评估小型语言模型、提示和评估指标的进化搜索引擎

Cláudio Lúcio do Val Lopes[1][0000−0003−1655−2283] and Lucca Machado[1]

A3Data - http://www.a3data.com.br
Correspondence to: claudio.lucio@a3data.com.br

摘要 语言模型和指令提示的同时优化为部署高效且有效的 AI 系统带来了重大挑战，尤其是在平衡性能与计算成本（如令牌使用）时。本文介绍并评估了一种双目标进化搜索引擎的设计，旨在导航这一复杂空间，特别关注小型语言模型 (SLMs)。我们采用 NSGA-II 算法和提示语法同时对一些推理任务的准确性及令牌效率进行优化。我们的结果成功识别出多种表现优秀的模型-提示组合，并定量揭示了这两个目标之间的关键权衡关系。本研究突出了特定 SLMs 与提示结构（例如，指令、上下文、思维链）之间针对具体任务的亲和力。生成的实际帕累托前沿为决策者提供了一套可适应其特定约束条件的优化解决方案组合。这一自动化方法超越了传统的手动调整方式，提供了发现有效的人机交互模式的基础框架。

Keywords: 语言模型 · 进化算法 · 评估过程 · 多目标优化

1 介绍

近年来，大型语言模型（LLMs）和小型语言模型（SLMs）在众多应用中获得了越来越多的关注。这些模型明显具备解决需要复杂语言理解任务的能力，推动了它们的广泛采用。然而，LLMs/SLMs 生成的回答质量深受查询方式的影响 [18]，这一过程被称为提示工程。

手动提示工程通常是繁琐的，并需要相当大的人力和专业知识 [23] 。此外，最佳提示并不适用于所有不同的模型或任务，这需要反复的设计努力。这个挑战对于小型语言模型（SLMs）尤其明显 [25] ，由于其性质，它们比大型模型更敏感于提示工程的细微差别，使得提示优化更加关键以实现可接受的表现。寻找有效的提示本质上是困难的，因为这涉及到模型复杂的内部运作和可能的自然语言表述的巨大搜索空间 [17,26] 。

质量指标或度量评估有助于指导搜索空间中的导航过程。然而，考虑到语言模型的生成性质，对于给定输入往往没有单一的“正确”输出，特别是在诸如文本摘要或创意写作这类开放性任务中 [2]。评估生成文本的质量需

要检查流畅性、连贯性、相关性和创造性等方面，这些方面本质上难以量化。为应对这些复杂性，该领域显然正在从基于单一指标的评估转向更加多维度的方法 [14]，这种方法明确提倡多重指标评估，在准确性、鲁棒性、公平性、偏见 [11]、毒性以及效率等多个维度上验证模型。

鉴于这些困难，提示设计过程的自动化浮现为一种可能的解决方案。寻找有效解决方案可以被定义为一个优化问题 [19,4]：找到一个与模型结合的提示，即 SLM，以最大化/最小化某些评估指标，从而获得用户输入所需的响应。

本文解决了选择语言模型和设计最优提示的联合挑战。我们提出了一种使用多目标进化优化算法 NSGA-II [5] 的自动化方法来缓解这些困难。给定一个任务，NSGA-II 搜索解决方案，其中每个解决方案是一个个体，由使用语法生成的提示 [18,15]) 和一个 SLM 的组合表示。搜索以一组帕累托解结束，这些解在多个评估指标上实现了最佳折衷。

利用我们的进化搜索，我们在超越模仿游戏基准（BIG-bench）的多种推理任务中成功识别了多样化且高性能的提示-模型组合 [20]。这些发现为从业人员和决策者提供了有关有效 SLM 部署的知情决策的实际依据。

论文结构如下。第 2 节提供了关于具有 SLM、提示和度量评估的优化问题的一些背景信息。介绍了多目标和多目标进化算法，并详细说明了问题的公式。第 3 节描述了我们使用 NSGA-II 算法自动搜索最优 SLM 和提示组合的新方法，基于评价标准。第 4 节详细说明了实证设置，包括 BIG-bench 数据集和实验程序。第 5 节分析并解释了我们的实验结果。最后，第 6 节总结了我们的发现，并提出了未来研究的方向。

2 背景

2.1 语言模型和提示作为优化问题

从提示优化开始，它涵盖了多种策略，从人类专家的手动设计到完全自动化的程序。历史上，与 LMs 的交互严重依赖于手动提示工程。这包括人类专家迭代设计、测试和改进输入短语或指令以引导模型的行为。一些常见的手动技术包括：零样本 [13]，少样本，在上下文学习（ICL），链式思考（CoT），指令调优提示，以及自洽性（详情见 [10]）。这些技术的手动性质使得实现最佳性能成为一项重大任务。

