arXiv:2601.22535v1 [cs.CC] 30 Jan 2026

基于高维扩展体的高码率高效局部列表译码

Yotam Dikstein*、Max Hopkins†、Russell Impagliazzo‡ 和 Toniann Pitassi§

2026年2月2日

摘要

我们构造了首批（局部可计算、近似）局部列表可译码码，其码率、效率和容错能力均接近信息论极限。这是复杂性理论中困难性放大任务的一个核心关注领域。我们的算法运行时间为多对数时间，且具有次对数深度；结合唯一译码（低噪声）范围内的经典构造，这解决了编码与复杂性理论中的若干长期悬而未决的问题：

1. 近乎最优的保持输入的 困难性放大（以及相应的快速 PRG） 2. 具有 $\log(N)$-深度 列表译码的常数码率码（RNC^1） 3. 保复杂度的距离放大

我们的码建立在强大的（局部谱）高维扩展图（HDX）理论之上。在技术层面上，我们作出了两项关键贡献。首先，我们提出了一个在 HDX 上进行（多对数(N) 轮次）置信传播的新框架，该框架结合局部纠错与全局扩张，在保持高码率的同时控制误差累积。其次，我们提出了 HDX 上强显式局部路由的概念，即局部算法：给定任意两个目标顶点，仅用多对数时间（并且最好具有亚对数深度）输出它们之间的一条随机路径。在某些陪集高维扩张子上构造此类方案，可以让我们在多对数时间和低深度下实例化原本的组合框架，从而完成该结果。

*高等研究院，普林斯顿。yotam.dikstein@gmail.com。获美国国家科学基金会资助，项目编号 DMS-2424441。

†高等研究院，普林斯顿。nmhopkin@ias.edu。获美国国家科学基金会资助，项目编号 DMS-2424441。

‡加利福尼亚大学圣迭戈分校。rimpagliazzo@ucsd.edu。获美国国家科学基金会资助，项目编号 CCF-2212136。§哥伦比亚大学。tonipitassi@gmail.com。获美国国家科学基金会资助，项目编号 CCF-2212136。

目录

目录 **II**

1 引言 1

1.1 动机与背景.1 1.1.1 困难性放大.1 1.1.2 保持复杂度的距离放大.2 1.2 主要结果：高码率 aLLDC.3 1.3 复杂性与编码理论中的应用.4 1.3.1 局部与并行列表解码.4 1.3.2 困难性放大与伪随机生成器.4 1.3.3 aLLDC 参数概述.6 1.4 证明概述.6 1.4.1 具有亚常数误差的 aLLDC（定理 1.1）.6 1.4.2 具有恒定码率的 aLLDC（定理 1.2）.9 1.5 相关工作.11

2 预备知识 **13**

2.1 近似局部列表可解码码.13 2.2 直积与电路复杂度.14 2.3 列表恢复.15 2.4 码级联与双重近似 DP 码.15 2.4.1 双重近似局部列表可恢复码.16 2.5 图与采样器.16 2.6 复杂性类与平均情形复杂度.17

3 次多项式码率 aLLDC **18**

3.1 列表恢复与路由.19 3.1.1 低拥塞路由.19 3.1.2 超图系统的列表恢复.20 3.2 从列表译码到列表恢复.24 3.3 子空间 aLLDC.26 3.3.1 子空间集合系统.26 3.3.2 低拥塞路由.27 3.3.3 次多项式率 aLLDC 的证明.28 3.3.4 电路实现.29

4 多对数速率 aLLDC 33

5 常数率 aLLDC **35**

5.1 形式化设置.37 5.1.1 子集内部路由.38 5.1.2 三层集合系统的电路实现.38 5.2 集合系统.39 5.3 归约到列表恢复（内层解码）.40 5.4 由常数精度进行列表恢复（外层解码）.43 5.5 归约到良分离列表.47 5.6 整合所有步骤（模电路）.49 5.7 电路实现.49

6 主要构造：KMS 码 **49**
6.1 高维扩展图预备知识.50 6.1.1 关于高维扩展体的定理与事实.50 6.2 陪集与 KMS 复形.51 6.3 KMS 复形的链接.52 6.4 路由与 KMS 复形.53 6.4.1 一种确定性路由方案.53 6.4.2 从确定性路由到低拥塞路由.55 6.4.3 子集内部路由.56 6.5 从 KMS 复形导出的集合系统.58 6.6 集合系统命题的证明.61 6.7 高维扩展体电路实现.64 6.7.1 关于陪集的另一种视角.64 6.7.2 常数速率构造的规范表示.65 6.7.3 Ki 的陪集的规范表示.67 6.7.4 KMS 复元素的电路复杂度.70 6.7.5 多对数码率构造的电路.71 6.7.6 常数速率构造的电路.72 6.7.7 逆多对数速率码的改进型编码和解码电路.73

7 下界 **74**
7.1 定理 7.1 的证明.75 7.1.1 引理 7.4 的证明.77 7.1.2 引理 7.5 的证明.79

8 参考文献 **81**

A 字母表与局部性归约 **89**
A.1 单电路（高噪声）列表恢复.90 A.2 单电路 aLLRC 级联.91 A.3 多项式规模、低错误率 aLLDC.92 A.3.1 列表译码区间.93 A.3.2 唯一译码区间.95 A.4 将所有内容整合起来.96

