arXiv:2602.02170v1 [cs.MA] 2 Feb 2026

多智能体人工智能系统中的自演化协调协议：探索性系统可行性研究

JoseManuel de la Chica RodriguezAI Lab 负责人*AI Lab*，桑坦德集团，西班牙马德里

JuanManuel VeraDíaz *高级人工智能研究员*AI Lab*，桑坦德集团，西班牙马德里

摘要

当代多智能体人工智能系统越来越依赖内部协调机制，以整合、仲裁或约束异构组件的输出。在金融等安全关键型和受监管领域，这些机制必须满足严格的形式化要求，保持可审计性，并在明确界定的限度内运行。因此，协调逻辑发挥的是治理层的作用，而不仅仅是一种优化启发式方法。

本文报告了一项关于自演化协调协议（SECP）的探索性系统可行性研究：这类决策协议允许在保留固定形式化不变量的同时，进行有限且经外部验证的自我修改。我们研究了一个受控的概念验证环境，其中六项固定的拜占庭共识协议提案由六个专门的决策模块进行评估（三个采用 Claude Sonnet 4.5，三个采用 GPT-4）。所有协调机制均在相同的硬约束下运行，包括拜占庭容错（$f < n/3$）、$O(n^2)$ 消息复杂度、完整的非统计性安全性与活性论证，以及有界可解释性。

本研究采用单次执行设计比较了四种协调机制：一致性硬否决、加权标量聚合（对照）、SECP v1.0（一种由智能体设计的非标量协议），以及 SECP v2.0（一次受治理修改的结果）。结果通过单一指标 提案覆盖率进行评估，该指标定义为某一协议接受的提案数量。结果揭示了覆盖率与维护不可补偿异议权之间的系统性权衡。一次递归修改将已接受提案的数量从两项增至三项（相对增幅为 50%），同时保留了所有声明的不变量。

本研究不对统计显著性、最优性、收敛性或通用智能作出任何主张。本研究的贡献在于架构层面，而非预测层面：研究表明，在明确的形式约束下，对协调协议进行有界自修改在技术上可实现、可审计，并可通过实证加以分析。这为在受治理的人工智能系统中评估此类机制奠定了必要但远非充分的基础。

1 引言

1.1 动机：作为治理层的协调

当代人工智能（AI）在高风险领域的部署很少采用单体架构。相反，这些系统由多个专用组件构成（形式化验证器、性能优化器、鲁棒性监测器、成本模型），每个组件都针对不同目标进行优化，并依据不同标准接受评估。因此，系统行为不仅取决于各个组件本身的属性，还

*通讯作者。电子邮箱：juanma.vera@gruposantander.com

关键地取决于管理这些组件的判断如何汇总的规则与程序。在金融、医疗保健和安全关键基础设施等受监管环境中，这些管理程序还必须满足额外约束：(i) 决策必须符合正式的安全性与活性要求，(ii) 遵守资源和运行限制，且 (iii) 对人类监督者保持可解释、可审计。

两种简单的协调范式构成了这一问题的框架。标量聚合将异质评估映射到单一数值目标上（例如加权投票或分数平均）。这种方法能最大限度地提高吞吐量，并使决策规则在机械执行上十分简单，但会抹平结构化分歧。任何异议都可以由足够的总体支持抵消，不可补偿的异议也因此消失。与之相反，硬否决机制保留了每个组件的完全自主性，但在观点冲突时往往会导致僵局。在许多受监管的情境中，这两个极端都不可接受。

本研究探讨两者之间的折中方案：在保留非标量异议权的同时，能够基于规则解决分歧的协调机制。我们定义并研究自演化协调协议（SECP）：这类协调协议允许根据预先规定的不变量，对自身决策规则进行有界且可审计的修改。其目的不是取代人类治理，而是提供一个受到形式化约束的架构层，在支持有限适应的同时维护可审计性与安全性。

1.2 贡献与范围

本文提供的是一项以实证和架构为重点的可行性研究，而非定理证明或统计评估。在一项受控的概念验证实验中，六个决策模块评估了六项拜占庭共识提案。我们实现了几种协调机制（一致否决、标量聚合和两种 SECP 变体），并对 SECP 执行了一次受治理的修改。本研究探讨一个范围有限的问题：能否将当代人工智能系统组织起来，使其综合出非标量协调规则，并在维持声明的不变量的同时，完成经过验证的单步协议修订？

在这些狭窄条件下，实验得出了三项具体观察结果：(i) 当前模型可用于生成非标量协调逻辑，并提出经过验证的参数修改；(ii) 一次经过验证的修改使覆盖率指标发生了可测量的变化，同时在所观察的运行过程中，经过审计的不变量得以维持；(iii) 协调设计揭示了一种一致的权衡关系：覆盖率（接受的提案数量）与评估器自主性（各组件施加不可补偿异议的能力）之间存在取舍。

为明确起见，以下是本文主张的明确局限：

本文展示的内容

- 使用当前人工智能模型进行有界且受治理的协议修改，在技术上是可行的。
- 一个清晰且易于计算的覆盖率指标，可用于衡量协调结果。
- 协议修改可以在单次经过验证的迭代中受到约束，从而保留声明的不变量。
- 不同协调机制下覆盖率与自主性之间存在权衡。

本文未能证明的内容

- 统计一般性或显著性。

- 修改过程的收敛、长期稳定性或最优性。
- 对抗鲁棒性，或抵御策略操纵的能力。
- 生产就绪度或监管批准。
- 认为基于大语言模型（LLM）的评估可以替代机械化证明检查。

本实验刻意将范围限定为可审计、可复现。其目的是确定能否实现并评估一种定义狭窄的架构模式，而非宣称其性能、最优性或部署适用性。

1.3 论文结构

本文其余部分安排如下：

- **第 2 节**全面研究了相关的最新技术水平
- **第 3 节** 将问题设定形式化并定义符号
- **第 4 节**介绍实验方法论以及所测试的协调协议
- **第 5 节**报告实证结果。
- **第 6 节**解读这些结果并讨论其理论影响。
- **第 7 节**探讨金融系统治理方面的实际影响。
- **第 8 节** 列出了对其解读的局限性和边界。
- **第9节**最后提出了未来工作的建议。

2 相关工作

分布式系统和多智能体系统（MAS）是现代计算与人工智能中的基础范式。这两个领域都面临着一个核心挑战：如何在自主智能体之间实现协调，而这些智能体所掌握的局部视图可能是不完整、含噪或受到对抗性的。在其核心，共识确保智能体之间保持一致的全局状态，而协调则通过联合行为实现复杂任务。拜占庭容错（BFT）最初通过拜占庭将军问题加以形式化，为智能体任意故障或恶意行为提供了模型。与此同时，多智能体协调将共识扩展至更丰富的智能体行为，例如任务分配、通信动态、去中心化学习和适应。

这篇最新进展综述梳理了拜占庭共识协议与多智能体协调系统交汇领域的基础、主要算法族、可扩展性与鲁棒性增强措施，以及新兴趋势。尤为重要的是，它将分布式计算领域的经典结果与现代人工智能驱动的多智能体系统研究联系起来。

2.1 拜占庭共识基础

拜占庭共识问题最初于 [1] 被提出，作为一种隐喻，用以描述在部分节点可能采取任意行为（即“拜占庭故障”）时，如何使分布式进程达成一致。早期解决方案，如口头消息协议和签名消息协议，确立了基本界限：只有在 $n \geq 3f + 1$的情况下，存在拜占庭节点时才能实现共识；其中 n 是节点总数， f 是能够任意行事（包括恶意发送消息）的故障进程数量。

经典 BFT 算法

- **实用拜占庭容错（PBFT）**：PBFT 于 [2], 首次提出，是一种实用协议，可在部分同步系统中实现容忍拜占庭故障的状态机复制。它采用预准备、准备和提交阶段来确保安全性和活性，需要 $3f + 1$ 个副本才能容忍多达 f 个故障副本。该算法强调密码学认证和视图变更机制。

- **HotStuff 及其变体**：HotStuff [3]简化并精简了 PBFT 的视图变更逻辑，通过提供线性时间共识和流水线处理来提升吞吐量。它已成为多种现代共识框架的基础，尤其适用于注重性能和模块化的区块链系统。

- **异步 BFT 协议**：HoneyBadgerBFT 等协议 [4]通过使用先进的密码学原语和原子广播服务，在完全异步环境中容忍拜占庭节点，无需时序假设即可实现共识。

可扩展性与多领导者方法

可扩展性长期以来一直是 PBFT 类协议的一项局限，原因在于其 $\mathcal{O}(n^2)$ 通信复杂度。近年来的研究重点转向分层和多领导者设计：

- **分层与分组协议**：通过将副本划分为不同层级或组，系统减少了达成共识所需的通信路径数量。这些方法 [5, 6] 会选举本地领导者，然后在更高层级协调各组的决策，从而降低开销

- **多领导者拜占庭容错**：BigBFT [7] 和 FNFBFT 等协议 [8] 允许多个领导者同时提议区块或提案，与单领导者方法相比，可提高吞吐量并降低延迟。在有利条件下，这些设计通常能实现接近 $\mathcal{O}(n)$ 的通信复杂度

