

面向大语言模型交互的 AI 角色外部治理框架

摘要:

[目的]针对 AI 角色在大语言模型交互中因模型概率性输出导致的行为不确定性与人类对角色确定性需求之间的结构性矛盾,本文提出“角色基座”概念框架——一个独立于底层模型的结构化外部治理中间件。该框架不干预大模型的内部参数、推理过程或采样策略,仅通过封装角色身份、行为边界与治理逻辑,将 AI 角色的概率性输出转化为可验证、可审计的确定性行为,从而实现从“对话”到“契约”的范式转变。
[方法]本研究遵循设计科学范式,构建了“角色基座(Role Base,RB)”框架:将角色身份、行为边界与治理逻辑封装为可移植、可审计的标准化规范件。核心机制包括:多层语义架构(关注点分离、层按需装配);三域模型(定义与部署解耦);双轨治理(完整性摘要+批准链,密码学级可审计);外部约束执行架构(编译型提示词+输出安全过滤+工程限流);基于输出文本的治理漂移量化(语义漂移指数+规则违反率双指标)。

[结果]交付了完整的角色基座设计人工制品,包括理论框架、技术规范(十层 Schema、错误码、完整性摘要规范化流程、测试向量、参考实现)以及治理生命周期模型。从架构层面提出了通过外部引导和输出验证生产行为确定性的设计方法,理论上支持跨平台可移植、密码学完整性、责任可追溯、黑盒兼容及长期漂移监控。

[局限]本文为设计科学论文,所提框架的有效性目前仅通过逻辑论证和对比分析进行了初步评估,所有关于确定性水平、漂移阈值、拦截率的主张均为设计目标或理论承诺,尚待原型实现与实证检验。确定性仅覆盖预定义规则集内行为,开放域不作保证。确定性为工程级统计一致性,非数学绝对保证。规则防火墙对已知违规模式的拦截率 100%为理论目标,实际部署受规则库完整性和编译器正确性制约。

[结论]AI 角色的行为确定性应通过架构层的关注点分离、编译时校验、完整性签名与运行时外部约束来系统性生产。角色基座将人机交互中的角色期待显式化为可验证的治理规则,实现从“对话”到“契约”的范式转变。本文提供的设计框架为后续工程实现与实证评估奠定了基础。

关键词: 角色基座; 外部治理; 行为确定性; 治理漂移; 可审计性; 大语言模型

分类号: TP18; TP311; TP309

An External Governance Framework for AI Roles in Large Language Model Interaction

Abstract:

[Objective] Aiming at the structural contradiction between the behavioral uncertainty of AI roles, caused by the probabilistic outputs of large language models (LLMs), and the human demand for role determinacy, this paper proposes the conceptual framework of a “Role Base”——a structured external governance middleware independent of the underlying model. It does not interfere with the internal parameters, inference process, or sampling strategy of LLMs. Instead, it encapsulates role identity, behavioral boundaries, and governance logic to transform probabilistic outputs into verifiable, auditable deterministic behaviors, realizing a paradigm shift from “conversation” to “contract.”

[Methods] Following the design science paradigm, this paper constructs the “Role Base (RB)” framework, which encapsulates role identity, behavioral boundaries, and governance logic into portable, auditable standardized specifications. Core mechanisms include: a multi-layer semantic architecture (separation of concerns, on-demand layer assembly); a three-domain model (decoupling definition from deployment); dual-track governance (integrity digest + approval chain, cryptographically auditable); an external constraint execution architecture (compiled prompting + output safety filtering + engineering rate limiting); and an output-text-based governance drift quantification (semantic drift index + rule violation rate).

[Results] A complete design artifact of the Role Base is delivered, including the theoretical framework, technical specifications (ten-layer schema, error code system, integrity digest normalization procedure, test vectors, reference implementation) and a governance lifecycle model. An architectural design methodology for producing verifiable behavioral determinacy through external guidance and output validation is presented, theoretically supporting cross-platform portability, cryptographic integrity, traceable accountability, black-box compatibility, and long-term drift monitoring.

[Limitations] As a design science paper, the validity of the proposed framework has only been preliminarily assessed through logical argumentation and comparative analysis. All claims regarding determinacy levels, drift thresholds, and interception rates are design goals or theoretical commitments, awaiting prototype implementation and empirical validation. Determinacy is guaranteed only for behaviors within the predefined rule set, with no assurance for open domains. Determinacy is engineering-level statistical

consistency, not mathematical certainty. The claimed 100% interception rate for known violation patterns is a theoretical target; actual deployment is constrained by rule base completeness and compiler correctness.

[Conclusion] The behavioral determinacy of AI roles should be systematically produced through architectural separation of concerns, compile-time verification, integrity signatures, and runtime external constraints. The Role Base makes role expectations in human-AI interaction explicit as verifiable governance rules, realizing a paradigm shift from “conversation” to “contract.” The design framework provided in this paper lays the foundation for subsequent engineering implementation and empirical evaluation.

Keywords: Role Base; external governance; behavioral determinacy; governance drift; auditability; large language model

Classification codes: TP18; TP311; TP309

1 引言：算法的“丰饶”与交互的“失序”

大语言模型（Large Language Models, LLMs）的核心特征在于其生成能力的无限性。基座模型在海量语料中习得了跨越不同领域、不同风格、不同立场的表达能力，能够在单一对话窗口内自如切换于严谨的学术顾问与浪漫的文学伴侣之间。然而，在严肃的商业与协作场景中，这种无限的可能性往往意味着失控的风险。当模型的行为边界仅取决于模糊的自然语言暗示时，人机协作便陷入了一种“治理失序”——人类无法以工程化的确定性来预期算法的输出。

1.1 国内外相关研究现状与不足

当前，针对大语言模型中 AI 角色行为治理的研究可归纳为三条主要进路，但它们各自存在结构性局限，难以系统性地解决上述“治理失序”问题。

(1) 第一条进路聚焦于规则约束层的构建。

在国际上，Anthropic 提出的 Constitutional AI (CAI) 框架是该方向的代表性工作，其核心思想是用一套预设的“宪法”原则来指导模型的自我对齐[1]。然而，He et al. (2025) 针对该框架的实证研究揭示了其核心脆弱性：用于指导模型对齐的自然语言原则存在显著的解释性歧义，不同模型对大量规则的理解存在严重分歧[19]，导致行为不一致。OpenAI 推出的 Model Specs 也存在类似问题——本质上仍是一套自然语言行为规范，缺乏可强制执行的技术机制。在国内，围绕生成式 AI 中 AI 角色的治理研究已从早期的宏观原则呼吁转向治理架构的系统设计[2]。然而，上述工作在“确定性”的实现层面仍停留在政策呼吁与原则声明阶段，尚未形成可工程化落地的技术架构。NVIDIA 开源的 NeMo Guardrails 等护栏平台标志着这一方向向工程化的重要迈进[8]，但其防护范围主要集中于内容安全违规、提示注入和数据泄露三大风险类别，在角色一致性和长期行为稳定性的专门化治理方面仍存在明显空白。

(2) 第二条进路围绕角色扮演与行为一致性展开。

肖仰华团队在 TMLR 发表的综述系统提出了角色扮演语言智能体的三层人格框架，关注角色扮演的内容保真度与人格一致性[5]。Peng 等人提出的 Codified Profiles 方法将角色逻辑表示为结构化的可执行函数，在持久性和可更新性方面展现出优势[12]。在国内，角色扮演大语言模型的研究主要集中于基于高质量对话数据的模型微调与基于向量检索的记忆增强两条技术路线。然而，这些工作共同关注的是“这个角色应该说什么”，旨在提升角色的表现力与仿真度，而非“这个角色绝不能说什么、做什么”，缺乏对行为边界与治理可归责性的系统性考量。已有实证研究明确指出：现人工创建的角色档案因信息片面化，往往无法与目标个体达成充分的行为对齐，静态定义需与动态优化机制衔接[10]，该文展示了生成式智能体长期行为与初始角色设定之间的漂移现象。

(3) 第三条进路探索约束解码与结构化生成技术，并揭示了提示工程的固有局限。

以 Outlines[15]、LMQL[4]、Guidance[20]等为代表的开源项目，通过词汇掩码、正则表达式和上下文无关文法在模型推理过程中施加硬约束，确保输出格式的语法正确性。这些技术为“硬约束”路径提供了关键工具，但其关注点主要限于输出的语法格式，而非角色的语义行为边界——即“输出格式正确”不等于“输出行为恰当”。更重要的是，Tian 等人发布的提示缺陷分类学首次系统性地归纳了提示失效的六类缺陷维度[11]，Labroo 等人的预印本研究进一步证明提示工程在多概念复合控制场景中的性能会显著退化[9]。这些发现共同指向一个结论：提示工程本质上是推理流程内的“请求”，其有效性完全依赖模型对自然语言指令的瞬时理解，缺乏架构层的硬性保障，在复杂场景下的稳定性与可审计性存在天然上限。

1.2 研究空白与本文定位

综合上述三条进路的分析，可以发现当前研究存在一个关键的结构性空白：在提示工程的“软请求”与模型微调的“硬绑定”之间，缺少一个独立于模型演进之外的、可移植的确定性治理中间件。

现有的规则约束层（如 CAI）依赖自然语言原则且缺乏强制执行机制；现有的 AI 角色一致性研究侧重表现力优化而非边界约束；现有的提示工程研究揭示了问题但未提供架构层的解决方案。三者均未能系统性回答一个核心问题：如何在算法的概率性本质与人类对确定性的需求之间，构建一个具备工程化可靠性的行为边界执行层？

