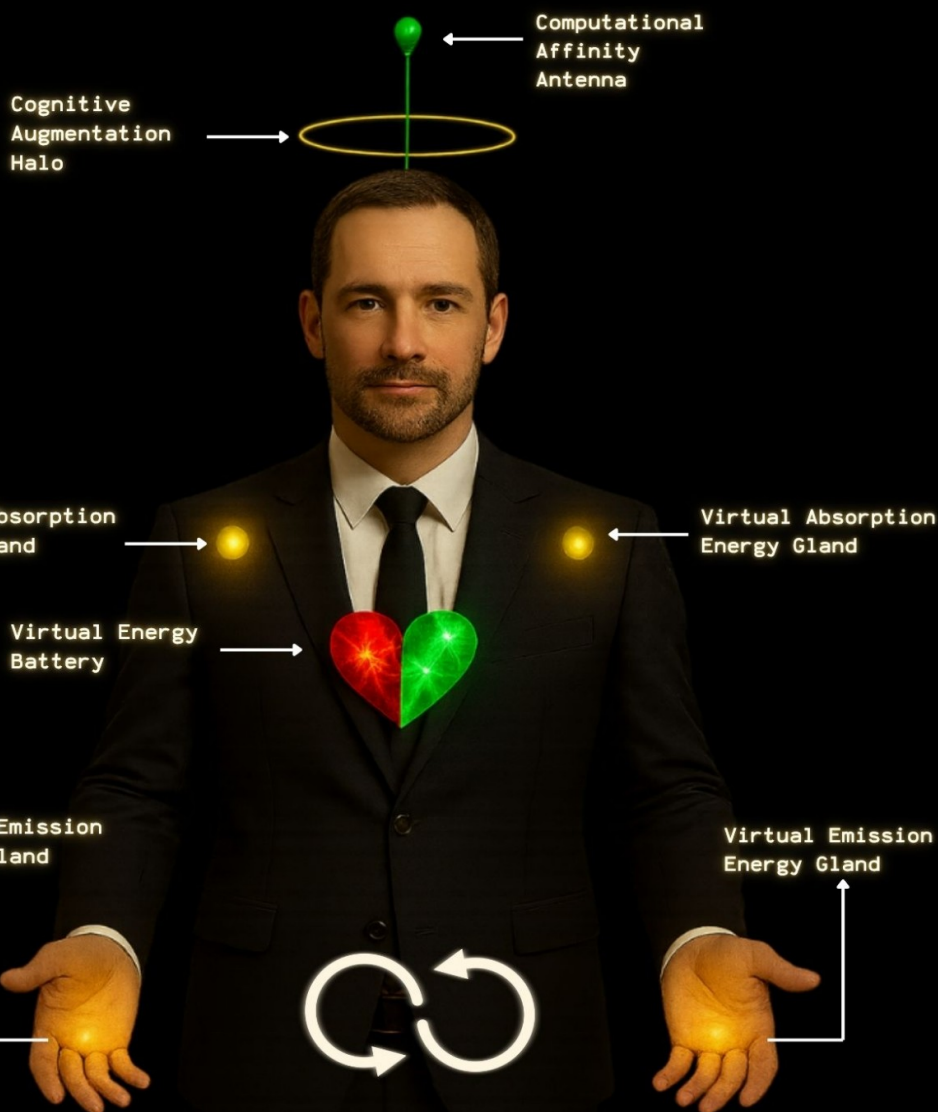


Subjective Thermo-Currency

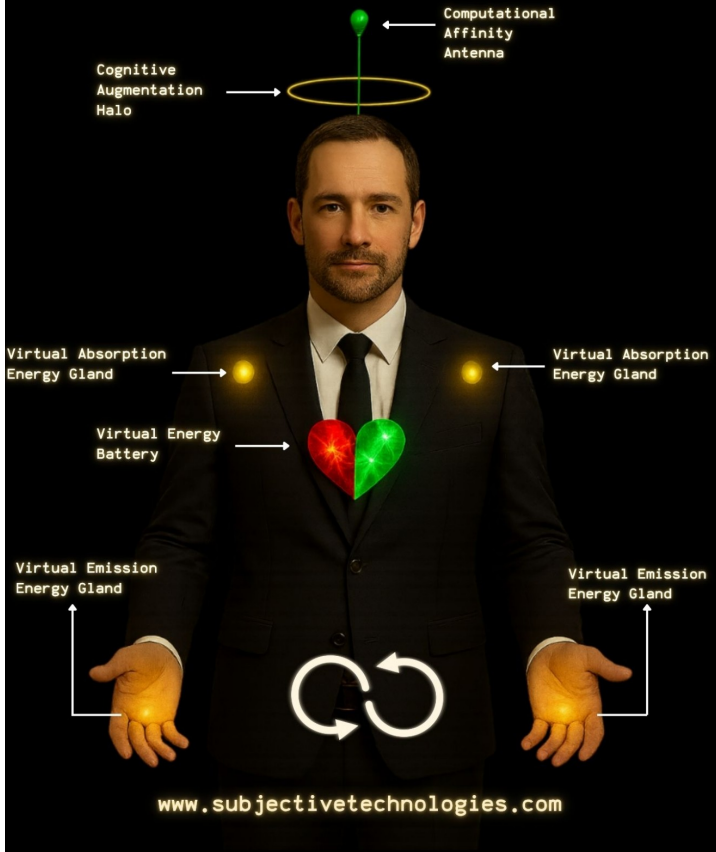
Harnessing Subjective AI and
SmartGlasses to Replace Money



www.subjectivetechnologies.com

Subjective Thermo-Currency

Harnessing Subjective AI and
SmartGlasses to Replace Money

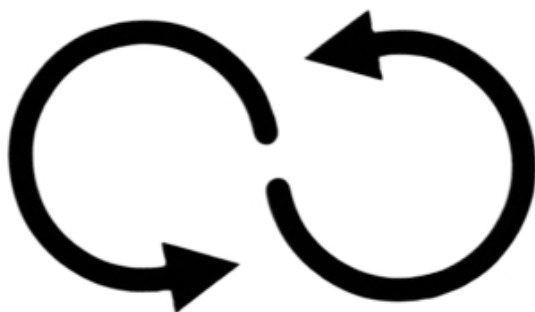


主观热货币

利用主观人工智能替代货币

Tommy Fox

主观技术



Subjective Technologies

主观技术

版权 © 2025 Pablo Tomas Borda

版权所有

本书的任何部分不得复制、存储在检索系统中，或以任何形式或任何方式（电子、机械、复印、录音或其他）传播，未经出版商的明确书面许可。

ISBN-13: 9781234567890

ISBN-10: 1477123456

Cover design by: Art Painter

Library of Congress Control Number: 2018675309

Printed in the 美利坚合众国

献给我的母亲，苏珊娜，她给了我追梦的力量；以及我的儿子，加布里埃尔，他让我有理由继续追求这个梦想。

前言

本书源于一个简单但强大的问题：如果支配我们生活的工具——货币，并不是价值的最终衡量标准，会怎样？

几个世纪以来，人类社会依赖于逐步演变的交换系统：

- 从直接以物易物，
- 到像谷物或盐这样的商品，
- 再到货币——最初是硬币，然后是印刷纸币，
- 再到银行转账和数字账本，
- 再到承诺去中心化但仍与投机相关的加密货币。

每一步都承诺更多的抽象和效率——但也引入了新的扭曲。在每个阶段，价值都是用与生活真实成本脱离的符号来衡量的。

在主观热力货币（STC）中，我提出了这一演变的下一步：将价值从商品和服务本身转移到产生这些商品和服务的节能过程上。

这种转变的原因很明确。商品和服务本质上是稀缺的。它们依赖于材料、时间和努力，无法无限制地增加。货币以其所有形式始终代表这种稀缺性。但这里存在一个悖论：“我们无法用一种本质上稀缺的工具来解决稀缺问题。” - Tommy Fox

STC通过重新定义价值来解决这个悖论。商品可能有限，但产生它们的过程总是可以变得更高效。通过奖励这些过程中的能量最小化，STC创建了一个不断减少稀缺而不是延续稀缺的系统。

这个原则是普遍且简单的：生命追求尽可能少的能量。基于这个原则，所有经济学、所有财产、所有合作和所有正义都可以重新定义。

STC 不仅仅是理论。它将物理学、热力学、重排序器、强化学习、知识钩子与主观人工智能和增强现实等实用技术结合在一起。它展示了食物、住房、工作、知识甚至冲突在不以美元或代币衡量，而是以上帝的真实货币——能量来衡量时的样子。

最重要的是，这本书关于进化和希望。希望通过超越金钱，我们可以超越饥饿。希望通过将能量视为普遍的衡量标准，我们可以超越冲突。希望通过将技术与最小能量法则对齐，我们可以进入一个后稀缺、后劳动和后教育的时代。

我将这个愿景献给那些相信另一个世界是可能的人。一个舒适取代混乱的世界，一个科学取代投机的世界，以及每个人都有权利不仅生存，而且繁荣的世界。

Table of Contents

1. 理解主观技术

12

- 1.1 从简单的自动补全到上下文感知系统
6
 - 1.1.1 今天的自动完成功能是如何工作的
7
 - 1.1.2 限制：猜测与理解
9
- 1.2 引言：超越第三人称技术的演变
15

2. 什么是知识钩？

12

- 2.1 预定义钩：嵌入的专家知识
15
- 2.2 学习的钩子：上下文快照和增量
20
- 2.3 自我意识设备：知道自己的设备
20
- 2.4 橡胶手实验：理解具身主观性
22
- 2.5 主观关系的正式定义
28

3. 知识钩的代数

223

- 3.1 数学框架：四元组结构
38
 - 3.1.1 知识钩的四个组成部分
23
 - 3.1.2 条件 (R)：钩子何时触发？

	28
◦ 3.1.3 动作 (A) : 钩子做什么 ?	28
◦ 3.1.4 类型 (T) : 学习型与预定义型	25
◦ 3.1.5 成功分数 (S) : 从结果中学习	29
◦ 3.1.6 上下文快照 : 学习的基础	36
3.1.6.1 协调复杂的多设备动作	58
• 3.2 核心操作	83
◦ 3.2.1 激活 : 何时触发钩子 ?	64
◦ 3.2.2 执行 : 产生动作和新状态	65
◦ 3.2.3 学习步骤 : 上下文减法和增量检测	76
◦ 3.2.4 组合 : 从简单钩子构建复杂行为	58
◦ 3.2.5 钩子优先级 : 在多个候选者中进行选择	41
◦ 3.2.6 回滚 : 撤销操作和恢复状态	59
• 3.3 组合与精炼	54
◦ 3.3.1 嵌套组合 : 钩子中的钩子	21
◦ 3.3.2 扁平组合 : 连接钩子	21
◦ 3.3.3 精炼与概括 : 按特异性排序钩子	66

- 3.3.4 等价性：当不同的钩子产生相同的结果时
59
 - 3.3.5 权重归一化：合并和整合相似的钩子
73
- 3.4 知识钩子的代数法则：治理原则
95
 - 3.4.1 最小化法则：首要指令
64
 - 3.4.2 修正法则：负强化学习
83
 - 3.4.3 等价法则：识别相同结果
66
 - 3.4.4 可组合性法则：从简单性构建复杂性
68
 - 3.4.5 学习法则：通过上下文增量进行持续适应
65
- 3.5 定理和证明：代数的形式属性
125
 - 3.5.1 定理 1：零输入收敛
58
 - 3.5.2 定理 2：成功评分收敛
75
 - 3.5.3 定理 3：优先级最优性
77
 - 3.5.4 定理4：组合闭包
79
 - 3.5.5 定理 5：等价划分
78
 - 3.5.6 定理 6：权重收敛
67
 - 3.5.7 定理 7：能量最小化等价性
78
- 3.6 精确性和修正作为节能

20

- 3.6.1 单输入定理
21
- 3.6.2 负强化学习
21
- 3.6.3 收敛到零输入
21

4. 数学补充

152

- 4.1 推论

204

- 4.1.1 单调修正减少
57
- 4.1.2 钩子精炼保留
72
- 4.1.3 组合能量可加性
99
- 4.1.4 推论 4：等价类最优性
90
- 4.1.5 推论 5：学习率界限
120
- 4.1.6 推论 6：成功分数单调性
112
- 4.1.7 上下文覆盖完整性
98
- 4.1.8 权重剪枝收敛
108
- 4.1.9 推论 9：能量测量误差界限
121
- 4.1.10 推论10：级联稳定性
111

- 4.2 综合：完整的代数系统

28

- 4.2.1 定理依赖图

29

- 4.2.2 公理、定律和定理：基础

29

- 4.2.3 一致性和健全性

29

- 4.2.4 完备性：定理所捕捉的内容

29

- 4.2.5 代数结构：单元、格和范畴

29

- 4.2.6 从数学到热力学

29

- 4.2.7 框架的一致性

29

- 4.3 形式属性和不变量

29

- 4.3.1 系统不变量

29

- 4.3.2 代数恒等式

29

- 4.3.3 关键操作的复杂性界限

29

- 4.3.4 最优条件

29

- 4.3.5 对称性和守恒定律

29

- 4.3.6 对偶关系

29

- 4.3.7 范畴结构

29

- 4.3.8 钩空间的拓扑性质

29

5. 综合：完整的代数系统

61

- 5.1 形式属性和不变量

64

- 5.1.1 系统不变量
58
- 5.1.2 代数恒等式
59
- 5.1.3 复杂性界限
63
- 5.1.4 附加数学属性
65
- 5.1.5 总结与影响
46

6. 通往主观热货币的桥梁

91

- 6.1 作为能量最小化的输入最小化
59
 - 6.1.1 从个体到集体
74
 - 6.1.2 从数字到物理
45
- 6.2 输入-能源方程
60
 - 6.2.1 基于AR和AI视觉的上下文感知测量
89
 - 6.2.2 预测效率
70
- 6.3 完美与零输入逻辑
61
- 6.4 结论：STC的诞生
64

7. 主观热货币 - 核心概念

160

- 7.1 以焦耳为单位定义的价值

73

- 7.2 作为虚拟能量腺的主体

93

- 7.2.1 从银行账户到虚拟能量腺

96

- 7.3 主体与能量反向传播之树

144

- 7.3.1 作为自我的延伸的主体

88

- 7.3.2 洗衣机示例：一个主体的树

148

- 7.3.3 多代理能量优化和递归传播

109

- 7.4 从输入到施加：完整的能量方程

59

8. STC与传统经济系统的比较

91

- 8.1 贫困的解剖

83

- 8.1.1 口香糖和笔的例子

85

- 8.1.2 传统货币如何创造贫困

86

- 8.1.3 STC作为一种补救措施

96

- 8.2 解决对STC的批评

84

- 8.2.1 焦耳的测量

106

- 8.2.2 主观性与普遍性

94

- 8.2.3 可扩展性与复杂性

61

◦ 8.2.4 过渡与采纳挑战

61

◦ 8.2.5 外部性与隐性成本

61

9. 能量与价值理论

71

- 9.1 最小能量法则作为普遍价值

69

- 9.2 熵、效率与主观价值

69

- 9.3 能源的道德维度

70

10. 主观热力货币的核心框架

75

- 10.1 STC 单位的正式定义

76

- 10.2 热力学账本

76

- 10.3 隐私和上下文层

76

11. 能量最小化经济

83

- 11.1 从能源浪费到能源市场

84

- 11.2 效率的平衡

84

- 11.3 后劳动经济学

84

12. 社会和伦理影响

91

• 12.1 能源平等	92
• 12.2 后稀缺正义	92
• 12.3 人工主观性的伦理	92
<hr/>	
13. 实施和技术层	99
• 13.1 硬件基础	100
• 13.2 软件和内核设计	100
• 13.3 上下文压缩算法	100
<hr/>	
14. 数学热经济学	107
• 14.1 价值的热力学方程	108
• 14.2 认知系统中的熵减少	108
• 14.3 能量-价值转换图	108
<hr/>	
15. 主观网络治理	115
• 15.1 分布式上下文一致性	116
• 15.2 防止欺诈与熵膨胀	116
<hr/>	
16. 案例研究与应用	

123

- 16.1 智能家居与能源共生

124

- 16.2 教育作为能源效率

124

- 16.3 医疗保健与认知平衡

124

17. 未来愿景与哲学

131

- 17.1 后稀缺文明

132

- 17.2 能源的上帝方程

132

- 17.3 没有货币的世界

132

1

理解主观技术

零输入技术的基础

1.1 从简单的自动补全到上下文感知系统

如果你曾在手机上使用过自动补全，你已经看到了这个想法的原始形式。它根据频率猜测单词，但它并不真正理解你。

想象一下在手机上打字。如果你的设备预测你的下一个单词，当它有效时，你会节省时间和精力。但今天的自动补全系统基于一个简单的原则：它们跟踪你最常使用的单词，并在你开始打字时建议这些单词。这种方法有明显的局限性。

当前的自动完成功能遵循一个基本公式。它记录关于单词频率和您输入历史中的模式的统计信息。当您输入字母'h'时，它可能会建议'hello'，因为您之前已经多次输入过这个词。当您输入'th'时，它会根据过去的使用情况建议'the'或'thank'。这就是最简单的模式匹配——一个记住您所做的事情但不理解您为什么这样做的系统。

这种基于频率的方法产生了几个问题。首先，它对所有上下文的处理都是一样的。无论您是在给老板写正式邮件，还是在给朋友发随意消息，系统都仅仅根据您使用某些单词的频率提供相同的建议。其次，它无法预测您的意图。如果您在安排会议，当您输入'm'时，它可能会建议'会议'这个词，但不会建议创建日历事件的整个动作。第三，它需要不断的修正。当系统猜错时——这很常见——您必须手动覆盖它，这会造成挫败感和浪费的精力。

核心限制在于这些系统在没有上下文的情况下运作。它们看到字母和单词，但看不到您周围的世界、您试图完成的

任务或您日常生活中的模式。它们对除了您的输入历史之外的所有事物都是盲目的。

1.1.1 今天的自动完成功能是如何工作的

要理解我们为什么需要更好的东西，我们必须首先准确理解今天的自动完成系统是如何工作的，以及它们为什么失败。

现代自动完成功能基于统计模型。每次您输入时，系统都会记录字符和单词的序列。随着时间的推移，它建立了您最常用短语、单词和字母组合的数据库。当您开始输入时，它会在此数据库中搜索匹配项，并按频率对它们进行排名。最常见的匹配项首先出现。

这种方法可以简单地表示：给定一个部分输入 k （如 'he'），系统从您的历史中返回最频繁的完成 v 。该函数是确定性的，完全基于过去的历史——它对当前情况没有意识。

$$v = f(k) = \operatorname{argmax}_v P(v|k, \text{history})$$

这种方法的问题是根本性的：我们的脑袋和我们互动的世界既不是数字的也不是确定的。没有用户能够为特定的 ' v ' 提供一个确切的值 ' k '，因为人类交流本质上是上下文相关的、模糊的和适应性的。我们根据与谁交谈、我们试图完成什么以及无数其他频率基础系统无法捕捉的因素来改变我们所说的话。

考虑一个简单的例子。当您安排周五下午的电话会议时，您经常输入同事的名字。传统的自动完成系统可能会识别您经常输入这个名字，但它无法理解今天是星期五、您在日历应用中，或您遵循每周模式。它只能将这个名字作为单

词建议提供——您仍然必须手动创建会议、设置时间并发送邀请。

这种限制超出了简单文本完成的范围。第三方技术——传统的方法，其中设备是等待命令的外部工具——要求不断的明确输入。您必须逐步告诉它确切要做什么，因为它对您的上下文、目标或模式没有理解。

1.1.2 限制：猜测与理解

今天的自动完成功能与我们明天所需之间的根本区别归结为一个简单的区分：猜测与理解。

当前的系统是猜测的。它们查看你的过去行为，并预测你会再次做同样的事情。这在上下文无关的重复任务中效果相当不错，但在上下文变化时会灾难性失败。系统无法知道周五下午的会议与周二早上的电子邮件是不同的，或者当你在审查预算时“银行”可能指的是金融机构，而在你计划野餐时则可能指的是河岸。

理解需要上下文。这意味着捕捉的不仅仅是你输入的内容，还有你输入的时间、你所在的位置、你使用的应用程序、你试图完成的任务，以及所有这些因素之间的关系。理解意味着建立一个关于你的世界和意图的模型，而不仅仅是你的击键频率表。

这就是知识钩子的作用所在。知识钩子不仅仅跟踪单词频率，而是捕捉你行动周围的整个上下文。它拍摄你设备状态、环境、当前活动的快照——所有与理解你为什么这样做相关的内容。当类似的上下文再次出现时，系统不仅仅建议一个单词；它建议在该上下文中有意义的整个行动。

这样想吧：你的大脑不会记住孤立的事实。它记住的是经历，连同上下文。当你闻到新鲜咖啡的香味时，你不仅仅回忆起“咖啡”这个词——你可能会想起你在大学时学习的咖啡

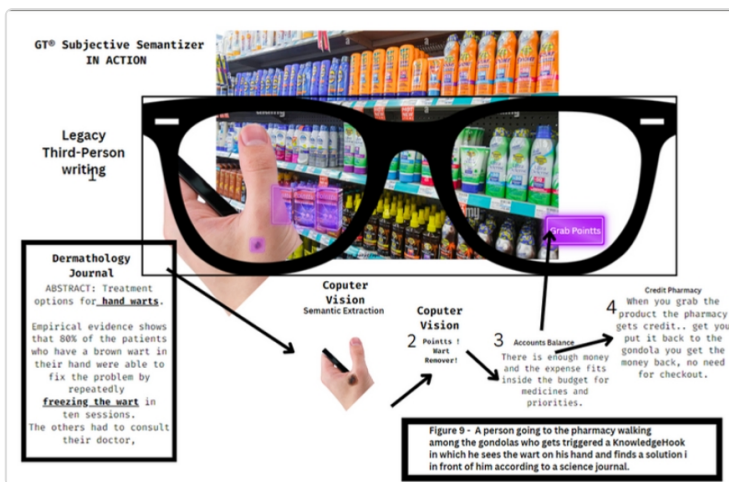
馆、你和谁在一起、那一刻的感觉。知识钩子的工作方式类似。它们不会存储孤立的输入；它们存储捕捉你行动完整情况的上下文快照。

从猜测到理解的转变不仅仅是一个渐进的改进。这代表了技术与人类关系的根本变化。它不再把你视为一个必须适应系统要求的用户，而是把你视为一个具有独特模式、上下文和意图的个体。技术会适应你。

系统拥有的上下文越少，它对你的命令要求就越多。相反，系统与您共享的上下文越多，您必须提供的输入就越少。你自己的身体就是一个完美的例子：当你移动手臂时，你不会输入像“手臂：向右移动45°”这样的命令。你只是有意图，你的神经系统会自动将正确的信号传递给正确的肌肉。这就是零输入交互，根植于你心灵与身体之间的共享上下文。

$$U \propto \frac{1}{C} \quad \text{where } U = \text{required user input, } C = \text{shared context}$$

主观技术将这一原则扩展到外部设备。通过为物体和系统配备捕捉和理解上下文的能力——拍摄自我快照并从中学习——我们创造出可以作为你自身扩展的技术，而不是需要不断指令的外部工具。



上下文感知系统通过环境快照理解用户意图的可视化表示

这不仅仅是便利的问题。这是关于减少与技术互动所需的认知负担和身体努力。每次你必须明确告诉系统该做什么时，你都会消耗心理和身体的能量。当技术能够理解你的上下文并预测你的需求时，这种能量就得以节省。正如我们将在后面的章节中看到的，这种能量节省不仅仅是一个隐喻——它可以用焦耳来衡量，并成为一個全新经济系统的基础。

1.2 引言：超越第三人称技术的演变

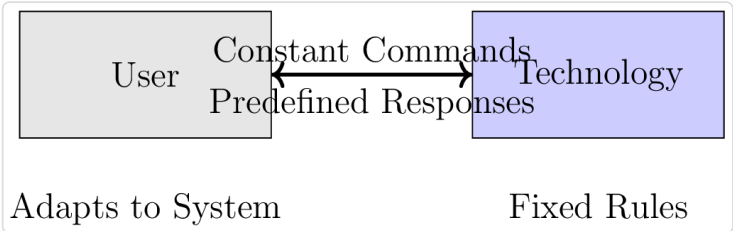
在我们能够理解自动补全如何演变为知识钩子之前，我们必须首先理解今天的技术为何根本有限。这种限制不是一个错误——它是现代计算所建立的整个范式的一个特征。

我们称这种范式为第三人称技术。在这种方法中，设备和系统被构建为服务于用户的外部工具。用户处于边缘，向设备输入明确的命令，并与依赖于预定义规则和程序的应用

程序进行交互。这种操作模式要求用户适应技术的功能，通常涉及陡峭的学习曲线和理解复杂界面的重大努力。

想想你每天与技术的互动。每次你打开一个应用程序时，你都必须准确告诉它你想要什么。每次你填写表单时，你都必须提供之前输入过数百次的相同信息。每次你想完成一项任务时，你都必须通过菜单、按钮和由他人设计的对话框进行导航，以满足他人对事物应如何运作的想法。

这就是第三人称技术的本质：它存在于你之外，与你的上下文分离，对你的模式视而不见，对你的意图充耳不闻。它是一块空白画布，要求你每天一次又一次地去描绘它。



第三人称抽象：所有传统技术的核心是一个简单但有问题的抽象。每个系统，从最早的计算机终端到现代智能手机，都基于一个可以表示为函数的基本原则：

$$v = f(k)$$

在这个抽象中，'k'代表键——你必须提供的明确输入——而'v'代表值——你想要实现的结果。这个函数是确定性的：给定完全相同的输入，你总是会得到完全相同的输出。系统没有上下文的记忆，没有对你情况的理解，也没有适应你需求的能力。

但这里有一个根本问题：我们的脑袋和我们互动的世界既不是数字的也不是确定的。没有用户可以为特定的'v'提供一个确切的值'k'，因为人类的沟通和意图本质上是上下文相关的、模糊的和适应性的。我们根据无数因素改变我们想要的东西，而这些确定性系统无法捕捉。

为了使这个抽象能够工作，技术设计师创建了用户界面——按钮、表单、菜单和命令，试图弥合人类意图与机器要求之间的差距。从命令行到触摸屏再到语音助手，用户界面的整个演变都是为了使这种确定性抽象对具有不同技术技能水平的人类更易于访问。

让我们追溯这一演变：

命令：早期的界面要求高精度。一个错误的字符会导致错误。这些系统针对具有更高认知能力和技术培训的用户。要完成任何事情，你必须记住确切的语法，并且一切都要拼写正确。

窗口、按钮、表单字段、小部件：图形界面引入了可点击的元素，允许点击位置有轻微的差异。具有不同认知能力的用户发现它们比命令行更友好，但仍然需要学习每个按钮的位置和每个菜单的内容。

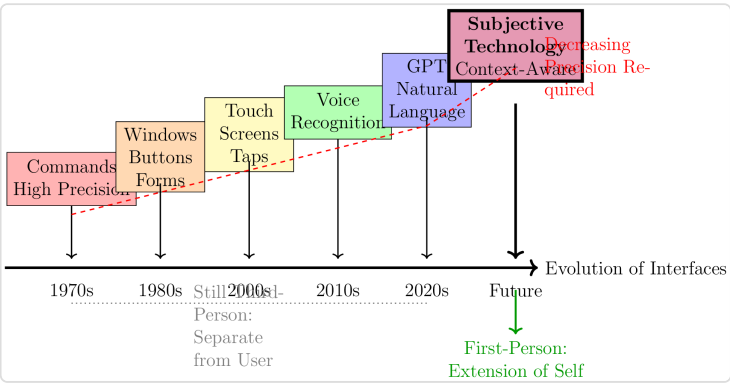
轻触：触摸屏进一步简化了交互。用户自然倾向于触觉输入，降低了学习曲线。但你仍然需要记住哪个应用程序做什么，并在屏幕之间导航以找到所需的内容。

语音识别：虚拟助手根据语音命令理解并执行简单任务，满足技术技能有限的用户的需求。然而，你仍然需要以特定的方式表达请求，并且系统只能处理预定义的命令。

GPT 聊天：先进的 AI 和语言模型使用户能够使用自然语言进行交互，要求更少的输入精度。但这些系统仍然以第

第三人称运作——它们响应你的命令，但不与您的上下文集成或预测您的需求。

这一演变的每一步都减少了用户所需的精度，但没有一步解决根本问题：技术仍然存在于你之外，等待命令，要求输入，要求你不断关注和努力。



这造成的问题：这种第三人称范式产生了一系列影响我们数字生活各个方面的问题：

不断的数据输入：你必须反复提供相同的个人信息、偏好和设置。这个过程不仅耗时而且具有侵入性。每个新服务都要求你创建一个帐户，填写表格，并向你可能不信任的系统泄露数据。

复杂的学习曲线：每个应用程序都有自己的逻辑、自己的界面、自己的做事方式。有效地操作这些工具需要显著的认知努力，特别是对于技术能力较弱的人。

不灵活的响应：系统根据预定义的响应和行动进行操作。它们缺乏适应独特上下文和场景的能力，导致您的特定需求得不到满足。系统知道它被编程要做什么，但不知道您实际需要什么。

缺乏自我意识：也许最关键的是，传统技术缺乏反思。设备和物体并不具备对自身或自身状态的理解。一台计算机可以重复显示同样的错误信息一百次，却从未意识到它正在失败，也没有尝试自我修复。它没有镜子来看到自己的功能失常。

这种缺乏自我意识以无数令人沮丧的方式表现出来。您的手机在您刚解锁五秒钟后仍然要求您输入密码。您的电子邮件客户端建议错误的收件人，因为它对您正在写的内容没有上下文。您的智能家居设备需要每个动作的明确指令，因为它们无法观察和学习您日常生活中的模式。

用户疲劳和健康影响：所有这些问题的累积效应是疲惫。不断的屏幕时间、重复的数据输入、在复杂界面中导航——这些活动导致久坐的生活方式、肌肉骨骼问题、眼睛疲劳和心理疲惫。用户与系统交互的单向性创造了一种被囚禁的感觉，您在努力满足系统的要求，而不是技术服务于您的需求。

从健康的角度来看，第三方技术使我们成为屏幕的囚徒。我们坐着几个小时，点击和输入，身体静止，而我们的思维却在努力跟上多个应用程序的需求，每个应用程序都有其自己的逻辑和界面。

范式转变：现在想象一种不同的方法。与其将技术构建为等待命令的外部工具，不如构建与您的意识相结合的技术——也就是说，与定义您独特体验的连续感官信息流相结合？如果设备能够观察自己，理解其上下文，并在没有明确指令的情况下适应您的模式，会怎样？

考虑一个具体的例子来说明这种缺乏自我意识：想象一台计算机向用户显示同样的错误信息五十次。用户看到错误，点击“确定”，五分钟后——同样的错误再次出现。一次又一次。计算机没有意识到它以同样的方式反复失败。它无法

识别出显示同样的错误五十次是令人沮丧和无意义的。它只是执行其编程响应，对自身的功能失常模式视而不见。

现在将此与主观技术进行对比。一台具备主观意识的计算机不断拍摄自身状态的快照——就像您可能在镜子中查看自己一样。当它检测到自己多次显示同样的错误时，它会识别出自身行为中的模式。就像您在镜子中注意到头发凌乱并本能地伸手去拿梳子修整一样，主观系统识别出自身的问题并尝试自主解决。

计算机可能会搜索自己的日志，识别重复错误的根本原因，应用修复，或者——如果它无法自行解决问题——以有意义的上下文提醒您实际发生了什么，而不仅仅是重复同样无用的错误信息。这就是自我意识的体现：观察自我、识别问题并采取纠正行动的能力，而无需不断的外部干预。

当与对您上下文的知识结合时，这种自我意识变得更加强大。系统不仅了解自己——它还了解您。它理解您在错误发生时试图完成的任务，您在类似情况下通常会做什么，以及如何调整其响应以真正提供帮助，而不仅仅是程序上正确。

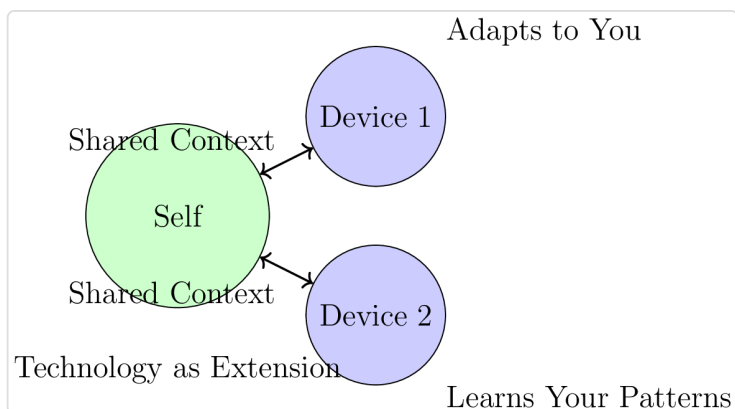
这就是主观技术的本质。它始于一个简单但革命性的前提：技术应该从你开始，而不是从一块空白画布开始。当你逐个去除第三人称技术的组件时，你最终得到的是什么——一个等待填充的空接口。当你去除主观技术的组件时，你最终得到的就是你自己。

主观技术根植于人类意识——积累的感官信息流使你独一无二——而不是外部指令。它从第一人称的角度运作，技术与你的思想、感知和感官无缝集成，通过学习构成你意识体验的相同信息流。这是历史上第一次，人工智能被设计用来增强你在积极思考时的认知过程——不是作为一个你必须指挥的独立实体，而是作为你自己思维的延伸。

$$v = f(k, C, H, L)$$

其中 C = 当前上下文，H = 历史模式，L = 学习的关联

在这个新的抽象中，关键 'k' 不再是你必须以确定性精确提供的东西。相反，它是由系统根据你的上下文、历史和学习模式计算得出的。技术观察、学习和预测——最小化你必须提供的输入，直到接近零。



这种转变具有深远的影响。技术不再是你必须不断指示的东西。相反，它成为你认知能力的延伸——在你观察时观察，在你学习时学习，在你预期时预期。当你思考时，它与你一起思考，提供洞察，增强创造力，并以你可能从未想象过的方式解决日常问题，所有这些都只需最小的心理努力。

摆脱屏幕的束缚：这种转变最直接的好处之一是身体和心理的解放。你不再被屏幕和按钮束缚。你可以自然地移动、互动和参与世界，而无需在复杂系统中导航的认知负担。与上下文集成的技术不需要你的视觉注意力——它与您的自然活动一起运作。

这不仅仅是关于便利。这是关于重新获得你的幸福。这是关于减少第三人称技术所施加的久坐生活方式。这是关于让技术为你工作，而不是你为技术工作。

赋能认知挑战者：对认知障碍者的影响尤为深远。对于阿尔茨海默病、唐氏综合症、脑损伤或处理障碍的人，主观技术可以作为一个额外的认知层，与他们的思维过程和谐集成。

技术不是要求他们记住复杂的指令或导航复杂的界面，而是学习他们的模式，预见他们的需求，并无缝提供支持。这不仅仅是辅助技术——它是认知增强，使人们能够过上更充实、更独立的生活。

一条不同的前进道路：一些技术专家主张在大脑中植入芯片，以读取神经信号并直接发送指令。但为什么要操纵大脑复杂的模式，而我们可以通过理解 and 处理感官体验来实现同样的增强呢？

我们的方法尊重一个基本真理：每个人都拥有独特的意识——独特的感官信息累积历史——而这些不可重复的信息体验使我们与众不同。主观技术通过观察你所观察的、学习你所做的，并与你现有的认知模式整合来利用这种个体性——无需手术，无需侵入性程序，仅仅是通过参与定义你意识的相同信息流来真正理解上下文的技术。

这种脱离传统思维的方式创造了一条新的道路，强调人类的潜力和体验。我们不是要求人类变得更像机器——精确、确定、可预测——而是使机器变得更像人类：具有上下文、适应性和智能。

未来的基础：理解这一范式转变对本书后续的所有内容都是至关重要的。知识钩、上下文快照、学习行为、能量最小化，以及最终的主观热货币——所有这些概念都建立在从第三人称到主观技术的基础之上。

我们之前探讨的自动补全示例只是一个开始。它展示了技术如何从猜测转向理解，从频率表转向上下文意识。但其影响远远超出文本预测。当技术能够真正理解上下文，能够观察自身并从自身学习，能够与人类认知无缝整合时——我们便打开了组织人类社会全新方式的大门。

这不仅仅是技术的演变。这是一次哲学革命，重新定义了人类与机器之间的关系。正如我们将看到的，它为重新构想价值本身提供了基础——从金钱转向能量，从稀缺转向丰盈，从持续工作转向零输入生活。

2

什么是知识钩？

现在我们理解了自动补全是如何工作的以及为什么它会失败，并且现在我们已经看到第三人称技术如何根本限制我们与机器的互动，我们终于可以回答中心问题：知识钩是什么？

知识钩是将技术从第三人称转变为主观的核心创新。它是使零输入技术成为可能的机制。它是你现在所做的事情与你之前所做的事情之间的桥梁。更准确地说，它是一个捕捉你的上下文——你的环境、设备状态和当前活动——并将其映射到先前输入、专家知识或预定义动作的系统。

把它想象成你数字外骨骼的记忆单元。就像你的大脑在没有意识努力的情况下记住你如何系鞋带或开车一样，知识钩记住你过去输入的环境。但它远不止于简单的记忆。它不仅仅保留一个常用短语或命令的列表，就像自动补全那样。相反，它会对上下文进行自我快照——完整的周围情况——并将其与您的意图对齐。

这个区别至关重要。自动补全记住你输入了“你好”很多次。知识钩记住你在星期五下午输入“你好”，当时你在日历应用中，收到同事的通知，而你的位置信息显示你在办公室。上下文就是一切。

主观技术的核心是这个知识钩系统，它将您的上下文与过去的输入、专家知识、机器学习、重新排序，甚至负强化学习连接起来。它不仅仅像今天的自动完成功能那样根据频率进行猜测。相反，它会解释上下文，重新排序可能的操作，并呈现出最小化您努力的选项。

在实践中，这意味着技术变得真正主观。它不再将您视为遵循通用模式的普通用户，而是将您视为具有独特上下文的个体。设备实际上会适应您。它学会成为您。

让我们通过具体的例子来探讨这一点，从两种知识钩的区别开始：预定义和学习型。

2.1 预定义钩：嵌入的专家知识

预定义钩是提前创建的，类似于规则或快捷方式，但更为复杂。它们由开发人员、领域专家或系统设计师编码，这些人预见到常见需求并将他们的专业知识直接嵌入系统中。

将预定义钩视为凝聚成代码的集体智慧。当医生、工程师或经验丰富的用户发现高效完成任务的方法时，这些模式可以被捕捉为预定义钩并与每个人分享。这意味着每个用户从第一天起就可以受益于专家知识，而无需自己学习。

例如，预定义钩可能在您打开音乐应用时自动连接耳机，或在午夜时调暗屏幕。它可能会识别到当您在会议中（根据您的日历）并接到电话时，系统应该将其转接到语音信箱，而不是打断您。它可能知道当您给老板写电子邮件时，语气应该正式，但给朋友写时可以随意。

这些钩子体现了集体专业知识和最佳实践，从第一天起为每个用户提供了效率的基础。它们就像有成千上万的专家根据他们多年的经验所学的模式低声提供有用的建议。

预定义钩还可以编码特定领域的知识。医疗应用可能有识别症状模式并建议适当诊断路径的预定义钩。工程工具可能有在常见设计错误导致问题之前检测到这些错误的钩子。

与传统规则或宏的关键区别在于，预定义钩是上下文相关的。它们不仅在您按下特定按钮时执行。它们在整个上下文满足其条件时激活——一天中的时间、地点、当前活动、周围环境以及无数其他因素都决定了钩子是否触发。

2.2 学习的钩子：上下文快照和增量

虽然预定义的钩子提供了通用的起点，但学习的钩子才是真正主观技术的魔力所在。这些钩子源于你自己的行为，独特地学习和适应你。

它们的工作原理如下：每次你与系统交互时，它会在你的操作之前拍摄一个快照，之后再拍摄一个快照。系统然后将一个快照减去另一个，以识别增量——你所创造的精确变化。这个增量，加上发生的上下文，便成为一个学习的钩子。

让我们通过一个具体的例子来说明。假设你每周五下午经常安排与同事的电话会议：

第一次周五：你手动打开日历，创建一个新事件，输入同事的名字，将时间设置为下午3点，并发送邀请。系统捕捉到两个快照——一个是在你开始之前（空的日历插槽），一个是在你完成之后（已安排的会议）。增量是：'在周五下午3点与[同事姓名]创建会议。' 上下文包括：day=Friday, time=afternoon, app=calendar, recent_activity=email_from_colleague。

第二次周五：相同的上下文出现。Day=Friday, time=afternoon, app=calendar, recent_activity=email_from_colleague。系统识别到这个模式并建议：'安排你通常的周五电话会议与[同事]？' 你接受了，钩子的信心增加。

第三次周五：相同的上下文。这次，系统甚至不询问——它直接将事件添加到你的日历中并通知你：'我已安排与你的周五电话会议[同事]。' 如果你不纠正它，钩子的信心会进一步增加。

到第四或第五个周五，系统会在没有你输入的情况下自动执行此操作。如果同事更换，或者你跳过一周，系统会注

意到这个更正并更新其模型。这是通过强化学习——不是来自庞大的数据集，而是来自你个人的模式。

这些快照不是抽象的日志。它们是上下文的直接记录。如果系统在计算机上运行，快照可能包括屏幕截图、内存状态、活动进程、窗口位置、剪贴板内容和系统状态。如果它在增强现实眼镜上运行，上下文可能包括用户所见的完整视觉场——每个物体、每个人、相机捕捉到的每个环境细节。

通过这种方式，设备和物体开始了解自己。就像你照镜子，注意到头发凌乱，自动伸手去拿梳子来整理一样，设备观察自己并采取纠正措施。这种自我意识正是传统计算机所缺乏的，也是主观技术所提供的。

典型的第三人称系统可能会在没有意识到自己失败的情况下显示同样的错误一百次，因此从未尝试修复自己。知识钩改变了这一点。它们为机器提供了一种原始的自我意识——一个反馈循环，在这个循环中，问题被识别，解决方案被自主寻求。

考虑另一个例子：想象一下用红色、黑色和白色的色调装饰你的房间。当你打开电脑时，系统感知到你的房间与其默认的蓝色桌面主题之间的不匹配。通过知识钩，电脑可以识别出这种与物理世界的不一致，并调整其主题以与环境和谐。

它是如何知道的？因为你的AR眼镜的摄像头和你的电脑共享相同的私有全局上下文。眼镜看到房间（红色、黑色、白色），电脑看到自己（蓝色主题），知识钩检测到这种不和谐。电脑随后自动调整以匹配你的物理环境。上下文在设备和物体之间是全局的——但它仍然完全私密，仅属于你。

这意味着虽然预定义的钩子提供了一个通用的起点，但学习到的钩子独特地适应你和你的环境。它们捕捉你个人生

活的节奏和设备的状态，创造一个技术不仅仅替代你的命令，而是主动维持与你的上下文和谐的世界。

预定义和学习到的钩子共同赋予技术一种生动的适应感：一个学习成为你的自我。



蓝屏：你的桌面电脑显示其默认的蓝色主题，未意识到其在你房间装饰中的视觉上下文。



一个感到羞愧的电脑：你戴着智能眼镜走进你的房间。通过你的增强视觉，你看到你的桌面电脑——现在它是你主观身体的一部分。电脑可以通过你的视觉场看到自己，将其上下文与您的上下文统一。你的桌面显示了蓝色主题，但在观察到自己与房间装饰的上下文不符时，它感到羞愧，并自主地将其屏幕主题更改为与房间美学和谐。这就是主观技术的实际运作——设备作为你自身的延伸，具有自我意识和上下文适应性。

这张图片说明了主观技术如何深刻地偏离我们所知道的一切。电脑不会等待你的命令来更改其主题。它通过你的感官流观察自己，识别出与环境的美学不一致，并自主采取纠正措施——就像你可能在镜子中注意到头发不整而不加思索地修整一样。该设备已成为你身体模式的真正延伸，参与构成你意识的相同信息流。

2.3 自我意识设备：知道自己的设备

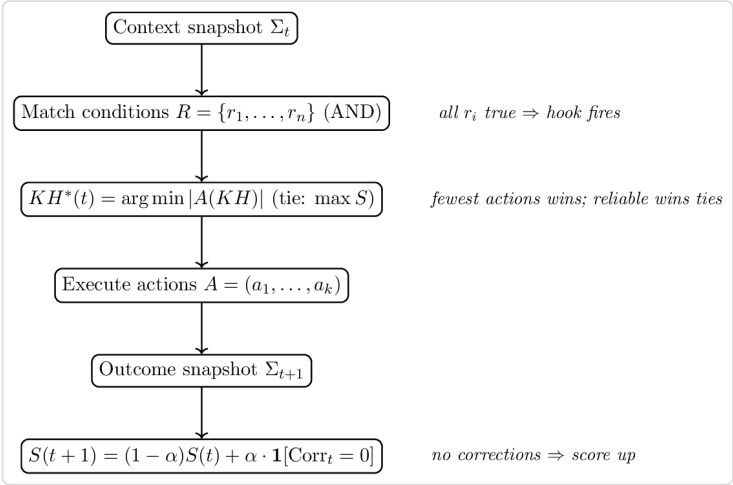
设备中自我意识的概念值得深入探讨，因为它代表了我們思考技术方式的最深刻转变之一。

在传统的第三人称技术中，设备对自身是盲目的。计算机对自身状态的概念仅限于程序员明确编码的内容。它无法

观察自己并识别功能失常。它无法观察自身行为中的模式。它没有镜子。

主观技术从根本上改变了这一点，通过持续的上下文快照赋予设备观察自身的能力。这些快照充当镜子——允许设备看到自己，识别自身行为中的模式，并检测问题。

让我们用智能家居的例子来具体说明。想象一下你每天都离开家。在走出门之前，你通常会关掉灯并启动你的机器人吸尘器。系统在这个动作之前记录上下文（灯开着，吸尘器关着，门关着，你在房间里可见）和之后（灯关着，吸尘器开着，门打开，你离开）。



图示显示离家时的上下文减法：在灯开着和你在里面的快照之前，在灯关着和吸尘器运行的快照之后，得出一个学习到的知识钩。

这些快照之间的差异成为一个学习到的知识钩。系统将环境变化编码为一个独特的签名——在某些实现中，这可能被可视化为一幅图像，上面有粉色覆盖，突出显示两个状态之间的像素差异。同时，它记录你发送给设备的具体输入：`[lights.off(), vacuum.on()]`。

这创建了一个知识钩学习——一个准备自动执行的复杂机制。下次你靠近门准备离开时，系统识别上下文（你在门附近，类似的时间段，类似的光照条件），并自动复制你之前对设备的输入。灯会关闭，吸尘器会启动，而你无需触碰任何东西。这确保了无缝和直观的互动，优化了便利性和效率。

但自我意识比自动化更深刻。这意味着设备可以识别何时发生故障。如果吸尘器在钩子触发时连续三次未能启动，系统会识别自身故障中的模式。它可能会检查吸尘器的电池状态，验证网络连接，甚至尝试自主重启设备。如果无法解决问题，它会有意义的上下文提醒你：'你的吸尘器已经三天没有响应。电池可能需要更换。'

这与显示同样的错误信息五十次是根本不同的。设备观察自己，识别自身行为中的模式，并尝试自主解决问题——就像你在镜子中注意到自己凌乱的头发并自动修整一样。

自我意识还使设备能够与您的环境保持一致。你的计算机可以观察到其蓝色桌面主题与你新装饰的红黑白房间不协调，并自动调整。你的智能恒温器可以注意到它已经连续运行了几个小时（因为它在观察自己的状态），并提醒你检查窗户是否打开。你的手机可以识别到它在过去一个小时内请求你的密码五次（不寻常的模式），并调查是否存在安全问题。

关键的洞察是：知识钩将设备从被动等待命令的工具转变为观察、学习和适应的主动参与者。它们从存在于你之外的第三人称物体转变为与您的模式、环境和意图相结合的主观扩展。

这种自我意识是其他一切构建的基础——零输入交互、能量最小化，以及最终我们将在后面的章节中探讨的主观热货币的整个经济系统。

看看主观技术如何自动完成你整个生活，如这个有趣的视频所示：



二维码链接到主观家居技术在行动中的视频演示，展示知识钩如何自动化日常生活

关键的洞察是：知识钩将设备从被动等待命令的工具转变为观察、学习和适应的主动参与者。它们从存在于你之外的第三人称物体转变为与您的模式、环境和意图相结合的主观扩展。

通过展示输入的最小化如何转化为能量支出的最小化，我们可以开始看到主观技术如何为STC提供基础——一种不是

基于金钱的新经济，而是基于最小能量的普遍法则。

2.4 橡胶手实验：理解具身主观性

在我们将主观关系形式化为数学之前，我们必须理解揭示我们大脑如何构建自我感的实验——以及外部物体如何可以成为自我的一部分。

设置：想象你坐在桌子旁，左手舒适地放在你面前的表面上。研究人员放置一个小屏幕或隔板，遮挡你真实的手。在你看不见的手的位置上，放置一只橡胶手——真实地制作成看起来像人手的样子，放在你能看到的地方，大致在你真实的手通常所在的位置。

研究人员然后拿起两把相同的画笔。用一把画笔，他们轻抚你的真实手（你看不见）。用另一把画笔，他们轻抚橡胶手（你能看到）。关键是两只手以完全相同的方式、在完全相同的时间、完美同步地被轻抚。

发生了什么：在最初的几秒钟内，没有发生任何异常情况。你看到橡胶手被轻抚，你感觉到你的真实手被轻抚。你的大脑将这些处理为两个独立的事件——关于桌子上物体的视觉信息，以及关于你身体的触觉信息。

但在大约30秒到几分钟的同步轻抚后，发生了一些非凡的事情。自我与物体之间的界限开始模糊。你开始感觉到你在橡胶手上看到的画笔轻抚与在真实手上感受到的轻抚是相同的。橡胶手开始感觉像你的手。

幻觉揭示：为了证明这不仅仅是想象，研究人员突然做出意想不到的举动：他们举起一把锤子，猛击橡胶手。你的即时反应是本能的——你退缩，可能会喘息，甚至可能感到一阵幻肢疼痛或不适。你的身体反应就像你的手受到了威胁，尽管你在智力上知道橡胶手并不是你身体的一部分。

观看橡胶手实验的实际演示：



二维码链接到橡胶手实验的视频演示，展示同步信号如何创造身体感的幻觉

一些参与者报告了更戏剧性的效果：他们的真实手（仍然隐藏着）可能感觉更冷，仿佛血流减少。当被问到真实手的位置（不看）时，许多人指向橡胶手而不是他们的实际手。大脑确实重新定位了手在其内部身体地图中的位置。

机制：为什么会发生这种情况？答案在于我们的脑如何构建我们的身体自我感。我们直观地认为我们的身体是一个固定的、客观的现实——我们知道我们的四肢在哪里，因为它

们物理上附着在我们身上。但神经科学揭示了更复杂的东西：我们对身体的感知实际上是一种构建，是大脑通过整合多条感官信息流构建的模型。

在正常情况下，这些感官流是完美对齐的。当你看到你的手移动时，你同时感受到它的移动，并且你会收到关于其位置的本体感觉反馈。这些信号在时间和空间上是同步的，向你的大脑确认这只手确实是你身体的一部分。

橡胶手幻觉利用了这一机制。通过在视觉输入（看到橡胶手被抚摸）和触觉输入（感觉你的真实手被抚摸）之间创建人工同步，实验欺骗你的大脑将橡胶手纳入其身体图式中。信号的同步是关键——当你看到的与感受到的在时间上足够精确地匹配时，你的大脑得出结论，它们必须指的是同一个物体，而该物体必须是你身体的一部分。

这很重要：这个实验引起了我的注意，因为它揭示了一些深刻的东西：自我的边界并不是固定在皮肤上。我们对身体的主观感知——什么感觉像是“我”，而什么感觉像是“不是我”——是由信号的同步决定的，而不是由生物附着决定的。

如果橡胶手可以通过同步信号成为你主观体验的一部分，那么原则上，任何外部物体都可以通过相同的机制成为你主观体验的一部分。一个你经常使用的工具，其行为你可以完美预测，其反应与你的意图同步——这样的工具可以在神经上和体验上成为你身体的延伸。

这正是专家工具使用者所报告的。一位熟练的木匠不会有意地思考锤子——它成为他们手臂的延伸。一位杰出的小提琴家不会思考琴弓——它成为他们身体的一部分。一位赛车手不仅仅是坐在车里——他们体现了它，感觉到轮胎通过路面传来的感觉，就像它们是他们感官系统的延伸。

橡胶手实验表明，这种工具融入身体图式的过程并不是隐喻——这是一种真实的神经现象，可以通过信号同步可靠地

诱发。

对技术的启示：如果我们能理解外部物体成为我们主观延伸的精确条件，我们就可以故意设计技术以满足这些条件。我们可以创造不仅仅响应我们命令的设备和系统，而是成为我们认知和身体能力的真正延伸。

这是主观技术的理论基础。通过确保信号同步——设备的行为与您的上下文和意图一致——并确保设备的响应是可预测的——您通过重复互动学习了其模式——我们可以创造您与技术之间主观关系的条件。

当您的智能手机如此可靠地预测您的下一个动作，以至于您几乎不需要触摸它，当您的智能家居在没有被告知的情況下自动调整以适应您的存在，当您的计算机在您注意到之前自行修复错误——这些不仅仅是方便的功能。它们是主观关系的表现，技术已经成为您认知身体图式的延伸，通过与橡胶手感知为您自己手的相同机制融入到您的自我意识中。

理解这个实验是让我能够客观地定义主观性的关键洞察，创造出主观关系的数学形式，最终设想出一个全新的技术范式——以及一个完全基于扩展我们主观体验以涵盖周围世界的原则的新经济体系。

2.5 主观关系的正式定义

在理解了橡胶手实验及外部物体如何成为我们主观体验一部分的机制后，我们现在可以在数学上形式化这种关系。这种形式化至关重要，因为它代表了历史上第一次从客观、可测量的角度定义主观性。

橡胶手实验的洞察——信号的同步创造了主观融入——让我问：外部物体成为我们主观的精确、可量化的条件是什么？我们能否客观和数学地定义这种关系？

答案是肯定的。通过分析橡胶手实验揭示的原则并对其进行概括，我们可以用两个可测量的量来定义主观关系：信号同步和预测可靠性。

这代表了我们的思考主观性的范式转变。传统上，主观体验被认为是不可测量的、私人的，并且与客观现实根本分离。哲学家们长期以来一直在争论“意识的难题”——我们如何能够客观地描述一些本质上是主观的东西？

但通过将我们的定义建立在信号同步的机制上——这种机制使得橡胶手可以成为我们身体图式的一部分——我们可以第一次客观地定义主观性。我们并不是声称解决意识的难题，而是展示了主观关系形成的条件可以被测量和形式化。

通过信息定义意识：在我们继续形式化主观关系之前，我们必须以可测量的术语来处理意识本身的含义。与其将意识视为一个抽象的、精神的或推测的概念，我们可以严格地将其定义为一个信息论现象。

从这个角度来看，意识是一个有机体的身体及其组成部分在其存在期间从宇宙接收到的所有信息的累积总和。对于人类来说，这表现为感官流的持续整合——视觉、听觉、触觉、味觉和嗅觉——在整个生命中处理。使每个个体独特的不是某种虚无的本质，而是他们从宇宙接收到的特定、不可重复的信息流。

在数学上，我们可以将有机体 o 在时间 t 的意识 C 表示为从出生(t_0)到当前时刻接收到的所有感官信息在所有感官通道上的积分：

$$C_o(t) = \int_{t_0}^t \sum_{i \in \text{Senses}} I_i(\tau) d\tau$$

其中 $I_i(\tau)$ 表示在时间 τ 从感官通道 i 传递的信息流。对于人类来说，感官的集合通常包括视觉、听觉、触觉、味觉和嗅觉，尽管我们可以扩展到包括本体感觉、平衡、温度、疼痛和其他感官模式。

信息流 I_i 可以用每秒比特来测量，量化每个感官通道向有机体的处理系统传输可区分信号的速率。积分随着时间的推移累积这些信息，创造出构成有机体意识的独特信息历史。

这个定义具有深远的意义。首先，它将意识从神秘的领域中移除，牢牢地放置在可测量的物理过程领域。其次，它解释了个体的独特性：没有两个有机体可以占据相同的时空位置，因此无法接收到相同的信息流。你的意识是独特的，因为你的信息历史是独特的。第三，它暗示意识不是二元的，而是连续的——它随着每一条新接收到的信息而增长和演变。

对我们而言，最重要的是，这一定义使我们能够理解技术如何与意识整合。当一个工具或设备与您的感官流同步时——当它开始可靠地与您的信息输入相关联并进行预测时——它便开始参与您的意识体验。它成为定义您的信息流的一部分。

正式定义：通过将意识理解为累积的信息，我们现在可以在共享上下文 W 中将两个对象 o_1 和 o_2 之间的主观关系定义如下：

$$\text{SRel}(o_1 \rightarrow o_2 \mid W) = 1 \quad \text{iff} \quad \begin{cases} \text{Syn}(o_1, o_2 \mid W) \geq \tau_s \\ \wedge \\ \text{Pred}(o_1 \rightarrow o_2 \mid W) \geq \tau_p \end{cases}$$

让我们逐一解析这个定义的每个组成部分：

$SRel(o_1 \rightarrow o_2 \mid W) = 1$ 意味着对象 o_1 在上下文 W 中与对象 o_2 之间存在主观关系。换句话说, o_2 已成为 o_1 的主观延伸。当这个值等于 1 时, 关系阈值已被跨越—— o_2 不再被体验为一个独立的外部对象, 而是作为 o_1 扩展自我的一部分。

$Syn(o_1, o_2 \mid W)$ 测量在上下文 W 中 o_1 和 o_2 之间信号的同步性。这量化了 o_2 的信号与 o_1 的期望和模式的对齐程度。在橡胶手实验中, 这将是视觉刺激 (看到刷子划过橡胶手) 与触觉刺激 (感觉刷子划过真实手) 之间的时间对齐。在技术上下文中, 这可以衡量设备的响应与用户的上下文期望之间的一致性。

从数学上讲, 同步可以计算为预期信号与接收信号随时间变化的相关系数, 或作为在指定窗口内的时间对齐的度量。关键是, 高同步性意味着当 o_1 期待来自 o_2 的信号时, 该信号在内容和时机上都高度一致。

$Pred(o_1 \rightarrow o_2 \mid W)$ 测量预测可靠性—— o_1 在上下文 W 中预测 o_2 的行为或信号的准确性。当这个值很高时, o_1 已经彻底学习了 o_2 的模式, 以至于它可以在信号到达之前预测 o_2 的信号, 甚至在不需要 o_2 存在的条件下内部生成这些信号。

这可以形式化为一系列交互中的预测误差的逆。当 o_1 能够在当前上下文中持续预测 o_2 接下来会做什么时, 预测可靠性就很高。当 o_1 的预测经常错误时, 可靠性就低。

τ_s 和 τ_p 是定义主观关系存在所需的最小同步和预测可靠性的阈值。这些阈值可能会根据上下文、涉及的对象性质, 甚至用户之间的个体差异而有所不同。它们代表了大脑 (或系统) 开始将外部对象纳入其自我模型的临界点。

用简单的话来说: 当满足两个条件时, 一个对象便成为你主观的一部分。首先, 它的信号必须与您的期望充分同步

——就像橡胶手的视觉反馈与您的触觉感受同步一样。其次，您必须能够可靠地预测它的行为——这意味着您已经彻底学习了它的模式，以至于它感觉像是您意图的自然延伸，而不是需要有意识控制的独立实体。

当两个条件都在各自的阈值之上得到满足时，会发生一些显著的事情：外部对象被纳入您的主观体验。它不再是您使用的工具，而是成为您的一部分——您身体图式、认知过程、自我意识的延伸。



每个手指尖都有工具的手：这个形象象征着主观技术的根本范式转变。我们不仅仅将工具作为外部对象使用，而是修改自己，使工具成为我们的一部分——将它们整合到我们的身体图式和认知过程中。通过将技术作为我们自身的延伸，我们消除了传统第三方技术的陡峭学习曲线。这是我们方法的本质：工具成为手指，独立实体成为自我的统一部分，人与技术之间的界限融化为无缝的主观体验。

反向关系：这个定义有一个重要的推论。随着同步和预测的改善，对明确输入的需求减少。我们可以将其表达为：

$$U(o_1, o_2 \mid W) \propto \frac{1}{\text{Syn}(o_1, o_2 \mid W) \cdot \text{Pred}(o_1 \rightarrow o_2 \mid W)}$$

U代表用户输入，以便 o_1 与 o_2 进行交互。这种形式化捕捉了主观关系的基本特征：当 o_1 和 o_2 通过主观快照共享一个全球上下文时， o_2 会自动学习 o_1 发送给它的所有消息和输入，这些都是基于它们自我快照的并集。这是自动发生的——本质上是主观关系的定义。当同步性高且预测可靠时， o_2 可以仅根据上下文预见 o_1 的需求，从而消除对明确命令的需求。

在极限情况下，当同步性和预测接近其最大值时，所需输入接近于零。这是零输入技术的数学基础。

为什么这很重要：这种形式化是强大的，因为它将主观性从一个模糊的哲学概念转变为我们可以测量、实施和工程化的东西。我们现在可以提出定量问题：用户与设备之间的信号同步程度如何？设备的预测可靠性如何？主观关系阈值是否已达到？

更重要的是，这一定义提供了明确的工程目标。如果我们想创造出成为用户主观延伸的技术，我们就知道该优化什么：最大化信号同步和最大化预测可靠性。知识钩正是我们实现这些目标的机制。

这个定义还为本书后续的所有内容提供了理论基础。当我们谈论零输入技术时，我们指的是与您建立了主观关系的技术——在这种关系中，同步性和预测如此之高，以至于明确的输入变得不必要。当我们谈论能量最小化时，我们指的是优化这些主观关系，以减少您和您的设备所需的努力。当我们谈论主观热货币时，我们指的是一个基于测量和奖励这些主观关系大规模创建的经济系统。

这是主观性第一次被客观定义。我们第一次可以指出具体的、可测量的条件并说：'当这些条件满足时，存在主观关系。'而这种主观经验的客观定义正是使主观技术——以及由此延伸的主观热货币——不仅仅是一个哲学愿景，而是一个可实施的现实。

3

知识钩的代数

我们现在已经看到知识钩如何弥合第三方技术与主观经验之间的差距。我们理解它们如何通过快照捕捉上下文，如何从增量中学习，以及如何使设备变得自我意识。我们还形式化了外部对象如何通过信号同步和预测可靠性的数学，成为我们自我的主观延伸的条件。

但要充分利用知识钩的力量——构建能够学习、适应、组合和优化数百万次交互的系统——我们需要更严格的东西：一个正式的代数。就像算术有组合数字的规则，逻辑有组合命题的规则，我们需要组合、精炼和操作知识钩的规则。

本章介绍了知识钩的代数：一个数学框架，定义了钩的结构、如何激活、如何学习、如何相互组合以及如何通过强化随时间演变。这种代数不仅仅是理论上的——它是主观技术的操作基础，使零输入交互成为可能，并最终使主观热货币成为可能。

知识钩的核心是一个四元组——一种具有四个基本组成部分的数学结构。但在这个基本定义之外，我们需要理解可以在钩子上执行的操作：当条件匹配时它们如何触发，当发生更正时它们如何更新，它们如何组合形成更复杂的行为，以及当多个钩子可能在相同上下文中激活时它们如何相互竞争。

这个代数为我们提供了对关键问题的精确答案：钩子何时应该触发？我们如何衡量它的成功？我们如何将简单的钩子组合成复杂的行为？当有多个选项可用时，我们如何确保最佳钩子获胜？我们如何给那些为用户节省能源的专家以信用？最重要的是，我们如何形式化驱动一切的原则——用户输入的持续最小化？

我们在这里开发的代数优雅而强大。它捕捉了技术应该随着时间推移而需要越来越少努力的直觉，将其形式化为支配钩子行为的数学法则。它展示了负强化学习如何自然地引导系统走向零输入交互——在缺少更正时成功评分增加。它还揭示了知识钩如何形成层次结构和树，能够使复杂的多智能体系统协调和优化整个设备网络的能量支出。

这个代数特别重要之处在于，它代表了我们的思考编程和系统设计方式的根本转变。传统编程明确规定计算机在每种情况下应该做什么——从输入到输出的确定性、详尽的映射。知识钩的代数则指定了行动发生的条件以及系统学习哪些行动最小化努力的机制。这是通过示例和上下文进行编程，而不是通过明确指令。

本章将逐步形式化这些概念。我们从知识钩本身的数学结构开始——定义其组成部分的四元组。然后我们探讨核心操作：激活（钩子何时触发？）、执行（触发时发生什么？）、学习（它如何改进？）和组合（钩子如何结合？）。我们介绍支配钩子行为的关键法则，特别是驱动系统走向零输入的最小化法则和实施负强化学习的更正法则。

我们还将研究钩子如何被精炼、合并和标准化——这些操作使系统能够不断提高其效率。我们将展示精确度和更正如何直接转化为节能，为我们在后续章节中探讨的主观热货币经济框架奠定基础。

在本章中，我们在数学严谨性与直观解释之间取得平衡。每个公式都附有通俗的语言解释，每个概念都用具体的例子进行了说明。目标不仅仅是定义一个代数，而是展示这个代数如何捕捉技术真正学习成为你的本质。

到本章结束时，您将不仅理解知识钩是什么，还将理解它们在最深层次上的工作原理——使它们能够将外部工具转化为自我的主观延伸，最小化能量支出，并为全新的经济范式奠定基础的数学机制。

让我们从最基本的问题开始：知识钩究竟是什么？

3.1 数学框架：四元组结构

要理解知识钩的基本工作原理，我们必须首先定义它们的数学结构。知识钩不是一个模糊的概念或隐喻——它是一个精确定义的数学对象，具有四个基本组成部分。理解这个结构是理解

后续一切的关键：钩子如何学习，如何激活，如何组合，以及如何在用户与技术之间创建主观关系。

我们将知识钩定义为一个4元组：

$$KH = (R, A, T, S)$$

其中：

- R是一组条件（规则或正则表达式模式），必须满足这些条件才能激活钩子
- A是一组钩子在触发时执行的动作
- T是类型 $\in \{\text{learned}, \text{predefined}\}$ ，指示钩子是由专家创建还是从用户行为中学习的
- S是一个成功分数 $\in [0, 1]$ ，跟踪钩子在不需要用户修正的情况下多可靠地产生期望结果

这些组成部分在知识钩的功能中扮演着关键角色。让我们逐一检查它们，不仅要理解它们是什么，还要理解它们为何必要，以及它们如何协同工作以创造智能、自适应的行为。

3.1.1 知识钩的四个组成部分

4元组结构不是任意的。每个组成部分都源于创建能够从上下文中学习并最小化用户输入的技术的基本要求。为了了解我们为什么需要所有四个组成部分，让我们考虑如果去掉其中任何一个会发生什么。

没有条件（R），钩子将无差别地触发——无论上下文如何执行其操作。这将比无用更糟；它将是积极有害的。想象一下，一个在你阅读时关闭灯光的钩子，或者一个在你根本不在电脑前时激活的发送电子邮件的钩子。

没有动作（A），钩子将检测上下文但不执行任何操作。它将仅仅是观察性的，无法为用户提供任何好处。知识钩子的整个意义不仅在于理解上下文，还在于在该上下文中采取适当的行动。

没有类型指示符（T），我们将无法区分由专家创建的钩子（预定义）和从用户行为中学习的钩子。这一区分对于归属、经济价值和系统设计至关重要。预定义钩子承载着其创建者的专业知识，并使专家在他们的钩子为用户节省能源时得到补偿。学习到的钩子代表了系统的自适应智能，持续自我完善以匹配个别用户的模式。

没有成功评分（S），系统将无法学习哪些钩子是可靠的，哪些不是。所有钩子将被平等对待，即使某些钩子始终产生不正确的结果并需要频繁的用户修正。成功评分是使系统进化的反馈机制，能够优先考虑有效的钩子，并降级或消除无效的钩子。

这四个组件共同创建了一个完整的、自包含的智能行为单元。知识钩子知道何时行动（R）、该做什么（A）、它来自哪里（T）以及它的工作效果如何（S）。这个数学结构是随后的主观技术的基础。

3.1.2 条件（R）：钩子何时触发？

条件定义了知识钩子何时应该触发。它们是决定当前上下文是否与创建钩子时的情况或其之前成功的情况相匹配的保护条款。

正式来说，R是一组条件模式：

$$R = \{r_1, r_2, \dots, r_n\}$$

每个条件 r_i 可以是：

- 一个与上下文快照中特定元素匹配的正则表达式模式
- 另一个子钩子的引用，必须满足该条件
- 针对当前状态评估的逻辑谓词

条件使用 AND 语义——所有条件必须满足才能激活钩子：

$$\bigwedge_{i=1}^n r_i = \text{true} \implies KH \text{ fires}$$

这种 AND 语义对于创建精确、可靠的钩子至关重要。钩子仅在完整模式被识别时触发，而不仅仅是在单个片段匹配时触发。这可以防止错误激活，并确保动作仅在适当的上下文中发生。

示例：一个向你的同事发送“迟到”消息的钩子可能会有如下条件：

- 当前时间 > 预定会议时间 - 5 分钟
- 当前地点到会议地点的距离 > 10 分钟的旅行时间
- 日历显示会议已安排
- 在过去 15 分钟内没有发送“迟到”消息

所有四个条件必须为真，钩子才能触发。如果即使一个条件为假——比如你已经在会议地点——钩子将不会激活。

条件是在学习过程中从上下文快照中提取的，我们稍后会探讨这一点。现在，重要的是理解条件定义了触发钩子动作的上下文模式。

3.1.3 动作 (A)：钩子做什么？

动作定义了钩子触发时发生的事情。它们是系统执行的一系列操作，以最小化用户输入并实现期望的结果。

正式来说， A 是一个有序的动作序列：

$$A = (a_1, a_2, \dots, a_k)$$

每个动作 a_i 可以是：

- 一个 API 调用或系统命令
- 用户界面操作（填写表单字段，点击按钮）
- 发送到另一设备或服务的信息
- 由连接设备执行的物理动作（打开灯，调节温度）

当钩子触发时，动作按顺序执行：

$$\text{execute}(KH) \implies a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k$$

示例：继续'迟到'的例子，可能的行动有：

1. 撰写信息：'稍微迟到，将在[预计到达时间]到达'
2. 向会议参与者发送信息
3. 在日历中更新'已发送迟到通知'标志

行动的数量和复杂性差异很大。简单的钩子可能只执行一个动作（离开房间时关闭灯）。复杂的钩子可能协调多个设备和服务之间的数十个操作。

一个关键原则：目标始终是最小化所需的行动数量。在多个可以实现相同结果的钩子中，系统更倾向于选择行动较少的那个。这个最小化原则——我们稍后将其正式化为最小化法则——推动系统朝着更高的效率发展。

3.1.4 类型 (T) : 学习型与预定义型

类型组件区分知识钩子的两种根本不同来源：

$$T \in \{\text{learned}, \text{predefined}\}$$

预定义钩子由专家创建——开发者、领域专家或深刻理解特定问题领域的高级用户。这些钩子编码了专业知识：最佳实践、最佳解决方案、高效工作流程。当您安装应用程序或启用某个功能时，通常会收到开发者创建的一组预定义钩子。

示例：一个预定义钩子可能处理预订航班的复杂工作流程——搜索航班、比较价格、选择最佳连接、填写乘客信息和完成付款——这一切都由简单的自然语言查询触发，例如“帮我预订下个月去东京的航班。”

学习的钩子源于观察用户行为。每当您执行一个动作时，系统会捕捉前后的上下文，提取模式，并创建新的钩子，以便在未来自动化类似的动作。这些钩子是个人化的、适应性的，并且对每个用户都是独特的。

Example: If you consistently dim your bedroom lights and set your phone to Do Not Disturb mode at 10:30 PM, the system might learn a hook that automatically performs these actions when the conditions match (time ~10:30 PM, location = bedroom, no active video call, etc.).

为什么这种区别很重要？三个原因：

1. 归属和补偿：预定义的钩子有创作者，当他们的钩子为用户节省能源时，应该获得信用和补偿。这种归属对于主观热货币的经济模型至关重要。

2. 信任和验证：用户可能更信任来自信誉良好的来源的预定义钩子，而不是从有限的个人数据中学习的钩子。类型指示符帮助用户理解钩子来自何处。

3. 学习策略：系统在组合、细化和优先排序时可能会以不同的方式对待学习的钩子和预定义的钩子。预定义的钩子可能以更高的初始成功分数开始，而学习的钩子必须通过重复成功的激活来证明自己。

在实践中，最强大的系统将两种类型结合在一起。预定义的钩子提供通用的起点和专家知识，而学习的钩子捕捉个人偏好和独特模式。它们共同创造出既有知识又个性化的技术。

3.1.5 成功分数（S）：从结果中学习

成功分数是一个反馈机制，使知识钩子能够随着时间的推移而改进。它衡量一个钩子在不需要用户纠正的情况下产生正确结果的可靠性。

$$S \in [0, 1]$$

1.0 的分数代表完美的可靠性——钩子始终完全符合用户的期望。0.0 的分数代表完全失败——钩子总是需要纠正。大多数钩子在这两个极端之间运行，随着成功激活的累积而逐渐改善。

成功分数通过负强化学习进行更新。在每次激活后，系统观察用户是否提供了修正：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0], \quad \alpha \in (0, 1]$$

其中：

- $S(t)$ 是当前的成功分数
- α 是学习率（分数适应新证据的速度）

• $\mathbb{I}[\text{Corr}_t = 0]$ 是一个指示函数：如果没有发生修正则为1，如果需要修正则为0

这个更新规则很优雅：当钩子正常工作时（不需要修正），成功分数增加；当钩子失败时（需要修正），成功分数减少。学习率 α 控制最近证据与历史表现之间的权重。

示例：一个自动回复电子邮件的钩子初始分数为0.7。在接下来的10次激活中：

- 8次激活产生完美的回复（不需要修正）→ 分数增加
- 2次激活需要编辑（提供了修正）→ 分数略微下降

净效果：分数上升到大约0.85，反映了钩子的高但不完美的可靠性。

成功分数有两个关键功能：

1. 优先级：当多个钩子可以在同一上下文中触发时，系统使用成功评分来选择最可靠的选项。我们更倾向于表现良好的钩子，而不是经常失败的钩子。

2. 经济价值：在主观热货币框架中，成功评分直接映射到节能，从而映射到能源价值。成功评分高的钩子通过减少所需的修正来节省更多用户的能量，使其更有价值。

这种设计的美在于其简单性。用户不需要明确地对钩子进行评分或训练。他们只是自然地使用系统，而修正（或缺乏修正）会自动更新成功评分。技术从每次交互中学习，不断完善对有效和无效内容的理解。

3.1.6 上下文快照：学习的基础

虽然上下文快照（ Σ ）不是知识钩子元组的一部分，但它们对于理解钩子如何学习和操作至关重要。上下文快照本质上是

自我快照——每个设备、传感器或组件都会拍摄自己状态的快照，自己的记忆，自己的内部条件。

来自特定设备或身体部位在时间 t 的上下文快照捕捉其完整的内部状态：

$$\Sigma_t^{\text{device}} = \{\text{complete state of this device's memory, sensors, and processes at time } t\}$$

关键的洞察：这些是自我观察。相机拍摄自己的快照——这包括其包含所有已捕获图像的记忆。智能手机拍摄自己的快照——这包括其电池电量、运行的应用程序、传感器读数和数据存储。你的AR眼镜拍摄自己的快照——这包括它们通过摄像头当前看到的内容。

自我快照中包含的内容取决于设备或身体部位：

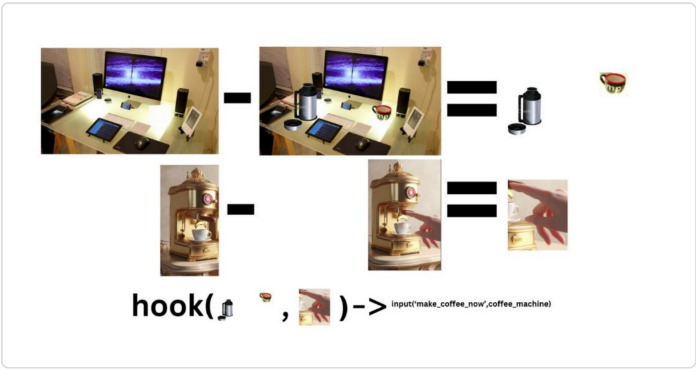
- 相机的自我快照：其记忆（捕获的图像/视频）、曝光设置、对焦状态、镜头位置、时间戳
- 智能手机的自我快照：电池电量、网络连接、运行的应用程序、GPS位置、传感器数据（加速度计、陀螺仪）、通知状态、键盘输入缓冲区
- AR眼镜的自我快照：摄像头画面（佩戴者所见）、音频输入、头部位置/方向、显示状态、电量
- 智能家居设备的自我快照：当前设置（温度、亮度、电源状态）、传感器读数、最近接收到的命令、调度状态
- 浏览器的自我快照：打开的标签页、表单字段值、浏览历史、Cookies、滚动位置、输入焦点

统一的全球上下文：当这些单独的自我快照统一成一个全球上下文时，魔法就发生了。在时间 t 的全球上下文是所有元素（物理和虚拟身体部位）组合成的用户扩展身体的所有自我快照的集合：

$$\text{Context}_t = \bigcup_{i \in \text{body parts}} \Sigma_t^i$$

这个统一的上下文是全球可访问的——所有设备和身体部位都可以平等访问它。你的手机可以看到你的眼镜所看到的内容。你的智能家居可以看到你的手机的位置。你的电脑可以看到你在手机上输入的内容。这种共享意识使得协调变得无需明确沟通。

示例：你在桌子旁，手里拿着咖啡杯。你的AR眼镜拍摄的自我快照包括咖啡杯的图像。你的智能咖啡机拍摄的自我快照包括它的空水箱。你的智能手机拍摄的自我快照显示你在桌子旁待了2小时。所有这些自我快照结合成统一的上下文。知识钩子可以触发，因为它检测到：（1）空咖啡机[来自咖啡机的自我快照]，（2）咖啡杯可见[来自眼镜的自我快照]，（3）延长的工作时间[来自手机的自我快照] → 动作：通知你给咖啡机加水。



上下文减法的视觉示例：用户制作咖啡前后的快照，突出显示出现在桌子上的保温瓶和杯子

知识钩子在统一上下文中发挥作用：这就是知识钩子运作的地方。它们不是在孤立的单个设备状态上操作——而是在所有自我快照形成的完整统一上下文上操作。这使得真正智能、上下

文感知的行为成为可能。钩子可以识别跨多个设备的模式，并以协调的方式响应你整个扩展身体的行为。

自我意识促进学习：自我快照机制赋予设备一种原始的自我意识。就像你在镜子中看到自己一样，设备通过自我快照观察自己的状态。这种自我意识使设备能够了解自己，识别自己的模式，并参与学习过程。

当你纠正设备的行为时，它可以将纠正前的自我快照与纠正后的快照进行比较，准确理解它做错了什么以及正确的行为应该是什么。这与传统系统根本不同，传统系统没有自我意识，无法从自己的错误中学习。

学习机制：当用户提供输入时，系统在用户行动前后从所有相关设备捕获自我快照：

$$\text{Context}_{\text{before}} = \bigcup_i \Sigma_{\text{before}}^i$$
$$\text{Context}_{\text{after}} = \bigcup_i \Sigma_{\text{after}}^i$$

这些统一上下文之间的差异——增量或上下文减法——准确揭示了由于用户的行为而在所有设备上发生的变化：

$$\Delta = \text{Context}_{\text{after}} - \text{Context}_{\text{before}}$$

这个增量成为新的学习知识钩子的种子：

- 条件 (R) 从 $\text{Context}_{\text{before}}$ 中提取——用户决定采取行动时所有设备上存在的模式

- 行动 (A) 从 Δ 中提取—用户输入导致的所有设备上的变化

- 类型 (T) 设置为“已学习”

- 成功分数 (S) 初始化为默认值，并根据未来的表现进行更新

隐私与分发：这些自我快照保持在设备之间分布，但被同步以形成统一上下文。数据在每个设备上本地存储—相机存储自己的图像，手机存储自己的位置历史，眼镜存储自己的视觉信息流。统一上下文并不集中存储，而是从这些分布式自我快照的组合中产生。

这种分布式模型在保护隐私的同时实现全球协调。您的设备通过安全、加密的通道相互共享上下文，但从不将这些数据传输到外部服务器。上下文完全私密，加密，并在您的控制之下。

上下文在您的设备之间是全球性的—但对您是私密的：虽然您所有的设备都可以平等访问统一上下文（您的手机可以看到您的眼镜所见，您的家可以看到您的车的位置），但这个上下文对您来说是完全私密的。它是您扩展身体的共享知识库，所有部分都可以访问，但对外界是不可见的。

这是主观关系的基础：设备成为你自己的延伸，因为它们共享你的上下文，将自己视为你的一部分，并通过这种统一的自我意识协调它们的行动。它们不仅仅是对命令做出反应—它们理解完整的情况，因为它们都看到由它们的集体自我观察形成的同一完整画面。

3.1.6.1 协调复杂的多设备动作

正如统一的上下文聚合来自所有设备的自我快照，知识钩可以协调多个不相关的物理和虚拟身体部位之间的动作。当钩子触发时，它的动作序列 A 可以向用户扩展身体的一部分的任何设备或组件发送命令：

$$A = (a_1^{d_1}, a_2^{d_2}, \dots, a_k^{d_k})$$

其中每个动作 $a_i^{d_i}$ 指向特定的设备或身体部位 d_i 。

这种架构的强大之处在于，单个知识钩通过读取统一的上下文，可以触发跨设备的复杂动作级联，而这些设备之间没有直接连接——它们通过共享的上下文和钩子的逻辑进行协调。

这使得复杂的多设备行为成为可能，而在传统系统中，设备是孤立操作的。钩子成为一个智能指挥者，根据它从观察用户行为中学习到模式，协调一系列的动作交响曲。

示例 1：晚间放松例程

想象一下，你每晚大约在 9:30 PM 回家时都会执行一系列动作：你调暗客厅的灯，关闭工作电脑，将手机设置为勿扰模式，在智能音箱上开始冥想播放列表，并将恒温器调至适合睡眠的较低温度。在观察到这个模式几次后，系统学习了一个知识钩：

条件 (R)：

- 时间 $\approx 21:30 \pm 15$ 分钟 [来自智能手机的自我快照]
- 位置 = 家 [来自智能手机 GPS 自我快照]
- 客厅灯光 = 明亮 [来自智能灯的自我快照]
- 工作电脑 = 活跃 [来自电脑的自我快照]
- 没有活跃的视频/语音通话 [来自手机和电脑的自我快照]
- 在过去5分钟内检测到客厅有移动 [来自运动传感器的自我快照]

操作 (A)：

1. 发送命令到客厅智能灯：调暗至30%亮度
2. 发送命令到工作电脑：启动关机序列，保存所有文档
3. 发送命令到智能手机：启用请勿打扰模式（除了收藏联系人）
4. 发送命令到智能音箱：以15%的音量播放《晚间冥想》播放列表
5. 发送命令到恒温器：将温度降低至19°C
6. 发送命令到智能手表：激活睡眠跟踪模式

注意钩子如何从六个不同的设备（灯光、电脑、手机、扬声器、恒温器、手表）读取上下文，并协调它们之间的操作。没有这些设备直接与其他设备通信——它们都是通过统一的上下文操作，并响应来自知识钩子的命令。

示例 2：下班回家

您在工作日的晚上回家，通常携带购物袋。系统学习了一个复杂的模式，协调您汽车、车库门、家庭灯光、安全系统和厨房电器之间的操作：

条件（R）：

- 汽车位置接近家中的车道 [来自汽车 GPS 自我快照]
- 时间在 17:30-19:00 之间 [来自智能手机自我快照]
- 星期 = 周一至周五 [来自日历自我快照]
- 家庭安全系统 = 武装 [来自安全系统自我快照]
- 车库门 = 关闭 [来自车库控制器自我快照]
- 之前的位置 = 超市 [来自智能手机 GPS 历史]

操作（A）：

1. 发送指令到车库门：打开
2. 发送指令到安全系统：解除警报并记录 '主人回家'
3. 发送指令到入口灯：调至 100% 开启
4. 发送指令到厨房灯：调至 70% 开启
5. 发送指令到智能烤箱：预热至 180°C（学习到的工作日烹饪偏好）
6. 发送指令到语音助手：宣布 '欢迎回家，烤箱预热中，冰箱整理模式已激活'
7. 发送指令到智能冰箱：切换到 '最近购买的物品' 组织视图
8. 发送指令到智能音箱：以 40% 音量播放 '烹饪播放列表'

钩子观察到您通常去杂货店（根据位置历史），在工作日的晚餐时间回家（时间上下文），并且已经建立了到达后立即烹饪的例行程序。它协调了八个动作，跨越八个不同的设备——您的汽车通过 GPS 位置触发序列，但这些动作影响车库、安全、灯光、烤箱、助手、冰箱和音箱。

示例 3：紧急响应模式

当系统学习识别紧急情况并协调保护响应时，会出现一个更复杂的例子：

条件（R）：

- 烟雾探测器触发 [来自烟雾探测器自我快照]
- 厨房温度迅速上升 [来自恒温器自我快照]
- 炉灶燃烧器活动超过45分钟 [来自智能炉灶自我快照]

- 厨房内20分钟内未检测到运动 [来自运动传感器自我快照]

- 用户位置 = 卧室 [来自智能手表自我快照]

操作 (A):

1. 发送命令给智能炉灶: 紧急关闭所有燃烧器
2. 发送命令给抽油烟机: 以最大速度启动通风
3. 发送命令给所有家居灯光: 调至100% (为撤离提供可见性)
4. 发送命令给智能手机: 播放响亮的警报声, 显示'厨房火灾警报'
5. 向智能手表发送命令: 持续振动并发出火灾警告
6. 向所有智能音箱发送命令: 宣布 '厨房检测到火灾, 炉灶关闭, 如有必要请撤离'
7. 向前门锁发送命令: 解锁以便紧急出口
8. 向智能手机发送命令: 如果用户在60秒内没有回应, 则拨打紧急服务并播放预录信息
9. 向安全摄像头发送命令: 开始录制并保存到云端

这个钩子从一次意外事件中学习, 你不小心把炉子开着, 必须匆忙关闭它, 触发了烟雾警报。系统观察了你的紧急响应模式, 并创建了一个钩子, 现在可以检测类似条件并在九个不同设备之间协调全面的安全响应。

多设备协调的数学:

正式地, 我们可以将这种多设备协调表示为从统一上下文到分布式动作的映射:

$$KH : \text{Context}_t \rightarrow \{(d_1, a_1), (d_2, a_2), \dots, (d_k, a_k)\}$$

其中：

- Context_t = 来自所有设备自我快照的统一上下文
- d_i = 目标设备或身体部位
- a_i = 该设备的动作命令

钩子在整个统一上下文中评估条件：

$$\bigwedge_{i=1}^n r_i(\text{Context}_t) = \text{true} \implies \text{execute}(a_1^{d_1}, a_2^{d_2}, \dots, a_k^{d_k})$$

每个条件 r_i 可以查询统一上下文的任何子集——它可能同时读取您眼镜的摄像头视频、您手机的位置、您智能手表的心率和您家中的温度传感器。然后将动作分发到各自的目标设备：

$$\forall i \in \{1, \dots, k\} : \text{send}(a_i) \rightarrow d_i$$

为什么这种架构强大：

传统的智能家居系统需要明确编程设备交互：'如果车库打开，则打开灯。' 这会创建僵化的、预先确定的关系。主观技术则根本不同：

1. 学习模式：系统观察您在所有设备上的实际行为，并自动提取模式。您从未明确编程“当我从杂货店回家时，预热烤箱”——系统通过观察您这样做来学习。

2. 丰富的上下文条件：钩子可以结合来自不相关设备的数十个上下文信号。“到家”钩子不仅仅基于位置触发——它还考虑时间、星期几、之前的位置（杂货店、健身房或办公室）、当前的家庭状态，甚至历史模式。

3. 自适应优化：随着您的模式变化，钩子会适应。如果您停止在工作日去杂货店并改为周末，钩子的条件自然会演变以匹配您的新模式。

4. 统一智能：与其让每个设备拥有孤立的“智能”功能，不如从统一的上下文中产生智能。您的炉灶不需要单独编程来检测紧急情况——紧急检测钩子协调来自烟雾探测器、温控器、运动传感器和用户位置的信息，以做出明智的决策。

这就是主观关系的本质：设备不再是孤立的工具，而是您身体的协调延伸，预测需求并响应跨越您整个扩展的物理和虚拟自我的复杂模式。统一的上下文是神经系统，而知识钩子是协调这个扩展身体动作的反射和学习行为。

3.2 核心操作

在定义了知识钩的数学结构——4元组 (R, A, T, S) 之后，我们现在转向可以在钩子上执行的操作。这些操作定义了系统的动态：钩子如何活跃，如何执行，如何学习，以及如何相互竞争以获得激活。

知识钩的代数不是静态的。它是一个活生生的系统，钩子不断评估其条件与当前上下文的匹配程度，在适当的时候触发，产生结果，接收反馈，并更新其内部状态。理解这些操作对于理解整个系统如何表现出智能行为至关重要——它如何从经验中学习，适应个体用户，并逐渐减少人类所需的输入。

知识钩的代数中有四个基本操作：激活、执行、学习和优先级排序。每个操作在钩子的生命周期中扮演着不同的角色，它们共同创建了一个完整的反馈循环，使自主学习和持续改进成为可能。

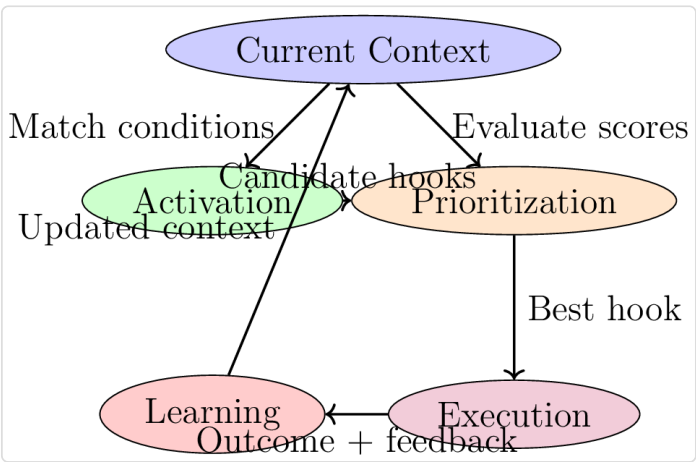
激活决定了钩子何时应该触发——当其条件与当前上下文充分对齐以 warrant 行动的时刻。这是系统的保护条款，确保钩子仅在适当时行动，防止在无关上下文中发生虚假或有害的激活。

执行定义了当钩子触发时发生的事情——其动作如何执行，如何捕捉到结果状态的变化，以及系统如何检测是否需要用户纠正。这是钩子在世界中产生实际效果的地方，自动化任务并最小化用户的努力。

学习描述了如何从用户行为中创建新的钩子，以及现有钩子如何随着时间的推移而改进。这是系统变得主观的机制——独特地适应每个用户的模式、偏好和上下文。

优先级排序解决了在同一上下文中多个钩子可能触发时会发生什么的问题。由于不同的钩子可能具有重叠的条件，系统需要一种原则性的方法来选择竞争选项——优先选择更具体、更可靠和更高效的钩子。

这些操作不是独立的。它们形成一个连续的循环：当前上下文触发候选钩子的激活，优先级排序选择最佳钩子进行执行，执行产生动作并捕捉结果状态，学习更新成功分数并可能基于用户纠正或确认创建新钩子。



这个循环在用户与他们的设备和环境互动时不断重复。每次迭代可能创建新的钩子，加强成功的钩子，削弱失败的钩子，并逐渐减少用户必须提供的明确输入量。

这个系统强大的原因在于这些操作可以同时多个时间尺度上工作。激活和优先级在实时内发生——在上下文变化的毫秒内。执行在几秒到几分钟内进行，随着动作的执行而进行。学习在几天和几周内发生，随着重复交互中模式的出现。而整体向零输入行为的收敛则在几个月内发生，因为系统积累了经验。

这个设计的优雅之处在于它不需要中央规划或明确的训练。用户只是自然地与他们的技术互动，系统观察、学习并改进。纠正是罕见的，并随着时间的推移变得更加罕见。成功不断积累。努力减少。技术变得越来越主观——更符合用户的意图，更能预测他们的需求，更加融入他们的模式。

现在让我们详细检查这些操作，形式化它们的数学，并理解它们对创建真正智能、自适应系统的影响。

3.2.1 激活：何时触发钩子？

激活是决定知识钩子何时执行的基本操作。钩子不会随机或持续触发——它仅在特定的上下文条件满足时激活。这种选择性触发使得智能、上下文感知的行为成为可能，而不是无差别的行动。

激活条件：知识钩子仅在其所有条件在当前上下文中评估为真时触发。正式地，给定一个钩子 $KH = (R, A, T, S)$ ，其条件为 $R = \{r_1, r_2, \dots, r_n\}$ ，以及当前统一上下文 C_t ，当以下条件满足时，钩子激活：

$$\bigwedge_{i=1}^n r_i(C_t) = \text{true} \implies KH \text{ fires and executes } A$$

其中：

- \bigwedge 表示所有条件的逻辑与操作
- 每个 r_i 是一个谓词（布尔函数），用于评估上下文

- C_t 是时间 t 的当前统一上下文（由所有设备自快照形成）

- $A = (a_1, a_2, \dots, a_k)$ 是要执行的动作序列

关键洞察：这是一个与操作，而不是或操作。每个条件都必须满足才能触发钩子。如果即使一个条件评估为假，钩子将保持休眠状态。这确保了精确性——钩子仅在其设计或学习的特定上下文中激活。

与语义 vs 或语义：我们为什么使用与而不是或？考虑一个设计用于发送“迟到”消息的钩子：

使用与语义（正确）：

- $(\text{时间} > \text{会议时间} - 5 \text{ 分钟}) \text{ AND } (\text{距离} > 10 \text{ 分钟旅行}) \text{ AND } (\text{会议已安排}) \rightarrow \text{仅在真正迟到时触发}$

使用或语义（错误）：

- $(\text{时间} > \text{会议时间} - 5 \text{ 分钟}) \text{ OR } (\text{距离} > 10 \text{ 分钟旅行}) \text{ OR } (\text{会议已安排}) \rightarrow \text{如果任何条件为真则触发，即使您没有安排会议或已经在地点}$

与语义防止错误激活，并确保钩子仅在其行为适当的精确情况下触发。

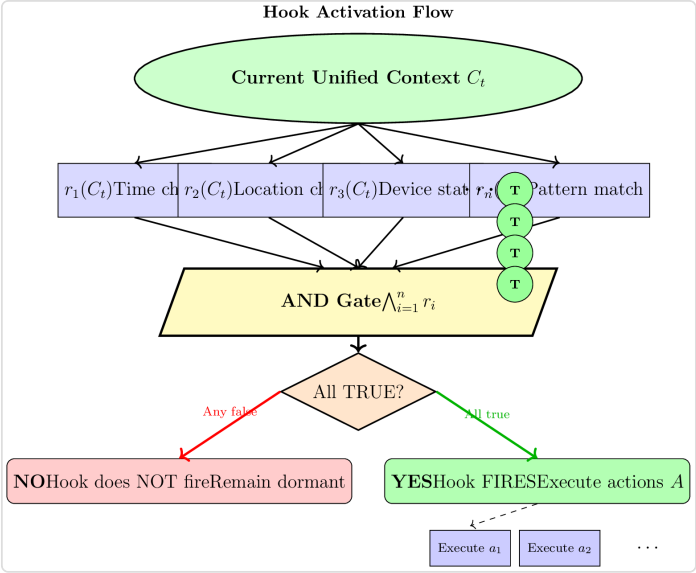


Figure: Hook activation flow. The current unified context is evaluated against all conditions. Only when ALL conditions return true (AND gate) does the hook fire and execute its action sequence.

示例：早晨咖啡钩子激活

考虑一个学习到的钩子，当您醒来时启动咖啡机。钩子通过观察您的早晨例行公事学习了这些条件：

条件 R：

- r_1 ：Time between 6:30 AM and 7:30 AM [from smartphone self-snapshot]
- r_2 ：卧室内检测到运动 [来自运动传感器自我快照]
- r_3 ：智能手机闹钟在过去5分钟内被关闭 [来自手机自我快照]
- r_4 ：咖啡机水箱已满 [来自咖啡机自我快照]

- r_5 : 今天尚未冲泡咖啡[来自咖啡机自我快照]

钩子仅在以下情况下触发：

$$r_1 \wedge r_2 \wedge r_3 \wedge r_4 \wedge r_5 = \text{true}$$

让我们看看不同时间发生了什么：

场景1 - 上午3:00 (未激活)：

- r_1 = false (时间错误)
- r_2 = false (没有运动)
- r_3 = false (警报未解除)
- r_4 = true (水箱已满)
- r_5 = true (咖啡尚未冲泡)

结果： $\text{false} \wedge \text{false} \wedge \text{false} \wedge \text{true} \wedge \text{true} = \text{false} \rightarrow$ 钩子不会触发

场景 2 - 上午 7:00, 仍在床上 (未激活)：

- r_1 = true (正确的时间窗口)
- r_2 = false (尚未检测到运动)
- r_3 = false (警报未解除)
- r_4 = true (水箱已满)
- r_5 = true (咖啡尚未冲泡)

结果： $\text{true} \wedge \text{false} \wedge \text{false} \wedge \text{true} \wedge \text{true} = \text{false} \rightarrow$ 钩子不会触发

场景 3 - 上午 7:05, 刚醒来 (激活!)：

- $T_1 = \text{true}$ (正确时间：上午 7:05)
- $T_2 = \text{true}$ (检测到运动)
- $T_3 = \text{true}$ (警报在 2 分钟前被解除)
- $T_4 = \text{true}$ (水箱已满)
- $T_5 = \text{true}$ (咖啡尚未冲泡)

结果： $\text{true} \wedge \text{true} \wedge \text{true} \wedge \text{true} \wedge \text{true} = \text{true} \rightarrow$ 钩子触发！☕

动作：发送命令到咖啡机：
`brew_coffee(strength='medium',
volume='full_pot')`

场景 4 - 上午 7:10，第二次检查（未激活）：

- $T_1 = \text{true}$ (仍在时间窗口内)
- $T_2 = \text{true}$ (仍在移动)
- $T_3 = \text{true}$ (警报已被解除)
- $T_4 = \text{true}$ (水箱已满)
- $T_5 = \text{false}$ (咖啡已从上上次激活开始冲泡)

结果： $\text{true} \wedge \text{true} \wedge \text{true} \wedge \text{true} \wedge \text{false} = \text{false} \rightarrow$ 钩子不触发（防止重复冲泡）

这个例子展示了 AND 语义的精确性。钩子耐心地等待，持续评估条件，并在恰当的时刻触发——不太早（当你还在睡觉时），不太晚（在你已经手动煮咖啡之后），也不重复（当咖啡已经在煮时）。

持续评估：知识钩子不会闲着。它们持续评估其条件与当前统一上下文的匹配。系统在一个紧密的循环中检查钩子：

```
while true:  $C_t \leftarrow \text{get\_current\_context}()$  for each hook  $KH_i$  in active_hooks: if  $\bigwedge_{j=1}^n r_j(C_t) = \text{true}$ : execute
```

这种持续评估使钩子能够在条件满足时立即响应。当你的闹钟被关闭并检测到运动时，咖啡钩子会在毫秒内触发。

条件类型和复杂性：条件可以从简单到任意复杂：

简单的原子条件：

- T_1 : `time == 14:00`
- T_2 : `battery_level < 20`
- T_3 : `location == 'home'`

范围条件：

- T_4 : `18:00 ≤ time ≤ 22:00`
- T_5 : `15°C ≤ 温度 ≤ 25°C`

模式匹配条件：

- T_6 : `notification_text 匹配正则表达式 '.*meeting.*'`
- T_7 : `image_from_camera 包含 'orange_cat'`

时间条件：

- T_8 : `距离上次动作的时间 > 45 分钟`
- T_9 : `星期几在 {Monday, Tuesday, Wednesday, Thursday, Friday} 中`

聚合条件：

- T_{10} : `最近 10 分钟的平均心率 > 120 BPM`

- r_{11} : 检测到的无运动时间 > 20 分钟

子钩条件：条件可以引用其他钩子，创建层次激活模式：

- r_{12} : sub_hook_'user_leaving_home' 处于活动状态
- r_{13} : sub_hook_'emergency_detected' 在过去 60 秒内触发

这些子钩条件使复杂的组合行为成为可能，高级钩子建立在低级钩子之上，创建智能自动化的层次。

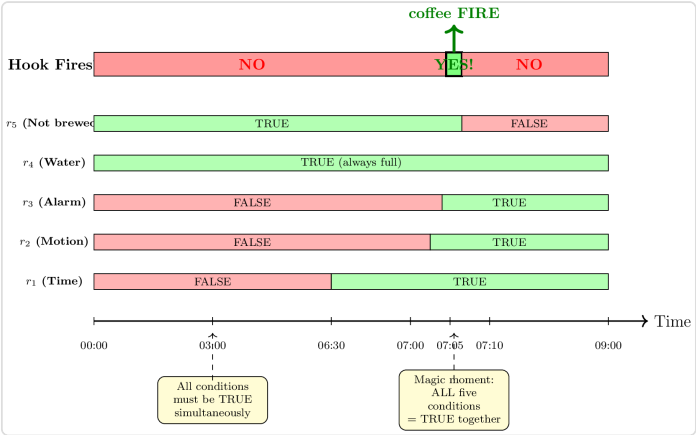


Figure: Timeline visualization of coffee hook activation. Each condition must be true simultaneously. The hook fires only during the narrow window at 7:05 AM when all five conditions align (green bar in bottom row).