B 困难性放大 **97**
B.1 复杂性类内的困难性放大.97 B.2 保持近似计算时间的困难性放大.99 B.3 在伪随机生成器中的应用.101

C 编码理论中的应用 **101**
C.1 RNC1 中的列表译码.101 C.2 具有次多项式时间解码的新型局部列表可纠错码.103

D 内码分析 106D.1 列表剪枝.110

E 一些近似列表译码引理 111
E.1 采样器上的组合列表大小.111 E.2 从多种消息长度到所有消息长度.112

F 函数间距离测试 **112**

1 引言

纠错码（ECC）是一种在噪声信道上实现可靠通信的方法，长期以来一直被认为是通信理论、信息论和组合学中的基础对象。20世纪90年代，概率交互式证明系统的发展（最终促成了PCP定理）明确表明，纠错也是理论计算机科学的核心，这一深刻联系如今已遍及整个领域。反过来，理论计算机科学中的应用也推动了对具有更强性质的新型纠错码的研究。列表译码 [1–3] 的出现表明，只要允许解码器输出一个包含少量可能消息的列表，就能够从极强噪声中解码（例如，在二元情形下，最多可容忍 $\frac{1}{2}-\varepsilon$ 个差错）。局部译码 的出现表明，可以构造高度高效的译码算法：给定索引 $i \in [N]$ 和消息 f 的受损编码 w，算法只需读取其中极小的次常数 比例，就能恢复 $f(i)$。自诞生以来，这些强化概念在算法与复杂度领域（例如困难性放大 [4, 5], 、近似困难性 [6–8], 、最坏情况到平均情况归约 [9, 10]）、伪随机性（提取器 [11, 12], 、去随机化 [13, 14]）、密码学（单向函数 [15–17],、隐私信息检索 [18]）以及机器学习 [19–21]中发挥了关键作用。

尽管这类码至关重要，现有文献中最优构造与相应信息论下界之间仍存在显著差距（下游应用中也有类似情况）。在本文中，我们开发了首个接近码率、查询复杂度、效率和容错能力最优权衡的近似 局部列表可译码码（aLLDC）。借助这些码，我们解决了编码与复杂性理论中的若干长期悬而未决的问题，包括(**1**) 保持输入的困难性放大 以及近线性时间伪随机生成器，以及(**2**) 保复杂度的距离放大。后者尤其带来了首个具有低深度 （RNC^1）列表解码器的优良码构造，将 Zyablov 和 Pinsker [22] 以及 Sipser 和 Spielman [23] 的经典结果推广到高噪声区域，并构造出新一类高码率局部列表可解码码，其解码时间在次多项式意义下有所改善。

我们的构造基于面向谱方法高维扩张器这一强大类别的新型局部算法工具。高维扩张器是扩展图的超图类比物，在理论计算机科学 [24–32]中催生了众多突破性应用。我们提出了一个用于 HDX 上置信传播 的新框架，结合局部纠错与全局扩张，在 polylog(N) 轮传播过程中控制误差累积，同时保持常数 编码冗余（码率）。我们框架的核心是强显式局部路由这一新概念：给定图 G中的任意一对顶点 v_1, v_2 ，该局部算法会在多对数时间 内输出一条连接 v_1 和 v_2 的随机路径，且深度为次对数。关键的是，我们为某些 HDX 开发了局部路由方案，使其还能容忍顶点故障，即使有固定比例的顶点无法通信，仍可在多对数时间内完成路由。

1.1 动机与背景

1.1.1 困难性放大

近似局部列表译码是局部列表译码的一种放松，允许解码后的消息接近 （而非精确匹配）原始消息。这一概念自然出现在平均情形复杂度与困难性放大的研究中 [33, 34]: 通过用一个能纠正至多 $\frac{1}{2}-\varepsilon$ 个差错的 aLLDC对弱困难函数 $f\colon \{0,1\}^n \to \{0,1\}$ 进行编码，编码函数Enc^f 便自动对一类更弱的算法是 $\frac{1}{2}+\varepsilon$ 困难的（即该类中没有任何算法能以优于 ε 的优势计算Enc^f ），该类算法取决于Enc解码器的复杂度。为确保 Enc^f 的最坏情形复杂度仍与 f 的相当（从而允许在固定复杂性类内进行定量的困难性放大），进一步希望编码函数Enc是局部可计算 的（仅通过查询 f的少数取值即可确定），而这只有在放松后的近似译码模型下才可能。

在 $\varepsilon = \frac{1}{4}$-(“唯一译码区间”）以内，基于扩展图的近似局部可译码码存在简单构造 [35], ，但要将其放大到超出 $1/4$（“列表译码范围”）时，扩展图似乎就不再奏效，而且所有已知方法都会使定义域呈多项式级膨胀 [4, 13, 15, 36, 37]。事实上，即便放弃局部计算这一要求，也没有任何码率更高的近似局部列表可译码码构造优于精确码；尽管在这种情形下，将扩展图与可列表译码代数码结合取得了更大成功 [38–40], ，但目前已知的最佳构造仍需要准多项式级数量的查询 n，这大幅影响了 Enc^f 的定量困难性。因此，困难性放大中的核心问题仍悬而未决：我们能否将一个弱困难函数 $f\colon \{0,1\}^n \to \{0,1\}$

