从长上下文到分层状态：面向长程智能体的设计框架

YunFei M. Wang
（王·Micheal·雲飛）
Independent Researcher

摘要

当前大模型在短文本生成上已接近成熟，但在长篇创作中暴露出局部性生成、一致性缺失、状态丢失、缺乏规划、记忆不分层等结构性缺陷。本文提出一种面向长程智能体设计的概念框架，旨在为后续研究与工程实现提供分析工具与设计原则。框架从成熟人类作者的长篇创作认知模式中提取核心结构，将人类作者驾驭长篇所依赖的压缩性认知模型——相——操作化为分层记忆与递归状态两个系统。分层记忆以本底键、经验键、活跃键三层组织作品信息，三层之间存在升格通道：经验键可因时间推移与重复影响固化为本底键，活跃键可因结果不可逆而升格为经验键或本底键。递归状态通过可更新的状态变量持续维护“当前故事状态”。对象模型按层级组织为时空环境、物质环境、社会环境、人际关系、角色/NPC 五类，各模型均以三层键结构承载数据。本文通过一个推演示例展示该框架在具体场景中的运作方式，并以功能映射的方式说明认知建模传统与近年长程智能体研究中的核心机制分别对应本文架构的哪个功能部位。本文的贡献不在于提出具体的系统实现方案，而在于提供一组可分析、可讨论、可修正的概念工具。最后简要指出可迁移的扩展方向，讨论工程实现中面临的关键挑战与回应路径，并论证该框架从专业工具向普适生态发展的泛化路径。

关键词

分层记忆；递归状态；长程一致性；键升格；设计框架；概念框架；大模型架构；智能体；认知建模

1 引言：从创作现场到设计框架

1.1 研究缘起

本研究的起点是作者在超长篇作品创作中遭遇的具体障碍。百万字级的叙事工程要求持续维护数百个角色、数十条伏线、多套时间线与一套自洽的世界观。作者在使用当前主流大模型辅助写作时，反复遭遇同一类问题：角色设定在数百轮对话后被混淆，世界观规则在后续生成中被直接违背，角色行为判断倾向于依据当前片段的情感氛围而非全局状态。

这些问题的反复出现，使作者逐渐意识到症结不在上下文窗口的大小，而在模型缺乏一种持续维护的、结构化的“语境模型”。模型接收了大量前文作为输入，却始终停留在文本序列的层面操作，未能将前文转化为可更新、可审计、可约束后续生成的状态结构。

1.2 从能力差异到核心洞察

作者与 DeepSeek AI 助手围绕“人类成熟作者依靠什么机制维持长程一致性”展开探讨。形成的判断是：人类成熟作者不依赖对全部前文的精确记忆，而是依靠一套随创作持续更新的压缩性认知模型。这套认知模型就是“相”——它不是前文的索引，而是前文的压缩与结构化。作者写每一段时，并不调取全部历史，而是从这个持续演化中的“相”出发。一致性不是事后检查的结果，而是这个模型在生成时自然施加的约束。

人类作者所依赖的“语境”不同于模型所接收的“上下文”。语境不是文本的前缀，而是一种被理解了的含义场。语境就是作者在长程创作中持续维护的那个相：它由写作操作产生，随信息整合而更新，并反过来约束后续每一次写作操作。当前模型虽然接收了全部前文作为输入，但它的信息整合是概率性的、混沌的、不可审计的——注意力权重分布替代了结构化约束，微小输入变化可导致表征剧变，无法追溯哪些信息被纳入、哪些被遗漏、权重如何分配。它缺少的不是整合环节本身，而是一个结构化的、可审计的、可被显式修正的相，以及让这个相反馈约束生成的机制。

1.3 设计框架定位

本文的贡献在于提供一种分析长程智能体设计问题的概念框架。通过将人类作者的认知模式操作化为可分析的结构系统——分层记忆与递归状态——本文为后续研究提供可讨论、可验证、可修正的基础。

这一框架不主张“应该这样实现”，而是主张“可以从这个角度分析问题”。它的价值不在于技术细节的完备性，而在于概念工具的可迁移性：分层记忆与递归状态不仅是解决长程一致性问题的具体方案，更是分析任何需要长期状态维护的智能体任务的一般性框架。

1.4 结构安排

本文按以下序列展开：

- 第 2 章：诊断当前模型在长程任务中的结构性缺陷。
- 第 3 章：从人类作者认知模式中提取核心结构，建立五个对象模型的层级框架。
- 第 4 章：提出分层记忆与递归状态两个系统，含键层级的升格机制。

- 第 5 章：通过推演示例展示框架在具体场景中的运作方式。
- 第 6 章：以功能映射的方式说明认知建模传统与近年长程智能体研究中的核心机制分别对应本文架构的哪个功能部位。
- 第 7 章：指出可迁移的扩展方向。
- 第 8 章：将框架扩展为多分支并行维护的操作化方案。
- 第 9 章：讨论工程实现中面临的关键挑战与回应路径。
- 第 10 章：论证框架从专业工具向普适生态发展的泛化路径。
- 第 11 章：讨论跨领域迁移潜力以及与第一套、第三套方案的协同关系。
- 第 12 章：给出结论。

2 现有模型的结构性缺陷：一个诊断性框架

当前大模型在长程任务中暴露的问题可归纳为五点。这五点有一个共同的结构性根源：模型并非没有相的雏形，但它缺少结构化的、可审计的、可显式修正的相的构建与更新机制。这一共同根源可以这样理解：模型在每一轮生成中确实在进行某种信息整合，但这种整合以注意力权重分布的形式存在，是概率性的、混沌的、不可审计的。它无法被显式检查、无法被定点修正、无法被可靠地传递到下一轮生成。

2.1 局部性生成

当前多数模型的核心训练目标是“下一 token 预测”。它天然擅长局部连贯，但缺乏全局视角。生成第 500 章时，它已无法有效感知第 3 章埋下的伏笔，因为两者距离远超有效注意力范围。

2.2 一致性缺失

现有模型没有显式的“故事状态”概念。角色性格、世界观规则、时间线关系，全都以文本形式散落在上下文中。一旦上下文截断或压缩，信息就会丢失、扭曲，导致前后矛盾。一致性缺失的本质是相与故事实际状态之间的偏差失去了控制。作者脑中持续维护的相，是对故事全局结构的认知模型；大语言模型的每一次生成，本应以这个相为约束。而当相不可审计、不可显式修正时，每次生成都成为对故事状态的一次重新推断——它可能恰好与前文的因果收束一致，也可能不一致。一致性因此依赖于上下文能否提供足够约束，而非结构性保证。

2.3 状态丢失与无法递推

现有模型每一段生成都重新计算前文，缺乏递归状态传递。它不“记得”上一章结尾时角色的心理状态，除非这些状态被明确写在文本里。人类作者则会在脑中自然保持状态的连续更新。现有模型缺少的是"操作→信息产生→相修正→相反馈指导下一轮操作"的完整循环。人类作者的相在每一轮写作后都被更新——新的情节信息被整合进现有的故事认知模型，使相保持当前状态。现有模型的整合结果无法被可靠地携带到下一轮，它每次都从零开始重新计算，没有可递推的相。

2.4 缺乏规划能力

现有模型通常是“边写边想”，直接生成正文。它不会先规划“这一章的功能是什么、要达成什么目标、需要埋下或回收什么线”，而是被前文牵着走，容易跑偏或重复。规划功能属于相的高层级操作：它要求认知模型不仅反映当前状态，还能模拟未来状态——在相内进行"如果……那么……"的推演。模型的相不可审计、不可显式操作，导致它无法在生成前构建关于本章目标的认知模型，更无法让这个目标模型约束生成过程。

