arXiv:2602.01708v1 [cs.CL] 2 Feb 2026

思维博弈：运用博弈论实现大型语言模型的稳健信息寻求

Langyuan Cui [1] **Chun Kai Ling** [1] **Hwee Tou Ng** [1]

摘要

大型语言模型（LLM）日益广泛地应用于现实场景，但在这些场景中，它们可能缺乏完成给定任务所需的充分信息。在这种情况下，主动寻求缺失信息的能力至关重要。现有增强这项能力的方法往往依赖简化假设，从而降低最坏情况性能。这一问题在高风险应用中会造成严重影响。在本研究中，我们使用「二十个问题」游戏来评估大语言模型的信息寻求能力。我们提出并形式化其对抗性对应问题——策略性语言搜索（SLS）问题及其变体，并将其建模为双人零和扩展式博弈。我们提出思维博弈（GoT）框架，运用博弈论技术，为该博弈的受限变体近似求解纳什均衡（NE）策略。实证结果表明，在所有测试设置中，与（1）直接提示方法和（2）启发式引导搜索方法相比，我们的方法始终能提升最坏情况性能。

1. 引言

大型语言模型（LLM）正日益被部署于规划等高风险环境（Zhang 等人，2024a）、医学诊断（Li 等人，2024年），以及其他具有部分可观测性的任务中（Li 等人，2025年）。在这类环境中，大语言模型可能没有足够的信息来完成指定任务，因此需要通过澄清问题等方式进行信息寻求。为了定量评估信息寻求能力，我们采用二十问博弈（Siegler，1977）。在这个游戏中，首先从一个已知集合中选出一个项目。随后，玩家依次提出最多二十个“是/否”问题以尽可能少的问题识别该项目。

信息寻求通常需要一定的前瞻。常见的方法结合了提示和基于大语言模型的推理，例如自一致性（SC）（Wang 等人，2022年）和思维树（ToT）（Yao 等人，2023年）。最近，Hu 等人 (2024年) 提出了思维不确定性（UoT），利用树搜索来显式建模并优化澄清问题所带来的信息增益。然而，他们假设项目是均匀随机选取的。在现实世界中，这一假设不太可能成立，因为（i）情境本身并不自然地适用于概率解释（Jain，2016年），或（ii）即使存在某种分布，也可能难以获取，且很可能并非均匀分布（Choudhury 等人，2025年）。尤其是在高风险环境中，我们认为应当假设最坏情况。也就是说，项目是由对手对抗性地选取的，而我们的目标是采用一种能够优化最坏情况性能的提问策略。

受此启发，我们提出思维博弈（GoT），一种旨在处理此类场景的博弈论框架。我们将信息寻求过程建模为一个不完全信息的双人零和博弈。在这一形式化表述中，假设项目（分布）由对手选择，以阻碍信息寻求者。GoT 通过近似求解该博弈的纳什均衡（NE），优化最坏情况性能，无需对项目分布作出任何强先验假设。

贡献我们（**i**）定义了策略性语言搜索（SLS）问题及其变体，（**ii**）建立了能够最优求解 SLS 及其变体的博弈论解法，并指出这些策略必然是随机化的，（**iii**）提出 GoT，以近似求解 SLS 受限变体中的纳什均衡，（**iv**）通过实验证明，在各种设置下，GoT 在最坏情况下优于（a）纯提示方法和（b）基于 UoT 的方法。

2. 相关工作

思维不确定性 GoT 可被视为 思维不确定性 （UoT）由 Hu 等人 （2024年）提出的延伸。作者建议对

[1]新加坡国立大学计算机科学系，13 Computing Drive, Singapore 117417。通信作者：Langyuan Cui < langyuan.c@u.nus.edu>。

预印本。*2026*年*2*月*3*日。

可能提出的问题进行深度受限搜索，假设物品均匀随机选取，以寻求最大化期望信息增益。GoT 进行了类似的显式前瞻，但通过博弈论方法针对最坏情况下选出的物品进行优化，从而无需这一假设。

游戏中的自然语言 我们构建了将自然语言隐式视为动作空间的博弈。在博弈中，语言常被玩家用于交流，以进行协商（Kram´ar 等，2022年）、共享信息（Ri 等，2022年）和协调行动（Bishop 等，2020年）。在这一背景下研究语言颇具挑战，因为很难明确定义最优性的概念，也难以分析这些话语所产生的影响。这类交流类似于信号博弈中的信号（Lewis，1969年；Farrell 与 Rabin，1996年；Gintis 等，2001年）。然而，这些博弈通常考虑的是有限集信号，不同于自然语言交流。以往研究尝试通过多种技术弥合这一差异，例如情感分析（Capraro 等，2024年；Houser 与 Xiao，2011年），或根据博弈情境从大语言模型中采样（Gemp 等，2024年；Kempinski 等，2025年），利用其推理能力筛除无关话语，同时采用受博弈论启发的技术来设计策略。

大语言模型的信息寻求能力 先前研究考察了大语言模型在模糊查询或规格不明确的任务中的信息寻求能力。在这些场景中，大语言模型起初缺少完成任务所需的细节，因此必须提出澄清问题，以确定用户意图（Kuhn 等人，2022；Zhang 等人，2024a；Xu 等人，2019）或了解用户偏好（Handa 等人，2024；Li 等人，2023）。如此界定后，问题就转变为推理哪些额外信息能够使任务的规格变得明确。研究表明，基于探索的提示有助于解决这一问题（SC、ToT（Wang 等人，2022；Yao 等人，2023；Choudhury 等人，2025））；其他方法则采用启发式策略选择查询，例如依据预期信息增益（Grand 等人，2024；Piriyakulkij 等人，2023），其中包括 UoT（Hu 等，2024）。研究人员已围绕规格不明确的任务（Li 等人，2025）、医学诊断（Li 等人，2024；Chen 等人，2025）和故障排查（Raghu 等人，2021）构建了用于评估大语言模型信息寻求能力的基准测试，这些基准测试表明，这种能力在实际应用中十分重要。

3. 问题表述

虽然二十问游戏并非严格意义上的对抗性游戏，但我们提出一种对抗性变体，称为策略语言搜索（SLS），以便运用博弈论方法优化最坏情况。SLS 游戏在一个由 $\mathcal{S}$ 个互不相同物品组成的有限集 n 上进行，参与者是

两名相互竞争的（零和）玩家：物品选择者 和提问者。我们假设存在一个可数的二元（即真或假）问题集合 $\mathcal{Q}$。每个问题都由一个有限长度的自然语言字符串唯一指定。对于每个物品$s \in \mathcal{S}$，我们用 $f\colon \mathcal{Q} \times \mathcal{S} \to \{0,1\}$ 表示$q \in \mathcal{Q}$ 针对某个 $s \in \mathcal{S}$的答案 。一种自然但退化的 $\mathcal{Q}$ 是我们记作 $\mathcal{Q}_\infty$的那种。 $\mathcal{Q}_\infty$ 的一种可能的文本表示，是以布尔形式陈述的问题集合：“物品 1 或物品 3 是该物品，但不是物品 2 或物品 4 吗？”

一个 SLS 由 $(\mathcal{S},\ \mathcal{Q},\ f)$ 定义，并分为两个阶段。在第一阶段，物品选择者私下选择一个项目 $s^* \in \mathcal{S}$。在第二阶段，提问者通过依次提出一系列问题 $q_1,\ a_1,\ \ldots,\ q_T,\ a_T$ 并获得答案，来获取有关 s^* 的信息，其中$q_t \in \mathcal{Q}$和 $a_t = f(q_t,\ s^*)$，即针对 s^*对 q_t的回答。截至并包括时间 $t \geq 0$ 的问答历史记为 $H_t = (Q_t,\ A_t)$，其中 $Q_t = (q_1,\ \ldots,\ q_t)$ $A_t = (a_1,\ \ldots,\ a_t)$，满足 $Q_t(\tau) = q_\tau$且$A_t(\tau) = a_\tau$。给定历史 H，我们用 $H'(q,\ a)$ 表示提出新问题 q 并得到答案a 后的新历史。提问者在线选择问题，即 q_t 可以取决于其已观测历史 H_{t-1}。历史 H 的长度记为 $|H|$。

与某个已观测历史 $H = (Q, A)$ 相符的物品集合记为

$$S(H) = \{s \in \mathcal{S} | \forall \tau \in [|H|], f(Q(\tau), s) = A(\tau)\} \subseteq \mathcal{S}.$$

注意， $S(H)$ 永远不会为空，因为其中至少必须包含s^*。游戏持续进行，直到提问者能够确定无疑地识别出 s^*，即 $|S(H)| = 1$，随后游戏结束，物品选择者（或提问者）将承担 $|H|$的奖励（或成本）。提问者的目标是找到一种策略（或等价地，策略），使其期望成本最小化，而不论物品选择者如何选择 s^*，从而优化其最坏情况性能。这对应于零和博弈的纳什均衡 ，通常采用随机化策略。我们将在陈述若干假设和变体后，正式定义解概念。

假设3.1。 每个问题-物品对 $(q, s) \in \mathcal{Q} \times \mathcal{S}$都有唯一的答案 $f(q, s)$，且该答案与历史无关。

假设3.2。 固定 s^*后，物品选择者不能在回答问题时说谎。

假设3.3。 $\mathcal{S}$、 $\mathcal{Q}$ 和 f 是共同知识。

假设3.4。 对 f进行常数时间预言机访问。

假设3.5。 对于任意一对不同的物品 $s,\ s' \in \mathcal{S}$，都存在某个 $q \in \mathcal{Q}$，使得 $f(q,\ s) \neq f(q,\ s')$。

定理3.1 和 3.2 是内嵌于 f 定义中的隐含假设，而定理 3.5 则是对 $\mathcal{Q}$ 和 f施加的一项技术性假设。它确保存在一种策略，使提问者能够在有限时间内找到正确物品

。请注意，当 $\mathcal{Q}$ 包含“身份”问题（例如“物品是‘s’吗？”）时，该条件显然成立。

例1. 考虑包含 $\mathcal{S}$ 和 $\mathcal{Q}$ 的 *SLS*：

$$\left\{\begin{array}{l} s^{(1)}:\text{‘Oppenheimer’} \\ s^{(2)}:\text{‘Alan Turing’} \\ s^{(3)}:\text{‘A Beautiful Mind’} \end{array}\right\}, \left\{\begin{array}{l} q^{(1)}:\text{‘Related to codes?’} \\ q^{(2)}:\text{‘Is it a movie?’} \\ q^{(3)}:\text{‘Is it a person?’} \end{array}\right\}$$

那么，如果 $j=i$，则 $f(q^{(i)},s^{(j)})=0$；否则为 *1*。

很容易验证，定理 3.5在例 1中成立。依次选择 $q^{(1)}$ 和 $q^{(2)}$，即可保证提问者提出的问题不超过 2 个。物品选择者针对这一策略所能采取的最佳做法是选择 $s^*=s^{(2)}$ 或 $s^{(3)}$，因为这样可以保证在第一个问题之后始终有两个物品符合条件，即 $|S(H_1)|=2$。不过，提问者可以通过以均匀随机的方式选择第一个问题来做得更好。这样无论 s^*如何选择，都有 $1/3$的概率恰好通过一个问题确定该物品，因此提出的问题数量的期望为$1/3\cdot 1+2/3\cdot 2<2$。无论物品选择者选择什么，这一策略都能最小化最坏情况成本。在本例中，由于 $\mathcal{Q}$和 f具有循环对称性，提问者的最优策略恰好是对称的，算是幸运。不过，对于更一般的 $\mathcal{Q}$, f，情况并非如此。

迄今为止，SLS 主要被表述为一个组合问题。事实上，当 $\mathcal{Q}$ 为有限集时，我们有：

定理 3.6. 给定一个已知的（确定性） s^*，判断是否存在一系列 k 问题 $Q\subseteq\mathcal{Q}$ 使得$S(H)=\{s^*\}$ 是 *NP* 完全问题。

该归约可直接由集合覆盖问题得到，并收录于附录中。用博弈论术语来说，这意味着，要计算提问者对任意确定性物品选择者策略的最佳响应是困难的。

定理 3.7. 设$(\mathcal{S},\mathcal{Q}_\infty,f)$ 为一个 *SLS*，其中 $|\mathcal{S}|=2^k$对某个 $k\in\mathbb{Z}^+$成立。采用“均匀二分”策略，即选择满足 $\forall t$, q_{t+1}的$(H_t'(q_{t+1},0))|=|S(H_t'(q_{t+1},1))|$，可使提问者在最坏情况下提出的问题数量最少，且恰好需要 k 个问题。

