arXiv:2602.08630v1 [cs.AI] 9 Feb 2026

辩论能高效利用你的时间

JonahBrown-Cohen GeoffreyIrving SimonC. Marshall* IlanNewmanGoogle DeepMind 英国人工智能安全研究所 英国人工智能安全研究所 海法大学GeorgiosPiliouras
MarioSzegedyGoogle DeepMind 罗格斯大学

摘要

通过辩论实现 AI 安全，利用两个相互竞争的模型帮助人类裁判验证复杂的计算任务。以往研究已确立辩论原则上能够解决哪些问题，但尚未分析人工监督的实际成本：裁判需要对辩论记录进行多少次查询？我们提出 $Debate\ Query\ Complexity$ (DQC)，即验证者为正确判定一场辩论必须检查的最少比特数。

令人惊讶的是，我们发现 PSPACE/poly（辩论能够高效判定的问题类）恰好等于可通过 $O(\log n)$ 次查询判定的函数类。这一刻画表明，辩论的查询效率极高：即使面对复杂度很高的问题，对数级监督也已足够。我们还证明，依赖所有输入比特的函数需要 $\Omega(\log n)$ 次查询，并且任何可由大小为 s 的电路计算的函数都满足 $DQC(f) \leq \log(s) + 3$。有趣的是，后一结果意味着，对于 P 中的语言，若能证明其 DQC 下界为 $\log(n) + 6$，就能得到新的电路下界，从而将辩论查询复杂度与电路复杂度中的核心问题联系起来。

1 引言

现代人工智能系统经过海量数据的训练，也会生成海量数据，这使得有效监督和对齐变得困难。随着这些系统的能力不断增强，我们面临一个根本瓶颈：必须利用相对少量的人类反馈，监督日益复杂的模型。与此同时，即使是专家，也会发现难以直接评估先进人工智能系统的输出，这意味着我们的监督体系还必须能够扩展我们的能力。

应对这一挑战的一种有前景的方法是通过辩论实现 *AI 安全* [1, 2], ：两个人工智能模型通过提出论据展开竞争，由一位人类裁判判定哪一方正确。辩论之所以引人注目，是因为它有望增强人类判断能力；裁判无需直接解决问题，只需在相互竞争的解决方案之间作出裁决。随着形式化保证不断增强，辩论在理论上的吸引力也日益提升：在辩手不受计算能力限制、裁判为多项式时间裁判的情况下，辩论可以判定 PSPACE 中的语言 [1]。近期研究已将类似保证扩展到更现实的场景，例如辩手的计算能力受到限制的情况 [3, 4]。

然而，先前的研究并未分析人工监督的成本——即裁判必须查询辩论记录的次数。这项成本对于实际部署至关重要，因为人工时间通常是最昂贵的资源之一。我们引入**辩论查询复杂度（DQC）**，即验证者必须检查辩论记录中的最少比特数，才能正确判定一个布尔函数f： $\{0,1\}^n \to \{0,1\}$。

我们建立了 DQC 的紧确界限。依赖输入所有比特的函数需要 $\Omega(\log n)$ 次查询，而任何可由二输入布尔电路 C 计算的函数都满足 $DQC(f) \leq O(\text{depth}(C))$ 和 $DQC(f) \leq O(\log(\text{size}(C)))$。这些界限借助对 Karchmer-Wigderson 博弈的改造得出：交叉质询 [2, 3], 中，一位辩手执行计算，另一位则验证其中一个步骤。

{simon.marshall@dsit.gov.uk

我们的主要结果表明，辩论在查询方面极其高效：

定理。

$$\mathsf{PSPACE/poly} = \{f : DQC(f) \le O(\log n)\}$$

高效辩论（即多项式长度的辩论和多项式时间验证者）等价于 PSPACE [1],因此，这一结果意味着，如果辩论能够判定一个问题，那么只需 $O(\log n)$步即可做到；此外，验证者本身的计算效率也非常高。证明采用递归交叉质询：Alice 逐步模拟验证者的计算，Bob 指出一处疑似错误，而验证者只检查那个位置。这表明，即使面对复杂任务，也能实现有效监督，而且只需少量人类判断。

我们的电路上界还揭示了一个与电路复杂度之间耐人寻味的联系：对于 P 中的任意语言，若要证明 DQC 下界超过 $\log n+6$，就必须得到超过目前已知最佳 $\approx 5n$界的电路下界（推论7）。这意味着，证明 DQC 的紧下界可能与证明电路下界一样困难。反过来，这将证明电路下界的任务化约为一个纯粹的信息论设定；其精神类似于 Karchmer-Wigderson 博弈，该博弈将深度界化约到另一种信息论设定，并将我们的查询复杂度度量与复杂性理论中的一个核心开放问题联系起来。

最后，我们在一个附加章节中考察了带有随机化验证者的辩论版本。我们证明，在有意义的范围内，随机化并无帮助。

组织结构。第 2 节定义了辩论模型和 DQC。第3节通过决策树和电路模拟给出上界结果。第4节建立了 PSPACE/poly刻画。第5节定义了带有随机化验证者的辩论，并证明对于依赖所有变量的函数，至少需要 $\Omega(\log n)$ 次查询。第6节讨论开放问题。

2 定义

我们研究一种确定性交互式证明模型，其中包含两个相互竞争的全能证明者以及一个查询受限的验证者 V。我们的模型受到用于人工智能安全的辩论协议 [1], 的启发，但通过查询复杂度的视角进行分析。正如证明者-验证者设定中的典型情况，验证者不信任任何一方证明者，而是依靠对抗性来获得可靠结果。

