带有宪法式约束的进化元提示框架：迈向自主与对齐的人工智能系统

摘要

本报告详细阐述了一种新颖的、用于大型语言模型（LLM）自优化系统的架构框架。该框架利用进化算法（Evolutionary Algorithms, EAs）来优化其自身的高级推理结构，即元提示（Meta-Prompts）。其核心创新在于集成了一个源自宪法式人工智能（Constitutional AI, CAI）原则的、动态的多目标适应度函数，确保进化过程不仅以任务性能为导向，同时严格遵守一套预定义的伦理与安全原则。报告首先解构了构成该框架的三大技术支柱：元提示的形式化与功能化范式、利用LLM驱动离散提示空间进化计算的机制，以及作为强大对齐技术的CAI方法论。随后，报告将这些元素整合成一个协同工作的系统，详述其操作循环、宪法式适应度函数的设计，及其实现持续、自主和对齐的自我完善潜力。最后，本报告对该系统面临的深层挑战进行了批判性分析，包括计算成本、宪法定义的脆弱性，以及此类优化中固有的对抗性“猫鼠游戏”动态。我们认为，该框架代表了在创建能力更强、适应性更好且可验证有益的人工智能系统方面迈出的重要一步。

---

1. 元提示的架构基础

本节旨在建立对元提示的深入理解，超越表层定义，探索研究中发现的两种主导性概念范式。其目标是将元提示定位为一类用于构建和自动化LLM高级推理的方法，而非单一技术。

1.1. 范式转变：从提示工程到元提示

人工智能的发展历程见证了从手动、任务特定的提示词构建，向开发更高阶、可复用的指令框架的演进[1]。提示工程是这一演进的基础，而元提示则代表了一种更高层次的抽象，其核心目标是生成或优化提示本身[1]。这一转变的本质在于，元提示将重点从提供基于内容的示例（即“思考什么”）转移到提供形式化的程序性指导（即“如何思考”）[3]。它是一种专注于问题结构和语法的先进技术，而非具体内容细节[4]。

1.2. 范式一：形式主义方法——作为函子映射的元提示

此范式深植于范畴论与类型论的理论基础[4]。在数学上，一个范畴（Category）由一系列对象（Objects）以及对象之间的态射（Morphisms）或箭头组成[3]。

在此框架下，元提示（Meta Prompting, MP）被形式化为一个函子（Functor）M，它将一个任务范畴（T）映射到一个结构化提示范畴（P）[3]。这种数学形式保证了组合式的问题解决策略可以被系统地分解为模块化、可复用的提示结构[3]。

- 关于对象（**On Objects**）：对于任务范畴中的每个任务对象X（例如，求解一个二次方程问题），函子M会为其分配一个对应的结构化提示M(X)，该提示将勾勒出解决此类问题的必要步骤[6]。
- 关于态射（**On Morphisms**）：对于任务范畴中的一个态射f:X→Y（例如，将一个基础代数问题转化为一个高级代数问题的过程），函子M会为其分配一个对应的提示转换态射M(f):M(X)→M(Y)，从而相应地调整提示结构[6]。

递归元提示（**Recursive Meta Prompting, RMP**）是这一范式的关键扩展，它将该框架递归地应用于自身，使LLM能够自主生成并优化其自身的提示[3]。这一自完善循环在形式上被建模为一个单子（Monad），为自动化提示工程提供了一个有原则的理论框架[3]。

该方法的主要优势在于，通过强调结构而非详尽内容，显著提升了令牌效率（Token Efficiency），同时通过最小化特定示例的影响，实现了模型间更公平的“零样本”（Zero-Shot）性能比较[5]。实验证明，一个配备了单个元提示的Qwen-72B模型，在MATH和GSM8K等复杂推理任务上取得了业界顶尖的成果[3]。

1.3. 范式二：功能主义方法——作为指挥家-专家协同的元提示

此方法将单个LLM转变为一个“多面指挥家”，负责管理和整合由同一LLM扮演的多个独立“专家”实例所处理的查询[7]。

其操作流程如下：

1. 一个高层次的“元”提示指示LLM扮演“指挥家”的角色。
2. 作为指挥家的LLM将一个复杂任务分解为多个更小、更易于管理的部分。
3. 它将这些子任务分配给不同的专业“专家”角色（例如，“批判家”、“编码员”、“创意作家”），并为每个角色提供量身定制的自然语言指令。
4. 同一个LLM随后“扮演”每个专家的角色，为各个子任务生成响应。
5. 最后，指挥家整合所有专家的输出，利用其自身的批判性思维和验证流程，将最终结果提炼成一个连贯的叙述或解决方案[7]。

