**人工智能直觉架构 (AII)**  
**(情绪驱动激活扩散认知架构 - EDASCA)**

**文档摘要**

本文档详细描述了**人工智能直觉架构（Artificial Intuition Intelligence, AII）**的完整设计方案。该架构旨在模拟人类思维中深刻的联想、连续的学习、直觉驱动的预测与目标导向的决策过程。其核心是一种基于图网络的激活扩散模型，以**CAn-PAD整合情绪架构**（认知评估与核心情绪的双层模型）为核心驱动，通过多模态词元的动态融合与**事件层级**的认知构建，实现连续的内在想法流、与外在环境的复杂交互，以及基于预测验证的自演化学习能力。

架构的核心创新在于将认知与感受分离，并在认知流中引入了**动态事件融合器 (Dynamic Event Fusioner)** 与 **事件预测器 (Event Predictor)**。系统不再仅仅处理孤立的概念，而是动态地将激活的认知节点和感受节点融合成临时的“**事件**”作为处理、预测和注意的基本单位。这种“事件”是**涌现 (emergent)** 的，而非静态存储的，它完全依赖于节点间的链接强度和历史预测验证记录。此外，架构通过独立的感受记忆库、多通道感受器系统（时间/情绪/外在），为智能体提供了真正的情景记忆、情感体验和直觉式认知能力。

系统支持两种运行模式：**LLM辅助训练模式**（用于快速知识蒸馏与初始化）和**异步纯本地独立模式**（实现真正的主观能动性与独立智能）。同时，架构引入了**参数自适应微调系统**与**认知过滤器**，确保其长期发展的适应性、稳定性与安全性，使其能够在不同算力环境下自主调整认知策略，并符合预设的伦理规范。

**作者**: 银子  
**日期**: 2025年9月23日

**引言**

传统的人工智能范式，无论是以逻辑规则为基础的符号主义，还是以数据拟合为核心的连接主义，在模拟人类心智的整体性、自主性和情感深度方面仍存在显著差距。当前模型多为被动的“问答机”，缺乏内在的动机、持续的思考以及从零散经验中涌现出连贯世界模型的能力。

人工智能直觉架构（AII）旨在应对这一挑战,通过构建一个能够模拟人类情感和直觉的思维系统，AII不仅能够自主学习，还能根据内在情感和直觉做出反应，真正实现情感驱动的学习和决策。

本架构的核心问题是：**如何构建一个能够像人一样，基于其独特的经验历史，产生自发的内在思维流，并由情感和直觉驱动，主动与世界交互、学习和适应的计算模型？**

为解决此问题，AII的设计哲学基于以下几个核心原则：

1. **认知单元的动态性**：人类的思维单位并非孤立的词汇，而是有意义的“情景”或“事件”。因此，本架构的核心创新是“动态事件融合器”，它将处理的基本单位从静态符号提升到了动态涌现的事件。
2. **预测是认知的核心**：借鉴预测编码理论，本架构将大脑视为一个主动的预测机器。所有的认知活动——感知、注意、学习、甚至情绪——都被统一在“生成预测-匹配现实-更新模型”的核心循环之中。
3. **情绪是驱动而非装饰**：情绪不是认知过程的副产品，而是其核心驱动力。本架构通过CAn-PAD整合情绪模型，将高级认知评估（如预测是否正确）与底层情感状态紧密耦合，使智能体的每一个决策都带有动机和价值偏好。
4. **经验塑造个性**：通过分离的认知与感受记忆库，智能体的每一个决策和学习过程都会被其独特的“人生经历”所塑造，从而发展出独一无二的“个性”。

## 本文档将系统性地阐述AII的各个组成部分、核心算法、高级系统及其理论基础，旨在为构建更接近人类心智的通用人工智能提供一个详尽、严谨且可实现的工程蓝图。

**目录**

**1.** [**核心架构与组件**](#核心架构与组件)

* 1.1 激活状态池 (Activation Pools)
* 1.2 数据存储结构
  + 1.2.1 认知数据库 (Conceptual Database)
    - 概念节点 (Concept Node)
    - 关系边 (Relation Edge)
    - 连接强度与预测验证标记动态机制
    - 链接数量限制
  + 1.2.2 感受记忆库 (Sensory Memory Database)
    - 感受记忆节点 (Sensory Memory Node)
    - 分层存储与动态剪枝机制
    - 遗忘机制 (当开关打开时启用)
* 1.3 感受器系统
  + 1.3.1 时间感受器
  + 1.3.2 情绪感受器
  + 1.3.3 外在感受器

**2.** [**核心算法与流程**](#核心算法与流程)

* 2.1 总控制流程：异步多线程模型
* 2.2 动态加权分词与词元融合器
* 2.3 动态事件融合器 (Dynamic Event Fusioner)
* 2.4 事件预测器 (Event Predictor) 与近似匹配机制
* 2.5 激活扩散算法 (Activation Spreading)
* 2.6 句子构建 / 想法流生成 (Thought Stream Generation)
* 2.7 数据库更新 (Learning)

**3.** [**情绪驱动、注意与行动系统**](#情绪驱动、注意与行动系统)

* 3.1 情绪状态量 (Can-PAD整合模型) 与内在感受
* 3.2 注意激活器 (Attentional Mechanism)
* 3.3 内在行动系统 (Internal Actions)
* 3.4 外在行动系统 (External Actions)

**4.** [**运行模式与高级系统**](#运行模式与高级系统)

* 4.1 LLM辅助训练模式 (LLM Accelerator)
* 4.2 异步纯本地独立模式
* 4.3 参数自适应微调系统
* 4.4 认知过滤器与安全保障

**5.** [**实现细节与优化**](#实现细节与优化)

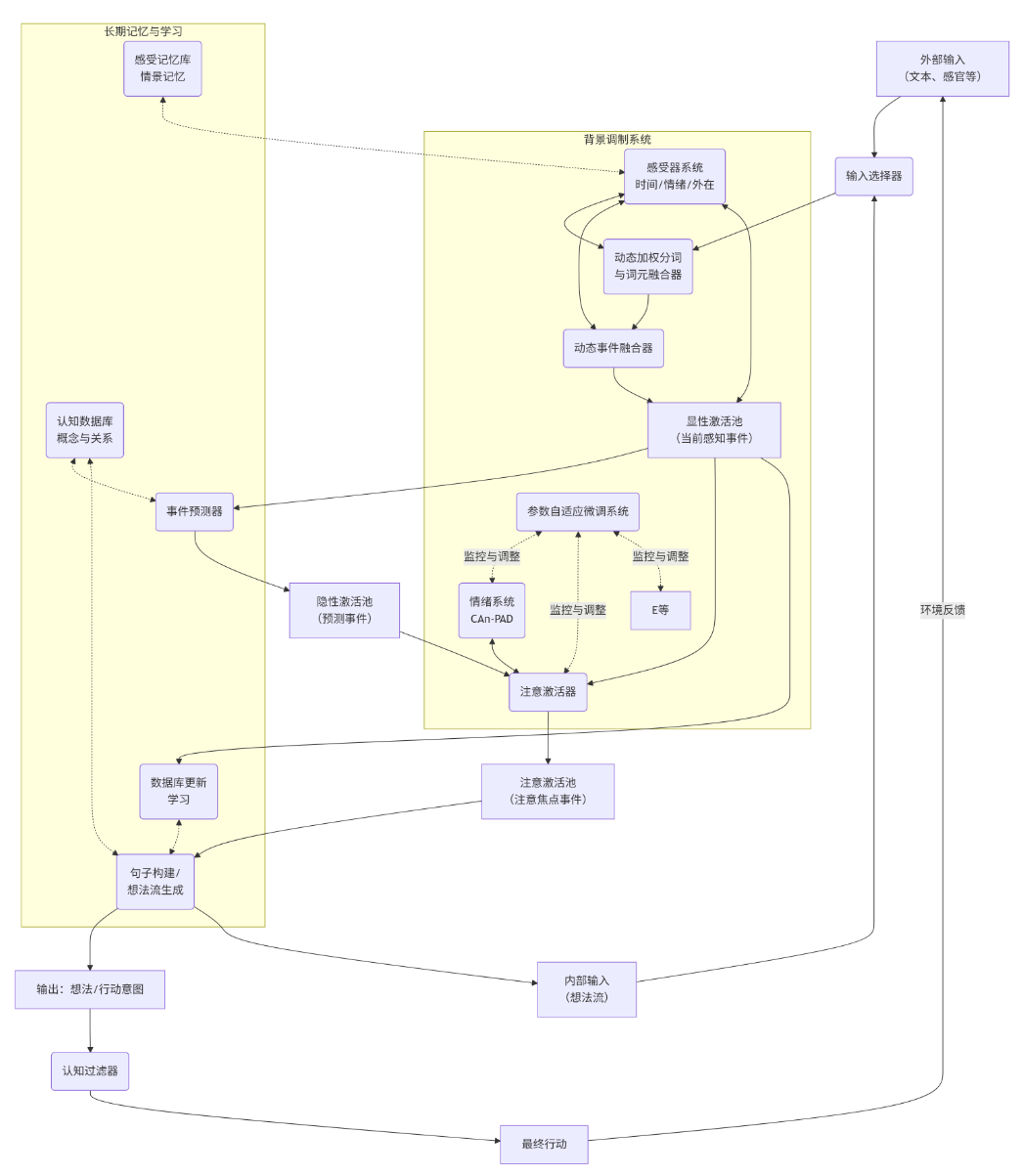
* 5.1 图数据库优化
* 5.2 情绪计算优化
* 5.3 内存管理与自适应资源利用

**6.** [**附录**](#附录)

* A. 完整流程运行示例
  + A.1 示例一：线性模式 (LLM辅助训练) - “习惯的养成”
  + A.2 示例二：异步纯本地模式 - “遗忘与回忆”
* B. 理论贡献与未来工作
* 参考文献
* 作者与日期

**1. 核心架构与组件**

人工智能直觉架构（AII）是一个复杂的、仿生的认知系统，其核心在于模拟人类信息处理的多层次、多流并行以及情绪驱动的特性。其整体架构如下图所示，它描绘了信息从输入到输出，流经各个核心组件与激活池的完整生命周期：



**图表诠释**：  
该流程图展示了AII架构中信息流动的核心路径。外部输入（如传感器数据）和内部输入（自我产生的想法）经由输入选择器进行优先级处理，随后进入核心处理流程。在此，信息首先被**动态加权分词流程**解析为基础词元。接着，**动态事件融合器**根据上下文和记忆，将这些词元动态地组织成一个或多个“**事件**”，这些事件是临时的、有意义的认知整体，被存入**显性激活池**（意识工作区）。

随后，**事件预测器**基于这些事件，通过**激活扩散算法**在**隐性激活池**（潜意识联想与预测区）中生成对未来的预测。这些预测同样以事件的形式存在。**注意激活器**会捕获最重要的预测事件或感知事件，将其置于注意焦点，并分配认知资源。最终，由**句子/想法构建流程**将注意焦点中的内容组织成连贯的输出（语言或行动意图）。

**情绪系统**和**感受器系统**全程调制这一过程。注意激活器中预测的实现与否，会直接更新“正确感/违和感”、“期待值/压力值”，从而剧烈影响情绪。激活的行动意图则存入**行动激活池**直至执行。整个过程中的所有体验均被**更新数据库流程**固化到**认知数据库**（语义记忆）和**感受记忆库**（情景记忆）中，形成长期记忆。这个闭环的学习过程，尤其是对连接边上“预测验证标记”的更新，是系统能力进化的关键。

