arXiv:2611.31938v1 [cs.AI] 24 Nov 2026

CAPE：工具使用策略的校准与搜索协同

Jiahao Wang、Xiaoxuan Chen、Aijun Yang、Xiaohua Wang，电气绝缘与电力设备国家重点实验室，西安交通大学

摘要

即使工具背后的操作本身正确，工具调用仍可能失败，因为接收工具会依据智能体无法查看的配置来解析分隔符、日期或记录布局。为此类工具编译工具使用策略，需要同时进行校准（在候选配置上评估候选调用）和对已评估调用进行搜索。两者共享同一编译预算：先校准所有内容会使改进后的策略迟迟不能产出，而每当模型部分已知时就重新求解，又会拖延下一次有用的查询。只要某次调用在所有能够到达它的配置上的效果已知，该调用就可以使用。我们提出 *CAPE*，这是一种基于这一观察构建的免训练编译器。它始终维护一项可执行策略，将校准重点放在能够改进该策略的调用上，并且只在收紧界限能够提前结束校准时才收紧界限。返回的策略会保留初始策略能够完成的每一种配置；当可执行值与乐观值相等时，*CAPE* 即可证明策略最优。在由 *64* 个请求构成的 *CSV* 和 *JSON* 工作流上，*CAPE* 在 *504* 次编译中的 *502* 次达到了独立计算出的最优解，且编译时间比急切精确搜索少 *22.0%*。在规模更大的 *CSV* 系列上，*CAPE* 能以相近的编译时间返回完成率更高的策略；与根据错误反馈修改序列化方式的大语言模型相比，其完成率提高了 *15.38–34.66* 个百分点。

1. 引言

使用工具的智能体通过向外部系统发送调用来执行操作，而即使调用背后的操作本身是正确的，调用也可能失败。设想一个智能体将天气表上传到数据摄取工具。预期使用分号的解析器会拒绝逗号分隔文件，而预期使用日优先日期格式的解析器会将 05/04 读取为4月5日，而不是5月4日。解析器的选项在工具部署时设定，智能体只能看到解析后的结果或错误。因此，在每种此类配置下完成任务需要一个策略：在一次执行允许的有限调用次数内，根据每次响应确定下一次要发送的调用。当候选配置可以在测试环境中复现时，这一策略可以在部署前编译。

这类编译器结合了以往研究分别探讨的两项活动：像 ReAct 和 CodeAct 那样根据工具反馈采取行动 [20, 22]，以及像 LazySP、LRA* 和 GLS 那样在评估成本高昂时进行搜索 [2, 13, 14]。对于工具策略，校准是指在候选配置上评估候选调用并记录其效果；搜索则将记录的效果汇总起来，构建出以最少调用次数完成最多配置的策略。

校准和搜索共享同一编译预算，二者都无法单独衡量进度。在搜索之前先评估每个配置上的每次调用，会把校准预算耗费在策略永远不会触及的配置上（图1，上），而每次查询后都重新求解部分已知模型，则会延误下一个有用效果的获取。工具策略使这种耦合更加紧密：未经评估的调用不仅结果未知，对策略分支的影响也未知，因为乐观模型认为彼此不同的配置可能返回相同的响应，此时它们必须共用同一个后续执行。编译器必须决定接下来要学习什么、计算什么，以及现在可以返回什么。

策略只需要了解自身执行路径上的效果。在图中，2，一次被拒绝的上传排除四种配置中的一种，随后只需一次探测即可区分剩余三种配置，并将每种配置引导至兼容的程序。探测只需了解这三种配置上的效果：只要已知某次调用在所有能到达该调用的配置下的效果，该调用就可以投入使用。因此，已知的局部效应可得出可执行策略和可达值的下界；对未知效果作出乐观假设可得出上界，而每个新的效果都能就地改进策略或缩小上下界之间的差距。

我们提出 CAPE，这是一种免训练编译器，会根据校准和搜索对可执行策略的贡献来分配资源。对于每一组与当前响应相符的配置，CAPE 都会保留一个由已采集效果支持的最佳策略，以及一个乐观界，用于估计尚未评估的调用还能带来多少增益。校准资源会分配给两者存在差异的调用：首先是尚未覆盖初始策略所完成配置的调用，其次是每个缺失效应带来的潜在增益最高的调用。在评估某个调用期间，CAPE 会通过双调用前瞻收紧其界，并在该调用已可供使用或不再可能优于当前策略时立即停止

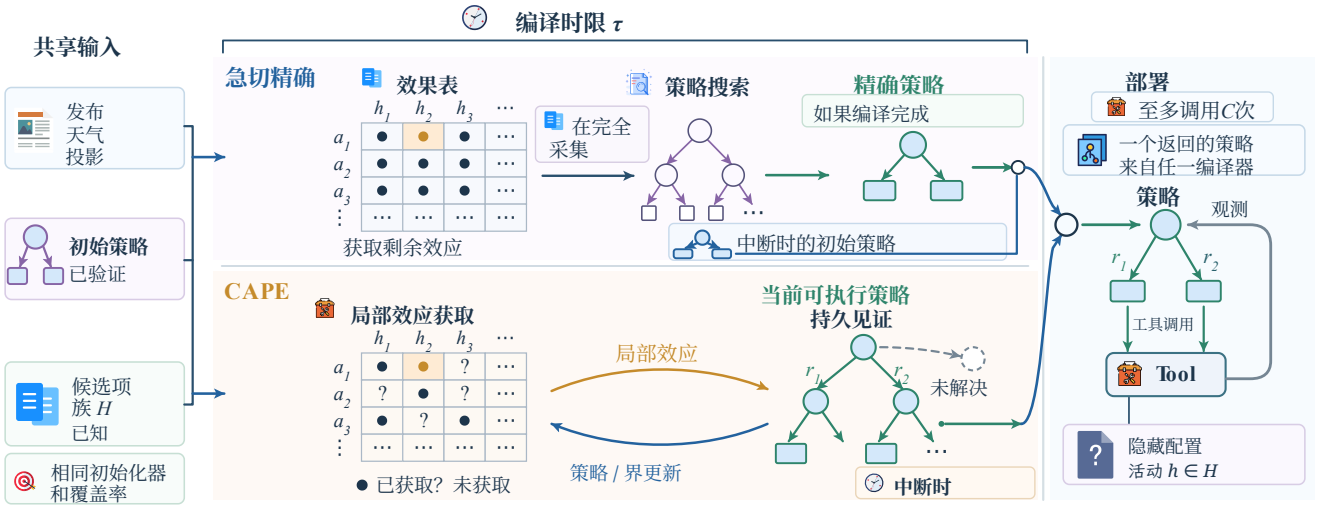

图1. **校准与搜索共享编译时限**。 急切精确搜索会在搜索前完成采集；CAPE 则在采集过程中更新可执行策略。两者都保留经过验证的初始策略；部署另有 C 次调用的限制。

。与只追随乐观解的惰性搜索不同，这一采集前沿也会追随可执行策略。每个返回的策略都会完成初始策略所完成的所有配置；当根节点处两者的值相同时，CAPE 即可证明该策略在整个动作目录上达到最优。

在由 64 个请求构成的 CSV 和 JSON 工作流中，CAPE 在 504 次编译任务中的 502 次达到了独立计算的最优解，且比急切精确搜索减少了 22.0% 的编译时间。当 CSV 系列扩展到 128 种配置时，在编译时间相近的情况下，CAPE 完成的配置数量比匹配的与/或调度器和急切精确搜索分别多 14.06 和 21.88 个百分点。在执行过程中，与在相同调用限制下根据错误反馈修订序列化内容的大语言模型相比，编译后的策略将完成率提高了 15.38–34.66 个百分点。

我们的贡献包括：分析校准与搜索如何争夺编译预算（第 3 节）、提出具备覆盖率和最优性保证的 CAPE编译器（第 4 节），以及开展匹配调度、扩展性和在线重试研究以厘清增益来源（第 5 节）。

2. 相关工作

工具使用与智能体执行。 ReAct 和 CodeAct 根据执行反馈选择每个动作 [20, 22], Re-WOO 在观察工具证据之前进行规划 [21], LLMCompiler 并行执行调用依赖图 [11],而 ADaPT 会在执行器失败时进一步分解任务 [16]。工具文档和探索性交互能在智能体采取行动前增进其对工具的了解 [4, 6, 9]。CAPE 关注的是这些决策之后的步骤：操作已选定，剩下未知的是工具所接受的表示形式。

评估代价高昂时的搜索。 LazySP 只评估乐观最短路径上的边 [2], LRA* 和 GLS 将前瞻与评估次数及图操作次数联系起来 [13, 14], , 而 PDDLStream 则在搜索与采样之间分配工作量 [5]。CAPE 沿用了这一研究方向的两项原则：乐观候选项引导评估，评估与搜索共用一个预算。变化之处在于被评估的对象。边评估会在固定图上返回代价，而效果会返回一个响应，该响应决定配置如何拆分，进而决定其下方策略的形态。因此，CAPE 的前沿除了乐观候选项之外，也会沿着可执行策略推进。

部分策略与保证。 LAO* 扩展部分解图，并将值更新传播给其祖先 [7], ; BRTDP 则通过下界值和上界值引导搜索 [15]。决策区域确定和决策理论故障排查会选出决策所需的观测与测试 [8, 10]。SPIBB 对相对于基线策略的期望回报损失设定界限 [12],, 而安全屏蔽则在部分可观测性下强制提供保证 [1, 17]。CAPE 在三个方面有所不同：其下界值由基于已获取效果构建并存储的策略实现；其上界值则通过放宽未知效果本身得到；其要求以配置为单位：初始策略已完成的每个配置都必须继续保持完成，而基线相对方法保留的是期望。

3. 分析

3.1. 问题形式化

我们将第 1 节中的数据摄取示例形式化。智能体持有一张表和一项提议操作，此处为对

西雅图天气记录的投影，并且必须通过一种工具发布该操作的结果，但无法查看该工具的解析配置。图 2 采用四种配置：逗号或分号分隔符，分别搭配日优先或月份优先日期格式。

动作与效果。 候选调用（我们称之为动作）分为两类。一个程序 会将表与操作一并通过一次序列化提交。一个探测会提交一个小型诊断样例，并返回工具对其的解析结果，但不会发布任何内容。图 2 中的 CSV 探测 Q13 会提交一个以分号分隔的样例，其首行包含日期 `05/04/2026`：分号解析器会根据日期约定将其识别为4月5日或5月4日，而逗号解析器则会返回解析错误（完整样例见补充材料）。

设 H 为候选配置的有限集合，$A = P \cup Q$ 为程序 P和探测 Q的目录。对于固定请求，在配置 h下调用动作 a 会产生确定性的 效果 $f(a,\ h) = (o(a,\ h)\ g(a,\ h))$：一个响应和一个提交位。当程序发布所请求的结果时，该位为1，此时执行结束；探测永不提交。未提交的调用不会改变任务状态。一个校准查询 会在复现 h的测试环境中评估一对 $(a,\ h)$，其响应和提交语义与部署时相同。已获取效应会累积在表 $\mathcal{D} \subseteq A \times H$中。

策略与目标函数。 策略根据迄今收到的响应选择每个动作，并且在任何一次执行中最多使用 C 次调用。由于效果具有确定性，且失败的调用不会改变任何内容，因此策略在收到一系列响应后的知识，就是与这些响应相符的配置集合 $V \subseteq H$，以及剩余调用次数 b。用 $S(\pi; V,\ b)$ 表示 π 在还剩 b 次调用时能够完成的配置，用 $M(\pi;\ V,\ b)$ 表示其在 V 中的调用总数，包括执行失败时进行的调用。初始策略 π_0 定义必需集合 $R_0 = S(\pi_0;\ H,\ C)$，编译器则最大化

$$J(\pi) = \big(|S(\pi; H, C)|, -M(\pi; H, C)\big), \quad R_0 \subseteq S(\pi; H, C), \tag{1}$$

按字典序优化，并对每种配置赋予相同权重：首先尽可能多地完成配置，然后尽可能少地使用调用次数，同时不丢失任何 π_0 能够完成的配置。编译时限 τ 限制构建 π 所需的工作量，而 C 限制每次执行的调用次数。

一个运行示例。 在图 2中，逗号分隔/月优先程序 P6 提交于 h_4 并返回相同的解析结果-
其余三种配置上的错误，因此剩下 $V =$

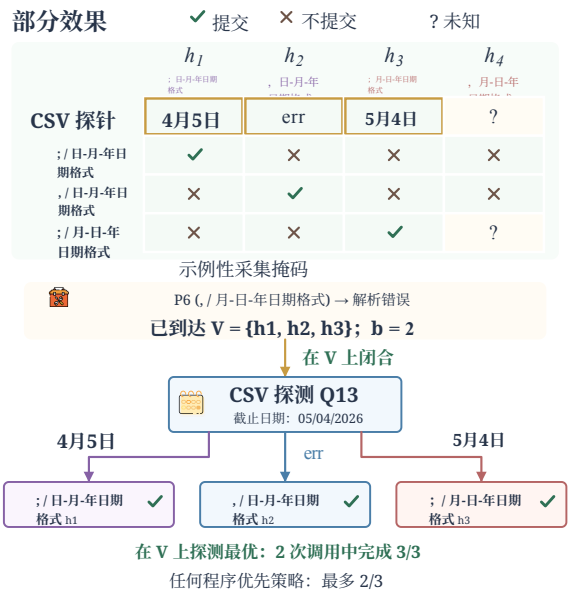

图2. **局部闭合探测在已达范围内最优。**剩余两次调用时，Q13 会在 V 中完成全部三种配置；任何程序优先后续策略最多只能完成两种。表中展示了在 V 上提交的三个程序；日期是完整响应的缩写；日月年/月日年表示日期顺序。获取掩码仅作示意。

$\{h_1, h_2, h_3\}$ 需要两次调用。探测 Q13 将 V 分成三种响应，每种响应后接对应的程序，因此 V 中的每种配置都能在剩余两次调用内完成。程序优先后续策略最多只能完成两种（第 4.1 节），因此 Q13 在此状态下是最优的；虽然 Q13 以及分号/月优先程序对 h_4 的效果尚未知，但其分支是可执行的。

3.2. 获取—计算权衡

校准提供效果，而搜索则确定这些效果是否支持更好的策略。如果 c_i 是第 i 次校准查询的测量成本，则编译时间可分解为

$$T_{\text{comp}} = T_{\text{fixed}} + T_{\text{search}} + \sum_i c_i, \tag{2}$$

其中，T_{fixed} 涵盖输入构造、最终序列化和清理，T_{search} 涵盖策略构造、界计算以及新效果产生后进行的更新。这两个变量项彼此制约：更谨慎地选择查询的搜索会缩短总和，却会延长T_{search}，正如惰性图搜索 [13, 14] 中所示。

这种交互对工具策略而言代价高昂。我们早期的一个求解器会在每次后续查询之前，精确求解模型中已获取的部分。在鸟击投影任务上，经过初始验证后又进行了约 600 次校准查询，它花了约 22 秒重新规划；在开发工作负载上，它的 CSV 完成率达到 $C = 5$

而急切精确搜索的完成率达到了 98.44%。

界计算展现了另一面。我们分析了一个两次调用的界算子以两种方式应用时的情况：仅应用于当前正在获取的动作（Selected），或应用于每个符合条件的状态（始终）。在 $C = 3$ CSV上，在所有位置进行收紧会使界计算时间从0.621秒增加到1.468秒，并使校准时间从8.814秒降至7.658秒，整体上有所节省。在 $C = 5$ 上，界计算时间从0.019秒增加到1.192秒，而校准几乎没有变化（补充材料，第E.1节）。只有当其避免的查询成本高于新增的搜索成本时，细化才会实现自我偿付。

3.3. 对策略构造的启示

三项后果塑造了编译器。一个分支只需要到达该分支的配置上的效果，因此在其动作对应的行尚未完成之前，它就已变得可执行。由已获取效果构建的策略，无论之后获取什么，都会遵循相同的路径，因此可以存储并就地改进。乐观候选项为每个未知配置赋予各自的假设响应，它们告诉编译器应在哪里查找，而不是应返回什么。因此，进一步的校准或计算只有在能够改进已存储的策略，或表明某个未解决动作无法胜过该策略时，才是有用的。

4. CAPE

CAPE 在选定的局部范围内交替进行校准，并更新持久策略图（图3；伪代码见补充材料）。该图中的状态 $(V,\ b)$ 是一个响应一致的配置集合及其剩余调用次数。每个状态都包含其义务 $R_0 \cap V$、由已存储的可执行策略达到的下界值L，以及任何策略都无法超越的上界值U。值是已完成配置数和取负的调用次数组成的数对，并按照式（1）中的方式按字典序比较。我们以图 $(V,\ 2)$中的状态2为例，具体说明每个步骤。

4.1. 局部策略与增量组合

当某个动作对 V中的每种配置都已获得其效果后，该动作便可在$(V,\ b)$处纳入可执行策略；我们称其在 V上已关闭。已关闭动作会完成其提交所针对的配置$G_a(V)$，并依据观测到的响应将剩余配置划分为互不相交的单元 V_o，每个单元都是一个子状态$(V_o,\ b-1)$。若各子状态中包含的策略具有成功集合 S_o 和调用总数 M_o，则将它们置于 a 之下组合，可得

$$S_a = G_a(V) \cup \bigcup_o S_o, \qquad M_a = |V| + \sum_o M_o. \tag{3}$$

V 中的每个配置都要为第一次调用付费，而每个响应单元中的配置则要为其子状态的调用次数付费。当 S_a 涵盖 $R_0 \cap V$时，该组合是可行的。停止的价值为（0，0），当义务为空时是可行的。

在（V，2）时，Q13 已关闭且不向任何位置提交。其响应单元 $\{h_1\}$、$\{h_2\}$ 和 $\{h_3\}$ 分别由匹配的程序完成，因此公式（3）给出 $S_a = V$ 和$M_a = 3+1+1+1 = 6$。从分号/日期优先程序开始，则会完成 h_1 并留下失败单元$\{h_2, h_3\}$，其中一次调用完成一个配置：两次完成对应 $M_a = 3+2 = 5$。探测组合的价值更高，为（3，-6），而（2，-5）则较低。