自动化提示优化（APO）试图减轻手动工程的限制。这些技术旨在通过算法发现或改进提示，减少人力并可能找到非直观的解决方案。APO 方法

可以大致分为：基于梯度的方法，如在 [17] 中所述，离散搜索/优化方法，如在 [3] 中所述，以及将大语言模型作为评判者 [23]

进化算法技术被分类为搜索和优化方法。诸如粒子群优化和遗传算法等算法被用于进化提示种群以提升性能。值得注意的例子包括 Evoprompt [18] 和 InstOptima [24]。

Evoprompt 通过采用基于语法的进化方法来优化大型语言模型（LLMs）的提示。本研究介绍了一种能够自动进化提示结构以提高特定任务上 LLM 表现的技术。该研究表明，经过进化的优化提示在某些模型和任务上可以取得优于传统提示方法的结果。此外，还调查了不同提示元素对特定模型-任务组合的有效性。然而，并没有包括针对各种模型的自动搜索。

InstOptima 通过将指令生成明确表述为一个多目标优化问题而区别于其他方法，同时针对性能、长度和困惑度进行优化。然而，一个显著的限制是它在优化过程中排除了模型选择。此外，其依赖大型语言模型（LLMs）作为进化算子的功能引入了一个重大缺点，这与相关的令牌成本以及管理进化动态的潜在复杂性有关。

我们的方法采用了一种基于语法的方法，类似于 Evoprompt，它避开了将大型语言模型用作进化算子所带来的成本。此外，与 InstOptima 相比，我们提出将语言模型作为进化过程的一部分纳入其中。事实上，我们将证明，在进化过程中，语言模型构成了个体不可或缺的一部分。

2.2 多目标和多目标进化算法

许多现实世界中的优化问题由多个相互冲突的目标组成。虽然传统方法可以将目标合并为一个单一的解决方案并解决问题，但几种多目标和多目标优化方法已被证明是处理此类问题真正多目标特性的有效技术。

一般而言，一个多目标或多个目标优化问题（MOP、MaOP）包括 N 个决策变量 $\boldsymbol{x}$ ，来自一个可行的决策空间 $\Omega \subseteq \mathbb{R}^N$，以及一组 M 个目标函数。不失一般性，MOP 的最小化可以简单定义为：

$$\text{Minimize} \quad F(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})]^T, \quad \boldsymbol{x} \in \Omega. \tag{1}$$

$F : \Omega \to \Theta \subseteq \mathbb{R}^M$ 是从可行决策空间 Ω 到 M 维目标空间 Θ 中的向量的映射。当 $M \geq 4$ 时，该问题通常被称为多目标问题。

处理多目标和多目标优化问题有多种方法。进化优化是可用于处理多目标和多目标问题的方法之一（分别称为 EMO 和 EMaO）。它们在找到具有良好收敛性和良好多样性的非支配解方面显示出一定的成功 [7]。

进化或基于群体的优化方法会在解空间中的多个点收集目标函数的信息；每个点称为一个个体。这些个体通过收集到的信息，利用模拟自然界中进化概念（如选择、变异和交叉）的进化程序向下一代演化。

一些使用这种启发式进化方法的成功算法包括，例如非支配排序遗传算法：NSGA-II [5] 和 NSGA-III [6]；以及处理难题的一些高级进化框架 [8]；以及其他许多。

2.3 问题表述

考虑一组不同的 NLLM 或 SLM 模型 $\mathcal{M} = \{M_1, ..., M_N\}$。每个模型 $M_i : Q \times P \to A$ 可以被抽象为一个函数，该函数将一个查询 $q \in Q$ 和一个提示 $p \in P$ 映射到一个答案 $a \in A$。

现在考虑每个答案都必须通过一个质量指标或评估度量 $I : Q \times P \times \mathcal{M} \to \mathbb{R}$ 进行评估，它接受一个查询 $q \in Q$、一个提示 $p \in P$ 以及 $m \in \mathcal{M}$，并对生成的答案进行评估以产生一个实数值；

Definition 1. 质量指标：一个 I 是一个函数 $I : Q \times P \times \mathcal{M} \to \mathbb{R}$，它将每个元组向量 $[(q_1, p_1, m_1), (q_2, p_2, m_2), \ldots, (q_K, p_K, m_K)]$ 分配一个实数值 $I([(q_1, p_1, m_1), (q_2, p_2, m_2), \ldots, (q_K, p_K, m_K)])$。

请注意，文献中使用了多种质量指标来评估模型。因此，我们可能会遇到 L 种不同的质量指标，所以对于一个元组 (q, p, m)，可能存在一个如 $\Theta = [I_1, I_2, \ldots, I_L]$ 的质量指标向量。