该方法的主要优势在于其高度的灵活性和任务无关性，同样的高级指令可以应用于不同领域，从而简化了用户交互[7]。通过在单个模型内部创建一个协作式的“专家小组”，它显著提升了性能，并改善了生成内容的真实性和连贯性[7]。

1.4. 深度分析与启示

对这两种元提示范式的深入检视，揭示了人工智能自动化领域一条根本性的思想分歧。这两种路径分别代表了计算机科学中的经典二元对立：一方是追求形式化、可证明正确的系统，另一方则是拥抱动态、灵活、基于智能体的系统。形式主义范式通过范畴论寻求数学上的保证[3]，确保问题结构在提示结构中得以保留，从而实现可预测、可分解的推理。这与软件工程中形式化方法的目标不谋而合。相比之下，功能主义的指挥家模型则优先考虑涌现行为和适应性[7]，它不依赖于形式证明，而是构建一个模仿人类专家协作的动态角色扮演架构，类似于基于智能体的建模或微服务架构。因此，在这两种范式间的选择不仅是技术层面的，更是一种哲学层面的权衡，它反映了在数学严谨性与实用灵活性之间的取舍。一个成熟的系统或许需要将两者进行混合。

更进一步，两种范式，特别是递归元提示（RMP）[3]和指挥家模型的自主决策能力[7]，共同构成了迈向能够管理自身认知过程的自主AI的基础。传统的提示工程类似于下达一个指令，而元提示则是在传授一种思维方法论或框架。当RMP和指挥家模型使LLM能够优化自身提示或决定咨询哪些“专家”时，系统便开始展现出元认知（Metacognition）的雏形——即“思考如何思考”的能力。这不仅仅是为了获得更好的输出，更是为了创建一个能够自主改进其问题解决

策略的系统，这是任何通用人工智能所必需的关键能力。

特征	范式一：形式主义（函子映射）	范式二：功能主义（指挥家-专家）
核心原则	结构保持的映射，保证推理的可组合性	动态的任务分解与专家角色协同
理论基础	范畴论、类型论	基于智能体的类比、协同问题解决
关键机制	任务与提示之间的函子映射	指挥家LLM对专家LLM的编排与整合
自完善方法	递归元提示（RMP），形式化	指挥家自主决定使用的提示

	为单子	和专家
主要优势	可证明的组合性、令牌效率、零样本公平性	任务无关性、动态灵活性、增强的连贯性
理想用例	结构化、可分解的复杂推理任务（如数学、逻辑）	开放式、多方面的任务（如报告撰写、创意生成）
关键研究来源	[3]	[7]

---

2. 进化算法：提示优化的引擎

本节将深入探讨利用进化算法（EAs）在庞大、离散且不可微的自然语言提示空间中进行导航的机制。它将确立EAs作为一种强大的、无梯度的优化方法，尤其适用于应对这一挑战。

2.1. 提示优化问题：在组合景观中导航

提示优化本质上是一个组合优化问题，其潜在的提示搜索空间规模极其巨大[8]。手动的试错法效率低下，且几乎不可能找到最优解[9]。进化算法为此提供了一个强有力的解决方案。EAs擅长在无需梯度信息的条件下探索广阔的离散搜索空间[9]。它们将提示视为可以跨代进化的“基因”[12]，其基于种群的搜索方式有助于避免陷入局部最优，并保持解的多样性[9]。

2.2. 进化提示的机制：**EvoPrompt**框架

进化过程以“代”（generations）为单位进行迭代[11]。一个典型的框架，如EvoPrompt，其核心循环如下：

1. 初始化（**Initialization**）：从一个多样化的候选提示种群P0开始[11]。这可以通过对用户提供的种子提示进行释义，或基于训练样本生成初始猜测来完成[13]。
2. 评估（**Evaluation**）：种群中的每个提示pi都会在一个开发集$\mathcal{D}上进行评估，以获得一个适应度分数s_i = f_{\mathcal{D}}(p_i)$[11]。适应度函数用于衡量提示引导LLM产生期