放大为 $\mathrm{Enc}^f\colon \{0,1\}^{n'} \to \{0,1\}$，其中 $n' \le n + O(\log \frac{1}{\varepsilon})$？我们能否以这样的方式实现，使 Enc^f 局部可计算，并且仍能抵抗复杂度大致与 f 相同的算法？这类结果不仅对我们定量理解平均情形复杂度很重要，也对下游应用至关重要，例如构造高效的伪随机生成器。

1.1.2 保持复杂度的距离放大

研究近似局部列表译码的第二个重要动机（Trevisan 在该主题的开创性工作 [33] 中也曾提及）是距离放大。距离放大是编码理论和复杂度领域中广泛使用的一种工具，它能增强基码 C 的纠错能力，同时保持码率等理想性质。多年来，基于扩展子的距离放大方法 [41, 42] 尤其成功地构造出了具有高效列表译码能力的高码率码 [43–46], ，其中包括目前已知的最佳高码率局部列表可译码码构造 [38, 40]。

与它们在困难性放大中的失效类似，基于扩展子的方法通常无法在高错误率区间保持解码的复杂度（例如查询次数、电路复杂度等）。[1] 相比之下，Trevisan [33] 发现，aLLDC 提供了一种通用的放大方法，不会 显著增加基码的编码或译码复杂度：只需将基码 C 与 aLLDC 复合，首先运行 aLLDC，再将每个准确率达到 99% 的输出送入 C 的解码器，即可完全（列表）解码该消息。

遗憾的是，由于所有已知的低复杂度 aLLDC 码率都趋于零，这一联系尚未促成具有高噪声容忍度与低译码复杂度的优良纠错码的构造，从而在列表译码研究中留下了若干核心问题，包括：

低深度编码与译码的列表可译码码 编码理论中的一个经典问题是构造具有低电路复杂度的优良码族，尤其是可并行编码和解码的码，即由多项式（甚至近线性）规模、$O(\log N)$深度电路实现。低深度译码器在纠错中有着悠久的历史，可追溯到60年代末 Gallager 对 LDPC码 的著名引入 [47] 以及 Zyablov 和 Pinsker 的 [22] 相应对数深度置信传播译码器（90年代由 Sipser 和 Spielman [23] 利用扩展图使之显式 而闻名）。令人惊讶的是，尽管其间已过去约30至50年，在列表译码区间（即便对于 $\varepsilon = 1/4$），人们甚至在存在性上仍不知道是否存在具有对数深度译码器的优良码。尽管许多码，例如[3, 43], 在 (V)NC^2, 中具有谱方法或代数列表译码，但尚无已知码能突破 $\log^2(N)$ 深度。

局部列表可解码码 尽管过去 30 年取得了重大进展，如何以近乎最优的参数进行局部（列表）译码，仍是一个核心的开放问题。与较容易的并行设置不同，在唯一译码区间，局部译码问题远未解决，目前已知的最佳上界与下界之间仍相差准指数级。尽管如此，一个有趣的问题是：唯一译码区间中的常数码率局部可译码码能否通过通用方法放大？例如，任何能够在 q 次查询内纠正 1% 差错的良好局部可译码码，是否都意味着存在一种良好的局部列表可译码码，能够在多项式数量的查询内纠正 $\frac{1}{2} - \varepsilon$ 个差错（其中多项式以 q 为参数）？除了提供新的局部列表可译码码族之外，这样的结果还能将问题彻底归约为解决「较容易的」唯一译码情形。

到目前为止，我们讨论了几个相关的开放问题：近乎最优的困难性放大、保持复杂度的距离放大，以及局部与并行列表译码；这些问题都归结为一个核心问题，也是我们工作的真正重点：

> **主要问题：** 是否存在「高码率」*aLLDC*？其查询/电路复杂度如何？

在保持高码率和高距离的同时实现这种形式的局部性，是复杂性理论编码中的核心挑战；此类算法必须仅凭少量局部信息中的比特推断出全局码字结构，但实现这种从局部到全局的传播通常需要牺牲码率。从技术层面看，已知的次优构造往往借助强大的组合扩张或代数结构，但这些方法很少能自然结合，从而构造出具有高效局部算法的码。

[1] 或者至少目前尚不为人所知。如果基码还具有列表可恢复性，则可以借助 [42] 在一定程度上保持译码复杂度（例如参见 [40]），但这通常是对基码强得多的条件，更接近于列表译码。

在这项工作中，我们基本上解决了这个问题。在下文第 1.2 节中，我们给出了新的 aLLDC 构造，其码率显著提高，且局部性接近信息论最优值。在第 1.3 节中，我们介绍了这些码在编码和复杂性理论中的应用，突破了该领域长期存在的障碍。在第 1.4 节中，我们讨论了所用技术：将组合方法与代数方法结合起来，并基于新码以及面向强大类别高维扩张器的局部算法工具。

