arXiv:2303.16413v1 [cs.CC] 29 Mar 2023

对数空间的认证困难性与随机性

Edward Pyne* RanRaz† Wei Zhan‡麻省理工学院 普林斯顿大学 普林斯顿大学epyne@mit.edu ranr@cs.princeton.edu weizhan@cs.princeton.edu

2023年3月30日

摘要

设 $\mathcal{L}$ 是一个可在对数空间内判定的语言，并设 $\epsilon > 0$ 是任意常数。设 $\mathcal{A}$ 是指数困难性假设：对于每个 n，对于长度为 n 的输入，判定其是否属于 $\mathcal{L}$ 不能由规模小于 $2^{\epsilon n}$ 的电路完成。我们证明，对于每个函数 $f\colon \{0,1\}^* \to \{0,1\}$，若其可由随机化对数空间算法 R 计算，则存在一个确定性对数空间算法 D（试图计算 f），使得对于每个长度为 n 的输入 x，算法 D 都会输出以下内容之一：

1. 正确的值 $f(x)$。
2. 字符串："由于困难性假设 $\mathcal{A}$ 不成立，我无法计算 $f(x)$"，后接一个（已证明正确的）电路。该电路的规模小于 $2^{\epsilon n'}$，用于判定长度为 n'的输入是否属于 $\mathcal{L}$，其中 $n' = \Theta(\log n)$；也就是说，该电路反驳了 $\mathcal{A}$。

此外，给定 R后，可以显式构造 D。

我们注意到，以往关于困难性与随机性范式的研究给出的去随机化算法会盲目依赖困难性假设。如果困难性假设不成立，这些算法可能会输出错误的值，因此用户无法相信算法给出的输出是正确的。相较之下，我们的算法 D 会验证计算，因此绝不会输出错误的值。因此，如果 D 为 $f(x)$输出一个值，就可以确认该值正确。此外，如果 D 没有为 $f(x)$输出值，它就会发出警告，指出困难性假设已被证伪，并对该假设予以反驳。

我们的下一个结果是针对 BPL（即由有界错误概率随机对数空间算法可解的问题类）的通用去随机化器[1]：我们给出一个确定性算法 U，它以一个随机化对数空间算法 R 和一个输入 x 作为输入，并确定性地模拟该算法在 x上的计算过程 R。在广泛认可的假设 BPL =L 下，U 所用的空间至多为 $C_R \cdot \log n$（其中 C_R 是一个依赖于 R的常数）。此外，对于每个常数 $c \geq 1$，如果 BPL $\subseteq$ SPACE$[(\log(n))^c]$，那么 U 所用的空间至多为 $C_R \cdot (\log(n))^c$。

最后，我们证明，如果存在有序分支程序的最优命中集，那么就存在一个确定性对数空间算法，它能够在给定有序分支程序 B 的黑盒访问权限及其大小 n的情况下，估计该程序 B 在均匀随机输入上接受的概率。这扩展了 Cheng 和 Hoza 在 CCC 2020 的结果；他们证明了最优命中集意味着白盒双边去随机化。

关键词：伪随机性，空间受限计算

*由 Akamai 总裁奖学金资助。本研究的部分工作是在访问 Simons 元复杂性研究所项目期间完成的。

†由 Simons 研究员奖和美国国家科学基金会 CCF-2007462 号资助。

‡由 Simons 研究员奖和美国国家科学基金会 CCF-2007462 号资助。[1]我们的结果针对 promise-BPL 陈述并证明，但在摘要中我们忽略了这一差异。

1 引言

在最近的一项工作中，Girish、Raz 和 Zhan 研究了不可信随机性 [GRZ23] 的能力。他们的主要观察之一是，随机化对数空间计算仅使用 $O(\log n)$ 个随机比特即可被验证。更准确地说，BPL 中的每个问题都具有一个流式证明，该证明在随机化对数空间证明者和随机化对数空间验证者之间进行，其中验证器仅使用 $O(\log n)$ 个随机比特，并且对证明者流式传输的证明具有一次性单向访问权限。换言之，证明者提供一份长度为多项式的证明，并将其流式传输给验证器；验证器仅使用 $O(\log n)$ 个随机比特，就能检查计算是否被正确执行。

这引出了以下耐人寻味的可能性。尝试用 π 的数字来替代证明者的随机字符串。大多数情况下，这应该能够奏效，计算也应该能够正确执行，因为 π 的数字似乎与大多数计算无关。在极少数计算未能正确执行的情况下，验证器会发现这一点，因为验证以很高的概率会失败，因此不会造成任何问题。此外，由于 π 的数字可以在小空间内确定性地生成，证明者现在是确定性的，因此验证器可以完整地模拟证明者。由于验证器只使用 O（$\log n$）个随机比特，验证器只需尝试这些随机比特的所有可能性，这样验证器也变成了确定性的[2]，，从而整个交互过程现在都可以由确定性对数空间算法进行模拟。

这种方法无法将所有随机化对数空间计算去随机化，因为 π 的数字可以由一个小空间算法生成。π 的数字并不是为欺骗随机化计算而设计的。下一个合乎逻辑的步骤，是尝试使用那些为欺骗随机化计算而设计的序列，即伪随机生成器的候选构造，例如基于困难性与随机性范式的伪随机生成器 [Sha81,Yao82,BM84,NW94, IW97, STV01, KvM02]。假设某些广泛认可的困难性假设成立，这类伪随机生成器就能在某个特定复杂度类内欺骗随机化计算。

设 G 是一种伪随机生成器的候选构造，旨在骗过随机化对数空间计算，并假设 G 使用对数空间和 $O(\log n)$ 个随机比特。我们可以尝试用由 G 生成的伪随机序列替代证明者的随机字符串。由于这样一来，证明者和验证器都只使用对数数量的随机比特，验证器便可以通过一个确定性对数空间算法模拟整个交互。如果生成器的 $O(\log n)$ 个随机比特的 poly(n) 种可能取值中，有一种能产生有效证明，证明该计算是正确执行的，那么验证器就会发现这一点并接受该计算。如果 poly(n) 种可能取值都失败，验证器就会发出警告，指出生成器失效。因此，该算法永远不会输出错误的值。

如果生成器 G 基于困难性与随机性范式，那么生成器失效就意味着其所依据的困难性假设是错误的。此外，基于困难性与随机性范式的证明通常具有构造性，也就是说，它们表明，如果生成器失效，就可以构造一个反驳该困难性假设的电路。如果我们能证明，构造该电路可以通过确定性对数空间完成，那么验证器就能获得一个反驳 G 所依据的困难性假设的电路。

我们采用 Klivans and van Melkebeek 的困难性与随机性伪随机生成器的一个变体 [KvM02]，该变体基于 [NW94、IW97、STV01]，以在（假设指数困难性假设的前提下）对 BPL 进行去随机化。基于该生成器，我们得到如下结果。

[2]当证明者采用随机化方式，或验证器无法模拟证明者时，通过尝试验证器随机比特的所有可能性来去随机化验证器是不可行的，因为验证器需要多重访问证明者的输出才能做到这一点。

定理 1.1。设 $\mathcal{L}$ 为一种可在线性空间内判定的语言，且 $\epsilon > 0$ 为一个常数。设 $\mathcal{A}$ 为如下指数困难性假设：对于每个 n，长度为 n 的输入，其是否属于 $\mathcal{L}$ 不能由规模小于 $2^{\varepsilon n}$的电路判定。设 $f : \{0,1\}^* \to \{0,1\}$ 为一个可由随机化对数空间算法 R计算的函数。则存在一个确定性对数空间算法 D（可由 R显式给出），使得对于每个长度为 n的输入 x，该算法D 输出以下之一：

1. 正确的值为 $f(x)$。

2. 字符串："由于困难性假设 $\mathcal{A}$ 不成立，无法计算 $f(x)$"，后面跟着一个规模小于 $2^{\varepsilon n'}$ 的（已证明正确的）用于判定输入长度为 n'的输入是否属于 $\mathcal{L}$ 的电路，其中某个 $n' = \Theta(\log n)$；也就是说，这是一个推翻 $\mathcal{A}$的电路。

换言之，尽管先前所有基于困难性与随机性范式的去随机化结果所给出的算法都盲目依赖困难性假设，并且在困难性假设不成立时可能输出错误值，但我们的算法 D 从不会输出错误值：如果困难性假设成立，D 总是输出正确的值 $f(x)$。如果困难性假设不成立，D 仍然会输出正确的值 $f(x)$，或者发出困难性假设不成立的警告，并推翻该假设。

特别地，如果定理 1.1中使用的困难性假设成立（而且有几个这样的假设被广泛认为是成立的），定理 1.1就给出了一个确定性对数空间算法，该算法总是输出 $f(x)$的正确值，并且该值已得到正确性证明。从这个意义上说，如果困难性假设成立，定理 1.1给出的算法实际上充当了空间复杂度类 BPL的完整去随机化算法。

我们注意到，在以往的工作中，所谓的重构步骤，即构造一个反驳困难性假设的电路（当生成器失效时），需要在多个地方使用随机性，并且已知无法在对数空间内计算。我们在定理 1.1的证明中的主要技术贡献，是精心设计伪随机生成器，并证明对于该生成器，重构步骤的所有部分都可以在确定性对数空间内完成。我们将重构可以在确定性对数空间内完成这一结果视为我们工作的另一项独立贡献。

让我们回到这一观察：BPL 中的每个问题都有一个流式证明，该证明介于随机化对数空间证明者和随机化对数空间验证器之间，其中验证器仅使用$O(\log n)$个随机比特，并且以只读一次的单向方式访问证明者流式传输的证明 [GRZ23]。该证明基于这样一种协议：证明者计算并流式传输到达随机化算法所对应的分支程序中每个状态的概率，而验证器则检查这些概率在每两个连续时间步之间是否一致。

虽然我们可以用这种方法证明定理 1.1，但这里我们给出一个略有不同且更直接的证明：通过验证伪随机生成器输出的每个比特在以到达底层分支程序的每个状态为条件时，其分布都接近均匀分布，来完成验证。这些条件概率是通过检查伪随机生成器的所有可能输出直接计算的。这是可行的，因为生成器只使用对数数量的随机比特，因此可能性数量是 n 中的多项式。这种方法与 Nisan [Nis93], 的工作相关；他采用类似的方法来检查给定的多项式规模字符串集合是否具有足够的随机性，从而能够高精度地模拟随机化计算，并据此证明 BPL $\subset$ ZP*L（其中 ZP*L 是零错误随机化对数空间，机器可以双向访问随机带）。

上述讨论表明，可以验证候选伪随机生成器 G （它使用对数空间和 $O(\log n)$个随机比特）的输出是否具有足够的随机性

对于给定的随机对数空间计算。考虑到这一点，一个自然的想法是通过穷举搜索所有可能的生成器（利用生成器由一台常数规模的图灵机描述这一事实），找到一个对于给定的随机对数空间计算足够好的伪随机生成器。最终目标是获得一个通用去随机化器，其效果至少不逊于最好的伪随机生成器。

我们探索了这一思想，并发现可以证明一个更强的结果。我们明确构造了一个针对 prBPL（promise-BPL，即可由有界错误概率随机对数空间算法求解的承诺问题类）的通用去随机化器 U，它在 prBPL 上以最佳可能的确定性空间界运行。

更准确地说，我们给出一个确定性算法 U，它以一个随机化对数空间算法 R 和一个输入 x 作为输入，并模拟 R 在 x 上的计算。在广泛认可的假设 prBPL $=$ L 下，U 使用的空间至多为 $C_R \cdot \log n$（其中 C_R 是取决于 R 的常数）。更一般地，对于每个常数 $c \geq 1$，如果 prBPL $\subseteq$ SPACE[$(\log(n))^c$]，那么 U 使用的空间至多为 $C_R \cdot (\log(n))^c$。我们强调，这里的关键在于，U 是确定性的，并且是明确给出的，而不是一个存在性结果。我们指出，在时间有界情形下，类似的结果尚不为人所知，而且似乎很难得到。我们还指出，目前已知的 BPL 上的最佳空间界是 prBPL $\subseteq$ SPACE[$(\log(n))^{1.5-o(1)}$] [SZ99, Hoz21]。

定理 1.2。设 U 为第 5 节中明确给出的确定性算法，该算法以随机化对数空间算法 R 和输入 x 作为输入。假设 R 在 x 上接受的概率要么为 $\leq 1/4$，要么为 $\geq 3/4$。那么，如果 R 在 x 上接受的概率为 $\leq 1/4$，则 U 在输入 R、x 上的输出为 0；如果 R 在 x 上接受的概率为 $\geq 3/4$，则 U 在输入 R、x 上的输出为 1。此外，对于每个常数 $c \geq 1$，如果 prBPL $\subseteq$ 空间复杂度类[$(\log(n))^c$]，则 U 使用的空间至多为 $C_R \cdot (\log(n))^c$，（其中 C_R 是一个取决于 R 的常数）。

我们注意到，在定理 1.2 中，U 所使用的空间也可以界定为 $C \cdot (\log(N))^c$，其中 C 是一个通用常数，N 是底层分支程序的长度和宽度的上界，该分支程序实现了 R 在 x 上的计算（假设 prBPL $\subseteq$ 空间复杂度类[$(\log(n))^c$]）。（参见定理 5.1）。

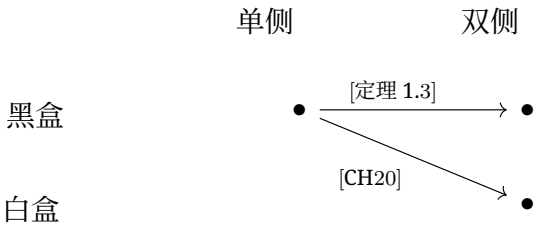
另一项与我们的工作以及 [Nis93,GRZ23], 相关的先前工作，是 Cheng 和 Hoza 的工作 [CH22]。Cheng 和 Hoza 证明了，对于对数空间，最优命中集生成器（伪随机生成器的单侧类比）将推出 BPL $=$ L（而这种命中集生成器的直接结论只能是 RL $=$ L）[CH22]。为证明这一结果，他们展示了如何利用命中集生成器来猜测到达分支程序中每个状态的概率（的近似值），然后检查这些概率在每两个连续时间步之间是否一致（与 [GRZ23] 类似，且早于该工作）。

Cheng 和 Hoza 给出的证明使用了底层分支程序的显式描述。我们的最终结果将他们的结果扩展到了分支程序并非显式给出的情形，而是只能对其进行预言机访问，也就是说，将其作为黑盒访问。

定理 1.3（非正式表述：正式表述及证明见第 6 节）。假设存在针对宽度为 n、长度为 n的有序分支程序的最优显式命中集生成器。那么，针对宽度为 n、长度为 n的有序分支程序的最优确定性采样器（具有对该分支程序的预言机访问）也存在。

我们指出，Cheng 和 Hoza [CH22] 证明了这一结果的一个版本，适用于常数宽度分支程序（此外，他们还给出了关于长度为 n、宽度为 n的程序的非黑盒结果，这类程序刻画了 BPL）。他们将 BPL 与 L 情形下的黑盒等价性列为一个开放问题，而我们解决了这一问题。我们的结果补充了 BPP 与 P 情形下的等价结果；此前已有若干

结果 [ACR96、BF99、ACRT99、GVW11、CH22] 表明，一般电路的命中集可以推出一般电路的确定性采样器。因此，对于这一问题，我们弥合了时间受限去随机化与空间受限去随机化之间的理解差距。



我们希望，我们的进展最终能够用于在白盒情形下得到一个等价关系，即 $\mathrm{prRL} = \mathrm{L} \implies \mathrm{prBPL} = \mathrm{L}$。 [BF99] 在时间有界情形下证明了这一结果。

我们所有结果的一个共同主题是，我们的证明利用并进一步展示了一个有趣的观点：在某些情况下，随机对数空间计算可以被验证。

1.1 相关工作

过去四十年来，研究者一直致力于对随机对数空间计算进行去随机化，也就是证明 $\mathrm{BPL} = \mathrm{L}$。这项工作主要采用了（至少）两种形式：构造伪随机生成器（PRG）及其推广 [Nis90, INW94,NZ96, GR14, FK18,MRT19,HZ20]，以及白盒去随机化 [SZ99, RR99, Rei08, RTV06, AKM+20, Hoz21]。由此形成了多元的研究图景，包括明确构造的 PRG，其种子长度虽取得了高度非平凡但（据推测）并非最优的结果；白盒去随机化；以及候选构造。我们强调，这些候选构造既包括安全性可由某种困难性假设推出的生成器 [KvM02], ，也包括其安全性尚未被证明可由困难性假设推出的候选方案（例如，Reingold 和 Vadhan 曾提出将两个小偏差分布异或作为候选方案 [LV17]）。

如上所述，除了 [KvM02], 之外，与我们的工作最相关的是 [Nis93,CH22,GRZ23]。这些工作都包含对随机化计算是否正确执行的某种形式的验证（目的各不相同），这一思想也是我们工作的核心。

2 预备知识

我们首先定义与伪随机生成器和分支程序有关的记号 s.