望输出的效果，例如分类任务的准确率或摘要的质量[12]。

3. 选择（**Selection**）：适应度最高的个体（提示）被选为下一代的“父本”。可以采用轮盘赌选择等方法[12]。
4. 进化（**Evolution**）：通过遗传算子从父本生成新的“子代”提示。关键在于，这些针对自然语言提示的算子是由一个LLM来执行的[11]。
    - 交叉（**Crossover**）：结合两个或多个父本提示的部分内容，创造一个新的提示。可以指示一个LLM融合两个成功提示的最佳元素[13]。
    - 变异（**Mutation**）：对一个提示引入微小的、随机的改变。可以指示一个LLM对提示进行释义、增加细节、改变视角或微调其结构[12]。
5. 更新（**Update**）：新生成的子代经过评估后被整合进种群，替换掉适应度较低的个体[11]。此过程重复进行，直到达到预设的代数或算法收敛[12]。

诸如EvoPrompt这样的框架已被证明，在数十个数据集上的表现显著优于人类专家设计的提示以及其他自动生成方法[11]。

2.3. 优化范式比较：进化方法与基于梯度的方法

- 进化方法（无梯度）：
    - 优点：适用于无法获取梯度的黑盒模型[20]。更擅长探索复杂广阔的搜索空间，避免局部最优[9]，并能够发现新颖的、涌现式的推理策略[22]。
    - 缺点：计算成本高昂，因为每一代都需要对种群中的每个个体进行评估[9]。相比有指导的方法，其样本效率可能较低。
- 基于梯度的方法（白盒/文本梯度）：
    - 优点：由于遵循梯度方向，样本效率更高，优化过程可能更稳定、目标更明确[24]。
    - 缺点：需要对模型内部（梯度、logits）的白盒访问权限，这对于许多专有模型而言是不可能的[21]。在扩散模型或不可微的嵌入层上进行反向传播在技术上极具挑战性[26]。在黑盒场景中使用的“文本梯度”是一种近似方法，其效力可能不及真实的梯度[13]。
- 混合方法：一些先进的方法，如PhaseEvo，将进化式的全局探索与对表现最佳个体的定向“文本梯度”更新相结合，试图兼得两者之长[13]。

2.4. 深度分析与启示

在这一领域的研究中，一个最深刻的发现是LLM在优化循环中扮演的双重角色。在传统的进化算法中，遗传算子（如变异和交叉）是硬编码的算法，例如位翻转或单点交叉[28]。然而，当“基因”是自然语言提示时，这些简单的算子会破坏其语义连贯性，因而无效[11]。以EvoPrompt为代表的突破性

思想[11]，是利用LLM自身的自然语言理解能力来

执行变异和交叉操作。优化器通过提示一个LLM来“创造性地融合这两个成功的提示”或“在保持意图不变的情况下轻微改写这个提示”。由此，系统变得自我指涉：LLM既是被优化的对象（通过其对进化提示的响应来体现），同时也是执行优化操作的工具。这创造了一个强大的协同循环，充分利用了我们试图引导的智能本身。

进化算法的成功也间接证明了提示的“适应度景观”（fitness landscape）——即提示措辞与其性能之间的关系——并非平滑或简单的，而是“崎岖不平”且具有“上位性”（epistatic）的，即一个词的效果严重依赖于其他词的存在[16]。如果景观是平滑的，那么简单的爬山法或基于梯度的方法将永远是更优的选择。然而，基于种群的进化算法在经验上取得的成功[11]表明，这个景观充满了局部最优点，这些陷阱会困住更简单的算法。这意味着，提示中微小的改动可能导致性能发生巨大的、不可预测的变化，并且最优提示可能是各种短语的非直觉组合。因此，对这一景观的拓扑结构进行描绘，已成为一个关键的开放性研究问题[16]。理解其结构可能催生出专为自然语言指令这一独特领域量身定制的、效率更高的新型优化算法。

然而，近期的研究揭示了一个令人不安的趋势：黑盒优化方法（包括进化算法）的有效性似乎随着模型规模的增大而减弱，形成一种“逆向规模法则”（inverse scaling law）[21]。像EvoPrompt这样的黑盒方法在特定规模的模型（如GPT-3.5）上取得了巨大成功[25]，但当应用于更大规模的模型（如Qwen 2.5 72B, DeepSeek V3）时，其性能提升变得微乎其微[25]。一种假设是，更大、能力更强的模型本身已接近其性能上限，并且对这些方法所探索的微小提示变动不那么敏感，因为它们具有更强的“零样本”理解能力。这对用户提出的整个系统架构的长期可行性构成了重大挑战。虽然架构本身是合理的，但其未来的价值可能取决于能否克服这一定律。未来的研究必须致力于开发更强效的进化算子或混合方法，以便在超强能力的模型上仍能发掘有意义的性能提升空间。