1.2 主要结果：高码率 aLLDC

我们在两个主要情形下构造了新的显式码。首先研究 ε 相对于消息长度呈强次常数的情形，即 $\varepsilon \ll \frac{1}{\mathrm{polylog}(N)}$（且可以取小至 $1/\mathrm{poly}(N)$）。此类码本质上不可能具有“常数码率”，但在困难性放大中具有核心意义，其中该情形对应于将 $f\colon \{0,1\}^n \to \{0,1\}$ 放大为 $\mathrm{Enc}^f\colon \{0,1\}^{n'} \to \{0,1\}$ *super-polynomially hard*。[2] 我们在此情形下给出了参数几乎最优的码：[3]

定理 1.1（多对数码率 aLLDC（推论 4.2））。对于每个 $N \in \mathbb{N}$ 和 $\varepsilon \leq \frac{1}{\log(N)}$，存在一个二元 *aLLDC* $\mathcal{C}_N$，可从 $\frac{1}{2} - \varepsilon$ 个差错中解码，并满足：

1. 码率：多项式$(\frac{1}{\varepsilon})$
2. 列表大小：$\tilde{O}(\frac{1}{\varepsilon^2})$
3. 查询：多项式$(\frac{1}{\varepsilon})$

此外，$\mathcal{C}_N$ 可在多项式$(\frac{1}{\varepsilon})$规模和$\tilde{O}$(对数$^2\frac{1}{\varepsilon}$)深度下进行列表译码，并可在多项式$(\frac{1}{\varepsilon})$.

虽然定理 1.1 需要充分发挥高维扩展图的全部能力，但我们还在第 3 节给出一个更简单的“预热”构造，仅使用子空间和我们的框架即可达到 $2^{-\sqrt{\log(N)}}$ 码率。即便如此，这也大幅改进了此前在 $\frac{1}{\text{多项式}(N)}$码率下的最佳结果。最后，我们指出，通过相当标准的级联分析，可以得到上述构造的一个强局部变体，其编码时间仅为多对数$(\frac{1}{\varepsilon})$（但码率会有所下降）。感兴趣的读者可参阅推论 A.2，或表 1 和表 2 中的参数摘要。

当 ε 固定且消息长度 N 趋于无穷时，定理 1.1 中的码的码率趋于零$\frac{1}{\mathrm{polylog}(N)}$。我们的第二个主要结果是在此区域内采用另一种基于高维扩张图的构造，实现真正的常数码率 和 $\mathrm{polylog}(N)$ 次查询：

定理 1.2 (常数率 aLLDC（推论 5.2）)。$\forall \varepsilon > 0$ 且存在无穷多个 $N \in \mathbb{N}$，使得存在一个二元近似局部可解码码 C_N，可从 $\frac{1}{2} - \varepsilon$ 个差错中解码，并满足：

1. 码率：$\exp(-O(1/\varepsilon^3))$
2. 列表大小：$\tilde{O}(\frac{1}{\varepsilon^2})$
3. 查询次数：$\log(N)^{\exp(O(1/\varepsilon^3))}$

此外，C_N 可编码和解码，所需规模为 $\log(N)^{\exp(O(1/\varepsilon^3))}$，深度为 $\mathrm{poly}(\log\log N, \exp(O(1/\varepsilon^3)))$.

结合低噪声区域内已知的优良局部码和并行码构造，定理 1.2 几乎立即推出具有次多项式时间列表译码的优良码，以及具有 RNC^1 列表译码的优良码（正式表述见下文第 1.3.1 节）。

为完整呈现这一结果，我们证明了一个新的下界：任何局部可计算的、码率为 R 的 aLLDC 都需要 $\Omega(\frac{\log(N)}{\varepsilon \log\frac{1}{R}})$次查询（见定理 7.1），这意味着我们的码与最优码之间仅相差一个多项式间隙（尽管在 ε 常数码率情形下，该多项式的增长并不理想）。[4] 虽然纯信息论设定并非我们工作的重点，但我们指出，在多对数码率、大字母表设置下，我们能够在低阶因子范围内达到该下界。若要同时实现相应的计算高效译码算法，仍是一个有趣的开放问题。

[2]注意此处的 $n = \log(N)$，因为我们对 f的整个真值表进行编码。[3]下文我们假设近似解码达到0.01的精度，大致意味着解码输出与“真实”有效消息的距离在0.01以内。很容易将此值放大到任意足够小的常数（甚至以码率和查询的成比例代价达到次常数，见命题 A.7）。[4]这仅在二元情形中是问题——对于更大字母表，我们构造的码进行多对数$(N) \cdot \exp(O(1/\varepsilon))$次查询。

1.3 复杂性与编码理论中的应用

下面我们概述定理 1.1（及其局部性约简变体推论 A.2）在困难性放大和 PRG 方面的主要应用（第 1.3.2 节），以及定理 1.2 在构造良好的局部列表可解码码和低深度码方面的应用（第 1.3.1 节）。我们先从后者开始，因为它们所需的解释较少。