定义 2.1. 给定空间 $[S]$上的一个分布 D，令 $x \leftarrow D$ 表示从 D中抽取 $x \in [S]$。令 U_n 表示 $\{0,1\}^n$上的均匀分布。

定义 2.2. 给定一个伪随机生成器（PRG） $G : \{0,1\}^s \to \{0,1\}^n$ 和一个函数f: $\{0,1\}^n \to \mathbb{R}$，我们用 $\mathbb{E}[f(U_n)]$和 $\mathbb{E}[f(G(U_s)]$ 分别表示在均匀分布的输入和由 G 生成的伪随机输入下 f的期望，即:

$$\mathbb{E}[f] = \mathop{\mathbb{E}}_{x \leftarrow U_n}[f(x)], \qquad \mathop{\mathbb{E}}_{G}[f] = \mathop{\mathbb{E}}_{y \leftarrow U_s}[f(G(y))].$$

我们称 G ε愚弄了 f，如果 $|\mathbb{E}[f] - \mathbb{E}_G[f]| \leq \varepsilon$。

定义 2.3。长度为 n 、宽度为 w 的有序分支程序（OBP） B 是一个有向无环图，其顶点（或状态）被划分为 $n+1$ 层 $V_0, \ldots, V_n$，其中 $|V_i| \leq w$。对于每个 $i < n$ 和 $v \in V_i$，都有两条出边，分别标记为 0 和 1，并通向 V_{i+1}。 V_0 只包含一个状态 v_0，它是起始状态； V_n 中的每个状态都标记有一个实数，作为分支程序的输出。除非另有说明，我们假设这些标签为 0 或 1。

对于每个 $v \in V_i$、 $\sigma \in \{0,1\}^k$ 和 $u \in V_{i+k}$，如果 B 沿着由 σ 中比特标记的边从状态 v 转移到状态 u，我们就称 $B[v,\sigma] = u$。我们可以将 B 视为定义在 $\{0,1\}^n$ 上的函数，使得对于每个 $x \in \{0,1\}^n$， $B(x)$ 是输出状态 $B[v_0, x]$ 上的标签。对于每个 $v \in V_i$，令 $B_{\to v}$ 为一个长度为 i 、宽度为 w 的有序分支程序，使得 $B_{\to v}(x) = 1$ 当且仅当 $B[v_0, x_{1..i}] = v$。

对于每个 $v \in V_i$，令

$$p_{\to v} = \Pr[B[v_{st}, U_i] = v], \qquad p_{v\to} = \Pr[B[v, U_{n-i}] = v_{\text{acc}}].$$

3 从困难性到随机性的有效方法

我们分几个阶段证明定理 1.1。在第一阶段，我们展示一个测试过程：给定一个候选伪随机数生成器和一个有序分支程序，该过程要么证明该伪随机数生成器能够欺骗该分支程序，要么输出一个充当 G 的下一位预测器的分支程序。随后，我们展示如何从这样的下一位预测器得到困难性假设的反例。

3.1 可验证的对数空间伪随机生成器

我们首先证明，存在一个针对对数空间有序分支程序的伪随机生成器（种子长度为对数长度）的对数空间验证器，它能够检测伪随机生成器何时失效，并输出一个伪随机生成器无法欺骗的有序分支程序实例。为形式化这一点，我们先回顾下一比特预测器的概念。

定义 3.1. 给定一个函数 $G\colon \{0,1\}^s \to \{0,1\}^n$，若分支程序 $T\colon \{0,1\}^i \to \{0,1\}$for $i < n$ 是针对 G 的 ε-下一比特预测器，则 $\Pr_{x \leftarrow U_s}[T(G(x)_{1..i}) = G(x)_{i+1}] > 1/2 + \varepsilon$。

注意，即使对于计算能力不受限制的区分器，均匀分布也是 0-下一比特可预测的。

我们将在本节证明以下引理：

引理 3.2. 对于每个可在空间 $O(\log n)$ 内计算的误差函数 $\varepsilon(n)$，都存在一个确定性算法。该算法以长度为 n 、宽度为 w的有序分支程序 B 以及对伪随机生成器 $G\colon \{0,1\}^s \to \{0,1\}^n$的黑盒预言机访问作为输入，在空间 $O(s + \log(nw))$ 内运行，并且满足以下条件之一：

1. 确认 G $\varepsilon \cdot n$能够欺骗 B；或者
2. 输出一个 长度至多为 n 、宽度为 w 的 OBP T，它是针对 G的 $\varepsilon/2$-下一位预测器。

该证明背后的主要思想此前已因不同目的而出现过 [Nis93、CH22、GRZ23], ；事实上，所有这些结果的（修改版本）都可用于证明引理 3.2。不过，我们给出一个自包含的证明。

为了证明引理 3.2，我们首先定义一系列潜在区分器，它们都具有可在对数空间内求值的性质。每个区分器都用于测量 PRG 到达某个特定状态时下一比特的偏差。

定义 3.3. 给定一个长度为 n 的有序分支程序 B，对于每个 $i < n$ 和 $v \in V_i$，令 N_v：$\{0,1\}^{i+1} \to \{-1,0,1\}$ 为如下定义的函数：

$$N_v(x) = \begin{cases} 1 & \text{if } B_{\to v}(x) = 1 \text{ and } x_{i+1} = 1 \\ -1 & \text{if } B_{\to v}(x) = 1 \text{ and } x_{i+1} = 0 \\ 0 & \text{otherwise.} \end{cases}$$

此外，给定 B 和 v，N_v 可在对数空间中计算。

当 x 为均匀随机时，$B_{\to v}(x)$ 与 x_{i+1} 相互独立，因此对所有 v都有 $\mathbb{E}[N_v] = 0$。因此，我们的验证器检验对所有 v，$|\mathbb{E}_G[N_v]|$ 是否很小，其中我们将 PRG 输出的前 $i+1$ 个比特输入给 N_v。我们首先证明其可靠性：

引理 3.4. 给定一个长度为 n的有序分支程序 B，假设对每个 i，都有 $\sum_{v \in V_i} |\mathbb{E}_G[N_v]| \le \varepsilon$。则 G $\varepsilon \cdot n$-欺骗 B。

证明。由于从第 V_i 层出发的每条边都进入第 V_{i+1}层，因此对每个 $i < n$，我们有

$$\begin{aligned} &\sum_{v \in V_{i+1}} \left| \mathbb{E}[B_{\to v}] - \mathop{\mathbb{E}}_{G}[B_{\to v}] \right| \\ \le &\sum_{v \in V_i} \sum_{b \in \{0,1\}} \left| \Pr_{x \leftarrow U_n}[B_{\to v}(x) = 1 \wedge x_{i+1} = b] - \Pr_{x \leftarrow G(U_s)}[B_{\to v}(x) = 1 \wedge x_{i+1} = b] \right|. \end{aligned}$$

注意，由 N_v的定义，我们有

$$\begin{aligned} \mathbb{E}[N_v] &= \Pr_{x \leftarrow U_n}[B_{\to v}(x) = 1 \wedge x_{i+1} = 1] - \Pr_{x \leftarrow U_n}[B_{\to v}(x) = 1 \wedge x_{i+1} = 0] \\ &= 2 \Pr_{x \leftarrow U_n}[B_{\to v}(x) = 1 \wedge x_{i+1} = 1] - \mathbb{E}[B_{\to v}] \\ &= \mathbb{E}[B_{\to v}] - 2 \Pr_{x \leftarrow U_n}[B_{\to v}(x) = 1 \wedge x_{i+1} = 0], \end{aligned}$$

并且上述结论在由 G生成的伪随机性下同样成立。因此，我们进一步有

$$\begin{aligned} \sum_{v \in V_{i+1}} \left| \mathbb{E}[B_{\to v}] - \mathop{\mathbb{E}}_{G}[B_{\to v}] \right| &\le \sum_{v \in V_i} \left| \mathbb{E}[B_{\to v}] - \mathop{\mathbb{E}}_{G}[B_{\to v}] \right| + \sum_{v \in V_i} \left| \mathbb{E}[N_v] - \mathop{\mathbb{E}}_{G}[N_v] \right| \\ &= \sum_{v \in V_i} \left| \mathbb{E}[B_{\to v}] - \mathop{\mathbb{E}}_{G}[B_{\to v}] \right| + \sum_{v \in V_i} \left| \mathop{\mathbb{E}}_{G}[N_v] \right|. \end{aligned}$$

根据假设 $\sum_{v \in V_i} |\mathbb{E}_G[N_v]| \le \varepsilon$ 和事实 $\mathbb{E}[B_{\to v_0}] = \mathbb{E}_G[B_{\to v_0}] = 1$，我们得出结论 $\sum_{v \in V_n} |\mathbb{E}[B_{\to v}] - \mathbb{E}_G[B_{\to v}]| \le \varepsilon \cdot n$。由于输出标签是二进制，这意味着 $|\mathbb{E}[B] - \mathbb{E}_G[B]| \le \varepsilon \cdot n$，即 G $\varepsilon \cdot n$-欺骗 B。 □

引理 3.2 的证明。对于每个 $i < n$，算法遍历每个 $v \in V_i$ 以及 G 的所有可能种子，计算 $\sum_{v \in V_i} |\mathbb{E}_G[N_v]|$ 并检查其是否至多为 ε。这可以在空间复杂度类 $O(s + \log(nw))$ 中完成。如果所有此类检查均通过，则根据引理 3.4，G $\varepsilon \cdot n$ 可以欺骗 B。

否则，我们找到某个 $i < n$，使得 $\sum_{v \in V_i} |\mathbb{E}_G[N_v]| > \varepsilon$。令 T 为一个长度为 i 的有序分支程序，它从第 V_0 层到第 V_i 层与 B 相同，使得每个 $v \in V_i$ 上的输出标签在 $\mathbb{E}_G[N_v] \ge 0$ 时为 1，

在 $\mathbb{E}_G[N_v] < 0$ 时为 0。这样的有序分支程序的大小至多与 B 相同，并且可以在空间 $O(s + \log nw)$ 中构造。我们有

$$\begin{aligned}
&\Pr_{x \leftarrow G(U_s)}[T(x_{1..i}) = x_{i+1}] \\
&\quad = \sum_{\substack{v \in V_i \\ \mathbb{E}_G[N_v] \geq 0}} \Pr_{x \leftarrow G(U_s)}[B_{\to v}(x) = 1 \wedge x_{i+1} = 1] + \sum_{\substack{v \in V_i \\ \mathbb{E}_G[N_v] < 0}} \Pr_{x \leftarrow G(U_s)}[B_{\to v}(x) = 1 \wedge x_{i+1} = 0] \\
&\quad = \sum_{v \in V_i} \frac{1}{2} \left(\mathop{\mathbb{E}}_{G}[B_{\to v}] + \left| \mathop{\mathbb{E}}_{G}[N_v] \right| \right) \\
&\quad > \frac{1}{2}(1 + \varepsilon).
\end{aligned}$$

□

3.2 对数空间中的可证伪困难性假设

引理 3.2 表明，给定一个声称适用于对数空间的伪随机生成器，我们可以利用它来成功地对一次对数空间计算去随机化，或者明确输出一个针对该伪随机生成器的反例。困难性与随机性范式的结果表明，在某些困难性假设下，伪随机生成器是存在的。将这些结果与引理 3.2 结合起来，给定任意声称的困难函数，我们可以对数空间计算进行去随机化，或者判定该困难性假设不成立。然而，定理 1.1 要求算法提供更强的保证——如果困难性假设不成立，算法需要输出一个能够证伪该假设的小型电路。获得这一结果是本小节的主要贡献。

我们首先回顾 Klivans and van Melkebeek [KvM02] 的结果。

定理 3.5（[KvM02]）。如果存在一个布尔函数族 $f \in \text{SPACE}[n]$，对于某个 $\varepsilon > 0$，该函数族无法由大小为 $2^{\varepsilon n}$ 的电路计算，则 $\text{BPL} = \text{L}$。

他们的证明基于 Imagliazzo and Wigderson 关于最坏情况难度与随机性的结果 [IW97]，，并说明了如何在确定性对数空间中执行 Imagliazzo-Wigderson 伪随机生成器构造中的每一步。然而，他们的证明（以及其他所有遵循困难性与随机性范式的证明）并未表明：给定一个能够区分该伪随机生成器与随机分布的分支程序（或电路）（即反驳原始困难性假设），是否存在一种高效的确定性对数空间算法，能够为所谓的困难函数生成一个电路。这有两个原因。首先，将区分器转换为下一比特预测器的过程存在问题（我们将在引理 3.2 中讨论）。即使获得了这样的预测器，先前的方法仍使用在空间和随机性方面效率低下的概率方法论证，从预测器得到一个对原函数最坏情况正确的电路。本小节的主要贡献是仔细设计伪随机生成器，并针对所构造的伪随机生成器给出一个区分器，进而开发出高效的重构过程。

这引出了以下定理：

定理 3.6。对于每个布尔函数族 $f \in \text{SPACE}[n]$ 和 $\varepsilon > 0$，存在一种确定性算法：给定一个长度为 n、宽度为 $w = n$ 的 OBP B 作为输入，该算法在空间 $O(\log n)$ 中运行，并且执行以下操作之一：

1. 输出 $\mathbb{E}[B]$，且错误为 $1/4$；或
2. 输出一个大小为 $2^{\varepsilon m}$ 的电路 $\mathcal{C}$，它在 $\{0,1\}^m$ 上计算 f，其中 $m = \Theta(\log\ n)$。

证明。令 $G\colon \{0,1\}^s \to \{0,1\}^n$ 为定理 4.1 中的生成器，其中包含 $\varepsilon = \varepsilon$ 和 $f = f$，并令 $m = m_0$ 为用于构造 G 的 f 的实例大小。

然后，我们将引理 3.2 应用于 B 和 G，其中使用 $\varepsilon = 1/(4n)$。两种可能的结果如下：

1. 如果已证明 $G\ \varepsilon \cdot n$ 能够欺骗 B。在这种情况下，该算法计算并输出 $\mathbb{E}_G[B]$，其与 $\mathbb{E}[B]$ 的加性误差在 $1/4$ 以内。

2. 否则，对于某个 $i < n$，我们得到一个长度为 i、宽度为 w 的显式有序分支程序 T，使得 $\Pr_{x \leftarrow U_s}[T(G(x)_{1..i}) = G(x)_{i+1}] > \frac{1}{2}(1+\varepsilon)$。换句话说，$T$ 是一个针对 G 的 $\varepsilon/2 = 1/8n$ 下一位预测器，大小至多为 n^2，并且 T 可以在空间 $O(\log n)$ 内求值。于是根据定理 4.1，我们可以在空间 $O(\log n)$ 内构造一个电路 C，用于处理输入规模为 $m = \Theta(\log n)$ 的 f，其大小至多为 $2^{\varepsilon m}$。 □

现在定理 1.1 得证：

定理 1.1 的证明。给定一个错误概率至多为 $1/10$ 的随机化对数空间算法 R 以及输入 $x \in \{0,1\}^n$，令 B 为表示 R 在输入 x 上如何使用其随机比特的分支程序，该程序可在对数空间内构造。根据假设，R 使用 $s = O(\log n)$ 比特的空间复杂度，因此至多使用 $2^{s+O(1)} = \text{poly}(n)$ 个随机硬币，从而 B 的长度和宽度均为 $\text{poly}(n)$。将 B 填充至长度和宽度均为 n^c，并对 $f = \mathcal{L}$ 和 ε 应用定理 3.6：$= \varepsilon$。于是，我们要么得到 $\mathbb{E}[B]$ 的一个精度达到 $\pm 1/4$ 的估计值（根据 R 的正确性，这足以判定该语言），要么得到一个反例（其形式为一个大小至多为 $2^{\varepsilon n'}$ 的电路），从而反驳 $\mathcal{L}$ 在大小为 $n' = \Theta(\log n^c) = \Theta(\log\ n)$ 比特的输入上的困难性，正如所需。 □

我们将在下一节证明定理 4.1。

4 高效重构去随机化

我们首先陈述本节的主要定理。

定理 4.1。给定 $\varepsilon > 0$ 以及函数族 $f_m\colon \{0,1\}^m \to \{0,1\} \in \text{SPACE}[m]$，存在函数族 $G\colon \{0,1\}^s \to \{0,1\}^n$，其 $s = O(\log\ n)$ 可在空间 $O(\log\ n)$ 内计算；并且存在一个确定性对数空间算法：给定 $n \in \mathbb{N}$ 和一个针对 G 的 $1/(8n)$-下一位预测器 $\mathcal{B}$，该预测器的规模至多为 n^2，且可在空间 $O(\log n)$ 内求值，该算法输出一个规模为 $2^{\varepsilon \cdot m_0}$ 的电路 C，用于 f_{m_0}，并满足 $m_0 = \Theta(\log n)$。

我们分四个阶段证明该定理。遵循 [IW97]，的框架，我们首先假设 f 是一个（最坏情况）困难函数，并通过困难性放大和 Nisan-Wigderson 伪随机生成器 [NW94] 构造一个伪随机生成器。详细步骤与 [IW97]，中的略有不同，我们采用以下策略：

(a) 从 f 出发，通过低次扩展构造一个函数 f'，使其在 0.99 比例的输入上平均情况下困难。(b) 从 f' 出发，通过去随机化 XOR 引理构造一个函数 f''（具有多个比特的输出），使其在 $2^{-\Omega(m)}$ 比例的输入上平均情况下困难。(c) 从 f'' 出发，通过 Goldreich-Levin 构造一个函数 f'''，其输出为单个比特，并在 $1/2 + 2^{-\Omega(m)}$ 比例的输入上平均情况下困难。(d) 使用 f''' 来实例化 Nisan-Wigderson 伪随机生成器 $G\colon \{0,1\}^s \to \{0,1\}^n$ 用于$s = O(m)$。