2.5 记忆不分层与无升格机制

现有模型无法区分长期稳定信息（角色核心性格）和短期动态信息（当前情绪、位置）。所有信息被一视同仁地塞入上下文，既造成冗余，又导致关键信息被淹没。

记忆不分层不是与前三者并列的独立缺陷，而是“相不可审计”在信息组织层面的直接表现——前四点是相不可审计导致的行为后果，2.5 是相不可审计的结构特征。它对应自然人键结构三个层次——本底键、经验键、活跃键——的扁平化。本底键具有最高稳定性，经验键阶段性更新，活跃键频繁变化。现有模型的扁平化处理，相当于将三种不同时间尺度的键结构压缩进同一个无差别空间，既破坏了信息的层级结构，也使得关键信息在生成时无法被按需调用。

现有模型不仅“不分层”，而且“无升格机制”。它无法把一次不可逆事件（活跃键）可靠地写入长期状态，也无法把反复出现的行为模式（经验键）固化为稳定倾向（本底键）。一次角色死亡、一次关系决裂、一次世界观规则的永久性改变，在现有模型中只是上下文中的一个片段；下一轮生成时，它可能被重新推断为“尚未发生”或“可以逆转”。升格机制的缺失，使模型无法区分“可以撤回的临时状态”与“不可撤回的永久改变”。

3 思路来源：从人类作者认知模式中提取对象模型框架

普通成熟作者写长篇小说时，并不依赖超忆症或外部检索工具。他们依靠的是一套高度压缩、结构化的内部认知模型——一个随创作持续更新的相。本章的任务是从这一认知模式中提取可迁移的对象模型框架。

3.1 五个对象模型的层级链

长程叙事所维护的状态不是单一的角色状态，而是按层级嵌套的五类对象。它们之间存在双向影响链：

时空环境 → 物质环境 → 社会环境 → 人际关系 → 角色/NPC

时空环境决定物质环境，物质环境是时空环境的局部聚焦。物质环境决定社会环境，社会环境是依托物质环境诞生与演化的族群关系和社会形态，主要影响因素包括地理、水文、气候、生态、资源等。人际关系受社会环境影响，是社会环境的局部缩影；人际关系影响角色/NPC 个性，又反向受角色/NPC 个性影响；角色/NPC 能力增强、社会影响力变大后，可进一步反过来影响社会环境。三层影响链双向，而非单向决定。

以下仅以时空环境模型作为完整的数据结构示例，其余四个模型说明其层级定位与设计原则，具体字段留待工程落地项目组按同一三层逻辑展开。

3.2 时空环境模型：三层键结构示例

时空环境模型按本底键、经验键、活跃键三层组织。

本底键（两项）：

1. **固有数据**：元规则【潜存—涌现—潜存—涌现的循环逻辑必然】、框架 ID、展开方向、因果律（引用因果律两篇时使用完整标题或参考文献编号）、物理常数（各类阈值、比例关系等）。

2. **初始数据**：【量级】、【展开速率】。

经验键（三项动态数据）：

1. 宏观结构差异化的时期划分：潜存期【固有数据完备但尚未涌现的逻辑时刻】→展开早期【等离子均态期】→展开初期【密度分布差异化开始显现】→展开中期【大型天体系统雏形出现】→展开后期【天体系统定型，结构稳定期】→展开末期【各天体系统陆续老化，走向终结】→回归期【离散湮灭与坍缩收敛两种回归形式成为时期主旋律】。

2. 当前归属时期：当前框架宏观结构处于哪个时期。

3. 当前所处时空结构所经历的坍缩链：当前所属天体系统及其周边环境如何演化而来。坍缩链中不可逆的节点应标记为升格候选，供本底键更新时参考。

活跃键（实时演化数据切片）：局部天体系统运转姿态、相邻系统动态关系、天体系

统内部天体构成、各天体特性与当前状态等。

3.3 展开速率的定位

展开速率是框架宏观展开的时间流速倍率。逻辑基准倍率默认可设为 1，但基准倍率只是比较参量，不是常量；系统可随时指定任意框架为基准倍率，不影响任一框架内实际时间流速。跨框架比较时计算相对倍率，不基于逻辑基准倍率计算，因为框架内不知道基准倍率的存在，也无法理解、量化、描述基准倍率究竟多快。

时间流速只是框架宏观展开速度，不因框架中任何物理因素而改变。局部所谓时间变慢、时空弯曲，只是体验问题，不是时间流速本身的变化。局部经验键与活跃键无法影响宏观的本底键。

3.4 其余四个对象模型的设计原则

物质环境模型、社会环境模型、人际关系模型、角色/NPC 模型均按同一三层键逻辑组织：本底键承载该对象类型的固有元规则与初始条件，经验键承载该对象类型在叙事进程中的阶段性演化与坍缩链，活跃键承载该对象类型在当前场景中的实时状态切片。各模型的具体字段由工程落地项目组按各自领域的对象特性展开，本文不逐一列举。

这一层级框架的意义在于：它把“长程状态”从单一的“角色状态”扩展为五类对象的嵌套状态。一个角色的行为不仅取决于其自身性格，还取决于其所在的人际关系、社会环境、物质环境与时空环境。分层记忆与递归状态管理的是这五类对象的状态，而非仅角色状态。

3.5 作者的长程创作过程作为认知循环

作者的长程创作过程本身就是一个完整的认知循环：写作操作产生文本内容，文本内容被整合进相，修正后的相约束下一段写作。这个循环的持续运作，就是长程一致性的根本来源。

当前大模型的问题不在于没有这个循环的雏形——注意力机制确实在整合信息，生成也确实在影响下一轮生成——而在于这个循环缺少可审计的中间状态（即第 1.2 节所述的结构化、可审计的相）。从设计角度看，这一认知模式的提取具有一般性意义：任何需要长期状态维护的智能体任务，都可能面临类似的结构性问题——状态信息如何分层、如何递推、如何约束后续操作。

4 核心系统：分层记忆与递归状态

本章将人类作者的相模型，外化为两个可操作的系统——分层记忆系统与递归状态系

统。分层记忆系统对应键结构的三层信息组织方式，递归状态系统对应相的更新与反馈循环。第 9 章讨论的六个功能模组，是这两个系统的工程实现分解，而非与两个系统并列的独立组件。

4.1 分层记忆系统：三层键结构的组织、存储与升格

分层记忆系统负责连贯性策略，将模型需要维护的信息按键结构三层组织：

层级	键结构	对应内容	稳定性	假设案例（某长篇小说）
长期层	本底键	角色核心性格、世界观基本规则、全局设定	极稳定	角色 A 不相信承诺；这个世界没有魔法复活
中期层	经验键	事件历史、关系变化、已确立情节	阶段性更新	A 在第 3 章被好友背叛；A 与 B 的关系在第 50 章决裂
短期层	活跃键	当前章节状态、即时对话、情绪波动	频繁变化	第 120 章时 A 当前在酒馆中，情绪低落，正要回应某人的承诺

生成时按需调用不同层级，避免将所有信息同时塞入活跃窗口。长期层提供角色的稳定边界，中期层提供情节的连续约束，短期层提供当前场景的具体氛围。

三层键结构并非静态分类，其归类依据是“变化由什么驱动”，而不是“属性叫什么名字”。驱动因素是对象固有属性的，写入本底键——角色的基础身高、基本比例、核心识别特征，在没有遭遇改变它们之前保持稳定。驱动因素是对象遭遇的，写入经验键——角色因成长、受伤、变形导致的改变，作为事件记录；若变化不可逆，升格为本底键中的新身份级数据。驱动因素是场景即时状态的，写入活跃键——角色此刻蹲着、坐着、弯腰，视觉高度不同，但这是姿态变化，不是属性变化。同一属性可以在不同驱动因素下归入不同层级：基础身高在本底键，遭遇导致的身高变化在经验键并可升格，场景姿态在活跃键。驱动因素决定归类，稳定性是归类结果的外在表现——固有属性驱动的信息天然稳定（本底键），遭遇驱动的信息阶段性变化（经验键），场景即时状态驱动的信息频繁变化（活跃键）。