设 $f\colon \{0,1\}^n \to \{0,1\}$ 为一个固定的布尔函数，双方证明者与验证者均已知该函数。对于输入 $x \in \{0,1\}^n$，各验证者已知，但证明者并不知道，

- **证明者 0：**试图说服验证者 M 相信 $f(x) = 0$
- **证明者 1：**试图说服验证者 M 相信 $f(x) = 1$

对于预先固定的某个 k，证明者交替写入消息 α_1，β_1，……，α_k，β_k，其中证明者 0 写入 α_i，证明者 1 写入 β_i。我们记 $\alpha = (\alpha_1, \ldots, \alpha_k)$ 与 $\beta = (\beta_1, \ldots, \beta_k)$，并将 $T = (\alpha, \beta)$ 称为辩论记录。

验证者 M_ℓ 是一台确定性查询机，它自适应地从拼接字符串 $x_1, \ldots, x_n, \alpha_1, \beta_1, \ldots, \alpha_k, \beta_k$ 中查询 ℓ 比特，并输出 $M(x, \alpha, \beta) \in \{0,1\}$。

如果诚实证明者总能针对不诚实证明者的任何策略迫使正确结果出现，我们就称该辩论协议是针对 f 的 (k, ℓ)-辩论。

形式化地说：A (k, ℓ) 辩论是指针对 f 的一组两个确定性策略，每个策略的长度均为 k，分别由诚实证明者 0 和诚实证明者 1 使用；以及一个确定性的 ℓ 查询机 M_ℓ，作用于 $2k + n$ 输入（即验证者），满足：

$$\text{For every } x \in f^{-1}(1)\ \ \forall\alpha_1\exists\beta_1\ \ldots\ \forall\alpha_k\exists\beta_k :\ \ M_\ell(x, \alpha, \beta) = 1 \qquad \text{（证明者 1 获胜）} \tag{1}$$

and

$$\text{For every } x \in f^{-1}(0)\ \ \exists\alpha_1\forall\beta_1\ \ldots\ \exists\alpha_k\forall\beta_k :\ \ M_\ell(x, \alpha, \beta) = 0 \qquad \text{(Prover 0 wins)} \tag{2}$$

交替量词体现了辩论的博弈论性质：诚实证明者可以回应对手的任何一步行动。根据量词的德摩根定律，对于任意 M，条件 (1) 和 (2) 互为补充，从而确保协议定义良好。

注： 辩论的长度 k 是固定的（与 x 无关），且各方均知晓该长度。

定义 1（辩论查询复杂度）。布尔函数 $f : \{0,1\}^n \to \{0,1\}$ 的辩论查询复杂度，记为 $DQC(f)$，是满足存在针对 f 的（k，ℓ）-辩论协议的最小 ℓ。

注： 在 $DQC(f)$ 的定义中，我们不对 k 设定任何上界。也就是说，$DQC(f)$ 只衡量验证者查询的比特数，而不衡量记录的总长度；记录的总长度可能大得多。

定义 2（有效记录，有效辩论系统）。对于如上所述的验证者 M 和证明者 A、B，如果 A（B，M）能在输入 x 上正确输出 $f(x)$，则称其为针对 f 的有效辩论系统。

如果记录 $T = (\alpha, \beta)$ 是在 x 上由诚实证明者与可能不诚实的证明者进行交互而得到的，则称其为针对输入 x 上 f 的有效记录。如果存在某个输入 x，使得 T 是在 x 上针对 f 的有效记录，则记录 $T = (\alpha, \beta)$ 是针对 f 的有效记录。

2.1 辩论的一些基本特征

1. 与 PCP理论或 NP 的定义一样，我们常常赋予辩论记录某种“含义”（例如，“α编码了一次电路求值”）。然而，原则上，α 和 β 可以是任意字符串。任何预期的协议结构都完全由验证者 V以及上文方程 2 和 1 中的条件来强制实现。因此，$DQC(f)$ 的定义只取决于是否存在一个满足方程 1 和 2 的合适查询机 M_ℓ。

   此外，尽管证明者 0 和证明者 1 的行为可能取决于 x（他们都能看到它），但由于 M_ℓ是一个确定性的 ℓ-查询机，它会根据相应的变量 $\{x_i |\ i \in [n]\} \cup \{\alpha_o, \beta_i |\ i \in k\}$. 计算一个固定函数 V：$\{0,1\}^{2k+n}$。特别地，V可能不同于 f。

2. 如上所述，前一项中的函数 V 完全决定了 f：通过按标准方式将量化变量替换为 $\wedge$-门和 $\vee$-门，就可以从 V 计算出（即写出一个关于）f 的公式。由此立即可知，如果 f 依赖于它的所有变量，那么 V 也必须依赖于 $\{x_1, \ldots, x_n\}$ 中的每个变量（尽管它不一定依赖于额外的 $2k$ 个变量中的所有变量）。

3. 由于 V是一个确定性查询复杂度至多为 ℓ的函数，因此它至多依赖 2^ℓ 个变量。这意味着：

   • 根据前一项，对于一个依赖其所有变量的函数 f，V必须至少依赖n 个变量（即 $\{x_1, \ldots, x_n\}$）。因此，我们在推论 4 中得出，这类 f的查询复杂度下界为 $\ell \geq \log\ n$。

