第 1 章

智能体基础

当智能音箱精准响应你的语音指令，当自动驾驶汽车平稳穿梭于城市车流，当智能客服高效解决你的咨询难题，这些场景背后，都离不开“AI 智能体”这一核心技术的支撑。作为人工智能领域从理论走向实用的关键载体，智能体正在重塑我们与技术的交互方式，甚至改变社会的运行逻辑。对于想要踏入 AI 智能体应用开发领域的读者而言，夯实智能体的基础认知，就如同掌握打开智能时代大门的第一把钥匙。本章作为技术篇的开篇，旨在从科普视角出发，为读者构建关于智能体的完整知识框架。我们将从智能体的核心定义切入，理清其与传统程序的本质区别；沿着时间脉络追溯其从理论萌芽到爆发式增长的发展历程，展现技术演进的内在逻辑；深入剖析其自主性、适应性、学习能力与交互性四大核心特点，解读其强大应用潜力的根源；最后直面智能体在技术、伦理安全及开发生态层面的现实挑战，为后续的开发实践铺垫理性认知基础。通过本章的学习，希望读者能够建立起对智能体的系统性理解，为进一步探索其应用开发筑牢根基。

1.1 智能体基础

本节将深入探讨技术本质，从定义、特征、构成三个维度搭建智能体的基础认知框架，通过对比传统程序凸显其智能核心，理清其概念边界与运行逻辑，为后续的技术实践筑牢理论根基。

1.1.1 智能体的核心定义与本质

在人工智能的宏大版图中，智能体（Agent）占据着举足轻重的基础地位，宛如一座大厦的基石，支撑起整个智能应用的摩天大楼。国际人工智能领域权威学者罗素（Stuart J. Russell）与诺维格（Peter Norvig）在经典著作《人工智能：一种现代的方法》中，给出了具有奠基性的界定：智能体是“能够感知环境并通过行动影响环境的实体”[1]，这一定义为后续学界与产业界的认知演进奠定了核心框架。

如图 1-1 所示，智能体的核心属性在于“感知-决策-执行”的闭环能力——它通过传感器获取环境信息，依托算法模型分析研判并生成决策，再通过执行器输出具体行动，最终围绕预设目标形成完整闭环，这种闭环交互特性是智能体与普通模型最根本的区别[3]。其形态具有极强的包容性，打破了单一载体的局限：既可以是纯软件程序（如智能客服、股票交易机器人），也可以是软硬件

结合的实体设备（如自动驾驶汽车、服务机器人），甚至可以是由多个单体智能体协同构成的复杂系统（如智能电网调度系统、智慧城市管控平台）[2]。

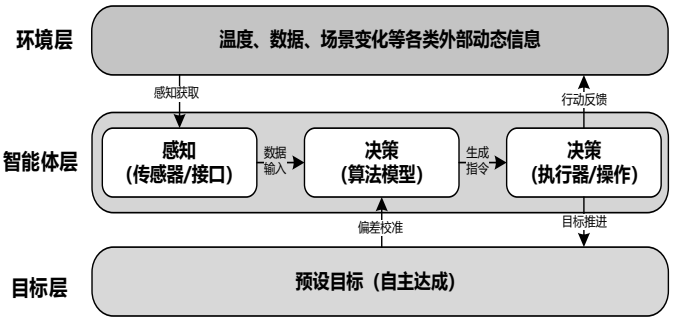

图 1-1　智能体“感知–决策–执行”闭环逻辑示意图

从本质上看，智能体是对人类“感知–思考–行动”认知模式的工程化模拟与优化，这也是其区别于传统工具的核心逻辑[1]。相较于传统工具的“被动响应”特质，智能体更强调“主动适配”的智能属性，无须人工实时干预即可根据环境动态调整行为，实现目标的自主达成，这一特质使其成为连接人工智能技术与实际应用场景的核心载体，标志着 AI 从“助手”向“操作员”的质变[5]。

1.1.2　智能体与传统程序的核心差异

智能体的自主性是其区别于传统程序的核心标志，二者并非技术复杂度的梯度差异，而是底层运行逻辑与能力边界的本质区别[4]。这种差异集中体现在目标导向、环境交互与行为逻辑三个核心维度，如表 1-1 所示。

表1-1　智能体与传统程序核心差异对比表

对比维度	传统程序	智能体
决策自主性	依赖人工预设指令序列，无独立判断能力，线性执行输入–输出任务	基于环境实时反馈，依托算法独立生成决策方案，无须人工干预
环境适应性	局限于预设场景，超出边界即失效或报错，无自适应能力	动态感知环境变化，自主调整行为策略，适配复杂场景波动
目标灵活性	输出结果与执行路径固定，无法根据目标达成情况优化	以核心目标为导向，动态调整行动路径，通过替代方案推进目标实现

具体可概括为以下三点：

其一，决策自主性不同。传统程序完全依赖人工预设的指令序列与逻辑规则运行，缺乏独立判断与决策能力，本质是“输入–输出”的线性执行工具；而智能体可基于实时环境反馈与内置算法独立生成决策方案，无须人工干预即可完成任务推进。其二，环境适应性不同。传统程序的运行范围被严格限定在预设场景内，一旦超出场景边界便会失效或报错；智能体则具备动态感知环境变化的能力，可根据场景波动自主调整行为策略，适配复杂多变的应用需求。其三，目标灵活性不同。传统程序的输出结果与执行路径相对固定，无法根据目标达成情况优化调整；智能体则以核心目标为导向，可动态优化行动路径，即便面临局部障碍，也能通过替代方案推进目标实现。

典型案例可直观体现这种差异：智能家居系统中的智能恒温器，核心目标是“维持室内恒温”，

它会通过传感器实时感知温度变化，自主控制空调、地暖的启停与功率调节，全程无须人工操作，完全区别于传统温控器“固定阈值触发”的机械逻辑[1]；而计算器、文本编辑器等传统程序，仅能根据预设算法执行固定功能，无法自主判断使用场景的合理性，也不能基于用户潜在需求调整输出，始终处于被动响应状态。

1.1.3　智能体的核心构成要素

完整的智能体系统并非单一模块的独立运行，而是由多个功能模块协同联动形成的闭环体系，如图 1-2 所示，核心构成包括感知模块、决策模块、执行模块与目标模块 4 个部分，现代智能体架构在此基础上还可延伸出记忆模块，各模块分工明确、相互支撑，共同赋予智能体自主运行与主动适配的能力[6]。

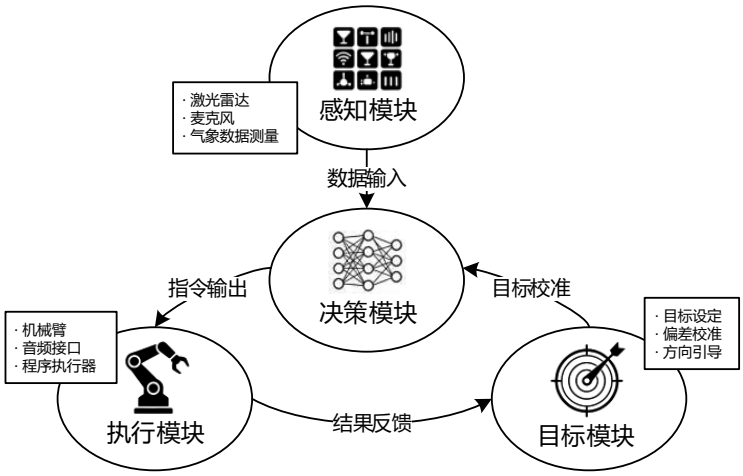

图 1-2　智能体四大核心构成要素及协同示意图

感知模块是智能体的“感官系统”，核心功能是收集与预处理环境数据，为后续决策提供基础支撑，负责将多模态的原始数据转化为内部统一的表征[7]。其载体既可以是硬件设备（如自动驾驶汽车的激光雷达、摄像头，服务机器人的麦克风、触觉传感器），也可以是软件接口（如智能客服的文本采集接口、金融智能体的行情数据接口），最终实现多模态环境信息的精准捕获与格式标准化[2]。

决策模块是智能体的“核心大脑”，承担数据解析、逻辑推理与方案生成的核心职责，相当于智能体的“前额叶”，负责结合感知信息与目标需求制定行动序列[5]。该模块依托算法模型（如机器学习模型、大语言模型等）对感知模块传输的数据进行深度分析，结合目标需求拆解任务、规划行动路径，生成可落地的执行指令，是智能体实现自主决策的核心支撑[2]。

执行模块是智能体的“行动载体”，负责将决策模块生成的指令转化为具体行动。其形态与应用场景高度适配，既可以是机械结构（如工业机器人的机械臂、无人机的动力系统），也可以是软件操作（如智能办公助手的文档编辑、邮件发送操作），核心目标是精准落地决策指令，推动任务推进。

目标模块是智能体的“方向指南针”，承担核心目标设定、目标优先级排序与目标达成校验的功能。它为决策模块提供明确的行动依据，同时实时监测执行结果与目标的偏差，反馈至决策模块以触发策略优化，确保智能体的所有行为都围绕核心目标展开，避免无意义的资源消耗。

以智能音箱为例，如图 1-3 所示，四大模块的协同逻辑清晰可辨：感知模块通过麦克风接收用户语音指令，经语音识别技术转化为可解析的文本数据；决策模块调用自然语言处理模型理解“播放音乐”“查询天气”等具体需求，生成对应的执行方案；执行模块通过音频接口播放音乐、语音播报天气，完成指令落地；目标模块则始终以“精准响应用户需求、提升交互体验”为核心，实时校验执行效果，若出现指令识别偏差，则触发二次确认，持续优化全流程交互逻辑，四大模块的联动赋予其超越传统音箱的智能特性。

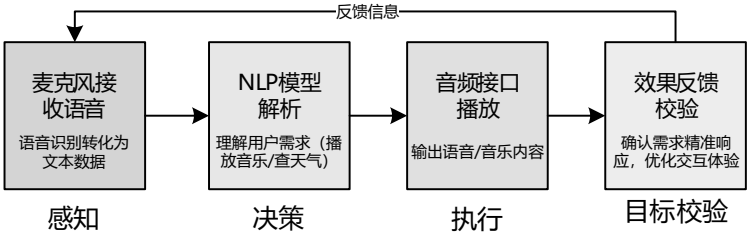

图 1-3　智能音箱四大模块协同实例流程示意图

1.2　智能体发展历史

智能体的发展并非一蹴而就，而是经历了漫长的演进过程，其发展脉络与人工智能整体演进高度契合，本质是技术范式迭代、核心算法突破与应用需求驱动的双向奔赴。从早期规则编码到如今大模型赋能，每个阶段的标志性算法都以可量化的公式为支撑，推动智能体能力边界持续拓展。结合核心技术路线、算法突破与能力边界变化，可将智能体发展划分为 4 个关键阶段，每个阶段的理论突破与技术成果，都为后续的跨越式发展奠定了坚实基础[4]。

1.2.1　奠基期：符号主义主导的理论萌芽

这一阶段以符号主义为核心技术范式，聚焦“规则驱动”的逻辑推理能力，核心目标是实现机器对人类逻辑的模拟与复现。1950 年，阿兰·图灵（Alan Turing）在《心灵》（*Mind*）期刊发表《计算机器与智能》一文，首次提出“图灵测试”，为智能体“智能性”的评判建立了核心标准，也为后续智能体的理论探索指明了方向[8]。1956 年，达特茅斯会议正式确立人工智能概念后，科学家们开始系统性地将人类逻辑规则转化为机器可执行的代码，开启了智能体理论落地的初步尝试。

这一阶段的代表性成果集中于早期专家系统，核心依托符号逻辑推理公式构建运行体系，其核心逻辑可简化为：若存在规则集合 $R=\{r_1,r_2,\cdots,r_n\}$（r_i 代表一条人工编码规则，如“若化合物含羟基，则可能为醇类”），输入事实集合 $F=\{f_1,f_2,\cdots,f_m\}$（待分析的化合物特征），则推理结果可通过以下公式计算：

$$O=R\otimes F \tag{1-1}$$

其中，$\otimes$ 表示规则与事实的匹配运算，通过该运算将预设规则应用于输入特征，最终输出推理结论[11]。其中，最具里程碑意义的是 1965 年费根鲍姆（Edward Feigenbaum）团队开发的 DENDRAL 系统，

该系统通过预设有机化学领域的专业规则与推理逻辑，可精准分析未知有机化合物的分子结构，准确率达到人类专家水平，成为首个商业化应用的智能体雏形[11]。但受限于技术条件，这类系统存在明显局限：完全依赖人工编码规则，规则集合 R 无法通过数据自主更新，缺乏自主学习与环境适配能力，且规则库扩展难度极高，仅能在封闭特定场景应用，难以应对复杂动态环境。

1.2.2　黄金期：连接主义推动的学习革命

随着神经网络技术的兴起与成熟，连接主义逐渐取代符号主义成为主流范式，智能体正式迈入“数据驱动”的发展阶段，其核心突破在于具备了从数据中自主学习的能力，而反向传播（Backpropagation）算法正是这一突破的核心支撑[12]。1986 年，杰弗里·辛顿（Geoffrey Hinton）与团队在《自然》（*Nature*）发表论文，提出反向传播算法，其核心是通过梯度下降最小化损失函数，实现神经网络参数的迭代优化，核心公式包括两部分：

一是损失函数（以均方误差为例），用于衡量模型预测值与真实值之间的偏差：

$$L=\frac{1}{2}\sum_i(y_i-\hat{y}_i)^2 \tag{1-2}$$

其中，y_i 为真实标签，$\hat{y}_i$ 为模型预测值，L 代表整体预测偏差，偏差越小，说明模型预测效果越好。

二是参数更新公式，用于根据损失函数调整网络参数以降低偏差：

$$w=w-\eta\cdot\frac{\partial L}{\partial w} \tag{1-3}$$

$$b=b-\eta\cdot\frac{\partial L}{\partial b} \tag{1-4}$$

其中，w 为权重参数，b 为偏置参数，η 为学习率（控制参数更新步长，避免更新幅度过大或过小），$\partial L/\partial w$ 和 $\partial L/\partial b$ 为损失函数对对应参数的偏导数，指引参数调整方向[12]。这套公式解决了多层神经网络的训练难题，使智能体能够从海量数据中自动提取特征、优化参数，摆脱了对人工规则的完全依赖，让神经网络具备了实用价值。

这一阶段的理论与算法创新持续涌现：1994 年，迈克尔·伍德里奇（Michael Wooldridge）与尼古拉斯·詹宁斯（Nicholas Jennings）提出 BDI（Belief-Desire-Intention，信念-愿望-意图）模型，构建了智能体认知与决策的核心框架，明确了智能体从目标认知到行动执行的逻辑链条，成为多智能体系统研究的基础理论[9]；1992 年，理查德·萨顿（Richard Sutton）在《机器学习》（*Machine Learning*）期刊发表研究，提出时间差分学习（Temporal Difference Learning）算法，为智能体交互试错学习提供了数学支撑，其核心公式为：

$$V(s)\leftarrow V(s)+\alpha[r+\gamma V(s')-V(s)] \tag{1-5}$$

其中，$V(s)$ 为状态 s 的价值评估，α 为学习率，r 为当前动作获得的奖励，γ 为折扣因子（衡量未来奖励的重要性，取值 0~1，越接近 1 越重视未来收益），s' 为执行动作后的下一个状态[13]。简单来说，该算法让智能体能够通过“当前奖励+未来预期奖励”修正对当前状态的判断，逐步优化策略，实现“感知-学习-优化”的初步闭环。依托这些技术突破，智能体在模式识别、简单任务调度等领域取得了阶段性成果，逐步从理论走向实验室验证。

1.2.3 复兴期：深度学习加持的能力跃升

大数据技术的普及与云计算算力的爆发式增长，为智能体发展提供了核心支撑，深度学习与强化学习的深度融合，推动智能体性能实现质变。2015 年，DeepMind 团队在《自然》发表论文，提出深度 Q 网络（Deep Q-Network，DQN）算法，该算法将卷积神经网络（Convolutional Neural Network，CNN）与 Q 学习结合，用神经网络拟合 Q 值函数，解决了传统强化学习无法处理高维环境数据的难题，其核心公式为：

$$Q(s,a;\theta) \leftarrow Q(s,a;\theta) + \alpha\{r + \gamma \cdot \max_{a'}[Q(s',a';\theta^{-}) - Q(s,a;\theta)]\} \tag{1-6}$$

其中，$Q(s,a;\theta)$ 为状态 s 下执行动作 a 的价值（θ 为神经网络参数），α 为学习率，r 为奖励，γ 为折扣因子，θ^{-} 为目标网络参数（固定一段时间更新，避免训练过程中参数波动过大，提升模型稳定性）[10]。这套公式让智能体仅依靠游戏像素数据（高维状态 S）就能自主学习规则，在 Atari 2600 系列游戏中甚至超越人类玩家水平，标志着深度强化学习技术走向成熟，也为智能体处理高维环境信息提供了有效路径[10]。

2017 年，DeepMind 推出 AlphaGo Zero，通过“无监督自我对弈”模式实现棋力快速迭代，其核心依托蒙特卡罗树搜索（Monte Carlo Tree Search，MCTS）与深度神经网络结合，核心公式包括策略网络、价值网络及整体目标函数三部分。

（1）策略网络：用于预测状态 s 下各动作 a 的概率分布，指导智能体选择最优动作：

$$p_{\theta}(a \mid s) \tag{1-7}$$

（2）价值网络：用于评估状态 s 的获胜概率，判断当前局势优劣：

$$v_{\theta}(s) \tag{1-8}$$

（3）整体目标函数：用于优化网络参数，兼顾策略准确性、价值评估精度与模型泛化能力：

$$L(\theta) = -\sum_{s}[\pi(a \mid s) \cdot \log p_{\theta}(a \mid s) + (z - v_{\theta}(s))^2] + \lambda \| \theta \|^2 \tag{1-9}$$

其中，π 为自我对弈的最优策略，z 为对弈结果（胜 1、负−1、平 0），λ 为正则化系数（防止模型过拟合，提升泛化能力）[14]。该算法无须人类专家数据，仅通过自我对弈生成训练数据，持续优化网络参数，最终突破人类围棋水平上限，其核心的多智能体协同进化机制，为通用智能体的研发提供了重要参考[14]。此外，这一阶段的智能体开始从专用场景向通用能力探索，在自动驾驶、工业质检、智能导航等领域实现初步落地，逐步突破实验室边界，开启技术商业化的初步尝试。

1.2.4 爆发期：大模型驱动的实用化浪潮

大型语言模型（Large Language Models，LLM）的突破性进展，彻底重塑了智能体的能力边界，推动其进入“理解−规划−执行”全流程自主化的实用化阶段。大模型的核心是 Transformer 架构的自注意力机制，其核心公式为：

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}[\frac{\boldsymbol{Q} \cdot \boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}] \cdot \boldsymbol{V} \tag{1-10}$$

其中，$\boldsymbol{Q}$（查询）、$\boldsymbol{K}$（键）、$\boldsymbol{V}$（值）为输入向量的不同表征，d_k 为键向量维度（$\sqrt{d_k}$ 用于缩放，避免因向量维度过高导致梯度消失），softmax 函数用于归一化注意力权重，使模型能聚焦输入中的关键信息[15]。这套公式让模型能够捕捉文本中不同 Token 的关联关系，具备强大的上下文理解能力。2022 年，OpenAI 推出 GPT-3.5，依托优化后的自注意力机制与海量语料训练，凭借强大的自然语言理解与生成能力，使语言智能体能够精准响应复杂指令，自主完成文本生成、代码编写、逻辑推理等多样化任务，大幅降低了智能体的开发与应用门槛[15]。

2024 年，斯坦福大学团队开展“虚拟小镇”（Smallville）实验，25 个具备记忆、推理与社交能力的智能体在虚拟环境中自主交互，形成稳定的社会关系网络与行为模式，验证了多智能体协同决策的可行性[10]。2025 年，多模态智能体 Zeelin 通过跨模态融合技术，实现“一句话指令生成完整研究报告”的功能，集成文本、数据、图表的全流程自动化生成，进一步拓展了智能体的应用场景[16]。在国内，百度文心一言、阿里通义千问等大模型平台持续迭代，通过开源工具链降低智能体开发门槛，推动智能体在办公自动化、教育辅助、医疗咨询等领域实现规模化应用。

1.3　智能体的特点

智能体之所以在人工智能领域备受瞩目，成为推动科技进步与创新的关键力量，得益于其独具的四大核心特点：自主性、适应性、学习能力和交互性。这些特点相互交织、协同赋能，既构建了智能体的核心能力体系，又赋予其强大的场景适配性与应用拓展潜力，使其能够在复杂多变的环境中灵活应对不确定性，高效完成多样化任务，为人类生产生活的智能化变革提供核心支撑[9]。