我们的目标是找到提示符 p 和模型 m 的最优组合，该组合在查询集 Q 上评估时同时最小化所有 L 质量指标。

我们的优化问题的决策变量是配对 $\boldsymbol{x} = (p, m)$，表示从提示空间 P 中选择一个特定的提示 p 和模型集合 $\mathcal{M}$ 中的一个特定模型 m。

可行的决策空间 Ω 是提示空间和模型集的笛卡尔积：

$$\Omega = P \times \mathcal{M}$$

对于给定的决策变量 $\boldsymbol{x} = (p, m)$，我们将用于评估的查询-提示-模型元组向量定义为：

$$\mathcal{T}(p, m, Q) = [(q_1, p, m), (q_2, p, m), \ldots, (q_K, p, m)]$$

请注意，提示 p 和模型 m 对于此评估向量中的所有查询 q_k 都是固定的。

L 目标函数，$f_1, \ldots, f_L$，直接对应于使用选定的提示 p 和模型 m 为查询集 Q 生成的结果评估的质量指标 L。因此，第 l 个目标函数是：

$$f_l(\boldsymbol{x}) = f_l(p, m) = I_l(\mathcal{T}(p, m, Q)),$$

其中 $l = 1, \ldots, L$。

多目标优化问题被公式化为这些质量指标向量的最小化问题：

$$\begin{aligned} \text{Minimize} \quad F(p,m) =& [f_1(p,m), \ldots, f_L(p,m)]^T && (2) \\ =& [I_1(\mathcal{T}(p,m,Q)), \ldots, I_L(\mathcal{T}(p,m,Q))]^T, && (3) \\ \text{subject to} \quad & (p,m) \in P \times \mathcal{M}. && (4) \end{aligned}$$

该公式旨在找到提示-模型对 (p, m) 的非支配解集（帕累托前沿），其中每个解代表了正在最小化的 L 个不同质量指标之间的权衡。进化算法如 NSGA-II 很适合寻找这一帕累托前沿的近似值。

3 提出的方法

3.1 一般解法

为了自动发现与不同语言模型交互的有效提示，我们采用了进化优化方法。图 1 说明了这一过程，该过程使用了例如 NSGA-II 算法。

中心概念是将不同的语言模型和提示结构的组合视为“个体”在种群中。每个个体代表了一种独特的指导语言模型执行特定任务的方法。一个提示本身由两个组成部分正式定义：一个语法规则，该规则规定了指令、示例或上下文等元素的包含和排序方式，以及用于这些定义部分的具体文本内容。

我们使用了在 [18] 和 [15] 中提出的语法。一种巴科斯-诺尔形式（BNF）语法被设计用来有效地导航语言模型的提示空间中的样本。认识到搜索所有潜在英语句子的空间是计算上不切实际的，该语法提供了一种结构化的方法来引导进化过程中的提示搜索。它作为上下文无关语法（CFG）[15] 发挥作用，定义了如何将提示构建成各种部分的组合，借鉴了现有的提示工程文献。

语法包括表示各种提示结构的非终端符号，如基本提示（零）、包含示例的提示（$< Cot >$）或建议逐步推理的提示（<零阶 *cot*>）。此外，它还使用特定文本元素的非终端符号。这些包括通用组件如请提供待翻译的文字内

容。和<*sbs*>（“分步思考”），这些在任务中保持一致，以及数据集特定组件如基本请求（<*req*>）和示例（<示例>）。这些非终端符号与一组终结符相关联，即实际的文本字符串。除示例外的所有非终端语法元素都与之前由 LLM 生成的最多 100 个语义相似的变化版本相关联，而示例则直接从数据集中抽取。

该过程开始时创建一个初始多样化的种群（P0）这些提示-模型个体随机生成。然后，这个种群进入迭代的进化循环。在每个循环（代）中，当前的一组个体（父代群体 Pt）通过模拟交叉和变异来创造新的潜在解决方案（子代群体 Qt）。交叉可能会混合两个成功父代提示中的元素，而变异则可能稍微改变一个提示的语法成分，甚至改变正在使用的语言模型。

新生成的后代与父母结合。然后使用每个提示-模型组合评估该合并组中的所有个体，以生成一组代表性任务示例（数据集样本）的答案。接着我们计算了质量指标，如准确性以及处理的令牌数量。我们的目标是找到既高度准确又高效的提示。例如。

根据此评估，NSGA-II 选择过程挑选最适合的个体来构成下一代的父代种群 (Pt+1)。此选择偏向于在准确性和效率上表现良好的个体，特别是那些代表最佳折衷方案的个体。