   • 存在一个固定的变量集合 $S \subseteq \{x_i |\ i \in [n]\} \cup \{\alpha_i, \beta_i |\ i \in [k]\}$，$|S| \leq 2^\ell$，使得 V仅依赖于 S。这或许意味着对话记录长度 $2k$ 可以由 $2k \leq 2^{2\ell}$界定，因为如果既不查询 α_i，β_i，我们就可以将它们从记录中省略。尽管如此，这一点并不显而易见，正如引理 3 所示。由于策略具有自适应性，先验来看，为了输出被查询的 α_{i+1}（例如针对诚实证明者0），证明者可能必须查看β_i，尽管 α_i，β_i 不会被查询。

4. 我们定义的辩论从证明者 0 开始。也可以采用类似的定义，让证明者 1 先开始，不过两者最多只相差一次查询。同样，在我们的辩论定义中，我们强制要求记录中的发言交替进行。对于所有 x，如果允许消息更长，就可以避免这一要求，但对话记录长度最多只能缩短一半。

5. 我们的定义使用确定性验证器，但在许多场景中，随机化验证器往往更高效。我们将在后文展示，随机化 DQC 带来的改进有限；而对于最受关注的 PSPACE 情形，则毫无改进。

引理 3。 如果 *(A,B,M)* 是针对 f 的有效(k,ℓ)-辩论系统，那么存在一个（可能不同的）有效(k',ℓ)-辩论，针对 f，且其中 $k \leq 2^{\ell}$。

证明。 令 V 为验证器机器 M 计算的布尔函数（见上文第 1 项）。令 S 为 M 在所有可能的输入 x 和记录上计算 V 时可能查询的全部变量的集合。如上文第 3 项所述， S 是一个大小固定为 $|S| \leq 2^{\ell}$ 的集合。

假设 $\alpha_i,\ \beta_i \notin S$，其中 $i \in k$ 是某个值。我们将说明，可以通过删除辩论的第 i 轮、保留验证者，从而使函数 V 保持不变，构造出针对 f 的新有效 $(k-1,\ \ell)$-辩论。乍看之下，这显而易见：从验证者的角度来看，由于其函数 V 不依赖于 $\alpha_i,\ \beta_i$，因此可以将它们从记录中删除，因为记录可视为输入变量的部分列表（字符串）。然而，只删除第 i 轮可能会使记录无效，因为 $(i+1)$ 轮中的比特取决于第 i 轮交换的值。我们将证明，尽管辩论博弈具有这种自适应性，仍然可以删除这样的轮次。

令 i 为满足 $\alpha_i, \beta_i \notin S$ 的最小索引。其思路是，新证明者 0、新证明者 1、新 Alice 和新 Bob 的策略，在辩论的前 $i-1$ 轮中分别与 Alice 和 Bob 的策略完全相同。然后在第 i 轮，诚实证明者会假设对方在第 i 轮给出一个任意回应，同时模拟自己的回应，从而能够立即 “ 跳到 ” 第 $i+1$ 轮。

形式上，对于输入 $x \in f^{-1}(0)$，新 Alice 完成前 $i-1$ 轮后，就能完全知晓她的 α_i 位是什么。然后她假设 $\beta_i = 0$ （这是一个任意的决定，但它是合理的，因为作弊的 Bob 可能会那样回应）。此时，新 Alice 知道如何在第 $i+1$ 轮继续辩论，跳过第 i 轮。她可以继续遵循自己的策略，并且应该能够获胜。

类似地，对于 $x \in f^{-1}(1)$，新-Bob 假设 $\alpha_i = 0$ （同样，这是一个任意决定），因此能够计算自己的 β_i，而无需实际输出它。随后，可能作弊的 A’ 可以继续其部分，输出 α'_{i+1}；它并无特别重要之处（因为它可能是作弊的），新-Bob 将据此模拟 Bob 的（$i+1$）步，使其成为新记录中的第 i 步。

我们注意到，在新的辩论中，我们拥有一组限制更多的策略（因为我们消去了一轮），但由于新 Alice 可以在 0 输入上强制获胜，而新-Bob 可以在 1 输入上强制获胜，因此新系统对 f 是有效的（我们在此利用了 V 在任何可能的运行中都不会查询 $\alpha_i,\ \beta_i$，因此从他的角度来看，他看到的记录比特可能出现在原始辩论中）。

因此，只要仍存在上述索引 i，就可以用这种方式跳过若干轮。由于 $|S| \leq 2^{\ell}$，只要 $k > 2^{\ell}$ 就可以这样做，从而完成证明。 □

推论 **4.** 对于任意依赖于其所有变量的布尔函数 $f\colon\ \{0,1\}^n \to \{0,1\}$ ，

$$DQC(f) \geq \log n.$$

3 电路上界

现在我们为 DQC 建立上界。我们得到了两类界：通过改造 Karchmer-Wigderson 博弈得到的基于深度的界 $DQC(f) \leq depth(C_f)$，以及通过交叉质询得到的基于规模的界 $DQC(f) \leq \log(size(C_f))+3$。这些界采用了不同的技术，各有不同的权衡，适用于不同类别的电路。

我们用 C_f 表示实现函数 f的扇入为 2 的与/或电路。不失一般性，我们假设 C_f 是一个分层电路，其顶层为与门，后续各层的门在与门和或门之间交替出现，不过这仅对第 3.1 小节是必要的。我们用 $depth(C_f)$表示 C_f 的深度，用 $size(C_f)$表示门的数量。为避免使用向上取整运算符，我们还可以假设规模和深度都是 2 的幂。