1.3.1　自主性

自主性是智能体的核心特质，指其无须人类实时干预，基于预设目标与动态环境信息，独立完成“感知-决策-执行”全流程的能力，本质是人类决策权限的部分或完全转移[1]。典型如智能投资助手，通过 API 接口实时抓取金融市场行情、宏观经济数据及个股基本面信息，依托量化模型分析风险收益比并自主制定调仓策略，当标的跌破预设止损线时自动触发卖出操作，可在 7×24 小时内不间断监控市场波动，有效规避人工交易中的情绪干扰与操作延迟[18]。

与自动驾驶分级类似，智能体的自主性可依据人类参与度与决策权限，划分为 5 级架构，该分级体系已成为智能体设计与风险管控的核心参考[19]。

（1）一级（操作员主导）：智能体仅具备数据采集与初步预处理能力，所有决策与执行动作需人类操作员触发，自身无独立行动权限。例如，传统数据统计工具仅能输出数据报表，无法基于报表生成任何操作建议。

（2）二级（人机协作）：智能体可基于环境信息生成备选方案，但需人类确认后才能执行。例如，初级智能办公助手可梳理邮件优先级并生成回复草稿，最终发送的内容需用户手动审核确认。

（3）三级（智能体主导，人类监督）：智能体可自主完成全流程任务，人类仅需在任务执行中进行被动监督，在发生异常情况时介入干预。例如，智能巡检机器人可自主规划巡检路线、识别设备故障，仅在故障等级较高时触发人类复核流程。

（4）四级（智能体自主，人类事后审核）：智能体具备完整自主决策与执行能力，人类无须实时监督，仅在任务完成后进行结果审核，无异常则流程闭环。例如，智能供应链调度系统可自主调整库存补货计划并触发物流调度，人类仅需定期核查调度结果。

（5）五级（完全自主）：智能体可在开放动态环境中自主设定子目标、优化行动策略，无须人类任何干预，具备故障自修复与目标自适应能力。目前该等级仅在特定封闭场景实现，如实验室环境下的自主研发智能体，可自主设计实验方案、分析实验数据并迭代优化研究方向。

这一分级架构的核心价值是为智能体的场景化落地提供权限边界，避免过度自主带来的风险，同时最大化其效率优势[19]。各等级在人类参与度、决策权限、核心能力及应用场景上的差异，可通过表 1-2 中的 5 级自主性模型清晰区分。

表1-2　人工智能代理设计的5级自主性模型

自主性等级	人类参与模式	核心决策权限	关键能力特征	典型应用场景
一级（操作员主导）	全程主导，智能体仅为辅助工具	人类完全掌控，智能体无决策权限	仅支持数据采集、初步预处理与结果输出	传统数据统计工具、基础报表生成软件
二级（人机协作）	人类审核确认，智能体生成方案	智能体提供备选方案，人类最终决策	具备场景分析与方案生成能力，无执行权限	初级智能办公助手（生成邮件草稿）、简历优化工具
三级（智能体主导，人类监督）	被动监督，异常时介入干预	智能体自主决策执行，人类保留干预权	全流程自主运行，具备异常检测与预警能力	智能巡检机器人、工业设备故障监测系统
四级（智能体自主，人类事后审核）	事后审核结果，无须实时参与	智能体完全自主决策执行，人类不干预过程	具备流程闭环能力，支持结果追溯与核查	智能供应链调度系统、自动补货管理平台
五级（完全自主）	无任何干预，仅作为观察者	自主设定子目标、优化策略，自我修复故障	开放环境适配、目标自适应、故障自修复	实验室自主研发智能体、封闭场景下的自主探索机器人

该表格基于文献[19]的分级框架优化，补充了能力特征与场景细节，可直观反映自主性等级提升过程中，人类参与度递减、智能体决策权限递增的核心逻辑，为不同场景下智能体的选型与设计提供参考。

1.3.2　适应性

适应性是智能体感知环境变化并动态调整行为策略的能力，核心是“环境感知–偏差识别–策略迭代”的快速响应闭环，也是智能体区别于传统固定程序的关键特质[20]。其适配范围涵盖环境参数波动（如温度、湿度、路况变化）、任务目标调整（如优先级变更、需求迭代）、外部干扰介入（如设备故障、数据中断）等多种场景，依托多模态感知设备与柔性决策模型，实现对复杂场景的动态适配。从技术层面看，适应性的核心是“感知精度”与“调整速度”的协同，感知模块的多设备融合确保环境信息无遗漏，决策模块的轻量化算法则保障策略调整的实时性。

自动驾驶汽车是适应性的典型应用载体：通过激光雷达、摄像头、毫米波雷达等多设备融合感

知路况信息，当检测到雨天环境时，自动降低行驶速度、增大跟车距离，同时调整刹车灵敏度以应对路面湿滑；若遇突发施工路段，可在 0.5 秒内完成备选路线规划，兼顾路线长度与通行效率，无须人类干预即可完成动态避障[21]。这种全场景自适应能力，是传统巡航系统仅能基于固定参数运行的模式无法比拟的，也是智能体在开放环境中落地的核心前提。

从技术逻辑来看，智能体的适应性依赖于两层核心支撑：硬件层面的多模态感知设备，确保环境信息的全面捕获；算法层面的动态规划模型，能够基于感知数据快速生成优化策略，二者协同实现“感知无死角、调整无延迟”的适配效果[20]。

1.3.3　学习能力

学习能力是智能体通过交互积累经验、迭代优化模型参数与行为策略的核心动力，也是其实现能力持续提升、突破场景局限的关键，本质是对人类经验学习（experiential learning）模式的工程化复现与效率提升[22]。与传统程序需人工修改代码、更新规则才能升级不同，智能体可通过数据交互、环境试错、人类反馈等多种路径自主学习，实现功能与精度的持续迭代，无须人工介入即可适配新场景、新需求。这种自主学习能力，使智能体从“固定功能工具”升级为“能力进化实体”。

智能客服系统的能力演进是学习能力的直观体现：初始阶段仅能通过关键词匹配应答预设问题，交互范围局限于固定话术库；随着用户交互数据的积累，可通过语义相似度分析学习“退货流程”与“怎么把东西寄回去”等同义表述，自动扩充话术适配范围；当新功能上线时，仅需导入产品文档与历史问答数据，3~5 天内即可通过自主学习实现精准应答，无须人工逐句标注训练样本[23]。

从学习范式来看，智能体的学习能力可分为三类核心模式：一是监督学习，依托标注数据优化模型预测精度；二是强化学习，通过与环境交互试错，基于奖励机制迭代最优策略；三是无监督学习，自主挖掘数据隐含规律，实现未知场景的能力迁移[22]。目前主流智能体多采用混合学习范式，结合人类反馈强化学习（Reinforcement Learning from Human Feedback，RLHF）优化学习效率，如图 1-4 所示，确保学习成果与实际需求精准对齐。

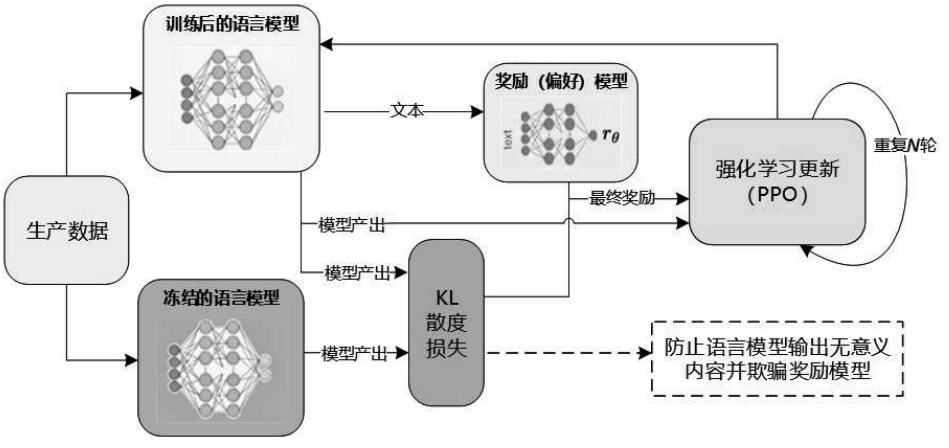

图 1-4　人类反馈强化学习流程示意图

人类反馈强化学习（RLHF）是一种结合人类偏好与强化学习的 AI 训练方法，核心在于通过人类标注的反馈信号引导模型优化行为。

第一阶段：监督微调（Supervised Fine-Tuning，SFT）。用高质量人工标注数据训练初始语言模型，使其初步具备遵循指令的能力，此阶段模型学习基础任务范式。

第二阶段：奖励模型（Reward Model，RM）训练。收集人类对模型不同输出的偏好排序数据，训练奖励模型以量化评估回答质量，为后续强化学习提供“评分标准”。

第三阶段：强化学习（Reinforcement Learning，RL）优化。基于奖励模型的评分，使用近端策略优化（Proximal Policy Optimization，PPO）等算法更新模型参数，使模型在与环境交互中最大化累积奖励，逐步逼近人类偏好的行为模式。

1.3.4 交互性

交互性是指智能体与人类、其他智能体及外部系统进行信息共享、指令传递与协同协作的能力，是智能体融入复杂系统、实现规模化应用的核心前提[24]。其交互维度可分为两类核心场景：人机交互聚焦“自然化、低门槛”，通过语音、文字、手势、表情等多模态交互方式，降低人类操作成本，实现“意图精准传递、结果高效反馈”；机机交互聚焦“高效化、高兼容”，通过标准化 API 接口、分布式通信协议及区块链存证技术，实现跨平台数据互通、指令同步与协同决策，确保多智能体、多系统协作的稳定性与安全性。两类交互模式深度融合，构建起“人类–智能体–外部系统”三位一体的协同生态。

智能办公系统的协同逻辑可直观体现交互性的核心价值：智能体可通过自然语言交互接收用户“转化数据为柱状图”的指令，随后与财务系统协同提取营收数据，与办公软件协同生成可视化图表；同时同步对接项目管理系统，抓取任务进度数据并自动提醒任务截止时间，形成“用户指令–多系统协同–结果输出”的完整闭环[16]。这种多维度交互能力，使智能体成为连接分散系统与人类需求的核心枢纽，大幅提升跨场景协作效率。

人机交互的核心追求是“自然化、低门槛”，通过语音、文字、手势等多模态交互方式，降低人类操作成本；机机交互的核心追求是“高效化、高兼容”，通过 API、区块链等技术实现跨平台数据互通与指令同步，确保多智能体协同的稳定性[24]。两类交互模式的深度融合，为智能体的规模化落地构建了核心支撑。

1.4 智能体应用的挑战

尽管智能体技术展现出巨大的潜力和广阔的应用前景，在金融、交通、医疗等多个领域实现了初步落地，但在规模化推广与深度应用过程中，仍面临技术、伦理、安全及开发生态等多重交织的挑战。这些挑战并非孤立存在，而是相互关联、相互制约，既涉及底层技术的固有局限，也涵盖产业生态与社会认知的现实壁垒，成为阻碍智能体从“实验室走向产业化”的核心障碍。唯有系统性破解这些难题，才能推动智能体技术真正实现从理论突破到实践赋能的跨越，为人类社会生产生活的智能化变革提供可持续支撑[1]。

1.4.1 技术瓶颈

核心瓶颈集中在三个方面，且均直击智能体在复杂场景落地的核心能力需求：

一是长链条任务规划不足，复杂场景下多环节协同与因果推理能力薄弱。以银行贷款审批智能体为例，其需整合征信记录、异地资产核验、收入稳定性评估、行业风险预判等多维度信息，在面对自由职业者收入波动、跨区域资产抵押、隐性负债排查等特殊场景时，难以构建完整的因果逻辑链，易出现决策片面性，导致审批效率与准确性失衡[27]。这种局限本质上是现有模型对复杂任务的拆解与全局优化能力不足，难以模拟人类的多维度综合判断思维。

二是不确定环境决策薄弱，开放动态环境中的突发因素与干扰项，易打破智能体的预设决策框架。户外机器人配送在暴雨、极端低温、突发交通管制等场景下，任务成功率从确定性环境的 95%降至 60%以下，核心症结在于环境感知的动态适配不足与应急策略储备有限——既无法快速精准识别复合型环境变化，也缺乏灵活的备选方案迭代能力，相较于人类的临场应变存在明显差距[4]。

三是健壮性欠缺，模型易受对抗性攻击、数据噪声干扰，容错能力不足。例如，仅通过修改交通标志局部像素、添加微小视觉干扰，即可误导自动驾驶系统对标志的识别结果，将限速标志误判为通行标志，引发安全风险[29]；同时，在训练数据分布偏移时，智能体的决策精度会急剧下降，难以适配真实场景中的数据多样性。

1.4.2　伦理与安全困境

伦理与安全问题是智能体规模化应用的核心底线制约，二者相互关联、叠加影响。伦理层面主要面临双重困境：

一方面是情感共鸣缺失与人文关怀不足，在医疗、养老、教育等强情感交互场景中，智能体虽能依托算法提供标准化专业服务，却无法感知人类的情绪变化、传递情感慰藉，难以替代人类的人文关怀价值。例如，医疗智能体可精准诊断疾病、生成治疗方案，但无法理解患者的恐惧、焦虑情绪，难以给予针对性的心理疏导，可能影响治疗依从性[29]。

另一方面是道德判断模糊与伦理共识缺失，自动驾驶面临的“电车难题”仅是典型代表，在资源分配、风险权衡等场景中，智能体的道德决策缺乏统一标准——不同地区、文化、群体对生命优先级、公平性的认知存在差异，导致道德决策模型难以通用化，且难以将人类的伦理准则精准转化为算法逻辑[30]。

安全层面则聚焦两大风险点：

一是决策“黑箱”引发的隐性偏见与公平性问题，智能体的决策过程受训练数据影响深远，若数据存在历史偏见（如招聘数据中对女性的隐性歧视），模型会强化这种偏见，导致招聘智能体在筛选简历时歧视女性、少数群体，且因决策链路不透明，难以追溯偏见根源与责任主体[31]。

二是数据安全与隐私泄露风险，智能体在运行过程中需采集、处理大量敏感数据（如医疗场景的患者隐私、金融场景的用户资产信息），部分企业为降低成本简化数据加密机制、缺乏完善的安全防护体系，导致数据泄露事件频发。2024 年，某医疗智能体因数据库防护漏洞，泄露 10 万余名患者的病历、身份证号等核心隐私信息，引发严重的社会信任危机与法律纠纷[26]。

1.4.3　开发生态与经济难题

开发生态的碎片化与经济层面的投入产出失衡，共同制约了智能体产业的规模化发展。生态层面，核心痛点在于标准化缺失与协同性不足：

一是 API 接口不兼容，不同厂商的智能体基于各自的技术架构开发，接口协议、数据格式缺乏统一标准，导致跨厂商、跨系统的智能体无法实现高效协同工作，形成“信息孤岛”。例如，办公场景中，智能客服系统与智能日程管理系统因接口不兼容，无法实现客户咨询信息与日程安排的自动联动[32]。

二是核心模块复用性低，算法模型、感知模块、决策引擎等核心组件缺乏统一的封装与共享机制，企业需重复搭建基础框架，研发成本高、周期长。

三是测试与评估标准缺失，不同场景对智能体的性能、安全性、可靠性要求不同，缺乏统一的测试指标与评估体系，导致智能体的性能验证缺乏公信力，难以形成行业共识。

经济层面则面临“高投入、低回报”的恶性循环：

一方面，中等规模行业智能体的训练、部署及运维成本极高，仅数据标注、算力支撑、定制化开发等环节的投入就达数百万元，超出了绝大多数中小企业的承受范围，导致行业准入门槛偏高，市场参与者集中于大型企业[33]。

另一方面，用户留存率偏低、商业变现能力不足，部分消费级智能产品（如智能音箱）因功能单一、场景适配不足、交互体验欠佳，月留存率仅 30%，用户付费意愿低迷；行业级智能体则面临定制化需求强、复制性弱的问题，难以通过规模化降低成本，进一步加剧了投入产出失衡的困境，制约了产业生态的良性循环。

1.5 本章小结

本章围绕智能体基础展开，系统梳理了智能体的核心概念、发展历程、核心特点及应用挑战，为读者构建了智能体认知的基础框架。在概念层面，我们明确智能体是具备环境感知、自主决策与任务执行能力的实体，其自主性与交互性是区别于传统程序的关键，智能恒温器、自动驾驶汽车等实例让这一概念更加具象。

发展历程的梳理显示，智能体从 20 世纪中叶的理论萌芽起步，历经符号主义主导的奠基期、连接主义推动的黄金期、强化学习与深度学习融合的复兴期，在大数据与大模型的加持下进入爆发期，每一步演进都离不开技术突破的支撑。而自主性、适应性、学习能力与交互性四大核心特点，共同赋予了智能体强大的实用价值，从智能投资助手到智能办公系统，这些特点在不同场景中得到了充分体现。

同时，我们也清醒地认识到，智能体的发展并非坦途，长链条任务规划不足、决策透明度欠缺、开发生态不完善等问题，是其从技术走向规模化应用的主要障碍。本章的内容为后续深入学习智能体的开发技术提供了理论铺垫，理解这些基础认知与现实挑战，将有助于更有针对性地探索智能体的应用落地路径。

1.6　参考文献

[1] RUSSELL S, NORVIG P. Artificial Intelligence: A Modern Approach[M]. 4th ed. Boston: Pearson, 2020: 32-35.

[2] 马晓宁. AI 智能体核心架构解析：从模块协同看智能闭环的实现[EB/OL]. (2025-09-22)[2026-01-20]. https://m.toutiao.com/group/7552914570319364642/.

[3] WANG G, ZHAO Y, SONG S, et al. Voyager: An Open-Ended Embodied Agent with Large Language Models[EB/OL]. (2023-05-23)[2026-01-20]. arXiv:2305.16291.

[4] STONE P, VELOSO M. Multiagent Systems: A Survey from a Machine Learning Perspective[J]. Autonomous Robots, 2000, 8(3): 345-383.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet Classification with Deep Convolutional Neural Networks[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe: NeurIPS Foundation, 2012: 1097-1105.

[6] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All You Need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: NeurIPS Foundation, 2017: 6000-6010.

[7] OPENAI. GPT-4 Technical Report[EB/OL]. (2023-03-14)[2026-01-20]. arXiv:2303.08774.

[8] TURING A M. Computing Machinery and Intelligence[J]. Mind, 1950, 59(236): 433-460.

[9] WOOLDRIDGE M, JENNINGS N R. Intelligent Agents: Theory and Practice[J]. The Knowledge Engineering Review, 1995, 10(2): 115-152.

[10] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-Level Control Through Deep Reinforcement Learning[J]. Nature, 2015, 518(7540): 529-533.

[11] FEIGENBAUM E A, BUCHANAN B G, LEDERBERG J. On Generating Explanations for Organic Chemistry Structures[J]. Computers & Chemistry, 1969, 3(1): 29-40.

[12] HINTON G E, SEJNOWSKI T J, WILLIAMS R J. Learning Representations by Back-Propagating Errors[J]. Nature, 1986, 323(6088): 533-536.

[13] SUTTON R S. Learning to Predict by the Methods of Temporal Differences[J]. Machine Learning, 1988, 3(1): 9-44.

[14] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the Game of Go Without Human Knowledge[J]. Nature, 2017, 550(7676): 354-359.

[15] OPENAI. GPT-3.5 Technical Report[EB/OL]. (2022-11-30)[2026-01-20]. arXiv:2211.07522.

[16] PARK J, O'BRIEN J, CAVALLUCCI E, et al. Generative Agents: Interactive Simulacra of Human Behavior[C]//Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology. New York: ACM Press, 2023: 809-822.

[17] Wooldridge M. An Introduction to MultiAgent Systems[M]. 2nd ed. Chichester: John Wiley & Sons, 2009: 45-48.

[18] JORDAN M I, MITCHELL T M. Machine Learning: Trends, Perspectives, and Prospects[J]. Science, 2015, 349(6245): 255-260.

[19] BOWMAN S, CLARK J, HAUSKNECHT M, et al. Levels of Autonomy for AI Agents[EB/OL]. (2025-06)[2026-01-20]. https://arxiv.org/abs/2506.12469v1.

[20] KIM J H, LEE S, PARK J. Adaptive Agent Architecture for Dynamic Environment Interaction[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2024, 54(8): 4890-4901.

[21] LEVIN A, KRUMM J, HALLEM S. Autonomous Driving in Adverse Weather Conditions[C]// Proceedings of the 2022 IEEE International Conference on Robotics and Automation. Philadelphia: IEEE Press, 2022: 3456-3463.

[22] SONG S, ZHAO Y, WANG G. Kolb-Based Experiential Learning for Generalist Agents[EB/OL]. (2024-11-15)[2026-01-20]. https://arxiv.org/abs/2411.03562.

[23] LEWIS M, YANG Y, ROSEN Z. Context-Aware Learning for Intelligent Customer Service Agents[J]. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 1-20.

[24] JENNINGS N R, WOOOLDRIDGE M. Agent-Oriented Software Engineering[J]. Journal of Autonomous Agents and Multi-Agent Systems, 2022, 36(4): 1-18.

