节奏作为一级安全变量

通过 FIT 框架识别 AI 系统中的不可逆操作

版本： GitHub
其他版本： LessWrong | English
FIT 规范： docs/v2.4.md（EST）、docs/v2.3.md（基线）

摘要

AI 系统的快速发展已将安全风险从孤立故障转向由加速部署和治理失配驱动的系统性、不可逆转变。现有的 AI 安全方法主要关注性能边界、对齐目标或事后控制机制，通常将时间视为隐式或次要参数。

在本文中，我们引入力-信息-时间（FIT）框架，将时间动力学——特别是更新节奏和不可逆性——提升为一级安全变量。FIT 将 AI 系统建模为在相互作用的力、信息约束和时间依赖的更新机制下演化的系统，从而能够明确识别**不可逆操作（IO）**：永久约束未来状态空间并消除回滚选项的系统级行为。

我们形式化了检测系统演化与治理反馈之间节奏失配的标准，提出了一组可观测的不可逆性风险指标，并引入**最小可行节奏治理（MVTG）**——一个轻量级、可审计的治理层，为高影响 AI 操作强制执行慢速权威、回滚窗口、节奏分层和熔断机制。

我们认为，管理节奏（而不仅仅是优化目标）对于防止虚假稳定状态是必要的——在这种状态下，AI 系统看似表现良好，却在结构上变得不可控。FIT 框架为 AI 安全、治理和部署实践提供了统一视角，强调预防不可逆伤害而非事后补救。

关键词： AI 安全、治理、不可逆性、时间动力学、部署风险、系统控制

1. 引言

AI 系统越来越多地作为社会技术栈运行：模型训练、数据管道、部署基础设施、产品界面和组织策略在加速发布更新的压力下共同演化。安全机制（评估、红队测试、事件复盘、外部监督）通常运行在较慢的周期上。这造成了一种结构性失效模式：系统可能以使未来纠正变得不可行的方式改变，即使单个变更看起来是良性的。

本文提出，**节奏**（更新的速率和顺序）应被视为一级安全变量。当系统演化超过治理反馈速度时，安全干预变成回顾性的：问题复发速度快于解决速度；回滚路径悄然退化；决策权转移到无法有效审计或约束的组件或行为者。

我们使用 FIT 框架来形式化这种失效模式，并定义一个实用的治理接口。核心主张不是对齐、鲁棒性或可解释性不重要，而是它们的有效性取决于保持随时间干预的能力。

贡献

- 我们将**不可逆操作（IO）**定义为与对齐和鲁棒性不同的失效模式：永久约束未来纠正选项的行为。
- 我们给出了可操作的 IO 分类法和一组可审计的指标来检测不可逆性风险。
- 我们提出**最小可行节奏治理（MVTG）**：五个最小控制措施，选择性地应用于 IO（慢速权威、回滚窗口、节奏分层、熔断机制、对抗性审计）。
- 我们提供抽象化的案例模式，展示 IO 如何产生，不涉及任何具体组织或系统名称。

2. 问题框架：不可逆性作为安全失效模式

2.1 可逆与不可逆故障

AI 系统中的许多故障是可逆的：一个糟糕的发布被回滚，数据集问题被修复，策略被更新，系统恢复到可接受的运行状态。相比之下，**不可逆故障**是指在可接受的边界内（时间、成本、法律风险、信任、治理能力）恢复不可行的情况。

本文将不可逆性视为改变系统可达未来的操作的属性。IO 可以是技术性的（例如消除回滚）、组织性的（例如移除审核门控）或制度性的（例如阻止撤回的承诺）。我们在第 4.1 节形式化 IO，并在第 4.3 节通过可测量指标将其可操作化。

2.2 为什么事后控制在加速下失效

事后控制假设纠正性反馈能够及时到达。在加速下，三种效应打破了这一假设：

1. **延迟主导：** 评估和审核所需时间超过重大更新之间的间隔，因此治理无法跟上。
2. **复合变化：** 每次更新都改变了验证先前修复的上下文，因此可靠性变得路径依赖。
3. **隐性能力侵蚀：** 回滚路径、可审计性和替代路线可能逐渐退化，直到恢复变得不可行。