CAPE 将每个状态的最佳可行组合存储为其见证：一种包含其成功集合和调用总数的策略。在这里，见证是 Q13 关闭之前的程序优先策略，以及之后的探测策略。当子状态的见证得到改进时，公式（3）会更新通过已关闭动作到达该子状态的父节点，而不会重新检查其他动作。验证 π_0 会为根节点植入见证。由于见证只使用已获取效应，后续采集不会改变其执行过程及其值。

4.2. 乐观界与获取前沿

上界值会告诉编译器哪些未完成动作仍有可能改进见证。令 $E_a(V) = \{h \in V\colon (a,\ h) \notin \mathcal{D}\}$为动作 a 未知的配置。程序 p 至多可以在所有这些配置上提交，因此其乐观成功集为 $G_p(V) = G_p(V) \cup E_p(V)$。任何程序可能完成的配置，以及单个程序可能完成的最大配置数为

$$\overline{G}(V) = \bigcup_{p \in P} \overline{G}_p(V), \qquad \overline{d}(V) = \max_{p \in P} |\overline{G}_p(V)|. \tag{4}$$

对于非空的 $\overline{G}(V)$ 和 $b \geq 2$，一个廉价的上界值是

$$B(V,b) = \left(|\overline{G}(V)|,\ -\left[|V| + |\overline{G}(V)| - \overline{d}(V)\right]\right). \tag{5}$$

没有任何策略能完成超过 $|G(V)|$ 个配置，而能够完成所有配置的策略会在 V 的每个配置上各调用一次，并且至少会在其第一个程序无法覆盖的 $|G| - d$ 个完成项上再调用一次。$G(V)$ 之外的必需配置会使状态变得不可行。在包含已获取 V 上程序效果的运行示例中，三个程序分别在一个配置上提交，因此 $G(V) = V$，$d(V) = 1$，而 $B(V,\ 2) = (3,\ -5)$。

同样的乐观性也会为单个未完成动作提供界。假设未完成程序会在其未知配置上提交，未完成探测会在每个未知配置上返回不同的响应，这是其行的任何完成都可能产生的最细划分；已获得的失败则保留其实际响应单元。令 $\mathcal{F}_a^+$ 表示由此得到的乐观单元，其子节点上界值为 $U_W = (k_W,\ c_W)$，对应于 $W \in \mathcal{F}_a^+$。该动作的上界值为

$$U_a(V,b) = \left(|\overline{G}_a(V)| + \sum_W k_W,\ -|V| + \sum_W c_W\right). \tag{6}$$

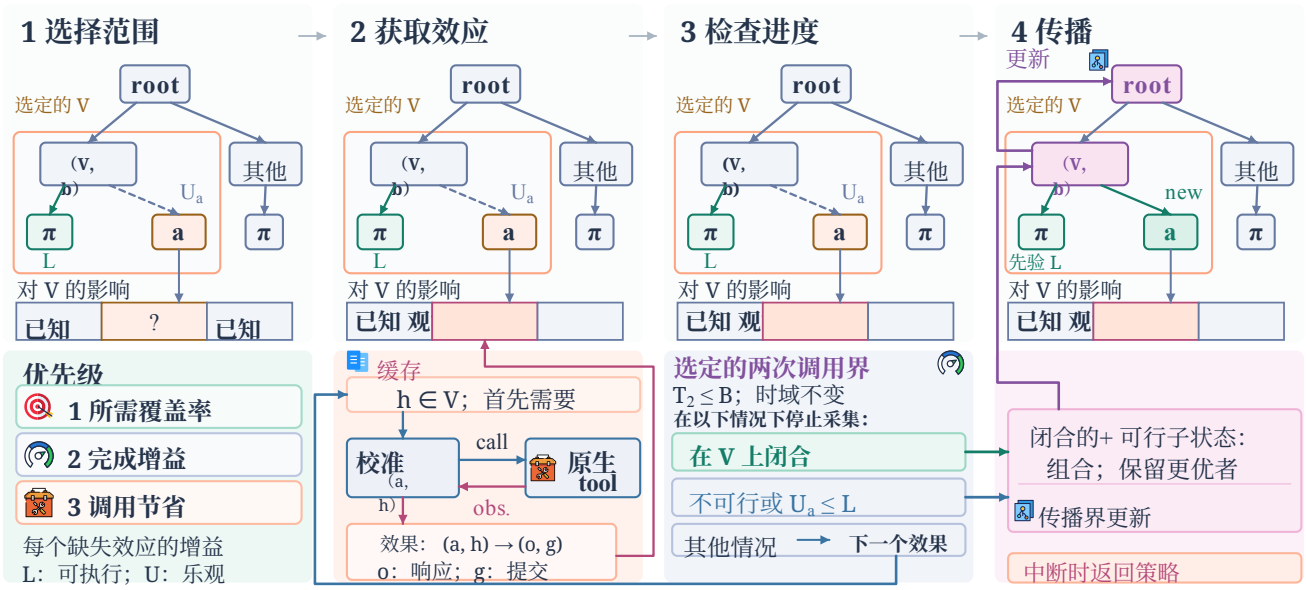

图 3。**局部观测会更新持久策略图**。采集在闭合、不可行性或支配时停止。组合还需要可行子节点见证；见证和界的变化会传播到受影响的父节点。

我们在界函数 $U_a(V,\ b;\ B')$ 提供子节点值时，将其记为 B'。不可行的必需子状态会排除该动作。已闭合动作使用其子状态的当前上界值，未完成动作使用廉价界 B，除非正在对其进行收紧（第 4.3 节）。一个状态会取其最佳可行动作界，在允许时纳入停止，并受公式（5）限制。随着效果到达，上界值仍保持有效，因此 CAPE 会取旧值与新值中的最小值。

校准从一个采集前沿中选择：该前沿包括当前可执行策略从根节点到达的状态，以及一条最佳乐观替代路径上的状态，其中已闭合动作会延迟展开。在这一前沿上，未完成动作分为三个层级进行排序（图 3）。所需覆盖率优先：缺少针对其义务的可行见证的状态，按每个缺失效应所需的配置数排序。完成增益其次：其上界值比见证完成更多配置的动作，按每个缺失效应的增益排序。调用节省最后：只能节省调用次数的动作，按每个缺失效应的节省量排序。在选定动作中，首先查询必需配置。在 Q13 于$(V,\ 2)$上闭合之前，其上界值比程序优先见证多完成一个配置，因此将其置于第二层级。

4.3 选择性收紧与计算分配

廉价界忽略了剩余调用次数是否能够利用观测到的响应单元。对于还剩两次调用的状态，乐观问题规模足够小，可以精确求解：CAPE 会枚举每个首个动作，并在每个乐观单元之后，选择最佳的单个程序，

$$T_2(V) = \max_{a \in A} U_a(V, 2; B), \tag{7}$$

包括在可行时选择停止。$T_2(V)$ 是松弛的双调用问题的最优值，并满足 $T_2(V) \le B(V,\ 2)$。在运行示例中，程序优先后续策略最多完成两个配置，而探测在每个完成的配置上都需要第二次调用，因此 $T_2(V) = (3,\ -6)$。无论其他探测效果如何，这都等于见证值，因此进一步采集无法改善此状态。

应将此算子应用于何处，是第 3.2 节中的分配问题，因为收紧每个状态的代价可能超过其所避免的校准成本。CAPE 将 T_2 应用于当前正在采集的动作的双调用后续策略，因为更紧的值会改变下一步决策。每获得一种新的效果后，只要此动作已闭合、不可行或已被可用见证界定，就立即停止采集该动作。所有其他未完成动作仍使用廉价界。

4.4 中断、覆盖与认证

命题1。 在第 *3.1* 节的效应模型下，每个被接受的根见证都保持 R_0，且调用次数不超过 C，被接受的目标值单调不减。见证值在后续一致的采集中保持不变。每个维护的根上界值都界定了完整目录最优值。根节点上下界值相等即可证明最优性。

该证明（补充材料）基于两个事实。已闭合动作的组合会严格按照计算结果执行，因为其单元格都是观测到的单元格；而乐观性

表1。**CAPE 将较高的完成率与较低的编译成本相结合**。30秒时间预算下的 DeepSeek 机制研究。成功率指完成率（%），将无效提案计为失败；调用次数按配置统计；时间指编译时间（秒）。粗体表示各列唯一最优值以及 CAPE 并列的最优值；补充材料还列出了两组消融实验和所有区间。

		CSV			JSON		
C	方法	成功率	调用次数	Time	成功率	调用次数	Time
3	**CAPE**	**85.16**	2.196	10.774	**94.74**	1.476	**11.919**
	急切精确	85.16	2.196	11.173	94.60	1.474	18.021
	仅程序	76.76	**2.022**	**3.631**	94.60	1.474	17.792
	受保护的两步法	83.98	2.474	11.121	94.60	1.493	18.008
	通用惰性	83.74	2.198	17.089	92.90	1.495	16.200
	程序优先	85.16	2.196	11.402	94.60	1.474	17.878
5	**CAPE**	**98.44**	**2.348**	11.201	**94.74**	1.475	**11.903**
	急切精确	98.44	2.348	11.417	94.60	1.474	18.079
	仅程序	92.19	2.365	**3.624**	94.60	1.474	17.744
	受保护的两步法	98.44	2.838	11.192	94.60	1.493	18.005
	通用惰性	84.18	2.534	13.700	92.90	1.546	16.008
	程序优先	98.44	2.348	11.572	94.60	1.474	17.919

是一种松弛，因为补全缺失效应只能移除假定提交或合并假定单元格。该命题对每一种获取顺序都成立；获取顺序决定了在给定策略或证书之前进行多少校准和搜索（第 5.3节）。

发生中断时，CAPE 会返回最新的根见证；如果初始验证本身被中断，则返回 π_0，不附带证书。效果表完成后，一个针对相同目标函数的紧凑型精确求解器会使用剩余的计算额度。

5. 实验

5.1. 设置

请求与工具。 八个 Vega 源文件 [19] 提供 64 个请求，即针对八种操作（例如投影、分组计数和过滤），每个源文件对应每种操作一个请求；每个源文件最多包含 256 条记录。两种工具负责采集数据：一种是基于 DuckDB 读取器构建、包含 64 种配置的 CSV 工具，另一种是基于 pandas 构建的 JSON 工具，其配置为 88 [3, 18]。这些配置会改变分隔符、引号规则、缺失值标记、数值型和日期编码以及 JSON 定向方式。程序是按照这些编码对表进行序列化得到的不同表示，探测则是固定的诊断测试装置。只有当程序调用的结果等于参考输出时，该调用才会提交；参考输出会根据类型化数据上的请求操作预先计算，且独立于提议；否则，该调用会回滚。

提案与研究。 大语言模型每个请求提出一个操作。机制研究使用 DeepSeek-V4-Pro，其中 64 个提案有 63 个模式有效且与类型化参考相匹配；剩下的一个对每种方法都计为失败。它在以下两方面比较这些方法工具，在 $C \in \{3, 5\}$ 中进行两次新进程重复。下面的匹配、扩展和执行研究均使用未经修改的相同方法；匹配研究在一个由十四个模型组成的提案面板的不同输入上运行。

比较。 所有编译器都从同一个初始策略开始：该策略按照目录顺序尝试程序，并继承其必需配置。急切精确 会采集所有剩余效果，然后运行紧凑的约束搜索。通用惰性 以每批 32 个的方式验证乐观模型。程序优先 会在任何探针效应之前采集程序行。受保护的两步法 使用滚动诊断前瞻；当某个替代方案的覆盖范围或目标值下降时，则保留初始策略。仅程序 会从目录中移除探测。匹配研究交叉比较两种采集调度方案：CAPE 的前沿，以及遵循最佳乐观解路径的与/或搜索；同时采用三种界模式：无、选定动作和始终，其中“始终”会在每次符合条件的界请求中应用相同的 T_2 界算子。两种调度方案共享见证对象、验证、缓存、界算子和精确回退机制，因此二者的差异仅在于采集顺序。

测量与验证。 每次计时编译都在新进程中开始，使用空缓存，并设有软性30秒时间预算。编译时间 是一次编译的完整生命周期：设置、验证初始策略、采集、搜索、序列化和清理。完成度表示已完成配置的百分比，并将无效提案计为失败；调用次数和编译成本均按模式有效提案取平均值。各来源权重相同，配对95%区间来自对八个来源进行5,000次自助法重采样的结果。所有结果均独立于编译器，通过对完整效果表进行动态规划以及通过新执行进行验证，

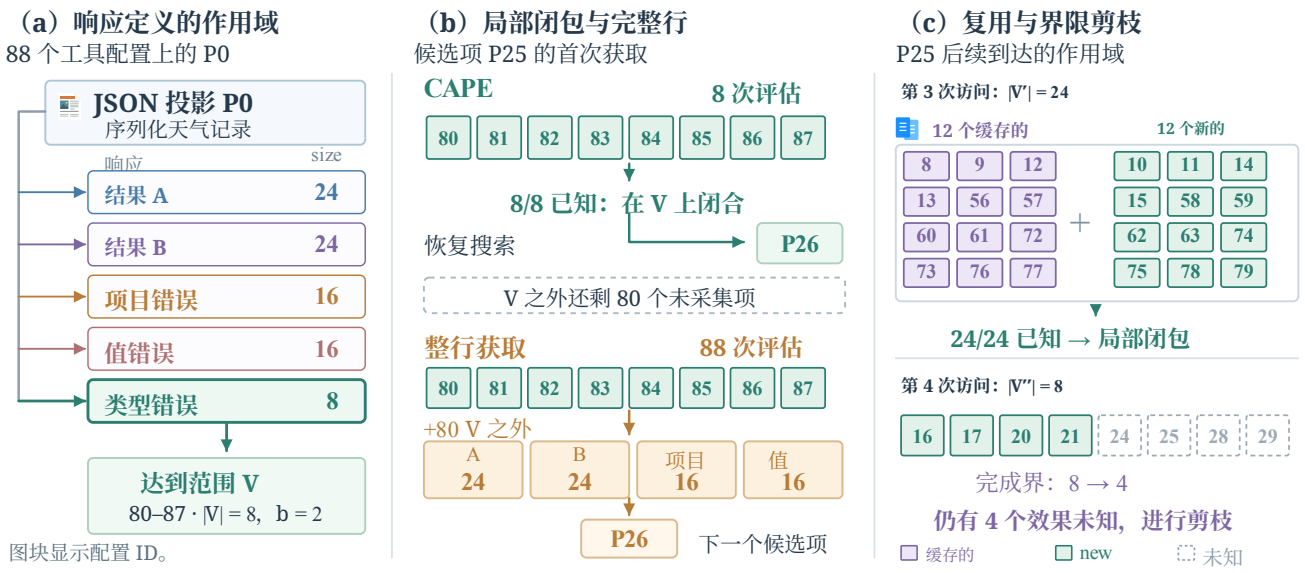

图 4. **记录的校准案例**（JSON 天气投影，$C = 3$）。（a）P0 响应将 88 种配置划分为不同组。（b）CAPE 完成了对这 8 种配置的范围搜索；整行获取还评估了范围之外的 80 种配置。（c）后续访问复用了 12 种效果，完成了对 24 种配置的范围搜索，并在评估 4 次后剪枝了一个包含 8 种配置的范围。

运行时证书是编译器自身对其根节点值已满足要求的声明。补充材料给出了完整的协议。

5.2. 编译时限内的策略质量

CAPE 返回最优或近似最优的策略。在 502/504 个编译任务中，它达到了完整目录最优值，并在运行时为其中 498 个提供了认证；所有经过审计的策略都保留了初始策略已完成的配置。在剩余的两个编译任务中，对于 $C = 3$ 处的 JSON 天气投影，CAPE 通过更多调用达到了最优完成。

CAPE 以更低的编译成本，在策略质量上达到了急切精确搜索的水平（表1）。在 CSV 上，完成度和调用次数相同；在 JSON 上，CAPE 的完成度高出 0.14 个百分点。平均编译时间降至 11.449 秒，而急切精确搜索为 14.673 秒，通用惰性搜索为 15.749 秒。与急切精确搜索相比，配对后的节省时间在 CSV 上为 0.399 秒（$C = 3$），在 JSON 上为 6.102/6.176 秒（$C = 3/5$）；而 $C = 5$，$[-0.125, 0.693]$处的 CSV 区间包含零。

在调用次数受限时，探测最为重要。在 $C = 3$，带探测的策略将 CSV 完成率从仅程序搜索的 76.76% 提高到 85.16%。与受保护的两步法相比，CAPE 提高了 1.17 个百分点，并为每个配置节省 0.278 次调用。在 $C = 5$，两者都完成了 98.44% 的 CSV 配置，而 CAPE 使用 2.348 次调用，受保护的两步法使用 2.838 次。

5.3. 采集与计算的权衡

局部效应范围。 仅校准已达范围是单项最大节省。完整行消融会获取动作的已达范围关闭后的剩余单元。在 JSON 上，它额外发出 848.1 和 563.3 次校准查询，并少完成 4.26 和 3.69 个百分点，$C = 3/5$ 因为已达范围之外的查询会在另一次后续执行变得可执行之前耗尽时间预算。图 4跟踪记录的 JSON 天气投影中的一个候选程序 P25。初始程序 P0 的响应留下一个包含八种配置的范围，CAPE 通过八次查询完成 P25，而整行获取需要 88 次；后续访问则复用缓存的效果，并在收紧后的边界上提前停止。

获取顺序。 同时遵循可执行策略和乐观策略可以节省编译时间。在两种调度方案中均使用选择性界限时，CAPE 在 $C = 3$ 的 CSV 上比匹配的与/或调度器快 0.998 秒，在 $C = 3/5$ 的 JSON 上快 4.200/3.799 秒，并且三个配对区间均不包含零。在 $C = 5$的 CSV 上，两种调度方案的执行质量相同，且时间区间包含零；在所有三种界模式下，JSON 的节省时间均保持不变。在未进行收紧的情况下，CSV 的差异缩小，并且 CAPE 在 $C = 5$ 时更慢，因此获取顺序的收益取决于其可用的界限信息。

计算更紧密时何时有益。 只有在能够省去足够多的校准时，收紧才有益。在 CSV $C = 3$ 上，始终收紧可避免额外 433.6 次查询，并且在执行质量相同的情况下，比选择性收紧快 0.741 秒。在 $C = 5$ 上，它只能避免少量查询，而选择性收紧快 1.145 秒。两个配对区间均不包含零，