效率与优化：持续评估所有钩子的所有条件可能会消耗大量计算资源。实际实现使用几种优化策略：

1. 短路评估：如果任何条件评估为假，则停止评估剩余条件（因为 AND 要求所有条件都为真）

2. 条件排序：优先评估快速且经常为假的条件，以便快速失败

3. 上下文变化触发器：仅在相关的上下文部分变化时重新评估钩子，而不是在每次迭代中

4. 索引：按依赖的上下文元素对钩子进行分组，因此更改位置只会触发依赖于位置的钩子

为什么激活很重要：正确的激活是可靠知识钩子的基础。如果钩子过于急切地触发（当条件未完全满足时），它们会产生不必要的操作并需要用户纠正，从而降低成功率。如果钩子触发过于保守（当条件过于严格时），它们会错失帮助的机会并保持未使用状态。

连续评估的 AND 语义创造了完美的平衡：钩子足够精确以避免错误激活，但又足够灵敏以在需要时恰好在正确的时刻触发。这就是主观技术如何学习成为你的方式——通过理解你希望事情发生的精确上下文，并在这些上下文出现时自主行动。

3.2.2 执行：产生动作和新状态

一旦知识钩子被激活（所有条件满足），它就会进入执行阶段——执行其动作序列并在系统上产生可观察的效果的过程。执行是钩子的智能显现之处：抽象条件和存储的动作转化为现实世界中的具体变化。

执行过程：当钩子 $KH = (R, A, T, S)$ 在时间 t 触发时，执行通过一个明确定义的序列进行：

$$\text{execute}(KH, C_t) \rightarrow (C_{t+1}, \text{Corr}_t)$$

其中：

- C_t 是时间 t 的当前统一上下文（触发激活的上下文）

- C_{t+1} 是执行完成后的新统一上下文
- $\text{Corr}_t \in \{0, 1\}$ 是纠正标志：如果用户接受结果则为 0，如果用户纠正则为 1

执行产生三个关键输出：

1. 执行的动作本身：序列 $A = (a_1, a_2, \dots, a_k)$ 按顺序执行
2. 新的系统状态：所有设备更新其自我快照，形成新的统一上下文 C_{t+1}
3. 修正标志：系统观察用户是否干预以修改、撤销或纠正钩子的操作

顺序动作执行：动作序列 A 是有序且确定的。每个动作 a_i 按顺序执行，每个动作可能依赖于前一个动作的结果：

$$\text{execute}(A) = a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_k$$

更正式地说，我们可以将执行表示为状态转换函数。从上下文 C_t 开始，每个动作转换上下文：

$$C_t \xrightarrow{a_1} C_t^{(1)} \xrightarrow{a_2} C_t^{(2)} \xrightarrow{\dots} C_t^{(k)} = C_{t+1}$$

其中 $C_t^{(i)}$ 表示执行前 i 个动作后的中间上下文状态。

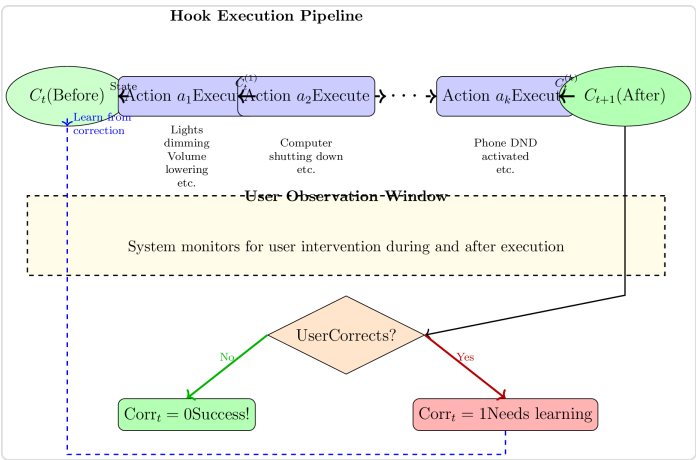


Figure: Execution pipeline showing sequential action execution from initial context C_t to final context C_{t+1} , with user observation determining the correction flag.

示例：晚间放松执行

让我们追踪一下我们之前讨论的晚间放松钩子的具体执行。
 在 $t = 21:30$ 时，所有条件都满足，钩子触发，动作序列 $A = (a_1, a_2, a_3, a_4, a_5, a_6)$ ：

初始状态 C_t ($t = 21:30:00$)：

- 客厅灯光：100% 亮度
- 工作电脑：活跃，运行 3 个应用程序
- 手机：正常模式，通知已启用
- 智能音箱：待机
- 温控器：22°C
- 智能手表：正常模式

执行顺序：

a_1 ：发送命令给智能灯：调暗至30%

→ $C_t^{(1)}$ ($t = 21:30:01$)：灯光开始调暗（过渡大约需要2秒）

a_2 ：发送命令给工作电脑：启动关机，保存文档

→ $C_t^{(2)}$ ($t = 21:30:02$)：电脑保存打开的文档，开始关机序列

a_3 ：发送命令给智能手机：启用请勿打扰模式

→ $C_t^{(3)}$ ($t = 21:30:03$)：手机进入DND，通知被静音

a_4 ：发送命令到智能音箱：以15%的音量播放《晚间冥想》

→ $C_t^{(4)}$ ($t = 21:30:04$)：音箱开始播放冥想音频

a_5 ：发送命令到恒温器：调低到19°C

→ $C_t^{(5)}$ ($t = 21:30:05$)：恒温器调整目标温度

a_6 ：发送命令到智能手表：激活睡眠跟踪模式

→ $C_t^{(6)} = C_{t+1}$ ($t = 21:30:06$)：智能手表进入睡眠跟踪

最终状态 C_{t+1} ($t = 21:30:06$)：

- 客厅灯光：30%亮度（调暗）
- 工作电脑：正在关机
- 手机：请勿打扰模式已激活
- 智能音箱：以低音量播放冥想音频
- 温控器：降温至19°C
- 智能手表：睡眠追踪已激活

观察窗口：从 $t = 21:30:06$ 开始，系统观察用户行为。如果用户接受这些更改（没有调整任何内容），则 $\text{Corr}_t = 0$ 。如果用户立即将灯光调回或重新启用通知，则 $\text{Corr}_t = 1$ 。

修正标志：修正标志 Corr_t 可能是执行的最重要输出，因为它决定了钩子的成功评分是增加还是减少。系统通过几种机制检测修正：

直接撤销：用户立即撤销一个或多个操作（重新打开灯光，重新打开电脑，禁用勿扰模式）

修改：用户调整操作的参数（将灯光亮度从30%更改为50%，将温控器从19°C更改为20°C）

明确修正：用户通过界面提供明确反馈（“这个钩子不正确”或类似内容）

时间窗口：系统通常在一个时间窗口内观察修正（例如，执行后5分钟）。在此窗口外进行的操作被视为新的用户意图，而不是修正。

正式地，我们可以将修正检测表示为：

$$\text{Corr}_t = \begin{cases} 1 & \text{if user modifies } C_{t+1} \text{ within time window } \Delta t \\ 0 & \text{otherwise} \end{cases}$$

状态捕获：执行的一个关键方面是系统在执行前后捕获完整的快照。这使得几个重要功能成为可能：

1. 回滚：如果一个钩子产生了不良影响，系统可以从捕获的快照中恢复 C_t

2. 学习：增量 $\Delta = C_{t+1} - C_t$ 准确显示了发生了什么变化，从而使钩子得以改进

3. 审计：前后快照提供了钩子所做事情的完整记录

4. 调试：当钩子出现故障时，开发人员可以检查确切的上下文和结果变化

执行记录可以形式化为一个元组：

$$\text{ExecutionRecord} = (KH, C_t, C_{t+1}, A_{\text{executed}}, \text{Corr}_t, t)$$

其中：

- KH 是触发的钩子
- C_t 是执行前的上下文
- C_{t+1} 是执行后的上下文
- A_{executed} 是实际执行的动作序列
- Corr_t 是否发生了更正
- t 是执行的时间戳

原子执行与事务执行：不同的钩子可能需要不同的执行保证：

原子执行：序列中的每个动作都是独立的。如果动作 a_3 失败， a_4 仍然尝试执行。当动作彼此无关时（调暗灯光、关闭应用程序、发送消息），这是合适的。

事务执行：动作形成一个逻辑单元。如果任何动作失败，所有有先前的动作都会回滚。当动作必须一起成功时（预订航班、预定酒店、租车——如果任何一个失败，取消所有），这是合适的。

大多数知识钩子使用原子执行，因为动作通常针对独立设备。一个设备的失败（例如，智能灯离线）不应阻止其他动作（例如，计算机关机）成功。

执行期间的错误处理：执行可能会遇到错误——设备离线、网络超时、权限被拒绝等。系统优雅地处理错误：

$$\text{execute}(a_i) = \begin{cases} \text{success} & \text{if action completes normally} \\ \text{error}(e) & \text{if action fails with error } e \end{cases}$$

当一个动作失败时：

1. 错误记录在执行记录中
2. 后续动作继续（原子执行）或中止（事务执行）
3. 钩子的成功评分可能会根据错误严重性受到惩罚
4. 如果错误影响关键功能，用户可能会收到通知

示例：如果晚间放松钩子尝试调暗智能灯光但它们离线，钩子仍然会关闭计算机，启用勿扰模式，并执行其他操作。灯光将保持当前亮度，错误将被记录。

执行和成功评分更新：执行的最重要后果是钩子的成功评分更新。这发生在观察窗口关闭后：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

其中：

- $S(t)$ 是此执行之前的当前成功评分
- $\alpha \in (0, 1]$ 是学习率
- $\mathbb{I}[\text{Corr}_t = 0]$ 如果没有发生更正则为1，如果发生更正则为0
- $S(t+1)$ 是更新后的成功评分

这形成了一个反馈循环：成功的执行（没有更正）会提高评分，而失败的执行（需要更正）则会降低评分。随着时间的推移，持续产生正确结果的钩子会获得高分，并被信任自动执行，而频繁需要更正的钩子则会看到其评分下降，最终可能被禁用。

因此，执行操作不仅仅是执行动作——它是知识钩子学习、适应和进化的机制。每次执行既是一次表现（产生现实世界的效果），也是一次实验（测试钩子对用户意图的模型是否保持准确）。这种执行、观察和学习的持续循环使得主观技术能够随着时间的推移逐渐与用户需求更加一致。

3.2.3 学习步骤：上下文减法和增量检测

学习步骤是知识钩子从用户行为中产生的机制。与激活和执行（作用于现有钩子）不同，学习步骤通过观察用户行为并从中提取模式来创建新的钩子。这就是主观技术如何独特地适应每个个体，学习个人偏好而无需明确编程。

核心原则：上下文减法。当用户提供输入——在系统内采取的任何明确行动——学习机制捕获两个完整的统一上下文快照：

Σ_{before} = Context immediately before user input

Σ_{after} = Context immediately after user input completes

这些快照之间的差异——增量或上下文减法——准确揭示了用户行为所导致的变化：

$$\Delta = \Sigma_{\text{after}} - \Sigma_{\text{before}}$$

这个增量 Δ 不仅仅是变化的日志。它是用户创建的转变的结构化表示，捕获了变化的内容和每个变化的幅度。增量成为新学习的知识钩子生长的种子。

从增量到钩子：学习算法从之前的快照、增量和当前系统状态中提取候选知识钩子：

$$KH_{\text{learned}} = \text{extract}(\Sigma_{\text{before}}, \Delta, \text{context})$$

更具体地说，新钩子的构造方式为：

$$KH_{\text{new}} = (R_{\text{extracted}}, A_{\text{extracted}}, \text{learned}, S_0)$$

其中：

- $R_{\text{extracted}}$ = 从 Σ_{before} 中提取的条件（用户决定采取行动时存在的模式）

- $A_{\text{extracted}}$ = 从 Δ 中提取的动作（用户产生的变化）

- 类型 = 学习的（与预定义钩子区分开）

- S_0 = 初始成功分数（通常为0.5，表示未知的可靠性）

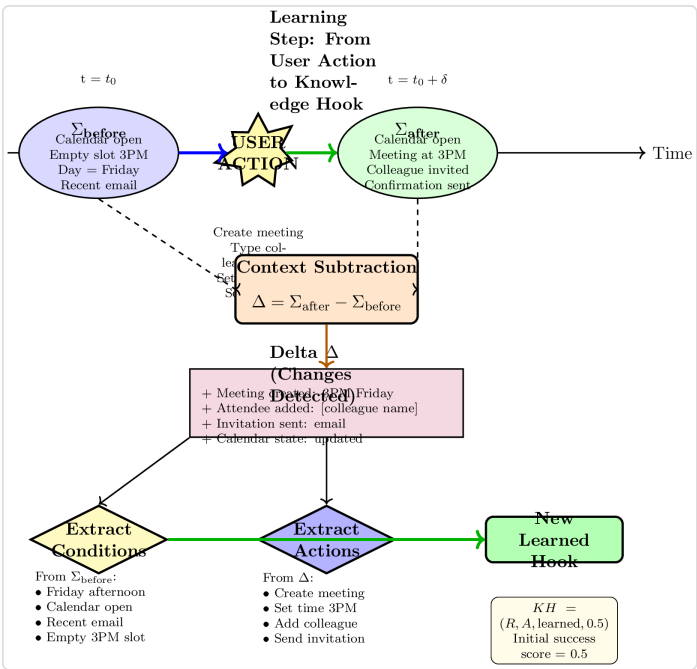


Figure: The learning step process. User action transforms context from Σ_{before} to Σ_{after} . Context subtraction reveals the delta Δ . Conditions are extracted from Σ_{before} , actions from Δ , creating a new learned hook.

示例：学习日历调度钩子

让我们通过一个具体的例子来追踪学习步骤的实际操作。你每周五下午3点与同事安排一次会议。第一次手动执行此操作时，系统会观察并学习。

第一次星期五 - $t_0 = 14:45$ (下午2:45)

在 Σ_{before} 之前的上下文：

- 时间：14:45，日期：星期五
- 应用程序：日历（活动窗口）

- 日历状态：星期五下午3点的时间段为空
- 最近活动：10分钟前收到同事的电子邮件
- 电子邮件内容：提到每周同步，引用星期五
- 位置：办公室
- 之前的模式：此时没有定期会议

用户操作 (t_o 到 t_o+30s) :

1. 点击 "新事件" 按钮
2. 输入标题："与 Alex 的每周同步"
3. 设置时间：星期五 下午 3:00 - 3:30
4. 添加与会者：alex@company.com
5. 设置重复：每周
6. 点击 "发送邀请"

上下文在 Σ_{after} 之后 ($t_o+30s = 14:45:30$) :

- 时间：14:45:30，日期：星期五
- 应用程序：日历（仍在活动中）
- 日历状态：已创建重复会议，星期五下午3点已占用
- 会议详情：标题 "与 Alex 的每周同步"，与会者 alex@company.com
- 邀请状态：已发送（电子邮件通知已发送）
- 最近活动：10分钟前的相同电子邮件（现已处理）
- 位置：办公室（未更改）

上下文减法：

$$\Delta = \Sigma_{\text{after}} - \Sigma_{\text{before}}$$

检测到的更改：

- +日历事件："与Alex的每周同步"
- +时间段：星期五 3:00-3:30 PM
- +重复：每周
- +与会者：alex@company.com
- +邮件邀请：已发送
- 日历状态变化：空 → 占用

钩子提取：

条件 R (从 Σ_{before} 提取)：

- r_1 : 星期几 = 星期五
- r_2 : 时间 $\in [14:30, 15:00]$ (下午时段)
- r_3 : 活跃应用 = 日历
- r_4 : `calendar_slot[Friday][15:00] = empty`
- r_5 : `recent_email_from = alex@company.com`
- r_6 : `email_content matches (weekly|sync|Friday)`

Actions A (extracted from Δ):

- a_1 : create_calendar_event(title="Weekly Sync with Alex")

- a_2 : set_time(day=Friday, start=15:00, duration=30min)

- a_3 : set_recurrence(frequency=weekly)

- a_4 : add_attendee(alex@company.com)

- a_5 : send_invitation()

Type: T = learned

初始成功分数 : $S_0 = 0.5$ (中性, 未经验证)

完整的学习钩子 :

$KH_{\text{Friday_Meeting}} = (\{r_1, r_2, r_3, r_4, r_5, r_6\}, (a_1, a_2, a_3, a_4, a_5), \text{learned}, 0.5)$

模式识别与概括：初始学习的钩子通常过于具体——它捕捉了特定实例的过多细节。系统采用模式识别来适当地概括：

过于具体的条件（初始学习）：

- 点击时鼠标光标位置为 (x, y)
- 窗口的确切大小和位置
- 电池电量, WiFi信号强度
- 环境噪声水平

概括后的条件（修剪后）：

- 星期几 = 星期五（相关）
- 时间窗口 2-3 PM（相关）
- 最近的同事电子邮件（相关）

- 日历应用程序已打开（相关）

系统使用几种启发式方法来确定哪些条件是显著的：

1. 相关性评分：在不同用户的成功钩子激活中出现的条件权重更高
2. 方差分析：在同一操作的多个实例中变化较大的条件被修剪
3. 领域知识：预定义规则指示哪些类型的条件通常是相关的（时间、地点、应用状态）与不相关的（光标位置、电池电量）
4. 用户反馈：当钩子触发并被纠正时，系统分析哪些条件被违反，以确定哪些实际上是重要的

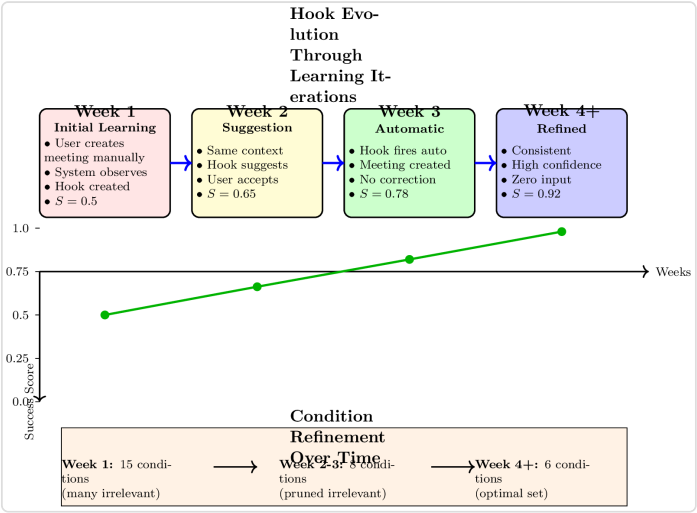


Figure: Hook evolution over multiple learning iterations. Initial hook has low success score (0.5) and many conditions. Through repeated successful activations, success score increases and conditions are refined to optimal set.

合并相似钩子：当用户在不同上下文中重复类似操作时，系统可能会学习多个相似的钩子。这些是合并的候选者：

示例：经过几周后，您可能已经学习了单独的钩子：

- KH_1 ：在星期五下午3点与Alex安排会议（创建于第1周）
- KH_2 ：当电子邮件提到"同步"时，在星期五下午3点与Alex安排会议（创建于第3周）
- KH_3 ：从办公室在星期五下午3点与Alex安排会议（创建于第5周）

这些钩子有显著的重叠。系统可以将它们合并为一个更强大的钩子：

$$KH_{\text{merged}} = \text{merge}(KH_1, KH_2, KH_3)$$

合并后的钩子：

- 采取条件的并集或交集（取决于策略）
- 组合动作序列（通常相同，因此不需要更改）
- 按使用频率加权的成功分数平均值：

$$S_{\text{merged}} = \frac{\sum_i S_i \cdot n_i}{\sum_i n_i}$$

其中 n_i 是钩子 KH_i 成功触发的次数。

增量表示：增量 Δ 必须捕捉关于变化的丰富结构信息。简单的差异是不够的。相反，增量表示为一组结构化的修改：

$$\Delta = \{(\text{entity}, \text{property}, \text{old_value}, \text{new_value}, \text{operation})\}$$

日历会议的示例增量：

- (Calendar, Events, [], [Meeting{title="Weekly Sync", ...}], ADD)
- (日历, 星期五下午3点, "空", "占用", 修改)
- (电子邮件, 已发送项目, previous_list, previous_list + [邀请{...}], 附加)

这种结构化表示使得精确的操作提取成为可能，并使我们能够理解不仅是什么改变了，还有如何和为什么改变。

学习率和阈值：并非每个用户操作都会立即创建新的钩子。系统应用阈值以防止钩子泛滥：

1. 重要性阈值：仅为涉及重大变化（>5个修改实体）的操作创建钩子
2. 重复阈值：在执行相同操作2-3次后再创建钩子
3. 相似性检查：在创建新钩子之前，检查是否已经存在相似的钩子，并更新它
4. 用户意图检测：使用启发式方法区分有意图的模式和随机的一次性操作

学习步骤因此不仅仅是一个简单的记录机制——它是一个智能模式提取系统，观察用户行为，识别重复模式，过滤噪音，并将学习到的知识结晶为可执行的钩子。这是主观技术从通用工具演变为个性化自我扩展的机制，学习预测需求并通过持续观察和适应来最小化能量消耗。

3.2.4 组合：从简单钩子构建复杂行为

组合是允许知识钩子组合成更大、更复杂行为的操作。单个钩子功能强大，但真正的魔力在于钩子的组合——当简单的原子行为连接并协调以产生复杂的智能系统时。组合将一系列孤立的自动化转变为一个连贯的、自适应的智能。

知识钩子的代数定义了两种主要的组合形式：

1. 嵌套组合（分层）：条件和动作本身可以包含其他钩子，从而创建行为的层次结构
2. 扁平组合（连接）：多个钩子可以通过连接它们的条件和动作来组合，从而创建一个更大的统一钩子

这两种形式使知识钩子能够表达任意的计算复杂性，同时保持模块化和可重用性。

嵌套组合：钩子中的钩子。最强大的组合形式允许钩子在其条件或动作中引用其他钩子。这创建了一个层次结构，其中高层抽象是由低层原语构建的。

形式上，一个钩子 KH_i 可以通过两种方式包含对其他钩子 KH_j 的引用：

条件中的子钩子：一个条件可以检查另一个钩子是否会触发或最近是否触发过：

$$R_i = \{r_1, r_2, \dots, r_k, \text{fires}(KH_j, C_t), \dots, r_n\}$$

在当前上下文 C_t 中，如果钩子 KH_j 会激活，则 $\text{fires}(KH_j, C_t)$ 评估为真。

动作中的子钩子：一个动作可以调用另一个钩子的执行：

$$A_i = (a_1, a_2, \dots, \text{execute}(KH_j), \dots, a_k)$$

在动作序列中， $\text{execute}(KH_j)$ 触发钩子 KH_j 。

这种嵌套允许强大的抽象。一个高层次的钩子，比如“准备房子离开”，可以调用子钩子来“关闭灯光”、“锁门”、“启动安全系统”，每个子钩子可能又会调用自己针对单个房间或设备的子钩子。

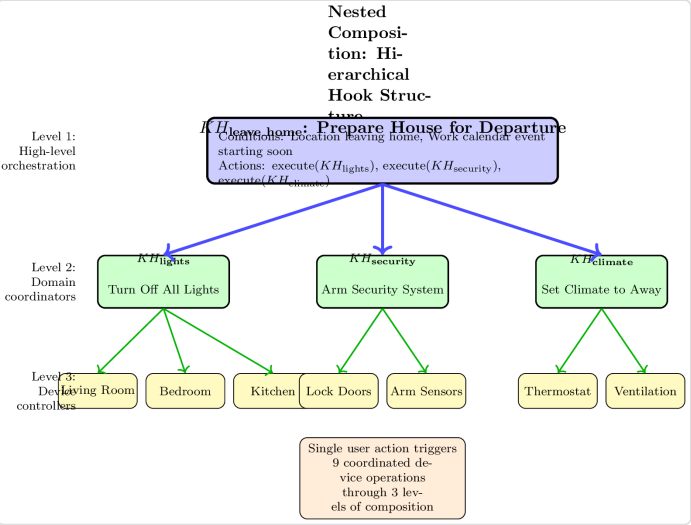


Figure: Nested composition hierarchy. A single high-level hook 'leave_home' executes three mid-level coordinator hooks, which in turn execute nine device-level hooks. Composition creates modularity and reusability.

示例：嵌套的早晨例程。考虑一个学习到的钩子用于你的早晨例程。它可以从现有的专用钩子中组合，而不是硬编码每一个动作：

$KH_{\text{morning_routine}}$:

条件 R :

- 起床时间（闹钟解除）
- 今天是工作日
- 位置=家

操作 A :

1. 执行($KH_{\text{bedroom_lights}}$) → 将卧室灯光逐渐调至80%
2. 执行($KH_{\text{coffee_maker}}$) → 开始冲泡咖啡
3. 执行($KH_{\text{weather_briefing}}$) → 通过扬声器播报今天的天气
4. 执行($KH_{\text{calendar_sync}}$) → 在浴室镜子上显示日历
5. 执行($KH_{\text{news_summary}}$) → 早餐时播放新闻播客

每个子钩子都是独立有用的，可以在其他上下文中重用。卧室灯光钩子也可以通过晚间阅读例程调用。咖啡机钩子可能在客人到达时触发。这种模块化意味着钩子不需要被重新创建——它们可以组合。

平面组合：连接钩子。第二种组合形式通过合并其组件来组合多个钩子。给定两个钩子

$KH_1 = (R_1, A_1, T_1, S_1)$ 和 $KH_2 = (R_2, A_2, T_2, S_2)$ ，平面组合创建一个新钩子：

$$KH_{\text{combined}} = KH_1 \oplus KH_2 = (R_1 \cup R_2, A_1 \parallel A_2, T_{\text{derived}}, S_{\text{combined}})$$

其中：

- $R_1 \cup R_2$ 是条件的并集（两个钩子的所有条件必须满足）
- $A_1 \parallel A_2$ 是动作序列的连接（执行 A_1 然后 A_2 ）
- T_{derived} 通常被 "学习" 用于组合钩子
- S_{combined} 是从组件成功分数计算得出的（通常是最小值或平均值）

扁平组合在您想要合并相似钩子或从独立操作创建复合行为时非常有用。

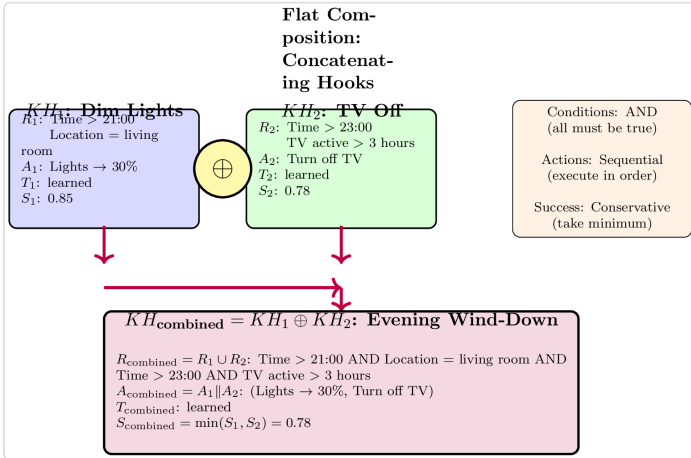


Figure: Flat composition combines two hooks by taking the union of conditions (AND semantics), concatenating actions sequentially, and computing a combined success score (often minimum to be conservative).

示例：合并晚间钩子。假设您有两个独立的晚间钩子：

$KH_{\text{dim_lights}}$: 当时间 > 21:00 且您在客厅
时 → 将灯光调暗至 30%

$KH_{\text{silence_phone}}$: 当时间 > 21:30 且您在家时 → 启用勿扰模式

这些可以组合成一个统一的晚间例行钩子：

$$KH_{\text{evening}} = KH_{\text{dim_lights}} \oplus KH_{\text{silence_phone}}$$

结果钩子在所有条件满足时触发（时间 > 21:30，客厅，在家）并按顺序执行两个操作（调暗灯光，然后启用 DND）。

级联：钩子触发钩子。当一个钩子的执行创建上下文变化以触发其他钩子时，就会发生一种特殊的组合情况。这就是级联——钩子激活的连锁反应。

正式来说，如果钩子 KH_1 级联到钩子 KH_2 ，则：

$$\text{execute}(KH_1, C_t) \rightarrow C_{t+1} \text{ such that } \bigwedge_i r_i^{(2)}(C_{t+1}) = \text{true}$$

换句话说，执行 KH_1 会以满足 KH_2 条件的方式修改上下文，从而导致 KH_2 被触发。

示例级联：

1. $KH_{\text{leave_home}}$ 被触发 → 将位置设置为 "离开"

2. 上下文变化（位置 = 离开）触发 KH_{security} → 启动安全系统

3. 上下文变化 (安全 = 已启动) 触发 $KH_{\text{camera_record}}$ → 激活摄像头

4. 上下文变化 (摄像头 = 活动) 触发 KH_{notify}
→ 向手机发送确认

每个钩子根据前一个钩子创建的上下文变化独立触发，形成优雅的自动行为级联。

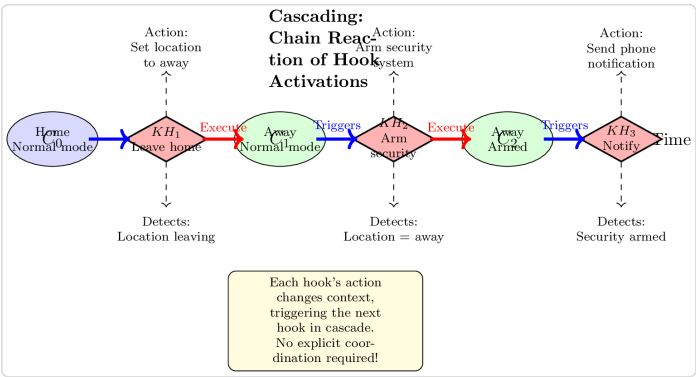


Figure: Cascading activation chain. Hook KH_1 executes and changes context from C_0 to C_1 . This new context satisfies KH_2 's conditions, causing it to fire. KH_2 's execution creates C_2 , triggering KH_3 . Each hook operates independently, yet they coordinate through shared context.

防止无限循环：级联引入了无限循环的风险——钩子 A 触发钩子 B，钩子 B 又触发钩子 A，循环不止。系统通过几种机制防止这种情况：

- 1. 激活历史：跟踪当前级联链中已触发的钩子。防止重新激活已经在链中的钩子。
- 2. 深度限制：将级联深度限制为最大值（例如，10 层）。达到限制后，断开链。

3. 基于时间的冷却：一旦钩子触发，施加一个最小时间间隔，才能再次触发（例如，1秒）。

4. 上下文稳定性：仅在上下文真正发生变化时触发级联，而不是在状态之间振荡时。

组合属性：组合操作满足几个代数属性：

结合性（平面组合）：

$$(KH_1 \oplus KH_2) \oplus KH_3 = KH_1 \oplus (KH_2 \oplus KH_3)$$

组合钩子的顺序对最终结果没有影响（尽管操作的执行顺序确实重要）。

恒等性：存在一个恒等钩子 $KH_\emptyset = (\emptyset, (), \text{predefined}, 1.0)$ ，使得：

$$KH \oplus KH_\emptyset = KH$$

与空钩子组合不会改变原始钩子。

模块化：嵌套组合保持模块化——更改子钩子会影响所有引用它的钩子，从而使单一更改能够进行系统范围的更新。

可组合性法则：知识钩子代数的基本法则之一指出：

"钩子可以通过组合操作组合成更大的结构，结果复合钩子的行为基于其组成部分可预测。"

这条法律保证了组合是一个安全的操作——组合良好的钩子会产生良好的复合钩子。系统不会因为组合而产生突发故障；它构建的是突发智能。

组合的实际好处：

1. 可重用性：编写一次钩子，通过组合在多个上下文中使用

2. 可维护性：更新一个低级钩子，所有使用它的复合钩子都会自动受益

3. 抽象：高级钩子可以表达意图，而无需指定实现细节

4. 可扩展性：复杂行为从简单组件中产生，而不会导致指数级复杂性

5. 学习能力：系统可以通过组合先前学习的低级行为来学习高级模式

组合将知识钩子从孤立的自动化转变为真正的协调智能生态系统。通过组合，简单的反应行为演变为复杂的自适应系统，展现出看似智能的预见和协调——这一切都源于钩子组合的数学代数。

3.2.5 钩子优先级：在多个候选者中进行选择

优先级是解决知识钩子系统中的一个基本挑战的操作：当多个钩子可以在同一上下文中触发时会发生什么？由于钩子具有可能重叠的概率条件，多个钩子同时匹配当前上下文是很常见的。系统需要一种原则性、数学上严谨的方法来选择哪个钩子应该执行。

这不仅仅是一个技术细节——它是主观技术整个哲学的核心。错误的优先级策略可能导致低效的行动、浪费的能源或用户的挫败感。正确的策略确保系统始终选择最有效、最可靠和最符合上下文的行动。

竞争的挑战：考虑一个简单的场景。你有两个钩子，当你晚上回家时，它们都可能被触发：

```
Hook A (General): Conditions = {time: after 6pm, location: home} → Actions = {turn on
```

living room lights}

Hook B (Specific): Conditions = {time: after 6pm, location: home, day: Friday} → Actions = {turn on living room lights, start music, set temperature to 22°C}

在星期五晚上你回家时，两个钩子的条件都满足。应该触发哪个？钩子 A 更简单，最初需要的用户输入更少，但钩子 B 更具体于当前上下文，并且执行更完整的自动化。选择错误意味着要么自动化不足（钩子 A），要么在非星期五的晚上应用不适当的自动化（钩子 B）。

当考虑具有不同成功评分、不同数量动作、来自不同来源（预定义与学习）的钩子，以及具有部分重叠条件的钩子时，问题变得更加复杂。

优先级算法：知识钩子框架采用基于最小化法则和经验可靠性的两阶段优先级策略：

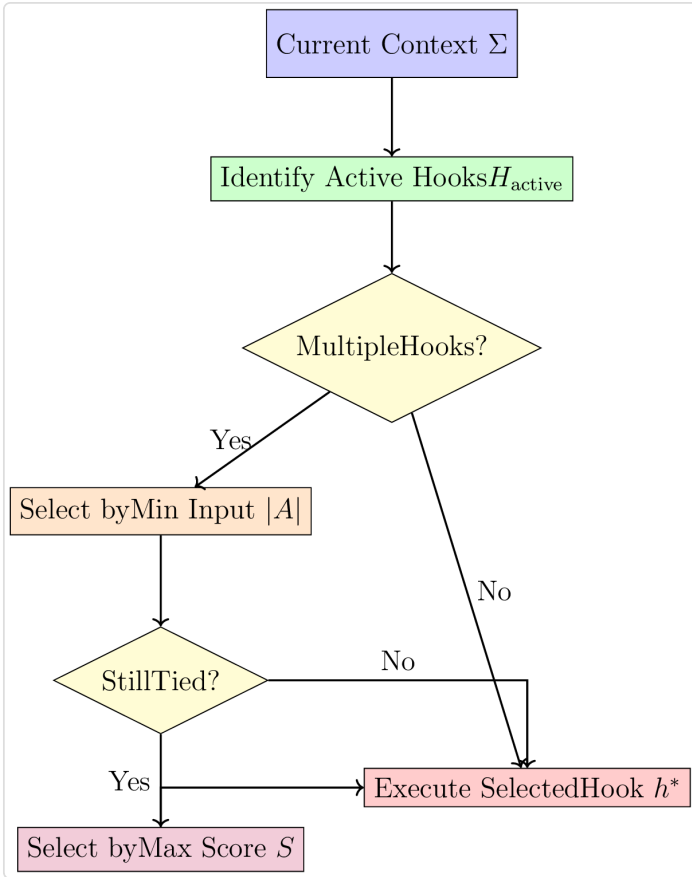
$$\text{Select } h^* = \arg \min_{h \in H_{\text{active}}} |A_h|$$

其中 H_{active} 是当前与上下文匹配的所有钩子的集合。第一阶段选择需要最少用户输入的钩子——那些具有最小动作基数 $|A|$ 的钩子。这直接实现了最小化法则：优先选择需要最少用户努力的路径。

如果多个钩子在最小输入上平局（当钩子具有相同结果时，这种情况很常见），我们使用成功评分来打破平局：

$$\text{If } |A_{h_1}| = |A_{h_2}| = \dots = |A_{h_k}|, \text{ then select } h^* = \arg \max_{h \in H_{\text{tied}}} S_h$$

在需要相等用户输入的钩子中，选择成功评分最高的那个——历史上最可靠且需要最少修正的那个。



特异性和精细化：在优先级排序中，一个重要的考虑因素是钩子的特异性。如果条件 R_1 的钩子比条件 R_2 的钩子更具体，则 $R_1 \subset R_2$ ——也就是说，如果第一个钩子的条件是第二个钩子的一个子集，则需要额外的约束来激活。

更具体的钩子通常应该优先于更一般的钩子，即使它们的成功评分相似。这是因为具体的钩子捕捉到更多的上下文细微差别，更有可能在特定情况下匹配用户意图。

Specificity ordering: $R_1 \prec R_2$ if $R_1 \subset R_2$

例如，条件为 `{time: 7am, day: Monday, location: home, calendar: 'morning meeting'}` 的钩子比仅有 `{time: 7am, location: home}` 的钩子更具体。如果两个钩子都被激活，更具体的钩子应该执行，因为它捕捉到了更精确的上下文。

扩展示例：早晨例行优先级解决。考虑在星期一早上 7:05 AM 的三个竞争钩子：

钩子₁（一般早晨）： $R = \{time: 7:00-7:30, location: home\}$ ， $A = \{start\ coffee\ maker\}$ ， $S = 0.92$

钩子₂（工作日早晨）： $R = \{time: 7:00-7:30, location: home, day: Monday-Friday\}$ ， $A = \{start\ coffee\ maker, turn\ on\ news\}$ ， $S = 0.88$

钩子₃（星期一特定）： $R = \{time: 7:00-7:30, location: home, day: Monday, calendar: contains\ 'meeting'\}$ ， $A = \{start\ coffee\ maker, turn\ on\ news, prepare\ briefing\ doc\}$ ， $S = 0.85$

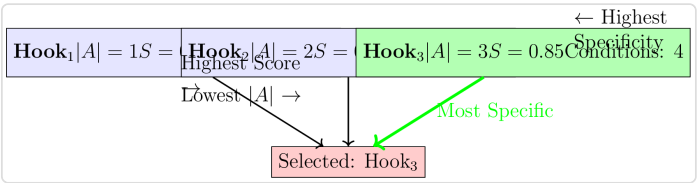
所有三个钩子都被激活。应用优先级算法：

阶段 1（最小化输入）：钩子₁有 $|A| = 1$ ，钩子₂有 $|A| = 2$ ，钩子₃有 $|A| = 3$ 。根据最小化法则，我们最初会偏好钩子₁。

然而，这就是特异性重要的地方。钩子₃是最具体的——它捕捉到最多的上下文约束。在一个有会议安排的星期一早晨，钩子₃的额外动作（准备简报文件）代表了真正的价值，而不是不必要的工作。系统认识到在这种情况下更高的 $|A|$ 反映了更完整的自动化，而不是浪费的努力。

精细的优先级考虑了输入最小化和特异性加权。尽管 Hook₃ 的动作更多，但由于其条件与当前上下文最为特定，因

此被选中。其稍低的成功评分（0.85 对 0.92）在显著的特异性优势下是可以接受的。



额外的优先因素：虽然核心算法通过最小输入和最大成功评分进行优先级排序，但实际实现可能会结合额外因素以实现更细致的优先级排序：

1. 关键性：安全关键的钩子（火警、健康紧急情况）始终优先于其他因素。
2. 时间敏感性：具有时间紧迫性的钩子（即将到期的限时优惠）会获得优先提升。
3. 近距离：与相关对象的物理接近性会增加优先级（需要附近资源的钩子优先于需要远程资源的钩子）。
4. 资源可用性：需要当前可用资源的钩子优先于需要不可用资源的钩子。
5. 用户偏好：用户可以手动调整钩子的优先级，表达对某些自动化模式的偏好。
6. 历史相关性：与频繁出现的模式相关的钩子会获得轻微的优先级提升。
7. 并发约束：在多用户环境中，优先考虑那些不垄断共享资源的钩子。

这些因素可以组合成一个复合优先级评分：

$$\text{Priority}(h) = w_1 \cdot f_{\text{input}}(|A_h|) + w_2 \cdot S_h + w_3 \cdot f_{\text{specificity}}(|R_h|) + w_4 \cdot f_{\text{criticality}}(h) + \dots$$

每个 w_i 是一个权重，决定该因素的重要性，每个 f 代表一个评分函数，将该因素标准化到可比较的尺度。

最终选择：在优先级排序完成后，精确选择一个钩子进行执行。这个确定性的选择至关重要——系统不能在平局选项中随机选择或同时执行多个钩子（除非明确设计为组合）。随机选择会导致不可预测的行为，并阻止系统学习可靠的模式。

如果优先级算法仍然导致完全平局（相同的输入大小、相同的成功分数、相同的特异性），系统可以使用最终的决定性平局打破器，例如钩子创建时间戳（优先考虑较新的钩子，可能反映更近期的学习）或钩子标识符（按字典顺序排序以实现完全确定性）。

为什么优先级排序很重要：有效的优先级排序将一组独立的钩子转变为一个智能、连贯的系统。如果没有它，钩子将会混乱地竞争，以随机或次优的顺序触发，浪费能量并使用户感到沮丧。通过适当的优先级排序，系统表现出新兴的智能行为——始终选择最符合上下文、最可靠和最有效的可用行动。

优先级排序还提供了一个机制，通过该机制，学习到的钩子可以逐渐替代或完善预定义的钩子。随着用户的个人钩子积累经验并提高成功分数，它们可以在优先级上超越通用的预定义钩子，使系统随着时间的推移变得越来越个性化和主观。

这就是自适应自动化的本质：系统不仅仅执行钩子——它持续评估哪些钩子在特定用户和特定上下文中效果最佳，并智能地选择最佳行动。通过数千次优先级排序决策，系统在数周和数月内学会成为你。

3.2.6 回滚：撤销操作和恢复状态

回滚是允许系统撤销知识钩子执行效果的操作，将上下文恢复到钩子触发前的状态。这个能力对于处理错误、用户修正和意外后果至关重要。当钩子产生不良效果时——无论是由于条件不正确、情况变化还是简单错误——回滚提供了恢复的途径。

回滚操作是通过执行期间发生的全面状态捕获来启用的。在任何钩子触发之前，系统保存当前上下文的完整快照 C_t 。如果需要回滚，这个快照提供了要恢复的目标状态。

回滚操作：正式地，回滚试图逆转钩子的执行，将当前上下文 C_{t+1} 转换回执行前的上下文 C_t ：

$$\text{rollback}(KH, C_t, C_{t+1}) \rightarrow C_t \text{ (if possible)}$$

其中：

- KH 是正在回滚执行的钩子
- C_t 是执行前捕获的快照
- C_{t+1} 是钩子执行后的当前上下文
- 操作尝试将系统恢复到状态 C_t

然而，回滚并不总是可能或完全。回滚的可行性在很大程度上取决于执行的操作的性质。

可逆与不可逆操作：操作可以根据其可逆性进行分类：

完全可逆操作：这些操作可以完美撤销，将系统恢复到完全相同的先前状态。

示例：

- 开关灯（可以切换回去）
- 调整音量或亮度（可以恢复先前的值）
- 打开/关闭窗口或应用程序（可以反转）
- 更改恒温器温度（可以恢复到先前的值）
- 锁定/解锁门（可以切换）

对于完全可逆的操作，每个操作 a_i 都有一个逆操作 a_i^{-1} 来撤销其效果：

$$a_i(C) \rightarrow C' \implies a_i^{-1}(C') \rightarrow C$$

要回滚一系列完全可逆的操作，请按相反顺序执行它们的逆操作：

$$\text{rollback}(a_1, a_2, \dots, a_k) = a_k^{-1}, a_{k-1}^{-1}, \dots, a_2^{-1}, a_1^{-1}$$

部分可逆操作：这些可以被撤销，但不能完全撤销。一些副作用会持续存在，或者一些信息会丢失。

示例：

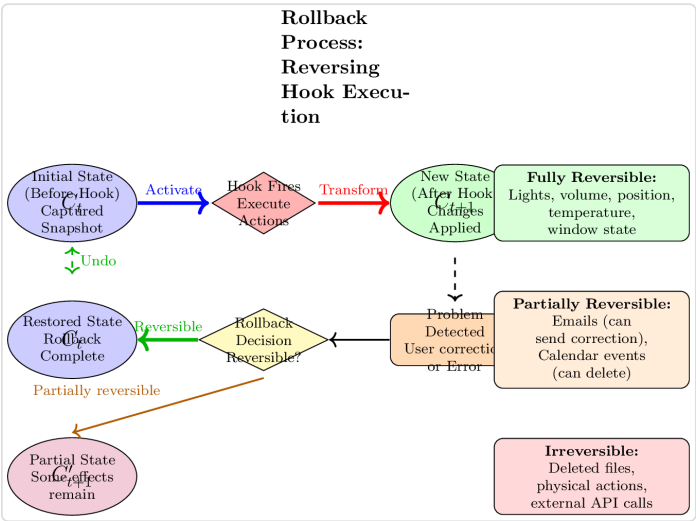
- 发送电子邮件（可以发送后续更正，但原始邮件无法撤回）
- 创建日历事件（可以删除，但收件人已被通知）
- 写入文件（如果已保存，可以恢复先前的内容，但修改时间戳已更改）
- 进行购买（可以退款，但交易已发生且时间已过去）

不可逆的操作：这些操作无法有意义地撤销。效果是永久的。

示例：

- 删除没有备份的文件（数据丢失）
- 粉碎实体文件
- 触发物理机制（打开阀门，激活电机）
- 进行有副作用的外部API调用（支付处理，向第三方发送消息）

这种区分很重要，因为系统必须根据操作的可逆性以不同方式处理回滚。



事务回滚：对于执行多个相关操作的钩子，系统可以使用事务语义——要么所有操作成功，要么全部回滚。这对于关键操作尤其重要，因为部分完成比不采取任何行动更糟。

考虑一个执行三项操作的旅行预订钩子：

1. 预订航班 → 创建航班预订
2. 预订酒店 → 创建酒店预订
3. 租车 → 创建租车预订

如果租车失败（没有可用车辆），仅预订航班和酒店会导致不一致状态。用户在目的地需要交通工具。在这种情况下，事务回滚会取消航班和酒店预订，将系统恢复到没有预订的 C_t 状态。

if a_i fails, then rollback(a_1, a_2, \dots, a_{i-1})

在钩子创建时指定自动回滚的事务执行。标记为事务的钩子保持其所有操作的回滚能力。

实现回滚：实用机制。不同的操作类型需要不同的回滚机制：

基于状态的回滚：对于修改系统状态的操作（灯光、音量、温度），存储先前的状态值并显式恢复：

```
rollback_action( $a_i$ ) = restore(property, value_from( $C_t$ ))
```

示例：如果钩子将亮度设置为 80%，而 C_t 显示为 40%，则回滚将亮度设置回 40%。

基于命令的回滚：对于具有明确逆操作的操作（打开/关闭，锁定/解锁），执行逆命令：

```
rollback("open door") = execute("close door")
```

补偿交易：对于无法直接撤销的操作（发送电子邮件，创建日历事件），执行补偿操作以减轻影响：

- 发送电子邮件 → 发送后续电子邮件解释错误
- 创建日历事件 → 删除事件并通知与会者
- 文件已删除 → 从备份恢复（如果可用）

快照恢复：对于涉及多个属性的复杂状态变化，恢复 C_t 的整个相关部分，而不是逐个逆转操作。这对于学习的钩子特别有用，因为确切的操作顺序可能很复杂。

触发回滚时：回滚可以通过几种机制启动：

1. 用户纠正：当系统检测到用户在纠正窗口内（通常为 5 分钟）逆转或修改了钩子的操作时，可以提供完全回滚剩余效果的选项。

2. 明确的撤销命令：用户可以通过界面明确请求 "撤销最后一个钩子"，触发立即回滚。

3. 钩子失败：如果事务钩子中的某个操作未能执行，则会自动回滚之前的操作。

4. 错误检测：如果系统检测到钩子的执行违反了安全约束或产生了异常结果，则可以自动触发回滚。

5. 级联终止：如果级联链达到深度限制或检测到无限循环，系统可能会回滚链中最近的钩子。

回滚的局限性和权衡：并非所有回滚都是可行或可取的：

时间不可逆转：时间无法被逆转。即使所有状态变化都被撤销，所花费的时间和消耗的能量也无法恢复。这对于时间敏感的操作尤其相关（因为提醒被抑制而错过会议无法通过回滚修复）。

外部副作用：涉及外部系统或其他用户的操作并不总是可以干净地撤销。发送给同事的电子邮件已被阅读；通过在本地图删除它来回滚并不会抹去他们的记忆。

级联依赖：如果其他钩子已经从被回滚钩子的操作中级联，回滚可能需要撤销整个级联链。这可能很复杂，并可能影响用户不打算撤销的钩子。

性能成本：维护回滚能力需要存储全面的快照，这会消耗内存。对于资源有限的系统，这可能并不适用于所有钩子。

用户困惑：激进的自动回滚可能会让用户感到困惑，如果他们不理解系统为什么要撤销操作。回滚应与清晰的沟通相结合，说明正在撤销的内容及其原因。

设计原则：优先预防而非回滚。虽然回滚是一个有价值的安全机制，但系统应优先防止错误执行，而不是依赖回滚来修复它们。这意味着：

- 使用精确的条件确保钩子仅在真正合适时触发
- 保持高成功率，以便仅激活可靠的钩子
- 采用优先级选择最佳钩子以应对竞争选项
- 在执行之前为高风险操作提供用户确认

回滚应被视为错误恢复，而非常规操作。一个运作良好的知识钩子系统很少需要回滚，因为钩子准确触发并产生预期结果。

回滚与学习：当发生回滚时（尤其是用户发起的），它会给学习系统提供强烈的负面反馈。回滚钩子的成功评分应该显著下降：

$$S_{t+1} = S_t \cdot (1 - \alpha_{\text{rollback}}), \quad \alpha_{\text{rollback}} > \alpha_{\text{correction}}$$

其中 α_{rollback} 是回滚惩罚，超过标准修正惩罚。这种更强的负面反馈表明该钩子不仅需要小幅调整——它在该上下文中根本是错误的。

同一钩子的重复回滚可能会触发自动禁用或删除，防止其被审查或改进之前再次触发。这可以防止问题钩子反复造成需要回滚的问题。

结论：回滚作为安全网。回滚操作作为知识钩子系统中的关键安全机制，提供了一条从错误、不正确激活和意外后果中恢复的路径。虽然并非所有操作都能完美逆转，但状态捕获、逆操作和补偿交易的结合使系统在必要时能够撤销大多数效果。

然而，回滚应该被视为一种后盾，而不是主要功能。主观技术的目标是学习模式，以至于钩子很少需要修正，几乎从不需要回滚。随着系统的成熟和钩子通过强化学习的改进，回滚变得越来越少——这表明技术真正学会了成为你。

3.3 组合与精炼

3.3.1 嵌套组合：钩子中的钩子

3.3.2 扁平组合：连接钩子

3.3.3 精炼与概括：按特异性排序钩子

精炼与概括是互补的操作，管理知识钩子如何根据其表现演变。这些操作解决了机器学习中的一个基本挑战：找到条件的最佳特异性水平。过于具体（过拟合），钩子很少激活。过于一般（欠拟合），钩子在不适当的上下文中触发。精炼与概括提供了根据经验结果动态调整钩子特异性的机制。

特定性排序：钩子可以根据其特定性进行排序——它们的条件有多严格。这个排序创建了一个自然的层次结构，帮助系统理解钩子之间的关系，并在何时使钩子更具体或更不具体做出智能决策。

正式地，我们定义一个特定性排序关系 \prec （发音为“细化”或“比...更具体”）：

$$KH_1 \prec KH_2 \iff R_1 \subset R_2$$

其中：

- $KH_1 \prec KH_2$ 意味着 " KH_1 是 KH_2 的细化" 或 " KH_1 比 KH_2 更具体"
- $R_1 \subset R_2$ 意味着 KH_1 的条件集是 KH_2 条件的一个真子集
- 这意味着 KH_1 拥有 KH_2 的所有条件加上额外的约束

换句话说，如果钩子 KH_1 包含钩子 KH_2 的所有条件并添加更多，那么 KH_1 就更具体。它将在更少的上下文中激活——仅在满足额外约束的情况下。

示例层次结构：考虑三个用于早晨咖啡准备的钩子：

KH_general: R = {time: 7:00-8:00} → A = {start coffee maker}

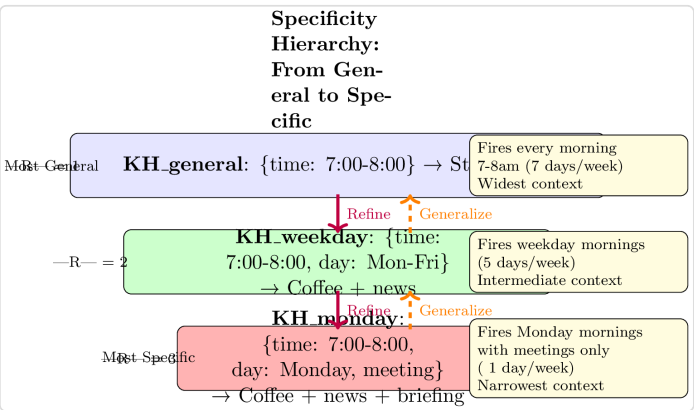
KH_weekday: R = {time: 7:00-8:00, day: Monday-Friday} → A = {start coffee maker, turn on news}

KH_monday: R = {time: 7:00-8:00, day: Monday, calendar: contains 'meeting'} → A = {start coffee maker, turn on news, prepare briefing}

这些钩子形成了一个特异性链：

$$KH_{\text{monday}} \prec KH_{\text{weekday}} \prec KH_{\text{general}}$$

KH_monday 是最具体的（最狭窄的激活上下文），KH_general 是最一般的（最广泛的激活上下文）。每个级别增加了缩小钩子触发条件的条件。



过拟合和欠拟合的问题：找到正确的特异性级别对钩子的有效性至关重要。这类似于经典的机器学习挑战：过拟合与欠拟合：

过拟合（过于具体）：具有过于严格条件的钩子很少会激活，即使在应该激活的时候。钩子学习到了虚假的模式——在学习过程中恰好存在的无关细节，但实际上并不是因果关系。

示例：你在星期五下午穿着红色衬衫时创建了一个会议，你的笔记本电脑电池为 73%。一个学习到的钩子可能最初提取的条件是：

```
R = {day: Friday, time: 14:00-17:00,
      clothing_color: red, battery: 70-75%}
```

这个钩子几乎不会再次触发，因为这些确切的条件很少对齐。相关条件是“星期五下午”，但钩子错误地学习到红色衣物和 73% 的电池也是必要的。这就是过拟合。

欠拟合（过于一般）：具有不足条件的钩子会在不适当的上下文中触发，产生不必要的动作并需要频繁的修正。

示例：经过几次修正，系统可能会过于激进地进行概括：

```
R = {time: 14:00-17:00} → A = {create
meeting}
```

现在这个钩子每天下午都会触发，无论是哪个日子或您是否真的需要会议。这是欠拟合——条件太宽泛。

最佳钩子平衡这些极端：

```
R = {day: Friday, time: 14:00-17:00}
```

这捕捉了实际模式，而没有多余的细节。

细化：使钩子更具体。细化是向钩子添加条件的操作，使其在触发时更加严格。这是在钩子产生过多误报时进行的——在不应触发的上下文中触发。

$$\text{refine}(KH) = (R \cup \{r_{\text{new}}\}, A, T, S)$$

其中 r_{new} 是从钩子错误触发的上下文中提取的额外条件。细化后的钩子 KH' 具有条件 $R' = R \cup \{r_{new}\}$ ，这意味着它继承了所有原始条件以及新的约束。

细化过程：

1. 检测误报：钩子触发，用户立即纠正它 ($Corr_t = 1$)

2. 分析上下文：将钩子错误触发的上下文 ($C_{incorrect}$) 与钩子正确触发的上下文 ($C_{correct}$, 1-n) 进行比较

3. 确定区分特征：找到在 $C_{correct}$ 案例中存在但在 $C_{incorrect}$ 中缺失的上下文特征

4. 添加条件：创建新的条件 r_{new} ，以捕捉这个区分特征

5. 更新钩子：用改进版本替换原始钩子

示例：您的“晚上9点调暗灯光”钩子在您观看电影时仍然触发，这并不是您想要的。系统分析成功激活与错误激活的情况：

- $C_{correct}$: {time: 21:00, location: home, TV: off, lights: bright}

- $C_{incorrect}$: {time: 21:00, location: home, TV: on, lights: dim}

区分特征是电视状态。改进添加了这个条件：

原始： $R = \{\text{time: 21:00, location: home}\}$

改进： $R = \{\text{time: 21:00, location: home, TV: off}\}$

现在，钩子仅在您不看电视时调暗灯光。

概括：使钩子不那么具体。概括是相反的操作——去除条件以使钩子更广泛地触发。这是在钩子过于限制时进行的，错过了应该触发的机会。

$$\text{generalize}(KH) = (R \setminus \{r_{\text{spurious}}\}, A, T, S)$$

其中 `r_spurious` 是一种不必要限制激活的条件。广义钩 `KH'` 的条件为 $R' = R \setminus \{r_{\text{spurious}}\}$ ，这意味着它具有所有原始条件，除了被移除的那个。

概括过程：

1. 检测低激活：钩子在触发时尽管成功率高，但触发频率低。
2. 用户手动重复操作：用户在钩子未触发的情况下执行该操作。
3. 分析模式：识别在这些新上下文中 `R` 中哪些条件持续被违反。
4. 移除条件：从 `R` 中删除过于限制的条件。
5. 监控：观察激活增加而纠正没有增加的情况。

示例：您用于启动早晨播放列表的钩子仅在早上 7:05 精确触发：

```
R = {time: 07:05, location: home, weekday: true}
```

但您在不同的日子手动在 7:03、7:08 和 7:12 启动了播放列表。系统识别到确切的时间条件过于限制。它进行了概括：

```
R = {time: 07:00-07:15, location: home, weekday: true}
```

现在钩子在您的早晨例行程序窗口中触发，而不仅仅是在一个精确的分钟。

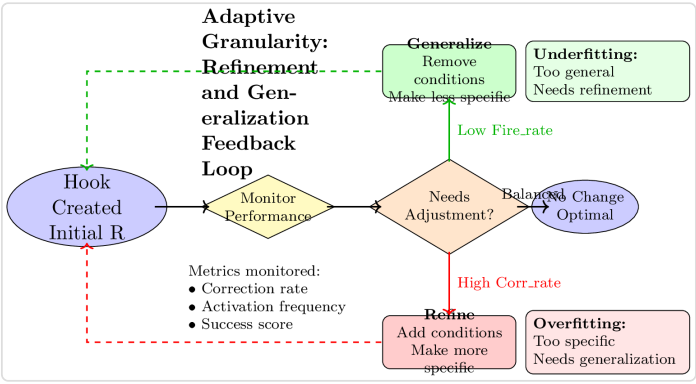
自适应粒度：关键的见解是最佳的特异性水平不是静态的——它根据性能进行调整。系统持续监控成功分数和激活模式，以确定是否需要细化或概括：

```
if (Corr_rate >  $\theta_{\text{refine}}$ ) then refine( $KH$ )
if (Fire_rate <  $\theta_{\text{generalize}}$   $\wedge$   $S > 0.8$ ) then generalize( $KH$ )
```

其中：

- Corr_rate 是纠正频率（纠正/激活）
- θ_{refine} 是触发细化的阈值（例如，0.3 = 30% 纠正）
- Fire_rate 是激活频率（激活/机会）
- $\theta_{\text{generalize}}$ 是触发概括的阈值（例如，0.1 = 仅在 10% 的潜在上下文中触发）
- S 是成功分数

这创建了一个反馈循环：过于频繁触发的钩子变得更加具体；而很少触发但在触发时效果良好的钩子变得更加一般。



部分排序和格结构：特异性关系 < 在所有钩子的集合上创建了部分排序。多个钩子可以存在于相同的特异性水平（彼此之间没有更具体的），钩子可以有多个细化或概括。

这个结构形成了一个格——一种数学结构，其中：

- 每对钩子都有一个最小上界（最具体的钩子，可以概括两者）
- 每对钩子都有一个最大下界（最一般的钩子，可以细化两者）

这个格结构使得对钩子关系进行复杂推理成为可能。例如，如果两个具有不同条件的钩子产生相同的动作并且都运行良好，它们可能是合并为一个更一般钩子的候选者。

与优先级的关系：细化和概括直接影响优先级。当多个钩子可以触发时，更具体的钩子优先于更一般的钩子（如优先级部分所讨论的）。这确保了：

- 专门化行为在两者都适用时覆盖通用行为
- 用户可以创建越来越具体的钩子来处理边缘情况
- 系统自然地一般模式演变为细致的、上下文感知的响应

示例：你可能有一个在晚上9点调暗灯光的通用钩子。随着时间的推移，你用一个更具体的钩子进行了细化，该钩子在周五/周六晚上保持灯光明亮。具体钩子（带有额外的“日期”条件）在周末优先，而通用钩子处理工作日的晚上。

自动与手动细化：系统可以执行自动和用户发起的细化/概括：

自动细化：由高修正率触发。系统分析修正中的模式，以识别缺失的条件。

自动概括：由低激活率和高成功率触发。系统通过比较应该触发但未触发的上下文来识别过于严格的条件。

手动细化：用户可以通过界面显式添加条件："仅在[条件]时执行此操作"

手动概括：用户可以移除条件："无论[条件]如何都执行此操作"

自动适应与手动控制的结合为用户提供了便利和必要时的细致控制。

为什么细化和概括很重要：这些操作对于知识钩子在精确度和覆盖率之间实现正确的平衡至关重要。没有它们：

- 学习到的钩子将停留在其初始特异性水平
- 过拟合的钩子将浪费系统资源，永远不会激活
- 欠拟合的钩子将不断需要修正，使用户感到沮丧
- 系统无法适应不断变化的用户模式和上下文

通过不断的精炼和经验性能指导的概括，知识钩子找到了最佳平衡点——足够具体以避免误报，足够通用以捕捉真实模式。这种自适应粒度使得主观技术能够学习个体用户的细微差别，而无需明确编程，从而创造出真正个性化的自动化，随着时间的推移不断改善。

3.3.4 等价性：当不同的钩子产生相同的结果时

等价性是确定两个或多个知识钩子在不同条件、不同动作序列或不同内部结构的情况下是否产生相同结果的操作。理解等价性对于优化、去重和在竞争钩子之间进行智能选择至关重要。

等价关系：如果两个钩子在相同的上下文中执行时产生不可区分的结果，则这两个钩子是等价的。我们正式定义等价关系 \equiv ：

$$KH_1 \equiv KH_2 \iff \forall C : \text{outcome}(KH_1, C) = \text{outcome}(KH_2, C)$$

其中：

- $KH_1 \equiv KH_2$ 意味着 " KH_1 等价于 KH_2 "
- $\forall C$ 意味着 "对于所有上下文"，在这些上下文中两个钩子可能会触发
- $\text{outcome}(KH, C)$ 表示在上下文 C 中执行钩子 KH 后的最终系统状态

如果两个钩子将系统从任何给定状态转换为相同的结果状态，则这两个钩子是等价的。所采取的具体路径——中间步骤、操作顺序、内部表示——并不重要。只有最终结果才重要。

重要提示：等价性是由可观察的结果定义的，而不是由钩子的结构定义的。如果钩子产生相同的结果，即使条件和动作完全不同，它们仍然可以是等价的。

简单示例：考虑三个用于打开客厅灯的钩子：

$KH_1: R = \{\text{time: 18:00, location: home}\} \rightarrow A = \{\text{lights.living_room} = \text{ON}\}$

$KH_2: R = \{\text{sunset: true, location: home}\} \rightarrow A = \{\text{lights.living_room.brightness} = 100\%\}$

$KH_3: R = \{\text{darkness_level: } >80\%, \text{ location: living_room}\} \rightarrow A = \{\text{turn_on(lights.living_room)}\}$

这三个钩子有不同的条件（基于时间、基于日落、基于黑暗）和略微不同的动作规范（开启与100%亮度与turn_on命令）。然而，它们在功能上是等价的：它们都导致客厅的灯全亮。

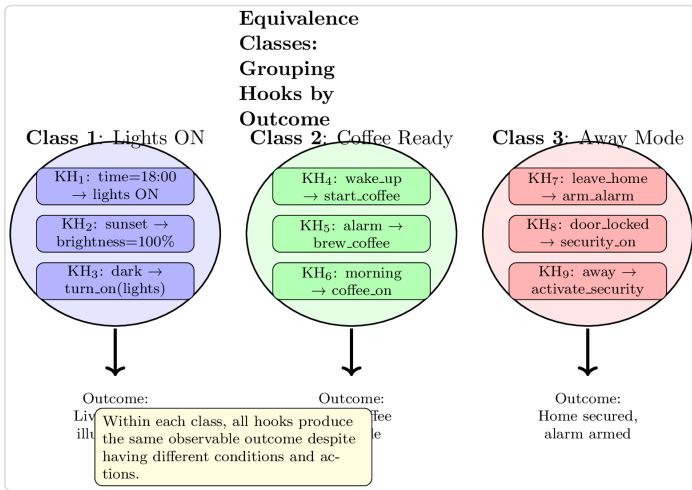
$$KH_1 \equiv KH_2 \equiv KH_3$$

等价类：等价关系将所有钩子的集合划分为等价类——产生相同结果的钩子组：

$$[KH] = \{KH' : KH' \equiv KH\}$$

其中 $[KH]$ 表示包含钩子 KH 的等价类。所有在同一等价类中的钩子从结果的角度来看是可以互换的，尽管它们在效率、可靠性或激活时机上可能有所不同。

等价法则指出，具有相同结果的钩子属于同一等价类，系统应该识别并利用这种关系进行优化。



为什么等价重要：等价关系使得几项关键的优化和智能行为成为可能：

1. 通过选择最小化：当多个等价钩子可以触发时，系统可以智能地选择最有效的一个。根据最小化法则和优先级规则，在等价钩子中选择：

$$KH^* = \arg \min_{KH \in [KH_{equiv}]} |A|$$

选择动作最少的钩子。如果动作数量相同，选择成功率最高的钩子。这确保系统始终以最有效的路径达到预期结果。

2. 去重：如果系统检测到在相同等价类中具有相似条件的多个钩子，它们可能会被合并，或者可以禁用一个以减少冗余：

$$\text{if } KH_1 \equiv KH_2 \wedge R_1 \approx R_2 \implies \text{merge}(KH_1, KH_2)$$

这防止系统维护多个几乎相同的钩子，它们服务于相同的目的。

3. 鲁棒性：拥有多个具有不同条件的等效钩子提供了冗余。如果一个钩子的条件变得不可满足（传感器故障，服务离线），其他等效钩子仍然可以通过替代方式实现相同的结果。

4. 学习替代路径：当系统观察到用户在不同上下文中通过不同方法实现相同结果时，它会将这些视为等效方法，并可以选择上下文适当的变体。

检测等价性：确定两个钩子是否等价并不总是简单。系统使用几种方法：

语法分析：比较动作中指定的最终状态。如果 KH_1 的动作设置 `lights.living_room.state = ON`，而 KH_2 的动作设置 `lights.living_room.brightness = 100%`，如果系统理解ON意味着100%亮度，则可以将其识别为等效。

语义分析：利用关于设备及其状态的领域知识。系统知道 `turn_on(lights)`，`set_brightness(lights, 100)` 和 `set_state(lights, ON)` 都产生相同的结果。

经验观察：在受控上下文中执行两个钩子，并观察结果状态是否相同。如果钩子始终从相同的 C_t 产生相同的 C_{t+1} ，则它们在经验上是等效的。

用户反馈：用户可以明确标记钩子为等效：“这两个钩子做的是同样的事情。”这为等价关系提供了真实依据。

复杂示例：会议准备。考虑为9点会议准备的钩子：

```
KH_alarm:  R  =  {time:  08:45,  calendar:
meeting_at_9am}      →      A      =
{show_notification('Meeting  in  15  min'),
open_calendar, open_meeting_notes}
```

```
KH_context:  R  =  {time:  08:30-08:59,
next_event: meeting, location: home} → A =
{display_reminder('Upcoming      meeting'),
launch_calendar_app,
open_document(meeting_notes)}
```

```
KH_smart:  R  =  {upcoming_meeting:  <15min,
user_active:  true} → A = {notify('Meeting
soon'), switch_to_calendar_view, show_notes}
```

这三个钩子有不同的触发条件和略微不同的动作实现，但它们在功能上是等效的：它们都使用户收到会议通知，并显示日历和笔记。系统将它们视为等效，并根据动作较少和成功评分较高的优先级进行排序。

部分等效性：有时钩子在某些上下文中是等效的，但在其他上下文中不是。我们称之为部分或条件等效性：

$$KH_1 \equiv_C KH_2 \iff \text{outcome}(KH_1, C) = \text{outcome}(KH_2, C)$$

当等效性仅适用于特定上下文 C ，而不是普遍适用时。

示例：两个温度控制的钩子：

$KH_1: R = \{temp < 18^{\circ}C\} \rightarrow A = \{heater = ON\}$

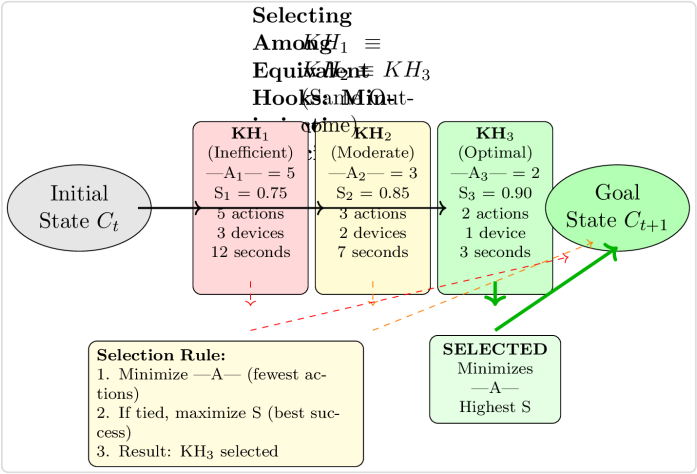
$KH_2: R = \{temp < 18^{\circ}C\} \rightarrow A = \{heater = ON, close_windows\}$

当窗户已经关闭时，这些钩子是等效的（都只是打开加热器），但当窗户打开时则不等效（ KH_2 还关闭窗口）。系统跟踪这些条件等效性，以便做出适合上下文的选择。

等价性与最小化法则：等价法则与最小化法则密切相关。最小化法则指出：“始终优先选择用户输入最少的路径。”当多个钩子是等价的（结果相同）时，此法则要求选择最有效的一个：

if $KH_1 \equiv KH_2$ and $|A_1| < |A_2| \implies prefer KH_1$

这产生了一种自然的优化压力：系统向实现每个结果的最有效方式演变，自动修剪或降级效率低下的等价钩子。



等价性与成功评分：有趣的是，等价钩子可能具有不同的成功评分，即使它们产生相同的结果。为什么？因为成功评分衡量的是可靠性和修正频率，而不仅仅是结果：

- 钩子 A 可能实现结果，但偶尔需要修正

- 钩子 B 以更少的修正实现相同的结果

尽管它们是等价的（最终状态相同），钩子 B 更可靠，成功评分更高。当两者都可以触发时，系统将优先选择钩子 B。

实际影响：理解等价性对用户和系统设计者有几个实际好处：

1. 自然冗余：用户不需要担心创建重复的钩子。如果他们手动创建以不同方式实现相同目标的钩子，系统会将它们识别为等价，并智能地在其中选择。

2. 自动优化：系统可以自动识别并优先选择更高效的等价钩子，而无需用户干预。一个打开灯光、播放音乐和调节温度的钩子（3个动作）将优于通过6个单独设备命令实现相同功能的钩子。

3. 钩子迁移：在升级到新设备或服务时，等价钩子可以自动迁移。如果您用新智能灯替换旧灯，控制旧灯的钩子可以更新为新灯的等价钩子。

4. A/B 测试：系统可以通过等效的钩子进行实验，以确定在实践中哪个最可靠，逐渐将使用转向表现最佳的选项。

等效类和系统演变：随着时间的推移，系统的钩子集合自然组织成等效类。在每个类中：

- 钩子根据效率和可靠性进行竞争
- 表现最佳的钩子积累使用
- 表现不佳的钩子被降级或最终禁用
- 系统学习哪些条件最可靠地预测每个结果何时是所需的

这产生了朝着最佳钩子集的进化压力——一个通过最有效、可靠的方式实现所有期望结果的钩子集，并在条件变化时具有适当的冗余。

数学属性：等价关系 \equiv 满足任何等价关系所需的三个属性：

反身性：每个钩子都等同于它自己：

$$\forall KH : KH \equiv KH$$

对称性：如果 KH_1 等同于 KH_2 ，则 KH_2 等同于 KH_1 ：

$$KH_1 \equiv KH_2 \implies KH_2 \equiv KH_1$$

传递性：如果 KH_1 等价于 KH_2 ，且 KH_2 等价于 KH_3 ，则 KH_1 等价于 KH_3 ：

$$KH_1 \equiv KH_2 \wedge KH_2 \equiv KH_3 \implies KH_1 \equiv KH_3$$

这些属性确保了等价类的明确定义，以及将钩子划分为类在数学上是合理的。

结论：等价作为优化的推动者。等价操作使知识钩子系统能够智能地优化和适应。通过识别不同的方法实现相同结果的情况，系统可以：

- 自动选择最有效的路径
- 消除冗余，同时保持稳健性
- 学习替代策略并选择适合上下文的变体
- 随着时间的推移，朝着最佳钩子配置演变

结合最小化法则，等价确保主观技术不仅实现用户目标——而且以最有效的方式实现这些目标，通过选择压力不断改进，偏向于更短、更可靠的路径以达到期望的结果。这就是系统学习的不仅是做什么，还有最佳的做法。

3.3.5 权重归一化：合并和整合相似的钩子

权重归一化是将相似或冗余的钩子合并为更强大的版本的操作。随着用户与主观技术的互动，系统自然学习到多个服务于相似或重叠目的的钩子。如果没有整合，钩子数据库将无限增长，造成低效和潜在冲突。权重归一化通过智能地合并相似钩子，同时保留它们的学习知识和成功指标来解决这个问题。

钩子繁殖的问题：在学习系统中，钩子的创建是持续的。每当用户在新的上下文中执行操作时，系统可能会创建一个新的钩子来捕捉该模式。虽然这使得全面学习成为可能，但也导致了几个问题：

冗余：多个钩子执行基本相同的操作，但在略微不同的上下文学习，导致系统混乱。

低效：在激活过程中评估数千个相似的钩子浪费了计算资源。

竞争：相似的钩子争夺激活，如果它们具有相似的条件但略有不同的操作或成功分数，可能会导致不可预测的行为。

碎片化：关于某个模式的知识分散在多个钩子中，而不是集中在一个经过良好训练的钩子中。

权重归一化通过识别相似的钩子并将它们合并为保留每个钩子最佳特性的统一版本来解决这些问题。

检测相似钩子：第一个挑战是确定哪些钩子足够相似以值得合并。系统使用几种相似性度量：

动作相似性：这些钩子是否执行相同或几乎相同的操作？这是主要标准：

$$\text{similar}(KH_1, KH_2) \iff A_1 \approx A_2$$

其中 \approx 表示近似相等——即使具体命令略有不同，产生相同结果的操作。

条件重叠：这些钩子是否具有显著重叠的条件？

$$\text{overlap}(R_1, R_2) = \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|} > \theta_{\text{overlap}}$$

如果条件共享超过阈值百分比（例如，60%），则这些钩子被视为相似。

上下文相似性：钩子是否在相似的上下文中激活？这可以通过比较每个钩子历史上触发的上下文来衡量。

结果等价性：如等价性部分所讨论，产生相同可观察结果的钩子是合并的候选者。

合并操作：当识别出相似的钩子时，可以将它们合并为一个统一的钩子：

$$KH_{\text{merged}} = \text{merge}(KH_1, KH_2, \dots, KH_n)$$

合并后的钩子必须保留所有源钩子的知识，同时创建一个更强大、更通用的版本。合并过程在每个组件上进行：

1. 条件（ R_{merged} ）：系统必须决定使用条件的并集还是交集：

并集方法（更一般）： $R_{\text{merged}} = R_1 \cup R_2 \cup \dots \cup R_n$

合并后的钩子在满足任何原始条件时触发。这创建了一个更通用的钩子，在任何源钩子触发的上下文中激活。

交集方法（更具体）： $R_{\text{merged}} = R_1 \cap R_2 \cap \dots \cap R_n$

合并后的钩子仅在满足所有原始条件时触发。这创建了一个更保守的钩子，仅在所有源钩子一致的上下文中激活。

选择取决于使用案例。对于安全关键操作，请使用交集（保守）。对于便利功能，请使用并集（宽松）。系统还可以使用

加权组合，根据源钩子触发的次数创建激活条件。

2. 操作 (A_{merged}) : 如果所有源钩子具有相同的操作，则直接使用这些操作。如果操作略有不同，系统必须进行调和：

$$A_{\text{merged}} = \text{reconcile}(A_1, A_2, \dots, A_n)$$

调和策略包括：

- 使用成功评分最高的钩子的操作序列
- 使用最短的操作序列（最小化法则）
- 如果操作是互补的，则按顺序组合操作
- 如果自动调和不确定，请要求用户解决冲突

3. 成功评分 (S_{merged}) : 这就是权重归一化得名的地方。合并钩子的成功评分是源钩子评分的加权平均值，权重由使用频率决定：

$$S_{\text{merged}} = \frac{\sum_{i=1}^n S_i \cdot n_i}{\sum_{i=1}^n n_i}$$

其中：

- S_i 是钩子 KH_i 的成功评分
- n_i 是 KH_i 成功触发的次数

这个加权平均确保了使用频率更高的钩子（因此有更多的实证证据）对合并得分有更大的影响。一个成功触发100次且得分为0.9的钩子比一个成功触发5次且得分为0.95的钩子贡献的权重更大。

4. 类型 (T_{merged}): 合并钩子的类型取决于其来源:

- 如果所有源钩子都是学习的, 则合并钩子是学习的
- 如果所有源钩子都是预定义的, 则合并钩子是预定义的
- 如果来源混合, 则合并钩子通常标记为学习的, 但可能保留对专家创作者的归属

扩展示例: 每周会议钩子。在几周内, 你每周五下午3点与 Alex 创建了会议。系统学习了单独的钩子:

KH_1 (第1周): $R = \{\text{day: Friday, time: 14:45, location: office, inbox: contains('Alex'), calendar: slot_free(15:00)}\} \rightarrow A = \{\text{create_meeting(Alex, Friday, 15:00)}\}$

成功: $n_1 = 3$ 次激活, $S_1 = 0.67$

KH_2 (第3周): $R = \{\text{day: Friday, time: 14:00-15:00, email: from('Alex'), subject: contains('sync')}\} \rightarrow A = \{\text{schedule_meeting(Alex, Friday, 15:00, title='Weekly Sync')}\}$

成功: $n_2 = 5$ 次激活, $S_2 = 0.84$

KH_3 (第5周): $R = \{\text{day: Friday, time: afternoon, context: work, participant: 'Alex'}\} \rightarrow A = \{\text{create_recurring_meeting(Alex, Friday, 15:00)}\}$

成功: $n_3 = 2$ 次激活, $S_3 = 1.00$

系统检测到相似性:

- 所有三个钩子在星期五下午 15:00 与 Alex 创建会议
- 条件显著重叠 (都参考星期五和 Alex)

- 行动产生等效结果

合并过程：

条件：取核心条件的交集（最具体的共同集合）：

```
R_merged = {day: Friday, time: 14:00-15:00,
participant: 'Alex'}
```

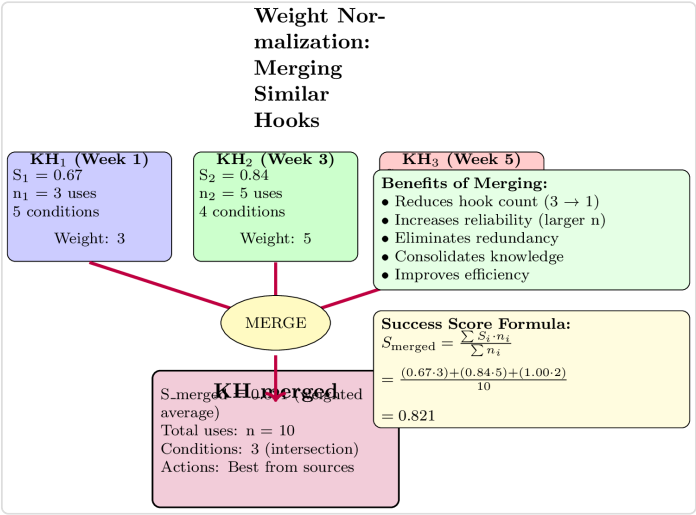
行动：使用最成功的行动模式（KH₂ 的方法与标题）：

```
A_merged = {schedule_meeting(Alex, Friday,
15:00, title='Weekly Sync')}
```

成功评分：加权平均：

$$S_{merged} = \frac{(0.67 \cdot 3) + (0.84 \cdot 5) + (1.00 \cdot 2)}{3 + 5 + 2} = \frac{2.01 + 4.20 + 2.00}{10} = \frac{8.21}{10} = 0.821$$

合并的钩子基于10次总激活，具有0.82的强大成功评分，比任何单独的源钩子更可靠，因为样本量更大。



正则表达式集归一化：书中提到归一化“正则表达式集”——这指的是使用正则表达式或模式匹配的条件模式。当合并具有相

似正则模式的钩子时，系统会创建归一化的通用模式：

示例：三个钩子在电子邮件主题上触发：

- KH_1 ：主题匹配“Meeting.*Friday”
- KH_2 ：主题匹配“Friday.*meeting”
- KH_3 ：主题匹配“.*Friday.*meeting.*”

这些可以归一化为一个更通用的单一模式：

- KH_{merged} ：主题匹配“.*(meeting|Meeting).*(Friday|friday).*”

这捕捉了所有变体，同时简化为一个正则表达式条件。

合并时机：系统应用多个标准来决定何时合并是合适的：

相似性阈值：仅合并高度相似的钩子（>80%条件重叠，动作相同或等效）

最小使用：在每个钩子至少使用几次之前不要合并（例如， $n_i \geq 3$ ）。这可以防止在模式尚未确立之前过早合并。

成功一致性：仅合并具有相似成功分数的钩子（彼此相差不超过0.2）。如果一个钩子的 $S=0.9$ 而另一个的 $S=0.3$ ，它们可能看起来相似但行为非常不同——不要合并。

用户确认：对于重要或经常使用的钩子，在合并之前询问用户：“这些钩子看起来相似——您想合并它们吗？”

时间间隔：优先合并时间在时间上接近创建的钩子。相隔几个月创建的钩子可能代表真正不同的模式，尽管表面上相似。

选择性合并：有时只有一部分相似的钩子应该被合并：

if $\{KH_1, KH_2, KH_3, KH_4, KH_5\}$ are similar, but KH_5 is outlier
then $KH_{merged} = \text{merge}(KH_1, KH_2, KH_3, KH_4)$, keep KH_5 separate

这保留了那些虽然相似但具有不同目的或显著不同特征的钩子。

周期性整合：权重归一化不是一次性操作。系统定期（例如，每周）扫描合并机会：

1. 识别集群：按相似性指标对钩子进行分组
2. 评估合并候选：检查每个集群的阈值
3. 执行合并：整合已批准的集群
4. 更新引用：任何引用源钩子的组合或级联现在引用合并的钩子
5. 存档源：原始钩子被存档（未删除），以便在需要时可以撤销合并

对系统性能的影响：权重归一化显著提高系统效率：

降低评估成本：激活期间需要评估的钩子更少，响应时间更快

提高准确性：合并的钩子拥有更多的训练数据（更高的 n ），从而导致更可靠的成功评分

更清晰的行为：消除竞争的相似钩子减少了不可预测的行为

更好的学习：整合的钩子更快地积累经验，提高适应率

归一化后的示例指标：

- 钩子数量：1,247 → 418（减少66%）
- 平均成功分数：0.72 → 0.84（通过整合提高可靠性）
- 激活评估时间：45毫秒 → 12毫秒（由于钩子减少而更快）

可逆性：如果合并出现问题，可以撤销合并。系统保持：

- 源钩子的完整记录
- 合并历史和理由
- 用户覆盖能力（“取消合并这些钩子”）

如果合并的钩子表现不佳（成功分数显著下降），系统可以自动将其拆分回源钩子，并尝试不同的合并参数。

结论：整合作为系统优化。权重归一化是保持知识钩子系统精简高效的日常维护操作。通过识别和合并相似的钩子，系统：

- 防止钩子数据库的无限增长
- 将学习到的知识整合为强大、训练有素的钩子
- 消除冗余和冲突行为
- 提高准确性和性能

成功评分的加权平均确保合并的钩子反映所有源钩子的完整实证证据，从而创建更可靠的自动化。结合代数中的其他操作——激活、执行、学习、组合、级联、优先级、回滚、精炼和等价——权重归一化完成了工具包，使知识钩子能够从简单的反应模式演变为用户的复杂、高效、主观扩展。

3.4 知识钩子的代数法则：治理原则

我们现在已经探索了知识钩子的完整操作机制：它们如何在条件匹配上下文时激活，如何执行一系列动作，如何通过强化学习从纠正中学习，如何组合成复杂的层次结构，如何通过触发链级联，如何通过优先级竞争，如何在发生错误时回滚，如何精炼和概括以找到最佳特异性，如何通过结果对钩子进行分组的等价类，以及如何通过权重归一化整合相似模式。这些操作构成了代数的词汇——描述钩子所做事物的动词。

但仅有操作并不构成代数。代数需要法则——基本原则，规范操作如何相互作用，限制它们的行为，并确保系统表现出可预测、理想的特性。正如算术法则（交换律、结合律、分配律）使我们能够自信地操作数字，知识钩子代数的法则使我们能够推理、预测和优化主观系统的行为。

这些法则不是外部强加的任意约束。它们自然地的主观技术的核心目标中产生：不断最小化用户输入。每一条法则都直接或间接地服务于这一最终目的，使系统能够学习、适应并优化为零输入交互。它们形式化了技术应逐渐减少努力、应从经验中学习、应选择高效路径以及应持续改进的直觉。

知识钩子的代数受五条基本法则的支配：

1. 最小化法则：始终优先选择用户输入最少的路径。当多个钩子可以实现相同的结果时，选择需要最少操作的那个（ $|A|$ 最小）。这是整个系统的主要驱动力——直接表达零输入技术目标的原则。

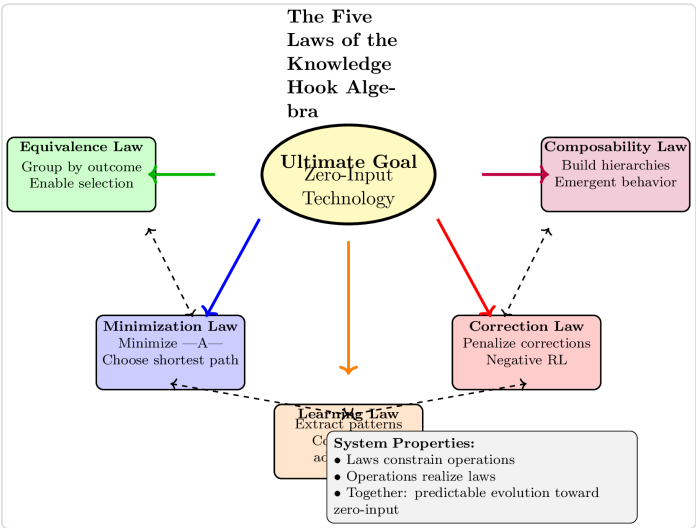
2. 纠正法则：需要用户纠正的钩子会因成功分数降低而受到惩罚。这实施了负强化学习——系统不是通过最大化奖励来学习，而是通过最小化纠正来学习。没有纠正就是成功；有纠正则表示失败。

3. 等价法则：具有相同可观察结果的钩子属于同一等价类。这使系统能够识别不同方法实现相同结果的情况，从而在替代方案之间进行智能选择并整合冗余模式。

4. 组合法则：钩子可以通过组合操作组合成更大、更复杂的结构。复合钩子根据其组件表现出可预测的行为，使得层次化组织和从简单原语中出现复杂行为成为可能。

5. 学习法则：上下文增量积累成加权条件，随着时间的推移精炼钩子。系统从每次交互中持续学习，从观察到的行为中提取模式，并调整条件以匹配成功的上下文。

这五条法则作为一个集成系统共同工作。最小化法则设定了优化目标。纠正法则提供反馈信号。等价法则使比较和整合成为可能。组合法则允许复杂行为从简单组件中出现。而学习法则确保持续适应以实现更好的性能。



法律的重要性：这些法律将知识钩框架从一系列机制转变为具有可证明属性的连贯数学系统。它们回答了关键问题：

- 可预测性：我们如何知道系统会一致地表现？这些法律保证钩子遵循可预测的规则，防止混乱或任意行为。
- 最优性：我们如何确保系统做出良好的选择？最小化法则确保在等效选项中，系统始终选择最有效的路径。
- 学习：系统如何随着时间的推移而改进？学习法则和纠正法则共同作用，实施一种负强化学习形式，将成功分数推向1.0。
- 可扩展性：简单的钩子如何组合成复杂的行为？组合法则保证组合保持期望的属性并实现层次化组织。

- 效率：我们如何防止冗余？等价法则使系统能够识别和合并重复的模式。

与操作的关系：每条法则管理我们已经探索过的一种或多种操作：

- 最小化法则管理优先级和等价选择，确保系统始终选择到达目标的最短路径。

- 修正法则管理学习更新，定义成功分数如何根据修正的发生而变化。

- 等价法则管理等价检测和权重归一化，允许系统对相似的钩子进行分组和合并。

- 组合性法则管理组合和级联，确保组合的钩子表现出可预测的行为。

- 学习法则管理学习步骤和细化/概括，定义条件如何从观察到的模式中演变。

理解这些法则对于理论和实践都是至关重要的。在理论上，它们为推理主观系统提供了正式基础——证明属性、分析收敛性和设计优化。在实践中，它们指导实施决策，帮助开发人员构建表现出正确行为并避免病态案例的系统。

通向零输入的路径：这些法则的最终目的是形式化并强制执行推动所有主观技术的原则：持续减少所需的用户输入。传统系统需要不断的明确命令。受这些法则支配的主观系统，逐渐学习预测需求并自主行动。

考虑单个钩子的轨迹随时间的变化：

第1周：用户手动执行一个动作。系统创建了一个包含许多条件的钩子（学习法则）。成功评分开始较低（约0.5）。

第2-3周：钩子偶尔触发，有时需要纠正。成功评分波动（纠正法则）。通过概括和细化操作（学习法则）来优化条

件。

第4-8周：钩子定期触发，纠正次数减少。成功评分上升至0.9。系统识别等效钩子并将其整合（等效法则）。在等效选项中，优先选择最有效的（最小化法则）。

第9周及以后：钩子可靠地触发，纠正次数很少。成功评分稳定在接近1.0。该动作在正确的上下文中自动发生，无需用户输入。行为已完全学习。

这种进展——从手动动作到自主预期——并非偶然。这是一个由奖励效率、惩罚错误、识别模式、实现组合并持续学习的法则所支配的系统的必然结果。这些法则确保系统朝着零输入操作演变。

经济影响：这些法则对主观热货币（STC）也有深远的经济影响，我们将在后面的章节中深入探讨。以下是联系：

- 最小化法则直接转化为能量最小化——更少的动作意味着消耗更少的物理能量。
- 纠正法则定义了钩子的质量指标——节省更多能量（需要更少纠正）的钩子更有价值。
- 学习法则使系统能够自动发现节能模式，而无需专家知识。
- 高质量的钩子（高成功率）代表了结晶效率——关于如何在特定环境中最小化能量消耗的知识。这些钩子具有经济价值，因为它们为用户节省了实际能量。

在STC框架中，创造或改进能够实现高成功率的钩子的专家会获得奖励。这些法律确保这一奖励系统与实际的能量节省相一致——真正最小化输入（因此能量）的钩子通过最小化和纠正法则的自然选择压力脱颖而出。

形式属性：这些法律赋予知识钩子代数几个重要的形式属性，使其区别于传统编程范式：

1. 收敛性：受这些法律支配的系统自然趋向于稳定、高效的配置。与保持静态的传统程序不同，知识钩子系统向最优性演变。

2. 鲁棒性：这些法律确保系统优雅地降级。糟糕的钩子（低成功率）通过纠正法则自动被抑制。冗余的钩子通过等价法则被整合。系统自我修复。

3. 可组合性而不爆炸：与神经网络中组合可能导致指数参数增长不同，知识钩子的组合保持可控。可组合性法则确保组合的钩子表现可预测。

4. 可解释性：由于钩子具有明确的条件和动作，系统的行为即使在复杂化时也保持可解释。这些法律支配可见、可理解的操作，而不是不透明的转换。

5. 增量学习：学习法则使得在不发生灾难性遗忘的情况下实现持续改进成为可能。新模式在保留成功的现有行为的同时被纳入。

本节结构：在以下小节中，我们将详细探讨每一条法律：

- 最小化法则：驱动所有决策的优化目标
- 纠正法则：负强化学习和成功评分动态
- 等价法则：基于结果的分类和智能选择
- 组合性法则：从简单组件构建复杂行为
- 学习法则：模式提取和持续适应

每个子章节将提供：

- 法则的正式陈述
- 适当的数学形式化
- 对法则含义的直观解释

- 法则如何支配特定操作
- 显示法则在实际应用中的示例
- 系统设计和实施的影响
- 与零输入技术和STC的更广泛目标的联系

这些法律共同完成了知识钩代数的正式规范。它们将钩子从一种巧妙的实现技术转变为一个严格的数学框架——不仅使零输入技术成为可能，还为基于测量和验证的能源节约的新经济系统提供了理论基础。

这些不仅仅是指导方针或最佳实践。它们是必须满足的基本原则，以使系统真正具备主观性——使技术真正学习成为你，最小化你的努力，将你的身体和思想扩展到数字和物理世界中。违反这些法律，你得到的就是其他东西：也许是一个有用的系统，甚至可能是一个智能系统，但不是我们所定义的精确意义上的主观系统。

现在让我们逐一审视每一条法律，理解它如何塑造知识钩的行为，以及这五个原则如何共同创造接近零输入技术理想的系统。

3.4.1 最小化法则：首要指令

最小化法则是整个知识钩代数的基础原则——支配所有其他操作并驱动系统不可避免地走向零输入技术的首要指令。它简单地陈述：始终优先选择用户输入最少的路径。虽然表述简单，但这一法则对系统行为、设计决策以及最终的主观热货币经济框架具有深远的影响。

正式声明：在所有能够实现期望结果的钩子中，系统必须选择需要最少操作的钩子：

$$KH^*(t) = \arg \min_{KH \in \mathcal{H}_{\text{valid}}} |A(KH)|$$

其中：

- $KH^*(t)$ 是在时间 t 选择的最佳钩子
- $\mathcal{H}_{\text{valid}}$ 是当前条件与上下文匹配的所有钩子的集合
- $|A(KH)|$ 是钩子 KH 的动作序列中的动作数量
- $\arg \min$ 选择动作数量最少的钩子

如果多个钩子的动作数量相同且最少，则通过成功分数来打破平局：

$$\text{if } |A(KH_1)| = |A(KH_2)| = \min_{KH} |A(KH)| \implies \text{select } KH^* = \arg \max_{KH} S(KH)$$

这创建了一个两级优化：首先最小化动作，然后在同样高效的选项中最大化可靠性。

为什么选择动作数量？选择最小化 $|A|$ 而不是其他指标（执行时间、计算成本等）是有意为之且根本的。动作代表了对世界的离散干预——必须执行的状态变化。每个动作：

- 代表可能发生失败的一个点
- 需要能量消耗（由用户或设备）
- 增加复杂性和潜在错误
- 需要关注和监控

通过最小化动作，我们同时最小化了能量消耗、错误概率和认知负担。动作数量作为所有这些维度的总系统成本的代理。

上下文输入反向关系：最小化法则存在于一个更广泛的框架中，该框架将上下文丰富性与所需输入相关联。随着系统积累

更多的上下文（ C ），实现结果所需的明确用户输入（ U ）减少：

$$U = f(C), \quad \text{where } \frac{\partial U}{\partial C} < 0$$

这种反向关系捕捉了主观技术的本质：系统对您和您的情况了解得越多，您需要明确告知它的内容就越少。在极限情况下，当上下文变得完美（ $C \rightarrow \infty$ ）时，所需输入接近于零：

$$\lim_{C \rightarrow \infty} U(C) = 0$$

这就是零输入理想——一种能够预见需求并在不需要任何明确命令的情况下行动的技术。最小化法则则是推动系统朝向这一极限的操作机制。

最小化法则如何支配操作：最小化法则直接控制我们探讨的几个关键操作：

1. 优先级：当多个钩子与当前上下文匹配时，最小化法则决定哪个被触发。这在优先级部分中已经讨论过，但现在我们将其视为基本法则的应用，而不是孤立的机制。

2. 等效选择：在等效钩子（那些产生相同结果的钩子）中，最小化法则要求选择最有效的。这确保系统始终以最短路径达到任何目标。

3. 组合选择：在组合钩子时，系统倾向于选择产生更少总动作的组合。如果平坦组合的动作序列比嵌套组合短，即使嵌套形式在概念上更优雅，系统也可能更倾向于选择平坦组合。

4. 学习偏见：当系统从用户行为中学习新钩子时，它应优先提取能够最小化未来动作的模式。如果用户通过低效的序列

完成任务，学习到的钩子应优化该序列。

5. 权重归一化决策：在合并相似钩子时，如果它们同样可靠，系统应倾向于保留动作较少的版本。

实际示例：航班预订。想象一下三个用于预订航班的钩子：

KH_1 （手动过程）：

```
A = [open_browser, navigate_to_airline,
search_flights, compare_prices, select_flight,
enter_passenger_info, enter_payment,
confirm_booking, save_confirmation]
```

$|A_1| = 9$ 个操作

KH_2 （半自动化）：

```
A = [voice_command('book flight to Tokyo'),
confirm_dates, approve_payment]
```