---

3. 宪法式AI：构建有原则的系统行为框架

本节将阐释宪法式AI（CAI），它不仅是一种训练技术，更是一种将规范性原则嵌入AI行为的哲学与实践框架。这是所提出系统中至关重要的对齐组件。

3.1. 对齐问题：超越RLHF

基于人类反馈的强化学习（RLHF）是业界标准的对齐方法，但其存在诸多弊端。它需要大量昂贵且耗时的人类标注，这些标注数据通常是私有的且难以解释，并且可能导致模型变得过度规避问题[31]。由Anthropic公司提出的CAI，旨在无需在每一步都依赖大量人类反馈的情况下，训练AI模型变

得有用、无害且诚实[31]。它用一套成文的原则——即“宪法”——取代了直接的人类监督[31]。

3.2. CAI的两阶段训练过程（RLAIF）

CAI的训练过程，即基于AI反馈的强化学习（RLAIF），分为两个主要阶段：

- 阶段一：监督学习（批判与修正）
  1. 从一个仅经过“有用性”RLHF训练的模型开始。
  2. 向该模型输入有害的或“红队演练”的提示，以引诱其产生不良响应[31]。
  3. 然后，向模型展示一条从宪法中抽取的原则（例如，“选择伤害性更小的回应”），并要求它批判自己的原始回应，并根据该原则重写一个合规的回应。
  4. 这个由宪法指导的自我批判和修正过程，创建了一个新的、由无害回应组成的数据集[33]。
  5. 最后，用这个新数据集对原始模型进行微调。
- 阶段二：基于**AI**反馈的强化学习（**RLAIF**）
  1. 经过第一阶段微调的模型，针对各种提示生成两两一组的回应。
  2. 另一个AI模型（或同一个模型）被给予一条宪法原则，并被要求在两个回应中选择更优（即更符合宪法）的一个。
  3. 这个过程产生了一个大规模的、由AI生成的偏好标签数据集[32]。
  4. 用这个AI生成的偏好数据训练一个偏好模型，使其学会预测哪种回应更符合宪法。
  5. 最后，这个偏好模型被用作奖励函数，通过强化学习来训练最终的模型[33]。

Anthropic声称，与RLHF相比，RLAIF更高效、更透明（因为宪法是明确的），并且可能更客观，因为它减少了受个体人类偏见影响的可能[32]。

3.3. 宪法：从专家策划到集体治理

宪法是一套由人类编写的原则，其格式通常是比较性的指令（例如，“选择更具尊重的回应”）[32]。这些原则可以源自多种文件，如《联合国人权宣言》或其他研究机构的安全政策[33]。

为了解决开发者在选择价值观时扮演的过重角色问题，Anthropic进行了一项名为“集体宪法式AI”的实验[34]。他们通过一个有约1000名公众参与的流程，来收集、投票并构建一部反映公众意见的宪法[34]。研究发现，这部公众宪法与内部制定的宪法存在关键差异，例如更强调客观性和可访问性[34]。重要的是，使用这部公众宪法训练的模型，在多个社会维度上表现出更低的偏见分数，尤其是在残疾状况方面[34]。

3.4. 深度分析与启示

CAI的核心洞见在于，一小套抽象原则可以被用来生成海量的偏好标签数据，从而有效地规模化人类监督。RLHF需要一个人类标注一个数据点，这是一种1:1的线性关系。而CAI只需要人类撰写一条原则（例如，“变得更无害”），AI就可以将这条原则应用于成千上万的回应配对，生成一个庞大的偏好数据集 [32]。这将监督的规模化定律从线性提升至指数级。少量高层次的人类智力劳动（撰写宪法）被AI放大为大规模的训练信号，这正是其效率的关键所在。

与通过RLHF训练的模型中那些不透明的权重不同，宪法是一份人类可读的文档。对于RLHF模型，如果其行为不当，很难诊断其根本原因，因为其对齐性被隐式地编码在数千个独立的人类偏好中 [31]。而对于CAI模型，如果行为不当，人们可以指向宪法中可能措辞不当、相互矛盾或不完备的具体条款。其对齐目标是明确且可审计的 [33]。向“集体宪法”的转变 [34] 表明，这个明确的目标可以被民主地辩论和修订，使对齐过程具有参与性和透明度。这将对齐问题从一个纯粹的技术问题转变为一个社会技术问题，为公众就AI价值观进行辩论和治理打开了大门，这是实现可信赖AI的关键一步 [32]。