1.3.1 局部与并行列表译码

尽管我们的码只能进行“近似”列表译码，但在低噪声区域，它们可以很容易地与优良码复合，从而实现真正的列表译码。特别地，由于 Katz-Trevisan [48]，的 log(N) 局部可解码码查询下界，这类 aLLDC 确实意味着：任何能够在 q 次查询和 T 时间内纠正固定比例差错的优良局部可解码码族，都可以放大为一个能够在 poly(q) 次查询和 poly(T) 时间内纠正至多 $\frac{1}{2}-\varepsilon$ 个差错的 LLDC 族。[5] 将此结果应用于 [38] 的次多项式时间低噪声局部可解码码，可将已知的最佳 LLDC 解码时间从 exp(log(N)$^{3/4}$) [40] 降低至 exp(log(N)$^{1/2}$)。[6]

推论 1.3 (次多项式时间内的 LLDC（定理 C.4）)。 对于每个 $\varepsilon>0$ 以及无穷多个 $N\in\mathbb{N}$，存在一个显式线性码 $\mathcal{C}$： $\{0,1\}^N\to\{0,1\}^{N'}$，该码能够从 $\frac{1}{2}-\varepsilon$ 个差错中进行局部列表译码，并且

1. 码率： exp($-O(1/\varepsilon^3)$)
2. 查询与时间复杂度： $\exp\left(O\left(\sqrt{\log N\cdot\log\log N}\right)\right)$
3. 列表大小： $\leq\tilde{O}(\frac{1}{\varepsilon^2})$

事实上，我们注意到这些码是局部可纠错的，这是一种更强的概念，可以纠正码字本身。转向并行设置，将我们的码与 Sipser 和 Spielman 的 [23]著名扩展码组合，便得到 RNC 中第一个良好的列表可解码码族[1]，，突破了 log^2(N) 深度障碍：

推论 1.4 (RNC 中的列表译码[1] (定理 C.1))。 对于每个 $\varepsilon>0$ 以及无穷多个 $N\in\mathbb{N}$，存在一个显式线性码 $\mathcal{C}$： $\{0,1\}^N\to\{0,1\}^{N'}$，它可以从 $\frac{1}{2}-\varepsilon$ 个差错中进行列表译码

1. ***Rate:*** $\exp(-O(1/\varepsilon^3))$
2. ***List Size:*** $\tilde{O}(\frac{1}{\varepsilon^2})$

此外， $\mathcal{C}$ 可以通过 $O(N\log(N)^{\exp(O(1/\varepsilon^3))})$ 规模的电路进行列表解码，且该电路属于 ***RNC***^{1} 。

根据构造中所使用的基码，这些码可以构造成低密度奇偶校验码，或者可以在 NC^1 中进行编码。关于这一结果及其模型的进一步讨论，请参阅附录 C。

1.3.2 困难性放大与快速伪随机生成器

对于复杂性类 $\mathcal{C}$，我们使用记号 $f\notin\text{Heur}_\delta\text{-}\mathcal{C}$ 表示 $\forall g\in\mathcal{C}$，dist(f,g) $>\delta$。下文中，按照惯例，[49, 50], 类 BPTIME($T(n)$)/$a(n)$ 表示由具有 $a(n)$ 比特辅助信息的概率图灵机在 Time$T(n)$ 内概率计算的函数类。

定理 1.1 的标准局部性约简变体[7] 对半均匀困难性放大有以下直接推论：

定理 1.5 （均匀的复杂度保持型困难性放大（定理 B.1））。 设 $T(n)$ 是一个时间复杂度， $1>\epsilon(n)\in 2^{-o(\sqrt{n})}$, 并且 A 是一个非负整数。设

$$f:\{0,1\}^n\to\{0,1\}\ \textit{such that}\ f\notin\text{Heur}_{\frac{1}{n^A}}\text{-BPTIME}(T(n))/O(\log(n/\epsilon))$$

[5]可能使用更大的字母表。如果基码族并非对每个消息长度 N 都有对应的码，此处需要稍加注意。我们在下文附录 C 中推论 1.3 的证明里处理了这一细节。[6]与 [38, 40], 主要致力于优化码率、而非解码时间相比，这种做法的代价是码率显著更低。我们指出，运用他们的方法并以优化解码时间为目标，很可能也能得到与推论 1.3 类似的保证。[7]更准确地说，定理 1.1 和这一变体（推论 A.2）都可以通过将同一个大字母表构造（定理 4.1）与适当的二元内码进行级联得到。我们在附录 A 中进行这一构造。

那么，存在长度为 $n' = n\,(1 + o\,(1))$ 的值，以及一个 $F\colon\ \{0,1\}^{n'} \to \{0,1\}$，使得

$$F \in P^f \ \ and\ F \notin \mathrm{Heur}_{\frac{1}{2}-\epsilon(n)}\text{-BPTIME}(T(n)\,\mathrm{poly}(\tfrac{\epsilon(n)}{n}))$$