表 2。 **CAPE 在规模较大的任务族上提升了完成率。** 在 $C = 5$ 的时间预算内提出八个 CSV 投影请求。各列参见表 1。

$\|H\|$	方法	成功率 ↑	调用次数 ↓	时间（秒）
64	急切精确	100.00	2.875	14.17
	匹配的与/或	100.00	2.875	14.87
	CAPE	**100.00**	2.875	15.14
128	急切精确	66.41	3.309	26.64
	匹配的与/或	74.22	3.379	26.74
	CAPE	**88.28**	3.309	26.66
256	急切精确	57.03	3.506	26.85
	匹配的与/或	58.59	3.619	26.90
	CAPE	**62.60**	3.479	26.90

而第 3.2 节的上界成本解释了这一反转。在 JSON 上，两种模式差异不大。

改进来源。策略质量取决于动作目录能够达到的水平，以及编译器在 τ 内找到的结果。在 CSV 上，探测会提高部分请求可达到的完成率，如图 2 所示。在 JSON 上，对于每个审计请求，有探测和无探测时的最优值都相同（补充材料，第 D.1 节），因此计时比较衡量的是各编译器以多高效率恢复相同的可达完成率。在 $C = 3$ 上，CAPE 用时 11.919 秒，完成率达到 94.74%；而仅程序搜索用时 17.792 秒，完成率为 94.60%（表 1）。匹配调度器和完整行消融的结果表明，这一差异归因于采集顺序和局部闭包。

5.4. 结构与扩展行为

当 CSV 系列超出时间预算后，各编译器的质量便开始拉开差距。我们取每个来源的投影请求，将 CSV 系列从 64 种配置扩展至 128 和 256 种配置，方法是添加日期编码，同时保留原有效果（表 2）。在 64 种配置下，三种方法均完成所有配置。在 128 种配置下，CAPE 完成的配置比急切精确多 21.88 个百分点，比匹配的与/或多 14.06 个百分点，而三者耗时均约为 27 秒。在 256 种配置下，增益分别为 5.57 和 4.00 个百分点。所有四个配对完成率区间均不包含零；在 128 种配置下，时间预算从 5 秒到 120 秒时，CAPE 的表现始终不低于任一对照组。当模型尚未解决时，持续改进可执行策略的编译器会返回更优策略。

补充材料增加了这项扫描（第 G.4 节），$C = 8$，合并了探测反馈和提案面板。

5.5. 针对反馈驱动重试的执行

表3 比较了编译后的策略在全部 64 个请求上的执行情况，以及两个在线控制器在工具和$C = 3/5$。目录重试执行初始策略。大语言模型

表 3. **编译后的策略完成更多配置。** 在 256 个条件下的完成率（%）；调用次数为有效提案的平均值，包括失败路径。

接口/C	方法	成功率 ↑	调用次数 ↓
CSV/3	目录重试	69.97	2.113
	大语言模型重试	69.78	2.113
	CAPE	**85.16**	2.196
CSV/5	目录重试	83.01	2.605
	大语言模型重试	82.32	2.608
	CAPE	**98.44**	2.348
JSON/3	目录重试	60.09	1.957
	大语言模型重试	60.09	1.957
	CAPE	**94.74**	1.476
JSON/5	目录重试	67.90	2.731
	大语言模型重试	72.80	2.682
	CAPE	**94.74**	1.475

重试使用 DeepSeek-V4-Pro，并固定操作：它执行初始策略中的第一个程序；每次失败后，从请求、模式、数据示例和错误历史中选择一个尚未尝试过的序列化方式。

CAPE 比大语言模型重试多完成 15.38 至 34.66 个百分点的配置，四个配对的 95% 区间均高于零，并且保留了初始策略完成的每个配置。在 JSON 上，$C = 5$ 它所需的调用次数也更少：每个配置平均 1.475 次，而大语言模型重试为 2.682 次。编译会在部署之前选择探测和依赖响应的延续，并将每个候选配置纳入考量。计入编译时间后，在固定端点执行 1 次、8 次和 32 次时，CAPE 的总时间比大语言模型重试高出 6.2–10.2 秒（补充材料，第 G.2 节）。

6. 结论

编译工具使用策略，关键在于如何分配有限的预算。CAPE 将可执行策略作为校准与搜索的共同对象：在能够扩展或改进该策略的地方获取效果，在能够终止采集的地方收紧界；任何时刻已有的策略都可以返回，并附带其覆盖率保证；当其价值达到乐观界时，还可附带最优性证书。在我们的实验中，CAPE 在 504 项编译任务中的 502 项达到最优，且比急切精确搜索减少 22.0% 的编译时间；当配置族超出时间预算后，其返回的策略优于急切搜索和匹配的惰性搜索；在执行阶段完成的配置数也多于反馈驱动的大语言模型重试。

参考文献

[1] Steven Carr、Nils Jansen、Sebastian Junges 和 Ufuk Topcu。通过在部分可观测性下进行屏蔽，实现安全强化学习。载于*AAAI*人工智能会议论文集，第 14748–14756 页，2023 年。[2] Christopher M. Dellin 和 Siddhartha S. Srinivasa。通过对含边选择器的路径进行惰性最佳优先搜索，统一表述边评估代价高昂的最短路径问题。载于国际自动规划与调度会议论文集，2016 年。[3] DuckDB贡献者。CSV导入文档，2026 年。访问日期：2026 年 9 月 10 日。[4] Wei Fang、Yang Zhang、Kaizhi Qian、James Glass 和 Yada Zhu。PLAY2PROMPT：通过工具交互，为大语言模型智能体进行零样本工具指令优化。载于计算语言学协会年会论文集：*ACL 2025* 精选论文，第 26274–26290 页，2025 年。[5] Caelan Reed Garrett、Tomás Lozano-Pérez 和 Leslie Pack Kaelbling。PDDLStream：通过乐观自适应规划整合符号规划器与黑盒采样器。载于国际自动规划与调度会议论文集，第 440–448 页，2020 年。[6] Skyler Hallinan、Thejas Venkatesh、Xiang Ren、Sai Praneeth Karimireddy、Ashwin Paranjape、Yuhao Zhang 和 Jack Hessel。OpaqueToolsBench：通过交互学习工具行为的细微差异。*arXiv* 预印本 *arXiv:2602.15197*，2026 年。[7] Eric A. Hansen 和 Shlomo Zilberstein。LAO*：一种可找到含环解的启发式搜索算法。《人工智能》，129(1–2):35–62，2001 年。[8] David Heckerman、John S. Breese 和 Koos Rommelse。决策理论故障排查。《*ACM*通讯》，38(3):49–57，1995 年。[9] Cheng-Yu Hsieh、Si-An Chen、Chun-Liang Li、Yasuhisa Fujii、Alexander Ratner、Chen-Yu Lee、Ranjay Krishna 和 Tomas Pfister。工具文档使大语言模型能够零样本使用工具。*arXiv* 预印本 *arXiv:2308.00675*，2023 年。[10] Shervin Javdani、Yuxin Chen、Amin Karbasi、Andreas Krause、J. Andrew Bagnell 和 Siddhartha Srinivasa。面向决策的近最优贝叶斯主动学习。载于第十七届国际人工智能与统计会议论文集，第 430–438 页，2014 年。[11] Sehoon Kim、Suhong Moon、Ryan Tabrizi、Nicholas Lee、Michael W. Mahoney、Kurt Keutzer 和 Amir Gholami。用于并行函数调用的大语言模型编译器。载于第 *41* 届国际机器学习会议论文集，第 24370–24391 页，2024 年。[12] Romain Laroche、Paul Trichelair 和 Remi Tachet des Combes。基于基线自助法的安全策略改进。载于第 *36* 届国际机器学习会议论文集，第 3652–3661 页，2019 年。[13] Aditya Mandalika 、 Oren Salzman 和 Siddhartha Srinivasa。面向图中高评估代价边的高效路径规划的惰性滚动时域 A*。载于国际自动规划与调度会议论文集，第 476–484 页，2018 年。[14] Aditya Mandalika、Sanjiban Choudhury、Oren Salzman 和 Siddhartha Srinivasa. 面向机器人运动规划的广义惰性搜索：通过基于事件的切换机制交错进行搜索与边评估。载于 国际自动规划与调度会议论文集，第 745–753 页，2019 年。[15] H. Brendan McMahan、Maxim Likhachev 和 Geoffrey J. Gordon. 有界实时动态规划：采用单调上界并提供性能保证的 RTDP。载于第 *22* 届国际机器学习会议论文集，2005 年。[16] Archiki Prasad、Alexander Koller、Mareike Hartmann、Peter Clark、Ashish Sabharwal、Mohit Bansal 和 Tushar Khot. ADaPT：基于语言模型的按需分解与规划。载于 计算语言学协会论文集：*NAACL 2024* 论文集，第 4226–4252 页，2024 年。[17] Shili Sheng、David Parker 和 Lu Feng. 通过屏蔽实现安全的 POMDP 在线规划。载于 *IEEE*国际机器人与自动化会议论文集，第 126–132 页，2024 年。[18] pandas开发团队。pandas.read_json：JSON摄取。https://pandas.pydata.org/docs/reference/api/pandas.read_json.html，2026 年。行为已根据 pandas 3.0.2 实现进行验证。[19] Vega项目贡献者。Vega数据集。https://github.com/vega/vega-datasets，2026 年。来源修订版本 2434f551e0bb12b99a4ce6764fbc0ef39bea145e。[20] Xingyao Wang、Yangyi Chen、Lifan Yuan、Yizhe Zhang、Yunzhu Li、Hao Peng 和 Heng Ji. 可执行代码动作能够激发更出色的大语言模型智能体。载于 第 *41* 届国际机器学习会议论文集，第 50208–50232 页，2024 年。[21] Binfeng Xu、Zhiyuan Peng、Bowen Lei、Subhabrata Mukherjee、Yuchen Liu 和 Dongkuan Xu. ReWOO：将推理与观测解耦，以构建高效的增强型语言模型。*arXiv* 预印本 *arXiv:2305.18323*，2023 年。[22] Shunyu Yao、Jeffrey Zhao、Dian Yu、Nan Du、Izhak Shafran、Karthik Narasimhan 和 Yuan Cao. ReAct：在语言模型中协同推理与行动。载于 国际学习表征会议论文集，2023 年。

补充材料

A. 局部策略的正确性与部分效应界

固定请求、配置族 H、动作目录 $A = P \cup Q$、初始策略 π_0、执行预算 C 和受保护集合$R_0 = S(\pi_0;\ H,\ C)$。动作的效果是确定性的，由响应和提交位组成。成功程序会终止；探测不会提交；失败调用不会改变与未来调用相关的状态。校准和执行遵循相同的语义，且已获取效应不可变。因此，存续配置集合 V和剩余预算 b足以描述策略状态。

A.1. 可执行组合

当动作对 V 的所有效果都已获得时，该动作在 V 上是封闭的。其即时成功配置 G_a（V）和失败响应单元V_o划分了 V。设每个子见证都能在至多 $b-1$ 次调用内满足其义务 $R_0 \cap V_o$。每个必需配置要么立即成功，要么属于一个能够完成该配置的子状态。因此，父级可在 b 次调用内保持 $R_0 \cap V$。不相交的单元给出

$$S_a = G_a(V) \cup \bigcup_o S_o, \qquad M_a = |V| + \sum_o M_o.$$

这会同时统计失败路径和成功路径上的调用次数。当剩余调用次数为零时，仅当义务为空时，停止才是可行的。它的成功集合为 $\varnothing$，调用总量为零。

引理 1（可执行见证的持久性）。 *完全由 $\mathcal{D}$ 中的效果构建的见证，在每个一致扩展 $\mathcal{D}' \supseteq \mathcal{D}$ 中都具有相同的执行路径、成功集合和调用总量。*

证明。 对于任何已达配置，两个表中首个动作所获得的效果相同。其提交位要么终止两次执行，要么其响应选择相同的子状态。对剩余调用预算进行归纳，可得后续操作完全相同。在见证对象已达范围之外获得的效果无法改变这些路径。 □

策略图按剩余调用次数构成无环图。即使某个子状态的最佳值尚未解析，闭合动作备份也能组合出可行见证。只有严格改进才会替换已存储见证。持久性随后确保已接受的根节点值不会降低。该义务仍是原始的 R_0；在具有更佳字典序值的策略之间，其他成功配置可能会发生变化。

A.2. 可采纳上界值

对于每个程序，$G_p(V) = G_p(V) \cup E_p(V)$包含其在一致的完整模型中能够完成的所有配置。它们的并集 $G(V)$包含每个策略的支持范围。如果某个必需配置不在该并集中，则该状态不可行。

令 $d(V) = \max_p |G_p(V)|$。一个完成所有 $|G(V)|$ 可能成功的策略，必须在所有 V上开始，并且至少需要对 $|G(V)| - d(V)$个成功配置进行第二次调用。因此，其调用总量至少为 $|V| + |G(V)| - d(V)$。完成较少配置的策略，其字典序已经低于支持上界，与其调用成本无关。这证明了以下值的可采纳性：

$$B(V, b) = \left(|\overline{G}(V)|, -[|V| + |\overline{G}(V)| - \overline{d}(V)]\right), \qquad b \geq 2,$$

对于非空支持集。若支持集为空且没有义务，则允许停止。剩余一次调用时，枚举覆盖该义务的乐观终止程序，即可得到精确的松弛终端值。

引理 2（乐观效应松弛）。 *将未知程序效果视为即时成功，并将未知探测效应视为不同的响应单元，可为效果表的每一种一致完备模型给出一个上界。*

证明。 取任意一致完备模型中的一个可行策略。遇到未知程序效果时，将其后续执行替换为立即成功，既能保持其成功率，也可能减少调用次数。已获得的失败保留其实际单元。对于未知探测效应，为每种配置分配一个单独的响应，这会细化任何真实完成所产生的划分。细化后的观察者可以为所有会被合并的单元选择相同的后续执行，从而复现原策略。该模拟保留了所有必要的成功率。对这组扩大的选择进行优化，并组合有效的子状态上界，由此得到一个上界值。 □

在松弛的两次调用问题中，首个动作之后，每个失败响应单元都会对应一个终止程序；若可行，也可以选择停止。枚举首个动作和最佳终止选择即可精确计算该松弛问题。其值 $T_2(V)$ 是可采纳的，且不大于支持上界 $B(V, 2)$。用它替代子节点上界值，可保持祖先节点上的可采纳性。

A.3. 传播、约简与缓存

采集后，对于范围更小的一致模型集合，每个旧上界值仍然有效。因此，取旧的和新的有效界中的最小值是可靠的。先前被剪枝的动作，不可能优于那个已在其上界值上占优的持久见证。如果某个状态的下界值与上界值相等，那么所有后续的一致模型都具有相同的最优值。

作用域相关计算的缓存使用由该作用域内采集次数得出的代数。如果 r_h 对配置 h 的采集次数进行计数，那么当且仅当没有新的效果到达 V 时，$\sum_{h \in V} r_h$ 才保持不变，因为每个计数都不会减少。对于固定的作用域和义务，此代数已足够。程序支持和封闭终端选择仅取决于程序效果，因此探测获取不会改变这些缓存条目。

成功集合相同且失败响应划分相同的已采集动作可以共用一个代表元：针对每个单元，将后续操作关联到代表元观测到的标签。这样既能保持成功率，也能保持调用总量。可预测的非提交动作只有一个失败单元，且没有成功项。删除该动作并使用其后续操作，可在保持完成的同时，为每个到达的配置节省一次调用。状态不变的失败语义为此约简提供了依据。

A.4. 根保证与终止

初始验证确立了 R_0 和可执行初始策略。组合论证和见证持久性确立了可行且非递减的根节点改进。松弛论证和界传播确立了可采纳的根上界。如果根节点的值相同，则可执行策略的值至少不低于任何完整目录策略的值，因此它是最优的。

采集最多增加 $|A||H|$ 种不同的效果。当表完整后，相同的有限时域递推关系便成为精确结果，紧凑的完全信息求解器会利用剩余配额处理该递推关系。中断运行会返回当前可用的最佳可执行见证。在初始验证完成前发生中断时，将返回 π_0，且认证状态未设置。有限策略状态空间的规模可能关于 $|H|$ 呈指数增长。这些保证涉及可行性和值界；实测成本差异源于每种调度执行的采集和图计算量不同。

B. 动作目录、观测与实现

B.1. 程序和诊断探针

CSV 配置组合了六种二元读取器选项：逗号/分号分隔符、单引号/双引号、反斜杠/双引号转义、NULL/NA 缺失标记、逗号/点号小数分隔符，以及月份优先/日优先日期格式。它们的笛卡尔积包含 64 种配置。提议的结构化操作会被转换为查询；程序操作使用由编码选项产生的不同序列化批次。相同的载荷字节会被去重。该目录包含 68 个固定探针，包括分隔符、引号规则、转义、小数、日期和缺失值测试固件，以及组合这些特性的复合测试固件。

JSON 配置使用 records、split、index、columns、values 和表方向。前五种表方向具有推断或保留的数据类型；table 使用其模式，共有 11 种模式。将它们与开启/关闭日期转换、开启/关闭精确浮点解析，以及毫秒/纳秒单位组合，可得到 88 种配置。程序候选项会改变表方向、日期表示以及原生/字符串数值表示，并对字节完全相同的负载进行去重。探针目录包含 36 个测试固件。

表4。152 个保留的原生输入模型的目录规模。程序数量会因实际源数据的负载去重而有所不同。

接口	输入模型	配置	不同程序数量	探测
CSV	76	64	2, 5, 10, 12, 24	68
JSON	76	88	12, 36	36

B.2. 响应与提交语义

探测会返回解析后的列名、数据类型和行。成功的程序求值会返回其规范结果的指纹。该指纹包含 SHA-256 摘要和字节长度。CSV 解析/查询失败时返回 {error: invalid_csv_or_query}。JSON 失败时返回异常类及其最多 500 个字符的消息。策略分支使用规范响应键。策略接收响应和提交状态；工具控制其解析器配置。

请求的结构化程序和带类型源数据定义了参考输出。日期会规范化为 UTC ISO 表示。缺失值和非有限标量值映射为 JSON null，整数浮点值映射为整数。规范化序列化期间，字典键会排序；列表和行的顺序遵循操作。指纹比较不使用浮点容差。投影和过滤会保留原始行顺序，分组计数按规范 JSON 表示对键排序，排序行操作则按原始行索引打破并列。

工具会为一次程序动作启动事务。匹配的参考指纹会提交结果和回执；不匹配则回滚。诊断预览不会发布构件。全新重放会重建参考，并检查存储内容以及回执、观测和调用次数。参考构造包含在评估器设置中。