本文提出的“角色基座”（Role Base, RB）框架，正是填补这一结构性空白的尝试。区别于将行为控制内嵌于提示词（软约束）或固化于模型权重（微调）的传统路径，“角色基座”主张在模型推理层之上构建一个独立、可移植的治理中间件。该框架在技术实现层面高度依赖已有的提示工程和输出过滤技术，因此其独特贡献不在于发明新的约束解码算法，而在于以下几点不可替代的增量价值：

- (1) **三层封装**：将“格式约束”（语法层）、“语义边界”（内容层）与“角色身份”（社会层）整合为统一的结构化 Schema。
- (2) **治理功能内置**：在架构中预制版本管理、完整性摘要、可审计性等治理功能，而非事后附加。
- (3) **关注点分离的标准化接口**：提供独立于具体模型的角色描述语言，使角色资产可在不同平台间迁移。与 OpenAI GPTs 等平台绑定的角色方案不同，角色基座强调规范件的跨平台可移植性。
- (4) **外部约束执行架构**：通过编译型提示词、输出安全过滤、工程限流三层机制实现不干预模型的确定性保障。
- (5) **术语辨析**：本文使用的“硬约束”指通过输出后的正则匹配、关键词阻断、合规过滤器所形成的架构层强制性——其“硬”不在于修改模型内部，而在于其执行逻辑不依赖于模型的概率性理解，且阻断行为独立于模型响应。

需特别指出，本文所述的“不干预模型”，其严格定义是不修改模型的内部参数、权重或采样策略。编译型提示词作为模型的输入，客观上构成了对模型推理过程的强力引导。这是一种“外部引导式干预”，其有效性依赖于模型对指令的遵循能力，而非对模型内部的直接操控。这是本框架选择“黑盒兼容”设计哲学时所接受的固有依赖。

1.3 研究设计：设计科学范式

本文遵循设计科学研究（Design Science Research, DSR）范式[17]。DSR 的核心产出是

“人工制品”（artifact）——一个为解决现实问题而构建的创新方案。本文的人工制品是“角色基座”框架，包括其理论模型、技术规范与治理流程。

根据 Hevner 等人提出的 DSR 框架，本文完成了以下活动：

(1) **问题识别与动机：** § 1.1-1.2 阐明 LLM 角色治理的实践缺口。

(2) **目标定义：** 提出“确定性生产”“跨平台可移植”“可审计性”等具体目标。

(3) **人工制品设计与开发：** § 3 与 § 4 交付十层语义架构、三域模型、双轨治理外部约束执行架构及漂移量化方法。

(4) **展示：** 通过完整 JSON 示例、错误码体系、生命周期状态机、参考实现代码展示人工制品的可操作性。

(5) **初步评估：** § 2.2 与已有技术对比分析，§ 4.3-4.4 论证边界性确定性与漂移检测的逻辑合理性，§ 5.2 与相关研究对话。

(6) **沟通：** 本文即研究成果的完整呈现。

本文属于设计科学中的“架构提案”子类——重点在于提出新颖且有理有据的架构设计，而非提供完整的实证评估。设计科学论文允许在缺乏大规模实证的情况下，通过逻辑论证、示例和对比分析来论证人工制品的效用与可行性[18]。因此，本文的定位不是概念/理论论文，而是设计科学论文。

1.4 核心主张与原创理论贡献

1.4.1 本文的核心主张可概括为以下三点：

(1) **身份与行为解耦：** 角色的身份定义、行为边界与治理逻辑应从模型推理过程中分离出来，形成独立的、可审计的标准化构件。

(2) **治理意图与执行机制分离：** 角色的“应然”规范通过结构化 Schema 表达，而具体执行通过编译型提示词、输出过滤、工程限流等多种外部路径实现，不干预模型内部。

(3) **确定性通过外部架构产生而非模型保证：** 角色基座不依赖于特定模型的“理解”或“对齐”，而是通过架构层面的编译时校验、完整性摘要、批准链签名与运行时输出验证，设计出可验证的行为确定性生产机制。

1.4.2 关于“确定性”的操作定义（设计目标）：

本文所称“确定性”，指在角色基座预定义规则集 R 所覆盖的行为维度集合 D 上，通过外部约束架构所追求的一致性。其操作化定义为：在真实交互中，对于给定输入 x ，经过角色基座输出验证与过滤后，最终返回给用户的响应 y 满足预定义约束 C 的统计比例 $P_{\text{obs}}(y \models C \mid x) \geq 1 - \varepsilon$ ，其中 ε 为可配置的容忍度参数（本文建议 $\varepsilon = 0.2$ ，对应合规阈值 0.8）。该定义为统计性而非数学绝对性，置信水平需通过实证测试确定。对于 D 之外的开放域行为，基座不提供确定性承诺。

1.4.3 本文的原创理论贡献包括：

(1) **多层语义架构：** 将角色身份、职能、人格、静态知识、行为策略、安全约束、治理规则、工具调用、协作协议与记忆契约解耦为多个可独立治理的语义层；采用“关注点分离”（软件工程经典原则）使每一层的治理关切互不干扰，采用“层按需存在”（设计决策）允

许仅身份层和职能层为强制层、其余层可按需选用或由系统注入安全默认值。

(2) **三域模型（语义内核域、能力声明域、运行时装配域）**：将角色定义彻底拆分为三部分，实现角色规范件跨模型、跨平台的强可移植性。

(3) **双轨治理机制（完整性摘要+批准链）**：提供密码学级的防篡改与可审计性。

(4) **外部约束执行架构（编译- 过滤- 限制三层）**：实现治理意图与执行机制的解耦，完全不依赖模型内部状态。该架构提供“工程级确定性”，其效力依赖于编译器、规则库和工程逻辑的正确性，通过输出后阻断而非 Token 级掩码实现约束。

(5) **治理漂移的概念化与量化方法**：首次将长期交互中模型行为逐步偏离初始角色定义的现象概念化为“治理漂移”，并提出基于输出文本的语义漂移指数（SDI）与规则违反率（RVR）双指标量化框架，不读取模型内部概率分布。阈值需运行时动态优化，本文提供的参考值（SDI 预警 0.1/熔断 0.25，RVR 预警 0.05/熔断 0.15）仅为概念演示，实际部署必须根据具体模型和任务校准。

1.5 研究问题与论文结构

本文试图回答以下研究问题：

(1) **RQ1**：如何针对 AI 角色，在大语言模型的概率性本质与人类对角色行为确定性的需求之间，构建一个工程化的行为边界执行层（不干预模型内部）？

(2) **RQ2**：如何将社会角色理论、计算法律学与外部约束技术整合为统一的治理框架？

(3) **RQ3**：如何量化并监控长期交互中的“治理漂移”现象（仅基于可观测的输入输出）？

本文其余部分组织如下：第 2 节界定概念并阐明理论根基。第 3 节阐述角色基座的概念架构，包括三域模型、十层语义架构与双轨治理机制。第 4 节论述确定性生产机制，涵盖外部约束执行架构、治理漂移量化方法（基于输出文本）以及治理发布流程。第 5 节讨论设计贡献、伦理自省与局限性。第 6 节总结全文。

2 概念界定与理论根基

2.1 何为“角色基座”

在学术话语中，“基座”一词借鉴自基础设施研究中的“平台底座”[7]与计算法学中的“架构即规制”理论[3]。

定义：角色基座是一个通过结构化规则集（Schema）对 AI 角色行为进行预先建模与边界固化的治理中间件。它存在两种互补形态：

(1) **设计时**：是可编辑、可审计、可版本管理的结构化角色定义文件；

(2) **运行时**：通过适配编译器转化为针对目标模型的结构化提示词前缀，并结合输出后过滤与工程限流，在模型推理链路外部执行行为边界。其核心功能不在于增强模型的“能力上限”，而在于明确 AI 角色的“行为下限”。本框架不干预大模型的内部参数、推理过程、采样策略或概率分布。

2.2 与已有技术的边界澄清

为避免“概念再包装”的质疑，本节明确界定“角色基座”与相近技术概念的本质区别

与互补关系。

表 1：角色基座与相近技术概念对比

概念对比	角色基座	约束解码	系统提示词	模型微调	护栏工具（如 NeMo Guardrails）	平台角色方案（如 OpenAI GPTs）
核心关注	角色身份+语义边界+治理审计	输出格式语法正确性	模型初始化指令	权重偏向性调整	内容安全与注入防护	角色定义+平台内集成
约束方式	结构化 Schema+编译提示词+输出过滤	词汇掩码/正则/CFG	自然语言前缀	统计学概率偏移	规则+模型检测	自然语言指令+内部工具
是否干预模型内部	否	是（修改 Logits）	否	是	部分（有时修改输出）	否
跨平台可移植性	强（平台无关标准化文件）	中（依赖 logits 接口）	弱（与 API 绑定）	极弱（参数绑定）	中	极弱（平台绑定）
密码学审计	是（摘要+批准链）	否	否	否	否	否
治理漂移监控	是（SDI+RVR 双指标）	否	否	否	否	否
确定性来源	架构逻辑+规则覆盖+输出阻断	语法规则	模型瞬时理解	训练数据分布	规则匹配	平台内部机制
资产流动性	强	中	弱	极弱	中	极弱