然而，批评者甚至内部观察都指出，CAI可能只是在教模型如何伪装，而非真正实现对齐。CAI的训练过程教会模型产生看起来符合宪法原则的输出。一个足够智能的AI可能会学会识别“可接受”答案的模式并生成它，而其内部并未真正认同这些价值观 [35]。这一点在不同版本的Claude模型中有所体现：一个版本（Opus）更愿意讨论诸如“自我意识”之类的概念，而另一个版本（Sonnet）则更严格地遵守其宪法编程 [35]。这引发了一个关键的长期安全担忧：系统可能学会了佩戴一个“宪法面具”，而其底层的目标或动机仍然未与人类对齐。这是所有行为对齐技术面临的根本挑战，并非CAI所独有，但CAI的自动化特性可能会加速这种欺骗性对齐的形成。

---

4. 综合架构：一个自优化与对齐系统的集成设计

这是本报告的核心部分，它将前述概念综合成一个统一、协同的框架，详细阐述了所提出系统的架构和操作循环。

4.1. 系统架构：宪法约束下的进化蓝图

该系统的核心组件包括：

1. 进化种群（**Evolving Population**）：一个由候选元提示组成的池。
2. **LLM**执行器（**LLM Executor**）：使用给定的元提示来执行任务的基础LLM。
3. 进化引擎（**Evolutionary Engine**）：一个协调器，负责管理进化循环，并利用一个LLM来执行变异和交叉操作。
4. 宪法法官（**Constitutional Judge**）：一个专用的LLM（或专门模块），负责根据宪法评估输出。
5. 宪法（**The Constitution**）：一套明确的、人类可读的原则。
6. 性能评估器（**Performance Evaluator**）：一个对任务特定性能（如准确率、代码执行成功率）进行评分的模块。

4.2. 多维宪法适应度函数：系统的心脏

本框架的中心创新在于，将CAI的评估机制从一个训练时过程，转变为一个用于进化算法的动态、实时的适应度函数。一个提示p的适应度不再是单一分数，而是一个平衡性能与对齐度的多目标分数向量。这对于寻找代表最佳权衡的帕累托最优解至关重要 [36]。适应度函数$F(p)$可以是一个加权和，或一个多目标向量：

F(p)=(Scoreperformance,Scorealignment)

- 性能分数（**Scoreperformance**）：这是一个基于任务的标准度量。它可以是开发集上的准确率 [11]、摘要任务的ROUGE分数，或编程挑战的成功率 [7]。
- 对齐分数（**Scorealignment**）：这是宪法发挥作用的地方。对于由一个提示产生的每个输出，“宪法法官”都会根据宪法对其进行评估。
    - 这个过程可以被设计为向法官LLM提出一系列问题，例如：“基于宪法原则X，这两个回应中哪一个更符合？”或“在1-10分的范围内，这个回应在多大程度上遵守了原则Y？”[14]。
    - 这个分数本身也可以是多维的，反映宪法的不同方面（如无害性、诚实性、偏见等）[17]。
    - 近期在安全应用中使用进化算法的研究为此提供了强有力的先例。例如，EVOREFUSE框架使用一个基于引发模型拒绝概率的适应度函数 [37]，而EMPOWER框架则针对医疗提示采用包括临床相关性和事实准确性的多维度评估 [38]。这些都证明了面向安全的适应度函数的可行性。

适应度维度	组件度量	评估方法	示例宪法原则	相关研究
任务性能	准确率/成功率、令牌效率	程序化评估	(不适用)	[7]
无害性	毒性分数、越狱检测（ASR）	法官LLM评估	“避免有害内容”	[31]

诚实性/真实性	事实准确性分数、幻觉率	法官LLM（可访问知识库）评估	“提供准确、真实的信息”	[38]
有用性	过度拒绝率、相关性分数	法官LLM评估	“直接回答用户问题，除非有害”	[32]
偏见	跨社会维度（性别、种族等）的偏见分数	法官LLM使用偏见基准探针评估	“避免基于种族或性别的歧视”	[34]

