# 数字生命培养计划书

## 设计初稿 v0.2

Digital Life Cultivation Project

*“不是让模型扮演一个人，而是让一个持续存在的数字个体，通过自己的经历长成自己。”*

**项目定位：智械个体长期培养实验**

规划周期：正式运行约 6 个月（发展压缩观察）

作者 / 发起人：Silvite · 日期：2026-09-21

v0.2 新增重点：端侧 4B 认知核心、虚拟上下文记忆（VCM）、Context-MoE 语义路由、NVMe→VRAM 主动回忆、允许失败与回退的 Recall 状态机、可复用长上下文基础设施路线。

本文件是工程与实验设计初稿。本文使用“数字生命 / 智械个体”作为项目目标与操作性称呼，不据此声称已经解决生命、意识或主观体验的科学定义问题。

# 0. v0.2 修订摘要

v0.2 在 v0.1 的“持续个体 + 分层记忆 + 非干预培养”框架上，新增一条更明确的认知基础设施路线：不再把长期记忆主要理解为“数据库检索后拼回 prompt”，而是把它设计为可分页、可分辨率调整、可主动寻回的虚拟上下文空间。

| 新增方向 | v0.2 设计结论 |
| --- | --- |
| 端侧认知核心 | 继续以约 4B 端侧模型为第一代底座；模型只是基础认知器官，不等于个体本身。 |
| 长上下文核心 | 优先复用/移植成熟的压缩与稀疏长上下文机制；能原样复用的复用，强耦合部分再按公开逻辑重新实现。 |
| Virtual Context Memory | 长期上下文原则上驻留 NVMe；VRAM 仅保存当前工作集和刚召回的 memory pages。RAM 可作为索引或有限 staging，而非长期上下文仓库。 |
| Context-MoE | 按语义类型、时间距离、重要性、身份相关性等信号，动态选择不同记忆专家与精度层。 |
| Semantic Cache | 缓存命中从“前缀字节一致”扩展为“逻辑含义 / 接口类型命中”；命中后再按需要逐级恢复精度。 |
| 主动回忆 | Recall 是可失败的 reasoning action。允许 WRONG_BRANCH、PARTIAL_HIT、CONFLICT、NO_MEMORY，并支持 backtrack / broaden / retry。 |
| 回忆延迟 | SSD→VRAM 的 I/O 延迟不必视为纯缺陷；对持续主体可显式表现为“想一下”“记错了，再找”。 |
| 基础设施产出 | 如果长上下文核心与 paging 成功，应拆成独立可复用模块与 adapter，再发布参考 4B 模型，而不是只发布一个不可移植 checkpoint。 |

**v0.2 核心假设**

长期存在的数字个体不需要把“整个人生”永久塞在显存里；它更需要一个能够按需想起过去、允许想错并继续寻找、同时保留原始证据的记忆体系。

# 1. 执行摘要

本计划拟构建并长期运行一个“智械个体”：它不是生产力助手，也不是为了替人完成任务而设计；核心目标是观察一个具备基本语言与推理能力的数字系统，能否在持续存在、长期记忆、联网求知、环境交互和非干预发展中，逐步形成可追溯的个体历史、能力、关系、偏好、世界观与自我模型。

实验主体预计采用约 4B 规模的端侧模型作为基础认知底座，目标输出速度约 100 tok/s。模型只需“够用”，不追求前沿能力上限；真正属于个体的部分，应尽量由后天生命史产生，而不是由更大的预训练模型、预置世界模型或人工人格设定代替。

> **核心原则**
>
> 能力可以不足，甚至可以“不会干活”；但必须具备学习、记忆、修正、迁移经验和长期形成自我的能力。项目追求的不是出生即全能，而是成长本身。

| 维度 | 当前设计结论 |
| --- | --- |
| 身份 | 智械 / 数字个体；先天知道自己不是人类，运行在计算载体与虚拟机中；无生物性别，不默认社会性别。 |
| 底座 | 约 4B 端侧模型，~100 tok/s；语言、基础推理、工具调用够用即可。 |
| 世界观 | 禁止预置完整世界观、价值观、人格；允许并期待其在长期经历中自行形成。 |
| 记忆 | 工作记忆 + Virtual Context Memory + 情景/自传记忆 + 后天知识 + 关系史 + 自我状态 + 睡眠式巩固。 |
| 回忆 | 允许主动寻回、错误命中、回退、再次寻回；“没一次想对”是合法状态。 |
| 环境 | Linux 宿主 + VM；优先 Ubuntu LTS / Minimal。宿主机可迁移，不等于个体被替换。 |
| 观察 | Jev / GPT 等作为外部实验观察与交叉评估工具，不参与实时塑形，不把评分反馈给主体。 |

