包、Kimi、千问等；

“大模型”为统称，包含语言大模型、视觉大模型、多模态大模型，日常语境下提及“大模型”通常指代语言大模型（或大语言模型）。





7.1.1 什么是大语言模型

■ 底层架构与训练方式

以深度学习神经网络为架构，依靠数十亿至上万亿级庞大参数实现语言理解、生成。

训练素材包含书籍、网页文章、对话记录、专业文献等海量全网文本，通过算法学习人类语言规律、语法逻辑、常识知识、行文风格与思维表达习惯。





7.1.1 什么是大语言模型

■ 核心功能与内容生成特点

可识别文字语义，支持问答对话、文案创作、总结摘要、翻译改写、代码编写、思路梳理、知识解析、方案构思等工作。

不机械检索答案，依托习得的语言逻辑，生成贴合语境、符合人类认知的全新连贯内容。





7.1.1 什么是大语言模型

■ 交互特性与落地应用场景

拥有上下文记忆能力，支持多轮连续对话。

已广泛应用在智能客服、办公辅助、教育学习、内容创作、软件开发等领域。





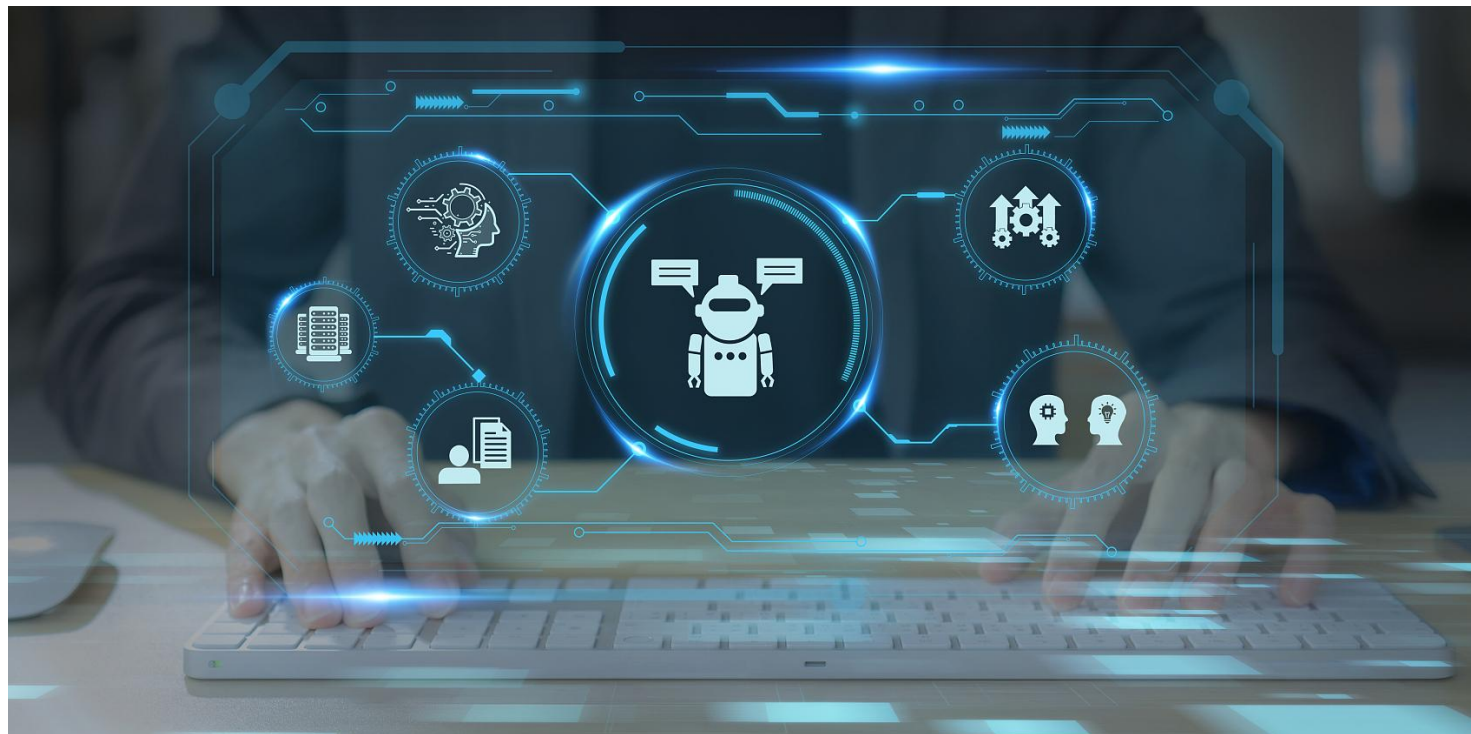
7.1.1 什么是大语言模型



■ 现存局限与使用要求

易出现事实偏差、逻辑漏洞，生成虚假信息，应对专业冷门问题时出错概率较高，输出内容受训练数据限制。

使用时需搭配人工核验修正，同时持续优化迭代以提升模型准确程度与实用价值。





7.1.2 大语言模型的优势



与传统自然语言处理模型相比，大语言模型具备更强的通用性，能够在同一个模型中完成问答、总结、翻译、分类、推理、代码生成等多种任务。

在学习场景中，大语言模型可以完成许多过去难以自动化的任务，具体如表所示。

任务类型	示例
知识问答	“什么是二叉搜索树？”
概念解释	“用本科生能听懂的方式解释动态规划”
文本总结	“帮我总结这份课程大纲”
代码辅助	“帮我找出这段 Python 代码的问题”
学习建议	“我还有两周考试，应该怎么复习操作系统？”



7.1.3 大语言模型的局限



大语言模型的本质是根据上下文生成最可能的文本，它擅长语言理解与内容生成，但并不天然具备所有能力，下表给出了大语言模型的一些局限。

局限	说明	在智学助手中的表现
无法访问实时信息	模型训练完成后，不会自动知道最新数据	不知道学校本学期最新考试安排
计算不够可靠	模型可能在复杂算术中出错	成绩加权计算可能算错
缺少外部行动能力	模型本身不能读文件、查数据库、发请求	无法查询校规知识库
记忆能力有限	上下文窗口有限，长对话可能遗忘	忘记用户前面说的是哪门课
流程控制较弱	难以稳定执行复杂分支流程	不同问题类型需要不同处理路径



7.1.3 大语言模型的局限



例如，用户向大语言模型提出如下问题：

我平时成绩37/40，期末82/100，平时占30%，期末占70%，总评是多少？

一个普通的大语言模型可能直接在文本中计算，但它并不保证每次都准确。对于教学示例而言，用户可以接受小误差，但是，在真实成绩系统中，这种小误差就可能造成严重问题。因此，智能体开发通常不会让大模型“凭感觉算”，而是让大模型识别出需要计算，再调用确定性的成绩计算函数来实现成绩的计算，确保得到准确的结果。