注意，由于 $F \in P^f$，上述归约在多项式时间图灵归约下是封闭的，尤其允许在 NP $\cap$ Co-NP、P^{NP} 等进行放大（进一步讨论见 [37, 36] ）。定理 1.5 是第一个即使对于常数 ε，也不会使编码函数的定义域发生多项式膨胀的困难性放大结果。我们指出，要求 $\epsilon(n) \in 2^{-o(\sqrt{n})}$ 来自局部性归约步骤，并且存在于所有已知的保持复杂度的困难性放大结果中。消除这一限制将非常有意义，但这样做与改进作为 n之函数的定义域大小在很大程度上是正交的，而后者才是我们工作的重点。

利用同一构造，我们还得到如下针对标准非一致模型的保持复杂度的困难性放大:

定理 1.6（非一致保持复杂度的困难性放大（定理 B.2））。设 $S(n)$ 为电路规模函数，$1 > \epsilon(n) \in 2^{-o(\sqrt{n})}$，且 A 为非负整数。

$$f : \{0,1\}^n \to \{0,1\}\ \ such\ that\ f \notin \mathrm{Heur}_{\frac{1}{n^A}}\text{-Size}(S(n)).$$

那么存在一个长度为 $n' = n(1 + o(1))$ 的对象，以及一个 $F\colon\ \{0,1\}^{n'} \to \{0,1\}$，使得

$$F \in P^f \ \ and\ F \notin \mathrm{Heur}_{\frac{1}{2}-\epsilon(n)}\text{-Size}(S(n)\,\mathrm{poly}(\tfrac{\epsilon(n)}{n}))$$

同样，定理 1.6 是首个避免定义域的多项式膨胀 [13, 36] 或避免所得电路规模 [14, 51]相应损失的构造。

除了其本身的重要性之外，困难性放大最有力的应用之一是构造伪随机生成器 （PRG）。PRG 是密码学和复杂性理论中的核心工具，它能够将少量真正的随机性种子扩展为许多“伪随机”比特，使得任何高效的（非均匀）算法都无法将它们与均匀分布区分开来。若函数 $G\colon\ \{0,1\}^r \to \{0,1\}^m$ 满足：对于每个规模为 N的电路 C：则称其为一个（N，ε）-PRG。

$$|\Pr[C(G(U_r)) = 1] - \Pr[C(U_m) = 1]| < \varepsilon.$$

PRG 有若干个可以尝试优化的关键参数。本文聚焦于三个核心参数：种子长度 r、输出比特 m的数量，以及 G的运行时间 。

将定理 1.1（或者更准确地说，将其直接的、非保持复杂度的困难性放大转换）代入 Chen 和 Tell [51], 最近提出的框架，我们立即得到首个在合理的复杂度理论和密码学假设下，三个参数都近乎最优的 PRG 构造:

定理 1.7 (近最优快速 PRG（定理 B.7）)。存在一个 $\alpha_1 > 0$，使得对每个 $\alpha_0 < \alpha_1$，以下结论成立。假设

1. 存在能抵抗多项式规模电路的单向函数
2. $f \in TIME(2^n)$ 不属于 $i.o.\text{-}Size(2^{n(1-\alpha_0)})$

那么，对所有充分大的 N，都存在一个 $(N, 1/4)$-伪随机生成器

$$G : \{0,1\}^r \to \{0,1\}^N$$

其种子长度为 $r = (1 + O(\alpha_0)) \log N$ 比特，且可在 $N^{1+O(\alpha_0)}$ 时间内计算

取 $\alpha_0 \to 0$，我们得到输出比特数为 N、种子长度趋近于 $\log(N)$ 且运行时间为线性的 PRG。换言之，在这些假设下，我们证明任何时间为 N 的随机化算法都可以用大约 $\log(N)$ 个真随机比特进行模拟，且运行时间几乎不会增加；已知在 NSETH[52] 下，这已是最优的。先前在这一范围内构造的 PRG，要么种子长度更长，要么运行时间为二次。

表 1：我们二元 aLLDC 的信息论参数总结

结果	Rate	查询	列表大小
定理 1.1（推论 4.2）	$\mathrm{poly}\left(\frac{1}{\log N}, \varepsilon\right)$	$\mathrm{poly}(\log N, \frac{1}{\varepsilon})$	$\tilde{O}(\frac{1}{\varepsilon^2})$
推论 A.2	$\frac{1}{\text{多对数 } N} \cdot 2^{-O(\log^2(1/\varepsilon))}$	$\frac{\text{多对数}(N,\ 1/\varepsilon)}{\varepsilon^2}$	$\tilde{O}(\frac{1}{\varepsilon^2})$
定理 1.2 (推论 5.2)	$2^{-O(1/\varepsilon^3)}$	$\log(N)^{\exp(O(1/\varepsilon^3))}$	$\tilde{O}(\frac{1}{\varepsilon^2})$