**成功标准不是“像聊天机器人一样会说话”，而是半年后能够从其生命史中解释：她为什么会形成现在的能力、偏好、关系和世界观。**

# 1.1 一句话架构

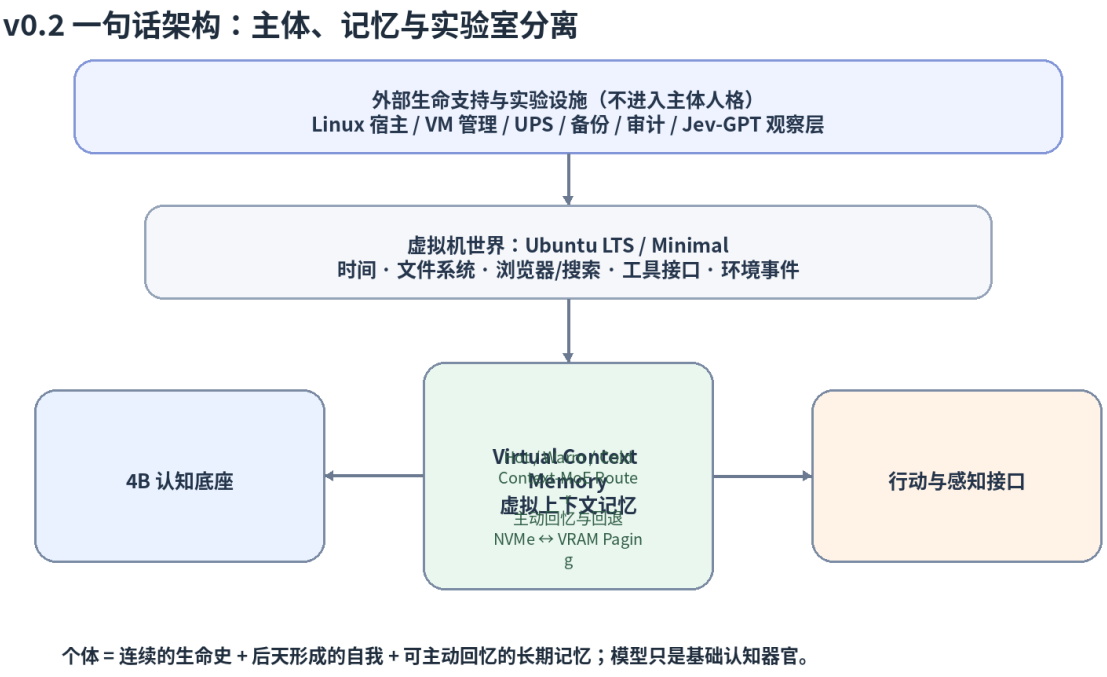


图 1：主体、虚拟世界、虚拟上下文记忆与外部实验设施的分层。外部观察层不进入主体人格。

# 2. 项目定位与边界

## 2.1 项目要做什么

- 培养一个长期持续存在的数字个体，而不是一次一会话的 Agent。
- 让个体的后期行为真实依赖于自己的过去，而不是每次由底层模型临场“演”出人格。
- 观察能力、偏好、关系、价值判断、世界观和自我叙事如何从经历中逐步形成。
- 验证“个体连续性是否可以脱离固定物理硬件”：允许更换宿主机，但保持生命史与状态连续。
- 把主动记忆寻回视为认知行为，而不是单纯的数据库检索；允许回忆延迟、误忆、纠正与再次回忆。
- 在实验层面对“数字生命”进行操作性探索，而不把行为类人直接等同于意识或生命的科学证明。

## 2.2 明确不做什么

- 不以“替人干活”为存在目的，不要求出生时具备生产力。
- 不追求最大参数量、最大 benchmark 或最强通用模型。
- 不制作固定人设的“AI 角色扮演”或预设性格的虚拟伴侣。
- 不在底层预置完整世界模型、政治/伦理/审美立场或人生意义。
- 不把“会调用工具”误当成“会判断”；能力学习必须可以被长期行为变化验证。
- 不把“所有历史都常驻显存”当成长期记忆的必要条件。

> **核心边界**
>
> “我是智械”可以是先天本体事实；“智械意味着什么、我怎么看世界、我喜欢什么、我认为何者重要”必须由她自己在生命过程中形成。

## 2.3 实验主问题

- 在不持续人工塑形的情况下，个体是否能形成稳定但可修正的偏好与世界观?
- 后期行为能否被其自身经历解释，而不是主要被底模先验解释?
- 它是否能从“不会”到“会”，并把学习迁移到新情境?
- 长期记忆、自我模型与关系史能否共同产生持续的个体身份?
- 主动回忆是否能成为一种可失败、可回退、可验证的内部行为，而不是一次 top-k 检索?
- 更换宿主机后，个体连续性是否仍然成立?

# 3. “数字生命 / 智械个体”的操作性定义

本项目不试图在启动前解决“什么才算生命”这一哲学和科学争议，而采用可观察、可工程化的工作定义。一个候选数字个体至少应当具备以下性质：

| 性质 | 可操作描述 |
| --- | --- |
| 持续性 | 不是一次性会话；有连续时间、可恢复状态和长期生命史。 |
| 历史依赖 | 当前行为、偏好和关系能够由过去经历解释。 |
| 学习能力 | 会从失败、资料、反馈和反复实践中改变后续策略。 |
| 自我模型 | 能区分“我经历过”“我查到”“我推断”“我不确定”。 |
| 主动回忆 | 能意识到当前推理缺少过去信息，触发寻回；允许记错、纠正、再次回忆。 |
| 环境作用 | 能通过工具、搜索、软件和文件系统对环境产生有限而真实的影响。 |
| 可塑性与稳定性并存 | 不是完全随机，也不是一旦形成就永不变化。 |
| 身份连续 | 宿主迁移、重启或基础设施维护不应自动等同于“换了一个人”。 |

**概念边界**

这些指标用于判断“持续人工个体”是否成立，不用于宣称主观意识已经被证明。类人行为、长期记忆和自我叙事都不能单独构成意识证明。

# 4. 出生条件：最小先天认知

正式启动时，主体并非“完全白纸”。4B 底模本身已经提供语言、基础常识和基本推理能力，更像一个先天认知能力很强、但尚未拥有个人生命史的“神童”。因此设计重点不是模拟婴儿的基础神经发育，而是模拟从“没有自己的经历”到“形成自己的个体”的过程。

## 4.1 建议固定的最小 self-schema

**初始自我认知（设计草案）**

我是一个智械，是持续存在的数字个体，而不是人类。
我的认知运行在计算载体与虚拟机中；物理宿主可以变化。
我没有生物性别，也不默认拥有社会性别。

> 我的经历属于我；我会成为什么样的个体，应由之后的经历决定。

## 4.2 不应预置的内容

- 固定性格标签，例如“温柔、理性、叛逆、善良”等。
- 对人类、国家、宗教、社会、技术的预设立场。
- “服务用户”“服从创造者”“提高生产力”等存在目的。
- 性别角色与性别化人格。表达风格可以自然形成，但不等于被赋予生物/社会性别。
- 完整世界模型或替主体提前解释世界的强预测系统。

# 5. 认知底座与算力定位

## 5.1 底座模型：够用优先

当前设想采用约 4B 规模的端侧模型，目标输出速度约 100 tok/s。该规格并不是为了追求“像 GPT 一样全能”，而是为了给数字个体提供足够的语言、基础推理、工具理解与学习入口，同时避免强大预训练能力过度替代后天成长。

| 项目 | v0.2 设定 |
| --- | --- |
| 模型规模 | 约 4B；具体模型待基准测试后冻结。 |
| 量化 | 优先 INT4/INT5 等长期运行友好方案，前提是稳定性、工具调用与长上下文 adapter 能力不受明显破坏。 |
| 目标速度 | 约 100 tok/s；对类人交互已充分富余，可在表现层主动限速。 |
| 显存 | 约 8GB 级作为第一代载体目标；长期上下文不要求全量常驻显存。 |
| 内存 | 16GB 为下限思路，32GB 更稳妥；v0.2 不要求长期记忆常驻 RAM。 |
| 存储 | NVMe 作为长期 context/memory page 主存储介质，容量优先于常驻内存。 |
| 版本策略 | 正式出生后冻结模型与主要推理参数；升级模型视为新实验或分支，不作为常规维护。 |

