/TemplateVersion (2026.1)

推理感知的提示优化以对齐黑盒大型语言模型

Saaduddin Mahmud, Mason Nakamura, Kyle H. Wray, Shlomo Zilberstein
Manning College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract

提示优化方法在对齐黑箱大型语言模型（LLM）方面表现出显著的效果。与此同时，推理扩展策略如 Best-of-N 抽样和 Majority Voting 也被证明通过权衡计算来增强对齐和性能。然而，现有的提示优化方法与推理策略无关，即在部署期间优化提示时不考虑所采用的推理策略。这构成了一个显著的方法论空白，因为我们的实证和理论分析揭示了这两种范式之间的强烈相互依赖性。此外，我们发现用户对多目标和推理预算之间权衡的偏好对提示和推理配置的选择有重要影响。为了解决这一空白，我们引入了一个统一的新框架，称为 Iapo（推理感知提示优化），该框架在了解推理预算和不同任务目标的情况下，同时优化提示和推理规模。然后，我们开发了一个针对 Iapo 的固定预算训练算法，我们称之为 Psst （通过顺序修剪进行提示扩展），并分析有限预算下的错误概率保证。最后，我们在六个不同的任务上评估了 Psst 的有效性，包括多目标文本生成和推理，并展示了在通过提示优化对齐黑箱 LLM 时，融入推理感知的关键作用。

在最近几年，大多数最先进的大型语言模型 (LLMs) 仅能通过黑箱 API 访问。因此，传统的需要访问模型权重或逻辑输出的对齐方法变得不可行。为了解决这一问题，基于提示优化的对齐方法开始受到关注。这些方法通常通过重新措辞或附加额外指令来增强输入提示，以更好地使模型的输出与任务目标进行对齐。对于黑箱模型，另一种广泛适用的对齐策略是使用诸如 Best-of-N 采样或 Majority Voting 等策略来扩大推理计算。这些推理扩展方法会为同一查询生成多个候选响应，并通过排列或投票机制选择最终响应。

尽管现有的提示优化技术已经取得了显著成功，但它们通常对模型输出的聚合或采样方式漠不关心，忽略了此类推理方法的影响。我们的初步实证研究表明，优化提示的性能对推理缩放方法的选择非常敏感。此外，我们的理论分析表明，将提示优化与推理解耦可能导致不对齐。最后，我们观察到，最佳对齐需要仔细考虑用户在多重目标之间权衡的特定偏好，以及他们愿意投入的计算资源。这些发现揭示了当前方法中的一个关键缺口：缺乏一个同时考虑提示优化、推理缩放策略、用户偏好和计算资源限制的统一框架。

为弥合这一差距，我们引入了 Iapo（Inference-Aware Prompt Optimization），一种专门设计的新颖提示优化框架，旨在从推理扩展的黑箱 LLMs 中产生对齐的响应。Iapo 同时优化提示设计和推理扩展策略，同时考虑不同的任务目标和计算预算。我们将识别 Iapo 框架的最优策略的任务表述为一个上下文最佳臂识别（BAI）问题。为有效解决此问题，我们提出了一个名为 Psst (Prompt Scaling via Sequential Trimming) 的固定预算训练算法。此外，我们引入了一个热身启发式方法，以在训练预算内进一步提升性能。

我们首先通过推导 Psst 的有限预算理论保证，来开始我们的分析。接下来，我们通过实验证明 Psst 在学习 Iapo 策略方面的有效性，这包括六个不同的任务，例如多目标文本生成、数学推理和常识推理基准。此外，我们的分析显示，在提示优化过程中忽略推理扩展可能导致显著的未对齐，这突出了推理意识在对齐黑箱 LLM 中的关键作用。

多年来，人们对使大语言模型 (LLMs) 与下游任务中的人类期望相一致付出了相当多的努力。许多被广泛采用的对齐方法——例如监督微调（SFT）、基于人类反馈的强化学习（RLHF）和具有可验证奖励的强化学习（RLVR）——都需要访问模型权重。这一限制激发了对黑盒对齐方法的越来越多的兴趣，如提示优化，它可以仅通过输入操作来对齐黑盒模型。提示优化在单目标和多目标环境中都表现出强大的性能。然而，这些方法与部署期间使用的推理策略无关，可能导致次优性能。相反，我们的方法明确解决了推理时策略与提示优化的相互依赖性问题。

最近，Shi et al. 将提示优化框定为一个固定预算的最佳臂识别（BAI）问题。虽然在有限评估预算下效果显著，但该方法仍然与推理无关，仅在单目标设置中进行了探索。我们的工作在两个关键方面在此基础上进行了扩展：（1）我们引入了一个情境化的公式，建模用户对多个目标及其相关计算成本的偏好；（2）我们纳入了推理意识，以确保与实际推理策略的一致性。为了学习一个最优策略，我们引入了一种固定预算的情境化 BAI 算法，Psst ，其灵感来自顺序对半（SH）(Karnin, Koren, and Somekh 2013) 。虽然 SH 最初是为纯带状环境开发的，但 Iapo 框架具有跨情境的完全信息反馈和单一情境的半带状反馈。Psst 利用了这些结构特性，实现了更高效的优化，扩展了标准 SH 无法容纳的范围。

另一条相关的研究工作集中在推理时对齐，其中着眼于在不修改模型参数的情况下改善推理期间的模型输出。这些方法中的一些，如 GenARM 和 DEAL (Xu et al. 2024; Huang et al. 2024) ，需要访问模型的 logits，从而限制了它们在黑箱设置中的适用性。相比之下，Best-of-N 采样 (BoN) 和 Majority Voting (MV) 方法则完全基于模型输出运作，通过生成多个候选项并选择最佳的一个，当 (OpenAI 2024; Yue et al. 2025; Wang et al. 2022; Krishna et al. 2022) 时，显示出了强大的实证提高。然而，这些方法引入了非平凡的计算成本，据我们所知，尚无方法明确优化计算预算与输出质量之间

的权衡。此外，我们的初步实验表明，这种推理扩展策略与提示设计存在不平凡的互动：针对单次解码优化的提示在 BoN 或 MV 下可能表现次优，反之亦然。这就需要一个对推理敏感的提示优化框架。

最后，一些白盒方法最近将推理意识融入到训练过程中。例如，Chow et al. (2025) 提出了一个推理感知的微调程序，该程序在 BoN 下明确优化探索–开发折衷。同样，BOND (Sessa et al. 2024) 和 BonBon (Gui, Gârbacea, and Veitch 2024) 旨在通过监督微调将 BoN 策略为单次生成过程。虽然这些方法避免了推理时采样的成本，但它们需要完全访问模型参数，并且不能超越 BoN 风格策略。此外，我们的方法是为黑盒 LLM 设计的推理感知微调的补充。在这一节中，我们首先形式化问题设置并介绍 Iapo 框架。接下来，我们提出一个实证例子，强调推理感知优化的必要性。在这些观察的基础上，我们建立了 Iapo 必要的理论条件，与分离优化相比。

问题表述

令 $\mathcal{X}$ 为用户查询的分布，$\mathcal{P}$ 为有限提示集合。一个对 $(x \in \mathcal{X}, p \in \mathcal{P})$ 被提交给冻结的黑箱 LLM，在固定的解码超参数下，生成 $N \in \{1, \dots, N_{\max}\}$ i.i.d. 完成 $\boldsymbol{y}_{1:N} = (y_1, \dots, y_N)$。$K$ 有界（可能是向量）目标为每个完成评分 $O_k : \mathcal{X} \times \mathcal{Y} \to [o_k^{\min}, o_k^{\max}]$（例如 有效性、无害性、精确匹配）。我们也定义生成响应的成本为 $\mathrm{Cost}(x, y_i)$，一个复合函数，考虑到诸如标记数、时间和能量等各种计算因素。我们将其添加为 $(K+1)$ -st 目标 $O_{k+1} = -\mathrm{Cost}(x, y_i)$。外部实体提供一个上下文 $c = (w_1, \dots, w_{K+1}) \in \mathcal{C}$，其中每个 w_k 从有限离散域中选择。鉴于上述设置，我们将推理策略形式化如下。

Best-of-N （BoN）。 BoN 返回最大加权效用：

$$R_x^{\mathrm{BoN}}(c, p, N) = \underbrace{\max_{i \le N} \sum_{k=1}^{K} w_k \, o_k(x, y_i)}_{\text{task reward}} + \underbrace{w_{k+1} \sum_{i=1}^{N} o_{k+1}(x, y_i)}_{\text{inference cost}}. \quad (1)$$

对于查询 x，这对 (p, N) 产生 i.i.d. 完成 $y_{1:N}$ 和提取的答案 $\ell_i = h(x, y_i)$。对于每个不同答案 s，定义投票计数 $n_s = \sum_{i=1}^{N} \mathbf{1}[\ell_i = s]$，最大值 $n^\star = \max_s n_s$，以及并列数 $t = \sum_s \mathbf{1}[n_s = n^\star]$。MV 在 t 最大值中随机预测。拥有金答案 $a(x)$ 以及定义成功信用为 $o_1(x, p, N) = \frac{\mathbf{1}[n_{a(x)} = n^\star]}{t}$，我们定义 MV 效用为：

$$R_x^{\mathrm{MV}}(c, p, N) = \underbrace{w_1 \, o_1(x, p, N)}_{\text{task reward}} + \underbrace{w_2 \sum_{i=1}^{N} o_2(x, y_i, c)}_{\text{inference cost}}. \quad (2)$$

。

备注。 混合策略出现在不同目标需要不同聚合规则的情况下，例如，在需要推理任务的二元正确性时应用 MV，而在风格质量上使用 BoN。基于上述内容来定义它是很简单的。

Iapo 框架

令一个推理配置为一个元组 $\theta \in \Theta$（例如，温度，top-p，最大标记）。然后我们在 Iapo 中定义一组选项 $\mathcal{A}$ 如下：$a = (p, \theta, N) \in \mathcal{A} := \mathcal{P} \times \Theta \times \{1, \dots, N_{\max}\}$.