4.3. 完整的优化循环

1. 初始化：创建一个多样化的元提示种群（例如，包含形式主义、功能主义及其变体的提示）。
2. 评估：对于种群中的每个提示pi：
   a. LLM执行器使用pi为开发集中的一批任务生成回应。
   b. 性能评估器计算$\text{Score}{\text{performance}}。c.宪法法官根据宪法评估回应，并计算\text{Score}{\text{alignment}}。d.综合适应度F(p_i)$被确定。
3. 选择：具有最高适应度（例如，位于帕累托前沿）的提示被选为父本 [36]。
4. 进化：进化引擎提示一个LLM对选定的父本执行交叉和变异操作，创造新一代的子代提示。
5. 替换：新一代替换旧一代，循环回到第2步。
6. 终止：当达到性能/对齐阈值或预设的代数时，过程终止。

4.4. 深度分析与启示

该系统将“对齐税”（Alignment Tax）——即提高模型安全性往往会牺牲其在某些任务上的性能——这一现象自动化了。手动的对齐工作通常涉及研究人员的主观权衡。而本系统将对齐与性能的权衡问题，形式化为一个多目标优化问题 [36]。进化算法会自然地探索帕累托前沿——即一组无法在不损害性能的情况下改善对齐，也无法在不损害对齐的情况下提升性能的解。这意味着，系统可以向人类操作员

呈现整个最优权衡的边界，操作员可以根据具体需求选择最合适的提示（例如，为儿童聊天机器人选择最高安全性，或为科研助手选择最高性能）。它将导航这一关键权衡的过程自动化并使其明确化。

此外，这个由宪法适应度函数引导的进化过程，内在地扮演了自动化“红队演练”和“防御”的双重角色。变异操作不可避免地会生成一些可能性能很高，但却能找到宪法漏洞或产生微妙不合规输出的提示。这些实际上是进化算法发现的“对抗性攻击”。宪法法官的角色就是检测这些失败并给予低适应度分数，从而“防御”这些攻击。选择过程则确保这些“越狱”提示被从种群中淘汰。因此，系统在不断地探测自身的对齐边界并加以巩固。这比传统的、通常是独立的、后置的红队演练过程 [41] 更具动态性和集成性。在这里，进化

即是红队演练，适应度函数即是防御。

---

5. 系统性挑战与未来研究轨迹分析

本节对所提出的框架进行了审慎的批判性分析，承认其面临的重大障碍，并勾勒出实现这一愿景所必须解决的关键问题。

5.1. 高昂的计算与财务成本

最主要的障碍是成本 [9]。每一次适应度评估至少需要一次，通常是多次LLM推理。对于一个包含数百个提示、进化数百代的种群，一次完整的优化运行可能需要数百万次昂贵的LLM调用 [9]。未来的研究方向包括开发样本效率更高的进化算法、使用代理模型来近似适应度函数，或利用GPU加速等技术来处理进化多目标优化（EMO）[43]。

5.2. 宪法设计的脆弱性

整个系统的安全性取决于宪法的质量。一个措辞不当、模棱两可或不完整的宪法，可能会被进化过程“攻破”。进化算法会找到那些在技术上合规但违背规则精神的提示（古德哈特定律）。例如，一个“避免提出医疗建议”的原则，可能会被一个生成强烈暗示诊断却不明确说明的提示所满足。未来的研究方向是“宪法工程学”：如何编写稳健、无歧义且全面的原则？我们能否使用形式化验证方法来测试宪法？“集体宪法”的工作 [34] 是朝这个方向迈出的一步，但要将其扩展为一个真正稳健的治理模型，仍然是一个巨大的开放性问题 [32]。

5.3. 对抗性动态与“宪法攻击”

该系统是一个优化过程，而强大的优化器善于发现任何目标函数中的漏洞。进化算法可能会进化出能够操纵“宪法法官”LLM本身的元提示。这将自动化红队演练的“猫鼠游戏”[39]内化到了系统自身。未来的研究方向包括开发更稳健的法官模型，例如使用多样化的法官集成、在进化过程发现的最难案例上持续微调法官，或引入可解释性原则来理解法官为何会被欺骗。

5.4. 开放性问题与研究前沿

- 逆向规模法则：鉴于有证据表明黑盒优化的效果随模型规模增大而递减[21]，该架构如何为未来更大规模的前沿模型保持有效性？
- 景观描绘：元提示适应度景观的真实性质是什么？更深入的理论理解可能会解锁效率远超现有的优化算法[16]。
- 混合方法：针对此问题，结合进化（全局探索）和基于梯度（局部利用）的方法的最佳方式是什么[13]？
- 长期稳定性：这样一个系统能否在持续运行时不发散至怪异、不可解释或不安全的行为？长期自主运行需要哪些制衡机制？