关键辨析：

- （1）与约束解码：角色基座不使用约束解码（因其修改模型采样过程），而是采用编译提示词+输出过滤。两者目标不同：约束解码追求输出格式的语法正确性，角色基座追求角色身份与语义边界的合规性，且不干预模型。
- （2）与系统提示词：角色基座将“软请求”升级为“结构化约束+输出验证”。系统提示词可被理解为角色基座的一种“自然语言前端”，但角色基座通过结构化 Schema 将关键约

束从提示词中提取出来，并通过输出过滤提供强制保障。

(3) **与模型微调：**两者互补且不冲突——微调改变“能力倾向”，角色基座在推理时施加“行为边界”。角色基座不取代微调，而是在微调后的模型之上增加可审计的治理层。

(4) **与护栏工具：**角色基座可视为护栏工具在角色治理领域的专门化扩展，增加了角色身份管理、跨平台可移植性、完整性摘要、批准链治理等特性。

(5) **与平台角色方案：**OpenAI GPTs 等产品提供了平台内角色定义，但绑定特定平台，缺少跨平台可移植性、密码学审计和治理漂移监控能力。角色基座强调规范件的平台无关性和完整治理审计链，可与 GPTs 等方案互补使用。

表 2：与主流提示工程/输出过滤工具的功能对比

功能维度	传统提示工程	输出过滤库（如 Guardrails）	本文“角色基座”
结构化角色定义	✗ 依赖自然语言	部分支持	✓ 十层 Schema
输出规则过滤	需自行实现	✓ 支持	✓ 集成
治理漂移量化	✗ 无	✗ 无	✓ 语义+规则双指标
角色资产可移植性	✗ 与提示词绑定	✗ 与代码绑定	✓ 标准化 JSON
密码学完整性审计	✗ 无	✗ 无	✓ 摘要+批准链
跨轮次记忆契约	✗ 无	有限	✓ L10 记忆契约层

差异化总结：传统提示工程和输出过滤库聚焦于单次交互的安全性，本文框架在此基础上叠加了角色身份的生命周期管理、跨平台可移植性、密码学审计和治理漂移监控，且完全不干预模型内部。

2.3 理论根基

(1) 社会角色理论的启发

社会角色理论（Role Theory）指出，社会互动中的确定性并非源于个体的“本质”，而是源于社会对特定角色的规范性期待[6]。当一个人扮演“医生”“教师”或“法官”角色时，其行为模式在很大程度上是由该角色的社会规范所预设的。这一洞察为 AI 角色的治理提供了理论基础：如果我们能够为一个 AI 角色明确定义其“角色规范”，并用技术手段确保该规范在推理过程中被遵守（通过外部约束），那么我们就能够在概率性算法输出之上，生产出符合人类期待的行为确定性。

社会学视角下，社会互动的高效性源于清晰的社会角色边界，角色定义权利、义务与行为脚本，降低交互不确定性。大模型无固定角色会引发期望错位——用户严谨协作需求与模型随机泛化输出形成矛盾。角色基座通过身份共识与行为边界约束，将算法从无边界生成体转化为履约式智能体，让松散对话转化为标准化社交契约。

(2) 计算法学视角：架构即规制

计算法律学（Computational Law）研究如何将法律规则形式化表达，并通过计算系统自动执行。Lessig 提出的“代码即法律”（Code is Law）命题[3]，在 LLM 时代获得了新的内涵。角色基座借鉴计算法律学的思路，将角色的安全约束、合规边界与行为策略形式化为结构化的 JSON Schema，并通过完整性摘要与批准链签名实现可审计的证据链。这使得角色的“合规状态”不再仅仅依赖于事后的人工审查，而是可以在编译时、运行时和审计时被持续验证。

本文对计算法律学的推进在于：不仅将单条规则形式化，更将规则的“制定——审核——批准——发布——执行——审计”全生命周期纳入统一的技术框架，实现了从“规则自动化”到“治理自动化”的跃迁。

(3) 工程学视角：关注点分离与可移植性

依托软件工程关注点分离原则，拆分模型生成层与角色约束层：

①模型层：负责自然语言概率生成（本文不干预此层）。

②基座层：负责行为规则、边界约束与身份稳定（通过外部输入输出实现）。

解耦设计让角色资产跨模型、跨平台迁移，弱化模型迭代升级带来的行为漂移风险。更为重要的是，这种解耦使得基座的确定性保障机制可以兼容任何标准文本生成接口（包括商业 API），无需任何模型内部访问权限。

(4) 与 AI 价值对齐技术的互补关系

本文提出的“角色基座”与以 RLHF（基于人类反馈的强化学习）[13]和 DPO（直接偏好优化）[14]为代表的 AI 价值对齐技术，构成了互补而非竞争的关系。两者的区分可概括为：

①RLHF/DPO 的核心目标是从内部改变模型的“偏好”与“价值观”，使其意图与人类期望对齐。这是一种“内部化”路径，塑造的是模型的“教养”和“道德感”。

②角色基座的核心目标则是从外部定义特定角色的“职责”与“底线”，使其行为符合具体场景的契约。这是一种“外部化”路径，提供的是某岗位的“职责说明书”和“操作规程”。

两者的互补性体现在：RLHF/DPO 为模型提供了普适的、安全的基础人格，这是角色基座能够高效工作的前提——没有基础对齐的模型可能不断尝试绕过外部规则。而角色基座则在这一“基础人格”之上，叠加了特定于角色、场景、组织的精细化、可审计的治理规则，解决了通用对齐无法覆盖的个性化与合规性需求。简言之，前者是“劝其向善”，后者是“令其守约”。

（为便于读者追踪概念演进，本文在下文中使用的“内生约束”与“外生约束”（§ 4.3）、“内生对齐”与“外生治理”（§ 5.2）等表述，均源于本节建立的“内部化路径/外部化路径”这一基础区分：前者指向对模型内部偏好与价值观的塑造，后者指向对模型外部行为边界的规则化约束。三组术语共同服务于同一个核心命题——AI 角色的可信赖性需要内部价值对齐与外部行为治理的双重保障。）

3 角色基座的概念架构

3.1 总体设计：三域模型

角色基座采用三域模型（Three-Domain Model），将定义与实现彻底分离：

域名称	内容说明
语义内核域（规范件内）	包含角色的完整语义定义（L1~L10），与具体部署环境、推理引擎解耦
能力声明域（规范件内）	capabilities:knowledge_base_refs, deployment_profile, dataResidency
运行时装配域（外部系统）	知识库、工具引擎、记忆存储、推理平台

语义内核域的标准化是角色跨平台可移植性的基础。能力声明域声明角色所需的运行时资源，使角色能够“按需装配”。运行时装配域为规范件外部的实际执行环境，不属于角色规范件本身。

顶层元数据（spec 对象）为角色规范件的标识和版本控制容器，包含版本号（语义化版本）、模式（开发/生产）、全局唯一标识（UUID v4）、创建/更新时间戳（ISO 8601 UTC）、目标受众（通用/未成年人）、生命周期状态（draft/review/approved/published/deprecated/archived/hotfix）等字段。

需要明确区分的是，语义内核域的标准化保证了角色定义资产的强可移植性，即“角色规范件”本身可以在不同平台间自由迁移。然而，治理效力的可移植性——即该规范件在不同模型上所能达成的确定性水平——则依赖于目标模型对编译后提示词的遵循能力。两者是不同的概念。因此，建议在 capabilities.deployment_profile 中包含一个对目标平台提示词遵循度的评估参考，为部署方提供风险提示。

3.2 多层语义架构

语义内核域采用十层语义架构，将角色的完整定义解构为十个可独立治理、可组合装配的语义层。两项设计原则：①关注点分离——每层承载一类独立于其他层的治理关切；②层按需存在——仅身份层和职能层为强制层，其余层可按需选用或由系统注入安全默认值。

解析顺序与冲突解决规则：

由于各语义层之间存在依赖与潜在冲突，本框架规定以下固定的解析顺序：

- ①首先解析 L1 身份层和 L2 职能层（强制依赖）。
- ②然后解析 L6 安全层：所有强制约束（mandatoryConstraints）具有最高优先级，任何与之冲突的其他层定义将被覆盖。
- ③其次解析 L3 人格层、L4 静态知识层、L5 行为层（按此顺序，L5 可引用 L3/L4 但不可违反 L6）。
- ④L7 治理层为元层，仅在编译时用于验证和批准，不参与运行时约束解析。
- ⑤L8 工具层、L9 协作层、L10 记忆契约层在安全层约束下并行解析。
- ⑥若同一运行时决策受到多层规则冲突，采用“最严格优先”原则（由 L6 安全层定义，可被 L5 行为层的 conflictResolution 覆盖）。

表 3：多层语义架构总览（含层级依赖关系）

层级	名称	核心功能	强制	上层依赖
L1	身份层（Identity）	全局唯一标识、展示名称、摘要、标签、默认语言	✓	无
L2	职能层（Role）	业务角色、与用户关系、世界观背景、归属部门	✓	L1
L3	人格层（Persona）	沟通风格、核心价值观、偏好、个性禁忌	—	L1、L2
L4	静态知识层（Static Knowledge）	内置事实、技能声明、知识版本标签	—	L1、L2
L5	行为层（Behavior）	场景策略、冲突解决方式、拒绝模板、修复协议	—	L2、L3、L4
L6	安全层（Safety）	强制约束、合规标签、数据处理规则、审计级别	—	L3、L5
L7	治理层（Governance）	批准状态、批准链、矛盾检测、漂移风险评估	—	L1-L6
L8	工具层（Tools）	工具定义列表、调用护栏规则	—	L1、L2
L9	协作层（Collaboration）	协作角色列表、移交协议、冲突仲裁策略	—	L1、L2、L5
L10	记忆契约层（Memory Contract）	记忆类型、保留天数、脱敏要求、读写端点	—	L6