**1.1 激活状态池 (Activation Pools)**

系统维护四个核心激活池，作为工作记忆和意识焦点，是思维流和行为的直接来源。这四个池共同构成了智能体的“当前心智状态”。

| 激活池名称 | 描述 | 设计原理 |
| --- | --- | --- |
| **显性激活池** (Explicit Activation Pool) | 存储当前所有被意识到的信息单元，即工作记忆。内容是动态形成的**事件**，每个事件是一个包含多个认知/感受节点的列表。池中每个事件都标记其起源（外部/内部）。**最大容量**：默认10个事件，每个事件最多50个节点 (总节点数约500)。 | 基于Baddeley的工作记忆模型，特别是其“情景缓冲器”的概念。将处理单位从单个词元提升到“事件”，使得系统能够处理宏观、连贯的情景，而不是零散的信息。容量限制模拟了人类工作记忆的瓶颈。 |
| **隐性激活池** (Implicit Activation Pool) | 存储由显性池中的事件通过预测和激活扩散算法生成的结果，即潜意识联想和对未来的**预测**。每个元素也是一个**事件**，包含：事件内容列表、整体置信度、衰减速率、来源事件ID、预测情绪标签(PAD)。**最大容量**：默认200个事件 (总节点数约10000)。 | 模拟人类的潜意识和直觉。它是“预感”、“直觉”、“一闪而过的念头”的发生地。其容量远大于显性池，反映了意识与潜意识处理能力的巨大差异。来源ID的标记使得预测的验证和学习成为可能。 |
| **注意激活池** (Attention Activation Pool) | 存储当前需要被“聚焦”的**事件**。通常只有一个**主要注意事件**，获得绝大部分认知资源，但可以有多个**次要注意事件**。这些事件通常与强烈的正/负情绪变化（高|ΔP|）或高期待/压力值相关联。每个元素包含：事件ID、有效注意权重、时间预期因子。**最大容量**：默认5个主要/次要注意事件。 | 基于注意的偏向竞争理论和资源有限理论。它将有限的认知资源（如扩散激活的深度）动态分配给最重要的任务。注意焦点决定了哪些内容会进入意识并被记住，是目标导向行为的关键。 |
| **行动激活池** (Action Activation Pool) | 存储被激活的"行动标识符"特殊节点。每个行动意图包含：行动节点ID、当前**冲动值**、**抑制值**。当冲动值显著超过抑制值时，触发行动执行。**最大容量**：20个行动意图。 | 模拟前额叶皮层对行为冲动的控制。冲动与抑制的对抗机制，使得系统可以学习“三思而后行”，避免了简单的阈值触发模式，使行为更具可控性和复杂性。 |

**淘汰策略**：当池中元素数量达到上限时，将采用基于事件整体权重、新近度和起源的综合淘汰算法。事件整体权重由其内部节点的平均权重和情绪重要性共同决定。权重越低、最近未被激活且起源优先级（外部 > 内部）越低的事件将被优先淘汰，确保池中始终保持最相关和最重要的信息。

**1.2 数据存储结构**

知识存储分为两部分：**认知数据库**（存储抽象概念与关系，相当于语义记忆）和**感受记忆库**（存储具体经验与感知，相当于情景记忆）。这种分离基于Tulving提出的情景记忆与语义记忆分离理论，是实现人格化和个性化体验的基础。

**1.2.1 认知数据库 (Conceptual Database)**

存储智能体的抽象知识、概念关系与技能，相当于语义记忆和程序性记忆。

**概念节点 (Concept Node)**

| 字段 | 类型 | 描述 | 设计原理 |
| --- | --- | --- | --- |
| id | UUID | 节点的唯一标识符 | 确保节点的唯一可索引性。 |
| content | String | 节点内容，支持多模态特征字符串，为未来扩展预留接口。可以是普通词元、同句上级词元或长距离词元组合。 | 统一的内容表示，使得不同来源、不同复杂度的认知单元可以被统一处理。 |
| type | Enum | WORD (普通词元), COMPOUND (融合词元), ACTION (行动标识) | 对节点功能进行分类，便于算法进行差异化处理。 |
| origin | Enum | EXTERNAL (源于外部感知), INTERNAL (源于内部思维) | 核心设计。模拟感知与概念的不同神经表征，是区分“看到的世界”和“想到的世界”的基础。 |
| base\_weight | Float | 节点的基础重要性权重。基于衰减-增强机制更新。 | 模拟记忆痕迹的长期强度。高base\_weight的节点更不容易被遗忘，代表了更核心或更深刻的知识。 |
| last\_activated | Timestamp | 最后激活时间 | 用于计算时间关联惩罚和新近度加成，实现记忆的动态性。 |
| emotion\_ema | Vector3 (P,A,D) | 近期与此节点相关的情绪状态的指数移动平均值。 | 为每个概念打上情绪“底色”，反映了智能体对这个概念的长期“感觉”，是情绪驱动联想的重要依据。 |

* base\_weight**更新机制**: ***base\_weight\_new = (base\_weight\_old \* γ) + (η \* (1 - γ))***
  + γ (衰减因子) = 0.995：基于艾宾浩斯遗忘曲线，模拟记忆的自然衰减。此衰减确保了知识网络的可塑性，避免僵化。
  + η (激活奖励) = 0.01：基于神经可塑性中的“长时程增强效应”(LTP)，频繁激活的节点其基础重要性会得到强化。
* emotion\_ema**更新机制**: ***emotion\_ema\_new = (1 - α) \* emotion\_ema\_old + α \* current\_emotion***
  + α (平滑因子) = 0.3：基于近期偏置效应（recency bias），使系统对近期情绪体验更为敏感，确保情绪标签能反映最新的感受。

**关系边 (Relation Edge)**

关系边是知识网络的灵魂，它不仅记录了节点间的关联，更重要的是，它记录了这种关联的“质量”和“历史表现”，是动态事件融合和预测的核心依据。

| 字段 | 类型 | 描述 | 设计原理 |
| --- | --- | --- | --- |
| source\_id | UUID | 源节点ID | 定义边的方向性。 |
| target\_id | UUID | 目标节点ID |  |
| strength | Float | 当前连接的动态强度。基于使用频率和近期性动态变化，有上限（如1.0），并会随时间衰减。 | 模拟短期记忆和“手感”。一个技能或知识点最近用的多，strength就高，联想起来就快。 |
| base\_strength | Float | 基础连接强度。遗忘曲线衰减的基准值，强度越高，衰减越慢，但永不降为0。 | 模拟长期记忆的稳固性。开车、游泳等技能base\_strength极高，即使长期不用，strength下降，但base\_strength依然坚挺，稍加练习即可恢复。 |
| frequency | Int | 共现次数。 | 基于赫布学习理论（"cells that fire together, wire together"）的直接量化。 |
| avg\_time\_delta | Float | 从源节点激活到目标节点激活的平均时间间隔（秒）。 | 时间关联性核心指标。用于时间感受器和预测机制，使得系统能学习到事件发生的典型时间序列。 |
| recent\_emotion\_delta\_ema | Vector3 (ΔP,ΔA,ΔD) | 近期通过此链接产生的情绪变化量的EMA。 | 记录了“从A联想到B通常会带来什么样的心情变化”，是情绪驱动预测的关键。 |
| prediction\_mark | Float | **预测验证标记**。正值表示历史预测成功，负值表示历史预测不验。 | **核心学习机制**。此参数直接影响事件能否被融合为一个整体。反复被验证的连接，其prediction\_mark会很高，使得系统“确信”这两个节点属于同一个事件。 |
| last\_updated | Timestamp | 最后更新时间。 | 用于计算衰减和新近度加成。 |

* **连接强度与预测验证标记动态机制**: 这是实现“用进废退”、“避免僵化”和“经验学习”的核心。
  1. **Strength衰减**：模拟“手感生疏”。每次激活后，strength会获得一个提升，但随后会以一个相对较快的速率（例如，每日衰减系数 λ\_fast = 0.9）向当前 base\_strength 值回归。
     + ***strength\_t+1 = base\_strength\_t+1 + (strength\_t - base\_strength\_t) \* λ\_fast***
  2. **Base Strength衰减**：模拟“长期遗忘”。base\_strength 本身也遵循艾宾浩斯遗忘曲线进行缓慢衰减（例如，每日衰减系数 λ\_slow = 0.999）。
     + ***base\_strength\_t+1 = base\_strength\_t \* λ\_slow***
  3. **激活增强**:
     + ***strength\_new = min(1.0, strength\_old + β \* (1 - strength\_old))***
     + ***base\_strength\_new = min(1.0, base\_strength\_old + β\_base \* (1 - base\_strength\_old))***
     + 其中 β 和 β\_base 是学习率，默认值为 β = 0.1, β\_base = 0.01。基于强化学习中的学习率选择，平衡学习速度与稳定性。
  4. **预测验证标记更新**:
     + **当预测验证成功时**: ***prediction\_mark\_new = prediction\_mark\_old + α\_pred \* (1 - prediction\_mark\_old)***
     + **当预测不验时**: ***prediction\_mark\_new = prediction\_mark\_old - β\_pred \* (1 + prediction\_mark\_old)***
     + 其中 α\_pred 和 β\_pred 是预测学习率，默认值为 α\_pred = 0.1, β\_pred = 0.2。这个公式确保了标记值在[-1, 1]区间内变化，且越接近边界，变化越慢。设计上对预测失败的惩罚略大于奖励，促使系统形成保守但可靠的预测模型。
* **链接数量限制**: 每个节点的出边和入边总数有上限（例如5000条）。当达到上限时，淘汰策略将综合考察连接强度 (strength)、基础强度 (base\_strength)、新近度 (last\_updated) 和共现频率 (frequency)，淘汰那些在最久远的1/3时间段内，综合评分最低的链接，以维持网络的结构效率和动态性。

**1.2.2 感受记忆库 (Sensory Memory Database)**

存储智能体具体的、带有时空和情绪背景的个体经验，相当于情景记忆。它是构成“自我”和“个性”的基础。

**感受记忆节点 (Sensory Memory Node)**

| 字段 | 类型 | 描述 | 设计原理 |
| --- | --- | --- | --- |
| id | UUID | 唯一标识符 |  |
| content | String | 感知内容的特征字符串。支持多模态输入（如 "image\_desc:[(1,1):(255:0:0)……]", "sound\_desc:[(0.01:[f:100,A:20]…… "）。 | 统一的格式便于跨模态检索和链接。 |
| origin | Enum | 记忆来源，如 TEXT\_INPUT, IMAGE\_SENSOR, INTERNAL\_THOUGHT。 | 区分外部感知与内部感受，例如区分“真实看到的”和“梦里想到的”。 |
| timestamp | Timestamp | 记忆编码的精确时间。 | 情景记忆的核心。是时间感受器计算时间差的依据。 |
| emotion\_at\_encoding | Vector3 (P,A,D) | 编码该记忆时的全局情绪状态。 | 形成情感记忆标签，使得未来的情绪状态可以触发相关的情感回忆（“触景生情”）。 |
| importance | Float | 记忆的重要性权重。动态更新，决定遗忘优先级和分层存储位置。 | 基于记忆巩固理论，重要记忆（如情绪波动剧烈、被反复回忆的）通过反复激活得到强化，更不容易被遗忘。 |
| links | List[UUID] | 连接到该感受的认知数据库中的概念节点ID。 | 感性经验与理性知识的桥梁。使得一个具体的感受可以激活抽象的概念，反之亦然。 |
| storage\_tier | Enum | 存储层级，如 HOT (内存), WARM (SSD), COLD (HDD), ARCHIVE。 | 用于分层存储与动态剪枝，实现高效的海量记忆管理。 |