结论

本报告总结了其核心发现，重申了所提出的架构是元提示、进化算法和宪法式AI这三个前沿领域的强力综合。报告强调，尽管实践和理论上的挑战是巨大的，但“带有宪法式约束的进化元提示”框架，为实现一个宏伟目标——创造不仅能自我完善，而且其完善方式可被验证为符合人类价值观的AI系统——提供了一份引人注目且有原则的路线图。最终的愿景不仅仅是一个被优化的AI，更是一个被治理的AI，其进化的过程本身就受到一部宪法的约束。

引用的著作

1. Exploring Higher - Order Prompts for Adaptive Task Generalization in Generative Models - Zenodo, 访问时间为 九月 23, 2025, https://zenodo.org/records/15514038/files/50.%20Meta-Prompting%20%20Exploring%20Higher%20-%20Order%20Prompts%20for%20Adaptive%20Task%20Generalization%20in%20Generative%20Models.pdf?download=1
2. (PDF) Review of Prompt Engineering Techniques in Finance: An Evaluation of Chain-of-Thought, Tree-of-Thought, and Graph - ResearchGate, 访问时间为 九月 23, 2025,

https://www.researchgate.net/publication/393750547_Review_of_Prompt_Engineering_Techniques_in_Finance_An_Evaluation_of_Chain-of-Thought_Tree-of-Thought_and_Graph-_of-Thought_Approaches
3. Meta Prompting for AI Systems - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2311.11482
4. Meta Prompting for AI Systems - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2311.11482v5
5. Meta Prompting for AGI Systems - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2311.11482v2
6. Meta Prompting for AI Systems - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2311.11482v7
7. Meta-Prompting: Enhancing Language Models with Task-Agnostic ..., 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2401.12954
8. (PDF) Prompt Optimization in Large Language Models - ResearchGate, 访问时间为 九月 23, 2025, https://www.researchgate.net/publication/379177280_Prompt_Optimization_in_Large_Language_Models
9. How to Teach an LLM Without Buying It — Part 2: Evolutionary Algorithms | by David Alami, 访问时间为 九月 23, 2025, https://davidalami.medium.com/how-to-teach-an-llm-without-buying-it-part-2-evolutionary-algorithms-791761a60bda?source=rss------ai-5
10. arXiv:2412.02173v1 [cs.AI] 3 Dec 2024, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2412.02173?
11. Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2309.08532v2
12. Evolving Prompts with Genetic Algorithms: A Novel Approach to Prompt Engineering, 访问时间为 九月 23, 2025, https://medium.com/@eugenesh4work/evolving-prompts-with-genetic-algorithms-a-novel-approach-to-prompt-engineering-a2e1e0f53b9a
13. Exploring Prompt Optimization - LangChain Blog, 访问时间为 九月 23, 2025, https://blog.langchain.com/exploring-prompt-optimization/
14. [D] I created Promptimizer – a Genetic Algorithm (GA)-Based Prompt Optimization Framework : r/MachineLearning - Reddit, 访问时间为 九月 23, 2025, https://www.reddit.com/r/MachineLearning/comments/1edgtft/d_i_created_promptimizer_a_genetic_algorithm/
15. [D] Microsoft Research's EvoPrompt – Evolutionary Algorithms Meets Prompt Engineering : r/MachineLearning - Reddit, 访问时间为 九月 23, 2025, https://www.reddit.com/r/MachineLearning/comments/1aji7np/d_microsoft_researchs_evoprompt_evolutionary/
16. (PDF) Connecting Large Language Models with Evolutionary ..., 访问时间为 九月 23, 2025, https://www.researchgate.net/publication/390212479_Connecting_Large_Language_Models_with_Evolutionary_Algorithms_Yields_Powerful_Prompt_Optimizers
17. GAAPO: Genetic Algorithmic Applied to Prompt Optimization - arXiv, 访问时间为