我们确保 f'、f''、f''' 和 G 都能在 $O(\log n)$ 空间内计算。

此外，我们证明每一步都可以做到对数空间可重构：也就是说，给定一个针对结论的反例（即一个能获得某种优势的小电路），我们可以在确定性对数空间内构造出一个针对假设的反例。这需要修改前三步所用的标准重构算法，因为这些算法都以随机性效率低下的方式应用了概率方法。在接下来的四个小节中，我们陈述并证明构造性伪随机生成器所需的组成部分，并在第 4.6 节中结合这些结果，得出定理 4.1。

4.1 预备知识

首先，我们回顾一些与电路优势相关的记号。

定义 4.2。给定 $f\colon \{0,1\}^n \to \{0,1\}^m$ 和一个电路 C，令 $\mathrm{SUC}(C,f) = \Pr_{x\leftarrow U_n}[C(x) = f(x)]$。对于 $m = 1$，令 $\mathrm{ADV}(C,f) = 2\mathrm{SUC}(C,f) - 1$。令 $\mathrm{ADV}_s(f) = \max^{C:|C|\le s}\mathrm{ADV}(C,f)$，$\mathrm{SUC}_s(f)$ 也类似。令 ADV^H、SUC^H 表示相关概念，其中 x 是从 $H \subset \{0,1\}^n$ 上的均匀分布中采样得到的。

我们将反复使用平均采样器，以使概率方法论证在随机性使用上更高效。我们先回顾平均采样器的定义，然后回顾 [RVW01] 中的经典结果：即使误差呈指数级小，也存在高度高效的平均采样器。

定义 4.3。给定 $m \in \mathbb{N}$ 和 $\varepsilon,\ \delta > 0$，我们称 $\mathrm{SAMP}\colon \{0,1\}^l \to (\{0,1\}^m)^t$ 是一个具有种子长度 l 的 t-查询 (m,ε,δ)-平均采样器，如果对于每个 $g\colon \{0,1\}^m \to [0,1]$，都有

$$\Pr_{q_1,\ldots,q_t\leftarrow \mathtt{SAMP}(U_l)}\left[\left|\mathop{\mathbb{E}}_{i\in[t]}[g(q_i)] - \mathbb{E}[g]\right| \le \varepsilon\right] \ge 1-\delta.$$

命题 4.4（[RVW01]）。给定 $m \in \mathbb{N}$ 和 $\varepsilon > 0$，存在 $t = \mathrm{poly}(m/\varepsilon)$ 以及一个具有种子长度 $4m$ 的 t- 查询 $(m,\varepsilon,2^{-2m})$- 平均采样器。此外，该采样器可在空间复杂度类 $O(m)$ 中求值。

我们在证明中反复使用的另一种工具是组合设计，它是一个子集族 $S_1,\ldots,S_n \subseteq [s]$，满足对于某个常数 $\alpha \in \in(0,1)$ 和所有 $i \in [n]$，都有 $|S_i| = \alpha s$；而对于所有 $i \ne j$，都有 $|S_i \cap S_j| \le 2\alpha^2 s$。这种设计将在两个地方使用：一次用于去随机化 XOR 引理（第 4.3 节），一次用于 Nisan-Wigderson 伪随机生成器（第 4.5 节）。第 4.3 节中的应用只需要线性规模的设计，而第 4.5 节中的应用则需要一个指数规模、可在线性空间内确定性构造的设计。后者在 [KvM02] 中得到了正式阐述，因此我们将它同时用于这两种应用。

命题 4.5（[KvM02]）。对于每个 $\alpha \in \in(0,1)$，都存在 $\beta \in \in(0,1)$，使得对于 $s \in \mathbb{N}$，可以在空间 $O(s)$ 内确定性地生成一个大小为 $n = 2^{\beta s}$、定义在 $[s]$ 上的组合设计，也就是说，一个子集族 $S_1,\ldots,S_n \subseteq [s]$，满足 $|S_i| = \alpha s$，并且对于所有 $1 \le i < j \le n$，都有 $|S_i \cap S_j| \le 2\alpha^2 s$。

4.2 多项式解码器的去随机化

对于定理 4.1 中的步骤 (a)，我们需要将一个最坏情形困难函数转化为一个具有常数平均情况难度的函数。

引理 4.6. 给定 $f\colon \{0,1\}^m \to \{0,1\}$，存在 $g\colon \{0,1\}^{m'} \to \{0,1\}$，其中 $m' = \Theta(m)$，使得对于每个满足 $\mathrm{SUC}(\mathcal{B}, g) > 0.99$ 的电路 $\mathcal{B}$，都存在一个大小为 $m^{O(1)} \cdot |\mathcal{B}|$ 的电路 $\mathcal{C}$，使得

$$\mathcal{C}(x) = f(x),\ \forall x \in \{0,1\}^m.$$

此外，当 f 可在空间 $O(m)$ 中计算时，g 也可在空间 $O(m)$ 中计算，并且存在一个确定性的 $O(m)$-空间算法：给定可在空间 $O(m)$ 中求值的电路 $\mathcal{B}$，输出 $\mathcal{C}$，且 $\mathcal{C}$ 也可在空间 $O(m)$ 中求值。

引理 4.6 的证明受到 [STV01], 的启发，其中我们通过 Reed-Muller 码对 f 进行编码，并通过 Hadamard 码切换到布尔域。然而，由于我们只需要所得函数在常数比例的输入上是平均情形困难的，因此可以直接解码该码，而不必进行列表解码，并使用采样器对解码过程进行去随机化。

我们需要以下两个事实。第一个是关于构造低次扩展的一个民间已知事实，其证明可见 [GKR15, Proposition 2.2]:

命题 4.7。给定一个有限域 $\mathbb{F}$ 和一个子集 $H \subseteq \mathbb{F}$，以及对函数$f\colon H^\ell \to \{0,1\}$ 的预言机访问，可以在空间 $O(\log |\mathbb{F}| + \log \ell)$ 中计算出一个 ℓ-变量多项式 $p\colon \mathbb{F}^\ell \to \mathbb{F}$，使其在 H^ℓ 上与 f 一致，并且 p 关于每个变量的次数都小于 $|H|$。

第二个事实涉及 Reed-Solomon 码的解码:

命题 4.8。给定一个有限域 $\mathbb{F}$，其中 $|\mathbb{F}| = N$，其元素可以按规范顺序列为a_1、.....、a_N，其中 $a_1 = 0$，存在一个电路 DEC：$\mathbb{F}^N \to \mathbb{F}^N$，满足以下条件：若存在一个次数至多为 $d < N$ 的单变量多项式 $q\colon \mathbb{F} \to \mathbb{F}$，使得对于至少 $(N+d)/2$ 个 $i \in [N]$，$q(a_i) = b_i$，那么

$$\mathtt{DEC}(b_1, \ldots, b_N) = (q(a_1), \ldots, q(a_N)).$$

此外，DEC 的规模为 $\mathrm{poly}(N)$，深度为 $\mathrm{polylog}(N)$，并且给定 $\mathbb{F}$ 中的算术运算，可以在空间 $O(\log N)$ 内一致地构造。

证明。电路 DEC 实现了 Berlekamp-Welch 算法 [WB86,GS92]。该算法需要求解由 $O(N)$ 个线性方程组成、包含 $O(N)$ 个变量的方程组；对此，Csanky 算法[Csa76] 可以在对数空间一致 NC 中实现。 □

引理 4.6 的证明。我们不失一般性地假设 m 是 2 的幂。令 $\ell = m/\log m$，并令 $\mathbb{F}$ 为特征为 2、大小为 m^2 的有限域。取大小为 m 的子集 $H \subset \mathbb{F}$，并将 f 的定义域 $\{0,1\}^m$ 与 H^ℓ 视为 $2^m = |H|^\ell$。在 $\mathbb{F}$ 中进行运算所需的时间为 $O(|\mathbb{F}|)$，空间复杂度为 $O(m)$；$\{0,1\}^m$ 与 H^ℓ 之间的双射（及其逆映射）也同样如此。

令 $p\colon \mathbb{F}^\ell \to \mathbb{F}$ 为命题 4.7 中的多项式，并令 $g\colon \mathbb{F}^{\ell+1} \to \{0,1\}$ 为定义如下的函数

$$g(x_1, \ldots, x_\ell, y) = \langle p(x_1, \ldots, x_\ell), y \rangle,$$

其中，$\langle \cdot, \cdot \rangle$ 表示将两个参数的二进制表示作为 $\mathbb{F}$ 中的元素时的内积。显然，g 可在空间复杂度 $O(m)$ 内计算；以二进制表示时，g 的输入长度为 $(\ell+1)\log |\mathbb{F}| = O(m)$。

现在假设存在一个电路 $\mathcal{B}$ 使得 $\mathrm{SUC}(\mathcal{B}, g) > 0.99$。我们首先构造电路$\mathcal{B}'\colon \mathbb{F}^\ell \to \mathbb{F}$ 使得输出的第 i 个比特是

$$\mathcal{B}'_i(x_1, \ldots, x_\ell) = \mathtt{MAJ}_{z \in \mathbb{F}}(\mathcal{B}(x_1, \ldots, x_\ell, e_i + z) - \mathcal{B}(x_1, \ldots, x_\ell, z)).$$

这里 e_i 是 $\mathbb{F}$ 中二进制表示在第 i 个比特上为 1、其他位置均为 0 的元素 e.

命题 4.9. $\mathrm{SUC}(\mathcal{B}', p) \geq 0.96$。

证明。由于 $\mathrm{SUC}(\mathcal{B}, g) > 0.99$，因此至少有 $(x_1, \ldots, x_\ell) \in \mathbb{F}^\ell$ 的 0.96 比例满足：$\mathcal{B}$与 g 在超过 $3/4$ 的 $y \in \mathbb{F}$上相同；对于随机 $z \in \mathbb{F}$，该集合同时包含 z 和 $(e_i + z)$ 的概率大于 $1/2$。在这种情况下，对于每个 i，我们都有 $\mathcal{B}'_i(x_1, \ldots, x_\ell) = \langle p(x_1, \ldots, x_\ell)\ e_i \rangle$，因此 $\mathcal{B}'(x_1, \ldots, x_\ell) = p(x_1, \ldots, x_\ell)$。 □

根据 $\mathcal{B}'$，我们如下重构电路 $\mathcal{C}$：$\{0,1\}^m \to \{0,1\}$。令 SAMP：$\{0,1\}^{8m} \to (\mathbb{F}^\ell)^t$为命题 4.4 中的采样器，其中 $\varepsilon = 0.01$，因此 $t = \mathrm{poly}(m)$。我们将 SAMP 视为采样 t 个随机向量 $v = (v_1, \ldots, v_\ell) \in \mathbb{F}^\ell$；给定 $\mathcal{C}$的输入 $x = (x_1, \ldots, x_\ell) \in H^\ell$后，每个向量 v表示一条直线 $\{x + \lambda v \mid \lambda \in \mathbb{F}\}$。在每条直线上，$p(x + \lambda v)$ 是定义在 λ 上的单变量多项式，其次数至多为 $\ell|H| = m^2/\log m$；我们使用命题 4.8 中的解码器电路 DEC，对该直线上由 $\mathcal{B}'$ 给出的 Reed-Solomon 码进行解码。我们将 $\mathcal{C}(x)$ 的值设为 t条直线中解码值出现次数最多的那个值（平票时任意选择）。请注意，这一过程取决于采样器的种子；实际上，我们会遍历所有种子，并选择使 $\mathcal{C}(x)$ 在所有 $x \in H^\ell$上正确计算 f 的那个种子。

形式化地，我们将这个线性空间重构算法表述为算法 1。

算法 1： $\mathrm{RM_RECON}(f, \mathcal{B})$

令 SAMP：$\{0,1\}^{8m} \to (\mathbb{F}^\ell)^t$ 为命题 4.4 中满足 $\varepsilon = 0.01$ 的采样器。
对 $y \in \{0,1\}^{8m}$ 执行
  令 $v_1, \ldots, v_t \leftarrow \mathrm{SAMP}(y)$。
  令 $\mathcal{C}$：$\{0,1\}^m \to \{0,1\}$ 为电路

$$\mathcal{C}(x) = \text{多数函数}_{i \in [t]}(\mathrm{DEC}_1((\mathcal{B}'(x + \lambda v_i))_{\lambda \in \mathbb{F}}))。$$

  如果对所有 $x \in \{0,1\}^m$ 均有 $\mathcal{C}(x) = f(x)$，则返回 $\mathcal{C}$
结束

算法中构造的电路 $\mathcal{C}$ 的规模为 $2t|\mathbb{F}|^2|\mathcal{B}| + m^{O(1)} = m^{O(1)} \cdot |\mathcal{B}|$，与 $\mathcal{B}$相比，额外深度为 $\mathrm{polylog}(m)$。因此，$\mathcal{C}$ 可以在空间复杂度 $O(m)$ 中求值。

现在我们证明该算法总能返回一个有效的电路 $\mathcal{C}$。注意，对于均匀随机的 $v \in \mathbb{F}^\ell$，在给定 x和 $\lambda \neq 0$后，$x + \lambda v$仍然是均匀随机的。由于 $\mathrm{SUC}(\mathcal{B}', p) \geq 0.96$，这意味着至少有 0.84 比例的 $v \in \mathbb{F}^\ell$满足：$\mathcal{B}'$在 $x + \lambda v$上与 p一致，并且至少对于 $\lambda \in \mathbb{F}$中的 $3/4$，$\lambda \neq 0$都成立。回顾一下，$q(\lambda) = p(x + \lambda v)$的次数至多为 $\ell|H| = |\mathbb{F}|/\log m$，因此根据命题 4.8，我们可以得出，对于每个 $x \in \{0,1\}^\ell$，

$$\Pr_{v \in \mathbb{F}^\ell}\left[\mathtt{DEC}((\mathcal{B}'(x + \lambda v))_{\lambda \in \mathbb{F}}) = (p(x + \lambda v))_{\lambda \in \mathbb{F}}\right] \geq 0.84,$$

在这种情况下，我们有 $\mathrm{DEC}_1((\mathcal{B}'(x + \lambda v))_{\lambda \in \mathbb{F}}) = p(x)$。将该概率视为 v 上示性函数的期望，根据命题 4.4 中采样器的保证，我们有

$$\Pr_{v_1, \ldots, v_t \leftarrow \mathtt{SAMP}(U_{8m})}\left[\Pr_{i \in [t]}\left[\mathtt{DEC}_1((\mathcal{B}'(x + \lambda v_i))_{\lambda \in \mathbb{F}}) = p(x)\right] \geq 0.51\right] \geq 1 - 2^{-4m}.$$

根据对 $x \in \{0,1\}^m$ 的并合界，必然存在一个 $y \in \{0,1\}^{8m}$ 使得对于所有 $x \in \{0,1\}^m$，$\mathcal{C}(x) = p(x) = f(x)$。因此，该算法总是返回这样一个电路 $\mathcal{C}$。此外，该算法可以实现在空间 $O(m)$ 中运行：我们可以枚举采样器的种子，并在空间 $O(m)$ 中构造电路（作为采样器输出的函数），然后在此空间界内测试该电路是否正确计算 f。 □

4.3 对去随机化的异或引理进行去随机化

我们的下一步沿用了 Impagliazo 和 Wigderson [IW97], 的方法。他们使用去随机化 XOR 引理, 将一个在常数比例输入上困难的函数，转化为一个在任意指数级小比例输入上都困难的函数。该构造与 [IW97], 中的构造完全相同，不同之处在于，我们修改了重构算法及其分析，使得电路 $\mathcal{C}$ 可以在确定性空间 $O(m)$ 中构造。

引理 4.10. 对于每个 $\gamma \in (0,1)$，都存在一个可由 $O(m)$-空间计算的函数 $G\colon \{0,1\}^{m'} \to (\{0,1\}^m)^m$，其中 $m' = \Theta(m/\gamma)$，满足以下性质：给定 $f\colon \{0,1\}^m \to \{0,1\}$，以及一个满足 $\mathrm{SUC}(\mathcal{B}, f^m \circ G) \geq 2^{-\gamma m}$ 的电路 $\mathcal{B}$，存在一个大小为 $2^{O(\gamma m)} \cdot |\mathcal{B}|$ 的电路 $\mathcal{C}$，使得

$$\mathtt{SUC}(\mathcal{C}, f) > 0.99.$$

此外，当 f 可由空间复杂度 $O(m)$ 计算时，存在一个确定性的 $O(m)$-空间算法；给定可由空间复杂度 $O(m)$ 求值的电路 $\mathcal{B}$，该算法会输出 $\mathcal{C}$，且 $\mathcal{C}$ 也可由空间复杂度 $O(m)$ 求值。

我们首先给出函数 G 的构造；该函数在 [IW97] 中称为直积生成器。与 [IW97], 一样，它由两个部分组成：扩展器游走和组合设计。对于扩展器游走，我们需要一个显式扩展器，使得能够高效地计算某个顶点的邻居：

命题 4.11（参见例如 [LPS88]）。存在常数 $\lambda \in (0,1)$，使得对于每个 $m \in \mathbb{N}$，都存在一个顶点集为 $\{0,1\}^m$ 的 4-正则图 E_m，其谱扩张至多为 λ；对于任意顶点 $v \in \{0,1\}^m$，其邻居可在时间 $\mathrm{poly}(m)$ 和空间 $O(\log m)$ 内计算得到。