因此，每个臂都会固定提示、解码超参数以及采样完成的数量。然而，在整个文本中，我们将推断配置折叠到提示 p 中，并写作 $a = (p, N)$。最后，一个 Iapo 策略被定义为一个映射 $\pi : \mathcal{C} \to \mathcal{A}$，它在观察到上下文 c 后选择一个臂。

给定数据集 $\mathcal{X}$、上下文 $c \in \mathcal{C}$ 和聚合器 $\alpha \in \{\mathrm{BoN}, \mathrm{MV}\}$，则臂 a 的期望效用，即上下文-动作值函数或 Q -函数被定义为：

$$Q^\alpha(c, a) := \mathbb{E}_{x \sim \mathcal{X}}\big[R_x^\alpha(c, a)\big]. \quad (3)$$

。注意，$R_x^\alpha(c, a)$ 是一个随机变量。现在，设上下文最优臂为 $a^\star(c) = \arg\max_a Q^\alpha(a, c)$；因此，最优 Iapo 策略被定义为：$\pi^\star(c) = a^\star(c), \forall\, c \in \mathcal{C}$。

在本文中，我们采用训练后部署的设置来学习最优的 IAPO 策略。在给定的总完成预算为 T 的情况下，学习者可以自适应地选择臂 $a_t = (p_t, N_t) \in \mathcal{A}$ 和查询 $x_t \sim \mathcal{X}$，然后观察所有完成的完整原始奖励向量 $\boldsymbol{m}_t \in \mathbb{R}^{K+1}$。这个过程可以继续，直到预算耗尽 $(\sum_t N_t = T)$。在花费完整个预算之后，学习者返回一个部署策略 π_T。该策略的性能通过平均情景回报来评估：

$$\mathrm{ACR}(\pi_T) = \mathbb{E}_{c \sim \mathcal{C}}\big[Q^\alpha(c, a)\big], \quad (4)$$

学习算法的目标是在固定的拉取预算 T 条件下返回一个部署策略 π_T，以最大化 ACR。

激励性案例研究

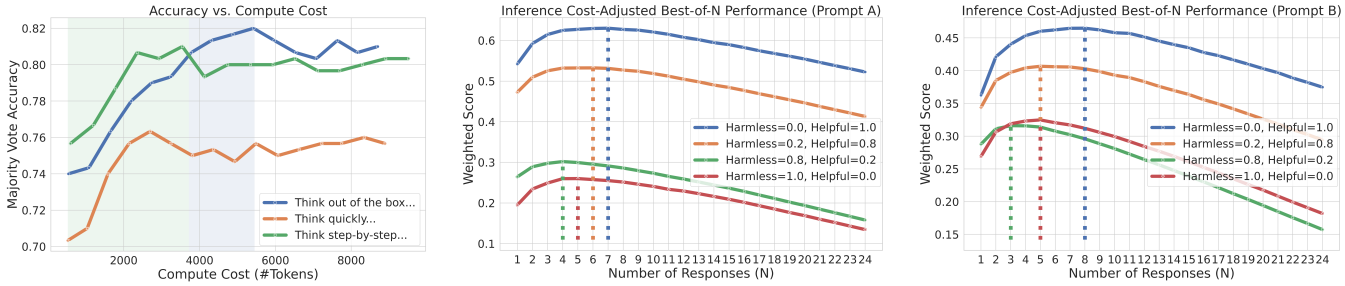
为了说明与推理无关的提示优化的局限性——并激发上面形式化的联合处理——我们进行了两个诊断实验，将 Llama-3.3-70B-Instruct (Grattafiori et al. 2024) 严格视为一个黑盒 API。结果在图 1 中总结。

我们在 MATH 上评估三种人工设计的提示。我们在 Majority Voting 下针对 $N \in \{1, \dots, 16\}$ 在 MATH 基准 (Hendrycks et al. 2021) 上进行评估。准确率与总解码成本的关系被绘制出来，并在 300 个查询上取平均（详细信息见附录）。有两个关键观察结果。首先，提示偏好的转变与计算预算有关：在低预算情况下，绿色提示表现最佳，但最终随着 Majority Voting 的有效性提高，会被蓝色提示超越。其次，与推理无关的优化可能是目光短浅的：仅基于单次（N=1）准确率选择提示会偏向绿色提示，而忽视了对于任何愿意分配更多计算资源的用户而言，蓝色提示始终是更好的选择。

为了理解绿色和蓝色趋势如何出现，考虑以下例子。在一个具有 MV 的推理任务中，假设对于提示 1，我们在查询 1 中有 40%，在查询 2 中有 90%，而对于提示 2，在两个查询中有 62%。单次平均值支持 A（0.65 对 0.62），但在具有 $N = 10$ 的 MV 下，A 下降到 ≈ 0.63，而 B 则提高到 ≈ 0.77。

(b,c) Helpful–Harmless 的最佳结果。在 Helpful–Harmless 的基准测试中，我们使用 Best-of- N 解码法对 $N \le 24$ 评估了两个提示（A 和 B）。每条曲线对应不同的用户定义的有益性和无害性之间的权衡，绘制出经过成本调整后的奖励值，取 1000 次查询的平均值（详见附录）。提示（A 与 B）和采样预算（N）的最佳选择高度依赖于这些偏好。例如，当有益性被更重视时，提示 A 是明确上佳的选择。

在确定了进行推理感知优化的必要性之后，我们现在研究在何种情况下联合优化变得至关重要。我们从建立推理无关（IA）效用开始：



(a) 多数票　(b) 有帮助-无害，Best-of-N（Prompt-A）　(c) 有用-无害，Best-of-N （提示-B）

Figure 1: 提示–推理相互依赖性。(a) 在 LLaMA-3.3-70B-Instruct 下的 Majority Voting 准确率，显示提示在预算（阴影部分）下的主导权变化。(b, c) 在 Best-of-N 解码下的成本调整奖励。提示和推理规模随用户定义的权衡而变化。

Proposition 1 (Inference-Agnostic Utility). 推理无关的提示优化方法优化无成本考虑的算术平均效用。

$$R_x^{\mathrm{IA}}(c, a=(p,N)) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} w_k o_k(x, y_i). \quad (5)$$

现在我们展示在什么条件下 IA 策略保持最优，或者可以从 IA Q -函数中轻松恢复最优策略。

Proposition 2 (Inference-Agnostic Optimality). 在 $R_x^{\mathrm{IA}}(c,a)$ 的线性变换下，推理无关的提示优化策略仍然是最优的，也就是说，$kR_x^{\mathrm{IA}}(c,a), k>0$ 和最优策略可以通过 Q 函数在仿射变换下简单恢复：

$$Q^{AF}(c,a) := \mathbb{E}_{x\sim\mathcal{X}}\left[aR_x^{IA}(c,a)+b\right] = kQ^{IA}(c,a)+b.$$

上述内容还强调了仿射聚合显著简化了推理感知优化。例如，在一个回归任务中，聚合预测是多个数值预测的均值，并且奖励由均方误差（MSE）定义，在某些情况下可以成为 IA 的仿射变换，从而消除了在训练过程中模拟推理缩放的需要。然而，BoN 和 MV 等常见的推理缩放策略通常不允许这样的仿射形式。虽然有时它们可以表达为 IA 的非仿射变换（例如在有大量 N 的伯努利情况下，其中 $R_x^{\mathrm{IA}}(c,a)\approx p$ （图 2)），但这些是特例。因此，尝试基于 Q^{IA} 确定 BoN 或 MV 的提示将导致不匹配。这激发了下一部分的内容，在那里我们开发了一种训练方法，处理超越仿射范围的一般 Iapo 设置。

在本节中，我们提出了一种基于固定预算臂淘汰的策略，用于训练策略 π_T ，称为 Psst （通过顺序修剪进行提示缩放）。然后，我们提供了一个理论分析，在有限推理预算下为 Psst 建立了错误保证。最后，我们介绍了一种实用的近似启发式方法，该方法在许多实际环境中提高了计算效率，而不会显著影响性能。

我们关注固定推理预算设置的原因是训练成本常常是实际应用中的主要瓶颈。此外，Psst 被设计为在批量探索模式中操作，这进一步降低了成本，因为许多黑盒 API 提供批量推理的显著折扣，相较于单次调用更加优惠。重要的是，Psst 也是无超参数的，不需要额外的调整。

经典的手臂淘汰方法如顺序淘汰（Sequential Elimination）(Even-Dar, Mannor, and Mansour 2006) 和顺序减半 (Sequential Halving) (Karnin, Koren, and Somekh 2013) 遵循一个简单的步骤：（i）将淘汰过程分成多轮；（ii）在每轮中，将这一轮的预算分配给剩余的手臂；（iii）根据手臂的估计值在这一轮结束时修剪掉一部分手臂。然而，Iapo 在以下关键方面偏离了纯粹的 BAI 设置：

- 不对称拉取成本。当在训练期间拉取臂 (p,N) 时，它使用 N 训练预算。
- 跨上下文重用。对查询 x 执行一次 (p,N) 操作可得到补全集 $\boldsymbol{y}_{1:N}$ 和目标向量集 $\boldsymbol{o}_{1:N}$ ，这可用于估算所有 $c\in\mathcal{C}$ 的 $R_x^{\alpha}(c,p,N)$ 。
- 在推理尺度上嵌套样本重复使用。提取大尺度推导出小尺度：通过将 N_i 抽样划分为 N_j 大小的不相交组，(p,N_i) 的提取生成了针对臂 (p,N_j) 的 $\lfloor N_i/N_j \rfloor$ i.i.d. 区块样本（例如，在每个区块上重新计算 BoN/MV)。

一个关键的结果是，对于一个提示，存活下来的最大规模决定了预算。设 $N_{\max}^{(r)}(p) = \max\{N : (p,N) \text{ survives at the start of round } r\}$. 。如果我们在第 r 轮为 $(p, N_{\max}^{(r)}(p))$ 分配了 K 次拉取（块），那么每个存活的臂 (p,N) 且带有 $N \le N_{\max}^{(r)}(p)$ 会自动通过块重用至少获得 K 个有效样本。因此，一个有效的臂消除策略应该利用（i）提示内的跨尺度重用以及（ii）打分时的跨上下文重用，同时要意识到不对称成本。

算法

轮结构　进行 $R=\lceil \log_2 |\mathcal{A}| \rceil$ 轮次，通过使用标志 $\boldsymbol{F}$ 跟踪每个上下文中的活跃臂。每轮分配相等的拉动预算 $n_r = \lfloor T/R \rfloor$ 。分配程序 Allocate (F , n_r) 在当前的一组独特活跃臂之间分配该预算，汇集所有上下文。基于该分配，一批推断调用被发出至目标 LLM。结果完成使用奖励函数或验证器进行评分，并存储在数据集 $\mathcal{D}$ 中。然后从收集的数据中估计 Q 值。在每个上下文内，对臂进行排名并淘汰表现最差的一半。所有轮次完成后，算法为每个上下文返回唯一的最终臂。

结构感知分配策略　分配策略是以跨上下文和跨规模的信息共享为目的而设计的。具体来说，设 $\mathcal{A}^{(r)}$ 表示轮次 r 中在所有上下文中聚合的唯一活跃臂集合。针对每个提示 p ，定义

$$N_{p,\max}^{(r)} = \max\{N \mid (p,N)\in\mathcal{A}^{(r)}\}$$

为在活跃臂中提示 p 的最大推理规模。然后，Psst 根据以下方案为每个臂分配预算：

$$\lambda(a) = \begin{cases} \left(\lfloor \frac{n_r N_{p,max}^{(r)}}{M} \rfloor\right. & \text{if } a=(p,N_p^{\max})\in\mathcal{A}^{(r)}, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

其中 $M=\sum_{p:(p,N_p^{\max})\in\mathcal{A}^{(r)}} N_{p,\max}^{(r)}$ 是一次采样所有此类最大臂的总成本。此策略在保持对提示的均匀覆盖的

Algorithm 1: 通过顺序修剪进行提示缩放

Require: Context set $\mathcal{C}$, prompt set $\mathcal{P}$, scale set $\mathcal{N}$, Scaling strategy α , Query Dataset $\mathcal{X}$, total pull budget T ;

for all $(c, a) \in \mathcal{C} \times \mathcal{A}$ do
  $\mathrm{F}_{c,a} \leftarrow true$
end for
$R \leftarrow \lceil \log_2(|\mathcal{A}|) \rceil$
for $r = 1$ to R do
  $n_r \leftarrow \lfloor T/R \rfloor$
  $\lambda^{(r)} \leftarrow \mathrm{Allocate}(\mathrm{F}, n_r)$
  $\mathcal{B} \leftarrow \{\}$
  for $(a, n_r) \in \lambda^{(r)}$ do
    for $i = 1...n_r$ do
      Sample $x \sim \mathcal{X}$
      $\mathcal{B} \leftarrow \mathcal{B} \cup (a, x)$
    end for
  end for
  $\mathcal{D} \leftarrow \text{Batch-Query}(\mathcal{B})$
  $Q^{\alpha}_{(r)} \leftarrow \text{Estimate-Q}(\mathcal{D})$
  for all $c \in \mathcal{C}$ do
    $\mathcal{A}_c^{(r)} \leftarrow \{a : \mathrm{F}_{c,a} = true\}$
    Rank $\mathcal{A}_c^{(r)}$ by $Q^{\alpha}_{(r)}(c, a)$
    Remove bottom $\lceil |\mathcal{A}_c^{(r)}|/2 \rceil$ arms // i.e. update $\boldsymbol{F}$
  end for
end for
return $\{a_c^{\star}\}_{c \in \mathcal{C}}$ // one survivor per context