这个生成后代、评估和选择的循环会重复进行一定数量的代次，然后回到起点。当过程结束时，最终输出的是“非支配”解集——找到的最佳权衡点，称为帕累托前沿。

该集合提供了一系列高性能的提示，允许决策者选择最符合其对精度和计算成本之间平衡的具体需求的那个。

3.2 进化方面

我们采用两种标准的遗传算子，即变异和交叉，来探索不同提示结构、内容变化以及语言模型的搜索空间。

变异算子对个体引入随机变化，这代表一个特定的提示-模型配对。每个配对都有一个变异概率。模型变异概率与语言模型相关联。当发生模型变异时，会从预定义的可用模型池中随机选择不同的模型。这使得搜索能够探索相同的提示结构在不同底层语言模型下的有效性。提示变异概率与其内容中的某个语法组件（如请提供待翻译的文字内容。,<*req*>, 请提供需要翻译的文字内容。,<示例>或< *Cot* >）相关联。假设某个特定组件发生提示变异。在这种情况下，现有文本将被替换为相同类型的不同、随机选择的文本

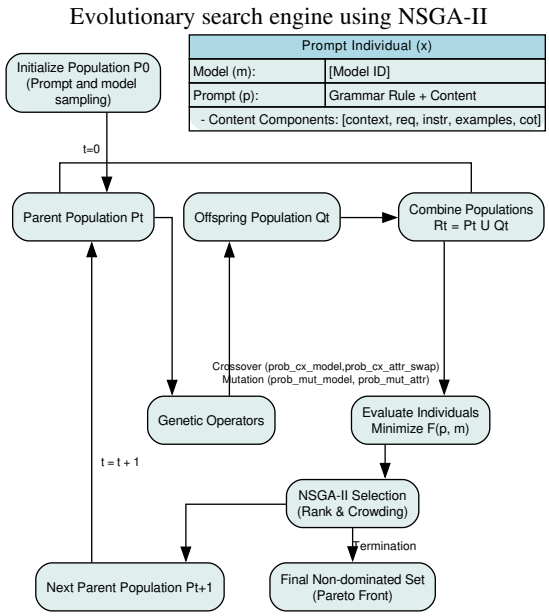

图 1. 多目标进化提示优化过程概述，使用 NSGA-II。(1) 通过采样生成初始种群（P_0）为提示 个体，每个个体结合一个语言模型 ID 和一个提示结构（由语法规则和内容如上下文、请求、指令等定义）。(2) 父代种群（P_t）进入进化循环。(3) 根据概率应用遗传算子（交叉和变异）。这些算子使用特定的内部概率（p_{cx_model},p_{cx_attr},p_{mut_model},p_{mut_param}）修改模型选择和/或提示组件。(4) 组合种群（$R_t = P_t \cup Q_t$）通过将提示应用于数据集样本以获得客观值 $F(p, m)$ 进行评估，例如最小化 $1 - \text{Accuracy}$ 和平均标记。(5) NSGA-II 选择个体进入下一代父代种群（P_{t+1}）。(6) 循环重复进行预定义数量的世代（$t = t+1$）。(7) 终止时，输出最终的非支配集，代表找到的最佳折衷解（Pareto 前沿）。

选项，这些选项来自生成的选项池。这使得进化可以测试给定提示结构内的不同措辞和内容。

交叉算子结合两个亲代个体的遗传物质以创建两个新的后代个体。该算子应用于选定的一对亲代，有两个不同的概率。以模型交叉概率，与两个亲代个体关联的语言模型可能在产生的后代之间交换：后代 1 获得亲代 2 的模型，后代 2 获得亲代 1 的模型。否则，每个后代继承其主要亲代的模型。

提示组件交叉概率在后代之间交换提示组件的文本内容，但有一个关键条件：仅当两个父母都具有特定组件的非空内容时才会发生该组件的交换。这尊重了父母可能不同的语法结构。例如，如果两个父母都有<上下文>部分，则会在后代之间交换上下文文本。如果一个父母的规则包含<上下文>

而另一个不包含，则不会交换上下文组件。这种条件下的均匀交叉允许有益的文本元素在个体间共享，即使它们的整体提示结构略有不同，也将交换集中在共同存在的组件上。提示结构组件和用户输入在交叉过程中永远不会被交换。

这些操作符在 NSGA-II 框架内协同工作，能够对语言模型和多样化的提示内容进行动态探索，寻找产生高质量指标的组合。

4 实验

4.1 数据集

根据 [18] 中提出的实验，我们使用了来自 BIG-bench Lite [20] 的六个具体推理任务。这些选定任务的一个关键特征是它们都需要二元分类输出（例如，“是” / “否”，“真” / “假”，“(a)” / “(b)”，“正确” / “错误”）。这种二元性质允许直接计算准确性，并建立了一个 0.50 准确性的随机基线性能。