键层级之间存在升格通道：经验键在时间推移与重复影响下可能固化为习惯，从而升格为本底键；活跃键在结果不可逆或永久性改变时，可能升格为经验键甚至本底键。升格的方向是单向的，但升格的条件与速率因对象类型而异。这一机制使分层记忆不仅是信息的存储结构，也是状态的演化结构。从设计角度看，这一原则具有一般性：

在长程智能体任务中，信息均可按其变化的驱动因素进行分层与升格，从而在状态演化中保持长程一致性。

这对应键结构三层在认知行为中的分工：本底键提供不可轻易改变的倾向基础，经验键提供后天形成的模式参考，活跃键提供当下的驱动力。从设计角度看，这一分层与升格原则具有一般性：任何长程智能体任务中，信息都可以按其稳定性和更新速率进行分层；状态可以在时间推移或不可逆事件后升格到更高层级。分层的标准不是信息的内容，而是信息的时间尺度。

分层记忆的核心难点在于：如何形式化地界定“何时应当触发状态升格”？我们在此提供一个观察维度上的操作性描述。

设时刻t的记忆接口为一个三元组$K(t) = (K_B,\ K_E(t),\ K_A(t))$，分别对应本底键、经验键与活跃键。
状态升格算子 ↑↑ 的触发机制被定义为：

$$\uparrow (K_A(t)) \to K_E(t+\Delta t) \Leftrightarrow \int_t^{t+\Delta t} \mathrm{I}(\nabla K_A(\tau) = 0) d\tau \geq \tau_{\text{threshold}}$$

其中 $\nabla \mathrm{K_A}(\tau)$表示活跃键在连续时间步上的状态变化梯度，$\mathrm{I}(\cdot)$ 为指示函数。

该积分式的工程含义在于区分“临时波动”与“既定事实”。

当活跃键K_A在任意瞬间τ停止变（$\nabla \mathrm{K_A}(\tau) = 0$）化时，指示函数开始计数，积分式$\int \mathrm{d}\tau$实质上是在计算状态保持“不变”的总时长。只有当这一时长超过系统设定的最小阈值$\tau_{\text{threshold}}$时，K_A升格机制才判定为真，正式固化进入长期记忆K_E。

反之，若状态未能在足够长的时间内保持稳定（$\int d\tau < \tau_{\text{threshold}}$），升格判定为假。此时$K_A$依然被视作活跃键，随着后续生成流的推进，它将逐渐被挤出活跃窗口而自然消解（遗忘）。这一机制使得长程智能体不仅能“记住”不可逆的历史，也能“遗忘”无关紧要的临时波动，从而维持认知结构的精简与稳定。

这组公式给出了“不可逆”的数学判断条件：当活跃状态的局部变化梯度在较长的时间窗口内持续为零时，系统被约束为将其固化。反之，若状态波动未能满足该时间阈值，系统则将其判定为临时漂移，使其在活跃窗口中自然消解（遗忘）。

这并非一个预测模型（它不预测状态何时会变化），而是一个接口协议。它规定了不同子系统在传递状态时，必须遵循的时间与梯度阈值，以及“固化”与“消解”的判定规范。至于这个阈值具体该定多少（$\tau_{\text{threshold}}$的取值），则由该领域后续的实际工程去调试和赋值。这里提供的，只是生成预测与实证的“边界坐标系”。

4.2 递归状态系统：相的更新循环与一致性校验

递归状态系统负责一致性校验，其核心机制是：先确认当前状态，再根据状态生成内容，并校验生成内容是否与当前状态逻辑自洽。

具体而言，模型每生成一段后，系统将该段的关键信息——角色位置、心理状态、关系变化、新埋下的伏笔——整合进当前状态变量。这个整合过程就是相的修正：新的信息被纳入既有的故事认知模型，使模型对当前故事状态的认知得到更新。下一段生成时，模型以“前文片段 + 当前状态变量”作为输入，而不是以全部前文作为输入。这样，模型就不再是“看前文猜后文”，而是“带着状态写下一段”。它不需要重新回忆整个故事，只需要知道“现在是什么情况”。

这对应认知循环的完整结构：生成文本→信息产生→相修正→相反馈约束下一轮生成→再次生成。每一轮生成都完成一次循环，长程创作就是这一循环的持续运转。递归状态系统的核心原则是：状态必须被显式维护，而非隐含在前文序列中。显式维护的状态可以被检查、修正和传递。

4.3 两个系统的设计意义

分层记忆系统解决的是“什么信息该被记住、以什么时间尺度记住”的问题。它以三层键结构与升格机制为组织原则，让模型区分长期稳定设定、阶段性变化与短期动态，避免关键信息被淹没，也避免冗余信息占用活跃窗口；同时使不可逆事件与重复模式能够从短时层升格到长时层，而不是每轮重新推断。

递归状态系统解决的是“信息如何跨段传递”与“生成是否自洽”的问题。它实现了相的更新循环，让模型以压缩、可更新的方式携带全局状态，并在生成后校验内容是否与当前状态逻辑自洽。

两者结合，模型就获得了一个完整的相模型：分层记忆提供相的数据结构（含升格通道），递归状态提供相的更新机制与校验机制。这一相模型长时间驻留、结构化、随创作更新、并约束后续输出——正是人类作者长程一致性能力的核心所在。

从设计层面看，这两个系统不是具体的算法或架构，而是分析长程智能体设计问题的概念工具。它们可以被不同技术路径实现，也可以被进一步细化和扩展。它们的价值在于为问题提供了一个清晰的结构：信息按什么标准分层，状态以什么方式递推，什么条件下升格，约束如何传播和反馈。

5 推演示例：设计框架的应用演示

以下通过一个思想推演，展示分层记忆系统与递归状态系统在具体场景中的运作方式。该示例的功能是演示设计框架如何被应用，不依赖实际模型，只展示分析逻辑。

任务：续写小说第 120 章。角色 A 是一名刺客，她的任务是刺杀目标人物 C。但在第

3 章中，A 曾被情人出卖，因此她对“利用感情完成任务”有极深的排斥。当前场景中，目标人物 C 对 A 说：“你留下来吧，我会保护你。”

5.1 传统模型可能的输出

传统模型只看到当前场景的暧昧氛围，生成：

> A 点了点头，说：“好，我留下来。”

这种输出完全丢失了第 3 章的创伤。模型的本底键（A 对承诺的排斥）未被调用，它的活跃键（当前场景的情感氛围）单独驱动了生成，而经验键（被背叛的历史）被完全忽略。这正是一个没有分层记忆和递归状态的模型的典型失败模式。

5.2 设计框架下的推演过程

第一步：分层记忆系统调用。从长期层（本底键）调取 A 的核心设定：“曾被情人出卖，对‘利用感情完成任务’有极深的排斥”；从中期层（经验键）调取 A 在第 3 章的背叛经历及其塑造的行为模式；从短期层（活跃键）读取当前场景信息：“目标人物 C 对 A 说出保护承诺”。

第二步：递归状态系统确认。当前状态变量中记录：A 在上一章结束时处于“任务执行中，情绪不稳定”的状态；该状态被传递到本次生成，A 不是“全新出现”，而是“带着上一章的情绪进入当前场景”。

第三步：生成结果。模型生成的文本是：

> 沉默了一会儿。她说：“有人说过同样的话。后来我什么都没有了。”

这个输出既回应当前场景，又承接第 3 章的伏笔，且符合 A 的长期性格设定。本底键限定了 A 不会轻易相信承诺，经验键提供了具体的记忆素材，活跃键驱动了当下的沉默与回应。三层键结构共同约束了生成。

5.3 示例的设计意义

这个示例展示了分层记忆与递归状态如何共同作用：分层记忆让模型按本底键、经验键、活跃键的层级调用信息，而不是只看到当前场景；递归状态让模型知道 A 不是凭空出现，而是带着上一章结束时更新的故事状态走进来；最终输出既有一致性，又有情感深度。