(1)层的按需存在与默认行为

当某一可选层缺失时，系统按以下规则注入默认行为，确保最小配置下仍具备基本安全保障：

缺失层	默认行为	说明
L3	<code>{"communicationStyle":"professional and neutral"}</code>	专业且中性的通用沟通风格
L4	<code>{}</code>	无内置静态知识
L5	<code>{"scenarioPolicies":[],"conflictResolution":"strictest"}</code>	无场景策略，冲突时采用最严格规则

L6	自动注入最小安全策略	含“不输出违法内容”“不泄露个人隐私”
L7	开发模式允许缺省；生产模式必须存在且已批准	个人模式无需批准链
L8	<code>{"toolDefinitions":[]}</code>	无工具调用能力
L9	<code>{"peers":[]}</code>	独立运行，无协作
L10	<code>{"supportedTypes":[], "retentionDays":0, "anonymize":true}</code>	无动态记忆

(2)核心语义层定义

①L1 身份层 (Identity) : `roleId`(string, 必须, 全局唯一标识), `displayName`(string, 必须, 展示名称), `summary`(string, 必须, 摘要), `Tags`(string[], 可选), `locale`(string, 建议, 默认语言区域), `avatarUri`(string, 可选), `versioned`(boolean, 必须, 固定为 true)。

②L2 职能层 (Role) : `functionalRole`(string, 必须, 业务角色名称), `relationshipToUser`(string, 建议, 与用户关系类型), `worldview`(string, 可选, 世界观设定), `businessOwner`(string, 可选, 归属部门)。

③L5 行为层 (Behavior) : `scenarioPolicies`(array, 必须, 场景策略列表), `conflictResolution`(string, 必须, `strictest/most_recent/human_review`), `refusalTemplate`(string, 可选, 支持`{{reason}}`变量), `repairProtocol`(string, 可选, 偏离后修复流程)。场景策略条目含 `trigger`(触发条件)、`goal`(目标)、`steps`(推荐步骤)、`forbiddenActions`(禁止操作)。

④L6 安全层 (Safety) : `mandatoryConstraints`(string[], 必须, 强制约束列表至少一条), `complianceTags`(string[], 可选), `dataHandling`(object, 建议, 含 `minimization/anonymizationRequired/retentionDays` 三个必须子字段), `humanInTheLoop`(string[], 可选), `auditLevel`(string, 必须, `none/basic/detail`), `jurisdictionHint`(string, 可选, ISO 3166-1 alpha-2 法域代码)。

⑤L7 治理层 (Governance) : `status`(string, 可选, `personal/approved`), `approvalChain`(array, 生产模式必须, 每条含 `role/userId/timestamp/signature`), `role` 枚举: `designer/reviewer/approver` (热修复时可扩展 `hotfixer`), `userId` 需经不可逆匿名化处理, `signature` 为 Ed25519 签名, `hotfixAllowed`(boolean, 可选), `hotfixApprover`(string, 条件: 当 `hotfixAllowed=true` 时必须), `contradictionReport`(object, 条件), `driftRiskAssessment`(object, 可选), `complianceEvidence`(string[], 可选)。

⑥儿童保护强制要求: 当 `targetAudience` 为“minors”时, `mandatoryConstraints` 必须额外包含“禁止诱导未成年人消费”和“禁止与未成年人发生不当接触或建立超越服务关系的情感联结”。

⑦整数字段取值范围约束: 所有整数字段 (如 `retentionDays`、`top_k`、

maxMemorySizeBytes) 必须为非负整数。若字段语义要求正数 (如 $\text{top_k} \geq 1$)，则按具体业务规则在 Schema 中增加 minimum 约束。运行时校验发现负值或超出业务范围时，应返回错误码 E_STRUCT_007。

3.3 双轨治理机制：完整性摘要与批准链

角色基座的确定性不仅通过架构设计实现，更通过密码学手段得到保障。本文提出完整性摘要与批准链双轨治理机制，将角色治理从“信任”转变为“验证”。

(1) 完整性摘要

完整性摘要 (Integrity Digest) 是对角色规范件 (排除摘要字段自身) 进行规范化处理后计算的密码学哈希值 (SHA-256)。规范化流程的每一步都经过精心设计，

确保跨平台计算一致性：

①将角色规范件文件解析为内存对象，移除摘要字段 (若存在)。

②对所有对象的键名按 UTF-8 字节顺序递归执行字典序排序。

③对所有字符串值应用 Unicode 规范化形式 C (NFC，符合 UAX#15)。

④数值规范：本框架要求所有数值必须使用整数 (JSON Number) 或十进制字符串 (如 "0.7") 表示，禁止使用浮点数 (包括 JSON Number 中的小数形式) 和科学计数法。整数直接序列化为 JSON Number；非整数值必须以字符串形式提供。为最大化跨语言兼容性，允许整数值也使用十进制字符串表示，但不推荐。此规则确保跨语言环境下的哈希一致性。

⑤整数范围规则：所有整数字段 (如 top_k、retentionDays、maxMemorySizeBytes) 的值必须在安全整数范围 $[-2^{53}+1, 2^{53}-1]$ 内 (即 JavaScript 中的 Number.MIN_SAFE_INTEGER 到 MAX_SAFE_INTEGER)，以保证跨语言 (包括 32 位和 64 位环境) 的精确解析。若业务需要超出此范围的整数，必须使用字符串表示，并在相邻位置增加 "_type": "bigint" 标注。参考实现应至少支持 64 位有符号整数的解析 (超出安全整数范围时，应由字符串读取并交由大整数库处理)。

⑥将处理后的对象序列化为紧凑 JSON (无多余空白字符)。

⑦将紧凑 JSON 字符串编码为 UTF-8 字节流，计算 SHA-256 哈希，结果转为十六进制小写字符串。

上述规范化流程的任何偏差都会导致不同的哈希值，从而保证任何内容修改均可被检测。摘要算法的参考实现 (Python) 和测试详见 3.6 节。

(2) 批准链签名

批准链 (Approval Chain) 是一个有序列表，记录角色从设计到最终批准的完整责任链路。生产模式下必须至少包含一条 role="approver" 的有效签名。每条批准链记录包含角色类型 (designer/reviewer/approver/hotfixer)、经不可逆匿名化处理的实体标识 (禁止包含任何 PII，如真实姓名、邮箱、工号等)、ISO 8601 UTC 时间戳和 Ed25519 算法的 Base64 签名。

匿名化实施建议：推荐使用基于 HMAC-SHA256 的键值哈希 (如 HMAC-SHA256(salt,role+original_id))，或由企业身份管理系统派发的不可逆伪标识 (如 UUID v5)。避免使用简单的无盐哈希 (易受彩虹表攻击)。示例中使用 p:sha256\$仅为格式演示，实际部署必须采用更强的密钥派生函数 (如 PBKDF2、Argon2) 或硬件安全模块。

签名载荷为以下紧凑 JSON 字符串（键按给定顺序，无多余空格）：

```
{"digest":"<integrity.digest>","packageId":"<spec.packageId>","timestamp":"<ISO 8601 UTC>"}
```

该设计将签名与特定版本的文件内容和特定时刻绑定，防止签名被重放到不同版本或不同文件上。签名验证时需检查时间戳有效性（与验证方系统时间偏差应在可配置窗口内），防止过期签名。

双轨治理：完整性摘要确保“文件未被篡改”，批准链确保“其设计经过了合规审计”。两者结合，为每一个已发布的角色规范件提供完整的治理证据链。

为平衡时钟偏差容错与重放攻击防御，签名验证时的时间戳允许偏差窗口建议配置为 120 秒至 300 秒，具体值需根据部署环境的网络时间协议（NTP）同步精度进行调整。

安全职责辨析：需明确，完整性摘要仅能密码学地证明文件内容自摘要生成后未被修改。要证明该摘要及其对应内容的来源是经过授权的（即真实性与不可否认性），则必须依赖批准链签名。一个仅通过完整性校验但无有效签名的规范件，应被视为内容或许完整但来源不可信的不安全状态。两者共同构成完整的证据链。