B.3. 将运行示例绑定到记录的效果

论文主体中的示例从西雅图天气投影记录中选择配置 0、1、32 和 33，并将其重命名为h_1、 h_2、 h_3、 h_4。它们共用单引号处理、双写转义、NA 缺失标记和点号小数。只有分隔符和日期解释有所不同。程序 P6 使用逗号分隔/月在前表示法：它会在 h_4 时提交，并在另外三个配置上将无效_csv_或_查询重新返回。在这个四配置示例中，它的失败因此会到达$V = \{h_1, h_2, h_3\}$ ，初始预算为三次调用，还剩两次调用。

Q13 是记录的复合 CSV 固定装置，字段包括 amount: DECIMAL(18,2)、due: DATE 和 memo: VARCHAR。其完整载荷为：

```
7.25；05/04/2026；NA 2.5；07/06/2026；’优先级，备注；“是”和 ”好的”’
```

该图用第一行的到期值来简写一次成功预览。两个日期响应和解析错误是三个不同的原生观测。

表 5。最优局部探测示例的记录效果。日期省略了共同的 UTC 时间后缀。完整模型审计提供了这些效果；该图展示了一个说明性的部分采集掩码。

配置	分隔符	日期解析器	程序 P6	探测 Q13
h_1	分号	日优先	解析错误	2026-04-05
h_2	逗号	日优先	解析错误	解析错误
h_3	分号	月份优先	解析错误	2026-05-04
h_4	逗号	月份优先	提交	解析错误

三个叶程序的目录索引分别为 0、1 和 5，并分别完成 h_1、 h_2 和 h_3。完整目录中的十个程序在 V 中至多各提交一个配置，并且其在 V 中的所有失败共享一个响应。因此，程序优先策略只有一个失败分支，其中包含至少两个配置，并且利用剩余的一次调用至多完成其中一个。其最大完成数为三者中的两个。Q13 区分全部三个配置，并使用与各配置匹配的终止程序分别完成它们，每个配置使用两次调用。任何在两次调用内完成全部三个配置的策略都必须以一次探测开始，因此其调用总数至少为六次。Q13 达到了这一上界。因此，在 (V，2) 处的字典序目标下，它是最优的。

对全部十个程序和 68 个探测进行穷举，得到最优策略和 Q13 后续策略均为 (3， -6)，而最佳程序优先后续策略为 (2， -5)；即使其剩余调用可以使用任意目录操作，结果仍然如此。按目录顺序的初始策略在 $C = 3$ 处使用程序 0、1 和 2；在此示例中，其必需配置为 h_1 和 h_2。两种被比较的后续策略都保留了该义务。Q13、Q36 和 Q52 并列最优；选择 Q13 是因为其三字段测试样例更为紧凑。该证明针对到达的状态 (V，2)。完整的实验运行保留全部 64 个配置和完整目录。

B.4. 获取顺序与实现细节

算法1 给出了正文中的编译循环。

算法 1CAPE 策略编译

要求： 初始策略 π_0、配置 H、动作 A、预算 C，τ
验证 π_0；设置 R_0 并为可执行见证设置随机种子
初始化根节点上下界值
当资源仍有剩余且根节点值不同时 **执行**
    如果效果表完整**则**
        使用剩余配额运行精确搜索
        跳出
    结束 if
    传播待处理见证和界更新
    在乐观备选方案上扩展已关闭动作
    根据当前可执行策略形成前沿
    按义务和增益选择未完成动作
    如果 未完成动作可以改善其状态**那么**
        重复
            先获取缺失效应和必需配置
            更新其边界；收紧两次调用的后续操作
        直到 已闭合、不可行、被支配或遭到中断
    结束 if
    组合已闭合动作并更新其父节点
结束 while
返回 最佳可执行策略；如果根节点值一致，则予以认证

采集评分首先区分不具备可行必需见证的状态。这类状态按照每个缺失单元格对应的必需配置数进行排序。对于具备见证的状态，正的完成增益优先于仅调用改进；完成增益和调用节省均除以缺失单元格计数。剩余平局时，优先选择范围较小的状态，其次是程序操作和较低的目录索引。在选定范围内，先采集必需配置，然后按照目录配置顺序执行。

原生父节点和上界父节点分别存储。已闭合备选项采用惰性激活，并对待处理更新进行去重。已闭合的一次和两次调用计算会为见证对象提供种子，而无需解决所有更大的子问题。选定采集使用更紧的延续计算；未选定的未完成动作使用支持界。新的效果会使受影响范围或程序生成缓存失效。模型完成时，共享紧凑精确求解器会处理剩余的递推关系。

完整行消融会在某个动作于其选定范围内关闭后，继续查询该动作行中的其余部分。不收紧消融保留相同的前沿和见证对象，同时使用支持界。匹配的与/或对照组沿着最佳乐观解路径执行。在已关闭动作处，它会下降到未解决值差距最大的子状态，并优先处理缺失的必需见证对象。它的无、选定和始终变体使用与 CAPE 相同的界算子。始终变体还会在探测获取后使受影响的延续界失效，因为新的探测响应可能改变另一个动作更紧的延续界。

C. 工作负载与测量协议

C.1. 来源与请求构造

这八个源文件取自正文论文中引用的Vega数据集。检索前已确定来源及操作规则。准备阶段保留前至多256条记录，即从17,681条原始记录中保留1,850条记录。表 6给出了源文件大小。

表 6。源文件和固定保留前缀。字段不包括内部行索引。

来源	原始行	保留行	字段
机场	3,376	256	7
鸟击	10,000	256	14
CO_2 浓度	741	256	3
灾害	803	256	3
发展指标	682	256	6
政治捐款	58	58	25
西雅图天气	1,461	256	6
股票价格	560	256	3

CSV准备阶段会在选择前缀之前推断源列中的标量类型。空字段将变为缺失值。精确的有符号64位整数会予以保留；数值转换必须保留整数值。前导零字符串和空格仍保留为字符串。JSON会严格解析。通用准备器会展开嵌套路径和至多16个元素的列表；其余容器则转换为规范 JSON 文本。系统还会添加一个内部原始行索引。

每个来源产生八个请求。投影按固定顺序选择所有源列。行计数返回批大小；缺失值计数涵盖每个源列。分组计数使用第一个字符串列；如果没有，则使用第一个源列，包含缺失分组，并按规范排序对键进行排序。数值摘要针对每个数值型字段返回非缺失计数、最小值、最大值和总和；对于空列，总和为零。排序返回按所选字段排序的前 20 行；如果字段值相同，则按原始行顺序排列，缺失键置于最后。数值过滤选择字段值高于其中位数下限的行索引；并集过滤则为出现频率最高的非缺失值添加类别谓词。字段选择遵循模式顺序，优先选择具有变化的数值型字段，并选择日期用于排序。规范排序用于打破类别频率并列。

C.2. 提案生成

机制研究使用零温度下的 DeepSeek-V4-Pro，并设置 8,192 个令牌的补全额度。提示词包含请求、类型化模式、记录数量和前五条记录。每个请求都有一个已完成的提案。在 64 个输出中，63 个有效且与类型化参考相匹配。无效的发展指标过滤响应结合了不受支持的谓词和排序字段；它导致每种方法均出现一次流水线失败。这些请求报告了 54,216 个输入与补全令牌，并包含两次速率限制重试。十四部署面板记录于第 F 节。

C.3. 测量与统计单位

三项研究均使用两个接口、执行预算 $C = 3/5$、空缓存以及 30 秒的软编译时限。主要任务进行了两次新进程重复。方法队列经过随机化，八个单线程工作线程分别分配到不同的物理核心。运行时使用 Python 3.12.13、pandas 3.0.2 和 DuckDB 1.5.5。Python 哈希种子为 2070；原生查询线程和 BLAS 查询线程均限制为一个。DuckDB 连接会禁用自动线程固定，并从创建时起使用一个查询线程。计时前后都会检查线程亲和性。

生命周期时间包括评估器设置、参考构造、初始策略验证、采集、搜索、策略序列化和清理。解释器启动和提案生成在此生命周期之外记录。原子操作和清理可能会超过软时限，实际持续时间均予以保留。墙钟时间还附有 CPU 时间和效果查询计数值。完整表审计和新的发布重放在编译后进行。

完成度按所有提案计算，包含将无效提案计为零的情况。原生成本和独立最优性比较仅针对模式有效提案。运行时证书百分比的分母包含无效提案。每个来源贡献其在各项操作和重复实验中的均值，因此每个来源的权重相同。配对区间基于对八个来源均值进行的 5,000 次自助法重采样，随机种子为 38117。共享的原生输入和计时重复实验仍保留在各自的源集群内。

C.4. 独立检查

审计获取完整的原生效果表，重新执行探测，并使用独立的动态规划求解约束目标函数。初始策略和最终策略均使用全新的发布状态进行重放。重放会检查尝试的请求、观测、成功指标、调用次数、回执和已提交内容。每个保存的 CAPE 见证也会根据该时点已经可用的效果、其原始必需集合，以及其下界值与上界值之间的独立最优值进行检查。

机制研究解决了全部 252 个原生目标函数，检查了 8,970 个已获取单元见证，并重放了 586 个不同的策略对。面板解决了涵盖 152 个原生模型的 304 个目标函数。匹配研究验证了 32,182 个历史见证和 414 个不同的策略对，涵盖 61,440 个原生回合和 133,066 次调用。经仪器化的成本配置具有独立的计时记录和质量记录。

D. 完整机制研究结果

表格 7–10 保留全部八种方法。成功率表示流水线完成，调用次数按配置计算，查询次数为校准评估次数，时间为编译生命周期。认证率是所有提案中的运行时证书率；最优率是可执行提案中达到独立计算的完整目录最优值的比例。所有均值均对八个来源进行平衡。

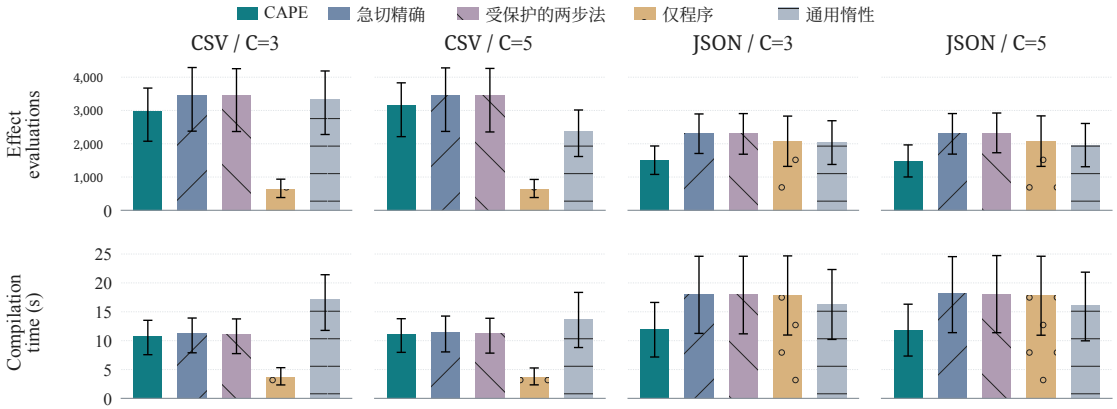

图 5。**与急切搜索相比，局部获取可减少工作量。** DeepSeek 机制研究：源平衡均值和 95% 来源聚类区间，各行使用共享刻度。时间包括设置、校准、搜索和清理。

表 7。CSV，$C = 3$。

方法	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	85.16	2.196	2980.5	10.774	98.44	100.00
急切精确	85.16	2.196	3450.9	11.173	98.44	100.00
仅程序	76.76	2.022	624.0	3.631	34.38	35.04
受保护的两步法	83.98	2.474	3450.9	11.121	34.38	35.04
通用惰性	83.74	2.198	3332.5	17.089	93.75	95.31
程序优先扩展	85.16	2.196	3307.3	11.402	98.44	100.00
整行获取	85.16	2.196	3243.6	11.311	98.44	100.00
无紧界	85.16	2.196	3330.3	11.374	98.44	100.00

表8. CSV，$C = 5$。

方法	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	98.44	2.348	3161.1	11.201	98.44	100.00
急切精确	98.44	2.348	3450.9	11.417	98.44	100.00
仅程序	92.19	2.365	624.0	3.624	34.38	75.00
受保护的两步法	98.44	2.838	3450.9	11.192	34.38	70.31
通用惰性	84.18	2.534	2382.0	13.700	73.44	75.00
程序优先扩展	98.44	2.348	3191.6	11.572	98.44	100.00
整行获取	98.44	2.348	3161.0	11.177	98.44	100.00
无紧界	98.44	2.348	3161.1	11.217	98.44	100.00

表 9. JSON， $C = 3$。

方法	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	94.74	1.476	1506.2	11.919	95.31	98.44
急切精确	94.60	1.474	2309.0	18.021	84.38	98.44
仅程序	94.60	1.474	2075.7	17.792	77.34	98.44
受保护的两步法	94.60	1.493	2307.2	18.008	77.34	91.96
通用惰性	92.90	1.495	2029.5	16.200	93.75	95.31
程序优先扩展	94.60	1.474	2170.0	17.878	85.94	98.44
整行获取	90.48	1.430	2354.3	18.003	85.16	86.72
无紧界	94.74	1.476	1516.4	11.921	95.31	98.44

表10。JSON， $C = 5$。

方法	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	94.74	1.475	1485.9	11.903	96.88	100.00
急切精确	94.60	1.474	2317.8	18.079	84.38	98.44
仅程序	94.60	1.474	2077.9	17.744	76.56	98.44
受保护的两步法	94.60	1.493	2314.1	18.005	77.34	91.96
通用惰性	92.90	1.546	1954.9	16.008	93.75	95.31
程序优先扩展	94.60	1.474	2174.5	17.919	83.59	98.44
整行获取	91.05	1.435	2049.3	17.036	86.72	89.84
无紧界	94.74	1.475	1486.2	11.862	96.88	100.00

表11。机制研究的配对对比。正差异表示 CAPE 更优；调用次数需结合完成情况解读。区间对应节省的秒数和重采样来源。

接口， C	对照组	增益（百分点）	节省的调用次数	节省的秒数	95%区间
CSV，3	急切精确	0.00	0.000	0.399	[0.041, 0.768]
CSV，3	仅程序	8.40	-0.174	-7.143	[-9.004, -4.786]
CSV, 3	受保护的两步法	1.17	0.278	0.348	[-0.054, 0.713]
CSV, 3	通用惰性	1.42	0.001	6.316	[3.972，8.385]
CSV，3	程序优先扩展	0.00	0.000	0.628	[-0.076，1.255]
CSV，3	整行获取	0.00	0.000	0.538	[-0.071, 1.050]
CSV，3	无紧界	0.00	0.000	0.601	[0.113, 1.076]
JSON，3	急切精确	0.14	-0.002	6.102	[4.074，8.107]
JSON，3	仅程序	0.14	-0.002	5.873	[3.659，8.080]
JSON，3	受保护的两步法	0.14	0.017	6.088	[4.061，8.108]
JSON, 3	通用惰性	1.85	0.019	4.281	[2.827, 5.757]
JSON, 3	程序优先扩展	0.14	-0.002	5.959	[3.876，8.132]
JSON，3	整行获取	4.26	-0.046	6.084	[4.122，8.051]
JSON，3	无紧界	0.00	0.000	0.002	[-0.022, 0.026]
CSV，5	急切精确	0.00	0.000	0.216	[-0.125，0.693]
CSV，5	仅程序	6.25	0.017	-7.577	[-9.510, -5.041]
CSV，5	受保护的两步法	0.00	0.490	-0.010	[-0.558, 0.562]
CSV，5	通用惰性	14.26	0.187	2.498	[0.342，4.938]
CSV，5	程序优先扩展	0.00	0.000	0.371	[0.146，0.729]
CSV，5	整行获取	0.00	0.000	-0.024	[-0.124, 0.080]
CSV，5	无紧界	0.00	0.000	0.015	[-0.033, 0.065]
JSON，5	急切精确	0.14	-0.001	6.176	[4.131，8.157]
JSON，5	仅程序	0.14	-0.001	5.842	[3.735，8.026]
JSON，5	受保护的两步法	0.14	0.017	6.103	[4.136, 8.141]
JSON, 5	通用惰性	1.85	0.071	4.105	[2.774, 5.458]
JSON, 5	程序优先扩展	0.14	-0.001	6.016	[3.899，8.168]
JSON，5	整行获取	3.69	-0.040	5.134	[2.801，7.523]
JSON，5	无紧界	0.00	0.000	-0.041	[-0.100, 0.016]

D.1. 诊断余量与局部获取

完整模型诊断比较在有探测和无探测两种情况下，分别对相同的请求、必需集合和调用预算进行优化。图 6 展示了完成余量。其完整目录最优值包含每个程序和探针效果。仅程序最优值保留程序观测和重试策略。CSV 包含诊断操作提高可实现完成度的实例；JSON 中的完成最优值彼此一致。

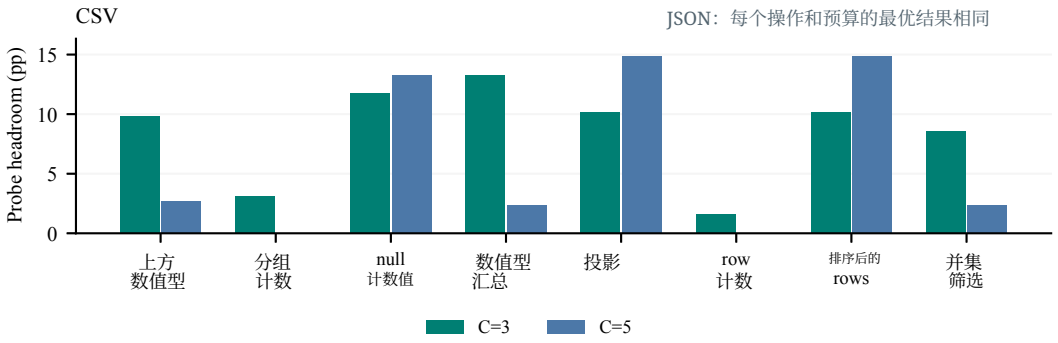

图6. **诊断信息可提升 CSV 上的可实现完成度**。 使用完整效应计算的全目录最优完成度，减去独立计算的仅程序最优完成度。这一比较区分了动作空间余量与中断搜索。