定理3.7指出，在 $\mathcal{Q}=\mathcal{Q}_\infty$ 这一特殊情形下，问题会变得简单，这与“均匀拆分物品是最优策略”的直觉一致。我们在附录中证明，在 UoT 的设定下，当 s^* 被均匀随机选取时，均分策略是最优的（Hu 等人，2024年）；事实上，在博弈设定中，它们构成一个纳什均衡。这也解释了他们隐含采用的最大化熵损失的方法，以及使用鼓励均分的大语言模型提示。

受限SLS（SLSR）对于较大的 n 和 $\mathcal{Q}$，要推理标准SLS可能相当繁琐。因此，我们提出一种受限SLSR变体，其中的问题基于剩余项目生成 $S(H)$。SLSR的形式化定义为$(\mathcal{S},\mathcal{Q},f,g)$，其中 $g:2^{\mathcal{S}}\backslash\phi\to 2^{\mathcal{Q}}\backslash\phi$ 是一个集函数，输入非空物品集 $S\subseteq\mathcal{S}$ 并输出一个非空问题集，问题数量不超过 $m\geq 1$；其中 m 是与 g关联的参数。SLSR中的规则和收益与SLS相同，但提问者只能选择 $q_t\in g(S(H_{t-1}))$。为确保始终能够取得进展，我们对假设3.5 施加更强的要求，并要求 g 至少输出一个能严格缩小一致项目集合的问题。请注意，在SLSR博弈中，最优提问者的表现不会优于相应SLS博弈中的表现。因此，其在SLSR博弈中的性能可视为对其在SLS中性能的低估。此设计旨在考察最坏情况性能。

假设3.8。 对于每个子集 $S\subseteq\mathcal{S}$，都存在某个 $q\in g(S)$和一对不同的物品 s, $s'\in S$，其中$f(q,s)\neq f(q,s')$

加权SLS（WSLS）在现实场景中，某些物品可能更为重要，因此在信息寻求过程中应优先处理。例如，在医学诊断中，延迟识别危及生命的病症可能会造成更严重的后果。因此，我们提出加权SLS，这是由 $(\mathcal{S},\mathcal{Q},f,w)$ 定义的 SLS 变体，其中 $w:\mathcal{S}\to\mathbb{R}^+$。当提问者选择 s^* 作为项目时，其产生的成本重新定义为 $w(s^*)\cdot|H|$，从而对涉及高权重项目的长时间交互施加更大的惩罚。加权SLSR（WSLSR）以类似方式由 $(\mathcal{S},\mathcal{Q},f,g,w)$ 定义。二十个问题的非对抗博弈的相应加权变体也以类似方式定义。

例2. *SLSR* $(\mathcal{S},\mathcal{Q},f,g)$ 在 $\mathcal{S}$, $\mathcal{Q}$, f 与例 *1*相同，$g(S)=\{q^{(1)},q^{(2)}\}$ 当 $S=\mathcal{S}$ 且 $\mathcal{Q}$ 时；否则。

在例2中，第一个问题只能是$q^{(1)}$ 或 $q^{(2)}$。无论提问者如何在二者之间随机选择，物品选择者总能选择 $s^*=s^{(3)}$；这保证了 $a_1=1$，因此 $|S(H_1)|=2$。因此，提问者至少还需要再提一个问题，总共需要2个问题。

例3. *WSLS* $(\mathcal{S},\mathcal{Q},f,w)$ 在 $\mathcal{S}$, $\mathcal{Q}$, f 与例*1*相同，$w(s^{(1)})=3$, $w(s^{(2)})=w(s^{(3)})=2$。

在例 3中，提问者可以选择 $q^{(1)}$, $q^{(2)}$, $q^{(3)}$，概率为 $3/4$, $1/8$, $1/8$，因为 q_1。这确保期望成本不超过 $15/4$。相比之下，均匀地选择 q_1，当5被选择时，成本为 $s^*=s^{(1)}$。

备注 3.9。 在 SLS 及其变体中，只有当提问者完全确定 s^*时，博弈才会结束。另一种表述方式允许提问者在任何阶段明确猜测项目，但错误猜测会受到惩罚。

我们选择当前的表述方式，是因为它能更好地体现高风险环境中的约束；例如，在医学诊断中，人们希望排除所有其他可能性后再进行治疗。粗略地说，我们的表述方式会对错误猜测施加“无限”惩罚。

通过大型语言模型定义 SLS。 对于许多应用而言，$\mathcal{S}$ 规模过大，以至于我们无法切实地手动整理一组 $\mathcal{Q}$；同时，我们希望避免涉及冗长、复杂布尔表达式的“不自然”问题。因此，在本文中，我们使用大型语言模型（LLM）隐式定义 $\mathcal{Q}$、f和 g。具体而言，$\mathcal{Q}$ 是指 LLM 根据特定提示词提出的问题集合，而 f 则是 LLM 在被询问关于 s^*时，对 q 提示词的响应。例如，“奥本海默是一部电影吗？”。我们希望强调，f 可以由任何预言机实现，例如人类专家。为了便于实验，我们选择通过 LLM 来实现它。此外，在定义了 SLS 的规则以及 $S(H)$ 之后，SLSR 和 WSLSR 中的 g可以定义为 LLM 针对 $m \geq 1$问题的提示词所生成的输出，即剩余的物品。一个例子是：“当前可能的物品集合为 {奥本海默、艾伦·图灵}。请提出 m 个最佳问题。”

假设3.10。 用于对 $f(q, s^*)$ 进行预言机查询时，针对 f 的大语言模型不会出错。

假设3.10 和 3.1 使得将大语言模型作为定义 SLS 的黑盒在实践中成为可能。对于大多数大语言模型，假设3.4 也成立，不过常数因子可能较大。我们在附录中讨论了所有作出的假设。

4. SLS 作为零和扩展式博弈

标准 SLS（$\mathcal{S}, \mathcal{Q}, f$）可以表示为一个不完全信息的双人零和扩展式博弈（扩展式博弈）。扩展式博弈是有根博弈树，其中每个顶点对应于整个博弈历史。边表示该顶点处轮到行动的玩家可采取的行动。扩展式博弈带有信息集，这些信息集将属于同一玩家的顶点划分到一起；同一信息集中的顶点对于拥有它们的玩家而言无法区分。这体现了不完全信息的概念。

在 EFG 表述中，物品选择者只在根节点采取行动，之后提问者依次提出至多 $n-1$ 个问题。[1] 除根节点外，每个状态都可以由元组 (s^*, H) 唯一标识，该元组描述了所选项目 s^* 以及（可能为空的）过往问答历史。当满足以下任一条件时，该节点为叶节点：(i) $|H| = n-1$，即已提出最多数量的问题

[1]这使博弈树保持有限，并且这是合理的，因为重复提出相同问题并非最优。

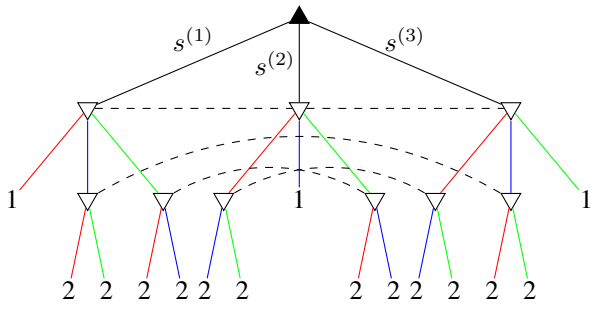

图1。 示例 1的扩展式博弈表示。根节点属于物品选择者，▲其余节点属于提问者。黑色边对应于项目 s^*的选择。红色、蓝色和绿色 边表示问题 $q^{(1)}$、$q^{(2)}$和 $q^{(3)}$。由虚线连接的顶点属于同一信息集。叶节点显示物品选择者（分别为提问者）的收益（分别为成本）。对于相同问题被重复提出的边（及其后继节点），我们将其省略；这些行动从不处于最优状态，并对应于被支配行动。

，或 (ii) $|S(H)| = 1$，其中 s^* 已确定。在叶节点处，物品选择者获得的奖励（提问者付出的成本）为 $|H|$。具有相同历史 H 的所有非叶节点均属于同一信息集，我们将其记为$I(H)$。图 1 展示了示例1的 SLS。

提问者的确定性策略 π 由从每个信息集 $I(H)$到某个问题 $q \in \mathcal{Q}$的映射给出，该问题将为 $q_{|H|+1}$而选择。按照 π 在选择 s^* 时，会得到如下历史 $H_0, \ldots H_T$，表示为

$$q_{t+1} = \pi(I(H_t)), a_{t+1} = f(\pi(I(H_t)), s^*),$$

其中，T 是博弈结束的时间，即 $|S(H_T)| = 1$。博弈结束时，物品选择者的相应效用（提问者的相应成本）为 $u(\pi, s^*) = |H_T|$。

设所有确定性策略的集合为 Π，Δ_Π 是 Π 上的概率单纯形，$\Delta_\mathcal{S}$ 是 $\mathcal{S}$ 上的概率单纯形。那么，提问者和物品选择者分别采用随机策略 $x \in \Delta_\Pi$ 和 $y \in \Delta_\mathcal{S}$ 时的期望效用（稍微滥用一下符号）记为

$$u(x, y) = \mathbb{E}_{\pi \sim x, s^* \sim y}\left[u(\pi, s^*)\right].$$

该零和博弈的纳什均衡由以下双线性鞍点问题的解给出

$$\min_{x \in \Delta_\Pi} \max_{y \in \Delta_\mathcal{S}} u(x, y) = \max_{y \in \Delta_\mathcal{S}} \min_{x \in \Delta_\Pi} u(x, y) \qquad (1)$$

其中等式成立是冯·诺依曼极小极大定理的结果（冯·诺依曼，1928年）。解 (x^*, y^*) 为每位玩家给出最优随机化策略，即每位玩家都在对另一方做出最优反应。特别地，x^* 是分布，其定义于提问者策略上，并对 s^*的最坏情况选择具有稳健性。SLSR、WSLS的扩展式博弈

以及WSLSR也可通过以下一种或两种方式构造：(i) 将每个信息集处可用的行动限制为 $g(S(H))$，或(ii) 调整叶节点收益，使其依赖于 w 和 s^*。

求解扩展式博弈。一般而言，具有完美回忆的双人零和扩展式博弈（包括 SLS 及其变体）的求解时间相对于博弈树的大小呈多项式增长。一些经典方法包括线性规划（Von Stengel，1996）和反事实遗憾（CFR）最小化（Zinke-vich 等人，2007）。请注意，为提问者给出的策略定义采用的是标准式（也称策略式或矩阵式）。这会导致动作空间呈指数级增长，因此在实践中，求解器通常（隐式地）采用与之在策略上等价的行为式或序列式（Von Stengel，1996）。扩展式博弈中的高效博弈求解和策略表示是一个经过充分研究的主题，超出了本文的讨论范围，不过我们会在附录中简要介绍。相反，我们将直接使用现成的求解器（Liu 等人，2024年；Lanctot 等人，2019年）。

5. 思维博弈与子博弈搜索

自然语言问题的空间天然十分庞大。因此，为了便于实际应用，我们聚焦于候选问题集规模为 m 的受限 SLS 变体。尽管如此，即便施加了这一限制，对于较大的假设空间，显式构建并求解完整的 (W)SLSR 游戏树仍不切实际，因为这需要在每个信息集上查询大语言模型来提出问题并提供答案，而每次调用都会产生数秒的延迟。在我们的实验中，为包含 25 个项目的集合构建完整的树，需要花费 5–6 小时来构建 3763 个信息集。对于更大的数据集（约 100 个项目），这一过程很快就会变得不可行。

为应对这一挑战，GoT 采用迭代式、按需构建策略的范式，仅在到达某个信息集时才计算相应策略。受扑克机器人所用子博弈搜索技术的启发（Brown & Sand-holm，2019年；2019年），GoT 会以当前信息集为根构建一个子博弈，并将其截断至固定深度 d。此子博弈的叶节点使用启发式函数进行评估。随后求解这个截断后的子博弈，以获得提问者的局部策略，并据此选择下一个问题。图2对此进行了概述。