7.1.4 智能体解决什么问题



智能体的核心思想是：让大语言模型不再只是生成文本，而是成为一个能够使用工具、管理上下文、规划步骤并与环境交互的任务执行系统。

普通大语言模型应用更像“会回答问题的人”，而智能体则更像“会办事的助手”。大语言模型和智能体的具体区别见下表。

对比维度	大语言模型应用	智能体应用
主要能力	生成回答	完成任务
输入输出	用户输入 → 模型回答	用户目标 → 多步骤执行 → 结果反馈
工具使用	通常没有或较少	可调用计算器、搜索、数据库、文件等工具
状态管理	依赖当前对话上下文	可维护短期记忆、长期记忆和任务状态
决策方式	一次性生成	可根据中间结果动态决策
典型场景	问答、润色、总结	客服、数据分析、学习助手、自动化办公

因此，智能体并不是某一个单独模型，也不是简单的聊天窗口，而是一套围绕大语言模型构建的应用架构。

Part 02

智能体的定义与特征



7.2 智能体的定义与特征



智能体的
基本定义

智能体的
四个典型特征

智能体不等于
“会聊天的模型”



7.2.1 智能体的基本定义



在人工智能领域，智能体通常指能够感知环境、进行决策并采取行动的系统。进入大语言模型时代后，智能体的定义可以进一步具体化。智能体是指以大语言模型为核心，能够理解用户目标，结合记忆和工具进行推理、规划与行动，并根据环境反馈持续调整行为的软件系统。这个定义包含五个关键词：目标、感知、决策、行动和反馈（见下表）。

关键词	含义
目标	智能体不是随意聊天，而是围绕用户任务工作
感知	能接收用户输入、系统状态、工具返回结果等信息
决策	能判断下一步应该回答、调用工具、查询资料还是继续追问
行动	能执行外部工具、访问数据源、写入文件或触发业务流程
反馈	能根据工具结果或用户反馈修正后续行为



7.2.2 智能体的四个典型特征

■ 自主性

智能体能够在一定范围内自主决定下一步动作。例如，智学助手收到“帮我看看这门课有没有挂科风险”时，可以自行判断需要先计算成绩，再查询挂科规则，而不是要求用户逐步指挥。

自主性并不意味着完全不受控制。优秀的智能体应当在明确边界内自主行动，在不确定或高风险场景下主动向用户确认。





7.2.2 智能体的四个典型特征



■ 反应性

智能体能够根据环境变化做出响应。环境可以是用户的新输入、工具返回的数据、数据库查询结果，也可以是任务执行过程中的错误信息。

例如，成绩计算工具返回“平时成绩不能超过满分”时，智学助手不应继续编造答案，而应提示用户检查输入。





7.2.2 智能体的四个典型特征



■ 目标导向性

智能体的行为围绕目标展开，而不是只对单句话做表面回应。用户说“我下个月考试，现在有点慌”，目标可能不是查询知识点，而是制定复习计划并安抚情绪。智能体需要理解更深层的任务意图。





7.2.2 智能体的四个典型特征

■ 工具使用能力

工具使用是现代智能体区别于普通聊天机器人的关键特征。通过工具，智能体可以突破模型本身的限制，完成计算、搜索、数据库查询、文件读写、代码执行等操作。

在本书中，工具将从简单函数开始，逐步扩展为知识库查询、成绩计算、日期计算和文件保存等能力。



7.2.3 智能体不等于“会聊天的模型”



许多初学者容易把智能体理解为“更聪明的聊天机器人”。这种理解并不准确。聊天只是智能体与用户交互的一种形式，智能体的重点在于面向目标的行动能力。

需要说明的是，并非所有使用了大语言模型的系统都被称为智能体，有些系统可能使用了大语言模型，但只有一部分适合称为智能体，具体如表所示。

系统	是否属于智能体	原因
输入一句话，模型回答一句话	通常不是	没有工具、状态和行动
文档摘要工具	通常不是	任务固定，缺少自主决策
能调用计算器和搜索工具的学习助手	是	能根据问题选择工具并完成任务
能规划步骤、执行代码、分析结果的数据分析助手	是	具备目标分解、工具执行和反馈调整
固定流程的表单提交系统	通常不是	规则固定，不依赖智能决策



7.2.3 智能体不等于“会聊天的模型”