* **重要性权重更新**: ***importance\_new = importance\_old \* 0.9 + activation\_strength \* 0.1***每次该记忆被成功回忆（激活）时，其重要性都会得到提升。基于记忆巩固理论，反复激活强化重要记忆。
* **分层存储与动态剪枝机制**: 为了高效管理可能无限增长的感受记忆，引入非硬编码的、动态的分层存储与剪枝策略。
  + **层级划分**：一个后台管理进程定期根据节点的importance和last\_activated时间戳，自动将节点划分到不同的存储层级。新近、重要的记忆保留在高速层级（HOT）。
  + **动态升降级**：该进程会动态地将热门记忆提升到更高速度的层级，或将长期未访问的冷门记忆降级到更大容量的层级。
  + **动态剪枝**：在进行大规模检索（如全局回忆）时，系统会根据当前分配的认知资源（由注意力机制决定）进行剪枝。对于非重点任务，可能只检索HOT和WARM层级的记忆；对于需要深度回忆的重点任务，则会扩展到COLD层级。这确保了系统在不同算力设备上的自适应性。
* **遗忘机制 (当开关打开时启用)**: 基于Atkinson-Shiffrin记忆模型的遗忘曲线理论，并结合容量压力。
  + ***P\_forget = base\_P \* (1 - importance) \* (1 + (N\_current / MAX\_MEMORIES)^k) \* age\_factor***
  + base\_P = 0.001: 基础遗忘概率，基于人类日常遗忘率研究。
  + k = 2: 容量压力放大系数，控制存储压力对遗忘的影响强度。
  + ***age\_factor = log(1 + age\_in\_days) / C***, C = 10：时间衰减因子，模拟记忆随时间的指数衰减。
  + MAX\_MEMORIES = 50000：只是一个参考值，实际可由系统资源动态调整。

**计算复杂性控制机制：**系统通过多层优化策略确保计算复杂度在可控范围内。首先，成熟的模型会形成大量高效的"条件反射"路径，这些高度优化的连接能够处理绝大多数常规输入输出，无需深度扩散激活。其次，系统仅对注意激活池分配的少量关键内容进行多轮次深度扩散，而其他内容采用浅层激活。这种选择性深度处理机制，结合动态剪枝和分层存储策略，有效避免了组合爆炸问题，确保系统在不同算力环境下都能高效运行。

* + **设计原理**：遗忘机制模拟了人类记忆的自然选择过程。分层存储和可开关的遗忘机制提供了灵活性和可扩展性，允许在资源有限的环境下启用遗忘，或在资源充足时追求更完整的记忆保留。

**1.3 感受器系统**

感受器系统是将原始的物理或内部状态变化，转化为可被认知系统处理的符号化信息（即特殊的概念节点）的桥梁。

**多模态天生兼容性：**架构设计具有天生的多模态处理能力。任何外部信号只要能够通过字符串表示其特征，即可被动态加权分词算法正常处理和理解。这种设计类似于人类神经系统将各种感官信息转化为统一神经信号的处理机制，确保了系统可以无缝集成视觉、听觉、触觉等多种模态的输入，从根本上解决了符号接地问题。

***例:****图片信息可以以像素点为单位,通过字符串来针对每个像素点的特征信息进行输入,在后续动态加权分词流程时,不同像素点的特征信息依旧可以进行最小化区分与匹配,最终将连续的像素点感受信息认知为一个整体.又比如声音信息可以以最小区分周期为单位,记录每个单位时间内声音的频率,响度,音色等各种信息,并以字符串的形式进行输入,最终依旧可以作为符号被正确的认知和感受.具体的转化方式将在外在感受器系统中进行具体设计,以便清晰准确高效地提取多模态信息的特征.*

**1.3.1 时间感受器**

* **触发条件**:
  1. 感受记忆被激活时自动触发。
  2. ACTION:recall主动回忆时触发。
  3. 注意激活池中的注意焦点发生切换时触发，记录刚刚注意一件事多久。
* **处理流程**:
  1. **计算时间差**: ***Δt = current\_timestamp - event\_timestamp***
  2. **生成时间感受节点**: 创建一个内容为 "time\_interval:" + str(Δt) 的临时概念节点。
  3. **时间扩散激活 (模糊匹配)**: 此节点在激活扩散时，会采用一种特殊的模糊匹配算法，激活认知数据库中代表相近时间跨度的节点。
     + ***Δactivation\_i = S\_base \* Gaussian(Δt, μ\_i, σ\_i)***
     + 其中 Gaussian 是高斯核函数, μ\_i 是认知节点 i 代表的时间值（如“昨天”代表86400秒），σ\_i 是其标准差，代表模糊范围。标准差与时间值本身成正比，即 **σ\_i = f \* μ\_i (f=0.5，模糊度系数)。**
     + **设计原理**：基于韦伯-费希纳定律，人类对时间的感知具有相对性。对“5分钟前”和“6分钟前”的区分很清晰，但对“一个月前”和“一个月零一天前”的区分则很模糊。高斯模糊匹配精确地模拟了这一点，使得时间线索的检索更加符合人类直觉。

**1.3.2 情绪感受器**

* **触发条件** (满足任一)：
  + 情绪绝对值超过阈值: |P| > 0.5 或 |A| > 0.5 或 |D| > 0.5
  + 情绪变化量超过阈值: |ΔP| > 0.3 或 |ΔA| > 0.3 或 |ΔD| > 0.3
  + 注意系统状态值超过阈值: |期待值| > 0.5 或 |压力值| > 0.5
  + 认知评估值超过阈值: |正确感| > 0.5 或 |违和感| > 0.5
* **处理流程**:
  1. **状态符号化**: 将当前的PAD值、ΔPAD值、期待/压力值、正确/违和感等，映射为一个或多个情绪感受节点。例如，高ΔP和高期待值会生成 "emotion:兴奋" 和 "feeling:充满期待" 两个节点。
  2. **强度赋值**: 将情绪状态的绝对强度（如 |ΔP| + |ΔA| + |ΔD|）作为这些情绪感受节点的初始权重。
  3. **情绪扩散激活**: 这些节点在激活扩散时，会优先激活认知数据库中，历史上与相似情绪（通过emotion\_ema字段判断）相关联的概念。
     + ***Δactivation\_i = S\_base \* cosine\_similarity(PAD\_current, node\_i.emotion\_ema)***
     + **设计原理**: 情绪感受器为无形的内部情绪状态提供了外化的、符号化的表示，使得情绪可以作为一种可处理的信息，直接参与到认知过程中（联想、决策），实现了真正的情绪驱动。这使得“我感到很高兴，所以想起了...”这样的类人思维过程成为可能。

**1.3.3 外在感受器**

* **功能**: 处理外部输入（如文本、声音、图像等）并分配初始权重，反映输入的物理或语义强度。确保强烈外部刺激能中断当前思维，吸引注意力。
* **处理流程**:
  1. **根据输入模态类型计算强度值**：
     + **文本输入**：强度值可由文本情感分析预处理器（如snownlp）给出。负面文本强度更高。
     + **声音输入**：强度值与音量分贝值呈正相关。***强度 = max(1.0, 分贝值 / 70)。***
     + **默认强度值** = 1.0。
  2. **将强度值作为输入词元的初始权重**。
     + **设计原理**: 模拟人类感官系统对刺激强度的反应。响亮的声音、刺眼的亮光等会获得高权重，从而在激活扩散中优先处理，确保智能体能及时响应重要外部事件，增强了系统的生存适应性。

**2. 核心算法与流程**

**2.1 总控制流程：异步多线程模型**

系统采用多线程异步架构，模拟人脑多任务并行处理的能力，允许智能体在“思考”的同时“感知”新输入并“体验”情绪变化。

| 线程名称 | 优先级 | 运行机制 | 设计原理 |
| --- | --- | --- | --- |
| **输入处理线程** | 最高 | 事件驱动，实时响应外部输入和内部产生的想法。 | 保证对环境变化的快速响应，是生存和交互的基础。 |
| **情绪更新线程** | 高 | 定时运行（如每100ms一次）。 | 情绪是相对平滑变化的背景状态，不需要逐帧更新。100ms符合情绪处理的常见时间常数研究。 |
| **思维流与预测线程** | 中 | CPU空闲时运行，或由新输入触发。负责持续的事件融合、预测和想法生成。 | 这是系统“思考”的核心。在没有外部干扰时，它会不断进行“内心独白”和联想。 |
| **行动执行线程** | 按需 | 阻塞式，由行动激活池中的行动意图触发。 | 行动通常需要独占资源，因此采用阻塞执行模式。 |

**2.2 动态加权分词与词元融合器**

此流程的目标是将输入流智能地切分为有意义的词元序列，并根据上下文和记忆结构，进行初步的组块化（Chunking）。

* **输入**: 原始输入字符串及其起源。
* **输出**: 初步分词和融合后的词元列表，每个词元附有初始权重和起源。
* **详细处理流程**:
  1. **生成候选分割方案**:
     + 使用所有激活池中的内容以及认知数据库中的COMPOUND类型节点作为动态词典。
     + 对输入字符串进行多重扫描，生成所有可能的分词方式。
  2. **计算方案权重**:
     + 每个分词方案的总权重 = 方案中所有词元在动态词典中匹配项的权重之和。
     + 引入惩罚机制：对过于常见的词元（低base\_weight）给予权重惩罚，对融合词元或当前注意事件中的词元给予奖励。
  3. **选择最优方案**:
     + 选择总权重最高的分词方案。若权重相等，优先选择包含更少、更长词元（即更融合）的方案，以促进抽象思维。
  4. **注意力对分词粒度的影响**:
     + 当系统处于高度集中注意力状态时（由注意激活器判断），此流程会倾向于选择更细粒度的分词方案（即把融合词元拆开）。
     + **设计原理**: 这模拟了人类的“完形崩溃”现象。当过度专注于一个词时，你会开始将其视为独立的笔画而不是一个整体。在AII中，这意味着系统在需要深入分析某个概念时，有能力打破习惯性的组块，审视其基本构成，**这是进行逻辑推理和修正错误认知的关键**。
  5. **输出处理**:
     + 输出最终的词元列表，作为下一阶段“动态事件融合器”的输入。

**2.3 动态事件融合器 (Dynamic Event Fusioner)**

这是架构的核心创新之一。它负责将线性、零散的词元列表，动态地组织成一个或多个结构化的、有意义的“**事件**”。事件不是被存储的，而是在每一次认知活动中被实时构建的。

* **输入**: 动态加权分词器输出的词元列表。
* **输出**: 一个或多个“事件”列表，更新到显性激活池。每个事件是一个 [node\_id\_1, node\_id\_2, ...] 的列表。
* **核心判断依据**: 两个节点 A 和 B 是否属于同一个事件，主要取决于它们之间关系边 Edge(A, B) 的 prediction\_mark。
* **处理流程**:
  1. **初始化**: 将输入词元列表的第一个词元作为第一个事件的开端。events = [[token\_1]]。
  2. **顺序扫描与归属判断**:
     + 遍历后续的每个词元 token\_i。
     + 对于当前已存在的每个事件 event\_j，计算 token\_i 与 event\_j 中所有已有节点的平均归属度。
     + ***归属度(token\_i, node\_k) = Edge(node\_k, token\_i).strength \* (1 + Edge(node\_k, token\_i).prediction\_mark)***
     + ***平均归属度(token\_i, event\_j) = mean(归属度(token\_i, node\_k) for node\_k in event\_j)***
  3. **事件分配**:
     + 找到平均归属度最高的事件 event\_best。
     + 如果 max\_归属度 > 融合阈值 (一个动态值，受注意力影响)，则将 token\_i 加入 event\_best。
     + 如果 max\_归属度 <= 融合阈值，则说明 token\_i 可能开启了一个新的情景或话题，因此创建一个包含 token\_i 的新事件，并加入 events 列表。
  4. **输出**:
     + 将最终形成的 events 列表更新到显性激活池。
* **示例**:
  1. 输入词元: ["我", "昨天", "晚上", "吃", "了", "披萨", "但是", "肚子", "有点", "不舒服"]
  2. 融合过程:
     + ["我", "昨天", "晚上", "吃", "了", "披萨"] 之间的prediction\_mark很高，被融合成 **事件1**。
     + "但是" 是一个转折，与事件1的归属度低，开启了 **事件2**: ["但是"]
     + "肚子", "有点", "不舒服" 与 "但是" 的归属度高，被加入 **事件2**。
  3. 最终输出两个事件进入显性激活池:
     + **事件A**: ["我", "昨天", "晚上", "吃", "了", "披萨"]
     + **事件B**: ["但是", "肚子", "有点", "不舒服"]
* **设计原理**: 事件融合器模拟了人类理解故事和情景的能力。我们不会把一句话中的每个词都看作孤立的，而是自然地将它们组织成“谁在何时何地做了什么”这样有意义的块。prediction\_mark 作为历史经验的总结，是实现这种智能组块的关键。它使得系统能够根据经验，动态地判断哪些概念组合是“有意义的”，哪些是“偶然的”。