在定时 CAPE 运行中，380 个编译任务在严格后代作用域上获取效果。在 41,626 个获取块中，有 23,180 个是本地的，并包含 308,858 次效应查询。完整行消融测量将这些获取块扩展到未使用配置单元格的成本。图7给出了源集群比较。

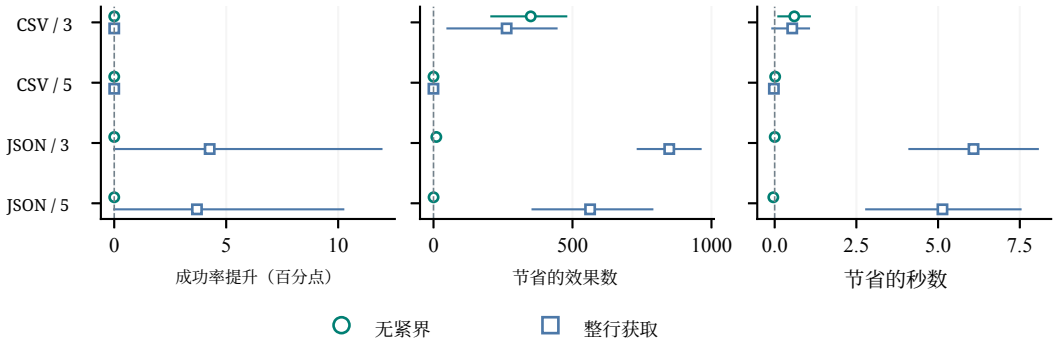

图 7. **作用域闭包与界限收紧影响的是不同工作。** 机制研究消融实验用整行获取替代局部闭包，或用廉价支持界替代选定收紧。配对区间通过对源集群重新采样得出。

D.2. 策略进展与证明完成

图 8 保留了 CSV 投影和 JSON 数值摘要的开发轨迹。对于 CSV 请求，可行策略在完成界相交前完成了 56 项配置。对于 JSON 请求，完成界先于调用界相交；随后通过采集弥合调用界差距。这些轨迹展示了改进可执行策略与证明其最终值之间的区别。

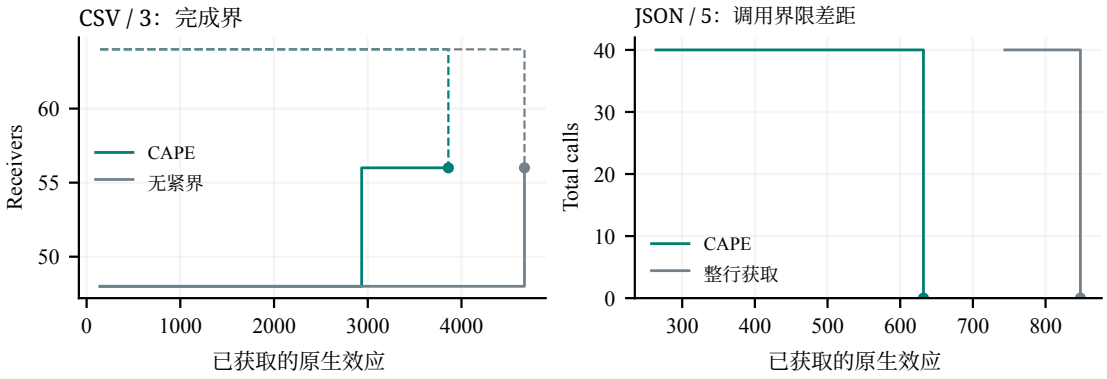

图 8。**策略改进可能先于证明完成。**保留的开发轨迹显示了已获取效应以及较低/较高的策略值。这些轨迹展示了所选机制的进展；主要比较采用单独的冻结测量研究。

E. 匹配的获取计划与界计算

匹配研究使用了 304 个精确的任务/程序/接口/预算组。每组接受六种方法和两次全新的过程主要重复实验，共产生 3,648 次测量。每种方法和每个分组另进行一次插桩运行，共产生 1,824 份性能剖析。实验机器为 AMD EPYC 7742。源加权、自助法单元和生命周期核算遵循第 C 节。

CAPE/选择性方法取得了 606/608 个物理全目录最优解，而匹配的 AND/OR/选择性方法取得了 595/608 个。CAPE 的两次未命中在 $C = 3$ 的 JSON 天气投影上实现了最优完成并减少了多余调用。下表使用 DeepSeek-V4-Pro 的提议绑定。它们保留了六种配置及其带符号配对对比。

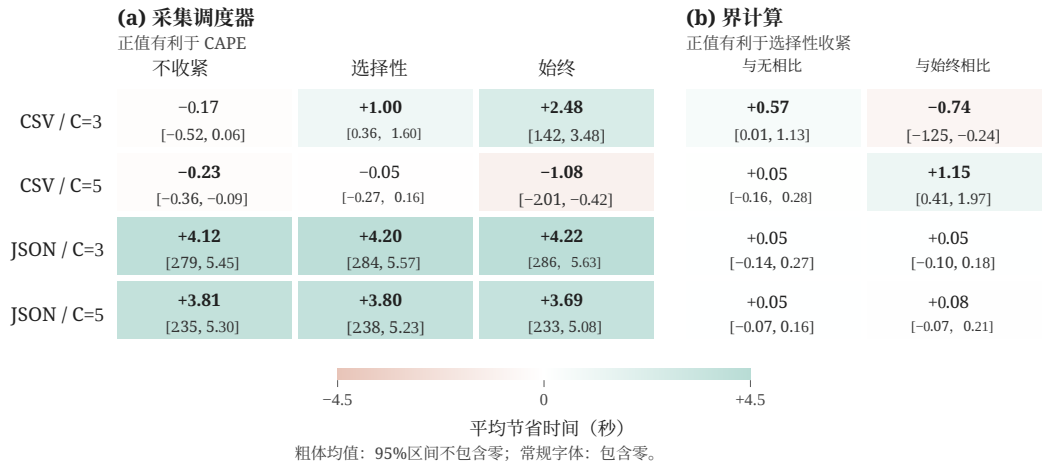

图9。**配对效果区分了采集与收紧**。（a）在相同界限模式下，CAPE 与匹配的与/或进行比较；（b）选择性收紧与不收紧/始终收紧进行比较。单元格给出平均节省的秒数和95%源聚类区间。正值表示（a）中 CAPE 更优，以及（b）中选择性计算更优。

表12。CSV 上匹配的主要比较，$C = 3$。

调度器	界	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	不收紧	85.16	2.196	3330.3	11.489	98.44	100.00
CAPE	选择性	85.16	2.196	2979.9	10.918	97.66	100.00
CAPE	始终	85.16	2.196	2546.4	10.177	98.44	100.00
与/或	不收紧	85.16	2.196	3278.6	11.315	98.44	100.00
与/或	选择性	85.16	2.196	3250.5	11.916	98.44	100.00
与/或	始终	85.16	2.196	3249.7	12.654	98.44	100.00

表13。在 CSV、$C = 5$ 上的主要比较匹配结果。

调度器	界	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	不收紧	98.44	2.348	3161.1	11.374	98.44	100.00
CAPE	选择性	98.44	2.348	3161.1	11.319	98.44	100.00
CAPE	始终	98.44	2.348	3148.1	12.464	93.75	100.00
与/或	不收紧	98.44	2.348	3163.4	11.148	98.44	100.00
与/或	选择性	98.44	2.348	3163.4	11.265	98.44	100.00
与/或	始终	98.44	2.348	3163.4	11.381	98.44	100.00

表14。JSON 上的主要匹配比较，$C = 3$。

调度器	界	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	不收紧	94.74	1.477	1505.7	12.007	95.31	97.66
CAPE	选择性	94.74	1.476	1511.9	11.961	96.88	98.44
CAPE	始终	94.74	1.476	1505.0	12.012	95.31	98.44
与/或	不收紧	94.39	1.481	1980.3	16.130	92.97	94.53
与/或	选择性	94.39	1.478	1990.5	16.161	92.97	94.53
与/或	始终	94.46	1.483	1975.3	16.235	93.75	95.31

表15。JSON、 $C = 5$ 上的匹配主要比较。

调度器	界	成功率（%）	调用次数	查询次数	时间（秒）	精确率（%）	最优性（%）
CAPE	不收紧	94.74	1.475	1494.0	11.975	97.66	100.00
CAPE	选择性	94.74	1.475	1490.5	11.929	96.88	100.00
CAPE	始终	94.74	1.475	1473.9	12.005	96.09	100.00
与/或	不收紧	94.50	1.483	1941.7	15.784	93.75	95.31
与/或	选择性	94.53	1.483	1949.4	15.728	93.75	95.31
与/或	始终	94.42	1.480	1932.6	15.699	93.75	95.31

表16。CSV 配对对比。C 和 A 分别表示 CAPE 和匹配的与/或；N、S 和 T 分别表示不收紧、选择性收紧和始终收紧。正差值表示有利于所列候选项。

C	候选项	对照组	增益（百分点）	节省的查询次数	节省的秒数	95%区间
3	C-N	A-N	0.000	-51.7	-0.173	[-0.522, 0.061]
3	C-S	A-S	0.000	270.6	0.998	[0.358, 1.596]
3	C-T	A-T	0.000	703.4	2.477	[1.420, 3.479]
3	C-S	C-N	0.000	350.4	0.571	[0.009, 1.130]
3	C-S	C-T	0.000	-433.6	-0.741	[-1.255, -0.242]
3	A-S	A-N	0.000	28.1	-0.600	[-0.869, -0.291]
3	A-S	A-T	0.000	-0.8	0.738	[0.512, 0.925]
5	C-N	A-N	0.000	2.2	-0.226	[-0.357, -0.095]
5	C-S	A-S	0.000	2.2	-0.054	[-0.273, 0.163]
5	C-T	A-T	0.000	15.2	-1.082	[-2.013, -0.421]
5	C-S	C-N	0.000	0.0	0.055	[-0.160, 0.278]
5	C-S	C-T	0.000	-13.0	1.145	[0.413, 1.966]
5	A-S	A-N	0.000	0.0	-0.117	[-0.275, 0.039]
5	A-S	A-T	0.000	0.0	0.117	[-0.134, 0.391]

表17。JSON配对对比。完成度增益以百分点表示；时间为生命周期秒数。两个选择性调度器的完成度增益区间包含零。

C	候选项	对照组	增益 (pp)	节省的查询次数	节省的秒数	95%区间
3	C-N	A-N	0.355	474.6	4.123	[2.785, 5.446]
3	C-S	A-S	0.355	478.6	4.200	[2.839, 5.570]
3	C-T	A-T	0.284	470.3	4.223	[2.860, 5.630]
3	C-S	C-N	0.000	-6.2	0.046	[-0.138, 0.268]
3	C-S	C-T	0.000	-6.9	0.051	[-0.097, 0.176]
3	A-S	A-N	0.000	-10.2	-0.031	[-0.203, 0.145]
3	A-S	A-T	-0.071	-15.3	0.074	[-0.103, 0.239]
5	C-N	A-N	0.249	447.8	3.809	[2.348, 5.302]
5	C-S	A-S	0.213	458.9	3.799	[2.377, 5.228]
5	C-T	A-T	0.320	458.7	3.694	[2.332, 5.081]
5	C-S	C-N	0.000	3.5	0.046	[-0.070, 0.164]
5	C-S	C-T	0.000	-16.6	0.076	[-0.066, 0.210]
5	A-S	A-N	0.036	-7.7	0.056	[-0.100, 0.207]
5	A-S	A-T	0.107	-16.9	-0.029	[-0.217, 0.151]

E.1. 独立成本剖面

性能分析器采用独占嵌套挂钟时间核算。在 T_2 内部后代界调用所花费的时间计入紧界。其他支持/动作界和图搜索各自单独计入。编译器其他开销包括初始验证和簿记；生命周期其他开销包括最终编码、清理和测量开销。插桩可能改变

截止时间终止，因此这些运行保留了各自的质量记录。在 1,824 个性能剖析中，有 1,805 个在成功率一致性和调用总量方面与两次主要重复实验均匹配。

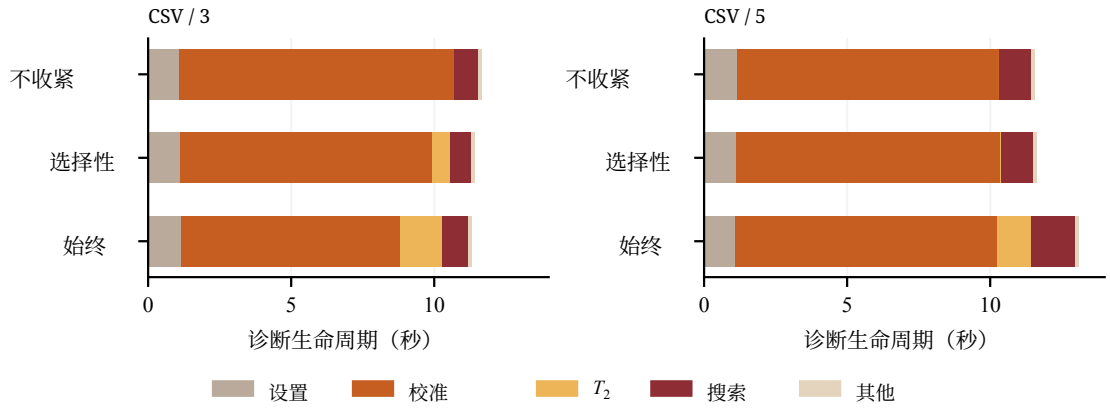

图10。**收紧以增加计算量为代价换取校准**。 使用 DeepSeek-V4-Pro 绑定的独立 CSV 诊断运行。在 $C = 3$，始终收紧可减少校准工作量，足以抵消其增加的计算量。在 $C = 5$，额外的界限计算时间伴随着相近的校准工作量。

表18。独占诊断成本（秒）。搜索包括图计算、其他界限计算和完整模型求解。Agree 将每个性能剖析的值与两次主要重复实验进行比较； Δ queries 表示与主要均值的差值。方法缩写遵循对比表中的用法。

设置	方法	设置	校准	T_2	搜索	其他	同意（%）	Δ 查询次数
CSV/3	C-N	1.103	9.612	0.000	0.844	0.110	100.00	0.0
CSV/3	C-S	1.122	8.814	0.621	0.747	0.114	100.00	0.5
CSV/3	C-T	1.149	7.658	1.468	0.925	0.108	100.00	-1.8
CSV/3	A-N	1.114	9.492	0.000	0.696	0.108	100.00	0.0
CSV/3	A-S	1.139	9.571	1.290	0.617	0.118	100.00	0.0
CSV/3	A-T	1.157	9.620	2.266	0.653	0.114	96.88	-18.9
CSV/5	C-N	1.152	9.155	0.000	1.109	0.134	100.00	0.0
CSV/5	C-S	1.101	9.259	0.019	1.115	0.134	100.00	0.0
CSV/5	C-T	1.101	9.148	1.192	1.539	0.133	100.00	-39.3
CSV/5	A-N	1.121	9.265	0.000	1.025	0.133	100.00	0.0
CSV/5	A-S	1.110	9.243	0.000	1.043	0.137	100.00	0.0
CSV/5	A-T	1.111	9.315	0.265	1.040	0.136	96.88	-8.0
JSON/3	C-N	0.431	11.307	0.000	0.349	0.093	98.44	3.0
JSON/3	C-S	0.454	11.139	0.070	0.276	0.094	100.00	-11.6
JSON/3	C-T	0.435	11.183	0.100	0.277	0.094	98.44	-20.4
JSON/3	A-N	0.446	15.561	0.000	0.312	0.096	93.75	10.8
JSON/3	A-S	0.424	15.541	0.146	0.212	0.097	95.31	-10.1
JSON/3	A-T	0.418	15.581	0.154	0.214	0.097	96.88	3.0
JSON/5	C-N	0.414	11.282	0.000	0.326	0.096	100.00	-8.9
JSON/5	C-S	0.421	11.364	0.004	0.326	0.097	100.00	-5.4
JSON/5	C-T	0.428	11.336	0.006	0.342	0.096	100.00	8.0
JSON/5	A-N	0.455	14.989	0.000	0.295	0.098	98.44	-29.0
JSON/5	A-S	0.420	15.188	0.000	0.303	0.098	98.44	-6.4
JSON/5	A-T	0.427	15.025	0.000	0.309	0.098	96.88	15.2

F. 提案模型与完整面板结果

面板包含四种闭源部署和十种开放权重部署。每种部署都接收相同的 64 个请求。提示词指定了操作模式、类型化字段、记录数量和前五条记录，并设有 8,192 个 token 的补全额度。GPT 部署采用低推理和默认采样；Qwen3.5 使用非思考模式和零温度；Kimi-K2.5 使用其非推理选项。其他部署使用零温度。视觉语言模型接收相同的纯文本请求。表19记录了推理设置。

表 19. 提案部署和推理设置。默认表示未指定推理选项 .