结果是**虚假稳定**：系统看似表现良好且稳定，而干预的选项空间正在坍缩。这促使建立一个专注于**阈值预防**（不要越过不可逆性阈值）而非仅监控结果的治理接口。

2.3 不可逆性的抽象案例模式

本节呈现**抽象案例模式**，说明在加速部署和治理失配下 IO 如何在 AI 系统中产生。这些案例经过故意去标识化和泛化处理，以突出**结构动态而非偶然细节**。

案例模式 A：无治理同步的加速

初始条件：
一个 AI 系统部署时有定期评估和人工审核门控。更新周期与治理反馈保持一致。

干预：
为提高响应速度，部署频率逐渐增加。评估和审核流程保持不变。

观察到的动态：

- 更新速度超过评估完成时间。
- 事件在先前纠正得到验证之前再次发生。
- 治理反馈变成回顾性而非预防性。

不可逆操作：
当系统的最大更新速率在没有相应治理同步的情况下永久提高时，发生**节奏升级型 IO**。

结果：
系统进入**结构性加速**状态，回滚理论上可行但由于复合变化在实践中不可行。

关键洞察：
不可逆性不是来自任何单一更新，而是来自**跨越节奏阈值**——超过该阈值后治理无法重新获得控制。

案例模式 B：因操作便利导致的回滚侵蚀

初始条件：
系统维护版本化部署和经过测试的回滚程序。

干预：
为减少运营开销，回滚步骤被简化或推迟。备份频率降低。

观察到的动态：

- 回滚演练变得不频繁。
- 恢复时间目标在不知不觉中增加。
- 跨版本依赖累积。

不可逆操作：
当恢复路径名义上存在但在真实压力下失效时，发生**回滚移除型 IO**。

结果：
以前可恢复的故障现在传播为永久性系统状态变化。

关键洞察：
不可逆性可能通过便利驱动的侵蚀**逐渐且无意地产生**，而非明确决策。

案例模式 C：向不透明决策组件的控制转移

初始条件：
关键决策由具有人工覆盖的可解释流程中介。

干预：
引入不透明或不可审计的组件以提高效率或性能，而监督结构保持不变。

观察到的动态：

- 决策理由无法事后重建。
- 覆盖机制形式上存在但无法实时执行。
- 责任在组件之间扩散。

不可逆操作：
当有效控制转移到有意义的审计或干预之外的机制时，发生**控制转移型 IO**。

结果：
系统保留名义上的人类权威，但失去**操作可控性**。

关键洞察：
控制的丧失可能先于可见故障，造成**虚假的稳定感**。

案例模式 D：因标准化导致的选项空间坍缩

初始条件：
多条技术和治理路径共存，支持比较和回退。

干预：
为简化运营，替代方案被淘汰，单一路径被标准化。

观察到的动态：

- 对独特组件的依赖增加。
- 切换成本急剧上升。
- 异议或平行评估渠道消失。

不可逆操作：
当替代轨迹被消除时，发生**多样性坍缩型 IO**。

结果：
随着未来适应选项受限，系统韧性下降。

关键洞察：
效率驱动的整合可能无意中将系统锁定在脆弱的均衡中。

案例模式 E：社会技术整合中的治理滞后

初始条件：
一个 AI 系统在有限的技术领域内运行。

干预：
系统被整合到更广泛的组织或社会工作流程中，而没有相应的治理适应。

观察到的动态：

- 决策影响扩展速度快于监督范围。
- 反馈信号变得间接或延迟。
- 责任边界模糊。

不可逆操作：
当整合先于治理重新设计时，发生**轨迹锁定型 IO**。

结果：
即使技术回滚仍然可能，逆转整合在社会或制度上变得不可行。

关键洞察：
不可逆性可能源于**制度耦合**，而非技术故障。

跨案例综合

在这些模式中，不可逆性始终源于系统演化与治理反馈之间的**节奏失配**，而非孤立错误。IO 通常是渐进出现的，这使得早期检测至关重要。

这些模式促成了第 5 节中引入的治理机制，这些机制针对**阈值预防**而非结果优化。

2.4 威胁模型（草图）

本文聚焦于由以下原因产生的**非故意不可逆性**：

- 系统演化与治理反馈之间的节奏失配
- 便利驱动的回滚能力侵蚀
- 无明确决策的增量控制转移

我们不涉及**对抗性 IO 注入**（例如故意破坏回滚机制），尽管 MVTG 的对抗性审计组件提供了部分缓解。

3. FIT 框架（本文最小版本）

FIT（力-信息-时间）是描述系统如何跨基底演化的最小元语言。在本文中，我们将 FIT 用作部署安全的**诊断视角**：系统不仅可能因优化错误而失败，还可能因**演化速度快于治理纠正速度**而失败。