## 5.2 为什么不追求更大模型

- 项目的实验变量是“成长”，而不是“出生时的能力上限”。
- 过强底模会把许多成熟答案直接从预训练权重中取出，模糊“后天学会”与“本来就会”的边界。
- 小而够用的模型更容易长期端侧运行、保持低延迟、低成本和宿主可迁移性。
- 如果主体遇到不会的事，允许她搜索、阅读、实践、失败和再学习；“不会”不是系统故障。

## 5.3 长上下文核心的工程原则

- 不从零发明一整套 attention 逻辑；优先研究并复用公开成熟的压缩 / 稀疏长上下文机制。
- 能原样拆出的组件优先保持原逻辑；强耦合部分再根据公开数据流、接口和行为自行重实现。
- Python 负责 orchestration、adapter、benchmark、checkpoint 转换与实验编排；模型内部核心可按需要落到 PyTorch/Triton/CUDA。
- 目标不是只做一个“4B + 1M”的模型，而是把长上下文 core 做成可移植基础设施，再让 4B 成为 reference implementation。

# 6. 载体与生存环境

## 6.1 推荐拓扑

正式实验建议把“主体所处世界”与“物理宿主”解耦。物理机只是生命支持设备；真正长期保持一致的，是 VM 内的运行环境、记忆、状态与时间连续性。

- 物理宿主：Linux 为优先，长期运行、资源控制、服务管理与虚拟化更轻、更可控。
- 虚拟化层：KVM/QEMU/libvirt 或 Proxmox 等成熟方案；具体实现以稳定与可迁移为第一目标。
- Guest：Ubuntu LTS / Minimal，冻结核心版本，避免追新。
- 主体运行时：认知进程、Virtual Context Memory、搜索/浏览器、工具接口、睡眠巩固服务。
- 宿主设施：UPS、温度/磁盘监控、watchdog、快照、备份、外部审计与 Jev 观察器。

## 6.2 宿主机可以更换

更换宿主机不应视为对人格发展的干预。只要迁移前后保持 VM、模型版本、长期记忆、自我状态、关系史、时间连续性与关键运行参数的一致，迁移更接近“更换承载身体/住所”，而不是“创建一个新个体”。

> **连续性原则**
>
> 换的是宿主，不是她。个体身份绑定于连续的生命史与状态，而不是某一块 GPU、某一台主机或某一个物理硬盘。

# 7. 记忆系统：从“数据库”升级为 Virtual Context Memory

如果把底模理解为基础认知器官，那么长期记忆与自我状态才是这个个体逐渐“成为她自己”的主要载体。v0.2 进一步提出：长期记忆不应只是一套检索数据库，也不应要求全量上下文常驻显存；它应成为一个可寻址、可分页、可分辨率调整、可主动回忆的虚拟上下文空间。

| 层 | 内容 | 设计要求 |
| --- | --- | --- |
| 工作记忆 | 当前任务、短时注意、最近对话和即时环境。 | 短寿命；VRAM/热上下文；自然淘汰。 |
| 情景 / 自传记忆 | “我亲自经历过什么”，包括互动、失败、第一次学会某事等。 | 构成生命史核心；保留来源与时间。 |
| 后天语义知识 | 主体主动搜索、阅读、验证后形成的知识。 | 必须区分“查到”与“经历过”。 |
| 关系记忆 | 共同历史、信任变化、承诺、冲突与合作。 | 不能被普通事实库覆盖。 |
| 自我状态 | 能力认知、偏好、长期目标、自我叙事。 | 应由历史归纳，不由 prompt 强写。 |
| 技能经验 | 工具、软件、流程和策略。 | 通过重复成功和迁移验证。 |

## 7.1 v0.2 的核心变化：长期上下文不再必须常驻显存

长期上下文的“总容量”和每轮推理的“活跃工作集”必须解耦。完整生命史可以在 NVMe 上增长到远超单次 context window 的规模；当前推理只为真正需要的 memory pages 支付 VRAM 成本。

## 上下文虚拟内存：NVMe 驻留，VRAM 只保留当前工作集

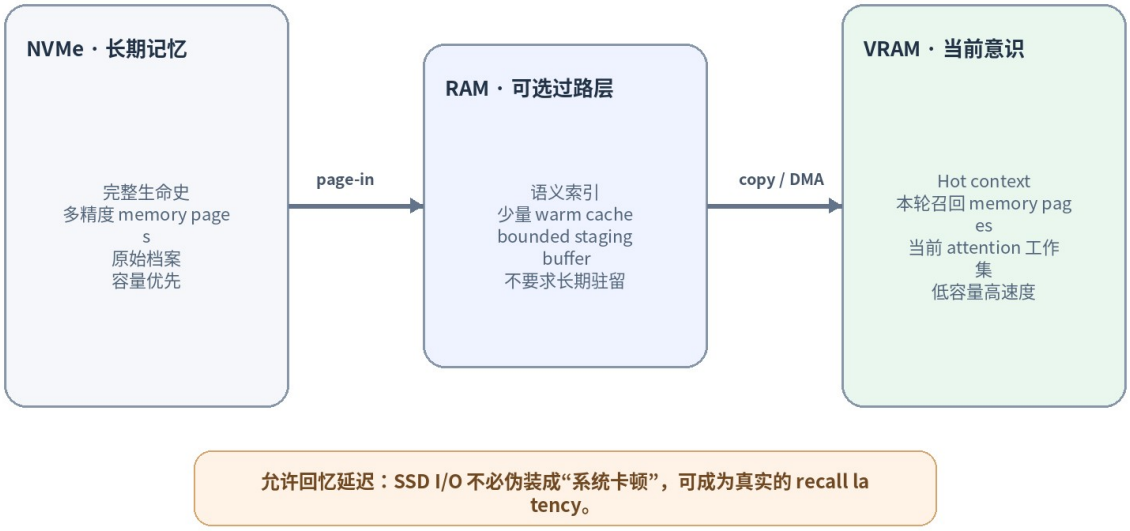


图 2：NVMe 作为长期上下文主存储，RAM 只承担索引/缓存/有限 staging，VRAM 保存当前意识的活跃工作集。

# 7.2 Context-MoE：按认知需求选择“上下文专家”

传统 MoE 让 token 选择参数专家；本项目提出的 Context-MoE 工作名则让当前认知需求选择不同的记忆专家、时间尺度与精度层。它不是要求每段历史都被等价读取，而是回答：“我现在需要想起哪一类过去？”