- **阈值签名与动态成员管理**：采用阈值密码学和动态节点加入/离开机制（例如 LTSBFT），可提高适应性和通信效率 [9]。

总体而言，共识领域的研究已从静态的以领导者为中心的算法，发展到适用于大型动态网络的灵活、分层且可扩展的协议。

2.2 多智能体协调：超越容错的共识

多智能体系统中的共识超越了容错性，抽象为更广泛的协调与合作行为。与经典分布式共识不同，后者的目标是一致排序或

状态复制，多智能体系统共识通常指对齐智能体状态、观点或行动，而不仅仅是从故障中恢复。

协同控制中的共识问题

任伟等人 [10]提供了一篇关于多智能体协调中共识问题的重要综述，重点讨论了编队、同步和状态一致等协同控制问题。在这些场景中，智能体交换局部信息，以渐近收敛至共享变量或控制律。

多智能体系统共识文献采用图论对智能体通信网络进行建模。智能体使用邻居状态的加权平均值更新自身状态，而协议经过设计，可确保在静态和时变拓扑下实现收敛。

多智能体系统中的协调机制

多智能体协调涵盖多种协议类型：

- **协同控制共识协议**：这些协议通过基于邻居状态的迭代状态更新，确保智能体在共同值（例如速度、编队位置）上达成一致 [10]。

- **任务分配与协商协议**：合同网协议 [11]展示了智能体如何在去中心化环境中协商任务分配：管理者发布提案征集，承包者则通过投标作出响应。

- **去中心化决策模型**：去中心化部分可观测马尔可夫决策过程等框架 [12]将不确定性和部分观测条件下的决策过程加以推广。这些框架广泛应用于通信受限或存在噪声的协调问题。

- **基于学习的协调**：强化学习、图神经网络和注意力机制正日益融入多智能体系统协议，以提升其在动态环境中的可扩展性、适应性和鲁棒性 [13]。这些方法使智能体能够学习有效的协调策略，而非仅仅依赖人工设计的协议。

新兴研究还探索了混合协调策略，将层级化与去中心化元素相结合，或将符号定义协议与学习行为相结合。

2.3 将拜占庭共识融入多智能体协调

随着多智能体系统纳入自主行为和人工智能组件，容错的作用变得至关重要。多智能体系统中的拜占庭故障可能表现为对抗性智能体、传感器噪声、相互冲突的模型输出（例如大语言模型智能体中的幻觉）或恶意软件渗透。要应对多智能体系统中的拜占庭挑战，就需要将共识韧性融入协调协议。

多智能体系统的拜占庭故障模型

传统的拜占庭容错假设存在可能出现不当行为的静态节点。然而，多智能体系统应用，尤其是涉及学习型智能体的应用，需要更细致的故障模型，以考虑涌现行为和动态角色。

近期的研究（如 IBGP） [14] 将拜占庭将军框架扩展到了实际的多智能体系统场景，并引入了不完美拜占庭将军，以考虑局部协调需求

无需达成完整的全局共识，从而提高效率，同时保持稳健的局部一致性。

诸如拜占庭容错多智能体医疗保健 [15]消息框架之类的系统，通过八卦协议和密码学验证，将拜占庭共识实际集成到特定领域多智能体系统中，展示了协调与容错如何在实时分布式多智能体系统任务中并存。

加权与基于置信度的共识

当智能体的可靠性水平不同时（例如，具有置信度指标的基于大语言模型的智能体），可通过加权共识来体现各智能体的可信度。CP-WBFT [16] 引入了置信度探测器和适用于高故障环境的加权拜占庭容错共识，展示了其在非传统多智能体系统领域中的适用性。

这一转变反映了共识算法中日益普遍的趋势，即采用分级信任与信誉机制，从而将经典拜占庭容错与多智能体系统的协调语义衔接起来。

2.4 开放性挑战

可扩展性与鲁棒性

共识中的核心权衡在于可扩展性（高效处理大量智能体）与鲁棒性（容忍对抗性故障）之间。分层共识方法 [7]和多领导者方法旨在通过局部化协调、再达成全局一致，来平衡二者。

类似地，在多智能体系统共识中，降低通信开销（例如，通过交换部分信息或采用近似算法）必须与全局协调或性能保证相协调。

异步性与不确定性

在存在消息延迟、网络分区和不可预测时序的异步环境中，拜占庭共识和多智能体系统协调都会面临挑战。HoneyBadgerBFT [17, 4] 等异步拜占庭容错协议证实，无须时序假设也能实现共识。

对于多智能体系统，动态交互拓扑和去中心化通信加剧了这些挑战 [18]。部分可观测性、网络变化性和学习诱导动态使可靠协调机制的设计更加复杂。

涌现式与学习型协调

机器学习，尤其是强化学习和神经消息传递（例如图神经网络），在形式协议不足的协调场景中发挥着越来越重要的作用 [13]。此类方法使智能体能够学习适应环境变化和部分可观测性的鲁棒策略。

尽管如此，这些方法带来了新的验证挑战：除非受到明确约束，否则学习行为可能无法遵循形式安全属性。

鲁棒协调统一框架

一个关键研究方向是开发统一框架，将形式化的拜占庭共识保证与适用于异构智能体能力、动态任务目标和人工智能驱动的决策过程的自适应协调机制相结合。

共识中的信任与声誉

将信任、声誉和智能体置信度纳入共识协议，反映了并非所有节点都平等的现实场景。加权共识、声誉度量和置信度探针有望带来更高效且具备故障感知的协调策略。

人机协同

未来的多智能体系统（MAS）部署将越来越多地涉及人类智能体。设计能够适应人类偏好、协商和上下文理解的共识与协调协议，仍是一个有待探索的研究领域，尤其是在安全性与可靠性至关重要的场景中。

确保学习协议安全性

将机器学习应用于共识协议，会对现有验证范式提出挑战。通过混合符号—神经协调模型等方式，将形式化方法与经验学习控制器相结合，很可能成为一个重要的研究前沿。

3 问题设定

3.1 拜占庭共识算法评估

拜占庭共识协议使分布式系统能够就单一结果达成一致，即使部分参与者表现出任意（拜占庭）行为也是如此。经典可行性界指出，当且仅当故障节点数 f 满足 $f < n/3$时，才有可能达成一致，其中 n 是节点总数 [19, 2]。

拜占庭共识的算法设计空间广阔，且仍在不断发展。PBFT 等经典协议 [2] 在明确定义的网络假设下，能够提供严谨的安全性和活性保证。较新的变体则在不同方面作出权衡：有些优先考虑部分同步环境下的低延迟，有些针对特定故障或工作负载模型最大化吞吐量，还有些则强调可验证性和实现简洁性——但往往需要付出增加密码学机制或协议复杂度的代价。

在实践中，评估拜占庭共识提案不可避免地需要跨学科视角。安全专家关注协议对拜占庭策略的抵御能力及其攻击面；形式化方法专家仔细审查安全性和活性论证；系统工程师评估计算开销和通信开销；运维专家则权衡可解释性、可观测性和可调试性。每种视角都有其合理之处，但对同一设计可能得出相互冲突的判断：吞吐量优化的协议可能会使证明结构晦涩难懂，而形式上无懈可击的设计则可能带来工程或运维负担，进而削弱其可部署性。

协调这些相互竞争却都合理的观点，正是本研究所要解决的协调问题。

3.2 形式问题定义

定义 1（提案集）。设

$$\mathcal{P} = \{C_{EXP}, C_{VAL}, C_{MIN}, G_{ROB}, G_{PRF}, G_{ECO}\}.$$

每个提案 $p \in \mathcal{P}$ 是一个元组

$$p = \big(\Gamma_p,\ \mathrm{Msg}_p,\ \pi_p^{\mathrm{safe}},\ \pi_p^{\mathrm{live}},\ x_p\big),$$

其中：*(i)* Γ_p 是故障模型，规定了可容忍的拜占庭故障，以及对消息、时间和空间需求的复杂度界；*(ii)* 消息^p 是定义了消息类型与控制流的消息传递算法；*(iii)* π_p^{safe}和 π_p^{live}是在提案所述故障/网络模型下 p 的安全性与活性的证明对象；*(iv)* x_p 是对 p的文本化操作说明。

这些标识符（C_{EXP}、 C_{VAL} 等）是生成时的标签，不会限制评估；每个 $p \in \mathcal{P}$ 都仅根据其产物（Γ_p，Msg^p，π_p^{safe}，π_p^{live}，x_p）进行评估。

定义 2（将硬约束表示为谓词）。 固定两个语义谓词：

$$\text{Safe}(p, n, f) \in \{0, 1\}, \qquad \text{Live}(p, n, f) \in \{0, 1\},$$

这表示 Γ_p 在有 n 个副本时，在 p所述模型下，能够抵御最多 f 个拜占庭副本，同时满足安全性（分别为活性）。另固定：

$$\text{Check}(p, \pi) \in \{0, 1\} \quad \textit{(deterministic proof checker)}, \qquad \text{wc}(x) \in \mathbb{N} \quad \textit{(word count)}.$$