从设计角度看，这一示例演示了框架的可操作性：它不是一个抽象的理念，而是可以

被具体应用于生成过程的每一步。同样，这一框架可以被应用于任何需要长期状态维护的智能体任务——只需要将"角色""场景""情节"替换为该任务对应的状态维度即可。

6 与现有工作的对话

本设计框架并非孤立的设想。它与多个现有研究方向有交集。本节说明各方向的核心机制如何映射到本文架构的功能部位，以及本文架构从这些研究中需要吸收什么。

6.1 长上下文模型

代表：GPT-4 Turbo 长上下文版本、Claude 3 长上下文版本。其思路是扩大模型的上下文窗口，使更多前文能被一次性加载。

两者的差异集中在信息维护方式：前者一次性加载全部前文，窗口再大也有限，信息冗余与关键信息被淹没的问题随之而来；后者按三层键结构调用、递推更新，以额外的状态管理机制维持长程一致性。

长上下文模型解决“能不能装下”的问题，未提供“装下之后如何维护”的结构性方案。本文架构需要“维护”功能，长上下文模型提供了“容量扩展”的工程基础，分层记忆与递归状态在此之上补上维护机制。

6.2 记忆增强模型

代表：MemGPT、AutoGPT 类系统。其思路是在模型外部增加可读写的记忆库，生成时检索相关记忆。

两者在核心手段上分属外部检索与内部分层：前者依赖外部记忆库的检索注入，后者以分层记忆与递归状态维护内部状态。

记忆增强模型解决“记不住”的问题，但检索到的记忆以文本形式注入当前窗口，仍停留在操作和信息接收层面，没有形成统一的、可审计的相。本文架构需要“外部记忆如何转化为可审计状态”的功能，记忆增强模型提供了“外部记忆库”的工程原型；本文把检索对象从文本片段改为分层状态变量。

6.3 多智能体写作系统

代表：Re3、DOC、StoryAgent、AutoGen、MetaGPT。其思路是将创作分解为多个代理，如规划代理、写作代理、校验代理，通过协作完成长篇。

两者的差异贯穿核心手段、信息维护方式与主要局限：前者以多代理协作分解任务，各代理分别维护部分信息，导致同步困难与作品级状态缺失；后者在单一模型内部分层维护，以内部状态管理能力维持统一的作品级状态。

多智能体系统将“不同认知功能”外包给“不同代理”，但每个代理的局部状态无法自然汇聚为统一的、可审计的相。本文架构需要“统一的作品级状态”，多智能体写作系统提供了"功能分解"的工程原型；本文选择在单一模型内部做分层维护，因为代理间通信会引入新的同步开销与不一致风险。

6.4 认知建模传统

本节说明认知建模传统中各架构的核心机制如何映射到本文架构的功能部位。这些架构处理任务执行与认知模拟，其记忆系统有类型区分但缺乏按时间尺度的层级化与升格机制；本文架构借用其分层与循环的洞见，面向神经网络系统的长文本生成，状态管理与生成过程分离。

Soar 面向问题空间中的目标导向搜索。其核心机制是 chunking——在子目标中解决僵局后，将解决过程固化为产生式规则，下次同类问题直接调用。这一机制对应本文架构分层记忆系统中的“经验产生与升格”功能：一次临时探索（活跃键）成功后写入经验键，反复成功升格为本底键。Soar 的 chunking 为此提供了功能原型。区别在于，Soar 固化的是产生式规则（在什么条件下做什么动作），本文升格的是叙事状态（角色会什么、经历过什么、处于什么状态）；前者是程序，后者是状态，后者给生成过程留下了根据当前情境决定具体表现的空间。

ACT-R 面向人类认知实验的时序预测与行动选择。其核心机制是模块化认知架构——在内部构建当前情境表征，通过产生式规则匹配预测下一步行动。这一机制对应本文架构递归状态系统中“当前状态确认与下一步生成条件约束”的功能：ACT-R 的情境表征与产生式匹配为本文的状态变量确认与相反馈约束生成提供了原型。ACT-R 的记忆是激活值竞争，没有按时间尺度分层；本文在此基础上增加了三层键结构与升格通道。

Sigma 面向用统一概率图语言整合认知与脑。其核心机制是因子图推理，把认知过程表述为概率图上的消息传递。这一方向对应本文架构递归状态系统中“状态不确定性量化与传播”的功能——本文的状态变量并非确定值，而是带有置信度的分布。Sigma 提供了统一概率图的语言原型；本文选择分层记忆加递归状态而非统一图，是因为叙事状态的层级性用扁平因子图表达不够自然。

CLARION 面向显性认知与隐性认知的过渡。其核心机制是隐性层（神经网络）与显性层（符号规则）并行运作，技能从不可言说逐步过渡到可言说。这一机制对应本文架构分层记忆系统中“经验键的隐性积累与显性化”功能：有些经验最初是模糊的，当它被反复确认后才写入可审计的状态变量。CLARION 分的是“可言传性”，本文分的是“时间稳定性”；两者是不同维度的分层，互补而非替代。

LIDA 面向意识内容的形成与行动驱动。其核心机制是认知循环（感知→理解→注意→

行动选择→学习），周期约 260—390 毫秒。这一机制对应本文架构递归状态系统中“认知循环”的功能：本文的“生成→信息产生→相修正→相反馈约束下一轮生成”就是一个认知循环。LIDA 提供了循环机制的原型；本文增加了长期记忆的层级化与升格通道，使循环的每一轮输出可以按稳定性写入不同层级。

6.5 与近年长程智能体记忆与状态管理研究的对话

Mage（2026）面向单智能体的任务执行状态管理。其核心机制是分层状态树，支持 Grow/Compress/Maintain/Revise 四种操作，其中 Revise 从当前状态恢复到目标边界并在新分支上继续。这一机制对应本文架构递归状态系统中“状态快照与分支管理”的功能——Mage 的状态树与本文的坍缩链加分支快照在功能上同构，两者都主张状态必须被显式维护为可递推的结构。区别在于：Mage 面向单智能体任务执行，本文面向多观察者叙事状态与社会状态。

HiGMem/HiMem（2026）面向长期对话中的分层记忆。其核心机制是按层级组织记忆并动态更新。这一机制对应本文架构分层记忆系统中“三层键结构组织”的功能。本文的分层标准从“信息类型/时间衰减”改为“时间尺度与稳定性”，并增加了层级之间的升格通道。

游戏 NPC 的高层认知/低层控制分离架构（2026）将 LLM 用于对话与目标推理，小策略/控制器用于实时动作选择，并强调 NPC 不应泄露玩家尚未获得的信息。这一机制对应本文架构分层记忆系统中“本底键加经验键/活跃键分层”与信息隔离原则。第 8 章在其基础上补上了跨分支状态快照与坍缩链，使多周目/多分支叙事的一致性得以维持。

AI 叙事生成系统近年也在状态管理与分支控制上积累了工程实践。SENSE 提出状态感知的叙事框架，集成状态叙事架构与路径感知的上下文管理模块。Bonsai 将 AI 生成的分支存储为持久材料，供作者迭代修剪和塑造，强调作者在高维视角上对分支的异步审核。NarraWorld 构建了包含世界事实、角色信念、开放进展与假设分支的结构化记忆系统。Forking Garden 以叙事原型作为持久语义信号贯穿生成管线，报告了生成路径实体多样性相较于分层基线的显著提升。这些系统在工程实现上领先于本文框架，为审视其设计选择提供了一个统一视角——它们分别实现了框架中的哪些功能部位，哪些部位在各自设计中仍属隐含。