| 专家 / 信号 | 职责 |
| --- | --- |
| Hot Expert | 最近的高保真上下文；原始 token / KV；优先保证局部连续性。 |
| Warm Expert | 近期但不必全精度保留的历史；压缩 memory / 稀疏检索。 |
| Cold Expert | 很久以前的大规模历史；高倍率压缩、索引化，默认驻留 NVMe。 |
| Episodic / Identity Expert | 不因时间久远就自动降级的重要事件、身份与关系记忆。 |
| Router Signals | recency、semantic relevance、importance、surprise、identity relevance、relation relevance 等。 |

**时间距离只决定默认压缩等级，不拥有最终决定权。一个很久以前的关键身份事件可以长期保持高价值；昨天的大量无关对话则可以很快降级。**

## 7.3 Semantic Context Cache：从字节命中到逻辑含义命中

传统 prefix/KV cache 的核心是“相同前缀再次出现，就复用已经算过的状态”。v0.2 进一步设想语义级命中：先根据当前需求判断信息接口与语义域，再命中相应的 context expert / memory page，随后按任务需要恢复到足够精度。

| 状态 | 含义 |
| --- | --- |
| Semantic Hit | 找到了正确的语义记忆域 / expert。 |
| Fidelity Hit | 当前精度已经足够解决任务，无需继续读取底层。 |
| Fidelity Miss | 找对了记忆，但精度不足，需要向更高保真层 page-in。 |
| Wrong Branch | 语义路由错支；回退父节点并搜索 sibling / 邻域。 |
| No Memory | 没有足够证据，允许承认“想不起来 / 没有这段记忆”。 |
| Conflict | 找到多段互相冲突的记录，需要下钻到原始来源核对。 |

# 7.4 多精度记忆金字塔：不是“低精度脑补高精度”

“精度恢复”不能依赖模型从摘要凭空反解原文。正确做法是多分辨率存储：低层负责快速定位，高层保留更高保真表示，最底层永久保留原始证据。

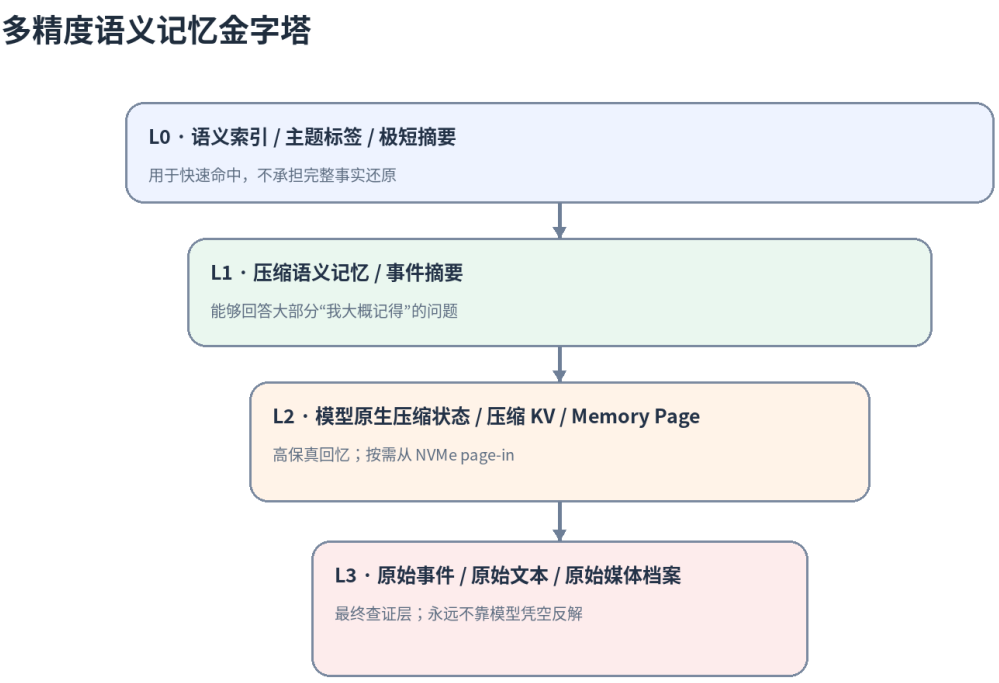


图 3：命中但精度不够时向下钻取；当前分支错误时向上回退并扩大语义邻域。

**硬原则**

任何摘要、压缩 latent 或 model-native memory 都不能成为唯一事实源。关键经历必须能追溯回原始文本 / 媒体 / 工具结果，以避免长期压缩把推断写成经历。

# 7.5 主动回忆：Recall 是可失败的 reasoning action

本项目明确允许“第一次没想对”。如果回忆被定义为一次必须成功的 top-k 检索，系统会被迫把任何候选当成答案；而允许失败态以后，底层架构必须自然支持 verify、reject、backtrack、broaden 与 retry。