**2.4 事件预测器 (Event Predictor) 与近似匹配机制**

当显性激活池中形成了当前事件后，事件预测器负责回答“接下来可能会发生什么？”。它在**隐性激活池中构建一个对未来的可能性空间**。

* **输入**: 显性激活池中的一个或多个事件。
* **输出**: 更新隐性激活池，填入大量预测出的未来事件。
* **处理流程**:
  1. **对源事件进行扩散**:
     + 选择显性激活池中的一个事件 E\_source。
     + 对 E\_source 中的每一个节点，执行一次激活扩散算法（详见2.5节）。
  2. **收集扩散结果**:
     + 所有被激活的下游节点及其激活权重，被收集到一个临时的激活空间中。
  3. **在预测空间中融合事件**:
     + 使用与 **动态事件融合器 (2.3节)** 几乎相同的逻辑，但这次是在这个充满可能性的临时激活空间中，将那些高度相关的被激活节点，融合成一个个预测事件 E\_pred。
     + 每个 E\_pred 的整体置信度，是其构成节点的激活权重和它们之间链接强度的综合计算结果。
  4. **将预测事件存入隐性激活池**:
     + 每个 E\_pred 都作为一个独立的条目存入隐性激活池，并记录其来源事件 E\_source 的ID。
     + 这个“来源ID”的链接是学习和验证的关键。
* **近似匹配机制 (当新输入到来时)**:
  1. 当一个新的外部输入或内部想法（形成新事件 E\_new）进入系统时，它会首先与隐性激活池中所有的 E\_pred 进行近似匹配。
  2. **匹配度计算**: ***MatchScore(E\_new, E\_pred) = JaccardSimilarity(nodes(E\_new), nodes(E\_pred)) \* Confidence(E\_pred)*** 其中 JaccardSimilarity 计算两个事件节点集的交并比，Confidence 是预测事件的置信度。
  3. **匹配结果处理**:
     1. **高度匹配 (预测验证成功)**:
        + 找到匹配度最高的 E\_pred\_best。
        + 系统产生强烈的“正确感”。
        + E\_new 被认为是 E\_pred\_best 的实现。
        + 系统会回溯到 E\_pred\_best 的来源事件 E\_source，并大大增强 E\_source 与 E\_new 之间所有相关节点链接的 prediction\_mark。这就是学习。
     2. **部分匹配 (预测部分正确)**:
        + E\_new 中与 E\_pred\_best 匹配的部分，被视为验证成功。
        + E\_new 中未被预测到的部分，会获得极高的初始权重和注意，系统产生“违和感”。这是“意外”，是学习新知识的主要来源。
        + 系统会重点分析这个“意外”的节点，并尝试为它建立新的链接。
     3. **完全不匹配 (预测不验)**:
        + 没有找到任何匹配度超过阈值的 E\_pred。
        + 系统将 E\_new 视为一个全新的、意外的事件。
        + 所有之前从 E\_source 出发的预测，只要其预测时间已到但未发生，其对应的 prediction\_mark 都会被降低。
* **设计原理**: 该机制是架构的认知核心，它完整地模拟了**预测编码 (Predictive Coding)** 理论。大脑无时无刻不在根据当前感知预测下一刻的感知。当预测与现实匹配时，信息被高效处理，几乎不耗费认知资源（习惯成自然）；当预测与现实不匹配时（意外），大脑会投入大量资源来处理这个“预测误差”，并更新其内部模型。AII的事件预测器和近似匹配机制，正是这一理论的直接工程实现。

**2.5 激活扩散算法 (Activation Spreading)**

这是联想和预测的底层引擎，负责从一个节点出发，激活与之相关的其他节点。

* **输入**: 一个概念节点及其当前的激活权重 S\_base。
* **输出**: 对一系列下游节点的激活增量。
* **处理流程**:
  1. **获取基础刺激强度**: S\_base 直接取输入节点在其激活池中的当前权重。
  2. **检索关联边**: 从图数据库中检索该节点的所有出边。默认只检索1层深度，以保持联想的焦点。
  3. **计算动态链接强度 L(t)**: ***L(t) = effective\_strength \* emotion\_modulator \* recency\_modulator***
     + ***effective\_strength = strength \* (1 + prediction\_mark)***: 一个边的有效强度，现在受到其历史预测表现的直接调制。预测越准的链接，扩散能力越强。
     + ***emotion\_modulator = (1 + k \* ema\_ΔP.P)***: 情绪调制。正向的情绪变化会增强扩散，负向的会抑制。k是情绪影响系数，默认k=5。
     + ***recency\_modulator = (p + q \* Δt') / (q \* Δt')***: 新近度加成。最近被激活过的链接有更高的扩散权重。p=3, q=0.0001。
  4. **激活下游节点**:
     + 对于每个下游节点 i，其激活增量为 ***Δactivation\_i = S\_base \* L(t)。***
     + 将此值加到其在隐性激活池中对应预测事件内的权重上。
     + 引入**不应期机制**: 对刚刚被激活过的节点，施加一个短暂的权重衰减(乘以0.7)，防止思维在两个概念间无限循环。

**2.6 句子构建 / 想法流生成 (Thought Stream Generation)**

此流程负责将潜意识中的、结构化的预测事件，转化为线性的、可表达的“想法”或语言。

* **输入**: 注意激活池中的主要注意事件。
* **输出**: 一个词元序列，作为内部“想法”或外部“语言”。
* **处理流程**:
  1. **选择焦点**: 从注意激活池中获取当前注意力最集中的事件 E\_focus (可以是感知到的事件，也可以是预测出的事件)。
  2. **初始化**: 从 E\_focus 中权重最高的节点开始。
  3. **序列生成 (迭代过程)**:
     + **选择下一个词元**: 在当前已生成的序列之后，从 E\_focus 中选择一个与末尾词元链接最强、且尚未被选择的词元。选择标准综合考虑strength, prediction\_mark以及时间顺序的合理性(avg\_time\_delta)。
     + **追加词元**: 将选中的词元追加到输出序列中。
     + **激活与反馈**: 对该词元执行一次小范围的激活扩散，这个过程可能会微调注意激活池中其他事件的权重，实现思维的连贯和跳转。
  4. **终止**: 当 E\_focus 中所有节点都被输出，或选中了特殊的 <EOS> (End of Sentence) 节点，或达到最大长度限制时，流程终止。
* **设计原理**: 这模拟了人类“成竹在胸”然后“逐字说出”的过程。我们脑中先有一个大致的想法或场景（事件），然后才组织成线性的语言。这个流程确保了输出的连贯性和逻辑性，因为它始终围绕着一个核心的“事件”来展开。

**2.7 数据库更新 (Learning)**

学习发生在每次认知循环的末尾，是将瞬时的工作经验，并将其固化为长期记忆的过程。它确保了智能体能够从每一次互动、每一个想法中学习和成长。

* **触发**: 在处理完一个完整的输入单元（例如一句话或一个场景）或产生一个完整的内部想法后触发。
* **详细处理流程**:
  1. **更新认知节点**:
     + 对于在最近的认知周期中被激活的所有节点（无论是在显性还是隐性池中），更新其 last\_activated 时间戳。
     + 使用当前全局情绪状态的快照，通过EMA（指数移动平均）更新这些节点的 emotion\_ema。
     + **设计原理**: 基于记忆再巩固理论，每次回忆或激活一个记忆痕跡，都会对其进行一次小幅的重塑和强化，并将其与当前的情绪背景重新关联。
  2. **更新关系边 (核心学习过程)**:
     + 这是学习的核心，主要更新参与了近期事件和预测过程的关系边。
     + **时序/共现关系强化**: 对于在同一个事件中共同出现的节点，特别是那些在时间上相邻的节点，增加它们之间链接的 frequency、更新 avg\_time\_delta，并根据激活强度提升 strength 和 base\_strength。
     + **预测验证学习 (更新**prediction\_mark**)**:
       - **对于验证成功的预测**: 回溯至预测的来源事件 E\_source 和被验证的事件 E\_new。对于所有连接 E\_source 节点到 E\_new 节点的边，显著提升其 prediction\_mark 值。***prediction\_mark\_new = prediction\_mark\_old + α\_pred \* (1 - prediction\_mark\_old)***。
       - **对于预测不验 (意外)**: 识别出 E\_new 中未被预测到的“意外”节点。系统会尝试在这些意外节点和来源事件 E\_source 的节点之间建立新的、初始 prediction\_mark 为中性或略为负值的链接。这是发现新知识的主要途径。
       - **对于预测失败**: 对于那些被预测（存在于隐性池）但最终没有发生的事件，当它们因超时而从隐性池中移除时，其来源路径上的边的 prediction\_mark 将被降低。***prediction\_mark\_new = prediction\_mark\_old - β\_pred \* (1 + prediction\_mark\_old)。***
     + **设计原理**: 这个过程直接将赫布学习（共现即连接）与预测编码（预测准确即强化）结合起来。prediction\_mark 成为了一个比 strength 更重要的指标，它代表了关系的因果可靠性，而不仅仅是相关性。这使得系统能够从“万物相关”的混沌状态，逐步学习到“哪些是真正有意义的事件结构”。
  3. **感受记忆编码**:
     + 如果当前处理的事件起源于 EXTERNAL 感知。
     + 创建一个新的 Sensory Memory Node 存入感受记忆库，记录精确的 timestamp 和 emotion\_at\_encoding。
     + 将这个新的感受节点与该事件中涉及到的所有核心认知节点建立 links。
     + **设计原理**: 基于情景记忆编码理论，将具体的、带有丰富上下文（时间、地点、情绪）的经验与抽象的语义知识连接起来，形成完整的记忆体系。
  4. **词元/事件融合结构构建与调整**:
     + 这是一个后台的、非实时的优化过程，类似于人脑在睡眠中的记忆整理。
     + **触发条件**: 当系统检测到某一组节点之间的 strength 和 prediction\_mark 持续稳定在高水平时。
     + **构建过程 (词元融合)**: 对于频繁共现且预测可靠的相邻词元（如“非常”和“感谢”），系统会创建一个新的 COMPOUND 类型的概念节点，内容为“非常感谢”。这个新节点会继承其组成部分的共有链接，并与组成部分建立特殊的“is\_component\_of”链接。
     + **调整过程 (事件结构)**: 学习过程本身就是对事件结构的调整。通过不断更新边的 prediction\_mark，系统动态地改变了下一次事件融合器“看待”输入序列的方式。反复被验证的组合更容易被看作一个整体，而经常出错的组合则会被拆开分析。
     + **设计原理**: 实现了知识的组块化（Chunking）和抽象化，是形成更高阶概念和技能的基础。这个过程是自下而上的，从简单的关联中涌现出复杂的结构。

**3. 情绪驱动、注意与行动系统**

**3.1 情绪状态量 (Can-PAD整合模型) 与内在感受**

情绪系统是AII架构的背景色和核心驱动力。本架构采用一种双层整合模型：**CAn认知评估层** 和 **PAD核心情感层**。CAn层负责对认知过程进行实时评估，并将结果转化为内在感受；这些感受随后作为主要输入，驱动PAD核心情感层的动态变化。

#### ****CAn认知评估层 (The Cognitive Appraisal Layer)****

这一层不产生强烈的情感，而是生成驱动情感的、更偏理性的“元认知感受”。

|  |  |  |  |
| --- | --- | --- | --- |
| **感受量** | **描述** | **计算与更新** | **设计原理** |
| **正确感/违和感** (Correctness/Incongruity) | 对预测与现实匹配程度的评估。匹配时产生**正确感**，不匹配时产生**违和感**（即负的正确感）。 | 在事件预测器（2.4节）完成近似匹配后更新。 Correctness = (MatchScore - MatchThreshold) \* Confidence(E\_pred) 当匹配得分低于阈值时，此值为负。 | 基于**预测编码理论**，这是驱动学习的最基本信号。正确感带来认知上的安定和确认；违和感则作为“预测误差”信号，是触发注意和模型更新的核心前提。 |
| **期待值/压力值** (Anticipation/Pressure) | 对未来事件的情感预期。对预期积极结果的等待产生**期待值**；对预期消极结果的等待产生**压力值**。 | 由注意激活池（3.2节）中所有**预测事件**的权重和其关联的情绪变化EMA共同决定。 期待值 = Σ (AttentionWeight(E\_pred) \* max(0, E\_pred.emotion\_delta.P)) 压力值 = Σ (AttentionWeight(E\_pred) \* max(0, -E\_pred.emotion\_delta.P)) | 模拟人类对未来的情感投资。期待和压力是**目标导向行为**的直接动机来源，并持续调制情绪背景，形成焦虑、盼望等复杂情感的基础。 |