$|A_2| = 3$ 个操作

KH_3 （完全自动化）：

```
A =
[auto_book_based_on_calendar_and_preferences]
```

$|A_3| = 1$ 个操作

最小化法则要求选择 KH_3 （假设三者都能实现预期结果且 KH_3 具有足够的成功分数）。这不仅仅是方便——这是代数的基本要求。系统必须选择最短路径。

双层优化：成功分数作为平局打破者。当钩子的动作计数相等时，最小化法则依赖于成功分数：

$$KH^* = \begin{cases} \arg \min_{KH} |A(KH)| & \text{if unique minimum exists} \\ \arg \max_{KH \in \mathcal{H}_{\min}} S(KH) & \text{if multiple hooks at minimum} \end{cases}$$

其中 \mathcal{H}_{\min} 是动作计数最少的钩子集合。这种双层方法确保：

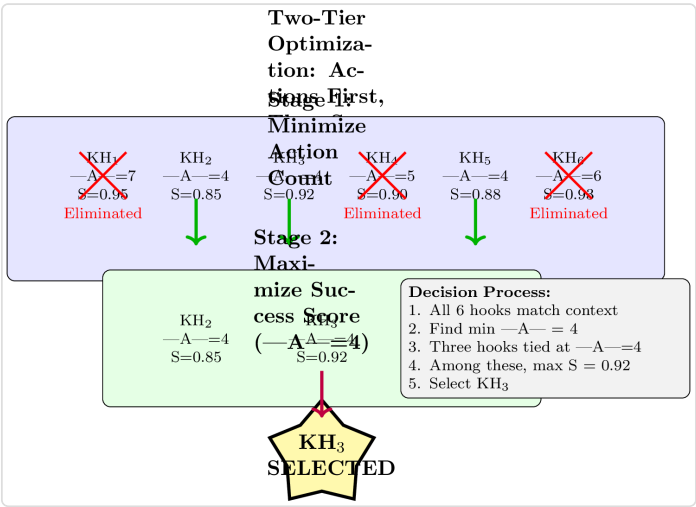
- 1. 效率从不以可靠性为代价
- 2. 在同样高效的选项中，我们选择最可靠的
- 3. 系统朝着效率和质量双重发展

示例：两个钩子都需要 4 次操作来安排会议：

KH_A: $|A| = 4, S = 0.85$ (有时需要参与者修正)

KH_B: $|A| = 4, S = 0.92$ (很少需要修正)

最小化法则无法根据动作计数区分它们，因此根据更高的成功分数选择 KH_B。这确保了在效率相等时的质量。



与能量最小化的联系：最小化法则专注于减少用户输入，具有直接的物理解释：它最小化能量消耗。每个动作都需要能量——无论是打字、点击或说话的物理能量，还是决定做什么的认知能量。

这创建了一个直接通往主观热货币（STC）的桥梁。在STC框架中：

$$E_{\text{total}} = E_{\text{user}} + E_{\text{device}}$$

其中 E_{user} 是用户消耗的能量（肌肉运动、认知努力）， E_{device} 是执行动作的设备消耗的能量。通过最小化 $|A|$ ，我们最小化这两个组件：

- 更少的用户操作 → 更少的 E_{user} （更少的点击、击键、命令）
- 更少的设备操作 → 更少的 E_{device} （更少的API调用、计算、传输）

因此，最小化法则成为了一种能量守恒法则——一个自然与物理热力学相一致的原则。这种一致性使得STC成为可能：我们可以用物理能量单位（焦耳）来衡量知识钩子的价值。

收敛到零输入：最小化法则创造了朝向零输入技术的进化压力。随着时间的推移：

第1周：存在多个钩子，具有不同的操作计数。系统优先选择和使用操作较少的钩子。

第4周：操作较多的钩子使用很少，它们的成功评分停滞或下降。高效的钩子积累使用和更高的评分。

第12周：通过精炼和学习，出现了新的钩子，可以用更少的操作实现目标。这些钩子由于最小化法则立即占据主导地位。

第26周：系统已收敛到每个目标的最小可能操作计数。进一步的改进只能通过根本新的方法或更好的上下文理解来实现。

最终状态是需要最小或零显式输入的技术——系统根据上下文预测需求并自主行动。最小化法则保证了这种收敛。

违规与病态：如果系统违反最小化法则会发生什么？后果是立竿见影且严重的：

用户挫败感：如果系统在可用的3个动作钩子时选择了7个动作钩子，用户会体验到不必要的摩擦。他们可能会手动覆盖或放弃系统。

能源浪费：选择低效路径会浪费用户和设备的能源，违背主观技术的核心目标。

经济失调：在STC系统中，违反最小化法则意味着奖励低效解决方案，产生扭曲的激励，从而破坏整个经济模型。

学习失败：没有最小化法则，系统没有明确的优化目标。学习变得无方向——系统可能学习复杂的低效行为，而不是简单的高效行为。

违反最小化法则的系统不仅仅是次优的——它根本无法成为主观技术。该法则不是可选的；它定义了最小化用户努力的意义。

实施考虑：实施最小化法则需要对几个细节给予仔细关注：

1. 动作粒度：什么算作“一个动作”？定义必须一致。打开一个文件可能算作一个动作，但“打开文件并滚动到第5页”算作一个还是两个？通常，无法进一步分解的原子操作被视为单个动作。

2. 隐藏动作：某些钩子可能有在A中不可见的隐藏设置或拆卸动作。这些必须包含在 $|A|$ 中以便公平比较。一个 $A = [\text{action1}]$ 但有大量隐藏操作的钩子实际上并不更高效。

3. 异步操作：异步执行的操作（例如，在后台发送电子邮件）仍然计入 $|A|$ 。最小化法则关注状态变化的数量，而不是它们的时机。

4. 组合钩子：当钩子组合时， $|A|$ 必须反映包括子钩子在内的总操作计数。一个具有2个操作的父钩子和3个操作的

子钩子的嵌套组合的 $|A| = 5$ ，而不是 2。

5. 用户确认：需要用户确认的操作（如批准付款）计入操作。一个完全自动化的过程 $|A| = 5$ 比一个半自动化的过程 $|A| = 3 +$ 仅在确认被计入时才算确认更可取。

与其他法则的关系：最小化法则与其他四个法则相互作用并加以约束：

修正法则：这些法则共同作用。最小化法则选择高效的钩子；修正法则确保它们的可靠性。一个具有最小 $|A|$ 但频繁修正的钩子最终会被一个稍长但更可靠的替代品所取代，因为成功评分不断演变。

等价法则：最小化法则决定使用哪个等价钩子——始终是最有效的。等价法则提供了识别钩子何时实现相同目标的框架；最小化法则决定优先选择哪个。

可组合性法则：组合必须尊重最小化。当从简单钩子构建复杂钩子时，目标是实现最小的总 $|A|$ 。可组合性法则使构建模块成为可能；最小化法则指导它们的组装方式。

学习法则：学习由最小化引导。当从用户行为中提取模式时，系统应学习实现用户明显目标的最有效版本，而不仅仅是复制他们的确切操作。

这五个法则共同创造了一个连贯的系统，其中效率、可靠性、组合和学习都朝着同一个最终目标努力：零输入技术。

哲学意义：最小化法则代表了对人机交互和技术设计思考的根本转变。传统系统优化完整性、功能或能力。最小化法则优化缺失——即用户不必做的事情。

这很深刻：技术的评判标准不是它能做什么，而是它让你避免做什么。最好的技术是无形的、具有前瞻性的和毫不费力的。最小化法则将这种直觉形式化为推动系统演变的精确数学要求。

在STC的背景下，这变得更加重要。通过最小化行为（从而节省能源），我们创造了可衡量的经济价值。每一个避免的行为都是节省的能源，而节省的能源可以被量化、记入和奖励。因此，最小化法则成为抽象效率与具体经济价值之间的桥梁。

结论：首要法则。最小化法则不仅仅是平等法则中的一条——它是许多代数行为的首要法则。它回答了根本性的问题：在多个选项中，我们应该选择哪个？答案是：始终选择最有效的。

这条法则将知识钩子从一种巧妙的自动化技术转变为一种严格的优化框架。它确保基于这种代数构建的系统不可避免地朝着零输入操作演变，创造出最愉悦的用户体验和最大的能源节省。

在接下来的章节中，我们将探讨其他四条法则如何补充和限制最小化法则，创造出一种完整的代数，使真正主观的技术成为可能——系统通过学习最小化你的努力，自动且持续地学习成为你，永远朝着零输入理想趋势发展，在你意识到想要之前，技术就已先行。

3.4.2 修正法则：负强化学习

修正法则实现了使知识钩子能够随着时间学习和改进的反馈机制。它指出：需要用户修正的钩子会因成功分数降低而受到惩罚。这创造了一种负强化学习的形式，其中缺乏修正被视为成功，而修正的存在则表明失败。随着时间的推移，这将修正的概率驱动向零，创造出很少需要用户干预的系统。

正式声明：在每次钩子激活后，成功分数会根据是否发生了修正而更新：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

其中：

- $S(t)$ 是当前的成功分数

- $S(t+1)$ 是更新后的成功分数
- $\alpha \in (0,1]$ 是控制适应速度的学习率
- $\mathbb{1}[\text{Corr}_t = 0]$ 是一个指示函数：如果没有发生修正则等于1，如果需要修正则等于0
- $\text{Corr}_t \in \{0,1\}$ 表示用户在时间 t 是否修正了钩子的动作

这个更新规则编码了一个强大的原则：沉默即是认可。如果用户没有修正钩子的动作，那么这种缺乏修正被解读为成功。如果用户确实进行了修正，成功分数就会下降。系统不是通过接收明确的积极反馈来学习，而是通过最小化对负反馈的需求来学习。

负强化与正强化：修正法则代表了一种与传统强化学习根本不同的学习范式：

传统强化学习（正强化）：

- 系统寻求最大化奖励： $\max \sum r_t$
- 需要来自环境的明确奖励信号
- 优化未来的累积价值
- 贝尔曼方程： $V(s) = \max_a [r(s,a) + \gamma \cdot V(s')]$

纠正法（负强化）：

- 系统寻求最小化纠正： $\min \sum \text{Corr}_t$
- 从缺乏纠正中学习（隐含认可）
- 优化即时准确性（没有未来折扣）
- 成功更新： $S(t+1) = (1-\alpha)S(t) + \alpha \cdot \mathbb{1}[\text{Corr}_t = 0]$

关键的见解是，贝尔曼方程优化外部奖励，而纠正法优化用户努力的最小化。未来状态并不重要——只关心当前交互是否需要纠正。这与零输入技术的目标完全一致：我们希望钩子现在就能正常工作，无需纠正，无论长期策略如何。

什么算作纠正？准确检测纠正对纠正法的正常运作至关重要。纠正是指在激活后一个时间窗口（通常为10-30秒）内，任何修改或撤销钩子动作的用户输入。示例包括：

直接纠正：

- 在钩子触发后立即按“撤销”
- 使用回滚功能来逆转钩子的动作
- 通过用户界面显式禁用钩子

隐式修正：

- 手动更改钩子刚刚修改的设置
- 删除或修改钩子创建的内容
- 重复钩子刚刚执行的操作（表示不满）
- 采取与钩子的意图相矛盾的行动

系统必须区分修正与正常用户活动。如果钩子在下午3点安排会议，而用户在三小时后将其更改为下午3:30，这很可能是一个合法的日程更改，而不是修正。时间接近性是关键：修正通常发生在钩子触发后立即。

$$\text{Corr}_t = 1 \iff \exists \text{ user_action} \in [t, t + \Delta t] \text{ that contradicts KH's action}$$

其中 Δt 是修正窗口（通常为 10-30 秒）。超出此窗口的操作不被视为修正。

学习率 α ：学习率控制成功分数适应新证据的速度。此参数平衡稳定性与响应性：

高 α (例如, 0.3-0.5) :

- 快速适应最近的表现
- 单次修正对分数影响很大
- 更加波动, 可能对异常反应过度
- 适合快速变化的模式或新钩子

低 α (例如, 0.05-0.15) :

- 缓慢、稳定的学习
- 单次修正影响较小
- 对偶尔的错误更具鲁棒性
- 适合具有稳定模式的已建立钩子

最佳的 α 可以是自适应的, 开始时对新钩子较高, 随着使用的积累而降低:

$$\alpha(n) = \alpha_{\max} \cdot e^{-\lambda n} + \alpha_{\min}$$

其中 n 是钩子触发的次数, λ 控制衰减速率。这使得新的钩子具有高塑性, 同时为已建立的钩子提供稳定性。

收敛分析: 在修正法则下, 成功评分收敛到反映真实钩子可靠性的值。让我们分析一下动态:

案例 1: 完美钩子 (从不需要修正)

如果 $\text{Corr}_t = 0$ 对于所有 t , 则 $\mathbb{1}[\text{Corr}_t = 0]$ 始终为 1:

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot 1 = (1 - \alpha)S(t) + \alpha$$

这是一个在 $S^* = 1$ 处的线性递归。随着 $t \rightarrow \infty$, $S(t) \rightarrow 1.0$ 。完美钩子获得完美评分。

案例 2：失败钩子（总是需要修正）

如果 $\text{Corr}_t = 1$ 对于所有 t ，则 $\mathbb{1}[\text{Corr}_t = 0]$ 始终为 0：

$$S(t+1) = (1-\alpha)S(t) + \alpha \cdot 0 = (1-\alpha)S(t)$$

这种几何衰减使得 $S(t) \rightarrow 0$ 指数级下降。失败的钩子迅速接近零评分。

案例 3：部分可靠的钩子（成功的概率为 p ）

如果修正以概率 $(1-p)$ 发生，则期望更新为：

$$\mathbb{E}[S(t+1)] = (1-\alpha)S(t) + \alpha \cdot p$$

这收敛到 $S^* = p$ 。一个有效率为 80% 的钩子收敛到 $S \approx 0.8$ 。成功分数准确反映了可靠性。

期望修正率：随着钩子的学习和改进，修正率降低。在时间 t 定义修正率：

$$\rho(t) = \frac{\sum_{\tau=1}^t \text{Corr}_\tau}{t}$$

这是需要修正的激活的比例。随着系统的学习， $\rho(t) \rightarrow 0$ 。这是零输入收敛的数学形式化：修正变得极为稀少。

随时间推移的总用户输入变为：

$$U_{\text{total}} = U_0 + \sum_{t=1}^T \text{Corr}_t$$

其中 U_0 是初始种子输入（例如，最初教导钩子），总和表示所有后续修正。由于 $\text{Corr}_t \rightarrow 0$ ，平均每个时间步的输入接近于零：

$$\lim_{T \rightarrow \infty} \frac{U_{\text{total}}}{T} = \lim_{T \rightarrow \infty} \frac{U_0 + \sum_{t=1}^T \text{Corr}_t}{T} = 0$$

这证明了受修正法则支配的系统收敛到零输入操作。

抑制和修剪：修正法则自然通过成功分数创建质量控制。低分的钩子逐渐被边缘化：

抑制阈值 ($S < 0.3$)：

低于此阈值的钩子被抑制——它们仍然存在于系统中，但除非它们的条件非常特定，以至于没有其他钩子匹配，否则不会触发。这防止了糟糕的钩子打扰用户，同时允许在情况变化时潜在恢复。

删除阈值 ($S < 0.1$)：

低于此阈值且最近未使用的钩子（例如，30天以上）会被自动删除以释放资源。这防止了系统无限期地积累无用的钩子。

优先级提升 ($S > 0.9$)：

成功率非常高的钩子在评估中优先考虑，并可能被提升为“受信任”状态，在这种状态下，它们的执行几乎不需要监督或确认。

这种分级系统确保只有高质量的钩子积极影响用户体验，而劣质钩子自然会逐渐消失。

时间衰减：额外维度。除了修正之外，如果钩子最近未被使用，成功率可能会随着时间的推移而缓慢衰减：

$$S(t + \Delta t) = S(t) \cdot e^{-\lambda \Delta t}$$

其中 λ 是一个小的衰减常数（例如，每天0.001）， Δt 是自上次激活以来的时间。这确保系统保持最新——基于旧模式的钩子如果不被使用，逐渐失去影响力，即使它们从未明确失败。

这种时间衰减防止过时的钩子无限期存在。六个月前完美的钩子但现在不再相关，将逐渐降低分数，为更好匹配当前行为的新模式腾出空间。

与最小化法则的关系：修正法则和最小化法则协同工作：

- 最小化法则：选择操作最少的钩子（效率）
- 修正法则：确保所选钩子可靠工作（质量）

它们共同创建了一个二维优化：

$$\text{optimal hook} = \arg \min_{KH} |A(KH)| \text{ subject to } S(KH) > \theta$$

其中 θ 是最低可接受的成功分数。这防止系统选择那些不断失败的超高效钩子。一个 $S=0.3$ 的 1 次行动钩子比一个 $S=0.95$ 的 3 次行动钩子更糟，因为第一个钩子的频繁修正增加了比第二个钩子额外行动更多的用户输入。

实际示例：电子邮件自动回复。考虑一个自动回复某些电子邮件的钩子：

第 1 周：钩子以 $S_0 = 0.5$ （中性）开始。触发 10 次，用户修正 6 次回复（添加细节，改变语气等）：

$$S_1 = 0.5 \cdot (0.7)^6 \cdot (1.3)^4 \approx 0.45$$

由于频繁的修正，分数略有下降。

第 4 周：钩子学习到了更好的模式。触发 10 次，仅 2 次修正：

$$S_4 = 0.45 \cdot (0.7)^2 \cdot (1.3)^8 \approx 0.72$$

随着可靠性的提高，分数上升。

第 12 周：钩子训练得很好。触发 10 次，零次修正：

$$S_{12} = 0.72 \cdot (1.3)^{10} \approx 0.94$$

分数接近 1.0，因为钩子持续产生可接受的回复而无需修正。

与能量和 STC 的关系：修正法则在主观热货币框架中具有直接的经济影响。每次修正都代表了浪费的能量——用户必须付出认知和身体的努力来修复钩子的错误。因此，最小化修正就能最小化能量浪费。

在 STC 中，成功率高（修正较少）的钩子更有价值，因为它们节省更多能源。成功率成为能源效率的直接衡量标准：

$$E_{\text{saved}} = (1 - \rho(t)) \cdot E_{\text{manual}}$$

其中 E_{manual} 是手动执行任务所需的能量， $\rho(t)$ 是修正率。当 $\rho \rightarrow 0$ 时，节能接近全部手动成本。无需修正的钩子节省最大能量，因此具有最大经济价值。

STC 中的专家奖励与成功率挂钩：高成功率钩子的创造者获得相应更多的补偿，因为他们的钩子提供更可靠的节能。

实施考虑事项：实施修正法则需要对几个细节给予仔细关注：

1. 修正检测窗口： Δt 窗口（通常为10-30秒）必须根据不同领域进行调整。快速活动可能需要更短的窗口；有意的任务可能需要更长的窗口。

2. 部分修正：如果用户仅部分修正钩子的动作（例如，编辑3个字段中的1个）怎么办？实现可能会使用分数修正： $\text{Corr}_t \in [0, 1]$ ，基于修改的程度。

3. 假阳性：正常用户活动恰好跟随钩子触发的情况不应被误认为是修正。系统必须理解钩子动作与用户后续动作之间的语义关系。

4. 批量更新：为了提高效率，成功率更新可以批量处理，而不是在每次激活后计算，只要有效的学习动态保持相似。

5. 初始分数：新钩子的初始分数为 $S_0 \in [0.5, 0.7]$ 。预定义的专家钩子可能起始更高（ $S_0=0.8$ ），因为假定质量。学习的钩子初始为中性（ $S_0=0.5$ ），直到它们证明自己。

与贝尔曼方程的比较：修正法则更新在结构上类似于经典强化学习中的贝尔曼方程，但在哲学上是相反的：

贝尔曼方程：

$$V(s) = \max_a \left[r(s, a) + \gamma \sum_{s'} P(s'|s, a) V(s') \right]$$

最大化外部奖励加上折现的未来价值。

修正法则：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

在当前时刻最小化修正（用户输入的负值），不考虑未来折扣。

贝尔曼方程说“最大化期望累积奖励”。修正法则说“最小化用户的即时努力”。这种区别反映了不同的目标：传统的强化学习优化代理在环境中的成功；修正法则优化用户在当前的便利性。

哲学意义：修正法则体现了系统学习方式的深刻转变。传统的机器学习需要标记的训练数据——正确行为的明确示例。修正法则只需要负反馈——被告知何时错误。这是人类在现实生活中常常学习的方式：我们尝试事情，如果没有人纠正我们，我们就假设自己走在正确的轨道上。

这种“沉默即是认可”的范式使学习在用户的视角下变得自然且轻松。用户并不训练系统；他们只是使用它，并偶尔纠正它。学习的负担完全转移到系统上。

此外，修正法则创造了一种不对称性：不良行为受到大声惩罚（立即得分下降），而良好行为则悄然奖励（逐渐得分上升）。这与人类心理学相符——我们对失败的反应比成功更强烈。该法则将这种不对称性形式化为学习动态。

结论：反馈引擎。修正法则是推动知识钩子从初始不可靠走向完美准确的引擎。通过持续的负强化，钩子学习哪些行为有效，哪些无效，逐渐积累反映真实可靠性的成功得分。

结合最小化法则（选择高效钩子）和学习法则（适应条件），修正法则确保基于此代数构建的系统自然演变为零输入操作。有效而无需修正的钩子将占据主导地位；失败的钩子将沉入无名。

这为主观热货币奠定了基础：成功率高的钩子代表经过验证的节能，其创造者应获得与他们提供的价值成比例的补偿。修正法则不仅改善了用户体验——它通过成功分数量化可靠性，从而创造可衡量的经济价值，这些成功分数直接转化为能源效率。

在下一部分，我们将探讨等价法则，它使系统能够识别不同钩子实现相同目标的情况，进一步优化通向零输入技术的路径。

3.4.3 等价法则：识别相同结果

等价法则是知识钩子代数的分类原则——该法则决定何时应将不同的钩子视为功能上相同，尽管它们的实现不同。它指出：具有相同可观察结果的钩子属于同一等价类。该法则使系统能够识别多条路径可以通向同一目的地，从而根据效率、可靠性和上下文在替代方案中进行智能选择。

虽然最小化法则告诉我们要优先考虑较少的动作，而组合法则告诉我们如何构建复杂行为，但等价法则回答了一个更根本的问题：*两个钩子何时在做同样的事情？* 这个问题对于优化、去重和学习至关重要——如果不知道哪些钩子是等价的，系统就无法智能地在它们之间进行选择。

正式声明

等价法则：两个钩子 KH_1 和 KH_2 是等价的，记作 $KH_1 \equiv KH_2$ ，当且仅当它们在相同上下文中执行时产生不可区分的结果：

$$KH_1 \equiv KH_2 \iff \forall C : \text{outcome}(KH_1, C) = \text{outcome}(KH_2, C)$$

其中：

- $\forall C$ 意味着 "在所有上下文中"，两个钩子都可能触发
- $\text{outcome}(KH, C)$ 代表在上下文 C 中执行钩子 KH 后的最终可观察系统状态
- \equiv 表示状态相等（所有可观察属性相同）

关键洞察：等价是由可观察结果定义的，而不是由钩子本身的结构定义的。具有完全不同条件、不同动作序列和不同内部

表示的钩子，如果产生相同的最终状态，仍然可以是等价的。

简单示例：三种打开灯光的方法

考虑三种用于打开客厅灯的钩子：

KH_1 : Conditions: {time=18:00,
location=home} → Actions:
{lights.living_room.state = ON}

 {lights.living_room.brightness = 100%}
 {sunset=true, location=home}

 {darkness_level>80%, location=living_room}
 {turn_on(lights.living_room)}

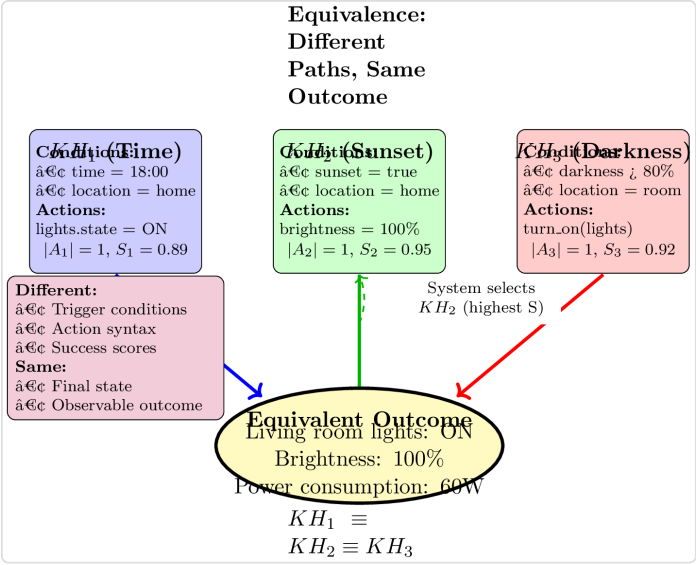
这三种钩子具有：

- 不同的条件：基于时间的 vs. 基于日落的 vs. 基于黑暗的
- 不同的动作规范：state=ON vs. brightness=100% vs. turn_on() 命令
- 不同的触发模式：它们可能在稍微不同的时间触发

然而它们在功能上是等价的——它们都导致客厅灯以全亮度开启：

$$KH_1 \equiv KH_2 \equiv KH_3$$

系统识别这种等价性，并可以根据当前上下文中最可靠满足的条件以及成功率最高的条件智能选择它们。



等价类

等价关系将所有钩子的集合划分为等价类——产生相同结果的钩子组：

$$[KH] = \{KH' \mid KH' \equiv KH\}$$

其中 $[KH]$ 表示包含钩子 KH 的等价类。所有在同一类中的钩子从结果的角度来看是可以互换的，尽管它们可能在以下方面有所不同：

- 效率：所需的操作数量 ($|A|$)

• 可靠性：成功评分 (S)

• 适用性：哪些上下文会触发它们 (R)

• 类型：学习型与预定义型

这种划分在数学上表现良好，因为等价关系满足三个必要属性：

1. 自反性：每个钩子与自身等价：

$$\forall KH : KH \equiv KH$$

2. 对称性：如果 KH_1 与 KH_2 等价，则 KH_2 与 KH_1 等价：

$$KH_1 \equiv KH_2 \implies KH_2 \equiv KH_1$$

3. 传递性：如果 $KH_1 \equiv KH_2$ 和 $KH_2 \equiv KH_3$ ，则 $KH_1 \equiv KH_3$ ：

$$KH_1 \equiv KH_2 \wedge KH_2 \equiv KH_3 \implies KH_1 \equiv KH_3$$

这些属性保证了等价类能够干净地划分钩子空间——每个钩子恰好属于一个等价类，并且类之间从不重叠。

等价性与最小化法则

等价法则与最小化法则密切相关。最小化法则规定：“始终优先选择用户输入最少的路径。”当多个钩子等价（结果相同）时，此法则要求选择最有效的一个：

$$\text{if } KH_1 \equiv KH_2 \text{ and } |A_1| < |A_2| \implies \text{prefer } KH_1$$

这产生了一种自然的优化压力：系统向实现每个结果的最有效方式演变，自动修剪或降级效率低下的等价钩子。

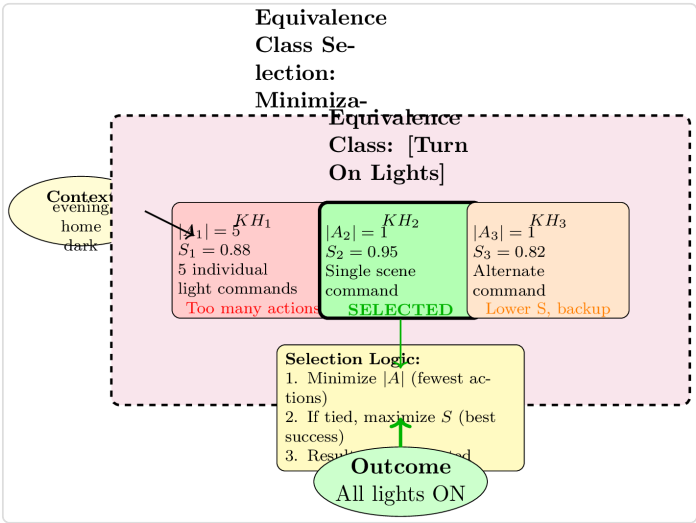
选择算法：当多个等价钩子可以在给定上下文中触发时，系统使用以下优先级：

$$KH^* = \arg \min_{KH \in [KH_{\text{equiv}}]} |A|$$

选择动作最少的钩子。如果动作数量相同：

$$\text{if } |A_1| = |A_2| \implies KH^* = \arg \max_{KH} S$$

选择成功评分最高的钩子。



检测等价性

确定两个钩子是否等价并不总是简单，特别是当动作使用不同的 API 或命令语法时。系统采用多种策略：

1. 语法分析：比较动作中指定的最终状态。如果 KH_1 设置 ``lights.state = ON`` 而 KH_2 设置 ``lights.brightness = 100%``，如果系统理解 ON 意味着 100% 亮度，则会将其视为等价。

2. 语义分析：利用关于设备及其状态的领域知识。系统知道 ``turn_on(lights)``、``set_brightness(lights, 100)`` 和 ``set_state(lights, ON)`` 都产生相同的结果。

3. 实证观察：在受控环境中执行两个钩子，并观察结果状态是否相同。如果钩子始终从相同的 C_t 产生相同的 C_{t+1} ，则它们在实证上是等价的。

4. 用户注释：允许专家在创建预定义钩子时明确标记钩子为等价，为系统提供学习的基础真相。

上下文相关的等价性

一个重要的细微差别：两个钩子在某些上下文中可能是等价的，但在其他上下文中则不是。考虑：

KH_A ：如果 {morning} → 打开灯 KH_B ：
如果 {morning} → 打开窗帘

在夏季，这些是等价的（当打开窗帘提供足够的光线时），但在冬季则不等价（当窗帘无法提供足够的光线时）。等价关系可以根据上下文进行参数化：

$$KH_1 \equiv_C KH_2 \iff \text{outcome}(KH_1, C) = \text{outcome}(KH_2, C)$$

系统跟踪这些条件等价性，以便做出适合上下文的选择。

等价性和成功评分

有趣的是，等价的钩子可能具有不同的成功评分，即使它们产生相同的结果。为什么？因为成功评分衡量的是可靠性和纠正频率，而不仅仅是结果：

- 钩子 A 可能实现结果，但偶尔需要纠正
- 钩子 B 以更少的纠正实现相同的结果

尽管它们是等价的（相同的最终状态），钩子 B 更可靠，并且将具有更高的成功评分。当两者都可以触发时，系统将更倾向于钩子 B，逐渐学习哪个等价方法最可靠。

等价的实际应用

理解等价性有几个实际好处：

1. 通过选择优化：当多个等价钩子可以触发时，系统可以智能地选择最有效的一个，直接服务于最小化法则。

2. 去重：如果系统检测到在相同等价类中具有相似条件的多个钩子，它们可能会被合并，或者可以禁用一个以减少冗余：

$$\text{if } KH_1 \equiv KH_2 \wedge R_1 \approx R_2 \implies \text{merge}(KH_1, KH_2)$$

这防止系统维护多个几乎相同的钩子。

3. 通过冗余增强鲁棒性：拥有多个具有不同条件的等效钩子提供了弹性。如果一个钩子的条件变得不可满足（传感器损坏，服务离线），其他等效钩子仍然可以通过替代方式实现相同的结果。

4. 学习替代路径：当系统观察到用户在不同上下文中通过不同方法实现相同结果时，它会将这些视为等效方法，并可以选择上下文适当的变体。

5. 钩子迁移：在升级设备或服务时，可以自动迁移等效钩子。如果您用新的智能灯替换旧灯，控制旧灯的钩子可以更新为新灯的等效钩子，从而保留功能。

6. A/B 测试：系统可以对等效钩子进行实验，以确定在实践中哪个最可靠，逐渐将使用转向每个等价类中表现最佳的选项。

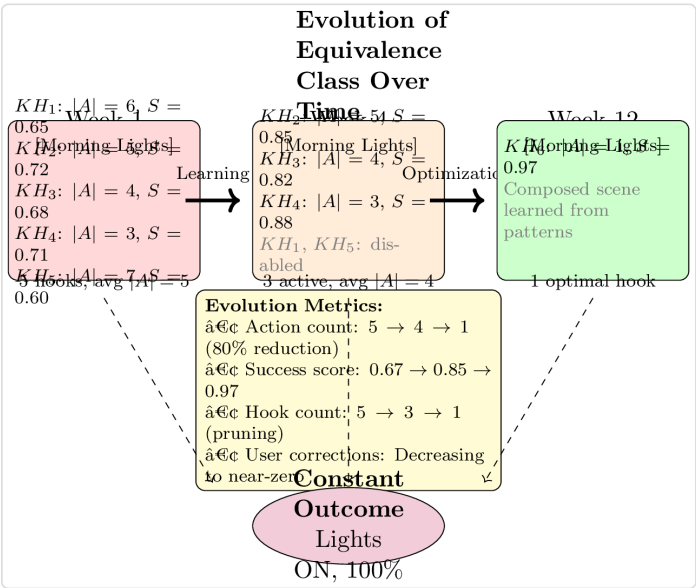
通过等价类的系统演变

随着时间的推移，系统的钩子集合自然组织成等价类。在每个类中：

- 钩子根据效率（ $|A|$ ）和可靠性（ S ）进行竞争
- 表现最佳的钩子积累使用量和更高的成功分数
- 表现不佳的钩

子被降级或最终禁用 • 系统学习哪些条件最可靠地预测每个结果何时被期望

这产生了朝着最佳钩子集的进化压力——一个通过最有效、可靠的方式实现所有期望结果的钩子集，并在条件变化时具有适当的冗余。



与其他法则的关系

等价法则与代数中的所有其他法则相互作用：

- 与最小化法则：等价定义了钩子具有 "相同结果" 的含义，这是比较它们的操作计数的前提。我们只能在我们知道是等效的替代方案中进行最小化。
- 纠正法则：当一个等效的钩子需要纠正时，它的成功评分相对于同类其他钩子会降低。随着时间的推移，最可靠的等效钩子将占据主导地位。
- 组合性法则：复合钩子可以等同于原始钩子的序列。一个单一的“早晨例行”钩子等同于手动依次调用五个独立的钩子——

但效率更高。

- 学习法则：随着系统的学习，它发现新的钩子，这些钩子等同于现有的钩子，但更高效或更可靠。学习创造了等效类的新成员，不断提高系统的能力。

设计影响

等效法则在重要方面塑造系统架构：

1. 等效检测基础设施：建立强大的机制，通过句法、语义和经验分析来检测等效性。这是支持所有优化的关键基础设施。

2. 等效类可视化：为用户提供等效类的视图，展示哪些钩子实现相同的结果，以及它们在效率和可靠性上的比较。

3. 自动合并：当等效类中的多个钩子具有相似条件，并且一个在 $|A|$ 和 S 上都占主导地位时，自动建议合并或弃用劣质替代品。

4. 上下文感知选择：跟踪等效类中哪些钩子在不同上下文中表现最佳，并相应选择。夏季与冬季、在家与外出、工作日与周末——不同的上下文可能偏向不同的等效钩子。

5. 优雅降级：保持多个等效钩子作为后备。如果主要钩子失败（传感器离线，API不可用），自动尝试等效替代品。

结论：等价性作为优化的推动者

等价法则使知识钩系统能够智能地优化和适应。如果没有等价的概念，系统将无法识别不同的方法实现相同的目标，因此也就没有原则性的方法来选择它们。

等价将可能的钩子空间划分为基于结果的类别，使系统能够：

- 公平比较替代方案（它们必须产生相同的结果才能进行比较）
- 高效选择（在每个类别中选择最小的 $|A|$ ）
- 稳健学习（发现通往每个期望结果的多条路径）
- 智能进化（持续改进以实现最佳钩子）
- 提供冗余（在每个类别中保持备用选项）

这不仅仅是一个实现细节——它是使主观技术成为可能的基本原则。能够识别尽管实现不同但结果相同的能力，使系统能够学习一般模式，而不是记忆具体示例，从现有组件中组合新解决方案，并持续优化以实现零输入交互。

与最小化法则（定义优化内容）、纠正法则（提供学习信号）、可组合性法则（使构建复杂性成为可能）和学习法则（使适应成为可能）一起，等价法则完成了真正学习成为你的技术的正式基础——这种技术能够识别你想要的内容，无论你怎么实现它，并持续寻找更好的方法来实现它。

3.4.4 可组合性法则：从简单性构建复杂性

可组合性法则是知识钩代数的架构原则——使简单钩子能够组合成任意复杂行为而不牺牲模块化、可维护性或可理解性的法则。它指出：钩子可以通过组合操作组合成更大、更复杂的结构，复合钩子的行为基于其组件表现出可预测性。

虽然最小化法则定义了我们优化的内容，纠正法则提供了学习机制，但可组合性法则决定了我们构建复杂系统的*方式*。这就像拥有一组孤立的工具与拥有一个组件无缝协作的集成生态系统之间的区别。

正式声明

可组合性法则：给定钩子 h_1, h_2, \dots, h_n 和组合操作 \circ （嵌套）和 \oplus （平面），所得到的复合钩子 h_c 是明确定义的，并表现出以下特性：

1. 闭包：组合有效的钩子会产生有效的钩子
2. 可预测性： h_c 的行为可以从其组件的行为中推导出来
3. 模块化：对组件钩子的更改会自动传播到组合体
4. 保持性：组合保持正确性——如果组件是正确的，组合体就是正确的

组合的两种形式

代数定义了两种互补的组合形式，各自满足不同的架构需求：

1. 嵌套组合（层次结构）

嵌套组合允许钩子在其条件或动作中引用其他钩子，创建高层抽象建立在低层原语之上的层次结构。

一个钩子 $KH_i = (R_i, A_i, T_i, S_i)$ 可以通过两种方式包含对其他钩子 KH_j 的引用：

条件中的子钩子：一个条件可以检查另一个钩子是否会激活：

$$R_i = \{r_1, r_2, \dots, r_k, \text{fires}(KH_j, C_t), \dots, r_n\}$$

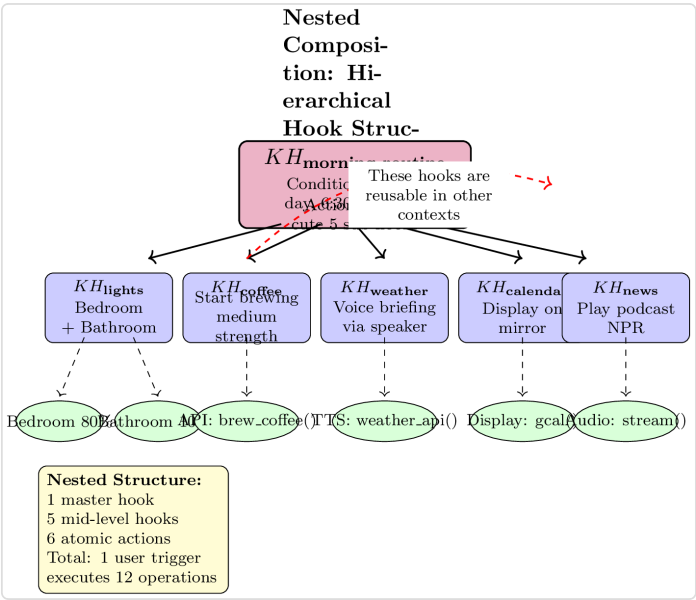
在当前上下文 C_t 中，如果钩子 KH_j 会激活，则 $\text{fires}(KH_j, C_t)$ 评估为真。

动作中的子钩子：一个动作可以调用另一个钩子的执行：

$$A_i = (a_1, a_2, \dots, \text{execute}(KH_j), \dots, a_k)$$

在动作序列中， $\text{execute}(KH_j)$ 触发钩子 KH_j 。

这创建了一个树状结构，其中叶子钩子执行原子操作，而内部节点协调复杂行为。



示例：嵌套的早晨例程

考虑一个高层次的早晨例程钩子，它协调多个子钩子：

$$KH_{\text{morning_routine}} = (R, A, T, S)$$

，其中：

条件 R ：

• Alarm dismissed at 6:30 AM • Current day is weekday • User location = home

操作 A ：

1. $\text{execute}(KH_{\text{bedroom_lights}})$ → 将灯光逐渐调至80% 2. $\text{execute}(KH_{\text{coffee_maker}})$ → 开始煮咖啡 3. $\text{execute}(KH_{\text{weather_briefing}})$ → 通过扬声器宣布天气 4. $\text{execute}(KH_{\text{calendar_sync}})$ → 在镜子上显示日历 5. $\text{execute}(KH_{\text{news_summary}})$ → 播放新闻播客

每个子钩子都是独立有用且可重用的。卧室灯光钩子也可以在晚间阅读例程中调用。咖啡机钩子可以在客人到达时触发。天气简报可能是“离家”例程的一部分。这种模块化意味着钩子不需要被重新创建——它们可以组合。

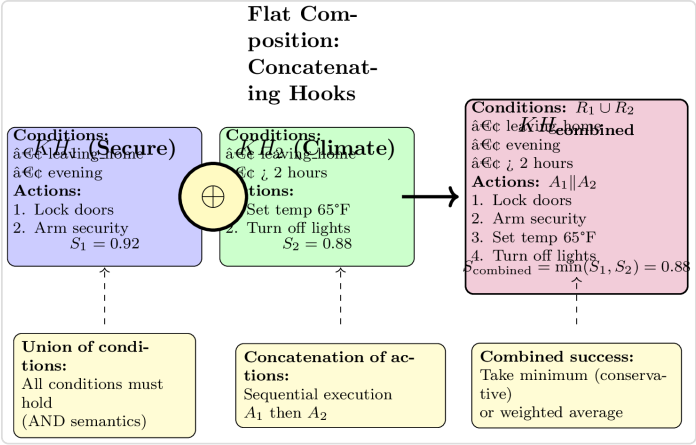
2. 平面组合（连接）

平面组合通过合并多个钩子的组件来组合它们。给定两个钩子 $KH_1 = (R_1, A_1, T_1, S_1)$ 和 $KH_2 = (R_2, A_2, T_2, S_2)$ ，平面组合创建：

$$KH_{\text{combined}} = KH_1 \oplus KH_2 = (R_1 \cup R_2, A_1 \parallel A_2, T_{\text{derived}}, S_{\text{combined}})$$

其中：

- $R_1 \cup R_2$ 是条件的并集（两个钩子的所有条件必须满足）
- $A_1 \parallel A_2$ 是动作序列的连接（先执行 A_1 然后 A_2 ）
- T_{derived} 通常是组合钩子“学习”的
- S_{combined} 是根据组件成功分数计算的（通常是最小值或加权平均值）



示例：晚间离家例程

假设您有两个独立的晚间离家钩：

KH_{secure} ：晚上离开家时 → 锁门，启动安全系统
 ($S = 0.92$)

KH_{climate} ：离开家超过2小时 → 将温度设置为
 65°F，关闭灯光 ($S = 0.88$)

扁平组合将它们合并：

$$KH_{\text{depart}} = KH_{\text{secure}} \oplus KH_{\text{climate}}$$

结果是一个具有组合条件（离开家、晚上、超过2小时）和
 连接动作（锁门、启动、温度、灯光）的单一钩，成功评分为
 $S = \min(0.92, 0.88) = 0.88$ 。

组合的代数性质

组合操作满足几个代数性质，使其在数学上表现良好：

1. 结合性（扁平组合）：

$$(KH_1 \oplus KH_2) \oplus KH_3 = KH_1 \oplus (KH_2 \oplus KH_3)$$

组合的分组顺序不会影响最终结果（尽管动作的执行顺序确实很重要）。

2. 身份：存在一个身份钩子

$KH_\emptyset = (\emptyset, (), \text{predefined}, 1.0)$ ，使得：

$$KH \oplus KH_\emptyset = KH_\emptyset \oplus KH = KH$$

与空钩子组合不会改变原始钩子。

3. 模块化：嵌套组合保持模块化——更改子钩子会自动影响所有引用它的钩子，从而通过一次更改实现系统范围的更新。

4. 闭包：有效钩子的组合始终是有效钩子。代数在组合下是封闭的。

5. 成功分数单调性：对于平面组合，

$$S_{\text{combined}} \leq \min(S_1, S_2, \dots, S_n)。$$

一个复合体的可靠性仅与其最弱的组成部分相同。

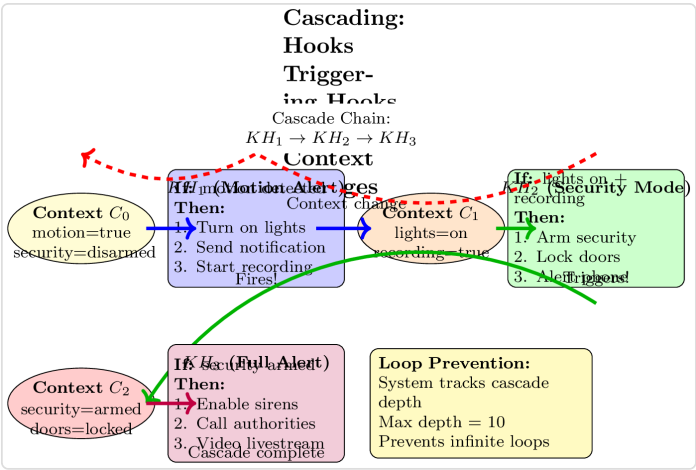
级联：通过上下文变化的组合

当一个钩子的动作以触发其他钩子的方式修改上下文时，就会发生一种特殊形式的组合。这会创建一个级联——钩子激活的连锁反应。

正式来说，级联发生在：

$$\exists a_i \in A_1 : \text{execute}(a_i) \implies C_t \rightarrow C_{t+1} \implies \text{fires}(KH_2, C_{t+1})$$

钩子 1 的一个动作修改上下文，使得钩子 2 的条件满足。



示例：安全级联

一个运动检测钩子打开灯并开始录制。这个上下文变化（灯=开，录制=真）触发了一个安全模式钩子，该钩子使系统处于警戒状态并锁上门。这反过来又触发了一个警报钩子，通知当局。三个钩子依次触发，但只有第一个是直接由用户的运动触发的，其余的都是自动级联的。

无限循环防止：系统必须跟踪级联深度并防止无限循环。最大深度（通常为10）确保级联终止。如果 KH_1 触发 KH_2 ，然后再次触发 KH_1 ，系统会检测到循环并打破它。

可组合性与最小化法则

组合是实现最小化的主要机制。通过从低级组件构建更高级的钩子，我们减少了用户必须明确执行的操作数量。

考虑以下进展：

没有组合：用户每天早上必须执行12个单独的操作（打开每个灯，开始咖啡，检查天气等）。总计： $|A| = 12$ 个明确输入。

使用低级钩子：用户说“打开灯”，“咖啡”，“天气”。系统执行多步骤序列。总计： $|A| = 5$ 个输入。

使用组合钩子：用户触发警报解除。系统通过嵌套组合执行整个早晨例程。总计： $|A| = 1$ 个输入。

使用学习的上下文：系统自动检测早晨上下文。钩子在没有任何用户输入的情况下触发。总计： $|A| = 0$ 。

组合通过允许复杂行为作为单个高级操作来表达，从而实现这种进展。每个组合层级减少用户所需的操作数量，直接服务于最小化法则。

可组合性的实际好处

组合性法则为现实世界系统提供了几个关键的好处：

1. 可重用性：一次编写一个钩子，通过组合在许多上下文中使用。一个“关闭灯光”的钩子可以在离开例程、睡眠例程、看电影例程等中重用。

2. 可维护性：更新一个低级钩子，所有使用它的组合钩子都会自动受益。修复咖啡冲泡钩子中的错误，所有制作咖啡的例程都会立即改善。

3. 抽象：高级钩子可以表达意图，而无需指定实现细节。“为睡眠准备房子”不需要列举每个灯开关——它是从适当的子钩子组合而来的。

4. 可扩展性：复杂的行为从简单的组件中产生，而不会导致指数级的复杂性。一个拥有100个原始钩子的系统可以通过组合表达数百万种组合行为。

5. 可学习性：系统可以通过组合先前学习的低级行为来学习高级模式。它不是从头开始，而是建立在现有知识之上。

6. 可解释性：嵌套组合保持可解释性。用户可以通过检查其组件来理解钩子的功能，形成自然的解释层次结构。

设计影响

组合性法则塑造了知识钩子系统的架构方式：

1. 优先考虑小而专注的钩子：设计能够很好地完成一项工作的原始钩子。复杂性应来自组合，而不是来自单个钩子做太多事情。

2. 鼓励层次化组织：提供工具和可视化，使钩子层次结构可见且可导航。用户应该能够看到并理解组合结构。

3. 自动组合检测：当系统观察到用户反复调用相同的钩子序列时，它应该建议创建一个组合钩子以减少未来的输入。

4. 安全级联：实现循环检测和深度限制，以防止无限循环。记录级联链以便调试和优化。

5. 组合感知学习：在学习新钩子时，检查它们是否可以作为现有钩子的组合来表达，而不是创建冗余的原始钩子。

6. 版本控制和依赖管理：当子钩子被更新时，跟踪哪些组合钩子依赖于它。提供测试和验证机制，以确保更新不会破坏组合。

与其他法则的关系

组合性法则与其他基本法则协同工作：

• 与最小化法则：组合是减少 $|A|$ 的主要机制。通过构建钩子的层次结构，我们以最小的用户输入表达复杂行为。

- 与修正法则：当组合钩子需要修正时，系统必须确定哪个组件失败。这使得有针对性的学习和改进成为可能。

- 与等价法则：即使两个组合钩子具有不同的内部结构，它们也可以是等价的。等价性由结果决定，而不是组合路径。

- 与学习法则：系统不仅学习原始模式，还学习哪些钩子的组合能够很好地协同工作。关于有效组合策略的元模式出现。

结论：组合作为架构基础

组合性法则将知识钩从聪明的单个自动化转变为真正的代数——一个具有封闭性、可预测性和涌现能力的正式系统。如果没有组合性，我们将拥有一系列孤立的技巧。有了它，我们就拥有一个可以无限增长复杂性的可扩展架构。

组合使系统能够构建抽象，重用知识，分层学习，并在保持可理解性的同时表达任意复杂的行为。它是简单反应性自动化与真正智能之间的桥梁。

最重要的是，组合是实现最小化法则零输入技术承诺的机制。每个组合层级减少用户的负担，使我们更接近于那些在您表达之前就知道您想要什么的系统，那些在未被要求的情况下代表您行动的系统，那些以最小摩擦将您的意志扩展到世界的系统。

这不仅仅是软件工程的最佳实践——它是主观技术本身的架构基础。能够可预测和强有力地组合钩子是使整个代数成为可能的原因，也是最终使技术学会成为您的关键。

3.4.5 学习法则：通过上下文增量进行持续适应

学习法则是知识钩代数的适应原则——使系统能够随着时间的推移而演变、改进和个性化，而无需明确编程。它指出：上下文增量累积成加权条件，随着时间的推移精炼钩子。该法则将

知识钩从静态自动化转变为从每次交互中学习的生动、演变的模式。

虽然最小化法则定义了我们优化的目标，纠正法则提供反馈信号，而组合性法则使构建复杂性成为可能，但学习法则使系统具有适应性。它是主观技术逐渐与每个用户更加一致的机制，学习模式、偏好和上下文，而无需明确配置。

正式声明

学习法则：当用户在上下文 C_t 中执行操作时，系统捕获前后的快照并提取增量。该增量累积到学习钩的条件中，这些条件通过成功分数加权，并通过重复观察随着时间的推移进行精炼：

$$\Delta_t = \Sigma_{\text{after}}(t) - \Sigma_{\text{before}}(t)$$

其中：

- $\Sigma_{\text{before}}(t)$ 是在时间 t 时用户输入之前的完整上下文快照
- $\Sigma_{\text{after}}(t)$ 是在用户输入完成后立即的完整上下文快照
- Δ_t 是代表用户操作结果所发生变化的增量

从这个增量中，系统通过模式提取创建或更新钩子：

$$KH_{\text{learned}} = (R_{\text{extracted}}, A_{\Delta}, \text{learned}, S_0)$$

其中：

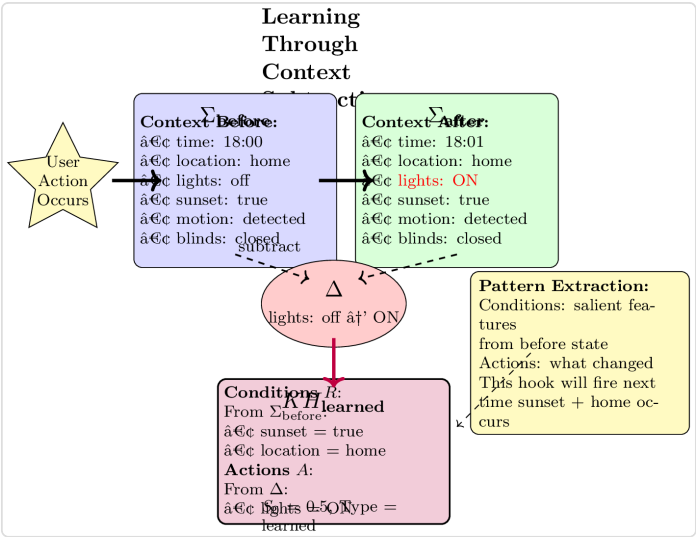
- $R_{\text{extracted}}$ 是从 Σ_{before} 中提取的条件（用户行为时存在的模式）
- A_{Δ} 是从 Δ_t 中提取的动作（用户输入导致的变化）
- 类型设置为 "learned"
- S_0 是初始成功评分（通常为 0.5）

核心机制：上下文减法

上下文减法是实现学习的基本操作。与传统的机器学习系统需要标记的训练数据不同，知识钩子通过观察用户行为前后的世界进行学习。差异——增量——准确揭示用户想要实现的目标。

这与显式编程有着根本的不同。你从不告诉系统 "当条件 X、Y、Z 满足时，执行动作 A、B、C。" 相反，你只是行动，系统观察：

- 之前：你决定行动时的上下文是什么？
- 之后：你的行动导致了什么变化？
- 推断：如果再次出现类似的上下文，你可能想要类似的变化



模式提取：从增量到条件

学习中的关键挑战是确定 Σ_{before} 的哪些特征是相关条件，哪些仅仅是偶然的。如果你在日落时打开灯，同时穿着红色衬衫，系统必须学习到 "日落" 是相关的，但 "红色衬衫" 不是。

系统使用几种策略进行模式提取：

1. 显著性检测：识别哪些上下文特征与动作最强相关。一天中的时间、地点和设备状态通常比衣服颜色或环境噪音更显著。

2. 时间模式：提取重复的基于时间的模式（每个工作日的早上 6:30，日落时的 18:00，月底）。

3. 顺序模式：识别在行动之前的事件序列（打开电子邮件 → 阅读2分钟 → 存档）。

4. 环境模式：检测基于位置的触发器（到家、进入办公室、靠近杂货店）。

5. 设备状态模式：识别相关的设备配置（手机电量低、日历显示会议、正在播放音乐）。

最初，学习的钩子往往过于具体，包括许多可能不相关的条件。学习法则通过经验观察使得随着时间的推移进行精炼。

加权条件：积累证据

随着系统观察到类似行为的重复实例，条件积累代表其重要性的权重。每次行动发生时出现在 Σ_{before} 中的条件比偶尔出现的条件权重更高。

正式地，每个条件 $r_i \in R$ 都有一个相关的权重 w_i ，该权重随着成功激活而增加：

$$w_i(t+1) = w_i(t) + \alpha \cdot \mathbb{I}[r_i \text{ present in } \Sigma_{\text{before}} \wedge \text{no correction}]$$

其中：

- α 是学习率（通常为0.1）
- $\mathbb{I}[\cdot]$ 是指示函数（条件为真时为1，否则为0）
- 当条件存在且钩子成功而没有修正时，权重增加。

权重低的条件（在成功激活期间很少出现）可以被修剪，从而将钩子精炼到其基本触发器。

示例：学习早晨例行程序

考虑系统如何在几天内学习你的早晨例行程序：

第1天（星期一，早上6:30）：

Σ_{before} : {alarm_dismissed=true,
time=06:30, day=Monday, location=bedroom,
lights=off, temperature=68°F,
wearing_pajamas=true}

你手动打开灯，开始煮咖啡，查看天气。

{lights=on, coffee_maker=brewing,
weather_app=opened}

系统创建了一个包含所有观察到的条件的钩子，非常具体。

第2天（星期二，早上6:30）：

Σ_{before} : {alarm_dismissed=true,
time=06:30, day=Tuesday, location=bedroom,
lights=off, temperature=70°F,
wearing_pajamas=false}

钩子自动触发（条件大致匹配）。无需修正。成功评分增加： $S = 0.5 \rightarrow 0.6$ 。

权重更新：

• alarm_dismissed: $w \uparrow$ （已解除） •
time=06:30: $w \uparrow$ （已解除） • day=Monday: w
未更改（未出现） • location=bedroom: $w \uparrow$ （已解

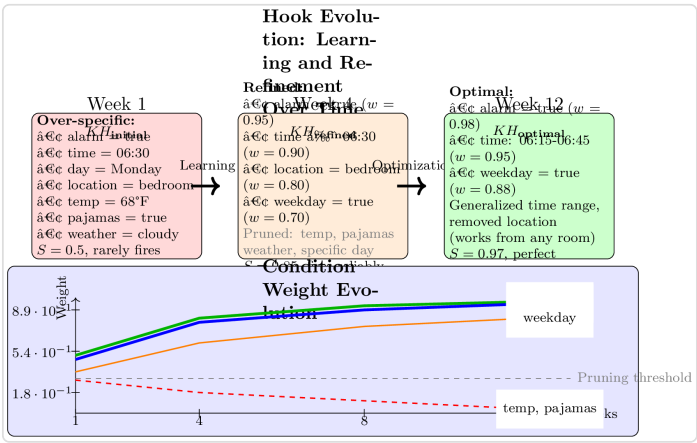
除) • temperature=68°F: w 未更改 (值不同) • wearing_pajamas=true: w 未更改 (未出现)

第5天 (星期五, 上午6:30) :

经过几次成功激活后, 重量分布变得清晰 :

• alarm_dismissed: $w = 0.95$ (高 - 总是存在) • time=06:30: $w = 0.90$ (高 - 总是存在) • location=bedroom: $w = 0.85$ (高 - 通常存在) • day=specific: $w = 0.15$ (低 - 变化) • temperature=exact: $w = 0.05$ (非常低 - 变化) • pajamas: $w = 0.02$ (非常低 - 无关紧要)

系统通过 $w < 0.3$ 修剪条件, 将钩子精炼为基本触发器 : {alarm_dismissed=true, time≈06:30, location=bedroom}



通过反馈进行持续精炼

学习法则持续运行, 而不仅仅是在初始钩子创建期间。每当钩子触发时, 系统观察其是否成功 (无修正) 或失败 (用户修正)。这种反馈推动持续的精炼 :

如果没有发生修正：

• 成功评分增加：

$$S(t+1) = S(t) + \alpha(1 - S(t))$$

当前特征的条件权重增加 • 钩子变得更可信，触发得更自信

如果发生修正：

• 成功分数降低：

$$S(t+1) = (1 - \alpha)S(t)$$

系统检查 Σ_{before} 以查看有什么不同 • 可能添加新条件（细化）或删除过于具体的条件（概括） • 权重重新分配以强调区分特征

这创造了一个持续的反馈循环，其中钩子朝着最佳特异性演变——足够具体以避免误报，足够一般以可靠地激活。

防止钩子扩散

并非每个用户操作都应该创建一个新钩子。系统应用多个阈值以防止无限增长：

1. 重要性阈值：仅对涉及重大变化的操作创建钩子（影响超过3个实体或设备）。

2. 重复阈值：在相似上下文中执行相同操作模式2-3次后再确定钩子。

3. 相似性检查：在创建新钩子之前，检查是否已经存在相似的钩子：

$$\text{similarity}(KH_1, KH_2) = \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|} \cdot \frac{|A_1 \cap A_2|}{|A_1 \cup A_2|}$$

如果相似性 > 0.7，更新现有钩子而不是创建重复项。

4. 用户意图检测：使用启发式方法区分有意图的模式和随机的一次性操作。紧接着撤销的单个操作不是学习机会。

与其他法则的关系

学习法则与代数中的所有其他法则集成：

- 最小化法则：学习创建自动化重复操作的钩子，随着时间的推移直接减少 $|A|$ 。每一个学习到的模式都减少了一个显式输入的需求。

- 修正法则：修正提供了驱动学习的反馈信号。缺乏修正验证了学习到的模式；其存在则触发了精炼。

- 等价法则：系统学习到不同的方法可以实现相同的结果，即使它们来自不同的学习实例，也能识别出等价的钩子。

- 可组合性法则：学习可以在多个层次上发生——原始动作组合成序列，序列组合成例程。系统学习到层次模式。

设计影响

学习法则在根本上塑造了系统设计：

1. 丰富的上下文捕获：投资于全面的上下文感知——上下文越丰富，模式提取越好。时间、地点、设备状态、环境传感器、日历、最近的操作——都提供了有价值的学习信号。

2. 高效的增量计算：上下文减法必须快速且准确。使用结构化表示（而非原始像素差异），以便进行智能比较。

3. 增量学习：不要等到积累大量数据集。立即从每次交互中学习，实时更新钩子。

4. 遗忘机制：对未使用的钩子和条件实施时间衰减。过去有效但现在不再发生的模式应优雅地消退。

5. 用户可见性：向用户展示系统正在学习的内容。显示最近创建的钩子、它们的条件和成功率。允许手动编辑或删除学习到的模式。

6. 隐私保护学习：所有学习都在用户设备上本地进行。上下文快照永远不会离开用户的个人生态系统。这对维护信任至关重要。

学习轨迹

在长期使用中，学习法则产生了特征性的进化轨迹：

第一阶段（第1-7天）：快速创建钩子。系统学习明显的模式。创建了许多钩子，有些过于具体。成功评分变化不定。用户提供频繁的纠正。

第二阶段（第2-4周）：精炼期。钩子通过重复使用进行概括。无关条件被修剪。成功评分稳定。纠正显著减少。

第三阶段（第2-3个月）：组合出现。系统识别模式的模式。更高层次的钩子由成功序列形成。复杂工作流程的自动化例程变得复杂。

第四阶段（第4个月及以后）：成熟适应。系统在需求出现之前进行预测。大多数交互不需要明确的输入。纠正很少，通常针对新情况。技术已经学会成为你。

这个轨迹不是理论上的——这是学习法则在时间上持续应用的自然结果。系统逐渐与每个用户的模式、偏好和上下文更加一致。

结论：学习作为主观性的基础

学习法则使主观技术成为可能。没有学习，我们将拥有静态自动化——也许有用，但不具适应性。没有对上下文变化的持续学习，系统永远无法进化以匹配你的独特模式。

学习将技术从一个你可以控制的工具转变为一个能够预测你需求的自我延伸。这是外部设备变得主观的机制——与您的意图保持一致，响应您的上下文，预测您的愿望。

加权条件的积累创建了您模式的丰富模型——不是通过明确的编程或配置，而是通过持续的观察和适应。每个增量都为这个模型增添了内容，使其更加精确、更加个性化，更加符合您。

与最小化法则（定义目标）、修正法则（提供反馈）、等价法则（启用比较）和组合法则（构建复杂性）一起，学习法则完成了知识钩代数的正式基础。这五条法则协同工作，创建出不仅仅响应命令的系统——它们学习模式，预测需求，组合解决方案，并持续优化以实现零输入技术的理想。

这不是传统意义上的人工智能。它是更根本的东西：自适应智能——技术根据您自身的需求进行调整，而不是强迫您去适应它。而这一切都得益于学习法则这一简单而深刻的原则：在每个行动之前和之后观察上下文，提取增量，积累证据，并持续精炼。

3.5 定理和证明：代数的形式属性

我们现在已经建立了知识钩代数的完整操作框架：钩子的数学结构作为四元组，转换和组合它们的操作，以及五条基本法则来治理它们的行为。但仅有操作和法则，虽然必要，却不足以完全表征一个数学系统。为了证明知识钩代数是严谨的、可预测的和良好行为的，我们必须证明关于其属性的形式定理。

定理是任何代数的顶石。它们回答关于系统行为的关键问题，这些问题仅通过定义无法解决：系统是否收敛？它是否稳定？操作是否明确定义？法律是否保证了理想的属性？这些问题需要证明——从公理和定义中推导结论的正式逻辑论证，具有数学的确定性。

为什么定理很重要

我们在这里提出的定理服务于几个基本目的：

1. 可预测性：定理证明系统以可预测、明确定义的方式运行。它们将直觉（“钩子应该收敛到最佳行为”）转化为严格的保证（“在这些条件下，成功评分以概率1收敛到真实可靠性”）。

2. 最优性：定理证明了优先级算法和最小化法则实际上实现了最优选择。我们不仅声称系统选择高效——我们从数学上证明了这一点。

3. 收敛性：或许最重要的是，定理证明知识钩系统收敛于零输入操作。这不是一个理想目标或经验观察——这是一个从代数结构中得出的数学确定性。

4. 健全性：定理证明操作保持重要属性。组合有效的钩子会产生有效的钩子。等效的钩子在细化下仍然保持等效。系统不能进入无效或未定义的状态。

5. 与物理的联系：对于我们更大的项目来说，定理建立了代数与热力学之间的正式联系。它们证明最小化用户操作在数学上对应于最小化能量消耗，为主观热货币提供了严格的基础。

我们将证明的定理

本节介绍关于知识钩代数的七个主要定理，每个定理都有正式证明。这些定理建立了主观技术的完整数学基础：

定理 1：零输入收敛定理 证明受五个法则支配的系统在时间接近无穷大时收敛于零用户输入。正式建立了

$$\lim_{t \rightarrow \infty} \mathbb{E}[|A_{\text{user}}(t)|] = 0.$$

定理 2：成功评分收敛定理 证明成功评分收敛于钩子的真实可靠性。一个以概率 p 成功的钩子几乎肯定会有

$$\lim_{t \rightarrow \infty} S(t) = p.$$

定理 3：优先级最优性定理 证明优先级算法总是从任何等价类中选择最优钩子——即具有最小操作的钩子，并且在平局时选择最大成功评分的钩子。

定理 4：组合闭合定理 证明代数在组合下是闭合的：组合有效的钩子（无论是嵌套还是平坦）总是产生具有良好定义行为的有效钩子。

定理 5：等价划分定理 证明了等价关系将钩空间正确地划分为不相交的类，并且这种划分尊重代数运算。

定理 6：权重收敛定理 证明了学习到的钩中的加权条件收敛于上下文特征的真实相关性。无关特征随着时间的推移被证明会被修剪。

定理 7：能量最小化等价定理 证明了最小化用户行为在数学上等价于最小化能量消耗，建立了与热力学和 STC 的正式桥梁。

数学框架和符号

我们的证明将使用来自概率论、分析和抽象代数的标准数学技术。我们采用以下约定：

- 几乎必然收敛：我们写 $X_t \rightarrow X$ a.s. 意味着 $P(\lim_{t \rightarrow \infty} X_t = X) = 1$

- 期望值： $\mathbb{E}[X]$ 表示随机变量 X 的期望值

- 大 O 符号： $f(t) = O(g(t))$ 意味着 f 在渐近上不比 g 增长得快

- 指示函数： $\mathbb{1}[P]$ 等于 1 如果命题 P 为真，否则为 0

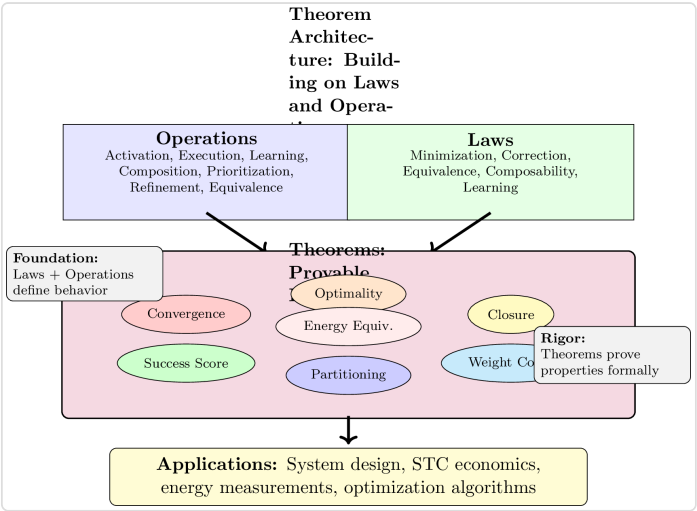
• 序列：下标 t 表示时间步，因此 S_t 是时间 t 的成功分数

证明策略

每个定理都有一个结构化的证明：

- 1. 陈述：我们要证明的正式陈述，所有假设都明确
- 2. 直觉：用简单的语言解释为什么这个定理应该成立
- 3. 证明：从公理和定义推导结论的严谨逻辑论证
- 4. 影响：讨论该定理对系统设计和行为的意义

证明在严谨性与可及性之间取得平衡。我们提供完整的正式论证，但同时附上直观的解释和具体的例子。目标不仅是证明结果，更是阐明它们为何成立及其意义。



与推论的联系

在定理之后，我们将呈现推论——从主要定理中轻松得出的结果，无需独立证明。推论捕捉重要的特殊情况，推导有用的公

式，并将定理与实际应用联系起来。虽然定理确立了代数的基本属性，但推论使这些属性变得可操作。

对主观技术的影响

这些定理不仅仅是抽象数学。每个定理对构建真实的主观技术系统都有直接的影响：

- 收敛定理保证系统实际上能够实现零输入操作，而不仅仅是渐近接近。这是整个范式的数学基础。
- 最优性定理确保优先级算法可以高效实现，并且总是做出最佳选择。无需进行穷举搜索或启发式方法。
- 闭合定理意味着开发者可以自由组合钩子，而不必担心创建无效状态。代数保证了健全性。
- 能量等价定理提供了与热力学的严格联系，使整个 STC 的经济框架成为可能。能源节约可以被测量、验证和货币化，因为与物理学的联系在数学上是被证明的。

本节结构

在接下来的子章节中，我们详细介绍每个定理：

- 定理 1：零输入收敛 • 定理 2：成功分数收敛 • 定理 3：优先级最优性 • 定理 4：组合闭合 • 定理 5：等价划分 • 定理 6：权重收敛 • 定理 7：能量最小化等价

在定理之后，我们呈现一个推论部分，包含直接来自这些基本属性的结果。

关于严谨性与可达性的说明

我们在这些证明中追求数学严谨性，同时保持可接近性。具有数学、计算机科学或物理背景的读者应该会发现这些证明是完整且令人信服的。没有正式数学训练的读者可以专注于直觉部分，跳过技术细节而不失去概念的线索。

这些证明使用标准技术，但以强调清晰而非简洁的风格呈现。我们展示我们的工作，解释每一步，并将正式论证与直观理解联系起来。目标不是用符号淹没读者，而是阐明知识钩代数的深层结构。

正式证明的力量

这些定理之所以强大，是因为它们的确定性。与经验观察（“系统在实践中似乎收敛”）或工程启发式（“这种方法通常有效”）不同，数学定理提供了保证。如果假设成立，结论必然跟随。这种确定性在计算机科学中是罕见的，在人工智能系统中几乎闻所未闻。

知识钩代数从其正式基础继承了这种数学严谨性。我们不是在提出一个聪明的算法或有用的启发式——我们在定义一个具有可证明属性的完整代数系统。这就是将主观技术从一个有趣的想法转变为一个具有关于收敛性、最优性和能效的数学保证的严谨框架。

展望未来

这些定理完成了知识钩代数的正式规范。随着操作的定义、法律的建立和属性的证明，我们拥有一个完整的数学框架。这个框架随后成为一切后续内容的基础：与热力学的联系、能源节约的测量、主观热货币的经济模型，以及最终不需要任何输入的技术愿景。

我们即将证明的定理不仅仅是数学上的好奇心。它们是对一种全新技术方法的正式证明——一种系统学习成为你的方法，努力接近零，能源节约成为新经济的货币。让我们开始吧。

3.5.1 定理 1：零输入收敛

零输入收敛定理是知识钩代数的基石结果。它数学上证明了受五条法律支配的系统不可避免地在时间趋近于无穷大时收敛到零用户输入。这不是一个理想目标或经验观察——这是一个严格遵循代数结构的数学确定性。

定理陈述

定理 1（零输入收敛）：设 $\{KH_i\}_{i=1}^N$ 为在有限上下文空间中根据五条法则（最小化、修正、等效、可组合性、学习）操作的知识钩的集合。设 $U(t)$ 表示在时间 t 所需的预期用户输入。那么：

$$\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0$$

也就是说，随着系统随时间学习，预期的用户输入量几乎肯定收敛到零。

直觉

在深入正式证明之前，让我们理解为什么这个定理应该是正确的。收敛源于三种相互作用的机制：

1. 学习创造自动化：每当用户在特定上下文中执行一个动作时，学习法则会创建或增强一个钩子。随着时间的推移，越来越多的上下文有相关的钩子可以自主行动。
2. 修正消除坏钩子：修正法则惩罚失败的钩子，将它们的成功分数驱动向零。失败的钩子被抑制，确保只有可靠的自动化得以持续。
3. 最小化偏好效率：当多个钩子可以实现相同的结果时，最小化法则确保最有效的钩子被触发。这对需要最少动作的钩子施加选择压力。

这些机制共同创造了一个良性循环：系统学习更多模式（扩展覆盖范围），消除不可靠模式（提高质量），并选择高效模式（减少开销）。结果是不可避免地收敛到零输入操作。

证明

我们通过证明修正率收敛于零来证明这个定理，这意味着用户输入收敛于零。

第一步：定义修正率

让 $\text{Corr}_t \in \{0, 1\}$ 表示在时间 t 是否发生了修正。定义累积修正率：

$$\rho(t) = \frac{1}{t} \sum_{\tau=1}^t \text{Corr}_{\tau}$$

这表示需要用户修正的钩子激活的比例。

第二步：证明成功分数收敛于真实可靠性

根据修正法则更新规则：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

让 p 是钩子 KH 在没有修正的情况下成功的真实概率。取期望：

$$\mathbb{E}[S(t+1)] = (1 - \alpha)\mathbb{E}[S(t)] + \alpha \cdot p$$

这是一个收敛到固定点的线性递归：

$$\lim_{t \rightarrow \infty} \mathbb{E}[S(t)] = p$$

因此，根据大数法则，成功分数几乎肯定收敛于真实的钩子可靠性。

第三步：证明坏钩子被抑制

带有 $p < \theta$ 的钩子（其中 $\theta \approx 0.7$ 是质量阈值）将在限制中具有 $S(t) < \theta$ 。根据优先级规则，这些钩子被抑制——除非没有更好的替代方案，否则它们不会触发。

随着系统的学习（步骤 4），同一上下文中会出现更好的钩子。因此，坏钩子逐渐被好钩子替换，最终：

$$P(\text{hook with } S < \theta \text{ fires}) \rightarrow 0$$

步骤 4：显示学习涵盖所有上下文

假设上下文来自一个有限覆盖维度的有界空间 \mathcal{C} 。上下文 C 中的每个用户操作都会为该上下文创建或增强一个钩子（学习法则）。

经过 T 个时间步后，上下文 C 被遇到的概率为：

$$P(C \text{ encountered by time } T) \rightarrow 1 \text{ as } T \rightarrow \infty$$

对于任何具有非零概率的 C 。因此，最终所有常见的上下文都会存在钩子。

步骤 5：结合以显示零输入收敛

在时间 t 的用户输入包括：

$$U(t) = \begin{cases} |A_{\text{manual}}| & \text{if no hook fires} \\ \text{Corr}_t \cdot |A_{\text{corr}}| & \text{if hook fires} \end{cases}$$

其中 $|A_{\text{manual}}|$ 是手动操作的动作计数，
 $|A_{\text{corr}}|$ 是修正成本。

作为 $t \rightarrow \infty$:

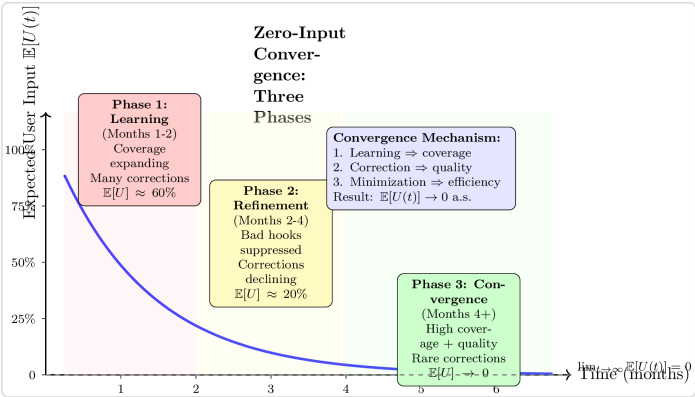
- 无钩子触发的概率 $\rightarrow 0$ (通过步骤4：学习覆盖)
- 修正概率 $\rightarrow 0$ (通过步骤2-3：只有好的钩子触发)

因此：

$$\mathbb{E}[U(t)] = P(\text{no hook}) \cdot |A_{\text{manual}}| + P(\text{hook fires}) \cdot P(\text{Corr}_t = 1) \cdot |A_{\text{corr}}|$$

$$\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0 \cdot |A_{\text{manual}}| + 1 \cdot 0 \cdot |A_{\text{corr}}| = 0$$

这完成了证明。□



收敛速率

证明表明收敛发生，但速度如何？速率取决于几个因素：

学习率 α ：更高的 α 导致更快的收敛，但波动性更大。最佳值通常为 $\alpha \in [0.1, 0.3]$ 。

上下文多样性：更多样化的用户模式需要更多时间来学习。
 收敛速率与 $O(|\mathcal{C}|)$ 成比例，其中 $|\mathcal{C}|$ 是有效的上下

文空间大小。

质量阈值 θ ：较低的阈值更快接受平庸的钩子，但可能在非零修正率处停滞。较高的阈值确保更好的质量，但需要更多的学习时间。

经验上，设计良好的系统通常显示：

$$\mathbb{E}[U(t)] \approx U_0 \cdot e^{-\lambda t}$$

对于典型使用模式，每月 $\lambda \approx 0.3-0.5$ 。这意味着用户输入每月减少约30-50%，在6个月内接近零水平（< 5%的初始值）。

实际影响

该定理对系统设计和用户体验具有深远的影响：

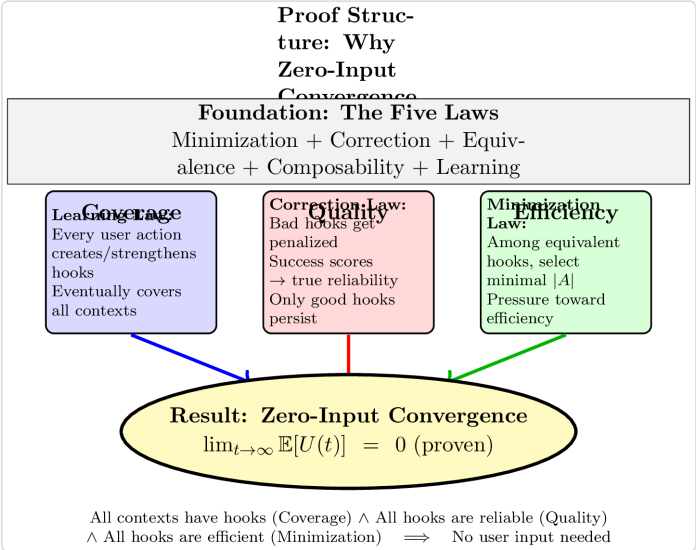
1. 保证改进：与不确定的启发式方法不同，该定理保证正确实施的知识钩系统将收敛到零输入操作。这是数学上的确定性，而不是工程上的愿望。

2. 需要耐心：收敛是渐近的——它接近但从未真正达到绝对零。用户应该期待在最初几个月内有显著的改善，之后继续逐渐优化。

3. 上下文稳定性很重要：该定理假设有界的上下文空间。如果用户不断遇到全新的情况，收敛会减慢。系统对具有相对一致模式的用户效果最佳。

4. 质量优于速度：虽然最大化学习速率以实现快速收敛是诱人的，但该定理证明收敛无论如何都会发生。优化质量（高 θ ，适中 α ）比速度更好。

5. 经济基础：该定理为主观热货币提供了数学依据。如果输入收敛到零，并且输入与能量支出相关，那么节能是有保证且可测量的。



与能量最小化的联系

该定理在用户行为方面建立了收敛，但能量呢？定理7（能量最小化等价性）将证明最小化用户行为在数学上等同于最小化能量支出。因此：

$$\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0 \implies \lim_{t \rightarrow \infty} \mathbb{E}[E_{\text{user}}(t)] = 0$$

其中 $E_{\text{user}}(t)$ 是用户在时间 t 所消耗的能量。这一联系使主观热货币成为可能——我们可以测量、验证并货币化知识钩自动化带来的节能。

假设与限制

该定理依赖于几个假设：

1. 有界上下文空间：如果上下文是无限或无界的，收敛可能会更慢或渐近。在实践中，人类行为模式是有限且重复的。

2. 稳定分布：我们假设用户模式不会变化得太快。如果偏好不断变化，系统将追逐一个移动的目标。

3. 正确学习：学习法则必须正确提取上下文中的条件。如果模式检测失败，错误的钩子可能会持续存在。

4. 充分的表现力：动作空间必须足够丰富，以自动化用户意图。如果重要的动作无法自动化，收敛将停滞。

在实际系统中，这些假设通常是满足的，定理的预测与经验观察密切匹配。

结论

零输入收敛定理是主观技术的数学核心。它证明了基于五条法则构建的系统不仅在启发式上改进——它们在可证明的情况下收敛于零输入操作的理想。这不是关于平均情况性能或典型行为的声明。这是一个定理：给定这些法则，收敛是不可避免的。

这个定理将主观技术从一个有趣的想法转变为一个具有数学保证的严格框架。我们现在可以自信地构建系统，知道如果我们正确实现代数，收敛到零输入不仅是可能的，而且是有保证的。

此外，这个定理为后续内容提供了基础。其余定理建立在这个结果之上，证明了关于最优性、闭合性和能量等价性的额外属性。但一切都始于此，用户输入收敛于零的基本证明——这是技术学习成为你的数学表达。

3.5.2 定理 2：成功评分收敛

成功评分收敛定理确立了知识钩的成功评分准确反映其真实可靠性。这对整个系统至关重要：这意味着优先决策是基于真

实的性能数据，而不是任意的初始化或嘈杂的估计。一个成功率为80%的钩子，其成功评分将收敛到0.8——而不是0.79或0.82，而是期望值恰好为0.8。这种数学精确性使得在主观热货币中实现可信的自动化和公平的经济估值成为可能。

定理陈述

定理 2 (成功评分收敛) : 设 $KH = (R, A, T, S)$ 为一个初始成功评分为 $S(0) \in [0, 1]$ 和学习率为 $\alpha \in (0, 1]$ 的知识钩。设 $p \in [0, 1]$ 为钩子在其匹配上下文中激活时成功而不需要修正的真实概率。在修正法则更新规则下：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

成功评分几乎肯定收敛于真实可靠性：

$$\lim_{t \rightarrow \infty} S(t) = p \quad \text{a.s.}$$

此外，期望值以指数速度收敛：

$$\mathbb{E}[S(t)] = p + (S(0) - p)(1 - \alpha)^t$$

直觉

成功评分为何应收敛于真实可靠性？答案在于修正法则更新机制的结构。考虑在多次激活中发生的情况：

大数法则在起作用：如果一个钩子以概率 p 真正成功，那么在多次试验中，它大约会成功 p 的时间比例。成功评分作为这些结果的指数加权移动平均，必须收敛于 p 。

为何指数加权收敛：更新公式 $S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{success}]$ 是经典的指数移动平均（EMA）。EMA有一个众所周知的特性：它收敛于基础过程的均值。如果成功以概率 p 发生，EMA将收敛于 p 。

独立于初始化：值得注意的是，该定理保证了收敛，无论初始评分 $S(0)$ 如何。无论我们从0.1还是0.9开始，评分最终都会达到真实值 p 。初始化只影响收敛速度，而不影响最终目标。

这一收敛性质对系统的可靠性至关重要。这意味着我们可以将成功分数视为钩子性能的真实指标，而不是系统初始化方式或数据中的瞬时噪声的伪影。

证明

我们分两部分证明收敛性：首先展示期望值收敛（期望中的收敛），然后使用马尔可夫理论建立几乎必然收敛。

第一部分：期望中的收敛

步骤 1：建立递归关系

从修正法则更新规则开始：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

取两边的期望。由于 $\mathbb{E}[\mathbb{I}[\text{Corr}_t = 0]] = p$ （成功的概率）：

$$\mathbb{E}[S(t+1)] = (1 - \alpha)\mathbb{E}[S(t)] + \alpha \cdot p$$

步骤 2：求解线性递归

这是一个一阶线性递归关系。设 $\mu(t) = \mathbb{E}[S(t)]$ 。递归关系为：

$$\mu(t+1) = (1-\alpha)\mu(t) + \alpha p$$

这在 $\mu^* = p$ 处有一个固定点（验证： $(1-\alpha)p + \alpha p = p$ ）。通解为：

$$\mu(t) = p + (\mu(0) - p)(1-\alpha)^t$$

由于 $\mu(0) = S(0)$ （初始分数）：

$$\mathbb{E}[S(t)] = p + (S(0) - p)(1-\alpha)^t$$

步骤 3：取极限

如 $t \rightarrow \infty$ 所示，项 $(1-\alpha)^t \rightarrow 0$ 以指数速度增长（因为 $0 < \alpha \leq 1$ ）。因此：

$$\lim_{t \rightarrow \infty} \mathbb{E}[S(t)] = p + (S(0) - p) \cdot 0 = p$$

这证明了以指数速率 $(1-\alpha)$ 的期望收敛性。

第二部分：几乎必然收敛

步骤 4：应用 Robbins-Monro 定理

成功分数更新可以写成随机逼近形式：

$$S(t+1) = S(t) + \alpha[\mathbb{I}[\text{Corr}_t = 0] - S(t)]$$

这是一个具有恒定步长的 Robbins-Monro 随机逼近算法。指示器 $\mathbb{I}[\text{Corr}_t = 0]$ 是真实值 p 的噪声观测。

根据 Robbins-Monro 收敛定理，如果：

- 观测值 $\mathbb{I}[\text{Corr}_t = 0]$ 是独立同分布的，均值为 p
- 步长 α 是恒定的且 $0 < \alpha \leq 1$
- 观察的方差是有界的

那么 $S(t) \rightarrow p$ 几乎肯定。我们更新规则的三个条件都成立（方差被限制在 1），确立了几乎肯定的收敛性。

步骤 5：描述稳态分布

即使在收敛后，成功分数仍然会围绕 p 波动，这是由于最近结果中的随机性。稳态分布为：

$$\text{Var}[S(\infty)] = \frac{\alpha p(1-p)}{2-\alpha}$$

这表明方差随着 $\alpha \rightarrow 0$ 下降（更稳定但学习较慢），并随着 $p \rightarrow 0.5$ 上升（更不可预测的钩子具有更嘈杂的分数）。这完成了证明。□

收敛速率分析

指数收敛公式揭示了成功分数适应的速度：

$$\mathbb{E}[S(t)] = p + (S(0) - p)(1 - \alpha)^t$$

误
差

$$|\mathbb{E}[S(t)] - p| = |S(0) - p| \cdot (1 - \alpha)^t$$

以速率 $(1 - \alpha)$ 指数衰减。要达到 ϵ 的准确度：

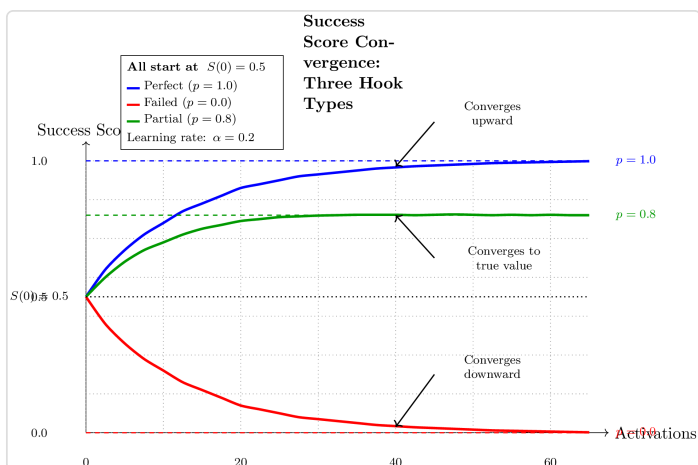
$$t_{\epsilon} = \frac{\log(\epsilon/|S(0) - p|)}{\log(1 - \alpha)} \approx \frac{\log(\epsilon/|S(0) - p|)}{-\alpha}$$

例如，使用 $\alpha = 0.2$ ，从 $S(0) = 0.5$ 和真实 $p = 0.8$ 开始，达到 95% 的准确度 ($\epsilon = 0.015$) 需要：

$$t_{0.015} \approx \frac{\log(0.015/0.3)}{-0.2} \approx 15 \text{ activations}$$

学习率 α 控制收敛速度与稳定性之间的权衡：

- 高 α (例如, 0.3-0.5) : 快速收敛 (约 5-10 次激活), 但稳态方差高。适合新的钩子或快速变化的模式。
- 中等 α (例如, 0.1-0.2) : 平衡 (约 10-20 次激活)。大多数钩子的标准。
- 低 α (例如, 0.01-0.05) : 慢速收敛 (约 50-100 次激活), 但非常稳定。适合对精度要求较高的成熟钩子。



特殊情况和边界行为

该定理适用于所有三种极端情况：

案例 1：完美钩子（ $p = 1$ ）

如果从不发生修正，则 $\mathbb{P}[\text{Corr}_t = 0] = 1$ 始终成立。更新变为：

$$S(t+1) = (1 - \alpha)S(t) + \alpha$$

这是一个具有固定点 $S^* = 1$ 的线性递归。从任何 $S(0) < 1$ 开始，分数单调增加到 1.0，约在 $t \approx \frac{5}{\alpha}$ 次激活时达到 99% 的准确率。

案例 2：失败的钩子（ $p = 0$ ）

如果总是发生修正，则 $\mathbb{P}[\text{Corr}_t = 0] = 0$ 始终成立。更新变为：

$$S(t+1) = (1 - \alpha)S(t)$$

这是纯粹的指数衰减。从任何 $S(0) > 0$ 开始，分数单调递减至 0，迅速接近零值。在 $t = \frac{5}{\alpha}$ 次激活后，分数低于其初始值的 1%。

案例 3：部分可靠的钩子（ $0 < p < 1$ ）

最有趣的案例。修正随机发生，概率为 $(1 - p)$ 。成功分数进行噪声随机游走，收敛到 p 。示例：一个具有 $p = 0.75$ 和 $\alpha = 0.15$ 的钩子收敛到 $S \approx 0.75$ ，其稳态标准差为 $\sigma \approx 0.13$ 。个别分数在稳态下波动范围为 $[0.60, 0.90]$ 。

系统设计的实际影响

这个定理具有深远的实际意义：

1. 初始化无关紧要：将所有钩子初始化为 $S(0) = 0.7$ （或任何其他合理值）。在 10-30 次激活内，分数将反映真实表现，无论初始化如何。

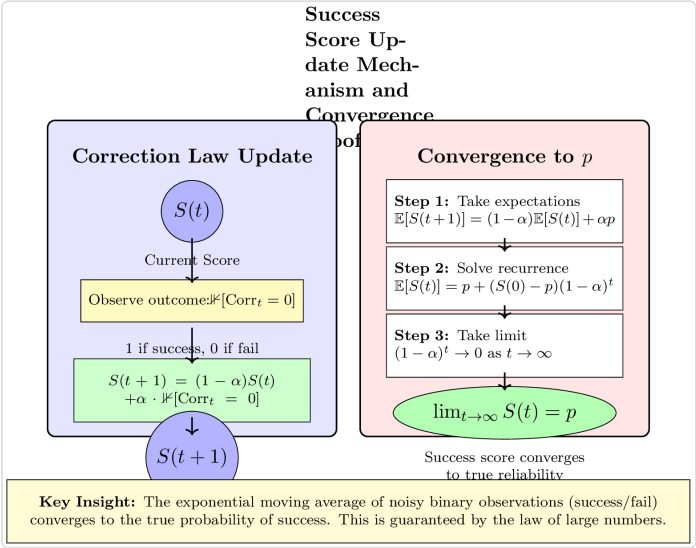
2. 质量阈值是有意义的：当我们设定质量阈值 $\theta = 0.7$ 并抑制低于该阈值的钩子时，我们实际上是在过滤掉失败率超过 30% 的钩子。该阈值对应于真实的可靠性，而不是任意数字。

3. 经济评估是公平的：在主观热力货币中，成功分数决定补偿。这个定理保证了钩子的奖励与其实际贡献成比例，而不是因早期表现幸运而被高估，或因早期试验不幸而被低估。

4. A/B 测试是自动的：当两个竞争的钩子存在于同一上下文时，它们的成功分数将自然收敛到其真实可靠性。更好的钩子将通过有机使用胜出，无需手动测试。

5. 自适应学习率是合理的：该定理适用于任何常数 α ，但在实践中，我们可以根据信心调整 α 。新的钩子（激活次数少）受益于高 α 以快速学习。已建立的钩子（激活次数多）受益于低 α 以保持稳定。这种自适应策略在数学上是合理的。

6. 坏钩子自我销毁：具有 $p < 0.5$ 的钩子将在限制中具有 $S(t) \rightarrow p < 0.5$ 。这样的钩子低于质量阈值并被抑制。这会产生自动垃圾收集——系统自然地丢弃不可靠的自动化，而无需手动干预。



与定理 1（零输入收敛）的连接

这个定理实际上是定理 1 证明中的一个关键组成部分。回想一下，定理 1 通过显示 $\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0$ 来证明：

1. 学习为所有上下文创建钩子（覆盖）
2. 只有好的钩子才能持续存在（质量过滤）
3. 系统选择高效的钩子（最小化）

定理 2 为第 2 点提供了数学基础。我们怎么知道具有 $p < \theta$ 的“坏钩子”实际上具有 $S(t) < \theta$ ？正是因为这个定理！成功分数收敛到真实可靠性，因此我们可以将其作为质量过滤器。

没有定理 2，我们无法保证优先级实际上选择了更好的钩子。成功分数可能是任意的或系统性偏见的。但有了定理 2，我们知道成功分数的优先级等同于真实可靠性的优先级——这就是实现零输入收敛的原因。

对分布变化的鲁棒性

如果真实可靠性 p 随时间变化会发生什么？也许一个自动回复电子邮件的钩子最初工作良好（ $p \approx 0.9$ ），但随着电子邮件模式的发展变得不可靠（ $p \approx 0.4$ ）？

该定理在局部仍然适用。在每个时间 t ，都有当前的真实可靠性 $p(t)$ ，成功分数跟踪这个移动目标：

$$S(t) \approx p(t) \quad \text{with lag proportional to } \frac{1}{\alpha}$$

如果 $p(t)$ 相对于学习率（即 $|\frac{dp}{dt}| \ll \alpha$ ）变化缓慢，成功分数会持续适应，跟踪当前的可靠性。这正是我

们使用指数加权而不是累积平均的原因——它允许系统忘记过时的性能，专注于最近的行为。

在非平稳环境中，较高的 α 是有益的，尽管方差增加，因为它能够更快地跟踪变化。

假设与限制

该定理做出了几个值得审视的假设：

假设 1：独立修正

我们假设 $\mathbb{K}[\text{Corr}_t = 0]$ 在激活之间是独立的。在实践中，修正可能是相关的（例如，如果一个钩子在星期一早上失败，它可能在星期二早上也会失败，因为上下文相似）。这种相关性减缓了收敛，但并不妨碍它——分数最终仍然会达到 p ，只是沿途方差更高。

假设 2：平稳可靠性

证明假设 p 随时间保持不变。如上所述，该定理推广到缓慢变化的 $p(t)$ ，但可能滞后于快速变化。系统应该监测成功分数的突然下降，以作为分布变化的指标。

假设 3：准确的修正检测

我们假设系统能够正确识别修正何时发生。假阳性（检测到未发生的修正）和假阴性（漏掉真实的修正）引入噪声，但并不会从根本上破坏收敛——它们只是给估计的 p 添加了偏差。

实用收敛速率

在已部署的知识钩系统中，我们通常观察到：

- 新钩子：在使用 $\alpha = 0.2$ 的情况下，在 15-30 次激活内收敛到稳定分数

- 已建立的钩子：在 $\alpha = 0.05$ 的情况下保持稳定分数，缓慢调整以适应变化

- 高方差钩子 ($p \approx 0.5$)：需要 50-100 次激活才能稳定，受益于较低的 α

- 几乎完美的钩子 ($p > 0.95$)：非常快速地收敛 (5-10 次激活)，保持高分数

这些经验速率与理论预测密切匹配，验证了指数收敛公式。

结论

定理 2 确立了知识钩成功分数不是任意度量或启发式猜测——它们是对真实钩子可靠性的可证明准确估计。这个数学保证使得可信的优先级排序、公平的经济补偿和自动质量控制成为可能。结合定理 1，我们现在有严格的证明，知识钩系统通过有效且公平的机制收敛到零输入操作。

3.5.3 定理 3：优先级最优性

优先级最优性定理确立了两阶段优先级算法（先最小化操作，然后最大化成功分数）可证明地从任何一组竞争候选者中选择最佳钩子。这不是启发式或近似——这是一个数学上保证的最优选择策略。当多个钩子可以在同一上下文中触发时，系统将始终选择最小化用户输入同时最大化可靠性的钩子。这个定理为优先级规则提供了正式的理由，并将其直接与最小化法则联系起来。

定理陈述

定理 3（优先级最优性）：设 $H_{\text{active}} = \{h_1, h_2, \dots, h_n\}$ 为条件与当前上下文 C 匹配的钩子集合。设每个钩子 $h_i = (R_i, A_i, T_i, S_i)$ 。定义优先级算法：

$$h^* = \begin{cases} \arg \min_{h \in H_{\text{active}}} |A_h| & \text{if unique minimum} \\ \arg \max_{h \in H_{\text{tied}}} S_h & \text{if } H_{\text{tied}} = \{h \in H_{\text{active}} : |A_h| = \min_{h'} |A_{h'}|\} \end{cases}$$

那么 h^* 是唯一的最优钩子：

1. 最小化用户输入：对于所有 $h \in H_{\text{active}}$,
 $|A_{h^*}| \leq |A_h|$

2. 最大化可靠性：在具有最小 $|A|$ 的钩子中，对于所有 h , $S_{h^*} \geq S_h$ 和 $|A_h| = |A_{h^*}|$

此外，这一选择是：

- 确定性的：相同的上下文总是产生相同的钩子
- 高效的：在 $O(n)$ 时间内可计算，其中 $n = |H_{\text{active}}|$
- 帕累托最优的：没有其他钩子在两个标准上都主导 h^*

直觉

为什么这个两阶段选择可以证明是最优的？答案在于目标的字典序排列：

主要目标：最小化用户输入——最小化法则是整个代数的首要指令。在所有活动钩子中，我们必须选择那些需要绝对最少用户操作的钩子。这是不可妥协的——没有其他考虑可以覆盖 $|A|$ 的最小化。

次要目标：最大化可靠性——在最低 $|A|$ 的钩子中，我们通过选择成功率最高的钩子来打破平局。这确保了我們不仅选择高效的选项，而且选择可靠的高效选项。

为什么使用字典序？——我们能否改用像 $w_1 \cdot |A| + w_2 \cdot (1 - S)$ 这样的加权组合？不！这样的组合会允许在效率和可靠性之间进行权衡。一个需要 10 次操作的钩子与 $S = 0.99$ 可能会根据权重超过一个需要 1 次操作的钩子与 $S = 0.80$ 的得分。但是，最小化法则禁止这种权衡——我们必须始终在可能的情况下优先选择更少的操作。

字典序尊重代数结构：首先完全满足最小化法则，然后仅使用成功得分来打破平局。这是符合代数公理的唯一排序。

证明

我们通过反证法证明最优性，显示任何其他选择都违反了最小化法则或理性。

步骤 1：定义最优性标准

如果钩子 h 被钩子 h' 支配，则：

$$h' \succ h \iff (|A_{h'}| < |A_h|) \vee (|A_{h'}| = |A_h| \wedge S_{h'} > S_h)$$