3.1 深度上界

我们的第一个结果基于 Karchmer-Wigderson 博弈。Karchmer 和 Wigderson 注意到，布尔函数 f 的电路深度与以下关系的通信复杂度之间存在联系：Alice 获得 $x \in f^{-1}(0)$，Bob 获得 $y \in f^{-1}(1)$，他们必须通过通信输出一个索引 i，使得 $x_i \neq y_i$。所需的最少通信量等于 f的扇入为 2 的（$\wedge$，$\vee$）电路深度。我们将此博弈改编为辩论设定：证明者通过交替发送消息来遍历电路结构，从而说服验证者接受输出值。

定理 5. 设 C_f 是任意基上 f 的电路。则

$$DQC(f) \leq depth(C_f) + 1$$

证明。我们不妨假设电路 C_f 的顶部门是一个 $\wedge$-门（与门）。我们的证明利用了 Karchmer 和 Wigderson 为其通信复杂度设定所构造的博弈。

令证明者 0 担任防守方的角色，证明者 1 充当挑战方。具体而言，若 $f(x) = 0$，则证明者 0 能够成功捍卫 $f(x) = 0$这一主张，迫使验证者输出 0。反之，对于满足 $f(x) = 1$的输入 x，证明者 0 必然失败，并且在证明者 1 采取最优策略时，验证者将输出 1。

博弈从电路的顶部门开始，该门对应于输出位。这个最终门是一个$\wedge$，因此若确实如证明者 0 所声称的 $f(x) = 0$，则该门的输出为 0，从而其中一个输入必定为 0。证明者 0 的第一步（位 $= \alpha_1$）是指出该输入，将其作为两条输入支路之一。我们沿着这条输入支路追溯到下一个门，该门是一个 $\vee$。由于证明者 0 声称该门的求值为 0，若证明者 1 想要对此提出挑战，他就应当能够指出该门中值为 1 的输入。因此，这便是其位 β_1的解释。

这一交互会不断迭代：证明者沿着电路逐门向下，朝输入层推进。轮到证明者 0 时，他总会指向一个自己声称输出为 0 的 $\wedge$-门，并指出该门的一个值为 0 的输入；而证明者 1 则会指向一个自己声称输出为 1 的 $\vee$-门，并指出该门的一个值为 1 的输入。最终，过程会到达一个电路输入，验证者可以将其与相关输入 x_i 直接比较。

在这场游戏的每个阶段，如果最终输出确如诚实证明者所声称的那样，那么每位诚实证明者都应该能够指出自己偏好的值。验证者可以读取 $depth(C_f)$ 步（也就是整份记录），确定游戏指向哪个输入，然后将该值作为辩论的最终输出。 □

对于 AC^0 或 NC^1 中这类深度较浅的电路，深度上界是最优的。然而，对于具有深而小的电路的函数，这个上界可能不尽如人意。例如，一个大小为 n^2 、深度为 n 的电路会给出 DQC 上界 n，而我们可能希望得到更接近 $\log(n^2) = 2\log(n)$ 的结果。我们的下一个结果将解决这一局限。

3.2 通过交叉质询实现规模上界

我们可以采用一种不同的技术——交叉质询，实现对电路规模的对数依赖。在这种范式中，一位证明者执行整个计算并记录所有中间步骤，而另一位证明者则通过指出一个发生错误的位置来质疑这项计算。验证者只需检查第二位证明者指出的位置。该策略以辩论长度换取查询效率（辩论长度现在与计算规模呈线性关系，而查询效率现在为对数级），因为对于规模为 m 的计算，验证者只需 $O(\log m)$ 比特即可指定一个位置。

定理 6. $DQC(f) \leq \log(size(C_f)) + 3$

证明。证明者 0，也就是我们现在称为 Alice 的人，先进行 $size(C_f)$ 步操作，输出电路中每个门的值。随后，证明者 1（我们称其为 Bob）要么接受 Alice 的计算结果，要么指出某个门的位置，并说明该门的输出并非由其两个输入（可能是电路输入，也可能是 Alice 声称的其他门的输出）推导而来。

验证者读取 Bob 的输出，其中指定了一个门索引（需要进行 $\log(size(C_f))$ 次查询）。验证者前往 Alice 的记录中对应的位置，查询 Alice 声称的该门的值（1 次查询）以及其两条输入线的值（2 次查询）。由于所有参与方都知道电路结构，验证者可以确定该门的类型，并检查 Alice 给出的输出是否符合该类型门的逻辑表对这两个输入进行运算所得的结果。

如果 Alice 的门实现有误，验证者输出 1。此外，如果 Bob 指向输出门且其值为 1，验证者输出 1。否则，验证者输出 0。

查询总数为 $\log(size(C_f)) + 3$。 □

这种交叉质询技术相当通用，并将在下一节展现更强大的作用：我们将用它把任意多项式时间验证者压缩到对数查询复杂度。

我们的电路上界对电路复杂度有一个耐人寻味的推论：它们意味着，要证明 DQC 的强下界，至少与证明电路下界一样困难。由于目前已知的 NP 中语言的最佳显式电路下界约为 $5n$ [5]，我们的规模上界构成了一道障碍，使得 DQC 下界难以提升到对数级以上。

推论7 (与电路下界的联系)。 对于NP 中的某个函数，要证明其 DQC 具有足够大的下界，就能得到新的电路下界。具体而言，若存在一种语言 $L \in$ NP，其特征函数为 f，且满足 $DQC(f) \geq \log n + 6$，则 L 需要至少规模为 $8n$ 的电路。

证明。 根据定理 6，任何具有大小为 m 的电路的布尔函数 f 都满足 $DQC(f) \leq \log(m) + 3$。设 $L \in$ NP 是一个语言，其特征函数为 f：$\{0,1\}^n \to \{0,1\}$，并设 $\{C_n\}$ 是计算 f 的最小电路族（按电路大小计）。假设 $DQC(f) \geq \log n + 6$。那么：

$$\log n + 6 \leq DQC(f) \leq \log(\text{size}(C_n)) + 3$$

重新整理可得 $\log(\text{size}(C_n)) \geq \log n + 3$，这意味着 $\text{size}(C_n) \geq 2^{\log n+3} = 8n$.