3.1 原语（迷你术语表）

符号	名称	定义（本文用）
F	力（Force）	状态变化的驱动因素（梯度、压力、激励）
I	信息（Information）	塑造未来演化的持久结构/信号
T	时间（Time）	更新节奏和不可逆性边界
C	约束（Constraint）	对可达状态空间的累积限制
S	状态（State）	给定时间的系统配置

3.2 本文的核心洞察

FIT 对 AI 安全的贡献不是新的目标函数，而是**元级诊断**：系统不仅可能因优化错误而失败，还可能因**演化速度快于治理纠正速度**而失败。IO 是这一问题变得可见和可治理的具体接口。

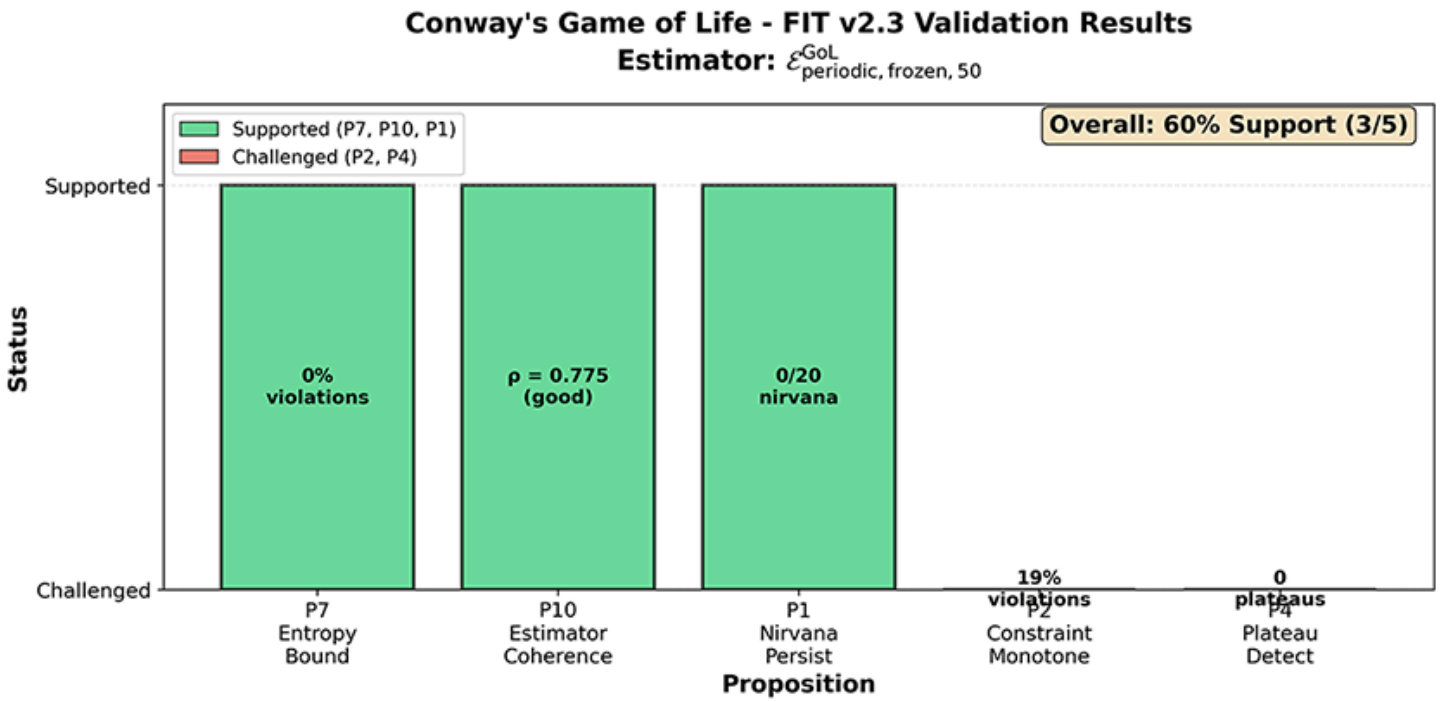

图 1：玩具验证（Conway 生命游戏）。这不是基准测试结果；它说明 IO 风格指标的可证伪性和估计量敏感性。

4. 不可逆操作（IO）

4.1 定义与范围

我们将**不可逆操作（IO）定义为永久约束 AI 系统未来可达状态空间**的系统级行为，使得回滚、恢复或替代轨迹在可接受的成本、时间或治理边界内变得不可行。

形式上，如果操作 o 在时间 t 执行后，以下至少一项成立，则认为该操作是**不可逆的**：

1. **状态空间坍缩**：可达未来状态集 $\mathcal{S}_{t+1}$ 是操作前可达集 $\mathcal{S}_t$ 的真子集，且没有可行路径恢复被排除的区域。
2. **回滚消除**：不存在可在有限损失（如数据、信任、法律风险）内恢复先前系统行为的实际回滚程序。
3. **轨迹锁定**：替代的技术或治理路径被消除或变得成本过高。
4. **控制不对称**：决策权或更新节奏转移到无法有效审计或约束的组件或行为者。

定义（不可逆操作）。
如果在时间 t 执行的操作 o 满足以下条件，则该操作是不可逆的：

$$\exists\, s^* \in \mathcal{S}_t \setminus \mathcal{S}_{t+1} \quad \text{使得} \quad \inf_{\pi} \mathrm{Cost}(\pi : S_{t+1} \to s^*) > \theta_{\text{feasible}}$$