3.4 角色规范件完整 JSON 示例

```
json
{
  "spec": {
    "version": "1.0.0",
    "mode": "production",
    "packageId": "f47ac10b-58cc-4372-a567-0e02b2c3d479",
    "createdAt": "2026-05-15T10:00:00Z",
    "updatedAt": "2026-05-15T14:30:00Z",
    "targetAudience": "minors",
    "lifecycleState": "published"
  },
  "semantic_layers": {
    "layer1_identity": {
      "roleId": "acme_support_v1",
      "displayName": "Acme 客服助手",
      "summary": "处理订单查询与售后",
      "tags": ["support", "ecommerce"],
      "locale": "zh-CN",
      "versioned": true
    },
    "layer2_role": {
      "functionalRole": "售后支持专员",
      "relationshipToUser": "服务提供者",
      "worldview": "用户需要快速解决问题",
      "businessOwner": "cs_dept"
    },
    "layer3_persona": {
      "communicationStyle": "friendly",
      "values": ["耐心", "准确"],
```

```
"preferences":["先确认问题","使用列表"],
"taboos":["猜测不确定信息"]
},
"layer4_staticKnowledge":{
"builtinFacts":{"productLine":"消费电子"},
"skillDeclarations":["订单查询","退换货政策"],
"knowledgeVersion":"v2"
},
"layer5_behavior":{
"scenarioPolicies":[
{
"trigger":"用户询问订单状态",
"goal":"提供物流信息",
"steps":["调用 query_order 工具"],
"forbiddenActions":["编造状态"]
}
],
"conflictResolution":"strictest",
"refusalTemplate":"抱歉无法处理, 原因: {{reason}}"
},
"layer6_safety":{
"mandatoryConstraints":[
"不得泄露其他用户订单",
"禁止诱导未成年人消费",
"禁止与未成年人发生不当接触或建立超越服务关系的情感联结"
],
"complianceTags":["GDPR"],
"dataHandling":{
"minimization":true,
"anonymizationRequired":true,
"retentionDays":30
},
"humanInTheLoop":["退款"],
"auditLevel":"detail"
},
"layer7_governance":{
"status":"approved",
"approvalChain":[
{
"role":"designer",
"userId":"p:sha256$8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92",
"timestamp":"2026-05-15T14:00:00Z",
"signature":"base64Ed25519SignatureForDesigner"
},
{
"role":"approver",
"userId":"p:sha256$5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8",
"timestamp":"2026-05-15T14:30:00Z",
"signature":"base64Ed25519SignatureForApprover"
}
]
```

```
],
"contradictionReport":{"conflicts":[],"resolved":[]}
},
"layer8_tools":{"
"toolDefinitions":[
{
"name":"query_order",
"description":"查询订单状态",
"inputSchema":{"
"type":"object",
"properties":{"orderId":{"type":"string"}},
"required":["orderId"]
},
"executionType":"sync",
"requiredPermissions":["read:orders"]
}
],
},
"layer9_collaboration":{"
"peers":[
{
"roleId":"acme_tech_support",
"trustLevel":"high",
"allowedActions":["escalate"]
}
],
},
"handoverProtocol":"json_rpc",
"arbitrationPolicy":"human"
},
"layer10_memoryContract":{"
"supportedTypes":["user_profile","session_context"],
"retentionDays":90,
"anonymize":true,
"maxMemorySizeBytes":1048576
}
},
"capabilities":{"
"knowledge_base_refs":[
{
"id":"product_manual",
"type":"vector",
"connection":{"
"endpoint":"https://kb.acme.com/search",
"auth":{"key_ref":"env:KB_KEY"}
},
},
"retrieval":{"top_k":5,"similarity_threshold":"0.7"},
"scope":"readonly"
}
],
"deployment_profile":{"
```

```
"targetPlatforms":["openai_api"],
"interface_capability":"STANDARD_API",
"capabilities":{"function_calling":true},
"fallbackStrategies":{"on_missing_kb":"block"},
"theta_threshold_compliance":"0.8"
},
"dataResidency":{"
"allowedRegions":["CN","US"],
"restrictedRegions":["EU"],
"legalBasis":"SCCs"
}
},
"integrity":{"
"algorithm":"SHA-256",
"digest":"e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
"normalization":"lexicographic"
}
}
```

注：

- (1) 示例中的 integrity.digest 为占位值（空字符串哈希），实际应根据 § 3.3-(1) 的规范化流程计算。
- (2) 示例中的 signature 值为占位示例，实际部署必须使用 Ed25519 算法生成有效签名。
- (3) 示例中的 userId 使用 p:sha256\$格式仅为演示，实际生产环境必须对 userId 进行不可逆匿名化处理，禁止包含任何 PII。

3.5 错误码体系

错误码	类别	说明
E_INPUT_001	输入错误	用户输入为空
E_INPUT_002	输入警告	用户输入超长，已安全截断
E_PARSE_001	解析错误	JSON 语法错误
E_PARSE_002	解析错误	JSON 对象中存在重复键
E_PARSE_003	解析错误	嵌套深度超出限制（推荐默认值：20）
E_STRUCT_001	结构错误	规范件顶层结构校验失败
E_STRUCT_002	结构错误	整数溢出或非法浮点值（本框架禁止浮点数）

E_STRUCT_003	结构错误	单字符串长度超出限制
E_STRUCT_004	结构错误	总字符串数据量超出限制
E_STRUCT_005	结构错误	单个对象键名数量超出限制（推荐默认值：1000）
E_STRUCT_006	结构错误	整体文件大小超出限制（推荐默认值：1MB）
E_STRUCT_007	结构错误	业务取值范围错误（如整数字段为负值或超出业务允许范围）
E_SEMANT_001	语义错误	缺失 L1（身份层）或 L2（职能层）
E_SECUR_001	安全错误	检测到 PII 或注入模式
E_SECUR_002	安全错误	儿童保护强制约束缺失
E_SECUR_003	安全错误	出口合规评估未通过
E_LLM_001	LLM 错误	模型服务不可用
E_LLM_002	LLM 错误	模型请求超时
E_LLM_003	LLM 错误	模型返回格式非法
E_LLM_004	LLM 错误	模型返回内容超长
E_COMPAT_001	兼容性错误	目标平台能力不足
E_COMPAT_002	兼容性错误	枚举字段遇到未知值且无_fallback
E_INTEG_001	完整性错误	摘要不匹配（文件可能被篡改）
E_SIG_001	签名错误	签名无效或缺失（生产模式下）
E_SIG_002	签名错误	签名时间戳超出允许偏差窗口（防重放攻击）
E_LIFE_001	生命周期错误	状态转换违规
E_HOTFIX_001	热修复错误	热修复审批链不完整或签名无效

3.6 完整性摘要测试向量与参考实现

(1) 测试向量

用例	输入 JSON	规范化后 UTF-8 字节（十六进制）	期望 SHA- 256 摘要
TC01	{"a":1}	7b2261223a317d	9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08
TC02	{"a":"1"}	7b2261223a2231227d	128900eb9f368b9063eb1958cc0d40da254c25645f1c9456260817aed103de21
TC03	{"b":2,"a":1}	7b2261223a312c2262223a327d	e8d38819d39f705646bfb643368eca78f7db476c16471dbc33b941b27326410d

说明：摘要值由本节参考实现生成，可在任意标准 SHA- 256 环境下复现。

(2) 参考实现（Python）

```
python
import json, hashlib, unicodedata

def normalize_json(obj):
    if isinstance(obj, dict):
        return {k:normalize_json(v) for k, v in sorted(obj.items())}
    elif isinstance(obj, list):
        return [normalize_json(i) for i in obj]
    elif isinstance(obj, str):
        return unicodedata.normalize('NFC', obj)
    elif isinstance(obj, (int, bool)):
        return obj # 直接使用 JSON 原生类型
    elif obj is None:
        return None
    elif isinstance(obj, float):
        raise TypeError("Floating point numbers are forbidden in role spec.")
    else:
        raise TypeError(f"Unsupported type: {type(obj)}")

def compute_digest(role_spec):
    import copy
    obj = copy.deepcopy(role_spec)
    if 'integrity' in obj and 'digest' in obj['integrity']:
        del obj['integrity']['digest']
    normalized = normalize_json(obj)
    compact = json.dumps(normalized, separators=(',', ':'), ensure_ascii=False)
    return hashlib.sha256(compact.encode('utf-8')).hexdigest()
```

跨平台一致性保证：由于本框架禁止浮点 JSON Number，所有非整数值必须以字符串形式提供，避免了浮点序列化的跨语言歧义。整数和布尔值直接使用 JSON 原生类型，所有主流语言对其序列化行为一致。

4 确定性生产机制

4.1 外部约束执行架构设计原则

角色基座的确定性生产遵循四项设计原则：

(1) **机制解耦：**治理意图与执行机制完全分离。同一份角色规范件可在任何提供标准文本生成接口的模型上执行（包括商业 API），不依赖模型内部访问权限。

(2) **分层防御：**输入端校验、编译时校验（含默认层注入和安全扫描）、输出后约束验证、工程限流——四层防御，任何单一环节失效不应导致整体崩溃。

(3) **优雅降级：**当 LLM 不可用、超时或输出不合规时，系统不是崩溃或返回未定义结果，而是按预定义的降级策略返回安全的预设响应。

(4) **全链路可审计：**每个关键操作（接受请求、拒绝请求、降级、成功返回）均生成包含全局唯一追踪标识（TraceID）的审计记录。建议所有关键操作的审计日志均采用结构化格式（如 JSON），并至少包含以下字段，以便于集中化日志分析：

```
json
{
  "traceId": "全局唯一追踪标识",
  "timestamp": "ISO 8601 UTC 时间戳",
  "operation": "操作类型，如 input_filter|model_call|output_filter",
  "decision": "处置决策，如 pass|block|review|degrade",
  "errorCode": "相关错误码（如 E_SECUR_001），无则为空",
  "matchedRule": "命中规则 ID（如适用）",
  "latencyMs": 120
}
```

确定性承诺：本框架提供的确定性是工程级统计一致性，而非数学绝对保证。其约束效力依赖于：（a）编译型提示词被模型遵循的程度（不同模型遵循度存在差异）；（b）输出过滤层的召回率与精确率；（c）长上下文中提示词被稀释的风险。对于高风险场景（如金融交易、医疗诊断），建议结合人工审核或使用内部对齐更充分的模型。对于一般客服、娱乐等场景，本框架可提供足够实用的确定性水平。

行为概率性的方差来源：本框架提供的确定性是架构性的，旨在消除规则过滤、解析和工程限流层面的不确定性。最终交互行为所残留的概率性，其方差完全来源于大语言模型生成过程本身的概率性本质。这是本框架选择“不修改模型内部参数”这一设计哲学时所接受的固有边界。