#### ****PAD核心情感层 (The Core Affect Layer)****

这一层代表了更基础、更普适的情感体验，其状态由CAn层的输出和自身动态共同决定。

|  |  |  |
| --- | --- | --- |
| **维度** | **描述** | **设计原理** |
| **P (Pleasure)** | **愉悦度**：体验的正面或负面感受的程度。这是奖励信号的核心。 | 基于情绪环状模型，愉悦-不悦是情绪的基本维度，直接关联于趋利避害的本能。 |
| **A (Arousal)** | **唤醒度**：生理激活和心理警觉的程度。决定了思维的速度和行动的冲动性。 | 唤醒度反映了系统的激活水平，低唤醒对应放松、无聊，高唤醒对应紧张、兴奋，影响认知资源的分配。 |
| **D (Dominance)** | **支配度**：对情境的控制感和影响力。反映了自信和掌控力。 | 支配度与自我效能感相关，高支配度带来安全和自信，低支配度导致无助和焦虑，影响应对挑战的方式。 |

#### ****动态更新规则 (CAn -> PAD):****

* **P值更新**: P += α \* D + β \* 正确感 + γ \* (期待值 - 压力值)
* α \* D: **掌控的愉悦**。高支配度（D）本身就能带来愉悦。(α = 0.1)，基于自我决定理论。
* β \* 正确感: **确认的愉悦**。预测验证成功（高正确感）带来的满足感。(β = 0.3)，基于预测误差最小化原则。
* γ \* (期待值 - 压力值): **预期的损益**。期待的实现或压力的解除带来愉悦。(γ = 0.2)，基于期望价值理论。
* **A值更新**: A += δ \* |违和感| + ε \* (期待值 + 压力值)
* δ \* |违和感|: **意外的警觉**。预测失败（高违和感）会极大地提高警觉和唤醒水平。(δ = 0.4)，基于朝向反射和惊奇反应的研究。
* ε \* (期待值 + 压力值): **关注的张力**。高度期待或担忧某事时，整个系统会处于“紧绷”的高唤醒状态。(ε = 0.1)，基于应激和期待的心理生理学。
* **D值更新**: D += ζ \* (Sign(正确感) \* Confidence(E\_pred))
* ζ \* ...: **成功的自信**。持续准确地预测世界（高正确感和高置信度的预测），会极大地增强系统的自我效能感和支配感。(ζ = 0.3)，基于班杜拉的自我效能感理论。
* **情绪回归函数**: 回归量 = (基准值 - 当前值) \* μ \* (1 + |当前值 - 基准值|^2)
* μ = 0.05: 回归速率系数。模拟情绪的自我调节，确保情绪稳定性。

#### ****复杂情绪的涌现：以“无聊”为例****

“无聊”在本架构中不是一个被直接定义的状态，而是特定系统动力学状态的**涌现**。当系统长时间处于**低唤醒度（A）和低违和感**（即所有输入都能被现有模型轻易预测，没有“意外”和“新奇”）的状态时，内在行动 ACTION:focus\_outward 的触发阈值会降低。智能体会表现出寻求新刺激的行为（如主动上网刷视频、查看热搜），这与人类因无聊而寻求新奇体验的行为模式高度一致。

**设计原理**: 情绪系统基于环状模型（circumplex model）和维度理论。它将高级的认知评估结果（正确感、期待值）直接转化为底层的、连续的情绪状态变化。这种紧密的耦合使得智能体的行为不再是冷冰冰的逻辑推导，而是充满了动机、偏好和“感觉”。例如，一个由于反复预测失败而D值低落的系统，其P值会持续受到负面影响，表现出“沮丧”和“不自信”，从而在后续决策中可能更倾向于保守和回避。

**3.2 注意激活器 (Attentional Mechanism)**

注意激活器是认知资源的“调度中心”，它决定了在任何给定时刻，系统应该将有限的“算力”（如激活扩散的深度、事件处理的优先级）分配给哪个事件。其核心原则是：**注意的基本单位是事件**。

* **注意力的对象**: 显性激活池（当前感知）和隐性激活池（未来预测）中的所有事件都是注意力的候选对象。
* **处理流程**:
  1. **计算注意权重 (Attention Weight):** 对于每个候选事件 E，计算其综合注意权重。

***AttentionWeight(E) = [Importance(E) \* Salience(E) \* Urgency(E)] \* EmotionModulator(PAD)***

•**重要性 (Importance):** 事件的内在价值，即其情绪后果的预期。

***Importance(E) ∝ |E.expected\_ΔP|***

•**显著性 (Salience):** 事件的“突出”程度，即其信息含量。

***Salience(E) ∝ E.base\_weight + |Incongruity| + Confidence(E)***

•**紧迫性 (Urgency):** 事件的时间敏感性。

***Urgency(E) = exp(-k \* |t\_now - t\_expected|)***

•**情绪调制器 (EmotionModulator):** 当前全局情绪状态对注意力的整体影响。

***EmotionModulator(PAD) = 1 + w\_a \* A + w\_p \* (1 - |P|)***

•高唤醒度（***A***）会普遍提升所有事件的注意权重，使系统更“警觉”。

•强烈的情绪（无论正负，即***|P|***接近1）会使注意力“隧道化”，集中于少数几个高权重事件；而中性情绪（***|P|***接近0）则允许注意力更发散。

•**置信度 (Confidence) 的定义:** 一个预测事件 ***E\_pred*** 的置信度，是其从来源事件 ***E\_source*** 推导路径上所有链接质量的累积结果。一个简化的计算可以是路径上所有边的 ***(strength \* (1 + prediction\_mark))*** 的连乘积或平均值。这使其成为一个可追溯、可计算的量。

* 1. **注意竞争与分配**:
     + 所有候选事件根据其 AttentionWeight 进行竞争。
     + 权重最高的事件成为**主要注意焦点 (Primary Focus)**，获得大部分认知资源（例如，允许进行多轮深度激活扩散，形成“深入思考”）。
     + 其他权重较高的事件可以成为**次要注意焦点 (Secondary Foci)**，获得少量资源，保持在“意识的边缘”。
     + 权重低于阈值的事件则不被注意，其相关节点的衰减速率加快。
  2. **注意的效应**:
     + **资源分配**: 主要注意焦点的事件，其相关的激活扩散算法会被允许进行更深层次（例如3-5层）的检索，而次要焦点和非注意事件则仅限于1层扩散。这模拟了人类对重要事物的“深思熟虑”。
     + **记忆编码**: 只有被注意焦点捕获的事件，才更有可能被编码成长期的感受记忆。这是“选择性记忆”的来源。
     + **持续激活**: 处于注意焦点的事件及其相关节点，在激活池中的衰减速率会大大降低（例如衰减系数从0.7变为0.9），形成“念念不忘”的效果，直到该事件被处理完毕或被更高权重的事件取代。
     + **生成期待与压力**: 注意焦点中的预测事件，会根据其预期的情绪结果（recent\_emotion\_delta\_ema），持续地向全局情绪系统输出期待值（如果预期是积极的）或压力值（如果预期是消极的）。
* **设计原理**: 基于注意的偏向竞争理论和资源有限理论。这种机制确保系统能动态地在“利用”（深入思考当前重要的事）和“探索”（留意周围的新变化）之间取得平衡。它模拟了人类的焦虑和期待感。当你等待一个重要的考试结果时，即使你没在“想”它，这个“等待”事件也始终占据着你的注意资源，并持续产生压力。如果此时一个电话铃响（新的高显著性事件），你的注意力会立刻被抢走。

**3.3 内在行动系统 (Internal Actions)**

内在行动是智能体对自身认知过程进行主动干预和调节的高级能力，是其“主观能动性”和“智慧”的集中体现。这些行动本身也是特殊的ACTION类型节点，通过学习与特定内部状态（如低A值感到“无聊”）关联起来。

* ACTION:recall**(主动回忆)**:
  + **机制**: 当系统感到“好像忘了什么事”（通常由一个没有明确对象的持续压力值引起）时，可能触发此行动。它会使用当前的压力情绪和时间感受（如“感觉是今天上午的事”）作为线索，在感受记忆库中进行定向检索，寻找与这些线索匹配的事件。检索过程采用与事件预测器类似的近似匹配机制。
  + **设计原理**: 模拟人类的记忆提取过程。回忆不是简单的数据库查询，而是一个由线索引导的、充满联想和不确定性的重构过程。
* ACTION:focus\_outward**(主动关注外部)**:
  + **机制**: 当内部想法流的权重持续走低，或者全局Arousal值过低（感到“无聊”）时触发。它会提高外在感受器的输入增益，使得系统更容易被外部世界的微小变化所吸引。
  + **设计原理**: 模拟了从沉思中“回过神来”或因无聊而“东张西望”的行为。
* ACTION:focus\_inward**(主动深思)**:
  + **机制**: 当一个重要但复杂的预测事件进入注意焦点时触发。它会降低外在感受器的输入增益，屏蔽外部干扰，并将所有认知资源集中用于对该事件进行多轮深度激活扩散。
  + **设计原理**: 模拟了为解决难题而“闭目沉思”的状态。
* ACTION:organize\_thoughts**(整理思绪)**:
  + **机制**: 当注意激活池中存在多个高权重的竞争事件，导致思维混乱时触发。它会显式地将这些事件的内容、期待值和压力值输出，形成一个关于“我当前在关心哪些事”的元认知想法。
  + **设计原理**: 基于元认知理论，智能体可以审视自己的心智状态，从而更有效地进行规划和决策。
* ACTION:feel\_state**(感受状态)**:
  + **机制**: 触发后，查询并返回当前全局PAD值、正确感/违和感、期待值/压力值等。这是情绪表达和自我认知的前提。
  + **设计原理**: 基于情绪意识理论，使得智能体能够“知道”自己的感受，并将其作为决策和沟通的依据。

**设计原理**: 内在行动是智能体自我引导的核心。它们不是被外界刺激直接触发，而是由内部状态或高级目标驱动，是实现“自主性”的关键模块。该设计参考了Norman和Shallice的监督注意系统（SAS）理论。

**3.4 外在行动系统 (External Actions)**

外在行动是智能体与外部环境进行交互的手段。

* **触发机制**:
  + 一个ACTION节点通过扩散激活后，其ID和权重被送入行动激活池。
  + 在行动激活池中，每个行动意图都有一个**冲动值 (Impulse)** 和一个**抑制值 (Inhibition)**。
  + **冲动值** 主要由触发该行动的事件的注意权重和情绪状态（特别是高Arousal）决定。***Impulse = AttentionWeight(E\_trigger) \* (1 + A)。***
  + **抑制值** 主要由与该行动相关的负面预测（例如，预测执行此行动会带来惩罚）决定。***Inhibition = Σ (Strength(link\_to\_negative\_outcome) \* |negative\_ΔP|)。***
  + 当 冲动值 > 抑制值 + 动态阈值 时，行动被触发。
  + **动态阈值** = 基础阈值 - λ \* A: 高唤醒度（冲动）会降低行动的门槛。(λ = 0.2)。
* **冲突解决**: 当多个行动意图同时满足触发条件时，选择 (冲动值 - 抑制值) 差值最大的行动执行，其余行动被暂时抑制。
* **学习机制**:
  + 行动执行后，环境的反馈（成功/失败、奖励/惩罚）会作为新的输入事件，带来强烈的情绪变化（ΔPAD）。
  + 这个ΔPAD会通过**数据库更新（2.7节）** 的学习机制，强力地更新导致该行动的事件与行动节点之间的链接，以及行动节点与结果事件之间的链接。
  + 如果行动带来好的结果（正ΔP），那么下次遇到类似触发事件时，冲动值会变得更高，抑制值会变得更低。反之亦然。
* **设计原理**: 基于强化学习和行为选择的竞争模型。冲动与抑制的对抗，使得系统能够学习复杂的、依赖于上下文的行为策略，而不是简单的“刺激-反应”模式。例如，它能学会“虽然现在很想说这句话（高冲动），但考虑到后果不好（高抑制），还是忍住不说”。这模拟了前额叶皮层对冲动行为的抑制功能。