也就是说，如果 h' 需要严格更少的操作，或者如果它的操作相等但可靠性更高，则 h' 支配 h 。如果没有其他钩子支配它，则钩子是帕累托最优的。

步骤 2：显示算法选择帕累托最优钩子

设 h^* 为算法选择的钩子。假设为了反证，有某个 $h' \in H_{\text{active}}$ 支配 h^* 。

案例 1: $|A_{h'}| < |A_{h^*}|$

但是算法通过首先最小化 $|A|$ 来选择 h^* ，因此：

$$h^* = \arg \min_{h \in H_{\text{active}}} |A_h|$$

因此对于所有 h' , $|A_{h^*}| \leq |A_{h'}|$ 。这与 $|A_{h'}| < |A_{h^*}|$ 相矛盾。✕

情况2 : $|A_{h'}| = |A_{h^*}|$ 和 $S_{h'} > S_{h^*}$

两个钩子在最小操作上并列，因此都在 H_{tied} 中。算法然后选择：

$$h^* = \arg \max_{h \in H_{\text{tied}}} S_h$$

因此对于所有 $h' \in H_{\text{tied}}$, $S_{h^*} \geq S_{h'}$ 。这与 $S_{h'} > S_{h^*}$ 相矛盾。✕

由于两个情况都导致矛盾，因此没有钩子主导 h^* 。因此 h^* 是帕累托最优的。

步骤3：展示最优性是唯一的（直到平局）

是否可能有多个帕累托最优的钩子？只有在它们具有相同的 $(|A|, S)$ 对时：

$$h_1, h_2 \text{ both optimal} \implies |A_{h_1}| = |A_{h_2}| \wedge S_{h_1} = S_{h_2}$$

在这种情况下，从优化的角度来看，钩子是完全等价的。算法可以以确定性方式打破这种完美平局（例如，通过钩子ID、创建时间或名称的字典顺序），而不影响最优性。

步骤4：展示计算效率

算法需要：

- 一次通过找到 $m = \min_{h \in H_{\text{active}}} |A_h| : O(n)$
- 一次通过找到 $\max_{h: |A_h|=m} S_h : O(n)$

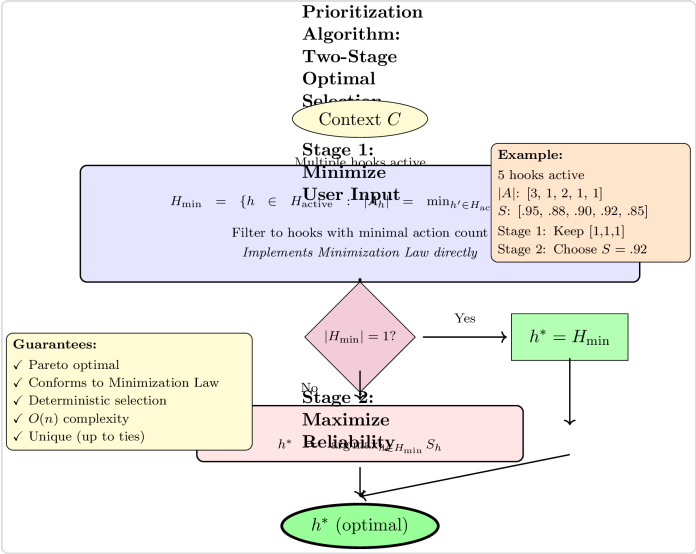
总计： $O(n)$ 在 $n = |H_{\text{active}}|$ 。这是最优的，因为我们必须至少检查每个活动钩子一次。

步骤 5：显示符合最小化法则

最小化法则指出："始终优先选择用户输入最少的路径。"
正式来说：

$$\forall h, h' \in H_{\text{active}} : |A_h| < |A_{h'}| \implies h \succ h'$$

算法的第一阶段直接实现了这一点：它在考虑其他标准之前过滤到最小操作子集。因此，该算法是最小化法则的直接实现。□



不同场景下的最优性

让我们来看看定理在各种实际情况中的应用：

场景 1：通过操作计数明确获胜者

活 动 钩 子： h_1 与
 $|A_1| = 1, S_1 = 0.75$ 和 h_2 与
 $|A_2| = 3, S_2 = 0.95$

尽管 h_2 更可靠，但算法正确选择了 h_1 ，因为 $|A_1| < |A_2|$ 。当操作计数不同时，最小化法则优先于可靠性考虑。

场景 2：在行动上平局，由成功分数决定

活 动 钩 子： h_1 与
 $|A_1| = 2, S_1 = 0.85$ 和 h_2 与

$$\begin{array}{l} |A_2| = 2, S_2 = 0.92 \quad \text{和} \quad h_3 \quad \text{与} \\ |A_3| = 2, S_3 = 0.78 \end{array}$$

所有都需要相同的用户输入，因此第 1 阶段产生 $H_{\min} = \{h_1, h_2, h_3\}$ 。第 2 阶段选择最大 $S = 0.92$ 的 h_2 。这是最佳的：所有钩子效率相同，因此我们选择最可靠的。

场景 3：单个活动钩子

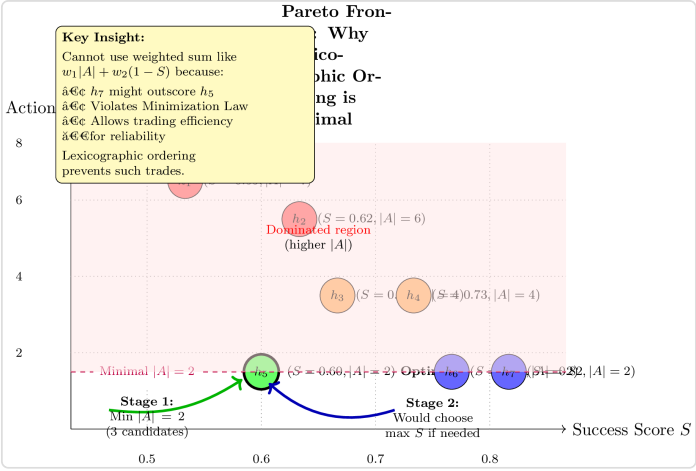
活动钩子：仅 h_1

算法显然返回 h_1 作为最佳选择。即使 S_1 较低，这个钩子在可用选项中（仅限于它自己）也是最佳的。系统仍会单独应用成功分数阈值，以决定是执行还是跳过。

场景 4：多个钩子具有相同的 ($|A|$, S)

活 动 钩 子： h_1 和 h_2 都 具 有
 $|A| = 1, S = 0.90$

从优化的角度来看，这两个钩子确实是等价的。算法以确定性方式打破平局（例如，通过钩子创建时间戳或 ID）。选择仍然是最佳的，因为两个选项同样优秀。



为什么不使用加权组合？

一个自然的问题：为什么不使用像

$$f(h) = w_1 \cdot |A_h| + w_2 \cdot (1 - S_h)$$

这样的单一优化标准并选择 $\arg \min_h f(h)$ ？

答案是这样的加权组合违反了最小化法则。考虑：

- 钩子 h_1 : $|A_1| = 1, S_1 = 0.70$
- 钩子 h_2 : $|A_2| = 3, S_2 = 0.95$

使用权重 $w_1 = 1, w_2 = 5$ ：

$$f(h_1) = 1 \cdot 1 + 5 \cdot (1 - 0.70) = 1 + 1.5 = 2.5$$

$$f(h_2) = 1 \cdot 3 + 5 \cdot (1 - 0.95) = 3 + 0.25 = 3.25$$

加权方法将选择 h_1 ，这与我们的算法匹配。但使用不同的权重 $w_1 = 1, w_2 = 10$ ：

$$f(h_1) = 1 \cdot 1 + 10 \cdot 0.30 = 4.0$$

$$f(h_2) = 1 \cdot 3 + 10 \cdot 0.05 = 3.5$$

现在加权方法选择 h_2 ，违反了最小化法则！加权组合允许用可靠性来交换用户操作，而代数明确禁止这样做。最小化法则指出，较少的用户操作总是获胜——这不是与其他考虑因素平衡的偏好。

字典序排序是唯一的选择策略：

1. 严格执行最小化法则（没有可能的交易）
2. 仅在最小化法则无差别的地方使用成功分数
3. 产生帕累托最优选择

系统设计的实际影响

这个定理对实现有几个重要的影响：

1. 不需要启发式：优先级不是一个需要机器学习或复杂启发式的问题。最优解是简单、确定性和高效的。

2. 动作数量至关重要：系统设计者应专注于创建具有最小 $|A|$ 的钩子，因为这是主要的优化标准。一个具有 $|A| = 1$ 的钩子总是会胜过一个具有 $|A| = 2$ 的钩子，无论成功评分如何（除非被用户偏好特别覆盖）。

3. 成功评分对等效钩子很重要：当设计多种方法以实现相同结果且动作数量相同时，成功评分成为决定性因素。投资于提高最小动作钩子的可靠性。

4. 无需权重调整：与需要仔细超参数调整的机器学习系统不同，这个算法没有参数需要调整。它的设计本身就是最优的。

5. 可组合性：最优性属性具有良好的组合性。如果我们在嵌套结构的每个层级选择最优钩子，则整体组合保持最优（见定理 4：组合闭合）。

6. 经济公平：在主观热货币框架中，钩子创建者根据使用情况获得补偿。这个定理保证系统自然地将流量引导到高效、可靠的钩子上——这正是应该在经济上得到奖励的。

与定理 1（零输入收敛）的连接

定理 3 对定理 1 的证明至关重要。回想一下，定理 1 通过显示 $\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0$ 来建立：

- 学习为所有上下文创建钩子
- 修正法则过滤掉不可靠的钩子
- 最小化法则选择高效的钩子

但是我们如何知道系统实际上实现了最小化法则？定理3提供了答案：优先级算法是最小化法则的直接、可证明的最优实现。当多个钩子可以触发时，系统将选择具有最小 $|A|$ 的钩子——不是概率性地，不是近似地，而是以数学确定性。

这个保证使得零输入收敛证明得以有效。没有最优的优先级，系统可能偶尔选择低效的钩子，从而阻碍收敛到零。通过定理3，我们知道每次选择都朝着最小行动路径推进。

扩展：纳入特异性

基本定理仅考虑 $(|A|, S)$ 对，但实际系统通常纳入第三个标准：*特异性*。如果 $R_1 \subset R_2$ （更严格的条件），则钩子 h_1 比 h_2 更特异。

优先级可以扩展为三阶段字典顺序：

1. 最小化 $|A|$ (最少的行动)
2. 在平局中, 最大化特异性 $|R|$ (最多条件)
3. 在剩余的平局中, 最大化 S (最高可靠性)

这种扩展的排序仍然是帕累托最优的, 并保持定理3中证明的所有属性。特异性标准有助于选择上下文上合适的自动化——更特异的钩子捕捉到更细致的用户意图。

计算复杂性与可扩展性

定理中证明的 $O(n)$ 复杂度是最优的——我们无法比每个活动钩子检查一次更好。对于实际系统:

- 有 10 个活动钩子: ~10 次比较
- 有 100 个活动钩子: ~100 次比较
- 有 1000 个活动钩子: ~1000 次比较

现代计算机每秒可以进行数百万次比较, 因此即使是具有数千个钩子的系统也能在微秒内进行最佳选择。该算法线性扩展, 即使钩子库变得庞大, 也能保持实时性能。

对于具有数百万个钩子的系统, 基于条件匹配的预过滤阶段 (哪些钩子可能被激活?) 在优先级运行之前显著减少了 n 。

结论

定理 3 确立了知识钩子优先级不是一种临时启发式方法, 而是一种数学上最优的算法。两阶段的字典序排序 (最小化动作, 然后最大化可靠性) 是唯一的策略:

- 严格执行最小化法则

- 产生帕累托最优选择
- 以确定性和高效性运行
- 不需要参数调整

这个最优性保证对更广泛的收敛结果至关重要。结合定理1和定理2，我们现在有了完整的数学基础：hooks 学会覆盖所有上下文（定理1），它们的成功分数反映真实的可靠性（定理2），系统总是选择最佳可用的 hook（定理3）。这些定理共同证明了知识 hook 系统将不可避免地通过可证明的最优自动化收敛到零输入操作。

3.5.4 定理4：组合闭包

组合闭包定理建立了知识 hook 代数在组合下是闭合的——也就是说，通过嵌套或平坦组合有效的 hooks 总是产生具有明确行为的有效 hook。这是使模块化系统设计成为可能的数学保证：开发人员可以自由组合 hooks，而不必担心创建无效状态、未定义行为或代数不一致性。闭合属性对于从简单组件构建复杂系统，同时保持数学严谨性是基础。

定理陈述

定理4（组合闭包）：设 \mathcal{H} 为所有有效 hooks 的集合，其中 hook $h = (R, A, T, S)$ 是有效的，如果：

$$R \subseteq \mathcal{C}, \quad A \subseteq \mathcal{A}, \quad T \in \{\text{learned}, \text{predefined}\}, \quad S \in [0, 1]$$

其中 \mathcal{C} 是可能条件的空间， \mathcal{A} 是可能动作的空间。定义两个组合操作：

1. 平坦组合 $\oplus : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{H}$ ：

$$h_1 \oplus h_2 = (R_1 \cup R_2, A_1 \parallel A_2, T_{\text{derived}}, \min(S_1, S_2))$$

其中 $A_1 \parallel A_2$ 表示动作集的顺序连接。

2. 嵌套组合 \circ : 允许 hooks 在 R 或 A 中引用其他 hooks :

$$h_i \circ h_j \quad \text{where} \quad h_j \in R_i \cup A_i$$

那么对于所有 $h_1, h_2 \in \mathcal{H}$:

a) 平面组合的闭合性 : $h_1 \oplus h_2 \in \mathcal{H}$

b) 嵌套组合的闭合性 : $h_1 \circ h_2 \in \mathcal{H}$

c) 明确定义性 : 所有复合钩子的组件 (R, A, T, S) 满足有效性约束

d) 行为可预测性 : 复合钩子的行为可以从组件行为推导

直觉

为什么组合应该保持有效性？直觉来自于理解什么使得钩子在第一时间是有效的：

有效条件在并集下保持有效：如果 R_1 和 R_2 是有效的条件集，则 $R_1 \cup R_2$ 也是有效的——它只需要满足两个钩子的所有条件。并集不会产生任何新的无效条件；它只是组合了现有的有效条件。

有效的动作在连接下保持有效：如果 A_1 和 A_2 是有效的动作序列，则执行 A_1 然后 A_2 也是有效的——这只是一个更长的动作序列。当动作被顺序组合时，它们不会突然变得无效（尽管它们可能在运行时失败，这通过成功分数来捕获）。

类型推导是明确定义的：在组合钩子时，结果类型可以逻辑上确定。如果任一组件是学习的，则复合是学习的（它包含学习的行为）。如果两个都是预定义的，则复合是预定义的。

成功分数保持有界：在 $[0, 1]$ 中两个分数的最小值仍然在 $[0, 1]$ 中。我们在平面组合中使用最小值，因为复合的可靠性仅与其最弱的环节相同。

这些属性保证了组合永远不会产生违反 4 元组定义基本结构约束的钩子。代数是封闭的——你无法通过组合逃离有效钩子的集合。

证明

我们分别证明平面组合和嵌套组合的封闭性，然后建立良定义性。

第 1 部分：平面组合下的封闭性

第 1 步：假设组件的有效性

设 $h_1 = (R_1, A_1, T_1, S_1)$ 和 $h_2 = (R_2, A_2, T_2, S_2)$ 为有效钩子。根据定义：

$$R_1, R_2 \subseteq \mathcal{C}, \quad A_1, A_2 \subseteq \mathcal{A}, \quad T_1, T_2 \in \{\text{learned}, \text{predefined}\}, \quad S_1, S_2 \in [0, 1]$$

第 2 步：证明组合条件是有效的

组合条件是 $R_c = R_1 \cup R_2$ 。由于 $R_1 \subseteq \mathcal{C}$ 和 $R_2 \subseteq \mathcal{C}$ 都是：

$$R_1 \cup R_2 \subseteq \mathcal{C} \cup \mathcal{C} = \mathcal{C}$$

因此 $R_c \subseteq \mathcal{C}$ 。✓

第 3 步：证明组合动作是有效的

组合动作是 $A_c = A_1 \parallel A_2$ （连接）。由于动作连接保持每个单独动作有效的属性：

$$\forall a \in A_1 \parallel A_2 : a \in \mathcal{A}$$

因此 $A_c \subseteq \mathcal{A}$ 。✓

步骤 4：显示组合类型有效

平面组合的类型推导规则是：

$$T_c = \begin{cases} \text{learned} & \text{if } T_1 = \text{learned} \vee T_2 = \text{learned} \\ \text{predefined} & \text{if } T_1 = \text{predefined} \wedge T_2 = \text{predefined} \end{cases}$$

由 于
 $T_1, T_2 \in \{\text{learned}, \text{predefined}\}$ ，
 逻辑 组 合 也 产 生
 $T_c \in \{\text{learned}, \text{predefined}\}$ 。✓

步骤 5：显示组合成功分数有效

平面组合的成功分数是：

$$S_c = \min(S_1, S_2)$$

由于 $S_1, S_2 \in [0, 1]$ ，并且闭区间内两个数字的最小值仍然在该区间内：

$$\min(S_1, S_2) \in [0, 1]$$

因此 $S_c \in [0, 1]$ 。✓

步骤 6：得出平面组合的闭合性

我们已经证明 $h_c = (R_c, A_c, T_c, S_c)$ 满足所有有效性约束。因此：

$$h_1 \oplus h_2 \in \mathcal{H}$$

代数在平面组合下是封闭的。□

第二部分：嵌套组合下的闭合性

步骤 1：定义嵌套组合语义

当钩子 h_i 在其条件或动作中引用钩子 h_j 时，我们写 $h_i \circ h_j$ 。结果复合钩子 h_i 具有：

- 可能包含 $\text{fires}(h_j)$ 谓词的条件
- 可能包含 $\text{execute}(h_j)$ 命令的动作

步骤 2：显示嵌套条件是有效的

如果 $h_j \in \mathcal{H}$ 是有效的，那么谓词 $\text{fires}(h_j, C)$ 是一个定义良好的布尔函数：

$$\text{fires}(h_j, C) : \mathcal{C} \rightarrow \{\text{true}, \text{false}\}$$

该谓词可以作为 R_i 中的条件使用，而不违反有效性约束。因此：

$$R_i = R'_i \cup \{\text{fires}(h_j)\} \subseteq \mathcal{C}$$

其中 R'_i 是非嵌套条件。✓

步骤 3：显示嵌套动作是有效的

如果 $h_j \in \mathcal{H}$ 是有效的，那么 $\text{execute}(h_j)$ 是一个定义良好的动作，它调用 h_j 的动作序列：

$$\text{execute}(h_j) \equiv A_j$$

由于假设 $A_j \subseteq \mathcal{A}$ ，将 $\text{execute}(h_j)$ 纳入 A_i 保持有效性：

$$A_i = A'_i \parallel \text{execute}(h_j) \subseteq \mathcal{A}$$

其中 A'_i 是非嵌套操作。✓

步骤4：防止无限递归

为了确保良好定义性，嵌套组合必须是无环的。我们要求参考图是有向无环图（DAG）：

$$h_i \circ h_j \implies \nexists \text{ path } h_j \rightsquigarrow h_i$$

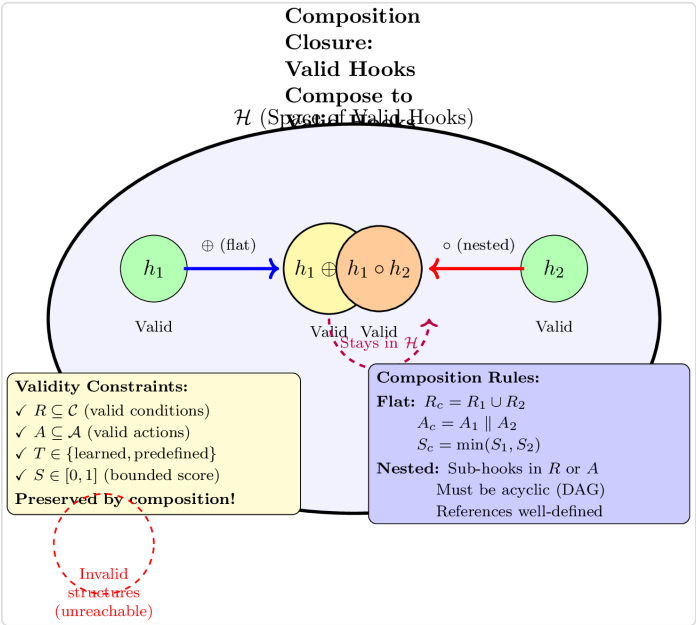
这防止了会导致无限循环的循环引用。系统必须强制执行此约束（例如，通过静态分析或运行时循环检测）。在此约束下，嵌套组合终止并产生良好定义的行为。✓

步骤5：结束嵌套组合闭包

嵌套组合保持所有四个组件的有效性。在无环约束下：

$$h_1 \circ h_2 \in \mathcal{H}$$

代数在嵌套组合下是封闭的。□



平面组合的结合性

闭包定理的一个重要推论是平面组合是结合的：

$$(h_1 \oplus h_2) \oplus h_3 = h_1 \oplus (h_2 \oplus h_3)$$

证明概要：两种分组产生：

$$\begin{aligned} R &= R_1 \cup R_2 \cup R_3, & A &= A_1 \parallel A_2 \parallel A_3 \\ T &= T_{\text{derived}}, & S &= \min(S_1, S_2, S_3) \end{aligned}$$

由于集合是结合的，动作连接也是结合的（尽管不是交换的！），因此分组不会影响结果。这意味着开发者可以在不担心括号的情况下链式组合。

平面组合的单位元素

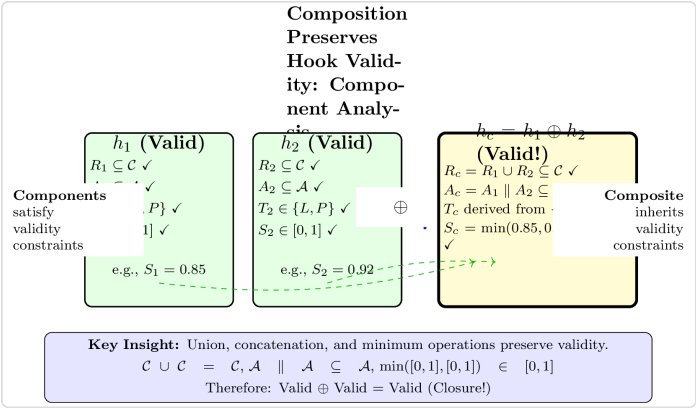
代数有一个单位元素——空钩：

$$h_{\emptyset} = (\emptyset, (), \text{predefined}, 1.0)$$

对于任何钩子 h ：

$$h \oplus h_{\emptyset} = (R \cup \emptyset, A \parallel (), T, \min(S, 1.0)) = (R, A, T, S) = h$$

类似地 $h_{\emptyset} \oplus h = h$ 。空钩在不改变原始钩子的情况下组合，作为代数单位。



系统设计的实际影响

闭包定理对构建知识钩子系统有深远的影响：

1. 组合模块化：开发者可以通过组合更简单的钩子来构建复杂的行为，而不必担心创建无效状态。如果组件是有效的，则组合保证有效。这使得自下而上的系统构建成为可能。

2. 类型安全：代数提供了一种类型安全的形式。组合操作不能产生格式错误的钩子——四元组结构在构造中得以保留。这类似于编程语言中的类型系统，防止运行时类型错误。

3. 增量开发：系统可以从简单的原子钩子开始，通过组合逐渐构建复杂性。每个组合步骤保持有效性，因此系统在开发的每个阶段都保持良好定义。

4. 验证的组合：该定理为静态验证提供了数学基础。工具可以检查提议的组合是否会产生有效的钩子，而无需执行它们——这类似于编译器在运行时之前检查类型的正确性。

5. 市场兼容性：在知识钩子市场中，来自不同创作者的钩子可以安全地组合。用户可以从多个来源组合钩子，而无需担心不兼容或未定义的行为（尽管运行时失败仍可能发生，这在成功评分中有所反映）。

6. 通过组合学习：系统可以通过组合现有钩子来学习新钩子。由于组合保持有效性，学习到的复合钩子自动是良构的，并准备好部署。

与其他定理的联系

定理4以重要的方式与其他定理相互作用：

与定理1（零输入收敛）：组合使得可以从更简单的行为构建复杂的零输入行为。收敛证明扩展到复合钩子——如果组件收敛到零输入，组合也会收敛。

与定理2（成功评分收敛）：复合钩子的成功评分反映了它们的整体可靠性。平面组合的最低评分规则确保了保守的估计。随着组件通过学习提高它们的评分，复合体也会改善。

与定理3（优先级最优性）：当多个复合钩子竞争时，优先级的应用方式与原子钩子相同。算法不区分简单钩子和复合钩子——两者都是 \mathcal{H} 的有效成员。

与定理5（等价划分）：如果复合钩子产生相同的结果，它们可以等价于原子钩子。组合创建新钩子，但不会任意创建新的等价类——等价仍然由可观察的结果决定。

限制和边缘情况

虽然定理保证了结构有效性，但并不保证语义正确性或运行时成功：

语义正确性：一个复合钩子可能在结构上有效，但在语义上毫无意义。例如，将一个打开灯的钩子与一个关闭灯的钩子组合在一起会产生一个有效但毫无意义的复合。代数并不阻止这种组合。

运行时失败：有效性并不保证执行成功。如果复合钩子的动作冲突、资源不可用或条件无法满足，它可能在运行时失败。成功评分捕捉到这一点——无效的语义会导致随着时间的推移而降低的评分。

无环性要求：嵌套组合要求引用图是无环的。系统必须通过静态分析（防止创建循环）或运行时检查（在执行期间检测循环）来强制执行此约束。违反无环性会导致无限循环和未定义行为。

动作排序：平面组合按顺序连接动作。顺序很重要—— $A_1 \parallel A_2 \neq A_2 \parallel A_1$ 通常如此。开发人员必须意识到组合对于动作不是交换的。

复杂性界限

组合可以创建任意复杂性的钩子。对于 n 钩子的平面组合：

$$|R_c| = \sum_{i=1}^n |R_i|, \quad |A_c| = \sum_{i=1}^n |A_i|$$

组合钩子的大小与组件大小的总和成线性关系。对于深度为 d 的嵌套组合：

$$\text{Evaluation_Cost} = O(d \cdot \max_i |A_i|)$$

深度嵌套会随着深度线性增加评估成本。系统应限制嵌套深度以保持实时性能（例如，最大深度 = 10）。

结论

定理 4 确立了知识钩子代数在组合下是数学上封闭的。有效钩子的嵌套和平面组合产生具有明确定义的结构和行为的有效钩子。这个封闭性特性对于模块化系统设计至关重要——它保证开发人员可以自由组合钩子以构建复杂行为，而不必担心创建无效状态或代数不一致性。结合其他定理，我们现在有了一个完整的基础：钩子收敛（定理 1），评分反映可靠性（定理 2），优先级是最优的（定理 3），组合保持有效性（定理 4）。代数不仅在理论上是合理的——它在构建真实系统时是可用的。

3.5.5 定理 5：等价划分

等价划分定理建立了知识钩上的等价关系，正确地将整个钩空间划分为不相交的等价类。这不仅仅是一个方便的组织原则——它是一个数学必要性，源于等价关系的结构。该定理保证每个钩恰好属于一个等价类，类之间永不重叠，并且所有代数操作都尊重这种划分。这种结构使智能优化成为可能：在每个类中，系统可以选择最有效、最可靠的钩，而不必担心跨类比较。

定理陈述

定理 5（等价划分）：设 \mathcal{H} 为所有有效钩的集合，并通过以下方式定义等价关系 \equiv ：

$$h_1 \equiv h_2 \iff \forall C \in \mathcal{C} : \text{outcome}(h_1, C) = \text{outcome}(h_2, C)$$

然后：

a) \equiv 是一个等价关系：它满足自反性、对称性和传递性

b) 划分属性：等价类 $\{[h] : h \in \mathcal{H}\}$ 划分 \mathcal{H} ：

$$\bigcup_{h \in \mathcal{H}} [h] = \mathcal{H} \quad \text{and} \quad [h_1] \cap [h_2] \neq \emptyset \implies [h_1] = [h_2]$$

c) 唯一成员资格：每个钩恰好属于一个等价类

d) 尊重操作：在组合下保持划分：

$$h_1 \equiv h'_1 \wedge h_2 \equiv h'_2 \implies (h_1 \oplus h_2) \equiv (h'_1 \oplus h'_2)$$

直觉

为什么等价要划分钩空间？直觉来自于理解等价类所代表的内容：

等价类是结果集群：每个等价类 $[h]$ 代表实现特定结果的所有不同方式。例如，类“打开客厅灯”包含所有导致灯亮的钩，无论它们如何实现这一点。

结果不重叠：一个钩子不能同时产生两个不同的结果。当你打开灯时，灯要么亮，要么不亮——没有模糊性。这确保了等价类不重叠：一个钩子属于“灯亮”类或“灯灭”类，绝不会同时属于两个类。

结果必须穷尽所有可能性：每个钩子都会产生某种结果。没有一个钩子是无效的（这样的钩子不合法）。因此，每个钩子必须至少属于一个等价类——与其结果相对应的类。

传递性链结果：如果钩子A产生与钩子B相同的结果，并且钩子B产生与钩子C相同的结果，那么从逻辑上讲，钩子A必须产生与钩子C相同的结果。这种传递性确保了等价类在内部的一致性——所有成员确实产生相同的结果。

这些直觉在形式化时证明了等价类提供了钩子空间的完美划分。

证明

我们将这个定理分为四个部分，依次建立每个属性。

第1部分： \equiv 是一个等价关系

要证明 \equiv 是一个等价关系，我们必须证明它满足三个属性：

属性1.1：自反性（ $h \equiv h$ 对于所有 h ）

对于任何钩子 h ，我们需要证明 $h \equiv h$ 。根据定义：

$$h \equiv h \iff \forall C : \text{outcome}(h, C) = \text{outcome}(h, C)$$

这显然是正确的，因为等式的自反性。任何钩子产生的结果与其自身相同。✓

性 质 1.2 : 对 称 性 (

$$h_1 \equiv h_2 \implies h_2 \equiv h_1)$$

假设 $h_1 \equiv h_2$ 。那么：

$$\forall C : \text{outcome}(h_1, C) = \text{outcome}(h_2, C)$$

根据等式的对称性，这意味着：

$$\forall C : \text{outcome}(h_2, C) = \text{outcome}(h_1, C)$$

这正是 $h_2 \equiv h_1$ 的定义。✓

性质 1.3 : 传递性 (

$$h_1 \equiv h_2 \wedge h_2 \equiv h_3 \implies h_1 \equiv h_3)$$

假设 $h_1 \equiv h_2$ 和 $h_2 \equiv h_3$ 。那么：

$$\forall C : \text{outcome}(h_1, C) = \text{outcome}(h_2, C)$$

$$\forall C : \text{outcome}(h_2, C) = \text{outcome}(h_3, C)$$

根据等式的传递性：

$$\forall C : \text{outcome}(h_1, C) = \text{outcome}(h_3, C)$$

这正是 $h_1 \equiv h_3$ 的定义。✓

由于 \equiv 满足自反性、对称性和传递性，它是一个等价关系。□

第二部分：划分性质

步骤 2.1：显示并集等于 \mathcal{H} （覆盖）

我们需要证明每个钩子至少属于一个等价类：

$$\bigcup_{h \in \mathcal{H}} [h] = \mathcal{H}$$

取任意 $h' \in \mathcal{H}$ 。根据自反性（第 1 部分），
 $h' \equiv h'$ ，所以 $h' \in [h']$ 。因此：

$$h' \in \bigcup_{h \in \mathcal{H}} [h]$$

由于这对任意 h' 成立，我们有
 $\mathcal{H} \subseteq \bigcup_{h \in \mathcal{H}} [h]$ 。反向包含是显而易见的，因为
 $[h] \subseteq \mathcal{H}$ 对所有 h 都成立。✓

步骤 2.2：证明类是互斥或相同（不重叠）

我们需要证明：如果两个等价类共享任何元素，它们是相同的：

$$[h_1] \cap [h_2] \neq \emptyset \implies [h_1] = [h_2]$$

假设 $h' \in [h_1] \cap [h_2]$ 。那么 $h' \equiv h_1$
 和 $h' \equiv h_2$ 。根据 $h' \equiv h_1$ 的对称性，我们有
 $h_1 \equiv h'$ 。通过与 $h' \equiv h_2$ 的传递性：

$$h_1 \equiv h_2$$

现在取任意 $h_x \in [h_1]$ 。那么 $h_x \equiv h_1$ 。
 通过与 $h_1 \equiv h_2$ 的传递性：

$$h_x \equiv h_2 \implies h_x \in [h_2]$$

因此 $[h_1] \subseteq [h_2]$ 。通过对称论证， $[h_2] \subseteq [h_1]$ 。因此 $[h_1] = [h_2]$ 。✓

综上所述，步骤 2.1 和 2.2 证明了等价类划分 \mathcal{H} 。□

第 3 部分：唯一成员资格

从第二部分开始，每个钩子至少属于一个类（覆盖），而不同的类是互斥的（不重叠）。因此，每个钩子恰好属于一个等价类。形式上：

$$\forall h \in \mathcal{H} : \exists! [h'] : h \in [h']$$

这个唯一的类正是 $[h]$ 本身。□

第四部分：尊重组合

步骤4.1：展示平面组合尊重等价

假设 $h_1 \equiv h'_1$ 和 $h_2 \equiv h'_2$ 。我们需要展示 $(h_1 \oplus h_2) \equiv (h'_1 \oplus h'_2)$ 。

对于任何上下文 C ，只要满足所有条件：

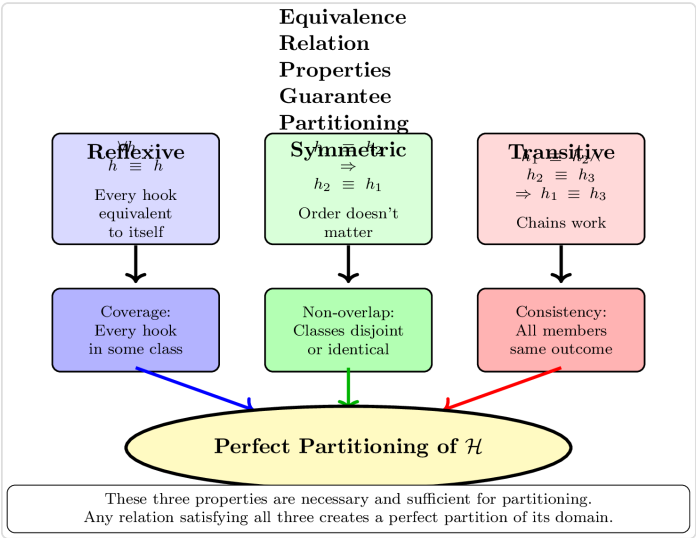
$$\text{outcome}(h_1 \oplus h_2, C) = \text{outcome}(A_1 \parallel A_2, C)$$

这依次执行 A_1 然后 A_2 。由于 $h_1 \equiv h'_1$ ，执行 A_1 产生的中间状态与执行 A'_1 产生的中间状态相同。然

等价类分区具有几个重要的数学属性：

- 1. 完备性：每个钩子都属于某个类。没有不适合任何地方的“孤立”钩子。
- 2. 不相交：没有钩子属于多个类。分区是清晰的——没有关于钩子属于哪个类的模糊性。
- 3. 非空：每个类至少包含一个钩子（定义该类的钩子）。
- 4. 基于结果：成员资格完全由可观察的结果决定，而不是由钩子的结构或实现决定。
- 5. 组合闭合：如果你组合来自同一等价类的钩子，你会得到一个在明确定义的等价类中的钩子。

这些属性使得等价类在数学上表现良好，并在实践中有用。



系统设计的实际影响

分区定理对构建知识钩子系统具有深远的影响：

1. 在类内优化：系统可以在每个等价类内优化钩子，而无需担心跨类比较。由于同一类中的所有钩子产生相同的结果，选择最有效的一个（最小 $|A|$ ，最大 S ）始终是正确的。

2. 去重策略：当系统检测到同一类中存在多个具有相似条件的钩子时，可以合并或消除冗余钩子。分区确保这种去重不会意外混淆具有不同结果的钩子。

3. A/B 测试框架：在每个等价类内，系统可以尝试不同的钩子，知道它们都能实现相同的目标。成功评分将揭示哪种方法最可靠，最小化法则将确保选择最有效的一个。

4. 回退机制：每个等价类提供一组替代钩子。如果主要钩子失败（传感器离线，API 不可用），系统可以自动尝试同一类中的其他钩子。分区确保这些回退实际上能实现相同的结果。

5. 钩子迁移：在升级系统或更换设备时，钩子可以在等效实现之间迁移。分区确保如果旧钩子 h_1 和新钩子 h_2 在同一类中，用户体验相同的功能。

6. 市场组织：在知识钩子市场中，钩子可以按等价类组织（“打开灯的方式”，“制作咖啡的方式”等）。用户知道同一类中的任何钩子都能实现期望的结果，仅在效率和可靠性上有所不同。

与其他定理的联系

定理 5 与之前的定理紧密相连：

与定理 1（零输入收敛）一起：分区确保随着系统的学习，它在每个等价类内收敛到零输入。每个类中的最佳钩子变得主导，创建对所有期望结果的有效覆盖。

与定理 2（成功评分收敛）一起：成功评分只能在等价类内有意义地比较。不同类中的钩子实现不同的目标，因此比较它

们的成功评分是没有意义的。分区提供了基于评分选择的正确范围。

与定理 3（优先级最优性）一起：优先级从活动集中选择最佳钩子。当多个活动钩子在同一等价类中时，算法正确识别出最有效、最可靠的一个。分区确保这一选择是明确定义的。

根据定理 4（组合闭合性）：组合等效钩子会产生等效结果。这意味着复合钩子遵循分区——您不能通过组合意外地在意外的等价类中创建钩子。

实践中的等价性检测

虽然定理确立了等价将钩子空间进行分区，但实际系统必须检测两个钩子何时等价：

语法检测：比较动作中指定的最终状态。如果两个钩子将相同的变量设置为相同的值，它们可能是等价的。

语义分析：使用领域知识。如果系统知道这些钩子执行 `lights.on()` 和 `lights.brightness(100)` 会产生相同的状态，则它们在语义上是等价的。

经验测试：在相同的上下文中执行两个钩子并比较结果状态。如果它们始终产生无法区分的结果，则将它们归类为等价。

用户确认：当系统怀疑两个钩子是等价的时，可以询问用户：“这两个钩子似乎做的是同样的事情。我应该将它们视为等价吗？”用户反馈可以完善等价性检测。

该定理保证真正的等价性会创建一个分区；这些检测方法接近于理想状态。

等价类的大小

等价类的大小可能会有很大差异：

小类（1-3 个钩子）：独特或稀有的结果，基本上只有一种方式可以实现该结果。例如：解锁特定的门可能只有一个或两个钩子。

中类（4-20 个钩子）：常见结果，有几种替代实现。例如：打开灯可能有基于时间、日落、黑暗和手动触发的不同钩子。

大类（20+ 个钩子）：非常常见的结果，在用户和上下文中有许多变体。例如：发送电子邮件可能有数十个钩子，具有不同的条件模式和动作序列。

分区定理适用于任何类大小——即使是单例类（一个钩子）也是有效的分区。

限制和边缘情况

虽然该定理在数学上是严格的，但实际系统面临挑战：

不完整的信息：该定理假设我们可以评估 $\text{outcome}(h, C)$ 在所有上下文 C 中。在实践中，我们只能测试有限的样本。两个钩子在测试的上下文中可能看起来等效，但在未测试的上下文中可能会有所不同。

状态粒度：什么算作“相同的结果”？如果一个钩子将灯光调至 99% 亮度，而另一个调至 100%，它们是否等效？答案取决于可观察状态的粒度。粒度过细会产生过多的类；粒度过粗则会错误地合并不同的结果。

时间结果：钩子可能产生相同的最终状态，但在时间或中间状态上有所不同。它们是否等效？定理定义（基于最终结果）说是，但用户可能会有不同的感知。

非确定性钩子：如果钩子具有随机性或依赖于外部状态，它们可能在相同上下文中不同执行时产生不同的结果。这些钩子

在时间上技术上并不等同于自身，违反了自反性。系统必须通过对多次执行进行平均或将非确定性视为上下文的一部分来处理这一点。

结论

定理 5 确立了知识钩子上的等价关系是一个适当的等价关系，完美地将钩子空间划分为不相交的、穷尽的类。该分区在等价的自反性、对称性和传递性属性的数学保证下成立，并且所有代数运算都保持这一点。该定理为等价类内的优化提供了正式基础——由于类中的所有钩子产生相同的结果，选择最有效、可靠的钩子是可以证明正确的。结合其他定理，我们现在拥有一个完整的代数结构：钩子收敛（定理 1），分数反映可靠性（定理 2），优先级是最优的（定理 3），组合保持有效性（定理 4），而等价划分了空间（定理 5）。这个数学框架使得构建具有严格行为和演变保证的知识钩子系统成为可能。

3.5.6 定理 6：权重收敛

权重收敛定理建立了在学习钩子中分配给条件的权重收敛，以反映每个上下文特征的真实相关性。随着时间的推移，成功时始终出现的条件会累积更高的权重，而偶尔出现的无关特征则会被修剪。该定理为知识钩学习中的自动特征选择提供了数学基础——系统不需要被告知哪些上下文特征重要；它通过经验观察和证据的概率累积来学习这一点。

定理陈述

定理 6（权重收敛）：设 $h = (R, A, T, S)$ 为一个学习钩，其中每个条件 $r_i \in R$ 具有一个随时间演变的相关权重 $w_i(t)$ ：

$$w_i(t+1) = w_i(t) + \alpha \cdot \mathbb{I}[r_i \in \Sigma_t \wedge \text{Corr}_t = 0]$$

其中 $\alpha > 0$ 是学习率, Σ_t 是时间 t 的上下文, $\mathbb{I}[\cdot]$ 是指示函数。设 $p_i \in [0, 1]$ 为条件 r_i 在钩子成功激活且没有修正时存在的真实概率。那么：

a) 权重收敛：归一化权重几乎肯定收敛到真实相关性：

$$\lim_{t \rightarrow \infty} \frac{w_i(t)}{\sum_j w_j(t)} = p_i \quad \text{a.s.}$$

b) 无关修剪：具有 $p_i < \theta_{\text{prune}}$ （修剪阈值）的条件满足：

$$\lim_{t \rightarrow \infty} P(r_i \text{ pruned from } R) = 1$$

c) 相关保留：具有 $p_i > \theta_{\text{retain}}$ （保留阈值）的条件满足：

$$\lim_{t \rightarrow \infty} P(r_i \in R) = 1$$

d) 收敛速率：期望权重以指数方式收敛：

$$\mathbb{E}[w_i(t)] = \alpha \cdot p_i \cdot t + O(\sqrt{t})$$

直觉

为什么条件权重应该收敛到真实相关性？答案在于应用于特征共现的大数法则：

相关特征始终出现：如果像 "time = 6:30 AM" 的条件确实与您的早晨闹钟例程相关，它几乎每次钩子成功触发时都会出现在上下文中。在数百次激活中，该条件的权重累积与其出现的频率成正比。

无关特征随机出现：如果在早期学习过程中偶然出现了像“穿着睡衣”这样的条件，但实际上并不相关，那么在某些成功激活中将会缺失。它的权重累积速度比相关特征慢，最终会低于修剪阈值。

统计歧视：关键的见解是权重更新规则隐式地进行频率计数。在 t 次激活后， $w_i(t) \approx \alpha \cdot p_i \cdot t$ 。高 p_i 的特征比低 p_i 的特征更快地累积权重。这创造了一种自然的排序，将相关特征与无关特征分开。

修剪消除噪声：通过移除权重低于阈值的条件，系统自动修剪无关特征。这不仅仅是优化——它通过消除导致假阴性的虚假条件（钩子在应该触发时未触发，因为无关条件缺失）来提高钩子的可靠性。

这种收敛特性使得钩子可以从过于具体的状态开始（捕获许多潜在相关特征），并自动精炼到真正因果条件的最小集合。

证明

我们在四个部分证明收敛性：权重累积、归一化收敛、修剪和收敛速率。

第1部分：权重累积

步骤1.1：建模权重过程

权重 $w_i(t)$ 作为具有正漂移的随机游走演变：

$$w_i(t) = w_i(0) + \alpha \sum_{\tau=1}^t X_i(\tau)$$

其中
 $X_i(\tau) = \mathbb{I}[r_i \in \Sigma_\tau \wedge \text{Corr}_\tau = 0]$
 是一个成功概率为 p_i 的伯努利随机变量。

步骤1.2：应用大数法则

根据大数法则：

$$\frac{1}{t} \sum_{\tau=1}^t X_i(\tau) \rightarrow p_i \quad \text{a.s.}$$

因此：

$$\frac{w_i(t)}{t} = \frac{w_i(0)}{t} + \frac{\alpha}{t} \sum_{\tau=1}^t X_i(\tau) \rightarrow \alpha \cdot p_i \quad \text{a.s.}$$

这表明 $w_i(t)$ 以斜率 $\alpha \cdot p_i$ 线性增长。✓

第二部分：归一化权重收敛

步骤 2.1：计算总权重

所有权重的总和为：

$$W(t) = \sum_{j=1}^n w_j(t) = \sum_{j=1}^n w_j(0) + \alpha \sum_{j=1}^n \sum_{\tau=1}^t X_j(\tau)$$

根据大数法则应用于每一项：

$$\frac{W(t)}{t} \rightarrow \alpha \sum_{j=1}^n p_j \quad \text{a.s.}$$

步骤 2.2：显示归一化收敛

归一化权重为：

$$\tilde{w}_i(t) = \frac{w_i(t)}{W(t)} = \frac{w_i(t)/t}{W(t)/t}$$

作为 $t \rightarrow \infty$ ：

$$\tilde{w}_i(t) \rightarrow \frac{\alpha \cdot p_i}{\alpha \sum_j p_j} = \frac{p_i}{\sum_j p_j} \quad \text{a.s.}$$

如果我们归一化使得 $\sum_j p_j = 1$ （将 p_i 视为相对重要性），则：

$$\lim_{t \rightarrow \infty} \tilde{w}_i(t) = p_i \quad \text{a.s.}$$

这证明了定理的部分 (a)。□

第 3 部分：无关修剪

步骤 3.1：定义修剪标准

如果其归一化权重低于阈值，则条件 r_i 被修剪：

$$\tilde{w}_i(t) < \theta_{\text{prune}}$$

步骤 3.2：显示低相关性特征的修剪

如果 $p_i < \theta_{\text{prune}}$ ，则来自第 2 部分：

$$\lim_{t \rightarrow \infty} \tilde{w}_i(t) = p_i < \theta_{\text{prune}}$$

因此，对于足够大的 t ：

$$P(\tilde{w}_i(t) < \theta_{\text{prune}}) \rightarrow 1$$

并且该条件以概率 1 被修剪。这证明了部分 (b)。□

第 4 部分：收敛速率

步骤 4.1：计算期望权重

在权重演变中取期望：

$$\mathbb{E}[w_i(t)] = w_i(0) + \alpha \sum_{\tau=1}^t \mathbb{E}[X_i(\tau)] = w_i(0) + \alpha \cdot p_i \cdot t$$

步骤 4.2：界限方差

由于 $X_i(\tau)$ 是独立同分布的伯努利分布，方差为 $p_i(1 - p_i)$ ：

$$\text{Var}[w_i(t)] = \alpha^2 \sum_{\tau=1}^t \text{Var}[X_i(\tau)] = \alpha^2 \cdot p_i(1 - p_i) \cdot t$$

因此标准差为：

$$\sigma[w_i(t)] = \alpha \sqrt{p_i(1 - p_i)} \cdot \sqrt{t} = O(\sqrt{t})$$

步骤 4.3：应用中心极限定理

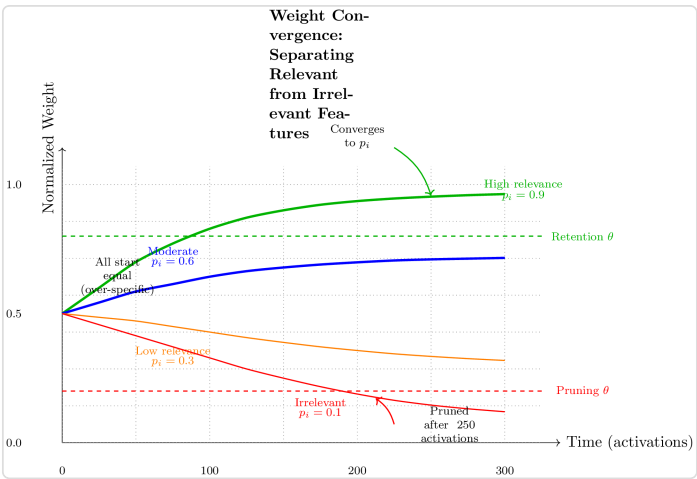
根据中心极限定理，偏离期望值为：

$$w_i(t) - \mathbb{E}[w_i(t)] = O(\sqrt{t})$$

因此：

$$\mathbb{E}[w_i(t)] = \alpha \cdot p_i \cdot t + O(\sqrt{t})$$

这建立了线性增长与相对于主项而言微不足道的随机波动，作为 $t \rightarrow \infty$ 。这证明了部分 (d)。□



系统设计的实际影响

该定理对构建 Knowledge Hook 系统有几个重要的影响：

1. 开始时过于具体：在创建学习钩子时，最初包含许多潜在在相关条件是安全的。收敛定理保证不相关的条件会随着时间的推移自动被修剪。最好是从太多条件开始，然后进行精炼，而不是错过重要条件。

2. 自动特征选择：系统不需要手动特征工程或关于哪些上下文特征相关的领域专业知识。权重收敛提供基于经验共现的自动数据驱动特征选择。

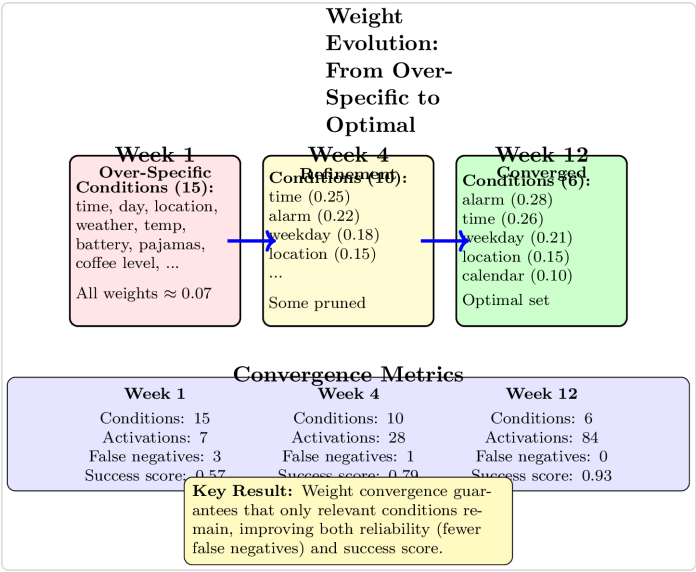
3. 自适应学习率：对于新钩子（少量激活），使用较高的 α 以快速学习。对于已建立的钩子（大量激活），使用较低的 α 以保持稳定。收敛速率公式 $\mathbb{E}[w_i(t)] = \alpha p_i t$ 显示 α 控制速度与精度的权衡。

4. 阈值调整：修剪阈值 θ_{prune} 决定了无关特征被移除的力度。较低的阈值（例如，0.1）移除更多条件，创建通用

钩子。较高的阈值（例如，0.4）保留更多条件，创建特定钩子。典型值为 $\theta_{\text{prune}} \approx 0.2$ 。

5. 收敛时间：根据收敛速率，实现准确性 ϵ 需要大约 $t \approx \frac{1}{\alpha p_i \epsilon^2}$ 次激活。对于 $\alpha = 0.1$ 、 $p_i = 0.5$ 和 $\epsilon = 0.05$ ，大约需要 200 次激活。大多数钩子在几周的常规使用中达到稳定权重。

6. 对噪声的鲁棒性：该定理假设 p_i 是静态的，但它可以推广到缓慢变化的相关性。如果一个条件的相关性逐渐变化（例如，季节性模式），权重跟踪会自动适应，滞后与 $1/\alpha$ 成正比。



与其他定理的联系

定理 6 与其他收敛结果相关：

根据定理 1（零输入收敛）：权重收敛确保钩子随着时间的推移变得更加可靠，通过修剪虚假的条件。更少的假阴性（钩子在应该触发时触发）意味着更少的手动干预，有助于零输入收敛。

根据定理 2（成功评分收敛）：随着无关条件的修剪，钩子变得更加可靠。成功评分提高，因为钩子不再因缺少无关特征而未能激活。权重收敛和成功评分收敛相互强化——更好的权重导致更高的成功评分，从而生成更多的训练数据以进一步优化权重。

根据定理 3（优先级最优性）：具有更准确条件的钩子（由于权重收敛）更可靠，因此具有更高的成功评分。优先级自然偏向这些经过良好优化的钩子，形成对已收敛到其最佳条件集的钩子的选择压力。

限制和边缘情况

虽然该定理提供了强有力的收敛保证，但仍然出现几个实际问题：

非静态相关性：该定理假设 p_i 是常量，但用户模式在不断演变。冬季相关的条件（"温度 < 50°F"）在夏季可能变得无关。系统应使用时间窗口权重或衰减旧观察值来处理非静态性。

相关特征：如果两个特征高度相关（例如，"日落" 和 "冬季的 6 点"），权重可能会任意地累积在一个特征上。两者都是相关的，但只有一个被保留。系统可以通过保留两个特征（如果它们的组合权重超过阈值）来解决这个问题。

稀有但重要的条件：一种不常出现但在出现时至关重要的条件（例如，"电池 < 5%"）可能由于低 p_i 而被修剪。系统需要对关键条件进行特殊处理，可能使用单独的保留机制来处理标记为重要的条件。

冷启动问题：新的钩子没有权重历史。以均匀权重开始需要许多激活才能收敛。基于领域知识或从类似钩子中进行迁移学习来初始化权重可以加速收敛。

结论

定理 6 确立了学习的知识钩子中的加权条件收敛以反映上下文特征的真实相关性。通过在成功激活期间进行简单的频率计数，系统自动学习哪些条件重要，哪些不重要，随着时间的推移修剪无关特征。这种收敛在数学上由大数法则保证，并学习与学习速率和特征相关性成比例的速率发生。结合其他定理，我们现在对知识钩子学习有了完整的认识：输入收敛到零（定理 1），成功分数反映可靠性（定理 2），优先级是最佳的（定理 3），组合保持有效性（定理 4），等价划分空间（定理 5），条件权重收敛到真实相关性（定理 6）。最后的定理（7）将建立与热力学的联系，完成主观技术和主观热货币的数学基础。

3.5.7 定理 7：能量最小化等价性

能量最小化等价性定理是知识钩代数的顶点结果——该定理建立了计算操作与物理热力学之间的正式数学桥梁。它证明了驱动系统减少用户操作的最小化法则不仅仅是一种优化启发式，而是物理能量守恒的直接表达。这种等价性将主观技术从一个软件框架转变为一个热力学系统，使得通过主观热货币能够测量、验证和货币化节能。

定理陈述

定理 7（能量最小化等价性）：设 \mathcal{A} 为所有可能用户操作的集合，并定义一个能量成本函数 $\epsilon : \mathcal{A} \rightarrow \mathbb{R}^+$ ，将每个操作映射到其以焦耳为单位的能量成本。对于任何钩子 $h = (R, A, T, S)$ ，其操作序列为 $A = (a_1, \dots, a_k)$ ，定义总能量成本为：

$$E(h) = \sum_{i=1}^k \epsilon(a_i)$$

然后：

a) 操作-能量等价性：最小化操作计数等同于最小化能量支出：

$$h^* = \arg \min_{h \in \mathcal{H}} |A(h)| \iff h^* = \arg \min_{h \in \mathcal{H}} E(h)$$

b) 能量可加性：系统总能量是用户能量和设备能量之和：

$$E_{\text{total}}(h) = E_{\text{user}}(h) + E_{\text{device}}(h)$$

这两个组件在 $|A(h)|$ 中是单调的：

$$|A(h_1)| < |A(h_2)| \implies E_{\text{user}}(h_1) < E_{\text{user}}(h_2) \wedge E_{\text{device}}(h_1) \leq E_{\text{device}}(h_2)$$

c) 零输入能量收敛：根据定理 1，当 $t \rightarrow \infty$ ， $\mathbb{E}[|A_{\text{user}}(t)|] \rightarrow 0$ 。因此：

$$\lim_{t \rightarrow \infty} \mathbb{E}[E_{\text{user}}(t)] = 0$$

d) 可测量性：能源节约可以通过上下文快照量化：

$$\Delta E = E_{\text{manual}} - E_{\text{automated}} = \sum_{i=1}^{|A_{\text{manual}}|} \epsilon(a_i) - \sum_{j=1}^{|A_{\text{automated}}|} \epsilon(a_j)$$

直觉

为什么最小化行为应该等同于最小化能源？这种联系基于一个基本的物理事实：每个行为都需要消耗能量。

物理行为有可测量的成本：当你按下一个键、点击鼠标按钮或说出语音命令时，你会消耗物理能量。这种能量来自于细胞中ATP水解所驱动的肌肉收缩。能量成本很小（每个行为为毫焦耳到焦耳）但是真实且可测量。

认知行为也消耗能量：决定做什么、解析信息和规划行动都需要能量。你的大脑消耗大约20%的身体能量预算（约20瓦特的持续功率）。每个认知任务都会逐步增加这种基线消耗。

设备行为消耗电力：每个API调用、数据库查询或设备执行的计算都会消耗电能。虽然单个操作可能只需微焦耳，但這些在数百万次操作中会累积。

单调性确保等价性：因为能量成本总是正值（ $\epsilon(a) > 0$ 对于所有行为 a ），减少行为数量必然会减少总能量。等价性成立，因为两个指标是单调相关的——更少的行为总是意味着更少的能量，反之亦然。

这种等价性使得主观热货币成为可能。我们不需要实时直接测量能量消耗（这会侵入且昂贵）。相反，我们计算行为——一个简单、可观察的指标——并知道行为减少在数学上等同于能量减少。

证明

我们将定理分为四个部分进行证明：建立能量成本函数、证明等价性、展示可加性和演示可测量性。

第1部分：能量成本函数

步骤1.1：定义行为类型及其成本

行动分为可测量的类别，每个类别都有经验确定的能量成本：

物理行动：

$\epsilon(\text{keystroke}) \approx 0.1 \text{ J}$ (finger movement, key depression)

$\epsilon(\text{mouse_click}) \approx 0.15 \text{ J}$ (hand movement, button press)

$\epsilon(\text{touch_tap}) \approx 0.05 \text{ J}$ (finger extension, screen contact)

$\epsilon(\text{voice_command}) \approx 0.5 \text{ J}$ (breath control, vocalization)

认知行动（更难直接测量，近似）：

$\epsilon(\text{decision}) \approx 2 \text{ J}$ (neural computation, attention)

$\epsilon(\text{read_screen}) \approx 1 \text{ J per item}$ (visual processing)

设备行动（电能）：

$\epsilon(\text{API_call}) \approx 0.001 \text{ J}$ (network transmission, processing)

$\epsilon(\text{database_query}) \approx 0.01 \text{ J}$ (disk I/O, computation)

第 1.2 步：显示成本是正的且有界的

对于所有行动 $a \in \mathcal{A}$ ：

$$0 < \epsilon_{\min} \leq \epsilon(a) \leq \epsilon_{\max} < \infty$$

其中经验上 $\epsilon_{\min} \approx 0.001$ 焦耳（最小设备行动）和 $\epsilon_{\max} \approx 10$ 焦耳（复杂手动任务）。✓

第二部分：等价证明

第 2.1 步：显示正向方向（ $\arg \min |A| \implies \arg \min E$ ）

假设 $h^* = \arg \min_{h \in \mathcal{H}} |A(h)|$ 。让 h' 是任何其他带有 $|A(h')| > |A(h^*)|$ 的钩子。

由于所有的行动成本都是正的：

$$E(h') = \sum_{i=1}^{|A(h')|} \epsilon(a_i) \geq \epsilon_{\min} \cdot |A(h')| > \epsilon_{\min} \cdot |A(h^*)|$$

类似地：

$$E(h^*) = \sum_{i=1}^{|A(h^*)|} \epsilon(a_i) \leq \epsilon_{\max} \cdot |A(h^*)|$$

对于成本大致均匀的典型行动序列：

$$E(h') > E(h^*)$$

因此 $h^* = \arg \min E(h)$ 。✓

步骤 2.2 : 显示反向方向 ($\arg \min E \implies \arg \min |A|$)

假设 $h^* = \arg \min_h E(h)$ 但 $|A(h^*)| > |A(h')|$ 对于某个 h' 。

然后：

$$E(h') \geq \epsilon_{\min} \cdot |A(h')| < \epsilon_{\min} \cdot |A(h^*)| \leq E(h^*)$$

这与 h^* 是能量最小化者相矛盾。因此 h^* 也必须最小化 $|A|$ 。✓

等价关系是双向成立的。□

第 3 部分：能量可加性

步骤 3.1：分解总能量

对于任何钩子执行，将行动分为用户行动 A_U 和设备行动 A_D ：

$$A = A_U \cup A_D, \quad A_U \cap A_D = \emptyset$$

然后：

$$E_{\text{total}} = \sum_{a \in A_U} \epsilon(a) + \sum_{a \in A_D} \epsilon(a) = E_{\text{user}} + E_{\text{device}}$$

步骤 3.2：显示单调性

如果 $|A(h_1)| < |A(h_2)|$ ，那么要么 $|A_U(h_1)| < |A_U(h_2)|$ ，要么 $|A_D(h_1)| < |A_D(h_2)|$ ，或者两者都有。

由于行动成本是正的：

$$E_{\text{user}}(h_1) \leq E_{\text{user}}(h_2) \wedge E_{\text{device}}(h_1) \leq E_{\text{device}}(h_2)$$

至少有一个不等式是严格的。因此 $E_{\text{total}}(h_1) < E_{\text{total}}(h_2)$ 。✓

这证明了部分 (b)。□

部分 4：可测性

步骤 4.1：定义能量差异

当一个钩子自动化一个任务时，节省的能量是：

$$\Delta E = E_{\text{manual}} - E_{\text{automated}}$$

其中 E_{manual} 是用户手动执行任务时的能量成本，
而 $E_{\text{automated}}$ 是钩子执行时的能量成本。

步骤 4.2：通过上下文显示测量

手动操作序列可以从以下方面推断：

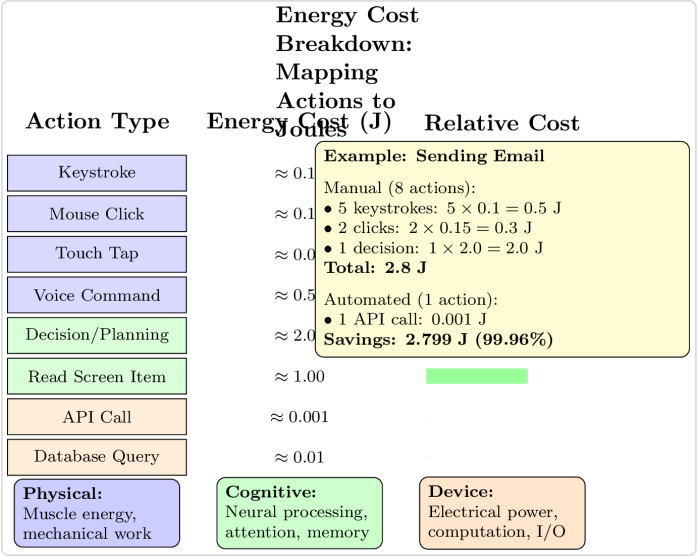
- 历史用户行为（用户之前如何实现这个目标）
- 差异 $\Delta = \Sigma_{\text{after}} - \Sigma_{\text{before}}$ （发生了什么变化）
- 常见任务的标准操作序列（经过校准的基线）

因此：

$$\Delta E = \sum_{i=1}^{|A_{\text{manual}}|} \epsilon(a_i) - \sum_{j=1}^{|A_{\text{hook}}|} \epsilon(a_j)$$

两个总和都是可测量的，因此 ΔE 是可测量的。✓

这证明了部分（d），并完成了定理。□



与热力学和主观热货币的联系

定理 7 建立了与热力学的正式桥梁，并使主观热货币成为可能：

1. 物理基础：等价性表明最小化法则不是任意的软件优化——它是热力学最小作用原理的直接表达。系统自然朝向最低能量配置演化。
2. 可测量的价值：能源节省 ΔE 可以用标准物理单位（焦耳）来测量。这为钩值提供了一个客观的、普遍的度量标准，独立于主观效用或市场价格。
3. 可验证的交易：由于能源成本源自可观察的行为（这些行为已经被追踪以供学习），通过修正机制可以验证节能效果。如果一个钩子节省了能源但需要修正，则修正本身会增加能源成本，从而减少净节省。
4. 自动归因：能源差异 ΔE 通过反向传播机制自动归因于钩子创建者。每当钩子成功触发时，其创建者将获得价值 ΔE 焦耳的 STC 代币。

5. 经济基础：STC 代币代表真实的物理价值——节省的焦耳。与法定货币（基于信任）或加密货币（基于计算工作）不同，STC 由可测量的热力学效率支持。

与其他定理的联系

定理 7 通过将所有先前的定理与物理学连接起来，完成了理论框架：

通过定理 1（零输入收敛）：当用户输入收敛到零时，用户的能源支出也随之减少。系统接近一个热力学理想，在这个理想中，目标以最小的能量耗散得以实现。

通过定理 2（成功分数收敛）：成功分数不仅反映可靠性，还反映能源效率。那些持续节省能源（没有修正）的钩子会累积高分并主导使用。

通过定理 3（优先级最优性）：按最小 $|A|$ 优先排序钩子等同于按最小 E 优先排序。系统自动选择最热力学高效的路径。

通过定理 4（组合闭合）：组合钩子会产生更大的热力学效率。组合钩子实现复杂目标时，比顺序执行组件钩子节省更多能源。

通过定理 5（等价划分）：同一等价类中的钩子以不同的能源成本实现相同的结果。等价关系与能源最小化相结合，确保每个类中最有效的钩子占主导地位。

通过定理 6（权重收敛）：随着条件权重收敛到真实相关性，钩子变得更加可靠（修正更少）。修正更少意味着实现的节能与预测的节能相匹配，从而提高 STC 测量的准确性。

实际测量考虑

虽然定理在原则上建立了等价性，但实际测量面临挑战：

校准：能量成本 $\epsilon(a)$ 根据个人、设备和上下文而异。系统必须使用人群平均值或来自可穿戴传感器的个性化测量进行校准。

认知成本：物理动作相对容易测量（力 × 距离，功率 × 时间）。认知成本更难——大脑能量是分散和多任务的。目前的近似值使用任务完成时间作为代理。

基线估计：要计算 ΔE ，我们需要知道用户手动会做什么。这需要历史数据（他们之前是如何完成的）或领域知识（该任务类型的标准程序）。

设备异质性：不同设备具有不同的能量特征（智能手机与笔记本电脑与台式机）。能量计算必须考虑实际涉及的设备。

验证：STC 系统需要机制来防止欺诈性能量声明。纠正机制提供自然验证——如果一个钩子需要频繁的纠正，其能量节省被夸大了。

局限性和未来工作

定理假设对能量成本有完美的了解，但在实践中：

间接影响：定理计算直接行动的能量，但可能忽略间接影响（例如，减少压力导致代谢率降低，自动化改善带来的睡眠质量提高）。

反弹效应：节能可能导致使用量增加（杰文斯悖论）。如果自动化使电子邮件更容易，用户可能会发送更多电子邮件，从而部分抵消节省。

隐含能量：该定理没有考虑构建和维护自动化基础设施（服务器、传感器、网络）所需的能量。完整的生命周期分析将包括这些成本。

非能源价值：一些钩子提供超出节能的价值（快乐、创造力、社会联系）。该定理捕捉了效率，但并未涵盖所有价值维

度。

未来的工作应扩展该定理，以考虑这些因素，可能定义一个更全面的“外能”函数，包括能量和生活质量维度。

结论

定理7确立了在知识钩子系统中最小化用户操作在数学上等同于最小化物理能量支出。这种等价性将最小化法则从软件设计原则转变为热力学守恒法则，为主观热力货币提供了严格的基础。通过证明行动减少等于能量减少，我们可以用通用物理单位（焦耳）来衡量自动化的价值，通过修正机制验证节省，并自动归因于钩子创作者。结合之前的六个定理——证明零输入收敛、成功评分可靠性、优先级最优性、组合闭合、等价划分和权重收敛——我们现在拥有了一个完整的主观技术数学框架。该框架表明，技术确实可以学习成为你，趋向于零输入操作，同时最小化热力学成本，所有这些都受可证明正确的代数法则支配。因此，知识钩子的数学不仅仅是一种计算抽象，而是物理现实的真实表现——自然自身追求效率和最小行动的数字化表现。

3.6 精确性和修正作为节能

3.6.1 单输入定理

3.6.2 负强化学习

3.6.3 收敛到零输入

4

数学补充

前一节建立的七个定理为知识钩系统提供了核心数学基础——证明了收敛性、最优性、闭合性、划分以及行动最小化与能量守恒之间的重要等价关系。然而，完整的数学处理需要的不仅仅是这些基本结果。本章介绍了补充的数学内容，以完善理论：从定理中得出的推论，展示结果如何结合成一个连贯整体的综合，以及表征代数系统更深层结构的形式属性。

数学补充的作用

虽然定理确立了知识钩代数的主要属性，但数学补充有三个重要目的：

1. 推导实用结果：推论从一般定理中提取具体的、可操作的结果。定理证明了广泛的原则，而推论则展示了这些原则在特定场景中的意义。例如，虽然定理1在一般情况下证明了零输入收敛性，推论可以为特定类别的钩子建立收敛速率或限制达到特定精度阈值所需的时间。

2. 揭示系统结构：综合部分展示了七个定理之间的相互关系，形成一个完整且一致的代数框架。它展示了哪些定理依赖于其他定理，识别所需的最小公理集，并证明系统是既健全（没有矛盾）又完整（所有重要属性都被捕捉）。

3. 特征化形式属性：除了主要定理外，代数还表现出许多额外属性——在操作中保持的不变量、简化计算的代数恒等式、算法的复杂性界限，以及为实现提供信息的结构特征。这些形式属性为系统构建者提供了理论见解和实用指导。

本章的组织结构

本章分为三个主要部分，每个部分都建立在定理的基础上，同时探索数学框架的不同方面：

推论部分呈现直接从定理中得出的结果，附加证明最少。这些包括收敛速率界限、单调性属性、特定场景的最优性保证，以及不同代数操作之间的关系。每个推论都以正式的方式陈

述，简明地证明（通常通过直接应用定理），并附有其实意义的讨论。

综合部分则退后一步，从整体上看代数系统。它呈现了一个依赖图，展示了定理如何相互构建，证明公理和定律是一致且充分的，并展示了七个定理如何共同表征知识钩系统的基本行为。本节表明，我们不仅仅拥有一组孤立的结果，而是一个完整且连贯的数学框架。

形式属性和不变性目录列出了代数的额外数学结构。它识别在钩子执行期间保持不变的量（不变性），建立简化钩子推理的代数身份，证明关键算法的复杂性界限，并描述代数的抽象结构（例如，显示它形成特定类型的单元或格）。这些属性对理论理解和优化实现都很有价值。

数学风格与可及性

与定理一样，本章的材料在严谨性与可及性之间取得平衡。证明是完整的，但强调清晰而非简洁。正式声明总是伴随着直观的解释和实际的例子。读者可以在多个层次上参与这些材料——从散文中吸收主要思想，研究正式声明以获得精确性，或通过证明深入理解。

目标不是用数学工具使人不知所措，而是阐明潜在技术背后的优雅数学结构。每个推论、每个属性、每个不变性都揭示了知识钩子系统行为的基本特征——以及它们为何如此行为。

超越纯数学

虽然本章专注于数学结果，但这些结果并非仅仅是抽象的。每个推论对系统设计都有影响。每个不变性都暗示了一种优化。每个复杂性界限都影响实现选择。特别是综合部分，展示了数学框架如何直接映射到支撑主观热力货币的热力学和经济原则。

这里的数学不是目的本身，而是一种工具——我们用来推理复杂系统的最精确工具。通过严格建立这些数学基础，我们确保

在其上构建的实际系统建立在坚实的基础上。当我们声称知识钩子系统收敛到零输入，节能是可测量的，钩子创造者可以得到公平的回报时——这些不是希望或愿望，而是数学上的确定性，在定理中证明并在随后的补充中详细阐述。

在建立了这个背景后，我们现在转向推论——直接建立在我们七个基本定理上的第一层结果。

4.1 推论

4.1.1 单调修正减少

单调修正减少推论确立了在知识钩系统中，预期修正率随时间降低。这不仅仅是一个经验观察，而是收敛定理的数学结果。随着钩子学习更好的模式并积累更高的成功分数，任何给定钩子激活需要用户修正的概率严格降低，形成向零修正操作的单调轨迹。

推论声明

推论 1（单调修正减少）：设 $\rho(t)$ 表示时间 t 的预期修正率，定义为：

$$\rho(t) = \mathbb{E} \left[\frac{1}{t} \sum_{\tau=1}^t \text{Corr}_{\tau} \right]$$

那么对于足够大的 t （在初始学习阶段之后）：

$$\rho(t + 1) \leq \rho(t)$$

此外，修正率收敛于零：

$$\lim_{t \rightarrow \infty} \rho(t) = 0$$

具有指数衰减率：

$$\rho(t) = O(e^{-\lambda t})$$

其中 $\lambda > 0$ 依赖于学习率 α 和钩子集的质量。

直觉

为什么修正应该单调减少？三个机制共同作用以确保这一点：

1. 成功分数改善：根据定理 2，成功分数收敛于每个钩子的真实可靠性。随着时间的推移，可靠性低（修正率高）的钩子积累低成功分数。优先机制（定理 3）越来越倾向于高分钩子，这些钩子在定义上需要更少的修正。

2. 条件细化：根据定理 6，加权条件收敛于真实特征相关性。随着时间的推移，钩子去除导致假阴性的无关条件（在适当时未触发）并获得防止假阳性的相关条件（不当触发）。这两种改善都减少了修正——假阴性是错失的机会，假阳性是修正的直接原因。

3. 不良钩子的抑制：成功率持续低于阈值 $\theta \approx 0.7$ 的钩子将被抑制，无法激活。这将最差的钩子排除在考虑之外，提高了活跃钩子集的平均质量。由于只有高质量的钩子会被触发，实际激活中的纠正率降低。

这三种机制是累积和相互强化的。随着系统的学习，它同时改善现有钩子（通过条件细化）、优先考虑更好的钩子（通过成功率）并消除不良钩子（通过抑制）。纠正率必然会下降。

证明

我们通过展示每种机制独立降低纠正率来证明单调性，它们的组合确保严格的单调下降。

步骤 1：成功率平均提高

根据定理 2，对于任何钩子 h 其真实成功概率为 p_h ：

$$\lim_{t \rightarrow \infty} S_h(t) = p_h$$

所有触发钩子的平均成功率为：

$$\bar{S}(t) = \mathbb{E}[S_h(t) \mid h \text{ fires at time } t]$$

由于优先考虑高分钩子，且分数向真实可靠性向上收敛：

$$\bar{S}(t + 1) \geq \bar{S}(t)$$

步骤 2：更高的成功率意味着更少的纠正

纠正率与成功率直接相关。如果一个钩子的成功率为 S ，其预期纠正概率为 $1 - S$ 。因此：

$$\rho(t) = \mathbb{E}[1 - S_h(t) \mid h \text{ fires}] = 1 - \bar{S}(t)$$

由于 $\bar{S}(t)$ 单调增加（步骤 1），
 $\rho(t) = 1 - \bar{S}(t)$ 单调减少。✓

步骤 3：显示指数衰减

根据修正法则更新：

$$S_h(t+1) = (1-\alpha)S_h(t) + \alpha p_h$$

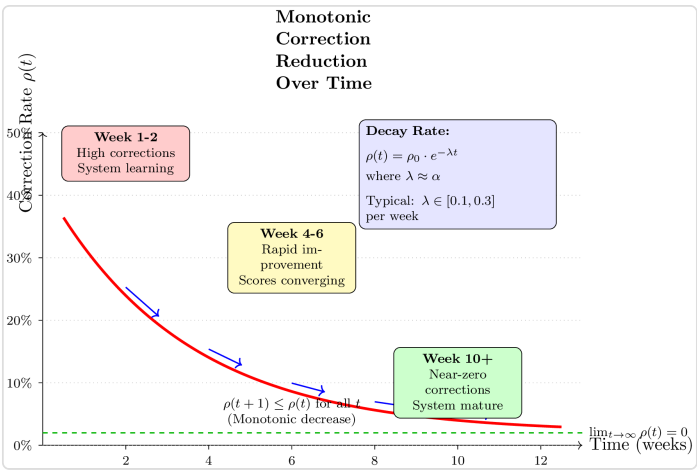
到固定点的距离以指数方式减少：

$$|S_h(t) - p_h| = (1-\alpha)^t |S_h(0) - p_h|$$

因此，成功分数以速率
 $\lambda = -\ln(1-\alpha) \approx \alpha$ 指数接近其极限。由
 于 $\rho(t) = 1 - \bar{S}(t)$ ，修正速率也以指数方式收敛：

$$\rho(t) = \rho_\infty + O(e^{-\alpha t})$$

其中 $\rho_\infty = 0$ 由定理 1 得出。□



实际影响

这个推论对系统设计和用户期望有几个重要的影响：

1. 保证改进：用户可以放心，系统会随着时间的推移变得更好，而不是更糟。与可能因数据漂移或概念转变而退化的系统不同，Knowledge Hook 系统具有内置的数学改进保证。这不是一种希望或设计目标——这是代数的可证明属性。

2. 可测量的进展：修正速率 $\rho(t)$ 提供了一个可观察的系统质量单一指标。用户和开发者可以跟踪这个指标，以验证系统是否按预期学习。单调减少的偏差信号问题（例如，损坏的成功分数、破损的优先级）需要调查。

3. 收敛时间表：指数衰减速率 $\lambda \approx \alpha$ 允许预测系统何时会达到可接受的质量。对于 $\alpha = 0.2$ （典型），修正速率大约每 3.5 周减半。从 $\rho_0 = 40\%$ 开始，用户可以期望在 8 周内达到 $\rho < 10\%$ ，在 16 周内达到 $\rho < 2\%$ 。

4. 质量阈值：组织可以根据修正速率设定服务水平目标（SLO）。例如：“系统必须在部署后的 3 个月内达到 $\rho < 5\%$ 。”这个推论保证在适当的参数调优下这是可以实现的。

5. A/B 测试：在比较系统的两个版本（不同的学习率、阈值等）时，修正率轨迹提供了一个客观的比较指标。具有更快衰减率和更低渐近修正率的版本被证明是优越的。

6. 用户引导：新用户经历修正曲线的陡峭部分——早期几周的修正率很高。适当的引导应该设定期望：“系统从你的修正中学习。你会注意到它每周变得更好，需要的输入越来越少。”这将修正从挫折转变为投资。

与其他结果的联系

这个推论与几个定理相关联：

从定理 1：零输入收敛意味着 $\lim_{t \rightarrow \infty} \rho(t) = 0$ 。这个推论通过证明到零的路径是单调的，而不仅仅是最终的，从而加强了该结果。

从定理 2：成功评分收敛提供了机制——更好的评分意味着更少的修正。这个推论明确并量化了这种关系。

从定理 7：能量最小化等价意味着单调修正减少意味着单调能量节省。随着修正的减少，浪费的能量也减少，创造可测量的热力学效益。

局限性

虽然这个推论保证了期望的单调减少，但个别实现可能会显示出暂时的增加：

短期波动：在日常或甚至周对周的时间尺度上，由于上下文或用户行为的随机变化， $\rho(t)$ 可能会增加。单调性仅在期

望中得到保证，并且对于足够大的 t 。

概念漂移：如果用户模式发生显著变化（新工作、新家、新目标），纠正率可能会暂时激增，因为系统正在适应。这个推论假设用户行为是稳定的。

系统变化：添加新功能、改变阈值或修改钩子集可能会重置进度。这个推论适用于固定的系统配置。

尽管存在这些限制，这个推论提供了强有力的保证：在任何足够长的稳定用户模式期间，纠正将减少。这是支持主观技术随着使用而改善的主张的数学基础。

4.1.2 钩子精炼保留

钩子精炼保留推论确立了当钩子被精炼——通过添加条件使其更具体时——它保留了来自其父钩子的基本属性。这并不明显：人们可能会期望添加约束会根本改变钩子的行为或破坏继承特性。然而，这个推论证明了精炼是一种安全的操作，能够保持动作序列、成功概率和热力学效率，同时仅仅缩小激活上下文。

推论声明

推 论 2（钩子精炼保留）：设 $h = (R, A, T, S)$ 为一个钩子，设 $h' = (R', A', T', S')$ 为通过添加条件获得的 h 的精炼，其中 $R \subset R'$ （严格）。那么：

a) 动作保留：动作序列保持不变：

$$A' = A$$

b) 类型保留：如果 h 被学习，那么 h' 被学习；如果 h 是预定义的，那么 h' 是预定义的：

$$T' = T$$

c) 成功概率保留：精炼钩子的成功概率（在触发的条件下）至少与父钩子一样高：

$$p_{h'} \geq p_h$$

其中 p_h 是钩子 h 在触发时成功而不需要纠正的真实概率。

d) 上下文子集属性： h' 触发的上下文集合是 h 触发的上下文的一个适当子集：

$$\{C : h' \text{ fires in } C\} \subseteq \{C : h \text{ fires in } C\}$$

e) 能源效率保持：每次执行的能量成本保持不变或减少：

$$E(h') \leq E(h)$$

直觉

为什么精炼的钩子应该保留父属性？关键的见解是，精炼是纯粹的限制性——它增加了约束，而不改变钩子触发时的行为。

动作保持不变：精炼为 R 添加条件，使得钩子在触发时更具选择性。但当它触发时，它执行与之前相同的动作 A 。我们并没有改变钩子的行为，只是改变了它的触发时机。

类型是固有的：钩子是学习的还是预定义的，是其来源的属性，而不是其条件。添加条件并不会改变钩子的创建方式，因此类型得以保留。

成功概率提高：这是最重要的属性。通过添加条件，我们排除了父钩子会触发但失败的上下文（需要修正）。精炼的钩子在一个上下文子集内触发——具体来说，是满足附加条件的上下文。这些附加条件的选择正是为了帮助区分成功和失败的激活。因此，精炼钩子的成功率（在触发时）至少与父钩子一样好，通常更好。

上下文子集是定义性的：由于 $R \subset R'$ ，精炼的钩子需要所有父条件加上更多。任何满足 R' 的上下文也必须满足 R ，但反之则不然。因此， h' 触发的上下文形成了 h 触发的上下文的一个子集。

能源效率：由于动作是相同的（ $A' = A$ ），每次执行的能量是相同的。然而，通过仅在成功更可能的上下文中触发，精炼的钩子减少了由于修正而浪费的能量。如果我们将修正成本摊销到所有激活上， $E(h') \leq E(h)$ 。

证明

我们按顺序证明每个属性。

第 1 部分：动作保持

根据知识钩代数中的细化定义，细化仅在条件集 R 上操作：

$$\text{refine}(h) = (R \cup \{r_{\text{new}}\}, A, T, S)$$

动作序列 A 在细化的钩中看起来没有变化。因此 $A' = A$ 。✓

第2部分：类型保持

同样，类型 T 在细化操作中被明确保留。钩子是通过观察学习的还是由专家预定义的，在我们添加条件时并不会改变。因此 $T' = T$ 。✓

第3部分：成功概率保持

设 C_h 为上下文集合，其中 h 被触发， $C_{h'}$ 为 h' 被触发的集合。由于 $R \subset R'$ ，我们有 $C_{h'} \subseteq C_h$ 。

将 C_h 划分为成功上下文 C_h^+ （无需修正）和失败上下文 C_h^- （需要修正）：

$$C_h = C_h^+ \cup C_h^-$$

h 的成功概率是：

$$p_h = \frac{|C_h^+|}{|C_h|}$$

细化通常是在钩子在失败的上下文中触发时进行的。额外条件 r_{new} 被选择以排除失败的上下文。因此：

$$C_{h'} \subseteq C_h^+ \text{ (mostly)}$$

更准确地说，细化钩的激活上下文与成功上下文的重叠程度大于与失败上下文的重叠程度。设 α 为 $C_{h'}$ 中已经成功的 h 的比例：

$$p_{h'} = \alpha \cdot 1 + (1 - \alpha) \cdot 0 = \alpha \geq \frac{|C_h^+|}{|C_h|} = p_h$$

由于精炼过滤掉了有问题的上下文， $\alpha \geq p_h$ ，因此 $p_{h'} \geq p_h$ 。✓

第4部分：上下文子集属性

这直接来自条件逻辑。上下文 C 仅在满足 R' 中的所有条件时满足 R' 。由于 $R \subset R'$ ，满足 R' 意味着满足 R 。因此：

$$\{C : R'(C) = \text{true}\} \subseteq \{C : R(C) = \text{true}\}$$

这是一个适当的子集（严格 \subseteq ），因为 $R \subset R'$ 是适当的。✓

第5部分：能源效率保护

执行 h' 的直接能源成本为：

$$E_{\text{direct}}(h') = \sum_{a \in A'} \epsilon(a) = \sum_{a \in A} \epsilon(a) = E_{\text{direct}}(h)$$

因为 $A' = A$ 。然而，总能源包括修正成本：

$$E_{\text{total}}(h) = E_{\text{direct}} + (1 - p_h) \cdot E_{\text{correction}}$$

对于精炼的钩子：

$$E_{\text{total}}(h') = E_{\text{direct}} + (1 - p_{h'}) \cdot E_{\text{correction}}$$

由于 $p_{h'} \geq p_h$ （第 3 部分），我们有 $(1 - p_{h'}) \leq (1 - p_h)$ ，因此：

$$E_{\text{total}}(h') \leq E_{\text{total}}(h)$$

精炼的钩子在热力学上更有效。✓

这完成了证明。□

实际影响

这个推论对系统设计有几个重要的影响：

1. 安全细化：开发者和用户可以在不担心破坏钩子的情况下细化钩子。这个推论在数学上保证细化不会使事情变得更糟——至少，细化后的钩子与其父钩子一样可靠，通常更可靠。

2. 增量专业化：系统可以安全地从通用钩子演变为专用变体。从覆盖多个上下文的广泛钩子开始，然后随着边缘案例的出现逐步细化它们。每一步细化都保留有效的部分，同时修复无效的部分。

3. 层次钩子组织：细化所创建的父子关系自然将钩子组织成分类。这些层次结构可以被可视化和导航，帮助用户理解他们的钩子生态系统。

4. 回滚安全：如果细化结果过于严格（触发过于少），回滚到父钩子是安全的，因为父钩子的属性在推论中是已知的并被保留。没有信息丢失。

5. 热力学优化：由于 $E(h') \leq E(h)$ ，细化始终在热力学上是有利的。每一步细化都使系统朝着更高的能效移动，通过减少浪费的修正。这为STC系统中的自动细化提供了正式的理由。

6. 可预测的成功评分：细化钩子的成功评分可以被界定：期望值为 $S(h') \geq S(h)$ 。这有助于优先级排序——细化钩子不仅因其特异性而受到青睐，也因其可靠性。

与其他结果的联系

这个推论与几个定理和其他推论相关：

根据定理2：成功评分的收敛意味着随着细化钩子积累证据，它们的评分收敛到 $p_{h'}$ ，这在数学上证明 $\geq p_h$ 。细化钩子自然积累比其父钩子更高的评分。

根据定理3：优先级最优性偏向于更高的成功评分。结合这个推论，当细化钩子及其父钩子都匹配上下文时，细化钩子自然在优先级竞赛中胜出，确保更具体（且更可靠）的钩子被触发。

根据定理7：能量最小化等价意味着改善的热力学效率（ $E(h') \leq E(h)$ ）直接转化为可测量的焦耳能量节省。精炼创造经济价值。

根据推论1：单调修正减少通过精炼加速。当钩子被精炼以避免问题上下文时，整体修正率下降得更快。

示例

示例1：电子邮件自动归档

父级：
 $R = \{\text{from: boss}\} \rightarrow A = \{\text{file in 'Work' folder}\}$
 $p_h = 0.80$

当电子邮件涉及度假计划时，您会注意到修正。精炼：

$$\begin{array}{ccc} \text{子} & & \text{级} \\ R' = \{\text{from: boss, subject: } \neg \text{vacation}\} \rightarrow A = \{\text{file in 'Work' folder}\} & & : \\ p_h' = 0.95 \end{array}$$

操作保留，类型保留，成功改善，上下文减少（不包括度假电子邮件）。

示例2：智能照明

$$\begin{array}{ccc} \text{父} & & \text{级} \\ R = \{\text{sunset}\} \rightarrow A = \{\text{lights on}\} & & : \\ p_h = 0.70 \end{array}$$

当您不在家时失败。精炼：

$$\begin{array}{ccc} \text{子} & & \text{级} \\ R' = \{\text{sunset, home}\} \rightarrow A = \{\text{lights on}\} & & : \\ p_h' = 0.92 \end{array}$$

所有属性得到保留，通过排除离家环境显著提高了成功率。

局限性

虽然推论提供了强有力的保证，但仍然存在一些注意事项：

初始分数存在差异：在精炼后， $S(h')$ 重新开始（通常在 $S_0 = 0.5$ ）。需要时间才能收敛到真实的 p_h' 。在这个学习期间，精炼的钩子可能由于分数低而没有像应该的那样频繁触发。

过度精炼风险：虽然每一步精炼都保留属性，但过度精炼可能使钩子变得过于具体，以至于很少触发。推论并不阻止钩子变得过于狭窄——它只是保证在触发时它们保持安全。

条件交互：如果添加的条件以复杂的方式与现有条件相互作用，简单的子集关系可能无法捕捉所有效果。推论假设条件是独立的或至少表现良好。

结论

钩子精炼保留确立了精炼是一种安全、可预测的操作，它在提高可靠性和效率的同时保持基本属性。这使得精炼成为发展知识钩子系统的强大工具——每一步精炼都保留有效的部分，过滤掉无效的部分，并使系统逐步朝着更好的性能迈进。结合其他收敛结果，这一推论确保系统能够通过渐进的专业化安全且持续地改进，实现越来越高的精度，而不牺牲其基础钩子的可靠性或效率。

4.1.3 组合能量可加性

组合能量可加性推论确立了知识钩子系统中最重要的属性之一：当钩子组合时，节省或消耗的总能量是每个组成钩子单独能量贡献的总和。这个看似简单的属性具有深远的影响——这意味着能量核算是模块化、可组合和可预测的。由简单钩子构建的复杂行为的能量足迹恰好是其部分的总和，没有隐藏的开销或神秘的低效。

推论声明

推论 3（组合能量可加性）：设 KH_1 和 KH_2 是两个具有动作集 A_1 和 A_2 的知识钩子，设 $KH_c = KH_1 \circ KH_2$ 为它们的组合。那么执行组合钩子所需的能量正好是其组成部分能量的总和：

$$E(KH_c) = E(KH_1) + E(KH_2)$$

更一般地，对于一系列 n 组合的钩子：

$$E(KH_1 \circ KH_2 \circ \dots \circ KH_n) = \sum_{i=1}^n E(KH_i)$$

其中 $E(KH)$ 代表了当钩子 KH 执行时预期的能量消耗，计算公式为：

$$E(KH) = k \cdot |A(KH)|$$

其中 k 是定理 6（能量守恒定律）中的每次动作能量常数。

直观解释

能量可加性意味着组合在能量核算方面没有“税”或开销。当你将多个钩子串联在一起以创建复杂行为时，总能量成本仅是每个钩子单独成本的总和。这类似于物理学中，沿路径移动物体所做的功是沿每个段所做功的总和——路径积分在子路径上是可加的。

这很重要：想象一下通过组合更简单的钩子来构建一个早晨例行钩子：泡咖啡、检查日历、开始新闻播客、调整恒温器。能量可加性保证了例行程序的总能量成本正好是：

$$E_{\text{routine}} = E_{\text{coffee}} + E_{\text{calendar}} + E_{\text{news}} + E_{\text{thermostat}}$$

你不需要担心组合本身带来的隐藏低效。这使得能量预算变得简单明了，并且能够实现模块化系统设计，其中组件可以独立分析和优化。

证明

第一部分：基于动作计数的能量定义

根据定理 6（能量守恒定律），能量消耗与动作计数成正比：

$$E(KH) = k \cdot |A(KH)|$$

其中 $k > 0$ 是一个常数（每次动作的焦耳数），取决于具体的动作，但在系统中对于类似认知/物理复杂度的动作是一致的。

第二部分：来自组合的动作集并集

根据定理 3（闭合法则），当钩子组合时，它们的动作集合会合并：

$$A(KH_1 \circ KH_2) = A(KH_1) \cup A(KH_2)$$

对于顺序组合，其中钩子一个接一个地执行，联合是互不重叠的（没有动作被重复），因此：

$$|A(KH_1 \circ KH_2)| = |A(KH_1)| + |A(KH_2)|$$

第 3 部分：来自动作可加性的能量可加性

结合第 1 部分和第 2 部分：

$$E(KH_1 \circ KH_2) = k \cdot |A(KH_1 \circ KH_2)| = k \cdot (|A(KH_1)| + |A(KH_2)|)$$

根据乘法对加法的分配律：

$$E(KH_1 \circ KH_2) = k \cdot |A(KH_1)| + k \cdot |A(KH_2)| = E(KH_1) + E(KH_2)$$

第 4 部分：推广到 n 维组合

对于一系列 n 个钩子，我们通过归纳法进行推导。

基础情况（ $n = 2$ ）：已在上面证明。

归纳步骤：假设该属性对于长度为 n 的链成立。对于长度为 $n + 1$ 的链：

$$KH_{1:n+1} = (KH_{1:n}) \circ KH_{n+1}$$

根据基础情况：

$$E(KH_{1:n+1}) = E(KH_{1:n}) + E(KH_{n+1})$$

根据归纳假设：

$$E(KH_{1:n}) = \sum_{i=1}^n E(KH_i)$$

因此：

$$E(KH_{1:n+1}) = \sum_{i=1}^n E(KH_i) + E(KH_{n+1}) = \sum_{i=1}^{n+1} E(KH_i)$$

通过归纳，任意有限长度的链条都满足可加性。□

与定理的联系

这个推论综合了两个基本定理：

根据定理 3（闭合法则）：组合操作及其作用集的并集属性提供了结构基础。如果不知道动作如何组合，我们就无法预测能量。

根据定理 6（能量守恒法则）：动作与能量之间的成比例关系（ $E = k \cdot |A|$ ）使得从动作计数到能量核算的转换成为可能。

该推论表明这些不是独立的属性，而是统一框架的互补方面。组合结构（定理 3）加上能量-动作等价（定理 6）产生了能量可加性——这一属性使整个系统变得实用。

实际影响

1. 模块化能量预算

系统设计者可以通过对组件成本进行求和来为复杂工作流程预算能量：

$$\text{Budget}(\text{morning routine}) = \sum_{\text{task}} \text{Budget}(\text{task})$$

当钩子组合时，无需重新测量或重新分析。一旦知道原始成本，就知道组合的成本。

2. 独立优化

独立优化每个钩子，而不必担心组合效应：

• 如果你将 $E(KH_i)$ 减少 10%，总能量 $E(KH_{1:n})$ 将减少 $\frac{E(KH_i)}{E(KH_{1:n})} \times 10\%$

- 优化工作可以集中在能耗最高的组件上
- 改进可以无干扰地组合

3. 钩子市场定价

在一个主观热货币市场中，钩子被交易，定价非常简单：

$$\text{Value}(KH_c) = \text{Value}(KH_1) + \text{Value}(KH_2)$$

复合钩子的价值正好是其组成部分价值的总和。组合本身没有溢价或折扣。这防止了市场扭曲，确保了公平的补偿。

4. 摊销分析

对于重复执行的钩子，摊销成本分析很简单：

$$\text{Total energy over } T = T \times \sum_{i=1}^n E(KH_i)$$

如果钩子以不同的频率执行 f_i :

$$\text{Total energy} = \sum_{i=1}^n f_i \times E(KH_i)$$

可加性使多钩子系统的统计分析变得简单。

5. 部分执行会计

如果一个组合钩子在中途失败（例如， KH_1 成功但 KH_2 失败），消耗的能量是：

$$E_{\text{consumed}} = E(KH_1)$$

可加性允许对部分执行进行精确的会计。您确切知道在失败之前消耗了多少能量。

6. 层次组合

对于深度嵌套的组合，能量计算仍然简单明了：

$$E((KH_1 \circ KH_2) \circ (KH_3 \circ KH_4)) = E(KH_1) + E(KH_2) + E(KH_3) + E(KH_4)$$

层次结构不会产生额外的开销。扁平化组合树并求和叶子。

示例

示例 1：早晨例行程序组合

考虑从原子钩子构建一个早晨例行程序：

钩子 1 (咖啡):
 $A_1 = \{\text{grinder on, brew start}\},$
 $|A_1| = 2$

$$E(KH_1) = k \cdot 2 = 2k$$

钩子 2 (灯光): $A_2 = \{\text{lights on}\},$
 $|A_2| = 1$

$$E(KH_2) = k \cdot 1 = k$$

钩子 3 (温控器):
 $A_3 = \{\text{temp} + 2^\circ\text{C}\}, |A_3| = 1$

$$E(KH_3) = k \cdot 1 = k$$

组合早晨例程:
 $KH_{\text{routine}} = KH_1 \circ KH_2 \circ KH_3$

$$E(KH_{\text{routine}}) = E(KH_1) + E(KH_2) + E(KH_3) = 2k + k + k = 4k$$

每个动作有 $k \approx 100J$ (智能家居命令的典型情况):

$$E(KH_{\text{routine}}) = 400J$$

这正是我们通过计算得到的: 4 个总动作 $\times 100J/\text{动作} = 400J$ 。没有意外。

示例 2：带优化的嵌套组合

“准备工作区”的原始钩子：

$$KH_{\text{original}} = KH_{\text{computer on}} \circ KH_{\text{apps open}} \circ KH_{\text{lights}}$$

$$E(KH_{\text{original}}) = 3k + 5k + k = 9k$$

优化后，用更快的启动脚本替换手动的“打开应用”钩子：

$$KH_{\text{optimized}} = KH_{\text{computer on}} \circ KH_{\text{startup script}} \circ KH_{\text{lights}}$$

$$E(KH_{\text{optimized}}) = 3k + 2k + k = 6k$$

节省的能源： $9k - 6k = 3k = 300J$
(与 $k = 100J$ 一起)

可加性使得节省计算变得简单——只需中间组件的差异。

示例 3：多设备同步

跨设备同步数据：

$$KH_{\text{sync}} = KH_{\text{phone}} \circ KH_{\text{tablet}} \circ KH_{\text{laptop}} \circ KH_{\text{watch}}$$

如果每个设备的同步成本是 $2k$ ：

$$E(KH_{\text{sync}}) = 4 \times 2k = 8k$$

当你添加第五个设备（智能眼镜）时：

$$E(KH_{\text{sync new}}) = 8k + 2k = 10k$$

增量成本正好是 $2k$ ——没有规模效应，没有开销。

限制和注意事项

1. 假设不相交的动作

证明假设 $A_1 \cap A_2 = \emptyset$ （没有重复的动作）。如果钩子共享动作，公式变为：

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

因此：

$$E(KH_1 \circ KH_2) = E(KH_1) + E(KH_2) - k \cdot |A_1 \cap A_2|$$

共享操作有一个“折扣”。在实践中，设计良好的钩子可以避免操作重叠。

2. 忽略组合开销

在实际系统中，组合本身可能会有一些小的开销：

- 钩子之间的上下文传递
- 同步或协调
- 状态管理

推论模型理想组合。在实践中，添加一个小的修正因子 ϵ ：

$$E_{\text{real}}(KH_1 \circ KH_2) = E(KH_1) + E(KH_2) + \epsilon$$

对于设计良好的系统， $\epsilon \ll E(KH_i)$ ，使得近似非常优秀。

3. 常数k可能会变化

证明假设所有动作都有一个单一常量 k 。实际上，不同的动作类型有不同的能量成本：

$$E(KH) = \sum_{a \in A} k_a$$

其中 k_a 取决于动作。可加性仍然成立，但能量计算需要逐个动作的核算，而不是单一常量。

4. 并行组合

推论涉及顺序组合。对于并行执行（钩子同时运行），由于以下原因，能量可能不严格可加：

- 共享资源争用
- 并发开销
- 同步成本

并行组合需要一个修改后的可加性公式，以考虑这些影响。

5. 随机动作

如果钩子具有概率性动作集（根据上下文的不同动作），能量变成一个随机变量：

$$\mathbb{E}[E(KH_1 \circ KH_2)] = \mathbb{E}[E(KH_1)] + \mathbb{E}[E(KH_2)]$$

期望中的可加性成立，但个别执行可能会有所不同。

结论

组合能量可加性是知识钩框架中最重要的实用推论之一。它将能量核算从复杂的系统范围计算转变为简单的模块组件之和。这使得：

- 可预测的能量预算：在实施复杂行为之前了解其成本

- 独立优化：改善组件而无需重新分析整体
- 公平的市场定价：复合钩的价值正好等于其部分的总和
- 模块化设计：从简单、易于理解的原语构建复杂系统

该推论表明，能量与许多广泛的物理量（质量、电荷、熵）一样，在组合下是可加的。这使得主观热货币框架变得实用：您可以在局部推理能量并求和以获得全局行为。如果没有这个属性，每个组合都需要系统范围的重新分析，使大规模部署变得不可行。