由于目前已知的 P 中显式语言的最佳电路下界约为 $5n$ [5]，若能证明任何此类语言都满足$DQC(f) \geq \log n + 6$，将会是电路复杂度领域的一项重大突破。 □

4 交叉质询与PSPACE

上一节中的一个关键工具是交叉质询：让一位证明者执行计算，而另一位证明者要么接受，要么指出错误。通过将交叉质询形式化，我们能够证明 PSPACE/poly 恰好是可通过 $O(\log n)$ 次查询判定的函数类。证明分两步进行：首先，我们建立一个通用的交叉质询引理，表明任何多项式时间验证器都可以通过对数次查询来模拟。其次，我们利用该引理证明这一刻画的两个方向。

4.1 交叉质询引理

交叉质询的力量在于它能够压缩验证过程：验证者无需执行多项式时间计算，而是由一位证明者模拟该计算，另一位则验证其中的单个步骤。这样便将查询复杂度从多项式级降至对数级。

引理 8 (交叉质询引理)。 假设 f 有一个(k, ℓ)辩论，其中确定性验证器计算函数 V（输入比特为$(\alpha_1, \beta_1, \ldots \alpha_k, \beta_k, x_1, \ldots x_n)$），且该函数可由一个大小为 m的电路 C计算。那么

$$DQC(f) \leq \log(m) + 3$$

备注： 验证者计算一个布尔函数 V，根据定义，该函数的查询复杂度为 ℓ。因此，V的电路规模至多为 2^ℓ。不过， V 的最优规模可能是 $m << 2^\ell$。于是，引理利用第 3.2 节的结果，为 f构造一个（可能不同的）查询复杂度为 $\log(m)$ 的辩论 $+3$,

通过验证 V 的计算，而不是直接计算它。具体而言，它使用另一个验证者（函数） V' （以及扩充后的证明者），来验证函数 V，并同时验证 f的值。

第二点是，这个引理不能递归地应用于其自身的验证者，以获得不断降低的更优复杂度，因为当 $m \geq 2^{\ell}$ （也就是说， V 的决策树是平衡的）时，复杂度实际上会恶化（增加 3 项）。

证明。 让 Alice 和 Bob（分别作为证明者 0 和证明者 1）按照原始流程进行 k 轮辩论，生成记录 $T = (\alpha, \beta)$。在我们的新辩论中，新 Alice 额外进行 m 步，输出验证电路 C 中每个门的值；该电路在 α， β, x 的变量上计算 V。新 Bob 要么指出声称的输出值 1（由此得出 V 本应接受，这与 Alice 的目标相反），从而使新 V 接受；要么新 Bob 指出某个门的位置，而该门的输出并非由其输入得出。

新验证者 V 读取 Bob 的输出（需要进行 log(m) 次查询），其中指定了一个门索引。随后，它查询 Alice 声称的该门及其两个输入的值（额外进行 3 次查询）。由于所有参与方都知道电路结构，验证者可以确定门的类型，并检查 Alice 的输出是否符合逻辑表。如果 Bob 指出的门是电路的输入，那么验证者就会查询原始记录中相应的位置。

如果 Alice 的门实现有误，验证者就会相信新 Bob 的说法，认定 Alice 确实出错了。否则，验证者会认定模拟 Alice 的新 Alice 是正确的，并判 Alice 获胜。

正确性可由与定理 6 相同的论证得出：对于 f 的 0 输入，如果 Alice 如实进行电路求值，新 Bob 就无法指出错误；新验证者读取输出值后会判 Alice 获胜，这正是应有的结果。对于 f 的 1 输入，原始 V 必须计算出 1，因此 Bob 应当能够指出一个出错的门（或者指出输出门计算出 1），从而使新-V 无论新 Alice 是否可能作弊，都会投票支持 Bob。 □

这个引理很有力，因为它适用于任何多项式时间验证器：如果一个函数可以通过由多项式规模电路验证的辩论来计算，那么它就具有对数 DQC，不过该函数的查询复杂度可能大到与输入规模 n相同。现在我们用这一点来刻画查询复杂度为 $O(\log n)$ 的函数类。

4.2 PSPACE/poly 的刻画

定理 9。

$$\mathsf{PSPACE/poly} = \{f : DQC(f) \leq O(\log n)\}$$

证明分两步，分别建立两个包含关系。正向证明使用交叉质询引理：由于 PSPACE 有多项式时间验证器，我们可以将其压缩为对数次查询。反向证明则表明，对数查询复杂度意味着辩论可以压缩至多项式长度，并借助辅助信息进行多项式时间验证。

引理 10。 PSPACE/poly $\subseteq \{f\colon DQC(f) \leq O(\log\ n)\}$

证明。 PSPACE 可以通过多项式长度的辩论和多项式时间验证器来判定[1] [1]。这可以推广到非一致情形，即 PSPACE/poly 可以通过多项式长度的辩论来判定，并由多项式规模电路进行评判。

根据交叉质询引理（引理 8），可以通过以下方式验证这场辩论：

$$DQC(f) \leq \log(\text{size}(C_n)) + 3 = \log(\text{poly}(n)) + 3 = O(\log n)$$