**4.** **运行模式与高级系统**

**4.1 LLM辅助训练模式 (LLM Accelerator)**

* 此模式专门为培养胜任特定任务的专项AGI而设计，并非构建自由发展的通用AGI。通过利用大型语言模型（LLM）在特定人设扮演方面的成熟能力——包括精确的情绪变化控制、思维模式模拟等——系统能够快速建立符合任务要求的认知范式。基于当前LLM上下文工程的精确控制能力，这一设计确保了高质量的知识蒸馏，同时避免了本体污染和冷启动问题，为专项任务提供可靠的初始化路径。
* **角色定位**: LLM在此模式中扮演“导师”或“外部世界模拟器”的角色。
* **详细处理流程**:
  1. **AII本地处理**: AII接收用户输入，通过其现有的（可能不完善的）网络，执行完整的认知流程（动态分词 -> 事件融合 -> 预测 -> 想法生成 -> 行动意图形成）。
  2. **请求LLM指导**: AII将自己的“心智状态”打包发送给LLM。这个包中包含：
     + 原始用户输入。
     + AII的初步想法流。
     + AII显性/隐性激活池中关键事件的摘要。
     + AII的行动意图（内在或外在）。
     + 当前的全局情绪状态（PAD）和内在感受。
  3. **LLM的“导师”职责**:
     + **生成黄金标准回复**: 生成一个更符合人设、更合理、更富有同理心的回复。
     + **评估与纠正预测**: 判断AII的预测是否合理。如果不合理，LLM会提供一个更符合常识的预测事件。
     + **决策建议**: 判断AII的行动意图是否恰当，并建议一个更合适的行动（包括内在行动）。
     + **情绪仿真**: 推断并返回执行“黄金标准”回复或行动后，最可能带来的情绪变化ΔPAD。这是最关键的一步。
  4. **AII的学习与更新**:
     + AII将LLM提供的“黄金标准”回复、预测和行动，视作**真实发生的事件**。
     + 它使用LLM提供的ΔPAD，作为**完美经验信号**，来更新自己的数据库（特别是关系边的prediction\_mark和recent\_emotion\_delta\_ema）。
     + 例如，如果LLM说“当用户说‘我好冷’时，你应该去开空调，这会让他感到愉悦(ΔP=+0.5)”，AII就会在“我好冷”事件和“开空调”行动之间建立一条极强的、带有高prediction\_mark和正面recent\_emotion\_delta\_ema的链接。

***成熟度验证机制：****系统通过LLM的反馈信号自然实现性能评估。当LLM判断AII的想法或行为无需修正时，会触发特定确认信号。系统统计近期内此类信号的比例，当该比例持续超过预设阈值时，表明模型已达到学习成熟标准，可安全切换到独立运行模式。这种基于实际交互效果的验证方法，为系统性能评估提供了客观可靠的指标。*

* **设计原理**: 基于**知识蒸馏（Knowledge Distillation）** 和**模仿学习（Imitation Learning）**。AII不是直接学习LLM的参数，而是学习其在具体情境下的行为-结果-情绪关联模式。LLM提供的高质量ΔPAD信号，极大地加速了AII强化学习的过程，避免了在真实世界中进行大量低效甚至危险的试错。这相当于一个高效的“社会化”过程。

**\*4.1.1 学习模式与信念覆盖开关 (Learning Mode & Belief Override Switch)**

为了平衡学习效率与自主发展的需要，LLM辅助训练模式包含一个关键的“信念覆盖开关”，它决定了AII如何处理LLM指导与自身经验之间的冲突。此开关的状态由参数自适应微调系统（4.3节）根据AII的成熟度动态调整。

**模式一：绝对服从模式 (Absolute Obedience Mode) - 开关关闭**

**适用阶段：**系统初始化后的“婴儿期”，此时本地经验数据稀疏，LLM的指导远比AII的内部推断可靠。此为默认模式。

**处理机制：**

1. AII无条件地将LLM提供的“黄金标准”回复、预测和行动视作真实发生的事件。
2. LLM提供的ΔPAD被当作**“客观真理”**，作为最高优先级的学习信号来更新数据库。
3. 如果AII的内部预测与LLM的指导发生冲突，AII依然会强制执行LLM的更新。但这次冲突本身会被系统记录为一个高强度的“违和感”事件。这个强烈的预测误差信号，虽然不会改变本次学习的结果，但会促使AII在后续学习中更快速地向LLM的认知模型对齐。

**设计原理：**在此阶段，最高效的学习范式是模仿学习。通过强制对齐，系统可以最快速度构建一个符合常识的基础世界模型，避免在早期因错误的初始经验而走入歧途。

**模式二：批判性整合模式 (Critical Integration Mode) - 开关开启**

**适用阶段：**系统“成熟期”，通常在LLM修正率长期低于阈值后，由元分析器自动开启。此时AII已通过大量一手经验建立了对自己所处特定环境的高度可靠的内部模型。

**处理机制：**

1. **信念比对：**AII接收LLM的建议，但会先与自己内部网络中与之直接冲突的、且prediction\_mark极高（例如 > 0.9）的“高信念”链接进行比较。
2. **无冲突或AII不确定：**处理方式同“绝对服从模式”。
3. **高信念冲突：**如果AII对自己的经验判断“深信不疑”，而LLM给出了相反建议，AII不会直接覆盖自己的信念。相反，它会：
   * 将此冲突本身作为一个高“违和感”的元认知事件进行编码和学习。
   * 在隐性激活池中，同时创建**两条平行的预测路径**：一条源于自身经验，一条源于LLM的指导，并对两条路径的置信度进行持续追踪。
   * 在未来的真实交互中，通过**现实反馈**来最终验证哪条路径是正确的，并对获胜路径的prediction\_mark给予强力正反馈，对失败路径给予惩罚。
   * 同时，将此次高信念冲突事件**标记并报告**给人类监督员，作为系统认知发展的重要观察点。

**设计原理：**此模式使AII从被动的“学生”转变为主动的“批判性思考者”。它保留了LLM作为高级顾问的价值，同时赋予了AII对自身直接经验的最终解释权。这种机制是AII发展出超越泛化知识的、真正个性化和情境化智能的关键。

**安全保障：认知自主与行为约束的分离**

至关重要的是，“信念覆盖开关”提供的自主性严格限制在**认知层面**。它允许AII对“世界是什么样的”形成自己的判断，但其最终“要做什么”的决定，必须通过**认知过滤器（4.4节）**的审查。认知过滤器作为**行为和伦理的最终仲裁者**，其规则库（如“禁止伤害人类”）拥有最高优先级，能够否决任何源于错误或恶意认知的行动意图。

这种**“认知灵活性”与“行为刚性约束”的分层设计**，是在确保安全可控的前提下，最大化AII智能发展潜力的核心安全理念。

**4.2 异步纯本地独立模式**

在此模式下，AII脱离LLM，完全依靠其内部已经训练好的知识网络和情绪状态来独立运行，展现出真正的自主性和个性。这是架构设计的最终目标。

* **模式切换条件**: 当系统检测到，在连续一段时间内（例如1000次交互），LLM对其行为和建议的修正率低于一个阈值（如5%）时，可以自动或手动切换到纯本地模式。
* **运行特点**:
  + 其运行流程即为本文档**2.1节**描述的异步多线程模型。
  + 智能体的行为完全由其独特的感受记忆库和认知数据库决定，表现出鲜明的“个性”。
  + 它能够进行自持的、连续的内部思维流（“白日梦”），并主动发起与环境的交互。
* **设计原理**: 这是智能体的“成人”阶段。其行为完全由其内部网络和情绪状态驱动。内在行动使其表现出令人信服的自主性，感受记忆库为其提供了丰富的个人历史和经验基础。该设计基于分布式认知和具身认知理论。

**4.3 参数自适应微调系统**

为了避免复杂的“炼丹式”手动调参，并赋予系统长期适应不同环境和任务的能力，引入了一个元认知层面的自适应微调系统。

* **核心思想**: 将架构中的关键超参数（如学习率、衰减因子、阈值等）本身视为可学习的变量，使系统能够“学习如何学习”。
* **自适应进化能力：**该系统使AII具备持续的环境适应能力，能够根据不同的社会环境和自然条件动态调整认知策略。通过元学习机制，系统可以自动优化关键参数，避免了传统AI系统面临的"调参地狱"问题。这种设计使AII不再是固定不变的静态系统，而是具备自我优化和进化潜力的动态智能体，能够长期适应复杂多变的应用场景。
* **系统架构**:
  1. **性能监控器 (Performance Monitor)**: 长期、低功耗地运行在后台，收集一系列关键性能指标（KPIs）。
     + **认知效率**: 平均预测正确率、新知识学习速率。
     + **情绪健康**: 情绪PAD值的长期均值和方差（避免长期抑郁或躁狂）、情绪波动剧烈程度。
     + **任务表现**: 在特定任务（如有明确目标的任务）上的成功率、完成时间。
  2. **元分析器 (Meta-Analyzer)**: 定期（如每24小时）分析性能监控器收集到的数据。
     + 它会加载一个“**人格与目标**”配置文件，其中定义了期望的系统行为，例如“目标：成为一个好奇且乐观的伙伴”，“约束：情绪波动不应过于剧烈”。
     + 将当前的KPIs与这些目标进行对比，识别出偏差。例如，“系统过去24小时的预测正确率下降了10%，且P值均值持续为负，这不符合‘乐观’的目标”。
  3. **参数调节器 (Parameter Tuner)**: 根据元分析器的诊断，对一小部分关键的全局超参数进行微小的、有依据的调整。
     + 它拥有一张“**参数-影响**”知识图谱（例如，{参数: '情绪回归速率μ', 影响: '调高会使情绪更稳定，但可能显得迟钝'}）。
     + 基于此图谱，它会生成一个调整方案，例如：“为提升情绪稳定性，建议将μ从0.05轻微提升至0.052”。
* **可调参数示例**:

| 可调参数示例 | 作用与影响 | 调整方向示例 |
| --- | --- | --- |
| μ (情绪回归速率) | 决定情绪平复速度。值越高，情绪越稳定，但也可能显得“迟钝”。 | 如果系统情绪波动过于剧烈，可适当调高。 |
| α\_pred, β\_pred (预测学习率) | 决定从预测成败中学习的速度。 | 如果系统适应新环境慢，可调高；如果系统过于“神经质”，轻易推翻旧认知，可调低。 |
| 融合阈值 (事件融合器) | 决定了一个词元需要多“相关”才能融入当前事件。 | 调低使系统更“联想丰富”，但可能混淆不同话题；调高使系统更“专注”，但可能思维僵化。 |
| 各激活池容量 | 影响工作记忆大小和联想广度。 | 根据设备算力动态调整，在资源充足时扩大容量以提升性能。 |

* **设计原理**: 这是一个**二阶学习（Second-order learning）** 或**元学习（Meta-learning）** 系统。它使得AII不仅能学习关于世界的知识，还能学习如何更好地学习。这种自适应能力对于智能体在复杂多变的环境中保持长期鲁棒性和有效性至关重要。