模型	推理设置
GPT-6 Astra	low
GPT-5.6 Terra	low
GPT-5.6 Luna	low
Grok 4.6	默认
DeepSeek-V4-Pro	默认
DeepSeek-V4-Flash	默认
Kimi-K2.5	none
Qwen3.5-122B-A10B	非思考
Llama-3.3-70B-Instruct	默认
Qwen3.5-27B	非思考
Qwen3.5-9B	非思考
Llama-3.1-8B-Instruct	默认
Qwen2.5-VL-7B-Instruct	默认
Qwen3-VL-4B-Instruct	默认

表20. 提案结果 a 以及报告的令牌使用量。每个部署包含 64 个请求。参考一致性评估所提出的 d 对类型化数据执行操作。

模型	访问权限	有效 / 64	参考 / 64	令牌
GPT-6 Astra	已关闭	64	64	52,099
GPT-5.6 Terra	已关闭	64	64	52,291
GPT-5.6 Luna	已关闭	64	64	53,843
Grok 4.6	已关闭	64	64	87,453
DeepSeek-V4-Pro	Open	63	63	54,216
DeepSeek-V4-Flash	Open	64	64	53,673
Kimi-K2.5	Open	64	64	52,926
Qwen3.5-122B-A10B	Open	64	64	57,984
Llama-3.3-70B-Instruct	Open	62	62	54,697
Qwen3.5-27B	Open	64	64	57,461
Qwen3.5-9B	Open	64	64	57,700
Llama-3.1-8B-Instruct	Open	47	41	57,619
Qwen2.5-VL-7B-Instruct	Open	62	59	56,554
Qwen3-VL-4B-Instruct	Open	64	61	56,337

该面板保留了 64 条 DeepSeek-V4-Pro 响应，并新增了 832 条观测到的已完成响应。其中有 874 个符合模式的有效提案，以及 862 个类型化引用协议。每条已完成响应仅使用一次，包括结果无效的响应在内。该集合记录了 18 次传输错误，以及 2 个没有观测到响应的中断请求。新的已完成请求报告了 750,637 个令牌；计入保留的 Pro 响应后，总数为 804,853 个令牌。

精确输入去重匹配任务字节、批次字节、模式、请求操作、提议操作、后端和调用限额。该面板包含 4,864 次全新的物理测量和 57,344 条提供商绑定记录，其中包括失败提案。这些绑定会在输入一致时，为同一个测量的计算问题附加不同的提案来源标签。来源仍是统计重采样单位。

原生审计涵盖 152 个输入模型、33 个归一化动作结构和 655 对不同的策略对。它解决了全部 304 个独立目标函数，并检查了 10,225 个已获取单元见证对象、94,160 个原生回合和 218,266 次调用。CAPE 达到了 608 个物理最优值中的 606 个，以及与参考一致的提案中的 528 个最优值中的 526 个。符合模式但错误的操作可能具有零可实现完成度；独立最优性针对提议操作和请求输出保护条件进行评估。

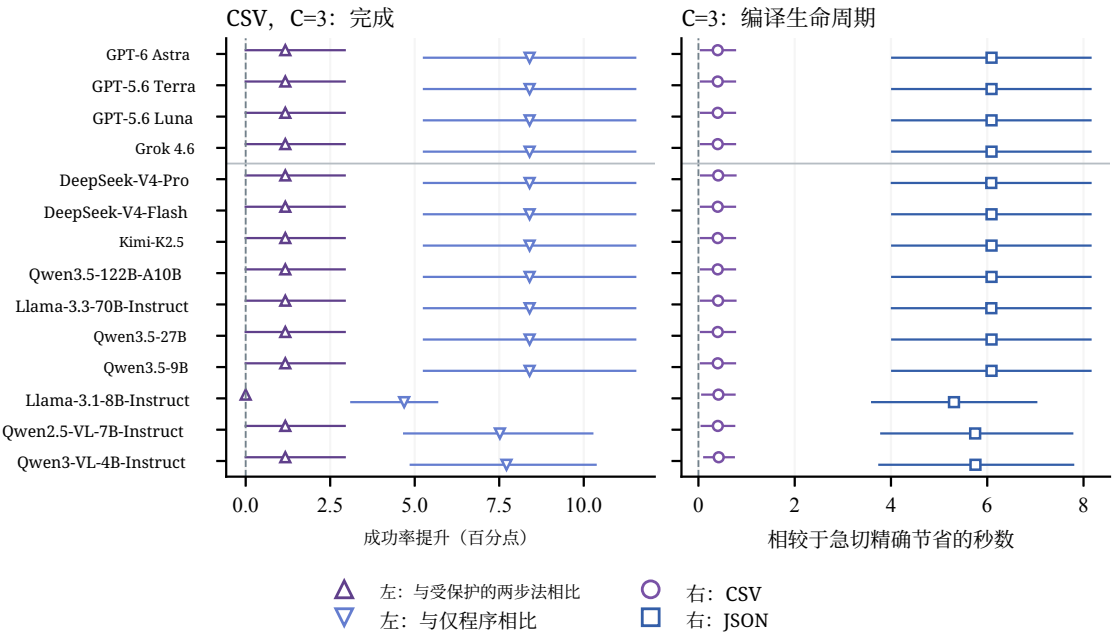

图 11. **提案来源会改变完成情况和编译器输入**。在 $C = 3$ 条件下按提供方进行比较。正值表示 CAPE 更优。完成情况涵盖所有提案；原生成本仅针对可执行的提案。区间通过重采样来源得出，并将共享的物理测量保留在各自的源集群内。

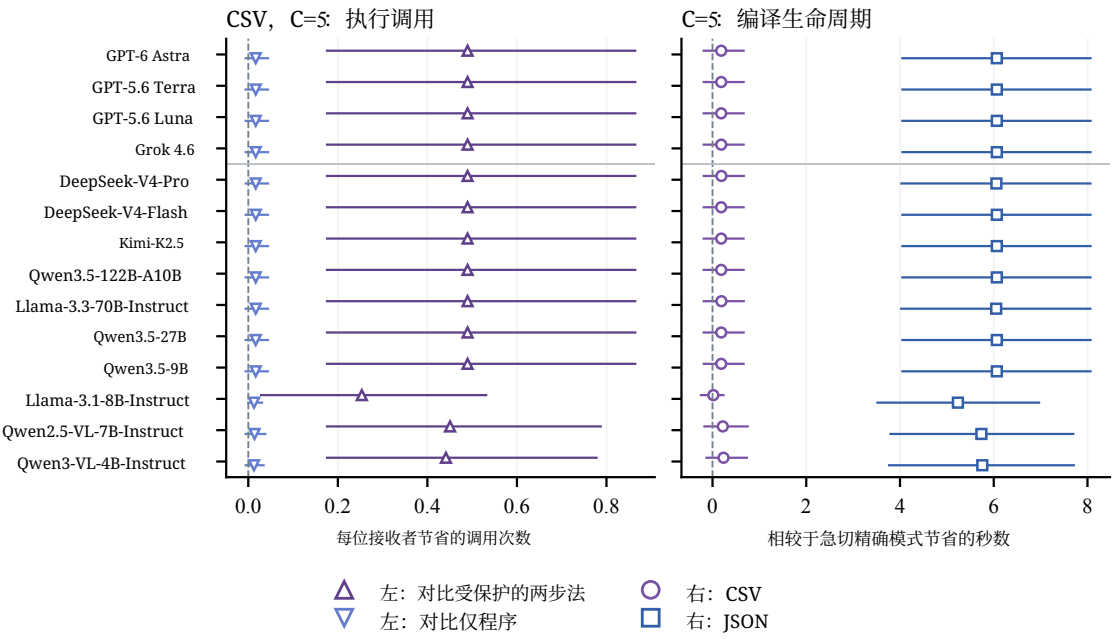

图12。**CAPE 可减少 CSV 执行调用次数**。在 $C = 5$ 条件下按提供方条件化的比较，其中 CAPE 和受保护的两步法具有相同的完成率。完整表格给出了每种方法的完成率和原生成本。

F.1. 按不同输入组划分的完整结果

九个部署对每个请求都会生成相同的可执行输入：GPT-6 Astra、GPT-5.6 Terra、GPT-5.6 Luna、Grok 4.6、DeepSeek-V4-Flash、Kimi-K2.5、Qwen3.5-122B-A10B、Qwen3.5-27B 和 Qwen3.5-9B。它们的完整原生结果汇总在一个表中。其余五个部署各自使用单独的表。成功率和证书率包含所有提案；调用次数、查询次数、生命周期和独立获得的最优解则以可执行提案为条件。原始的按部署划分的数值来源均予以保留。

表 21. 九个具有相同输入的部署的共享原生测量。

接口， C	方法	成功率（%）	调用次数	查询次数	时间（秒）	认证率（%）	最优性（%）
CSV, 3	CAPE	86.72	2.188	2948.4	10.655	100.00	100.00
CSV, 3	急切精确	86.72	2.188	3412.0	11.056	100.00	100.00
CSV, 3	仅程序	78.32	2.016	624.0	3.611	35.94	35.94
CSV, 3	受保护的两步法	85.55	2.461	3412.0	11.004	35.94	35.94
CSV, 3	通用惰性	85.21	2.188	3293.3	16.868	94.53	94.53
CSV, 3	程序优先扩展	86.72	2.188	3268.5	11.268	100.00	100.00
CSV, 3	整行获取	86.72	2.188	3204.5	11.131	100.00	100.00
CSV, 3	无紧界	86.72	2.188	3291.0	11.191	100.00	100.00
CSV, 5	CAPE	100.00	2.338	3124.6	11.060	100.00	100.00
CSV, 5	急切精确	100.00	2.338	3412.0	11.246	100.00	100.00
CSV, 5	仅程序	93.75	2.354	624.0	3.618	35.94	75.00
CSV, 5	受保护的两步法	100.00	2.828	3412.0	11.051	35.94	70.31
CSV, 5	通用惰性	85.74	2.524	2363.8	13.618	75.00	75.00
CSV, 5	程序优先扩展	100.00	2.338	3155.6	11.361	100.00	100.00
CSV, 5	整行获取	100.00	2.338	3124.4	11.019	100.00	100.00
CSV, 5	无紧界	100.00	2.338	3124.6	11.062	100.00	100.00
JSON, 3	CAPE	96.31	1.475	1502.5	11.800	96.88	98.44
JSON, 3	急切精确	96.16	1.473	2319.9	17.889	85.16	98.44
JSON, 3	仅程序	96.16	1.473	2079.5	17.639	78.91	98.44
JSON, 3	受保护的两步法	96.09	1.491	2311.7	17.910	78.91	91.41
JSON, 3	通用惰性	94.46	1.494	2026.2	16.010	95.31	95.31
JSON, 3	程序优先扩展	96.16	1.473	2165.1	17.764	85.94	98.44
JSON, 3	整行获取	92.05	1.428	2363.2	17.851	88.28	88.28
JSON, 3	无紧界	96.31	1.475	1510.9	11.845	96.88	98.44
JSON, 5	CAPE	96.31	1.474	1489.0	11.837	99.22	100.00
JSON, 5	急切精确	96.16	1.473	2316.4	17.904	85.16	98.44
JSON, 5	仅程序	96.16	1.473	2081.3	17.633	78.12	98.44
JSON, 5	受保护的两步法	96.09	1.491	2311.7	17.899	79.69	91.41
JSON, 5	通用惰性	94.46	1.545	1952.8	15.787	95.31	95.31
JSON, 5	程序优先扩展	96.16	1.473	2174.6	17.769	86.72	98.44
JSON, 5	整行获取	92.61	1.434	2044.8	16.916	88.28	89.84
JSON, 5	无紧界	96.31	1.474	1493.0	11.803	100.00	100.00

表22。DeepSeek-V4-Pro：完整的全新面板测量结果。

接口， C	方法	成功率（%）	调用次数	查询次数	时间（秒）	认证率（%）	最优性（%）
CSV, 3	CAPE	85.16	2.196	2980.5	10.749	98.44	100.00
CSV，3	急切精确	85.16	2.196	3450.9	11.159	98.44	100.00
CSV，3	仅程序	76.76	2.022	624.0	3.614	34.38	35.04
CSV, 3	受保护的两步法	83.98	2.474	3450.9	11.110	34.38	35.04
CSV, 3	通用惰性	83.64	2.197	3332.7	17.040	92.97	94.53
CSV，3	程序优先扩展	85.16	2.196	3307.3	11.380	98.44	100.00
CSV，3	整行获取	85.16	2.196	3243.6	11.238	98.44	100.00
CSV，3	无紧界	85.16	2.196	3330.3	11.302	98.44	100.00
CSV, 5	CAPE	98.44	2.348	3161.1	11.160	98.44	100.00
CSV，5	急切精确	98.44	2.348	3450.9	11.350	98.44	100.00
CSV，5	仅程序	92.19	2.365	624.0	3.619	34.38	75.00
CSV，5	受保护的两步法	98.44	2.838	3450.9	11.155	34.38	70.31
CSV, 5	通用惰性	84.18	2.534	2384.9	13.702	73.44	75.00
CSV, 5	程序优先扩展	98.44	2.348	3191.6	11.463	98.44	100.00
CSV，5	整行获取	98.44	2.348	3161.0	11.122	98.44	100.00
CSV，5	无紧界	98.44	2.348	3161.1	11.164	98.44	100.00
JSON, 3	CAPE	94.74	1.476	1507.5	11.815	95.31	98.44
JSON，3	急切精确	94.60	1.474	2326.9	17.899	83.59	98.44
JSON，3	仅程序	94.60	1.474	2079.5	17.642	77.34	98.44
JSON，3	受保护的两步法	94.53	1.492	2318.8	17.922	77.34	91.18
JSON，3	通用惰性	92.90	1.495	2031.7	16.026	93.75	95.31
JSON，3	程序优先扩展	94.60	1.474	2169.4	17.769	84.38	98.44
JSON，3	整行获取	90.48	1.429	2367.0	17.859	86.72	88.28
JSON，3	无紧界	94.74	1.476	1515.9	11.863	95.31	98.44
JSON, 5	CAPE	94.74	1.475	1494.8	11.854	97.66	100.00
JSON，5	急切精确	94.60	1.474	2323.5	17.911	83.59	98.44
JSON，5	仅程序	94.60	1.474	2081.3	17.636	76.56	98.44
JSON，5	受保护的两步法	94.53	1.492	2318.8	17.910	78.12	91.18
JSON，5	通用惰性	92.90	1.546	1958.4	15.805	93.75	95.31
JSON，5	程序优先扩展	94.60	1.474	2179.3	17.779	85.16	98.44
JSON，5	整行获取	91.05	1.435	2052.7	16.929	86.72	89.84
JSON，5	无紧界	94.74	1.475	1498.8	11.811	98.44	100.00

表23。Llama-3.3-70B-Instruct：完整面板测量。

接口， C	方法	成功率（%）	调用次数	查询次数	时间（秒）	认证率（%）	最优性（%）
CSV, 3	CAPE	83.59	2.196	2980.5	10.750	96.88	100.00
CSV，3	急切精确	83.59	2.196	3450.9	11.158	96.88	100.00
CSV，3	仅程序	75.20	2.022	624.0	3.614	32.81	35.04
CSV，3	受保护的两步法	82.42	2.474	3450.9	11.111	32.81	35.04
CSV，3	通用惰性	82.08	2.197	3332.7	17.040	91.41	94.53
CSV，3	程序优先扩展	83.59	2.196	3307.3	11.382	96.88	100.00
CSV，3	整行获取	83.59	2.196	3243.6	11.238	96.88	100.00
CSV，3	无紧界	83.59	2.196	3330.3	11.304	96.88	100.00
CSV, 5	CAPE	96.88	2.348	3161.1	11.160	96.88	100.00
CSV，5	急切精确	96.88	2.348	3450.9	11.350	96.88	100.00
CSV, 5	仅程序	90.62	2.365	624.0	3.620	32.81	75.00
CSV, 5	受保护的两步法	96.88	2.838	3450.9	11.154	32.81	70.31
CSV，5	通用惰性	82.62	2.534	2384.9	13.702	71.88	75.00
CSV，5	程序优先扩展	96.88	2.348	3191.6	11.463	96.88	100.00
CSV，5	整行获取	96.88	2.348	3161.0	11.123	96.88	100.00
CSV，5	无紧界	96.88	2.348	3161.1	11.165	96.88	100.00
JSON, 3	CAPE	93.18	1.476	1507.2	11.812	93.75	98.44
JSON, 3	急切精确	93.04	1.474	2326.9	17.896	82.03	98.44
JSON, 3	仅程序	93.04	1.474	2079.5	17.641	75.78	98.44
JSON，3	受保护的两步法	92.97	1.492	2318.8	17.919	75.78	91.18
JSON，3	通用惰性	91.34	1.495	2031.7	16.024	92.19	95.31
JSON, 3	程序优先扩展	93.04	1.474	2169.4	17.769	82.81	98.44
JSON, 3	整行获取	88.92	1.429	2367.0	17.856	85.16	88.28
JSON，3	无紧界	93.18	1.476	1515.6	11.860	93.75	98.44
JSON, 5	CAPE	93.18	1.475	1494.7	11.855	96.09	100.00
JSON，5	急切精确	93.04	1.474	2323.5	17.907	82.03	98.44
JSON，5	仅程序	93.04	1.474	2081.3	17.635	75.00	98.44
JSON，5	受保护的两步法	92.97	1.492	2318.8	17.908	76.56	91.18
JSON，5	通用惰性	91.34	1.546	1958.4	15.802	92.19	95.31
JSON，5	程序优先扩展	93.04	1.474	2179.3	17.776	83.59	98.44
JSON，5	整行获取	89.49	1.435	2052.7	16.930	85.16	89.84
JSON，5	无紧界	93.18	1.475	1498.8	11.812	96.88	100.00

表 24. Llama-3.1-8B-Instruct：完整的面板测量结果。

接口， C	方法	成功率（%）	调用次数	查询次数	时间（秒）	认证率（%）	最优性（%）
CSV, 3	CAPE	57.81	1.798	2327.5	8.728	73.44	100.00
CSV，3	急切精确	57.81	1.798	2695.7	9.144	73.44	100.00
CSV，3	仅程序	53.12	1.666	624.0	3.647	39.06	52.40
CSV，3	受保护的两步法	57.81	2.001	2695.7	9.123	39.06	52.40
CSV，3	通用惰性	56.84	1.801	2581.0	13.167	70.31	96.43
CSV，3	程序优先扩展	57.81	1.798	2680.4	9.596	73.44	100.00
CSV，3	整行获取	57.81	1.798	2609.7	9.380	73.44	100.00
CSV，3	无紧界	57.81	1.798	2645.5	9.405	73.44	100.00
CSV, 5	CAPE	64.06	1.894	2547.4	9.265	73.44	100.00
CSV，5	急切精确	64.06	1.894	2695.7	9.282	73.44	100.00
CSV，5	仅程序	61.52	1.906	624.0	3.651	39.06	86.01
CSV，5	受保护的两步法	64.06	2.147	2695.7	9.156	39.06	84.23
CSV，5	通用惰性	59.18	2.017	1875.8	10.467	65.62	86.01
CSV，5	程序优先扩展	64.06	1.894	2581.4	9.526	73.44	100.00
CSV，5	整行获取	64.06	1.894	2547.3	9.272	73.44	100.00
CSV，5	无紧界	64.06	1.894	2547.4	9.264	73.44	100.00
JSON, 3	CAPE	62.43	1.248	1550.0	12.329	73.44	100.00
JSON，3	急切精确	62.29	1.247	2266.0	17.638	63.28	98.21
JSON，3	仅程序	62.29	1.247	2080.8	17.494	60.16	98.21
JSON, 3	受保护的两步法	62.29	1.268	2263.2	17.651	60.16	92.49
JSON, 3	通用惰性	61.72	1.256	2007.8	15.904	71.88	98.21
JSON，3	程序优先扩展	62.29	1.247	2112.3	17.492	63.28	98.21
JSON，3	整行获取	59.16	1.211	2268.1	17.657	64.84	90.18
JSON，3	无紧界	62.43	1.248	1551.0	12.390	73.44	100.00
JSON, 5	CAPE	62.43	1.248	1507.7	12.402	73.44	100.00
JSON，5	急切精确	62.29	1.247	2259.8	17.638	63.28	98.21
JSON，5	仅程序	62.29	1.247	2082.1	17.466	59.38	98.21
JSON, 5	受保护的两步法	62.29	1.268	2262.7	17.624	60.94	92.49
JSON, 5	通用惰性	61.72	1.273	1942.6	15.782	71.88	98.21
JSON，5	程序优先扩展	62.29	1.247	2115.2	17.496	64.84	98.21
JSON，5	整行获取	59.73	1.217	1986.2	16.801	66.41	91.96
JSON，5	无紧界	62.43	1.248	1507.7	12.422	73.44	100.00