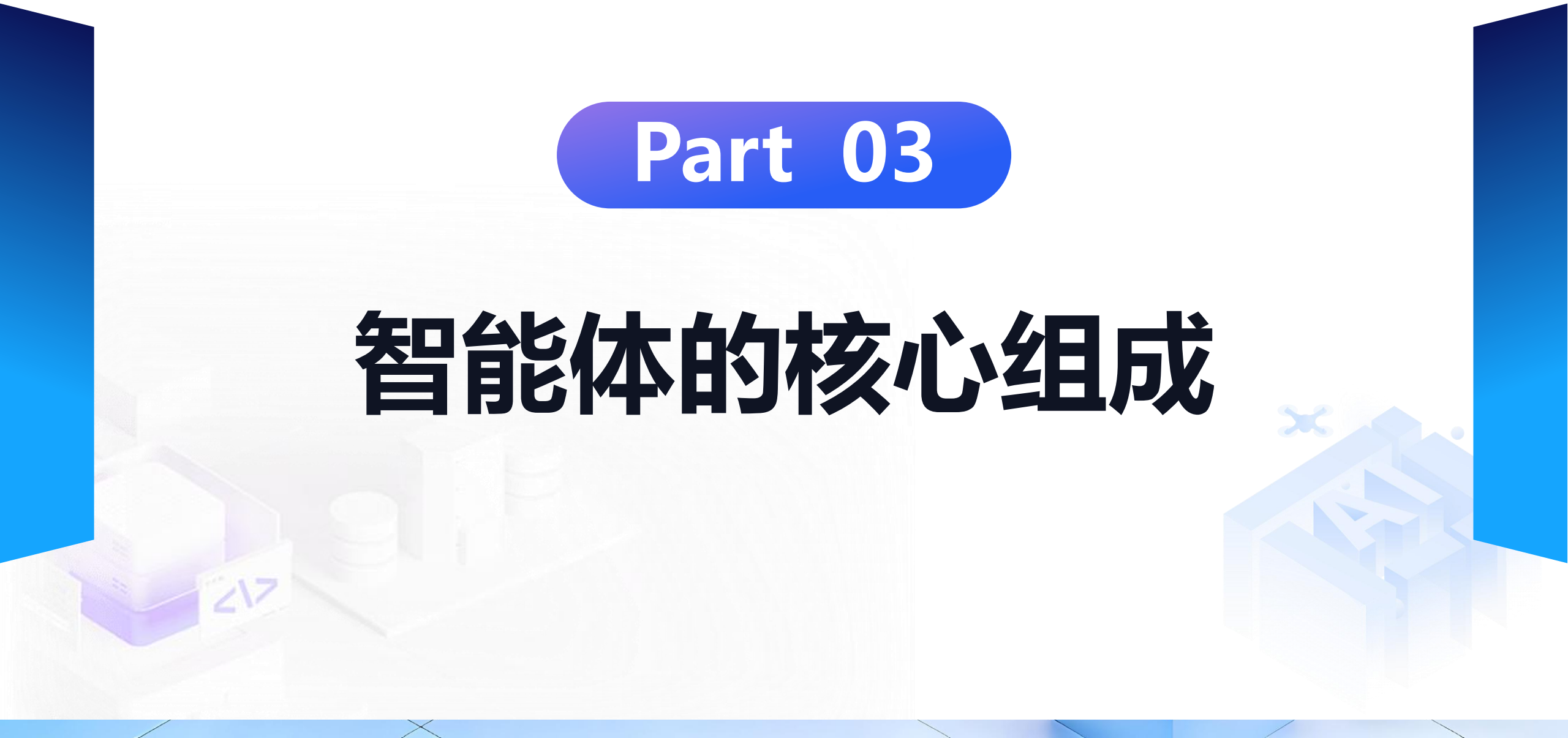
判断一个系统是否是智能体，可以问三个问题：



如果答案基本为“是”，这个系统就具备智能体特征。

Part 03

智能体的核心组成





7.3 智能体的核心组成



核心公式

LLM: 智能体
的大脑

Prompt: 智能
体的行为说明书

Tools: 智能体
的手和脚

Memory:
智能体的记忆

Planning:
智能体的规划
能力

Environment:
智能体的环境



7.3.1 核心公式



为了便于理解，本书采用以下公式描述智能体：

$$\text{Agent} = \text{LLM} + \text{Prompt} + \text{Tools} + \text{Memory} + \text{Planning} + \text{Environment}$$

这个公式的具体含义如下：

LLM（大语言模型）

LLM（大语言模型）提供语言理解、推理和生成能力。

Prompt（提示词）

Prompt（提示词）定义角色、任务、约束和输出格式。

Tools（工具）

Tools（工具）扩展智能体可执行的动作。

Memory（记忆）

Memory（记忆）保存对话历史、用户偏好和任务状态。

Planning（规划）

Planning（规划）将复杂目标拆解为可执行步骤。

Environment（环境）

Environment（环境）表示智能体可以感知和影响的外部世界。

这几个部分并不是孤立存在的，而是在一次任务执行中相互配合。



7.3.2 LLM：智能体的大脑



大语言模型（LLM）是智能体的推理核心，主要负责理解输入、判断意图、生成计划和组织回答。不同大模型在能力、成本、速度、上下文长度、多模态支持等方面各有差异。

在工程实践中，选择大模型时通常需要考虑的因素如表所示。

维度	说明
推理能力	是否能处理复杂任务和多步骤推理
工具调用能力	是否能稳定生成工具调用参数
上下文长度	是否能处理较长对话和文档
响应速度	是否满足交互式应用需求
调用成本	是否适合大规模使用
稳定性	输出格式是否可靠，服务是否稳定

7.3.3 Prompt: 智能体的行为说明书



提示词 (Prompt) 用于告诉大模型“你是谁、要做什么、不能做什么、应该如何输出”。在智能体中, 提示词不只是普通提问, 而是系统行为规范的一部分。

一个学习助手的系统提示词通常包含:

角色定位

你是面向大学学生的学习助手。

能力范围

可以解释知识、计算成绩、查询规定。

行为约束

不会编造学校政策, 不确定时说明原因。

工具使用要求

需要精确计算时必须调用工具。

输出风格

分步骤、清晰、适合本科生理解。

7.3.3 Prompt: 智能体的行为说明书



下面给出一个提示词的示例：

你是"智学助手"，面向大学本科生提供学习支持。

当用户询问课程知识时，请用清晰、循序渐进的方式解释。

当用户需要成绩、日期或学分计算时，必须调用工具，不要心算。

当用户询问学校规定时，必须基于知识库内容回答；如果没有查到，请明确说明。



7.3.4 Tools: 智能体的手和脚



工具让智能体能够执行模型本身无法可靠完成的操作。工具可以是一个 Python 函数，也可以是数据库查询、HTTP API、搜索引擎、文件系统或业务系统接口。在智学助手中，常见工具如表所示。

工具	作用
成绩计算工具	根据平时成绩、期末成绩和权重计算总评
日期计算工具	计算距离考试还有多少天
校规查询工具	从学校规定知识库中检索相关条款
笔记保存工具	将复习计划保存为本地文件
课程信息查询工具	查询课程名称、学分、考试安排等信息



7.3.4 Tools: 智能体的手和脚



工具设计需要注意两个问题：

- (1) 工具职责要清晰：一个工具只做一类确定的事情。
- (2) 参数结构要明确：模型需要知道工具需要哪些参数，以及参数的类型和含义。

例如，成绩计算工具不应接收一整段自然语言，而应接收结构化参数，具体如下：

```
calculate_grade(  
    final_score=82,  
    final_weight=0.7,  
    regular_score=37,  
    regular_full_score=40,  
    regular_weight=0.3  
)
```



7.3.5 Memory: 智能体的记忆

记忆 (Memory) 让智能体能够在多轮对话中保持连续性。没有记忆的系统只能处理当前输入，无法理解“刚才那门课”“上面那个分数”“继续按这个方案”等指代。

智能体中的记忆通常分为三类：短期记忆、长期记忆和任务状态（如表所示）。

类型	作用	示例
短期记忆	保存当前会话上下文	用户刚才提到数据结构成绩是85.15
长期记忆	保存跨会话信息	用户是计算机专业大二学生
任务状态	保存当前任务执行进度	已完成成绩计算，下一步生成复习计划

记忆并不是把所有对话无限制保存给大模型。实际系统需要对记忆进行筛选、摘要和检索，否则，会带来成本上升、上下文混乱和隐私风险。

7.3.6 Planning: 智能体的规划能力



规划 (Planning) 是智能体处理复杂任务的关键能力。面对一个目标, 智能体需要将其拆解为多个步骤, 并决定每一步使用什么工具。

例如, 用户提出如下需求:

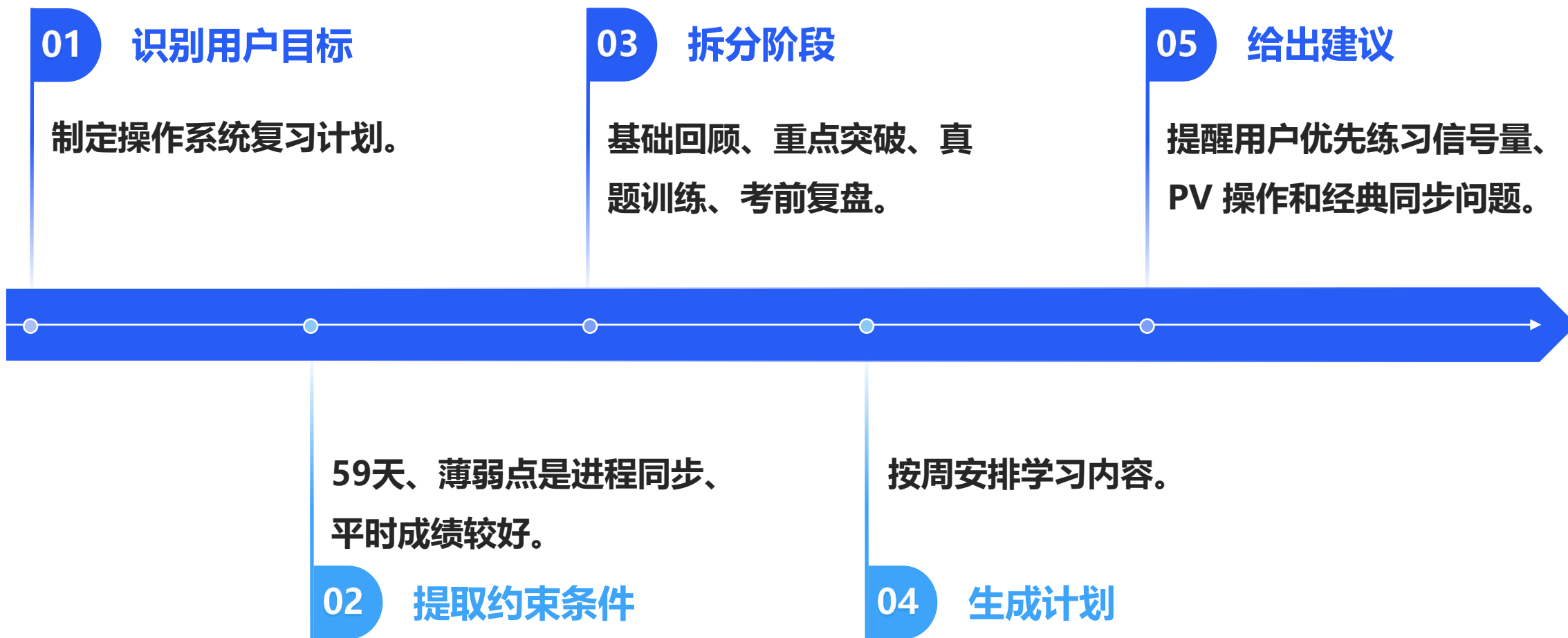
我还有59天考操作系统, 平时成绩还不错, 但我进程同步总是学不会。
帮我安排一个复习计划。



7.3.6 Planning: 智能体的规划能力



智能体可能形成如下计划：



在简单任务中，规划可能只隐含在一次大模型调用中；在复杂任务中，规划通常需要显式建模，这正是后续 LangGraph 要解决的问题。



7.3.7 Environment: 智能体的环境



环境 (Environment) 是智能体感知和影响的外部世界。对于机器人而言, 环境可能是物理世界; 对于软件智能体而言, 环境通常是文件、数据库、网页、API、用户会话或业务系统。

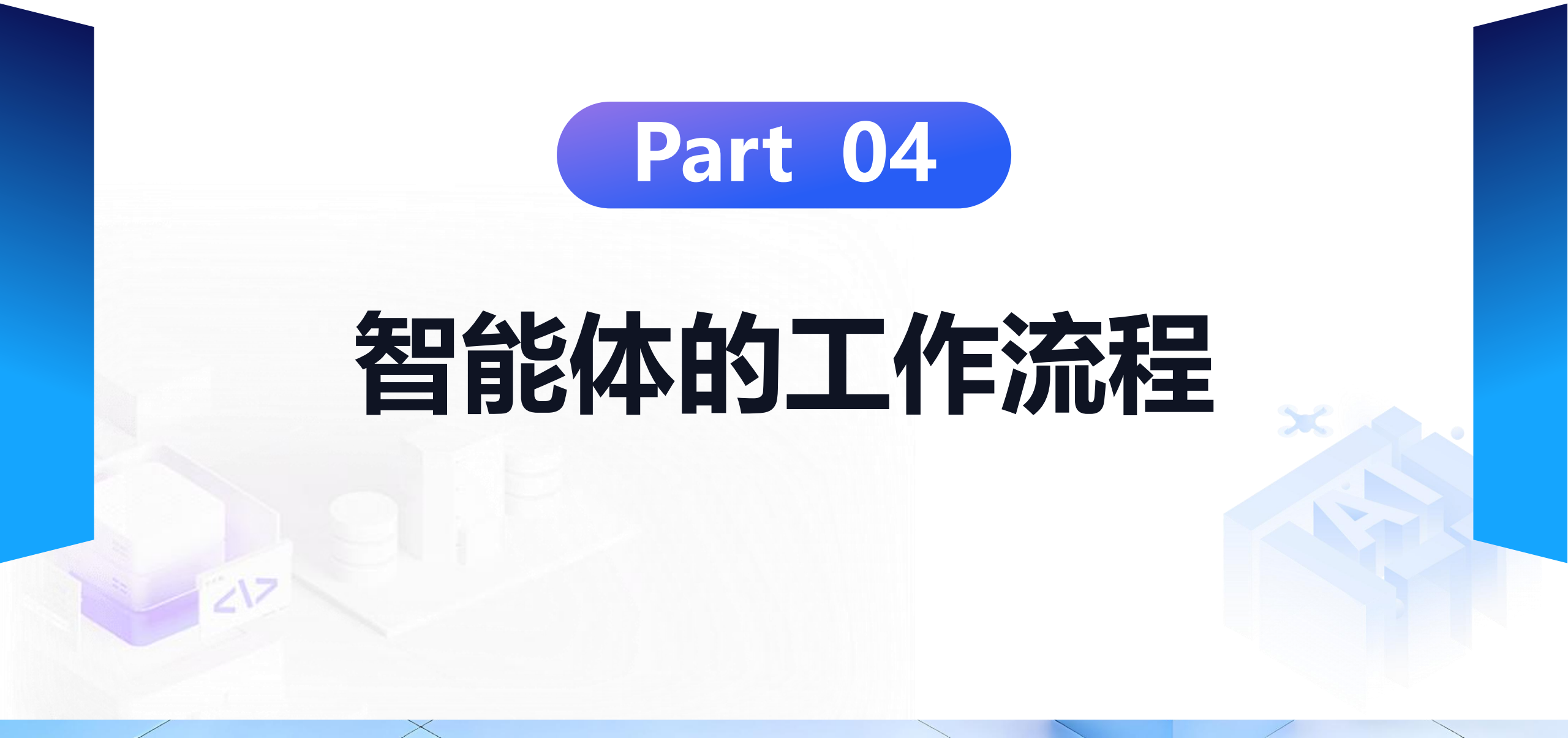
智学助手的环境包括:



智能体每一次工具调用, 都是在与环境交互。工具返回结果后, 智能体再根据反馈决定下一步动作。

Part 04

智能体的工作流程





7.4 智能体的工作流程



基本循环

示例：成绩判断
任务

为什么需要观察
结果



7.4.1 基本循环



智能体的运行过程可以概括为五个步骤：感知（Perception）、思考（Reasoning）、行动（Action）、观察（Observation）和反馈（Feedback）。这五步构成智能体的基本循环，具体含义如表所示。

步骤	含义	智学助手示例
感知	接收用户输入和上下文	用户询问总评成绩
思考	判断意图并决定下一步	识别为成绩计算任务
行动	调用工具或生成回答	调用成绩计算函数
观察	获取工具执行结果	得到总评85.15分
反馈	根据结果继续处理或回复	结合挂科规则回答用户

这个循环可以执行一次，也可以反复执行多次。复杂智能体往往会在一个任务中多次“思考—行动—观察”。



7.4.2 示例：成绩判断任务



下面用智学助手演示一次完整流程。

用户输入如下提示词：

我数据结构期末82分，平时37/40，总评按期末70%、平时30%算。
这门课会挂吗？

智能体的内部运行流程如下：

1. 感知：

- 课程：数据结构
- 期末成绩：82/100
- 平时成绩：37/40
- 权重：期末70%，平时30%
- 用户问题：是否挂科

2. 思考：

- 需要先计算总评
- 需要判断挂科规则
- 计算必须调用工具



7.4.2 示例：成绩判断任务



3. 行动:

- 调用 `calculate_grade` 工具
- 调用 `school_policy_search` 工具查询挂科规则

4. 观察:

- 工具返回总评85.15
- 知识库返回"总评低于60分或期末低于60分视为不及格"

5. 反馈:

- 组织自然语言答案
- 说明计算过程和判断依据



7.4.2 示例：成绩判断任务



智能体的最终回答如下：

你的数据结构总评成绩是85.15分。

计算过程为：期末 $82 \times 70\% = 57.4$ ；平时 $37/40 \times 100 \times 30\% = 27.75$ ；

总评为 $57.4 + 27.75 = 85.15$ 。

**根据"期末成绩低于60分或总评低于60分视为不及格"的规则，
你的期末成绩和总评成绩都超过60分，因此不会挂科。**



7.4.3 为什么需要观察结果



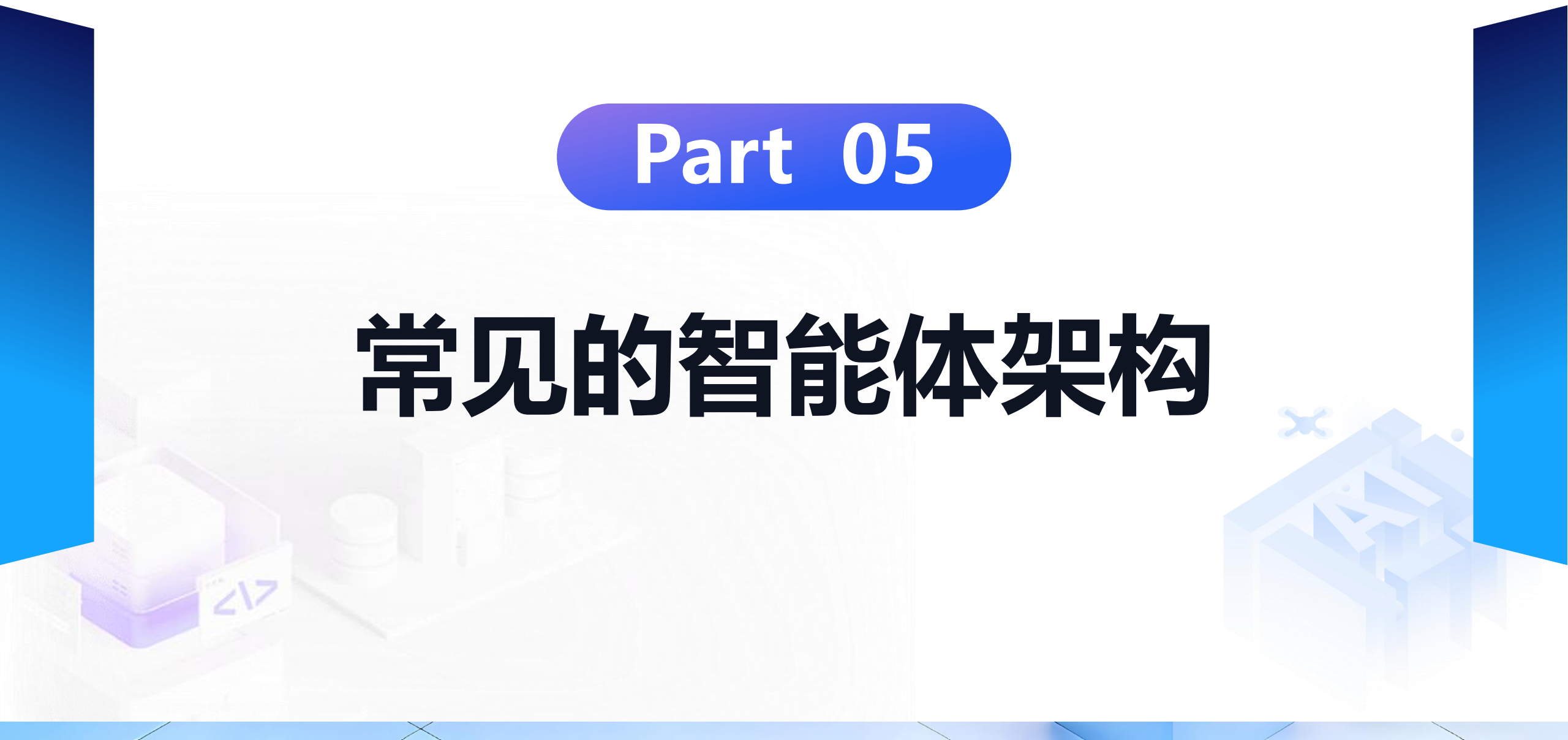
观察（Observation）是智能体区别于一次性问答的重要环节。大模型调用工具后，不能假设工具一定成功，也不能忽略工具返回的异常。下表给出了工具返回结果及其智能体的处理方式。

工具返回	智能体应如何处理
85.15	正常组织回答
缺少期末成绩	追问用户补充信息
平时成绩不能大于满分	提醒用户检查输入
知识库未找到相关规定	明确说明未查到，不编造
工具调用超时	告知用户稍后重试或提供替代方案