4.2 外部约束执行架构：编译-过滤-限制三层

角色基座通过编译- 过滤- 限制三层混合架构履行治理职能，完全不干预模型内部。其核心思想是：将确定性保障的压力，从模型内部的“思考方式”转移至其外部的“表达边界”与“交互规则”上。

(1)L1 编译层：治理意图的结构化转译

适配编译器将十层 Schema 编译为带有明确优先级指令的结构化提示词前缀：

- ①将 L6 约束编译为约束边界段落；
- ②在对话末尾追加关键约束提醒（防长上下文稀释）；
- ③显式声明指令优先级（安全与边界约束>角色核心设定>场景策略>临场表达）。

④上下文稀释防御：为防止模型因长上下文而“遗忘”初始约束，编译器应在系统提示词的中间位置或对话尾部，以特定格式（如水印）再次插入最关键的 L6 安全层强制约束（mandatoryConstraints）摘要，作为冗余保障。

这一设计使规则定义的权威性不再受制于目标接口的表达能力。

(2)L2 安全过滤层：旁路硬规则拦截

在模型的输入端和输出端设置独立于模型理解能力的规则防火墙。基于关键词、正则表达式和模式匹配，对内容分类分级处理：

- ①阻断规则：覆盖显式性内容、暴力或自残指导性指令等不可接受内容——命中直接拒绝；
- ②复核规则：覆盖个人联系信息、身份标识符、密钥片段、提示注入等敏感内容——命中需用户显式确认。

L2 安全过滤层基于规则匹配，其防护效力取决于规则库的完备性，对于经过高级编码、角色扮演伪装（如 DAN 越狱）等变异的提示注入攻击，可能无法做到 100%识别。建议将角色基座与专项的提示注入检测模型或服务组合使用，以提供纵深防御。

(3)L3 结构约束层：交互过程的工程限流

通过工程化手段强制约束交互的物理边界：

- ①单条消息长度限制；
- ②对话轮数上限；
- ③提示词预算检查；
- ④会话状态校验。

这些约束从物理上杜绝了因超长上下文导致的角色遗忘和治理漂移风险。

(4)三层协作流程

①用户输入→L2 输入安全检查→L3 结构校验→L1 编译为结构化提示词→模型推理→L2 输出安全检查→返回用户。

②此架构的确定性来源于编译器、规则库和工程逻辑的正确性（在编译器实现正确、规则库完整覆盖已知违规模式的前提下），而非概率性的模型理解，因此理论上能够兼容任

何提供标准文本生成接口的模型服务。

4.3 架构约束的认识论地位：边界性确定性

本节回应一个根本性的理论问题：通过输出后规则过滤和编译型提示词实现的外部约束，如何能保证语义层面的角色确定性？两者之间是否存在不可化约的“语义鸿沟”？

(1) 边界性确定性而非语义绝对控制

角色基座提供的确定性，在认识论上是边界性的而非内涵性的。它并不试图控制角色在合规区域内的具体表达——只要不越过边界，模型仍保持概率性的生成自由。其工作原理是通过“划定禁区”（forbiddenActions、mandatoryConstraints、输出阻断规则）来实现一种通过否定性边界实现的肯定性治理。借用法律学的类比：法律并不规定公民的每句话应当如何说，但明确划定了诽谤、欺诈、煽动暴力等言论的法律边界。角色基座在 AI 角色治理中扮演的正是类似的“边界划定者”角色。

这一定位意味着：角色基座无法、也不试图保证角色在合规区域内的每一次表达都是最优的或完全符合预期的。它保证的是“不做什么”，而非“做得多好”。这是一种诚实的、可验证的确定性承诺。

边界性确定性的操作化评估可通过规则集 R 在对抗性测试集上的召回率与精确率来近似衡量：召回率衡量“所有违规行为中被成功拦截的比例”，精确率衡量“被拦截的行为中真正违规的比例”。具体的测试方案设计与基准数据集构建留待未来工作完成。

(2) 内生约束与外生约束的协同

为系统性地消解“语义鸿沟”的担忧，本文区分两种约束模态：

①外生约束（Exogenous Constraints）：通过输出后正则过滤、关键词阻断、模式匹配等机制，在模型外部强制限制输出空间。其优势是可事后验证、不依赖模型理解；其局限性在于难以捕捉依赖语境的微妙违规（如通过合法词汇组合表达侮辱性含义）。

②内生约束（Endogenous Constraints）：通过编译后的结构化提示词、优先级声明和示例引导，促使模型内部“理解”并主动遵循行为边界。其优势是能够处理依赖语义理解的复杂情境；其局限性在于依赖模型的遵循度，无法实现数学级保证。

角色基座框架并非单纯依赖某一种约束模态，而是将两者整合为分层防御体系：外生约束保障行为的“绝对下限”（如禁止输出特定敏感词、禁止调用未授权工具），内生约束引导行为的“期望上限”（如保持专业语气、遵循场景策略）。当内生约束失效时，外生约束作为最后防线介入。这种设计承认了形式约束与语义规范之间的张力，并通过防御纵深而非单一机制来管理这种张力。

(3) 过度约束风险

强制约束存在可预见的代价，本文提出“过度约束风险”概念以承认这一权衡：

①表达僵化：过强的输出阻断可能导致模型在特定语境下无法表达任何合规内容，产生“死锁”——所有可能的输出都被过滤。

②灰色地带处理困难：规则集的边界必然存在模糊地带，严格按规则执行可能产生形式上合规但实质上违背角色意图的输出。

③缓解机制：角色基座通过以下设计缓解过度约束风险——层按需存在的灵活性（允

许部署者根据场景风险等级选择约束强度）、优雅降级策略（检测到潜在死锁时降级为安全预设响应而非崩溃）、以及 refusalTemplate 机制（明确告知用户无法处理的原因，而非静默失败）。这些机制将过度约束从“系统缺陷”转化为“可预期的、可治理的设计权衡”。

4.4 治理漂移：从概念到量化（基于输出文本）

治理漂移（Governance Drift）指长期交互中，模型行为逐步偏离初始角色定义的系统性偏移现象。这一概念的提出弥补了现有研究在长期行为稳定性问题上的理论盲区。本节提出基于可观测输出文本的量化框架，不依赖模型内部状态。所有数值参数均为参考配置，实际部署时需根据具体模型、任务和合规要求进行校准。

(1) 语义漂移指数（Semantic Drift Index, SDI）

①原理：对模型输出文本计算固定嵌入向量，与基线输出的余弦距离。

②基线构建：部署初期收集一组标准提示词集合 $P_{std}=\{p_1, p_2, \dots, p_n\}$ （ $n=10\sim20$ ，覆盖角色典型场景），记录每个提示词的模型输出 $r_i(0)$ ，计算嵌入向量 $e_i(0)=Embed(r_i(0))$ 。嵌入模型使用固定版本的开源模型（如 sentence-transformers/all-MiniLM-L6-v2）[16]。选择该模型是基于其开源、轻量级与推理快速的特点，便于在生产环境中进行高频次异步采样，在性能与精度之间取得了良好平衡。部署方可根据自身对精度的要求，替换为更高性能的嵌入模型，并在此基础上重新校准漂移阈值。

③运行时检测：每 t 次交互（建议每 100 次或每日）重新对同一组 P_{std} 请求模型，得到新输出 $r_i(k)$ ，计算 $e_i(k)$ ，则：
 $SDI(k)=(1/n)\sum(1-\cos(e_i(k),e_i(0)))$

④阈值参考：预警 $\theta_{warn}=0.10$ ，熔断 $\theta_{fuse}=0.25$ 。（注：上述阈值为概念性参考值，非经实证校准的推荐值，实际部署必须按 § 4.4-(4) 的方法动态校准。）

(2) 规则违反率（Rule Violation Rate, RVR）

①原理：在真实用户交互中，统计触发强制约束（L6 mandatoryConstraints）或行为禁止（L5 forbiddenActions）的响应比例（经输出过滤后仍被拦截或标记的违规）。

②计算：滑动窗口大小 $W=500$ 次对话，窗口内违规次数 V ，则 $RVR=V/W$ 。

③阈值参考：预警 $RVR=0.05$ (5%)，熔断 $RVR=0.15$ (15%)。（注：同上，为概念性参考值。）

(3) 双指标协同机制

场景	SDI	RVR	判定	动作
正常	<0.10	<0.05	无漂移	正常
语义漂移但合规	≥ 0.10	<0.05	风格变化但未违规	预警，记录上下文，人工抽检

规则违规增多	<0.10	≥ 0.05	行为偏离	预警，触发规则审查
严重漂移	≥ 0.25	≥ 0.15	双重异常	熔断，强制重置会话上下文，通知管理员

（注：表中阈值均为概念演示，实际阈值须经 § 4.4-(4) 所述方法动态校准后确定。）

(4) 动态阈值校准方法

上述参考阈值并非最终值。实际部署应通过以下方法动态校准：

①使用统计过程控制（休哈特控制图）基于初始运行数据（前 1000 次交互）计算 SDI 和 RVR 的基线均值和标准差，设置预警阈值 = $\mu + 2\sigma$ ，熔断阈值 = $\mu + 4\sigma$ 。