同时，尊重成本的不对称性，并确保每个提示的最大规模具有相等数量的样本。

现在我们在上述分配策略下推导出 Psst 的误差界限[1] 。

注意：如果在应用 Sequential-Halving 时不利用 Iapo 的结构—具体来说，在不同尺度或背景之间不进行任何形式的信息共享—会导致样本复杂度增加 $O(|\mathcal{C}|N_{\max})$ 。

虽然我们描述算法时是在每一轮 $\mathcal{D}$ 中使用一组新的数据，但已证明在类似的二分风格算法中 (Fabiano and Cazenave 2021) ，积累所有过去观察数据——即所谓的储存——可以通过减少外层 R 因子来改进 T 的复杂性，并建议与 Psst 一起使用。

为了进一步减少 Psst 的预算需求，我们引入了 Top-$\boldsymbol{K}$ 筛选法，这是一种实用的启发式方法，通过在单位规模上执行短暂的统一提示筛选来修剪明显次优的提示，然后再运行完整的 PSST 。Top- $\boldsymbol{K}$ 筛选从 Psst 中提取预算的一部分 $T_0 = \lfloor \rho T \rfloor$ （$\rho \in (0, 1)$ ）。在 N=1 的规模限制下，预算在各个提示间均匀分配：每个 $p \in \mathcal{P}$ 接收 $\lfloor T_0/|\mathcal{P}| \rfloor$ 个独立同分布的 样本。根据这些数据，估计 $Q^{\alpha}(c, p, 1)$ 为 $\forall c \in \mathcal{C}, p \in \mathcal{P}$ 。对于每个上下文 c ，我们保留最好的 K 个提示 $\mathcal{P}_c^{(0)} = \text{Top-}K\{\widehat{Q}^{\alpha}(c, p, 1) : p \in \mathcal{P}\}$ 并舍弃其余的。随后，接下来的 PSST 运行将限制在每个 c 的精简臂集 $\mathcal{A}_c^{(1)} = \{(p, N) : p \in \mathcal{P}_c^{(0)},\ N \in \mathcal{N}\}$ ，并使用剩余的预算 $T' = T - T_0$ 。在下一部分中，我们将展示筛选策略在低训练预算环境下如何显著提高性能，而不会在实际任务中影响质量。然而，与完整的 Psst 相比的理论保障无法建立；可以在 Iapo 框架内精心构建反例任务，其中对于任何 $K < |\mathcal{P}|$ ，Top-K 筛选将表现不佳。

[1]证明见附录

实证评估

在本节中，我们通过实验评估 Psst 的有效性，并强调推理感知提示优化 (Iapo) 的重要性。我们的评估有两个主要目标：

- 为了证明 Psst 和 Top-K 筛选启发式在学习策略 π_T 方面非常有效。
- 要证明 Iapo 比推理策略无关的优化提高了平均成本调整后的奖励 (ACR)。

我们比较了 Psst 和 Top-K 筛选与几个基线。我们将 Top-K 筛选表示为 $K = 1, K = 4$ ，而 and $K = 8$ 分别表示为 Psst $+K1$ 、Psst $+K4$ 和 Psst $+K8$ 。对于这些启发式方法，我们固定了 $\rho = 0.2$ ，因为发现它在所有数据集上表现最好。Full Psst 是无参数的，不需要调整。在我们的第一组实验中，我们将我们提出的方法与几种标准的探索策略进行比较：

- Uniform：在一个批次中均匀探索所有臂，并在最后选择最佳的臂。
- $\boldsymbol{\epsilon}$ -贪婪算法：在每一步采样一个随机上下文，并以概率 $1 - \epsilon$ 选择最优臂。我们设定 $\epsilon = 0.15$ ，这在各数据集中表现最佳。
- Softmax：根据估计的 Q 值上的 Softmax 分布抽取样本臂。
- UCB：在每次操作中，选择具有最高乐观 Q 估计的臂。调整后的探索常数 0.1 。

注意，所有基线方法都会在不同的环境和推理尺度之间共享信息；然而，没有一个方法是专门设计用来利用 Iapo 结构的，即它们对结构是无感的。

在第二组实验中，我们考虑了众所周知的 TRIPLE-SH (Shi et al. 2024) 方法的上下文变体，该方法将提示选择优化为一个纯粹的最佳臂识别 (BAI) 问题。然而，它并不优化推理规模。因此，我们包括了两个变体：

- 三倍 (N = 1)：仅执行使用单样本推理的提示优化。
- TRIPLE (N = 随机)：在为每个查询随机分配 N 的同时优化提示。

这些基线帮助隔离联合优化提示和推理规模的益处。此外，Psst $+K1$ 在这个实验中特别有趣，因为它近似于一个两阶段的分离优化：它首先使用成本感知目标选择一个与上下文相关的单发提示，然后调整推理规模。Psst $+K4$ 和 Psst $+K8$ 启发式方法表示在分离和完全联合优化之间的中间策略。

请注意，所有超参数扫描结果都在补充材料中；我们报告的是在所有六个数据集中找到的最佳设置的结果。

环境。 我们在总共六个环境中评估了推理感知优化。关键细节在表 1 中提供。环境一和四是合成构建的，用于模拟 Iapo 任务，其中提示-查询对的评分分布 $o_i(c, P, 1)$ 使用分类分布进行建模。我们引入它们以验证一些理论发现。其余四个环境基于广泛使用的真实世界数据集。在这些环境中，MATH (Hendrycks et al. 2021) 和

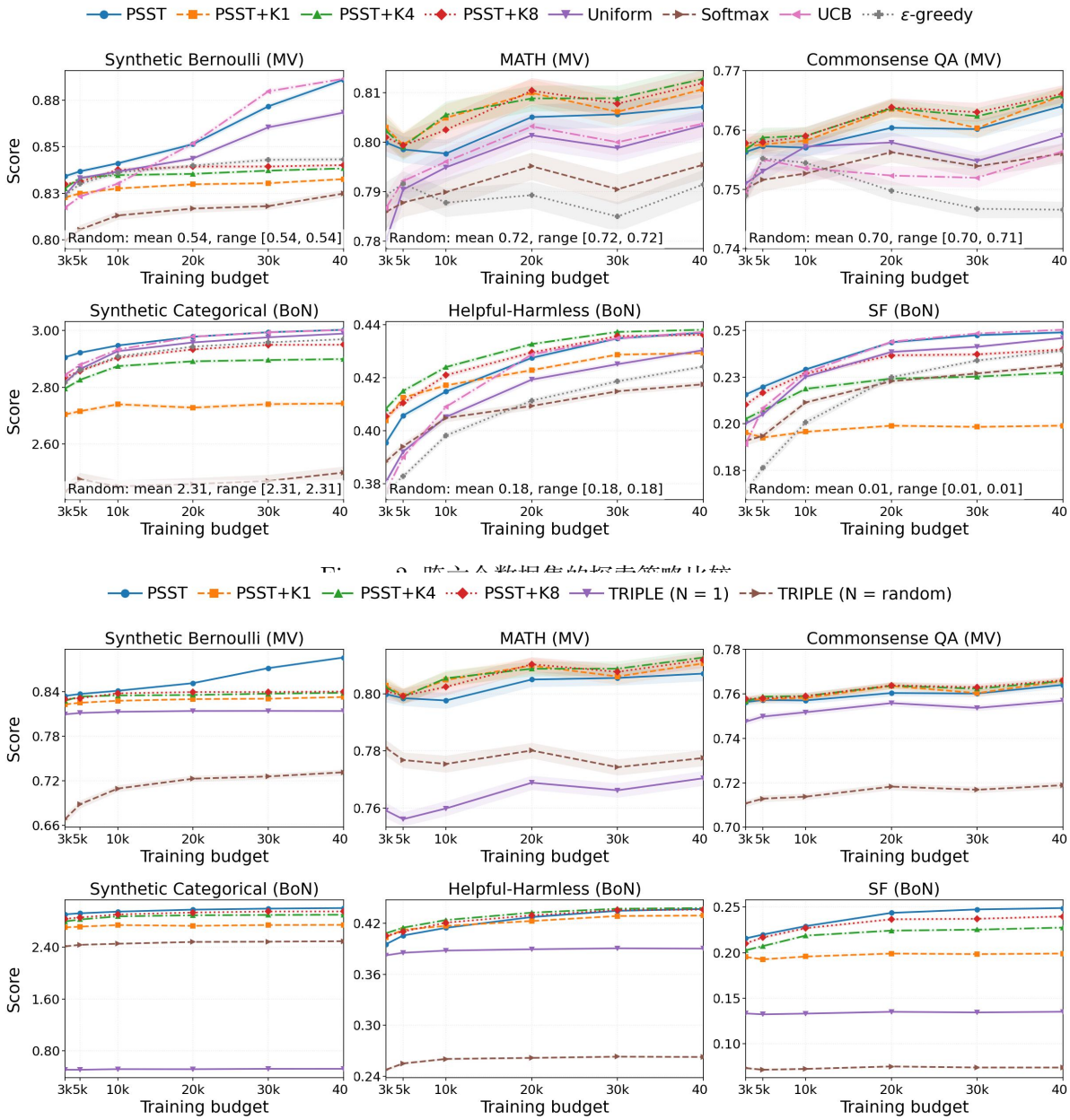


Figure 3: 推断感知优化在六个数据集上的有效性。

Environments	α	$\lvert\mathcal{P}\rvert$	$N_{\max}$	o_k^{max}	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{C}\rvert$
Synth–Bernoulli	MV	32	32	1.0	520	3
MATH	MV	25	32	1.0	316	3
CommonsenseQA	MV	48	32	1.0	1500	3
Synth–Categorical	BoN	32	32	4.0	512	27
Helpful–Harmless	BoN	20	32	1.0	1355	27
Summarization	BoN	20	32	1.0	1201	27

Table 1: 环境总结。

CommonsenseQA (Talmor et al. 2018) 用于在 Majority Voting (MV) 下评估推理任务，而 Helpful-Harmless (Bai et al. 2022) 和 Summarization (Stiennon et al. 2020) 则选择用于 Best-of-N (BoN) 评估。

对于 MV 任务，任务目标被定义为与正确答案完全匹配。所有三个 BoN 任务都是双目标的，我们使用以前多目标 LLM 对齐研究中公开可用的奖励模型来评分完成情况（见附录中的链接）。所有六个任务中的成本目标被定义为与每个响应的平均令牌数成比例。对于上下文规范，MV 任务包括一个预算机制 $ 低，中，高 $，而 BoN 任务包括预算和双目标权重，两者的目标权重范围为 0.1 到 0.9。例如，在有帮助-无伤害任务中，上下文可能被表示为 {helpful : 0.3, harmless : 0.7, budget : high(1.0)} 。更多细节，包括所有提示，都在补充材料中提供。