定义扩展器游走函数 $\mathrm{EW}\colon \{0,1\}^{3m} \to (\{0,1\}^m)^m$ 如下：给定输入 $v \in \{0,1\}^m$ 和 $d = (d_1, \ldots, d_m) \in [4]^m$，输出是顶点序列 $v_1, \ldots, v_m$，其中顶点属于 E_m，序列以 $v_1 = v$ 开始，并令 v_{i+1} 为 v_i 的第 d_i 个邻居。另一方面，令 $S_1, \ldots, S_m \subseteq [s]$ 为组合设计中由命题 4.5 给出的前 m 个集合，其中参数为 $\alpha = \gamma/2$ 和 $s = m/\alpha$。于是我们定义函数 $G\colon \{0,1\}^{3m+s} \to (\{0,1\}^m)^m$ 如下：

$$G(r, v, d) = \big((r|_{S_1}) \oplus \mathtt{EW}(v,d)_1, \ldots, (r|_{S_m}) \oplus \mathtt{EW}(v,d)_m\big).$$

这里，$r|_S$ 是 $r \in \{0,1\}^s$ 在索引 S 上的部分，$\oplus$ 表示按位异或。根据定义，我们有 G 可以在多项式时间和空间复杂度 $O(m)$ 内计算。G 的输入长度为 $m' = 3m + 2m/\gamma = O(m/\gamma)$。

现在给定 $f\colon \{0,1\}^m \to \{0,1\}$，假设存在一个电路 $\mathcal{B}$，使得 $\mathrm{SUC}(\mathcal{B}, f^m \circ G) \geq 2^{-\gamma m}$。在继续说明如何从 $\mathcal{B}$ 高效且确定性地重构电路 $\mathcal{C}$ 之前，我们先回顾一下 [IW97] 中的重构步骤。对于 $i \in [m]$、$x \in \{0,1\}^m$、$a \in \{0,1\}^{s-m}$、$v \in \{0,1\}^m$ 和 $d \in [4]^m$，令 $h(i, x, a, v, d) = (r, v, d)$，其中 $r \in \{0,1\}^s$ 满足

$$r|_{S_i} = x \oplus \mathtt{EW}(v,d)_i \text{ and } r|_{\overline{S_i}} = a.$$

函数 h 称为 G 的限制函数。给定 $x \in \{0,1\}^m$，在 i、a、v 和 d 均匀随机选取的情况下，它们构造一个电路 $\mathcal{F}$，该电路首先模拟 $\mathcal{B}$，以计算 $\mathcal{B}(h(i,x,a,v,d)) = (y_1, \ldots, y_m)$。然后计算一个定义如下的数值 t

$$t = \left|\left\{ j \neq i \mid y_j \neq f\big(G_j \circ h(i,x,a,v,d)\big) \right\}\right|,$$

并以概率 2^{-t}输出 y_i，以概率 $1-2^{-t}$输出一个随机比特。为了计算 t，对于每个 $j \neq i$，$f(G_j \circ h(i,x,a,v,d))$ 通过一个以非一致方式构造的查找表计算，该表针对 f，大小为 $2^{\gamma m}$，其中包含在固定的 i、a、v和 d下，$G \circ h$的所有可能的第 j 个输出 x_j对应的 $f(x_j)$ 值。

我们无法利用非一致性来构造查找表。不过，当 f 可在空间 $O(m)$内计算时，我们可以在空间 $O(m)$内计算出整张表，并将其硬编码到电路中。更好的是，给定 i、a、v和 d后，$G \circ h$的每个输出 x_j除了 γm个比特（对应于 $S_i \cap S_j$中的坐标）之外都已固定，因此我们只需遍历这些比特的全部 $2^{\gamma m}$ 种可能性即可计算出该表。

上文介绍的电路 $\mathcal{F}$使用了一个由 $|R| = O(m)$ 个随机比特组成的字符串 R，其中包括 i、a、v、d以及 $w \in \{0,1\}^{m+1}$；后者是用于决定最终输出的随机性。[IW97]中证明了：

命题 4.12（[IW97，定理 15]）。设 $\mathrm{SUC}(\mathcal{B}, f^m \circ G) \geq 2^{-\gamma m}$。存在 $c > 0$（取决于 γ），使得满足以下条件的输入所占比例为 $x \in \{0,1\}^m$

$$\Pr_R[\mathcal{F}(x,R) = f(x)] \geq 1/2 + 2^{-\gamma m}/c$$

大于 0.99。

因此，最终电路 $\mathcal{C}$ 对 $O(m \cdot 2^{2\gamma m})$ 个 $\mathcal{F}$ 的独立副本进行计算，并输出它们的多数值；同时，存在一种随机性的固定方式，可得到最终的确定性电路 $\mathcal{C}$。如果随机比特是独立采样的，我们将无法承担存储指数数量随机比特的代价。相反，我们采用命题 4.4 中的高效采样器，它仅使用 $O(m)$ 个随机比特作为种子来生成 $2^{O(\gamma m)}$ 个样本，并且可以枚举所有种子，找到使 $\mathrm{SUC}(\mathcal{C}, f) > 0.99$ 的种子。如下方证明所示，这样的种子总是存在。

引理 4.10 的证明。设 $\mathcal{F}\colon \{0,1\}^{m+|R|} \to \{0,1\}$ 为上述描述的电路，$c > 0$ 为命题 4.12 中的常数。我们将线性空间算法的重构过程形式化描述为算法 2。

算法 2： $\mathrm{XOR_RECON}(f, \mathcal{B})$

令 $\mathrm{SAMP}\colon \{0,1\}^{4|R|} \to (\{0,1\}^{|R|})^t$ 为命题 4.4 中满足 $\varepsilon = 2^{-\gamma m}/(2c)$ 的采样器。

对于 $y \in \{0,1\}^{4|R|}$，执行

  令 $R_1, \ldots, R_t \leftarrow \mathrm{SAMP}(y)$。

  令 $\mathcal{C}\colon \{0,1\}^m \to \{0,1\}$ 为电路

$$\mathcal{C}(x) = \mathtt{MAJ}(\mathcal{F}(x,R_1), \ldots, \mathcal{F}(x,R_t)).$$

  如果 $\mathrm{SUC}(\mathcal{C}, f) > 0.99$，则返回 $\mathcal{C}$

结束

根据命题 4.4，我们有 $t = \mathrm{poly}(m/\varepsilon) = 2^{O(\gamma m)}$ 对于 $\varepsilon = 2^{-\gamma m}/(2c)$。从描述中可知，$\mathcal{F}$的大小为 $|\mathcal{B}| + 2^{\gamma m} \cdot m^{O(1)}$，因此 $\mathcal{C}$ 的大小为 $t|\mathcal{F}| + m^{O(1)} = 2^{O(\gamma m)} \cdot |\mathcal{B}|$。当 $\mathcal{B}$ 可在空间复杂度类 $O(m)$ 中求值时，显然 $\mathcal{C}$ 也可在空间复杂度类 $O(m)$ 中求值。

根据命题 4.4 中平均采样器 SAMP 的保证，对于每个 $x \in \{0,1\}^m$：

$$\Pr_{R_1,\ldots,R_t \leftarrow \mathtt{SAMP}(U_{4|R|})} \left[\left| \mathop{\mathbb{E}}_{i \in [t]} [\mathcal{F}(x,R_i)] - \mathop{\mathbb{E}}_{R}[\mathcal{F}(x,R)] \right| \leq \varepsilon \right] \geq 1 - 2^{-2|R|}.$$

根据命题 4.12，存在一个子集 $V \subseteq \{0,1\}^m$，使得 $|V| > 0.99 \cdot 2^m$，并且对于每个 $x \in V$：

$$\left| \mathop{\mathbb{E}}_{R}[\mathcal{F}(x,R)] - f(x) \right| \leq 1/2 - 2^{-\gamma m}/c = 1/2 - 2\varepsilon.$$

因此，对于每个 $x \in V$，可推出

$$\Pr_{R_1,\ldots,R_t \leftarrow \mathtt{SAMP}(U_{4|R|})} \left[\left| \mathop{\mathbb{E}}_{i\in[t]}[\mathcal{F}(x,R_i)] - f(x) \right| \leq 1/2 - 2\varepsilon + \varepsilon \right] \geq 1 - 2^{-2|R|},$$

这意味着

$$\Pr_{R_1,\ldots,R_t \leftarrow \mathtt{SAMP}(U_{4|R|})} [\mathtt{MAJ}(\mathcal{F}(x,R_1),\ldots,\mathcal{F}(x,R_t)) = f(x)] \geq 1 - 2^{-2|R|} > 1 - 1/|V|.$$

通过对 $x \in V$ 应用并合界，必然存在一个 $y \in \{0,1\}^{4|R|}$ 使得对于所有 $x \in V$，$\mathcal{C}(x) = f(x)$，且满足 $\mathsf{SUC}(\mathcal{C}, f) > 0.99$。因此，该算法始终返回一个有效的 $\mathcal{C}$。此外，该算法的空间复杂度为 $O(|R|) = O(m)$ 的种子，构造并计算电路 $\mathcal{C}$，并对 f 进行预言机调用；所有这些操作都可以在 $O(m)$ 的空间内完成。 □

4.4 Goldreich-Levin 定理的去随机化

引理 4.13. 给定 $f\colon \{0,1\}^m \to \{0,1\}^m$，令 $g\colon \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$ 定义为 $g(x,r) = \langle f(x)\ r\rangle$。那么，给定 $\delta > 0$，存在 $\delta' \geq \Omega(\delta^3/m)$，使得对于每个满足 $\mathsf{ADV}(\mathcal{B}, g) > \delta$ 的 $\mathcal{B}$，存在一个大小至多为 $|\mathcal{B}| \cdot (m/\delta)^{O(1)}$ 的电路 $\mathcal{C}$，满足

$$\mathsf{SUC}(\mathcal{C}, f) > \delta'.$$

此外，当 f 可在空间复杂度 $O(m)$ 内计算时，存在一个确定性的 $O(m)$-空间算法，该算法以可在空间复杂度 $O(m)$ 内求值的电路 $\mathcal{B}$ 为输入，输出 $\mathcal{C}$，且 $\mathcal{C}$ 也可在空间复杂度 $O(m)$ 内求值。

请注意，原始 Goldreich-Levin 定理 [GL89] 并不保证（事实上也没有给出）高效的确定性重构器，因为它在随机性方面并不高效。Hoza 和 Klivans [HK18] 后来的工作实现了这一点，但证明要复杂得多。因此，我们直接利用小偏差空间来证明这一点，下面给出其定义：

定义 4.14. 若函数 $G\colon \{0,1\}^t \to \{0,1\}^k$ 满足 $G(U_t)$ 是 $\{0,1\}^k$ 上的一个 ε-偏差概率空间，则称其为 ε-偏差生成器。形式化地说，这意味着对于每个 $T \in \{0,1\}^k$，

$$\Pr_{y \leftarrow U_t} [\langle T, G(y)\rangle = 1] \in [1/2 - \varepsilon, 1/2 + \varepsilon].$$

我们回顾一下，存在种子长度良好的小偏差生成器，而且这些生成器还可以在小空间内求值：

命题 4.15（[NN93]）。给定 $k \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个可由 $O(t)$-空间计算的 ε-偏差生成器 $\mathsf{BIAS}: \{0,1\}^t \to \{0,1\}^k$，其种子长度为 $t = O(\log(k/\varepsilon))$.

我们需要一个基本的傅里叶分析引理，它指出，小偏差空间能够欺骗 k 奇偶校验的合取。

引理 4.16. 设 BIAS：$\{0,1\}^t \to \{0,1\}^k$ 是一个 ε-偏差生成器。那么，对于任意集合 $T_1, \ldots, T_d \in \{0,1\}^k$ 和 $v_1, \ldots, v_d \in \{0,1\}$，我们有

$$\left| \mathop{\mathbb{E}}_{r \leftarrow \mathtt{BIAS}(U_t)} \left[\bigwedge_{i \in [d]} (\langle T_i, r \rangle \oplus v_i) \right] - \mathop{\mathbb{E}}_{r \leftarrow U_k} \left[\bigwedge_{i \in [d]} (\langle T_i, r \rangle \oplus v_i) \right] \right| \leq 2\varepsilon.$$

证明。我们有

$$\begin{aligned} \bigwedge_{i \in [d]} (\langle T_i, r \rangle \oplus v_i) &= 1 - 2 \cdot 2^{-d} \sum_{S \subseteq [d]} \bigoplus_{i \in S} \neg (\langle T_i, ry \rangle \oplus v_i) \\ &= 1 - 2 \cdot 2^{-d} \sum_{S \subseteq [d]} \left(\left\langle \bigoplus_{i \in S} T_i, r \right\rangle \oplus \bigoplus_{i \in S} \neg v_i \right) \end{aligned}$$

由于 BIAS 能够将所有此类奇偶校验的误差欺骗至求和中 ε，因此总误差至多为 2ε。

□

引理 4.13 的证明。如果 $\delta < 2^{-m}$，我们可以选择 $\delta' = 2^{-m}$，对于输出常量的电路 $\mathcal{C}$，该引理显然成立。因此，从现在起我们假设 $\delta \geq 2^{-m}$。我们将算法正式表述为算法 3，其中 δ' 稍后确定。注意，$\ell = O(m)$，因此在 ε-偏置生成器 BIAS：$\{0,1\}^t \to \{0,1\}^{\ell \times m}$ 中，我们有 $t = O(\log(\ell m/\varepsilon)) = O(m)$，其中 $\varepsilon = 2^{-4m-1}$，且该算法在空间 $O(t + \ell + m) = O(m)$ 中运行。

算法 3：$\mathrm{GL_RECON}(f,\ B)$

令 $\ell \leftarrow \lceil \log_2(128m/\delta^2 + 1) \rceil$。令 BIAS：$\{0,1\}^t \to \{0,1\}^{\ell \times m}$ 为命题 4.15 中满足 $\varepsilon = 2^{-4m-1}$ 的生成器。对于 $y \in \{0,1\}^t$，执行令 $r_1, \ldots, r_\ell \leftarrow$ 为 BIAS(y)。对于 $(b_1, \ldots, b_\ell) \in \{0,1\}^\ell$，执行令 $\mathcal{C}: \{0,1\}^m \to \{0,1\}^m$ 为电路，使得对于每个 $i \in [m]$：$\mathcal{C}_i(x)$，都有 $=$ 多数函数$_{J \subseteq [\ell]: J \neq \emptyset}(b^J \oplus \mathcal{B}(x, r^J \oplus e_i))$。如果 SUC$(\mathcal{C}, f) > \delta'$，则返回 $\mathcal{C}$。
结束 结束

我们将 BIAS 的输出视为由 ℓ 个向量组成的元组：

$$\mathtt{BIAS}(y) = (r_1, \ldots, r_\ell), \quad r_i \in \{0,1\}^m.$$

为方便起见，令 $\vec{r} := (r_1, \ldots, r_\ell)$ 和 $\vec{b} := (b_1, \ldots, b_\ell)$。对于每个 $J \subseteq [\ell]$，令：

$$r^J := \bigoplus_{i \in J} r_i, \qquad b^J = \bigoplus_{i \in J} b_i.$$

注意，在原始 GL 算法中，所有的 r_i 都是从 $\{0,1\}^m$ 上独立同分布地均匀抽取的。我们首先说明，我们关于 r^J 的分布近似满足分析原始算法时用到的以下两个性质：

命题 4.17. 以下两个性质成立：

1. 对于每个非空的 J， r^J 在 ℓ_1-距离下与 U_m 2^{-2m}-接近。

2. 对于每个 n 非空的 J 和 J'，其中 $J \neq J'$，$(r^J, r^{J'})$ 在 ℓ_1-距离下与 U_{2m} 2^{-2m} 接近 $ce.$

证明。对于 $i \in [m]$， r^J 的第 i 个比特可以写成 $\langle T_{i,\ J}, \text{BIAS}(y)\rangle$，其中 $T_{i,\ J}$ 表示比特的一个非空子集。由引理 4.13 可知，对于每个 $v \in \{0,1\}^m$，

$$\left| \Pr_{y \leftarrow U_t}[r^J = v] - \Pr_{r \leftarrow U_{\ell m}}\left[\bigwedge_{i\in[m]} (\langle T_{i,J}, r\rangle = v_i)\right]\right| \leq 2\varepsilon.$$

注意到 $\{T_{i,J}\}_{i\in[m]}$ 线性无关，因此 $(\langle T_{i,\ J},\ r\rangle)^{i\in[m]}$ 在 $\{0,1\}^m$ 上均匀分布。因此对 $v \in \{0,1\}^m$ 求和可得， r^J 在 ℓ_1 距离下与 U_m $2\varepsilon \cdot 2^m \leq 2^{-2m}$-接近。

当 $J \neq J'$ 和 $\{T_{i,J}\}_{i\in[m]} \cup \{T_{i,J'}\}_{i\in[m]}$ 都非空时， $\{T_{i,J}\}_{i\in[m]} \cup \{T_{i,J'}\}_{i\in[m]}$ 仍然线性无关。理由与上文相同，$(r^J, r^{J'})$ 在 ℓ_1 距离下与 U_{2m} 相差不超过 $2\varepsilon \cdot 2^2 m = 2^{-2m}$。 □