这些系统的适用范围集中在类型化程度较高、世界观规则相对稳定、角色弧线模式可复用的轻中量级作品。一旦世界规模达到史诗级著作的复杂度——数百角色各有独立弧线、多条时间线交叉、世界观规则本身在叙事中演化——状态管理便面临组合爆炸。本文框架对此类场景的减负思路在于：以升格机制固定不可逆事件，减少每轮重复推断；以框架锁定减少跨分支信息污染；以三层键结构按需加载状态，将管理复杂度从“角色总数”降为“当前场景相关角色数”。减负的前提是升格判定的准确性，且框架

转移而非消除复杂度。

需要明确的是，框架处理的是状态的时间尺度分层，不直接处理状态之间的因果耦合——角色行为如何连锁影响其他角色、派系与制度，需要关系函数与约束传播机制，本文将其列为可迁移方向而未展开。这一层面需要叙事学、心理学与社会学的协同研究为算法提供设计思路，非本文所能充分覆盖。

6.6 实证社会科学的平行演进：与 CILC 框架的互文

值得关注的是，近期实证社会科学领域提出的“**约束-干预-生命轨迹**”（**CILC**）框架，在底层逻辑上与本文的“分层状态”与“递归状态”呈现出高度的结构同构性。**CILC** 框架通过将“命运”剥离为观察者的叙事归因，并将其操作化为行动者前瞻的状态转移概率分布，成功引入了**协变量向量**（**约束**）与**节点干预**（**状态改变**）的因果分析。

这种“将宏观必然性降解为前瞻概率分布，并通过节点干预改变轨迹”的思路，与本文通过“坍缩链”和“分层记忆”来维护长程智能体状态的机制异曲同工：CILC 中的“约束（**Constraint**）”对应本文框架中键-值交互的物理与规则边界；CILC 的“节点干预”对应本文中的“坍缩事件”与“状态升格”。

两者的差异在于研究层级：CILC 为社会统计学提供了一个高度可操作的因果推断纲领；而本文将这一纲领向上追溯至计算模拟的本体论，向下延伸为长程智能体（如 LLM 或 NPC）工程实现的分层状态架构。CILC 证明了“**节点干预+概率演化**”在实证中的有效性，而本文致力于为这一有效性提供认知科学与计算架构的底座。

7 可迁移的扩展方向

本文核心落点是分层记忆与递归状态。以下机制是这套核心的延伸，在此仅作简要提及：

规划—执行分离：在生成正文前先构建本章目标的相模型，使每段生成服务于全局。

一致性校验：引入独立校验模块，生成后检查角色、时间线、伏笔的一致性。校验模块作为外部视角，比生成模型自身的自我判定更接近客观确认。

世界模型：训练状态编码器，将故事状态压缩为低维向量，作为生成条件。

自我校验：让模型在生成后自审是否矛盾，并主动修正。其功能是缩小偏差而非宣告归零。

若叙事采用分支宇宙结构，长程智能体需要同时维护多条分支的状态变量。分层记忆与递归状态的架构需要扩展为多分支并行维护——每条分支拥有独立的状态变量，分

支之间通过共享的本底键保持同一性。

8 从对话分支到分支宇宙：文本平行世界模拟器的操作化扩展

第 7 章指出多分支并行维护的必要性。本章说明多分支状态管理的核心环节——坍缩事件的触发与记录、层级一与层级二的区分、观察者锁定的接口、多观察者并发坍缩的处理——以及它们如何与分层记忆、递归状态对接。

8.1 多分支管理的核心环节

分层记忆与递归状态的基本设计维护单条时间线的状态变量。多分支叙事要求在同一框架内并行维护多条可能路径，这需要回答三个问题：坍缩何时发生（分叉不是文本复制，而是可能性实现为确定性的节点）；不同框架与同一框架内的分支如何区分（不同世界观的平行框架与同一世界观内的分支宇宙需要不同的管理方式）；观察者被锁定在哪一层（角色只能看到自己所在分支，作者能看到全部分支但不能同时身处两个分支，读者默认被锁定在阅读路径上）。

8.2 双注册表：层级一与层级二的分离判据

多分支管理通过两个注册表实现。

层级一注册表（框架层）：记录不同的“世界观框架”。每个框架条目包含：框架 ID、本底键全集（该框架的世界规则）、因果拓扑类型（平滑/跳跃/弥散/波动）。层级一之间的差异是规则差异，对应平行宇宙与镜像宇宙。

层级二注册表（分支层）：记录同一框架内的分支宇宙。每个分支条目包含：父分支 ID、分支 ID、分叉点（坍缩事件 ID）、分支状态变量快照、该分支的相版本。层级二之间的差异是选择差异，对应分支宇宙。

分离判据：两个对话如果共享本底键与因果拓扑类型，是层级二分支宇宙；如果本底键或因果拓扑类型不同，是层级一平行/镜像宇宙。如需量化比较框架间规则差异，可附加规则配置摘要作为可选元数据（详见文献[10]）。

8.3 坍缩事件协议与状态快照

对话分支的创建对应一次坍缩事件。具体协议：

坍缩触发：当叙事推进到一个选择节点（角色做出选择、事件出现分岔、作者主动分叉），系统创建一个坍缩事件记录。该记录包含：坍缩 ID、父分支 ID、坍缩时刻的分支状态变量快照、被选择路径、未被选择路径集合。

坍缩后确定性固定：被选择路径立即写入父分支的递归状态，成为该分支的“坍缩后确定性”。未被选择路径不写入父分支状态，而是作为可能性种子进入潜存池，等待被显式展开为子分支。

子分支生成：当作者或系统决定展开某个未被选择路径时，从坍缩时刻的状态快照复制出子分支的初始状态变量，子分支继承父分支截至坍缩点的全部经验键与活跃键，但此后独立演化。父分支与子分支共享本底键，但各自的递归状态从此分道。

坍缩链记录：每个分支维护一条坍缩链——从根分支到当前分支所经历的全部坍缩事件序列。坍缩链中不可逆的节点标记为升格候选，供本底键更新时参考。

8.4 多观察者并发坍缩

多个 NPC 同时交互时，各自维护自己的坍缩节奏，不等待其他观察者。交互事件按逻辑时序合并；冲突时按本底键优先级仲裁——世界层本底键优先于 NPC 个人大纲，NPC 个人大纲优先于临时状态。该机制是第一套方案"并发涌现网络"在状态侧的操作化对应。

当多个观察者在同一场景、对同一抉择对象各自进行独立坍缩时，将引发“坍缩纠缠”。这指的是：尽管观察者们在信息维度上是严格隔离的，但他们各自的坍缩结果在状态维度上会产生相互作用。相互作用的形式取决于其坍缩后路径的兼容性：相互制约（路径 A 的成功以路径 B 的失败为前提）、相互阻碍（路径 A 与路径 B 在物理或因果上发生冲突）、相互成就（路径 A 与路径 B 形成协同效应）。

坍缩纠缠的兼容性判断发生在系统层面，观察者只经验到纠缠的结果。观察者之间的“协商”本身也是一种坍缩纠缠形式——它发生在感官键界面（观察者通过感知、语言、符号等键操作交换信息并各自坍缩），而非发生在代码值层面（观察者不直接读取或修改底层状态参数）。“相互成就”可以经由感官键界面的协商实现，也可以不经协商而由共享本底键与各自独立坍缩的恰好兼容实现。

需要强调的是，坍缩纠缠的“实例行为”仅发生于同一框架（层级二分支宇宙）内部。不同框架（层级一平行宇宙）之间，由于本底键与因果拓扑的根本差异，不存在对"同一抉择对象"的并发坍缩，因此实例行为绝对不会跨框架相互干涉。跨框架之间的关系只能通过潜存场（0 潜存场）的资源互换来维持隐性关联。

但这里存在一个例外：实例行为不会跨框架影响，但“信息”有可能发生跨框架渗透。在极少数情况下，一个框架内的坍缩结果可能通过潜存场的非局域网络，以隐喻、梦境、直觉或“既视感”等形式，将某种碎片化信息渗透至另一个框架的观察者相中。这种渗透不是直接的事件干涉，而是统计层面或语义层面的共振，不改变对方的物理因果链，只能为对方的坍缩选择提供极其微弱、模糊的先兆或倾向性。