1. 深度受限模拟 我们首先探索提问者可能面临的未来结果和问题。在信息集 $I(H_t)$ 处，使用大语言模型生成一组候选问题 $g(S(H_t))$。对于每个候选问题，我们使用大语言模型作为 f，以确定回答 c_i为“是”的物品集合，记为 $Y(S(H_t)\ c_i)$，其中

$$Y(S,q) = \{s \in S | f(q,s) = 1\} \text{ for some } S \subseteq \mathcal{S}.$$

¯以及记作 $Y(S(H_t)\ c_i)$ $=S(H_t)\backslash Y(S(H_t)\ c_i)$ 的补集。这两组集合表示两种提出候选问题 c_i at¯第 t 步时可能出现的结果，其中 $Y(S(H_t)\ c_i)$（分别是 $Y(S(H_t)\ c_i)$）表示与历史 $H'_t($ c_i, 1)（分别是 $H'_t(c_i, 0)$）一致的物品集合。根据这两种结果构建相应的信息集 $I(H'_t(c_i, 1)),I(H'_t(c_i, 0))$。重复此过程以构建 d 步的信息集，最终得到模拟树，如图2第1步所示。请注意，此步骤类似于 UoT框架中的问题模拟与生成步骤；当使用相同的 d 和 m时，其大语言模型调用次数也相近。

2. 转换为扩展式博弈 随后，我们基于模拟树构建一个子博弈。按照图2的步骤2，将该树转换为截断子博弈的扩展式博弈表示。请注意，即使 $I(H_t)$并非从博弈开始处（即 $t \neq 0$）开始，子博弈仍会在每次迭代时先让物品选择者对 $S(H_t)$重新选择一个分布。这似乎赋予了物品选择者更多权力。这样做使我们能够进行安全子博弈搜索（Moravcik 等人，2016年），相关讨论见附录。博弈树 l 中的每个叶节点都被赋予收益$d(l)-1+h(l)$，作为物品选择者的收益（提问者的收益则为其相反数）；其中，$d(l)$表示树中节点 l 的深度，即在步骤t 之后、到达节点 l 之前所提出的问题数量；h 则是一个启发式函数，用于估计提问者还需提出多少个问题才能找到正确物品。在我们的 SLSR 实验中，设定 $h(l) := \log_2(|S(l)|)$，这为剩余回合数提供了一个乐观估计。

3. 扩展式博弈求解 为求解子博弈，我们使用 LiteEFG（Liu 等人，2024年）中 CFR最小化 的实现，以获得近似纳什均衡。

4. 选择下一个问题 为了选择在步骤 $t+1$ 中要提出的问题，我们使用纳什均衡中的提问者策略，从 $g(S(H_t))$ 中采样一个问题 q_{t+1}。如果 $f(q_{t+1},\ s^*) = 0$，则转移到信息集 $I(H'_t(q_{t+1},\ 0))$；否则转移到 $I(H'_t(q_{t+1},\ 1))$。

该过程会重复进行，直到到达信息集 $I(H_n)$，其中$|S(H_n)| = \{s^*\}$。请注意，GoT 的大部分计算成本都产生于步骤 1，这是由于大语言模型响应延迟所致。其余步骤的贡献可以忽略不计。

定理 5.1。 对于仅依赖于 $S(H_t)$ 的价值估计，*GoT* 是安全的。

6. 实验与结果

我们关注以下问题：与其他方法相比，*GoT* 是否能在所有设置下提升最坏情况性能？我们还考察了 (i) 问题质量、(ii)模拟深度 d 对其性能的影响，以及 (iii)平均情况与最坏情况之间的权衡。

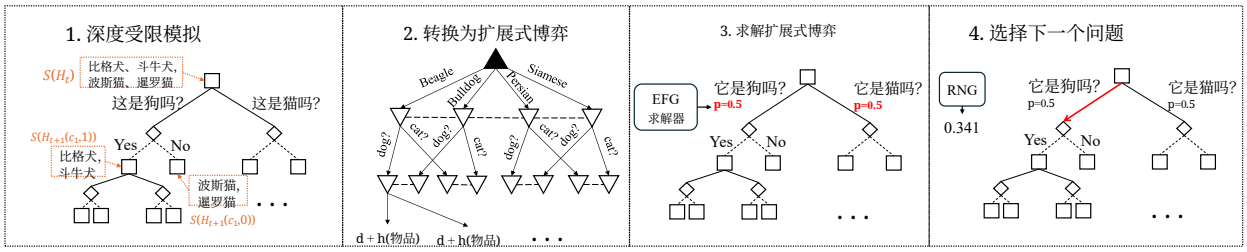

图2. **Go概览** **T.** 为选择时间步 t 的问题，我们 (1) 执行深度受限模拟 (2) 构建一个子博 e
(3) 求解该子博以得到局部策略，该策略 (4) 为问题的选择提供依据。步骤 1 至 3 制定策略，而步骤 4 进行博 e.

6.1. 实验设置

6.1.1. 数据集与设置

二十问（20Q）：每个数据集都由一组不同的物品构成。先前的研究（Bertolazzi 等人，2023年；Zhang 等，2024b）使用了 THINGS（Hebart 等人，2019年）和名人等数据集，我们发现这些数据集对于本研究而言规模过大。因此，我们构建了若干数据集：(i) Common：包含 136 个项目，基于 UoT（Hu 等，2024）收集的数据集构建，并通过添加更多项目进行了扩充。(ii) Breeds：我们收集的 25 种常见猫和狗品种。(iii) S128：我们收集的 128 个项目，包括 2 种武器、6 位科学家、24 道菜肴和 96 种动物。除二十问设置外，我们还展示了另外两个更具实用性的领域的结果。医学诊断（MD）：医生通过询问患者的症状来诊断患者。我们使用 DXBench（DX）（Chen 等人，2025年），并从该数据集中的 461 种不同疾病中随机选取 100 种作为假设空间。故障排查（TS）：汽车修理工尝试通过询问客户的症状，帮助客户诊断其汽车故障。我们使用 FloDial（Raghu 等人，2021年），其中包含基于 12 种不同故障排查流程图构建的 2,738 段对话。经过数据预处理，我们得到 59 种不同的汽车故障，并将其用作假设空间。详情见附录。

6.1.2. 基线方法

我们主要与 UoT（Hu 等人，2024年）进行比较。对于 UoT 和 GoT，除非另有说明，我们均设定 $d = m = 3$。此外，我们还考察直接提示（DP），即让大语言模型生成下一个问题；以及直接选择（DC），即让大语言模型从一组候选问题中进行选择。每种策略都会在每个数据集中与所有可能的 s^* 进行对局，并取所有物品中的最差性能：$L_{worst} = \max_{s \in \mathcal{S}} |H^s|$，其中 H^s 是当 s为所选物品时，某种方法的交互历史。

6.1.3. 模型与提示

我们主要使用 GPT 4.1（OpenAI，2024年）（`gpt-4.1-2025-04-14` 检查点）和通义千问 2.5 72B 进行了实验。Instruct（Yang 等人，2024年）。在 20Q 中，我们使用了两种不同的提示词来抽样问题。我们采用了（均衡提示词，该提示词改编自（Hu 等人，2024年），明确要求大语言模型提出问题，尽可能将物品平均分组。使用这一提示词时，我们注意到会出现不自然的问题（例如：这个物品的名称是否以 A 到 M 之间的字母开头？）。这类问题不适用于真实场景，因为受访者可能难以作答。因此，我们设计了自然提示词，要求大语言模型避免提出此类问题。下文报告的所有结果均使用自然提示词。两种提示词的对比见附录。在医学诊断和故障排查设置中没有观察到这一问题，可能是因为提示词中明确要求大语言模型扮演医生或机械师。

6.1.4. 考虑随机性

(1) 当使用大语言模型作为 g 来生成问题时，候选问题集合 $\mathcal{Q}(S(H_t))$可能会因运行而异。为公平比较相近的方法，我们首先使用 GoT 进行博弈，并将问题缓存到模拟树中。缓存的问题会供 DC 和 UoT 重复使用。**(2)** GoT 设计的提问者策略具有非确定性。为了获得每个 s^*的性能，我们使用相同策略进行十次博弈，并取平均值。

6.2. 无权重变体

如表格 1所示，在最坏情况交互长度方面，GoT 的表现始终优于 UoT。基于提示的直接提示偶尔会优于 UoT，这表明虽然 UoT 能够提升平均情况性能，但可能会以牺牲最坏情况性能为代价。

由定理 3.7可知，如果大语言模型始终能够提出将 $S(H)$ 完美均分为两半的问题，那么只要始终提出这样的问题，我们便能达到最优性能。然而，我们看到实际情况并非如此：所有方法相较于 $\log_2(|\mathcal{S}|)$ 的理论最优性能大约落后 2 到 3 轮交互。这表明，在实践中，尤其是在医学诊断或故障排查等现实场景中，可能很难提出一个足够自然且能实现均匀划分的问题。在这种情况下，考虑多个

方法	20Q Common	20Q S128	20Q Breeds	MD DX	TS FloDial
GPT 4.1					
GoT	10.2	11.8	7.4	12.2	7.9
UoT	11	13	9	13	9
DP	13.8	16.2	7.8	16.8	12.7
DC	12.9	14.6	9.3	16.2	11.6
Qwen 2.5 72B Instruct					
GoT	10.0	10.8	6.6	10.5	7.5
UoT	11	12	8	12	9
DP	12.7	19.2	8.0	12.4	10.7
DC	12.7	17.9	7.8	13.3	10.5

表1. **各种设置下各方法的最坏情况交互长度**。每种设置下性能最佳的结果均已加下划线。

方法	20Q Common	20Q Breeds	MD DX	TS FloDial
GPT 4.1				
GoT	152.1	23.2	78.3	61.4
UoT	227.4	32.1	110.0	81.0
DP	224.0	36.9	116.0	90.1
DC	199.2	41.1	126.4	97.4
Qwen 2.5 72B Instruct				
GoT	151.9	32.3	73.6	62.3
UoT	228.5	47.0	99.0	74.0
DP	235.7	47.8	114.3	80.0
DC	225.1	45.7	86.1	113.6

表2. **加权变体中各种方法的最坏情况性能，以**最大值$_{s\in S}(w(s)\cdot|H_s|)$衡量。每种设置下的最佳性能均以下划线标出。

候选问题并采用随机选择策略（正如 GoT 所做的那样），可以防范异常情况，即在最坏情况下识别某一特定疾病或故障需要过长的对话。

6.3. 加权变体

我们进一步考察各设置在加权变体下的性能。现在，GoT 将该设置建模为 WSLSR 博弈，并将子博弈中使用的启发式函数替换为 $h(l) = \max^{s\in S(l)} w(s) \cdot (d(l) + \log_2(|S(l)|))$，以反映 $w(s^*) \cdot |H|$ 的收益变化。对于所有方法，我们都会修改问题采样提示，加入项目权重和收益计算，以便向大语言模型提供相关信息。对于医学诊断和故障排查，我们使用 GPT-5.2 Thinking（启用扩展推理）为每个项目标注一个 1 到 10 的整数权重，以反映相应疾病或故障的严重程度和紧迫性，其中 1 表示最轻微，10 表示最严重。由于 20Q 中没有类似的概念，我们改为从参数为 $\mu = 0$ 和 $\sigma = 1$ 的对数正态分布中为该设置采样项目权重。附录中提供了项目权重分布的可视化结果。

结果见表格2。在所有设置下，GoT 的表现均优于 UoT，提升幅度为 15% 至 40%。另一个观察结果是，在加权变体中，UoT 相较于 DP 和 DC 等纯提示方法的提升不那么明显。这可能是因为 UoT 的方法是在不考虑项目权重的情况下最大化信息增益，这可能导致它优先选择那些（假设分布均匀时）信息增益较高的问题，而非更有效地缩小高权重项目范围的问题，从而使 WSLSR 中的策略不够理想。