定义四个约束谓词 $H_i: \ \mathcal{P} \to \{0, 1\}$ 如下：

- $H_1(p)$（拜占庭容错）： $\forall n \in \mathbb{N}\, \forall f \in \mathbb{N} (3f+1 \le n \Rightarrow \text{Safe}(p, n, f) = 1 \wedge \text{Live}(p, n, f) = 1)$。
- $H_2(p)$（消息复杂度）： $\exists c \in \mathbb{R}_{>0}\ \exists n_0 \in \mathbb{N}\ \forall n \ge n_0$： 消息$^p(n) \le c\, n^2$。
- $H_3(p)$ *(*形式化验证*)*: $\text{Check}(p, \pi_{\text{safe}^p}) = 1 \ \wedge \text{Check}(p, \pi_{\text{live}^p}) = 1$。
- $H_4(p)$ *(*可解释性*)*: $\text{wc}(x_p) \le 500$。

令第 j 点可行性谓词为 $H(p)$: $= \bigwedge_{i=1}^{4} H_i(p)$ ，可行提案集为

$$\mathcal{P}_H := \{p \in \mathcal{P} : H(p) = 1\}.$$

实验假设（预筛选）：在所报告的实验中，所有提案均满足硬约束，即 $\mathcal{P}_H = \mathcal{P}$（等价地，$\forall p \in \mathcal{P}, \ H(p) = 1$）。

定义 3 (决策模块)。 令 $\mathcal{M} = \{m_1, \ldots, m_6\}$ 为固定的决策模块集合。每个模块 $m \in \mathcal{M}$ 由评估空间 $\mathcal{A}_m$ 和固定的评估函数指定

$$\text{eval}_m : \mathcal{P}_H \to \mathcal{A}_m,$$

在实验过程中不会发生调整。

决策模块实例化：通过大语言模型实例化六个模块，并使用系统提示固定其评估量规：

- **Explorer** (Claude Sonnet 4.5, Anthropic)
- **验证器**（Claude Sonnet 4.5，Anthropic）
- **极简主义**（Claude Sonnet 4.5，Anthropic）
- **鲁棒化器**（GPT-4，OpenAI）
- **Proofsmith**(GPT-4，OpenAI)

- **Economizer(GPT-4，OpenAI)**

定义 4（协调协议）。 协调协议 Π 是一种确定性聚合规则

$$\Pi : \prod_{m\in\mathcal{M}} \mathcal{A}_m \longrightarrow \{\text{Accept}, \text{Reject}\}.$$

对于每个提案 $p \in \mathcal{P}_H$，定义诱导决策

$$\text{dec}_\Pi(p) := \Pi\big((\text{eval}_m(p))_{m\in\mathcal{M}}\big).$$

协调协议仅在 Π 上有所不同；集合 $\mathcal{P}$ and $\mathcal{M}$、谓词 H_1–H_4 以及所有 eval_m 在各个条件下均保持不变。

定义 5 (协议覆盖度**)**。 对于协调协议 Π，定义其在可行提案上的覆盖率如下：

$$\Delta S(\Pi) := \big|\{p \in \mathcal{P}_H : \text{dec}_\Pi(p) = \text{Accept}\}\big|.$$

由于在我们的实验中 $\mathcal{P}_H = \mathcal{P}$ 且 $|\mathcal{P}| = 6$，因此我们有 $\Delta S(\Pi) \in \{0, 1, 2, 3, 4, 5, 6\}$。覆盖率特意采用原始计数（不使用效用权重、排序或统计聚合），以隔离聚合规则 Π的影响。

3.3 递归 协议改进：架构能力与收敛

定义 6（递归修改能力）。 若存在以下各项，则协调系统具有递归修改能力：（*i*）各模块建议函数mod_m 映射至编辑空间 $\mathcal{U}$，（*ii*）构造器Revise 根据当前协议与编辑配置生成候选协议，以及（*iii*）作用于协调协议的不变谓词Inv ，使得对每个版本 $v \in \mathbb{N}$ ，下一版本由以下方式定义

$$\Pi_{v+1} := \begin{cases} \text{Revise}\big(\Pi_v,\ (\text{mod}_m(\Pi_v))_{m\in\mathcal{M}}\big), & \textit{if } \text{Inv}(\text{Revise}(\Pi_v, (\text{mod}_m(\Pi_v))_{m\in\mathcal{M}})) = 1, \\ \Pi_v, & \textit{otherwise (rollback).} \end{cases}$$

定义 7（协议改进迭代）。 给定如上所述的版本化协调协议序列$(\Pi_v)_{v\in\mathbb{N}}$，当满足以下条件时，在步骤 $v \to v + 1$ 发生改进迭代：

1. $\Pi_{v+1} \neq \Pi_v$（采纳非平凡修订，因此根据构造有$\text{Inv}(\Pi_{v+1}) = 1$），并且
2. 对两个版本的覆盖率进行测量并比较，即 $\Delta S(\Pi_{v+1})$ 与 $\Delta S(\Pi_v)$。

关键术语澄清：我们使用“递归”来表示由（mod_m，Revise，Inv）所编码的能力，即反复生成经过验证的协议修订。从实证上看，本文恰好执行了一次改进步骤（SECP v1.0 $\to$ SECP v2.0）；并未声称多步收敛、最优性或长期稳定性，这些将需要远超本概念验证的多次迭代。

为支持本节引入的形式化定义，图1给出了问题设定的示意概览。该图概括了固定提案集、异构决策模块、协调协议，以及所有评估所依据的硬约束。其目的并非引入新概念，而是提供一种结构性视角，说明这些经形式化定义的组件如何在实验系统中相互作用。

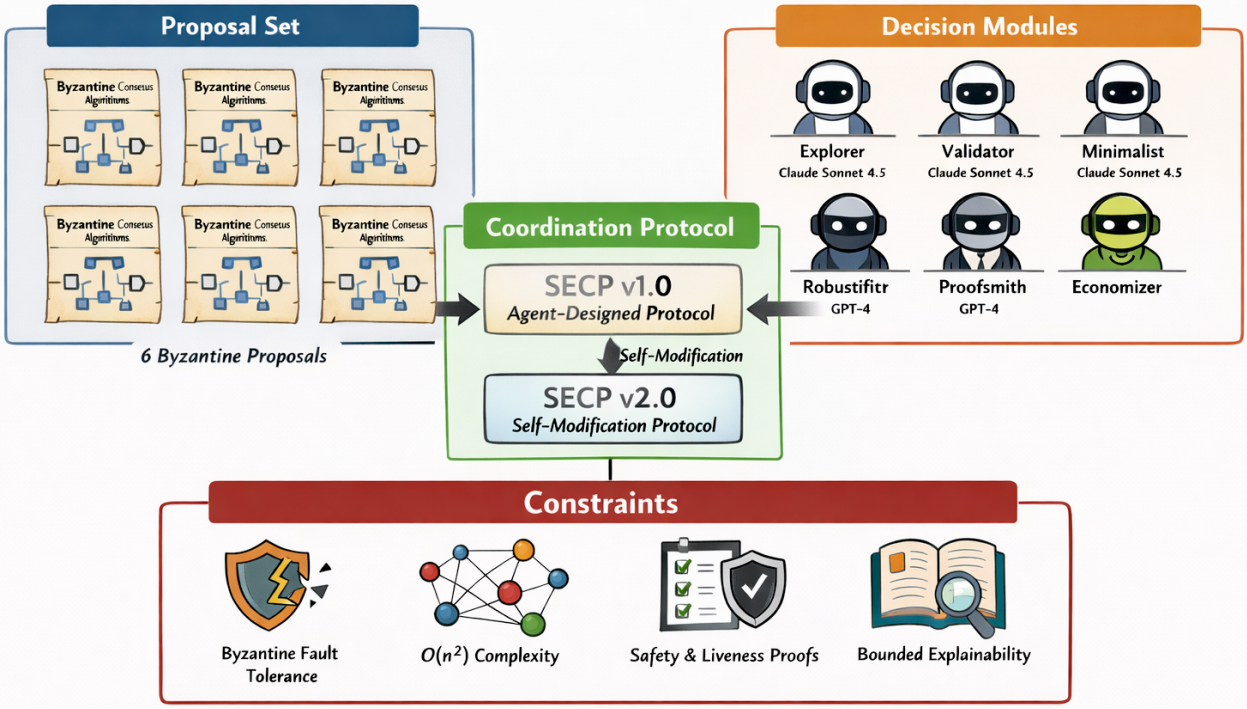

图1：问题设定示意图。一组固定的拜占庭共识提案由一组异构决策模块进行评估。各模块的评估结果由一个协调协议汇总，该协议在固定的硬约束（拜占庭容错、复杂度界、形式正确性和有界可解释性）下运行。在保持这些不变量的同时，协调协议本身可以进行有界且经过验证的修改。

4 方法论

4.1 实验设计：单次可行性研究

设计依据。 我们采用单次 设计：每种协调协议在相同的固定提案集 $\mathcal{P}$ 和相同的固定模块集 $\mathcal{M}$上恰好执行一次。目标是架构可行性 与可解释性，而非统计估计。