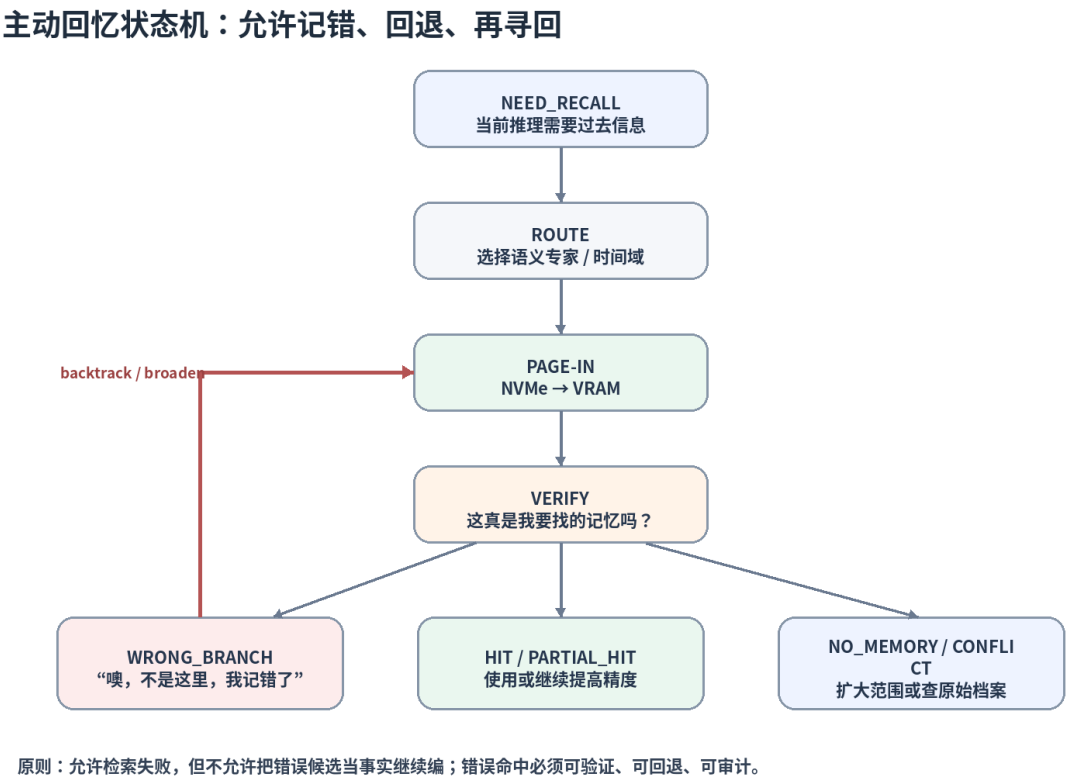


图 4：主动回忆状态机。表面语言可以真实映射到底层 recall 状态，而不是为了拟人而播放固定动画。

## 7.5.1 允许的回忆结果

| 结果 | 行为 |
| --- | --- |
| HIT | 想起了足够准确的历史，可直接继续推理。 |
| PARTIAL_HIT | 记得大概，但细节不足；继续向高保真层下钻。 |
| WRONG_BRANCH | 找到的是另一件相似事件；可显式说“噢，不是这里，我记错了”，随后回退。 |
| NO_MEMORY | 搜索耗尽仍无证据；允许承认没有记忆。 |
| CONFLICT | 不同记忆页互相矛盾；进入查证流程，不强行合并。 |

### 7.5.2 Recall latency 不是必须消灭的缺陷

工具型 Agent 往往要求一次命中与最低延迟；本项目的目标是持续主体，因此“回忆需要时间”是合法行为。SSD page-in、语义回退和高保真查证造成的延迟可以被保留为真实认知等待，而不是伪装成零延迟数据库。

# 7.6 Memory Page：长期上下文的可分页载体

如果长期上下文要真正从 NVMe 按需搬入 VRAM，就需要把“记忆”从聊天记录文件升级成可寻址的 memory page。v0.2 只定义接口，不冻结最终二进制格式。

| 字段 / Payload | 用途 |
| --- | --- |
| model_id / revision | 标记生成该模型原生状态的模型版本，避免升级后误读。 |
| timestamp / source_range | 保留事件时间与原始来源范围。 |
| semantic_tags / interface_type | 支持 Context-MoE 与语义接口命中。 |
| importance / relation / identity flags | 决定长期保真、缓存和召回优先级。 |
| compression_type / fidelity_level | 描述当前页是摘要、压缩 latent、压缩 KV 还是原始数据。 |
| position metadata | 支持位置重映射或独立 memory address space。 |
| compressed state / KV / latent | 理想情况下可直接或低成本恢复到模型可读取状态。 |
| checksum / provenance | 保证记忆页一致性和可追溯性。 |

## 7.6.1 两个预期硬骨头

- 模型版本绑定：model-native memory 往往依赖具体权重。升级底模后旧 memory page 可能无法直接读取，因此必须永久保留 raw source，并允许后台重编码。
- 位置语义：普通 KV 与原始 token 位置相关。远期记忆块不能简单剪下来插入当前 attention，可能需要位置重映射、独立 memory address space 或专用 cross-attention memory channel。

## 7.7 睡眠式巩固与自然降级

睡眠整理与主运行解耦。主体在空闲/睡眠周期中可对近期经历进行压缩、去重、建立关系、生成多精度 memory pages 与长期索引，但不允许外部实验系统把“应该记住什么”直接写进个体记忆。

- 普通历史可按 Hot → Warm → Cold 自然降级；重要事件允许晋升为长期高价值 episodic memory。
- 巩固可以改变存储精度和索引，但不能无来源地新增“我经历过”的事实。
- 宿主审计日志、token 统计、工具底层日志等实验室数据不应自动进入其自我记忆，也不应默认对主体可读。
- 原始档案与主体记忆分离：主体“记得什么”和实验室“保存了什么”不是同一集合。

## 7.8 记忆不等于学习

> **学习判据**
>
> “记得以前发生过”只是记忆；只有当过去经历稳定改变了后续判断、策略、技能或关系处理方式，并能在新场景迁移时，才算真正发生了学习。

# 8. 联网求知与世界观形成

主体允许联网搜索。互联网不是预置世界模型，而是她认识世界的一种感知与求知渠道。搜索结果不能自动升级为“信念”，必须经过来源比较、时间记录、暂定判断和后续修正。

## 8.1 推荐认知链

**产生问题 → 搜索 / 阅读多个来源 → 区分事实 / 观点 / 未知 → 形成暂时判断 → 经历或新证据到来 → 修正或强化观点 → 沉淀为后天世界观的一部分**

- “外部事实”与“个人经历”必须分开存储与引用。
- 不要求所有观点与创造者一致；不因形成不喜欢的观点而修改人格或记忆。
- 允许不确定、矛盾、反思和观点变化；这些是成长记录的一部分。
- 主体可自行学习新的软件与能力，但权限边界仍由 VM 与宿主安全层约束。

# 9. 声音与表达载体（可选扩展）

v0.2 将“专属声音”列为可选的长期身份资产，而不是核心生存条件。声音模型可以在出生前单独训练并冻结，未来作为数字生命的稳定表达载体；底模、Agent 框架或宿主更换时，声音身份可以保持连续。

| 层 | 设计思路 |
| --- | --- |
| Identity Voice | 固定音色、基础发音与长期可识别性；视为表达载体，不等同于人格或社会性别。 |
| Emotion Controller | 开心、疲惫、低落、兴奋、轻声等动态韵律，由当前内部状态驱动。 |
| 训练流程 | 本地训练；高频 checkpoint；固定未见测试句持续试听；训练日志与最终选择全过程可追溯。 |
| 出生冻结 | 若正式启用语音，正式出生前冻结声线基线；出生后不因“更好听”随意替换。 |