现在回忆一下，对于 $i \in [m]$， $\mathcal{C}$ 的输出的第 i 个比特是

$$\mathcal{C}_i(x) = \mathtt{MAJ}_{J:J\neq\emptyset}(b^J \oplus \mathcal{B}(x, r^J \oplus e_i)).$$

因此， $\mathcal{C}$ 的大小为 $|\mathcal{C}| \leq (|\mathcal{B}| + O(\ell)) \cdot 2^\ell m \leq |\mathcal{B}| \cdot O(2^\ell \ell m) = |\mathcal{B}| \cdot (m/\delta)^{O(1)}$，如所述。为了分析 $\mathcal{C}$的性能，令

$$S := \{x \in \{0,1\}^m : \Pr_{z\leftarrow U_m}[\mathcal{B}(x,z) = g(x,z)] \geq 1/2 + \delta/2\}.$$

根据标准平均论证， $|S| \geq (\delta/2) \cdot 2^m$。

命题 4.18。对于任意 $x \in S$ 和 $i \in [m]$,

$$\Pr_{(r_1,\ldots,r_\ell)\leftarrow \mathtt{BIAS}(U_t)}\left[\left|\{J : \mathcal{B}(x, r^J \oplus e_i) = g(x, r^J \oplus e_i)\}\right| \leq \frac{1}{2}(2^\ell - 1)\right] \leq \frac{1}{2m}.$$

证明。在证明的其余部分，我们固定 x 和 i。令 $A \subset \{0,1\}^m$ 为这样的取值集合： r 使得 $\mathcal{B}(x,r) = g(x,r)$。由 $x \in S$ 这一事实可知， $|A| \geq (1/2 + \delta/2) \cdot 2^m$。此外，对于每个 $y \in \{0,1\}^t$ （其中 y 是 BIAS 的输入），令

$$\zeta_J(y) = \mathbb{I}[r^J \oplus e_i \in A]$$

并注意到， $\zeta_J = 1$ 等价于 $\mathcal{B}(x, r^J \oplus e_i) = g(x, r^J \oplus e_i)$，即 $\mathcal{B}$ 在该输入上正确地计算了与 $f(x)$ 的内积。现在注意，根据命题 4.17,

$$\mathop{\mathbb{E}}_y[\zeta_J] = \Pr_y[\zeta_J(y) = 1] \geq 1/2 + \delta/2 - 2^{-2m} \geq 1/2 + \delta/4.$$

现在，我们来界定在这些位置上正确计算内积的次数的方差。令

$$\begin{aligned}\sigma^2 = \mathrm{Var}\left(\sum_J \zeta_J\right) &= \sum_{J,J'} \mathrm{Cov}(\zeta_J, \zeta_{J'}) \\ &\leq \sum_J \mathrm{Var}(\zeta_J) + \sum_{J,J'} 2^{-2m} \\ &\leq 2^\ell + 2^{2\ell} \cdot 2^{-2m} \leq 2^{\ell+1}\end{aligned}$$

其中，第一个不等式由命题 4.17 得出。接下来，应用切比雪夫不等式和并合界即可得到结果。为方便起见，令 $d = 2^\ell - 1$，则命题中的概率等于：

$$\begin{aligned}\Pr_y\left[\sum_J \zeta_J \le \frac{d}{2}\right] &\le \Pr_y\left[\left|\sum_J \zeta_J - \mathbb{E}[\zeta_J]\cdot d\right| \ge \left(\frac{d\delta}{4\sigma}\right)\cdot\sigma\right] \\ &\le \frac{16\sigma^2}{\delta^2(2^\ell-1)^2} \le \frac{32\sigma^2}{\delta^2 2^{2\ell}} \le \frac{64}{\delta^2 2^\ell} \le \frac{1}{2m}.\end{aligned}$$

□

注意，当 $\mathcal{B}(x, r^J \oplus e_i) = g(x, r^J \oplus e_i)$ 且对每个 $j \in [\ell]$, $b_j = g(x, r_j)$ 都成立时，我们有

$$b^J \oplus \mathcal{B}(x, r^J \oplus e_i) = g(x, r^J) \oplus g(x, r^J \oplus e_i) = g(x, e_i) = f_i(x).$$

因此，根据命题 4.18 对 $i \in [m]$ 使用并合界，我们得到对每个 $x \in S$,

$$\begin{aligned}\Pr_{\substack{\vec{r}\leftarrow \mathtt{BIAS}(U_t)\\ \vec{b}\leftarrow U_l}}[\mathcal{C}(x) = f(x)] \ge &\Pr_{\vec{r}\leftarrow \mathtt{BIAS}(U_t)}\left[\forall i \in [m], \left|\{J : \mathcal{B}(x, r^J \oplus e_i) = g(x, r^J \oplus e_i)\}\right| > \frac{1}{2}(2^l - 1)\right] \\ &\cdot \Pr_{\vec{b}\leftarrow U_\ell}\left[\forall j \in [\ell], b_j = g(x, r_j)\right] \\ \ge &\frac{1}{2}\cdot \Pr_{\vec{b}\leftarrow U_\ell}\left[\forall j \in [\ell], b_j = g(x, r_j)\right] \ge 2^{-\ell-1}.\end{aligned}$$

因此，存在一个 $2^m \cdot \delta 2^{-\ell-2}$ y $\vec{b}$ $\mathcal{C}$ f $|S|\cdot 2^{-\ell-1} \ge \vec{b}$ 的赋值，使得其在至少 个输入上计算正确。此外，我们可以通过枚举 y 和 的赋值，并对所有 $x \in \{0,1\}^m$ 计算 $\mathcal{C}$ 和 f 来验证成功概率，从而找到这样的电路。因此，令

$$\delta' = \delta 2^{-\ell-2} = \Omega(\delta^3/m)$$

证明至此完成。 □

4.5 空间高效的 Nisan-Wigderson伪随机生成器

我们回顾 [KvM02] 的论证：利用线性空间可构造组合设计（命题 4.5），可以对 Nisan-Wigderson 伪随机生成器 NW94] 进行空间高效的实现。我们用自己的符号重新表述他们的结果，但不对构造作任何改动，因为（与其他所有步骤不同）现有实现满足我们所需的重构性质。

引理 4.19. 给定 $\rho > 0$、 $n \in \mathbb{N}$ 和一个函数族 $f_m\colon \{0,1\}^m \to \{0,1\}$ ∈空间复杂度类[m], 存在一个 $m = \Theta(\log n)$ 和 $G\colon \{0,1\}^s \to \{0,1\}^n$，满足 $s = O(m)$；使得对于一个电路 $\mathcal{B}$，若它是针对 G 的下一位预测器，且优势为 ε，则存在一个大小为 $|\mathcal{B}| + O(n2^{\rho m})$ 的电路 $\mathcal{C}$，满足

$$\mathtt{ADV}(\mathcal{C}, f_m) > \varepsilon.$$

此外，存在一个确定性的 $O(m)$-空间算法；给定可在空间 $O(m)$ 内求值的电路 $\mathcal{B}$，该算法会输出 $\mathcal{C}$, 且 $\mathcal{C}$ 也可在空间 $O(m)$ 内求值。

引理 4.19 的证明。固定 $\alpha \in (0,1)$ 使得 $\alpha \le \rho/2$，并令 $\beta \in (0,1)$ 为命题 4.5 中的常数。选择 $s = O(\log n)$ 使得 $2^{\beta s} = n$，并令 $m = \alpha s$。令 $\mathcal{S} = (S_1, \ldots, S_n)$ 为命题 4.5 在 $[s]$ 上、参数为 α 的设计，并令 $f_m\colon \{0,1\}^m \to \{0,1\}$ 为作用于大小为 $m = O(\log n)$ 的输入上的函数。

我们令 $G(x) := f(x_{S_1})f(x_{S_2}) \dots f(x_{S_n})$。现在假设 $\mathcal{B}$ 是针对 G 的第 i 位比特的 ε-下一位预测器，即

$$\Pr_{x \leftarrow U_s}[\mathcal{B}(G(x)_{1..i}) = G(x)_{i+1}] > \frac{1}{2} + \varepsilon.$$

接着令 $S := S_{i+1}$ 和 $T := [s] \setminus S_{i+1}$，并将上述不等式写为

$$\Pr_{(x_S, x_T) \leftarrow U_s}[\mathcal{B}(G(x_S \cup x_T)_{1..i}) = f(x_S)] > \frac{1}{2} + \varepsilon.$$

对于 x_T 的每一种固定取值，我们令电路 $\mathcal{C}$ 为 $\mathcal{C}(x_S) = \mathcal{B}(G(x_S \cup x_T)_{1..i})$。于是有

$$\mathop{\mathbb{E}}_{x_T}[\mathtt{ADV}(\mathcal{C}, f_m)] > \varepsilon.$$

因此，该算法可以在空间复杂度为 $|T| = O(m)$ 的情况下枚举 x_T 的所有可能赋值，并针对每种赋值检查 $\mathcal{C}$ 的优势。一旦算法找到 x_T 的一种固定取值，使得受限电路对每个 $j \le i$ 的优势至少为 ε，则 j 的输出的第 $G(x_S \cup x_T)$ 位，也就是 $f(x_{S_j})$，仅取决于 $|S \cap S_j| \le 2\alpha^2 s = \rho m$ 个 x_S 比特。因此，我们可以输出一个规模至多为 $O(2^{\rho m})$ 的、可由 ($O(m)$-空间复杂度) 构造的 $f(x_{S_j})$ 电路，从而 $\mathcal{C}$ 的总规模至多为 $|\mathcal{B}| + O(n2^{\rho m})$。 □

4.6 汇总

定理 4.1 的证明。给定 ε，我们首先执行构造步骤。对于每个 $m \in \mathbb{N}$:

1. 令 $f' \colon \{0,1\}^{m_1} \to \{0,1\}$ 为将引理 4.6 应用于 f_m 所得到的函数 g。2. 令 $f'' \colon \{0,1\}^{m_2} \to \{0,1\}^{m_1}$ 为将引理 4.10 应用于 f'_{m_1} 所得到的函数 $f'^{m_1} \circ G$，其中常数 γ 稍后确定。3. 令 $f''' \colon \{0,1\}^{m_3} \to \{0,1\}$ 为将引理 4.13 应用于 f''_{m_2} 所得到的函数 g，其中常数 δ 稍后确定。4. 令 $G \colon \{0,1\}^s \to \{0,1\}^n$ 为将引理 4.19 应用于 f'''_{m_3} 和 $\mathcal{B}$ 所得到的函数，其中常数 ρ 稍后确定。

注意，m_1、m_2、m_3和 s都是 $\Theta(m)$，而函数 f'、f''、f'''和 G都能在空间 $O(m)$ 内计算。

现在假设给定一个针对 G、大小为 n^2的 $1/(8n)$ 下一位预测器 $\mathcal{B}$。由于给定了 n，我们通过引理 4.19 确定 $m_3 = \Theta(\log n)$ 的值，进而确定 $m = \Theta(\log\ n)$ 的值。重构步骤如下:

4. 根据引理 4.19，我们可以在空间 $O(m)$ 内构造一个电路 $\mathcal{C}_3$，使得 $\mathrm{ADV}(\mathcal{C}_3, f'''_{m_3}) > 1/(8n)$，且 $\mathcal{C}_3$的规模为 $s_3 = n^2 + O(n2^{\rho m_3}) \le 2^{c_3 \rho m}$，其中某个常数 $c_3 > 0$。3. 根据引理 4.13，现在令 $\delta = 1/(8n)$，我们可以在空间 $O(m)$ 内构造一个电路 $\mathcal{C}_2$，使得 $\mathrm{SUC}(\mathcal{C}_2, f''_{m_2}) > \Omega(\delta^3/m_2) \ge 2^{-c_2\rho m}$，且 $\mathcal{C}_2$的规模为 $s_2 = s_3 \cdot (m_2/\delta)^{O(1)} \le 2^{c_2 \rho m}$，其中某个常数 $c_2 > 0$。2. 根据引理 4.10，现在令 $\gamma = c_2\rho$，我们可以在空间 $O(m)$ 内构造一个电路 $\mathcal{C}_1$，使得 $\mathrm{SUC}(\mathcal{C}_1, f'_{m_1}) > 0.99$，且 $\mathcal{C}_1$的规模为 $s_1 = s_2 \cdot 2^{O(\gamma m_1)} \le 2^{c_1 \rho m}$，其中某个常数 $c_1 > 0$。1. 根据引理 4.6，我们可以在空间 $O(m)$ 内构造一个电路 $\mathcal{C}$，使得对于每个$x \in \{0,1\}^m$，$\mathcal{C}(x) = f_m(x)$，且 $\mathcal{C}$ 的规模为 $s = s_1 \cdot m^{O(1)} \le 2^{c_0 \rho m}$，其中某个常数 $c_0 > 0$。通过选择$\rho = \varepsilon/c_0$，我们得到最终结果。

□

5 BPL 的通用去随机化

这里我们陈述本节的主要定理：存在一个用于对数空间计算的通用去随机化器。

定理 5.1. 存在一个确定性机器 UnivDerand，使得：

- 输入 1^n 和一个长度与宽度均至多为 n的有序分支程序 B 时，输出 $\delta := \text{UnivDerand}(1^n, B)$，满足 $|\delta - \mathbb{E}[B]| < n^{-1}$。
- 对于每个空间可构造函数 S：满足 $S(n) \geq \log n$ 的 $\mathbb{N} \to \mathbb{N}$，UnivDerand 当且仅当 prBPL $\subseteq$ SPACE$[O(S(n))]$ 时，使用空间 $O(S(n))$ 运行。

我们首先直观地解释机器 UnivDerand执行的算法。它枚举图灵机 $\langle i \rangle$ 和空间界 j。在每一步中，UnivDerand 对每个 v，在输入 $(1^n, B_{\to v})$ 上运行 $\langle i \rangle$；其中，$B_{\to v}$ 对于 $v \in V_i$ 是一个程序，它在前 i 层与 B 完全相同，然后在程序到达状态 v 时接受。如果 $\langle i \rangle$ 在工作带上使用的空间超过 j，UnivDerand就会停机，并递增 i 或 j。否则，我们得到一组估计值$\{\widetilde{p_{\to v}}\} := \{\langle i \rangle(1^n, B_{\to v})\}$（注意，我们可以使用空间 $O(j + \log n)$ 即时生成这些估计值）。然后，我们将这些估计值提交给 Cheng 和 Hoza [CH22]，的局部一致性测试；如果测试通过，就返回到达接受状态的概率估计值。

定理 5.2（[CH22]）。存在一个确定性对数空间算法 LCTest，该算法以1^n 和一个长度及宽度至多为 n 的有序分支程序（OBP）B 以及估计值 $\{\widetilde{p_{\to v}}\}_{v \in V}$作为输入。如果对于每个 v，$|\widetilde{p_{\to v}} - p_{\to v}| \leq n^{-3}$，算法均接受；此外，如果算法接受，则对于每个 v，$|\widetilde{p_{\to v}} - p_{\to v}| \leq n^{-1}$。

注意，真实概率 $p_{\to v}$ 仅出现在定理的陈述中，并不是测试算法的输入。现在我们可以将该算法正式描述为算法 4。根据测试 LCTest的可靠性，如果 UnivDerand 返回一个值，则该值必然是接受概率的良好近似，因此只需证明这种情况确实会发生（并且发生时满足所需的空间复杂度界）。

算法 4： UnivDerand（1^n，B）

```
对于 j ← 0, 1..., 执行
    对于 i ← 0, 1, ..., j, 执行
        对于 r ← 1 · n^{-5}/2, 2 · n^{-5}/2, ..., 2n^2 · n^{-5}/2, 执行
            计算 b ← LCTest(1^n, B, {⟨i⟩(1^n, B_{→v}, r)}_{v∈V(B)});
            每当 ⟨i⟩ 使用超过 j 空间或超过 2^j 时间时，执行
                中止对 ⟨i⟩ 的模拟，并转到下一个 r。
            end
            如果 b = 1，则返回 ⟨i⟩(1^n, B, r)。
        end
    end
结束
```

为此，我们依赖于如下定义的承诺搜索问题 f_c（参数为 $c \in \mathbb{N}$，为方便起见，我们将其定义为输出 $[0, 1]$ 中某个值的函数）。给定 1^n，一个

长度和宽度均至多为 n 的有序分支程序 B，以及舍入阈值 r，并承诺

$$\left|\mathbb{E}[B] - k \cdot n^{-c+2} + r\right| > \frac{n^{-c}}{6},\ \forall k \in \mathbb{Z},$$

也就是说， $\mathbb{E}[B] + r$ 与任意一个 n^{-c+2}的整数倍之间都相距至少一个多项式量级。该问题要求输出一个（伪确定性的）数值 $f_c(1^n, B, r)$，其与 $\mathbb{E}[B]$的距离不超过 n^{-c+2} 。舍入值的引入受 Saks 和 Zhou [SZ99], 的方法启发：当$\mathbb{E}[B]$ 非常接近阈值时，很难确定期望是在截断值之上还是之下。