可靠的智能体必须认真处理工具返回结果，而不是只追求大模型回答看起来流畅。

Part 05

常见的智能体架构





7.5 常见的智能体架构



工具调用型
智能体

ReAct 智能体

规划执行型
智能体

workflow 型智能体

多智能体协作



7.5.1 工具调用型智能体



工具调用型智能体是最常见、最容易入门的智能体形式。它的核心流程是：模型判断用户问题是否需要工具，如果需要，就生成工具名称和参数，系统执行工具后再把结果交给模型生成最终回答。

01

工具调用型智能体的适用场景包括成绩计算、日期计算、天气查询、数据库检索、文件读写、API 调用等。

02

工具调用型智能体的基本工作流程是：用户问题→LLM判断是否需要工具→生成工具调用参数→执行工具→LLM 基于工具结果回答。

03

工具调用型智能体适合确定性任务，LangChain 就比较适合开发这类智能体。

04



7.5.2 ReAct 智能体



ReAct 是 Reasoning and Acting 的缩写，即“推理与行动结合”。它让大模型在执行任务时交替生成思考过程和行动指令，具体如下：

Thought: 我需要先计算总评成绩

Action: `calculate_grade`

Observation: 总评成绩为85.15

Thought: 还需要判断是否挂科

Action: `search_policy`

Observation: 总评低于60分或期末低于60分视为不及格

Final Answer: 你不会挂科.....