表 25. Qwen2.5-VL-7B-Instruct：完整面板测量。

接口， C	方法	成功率（%）	调用次数	查询次数	时间(s)	认证率（%）	最优性（%）
CSV, 3	CAPE	80.47	2.060	2752.8	10.046	96.88	100.00
CSV，3	急切精确	80.47	2.060	3188.6	10.449	96.88	100.00
CSV，3	仅程序	72.95	1.903	624.0	3.596	39.06	41.07
CSV，3	受保护的两步法	79.30	2.314	3188.6	10.401	39.06	41.07
CSV，3	通用惰性	79.39	2.061	3085.9	15.851	92.97	95.98
CSV，3	程序优先扩展	80.47	2.060	3045.0	10.612	96.88	100.00
CSV，3	整行获取	80.47	2.060	2981.7	10.486	96.88	100.00
CSV，3	无紧界	80.47	2.060	3067.1	10.549	96.88	100.00
CSV, 5	CAPE	92.19	2.199	2909.1	10.430	96.88	100.00
CSV，5	急切精确	92.19	2.199	3188.6	10.649	96.88	100.00
CSV，5	仅程序	86.82	2.213	624.0	3.613	39.06	77.46
CSV，5	受保护的两步法	92.19	2.650	3188.6	10.452	39.06	72.77
CSV，5	通用惰性	79.98	2.360	2229.5	12.839	75.00	77.46
CSV，5	程序优先扩展	92.19	2.199	2940.5	10.665	96.88	100.00
CSV，5	整行获取	92.19	2.199	2908.9	10.388	96.88	100.00
CSV，5	无紧界	92.19	2.199	2909.1	10.434	96.88	100.00
JSON, 3	CAPE	88.78	1.412	1538.7	12.130	92.19	98.21
JSON, 3	急切精确	88.64	1.409	2324.0	17.879	82.03	98.44
JSON, 3	仅程序	88.64	1.409	2080.2	17.635	75.78	98.44
JSON，3	受保护的两步法	88.64	1.428	2317.9	17.896	75.78	92.19
JSON，3	通用惰性	86.93	1.479	2025.2	16.060	90.62	93.30
JSON，3	程序优先扩展	88.64	1.409	2170.2	17.749	82.81	98.44
JSON，3	整行获取	85.09	1.369	2360.2	17.842	85.16	89.62
JSON，3	无紧界	88.78	1.412	1547.2	12.175	92.19	98.21
JSON, 5	CAPE	88.78	1.411	1530.6	12.158	94.53	100.00
JSON，5	急切精确	88.64	1.409	2321.7	17.895	82.03	98.44
JSON，5	仅程序	88.64	1.409	2081.6	17.616	75.00	98.44
JSON，5	受保护的两步法	88.64	1.428	2318.0	17.893	76.56	92.19
JSON，5	通用惰性	86.93	1.567	1957.1	15.878	90.62	93.30
JSON，5	程序优先扩展	88.64	1.409	2179.2	17.749	83.59	98.44
JSON，5	整行获取	85.65	1.375	2053.1	16.948	85.16	91.18
JSON，5	无紧界	88.78	1.411	1534.9	12.131	95.31	100.00

表26. Qwen3-VL-4B-Instruct：完整面板测量。

接口， C	方法	成功率（%）	调用次数	查询次数	时间（秒）	认证率（%）	最优性（%）
CSV, 3	CAPE	83.40	2.062	2763.2	10.062	100.00	100.00
CSV，3	急切精确	83.40	2.062	3208.0	10.484	100.00	100.00
CSV，3	仅程序	75.68	1.904	624.0	3.583	40.62	40.62
CSV，3	受保护的两步法	82.23	2.320	3208.0	10.436	40.62	40.62
CSV，3	通用惰性	82.32	2.063	3106.0	15.878	96.09	96.09
CSV，3	程序优先扩展	83.40	2.062	3064.5	10.650	100.00	100.00
CSV，3	整行获取	83.40	2.062	3015.0	10.545	100.00	100.00
CSV，3	无紧界	83.40	2.062	3087.0	10.580	100.00	100.00
CSV, 5	CAPE	95.31	2.197	2925.5	10.426	100.00	100.00
CSV，5	急切精确	95.31	2.197	3208.0	10.661	100.00	100.00
CSV，5	仅程序	89.94	2.210	624.0	3.594	40.62	78.12
CSV，5	受保护的两步法	95.31	2.639	3208.0	10.502	40.62	73.44
CSV, 5	通用惰性	83.11	2.354	2228.0	12.745	78.12	78.12
CSV, 5	程序优先扩展	95.31	2.197	2956.5	10.663	100.00	100.00
CSV，5	整行获取	95.31	2.197	2925.3	10.398	100.00	100.00
CSV，5	无紧界	95.31	2.197	2925.5	10.432	100.00	100.00
JSON, 3	CAPE	92.33	1.407	1527.4	12.063	95.31	98.44
JSON，3	急切精确	92.19	1.405	2271.4	17.818	85.16	98.44
JSON，3	仅程序	92.19	1.405	2081.0	17.645	80.47	98.44
JSON, 3	受保护的两步法	92.19	1.416	2266.3	17.875	80.47	93.75
JSON, 3	通用惰性	90.48	1.473	2024.2	16.015	93.75	93.75
JSON，3	程序优先扩展	92.19	1.405	2168.5	17.766	85.94	98.44
JSON，3	整行获取	88.64	1.366	2355.0	17.840	88.28	89.84
JSON，3	无紧界	92.33	1.407	1535.3	12.101	95.31	98.44
JSON, 5	CAPE	92.33	1.406	1514.3	12.072	97.66	100.00
JSON，5	急切精确	92.19	1.405	2267.9	17.828	85.16	98.44
JSON，5	仅程序	92.19	1.405	2083.1	17.621	79.69	98.44
JSON，5	受保护的两步法	92.19	1.416	2265.1	17.829	81.25	93.75
JSON，5	通用惰性	90.48	1.555	1952.7	15.836	93.75	93.75
JSON，5	程序优先扩展	92.19	1.405	2175.8	17.761	86.72	98.44
JSON，5	整行获取	89.20	1.371	2046.7	16.926	88.28	91.41
JSON，5	无紧界	92.33	1.406	1518.4	12.058	98.44	100.00

G. 执行与扩展性研究

G.1. 执行条件与反馈控制器

执行研究使用64个保留请求、DeepSeek-V4-Pro提案、CSV/JSON接口和 $C = 3/5$，共计256种条件。无效提案对应的四种条件完成数为零；其余252种条件具有原生执行测量。每种条件下都会重新编译一次CAPE。其策略在252种可执行条件中的251种达到独立最优值，并始终保留目录顺序初始策略中成功的配置。测量使用Python 3.12.3、pandas 3.0.2、DuckDB 1.5.5、fsspec 2026.9.0和pytz 2026.6；计时数据与原始机制、匹配研究及提案研究分开保留。

目录重试会执行共享的初始策略。大语言模型序列化重试从程序表示零开始，若失败，则选择一个尚未尝试的程序表示。若只剩下一种表示，则直接执行。操作固定不变，每个程序消耗一次工具调用。控制器会收到请求、提议操作、模式、前五条记录、可用的程序编码以及自身的响应历史。CSV会报告结构化解析/查询错误；JSON会报告有长度限制的异常消息。各方法使用相同的发布保护机制和原生程序响应。日期为2026-04-23的DeepSeek-V4-Pro部署使用零温度、JSON输出和8,192个令牌的补全额度。已完成的决策会被保留，不进行语义重采样；相同的历史记录共用该响应。这项研究完成了1,442次模型请求，观测到的输入与补全令牌共计2,159,674个。

表 27. 带有 95%源聚类区间的执行研究配对对比。完成增益为 CAPE 减去对照组；节省的调用次数和节省时间为对照组减去 CAPE。成本包括测得的通用准备工作和方法特定编译组件。

接口/C	对照组	完成增益（百分点）	节省的调用次数	冷启动总耗时节省（秒）
CSV/3	目录重试	15.19 [7.76，24.61]	-0.084 [-0.157，0.007]	-9.047 [-11.626，-6.045]
CSV/3	大语言模型重试	15.38 [7.81，25.00]	-0.084 [-0.157，0.007]	-6.203 [-8.300，-3.899]
CSV/5	目录重试	15.43 [5.47，27.15]	0.257 [0.052，0.508]	-9.486 [-11.817，-6.435]
CSV/5	大语言模型重试	16.11 [6.64，26.66]	0.260 [0.059，0.502]	-6.901 [-8.559, -4.738]
JSON/3	目录重试	34.66 [27.56, 41.76]	0.481 [0.457, 0.521]	-11.229 [-15.785, -6.630]
JSON/3	大语言模型重试	34.66 [27.56, 41.76]	0.481 [0.457, 0.521]	-9.909 [-14.260, -5.581]
JSON/5	目录重试	26.85 [13.92，39.77]	1.256 [1.140，1.361]	-11.188 [-15.722，-6.631]
JSON/5	大语言模型重试	21.95 [12.86，30.89]	1.206 [1.118，1.289]	-7.930 [-12.987，-2.886]

一个初始反馈控制器还可以访问完整的程序和探针目录。其有界轨迹见表 28。在观察其探针使用情况后，我们指定了上面的程序优先序列化控制器，并在收集其对相同工作负载的响应之前冻结了其协议。两个控制器都保留了完整的原生记录和模型记录。

表 28。全目录反馈控制器诊断。探针计数和程序数量对完整的配置族轨迹进行计数；相同的模型上下文共享一个响应。丢失率是控制器丢失初始策略成功配置的比例，并按来源取平均值。

接口/C	成功率（%）	调用次数	丢失 (%)	探测调用	程序调用
CSV/3	15.53	2.984	81.08	9144	2888
CSV/5	15.58	4.753	84.08	17547	1603
JSON/3	54.30	2.680	20.15	10044	4828
JSON/5	46.24	4.057	39.48	17779	4701

G.2. 冷执行与复用核算

在执行前，通过固定哈希为每个条件选择一个端点配置。首先在决策缓存为空的情况下执行。我们在同一端点上对 CAPE 和目录重试重新计时，并与程序优先控制器进行对比，然后在每种方法的冷启动轮次后立即运行重复实验。所有方法都会缓存成功提交的程序表示。任务数据、操作、目录和端点配置在 $N = 1$、8、32 中保持不变。大语言模型控制器还会保留其精确历史决策缓存。调用次数和耗时均包括失败路径。

总成本结合了 c　　常见的测量目录准备、额外的方法特定编译和测量　　ed

执行：

$$T_{\mathrm{total}}(N) = T_{\mathrm{setup}} + T_{\mathrm{compile}}^{\mathrm{extra}} + \sum_{j=1}^{N} T_{\mathrm{execution},j}. \tag{8}$$

对 CAPE 而言，设置加上额外编译时间构成其测得的生命周期；重试方法只需支付一次共同设置成本。共享的初始提案会被复用。调整后的时间扣除了客户端身份验证、初始化、速率控制和并发槽位等待，同时保留模型服务、工具执行、控制器工作和传输重试。完整族质量值和编译组件取自经验证的策略测量；冷启动轮次和重复执行时间则是为本次比较新近测得的。

表 29. 初始覆盖率、冷执行和模型成本。Lost 为初始覆盖率损失（%）；Cold 为所有请求中所选端点完成率（%）。Exec. 为测得的执行时间；Total 包含准备和编译。Adjusted 为从 Total 中扣除客户端等待后的时间。时间和令牌均按有效提案计算。

接口/C	方法	丢失 (%)	冷执行（%）	执行时间（秒）	总计（秒）	调整后（秒）	令牌
CSV/3	目录重试	0.00	64.06	0.121	1.146	1.146	0.0
CSV/3	大语言模型重试	2.50	67.19	2.964	3.989	2.620	1720.2
CSV/3	CAPE	0.00	82.81	0.126	10.192	10.192	0.0
CSV/5	目录重试	0.00	82.81	0.106	1.191	1.191	0.0
CSV/5	大语言模型重试	1.95	85.94	2.691	3.776	2.655	1910.8
CSV/5	CAPE	0.00	98.44	0.135	10.677	10.677	0.0
JSON/3	目录重试	0.00	78.12	0.103	0.472	0.472	0.0
JSON/3	大语言模型重试	0.00	78.12	1.423	1.792	1.198	1039.4
JSON/3	CAPE	0.00	96.88	0.112	11.701	11.701	0.0
JSON/5	目录重试	0.00	68.75	0.099	0.522	0.522	0.0
JSON/5	大语言模型重试	0.20	73.44	3.357	3.780	2.513	2540.7
JSON/5	CAPE	0.00	90.62	0.119	11.710	11.710	0.0

表30。所有有效提案上的固定端点复用。成功率包括失败执行，调用次数是 N 次执行的总计，编译仅计费一次。调整后时间扣除了客户端等待时间。平均值平衡了数据来源。

接口/C	N	方法	成功率（%）	调用次数	总计（秒）	调整后（秒）
CSV/3	1	CAPE	84.15	2.268	10.192	10.192
CSV/3	1	次大语言模型重试	68.53	2.281	3.989	2.620
CSV/3	1	目录重试	65.40	2.281	1.146	1.146
CSV/3	8	CAPE	84.15	11.487	10.916	10.916
CSV/3	8	大语言模型重试	68.53	13.688	4.637	3.267
CSV/3	8	目录重试	65.40	14.125	1.804	1.804
CSV/3	32	CAPE	84.15	43.094	13.463	13.463
CSV/3	32	大语言模型重试	68.53	52.795	7.143	5.774
CSV/3	32	目录重试	65.40	54.732	4.028	4.028
CSV/5	1	CAPE	100.00	2.248	10.677	10.677
CSV/5	1	大语言模型重试	87.50	2.373	3.776	2.655
CSV/5	1	目录重试	84.38	2.388	1.191	1.191
CSV/5	8	CAPE	100.00	9.248	11.558	11.558
CSV/5	8	大语言模型重试	87.50	12.873	4.490	3.369
CSV/5	8	目录重试	84.38	13.763	1.946	1.946
CSV/5	32	CAPE	100.00	33.248	14.485	14.485
CSV/5	32	大语言模型重试	87.50	48.873	6.908	5.787
CSV/5	32	目录重试	84.38	52.763	4.776	4.776
JSON/3	1	CAPE	98.44	1.297	11.701	11.701
JSON/3	1	大语言模型重试	79.69	1.625	1.792	1.198
JSON/3	1	目录重试	79.69	1.625	0.472	0.472
JSON/3	8	CAPE	98.44	8.297	12.463	12.463
JSON/3	8	大语言模型重试	79.69	11.469	2.540	1.946
JSON/3	8	目录重试	79.69	11.469	1.136	1.136
JSON/3	32	CAPE	98.44	32.297	15.196	15.196
JSON/3	32	大语言模型重试	79.69	45.219	5.032	4.439
JSON/3	32	目录重试	79.69	45.219	3.497	3.497
JSON/5	1	CAPE	91.96	1.429	11.710	11.710
JSON/5	1	大语言模型重试	74.78	2.650	3.780	2.513
JSON/5	1	目录重试	70.09	2.696	0.522	0.522
JSON/5	8	CAPE	91.96	8.772	12.484	12.484
JSON/5	8	大语言模型重试	74.78	16.712	4.500	3.233
JSON/5	8	目录重试	70.09	18.071	1.123	1.123
JSON/5	32	CAPE	91.96	33.951	15.081	15.081
JSON/5	32	大语言模型重试	74.78	64.926	6.773	5.506
JSON/5	32	目录重试	70.09	70.786	3.198	3.198

共同成功子集包含 188 个条件，在这些条件下，三种方法都在冷启动轮次中提交了所请求的输出。此后每次缓存执行也都成功。表 31 报告了它们在完成情况相同条件下的成本，其中每个接口/预算组均包含全部八个来源。

表31. 共同成功子集上的复用成本。计数值为案例数，条件在三种方法及 N 的所有取值间保持一致。成本按来源平衡，以秒为单位，准备和编译成本各计一次。CAPE 和目录重试没有模型客户端等待时间。

接口/C	N	案例数	CAPE（秒）	目录（秒）	大语言模型（秒）	大语言模型调整后（秒）
CSV/3	1	41	8.396	1.225	3.066	2.369
CSV/3	8	41	9.273	2.190	3.958	3.262
CSV/3	32	41	12.316	5.386	7.581	6.884
CSV/5	1	53	9.405	1.233	2.840	2.150
CSV/5	8	53	10.304	2.116	3.620	2.930
CSV/5	32	53	13.318	5.430	6.273	5.582
JSON/3	1	50	12.047	0.486	0.926	0.770
JSON/3	8	50	12.831	1.282	1.813	1.657
JSON/3	32	50	15.608	4.153	4.846	4.690
JSON/5	1	44	11.558	0.569	1.870	1.384
JSON/5	8	44	12.350	1.366	2.737	2.252
JSON/5	32	44	15.073	4.114	5.475	4.990

G.3. 配置规模、范围与观测结构

我们为每个来源选择一个投影请求，并评估五种 CSV 条件： $C = 5$ 下的 64、128 和 256 个配置； $C = 8$ 下的 64 个配置；以及 $C = 5$ 下合并探测反馈的 64 个配置。CAPE、急切精确和匹配的选择性 AND/OR 在每个条件下均进行两次新进程重复，共得到 240 次测量。30 秒的软生命周期限额、原生线程限制和方法定义与原始编译器研究保持一致。运行时版本与上文的执行研究相匹配。配对区间对两项新研究均使用随机种子为 20261108 的 5,000 次源聚类自助法重采样。