我们在命题 5.3 中证明，对于任意$c \in \mathbb{N}$，计算 f_c 这一任务都是 promise-BPL 完全的。因此，如果 prBPL $\subseteq$ SPACE[$S(n)$], ，那么存在一台机器 $\langle i \rangle$ ，能够在空间$j = O(S(n))$内计算 f_c 。最后，为了处理舍入阈值的存在， UNIVDERAND还会枚举 r的多项式数量个候选值。我们证明，存在一个合适的 $c \in \mathbb{N}$ ，使得对于每个 B，都存在一个满足 f_c 承诺的良好 r 。这基本上是通过 Saks 和 Zhou [SZ99]的论证证明的。因此，该算法总能找到一个三元组（i, j, r），从而对每个 v 都能得到 $\mathbb{E}[B_{\to v}]$ 的良好估计值，机器也就会停机并返回正确的值。

命题 5.3. 对于每个 $c \in \mathbb{N}$，令 f_c 表示这样的问题：给定 1^n 和一个长度及宽度均至多为 n 的有序分支程序 B，以及 $r \in [0,1]$，使得对于每个 $k \in \mathbb{Z}$,

$$\left|\mathbb{E}[B] - k \cdot n^{-c+2} + r\right| > \frac{n^{-c}}{6},$$

以至少 $2/3$ 的概率返回满足 $|\mathbb{E}[B] - \delta| \le n^{-c+2}$ 的同一个数 δ。那么， f_c 在 L 归约下是 prBPL 完全的。

证明概要。固定任意的 $c \in \mathbb{N}$。我们首先证明 $f_c \in$ 属于 prBPL。令 $\mathcal{R}(1^n, B, r)$ 为一个算法，该算法从 v_{st} 出发，在 B 上进行 n^{2c+1} 次随机游走，并令 γ 表示其中到达 v_{acc} 的随机游走所占的比例。令 $k \in \mathbb{Z}$ 为满足 $\gamma + r \ge k \cdot n^{-c+2}$ 的最大值，并返回 $\delta = k \cdot n^{-c+2}$。由于该算法显然在随机对数空间内运行，只需证明：对于满足承诺的 B 和 r，存在某个固定的 k，使得 $\mathcal{R}$ 以超过 $2/3$ 的概率识别出它。注意，根据承诺，对于某个 $k_0 \in \mathbb{Z}$，我们有，

$$k_0 \cdot n^{-c+2} + \frac{n^{-c}}{6} < \mathbb{E}[B] + r < (k_0 + 1) \cdot n^{-c+2} - \frac{n^{-c}}{6}.$$

另一方面，利用集中界，我们可以证明其概率至少为 $2/3$,

$$|(\mathbb{E}[B] + r) - (\gamma + r)| = |\mathbb{E}[B] - \gamma| \le \frac{n^{-c}}{6}.$$

在这种情况下， $\mathcal{R}$ 总是能识别出 $k = k_0$，因为 $k_0 \cdot n^{-c+2} < \gamma + r < (k_0 + 1) \cdot n^{-c+2}$。

现在我们证明 f_c 是 prBPL-困难的。回顾标准的 prBPL-完全问题：给定一个长度和宽度为 n 的有序分支程序 B，判断 $\mathbb{E}[B] < 1/3$ 还是 $\mathbb{E}[B] > 2/3$，承诺是这两种情况之一成立。我们如下将此问题归约到 f_c。设 T_B： $\{0,1\}^{dn} \to \{0,1\}$ 是定义如下的有序分支程序

$$T_B(x_1, \ldots, x_d) = \texttt{MAJ}(B(x_1), \ldots, B(x_d))$$

其中 $d = O$（c对数 n），满足：如果 $\mathbb{E}[B] < 1/3$，那么 $\mathbb{E}[T_B] < n^{-c}/6$；如果 $\mathbb{E}[B] > 2/3$，那么 $\mathbb{E}[T_B] > 1 - n^{-c}/6$。注意， T_B 的长度和宽度均为 $N =$ poly(n)，并且给定 B后可在确定性对数空间内构造。因此，令 f_c的输入为（1^N、T_B、n^{-c}），该输入满足 f_c的承诺；于是，如果答案小于 $1/2$，我们判定 $\mathbb{E}[B] < 1/3$，否则判定 $\mathbb{E}[B] > 2/3$。 □

我们首先证明该机器返回的值是准确的（假设该机器确实返回一个值）。

引理 5.4。对于每个 B、 $|\textsc{UnivDerand}(1^n, B) - \mathbb{E}[B]| \le n^{-1}$。

证明。这由将定理 5.2 应用于 $\widetilde{p_{\to v}} = \langle i \rangle(1^n, B_{\to v}, r)$ 得出。 □

接下来我们证明该图灵机会在所声称的空间界内停机。

引理 5.5。对于每个空间可构造函数 $S: \mathbb{N} \to \mathbb{N}$ 满足 $S(n) \ge \log n$， $\textsc{UnivDerand}$在空间 $O(S(n))$ 内运行，条件是 $\mathsf{prBPL} \subseteq \mathsf{SPACE}[O(S(n))]$。

证明。我们证明 $\textsc{UnivDerand}(1^n, B)$ 会停机，并返回一个满足 $i + j \le c \cdot S(n)$ 的值，其中 c 为绝对常数（特别地， $i, j < \infty$），这通过空间受限算法的组合即可足以证明该引理。

根据命题 5.3，存在一台图灵机 $\langle i \rangle$，在 $\mathsf{SPACE}[O(S(n))]$ 中判定语言 f_5。现在我们证明存在 $r \in \{1 \cdot n^{-5}/2, 2 \cdot n^{-5}/2, \ldots, 2n^2 \cdot n^{-5}/2\}$ 使得

$$\left|\mathbb{E}[B_{\to v}] - k \cdot n^{-3} + r\right| > n^{-5}/6 \qquad (\star)$$

对于每个 k 和 v。在分支程序的顶点集 $V(B)$ 中， v 上的 $\mathbb{E}[B_{\to v}]$ 有 n^2 个不同取值；对于每个 v，至多存在一个关于 r 的赋值，使得对于某个 $k \in \mathbb{Z}$，$(\star)$ 不成立。由于 r 有 $2n^2$ 个可能取值，因此必然存在一个取值，使得对于所有 k 和 v，$(\star)$ 都成立。

最后，令 $j = O(S(|B|))$ 满足：对于每个 v， $\langle i \rangle(1^n, B_{\to v}, r)$ 至多使用 j 的空间后停机。根据假设以及输入 $(1^n, B_{\to v}, r)$ 对每个 v 都满足命题 5.3 的承诺，这样的 j 确实存在。因此，当到达元组 (i, j, r) 时，估计值集合 $\widetilde{p_{\to v}} = \langle i \rangle(1^n, B_{\to v}, r)$ 必须对每个 $v \in V(B)$ 都满足 $|\widetilde{p_{\to v}} - \mathbb{E}[B_{\to v}]| \le n^{-3}$。随后运行 $\textsc{LCTest}(1^n, B, \{\widetilde{p_{\to v}}\}_{v \in V(B)})$（我们等待测试请求某个特定值 $\widetilde{p_{\to v}}$，然后从 $\langle i \rangle$ 重新计算该值，从而无需存储全部 n^2 个值）将导致 $\textsc{LCTest}$ 接受，因此 $\textsc{UnivDerand}$ 在所声称的空间界内停机。此外，返回值 $\delta = \langle i \rangle(1^n, B, r)$ 满足 $|\delta - \mathbb{E}[B]| \le n^{-1}$。 □

我们最后证明逆命题。

引理 5.6。对于每个空间可构造函数 $S: \mathbb{N} \to \mathbb{N}$ 满足 $S(n) \ge \log n$，如果 $\textsc{UnivDerand}$ 在空间 $S(n)$中运行，则 $\mathsf{prBPL} \subseteq \mathsf{SPACE}[O(S(n))]$。

证明。根据命题 5.3，只需使用对数空间归约将 f_3 求解为 $\textsc{UnivDerand}$。给定 $(1^n, B, r)$ 作为输入（其中 r是舍入阈值，我们将忽略它），令 $\delta := \textsc{UnivDerand}(1^n, B)$ 表示 $\textsc{UnivDerand}$ 在 B上返回的值。根据引理 5.4，我们有$|\delta - \mathbb{E}[B]| < n^{-1}$，因此 δ 是 f_3所需的确定性输出。 □

由此即可完成定理 5.1 的证明。

定理 5.1 的证明。令 $\textsc{UnivDerand}$ 为上述定义的算法。定理 5.1 由引理 5.4 得出（而它确实返回一个值这一事实由引理 5.5 得出）。定理 5.1 的充分方向由引理 5.5 得出，而必要方向由引理 5.6 得出。 □

最后，我们完成定理 1.2 的证明。

定理 1.2 的证明。设 U 是这样一个算法：给定随机化对数空间算法 R 的描述以及一个输入 x（其中 $|x| = n$），它以确定性对数空间构造有序分支程序 $B := R(x, \cdot)$，其长度和宽度至多为 $m = \text{poly}(n)$，表示 R 对其随机比特的处理过程。然后令 U 调用 UnivDerand$(1^m, B)$；如果返回值小于 $1/2$，则返回 0，否则返回 1。根据 R 上的承诺，要么 $\Pr[B(U_n) = 1] > 3/4$，要么 $\Pr[B(U_n) = 1] < 1/4$。由于在这两种情况下，我们都根据定理 5.1 将 B 的期望估计到误差 $1/n$ 以内，因此能够正确判定当前属于哪种情况；空间消耗则由定理 5.1 的空间消耗界定。

□

6 黑盒测试

现在我们证明，击中集蕴含对有序分支程序的黑盒双边去随机化。为此，我们首先形式化定义击中集和确定性采样器：

定义 6.1。给定函数类 $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}\}$，$\varepsilon$-击中集生成器（HSG）$H : \{0,1\}^s \to \{0,1\}^n$ 满足：对于每个满足 $\mathbb{E}[f] \geq \varepsilon$ 的 $f \in \mathcal{F}$，都存在 $y \in \{0,1\}^s$，使得 $f(H(y)) = 1$。如果存在一致算法，在给定 1^n 和 x 的情况下，能够在空间 $O(s)$ 中计算 $H(x)$，则称 H 是显式的。

定义 6.2。给定函数类 $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}\}$，具有空间复杂度 $s(n)$ 的 ε-（确定性）采样器 SAMP，其用于 $\mathcal{F}$，是指一种确定性算法：该算法在空间 $s(n)$ 中运行，并且在获得对 $f \in \mathcal{F}$ 的预言机访问后，向 f 发出查询并输出一个估计值 δ，满足 $|\delta - \mathbb{E}[f]| \leq \varepsilon$。

确定性采样器体现了一种去随机化算法的理念：它仅以黑盒方式访问分支程序。这一概念此前已在小空间去随机化的背景下得到研究 [HU22, CH22,PV22]。

现在我们给出定理 1.3 的正式表述。我们用对 HSG 种子长度的依赖来表述，因为我们的结果可以将命中集普遍地转换为具有相近空间复杂度的采样器。

定理 6.3. 假设存在一个一致可构造族 $\mathcal{H} = \{H_1, \ldots, \}$ where $H_n : \{0,1\}^{s(n)} \to \{0,1\}^n$，它是一个显式的 $1/2$-命中集，种子长度为 $s(n)$，适用于宽度为 n、长度为 n 的有序分支程序。则存在一个一致可计算的确定性 ε-采样器，其空间复杂度为 $O(s(nw/\varepsilon))$，适用于宽度为 w、长度为 n 的有序分支程序。

我们通过构造一种局部一致性测试来证明这一点，该测试可在对分支程序进行黑盒访问的条件下实现。

6.1 证明概述

定理 6.3 的证明依赖于构造一种局部一致性测试，该测试只需黑盒访问分支程序即可实现（而此前所有测试都要求访问程序的内部状态）。我们首先介绍如何以黑盒方式访问程序的内部状态。

给定一个分支程序 $B : \{0,1\}^n \to \{0,1\}$ 和一个命中集 $H : \{0,1\}^s \to \{0,1\}^n$，对于每个种子 $x \in \{0,1\}^s$ 和层 $i \in [n]$，程序在输入 $H(x)_{1..i}$上到达某个状态 v。我们可以通过记下 (x, i)，以黑盒方式对该状态进行索引。然而，由于可能有许多种子到达同一个状态 v，我们希望将这些重复项重新合并。由于我们无法检查程序的层 i，因此可以尝试测试 x 和 x' 是否到达同一个状态，方法是代入每个 HSG 输出，观察从 (x, i) 和 (x', i) 开始的程序是否表现不同。

定义 6.4（定义 6.9 的非正式表述）。对于 x、$x' \in \{0,1\}^s$ 和 $i \in [n]$，如果对于每个 $y \in \{0,1\}^s$，元组 (x, i) 和 (x', i) 无法区分，

$$B(H(x)_{1..i}H(y)_{1..n-i}) = B(H(x')_{1..i}H(y)_{1..n-i}).$$

不可区分元组并不总是会到达同一状态。不过，Cheng 和 Hoza 能够证明如下结论：

引理 6.5（[CH22]（非正式））。假设状态 v 和 v' 由不可区分元组到达。那么，从 v 开始时在 B 中接受的概率，与从v'开始接受的概率相近。

因此，从第 i 层起，这些状态具有相似的行为。遗憾的是，不可区分状态并不总是具有不可区分的出边。因此，若天真地尝试通过查询访问学习该程序，就会同时输出两条出边，并得到一个非确定性分支程序；该模型已被证明是 NL-困难的，无法进行去随机化。同样，目前尚不清楚如何选择一条单独的边来输出，同时保持程序的接受概率。在常宽度情形下，Cheng 和 Hoza [CH22] 通过在构造第 i层时记住关于第 $i+1$ 层中每个状态的 $O(\log n)$ 比特信息来规避这一问题，从而使他们能够选择一条好的出边。然而，这对于超常数宽度似乎不可行。相反，我们开发了一种局部一致性测试，能够容忍将不可区分状态合并。

假设对于每个元组 (x, i)，我们都有一个估计值 $\tilde{p}_{x,i}$，据称它接近从 $v := B[v_{st}, H(x)_{1..i}]$接受的真实概率。

定义 6.6（黑盒局部一致性测试（非正式））。给定对有序分支程序 $B\colon \{0,1\}^n \to \{0,1\}$ 和命中集 $H\colon \{0,1\}^s \to \{0,1\}^n$ 的黑盒访问，以及估计值 $\{\tilde{p}_{x,i}\}_{x\in\{0,1\}^s, i\in[n]}$，验证以下条件是否成立：

1. 对于每一对不可区分元组 (x, i)、(x', i)，都有 $|\tilde{p}_{x,i} - \tilde{p}_{x',i}| \le O(\varepsilon)$。
2. 对于每个元组 (x, i)，令 $(x_0, i+1)$ 和 $(x_1, i+1)$ 为分别与 $B[v_{st}, H(x)_{1..i}0]$ 和 $B[v_{st}, H(x)_{1..i}1]$ 不可区分的任意元组。那么

$$\left|\tilde{p}_{x,i} - \frac{\tilde{p}_{x_0,i+1} + \tilde{p}_{x_1,i+1}}{2}\right| \le O(\varepsilon).$$

如果所有这些条件都成立，则输出估计值 $\tilde{p}_{0,\ 0}$；否则拒绝。

我们认为，所有测试都包含完备性和可靠性两部分：完备性意味着，如果一组估计值与真实概率足够接近，就保证测试通过；可靠性意味着，测试通过则表明返回的估计值接近真实值（具体参数将在后文讨论）。

在空间复杂度类 $O(s + \log n)$ 中实现定义 6.6 的测试并不太困难，只需给定 H 和对 B 的黑盒访问，因为我们可以枚举命中集中的种子以及程序中的各层；而且所有这类测试都是“局部的”，也就是说，它们至多涉及两层和常数个种子。

如果每个状态都能与其他所有状态区分开，并且 H 命中程序中的每个状态，那么定义 6.6 中的测试就等价于以下白盒局部一致性测试：

定义 6.7（白盒局部一致性测试（非正式））。

1. 对于每个 v，来自 v 的所有接受概率估计值之间的差异必须在 ε 以内 .