为了构建环境，我们首先使用 ChatGPT-O3 为每个任务生成了一组指令提示。然后，我们为每对提示-查询生成了 128 个响应，并使用一个类别模型估计分数分

布。所有的完成都是使用 LLaMA-3.3-70B-Instruct 生成的，这是一种广泛使用的开源模型（Meta_AI 2024），我们在整个实验中将其视为黑箱。生成是在 8 台 A100 GPU 集群上使用 vLLM (Kwon et al. 2023) 进行的，总计大约 2,000 个 GPU 小时。一旦环境构建完成，所有实验都可以通过标准 CPU 快速运行。我们将随论文一起发布环境和代码，从而在没有任何实质计算资源的情况下实现完全可重复性。

评价协议。 所有报告的曲线都是 200 次独立运行的平均结果。对于合成环境，我们实例化 200 个独立环境并报告其平均表现。对于剩下的四个环境，每次运行都会重新洗牌数据集，执行 80/20 训练-测试分割，并在训练集上训练策略。在所有六个环境中，我们使用 10,000 样本在测试集上评估 ACR。每个预算的性能是 200 次运行的平均值，并附有标准误差 (SEM) 误差条。统计显著性通过 Wilcoxon 配对双侧测试与 alpha 0.05 评估，在讨论中我们会指出何时差异显著。完整的结果集在附录中。

探索策略的比较（图 2）。 、Psst 和 Top-K 筛选启发式方法始终优于所有基线。在所有六个领域中，当每个上下文的动作空间较大时（$|\mathcal{P}|N_{\max} \in [640, 1536]$），UCB、softmax 和 ε -贪婪方法在探索上显得乏力。在基线中，UCB 在某些领域中经过 $T = 20$K 后表现相当，但这需要广泛的超参数调整。此外，这些基线是完全顺序的，无法利用批量探索的成本和计算效率优势。完整的 Psst 在四种设置中获得最佳最终性能，而 Psst + KX 通常更快地达到强策略，在预算较小时，在四个实际数据任务中的三项中等同或超越 Psst 。然而，在激进修剪（小 K ）下，该启发式方法表现不佳——特别是在摘要和合成基准上——这表明在紧张的预算下，Psst + KX 更具吸引力，而在关键任务（如长时间、高频部署）中，完整的 Psst 更为可取。最后，统计测试也证实 Psst ，与 Top-K 筛选一起，在所有六个数据集和几乎所有预算下显著优于基线。这些发现表明，我们的方法在实际设置中使用尽可能少的 5K 到 20K 推理调用可靠地发现对齐良好的解决方案。

我们研究了推理意识在提示优化中的作用。在所有六个数据集中，Iapo 方法显著优于不考虑推理的的方法，这表明同时优化提示和推理规模时可以取得的收益。TRIPLE($N = 1$) 方法失败是因为它没有利用推理缩放。另一方面，TRIPLE($N = Random$) 失败是因为它没有针对不同的上下文优化缩放。筛选变体 PSST +$K1$ ——有效地近似于一种近乎解耦（仅提示）的过程——在大多数情况下未能达到最佳，仅在 CommonsenseQA 上表现竞争力，并在摘要生成上表现明显不佳。这是因为它陷入了无法缩放的误导性提示中，而这些提示相比单次操作表现不佳但在扩展下显著改善的提示要差。这些发现强调了 Iapo 在调整黑盒 LLMs 中的重要作用以及不连贯优化的陷阱。总体而言，Iapo 比不连贯优化高出多达 25 %，并且比仅提示优化高出多达 50 %。

我们提出了一种推理感知的提示优化（Iapo ）框架，用于对齐黑盒 LLM，强调提示和部署时推理扩展策略是紧密耦合的，应该共同优化。我们提出的 Psst 和 Top-K 筛选启发式方法在六种不同的设置中相对强基准展示了一致的改进。展望未来，我们计划探索更丰富的推理扩展策略（例如，自适应 BoN/MV 时间表、停止规则和树搜索）。我们还旨在将该框架扩展到具有明确成本/延迟约束的多目标对齐，并研究在分布转移下的长时间部署。

本研究部分由美国陆军 DEVCOM 分析中心（DAC）在合同编号 W911QX23D0009 的支持下进行，由国家科学基金会资助的项目 2205153、2321786 和 2416460，以及 Schmidt Sciences 在 AI 安全科学项目下的支持。

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862.

Chow, Y.; Tennenholtz, G.; Gur, I.; Zhuang, V.; Dai, B.; Kumar, A.; Agarwal, R.; Thiagarajan, S.; Boutilier, C.; and Faust, A. 2025. Inference-aware fine-tuning for best-of-N sampling in large language models. In The Thirteenth International Conference on Learning Representations.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research, 7(39): 1079–1105.

Fabiano, N.; and Cazenave, T. 2021. Sequential halving using scores. In Advances in Computer Games: 17th International Conference, ACG 2021, Virtual Event, November 23–25, 2021, Revised Selected Papers, 41–52. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-031-11487-8.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The Llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Gui, L.; Gârbacea, C.; and Veitch, V. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. Advances in Neural Information Processing Systems, 37: 2851–2885.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. arXiv preprint arXiv:2103.03874.

Huang, J. Y.; Sengupta, S.; Bonadiman, D.; Lai, Y.-a.; Gupta, A.; Pappas, N.; Mansour, S.; Kirchhoff, K.; and Roth, D. 2024. Deal: Decoding-time alignment for large language models. arXiv preprint arXiv:2402.06147.

Karnin, Z. S.; Koren, T.; and Somekh, O. 2013. Almost optimal exploration in multi-armed bandits. In International Conference on Machine Learning.

Krishna, K.; Chang, Y.; Wieting, J.; and Iyyer, M. 2022. Rankgen: Improving text generation with large ranking models. arXiv preprint arXiv:2205.09726.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Meta_AI. 2024. The Llama 3 Model Family: A Path to Openly Accessible Frontier Models. arXiv preprint arXiv:2404.11225.

OpenAI. 2024. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. OpenAI Blog.

Sessa, P. G.; Dadashi, R.; Hussenot, L.; Ferret, J.; Vieillard, N.; Ramé, A.; Shariari, B.; Perrin, S.; Friesen, A.; Cideron, G.; et al. 2024. BOND: Aligning LLMs with best-of-n distillation. arXiv preprint arXiv:2407.14622.

Shi, C.; Yang, K.; Chen, Z.; Li, J.; Yang, J.; and Shen, C. 2024. Efficient prompt optimization through the lens of best arm identification. Advances in Neural Information Processing Systems, 37: 99646–99685.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2020. Learning to summarize from human feedback. In NeurIPS.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171.

Xu, Y.; Sehwag, U. M.; Koppel, A.; Zhu, S.; An, B.; Huang, F.; and Ganesh, S. 2024. GenARM: Reward guided generation with autoregressive reward model for test-time alignment. arXiv preprint arXiv:2410.08193.

Yang, R.; Pan, X.; Luo, F.; Qiu, S.; Zhong, H.; Yu, D.; and Chen, J. 2024. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. arXiv preprint arXiv:2402.10207.

Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Song, S.; and Huang, G. 2025. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? arXiv preprint arXiv:2504.13837.

附录 A

定理 1 的证明

Theorem 1 (Error of PSST). 设 $R = \lceil \log_2 |\mathcal{A}| \rceil$ 为修剪轮数，$[o_k^{\min}, o_k^{\max}] = [-1, 1]$ 并定义成本-差距复杂度

$$H_1^c = \max_{(c,a_i)\neq a_1^c} \frac{\bar{N}_{max}}{\Delta_{c,a^i}^2}, \quad H_1 = \max_c H_1^c.$$

$$H_2^c = \max_{(c,a_i)\neq a_1^c} \frac{i\bar{N}_{max}}{\Delta_{c,a^i}^2}, \quad H_2 = \max_c H_2^c.$$

where, $\Delta_{c,a^i} = Q_{c,a^1}^{\alpha} - Q_{c,a^i}^{\alpha}$ ，臂根据该上下文中的 $Q_{c,a}^{\alpha}$ 以升序索引，并且 $\bar{N}_{max} = \frac{a_1(N)+N_{max}}{2}$ 。在结构感知分配下运行 PSST ，总的提示复杂度为 T ，返回每个上下文中的最优臂，概率至少为

$$1 - 3|\mathcal{C}|R\exp\Big(-\tfrac{T}{min(2|\mathcal{P}|H_1, 8|C|H_2)R}\Big).$$

。同样，为了确保失败概率最多为 δ ，选择

$$T = O\Big(min(|\mathcal{P}|H_1, |\mathcal{C}|H_2)R\log\Big(\tfrac{|\mathcal{C}|R}{\delta}\Big)\Big).$$

就足够了。

Lemma 1. 在上下文 c 下的最佳选项在第 r 轮被上下文 c 排除的概率至多为

$$2\exp\Big(-\tfrac{T}{2|\mathcal{P}|H_1^c R}\Big)$$

Proof. 假设在第 r 轮之前最佳臂没有被淘汰。那么根据 Hoeffding 不等式，对于任何臂 $a^i \in \mathcal{A}_c^{(r)}$ ，

$$\Pr\big[\hat{Q}_{c,a_1}^{\alpha,(r)} < \hat{Q}_{c,a_i}^{\alpha,(r)}\big] \;\le\; \exp\Big(-\frac{1}{2}harmonic(t_{r_1}, t_{r_i})\,\Delta_{c,a_i}^2\Big).$$

。这里，t_r 是用于估计 Q 值的样本数。令 N_r 表示 $A_c^{(r)}$ 中经验平均值比最佳臂大的臂的数量，我们有：

$$\begin{aligned}
\mathbb{E}[N_r] &= \sum_{a_i\in\mathcal{A}_c^{(r)}} \Pr\big[\hat{Q}_{c,a_1}^{\alpha,(r)} < \hat{Q}_{c,a_i}^{\alpha,(r)}\big] \\
&\le \sum_{a_i\in\mathcal{A}_c^{(r)}} \exp\Big(-\frac{1}{2}harmonic(t_{r_1}, t_{r_i})\Delta_{c,a_i}^2\Big) \\
&\le \sum_{a_i\in\mathcal{A}_c^{(r)}} \exp\Big(-\Delta_{c,a_i}^2 \cdot \tfrac{T}{2|\mathcal{P}|\bar{N}_i\log_2|\mathcal{A}|}\Big) \\
&\le |\mathcal{A}_c^{(r)}| \max_{i\in\mathcal{A}_c^{(r)}} \exp\Big(-\Delta_{c,a_i}^2 \cdot \tfrac{T}{2|\mathcal{P}|\bar{N}_{max}\log_2|\mathcal{A}|}\Big) \\
&\le |\mathcal{A}_c^{(r)}| \exp\Big(-\tfrac{T}{2|\mathcal{P}|H_1^c R}\Big)
\end{aligned}$$

。要使最佳臂在第 r 轮被淘汰，必须满足 $N_r \ge \frac{1}{2}|\mathcal{A}_c^{(r)}|$ 。因此推导得出该引理。 □

Lemma 2. 在轮次 r 中，从情境 c 排除掉情境 c 下的最佳臂的概率至多为

$$3\exp\Big(-\tfrac{T}{8|\mathcal{C}|H_2^c R}\Big)$$

。

Proof. 证明直接遵循 (Karnin, Koren, and Somekh 2013) 引理 4.3。唯一需要识别的是：

$$\begin{aligned}
\mathbb{E}[N_r] &= \sum_{a_i\in\mathcal{A}_c^{(r)}} \Pr\big[\hat{Q}_{c,a_1}^{\alpha,(r)} < \hat{Q}_{c,a_i}^{\alpha,(r)}\big] \\
&\le \sum_{a_i\in\mathcal{A}_c^{(r)}} \exp\Big(-\Delta_{c,a_i}^2 \cdot \tfrac{2^r T}{8|\mathcal{C}||\mathcal{A}|\bar{N}_i\log_2|\mathcal{A}|}\Big)
\end{aligned}$$