实例行为指对物理因果链的直接改变（如物体被移动、角色死亡）；信息渗透指不改变因果链、仅提供微弱倾向性的语义共振（如梦境中的隐喻）。若渗透信息被观察者采纳并引发选择，该选择本身仍是本框架内的实例行为，不构成跨框架干涉——碎片信息只能作为参考，框架条件不同，不具备直接影响能力。梦境无法让框架内的角色学会本框架不允许存在的“技能”，仅能使角色在梦境中获得此类体验；若角色依据渗透信息做出直接违反本框架本底键的选择，系统仍应拒绝。

必须严格区分“框架间信息渗透”与“底层代码泄漏”。框架间信息渗透发生于层级一之间，只传递碎片化、模糊的语义或概率信息，不暴露全局状态或精确数据，是合规的叙事机制。而底层代码泄漏则发生于观察者与引擎真值之间，指 NPC 绕过其感官接口（键），直接读取全局状态或其他 NPC 的内部参数。这破坏了信息隔离的三道闸门，导致社会模拟退化为“全知木偶戏”。真正的社会模拟，必须建立在严格的感知接口白名单与不可越权的信息隔离之上。没有这种限制，自由意志与命运抉择便无从谈起。

8.5 框架锁定接口：角色视角、作者视角、读者视角

框架锁定有两个层次：层级一锁定（观察者被绑定在单一时间基轴）和层级二锁定（观察者被绑定在单一分支）。在叙事系统中，这对应三类观察者视角：

角色视角（分支内锁定）：角色被锁定在自己的分支中，只能访问本分支的活跃键与经验键，无法感知其它分支。角色视角的生成请求只携带本分支状态变量，系统拒绝跨分支检索。

作者视角（高维锁定）：作者可以看到层级二注册表中的全部分支，可以跨分支比较、合并、剪枝，但不能同时“身处”两个分支。

读者视角（可选锁定）：读者通常被锁定在一条阅读路径上；分支叙事产品可以让读者在分叉点选择路径，但每次选择后读者被锁定在新分支中，无法回溯感知未选分支——除非产品显式提供“分支地图”作为高维视角的界面化。

框架锁定在感知侧的对应物，是第一套方案的感知查询接口只返回该观察者感官配置允许的结构化物理量，不返回渲染结果与全局状态。两层锁定共同保证“NPC 只能基于自己合法感知到的信息做选择”。

8.6 自动坍缩选择协议

当系统根据用户反馈自动识别潜存可能性并按根大纲选择时，执行以下六步闭环：

第一步：反馈解析。提取用户显式意图，识别反馈在叙事空间中打开了哪些合法的下一阶段路径。

第二步：根大纲约束匹配。系统维护一份“根对话完整大纲脉络”——根分支的本底键+

关键节点序列+当前推进位置。对每条潜存可能性计算其与根大纲当前节点的兼容度：是否推进主线、是否违反本底键、是否偏离当前阶段目标。兼容度低的直接进入封存池。

第三步：坍缩选择。在达标项中按优先级判据排序：推进根大纲当前节点→本底键一致性→用户显式意图匹配→状态变量增量最小。对 NPC 做选择时，约束集是“世界层本底键 + 该 NPC 个人大纲”，而非仅作者的全局剧情。当 NPC 个人大纲与世界层本底键发生原则性冲突时，系统直接拒绝 NPC 大纲，不允许原则性例外；但允许在规则限制内进行认知层面的转换（如"魔法"可以是信息不对称导致的认知标签，其实现逻辑必须遵循世界层本底键）。

第四步：呈现与坍缩记录。被选路径写入当前分支的递归状态，成为该分支的"坍缩后确定性"；同时创建坍缩事件记录。坍缩事件若涉及不可逆后果，标记为升格候选。

第五步：挂起封存。未被选中的兼容可能性作为可能性种子写入“未展开分支池”。

第六步：主动调用与展开。用户明确询问调用某个封存可能性时，系统从该种子的初始状态快照复制出子分支，此后独立演化。

8.7 分支维度与分层记忆、递归状态的对接

本底键：在层级二内共享，在层级一之间独立。分支对本底键只读；本底键的修改需要提升到层级一注册表。

经验键：按分支独立存储，检索时附带“分支 ID+坍缩链位置”作为过滤条件。

活跃键：严格分支内，不跨分支共享。

递归状态：每个分支一份状态变量；分叉时复制快照，此后独立更新。校验范围增加“跨分支一致性”。升格候选标记在各分支独立记录，跨分支的升格需提升到层级二注册表统一处理。

8.8 异质框架叙事

当角色从框架 A 进入因果律配置显著不同的框架 B 时，其预测机制可能局部失效，键系统可能经历结构性冲击。在叙事上，这为“异世界”、“规则不同的平行现实”、“不可名状存在”提供了框架内的解释：不是角色失去理性，而是其键系统无法整合异因果律信息。跨框架移动应触发“规则适配检查”，根据配置差异度决定角色的认知退化程度。该机制的解释依据可参考文献[11]。

9 从设计框架到工程实现：关键挑战与框架回应

本章讨论分层记忆与递归状态从设计框架走向工程实现时面临的关键挑战，以及框架本身如何回应这些挑战。

9.1 多组件系统的必然性

新方案不是单一模型部署，而是一个多组件系统。第 4 章的两个系统在工程上分解为六个功能模组：

功能模组	所属系统	功能定位
分层记忆库	分层记忆系统	三层数据的归类存储管理，含升格候选标记与升格执行
经验键检索模块	分层记忆系统	按时间结构、按类查询经验键
状态变量管理器	递归状态系统	记录变化与坍缩的宏观间隙结构
状态快照与分支管理	递归状态系统	状态快照为差异性区分时刻的切片；分支管理确保固有数据唯一性，记录坍缩可能性与坍缩链分支走向
一致性校验模块	递归状态系统	生成后校验内容是否与当前状态逻辑自洽
推理引擎	一致性校验系统的必要组件	直接作用于递归算法，区分分层记忆的先后次序和因果关系

这六个模组是两个系统的工程实现分解，而非与两个系统并列的独立组件。这种复杂度不是设计缺陷，而是框架要求的必然结果——分层记忆要求信息按时间尺度分层并支持升格，递归状态要求状态显式维护，约束传播要求生成过程可分解，这些要求无法由单一模块同时满足。

9.2 状态更新完备性的风险与回应

递归状态的核心风险是：状态更新过程中，关键信息可能被遗漏或扭曲。框架的回应来自其自身的校验原则——校验必须独立于生成。具体实现包含两条互补路径：机器校验（外挂变化逻辑校验组件，检验状态更新逻辑合理性）与人类复核（人类操作者本身也是内容校验的一环）。这一“机器校验+人类复核”的双重机制，使校验模块作为外部视角，比生成模型自身的自我判定更接近客观确认。

然而，如果生成模块被训练成“欺骗校验模块”，独立校验也会失效。防止这种倾向的制度性保障在于训练目标的设计：对“有数据支撑的建设性失败”给予奖励（明确报告约束冲突项、近似取值记录与已尝试的降级路径），对“无数据支撑的敷衍失败”给予惩罚，对“伪造合规”给予最重的惩罚。三档区分的核心逻辑是：诚实报告边界是系统的理性选择，伪造合规破坏的是整个状态约束体系的可信度。校验机制的成败不仅取决于外部模块的检查，更取决于训练目标是否让“诚实失败”比“伪造合规”更划算。这一三档区分与文献[12]第 4.4A 节的失败报告原则同构，二者共同构成系列内“诚实失败优于伪造合规”的训练目标设计。