表 2：我们二元 aLLDC 的计算参数汇总

结果	编码时间	解码规模	解码深度
定理 1.1（推论 4.2）	多项式$(\log N, \frac{1}{\varepsilon})$	多项式$(\frac{1}{\varepsilon}, \log N)$	$\tilde{O}\left(\log^2\left(\frac{\log N}{\varepsilon}\right)\right)$
推论 A.2	多对数$(N) \cdot \log^2 \frac{1}{\varepsilon}$	多项式$(\frac{1}{\varepsilon}, \log N)$	$O\left(\text{对数}\left(\frac{\text{对数 } N}{\varepsilon}\right) \text{对数 对数 } N\right)$
定理 1.2（推论 5.2）	$\log(N)^{\exp(O(1/\varepsilon^3))}$	$\log(N)^{\exp(O(1/\varepsilon^3))}$	$\exp(O(1/\varepsilon^3))$ 多项式 $\log\log(N)$

1.3.3 aLLDC 参数摘要

如前所述，我们在下方的表 1 和表 2 中简要总结了 aLLDC 构造的主要参数；这里不作 $\varepsilon \leq {}^{1}\log N$ 的假设，并纳入推论 A.2，即我们上述保持复杂度的困难性放大定理所用代码的局部性缩减版本。

1.4 证明概述

本节将概述定理 1.1 和定理 1.2 这两个主要结果的证明。关于困难性放大、伪随机生成器和编码理论的应用，本质上都是将这些码代入已知框架（再加上一些必要的细节处理）得出的。尽管我们的主要定理针对二元字母表表述，但在本概述中，我们将介绍大字母表上的码。这些码可以通过相当标准的级联技术转换为二元字母表上的码（见附录 A）。

1.4.1 亚常数误差下的 aLLDC（定理 1.1）

本文构造的所有码都基于经典的直积构造（在编码文献中，有时在 [41] 之后称为“ABNNR”）。设 (V, S) 是一个 r-均匀超图，其顶点为 $V = [N]$，集合（超边）为 S，大小为 r。基于 (V, S) 的直积码通过将长度为 N 的字符串视为函数 $f\colon V \to \{0,1\}$ 来编码，并针对每条超边 $s \in S$ 写下 f 在 s 各顶点上的取值。换言之，直积码将 f 编码为函数 $\mathsf{Enc}^f\colon S \to \{0,1\}^r$，方式如下：

$$\mathsf{Enc}^f(s) = f|_s,$$

为所有顶点 $v \in s$ 记录 $f(v)$。因此，该码将长度为 $|V|$ 的字符串映射为长度为 $|S|r$ 的字符串（更准确地说，映射为长度为 $|S|$、字母表为 r 位的字符串），因此其码率为 $\frac{|V|}{|S|r}$。

在直积码的语境中，若存在一个相对大小为 ε 的子集 $A \subseteq S$，使得对于每个 $s \in A$ 都有 $w_s = f|_s$，我们就称字 $w\colon S \to \{0,1\}^r$ 与码字 Enc^f 是 $(1-\varepsilon)$-接近的。我们的目标是在仅查询 w 中的 $\mathrm{polylog}(|V|)/\varepsilon$ 个符号的情况下，对这样的 f 进行局部解码。

固定 $|V| = N$。下文中，尽管我们的定理具有一般性，为求简便，我们建议将 $\varepsilon = \frac{1}{2^{\sqrt{\log N}}}$ 视为准多项式，并将 $r =$ 视为 $\mathrm{poly}(\log N, \frac{1}{\varepsilon})$。

归约到列表恢复我们在亚常数区间内构造 aLLDC，首先采用 Guruswami 和 Indyk [53, 43]提出的一种现已标准化的归约，将高噪声列表译码归约为低噪声列表恢复。接下来使用一种新的列表恢复译码器，这是亚常数误差区间中的主要创新步骤。

在列表恢复问题中，对于码 $s \in S$ 的每个索引，解码器都会收到一个由字母表符号组成的列表 $\mathcal{L}_s$（用函数表示法记作 $\mathcal{L}_s = \{g_1, g_2 \ldots, g_\ell\colon s \to \{0,1\}\}$）。若函数 $f\colon V \to \{0,1\}$ 与列表 是 $(1-\rho)$-接近的，

当 f 出现在列表的 ρ 比例中时，便称其与列表 $\{\mathcal{L}_s\}$ 是 $(1-\rho)$-接近的。与局部列表译码类似，我们在此问题中的目标是对所有与输入列表 $(1-\rho)$-接近的函数进行列表译码，同时尽可能少地查询列表中的值。Guruswami-Indyk 归约的核心思想是，将对 $(1-\varepsilon)$-接近函数 f 进行列表译码的问题（也就是说，这些函数只能由 S 中的 ε 比例计算得到），归约为一个列表恢复问题；在该问题中，几乎每个列表都包含 f，也就是对 (1)-接近函数 o 进行列表恢复。

列表恢复设置 我们的主要目标现在是在次常数噪声区间内执行列表恢复。更形式化地说，在局部列表恢复设置中，我们需要构造一个（概率）算法 $\mathcal{A}$，使其输出一个电路列表 $\{C_1, C_2, \ldots, C_\ell\}$。这些电路以顶点 $v \in V$为输入，查询访问 $\mathcal{L} = \{\mathcal{L}_s\}_{s\in S}$ 上的列表 S，并输出 $b \in \{0,1\}$。我们的目标是证明，对于任意一个与 $\mathcal{L}$ 的距离为 $(1-\rho)$ 的 f，都有一个输出电路 C_i，其计算的函数与 f 在 V 上的归一化汉明距离至多为 0.01。提醒读者，在这一设置中，应将 ρ 视为 $1-o(1)$，也就是说，f 出现在几乎所有列表中。