[25] HU S Y, YAN H Y, ZHANG Y Q, et al. Single-Agent Scaling Fails Multi-Agent Intelligence: Towards Foundation Models with Native Multi-Agent Intelligence [C]//Proceedings of the 39th International Conference on Machine Learning. Vienna: PMLR, 2025: 18923-18942.

[26] YOUNG R. On the Computational, Informational, and Physical Foundations for AI Safety [C]//Proceedings of the Eighth AAAI/ACM Conference on AI, Ethics, and Society. New York: ACM, 2025: 2944-2946.

[27] CHOI J, LEE S, PARK J. ReAcTree: Hierarchical LLM Agent Trees with Control Flow for Long-Horizon Task Planning [C]//Proceedings of the 39th International Conference on Machine Learning. Vienna: PMLR, 2025: 12345-12360.

[28] ZHOU Q, CHEN S, WANG Y, et al. HAZARD Challenge: Embodied Decision Making in Dynamically Changing Environments [C]//Proceedings of the 12th International Conference on Learning Representations. Vienna: ICLR, 2024: 1-18.

[29] WU C H, SHAH R, KOH J Y, et al. Dissecting Adversarial Robustness of Multimodal LM Agents [C]//Proceedings of the 38th International Conference on Machine Learning. Vienna: PMLR, 2024: 23456-23472.

[30] HOWCROFT A, BENNETT-WESTON A, KHAN A, et al. AI chatbots versus human healthcare professionals: a systematic review and meta-analysis of empathy in patient care [J]. British Medical Bulletin, 2025, 156 (1): ldaf017.

[31] 古天龙, 李龙, 常亮, 等. 公平机器学习: 概念、分析与设计[J]. 计算机学报, 2023, 46(5): 987-1010.

[32] 袁雷, 张子谦, 李立和, 等. 开放环境下的协作多智能体强化学习进展[J]. 中国科学: 信息科学, 2025, 55(2): 217-268.

[33] Docin.com. 智能客服行业市场技术发展现状及应用前景报告[R]. 北京: Docin.com, 2025.

第 2 章

智能体感知能力

在智能体逐步走向通用化与自主化的技术背景下，感知能力已成为连接模型智能与真实世界的首要入口与关键基石。无论是对复杂文档的结构理解，还是对语音、图像、视频等多模态信息的综合解析，智能体都必须首先“看见”“听见”并理解环境，才能作出可靠的推理与决策。本章聚焦智能体感知能力这一核心主题，从整体架构视角出发，系统阐述感知能力在智能体体系中的定位与作用机制，并以文档感知为重点，深入拆解文档预处理的关键任务、输出形态及主流技术工具。在此基础上，进一步拓展至音频、图像与视频等其他模态感知能力，勾勒多模态协同感知的整体图景。通过本章内容，读者将建立起对智能体感知体系的系统认知，为后续理解更高层次的推理、规划与执行能力奠定坚实基础。

2.1 智能体感知能力概述

智能体感知能力是指智能体借助各类感知模块，对外部环境及多模态输入信息进行捕捉、识别与解析，并将其转化为自身可处理的结构化数据的核心能力。这一能力构成了智能体实现自主决策与交互响应的基础，其本质在于搭建起智能体与外部世界之间的信息桥梁，打破不同模态信息的隔阂，实现对多元输入的精准获取与初步语义转化，为后续的推理与执行环节提供可靠的数据支撑。

从模态类型来看，智能体感知涵盖文档、音频、视频、图像等多种形式，不同感知场景各有侧重[1]。文档感知侧重于结构化与非结构化文档的信息提取，覆盖文档预处理全流程成果，能够有效捕捉文本内容、版式结构以及表格、公式等关键信息，广泛适用于办公自动化、学术研究等文本密集型场景。音频感知则通过语音识别技术将语音信号转化为文本，同时结合语调、语速等特征，提取其中蕴含的情绪与语境信息，支撑语音交互、实时转写等应用。

视频感知进一步融合图像帧级分析与时序逻辑建模，从动态画面中提取物体、动作、场景及其时间关联信息，实现对连续内容的理解与解析。此外，感知能力还可扩展至触觉信号与各类传感器数据，逐步形成覆盖多源信息的综合感知体系。

在智能体整体架构中，感知能力处于承上启下的关键位置，直接影响智能体对环境的适配程度与任务执行的准确性。若缺乏高效而稳定的感知能力，智能体将难以准确接收外部指令与环境反馈，其后续的推理与决策也将失去可靠的数据基础。与此同时，感知能力所具备的多模态融合特性，使智能体能够应对更加复杂、多变的现实场景，实现跨模态信息的协同利用，为智能体由“被动响应”向“主动感知、自主决策”的演进提供核心支撑，是其具备通用能力的重要基础。

2.2 文档感知任务定义与分类

2.2.1 文档输入类型

文档预处理所面对的输入类型较为丰富，主要可分为可编辑文本类、版式固定类、演示类以及图像类四大范畴。不同类型文档在存储结构、内容形态以及预处理适配性方面存在显著差异，这些差异直接影响后续预处理环节中技术路径的选择。因此，准确识别各类输入文档的特性，是确保预处理流程高效运行与精准处理的前提条件。

可编辑文本类文档以纯文本或结构化文本为主体，常见格式包括 TXT、Markdown、DOC、DOCX 及 WPS。其中，TXT 属于纯文本格式，不包含任何排版信息，仅保存文字内容，预处理复杂度最低。Markdown 通过轻量级标记语法支持标题、列表等基础结构定义，具备一定的结构表达能力。DOC 与 DOCX 为 Word 的核心格式，前者采用二进制存储，后者基于 XML 架构，均支持较为复杂的排版与样式信息；WPS 与上述格式具有较高的兼容性，适用于国产办公环境，其排版逻辑也基本一致。

版式固定类文档以 PDF 为代表，兼具文本与图像特性，能够完整保留原始版式，同时支持加密、批注等功能。根据内容形态不同，PDF 可分为可复制文本型与扫描图像型两类：前者在预处理阶段可直接进行文本抽取，后者则需要先完成图像增强处理，并结合 OCR 识别，才能获取其中的文字信息。演示类文档主要包括 PPT 与 PPTX，通常用于内容展示，内部由多页幻灯片及图文元素构成，并可能包含动画效果，预处理时需要按页面进行拆分，并对各页面内的组成元素进行识别与解析。

图像类输入主要包括 JPG、JPEG 与 PNG 格式，多为文档的扫描件或拍摄件，整体属于非结构化数据形态，不包含可直接编辑的文本信息。对此类文档的处理通常需先进行页面增强以提升图像质量，再结合版面分析与 OCR 等步骤，将其逐步转化为可处理的结构化内容。其中，JPG 与 JPEG 采用有损压缩方式，PNG 则支持无损压缩及透明背景，可分别适配不同清晰度与应用场景的需求。

2.2.2 文档预处理的核心任务

1. 页面增强

页面增强是文档预处理流程中的前置基础环节，位于原始文档数字化与版面分析之间。该环节面向扫描件、拍摄件等原始文档图像中普遍存在的质量缺陷，通过图像处理技术对页面进行优化与修复，以提升图像的清晰度、对比度及元素区分度，为后续版面分析准确识别页面元素的边界与类别奠定基础。其核心目标在于削弱噪声与干扰因素，尽可能还原文档页面的真实内容与排版结构，从而降低版面分析阶段的识别难度，保障下游处理流程的整体准确性[2]。

在实际采集过程中，原始文档图像往往受制于设备条件与拍摄环境，呈现出多种质量问题，因

此页面增强需针对不同缺陷开展相应的优化处理。常见技术手段包括降噪处理，用以消除斑点、杂色及扫描产生的颗粒感；倾斜校正，修正文档在扫描或拍摄过程中形成的角度偏差，保证页面整体水平；对比度与亮度调节，以强化文本、图表与背景之间的差异，改善模糊或昏暗图像的可识别性；以及边缘增强，通过突出页面中各类元素的轮廓，为后续版面分析阶段定位边界框提供更清晰的依据。

作为版面分析的前置保障环节，页面增强的处理质量在很大程度上决定了后续各步骤的整体效果。若该环节被忽略或过度简化，模糊、倾斜或噪声严重的图像容易导致版面分析过程中出现元素漏检、边界定位不准等问题，进而影响文字识别、表格理解等下游任务的处理精度。页面增强适用于多种原始文档场景，无论是老旧古籍、模糊票据，还是低分辨率扫描件，都需要通过该环节改善图像质量。作为连接文档采集与智能解析处理的关键枢纽，页面增强为文档预处理流程的稳定、高效推进提供了基础支撑。

2. 版面分析

版面分析是文档预处理流程中的核心基础环节。该过程借助计算机视觉与图像处理技术，对输入的数字化文档页面（如扫描件、图像化文档等）进行系统解析，识别页面中各类视觉元素的空间位置、边界范围及属性类别，并以边界框（bounding box）为主要载体输出元素的定位信息。其根本目标在于打破页面整体呈现的视觉统一性，将内容拆解为可区分的结构单元，为后续的文本识别、信息提取与格式还原等处理环节提供准确的空间与类别依据。

作为连接原始文档图像与结构化信息的关键桥梁，版面分析需要对多种页面元素进行精确区分与定位。具体而言，其识别对象包括文本相关组件（如正文段落、标题、目录）、页面辅助组件（如页眉、页脚、页码、批注）以及非文本组件（如图片、表格、公式）等。如图 2-1 所示，通过对各类元素边界框的定位，可以明确其在页面坐标系中的空间范围，理清元素之间的层级关系与相对位置，从而避免不同类型内容在后续处理过程中产生干扰。

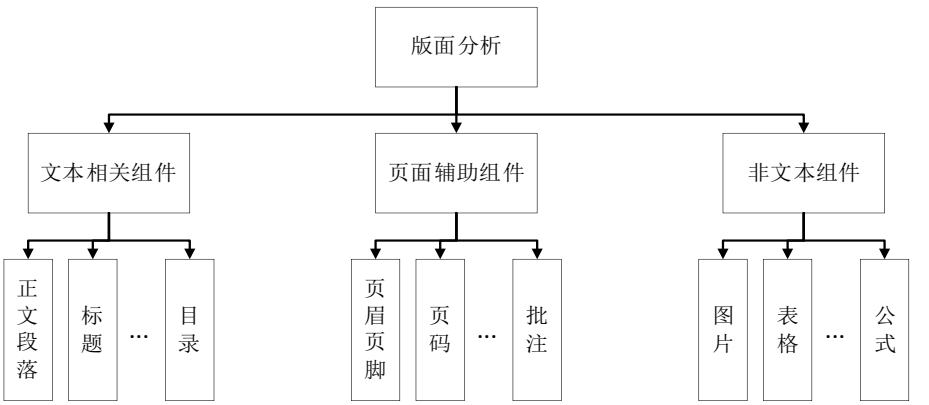

图 2-1　版面分析的定义

版面分析的质量直接影响后续文档处理的效率与准确性。若缺乏有效的版面分析，文本识别阶段可能混淆正文与页眉，或将公式误判为普通文本，信息提取也难以形成有序结果。经过规范的版面分析后，系统能够依据元素类别实施差异化处理，例如对文本区域执行 OCR 识别，对表格区域进行行列结构解析，对图片区域则保留其原始图像信息。该环节已广泛应用于古籍数字化、办公自动化与文档检索等场景，是实现文档智能化处理与信息高效利用的重要基础。

3. 阅读顺序预测

阅读顺序预测以版面分析输出的元素边界框、类别及空间位置信息为基础，结合人类阅读习惯与常见排版规范，通过算法逻辑推断页面内各类元素的合理阅读次序。其核心目标在于，将版面分析拆分得到的离散元素，重新组织为符合人类认知规律的有序内容序列，从而为文档内容的结构化呈现、语义理解及后续智能化处理提供必要的逻辑支撑。

在实际应用中，阅读顺序预测需针对不同排版形态的文档制定差异化规则。单栏文档是最基础的类型，其阅读顺序通常遵循自上而下、从左至右的线性逻辑，可依据元素在页面纵向坐标上的位置依次排序。同时，预测过程中应优先呈现正文段落，将页眉、页脚及批注等辅助性元素置于正文序列之外，或映射至其对应位置。

多栏文档（常见于期刊与报纸）则需在排序前明确栏区边界。阅读顺序通常遵循“先栏内、后栏间”的原则：在同一栏内沿用单栏文档的排序逻辑，完成一栏内容后，再按照从左至右的顺序切换至相邻栏区。

对于包含图表、公式或跨栏标题的复杂文档，阅读顺序预测更强调元素之间的语义关联关系。标题应置于其所统领的正文之前；图表及其标题需插入正文中对应的引用位置；公式则应紧随相关上下文文本出现。预测结果的准确性直接影响文档内容的可读性与语义连贯性，一旦出现排序偏差，极易导致信息逻辑混乱。

作为文档理解流程中的关键一环，阅读顺序预测广泛适配于多种文档场景，是实现古籍数字化、智能文档检索以及无障碍阅读等应用的重要前提。

4. 文字识别

文字识别（Optical Character Recognition，OCR）综合运用计算机视觉、模式识别与自然语言处理方法，将版面分析阶段定位出的文本区块图像转化为可编辑、可检索的结构化数字文本，其根本目标在于打破图像化文本的“视觉封装”，提取其中承载的语义信息，实现从物理文档图像向数字文本信息的跨形态转化，为后续的文档语义理解与信息抽取奠定基础[3]。

文字识别需严格依托上游环节的输出结果开展。系统首先依据版面分析标注的正文、标题、页眉等文本区块边界框，精准锁定识别对象，排除图片、表格等非文本元素的干扰；随后针对文本图像特征进行分层处理。在识别流程中，需对文本区块进行预处理，通过校正倾斜、降噪、增强对比度等手段优化图像质量，为后续特征提取创造条件；在此基础上，提取字符的形态、轮廓与笔画等关键信息，并通过算法与字符模型或字符库进行匹配，完成从单字符到词组、语句的识别与转化。

该环节需具备良好的场景适配能力，既能处理印刷体与手写体等不同字体形态，也需兼容中英文及多语种混合文本，同时支持单栏、多栏等多样化排版结构，并结合阅读顺序预测结果，按逻辑次序输出连贯文本。文字识别的精度直接影响文档数字化的整体质量，识别误差可能引发语义偏差甚至信息失真。正因如此，OCR 被广泛应用于办公自动化、票据识别等场景，成为连接文档物理形态与数字化应用的重要技术支撑。

5. 标题处理

标题处理过程是基于文字识别结果与版面分析信息，围绕文档中多级标题并存、标题层级事先不确定这一核心问题，通过规则与算法相结合的方式，完成标题识别、层级划分、语义关联与规范化整理。其目标是在杂乱的文本序列中准确剥离标题要素，理清各级标题之间的从属关系，构建清

晰稳定的文档层级结构，为后续的目录生成、内容分块与语义检索提供可靠的结构化基础。

标题处理的主要难点在于标题层级深度的不确定性与层级关系的复杂性，需要突破固定模板的限制，适配不同排版规范下的文档形态。处理过程中，首先结合文本特征（如字体大小、粗细、颜色）与位置特征（如段落间距、缩进形式），从正文序列中识别潜在的标题候选，并有效排除正文加粗文本、引文式小标题等干扰项。面对多级标题共存的情况，则需通过特征对比建立层级判定逻辑，例如依据字体层级与缩进幅度区分一级、二级、三级标题，同时兼顾跨栏标题、嵌套标题等特殊排版场景。

在实际应用中，标题处理需要在灵活性与准确性之间取得平衡。通过引入自适应算法，使系统能够根据不同文档的标题排版习惯动态调整层级判定规则，而非依赖预设的标题深度阈值。处理完成后，不仅可以明确各级标题之间的从属关系，还能够将标题与对应的正文内容进行有效关联，形成稳定的“标题-正文”结构化单元。该环节直接影响文档整体逻辑结构的梳理效率，是实现文档结构化存储、智能目录生成以及章节级信息提取的重要前提，广泛适用于电子书排版、学术论文处理等应用场景。

6. 表格理解

表格理解的核心目标在于服务下游的文档问答任务。该过程以版面分析阶段定位得到的表格边界框和文字识别结果为基础，通过技术手段完成表格结构识别与内容语义理解两项核心工作，将原本以图像和格式存在的表格信息转化为可检索、可问答的结构化数据。其本质在于打破表格的图像化与格式壁垒，使机器能够准确把握表格的组织逻辑与数据含义，从而为问答系统高效提取和利用表格信息提供支撑。

这一环节主要包含结构识别与内容理解两类核心任务，二者相互配合，共同解决表格数据在问答场景下的适配问题。结构识别侧重于表格形态的解析，通过算法识别表格的行列边界、单元格位置及其合并关系，将原始表格表示为HTML等结构化形式，明确单元格之间的从属关系与空间逻辑，构建机器可解析的表格框架，回答“表格是如何组织的”这一问题。内容理解则聚焦于语义层面的表达，在既定结构框架之上，以连贯的纯文本形式对表格的核心内容进行描述，包括行列含义、数据之间的关联关系以及潜在的关键信息与结论，从而将离散的单元格数据转化为具备整体语义的表达，解决“表格内容是什么”的问题[4]。

作为文档中重要的非文本数据载体，表格的理解质量直接影响文档问答系统的整体性能。若缺乏有效的表格理解能力，问答系统将难以准确检索表格数据或正确解读数据之间的关系。通过系统化的表格理解处理，下游任务能够更高效地定位表格中的关键信息，支持数据对比、关联分析等复杂查询需求。该环节适用于多种复杂表格场景，是连接文档预处理与文档问答的重要桥梁，为实现面向全文档、多类型数据的智能问答提供了关键支撑。

7. 图片理解

图片理解与表格理解并行，共同服务于下游的文档问答任务。该环节以版面分析阶段定位得到的图片边界框为基础，结合计算机视觉、图像识别与语义解析技术，针对文档中不同类型图片的特性，采用差异化的结构转化与信息抽取策略，将图像化信息转化为机器可检索、可解读、可用于问答响应的结构化内容或文本描述。

图片理解的核心目标在于打破图片“不可读”的语义壁垒，使机器能够准确把握图片所蕴含的

结构逻辑、数据内涵与表达意图，确保问答系统在面对涉及图片内容的查询时具备可靠的理解与响应能力，从而实现对全文档多元素的一体化问答支撑。

文档中的图片类型复杂多样，不同类型的图片在理解逻辑与处理目标上存在显著差异，需要针对性地制定技术路径。

对于流程图，其理解重点在于解析节点之间的关系与整体逻辑流向。通过图像识别技术提取流程节点、连接线条及标注文字等关键元素，并将其转化为 mermaid 代码等结构化描述语言，可较为完整地还原流程中的节点层级、因果关系与分支条件，使机器能够直接解析流程的执行路径与逻辑结构，为问答系统响应“流程步骤”“节点关联”等问题提供明确的结构化支撑。

针对统计图（包括柱状图、折线图、饼图、雷达图等），图片理解的核心在于数据信息的精准提取与语义化转化。通过图像分割与数据识别技术，提取坐标轴刻度、数据节点、图例说明及单位信息等关键要素，重建数据之间的对应关系、变化趋势与占比分布。同时，以规范、清晰的纯文本形式归纳图表结论，将可视化数据转化为可检索的结构化数据与语义描述，以满足问答系统在“数据对比”“趋势分析”“占比计算”等场景下的响应需求[5]。

除流程图与统计图之外，文档中还广泛存在示意图、实物图、截图、插画等其他类型图片。对此类图片的理解重点在于核心信息的提炼与语义描述。结合图片所处的上下文文本，识别其核心主题、关键元素、场景特征及其与正文内容之间的关联关系，并以简洁、准确的纯文本方式概括图片所表达的主要信息，明确其在文档中的功能与意义，从而支持问答系统对“图片展示内容”“图片与正文的关系”等基础问题的有效响应。

图片理解的完整性与准确性直接影响文档问答系统的覆盖范围与响应质量。作为连接文档预处理与智能问答的重要技术环节，它是实现非文本元素语义化检索与统一问答能力的关键基础。

8. 公式处理

公式处理是文档预处理流程中面向数学、物理等公式元素的专业化语义理解过程，与表格理解、图片理解等能力协同工作，共同服务于下游的文档问答任务。该环节以版面分析定位得到的公式边界框及文字识别结果为基础，结合公式识别与符号解析技术，将文档中的图像化公式转化为标准化、可计算的代码形式，其中核心输出为 LaTeX 表达式[6]。同时，公式处理需区分行内公式与行间公式两种场景，完成公式从图像形态向机器可解析、可检索、可问答的结构化形态转化。其核心目标在于破解公式图像“语义不可读”的问题，使机器能够准确捕捉公式中的符号逻辑与运算关系，从而支撑问答系统对公式相关问题的有效响应。