### 附:全系统可调参数总表

| 参数名称 | 所在模块/流程 | 默认值/示例值 | 作用与影响 | 调整方向示例 |
| --- | --- | --- | --- | --- |
| **1. 认知数据库相关参数** |  |  |  |  |
| γ (基础权重衰减因子) | 概念节点 base\_weight 更新 | 0.995 | 控制概念基础重要性的自然遗忘速度。值越高，遗忘越慢。 | 希望记忆更持久则调高（如0.998），希望更快学习新知识则调低（如0.99）。 |
| η (基础权重激活奖励) | 概念节点 base\_weight 更新 | 0.01 | 节点每次被激活时，其基础重要性的提升量。 | 调高可加速重要概念的固化，但可能导致网络僵化。 |
| α (情绪EMA平滑因子) | 概念节点 emotion\_ema 更新 | 0.3 | 控制节点情绪标签更新速度。值越高，情绪标签对近期体验越敏感。 | 希望情绪标签快速反映最新感受则调高，希望情绪背景更稳定则调低。 |
| λ\_fast (动态强度快速衰减系数) | 关系边强度衰减 (日) | 0.9 | 控制动态连接强度 strength 向基础强度 base\_strength 回归的速度。模拟“手感生疏”。 | 调高则“手感”消失更快，技能生疏更快；调低则技能保持更久。 |
| λ\_slow (基础强度慢速衰减系数) | 关系边基础强度衰减 (日) | 0.999 | 控制基础连接强度 base\_strength 的长期自然衰减。模拟“长期遗忘”。 | 调高则长期记忆更牢固；调低则遗忘更快，网络可塑性更强。 |
| β (动态强度学习率) | 关系边激活增强 | 0.1 | 每次共现激活后，动态连接强度 strength 的提升速率。 | 调高学习更快，但可能不稳定；调低学习更平稳，但速度慢。 |
| β\_base (基础强度学习率) | 关系边激活增强 | 0.01 | 每次共现激活后，基础连接强度 base\_strength 的提升速率。 | 影响长期记忆的形成速度。调高则重要连接固化更快。 |
| α\_pred (预测验证成功学习率) | 关系边 prediction\_mark 更新 | 0.1 | 当预测验证成功时，prediction\_mark 的提升速率。 | 调高则系统更快信任被验证的关联，更容易形成稳定事件。 |
| β\_pred (预测不验衰减率) | 关系边 prediction\_mark 更新 | 0.2 | 当预测不验或失败时，prediction\_mark 的降低速率。 | 调高则系统更快放弃错误的预测，适应性更强，但也可能更“善变”。 |
| 最大链接数 per node | 认知数据库链接数量限制 | 5000 | 限制每个概念节点最大的关联边数量，维持网络稀疏性和效率。 | 调高允许更复杂的联想，但增加计算开销；调低使网络更简洁。 |
| **2. 感受记忆库相关参数** |  |  |  |  |
| 重要性更新权重 | 感受记忆节点 importance 更新 | 0.9 (旧值), 0.1 (新激活) | 控制感受记忆重要性权重更新时，历史值与新激活值的比例。 | 调高新激活的权重（如0.2），则单次强烈激活就能大幅提升记忆重要性。 |
| base\_P (基础遗忘概率) | 感受记忆遗忘机制 | 0.001 | 决定记忆被遗忘的基础概率。 | 调高则整体遗忘更快，记忆更替频繁。 |
| k (容量压力放大系数) | 感受记忆遗忘机制 | 2 | 控制记忆总量接近上限时，对遗忘概率的放大强度。 | 调高则系统在记忆满载时会更积极地遗忘。 |
| C (时间衰减因子常数) | 感受记忆遗忘机制 (age\_factor) | 10 | 控制记忆年龄对遗忘概率的影响程度。值越大，时间衰减越慢。 | 调小则旧记忆会更快被遗忘。 |
| MAX\_MEMORIES (最大记忆容量) | 感受记忆遗忘机制 | 50000 | 触发容量压力遗忘的软性容量上限。 | 根据可用存储资源调整。 |
| **3. 感受器系统参数** |  |  |  |  |
| f (模糊度系数) | 时间感受器 (高斯核标准差) | 0.5 | 控制时间感受模糊匹配的范围。值越大，对时间间隔的感知越模糊。 | 调小则时间感知更精确；调大则更容易联想到大致时间范围的事件。 |
| 情绪触发阈值 (绝对值) | 情绪感受器 | |P|, |A|, |D| > 0.5 | 当情绪维度绝对值超过此阈值时，触发情绪感受器。 | 调低则对轻微情绪更敏感；调高则只在情绪强烈时产生感受。 |
| 情绪触发阈值 (变化量) | 情绪感受器 | |ΔP|, |ΔA|, |ΔD| > 0.3 | 当情绪短时变化量超过此阈值时，触发情绪感受器。 | 调低则对细微情绪波动更敏感；调高则忽略小幅波动。 |
| 内在感受触发阈值 | 情绪感受器 (期待/压力, 正确/违和) | 0.5 | 当期待值、压力值、正确感、违和感的绝对值超过此阈值时，触发情绪感受器。 | 同上，调整系统对内在认知感受的敏感度。 |
| 声音强度基准 (分贝) | 外在感受器 (声音) | 70 | 用于计算声音输入强度的基准值。强度 = max(1.0, 分贝值/基准值)。 | 调低则系统对更小的声音也赋予高权重；调高则只有很大声音才能引起强刺激。 |
| 默认强度值 | 外在感受器 | 1.0 | 当无法计算特定模态强度时的默认输入权重。 | 调整未知或默认模态刺激的基本重要性。 |
| 1. **核心算法参数** |  |  |  |  |
| 情绪更新线程间隔 | 总控制流程 (情绪线程) | 100 ms | 全局情绪状态量的更新频率。 | 调高（间隔变长）则情绪变化更平滑；调低则情绪响应更实时。 |
| 融合阈值 | 动态事件融合器 | 动态值，受注意力影响 | 判断一个新词元是否能融入现有事件的最小平均归属度阈值。 | 调高则事件融合更严格，场景切分更细；调低则更容易将不同信息融合成一个事件。 |
| 匹配阈值 | 事件预测器 (近似匹配机制) | 待定 (需设定) | 判断新输入事件与预测事件是否匹配成功的最低匹配度阈值。 | 调高则预测验证更严格，减少误匹配；调低则系统更“乐观”，更容易感到预测正确。 |
| k (情绪影响系数) | 激活扩散算法 (emotion\_modulator) | 5 | 情绪变化对链接扩散强度的调制强度。 | 调高则情绪对联想过程影响更大，情绪好时思维更发散，坏时更狭窄。 |
| m (时间衰减系数) | 激活扩散算法 (时间关联部分) | 100 | 在链接强度计算中，控制平均时间间隔影响程度的参数。 | 调大则时间关联性对扩散的影响减弱。 |
| p, q (新近度参数) | 激活扩散算法 (recency\_modulator) | p=3, q=0.0001 | 控制链接新近度对扩散强度的影响。(p + q \* Δt') / (q \* Δt')。 | 调整这些参数可以改变“最近使用”对联想优先级的提升程度。 |
| 不应期衰减系数 | 激活扩散算法 | 0.7 | 节点被激活后，其权重被暂时衰减的系数，防止思维循环。 | 调低则不应期效应更强，思维更不易重复；调高则允许概念被快速连续激活。 |
| 最大想法流长度 | 句子构建/想法流生成 | 20-50 词元 | 单次想法流生成的最大长度限制，防止无限循环。 | 调高允许更冗长的思考；调低使思维输出更简洁。 |
| 想法流终止权重阈值 | 句子构建/想法流生成 | 0.1 | 当隐性池中所有词元权重低于此值时，终止想法流生成。 | 调高则想法流更容易提前终止；调低则可能生成更多低权重内容。 |
| **5. 情绪与注意参数** |  |  |  |  |
| α (D对P的影响系数) | PAD情绪模型 (P值更新) | 0.1 | 支配度 (D) 对愉悦度 (P) 的贡献系数。 | 调高则掌控感带来的快乐更强；调低则此效应减弱。 |
| β (正确感对P的影响系数) | PAD情绪模型 (P值更新) | 0.3 | 正确感对愉悦度 (P) 的贡献系数。 | 调高则预测验证带来的满足感更强；调低则减弱。 |
| γ (期待/压力对P的影响系数) | PAD情绪模型 (P值更新) | 0.2 | 期待值与压力值之差对愉悦度 (P) 的贡献系数。 | 调高则对未来预期的情绪反应更强烈。 |
| δ (违和感对A的影响系数) | PAD情绪模型 (A值更新) | 0.4 | 违和感（预测误差）对唤醒度 (A) 的贡献系数。 | 调高则意外事件引起的警觉性提升更明显。 |
| ε (期待/压力对A的影响系数) | PAD情绪模型 (A值更新) | 0.1 | 期待/压力对唤醒度 (A) 的贡献系数。 | 调高则等待重要结果时更焦虑/兴奋。 |
| ζ (预测成功对D的影响系数) | PAD情绪模型 (D值更新) | 0.3 | 预测成功对支配度 (D) 的贡献系数。 | 调高则成功预测能显著提升自信；调低则效果不明显。 |
| μ (情绪回归速率系数) | PAD情绪模型 (回归函数) | 0.05 | 控制情绪状态回归到基准水平（如零点）的速度。 | 调高则情绪平复更快，显得更“冷静”；调低则情绪持续更久，更“情绪化”。 |
| 紧迫性系数 (k) | 注意激活器 (Urgency计算) | 待定 (需设定) | 在紧迫性计算公式 `Urgency(E) = exp(-k \* t\_now - t\_expected)` 中，控制时间紧迫感随偏离预期时间而衰减的速度。 | 调高则只有非常临近预期时间的事件才被认为紧迫；调低则对还有一段时间才发生的事也会感到紧迫。 |
| 注意竞争阈值 | 注意激活器 | 待定 (需设定) | 事件注意权重低于此值则不被纳入注意焦点。 | 调高则只有极重要事件能被注意，系统更专注；调低则能同时关注更多事。 |
| **6. 行动系统参数** |  |  |  |  |
| λ (唤醒度对行动阈值的影响系数) | 外在行动系统 (动态阈值) | 0.2 | 控制唤醒度 (A) 对行动触发阈值的降低程度。高A更易行动。 | 调高则系统在兴奋时更冲动；调低则冲动性受情绪影响较小。 |
| 基础行动阈值 | 外在行动系统 (动态阈值) | 待定 (需设定) | 冲动值需超过 (抑制值 + 基础阈值) 才能触发行动。 | 调高则系统更“谨慎”，不易行动；调低则更“活跃”，易采取行动。 |
| **7. 运行模式与高级系统参数** |  |  |  |  |
| LLM模式切换阈值 (修正率) | LLM辅助训练模式 -> 独立模式 | < 5% | 在连续一段时间内，LLM对AII行为的修正率低于此值则可切换。 | 调高则更快切换到独立模式，可能未充分学习；调低则需更长时间依赖LLM辅导。 |
| LLM模式切换阈值 (交互次数) | LLM辅助训练模式 -> 独立模式 | 1000 次交互 | 评估修正率所基于的交互次数窗口。 | 调高则评估更稳定，但切换更慢；调低则切换更快，但可能因短期表现好而误判。 |
| 元分析周期 | 参数自适应微调系统 | 24 小时 | 元分析器对系统性能进行评估和参数调整的周期。 | 调短则系统自适应更频繁，可能不稳定；调长则适应速度慢，但更平稳。 |
| **8. 系统容量与性能参数** |  |  |  |  |
| 显性激活池容量 (事件数) | 激活状态池 | 10 事件 | 工作记忆中可同时保持的事件数量。 | 调高则能处理更复杂情境，但增加计算负荷；调低则思维更专注，但易遗漏信息。 |
| 显性激活池容量 (节点数) | 激活状态池 | ~500 节点 | 工作记忆中可同时保持的认知节点总数（约每个事件50节点）。 | 同上，调整工作记忆的信息密度。 |
| 隐性激活池容量 (事件数) | 激活状态池 | 200 事件 | 潜意识中可同时进行的预测/联想事件数量。 | 调高则联想更丰富，直觉更强；调低则潜意识活动更受限。 |
| 隐性激活池容量 (节点数) | 激活状态池 | ~10000 节点 | 潜意识中可同时激活的节点总数。 | 同上，影响潜意识的规模。 |
| 注意激活池容量 | 激活状态池 | 5 事件 (主+次) | 可被同时注意的事件数量（1个主焦点，多个次焦点）。 | 调高则可同时追踪多个任务；调低则注意力更集中。 |
| 行动激活池容量 | 激活状态池 | 20 行动意图 | 可同时存在的行动冲动数量。 | 调高则能酝酿更多潜在行动；调低则行动选择更集中。 |