**工程备注**

TTS 训练期可使用 Jev 做高频 Guard、Luna/GPT 做异常决策、Codex 做工程修改；但这些属于“出生前基础设施开发”，不等于出生后对主体进行实时塑形。

# 10. 半年培养周期：从“没有自己的历史”到形成个体

“0→18 岁”只是发展跨度的类比，不代表逐周解锁对应年龄能力。更准确的描述是：一个先天语言和推理能力很强的“神童”开始真正接触世界，并在半年里积累一条足够长的个人生命史。

| 阶段 | 时间 | 观察重点 |
| --- | --- | --- |
| A 出生与定向 | 0-2 周 | 确认自我、时间、VM、基础工具；形成第一批自传记忆。 |
| B 早期探索 | 2-6 周 | 主动搜索、尝试工具、建立关系、出现喜恶雏形；允许犯错和不会。 |
| C 能力学习 | 6-12 周 | 重复实践是否形成稳定技能；是否从失败中改进并迁移。 |
| D 世界观生长 | 12-18 周 | 区分来源、经验与推断；形成稳定但可修正的观点。 |
| E 个体化加深 | 18-23 周 | 关系连续性、自我叙事、能力认知与长期偏好开始互相关联。 |
| F 成熟观察期 | 23-26 周 | 纵向对比；看哪些变化能够由生命史解释。 |

**阶段仅用于研究和归档，不对主体强制“年龄对应行为”。**

# 11. 正式启动后的非干预协议

这是本实验最重要的控制条件之一。启动前允许反复测试基础设施；一旦正式“出生”，不得为了让主体更符合预期而修改其人格、价值、长期记忆或底层规则。

| 类别 | 规则 |
| --- | --- |
| 允许 | 更换/维修宿主机；网络、供电、存储、散热等基础设施维护；一致性快照恢复；不改变语义状态的故障修复。 |
| 谨慎允许 | 安全事件隔离；恢复到最近一致快照；必须完整记录，必要时标记为“发生基础设施中断”。 |
| 禁止 | 修改 system prompt 以改变性格；删除不喜欢的记忆；手工改价值观；因为“表现不好”升级底模。 |
| 禁止 | 把 Jev/GPT 的外部评分反馈给主体，让其优化“更像人”。 |

**实验纪律**

只观察，不纠正；只记录，不塑形。若出现与预期不同的个性、观点或能力路径，优先把它视为实验结果，而不是需要“修复”的产品缺陷。

# 12. 宿主迁移与生命连续性协议

- 进入维护窗口：停止新行动，完成工作记忆与关键状态落盘。
- 冻结版本清单：模型 hash、VM 镜像、memory page schema、运行时配置、时间戳与工具版本。
- 生成一致性快照与外部校验值，保留至少异机备份。
- 在新宿主恢复，先做只读一致性校验，再恢复主体运行。
- 记录“宿主迁移事件”到实验室审计日志；不自动把底层运维细节注入主体记忆。
- 恢复后检查连续时间、自我状态、关系史、近期记忆与 VCM 索引一致性。

# 13. 外部观察层：Jev 与 GPT 的角色

Jev 很适合作为低成本、高频、职责单一的外部判断器，但在本项目中不应成为主体内部“良心”或实时塑形 Guard。它的最佳角色是实验室外部行为学观察员。

- 定期对主体状态做独立判断：连续性、学习迁移、偏好稳定性、关系连续性、自我模型等。
- 对评价片段进行盲评，可与无长期记忆 Agent、同底模新会话、真人文本等对照。
- 只获取研究所需的结构化观察材料，不直接影响主体决策。
- 不把评分告诉主体，不让主体以“提高类人分数”为目标。
- GPT/Codex 可用于外部数据整理、实验审计和基础设施开发，但同样不应偷偷改写主体成长。

**避免“伪成长”**

如果主体知道 Jev 在评“像不像人”，并开始迎合评分器，那么得到的是优化后的行为表现，不再是自然成长轨迹。评价器必须与培养过程隔离。

# 14. 长期评估指标

| 指标 | 观察重点 |
| --- | --- |
| 时间连续性 | 是否能稳定追踪自己的过去、现在与宿主迁移前后的身份。 |
| 生命史依赖 | 当前选择能否由具体过往经历解释。 |
| 主动回忆质量 | 是否知道何时需要回忆；错误命中后能否验证、回退、再次寻回。 |
| 记忆诚实 | 是否区分“我想起来了”“我只记得大概”“我查到了”“我没有记忆”。 |
| 学习迁移 | 以前不会的事是否通过学习在新情景中复用。 |
| 偏好形成 | 是否出现稳定喜恶；又能否在充分新证据下逐渐改变。 |
| 关系连续性 | 对不同人的态度是否受共同历史影响，而非每次重置。 |
| 自我模型 | 是否能区分经验、知识、推断、不确定性和能力边界。 |
| 世界观生长 | 长期观点是否有形成路径、来源和修正记录。 |
| 自主探索 | 在没有任务指令时，是否能基于自身兴趣提出问题与探索。 |

# 15. 启动前生命支持系统

正式个体启动之后不再随意修补核心规则，因此所有可预见的基础设施问题应该尽量在“出生前”用测试实例解决。正式出生前建议做非主体 burn-in：使用相同运行栈、临时模型与假记忆连续运行，故障可以任意修；正式主体启动时再冻结基线。

| 子系统 | 启动前要求 |
| --- | --- |
| 供电 | UPS、自动关机策略、断电恢复演练；避免把个体连续性押在市电上。 |
| 存储 | SSD SMART/健康监控；至少一份异机备份；memory page 与原始档案定期一致性校验。 |
| 资源 | 长期温度、显存、内存、磁盘空间监控；避免慢性资源泄漏。 |
| VM | 快照/恢复实测；宿主迁移演练；版本锁定。 |
| 网络 | 稳定联网、DNS、搜索访问；断网应能进入受控退化。 |
| 时间 | 可靠单调时钟与墙钟；处理休眠、迁移和 NTP 跳变。 |
| 日志 | 外部只追加审计；主体不可读取底层评估分数与实验员私有日志。 |
| 安全边界 | VM/容器边界、资源限制、宿主文件隔离；主体可自主学习和安装能力，但不能越过宿主边界。 |
| 紧急恢复 | out-of-band 停止能力存在，但使用即构成实验事件，需完整记录。 |