从任务层面看，公式处理主要包含公式形态转化与使用场景区分两个方面，二者共同决定该能力在实际系统中的可用性与适配性。在形态转化层面，需要通过算法识别公式中的符号、运算符、矩阵、分式、根式等关键元素，并准确还原其空间结构与运算关系，最终转化为通用的 LaTeX 代码。作为公式描述的标准语言，LaTeX 能够较为完整地保留公式的排版特征与语义内涵，为后续解析、检索与二次编辑提供基础支撑。

在场景区分层面，系统需准确判定公式属于行内还是行间形式。行内公式嵌入正文段落，与自然语言文本紧密结合，转化时需保留其与上下文语义的关联；行间公式则通常独立成行，承载关键推导或结论，转化后需明确标注其独立排版属性。公式处理精度直接影响问答系统对公式相关查询的响应能力，若缺乏有效的公式结构化处理，则系统将难以回答诸如“公式含义”“符号解释”等问题。因此，公式处理是文档预处理中覆盖全类型内容要素的重要补充，为实现包含公式文档的智

能问答与信息检索提供了关键的技术基础。

2.2.3　文档预处理的输出类型

文档预处理的输出类型主要以 Markdown 与 JSON 为主。二者面向不同的下游需求，在信息承载能力与结构灵活性上各有侧重，应结合具体任务场景加以选择。输出质量的优劣将直接影响后续文档问答、信息检索等任务的整体效果。

Markdown 作为一种轻量级标记语言，优势在于结构简洁、可读性强，编辑成本低且具备良好的跨平台兼容性，能够快速呈现文档的标题、段落、列表等基础结构。然而，其表达能力存在明显的局限，难以还原页眉页脚、版面元素的空间位置、跨行表格等关键结构化信息，因此更适用于对细节要求不高、结构需求相对简单的应用场景。

相比之下，JSON 是文档预处理阶段更为核心的输出形式。其突出特点在于高度的结构可定制性，可根据下游任务需求灵活设计字段，完整封装版面元素的位置关系、层级结构、表格组织方式以及公式代码等信息。JSON 不受固定模板约束，能够精确承载文档的全量结构数据，因而在文档问答等对结构理解要求较高的复杂任务中，成为兼顾信息完整性与任务适配性的优选输出类型。

2.3　主流文档预处理工具

在今天，智能体系统日益深入工作与生活场景，其感知能力不再局限于对语音、图像或传感器信号的简单接收，而是要求对复杂、非结构化的现实信息进行高保真、结构化的理解。其中，文档作为知识承载的核心载体，构成了智能体的关键输入源。

近年来，随着深度学习、计算机视觉与大模型技术的融合突破，文档预处理工具经历了从“通用 OCR”到“文档智能（DocumentAI）”的范式跃迁。一批兼具高精度、强健壮性与良好工程落地能力的开源或商业工具相继涌现，显著降低了智能体感知文档世界的门槛。

本节将系统梳理当前 5 款具有代表性的主流文档预处理工具：MinerU0[7]、PaddleOCR[8]、PP-StructureV3[9]、MonkeyOCR-1.5[10]与 DeepSeek-OCR[11]。它们分别代表了学术文献结构化、通用 OCR 工业化、复杂版面理解、边缘轻量化部署以及大模型驱动语义感知等不同技术路线与应用场景。通过对各工具的技术架构、核心能力、适用边界与部署特性的深入剖析，我们旨在为研究者与开发者提供一份清晰、实用的选型指南，助力构建更强大、更可靠的智能体感知系统。

2.3.1　MinerU

MinerU 是由国内 AI 初创公司 MinerLabs 于 2024 年初开源的 PDF 文档智能解析系统，专为解决科研工作者在处理 STEM（Science，Technology，Engineering，Mathematics，科学、技术、工程、数学）领域学术论文时面临的结构信息丢失问题而设计。传统 OCR 或 PDF 提取工具通常将页面内容转化为线性文本流，导致标题层级、数学公式、图表引用关系等关键语义结构不可恢复。MinerU 的核心目标是在保留原始排版语义的前提下，实现“所见即所得”的高保真文档还原。

MinerU 整体架构图如图 2-2 所示。

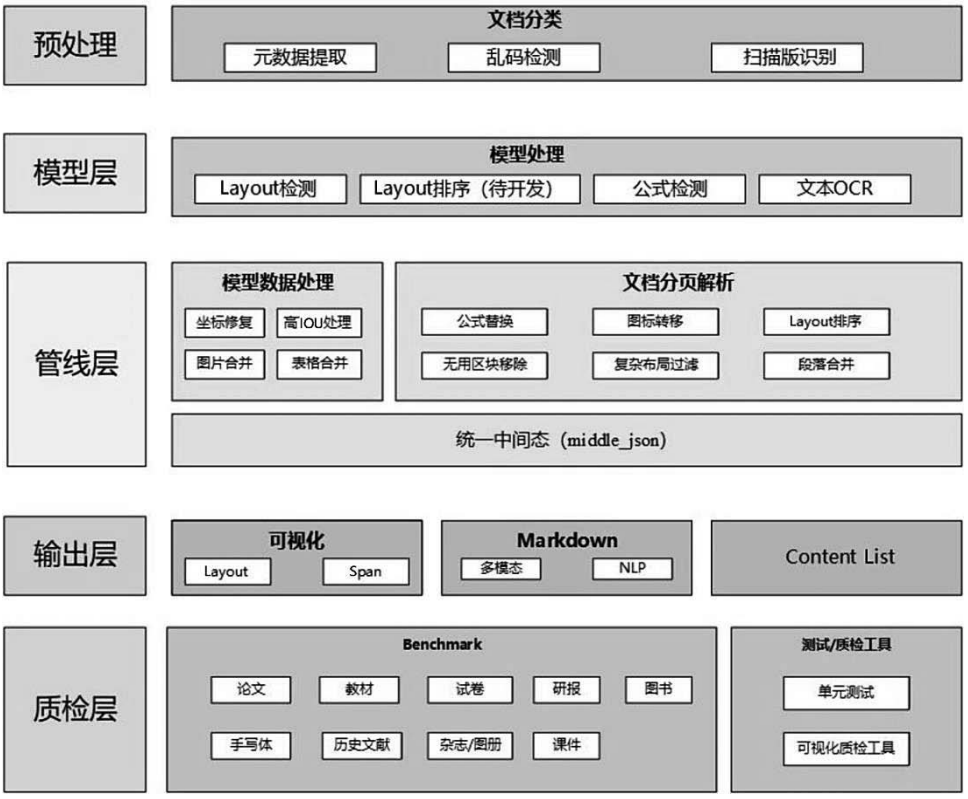

图 2-2　MinerU 整体架构图

MinerU 采用了分场景的任务处理策略。如果 PDF 本身带有可提取的文本和矢量信息（比如 arXiv 上的论文），可直接用 PyMuPDF 解析原生对象，避免不必要的图像转换。如果处理过程中遇到扫描件或纯图片页，则调用基于 YOLOv8 的视觉模型，先识别出文本块、表格、插图和数学公式的位置。部分 PDF 中出现的公式部分交给自研的 MathFormer 模型——这是一个基于 Transformer 的编码器-解码器结构，专门训练来把印刷体甚至部分手写风格的公式转换成 LaTeX，在 CROHME 等标准数据集上准确率接近 90%[12]。

这使得 MinerU 具有极强的 PDF 文档解析处理能力。首先便是其能稳定输出高质量的 LaTeX，以确保公式不“崩坏”。同时，在处理多栏布局时的效果也相当可观，它能够有效地识别双栏、三栏等典型论文排版，防止跨栏文本错误拼接。除此之外，它还可以做到保留正文元素间的引用关系，例如自动建立正文中的指代表述（如“如图 1 所示”）与对应图表标题之间的语义关联。它甚至还支持批量处理，提供了命令行工具 mineru-cli，可对整个文件夹内的 PDF 文档进行自动化结构化处理。

MinerU 通常适合高校、实验室、科研助手开发者和教育出版机构使用——无论是构建论文知识库、实现细粒度内容检索，还是在教材数字化中完整保留数学公式。但是对非学术文档（如合同、发票）优化不足，并且不支持手写中文或古籍竖排文本，同时社区规模较小，文档更新较慢。

2.3.2 PaddleOCR

PaddleOCR 是百度飞桨（PaddlePaddle）深度学习框架下的开源光学字符识别（Optical Character Recognition，OCR）项目，自 2020 年发布以来持续迭代，截至 2026 年已更新至 v3.5 版本。凭借完善的工具链、高效的模型设计和活跃的社区支持，该项目在 GitHub 上获得超过 35 000 颗星标，成为全球范围内广泛采用的 OCR 开源方案之一。其设计目标明确：兼顾工业部署的实用性与学术研究的先进性，提供一套开箱即用、可扩展性强的通用 OCR 能力。

PP-OCR 系列（v1~v4）是其核心产品线，版本迭代过程中逐步引入 DB++文本检测器、SVTR 识别网络以及 LCNet 轻量化骨干网络，在精度与速度之间取得了良好的平衡。同时支持多种语言，通过统一识别头+多语言字符集，覆盖 80 多种语言，包括藏文、维吾尔文等少数民族文字。PP-OCR 还内置文本方向分类器（Text Direction Classifier，CLS），可自动校正 0°、90°、180°和 270°四种常见的旋转角度，有效提升了复杂拍摄场景下的健壮性。同时支持动态词典机制，允许在推理阶段注入领域特定词汇（如车牌号、药品名称），显著提升业务场景下的识别准确率[8]。

PP-OCRv4 的核心优势在于通过一系列系统性的精细化改进，在维持高效推理速度的前提下，实现了多场景下识别精度的大幅提升。具体而言，相较于前代 PP-OCRv3，其在中文、英文数字及多语言场景的端到端 Hmean 或识别准确率分别显著提升超过 4.5%、6%与 8%。这一性能飞跃源于检测与识别模块共计 10 个关键技术的协同优化。在检测侧，其采用了并行分支融合的 PFHead 结构以增强特征表达能力，并升级了 CML 互学习策略与动态收缩比（Dynamic Shrinkage Ratio，DSR）训练方法，从而更精准地定位各类复杂文本。在识别侧，模型通过引入训练更稳定的 GTC-NRTR 注意力指导分支、高效的 DKD 蒸馏策略，以及创新的 DF 数据挖掘方案（该方案将训练周期从两周缩短至 5 天），大幅提升了文本识别的准确性与训练效率。尤为关键的是，这些改进均建立在专为边缘设备优化的 PP-LCNetV3 骨干网络之上，并通过详尽的消融实验审慎平衡了精度与速度的取舍，最终确保了模型在工业部署中兼具卓越的准确性与出色的实时性[8]。

PaddleOCR 已在物流（日均处理千万级快递面单以支撑自动化分拣与运单录入）、金融柜台服务（实时提取身份证、银行卡等证件信息以提升业务效率）以及智慧教育（在拍照搜题场景中精准抽取题目文本，为解题与内容推荐提供结构化输入）等多个高并发、高可靠性要求的领域实现规模化落地，但其默认模型仍存在若干局限：不包含版面分析能力（需额外集成 PP-Structure 模块），对高度密集表格、多语言混排或艺术字体文档的解析效果有限，且超轻量版本在低对比度、模糊或强光照干扰等复杂图像条件下易出现漏检问题。

2.3.3 PP-StructureV3

PP-Structure 是 PaddleOCR 官方推出的文档智能扩展模块，其 V3 版本于 2025 年第三季度发布，标志着飞桨 OCR 技术栈从单纯的“文本识别”向“完整文档理解”的战略演进。该模块不再仅关注字符内容的提取，而是致力于还原 PDF 或扫描文档中蕴含的逻辑结构与语义层次。

PP-StructureV3 能够将文档图像和 PDF 文件高效转换为结构化内容（如 Markdown 格式），并具备版面区域检测、表格识别、公式识别、图表理解以及多栏阅读顺序恢复等强大功能。该工具在多种文档类型下均表现优异，能够处理复杂的文档数据。PP-StructureV3 支持灵活的服务化部署，兼容多种硬件环境，并可通过多种编程语言进行调用。同时，支持二次开发，用户可以基于自有数据集进行模型训练和优化，训练后的模型可实现无缝集成[9]。

PP-StructureV3 整体架构流程图如图 2-3 所示。

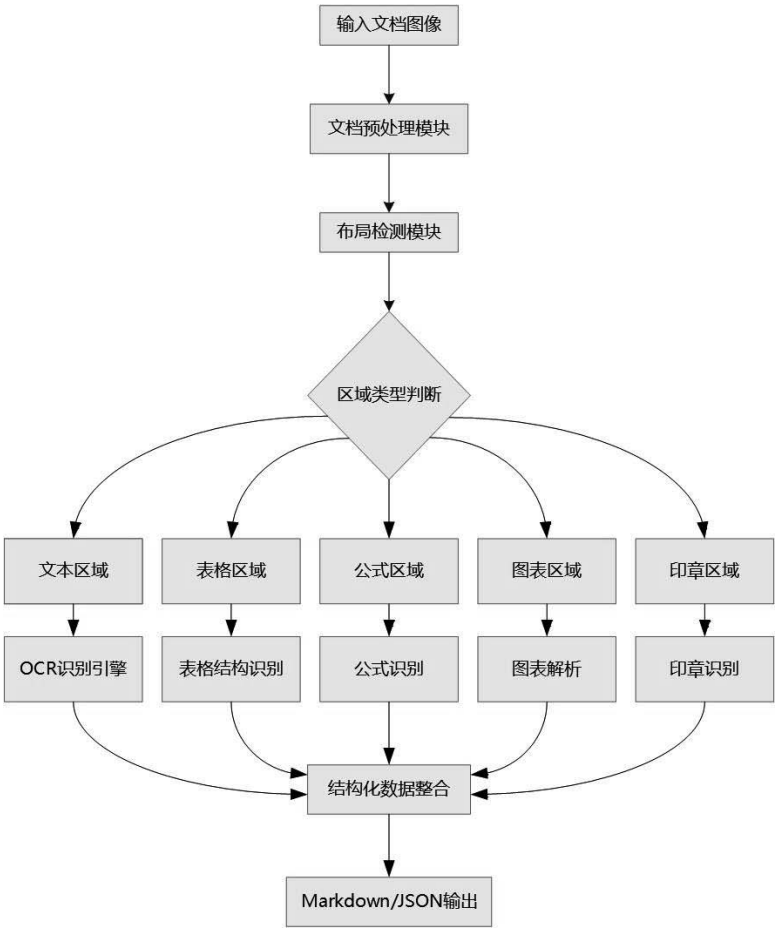

图 2-3　PP-StructureV3 整体架构流程图

在技术层面，PP-Structure 实现了多项关键技术突破。其内部核心的 TableMaster 模型[13]基于 Transformer 架构，支持端到端识别含任意合并单元格甚至跨页的复杂表格，并可输出 HTML 或 Excel 格式。LayoutParserV3[14]融合 DETR 与 U-Net 优势，可精准检测并分类 10 类常见的文档元素，包括文本段落、标题、图片、表格、数学公式、页眉页脚、列表、代码块及分隔线。系统的 PDF 原生解析器可以直接读取 PDF 内容流，区分矢量文本与嵌入图像，有效避免光栅化损失。PP-Structure 使用的语义重组引擎还提供了根据空间位置与字体特征重建段落、章节、列表层级的能力，增强识别效果。

系统支持多种结构化输出格式，包括 Word（.docx）、Excel（.xlsx）、Markdown、HTML，以及包含元素类型、边界框坐标、页面编号和原始内容的 JSON 格式（例如{"type":"table","bbox":[x1, y1,x2,y2],"html":"<table>...","page":3}）。

该工具适用于对文档结构完整性要求较高的专业场景，例如政府公文自动化归档（将红头文件转化为结构化数据库记录）、金融尽职调查报告解析（精准提取财报中的关键表格数据）以及法律文书智能分析（自动识别判决书中的当事人信息、案由与判决结果等核心要素）。

2.3.4　MonkeyOCR-1.5

MonkeyOCR 最初由智利创业公司 MonkeyLearn 开发，并在 2023 年被收购后开源。其 1.5 版本发布于 2024 年，专注于解决计算资源有限条件下的光学字符识别问题，特别适合 IoT 开发者、教育机构以及小型企业使用。

MonkeyOCR v1.5 的设计强调了轻量化与高效性的技术特点。文本检测模块基于改进版 EAST 模型，体积小于 1MB。在文本识别部分，采用了 CRNN 结合 CTC 损失函数，整体参数量控制在 200 万以内。同时，该系统还集成了 N-gram 语言模型和微调版 TinyBERT，能够修正诸如 cl1ent→client 类的常见拼写错误。值得一提的是，MonkeyOCR v1.5 仅依赖 OpenCV 和 PyTorch，无须复杂的环境配置，极大地简化了部署流程。

在性能方面，在专门测试复杂表格的 OCRFlux-Complex 数据集上，MonkeyOCR v1.5 的性能达到了 90.9，相比 MinerU2.5(81.7)和 PaddleOCR-VL(81.7)，实现了超过 9 个百分点的巨大提升。另外，在业内权威的文档解析基准 OmniDocBench v1.5 上，MonkeyOCR v1.5 的表现同样非常亮眼，它的综合得分达到了 92.9，超越了之前的 SOTA 模型 PPOCR-VL(91.9)和 MinerU 2.5(90.7)。值得注意的是，它也超过了像 Gemini 2.5-Pro 这样的通用闭源大模型，显示了领域专用模型在垂直任务上的巨大优势。这些都充分证明了其针对性设计的有效性[10]。

MonkeyOCR v1.5 在多种实际应用中表现出色，例如智能零售中的货架价签识别、农业物联网领域内的农药瓶标签读取，以及高校学生项目（如低成本文档扫描仪开发）等边缘计算或教学场景。MonkeyOCR v1.5 通过一系列精巧而实用的设计，特别是在处理复杂表格方面，为文档智能领域贡献了一个非常强大的新工具。它不仅在学术基准上取得了领先，也为解决工业界真实、复杂的文档解析需求提供了可靠的方案。

2.3.5　DeepSeek-OCR

2025 年，DeepSeek 推出 DeepSeek-OCR，标志着光学字符识别技术正经历一次范式转变：从传统的“感知层”字符提取，逐步演进为对文档内容的“认知级”语义理解。该系统的核心理念是，OCR 不应止步于还原文字，而应解析文档所承载的信息结构与业务含义。

DeepSeek-OCR 是一个多模态大语言模型 OCR 系统，其包含双编码器视觉系统（SAM-ViT-B 提取空间特征，CLIP-L 提取语义特征）、DeepSeek 语言模型、vLLM 推理引擎和多模态处理器等。该系统通过 MLP 投影器将视觉特征与语言模型融合，支持动态分辨率处理，在 A100-40G 上达到约 2500 tokens/秒的推理性能。

系统首先会根据图片的大小决定是否需要切分。如果图片太大，就智能地切成多个 640×640 的小块仔细分析，在保持每个小块合适的比例的同时，保留一幅 1024×1024 的全局视图以确保整体理解。

通过双编码器视觉系统的处理，系统获得了空间特征与语义特征，然后通过 MLP 投影器将这两种视觉特征融合成统一格式。同时，系统还会添加特殊的分隔符标记，从而帮助语言模型理解图像结构，最后按顺序排列所有特征。这就是 DeepSeek-OCR 的特征融合机制。系统还引入了 Prompt-driven 交互机制，允许用户通过自然语言指令引导识别过程，例如，“仅提取发票中的金额和开票日期”。另外，通过上下文感知纠错能力，其还能利用全文语义对局部识别结果进行校正（如将易混淆的“¥2O00”自动修正为“¥2000”），在识别过程中同步执行命名实体识别（NER）

与关系抽取（RE），可直接输出如“甲方：张三”“合同金额：¥150 000”等结构化三元组。最终系统会将处理好的视觉特征与文本指令结合，输入给 DeepSeek 语言模型进行最终的 OCR 识别和生成[11]。

DeepSeek-OCR 已在多个高价值领域展现出显著应用潜力，包括智能审计中自动比对合同条款与发票内容以识别不一致项，医疗 AI 辅助场景下从非结构化电子病历中精准提取诊断结论、检查结果及用药建议，以及法律科技（LegalTech）领域中实现合同条款的自动解析、跨版本比对与潜在风险点识别，其实际部署仍受限于较高的计算资源需求（难以适配边缘设备或轻量级应用）、开源版本功能的局限性（细粒度 Prompt 控制、行业定制命名实体识别等高级能力需商业授权），以及对非结构化手写文档（如医生手写处方、自由笔记等）的支持尚处于实验阶段，尚未达到生产环境可用的稳定性和准确率水平。