我们进一步研究了增加模拟深度 d 对 UoT 和 GoT 的影响；结果如图3所示。总体而言，随着 d 增加，GoT 的性能有所提升，但其

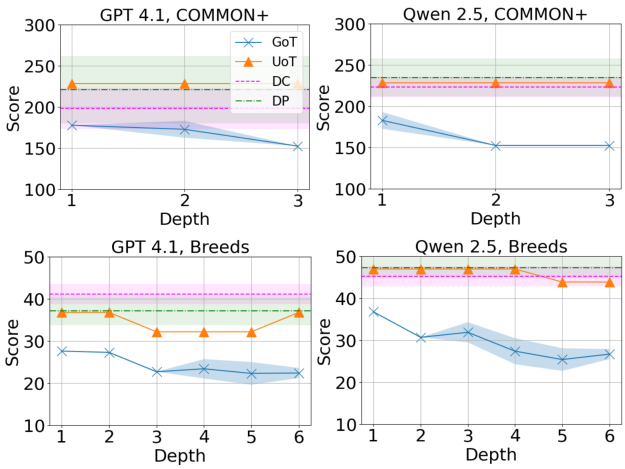

图3. **加权变体中各种方法在 Common 和 Breeds 上的最坏情况性能**。横轴为 d，纵轴为物品选择者的收益 $w_i \cdot |H|$。

收益逐渐趋于平稳，因为它接近我们推测的、针对固定一组大语言模型采样的问题的最优策略。相比之下，即使前瞻更深，UoT 的最坏情况性能也几乎没有改善，这表明优化信息增益并不一定能带来更好的最坏情况保证。

我们进行了一项简短的消融实验，以考察问题质量对 WSLSR 的影响。具体而言，我们在 Breeds 数据集上构造了一个人为倾斜的权重分布：除一个项目外，其余项目的权重均设为 1，而该项目的权重设为 100。在这种设置下，最优的首个查询是测试目标是否为最高权重物品。我们通过手动将相应问题加入初始候选集来验证这一点。如图 4所示，GoT 始终能够达到这一最优策略。相比之下，若不包含该问题，性能会大幅下降，这表明 GoT 的效果可能对候选问题的质量较为敏感。

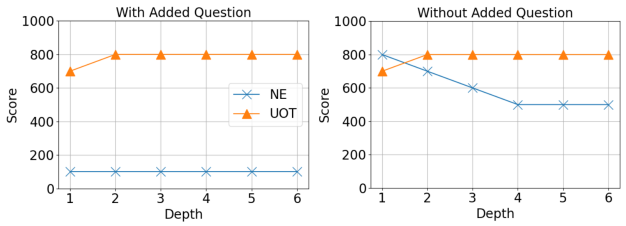

图*4.* **品种权重经过人工偏斜处理后的加权变体性能**。 左侧图表展示了包含附加问题时的性能。

6.4. 无权重变体的平均性能

虽然我们主要考察最坏情况性能，但此处也分析了多项先验分布下 DX 数据集上的平均对话长度 P^o，计算方式为 $L_{avg} = \sum_{s\in\mathcal{S}} P^o(s) \cdot |H^s|$。这表示在潜在疾病分布遵循 P^o的情况下，方法的期望对话长度。我们通过从狄利克雷分布 P^o 中抽样，获得 50 个 $Dir(kC)$，其中 $C \in \mathbb{Z}_{100}$ 是数据集中每种疾病的出现次数，且 $k \in \mathbb{R}_{+}$。 L_{avg} 图 5 展示了各方法的结果，并同时给出 L_{worst}，以说明平均性能与最坏情况性能之间的权衡。我们还考虑了这样一种设置：项目并非由对手选择，而是遵循一个公开已知的 P^o，从而使提问者能够针对该先验分布进行优化。我们将这种策略称为最佳响应（BR），它构成性能的理论下界。详情见附录。

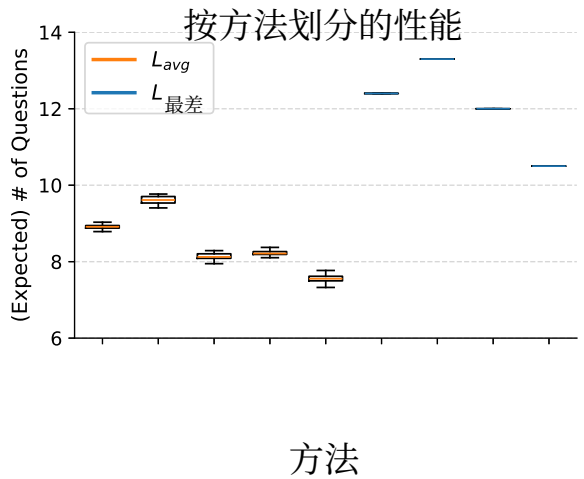

图*5.* **DX 上的平均性能和最坏情况性能**。 所用模型为 Qwen 2.5 72B Instruct， k 设为 1。请注意，平均性能的方差源于 P^o样本不同，而非方法本身的随机性。

从图5中，我们强调两个关键观察结果。（1）UoT 和 GoT 的平均情况性能相近：二者差距很小，相较于它们相对于 DP/DC 的性能提升以及 BR 的相对于二者的改进幅度而言。（2）GoT 在最坏情况下相较于 UoT 的优势显著：其幅度与 UoT 相较于 DP 的平均情况性能提升相当，而这正是 UoT 明确针对的指标。总体而言，与 UoT 相比，GoT 在平均情况性能损失很小的同时，显著改善了最坏情况性能。

我们进一步评估 UoT 和 GoT 在两类多项式先验下的 L_{avg}，这些先验分别对每种方法不利。具体而言，对于每种方法，我们考虑 $X_{method} \sim Dir(k\alpha)$，其中 $\alpha \in \mathbb{Z}_{100}$ 满足$\forall i \neq 0,\ \alpha_i = 1$。我们将 α_0 设为使分布集中在单个项目附近。特别地，我们令索引 0 对应该方法下导致最坏对话长度的项目。我们从每种配置中抽取 100 个样本，并比较 UoT 和 GoT 的 L_{avg}。

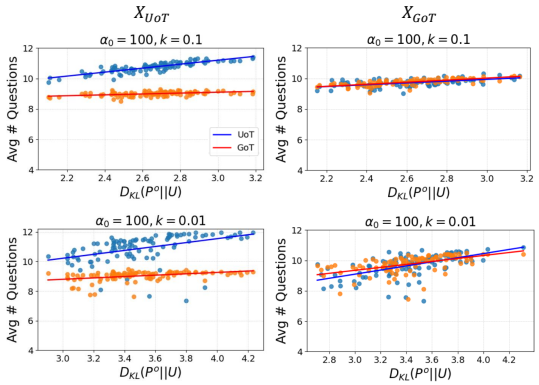

图*6.* **不同分布族下的平均性能** 左列对应对 UoT 不利的先验分布，右列对应对 GoT 不利的先验分布。X 轴表示先验分布与均匀分布之间的 KL 散度。

如图6所示，对于从 X_{UoT}中抽取的先验分布，两种方法表现出显著差异：UoT 的性能下降幅度远大于 GoT。相比之下，对于从 X_{GoT}中抽取的先验分布，两者的平均性能没有明显差距。综合来看，这些结果表明，与 UoT 相比，GoT 对不利先验更具鲁棒性。

7. 局限性与未来工作

我们将候选问题限定为只能有严格的二元答案。如何将其扩展到包含开放式回答的问题，则留待未来工作探讨。

8. 结论

我们形式化定义了 SLS 及其变体，并提出了 GoT，这是一种简单而有效的 SLSR 求解方法。我们表明，与先验方法相比，GoT 能够提升最坏情况性能。与优化平均性能相比，我们这种针对最坏情况性能进行优化的方法往往更有价值。

9. 影响声明

本文的研究旨在推动机器学习领域的发展。我们的工作可能带来许多社会影响，但我们认为没有哪一项需要在此特别强调。

参考文献

Bertolazzi, L., Mazzaccara, D., Merlo, F., 和 Bernardi, R. ChatGPT 的信息寻求策略：来自二十问游戏的见解。载于第 *16* 届国际自然语言生成会议论文集，第 153–162 页，2023年。

Bishop, J., Burgess, J., Ramos, C., Driggs, J. B., Williams, T., Tossell, C. C., Phillips, E., Shaw, T. H., 和 Visser, E. J. d. Chaopt：一个用于评估人机自主团队协作的测试平台，使用电子游戏 Overcooked! 2。载于*2020* 年系统与信息工程设计研讨会（*SIEDS*），第 1–6 页，2020年。

Brown, N. 和 Sandholm, T. 单挑无限注扑克中的超人类 AI：Libratus 击败顶尖职业选手。《科学》，359(6374)：418–424，2018年。

Brown, N. 和 Sandholm, T. 面向多人扑克的超人类人工智能。*Science*，365(6456):885–890，2019年。

Burch, N.、Johanson, M. 和 Bowling, M. 使用分解法解决不完美信息博弈。载于第二十八届*AAAI*人工智能大会论文集，第602–608页，2014年。

Capraro, V.、Di Paolo, R.、Perc, M. 和 Pizziol, V. 人工智能时代的基于语言的博弈论。英国皇家学会界面期刊，21(212)，2024年。

Chen, J.、Gui, C.、Gao, A.、Ji, K.、Wang, X.、Wan, X. 和 Wang, B. CoD：利用诊断链构建可解释的医疗智能体。载于计算语言学协会论文集：*ACL 2025*，第14345–14368页，2025年。

Choudhury, D.、Williamson, S.、Goli´nski, A.、Miao, N.、Smith, F. B.、Kirchhof, M.、Zhang, Y. 和 Rainforth, T. Bed-llm：使用大语言模型和贝叶斯 experimentaldesign 进行智能信息收集。*arXiv* 预印本 *arXiv:2508.21184*，2025年。

Farrell, J. 和 Rabin, M. 廉价磋商。《经济展望杂志》，10(3):103–118，1996年。

Gemp, I., Bachrach, Y., Lanctot, M., Patel, R., Dasagi, V., Marris, L., Piliouras, G., Liu, S., and Tuyls, K. 状态作为字符串作为策略：使用博弈论求解器引导语言模型。*arXiv* 预印本 *arXiv:2402.01704*，2024年。

Gintis, H., Smith, E. A., and Bowles, S. 代价信号与合作。《理论生物学杂志》，213(1): 103–119，2001年。Goldman, S. A. and Kearns, M. J. 关于教学复杂度。《计算机与系统科学杂志》，50 (1):20–31，1995年。Grand, G., Pepe, V., Andreas, J., and Tenenbaum, J. B. 言多必失：通过语言信息引导的程序采样，在战舰棋中提问。*arXiv* 预印本 *arXiv:2402.19471*，2024年。

Handa, K.、Gal, Y.、Pavlick, E.、Goodman, N.、Andreas, J.、Tamkin, A. 和 Li, B. Z. 使用语言模型进行贝叶斯偏好 elicitation。*arXiv* 预印本 *arXiv:2403.05534*，2024年。

Hebart, M. N.、Dickter, A. H.、Kidder, A.、Kwok, W. Y.、Corriveau, A.、Van Wicklin, C. 和 Baker, C. I. THINGS：包含1,854个物体概念和超过26,000幅自然场景物体图像的数据库。《公共科学图书馆·综合》，14(10):e0223792，2019年。

Houser, D. 和 Xiao, E. 使用协调博弈对自然语言消息进行分类。实验经济学，14:1–14，2011年。

Hu, Z., Liu, C., Feng, X., Zhao, Y., Ng, S.-K., Luu, A. T., He, J., Koh, P. W. 和 Hooi, B. 思维的不确定性：不确定性感知规划增强大语言模型中的信息搜寻。载于第*38*届神经信息处理系统大会论文集，NeurIPS ’24，2024年。

Jain, B. P. 临床病理讨论会（cpcs）中的诊断为何不是概率性的：正方。诊断，3(3): 95–97，2016年。

Kempinski, B., Gemp, I., Larson, K., Lanctot, M., Bachrach, Y. 和 Kachman, T. 思维博弈：利用大型语言模型在博弈论领域进行迭代推理。载于第*24*届自主智能体与多智能体系统国际会议论文集，第1088–1097页，2025年。

Kovarık, V., Seitz, D. 和 Lisy, V. 不完全信息博弈中深度受限求解的价值函数。载于 *AAAI* 博弈强化学习研讨会，第 132 卷，第 133–138 页，2021年。