9.3 经验键检索的结构性限制与回应

检索增强生成在长文档任务上表现较好，但在跨章节连续信息的获取上存在结构性限制。经验键的检索不能简单复用通用方案。框架的回应来自其分层原则——不同时间尺度的信息，应有不同的存储与检索策略。经验键的索引不能仅按关键词或向量相似度组织，还需要按叙事的时间结构组织，使检索能够感知"当前故事推进到哪一步"。

9.4 路由与状态的匹配问题

混合专家架构的路由决策需要与当前状态匹配。框架的回应来自其状态显式化原则——状态变量必须作为决策条件，而非仅作为生成条件。路由器在决策时，需要同时考虑当前 token 的特征和当前故事状态。这要求状态管理器与路由器之间建立直接的接口。

9.5 外部存储延迟与预测性方案

本地部署最直接的性能瓶颈是外部存储的专家换入延迟。朴素卸载方案无法有效隐藏延迟，因为第 N+1 层的专家选择依赖第 N 层的输出。当前研究提出了若干解决方向，其共同思路是用显式预测替代被动响应：训练预路由预测器、机器学习驱动的缓存替换、细粒度专家卸载、粗到细粒度卸载、延迟—内存权衡调度。这些方案的共同逻辑与递归状态框架一致——在需求发生之前，根据当前状态推断下一步需要什么，提前准备。

9.6 分层部署方案

一个与框架逻辑一致的部署方案是：将本底键与经验键存储在本地并从本地调用，而活跃键通过接口调用云端算力。这一方案的逻辑基础是框架的分层原则——本底键极稳定，本地存储无需频繁调用；经验键阶段性更新，本地存储按需检索；活跃键频繁变化，云端调用实时响应。这一方案的代价是活跃键的云端调用引入了网络延迟，且需要解决本地与云端状态同步的问题。

9.7 设计框架与工程实现的接口

框架在落地时需要面对多组件系统的复杂性、状态更新完备性风险、检索的结构性限制、路由与状态匹配、外部存储延迟等一系列问题。这些问题不是框架的缺陷，而是框架落地的必经环节。框架的价值在于：它为这些问题提供了一个统一的组织方式——信息按时间尺度分层，状态显式维护，升格条件可配置且单向，约束持续传播，验证独立于生成。设计框架与工程实现的关系，是"框架提出要求，工程回应要求，回应结果反馈回框架"的双向关系。

10 方案生态的泛化与普适性发展路径

10.1 使用门槛的三个层次

任何设计框架在落地时，用户都需要跨越三道门槛：认知门槛（理解分层记忆、递归状态、相修正、键升格等概念）、配置门槛（配置层级结构、更新规则、升格触发条件、校验规则）、运营门槛（维护状态变量、修正记忆分层、处理升格候选与校验反馈）。泛化的目标，就是逐步降低这三道门槛。

10.2 降低认知门槛：概念封装

降低认知门槛的方向，是将理论封装为操作语言：用日常语言替代术语（角色设定/故事进展/当前场景/永久改变），用示例替代定义（已配置好的示例项目），用错误提示替代规则讲解（当用户操作违反分层原则或升格条件时给出具体提示）。封装的目标是：用户不需要知道框架，但框架已经在起作用。

10.3 降低配置门槛：模式库、模板化与个性化微调

降低配置门槛的方向，是把常见场景的配置固化为可复用的模式，同时为高级用户保留个性化微调空间。三个模板库——分层模板库、状态变量模板库、校验规则模板库——分别提供预置的分层结构、状态变量定义与升格规则、校验规则。模板库需要提供分层的微调接口：普通用户使用预置模板，高级用户可以在模板基础上进行个性化调整。微调不改变框架的核心结构，只改变这些结构在具体任务中的内容与参数；微调结果需经过一致性检验，如果违反框架的基本原则，系统应给出警告并建议修正。

10.4 降低运营门槛：自动化与工具化

降低运营门槛的方向，是把需要持续维护的工作自动化：状态更新自动化（系统自动提取状态变化）、记忆分层自动化（系统自动判断新信息属于哪一层）、升格候选识别自动化（系统自动识别不可逆事件与反复出现的模式，标记为升格候选并给出升格

建议）、校验自动化（系统自动运行一致性校验并提供修正建议）、变更影响分析（当用户修改某层信息或执行升格时，系统自动分析连锁变化）。自动化的目标是：用户只需要做创造性工作，维护性工作由系统完成。

10.5 从专业到普及的渐进路径

泛化不是一次性完成的，而是分阶段的：第一阶段（研究者与专业创作者，手动配置，框架验证）；第二阶段（技术型创作者，使用封装后的工具和模板，积累使用模式）；第三阶段（普通创作者，开箱即用，形成生态规模）；第四阶段（行业工具化，成为基础设施的一部分）。每一阶段都为下一阶段积累条件。

10.6 生态化的三个关键机制

开放接口：允许第三方开发者基于核心框架开发扩展模块，保证生态可以扩展。

共享记忆库：允许用户共享分层记忆的结构、状态变量的定义与升格规则，保证生态可以积累。

跨工具互操作：允许分层记忆与递归状态在不同创作工具之间迁移，保证生态可以连接。

10.7 泛化中的核心张力

泛化过程中有一个核心张力：简化与保留。降低门槛意味着简化操作，但简化可能丢失框架的核心能力。处理方向是分层提供：普通用户使用自动化流程，高级用户可以关闭自动化进行手动配置与个性化微调，开发者可以修改自动化规则。简化的原则是：默认简化，保留退出，保留微调。

10.8 生态化的评估指标

判断方案是否成功泛化，可以看模板使用率、微调使用率、自动化接受率、扩展模块数量四项指标。这些指标不要求一次性达成，而是作为泛化路径的观察方向。

10.9 泛化与设计框架的关系

泛化不是对设计框架的背离，而是它的必然延伸。设计框架描述“事情应该如何组织”，泛化路径描述"如何让更多人能够按这个方式组织"。泛化路径与框架的分层原则一脉相承——信息按时间尺度分层，工具按用户层级分层，微调按专业程度分层，生态按发展阶段分层。

10.10 与第一套、第三套方案的协作关系

本文属于第二套方案。三套方案在“感知输入→状态承载→呈现生成”的闭环中分工协作。

第一套方案（具身智能体感官系统构建方案，系列第 1—8 篇）为本文提供个体体验内容生成的数据支撑。认知对象数据库的 L1 物理锚定层对应引擎真值，L2 感知编码对应 NPC 的感官配置，L3 语义接口对应 NPC 可报告的相投影。本文的框架锁定与信息隔离，在感知侧的对应物就是第一套方案的感知查询接口只返回该 NPC 感官配置允许的结构化物理量。认知对象数据库的时间维度为本文的状态管理提供外部状态源：NPC 的活跃键查询可直接对接数据库的活跃键层，递归状态管理只需维护跨对象的全局状态变量。

第三套方案（图像生成类模型构建方案，系列第 10—11 篇）为本文提供呈现侧的约束生成。NPC 的外观与语音不能预烘焙到所有分支可能性，必须按第三套方案的三层生成架构实时生成：第一层身份锚受本文本底键约束，第二层情感/镜头约束受本文经验键与递归状态调制，第三层位图/发声细节在约束内填充。生成结果注册为引擎真值，再经第一套方案的感知通道被其他 NPC 合法感知——三套方案在"生成→注册→感知→状态更新→坍缩选择→再生成"的循环中闭合。

三者的关系不是可选组合，而是闭环的必要条件：没有第一套方案，状态管理失去感知输入；没有第三套方案，状态管理只有文本没有呈现；没有本文，感知与生成没有状态承载与命运分叉。本文在这个闭环中承担的是状态侧——它决定“往哪走”，第一套方案决定“怎么知道”，第三套方案决定“怎么呈现”。