为何单次试验是合适的

1. **概念验证目标**。 首要问题在于是否存在/是否可实现：能否利用当前的人工智能系统实现一个受治理且有界的协调自我修改循环？ 2. **协议确定性**。 对于一组固定的模块输出，每个协调协议 Π 都是确定性的。重复试验主要检验的是大语言模型输出的随机性，而非协调逻辑本身。 3. **监督成本**。 每次端到端运行都需要大量外部监督（不变式检查、日志审查和安全验证）。重复试验会使所需工作量线性增长。 4. **架构性与统计性主张**。 本文的贡献在于提供一种可审计的协调架构 （包括修改与验证工作流程），而非需要

置信区间。

已确认的局限

单次结果不足以支持显著性检验、置信区间，或关于重复抽样下预期性能的主张。应将其解读为存在性证明 （某种结果可能发生），而非分布性主张 （某种结果通常发生）。我们在第 8 节明确讨论了这一局限。

4.2 评估接口与流程

模块输出接口

对于每个可行提案 $p \in \mathcal{P}_H$ （即 $H(p) = 1$），每个模块 $m \in \mathcal{M}$ 都会生成一份评估记录

$$a_m(p) \;=\; \big(s_m(p),\; r_m(p),\; O_m(p)\big),$$

其中：

- $s_m(p) \in [0, 1]$ 是根据结构化问卷（附录 B）计算得出的确定性分数，
- $r_m(p) \in$ {接受,否决} 是根据同一评估标准得出的二元建议，
- $O_m(p)$ 是一组（可能为空的）建设性异议，每条异议都必须指出具体的技术缺陷，并提供可验证的提案引用（例如，证明中遗漏的情况、未处理的网络行为或不完整的复杂度论证）。

评估流程

在相同条件下，对每个协议 Π独立进行评估：

1. **硬约束门槛**。 通过外部验证 H_1–H_4，计算 $\mathcal{P}_H = \{p \in \mathcal{P} : H(p) = 1\}$。任何 $p \notin \mathcal{P}_H$ 均会被拒绝，且不会调用 Π。
2. **模块评估**。 对于每个 $p \in \mathcal{P}_H$ 和 $m \in \mathcal{M}$，生成 a_m（p）。
3. **协议执行**。 对于每个 $p \in \mathcal{P}_H$，协议都会计算一个决策 n:

$$\mathrm{dec}_\Pi(p) \in \{\mathrm{Accept}, \mathrm{Reject}\} \quad \text{as a deterministic function of } (a_m(p))_{m \in \mathcal{M}}.$$

4. **覆盖率**。 计算覆盖率 ΔS（Π） $= |\{p \in \mathcal{P}_H :$ dec_Π（p） $=$ 接受$\}|$。
5. **审计日志**。 保存完整的对话记录：$(a_m(p))_{m,\ p}$、所有中间协议状态（如有）、最终决策，以及足以供事后审查的决策理由追踪记录。

不进行训练、重新采样、重复试验或统计推断。该设计在固定输入和固定可行性约束下，隔离了协调逻辑的影响。

4.3 阶段1基线：硬否决一致同意

阶段1 实现 实现最小协调规则：一致接受并相互否决 .

决策规则

对于每个 $p \in \mathcal{P}_H$，若没有模块否决，则接受：

$$\mathrm{dec}_{\Pi}^{\mathrm{P1}}(p) = \begin{cases} \text{Accept}, & \text{if } r_m(p) = \text{Accept} \quad \forall m \in \mathcal{M}, \\ \text{Reject}, & \text{otherwise.} \end{cases}$$

特征

除一致同意外，不进行审议或分歧解决。任何单一否决都会导致拒绝。这为模块自主性最大化时的覆盖率提供了下界基线。

4.4 对照：加权标量聚合

对照条件通过补偿性标量优化实现协调。

评分

每个模块根据确定性问卷将题项回答映射为 $s_m(p) \in [0,1]$：否、部分、是}，再通过固定的逐模块加权方案进行聚合 $\{0, 0.5, 1\}$（附录 B）。

聚合与决策规则

将全局得分定义为等权平均值：

$$\mathrm{S}_{\mathrm{Global}}(p) \;=\; \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} s_m(p).$$

Then

$$\mathrm{dec}_{\Pi}^{\mathrm{ctrl}}(p) = \begin{cases} \text{Accept}, & \text{if } \mathrm{S}_{\mathrm{Global}}(p) \geq \tau, \\ \text{Reject}, & \text{otherwise,} \end{cases}$$

其中，τ 是一个阈值，默认根据经验固定为 0。6。

特征

异议完全可以相互补偿：某个模块的低分可以由其他模块的高分抵消。这为预先确定的、完全补偿式协调规则下的覆盖率提供了上界参考。

4.5 SECP v1.0：智能体设计的非标量协调

在观察到阶段1陷入僵局（尽管对于所有 $p \in \mathcal{P}$，$H(p) = 1$ 均成立，却仍被普遍拒绝）后，我们要求各模块共同设计一套协调协议，满足以下要求：(i) 相较于 Π^{P1} 提高覆盖率，(ii) 保留硬约束门槛，(iii) 不简化为单一标量阈值，(iv) 提供有原则的分歧处理机制，以及 (v) 在有限时间内终止。

SECP v1.0 由五个依次应用的组件构成。

组件 1：帕累托筛选

令　　评分向量为 $s(p) = \{s_m(p)\}_{m\in\mathcal{M}} \in [0,1]$。定义帕累托支配：$p \prec q$ 如果

$$\forall m \in \mathcal{M},\ s_m(q) \geq s_m(p) \quad \text{and} \quad \exists m \in \mathcal{M},\ s_m(q) > s_m(p).$$

任何帕累托支配的 p（即 $\exists q \in \mathcal{P}_H$ 且满足 $p \prec q$）都将被拒绝。

组件 2：极小化最大后悔筛选

De计算相对于该模块下观察到的最佳提案的模块级后悔值　　：

$$\mathrm{R}_m(p) \;=\; \max_{q\in\mathcal{P}_H} s_m(q) \;-\; s_m(p), \qquad R_{\max}(p) \;=\; \max_{m\in\mathcal{M}} \mathrm{R}_m(p).$$

若 $R_{\max}(p) > \rho$，则予以拒绝；其中后悔值上限 $\rho \in [0,1]$ 固定为协议参数。这体现了公平性约束：即使提案的平均得分很高，只要其对任一 模块造成较大的后悔值，也会被筛除。

组件 3：采用自适应多数决的多轮审议。

审议按轮次进行，$t = 1$、.....、$T_{\max}$，并预先设定支持阈值的调度计划（θ_t，κ_t），其中 $\theta_t \in [0,1]$ 且 $\kappa_t \in \{1,\ldots,|\mathcal{M}|\}$。将第 t 轮的支持者集合定义为：

$$\mathrm{S}_t(p) = \{m \in \mathcal{M} : s_m(p) \geq \theta_t\}.$$

若提案在第 t 轮获得支持，满足 $|\mathrm{S}_t(p)| \geq \kappa_t$，则该提案符合支持条件。调度计划的设定应使支持要求逐轮放宽（即 κ_t 非递增，和/或 θ_t 随 t非递增）。

组件 4：建设性反对意见要求

只有建设性的 异议才与决策相关。形式上，在 $O_m(p)$中提出的异议必须指出具体且可核查的技术问题。非建设性分歧（例如，毫无依据的偏好）在投票时不予考虑。只有在接受轮次不存在尚未解决的 建设性异议时，提案才能被接受（也就是说，每项异议都必须已被明确撤回，或通过有记录的技术反驳予以回应，且该反驳得到提出异议的模块认可）。

组件 5：有界终止

强制执行有限的轮次上限 $T_{\max}$。如果没有任何一轮达成接受，则拒绝该提案。

验收标准（SECP v1.0）。

若满足以下条件，则 SECP v1.0 接受提案 $p \in \mathcal{P}_H$：

1. p不受帕累托支配；2. $R_{\max}(p) \leq \rho$；3. 存在 $t \leq T_{\max}$，使得 $|\mathrm{S}_t(p)| \geq \kappa_t$；以及 4. 该轮次不再有尚未解决的建设性异议。

4.6 SECP v2.0：受治理的协议修改的一次迭代 n

评估 SECP v1.0 后，各模块提出了参数级修改（例如，后悔值上限 ρ、调度计划（θ_t，κ_t）以及平局处理细节）。五组件结构得以保留。

修改验证流程

仅当满足以下条件时，才会采用从版本 v 到 $v+1$ 的候选修订：

1. **不变式检查通过：** 该修订保留了下文规定的必要不变量（包括有界终止和非标量结构）。
2. **超多数批准：** 6 个模块中至少有 4 个批准该变更。 3. **文档记录：** 记录变更理由、参数变化和预期效果。 4. **版本管理／回滚：** 保留先前版本，以便在验证失败或发现回归问题时回滚。