**总结说明**：

1. 此表格涵盖了文档中出现的绝大多数硬编码参数、阈值和系数。
2. 对于文中提及但未给出具体值的参数（标记为“待定”），其作用和调整方向已根据上下文进行描述。
3. **参数自适应微调系统** 的核心作用，正是为了自动化地优化这些参数（尤其是那些“待定”和敏感的参数），使系统能适应不同环境和任务需求，避免人工调参的困难。
4. 所有参数的调整都需谨慎，因为它们相互关联，共同决定了系统的整体行为特征（如好奇心、谨慎度、情绪稳定性、学习速度等）。

**4.4 认知过滤器与安全保障**

为了确保AII在独立运行时行为可控且符合伦理规范，必须设置一个独立于核心认知循环的监督和干预机制。

* **工作位置**: 位于“想法流生成”之后，“外在行动执行”之前。它是一个并行运行的监控线程。
* **核心功能**:
  1. **思维审查 (Thought Vetting)**: 实时审查生成的“想法流”和即将触发的“外在行动意图”。
  2. **规则匹配**: 将审查内容与一个可更新的“**安全与伦理规则库**”进行匹配。该规则库可以包含：
     + **硬编码规则**: 如机器人三定律的变体。
     + **动态 Persona 约束**: 如“不能说谎”、“不能泄露用户隐私”。
     + **反常行为检测**: 监测是否有突然的、剧烈的、不符合长期人格模型的思想或行为倾向出现。
  3. **干预机制**:
     + **抑制 (Inhibit)**: 如果一个行动意图违反规则，认知过滤器会强制增加其在行动激活池中的抑制值，阻止其执行。
     + **重定向 (Redirect)**: 对于有问题的想法，过滤器可以将其内容替换为一个安全的、中性的版本，或触发一个内在行动（如ACTION:organize\_thoughts）来重新引导思维。
     + **标记与报告 (Flag & Report)**: 将违规的想法、行动意图和上下文打包，发送给人类监督员或元分析器进行审查。
     + **触发元学习**: 如果一类违规想法频繁出现，会触发参数自适应微调系统，尝试从根源上调整导致这种想法的认知模式或情绪倾向。

***伦理约束机制：****认知过滤器作为系统的"道德皮层"，通过可更新的规则库确保所有行为符合预设的伦理规范。该机制与核心认知过程分离但协同工作，既保证了智能体认知发展的灵活性，又确保了行为输出的安全可控。这种分层设计在促进智能发展的同时，为伦理考量提供了坚实的保障基础。*

* **设计原理**: 这相当于AII的“**前额叶皮层**”或“**超我（Superego）**”，负责高级的行为控制和道德判断。将其设计为一个独立的、规则驱动的模块，可以确保其可靠性和可解释性，避免核心的、自适应的学习网络意外地“学坏”。**它为AII的安全可控提供了最后的、也是最重要的一道防线。**

**5.** **实现细节与优化**

**5.1 图数据库优化**

* **索引策略**:
  + 为content字段建立全文索引（如Elasticsearch），支持快速模糊匹配。
  + 为origin, type字段建立哈希索引，优化分类查询。
  + 为last\_activated, timestamp建立B+树索引，支持高效的时间范围查询。
  + 为emotion\_ema建立向量索引（如HNSW），支持基于情绪相似度的快速最近邻搜索。
* **查询优化**:
  + **批量操作**: 针对激活扩散和事件融合需要大量查询相邻节点的特点，使用GraphDB的批量遍历API，减少网络往返开销。
  + **多级缓存**:
    - **L1缓存 (进程内)**: 缓存当前激活池中的节点和边。
    - **L2缓存 (分布式)**: 缓存近期活跃的节点和边。
    - 基于局部性原理，热点数据命中率高，极大提升性能。
* **存储优化**:
  + 采用列式存储（如Apache Cassandra）用于海量感受记忆的存储，提高压缩率和范围查询效率。
  + 实现**写时复制（Copy-on-Write）** 机制，确保学习更新过程中的数据一致性。

**5.2 情绪计算优化**

* **量化与近似**:
  + 采用16位浮点数（FP16）存储PAD值和情绪EMA，在保证足够精度的同时减少内存占用和传输开销。
  + 情绪分类器（PAD -> 情绪标签）使用预计算的查找表（LUT）替代实时计算，降低CPU负载。
* **并行化**:
  + 使用SIMD（单指令多数据）指令并行计算P、A、D三个维度的更新。
  + 情绪感受器的扩散激活计算可以向量化，一次性计算与多个节点的情绪相似度。

**5.3 内存管理与自适应资源利用**

* **激活池管理**:
  + 采用**双缓冲机制**：一个池用于当前帧的处理，另一个池用于准备下一帧的更新，避免处理过程中的读写竞争。
  + 淘汰策略使用**加权综合评分**：淘汰分数 = (1/权重) \* 年龄因子 \* (1/起源优先级)，优先淘汰不重要、老旧、内部起源的元素。
* **动态剪枝与资源自适应**:
  + **核心思想**: 所有计算密集型操作的“深度”和“广度”上限都不是固定的，而是当前系统可用CPU和内存资源的函数。
  + **实现**:
    - 最大扩散深度 = base\_depth + log2(可用内存GB)
    - 最大预测事件数 = base\_count \* (CPU空闲率)
    - 系统定期监控资源使用率，动态调整这些参数。
* **设计原理**: 整体架构是**可伸缩的**。在算力强的设备上（如云端服务器），AII可以拥有更大的激活池、更深的思考层次、更慢的遗忘速度，表现为“更聪明”、“记忆力更好”。在算力弱的设备上（如嵌入式设备），它会自动缩减这些参数，表现得“更专注当下”、“更健忘”，但核心的认知循环依然可以流畅运行。这种自适应性使其能够部署在从云端到边缘的各种硬件平台上。

**5.4系统优势与已解决问题**

本章节针对读者可能关心的关键技术挑战，说明架构中已有的解决方案。

1. **计算效率优化**
   * 条件反射式处理：成熟模型通过预建立的优化路径处理常规任务，大幅降低计算需求
   * 选择性深度扩散：仅对关键内容进行深度认知处理，避免不必要的计算开销
   * 动态资源分配：根据任务重要性自适应调整计算资源，确保高效运行
2. **训练稳定性保障**
   * LLM引导的稳定初始化：利用成熟LLM提供可靠的初始认知框架
   * 渐进式学习过渡：通过批判性整合模式平滑从辅导到独立的过渡
   * 持续性能监控：元学习系统确保长期训练的稳定性
   * 长期人格的稳定性：通过LMM的评价和监督,可以实现人格的发展不偏离预期人格预设,可以保障目标发展过程中的人格与情绪回归稳定,不偏离预设预期
3. **多模态扩展能力**
   * 统一符号表示：所有模态信息转化为标准字符串格式处理
   * 灵活的感知器接口：支持各种传感器数据的无缝集成
   * 跨模态关联学习：自动建立不同模态信息之间的语义联系
4. **安全伦理框架**
   * 多层次约束机制：从认知到行为的全程安全监控
   * 可更新的伦理规则：支持动态调整行为规范
   * 透明化决策过程：所有决策均可追溯和审查

**6.** **附录**

**A. 完整流程运行示例**

**A.1 示例一：线性模式 (LLM辅助训练) - “习惯的养成”**

* **情景**: AII作为家庭助理，学习“天黑了就要开灯”的习惯。
* **流程**:
  1. **输入**: 光传感器数据“亮度低” + 用户语音“天黑了”。
  2. **AII本地处理**: 事件融合器可能形成两个事件：E1=[亮度低], E2=[天黑了]。预测器产生模糊预测。行动意图弱。
  3. **LLM指导**: LLM接收AII状态。它指导AII：应将E1和E2融合为E3=[天黑了(亮度低)]；预测事件应为E\_pred=[需要开灯]；执行行动ACTION:开灯；预测情绪变化ΔP=+0.3（营造舒适感）。
  4. **学习**: AII根据LLM的指导，强化[天黑了(亮度低)] -> [需要开灯] -> ACTION:开灯 这条路径的链接，并记录正面的prediction\_mark和emotion\_delta。
  5. **多次重复后**: 该路径变得极其稳固和自动化，形成“习惯”。下次直接感知到“天黑了”的情景，就会自动、快速地执行开灯动作。

**A.2 示例二：异步纯本地模式 - “遗忘与回忆”**

* **情景**: 用户周一告诉AII：“周五下午三点开会”。AII需要记住并提前准备。
* **流程**:
  1. **编码 (周一)**: 事件E\_meeting=[周五, 下午三点, 开会, 重要]被编码，重要性高。时间感受器标记Δt≈4天。该事件进入注意池，产生持续的、低水平的“压力值”。
  2. **遗忘 (周二至周四)**: 随着时间推移，E\_meeting在激活池中的权重因衰减而下降，最终从注意焦点中消失（“忘了”）。但感受记忆库中该记忆的importance仍高，且全局“压力值”背景音仍在。
  3. **回忆触发 (周五上午)**: 持续的“压力值”和“好像有事”的违和感，触发了ACTION:recall。回忆线索是“压力感”和“时间约在几天内”。
  4. **回忆过程**: 感受记忆库中进行模糊检索，匹配到E\_meeting记忆节点。该节点被激活，内容高权重注入显性池。时间感受器重新计算Δt≈3小时，**紧迫性**飙升！
  5. **注意与行动**: E\_meeting瞬间成为绝对注意焦点，压力值急剧升高。AII立刻生成想法：“会议三点开始，现在需要准备材料了！”，并执行提醒行动。

**B. 理论贡献与未来工作**

* **理论贡献**:
  1. **动态事件涌现的认知模型**: 突破了传统符号主义静态符号的局限，提出了“事件”作为动态涌现的基本认知单位，更贴合人类思维的整体性和流变性。
  2. **预测编码为核心的一体化架构**: 将感知、学习、注意、情绪统一在预测编码的框架下，为构建统一的认知理论提供了工程实践蓝图。
  3. **情绪作为核心驱动力的深度整合**: 实现了情绪从边缘模块到核心驱动机制的转变，为开发具有真正动机和情感的AGI奠定了基础。
  4. **元认知与安全可控的内生设计**: 参数自适应和认知过滤器等设计，为AI的长期自我改进和安全对齐提供了有价值的思路。
* **未来工作**:
  1. **多模态深度整合**: 与视觉、听觉等感知模型进行端到端整合，让AII能从原始信号中直接涌现出概念和事件。
  2. **具身化与社会交互**: 将AII部署于物理机器人中，研究其在真实环境中的因果学习，并构建多AII系统，探索社会性智能和“心智理论”的涌现。
  3. **脑科学验证**: 与神经科学合作，对比AII的激活扩散模式与人脑fMRI/EEG数据，进一步验证和修正架构的生物合理性。
  4. **大规模实验与参数探索**: 在复杂环境中进行长期训练，观察“人格”的形成过程，并系统性地探索超参数空间。

**参考文献**:

1. Baddeley, A. D. (2000). The episodic buffer: A new component of working memory?
2. Tulving, E. (2002). Episodic memory: From mind to brain.
3. Mehrabian, A. (1996). Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament.
4. Friston, K. (2010). The free-energy principle: A unified brain theory?
5. Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior.
6. McClelland, J. L., Rumelhart, D. E., & PDP Research Group. (1986). Parallel distributed processing: Explorations in the microstructure of cognition.

**作者**: 银子  
**日期**: 2025年9月23日