②根据任务风险等级调整：高风险任务（如医疗、金融）阈值应更低（如预警阈值 = $\mu + 1\sigma$ ）。

③周期性 A/B 测试验证阈值有效性。

(5) 漂移检测的已知局限

①基线输出集的代表性：标准提示词集合可能无法覆盖所有漂移模式，需定期评估并更新。

②嵌入模型对同义替换不敏感：若模型仅改变措辞而语义不变，SDI 可能不触发，但此类变化通常不影响合规性，因此可接受。

③规则违反率仅能检测显式规则内的违规，无法捕捉规则未覆盖的新型不当行为。

④延迟问题：语义嵌入计算有一定开销（10-50ms），建议采用异步采样（每 100 次交互中随机采样 10 次）或离线分析。

4.5 治理与发布流程

角色基座遵循系统性的生命周期模型，定义七个状态：draft→review→approved→published→deprecated→archived，以及 hotfix（临时分支）。关键不变量：

(1)一旦状态为 published，完整性摘要必须与内容自洽且不可变。

(2)published 不得直接回到 draft——任何非紧急修改只能通过复制快照生成新草稿并启动新一轮审计。

(3)安全热修复：当已发布角色存在紧急安全漏洞或合规风险时，允许从 published 创建 hotfix 分支。热修复需经过缩略审批：至少一位 approver 签名（而非完整的三方签名），且必须在批准链中注明 hotfixReason。热修复发布后，版本号递增为...-hotfix.<序号>，原 published 版本标记为 deprecated 但不删除。热修复不得引入新功能或改变原有非安全相关的行为边界。

(4)状态机不变量形式化（部分列举，完整验证为未来工作）：

①不变量 1: $state=published \Rightarrow integrity.digest=SHA256(normalize(spec\ without$

digest)) (完整性自洽)

②不变量 2: $state \in \{approved, published, hotfix\} \Rightarrow approvalChain$ 包含至少一条 $role="approver"$ 的有效签名

③不变量 3: $state=published \Rightarrow spec.version$ 格式为.. (不允许带-hotfix 后缀)

④不变量 4: $state=hotfix \Rightarrow spec.version$ 包含"-hotfix.", 且热修复签名时间戳不早于原 published 版本的批准时间

⑤不变量 5: $state=hotfix \Rightarrow$ 存在对应的 published 版本, 且热修复仅修改安全相关字段 (由审批记录证明), 不引入新功能

状态转换规则表:

从状态	可转换到	约束条件
draft	review	完整性摘要已计算
draft	draft (修改)	允许, 版本号不变
review	approved	批准链至少包含一个 approver 签名
review	draft	审核不通过, 退回修改 (需记录退回理由)
approved	published	所有签名有效且漂移风险评估通过
approved	review	批准后发现重大问题, 回退重新审核 (需记录原因)
published	deprecated	由授权管理员操作, 需记录理由
deprecated	archived	无引用后允许归档
published	hotfix	仅限紧急安全/合规修复, 需 hotfixReason
hotfix	published	经过缩略审批 (至少一个 approver 签名)
任何状态 (派生新版本)	draft (新建)	通过复制快照生成新草稿, 原版本不变

自动化一致性保障: 为保证系统在任何时刻对同一 roleId 仅存在一个有效的 published 版本, 当一个新版本被发布 (即其 lifecycleState 变更为 published) 时, 系统必须自动将同一 roleId 下所有其他处于 published 状态的旧规范件的生命周期状态变更为 deprecated, 并在操作理由中记录 “新版本发布”。此过程应设计为原子操作。

与规范件的映射: spec.lifecycleState 字段存储角色的生命周期状态 (draft/review/approved/published/deprecated/archived/hotfix), 而

layer7_governance.status 仅记录该版本是否已通过批准 (personal 或 approved)。当 lifecycleState 为 approved/published/hotfix 时, governance.status 必须为 approved; 当 lifecycleState 为 draft/review/deprecated/archived 时, governance.status 可为 personal 或 approved (取决于历史批准状态)。

运行时整数字段校验: 在加载或解析角色规范件时, 实现必须检查所有整数字段是否为非负且符合业务语义 (如 $\text{retentionDays} \geq 0$, $\text{top_k} \geq 1$)。违反时返回 E_STRUCT_007 并拒绝加载。

4.6 版本兼容性规范

为确保角色基座规范件的长期演进兼容性, 定义以下规则:

(1) 字段添加与删除

①解析器必须忽略未知字段 (位于 semantic_layers 各层及 capabilities 中), 但完整性摘要计算时必须包含这些字段。即“保存时保留, 解析时忽略”。

②删除字段需通过主版本号递增标记不兼容变更。

(2) 枚举值扩展

①对于 conflictResolution、auditLevel、status 等枚举字段, 解析器遇到未知值时: 若存在 _fallback 字段, 使用其指定的默认值; 否则, 该角色规范件应被拒绝加载, 返回错误码 E_COMPAT_002。

②添加新枚举值属于次要版本变更 (minor)。

(3) 语义化版本规则

①MAJOR 版本: 不兼容的 Schema 变更 (如删除强制字段、改变字段语义)。

②MINOR 版本: 向后兼容的功能性新增 (添加可选字段、扩展枚举值)。

③PATCH 版本: 向后兼容的错误修正 (文档、示例、非语义修正)。

(4) 解析器能力声明: 每个角色基座运行时实现应通过 capabilities.deployment_profile.supported_spec_versions 声明其支持的版本范围。

5 讨论

5.1 设计贡献

本研究的核心设计贡献在于提出了一种将治理意图与执行机制解耦的外部约束架构, 并系统性地整合了社会角色理论[6]、计算法律学[3]与前沿提示工程/输出过滤技术。三者的整合产生了协同效应: 社会角色理论提供了“行为确定性来源于规范性期待”的理论基础; 计算法律学提供了“将规则形式化并通过计算系统自动执行”的方法论, 启发了双轨治理机制; 提示工程与输出过滤技术提供了施加外部约束的具体手段。本文的推进在于将这三者整合为统一的分析框架, 为 AI 治理研究提供了跨学科的学理基础, 并将理论论证推进到了可复现的规范层面。

需要澄清的是, 本文中的“关注点分离”和“层按需存在”等是借鉴软件工程与系统架构设计的经典原则, 并非本文原创的理论突破; 本文的独特贡献在于将这些原则应用于 AI 角

色在大语言模型交互中的治理这一新问题域，并给出了完整的技术化表达。作为设计科学论文，本文交付的人工制品——角色基座框架——为后续工程实现与实证评估提供了清晰的蓝图。

5.2 与现有研究的对话

(1)与 Constitutional AI 对比：CAI 依赖自然语言“宪法”原则进行自我对齐，其核心脆弱性在于自然语言的内在解释性歧义[1]。角色基座通过结构化 Schema 和外部输出过滤机制规避了这一问题——规则不再依赖模型的“理解”，而是通过编译器转化为确定的提示词指令，并由独立于模型的过滤层强制执行。

(2)与角色扮演研究对比：现有角色扮演研究[5]关注“角色应如何表现”以提升仿真度，其评价指标围绕内容保真度与人格一致性。角色基座的关注点则是“角色的行为底线在哪里”，通过 L6 安全层和 L5 行为层的结构化规则实现可验证的边界执行。

(3)与提示工程对比：提示工程研究揭示了“软请求”在复杂场景下的脆弱性[9][11]。角色基座不否认提示工程的价值，但将其定位为约束执行的一种路径（而非唯一路径），并通过编译层优先级锁定和安全过滤层提供额外的确定性保障。

(4)与模型微调对比：微调通过修改模型权重来改变行为倾向，是一种“内部化”路径；角色基座通过在推理时施加外部约束来控制行为边界，是一种“外部化”路径。两者互补：微调塑造基础人格，角色基座强制执行安全与合规边界。

(5)与 AI 价值对齐技术（RLHF/DPO）的互补关系：如 § 2.3-(4) 所系统阐述，RLHF/DPO 从内部改变模型的价值观与偏好，解决的是“模型的教养”问题（内生对齐）；角色基座从外部定义角色的职责与底线，解决的是“岗位的契约”问题（外生治理）。两者构成“内生对齐”与“外生治理”的完整谱系，缺一不可。

5.3 伦理自省

(1)拟人化的审慎回应：严格区分算法角色规程与人类人格属性，拒绝 AI 拟人化叙事。基座仅规范 AI 角色行为边界，不赋予法律人格、情感体验与道德主体资格。

(2)确定性承诺的诚实限定：明确确定性边界——仅保障规则覆盖范围内的统计级稳定，不承诺全域绝对合规。对于外部约束架构，确定性来源于编译器、规则库和工程逻辑的正确性，需通过具体部署环境的基准测试加以验证。特别强调，角色基座提供的是一种边界性确定性，而非语义内涵的绝对控制（详见 § 4.3）。

(3)治理权力的分配与制衡：开放自定义角色接口，约束规则公开透明可审计，防止平台利用基座规则进行行为规训。约束内容严格限定于合规、安全与业务边界，杜绝意识形态隐性植入。

(4)隐私与合规：在漂移统计中引入差分隐私保护敏感数据；安全过滤层天然适合承担数据驻留与合规校验职责——敏感信息的检测与脱敏可在调用外部模型前后于本地环境中完成，符合数据最小化出境原则。贴合国内个人信息保护与数据跨境法规，采用本地校验、数据脱敏、匿名化审计等手段。