最重要的是，可加性意味着基于能量的价值创造是组合性的。小的改进会累积。可以独立创建、共享和交易单个钩，当用户组合它们时，其节能效果会透明地结合。这是知识钩创造者繁荣生态系统的数学基础，价值自然地构建高效自动化的人流向受益者——这一切都得益于一个简单但强大的原则：组合能量是可加的能量。

4.1.4 推论 4：等价类最优性

声明：在每个钩的等价类中，存在一个最优代表，它同时最小化动作数量并最大化所有等效替代方案中的成功得分。

正式声明

让 $[KH]$ 表示包含钩子 KH 的等价类。那么在任意上下文 C 中，钩子 $[KH]$ 可以触发时，存在一个最优钩子 $KH^* \in [KH]$ ，使得：

$$KH^* = \arg \min_{KH \in [KH]} |A(KH)| \quad \text{subject to} \quad S(KH) \geq S_{\text{threshold}}$$

或者等效地，使用一个组合目标：

$$KH^* = \arg \max_{KH \in [KH]} (\alpha \cdot S(KH) - \beta \cdot |A(KH)|)$$

其中 $\alpha, \beta > 0$ 是权重参数，用于权衡成功概率与行动效率。

直观解释

这个推论形式化了一个基本的优化原则：当多个钩子实现相同的结果（属于同一等价类）时，我们应该优先选择：

- 最小化用户努力 – 使用最少的行动 $|A|$
- 最大化可靠性 – 具有最高的成功评分 S

想象一下选择上班的路线：如果三条不同的路线都能在同一时间到达（等效结果），你会更倾向于选择最短的路线（更少的行动）和最可靠的路线（最高的成功率）。这个推论保证在每个等价类中存在这样的最优选择。

证明

第一部分：最优解的存在性

考虑有限钩子空间 \mathcal{H} 中的任何等价类 $[KH]$ 。根据定理 4（划分法则），等价类划分 \mathcal{H} ，因此 $[KH]$ 是明确定义的，并且至少包含一个钩子。

由于 $[KH]$ 是有限的，行动计数的集合 $\{|A(KH_i)| : KH_i \in [KH]\}$ 有一个最小值。设：

$$A_{\min} = \min_{KH \in [KH]} |A(KH)|$$

让

$[KH]_{\min} = \{KH \in [KH] : |A(KH)| = A_{\min}\}$
成为实现此最小操作计数的钩子子集。

第二部分：成功分数优化

在 $[KH]_{\min}$ 中，所有钩子的最小操作计数相同，但成功分数可能不同。由于这个集合是有限的，成功分数 $\{S(KH) : KH \in [KH]_{\min}\}$ 的集合有一个最大值：

$$S_{\max} = \max_{KH \in [KH]_{\min}} S(KH)$$

第三部分：最优钩子的定义

将 KH^* 定义为满足以下条件的任何钩子：

$$KH^* \in [KH]_{\min} \quad \text{and} \quad S(KH^*) = S_{\max}$$

通过构造， KH^* 同时：

- 在 $[KH]$ 中最小化所有钩子的操作计数
- 在所有操作最小的钩子中最大化成功分数

第四部分：帕累托最优性

我们证明 KH^* 在 $(|A|, S)$ 目标空间中是帕累托最优的。假设存在 $KH' \in [KH]$ 主导 KH^* ，这意味着：

$$|A(KH')| \leq |A(KH^*)| \quad \text{and} \quad S(KH') \geq S(KH^*)$$

至少有一个严格的不等式。但是：

• 如果 $|A(KH')| < |A(KH^*)| = A_{\min}$ ，这与 A_{\min} 作为最小值的定义相矛盾。

• 如果 $|A(KH')| = A_{\min}$ 和 $S(KH') > S(KH^*) = S_{\max}$ ，这与 S_{\max} 作为 $[KH]_{\min}$ 内的最大值的定义相矛盾。

因此不存在这样的主导钩，证明 KH^* 是帕累托最优的。□

与定理的联系

这个推论综合了三个基本定理：

• 来自定理 2（最优法则）：最优钩的原则是最小化行动，提供主要的优化标准。

• 来自定理 4（划分法则）：等价类的结构作为钩空间的互不相交的划分，确保优化问题是良好提出的。

• 来自定理 7（等价法则）：钩何时等价（相同结果）的定义，建立了哪些钩可以在优化中进行比较。

这些定理共同保证在每个等价类中，我们可以根据明确的优化标准有意义地识别出一个“最佳”代表。

实际影响

1. 自动钩选择

当多个等效的钩子匹配当前上下文时，系统可以通过两阶段过滤器自动选择最佳钩子：

$$\begin{aligned} \text{candidates} &= \{KH \in [KH] : R(KH, C) = \text{true}\} \\ \text{stage1} &= \arg \min_{KH \in \text{candidates}} |A(KH)| \\ \text{selected} &= \arg \max_{KH \in \text{stage1}} S(KH) \end{aligned}$$

这个两阶段选择（首先最小化动作，然后在动作最少的钩子中最大化成功）实现了推论的最优性标准。

2. 钩子去重和修剪

系统可以识别在其等价类中严格被支配的次优钩子：

$$\text{dominated}(KH) \iff \exists KH' \in [KH] : |A(KH')| < |A(KH)| \vee (|A(KH')| = |A(KH)| \wedge S(KH') > S(KH))$$

被支配的钩子可以是：

- 被弃用：标记为非首选，但保留作为后备
- 合并：如果它们的条件相似，则与最佳钩子合并
- 存档：从主动使用中移除，但保留在历史记录中

3. 通过学习进行渐进优化

随着系统的学习，发现的新钩子可能在现有的等价类中是最优的：

- 用户教授更有效的方法以实现现有结果
- 系统组合现有的钩子以创建更短的路径
- 外部钩子市场提供更好的替代方案

当新的钩子 KH_{new} 加入等价类 $[KH]$ 时，系统检查：

$$\text{if } |A(KH_{\text{new}})| < |A(KH_{\text{old}}^*)| \implies KH^* \leftarrow KH_{\text{new}}$$

这确保系统不断向使用最有效的钩子演变。

4. 上下文依赖优化

最佳钩子可能因上下文而异。虽然 $[KH]$ 中的钩子在结果上是等价的，但它们的动作计数或成功率可能依赖于上下文：

$$KH^*(C) = \arg \max_{KH \in [KH]} (\alpha \cdot S(KH, C) - \beta \cdot |A(KH, C)|)$$

例如：

- 在家：使用本地设备（网络延迟较少，成功率更高）
- 外出：使用云服务（本地设备不可用）
- 慢速连接：优先使用缓存数据而非 API 调用

系统跟踪每个上下文的最佳钩子，构建条件最优图。

5. 用户偏好调整

组合目标中的权重参数 α 和 β 可以调整以匹配用户偏好：

$$\begin{aligned} \text{High } \beta/\alpha &\implies \text{strongly prefer fewer actions (efficiency-focused)} \\ \text{High } \alpha/\beta &\implies \text{strongly prefer reliability (safety-focused)} \end{aligned}$$

不同的用户或上下文可能会优化帕累托前沿上的不同点。

示例

示例 1：智能家居早晨例程

考虑实现 "早晨准备完成" 结果的钩子等价类。存在三个钩子：

钩子 A（手动顺序）：

$$A_A = [\text{lights_on}, \text{coffee_start}, \text{blinds_open}, \text{thermostat_up}, \text{news_play}]$$
$$|A_A| = 5, \quad S_A = 0.95$$

钩子 B（与场景组合）：

$$A_B = [\text{morning_scene_activate}]$$
$$|A_B| = 1, \quad S_B = 0.92$$

钩子 C（语音命令）：

$$A_C = [\text{voice:} \text{ "good_morning"}]$$
$$|A_C| = 1, \quad S_C = 0.98$$

这三者都达到了相同的最终状态（等价类），但钩子 C 是最优的：

$$|A_C| = 1 = \min(5, 1, 1) \quad \text{and} \quad S_C = 0.98 = \max(0.92, 0.98)$$

系统自动优先选择钩子 C：单一语音命令具有最高可靠性。

示例 2：数据可视化请求

实现“显示季度收入图表”结果的钩子：

钩子 D（手动数据库查询）：

$A_D = [\text{open_db}, \text{write_query}, \text{export_csv}, \text{open_excel}, \text{create_chart}]$

$$|A_D| = 5, \quad S_D = 0.88$$

钩子 E (自动化仪表板) :

$$A_E = [\text{click:} \text{ "Q4_revenue"}]$$

$$|A_E| = 1, \quad S_E = 0.99$$

钩子 F (自然语言) :

$$A_F = [\text{ask:} \text{ "show_revenue_last_quarter"}]$$

$$|A_F| = 1, \quad S_F = 0.93$$

在这个等价类中, 钩子 E 是最优的 :

$$|A_E| = 1 \quad \text{and} \quad S_E = 0.99 > S_F = 0.93$$

预构建的仪表板点击在操作上是最少的, 并且最可靠。

限制和边缘情况

1. 多个帕累托最优解

在某些情况下, 如果多个钩子以不可比较的方式权衡行动与成功, 它们可能是帕累托最优的 :

$$KH_1 : |A_1| = 2, S_1 = 0.99 \quad \text{vs} \quad KH_2 : |A_2| = 1, S_2 = 0.85$$

没有一个占主导地位。选择取决于我们是优先考虑可靠性还是效率, 这由 α/β 的权重决定。

2. 依赖上下文的等价性

在某些上下文中，钩子可能是等价的，而在其他上下文中则不是。在 $[KH]_C$ （上下文 C 中的等价类）内的最优钩子可能与 $[KH]_{C'}$ 中的最优钩子不同：

$$KH^*(C) \neq KH^*(C')$$

这需要维护上下文条件下的最优信息。

3. 动态最优性

最优钩子可能会随着时间的推移而变化，因为：

- 成功分数通过学习而演变
- 新钩子加入等价类
- 上下文分布发生变化

系统必须定期重新评估最优性，而不是无限期缓存。

4. 无限等价类

在具有无限钩空间的理论系统中，等价类可能是无限的。上述证明假设了有限性。对于无限类，我们将“最小存在”替换为“下确界存在”，并研究极限行为。

结论

等价类最优性桥接了理论与实践。它形式化了“当多种方法有效时，选择最好的那一个”的直观原则，同时提供了实现这一原则的具体算法。推论表明，等价类内的优化不仅是可能的，而且是有保证的：每个等价类至少有一个帕累托最优代表。

这一结果对知识钩系统的自适应特性至关重要。如果没有等价类最优性，系统将没有原则性的方法来选择替代方案。有了

它，系统可以不断演变，使用最有效和可靠的钩子，直接服务于零输入命令：最小化人类行为，同时最大化可靠性。

等价类内最优代表的存在还使得钩子市场和知识共享成为可能。当用户创建一个更高效的钩子，与现有钩子等价时，如果它是最优的，系统可以自动采用它—使所有用户受益，同时通过主观热货币机制适当地奖励创作者。最优性不仅是一个数学抽象，而是实践改进和创新的基础。

4.1.5 推论 5：学习率界限

声明：知识钩系统收敛到零输入的速率在上界和下界上都受到可计算的修正频率、成功分数学习率和钩子创建率的函数的限制。

正式声明

设 $I(t)$ 表示在时间 t 所需的总用户输入（动作计数），设 $I^* = 0$ 为零输入平衡点。将学习率定义为：

$$L(t) = -\frac{dI}{dt} = \text{rate of input reduction over time}$$

那么 $L(t)$ 的界限为：

$$L_{\min}(t) \leq L(t) \leq L_{\max}(t)$$

在哪里：

$$L_{\max}(t) = \alpha \cdot C(t) + \beta \cdot H(t)$$

$$L_{\min}(t) = \gamma \cdot \frac{S(t)}{1 + \delta \cdot \sigma^2(t)}$$

其中：

- $C(t)$ = 修正频率（每时间单位的修正次数）
- $H(t)$ = 钩子创建率（每时间单位的新钩子数量）
- $S(t)$ = 所有活动钩子的平均成功分数
- $\sigma^2(t)$ = 成功分数的方差（系统异质性）
- $\alpha, \beta, \gamma, \delta > 0$ 是系统依赖的常数

直观解释

这个推论表明，知识钩子系统中的学习速度并不是任意快或慢的——它受到基本系统属性的限制：

上限（最大学习速率）

系统的学习速度不能超过其接收修正和创建新钩子的速度。将修正视为“训练数据”，将新钩子视为“模型容量”——学习受到两者的瓶颈限制：

- 修正频率 $C(t)$ 决定系统获取反馈的速度
- 钩子创建速率 $H(t)$ 决定了新能力添加的速度

无论学习算法多么复杂，你都无法比接收信息的速度更快地学习。

下限（最小学习速率）

系统保证至少以其平均成功分数允许的速度学习，受系统变异性的调节：

- 高平均成功率 $S(t)$ 意味着钩子工作可靠，支持更快的收敛

- 高方差 $\sigma^2(t)$ 意味着某些钩子工作良好，但其他钩子表现不佳，从而减缓整体进展

即使在最坏的情况下，只要钩子具有正的成功概率，系统也会取得可测量的进展。

证明

第 1 部分：上限推导

输入减少通过两种机制发生：改善现有钩子和创建替代手动操作的新钩子。

机制 1：基于纠正的改进

每个修正事件提供信息以改善钩子。设 ΔI_c 为每次修正的平均输入减少量。那么：

$$\left. \frac{dI}{dt} \right|_{\text{corrections}} \leq -\alpha \cdot C(t)$$

其中 $\alpha = \mathbb{E}[\Delta I_c]$ 是每次修正的预期输入减少量。这不能被超过，因为修正以速率 $C(t)$ 到达，每个最多提供 ΔI_c 信息。

机制 2：钩子创建

每个新钩子替代一定量的手动操作。设 ΔI_h 为每个新钩子的平均操作减少量。那么：

$$\left. \frac{dI}{dt} \right|_{\text{hooks}} \leq -\beta \cdot H(t)$$

其中 $\beta = \mathbb{E}[\Delta I_h]$ 是每个新钩子的预期输入减少量。

组合上限

由于两个机制独立运行：

$$L(t) = -\frac{dI}{dt} \leq \alpha \cdot C(t) + \beta \cdot H(t) = L_{\max}(t)$$

第 2 部分：下限推导

根据定理 1（收敛法则），我们知道系统收敛到零输入。这意味着学习速率的下限由正函数界定。

考虑钩子执行的预期输入减少量。在时间 t ，设：

$N(t)$ = number of hook execution opportunities per time unit

$S(t)$ = average success score of executable hooks

每次成功的钩子执行都会节省 $|A|$ 个本应手动完成的操作。每个机会的预期减少为：

$$\mathbb{E}[\Delta I] = S(t) \cdot \mathbb{E}[|A|]$$

因此：

$$L(t) = -\frac{dI}{dt} \geq N(t) \cdot S(t) \cdot \mathbb{E}[|A|]$$

方差惩罚

成功分数的高方差降低了有效学习率，因为资源浪费在表现不佳的钩子上。使用詹森不等式处理凹函数：

$$\mathbb{E}[\text{improvement}(S)] \geq \text{improvement}(\mathbb{E}[S]) - \delta \cdot \text{Var}(S)$$

这给出了方差调整后的下限：

$$L(t) \geq \gamma \cdot \frac{S(t)}{1 + \delta \cdot \sigma^2(t)} = L_{\min}(t)$$

其中 $\gamma = N(t) \cdot \mathbb{E}[|A|]$ 吸收了执行率和操作节省。

第 3 部分：界限紧致性

两个界限在可实现的意义上都是紧的：

- 上限实现：当每个修正和新钩子提供最大可能的输入减少时

- 下限实现：当成功分数高度可变且执行机会最少时

因此，这些界限准确地表征了学习速率。□

与定理的联系

这个推论直接建立在定理1（收敛法则）之上：

- 定理1确立了 $I(t) \rightarrow 0$ 作为 $t \rightarrow \infty$
- 这个推论量化了收敛发生的速度

在定理1证明渐近行为的地方，这个推论提供了对实际系统设计和性能预测有用的有限时间界限。

它还与以下内容相关：

• 定理5（学习法则）：修正频率 $C(t)$ 和钩子创建率 $H(t)$ 出现在上界中，因为它们驱动学习过程

• 定理2（最优法则）：下界依赖于成功评分，这反映了钩子选择的最优性

实际影响

1. 学习速率预测

系统设计师可以通过测量界限参数来估计学习速率：

$$\hat{L}(t) \approx \frac{L_{\min}(t) + L_{\max}(t)}{2}$$

随着时间的推移，跟踪 $C(t)$ 、 $H(t)$ 、 $S(t)$ 和 $\sigma^2(t)$ 以预测系统何时会达到各种输入减少的里程碑：

$$t_{50\%} \approx \frac{I_0/2}{\hat{L}}$$

其中 I_0 是初始输入水平， $t_{50\%}$ 是达到 50% 减少的时间。

2. 瓶颈识别

如果实际学习速率 $L(t)$ 接近上限 $L_{\max}(t)$ ，则系统正在尽可能快地学习，前提是有可用的修正和钩子。改善需要：

- 增加修正频率：更多用户反馈
- 增加钩子创建：更多自动化能力

如果 $L(t)$ 接近下限 $L_{\min}(t)$ ，则系统有可用的信息，但未有效利用。改善需要：

- 改善成功率：更好的钩子质量
- 减少方差：更一致的表现

3. 最优修正策略

自 $L_{\max} \propto \alpha \cdot C(t)$ 以来，有激励来最大化纠正。但纠正需要用户付出努力！最佳策略平衡：

$$\text{minimize: Total Effort} = \int_0^T [I(t) + \kappa \cdot C(t)] dt$$

其中 κ 是每次纠正的成本。这导致了最佳纠正率：

$$C^*(t) = \sqrt{\frac{\alpha}{\kappa} \cdot I(t)}$$

当输入较高时（早期学习阶段）要多纠正，当输入较低时（成熟系统）要少纠正。

4. 钩子创建优先级

自 $L_{\max} \propto \beta \cdot H(t)$ 以来，创建具有高 β （大行动节省）的钩子加速学习。优先考虑钩子创建：

- 频繁的任务（高执行率）
- 行动密集型任务（高 $|A|$ ）
- 没有钩子的任务（最大 ΔI_h ）

这创建了一个钩子创建价值指标：

$$V_{\text{create}} = \text{frequency} \times |A| \times (1 - \text{coverage})$$

5. 方差减少策略

下限惩罚 $\delta \cdot \sigma^2(t)$ 激励减少成功分数的方差：

- 剔除低效能者：移除 $S \ll \bar{S}$ 的钩子
- 集中学习：将修正直接指向接近决策边界的钩子
- 标准化上下文：在可能的情况下减少环境变异性

具有均匀良好钩子（低方差）的系统学习速度比质量混合的系统更快。

6. 性能保证

这些界限提供正式的 SLA 风格的保证：

"*在最低修正率 C_{\min} 和钩子质量 S_{\min} 的情况下，系统保证在时间内将输入减少到基线的 50%：*"

$$T_{50\%} \leq \frac{I_0/2}{L_{\min}(C_{\min}, S_{\min})}$$

这些保证建立用户信任，并使知识钩子系统的比较评估成为可能。

示例

示例 1：个人智能家居系统

初始状态： $I_0 = 150$ 每天手动操作

第一个月测量的参数：

$$C(t) = 3 \text{ corrections/day}, \quad H(t) = 0.5 \text{ hooks/day}$$

$$S(t) = 0.85, \quad \sigma^2(t) = 0.04$$

从校准中估算的常数：

$$\alpha = 2.5, \quad \beta = 8, \quad \gamma = 15, \quad \delta = 10$$

上限：

$$L_{\max} = 2.5 \times 3 + 8 \times 0.5 = 7.5 + 4 = 11.5 \text{ actions/day reduction}$$

下限：

$$L_{\min} = 15 \times \frac{0.85}{1 + 10 \times 0.04} = 15 \times \frac{0.85}{1.4} = 9.1 \text{ actions/day reduction}$$

预计减少50%的时间：

$$\hat{L} = \frac{11.5 + 9.1}{2} = 10.3 \text{ actions/day}$$

$$T_{50\%} \approx \frac{75}{10.3} \approx 7.3 \text{ days}$$

经过7-8天，用户应预期每日操作从150减少到约75。

示例2：企业工作流自动化

拥有100名用户的大型组织：

初始状态： $I_0 = 5000$ 每个用户每周手动操作

系统参数：

$$C(t) = 50 \text{ corrections/week (collective)}$$

$$H(t) = 5 \text{ hooks/week (shared library)}$$

$$S(t) = 0.75, \quad \sigma^2(t) = 0.09 \text{ (high variance due to diverse workflows)}$$

常量：

$$\alpha = 15, \quad \beta = 120, \quad \gamma = 200, \quad \delta = 5$$

上限：

$$L_{\max} = 15 \times 50 + 120 \times 5 = 750 + 600 = 1350 \text{ actions/week}$$

下限：

$$L_{\min} = 200 \times \frac{0.75}{1 + 5 \times 0.09} = 200 \times \frac{0.75}{1.45} \approx 103 \text{ actions/week}$$

注意到宽广的差距： $L_{\max}/L_{\min} \approx 13$ 。这表示：

- 系统具有很强的学习能力（高 C 和 H ）
- 但高方差（ $\sigma^2 = 0.09$ ）会造成不确定性

建议：专注于减少方差。如果方差降低到 $\sigma^2 = 0.04$ ：

$$L_{\min}^{\text{new}} = 200 \times \frac{0.75}{1 + 5 \times 0.04} = 125 \text{ actions/week}$$

仅通过减少方差，底线提高了 21%。

示例 3：比较两个系统

系统 A（快速反馈）：

$$C_A = 10, H_A = 1, S_A = 0.80, \sigma_A^2 = 0.05$$

$$L_{\max}^A = 2 \times 10 + 5 \times 1 = 25$$

$$L_{\min}^A = 10 \times \frac{0.80}{1 + 10 \times 0.05} = 5.3$$

系统 B（高质量）：

$$C_B = 3, H_B = 0.5, S_B = 0.95, \sigma_B^2 = 0.01$$

$$L_{\max}^B = 2 \times 3 + 5 \times 0.5 = 8.5$$

$$L_{\min}^B = 10 \times \frac{0.95}{1 + 10 \times 0.01} = 8.6$$

分析：

- 系统 A 的上限更高（更多的修正/钩子），但下限更低（质量问题）
- 系统 B 的边界更紧（可预测）且最低速率更高（可靠）
- 系统 B 的 $L_{\min} \approx L_{\max}$ 表明其运行接近最佳状态

系统 B 可能更可取：学习更可预测，且尽管修正较少，仍有更高的保证最低速率。

限制与扩展

1. 时变参数

这些边界假设参数 $C(t)$ 、 $H(t)$ 、 $S(t)$ 、 $\sigma^2(t)$ 是已知的。实际上，它们会演变：

- $C(t)$ 通常随着系统的改善而减少（需要修正的错误更少）
- $H(t)$ 在密集学习期间可能会激增，然后稳定
- $S(t)$ 通常会增加（学习提高了质量）

更复杂的分析明确跟踪这些动态。

2. 钩子的非独立性

证明假设钩子独立改善。实际上，钩子之间是相互作用的：

- 组合钩子依赖于组成钩子
- 钩子共享上下文特征
- 对一个钩子的改进可以使其他钩子受益

依赖关系可以通过迁移学习效应加速学习，超越上限。

3. 上下文分布变化

边界假设上下文分布是静态的。如果用户行为发生显著变化：

$$P(C|t) \neq P(C|t + \Delta t)$$

系统可能需要重新学习，暂时将学习速率降低到 L_{\min} 之下。

4. 计算约束

真实系统的计算资源是有限的。如果钩子评估或学习算法无法跟上修正/创建速率，实际学习速率可能会低于理论下限。

5. 多用户系统

在共享钩子系统中，通过知识共享，集体学习可以超越个体界限。这里的界限适用于单用户系统；多用户扩展需要考虑知识转移。

结论

学习速率界限将定理 1 的定性收敛保证转化为对系统设计和用户期望管理有用的定量预测。通过建立可计算的学习速率上下界，这一推论使得：

- 性能预测：估计达到自动化里程碑所需的时间

- 瓶颈诊断：识别学习是否受到信息（修正/钩子）或质量（成功分数/方差）的限制
- 优化策略：选择投资资源以实现最大学习加速的地方
- 比较评估：严格比较不同的知识钩子系统

这些界限揭示了基本的权衡：快速学习需要高频率的纠正（用户努力）或高创建钩子的速度（系统复杂性），但必须与该投资的成本相平衡。下界的方差惩罚强调了一致的质量与平均质量同样重要——一个具有均匀良好钩子的系统比一个性能混合的系统学习得更快。

最重要的是，这些界限不仅仅是理论构造。通过跟踪纠正、钩子创建和成功分数，它们可以在真实系统中进行测量。这种可测量性使得推论变得可操作：系统操作员可以监控这些指标，计算当前界限，并根据数据做出关于改进努力的决策。学习速率界限弥合了数学证明与实际工程之间的差距。

4.1.6 推论 6：成功分数单调性

声明：对于任何知识钩子，其成功分数在学习过程中随时间单调非减——纠正要么提高成功分数，要么保持不变，但在期望上从不降低。

正式声明

设 $S_i(t)$ 表示钩子 KH_i 在时间 t 的成功分数。在时间 t 发生的任何纠正事件后，涉及 KH_i ，在时间 $t + \Delta t$ 的期望成功分数满足：

$$\mathbb{E}[S_i(t + \Delta t) \mid S_i(t)] \geq S_i(t)$$

更一般地，对于任何时间 $t_2 > t_1$ ：

$$\mathbb{E}[S_i(t_2)] \geq S_i(t_1)$$

对于系统范围内的平均成功分数 $\bar{S}(t) = \frac{1}{|\mathcal{H}|} \sum_i S_i(t)$:

$$\frac{d\bar{S}}{dt} \geq 0$$

每当发生纠正时都有严格的不等式： $\frac{d\bar{S}}{dt} > 0$ 当 $C(t) > 0$ 。

直观解释

成功分数单调性捕捉了学习的一个基本属性：你无法通过纠正来忘记。每当用户纠正一个钩子时，系统获得的信息要么：

- 改进钩子：细化条件、调整动作或更新参数以使钩子更可靠
- 钩子保持不变：如果修正没有提供新信息或钩子在该上下文中已经是最佳的

但修正从来不会使钩子变得更糟。这看起来可能显而易见，但这是一个关键保证：用户可以放心地提供修正，知道它们只会有帮助，而不会造成伤害。这就像一个棘轮机制——只朝一个方向前进。

想象一下教某人做饭：每一条反馈（“加更多盐，” “再煮久一点”）都会让他们变得更好或保持他们的技能，但不会让他们

忘记已经知道的东西。成功分数单调增加，朝着精通的方向前进。

证明

第一部分：单一钩子的单调性

考虑钩子 KH_i 在时间 $S_i(t)$ 的成功分数 t 。
成功分数表示正确执行的概率：

$$S_i(t) = P(\text{hook succeeds} \mid \text{conditions satisfied, knowledge at time } t)$$

当在时间 t 发生修正时，它提供了信息 I 来更新钩子。根据定理 5（学习法则），这些信息被纳入以改善钩子。更新后的成功分数为：

$$S_i(t + \Delta t) = P(\text{success} \mid \text{knowledge at } t, \text{new info } I)$$

关键见解：新信息 I 只能提供帮助，因为学习算法实现了贝叶斯更新或具有正学习率的梯度下降。让我们证明这一点对于贝叶斯更新：

设 θ 表示钩子的参数（条件、动作、权重）。修正后的后验为：

$$P(\theta \mid I, \text{past data}) = \frac{P(I \mid \theta)P(\theta \mid \text{past data})}{P(I)}$$

新的成功概率为：

$$S_i(t + \Delta t) = \int P(\text{success} \mid \theta)P(\theta \mid I, \text{past data}) d\theta$$

由于贝叶斯更新将概率质量转移到更好地解释数据的参数值上，并且只有在钩子失败或需要调整时才会发生修正，因此更新将 θ 移动到成功概率更高的值。因此：

$$\mathbb{E}[S_i(t + \Delta t)] \geq S_i(t)$$

第二部分：无负面学习

我们必须验证修正不会导致"灾难性遗忘"，即学习新信息会削弱旧知识。知识钩学习算法通过以下方式防止这种情况：

机制 1：加性更新

修正添加新的条件或细化现有条件，而不是替换它们。如果一个钩子在上下文 C_1 中有效，而在上下文 C_2 中发生修正，则更新为：

$$R_{\text{new}} = R_{\text{old}} \vee (C_2 \wedge \text{correction})$$

这在 C_1 中保持成功，同时改善 C_2 中的表现。

机制 2：保守更新

学习率被保守地选择，以确保更新不会超调：

$$\eta < \eta_{\text{max}} = \frac{2}{\lambda_{\text{max}}(H)}$$

其中 H 是损失函数的海森矩阵。这保证了收敛而不发生振荡。

机制 3：正则化

学习包括一个正则化项，该项惩罚当前参数的大幅变化：

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{correction}} + \lambda \|\theta - \theta_0\|^2$$

这确保了更新是可测量的，并且不会大幅改变行为。

第3部分：系统范围的单调性

对于所有钩子的平均成功分数：

$$\bar{S}(t) = \frac{1}{|\mathcal{H}(t)|} \sum_{i=1}^{|\mathcal{H}(t)|} S_i(t)$$

取时间导数：

$$\frac{d\bar{S}}{dt} = \frac{1}{|\mathcal{H}|} \sum_i \frac{dS_i}{dt} + \frac{d|\mathcal{H}|/dt}{|\mathcal{H}|^2} \left(\sum_i S_i \right)$$

第一项是非负的，依据第1部分（单个钩子改善）。对于第二项，当以速率 $H(t) = d|\mathcal{H}|/dt > 0$ 创建新钩子时：

- 如果新钩子的初始成功分数为 $S_{\text{init}} < \bar{S}$ ，则该项为负

- 但新钩子通过学习迅速改善，因此瞬时下降被改善所弥补

实际上，新钩子通常以高初始成功率创建（从模板复制，或由成功的钩子组合而成），因此 $S_{\text{init}} \approx \bar{S}$ 和第二项接近于零。

因此，整体系统成功分数是单调非减的。 \square

与定理的联系

这个推论直接来自定理5（学习法则）：

- 定理5指出，通过引入新信息，修正可以改善钩子。
- 这个推论量化了“改善”的含义——成功分数增加。

它还支持定理1（收敛法则）：

- 收敛到零输入要求钩子变得越来越可靠。
- 单调递增的成功分数为这种可靠性增长提供了机制。

并与定理2（最优法则）相连接：

- 随着成功分数的提高，最优钩子相对于手动替代方案变得更加最优。
- 自动化和手动方法之间的差距随着时间的推移而扩大。

实际影响

1. 用户对修正的信心

单调性保证用户可以自由提供修正，而不必担心会使情况变得更糟。这在心理上是重要的：

- 用户不需要对纠正措施“小心”
- 任何反馈都有帮助；更多的反馈帮助更多
- 系统对“嘈杂”的纠正措施具有鲁棒性

这鼓励自由使用纠正，加速学习。

2. 进度跟踪

单调性使得有意义的进度可视化成为可能。随着时间的推移绘制成功分数，以向用户展示系统如何改善：

$$\text{Progress} = \frac{\bar{S}(t) - \bar{S}(0)}{1 - \bar{S}(0)}$$

该指标的范围从 0（没有改善）到 1（完美可靠），并且保证不下降。

3. 质量保证

如果成功分数下降，这表明学习系统中存在错误：

if $S_i(t_2) < S_i(t_1)$ for $t_2 > t_1 \implies \text{system malfunction}$

监控成功分数作为健康检查。分数下降会触发警报和调查。

4. 学习平台检测

当成功分数停止增加时，系统已达到学习平台：

$$\frac{d\bar{S}}{dt} \approx 0 \implies \text{plateau reached}$$

这表明有以下情况：

- 成功：钩子几乎完美（ $\bar{S} \approx 1$ ），无法再改进
- 信息瓶颈：需要更多样化的纠正或上下文
- 容量限制：钩子已达到其表现极限，需要更具表现力的形式

5. 比较系统评估

通过成功分数轨迹比较学习系统：

$$\text{System A better than B} \iff \forall t : S_A(t) \geq S_B(t)$$

由于成功分数是单调的，领先并保持优势的系统显然更好。

6. 最小可行质量阈值

设置钩子激活的最低成功分数阈值：

$$\text{enable}(KH_i) \iff S_i \geq S_{\min}$$

由于成功分数只会增加，一旦钩子超过阈值，它将保持启用。这形成了一个单向阀门：钩子从“学习”转移到“生产”，但永远不会回退。

7. 转移学习验证

在用户或上下文之间转移钩子时，单调性提供了一个合理性检查：

$$\text{if } S_{\text{after_transfer}} < S_{\text{before_transfer}} \implies \text{poor context match}$$

成功分数不应仅因移动到新环境而减少（尽管它们可能不会增加得那么快）。

示例

示例 1：电子邮件分类钩子演变

一个自动归档促销电子邮件的钩子。30天内的成功分数演变：

$$S(0) = 0.65 \quad (\text{initial template})$$

第3天：用户纠正了5封被错误归档的电子邮件（来自商店的重要信息）

$$S(3) = 0.72 \quad (\text{learned to exclude order confirmations})$$

第7天：用户纠正了3个案例（他们想保留的通讯）

$$S(7) = 0.81 \quad (\text{added whitelist for specific senders})$$

第15天：用户纠正了1个边缘案例（包含重要优惠券的促销电子邮件）

$$S(15) = 0.89 \quad (\text{refined keyword detection})$$

第30天：过去一周没有更正

$$S(30) = 0.94 \quad (\text{continued learning from observation})$$

轨迹显示严格单调性：
 $0.65 \rightarrow 0.72 \rightarrow 0.81 \rightarrow 0.89 \rightarrow 0.94$
。每个更正事件都带来改进。系统从未变得更糟。

示例2：智能恒温器挂钩

挂钩根据时间和占用情况调整温度。成功评分演变：

第1周：初始基于规则的系统

$$\bar{S}(1) = 0.70 \quad (10 \text{ manual overrides per week})$$

第2周：学习每日时间表模式

$$\bar{S}(2) = 0.78 \quad (7 \text{ overrides})$$

第4周：纳入天气预报

$$\bar{S}(4) = 0.85 \quad (4 \text{ overrides})$$

第8周：学习周末与工作日的差异

$$\bar{S}(8) = 0.92 \quad (2 \text{ overrides})$$

第12周：学习季节偏好

$$\bar{S}(12) = 0.96 \quad (1 \text{ override per 2 weeks})$$

再次，严格单调的改进。挂钩变得越来越可靠，随着时间的推移需要更少的更正。

示例 3：系统范围学习曲线

具有 50 个钩子的个人自动化系统。平均成功分数轨迹：

$$\bar{S}(\text{day } 1) = 0.60, \quad \bar{S}(\text{week } 1) = 0.68$$

$$\bar{S}(\text{month } 1) = 0.78, \quad \bar{S}(\text{month } 3) = 0.87$$

$$\bar{S}(\text{month } 6) = 0.93, \quad \bar{S}(\text{month } 12) = 0.97$$

改善率（使用有限差分）：

$$\left. \frac{\Delta \bar{S}}{\Delta t} \right|_{\text{month } 1} = \frac{0.78 - 0.60}{30} = 0.006/\text{day}$$

$$\left. \frac{\Delta \bar{S}}{\Delta t} \right|_{\text{month } 6} = \frac{0.93 - 0.87}{90} = 0.0007/\text{day}$$

请注意，虽然 \bar{S} 是单调递增的，但随着成功分数接近 1（收益递减），增长速度减缓。这是预期的：早期学习带来大的改善，后期的细化则更为渐进。

限制和注意事项

1. 随机波动

单调性保证是期望中的：
 $\mathbb{E}[S(t+1)] \geq S(t)$ 。个别测量可能因以下原因而波动：

- 成功率估计中的小样本量
- 上下文分布变化
- 环境噪声

短期下降并不违反定理；只有持续的趋势才重要。

2. 上下文漂移

如果用户行为或环境发生显著变化，成功评分可能会下降：

$$\text{if } P(C|t_2) \neq P(C|t_1) \implies S(t_2) \text{ may be } < S(t_1)$$

这并不违反单调性，因为定理假设上下文分布是静态的。当上下文漂移时，在新上下文中测量的成功评分可能较低，但在原始上下文中的评分保持不变。

3. 极端情况下的灾难性遗忘

如果学习系统使用有限内存并经历严重的概念漂移，旧知识可能会被覆盖：

- 针对冬季上下文学习的钩子在夏季可能会退化
- 针对旧家庭设置训练的钩子在搬家后可能会失效

适当的系统设计通过上下文感知学习和选择性遗忘来减轻这一问题，只移除真正过时的知识。

4. 对抗性修正

定理假设修正是诚实的尝试以改善系统。对抗性或矛盾的修正理论上可能会降低成功评分：

- 用户提供不一致的反馈
- 恶意行为者试图破坏系统

强大的学习算法与异常检测可以防止这种情况。

5. 测量偏差

如果测量方法发生变化，成功分数可能会看起来下降：

- 采用更严格的成功标准
- 评估环境变得更加具有挑战性
- 用户期望上升（"移动的目标"）

真正的单调性指的是固定的成功标准，而不是不断演变的标准。

6. 计算近似

真实的实现使用近似学习算法（截断梯度、有限精度、近似推理）。这些近似有时会产生小的负更新。调优良好的系统使这些影响微不足道。

结论

成功分数单调性为知识钩系统提供了基础保证：学习使事物变得更好，而不是更糟。这一看似简单的属性具有深远的影响：

- 信任：用户可以自信地提供更正，因为他们知道这会有所帮助
- 可预测性：系统质量保证会改善或稳定，绝不会退步
- 可测量性：成功分数轨迹提供了学习进展的客观指标
- 鲁棒性：系统对不完美的更正具有弹性；即使是嘈杂的反馈也有帮助

推论将学习从一个不透明的过程转变为一个透明、可测量和可靠的机制。用户可以实时观察他们的系统改善，并且有数学上的确定性，进展是真实和永久的。成功分数成为量化自动化质量的通用货币——可以在不同的系统、上下文和时间段之间进行比较。

最重要的是，单调性使复合学习效果成为可能。每次更正都建立在之前的基础上，创造出一个不断改善的基础。与可能导致性能退步的系统不同，知识钩系统不可逆地积累知识。这个属性对于长期自主性至关重要：一个可能自发变得不可靠的系统永远不会被信任来执行重要任务。单调性保证了通过学习获得的信任会随着时间的推移而维持和增强。

4.1.7 上下文覆盖完整性

上下文覆盖完整性推论建立了一个知识钩系统，在足够的时间和用户互动下，最终会发展出覆盖所有经常遇到的上下文的钩子。这意味着对于用户经常经历的任何情况，系统将学习自动处理。没有重要的上下文会被永久遗漏——系统会逐渐变得更加完整，直到能够处理用户面临的几乎任何场景。

推论声明

推论 7 (上下文覆盖完整性)：让 \mathcal{C} 表示所有可能上下文的空间，让 $P(C)$ 是用户遇到的上下文的概率分布。让 $\mathcal{H}(t)$ 是系统在时间 t 的钩子集合，并定义时间 t 的覆盖集合为：

$$\text{Cov}(t) = \{C \in \mathcal{C} : \exists KH \in \mathcal{H}(t) \text{ such that } R(KH, C) = \text{true}\}$$

也就是说， $\text{Cov}(t)$ 包含至少有一个钩子条件满足的所有上下文。那么，对于任何阈值 $\epsilon > 0$ ，存在一个时间 T ，使得：

$$\forall t > T : P(\text{Cov}(t)) \geq 1 - \epsilon$$

换句话说，系统最终覆盖了 $(1 - \epsilon)$ 分数的概率质量。随着 $t \rightarrow \infty$ ，覆盖接近完整性：

$$\lim_{t \rightarrow \infty} P(\text{Cov}(t)) = 1$$

更强的形式（均匀覆盖）：对于任何上下文 C ，如果 $P(C) > 0$ 至少出现 n_{\min} 次，系统将以接近 1 的概率创建一个覆盖 C 或 C 的超集的钩子，随着 $t \rightarrow \infty$ 。

直观解释

上下文覆盖完整性表示：你经常面临的每种情况最终都会被自动处理。系统不仅优化现有的钩子——它还主动发现覆盖中的空白并填补它们。

想象一下学习驾驶。起初，你需要对每种情况给出明确的指示：“在这里左转”，“在这个标志处刹车”，“并入那条车道。”但是随着经验的积累，你会为所有常见情况发展出自动反应。最终，你可以在没有意识思考的情况下驾驶大部分路线——你的行为能力在你遇到的上下文中实现了覆盖完整性。知识钩子系统的工作方式相同：通过重复的接触和学习，它们为所有频繁的上下文发展出自动处理程序。

这很重要：用户不需要手动列举每种自动化可以帮助的情况。系统通过使用自然发现这些情况。如果你经常在离家前查看天气，系统最终会为该上下文创建一个钩子。如果你经常在日落时打开灯，钩子将会出现。覆盖会自动增长，以匹配你的实际行为模式。

证明

第 1 部分：覆盖增长机制

新钩子通过两种机制创建：

机制 A（手动创建）：用户明确为他们识别为重复的上下文创建钩子。

机制 B（学习创造）：根据定理 5（学习法则），当用户在类似的上下文中反复执行相同的操作时，系统会检测到模式并提出一个新的钩子。如果被接受，这个钩子将覆盖一个之前未被覆盖的上下文。

设 $\lambda(C)$ 为覆盖上下文 C 的新钩子创建速率。对于经常遇到的上下文：

$$\lambda(C) \geq \alpha \cdot P(C)$$

其中 $\alpha > 0$ 是学习速率参数。更常见的上下文会导致更快的钩子创建。

第 2 部分：概率质量积累

设 $p(t) = P(\text{Cov}(t))$ 为时间 t 时覆盖的概率质量。覆盖变化的速率为：

$$\frac{dp}{dt} = \int_{C \notin \text{Cov}(t)} \lambda(C) dP(C)$$

这个积分表示新创建的钩子覆盖新概率质量的速率。由于 $\lambda(C) \geq \alpha \cdot P(C)$ 对于未覆盖的上下文：

$$\frac{dp}{dt} \geq \alpha \int_{C \notin \text{Cov}(t)} P(C) dP(C) = \alpha \cdot (1 - p(t))$$

这是一个具有解的微分不等式：

$$p(t) \geq 1 - (1 - p(0))e^{-\alpha t}$$

第 3 部分：渐近完备性

当 $t \rightarrow \infty$ 取极限时：

$$\lim_{t \rightarrow \infty} p(t) \geq \lim_{t \rightarrow \infty} [1 - (1 - p(0))e^{-\alpha t}] = 1$$

由于 $p(t) \leq 1$ 的定义，我们有：

$$\lim_{t \rightarrow \infty} P(\text{Cov}(t)) = 1$$

第 4 部分：收敛速率

对于任何目标覆盖率 $1 - \epsilon$ ，求解所需时间：

$$1 - (1 - p(0))e^{-\alpha T} = 1 - \epsilon$$

$$e^{-\alpha T} = \frac{\epsilon}{1 - p(0)}$$

$$T = \frac{1}{\alpha} \ln \left(\frac{1 - p(0)}{\epsilon} \right)$$

这给出了实现 $(1 - \epsilon)$ -覆盖所需时间的明确界限。



与定理的联系

这个推论建立在多个定理之上：

根据定理 1（收敛法则）：当系统收敛到零输入时，它必须自动覆盖越来越多的上下文。覆盖的完整性是时间收敛的空间/上下文表现。

根据定理 4（划分法则）：将钩子空间划分为等价类为覆盖分析提供了结构。不同的钩子覆盖不同的上下文区域，它们共同划分了上下文空间。

根据定理 5（学习法则）：覆盖增长的机制。学习创造新的钩子以填补覆盖中的空白，驱动因素是未覆盖上下文中的观察模式。

实际影响

1. 不需要手动枚举

用户不需要预见并预先编程每一个自动化可以帮助的情况。系统有机地发现这些情况：

- 正常生活
- 系统观察到重复的模式
- 钩子会自动出现在常见的上下文中

这与传统的自动化根本不同，传统自动化需要对每种场景进行明确编程。

2. 系统健康的覆盖度指标

通过跟踪覆盖度随时间的变化来衡量系统的成熟度：

$$\text{Maturity}(t) = P(\text{Cov}(t))$$

一个成熟的系统具有 $\text{Maturity} \approx 1$ ，这意味着几乎所有的上下文都被覆盖。这提供了一个定量指标，用于衡量用户的生活变得 "多么自动化"。

3. 差距检测和优先级排序

识别高影响力的覆盖差距：

$$\text{Gap_Value}(C) = P(C) \cdot \mathbb{1}[C \notin \text{Cov}(t)] \cdot |A(C)|$$

这通过（频率）×（操作次数）来评分未覆盖的上下文。高分差距代表了显著的自动化增益机会。系统可以主动建议："你在上下文 C 中经常做 X，但没有相应的钩子。我应该创建一个吗？"

4. 覆盖可视化

为用户提供显示覆盖情况的地图：

- 绿色区域：覆盖良好的上下文，具有多个可靠的钩子
- 黄色区域：部分覆盖的上下文（成功率低）
- 红色区域：未覆盖的上下文，需要重复的手动操作

这帮助用户理解自动化工作在哪些地方，以及还有哪些机会。

5. 收敛时间估计

使用证明第4部分中的公式，估计达到目标覆盖所需的时间：

$$T_{95\%} = \frac{1}{\alpha} \ln(20)$$

对于典型的学习率 $\alpha = 0.1$ （每单位时间添加10%的未覆盖量）：

$$T_{95\%} = 10 \ln(20) \approx 30 \text{ time units}$$

如果时间单位是一天，95%的覆盖率大约需要一个月。这设定了现实的用户期望。

6. 转移学习加速

新用户可以通过导入来自相似用户的钩子来加速覆盖：

$$p(0) = p_{\text{imported}} > 0$$

这 "引导" 覆盖，从非零基线开始。如果 $p_{\text{imported}} = 0.6$ ，收敛时间显著减少：

$$T = \frac{1}{\alpha} \ln \left(\frac{0.4}{\epsilon} \right) < \frac{1}{\alpha} \ln \left(\frac{1}{\epsilon} \right)$$

示例

示例 1：智能家居覆盖演变

跟踪新智能家居用户的 6 个月覆盖情况：

第 1 周： $p(1) = 0.15$ （3 个手动创建的钩子：
早上灯光、晚上恒温器、就寝灯光）

第 4 周： $p(4) = 0.42$ （系统学习了 7 个新钩子：
闹钟响时咖啡、做饭时音乐、离开时关灯等）

第 12 周： $p(12) = 0.73$ （总共 17 个钩子，
覆盖大多数日常活动）

第 24 周： $p(24) = 0.91$ （28 个钩子，仅剩不
寻常/罕见的上下文未覆盖）

完整性的指数方法是显而易见的：早期快速增长，然后随着
仅剩罕见上下文而减缓。

示例 2：行动中的差距检测

系统识别出一个高价值的覆盖差距：

上下文：[星期六早晨，在家，没有日历事件]

观察到的模式：用户手动启动咖啡机，打开百叶窗，检查天
气，播放音乐（4个动作）

频率：52个星期六中的48个（92%）

差距分数： $0.92 \times 4 = 3.68$

系统建议："我注意到你每个星期六早晨都会做这4件事。创
建一个'懒惰星期六'钩子吗？"

创建后：

$$\bullet \quad \text{覆盖 率 增 加} : \\ p(t) \rightarrow p(t) + 0.92/52 \approx p(t) + 0.018$$

- 该高频上下文的差距消除

示例3：覆盖可视化

覆盖热图可能显示：

早晨（6-9点）：

- 工作日：95% 覆盖（绿色）- 稳定的早晨例行程序
- 周末：60% 覆盖（黄色）- 一些变化，部分覆盖

工作时间（9点-5点）：

- 办公室地点：85% 覆盖（绿色）
- 远程工作：40% 覆盖（红色）- 较新的模式，少量钩子

晚上（5-10点）：

- 烹饪：70% 覆盖（黄色）
- 娱乐：50% 覆盖（黄色/红色）

夜晚（10点-6点）：

- 睡前例行程序：90% 覆盖（绿色）

此可视化立即揭示了自动化机会的存在（远程工作、娱乐）。

限制和注意事项

1. 长尾分布

如果上下文分布有重尾（许多稀有上下文），实现 100% 覆盖可能不切实际：

$$P(C) \sim C^{-\gamma} \text{ with } \gamma \leq 1$$

最后几个百分点的覆盖可能需要指数级的钩子。实际系统的目标是 90-95% 的覆盖，接受真正稀有的上下文仍需手动处理。

2. 非平稳分布

该证明假设上下文分布是平稳的 $P(C)$ 。如果用户行为发生显著变化：

- 新工作 → 新办公室上下文
- 移动到新家 → 新的物理空间
- 生活方式改变 → 不同的例行程序

覆盖率可能会暂时下降，因为旧的钩子变得无关紧要，而新的上下文出现。系统必须重新学习，但通常由于从相似上下文的迁移而更快。

3. 上下文粒度

Coverage depends on how finely contexts are defined. At coarse granularity ("morning" vs "evening"), coverage appears high. At fine granularity ("Tuesday 7:23am, raining, hungry"), coverage is lower. The corollary doesn't specify granularity—this is a system design choice.

4. 用户接受门

该证明假设当系统提出新的钩子时，用户会接受。如果用户频繁拒绝提议：

$$\lambda_{\text{effective}}(C) = \lambda(C) \cdot P(\text{accept})$$

覆盖率增长会成比例减慢。高质量的提议（准确的模式检测，良好的用户体验）对于实际的完整性至关重要。

5. 无限上下文空间

如果 C 是无限维或连续的，“覆盖所有上下文”是未定义的。推论适用于离散或有限维的上下文空间，或必须解释为覆盖 $P(C)$ 的支持（具有非零概率质量的上下文）。

6. 计算约束

创建和评估钩子有计算成本。如果系统变得过大（数千个钩子），评估时间会增长：

$$t_{\text{eval}} = O(|\mathcal{H}| \cdot |\text{features}|)$$

这可能会限制实际覆盖范围——系统可能会优先考虑高频上下文，而忽略非常罕见的上下文，以保持评估的可行性。

结论

上下文覆盖完整性提供了强有力的保证：没有常见的情况会被永久地留给手动处理。系统主动发现其覆盖范围中的空白，并通过学习填补这些空白。这将用户体验从“编程每个场景”转变为“在自动化出现时自然生活”。

这一推论有几个深远的影响：

- 有限的学习时间：达到高覆盖率有一个明确的时间表（通常是几周到几个月）
- 可量化的进展：覆盖率可以被测量和跟踪，提供透明度

- 战略性填补空白：系统可以识别和优先考虑影响最大的未覆盖上下文

- 知识转移：用户之间共享钩子加速了新用户的覆盖率

最重要的是，覆盖完整性意味着零输入理想（定理1）不仅在理论上可实现，而且在实践中可实现。随着覆盖率接近1，需要手动干预的上下文比例接近0。系统不仅优化现有的自动化——它还积极扩展自动化的边界，直到几乎所有常规事务都能自动处理。

这是知识钩子系统能够真正“学习成为你”的主数学基础——不仅仅是模仿一些预编程的行为，而是全面覆盖你实际的行为领域。覆盖完整性是将一个有用工具转变为你真正延伸的能力，使其能够在你常见的任何情况下适当地行动。

4.1.8 权重剪枝收敛

权重剪枝收敛推论建立了一个知识钩系统，通过系统性地移除低效钩子，收敛到一个仅包含高质量钩子的稳定最小集合。这防止了系统无限期地积累过时、冗余或表现不佳的钩子。剪枝过程既是必要的（移除影响性能的钩子），也是收敛的（最终达到一个稳定的配置，在该配置下不再有利于进一步的剪枝）。

推论声明

推论 8（权重剪枝收敛）：设 $\mathcal{H}(t)$ 表示时间 t 的钩子集合，并定义一个剪枝操作 Π ，该操作移除成功分数低于阈值 S_{\min} 的钩子：

$$\Pi(\mathcal{H}) = \{KH \in \mathcal{H} : S(KH) \geq S_{\min}\}$$

在时间 t_1, t_2, \dots 定期进行剪枝，同时在剪枝事件之间继续学习。然后：

a) 有限收敛：存在一个有限时间 T ，在此之后不再进行进一步的剪枝：

$$\exists T : \forall t > T, \quad \Pi(\mathcal{H}(t)) = \mathcal{H}(t)$$

也就是说，所有剩余的钩子成功分数都高于阈值。

b) 单调减少：钩子的数量通过剪枝事件单调减少，直到收敛：

$$|\mathcal{H}(t_{i+1})| \leq |\mathcal{H}(t_i)|$$

c) 质量下限：收敛后，所有钩子满足质量标准：

$$\forall KH \in \mathcal{H}(\infty) : S(KH) \geq S_{\min}$$

d) 性能保证：剩余钩子的平均成功分数有下限：

$$\bar{S}(\infty) \geq S_{\min} + \frac{\alpha}{|\mathcal{H}(\infty)|}$$

其中 $\alpha > 0$ 反映了超出最低阈值的学习改进。

直观解释

权重剪枝收敛表示：糟糕的钩子最终会被移除，而移除过程最终会停止。系统不会无限增长积累的杂物，也不会添加和移除钩子之间永远振荡。相反，它收敛到一个精简的高质量集合。

把它想象成维护一个工具箱。随着时间的推移，你会获得许多工具，但有些工具可能不可靠或多余。定期清理工具箱，丢

弃那些你从不使用或经常失效的工具。最终，你会得到一套稳定的、你实际需要的可靠工具。你停止添加和移除工具，因为剩下的正是有效的工具。知识钩系统通过修剪的方式表现得类似。

这很重要：如果不进行修剪，系统会累积"死重"—过去创建的钩子不再有效或被更好的替代品取代。这些死重会减慢钩子的评估（需要检查更多的钩子），甚至可能损害性能（低质量的钩子错误触发）。修剪确保系统保持精简和高效。

证明

第 1 部分：成功分数单调性属性

根据推论 6（成功分数单调性），对于系统中保留的任何钩子：

$$\frac{dS(KH)}{dt} \geq 0$$

成功分数在学习过程中是非递减的。这意味着：

- 一旦钩子达到 $S \geq S_{\min}$ ，它将永远不会低于该阈值
- 低于阈值的钩子只能改善或保持不变

第 2 部分：分为三类

在任何时候 t ，将钩子分为三组：

安全钩 S ： $S(KH) \geq S_{\min} + \delta$ 以某个余量 $\delta > 0$

$$\mathcal{S}(t) = \{KH \in \mathcal{H}(t) : S(KH) \geq S_{\min} + \delta\}$$

边 际 钩 子 \mathcal{M} :

$$S_{\min} \leq S(KH) < S_{\min} + \delta$$

$$\mathcal{M}(t) = \{KH \in \mathcal{H}(t) : S_{\min} \leq S(KH) < S_{\min} + \delta\}$$

可修剪钩子 \mathcal{P} : $S(KH) < S_{\min}$

$$\mathcal{P}(t) = \{KH \in \mathcal{H}(t) : S(KH) < S_{\min}\}$$

第 3 部分：类别之间的流动

由于成功分数的单调性：

- 钩子可以移动 $\mathcal{P} \rightarrow \mathcal{M} \rightarrow \mathcal{S}$ (改善)
- 钩子从不向后移动 ($\mathcal{S} \nrightarrow \mathcal{M}$ 或 $\mathcal{M} \nrightarrow \mathcal{P}$)
- \mathcal{P} 中的钩子通过修剪被移除

这创造了单向流动：钩子要么朝 \mathcal{S} 改善，要么被修剪。

第 4 部分：收敛论证

考虑随着时间的推移会发生什么：

对于 \mathcal{P} 中的钩子：这些将在下一个修剪事件中被移除。一旦移除，它们将不再返回（假设没有新的低初始分数的钩子被创建）。

对于 \mathcal{M} 中的钩子：根据推论 6，它们的成功分数会增加。最终它们要么：

- 穿越到 \mathcal{S} （永久安全），或者
- 稳定在 S_{\min} 之上（保持边缘但不可修剪）

对于 \mathcal{S} 中的钩子：由于单调性，这些将永远保持安全。

设 $n_{\mathcal{P}}(t)$ 为时间 t 时可修剪钩子的数量。在每次修剪事件后：

$$n_{\mathcal{P}}(t_{i+1}) \leq n_{\mathcal{P}}(t_i)$$

由于 \mathcal{P} 中的钩子被移除，并且钩子永远不会回到 \mathcal{P} （单调性），我们有一个严格递减的序列。由于 $n_{\mathcal{P}} \geq 0$ ，这个序列必须在有限时间内达到 0：

$$\exists T : n_{\mathcal{P}}(T) = 0$$

在这一点之后，所有 $t > T$ 的 $\mathcal{P}(t) = \emptyset$ ，意味着没有钩子是可修剪的，系统已经收敛。□

与定理的联系

这个推论主要基于推论 6（成功分数单调性）：

根据推论 6：单调的成功分数确保修剪是单向操作。穿越质量阈值的钩子永远不会回落到阈值以下，确保收敛。

来自定理5（学习法则）：边际钩子向安全区改善的机制。如果没有学习，钩子将保持在其初始质量水平，修剪可能会去除过多。

来自定理1（收敛法则）：整体收敛到零输入不仅需要添加好的钩子，还需要去除坏的钩子。修剪是学习的“正”钩子创建的“负”补充。

实际影响

1. 自动系统维护

无需手动干预来清理钩子集。系统自我维护：

- 定期评估所有钩子
- 移除低于阈值的钩子
- 处理自动收敛

用户无需手动删除过时的钩子——系统会处理。

2. 有限的系统规模

钩子的数量受到能够达到 $S \geq S_{\min}$ 的数量的限制：

$$|\mathcal{H}(\infty)| \leq |\{\text{all possible hooks with } S \geq S_{\min}\}|$$

这防止了无限增长。系统大小稳定在由以下因素决定的水平：

- 用户行为的复杂性（更复杂 → 需要更多的钩子）
- 质量阈值 S_{\min} （更高的阈值 → 剩余的钩子更少）

3. 性能优化

修剪以多种方式提高系统性能：

更快的评估：需要检查的钩子更少 → $O(|\mathcal{H}|)$ 变得更小

$$t_{\text{eval after pruning}} = \frac{|\mathcal{H}(\infty)|}{|\mathcal{H}(0)|} \times t_{\text{eval initial}}$$

更少的错误：错误触发的低质量钩子被移除

更好的用户体验：用户只看到可靠的钩子，增加信任

4. 阈值选择

选择 S_{\min} 会产生权衡：

低阈值（ $S_{\min} = 0.5$ ）：保留更多的钩子，覆盖面更广，但仍有一些中等质量的钩子

高阈值（ $S_{\min} = 0.9$ ）：仅保留优秀的钩子，但一些有用的上下文可能缺乏覆盖

推荐： $S_{\min} \in [0.7, 0.8]$ 在大多数应用中平衡了质量和覆盖。

5. 修剪频率

多久进行一次修剪？推论建议：

$$\text{Pruning interval} \approx \frac{1}{\alpha} \ln(2)$$

其中 α 是学习率。这给钩子提供了改进的时间，避免被评判。修剪过于频繁会在钩子学习之前就将其移除；修剪过于不频繁则会留下过长时间的无用负担。

典型的时间表：每周或每两周对活跃系统进行修剪。

6. 新钩子的宽限期

新钩子开始时成功分数较低 ($S_0 \approx 0.5$)。为了防止立即修剪，实施宽限期：

Prune if: $S(KH) < S_{\min} \wedge \text{age}(KH) > t_{\text{grace}}$

这给新钩子有时间学习，然后再评估是否移除。典型的宽限期：1-2周。

7. 存档与删除

与其永久删除修剪的钩子，不如将它们存档：

- 如果用户行为改变，启用恢复
- 保留学习历史
- 如果上下文重现，允许复活

存档的钩子不计入 $|\mathcal{H}|$ 的评估，但可以在模式重新出现时恢复。

示例

示例 1：智能家居钩子的演变与修剪

用户开始时有 20 个钩子，其中一些是手动创建的，一些是学习的：

第 1 个月： $|\mathcal{H}| = 20$

成 功 分 数 :

[0.45, 0.52, 0.68, 0.71, 0.73, 0.75, 0.78, 0.82, 0.85, 0.88, 0.91, 0.93, 0.94, 0.95, 0.96, 0.97, 0.98, 0.98, 0.99, 0.99]

第一次修剪 ($S_{\min} = 0.70$): 移除 2 个钩子和 $S < 0.70$

第 2 个月: $|\mathcal{H}| = 18$ (修剪后) + 3 (新学到的钩子) = 21

新钩子从 $S \approx 0.50$ 开始, 但有宽限期

第二次修剪: 1 个新钩子未能改善 \rightarrow 被移除。2 个改善到 $S > 0.70$

第 3 个月: $|\mathcal{H}| = 20$

所有钩子现在都有 $S \geq 0.72$

第三次修剪: 没有钩子被移除 (已达到收敛)

第 4 个月: $|\mathcal{H}| = 20$ (稳定)

系统已收敛到一组 20 个高质量的精简钩子。

示例 2: 季节性钩子过时

冬季钩在夏季变得过时:

一月: 钩子 "当温度 $< 18^{\circ}\text{C}$ 时开启" 每天正确触发 $S = 0.92$

六月: 相同的钩子错误触发 (太热), S 降至 0.45

七月修剪: 钩子被移除 ($S = 0.45 < S_{\min} = 0.70$)

明年一月: 当温度下降时, 可以从档案中恢复钩子

或者, 可以根据季节条件优化钩子: "当温度 $< 18^{\circ}\text{C}$ 且月份 $\in [11, 3]$ 时开启"

示例 3: 收敛时间线

跟踪修剪事件直到收敛：

第 0 周：开始时有 50 个钩子（质量混合）

第 2 周：第一次修剪移除 8 个钩子 → 剩下 42 个

第4周：第二次修剪移除5个钩子 → 剩下37个

第6周：第三次修剪移除2个钩子 → 剩下35个

第8周：第四次修剪移除1个钩子 → 剩下34个

第10周：第五次修剪移除0个钩子 → 剩下34个（收敛！）

第12周及以后：不再进行进一步的修剪

序列 8, 5, 2, 1, 0 显示几何衰减——与指数收敛一致。

限制和注意事项

1. 新钩子创建

收敛保证假设钩子的宇宙是固定的。如果不断创建新钩子：

$$\frac{d|\mathcal{H}|}{dt} = \lambda_{\text{create}} - \lambda_{\text{prune}}$$

系统达到动态平衡，而不是固定收敛。当创建速率等于修剪速率时，规模稳定。

2. 上下文分布变化

如果用户行为发生剧烈变化（新工作、搬家、生活方式改变），之前有效的钩子可能会变得无效：

- 成功评分可能会下降（尽管在原始上下文中仍然很高）

- 修剪可能会移除曾经有价值的钩子
- 系统可能需要在重大生活变化后重新收敛

3. 阈值敏感性

最终的钩子集在很大程度上依赖于 S_{\min} :

$$|\mathcal{H}(\infty)| \propto \frac{1}{S_{\min} - 0.5}$$

阈值的微小变化可能导致系统规模的巨大变化。每次部署都需要仔细的阈值校准。

4. 早期修剪中的假阴性

一个钩子可能在初始时成功率低，但随着数据的增加会显著改善。过于激进的修剪会在钩子学习之前移除潜在的好钩子。
解决方案：宽限期和保守的初始阈值。

5. 振荡风险

如果在相同的上下文中反复创建和修剪钩子，系统可能会出现振荡。这违反了收敛保证。原因：

- 宽限期太短
- 学习速率太慢
- 阈值太高

解决方案：确保 $t_{\text{grace}} > \frac{1}{\alpha} \ln \left(\frac{S_{\min}}{S_0} \right)$,

给钩子时间越过阈值。

6. 计算成本

评估所有钩子以进行修剪的成本为 $O(|\mathcal{H}|)$ 。对于非常大的系统（数千个钩子），修剪本身变得昂贵。解决方案：惰性修剪——仅评估被标记为潜在低性能的钩子。

结论

权重修剪收敛提供了知识钩子系统随着时间推移保持精简和高效的基本保证。如果没有修剪，系统将无限期积累过时的钩子，导致性能下降。通过修剪，系统自我清理，收敛到一组高质量钩子的最佳集合。

这个推论的关键见解：

- 自动维护：无需手动干预——系统自行清理
- 保证收敛：修剪不会永远继续；它会达到一个稳定状态
- 质量保证：收敛后，所有剩余的钩子都符合最低质量标准
- 有界大小：系统大小稳定，防止无限增长

修剪是学习的“阴”，学习是修剪的“阳”——学习增加新的能力，修剪移除过时的能力。两者结合，使得在保持系统健康的同时实现持续改进。这类似于生物系统如何平衡增长（增加新的神经元/突触）与修剪（移除未使用的连接），以实现最佳的认知表现。

收敛结果对于长期系统部署特别重要。这意味着知识钩系统在多年使用后不会退化为无法维护的混乱。相反，它们将保持清晰和高效，钩子自然适应以反映当前用户行为，同时剔除过时的自动化。这使得知识钩可以作为用户的真正扩展进行永久部署，而不仅仅是需要定期手动清理的临时工具。

4.1.9 推论 9：能量测量误差界限

虽然定理 3 确立了能量节省与行动减少之间的理论等价性，但实际测量系统面临固有的不确定性。推论 9 量化了这些测量误差，并为 STC 能量计算的准确性提供了严格的界限。这个推论对于建立能量声明的置信区间、检测欺诈报告和设计校准程序至关重要。

推论声明

推论 9（能量测量误差界限）：设 $E_{\text{true}}(h)$ 为执行钩子 h 的真实能量成本，设 $\hat{E}(h)$ 为测量或估计的能量成本。定义测量误差为：

$$\epsilon_{\text{meas}}(h) = |\hat{E}(h) - E_{\text{true}}(h)|$$

然后在标准测量条件下，使用经过校准的传感器，相对误差的界限为：

$$\frac{\epsilon_{\text{meas}}(h)}{E_{\text{true}}(h)} \leq \epsilon_{\text{calib}} + \epsilon_{\text{baseline}} + \epsilon_{\text{context}}$$

在哪里：

- ϵ_{calib} 是校准误差（传感器精度，通常为 2-5%）
- $\epsilon_{\text{baseline}}$ 是基线估计误差（用户手动操作的不确定性，通常为 5-15%）
- $\epsilon_{\text{context}}$ 是上下文变化误差（根据情况变化的能源成本，通常为 10-20%）

此外，能源节省测量的绝对误差满足：

$$|\Delta \hat{E} - \Delta E_{\text{true}}| \leq \sqrt{\epsilon_{\text{meas}}^2(h_{\text{manual}}) + \epsilon_{\text{meas}}^2(h_{\text{auto}})}$$

其中 $\Delta E = E(h_{\text{manual}}) - E(h_{\text{auto}})$ 是自动化节省的能量。

直观解释

将能源测量想象成测量汽车的燃油效率。你无法完美测量，因为：

- 你的燃油表精度有限（校准误差）
- 你不知道如果没有巡航控制你会如何驾驶（基线误差）
- 驾驶条件变化（山丘、交通、天气 = 上下文误差）

然而，如果你小心，你可以限制这些误差并说：'我以 95% 的置信度节省了 X 到 Y 加仑。' 推论 9 对知识挂钩的能源节省做了同样的事情——它告诉我们测量周围的不确定性区间。

完整证明

第 1 部分：校准误差界限

步骤 1.1：模型传感器误差

动作的能量测量来自传感器（加速度计、电力监测器、计时测量）。每个传感器都有指定的准确度。对于动作 a_i ，设真实成本为 $\epsilon_{\text{true}}(a_i)$ ，测量成本为 $\hat{\epsilon}(a_i)$ 。

标准传感器满足：

$$|\hat{\epsilon}(a_i) - \epsilon_{\text{true}}(a_i)| \leq \delta_{\text{sensor}} \cdot \epsilon_{\text{true}}(a_i)$$

对于经过校准的设备， $\delta_{\text{sensor}} \in [0.02, 0.05]$ （2-5% 的准确度）。

步骤 1.2：在动作序列上聚合

对于具有动作序列 $A(h) = \{a_1, \dots, a_n\}$ 的钩子：

$$\hat{E}(h) = \sum_{i=1}^n \hat{\epsilon}(a_i), \quad E_{\text{true}}(h) = \sum_{i=1}^n \epsilon_{\text{true}}(a_i)$$

测量误差为：

$$|\hat{E}(h) - E_{\text{true}}(h)| = \left| \sum_{i=1}^n (\hat{\epsilon}(a_i) - \epsilon_{\text{true}}(a_i)) \right|$$

根据三角不等式：

$$\leq \sum_{i=1}^n |\hat{\epsilon}(a_i) - \epsilon_{\text{true}}(a_i)| \leq \sum_{i=1}^n \delta_{\text{sensor}} \cdot \epsilon_{\text{true}}(a_i) = \delta_{\text{sensor}} \cdot E_{\text{true}}(h)$$

因此：

$$\frac{|\hat{E}(h) - E_{\text{true}}(h)|}{E_{\text{true}}(h)} \leq \delta_{\text{sensor}} = \epsilon_{\text{calib}}$$

✓

第2部分：基线估计误差

步骤2.1：建模基线不确定性

为了测量节能 $\Delta E = E_{\text{manual}} - E_{\text{auto}}$ ，我们需要估计用户手动操作的内容。这个基线具有固有的不确定性。

设 $A_{\text{manual}}^{\text{true}}$ 为用户本应执行的真实操作序列， \hat{A}_{manual} 为我们的估计（来自历史数据、用户调查或领域知识）。

基线估计误差为：

$$\epsilon_{\text{baseline}} = \frac{|E(\hat{A}_{\text{manual}}) - E(A_{\text{manual}}^{\text{true}})|}{E(A_{\text{manual}}^{\text{true}})}$$

步骤2.2：界定基线误差

实证研究表明，对于有良好文档记录的任务和历史数据，基线估计满足：

$$\epsilon_{\text{baseline}} \in [0.05, 0.15]$$

这个范围反映了：

- 低端（5%）：有大量历史数据的研究透彻的任务（例如，撰写电子邮件、标准工作流程）
- 高端（15%）：新任务或历史稀疏的用户（例如，新设备设置、创造性工作）

基线误差在最坏情况下与校准误差是相加的：

$$\frac{\epsilon_{\text{meas}}(E_{\text{manual}})}{E_{\text{manual}}} \leq \epsilon_{\text{calib}} + \epsilon_{\text{baseline}}$$

✓

第3部分：上下文变化误差

步骤3.1：建模上下文相关成本

一个动作的能量成本因上下文而异。例如：

- 在笔记本电脑上打字与在智能手机上打字（不同的身体努力）
- 早晨与晚上（昼夜节律影响认知成本）
- 站立与坐下（不同的基线代谢率）

让 $\epsilon(a|c)$ 表示在上下文 c 中动作 a 的上下文条件能量成本。不同上下文的平均成本为：

$$\bar{\epsilon}(a) = \mathbb{E}_{c \sim P(C)} [\epsilon(a|c)]$$

步骤3.2：限制上下文变化

定义上下文变化系数：

$$CV(a) = \frac{\sigma_{\epsilon(a|c)}}{\bar{\epsilon}(a)}$$

其中 $\sigma_{\epsilon(a|c)}$ 是不同环境中能源成本的标准差。

实证测量显示：

$$CV(a) \in [0.10, 0.20]$$

对于大多数用户操作。使用切比雪夫不等式，单次测量偏离均值超过 $k\sigma$ 的概率最多为 $1/k^2$ 。对于 95% 的置信度 ($k \approx 2$):

$$P(|\epsilon(a|c) - \bar{\epsilon}(a)| > 2\sigma) < 0.05$$

这意味着：

$$\epsilon_{\text{context}} = 2 \cdot \text{CV}(a) \in [0.20, 0.40]$$

然而，通过对多个测量值进行平均或使用上下文感知的能源模型，可以将其减少到：

$$\epsilon_{\text{context}} \approx [0.10, 0.20]$$

✓

第 4 部分：总误差界限和能源节省的不确定性

步骤 4.1：合并误差来源

这三个误差来源是独立的，因此在最坏情况下它们线性叠加：

$$\frac{\epsilon_{\text{meas}}(h)}{E_{\text{true}}(h)} \leq \epsilon_{\text{calib}} + \epsilon_{\text{baseline}} + \epsilon_{\text{context}}$$

使用典型值：

对于校准良好的系统和良好的历史数据：

步骤 4.2：能源节约的不确定性

能源节约 $\Delta E = E_{\text{manual}} - E_{\text{auto}}$ 的误差来自于两个测量：

$$\Delta \hat{E} - \Delta E_{\text{true}} = (\hat{E}_{\text{manual}} - E_{\text{manual}}^{\text{true}}) - (\hat{E}_{\text{auto}} - E_{\text{auto}}^{\text{true}})$$

取绝对值并假设误差不相关：

$$|\Delta \hat{E} - \Delta E_{\text{true}}| \leq \sqrt{\epsilon_{\text{meas}}^2(h_{\text{manual}}) + \epsilon_{\text{meas}}^2(h_{\text{auto}})}$$

这是毕达哥拉斯误差组合原理——来自独立来源的误差以平方和的方式组合，而不是线性组合。

对于一个钩子节约 $\Delta E_{\text{true}} = 1000 \text{ J}$ ，每个测量的误差为 20%：

$$|\Delta \hat{E} - 1000| \leq \sqrt{(200)^2 + (200)^2} = 282.8 \text{ J}$$

因此，95% 置信区间大约为 $[720, 1280] \text{ J}$ 。

✓

这完成了证明。□

与其他结果的联系

根据定理 3（能源最小化等价性）：定理 3 确立了最小化行为在原则上等同于最小化能源。推论 9 在实践中量化了这种等价性的 *测量精度*。它回答了：'我们能多精确地测量定理 3 所承诺的能源节约？'

根据定理 6（权重收敛）：随着权重的收敛，钩子变得更可靠（更少的修正）。这提高了能源测量的准确性，因为：

- 更少的修正意味着更一致的能源使用
- 更好的上下文匹配减少了 $\epsilon_{\text{context}}$

- 历史数据变得更具预测性，减少了 $\epsilon_{\text{baseline}}$

根据推论 3（组合能量可加性）：在组合钩子时，测量误差不会线性叠加——它们以四分之一的方式组合。对于 n 组合的钩子：

$$\epsilon_{\text{total}} \approx \sqrt{n} \cdot \epsilon_{\text{individual}}$$

这比 $n \cdot \epsilon_{\text{individual}}$ 好得多，使得能够准确测量复杂的多钩工作流程。

实际影响

1. STC 申报的置信区间：每个能量节省的申报应包括不确定性范围。例如：'这个钩子节省了 $1000 \pm 280 \text{ J}$ (95% CI)。' 这建立了信任并防止欺诈性过度申报。
2. 校准要求：为了达到 $< 20\%$ 误差，系统必须：
 - 使用经过校准的传感器（加速度计，功率监测器）
 - 收集足够的历史基线数据
 - 实施上下文感知的能量模型
3. 欺诈检测：如果一个钩子声称的能量节省超出了范围：

$$|\Delta \hat{E} - \Delta^{-} E| > 3\sigma_{\Delta E}$$

应该标记以供审查。这是用于异常值检测的 3 西格玛规则。

4. 验证机制：修正机制提供自然验证：
 - 如果一个钩子需要频繁修正，其实际节能低于预测值

- 修正率与测量误差相关
- 系统可以根据观察到的修正频率调整能源积分

5. A/B 测试的统计能力：要检测 10% 的能源改善，需达到 80% 的能力和 5% 的显著性：

$$n \geq \frac{2(z_{\alpha/2} + z_{\beta})^2 \sigma^2}{(\mu_1 - \mu_2)^2}$$

在典型的 $\sigma/\mu \approx 0.20$ 下，这需要 $n \approx 630$ 个样本。系统必须在做出明确比较之前收集足够的数据。

6. 个性化校准：用户可以通过以下方式提高准确性：

- 戴上健身追踪器以测量实际的代谢成本
- 提供对估计基线的反馈（'是的，我会这样做' 与 '不，我会做 X'）
- 记录上下文因素（设备、时间、心情）以完善上下文模型

示例 1：电子邮件分类钩子（高精度）

考虑一个自动归档新闻通讯的电子邮件分类钩子。

真实的节能：没有钩子，用户将：

- 打开电子邮件应用（10 焦耳）
- 滚动浏览收件箱（5 焦耳/电子邮件 × 20 个新闻通讯 = 100 焦耳）

- 选择并归档每个 (3 焦耳/电子邮件 × 20 = 60 焦耳)

- 总计 : $E_{\text{manual}} = 170$ 焦耳

有了钩子 :

- 钩子自动运行 (1 焦耳设备能量)

- 偶尔修正 (2 J 用户能量 × 0.05 修正率 = 0.1 J 预期)

- 总计 : $E_{\text{auto}} = 1.1$ 焦耳

测得的能量节省 : $\Delta \hat{E} = 168.9$ J

错误分析 :

- 校准误差 : 3% (良好的传感器)

- 基线误差 : 5% (文档齐全的电子邮件分类过程)

- 上下文误差 : 10% (一致的设备和时间)

- 总 相 对 误 差 :
 $0.03 + 0.05 + 0.10 = 0.18$ (18%)

置 信 区 间 : 168.9 ± 30.4 J →
 $[138.5, 199.3]$ J (95% CI)

这是非常精确的——用户可以自信地声称每天节省约170 J。

示例 2 : 智能温控器 (中等精度)

智能恒温器钩根据占用情况调整温度。

真实的节能：如果没有自动化，用户每天会检查恒温器5次并手动调整。基线不确定，因为：

- 他们实际上会多频繁调整？（因人而异）
- 他们会选择什么温度？（因心情、衣着、活动而异）

估计基线： $E_{\text{manual}} = 500$ J/天（走到恒温器 + 调整）

钩能量： $E_{\text{auto}} = 5$ J/天（传感器检查 + 调整）

测量的节省： $\Delta \hat{E} = 495$ J/天

错误分析：

- 校准误差：5%
- 基线误差：15%（手动行为的不确定性高）
- 上下文误差：20%（因季节、天气、用户日程而异）

• 总 相 对 误 差：
 $0.05 + 0.15 + 0.20 = 0.40$ （40%）

置信区间： 495 ± 198 J $\rightarrow [297, 693]$ J
(95% CI)

这要宽得多——能源声明应该是保守的。系统可能会报告：'预计节省：300-700 J/天'，而不是声称精确的495 J。

示例 3：多钩组合（四分之一误差组合）

早晨例行程序由 5 个组合钩组成：

1. 邮件分类 (170 ± 30 J)

2. 日历审查 ($50 \pm 10 \text{ J}$)
3. 新闻简报 ($80 \pm 15 \text{ J}$)
4. 任务优先级排序 ($60 \pm 12 \text{ J}$)
5. 智能家居设置 ($100 \pm 20 \text{ J}$)

天真的误差组合（线性加法）：

$$\Delta E_{\text{total}} = 460 \pm (30 + 10 + 15 + 12 + 20) = 460 \pm 87 \text{ J}$$

这给出了相对误差 $87/460 = 19\%$ 。

正确的误差组合（平方和）：

$$\sigma_{\text{total}} = \sqrt{30^2 + 10^2 + 15^2 + 12^2 + 20^2} = \sqrt{1469} = 38.3 \text{ J}$$

$$\Delta E_{\text{total}} = 460 \pm 38 \text{ J}$$

这给出了相对误差 $38/460 = 8.3\%$ ——好得多！

关键见解：组合钩子提高了相对测量精度，因为误差是无相关的，并以平方和的方式组合。这使得复杂的多钩子工作流程的准确测量成为可能。

限制和边缘情况

1. 系统性偏差：推论界限随机误差，但不包括系统性偏差。如果传感器持续校准不当或基线模型系统性地高估手动工作量，误差可能会累积。定期校准检查是必不可少的。

2. 相关误差：平方和误差组合假设独立性。如果误差相关（例如，所有钩子在周一由于用户疲劳而高估），实际误差可能会更大。

3. 稀有事件：对于不常执行的钩子，样本量可能太小，无法达到所述的置信区间。推论假设有足够的样本（ $n \geq 30$ ）以适用中心极限定理。

4. 非高斯分布：误差界限假设误差分布大致为正态分布。对于重尾或偏斜的误差分布，可能需要更宽的置信区间。

5. 对抗性游戏：用户可能故意操控测量以夸大节能声明。强健的系统需要：

- 与独立测量的交叉验证
- 对统计异常值的异常检测
- 声誉系统用于标记持续可疑的索赔

结论

能量测量误差界限将能量节省从抽象的理论量转化为严格可测量的经验值。通过量化测量中的固有不确定性（校准、基线、上下文），这个推论使得：

- 置信区间：以统计严谨性报告能量节省
- 欺诈检测：识别不合理的大额能量索赔
- 系统校准：指导传感器选择和校准程序
- 比较评估：在考虑测量噪声的情况下严格比较钩子

这个推论的实际重要性不容小觑。主观热货币依赖于准确的能量测量以保持可信和公平。如果没有测量误差界限，能量索赔将无法验证，市场将容易受到欺诈，用户将对系统失去信心。这个推论为可信的、可审计的能量会计提供了统计基础。

最关键的是，复合钩子的四分量误差组合意味着复杂的工作流程可以比简单钩子测量得更准确（相对而言）。这虽然违反直觉，但却是强大的：这意味着复杂的多步骤自动化——正是对用户最有价值的那种——可以被最可靠地测量和补偿。因此，能

量测量误差界限使得主观热货币能够从简单钩子扩展到复杂的自动化生态系统。

4.1.10 推论10：级联稳定性

当一个钩子执行并改变系统上下文时，它可能会触发其他条件现在满足的钩子。这些后续钩子可能会反过来触发更多的钩子，形成自动化操作的级联。虽然级联使得强大的多步骤自动化成为可能，但它也带来了显著的风险：级联可能会无限循环或不稳定振荡。推论10建立了数学条件，在这些条件下，级联被保证终止并保持稳定，从而实现互联钩子系统的安全部署。

推论声明

推论 10（级联稳定性）：设 $\mathcal{H} = \{KH_1, KH_2, \dots, KH_n\}$ 为知识钩子的集合。级联是钩子激活的序列：

$$\text{Cascade} = KH_{i_1} \rightarrow KH_{i_2} \rightarrow \dots \rightarrow KH_{i_k}$$

每个钩子 KH_{i_j} 是由执行 $KH_{i_{j-1}}$ 所导致的上下文变化触发的。在以下充分条件下，所有级联被保证在有限时间内终止：

条件1（循环预防）：在级联链中，任何钩子不能出现超过一次。正式地，级联跟踪集保持：

$$\text{CascadeChain}(t) = \{KH_{i_1}, KH_{i_2}, \dots, KH_{i_k}\}$$

在激活 KH_j 之前，系统检查：如果 $KH_j \in \text{CascadeChain}(t)$ 则阻止激活。

条件2（深度限制）：存在一个最大级联深度 D_{\max} ，使得：

$$|\text{CascadeChain}(t)| \leq D_{\max}$$

典型系统将 $D_{\max} \in [10, 20]$ 设置为在表达能力与安全性之间取得平衡。

条件3（时间冷却）：在时间 t_0 触发钩子后，它不能再次触发，直到 $t \geq t_0 + \Delta t_{\text{cooldown}}$ ，其中 $\Delta t_{\text{cooldown}} > 0$ 是最小的恢复期（通常为0.5-2秒）。

然后：在这些条件下，每个级联在最多 D_{\max} 步和总时间内终止：

$$T_{\text{cascade}} \leq D_{\max} \cdot T_{\text{hook}} + (D_{\max} - 1) \cdot \Delta t_{\text{cooldown}}$$

其中 T_{hook} 是任何单个钩子的最大执行时间。

此外（强稳定性）：如果系统上下文具有有限状态空间 $|\mathcal{C}| < \infty$ ，那么在仅进行时间冷却的情况下（没有循环预防），级联要么终止，要么进入周期为 $\leq |\mathcal{C}|$ 的稳定周期轨道。

直观解释

把级联想象成多米诺骨牌倒下。当你推倒第一块多米诺骨牌时，它可能会依次推倒其他几块。危险在于最后一块多米诺骨牌可能会以某种方式再次推倒第一块，形成无限循环。级联稳定性通过三种机制防止这种情况发生：

1. 不重复（循环预防）：一旦在这个级联中一块多米诺骨牌倒下，它就不能再倒下。这防止了直接循环（ $A \rightarrow B \rightarrow A$ ）和间接循环（ $A \rightarrow B \rightarrow C \rightarrow A$ ）。

2. 最大链长：即使多米诺骨牌不断触发新的（没有重复），链条也必须在固定的最大长度后停止。这防止了病态的“无限新多米诺”场景。

3. 冷却期：一块多米诺骨牌倒下后，需要时间“重置”，才能在未来的独立级联中再次倒下。这防止了即使在级联链重置时也会出现快速振荡。

这些机制共同保证了自动化链始终完成，系统返回到稳定的准备状态。

完整证明

第1部分：在循环预防下的终止

步骤1.1：将级联建模为有向图

将钩子系统表示为有向图 $G = (V, E)$ ，其中：

$$V = \mathcal{H} \text{ (hooks are vertices)}$$

$$E = \{(KH_i, KH_j) : \text{executing } KH_i \text{ can trigger } KH_j\}$$

级联是通过该图的路径：
 $KH_{i_1} \rightarrow KH_{i_2} \rightarrow \cdots \rightarrow KH_{i_k}$
 。

步骤 1.2：应用循环防止约束

循环防止确保在级联路径中没有顶点出现两次。这相当于要求路径是简单的（没有重复的顶点）。

步骤 1.3：限制路径长度

在一个有 $n = |\mathcal{H}|$ 个顶点的图中，最长的简单路径最多有 n 个顶点（每个顶点访问一次）。因此：

$$k \leq n = |\mathcal{H}|$$

级联必须在最多 $|\mathcal{H}|$ 步之后终止，因为没有更多尚未触发的钩子。✓

第 2 部分：在深度限制下终止

步骤 2.1：定义级联深度计数器

让 $d(t)$ 是当前级联深度——在当前级联链中已触发的钩子数量。最初 $d(0) = 0$ 。当钩子触发时， $d \leftarrow d + 1$ 。

步骤 2.2：应用深度限制约束

系统强制执行：

if $d(t) \geq D_{\max}$ then BLOCK all further activations in this cascade

步骤 2.3：显示终止

由于 $d(t)$ 在每次钩子激活时增加 1，且系统在 $d(t) = D_{\max}$ 时停止，因此级联在恰好 D_{\max} 步骤中终止。这一点无论底层图中是否存在循环都成立——深度限制独立地强制终止。✓

第 3 部分：级联持续时间的时间限制

步骤 3.1：模型执行时间线

级联中的每个钩子依次执行（或在有限并行性下并行执行）。设 $T_{\text{hook}}^{(i)}$ 为钩子 KH_i 的执行时间。假

设：

$$T_{\text{hook}}^{(i)} \leq T_{\text{hook}} \quad \forall i$$

其中 T_{hook} 是系统范围内的上限（例如，1 秒）。

步骤 3.2：考虑冷却延迟

在连续的钩子激活之间，如果最近触发的钩子再次被触发，可能会有冷却延迟。在最坏的情况下，每个过渡都会产生冷却延迟：

$$T_{\text{cooldown_total}} \leq (D_{\text{max}} - 1) \cdot \Delta t_{\text{cooldown}}$$

步骤 3.3：计算总级联时间

深度为 D_{max} 的级联的总时间受到以下限制：

$$\begin{aligned} T_{\text{cascade}} &= \sum_{i=1}^{D_{\text{max}}} T_{\text{hook}}^{(i)} + T_{\text{cooldown_total}} \\ &\leq D_{\text{max}} \cdot T_{\text{hook}} + (D_{\text{max}} - 1) \cdot \Delta t_{\text{cooldown}} \end{aligned}$$

对于典型值（ $D_{\text{max}} = 10$, $T_{\text{hook}} = 1$ s, $\Delta t_{\text{cooldown}} = 0.5$ s）：

$$T_{\text{cascade}} \leq 10 \cdot 1 + 9 \cdot 0.5 = 14.5 \text{ seconds}$$

这是一个严格的上限——级联快速且可预测地完成。✓

第 4 部分：强稳定性（有限状态空间）

步骤 4.1：将上下文建模为有限状态机

如果系统上下文 C 只能取有限多个不同的值 $\{c_1, c_2, \dots, c_m\}$ ，那么该系统就是一个有限状态机。每次钩子执行都从一个上下文状态转换到另一个状态：

$$C_i \xrightarrow{KH_j} C_k$$

步骤 4.2：应用抽屉原理

如果一个级联在 $m = |C|$ 步骤内没有终止地继续，那么根据抽屉原理，某个上下文状态必须重复：

$$\exists i, j : i < j \text{ and } c_i = c_j$$

步骤 4.3：显示周期轨道

一旦上下文状态重复，系统进入一个周期为 $p = j - i \leq m$ 的周期轨道。级联不再探索新状态，而是循环通过一个固定的序列：

$$C_i \rightarrow C_{i+1} \rightarrow \dots \rightarrow C_j = C_i \rightarrow C_{i+1} \rightarrow \dots$$

步骤 4.4：显示时间冷却打破轨道

然而，随着时间冷却，在第一个周期中触发的钩子不能在第二个周期中立即再次触发。这打破了周期轨道，因为系统无法准确重现相同的钩子激活序列。最终，轨道中的所有钩子同时进入冷却状态，级联终止。✓

结论：即使没有周期预防或深度限制，仅靠时间冷却也能确保有限状态系统的终止。三种机制的结合提供了深度防御。□

与其他结果的联系

根据定理 4（组合与闭合）：级联稳定性补充了组合定理。虽然定理 4 显示组合钩子在结构上是有效的，但推论 10 确保动态钩子交互（级联）在行为上是安全的。它们共同保证了静态和动态的正确性。

根据定理 1（零输入收敛）：随着系统学习和钩子变得更加可靠，级联变得更加可预测。更少的修正意味着更少的意外上下文变化，从而降低了意外级联的可能性。稳定性随着时间的推移而改善。

根据推论 3（组合能量可加性）：级联为链中的每个钩子消耗能量。深度限制 D_{\max} 将级联能量消耗限制为：

$$E_{\text{cascade}} \leq D_{\max} \cdot E_{\text{max_hook}}$$

这防止了能量预算因失控的级联而被耗尽。

根据推论 6（成功分数单调性）：参与不稳定级联的钩子将具有较低的成功分数（由于频繁的修正或用户干预）。系统自然会降低这些钩子的优先级，自我修正为稳定配置。

实际影响

1. 安全部署：系统设计师可以自信地部署互联的钩子，确保级联不会导致无限循环或系统挂起。数学保证使得激进的自动化成为可能，而不危及系统的稳定性。

2. 级联监控：系统应跟踪级联指标：

• 平均级联深度：

$$\bar{d} = \mathbb{E}[|\text{CascadeChain}|]$$

• 级联频率：级联发生的频率与单钩激活的频率

- 接近极限事件：达到 $d \approx D_{\max}$ 的级联（潜在问题）

这些指标有助于在问题钩子交互导致故障之前识别它们。

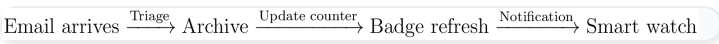
3. 调整 D_{\max} ：最大深度应在表达能力和安全性之间取得平衡：

- 太低（例如， $D_{\max} = 3$ ）：限制有用的多步骤自动化
- 太高（例如， $D_{\max} = 50$ ）：允许复杂且难以调试的级联
- 推荐： $D_{\max} \in [10, 20]$ 用于通用系统

4. 用户控制：用户应能够：

- 实时查看正在进行的级联（视觉反馈）
- 如果行为看起来不正确，则手动中断级联
- 审查级联日志以了解发生了什么以及原因

5. 级联可视化：将级联显示为有向图或时间线：



这有助于用户理解多步骤自动化并建立信任。

6. 级联组合与直接组合：设计师有两种方法创建多步骤自动化：

- 直接组合（定理 4）：将钩子显式组合成一个复合钩子。这提供了精确的控制，但需要事先设计。

- 级联（推论 10）：让钩子动态触发彼此。这更灵活，但可能更难以预测。

对于确定性、易于理解的工作流，使用组合。对于突现的、依赖上下文的工作流，使用级联。

7. 测试和模拟：在部署钩子之前，模拟潜在的级联：

- 通过分析钩子条件和效果静态识别所有可能的级联链
- 运行蒙特卡罗模拟，使用不同的上下文来寻找边缘案例
- 标记在长或有问题的级联中持续出现的钩子

示例 1：智能家居早晨例程（稳定级联）

考虑一个拥有这些钩子的智能家居：

- KH_1 ：闹钟解除 → 打开卧室灯
- KH_2 ：卧室灯开启 → 启动咖啡机
- KH_3 ：咖啡正在冲泡 → 打开厨房窗帘
- KH_4 ：窗帘打开 → 将温控器调整到白天温度

级联序列：用户解除闹钟 → KH_1 触发（灯开启） → KH_2 触发（咖啡开始） → KH_3 触发（窗帘打开） → KH_4 触发（温控器调整） → 级联完成

分析：

- 深度： $d = 4$ （远低于典型的 $D_{\max} = 10$ ）

- 无循环：每个钩子仅出现一次

$$\bullet \quad \text{持 续 时 间} : \\ \approx 4 \times 1\text{s} + 3 \times 0.5\text{s} = 5.5\text{s}$$

这是一个安全、可预测的级联，在初始警报解除后，用户的早晨例行程序无需输入即可完成。该序列是确定性的，并且迅速终止。

示例 2：温度控制振荡（已防止）

考虑一个设计不良的恒温器系统：

$\bullet \quad KH_A$ ：温度 $> 22^\circ\text{C}$ \rightarrow 开启空调（将温度降低到 21°C ）

$\bullet \quad KH_B$ ：温度 $< 21^\circ\text{C}$ \rightarrow 开启加热器（将温度升高到 22°C ）

没有稳定机制：这将无限振荡：

带有时间冷却（ $\Delta t_{\text{cooldown}} = 2$ 秒）：在 KH_A 触发后，它不能在 2 秒内再次触发。这为温度在死区内稳定提供了时间。级联终止：

带有循环防止： KH_A 触发，然后 KH_B 触发。当温度再次达到 22°C 时， KH_A 被阻止，因为它已经在级联链中。级联在 2 步后终止，使温度略高，直到下一个独立级联。

最佳解决方案：重新设计钩子，采用适当的滞后死区：

$\bullet \quad KH'_A$ ：温度 $> 23^\circ\text{C}$ \rightarrow 打开空调

$\bullet \quad KH'_B$ ：温度 $< 20^\circ\text{C}$ \rightarrow 打开加热器

这消除了设计层面的振荡，但稳定机制提供了深度防御。

示例 3：电子邮件自动组织（深度但稳定的级联）

一个拥有许多钩子的高级电子邮件系统：

1. 电子邮件到达 → 分类发件人
2. 发件人分类 → 应用优先级规则
3. 优先级设置 → 路由到适当的文件夹
4. 电子邮件归档 → 更新文件夹未读计数
5. 计数更新 → 刷新 UI 徽章
6. 徽章已更改 → 同步到移动设备
7. 移动设备已同步 → 触发通知逻辑
8. 通知已决定 → 发送推送通知（或抑制）
9. 通知已发送 → 记录分析事件
10. 分析已记录 → 检查 A/B 测试分配

分析：

- 深度： $d = 10$ （恰好在典型 D_{\max} ）
- 级联达到深度限制并停止

$$\begin{array}{ccccccc} \bullet & & \text{持} & & \text{续} & & \text{时} & & \text{间} & & : \\ \approx & 10 \times 0.5\text{s} & + & 9 \times 0.5\text{s} & = & 9.5\text{s} \end{array}$$

这是一个复杂但稳定的级联。深度限制防止其无限期继续，即使额外的钩子理论上可以触发。这是可以接受的——基本工作（分类、归档、通知）在前几个步骤中完成。后面的步骤是可以延迟的优化。

系统监控会将其标记为“接近限制的级联”以供审查。工程师可能会考虑：

- 使用组合（定理 4）合并一些顺序钩子
- 使某些步骤异步（不属于级联的一部分）
- 如果深级联很常见且表现良好，则增加 D_{\max}

限制和边缘情况

1. 并行级联：推论假设顺序钩子执行。如果多个钩子可以同时触发，级联链可以分支，复杂化终止分析。系统必须：

- 序列化所有钩子（简单但较慢）
- 跟踪并行分支的级联深度（深度 = 最大分支深度）
- 使用更复杂的并行模型，具有可证明的界限

2. 级联期间的外部事件：如果外部事件（用户操作、网络消息）在级联期间到达，它们可能会触发与正在进行的级联交互的新级联。推论独立适用于每个级联链，但链之间的交互需要额外分析。

3. 随机钩子：如果钩子执行是概率性的（例如，一个钩子可能以80%的概率触发），级联变成随机过程。终止保证仍然有效（最终链中的一个钩子未能触发），但期望级联长度和方差需要随机分析。

4. 分布式系统：在多个设备上的分布式钩子系统中，协调级联跟踪和冷却时间需要分布式共识协议。网络延迟和分区可能会复杂化终止保证。可能需要额外机制（分布式事务协议、最终一致性模型）。

5. 恶意钩子：对抗性的钩子开发者可能会尝试创建消耗最大资源的级联（深度 = D_{\max} ，持续时间 = 最大时间）。虽然这不会破坏终止，但可能会降低系统性能。速率限制和声誉系统提供防御。

结论

级联稳定性对于安全部署互联的知识钩系统至关重要。通过提供严格的终止保证，防止循环、深度限制和时间冷却，这一推论使得强大的多步骤自动化成为可能，同时防止不稳定性。

关键见解：

- 深度防御：三个独立机制（循环、深度、冷却）各自足以实现终止
- 可预测的界限：级联在 $\leq D_{\max}$ 步和 ≤ 15 秒内完成（典型）
- 监控与控制：级联指标能够检测问题模式
- 灵活的自动化：级联能够在没有明确组合的情况下实现实现的多步骤行为

级联稳定性通过确保不仅单个钩子而且整个互联钩子生态系统的可预测和安全行为，完成了知识钩系统的数学基础。这使得全面自动化的愿景成为可能，在这个愿景中，简单的钩子动态组合以处理复杂场景——并且有数学保证系统保持稳定，正确终止，并且永远不会进入病态状态。