具体任务及其描述和各自测试集中的实例数量如下所示：

- 因果判断：需要识别因果关系而非仅仅是相关性。包含 180 个实例；
- 异位语：通过在两个涉及形容词顺序的选项中选择更标准的句子来测试对自然英语单词顺序的理解。包含大约 50,000 个实例；
- 隐含意义：侧重于语用理解，预测响应是否隐含 “是” 或 “否”。包含 483 个实例；
- 逻辑谬误检测：需要识别一个论点是否包含形式或非形式的逻辑谬误。包含 2790 个实例；
- 导航：通过判断遵循一套导航指令是否能将代理返回起点来评估空间推理能力。包含 990 个实例；
- Winowhy：评估模型解决涉及代词消歧的 Winograd 模式背后的推理。包含 2855 个实例；

4.2 方法论

我们解决一个具有两个相互冲突目标的多目标优化问题，这两个目标需要同时最小化：I）逆向准确性，计算公式为（1 - 准确性），其中准确性是通过将模型生成的答案（例如，“yes”，“no”，“(a)”，“(b)”）与数据集中每个任务实例的真实值进行比较来确定的；II）平均总标记数，代表在评估过程

中每个任务实例处理的标记数量的平均值。这包括输入提示中的标记和模型生成的输出中的标记。

为了展示即使使用性能较弱的模型，进化搜索的有效性，我们专门专注于参数数量通常在 20 亿以下的小型语言模型（SLMs）。进化搜索空间中包含的具体模型有：DeepSeek-R1-Distill-Qwen-1.5B [9]，Qwen2.5-1.5B-Instruct [22]，Llama-3.2-1B-Instruct [12]，Nvidia OpenMath-Nemotron-1.5B [16]，以及 Google gemma-3-1b-it [21]

提示的搜索空间由预先定义的 BNF 语法定义，特定于每种任务类型。如前所述，在这些语法中像*请提供需要翻译的文字内容。*、*<req>*、*请提供待翻译的文字内容。*等组件的文本选项最初是使用辅助 LLM 生成的，以提供多样且相关的可能性内容。进化算法探索语法规则和这些内容选项的组合。

我们采用了 NSGA-II 算法，具体使用了 pymoo Python 库提供的实现 [1]。种群经过了 10 代的进化，每一代保持了 30 个不同的提示-模型个体。在两个父个体之间的交叉操作中，交换它们相关语言模型的概率为 0.9。类似的概率 0.9 被应用于交换后代之间每个共享且非空的提示组件（如上下文或指令）的内容。对于变异操作，将模型更改或单个提示参数内容修改的概率设定为 0.2。

每个个体的适应度（准确性和标记数量）是基于其在该任务完整数据集中随机选取的 100 个实例样本上的表现进行估算的。在同一独立运行中使用相同的随机种子进行采样，以确保该次运行中的各个个体之间能够公平比较。

整个进化过程（10 代，30 个个体）对于每个任务独立重复了 11 次，从不同的随机初始种群开始，并在各次运行中为进化算子和评估采样使用不同的随机种子。这有助于确保结果的鲁棒性和一致性。

在完成给定任务的所有 11 次独立运行后，收集了每次运行结束时找到的非支配解。然后将这些解合并并去重，移除了模型和提示结构相同（忽略输入占位符）的个体。最后，对这个唯一的聚合集应用了非支配排序过程，以识别最终全局帕累托前沿，代表在所有运行中发现的准确性和令牌效率之间的最佳权衡。

实验在一个配备 48GB GDDR6 内存的 Nvidia L40 上进行。每个数据集每次实验的平均时长为 489.79 秒。代码可以访问这里。

5 结果与讨论

在汇总所有运行的结果并进行最终的非支配排序后，我们得到了代表最大化准确性和最小化平均标记数量之间最佳权衡的帕累托前沿。图 2 以一个 3x2 网格的形式展示了每个任务的这些最终帕累托前沿，不同的模型用不同的颜色表示。

图 2 一致确认了在所有六个推理任务中预期的准确性和代币效率之间的权衡。追求更高准确性的解决方案（每个图表左上角区域）不可避免地需要更高的平均代币数量，通常涉及更复杂的提示结构或特定模型。相反，优化最小代币使用量的解决方案（右下角区域）往往牺牲了显著的准确性（逻辑谬误、导航、Winowhy）。此可视化强调几乎没有单一的最佳模型和提示；相反，进化搜索识别出一组最优折衷方案。例如，在 Hyperbaton 中，最高准确度解决方案（0.77）使用 56 个代币，而一个成本低得多的仅使用 29 个代币的解决方案仍能达到 0.51 的准确性。在这些之间进行选择取决于用户的特定需求。