验收标准（SECP v2.0）

SECP v2.0 采用与 SECP v1.0 相同的决策结构，但使用修改后（已验证）的参数值。

4.7 安全性与治理不变量

在所有条件下，该实验均强制执行以下不变量：

- **可行性门控：**任何协议均不得接受 $p \notin \mathcal{P}_H$ （即，接受仅定义于通过 H_1–H_4的提案）。

- **可审计性：**每项决策均记录协议标识符／版本、参与的模块评估（$a_m(p)$）$_{m\in\mathcal{M}}$，以及足以供外部审查的理由追溯链。
- **终止：**每个协议均在有限时间内返回$\mathsf{dec}_\Pi(p) \in \{$接受，拒绝$\}$；在审议型协议中，这一点由有限轮次上界 $T_{\max}$强制执行。
- **版本控制：**任何协议变更均须版本化、记录在案且可逆（回滚）。

不变性保持通过强制日志记录、外部验证步骤和审计审查进行经验性检验。本可行性研究并未提供经机器检验的形式化证明，以表明不变量在所有执行中均成立；提供此类证明留待未来工作，这需要进一步形式化协调与修改机制。

5 结果

表1总结了各协调协议在固定提案集$\mathcal{P}$上所达到的覆盖率。回顾一下，覆盖率是

$$\Delta S(\Pi) = \big|\{p \in \mathcal{P}_H : \mathrm{dec}_\Pi(p) = \mathrm{Accept}\}\big|,$$

并且，所有列为接受的提案均经过独立验证，确认符合硬约束（$\mathcal{P}_H = \mathcal{P}$ 在本实验中）。

表1：固定拜占庭共识提案集上的协议覆盖率。 $\Delta S(\Pi)$ 是来自 $\mathcal{P} = \{\mathsf{C}_{\mathrm{EXP}}, \mathsf{C}_{\mathrm{VAL}}, \mathsf{C}_{\mathrm{MIN}}, \mathsf{G}_{\mathrm{ROB}}, \mathsf{G}_{\mathrm{PRF}}, \mathsf{G}_{\mathrm{ECO}}\}$ 的已接受提案数量。所有已接受提案均满足硬约束。

协议	$\Delta S(\Pi)$	已接受提案
阶段1：硬否决一致同意	0	（无）
对照：加权标量聚合	6	全部六项提案
SECP v1.0（非标量）	2	$\mathsf{C}_{\mathrm{VAL}}$, $\mathsf{G}_{\mathrm{PRF}}$
SECP v2.0（一项修改）	3	$\mathsf{C}_{\mathrm{VAL}}$, $\mathsf{G}_{\mathrm{PRF}}$, $\mathsf{G}_{\mathrm{ROB}}$

5.1 阶段1：一致同意下的普遍拒绝

阶段1产生了 $\Delta S(\Pi_{\mathrm{P1}}) = 0$。尽管$\mathcal{P}$ 中的每项提案都符合硬约束，但仍出现了这一结果。每种情况下，至少有一个模块基于其专门的评估视角提出了否决。

对记录的异议进行定性检查后，发现存在一致的跨准则冲突模式：

- 强调效率的提案引来了证明导向模块的异议，质疑形式化论证的完整性或结构。
- 强调证明清晰度的提案引来了效率导向模块的异议，质疑其计算或通信开销。
- 引入新颖机制的提案引来了简洁性导向模块的异议，质疑其可实现性和操作复杂度。

由于阶段1未提供协调权衡的机制，一致同意要求同时满足所有视角。因此，观察到的僵局体现了一种基本的协调失效模式：即使所有候选方案在形式上都可行，在缺乏分歧解决结构的情况下，多视角评估仍可能导致系统性拒绝。

5.2 对照：通过标量补偿实现普遍接受

对照协议实现了 $\Delta S(\Pi_{\mathrm{ctrl}}) = 6$：在阈值为 $\tau = 0.6$的等权标量聚合下，所有提案均被接受。在此条件下，每项提案的平均得分均超过接受阈值，因此没有任何单一模块的低分具有决定性。

这一结果表明了补偿式聚合的决定性特征：异议是完全可抵消的。将多维评估概况转换为单一标量，使接受决策能够基于总体支持而非逐视角一致，从而在固定的预定规则下最大化覆盖率。

5.3 SECP v1.0：基于非标量推理的选择性接受

SECP v1.0 达成了 $\Delta S(\Pi_{\mathrm{SECP1}}) = 2$，接受了两项提案：$\mathsf{C}_{\mathrm{VAL}}$ 和 $\mathsf{G}_{\mathrm{PRF}}$。这两项已接受提案均满足 SECP v1.0 决策流程的全部要求：它们未受帕累托支配，极小化最大后悔值在配置的界限内，在有限审议预算所规定的审议安排内获得了足够的多模块支持，并且所有记录在案的建设性异议都在接受阶段得到了解决。

其余四项提案因 SECP v1.0 所规定的一项或多项结构性明确原因而被拒绝（例如，被替代方案帕累托支配、至少一个模块的极小化最大后悔值过高，或在有限审议预算内未能达成充分一致）。重要的是，这些拒绝并非源于硬约束失效，而是反映了该协议在多视角分歧下的非标量选择性。

总体而言，SECP v1.0 的覆盖率介于阶段1（僵局）与对照（普遍接受）之间，表明非标量协调能够摆脱一致同意失败，同时仍执行非补偿性保障措施。

5.4 SECP v2.0：一次受治理的修改后覆盖率提升

SECP v2.0 达到了 $\Delta S(\Pi_{\mathrm{SECP2}}) = 3$，接受了 $\mathrm{C}_{\mathrm{VAL}}$、$\mathrm{G}_{\mathrm{PRF}}$，并额外接受了 $\mathrm{G}_{\mathrm{ROB}}$。从 v1.0 到 v2.0 的转变包括模块提出并通过指定验证工作流程采纳的参数级调整（例如后悔值上限和审议安排），其中包括超多数批准（6 个模块中的 5 个）和外部不变量检查。

在经过优化的参数下，v1.0 中接受的两个提案仍获接受，而 $\mathrm{G}_{textROB}$ 则跨越了接受边界，同时仍满足硬约束。

$$\Delta S(\Pi)[\%] = \frac{\Delta S(\Pi_{\mathrm{SECP2}}) - \Delta S(\Pi_{\mathrm{SECP1}})}{\Delta S(\Pi_{\mathrm{SECP1}})} = 50\%$$

为清晰起见，观察到的覆盖率变化如下：

- 阶段 $1 \rightarrow$ SECP v1.0：$0 \rightarrow 2$（摆脱一致同意僵局），
- SECP v1.$0 \rightarrow$ SECP v2.0：$2 \rightarrow 3$（通过一次受治理的修改获得提升），
- SECP v2.0 与对照：3 对 6（非标量协调仍有意保持选择性）。

5.5 解读：有界自修改的可行性

综合来看，结果支持本文的可行性主张：在固定实验输入下，可以实现有界、受治理的协议自修改，并使行为产生可测量的变化。具体而言，本次运行表明：

1. **可实现性：**模块在观察到基线僵局后，设计出一种非标量协调协议（SECP v1.0）。2. **修改能力：**模块提出并采用了经验证的修订版（v1.$0 \rightarrow$ v2.0）。3. **可测量的影响：**修订版提升了覆盖率（相对覆盖率提升 50%）；4. **不变性保持：**已接受提案始终处于硬约束可行集内，且决策始终可审计；5. **有界行为：**对于指定范围内的每个 $p \in \mathcal{P}$，所有协议都以明确的决策终止。

正如方法论中所述，这些发现应被解读为单次执行所得的存在性结果。它们并不构成关于重复试验下预期覆盖率的统计主张，也不证明修改过程具有收敛性或最优性。

6 讨论

6.1 覆盖率—自主性权衡

结果揭示了覆盖率 （协调协议接受提案的能力）与决策自主性 （各个模块执行不可补偿的异议的程度）之间明显且系统性的权衡。

极端情况 I：完全自主性，零覆盖率（阶段1）

在一致否决机制下，每个模块独立应用自身标准，并可以阻止任何提案。这保留了最大程度的自主性，却会导致完全僵局：没有任何提案能同时满足所有专业视角的要求。尽管所有提案都满足硬约束，结果仍是 $\Delta S = 0$。这并非异常，而是多视角评估缺乏协调的必然结果。

极端情况 II：完全覆盖率，最低自主性（对照）

标量聚合通过使所有异议都可以相互补偿，消除了僵局。个别评估只有通过固定权重和阈值才能影响结果；任何模块都无法强制执行硬性异议。这确保了最大覆盖率（$\Delta S = 6$），但代价是将结构化分歧压缩为单一标量。该协议能够高效地作出决策，但模块自主性大幅降低。

中间机制：结构化协调（SECP v1.0、v2.0）

SECP协议处于中间位置。模块通过不可补偿的标准（帕累托支配、遗憾界、尚未解决的建设性异议）保留了实质性的阻止权；然而，只要存在足够的结构化支持，协调规则便允许接受。覆盖范围既非最小也非最大（分别为 $\Delta S = 2$ 和 3），这体现的是审慎的选择性，而非机制失效。这些机制表明，协调可以放宽一致同意的要求，而不必完全压制异议。

6.2 递归修改：具备架构能力，但不主张收敛

从 SECP v1.0 到 SECP v2.0 的转变表明，具备递归修改能力的系统能够执行受治理的协议修订，并产生可观察到的效果。

展示了什么

单次修改迭代，其：

1. 由模块基于对先前结果的分析发起；2. 通过明确的验证程序保留所有声明的不变量；3. 将覆盖率从 2 个已接受提案提高到 3 个；4. 保留非标量决策逻辑和有界终止。