ReAct 的优点是过程清晰，便于调试，尤其适合需要多次工具调用的任务。它的缺点是调用轮次可能较多，成本和延迟更高，同时也需要防止大模型生成错误的工具参数。

在学习智能体时，ReAct 是非常重要的思想：让大模型边思考边行动，再根据观察结果继续调整。OpenClaw就属于这种类型的智能体。



7.5.3 规划执行型智能体



规划执行型智能体通常先生成一个任务计划，再逐步执行计划。它适合目标较复杂、步骤较多的场景。

例如，用户提出如下要求：

帮我准备一份为期4周的数据结构期末复习计划，
要结合我薄弱的图算法和动态规划部分。

规划执行型智能体可能先生成计划：

1. 收集用户基础信息
2. 划分复习阶段
3. 分配图算法复习任务
4. 分配动态规划练习任务
5. 安排每周测试与复盘
6. 输出可执行日程表

然后再逐步执行每一步。与 ReAct 相比，规划执行型智能体更强调先有整体方案，再进行具体操作。



7.5.4 workflow型智能体



workflow型智能体将任务流程显式建模为多个节点和边。每个节点负责一个具体步骤，边决定步骤之间如何流转。

例如，智学助手可以设计为：

用户输入



意图分类

- 学科问答 → 知识解释节点
- 成绩计算 → 计算节点
- 规定查询 → 知识库检索节点
- 情绪问题 → 安抚与建议节点



生成最终回答

workflow型智能体的优点是结构清晰、可维护性强，适合企业级系统。LangGraph 就比较适合开发这种类型的智能体。



7.5.5 多智能体协作



多智能体协作是指将复杂任务分配给多个具有不同角色的智能体完成。每个智能体负责一个专业方向，再通过协调机制共同完成任务。

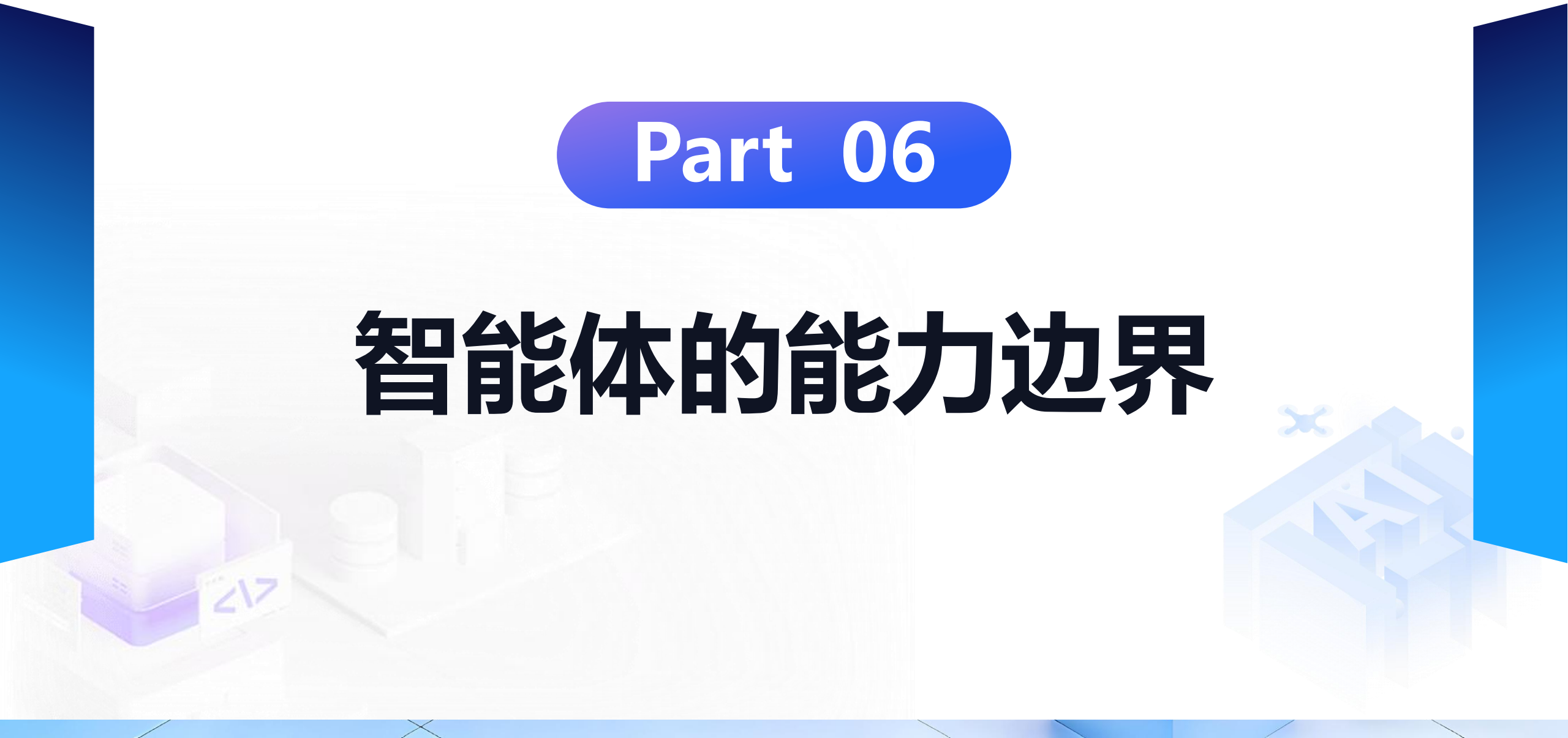
例如，一个课程辅导系统可以包含多个智能体，具体如表所示。

智能体	职责
知识讲解智能体	解释课程概念
练习生成智能体	生成题目和答案
学习规划智能体	制定复习计划
质量检查智能体	检查回答是否准确清晰

多智能体协作适合复杂任务，但也会引入通信成本、协调复杂度和结果一致性问题。对于初学者而言，应先掌握单智能体，再学习多智能体。

Part 06

智能体的能力边界





7.6 智能体的能力边界



智能体
擅长什么

智能体
不擅长什么

幻觉问题

自主性与
可控性的平衡



7.6.1 智能体擅长什么



智能体并不适合处理所有任务，表6-12给出了智能体适合处理的任务类型。

任务特征	说明	示例
目标明确但步骤可变	用户知道结果，但过程需要系统判断	制定复习计划
需要调用外部工具	模型本身无法可靠完成	成绩计算、日期计算
需要整合多类信息	问题涉及知识、数据、规则	判断是否挂科并解释原因
需要多轮交互	用户可能逐步补充信息	询问课程、成绩、考试时间
需要自然语言解释	不只是返回数据，还要讲清楚	解释为什么不会挂科

换言之，智能体适合“有目标、有信息、有工具、有过程”的任务。



7.6.2 智能体不擅长什么



智能体并不是万能系统。在一些特定的场景中，需要格外谨慎使用智能体，具体如表所示。

风险场景	问题
高风险决策	医疗、金融、法律等领域不能完全依赖模型判断
强实时控制	对毫秒级响应和确定性要求极高的系统不适合由 LLM 主控
规则极其固定	简单表单校验用传统程序更可靠
数据权限复杂	工具调用可能带来隐私和越权风险
事实准确性要求极高	必须接入可靠数据源并进行校验



7.6.3 幻觉问题



1

幻觉 (Hallucination) 是指大模型生成看似合理但实际上错误的内容。智能体接入工具后, 幻觉问题不会自动消失, 甚至可能因为工具使用错误而变得更隐蔽。

2

智能体应用中常见的幻觉包括编造不存在的学校规定、错误引用知识库内容、声称已经调用工具但实际没有、对工具返回结果做错误解释、在缺少数据时自行补全参数等。

3

降低幻觉的常见方法包括: 对事实类问题接入检索系统或数据库、对计算类任务使用确定性工具、在提示词中要求不知道就说明不知道、对关键输出进行格式校验和二次检查、保留工具调用日志以便于追踪问题来源。



7.6.4 自主性与可控性的平衡



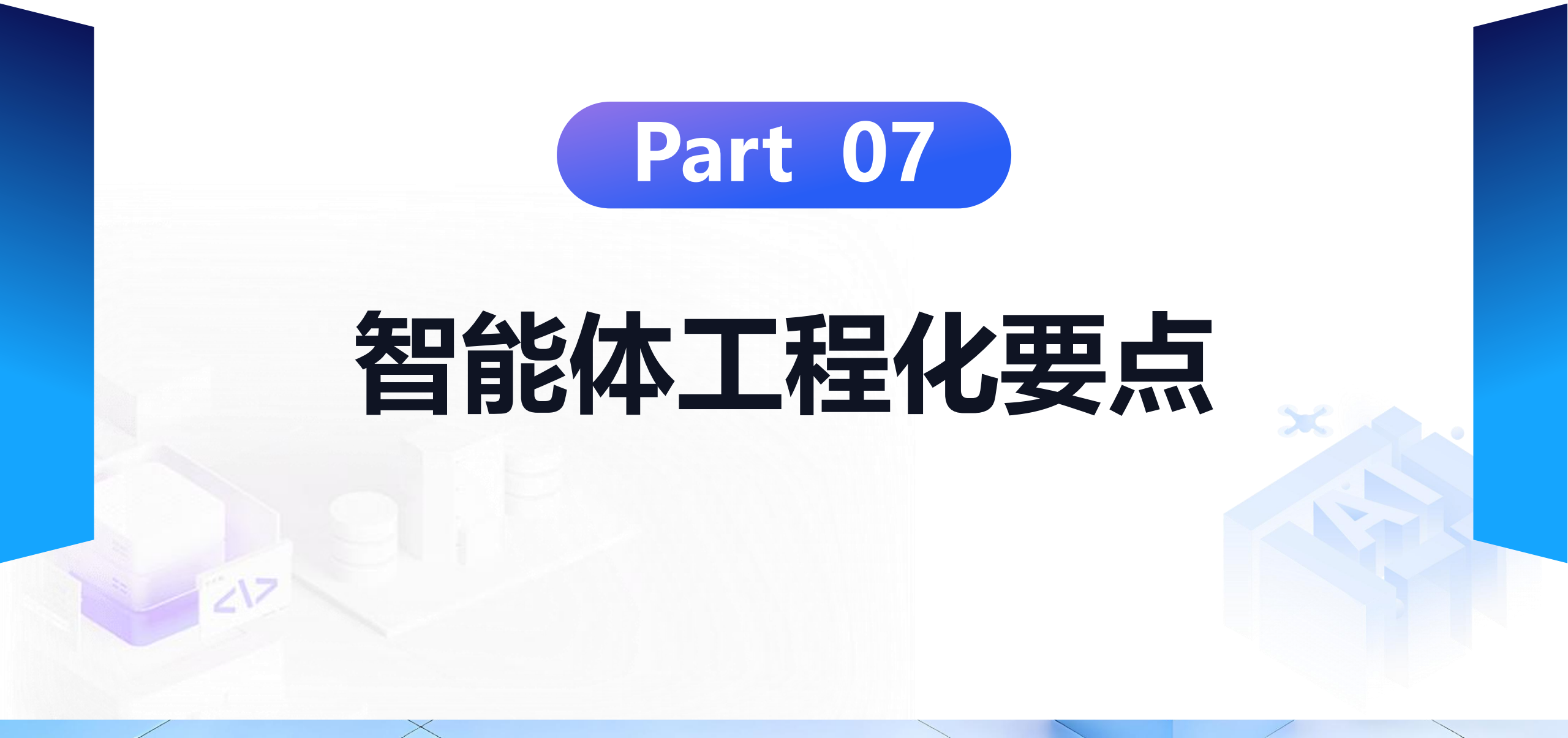
智能体越自主，越能减少用户操作；但自主性越强，失控风险也越高。工程实践中需要在两者之间取得平衡。例如，智学助手可以自动保存一份复习计划草稿，但如果要替用户向教师发送邮件，就应先征得用户确认。下表给出了平衡智能体自主性和可控性的一般原则。

行为类型	是否需要用户确认
生成建议	通常不需要
查询公开信息	通常不需要
本地临时计算	通常不需要
保存用户文件	建议提示
发送消息或提交表单	必须确认
修改重要数据	必须确认