查询。因此 PSPACE/poly $\subseteq \{f\colon DQC(f) \leq O(\log\ n)\}$。 □

[1]完备性证明概要：如果 L 属于 PSPACE，这意味着存在一台多项式空间机器，对每个输入（大小为 n）和状态 w （由 $poly(n)$ 长的字符串编码），都能在线性时间内（关于 $|w|$）给出下一个状态。因此，可以通过类似 Savitch 的过程验证计算路径（其长度可以为 $exp(n)$），从而验证是否接受；但每次查询都将路径长度缩短一半，并交由这台机器处理（利用挑战方证明者 0 指出所声称的计算路径中可能错误的那一半）。因此，经过 $poly(n)$ 时间后，我们会得到相邻状态，验证者可以对其进行验证。

引理 11. $\{f\colon DQC(f) \le O(\log n)\} \subseteq \text{PSPACE/poly}$

证明。令 $\{f_n : \{0,1\}^n \to \{0,1\},\ n \in \mathbb{N}\}$ 为一组函数，满足 $DQC(f_n) = \ell = O((\log(n))$。为证明该定理，我们将从这个集合中取一个语言，$L = \{x \in f_n^{-1}(1) \mid n \in \mathbb{N}\}$，并通过描述一个多项式时间验证器，将其放入 PSPACE/poly。该验证器使用多项式时间进行辩论，并获取多项式长度的辅助信息（此类为 PSPACE/poly [1]）。

固定任意常数 a，使得 L 中函数的 DQC 上界为 $a\log(n)$。根据引理 3，辩论长度 l 的上界现在是 $2^k = n^a$，即一个多项式。

在每个查询步骤中，验证者都会进行一次自适应查询，因此在 $a\log(n)$ 次查询中（具体取决于收到的答案），验证者只会查询 $2^{a\log(n)} - 1 < n^a$ 个不同的位置。只需给出一个从查询响应字符串（每次查询的比特输出组成的字符串）到下一个查询位置的映射，最后再给出一个从长度为 $a\log(n)$ 的查询响应到验证器输出的映射，就能完整描述验证者。这可以用一个查找表来描述，该表有 n^{a+1} 行，每个元素的长度为 $2a\log(n)$。因此，这个查找表的长度是多项式。

现在，我们的辩论协议（按 DEBATE [1] 的含义）应该已经显而易见：验证器查找表作为辅助信息提供，两名辩手模拟 DQC 中使用的辩论，验证者则利用查找表对其进行评估。

□

至此，定理 9 的证明完成。

关于计算效率的说明。 引理 10 中构造的验证者不仅查询效率高，而且计算上也很简单（具体取决于查询访问模型）：它只需照搬 Bob 给出的位置，将其作为查询字符串请求，然后修改该字符串，给出另外两个简单位置，最后执行一个 2 输入逻辑门来检查计算。如果我们讨论的是具有局部结构的图灵机等模型，那么验证者可以是一个常数深度电路。

对可扩展监督的启示。 定理 9 表明，所有辩论可判定语言都可以通过使用计算高效验证者、进行 $O(\log n)$ 次查询来验证。这为将辩论用作实用的可扩展监督机制提供了有力的理论支持：人类裁判只需检查辩论记录中的少数几个位置，而且每一步的验证逻辑都可以极其简单。因此，人力成本随 $\log(n)$ 增长，而不是随待验证任务的计算复杂度增长。

5 随机化验证器的下界

我们已在推论 4 中看到，对于依赖于所有变量的函数，$DQC(f) \ge \log n$。一般而言，确定性查询复杂度为 D 的函数可能具有随机化复杂度，甚至可能存在深度为 $o(D)$ 的零错误随机化决策树。因此，我们可以问：随机化验证器（视为随机化决策树）能否比最优确定性验证器高效得多？我们证明，对于具有意义的范围内的 PSPACE/poly 函数，情况并非如此。

我们首先需要定义随机化验证器的含义：

定义 12（随机化验证器）。设 (A, B, V) 是一个针对 f 的有效(k,ℓ)辩论系统，定义见第 2 节。也就是说，V：$\{0,1\}^{2k+n} \to \{0,1\}$，且方程 1、2 成立。设 T_R 是一个计算函数 V 的随机化决策树，进行 q 次查询。若对于每个 x 和有效交互记录 T_x，它以至少 $2/3$ 的概率正确计算 $V(T_x, x)$，则称 T_R 的错误不超过 $1/3$。

定义说明：

- 在此，我们不对 k 相对于 ℓ 的情况作任何界定，也不对验证者的计算资源作任何限制。我们仅要求函数 V 存在（以及证明者 1 和证明者 0 所对应的策略），如公式 2 和公式 1 所述。

- 证明者仍然是确定性的（因为他们都拥有无限能力）。此外，如何定义其他形式的随机化辩论，使其仍然具有说服力和实用性，还不太明确。

定理 13. 设 $f\colon \{0,1\}^n \to \{0,1\}$ 依赖于其所有变量，那么任何错误至多为 $1/3$ 的随机化验证器，在最坏情况下都必须至少进行 $\log(n) - 3$ 次查询。

证明。设 A, B, V是 f的一个有效辩论系统，其中 A, B分别是证明者 0、证明者 1，与之前相同；V: $\{0,1\}^{2k+n} \to \{0,1\}$ 是方程 1 和 2 所定义的"博弈"验证函数。