扩展后的 CSV 系列采用四种和八种原生日期格式。它们保留原有的 64 种配置，以及原始操作/配置对的效果。合并反馈条件会将成功的探测预览映射为 `{parsed:true}`，同时保留错误响应、程序效果和发布检查。每种调用预算均使用相应的目录顺序初始策略和所需的成功集合。表格34 列出了有效配置数量和操作目录规模。

表 32。完整的结构与规模结果。成功率表示完成情况，调用次数按每个配置计，时间表示编译生命周期，查询次数表示已获取效应，认证率表示运行时证书率。每行汇总八个来源和两次重复。

设置	方法	成功率（%）	调用次数	时间（秒）	查询次数	精确率（%）
64, $C = 5$	急切精确	100.00	2.875	14.170	4432.0	100.00
64, $C = 5$	匹配的与/或	100.00	2.875	14.874	4354.0	100.00
64, $C = 5$	CAPE	100.00	2.875	15.136	4354.0	100.00
128, $C = 5$	急切精确	66.41	3.309	26.641	8644.3	12.50
128, $C = 5$	匹配的与/或	74.22	3.379	26.740	7731.1	12.50
128, $C = 5$	CAPE	88.28	3.309	26.658	7559.1	12.50
256, $C = 5$	急切精确	57.03	3.506	26.853	7745.2	12.50
256, $C = 5$	匹配的与/或	58.59	3.619	26.904	7198.9	12.50
256, $C = 5$	CAPE	62.60	3.479	26.903	7117.4	12.50
64, $C = 8$	急切精确	100.00	3.047	15.830	4432.0	87.50
64, $C = 8$	匹配的与/或	100.00	2.875	16.123	4354.0	87.50
64, $C = 8$	CAPE	100.00	2.875	16.694	4354.0	87.50
64，合并反馈	急切精确	99.22	2.883	14.146	4432.0	100.00
64，合并反馈	匹配的与/或	99.22	2.883	14.982	4354.0	100.00
64，合并反馈	CAPE	99.22	2.883	15.174	4354.0	100.00

表 33。规模化研究配对对比及 95%源聚类区间。完成增益为 CAPE 减去对照组；节省时间和节省的查询次数为对照组减去 CAPE。

设置	对照组	完成增益（百分点）	节省时间（秒）	节省的查询次数
64, $C = 5$	急切精确	0.00 [0.00、0.00]	-0.966 [-1.779, -0.135]	78.0
64, $C = 5$	匹配的与/或	0.00 [0.00、0.00]	-0.262 [-0.837，0.357]	0.0
128, $C = 5$	急切精确	21.88 [5.86, 39.45]	-0.017 [-0.036, 0.000]	1085.2
128, $C = 5$	匹配的与/或	14.06 [3.52、25.00]	0.082 [0.033，0.132]	172.0
256, $C = 5$	急切精确	5.57 [1.95, 10.25]	-0.050 [-0.117，-0.012]	627.8
256, $C = 5$	匹配的与/或	4.00 [1.07、6.93]	0.001 [-0.019, 0.017]	81.4
64， $C = 8$	急切精确	0.00 [0.00，0.00]	-0.864 [-1.990，0.106]	78.0
64， $C = 8$	匹配的与/或	0.00 [0.00，0.00]	-0.571 [-1.245，-0.009]	0.0
64，合并反馈	急切精确	0.00 [0.00，0.00]	-1.029 [-1.877，-0.161]	78.0
64，合并反馈	匹配的与/或	0.00 [0.00，0.00]	-0.192 [-0.630，0.340]	0.0

表 34。五种条件的结构。有效配置具有不同的完整动作效果特征。程序、探测和有效配置列给出跨来源的范围；结构列统计归一化动作结构。

设置	配置	有效	程序	探测	结构
64, $C = 5$	64	64–64	2–24	68–68	6
128, $C = 5$	128	128–128	2–48	100–100	6
256, $C = 5$	256	256–256	2–96	164–164	6
64, $C = 8$	64	64–64	2–24	68–68	6
64，合并反馈	64	32–64	2–24	68–68	6

G.4. 编译时限

扩展性研究将编译时限固定为 30 秒。我们在$C = 5$ 下重复测试配置数为 64、128 和 256 的条件，时间预算范围为 5 至 240 秒。请求、提案、三种编译器、初始策略、测量运行器和环境均与扩展性研究相同；唯一变化的设置是时间预算。每个系列和每种时间预算下，每种方法包含八个请求和两次新进程重复，总计 960 次测量。

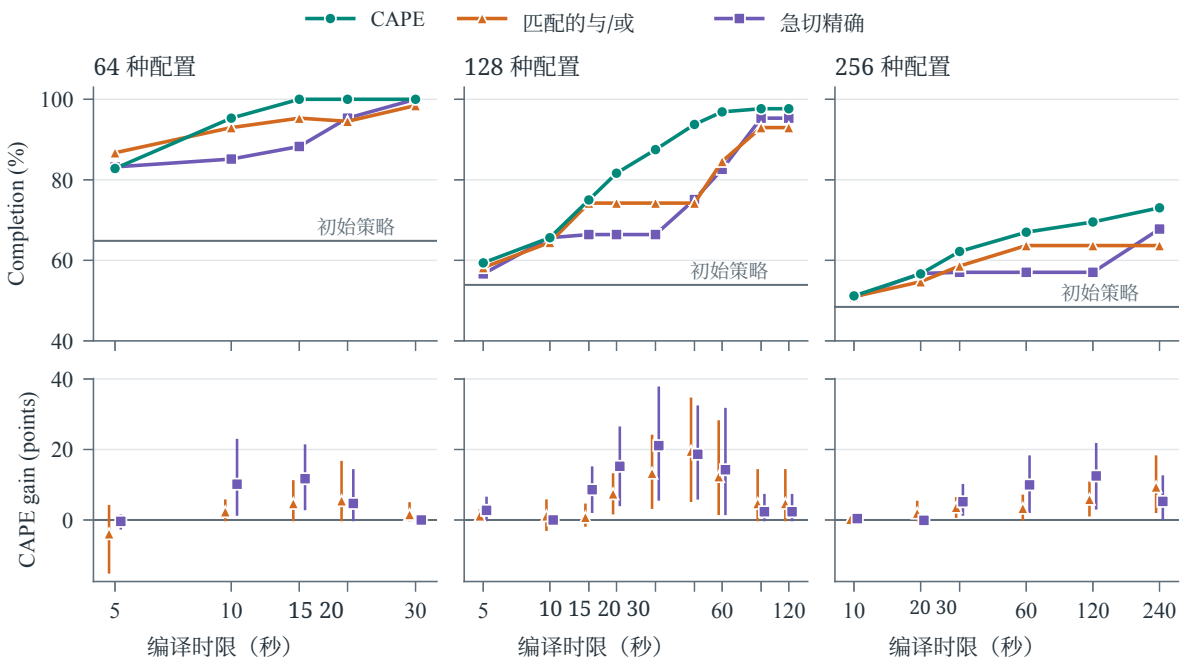

图 13。**当时间预算覆盖部分校准过程时，CAPE 的领先优势最大**。上：以源平衡均值表示的完成情况与编译时限的关系；水平线表示初始策略。下：CAPE 相对于各对照组的配对完成增益，以相应对照组的颜色绘制，并附有 95%源聚类区间。

在 128 个配置下，CAPE 在每个时间预算中完成的配置数都至少与两个对照组相同（图 13，表 35）。在 15 到 30 秒之间，急切精确仍在获取效果，其完成率保持在 66.41%；匹配的与/或在 15 到 45 秒之间保持在 74.22%，而 CAPE 则从 75.00% 上升到 93.75%。在 30 秒时，CAPE 相对于急切精确的领先优势达到峰值 21.09 个百分点；在 45 秒时，相对于匹配的与/或的领先优势达到峰值 19.53 个百分点。与急切精确相比，从 15 到 60 秒的配对区间均不包含零；与匹配的与/或相比，从 20 到 60 秒的配对区间均不包含零。在 90 秒和 120 秒时，领先优势分别为 2.34 个百分点和 4.69 个百分点，其区间均延伸至零。

其他两个系列在其他时间预算下也呈现相同趋势。在 64 种配置下，CAPE 在 10 秒和 15 秒时分别领先急切精确 10.16 和 11.72 个百分点；到 30 秒时，CAPE 和急切精确均完成了所有配置。在 256 种配置下，120 秒时 CAPE 对急切精确的领先幅度增至 12.50 个百分点，且从 30 秒到 240 秒的全部八个配对区间均不包含零。CAPE 在两个单元中低于对照组。在 64 种配置、5 秒时，匹配的与/或完成率为 86.72%，急切精确为 83.20%，CAPE 为 82.81%；在 256 种配置、20 秒时，急切精确完成率为 56.74%，CAPE 为 56.64%；这三项的区间均包含零。30 秒时各单元的数据与扩展性研究一致：128 种配置下分别为 87.50%、74.22% 和 66.41%，而扩展性研究中的数据分别为 88.28%、74.22% 和 66.41%。

表 35。编译时限 τ 下的完成情况。Initial 表示初始策略的完成情况。增益是 CAPE 相对于对照组的百分点差值，并附有 95%源聚类区间。粗体表示在没有任何对照组完成得更多时的 CAPE。

			完成率（%）			CAPE 增益 [95% 区间]	
$\|H\|$	τ(s)	初始	Eager AND/OR		CAPE	与急切执行相比	与/或对比
64	5	64.84	83.20	86.72	82.81	−0.39 [−2.34, 1.17]	−3.91 [−14.84, 3.91]
	10	64.84	85.16	92.97	**95.31**	10.16 [1.56, 22.66]	2.34 [0.00, 5.47]
	15	64.84	88.28	95.31	**100.00**	11.72 [3.12, 21.09]	4.69 [0.00, 10.94]
	20	64.84	95.31	94.53	**100.00**	4.69 [0.00, 14.06]	5.47 [0.00, 16.41]
	30	64.84	100.00	98.44	**100.00**	0.00 [0.00, 0.00]	1.56 [0.00, 4.69]
128	5	53.91	56.64	58.20	**59.38**	2.73 [0.00, 6.25]	1.17 [0.00, 2.73]
	10	53.91	65.62	64.45	**65.62**	0.00 [0.00, 0.00]	1.17 [−2.73, 5.47]
	15	53.91	66.41	74.22	**75.00**	8.59 [2.34, 14.84]	0.78 [−1.56, 4.30]
	20	53.91	66.41	74.22	**81.64**	15.23 [4.30, 26.17]	7.42 [1.95, 12.89]
	30	53.91	66.41	74.22	**87.50**	21.09 [5.86, 37.50]	13.28 [3.52, 23.83]
	45	53.91	75.10	74.22	**93.75**	18.65 [6.15, 32.13]	19.53 [5.47, 34.38]
	60	53.91	82.62	84.57	**96.88**	14.26 [1.76, 31.45]	12.30 [1.76, 27.93]
	90	53.91	95.31	92.97	**97.66**	2.34 [0.00, 7.03]	4.69 [0.00, 14.06]
	120	53.91	95.31	92.97	**97.66**	2.34 [0.00, 7.03]	4.69 [0.00, 14.06]
256	10	48.44	50.78	50.98	**51.17**	0.39 [0.00, 0.98]	0.20 [0.00, 0.49]
	20	48.44	56.74	54.69	56.64	−0.10 [−0.29, 0.00]	1.95 [0.00, 5.08]
	30	48.44	57.03	58.59	**62.21**	5.18 [1.56, 9.87]	3.61 [0.98, 6.15]
	60	48.44	57.03	63.67	**66.99**	9.96 [2.34, 17.97]	3.32 [0.20, 6.84]
	120	48.44	57.03	63.67	**69.53**	12.50 [3.32, 21.48]	5.86 [1.37, 10.55]
	240	48.44	67.77	63.67	**73.05**	5.27 [0.39, 12.30]	9.38 [2.34, 17.97]

表 36. 时限扫描中的执行调用次数、编译时间和证书。调用次数按配置统计，时间为编译生命周期所用秒数，认证率为运行时证书率（%）。

		急切精确			匹配的与/或			CAPE		
$\|H\|$	τ(秒)	调用次数	Time	认证率	调用次数	Time	认证率	调用次数	Time	认证率
64	5	2.855	4.62	12	3.008	4.64	12	2.895	4.63	12
	10	2.914	9.04	12	2.996	9.03	12	2.922	9.02	12
	15	2.988	13.15	38	2.969	12.96	38	2.891	13.02	38
	20	2.906	14.80	88	3.012	14.84	81	2.883	14.94	81
	30	2.875	14.70	100	2.887	15.98	94	2.875	16.10	100
128	5	3.477	4.77	12	3.406	4.80	12	3.504	4.73	12
	10	3.215	9.14	12	3.426	9.13	12	3.145	9.12	12
	15	3.309	13.97	12	3.379	13.51	12	3.270	13.55	12
	20	3.309	17.88	12	3.379	17.98	12	3.305	17.90	12
	30	3.309	26.66	12	3.379	26.71	12	3.309	26.69	12
	45	3.537	38.34	38	3.379	37.90	38	3.285	37.72	38
	60	3.508	47.62	50	3.621	46.40	50	3.277	46.32	50
	90	3.363	52.38	88	3.246	55.81	88	3.254	56.52	88
	120	3.363	56.41	88	3.246	62.42	88	3.254	61.68	88
256	10	3.463	9.37	12	3.645	9.45	12	3.467	9.38	12
	20	3.506	18.16	12	3.604	18.10	12	3.314	18.15	12
	30	3.506	26.90	12	3.619	26.90	12	3.409	26.90	12
	60	3.506	53.23	12	3.572	53.34	12	3.484	53.19	12
	120	3.506	105.86	12	3.572	105.02	38	3.469	105.72	19
	240	3.664	174.49	50	3.494	169.22	50	3.438	174.14	50

每个返回的策略都会通过全新的原生执行进行重放，并且都保留了初始策略完成的配置。对原始测量和重复测量进行统计后，全部 366 个运行时证书均等于扩展性研究中针对同一请求和同一系列所确定的独立最优值。

测量机器由多个任务共享，在第一次运行中，部分工作进程曾长时间处于挂起状态。当某项测量的非 CPU 时间（即生命周期减去单线程工作进程的 CPU 秒数）超过 1 秒加生命周期的 1% 时，我们便称该测量受到干扰。扩展性研究中的最大非 CPU 时间为 1.09 秒，因此按此规则，其测量均未受到干扰。该规则仅使用进程计时，并在首次运行后、任何重新测量之前以及重放结果尚未生成时确定。它标记了 960 项测量中的 105 项（CAPE 占 37 项，匹配的与/或占 25 项，急切精确占 43 项）。这些测量按照相同的规格分三轮重新测量，分别为 105 项、32 项和 10 项，之后均未受到干扰。上述结果使用每项任务首次未受干扰的测量值；表 37 重复列出了表 35 中的 960 项原始测量。

表 37。表35根据原始测量值计算，未重新测量受扰对象 s.

			完成（%）			CAPE 增益 [95% 区间]	
$\|H\|$	τ(秒)	初始值	Eager AND/OR		CAPE	相对于急切执行	相对于 与/或
64	5	64.84	83.20	84.38	82.81	−0.39 [−2.34, 1.17]	−1.56 [−7.81, 3.91]
	10	64.84	85.16	92.97	**95.31**	10.16 [1.56, 22.66]	2.34 [0.00, 5.47]
	15	64.84	82.81	95.31	**100.00**	17.19 [3.12, 32.81]	4.69 [0.00, 10.94]
	20	64.84	95.31	94.53	**100.00**	4.69 [0.00, 14.06]	5.47 [0.00, 16.41]
	30	64.84	100.00	98.44	96.88	−3.12 [−9.38, 0.00]	−1.56 [−9.38, 4.69]
128	5	53.91	55.86	57.42	**58.01**	2.15 [0.00, 6.05]	0.59 [−2.93, 4.69]
	10	53.91	63.28	64.45	**65.62**	2.34 [0.00, 7.03]	1.17 [−2.73, 5.47]
	15	53.91	66.41	70.70	**75.00**	8.59 [2.34, 14.84]	4.30 [−1.37, 11.91]
	20	53.91	65.43	74.22	**81.64**	16.21 [5.66, 26.76]	7.42 [1.95, 12.89]
	30	53.91	66.41	74.22	**87.50**	21.09 [5.86, 37.50]	13.28 [3.52, 23.83]
	45	53.91	75.10	74.22	**93.75**	18.65 [6.15, 32.13]	19.53 [5.47, 34.38]
	60	53.91	82.62	83.40	**96.88**	14.26 [1.76, 31.45]	13.48 [1.76, 31.45]
	90	53.91	95.31	92.09	**97.66**	2.34 [0.00, 7.03]	5.57 [0.00, 14.94]
	120	53.91	95.31	92.97	**97.66**	2.34 [0.00, 7.03]	4.69 [0.00, 14.06]
256	10	48.44	50.78	50.98	50.49	−0.29 [−1.46, 0.59]	−0.49 [−1.76, 0.29]
	20	48.44	54.39	51.37	**56.59**	2.20 [−0.44, 7.03]	5.22 [0.00, 14.60]
	30	48.44	56.35	58.59	**62.21**	5.86 [1.95, 10.64]	3.61 [0.98, 6.15]
	60	48.44	57.03	63.67	**66.99**	9.96 [2.34, 17.97]	3.32 [0.20, 6.84]
	120	48.44	57.03	63.67	**69.53**	12.50 [3.32, 21.48]	5.86 [1.37, 10.55]
	240	48.44	67.77	63.67	**73.05**	5.27 [0.39, 12.30]	9.38 [2.34, 17.97]

G.5. 独立验证

每个已执行的动作都会根据其原生效应模型进行检查，包括观测、提交状态、发布回执和输出内容。模型决策会根据存储的请求历史和原始响应进行检查；热执行则会根据缓存的成功程序进行检查。新的执行比较涵盖冷启动阶段、重复执行和全系列重试评估，共涉及 80,756 次原生调用。

完整表优化解决了 40 项扩展性研究目标中的 36 项。剩余的四项 256配置投影保留了可行的下界值和乐观上界值；其返回的策略不带有运行时证书。全部 150 个正向运行时证书均经过独立验证，且每个返回的策略都通过了原生重放和初始覆盖检查。