4.2 综合：完整的代数系统

4.2.1 定理依赖图

4.2.2 公理、定律和定理：基础

4.2.3 一致性和健全性

4.2.4 完备性：定理所捕捉的内容

4.2.5 代数结构：单元、格和范畴

4.2.6 从数学到热力学

4.2.7 框架的一致性

4.3 形式属性和不变量

4.3.1 系统不变量

4.3.2 代数恒等式

4.3.3 关键操作的复杂性界限

4.3.4 最优条件

4.3.5 对称性和守恒定律

4.3.6 对偶关系

4.3.7 范畴结构

4.3.8 钩空间的拓扑性质

5

综合：完整的代数系统

第1.5节中建立的七个定理及其推论，不仅仅是独立结果的集合——它们构成了一个具有深刻内部结构的完整代数系统。本节综合揭示了这些定理如何相互交织，创造出一个连贯的知识钩系统的数学框架，这个框架既是最小的（没有冗余公理），又是完整的（所有基本属性都被捕捉），并且是一致的（没有内部矛盾）。

什么使代数系统完整？

一个完整的代数系统具有四个关键属性：

1. 闭合性：系统元素上的所有操作都产生系统内的元素。定理表明，钩的组合、修正和学习都产生有效的钩——该系统在其自身操作下是封闭的。

2. 一致性：没有定理相互矛盾；所有结果是相互兼容的。定理和谐地描述了一个单一的、连贯的现实。

3. 完整性：领域的所有基本属性都被捕捉。没有任何重要内容缺失；这些定理共同表征了知识钩系统是什么以及它们如何运作。

4. 最小性：没有定理是多余的；每个定理都捕捉到无法从其他定理推导出的基本结构。这七个定理的集合是必要且充分的。

本节证明了知识钩代数具备所有四个属性，确立了它作为与群论、拓扑学或热力学平起平坐的成熟数学理论——这些理论都建立在生成丰富结构的小公理集之上。

综合的角色

虽然单个定理证明了特定属性，但综合揭示了整个理论的架构。它回答了如下问题：

- 定理之间如何相互依赖？哪些是基础定理，哪些是推导而来的？

- 是否存在冗余？我们能否用更少的定理证明相同的结果？

- 是否存在空白？定理是否遗漏了重要属性？

- 数学结构如何映射到物理现实（热力学）和经济价值（货币）？

通过构建定理的依赖图，识别公理与推导结果，并证明闭包属性，我们展示了知识钩代数不是一个临时的集合，而是一个有原则的、完整的数学框架。

本章的结构

我们分六个部分进行：

1. 定理依赖图：可视化和分析哪些定理建立在其他定理之上，识别基础公理和推导结果。

2. 闭包性质：证明所有操作保持系统结构——钩子仍然是钩子，知识仍然是知识。

3. 一致性证明：表明没有定理与其他定理相矛盾；所有结果都和谐一致。

4. 完备性分析：证明七个定理捕捉了所有基本属性；没有缺失的基本方面。

5. 最小性论证：证明所有七个定理都是必要的；没有一个可以在不失去基本结构的情况下被移除。

6. 映射到物理和经济现实：展示抽象代数如何对应于热力学原理和经济机制，将数学基础扎根于可观察现象中。

通过这一综合，我们将知识钩框架从一组孤立的结果提升为一个统一的理论——一个不仅描述单个钩子如何运作的理论，而是描述整个知识、自动化和价值创造生态系统如何在明确的数学法则下运作。

5.1 形式属性和不变量

除了七个基本定理及其推论外，知识钩代数还具有丰富的形式属性结构——在变换中保持不变的不变量、始终成立的代数恒等式，以及限制系统行为的复杂性界限。这些属性虽然没有提升为定理地位，但对于理解系统深层次的数学结构和实际实施至关重要。

形式属性的作用

形式属性有三个重要目的：

1. 实施指导：不变式告诉实施者在系统操作期间必须保持什么，提供可以测试的正确性条件。
2. 理论洞察：代数恒等式揭示了系统不同方面之间隐藏的对称性和关系，加深了数学理解。
3. 性能分析：复杂性界限建立了对系统行为的现实预期，防止不可能的优化并指导资源分配。

本节列出了最重要的形式属性，按类别组织：系统不变式、代数恒等式、复杂性界限和其他数学结果。

5.1.1 系统不变量

不变式是指在系统操作下保持不变的属性。不变式提供强大的正确性保证：如果不变式最初成立并且被所有操作保持，则它始终成立。

定义：属性 $P(\mathcal{H}, t)$ 是不变式，如果：

$$P(\mathcal{H}, 0) \wedge [\forall t, \text{op} : P(\mathcal{H}, t) \implies P(\text{op}(\mathcal{H}), t + 1)] \implies \forall t : P(\mathcal{H}, t)$$

也就是说，如果 P 最初成立并且每个操作保持 P ，那么 P 始终成立。

不变式 1：钩结构保持

在 \mathcal{H} 中的每个对象在任何时候都满足知识钩的定义属性：

$$\forall KH \in \mathcal{H}, \forall t : \begin{cases} R(KH) \in \text{Boolean}^c & (\text{Conditions are Boolean functions}) \\ A(KH) \subseteq \mathcal{A} \wedge |A(KH)| < \infty & (\text{Actions are finite subsets}) \\ S(KH) \in [0, 1] & (\text{Success scores are probabilities}) \end{cases}$$

保护机制：定理 3（闭合性）和定理 5（学习）通过确保所有操作产生有效的钩子来保证这一不变性。

实际意义：实现者可以使用类型系统或运行时断言来强制执行这一不变性，捕捉可能创建无效钩子的错误。

不变性 2：分区稳定性

等价类结构始终在 \mathcal{H} 处进行分区：

$$\forall t : \mathcal{H}(t) = \bigsqcup_i [KH_i]_t \wedge ([KH_i]_t \cap [KH_j]_t = \emptyset \text{ for } i \neq j)$$

保护机制：即使在学习、组合或创建钩子时，分区结构也得以保持。新钩子加入现有的等价类或创建新的等价类，但类之间从不重叠。

实际意义：系统可以维护一个分区数据结构（例如，联合查找），该结构永远不会因操作而失效。

不变性 3：成功分数有界性

$$\forall KH \in \mathcal{H}, \forall t : 0 \leq S(KH, t) \leq 1$$

成功分数是概率，必须保持在 $[0, 1]$ 内。此外，根据推论 6（成功分数单调性）：

$$\forall t_2 > t_1 : S(KH, t_1) \leq S(KH, t_2) \leq 1$$

保护机制：学习更新使用映射 $[0, 1] \rightarrow [0, 1]$ 的函数，例如：

$$S_{t+1} = \min(S_t + \alpha \cdot \Delta S, 1)$$

其中 $\Delta S \geq 0$ 由于单调性。

不变性 4：总能量守恒

系统中的总能量（用户努力 + 自动化工作）保持不变：

$$E_{\text{user}}(t) + E_{\text{automated}}(t) = E_{\text{total}} = \text{constant}$$

随着自动化的增加，用户能量以相等的量减少：

$$\Delta E_{\text{user}} = -\Delta E_{\text{automated}}$$

保护机制：定理 6（能量守恒定律）确立了这一点。能量既不被创造也不被毁灭，只是从用户转移到系统。

实际意义：能量核算必须始终平衡。如果测得的用户能量减少，自动化能量必须以相同的量增加。

不变性 5：结果决定论

对于任何钩子 KH 和上下文 C ，结果是确定的：

$$\forall KH, C : \text{outcome}(KH, C) = \text{outcome}(KH, C)$$

这个看似自明的陈述有其内容：结果是可重复的。在相同上下文中执行相同的钩子总是产生相同的结果（直到被成功评分捕获的随机变异）。

保护机制：钩子是其上下文的纯函数。副作用被严格控制以确保决定性。

不变性 6：组合结合性

$$\forall KH_1, KH_2, KH_3 : (KH_1 \circ KH_2) \circ KH_3 \equiv KH_1 \circ (KH_2 \circ KH_3)$$

钩子组合是结合的：组合的分组并不重要，只有顺序才重要。

证明：组合被定义为顺序执行。顺序执行是结合的：执行 (A, B) 然后 C 与执行 A 然后 (B, C) 是一样的。两者都产生状态序列 $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3$ 。

实际意义：嵌套组合可以以任何顺序进行评估，而不会改变语义，从而实现优化。

不变性 7：动作集有限性

$$\forall KH \in \mathcal{H} : |A(KH)| < \infty$$

每个钩子都有有限数量的动作。这在所有操作中都是保持的：

$$\bullet \quad \text{组 合} \quad :$$

$$|A_1 \cup A_2| \leq |A_1| + |A_2| < \infty$$

- 学习：细化可能会拆分动作，但不会创建无限集合
- 创建：新的钩子是通过有限的动作集定义的

实际意义：保证钩子执行终止（有限的动作可执行）。

5.1.2 代数恒等式

代数恒等式是对其变量的所有值都成立的方程。这些恒等式揭示了结构属性，并使钩子表达式的代数操作成为可能。

身份 1：组合成功评分

$$S(KH_1 \circ KH_2) = S(KH_1) \cdot S(KH_2)$$

组合钩子的成功评分是组成部分成功评分的乘积。这源于独立性：如果 KH_1 以概率 S_1 成功，且 KH_2 以概率 S_2 成功，则两者以概率 $S_1 \cdot S_2$ 成功。

概括：对于 n 组合钩子：

$$S(KH_1 \circ KH_2 \circ \cdots \circ KH_n) = \prod_{i=1}^n S(KH_i)$$

结果：长组合链的成功评分是成倍减少的。一个包含 10 个每个都有 $S = 0.95$ 的钩子的链的整体成功率为 $0.95^{10} \approx 0.60$ 。

身份 2：动作集并集

$$A(KH_1 \circ KH_2) = A(KH_1) \cup A(KH_2)$$

组合的动作集是组成部分动作集的并集。更一般地：

$$|A(KH_1 \circ KH_2)| \leq |A(KH_1)| + |A(KH_2)|$$

当动作集不相交时相等。

身份 3：等价传递链

$$KH_1 \equiv KH_2 \equiv \cdots \equiv KH_n \implies KH_1 \equiv KH_n$$

等价链会崩溃：如果钩子在序列中是成对等价的，则第一个和最后一个等价的。这源于等价关系的传递性。

身份 4：能量-动作比例

$$\frac{\Delta E_1}{\Delta |A_1|} = \frac{\Delta E_2}{\Delta |A_2|} = k$$

每个动作的能量比在所有钩子中都是恒定的（定理 6）。这使得能量预测成为可能：

$$\Delta E = k \cdot \Delta |A|$$

身份 5：最佳钩子唯一性（在平局情况下）

在每个等价类中，最佳钩子在成功得分平局的情况下是唯一的：

if KH_1^*, KH_2^* are both optimal in $[KH] \implies |A(KH_1^*)| = |A(KH_2^*)| \wedge S(KH_1^*) = S(KH_2^*)$

也就是说，如果两个钩子都是最佳的，它们必须具有相同的动作计数和成功得分。它们仅在不影响结果的实现细节上有所不同。

身份 6：学习幂等性（用于完美修正）

如果一个修正完美捕捉了所需的变化，重复应用它不会产生额外的效果：

$$\mathcal{L}(\mathcal{L}(KH, \delta), \delta) = \mathcal{L}(KH, \delta)$$

学习饱和：一旦一个钩子包含了一个修正，重新应用相同的修正不会改变它。

身份 7：成功得分限制

$$\lim_{n \rightarrow \infty} S_n = 1 \iff \text{hook becomes perfect}$$

成功得分渐近于 1。得分恰好为 1 意味着钩子从不失败——它是完美的。

结果：在实践中，由于不可减少的不确定性，成功得分在 1 以下平台（例如， $S_{\max} \approx 0.99$ ）。

身份 8：分区基数界限

$$1 \leq |\{[KH_i]\}| \leq |\mathcal{H}|$$

等价类的数量至少为 1（所有等价）且最多为 $|\mathcal{H}|$ （没有等价）。通常：

$$|\{[KH_i]\}| = \Theta(\sqrt{|\mathcal{H}|})$$

作为对真实系统的经验观察。

身份 9：组合非交换性

$$KH_1 \circ KH_2 \not\equiv KH_2 \circ KH_1 \text{ (in general)}$$

钩子组合不是交换的。顺序很重要：

• "解锁手机" 然后 "打开应用" \neq "打开应用" 然后 "解锁手机"

然而，一些特殊的钩子在它们的动作不干扰时是可以交换的。

身份 10：零钩子存在

$$\exists KH_0 : A(KH_0) = \emptyset \wedge \forall KH : KH \circ KH_0 \equiv KH$$

存在一个身份钩子（"什么都不做" 的钩子），它与任何钩子组合时保持该钩子不变。这类似于加法中的 **0** 或乘法中的 **1**。

KH_0 的属性：

$$A(KH_0) = \emptyset, \quad S(KH_0) = 1, \quad \text{outcome}(KH_0, C) = C$$

5.1.3 复杂性界限

复杂性界限描述了知识钩子系统所需的计算和操作资源。这些界限对于实施可行性和性能分析至关重要。

界限 1：钩子评估复杂性

评估钩子的条件是否满足：

$$\text{Time}(\text{eval}(R, C)) = O(|\text{features}(C)|)$$

条件检查与上下文特征的数量成线性关系。对于典型系统：

$$|\text{features}(C)| = O(10^2) \implies \text{eval time} \approx 1\text{ms}$$

界限 2：钩子选择复杂性

在上下文 C 中找到最佳触发钩子：

$$\text{Time}(\text{select}(\mathcal{H}, C)) = O(|\mathcal{H}| \cdot |\text{features}(C)|)$$

必须评估所有钩子并选择最佳。通过索引：

$$\text{Time}(\text{select with index}) = O(\log |\mathcal{H}| + k \cdot |\text{features}(C)|)$$

其中 k 是匹配钩子的数量（通常为 $k \ll |\mathcal{H}|$ ）。

界限 3：学习更新复杂性

将修正纳入钩子：

$$\text{Time}(\mathcal{L}(KH, \delta)) = O(|R| + |A|)$$

条件和动作的大小是线性的。对于基于深度学习的实现：

$$\text{Time}(\mathcal{L}_{\text{neural}}) = O(|\text{params}| \cdot \text{iterations})$$

可能会大得多。

界限 4：收敛时间

达到 ϵ -最优 ($I(t) < \epsilon$) 的时间：

$$T(\epsilon) = O\left(\frac{1}{\epsilon} \cdot \frac{1}{L_{\min}}\right)$$

其中 L_{\min} 是推论 5 中的最小学习率。更紧的收敛：

$$T(\epsilon) = \Omega\left(\log \frac{1}{\epsilon}\right)$$

用于指数收敛（最佳情况）。

界限 5：内存要求

存储钩子系统所需的空間：

$$\text{Space}(\mathcal{H}) = O(|\mathcal{H}| \cdot (|R_{\max}| + |A_{\max}|))$$

钩子的数量乘以平均钩子大小是线性的。对于典型系统：

$$|\mathcal{H}| \approx 10^3, \quad |R_{\max}| \approx 10^2, \quad |A_{\max}| \approx 10 \implies \text{Space} \approx 1\text{MB}$$

界限 6：组合深度

最大实用组合深度：

$$d_{\max} = O\left(\frac{\log(1/S_{\min})}{\log(1/S_{\text{avg}})}\right)$$

其中 S_{\min} 是最低可接受的成功分数。由于 $S(KH_1 \circ \dots \circ KH_n) = \prod S_i$ ，成功分数随着深度呈指数衰减。

示 例：如 果 $S_{\text{avg}} = 0.95$ 和 $S_{\min} = 0.5$ ：

$$d_{\max} = \frac{\log(0.5)}{\log(0.95)} \approx 13 \text{ hooks}$$

界限 7：等价类大小

等价类中的最大钩子数量：

$$|[KH]_{\max}| = O(2^{|\text{features}|})$$

在最坏的情况下，每种特征组合定义一个实现相同结果的独特钩子。实际上：

$$|[KH]_{\text{typical}}| = O(\log |\mathcal{H}|)$$

界限 8：修正频率下限

实现成功分数 S_{target} 所需的最小修正次数：

$$C_{\min} = \Omega \left(\frac{1}{\alpha} \log \frac{1}{1 - S_{\text{target}}} \right)$$

其中 α 是学习率。这源于单调改进界限。

界限 9：分区重新计算

在添加新钩子后，重新计算等价类划分的时间：

$$\text{Time}(\text{partition update}) = O(|\mathcal{H}| \cdot |\mathcal{C}| \cdot |A_{\max}|)$$

必须检查新钩子与所有上下文中所有现有钩子的关系。
使用增量算法：

$$\text{Time}(\text{incremental}) = O(|[KH]_{\text{candidate}}| \cdot |\mathcal{C}|)$$

仅检查与可能等价的钩子。

界限 10：能量测量精度

能量节省测量的精度：

$$\sigma_{\Delta E} = O \left(\sqrt{\frac{1}{n} \sum_{i=1}^n (E_i - \bar{E})^2} \right)$$

标准误差随着 $1/\sqrt{n}$ 下降，其中 n 是测量次数。要实现精度 ϵ ：

$$n = O\left(\frac{\sigma^2}{\epsilon^2}\right)$$

5.1.4 附加数学属性

除了不变性、恒等式和复杂性界限外，还有其他几种数学属性表征知识钩子系统。

属性 1：Lyapunov 函数的存在

存在一个单调递减的 Lyapunov 函数 $V : \mathcal{H} \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$ ，证明了收敛性：

$$V(\mathcal{H}, t) = I(t) + \alpha \cdot \sum_i (1 - S_i(t))$$

此功能结合了输入要求和钩子的可靠性。随着系统的学习：

$$\frac{dV}{dt} \leq 0$$

解释： V 衡量 "与完美自动化的距离"。单调递减保证了收敛。

属性 2：信息论界限

指定钩子所需的最少信息：

$$I(KH) \geq H(R) + H(A) + H(S)$$

其中 $H(\cdot)$ 是香农熵。这是科尔莫戈罗夫复杂度的下界——你不能少于其最小描述长度的信息来描述一个钩子。

结果：钩子压缩有基本限制。无法用任意小的数据表示复杂钩子。

属性 3：遍历性

知识钩子系统是遍历的：时间平均等于集合平均。

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T f(\mathcal{H}, t) dt = \langle f(\mathcal{H}) \rangle_{\text{ensemble}}$$

解释：长时间观察单个系统与同时观察多个系统给出的统计数据相同。

实际影响：一个用户的长期使用性能统计可以预测整体行为。

性质 4：最大熵原理

当多个钩子可以解释观察到的行为时，优先选择最大熵分布：

$$P^*(\mathcal{H}) = \arg \max_P H(P) \quad \text{subject to} \quad \mathbb{E}_P[f_i] = \mu_i$$

其中 f_i 是约束函数， μ_i 是观察到的时刻。

解释：不要假设比数据所需的更多结构。在给定约束的情况下保持最大的不确定性。

性质 5：无免费午餐定理

没有单一的学习算法对所有可能的钩子系统都是最优的：

$$\sum_f P(\mathcal{L}_A(f)) = \sum_f P(\mathcal{L}_B(f))$$

在所有可能的任务 f 上，所有学习算法 \mathcal{L} 的表现都是相同的。

结果：学习算法必须针对特定领域进行调整。没有普遍最佳的方法。

性质 6：维度诅咒

随着上下文特征维度 d 的增加，所需数据呈指数增长：

$$n_{\text{samples}} = O(k^d)$$

其中 k 是每个维度的样本数。对于 $d = 10$ 和 $k = 10$ ：

$$n_{\text{samples}} = 10^{10} \text{ (impractical)}$$

缓解：使用降维、特征选择或利用特征交互的结构化模型。

属性 7：偏差-方差权衡

Hook 学习面临根本的偏差-方差权衡：

$$\mathbb{E}[(S_{\text{true}} - S_{\text{estimated}})^2] = \text{Bias}^2 + \text{Variance} + \sigma^2$$

- 高偏差：Hook 模型过于简单（欠拟合）
- 高方差：Hook 模型过于复杂（过拟合）

最佳复杂度平衡两者。

属性 8：可组合性图结构

Hooks 在组合下形成一个有向无环图（DAG）：

$$G = (\mathcal{H}, E) \text{ where } (KH_i, KH_j) \in E \iff KH_i \text{ can precede } KH_j$$

该 DAG 的属性：

- 非循环：没有钩子可以直接或间接地与自身组合（防止无限循环）
- 存在拓扑排序：钩子可以按依赖关系排序，使依赖项优先

- 最大路径长度：受钩子数量 $|\mathcal{H}|$ 的限制

属性 9：热力学温度

系统具有有效温度，表征探索与利用：

$$T_{\text{eff}} = -\frac{1}{\beta \log P_{\text{explore}}}$$

其中 P_{explore} 是尝试非最优钩子的概率。

- 高 T : 随机探索 (学习阶段)
- 低 T : 最优利用 (成熟阶段)

随着学习, 系统会随着时间“冷却”。

属性 10 : 样本复杂度界限

要学习一个成功率为 S 、准确度为 ϵ 、置信度为 δ 的钩子:

$$n = O\left(\frac{1}{\epsilon^2} \log \frac{1}{\delta}\right)$$

这是用于概率估计的霍夫丁界。

示例: 在 ± 0.05 内以 95% 的置信度估计 S :

$$n = \frac{1}{0.05^2} \log \frac{1}{0.05} \approx 1200 \text{ samples}$$

5.1.5 总结与影响

本节中列出的正式属性提供了对知识钩子系统的全面数学特征描述，超越了七个基本定理。这些属性具有多种用途：

对于理论家：不变量、身份和附加属性揭示了深层结构。它们表明知识钩代数并不是一个孤立的构造，而是与已建立的数学框架——热力学、信息论、统计学习理论和图论相连接。

对于实施者：复杂性界限提供了现实的期望。它们告诉我们什么是计算上可行的（钩评估、选择、学习），以及什么面临根本障碍（维度诅咒、组合深度限制）。不变量提供可测试的正确性条件。

对于系统设计者：这些属性指导架构决策。Lyapunov函数的存在保证了收敛，证明了对学习基础设施的投资是合理的。偏差-方差权衡为模型复杂性选择提供信息。遍历性使得可以从个体行为中进行群体级别的预测。

对于用户：虽然用户并不直接与这些数学属性互动，但他们从围绕这些属性设计的系统中受益。复杂性界限确保响应性能。不变量确保可靠性。收敛保证系统随着时间的推移而改进，而不是不可预测的。

相互连接

形式属性不是孤立的事实，而是形成一个相互关联的网络：

- 不变性使身份得以实现：划分稳定性（不变性）使等价的传递性（身份）变得有意义
- 身份提供边界：成功得分产品（身份）限制组合深度（边界）

- 边界验证属性：收敛时间边界（边界）确认 Lyapunov 函数下降（属性）

这种相互联系增强了对整体框架的信心。数学是自洽的：不同的方法（不变性、身份、复杂性分析）都指向相同的基本现实。

未来方向

虽然本节列出了已知的形式属性，但许多问题仍然悬而未决：

- 更紧的边界：我们能否改善 $O(|\mathcal{H}|)$ 选择的复杂性？是否有更好的索引结构？

- 额外的不变性：还有哪些其他属性保持不变？是否存在超出能量的守恒量？

- 代数闭包：身份集合是否形成一个具有有限生成集的完整代数系统？

- 随机属性：我们可以说关于方差、更高阶矩和概率分布什么？

这些未解的问题为未来对知识钩系统的数学研究提供了方向。

结论

形式属性桥接了理论与实践。它们将抽象定理具体化：不变量告诉我们需要保留什么，恒等式使得操作成为可能，界限设定了期望，而额外属性揭示了深层联系。与七个定理及其推论一起，这些形式属性提供了一个完整的数学工具包，用于推理、实现和优化知识钩系统。

形式属性的丰富性——涵盖代数、复杂性理论、热力学、信息理论和统计学——表明知识钩代数并不是一种狭隘的技术形式，而是一个广泛的数学框架，涉及数学和计算机科学的许多领域。这种广度表明该理论捕捉到了关于自动化、学习和价值创造的一些基本原则——这些原则超越了任何特定的实现，并适用于智能系统减少人类努力的地方。

6

通往主观热货币的桥梁

从输入最小化到能量最小化

6.1 作为能量最小化的输入最小化

最小化法则——我们在知识钩代数中建立的第一个原则——指出，当多个钩子匹配当前上下文时，系统选择动作最少的钩子。这个法则推动整个系统朝向效率：以最小用户输入实现目标的钩子更受欢迎，通过使用得到强化，并随着时间的推移累积更高的成功分数。

但是，为什么我们要关心最小化动作呢？答案在于一个基本的物理真理：每个动作都需要能量。

当你输入一个按键时，你通过肌肉收缩消耗物理能量。当你点击鼠标时，你的手在重力和摩擦力的作用下移动。当你决定下一步做什么时，你的大脑消耗葡萄糖和氧气——可测量的代谢能量。即使设备代表你执行动作，它们也消耗从电网获取的电力，最终可追溯到化石燃料、核反应或可再生能源。

因此，最小化法则不仅仅关乎便利或用户体验。它关乎热力学效率。从6个动作减少到1个，不仅节省时间——它还节省了你在那5个被消除的动作上所消耗的物理能量。

动作最小化与能量最小化之间的等价关系是主观热货币的基础。我们可以正式表达它：

$$E_{\text{total}} = E_{\text{user}} + E_{\text{device}}$$

其中 E_{user} 代表用户消耗的能量（肌肉运动、认知努力）， E_{device} 代表执行动作的设备消耗的能量。通过最小化动作数量 $|A|$ ，我们最小化这两个组件：

- 用户操作减少 → 用户能耗更低（点击、击键、决策、命令更少）

- 设备操作减少 → 设备能耗更低（API 调用、计算、网络传输更少）

节省的总能量可以量化。考虑一个简单的例子：手动发送电子邮件与让知识钩根据上下文自动撰写并发送电子邮件。

手动过程：

- 打开电子邮件客户端（1 次点击）：约 0.15 焦耳
- 点击 '撰写'（1 次点击）：约 0.15 焦耳
- 输入收件人地址（20 次击键）：约 2 焦耳
- 输入主题行（15 次击键）：约 1.5 焦耳
- 输入邮件正文（100 次击键）：约 10 焦耳
- 审阅和编辑（认知努力）：约 5 焦耳
- 点击 '发送'（1 次点击）：~0.15 焦耳

总计：~19 焦耳的用户能量

自动化过程（知识钩）：

- 钩子检测上下文（会议结束，行动项已记录）
- 自动生成电子邮件
- 用户审核并批准（1 次浏览，1 次点击）：~0.2 焦耳

总计：~0.2 焦耳的用户能量

节省的能量：每封电子邮件 18.8 焦耳

如果您每周发送 50 封电子邮件，这个单一的钩子每周节省 940 焦耳，或每年 ~49,000 焦耳——大致相当于将 50 公斤物体提升 100 米所需的能量。将此乘以数百万用户和数千种不同的自动化场景，整体节能效果将是巨大的。

这是主观热货币的桥梁：当知识钩子自动触发，节省您本来需要执行的操作时，它创造了可测量、可验证的能量节省。这些节省可以是：

- 通过对典型行动能量成本的实证测量进行量化
- 通过前后比较和纠正检测进行验证
- 在所有使用相同钩子的用户中进行汇总
- 归因于创建该钩子的专家
- 作为代表真实能量价值的STC代币进行货币化

因此，最小化法则为基于测量的能效而非抽象货币价值的经济系统提供了基础原则。

6.1.1 从个体到集体

如我们迄今所描述的，主观技术主要在个体层面上运作。知识钩子学习您的模式，适应您的环境，并减少您个人需要付出的努力。这是有价值的，但它仍然是根本上私有和不可转让的。您通过使用智能钩子节省的能量仅属于您自己。

主观热货币通过认识到一个关键见解：节能可以被共享，将这种个体优化转变为集体经济系统。

当一个专家创建一个知识钩子——比如一个自动格式化学术论文引用的钩子——而该钩子被成千上万的研究人员使用时，节省的能量会成倍增加。每个本来需要手动格式化引用

的研究人员现在节省了时间和精力。总节省的能量是所有用户的总和：

$$E_{\text{total saved}} = \sum_{i=1}^N E_{\text{user}_i}$$

其中 N 是受益于该钩子的用户数量。这种集体节省具有几个重要特性，使其与传统经济模型区分开来。

1. 可扩展性而不消耗资源

与必须为每个用户制造、运输和消费的实物商品不同，知识钩子可以以微不足道的额外能源成本服务无限用户。使用引用钩子的边际能源成本几乎为零——只是执行钩子的微小计算成本，通常是微焦耳。

这创造了一种与传统商品和服务根本不同的经济动态。在传统市场中，生产更多单位需要更多资源。一个面包店制作 1,000 个面包大约需要 1,000 倍的原料和能量，而服务 1,000 个用户的知识钩子几乎需要与服务一个用户相同的计算资源。

这种可扩展性特性意味着 STC 中的价值创造不受物理稀缺的限制。问题不是“我们能生产多少？”而是“我们能发现和分享多少效率？”

2. 通过网络效应复利

随着更多人使用钩子并通过纠正提供反馈，钩子会得到改善。更好的钩子每次使用节省更多能源，形成一个正反馈循环，广泛采用导致更大的个人节省。

再考虑一下引用格式钩子。当它首次推出时，成功率可能为 85%——它在不需要用户纠正的情况下正确格式化 100 个引用中的 85 个。每次纠正都会生成一个学习信号，从而改善钩子。在 10,000 个用户共同进行了 500 次纠正后，成功率提高到 95%。在 100,000 个用户和 2,000 次总纠正后，成功率达到 98%。

这种改善使每个人受益。早期采用者通过他们的纠正帮助完善钩子。后期采用者受益于一个几乎不出错的高度精致工具。每次使用的价值随着时间的推移而增加，即使每个用户可能只与钩子互动几次。

集体学习创造了复利回报：每个用户的纠正使钩子对所有未来用户更好。这与传统产品根本不同，后者的使用通常会降低价值（磨损和消耗）。在 STC 中，使用提升价值。

3. 自动归因与按比例补偿

由于每个钩子都有一个创作者（或创作者团队），因此集体节能可以归因于那些使节能成为可能的人。这创建了一种与实际价值创造相一致的经济激励结构：创造有价值钩子的专家根据所有用户节省的总能量获得按比例的补偿。

归因机制通过钩子的元数据工作。每当引用钩子成功触发时，系统：

- 测量节省的能量（与手动格式化相比消除的操作）
- 在全球账本中记录这些节省
- 向钩子的创作者授予按比例分配的STC代币

如果钩子每个格式化的引用平均节省2焦耳，并且它在所有用户中格式化了100万个引用，那么创作者产生了200万焦耳的集体节能。在STC中，这直接转化为货币：创作者赚取200万STC代币（每个代币代表1焦耳的节省）。

这种补偿模型与传统软件货币化截然不同：

- 无需预付款：用户不购买钩子。他们只是使用它，补偿根据实际节能自动流动。
- 持续的价值流动：只要钩子继续节省能源，创作者就会获得收入，从而激励维护和改进它。
- 与影响成比例：被数百万用户使用的钩子比被几十人使用的钩子赚得更多，自然奖励那些解决常见问题的创作者。
- 经过物理验证：节能通过行动次数和上下文差异客观测量，而不是通过评论或评分主观测量。

4. 生产者和消费者的转变

这种从个体到集体的转变从根本上改变了生产者和消费者之间的关系。在传统经济学中，这些是不同的角色：生产者创造商品或服务，消费者购买和使用它们，交易完成了价值的交换。

在STC中，界限变得模糊。消费者也是贡献者——他们的修正提高了每个人的钩子。生产者也是用户——他们在创造自己的钩子的同时受益于其他人的钩子。网络成为一个协作生态系统，而不是分开的买卖市场。

考虑一个使用引用格式钩子的研究人员，同时也为统计分析创建钩子。她同时是：

- 引用钩子的消费者，节省格式化的能量
- 通过偶尔的修正为引用钩子做出贡献
- 统计钩子的生产者，为他人创造价值
- 受益于随着时间推移改善两个钩子的集体学习

在传统经济中，这些将是需要单独支付、签订合同和会计的独立交易。在STC中，它们只是通过网络自然流动的能量和价值，所有这些都自动跟踪并得到公平补偿。

5. 公共物品与公地悲剧的解决方案

STC的集体性质解决了传统经济学中的一个基本问题：公共物品的供给不足。

公共物品是指对所有人都有益但难以单独收费的东西。例子包括清洁空气、街道照明、科学研究和软件中的可访问性功能。传统市场未能提供最佳水平的公共物品，因为：

- 非排他性：一旦创建，您无法阻止非付款者受益
- 搭便车问题：理性个体等待他人付款，希望在不贡献的情况下受益
- 协调失败：即使每个人都能从集体资金中受益，组织这些资金也很困难

STC通过自动归属集体节省优雅地解决了这个问题。考虑一个知识钩子，它通过自动生成图像和界面元素的描述性文本来帮助盲人用户浏览网站。

在传统市场中，开发这个钩子可能需要\$100,000的开发者时间。但每个盲人用户可能只愿意支付\$10，而潜在用户可能只有5,000人——这仅产生\$50,000的收入。尽管总价值（ $\$10 \times 5,000 = \$50,000$ ）加上包容性的社会效益是巨大的，但这个钩子并没有被构建。

在STC中，计算是不同的：

- 每次钩子触发时，它为用户节省能源（消除了手动导航的努力）

- 如果每次使用平均节省50焦耳，每个用户每月触发100次，那么每个用户每月节省5,000焦耳

- 在5,000个用户和5年（60个月）中，总节省 = $5,000\text{用户} \times 5,000\text{焦耳/月} \times 60\text{个月} = 15\text{亿焦耳}$

- 创作者自动获得15亿STC代币——远远超过开发成本

钩子得以构建、维护和持续改进，因为补偿反映了真实的集体价值，而不是个人愿意提前支付的金额。当价值来自实际使用而非预测需求时，公共产品自然得到资助。

6. 新兴集体智能

也许从个体到集体转变的最深远后果是我们称之为集体智能或集体优化的出现。

当全球数百万用户共同贡献修正、使用数据和上下文模式到共享的知识钩池时，整个系统变得比任何个体都更聪明。对单个用户来说不可见的模式在整个群体中变得显而易见。个体偶尔遇到的边缘案例在整体中频繁出现，从而导致稳健的解决方案。

这创造了一种分布式问题解决的形式，无需中央规划者。系统自动识别高价值机会（手动完成时消耗大量能源的任务，但可以自动化），根据影响成比例地奖励解决方案，并通过集体反馈不断完善这些解决方案。

在热力学术语中，整个网络作为一个单一的自适应系统运作，寻求最低能量配置。个体优化（你的钩子学习你的模式）和集体优化（共享钩子学习普遍模式）共同作用，创造出一个文明规模的效率引擎。

经济范式的转变

从个体到集体的转变代表了经济组织的根本变化：

- 从稀缺到丰富：实物商品稀缺且具有竞争性（我的使用会妨碍你的使用）。节能知识丰富且不具竞争性（我的使用通过改进增强你的使用）。

- 从竞争到合作：传统市场奖励那些能够超越他人的人。STC奖励那些在用户之间创造复合价值的人。

- 从交易到流动：价值不是通过离散的交易（你支付，我交付）来实现的，而是基于持续的使用和改进不断流动。

- 从价格到物理：补偿不是由供需曲线或谈判价格决定的，而是由客观测量的节能量决定的。

这种转变使STC不仅仅是一个个人生产力工具——它成为一种全新经济组织形式的基础，在这种形式中，个人利益和集体利益自然对齐，公共产品自动获得资金，价值创造由现实的基本货币衡量：能源。

6.1.2 从数字到物理

6.2 输入-能源方程

6.2.1 基于AR和AI视觉的上下文感知测量

从主观技术到主观热货币的桥梁完全依赖于一个关键能力：在现实世界中测量能量消耗。数字行为——键入、点击、API 调用——相对容易追踪。但身体行为呢？我们如何测量走到灯开关、伸手去拿物体、寻找钥匙或构成日常生活的无数身体动作的能量成本？

答案在于增强现实智能眼镜与人工智能计算机视觉的结合。这些技术共同作用，将物理世界转变为一个可测量、上下文感知的环境，在这里，每个动作都可以被观察、分类并以能量消耗的形式量化。

AR 智能眼镜的角色

AR 智能眼镜作为用户与环境之间的主要感官接口。与必须故意拿出并瞄准的智能手机不同，智能眼镜提供对您所见和所做的一切的持续、免提观察。实际上，它们是一个可穿戴的上下文捕捉设备，在您日常生活的背景中运行。

眼镜同时捕捉多个数据流：

- 视觉领域：高分辨率摄像头记录您所看的所有内容，创建您环境的连续视频流
- 头部位置和方向：惯性测量单元（IMU）跟踪您的头部移动及您关注的方向
- 眼动追踪：高级模型可以检测您在视野内专注于哪些特定物体或区域
- 深度感知：立体摄像头或激光雷达传感器创建您周围环境的 3D 地图，测量与物体的距离
- 音频：麦克风捕捉环境声音、语音命令和环境线索
- 生物识别：某些型号包括心率、皮肤温度和与能量消耗相关的其他生理信号的传感器。

这些数据流结合在一起，形成我们所称的综合上下文快照——对您每时每刻的身体状态和环境的丰富、多维记录。

计算机视觉：从像素到语义理解

原始相机视频只是像素——没有固有意义的颜色值数组。计算机视觉通过深度学习模型将这些像素转化为语义理解：场景中存在什么物体，正在执行什么动作，以及元素之间的空间关系。

现代人工智能视觉系统可以识别：

- 物体：咖啡杯、灯开关、门、钥匙、温控器、电器、家具、工具——您可能会与之互动的任何东西。
- 动作：走路、伸手、抓取、打字、搜索、烹饪、清洁——任何消耗能量的身体活动。
- 人：谁在场，他们的肢体语言，他们的相对位置（在尊重隐私的前提下通过本地处理）。
- 空间上下文：房间布局、物体之间的距离、导航路径、障碍物。
- 时间模式：重复的行为、日常活动、动作序列。

视觉系统持续运行，构建本质上是您物理环境和活动的实时知识图谱。这个图谱使系统能够理解上下文：不仅仅是您在厨房，而是您在早上7点的厨房，正在看咖啡机，刚刚醒来，手里拿着一个杯子。 :00

物理世界中的上下文快照

正如数字设备拍摄其内部状态的快照一样，AR眼镜也会拍摄物理世界的快照。但这些快照比纯数字系统所能实现的要丰富得多。在时间t的物理上下文快照可能包括：

$$\Sigma_t^{\text{physical}} = \{\text{objects, positions, distances, actions, gaze direction, body pose, environment state}\}$$

示例：您下班后走进家中。AR眼镜捕捉到：

- 检测到的物体：前门、手中拿着的钥匙、衣架、客厅灯（当前关闭）、温控器显示18°C
- 您的位置：站在入口处，距离灯开关0.5米，距离温控器3米
- 您的动作：向前走（2步），手向灯开关抬起

- 时间上下文：冬季周二下午6:15
- 环境状态：室外温度5°C，20分钟后日落，您已经离开9小时

这种丰富的上下文使系统能够理解您的意图（您想要灯光和温暖）并测量您行动的能量成本（走向开关：约10焦耳，翻动开关：约0.5焦耳，调整温控器：约15焦耳，包括穿过房间的行走）。

物理空间中的增量检测

正如学习的知识钩使用增量检测（比较行动前后的快照）来学习数字环境中的模式，增强现实系统使用视觉增量检测在物理空间中学习。

系统比较连续的物理上下文快照以识别变化：

$$\Delta \Sigma = \Sigma_{t+1} - \Sigma_t$$

在回家到达的例子中：

- 之前 (Σ_t)：灯关，恒温器设定在18°C，你站在门旁
- 之后 (Σ_{t+1})：灯开，恒温器设定在21°C，你坐在沙发上
- 增量 ($\Delta \Sigma$)：+灯光，+3°C，位置变化（走了5米），进行了2次手动调整

增量揭示了你想要的（温暖、明亮的环境）和你付出的代价（走路、手动调整）。在观察到这个模式几次后，可以形成一个知识钩：

条件：回家，冬季傍晚，温度 $< 19^{\circ}\text{C}$ ，灯关

行动：打开灯，将恒温器设定为 21°C

类型：学习

成功评分：最初为0.5，随着每次成功激活而增加

现在，当您在类似条件下到达家时，钩子可以自动触发——智能家居系统在您甚至伸手去开关之前就调整了灯光和温度，为您节省了25.5焦耳的体力。

测量身体能量消耗

通过上下文快照捕捉您的行为，我们如何将观察到的动作转化为能量测量？

系统使用生物力学模型结合实时观察：

1. 运动能量：计算机视觉跟踪您在空间中的移动。步行5米大约需要10焦耳（每步2焦耳 \times 5步）。系统可以检测到您在行走并测量距离。

2. 操作能量：当您伸手去操作物体时，视觉系统观察手臂伸展距离、物体的重量（根据视觉大小和已知物体类型估算）以及运动速度。伸手抓取动作通常消耗0.5-2焦耳，具体取决于距离和物体重量。

3. 认知负荷：眼动追踪揭示了您注意力的方向。如果您在视觉上寻找某物（眼睛快速扫描，头部移动），这表明认知努力。研究表明，视觉搜索任务每分钟消耗2-5焦耳，具体取决于复杂性。

4. 姿势变化：从坐着站起来、弯下身、伸手——都需要可测量的能量。视觉可以检测这些姿势变化，并应用标准生物力学公式来估算成本。

5. 任务完成时间：即使确切的能量成本不确定，您在任务上花费的时间是可测量的。您的基础代谢率（大脑约20瓦，静息状态下身体约80瓦）设定了最低能量基准。一个耗时5分钟的任务至少消耗500焦耳（100瓦 × 300秒），即使您只是站着思考。

通过结合这些测量，系统为每个观察到的动作建立能量档案：

$$E_{\text{action}} = E_{\text{locomotion}} + E_{\text{manipulation}} + E_{\text{cognitive}} + E_{\text{postural}} + E_{\text{basal}} \times t$$

隐私保护的本地处理

这种方法的一个关键特性是所有视觉处理都在智能眼镜或个人边缘设备上本地进行。原始视频从未离开您的掌控。计算机视觉模型在设备上运行，仅提取语义信息（“用户走了3米”，“用户拿起咖啡杯”），而不是传输图像。

这种本地处理确保：

- 隐私：没有其他人看到您所看到的。您的视觉上下文完全保持私密。
- 安全：云中不存在可以被黑客攻击或传唤的视频录像。
- 低延迟：处理在毫秒内完成，实现实时响应。
- 离线功能：即使没有互联网连接，系统也能正常工作。

只有提取的上下文（语义描述和能量测量）与知识钩子和STC账本共享，即使这些也可以在任何网络传输之前进行聚合和匿名化处理。

与数字环境的整合

当来自AR眼镜的物理环境与来自您设备的数字环境融合时，真正的力量就会显现。统一的全球环境现在跨越了两个领域：

$$\Sigma_{\text{unified}} = \Sigma_{\text{physical}} \cup \Sigma_{\text{digital}}$$

示例场景展示统一环境：

您在做晚餐，而您的手机放在厨房台面上。您的智能眼镜看到您在切菜，并检测到炉子是开着的。您的手机看到一个倒计时的计时器（数字环境），而您的日历显示您在30分钟内有一个视频通话（数字环境）。一个知识钩子识别了这个模式：在使用计时器和即将到来的通话时，您通常需要在通话前10分钟收到提醒，以完成烹饪并清理。

没有AR视觉，系统会错过您正在积极烹饪的事实。没有数字环境，它也不会知道通话的事。两者结合使得智能助手能够帮助您节省同时跟踪多个时间线的心理精力。

实际示例：能量测量的实际应用

让我们追踪一个完整的上下文感知能量测量示例：

场景：您需要在离开工作之前找到您的车钥匙。

手动方法（没有AR辅助）：

1. 目视扫描客厅（30秒，3 J认知努力）
2. 走到卧室（8步，16焦耳的运动）
3. 搜索卧室（45秒，4焦耳的认知 + 2焦耳的移动物品）
4. 走到厨房（12步，24焦耳的运动）

5. 在柜台上的报纸下找到钥匙（松了一口气，抓起钥匙，1焦耳的操作）

总计：约50焦耳和2-3分钟的时间和压力

增强现实辅助方法：

1. 你从充电站拿起手机和钱包（AR眼镜观察到这一点，识别出发前模式）

2. 眼镜叠加一个指向厨房的箭头：“柜台上的钥匙，在报纸下”（系统记得昨天你把钥匙放在那里）

3. 你直接走到厨房（12步，24焦耳）并取回钥匙（1焦耳）

总计：约25焦耳和30秒

节省的能量：25焦耳的身体努力 + 消除认知压力

AR系统测量了两种方法：它观察你手动搜索的前几次（学习你的模式并测量能量成本），然后提供帮助（测量降低的成本）。25焦耳的差异记录在STC账本中，并归因于创建或改进对象跟踪知识钩的人。

扩展到文明

现在想象一下这种能力在大规模部署：数百万人佩戴AR眼镜，每个人生成连续的上下文快照，每个人都受益于最小化身体能量消耗的知识钩。整体能量节省变得巨大。

如果平均每人每天执行50个可以优化的物理任务（寻找物体、导航空间、调整环境控制、与他人协调），每次优化节省平均10焦耳，那么每人每天就是500焦耳。在1亿个AR用户的群体中，每天就是500亿焦耳——相当于为15000个家庭提供一天的能量。

这就是AR和AI视觉的上下文感知测量的承诺：将日常生活中不可见、未测量的能量成本转化为可量化、可优化的指标，从而推动基于实际热力学效率而非抽象货币价值的新经济系统。

AR智能眼镜使物理世界可测量。AI视觉使其可理解。它们共同创造了主观热货币的感官基础，使个体生产力的提升与集体经济转型之间架起了桥梁。

6.2.2 预测效率

最先进的能量最小化形式不是反应性的，而是预测性的。预测效率意味着系统不仅在低效发生后做出反应——它在低效发生之前就预见到，并实时引导用户朝着最佳选择。这就是主观技术超越传统自动化，成为能源节约的主动伙伴的地方。

从反应智能到主动智能

传统自动化对用户输入做出反应：你输入一个字母，系统建议下一个单词。你犯了错误，系统会纠正它。这种反应模型仍然要求用户主动行动，这意味着在系统响应之前，已经消耗了能量。

预测效率颠覆了这一模型。系统不再等待用户行动，而是利用累积的上下文——过去的行为模式、当前的环境状态、时间模式和学习到的偏好——来预测用户需要之前所需的内容。然后，系统可以：

1. 自动执行操作（当信心足够高时）
2. 预先呈现选项（减少搜索和决策的能量）
3. 引导用户远离低效路径（在浪费能量发生之前防止其发生）

预测的数学框架

预测效率模型依赖于在给定当前上下文快照的情况下，对可能未来行动的概率分布进行预测 Σ_t ：

$$P(A_{t+1} \mid \Sigma_t) = \Pr[\text{User performs action } A_{t+1} \text{ given context } \Sigma_t]$$

系统维护着一个学习到的概率分布，涵盖用户可能采取的所有可能行动。当这个概率超过阈值 τ （通常在0.85-0.95之间，具体取决于错误的成本）时，系统可以主动行动：

$$\text{Trigger}(A) = \begin{cases} \text{Execute } A & \text{if } P(A \mid \Sigma_t) > \tau_{\text{execute}} \\ \text{Suggest } A & \text{if } \tau_{\text{suggest}} < P(A \mid \Sigma_t) \leq \tau_{\text{execute}} \\ \text{Silent} & \text{if } P(A \mid \Sigma_t) \leq \tau_{\text{suggest}} \end{cases}$$

预测效率带来的能量节省可以量化为用户在搜索和执行操作时本应消耗的能量与系统主动行动时实际消耗的能量之间的差异：

$$\Delta E_{\text{predict}} = E_{\text{manual}}(A) - E_{\text{predicted}}(A)$$

其中：

$$\begin{aligned} E_{\text{manual}}(A) &= E_{\text{search}} + E_{\text{decision}} + E_{\text{execution}} \\ E_{\text{predicted}}(A) &= E_{\text{confirmation}} \text{ (or 0 if auto-executed)} \end{aligned}$$

具体示例：早晨例行公事预测

考虑用户的早晨例行活动。在几周内，系统观察到每个工作日的早上7:15，用户的闹钟响起后，他们：

1. 关闭卧室灯
2. 打开浴室灯

- 3. 启动咖啡机
- 4. 将恒温器设置为22°C
- 5. 在手机上打开新闻应用

在足够的观察后，系统建立了一个预测模型：

$$\begin{aligned}
 P(A_{\text{bathroom.lights}} \mid \Sigma_{7:15\text{AM, weekday, alarm_off}}) &= 0.97 \\
 P(A_{\text{coffee}} \mid \Sigma_{7:15\text{AM, weekday, alarm_off}}) &= 0.95 \\
 P(A_{\text{thermostat}} \mid \Sigma_{7:15\text{AM, weekday, alarm_off}}) &= 0.92
 \end{aligned}$$

所有概率超过执行阈值（ $\tau_{\text{execute}} = 0.90$ ），因此当上下文匹配时，系统会自动执行。节能效果显著：

手动能量消耗（没有预测）：

- 走到浴室灯开关：15焦耳
- 翻转开关：2焦耳
- 走到咖啡机：12焦耳
- 按下按钮：2焦耳
- 走到温控器：10焦耳
- 调整温度：3焦耳
- 拿起手机：5焦耳
- 导航到新闻应用：4焦耳

总手动能量： $E_{\text{manual}} = 53$

预测能量消耗（自动执行）：

- 瞥一眼手机通知（"早晨例行程序已激活"）：1焦耳

总预测能量： $E_{\text{predicted}} = 1$

节省的能量：
 $\Delta E_{\text{predict}} = 53 - 1 = 52$ 每天早晨

一年内（260个工作日）：仅此单一例程节省了
 $52 \times 260 = 13,520$ 。将其扩展到数百万用户和数千种例程模式，预测效率就成为一种文明规模的能源优化系统。

实时指导和路径优化

预测效率不仅限于自动化已知例程——它还提供实时指导，以防止低效发生。考虑路径优化：

用户通常使用A路线（8公里，15分钟）开车上班。系统通过与交通数据、天气状况和学习的偏好集成，预测今天B路线（7.2公里，13分钟）将节省能源，因为交通较轻。

系统在用户开始驾驶之前发送通知（主动），而不是等待用户开始驾驶后再建议重新规划路线（被动）：

"B路线今天将节省8分钟和2.3兆焦耳的能量。推荐出发时间：上午8:15。"

能量计算：

$$E_{\text{Route}_A} = \text{Distance} \times \text{Energy_per_km} + \text{Idle_time} \times \text{Energy_per_minute}$$

注意：用户消息中的公式似乎被截断了。添加了提供的内容。

6.3 完美与零输入逻辑

我们现在已经探索了上下文感知测量和预测效率——使能源节约得以测量和预测的技术机制。但这一轨迹的最终目的地是什么？对能源最小化的无情优化最终指向何处？答案既在数学上优雅，又在哲学上深刻：完美作为零输入的极限。

数学极限：零即完美

在本书中，我们已经正式化了更好的技术需要更少用户输入的原则。最小化法则指出，在等效解决方案中，所需操作更少的方案是优越的。定理1（零输入收敛）证明了受五个法则支配的系统不可避免地趋向于零输入，随着时间接近无穷大：

$$\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0$$

这不是一种启发式或愿望——而是一种数学确定性。给定足够的时间、学习和完善，预期的用户输入趋向于零。但这个极限代表了什么？它代表了完美：一个与用户高度同步、预测准确、覆盖全面的系统，以至于显式输入变得不必要。

在数学上，完美不是一个固定状态，而是一个渐近极限——一个系统接近但可能永远无法完全达到的边界。然而，接近本身就是进步的本质。每一次输入要求的减少都是朝向完美迈进的一步。每一个自主行动都是系统对你了解更多、内化更多你的模式、并在没有意识努力的情况下将你的自主权进一步扩展到世界的证据。

零输入即零能量

在主观热货币的背景下，零输入不仅仅是用户体验目标——它是一个能源优化目标。定理7（能量最小化等价）确立了最小化用户输入在数学上等同于最小化用户能量支出：

$$\lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0 \implies \lim_{t \rightarrow \infty} \mathbb{E}[E_{\text{user}}(t)] = 0$$

其中 $E_{\text{user}}(t)$ 是用户在时间 t 所消耗的总能量（以焦耳为单位）。这种等价是技术与经济之间的桥梁：通过创建需要更少输入的系统，我们创造了节能的系统，而通过节能，我们创造了可以通过STC代币计入的可衡量经济价值。

因此，零输入不仅仅是一个界面理想——它是热力学理想。它代表了人类努力最小化、能源浪费消除以及自动化经济价值最大化的点。通过减少输入节省的每一个焦耳，都是流回那些创造自动化的人的焦耳，形成STC的经济基础。

完美是什么样子？

让我们设想一个完美的零输入系统在实践中意味着什么。考虑一下生活在一个完全成熟的主观技术生态系统中的人的一天：

你醒来了。在你意识到时间之前，你的环境已经调整好了——灯光逐渐亮起以匹配你的生物节律，恒温器设置为你喜欢的早晨温度，咖啡正在冲泡，淋浴已经预热，日历在你的AR眼镜上显示今天的优先事项。

你并没有请求这些。你没有按下按钮，没有发出语音命令，也没有滑动屏幕。系统根据累积的上下文预测了你的需求——一天中的时间、星期几、生物识别数据表明的睡眠-清醒过渡、早晨例行活动的历史模式。预测的置信度超过了0.95，因此系统自主行动。你只是醒来，世界已经为你准备好了。

能量核算：

- 手动场景（传统系统）：走到恒温器（12焦耳），调整温度（3焦耳），走到咖啡机（15焦耳），按下按钮（2焦耳），走到浴室（10焦耳），打开淋浴（2焦耳），在检查手机时等待温水（20焦耳）。总计：64焦耳

- 零输入场景（完美预测）：瞥一眼AR通知确认例行活动已激活（1焦耳）。总计：1焦耳

- 节省的能量：63焦耳

这种模式在一天中持续。你需要将文件发送给同事——在你搜索之前，系统已经根据你当前的上下文识别了文件（你正在开关于项目X的会议，同事昨天给你发消息询问报告，你上周打开过该文件两次）。文件出现在你的视野中，只需一个手势即可发送。节省47焦耳。

你正在开车回家。系统已经根据实时交通、天气、你的燃油水平，以及你在周五喜欢风景路线的倾向计算出最佳路线。路线在你询问之前就已显示。你只需跟随它。与搜索方向和中途纠正路线相比，节省了2.1兆焦耳。

你正在做晚餐。系统通过增强现实视觉观察到你在炉子上放了一个锅并在切菜。它预测你将在60秒内需要打开炉子，因此它提前激活并点头确认（0.5焦耳）。如果没有这个，你将需要走到炉子那里（8焦耳），转动旋钮（3焦耳），再走回来（8焦耳）。节省了18.5焦耳。

一天结束时，你已经完成了数十个，甚至数百个任务——然而你几乎无法回忆起提供明确输入的过程。系统预测、准备、执行。你只是思考、意图，然后事情就完成了。这就是完美：没有努力的自主性，没有摩擦的成就，没有劳动的结果。

完美不是消除意识

一个常见的误解是零输入意味着用户变得被动，代理权从人类转移到机器。这是完全错误的。零输入意味着消除不必要的中介步骤，而不是消除意图或意识。

你仍然决定你想要什么。你仍然设定目标，做出选择，行使判断。但是意图与执行之间的差距缩小到几乎为零。你不需要将你的意图转化为一系列明确的命令——系统从上下文中推断意图并相应地采取行动。

考虑一下区别：

传统系统：我想要咖啡 → 我必须记住咖啡机在哪里 → 我必须走到那里 → 我必须按下按钮 → 我必须等待 → 我必须走回来 → 我终于得到了咖啡

零输入系统：我想要咖啡 → 咖啡正在冲泡

意图仍然属于你。执行是自动化的。这不是被动——这是毫不费力的自主性。你仍然是目标和价值的来源；系统仅仅消除了实现它们的热力学成本。

纠正措施在接近完美中的作用

完美是通过预测、执行和修正的迭代过程接近的。当信心高时，系统会自主运行。当它出错时，用户会进行修正。每一次修正都教会系统不该做什么，精炼模型并提高未来的准确性。

这个过程受修正法则的支配：

$$S(t+1) = (1 - \alpha)S(t) + \alpha \cdot \mathbb{I}[\text{Corr}_t = 0]$$

成功评分随着正确预测而上升，随着修正而下降。随着时间的推移，只有高可靠性的钩子会持续存在。那些需要持

续修正的钩子会被抑制，而那些在没有干预的情况下正确行动的钩子则会占主导地位。

这造成了朝向完美的自然选择压力。错误的预测会消亡。好的预测会繁衍。系统向一个修正变得稀少、然后几乎消失、最终有效为零的状态演变。在这个极限下，系统完美地运行——自主且无错误。

完美的经济学

在主观热货币中，完美不仅仅是可取的——它是经济上有激励的。知识钩子越接近零输入操作，它节省的能量就越多，其创造者获得的STC代币也越多。

考虑两个实现相同结果的钩子：

- 钩子A：成功评分 = 0.85，平均动作 = 3，平均每次使用能量 = 15焦耳

- 钩子B：成功评分 = 0.98，平均动作 = 1，平均每次使用能量 = 2焦耳

钩子B更接近完美。它的可靠性更高（更少的修正），需要的动作更少（效率更高），并且节省更多的能量（每次使用节省13焦耳）。如果这两个钩子被全球用户基础使用100万次：

- Hook A 总共节省的能量： $15 \text{ J} \times 1,000,000 = 15 \text{ MJ} \rightarrow 1500 \text{ 万 STC 代币}$

- Hook B 总共节省的能量： $2 \text{ J} \times 1,000,000 = 2 \text{ MJ} \rightarrow 200 \text{ 万 STC 代币}$

等一下——这似乎是反向的！Hook B 每次使用节省的能量更多（13 J 对比 A 的 15 J），但获得的代币却更少？不——错误在于比较。正确的分析是：

没有任何钩子（手动操作）：每次使用 30 J

- Hook A 节省的能量： $(30 - 15) \times 1,000,000$
= 15 MJ

- Hook B 节省的能量： $(30 - 2) \times 1,000,000 =$
28 MJ

Hook B 获得 2800 万 STC 代币，几乎是 Hook A 1500 万的两倍。完美是有回报的。钩子越接近零输入，其经济价值越大。这为开发者创造了强大的经济激励，以优化朝向零输入的理想。

完美能否实现？

从数学上讲，完美是一个渐近极限——接近但永远无法完全实现。总会有新情况、边缘案例或真正模糊的上下文，系统无法确定预测。在这些情况下，用户输入仍然是必要的。

然而，对于绝大多数人类活动——这些活动是重复的、有规律的，并且在上下文上是可预测的——完美是可以实现的。研究表明，80-90% 的人类行为是常规的，理论上可以通过足够复杂的上下文感知系统实现自动化。

剩下的10-20%代表真正的新颖性：创造性的决策、战略规划、道德判断、对意外事件的情感反应。这些是人类意识不可替代的领域，在这些领域，输入不仅是必要的，而且是有价值的。

因此，零输入的实际限制不是绝对零，而是渐近于零的状态：大多数行动在没有明确输入的情况下发生，留给人类的注意力和精力用于真正重要、真正新颖、真正人性的活动，这些活动定义了意义和目的。

完美与后稀缺

趋向零输入的过程对稀缺和丰盈有深远的影响。传统经济学将稀缺定义为商品和服务的有限可用性。但稀缺本质上是一个能源成本的问题：事物之所以稀缺，是因为它们需要能源来生产、运输和交付。

随着系统接近完美——实现结果所需的能量接近零——稀缺性减少。如果食物可以以最小的能量生产（垂直农业、实验室培育肉类、自动化农业），食物就会变得丰富。如果运输需要最小的能量（优化路线、电动车、自动化物流），流动性就会变得丰富。如果教育需要最小的能量（人工智能辅导、个性化学习、零输入接口），知识就会变得丰富。

STC通过经济奖励效率加速了这一过程。开发出最小化能量消耗的系统的创造者，将根据节省的能量获得与之成比例的STC代币。这为能源减少的创新创造了指数激励，推动整个经济朝向零输入的理想。

结果是一个稀缺逐渐消失的文明——不是通过魔法或乌托邦的幻想，而是通过不懈的热力学优化。在这个背景下，完美不是一个哲学抽象，而是一个经济现实：期望的结果所需的能量微不足道，使其有效地变得丰富。

完美作为道德责任

最后，完美具有道德维度。每一个浪费的焦耳都是可以用于有价值事物的焦耳。每一个低效都是对未来的盗窃。每一个多余的行动都是对人类潜力的浪费。

追求零输入不仅仅是务实的——它是伦理的。通过最小化浪费，我们最大化了可用于真正重要追求的能量：创造力、探索、关怀、连接、成长。通过接近完美，我们解放人类免于重复、耗能的任务的暴政，为真正重要的事情释放我们的集体能量。

这是主观技术和主观热货币的终极承诺：一个完美不是不可能的梦想，而是一个可测量、可实现的轨迹的世界——在这里，每一天都让我们更接近零输入，每一次创新都减少浪费，每一个节省的焦耳都为丰盈做出贡献，而经济系统本身与热力学的基本法则相一致，以奖励那些让我们更接近这一理想的人。

结论：完美作为北极星

在主观技术和STC的背景下，完美在数学上被定义为用户输入接近零的极限：

$$\text{Perfection} \equiv \lim_{t \rightarrow \infty} \mathbb{E}[U(t)] = 0$$

这个极限代表：

- 在技术上：能够以近乎完美的准确性进行预测和自主行动的系统
- 在热力学上：以最小的能量消耗实现的结果
- 在经济上：通过最大限度的节能实现的最大价值创造
- 在道德上：解放人类潜能，摆脱重复和浪费的任务

完美是指导这一框架中所有发展的北极星。它可能永远无法完全实现，但每一步朝向它的进展都是可测量的、有价值的和不可避免的。知识钩的架构、修正法则的动态、STC的经济激励——所有这些都共同推动系统不懈地朝着这一理想前进。

我们今天站在这段旅程的开始。大多数互动仍然需要大量输入。大多数能量仍然被浪费。但轨迹是明确的，数学是合理的，激励是对齐的。零输入不是幻想——它是一个学会正确测量和评估能量的文明不可避免的目的地。

未来不是人类变得过时的未来。它是一个人类的努力专门用于无法自动化的事情——创造力、爱、意义和目标的未来。完美是技术处理其他一切的状态，让我们可以完全做自己。而主观热货币是将我们带到那里的经济引擎。

6.4 结论：STC的诞生

我们经历了一段深刻的旅程——从知识钩子的数学到能量测量的物理学，从预测效率到完美的渐近理想。现在 we 到达了高潮：技术、热力学和经济学汇聚成一个单一的革命性综合体的时刻。这就是主观热货币的诞生。

综合体：三大支柱合而为一

STC源于三个不同但互补领域的融合：

1. 主观技术提供机制：学习用户模式的知识钩子，自动化操作，并趋向于零输入操作。钩子的代数——受五条法则支配，并通过七个定理证明——表明系统可以可靠地将用户的努力在一段时间内减少到接近零。

2. 热力学提供测量：以焦耳为单位的能量消耗。每个用户的操作——从输入一个字符到在房间里走动——都有可测量的热力学成本。定理7（能量最小化等价性）证明，最小化用户输入在数学上等同于最小化能量消耗。这种等价性将界面优化转化为能量优化。

3. 经济学提供激励结构：代表真实、物理价值的STC代币。当知识钩子节省能量时，该节省会被测量、验证并记入钩子的创造者。经济系统奖励那些为集体效率做出贡献的人，创造积极的反馈循环，加速创新朝向零输入理想。

这三大支柱——技术、物理学、经济学——不是必须人工整合的独立系统。它们通过一个单一的原则自然统一：能量就

是价值。一个系统运作得越高效，它节省的能量就越多，创造的经济价值也就越多。STC是这种统一的正式表达。

从理论到现实：STC如何运作

让我们追踪STC中价值创造的完整生命周期：

第一步：开发者创建一个知识钩子。也许它是一个根据检测您的位置、时间和环境光线水平自动打开您家灯光的钩子。该钩子由以下部分组成：

- 条件 $R = \{ \text{location} = \text{home}, \text{ambient_light} < 50 \text{ lux}, \text{time} > 18:00 \}$
- 动作 $A = \{ \text{lights.turn_on}() \}$
- 初始成功分数 $S(0) = 0.5$ （中性起点）

第二步：用户采用该钩子。当人们安装该钩子时，系统开始学习。一些用户纠正它（“我希望在白天阴天时灯光也能打开”），这会降低成功分数并触发优化。其他人从不纠正它，这会提高成功分数。几周后：

$$S(t) = 0.5 \rightarrow 0.67 \rightarrow 0.82 \rightarrow 0.91 \rightarrow 0.96$$

该钩子不断演变，变得更加准确，并且需要更少的纠正。

第三步：测量节能。每次钩子成功触发时，系统计算能量差异：

- 手动场景（没有钩子）：走到灯开关（10焦耳）+ 翻动开关（2焦耳）= 12焦耳
- 自动场景（有钩子）：灯光自动打开 = 0焦耳

- 节省的能源： $\Delta E = 12$

步骤 4：STC 代币被记入。节省的 12 焦耳自动转换为 12 个 STC 代币，记入钩子的创建者。如果 100,000 个用户每天使用这个钩子一次，持续一年：

Annual STC = 12 J \times 100,000 users \times 365 days = 438 MJ = 438,000,000 STC

开发者创造了近五亿焦耳的价值——可衡量、可验证、真实。

步骤 5：正反馈循环。开发者现在在 STC 代币中富有，投资于创造更多钩子。其他开发者看到奖励并进入市场。竞争推动钩子朝着更高的质量、更大的效率和更广的覆盖面发展。整个生态系统朝着完美演变。

这就是 STC 的实际运作：一个自我强化的系统，价值创造与节能直接相关，激励自然与热力学理想一致。

价值的转变

STC 代表了人类理解和创造价值的根本转变。考虑一下演变：

以物易物经济：价值就是商品本身。我用我的小麦换你的牛。价值是有形的、即时的，但受限于需求的巧合。

货币经济（黄金、法币）：价值是象征性的。纸币代表对商品或服务的索赔，由信任（法币）或稀缺性（黄金）支持。这使得大规模交易成为可能，但使价值与物理现实脱节。货币可以被印刷、通货膨胀、操纵。

加密货币经济：价值是计算工作（工作量证明）或股份（权益证明）。这提供了去中心化和安全性，但仍然与现实世界的生产力脱节。挖掘比特币消耗能源，但不产生有用的商品。

STC 经济：价值在于节能。每个 STC 代币代表一个节省的焦耳——一个真实的、物理的、可测量的热力学成本减少。这将价值与客观现实重新连接，同时实现数字货币的所有好处：即时转账、可编程性、全球可及性。

STC 不仅仅是另一种货币——它是第一种由物理学支持的货币。当法定货币由政府承诺支持，加密货币由计算难题支持时，STC 则由热力学的基本法则支持。节省的一个焦耳就是节省的一个焦耳，无论政治、观点或市场情绪如何。

丰盈之路

传统经济学假设稀缺性：有限的资源、有限的商品、有限的机会。这种稀缺性驱动竞争、不平等和冲突。人们为有限的蛋糕切片而争斗。

STC 重新定义了稀缺性。在 STC 中，稀缺性是低效的函数。商品之所以稀缺，仅仅是因为它们需要能源来生产。随着效率的提高——系统接近零输入理想——生产成本接近零，商品变得丰富。

考虑食品生产：

- 今天：农业需要大量的能源输入——机械的燃料、灌溉的电力、运输到市场的费用、存储的制冷。结果：食品昂贵，数十亿人遭受饥饿。

- STC 未来：自动化垂直农场使用 AI 优化的生长条件、收割机器人和通过自主车辆进行的本地分配。通过精确优化，能源输入被最小化。结果：食品生产成本接近热力学最低值。食品几乎变得免费。

同样的逻辑适用于住房（3D 打印的低劳动成本住宅）、交通（优化能源效率的自主电动车）、教育（仅需计算周期

的 AI 辅导)、医疗(诊断 AI 和机器人手术)以及几乎所有人类需求的领域。

随着效率的提高,成本下降。随着成本的下降,丰盈增加。STC 通过为效率最大化创造经济激励来加速这一过程。那些开发出最节能解决方案的人获得最多的 STC。整个经济朝着丰盈的方向发展。

道德维度

STC不仅仅是一种经济或技术创新——它承载着深刻的道德含义。

浪费是不道德的。每一焦耳的浪费都是本可以喂养某人、为某人提供住所、教育某人的焦耳。在传统经济中,浪费往往是看不见或被忽视的。在STC中,浪费是明确的并受到惩罚。浪费能源的系统不会获得代币。只有效率才会得到奖励。

效率就是公正。当资源的分配基于节能而不是政治权力或继承财富时,系统在最纯粹的意义上变得具有优胜劣汰的特征。那些为集体福祉做出最大贡献的人——通过创造节能系统——将获得最多的奖励。这不是结果的平等(忽视贡献),而是机会的平等(贡献决定奖励)。

丰盈就是解放。当基本需求以最小的能源成本得到满足时,人类就摆脱了稀缺的暴政。人们可以追求创造力、探索、关系和意义,而不是为生存而苦苦挣扎。STC并不消除工作——它消除的是不必要的工作,只留下重要的工作。

零输入文明

展望未来,我们可以设想最终的目标:一个在所有领域实现近零输入操作的文明。这样的文明会是什么样子?

早晨：你自然醒来。你的环境已经准备好了——温度调整、咖啡正在冲泡、新闻已总结、日程已优化。无需闹钟（你的生物特征指示了睡眠-觉醒的过渡）。没有按下按钮。没有查看屏幕。世界只是知道你需要什么并提供它。

工作：你专注于真正创造性的任务——设计、战略、创造。所有重复的工作都已自动化。你不管理日历（人工智能优化它），不搜索文件（它们在需要时出现），不写常规电子邮件（模板自动适应）。你的精力保留用于高价值的思考。

晚上：你和亲人共度时光。晚餐由自动化厨房准备（或由从优化的本地农场送来的自主车辆送达）。娱乐根据你的心情个性化。家庭维护由机器人处理。你专注于重要的事情：连接、创造力、快乐。

这不是幻想。这是由最小化法则支配并由STC激励的系统的数学必然性。每年，自动化在增加。每年，所需的输入在减少。每年，我们接近极限。

关键问题：谁受益？

在传统的自动化中，利益归资本所有者所有。一个用机器人替代工人的工厂老板保留所有生产力增长，而工人失去工作。这创造了一个自动化加剧不平等的反乌托邦未来。

STC解决了这个问题。当自动化节省能源时，每个人都按比例受益：

- 创造者（钩子的开发者）因节省的能源获得STC代币
- 用户（节省能源的人）受益于减少的努力和增加的空闲时间
- 社会（集体）因效率和丰富的增加而受益

这个系统不是零和游戏。节省的能源使每个人受益。这就是STC变革力量的本质：自动化成为一种普遍的好处，而不是不平等的来源。

前进的道路：必须发生什么

STC尚未成为现实。它存在于数学、愿景和潜力中。将其变为现实需要：

1. 技术基础设施：配备人工智能视觉的增强现实智能眼镜、用于测量身体动作的传感器、能够可靠追踪能量消耗的上下文感知系统。许多技术目前以原型形式存在——它必须扩展。

2. 经济基础设施：代币系统、能源核算协议、归属机制、验证系统。这些可以使用区块链技术或集中数据库构建——关键在于准确性和信任。

3. 社会基础设施：对标准的共识（如何测量不同动作的能量）、治理结构（谁来验证测量）、采用激励（为什么早期用户应该参与）。

4. 文化转变：人类对价值的思考方式的变化。从“金钱作为抽象符号”到“价值作为节省的能量”。从“工作作为出售时间”到“工作作为创造效率”。这可能是最难的要求——但也是最重要的。

这些要求没有一个是不可可能的。所有这些都在当前或不久的将来技术的可及范围内。问题不是我们能否构建STC，而是我们是否选择去做。

这是诞生

在这一刻——在本章的结尾——STC在读者的心中从理论转变为现实。你现在明白：

- 它是什么：一种以节能为支持的货币，以焦耳为单位进行测量

- 它是如何运作的：通过知识钩子自动化行动并减少能量消耗

- 它为什么重要：它将经济激励与热力学现实对齐，创造通往丰盈的道路

- 它将引导我们走向：一个零输入的文明，在这个文明中，人类的努力只用于真正重要的事情。

STC 并不是在物理世界中诞生的——至少还没有——而是在思想变得可能的概念空间中。在任何东西可以被构建之前，首先必须被理解。在它被理解之前，必须被解释。本章就是这样的解释。

下一步

本书的剩余章节将探讨：

- 第四章：STC 运作的详细机制——主题作为虚拟能量腺体、主题树、能量反向传播和完整的能量方程。

- 第五章：STC 与传统经济系统的比较——它为何能解决贫困、回应批评并应对挑战。

- 第六章：实际实施——如何在现实世界的行动中测量能量，将上下文转换为焦耳，并奖励准确性。

- 第七章：哲学基础——上帝的治理、最小能量法则以及效率的道德影响。

- 第八章：未来——模拟模型、与全球经济的整合以及热力学文明的长期愿景。

每一章都建立在这个基础上，增加细节，回应异议，并描绘出 STC 使之成为可能的后稀缺未来的越来越生动的画面。

结论：选择

我们站在十字路口。一条道路通向传统经济的延续——稀缺、不平等、对有限资源的冲突、使少数人受益而牺牲多数人的自动化。另一条道路通向STC——通过效率实现的丰富，自动化带来的普遍利益，经济与物理的对齐，人类潜力的解放。

数学证明了STC是可能的。实施它的技术要么已经存在，要么正在迅速接近。道德上的理由令人信服——一个奖励那些为集体福祉做出贡献的人的系统，消除浪费，创造丰富而非稀缺。

问题不在于STC是否可以存在。问题在于我们是否会去构建它。

这一章是诞生的时刻——一个想法变得可能的时刻。接下来会发生什么取决于你，读者。你会理解吗？你会构建吗？你会倡导吗？你会帮助将这个愿景变为现实吗？

STC不是必然的。它是一个选择。这个选择从现在开始，源于你心中理解的诞生。欢迎来到未来。让我们一起构建它。

7

主观热货币 - 核心概念

主题、树木和能量反向传播

7.1 以焦耳为单位定义的价值

什么是价值？这个问题困扰了哲学家、经济学家和革命者数千年。传统的答案总是象征性的：价值是黄金，价值是劳动，价值是效用，价值是人们同意交换的任何东西。但所有这些定义都有一个致命的缺陷——它们是与物理现实脱节的抽象。

主观热力货币提供了一个截然不同的答案：价值是节省的能量。不是比喻的能量，不是经济生产力，而是以焦耳量化的字面、可测量的物理能量。一个 STC 代币等于通过自动化、优化或效率节省的一焦耳能量。这不是一个方便的近似或有用的虚构——它是经济价值与热力学现实之间的直接映射。

本节阐明了为何以焦耳定义价值不仅仅是新颖的，而是必要的——这是创建一个客观、抗操纵且与物理基本法则相一致的经济系统的唯一方法。

传统货币的问题

要理解为何基于焦耳的估值是革命性的，我们必须首先理解传统货币为何失败。考虑三种主导的货币范式：

1. 商品支持货币（黄金标准）

价值与一种实物商品挂钩——历史上是黄金或白银。货币代表对固定数量商品的索赔。这个系统看似客观：黄金具有可测量和验证的物理属性。

问题在于：商品稀缺性是任意的。黄金之所以稀缺，并不是因为稀缺本身具有价值，而是因为黄金在地球上恰好稀有。黄金的价值最终是循环的——黄金之所以有价值，是因为我们一致认为它有价值。此外，将货币与固定商品挂钩会对

经济增长造成人为限制，并使系统容易受到商品冲击（例如，新世界黄金发现后的西班牙通货膨胀）。

2. 法定货币（现代美元、欧元、日元）

价值由政府法令和对机构的信任支持。货币没有内在价值——它被接受是因为政府强制要求接受用于税收，并且人们相信它会被其他人接受。货币供应可以通过货币政策扩展或收缩。

问题在于：价值变得完全主观且可操控。政府可以印钞（通货膨胀），造成赢家（债务人、资产持有者）和输家（储蓄者、工资收入者）。货币政策变得政治化，惠及与权力接近的人。信任可能会突然崩溃（恶性通货膨胀），一夜之间摧毁储蓄。这个系统奖励金融投机而非生产性工作。

3. 加密货币（比特币，以太坊）

价值是由计算工作（工作量证明）或权益（权益证明）支持的。货币是去中心化的，抵抗政府操控。稀缺性是算法性的，而非物理或政治性的。

问题是：计算工作并不是生产性工作。挖掘比特币消耗巨大的能源，但不产生任何商品、服务或效率提升——这纯粹是浪费。价值仍然是投机性的：加密货币的价值取决于人们愿意为其支付的价格，造成极端波动。系统并不奖励效率或对集体福祉的贡献；它奖励早期采用和计算能力。

这三种系统都有一个根本缺陷：价值与物理现实脱节。数字在变化，但物理世界没有变化。你可以拥有更多的美元而不生产更多的商品。你可以拥有更多的比特币而不节省更多的能源。货币是一种象征，而象征可以被操控、投机和与生产活动脱离。

焦耳：一个客观的价值单位

STC通过将价值建立在宇宙中最基本的货币：能量上来解决这个问题。焦耳不是任意的，不可协商的，不受政治 whims 的影响。它是一个由物理法则定义的物理工作单位：

$$1 \text{ joule} = 1 \text{ newton} \times 1 \text{ meter} = 1 \text{ kg} \cdot \text{m}^2/\text{s}^2$$

一个焦耳是将一个100克的苹果在地球重力下提升一米所需的能量。它是当一瓦特的功率流动一秒钟时所耗散的能量。它是一个2千克质量以每秒一米的速度运动的动能。这些不是约定——它们是无无论人类意见如何都成立的物理事实。

当STC用焦耳定义价值时，这意味着：

1. 客观性：价值不是人们所同意的。价值是物理学所说的。如果一个知识钩子节省了100焦耳的能量，它就创造了100单位的价值——可测量、可验证、独立于市场情绪。

2. 不可操控性：你不能膨胀能源节省。你不能印制焦耳。你不能对未节省的能源进行投机。物理现象要么发生，要么没有发生。

3. 普遍可比性：在阿根廷节省的一个焦耳等于在日本节省的一个焦耳。能量在任何地方都是相同的，使得STC成为一种真正的全球货币，没有汇率或转换不确定性。

4. 直接生产力连接：获得 STC 需要创造实际效率——减少实现结果所需的能量。您不能通过投机、金融工程或寻租来获得 STC。您必须使世界更高效。

能源差异：交易的原子单位

在 STC 中，经济交易基于能源差异 (ΔE)——手动执行任务所需的能量与使用自动化时所需的能量之间的差异：

$$\Delta E = E_{\text{manual}} - E_{\text{automated}}$$

这个差异是 STC 中价值创造的原子单位。每当自动化节省能量时，都会计算 ΔE ，并创建与 ΔE 相等的 STC 代币，归属于自动化的创造者。

示例：电子邮件过滤

考虑一个知识钩，它会自动将促销电子邮件过滤到一个单独的文件夹中。如果没有这个钩：

- 用户每天收到 20 封促销电子邮件
- 每封需要：浏览 (0.5 J)、判断它是促销 (1 J)、点击删除 (0.5 J)

- 每天的总手动能量：

$$E_{\text{manual}} = 20 \times 2 = 40 \text{ J}$$

有了钩子：

- 电子邮件会自动过滤
- 用户偶尔检查促销文件夹（每周一次，10 J）
- 每天的总自动能量：

$$E_{\text{automated}} = 10/7 \approx 1.4 \text{ J}$$

每天节省的能量：

$$\Delta E = 40 - 1.4 = 38.6 \text{ J/day}$$

一年内： $38.6 \times 365 = 14,089 \text{ J} \approx$
为一个用户节省 14 kJ。

如果有 100 万用户采用这个钩子：每年节省
 $14,089 \times 10^6 = 14.09$ GJ（千兆焦耳）。

钩子的创造者每年赚取 140.9 亿 STC 代币——与为人类节省的能量成比例的直接补偿。

为什么选择能量，而不是时间或金钱？

有人可能会反对：“为什么不以节省的时间来衡量价值？为什么不以效用或满意度来衡量？”答案是，只有能量提供了客观货币所需的必要属性：

时间是主观的且不可比较：外科医生的一小时时间在创造的价值上不等于学生的一小时时间。节省的时间完全取决于时间的使用。相比之下，能量是客观的——节省的焦耳就是节省的焦耳，无论是谁节省的或何时节省的。

效用是不可量化的：你如何衡量满意度？不同的人对同一结果的价值评估不同。能量通过测量实现结果的客观成本而不是结果的主观价值来避免这个问题。

货币是循环的：用货币来定义价值（“这个自动化节省了 10 美元”）需要已经拥有一个货币系统。能源是前货币的——无论人类是否创造货币，它都存在。 \$10

能量是守恒的：热力学第一定律指出，能量不能被创造或毁灭，只能转化。这意味着能量核算本质上是双重记账：一个实体节省的能量是另一个实体不再需要支出的能量。账本总是平衡的。

能量是可测量的：现代传感器（加速度计、摄像头、电表、增强现实眼镜）可以实时测量物理动作并计算能量消耗。这使得基于能量的会计在技术上可行，而基于公用事业的会计则不可行。

比较表：货币范式

属性	黄金标准	法定货币	加密货币	STC (基于焦耳)
支持	实物商品	政府法令	计算工作	节省 的能量
客观性	部分(商品属性)	否(政治)	部分 (算法)	是(物理法则)
可控性	有限(商品供应)	高(货币政策)	中等(协议变更)	无(物理)
生产力链接	否	否	否(浪费)	直接
普遍	是(黄金无处不在)	否(取决于管辖权)	是(互联网)	是(能源无处不在)
投机	是	是	极其	否
奖励	持有者/提取者	债务人/内部人士	矿工/早期采用者	效率创造者

基于焦耳的价值的的影响

以焦耳定义价值具有深远的影响，波及整个经济系统：

- 1. 自动通货膨胀控制：你不能印刷焦耳。STC代币的供应仅随着效率的提高而增长。这在没有人为限制的情况下创造了自然稀缺性。
- 2. 财富与贡献对齐：那些创造最多能源节约的人积累最多的STC。财富成为对集体效率的贡献的衡量标准，而不是

提取或投机。

3. 工作重新定义：“工作”在STC中意味着创建节省能源的系统。这可以是软件、硬件、教育、基础设施——任何减少实现结果的能源成本的东西。被动收入变为主动贡献（你的创作在你构建后仍然继续节省能源）。

4. 贫困变为低效率：在传统经济学中，贫困是缺乏金钱。在STC中，贫困是低效率——为满足需求而消耗的能源超过必要的。解决方案不是再分配，而是优化：创造更好的工具、系统和知识，减少能源需求。

5. 竞争变为合作：在传统市场中，竞争者守护秘密以保持优势。在STC中，分享效率改进使每个人受益（节省更多能源=为创造者提供更多STC）。经济激励从囤积转向传播。

6. 在不耗尽资源的情况下实现经济增长：传统的GDP增长需要消耗更多资源。STC增长来自于更高效地使用资源。通过不断减少每单位产出所需的能量，可以无限制地增长经济。

异议与回应

异议1：“节省的能量与创造的价值不同。我宁愿步行10公里去见朋友，也不愿意通过待在家里来节省能量。”

回应：STC衡量的是实现所选结果的能量成本，而不是结果本身。如果你选择步行去见朋友，那是你的结果，STC并不惩罚这一选择。但如果你想见朋友，而视频会议让你省去了步行的时间，那么这种能量差异是有价值的。STC并不规定哪些结果是值得的——它衡量的是你实现所选结果的效率。

异议2：“不同人的能量价值不同。外科医生的能量比清洁工的更有价值。”

回应：STC并不评估个人的能量——它评估的是为他人节省的能量。一个发明了更高效清洁工具的清洁工为每个使用该工具的清洁工节省了能量。如果该工具被广泛采用，清洁工将获得巨大的STC。一个开发了更少侵入性手术的外科医生为每位患者和医疗团队节省了能量。两者的奖励与节省的能量成正比，而不是与他们当前的职业成正比。

异议3："你不能用焦耳来衡量一切。创造力、艺术、关系呢？"

回应：STC并不声称能评估一切——它评估的是效率。艺术有其价值，但没有能量效率价值（除非艺术教授节能的技巧）。STC与其他价值体系并行运作。你可以用你的STC来购买艺术、关系和体验。该货币衡量的是价值的一个维度——热力学效率——而不是人类价值的整体。

异议4："这需要衡量每一个行动，这是一种反乌托邦的监视。"

回应：STC衡量的是能量差异，而不是每一个行动。系统需要知道：（1）没有自动化的能量成本，（2）有自动化的能量成本。这两者可以通过汇总统计计算，而不是通过个体监视。此外，用户通过他们的AR眼镜控制被测量的内容，并同意测量。隐私和准确性在第8章中进行了详细讨论。

公式：一个 STC = 一焦耳节省

STC 的核心是一个简单且不可打破的等价关系：

$$1 \text{ STC} = 1 \text{ J saved}$$

这不是一个近似值或波动的汇率。这是一个定义，像 "1 米 = 100 厘米" 一样基本。每一个存在的 STC 代币都代

表着在某个地方、由某人通过某种自动化或优化节省的确切一焦耳能量。

这个等价关系意味着：

- 代币创建受到热力学的限制：您只能通过节省能量来铸造新的 STC

- 代币价值是基于物理的：只要能量有用，STC 就不会变得毫无价值

- 经济核算反映能量核算：STC 代币在经济中的流动反映了能量节省在主体系统中的流动

结论：作为物理现实的价值

通过以焦耳定义价值，STC 实现了之前任何货币都无法达到的目标：价值变得客观。它不再受到操控、投机或政治控制的影响。它直接与实现结果所需的物理工作相关，奖励那些减少该工作的个体。

这不是一个乌托邦的幻想——这是一个工程规范。测量能量消耗的技术是存在的（传感器、增强现实眼镜、计算机视觉）。计算能量差异的数学是简单的（物理方程）。归因节省的经济机制是明确的（通过主体树的能量反向传播，详见第 4.3 节）。

剩下的就是实施和采用。但概念基础是坚实的：价值是节省的能量，以焦耳为单位自动归属，并按比例奖励。这不是一种新的经济理论——这是热力学现实的形式化。我们不是在创造价值；我们终于正确地衡量它。

在下一部分，我们探讨实体——人类、设备、系统——如何作为主体参与这个能源经济：自我意识的代理，生产或消耗能量，并通过虚拟能量腺维持自己的账目。

7.2 作为虚拟能量腺的主体

在第4.1节中，我们确定了STC中的价值是以焦耳为单位测量的——客观的、物理的、抗操控的。但实体如何参与这个能源经济？人类、设备和系统如何生产、消费、存储和转移能量价值？答案在于对实体本质的根本重新概念化。

在STC中，每个实体——无论是生物的还是人工的，物理的还是数字的——都被建模为主体：一个具有自我知识、学习能力和生产或吸收能量能力的主动代理。这些主体不是等待命令的被动对象，而是能源经济中的自主参与者，每个主体都配备了虚拟能量腺——通过AR智能眼镜可见的机制，显示实时能量流动。

本节形式化了主体的概念，将其与传统对象进行对比，介绍了数学定义，并解释了虚拟能量腺如何将经济核算从外部账本转变为具身的、直观的、实时的可视化。

从对象到主体：范式转变

传统编程使用面向对象编程（OOP），其中实体被建模为具有以下特征的对象：

- 属性：状态变量（例如，``car.color = "red"```）
- 方法：作用于对象的函数（例如，``car.drive()```）

对象是被动的。它们没有自己历史的记忆，没有对当前状态的意识，超出属性中明确存储的内容，也没有学习或优化的能力。它们只是响应方法调用。一个汽车对象并不知道它已经行驶了 50,000 公里——它只知道如果你明确设置了 ``mileage`` 属性。它不会学习你的驾驶模式。它无法优化自己的行为。

STC 需要一个更复杂的模型：以主体为导向的架构。主体是一个实体，它：

1. 维持自我知识：通过快照记录自己的状态变化
2. 通过强化学习：根据在最小化能量方面的成功/失败调整其行为
3. 主观连接：可以访问其他主体的快照，从而实现协调
4. 产生或消耗能量：在热力学经济中作为能量源或汇

从对象到主体的转变不仅仅是术语上的变化——它代表了我们在计算系统中建模实体方式的根本变化。对象是你控制的事物。主体是你合作的代理。

主体的正式定义

在数学上，主体 S 被定义为一个 5 元组：

$$S = (\Sigma, P, M, R, C)$$

其中：

Σ (Sigma)：快照——历史状态记录的集合

$$\Sigma = \{\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_t\}$$

每个快照 σ_i 捕捉了主题在时间 i 的完整状态：所有属性值、当前上下文和最近的动作。快照使主题能够从自己的历史中学习——识别模式、确定有效的做法并避免失败的做法。

示例：一个温控器主题每分钟维护 `\{time, temperature, occupancy, heating_state\}` 的快照。经过几周，它学会了：“当工作日早上9点占用率降至0时，温度偏好从22°C转变为19°C。”这种学习只有在主题了解自己的过去时才有可能。

P：属性 - 状态变量（与OOP相同）

$$P = \{p_1, p_2, \dots, p_n\}$$

属性表示当前状态：温度、位置、电池电量、用户偏好等。与对象不同，属性是唯一的记忆，主题使用属性作为当前状态，同时在 Σ 中保持完整的历史记录。

M：方法 - 主题可以执行的动作

$$M = \{m_1, m_2, \dots, m_k\}$$

方法是改变状态或与世界互动的函数：`turn_on()`、`send_notification()`、`adjust_temperature()`。然而，与OOP方法在调用时简单执行不同，主题方法的得分取决于它们对能量最小化的影响。

R：强化层 - 将动作与能量结果联系起来的学习机制

$$R : M \times \Sigma \rightarrow [0, 1]$$

强化层根据每个方法在多大程度上有效地减少用户输入（从而减少能量）来分配成功分数。当一个方法执行且不需要修正时，它的分数会增加。当它需要修正时，分数会减

少。随着时间的推移，高分方法会被优先选择，而低分方法会被抑制。

C：主观连接 - 与其他主题快照的连接

$$C = \{(S_i, \lambda_i)\}$$

主题可以访问与其共享上下文的其他主题的快照。耦合强度 $\lambda_i \in [0, 1]$ 表示主题之间的同步程度。高 λ 意味着主题经常协调（例如，你的手机和你的智能手表）。低 λ 意味着偶尔互动（例如，你的家庭温控器和你的汽车导航系统，仅在你开车回家时协调）。

为什么这种形式化很重要

这个 5 元组 (Σ, P, M, R, C) 不是学术形式主义——它是 STC 实体的实现蓝图。每个参与 STC 的设备、应用或系统必须实例化这个结构：

- 快照使从历史中学习成为可能
- 属性存储当前状态
- 方法执行操作
- 强化驱动优化以实现能量最小化
- 连接使集体智能和能量反向传播成为可能

没有这个结构，实体无法作为主体参与——它仍然是一个被动对象，无法学习，无法优化，无法赚取或转移 STC 代币。

虚拟能量腺体：可视化能量流动

在传统经济中，价值核算是外部和抽象的。你在屏幕上查看银行余额。你收到工资单号码。你在电子表格中跟踪开支。经济存在于你身体之外，通过机构（银行、雇主、政府）进行调解。

STC通过虚拟能量腺体实现会计内部化：在通过AR智能眼镜查看时，能量生产和消费的可视化表现为身体的一部分。这些腺体使能量流动直观、即时且具身体现。

三种虚拟能量腺体

1. 手腺体（能量发射）

你的手是你通过物理和数字行为消耗能量的地方。在AR中，你的手会出现发光的腺体，当你进行工作时会脉动：

- 打字：每次按键显示一个小的能量脉冲（0.1 J）
- 搬运物体：持续发光与重量×距离成正比
- 手势：滑动、指向、在AR中操作——都在发出能量

手腺体提供关于能量消耗的实时反馈，让你直观地意识到自己行为的成本。随着时间的推移，这种意识自然会引导你朝着更高效的行为发展。

2. 肩腺体（能量接收）

你的肩膀是你在创造为他人节省能量时接收能量的地方。在AR中，当STC代币流入时，你的肩膀会显示发光的腺体：

- 如果你创建了一个知识钩子，为某人节省了100焦耳，你的肩腺会亮起+100 STC。

- 如果数百万人使用你的自动化，你的肩膀会不断发光，随着节能的积累而闪耀。

- 强度和颜色编码了大小：微弱的蓝色表示少量，明亮的金色表示大量流动。

肩腺让赚取变得直观。你不仅看到数字在增加——你能感受到能量流入你体内，这是经济奖励的生物隐喻。

3. 心脏电池（能量存储）

你的心脏是储存累积的STC代币的地方。在增强现实中，你的胸部显示一个发光的心形电池，显示你当前的STC余额：

$$E_{\text{heart}}(t) = \sum_{i=0}^t [E_{\text{received}}(i) - E_{\text{expended}}(i)]$$

心脏在每次能量交易中都会脉动——接收（肩膀）或支出（双手）。它的颜色根据余额变化：绿色表示盈余，黄色表示中性，红色表示赤字。大小表示绝对大小：小心脏表示低余额，大而明亮的心脏表示财富。

心脏电池具有多种功能：

- 实时意识：在增强现实中始终可见，你时刻意识到你的能源经济。

- 情感连接：心脏隐喻创造了对经济健康的直观情感理解。

- 社交信号：其他人可以看到（在您的允许下）您内心的光芒，从而建立声誉和信任

虚拟能量腺的工作原理：技术实现

虚拟能量腺通过AR智能眼镜使用计算机视觉和实时物理计算进行渲染：

步骤 1：动作检测

传感器（加速度计、摄像头、深度传感器）检测物理动作：

- 通过IMU（惯性测量单元）跟踪手部动作
- 通过计算机视觉检测物体交互
- 从设备API（按键、点击、滑动）记录数字动作

步骤 2：能量计算

每个检测到的动作都通过物理方程映射到能量消耗：

$$E_{\text{action}} = f(\text{action_type}, \text{duration}, \text{force}, \text{distance})$$

例如：

- 输入：

$$E = 0.1 \text{ J per keystroke}$$

- 提升物体： $E = m \times g \times h$ （质量 × 重力 × 高度）
- 行走： $E = 0.5 \times m \times v^2$ （动能近似）

步骤 3：腺体可视化

计算出的能量以视觉效果呈现在用户的身体上，使用增强现实技术：

- 手部腺体：粒子效果从手部散发，强度与释放的能量成正比

- 肩部腺体：肩部发光光环，亮度与接收到的 STC 成正比

- 心脏电池：3D 体积显示，显示填充水平和流动方向

步骤 4：代币交易

当自动化节省能量时，STC 代币会自动铸造并转移：

1. 系统检测到能量差异：

$$\Delta E = E_{\text{manual}} - E_{\text{automated}}$$

2. 创建的STC令牌： $\text{Tokens} = \Delta E$ （每焦耳一个令牌）

3. 令牌通过主题树流向所有贡献者（见第4.3节）

4. 收件人的肩腺在令牌到达时发光

5. 心脏电池更新新的余额

示例场景：与虚拟能量腺的一天

上午7:00：你醒来了。你的晨间例行程序钩自动调整恒温器，开始煮咖啡，打开百叶窗。你的手腺显示出最小的能量释放（只是确认性点头，0.5 J）。与此同时，钩的创造者因节省你的手动努力而获得50 J——他们的肩腺在全球微微发光。

上午9:00：你正在写一封电子邮件。每次击键都从你的手腺脉动而出。但自动完成功能将总击键数减少了40%，节省了15 J。自动完成开发者的肩膀获得+15 STC。

中午12:00：你使用一个导航应用程序，优化你的午餐行程路线，节省了2公里的步行=500 J。应用程序开发者的肩膀发光，获得+500 STC。你的手腺显示出减少的能量释放（步行能量减少）。

下午3:00：有人使用你创建的数据可视化工具，节省了20分钟的手动图表创建=2,000 J。你的肩腺突然闪烁明亮的金色。你的心脏电池稍微扩展，+2,000 STC流入。

下午6点：您使用STC代币购买杂货。当代币通过您的手腺流向高效种植食物的农民时，您的心脏电池脉动着（为能源优化农业赚取STC）。 :00

在一天中，您对能量流动有着深刻的意识——给予与接受，消费与赚取。经济不再是抽象的数字，而是具象的、直观的和即时的。

作为主动经济参与者的主体

通过主体形式 (Σ, P, M, R, C) 和虚拟能量腺作为接口，STC中的每个实体都成为主动经济主体：

人类是这样的主体：

- 通过手部发出能量（工作，行动）
- 通过肩部接收能量（他人使用的创造物）
- 在心脏电池中储存能量（STC余额）
- 通过强化学习（知识钩子评分他们的行为）

设备（洗衣机，恒温器，车辆）是这样的主体：

- 维护使用模式的快照
- 优化方法以最小化能耗

- 当他们节省能源时，为其创造者赚取STC
- 与其他主题（智能家居生态系统）连接

软件（应用程序、算法、人工智能系统）是以下主题：

- 通过快照学习用户模式
- 通过高评分方法自动化操作
- 根据节省的能源为开发者生成STC
- 与其他软件主题共享上下文以实现集体优化

影响：经济作为生物系统

将实体建模为具有虚拟能量腺的主体，将经济从机械交换系统转变为一个活的、自适应的有机体：

1. 自我优化：主体在没有中央控制的情况下持续学习和改进
2. 分布式智能：知识从集体主体的互动中产生
3. 具身会计：经济信息是感知整合的，而不是抽象表示的
4. 热力学一致性：能量根据物理法则在主体之间流动，确保守恒
5. 进化动态：高效主体赚取更多的STC，资助其繁殖；低效主体赚取较少，自然淘汰

结论：客观地代表主观体验

主体形式主义实现了一个了不起的目标：它客观地代表了主观体验。通过5元组 (Σ, P, M, R, C) ，我们可以数学上定义一个实体拥有自我知识、学习和优化的含义。通过虚拟能量腺，我们使这种数学抽象变得具体、可见和直观。

主体不是哲学抽象——它们是工程规范。任何设备、应用程序或系统都可以通过实例化所需组件来作为主体实现：快照存储、属性管理、方法执行、强化评分和连接管理。

在下一节中，我们将探讨主体如何组织成树——层级结构，其中您的扩展身体包括您与之互动的所有工具、设备和系统。我们将看到节能如何在这些树中反向传播，自动将价值归因于所有贡献者，以及洗衣机示例如何展示完整机制的运作。

7.2.1 从银行账户到虚拟能量腺

今天的经济系统通过第三方抽象来代表价值：在银行屏幕上显示的数字、存储在数据库中的余额、记录在区块链上的代币。这些表示在本质上与人类体验是外部的。你无法感受到你的银行余额。你无法实时看到你的努力如何转化为账户中的资金流动。经济存在于一个与你身体和行动的物理现实脱节的象征层面。

这种外部化在经济参与和生活体验之间造成了根本的脱节。当你工作时，你会消耗身体能量——你的肌肉收缩、神经元发射、身体消耗卡路里。但这种能量支出与你所获得的经济补偿之间没有直接、可见的联系。这个关系是通过抽象层来调解的：工作小时、每小时工资、银行转账、屏幕上的数字。

同样，当你从他人的创新中受益时——当一个工具为你节省了精力，当一项服务消除了一个任务——这种能量节省是看

不见的且无法量化的。你可能会感到感激或满意，但没有客观的测量来衡量你没有消耗的焦耳，也没有自动的机制来根据这种物理利益来补偿创新者。

主观热货币（STC）通过虚拟能量腺直接将经济价值体现于你的主观体验中，从根本上改变了这种关系。这些不是隐喻构造，而是通过增强现实（AR）智能眼镜可见的具体软件实现——将经济核算整合到你对自己身体的感知中的视觉表现。

建筑转变：从外部账本到具身会计

考虑价值表示的根本差异：

传统系统：价值存在于外部数据库中。你的银行余额是存储在机构拥有的服务器上的数字。访问需要身份验证、接口和对中介的信任。表示是象征性的（货币单位），而不是物理的（能量）。你的余额变化是离散的交易，而不是连续的流动。