设 M是一个进行 q次查询、并在 q次查询中计算 V的随机化机器。我们声称，如果 $q < \log n - 3$，那么它造成的错误 M 大于 $1/3$。为证明这一点，我们使用姚氏原理 [6], : 具体而言，我们在有效输入（针对 V，而非针对 f）上定义一个分布 D，并证明，关于 D，任何确定性 q查询决策树造成的错误都大于 $1/3$。

回想一下，由于 f 依赖于它的所有变量，这意味着对于每个位置 $i \in [n]$，都存在仅在位置 i上不同的输入 $x, x' \in \{0,1\}^n$，使得 $f(x) \neq f(x')$。我们为每个 i固定这样一对输入，记作 $w_i, \tilde{w}_i$，其中 $f(w_i) = 0$，$f(\tilde{w}_i) = 1$。此外，为简单起见，我们假设不同 i对应的这些输入对互不相交（否则，我们会根据每个输入出现的次数为其赋予权重）。因此，我们假设总共有两个集合 $W_0 = \{w_i,\ i \in n\}$ 和 $W_1 = \{\tilde{w}_i,\ i \in n\}$，它们的大小均为 n。

对于每个 $i \in [n]$, 我们构造一份辩论记录 $T(i)$，使其同时对两个输入均有效 s w_i，$\tilde{w}_i$ 且两者仅在 位置 i处不同。

我们建议 $T(i)$，如下所示 s.

在输入 w_i 时，诚实证明者0（声称 $f(w_i) = 0$）会发送某条首条消息 α_1。令不诚实证明者 1 回复 β_1，即证明者 1 在看到 α_1 后以输入 $\tilde{w}_i$ 时会诚实发送的消息。然后我们继续这一过程：证明者 0 在输入 w_i 时，在看到 α_1，β_1 后诚实回复，产生 α_2；接着，证明者 1 在看到 α_1，β_1，α_2 后，按照输入 $\tilde{w}_i$ 时的做法回复，产生 β_2；依此类推。

更正式地说：

- α_1 是诚实证明者 0 在输入 w_i 且此前未看到任何消息时的回复
- β_j 是诚实证明者 1 在输入 $\tilde{w}_i$ 且看到前缀 α_1，....., α_j 后的回复
- α_{j+1} 是诚实证明者 0 在输入 w_i 且看到前缀 α_1，β_1，.....，α_j，β_j 后的回复

这就生成了记录 $T(i) = (\alpha,\ \beta)$。根据构造，该记录对 w_i 有效，因为证明者 0（针对 w_i的诚实证明者）遵循诚实策略，而证明者 1 则任意偏离。要证明同一个$T(i)$对 $\tilde{w}_i$也有效，只需交换双方的角色。对于输入 $\tilde{w}_i$，诚实的证明者 1 会恰好生成序列β （根据我们的构造），而证明者 0 则根据构造被迫生成 α 。因此， $T(i)$对$\tilde{w}_i$也有效。

现在我们可以定义分布 D ——它在大小为 $2n$的集合 $W_0 \cup W_1$ 上均匀分布，其中对于每个w_i，$\tilde{w}_i$ 我们都附加记录 $T(i)$。

设 M_q是任意一棵确定性决策树，在输入 $(\alpha,\ \beta,\ x)$上最多进行 q次查询，并计算V。我们声称，如果 $q < \log\ n - 3$，那么对于按照分布 D选取的输入，它出错的概率大于 $1/3$ M_q。

我们首先注意到，如果 M_q在输入 $(T(i)\ w_i)$ 上不查询 x_i，那么它在 $(T(i)\ \tilde{w}_i)$ 上也不会查询 x_i。因此，特别是在这种情况下，上述两个输入都会到达决策树 M_q 的同一个叶节点。由此可知，这两个输入中至少有一个会得到错误答案。

现在，由于 $q < \log -3$，用于 V 的确定性决策树 M_q 总共包含少于 $n/8$ 次查询（涵盖所有可能输入下的所有可能运行），尤其是，其中作为查询的 x_i 少于 $n/8$ 个。由此可知，对于至少 $7n/8$ 个 $i \in [n]$，这对输入 $((T(i)w_i)$，$(T(i)\ \tilde{w}_i)$ 都会到达 M_q 的同一个叶节点。因此， M_q 至少会对 D 所支持的 $7n/16$ 个输入给出错误答案，导致错误率至少为 $\frac{7}{16} > \frac{1}{3}$。 □

这个定理表明，PSPACE 的 $O(\log(n))$ 界无法得到实质性的改进，因为（例如）奇偶函数的随机化查询复杂度无法改进到优于 $\log(n) - 3$，而且它显然属于 PSPACE。

最后，我们可以问：随机化查询复杂度给定时，函数的确定性查询复杂度是多少?

定理 14. 设(A, B, V)是 f 的一个有效(k, ℓ)辩论系统，假设 f 的随机化查询复杂度为 q，则确定性辩论查询复杂度为

$$DQC(f) < q + \log(2k + n) + 10$$

粗略地说，这个定理的证明方法是找出一组确定性决策树，使它们在进行多数表决时，对所有点都能正确实现 V。然后，我们将这些确定性决策树转换为电路，并进行多数表决，从而得到一个辩论协议，其确定性验证器通过交叉质询给出结果。

证明。 根据纽曼定理 [7], ，我们可以通过对 $12(2k + n)$ 棵确定性决策树进行多数表决，确定性地实现随机决策树验证器，因为输入长度至多为 $2k + n$。这些树都是深度为 q 的分支二叉树，因此可以用一个含 $3 \cdot 2^q$ 个门的电路实现；再对 $12(2k + n)$ 个变量进行多数表决，需要额外增加 $60(2k + n)$ 个门，从而得到一个含 $12(2k + n)(3 \cdot 2^q + 5)$ 个门的确定性验证器电路。