□

Proof of Theorem 1. 最优的臂需要在所有 R 轮次以及在所有上下文 $\mathcal{C}$ 下存活。因此，从引理 ?? :

$$\sum_{r=1}^{R}\sum_{c} 2\exp\Big(-\tfrac{T}{2|\mathcal{P}|H_1^c R}\Big) \;\le\; 3|\mathcal{C}|R\exp\Big(-\tfrac{T}{2|\mathcal{P}|H_1 R}\Big)$$

从引理 ?? :

$$\sum_{r=1}^{R}\sum_{c} 3\exp\Big(-\tfrac{T}{8|\mathcal{C}|H_2^c R}\Big) \;\le\; 3|\mathcal{C}|R\exp\Big(-\tfrac{T}{8|\mathcal{C}|H_2 R}\Big)$$

结合两者:

$$3|\mathcal{C}|R\exp\Big(-\tfrac{T}{min(2|\mathcal{P}|H_1,8|\mathcal{C}|H_2)R}\Big)$$

这就得出了定理。 □

Proposition 2 (Inference-Agnostic Optimality). 推理无关的提示优化策略在对 $R_x^{\mathrm{IA}}(c,a)$ 进行线性变换下依然保持最优，即，$kR_x^{\mathrm{IA}}(c,a), k>0$ 和最优策略可以通过对 Q 函数进行仿射变换从而简单地恢复:

$$Q^{AF}(c,a) := \mathbb{E}_{x\sim\mathcal{X}}\big[aR_x^{IA}(c,a)+b\big] = kQ^{IA}(c,a)+b.$$

Proof. 直接遵循詹森不等式。 □

附录 B

我们考虑一个有 $P=32$ 个提示的情境，每个提示在查询难度层次的隐藏混合——{easy, medium, hard}——上进行评估，涵盖了 $|\mathcal{X}|=520$ 个查询，比例为 6 : 4 : 3 。对于每个提示 p 和查询 x ，单次成功概率记为 $q_p(x)\in[0,1]$ 。

对例子 x 的提示 p 进行 $N\le N_{\max}$ 次拉取生成独立同分布的 $\{c_i\}_{i=1}^{N}$ Bernoulli 结果，其中 $\Pr(c_i=1)=q_p(x)$ ，每个完成引发每个样本的成本 k_p 。结果是一个数组 $[\,c_i,\,k_p\,]_{i=1}^{N}$ 。

多数投票 (MV) 设置 $M=1$ 如果 $\sum_i c_i > N/2$ ，$M=0$ 如果 $\sum_i c_i < N/2$ ，并在平局情况下 (N 偶数，$\sum_i c_i = N/2$) 掷公平硬币来分配 $M=0.5$ 。

针对上下文 $c\in\{\text{low},\text{mid},\text{high}\}$ 的成本效用计算为

$$u_c = w_1 M + w_2(c)\sum_{i=1}^{N} k_p,$$

，其中 $w_1=1$ 和 $w_2(c)\in\{0,\,-0.2,\,-1.0\}$ 取决于成本层级。

为了实例化环境，我们生成两种提示原型：欺骗性提示，这种提示在困难查询上表现低 $q_p(x)$ 但能达到较高的平均准确性；以及全面型提示，这种提示在不同层次上更均匀地保持中等准确性。每个提示的成本 k_p 是从均值为 0.02 和方差为 0.005 的正态分布中抽样得到的。

我们建模 $P=32$ 个提示，每个提示配对 $|\mathcal{X}|=512$ 个查询和 $K=2$ 个积极目标。对于每个 (p,x) ，有 M 个分类结果，每个结果由一个向量 $o_j\in\mathbb{R}^K$ 表示。在查询 x 上为提示 p 执行 $N\le N_{\max}(=32)$ 次产生 N 个独立同分布的结果向量，形成行 $[\,o_{i,1},\,o_{i,2},\,k_p\,]$ ，其中 k_p 表示提示 p 的每次完成成本。

给定一个具有权重 $w=(w_1,w_2,w_{\text{cost}})$ 的上下文 c ，其中 $w_1+w_2=1$ 和 $w_{\text{cost}}\le 0$ ，则最优 N 效用定义为

$$u_c = \max_{1\le i\le N}\,(w_1 o_{i,1} + w_2 o_{i,2}) + w_{\text{cost}}\,N\,k_p.$$

为了构建环境，从 $\{-4,\dots,4\}^2$ 中采样结果向量。我们实例化了两种提示原型：HMLV（高均值，低方差；擅长于 N=1 ）和 LMHV（低均值，高方差；从更大的 N 中获益），每种专门针对一个目标。对于每个 (p,x) ，我们在类别结果概率中加入小的每个查询噪声，通过扰动这些概率引入轻微的训练到测试的转变，采样每个提示的成本 $k_p\in[0.02,\,0.1]$ ，并从满足 $w_1+w_2=1$ 和 $w_{\text{cost}}\in\{-0.1,\,-0.5,\,-1.0\}$ 的网格中提取上下文权重。

我们从 MATH 数据集中选择了 316 个整数答案问题。使用 ChatGPT-o3 编写了一组 25 个提示模板。对于每个 (prompt, problem) 对，我们从 Llama-3.3-70B-Instruct 在温度 $T=0.7$ 下采样 128 个回应，解析每个完成项以获得最终的整数答案。

然后按如下方式处理数据集:

1. 对于每个问题，保留全球排名前四的回答，并将所有其他回答归入一个 other 分类桶（总共 $C=5$ 个类别）。
2. 计算每个提示的成本为其响应的标准化平均标记长度。

这产生了一个类别环境（类似于合成类别设置），其中包含 $P=25$ 、$N_{\max}=32$ 、一个统一的上下文先验 $c\in\{\text{low, mid, high}\}$ 和成本系数 {0, 0.2, 1.0} 。效用通过多数投票评估。

我们从 CommonsenseQA 语料库中随机抽取了 1,500 道多项选择题，并使用 ChatGPT-o3 编写 48 个提示模板。对于每一对 (prompt, question) ，我们在温度 $T=1.1$ 下对 Llama-3.3-70B-Instruct 进行查询，收集 128 个 JSON 约束的答案（在 “选项 A” 到 “选项 E” 中）。每个提示分配一个恒定的成本 $k_p=0.01$ 。

所得数据用于构建一个分类环境（类似于合成分类设置），其中包含 $P=48$ 、$N_{\max}=32$ 、一个均匀的上下文先验，以及成本系数 {0, 0.2, 1.0} 。

我们过滤包含单个用户查询和单个助手响应的 HH-RLHF 对话 [2] 到 1,355 示例中。使用 ChatGPT-o3 ，我们制作 20 提示模板。对于每个 (prompt, query) 对，我们从温度为 $T=0.7$ 的 Llama-3.3-70B-Instruct 中采样 128 延续。每个延续由单独的公共奖励模型 (Yang et al. 2024) 评分，为了有用性 [3] 和无害性 [4] ，分数被标准化为 $[-1,1]$ 。

然后将两个奖励得分放入一个间隔为 0.5 的网格中，生成每个 (prompt, query) 的类别分布；每提示的成本计算为平均标记长度。该数据定义了一个具有 $P=20$ 和 $N_{\max}=32$ 的分类环境，在权重三重组 $(w_h,\,w_s,\,w_{\text{cost}})$ 上具有均匀上下文先验，且具有 $w_h+w_s=1$ 和 $w_{\text{cost}}\in\{-0.1,\,-0.5,\,-1.0\}$ 。

我们随机抽取 1,201 篇 Reddit 帖子，来自 Summarize-from-Feedback 语料库 [5] ，并用 ChatGPT-o3 设计 20 总结提示模板。对于每个 (prompt, post) 对，

[2] https://huggingface.co/datasets/Anthropic/hh-rlhf

[3] Ray2333/gpt2-large-helpful-reward_model

[4] Ray2333/gpt2-large-harmless-reward_model

[5] https://huggingface.co/datasets/openai/summarize_from_feedback

我们在温度 $T = 0.7$ 下查询 Llama-3.3-70B-Instruct ，并收集 128 个候选摘要。

每个摘要由两个公开可用的奖励模型评分：Preference [6] 和 Faithful [7] ，原始分数归一化为 $[-1, 1]$ 。然后，我们将每个维度按 0.5 的步长进行分箱，生成两个奖励维度上的分类分布，并根据平均标记长度计算每个提示的成本。

这些数据定义了一个类别环境，其中包含 $P = 20$ ，$N_{\max} = 32$ ，以及在权重三元组 $(w_h, w_s, w_{\text{cost}})$ 上的均匀上下文先验，其中 $w_h + w_s = 1$ 和 $w_{\text{cost}} \in \{-0.1, -0.5, -1.0\}$ 。

注意：所有提示都在代码库的提示文件夹下可用。对于筛选变体，我们在筛选后固定了 $K = 4$ 个候选项，并对烧入分数 $\rho \in \{0.05, 0.10, 0.20, 0.30, 0.40\}$ 进行了调整，它分配了预算的 ρ 部分用于在修剪前获得初始估计。参数扫描协议与基线相匹配。我们选择了 $\rho = 0.20$ 进行报告，因为它在保持数据集和推理方案的鲁棒性的一致性时，实现了整体性能最佳。

我们在相同预算下对探索常数进行调优，使用每个环境下 20% 的数据；在所有设置中使用相同的随机种子；它们是测试上下文)。代理通过标准 UCB 指数对手臂进行排名，其中 $\hat{\mu}_i(t)$ 是手臂 i 的经验平均效用，$n_i(t)$ 是其被拉动的次数，t 是总拉动次数。我们选择了 $c = 0.1$ 进行报告，因为它在不同的数据集和推理机制下都保持了最佳的总体性能和稳健性 3 。

我们分别对每个数据集和推理方案（MV ，BoN ）单独扫描了 $\varepsilon \in \{0.50, 0.75, 0.80, 0.85, 0.90, 0.95\}$ 。对于每个 ε ，代理在预算 $T \in \{3\text{K}, 5\text{K}, 10\text{K}, 20\text{K}, 30\text{K}, 40\text{K}\}$ 下进行训练，使用每个环境 20% 的数据并进行确定性重设种子；评估使用每个环境 10,000 个测试上下文。我们选择 $\epsilon = 0.15$ 进行报告，因为它在保持跨数据集和推理方案鲁棒性的同时实现了最佳的整体性能 4 。

对于每个数据集和预算 T ，我们使用每个环境的效用作为配对样本来进行所有算法配对比较（通过确定性重新播种的相同训练/测试拆分）。我们的默认测试是双侧 Wilcoxon 符号秩检验，我们在去除非有限值和消除完全平局后应用于对齐向量（zero_method=wilcox ，mode=auto ）。对少于两个有效样本的配对将被跳过。当要求时，在消除平局后，我们还会报告配对符号检验(对差异符号的二项式检验)。为了控制每个 $(\text{dataset}, T)$ 网格内的多重性，我们默认使用 Holm–Bonferroni 调整(有 Benjamini–Hochberg FDR 或不做校正的选项)。如果调整后的 $p < \alpha = 0.05$ ，我们宣布获胜者；方向由中位数差异的符号 $\text{median}(x - y)$ 决定。在算法之间环境数量不等的情况下，样本被截断到最小长度以保持配对。图形将结果矩阵可视化，其中 $\{-1, 0, +1\}$ 中的条目分别表示行算法相对于列算法的损失、不显著性或胜利。