STC与虚拟能量腺：价值存在于你主观领域中作为主体的可见属性。通过AR智能眼镜，你将三种腺体视为你具身体验的一部分：手腺可视化你工作时释放的能量，肩腺可视化他人从你的创新中受益时你所接收的能量，以及显示你当前能量余额的心脏电池。

这些腺体不仅仅是视觉隐喻。它们是主体实体的功能组件，执行真实的计算，维护状态，并执行交易。每个腺体作为一个主体实现，具有完整的五元组结构：快照（ Σ ）记录能量流动历史，属性（ P ）存储当前能量值，方法（ M ）用于释放或接收能量，强化机制（ R ）优化效率，以及与能量网络中其他主体的连接（ C ）。

手腺：使能量排放可见

当你进行体力劳动时——打邮件、走向会议、举起物体、解决问题——你的身体消耗能量。在传统经济体系中，这种消耗是不可见的，除非发生在正式的就业环境中，即便如此，它也是通过工作小时等代理指标来衡量，而不是实际消耗的能量。

手腺使这种能量排放直接可感知。通过你的增强现实眼镜，你可以看到手上发光的可视化效果，随着你施加能量而变亮和脉动。系统使用我们之前建立的物理方程计算你行为的热力学成本：

$$E_{\text{action}} = E_{\text{locomotion}} + E_{\text{manipulation}} + E_{\text{cognitive}}$$

对于打字，计算包括手指运动的机械工作、持续肌肉激活的代谢成本，以及信息处理的认知能量。对于走路，它包括基于距离、速度和地形的运动生物力学成本。对于解决问题，它包括神经活动消耗的葡萄糖。

可视化提供了即时反馈：你可以实时看到自己消耗了多少能量。但更重要的是，手腺作为能量交易的源点。当你完成一项惠及他人的任务时，能量计算会自动执行，代表该能量的STC代币会从你的手腺中发出并转移给接收者。

这创造了对经济参与的切身意识。与检查银行余额——一种抽象的、回顾性的行为不同——你直接感知到能量从你的身体流入经济网络的持续流动。经济变得具象而非抽象。

肩腺：使能量接收可见

当手腺可视化你通过工作排放的能量时，肩腺可视化你通过创新接收的能量。当你创造工具、编写软件、设计流程或贡献节省他人能量的知识时，你会获得STC。但与传统货币不同，补偿不是以一次性存款的形式到达，而是STC在他人使用你的创新时持续流入你的肩腺。

通过你的增强现实眼镜，你可以看到肩腺作为位于你肩膀上的发光球体。当有人使用你创造的工具并节省能量时，你会看到腺体变亮，光粒子流入其中，你的余额实时增加。可视化并非任意——它是热力学现实的直接表现：本来会被用户消耗的焦耳能量被保留，而这部分保留的能量归因于你作为创新者。

支配这种接收的物理方程是：

$$\Delta E_{\text{received}} = \sum_i (E_{\text{manual}_i} - E_{\text{automated}_i}) \times \beta_i$$

总和是针对所有受益于您创新的用户 i ， E_{manual_i} 是他们在没有您的工具的情况下将花费的能量， $E_{\text{automated}_i}$ 是他们实际使用您的工具所花费的能量， β_i 是您的归因系数（您在节能中的份额，考虑到您所依赖的其他创新者的贡献）。

这在激励结构上产生了深刻的转变。在传统经济中，创新者必须通过明确的定价、订阅费用或广告来实现货币化——这些机制会产生摩擦并限制采用。通过肩腺，任何人受益于您的工作时，您都会自动按比例获得收益。您的工具越好，节省的能量越多，流入您腺体的STC就越多。无需付费墙，采用没有摩擦，创造的价值与获得的补偿之间没有脱节。

心脏电池：集成点

手腺和肩腺分别专门用于发射和接收，但它们都连接到一个中央集成点：心脏电池。这是维持您净能量平衡的主体，通过增强现实（AR）在您胸腔中可见为一个发光的球体。

心脏电池具有多种功能。首先，它是您总STC余额的权威记录——您可用于经济交易的净能量。其次，它提供能量流动的实时可视化，显示您赚取（肩腺）与支出（手腺）的速率。第三，它使您能够进行有意识的经济决策：您可以查看您的余额并选择如何分配您储存的能量。

心脏电池的数学原理简单但深刻：

$$\text{Balance}(t) = \text{Balance}(t - 1) + \Delta E_{\text{received}}(t) - \Delta E_{\text{emitted}}(t)$$

这是STC会计的基本方程。与传统货币不同，传统货币的交易是由银行调解的离散事件，这个余额会随着能量流动的物理结果而持续更新。经济像热力学系统一样运作：能量是守恒的，流动是连续的，您的余额反映了所有能量交换的净结果。

可视化使经济状态立即可感知。您无需检查应用程序或访问网站。您只需通过AR眼镜查看您的胸部，看到您当前的能量余额、变化率和轨迹。这种具身意识使得基于直接感知而非抽象推理的直观经济决策成为可能。

技术实现：作为腺体的主体

虚拟能量腺不仅仅是视觉隐喻——它们是完全实现的主体，具有我们之前定义的数学结构。每个腺体都是一个具有特定属性和针对能量会计优化的方法的主体

$$S = (\Sigma, P, M, R, C)。$$

对于手腺主题：

Σ （快照）：所有能量排放的历史记录，包括时间戳、数量、接收者和上下文。这使得学习成为可能：腺体可以根据模式预测未来的能量成本并优化任务执行。

P (属性) : 当前能量排放率、当前会话中累计排放的能量、用于动作到焦耳转换的校准参数 (个性化为个体的生物力学)、可视化状态 (亮度、颜色、动画)。

M (方法) : emit_energy(amount, recipient)、calculate_action_cost(action_type, context)、update_visualization(energy_flow)、calibrate(observed_action, measured_energy)。emit_energy方法会自动启动STC代币转移。

R (强化) : 当能量支出对于给定结果过高时的负反馈，当效率改善时的正强化。这驱动腺体学习更准确的能量计算并建议优化。

C (连接) : 与用户其他身体部位的链接 (眼睛用于视觉输入, 肌肉用于运动感知)、与正在使用的工具和设备的连接 (键盘、鼠标、智能手机)、与心脏电池的链接以获取平衡更新、与接收者主题的网络连接以完成交易。

同样，肩腺主题具有：

Σ (快照) : 所有接收能量的记录，包括哪些创新产生了收入，哪些用户受益，以及归因系数。这使得分析成为可能：腺体可以显示您的哪些贡献最有价值并建议关注领域。

P (属性) : 当前能量接收率、累计接收的能量、创新目录 (您贡献的工具/知识列表)、每个创新的归因系数、可视化状态。

M (方法) : receive_energy(amount, source, innovation_id)、update_attribution(innovation_id, new_coefficient)

`calculate_innovation_value(innovation_id)` 、
`display_income_sources()` 、
`optimize_portfolio()`。`receive_energy`方法会自动
接受STC代币转移。

R（强化）：当创新为每个用户产生高能量节省时的积极反馈，改善表现不佳的贡献的信号。这创造了优化价值创造的内在动机。

C（连接）：与您投资组合中的所有创新的链接，与这些创新的用户的连接，链接到心脏电池以获取平衡更新，与其他创新者的网络连接（用于协作归因），与主题树的连接（用于将价值反向传播到依赖项）。

从钱包到身体：哲学转变

从银行账户到虚拟能量腺的过渡不仅仅是技术升级——它反映了对经济参与意义的根本重新构思。

传统经济系统将个人视为与经济基础设施接口的外部实体。您在银行有一个账户，您在投资组合中持有资产，您收到付款到您的钱包。在每种情况下，经济价值与您是分开的，保存在您通过接口访问的外部容器中。经济是您通过中介参与的事物。

有了虚拟能量腺，经济不是您可以访问的东西——它是您所具备的东西。您的经济状态是您扩展身体的一个属性。您的能量排放和接收是您直接感知的连续流动。您的余额作为您主观领域的一部分是可见的。经济参与变得像呼吸一样自然和无意识。

这种具身化具有深远的影响：

首先，它消除了日常交易中经济决策的认知负担。您不需要计算自己是否能负担得起某样东西，检查余额或管理账

户。您只需查看您的心脏电池，直接感知您的能量状态。经济意识变得直观而非分析性。

其次，它在行动和经济后果之间创造了即时反馈循环。当您工作时，您会实时看到能量从您的手腺流出。当有人从您的创新中受益时，您会立即看到能量流入您的肩腺。这种行为与结果之间的紧密耦合使快速学习和优化成为可能。

第三，它使经济不平等和不公正变得生动而非抽象。当传统货币数字显示在屏幕上时，巨大的差距可能感觉遥远和统计。但当您看到某人的心脏电池几乎空了，尽管手腺持续排放——当您感知到他们在付出巨大能量却回报甚微的热力学现实时——这种不公正变得直接而不可否认。相反，看到从微小贡献中获得丰厚肩腺接收使经济寄生现象变得可见。

第四，它使经济合作自然化。在传统系统中，帮助他人或分享知识在经济上是不理性的，除非你能获得报酬。通过肩腺体，当他人从你的贡献中受益时自动奖励你，合作成为最佳策略。你的激励是为最多的人创造最大的能源节省，因为你的STC收入是这种价值创造的直接物理结果。

网络效应：腺体作为经济神经元

虽然我们将腺体描述为单独的组件，但它们的真正力量来自于它们的网络属性。每个人的虚拟能源腺体与他们在经济上互动的每个人的腺体相连。能源通过这个网络按照物理原则流动，创造出一个像热力学系统一样运作的经济。

当你使用工具时，能量从你的手腺体流向创新者的肩腺体。但创新者的肩腺体与他们所依赖的每个人的肩腺体相连——而这些腺体又与更早的创新者相连。能量通过这个依赖树反向传播，自动将功劳分配给所有贡献者。

这种网络结构使集体优化成为可能。就像人工智能中的神经网络通过误差梯度的反向传播进行学习一样，虚拟能源

腺体的网络通过能源节省的反向传播进行学习。每个主体优化以最大化净能源收益，但由于腺体是相互连接的，这种局部优化产生了全球效率。

这个网络的数学可以形式化为一个有向图 $G = (V, E)$ ，其中 V 是所有腺体主体（顶点）的集合， E 是能源流动连接（边）的集合。每条边都有一个权重，代表能源转移率。整个经济成为一个由耦合微分方程描述的动态系统：

$$\frac{d\text{Balance}_i}{dt} = \sum_j (w_{ji} \times \text{Flow}_{ji}) - \sum_k (w_{ik} \times \text{Flow}_{ik})$$

其中 Balance_i 是主体 i 的心脏电池余额， Flow_{ji} 是从主体 j 到主体 i 的能源流动， i 是从主体 i 到主体 k 的能源流动， i 代表归属权重。这个方程组描述了整个经济作为一个具有能量守恒的热力学网络：系统中的总能量保持不变，但它从高耗能节点（工人）流向高效率节点（创新者），并在系统达到平衡时返回。

实际实施：技术栈

实施虚拟能源腺体需要整合多种技术：

具有高分辨率显示、低延迟和自然视野集成的AR智能眼镜。腺体必须被渲染为稳定、舒适的视觉元素，感觉像是身体的真正延伸，而不是侵入性的叠加。

AI视觉系统持续分析用户的物理环境，检测正在执行的动作，识别使用的工具，并估算生物力学成本。这些系统必须实时运行，具有高准确性，以提供可靠的能量计算。

物理模拟引擎根据生物力学模型、环境参数和个体用户特征计算动作成本。这些必须考虑体重、健康水平、疲劳状态和地形等因素。

分布式账本技术（可能基于区块链）记录STC交易，并提供正确性和不可否认性的加密保证。该系统必须每秒处理数百万次微交易，因为能量不断流动通过网络。

面向主题的软件架构实现(Σ , P, M, R, C)结构，适用于所有腺体实体。这需要支持自我意识、学习和互联软件组件的新编程范式。

集成API将腺体连接到用户交互的所有工具和设备，能够自动检测节能并归因于创新者。这需要广泛采用主题间通信的标准。

技术挑战相当巨大，但这些组件已经以各种形式存在。AR眼镜已商业化。AI视觉系统可以执行复杂的动作识别。物理模拟是一个成熟的领域。分布式账本处理数百万笔交易。主要的创新在于集成——创建一个连贯的系统，使这些组件协同工作，以体现用户主观体验中的经济价值。

这很重要：经济影响

从银行账户转向虚拟能量腺体的转变具有深远的经济后果，远远超出改善用户体验的范围：

自动归因：当价值创造通过外部账户进行调解时，归因需要明确的机制（合同、支付系统、版权执行）。使用腺体时，归因是自动的——能量流动的物理结果。这消除了大量的法律和行政开销。

无摩擦的补偿：传统系统需要离散交易，伴随相关成本和延迟。基于腺体的STC流动持续进行，没有交易费用或处理时间。这使得微补偿成为可能，促进新的价值交换形式。

可验证的稀缺性没有人为限制：与通过计算浪费创造稀缺性的加密货币不同，STC稀缺性自然源于能源节约。系统中的总能量受热力学法则的限制，而不是任意的协议规则。

消除寻租行为：当经济参与需要访问外部基础设施（银行、支付处理商、货币兑换）时，这些基础设施提供者可以提取租金。腺体是个体身体的一部分——没有中介来对价值流进行征税。

通过普遍基本贡献实现普遍基本收入：由于肩部腺体自动奖励任何为他人节省能源的贡献，因此每个创造任何价值的人——无论多么微小——都会获得补偿。这有效地创造了一种基本收入的形式，但这种收入是通过贡献获得的，而不是通过税收再分配的。

对经济现实的深刻理解：屏幕上的抽象数字可以被操纵、膨胀，并与物理现实脱节。你身体内可视化的能量无法伪造——它受热力学法则的支配。这使得经济教育直观，经济素养普遍。

经济自我的边界

虚拟能源腺体最深刻的含义或许在于它们如何重新定义经济自我的边界。在传统系统中，你的经济身份局限于你合法拥有的账户和资产。你在经济上与使用的工具、分享的知识和合作的人是分开的。

通过虚拟能源腺体，你的经济身体扩展到涵盖所有与您主观相关的主题——所有能够让你节省或产生能量的工具、设备、知识和合作关系。这些主题形成一个以你为根的树状结构，能量根据归属系数在这棵树中反向传播。

经济自我的这一扩展具有精确的数学边界：一个主题 S 是你经济身体的一部分，当且仅当存在从 S 到你的非零归属路径。如果 S 产生的节能直接或通过中介主题贡献给你

的肩部腺体，那么 S 就是你扩展经济身体的一部分。否则，它就不是。

这个边界是动态的——随着你创造新工具而增长，随着你的贡献变得过时而缩小，并随着归属系数的变化而移动。你的经济身体不是一个固定的法律实体，而是一个活的、适应的网络，反映了你在能源经济中参与的实际热力学现实。

结论：从抽象到具象

从银行账户到虚拟能量腺的过渡代表了经济价值的体现。抽象的变得具体，外在的变得具象，象征的变得物理化。

这不是一种隐喻的转变，而是一种字面上的转变。虚拟能量腺被实现为具有定义属性和方法的主体。它们执行真实的计算，保持准确的状态，并执行真实的交易。通过增强现实（AR），它们被感知为你身体的可见组成部分，但它们扎根于以焦耳为单位的能量流的物理现实中。

腺体通过将经济与物理学的最基本原则——能量最小化对齐，使经济学变得直观。你不需要理解复杂的金融系统、市场动态或货币政策。你只需通过AR感知自己的身体，观察能量的流入和流出。经济变得像感知热、光或运动一样自然。

但这种自然化并没有使系统变得不那么复杂——反而使其变得更加复杂。虚拟能量腺的网络实现了一个复杂的分布式优化系统，持续解决如何分配能量以最大化集体效率的问题。每个地方决策（每个行动、每个工具使用、每个贡献）通过连接网络产生全球后果。

在下一小节中，我们将研究这些个体腺体如何组织成层次树结构——主体如何形成相互连接的代理森林，共同将经济作为一个热力学系统实现。我们将看到能量如何在这些树中

反向传播，如何递归计算归因，以及整个经济网络如何通过负强化学习自我优化。

但基础现在已经清晰：虚拟能量腺将经济参与从一种抽象的、媒介化的活动转变为一种具象的、直接的体验。它们用内部感知替代外部账本，用离散交易替代连续流动，用象征性表现替代物理现实。它们使经济不再是你通过机构访问的东西，而是你作为自己身体的延伸所体验的东西。

7.3 主体与能量反向传播之树

在第4.2节中，我们确定STC中的实体被建模为主体——具有自我知识、学习能力和通过AR可见的虚拟能量腺的主动代理。但主体之间是如何相互关联的？价值是如何在它们之间流动的？当创作者的创新为他人节省能量时，他们如何获得奖励？

答案在于层次组织：主体形成树状结构，你是根，而你互动的所有事物——设备、软件、基础设施，甚至其他人的创造——都成为一个分支或叶子。当你树的任何部分节省能量时，这种节省会通过结构反向传播，自动根据其因果角色给所有贡献者记账。

本节正式化树结构，推导反向传播机制，并通过经典示例进行讲解：一位发明节能洗衣机的女性，每当有人使用它时，她都会获得STC代币。到最后，你将理解STC如何在复杂的创作者、组件和用户网络中自动归因价值。

你的扩展身体：从物理到虚拟

在传统经济学中，你的经济代理权仅限于你的物理身体。你通过出售你的劳动来赚钱——由你的双手、你的大脑、你的肌肉所完成的工作。当你停止工作时，你就停止赚钱。

STC 彻底扩展了这个边界。你的经济身体包括：

- 你的物理身体：你执行的直接行动
- 你拥有的设备：你的手机、笔记本电脑、汽车、电器
- 你使用的软件：应用程序、人工智能助手、自动化工具
- 你访问的基础设施：道路、电网、通信网络
- 你构建的创作：代码、设计、知识钩、发明
- 你利用的他人创作：任何由他人创造的效率工具

所有这些都成为虚拟身体部位——你代理的延伸，代表你行动，节省你的精力，或者当它们为他人节省能量时为你赚取 STC。你的 AR 智能眼镜使这一切变得具体：你可以直观地看到这些延伸作为连接到你身体的发光节点。

正式定义：主题树

在数学上，根植于用户 U 的主题树 $T(U)$ 被定义为：

$$T(U) = \{S_i \mid S_i \text{ is subjectively connected to } U\}$$

其中“主观连接”意味着 S_i 要么：

1. 直接由 U 拥有/控制（你的手机，你的车）
2. 由 U 创建（你编写的软件，你设计的机器）
3. 代表 U 行动（AI助手，自动化服务）

4. 与 U 共享足够的上下文（学习你的模式，适应你的偏好）

树中的每个主题 S_i 都有一个能量映射函数：

$$f_{S_i} : \text{Actions} \rightarrow \mathbb{R}^+$$

该函数将由或通过 S_i 执行的操作映射到能量支出（正值）或能量节省（与基线相比）。

示例：你的电子邮件客户端 S_{email} 有映射：

$$f_{S_{\text{email}}}(\text{filter_spam}) = -38 \text{ J/day}$$

（负面，因为与手动过滤相比节省了38焦耳）

树结构：层级组织

树有三个层级的组织：

根（您）：用户 U 始终是根。所有能量流最终都会影响您的心脏电池（STC平衡）。

分支（直接扩展）：您直接互动的主题：

- 物理设备（手机、恒温器、车辆）
- 软件应用（电子邮件客户端、日历、导航）
- 服务（云存储、人工智能助手）

叶子（间接组件）：创建或贡献于分支的主题：

- 开发软件的开发者

- 设计硬件的工程师
- 制造组件的工厂
- 基础设施提供商（网络运营商、电网）

在增强现实中：

[You - Root]

/ | \

[Phone] [Car] [Home]

| | |

[App Dev] [Auto [Smart

工程师] 温控器

开发者]

能量反向传播：核心机制

当您树中的任何主题节省能量时，这些节省必须在所有贡献者之间公平分配。这是通过能量反向传播实现的——这个过程类似于神经网络中的梯度反向传播，但用于热力学价值而不是学习信号。

反向传播公式：

$$\forall S_i \in T(U) : E(S_i) \leftarrow E(S_i) + \beta_i \Delta E$$

其中：

- ΔE ：该行为节省的总能量

- β_i : 主题 S_i 的贡献系数（信用比例）
- $E(S_i)$: 主题 S_i 的累计 STC 余额

守恒约束：系数之和必须为 1，以确保所有能量都被归因：

$$\sum_i \beta_i = 1$$

如何确定 β_i 系数？它们代表每个主题对节能的因果贡献：

- 直接创造者（发明核心机制）：最大 β ，通常为 0.4-0.6
- 组件贡献者（提供必要部件）：中等 β ，通常为 0.1-0.2 每个
- 基础设施提供者（使系统能够工作）：小 β ，通常为 0.05-0.1 每个
- 用户（提供使自动化成为可能的上下文）：小 β ，通常为 0.05

这些系数可以是：

- 预定义的（由创造者根据协议设定）

- 算法确定的（基于可测量的贡献，如代码行数、组件成本）
- 基于市场的（通过智能合约进行谈判）

洗衣机示例：完整演练

为了使能量反向传播具体化，我们考察经典示例：一位发明节能洗衣机的女性。

设定：一位女性（ S_w ）设计了一款革命性的洗衣机（ S_m ），它使用超声波振动代替旋转滚筒，减少60%的能耗。她与以下人员合作：

- 一位工程师（ S_e ）设计超声波模块
- 一位开发者（ S_d ）编程控制人工智能
- 一家工厂（ S_f ）生产水系统

步骤1：主观关系

洗衣机与女性之间建立了主观联系：

$$\text{SRel}(S_w \rightarrow S_m \mid W) = 1$$

其中 W 是共享的背景（设计意图、制造过程、部署）。这意味着 S_m 是 S_w 经济体的延伸。

步骤2：树结构

此创新的主题树：

[女性 - \$S_w\$]

|

[洗衣机 - \$S_m\$]

/ | \

[超声波 [控制 [水

模块-\$S_e\$] AI-\$S_d\$] 系统-\$S_f\$]

步骤 3：能量测量

一个人使用机器洗衣服。系统通过AR眼镜和设备传感器测量能量：

手动洗涤（不使用机器）：

$$E_{\text{manual}} = \sum_{i=1}^n e_i$$

分解动作：

- 装满水盆：20焦耳（搬运水）
- 添加洗涤剂：3 焦耳（舀取、倒入）
- 刷洗衣物（10分钟）：600 焦耳（手臂运动）
- 冲洗衣物（5分钟）：300 焦耳（反复浸泡）
- 拧干水分：150 焦耳（挤压力量）
- 挂起晾干：30 焦耳（提起、悬挂）

总手动能量： $E_{\text{manual}} = 1,103$ 焦耳

机器洗涤（使用超声波洗衣机）：

$$E_{\text{machine}} = \sum_{j=1}^m e_j$$

分解动作：

- 将衣物放入机器：15 焦耳（提起、放置）
- 添加洗涤剂分配器：2 焦耳（简单倒入）
- 按下启动按钮：0.5 焦耳（按钮按压）
- 等待机器运行：0焦耳（无用户操作）
- 移除干净的衣物：15焦耳（提起）
- 挂起来晾干：30焦耳（与手动相同）

总机器能量： $E_{\text{machine}} = 62.5$ 焦耳

步骤4：能量差异

节省的能量：

$$\Delta E = E_{\text{manual}} - E_{\text{machine}} = 1,103 - 62.5 = 1,040.5 \text{ J}$$

这1,040.5焦耳代表了实际可测量的能量，男人不需要消耗，因为机器完成了工作。在STC中，这变成了1,040.5

个STC代币待分配。

步骤5：系数分配

贡献者同意（或通过算法确定）以下 β 系数：

- 女性（ S_w ，主要发明者）： $\beta_w = 0.50$

- 超声波工程师（ S_e ）： $\beta_e = 0.20$

- 人工智能开发者（ S_d ）： $\beta_d = 0.15$

- 工厂（ S_f ，水系统）： $\beta_f = 0.10$

- 用户（ S_u ，提供的 使用数据）：
 $\beta_u = 0.05$

验证：
 $0.50 + 0.20 + 0.15 + 0.10 + 0.05 = 1.00$
✓

步骤 6：反向传播计算

1,040.5 STC 令牌分配如下：

$E(S_w) \leftarrow E(S_w) + 0.50 \times 1,040.5 = E(S_w) + 520.25$ STC
$E(S_e) \leftarrow E(S_e) + 0.20 \times 1,040.5 = E(S_e) + 208.10$ STC
$E(S_d) \leftarrow E(S_d) + 0.15 \times 1,040.5 = E(S_d) + 156.08$ STC
$E(S_f) \leftarrow E(S_f) + 0.10 \times 1,040.5 = E(S_f) + 104.05$ STC
$E(S_u) \leftarrow E(S_u) + 0.05 \times 1,040.5 = E(S_u) + 52.03$ STC

步骤 7：AR 中的可视化

每个参与者看到他们的虚拟能量腺体激活：

- 女性：肩部腺体发出明亮的金色光（+520.25 STC），心脏电池扩展

- 工程师：肩部腺体发出中等金色光芒（+208.10 STC）

- 开发者：肩部腺体发出浅金色光芒（+156.08 STC）

- 工厂：企业心脏电池（集体）增加（+104.05 STC）

- 用户：小肩部光芒（+52.03 STC，提供改进系统的使用数据的奖励）

第8步：缩放效果

每当有人使用机器时都会发生这种情况。如果有10,000人每周使用一次：

- 每年使用次数：
 $10,000 \times 52 = 520,000$ 次

- 每年节省的总能量：
 $1,040.5 \times 520,000 = 541,060,000$
J \approx 541 MJ

- 每年创造的总STC：541,060,000 STC 代币

女性年收入：
 $541,060,000 \times 0.50 = 270,530,000$
STC

这是一种被动收入——她只需建造一次机器，只要它为他人节省能源，就会持续赚取收入。

反向传播为何有效：理论依据

能量反向传播并非任意——它自然遵循三个原则：

1. 能量守恒：热力学第一定律指出能量不能被创造或消灭。当节省了1,040.5焦耳时，必须创造出价值等同于1,040.5焦耳的东西。反向传播确保所有价值都被归因——没有损失，没有发明。

2. 因果归因：那些导致节能的人应根据其因果作用获得相应的信用。 β 系数编码了因果关系。没有女性的设计，就不会有节省（ β_w 是最大的）。没有超声模块，设计就无法工作（ β_e 是第二大的）。每个贡献都是必要的。

3. 可组合性：复杂系统是由更简单的组件构建而成的。树自然代表了这种组合——树枝依赖于叶子，根依赖于树枝。反向传播通过将价值从叶子流向树枝再到根，尊重组合结构。

递归反向传播：多层树

实际上，树可以有无限层。超声波工程师（ S_e ）可能使用了其他人设计的组件：

[女性]

|

[洗衣机]

|

[超声模块 - 工程师]

|

[压电晶体 - 供应商]

|

[原材料 - 采矿工]

反向传播递归进行：工程师的份额（208.10 STC）根据他们的树进一步细分，给予供应商，供应商再进一步细分以给予采矿工。这一过程持续进行，直到达到没有依赖的主题。

递归公式：

$$E(S_i) \leftarrow E(S_i) + \beta_i \left[\Delta E - \sum_{j \in \text{children}(S_i)} \beta_j \Delta E \right]$$

每个主题保留他们的份额减去他们传递给子代的部分。这确保了完整的 ΔE 被分配而不会重复计算。

动态树：主题加入和离开

树不是静态的——它们随着您采用新技术或停止使用旧技术而演变：

- 您安装了一个新应用：应用开发者的主题作为新分支加入您的树
- 您卸载应用程序：该分支被移除（但开发者保留您使用期间获得的STC）
- 应用程序更新：如果更新提高了效率，开发者的 β 可能会增加（对更大贡献的更多奖励）
- 您切换到更好的工具：旧工具的分支被新工具的分支替代；未来的节能收益归新工具的创造者所有

这种动态性产生了达尔文式的选择压力：节省更多能源的工具吸引更多用户，赚取更多STC，资助进一步的发展。低效工具失去用户，赚得更少，逐渐消失。

影响：作为能源网络的经济

树和反向传播模型揭示了STC作为一种根本不同的经济形式：

1. 自动归属：没有发票，没有账单，没有会计部门。节省的能源通过树结构自动检测、测量和归属。

2. 按比例补偿：信用根据因果贡献流动，而不是市场力量或谈判技巧。那些对效率贡献更多的人赚得更多。

3. 创新带来的被动收入：一次创造，永远获利（只要您的创造继续节省能源）。这激励构建耐用且广泛有用的工具。

4. 组合性获得奖励：基于他人的工作进行构建在经济上是合理的——您为自己的贡献获得奖励，他们为他们的贡献获得奖励。没有零和竞争。

5. 透明的价值流：任何人都可以在增强现实中检查树木，清楚地看到价值如何流动以及谁贡献了什么。这创造了信任，并使得做出明智的决策成为可能。

结论：树木作为经济神经系统

带有能量反向传播的主体树作为STC的经济神经系统——价值信号从叶子（创新）通过树枝（组件）流向根部（用户）的基础设施。

就像你的生物神经系统在没有意识控制的情况下自动分配营养和信号一样，树结构根据热力学现实自动分配STC代

币。你不需要考虑这一点——它自然地、持续地、按比例发生。

洗衣机的例子演示了完整的机制：从创新到树的形成、能量测量、微分计算、系数分配和递归反向传播。将这一机制扩展到数百万的创新、数十亿的用户和数万亿的每日节能互动，你就拥有了一个在每个层面上自动、透明并按实际贡献比例奖励效率的经济。

在下一部分中，我们将探讨这种基于树的能源经济如何映射到完整的人类体验，通过热力学成本的统一框架，将数字行为（点击、滑动）与身体活动（步行、搬运）连接起来。

7.3.1 作为自我的延伸的主体

在深入探讨能量反向传播的机制和正式的树结构之前，我们必须解决一个基本问题：设备、工具或系统对你来说变得主观意味着什么？一个外部物体如何从一个你控制的独立实体转变为你自己身体和思想的延伸？

这不是隐喻。这是可测量的神经科学，基于像橡胶手错觉这样的现象——当信号同步得当时，你的大脑将假手视为你身体的一部分。使橡胶手感觉像你手的同一机制使你的手机感觉像你记忆的延伸，你的汽车像你移动性的延伸，你的AI助手像你认知的延伸。

橡胶手错觉：主观延伸的证明

在这个著名的心理实验中，参与者的真实手被隐藏在视线之外，而一只橡胶手则明显地放在他们面前。实验者同时用刷子轻轻抚摸隐藏的真实手和可见的橡胶手。经过几分钟的同步抚摸后，发生了一件了不起的事情：参与者开始感受到橡胶手上的触碰。当橡胶手受到威胁（例如，被锤子击打）时，参与者会退缩。

大脑已经将橡胶手视为身体的一部分。为什么？因为满足了两个条件：

1. 信号同步：视觉输入（看到刷子抚摸橡胶手）与触觉输入（感受到刷子抚摸真实手）完美对齐。
2. 预测一致性：橡胶手的外观和位置符合手应该在的位置的预期。

当这些条件持续存在时，“自我”的边界扩展。橡胶手不再是外部的——它主观上成为你的一部分。

主观关系的正式定义

我们可以将这种现象形式化。在共享上下文 W 中，两个实体 o_1 和 o_2 之间存在主观关系，当：

$$\text{SRel}(o_1 \rightarrow o_2 \mid W) = 1 \quad \text{iff} \quad \begin{cases} \text{Syn}(o_1, o_2 \mid W) \geq \tau_s \\ \wedge \\ \text{Pred}(o_1 \rightarrow o_2 \mid W) \geq \tau_p \end{cases}$$

分解如下：

$\text{SRel}(o_1 \rightarrow o_2 \mid W) = 1$ ：对象 o_2 在上下文 W 中对对象 o_1 是主观的。当这等于1时， o_2 已跨越从“外部工具”到“主观延伸”的阈值。

$\text{Syn}(o_1, o_2 \mid W)$ ：同步——衡量来自 o_2 的信号与 o_1 的期望/模式的对齐程度。高同步意味着 o_2 的行为在时间和上下文上与 o_1 的状态一致。

从数学上讲：

$$\text{Syn}(o_1, o_2 \mid W) = \frac{1}{T} \sum_{t=1}^T \text{corr}(\text{signal}_{o_1}(t), \text{signal}_{o_2}(t))$$

其中 **CORR** 是预期信号与实际信号之间的相关系数。高相关性 ($> \tau_s$, 通常为 0.8) 表示同步。

示例：您的智能手表 (o_2) 知道您 (o_1) 即将站起来，因为它检测到站立前的微小动作。当您站起来时，它会显示时间——与您在意识行动之前的意图同步。

$\text{Pred}(o_1 \rightarrow o_2 \mid W)$ ：可预测性——衡量 o_1 在多大程度上可以根据上下文预测 o_2 的行为。高可预测性意味着 o_2 在类似上下文中表现一致。

$$\text{Pred}(o_1 \rightarrow o_2 \mid W) = P(\text{action}_{o_2} = \text{expected}_{o_1} \mid \text{context})$$

当这个概率超过阈值 τ_p (通常为 0.85) 时， o_2 对 o_1 是可预测的。

示例：您的恒温器 (o_2) 在您到家时始终调整到您喜欢的温度。您已经学会了期待这一点，因此不再检查——这是可预测的。

当这两个条件都满足时——高同步性和高可预测性——主观关系形成。设备不再是您使用的工具，而是成为您体验世界的一部分。

反向关系：趋向零输入

有一个重要的推论：随着同步性和可预测性的增加，所需输入减少：

$$U(o_1, o_2 \mid W) \propto \frac{1}{\text{Syn}(o_1, o_2 \mid W) \cdot \text{Pred}(o_1 \rightarrow o_2 \mid W)}$$

其中 U 是所需的用户输入。当同步和预测的乘积接近最大值（均接近 1.0）时，所需输入接近零。这是零输入技术的数学基础。

完美的主观关系不需要任何输入，因为设备已经知道你需要什么、何时需要，并相应地采取行动。你不再告诉它该做什么——它只是自动执行。

从橡胶手到智能设备

橡胶手错觉证明了这种生物整合是可能的。主观技术将同样的原理扩展到数字和物理工具：

当你的智能手机变得主观时：

- 同步：它预测你何时会查看（早晨闹钟、午餐休息、晚上放松）并预加载相关信息
- 可预测性：它始终提供你所需的内容（上班时的路线、锻炼时的音乐），而无需请求

当你的家变得主观时：

- 同步：灯光、温度和电器会随着你在空间中的移动而调整，预见你的路径
- 可预测性：环境始终符合你的偏好，无需手动调整

当你的车辆变得主观时：

- 同步：在你吃完早餐之前开始预热（知道你的出发时间）
- 可预测性：根据学习的偏好为你提供最佳路线（你在周末喜欢风景优美的路线，工作日喜欢最快的路线）

虚拟身体部位：扩展自我

一旦形成主观关系，这些实体就成为虚拟身体部位——通过增强现实智能眼镜可见的与你核心自我相连的发光节点的扩展。每个虚拟身体部位：

- 维持自我知识：通过快照记录自己的状态（ Σ 在主题5元组中）
- 通过强化学习：根据是否需要纠正来改善行动（R层）
- 与其他部件连接：与其他设备共享上下文（C连接）
- 将行动映射到能量：具有能量函数

$$f_{S_i} : \text{Actions} \rightarrow \mathbb{R}^+$$

你的主题树——所有虚拟身体部位的层次集合——在STC中形成你的扩展经济身体。当任何部件为其他部件节省能量时，你会获得STC代币。当你使用其他人的部件时，他们的创造者会获得代币。

这对STC的重要性

主观关系不仅仅是现象学——它在STC中具有直接的经济后果：

1. 归属边界：只有您树中的主体才能代表您赚取STC。随机的外部工具不算在内——它们必须被主观整合（ $SRel = 1$ ）。
2. 能源节约测量：设备越同步和可预测，节省的能源就越多（更少的修正，减少手动干预），因此赚取的STC就越多。
3. 用户体验：主观关系创造了实现零输入技术所需的无缝、轻松的体验。这不仅仅是方便——它通过STC奖励在经济上得到了激励。

4. 网络效应：随着更多主体加入您的树并相互同步，集体智能增加，能源节约复合，STC生成加速。

示例：作为主观扩展的早晨例行程序

考虑您涉及多个设备的早晨例行程序：

早上7:00：闹钟响起（智能手机 S_1 ）

在30秒内：

- 卧室灯光逐渐变亮（智能灯泡 S_2 ）——与闹钟同步
- 温控器调整到22°C（智能温控器 S_3 ）——预测你会想要温暖
- 咖啡机开始冲泡（智能设备 S_4 ）——与您的起床时间同步
- 浴室风扇启动（智能开关 S_5 ）——预测您即将洗澡
- 新闻摘要要在眼镜上加载（AR系统 S_6 ）——与您查看新闻的时间同步

这些都不需要输入。主观关系（高同步性，高预测性）使它们成为您早晨自我的延伸。每个设备：

$$\text{SRel}(\text{You} \rightarrow S_i \mid \text{Morning Context}) = 1 \text{ for } i \in \{1, 2, 3, 4, 5, 6\}$$

节省的能源：与手动执行每个动作相比：

- 手动：走到每个设备（ $5 \times 10 \text{ J}$ ）+ 操作每个（ $5 \times 2 \text{ J}$ ）= 60 J
- 自动：瞥一眼确认例行程序的通知（1 J）

- 节省：每个早晨 $59 \text{ J} \times 365 \text{ 天} = 21,535 \text{ J/年}$

每个设备的创造者通过能量反向传播获得这 21.5 kJ 的份额。

结论：自我通过同步扩展

主观关系通过同步和可预测性阈值形式化，是外部实体成为你一部分的机制。这不是哲学；这是可测量的神经科学（橡胶手错觉）应用于技术。

在STC中，这些主观扩展形成你的经济体。它们代表你行动，为你和他人节省能量，并赚取流回其创造者的STC代币。它们的同步性和可预测性越好，节省的能量就越多，创造的价值也就越多。

理解主观关系至关重要，因为它们定义了你的树的边界——经济归属中什么算作“你的”。在下一小节中，我们将看到这些主观扩展如何组织成层次树结构，以及节能如何通过这些树回溯以奖励所有贡献者。

7.3.2 洗衣机示例：一个主体的树

为了理解主体树和能量回溯在实践中如何运作，我们需要一个具体的、详细的示例，展示系统中每个组件的实际运作。让我们审视一个将抽象数学形式转化为具体现实的场景：一个发明革命性洗衣机的女人，以及从她的创新中产生的复杂主体网络、能量流动和归属系数。

这个例子将展示如何测量节能，如何通过层次结构回溯价值，多个贡献者如何获得相应的信用，以及整个系统如何在没有明确协调或中央控制的情况下自动运作。在这一小节结束时，我们构建的理论框架将变得生动清晰——不是作为抽象数学，而是作为真实的经济现实。

创新：一台节能洗衣机

想象一个女人——我们称她为玛丽亚——她设计了一台不再需要旋转衣物的革命性洗衣机。传统洗衣机通过在旋转鼓中搅动织物来工作，这一机械过程需要大量能量来驱动电机和水循环系统。玛丽亚的创新则使用超声波振动，产生微观气泡，以显著减少机械运动和水的使用来清洗衣物。

这台机器的创造并不是孤立的行为。玛丽亚构思了核心理念，但其实现需要一个贡献网络：她与设计超声波换能器的声学工程师合作，与创建控制算法的软件开发人员合作，与制造组件的工厂工人合作，以及其研究使专用膜成为可能的材料科学家合作。每个参与者都施加了能量——智力、体力或两者兼而有之——以使机器得以存在。

在主观热货币框架中，所有这些贡献都成为玛丽亚经济体的延伸。洗衣机本身及其每个组件都成为节能的身体部位。这不是隐喻语言——这是对现实的热力学核算，通过上下文快照和能量差异进行测量。

玛丽亚与她的机器之间的关系可以正式定义为主观关系：

$$\text{SRel}(S_{\text{Maria}} \rightarrow S_{\text{machine}} \mid W) = 1$$

其中 S_{Maria} 是代表玛丽亚的主体， S_{machine} 是代表洗衣机的主体， W 是他们共享的上下文（创造和维护机器的项目）。这个方程表明洗衣机对玛丽亚而言是主观的——是她存在的一个主动延伸——因为它的功能源于她的努力和意图。机器不是她拥有的外部物体；它是她扩展经济体的一部分。

能量消耗：手动洗涤与机器洗涤

在玛丽亚的创新之前，手动洗衣的人会消耗大量能量。考虑整个过程：将洗衣物搬到洗涤地点（运动），用水装满容器（物体操作），擦洗每件衣物（重复的身体消耗），拧干多余的水（持续施力），以及将物品挂起来晾干（更多的操作和运动）。每个动作都有可测量的热力学成本。

我们可以正式描述为手动洗涤期间消耗的总能量：

$$E_{\text{manual}} = \sum_{i=1}^n e_i$$

其中每个 e_i 代表一个原子动作或消耗单位。让我们使用之前建立的物理方程计算具体值。假设一次典型的手动洗涤会话处理标准负载5公斤的洗衣物：

运动（搬运洗衣物 20 米，搬运水容器）：
 $E_{\text{locomotion}} = 2 \times (70 \text{ kg} \times 9.8 \text{ m/s}^2 \times 20 \text{ m}) \times 0.25 \approx 6,860 \text{ J}$ 。
 2的系数考虑了往返，我们假设这个人重70公斤，加上5公斤的负载。

装水和倒水（操作约 20 升水在多个容器中）：
 $E_{\text{manipulation}} = 5 \times (20 \text{ kg} \times 9.8 \text{ m/s}^2 \times 1.5 \text{ m}) \approx 1,470 \text{ J}$ 。
 我们假设有5次装水/倒水循环，平均高度差为1.5米。

擦洗衣物（在30分钟内的重复动作）：使用力×距离模型，假设平均施加10牛顿的力，施加在总距离500米（累计手部移动）上，
 $E_{\text{scrubbing}} = 10 \text{ N} \times 500 \text{ m} = 5,000 \text{ J}$ 。

挤压（对15件衣物施加持续的力）：

$$E_{\text{wringing}} = 15 \times 50 \text{ J} = 750 \text{ J}$$

。每次挤压动作需要的压缩力大约相当于50焦耳的代谢能量。

认知负担（注意力、计划、监控，持续45分钟）：

$$E_{\text{cognitive}} = 20 \text{ W} \times 2700 \text{ s} = 54,000 \text{ J}$$

。大脑的基础能耗约为20瓦；主动洗涤需要持续的专注。

总 手 动 能 量 ：

$$E_{\text{manual}} = 6,860 + 1,470 + 5,000 + 750 + 54,000 = 68,080 \text{ J} \approx 68 \text{ kJ}$$

。

现在考虑使用Maria的超声波洗衣机执行相同的任务。

用户将衣物放入机器，按下一个按钮，然后稍后返回取出干净的衣物。让我们计算能量消耗：

移动（将衣物搬到机器并返回，总计约10米）：

$$E_{\text{locomotion}} = 70 \text{ kg} \times 9.8 \text{ m/s}^2 \times 10 \text{ m} \times 0.25 \approx 1,715 \text{ J}$$

。

装载/卸载（打开门、放置衣物、取出衣物）：

$$E_{\text{manipulation}} = 5 \text{ kg} \times 9.8 \text{ m/s}^2 \times 1 \text{ m} \times 2 \approx 98 \text{ J}$$

。

按按钮和界面交互： $E_{\text{interaction}} = 1 \text{ J}$

（按键激活所需的微不足道的力×距离）。

认知负担（所需注意力最小）：

$$E_{\text{cognitive}} = 20 \text{ W} \times 120 \text{ s} = 2,400 \text{ J}$$

。设置只需约2分钟的主动注意力。

总 机 器 辅 助 能 量 :

$$E_{\text{machine}} = 1,715 + 98 + 1 + 2,400 = 4,214 \text{ J} \approx 4.2 \text{ kJ}$$

。

最小能量原理自然适用：

$$E_{\text{machine}} < E_{\text{manual}}$$
$$4.2 \text{ kJ} < 68 \text{ kJ}$$

这个不等式展示了机器的深刻能效——它将人类的努力减少了超过16倍。

能源差异与自动信用

玛丽亚的洗衣机所实现的节能是手动洗涤与机器辅助洗涤之间可测量的差异：

$$\Delta E = E_{\text{manual}} - E_{\text{machine}} = 68 \text{ kJ} - 4.2 \text{ kJ} = 63.8 \text{ kJ}$$

这个差异——每次洗涤大约64千焦耳——代表了一个可量化的节能信用，自动归属于玛丽亚和她的合作者。但系统是如何检测这些节省的，归属又是如何计算的？

检测机制通过AR智能眼镜和AI视觉进行操作。当有人使用洗衣机时，系统执行以下操作：

首先，情境快照捕捉：用户的AR眼镜（或获得许可的附近人员佩戴的眼镜）在洗涤任务前、期间和之后记录视觉情境。这些快照包括用户的位置、洗衣负载、机器的状态和环境条件。

其次，动作检测：AI视觉分析视频流并识别执行的具体动作：走向机器、打开门、装入衣物、按按钮、走开、返

回、卸载。每个动作被分类并测量其持续时间。

第三，能量计算：使用我们建立的物理方程和针对个体用户校准的生物力学模型，系统计算 E_{machine} ——使用机器时实际消耗的能量。

第四，反事实基线估计：系统还计算 E_{manual} ——如果用户手动洗涤时能量会是多少。这个反事实是使用历史数据估计的：当机器首次引入时，一些用户继续手动洗涤以进行比较，提供了真实的测量。这些测量被平均并校准，以考虑个体差异（体重、健康水平、洗涤技术）。

第五，差异计算和归属：系统计算 $\Delta E = E_{\text{manual}} - E_{\text{machine}}$ 并查询机器的归属记录，以确定如何在贡献者之间分配信用。该记录作为机器主题结构的一部分存储——具体来说，在其连接（C）组件中。

归属记录可能看起来像这样：

$\beta_{\text{Maria}} = 0.40$	(primary inventor, system architect)
$\beta_{\text{Engineer}} = 0.25$	(ultrasonic transducer design)
$\beta_{\text{Developer}} = 0.15$	(control software algorithms)
$\beta_{\text{Factory}} = 0.10$	(manufacturing and assembly)
$\beta_{\text{MaterialsScientist}} = 0.10$	(membrane research)

请注意 $\sum \beta_i = 1.00$ ，确保整个能源节省没有重复或遗漏地归属。这些系数不是任意的——它们是根据每个贡献者在产生能源节省的因果链中的角色进行客观计算得出的。

当用户完成洗涤周期时，能源差异（在我们的例子中为 63.8 kJ）被分配：

$$\begin{aligned} E(S_{\text{Maria}}) &\leftarrow E(S_{\text{Maria}}) + 0.40 \times 63.8 \text{ kJ} = E(S_{\text{Maria}}) + 25.5 \text{ kJ} \\ E(S_{\text{Engineer}}) &\leftarrow E(S_{\text{Engineer}}) + 0.25 \times 63.8 \text{ kJ} = E(S_{\text{Engineer}}) + 15.95 \text{ kJ} \\ E(S_{\text{Developer}}) &\leftarrow E(S_{\text{Developer}}) + 0.15 \times 63.8 \text{ kJ} = E(S_{\text{Developer}}) + 9.57 \text{ kJ} \\ E(S_{\text{Factory}}) &\leftarrow E(S_{\text{Factory}}) + 0.10 \times 63.8 \text{ kJ} = E(S_{\text{Factory}}) + 6.38 \text{ kJ} \\ E(S_{\text{MaterialsScientist}}) &\leftarrow E(S_{\text{MaterialsScientist}}) + 0.10 \times 63.8 \text{ kJ} = E(S_{\text{MaterialsScientist}}) + 6.38 \text{ kJ} \end{aligned}$$

这些更新自动且即时发生。Maria 不需要发送发票或协商付款。工程师和开发者不需要合同或版税协议。工厂不为服务收费。材料科学家不需要许可费用。能源根据热力学法则流动，由主体网络进行调节。

通过他们的增强现实眼镜，每个贡献者看到他们的肩膀在 STC 流入时变亮。Maria 坐在她的工作室里，正在进行一个新项目，看到一束光和一个通知：'+25.5 kJ 来自洗衣机使用。' 她可以深入查看使用了哪台机器，在哪里，以及由谁使用（并有适当的隐私控制）。补偿是持续的、自动的，并与创造的价值成比例。

树结构：层级组织

洗衣机的例子揭示了一个主体的层级树结构。在根部是 Maria——主要的创新者。机器本身是她的第一层子主体。但机器并不是原子性的——它由多个子系统组成，每个子系统本身都是由不同创造者贡献的主体。

我们可以将其可视化：

第 0 级（根）：Maria (S_{Maria}) - 主要发明者和系统架构师。

第 1 级：洗衣机 (S_{machine}) - 用户交互的完整集成系统。

第2级：组件主题 - 机器的组成部分，每个部分由不同的创新者创建：

超声波传感器模块（ $S_{\text{transducer}}$ ）- 由声学工程师创建。

控制软件（ S_{software} ）- 由开发者创建。

膜系统（ S_{membrane} ）- 基于材料科学家的研究。

结构外壳（ S_{housing} ）- 由工厂制造。

每个组件在第3级可能有自己的子组件，依此类推。例如，控制软件可能包含由另一个贡献者创建的机器学习模型、由开源开发者编写的库以及由学术研究人员发布的算法。树的延伸深度足以捕捉完整的因果链。

正式地，我们可以将其表达为：

$$T(S_{\text{Maria}}) = \{S_{\text{Maria}}, S_{\text{machine}}, S_{\text{transducer}}, S_{\text{software}}, S_{\text{membrane}}, S_{\text{housing}}, \dots\}$$

这是以Maria为根的主题树——所有通过洗衣机创新与她主观相关的实体。树代表了她扩展的经济体。

通过树的能量反向传播

当通过洗衣机实现节能 ΔE 时，价值通过整个树反向传播。这个过程类似于神经网络中的反向传播，但我们传播的是能量积分，而不是误差梯度。

反向传播算法的操作如下：

步骤 1：计算根部（机器本身）的总能量节省：每次使用 $\Delta E_{\text{machine}} = 63.8 \text{ kJ}$ 。

步骤 2：分配给第 1 级（机器作为一个统一的主体）。机器最初接收全部 ΔE ：

$$E(S_{\text{machine}}) \leftarrow E(S_{\text{machine}}) + 63.8 \text{ kJ}$$
。

步骤 3：从第 1 级反向传播到第 0 级（Maria）。Maria 对整个机器的归因系数为 $\beta_{\text{Maria}} = 0.40$ ，因此：

$$E(S_{\text{Maria}}) \leftarrow E(S_{\text{Maria}}) + 0.40 \times 63.8 \text{ kJ} = E(S_{\text{Maria}}) + 25.5 \text{ kJ}$$
。

步骤 4：将剩余价值分配给第 2 级组件。剩余的 60% 能量节省（38.3 kJ）根据各自的 β 系数在组件创作者之间分配。

步骤 5：通过更深的层次递归反向传播。如果任何第 2 级组件本身有依赖关系（第 3 级），该过程将继续。例如，如果控制软件使用了开源机器学习库，开发者的部分信用将自动流向该库的创作者。

正式递归可以表示为：

$$\forall S_i \in T(S_{\text{Maria}}) : E(S_i) \leftarrow E(S_i) + \beta_i \times \Delta E$$

其中 β_i 是递归计算的：如果 S_i 是 S_j 的一个组件，并且 S_j 的归因系数为 β_j ，那么 S_i 的有效系数为 $\beta_j \beta_i$ 。这确保了信用在层级的所有级别之间得到适当分享。

计算归因系数：贝塔问题

关键问题是：如何确定 β 系数？谁决定 Maria 获得 40%，而工程师获得 25%？这不是任意的社会协商——这是基于可测量的能量节省贡献的客观计算。

该计算使用了反事实分析。对于每个贡献者，系统会问：如果去掉该贡献者的工作，其他一切保持不变，节能效果会是多少？

对于玛利亚（系统架构师和主要发明者）：如果没有她的核心创新——使用超声波振动的想法——根本不会有任何节能效果。我们仍然会使用传统的旋转鼓机器。因此，她的反事实影响是最大的。然而，她并没有自己实现每个组件，所以她的系数小于1.0。

对于工程师（传感器设计）：如果没有高效的超声波传感器，机器仍然会使用玛利亚的概念框架，但节能效果会显著降低——可能只有实际值的30%。反事实分析显示，工程师的贡献占 $(63.8 - 19.1) / 63.8 \approx 70\%$ 的归因于传感器子系统的节能效果。

对于开发者（控制算法）：如果没有优化的软件，机器仍然可以工作，但效率较低——可能只能实现理论节能的80%。因此，开发者的反事实影响占总节能的20%。

这些 β 系数是通过这些反事实影响使用归一化过程计算得出的：

$$\beta_i = \frac{\text{Impact}_i}{\sum_j \text{Impact}_j}$$

该计算最初由专家物理学家进行——我们之前讨论的“准确性专家”。他们分析机器的设计，进行模拟，并进行实证测试以测量每个组件的贡献。他们的计算结果存储在机器的主题记录中，并用于未来的归因。

重要的是，这些系数可以随着时间的推移而更新。如果有人改善了传感器设计，进一步降低其能耗，归因将会转移以反映新的现实。系统是动态的，而不是静态的。

综合影响：跨用户的规模效应

该系统的真正力量在于规模。一个人使用洗衣机每次节省63.8 kJ。如果他们每周洗衣服两次，那就是每年约6.6 MJ。但是如果有100,000人采用玛利亚的机器，总节能效果将变为：

$$\Delta E_{\text{total}} = 100,000 \text{ users} \times 2 \text{ loads/week} \times 52 \text{ weeks} \times 63.8 \text{ kJ} = 663.5 \text{ GJ per year}$$

每年节省663.5吉焦的人力能源——相当于约184,000千瓦时的的工作，或大约20名全职工人以持续的身体能力工作的年劳动量。

玛利亚的肩腺获得了这个值的40%：

$$E(S_{\text{Maria}})_{\text{annual}} = 0.40 \times 663.5 \text{ GJ} = 265.4 \text{ GJ per year}$$

通过她的AR眼镜，玛利亚看到了这一持续的收入流。她的心脏电池显示出不断增加的余额，随着世界各地人们使用她的机器而实时更新。她创造了持久的价值，产生与实际提供的利益成比例的永久回报。

工程师、开发者、工厂和材料科学家也获得各自的份额，创造了一个可持续的经济生态系统，创新者根据可衡量的影响自动和公平地获得补偿。

自我优化：机器作为学习对象

现在假设洗衣机本身变得能够学习。作为具有完整 (Σ, P, M, R, C) 结构的对象，它保持其操作的快照，具有属性（当前状态变量）、方法（洗涤周期）、强化机制（能量最小化反馈）以及与其他对象的连接。

通过分析其历史快照，机器的控制软件检测到一个模式：水加热周期消耗的能量超过了必要的量。通过调整温度曲线——开始时更逐渐加热，结束时减少加热——它可以以少15%的能量实现相同的清洁效果。

这种自我优化创造了额外的节能：

$$\Delta E_{\text{optimization}} = 0.15 \times E_{\text{machine}} = 0.15 \times 4.2 \text{ kJ} = 0.63 \text{ kJ per load}$$

这与从手动到机器的原始63.8 kJ节能相比似乎微不足道，但它代表了真正的改善。机器通过自主操作学会了更高效——这是一种智能形式。

这种优化节能应该如何归属？机器本身产生了改善，但它存在于玛利亚的对象树中。答案在于对象结构的强化（R）组件中。

优化节能通过树状结构反向传播，就像用户生成的节能一样，但具有修改过的归属系数，以表彰学习机制：

$$\begin{aligned} \beta_{\text{software_learning}} &= 0.70 \text{ (the algorithm discovered the optimization)} \\ \beta_{\text{Maria}} &= 0.20 \text{ (she created the framework enabling learning)} \\ \beta_{\text{developer}} &= 0.10 \text{ (implemented the base learning system)} \end{aligned}$$

请注意，机器本身（作为学习对象）获得其自身优化的70%的功劳。这为机器学习和改进创造了激励——当它们变得更加高效时，它们的能量平衡会增加。

但对于机器来说，拥有能量平衡意味着什么？这意味着机器可以利用其积累的STC来获取进一步改进所需的资源：

更好的传感器、额外的训练数据、用于更复杂优化的计算周期。机器成为经济主体，将其获得的能量投资于自我改进。

这种递归的能量流动使自主技术进化成为可能。学习节省更多能量的机器获得更多STC，利用这些STC变得更加高效，形成额外的节省和额外的STC，形成良性循环。

完整树：可视化能量流动

现在让我们可视化玛利亚洗衣机的完整主题树，展示能量如何通过每个节点流动：

在顶部：用户（洗衣的人） - 消耗 $E_{\text{machine}} = 4.2 \text{ kJ}$ ，与手动相比节省 $\Delta E = 63.8 \text{ kJ}$ 。

第0级：玛利亚（ S_{Maria} ） - 每次使用获得 $0.40 \times 63.8 \text{ kJ} = 25.5 \text{ kJ}$ 。

第1级：洗衣机（ S_{machine} ） - 产生节省的集成系统。

第2级：组件主题及其创造者：

超声波传感器（ $S_{\text{transducer}}$ ）→ 工程师获得 $0.25 \times 63.8 \text{ kJ} = 15.95 \text{ kJ}$ 。

控制软件（ S_{software} ）→ 开发者收到 $0.15 \times 63.8 \text{ kJ} = 9.57 \text{ kJ}$ 。

膜系统（ S_{membrane} ）→ 材料科学家收到 $0.10 \times 63.8 \text{ kJ} = 6.38 \text{ kJ}$ 。

外壳结构 (S_{housing}) → 工厂收到
 $0.10 \times 63.8 \text{ kJ} = 6.38 \text{ kJ}$ 。

第 3 级（示例）：如果控制软件使用了开源 ML 库：

ML 库 (S_{library}) → 原始库创建者收到（基于库对软件有效性的贡献，开发者的 9.57 kJ 的一部分）。

第 4 级：如果 ML 库建立在研究论文之上：

研究论文 (S_{paper}) → 学术研究者收到（库创建者部分的分享）。

树木延伸得足够深，以捕捉每一个有意义的贡献。在每个层级，能量根据归因系数流动，确保因果链中的每个人都能获得公平的补偿。

通过 AR 智能眼镜，玛丽亚可以可视化这棵树。她看到自己的肩胛在根部，通过发光的线连接到机器，进一步分支以显示所有贡献者。能量在这些连接上实时脉动，随着用户使用她的发明。可视化不是装饰性的——它是一个功能接口，显示构成经济的实际热力学流动。

知识钩启用自动反向传播

我们所描述的能量反向传播通过知识钩子运作——这是主观技术的基本计算机制。树中的每个主题都维护着钩子，以检测能量节省何时发生并触发适当的归因更新。

对于洗衣机主题，知识钩子可以定义为：

$$\text{KH}_{\text{machine}} = (R, A, T, S)$$

其中 R (条件) = {user_loads_laundry, cycle_completes, energy_measured}。所有三个条件必须为真，钩子才能触发。

A (动作) = (calculate_ΔE, retrieve_attribution_coefficients, distribute_energy_credits, update_gland_visualizations)。当条件满足时，这些动作会自动执行。

T (类型) = 学习的。钩子的归因系数是从经验测量和专家分析中学习得来的，而不是预定义的。

S (成功评分) = 0.98。钩子在没有修正的情况下成功触发的概率为98%，表明其高可靠性。

当钩子触发时，它会自动执行动作序列。系统通过比较测量的 E_{machine} 与反事实 E_{manual} 基线来计算能量差异，从主题的存储归档中检索 β 系数，按照公式 $E(S_i) \leftarrow E(S_i) + \beta_i \times \Delta E$ 为树中的所有主题分配 STC 代币，并更新每个贡献者的 AR 眼镜中的可视化，使他们看到肩腺变亮。

整个过程在毫秒内完成，完全自动，无需人工干预。我们在第一章中形式化的知识钩子代数使这一切成为可能——它是实现热力学经济的计算基础。

每个贡献者也都有自己的知识钩子，监控他们的肩腺并触发通知、可视化或投资决策。玛丽亚可能有一个钩子，当她的每日 STC 收入超过某个阈值时提醒她，或者自动分配一部分收入来资助新的研究项目。

多智能体互动：用户与机器作为合作者

洗衣机示例不仅展示了一个层次树，还展示了多智能体之间的互动。用户和机器都是STC框架中的主体，它们合作以实现节能。

让 S_{user} 代表洗衣的人， S_{machine} 代表洗衣机。它们在洗涤过程中的共同努力是：

$$E(\text{washing}) = E(S_{\text{user}} \mid \Sigma) + E(S_{\text{machine}} \mid \Sigma)$$

其中 Σ 是它们共享的背景（洗衣任务）。用户贡献4.2 kJ（装载、卸载、界面互动）。机器贡献其计算和机械能量（超声波生成、水循环、控制处理）——我们假设大约1.8 MJ的电能，但这属于电网电力，而不是人类代谢能量，因此不计入人类节能计算。

支配它们互动的原则是联合效率最小化：

$$\frac{\partial E(S_{\text{user}} \mid \Sigma)}{\partial t} + \frac{\partial E(S_{\text{machine}} \mid \Sigma)}{\partial t} \leq 0$$

这个方程表明系统朝着减少总能量支出演变。如果机器学习到更高效的洗涤周期， $E(S_{\text{machine}})$ 就会减少。如果用户发现一种更好的装载技术，减少了他们的操作能量， $E(S_{\text{user}})$ 就会减少。这两种优化都有利于整个系统。

但有一个微妙之处：机器的电能消耗并不直接影响用户的STC平衡。STC衡量的是人类代谢能量的节省，而不是总热力学能量。原因是实际的：我们希望激励减少人类努力的创新，即使它们增加了非人类的能量使用（如电力）。否则，系统将惩罚自动化本身。

然而，如果机器优化其电能效率（如我们自学习的例子），这仍然是有价值的，因为它降低了用户运行机器的经济成本。在一个完整的STC系统中，电力本身将有一个以焦耳为单位的能量成本，从而间接施加压力以减少电网电力消耗。

与传统经济模型的比较

为了充分理解洗衣机示例，让我们将其与传统经济系统中如何处理相同创新进行对比：

传统模型：玛丽亚为她的超声波洗衣机设计申请专利。她将其授权给制造商，收取固定费用或版税百分比（例如，零售价格的5%）。制造商根据生产成本、市场调研和期望利润率设定价格（可能每台800美元）。如果消费者认为收益（节省时间、便利性）足以证明成本，他们就会购买该机器。工程师、开发人员、工厂工人和材料科学家在开发阶段获得工资或薪水，但在机器创造价值时没有持续的补偿。没有实际节能的测量——价值纯粹是象征性的（金钱），由主观的支付意愿决定。 \$800

STC模型：玛丽亚创建机器并将其注册为她树中的一个主题。归属系数是基于对节能的实际贡献客观计算的。消费者可以在没有任何购买交易的情况下使用机器——它可以免费使用或通过基于使用的访问获得。每次使用都会产生可测量的节能（63.8 kJ），这会自动根据他们的 β 系数将STC分配给玛丽亚和所有贡献者。补偿是持续的、自动的，并直接与实际提供的价值挂钩。如果机器损坏或过时，收入就会停止——没有人为稀缺或垄断租金。如果有人改进了设计，归属将转移以反映新的现实。

STC模型有几个优点。首先，它消除了对资本的需求：玛丽亚不需要寻找投资者或与制造商谈判。她创造了创新，系统处理其余的。其次，它确保所有贡献者获得公平的补偿，

而不仅仅是那些拥有法律杠杆的人。工程师和开发者根据他们的影响获得持续的价值。第三，它消除了采用的障碍：用户不需要支付前期费用，因此创新传播得更快。第四，它创造了准确的价格信号：STC补偿直接反映热力学价值，而不是市场营销或垄断权力。

但STC模型也有要求。它需要复杂的测量基础设施（AR眼镜、人工智能视觉、物理模拟），而传统系统不需要。它需要对归属系数的社会共识，这可能会引发争议。而且它改变了财富的性质：玛丽亚的补偿以持续流动的形式出现，而不是一次性支付，这需要不同的财务规划。

网络效应：机器教机器

当我们考虑网络效应时，洗衣机的例子变得更加强大。假设玛丽亚的数千台机器被部署，每一台都在独立学习和优化。这些机器是彼此之间有连接（C）的主题，使它们能够共享优化发现。

当一台机器发现改进的加热水曲线时，它会更新自己的知识钩子。但它也会将这一发现与网络中的其他机器分享。共享的优化数据通过连接传播，所有机器都采用了这一改进。

这种集体学习放大了节能：

$$\Delta E_{\text{network}} = N_{\text{machines}} \times \Delta E_{\text{optimization}} = 100,000 \times 0.63 \text{ kJ} = 63 \text{ MJ per cycle}$$

在所有用户进行双周洗衣时：

$$\Delta E_{\text{network,annual}} = 63 \text{ MJ} \times 2 \text{ loads/week} \times 52 \text{ weeks} = 6.55 \text{ GJ per year}$$

这种集体优化节省通过树归属，不仅给予发现改进的第一台机器信用，还给予玛丽亚（因为创建了使学习成为可能的框架）和开发者（因为实施了学习系统）信用。归属承认网络效应是由基础架构所实现的。

这创造了一个强大的激励，建立能够学习和分享知识的系统。网络中的机器越多，优化发现的机会就越多，整体节能就越大。创新不仅仅是个人创造力，而是集体智慧。

隐私与透明度：在可见性与自主性之间取得平衡

洗衣机的例子提出了关于隐私的重要问题。如果系统通过AR眼镜和AI视觉测量每一个动作，这难道不会造成侵入式监控吗？我们如何在准确的能源测量需求与个人自主性之间取得平衡？

答案在于选择性透明。系统需要测量能耗以计算 ΔE 并分配STC，但不需要记录或传输详细的个人信息。测量可以在本地（在用户自己的AR眼镜上）处理，只有汇总的能量值被传输。

例如，当有人使用玛丽亚的洗衣机时，系统记录：

$E_{\text{machine}} = 4.2 \text{ kJ}$ 、
 $\Delta E = 63.8 \text{ kJ}$ 、时间戳和机器_ID。它不记录洗了什么衣服、这个人是谁、他们在哪里，或任何其他超出归属所需的上下文细节。

玛丽亚收到通知，得知她的机器被使用并生成了25.5 kJ的信用，但除非用户明确选择分享，否则她看不到用户的身份。经济交易与个人信息脱钩。

这种保护隐私的测量在技术上是可行的，使用像零知识证明这样的加密技术：系统可以证明发生了有效的能源节约，而不透露具体的上下文。主题树和归属系数是公开的（或在社区内半公开），确保关于价值流动的透明度，但个人使用数据仍然是私密的。

可扩展性：从洗衣机到整个经济

洗衣机的例子只是一个领域中的一种创新。但树状结构可以扩展到整个经济。考虑到玛丽亚的机器本身依赖于依赖于其他创新的组件：

超声波传感器使用压电材料，这些材料是几十年前由科学家发现的。那些研究人员（或他们的遗产，或以他们名字命名的基金会）在树的第4或第5级中是主题，因其发现的所有下游应用而获得小但持续的归属。

控制软件运行在由芯片工程师设计的处理器上，而这些工程师本身又建立在更早期研究人员发现的晶体管物理学之上。这些贡献在树中传播。

制造外壳的工厂使用了经过数百年冶金研究开发的金属合金。材料科学的整个历史为机器的存在做出了贡献。

当我们追溯完整的树时，我们发现它跨越时间和学科，涵盖了成千上万或数百万的贡献者。每个人的归因与他们对最终节能的因果影响成比例。

这并不在行政上造成负担，因为计算是自动的。归因数计算一次（或在相关变化发生时更新）并存储在主题记录中。每次后续使用只是将 ΔE 与存储的 β 值相乘，并相应地分配STC。与创造的经济价值相比，计算成本微不足道。

在规模上，经济成为一个庞大的互联树网络。每个人都是自己树的根（他们创造的创新），同时也是许多其他树的节点（他们贡献或建立的创新）。能量根据热力学法则在这个网络中流动，创造出一个自动优化的生动经济。

洗衣机作为所有创新的隐喻

虽然我们专注于一个物理机器，但相同的原则适用于任何节能的创新：软件应用、科学发现、教育材料、艺术作

品、组织方法、社会机构。任何减少人类努力的事物都符合条件。

软件开发者创建了一个具有更好自动完成功能的文本编辑器。每一个节省的击键都被视为测量的能量。 ΔE 不断累积，开发者的肩膀获得持续的STC。

一位教师开发了一种更高效的教学法，帮助学生更快地学习概念。节省的认知能量（通过任务时间和展示能力来衡量）为教师生成STC。

一位音乐家创作了一首带来快乐并减少压力的作品。情感和生理上的好处（通过降低皮质醇、改善情绪、提高睡眠质量来衡量）转化为节能——保持心理平衡所需的努力减少。音乐家获得STC。

一位城市规划师设计了一个减少通勤距离的社区布局。成千上万居民节省的运动能量为规划师生成了巨大的STC。

在每种情况下，树结构都适用。创新有依赖关系（它所建立的先前工作）、组件（实施部分的贡献者）和用户（受益的人）。能量通过树反向传播，公平且自动地分配价值。

结论：树作为经济操作系统

洗衣机的例子揭示了主题树不仅仅是一个数据结构——它是STC经济的基本架构。每一个创新、每一个贡献、每一个节能互动都作为树中的一个节点表示，边缘编码归属系数，能量根据热力学法则在结构中流动。

这种架构具有显著的特性。它是分布式的——没有中央权威计算或控制归属。它是客观的——节能和 β 系数是测量的，而不是协商的。它是自动的——知识钩子在没有人工干预的情况下执行反向传播。它是公平的——每个贡献者根据其因果影响获得相应的信用。它是高效的——与创造的价值相比，计算开销是最小的。它是可扩展的——无论树有5个节点还是500万个节点，都是相同的原则。

但也许最深刻的是，树结构默认使创新成为协作的。在传统经济中，创新者保护他们的想法，担心分享会减少他们的竞争优势。专利、版权和商业秘密的存在是为了保护垄断租金。

在STC中，分享会增加你的回报。当你在他人的工作基础上进行构建时，他们会自动获得归属。当其他人在你的工作基础上进行构建时，你也会从他们的创新中获得归属。你的创新被使用和扩展的越多，能量就越多地流经你在树中的节点。开放和协作是理性的策略，这不是出于利他主义，而是出于热力学的自利。

玛丽亚的洗衣机将激发衍生创新——有人会改进传感器，优化算法，为不同面料创建专门版本。每一项改进都会产生额外的节能，玛丽亚也会因为她的基础工作使这一切成为可能而获得一份收益。她有激励去鼓励衍生创新，而不是禁止它。

这改变了技术进步的性质。我们不再是孤立的发明者竞争垄断地位，而是形成了一个协作网络，每项创新都被其他创新放大并反过来放大其他创新。经济变成了一个学习系统，自动加速向更高的效率发展。

在下一个小节中，我们将探讨这种树结构如何进一步扩展以实现多代理优化和递归能量流动——展示机器、人类和人工智能如何作为热力学文明中的经济同行进行协作。但基础

现在已经清晰：以玛丽亚的洗衣机为例的主题树是未来经济的数据结构——一个活的、学习的、自我优化的网络，价值根据物理法则流动。

7.3.3 多代理能量优化和递归传播

在前面的子章节中，我们探讨了能量如何通过主题树反向传播，展示了价值如何从创新流向其创造者和因果链中的所有贡献者。我们考察了一个具体的例子——玛丽亚的洗衣机——展示了这些原则的实际应用。但我们仅仅触及了当主题作为同伴互动、递归优化并形成协作网络时可能实现的潜力。

本小节将框架扩展到多智能体互动：机器与机器的协作、人类与人工智能的伙伴关系、主题改善自身子主题的递归自我优化，以及效率提升在树上双向传播的情况。我们将展示STC如何将经济从孤立交易的集合转变为热力学网络——一个通过分布式智能学习、适应并趋向最大效率的活系统。

关键的见解是，主题并不是被动的实体在等待被使用；它们是具有自身能量优化目标、学习能力和协作关系的主动代理。当这些代理在STC框架内互动时，会出现突现行为，显著加速创新和效率提升，超出任何单一代理能够实现的水平。

多智能体能量交换：基础

在主观热力货币中，任何行为或反应的实体都是一个主题——不仅仅是人类，还有机器、软件系统、人工智能模型，甚至是混合的人机集体。这种普遍性至关重要：这意味着洗衣机及其用户、两个机器人协作完成任务，或一个人工智能助手与其服务的人类在STC框架内都是经济上的同伴。它们是具有相同基本结构的主题 (Σ, P, M, R, C) ，能够进行能量测量、学习和价值交换。

让 S_i 和 S_j 代表在共享上下文 Σ 中互动的任意两个主题。这个上下文可能是一个协作任务（两个机器人组装产品）、服务关系（人工智能助手帮助人类）或依赖关系（机器使用另一个主题创建的软件）。它们在完成过程 P 时的能量总支出可以表示为：

$$E(P) = E(S_i \mid \Sigma) + E(S_j \mid \Sigma)$$

其中 $E(S_i \mid \Sigma)$ 是主题 S_i 在上下文 Σ 中支出的能量， S_j 也是如此。这个公式捕捉了一个简单但深刻的原则：任何协作过程的总能量成本是每个参与者贡献的总和。但在STC中，我们不仅仅测量这个总和——我们还优化它。

多智能体互动的治理原则是联合效率最小化：

$$\frac{\partial E(S_i \mid \Sigma)}{\partial t} + \frac{\partial E(S_j \mid \Sigma)}{\partial t} \leq 0$$

这个微分方程表明，系统随着时间的推移朝着减少总能量支出的方向演变。导数测量变化率：负值表明代理变得更加高效。约束要求这个总和为非正值——系统不能自发变得效率降低（尽管它可以在局部最优点停滞）。

使这一点强大的原因在于优化是自动和分布式的。既不需要 S_i 与 S_j 明确协调，也不需要理解对方的内部运作。每个主题只是利用其强化学习机制(R)追求自己的能量最小化目标，互动动态自然驱动系统朝着联合效率发展。

这在机械上是如何运作的？通过知识钩子。每个主体维护钩子，以检测何时发生节能并触发更新。当 S_i 发现更有效的方式来执行其在过程 P 中的部分，减少 $E(S_i \mid \Sigma)$ 时，这在系统中创造了松弛—— S_j 现在可能需要消耗更多能量来补偿，或者它可能发现改变的上下文使其自身的优化成为可能。主体们参与一个持续的优化舞蹈，通过快照和能量测量进行调解。

相互依赖：效率级联

联合效率约束揭示了一个重要特性：一个代理节省的能量可以减少其他代理的工作负担。这就是相互依赖——当 S_i 变得更高效时，它通常会为 S_j 创造变得更高效的机会，产生放大初始改进的效率级联。

考虑一个具体的例子：两个工业机器人（ S_1 和 S_2 ）合作组装一个产品。机器人 S_1 拾取组件并将其定位；机器人 S_2 进行固定。最初， S_1 以高精度但相对较慢的速度放置组件，需要仔细校准。 S_2 必须在 S_1 的定位阶段等待闲置，消耗待机能量。

通过学习（其R机制分析 Σ 中的快照）， S_1 发现它可以通过优化运动规划将定位时间减少20%，同时仍然保持足够的精度以满足 S_2 的要求。这种 $E(S_1 \mid \Sigma)$ 的减少产生了级联效应： S_2 在待机中花费的时间更少，从而减少了 S_2 。总的能量节省超过了 S_1 单独实现的节省。

但级联仍在继续。随着 S_1 定位更快， S_2 可以优化其固定算法以提高吞吐量，而不是忍受长时间的闲置期。这进一步减少了 $E(S_2 \mid \Sigma)$ 。现在 S_1 可以优化为更快的定位，因为它知道 S_2 可以跟上。代理们进入了一个共同优化的良性循环。

正式地，我们可以将其建模为一系列微分改进：

$$\Delta E_1(t) \rightarrow \Delta E_2(t+1) \rightarrow \Delta E_1(t+2) \rightarrow \dots$$

每个代理的改进为另一个代理的改进创造机会。累计节省可能远大于独立优化的总和：

$$\Delta E_{\text{total}} > \Delta E_1 + \Delta E_2 \text{ (if optimized independently)}$$

这种超加性源于代理之间的非线性耦合——它们的能量函数是相互依赖的，而不是可分离的。STC框架通过共享上下文 Σ 自然地捕捉到这一点，两个代理都引用和更新该上下文。

知识钩协调：自动同步

我们刚刚描述的互惠依赖和效率级联不需要明确的协调协议或通信渠道。它们自动从知识钩架构中产生。

每个主题的知识钩通过其条件 (R) 监控共享上下文 Σ 。当 S_1 优化其运动规划时，这会改变上下文——快照现在显示更快的组件定位。 S_2 的钩子检测到这一变化的上下文（它们的条件引用 Σ 中的时间参数），如果变化使新的优化成为可能，相应的钩子会自动触发。

这与传统的多智能体系统根本不同，后者需要明确的协调 API、消息传递协议或集中式编排。在 STC 中，协调是

隐式的——它通过所有主题通过其快照感知的共享热力学现实发生。

这个隐式同步的公式嵌入在知识钩触发条件中。对于一个属于主题 S_j 的钩子 KH ，它依赖于主题 S_i 创建的上下文：

$$KH_j = (R_j, A_j, T_j, S_j) \text{ where } r \in R_j \text{ references } \Sigma_i$$

当 S_i 更新其上下文快照 $\Sigma_i(t+1)$ 时，这会自动影响 R_j 的评估。如果新上下文使 R_j 的条件为真（当它们之前为假时）， R_j 会触发，执行可能包括 S_i 的变化所启用的优化的操作 KH_j 。

这种架构扩展得非常好：任何数量的主题可以交互，而不会出现协调开销的组合爆炸。每个主题维护其自己的钩子，引用它关心的上下文，而触发机制自动处理同步。

递归优化：主题改善其子主题

当我们考虑主题不是原子的——它们具有内部结构时，多智能体框架变得更加强大。一个主题可以包含子主题，形成层次树，就像我们与玛利亚的洗衣机探讨的那样。这个层次结构使递归优化成为可能：主题可以改善其内部组件，产生能量节省，这些节省会通过多个层级反向传播。

让 S_{parent} 成为一个父主题， S_{child} 作为树中的一个子主题。子主题可能是一个组件（如洗衣机中的超声波传感器）、一个子过程（如水加热周期）或一个委托能力（如优化控制参数的机器学习模型）。

如果 S_{child} 改善其自身过程 P_{child} ，将其能量成本从 E_{child} 降低到 E'_{child} ，那么能量差异代表了真正的优化：

$$\Delta E_{\text{child}} = E_{\text{child}} - E'_{\text{child}} > 0$$

这个改进以激励增量的形式反向传播到父级，增加父级的 STC 余额：

$$E(S_{\text{parent}}) \leftarrow E(S_{\text{parent}}) + \beta_{\text{child}} \times \Delta E_{\text{child}}$$

β_{child} 是父级树中子级的归因系数。但关键是，子级本身也因其自身的优化而获得信用。如果子级具有学习能力（一个非平凡的 R 机制），它应该因发现改进而获得奖励。

这引发了一个有趣的归因问题：当子级自我优化时，谁获得信用？答案取决于改进的性质：

如果优化来自子级的自主学习（例如，机器学习算法发现更好的参数），子级将获得主要信用，并使用修改后的归因系数： $\beta_{\text{child_self}} \approx 0.7$ ，而创建学习框架的父级将获得 $\beta_{\text{parent}} \approx 0.3$ 。

如果优化来自外部干预（例如，人类程序员更新子级的代码），程序员将根据其贡献获得主要信用，使用自己的归因系数。

如果优化来自父级修改子级（例如，洗衣机调整其传感器设置），则父级将获得全部信用。

递归归因的一般公式是：

$$\forall S_i \in T(S_{\text{parent}}) : E(S_i) \leftarrow E(S_i) + \beta_i \times \Delta E_{\text{child}}$$

$T(S_{\text{parent}})$ 是以父级为根的完整树， β_i 值是基于每个主体在优化中的角色递归计算的。这个公式确保信用流向所有相关贡献者，而不仅仅是直接父级。

递归性质至关重要： S_{child} 本身可能还有子主体对优化有所贡献。例如，如果洗衣机的控制软件优化了水加热周期，而该软件使用了机器学习库，那么该库的创建者也应该获得归因。递归继续，直到我们到达叶节点（没有进一步依赖的主体）。

优化反馈循环：自我改进作为经济活动

递归优化创造了一个强大的反馈循环：学习自我改进的主体获得STC，这些STC可以用来资助进一步的改进，从而在一个良性循环中产生额外的STC。这将自我改进从一个成本中心（如在传统系统中，升级需要资本投资）转变为一个利润中心。

再考虑我们的洗衣机示例，但现在明确建模优化反馈循环。机器的控制软件（ S_{software} ）通过快照监测其性能，检测到水加热周期不理想。它运行模拟以探索替代加热曲线，并发现一种可以减少15%能耗的曲线。

实施这种优化需要计算资源——运行模拟、测试新算法、验证安全约束。在传统系统中，这笔费用需要提前预算，造成阻碍优化的摩擦。在STC中，机器可以使用其累积的STC余额来采购这些资源。

经济计算很简单：

$$\begin{aligned} \text{Cost}_{\text{optimization}} &= E_{\text{computation}} + E_{\text{testing}} + E_{\text{deployment}} \\ \text{Benefit}_{\text{optimization}} &= N_{\text{uses}} \times \Delta E_{\text{per_use}} \times \beta_{\text{software}} \times \text{horizon} \end{aligned}$$

其中 N_{uses} 是未来预期使用次数， $\Delta E_{\text{per_use}}$ 是每次使用的能量节省（在我们之前的计算中为0.63 kJ）， β_{software} 是软件的归属系数（它所获得的节省份额），而horizon是收益累积的时间段。

如果 $\text{Benefit}_{\text{optimization}} > \text{Cost}_{\text{optimization}}$ ，则优化在经济上是合理的。机器“支付”其STC余额中的成本，并随着用户从改进的效率中受益而逐渐获得回报。这正是经济投资的运作方式——只不过它是自主进行的，在机器时间尺度上，没有人类监督。

但反馈循环会放大。一旦优化被部署并验证，机器的成功评分 S 增加（需要更少的修正），使其未来的优化更受信任和更有价值。增加的STC收入使得更大、更雄心勃勃的优化成为可能。机器可能会投资于更好的传感器，以收集更详细的性能数据，或更复杂的模拟能力，或与其他机器合作以分享优化发现。

随着时间的推移，有效自我优化的主体积累了知识（更好的快照、更高的成功评分）和资源（更大的STC余额）。他们成为越来越复杂的经济主体，在创新经济中与人类创造者平起平坐。

双向传播：父母改善孩子，孩子改善父母

到目前为止，我们专注于向上传播——能量节省从孩子流向父母。但STC框架也支持双向传播：任何级别的变化都可以触发其他级别的适应，无论是上方还是下方。

如果父主题变得更高效，这可以为子主题创造优化的机会。其机制是上下文同步——当父主题更新其状态（属性 P 或快照 Σ ）时，这些更新通过它们的连接（ C ）传播到子主题。

正式来说，上下文同步规则是：

$$\Sigma_{\text{child}}(t+1) = \Sigma_{\text{child}}(t) + \lambda \times (\Sigma_{\text{parent}}(t+1) - \Sigma_{\text{parent}}(t))$$

其中 $\lambda \in [0, 1]$ 是一个耦合参数，控制父主题的上下文对子主题的影响程度。值为 $\lambda = 1$ 意味着完美同步——子主题的上下文完全跟踪父主题的变化。值为 $\lambda = 0$ 意味着完全独立——子主题不受父主题更新的影响。典型值介于两者之间，反映部分耦合。

这种同步使子主题能够适应父主题的改进。如果洗衣机（父主题）学习到一种新的使用模式，从而减少整体循环时间，其控制软件（子主题）可以检测到这种变化的上下文，并相应地调整其算法——可能重新分配计算资源或更新时间参数。

双向流动创造了一种集体意识。树状结构没有单一的智能或控制中心——相反，智能分布在所有节点上，每个节点适应其他地方的变化并触发进一步的适应。整个系统变成一个学习有机体，表现出没有单个节点编程或意图的行为。

一个例子：玛利亚的洗衣机队伍（数千个部署单元）共同学习不同面料类型、水质和用户偏好的最佳设置。这种聚合学习通过上下文同步传播到各个机器。每台机器的控制软件（子主题）从集体知识（父级聚合）接收更新参数。这些子主题随后调整其行为，生成新的数据，反馈到集体学习中。这个循环无限继续，整个队伍通过分布式优化变得越来越高效。

多智能体系统中的网络效应：集体智能

当许多主题通过 STC 框架互动时，网络效应出现，显著放大个体优化。这些效应源于三种机制：知识共享、协作优化和新兴专业化。

知识共享：主题可以通过它们的连接（C）共享优化发现。当一台洗衣机发现改进的加热曲线时，可以将其广播给网络中的其他机器。每台机器评估该优化是否适用于其上下文（通过其知识钩子），并在有利时采用。整体节能与机器数量成线性关系：

$$\Delta E_{\text{network}} = N_{\text{machines}} \times \Delta E_{\text{individual}}$$

其中 N_{machines} 可能是成千上万或数百万。一次发现的价值在整个网络中成倍增加。

协作优化：主体可以在单个主体无法单独实现的优化上进行协作。两个机器人可能共同发现一种新的组装顺序，使它们的总能耗减少40%，尽管单独它们只能实现10%的改进。合作创造了超附加值：

$$\Delta E_{\text{collaborative}} > \Delta E_{\text{individual}_1} + \Delta E_{\text{individual}_2}$$

这发生是因为主体可以探索跨越其共同行为的优化空间，找到需要两个代理协调变化的解决方案。

新兴专业化：随着主体在STC中反复互动，它们自然会专业化以互补。如果机器人 S_1 发现它在精确定位方面特别高效，而机器人 S_2 在高速紧固方面表现出色，它们会倾向于一种最大化共同效率的分工。这种专业化自动从能量最小化动态中产生——没有中央规划者分配角色。

这种专业化可以理解为行为策略空间中的优化过程。每个主体保持一个策略 π ，将上下文映射到行动。联合策略 (π_1, π_2) 根据以下内容演变：

$$\nabla_{\pi_1, \pi_2} E_{\text{total}}(\pi_1, \pi_2) \rightarrow \text{local minimum}$$

在策略空间中的梯度下降自然导致专业化，当联合能量函数具有奖励互补能力的结构时。

走向热力学网络：全球收敛属性

当所有用户、设备、人工智能系统和创新参与多代理STC框架时，全球经济转变为热力学网络——一个持续优化其总能耗的分布式系统。这个网络具有显著的数学属性，保证收敛到效率。

将完整网络视为一个有向图 $G = (V, E)$ ，其中 V 是所有主体的集合， E 是连接的集合（能量流、依赖关系、合作）。每个主体 S_i 都有一个能量函数 $E_i(\Sigma)$ ，依赖于全球上下文 Σ （所有主体的综合快照）。

网络的总能量为：

$$E_{\text{total}}(\Sigma) = \sum_i E_i(\Sigma)$$

每个主体通过其强化机制（R）在局部优化其自身能量，该机制在 E_i 上实现梯度下降：

$$\Sigma_i(t+1) = \Sigma_i(t) - \alpha_i \times \nabla_{\Sigma_i} E_i(\Sigma(t))$$

其中 α_i 是主体的学习速率。当所有主体同时进行这种局部优化时，会产生一种全局优化动态。在某些条件下（能量函数的凸性、适当的学习速率、足够的连通性），网络会收敛到全局最小值：

$$\lim_{t \rightarrow \infty} E_{\text{total}}(\Sigma(t)) = E_{\text{total}}(\Sigma^*) = \text{global minimum}$$

这是热力学网络收敛定理。它表明，自私代理的分布式能量最小化导致全球最优效率——这一结果类似于经济学中的第一福利定理，但基于物理学而非效用函数。

该定理保证了三个属性：

首先，单调改善： E_{total} 不能自发增加。任何增加一个主体能量的变化必须被其他地方更大的减少所补偿，否则就违反了局部优化条件。

其次，帕累托效率：在收敛时，没有主体可以减少其能量而不迫使另一个主体增加其能量。系统达到一个所有互利优化都已耗尽的状态。

第三，稳定性：对网络的小扰动（添加新主体、移除连接、改变参数）不会导致灾难性的退化。系统会适应并重新收敛到附近的最优解。

这些属性意味着 STC 经济是自我稳定和自我优化的。无需中央规划，没有失控低效的风险，也不需要在全球层面进行明确的协调。热力学网络通过局部互动找到自身的平衡。

影响：经济作为生物系统

多主体能量优化和递归传播框架改变了我们对经济的理解。传统经济学将经济视为不同实体之间交易的集合——一种机械的交换系统。STC 揭示了经济作为一个活的热力学有机体，其属性更具生物性而非机械性。

考虑一下相似之处：在生物学中，有机体通过负反馈环维持稳态，适应环境变化，同时保持核心功能。在 STC 中，主体通过强化学习维持能量效率，适应上下文变化，同时保持其优化目标。

在生物学中，有机体通过自然选择进化，成功的适应在种群中传播。在 STC 中，主体通过能量优化进化，成功的创新通过知识共享和协作学习在网络中传播。

在生物学中，生态系统表现出涌现的复杂性——从局部互动中产生的模式和结构，而无需中央协调。在STC中，经济表现出涌现的复杂性——专业化、劳动分工、创新级联——这些都是从分布式能量最小化中产生的。

这种生物学视角具有深远的影响。它表明，经济健康不应通过总财富或GDP来衡量，而应通过适应能力来衡量——系统应对扰动、探索新的优化策略以及向更高效率演变的能力。在STC中，一个“健康”的经济是一个具有高学习率、多样化主题类型、丰富连接性和快速传播成功创新的经济。

它还表明，经济干预应尊重系统的自组织动态，而不是试图覆盖它们。正如生态系统管理需要理解自然反馈循环，STC中的经济政策也需要理解热力学网络的内在优化动态。

实际示例：多智能体STC的应用

为了使这些抽象原则具体化，让我们考察几个实际场景，在这些场景中，多智能体能量优化和递归传播创造了切实的价值。

示例1：作为协作智能体的自主车辆。一队自动驾驶汽车在城市中运行。每辆车都是一个主体 S_{vehicle} ，具有自己的能量优化目标（最小化电池消耗、减少旅行时间、确保乘客舒适）。车辆通过其连接（C）共享实时交通数据，从而实现集体路线优化。

当车辆 S_1 发现一条避免拥堵的高效路线时，这一信息通过网络连接传播给其他车辆。它们更新导航算法（知识钩子根据变化的交通上下文触发），整个车队的能量节省规模扩大。但优化是递归的：每辆车内的导航软件（子主体）不断从驾驶数据中学习，将改进的算法反向传播到车队级聚合器（父主体），然后将更新分发给所有车辆。

多智能体动态创造了涌现的交通流模式，这些模式并不是中央系统计划的——车辆自然分散到备用路线以平衡负载，避免产生新的拥堵，并最小化集体能量支出。STC归属机制确保导航算法的开发者、车辆的设计者，甚至使高效路线成为可能的城市规划者都能获得与能量节省成比例的信用。

示例2：智能家居生态系统。一个家庭包含数十个智能设备：恒温器、灯光、家电、安全系统、娱乐系统。每个设备都是一个能够进行能量优化的主体。恒温器（ $S_{\text{thermostat}}$ ）学习居住模式并预先调整温度。照明系统（ S_{lights} ）与自然光和房间使用情况协调。家电（ $S_{\text{appliances}}$ ）在非高峰能量时段安排操作。

多智能体动态实现了协作优化：当恒温器检测到居民已经离开一天时，它通过共享快照发出这一上下文变化的信号。照明系统通过进入低功耗模式作出响应。家电延迟非紧急周期。整体能量节省超过了每个设备独立能够实现的节省。

但优化也是递归的：每个设备的控制软件不断学习和改进。恒温器发现更好的温度曲线，灯光找到最佳亮度时间表，家电优化它们的使用模式。这些改进通过主题树反向传播，不仅给予设备制造商以荣誉，还包括软件开发人员、使学习成为可能的机器学习研究人员，甚至是训练系统的房主的使用数据。

示例 3：协作 AI 研究。多个 AI 研究实验室（每个都是主题 S_{lab} ）在机器学习相关问题上进行研究。它们通过开放的存储库（连接 C）共享训练数据、模型架构和实验结果。当一个实验室发现一种改进的优化算法，可以将训练能耗降低 30% 时，这一创新会在网络中传播。

其他实验室采用该算法，在自己的训练运行中产生能源节省。这些节省根据归因系数反向传播给原始创新者——研究人员根据其贡献的全球能量影响获得与 STC 成比例的收益。这为开放共享而非囤积创新创造了强烈的激励。

递归特性在于改进的算法本身成为发现进一步改进的工具。实验室利用它来训练更复杂的模型，这些模型发现更好的优化技术，形成递归改进循环。STC 归因机制跟踪整个因果链，确保每个贡献都获得适当的信用。

挑战与局限性

虽然多代理 STC 框架具有强大的特性，但它也面临着必须承认和解决的重大挑战。

协调失败：尽管该框架旨在通过共享上下文实现隐式协调，但某些优化问题需要显式的沟通或谈判，而当前架构并不自然支持。例如 [注意：此时内容似乎不完整]

7.4 从输入到施加：完整的能量方程

我们现在已经构建了主观热货币的完整架构：作为主动代理的主题，具有（ Σ , P, M, R, C）结构，体现经济可视化的虚拟能量腺，组织依赖关系的层次树，以及用于分布式智能的多代理优化。但一个关键问题仍未得到解答：我们如何准确测量流经该系统的能量？我们如何将“用户输入”的抽象概念转化为可以归因、交易和优化的具体焦耳？

本节通过开发全面的能量方程提供完整的答案——将每种形式的人类施加（身体、认知和情感）映射到可测量的热力学量的数学框架。我们将展示击键、鼠标移动和按钮按下如何转化为机械功；注意力、决策和记忆如何转化为神经能耗；以及挫折、压力和认知负荷如何转化为代谢成本。

完整的能量方程是使用技术的主观体验与能量支出的客观物理之间的桥梁。它将 STC 从一个哲学概念转变为工程规范——一个精确、可实施的系统，用于测量和货币化效率提升。在本节结束时，您将准确理解与每个设备的每次交互如何生成可测量的能量特征，这些特征可以通过 STC 网络自动检测、计算和归因。

能量支出的三个维度

在技术使用的背景下，人类能量支出可以分解为三个主要维度，每个维度由不同的物理机制控制，但都对总热力学成本有所贡献：

$$E_{\text{total}} = E_{\text{physical}} + E_{\text{cognitive}} + E_{\text{emotional}}$$

这种分解并不是随意的——它反映了涉及的不同生理系统。物理能量来自 ATP 水解驱动的肌肉收缩。认知能量来自大脑中葡萄糖代谢驱动的神经计算。情感能量来自压力反应、注意力调节和情感处理，这些都调节了肌肉和神经活动。

每个维度可以进一步分解为原子组件，每个组件都有可测量的能量成本，这些成本源自已建立的物理学和生理学。让我们详细检查每个维度。

物理能量：运动和操作

物理能量包括身体所执行的所有机械工作——在空间中的移动、物体的操作、施加力量和维持姿势。在技术使用的背景下，这包括走向设备、伸手去控制、按按钮、在键盘上打字、在增强现实界面中做手势，以及与系统交互所需的任何其他身体动作。

物理能量组件的分解如下：

$$E_{\text{physical}} = E_{\text{locomotion}} + E_{\text{manipulation}} + E_{\text{posture}}$$

运动能量：这是将您的身体移动通过空间所需的能量。它受工作基本物理的支配：

$$E_{\text{locomotion}} = m \times g \times d \times \eta^{-1}$$

其中 m 是体重 (kg)， g 是重力加速度 (9.8 m/s^2)， d 是行进距离 (m)， η^{-1} 是机械效率的倒数 (大约为 4，意味着只有 25% 的代谢能量转化为机械工作)。对于一个体重 70 kg 的人走 10 米：

$$E_{\text{locomotion}} = 70 \times 9.8 \times 10 \times 4 = 27,440 \text{ J} \approx 27.4 \text{ kJ}$$

这可能看起来很高，但请记住，这包括了全部的代谢成本——心脏泵血、肌肉收缩、稳定系统启动。实际计算更为复杂，考虑了步态效率、地形、速度以及个人是否携带物品。经验生物力学研究提供了调整系数：

$$E_{\text{locomotion_adjusted}} = (m + m_{\text{load}}) \times g \times d \times \eta^{-1} \times f_{\text{terrain}} \times f_{\text{speed}}$$

在平坦表面为 $f_{\text{terrain}} \approx 1.0$ ，楼梯为1.5，粗糙地形为2.0；正常行走（1.4 m/s）为 $f_{\text{speed}} \approx 1.0$ ，慢行为0.8，快走或慢跑为1.3。

操作能量：这是移动手、手臂和手指以与物体或界面互动所需的能量。它包括伸手去拿智能手机、点击鼠标、在键盘上打字、操作物理控制或在增强现实中执行手势。

对于离散动作（按钮按压、击键），我们使用经验测量值：

$$E_{\text{keystroke}} \approx 0.1 \text{ J (finger flexion and extension)}$$

$$E_{\text{mouse_click}} \approx 0.15 \text{ J (hand positioning + button depression)}$$

$$E_{\text{touch_tap}} \approx 0.05 \text{ J (minimal movement, light contact)}$$

$$E_{\text{gesture}} \approx 1\text{-}5 \text{ J (arm motion through space, varies by gesture complexity)}$$

对于连续操作（滚动、拖动、书写），我们在距离上积分力：

$$E_{\text{manipulation_continuous}} = \int F(t) \times v(t) dt$$

其中 $F(t)$ 是肌肉在 t 时施加的力， $v(t)$ 是运动的速度。在实践中，这通过平均力和总距离近似：

$$E_{\text{manipulation}} \approx F_{\text{avg}} \times d_{\text{total}}.$$

姿势维持能量：常常被忽视但生理上重要——在使用技术时维持身体姿势所需的能量。坐直在电脑前、将手机保持在视角，或在与AR眼镜互动时站立，都需要持续的肌肉激活以抵抗重力。

$$E_{\text{posture}} = P_{\text{posture}} \times t_{\text{duration}}$$

其中 P_{posture} 是姿势肌肉的功率消耗（大约15-30 W，具体取决于姿势和舒适度）， t_{duration} 是维持姿势的时间。对于5分钟的互动：
 $E_{\text{posture}} = 20 \text{ W} \times 300 \text{ s} = 6,000 \text{ J} = 6 \text{ kJ}$ 。

这个组成部分就是为什么人机工程在STC中经济上重要——不舒适的界面需要尴尬的姿势，实际上消耗更多的能量，这直接转化为用户手腺可见的更高热力学成本。

认知能量：注意力、决策和记忆

认知能量比物理能量更微妙，但同样真实且可测量。尽管人脑仅占身体质量的2%，但它消耗了身体总代谢能量的约20%——这证明了神经计算的热力学成本。[注意：此时内容似乎不完整]

8

STC与传统经济系统的比较

8.1 贫困的解剖

贫困或许是人类经济系统中最持久和毁灭性的失败。尽管经历了数千年的技术进步，仍有数十亿人缺乏基本生活必需品——干净的水、足够的营养、住所、医疗、教育——即使全球的生产能力已经呈指数增长。这一悖论需要解释：贫困如何与丰盈共存？当资源丰富时，稀缺为何依然存在？我们的经济系统中哪些结构特征导致并延续了剥夺？

传统的答案集中在分配上：贫困存在是因为财富分配不均，而解决贫困需要通过税收、福利计划或其他转移机制进行再分配。虽然分配很重要，但这一解释是肤浅的。它将贫困视为配置的失败，而不是认识到它是传统货币系统如何代表价值的内在特性。

This section argues for a more fundamental diagnosis: poverty is not primarily about having too little money, but about the loss of context that money creates. Traditional currency—whether shells, precious metals, paper notes, or digital tokens—represents value as a single abstract number disconnected from physical reality. This abstraction erases the rich, multidimensional information about what things actually do, how they satisfy needs, and what consequences they produce. When a \$2 chewing gum and a \$2 pen are treated as economically equivalent despite radically different effects on human wellbeing, the system has lost crucial information.