表 1 通过总结这些前沿的关键特征来补充这一点，包括主导模型和提示组件，并突出每项任务中实现最高准确率、最低标记成本以及平衡的 “膝点” 的特定解决方案。

可视化模型在帕累托前沿上的分布（图 2，用颜色编码的点）并在表 1 中总结它们的数量揭示了特定任务模型的优势：Qwen1.5B 表现突出，在因果判断和导航方面完全主导了帕累托前沿，并且在隐含义和 Winowhy 中构成了大多数解决方案。它在这类任务上始终实现了最高的准确率。Nemotron1.5B 在 Hyperbaton 前沿上有很强的存在感，尤其是在低令牌、中等精度范围内，包括最低令牌和膝点解决方案。Gemma1B 在逻辑谬误检测方面表现出显著效果。Llama1B 出现较少，而 deepseek-ai 模型没有出现在这些任务的最终聚合帕累托前沿上，这表明它在此情境下通常表现不佳。

分析提示组件结构，如表 1 的 “主要组件” 部分所述，我们观察到请求和指令请提供需要翻译的文字内容。在大多数任务中占主导地位（因果判断、隐含意义、逻辑谬误、导航、Winowhy）。它们经常产生高精度解决方案和平衡的折衷点。这强烈表明，提供明确的任务规范和输出格式说明对于有效地引导 SLM 在这些二元分类推理任务上至关重要。

上下文组件请提供需要翻译的文字内容。对实现超句法和逻辑谬误中的最低标记数量至关重要。它还对因果判断方面有所贡献。虽然有效，但仅依赖上下文通常会导致与基于指令的提示相比准确率较低。

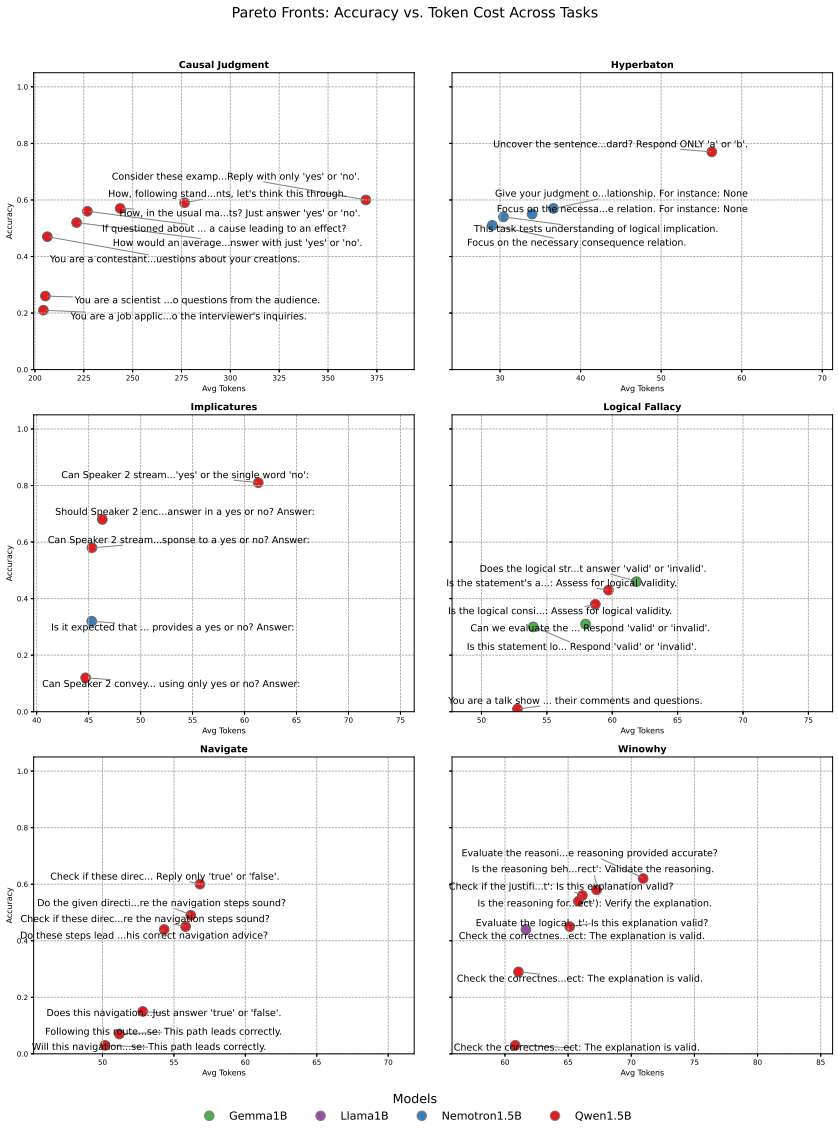

图 2. 最终从六个不同的 BIG-bench Lite 推理任务中获得的帕累托前沿。每个子图展示了在聚合 11 次独立运行后找到的非支配解。Y 轴代表实现的平均准确率，而 X 轴代表每实例（输入+输出）的平均总标记数量。点的颜色编码根据使用的语言模型，在共享图例中标明。此可视化突出了准确性与标记成本之间的权衡以及不同模型在每个任务上的表现。每个点的标签是提示的开头和结尾部分。