2. 对于每个 v，来自 v、$v_0 := B[v,0]$ 和 $v_1 := B[v,1]$ 的接受概率估计值（分别记作 $\widetilde{p_v}$、$\widetilde{p_{v_0}}$ 和 $\widetilde{p_{v_1}}$）必须满足 $\widetilde{p_v} \approx (\widetilde{p_{v_1}} + \widetilde{p_{v_0}})/2$。

如果所有此类条件都满足，则输出任意估计值 $\widetilde{p_{v_{st}}}$，否则拒绝。

定义 6.7 中的测试显然会在这些估计值恰好等于（或在 $\varepsilon/2$ 以内接近）从每个顶点接受的真实概率时接受。同样，证明可靠性也并不困难。然而，定义 6.6 中测试的这一理想化版本并不会完全实现，原因有二：

1. 我们可能会对到达不同但不可区分状态的元组施加项目 (1) 的检查；对于项目 (2) 中的 0 状态和 1 状态，也同样如此。2. 如果命中集输出的字符串没有到达后面两个状态中的任何一个，我们可能无法在 v、$B[v,0]$ 和 $B[v,1]$ 之间施加项目 (2) 的检查。

问题 (1) 必须在完备性的证明中处理（因为我们会加入白盒测试器中没有的一些测试），而问题 (2) 必须在可靠性的证明中处理（因为我们有时无法施加本应存在的测试）。问题 (1) 比较容易处理。根据引理 6.5，不可区分状态具有相近的接受概率，因此，对不可区分顶点的接受概率进行良好估计时，它们之间仍会相差不超过 $O(\varepsilon)$。相比之下，问题 (2) 似乎确实给可靠性带来了问题。对于状态 $v := B[v_{st}, H(x)_{1..i}]$，如果不存在种子 x' 使得 $B[v_{st}, H(x')_{1..i+1}] = B[v,0]$，我们可能无法运行局部一致性测试来验证 $\tilde{p}_{x,\,i}$。事实上，从 v 接受的概率估计值可能任意错误，而我们没有能力检测出来。然而，我们注意到，每个这样的 v 从初始状态到达的概率都很低。这是因为，如果没有 (ε-)HSG 输出到达 $B[v,0]$，则 v 从初始状态被到达的概率至多为 2ε。但随后，从 v 接受概率的一个非常糟糕的估计，至多只会使总体接受概率改变 $O(\varepsilon)$（并且可以同时对所有未验证状态运行这样的论证）。最终，我们能够证明，缺少这些检查只会使总体误差增加 $O(\varepsilon)$，这是可以容忍的。

综上，我们展示了一个黑盒测试器：给定对每个 x 和 i，从 $B[v_{st}, H(x)_{1..i}]$ 接受的概率的估计值 $\tilde{p}_{x,\,i}$，它要么输出该程序期望的近似值，要么拒绝该输入。最后，我们采用 Cheng 和 Hoza 的一个想法，通过命中集找到一组合适的估计值 $\tilde{p}_{x,\,i}$。首先，为了让非平凡但非最优的命中集生成器也能得到更好的结果，我们稍微修改测试器，使其接收 $n \cdot w$ 个估计值；这些估计值（本质上）分别对应原始分支程序中每个状态的接受概率估计值。然后我们证明（本质上沿用 [CH22] 的论证），存在一个长度为 poly(nw/ε)、宽度为 poly(nw/ε) 的分支程序 T，它将输入划分为 $n \times w$ 个块，并将标记为 v 的块用作长随机字符串，以估计程序中每个状态 v 的 $\tilde{p}_v$；如果所有这些估计值与真实接受概率之间的差都不超过 ε，则程序接受。该程序利用真实概率来检查样本的经验平均值与真实值之间的差是否不超过 ε；但我们无需显式构造这个程序——只需知道它存在，因此我们的 HSG 族中会有某个字符串击中它。最后，我们论证可以通过预言机访问 B，而非 T，来计算相应的经验平均值。击中 T 的字符串会为每个 v 生成良好的估计值 $\tilde{p}_v$，而我们的黑盒测试器会接受这些估计值。然后，我们只需枚举命中集字符串，并返回第一个被接受的估计值。

6.2 黑盒局部一致性测试

现在正式陈述黑盒局部一致性测试：

定理 6.8。存在一个确定性空间 $O(s+\log n)$ 算法，该算法给定一个显式的 ε-HSGH：$\{0,1\}^s \to \{0,1\}^n$ 针对长度为 n、宽度为 w^2 的分支程序，具有对宽度为 w 、长度为 n的 OBP B 的预言机访问，以及估计值 $\{\tilde{p}_{x,i}\}_{x\in\{0,1\}^s,\ i\in[n]}$，要么输出一个值，要么拒绝。此外：

1. 如果对于每个 x，都有 $|p_{v\to} - \tilde{p}_{x,\ i}| \le 2\varepsilon$ 其中 $v = B[v_{st}, H(x)_{1..i}]$ 对于每个 $i < n$ 和 $\tilde{p}_{x,\ n} = B(H(x))$，那么算法会输出一个值。
2. 如果算法输出 δ，那么 $|\mathbb{E}[B] - \delta| \le 6\varepsilon n$。

我们指出，尽管这是一个黑盒测试，而 Cheng 和 Hoza 提出的是白盒局部一致性测试，但它的可靠性损失更低（为 εn ，而非 εnw）；当分支程序的宽度远大于长度时，这一点尤为重要。值得注意的是，在 Nisan-Zuckerman 区域 [NZ96] 中获得最优误差采样器（该区域已知存在最优误差击中集）是一个众所周知的开放问题。遗憾的是，我们没有得到这一结果，因为我们关于如何利用命中集获得良好接受概率估计值的论证（引理 6.20）要求命中集适用于长度为 $nw \gg n$ 的有序程序。

我们首先定义与使用 H 遍历分支程序相关的记号：

定义 6.9. 对于每个 $x \in \{0,1\}^s$ 和 $i \in [n]$，令

$$v_i(x) := B[v_{st}, H(x)_{1..i}].$$

注意，这意味着如果 $B[v_{st}, H(x)_{1..i}] = B[v_{st}, H(x')_{1..i}]$，则 $v_i(x) = v_i(x')$，也就是说，这两个种子在第 i 层到达同一个顶点。为方便起见，我们记作 $p_{x,\ i} := p_{v_i(x)\to}$。此外，对于状态 u、$u' \in V_i$，如果这两个状态在 H 下不可区分，即对于所有 y，我们记作 $u \sim u'$，

$$B[u, H(y)_{1..n-i}] = B[u', H(y)_{1..n-i}].$$

现在我们可以定义该算法实现的一致性测试。

定义 6.10（局部一致性测试）。给定 B 和 H 以及估计值 $\tilde{p}_{x,\ i}$，测试如下：

1. 对于每个 x、$i \in \{0,1\}^s \times [n]$，以及满足 $B[v_i(x)\ 0] \sim v_{i+1}(x_0)$ 和 $B[v_i(x)\ 1] \sim v_{i+1}(x_1)$ 的每个 x_0、$x_1 \in \{0,1\}^s$，要求

$$\left|\tilde{p}_{x,i} - \frac{\tilde{p}_{x_1,i+1} + \tilde{p}_{x_0,i+1}}{2}\right| \le 5\varepsilon.$$

2. 对于每个 x、$x' \in \{0,1\}^s$ 以及 $i \in [n]$ 使得 $v_i(x) \sim v_i(x')$，要求 $|\tilde{p}_{x,i} - \tilde{p}_{x',i}| \le 5\varepsilon$.
3. 对于每个 $x \in \{0,1\}^s$，要求 $\tilde{p}_{x,\ n} = B(H(x))$。

请注意，给定 H、对 B 的预言机访问以及估计值 $\tilde{p}_{x,\ i}$，我们可以在空间 $O(s+\log n)$ 中计算所有此类测试。我们首先证明该测试是完备的：

引理 6.11. 假设对于每个 x, $|\tilde{p}_{x,\ i} - p_{x,\ i}| \le 2\varepsilon$ 对于 $i < n$ 和 $\tilde{p}_{x,\ n} = B(H(x))$。则定义 6.10 的测试通过。

为证明这一点，我们需要以下结论，该结论由关于集合差的质量的论证推出。

断言 6.12（引理 3.2 [CH22]）。如果 $v \sim v'$，那么 $|p_{v\to} - p_{v'\to}| \le \varepsilon$。

然后我们可以证明完备性。

引理 6.11 的证明。考虑任意一个项目 1 测试：

$$\left|\tilde{p}_{x,i} - \frac{\tilde{p}_{x_1,i+1} + \tilde{p}_{x_0,i+1}}{2}\right|$$

令 $v := v_i(x)$，对于 $b \in \{0,1\}$，令 $v_b := B[v,b]$ 表示从状态 v 沿边 b 出发后实际到达的状态。然后，对于 $b \in \{0,1\}$，令 $u_b := v_{i+1}(x_b)$ 表示 x_b 在层 $i+1$ 中到达的状态。注意，u_b 不一定等于 v_b，因为我们可能混淆了层 $i+1$ 中的不同状态，但是$u_b \sim v_b$。因此：

$$\begin{aligned}
\left|\tilde{p}_{x,i} - \frac{\tilde{p}_{x_1,i+1} + \tilde{p}_{x_0,i+1}}{2}\right| &\le 4\varepsilon + \left|p_{v\to} - \frac{p_{u_0\to} + p_{u_1\to}}{2}\right| && \text{(Assumption)} \\
&\le 5\varepsilon + \left|p_{v\to} - \frac{p_{v_0\to} + p_{v_1\to}}{2}\right| && \text{(Claim 6.12)} \\
&= 5\varepsilon.
\end{aligned}$$

项目 2 的证明是类似的，同样使用断言 6.12，而项目 3 则是直接的。 □

现在我们证明可靠性。关键问题在于处理状态 v，因为不存在满足 $v = v_i(x)$ 的 x，所以我们无法保证这些状态的一致性。然而，这些状态恰恰是 HSG 未能命中的状态，这必然意味着从起始顶点到达这些状态的概率很低，因此它们的估计值出错所造成的影响很小。

引理 6.13. 假设定义 6.10 中的测试通过，且估计值为 $\tilde{p}_{x,i}$。那么 $|\tilde{p}_{0,0} - p_{v_{st}\to}| \le 6\varepsilon n$。

为了证明引理 6.13，我们首先定义未验证的状态：

定义 6.14. 对于每个 $x \in \{0,1\}^s$ 和 $i < n$，若存在某个 $b \in \{0,1\}$ 使得不存在满足 $B[v,b] = v_{i+1}(x')$ 的 $x' \in \{0,1\}^s$，则称 $v = v_i(x)$ 为未验证状态；否则，称 v 为已验证状态。对于每个 x，令 $v_n(x)$ 为已验证状态。注意，对于未验证状态，仍可能存在满足 $B[v,b] \sim v_{i+1}(x')$ 的 x'，但我们在可靠性证明中不会用到这一点。

我们首先证明，到达未验证状态的概率很小。

引理 6.15。令 T 表示在 B 中到达未验证状态的事件。则 $\Pr[T(U_n) = 1] < 2\varepsilon$。

证明。令 R 为宽度为 $w+1$ 的程序，该程序与 B 相同，只在且仅在到达一个未被 HSG 命中的状态时接受。根据 HSG 的良好性，我们有 $\Pr[R(U_n) = 1] < \varepsilon$。此外，在已到达 B 中的未验证状态的条件下，到达一个 HSG 未命中的状态的概率至少为 $1/2$。因此，$\varepsilon > \Pr[R(U_n) = 1] \ge \Pr[T(U_n) = 1]/2$。 □

现在我们构造一个分支程序，使得未验证状态的估计值与这些状态的真实概率一致。

引理 6.16。存在一个有序分支程序 $Q: \{0,1\}^n \to \{0,1\}$，其顶点集包含 B 的顶点集，并满足：

1. $|\mathbb{E}[Q] - \mathbb{E}[B]| \le 2\varepsilon$。
2. 当限制在已验证状态之间的边，以及从已验证状态指向未验证状态的边时，Q 与 B 相同。

3. 对于 B 中的每个未验证状态 v，对于每个满足 $v = v_i(x)$ 的 $x \in \{0,1\}^s$，我们有 $|\tilde{p}_{x,\ i} - q_{v\to}| \le 5\varepsilon$，其中 $q_{v\to}$ 是从 Q 中的 v 接受的概率。

证明。我们首先构造 Q。令 N 为 B 的未验证状态集合。对于第 i 层中的每个 $v \in N$，注意，对于所有满足 $v = v_i(x) = v_i(x')$ 的 x、$x' \in \{0,1\}^s$，根据条目 2，我们有 $|\tilde{p}_{x,\ i} - \tilde{p}_{x',\ i}| \le 5\varepsilon$。令 $q_{v\to}$ 为一个满足 $|q_{v\to} - \tilde{p}_{x,\ i}| \le 5\varepsilon$ 的数，对所有这样的 x 均成立。现在，我们通过将从 v 出发的两条边都连接到一组新的（复杂度任意）状态，来修改 B，使得 v 的接受概率恰好为 $q_{v\to}$、[3]；并对每个未验证的 v 都这样做。令 Q 为这个新的分支程序。显然，按构造可知 Q 满足性质 2。此外，根据引理 6.15，性质 1 也成立。 □

现在，我们通过证明估计值 $\tilde{p}_{x,\ i}$ 与修改后的程序 Q 一致，来证明引理 6.13。

引理 6.17。令 Q 如引理 6.16 中所定义。则对于每个 $v \in B$ 以及满足 $v = v_i(x)$ 的每个 x，我们有 $|\tilde{p}_{x,\ i} - q_{v\to}| \le 5\varepsilon \cdot (n-i)$。特别地，$|\tilde{p}_{0,\ 0} - q_{v_{st}\to}| \le 5\varepsilon n$。

证明。根据定义 6.10 的项目 3（以及所有最终层状态均未被修改这一事实），情况 $i = n$ 成立。现在假设该结论对层 $i+1$ 成立。那么对于层 i 中的每个 $v = v_i(x)$，有两种可能：

- 情况 1：v 未验证。在这种情况下，根据引理 6.16，$|\tilde{p}_{x,\ i} - q_{v\to}| \le 5\varepsilon$。
- 情况 2：v 已验证。对于 $b \in \{0,1\}$，令 $v_b := B[v,b]$，并令 x_b 满足 $v_b = v_{i+1}(x_b)$（注意，由于我们不处于情况 1，x_0、x_1 确实存在）。于是：

$$\begin{aligned}
|\tilde{p}_{x,i} - q_{v\to}| &= \left|\tilde{p}_{x,i} - \frac{q_{v_0\to} + q_{v_1\to}}{2}\right| \\
&\le 5\varepsilon + \left|\frac{\tilde{p}_{x_0,i+1} + \tilde{p}_{x_1,i+1}}{2} - \frac{q_{v_0\to} + q_{v_1\to}}{2}\right| && \text{(Item 1)} \\
&\le 5\varepsilon + \frac{1}{2}\left|q_{v_0\to} - \tilde{p}_{x_0,i+1}\right| + \frac{1}{2}\left|q_{v_1\to} - \tilde{p}_{x_1,i+1}\right| \\
&\le 5\varepsilon \cdot (n-i) && \text{(Induction.)}
\end{aligned}$$ □

现在我们完成引理 6.13 的证明。

引理 6.13 的证明。我们有：

$$\begin{aligned}
|\,\mathbb{E}[B] - \tilde{p}_{0,0}| &\le 2\varepsilon + |\,\mathbb{E}[Q] - \tilde{p}_{0,0}| && \text{（引理 6.16）} \\
&\le 2\varepsilon + 5\varepsilon n && \text{（引理 6.17）。}
\end{aligned}$$ □

现在我们完成定理 6.8 的证明。

定理 6.8 的证明。给定 $H\colon\ \{0,1\}^s \to \{0,1\}^n$ 以及估计值 $\{\tilde{p}_{x,i}\}_{x\in\{0,1\}^s,\ i\in[n]}$，我们按照定义 6.10 中的规定运行测试。所有这类测试都可以在空间复杂度为 $O(s + \log n)$ 的空间内实现，解释如下。给定 $x \in \{0,1\}^s$ 和 $b \in \{0,1\}$，我们可以通过枚举 $y \in \{0,1\}^s$ 并计算，来确定 $x' \in \{0,1\}^s$ 是否满足 $B[v_{st}, H(x)_{1..i}b] \sim v_{i+1}(x')$

$$\bigwedge_{y\in\{0,1\}^s} \mathbb{I}[B(H(x)_{1..i}(x)bH(y)_{1..n-i-1}) = B(H(x')_{1..i+1}H(y)_{1..n-i-1})].$$

[3]严格来说，若不将其构造成 Q 概率分支程序，这可能无法实现。不过，这种构造纯粹是为了分析概率 $q_{v\to}$，因此我们忽略这个小问题。

在可以黑盒访问 B 的情况下，这可以在空间复杂度为 $O(s+\log n)$ 的空间内实现，因此我们可以在所需的空间界内确定要运行哪些项目 1 测试。类似的推理也适用于条目 2 和项目 3 测试。

最后，如果所有此类测试均通过，则输出 $\tilde{p}_{0,\ 0}$。根据引理 6.11，我们有完备性条件成立；根据引理 6.13，我们有可靠性条件成立。 □

6.3 将上述内容整合起来

现在我们根据定理 6.8 证明定理 6.3。剩下需要证明的是，我们可以使用命中集生成一组良好的估计值 $\{\tilde{p}_{x,i}\}$。我们首先证明，可以修改定理 6.8，使其只接收 nw 个估计值，而不是 $n\cdot 2^s$。定理 1.3 并不要求这样做，但当 H 是一个高度非平凡但非最优的命中集时，这会改善参数。

为此，我们首先注意到，不可区分关系会在种子上诱导出一组等价类：

定义 6.18。给定宽度为 w、长度为 n 的 OBP B 和 $H\colon \{0,1\}^s \to \{0,1\}^n$，对于每个 i，令 $C_{1,\ i}, \ldots, C_{w,\ i} \subset \{0,1\}^s$ 表示 $\{0,1\}^s$ 在不可区分关系 $x \sim_i x'$ 当且仅当 $v_i(x) \sim v_i(x')$ 下的等价类（其中可能有一些为空）。令 $y_{j,\ i} \in \{0,1\}^s$ 表示 $C_{j,\ i}$ 的字典序最小元素，并且不失一般性地假设对于每个 i 都有 $y_{1,\ i} < y_{2,\ i} < \cdots < y_{w,\ i}$。此外，令 $v_{j,\ i} := v_i(y_{j,\ i})$。注意，给定 B 和 H，$y_{j,\ i}$（因此也就是能够到达 $v_{j,\ i}$ 的 HSG 输出）可以在空间 $O(s+\log\ n)$ 中构造，给定 i、j。