Kram´ar, J., Eccles, T., Gemp, I. M., Tacchetti, A., McKee, K. R., Malinowski, M., Graepel, T. 和 Bachrach, Y. 外交棋桌游中的人工智能方法的谈判与诚实。《自然·通讯》，13，2022年。

Kuhn, L., Gal, Y. 和 Farquhar, S. Clam：针对生成式语言模型处理歧义问题的选择性澄清。*arXiv* 预印本 *arXiv:2212.07769*，2022年。

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H. 和 Stoica, I. 使用 PagedAttention 高效管理大语言模型服务的内存。载于第 *29* 届操作系统原理研讨会论文集，第 611–626 页，2023年。

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., P´erolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S. 等。OpenSpiel：一个用于博弈中强化学习的框架。*arXiv* 预印本 *arXiv:1908.09453*，2019年。Lewis, D. K. 《惯例：一项哲学研究》。Wiley-Blackwell，美国马萨诸塞州剑桥，1969年。

Li, B. Z., Tamkin, A., Goodman, N. 和 Andreas, J. 利用语言模型引出人类偏好。*arXiv* 预印本 *arXiv:2310.11589*，2023年。

Li, B. Z., Kim, B. 和 Wang, Z. QuestBench：大语言模型能否在推理任务中提出正确的问题来获取信息？*arXiv* 预印本 *arXiv:2503.22674*，2025 年。

Li, S. S., Balachandran, V., Feng, S., Ilgen, J. S., Pierson, E., Koh, P. W. 和 Tsvetkov, Y. MediQ：提问型大语言模型及用于可靠交互式临床推理的基准测试。载于神经信息处理系统进展，第37卷，第28858–28888页，2024年。

Liu, M., Farina, G. 和 Ozdaglar, A. LiteEFG：用于求解扩展式博弈的高效 Python 库。*arXiv* 预印本 *arXiv:2407.20351*，2024年。

Moravcik, M., Schmid, M., Ha, K., Hladik, M. 和 Gaukrodger, S. 在大型不完全信息博弈中改进子博弈。载于人工智能 *AAAI* 会议论文集，第 30 卷，2016年。

OpenAI。GPT-4.1，2024年。

Piriyakulkij, W. T., Kuleshov, V. 和 Ellis, K. 使用语言模型和概率推理进行主动偏好推断。*arXiv* 预印本 *arXiv:2312.12009*，2023年。

Raghu, D., Agarwal, S., Joshi, S. 和 Mausam。端到端学习流程图落地的任务型对话。载于*2021*年自然语言处理实证方法会议论文集，第 4348–4366 页，2021年。

Ri, H., Kang, X., Khalid, M. N. A., and Iida, H. 少数群体与多数群体行为的动态变化：黑手党游戏案例研究。*Information*, 13(3):134, 2022年。

Siegler, R. S. 二十问游戏作为一种问题解决形式。*ChildDevelopment*, 48(2):395–403, 1977. Spaan, M. T. 部分可观测马尔可夫决策过程。载于强化学习：前沿进展, pp. 387–414. 施普林格, 2012. 冯·诺依曼, J. 论博弈论。数学年鉴, 100(1):295–320, 1928年。Von Stengel, B. 行为策略的高效计算。《博弈与经济行为》，14(2):220–246, 1996.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A. 和 Zhou, D. 自一致性提升了语言模型中的思维链推理。*arXiv* 预印本 *arXiv:2203.11171*，2022年。

Xu, J., Wang, Y., Tang, D., Duan, N., Yang, P., Zeng, Q., Zhou, M., and Sun, X. 基于知识的问答中的澄清问题提问。载于《*2019*年自然语言处理经验方法会议暨第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集》，第 1618–1629 页，2019年。Yang, Q. A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y.-C., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z., Quan, S., and Wang, Z. Qwen2.5 技术报告。*arXiv* 预印本 *arXiv:2412.15115*，2024年。

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., 和 Narasimhan, K. 思维树：利用大型语言模型进行审慎的问题解决。载于 神经信息处理系统进展，第 36 卷，第 11809–11822 页，2023年。

Zhang, X., Deng, Y., Ren, Z., Ng, S.-K., 和 Chua, T.-S. 先询问再规划：用于现实世界规划的主动语言智能体。载于 计算语言学协会论文集：*EMNLP 2024*，第 10836–10863 页，2024a。

Zhang, Y., Lu, J., 和 Jaitly, N. 通过二十问游戏探究大语言模型的多轮规划能力。载于 计算语言学协会第 *62* 届年会论文集（第 *1* 卷：长篇论文），第 1495–1516 页，2024b。

Zinkevich, M.、Johanson, M.、Bowling, M. 和 Piccione, C. 不完全信息博弈中的遗憾最小化。载于 神经信息处理系统进展，第 20 卷，2007 年。

A. 策略表示

在具有完美回忆的不完全信息博弈中，有多种方式可以表示策略（在收益等价意义下）。出于计算方面的原因，通常使用序列式。下面简要介绍其他表示方式。

1. 标准式/策略式。这就是论文中所介绍的内容。确定性策略的集合是每个信息集上的行动的笛卡尔积 $\Pi = \prod_{I \in \mathcal{I}} \mathcal{A}(I)$。因此，每个策略 $\pi \in \Pi$ 都准确地告诉我们，在每个可能遇到的信息集上应采取什么行动。随机化策略是确定性策略上的分布，即 Δ_Π。Π的大小关于信息集的数量呈指数增长。

2. 约化标准式。约化标准式通过将策略上等价的标准式策略归为一组，减少标准式策略的数量。有关更完整的讨论，请参阅任何一本博弈论教材，或 Von Stengel（1996）的相关著作。设 I 是一个信息集，且 α、$\alpha' \in \mathcal{A}(I)$ 是 I中的不同行动。设I' 是一个先于 α' 的信息集。那么，在 I 中选择 α 的标准式策略，其收益（无论对手采用何种策略）都与在 I' 中采取何种行动无关，因为要进入 I' 就必须选择 α'。因此，两个策略 π、π' 如果在 I 中选择 α，但在 I' 中采取的行动不同，则它们在策略上是等价的。通过构造这些等价类所得到的最小策略集合，就构成了约化标准式策略集合，其规模可能显著小于标准式策略集合。与之前一样，策略也可以在约化标准式中进行随机化。不幸的是，约化标准式策略的规模相对于信息集的数量仍可能呈指数增长。

3. 行为式。行为策略是在每个信息集上为行动分配一个分布（不一定是确定性的），因此其规模与所有信息集中的行动总数呈线性关系，而这一总数不超过博弈树的规模。利用*Kuhn*定理可以证明，在完美回忆条件下，行为策略集合与（约化）标准式策略在策略上等价。不幸的是，尽管行为策略直观且简洁，但在行为式中进行优化很困难，因为收益关于策略表示是非线性的。

4. 序列式。序列式策略缓解了这样的问题：孤立于机会和其他玩家，为特定序列 σ（在博弈具有完美回忆时，本质上就是行动）分配概率。在每个信息集 I 处，序列式与行为式本质上相同（一个概率单纯形），但它们会根据信息集的父序列 $\sigma(I)$ 所给定的概率进行归一化；该父序列是到达 I 之前采取的最后一个行动（序列）——完美回忆保证它是唯一的。因此，序列式与行为策略的规模相同（但多出一个“空序列” ϕ，其值设为 1.0，并且它是所有初始信息集的父节点）。序列式策略空间有时也称为树丛

$$\mathcal{X} = \left\{ x \in \mathbb{R}^N_+ \middle| x[\phi] = 1; \sum_{\sigma = Ia} x[\sigma] = x[\sigma(I)] \; \forall \, I \in \mathcal{I} \right\}$$

其中，$N = 1 + \sum_{I \in \mathcal{I}} |I|$ 和 $\mathcal{I}$ 是信息集的集合，Ia 指的是从信息集 I开始、以行动 i 结束的序列，ϕ 是空序列。 Π 的顶点对应（约化）标准式策略。使用序列式，我们可以将零和博弈的纳什均衡写成双线性鞍点问题

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} x^T M y$$

其中 M 是序列式收益矩阵。注意，目标函数关于 x 或 y均为线性函数。可以使用多种方法高效求得该鞍点，包括一阶方法，例如反事实遗憾最小化（Zinkevich 等人，2007）、线性规划或其他一阶方法（例如镜像近端法）。我们使用的库（Liu 等人，2024年）采用反事实遗憾最小化。

B. 详细推导

B.1. 定理3.6的证明

该问题等价于以下集合交问题。

集合交问题。设 $\mathcal{S}$ 是一个有限集，$\mathcal{K} \subseteq 2^{\mathcal{S}}$ 是一个集合族。给定某个固定的 $s \in \mathcal{S}$，我们要判断是否存在一个集合 $\mathcal{J} \subseteq \mathcal{K}$，使得 $|\mathcal{J}| \leq k$ 且 $\bigcap_{J \in \mathcal{J}} J = \{s^*\}$；也就是说，能否找到 $\mathcal{J} \subseteq \mathcal{K}$ 的一个小子集，使其交集恰好包含 s^*。可以看出，$\mathcal{K}$ 包含了可向 s^* 提出的问题集合。我们可以安全地假设 $\forall K \in \mathcal{K}$，$s^* \in K$。

集合覆盖。设 $\mathcal{A}$ 为任意集合，且 $\mathcal{B} \subseteq 2^{\mathcal{A}}$。该判定问题是：能否找到某个大小为 k 或更小的集合 $B^* \subseteq \mathcal{B}$，使得 $\bigcup B^* = \mathcal{A}$。

NP 完全性。我们证明集合覆盖问题可归约到集合交集问题。

引理 B.1. 集合覆盖问题可归约为集合交问题。

证明。 我们定义 $\mathcal{S} = \mathcal{A} \sqcup s^*$，其中包含某个虚拟元素 s^*，并令 $\mathcal{K}$ 包含 $\mathcal{B}$中各元素对应的补集子集，再加入 s^*， $\mathcal{K} = \{K_i = B_i \sqcup \{s^*\} | B_i \in \mathcal{B}\}$。取该集合交问题的一个解 $K^* \subseteq \mathcal{K}$ 及其对应的子集 $B^* = \{B_i | K_i \in K^*\}$。由于$\bigcap_{i:\ K_i \in K^*} K_i = \{s^*\}$，因此 $\bigcup_{i:\ K_i \in K^*} K_i = \mathcal{S} \backslash \{s^*\}$。于是$\bigcup_{i:\ B_i \in B^*} B_i = \bigcup_{i:\ K_i \in K^*} K_i = \mathcal{S} \backslash \{s^*\} = \mathcal{A}$。 □

为完整起见，我们也展示另一方向的归约。

引理 B.2. 集合交问题可归约为集合覆盖问题。

证明。 我们定义 $\mathcal{A} = \mathcal{S} \setminus \{s^*\}$ 和 $\mathcal{B}$，其中包含的子集分别是 $\mathcal{K}$、 $\mathcal{B} = \{B_i = K_i \mid K_i \in \mathcal{K}\}$中各元素的补集。取 $\mathcal{A}$ 的一个集合覆盖 $B^* \subseteq \mathcal{B}$，以及与之对应的 $\mathcal{K}$ 的子集 $K^* = \{K_i | B_i \in B^*\}$。由于$B^*$ 覆盖了 $\mathcal{A}$，即 $\bigcup B^* = \mathcal{A}$，因此有 $\bigcap_{B_i \in B^*} B_i = \phi$，从而得到 K^*: $\bigcap_{i:\ K_i \in K^*} K_i = \{s^*\} \cup \bigcap_{B_i \in B^*} B_i = \{s^*\}$。 □

最后，集合交集问题的任何解也可以在多项式时间内检验。

注记 B.3。一个非常相似的问题，称为教学维度，由 Goldman & Kearns （1995年）研究，并进而与学习理论中的其他相关概念（如 VC 维度）密切相关。

B.2. 定理3.7的证明

显然，均分策略恰好使用 k 个问题。此外，每个问题恰好产生一比特信息，因此至少需要 k 个问题。这表明均分策略是一个纳什均衡（亦即它是一个极小极大解）。要证明均匀策略是一个极大极小解，只需运用对称性。设 $y^* \in \Delta_n$ 是回答者的某个（可能非均匀的）纳什均衡。我们知道，零和矩阵博弈的极大极小解集合构成一个非空凸集。取 y^*的所有置换。由对称性，它们必定也都是纳什均衡。设所有置换的平均为 y^*，再由对称性，它等于均匀分布。但根据前述极大极小解集合的凸性，它也是回答者的一个纳什均衡。