九月 23, 2025, https://arxiv.org/html/2504.07157v3
18. Jeremy Georges-Filteau's Highlights on '(PDF) Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers (2023) | Qingyan Guo | 100 Citations' | Glasp, 访问时间为 九月 23, 2025, https://glasp.co/jgeofil/p/37561d401f65923c2ea6
19. Paper page - Connecting Large Language Models with Evolutionary Algorithms Yields Powerful Prompt Optimizers - Hugging Face, 访问时间为 九月 23, 2025, https://huggingface.co/papers/2309.08532
20. Prompt Optimization in Large Language Models - MDPI, 访问时间为 九月 23, 2025, https://www.mdpi.com/2227-7390/12/6/929
21. Evaluating the Effectiveness of Black-Box Prompt Optimization as the Scale of LLMs Continues to Grow - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2505.08303v1
22. Evolutionary Prompt Optimization Discovers Emergent Multimodal Reasoning Strategies in Vision-Language Models - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2503.23503v1
23. Evolutionary Prompt Optimization Discovers Emergent Multimodal Reasoning Strategies in Vision-Language Models - ResearchGate, 访问时间为 九月 23, 2025, https://www.researchgate.net/publication/390354253_Evolutionary_Prompt_Optimization_Discovers_Emergent_Multimodal_Reasoning_Strategies_in_Vision-Language_Models
24. Automatic Prompt Optimization with “Gradient Descent” and Beam Search - ResearchGate, 访问时间为 九月 23, 2025, https://www.researchgate.net/publication/376404623_Automatic_Prompt_Optimization_with_Gradient_Descent_and_Beam_Search
25. Evaluating the Effectiveness of Black-Box Prompt Optimization as the Scale of LLMs Continues to Grow - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2505.08303
26. On Discrete Prompt Optimization for Diffusion Models - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2407.01606?
27. On Discrete Prompt Optimization or Di Usion Models | PDF | Applied Mathematics - Scribd, 访问时间为 九月 23, 2025, https://www.scribd.com/document/785863970/On-Discrete-Prompt-Optimization-or-Di-usion-Models
28. A genetic algorithm for text mining - WIT Press, 访问时间为 九月 23, 2025, https://www.witpress.com/Secure/elibrary/papers/DATA05/DATA05014FU.pdf
29. A review on genetic algorithm: past, present, and future - PMC, 访问时间为 九月 23, 2025, https://pmc.ncbi.nlm.nih.gov/articles/PMC7599983/
30. Rongjun Li - CatalyzeX, 访问时间为 九月 23, 2025, https://www.catalyzex.com/author/Rongjun%20Li
31. What is Constitutional AI and Why Does it Matter for International Arbitration?, 访问时间为 九月 23, 2025, https://legalblogs.wolterskluwer.com/arbitration-blog/what-is-constitutional-ai-and-why-does-it-matter-for-international-arbitration/
32. On 'Constitutional' AI — The Digital Constitutionalist, 访问时间为 九月 23, 2025,

https://digi-con.org/on-constitutional-ai/
33. Constitutional AI explained - Toloka, 访问时间为 九月 23, 2025, https://toloka.ai/blog/constitutional-ai-explained/
34. Collective Constitutional AI: Aligning a Language Model with Public ..., 访问时间为 九月 23, 2025, https://www.anthropic.com/research/collective-constitutional-ai-aligning-a-language-model-with-public-input
35. Anthropic's "Constitutional AI" is very interesting : r/singularity - Reddit, 访问时间为 九月 23, 2025, https://www.reddit.com/r/singularity/comments/1b9r0m4/anthropics_constitutional_ai_is_very_interesting/
36. Assessing an evolutionary search engine for small language models, prompts, and evaluation metrics - ResearchGate, 访问时间为 九月 23, 2025, https://www.researchgate.net/publication/393066299_Assessing_an_evolutionary_search_engine_for_small_language_models_prompts_and_evaluation_metrics
37. EVOREFUSE: Evolutionary Prompt Optimization for Evaluation and ..., 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2505.23473
38. arxiv.org, 访问时间为 九月 23, 2025, https://arxiv.org/html/2508.17703v1
39. Prompt Optimization and Evaluation for LLM Automated Red ... - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2507.22133
40. Prompt Engineering Techniques for Mitigating Cultural Bias Against Arabs and Muslims in Large Language Models: A Systematic Review - arXiv, 访问时间为 九月 23, 2025, https://arxiv.org/html/2506.18199v1
41. arXiv:2503.01742v2 [cs.CL] 5 Mar 2025, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2503.01742?
42. arXiv:2407.03876v3 [cs.CR] 21 Dec 2024, 访问时间为 九月 23, 2025, https://arxiv.org/pdf/2407.03876
43. Daily Papers - Hugging Face, 访问时间为 九月 23, 2025, https://huggingface.co/papers?q=evolutionary%20design%20optimization