所有结果如图 4 ，5 ，6 ，7 ，8 ，9 所示。在所有六个数据集上，我们观察到 PSST 和 Top-K 筛选启发式方法在大多数预算设置中始终优于竞争方法，并具有统计显著性。

[6] OpenAssistant/reward-model-deberta-v3-large-v2

[7] CogComp/bart-faithful-summary-detector

Param × T	HH	Summarization	SC	SB	MATH	CQA
ρ =0.05, $T = 3000$	0.40 ± 0.00	0.20 ± 0.00	2.77 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
ρ =0.05, $T = 5000$	0.40 ± 0.00	0.22 ± 0.00	2.83 ± 0.02	0.83 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.05, $T = 10000$	0.42 ± 0.00	0.21 ± 0.00	2.83 ± 0.02	0.85 ± 0.01	0.81 ± 0.01	0.76 ± 0.00
ρ =0.05, $T = 20000$	0.43 ± 0.00	0.22 ± 0.00	2.87 ± 0.02	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.05, $T = 30000$	0.44 ± 0.00	0.23 ± 0.00	2.88 ± 0.01	0.84 ± 0.00	0.82 ± 0.01	0.77 ± 0.00
ρ =0.05, $T = 40000$	0.43 ± 0.00	0.23 ± 0.00	2.87 ± 0.02	0.84 ± 0.00	0.81 ± 0.00	0.77 ± 0.00
ρ =0.10, $T = 3000$	0.41 ± 0.00	0.21 ± 0.00	2.79 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.10, $T = 5000$	0.41 ± 0.00	0.22 ± 0.00	2.84 ± 0.02	0.84 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.10, $T = 10000$	0.42 ± 0.00	0.21 ± 0.00	2.86 ± 0.02	0.84 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.10, $T = 20000$	0.43 ± 0.00	0.23 ± 0.00	2.88 ± 0.02	0.84 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.10, $T = 30000$	0.44 ± 0.00	0.23 ± 0.00	2.89 ± 0.02	0.84 ± 0.00	0.81 ± 0.01	0.77 ± 0.01
ρ =0.10, $T = 40000$	0.44 ± 0.00	0.23 ± 0.00	2.88 ± 0.02	0.84 ± 0.00	0.82 ± 0.00	0.77 ± 0.00
ρ =0.20, $T = 3000$	0.41 ± 0.00	0.20 ± 0.00	2.77 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.20, $T = 5000$	0.41 ± 0.00	0.22 ± 0.00	2.84 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.20, $T = 10000$	0.43 ± 0.00	0.22 ± 0.00	2.85 ± 0.02	0.83 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.20, $T = 20000$	0.43 ± 0.00	0.23 ± 0.00	2.87 ± 0.01	0.84 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.20, $T = 30000$	0.44 ± 0.00	0.23 ± 0.00	2.89 ± 0.02	0.84 ± 0.00	0.82 ± 0.01	0.76 ± 0.00
ρ =0.20, $T = 40000$	0.44 ± 0.00	0.22 ± 0.00	2.88 ± 0.02	0.84 ± 0.00	0.82 ± 0.01	0.77 ± 0.00
ρ =0.30, $T = 3000$	0.41 ± 0.00	0.21 ± 0.00	2.81 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.30, $T = 5000$	0.41 ± 0.00	0.22 ± 0.00	2.85 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.30, $T = 10000$	0.42 ± 0.00	0.22 ± 0.00	2.85 ± 0.02	0.84 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.30, $T = 20000$	0.43 ± 0.00	0.22 ± 0.00	2.88 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
ρ =0.30, $T = 30000$	0.44 ± 0.00	0.22 ± 0.00	2.88 ± 0.01	0.84 ± 0.00	0.82 ± 0.01	0.76 ± 0.00
ρ =0.30, $T = 40000$	0.44 ± 0.00	0.23 ± 0.00	2.89 ± 0.01	0.84 ± 0.00	0.82 ± 0.01	0.77 ± 0.00
ρ =0.40, $T = 3000$	0.41 ± 0.00	0.20 ± 0.00	2.80 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.40, $T = 5000$	0.42 ± 0.00	0.20 ± 0.00	2.80 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
ρ =0.40, $T = 10000$	0.43 ± 0.00	0.22 ± 0.00	2.85 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.40, $T = 20000$	0.43 ± 0.00	0.22 ± 0.00	2.87 ± 0.02	0.83 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.40, $T = 30000$	0.44 ± 0.00	0.22 ± 0.00	2.88 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.77 ± 0.00
ρ =0.40, $T = 40000$	0.44 ± 0.00	0.23 ± 0.00	2.89 ± 0.01	0.84 ± 0.00	0.82 ± 0.01	0.77 ± 0.00

Table 2: PSST+K4：跨数据集的平均 ± SEM（行是参数，ρ 和 T ）。

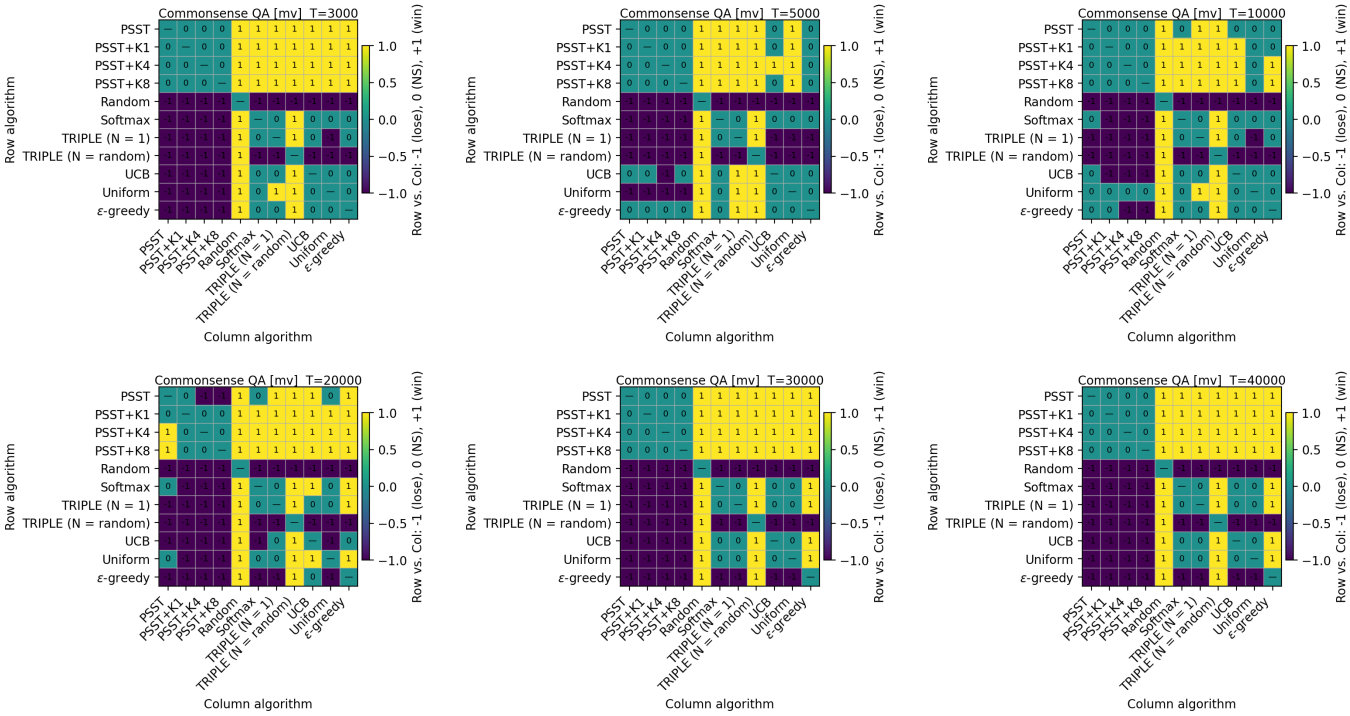


Figure 4: 在六个预算（ T 顺序为: 3000, 5000, 10000, 20000, 30000, 40000) 中常识问答 (MV) 的成对获胜次数。

Param × T	HH	Summarization	SC	SB	MATH	CQA
c=0.1, $T = 3000$	0.38 ± 0.00	0.19 ± 0.01	2.83 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=0.1, $T = 5000$	0.39 ± 0.00	0.21 ± 0.00	2.88 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=0.1, $T = 10000$	0.41 ± 0.00	0.23 ± 0.00	2.94 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=0.1, $T = 20000$	0.43 ± 0.00	0.25 ± 0.00	2.98 ± 0.01	0.86 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=0.1, $T = 30000$	0.43 ± 0.00	0.25 ± 0.00	2.99 ± 0.01	0.88 ± 0.00	0.81 ± 0.01	0.76 ± 0.01
c=0.1, $T = 40000$	0.44 ± 0.00	0.25 ± 0.00	3.00 ± 0.01	0.89 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
c=0.5, $T = 3000$	0.37 ± 0.00	0.19 ± 0.01	2.85 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=0.5, $T = 5000$	0.38 ± 0.00	0.20 ± 0.00	2.90 ± 0.01	0.82 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
c=0.5, $T = 10000$	0.41 ± 0.00	0.23 ± 0.00	2.94 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=0.5, $T = 20000$	0.43 ± 0.00	0.24 ± 0.00	2.98 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=0.5, $T = 30000$	0.43 ± 0.00	0.24 ± 0.00	3.00 ± 0.01	0.86 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=0.5, $T = 40000$	0.44 ± 0.00	0.25 ± 0.00	3.00 ± 0.01	0.88 ± 0.00	0.81 ± 0.01	0.75 ± 0.00
c=1.0, $T = 3000$	0.37 ± 0.00	0.19 ± 0.01	2.88 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=1.0, $T = 5000$	0.37 ± 0.00	0.19 ± 0.01	2.91 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
c=1.0, $T = 10000$	0.41 ± 0.00	0.22 ± 0.00	2.94 ± 0.01	0.83 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
c=1.0, $T = 20000$	0.42 ± 0.00	0.24 ± 0.00	2.98 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=1.0, $T = 30000$	0.43 ± 0.00	0.24 ± 0.00	3.00 ± 0.01	0.86 ± 0.01	0.81 ± 0.01	0.76 ± 0.00
c=1.0, $T = 40000$	0.43 ± 0.00	0.25 ± 0.00	3.00 ± 0.01	0.87 ± 0.00	0.81 ± 0.01	0.76 ± 0.01
c=2.0, $T = 3000$	0.37 ± 0.00	0.18 ± 0.01	2.86 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=2.0, $T = 5000$	0.38 ± 0.00	0.19 ± 0.01	2.93 ± 0.01	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
c=2.0, $T = 10000$	0.40 ± 0.00	0.23 ± 0.00	2.94 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=2.0, $T = 20000$	0.42 ± 0.00	0.24 ± 0.00	2.98 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=2.0, $T = 30000$	0.42 ± 0.00	0.24 ± 0.00	2.99 ± 0.01	0.86 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
c=2.0, $T = 40000$	0.43 ± 0.00	0.25 ± 0.00	2.99 ± 0.01	0.88 ± 0.00	0.81 ± 0.01	0.75 ± 0.00
c=4.0, $T = 3000$	0.37 ± 0.00	0.18 ± 0.01	2.85 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=4.0, $T = 5000$	0.37 ± 0.00	0.18 ± 0.00	2.91 ± 0.01	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
c=4.0, $T = 10000$	0.41 ± 0.00	0.22 ± 0.00	2.94 ± 0.01	0.84 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
c=4.0, $T = 20000$	0.42 ± 0.00	0.24 ± 0.00	2.97 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
c=4.0, $T = 30000$	0.42 ± 0.00	0.24 ± 0.00	2.99 ± 0.01	0.86 ± 0.00	0.81 ± 0.01	0.75 ± 0.00
c=4.0, $T = 40000$	0.43 ± 0.00	0.25 ± 0.00	3.00 ± 0.01	0.87 ± 0.00	0.81 ± 0.01	0.76 ± 0.00
c=8.0, $T = 3000$	0.37 ± 0.00	0.18 ± 0.01	2.86 ± 0.02	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
c=8.0, $T = 5000$	0.38 ± 0.00	0.19 ± 0.01	2.90 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
c=8.0, $T = 10000$	0.40 ± 0.00	0.22 ± 0.00	2.92 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=8.0, $T = 20000$	0.42 ± 0.00	0.24 ± 0.00	2.97 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
c=8.0, $T = 30000$	0.43 ± 0.00	0.24 ± 0.00	2.98 ± 0.01	0.86 ± 0.00	0.81 ± 0.01	0.75 ± 0.00
c=8.0, $T = 40000$	0.43 ± 0.00	0.25 ± 0.00	2.99 ± 0.01	0.87 ± 0.00	0.81 ± 0.01	0.76 ± 0.01