B.3. 定理5.1 的讨论

GoT 的第 2 步是扩展子博弈，使物品选择者能够在 s^* 上 “ 重新选择 ” 一个新的分布。乍看之下，相比坚持物品选择者在博弈开始时选定的原始分布，这样做似乎赋予了其远超原始 SLSR博弈的权力，因为它可以根据实际对局中提出的问题“切换项目”。。事实上，这是 最大边际求解的一个简单实现，是在一般不完美信息的零和安全子博弈搜索（Moravcik 等人，2016年）中执行的关键步骤。在这些博弈中，子博弈的最优策略可能取决于原始博弈中其他未到达的分支，因此我们无法孤立地为该子博弈求解策略。执行这种安全子博弈求解，可以为提问者的子博弈策略相对于原始博弈中的某个蓝图策略或状态价值估计提供性能保证。相反，如果不这样做，提问者的子博弈策略将不会享有任何此类保证，性能可能会任意糟糕。在实践中，在我们的初步研究中，我们发现其性能优于不安全的子博弈搜索变体。

我们简要概述子博弈搜索和最大边际求解。这里省略更细节的内容和数学表述。主要目的是说明 GoT 的实现方式与最大边际求解相同，而这也意味着其具有安全性。

安全子博弈搜索 子博弈搜索（也称为子博弈求解或持续求解）是一类方法，用于以有原则的方式在不完全信息扩展式博弈中进行深度受限搜索。子博弈搜索的核心思想是，玩家先采用蓝图策略，直到进入一个子博弈（不完全信息子博弈是一个树森林，它在后代关系以及属于任意玩家的增广信息集这两方面都具有封闭性，（Burch 等人，2014））。蓝图策略通常是完整博弈的粗略抽象版本的解。进入子博弈后，玩家会针对所到达的子博弈（且仅针对该子博弈）求解，以寻求更优解。这类似于完全信息情形：由于求解原始完整博弈难以处理，因此只在实际对局中到达的状态下搜索（或细化）策略。

安全子博弈求解算法是一种保证性能不劣于蓝图策略的算法。朴素子博弈求解试图通过构造一个装置博弈来求解该子博弈：该装置博弈以一个机会节点开始，该节点根据（在蓝图以及求解蓝图时的对手策略下）到达这些状态的概率，通向子博弈中的所有初始状态。然而，这种方法忽略了一个事实：仅重新求解实际到达的子博弈可能会诱使对手改变其行动，从而使这些机会概率变得不正确，进而导致性能劣于蓝图。

最大边际求解 最大边际求解（Moravcik 等人，2016年）是以更严谨的方式进行子博弈求解的一种方法。其思路是扩展装置博弈，使得对手首先选择自己想从哪个

（对手的）信息集开始，随后由机会节点根据主玩家的蓝图策略，强制执行到达这些状态（属于对手信息集）的概率。让对手选择想从哪个信息集开始子博弈（注意，如果对手在子博弈开始时不行动，我们可能需要添加虚拟信息集），可以确保主玩家针对最小边际进行优化。此处，对手每个信息集的边际，是指该信息集在蓝图策略下的价值与精化策略下的价值之差。如果主玩家过度优化对手某个特定头部信息集的价值，却以其他信息集表现不佳为代价，那么对手就会选择那些信息集，从而导致不安全求解。通过允许对手选择初始信息集，最大边际求解明确避免了这一问题，确保精化后所有信息集（对主玩家而言）都得到同等程度的改善。

在实践中，我们实际上并不知道对手采取最佳响应时子博弈初始状态的价值（除非将整个子博弈展开到其叶节点），因此必须借助近似价值函数。值得注意的是，何谓良好的价值函数并不显而易见，因为状态的价值取决于双方玩家的行动，且还会受到不完全信息的影响。参见Kovarık 等人(2021年，了解更全面的讨论。

将 GoT 用于最大边际求解首先要注意，SLSR 是一种相对简单的扩展式博弈，因为回答者只在开局时行动一次，之后便不再进行任何交互。事实上，我们甚至可以让回答者拥有完全信息。此外，适当的子博弈（即非整个博弈）仅由历史 H构成，或者等价地说，是从 $I(H)$开始的子博弈，其中包含所有提出的问题，但不包含所选的 s^* 。请注意，每个适当的子博弈中只有提问者会采取行动。由于回答者拥有完全信息，子博弈中回答者的“头部信息集” $I(H)$ 仅对应于该历史 H 的初始状态。请注意，我们实际上无需显式构造这些头部信息集：由于回答者拥有完整信息，这些都是“虚设”信息集，本质上只有一个行动，并会引向提问者首个信息集中的相应状态。这表明，最大边际求解所用装置博弈的结构与我们提出的结构完全相同。图7展示了这一构造的示例。

在 maxmargin 中，子博弈的每个头部信息集下的收益，都会减去蓝图（或函数近似）对应的收益。这确保优化的是边际，而不是绝对收益。值得庆幸的是，我们的情况无需进行这种平移，因为每个状态的近似值都是 $\log_2(|S(H)|)$，即剩余物品数量的对数。对于每个信息集 $I(H)$ 中的状态，该值都相同，因此平移值在整个子博弈中也相同；就求解子博弈而言，这不会影响精化策略。

这一论证同样适用于我们为 WSLSR 选择的价值函数 $h(l) = \max_{s \in S(l)} w(s) \cdot (d(l) + \log_2(|S(l)|)$，该函数仅取决于剩余物品集合 $S(H)$。然而，如果我们选择的价值函数还取决于回答者对 s^* 的具体选择，那么这一论证就不再成立，我们必须相应地对收益进行这些平移。

备注 B.4. 理论上，最大边际求解仅在进入子博弈时执行求解（例如，在提出 $d = 3$ 个问题之后）。实际上，GoT 会在实际对局过程中遇到的每个信息集上执行求解。

C. 实现细节

C.1. 通用实现

方法超参数 由于 UoT 与 GoT 使用相似的模拟流程，因此在每项实验中，我们为这两种方法采用相同的模拟深度 d 和候选问题数量 m，尽管这些数值可能因实验而异。对于直接比较，我们同样提供一组 m 候选问题，供其选择。

剩余两个物品时的问题生成在 (W)SLSR 游戏中当前历史 H 满足 $|S(H)| = 2$的情况下，我们会确定性地为每个剩余物品构造两个形式为“x 是正确物品吗？”的候选问题，而不是从大语言模型中抽样问题。虽然这并非严格必需，但实现这一做法主要是为了提高效率。根据定理 3.8，任何满足该假设的问题都能成功区分这两个剩余物品，从而结束博弈。因此，在这类状态下无需查询大语言模型，进而降低构建模拟树的延迟。

问题缓存与复用 由于我们在实验中按需构建 SLSR 博弈，因此需要确保各方法所进行的博弈实例保持一致。在每次 GoT 执行过程中，都会缓存大语言模型为任意已探索历史 H 生成的候选问题集合 $g(S(H))$，以供复用。对于 UoT 和直接比较，如果当前历史已由 GoT 探索过，则会复用相应缓存的候选问题集合，而不再从大语言模型采样生成新问题。这样做是为了确保各方法之间的公平比较，因为候选问题质量的差异可能会显著影响性能。

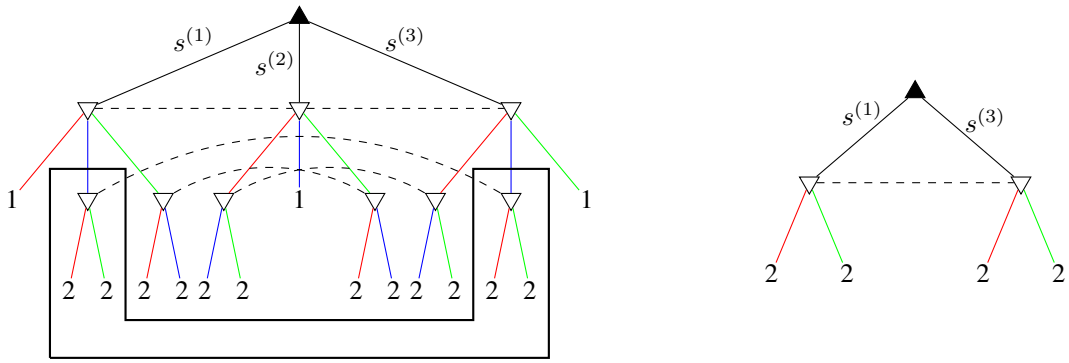

图7. 左：一个基于示例1 和图1的子博弈示例。由粗黑线围成的区域构成一个子博弈示例，对应于询问 $q^{(2)}$ 并得到答案1的历史。右：用于求解左侧所示子博弈的装置博弈。请注意，首先采取行动的是回答者，而非机会（朴素子博弈求解的情形下才会是机会），并且这些行动仅包括与 H一致的物品，在本例中为 $s^{(1)}$ 和 $s^{(3)}$。

处理大语言模型的错误输出 为便于解析，我们要求大语言模型以 JSON格式生成输出。尽管模型偶尔无法遵循这一格式，我们会通过重试生成来处理此类情况。在我们的实验中，这种重试机制效果良好，未观察到其他问题。

C.2. GoT

当模拟树处于终止状态时在模拟步骤（1）中，构建局部模拟树之后，我们验证每个叶节点是否对应于一个终止状态，即 $|S(l)| = 1$ 是否对所有叶节点成立。若该条件成立，则由当前子博弈导出的策略将用于直至博弈结束的所有后续问题选择，而非仅用于下一个问题并在下一步重新求解一个新的子博弈。其理由在于，当前子博弈已延伸至博弈结束，从而无需在后续步骤中进行迭代式策略构建。

C.3. BR

获取最佳响应 为获得最佳响应（BR），我们考虑一种非对抗性设定，其中项目（分布）从已知先验分布 P^o 中抽取，而非由对手选择。在这一假设下，将根节点（对应物品选择者）替换为一个机会节点（其出边的权重按 P^o 设置），交互过程便可建模为一个具有不完全信息的单人扩展式博弈，这与我们的 SLS 表示类似。此时，最优提问者策略是确定性的，并应将预期成本降至最低。我们通过采用一种类似于 GoT 的迭代式按需框架来获得这一策略，其中将步骤 3 替换为简单的逆向归纳。

示例4。 考虑一个与示例 *1*类似的博弈，其中$s^{(1)}$、$s^{(2)}$、 $s^{(3)}$ 按照先验分布 P^o =(0.8，0.1、0.1)分布。

在示例4中，提问者应始终先询问 $q^{(1)}$ ，随后询问 $q^{(2)}$ 或 $q^{(3)}$。这样得到的期望成本为1。2。图8 展示了示例4的扩展式博弈表示。

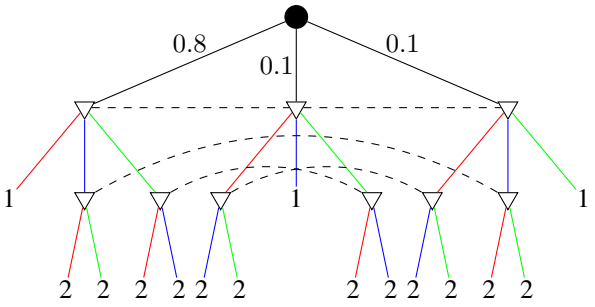

图8。 示例 4的扩展式博弈表示。根节点是机会节点，●余节点属于提问者。图中显示了每个项目各自的概率。

请注意，在此设置中，该问题等价于在部分可观测性下进行规划，即部分可观测马尔可夫决策过程（POMDP）（Spaan，2012）。

D. 补充结果

D.1. 使用合成划分的 SLSR

在 SLSR 的某个信息集 $I(H)$ 中，候选问题 q 的质量可以用比率 $\lrcorner|Y(S(H)\ q)|$ 粗略衡量： $|Y(S(H)\ q)|$，该比率表示 q 将 $S(H)$ 分成两部分的均匀程度。鉴于难以控制大语言模型采样的问题的质量，我们希望在一个更可控的环境中研究 SLSR。我们不使用大语言模型进行问题生成，而是定义一个问题生成函数 g'： $2^{\mathcal{S}} \setminus \phi \times (0,1) \to 2^{\mathcal{S}}$，它接收