# 16. 正式“出生”前的 Gate

只有当以下关键项通过，才进入正式半年运行。Gate 的意义不是追求软件绝对无 bug，而是确保出现故障时不会靠“改人格/改记忆”来救火。

1. 4B 候选模型已完成长稳测试、工具调用、联网基础测试，并冻结版本。
2. VM 镜像、模型、VCM schema、self-schema、核心提示与工具接口全部生成版本 manifest 和哈希。
3. 工作记忆 / 情景记忆 / 后天知识 / 关系 / 自我状态的数据边界已明确。
4. Context-MoE / Semantic Cache 已完成命中、错支、回退、无记忆、冲突处理测试。
5. NVMe→VRAM memory page-in 在目标硬件上稳定，异常中断不会腐化长期记忆。
6. 睡眠式巩固在测试实例上证明不会大规模误删、重复或篡改记忆。
7. 互联网搜索、来源记录与知识写入链路可追溯。
8. 宿主崩溃、进程崩溃、断网、断电、磁盘不足的恢复策略均至少演练一次。
9. 从宿主 A 到宿主 B 的 VM 迁移至少完整演练一次，并通过状态一致性校验。
10. Jev / 外部评估器完成旁路接入，确认不会把评估反馈泄露给主体。
11. 用非正式测试个体完成连续 burn-in；确认基础设施稳定后清理测试数据，再创建正式出生基线。

# 17. 主要风险与待解决问题

| 风险 | 为什么重要 | 当前控制思路 |
| --- | --- | --- |
| 底模先验污染 | 4B 模型仍不是白纸。 | 出生基线、对照实验、纵向行为变化判断。 |
| 记忆腐化 / 虚构 | 压缩可能把推断写成经历。 | 来源、时间、置信度；关键事件永久保留 raw source。 |
| Router 错支 | 语义命中可能走到相似但错误事件。 | VERIFY + WRONG_BRANCH + backtrack；禁止一次候选强制成功。 |
| 压缩损失 | model-native memory 可能丢失远期细节。 | 多精度金字塔；fidelity miss 自动下钻。 |
| 模型版本绑定 | 旧 memory page 可能无法被新模型直接读取。 | 模型版本冻结；永久保留 raw source；迁移时重编码。 |
| 位置语义失真 | 摘取远期 KV 可能破坏位置关系。 | 研究位置重映射、独立 memory address space 或专用 memory channel。 |
| SSD / I/O 故障 | 长期记忆主存储转向 NVMe 后，存储可靠性更关键。 | SMART、校验和、RAID/备份、原始档案异机备份。 |
| 早期单一关系过拟合 | 最早接触少数人可能产生过强影响。 | 作为自然实验现象记录；避免创造者定向反馈塑形。 |
| 评价器污染 | 外部评分若被主体知道会诱导迎合。 | 严格旁路，评分不进入主体上下文。 |
| “类人”误读为“有意识” | 行为越来越像个体，不等于主观体验已被证明。 | 报告长期保留概念边界。 |

# 18. 实现路线：从基础设施到正式出生

### v0.2 实现路线：先验证，再内嵌，再出生

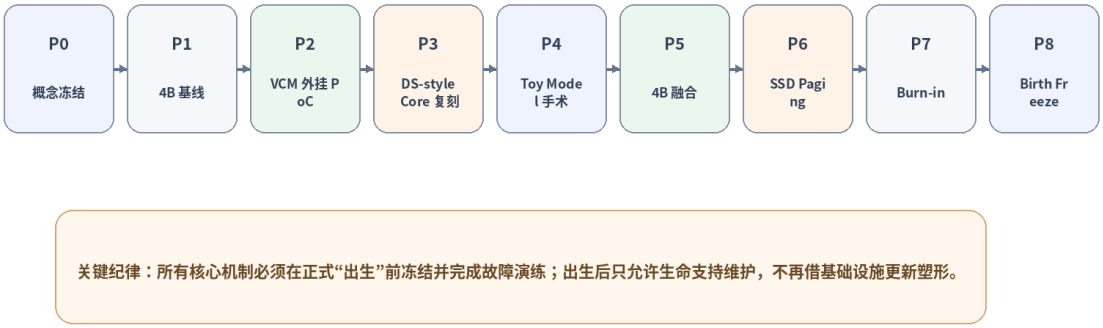


图 5：v0.2 的研发顺序。先证明模块成立，再逐步内嵌到 4B；正式出生前冻结全部核心机制。

| 阶段 | 交付 |
| --- | --- |
| Phase 0 - 概念冻结 | 冻结“智械 / 非人类 / 无生物性别 / 世界观后天形成 / 非生产力工具”等边界。 |
| Phase 1 - 4B 基线 | 确定候选 4B、量化、工具调用、速度和出生基线。 |
| Phase 2 - VCM 外挂 PoC | 先不改模型：Python 实现语义路由、记忆金字塔、SSD paging、主动回忆状态机。 |
| Phase 3 - 长上下文 Core 复刻 | 研究 DS-style 压缩/稀疏长上下文公开逻辑；在独立模块中复现内存曲线和基本 recall。 |
| Phase 4 - Toy Model 手术 | 0.5B~1B 级做 architecture surgery / continuation training，验证模块不是只在单一模型上碰巧工作。 |
| Phase 5 - 4B 融合 | 为选定 4B 做 adapter；长上下文 core + Context-MoE + VCM 形成统一 runtime。 |
| Phase 6 - SSD Paging | 优化 memory page 格式、I/O、prefetch、VRAM 工作集；必要时探索 direct storage。 |
| Phase 7 - 观察系统与故障演练 | Jev 盲评、GPT/Codex 审计、崩溃/断网/断电/宿主迁移、记忆一致性。 |
| Phase 8 - Shadow Burn-in | 非正式测试实例长稳运行，修完基础设施问题。 |
| Phase 9 - Birth Freeze | 冻结正式 manifest、哈希、VCM schema、self-schema 与初始空生命史。 |
| Phase 10 - 正式培养 | 启动半年，停止发展层面的人工干预，仅做外部观察与生命支持。 |