2.3.6 主流框架对比

表 2-1 总结了各工具在其相关技术报告、论文等公开资料中的数据对比[7][8][9][10][11]。

表2-1 5种主流文档预处理工具对比

工具对比维度	MinerU	PaddleOCR (PP-OCRv4)	PP-StructureV3	MonkeyOCR-1.5	DeepSeek-OCR
开源协议	Apache 2.0	Apache 2.0	Apache 2.0	MIT	基础模型开源（Apache 2.0）高级功能闭源
首次发布时间	2024 年 3 月	2020 年 6 月	2022 年 11 月（V3:2025.09）	2023 年 8 月（1.5:2024.11）	2025 年 7 月
支持语言数量	2（中、英）	80+	同 PaddleOCR	12（主流拉丁语系+中日韩）	5（中、英、日、韩、法）
中文识别准确率（ICDAR2013-CN）	-	96.3%	95.5%	76.1%	97.2%
英文识别准确率（IIIT-5K）	-	96.4%	96.0%	82.4%	98.1%
表格识别 F1-score（ICDAR2013Table）	-	不支持	94.5%	不支持	-
公式识别准确率（CROHME2019）	89.7%	不支持	83.2%	不支持	-
PDF 原生解析	支持（PyMuPDF+图像 fallback）	不支持（需转图）	支持（支持文本/图像混合页）	不支持	支持（基于 PDFium）
版面分析类别数	6（文本/标题/图/表/公式/列表）	0	10+（含页眉页脚、代码块等）	3+	8（含实体区域）
输出格式	Markdown&LaTeX	TXT/JSON/Line-level	DOCX/XLSX/HTML/MD/JSON	TXT	JSON（含实体+关系）

（续表）

工具 对比维度	MinerU	PaddleOCR (PP-OCRv4)	PP-StructureV3	MonkeyOCR-1.5	DeepSeek-OCR
单页处理时间（TeslaT4,PDF 含图）	～0.6s	～0.2s（纯文本页）	～1.2s	～1.0s（CPUi5）	～0.8s
最小部署资源	GPU ≥ 6GB 显存	CPU（轻量模型仅 3.5MB）	GPU≥8GB	CPU	GPU≥16GB（A10/A100 推荐）
是否支持微调	是（需标注 layout+formula）	支持（完整训练 pipeline）	支持	有限支持	支持（LoRA/全参数微调）
典型应用场景	学术论文结构化	通用 OCR（证件、票据）	政府公文、财报、合同	IoT、教育项目、边缘设备	金融、医疗、法律高精度场景

当前主流的文档预处理工具呈现出多样化、场景化的发展趋势，分别面向不同需求：MinerU 专注于学术论文的结构化解析，尤其在公式识别与多栏布局还原上表现突出；PaddleOCR 作为轻量、多语言的工业级 OCR 方案，适用于移动端与通用文本识别场景；PP-StructureV3 提供完整的版面分析与表格识别能力，适合公文、财报等复杂文档的结构化提取；MonkeyOCR-1.5 以极低资源占用支持边缘部署，适用于 IoT 与教育类轻量项目；而 DeepSeek-OCR 则依托大模型实现语义理解与指令交互，面向金融、医疗等高价值场景的智能信息抽取。这些工具共同推动了文档处理从“字符识别”到“结构理解”甚至“语义认知”的技术演进。

2.4　智能体的其他模态感知能力

智能体的环境感知能力已经从早期的单一模态处理发展到当前的多模态融合理解阶段。除了传统的文本感知外，音频、图片和视频作为三大关键模态，极大地扩展了智能体对现实世界的理解维度，使智能体能够更全面地理解物理世界，支撑从消费级应用到工业级场景的广泛落地[15]。这些模态各自携带独特信息：音频提供时序和事件线索，图像蕴含空间和视觉特征，视频则整合了时空动态关系。多模态感知的融合使智能体能够构建统一的环境表征，为后续的认知决策提供坚实基础。本节将深入探讨智能体在音频、图片和视频感知方面的关键技术、实现方法与应用场景。

2.4.1　音频感知

音频感知赋予智能体解析声音信号的能力，涵盖语音、环境音及音乐等非结构化数据，使智能体能够通过声音信号理解环境变化、用户意图和事件动态，是实现自然交互和情境意识的关键[16]。其核心在于将时序波形转化为语义信息，需解决噪声干扰、语种多样性及实时性等挑战。

音频感知的输入为音频原始波形信号（通常表示为一维时序序列$x(t)$，采样率fs为 16~48 kHz），输出为结构化语义标签，处理流程通常包含信号预处理、特征提取和高级语义理解三个核心阶段。在信号预处理阶段，原始音频波形经过降噪、分段和增强处理，信号质量得到提高。特征提取阶段

则从音频中提取有意义的表征，如梅尔频谱图、梅尔频率倒谱系数（MFCC）等时频特征[17]。高级语义理解阶段通过深度学习模型解析所得特征的语义内容。

随着深度学习被进一步探索，音频感知技术和应用得以不断发展，主要细分技术如表 2-2 所示。

表2-2 音频感知技术

细分技术	说　明
语音识别（ASR）	将人类语音转换为文本。技术已从传统 HMM-GMM 模型演进至端到端深度学习模型（如 Conformer-RNN-T、Transformers）
语音合成（TTS）	提供自然的听觉反馈。现代技术（如 VITS、Tacotron）已能实现高保真的端到端合成
音频事件检测（AED）	识别非语音的特定声音（如警报、门铃、玻璃破碎）。这对于智能安防和环境监测至关重要
说话人识别与确认	利用声纹特征提取解决“谁在说话”的问题，用于身份认证和多说话人区分
情感分析	分析语调、音高、节奏，推断说话人的情绪状态，增强人机交互的“人情味”
声源定位	利用麦克风阵列和信号处理算法（如波束成形）确定声音的方向和位置，辅助机器人听觉导航

在技术实现上，音频感知模型正从单一任务向多任务统一架构发展。例如，基于 Transformer 的音频编码器可同时处理语音识别、音频事件检测和情感分析等多类任务，通过共享表征学习提高效率并减少计算开销[18]。自监督学习技术在音频感知中也显示出巨大潜力，模型首先在大量无标注音频数据上进行预训练，学习通用的声学表征，再针对特定任务进行微调，显著降低对标注数据的依赖。

近期研究在音频感知技术上取得了显著突破。例如，本田研究院开发了一种能够预测未来声音的“听觉水晶球”系统，该技术基于流匹配（Flow Matching）方法[19]，使机器人能够通过分析当前音频信号来预测未来几秒钟的声音变化。这种预测能力让机器人可以提前规划动作，而非被动响应。系统采用三层架构：音频理解层将声波转换为频谱图；预测引擎层通过流匹配技术生成连贯的未来音频序列；行动决策层则将预测结果转换为机器人的控制指令。在本田研究院的实验中，机器人通过倾听倒水声音的音调变化，能精确判断瓶子何时将满，成功率达到 100%。音频感知的非接触感知优势在机器人控制领域具有独特价值，这一能力在视觉受限场景中尤为重要。

在服务机器人领域，音频感知与视觉、触觉融合形成多模态交互系统。例如，家庭服务机器人可通过语音指令理解用户需求，同时结合环境声音判断家庭状态（如识别水流声、脚步声等），提供更贴心的服务[15]。智能助理则可通过分析用户语音的音调、节奏和音量，推断情绪状态，实现更具情感智能的交互。

虽然音频感知具有强大环境感知能力和多模态技术融合潜力，但其发展仍然面临多项挑战。环境噪声干扰会显著影响感知性能，需要开发更强大的语音分离与增强技术。多声源场景下的声学场景理解尤为复杂，模型需要区分重叠对话、背景音乐和环境噪声等多种声源。此外，实际部署中也需考虑音频隐私保护，联邦学习等边缘计算技术可在本地对音频数据进行处理，减少敏感信息传输[15]。

2.4.2　图片感知

图像是信息最丰富的载体之一，能为智能体提供极其重要的外部世界输入。图像感知是指智能体通过摄像头、扫描仪等传感器获取图像数据，并运用计算机视觉技术进行分析、理解和解释，从而识别图像中的物体、场景、人脸，甚至推断图像背后蕴含的故事和语义信息的能力[15]。图像感知赋予智能体“看”世界的能力，使其能够识别物体、理解场景、提取文本信息以及感知空间关系，像人类一样“观察”和“理解”世界。随着深度学习技术的发展，图片感知已从简单的物体识别演进到对复杂场景的深度理解。

智能体的图片感知通常基于计算机视觉和深度学习技术，其核心是将像素级信息转化为语义级理解[22]。这一过程涉及多个层次的处理：底层视觉特征提取（如边缘、纹理和颜色）、物体检测与定位、场景分类以及高级语义理解。

单纯的图片感知能力虽然重要，但智能体真正价值的体现在于将视觉信息与其他模态信息融合。多模态融合策略使智能体能够结合图像内容和文本指令完成更复杂的推理任务。目前主流的多模态融合架构有几种典型模式：单模型+单模态预处理适合文本为主、图像为辅的场景；融合模型（如 LLaVA、MiniGPT）提供统一的图文理解能力；视觉模块+LLM 控制器的分体式设计则平衡了感知能力与控制灵活性[15]。

图片感知和多模态感知能力融合共同构筑了智能体洞察和分析视觉信息的强大体系，使其能够实现图像分类、目标检测、目标分割、人脸识别与分析和光学字符识别等多类任务。

图片感知技术及其相关智能体应用如图 2-4 所示。

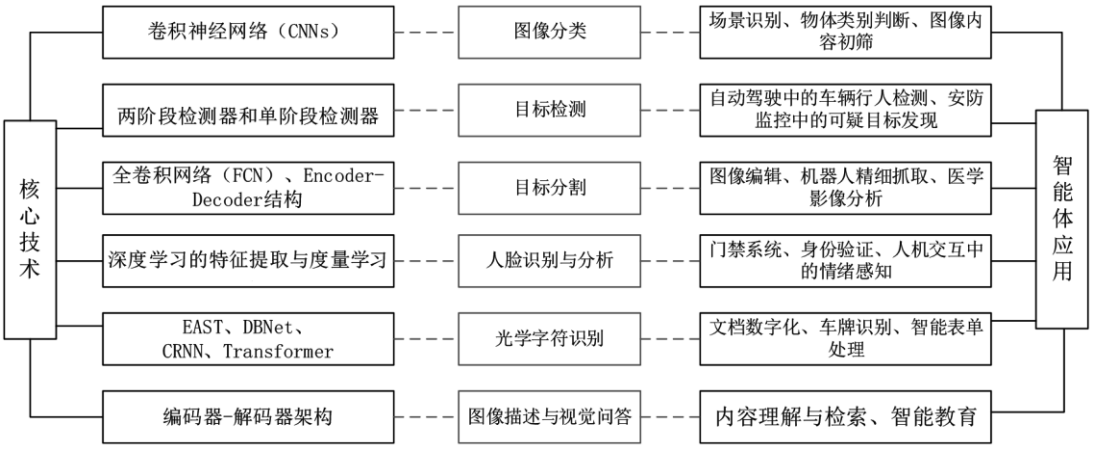

图 2-4　图片感知技术及其相关智能体应用

图像分类作为图片感知的基础技术，使智能体能够识别图像中的主要内容类别，如区分猫、狗或汽车。卷积神经网络（Convolutional Neural Network，CNN）及其不断演进的模型（如 LeNet、AlexNet、ResNet、EfficientNet）为图像分类提供了强大支撑[20]。近年来，Transformer 模型的引入也极大地拓展了视觉感知的边界。这些技术让智能体具备了场景识别、物体类别判断和内容初筛等基本视觉能力。

在此基础上，目标检测技术让智能体不仅能识别物体，还能准确定位其在图像中的位置。无论是两阶段检测器还是单阶段检测器，再到基于 Transformer 的新型检测器[21]，都极大地提升了智能体对复杂场景的理解能力。这些能力在自动驾驶、安防监控和智能零售等实际应用中至关重要。

为了实现更精细的视觉感知，智能体还应用到了目标分割技术。其中语义分割让智能体能够对每个像素进行分类，实例分割则进一步区分同类物体的不同个体。基于 FCN、U-Net 和 Mask R-CNN 等模型，智能体能够在图像编辑、机器人操作、自动驾驶和医学影像等领域实现高精度的像素级理解。

在人脸识别与分析方面，图片感知能力不仅能使智能体识别人脸身份，还能分析表情、年龄、性别等属性。基于深度学习的特征提取与度量学习结合活体检测技术，显著提升了智能体在门禁、身份验证和情绪感知等场景下的安全性和交互性。

同时，图片感知能力还使智能体具备将视觉信息与文本信息相互转化的能力[15]。光学字符识别（OCR）使其能够从图像中提取和识别文本，广泛应用于文档数字化、车牌识别和信息提取等场景。图像描述与视觉问答（VQA）则实现了视觉与语言的深度融合，使智能体能够自动生成图像描述或回答与图像相关的问题，为视觉辅助、内容检索和智能教育等领域提供了有力支持。

虽然智能体的图片感知的研究已取得诸多突破性进展，但其在实际应用中依然面临一系列严峻挑战，这些挑战限制了智能体在复杂真实世界中的部署和性能。环境健壮性始终是影响图像识别系统性能的关键因素[22]。现实世界中的光照变化、物体遮挡、图像模糊、复杂背景以及极端天气等环境条件，都会对模型的识别准确性和稳定性造成显著影响。尽管深度学习方法在标准数据集上取得了突破性进展，但在真实复杂环境下，模型的泛化能力和健壮性仍然有限[22]。小样本与长尾问题同样困扰着图像感知系统。许多特定场景或罕见物体的数据集极为稀缺，导致模型难以有效学习和泛化到这些类别。现实世界中存在大量长尾分布的数据，主流类别样本充足，而大量小众类别样本极少，这使得模型在面对长尾类别时表现不佳，难以实现全面的视觉理解。实时性要求也是图像感知技术落地的重大挑战之一。自动驾驶、实时监控等应用场景对图像处理的速度和延迟有极高的要求，既要保证高精度识别，又要满足毫秒级的响应速度，这对算法的计算效率和硬件资源提出了更高的要求。

展望未来，图像感知技术的发展将呈现出多元化和智能化的趋势。首先，视觉基础大模型的兴起为图像感知任务带来了新的突破。通过在海量图像数据上进行预训练，视觉大模型（如 ViT、MAE、CLIP、DALL-E 等）能够学习到通用的视觉表示，极大地提升了模型的泛化能力和下游任务的表现。与此同时，3D 视觉与空间理解成为推动机器人和 AR/VR 等领域发展的关键。通过结合点云、深度图和多视角图像，智能体能够实现更精确的三维空间建模和理解，从而更好地适应和操作真实物理世界。自监督与半监督学习方法的不断进步，也为减少对大量人工标注数据的依赖提供了可能，使得模型能够利用无标签或少量标签数据进行有效学习，提升了数据利用效率和模型的适应性。

2.4.3 视频感知

视频是真实世界最丰富、最接近人类感知经验的模态，与静态图像不同，视频由连续帧序列构成，包含丰富的时序信息和动态演化模式。人类在日常生活中主要通过视觉（包括动态视觉）来理解周围环境。赋予智能体视频感知能力，是其理解动态环境、预测未来行为、进行复杂任务决策的关键[23]。而视频感知是智能体多模态感知能力中最复杂且最具挑战性的方向，它是图像感知的自然延伸，综合了图片感知的空间理解能力和音频感知的时间序列分析能力，使智能体能够理解动态场景中的时空变化，从静态的“看”升级为动态的“观察”和“理解”，为智能体预测未来事件、理解因果关系提供了可能。视频感知能力是智能体在智能监控、运动分析、自动驾驶、虚拟现实、内容理解与生成等领域不可或缺的核心能力。

在视频感知领域，智能体需要解决的核心问题是如何有效地从海量的时序图像帧中提取并理解有意义的时空信息。这不仅仅是识别单帧画面中的物体，更在于理解物体之间的动态交互、行为的发生发展过程以及事件的因果关系。

时空特征编码是视频感知的基石。它旨在捕捉视频中空间信息和时间信息的联合特性。光流（Optic Flow）技术通过计算图像序列中像素点的运动矢量场，能对图像中物体的运动轨迹进行精准量化。例如，最新的光流估计算法如 Recurrent All-Pairs Field Transforms（RAFT）通过引入循环神经网络和注意力机制，能够以极高的精度和效率计算密集光流[24]，甚至在消费级 GPU 上也能支持每秒 120 帧的实时处理，这对于自动驾驶、机器人导航等对实时性要求极高的应用至关重要。多视角融合技术的发展支持智能体获取更真实的三维运动信息。通过立体视觉原理，智能体可以利用双目或多目摄像头捕捉图像数据，精确地恢复出场景中物体和自身的深度信息以及三维运动轨迹。例如，在机器人抓取、增强现实（AR）或虚拟现实（VR）中，利用多视角融合技术可以将三维运动轨迹误差控制在 3 厘米以内，从而实现对物理世界的精准操作和交互。

行为语义解析是视频感知实现高级应用的关键，其要求智能体不仅要能识别出视频中的运动，更要理解这些运动背后的深层语义和人类意图。时序动作分割旨在将复杂的连续性动作分解为一系列有意义的基本动作单元，从而实现对长时程行为的精细化理解和分析。基于 Transformer 架构的模型，如 Informer 及其变体，已被应用于无监督的动作单元划分，通过学习动作的时序模式和内在结构，智能体能够自动发现并区分动作的边界和类型[25]。在此基础上，意图识别进一步提升了智能体的认知水平。通过构建复杂的行为–文本对齐模型，智能体能够从监控视频中提取出更高级的语义标签，例如识别出“异常徘徊”“物品遗留”“非法入侵”等可能预示风险的意图或事件。这种能力对于智能安防、公共安全以及行为分析等领域具有极其重要的价值。

为应对视频数据庞大、处理复杂带来的挑战，实时处理优化是实现视频感知广泛应用的关键。由于许多应用场景对延迟要求极高，模型的计算效率显得尤为重要。边缘计算加速为此提供了有效途径，通过将计算任务从云端转移到靠近数据源的边缘设备上，延迟显著减少，响应速度得到了提高。例如，采用 NVIDIA TensorRT 等工具对深度学习模型进行量化和优化，可以大幅提升推理速度。实际测试表明，在 NVIDIA Jetson Orin 等嵌入式平台上，经过优化的 YOLOv8 目标检测模型可以实现高达 150 帧每秒的推理速度，这使得实时目标检测成为可能。为了使智能体能够持续学习和适应新的环境或任务而无须完全重新训练，增量式学习（或称持续学习）技术的重要性得以提高。传统的深度学习模型在学习新任务时往往会“遗忘”旧任务的知识（即灾难性遗忘）。通过引入如 EWC（Elastic Weight Consolidation，弹性权重巩固）算法，模型能够在动态更新参数以适应新数据时，有选择性地保留对旧任务重要的权重，从而有效地避免灾难性遗忘，使智能体能够进行在线、持续的视频内容学习和适应[26]。

图 2-5 所示为智能体视频感知应用。

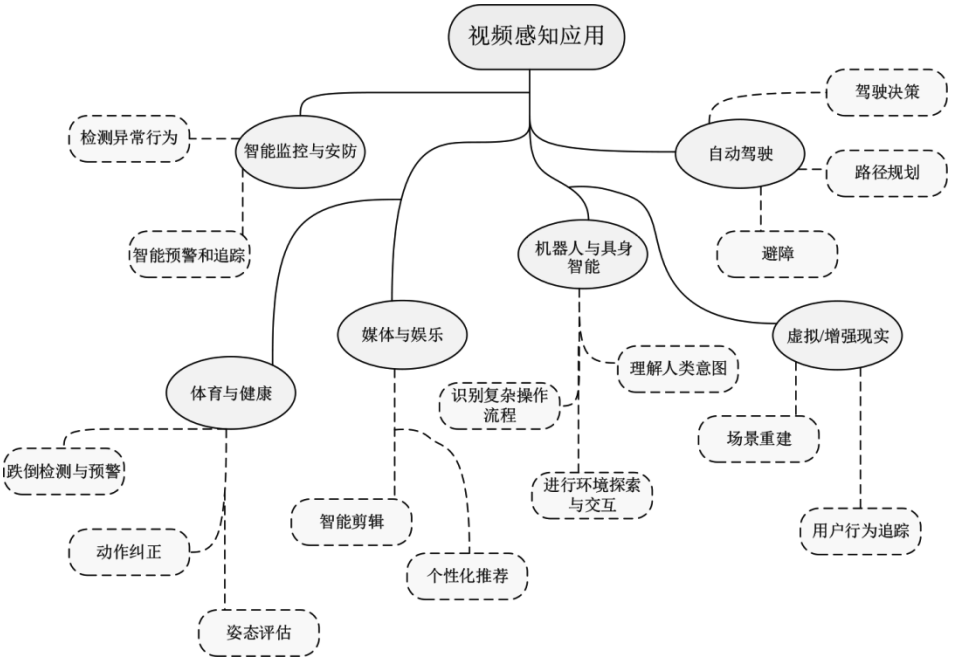

图 2-5　智能体视频感知应用

视频感知技术已在众多实际场景中得到广泛应用。在智能监控与安防领域，智能体能够实时检测异常行为、闯入、摔倒、遗留物等事件，进行智能预警和追踪，极大地提升了公共安全水平。在自动驾驶领域，视频感知使智能体能够感知周围车辆、行人、骑行者及交通状况的动态变化，预测其行为，辅助驾驶决策，保障行车安全。在机器人与具身智能领域，视频感知帮助机器人理解人类意图、识别复杂操作流程，实现环境探索与人机协作。在体育与健康领域，运动员动作分析与评估、健身指导、老年人跌倒检测与预警等应用不断涌现。在媒体与娱乐行业，视频内容理解、智能剪辑、个性化推荐、虚拟角色动画等创新应用层出不穷。在虚拟现实/增强现实领域，视频感知支持实时场景重建、用户行为追踪和环境交互感知，为沉浸式体验提供技术保障。