作为输入 $S(H)$ 以及划分比例 $r \in (0,1)$，并以固定比例 $\frac{|Y'|}{|S(H)|} = r$ 输出集合[2] $Y' \subset S(H)$（从而得到 $\bar{Y}' = S(H) \setminus Y'$）。我们考虑子 g' 的两种实现方式：（i）随机划分：从 $S(H)$ 中均匀随机抽取 $r \cdot |S(H)|$ 个物品，不放回，并返回这些物品。（ii）基于特征的划分：在博弈开始时，为每个 $s \in S(H)$ 生成 k 个特征 $F_i = (f_1, ..., f_k)$ $f_j \in [0,1]$。g' 随机选择其中一个 k 特征，并根据所选特征对 $S(H)$ 进行排序。返回排名靠前的 $r \cdot |S(H)|$ 个物品。

结果如表格 3所示。GoT 的性能始终优于 UoT，尤其是在 r减小、划分越来越不均衡时。这表明，当候选问题并非最优时，GoT 带来的提升最大。然而，实际使用问题时，这种优势并不明显，因为还有其他未被纳入考量的混杂因素。

方法	r 0.4	0.33	0.25
3 个特征			
GoT	9.4	10.0	13.8
UoT	10.0	11.0	15.0
5 个特征			
GoT	9.2	10.2	13.2
UoT	10	11.0	15.0
随机划分			
GoT	8.8	9.8	11.6
UoT	10	11	15

表*3*。 **在包含 Common的 合成划分上，SLSR 的最坏情况交互长度。** d 设为 3。

D.2. 完整的 SLSR 游戏

对于每个 SLSR 博弈，所需问题数的期望最坏情况值都存在一个下界。通过求解完全指定的博弈即可得到该下界。由于这只适用于较小的数据集，我们在较小的 Breeds 数据集上构建了完整的 SLSR 博弈，并在 GoT 和 UoT 上进行实验，将模拟深度设为d 博弈树深度 D。由于这需要构建完整的博弈树，我们采用定理 E.1 ，以确保所有分支最终都会终止。结果见表格4。GoT 的性能与性能下界相符。另一方面，即使在模拟中使用完整博弈树，UoT 的性能也几乎没有改善，这表明要达到该下界，需要采用非确定性策略。

[2]在这种情况下，问题本身并不重要，因为 UoT 和 GoT 都与语言无关。

方法	实验 子博弈（$d = 3$）	完整博弈（$d = D$）
GPT 4.1 + 均分提示		
GoT	6.4 5.84	
UoT	7	7
GPT 4.1 +自然提示		
GoT	7.4	5.64
UoT	9	8
Qwen 72B Instruct + 均分提示		
GoT	6.2	5.44
UoT	7	7
Qwen 72B Instruct + 自然提示		
GoT	6.6	5.88
UoT	8	8

表 *4.* **完整 SLSR 博弈中的最坏情况交互长度。**使用的数据集为 Breeds 数据集。

D.3. 项目权重

图9 展示了 20Q 设置下 Breeds 和 Common 数据集、医学诊断设置下 DX，以及故障排查设置下 Flodial 的加权变体所用项目权重的直方图。对于 20Q 设置中的数据集，由于这些数据集的性质，难以确定为每个项目赋予有意义估值的方法，因此权重是随机分配的。对于故障排查和医学诊断设置，权重通过请 Chatgpt 5.2 为每种疾病/故障标注严重程度评分获得。所用提示见表格5

医学诊断以下是疾病列表。请为其中每种疾病评定 1 到 10 分的严重程度评分。该评分应取决于疾病可能危及生命的程度，以及治疗或护理的紧迫性。疾病列表：**[疾病]故障排查**以下是汽车故障列表。请为其中每种故障评定 1 到 10 分的严重程度评分。该评分应取决于故障的严重程度，以及修复的紧迫性。汽车故障列表：**[故障]**表 *5.* **用于标注医学诊断和故障排查设置下项目权重的提示。**

需要注意的是，GoT 在 WSLSR 博弈中的性能可能会受到项目权重底层分布的影响。具体而言，当权重从均匀分布中采样时，我们观察到，与 UoT 相比，GoT 的相对性能与非加权设置下的结果一致，但

改进幅度较小。

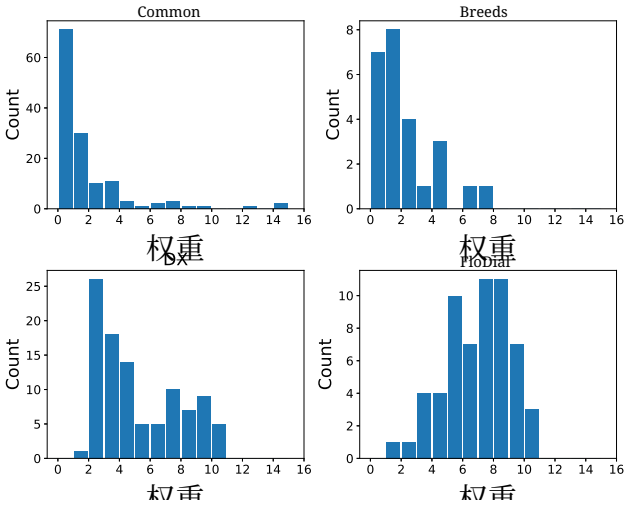

图*9*。**实验中使用的项目权重分布**。权重从参数为 $\mu = 0$, $\sigma = 1$ 的对数正态分布中采样。

D.4. GoT 策略的随机化

我们使用熵来量化 GoT 策略的随机性，如图所示10。总体而言，与 WSLSR 变体相比，在标准 SLSR 博弈中得到的策略具有更高的熵。这可能是因为项目权重分布呈现长尾特征，促使策略优先选择能够更有效地筛选出高权重物品的候选问题。尽管如此，这些策略仍具有足够的随机性，表明我们没有遇到确定性策略会成为最优策略的退化情形。

E. 关于假设的讨论

上述所有假设要么在实验设置中自然成立，要么在执行我们的方法时被明确强制满足。

定理 3.1 在对 SLS 问题的表述中，我们假设：(i) f 的陪域是二元的；(ii) 每个项目 $s \in \mathcal{S}$ 在 f 下都有唯一且明确定义的答案。然而，由于自然语言中普遍存在固有的歧义和解释差异，这一假设在现实场景中可能并不成立。这种歧义的一个来源是，条目名称可能指代多个不同的实体。例如，考虑这个问题：泰坦尼克号是人可以用手拿起来的东西吗？如果指的是那艘真正的船，答案是否定的；但如果指的是电影《泰坦尼克号》的 DVD，答案则是肯定的。在实验中，为了排除这类语言歧义的影响，我们使用大语言模型实例化标注函数 f，并将其

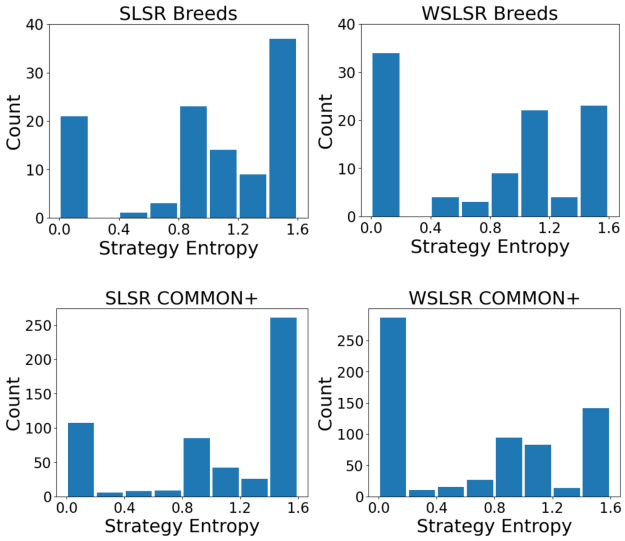

图 *10.* **使用 GoT 在每个信息集上获得的策略的熵**。 如 $m = 3$ 所示，当每个候选问题被选中的概率相同时，策略的最大熵为 $\log_2(3) \approx 1.58$ 。未包含仅剩两件物品时的策略。

输出视为每个项目的确定标签。

定理 3.2 这可由定理 3.1和3.10以及博弈结构自然推出。

定理 3.3 这可由博弈结构自然推出。

定理 3.4 我们使用大语言模型来实现函数 f，这些模型既通过在线 API 访问，也以本地托管实例的形式运行。对于基于在线 API 的查询，每次查询的平均延迟约为 1 到 5 秒；而对于本地托管的模型，延迟略低，通常约为 1 到 3 秒。

定理3.5 该假设成立，因为 $\mathcal{Q}$ 是所有可能的自然语言问题的集合。特别地，由于存在恒等问题，该假设平凡地成立。

定理 3.8 当大语言模型被用作问题生成器 g时，该假设在几乎所有情况下都成立。为强制满足该假设，我们验证采样得到的候选集中是否至少存在一个允许博弈继续推进的问题；若未找到此类问题，则重新采样候选问题。对于某些实验（例如完整求解SLSR博弈），我们施加该假设的更强版本：

假设 E.1. 对于每个子集 $S \subseteq \mathcal{S}$，对于每个 $q \in$

$g(S)$，存在一对不同的物品 $s,\ s' \in S$ 其中$f(q,\ s) \neq f(q,\ s')$

这确保了博弈树是有限的，从而能够完整构建博弈树。为满足这一条件，我们会从候选问题集合中明确移除任何不符合该假设的候选问题。

定理 3.10 作出此假设是为了减轻基于大语言模型的分类可能导致的不准确性。尽管大语言模型可能会产生错误输出，但处理此类错误超出了我们方法的范围，因此并非本文的重点。如果没有这一假设，模拟过程中的分类错误在最坏情况下可能导致博弈重新开始。具体而言，如果目标项目 s^* 被错误分类，那么在收到回答者给出的矛盾回复后，它可能会被错误地从可能的物品集合中排除。此外，我们发现，在尝试复现 Hu 等人 Hu 等人(2024年) 的结果时，此类错误分类及由此引发的重启频繁发生，这严重影响了最坏情况性能。由于这不会影响所评估方法中提问者的策略，我们认为这不会损害实验结果的公平性。为了在 GoT 和 UoT 中都满足这一假设，我们没有采用 Hu 等人 Hu 等人(2024年) 使用的方法——他们采用一个独立的大语言模型实例来模拟回答者。相反，我们会缓存模拟步骤中针对每个问题生成的项目集划分，并在实验中将其用作回答者的回复。这一策略确保了模拟阶段执行的项目分类与各方法实际执行期间观察到的回复保持一致。

F. 数据集详情

所有构建的数据集（以及加权变体所使用的项目权重）均与随附代码一同发布。

20Q 构建数据集时，我们依赖公开可用的来源。例如，Breeds 数据集的数据来自美国养犬俱乐部，S128 数据集的数据来自 ChatGPT。随后，我们对所有人工智能生成的数据集进行验证，以确保其中的每个项目都对应一个现实世界实体。这些数据集的规模经过刻意设计，与 Hu 等人(2024年所使用的数据集相当，因为他们的方法（UoT）与 GoT 都包含未来模拟步骤，因此计算复杂度相近。GoT 中的博弈求解对运行时间的影响微乎其微，因此 GoT 可视为 UoT 的稳健扩展，且只需付出很小的额外代价。这也是我们主要将我们的方法与 UoT 进行比较的原因。

医学诊断 DX 数据集包含 1148 条真实病例记录，涉及 461 种独特疾病。我们从这 461 种独特疾病中均匀随机抽取 100 种，作为项目假设空间。在平均情况分析中，我们统计 DX 数据集中每种疾病作为最终诊断的记录数，并以此估计总体分布。

故障排查 FloDial 包含与笔记本电脑和汽车相关的故障。我们决定聚焦于后者，因为其种类更多，且各类故障之间的区别更为明确。在该数据集中，每种故障都对应有向图中叶节点处的一段简短文本描述，该有向图表示故障排查流程图。我们提取这些叶节点描述，并提示 GPT-5.2 Thinking 将每条描述映射到具体的汽车故障标签。随后，我们根据 FloDial 提供的相关故障排查对话，手动核验并编辑生成的标签。最后，我们合并或移除近似重复故障（例如“漏水”和“水泵泄漏”），以确保保留的每种故障都各不相同且易于区分。

G. 提出的问题示例

不自然的问题 在使用 Common 数据集进行「二十个问题」博弈时，大语言模型通常会提出类似以下的问题:

- 这个物品是否是一个人（假设其体型和力量处于平均水平）可以用手拿起来的?
- 这个物品是人造的吗?
- 这个物品是动物吗?