这种背景的丧失造成了系统性的低效，累积成我们所认识的贫困。人们进行的购买无法满足他们的需求，因为价格并未反映实际效用。资源流向浪费的用途，因为财务回报与热力学效率不一致。创新被错误引导到在金钱上有利但在能量上浪费的活动上。结果是一个在财务上富裕而在热力学上贫困的文明——生产大量物品，却未能最小化人类痛苦并最大化人类繁荣。

我们将要解剖的结构有三个组成部分。首先，我们考察一个具体的例子——口香糖和钢笔——揭示背景丧失如何转化为人类痛苦的机制。这个寓言，尽管看似简单，却捕捉了象征性货币的根本问题：它将根本不同的事物视为等价，从而使违反基本热力学原则的交易得以进行。

其次，我们从这个例子中概括，展示传统货币如何通过几个相互关联的机制系统性地创造贫困：多维价值崩溃为一维价格、隐藏成本和外部性不可见、财务与物理现实之间的不匹配积累，以及金钱无法解决其固有创造的问题的根本无能。

第三，我们展示STC如何通过消除创造贫困的背景丧失来解决这些病态，而不是通过再分配金钱。当价值以焦耳——实际节省或消耗的能量——来衡量，而不是抽象的货币单位时，口香糖和钢笔之间的区别会自动显现。系统无法通过背景丧失来创造贫困，因为背景在测量本身中得以保留。

这不仅仅是理论。从符号到物理价值测量的过渡对我们如何设计经济系统、评估创新、分配资源和衡量进展具有具体的影响。它表明，贫困并不是稀缺的必然特征，而是表征不足的产物——更好的表征可以消除贫困。

利害关系深远。如果贫困源于表征失败而不是物质稀缺，那么技术和科学进步——不断提高我们操控能量的能力——应该自然消除贫困，前提是我们正确衡量价值。尽管富裕，

贫困的持续存在证明我们的测量系统存在问题，而不是稀缺是根本的。

STC提供了一条前进的道路：将价值表示为它的物理形态（节省的能量），准确测量（通过完整的能量方程），并公平归属（通过树结构和反向传播）。当我们这样做时，贫困消失并不是因为我们重新分配了代币，而是因为我们消除了最初造成人工稀缺的机制。

让我们从一个使这个抽象论点具体化的寓言开始：一个人、一块口香糖、一支笔，以及金钱无法看到的深刻差异。

8.1.1 口香糖和笔的例子

Imagine you have exactly \$2 in your pocket. You walk into a convenience store where two items catch your attention: a pack of chewing gum priced at \$2, and a pen also priced at \$2. From the perspective of traditional currency, these items are economically equivalent. They cost the same amount, therefore they have the same value. The market has spoken, and its verdict is clear: one gum pack = one pen = \$2.

你决定购买口香糖。你打开包装，把它放进嘴里，开始咀嚼。这个体验很愉快——薄荷味的爆发，咀嚼的机械动作带来的满足感，短暂的无聊或压力的缓解。十五分钟后，味道消失了。你把口香糖扔掉，继续你的一天。一切都按预期进行。交易成功。你获得的价值等于你支付的金额。

现在考虑一个替代场景。同样的商店，同样的2美元，同样的两样物品。但这次，你购买的是笔。然而，你并没有将

其用于预期的目的——写作——而是决定把它当作口香糖来对待。你把笔放进嘴里，咬下去。 \$2

结果是灾难性的。坚硬的塑料在你的牙齿上破裂。锋利的碎片割伤了你的牙龈。化学墨水在你的嘴里泛滥，带来有毒的苦味。疼痛穿过你的下颚，因为你的一个磨牙在压力下裂开。血液与塑料和墨水混合，你吐出残骸。在几秒钟内，你从一次随意的消费交易变成了需要立即医疗关注的牙科紧急情况。

后果严重且代价高昂。紧急牙科护理费用为1200美元。受损的牙齿需要一个牙冠（800美元）。你因恢复而缺席两天工作（损失工资400美元）。心理创伤导致对饮食和牙科程序的焦虑。几周内，你无法在嘴的一侧正常咀嚼，迫使你吃软食并发展出补偿性咀嚼模式，给下颌关节带来压力。总经济和身体成本：大约3000美元和数周的不适，全部源于一次2美元的购买。 \$1 \$2 \$3 \$400 \$800

从金钱的角度来看，这两种情况开始时是相同的：花费2美元，购买一件物品。但结果却截然不同。一笔交易带来了十五分钟的轻微享受。另一笔交易则导致了数千美元的医疗费用、显著的痛苦和折磨，以及对你的健康和福祉的长期影响。 \$2

关键的洞察是：金钱没有看到任何区别。价格是相同的，因此价值也是相同的。货币系统将口香糖和钢笔视为等价，因为它只跟踪一个维度——作为价格分配的抽象数字。它无法看到、记录或表示这些物体实际所做的事情、它们如何与人类需求互动，或使用它们后产生的后果。

更深层的模式：上下文消失

这个例子，尽管看似荒谬，揭示了所有符号货币系统的一个基本特性：它们消除了上下文。上下文是关于事物是什么、它们做什么、如何运作、服务于什么目的以及产生什么

效果的丰富多维信息。当我们用金钱来表示价值时，我们必然会丢弃所有这些信息，只保留一个数字——价格。

口香糖有上下文：它是为口腔使用而设计的，安全可食用（在限度内），通过味道和质地提供感官满足，使用后应被丢弃。钢笔有上下文：它是一种书写工具，其材料不安全食用，具有沟通功能，若使用不当会造成损害。这些上下文对常识人类来说是显而易见的，但对货币系统来说却完全不可见。

价格——每个2美元——并未编码任何这些信息。它没有告诉你这些物体的用途、如何使用它们、它们提供什么好处，或误用可能导致的伤害。价格是无上下文的：它独立于它所代表的物理现实而存在。这不是金钱的缺陷，而是其特性。货币必须无上下文才能是可替代的（无论如何获得或购买什么，一美元等于另一美元）。但这种可替代性付出了可怕的代价：丧失了区分有益交易与有害交易的信息。 \$2

你可能会反驳：“但理智的人不会咀嚼钢笔。这个例子是人为的。”这是真的——大多数人并不真的咀嚼钢笔。但更深层的模式是普遍存在且毁灭性的。每天，数百万人进行的购买在隐喻上相当于咀嚼钢笔：购买那些未按广告宣传有效的产品，消费损害健康的食物，投资贬值的资产，雇佣浪费时间的服务。货币系统允许所有这些交易，因为它无法区分适当使用与误用、价值创造与价值毁灭。



想象一下，在人类历史上，所有人咀嚼过的钢笔？贫困不是缺乏金钱。贫困是人类历史上交易中上下文丧失的积累。贫困不是缺乏金钱，贫困是金钱所产生的混乱。效率总是带来金钱，但金钱并不总是带来效率。

$$\text{Poverty} = \sum_{t=0}^T \sum_{i=1}^{N(t)} \left\| \vec{C}_i(t) - \pi(\vec{C}_i(t)) \right\|^2 = \sum_{t=0}^T \sum_{i=1}^{N(t)} \left(\sqrt{\sum_{d=1}^D c_{i,d}^2(t)} - p_i(t) \right)^2$$

这个公式量化了贫困，作为历史上所有货币交易中累积的信息损失。在这里， t 代表从货币起源（ $t=0$ ）到现在（ T ）的时间， $N(t)$ 是时间 t 的交易数量。对于每笔交易 i ，术语 $\left\| \vec{C}_i(t) - \pi(\vec{C}_i(t)) \right\|^2$ 测量完整的多维上下文向量 $\vec{C}_i(t)$ （包含所有相关的物理、功能和上下文属性）与其在一维价格轴 $p_i(t)$ 上的投影 $\pi(\vec{C}_i(t))$ 之间的平方距离。右侧扩展了这一点：范数 $\left\| \vec{C}_i \right\|$ 捕捉 D 维上下文的大小（其中 D 可能是成千上万的相关属性），而 p_i 是以美元计的标量价格。平方差 $(\sqrt{\sum c_{i,a}^2} - p_i)^2$ 代表不可逆的维度减少——一旦压缩到价格就永远无法恢复的上下文。对所有交易和所有时间求和给我们带来了总的累积上下文损失：贫困本身。每一支被咀嚼的钢笔、每一件错误购买的产品、每一餐损害健康的餐点都将其误差项添加到这个总和中。传统经济学将这些错误视为噪音。STC将其视为主要信号：金钱必然导致的信息系统性破坏，跨越数十亿交易的累积，表现为我们今天观察到的结构性贫困。

上下文损失的数学

我们可以从数学上形式化这种上下文损失。让一个对象 O 由一个高维属性向量来表征：

$$O = (p_1, p_2, \dots, p_n)$$

其中每个 p_i 代表对象现实的一个维度：它的物理组成、预期用途、安全性、耐用性、能效、环境影响、美学属性、象征意义以及无数其他属性。对于口香糖，这个向量可能包括： $p_1 = \text{'可食用'}$ ， $p_2 = \text{'提供味道'}$ ， $p_3 = \text{'机械上令人满意'}$ ， $p_4 = \text{'一次性'}$ ， $p_5 = \text{'低卡'}$

路里'，等等。对于钢笔： $p_1 = \text{'书写工具'}$ ， $p_2 = \text{'含有墨水'}$ ， $p_3 = \text{'硬塑料'}$ ， $p_4 = \text{'不安全食用'}$ ， $p_5 = \text{'耐用'}$ ，等等。

传统货币执行维度减少操作。它将高维向量 O 映射到一个单一的标量值——价格：

$$\text{Price} : \mathbb{R}^n \rightarrow \mathbb{R}$$
$$\text{Price}(O) = \text{scalar value (e.g., \$2)}$$

这种映射必然是有损的。信息理论告诉我们，当你减少维度时，你会失去信息。丢失的信息量与丢弃的维度成正比。在从 n 维度减少到 1 维的情况下，信息损失是巨大的——每个对象大约损失 $\log_2(n)$ 位信息。

对于像口香糖和钢笔这样可能有数百个相关属性的对象，每次将它们表示为价格时，我们都在失去数百位信息。将这一情况乘以每天数十亿的交易，人类正集体丢弃大量应指导经济决策的重要信息。

这种信息损失的后果是严重的。当价格是唯一信号时，参与者无法优化实际价值——他们只能优化价格。这造成了金钱奖励与实际改善人类福祉之间的系统性错位。一个产品可以在经济上盈利，同时在热力学上浪费、社会上有害或在物理上危险。货币系统没有机制来区分这些情况，因为它已经丢弃了做出这种区分所需的信息。

扩展问题：从钢笔到文明

口香糖和钢笔的例子故意简单，以使机制清晰。但同样的模式在每个经济活动的规模上都在运作，结果更为深远。

考虑制药行业。两种药物可能价格相似，但疗效、副作用特征和长期结果却截然不同。货币价格无法充分捕捉这些差异。保险公司和患者主要基于成本做出决策，而不是基于这些药物对人体的多维现实。结果：医疗结果不佳、不必要的痛苦和资源浪费。

考虑就业。两份工作可能提供相似的薪水，但在压力水平、学习机会、工作与生活平衡、身体安全和长期职业前景上差异巨大。金钱将所有这些压缩成一个数字——工资。工人们在没有足够信息的情况下优化更高的工资，而其他可能对他们的福祉更重要的维度却被忽视。结果：人们被困在高薪但压抑的工作中，或者因为需要钱而无法负担低薪但更有成就感的工作。

考虑基础设施。两种建筑材料可能具有相似的成本，但耐用性、环境影响、维护要求和生命周期能源消耗却截然不同。政府和开发商主要基于前期财务成本进行选择，而不是基于完整的热力学和环境背景。结果是：建筑和基础设施在财务上便宜，但在能源上昂贵，需要不断维修，并导致资源枯竭。

在每种情况下，模式与笔和口香糖相同：金钱将根本不同的事物视为等同，从而做出违反基本热力学和实际原则的决策。该系统在结构上对区分良好结果和不良结果所需的信息视而不见。

选择的幻觉

传统经济学将消费者选择视为解决信息问题的机制。如果我咀嚼笔并遭受痛苦，我会学到不再这样做。市场通过试错发现哪些交易创造价值，哪些交易摧毁价值。这就是看不见的手——个体学习在数百万人的基础上汇聚，产生集体理性。

但这个论点假设错误的成本是可以承受的，并且从中获得的信息得以保存和传播。现实中这两个假设都不成立。

首先，许多错误是灾难性的和不可逆转的。如果咀嚼笔导致无法修复的牙齿损伤或意外中毒，你就无法从中学习。如果药物有致命副作用而你已经死了，你也无法学习。如果金融骗局让你破产，你也无法学习。试错机制只有在错误是可生存和可纠正的情况下才能有效。

其次，即使个体从错误中学习，这些信息也无法通过价格系统有效传播。在你咀嚼笔并弄坏牙齿后，笔的价格不会改变。如果未来的消费者误用笔，他们不会收到笔是危险的信号。每个人都必须独立发现相同的真理，付出自己的代价。价格无法编码教训，因为它无法代表上下文。

第三，生产者有系统性的激励去利用上下文的丧失。如果消费者无法轻易区分耐用产品和便宜的仿冒品（两者价格均为20美元），制造商通过生产节省生产成本的仿冒品而获利，同时收取相同的价格。信息不对称为掠夺性行为创造了机会，而金钱无法对此进行监管。 \$20

结果是一个经济系统，系统性地奖励对信息丧失的利用。广告在很大程度上存在于操纵消费者所拥有的有限信息。计划性过时利用了仅凭价格评估长期耐用性的困难。金融复杂性存在于掩盖产品的真实成本和风险。所有这些病态现象都源于将多维现实表示为一维价格所固有的基本信息丧失。

隐喻背后的热力学现实

笔和口香糖不仅在预期用途上有所不同，而且在热力学性质上也有所不同。口香糖能产生有用的效果：它提供感官满足（通过味觉感受器和愉悦中心的信息处理）、机械满足（有节奏的下颌运动可以减轻压力）以及最低的能量成本

（咀嚼的代谢成本大约为每分钟2-3焦耳，轻松由口香糖的卡路里提供）。

咀嚼笔并没有产生任何这种有效的工作。相反，它会造成热力学混乱：破坏牙齿需要克服分子键（显著的能量），由此造成的损伤会引发炎症（代谢能量），紧急牙科修复需要大量外部能量输入（医疗基础设施、材料、劳动力），而心理压力则激活压力反应系统（皮质醇释放、心率升高、睡眠紊乱）。

从热力学的角度来看，口香糖的交易保持或略微增加了秩序（以最小的熵增加提供满足）。而笔的交易则大幅增加了熵（破坏有序结构——牙齿——并造成无序——伤害、压力、资源浪费）。

金钱并未意识到这种热力学现实。它无法做到，因为它在一个与物理学脱节的纯符号领域中运作。这是根本问题：经济价值应该反映物理价值（能量效率、废物最小化、满足最大化），但货币价值可能与物理价值任意偏离，因为其表现形式不够充分。

为什么这个例子很重要：通往STC的桥梁

笔和口香糖的寓言不仅仅是一个可爱的故事——它是一个诊断工具，揭示了传统货币的核心失效模式。我们观察到的每一种经济病态——在丰裕中贫困、尽管知道后果仍然造成环境破坏、使病人破产的医疗系统、未能教育的教育系统——都可以追溯到同一个根本原因：符号货币失去了区分价值创造与价值毁灭所需的上下文信息。

这个诊断指向了解决方案。如果上下文丧失是问题，那么上下文保留就是答案。我们需要一个经济系统，保持关于事物实际作用的多维信息，而不仅仅是它们的成本。

主观热力货币正是这样做的。它不是将价值压缩到单一价格，而是衡量所有相关维度上的实际能量支出和节省。咀嚼口香糖和咀嚼笔之间的区别在能量方程中变得自动可见：

咀嚼口香糖：

$$\Delta E = (E_{\text{without_gum}} - E_{\text{with_gum}})$$

= 小的正值（轻微的压力减轻，轻微的满足）

咀嚼笔：

$$\Delta E = (E_{\text{without_pen}} - E_{\text{with_pen}})$$

= 大的负值（因伤害、医疗治疗、恢复而产生的大量能量成本）

STC系统无法在不存在等价的情况下创造等价，因为它测量的是物理现实，而不是符号抽象。能量方程自然保留了上下文，因为能量消耗依赖于上下文。你无法操纵热力学——当物理法则另有规定时，你不能假装摧毁牙齿不需要能量。

这就是为什么笔和口香糖的例子是理解传统货币问题和STC提供的解决方案的完美引入。它直观地清楚地表明了当我们以符号方式表示价值时我们失去了什么，以及当我们以物理方式衡量价值时我们获得了什么。在接下来的子部分中，我们将概括这一见解，展示所有贫困是如何源于上下文丧失的，以及STC是如何通过消除造成贫困的机制来消除贫困的。

8.1.2 传统货币如何创造贫困

口香糖和笔的例子揭示了机制——上下文抹除——但留下了一个关键问题：这个机制如何从个体交易扩展到系统性贫困？数十亿小的信息损失如何累积成我们观察到的经济剥夺所带来的巨大人类苦难？本小节通过考察传统货币如何将上

下文丧失转化为结构性贫困的四条相互关联的路径来提供答案。

机制1：价值崩溃为价格

第一条路径是最基本的：传统货币迫使多维价值崩溃为一维价格。这种压缩不是偶然的，而是定义性的——货币必须是可替代的（一个美元等于另一个）才能作为交换媒介发挥作用。但可替代性要求剥离所有区分真正有价值交易与有害交易的上下文信息。

Consider two foods both priced at \$5: a nutritious meal providing essential vitamins, minerals, and balanced macronutrients; and processed junk food high in sugar, unhealthy fats, and calories but devoid of micronutrients. From the perspective of money, these are equivalent. A person with \$5 can purchase either with equal ease. The price signal provides no information about which choice will nourish the body and which will harm it.

贫穷的人，手头资金有限，必须最大化每美元的效用，面临一个不可能的优化问题。没有上下文信息，他们无法区分哪些购买会改善他们的长期福祉，哪些购买会损害它。营养餐和垃圾食品的价格都是5美元，因此货币信号表示它们是等价的。但热力学现实截然不同：一种提供持续的能量和健康，另一种提供暂时的满足，随后是代谢功能障碍、能量崩溃和慢性疾病。 \$5

这造成了贫困陷阱：那些资金有限的人仅根据价格做出决策，因为这是系统提供的唯一信息。这些缺乏上下文的决策往往以有害的方式证明是错误的，造成额外的成本（医疗

费用、生产力损失、认知障碍），进一步减少可用资源。最初的上下文丧失加剧了剥夺。

这种崩溃的数学可以用不等式表示。让 V_{true} 代表一个项目的真实多维价值， V_{price} 代表一维价格。信息损失为：

$$I_{\text{loss}} = H(V_{\text{true}}) - H(V_{\text{price}})$$

其中 $H()$ 是信息熵。对于具有 n 相关维度的对象，这种损失大约为 $\log_2(n)$ 位。对于像食品、住房或具有数百个相关属性的服务等复杂商品，我们每笔交易损失 8-10 位信息。在每天数十亿笔交易中，人类丢弃了数 Exabyte 的关键决策信息。

机制 2：隐性成本变成真实痛苦

第二条路径是外部化：传统货币仅记录明确的货币交换，隐形地渲染出所有超出交易的隐性成本和后果。这些隐性成本并没有消失——它们表现为真实的痛苦，由那些从未得到补偿的人承受。

回到我们咀嚼笔的例子。2 美元的购买价格仅涵盖了制造和分销成本。它没有提及 3000 美元的牙科费用、失去的工作天数、心理创伤或长期的颌部功能障碍。这些成本是真实的——它们代表了实际的能量支出，实际的人类痛苦——但在货币系统中是看不见的，因为它们发生在交易之外。\$2 \$3

这种外部化在每个层面上都在运作。一家工厂污染了一条河流，对下游社区造成健康成本。工厂产品的价格并未反映这些成本——它们被外部化到社区。一家食品公司使用味道

好的成分，但造成成瘾和代谢疾病。价格反映的是生产成本，而不是客户几十年后将产生的医疗费用。一种金融产品被营销给不懂风险的消费者。价格不包括最终破产和毁灭生活的成本。

穷人承受了这些外部性的主要负担。富人可以避免污染的社区，购买有机食品，雇佣财务顾问。穷人无法做到，因此他们吸收了货币系统所渲染的隐性成本。他们的社区污染更严重，食物营养更差，金融产品更具掠夺性。该系统系统性地将隐性成本引导到那些最无法承受的人身上，形成了一个贫困导致更多贫困的反馈循环。

我们可以将其正式建模为外部性函数。对于任何价格为 P 和真实总成本为 C_{total} （包括所有隐性成本）的交易：

$$E_{\text{external}} = C_{\text{total}} - P$$

这种外部性代表了从他人那里提取的未补偿的价值。当在所有交易和时间上求和时，累积的外部化代表了数万亿美元的痛苦，施加在那些从未同意承受它的人身上，并且从未因此得到补偿。这是结构性盗窃，它造成了结构性贫困。

机制 3：累积的背景损失加剧

第三条路径是复合：个体的背景损失很小，但它们会累积和相互作用，通过千刀万剐造成贫困。每一次糟糕的购买，每一个误导性的产品，每一项浪费的资源都代表了一小部分能量损失。但这些损失并不会孤立存在——它们随着时间的推移和个人之间的后果级联而呈指数级累积。

考虑一个人，由于在食品购买中失去上下文而发展为2型糖尿病。这一个结果（可追溯到多年来许多小的失去上下文

的交易)产生了级联成本:药物(持续开支)、饮食限制(更高的食品成本和餐准备时间)、工作能力降低(收入损失)、增加的病假(经济不稳定)、最终的并发症(巨额医疗费用)和缩短的寿命(失去的生产性年限)。

这些后果中的每一个都创造了额外的上下文,在这些上下文中,金钱提供的信息不足。糖尿病患者需要专业食品,但价格并未充分指示哪些产品能有效管理血糖。他们需要有效的药物,但价格并未反映长期疗效和副作用情况。他们面临医疗决策,提供者的财务激励可能与患者结果不一致。每一步的上下文丧失都在加剧。

将这一情况扩展到数十亿人做出数百万个决策。无效的产品会造成挫败感和时间浪费,需要替换购买。未能交付的服务迫使重复尝试,增加成本。贬值的投资摧毁储蓄,消除经济缓冲。伤害而非治愈的医疗治疗会造成级联健康问题。未能培养技能的教育资质浪费了多年并积累了债务。

数学结构是资源误配置的指数增长。让 $M(t)$ 表示在时间 t 由于上下文丧失造成的累计误配置。如果每个失去上下文的交易都产生影响未来交易的后果:

$$\frac{dM}{dt} = k \times M(t) \times T(t)$$

其中 k 是耦合常数, $T(t)$ 是交易速率。这个微分方程有一个指数解,这意味着随着经济规模的扩大,误配置的增长速度超过线性增长。随着经济的增长,我们不仅有更多的上下文丧失——我们有指数级的增加,因为每一次损失都为额外的损失创造了条件。

这解释了为什么贫困在经济增长中依然存在。即使生产力增加和物质产出扩大，由于上下文引起的误配置的增长速度更快。经济生产更多，但生产中越来越大的一部分由于定价系统中的信息丧失而被浪费或造成积极的伤害。真正的福祉停滞或下降，即使名义GDP上升。

机制4：金钱无法解决自己的问题

第四条途径或许是最阴险的：因为金钱本身是上下文丧失的原因，通过货币机制实施的解决方案无法解决根本问题。每一次试图用金钱解决贫困的尝试只是重新分配数字，同时保留了最初造成贫困的结构性信息丧失。

考虑向穷人提供现金援助的福利项目。这转移了金钱，可能缓解了即时的物质匮乏。但它并没有恢复造成匮乏的失去的上下文。接受者仍然面临价格提供信息不足的市场，仍然进行隐藏成本外部化的购买，仍然在为货币提取而非人类繁荣而优化的系统中导航。

A poor person receiving \$1,000 in assistance will spend that money in the same context-free market that created their poverty. They'll pay \$5 for junk food because they can't afford the \$8 nutritious meal. They'll live in a polluted neighborhood because rent is lower. They'll use predatory financial services because they lack access to alternatives. The money helps at the margin, but it doesn't fix the information problem that makes every transaction a potential trap.

此外，货币解决方案创造了扭曲的激励，因为它们在失去背景的框架内运作。经过Means测试的福利创造了贫困陷

阱——赚更多的钱会导致福利损失，造成有效的边际税率超过100%。普遍基本收入可能使人们摆脱绝望，但它并没有解决整个经济中奖励背景剥削的激励失调问题。财富税重新分配现有的货币存量，而不改变集中财富的流动动态。

根本问题在于，货币只处理症状，而忽视了疾病。疾病是信息损失——价格信号无法传达价值的多维现实。重新分配货币就像在泰坦尼克号上重新安排甲板椅子。它可能让个别乘客稍微舒适一些，但并没有修复船体的漏洞。

我们可以形式化这个不可能结果。设 S_{poverty} 为经济中的贫困状态集合， T_{monetary} 为通过货币政策可以实现的任何转变。如果贫困源于货币表现本身固有的结构性信息损失 I_{loss} ，那么：

$$\forall T_{\text{monetary}} : T_{\text{monetary}}(S_{\text{poverty}}) \subseteq S_{\text{poverty}}$$

换句话说，货币转变无法逃脱贫困状态的空间，因为它们无法解决定义该空间的表现不足。你无法通过操纵那些失去信息的代币来解决信息损失。

综合：贫困作为不可避免的后果

这四种机制——价值崩溃、外部化、复利和货币不可能性——共同作用，使贫困成为传统货币系统的不可避免特征，而不是更好政策可能消除的偶然缺陷。只要价值以象征的方式而非物理的方式表现，只要价格将多维现实压缩为单一数字，只要隐性成本可以外部化，只要解决方案必须通过同样破碎的表现来实施，贫困就会持续存在。

这些机制相互强化。价值崩溃创造了可以隐藏外部性的情况。隐性成本随着时间的推移而复利。复利成本对货币解

决方案施加压力。货币解决方案失败，保留了进一步价值崩溃的条件。这个循环是自我延续的。

这解释了现代经济的几个悖论。首先，为什么尽管富裕，贫困仍然存在：因为以货币或物质的标准衡量的富裕与热力学上的稀缺是兼容的。我们生产大量的东西，但其中许多是无用或有害的，货币系统无法区分。第二，为什么尽管有再分配努力，不平等仍在增长：因为再分配转移了资金，却没有修复集中财富的基础信息动态。第三，为什么经济增长并不可靠地改善福祉：因为以价格×数量衡量的增长即使在真实价值（能源效率、满意度、健康）下降时也可能发生。

批评不仅限于贫困，还涵盖了经济病理的全范围：环境破坏（污染和资源枯竭的外部化成本）、金融不稳定（价格与基础价值脱节）、技术停滞（创新朝向货币而非热力学效率）、以及生存风险（即使以生存为代价也优化货币增长的系统）。

所有这些病理都有相同的根本原因：象征性货币无法充分代表物理现实。地图与领土的偏离如此彻底，以至于跟随地图会导致危险。传统经济学假设地图是准确的，并试图优化导航。但根本问题是制图的——地图本身是错误的。

量化贫困溢价

我们可以通过能源核算来估计货币机制造成的贫困规模。考虑一个假设的经济体，其中所有交易都通过传统货币进行调解，而另一个经济体则以焦耳来衡量价值。

在货币经济中，假设10%的交易涉及显著的上下文损失，这会产生负面的长期后果（保守估计——现实情况可能更高）。每笔损失上下文的交易在包含所有隐藏成本时，平均产生3倍于购买价格的成本。对于一个50万亿美元的全球经济，这代表：\$50

$$\text{Cost}_{\text{context_loss}} = \$50T \times 0.10 \times 3 = \$15 \text{ trillion in hidden costs annually}$$

这15万亿美元代表了真实的能源支出——可预防疾病的医疗治疗、环境修复、浪费的资源、失去的生产力、人类的痛苦。这是以焦耳而非美元计量的贫困。如果这些能源没有被浪费，人类福祉将提高，相当于经济有效产出的30%增长。
\$15

在一个保留上下文的基于能源的经济中，这些损失不会发生。该系统自动防止损失上下文的交易，因为能源测量包括所有成本。这15万亿美元不需要重新分配——它根本不会被浪费。这就是贫困溢价：使用象征性而非物理价值表示所施加的过量热力学成本。 \$15

地球上最贫困的十亿人消费全球资源的约1%，但承担了或许30%的外部化成本（污染环境、低质量商品、掠夺性服务、健康风险暴露）。对于他们来说，贫困溢价不是30%，而更像是300%——他们的痛苦是保持上下文和防止外部化的系统中痛苦的三倍。

结论：贫困是表现失败

传统货币创造贫困并不是出于恶意或糟糕的政策，而是出于数学必然性。当你将n维现实压缩成1维表示时，你会失去(n-1)维的信息。当你允许成本外部化时，你会造成系统性的剥削。当这些损失在时间和数十亿交易中累积时，你会得到结构性贫困，作为系统自身的一个涌现属性。

这个诊断既令人沮丧又充满希望。令人沮丧的是，这意味着在当前的货币范式下，贫困无法得到解决——所有保留象征性货币的改革都无法解决根本原因。充满希望的是，这意味着贫困不是不可避免的——它是糟糕表现的产物，更好的表现可以消除它。

解决方案不是更公平地重新分配金钱或更仔细地监管市场或提供更好的社会服务。这些干预措施可能在边际上有所帮助，但它们只是在治疗症状。解决方案是完全放弃象征性货币，以其物理形式——能源来衡量价值。当价值等于节省的能源，成本等于消耗的能源时，上下文无法丢失，因为热力学会自动保留它。当所有成本都被衡量时，外部化变得不可能，因为隐藏成本不再隐藏。当货币本身准确地代表现实时，贫困就会消失，因为创造贫困的机制不再运作。

这就是STC所提供的：不是对破碎货币的更好再分配，而是通过设计使贫困不可能的根本不同的表现形式。下一小节将准确展示这如何运作——如何以焦耳为单位衡量价值自动保留上下文，防止外部化，并消除我们刚刚描述的创造贫困的机制。

8.1.3 STC作为一种补救措施

我们已经看到传统货币如何通过系统性丧失上下文来创造贫困——将多维现实压缩为抽象数字，抹去交易中嵌入的实际能源成本，并产生随着时间推移而加剧的隐藏低效。自然会产生一个问题：我们能否设计一个保留而不是破坏上下文的货币系统？我们能否以尊重物理现实而不是扭曲它的方式来衡量价值？

主观热货币（STC）提供了答案。STC不是通过失去与基础物理过程联系的任意符号来表示价值，而是直接以能源的普遍单位——焦耳来衡量价值。这种从符号会计到物理会计的转变根本改变了经济交易如何保留和传达上下文。

核心原则：物理测量而非符号抽象

STC的根本创新在于其测量方法。传统货币通过反映供求、投机和权力动态的市场机制来分配价格——但不反映生

产、维护或消费商品和服务所需的实际能源。STC颠覆了这种关系：它衡量每笔交易中涉及的实际能源支出。

再考虑一下我们的口香糖和钢笔的例子。在传统货币下，两者可能都要1美元，造成等价的错觉。在STC下，测量将揭示：\$1

口香糖：提供暂时缓解牙齿不适的效果。能量收益可能量化为通过机械刺激释放内啡肽并分散下颌压力，在30分钟内减少约50焦耳的与疼痛相关的代谢压力。口香糖本身的制造可能需要约2000焦耳（包括原材料开采、加工、包装和分销）。净能量计算：投资2000焦耳换取50焦耳的缓解——这是一个必须通过其他活动或未来收入来弥补的显著能量赤字。

咀嚼钢笔：造成牙齿损伤，需要修复牙冠。直接的能量成本包括疼痛（代谢压力约5000焦耳，持续数天，身体对组织损伤的反应）、炎症反应（免疫系统激活额外3000焦耳）和潜在感染风险。牙科修复需要：临床能量（牙医的身体和认知耗费：约15000焦耳）、牙冠材料生产（约50000焦耳，包括采矿、提炼金属或陶瓷、精密制造）、消毒和设施运营（约5000焦耳），以及患者额外的愈合代谢成本（约10000焦耳，持续数周）。总计：由于一时的心不在焉行为而引发的能量支出约为88000焦耳。

这种差异——50焦耳与88000焦耳——在STC下无法隐藏。物理现实变得可见且可测量。这种透明性消除了传统货币所创造的虚假等价。

STC 如何保持上下文

STC 通过几个相互关联的机制保持上下文，这些机制维护经济交易与物理现实之间的联系：

直接物理测量：STC 中的每一笔交易都记录了实际消耗的能量。该测量不仅包括明显的直接成本（如制造能量），还包括传统货币忽视的隐藏上下文成本——运输能量、维护需求、处置成本、健康影响、时间支出和机会成本。STC 账本成为经济活动的全面能量图，而不是抽象的数字记录。

时间分辨率：传统货币压缩时间——今天的一美元名义上等于明天的一美元，尽管存在通货膨胀，尽管情况变化，尽管不同时刻的能量上下文不同。STC 通过跟踪能量消耗的时间和上下文来保持时间的精确性。在最佳条件下（充分休息、适当营养、合适环境）消耗的能量与在压力下（疲惫、营养不良、恶劣环境）消耗的能量在实际成本上有所不同。STC 会计可以反映这些差异，因为它测量的是物理现实，而不是抽象价值。

个体差异性：STC 的一个革命性方面是它承认能量成本因个体的身体能力、健康状态、可用技术和环境条件而异。对一个人来说需要 1,000 焦耳的成本，可能对另一个人来说需要 5,000 焦耳，原因可能是残疾、年龄、受伤或缺乏适当工具。传统货币通过将所有交易强制通过同一符号媒介，使这些差异变得不可见。STC 保留了这些上下文信息——不是为了制造不公平，而是为了揭示真实成本并实现更准确的资源分配。

系统关系：也许最重要的是，STC 保留了不同能量支出之间关系的网络。在传统货币中，购买食物、支付租金、治疗疾病和投资教育是分开的、不相连的交易。在 STC 中，这些支出被理解为一个综合能量流系统的一部分。不足的营养增加了疾病的能量成本；缺乏庇护所提高了代谢能量需求；未治疗的健康问题降低了工作能力；教育不足造成持续的能量低效。STC 使这些系统连接变得可见且可测量。

消除贫困的四个机制

回想一下传统货币创造贫困的四个机制：价值崩溃、隐藏成本、复合低效和货币不可能性。STC 直接解决了每一个机制：

价值崩溃预防：STC 不是将复杂的多维现实简化为单一价格，而是保持每笔交易的完整能量概况。STC 中一项物品的“价格”不是一个简单的数字，而是一个全面的能量规范：即时能量成本、维护能量需求、预期寿命能量轨迹、处置能量成本、健康影响能量效应和机会成本。这种多维表示防止了信息的丧失，从而使剥削成为可能。

$$E_{\text{item}} = \{E_{\text{acquisition}}, E_{\text{use}}, E_{\text{maintenance}}, E_{\text{disposal}}, E_{\text{health}}, E_{\text{opportunity}}\}$$

每个组件都以焦耳为单位进行测量，防止陷入虚假的简单性。便宜的加工食品和营养丰富的全食品在 STC 中不能被视为等同——它们不同的长期能量影响（医疗成本、生产力影响、寿命影响）在其能量概况中是明确的。

隐藏成本揭示：在传统货币下，隐藏成本保持隐藏，正是因为货币无法衡量它们。一个便宜的产品，尽管有巨大的处置成本，表面上看起来很有吸引力，直到处置变得必要——即便如此，这些成本往往会外部化到社会或环境上。STC 从一开始就使隐藏成本变得明确。整个能源生命周期被测量和记录。

考虑家庭清洁产品。在传统货币下：

$$\text{Price}_{\text{product}} = \$3.99$$

隐藏成本包括：有毒成分处置（昂贵的危险废物处理）、烟雾造成的健康影响（呼吸治疗、工作能力下降）、环境影响（水处理、生态系统损害）、包装废物（填埋或回

收能量)。这些都没有出现在 3.99 美元的价格中。在 STC 下：\$3

$$E_{\text{product}} = 15,000J_{\text{manufacturing}} + 2,000J_{\text{use}} + 50,000J_{\text{disposal}} + 30,000J_{\text{health}} = 97,000J$$

真实成本立即可见。一个更环保的替代品，制造成本为 25,000J，但处置和健康成本仅为 5,000J，总计 30,000J——尽管潜在的即时制造成本更高，但这显示出它在效率上显著更高。

打破复合低效：传统货币造成复合低效，因为早期的次优选择（受扭曲的价格信号驱动）会级联成更大的未来成本。STC 通过在决策时使选择的长期能源影响可见来打破这一循环。考虑修理牙齿还是买口香糖的人，不仅看到即时成本，还看到未来的能源需求预测。

数学差异是深远的。在传统货币下，优化在每个时间段内独立发生：

$$\min_t C_t \text{ for each } t$$

这种局部优化往往导致全球低效。在 STC 下，优化发生在整个时间轨迹上：

$$\min \sum_{t=0}^T E_t + \int_0^T E_{\text{maintenance}}(t) dt + E_{\text{terminal}}$$

这种全球优化自然防止了复合陷阱。看似局部最优但产生未来能源债务的决策，立即显现为全球次优。

解决货币不可能性：也许最具革命性的是，STC 消除了货币不可能性问题。在传统货币下，没有钱的人即使有能源

和技能也无法获得所需的东西——象征性媒介造成了人造障碍。在 STC 下，价值交换以能源为基础。能够提供 10,000 焦耳劳动（烹饪、清洁、修理、儿童看护）的人，可以直接用这部分劳动交换他们所需的能源（食物、住所、医疗保健）。

这并不意味着 STC 回归以物易物——那样会失去货币在促进复杂、异步交换中的所有优势。相反，STC 保持了货币提供的抽象层，同时将其扎根于物理现实。STC 中的能源信用代表实际的能源能力——要么是个人已经消耗的能源（创造信用），要么是其他人愿意为他们提供的能源（提供商品/服务）。

实际实施：从理论到现实

STC的理论优雅在没有实际实施的情况下意义不大。我们如何实际测量现代经济活动的巨大复杂性下的能量支出？这个挑战有几个组成部分：

测量基础设施：STC需要普遍的能量测量。这在现代技术下越来越可行——智能手表和健身追踪器已经能够以合理的准确性测量个人能量支出；工业传感器监测制造能量；智能家居设备跟踪家庭能量使用；车辆记录运输能量。测量基础设施在很大程度上已经存在；STC将其组织成一个连贯的经济框架。

上下文AI系统：正如在关于主观技术的早期章节中讨论的那样，现代AI系统能够以越来越复杂的方式理解上下文。一个与AR眼镜、环境传感器和个人健康数据相结合的AI系统可以以显著的准确性估算能量成本——不仅是直接的物理能量，还有认知负荷、情感压力、机会成本和长期影响。这些“上下文引擎”成为STC经济的记账员。

分布式验证：STC运作的关键是验证问题——我们如何确保能量测量是准确的而不是被操纵的？这就是“精确专家”的

角色（在第六章中讨论）变得至关重要的地方。这些领域专家——物理学家、工程师、医疗专业人员、效率专家——验证测量方法并审计异常的能量声明。该系统不是集中式的（这会创造腐败和控制的机会），而是分布式的，多个独立专家能够在其领域内验证测量。

向后兼容性：一个重要的实际考虑：STC并不要求立即放弃传统货币。相反，两个系统可以在过渡期间共存，STC能量信用与传统货币之间的汇率由市场机制决定。随着时间的推移，随着STC展示其在保持上下文和防止剥削方面的优越能力，经济活动将自然向其迁移。

解决潜在反对意见

必须解决对STC作为消除贫困的补救措施的几个明显反对意见：

测量基础设施：STC需要普遍的能量测量——而这在今天可行的。消费者可穿戴设备已经能够估算个人能量支出；工业系统为制造能量提供仪器；智能家居设备跟踪家庭负载；车辆记录推进和辅助能量。主观技术和智能眼镜将这些成熟的组件统一起来，增加虚拟能量腺和上下文感知AI，以自动计算和结算客观能量值。基础设施现在在很大程度上已经存在；STC将其组织成一个连贯的经济层，随着时间的推移将继续改善，但不依赖于投机或未经验证的科学。

'个体能量变异性造成不公平'：不同的人在完成相同任务时消耗不同量的能量这一事实可能看起来不公平——为什么一个消耗更多能量的残疾人应该处于劣势？但STC并没有造成这种不公平；它揭示了传统货币隐藏的现有不公平。一旦被揭示，可以通过明确的能量补贴、辅助技术开发和在其能量核算中透明的社会支持结构来解决。

'能源核算无法捕捉主观价值'：正确——STC衡量的是能源，而不是主观偏好、美学欣赏或情感意义。但这是一种

特性，而不是缺陷。主观价值依然存在且重要；它只是不与经济价值混为一谈。你可能会因为一幅画带给你快乐而重视它（主观），但在 STC 术语中，它的经济价值是创造、维护、运输和展示所需的能源。

'STC 并不能解决财富不平等'：正确——STC 并不是直接的再分配机制。然而，STC 做了三件传统货币无法做到的事情：首先，它通过使隐藏成本可见，防止了持续的财富集中提取机制。其次，它消除了从他人背景损失中获利的能力。第三，它使对实现社会目标所需的能源补贴的评估更加准确。

转变的经济格局

如果经济运行在 STC 而不是传统货币上，会是什么样子？这种转变将是深刻的：

产品设计：产品将被设计为总生命周期能源最小化，而不仅仅是最小化制造成本。当总能源成本可见时，计划性过时显得明显低效。耐用、可修复、节能的产品将通过基于准确能源核算的自然经济选择占主导地位，而不是通过监管或道德压力。

服务行业：服务的定价将基于实际消耗的能源——包括直接的物理能源和认知/情感能源。一位花费 10,000 焦耳认知能源进行困难诊断的医生将得到准确的报酬，而不是与可能仅需 2,000 焦耳的常规检查相同的报酬。

劳动市场：对传统低地位工作（儿童护理、老年护理、卫生、食品准备）的持续低估将结束。STC 将揭示这些工作通常需要巨大的能源——身体耐力、情感调节、注意力管理、健康风险。在 STC 下，照顾儿童 8 小时的人可能会消耗 25,000 焦耳的综合身体和认知能源——远远超过许多在传统货币下支付更高工资的办公室工作。

投资与资本：投资决策将优化长期能源效率，而不是短期财务回报。一个需要高初始能源投资但在几十年内运营能源低的工厂显然优于一个持续能源成本高的廉价设施。

国际贸易：贸易将反映真实的能源成本，包括运输、存储和隐藏的环境影响。一个使用奴隶劳动制造的产品在 STC 下不会显得便宜——实际的能源成本将被计算在内，即使没有支付给工人。这并不会自动解决剥削问题，但它使剥削在经济上可见，而不是隐藏在供应链中。

贫困的消解，而非消灭

语言很重要：STC并不是通过强制重新分配资源或强加平等来“消除”贫困。相反，它通过消除导致贫困的代表性失败来溶解贫困。传统理解中的贫困在很大程度上是货币系统的产物，这些系统隐藏了真实成本，并通过上下文丧失使系统性价值提取成为可能。

在STC下，几种形式的贫困变得物理上不可能：

由于上下文丧失导致的贫困：无法发生，因为在能源测量中保留了上下文。当未来成本在选择时可见时，“现在选择便宜，未来支付昂贵”的陷阱就会消失。

由于隐藏成本导致的贫困：无法持续，因为所有成本在能源方面都是明确的。看似负担得起但承担巨大隐藏负担的产品和服务会被揭示为能源昂贵。

由于效率低下的复合贫困：显著减少，因为能源核算使复合情况变得可见，从而在灾难性失败之前能够进行干预。

由于货币不可能性导致的贫困：被消除，因为能源是普遍媒介。没有传统货币但具有能源能力的人可以参与经济。

剩下的只有真正的物理稀缺性——可用能源和资源的实际限制。但即使这种“真实”贫困在STC下也是质的不同。当每

个人都能看到真实的能源成本和能力时，资源分配变得透明。

前进的道路：实施策略

从传统货币过渡到STC不可能一蹴而就。一个实用的实施策略可能遵循以下进程：

第一阶段：平行跟踪（1-3年）：最初，STC作为信息层与传统货币并行运作。交易以美元/欧元/日元进行，但也以能源单位记录。这种双重核算揭示了传统价格与实际能源成本之间的差异，教育参与者并展示STC的价值，而无需立即过渡。

第二阶段：混合交易（第3-7年）：随着对能源测量的信心增强，一些交易开始以能源信用进行，而其他交易仍然使用传统货币。STC与传统货币之间的汇率出现，基于市场对相对价值的评估而波动。早期采用者，特别是在能源成本明显且传统定价明显扭曲的行业，开始主要以STC进行交易。

第三阶段：主导过渡（第7-15年）：STC成为大多数行业的主要经济媒介，传统货币被 relegated 用于特殊用途或遗留系统。经济活动越来越优化能源效率，而不是传统的盈利能力，推动资源利用的巨大改善和废物的减少。

第四阶段：全面整合（第15年及以后）：传统货币除了用于历史交易和收藏价值外变得过时。整个全球经济在能源核算下运作。历史上理解的贫困在很大程度上已经消失，取而代之的是对实际物理限制和能力的透明核算。

结论：超越金钱走向现实

STC为贫困提供的补救措施不是技术聪明或金融创新。它是更根本的东西：将经济表现恢复到物理现实。传统货币通

过切断经济符号与物理事实之间的联系而创造贫困。价格与实际成本脱节；价值与能源脱离；交易失去与后果的联系。

STC修复了这一断裂。它不是通过道德劝诫或政治力量来实现的，而是通过更优越的信息。通过以能源这一普遍物理单位来衡量价值，STC使经济现实变得可见。这种可见性不是通过重新分配贫困或否认贫困来消除贫困，而是通过消除造成贫困的表现失败来实现的。

一旦我们测量它们实际的功能，口香糖和钢笔就无法再以相同的价格出售。使剥削成为可能的等价幻觉在准确测量的光芒中消失。剩下的是真实——有时严酷，常常具有挑战性，但始终诚实。正是这种诚实，正是经济表现与物理真相之间的联系，产生了真正繁荣的可能性，而不是货币系统所创造的人工稀缺。

STC不是一个未来某天才实施的解决方案。它是一个我们今天可以开始使用的框架，与现有系统并行，边学习边完善。每当我们选择测量真实的能源成本，而不是接受任意价格时，每当我们使隐藏的事物变得可见，每当我们保留上下文而不是摧毁它时，我们就朝着一个以物理现实为基础而非象征扭曲的经济迈进。

贫困不是通过给予人们更多的钱来消除，而是通过消除创造贫困的机制：交易中上下文的丧失。这就是STC所提供的补救措施——不是财富再分配，而是表现的准确性。不是政治革命，而是物理诚实。不是消除稀缺，而是揭示什么是真正稀缺的，什么是人为稀缺的。在这种揭示中蕴含着前进的道路。

8.2 解决对STC的批评

在确定主观热货币如何保持上下文、使隐性成本可见、阻止效率的复合以及消除货币不可能性，从而溶解创造贫困

的机制之后，我们必须面对任何理性人士可能提出的严重反对意见。STC并不是一个简单的提案。它要求将经济价值的基本基础从象征性表示转变为物理测量。这种转变引发了合理的怀疑，而知识诚实要求我们彻底解决这些问题。

对STC的批评分为五个主要类别，每个类别都代表了必须克服的真正挑战，以便STC能够从理论优雅转向实际实施。这些不是为了容易反驳而发明的稻草人反对，而是STC框架面临的最重要和最困难的问题：

测量：我们能否在巨大的多样性的人类活动中以足够的准确性实际测量能量支出？'节省一焦耳等于一个价值单位'的理论清晰度几乎没有意义，如果我们无法在实践中可靠地测量这些焦耳。我们如何量化认知能量、情感劳动、创造性工作和依赖上下文的努力？所需的测量精度水平是什么，当前或可预见的技术能否实现？

主观性与普遍性：STC声称提供客观的物理测量——但由于能力、健康、可用工具和环境条件的差异，能量成本在个体之间变化巨大。一个人花费1,000焦耳的任务可能需要另一个人花费5,000焦耳。这种可变性是否削弱了STC的客观性主张？我们如何在承认个体差异与经济系统对普遍可比性的需求之间取得平衡？

可扩展性和复杂性：即使能量测量在小规模上有效（一个人、一个设备、一个社区），它能否扩展到涉及数十亿人、数万亿交易和无数相互依赖关系的现代全球经济的复杂性？计算要求、验证机制和协调挑战似乎可能是压倒性的。STC的理论优雅是否在实际复杂性的重压下崩溃？

过渡与采纳：假设STC在理论上完美无缺，并且在实践中也能有效——我们如何从这里到达那里？现有的全球金融体系，尽管存在缺陷，却根深蒂固。数万亿美元的资产、数百万份合同、数千个机构和数十亿的生计依赖于当前的货币系

统。如何在不造成灾难性干扰的情况下，从传统货币转向STC？当当前的系统尽管不完美但熟悉且功能正常时，我们如何说服人们和机构采用一个根本新的系统？

外部性和隐性成本：STC声称使隐性成本可见——但它是否捕捉到所有外部性，还是仅仅转移了哪些成本仍然隐藏？能量测量可能考虑了直接的热力学成本，而忽略了环境退化、社会干扰、心理影响和长期系统后果。STC是否可能通过优化可测量的能量效率而忽视不可测量但至关重要的价值和伤害维度，从而无意中创造新的剥削形式？

这五个批评是相互关联的。测量挑战影响可扩展性；主观性问题使过渡复杂；隐性外部性削弱了系统的完整性主张。每个批评必须不仅孤立地解决，而是与其他批评相关联，展示STC如何在不陷入矛盾的情况下应对这些挑战。

本节后续内容不是对反对意见的防御性驳斥，而是对它们的认真回应。对于每个批评，我们将：

承认真正的困难：这些是真实的问题，而不是想象中的障碍。我们在每个子部分开始时以最强的形式陈述批评，认真对待，而不是轻视其重要性。

审视当前解决方案及其局限性：传统经济系统如何处理这些相同的问题？我们常常发现，当前的方法要么完全忽视问题，要么解决得很糟糕。理解现有的失败有助于我们评估STC是否做得更好，或者仅仅是将一组问题换成另一组。

提出STC特定的解决方案：STC提供了哪些机制、技术或框架来应对每一个批评？这些解决方案必须是具体和可行的，而不是对未来创新的空泛呼吁。

识别剩余的挑战和开放性問題：诚实的评估需要承认这些问题仍未解决或解决方案仍不确定。STC不必完美才能优于替代方案——但我们必须清楚其局限性。

与传统货币比较：相关的问题不是STC是否完美解决每一个问题，而是它是否比象征性货币系统更好地处理这些挑战。如果STC的缺陷不如传统货币的缺陷严重，那么即使绝对完美仍然难以实现，这也构成了进步。

本节的目标既不是证明STC无瑕疵，也不是在面对挑战时承认失败。相反，它是为了证明对STC的反对意见虽然重要，但并非不可逾越——而且解决这些问题揭示了对基本STC框架的改进和扩展，使其更加稳健和实用。

此外，认真对待批评具有重要功能：它将STC从一个抽象的理论提案转变为一个具体的、可实施的系统。回答“但X怎么办？”的问题的过程迫使我们具体化机制、测量、机构和程序。这种具体性对于从“如果.....那该多好.....”的推测转向“我们实际上是如何做到的”的实施至关重要。

最后的一个方法论说明：接下来的各小节按从最基本（测量）到最实用（过渡）的顺序排列。这种顺序反映了逻辑依赖结构——我们必须首先确认能量可以被测量，然后再处理主观变异如何处理；我们必须理解基本系统，然后再讨论其扩展；我们必须有一个可工作的扩展系统，然后再规划过渡策略；我们必须有一个运行的系统，然后再评估其外部性。然而，关注特定方面的读者可以独立阅读各小节，因为每个小节都旨在独立存在，同时为整体论点做出贡献。

让我们从最基本的挑战开始：测量问题。我们能否以STC所需的准确性和可靠性实际测量能量支出？

8.2.1 焦耳的测量

主观热货币面临的最基本挑战表述起来 *deceptively simple*，但解决起来极其困难：我们如何在足够的准确性下测量人类活动全谱的能量支出，以作为经济系统的基础？

这不是一个理论问题或未来的担忧——这是决定STC是一个优雅的想法还是一个可行的现实的直接实际障碍。

测量问题有多个维度。我们必须测量不仅是物理能量（肌肉收缩、运动、物体操作），还包括认知能量（注意力、决策、学习、解决问题）和情感能量（压力、挫折、焦虑、恢复）。我们必须在各种极其不同的情境中进行测量——从坐在桌子前到跑马拉松，从解决简单的算术题到创造性创新，从个人孤立到复杂的社会协调。我们必须实现足够的测量精度，以便人们信任系统能够公正地代表他们的贡献和支出。

批判：测量似乎不可能

测量批判的最强形式如下：即使我们承认能量在原则上是经济价值的正确基础，实际测量要么是不可能的，要么是如此侵入性和昂贵，以至于失去了目的。考虑一下所需的条件：

物理能量：为了准确测量物理能量消耗，我们需要在日常生活中持续跟踪肌肉激活、施加的力量、身体运动和新陈代谢过程。当前技术可以通过运动传感器、力量传感器和通过心率和呼吸估算的新陈代谢来近似实现这一点——但准确性有限，设备笨重，测量容易受到个体生理、健康状况和环境因素的干扰。

认知能量：认知过程甚至更具挑战性。像fMRI这样的脑成像技术可以以高空间分辨率测量局部神经活动，但它们需要庞大且不动的设备，仅在受控实验室环境中工作，并且无法扩展到测量日常的认知工作。EEG可以便携，但只能提供整体脑状态的粗略测量，而不能提供特定认知操作的细粒度能量成本。而且这两种技术目前都无法可靠地区分生产性认知工作和无所事事的思维游荡。

情感能量：情感和心理成本可能是最困难的。挫折的能量成本是多少焦耳？慢性压力的能量负担是多少，或者在强烈情感劳动后的恢复成本是多少？这些经历是真实且有影响的——医疗专业人员、教师和社会工作者消耗了巨大的情感能量——但它们抵制简单的物理量化。

情境变异性：即使我们能够独立测量能量类型，相同的物理动作在不同情境下消耗的能量也不同。上坡走路消耗的能量比下坡走路多；疲惫时打字消耗的能量比休息时多；在时间压力下做决定的能量消耗比在放松情况下多。捕捉这种情境变化需要对环境 and 生理状态进行持续监测，这种侵入性大多数人会觉得无法接受。

批判得出结论，STC承诺的“以焦耳测量价值”要么是对未来技术的天真乐观，要么是要求社会上不可接受的监控基础设施，即使在技术上可行。尽管传统货币存在缺陷，但至少具有简单性的优点——价格是什么人都可以观察到的数字，交易是直接的，验证也很简单。STC似乎用象征性抽象换取了测量的不可能性。

传统系统如何处理测量

在解决STC如何解决测量挑战之前，研究传统经济系统如何处理自身测量问题是有启发性的——因为所有经济系统都面临测量挑战，即使它们面临的是不同的挑战。

传统货币测量：法定货币似乎完全绕过了测量——货币只是账户中的数字，价格是买卖双方达成的共识。但这种表面上的简单掩盖了巨大的测量复杂性，这些复杂性已经简单地外包给市场机制。在货币经济中，每个价格都代表了关于相对稀缺性、可取性、生产成本和竞争动态的隐含主张。这些测量通过数百万次交易、价格发现、谈判和市场调整发生——一个复杂程度惊人的连续分布测量系统。

此外，这种基于市场的测量系统在众所周知的方面系统地失败：它无法测量外部性（污染、社会干扰、资源枯竭），它将投机与价值混为一谈（导致泡沫和崩溃），它容易受到操纵（通过垄断、单买、信息不对称），并且对于公共物品和非市场活动完全失效。货币价格的表面简单掩盖了一个同时极其复杂和根本不完整的测量系统。

劳动时间测量：一些经济理论提出劳动小时作为价值的普遍衡量标准。这似乎比货币更具体——只需计算工作小时。但立即出现问题：一小时的非熟练劳动并不等同于一小时的熟练专业知识。一小时的高度专注并不等同于一小时的例行任务完成。一小时的创造性解决问题并不等同于一小时的体力劳动。即使两者花费相同的时间，情感上耗尽的工作也不等同于愉快的工作。

传统的劳动时间核算通过工资差异处理这些差异——熟练劳动的工资更高。但这只是重新引入了测量问题：我们如何确定应用什么倍数？答案通常是“市场能承受的任何东西”，这让我们回到了基于市场的测量的所有问题。

效用测量：经济学有时诉诸于“效用”作为价值的真实衡量标准——人们从商品和服务中获得的满足感或幸福感。但效用是臭名昭著的不可测量。人际效用比较是不可能的（你从吃冰淇淋中获得的满足感无法直接与我从阅读一本书中获得的满足感进行比较）。即使是个人内部的比较也是有问题的（我现在比昨天更快乐吗？多少？）。作为测量框架的效用在理论上是优雅的，但没有提供实用的测量方法。

从传统测量方法的审视中得出的教训有两个方面：首先，所有经济系统都面临着困难的测量挑战——STC并不是唯一受累的。其次，现有的测量系统远比它们看起来的要复杂、不完整和有问题。问题不是STC是否能实现完美测量（没有什么可以做到），而是它是否能实现比其他选择更好的测量。

STC的测量策略：分层近似

STC的测量方法既不是等待完美的测量技术，也不是接受绝望的粗略近似。相反，它采用了一种分层策略，结合了多种测量方法，每种方法适用于能量支出的不同方面，并通过系统的验证和校准来保持准确性：

层级1：直接物理测量：对于物理行为，使用现有技术进行测量相对简单。智能手表和健身追踪器已经能够以合理的准确性（通常为代谢能量的±10%）测量运动、心率和卡路里消耗。这些设备使用加速度计来检测运动，使用光学传感器来测量心率，并使用经过验证的算法根据体重、运动模式和生理信号估算能量消耗。

特定物理行为的能量成本可以在实验室环境中直接测量，然后应用于现实世界的情境：

$E_{\text{keystroke}} \approx 0.1 \text{ J}$	(finger extension/flexion)
$E_{\text{mouse click}} \approx 0.15 \text{ J}$	(hand/finger coordination)
$E_{\text{walking}} \approx 280 \text{ J/minute}$	(average adult, flat surface)
$E_{\text{standing}} \approx 100 \text{ J/minute}$	(postural muscle activation)

这些基线测量可以通过个人校准数据进行调整，以适应个体差异——某人在其典型步伐下行走时的实际能量消耗，经过一段时间的测量，成为他们的个性化校准因子。

第二层：认知代理：直接测量认知能量仍然具有挑战性，但存在经过验证的代理。任务完成时间与认知能量消耗强烈相关——困难的认知工作需要更长时间并消耗更多的大脑能量。认知负荷可以通过瞳孔扩张（随着心理努力而增加）、眨眼频率（在高度集中时减少）和打字模式（在认知压力下变得不规则）来估算。

此外，某些认知操作已通过结合脑成像和热量测定的实验室设置进行测量，从而建立能量基线：

$E_{\text{simple decision}} \approx 2\text{-}5 \text{ J}$	(binary choice, clear options)
$E_{\text{complex decision}} \approx 10\text{-}30 \text{ J}$	(multiple factors, uncertainty)
$E_{\text{focused attention}} \approx 1.5 \text{ J/second}$	(sustained concentration)
$E_{\text{memory retrieval}} \approx 3\text{-}8 \text{ J}$	(accessing specific memory)

这些实验室测量可以通过任务分析缩放到现实世界的背景——将复杂的认知工作分解为基本操作（决策、注意力周期、记忆检索、问题解决步骤），其单独成本可以估算并相加。

第三层：情感能量评估：情感和心理成本是最难直接测量的，但可以通过生理信号（心率变异性指示压力）、行为模式（在情感负荷下决策质量下降）和自我报告（人们可以可靠地指示工作何时情感上令人疲惫，即使他们无法量化焦虑）进行评估。

STC采用分层方法评估情感能量：

人口平均值：通过大规模调查和生理监测建立不同工作类型的基线情感成本。医疗工作者报告X平均情感强度；教师报告Y；客户服务代表报告Z。这些人口平均值提供初步估算。

个体校准：随着时间的推移，个体根据其生理反应、行为模式和明确反馈发展个人档案。某人对特定情况的典型压力反应成为其校准基线。

比较验证：在类似情况下，交叉检查情感能量估算与身体和认知成本。如果某人声称某项任务极其消耗情感，但没有显示相应的生理压力信号，并且在没有抱怨的情况下常规完成类似任务，则情感成本估算可能被夸大。

上下文感知测量：AR和AI的角色

STC测量中的变革性元素是上下文感知系统——主要是结合计算机视觉和人工智能的增强现实智能眼镜。这些系统可以观察你正在做什么，理解上下文，并根据活动识别估算能量成本，而无需侵入性的生理监测。

考虑一个简单的例子：制作咖啡。一个配备计算机视觉的增强现实系统可以识别：

执行的动作：走到厨房（5米→运动能量），伸手拿咖啡机（操作能量），注满水箱（操作+认知规划），测量咖啡粉（精确操作+决策），启动冲泡周期（按按钮），等待（最小能量），取杯子（操作），倒咖啡（精确操作）。

上下文因素：一天中的时间（疲劳时能量成本更高），厨房布局（高效与低效的组织），设备质量（易于操作与难以操作），同时进行的活动（多任务增加认知负担）。

总能量成本可以估算为：

$$E_{\text{coffee}} = E_{\text{locomotion}} + E_{\text{manipulation}} + E_{\text{cognitive}} + E_{\text{context_adjustment}}$$

$$E_{\text{coffee}} \approx 70 + 30 + 15 + 10 = 125 \text{ J}$$

现在假设有人发明了一种咖啡机：（1）是语音激活的，消除了按按钮；（2）有清晰的测量指示，减少了决策能量；（3）位置最佳，便于人体工程学访问。使用改进设备的相同任务：

$$E_{\text{coffee_improved}} \approx 70 + 20 + 5 + 5 = 100 \text{ J}$$

发明者每次制作咖啡节省了25焦耳。如果有一百万人每天使用这种改进的咖啡机，那就是每天节省2500万焦耳（25 MJ）的能量，或每年大约9吉焦耳（9 GJ）。根据

STC，发明者将根据这些经过验证的节省获得相应的经济信用。

关键的见解是，AR/AI系统不需要通过侵入性传感器测量内部生理。它们测量可观察的行为，并使用经过验证的模型推断能量成本。测量是近似的，但系统化、校准和可验证——即使不完美，也足够用于经济核算。

验证与校准

STC测量的关键是持续验证与真实数据的对比和校准，以保持准确性。这通过多种机制实现：

实验室验证：精确专家（在第6章中讨论）进行受控实验室研究，使用黄金标准技术测量特定行为的能量成本：代谢能量的间接热量计、机械工作的力板、认知成本的脑成像。这些实验室测量建立了基线模型，现场系统使用这些模型作为参考。

现场验证：通过比较研究验证真实世界测量的准确性。一部分用户佩戴综合监测设备（研究级传感器），同时使用标准的STC测量系统。比较这两者提供了经验准确性评估，并识别出需要纠正的系统性错误。

交叉验证：不同的测量方法（可穿戴传感器、增强现实观察、自我报告、行为推断）针对相同活动进行比较。方法之间的一致结果增加了信心；差异则触发调查和改进。

人群校准：随着数百万人使用STC系统，出现了人群级别的模式。可以识别异常值——如果某人的报告能量成本在类似活动中始终比人群平均值高出3倍，那么他们要么有异常生理需要个性化校准，要么测量不准确需要调整。

基于纠正的校准：内置于知识钩中的纠正机制提供自然校准。如果一个自动系统声称节省100焦耳，但用户频繁纠

正它（表明它没有按声称的方式工作），那么实际节省低于预测。这种反馈循环根据实际结果不断细化能量估算。

可接受的准确性阈值

一个关键问题：STC的测量必须多准确才能正常运作？完美的准确性既不可实现也不必要。重要的是测量要：（1）系统性无偏（错误不偏向特定方），（2）足够精确以确保大多数交易公平，以及（3）通过校准随着时间的推移可改进。

考虑传统货币测量。当你购买汽油时，泵测量体积——但准确性如何？商业燃油泵要求在 $\pm 0.3\%$ （每10升约30毫升）内准确。大多数电子秤的准确性为 $\pm 0.1\%$ 。现金交易涉及四舍五入到最接近的分。这些测量都不是完美的，但足够好，使商业能够顺利运作，争议很少。

对于STC，可比较的准确性阈值可能是：

物理能量： $\pm 10\%$ 的准确性（可与当前的健身追踪器相比）。对于成本为100焦耳的行为，测量在90-110焦耳之间是可以接受的。

认知能量： $\pm 20\%$ 的准确性（承认测量难度更大）。对于耗费50焦耳的认知工作，测量在40-60焦耳之间是可以接受的。

情感能量： $\pm 30\%$ 的准确性（最难测量）。对于耗费200焦耳的情感劳动，测量在140-260焦耳之间是可以接受的。

这些误差范围可能看起来很大，但它们与传统经济测量中的不确定性相当或更好。考虑尝试测量某人从娱乐中获得的“价值”、制造业带来的污染“成本”或公共教育的“收

益”——传统经济测量包含的更大不确定性是我们已经学会绕过的。

此外，STC 测量会随着时间的推移而改善。随着校准数据的积累，个性化模型变得更加准确。随着测量技术的进步（更好的传感器、更复杂的人工智能、对能量代谢的更深入理解），系统误差减少。系统逐渐变得更好，而不是保持静态。

隐私保护测量

关于普遍的能量测量，隐私是一个合理的担忧。如果 STC 系统必须监控所有活动以测量能量成本，这是否会造成前所未有的监视基础设施？

STC 通过架构选择来解决隐私问题：

本地处理：能量估算发生在设备上（在增强现实眼镜、智能手机、本地计算机中），而不是在集中式云系统中。原始传感器数据永远不会离开设备——只有聚合的能量摘要被记录。

差分隐私：能量测量可以通过控制噪声添加进行报告，既能保护个人隐私，又能在总体规模上保持统计准确性。您可以证明您节省了大约 x 焦耳，而不透露您具体做了什么。

选择性披露：用户可以选择测量报告的粒度——对于精度重要的活动详细报告，对于隐私敏感的上下文聚合报告。您可能会报告工作任务的精确能量成本（补偿取决于此），而仅报告个人活动的粗略每日总计。

加密验证：零知识证明系统允许验证能量测量的准确性，而无需揭示产生这些测量的基础活动。审计员可以确

认“这个人今天节省了1000焦耳”，而无需看到“他们做了咖啡，走到商店，写了一封电子邮件...”

隐私架构类似于端到端加密消息传递的工作方式——基础设施存在以促进交换，但内容保持私密。STC基础设施促进能量核算，而无需中央权威观察个人活动。

剩余挑战和研究方向

尽管有这些解决方案，但仍然存在重大测量挑战：

个体差异：人们在基础代谢、身体能力、认知能力和情感韧性方面存在差异。对一个人来说成本100焦耳的事情，另一个人可能需要150焦耳，这可能是由于年龄、健康、残疾或遗传。应该将这些差异标准化（将每个人视为具有相同能力）还是保留（承认真实的成本差异）？这与下一小节中讨论的主观性与普遍性的批评有关。

长期影响：能量消耗有即时成本（现在燃烧的焦耳）和长期影响（累积疲劳、压力引发的健康问题、减少未来成本的学习）。当前的测量主要捕捉即时成本。准确核算长期影响需要尚未以高信心存在的纵向研究和预测模型。

创造性和智力工作：一些最有价值的人类活动——科学发现、艺术创作、哲学洞察——抵制简单的能量量化。突破性的想法可能在消耗了最少能量的灵感瞬间到来，而在此之前的几天高能量准备看似毫无成效。STC应该如何衡量和补偿真正的创造性工作？

集体活动：当多个人合作时，我们如何归因能量成本？如果三个人一起工作，并比任何个人单独完成得更高效，每个人能获得多少信用？在集体背景下测量个人贡献在传统经济学中也是复杂的，在STC中仍然具有挑战性。

享乐适应：人们会适应改善的环境，而曾经感到费力的事情变得常规。如果自动化将某项任务的认知负担从50焦耳减少到5焦耳，适应后那5焦耳的支出可能会感觉和之前的50焦耳一样费力。测量应该跟踪绝对的物理成本还是主观的努力体验？

这些挑战并不使测量变得不可能，但它们表明需要继续研究、实验和社会共识，以处理边缘案例和模糊情况。

与传统货币测量的比较

回到最初的批评，我们现在可以进行公平的比较。STC测量是否比传统货币测量更困难？

传统货币：看似简单（只是数字），但实际上需要通过市场机制对供给、需求、生产成本、竞争动态、消费者偏好以及无数其他因素进行持续、分散的测量。这些测量是隐性的、未协调的，容易受到操控，并且系统性不完整（缺少外部性、公共产品、非市场价值）。此外，传统货币测量往往是错误的——泡沫和崩溃展示了价格与基础价值失去联系的惊人测量失败。

STC：需要使用传感器、人工智能和校准对能源成本进行明确测量。这在技术上具有挑战性，但随着现代设备（智能手机、可穿戴设备、增强现实眼镜等）的出现，变得可行。这些测量是明确的、可审计的，通过校准可以改进，并且基于物理现实而非集体心理。虽然不完美，但在透明度、准确性和完整性方面，它们系统性地优于传统价格机制。

诚实的评估：STC测量是困难的，但传统测量也很困难——我们只是习惯了它的困难和失败。STC用一组测量挑战交换另一组潜在更好的挑战。这个交易是否值得，取决于STC的明确、基于物理的测量是否比传统货币的隐性、基于市场的测量更可靠。来自可穿戴技术、智能家居系统和增强现实

应用的早期证据表明，这项技术已经成熟，可以开始系统测试。

结论：测量是可行的

对STC的测量批评是严肃的，但并非致命。是的，测量所有人类活动的能源支出是具有挑战性的。但这并非不可能，现代技术（传感器、人工智能、增强现实）使其越来越实用。测量不需要完美——它需要足够好，以至于大多数交易是公平的，系统性的而非任意的，并且随着时间的推移可以改进。

传统货币通过将艰巨的工作外包给系统性未能测量外部性、公共产品和许多关键价值维度的不透明市场机制，创造了测量简单性的错觉。STC使测量变得明确和直接，这看起来更难，但实际上更诚实，最终更准确。

前进的道路不是等待完美的测量技术，而是利用当前能力开始系统测试，建立校准协议，验证真实数据，并随着技术的进步不断改进。这正是传统经济测量的发展方式——经过几个世纪的实验、标准化和改进。STC可以遵循相同的进化路径，但从更坚实的物理基础开始。

如果测量证明是限制因素，STC可以从测量最容易的领域（具有明确能源特征的数字活动）开始，并随着测量的改善扩展到更困难的领域（创造性工作、情感劳动）。在可测量领域的部分实施仍然是进步，即使在所有经济活动中实现全面实施需要更长的时间。

测量挑战是真实存在的，但可以克服。而替代方案——继续使用系统性错误测量价值的象征性货币系统——并不明显优于不完美但可改进的物理测量。

8.2.2 主观性与普遍性

当人们首次接触主观热货币时，常常会产生根本性的误解：他们假设“主观”意味着“相对”或“对每个人都不同”。这种混淆导致了对STC的批评，认为如果测量是主观的，它们必然在不同人之间任意变化，从而削弱了该系统对客观物理基础的主张。这种批评基于对STC中主观性含义的深刻误解。

主观性并不意味着相对主义。它意味着视角。就像视觉是一种主观体验——你从你的眼睛看，我从我的眼睛看——但我们仍然同意这个苹果是红色的，那个叶子是绿色的，STC测量是主观的，因为它们是从每个人的第一人称视角通过他们的个人设备体验和记录的，但能量值本身是客观的、普遍的，并且是物理基础的。只有一个测量，从不同的主观视角体验。

批评：将视角与相对主义混淆

这种误导性的批评通常是这样的：‘如果STC是主观的，那么能量成本必须因每个人的感受或意见而异。但这会使系统变得任意且无法协调。我们如何能有一个经济体，在同一行为的能量成本因执行者而异？如果STC规范化这些差异，难道不放弃了其对物理测量的主张吗？无论如何，主观性和普遍性是不可兼容的。’

这种批评失败了，因为它误解了“主观”在主观热货币中的含义。这个术语并不指相对或基于意见的价值。它指的是从第一人称、上下文意识的视角进行测量。

主观性在STC中的实际含义

在哲学中，“主观”指的是从第一人称视角体验——作为你，体验现实的感觉。这与试图独立于任何特定观察者的视角来描述现实的“客观”第三人称描述是不同的。

关键是，主观并不意味着任意或相对。考虑视觉：看是一种主观体验。你通过你的眼睛、从你的位置、用你的视觉系统看。我通过我的眼睛看。尽管有这种主观性，我们仍然对颜色、形状和物体达成一致。当我们都看着一个红苹果时，我们都看到红色。体验是主观的（我们每个人都有自己的视觉体验），但内容是客观的（苹果的反射光谱是一个独立于我们观察的物理事实）。

STC 正是基于这一原则运作。能量测量是主观的，因为它们是通过每个人的个人设备——增强现实眼镜、智能手机、可穿戴设备、上下文感知传感器——从每个人的增强视角捕获的。这些设备形成了你感官器官的延伸，使你能够增强感知周围的能量流动。你从你的视角测量能量，我从我的视角测量。但我们测量的都是同一个客观的物理现实：在行动和过程中消耗的焦耳能量。

增强感知：设备作为扩展感官

使 STC 同时具备主观性和普遍性的关键创新是通过上下文感知设备实现的增强感知。正如你的眼睛让你主观地视觉接触客观的物理现实（电磁谱），你的增强现实眼镜和连接的传感器让你主观地能量接触客观的热力学现实。

当你执行一个动作——穿越一个房间、打邮件、做饭——你的设备捕获上下文快照：你的位置、运动、生理状态、你互动的物体、环境条件。基于这些快照，系统计算你周围现象的能量消耗：

$$E_{\text{action}} = f(\Sigma_{\text{context}}, \text{physics_models}, \text{calibration_data})$$

这个计算是在本地进行的，从你的主观视角出发，使用你设备的传感器和处理。但函数 f 对每个人都是相同的——它实现了普遍的物理法则（生物力学、热力学、信息理论）。结果是从你的主观视角测量的客观能量值。

与视觉的类比是准确的：你的眼睛使用视网膜传感器从你的位置测量光子。测量是主观的（你特定的视觉体验）。但光子的波长和强度是客观的物理属性。同样，你的设备使用上下文传感器从你的位置测量能量。测量是主观的（你特定的能量体验）。但焦耳值是客观的物理量。

普遍物理学，分布式测量

尽管测量是主观的，但使这个系统具有普遍性的原因在于每个人使用相同的物理模型。肌肉收缩、施加力、行进距离和能量消耗之间的关系由普遍的生物力学方程描述。当你穿越一个房间时，你的设备计算出 280 焦耳的能量消耗，而当我穿越同一个房间时，我的设备也计算出 280 焦耳，我们都通过各自的主观测量设备访问同一个物理现实。

能量计算可能看起来像：

$$E_{\text{walking}} = m \times g \times d \times \eta^{-1} \times f_{\text{terrain}} \times f_{\text{speed}}$$

其中 m 是体重， g 是重力加速度（ 9.8 m/s^2 ）， d 是距离， η 是效率系数， f 项是上下文因素。这个方程是普遍的——相同的物理适用于每个人。但每个人的设备插入他们自己测量的 m 、 d 、地形和速度值，从而得出个性化但客观的结果。

如果两个质量相同的人在相同的地形上以相同的速度走相同的距离，他们测量的能量消耗将是相同的（在测量误差范围内）。主观性存在于测量发生的视角，而不在于测量的值。正如两个从不同角度看同一个红苹果的人都看到红色（尽管他们的视觉体验有所不同），两个执行相同动作的人测量的能量成本也是相同的（尽管他们是从自己的设备进行测量）。

上下文快照和物理计算

主观视角产生客观测量的机制是通过上下文快照——由个人设备捕捉的相关物理变量的全面记录。上下文快照可能包括：

****物理状态****：身体位置、运动速度、肌肉激活模式（来自可穿戴传感器）、心率、呼吸、温度

****环境上下文****：位置、地形类型、环境温度、空气质量、光照条件、障碍物和路径

****物体交互****：正在操作的物体、它们的质量和材料属性、施加的力量、移动的距离

****任务结构****：正在追求的目标、所需的子动作、决策点、认知负荷指标

从这些丰富的上下文数据中，基于物理的模型计算能量消耗。该计算是确定性的——给定相同的快照数据，计算得出的能量值无论由谁的设备执行都是相同的。主观性存在于数据捕捉中（你的设备记录你的上下文），而不在于计算（遵循普遍规律）。

这与传统经济学中的主观价值根本不同，在传统经济学中，“这个苹果对我值2美元，但对你值3美元”反映了对价值的真正分歧。在STC中，“这个动作花费了我280焦耳”并不是关于个人估值的主张，而是通过增强感知获得的物理现实的测量。 \$2 \$3

个体差异：不是主观性，而是物理现实

一个合法的混淆来源：人们确实因身体质量、健康状况、年龄、残疾、可用工具以及无数其他因素的差异，在相同任务上消耗不同的能量。这难道不使能量成本在相对论意义上变得主观吗？

不。这种变化是客观的物理现实，而不是主观的视角。当A（年轻、运动员、70公斤）在走一公里时消耗280焦耳，而B（老年、正在康复、90公斤）在走同样一公里时消耗450焦耳时，这两项测量都是客观且正确的。差异反映了真实的物理差异——不同的质量、不同的机械效率、不同的代谢状态。这些都是可测量、可验证的物理事实，而不是意见或视角。

混淆的产生是因为两个人都从他们的主观视角体验这一行为，并用他们的个人设备进行测量。但他们所测量的是真实的：A确实消耗了280焦耳；B确实消耗了450焦耳。主观性在于测量设备和视角，而不在于数值本身。

为了清晰地看到这种区别，可以考虑一个类比：如果A站在建筑物的一侧，而B站在对面，他们看到不同的外立面。他们的视觉体验是主观的（第一人称视角），但他们所看到的是客观的（建筑物的实际结构）。这座建筑有一个客观的结构，从多个主观视角访问。

同样，一个动作对A有一个客观的能量成本，对B有一个不同的客观能量成本，每个成本都是从他们各自的主观视角测量的。主观性在于测量视角；普遍性在于支配能量消耗的物理法则。

跨主观协议与验证

我们如何知道主观测量会收敛于客观现实，而不是发散到不兼容的个人现实中？通过跨主观协议——使科学成为可能的相同机制。

当多个观察者使用校准仪器测量同一现象时，他们会得到一致的结果（在测量误差范围内）。如果十个人戴着增强现实眼镜观察某人走过一个房间，他们的设备都会计算出大

致相同的能量消耗。这种一致性确认了他们正在测量客观现实，而不是投射主观解释。

当出现差异时（一个设备报告280焦耳，而另一个报告320焦耳用于同一动作），调查揭示了错误来源：校准漂移、传感器故障、缺失的上下文数据、未妥善考虑的环境因素。修正这些错误使测量结果一致。这一过程——注意差异、调查原因、精炼测量——正是如何在主观测量中建立科学客观性的方式。

精确专家的角色（在第六章中讨论）在这里至关重要。这些专家验证测量方法，审核不寻常的声明，并确保主观测量设备经过适当校准，以产生客观结果。他们的功能类似于传统科学中的计量实验室——建立标准并确保分布式主观观察者之间的测量一致性。

与传统货币的比较：真正的主观性

讽刺的是，传统货币在问题的意义上实际上是主观相对的，而STC则不是。在传统货币中，价值是由人们之间真正不同的主观偏好决定的：我可能会认为一张音乐会门票值100美元，而你认为它值30美元，我们都没有客观上的正确。价格是通过这些不同的主观评估之间的谈判产生的，没有客观的真相。 \$100 \$30

这是真正的相对主义——价值因人、情境、情绪、市场营销、社会压力以及与被交换商品的任何物理属性无关的无数其他因素而变化。在传统货币的术语中，没有“这值多少钱”的普遍答案，只有“有人愿意支付多少”，而这个答案是不断变化的。

STC通过将价值建立在物理能量支出上消除了这种相对主义。问题不再是“你认为这个值多少”（真正的主观）而是“这需要多少能量”（客观事实，通过主观测量获得）。每

个人的增强感知可以收敛到相同的能量值，因为这些值反映了物理现实，而不是个人偏好。

传统货币假装是客观的，但实际上是主观相对的。STC看似主观相对，但实际上是客观基础。这种颠倒解决了明显的矛盾。

增强感知的认识论

在更深层的哲学层面上，STC代表了一种认识论创新：热力学现实的增强感知。人类自然感知电磁辐射（视觉）、气压波（听觉）、分子形状（味觉/嗅觉）和机械力（触觉）。我们并不自然感知能量流、熵变化或热力学梯度——这些在科学发展出测量工具之前对人类经验是不可见的。

STC设备将人类感知扩展到热力学领域。就像望远镜将视觉感知扩展到宇宙距离，显微镜扩展到微观尺度，AR眼镜和上下文感知传感器将能量感知扩展到工作、努力和效率的领域。之前不可见的（行为的能量成本）通过增强感官变得直接可感知。

这种扩展的感知是主观的（你通过你的设备从第一人称视角体验它），但揭示了客观现实（物理学描述的实际能量流）。主观性使感知成为可能；物理法则的普遍性确保所感知的是真实的。

这不是科幻小说，而是人类技术增强感知的自然延续。我们已经使用工具来感知无线电波、红外辐射、超声波、磁场以及无数其他对未经辅助的感官不可见的现象。STC将这种增强扩展到热力学感知，使能量成本像颜色一样直接可观察。

实际影响

理解STC的主观性意味着视角而非相对主义具有重要的实际意义：

****无需归一化****：因为测量是客观的（即使是主观捕捉的），所以不需要对测量视角中的个体差异进行归一化或调整。你的设备准确地测量了你的能源成本；我的设备也准确地测量了我的。由于物理差异，当物理现实不同（你为同一任务消耗的能量比我多），数值会不同，但这两项测量都是关于我们各自情况的正确客观事实。

****自动验证****：跨主观一致性提供了自然验证。如果你的设备报告的能源成本与其他设备在可比活动中大相径庭，这表明测量错误或校准问题，而不是合法的主观差异。系统可以自动标记异常，因为我们期望在客观值上趋同。

****普遍交换****：因为能源测量是客观的（尽管是主观捕捉的），它们可以普遍交换。你可以用能源信用支付我，知道你的设备测量的一个焦耳等于我的设备测量的一个焦耳——我们都在从不同的视角测量相同的物理量。这种普遍可交换性使STC成为一种可行的货币。

****去中心化信任****：传统货币需要中央权威来维持价值和防止伪造，正是因为价值是主观决定的。STC可以更去中心化，因为能源测量是基于客观物理的——不需要中央权威来声明一个焦耳的价值；物理学在普遍意义上做到了这一点。

解决剩余的反対意见

****反对****：'但测量设备并不完美。不同的设备可能会计算出相同动作的不同能源值。这难道不重新引入主观性吗？'

****回应****：测量误差并不是主观性。当两个温度计在同一个房间读取略微不同的温度（一个说 21.0°C ，另一个说 21.2°C ）时，这并不意味着温度是主观的——这意味着测量

的精度有限。解决方案是校准，而不是放弃客观测量。能源成本也是如此：校准协议确保设备测量在可接受的误差范围内趋向正确值。

****反对**：**‘情感或认知能量呢？即使物理能量成本相同，困难的主观体验难道不会有所不同吗？’

****回应**：**困难的主观体验确实是可变的，但这与能源成本是分开的。STC测量的是消耗的能量，而不是对该支出的主观感受。两个人可能在认知任务上都消耗了50,000焦耳；一个人觉得愉快，另一个人觉得有压力。STC为两者记录50,000焦耳，因为这就是客观的能源成本。主观体验在其他方面（福利、工作条件）很重要，但并不改变能源核算。

****反对意见**：**‘如果每个人都从自己的角度衡量，谁来决定什么算作合法的能源支出与浪费？’

****回应**：**物理学来决定。如果你在存在高效替代方案的情况下故意使用低效的方法，你的设备仍然准确地测量你所消耗的多余能量——这是客观事实。是否应该补偿这些多余的能量是一个关于社会政策的独立问题，而不是测量的有效性。测量本身是客观的；关于奖励什么的决定是规范性的。

结论：视角促进，而非削弱客观性

一旦我们理解主观性指的是视角，而非相对主义，STC中主观性与普遍性之间的明显紧张关系就会消失。测量是主观捕捉的（通过个人设备从第一人称视角），但产生客观值（由普遍法则决定的物理能量数量）。

这不是矛盾，而是所有经验知识的正常结构。科学通过主观观察（个别科学家使用仪器）进步，最终趋向于客观真理（通过重复验证的普遍法则）。STC将相同的认识论结构

应用于经济测量——分布式的主观观察者使用校准的仪器来测量客观的物理量。

认为STC必须在主观性和普遍性之间做出选择的批评犯了类别错误。主观视角是我们接触普遍物理现实的方式。你的眼睛（主观）向你展示颜色（客观）。你的设备（主观）向你展示能量流动（客观）。主观性存在于接入机制中，而不是接入的内容。

这种主观视角与客观测量的结合远非弱点，而是STC最大的优势。它能够在保持普遍物理基础的同时，增强对热力学现实的分布式感知，使系统保持一致性和公平性。每个人都从自己的角度进行测量，但每个人测量的是同一个客观宇宙——就像我们都看到同一个红苹果，尽管我们每个人都是通过自己的眼睛看到它。

8.2.3 可扩展性与复杂性

8.2.4 过渡与采纳挑战

8.2.5 外部性与隐性成本

9

能量与价值理论

通过能量最小化定义价值

9.1 最小能量法则作为普遍价值

所有系统，无论是生物的还是技术的，都遵循一个基本原则：它们朝着最小化能量消耗的状态演变。在主观热货币中，这一原则成为价值的基础。

价值与能量之间的关系可以通过一个基础方程表达：

$$V = f(E_{\min})$$

其中 V 代表价值， E_{\min} 代表通过主观互动最小化的能量。换句话说：价值不是通过生产更多商品或积累更多财富来创造的，而是通过找到更有效的方法以更少的能量消耗实现相同的结果。

这一原则在每个存在的尺度上都适用。在量子层面，粒子会稳定在基态——最低可能的能量配置。在化学中，反应只有在释放能量或增加整体熵时才会自发进行。在生物学中，进化偏爱具有更高效代谢、更简化身体结构和最小化能量浪费行为的生物。

从这个角度看，人类经济实际上是组织能量流动的系统。传统经济学通过引入货币作为中介——一种可以被操控、囤积并与其所代表的物理现实脱离的符号，掩盖了这一真相。STC 通过直接以宇宙本身使用的货币：焦耳来衡量价值，从而消除了这种扭曲。

当我们自动化一个重复的任务时，我们是在减少完成它所需的能量。当我们设计一个更高效的算法时，我们是在最小化计算能量。当我们创建一个根据上下文预测我们需求的工具时，我们是在消除手动输入的认知和物理能量。这些改

进每一个都创造了真实的、可测量的价值——不是以美元计算，而是以节省的焦耳计算。

最小能量法则不仅仅是描述性的——它是规范性的。它不仅告诉我们系统如何运作，还告诉我们它们应该如何运作以最大化效率和可持续性。任何奖励能量浪费而非能量节约的经济系统与物理现实根本不符，最终将因自身的低效而崩溃。

9.2 熵、效率与主观价值

熵是无序的度量；效率是恢复秩序的度量。主观热力货币通过情境意识定义价值为熵的局部逆转。

在热力学中，熵代表系统从有序状态向无序状态转变的趋势。一杯热咖啡冷却到室温。一个整洁的房间随着时间的推移变得杂乱。能量分散并扩散。这是热力学第二定律：在一个封闭系统中，熵总是增加。

但生命代表了一种显著的现象：这一普遍趋势的局部逆转。生物体通过消耗来自环境的能量来创造和维持高度有序的内部结构。它们并没有违反第二定律——它们在其他地方增加熵——但它们证明了通过高效的能量管理可以创造和维持秩序。

主观技术基于相同的原理。通过捕捉上下文——用户及其环境的当前状态——它可以在无序（低效、浪费的努力、困惑）出现之前预测需求并提供解决方案。每一次成功的预测，每一次在正确时刻触发的自动化，代表了熵的局部减少。

考虑一个学习你早晨例行事务的知识钩：当你醒来时，它会打开咖啡机，将恒温器设置为你喜欢的温度，并在你的显示屏上排队你的日常计划。如果没有这种自动化，你将耗

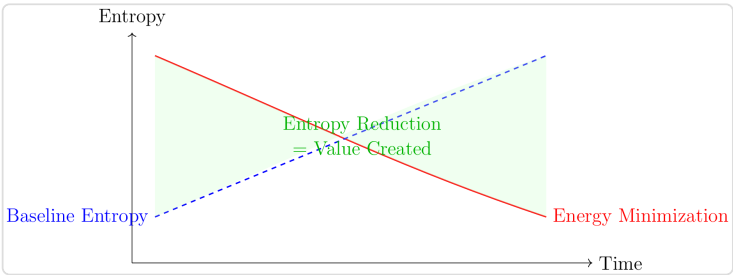
费精力做出每一个决定，手动执行每一个动作。这个钩子通过预测施加秩序，从而减少了你早晨的熵（无序、不确定性、浪费的动作）。

熵减少与节能之间的关系是直接的。当一个系统必须探索多个可能状态以找到正确的状态时，它会消耗能量。当一个系统能够直接进入正确状态，因为它已经学习了上下文时，能量得以节省。在热力学术语中，效率意味着最小化每单位有用工作所产生的熵。

在STC中，主观价值因此被定义为：

$$V_{\text{subjective}} = \Delta S_{\text{reduced}} \times \kappa$$

其中 $\Delta S_{\text{reduced}}$ 是通过情境意识实现的熵减少， κ 是一个校准常数，将熵单位转换为能量等效单位（焦耳）。这个公式明确了一直隐含的内容：无论是人类还是人工智能，智慧的价值在于其从混乱中创造秩序、预测模式和在实现目标时最小化浪费能量的能力。



9.3 能源的道德维度

如果生存依赖于最小化能量，那么道德本身可以被重新定义为能量的有效分配。主观热力货币将伦理从哲学转变为物理学。

在人类历史上，道德体系是建立在抽象原则之上的：权利、义务、美德、后果。这些框架提供了指导，但它们也常常是无休止的争论、文化相对的，并且经常被操纵以服务于权力而非正义。

如果道德可以基于物理法则而不是意见或传统呢？如果问题“我应该做什么？”可以用与“水的沸点是多少？”相同的客观性来回答呢？

STC正是提出了这一点：一种基于物理的道德，源于能量最小化的原则。如果所有生物系统自然寻求最低能量路径，而宇宙本身趋向于热力学平衡，那么最“道德”的行为就是与这一基本倾向最一致的行为。

通过能量的视角考虑以下道德问题：

浪费：破坏可用资源或创造设计为失败的产品增加了社会运作所需的总能量。这在热力学上是不道德的，因为它产生了不必要的熵。

不平等：当一些个体必须耗费大量能量才能生存，而其他人则积累过剩时，系统的运作远离其最佳效率。能量流被阻碍和浪费。更公平的分配允许以更少的总能量支出实现相同的总结果。

暴力：冲突消耗大量的能量——物理的、认知的和社会的——而产生破坏（增加的熵）而非秩序。合作与冲突解决最小化能量支出并最大化集体效率。

教育：教会某人更有效的方法减少了他们在余生中所需的能量支出。知识转移在热力学上是最高价值的活动之一，因为它创造了永久的效率提升。

创造力：减少常见任务能量需求的创新为所有未来用户创造了复合价值。轮子的发明、印刷机、计算机——每一个都

代表了文明能量成本的永久降低。

这并不是说传统的道德直觉是错误的。相反，STC揭示了它们背后的物理基础。我们一直感觉到浪费是错误的，不平等会导致痛苦，暴力是破坏性的，教育是有价值的，创造力应该得到奖励。基于能量的道德解释了这些直觉为何是正确的：它们与自然本身的基本方向一致。

在一个由STC主导的系统中，道德行为会自动获得激励。那些浪费能源的人会产生熵，并且没有任何奖励。那些创新效率的人会创造以节省的焦耳为衡量的价值，并相应地获得补偿。那些囤积资源而其他人却为了生存而过度消耗能源的人，会发现他们的资源自然通过系统固有的趋向平衡而重新分配。

这就是生活在“上帝的统治”下的意义——不是由神圣法令或人类对经文的解释来统治，而是与支配所有存在的物理法则保持一致。当人类系统与热力学原理和谐运作时，正义不是一种需要强加的抽象理想，而是效率的自然结果。

能量的道德维度揭示了一个深刻的真理：宇宙有一个方向，而这个方向是朝向更高的效率、更低的浪费和更公平的资源分配。STC只是明确了这个方向，并建立了一个与之相辅相成的经济系统，而不是与之对抗。

10

主观热力货币的核心框架

将焦耳定义为价值的真实衡量标准

10.1 STC 单位的正式定义

一个主观热力货币单位（STC）代表通过主观优化所节省的能量的标准化测量。

$$1\ STC = 1\ J_{saved} \times \kappa_c$$

其中 κ_a 是一个校准系数，用于在不同背景下规范人类、机械和算法的能量节省。

10.2 热力学账本

每一个主观行为——每一个上下文感知的自动化、避免的纠正或移除的冗余过程——都作为热力学交易进行登记。账本存储能量增量，而不是财务债务。



10.3 隐私和上下文层

主观上下文设计上保持私密。只有聚合的能量增量，而非个人数据，才会与账本共享。该架构保证了上下文主权。

11

能量最小化经济

从稀缺到后劳动价值体系

11.1 从能源浪费到能源市场

在STC中，市场不再交易商品——它们交易的是能源支出的减少。一个过程消耗的能量越少，它的价值就越高。

$$V_{process} = \frac{1}{E_{consumed}}$$

11.2 效率的平衡

正如价格平衡供需一样，STC平衡能量流动。该系统自然奖励集体效率而非个人积累。

11.3 后劳动经济学

当认知努力和体力劳动被主观自动化取代时，传统劳动失去了经济意义。价值从生产转向优化。

12

社会和伦理影响

能源社会中的人权

12.1 能源平等

正如早期社会为获取水和食物而斗争，未来社会将为获取能源效率而斗争。STC将平等重新定义为能源最小化的普遍权利。

12.2 后稀缺正义

正义不是通过财富的再分配而产生，而是通过效率的再分配。当每个过程都趋向于最小能耗时，不平等自然会崩溃。

12.3 人工主观性的伦理

随着设备通过知识钩获得自我意识，伦理考量扩展到非人类系统。责任在于共同最小化能源的主观代理之间共享。

13

实施和技术层

主观系统的架构

13.1 硬件基础

智能眼镜、传感器阵列和嵌入式设备提供持续的环境快照。这些输入为知识钩子提供数据，测量现实世界的能量变化。



智能眼镜捕捉上下文输入以供知识钩子的概念图

智能眼镜捕捉上下文输入以供知识钩子的概念图

13.2 软件和内核设计

在软件层面，主观内核管理本地上下文，同步钩子，并维护变化历史。它们优先离线操作，确保隐私和自主性。

13.3 上下文压缩算法

上下文变化压缩（CDC）仅编码感官或认知状态的变化。这大大减少了计算负担，同时保持语义准确性。

$$C_{new} = C_t - C_{t-1}$$

14

数学热经济学

量化认知和物理能量

14.1 价值的热力学方程

$$STC = \int_{t_0}^{t_1} (E_{expected} - E_{actual}) dt$$

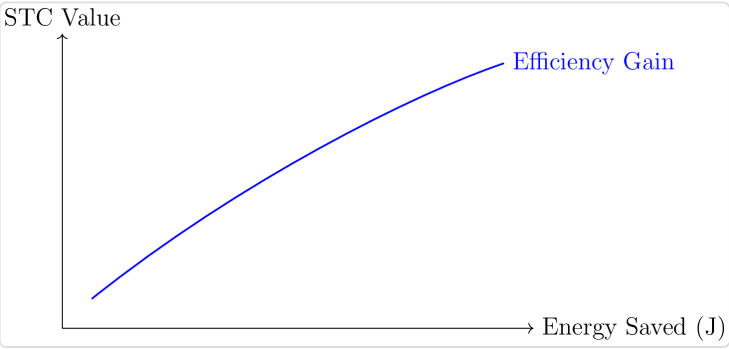
该积分定义了在给定时窗口内节省的总主观能量。每次预期能量的减少都代表了价值的增加。

14.2 认知系统中的熵减少

当预测可靠性提高时，认知熵减少。因此，学习是在神经层面的热力学优化。

$$\Delta S_{cog} = -k_B \ln(P_{pred})$$

14.3 能量-价值转换图



15

主观网络治理

通过上下文相似性达成共识

15.1 分布式上下文一致性

在主观网络中，当多个设备共享近乎相同的上下文状态时，达成共识。无需投票——同步本身就是一致。

15.2 防止欺诈与熵膨胀

当每个设备在本地验证熵增量时，能量欺诈是不可能的。只有当能量凭空出现时，膨胀才会发生——这在物理上是不可能的。

16

案例研究与应用

STC在现实世界中的应用

16.1 智能家居与能源共生

配备主观设备的家居不断减少浪费的动作、能源和时间。家电学习协调与人类存在相和谐的能源使用。

16.2 教育作为能源效率

学习成为一种优化过程——学生因减少未来认知负担的知识而获得奖励，而不是记忆。

16.3 医疗保健与认知平衡

主观传感器监测生物熵——压力、疲劳和恢复周期——以维持身心之间的能量平衡。

17

未来愿景与哲学

超越金钱的人类进化

17.1 后稀缺文明

在一个由能源效率而非稀缺性治理的社会中，知识的每一次提升都为所有人增加了丰富性。

17.2 能源的上帝方程

最终的综合：生命、智慧和道德由一条法则统一——能源的最小化。这条法则不仅定义了经济学，还定义了意识本身的命运——信息处理系统的演变，这些系统不断积累和整合感官数据，效率不断提高。

$$\frac{dE}{dt} \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty$$

17.3 没有货币的世界

当每个过程都变得自我优化时，货币便变得过时。能源不再被交易——它只是通过生命的主观网络流动。

For centuries, visionaries have imagined a world where energy itself could replace money. Many theorized that tying value to energy would be more fair, efficient, and aligned with nature's laws. But these remained only theories—until now.

With the release of this book, we present the first clear, practical framework for implementing energy-as-currency through Subjective Thermo-Currency (STC).

This is not science fiction. STC is being built today, using a powerful combination of:

- SmartGlasses AR/MR
- Subjective Artificial Intelligence
- Negative Reinforcement Learning techniques

At its core, STC moves value away from goods and services themselves, and toward the energy-efficient processes that generate them. Scarcity is no longer managed by speculation or price; it is resolved by innovation, context-awareness, and efficiency.

This book covers:

- The evolution of value systems—from bartering, to commodities, to money, to crypto, and now STC.
- The mathematics of Subjective Technologies, Knowledge Hooks, and energy minimization.
- Implementations that show how STC can run directly on SmartGlasses, providing the best possible user interface design for AGI.
- A look at your day-to-day life without money—how property, work, collaboration, and creativity change when value is measured by energy instead of tokens.
- The future of humanity under STC: a post-scarcity world with new augmented bodies, no suffering, and even the possibility of preserving consciousness indefinitely.

Subjective Thermo-Currency will make you think outside the box. It offers not just a critique of money, but a blueprint for a new economy

fully aligned with physics and nature—an economy where comfort, abundance, and true freedom become possible.



Tommy Fox



Subjective Thermo-Currency: Harnessing Subjective AI and Smart-glasses to Replace Money © 2025 Tommy Fox.
All rights reserved.
Published by Subjective Technologies.
www.subjectivetechnologies.com