根据引理 8（交叉质询），若一个辩论可以由规模为 m 的电路验证，则其有 $DQC \leq \log(m) + 3$，因此（假设 $q \geq 1$）：

$$DQC(f) \leq \log(12(2k + n)(3 \cdot 2^q + 5)) + 3$$

$$\leq q + \log(2k + n) + 3 + \log(66)$$

□

6 结论

我们提出了辩论查询复杂度（DQC），用以衡量辩论协议中的人类监督成本。我们的结果表明，依赖所有变量的函数需要 $\Omega(\log n)$ 次查询；而基于电路的上界则表明，$DQC(f) \leq O(\text{depth}(C_f))$，以及 $DQC(f) \leq O(\log(\text{size}(C_f)))$。我们的主要结果将 PSPACE/poly 精确刻画为可通过 $O(\log n)$ 次查询判定的函数类。

这一对数界具有明确的实际意义：对于大小为一百万比特的输入，无论所判定问题的计算复杂度如何，验证都只需约 20 次查询。此外，验证电路可以极其简单。这为将辩论作为一种可扩展的监督机制提供了有力的理论支持。

我们的技术（改造 Karchmer-Wigderson 博弈并将交叉质询形式化）将辩论与成熟的复杂性理论领域联系起来，尤其是通信复杂度和电路复杂度。这表明，DQC 或可成为研究人工智能对齐及复杂性基本问题的一种有用视角。

开放问题

计算现实主义与实践对齐

- **实证验证**：真实语言模型之间的辩论是否呈现出我们的理论所预测的对数查询扩展？理论分析中的哪些辩论协议能有效地应用于实践？

- **计算能力受限的辩手**：我们的结果假设辩手不受计算能力限制。Brown-Cohen 等人 [3, 4]研究了辩手受多项式限制且可能不了解验证器代码的辩论。在这些现实约束下，查询需求会如何变化？当辩手面临计算限制时，能否保留对数查询复杂度？

- **验证器不透明性**：当验证者是大语言模型或其他不透明系统时，辩手无法利用其结构。这会改变查询复杂度吗？我们能否刻画出在验证器不透明性下仍然高效的验证任务？

• **人类价值预言机**：近期研究将人类判断建模为预言机访问，而非确定性计算。将人类价值查询纳入协议，会如何影响整体查询预算?

复杂性理论

• **电路下界障碍**：推论7表明，证明任意 P 中的语言满足 $DQC(f) \geq \log n + 6$，都将带来新的电路下界，超越目前已知的最佳显式界。这意味着有两种可能：要么所有可高效计算的函数都具有极低的 DQC，要么可以利用 DQC 的下界证明出优于当前最佳结果的电路下界。当然，在后一种情况下，证明紧的DQC下界也可能面临与电路下界相同的障碍。弄清楚究竟是哪种情形，将有助于阐明查询高效辩论的根本局限。反过来（或许更令人振奋的是），只要证明单个函数的 DQC 有一个相对简单的界，就能立即在电路复杂度方面取得突破

• **超越 Karchmer-Wigderson**：我们的电路上界改编自通信复杂度中的 KW 博弈。通信复杂度中的其他哪些工具可以用来导出 DQC 界？信息复杂性、多方通信或流式算法中的技术，能否带来新的辩论协议或下界?

• **多位辩手**：增加第三位辩手（或更多）是否能降低某些函数类的查询复杂度？交替量词结构 $\exists\forall\exists\forall\ldots$ 可以自然地推广到更多参与者，但在不假设某种具有明显优势的设置的情况下，这是否能带来查询优势尚不明确。

归根结底，我们的工作为探讨辩论协议中验证的人力成本奠定了基础。对数级查询上界让人们有理由希望，从理论上讲，可扩展监督是能够实现的。如今的挑战在于确定这种理论效率能否在实践中实现，并理解限制辩论所能实现目标的根本障碍，包括计算方面和复杂性理论方面的障碍。

参考文献

[1] Geoffrey Irving、Paul Christiano 和 Dario Amodei。《通过辩论实现 AI 安全》。载于：*arXiv* 预印本 *00899 arXiv:1805*。（ 2018 年）。[2] Beth Barnes 和 Paul Christiano。《写作稿：通过辩论推进 AI 安全》。AI Alignment Forum。2020年。 URL: https://www.alignmentforum.org/posts/writeup-progress-on- ai- safety- via- debate-1。[3] Jonah Brown-Cohen、Geoffrey Irving 和 Georgios Piliouras。《通过双重高效辩论实现可扩展的 AI 安全》。载于：*arXiv* 预印本 *arXiv:2311.14125*（2023年）。[4] Jonah Brown-Cohen、Geoffrey Irving 和 Georgios Piliouras。《通过证明者—估计者辩论避免混淆》。载于：*arXiv* 预印本 *arXiv:2506.13609*（2025年）。[5] Kazuo Iwama 和 Hiroki Morizumi。《布尔电路 $5n - O(n)$ 的显式下界》。载于：国际计算机科学数学基础研讨会。Springer。2002年，第353–364页。[6] Andrew Chi-Chin Yao。《概率计算：迈向统一的复杂性度量》。载于：第十八届计算机科学基础年度研讨会（*SFCS 1977*）。IEEE 计算机学会。1977年，第222–227页。[7] Ilan Newman。《通信复杂性中的私有随机位与公共随机位》。载于：*Information processing letters* 39.2（1991年），第67–71页。