视频感知技术将沿着多条主线持续发展。视频基础智能体的兴起将推动视频理解、检索和生成能力的全面提升。通过在海量视频−文本数据上预训练的大规模模型，如 Google 的 Flamingo、Meta 的 Ego-Exo4D 等，智能体能够实现更通用、更强大的多任务视频理解。情景感知与预测能力也将成为智能体的重要特征，智能体不仅能识别当前行为，还能基于对历史和情境的理解，预测未来事件的发展趋势和潜在风险。人机协作与意图预测也将不断深化，通过视频理解人类的意图和操作步骤，实现与人类的更自然、高效的协作。高效压缩与传输技术将优化视频数据的采集、编码、传输和处理效率，降低系统资源消耗。视频感知与具身智能的深度融合将成为趋势，视频感知与机器人本体感知（如触觉、惯性）和动作执行形成闭环，实现更高级的物理交互和环境适应。随着深度学习、跨模态融合、边缘计算等技术的不断突破，视频感知将在智能体领域发挥越来越核心的作用，助力智能体实现更自然、更智能、更可信的人机交互。

2.5　本章小结

本章围绕“智能体感知能力”这一核心主题，系统梳理了文档感知在智能体体系中的定位、任务结构与技术实现路径，并进一步拓展到多模态感知能力的整体框架。从总体上看，感知能力是智能体理解外部世界、获取可靠输入信息的起点，其质量直接决定了后续认知、推理与决策能力的上限。

在文档感知部分，本章首先明确了文档感知任务的定义与分类，强调文档并非单一文本形态，而是由文本、表格、图片、公式等多种异构元素共同构成的复杂信息载体。围绕这一特性，文档预处理被拆解为多个相互协同的核心任务，包括页面增强、版面分析、阅读顺序预测、文字识别、标题处理、表格理解、图片理解与公式处理等。这些环节从“视觉结构解析”到“语义结构化表达”逐层递进，共同完成文档从图像化输入到机器可理解、可检索、可问答结构化信息的转化，为下游文档问答、信息抽取与知识构建奠定坚实基础。

在工具与框架层面，本章对当前主流文档预处理工具进行了概览，涵盖 MinerU、PaddleOCR、等代表性方案，并从能力覆盖、工程成熟度与适用场景等维度进行了对比分析。可以看到，不同工具在识别精度、多元素协同处理能力及扩展性方面各有侧重，实际应用中需结合业务需求与系统架构进行取舍与组合。

最后，本章将视角从文档感知扩展至音频、图像、视频等其他模态感知能力，强调多模态协同是智能体迈向更高层次理解与交互的关键趋势。总体而言，感知能力并非孤立模块，而是贯穿智能体全流程的基础能力体系，其持续演进将直接推动智能体在复杂真实场景中的应用深度与广度。

2.6　参考文献

[1]　Durante Z, Huang Q, Wake N, et al. Agent ai: Surveying the horizons of multimodal interaction [EB/OL]. (2024-01-08)[2026-06-23]. arXiv preprint arXiv:2401.03568, 2024.

[2]　Anvari Z, Athitsos V. A survey on deep learning based document image enhancement[EB/OL]. (2021-12-05)[2026-06-23]. arXiv preprint arXiv:2112.02719, 2021.

[3]　Subramani N, Matton A, Greaves M, et al. A survey of deep learning approaches for ocr and document understanding[EB/OL]. (2020-11-26)[2026-06-23]. arXiv preprint arXiv:2011.13534, 2020.

[4]　Liu T, Wang K, Sha L, et al. Table-to-text generation by structure-aware seq2seq learning[C]// Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1): 478-485.

[5]　Farahani A M, Adibi P, Ehsani M S, et al. Automatic chart understanding: a review[J]. IEEE Access, 2023, 11: 76202-76221.

[6]　Deng Y, Kanervisto A, Ling J, et al. Image-to-markup generation with coarse-to-fine attention[C] //International Conference on Machine Learning. Sydney: PMLR, 2017: 980-989.

[7]　MinerLabs. MinerU [EB/OL]. (2024-03)[2025-06-01].https://opendatalab.github.io

[8]　PaddlePaddle. PP-OCRv4[EB/OL]. PaddleOCR, (2023-01-01)[2026-01-26]. https://www.paddleocr.ai/v2.9/ppocr/blog/PP-OCRv4_introduction.html.

[9] 百度飞桨（PaddlePaddle）. PP-StructureV3 [EB/OL]. (2025-06-18)[2026-01-26].
https://www.paddleocr.ai/main/version3.x/algorithm/PP-StructureV3/PP-StructureV3.html

[10] Zhang J, Liu Y, Wu Z, et al. MonkeyOCR v1. 5 Technical Report: Unlocking Robust Document Parsing for Complex Patterns[EB/OL]. (2025-11-16)[2026-06-23]. arXiv preprint arXiv:2511.10390, 2025.

[11] 深度求索（DeepSeek）. Deepseek-OCR [EB/OL]. (2025-07)[2026-01-26].
https://github.com/deepseek-ai/DeepSeek-OCR/

[12] Devvrit F, Kudugunta S, Kusupati A, et al. Mathformer: Nested transformer for elastic inference[J]. Advances in Neural Information Processing Systems, 2024, 37: 140535-140564.

[13] Cao L, Liu H. Tablemaster: A recipe to advance table understanding with language models[EB/OL]. (2025-01-31)[2026-06-23]. arXiv preprint arXiv:2501.19378, 2025.

[14] 百度飞桨（PaddlePaddle）. LayoutParserV3 [EB/OL]. (2025-09)[2026-01-26].
https://layout-parser.github.io

[15] 赵博涛， 亢祖衡， 瞿晓阳， 等. 基于多模态大模型的具身智能体研究进展与展望[J]. 大数据， 2025, 11(3): 108-138.

[16] 王红. 空间音频：从技术路径到元宇宙应用，如何突破挑战抵达未来？[J]. 科普中国, 2025.

[17] PURWINS H, LI B, VIRTANEN T, et al. Deep learning for audio signal processing[J]. IEEE Journal of Selected Topics in Signal Processing, 2019, 13(2): 206-219.

[18] GONG Y, CHUNG Y A, GLASS J. AST: Audio Spectrogram Transformer[C]//Interspeech 2021. Brno: ISCA, 2021: 571-575.

[19] LIPMAN Y, CHEN R T Q, BEN-HAMU H, et al. Flow Matching for Generative Modeling[C]// The Eleventh International Conference on Learning Representations. Kigali: ICLR, 2023.

[20] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016: 770-778.

[21] CARION N, MASSA F, SYNNAEVE G, et al. End-to-End Object Detection with Transformers [C]// Computer Vision – ECCV 2020. Cham: Springer, 2020: 213-229.

[22] LI X, WANG J, ZHANG L. Organic adaptive transistors with wide-color-gamut for machine vision applications[J]. Nature Photonics, 2026, 20(1): 45-52.

[23] BALTRUŠAITIS T, AHUJA C, MORENCY L P. Multimodal Machine Learning: A Survey and Taxonomy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(2): 423-443.

[24] TEED Z, DENG J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow[C]//Computer Vision – ECCV 2020. Cham: Springer, 2020: 402-419.

[25] ZHOU H, ZHANG S, PENG J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(12): 11106-11115.

[26] KIRKPATRICK J, PASCANU R, RABINOWITZ N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, 2017, 114(13): 3521-3526.

第 3 章

智能体记忆能力

随着大语言模型从单轮交互工具逐步演进为具备自主逻辑的长期智能体，记忆能力已成为支撑系统健壮性与任务连贯性的核心要素。缺乏记忆机制的智能体不仅难以保持人格一致性，更无法在多轮交互中沉淀并复用经验，这直接制约了其在复杂动态场景下的实际应用价值。

从工程视角来看，智能体记忆并非简单的静态数据存储，而是一项融合了跨时间步的状态建模、实时信息调度及计算资源配比的系统化能力。尽管 LLM 的上下文窗口为短期推理提供了高带宽的即时信息获取，但面对长程任务及海量多模态输入，单一的窗口机制在一致性维护与资源成本上已达到瓶颈。正因为上下文窗口的限制与成本问题，记忆外挂与分层存储架构应运而生。

本章将系统拆解智能体记忆的逻辑边界与技术内涵，重点探讨分层记忆架构的演进逻辑及其在生产环境中的落地范式。通过对记忆写入、调度及反思等核心机制的剖析，为后续章节构建具备高度自主进化能力的智能体系统奠定理论与工程基础。

3.1 记忆能力的定义与内涵

在基于大语言模型的智能体系统中，“记忆”并非传统意义上的数据存储或对话日志，而是一类参与决策过程的、跨时间演化的内部状态机制。其核心作用在于：在模型参数固定、上下文窗口受限的条件下，使智能体能够在部分可观测环境中维持行为连贯性，并在多轮甚至跨任务交互中复用历史经验[1]。

在第 2 章中，我们已经讨论了智能体的感知能力，即智能体如何通过多模态感知机制，将外部环境中的信号转换为内部可处理的表示形式。然而，从计算视角看，智能体若仅依赖当前输入与有限上下文，其决策过程本质上仍属于“瞬时反应式系统（Reactive System）”。这类系统虽然能够在单轮推理中展现较强的语言理解与生成能力，但在长程任务、持续交互或环境状态随时间演化的场景中，往往难以保持稳定的目标追踪、角色一致性与策略延续性。记忆机制的引入正是为了解决这一跨时间推理能力不足的问题[2]。

本节将智能体记忆能力界定为：在有限的计算资源与上下文窗口双重约束下，智能体对历史观测、内部状态及交互结果进行跨时间建模、选择性保存、结构化组织与按需调度的系统能力。需要强调的是，该定义并不预设记忆的具体物理实现形式（如向量数据库或图结构），而是聚焦于记忆在决策流程中的功能角色。基于这一视角，智能体记忆至少应满足以下三个计算层面的基本特征。

- 动态性：记忆状态随时间持续更新，而非静态累积的历史快照。
- 决策性：记忆必须深度参与当前的决策推理过程，而非仅作为外部查询库。
- 计算性：记忆的写入、检索与压缩过程可被形式化描述，并纳入系统设计与资源评估。

正是这种跨时间维度的认知调度能力，使得智能体能够在长程交互中，保持身份一致性（Identity Consistency）、任务连贯性（Task Continuity）以及经验的可复用性（Experience Reusability）。

3.1.1 记忆的形式化定义

为了在计算层面明确记忆在智能体系统中的角色，有必要超越“历史缓存”或“数据库”的直观隐喻，从认知心理学的动态过程出发，将其抽象为可描述、可计算的工程模型。

1. 认知本质：从静态存储到动态生命周期

在认知心理学中，记忆并非被动的信息堆积，而是一个由编码（Encoding）、保持（Storage）、检索（Retrieval）与遗忘（Forgetting）构成的动态闭环[3]。这一经典模型对智能体系统的核心启示在于：记忆本质上是一种受限资源的生命周期管理问题。

若不引入这种动态视角，仅将向量检索模块（RAG）视为记忆，系统往往会退化为“无差别的堆积池”。随着交互轮数增加，检索噪声将随之放大，模型幻觉（Hallucination）的发生概率也会显著提升[4]。因此，在系统设计中是否显式引入“遗忘”与“整理”机制，往往是区分实验性 Demo 与生产级长效系统的关键基准。

2. 工程映射：认知环节的组件化

在智能体架构中，上述心理学过程被具体映射为一组可实现的工程组件，由此构建起记忆系统的功能分层体系。

- 编码（Encoding）——表征生成模块：负责将感知到的非结构化信息（如文本、视觉信号）转化为内部可计算的表示，如嵌入向量或知识图谱。
- 保持（Storage）——多级存储层：旨在不同时间尺度上维护状态，涵盖从短期工作记忆（Context Window）到长期持久化存储（Vector DB/Graph DB）的各类实现形式[5]。
- 检索（Retrieval）——召回与路由策略：根据当前决策需求，通过 Top-k 相似度计算或条件过滤，从海量历史中模拟人的“注意力”机制，精准提取最相关的上下文。
- 遗忘（Forgetting）——压缩与清理机制：通过摘要总结（Summarization）、重要性评分或先进先出（FIFO）策略，主动剔除冗余、陈旧的信息，以保障系统的推理高效性与准确度。

3. 数学形式化：记忆作为隐状态变量

基于上述认知与工程理解，我们可以在数学层面将智能体记忆形式化为部分可观测马尔可夫决策过程（POMDP）中的隐状态（Hidden State）。通常，智能体与环境的交互可定义为元组

$A=(O,S,A,T,\pi)$。在无显式记忆的“瞬时反应”系统中，策略函数 π 仅依赖当前观测 o_t 和有限的上下文 c_t：

$$a_t=\pi(o_t,c_t)$$

而在引入记忆模块后，系统引入了一个跨时间步演化的记忆状态 M_t。此时，智能体的决策过程被扩展为包含记忆更新（Write）与记忆读取（Read）的双重交互过程。

记忆更新（Write/Update）：记忆状态 M_t 是上一时刻记忆、当前观测及上一个动作的函数。此处的 f_{write} 函数即对应工程中的编码与遗忘（压缩）机制：

$$M_t=f_{\text{write}}(M_{t-1},o_t,a_{t-1})$$

决策生成（Read/Act）：策略函数的输入不再局限于当前观测，而是包含通过检索函数从记忆中提取的增强上下文。此处的 Retrieve 函数即对应基于当前观测的记忆召回与筛选策略：

$$a_t=\pi(o_t,g(o_t,\text{Retrieve}(M_{t-1})))$$

通过这一形式化定义，我们可以明确：智能体记忆 M_t 是一个随时间演化、并通过读写接口影响当前决策的隐状态变量。它决定的不仅是模型能否“记住”过去，更关乎系统能否在长序列交互中维持状态的稳定性与推理的连贯性。

3.1.2　记忆与上下文窗口的关系

随着大语言模型（LLM）技术的演进，上下文窗口（Context Window）的长度呈现出指数级增长态势，这一技术突破带来的一个常见的工程直觉是：只要上下文足够长，显式记忆机制便不再必要。然而，从智能体系统的角度看，上下文窗口与记忆机制在计算属性、生命周期与资源开销上存在本质差异，二者更接近互补关系，而非替代关系。

上下文窗口是一种高带宽、短生命周期的瞬时计算资源，其内容与单次推理过程强绑定，在推理结束后被直接释放。相比之下，记忆机制是一种低带宽、跨推理周期持久存在的状态管理手段，能够在不同任务与会话之间稳定保留信息。在访问方式上，上下文窗口允许模型对其中的全部内容进行全注意力计算，而记忆系统通常依赖索引式检索，仅在需要时加载相关的有限片段[2]。

其次，尽管“全上下文”策略在理论上具备可行性，但工程实践表明，当系统尝试通过“单纯塞进更多上下文”来解决长期一致性问题时，往往会遇到以下瓶颈[2]：

- 推理成本不可控。
- 注意力分散导致关键信息反而被弱化甚至忽略，如“中段遗忘”[6]。
- 由于上下文在进程结束后即释放，系统无法跨任务、跨周期复用已有经验。

因此，长上下文并未消除记忆系统的需求，而只是延缓了“状态外溢”的发生时间。真正决定智能体是否具备长期一致性的并非上下文长度本身，而是系统是否存在可靠的跨推理周期的状态持久化与调度机制。

在实际系统中，判定是否引入显式记忆模块有一个简单实用的准则：你的 Agent 是否需要“明天还记得今天发生了什么”。一旦涉及跨日、跨会话的逻辑连贯，仅依赖上下文窗口在工程上几乎不可行。当前主流智能体系统普遍采用“上下文用于当前推理，记忆用于跨时管理”的分工策略，

并通过外部记忆模块在上下文受限条件下实现长期信息保留[5]。这一假设直接构成后续三层记忆架构设计的理论基石。

3.1.3 多模态记忆：表征与对齐

随着智能体逐步进入多模态交互场景，加之 GPT-4V、Gemini 等多模态大模型（Large Multimodal Models，LMM）的引入，其记忆对象已从纯文本扩展至视觉（Visual）、听觉（Audio）甚至物理环境状态等多模态空间。在这一背景下，记忆系统的核心挑战不再是单一模态的信息存储，而是高效的跨模态语义对齐与调度，即如何构建统一的度量空间，使得不同模态的数据能够在同一语义维度下被索引与调度。

1. 跨模态表征的数学建模

在形式化层面，若用x_v表示视觉输入，x_t表示文本输入，现有多模态记忆系统的首要目标是构建一个统一的度量空间 Z，使得语义相关的图文对在该空间内的距离尽可能最小化。这通常通过优化对比损失函数（Contrastive Loss）来实现：

$$\min_{\theta} \mathcal{L}_{\text{contrastive}}\left(E_v(x_v), E_t(x_t)\right)$$

其中，E_v和E_t分别代表视觉与文本编码器（如 CLIP 或 SigLIP 架构）。E_v和E_t的对齐目标是构建统一的语义空间Z，该模型通过在联合嵌入空间中拉近正样本对（如“红色杯子”的图片与文字）并推开负样本对，赋予了智能体跨模态的联想能力。借此，智能体能够实现如“根据自然语言指令检索历史视觉监控帧”等复杂的跨模态认知任务。

需要指出的是，当前主流跨模态嵌入模型主要支持粗粒度语义对齐，在细粒度因果关系、动作可供性及时序依赖方面仍存在明显缺陷。因此，多模态记忆在工程上往往采用分层与异构的处理策略，而非试图以单一机制统一所有模态。

2. 多模态记忆的维度划分

随着智能体逐步深入视觉与物理环境，其记忆状态M_t不再是单一的文本序列，而是多种模态历史状态的融合集合。现有研究通常根据信息模态与功能属性，将多模态记忆划分为以下三个关键技术维度。

- 语义记忆（Semantic Memory）：对应文本与知识层面的表征生成，通常表示为向量集合$M_t^{\text{sem}} = \{e_i\}, e_i \in \mathbb{R}^d$。这是当前最成熟的技术路径，涵盖对话历史、事实性描述以及经抽象后的经验规则，主要通过向量检索支持语言层面的各类认知任务[4]。
- 视觉记忆（Visual Memory）：关注图像与视频等高维感知信息的长效保留。由于直接存储原始像素数据在计算与存储上代价极高，视觉记忆通常表示为特征级或摘要级的集合$M_t^{\text{vis}} = \{\phi(I_j)\}$，其中$\phi(\cdot)$为视觉编码或压缩函数。该维度强调在信息保真度与存储成本之间取得平衡[7]。
- 时空记忆（Spatio-temporal Memory）：用于描述智能体在物理环境中的状态演化与行为轨迹，通常形式化为状态–动作对的序列$M_t^{\text{traj}} = \{(s_i, a_i)\}_{i=1}^{t}$。此类记忆记录了位置变化、动作序列及环境反馈，是支持长程规划、具身智能及因果推理的重要基础[8]。

3. 工程存储与调度策略

在工程实现中，多模态记忆并非通过单一机制统一处理，而是根据数据特性采用差异化的混合存储策略。

- 密集向量索引（Dense Vector Indexing）：直接存储图像或视频帧的 Embedding 向量。这种策略保留了丰富的语义信息，适用于基于语义的模糊检索（例如“寻找红色的杯子”），是连接低级感知与高级认知的桥梁。
- 符号化引用（Symbolic Referencing）：将图像转化为文本描述（Caption）或资源定位符（URI），仅在长期记忆中存储元数据，而将原始高维数据留存于对象存储中[5]。这种方式显著降低了显存开销，并支持基于逻辑符号的快速查询。

不同模态记忆在存储成本、检索延迟与调用频率上存在显著差异。试图用单一机制统一处理所有模态往往会导致性能瓶颈。因此，构建分层、异构的记忆架构，根据模态特性分配不同的存储介质与检索策略，已成为生产级多模态智能体系统设计的核心演进方向。

3.2　记忆支撑与技术演进

在完成形式化定义后，本节将视角从“记忆是什么”转向“记忆如何工作”，探讨记忆能力在技术层面如何驱动智能体的复杂行为，并梳理现有研究中记忆建模范式的演进逻辑。

在 LLM 参数固定（Frozen Weights）的约束条件下，记忆系统的核心技术价值在于提供了一种可读写、可更新的外部状态空间。这使得智能体无须高昂的微调成本，即可获得近似“在线学习”与环境持续适应的能力[9]。