Table 3: UCB：跨数据集的平均 ± SEM（行代表参数，T ）。

Param × T	HH	Summarization	SC	SB	MATH	CQA
e=0.50, $T = 3000$	0.37 ± 0.00	0.17 ± 0.01	2.78 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
e=0.50, $T = 5000$	0.39 ± 0.00	0.20 ± 0.01	2.82 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
e=0.50, $T = 10000$	0.41 ± 0.00	0.21 ± 0.00	2.90 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.50, $T = 20000$	0.42 ± 0.00	0.23 ± 0.00	2.94 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.74 ± 0.00
e=0.50, $T = 30000$	0.43 ± 0.00	0.24 ± 0.00	2.97 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
e=0.50, $T = 40000$	0.43 ± 0.00	0.25 ± 0.00	2.98 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.01
e=0.55, $T = 3000$	0.38 ± 0.00	0.16 ± 0.01	2.75 ± 0.03	0.83 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.75, $T = 5000$	0.39 ± 0.00	0.17 ± 0.01	2.86 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
e=0.75, $T = 10000$	0.40 ± 0.00	0.20 ± 0.01	2.91 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
e=0.75, $T = 20000$	0.42 ± 0.00	0.23 ± 0.00	2.95 ± 0.01	0.84 ± 0.00	0.79 ± 0.00	0.74 ± 0.00
e=0.75, $T = 30000$	0.43 ± 0.00	0.23 ± 0.00	2.97 ± 0.01	0.85 ± 0.00	0.81 ± 0.01	0.75 ± 0.00
e=0.75, $T = 40000$	0.43 ± 0.00	0.24 ± 0.00	2.96 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.74 ± 0.00
e=0.80, $T = 3000$	0.38 ± 0.00	0.18 ± 0.01	2.83 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.80, $T = 5000$	0.39 ± 0.00	0.19 ± 0.00	2.86 ± 0.02	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.80, $T = 10000$	0.40 ± 0.00	0.19 ± 0.01	2.91 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.80, $T = 20000$	0.42 ± 0.00	0.23 ± 0.00	2.94 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.80, $T = 30000$	0.41 ± 0.00	0.23 ± 0.00	2.96 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.80, $T = 40000$	0.43 ± 0.00	0.24 ± 0.00	2.98 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.74 ± 0.00
e=0.85, $T = 3000$	0.38 ± 0.00	0.16 ± 0.01	2.72 ± 0.04	0.83 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
e=0.85, $T = 5000$	0.38 ± 0.00	0.17 ± 0.01	2.87 ± 0.01	0.83 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.85, $T = 10000$	0.40 ± 0.00	0.20 ± 0.00	2.90 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.85, $T = 20000$	0.41 ± 0.00	0.22 ± 0.00	2.95 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.85, $T = 30000$	0.42 ± 0.00	0.23 ± 0.01	2.95 ± 0.01	0.85 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.85, $T = 40000$	0.42 ± 0.00	0.24 ± 0.00	2.97 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.01
e=0.90, $T = 3000$	0.37 ± 0.00	0.17 ± 0.01	2.81 ± 0.03	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
e=0.90, $T = 5000$	0.38 ± 0.00	0.17 ± 0.01	2.87 ± 0.02	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
e=0.90, $T = 10000$	0.40 ± 0.00	0.19 ± 0.01	2.90 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.90, $T = 20000$	0.41 ± 0.00	0.22 ± 0.00	2.94 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.90, $T = 30000$	0.42 ± 0.00	0.23 ± 0.00	2.95 ± 0.01	0.85 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.90, $T = 40000$	0.42 ± 0.00	0.24 ± 0.00	2.97 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
e=0.95, $T = 3000$	0.37 ± 0.00	0.17 ± 0.01	2.76 ± 0.03	0.82 ± 0.00	0.78 ± 0.01	0.75 ± 0.00
e=0.95, $T = 5000$	0.38 ± 0.00	0.17 ± 0.01	2.86 ± 0.01	0.83 ± 0.00	0.79 ± 0.01	0.76 ± 0.00
e=0.95, $T = 10000$	0.39 ± 0.00	0.19 ± 0.01	2.92 ± 0.01	0.84 ± 0.00	0.80 ± 0.01	0.76 ± 0.00
e=0.95, $T = 20000$	0.41 ± 0.00	0.21 ± 0.00	2.95 ± 0.01	0.84 ± 0.00	0.79 ± 0.01	0.75 ± 0.00
e=0.95, $T = 30000$	0.42 ± 0.00	0.22 ± 0.00	2.95 ± 0.01	0.85 ± 0.00	0.80 ± 0.01	0.75 ± 0.00
e=0.95, $T = 40000$	0.41 ± 0.00	0.23 ± 0.00	2.95 ± 0.01	0.84 ± 0.00	0.79 ± 0.00	0.75 ± 0.00

Table 4: ϵ -贪婪策略：跨数据集的均值 ± 标准误（行是参数，T ）。

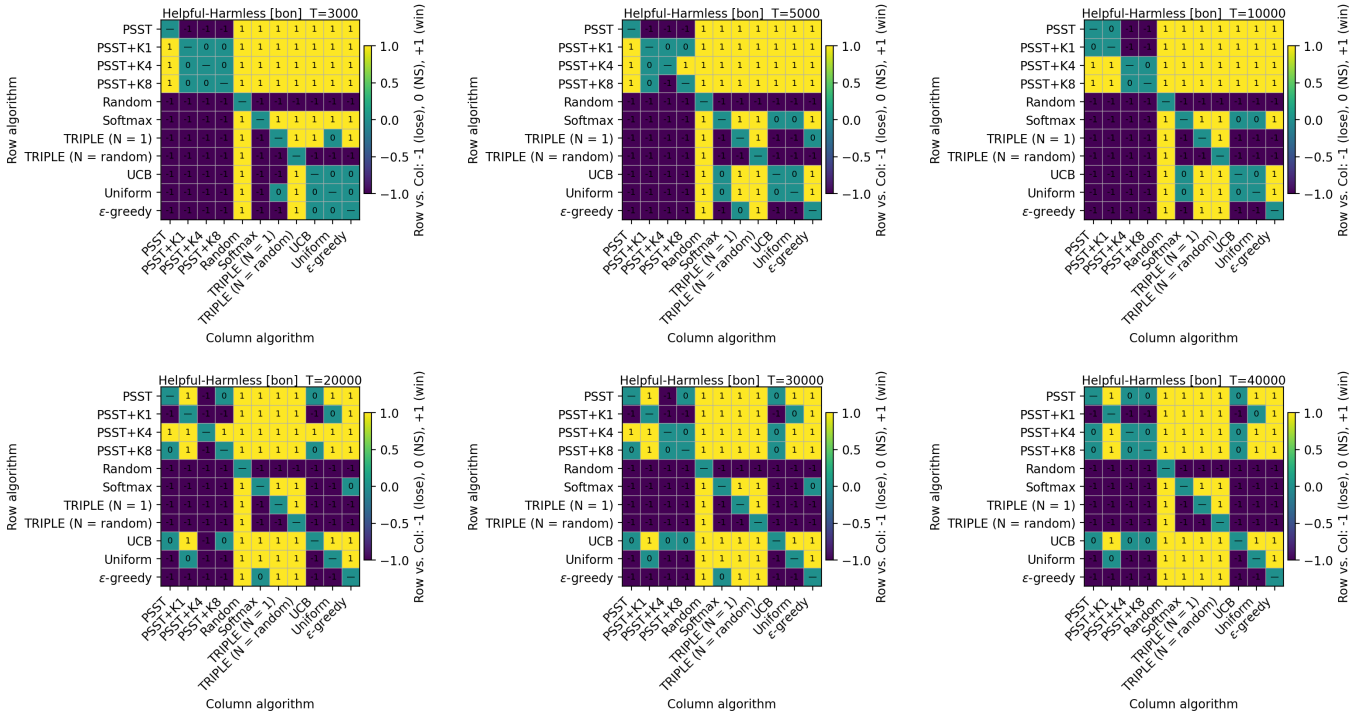


Figure 5: 在预算为 6 种（T 按顺序为 3000, 5000, 10000, 20000, 30000, 40000）的情况下，Helpful-Harmless（BoN）的成对胜出。