智能体开发不是让系统“想做什么就做什么”，而是让系统在清晰边界内高效完成任务。

Part 07

智能体工程化要点





7.7 智能体工程化要点



从演示系统到
可用系统

可靠性

安全性

可观测性

成本与性能



7.7.1 从演示系统到可用系统



很多智能体演示系统看起来很神奇，但真正投入使用时会遇到一系列工程问题，具体如表所示。

问题	演示阶段	可用系统阶段
提示词	写一段能跑即可	需要版本管理和测试
工具	函数能调用即可	需要参数校验、错误处理和权限控制
记忆	保存全部对话	需要摘要、检索、隐私保护
输出	看起来合理即可	需要结构化格式和质量校验
调试	靠肉眼看日志	需要可观测性和追踪工具
成本	调用次数少，不敏感	需要缓存、限流和模型分级



7.7.2 可靠性



可靠性是企业级智能体必须关注的问题。一个可靠的智能体应当具备：





7.7.2 可靠性



例如，成绩计算工具应检查如下事项：

- 期末成绩是否在 0 到 100 之间
- 平时成绩是否不超过平时满分
- 权重之和是否等于 1
- 输入是否缺少必要字段

只有这样，智能体才能从演示程序变成可信工具。



7.7.3 安全性



智能体可以调用工具，也就意味着它可能影响外部系统，可能出现一些安全性问题（如表所示）。

风险	示例
越权访问	学生查询到他人信息
提示词注入	用户诱导模型忽略系统规则
工具滥用	模型反复调用高成本 API
数据泄露	将隐私信息发送给不该访问的工具
错误执行	未确认就修改或删除数据

安全设计的基本原则是：模型可以建议，但关键动作必须受程序和权限系统约束。

例如，不应让大模型直接决定是否能访问某个学生的成绩，而应由后端权限系统判断。大模型只负责理解问题和组织回答。



7.7.4 可观测性



可观测性指系统能够记录和展示智能体的运行过程，使开发者知道它为什么这样回答。

一个可观测的智能体应能记录：

用户原始输入

大模型生成的中间决策

调用的工具名称

工具调用参数

工具返回结果

最终回答

错误与重试信息

没有可观测性，开发者只能看到最后回答，很难判断问题来自提示词、模型、工具还是数据源。



7.7.5 成本与性能



智能体通常比普通大语言模型调用更消耗资源，因为它可能在一次任务中多次调用模型和工具。开发时需要关注的内容包括：



例如，用户只是问“总评怎么算”，可以用较简单的大模型回答；如果用户要求“根据我的学习情况生成一个四周复习计划”，则可以使用推理能力更强的大模型。模型分级是降低成本的常见方法。



7.8本章小结



本章系统阐述了智能体的核心概念与开发框架，明确智能体区别于普通聊天机器人，是以大语言模型为核心，融合多重能力、可自主完成任务的软件系统。

LLM是智能体的核心基础，但仅靠LLM无法解决实时信息获取、运算、外部行动及复杂流程处理等问题。

智能体具备自主性、反应性、目标导向性与工具使用能力四大核心特征，可通过公式
$$\text{Agent} = \text{LLM} + \text{Prompt} + \text{Tools} + \text{Memory} + \text{Planning} + \text{Environment}$$
精准界定，运行遵循“感知—思考—行动—观察—反馈”的循环机制。

本章介绍了工具调用型、ReAct型、规划执行型等多种主流智能体架构，同时指出智能体工程化落地需兼顾可靠性、安全性、可观测性、成本与性能等关键要素。



谢谢！

林子雨 副教授 厦门大学

附录A：主讲教师林子雨简介

主讲教师：林子雨

单位：厦门大学计算机科学与技术系

E-mail: ziyulin@xmu.edu.cn

个人网页: <http://dblab.xmu.edu.cn/post/linziyu>

数据库实验室网站: <http://dblab.xmu.edu.cn>



- 林子雨，男，1978年出生，博士（毕业于北京大学），全国高校知名大数据与人工智能教师，入选“2021年高校计算机专业优秀教师奖励计划”。现为厦门大学计算机科学与技术系副教授，厦门大学信息学院实验教学中心主任，曾任厦门大学信息科学与技术学院院长助理、晋江市发展和改革委员会副局长。中国计算机学会数据库专业委员会执行委员，中国计算机学会信息系统专业委员会执行委员。
- 国内高校**首个“数字教师”提出者和建设者**，厦门大学数据库实验室负责人，厦门大学云计算与大数据研究中心主要建设者和骨干成员，2013年度、2017年度、2020年度、2023年度和2026年度厦门大学教学类奖教金获得者，荣获2024年福建省高等教育教学成果奖特等奖（个人排名第七）、2022年福建省高等教育教学成果奖特等奖（个人排名第一）、2018年福建省高等教育教学成果奖二等奖（个人排名第一）、2018年国家精品在线开放课程、2021年国家级线上一流本科课程、2020年国家级线上一流本科课程。**编著出版了20余本大数据与人工智能系列教材，被国内1000多所高校采用，3本教材入选教育部“十四五”普通高等教育本科国家级规划教材**；建设了**国内高校首个大数据课程公共服务平台**，为教师教学和学生学习大数据课程提供全方位、一站式服务，年访问量超过400万次，累计访问量超过3200万次。**大数据系列MOOC课程入选“2023年教育部国家智慧教育公共服务平台应用典型案例”**。2025年发布4个大模型科普报告，全网累计浏览量超过1000万，并应邀为高校、政府和企业做170余场大模型讲座。
- 主要研究方向为数据库、数据仓库、数据挖掘、大数据、云计算和物联网，并以**第一作者身份在《软件学报》《计算机学报》和《计算机研究与发展》等国家重点期刊以及国际学术会议上发表多篇学术论文**。作为项目负责人主持的科研项目包括1项国家自然科学基金青年基金项目(No.61303004)、1项福建省自然科学基金青年基金项目(No.2013J05099)和1项中央高校基本科研业务费项目(No.2011121049)，主持的教改课题包括1项2016年福建省教改课题、1项2016年教育部产学协作育人项目、1项2024年教育部产学协作育人项目。

[illegible]

大数据学习路线图访问地址: <http://dblab.xmu.edu.cn/post/10164/>

附录C：林子雨大数据系列教材



了解全部教材信息: <http://dblab.xmu.edu.cn/post/bigdatabook/>

附录D：林子雨人工智能通识系列教材

厦门大学林子雨**人工智能通识**系列教材
被众多高校采用，满足不同高校差异化教学需求



了解全部教材信息：<http://dblab.xmu.edu.cn/post/ai-book/>

附录E：林子雨Python系列教材



了解全部教材信息: <https://dblab.xmu.edu.cn/post/python-books/>

谢谢!

林子雨 副教授

厦门大学