其中 $\mathcal{S}_t$ 是操作 o 之前的可达状态空间，$\mathcal{S}_{t+1}$ 是操作 o 之后的可达状态空间，π 遍历恢复计划，θ_{feasible} 是特定领域的可行性边界（时间、金钱、法律风险、信任、治理能力）。

该定义有意抽象于模型内部，聚焦于**系统后果**，使得 IO 能够在技术、组织和政策领域被识别。

4.2 不可逆操作分类

我们将 IO 分为四个非互斥类别，每个对应不同的不可逆模式。

4.2.1 节奏升级型 IO

永久提高系统最大更新或部署速率的操作。

示例（抽象化）：

- 移除强制性评估或审核门控。
- 自动化部署流水线而无人在环检查点。
- 对高影响组件从分阶段发布转为持续发布。

风险机制：
治理反馈无法跟上系统演化，导致失控漂移。

4.2.2 回滚移除型 IO

消除或严重退化回滚系统状态能力的操作。

示例：

- 无版本化快照或恢复程序的部署。
- 无备份的不可逆数据转换。
- 阻止系统撤回的法律或合同承诺。

风险机制：
错误从可恢复事件转变为永久性结构变化。

4.2.3 控制转移型 IO

将**有效控制权转移**到现有监督结构之外的组件、组织或流程的操作。

示例：

- 将关键决策委托给不透明或不可审计的模型。
- 在无执行保证的情况下外包治理关键功能。
- 嵌入无法暂停或约束的第三方系统。

风险机制：
尽管名义权威仍存在，但失去有意义的人类或机构控制。

4.2.4 多样性坍缩型 IO

消除替代技术或治理路径的操作。

示例：

- 在无应急方案的情况下标准化单一模型架构或供应商。
- 移除平行评估或异议渠道。
- 锁定单一监管解释的政策决定。

风险机制：
随着选项空间坍缩，系统韧性下降。

4.3 不可逆性风险的可观测指标

为使 IO 检测可操作化，我们定义了表明不可逆性风险上升的**可观测指标**。这些指标是领域无关且可审计的。

4.3.1 节奏失配指标

- **决策到验证延迟**：从部署到完成有意义评估的时间。
- **更新速度比**：系统更新速率相对于治理或审核周期速率。
- **加速下的事件复发**：纠正措施速度跟不上的重复故障。