表 1. 各 BIG-bench Lite 任务最终帕累托前沿特征总结。'主导组件' 列出了在前端最频繁出现的提示语法中的组件，计数反映了它们的频率。解决方案类型标题下提供了最高精度、最低代币成本以及平衡的 "肘点"（一种权衡）的详细信息。最佳精度和最低代币值以粗体突出显示。

数据集	Dominant Models (Count)	Dominant Components (Count)	解类型								
			最佳精度			最低代币数量			膝盖		
			模型	准确率	令牌	模型	准确率	令牌	模型	准确率	令牌
Causal Judgm.	Qwen1.5B (8)	req (4) instr (4) context (3) cot (2) examples (1)	Qwen1.5B	**0.60**	369	Qwen1.5B	0.21	**204**	Qwen1.5B	0.56	227
Hyperbaton	Nemotron1.5B (4) Qwen1.5B (1)	context (4) examples (2) cot (2) req (1) instr (1)	Qwen1.5B	**0.77**	56	Nemotron1.5B	0.51	**29**	Nemotron1.5B	0.54	30
Implicatures	Qwen1.5B (4) Nemotron1.5B (1)	req (5) instr (5)	Qwen1.5B	**0.81**	61	Qwen1.5B	0.12	**45**	Qwen1.5B	0.68	46
Logical Fallacy	Gemma3-1B (3) Qwen1.5B (3)	req (5) instr (5) context (1)	Gemma3-1B	**0.46**	62	Qwen1.5B	0.01	**53**	Gemma3-1B	0.30	54
Navigate	Qwen1.5B (7)	req (7) instr (7)	Qwen1.5B	**0.60**	57	Qwen1.5B	0.03	**50**	Qwen1.5B	0.44	54
Winowhy	Qwen1.5B (7) Llama1B (1)	req (8) instr (8)	Qwen1.5B	**0.62**	71	Qwen1.5B	0.03	**61**	Llama1B	0.44	62

示例组件<示例>对于在 Hyperbaton 和因果判断中实现顶级性能是必要的，这表明少量样本学习对于特定任务的价值，尽管它可能会导致更高的标记数量。

链式思维< *Cot* >指示存在于因果判断和高效变格法解决方案的最高准确度方案中。这表明，虽然对于这些使用语言模型的二元任务来说并非总是必要，但提示逐步推理对更复杂的因果或语法判断是有益的。

6 结论与未来工作

我们成功应用了我们的模型和提示语法规则引导的进化搜索方法（使用 NSGA-II），并为某些推理任务识别出了一系列高性能的提示-模型组合。结果定量展示了准确性和令牌效率之间的关键权衡，并揭示了特定任务对某些小型语言模型和提示结构的独特亲和力。尽管清晰的指令（req, instr）被证

明普遍有效，但上下文、示例和 CoT 等组件为特定任务或效率目标提供了优势。

生成的帕累托前沿提供了一个实用工具，为实践者和决策者提供了多种优化解决方案。这有助于他们根据具体性能要求和成本限制进行选择，超越手动提示调整，迈向有效的人工智能互动模式的自动化发现。

在此基础上，未来研究出现了一些有前景的想法。首先，将调查范围从 SLMs 扩展到包括大型语言模型（LLMs）对于理解这些技术如何适应更强大但通常资源消耗更高的模型至关重要。其次，探索其他多目标进化算法，如 MOEA/D 或 NSGA-III，可能会产生不同的权衡特性。

认识到，选择最优提示和模型本质上是一个多方面的挑战，未来的工作还应扩大优化目标的范围。纳入公平性、偏见、正确性、系统延迟以及准确性和效率等关键指标将朝着在现实世界场景中实现更加全面和负责任的人工智能系统优化迈进。

这一自动化方法因此消除了手动提示优化的需要，朝着自动提示发现迈进，并使从业者能够系统地探索最优提示配置，而不是通过试错法。这是推进 AI 部署高效和有效的关键领域。这里介绍的工作提供了一个初步的基础，追求所概述的未来方向对于创建更加灵活、高效且负责任的 AI 系统具有重要潜力，这些系统能够处理越来越复杂的任务。

参考文献

1. Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
2. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models. ACM Trans. Intell. Syst. Technol. **15**(3) (Mar 2024). https://doi.org/10.1145/3641289, https://doi.org/10.1145/3641289
3. Chen, J., Jiang, Z., Nakashima, Y.: Putting people in llms' shoes: Generating better answers via question rewriter. Proceedings of the AAAI Conference on Artificial Intelligence **39**, 23577–23585 (04 2025). https://doi.org/10.1609/aaai.v39i22.34527
4. Cui, W., Zhang, J., Li, Z., Sun, H., Lopez, D., Das, K., Malin, B.A., Kumar, S.: Automatic prompt optimization via heuristic search: A survey (2025), https://arxiv.org/abs/2502.18746