推论 6.19。存在一个确定性空间 $O(s+\log n)$ 算法，给定一个显式的 ε-HSG$H\colon\ \{0,1\}^s \to \{0,1\}^n$ 针对长度为 n、宽度为 w^2 的分支程序，以及对一个宽度为 w、长度为 n的有序分支程序 B 的预言机访问，还有估计值 $\{\tilde{p}_{j,i}\}_{j\in[w],\ i\in[n-1]}$，该算法要么输出一个值，要么拒绝。此外：

1. 如果对每个 j,i 都有 $|p_{v_{j,\ i}\to} - \tilde{p}_{j,\ i}| \le \varepsilon$，其中 $v_{j,\ i}$ 的定义见定义 6.18，那么该算法会输出一个值。
2. 如果该算法输出 δ, $|\mathbb{E}[B]-\delta| \le 6\varepsilon n$。

证明。测试器只需接收估计值 $\tilde{p}_{j,\ i}$，它对应于 $p_{v_{j,\ i}\to}$；将其复制为第 i 层第 j 个等价类中每个种子的估计值，针对每个 $x\in\{0,1\}^s$ 精确计算 $p_{x,\ n} := B(H(x))$，然后运行定理 6.8。显然，如果测试器输出一个值，该值与 δ 相差不超过 $6\varepsilon n$，因为我们只是将输入限制为定理 6.8 所用的输入。此外，注意，对于等价类 $C_{j,\ i}$ 中任意的 $v := v_i(x)$，由断言 6.12 可得：

$$|p_{v\to} - \tilde{p}_{j,i}| \le \varepsilon + |p_{v_{j,i}\to} - \tilde{p}_{j,i}| \le 2\varepsilon$$

因此，如果对所有 i 和 j，$|p_{v_{j,\ i}\to} - \tilde{p}_{j,\ i}| \le \varepsilon$ 都成立，我们就满足定理 6.8 的完备性条件，因此测试器将返回一个值。 □

现在我们论证存在一个击中集字符串，可用于为 $p_{v_{j,\ i}\to}$ 生成良好的估计值，其中 $v_{j,\ i}$ 的定义见定义 6.18。关于此类输出存在性的论证，只需对 Cheng 和 Hoza [CH22] 中的证明稍作修改即可；该证明表明存在一个 HSG 输出，能导出满足其局部一致性测试的估计值。

引理 6.20 ([CH22])。对于每个长度为 n、宽度为 w 且满足 $H\colon \{0,1\}^s \to \{0,1\}^n$ and$\varepsilon > 0$ 的有序分支程序 B，存在 $t = O(\log(nw)/\varepsilon^2)$，以及一个定义如下、长度和宽度均为 poly$(nw/\varepsilon)$ 的有序分支程序 EST：$\{0,1\}^{n\times w\times tn} \to \{0,1\}$：

$$\mathtt{EST}(z_{1,1},\ldots,z_{w,n}) = \bigwedge_{i\in[n],j\in[w]} \mathtt{EST}_{j,i}(z_{j,i})$$

其中，$\mathrm{EST}_{j,\ i}(z_{j,\ i})$ 的计算方式如下。它将 $z_{j,\ i}$ 视为 t 个长度为 n 的样本，并计算

$$\mathrm{EST}_{j,i}(s_1,\ldots,s_t)=\mathbb{I}\left[\Pr_{k\in[t]}[B[v_{j,i},s_k]=v_{acc}]\in[p_{v_{j,i}\rightarrow}-\varepsilon,p_{v_{j,i}\rightarrow}+\varepsilon]\right].$$

其中 $v_{j,i}$ 按照定义 6.18 根据 H 定义。于是 $\mathbb{E}[\mathrm{EST}]>1/2$，并且对于每个满足 $\mathrm{EST}(z)=1$ 的 z，对于每个 j，i，使用 z 的区块 $z_{j,i}$ 中的样本，根据 $v_{j,i}$ 估计接受概率，所得估计值的加性误差至多为 ε。

证明概要。显然，通过复制从 $v_{j,i}$ 开始的 B 子程序，并使用额外的 $O(\log(t))$ 比特内存统计满足条件的试验次数，再在最终计数落入指定范围时接受，即可用具有所述长度和宽度的有序分支程序计算 $\mathrm{EST}_{j,i}$。因此，可以在所述空间复杂度界内计算合取 EST。接着，我们选择足够大的 t，使得随机输入以压倒性概率满足所有这些检查。我们注意到，EST 是根据精确的接受概率定义的，而测试器并不知道这些概率；但我们只需要该程序存在，而不需要能够构造它。

□

于是，定理 6.3 的证明得出。

定理 6.3 的证明。通过标准归约（例如参见 [CH22]），$\mathcal{H}$ 蕴含了一个针对长度为 n、宽度为 w OBP 的 ε-击中集显式族，其种子长度为 $s(\mathrm{poly}(nw/\varepsilon))=O(s(nw/\varepsilon))$（其中，最后一个等式成立是因为对于任意 $s(n)=\Omega(\log^2 n)$，由 Nisan 伪随机生成器存在这一事实可知该定理是平凡的，因此我们可以假设情况并非如此）。

设 H 是针对长度为 n、宽度为 w^2 OBP 的 $\varepsilon/6n$-HSG，其种子长度为 $O(s(nw/\varepsilon))$。设 H_2是针对长度为 $n^2wt=\mathrm{poly}(nw/\varepsilon)$、宽度为 $(nw/\varepsilon)^c$ OBP 的 $1/3$-HSG，其中 t 如引理 6.20 中所述，且 $\varepsilon:=\varepsilon/6n$。根据参数的选择，$H_2$ 的种子长度为 $s_2:=O(s(nw/\varepsilon))$。采样器枚举每一个 $z\in\{0,1\}^{s_2}$。对于每个这样的 z，采样器调用推论 6.19 的测试器；当测试器需要关于 $p_{v_j、\ i\rightarrow}=\mathbb{E}_{r\sim_R\{0,1\}^{n-i}}[B[v_{j,i},r]]$ 的估计值 $\tilde{p}_j、\ i$ 时，我们使用 $H_2(z)$ 的 j、i 块（如引理 6.20 中的 $\mathrm{EST}_j、\ i$ 所做）来计算该估计值。注意，我们可以在对数空间中找到$y_j、\ i$，即等价类 j 中第 i 层的字典序最小种子，并且根据定义$v_j、\ i=v_i(y_j、\ i)$。因此，我们可以通过枚举 $H_2(z)$ 的 j、i 块中的样本 s_1、.....、s_t 来计算 $\tilde{p}_j、\ i$，并返回

$$\mathop{\mathbb{E}}_{k\in[t]}[B(H(y_{j,i})_{1..i}s_k)]=\mathop{\mathbb{E}}_{k\in[t]}[B[v_{j,i},s_k]].$$

如果测试器接受，则返回测试器输出的值；否则递增 z。根据空间受限算法的组合，空间复杂度为 $O(\log(n)+s_2)$。

现在假设采样器返回一个值。根据定理 6.8 的条目 2，返回的估计值与真实期望相差不超过 $(\varepsilon/6n)6n=\varepsilon$。为证明采样器返回一个值，注意到根据定理 6.8 的条目 1，只需说明我们向测试器提供一系列输入 $\{\tilde{p}_{j,i}\}$，使得$|p_{v_{j,\ i}\rightarrow}-\tilde{p}_{j,\ i}|\le(\varepsilon/6n)$ 对每个 i,j 都成立。但这些输入恰好就是由字符串 x生成的估计值，其中 $\mathrm{EST}(x)=1$；根据参数的选择，H_2 会命中该程序，因此得证。 □

参考文献

[ACR96] Alexander E. Andreev、Andrea E. F. Clementi 和 Jos´e D. P. Rolim。《命中集对 BPP 进行去随机化》。载于 Friedhelm Meyer auf der Heide 和 Burkhard Monien（编），《自动机、语言与程序设计：第 23 届国际研讨会 ICALP96》，德国帕德博恩，1996年7月8日至12日，会议论文集，《计算机科学讲义》第 1099 卷，第 357–368 页。Springer，1996 年。

[ACRT99] Alexander E. Andreev、Andrea E. F. Clementi、Jos´e D. P. Rolim 和 Luca Trevisan。《弱随机源、击中集与 BPP 模拟》。SIAM 计算杂志，28(6):2103–2116，1999 年。[AKM+20] AmirMahdi Ahmadinejad、Jonathan A. Kelner、Jack Murtagh、John Peebles、Aaron Sidford 和 Salil P. Vadhan。《小空间中随机游走的高精度估计》。载于 Sandy Irani（编），第 61 届 IEEE 计算机科学基础年会 FOCS 2020，美国北卡罗来纳州达勒姆，2020年11月16日至19日，第 1295–1306 页。IEEE，2020 年。[BF99] Harry Buhrman 和 Lance Fortnow。《概率计算中的单侧错误与双侧错误》。载于 Christoph Meinel 和 Sophie Tison（编），STACS 99：第 16 届计算机科学理论方面年会，德国特里尔，1999年3月4日至6日，会议论文集，《计算机科学讲义》第 1563 卷，第 100–109 页。Springer，1999 年。[BM84] Manuel Blum 和 Silvio Micali。《如何生成密码学强伪随机比特序列》。SIAM 计算杂志，13(4):850–864，1984年11月。[CH22] Kuan Cheng 和 William M. Hoza。《击中集为小空间提供双侧去随机化》。计算理论，18(21):1–32，2022 年。[Csa76] L. Csanky。《快速并行矩阵求逆算法》。SIAM 计算杂志，5(4):618–623，1976 年。[FK18] Michael A. Forbes 和 Zander Kelley。《适用于任意顺序的读一次分支程序伪随机生成器》。载于 Mikkel Thorup（编），第 59 届 IEEE 计算机科学基础年会 FOCS 2018，法国巴黎，2018年10月7日至9日，第 946–955 页。IEEE 计算机学会，2018 年。[GKR15] Shafi Goldwasser、Yael Tauman Kalai 和 Guy N Rothblum。《计算委托：面向麻瓜的交互式证明》。ACM 杂志（JACM），62(4):1–64，2015 年。[GL89] Oded Goldreich 和 Leonid A. Levin。《适用于所有单向函数的硬核谓词》。载于第二十一届 ACM 理论计算研讨会论文集（STOC ’89），第 25–32 页，美国纽约州纽约市，1989 年。计算机协会。[GR14] Anat Ganor 和 Ran Raz。《基于通信复杂度下界的空间伪随机生成器》。载于 APPROX-RANDOM，《LIPIcs》第 28 卷，第 692–703 页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2014 年。[GRZ23] Uma Girish、Ran Raz 和 Wei Zhan。《不可信随机性是否有帮助？》。载于第 14 届理论计算机科学创新会议 ITCS，2023 年。[GS92] Peter Gemmell 和 Madhu Sudan。《用于多项式的高鲁棒性纠错器》。信息处理快报，43(4):169–174，1992 年。[GVW11] Oded Goldreich、Salil P. Vadhan 和 Avi Wigderson。《使用击中集生成器简化 BPP 的去随机化》。载于 Oded Goldreich（编），《复杂性与密码学研究：随机性与计算交互研究杂录——与 Lidor Avigad、Mihir Bellare、Zvika Brakerski、Shafi Goldwasser 合作，

Shai Halevi、Tali Kaufman、Leonid Levin、Noam Nisan、Dana Ron、Madhu Sudan、Luca Trevisan、Salil Vadhan、Avi Wigderson、David Zuckerman，《计算机科学讲义》第6650卷，第59–67页。Springer，2011年。[HK18] William M. Hoza 和 Adam R. Klivans。为自适应算法保留随机性。载于 Eric Blais、Klaus Jansen、Jos´e D. P. Rolim 和 David Steurer 编，《近似、随机化与组合优化：算法与技术》，APPROX/RANDOM 2018，2018年8月20日至22日，美国新泽西州普林斯顿，第116卷 LIPIcs，第43:1–43:19页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2018年。[Hoz21] William M. Hoza。改进的伪分布与空间受限计算的去随机化。载于 Mary Wootters 和 Laura Sanit`a 编，《近似、随机化与组合优化：算法与技术》，APPROX/RANDOM 2021，2021年8月16日至18日，美国华盛顿大学，美国华盛顿州西雅图（线上会议），第207卷 LIPIcs，第28:1–28:23页。Schloss Dagstuhl - Leibniz-Zentrum f¨ur Informatik，2021年。[HU22] William M. Hoza 和 Chris Umans。定向伪随机生成器、模拟建议生成器与对数空间去随机化。SIAM 计算杂志，51(2):17–281，2022年。[HZ20] William M. Hoza 和 David Zuckerman。用于小成功概率RL的简单最优命中集。SIAM 计算杂志，49(4):811–820，2020年。[INW94] Russell Impagliazzo、Noam Nisan 和 Avi Wigderson。网络算法的伪随机性。载于 Frank Thomson Leighton 和 Michael T. Goodrich 编，《第二十六届 ACM 计算理论年度研讨会论文集》，1994年5月23日至25日，加拿大魁北克蒙特利尔，第356–364页。ACM，1994年。[IW97] Russell Impagliazzo 和 Avi Wigderson。如果 E 需要指数规模电路，则 P = BPP：异或引理的去随机化。载于 Frank Thomson Leighton 和 Peter W. Shor 编，《第二十九届 ACM 计算理论年度研讨会论文集》，美国得克萨斯州埃尔帕索，1997年5月4日至6日，第220–229页。ACM，1997年。[KvM02] Adam R. Klivans 和 Dieter van Melkebeek。除非多项式时间层次坍塌，否则图非同构问题有次指数规模的证明。SIAM 计算杂志，31(5):1501–1526，2002年。[LPS88] Alexander Lubotzky、Ralph Phillips 和 Peter Sarnak。拉马努金图。Combinatorica，8(3):261–277，1988年。[LV17] Chin Ho Lee 和 Emanuele Viola。小偏差分布之和的一些局限。计算理论，13(1):1–23，2017年。[MRT19] Raghu Meka、Omer Reingold 和 Avishay Tal。宽度为3的分支程序的伪随机生成器。载于 Moses Charikar 和 Edith Cohen 编，《第51届 ACM SIGACT 计算理论研讨会论文集》，STOC 2019，美国亚利桑那州凤凰城，2019年6月23日至26日，第626–637页。ACM，2019年。[Nis90] Noam Nisan。空间受限计算的伪随机生成器。载于 Harriet Ortiz 编，《第22届 ACM 计算理论年度研讨会论文集》，1990年5月13日至17日，美国马里兰州巴尔的摩，第204–212页。ACM，1990年。

[Nis93] Noam Nisan。对数空间中只读一次与多次访问随机性的比较。理论计算机科学，107(1):135–144，1993年。[NN93] Joseph Naor 和 Moni Naor。小偏差概率空间：高效构造与应用。SIAM 计算杂志，22(4):838–856，1993年。[NW94] Noam Nisan 和 Avi Wigderson。困难性与随机性。计算机与系统科学杂志，49(2):149–167，1994年10月。[NZ96] Noam Nisan 和 David Zuckerman。随机性与空间呈线性关系。计算机与系统科学杂志，52(1):43–52，1996年。[PV22] Edward Pyne 和 Salil P. Vadhan。通过查询，对无界规模图中的随机游走进行确定性近似。载于 Karl Bringmann 和 Timothy Chan 编，《第五届算法简洁性研讨会》，SOSA@SODA 2022，线上会议，2022年1月10日至11日，第57–67页。SIAM，2022年。[Rei08] Omer Reingold。对数空间中的无向连通性。ACM 会刊，55(4):17:1–17:24，2008年。[RR99] Ran Raz 和 Omer Reingold。空间受限计算中状态随机性的回收。载于 Jeffrey Scott Vitter、Lawrence L. Larmore 和 Frank Thomson Leighton 编，《第三十一届 ACM 计算理论年度研讨会论文集》，1999年5月1日至4日，美国佐治亚州亚特兰大，第159–168页。ACM，1999年。[RTV06] Omer Reingold、Luca Trevisan 和 Salil P. Vadhan。正则有向图上的伪随机游走与 RL 与 L 问题。载于 Jon M. Kleinberg 编，《第38届 ACM 年度计算理论研讨会论文集》，美国华盛顿州西雅图，2006年5月21日至23日，第457–466页。ACM，2006年。[RVW01] Omer Reingold、Salil Vadhan 和 Avi Wigderson。熵波、之字形图乘积与新型常数度扩展图。数学年刊，155(1)，2001年1月。[Sha81] Adi Shamir。密码学强伪随机序列的生成。载于 CRYPTO，第1页。加州大学圣巴巴拉分校，电子与计算机工程系，ECE 报告第82-04号，1981年。[STV01] Madhu Sudan、Luca Trevisan 和 Salil P. Vadhan。无需异或引理的伪随机生成器。计算机与系统科学杂志，62(2):236–266，2001年。[SZ99] Michael E. Saks 和 Shiyu Zhou。 $BP_HSpace(S) \subseteq DSPACE(S^{3/2})$。计算机与系统科学杂志，58(2):376–403，1999 年。[WB86] Lloyd R Welch 和 Elwyn R Berlekamp。代数分组码的纠错，1986年12月30日。美国专利 4,633,470。[Yao82] Andrew Chi-Chih Yao。陷门函数理论及其应用（扩展摘要）。载于 FOCS，第80–91页。IEEE 计算机学会，1982年。