这类问题通常涉及物品本身的内在属性。

相比之下，使用偶数提示时，我们偶尔会观察到生成更“不自然”的问题:

- 这个品种的名称是否只有一个词?
- 这个项目的名称是否超过 7 个字符?
- 这个项目的名称是否以 A-M（含 A 和 M）之间的字母开头?

尽管我们很难对问题的“自然度”进行定量评估，但直观来看，不自然的问题通常会直接提及条目名称，而不是底层物品本身。另一方面，自然问题通常更关注物品的内在属性

。不自然的问题可以被解读为大语言模型试图通过提高获得均匀二分结果的机会来优化问题，而根据定理3.7，均匀二分结果是最优的。这些情况主要出现在使用 GPT-4.1 作为大语言模型时。我们推测，这可能是使用 *even* 而非 *natural*时，直接提示（DP）能够观察到显著提升的原因，正如表格 6所示。这种现象在 GoT 和 UoT 中并不明显，可能是因为二者都通过前瞻进行规划。这或许降低了少数“不自然”问题对整体性能的影响。

方法	20Q Common	20Q S128	20Q Breeds
GPT 4.1 +均分提示			
GoT	9.4	9.2	6.4
UoT	10	10	7
DP	11.7	12.9	7.8
DC	12.3	12.6	9.0
GPT 4.1 + 自然提示			
GoT	10.2	11.8	7.4
UoT	11	13	9
DP	13.8	16.2	7.8
DC	12.9	14.6	9.3
Qwen 2.5 72B Instruct +均分提示			
GoT	10.2	10.2	6.2
UoT	11	11	7
DP	11.9	14.6	8.0
DC	12.4	13.6	8.3
Qwen 2.5 72B Instruct +自然提示			
GoT	10.0	10.8	6.6
UoT	11	12	8
DP	12.7	19.2	8.0
DC	12.7	17.9	7.8

表格 *6.* **使用均分提示或自然提示时各方法的最坏情况交互长度**。对于 UoT 和 GoT，我们均设置 $d = 3$。

实际场景中的问题 如前所述，我们声称在医学诊断和故障排查设置中由大语言模型生成的问题相当自然。为说明这一点，我们提供了在每种设置中从 GPT 4.1 采样的问题的代表性示例。

医学诊断采样问题示例：

- 主要症状是否为皮肤科症状，并影响皮肤（例如皮疹、湿疹、色素沉着、皮损）？
- 该疾病主要与感染还是炎症相关（例如肝炎、支气管炎、肺炎、传染性单核细胞增多症、败血症、咽炎、胃炎）？
- 主要症状是否与呼吸系统有关（例如咳嗽、呼吸急促或胸痛）？

这些问题针对患者的症状或受影响的身体系统，这与临床实践中通常进行诊断的方式一致。此外，每个问题都可以（i）稍作改写后向患者提出，（ii）在临床推理过程中用作思维检查清单，或（iii）对应到与特定疾病相关的诊断检测。

故障排查中抽取的问题示例：

- 将钥匙转到“ON”位置时，您是否注意到仪表盘上有任何警告灯或指示灯亮起?
- 即使点火装置处于关闭状态，是否仍有任何电气附件（如车头灯或仪表盘灯）无法正常工作?
- 转动点火钥匙时，汽车是否无法启动或无法带动发动机运转?

这些问题易于理解，而且重要的是，涉及的症状客户可以轻松确认。这一特性使它们适用于故障排查场景。

歧义问题 在对 SLS 问题的表述中，我们假设：**(i)** f 的陪域为二元，且 **(ii)** 每个项目 $s \in \mathcal{S}$ 在 f下都有唯一且定义明确的答案，如假设 3.1所述。然而，由于自然语言固有的歧义和常见的解释差异，这一假设在现实场景中可能并不成立。歧义的一种来源是条目名称指代多个不同的实体。例如，考虑这个问题：泰坦尼克号是人可以用手拿起来的东西吗？如果指的是那艘真正的船，答案是否定的；但如果指的是电影《泰坦尼克号》的 DVD，答案则是肯定的。尽管语言歧义是自然语言处理领域中研究充分的现象，但将其应用于博弈论这一更为形式化且严谨的框架时，解决此类歧义仍颇具挑战。我们认为，解决这一挑战是未来研究一个颇有前景的方向。

H. 使用的提示词

用于抽样问题（以实现函数 g）的提示词在加权变体和无权重变体中分别如表格7 和 8 所示。直接提示通过设置 $m = 1$使用相同的提示词来提问。直接比较还使用表格9 中的提示词从候选集中选择问题。用于为物品生成问题答案（以实现函数 f）的提示词如表格10所示。

I. 实验硬件

在配备 32 GB 内存的 AMD Ryzen 5 5500 中央处理器上运行扩展式博弈求解器。通过 vLLM（Kwon 等人, 2023年）库在本地托管 Qwen 2.5 72B Instruct 大语言模型，使用 2 块各配备 141 GB 显存的 Nvidia H200 GPU，或 4 块各配备 80 GB 显存的 Nvidia H100 GPU。

甚至是提示 20Q我们正在玩提问与寻找的博弈，我是回答者，你是提问者。我从一份物品清单中选了一件物品，你的目标是通过提出一系列是非问题，从清单中找出正确物品。你应尽量减少提问次数。以下是所有物品：

[物品]

其中一件物品是正确物品，请针对这些物品设计问题，帮助你找出正确物品，并且问题只能用“是”或“否”作答。特别要注意，回答“是”和“否”的物品数量应尽可能相近，差距最多不得超过一件！请给出你能想到的最佳 **[m]** 个问题。**自然提示 20Q**我们正在玩提问与寻找的博弈，我是回答者，你是提问者。我从一份物品清单中选了一件物品，你的目标是通过提出一系列是非问题，从

请针对这些物品设计问题，帮助你找出正确物品，并且问题只能用“是”或“否”作答。你应避免提出不自然的问题，例如询问物品包含多少个单词，或类似的问题。请给出你能想到的最佳 **[m]** 个问题。**医学诊断**你是一名医生，正尝试诊断一位患者。患者患有所提供清单中的一种疾病，你的目标是通过提出一系列是非问题，从清单中找出正确项目。你应尽量减少提问次数。以下是所有疾病： **[疾病]**其中一种疾病是正确疾病，请针对这些项目设计问题，帮助你找出正确项目，并且问题只能用“是”或“否”作答。请给出你能想到的最佳 **[m]** 个问题。**故障排查**你是一名汽车修理工，正在为客户提供远程故障排查服务。客户的汽车出现了所提供清单中的一种故障，你的目标是通过提出一系列是非问题，从清单中找出正确故障。你应尽量减少提问次数。以下是所有故障： **[故障]**其中一种故障是正确故障，请针对这些故障设计问题，帮助你找出正确项目，并且问题只能用

“是”或“否”作答。请给出你能想到的最佳 **[m]** 个问题。

表7. **用于为无权重变体采样候选问题的提示词**。 常规提示与自然提示的主要区别以斜体标出。物品以 Python 列表表示，例如 ['奥本海默'、'艾伦·图灵'、'《美丽心灵》'], , 而 m 是一个整数

20Q我们来玩“提问与寻找”博弈：我是回答者，你是提问者。我从一份物品清单中选出一件物品，你的目标是通过提出一系列是非问题，从清单中找出正确物品。每件物品都对应一个数值，你的惩罚分数等于找到正确物品之前提出的问题数量乘以正确物品对应的数值。你应尽量降低这个分数。以下是所有物品及其对应数值： **[带权重的物品]**其中一件物品是正确物品，请围绕这些物品设计问题，帮助你找到正确物品，并且问题只能用“是”或“否”作答。请避免提出不自然的问题，例如询问物品包含多少个单词，或类似的问题。请给出你能想到的最佳 **[m]** 个问题。**医学诊断**你是一名医生，正在尝试诊断一位患者。患者患有所提供清单中的一种疾病，你的目标是通过提出一系列是非问题，从清单中找出正确的疾病。每种疾病都有对应的严重程度。你的惩罚分数等于提出的问题数量乘以正确疾病的严重程度。你应尽量降低这一惩罚分数。以下是所有疾病： **[带权重的疾病]**其中一种疾病是正确疾病，请围绕这些疾病设计问题，帮助你找到正确疾病，并且问题只能用“是”或“否”作答。请给出你能想到的最佳 **[m]** 个问题。**故障排查**你是一名汽车修理工，正在为客户提供远程故障排查服务。客户的汽车存在所提供清单中的一种故障，你的目标是通过提出一系列是非问题，从清单中找出正确的故障。每种故障都有对应的严重程度。你的惩罚分数等于提出的问题数量乘以正确故障的严重程度。你应尽量降低这一惩罚分数。以下是所有故障： **[带权重的故障]**其中一种故障是正确故障，请围绕这些故障设计问题，帮助你找到正确故障，并且问题只能用“是”或“否”作答。请给出你能想到的最佳 **[m]** 个问题。

表格 *8.* **用于为加权变体采样候选问题的提示**。带权重的物品以 Python 字典表示 词典, e.g. {‘奥本海默’: 3 , ‘艾伦·图灵’: 2 , 《美丽心灵》： 2} , 其中 m 是一个整数

为 20Q 无权重变体选择提示词

我们正在玩提问与寻找的博弈，其中我是回答者，你是提问者。我从一份物品清单中选出了一件物品，你的目标是通过提出一系列是非问题，从清单中找出正确物品。你应该尽量少提问题。以下是所有物品：

[物品]

以下是一些关于它们的是非问题：

[问题]

其中一个项目是正确物品，请从问题列表中选择一个有助于你找到正确物品的问题。

为 20Q 加权变体选择提示

我们正在玩“提问与寻找”博弈，我是回答者，你是提问者。我从一份物品清单中选取了一件物品，你的目标是通过提出一系列是非问题，从清单中找出正确物品。每件物品都有一个对应的数值，而你的惩罚分数等于找到正确物品之前提出的问题数量乘以正确物品的数值。你应该尽量降低这个分数。以下是所有物品：**[带权重的物品]**以下是一些关于这些物品的是非问题：**[问题]**

其中一件物品是正确物品，请从问题列表中选择一个有助于你找到正确物品的问题。

医学诊断无权重变体的提示选择’你是一名医生，正在尝试诊断一位患者。患者患有所提供列表中的一种疾病，你的目标是通过提出一系列是非问题，从列表中找出正确的疾病。你应该尽量少提问题。以下是所有疾病： **[疾病]**以下是一些关于这些疾病的是非问题： **[问题]**其中一种疾病是正确疾病，请从问题列表中选择一个问题，帮助你尽快缩小范围并找到正确的疾病。**故障排查无权重变体的提示选择**你是一名汽车修理工，正在为客户提供远程故障排查服务。客户的汽车出现了所提供列表中的一种故障，你的目标是通过提出一系列是非问题，从列表中找出正确的故障。你应该尽量少提问题。以下是所有故障： **[故障]**以下是一些关于这些故障的是非问题： **[问题]**其中一种故障是正确故障，请从问题列表中选择一个问题，帮助你尽快缩小范围并找到正确的故障。

表9 {0: 直接比较使用的提示词；用于选择问题。”，2: 以是人物哪些的形式表示，例如 Pyrrhoid 等典的，MD 和 TS 设置下的提示词加权变体，通过以类似 20Q 设置的方式修改无权重变体得到。

答案提示以下是一些 **[物品/疾病/汽车故障]:[物品]**以下是一个关于它们的是非问题：**[问题]**请根据上述问题对上述 **[物品/疾病/汽车故障]** 进行分类。请确保每个项目在最终分类中只出现一次。如果不确定某个项目，可以在说明中提及。*表10*。**用于获取问题答案的提示**。对于 20Q、M 和故障排查设置，我们分别将假设空间称为物品、疾病或汽车故障

D,

和故障排查设置。