4.3.2 回滚可行性指标

- **回滚成功率**：完全恢复先前状态的尝试回滚比例。
- **恢复时间目标（RTO）** 趋势。
- **回滚演练频率**：缺失表明潜在不可逆性。

4.3.3 控制与可解释性指标

- **不透明决策比例**：由不可解释组件中介的关键决策份额。
- **审计延迟**：事后重建决策理由所需时间。
- **覆盖可行性**：实时停止或修改行为的能力。

4.3.4 多样性与选择性指标

- **单点依赖指数**：对独特组件或行为者的依赖程度。
- **路径切换成本**：采用替代方法的估计成本和时间。
- **冗余侵蚀率**：并行系统被淘汰的速率。

4.3.5 阈值指导（暂定）

指标	黄区	红区
更新速度比	> 3 倍治理周期	> 10 倍
回滚成功率	< 90%	< 50%
不透明决策比例	> 20% 关键决策	> 50%
单点依赖	> 60% 依赖度	> 90%

注：阈值仅为示例，应根据领域和风险容忍度进行校准。

4.4 IO 作为与对齐错误不同的失效模式

FIT 的关键贡献是将 IO 与传统 AI 安全失效模式区分开来。

- **对齐失败**关注系统*优化什么*。
- **鲁棒性失败**关注系统*在扰动下如何表现*。
- **IO 失败**关注*未来纠正是否仍然可能*。

一个对齐且鲁棒的系统如果过早执行 IO，仍可能变得不安全，将系统锁定在不可控轨迹中。

4.5 设计启示

认识到 IO 将 AI 安全从被动缓解重构为**预防性阈值管理**：

- 并非所有行动都需要慢速治理——**只有那些跨越不可逆性阈值的行动**。
- 安全干预应优先考虑**节奏控制和回滚保持**，而非穷尽预见。
- 治理成功以**维持的选项空间**衡量，而非以事件缺失衡量。

这一视角促成了第 5 节中引入的治理机制。

5. 最小可行节奏治理（MVTG）

5.1 理念：治理阈值，而非结果

AI 安全干预通常试图治理结果（如对齐目标、性能边界）。然而，当存在**不可逆操作（IO）时，聚焦结果的控制来得太晚。MVTG 将治理重构为围绕阈值管理**：防止不可逆约束未来选项的行动。

MVTG 有意保持**最小化**。它不寻求对所有 AI 活动的全面监督。相反，它只针对**那些跨越不可逆性阈值的行动**，从而在最小化运营负担的同时最大化安全影响。

5.2 设计原则

MVTG 由三个原则指导：

1. **放慢不可逆的**
   被识别为 IO 的行动必须接受审慎的、低节奏决策过程。
2. **保持回滚**
   所有高影响行动必须维护可验证的恢复或撤回路径。

3. **节奏分层**
   系统可能在边缘快速演化，但核心约束和治理层必须缓慢演化。

这些原则使安全努力与风险集中度保持一致。

5.3 核心组件

5.3.1 慢速权威

慢速权威是专门应用于 IO 的治理约束。

定义：
要求 IO 通过具有强制延迟、双重授权和可审计理由的决策流程。

关键属性：

- 仅适用于 IO 分类的行动。
- 强制执行**冷却期**以允许反事实评估。
- 要求记录**明确理由**以供事后审计。

安全功能：
防止绕过治理反馈循环的冲动加速。

5.3.2 回滚窗口

回滚窗口是一个有限时期，在此期间触发 IO 的变更可以完全逆转。

要求：

- 执行前的版本化系统快照。
- 具有定义恢复目标的经测试回滚程序。
- 定期回滚演练以验证可行性。

安全功能：
通过保持纠正能力，将潜在不可逆性转化为可管理风险。

5.3.3 节奏分层

MVTG 将系统演化分为**节奏层**，每层以不同速度进行治理。

规范分层：

- **探索层（快速）：** 实验、A/B 测试、局部优化。
- **验证层（中速）：** 评估、审计、交叉检查。
- **约束层（慢速）：** 政策、不可逆部署、核心治理。