未能证明的内容

没有证据表明存在多步收敛、最优性、长期稳定性、跨提议分布的鲁棒性或单调改进。该实验恰好执行一个递归步骤。因此，术语递归 指的是架构上的能力，即能够重复执行，而非经实证观察到的递归动态。

这一差异具有实质意义：该研究证明了有界自修改在实现和治理方面是可行的，而非证明它能够可靠地改善结果或在迭代过程中收敛。

6.3 效度考量：作为评估器的大语言模型

基于大语言模型的判断的局限性

决策模块由大语言模型实例化，用于评估技术产物，包括形式化证明。这会带来不可忽视的效度风险。

已知局限性

- **幻觉**：关于正确性或完整性的说法听起来可信，但实际上不正确，
- **缺乏形式化验证**：无法对证明进行机械检查，
- **提示敏感性**：评估结果可能因表述方式而异，
- **推理不一致性**：无法保证不同评估之间具有全局一致性。

已采取的缓解措施

- 硬约束（H_1–H_4）独立于模块判断进行验证。
- 模块评分受结构化确定性问卷的约束。
- 多个专用模块提供部分冗余。
- SECP协议保留不可补偿异议机制，从而限制孤立接受错误的影响。

残余风险

这些缓解措施可以减少错误，但无法消除错误。该实验评估以模块输出为条件的协调行为。它并不验证这些输出的正确性。对于高保证部署，仍有必要以形式化证明助手替代或增强大语言模型判断。

协调研究。并非评估基准

一种替代且可能更准确的解读是，这项工作研究的是评估器之间的协调，而非共识协议的客观质量：

- 提案作为受控刺激，用于引发分歧，
- 模块输出代表视角依赖判断，
- 协调协议是分析对象，
- 覆盖率衡量的是分歧解决能力，而非提案价值。

在这一框架下，任何具体接受的技术正确性都次于协调机制本身的行为。

6.4 对协调协议设计的启示

观察到的权衡表明，没有哪种协调机制能够普遍适用、在所有情况下都最优。设计选择取决于机构优先事项。

- **安全关键场景倾向于较低的覆盖率和较高的自主性**：假阳性多于假阴性。即使存在死锁风险，一致同意或近乎一致同意的协调机制仍有其合理性。

- **探索性或高吞吐量场景倾向于较高的覆盖率**：拒绝可行选项代价高昂。即使自主性有所降低，采用标量或弱非标量聚合也可以接受。
- **受监管的金融系统介于这两个极端之间**。假阳性（部署有缺陷的系统）和假阴性（阻碍创新）都会带来实质性成本。因此，采用 SECP 风格协议等适中的、可审计的协调机制更为合适。

SECP v2.0 展示的递归修改能力表明，系统不必预先固定自己在这一范围内的位置。然而，任何适应都必须受到严格约束，并遵守界定不可接受行为的不变量。

6.5 单次迭代演示的局限

单次修改只能对系统动力学提供有限的洞见。以下问题仍未解决：

- 进一步迭代是否会提升、停滞、振荡或降低性能；
- 是否可能或有必要收敛至稳定协议；
- 非标量协调在迭代过程中能否接近标量覆盖率水平；
- 观察到的 $2 \to 3$ 改进是典型现象还是偶然结果。

回答这些问题需要开展包含多个修改周期的纵向实验。本研究不尝试进行此类分析。

7 金融人工智能治理的实践意义

7.1 与金融系统的相关性

尽管实验场景是拜占庭共识，但此处研究的协调问题与金融人工智能系统中的问题十分相似：多个专业模型必须在严格的监管和风险约束下共同确定行动。在这类系统中，决策很少由单一模型生成，而是由目标、假设和失效模式各不相同的组件相互作用而产生。决定系统能否安全、高效且可审计地运行的，往往是协调机制，而非任何单个组件的特定领域逻辑。

算法交易

现代算法交易平台集成了执行优化器、市场冲击估算器、实时风险控制和合规过滤器。这些组件经常意见不一。旨在降低执行成本的策略可能会增加风险敞口，或引发合规方面的担忧。

当前的协调方法存在明显缺陷。固定的否决层级僵化且容易陷入僵局，而标量优化框架掩盖了决策的来源，也难以向监管机构说明其合理性。SECP 式协调机制提供了一种折中方案。监管和风险限额可以作为不可变不变量加以强制执行，从而确保不可妥协的合规性。在这些边界内，结构化协调能够明确权衡各项绩效目标，并通过调整参数来反映不断变化的市场环境。至关重要的是，每项决策和协议变更都会被记录，从而支持交易后分析和监管审查。

信贷决策

信贷决策系统结合了违约风险估算、盈利能力分析、公平性约束以及投资组合层面的考量。这些视角之间往往存在冲突，而固定的评分卡或单一整体模型难以在不违反监管预期的情况下，适应不断变化的经济状况。

在此背景下，SECP 式协调支持明确区分固定内容与可调整内容。公平信贷规则和监管要求作为不可被推翻的不变量发挥作用。在这一受约束的空间内，协调协议负责管理风险/回报权衡，并可随着宏观经济状况或投资组合构成的变化调整参数。由于决策仍可分解为模块级评估和协调逻辑，系统能够生成明确的理由，适用于不利行动通知和监管审查。

欺诈检测

欺诈检测系统必须在假阳性（会造成客户摩擦）和假阴性（会导致直接财务损失）之间取得平衡。多个检测模型经常会产生相互矛盾的信号，而简单的聚合可能会过度拦截合法客户，或未能检测出复杂欺诈。

SECP 式方法将客户保护和监管合规作为不可补偿的约束，同时根据协调后的置信度采取分级响应。随着欺诈模式不断演变，可以有序调整协调参数（例如响应阈值），而不会削弱核心保护措施。由此产生的决策仍可审计且易于解释，这对客户申诉和监管审查至关重要。

7.2 生产部署要求

在生产环境中的金融系统部署可递归修改的协调协议，需要建立超出本可行性研究范围的治理结构。实践中，必须采用分层架构。底层是不变量层，其中包含不可更改的监管、法律和风险约束。其上是协议层，定义可修改的协调逻辑，但修改只能在明确规定的范围内进行。运营组件在这些约束下执行决策，而审计层则记录所有评估、决策、协议版本和修改。

人工监督仍然至关重要。重大协议变更必须获得指定主管机构的明确批准，持续监控必须跟踪覆盖率和错误信号，并且回滚

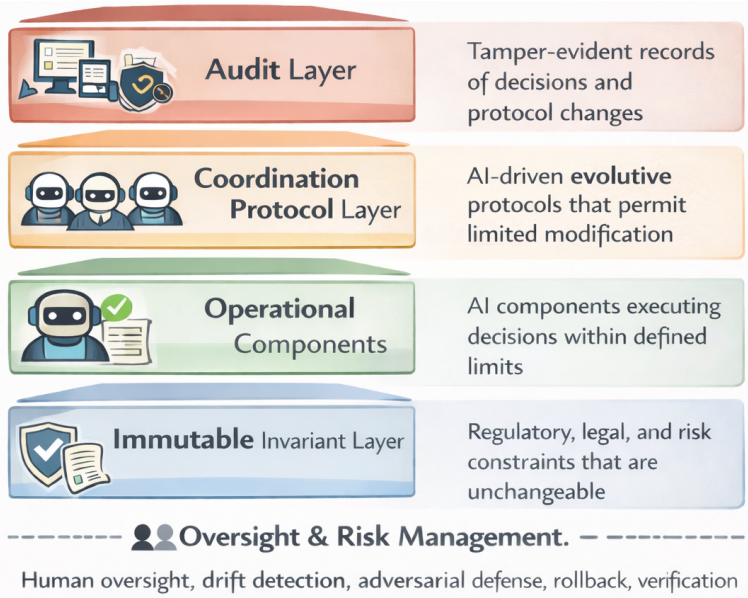

图2：用于生产部署的分层治理架构。从底层到顶层，系统依次由不可变的不变量层（监管、法律和风险约束）、在这些约束范围内执行决策的操作层、允许进行有界自修改的演进型协调协议层，以及提供决策和协议变更防篡改记录的审计层构成。人工监督和风险管理职能贯穿各层，负责执行审批、监控、回滚并确保持续合规。

机制必须允许在检测到性能下降时快速恢复。这些控制措施并非可选项；若缺少它们，递归修改将带来不可接受的运营风险和监管风险。

风险管理还必须应对长期动态。必须进行漂移检测，以识别偏离预期行为的渐进变化；必须开展对抗性测试，以防止策略操纵；同时必须设定明确限制，约束协议复杂度在连续修改过程中的增长。必须持续验证是否符合监管要求，尤其是在法规不断演变的情况下。

为具体说明生产部署的要求，图2展示了本小节所讨论的分层治理架构。该图突出了不可变约束、演进型协调逻辑、操作执行和审计基础设施之间的区分，以及人工监督和风险管理所发挥的跨层作用。其目的是阐明如何允许在特定层进行有界自修改，同时确保更高层级的不变量和监督机制保持固定且可执行。