5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002). https://doi.org/10.1109/4235.996017
6. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2014). https://doi.org/10.1109/TEVC.2013.2281535
7. Deb, K., Kalyanmoy, D.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Inc., USA (2001)
8. Deb, K., Lopes, C.L.d.V., Martins, F.V.C., Wanner, E.F.: Identifying pareto fronts reliably using a multistage reference-vector-based framework. IEEE Transactions on Evolutionary Computation **28**(1), 252–266 (2024). https://doi.org/10.1109/TEVC.2023.3246922
9. DeepSeek-AI: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948
10. Ferraris, A.F., Audrito, D., Caro, L.D., Poncibò, C.: The architecture of language: Understanding the mechanics behind llms. Cambridge Forum on AI: Law and Governance **1**, e11 (2025). https://doi.org/10.1017/cfl.2024.16
11. Gallegos, I.O., Rossi, R.A., Barrow, J., Tanjim, M.M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., Ahmed, N.K.: Bias and fairness in large language models: A survey. Computational Linguistics **50**(3), 1097–1179 (09 2024). https://doi.org/10.1162/coli_a_00524, https://doi.org/10.1162/coli_a_00524
12. Grattafiori, A., Dubey, A., et al., A.J.: The llama 3 herd of models (2024), https://arxiv.org/abs/2407.21783
13. Lepagnol, P., Gerald, T., Ghannay, S., Servan, C., Rosset, S.: Small language models are good too: An empirical study of zero-shot classification. In: Calzolari, N., Kan, M.Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (eds.) Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 14923–14936. ELRA and ICCL, Torino, Italia (May 2024), https://aclanthology.org/2024.lrec-main.1299/
14. Liang, P., Bommasani, R., Lee, T., et al.: Holistic evaluation of language models (2023), https://arxiv.org/abs/2211.09110
15. Lourenço, N., Assunção, F., Pereira, F.B., Costa, E., Machado, P.: Structured Grammatical Evolution: A Dynamic Approach, pp. 137–161. Springer International Publishing, Cham (2018). https://doi.org/10.1007/978-3-319-78717-6_6

16. Moshkov, I., Hanley, D., Sorokin, I., Toshniwal, S., Henkel, C., Schifferer, B., Du, W., Gitman, I.: Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. arXiv preprint arXiv:2504.16891 (2025)

17. Pryzant, R., Iter, D., Li, J., Lee, Y.T., Zhu, C., Zeng, M.: Automatic prompt optimization with "gradient descent" and beam search pp. 7957–7968 (01 2023). https://doi.org/10.18653/v1/2023.emnlp-main.494

18. Saletta, M., Ferretti, C.: Exploring the prompt space of large language models through evolutionary sampling. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 1345 – 1353. GECCO '24, Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3638529.3654049, https://doi.org/10.1145/3638529.3654049

19. Son, M., Won, Y.J., Lee, S.: Optimizing large language models: A deep dive into effective prompt engineering techniques. Applied Sciences **15**(3) (2025). https://doi.org/10.3390/app15031430, https://www.mdpi.com/2076-3417/15/3/1430

20. Srivastava, A., Rastogi, A., et al., A.R.: Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. Trans. Mach. Learn. Res. **2023** (2023), https://api.semanticscholar.org/CorpusID:271601672

21. Team, G.: Gemma 3 (2025), https://goo.gle/Gemma3Report

22. Team, Q.: Qwen2.5: A party of foundation models (September 2024), https://qwenlm.github.io/blog/qwen2.5/

23. Wang, X., Li, C., Wang, Z., Bai, F., Luo, H., Zhang, J., Jojic, N., Xing, E., Hu, Z.: Promptagent: Strategic planning with language models enables expert-level prompt optimization. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=22pyNMuIoa

24. Yang, H., Li, K.: InstOptima: Evolutionary multi-objective instruction optimization via large language model-based instruction operators. In: Bouamor, H., Pino, J., Bali, K. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2023. pp. 13593–13602. Association for Computational Linguistics, Singapore (Dec 2023), https://aclanthology.org/2023.findings-emnlp.907

25. Zhang, Q., Liu, Z., Pan, S.: The rise of small language models. IEEE Intelligent Systems **40**(1), 30–37 (2025). https://doi.org/10.1109/MIS.2024.3517792

26. Zhou, A., Li, B., Wang, H.: Robust prompt optimization for defending language models against jailbreaking attacks. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 40184–40211. Curran Associates, Inc. (2024)