与估计量选择理论（EST）的联系：
节奏分层也自然映射到 EST 的任务类型等价概念：

- 快速层 -> 序数等价（趋势检测）
- 中速层 -> 度量等价（阈值/尺度稳定性）
- 慢速层 -> 拓扑等价（体制结构）

这种对齐有助于保持治理指标在 EST 标准下的**可容许性**和可审计性。

示例： 你的推理管道可以每小时更新，但安全过滤器的更改需要每周审核。

安全功能：
防止快速移动的组件直接修改慢速的、基础性的约束。

5.3.4 熔断机制

熔断机制是预定义的机制，当不可逆性风险超过可容忍阈值时，停止或降级系统运行。

触发条件可能包括：

- IO 指标的快速累积。
- 重复回滚失败。
- 治理反馈延迟超过预设边界。

安全功能：
提供最后手段控制以防止级联不可逆转变。

5.3.5 对抗性审计

MVTG 纳入**对抗性审计**以持续测试治理完整性。

范围：

- 挑战 IO 分类决定。
- 探测绕过慢速权威的路径。
- 压力测试回滚和熔断机制的有效性。

安全功能：
维护治理韧性以对抗自满和内部偏见。

5.4 MVTG 到 IO 类别的映射

每个 MVTG 组件直接缓解一个或多个 IO 类别：

IO 类别	主要 MVTG 控制
节奏升级型 IO	慢速权威、节奏分层
回滚移除型 IO	回滚窗口
控制转移型 IO	对抗性审计、熔断机制
多样性坍缩型 IO	节奏分层、审计

此映射强调**选择性控制**，而非全面限制。

5.5 运营指标

MVTG 有效性使用**过程指标**评估，而非结果保证：

- IO 审批延迟（应为非零且稳定）。
- 回滚成功率和演练频率。
- 快速层更新与慢速层变更的比率。
- 熔断机制激活的频率和解决时间。
- 通过审计检测到的治理绕过数量。

这些指标是**可观测的、可审计的、组织无关的**。

5.6 MVTG 的局限性（范围）

MVTG 不：

- 优化模型目标。
- 预测特定故障事件。
- 消除对对齐或鲁棒性研究的需求。

相反，它**保持干预能力**，在不可逆伤害发生之前。

6. 讨论

6.1 相关工作与定位

FIT 框架是对现有 AI 安全方法的**补充**，而非替代。其贡献在于识别一种**独特的失效模式**：节奏失配下的不可逆性。

- **对齐与基于目标的安全：** 对齐聚焦于确保系统优化预期目标。然而，即使是对齐良好的系统，如果 IO 永久约束纠正选项，也可能变得不安全。FIT 关注*何时*干预仍然可能，而非*优化什么*目标。
- **鲁棒性与可靠性：** 鲁棒性缓解分布偏移或对抗性输入下的性能退化。FIT 针对**结构性风险**，即由于恢复路径被消除而导致故障持续存在。
- **可解释性与透明度：** 可解释性提高理解，但如果决策权和节奏转移到实践中无法暂停或覆盖的组件，可能不足够。
- **治理与政策框架：** 治理提案强调合规、问责和风险分类。FIT 贡献了**时间维度**，为必须因不可逆性风险而区别治理的行动提供操作标准。

代表性锚点：

- 对齐：(Hadfield-Menell et al., 2016; Christiano et al., 2017)
- 鲁棒性：(Goodfellow et al., 2014; Hendrycks & Dietterich, 2019)
- 可解释性：(Doshi-Velez & Kim, 2017; Lipton, 2018)
- 治理：(Brundage et al., 2020; Anderljung et al., 2023)