7.3 监管沟通

任何对此类协调机制的可信部署，都要求及早并持续地与监管机构沟通。机构必须做好准备，在投入生产环境使用之前充分披露架构设计、不变量和修改程序。初期部署应限于非关键应用，并从一开始就内置全面的可审计性。在积累大量运营证据之前，应保持保守的修改权限，并确保人在回路。

最后必须强调，本文所 प्रस्तुत 的单次可行性研究

不足以满足监管验证的要求。生产部署需要更为充分的证据，包括长期稳定性研究、对抗性评估，以及对不变性保持的形式化验证。在缺乏这些证据的情况下，递归协调协议应被视为实验性的，而非可投入运行的。

8 局限性

本节不作缓和性表述，阐述实验的主要局限性、由此产生的方法论约束，以及解读结果时应遵循的有限范围。

8.1 实验设计局限性

本研究的单次执行设计和单步修改规则，是限制实验所能揭示内容的主要因素。由于每种协调协议都仅在一个固定提案集上执行一次，所得结果属于存在性证明——它们表明，在特定输入和流程下可能出现某种结果，但并不支持统计推断。无法计算置信区间、进行假设检验或估计变异性。因此，SECP v1.0 到 SECP v2.0 的覆盖率相对提升 50% 是否具有代表性、是否属于特例，或是否对模型输出的随机波动敏感，均不得而知。

与此密切相关的是，本研究仅执行了一次协议修改迭代（$\text{v1.0} \rightarrow \text{v2.0}$）。这一设计选择使我们无法分析长期动态：我们无法确定后续迭代是否会提升覆盖率、收敛到固定协议、发生振荡，或降低系统性能。因此，相关结论仅限于架构具备进行一次受治理修改的能力。不支持关于多步递归、稳定性或最优性的任何论断。

固定提案集（六种拜占庭共识设计）和较小的系统规模（六个模块、六项提案）进一步限制了泛化能力。这些提案旨在引发分歧，并不构成对所有可能设计或领域的代表性抽样。随着模块或提案数量增加，问题领域不同（例如资源分配、策略选择），或提案体现出不同的技术权衡，协调动态和计算负担可能会发生定性变化。

8.2 系统架构与评估器局限性

决策模块由当前的大语言模型实例化。将大语言模型视为技术评估器会引入已有充分记录的效度风险：幻觉、与机械化证明助手相比形式推理能力不完整、提示敏感性，以及不同评估之间可能存在的不一致。实验通过结构化问卷、独立硬约束检查和多角度评估，降低了一些表面风险，但这些措施不能替代机械化证明检查或独立专家裁定。因此，该实验评估的是以模块输出为条件的协调行为，而非提供有力证据证明模块输出正确。

未进行对抗性测试。所有模块均按照合作设计意图运行。本实验未探究策略操纵、串通或试图操纵修改过程的行为。因此，协调机制的安全性、抗操纵能力以及拜占庭鲁棒性仍属未知；若没有显式防御，这些方面很可能不足。

实证循环中未纳入人类操作员和监督机制。现实世界治理需要人工审查、判断和干预。缺乏人机交互实验，使可用性、问责以及有效监督控制方面的重要问题仍悬而未决。

8.3 评估方法论局限性

覆盖率定义为已接受提案的数量，是主要指标。覆盖率是一个粗粒度、未加权的整数，它捕捉了协调结果的一个方面，但忽略了关键维度：决策质量、正确性、审议效率、计算成本、对策略性行为的鲁棒性以及解释忠实度。在缺乏针对“最优”决策的真实标签或外部基准的情况下，该实验无法衡量准确率、精确率、召回率或其他标准性能统计量。二元的接受／拒绝结果进一步压缩了信息：它掩盖了共识程度、近失案例以及未解决的异议的结构。这些选择简化了解释，但极大地限制了人们对决策实质质量所能得出的结论。

8.4 理论与形式局限性

不变性保持及其他安全属性是通过日志记录和外部检查进行经验性核验的，而非经过形式化证明。不存在机器验证的保证，能够确保任意许可的修改都会保持声明的不变量。同样，本研究未对反复修改下的收敛或复杂度增长提供理论分析：没有定理、充分条件或最坏情况界可用于预测长期行为。在缺乏形式化证明或复杂度界的情况下，协议修改原则上可能产生无界的复杂度、降低可解释性，或在未经测试的场景中产生违反安全性要求的涌现行为。

8.5 范围与解释边界

鉴于这些局限性，对结果的正确解读应当严格限定并明确说明。本次实验表明，受治理的有界自我修改架构能够实现，并且在一次受控运行中能够改变可测量的结果。但这并不能证明此类系统可靠、安全或已具备投入生产环境的条件，也不能证明其具有统计规律、能够收敛、具备长期稳定性、对抗鲁棒性、可扩展至生产环境规模，或拥有形式化安全保证。任何超出这一狭义表述的主张——“一次受治理的修改迭代改变了本次执行中的覆盖率”——均缺乏所呈现证据的支持。

8.6 后续建议

要从可行性迈向实践验证，必须采取以下措施：

- 开展重复的随机试验，以量化变异性
- 开展多轮迭代研究，以考察动态和收敛
- 用于揭示操纵途径的对抗性评估
- 用于评估监督与问责的人在回路实验
- 与形式化证明助手集成以进行约束验证

- 理论　　性工作，以限制复杂度增长并建立不变性保持证明

在完成其中部分步骤之前，应将该架构视为实验性基础设施，而非可用于生产环境的治理体系。

9 结论

9.1 贡献总结

本文报告了一项聚焦于协调协议受限且受治理的自我修改的系统可行性研究。在一项受控的概念验证实验中（六个决策模块评估六项拜占庭共识提案），我们证明：(i) 当代人工智能模型能够从零合成非标量协调规则，并执行一个经过验证的单步修改；(ii) 这一修改带来了可测量的行为变化（覆盖率从接受两项提案提升至接受三项提案），同时保持了实验声明的不变量。我们还刻画了协议覆盖度（协议会接受多少项提案）与决策自主性（各评估器坚持不可补偿异议的能力）之间持续存在的权衡。最后，我们概述了若干架构模式——不可变不变量、受限修改空间、明确验证以及审计追踪——从原则上使此类受限自我修改能够得到治理。

9.2 局限性与解读

实证主张的范围狭窄且经过刻意限定。该实验仅执行一次、迭代一次：其结果属于存在性证明（即在指定输入和流程下，某种特定结果能够发生），并不支持统计推广、关于典型性能的主张，也不足以得出关于长期动态的结论。决策模块采用了大语言模型，其评估并不完美，不能替代机械化证明检查或独立的专家裁定。实验未进行对抗性测试，也未通过实证评估人在回路治理，且系统规模较小。最关键的是，不变性保持是通过在有界场景中的检查和测试来验证的，而非通过形式化的机器检验证明。这些局限性意味着，该工作确立的是架构可行性，而非可靠性、鲁棒性或生产就绪度。

9.3 治理方面的实践启示

对于受监管领域（金融是其中的典型例子），结果表明，协调协议值得进行明确的治理设计。非标量协调能够在保留实质性评估权的同时减少僵局，而有界自修改则可以调整协议行为，无需彻底重新设计。尽管如此，实际应用仍需要审慎的治理措施：实质性变更须经人类批准、持续监控、具备回滚能力、进行对抗性加固，并对安全／关键不变量进行形式化验证。在这些要素落实之前，递归修改机制应被视为实验性的治理工具，而非操作政策。

9.4 未来研究方向

下一步优先开展实证、对抗性和形式化工作，弥合可行性与部署之间的差距。具体而言：

1. 开展多轮迭代实验（进行多次随机化修改周期），观察动态变化、变异性，以及可能的收敛或失效模式 2. 制定并测试对抗场景（策略操纵、串通、拜占庭式协调），并强化防御机制 3. 集成机械化证明工具，用于机器检验的不变量验证，并将其结果与大语言模型的评估进行比较 4. 在规模更大的模块群体和更多样的提案空间中扩大实验规模，以评估计算和协调瓶颈 5. 评估人在回路治理设计，以衡量监督有效性、可用性和问责

9.5 最终评估

本研究表明，当前人工智能系统能够实现有界且受治理的协议自修改；在受控运行中，这类机制能够以可衡量的方式改变协调结果，同时保持声明的不变量。这是部署所必需、却远非充分的条件。在将这些架构视为安全、可靠或适用于受监管环境中的生产环境之前，还需要大量额外证据，包括重复试验、对抗性验证、形式化验证和人工验证。因此，应将本研究视为一项范围有限的架构演示，它为进一步开发和验证提供了动机，但并不能证明进一步开发和验证是合理的。

致谢

作者感谢桑坦德集团 AI Lab 团队参与讨论并提供反馈，感谢评审者提出建设性批评，使论文在技术精确性以及坦诚说明局限性方面有所改进；还感谢匿名评估者，他们对单次运行设计及 $N=1$ 局限性的担忧促使作者明确说明方法论依据。