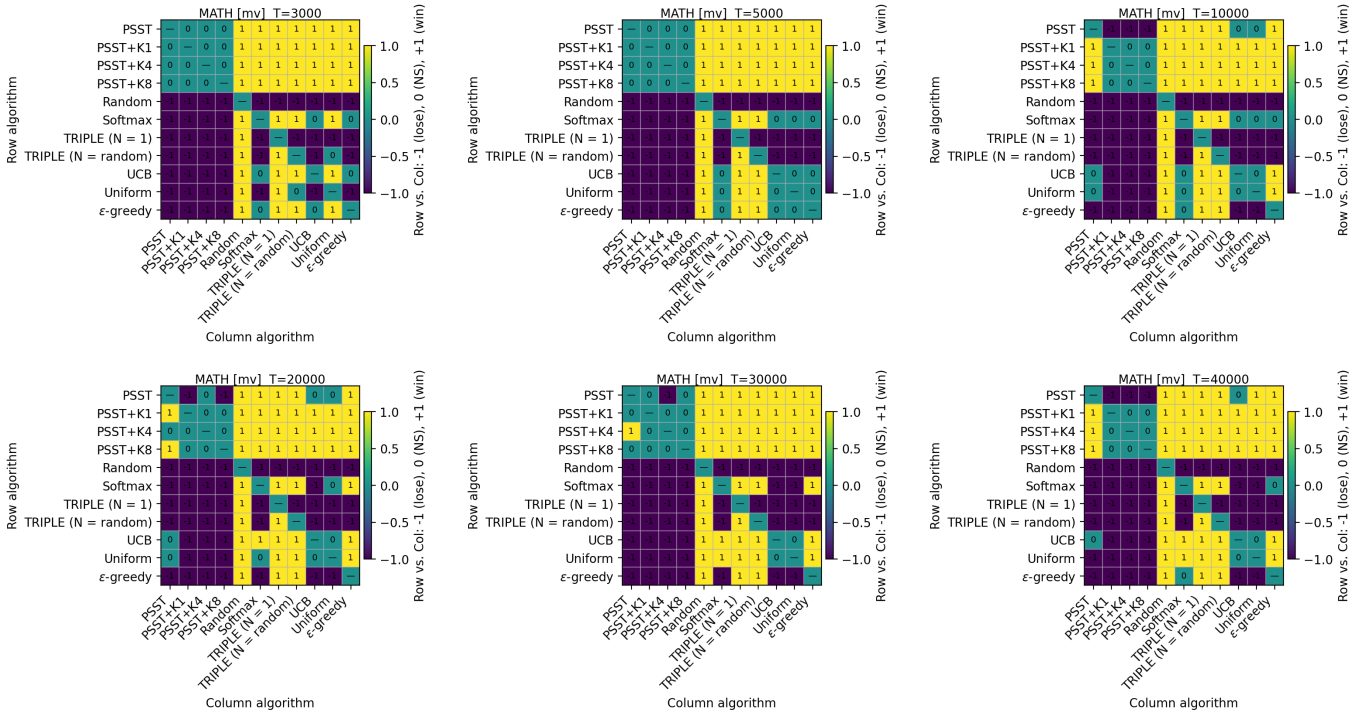


Figure 6: 在六个预算（按顺序为：3000、5000、10000、20000、30000、40000）中，MATH（MV）的成对胜利。

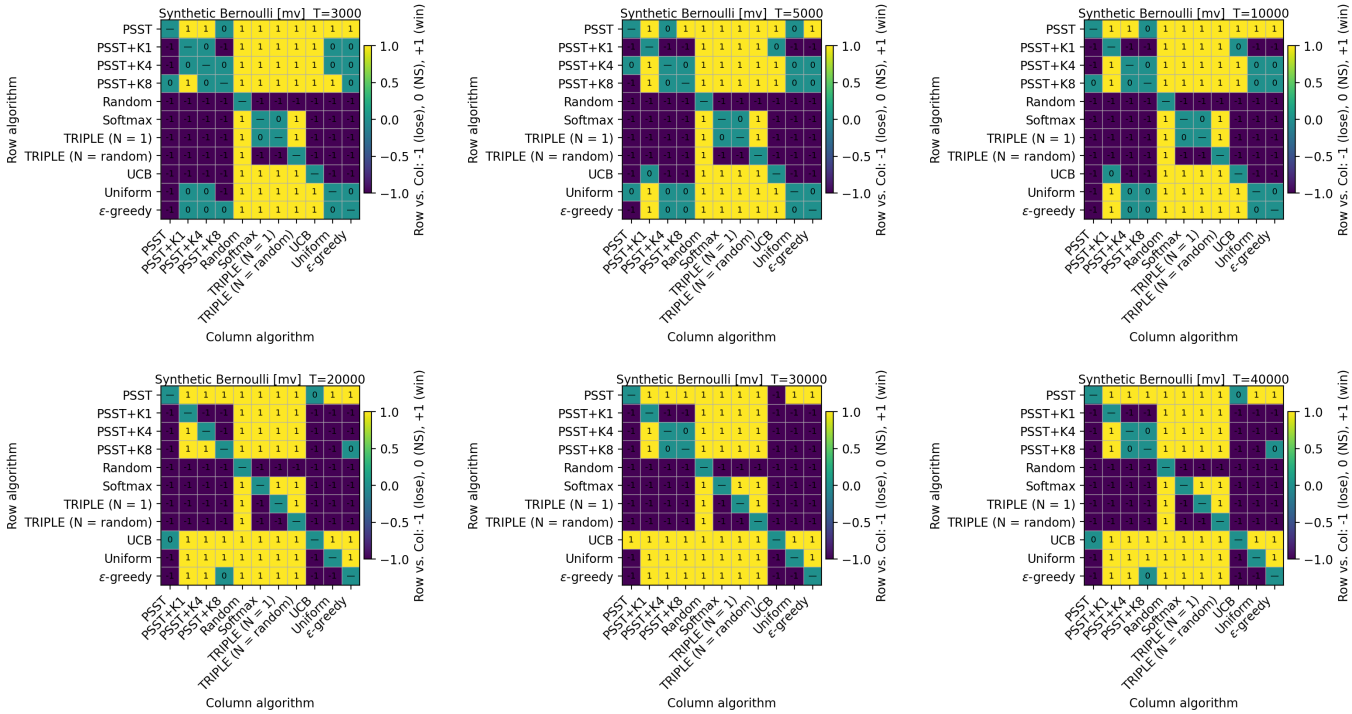


Figure 7: 合成伯努利（MV）在六个预算范围（T 依次为：3000, 5000, 10000, 20000, 30000, 40000）中的两两胜出。

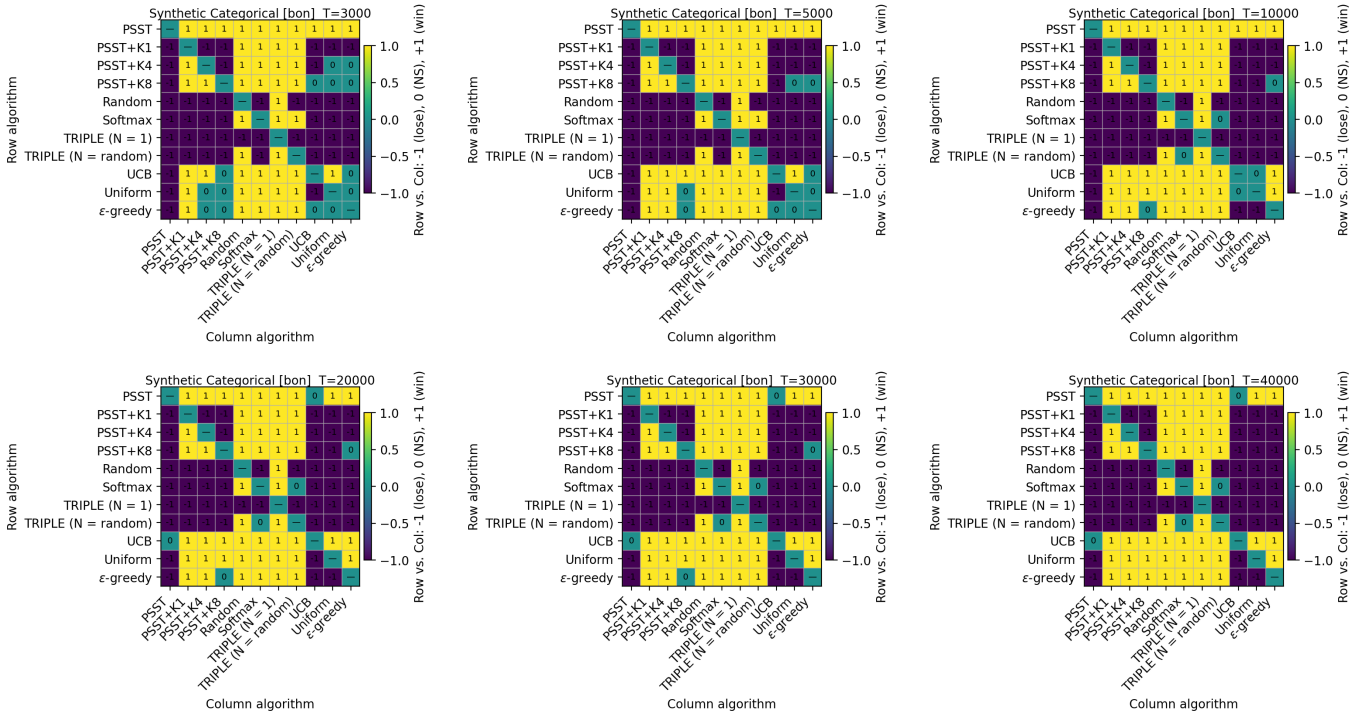


Figure 8: 6 个预算（T 顺序为：3000、5000、10000、20000、30000、40000）下合成类别（BoN）的成对胜利。

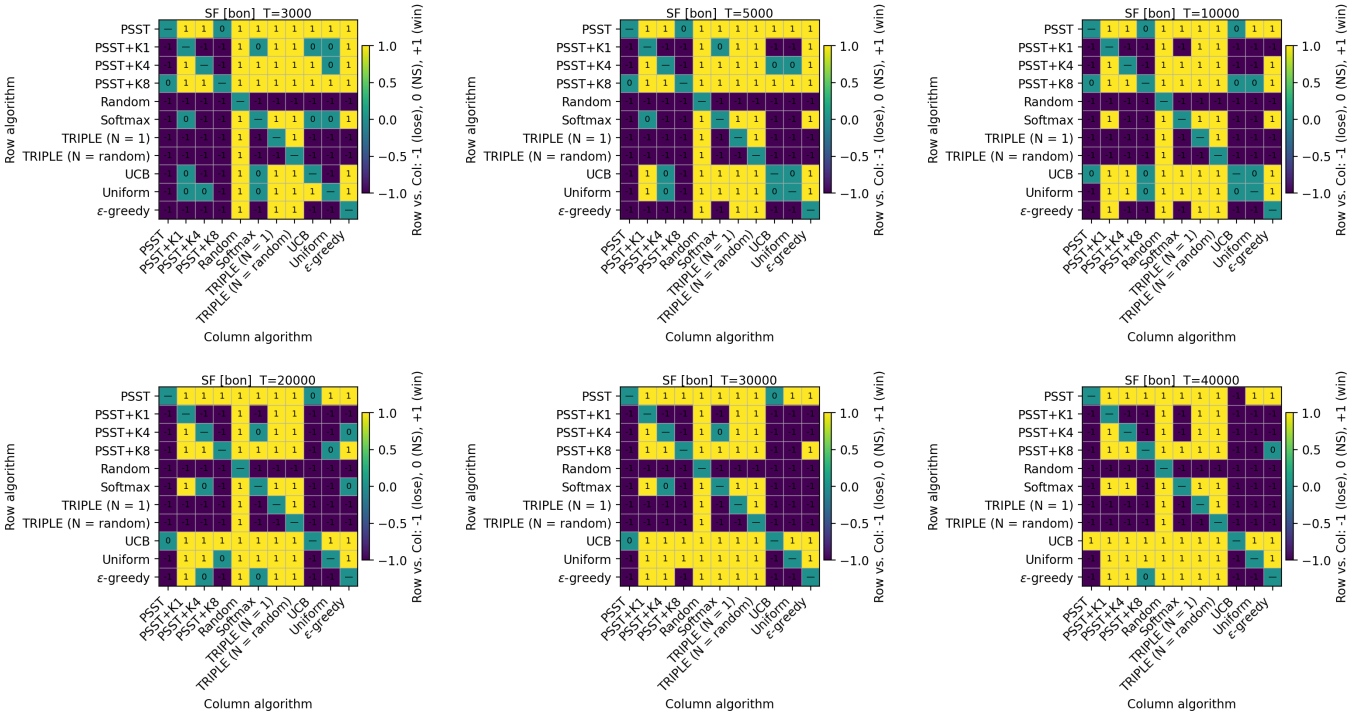


Figure 9: 在六个预算（T 分别为：3000, 5000, 10000, 20000, 30000, 40000）情况下，摘要的成对胜利（BoN）。