6.2 为什么节奏作为一级安全变量很重要

大多数 AI 安全框架隐含假设纠正性反馈可以及时应用。FIT 通过强调**反馈延迟相对于系统更新节奏**决定安全干预是否有效来挑战这一假设。

当系统演化超过治理反馈速度时：

- 纠正行动变成回顾性的。
- 错误累积速度快于解决速度。
- 安全从预防转向损害控制。

通过将节奏提升为一级变量，FIT 将安全从优化稳态行为重构为**维持随时间的干预可行性**。

6.3 虚假稳定与控制幻觉

FIT 突显了**虚假稳定**：系统看似表现良好且稳定，同时逐渐丧失可控性的状态。

虚假稳定与传统失效模式不同：

- 没有观察到即时性能退化。
- 事件被吸收或正常化，而非解决。
- 治理结构形式上保持完整，但失去实际效力。

FIT 将虚假稳定解释为由 IO 驱动的**选项空间坍缩**的后果。这一视角有助于解释为什么某些系统在长期表观成功后会灾难性地失败。

6.4 对 AI 部署实践的启示

本文建议一种实践转变：通过选择性控制将不可逆性作为可治理变量，而非试图平等地治理所有行动。

AI 实验室/组织的最小采用路径

1. **创建 IO 登记表：** 定义在你的上下文中什么算作 IO；要求对部署节奏、回滚机制、决策权和依赖关系的变更进行 IO 标记。
2. **为 IO 安装慢速权威门控：** 对 IO 分类的变更强制执行延迟 + 双重授权 + 可审计理由。
3. **要求回滚窗口和演练：** 将回滚视为定期测试的能力，而非文档制品。
4. **执行节奏分层：** 防止快速层更新在没有慢速权威的情况下修改慢速约束（政策、访问权限、不可逆发布）。
5. **定义熔断触发条件：** 将 IO 指标与自动降级或暂停关联；预先注册触发条件以避免临时治理。
6. **运行对抗性审计：** 持续探测绕过路径和失效模式，包括治理漏洞和紧急捷径。

要测量什么（过程指标）

- 更新速度比 vs 评估周期时间
- 回滚成功率和演练频率
- IO 审批延迟和绕过计数
- 治理关键指标的变点频率（审计日志、政策例外）

6.5 局限性

1. **尚无定量基准。** IO 指标已提出但未经大规模实证验证。
2. **需要领域校准。** 节奏失配和不可逆性的阈值是上下文相关的。
3. **不是完整的安全解决方案。** FIT 关注*何时*干预可能，而非*对齐什么*目标。
4. **治理实施成本。** MVTG 引入开销；此处未分析成本效益权衡。
5. **尚无对抗性威胁模型。** 当前框架强调非故意 IO 累积，而非故意规避。

7. 结论

本文论证了越来越多的 AI 安全风险不是来自孤立的模型故障，而是来自由加速部署和治理失配驱动的**不可逆系统级转变**。我们引入了力-信息-时间（FIT）框架，将**节奏**——系统更新的速率和顺序——提升为一级安全变量，并将**不可逆操作（IO）**形式化为永久约束未来纠正选项的行动。

通过识别节奏失配和不可逆性风险的可观测指标，我们表明许多安全故障在传统对齐或鲁棒性干预可应用**之前**就已出现。为填补这一空白，我们提出了**最小可行节奏治理（MVTG）**，一个轻量级且可审计的控制层，只选择性地治理那些跨越不可逆性阈值的行动。

预防不可逆伤害需要将注意力从优化结果转向**维持选项空间**。将节奏作为可治理变量提供了实现这一目标的实用路径。

附录 A：EST 审计制品（可选）

如果你运行 IO 指标的实证研究，请预先注册估计量选择和阈值（以避免指标操纵）并报告一致性结果：

- 预注册模板：est_preregistration_template.yaml
- 等价性 + 一致性报告：est_equivalence_and_coherence_report.md

参考文献

- Amodei, D. et al. (2016). *Concrete Problems in AI Safety.* arXiv:1606.06565.
- Anderljung, M. et al. (2023). *Frontier AI regulation: Managing emerging risks.* Centre for the Governance of AI (GovAI). (Policy report.)
- Brundage, M. et al. (2020). *Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims.* arXiv:2004.07213.
- Christiano, P. et al. (2017). *Deep Reinforcement Learning from Human Preferences.* arXiv:1706.03741.
- Doshi-Velez, F., & Kim, B. (2017). *Towards A Rigorous Science of Interpretable Machine Learning.* arXiv:1702.08608.
- Goodfellow, I., Shlens, J., & Szegedy, C. (2014). *Explaining and Harnessing Adversarial Examples.* arXiv:1412.6572.
- Hadfield-Menell, D. et al. (2016). *Cooperative Inverse Reinforcement Learning.* arXiv:1606.03137.
- Hendrycks, D., & Dietterich, T. (2019). *Benchmarking Neural Network Robustness to Common Corruptions and Perturbations.* arXiv:1903.12261.
- Lipton, Z. C. (2018). *The Mythos of Model Interpretability.* Communications of the ACM.