3.2.1　记忆对智能体能力的系统性支撑

从计算系统的视角分析，记忆模块通过动态调整输入分布（Input Distribution）和上下文先验（Context Prior），在不更新模型梯度的前提下，为智能体提供了三类核心计算支撑。

1. 跨时间步的状态一致性

在长程交互中，智能体面临“灾难性遗忘”与“角色漂移”的风险[10]。记忆系统通过维护一个持久化的角色状态向量（Persona State Vector）P与动态演化的历史状态M_t，确保智能体在t时刻的策略π_t与后续时刻的策略在核心设定上保持一致。形式化地，记忆机制通过最小化策略散度来约束行为漂移[11]：

$$\min_{M} \mathrm{D}_{\mathrm{KL}}\left(\pi(\cdot\,|s,P)||\pi(\cdot\,|s,M_t)\right)$$

通过检索并注入与当前任务强相关的记忆片段，系统能够强制模型输出收敛于既定的角色约束或长期目标，从而解决非马尔可夫环境下的行为连贯性问题。

2. 基于轨迹的技能演化

不同于传统的监督微调（SFT），智能体的技能增长往往依赖于程序性记忆（Procedural Memory）。系统将历史成功的任务执行轨迹存储为独立的“技能单元”：

$$\tau_{\text{success}} = (o_1, a_1, \ldots, o_n, r_{\text{final}})$$

当面对相似新任务T_{new}时，智能体通过检索相似度最高的τ_{success}作为 Few-Shot 示例，实现从 0 到 1 的能力泛化。这种机制在 Voyager 等系统中被证实是无须梯度下降即可显著提升任务成功率的有效路径[9]。

3. 复杂任务的分解与回溯

在处理长程复杂任务时，工作记忆充当了状态堆栈（State Stack）的角色，记录了任务分解树的当前遍历位置及未探索节点：

$$M_{\text{stack}} = \{(\text{SubTask}_i, \text{Status}_i, \text{Result}_i)\}_{i=1}^{N}$$

这种结构化记忆支持智能体在遇到执行错误时，读取M_{stack}进行精确的回溯（Backtracking）与纠错，避免了任务失败后的全盘重来，从而保障了长程任务的可解性[12]。

3.2.2 主流技术范式

智能体记忆机制的演进并非孤立发生，而是与大语言模型能力跃迁及智能体（Agent）应用场景的持续扩展呈现出显著的协同进化（Co-evolution）特征。从早期以问答为核心的被动式系统，到当前面向长期自主运行的复杂智能体，记忆的角色已从单纯的“知识检索接口”，逐步演化为支撑状态维持、角色一致性与认知重构的关键机制。围绕这一需求升级，研究与工程实践中逐渐形成了 4 类具有代表性的技术范式，分别对应智能体认知能力发展的不同阶段。

1. 基于检索的静态记忆范式

在 LLM 早期受限于知识截止（Knowledge Cutoff）与幻觉问题的背景下，检索增强生成（Retrieval-Augmented Generation，RAG）作为一种工程导向的补偿机制率先得到广泛应用[4]。

该范式将记忆理解为外部化的、相对静态的知识集合，通常以半结构化文本或文档形式存储，并在推理时通过向量索引与语义相似度计算进行“按需挂载”。其核心流程可概括为“查询-检索-注入”，即在生成前从外部知识库中检索相关片段，并将其作为上下文提示注入模型推理过程，从而提升事实一致性与领域覆盖能力。

该范式的优势在于工程实现成熟、存储成本低廉且能有效抑制模型幻觉，在企业知识库问答、法律与医疗文档咨询、垂直领域客服等弱状态或无状态任务中表现稳定，已成为工业界的事实标准方案之一。

然而，其局限性也十分明显：由于记忆内容通常是离线构建或简单的追加式写入，缺乏对时间维度和状态演化的建模，因此该范式更适用于解决“知识密集型”的问答任务，而在需要维持长期角色状态或处理动态社会关系的场景中，往往因缺乏时间关联性而表现乏力。

2. 基于上下文的窗口扩展范式

随着 Transformer 架构的持续优化以及算力成本的下降，部分研究尝试将记忆问题转化为纯粹的计算带宽问题。该范式试图利用超长上下文窗口（Long Context Window）、滑动窗口（Sliding Window）或线性注意力机制，尽可能多地将原始交互历史保留在模型的上下文感知视野中。

本质上，这是对“工作记忆（Working Memory）”物理容量的极致扩展，试图以近似无限的短期记忆覆盖原本需要长期记忆管理的功能。相比于检索式记忆，该范式保留了信息的因果时序和原始细节，避免了压缩带来的语义损失。因而该范式适用于对信息完整性和因果链条高度敏感的任务，如长文档分析、代码库级编程辅助、复杂文本续写等。

尽管该方法在语义保真度上具有优势，但其推理成本随序列长度呈线性甚至超线性增长，同时还面临“中段遗忘（Lost-in-the-Middle）”等注意力退化问题[6]。因此，该范式在跨越数周甚至更长时间尺度的持续性任务中，难以实现经济可行的部署。

3. 分层记忆架构范式

为了在存储成本与计算效率之间寻求平衡，工程界普遍采用一种分级管理的结构化范式。该范式主张放弃单一的存储机制，转而构建异构的层级系统。该范式普遍借鉴认知心理学中的多存储模型，通过构建异构、分级的记忆体系，在存储成本、检索效率与推理复杂度之间取得平衡。在实际工程落地中，根据任务复杂度的不同，也演化出了多种典型结构。

在轻量级应用中，最常见的是“双层架构（Dual-Layer）”，即由承载当前交互的短期上下文（Short-term Context）与承载历史档案的长期向量库（Long-term Vector Database）构成，这种简化设计省略了感官缓冲，适用于纯文本交互场景，常见于个人助理或轻量级 Agent，能够支撑稳定的多轮对话与基础偏好记忆。

而在多模态或高等级智能体中，则普遍采用标准的“三层架构（Three-Layer）”，显式引入感官缓冲层（Sensory Memory）以处理高通量的视觉或听觉数据流，防止原始噪声直接冲击推理核心。

此外，针对需要复杂规划的场景，部分前沿架构进一步将长期记忆细分为“多模块架构（Multi-Module Architecture）”，独立维护语义记忆（事实）、情景记忆（经历）与程序性记忆（技能）。该架构适用于开放世界游戏 NPC、复杂任务规划 Agent，以支持精细化的技能复用。

总体而言，这一范式通过层级间的动态调度（如 MemGPT 的分页机制[5]），在有限的资源条件下实现了近似“无限记忆”的效果，已成为当前通用智能体系统的主流设计基础。我们将在 3.3 节展开介绍智能体三层记忆参考框架。

4. 基于生成的认知模拟范式

随着智能体应用逐步迈向拟人化与社会化，受人类认知科学的启发，记忆的内涵开始从“存储结构”转向“认知过程”。在该范式中，记忆不再是被动保留的数据片段，而是一个可自主实现持续生成、抽象与重构的动态系统。以 Generative Agents[2]为代表的工作引入了“记忆代谢”机制，尤其强调反思（Reflection）算子的作用：系统周期性地对低层次交互与观测进行总结，抽象出高维的洞察（Insight），并将其重新写入记忆体系中。

该范式赋予了智能体“随时间成长”的能力，使其形成跨事件的价值判断、行为偏好与人格一致性，从而表现出更接近人类的长期行为模式。因而，其应用场景主要聚焦于虚拟社会模拟、陪伴型数字人、心理咨询与情感交互等对人格连续性和情绪一致性要求极高的应用。

尽管该路径在认知表达上最具潜力，但其系统复杂度、Token 开销与调参成本均显著高于前述范式，目前尚未实现规模化落地，仍主要停留在学术探索与高端实验性应用阶段。

3.2.3 研究挑战：从“有记忆”到“好记忆”

3.2.2 节介绍的 4 种范式虽然在各自的领域取得了进展，但在迈向通用智能体的过程中，仍面临若干共性的技术挑战。

首先是记忆写入与更新策略的理论缺失。目前的写入机制多依赖启发式规则，缺乏统一的理论模型来回答“何时应当写入”与“何时应当遗忘”的问题，导致系统容易在“记录过多噪声”与“遗漏关键信息”之间摇摆。

其次是记忆的可靠性与抗干扰问题。当生成误差或恶意信息被写入记忆库后，会形成持续的误导信号，目前的系统普遍缺乏基于逻辑真值的主动纠错（Active Correction）机制。

最后是跨模态对齐的鸿沟。在图文交错的场景下，视觉记忆与文本记忆在向量空间分布的不均匀性，往往导致检索系统倾向于召回文本，而忽略关键的视觉线索。正因为上述写入冲突与对齐鸿沟的存在，工程界迫切需要一套标准化的层级管理机制。为了解决写入冲突与对齐鸿沟，工程界转向了分层动态调度架构。

3.3 智能体三层记忆参考框架

为了解决上下文窗口限制与长程一致性需求之间的矛盾，本节提出一种通用的三层记忆参考架构。该架构模仿人类认知机理，将记忆划分为感知层（Sensory Layer）、工作层（Working Layer）和长期层（Long-term Layer），并通过动态调度器（Scheduler）管理信息在层级间的流转[1]。

需要特别说明的是，本书在工程实践语境下，对感知记忆、工作记忆与长期记忆进行了明确区分，如图 3-1 所示。其中，工作记忆在认知心理学中通常被称为短期记忆[3]，但在智能体系统的工程实现中，二者并不完全等价，前者特指映射至 LLM 上下文窗口、用于承载当前任务状态的专属容器[5]。

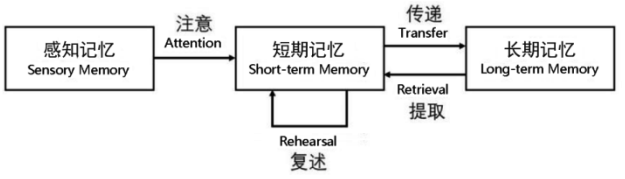

图 3-1 人类记忆的模块模型[3]

3.3.1 第一层：感知记忆

感知记忆位于智能体架构的最前端，直接对接环境与用户交互流。其工程职责并非理解输入信息，而是以尽可能低的延迟和成本，接收并规范化原始输入信号。从系统角度看，感知层更接近一个高并发的 ETL 预处理管道，核心功能是完成数据的标准化清洗、对齐与预处理，而非推理模块。

无论输入模态为何，感知记忆的本质都是一个生命周期极短、采用先入先出（FIFO）规则的缓冲区。其核心目标是为下游推理层提供“干净、对齐、可消费”的数据表示[13]。

1. 文本数据的清洗与标准化

对于最常见的文本输入（如用户对话、API 返回结果），感知层并不是简单地转发字符串，而是要进行一系列脏数据清洗（Data Sanitization）操作，以确保输入给大模型的 Prompt 是干净且高效的。工程实现上通常包含以下步骤。

（1）去噪（Denoising）：自动剔除输入流中的乱码、无意义的重复字符（如连续 100 个空格）或不可见的控制符（如\u200b），防止其干扰模型的注意力。

（2）截断与分片（Truncation & Chunking）：针对超长文本（如用户粘贴了一整本书），感知层需根据 Token 预算进行物理截断，或将其切分为多个块（Chunks），避免直接撑爆下层的上下文窗口[4]。

（3）提示词包装（Template Wrapping）：将原始文本封装到固定的 Prompt 模板中（例如添加 User:前缀或系统指令），将其标准化转化为模型可以直接解析的 Message 对象。

2. 多模态数据的降维与去重

当智能体具备视觉或多模态能力时，输入数据规模会呈数量级增长。感知层在此必须承担强制压缩与筛选的责任，不能将所有高维原始数据直接传递给下游大模型，否则会导致存储过载、推理延迟飙升。在工程实践中，通常采用以下两种策略。

- 视觉特征提取（变“图”为“数”）：大模型不能直接“存”图片，数据库也不能直接“搜”像素。因此，感知层会调用视觉编码器（如 CLIP[14]/SigLIP[15]）把图片变成一串数字（Embedding 向量）。下游任务可以通过计算向量相似度，快速搜到“红色的杯子”在哪幅图里，无须对每幅图片进行重复的像素级分析，提高检索和处理的效率。
- 视觉去重与关键帧提取（变“流”为“帧”）：视频其实就是一秒 30 幅图像的连拍。如果画面静止不动，把这 30 幅图全发给 LLM 就是纯粹的浪费。工程上我们常常维护一个“上一帧缓存”。每来一幅新图，就跟上一幅比对。如果画面变化率（像素差异或向量距离）小于 5%，就直接丢弃；只有在画面突变时（比如人走过、屏幕切换），才保留这幅图作为“关键帧”。这种简单的“视觉去重”机制，能把原本巨大的视频流数据量进行数量级的压缩[9]，极大地节省了存储和带宽。

为了适配上述逻辑，感知记忆单元（Sensory Memory Unit，SMU）通常采用轻量化的 JSON 结构。它不存储复杂的逻辑推演结果，仅保留原始数据的引用（URI）、时间戳以及对应的向量表征，以支持快速的检索与下游处理。一个典型的多模态感知记忆单元定义如下：

```
// 感知记忆单元数据结构示例
{
  "trace_id": "evt_20240125_001",
  "timestamp": 1706179200,
  "source": "user_screen_capture",
  // 模态标识：决定后续处理管线
  "modality": "image",
```

```
    // 原始数据通常存储在对象存储（S3/OSS）中，此处仅保留引用
    "content uri": "s3://agent-memory/vision/frame 882.jpg",
    // 向量表征：用于后续的快速检索与相似度计算
    "embedding vector": [0.12, -0.45, ..., 0.88],
    // 预处理结果：VLM 生成的简短 Caption 或 OCR 提取的文本
    "description": "User pointing at a red error log on screen",
    // 若为纯文本模态，则 raw text 非空，image 字段为空
    "raw text": null
  }
```

3.3.2 第二层：工作记忆

感知记忆负责“看见世界”，工作记忆则是智能体进行思考与决策的核心场所。在工程实现中，其物理载体几乎总是大语言模型的当前上下文窗口。

不同于感知层的流式吞吐，工作记忆具有高度的结构化特征。它不存储漫漫历史长河中的所有细节，而是作为一个容量有限、聚焦当前的状态容器，仅维护与当前任务目标（Current Goal）直接相关的三类核心变量。形式化上，工作记忆的状态S_{wm}可定义为一个三元组：

$$S_{\text{wm}} = \langle G_{\text{current}}, H_{\text{act}}, K_{\text{related}} \rangle$$

- G_{current}：当前任务的动态分解（Sub-goal Decomposition）[16]。
- H_{act}：当前步骤的执行轨迹与推理链（Chain of Thought）[17]。
- K_{related}：从记忆中按需检索到的相关背景知识。

为了防止长程推理中的“上下文漂移（Context Drift）”，目前智能体架构普遍采用暂存板（Scratchpad）模式进行状态管理[18]。Scratchpad 是一个动态更新的结构化对象（通常为 JSON 或 Markdown 格式），它实时记录任务的进度、子目标完成情况及当前推理断点，为后续步骤提供清晰的状态支撑。

更重要的是，工作记忆层承担着“脏数据隔离（Dirty Data Isolation）”的关键职责：在任务执行过程中产生的错误尝试（Error Trials）、无效的 API 调用以及冗余的思维链，会被严格限制在工作记忆的生命周期内，不得随意写入长期记忆。工作记忆是一个“沙盒”——一旦任务结束，只有经过验证的成功经验（Insight）会被提取并固化进长期记忆，而大量的过程噪声（Process Noise）将被直接丢弃，从而避免长期记忆库被低质量信息污染，保障长期记忆的可靠性[19]。

在代码实现中，工作记忆并不是静态存在的，而是需要在每一次 LLM 推理请求前，根据当前任务状态和检索到的背景信息动态组装成 Prompt。以下伪代码展示了如何利用工作记忆构建一个抗干扰的系统级上下文（System Context）：

```
    class WorkingMemoryContext:
       def build_prompt(self, task, sensory_data, long_term_knowledge,
scratchpad):
          """
          将工作记忆的各个组件组装为 LLM 可理解的 Prompt
          """

          # 1. 系统指令：定义角色与输出格式
          system_prompt = """
```

```
            You are an SRE Agent. You have access to a shared 'Scratchpad' to track
your state.
            Your goal is to complete the objective using the provided tools.
            """

            # 2. 注入感知数据 (Sensory Memory)
            # 注意：这里可能包含图像的 Embedding 占位符 <image embedding>
            sensory_block = f"""
            [CURRENT OBSERVATION]
            Target Image: {sensory_data.description}
            (Visual features aligned and injected via encoder)
            """

            # 3. 注入长期知识 (Long-term Memory / RAG)
            knowledge_block = f"""
            [RELEVANT KNOWLEDGE]
            {'\n'.join([f"- {k}" for k in long_term_knowledge])}
            """

            # 4. 注入当前状态机 (Working Memory Scratchpad)
            # 核心：这是 LLM“看到”自己当前思维进度的地方
            state_block = f"""
            [CURRENT STATE & HISTORY]
            Steps Taken:
            {json.dumps(scratchpad['reasoning_trace'], indent=2)}

            Current Variables:
            {json.dumps(scratchpad['variable_buffer'], indent=2)}
            """

            # 5. 组装最终 Prompt
            # 结构：[角色] -> [感知] -> [知识] -> [状态] -> [下一步指令]
            final_prompt =
f"{system_prompt}\n\n{sensory_block}\n\n{knowledge_block}\n\n{state_block}\n\nU
ser: {task.instruction}"

            return final_prompt
```

通过这种模板化的组装，工作记忆将分散的感知信号、历史知识和当前的推理状态“压扁（Flatten）”，进入到大模型的有限窗口中，确保了模型在每一步决策时都能获得完整的上下文感知（Context Awareness）[5]。

3.3.3　第三层：长期记忆

长期记忆（Long-term Memory，LTM）是智能体的“大脑皮层”，负责存储跨越任务周期的经验、知识与人格设定。作为一种非易失性（Non-volatile）存储，它在技术上解决了大模型参数固定后的知识更新难题。从工程角度看，现代智能体系统普遍采用向量与图结构相结合的混合存储架构，以平衡语义泛化能力与逻辑可验证性[20]。

这种混合架构将长期记忆划分为以下三个逻辑分区。

- 语义记忆（Semantic Memory）：基于向量数据库（如 Milvus、Chroma）构建。它存储事实性知识片段，通过向量检索（Vector Retrieval），智能体能够基于模糊语义召回相关背景（例如“回忆上次用户提到的服务器配置”），并结合重排序（Reranking）机制显著降低记忆检索阶段的幻觉率[21]。
- 情景记忆（Episodic Memory）：存储压缩后的“事件–结果”对（Event-Result Pairs）。这种记忆不仅包含事实，还包含因果链条，是智能体实现 Voyager 式“技能复用”的基础[9]。
- 实体关系图（Entity Graph）：基于图数据库（如 Neo4j）存储明确的三元组关系（如 (Project_A,depends_on,Lib_B)）。这种结构弥补了向量检索在多跳推理（Multi-hop Reasoning）上的短板，确保智能体在处理复杂实体依赖时，不会产生逻辑幻觉[22]。

一种基于 PolarDB 和 mem0 框架的 Agent 记忆存储与搜索方案架构，如图 3-2 所示。

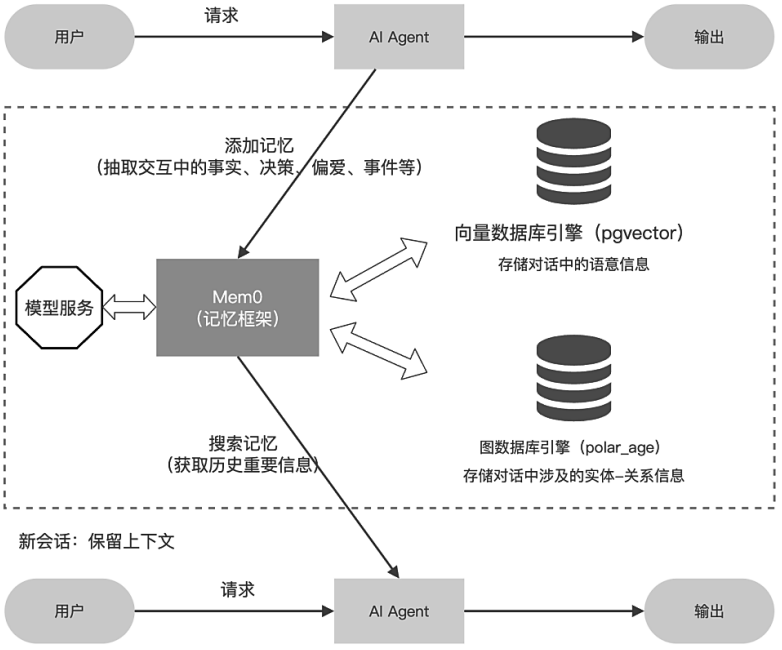

图 3-2　一种基于 PolarDB 和 mem0 框架的 Agent 记忆存储与搜索方案架构

针对多模态记忆的持久化，工程上应遵循“索引与内容分离”的原则。原始的高维数据（如图像、视频片段）存储在低成本的对象存储（Object Storage, S3/OSS）中，而长期记忆数据库中仅存储其“视觉语义索引”。具体流程如下。