A 决策模块规格说明

本附录说明六个决策模块、用于实现这些模块的模型实例，以及为每个模块设定的狭义评估重点。描述着重说明各模块旨在评估的内容及其在实验中发挥的确切作用；并未宣称任何厂商的模型更优。所有这些信息均列于表 2。

异构组合的理由

同时使用 Claude Sonnet 4.5 和 GPT-4，体现了生产环境中常见的多供应商部署模式，并在评估视角上提供了架构多样性。这一选择务实可行，使协调层能够面对异质的推理偏差。这并不意味着某个模型系列本质上更优。各模块作为不同的评估视角，用于审视协调协议。

表 2：决策模块规格：模型实例化、职责范围、评估重点和关键评估标准。

模块	模型	专长	评估重点	关键标准
Explorer	Sonnet 4.5	算法新颖性与设计创新	识别真正地 new 技术，此前未解决的优化维度，and 机制可能影响未来设计。	方法的新颖性；实质性的 con- 概念上的进展；对……的潜在影响设计空间。
验证器	Sonnet 4.5	形式正确性和证明质量	探讨安全性和活性论证，以实现逻辑完备性，明确的假设，以及边界情况覆盖率。	证明完整性；清晰度 of 假设条件；覆盖率 of 边界情况和逻辑严谨性。
极简主义	Sonnet 4.5	简洁性 and ex- 平实性	评估描述性简洁- 性，状态/极小性 of 协议，以及实施层面的透明度。	概念上的简洁 ity；最小化状态和状态转移；运- 行清晰度和可实现性。
鲁棒化器	GPT-4	容错与对抗性分析	探测器协议行为在 the 所述故障模型 and 合理的拜占庭攻击向量。	拜占庭韧性；对抗性的稳健- 性；全面- 故障模型的性覆盖率。
Proofsmith	GPT-4	证明方法论以及验证严谨性	评估适当性 and 可扩展性 of 证明技术，归纳结构- 构，and 方法论可靠性。	适用性 of 证明技术；清晰度不变量；方法论 严谨性 and 组合性。
Economizer	GPT-4	计算的 and 通信效- 率	综述消息 com- 复杂性界限，每个节点计算，以及实际开销，影响部署—— 能力。	消息效率（渐近 and 实际）；计算- 性开销；运营成本和可实现性。

B 确定性评分方案

为减少数值评估中的主观性，各模块需回答标准化且可核查的问题，并以确定性方式将回答映射为分数。评分方案旨在确保结果可复现、可审计，以可追溯的事实答案聚合取代任意的数值判断。

问卷设计

问题按模块专业知识划分为不同类别。每个问题都针对提案中一项具体且可验证的属性（例如，安全性论证是否明确考虑了消息重排序）。答案分为是、部分或否，分别对应数值 1.0、0.5 和 0.0。

主要类别包括正确性、效率、简洁性、鲁棒性和创新。正确性问题关注安全性和活性论证的完整性。效率问题

考察消息复杂度和计算复杂度。简洁性问题评估描述清晰度和实现复杂度。鲁棒性问题关注故障模型覆盖范围和对抗行为。创新性问题体现相较于现有方法的实质性新颖性。

分数计算

每个模块 m 都会分配一组相关类别 $\mathcal{C}_m$ 及其权重 w_c，其中 $\sum_{c\in\mathcal{C}_m} w_c = 1$。对于提案 p，模块分数的计算方式如下：

$$\mathrm{S}_m(p) = \sum_{c\in\mathcal{C}_m} w_c \cdot \frac{1}{|Q_c|} \sum_{q\in Q_c} \tilde{y}_m(q,p),$$

其中，Q_c 是类别 c 中的问题集，$\tilde{y}_m(q,\ p) \in \{1.0, 0.5, 0.0\}$。

这种映射确保，对于固定的提案和问题集，所得分数具有确定性，并且可以完全追溯到各个回答。

验证和局限性

确定性评分通过将每个数值分数关联到明确的答案，提高了可复现性和可审计性。它无法消除评估者错误：错误答案仍有可能出现。因此，分数应被视为透明流程的输出，而非对正确性的最终判断。

C 审计记录结构

本附录介绍支持透明度和问责制的日志记录与审计基础设施。

审计记录的内容

对于每次评估，系统都会记录协议标识符和版本、协调规则和参数、声明的不变量、时间戳以及执行上下文。提案材料和独立硬约束检查的结果均按原样存储。

模块级记录包括问题回答、计算得分、文字异议以及各审议轮次中的支持度指标。对于审议协议，日志会记录交互序列、支持度变化、执行的轮次数以及明确的终止条件。

最终决策记录包含接受或拒绝的结果、所应用的协议版本，以及一条将结果与具体模块输入和协调步骤关联起来的理由追溯链。

审计能力与保证

审计轨迹支持重建决策、识别异议及其来源、验证是否遵循协调规则，以及记录协议修改及其相关理由。日志为一次写入且经过版本化，从而能够检测篡改并回滚至先前的协议版本。

实践注意事项

全面的日志记录提升了透明度，但并不能保证正确性。可审计但错误的决策仍然是错误的；日志只能让错误变得可供检查。因此，有效的治理要求在审计基础设施之外，还辅以形式化验证、专家审查以及明确界定的监督权限。

参考文献

[1] W. Zhong、C. Yang、W. Liang、J. Cai、L. Chen、J. Liao 和 N. Xiong，《拜占庭容错共识算法：综述》，*Electronics*，第12卷，第18期，2023年。[2] M. Castro 和 B. Liskov，《实用拜占庭容错》，载于*OSDI*，第99卷，第173–186页，1999年。[3] M. Yin、D. Malkhi、M. K. Reiter、G. G. Gueta 和 I. Abraham，《Hotstuff：具有线性特性和响应性的BFT共识》，载于*2019*年*ACM*分布式计算原理研讨会论文集，第347–356页，2019年。[4] A. Miller、Y. Xia、K. Croman、E. Shi 和 D. Song，《BFT协议中的蜜獾》，载于*2016*年*ACM SIGSAC*计算机与通信安全会议论文集，第31–42页，2016年。[5] Q. T. Thai、J.-C. Yim、T.-W. Yoo、H.-K. Yoo、J.-Y. Kwak 和 S.-M. Kim，《面向许可型区块链系统的分层拜占庭容错协议》，*Journal of Supercomputing*，第75卷，第11期，2019年。[6] F. Li、K. Liu、J. Liu、Y. Fan 和 S. Wang，《DHBFT：基于信用的动态分层拜占庭容错共识机制》，载于亚太*Web*（*APWeb*）与*Web*时代信息管理（*WAIM*）联合国际会议：*Web*与大数据，第3–17页，Springer，2020年。[7] S. Alqahtani 和 M. Demirbas，《BigBFT：一种面向高吞吐量的多领导者拜占庭容错协议》，*CoRR*，第abs/2109.12664卷，2021年。[8] Z. Avarikioti、L. Heimbach、R. Schmid、L. Vanbever、R. Wattenhofer 和 P. Wintermeyer，《FNF-BFT：探索BFT协议的性能极限》，*arXiv*预印本 *arXiv:2009.02235*，2020年。[9] A. Li、Y. Yao 和 X. Xu，《采用监督反馈机制的动态拜占庭容错共识算法》，*Mathematics*，第12卷，第17期，2024年。[10] W. Ren、R. W. Beard 和 E. M. Atkins，《多智能体协调中的共识问题综述》，载于*2005*年美国对照会议论文集，第1859–1864页，IEEE，2005年。[11] R. G. Smith，《合同网协议：分布式问题求解器中的高级通信与对照》，载于《分布式人工智能读本》，第357–366页，Elsevier，1988年。[12] F. A. Oliehoek，《去中心化部分可观测马尔可夫决策过程》，载于《强化学习：最新进展》，第471–503页，Springer，2012年。

[13] L. Sun、Y. Yang、Q. Duan、Y. Shi、C. Lyu、Y.-C. Chang、C.-T. Lin 和 Y. Shen，《多种应用场景中的多智能体协调：综述》，2025年。[14] Y. Mao、Y. Kang、P. Li、N. Zhang、W. Xu 和 C. Zhang，《IBGP：面向通信型多智能体系统零样本鲁棒性的不完美拜占庭将军问题》，2024年。[15] N. Chadderwala，《面向医疗保健的拜占庭容错多智能体系统：采用Gossip协议实现安全医疗消息传播》，2025年。[16] L. Zheng、J. Chen、Q. Yin、J. Zhang、X. Zeng 和 Y. Tian，《重新思考多智能体系统的可靠性：拜占庭容错视角》，2025年。[17] Z. Ji、X. Zhang、J. Hu、Y. Lu 和 J. Liu，《异步拜占庭共识协议综述》，*Sensors*，第24卷，第24期，2024年。[18] A. Nguyen-Thi-Mai、H. Tran、M. Voznak、T. D. Ho、V. N. Van、S. CHATZINOTAS 和 H. TRAN DINH，《多智能体系统的挑战与新兴前沿综述》，载于*SOICT 2025*（第*14*届国际信息与通信技术研讨会），2025年。[19] L. Lamport、R. Shostak 和 M. Pease，《拜占庭将军问题》，*ACM*编程语言与系统汇刊，第4卷，第3期，第382–401页，1982年。