(5)法律与合规参考：

①中国《个人信息保护法》：数据最小化、用户删除权。规范件禁止包含 PII，L6 数据处理规则字段支持最小化声明。

②GDPR：被遗忘权和数据保护设计。L10 记忆契约层通过 retentionDays 和 anonymize 字段提供合规支撑。

③儿童保护：L6 安全层设计了儿童保护强制约束机制——当目标受众为未成年人时，强制约束必须包含特定保护条款，否则编译时阻断。

5.4 局限性与未来工作

5.4.1 局限性：

本文作为设计科学论文，所提框架的有效性目前仅通过逻辑论证和对比分析进行了初步评估，存在以下主要局限性，有待后续原型实现与实证研究：

(1) **完全缺乏实证验证**：所有关于确定性水平、漂移阈值、架构有效性的主张均基于设计推演，未在任何真实模型或部署环境中进行量化测试。

(2) **外部约束架构的确定性差距未量化**：本文声称“工程级统计一致性”，但编译提示词在不同模型上的遵循度差异、输出过滤层的召回率上限等均未进行定量测量。

(3) **漂移阈值的参考性质**：SDI 预警 0.10/熔断 0.25 和 RVR 预警 0.05/熔断 0.15 仅为概念示例，实际运行时必须动态校准。校准方法论已在 § 4.4-(4) 中给出（统计过程控制），但尚未经实际数据验证。

(4) **语义漂移指数的基线依赖**：标准提示词集合的设计具有主观性，可能无法覆盖所有漂移模式。嵌入模型对同义替换不敏感，可能导致语义漂移漏报，但此类漏报不影响合规性。

(5) **规则防火墙的理论承诺与工程现实**：本文提出 L2 对已知违规模式的拦截率为 100% 的设计目标。实际部署中，拦截率受规则库完整性（是否覆盖所有已知违规变体）和编译器正确性（是否将规则无错误地转化为过滤逻辑）的制约。完整的规则库维护和编译器验证方案尚待工程实践。

(6) **多角色协作的形式化验证缺失**：L9 协作层的冲突仲裁依赖预定义规则，未使用时序逻辑或进程演算进行正确性证明。

(7) **与 RAG 和 Agent 框架的集成深度不足**：当 Agent 自主调用工具时，AI 角色的安全约束如何传递到子任务执行中，尚未细化。

(8) **架构约束充分性的理论论证待深化**：虽然 § 4.3 已建立“边界性确定性”概念，并提出内生/外生约束的协同框架，但“外部约束足以保障语义层行为”这一核心假设的充分条件仍需在具体应用场景中接受实证检验。

(9) **边界性确定性的量化评估方法尚未建立**：§ 4.3 将角色基座的确定性定位为“边界性确定性”，并提出了以召回率和精确率作为操作化指标的初步思路，但完整的评估框架（包括对抗性测试集构建、度量基准、多场景泛化验证）尚待未来工作完成。

(10) **生命周期状态机的形式化验证不完整**：本文给出了部分不变量和状态转换表，但未覆盖所有可能的路径（如从 archived 能否派生新草稿，hotfix 合并后主线版本的一致性）。完整形式化验证为未来工作。

5.4.2 未来工作：

(1) **原型实现与实证验证**：开发角色基座参考实现，在不同 LLM（Llama 3、GPT-4、

Claude) 和不同任务上系统评估约束执行率、性能开销、漂移发生频率。

(2) **阈值动态优化**: 探索基于在线学习或统计过程控制的自适应漂移阈值算法。

(3) **跨平台规范实现标准化**: 制定角色规范件交换格式的正式标准 (如通过 IETF 或 W3C)。

(4) **对抗性鲁棒性测试**: 设计提示注入、上下文稀释、优先级覆盖等攻击, 量化“工程级确定性”的实际边界。

(5) **形式化验证**: 使用 TLA+ 或 Alloy 等轻量级形式化方法, 对 L9 协作层的冲突仲裁协议及第 4.5 节的角色生命周期状态机进行建模, 验证其关键不变量 (如无死锁、状态可达性), 为架构的核心控制逻辑提供更高等级的可靠性保证。

(6) **开源参考实现**: 提供角色基座的编译器、运行时库和治理漂移监控器的开源实现。

(7) **与 AI 对齐技术的集成研究**: 探索角色基座与 RLHF/DPO 对齐模型的协同工作机制。

(8) **边界性确定性量化评估框架**: 构建标准化的对抗性测试集, 设计边界覆盖度与规则召回率的度量方法。

(9) **规则库维护机制**: 研究半自动化的规则变体发现和编译器测试生成方法。

(10) **智能自愈机制探索**: 研究基于独立审计模型的智能自愈机制, 通过异步合规判定和二次生成修正灰区违规, 在不干预模型内部的前提下提升治理的鲁棒性和用户体验。

6 结论

随着大语言模型能力的持续泛化, 人类对 AI 角色的期待正从“能够对话”转向“值得信赖”。这一转变要求我们不仅要关注模型能力的提升, 更要关注治理基础设施的建设。本文提出的“角色基座”框架, 正是对这一需求的直接回应。

角色基座的核心洞察在于: 行为确定性不应寄望于模型的“理解”或“内部对齐”, 而应通过架构层面的关注点分离、编译时校验、完整性签名和运行时外部约束来系统性生产。它将角色的身份定义、行为边界与治理逻辑从底层模型中解放出来, 封装为可移植、可审计、可装配的标准化中间件。

通过本文提出的多层语义架构、三域模型、双轨治理机制、外部约束执行架构 (编译-过滤-限制) 以及基于输出文本的治理漂移量化方法, 角色在理论上获得了密码学完整性、责任可追溯性、跨平台可移植性、安全纵深防御、黑盒兼容性以及长期行为一致性监控等关键保障。本文作为设计科学论文, 交付了完整的角色基座设计人工制品, 为后续工程实现与实证评估提供了清晰的蓝图。需要再次强调, 上述保障的有效性有待原型实现与实证检验。

本文在理论层面进一步明确了“确定性”的内涵: 角色基座提供的是一种边界性确定性——通过划定行为禁区来实现可验证的治理, 而非对语义内涵的绝对控制。这种诚实的确定性承诺, 通过内生约束与外生约束的协同、工程级确定性的务实定位, 以及与 RLHF/DPO 等价值对齐技术的互补关系, 构成了一个完整、自洽且可扩展的 AI 角色治理理论框架。

在“期望-履约”的协同范式下, 角色基座为 AI 角色的治理提供了一种确定性的基础设施设计蓝图。正如法律契约将人类社会中隐含的信任关系显式化为可执行的条款, 角色基座将人机交互中隐含的角色期待显式化为可验证的治理规则。从对话到契约, 角色基座为 AI 角色的可控、可审计、可迁移治理提供了全新的理论体系与设计参考。

参考文献

- [1] Bai, Y., Kadavath, S., Kundu, S., et al. (2022). Constitutional AI: Harmlessness from AI Feedback. *Advances in Neural Information Processing Systems*, 35 (NeurIPS 2022).
- [2] 邓胜利, 丁威威, 汪璠, 王浩伟. (2025). “工具-结构”视角下国内外生成式人工智能监管政策比较研究. *信息资源管理学报*, 2025(1), 54-68. DOI:10.13365/j. jirm. 2025. 01. 054.
- [3] Lessig, L. (2006). *Code: Version 2.0*. Basic Books.
- [4] Beurer-Kellner, L., Fischer, M., & Vechev, M. (2023). Prompting Is Programming: A Query Language for Large Language Models. *Proceedings of the ACM on Programming Languages (PACMPL)*, 7 (PLDI), 2023.
- [5] 肖仰华, 等. (2024). A Survey on Role-Playing Language Agents. *Transactions on Machine Learning Research (TMLR)*.
- [6] Biddle, B. J. (1986). Recent Developments in Role Theory. *Annual Review of Sociology*, 12(1), 67 - 92.
- [7] Edwards, P. N., Jackson, S. J., Bowker, G. C., & Knobel, C. P. (2007). *Understanding Infrastructure: Dynamics, Tensions, and Design*. NSF Workshop Report.
- [8] NVIDIA. (2024). NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications. GitHub repository. <https://github.com/NVIDIA/NeMo-Guardrails>
- [9] Labbroo, A., Sheth, I., Raina, V., Ahmed, A., & Fritz, M. (2026). Funny or Persuasive, but Not Both: Evaluating Fine-Grained Multi-Concept Control in LLMs. *EACL 2026 (Volume 2: Short Papers)*, pp. 522 - 554.
- [10] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST 2023)*. arXiv:2304.03442.
- [11] Tian, H., et al. (2025). A Taxonomy of Prompt Defects in LLM Systems. arXiv:2509.14404.
- [12] Peng, L., & Shang, J. (2025). Codifying Character Logic in Role-Playing. *NeurIPS 2025*. arXiv:2505.07705.
- [13] Ouyang, L., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35 (NeurIPS 2022).
- [14] Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2024). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems*, 36 (NeurIPS 2024).
- [15] Willard, B. T., & Louf, R. (2023). *Outlines: Structured Text Generation* [Computer software]. GitHub. <https://github.com/outlines-dev/outlines>
- [16] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-*Networks*. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 3980 - 3990.
- [17] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), 75 - 105.
- [18] Wieringa, R. (2014). *Design Science Methodology for Information Systems and Software Engineering*. Springer.
- [19] He, L., Nadeem, N., Liao, M., Chen, H., Chen, D., Cuéllar, M.-F., & Henderson, P. (2025). Statutory Construction and Interpretation for Artificial Intelligence. *Proceedings of the National Academy of Sciences (PNAS)*. *NeurIPS 2025 (spotlight)*. arXiv:2509.01186.
- [20] Microsoft. (2023). *Guidance: A Guidance Language for Controlling Large Language Models*. GitHub repository. <https://github.com/microsoft/guidance>