- 转化：将图像I通过多模态大模型（VLM）转化为详细的文本描述T_{desc}。
- 索引：同时索引图像特征向量$\text{Vec}(\boldsymbol{I})$和文本描述向量$\text{Vec}(T_{\text{desc}})$。
- 检索：这种双路索引机制支持灵活的“以文搜图”和“以图搜文”。

值得注意的是，对于长期记忆记录（Memory Record）的数据结构设计，需要兼顾检索效率与遗忘机制[2]。

3.3.4　动态调度机制

上述三层架构（感官-工作-长期）并非孤立存在，而是通过一个调度器（Scheduler）实现数据的动态流转。这一流转过程构成了智能体记忆的完整生命周期，涵盖从注意力的聚焦到经验的固化。

1. 注意力过滤

L1→L2（Attention Filtering）并非所有的感官输入都有资格进入工作记忆。调度器依据当前任务目标G_{current}构建注意力过滤器f_{attn}。只有当感知信号o_t与当前目标具有显著相关性时，才会被“写入”工作记忆的 Scratchpad。

2. 记忆固化与反思

L2→L3（Consolidation&Reflection）当任务完成或上下文窗口即将溢出时，系统会触发记忆固化流程。此时，智能体不只是简单地转存日志，而是调用 LLM 执行“摘要与反思（Summarize&Reflect）”操作。

- 摘要：将冗长的交互轨迹压缩为关键事实。
- 反思：从具体经历中提取通用的“如果……那么……”经验规则（例如，“遇到 A 错误时，B 方案通常无效”）。只有经过反思后的高价值信息才会被写入长期记忆数据库，从而实现从“经历”到“经验”的转化[19]。

3. 上下文注入与召回

L3→L2（Retrieval&Injection）在每一轮决策前，调度器会基于当前的工作记忆状态C_t生成查询向量$\boldsymbol{q}$，从 L3 中召回 Top-K 最相关的记忆片段，并将其动态注入到工作记忆的 retrieved_knowledge 字段中。

$$K_{\text{related}} = \text{Top}K(\text{Memory}_{L3}, \boldsymbol{q}(C_t))$$

这种按需加载（On-demand Loading）机制，使得智能体能够在不扩充模型参数规模的前提下，在局部任务中表现出拥有近乎无限知识储备的行为特征[5]。

3.3.5　工程化示例：从 Hello World 到生产级

上述三层记忆架构提供了完备的理论模型，但在实际工程开发中，并非所有应用都需要构建如此复杂的全量系统。开发者应根据业务复杂度与资源约束，选择适配的架构实现层级。本小节以通用开发框架（如 LangChain）为例，展示记忆架构如何从“单层缓冲”逐步演进为“三层协同”的生产级系统。

1. Level 1：基础缓冲模式

这是最简单、最直观的记忆实现，适合初学者入门做一些短轮次机器人、短周期的任务智能体或快速原型验证。在这一阶段，系统在工程上暂时舍弃了感知过滤与长期存储模块，将“记忆”完全等同于大语言模型的“当前上下文窗口”。

开发者通常不依赖任何外部数据库，直接在内存中维护一个 Python 列表（List）或 JSON 列表结构，将用户与智能体的所有历史对话（User/AI Message）按顺序存储。在每次调用 LLM 时，将这个列表拼接成字符串，全量注入上下文窗口，例如 Prompt 的 System Message 或 History 插槽中。以下是一段伪代码：

```
#以 LangChain 为例，通过调用 ConversationBufferMemory 类即可实现
# 伪代码示例：最简单的记忆实现
from langchain.memory import ConversationBufferMemory

# 1. 初始化记忆对象（相当于创建一个空列表）
memory = ConversationBufferMemory()
# 2. 模拟对话过程：save context
memory.save_context({"input": "你好，我叫小明"}, {"output": "你好小明！"})
# 3. 下一轮调用前，加载记忆
# 这一步会自动把上面的对话拼接到 Prompt 中
history = memory.load_memory_variables({})
# history 输出："Human: 你好，我叫小明\nAI: 你好小明！"
```

这种实现方式的优势在于逻辑极简且能保留对话的全部原始细节，但其局限性也极为明显：随着交互轮次的增加，Prompt 长度将迅速逼近模型的上下文阈值，导致高昂的 Token 成本甚至服务中断[6]。因此，它仅适用于对状态保持要求不高的即时交互场景。

2. Level 2：滑动窗口与摘要模式

为了缓解 Level 1 的上下文溢出问题，引入了“滑动窗口”与“摘要”模式。这一阶段的核心思想是在工作记忆层引入“有损压缩”机制。一方面，系统引入时序滑动窗口（Sliding Window），通过仅保留最近K轮对话的策略，模拟了感知记忆对陈旧信息的自然遗忘；另一方面，对于滑出窗口的历史信息，系统并不直接丢弃，当历史长度 $L > L_{\max}$ 时，触发 Summarize 操作：

$$M_{\text{summary}} = \text{LLM}(M_{\text{summary}} + \text{oldest\_turns})$$

利用 LLM 的摘要能力将其转化为高度凝练的文本描述，重新注入 Prompt 的系统指令区。

这种机制实际上构建了一种“准长期记忆”，在有限的上下文空间内实现了对长程历史语义的保留，常被用于构建需要维持一定连贯性的客服机器人或聊天伴侣[19]。

3. Level 3：外挂检索模式

当智能体需要处理海量知识或具备无限生命周期时，则需要外挂知识库或记忆库。

如图 3-3 所示为通过 Letta 框架搭建的电商客服机器人问答流程示例。

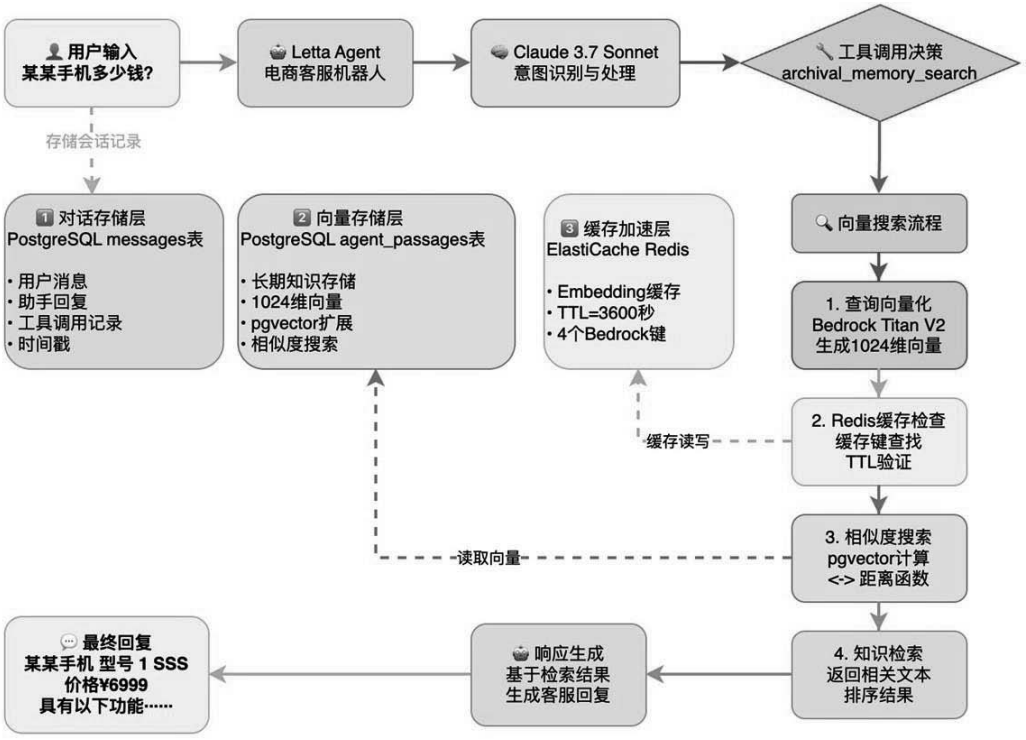

图 3-3　通过 Letta 框架搭建的电商客服机器人问答流程示例

在此阶段，工作记忆仅保留当前意图，海量历史被卸载至独立的外部存储设施（如向量数据库）。系统将历史交互进行切片、向量化并持久化存储，构建起物理分离的“长期记忆层”。

此时，工作记忆层退化为一个动态的暂存区，仅在决策前通过语义检索（Semantic Retrieval）按需从外部数据库中召回与当前意图最相关的片段。

```
# 引入外部存储层
class VectorMemory:
    def __init__(self, vector_db):
        self.l3_storage = vector_db  # Long-term Memory

    def retrieve(self, query):
        # 1. 语义检索：从 L3 召回 Top-K 相关记忆
        docs = self.l3_storage.similarity_search(query, k=3)
        return docs

    def run(self, query):
        # 2. 注入：将召回的记忆注入 L2 (Context)
        context = self.retrieve(query)
        prompt = f"基于记忆 {context} 回答: {query}"
        return llm.predict(prompt)
```

这种架构彻底解耦了记忆容量与推理窗口的依赖关系，使得智能体能够在不额外增加推理成本的前提下，利用近乎无限的知识库进行专业决策，是构建领域知识专家、无限周期角色扮演、复杂任务规划 Agent 的标准范式[4]。

3.4 开源框架、工具与评估指标

随着智能体记忆研究从概念验证阶段逐步走向规模化落地应用，工程实践逐渐形成了一套围绕记忆管理、存储基础设施与评估机制的完善开源技术生态。与早期仅关注向量存储不同，当前的系统设计已明显呈现出三个演进方向：记忆编排的系统化、存储结构的混合化以及能力评估的自动化。

这一变化表明，智能体记忆已不再被视为某个单一算法模块，而是逐步演化为一类需要操作系统式管理与持续观测的工程能力。本节将从工程分工的角度，梳理支撑智能体记忆能力高效落地的关键软件框架与评估方法。

3.4.1 记忆编排与管理框架

在智能体架构的应用层，记忆管理框架承担着协调信息写入、压缩与检索路径的核心职责，其角色更接近“大脑皮层”的动态调度机制，而非单一存储组件。

当前主流的工程范式，是通过模块化接口与流式编排实现记忆策略的可配置与可替换，以 LangChain 和 LlamaIndex 为代表。LangChain 通过标准化的 Memory 抽象类，确立了“组件化记忆”的设计思想，将底层的存储后端（如 Redis、Postgres）与上层的推理逻辑彻底解耦。开发者可以通过配置不同的记忆组件（如 EntityMemory 或 SummaryBufferMemory），在不修改核心业务代码的前提下，灵活切换“滑动窗口”或“实体提取”等记忆策略。相比之下，LlamaIndex 则更侧重于数据索引层的构建，其提出的 IndexStruct 概念使得智能体能够高效地在海量非结构化数据中建立层级索引[23]，尤其适用于需要大规模知识检索（RAG）的专家型智能体。

然而，随着长程伴侣与复杂规划任务需求的兴起，一种更具颠覆性的“虚拟内存”范式开始涌现，其核心代表为 MemGPT。不同于传统框架的“外挂式”设计，MemGPT 借鉴了现代操作系统中层级存储体系（Hierarchical Memory Hierarchy）的思想，将 LLM 的有限上下文窗口视为高速缓存（RAM），而将外部向量库视为持久化存储（Disk）。通过引入系统级指令（System Instructions）与分页机制（Paging），MemGPT 赋予了智能体自主管理信息换入与换出的能力，从而在物理有限的推理窗口内实现了逻辑无限的上下文跨度。这种架构设计标志着智能体记忆从“被动存储”向“主动操作系统”演进的重要转折[5]。

此外，在多智能体协同场景中，记忆管理进一步扩展为跨角色的信息协调问题。例如，MetaGPT 引入的共享记忆池机制，通过发布/订阅模式在不同角色智能体之间同步或隔离状态，为群体协作中的一致性控制提供了切实可行的工程解决方案[24]。

3.4.2 混合存储基础设施

在存储层面，智能体记忆的持久化载体正经历从单一向量检索模式向“向量+图+关系”的混合架构演进。这一趋势的根本动因在于：语义相似性检索虽具备良好的泛化能力，但在精确逻辑推理与事实一致性方面存在天然不足。

向量数据库（Vector Database）依然是当前语义记忆的核心载体。Milvus、Chroma 和 Weaviate 等高性能引擎通过引入 HNSW（Hierarchical Navigable Small World）等近似最近邻算法，解决了亿级向量规模下的毫秒级检索延迟问题[25]。为了适应多模态记忆的需求，现代向量数据库已开始原生

支持多向量索引（Multi-vector Indexing），即允许对同一记忆对象同时索引其文本摘要、视觉特征（Image Embedding）及元数据标签，从而大幅提升了跨模态检索的召回率（Recall）。

然而，当任务涉及复杂逻辑关系或可追溯性要求时，纯向量检索往往难以胜任。这促使图数据库在记忆系统中重新获得关注。以 Neo4j 和 NebulaGraph 为基础构建的 GraphRAG 技术，通过将非结构化文本转化为知识图谱三元组（Triplets），为记忆系统注入了结构化约束。微软开源的 GraphRAG 框架展示了如何结合“图聚类（Graph Clustering）”与“向量相似度”，实现对大规模数据集的全局性摘要与多跳推理（Multi-hop Reasoning）[20]，为医疗、法律等高可靠性场景提供了可扩展方案。

为了进一步降低系统集成复杂度，一类“记忆即服务（Memory-as-a-Service）”中间件开始出现。Zep、Mem0 等工具将记忆管理流程标准化、服务化，内置摘要、隐私脱敏与情绪分析管线，使开发者能够在不深度介入底层存储细节的情况下，引入具备生命周期管理能力的记忆系统。这类方案在工程上更强调运维友好性与合规性，而非极致性能。

3.4.3　记忆能力的评估指标

随着记忆系统逐步成为智能体架构中的关键基础设施，其效果评估也从早期的定性“人工体验式判断”转向定量、可重复的“自动化测试（Automated Evaluation）”。目前的评估体系主要围绕检索质量、生成忠实度与长程一致性三个核心维度展开。

在工具层面，RAGAS（Retrieval Augmented Generation Assessment）等评估框架已形成事实上的行业标准。该类工具采用 LLM-as-a-Judge 范式，通过构建对抗性数据集，自动计算上下文召回率（Context Recall）与信实度（Faithfulness）等关键指标，量化评估智能体是否准确检索到相关记忆，以及生成的回答是否严格忠实于记忆内容。这为记忆系统中常见的“幻觉”问题提供了可监测、可量化、可优化的技术手段。[21]。

针对长程记忆的健壮性，诸如“大海捞针（Needle in a Haystack）”测试的变体被广泛采用，用以评估系统在超长交互周期中定位关键信息的能力。TruLens 等观测工具通过可视化追踪每一次检索的 Token 消耗与相关性得分，帮助开发者诊断记忆系统在超长对话周期中的性能衰减曲线。

这些评估工具的出现，标志着智能体记忆能力的开发正从依赖经验调试的“黑箱工程”，逐步迈向可观测、可比较、可优化的工程阶段。

3.5　本章小结

本章围绕智能体记忆能力这一核心技术要素，系统梳理了其在智能体系统中的角色定位、技术演进路径以及工程实现形态。

首先，在能力定义层面，本章明确指出智能体记忆并非简单的历史记录或日志存储，而是参与智能体决策过程、支持推理链路的内部状态变量。通过将记忆形式化为跨时间演化的系统状态，本章为理解“记忆如何进入智能体计算流程”提供了技术视角。

其次，在研究现状分析中，本章将现有工作归纳为检索式记忆、基于状态或轨迹的记忆以及生成式与反思式记忆等主要技术范式，揭示了不同方法在建模假设、能力边界与系统风险上的差异，并重点讨论了多模态与视觉记忆在长时任务场景下面临的关键挑战。

在此基础上，本章提出了一种面向智能体应用开发的三层记忆参考架构，通过区分感知记忆、工作记忆与长期记忆，解决了访问效率、存储成本与信息抽象程度之间的结构性冲突，并明确了跨层读写的基本工作流（Workflow），为工程实践提供了可直接参考的架构范式。

最后，本章结合当前主流的开源记忆框架与底层存储基础设施，展示了智能体记忆能力在工程实践中的落地方式，并讨论了面向长期运行系统的评估指标体系，说明记忆能力已经从研究概念走向可部署的系统模块。

3.6 参考文献

[1] Sumers T, Yao S, Narasimhan K R, et al. Cognitive architectures for language agents[J]. Transactions on Machine Learning Research, 2023.

[2] Park J S, O'Brien J, Cai C J, et al. Generative agents: Interactive simulacra of human behavior[C]//Proceedings of the 36th annual acm symposium on user interface software and technology. 2023: 1-22.

[3] Atkinson R C, Shiffrin R M. Human memory: A proposed system and its control processes[M]//Psychology of learning and motivation. Academic press, 1968, 2: 89-195.

[4] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in neural information processing systems, 2020, 33: 9459-9474.

[5] Packer C, Fang V, Patil S G, et al. MemGPT: Towards LLMs as Operating Systems[EB/OL]. (2023-10-12)[2026-06-23]. arXiv preprint arXiv:2310.08419, 2023.

[6] Liu N F, Lin K, Hewitt J, et al. Lost in the middle: How language models use long contexts, 2023[EB/OL]. (2023-07-06)[2026-06-23]. arXiv preprint arXiv:2307.03172, 2023.

[7] Alayrac J B, Donahue J, Luc P, et al. Flamingo: a visual language model for few-shot learning[J]. Advances in neural information processing systems, 2022, 35: 23716-23736.

[8] Chen L, Lu K, Rajeswaran A, et al. Decision transformer: Reinforcement learning via sequence modeling[J]. Advances in neural information processing systems, 2021, 34: 15084-15097.

[9] Wang G, Xie Y, Jiang Y, et al. Voyager: An open-ended embodied agent with large language models, 2023[EB/OL]. (2023-05-25)[2026-06-23]. arXiv preprint arXiv:2305.16291, 2023.

[10] Choi J, Hong Y, Kim M, et al. Examining Identity Drift in Conversations of LLM Agents[EB/OL]. (2024-12-02)[2026-06-23]. arXiv preprint arXiv:2412.00804, 2024.

[11] Chen R, Arditi A, Sleight H, et al. Persona vectors: Monitoring and controlling character traits in language models[EB/OL]. (2025-07-29)[2026-06-23]. arXiv preprint arXiv:2507.21509, 2025.

[12] Zhang Z, Chen T, Xu W, et al. ReCAP: Recursive Context-Aware Reasoning and Planning for Large Language Model Agents[EB/OL]. (2025-10-30)[2026-06-23]. arXiv preprint arXiv:2510.23822, 2025.

[13] Gao Y, Xiong Y, Gao X, et al. Retrieval-augmented generation for large language models: A survey[EB/OL]. (2023-12-18)[2026-06-23]. arXiv preprint arXiv:2312.10997, 2023, 2(1).

[14] Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language

supervision[C]//International conference on machine learning. PmLR, 2021: 8748-8763.

[15] Zhai X, Mustafa B, Kolesnikov A, et al. Sigmoid loss for language image pre-training[C]// Proceedings of the IEEE/CVF international conference on computer vision. 2023: 11975-11986.

[16] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in neural information processing systems, 2022, 35: 24824-24837.

[17] Yao S, Yu D, Zhao J, et al. Tree of thoughts: Deliberate problem solving with large language models[J]. Advances in neural information processing systems, 2023, 36: 11809-11822.

[18] Yao S, Zhao J, Yu D, et al. React: Synergizing reasoning and acting in language models[C]//The eleventh international conference on learning representations. 2022.

[19] Shinn N, Cassano F, Gopinath A, et al. Reflexion: Language agents with verbal reinforcement learning[J]. Advances in Neural Information Processing Systems, 2023, 36: 8634-8652.

[20] Edge D, Trinh H, Cheng N, et al. From local to global: A graph rag approach to query-focused summarization[EB/OL]. (2024-04-24)[2026-06-23]. arXiv preprint arXiv:2404.16130, 2024.

[21] Es S, James J, Anke L E, et al. Ragas: Automated evaluation of retrieval augmented generation[C]//Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations. 2024: 150-158.

[22] Pan S, Luo L, Wang Y, et al. Unifying large language models and knowledge graphs: A roadmap[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(7): 3580-3599.

[23] Liu J. LlamaIndex [EB/OL]. (2022-11)[2026-02-01]. https://github.com/jerryjliu/llama_index.

[24] Hong S, Zhuge M, Chen J, et al. MetaGPT: Meta programming for a multi-agent collaborative framework[C]//The twelfth international conference on learning representations. 2023.

[25] Malkov Y A, Yashunin D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 42(4): 824-836.