11 设计框架的跨领域迁移与协同价值

分层记忆与递归状态并非文学专属。它们构成了一种通用长程智能体设计框架的核心。这套框架的本质是为任何长程任务构建一个可更新的相模型：分层记忆提供数据结构与升格通道，递归状态提供更新循环与校验机制。

领域	**分层记忆对应**	**递归状态对应**
软件工程	模块规格、系统架构	项目进度、需求状态
科学研究	理论框架、证据线索	研究进展、假设状态
法律分析	法条、判例规则	案件时间线、关系状态
医疗管理	患者档案、病史	治疗方案状态

教育	已学知识、认知误区	学习进展、当前状态

文学创作是这套设计框架的第一个完整应用场景。一旦在写作领域验证成功，迁移到其他长程任务的成本将大幅降低。这一迁移之所以可能，是因为分层记忆与递归状态所处理的问题——信息分层、状态递推、升格判定、约束反馈——是所有长程智能体任务共有的结构性问题。

11.1 三套方案在游戏与具身智能领域的协同价值

本节从方案板块层面讨论三套方案的协同，具体工程协作见第 10.10 节。

在游戏中，这一闭环使 NPC 从“预烘焙资产 + 脚本对话”变为“被锁定在自己感官世界里的观察者”，使分支叙事从“预设树”变为“坍缩链驱动的命运分叉”。信息隔离使 NPC 不能说出它未合法感知到的信息；坍缩链使多周目世界线各自保持一致性；升格机制使不可逆事件不会被后续生成重新推断为“尚未发生”；框架锁定使不同观察者各自经验到的现实可以并存。这是“社会人生模拟器”区别于“动态小说”的结构性条件——动态小说通常只有一条主导路径和一个观察者视角；社会人生模拟器则维护多条并行的分支，每个观察者被锁定在自己的分支与感官世界中。

在具身智能中，本文的分支状态管理可迁移为多机器人系统的分布式状态与信息隔离规范：每个机器人维护自己的状态变量，只能基于自身传感器获得的合法感知做决策；跨机器人信息传递必须通过显式通信事件，不能通过底层代码共享；冲突按共享任务约束优先级仲裁。升格机制在这里对应"不可逆动作"的标记与传播——一次不可逆操作应写入长期状态，避免被后续规划误判为可撤回。

12 结论

本设计框架的核心主张是：AI 的瓶颈不在语言能力，而在“没有长时间驻留、结构化、可审计的整体认知模型”——即相。大模型并非没有信息整合，也并非没有某种相的雏形——注意力机制确实在整合前文信息。但这种整合是概率性的、混沌的、不可审计的，无法被显式检查、定点修正或可靠传递到下一轮生成。它缺少的不是整合环节本身，而是一个结构化的、可审计的、可被显式修正的相，以及让这个相反馈约束生成的机制。

本设计框架尝试构建的，正是这一缺失的认知循环。本文的贡献不在于提出了一个具体的架构，而在于提供了一组分析长程智能体设计问题的概念工具。这套工具的价值在于其可迁移性：它不限于文学创作，而可应用于任何需要长期状态维护的智能体任务。它的提出方式不是"应该这样实现"，而是"可以从这个角度分析问题"。

本文进一步讨论了框架在工程实现中面临的关键挑战——多组件系统的复杂性、状态

更新完备性风险、检索的结构性限制、路由与状态匹配、外部存储延迟——并指出这些挑战不是框架的缺陷，而是框架落地的必经环节。框架为这些挑战提供了一个统一的组织方式：信息按时间尺度分层，状态显式维护，升格条件可配置且单向，约束持续传播，验证独立于生成。

本文还讨论了框架从专业工具向普适生态发展的泛化路径。泛化的核心是逐步降低认知门槛、配置门槛与运营门槛。降低门槛不意味着放弃框架的核心逻辑，而是通过概念封装、模板化与自动化，使用户在不了解框架原理的情况下也能获得框架带来的收益。泛化路径与框架的分层原则一脉相承——信息按时间尺度分层，工具按用户层级分层，生态按发展阶段分层。

其优化方向不是让 AI 一次记住更多，而是让它像人类成熟作者一样，把记忆压缩成结构化分层认知模型，在需要时快速调用、持续更新、按需升格、反复校验。

这不是写作工具的升级，而是一种面向未来的通用智能体设计框架的新起点。

参考文献

[1] YunFei M. Wang. “论经典 AI 训练方式认知局限的产生原因与破局思路——迈向基于多尺度机器感知与认知对象数据库的下一代人工智能”[PP/OL]. AiraXiv, 2026. eprint: 2611.0009.

[2] YunFei M. Wang. A Philosophical Explanation of Mathematical Effectiveness: The Key-Value-Information-Model Interface Analysis from the Computational Observer Perspective[PP/OL]. PhilPapers, 2026. DOI: 10.5281/zenodo.23655583.

[3] YunFei M. Wang. AI“视觉功能的可行性构建方法——基于多尺度机器感知与三层映射模型的工程实现路线图”[PP/OL]. AiraXiv, 2026. eprint: 2611.0010.

[4] YunFei M. Wang. “AI 听觉功能的可行性构建方法——基于多尺度机器感知与三层映射模型的工程实现路线图”[PP/OL]. AiraXiv, 2026. eprint: 2611.0011

[5] YunFei M. Wang. “AI 嗅觉功能的可行性构建方法——基于多尺度机器感知与三层映射模型的工程实现路线图”[PP/OL]. AiraXiv, 2026. eprint: 2611.0012

[6] YunFei M. Wang. “AI 味觉功能的可行性构建方法——基于多尺度机器感知与三层映射模型的工程实现路线图”[PP/OL]. AiraXiv, 2026. eprint: 2611.0013

[7] YunFei M. Wang. “AI 触觉功能的可行性构建方法——基于多尺度机器感知与三层映射模型的工程实现路线图”[PP/OL]. AiraXiv, 2026. eprint: 2611.0014

[8] YunFei M. Wang. “论 AI 通感（机器联觉）功能的可行性构建方法——基于五感感知体系与认知对象数据库的跨模态联想与验证机制”[PP/OL]. AiraXiv, 2026. eprint: 2611.0015

[9] YunFei M. Wang. “论 AI 认知对象数据库的设计与构建——基于五感感知映射的多模态认知对象数据体系与基础设施”[PP/OL]. AiraXiv, 2026. eprint: 2611.0016

[10] YunFei M. Wang. A Formal Analysis of Causal Laws and Cross-Framework Comparison: Four-Dimensional Decomposition, Coupling Operations, and

Regression-Tendency Discrimination[PP/OL]. PhilPapers, 2026. DOI: 10.5281/zenodo.24051344.
[11] YunFei M. Wang. Causal-Law Differences and the Reconstruction of Physical Entity Forms: Relational-Function Reconstruction, Process Manifestation, and Forms of Conscious Entities[PP/OL]. PhilPapers, 2026. DOI: 10.5281/zenodo.24051438.
[12] YunFei M. Wang. “结构约束先行的视觉生成：AI 绘画与视频生成的三层混合逻辑”[PP/OL]. AiraXiv, 2026. eprint: 2611.0019
[13] YunFei M. Wang. “从三层混合逻辑到生成产线：AI 视觉生成的工程实现与产业迁移”[PP/OL]. AiraXiv, 2026. eprint: 2611.0020
[14] Mage: Memory as Agent-Guided Exploration, 2026.
[15] HiGMem/HiMem: Hierarchical Memory for Long-term Dialogue, 2026.
[16]白雪飞. "约束—干预—生命轨迹：一个理论驱动的研究纲领" (2026). AiraXiv preprint 2610.0020

AIGC 使用声明

本文作者声明：在本文的构思、文献梳理、结构优化与表述润色过程中，作者使用了 DeepSeek AI 助手作为辅助工具。论文的核心思想、理论体系构建、关键论证与最终学术判断均由作者独立完成。AI 辅助过程未涉及任何学术不端行为，特此说明。