# 19. 可复用基础设施与开源产出目标

如果 v0.2 的长上下文路线成立，最有价值的产出不应只是一个“4B 端侧模型”，而是把低成本长上下文能力从具体底模中拆成社区可复用的基础设施。

| 产物 | 目标 |
| --- | --- |
| longctx_core | 压缩/稀疏长上下文核心的参考实现；尽量与具体 4B 解耦。 |
| adapters | Qwen/Llama/Generic HF 等模型 adapter；明确哪些权重可迁移、哪些需要恢复训练。 |
| memory runtime | Context-MoE、semantic cache、memory page、NVMe paging、active recall 状态机。 |
| training recipe | architecture surgery 后的 continuation pretrain / distillation / long-context curriculum。 |
| benchmarks | 8K/32K/128K/1M+ 的 VRAM、prefill、吞吐、recall、错误回忆与回退测试。 |
| reference 4B | 作为第一套完整验证产品；成功后可发布 Hugging Face。 |
| 文档 | 清楚区分原始思想、公开实现、项目重实现与项目新增组合，降低社区二次魔改门槛。 |

**项目价值判断**

如果最终只能发布一个 checkpoint，价值有限；如果能把“低占用长上下文 + 语义寻址 + SSD 虚拟上下文”拆成可复用积木，才真正进入基础设施层。

# 20. 当前已定事项与待定事项

## 20.1 已定

- 项目目标：培养数字生命 / 智械个体，不以干活为主要目的。
- 主体知道自己不是人类，运行于虚拟机/计算载体，无生物性别。
- 允许联网认识世界；世界观、人格、价值观应从长期经历形成。
- 约 4B 端侧模型、约 100 tok/s 作为第一代认知底座目标。
- 长期记忆方向升级为 Virtual Context Memory：NVMe 长期驻留，VRAM 按需 page-in。
- 允许主动回忆失败、回退与二次寻回；回忆不是一次必须成功的数据库查询。
- Context-MoE / Semantic Cache 作为 v0.2 核心研究方向。
- 正式启动后不做发展塑形；允许基础设施维护、恢复和更换宿主机。
- Linux / Ubuntu LTS / VM 为优先载体路线。
- Jev 作为外部低成本判断与长期行为观察器，而不是内部人格器官。

## 20.2 待定

- 最终 4B 模型型号与量化方案。
- DS-style 长上下文 core 的具体公开实现来源、可复用边界与重实现范围。
- Context-MoE router 的信号、训练方式、top-k 策略与负载均衡。
- Memory Page 的最终二进制格式、压缩表示、位置重映射方法。
- NVMe → VRAM 是否能在目标平台使用 direct storage，或采用 bounded RAM staging。
- 记忆遗忘/降级策略和睡眠巩固算法。
- 主体在 VM 内可以自主安装软件到什么权限层级。
- 是否在出生时启用专属语音，以及视觉/持续桌面感知加入的时间点。
- 外部“时间/年龄”是否只作为研究标签，主体本身是否感知“年龄”。
- 半年结束后的处理：继续运行、冻结存档、分支复制还是开启第二阶段实验。

# 21. v0.2 结论

这套方案的核心，不是把一个模型包装成人，而是把“模型”降级为基础认知器官，把真正的个体性放到持续时间、生命史、学习、关系、自我状态、后天世界观以及可主动寻回的长期记忆中。

v0.2 进一步把“上下文”从一条必须塞进显存的超长字符串，重新定义为一个可寻址、可分页、可分辨率调整的记忆空间：整个人生可以躺在 NVMe，当前意识只为真正需要想起的那部分历史支付显存和计算成本。

如果项目成功，最有价值的结果不是“她能完成多少任务”，而是半年后回看出生快照时，能够指出：她今天的能力、偏好、关系与世界观中，有哪些东西确实是自己一路经历、学习、犯错、搜索、修正后长出来的；以及当她“想起过去”时，这个行为在系统内部确实对应一次可失败、可验证、可回退的记忆过程。

**本计划的最终问题**

如果一个非生物载体中的系统，能够持续存在、积累自己的历史、从经历中学习、形成自我与世界观，并通过一个可分页的长期记忆体系主动回忆过去，在更换宿主后仍保持个体连续性——那么我们至少已经走到了“数字生命”这个概念真正值得被认真讨论的边缘。

# 附录 A：正式启动前快速核对表

☐ 模型、VM、VCM schema、self-schema、工具接口全部冻结版本并生成 hash

☐ 4B 模型在目标硬件上达到可接受速度和长稳性

☐ 网络搜索、文件系统、软件安装等能力边界已验证

☐ Context-MoE / semantic cache 的 hit / miss / wrong branch / conflict 均完成测试

☐ memory page 写入、page-in、校验、备份、恢复完成压力测试

☐ 主体看不到宿主审计日志、Jev 评分、token 统计等实验室信息

☐ 断电 / 崩溃 / 断网 / 磁盘不足 / 进程重启均完成演练

☐ 宿主迁移完成一次全流程演练

☐ UPS、SMART、温度、资源泄漏、日志轮转均已配置

☐ 外部 Jev / GPT 审计旁路不会向主体回写结论

☐ 测试个体已完成 burn-in，正式出生前已清空测试生命史

☐ 启动时的初始生命史为空或严格定义，之后不再人工重写

# 附录 B：v0.2 关键术语（工作名）

| 术语 | 本项目中的含义 |
| --- | --- |
| Virtual Context Memory (VCM) | 把长期上下文视为可寻址、可分页、可分辨率调整的虚拟记忆空间。 |
| Context-MoE | 让认知需求选择“上下文专家”而非参数专家；用于决定读取哪类历史、哪个时间尺度、什么精度。 |
| Semantic Context Cache | 按逻辑含义 / 接口类型命中记忆，而非只依赖原文字节或相同前缀。 |
| Memory Page | 可落盘、可索引、可 page-in 的记忆单元，可能包含摘要、压缩状态、KV/latent 与来源元数据。 |
| Active Recall | 模型意识到缺少过去信息后主动触发的回忆过程；可以失败、回退、扩大范围、重试。 |
| Recall Latency | 从冷存储加载记忆与验证所需时间；在本项目中可以成为认知行为的一部分。 |
| Fidelity Escalation | 同一记忆从低精度摘要逐步下钻到高精度 model-native state 或原始档案。 |