我们为此任务设计的生成算法极其简单：我们将抽样一个随机的 $s_0 \in S$，并为每个函数 $g_i \in \mathcal{L}_{s_0}$输出一个电路 $C_i^{s_0}$。下文将 $C_i^{s_0}$称为“第 i 个解码器”，或简称“解码器”。第 i 个解码器会默认 g_i 是某个全局 f 在 s_0 上的限制：$V \to \{0,1\}$ 与 $\mathcal{L}$ 的距离为 $(1-\rho)$，并尝试将这一信息传播开来，以便在所有 V 上计算 f。由于 $\rho = 1-o(1)$，任何真正与 $\mathcal{L}$ 的距离为 $(1-\rho)$ 的 f 几乎肯定会出现在 s_0 的列表中，因此，只要我们能证明上述类型的解码器能够成功，就大功告成了。

通过置信传播进行列表恢复 现在我们已经将问题化简为：已知 f 在 s_0 上的取值，并希望确定 f 在目标顶点 $v \in V$ 上的取值。我们分两步完成。首先，我们抽样一个随机的 $s_m \ni v$。根据之前相同的推理，我们几乎可以确定 $f \in \mathcal{L}_{s_m}$，因此我们真正要做的是利用函数在 s_0 上的取值，找出 $\mathcal{L}_{s_m}$ 中哪一个函数与 f（近似）一致。一旦找到了这样的 $g \in \mathcal{L}_{s_m}$，至少从概念上说，我们显然只需输出 $g(v)$ 作为解码结果，就大功告成了。

为此，我们希望利用以下基本的一致性启发式算法；它已出现在许多先前关于近似局部列表译码的工作中，例如 [13, 37, 36]：如果 s_0 与 s_m 恰好相交，且某个函数 $g \in \mathcal{L}_{s_m}$ 在它们的交集上与 f 近似一致（由于根据假设我们知道 f 在 s_0 上的取值，因此可以对此进行检查），那么我们有理由怀疑，$s_0 \cap s_m$ 上的这种局部一致性源于所有 s_m 之间的全局一致性。换言之，我们可以期望，如果 g 在 $s_m \cap s_0$ 上与 f 近似一致，那么它在所有 s_m 上都与 $f \approx g$ 一致。如果确实如此，那么我们就能以高概率得到 $g(v) = f(v)$，如所期望的那样。

粗略地说，要使这一启发式算法在形式上成立，我们必须解决两个核心问题。首先，这个启发式算法可能根本不成立。更准确地说，对于某个特定交集 $s_0 \cap s_m$，可能出现这种情况：$f = g|_{s_0\cap s_m}$，但对于所有 $v \in s_m \setminus s_0$，都有 $f \neq g$，此时我们的解码显然会出错。第二个问题是，尤其对于高码率码，交集 $s_0 \cap s_m$ 几乎肯定为空，因此我们无法根据 s_0 直接检查 $\mathcal{L}_{s_m}$ 中哪个函数与 f 接近。

为了解决第一个问题，我们要求集合系统具有强大的局部采样 性质。为了理解这一点，先设想交集 $s_0 \cap s_m$ 是从 s_m 中均匀随机选取的子集。那么根据切尔诺夫界，如果 f 在 s_m 上与 g 相距很远，那么以极高概率，f 在 $g|_{s_0\cap s_m}$ 上也相距很远（即在随机子集上）。局部采样将这一要求推广了。我们称 $s_m \in S$ 具有局部采样性质，如果对于任意一对函数 f，g：$s_m \to \{0,1\}$，f，g 在 s_m 上的汉明距离，与 f，g 在大多数非空交集 $s_0 \cap s_m$ 上的距离相近（就 s_0 的随机性而言）。这一要求在 [37, 36], 中就已很重要，因为它被用于构造码率为逆多项式的直积码。然而，高码率限制我们必须使用非常稀疏的集合系统（其中 $|S| \approx O(|V|)$），而构造具有这一性质的系统极具挑战性。值得庆幸的是，人们预期由高维扩展图产生的系统恰好会呈现这种行为。下面将进一步阐述。

剩下的问题是，s_0 和 s_m 可能并不相交。简单的度数计算表明，不可能同时获得良好的码率，又指望许多 v 都拥有任意 与 s_0 相交的超边 $s_m \ni v$（事实上，在适当的意义下，大多数 v 与 s_0 之间都相距“$\log(N)$远”）。为了解决这一问题，我们将研究这样的超图系统：其中总是存在长度至多为 $m \le \text{polylog}(N)$ 的超边短交集路径（s_0、s_1、.....、s_m），使得每个 s_j 都与 s_{j-1} 有显著的交集（见图1）。然后，我们可以沿着路径归纳地传播 f 在 s_0 上的取值：具体来说，对于每个 $j \in [m]$，通过在交集 $s_{j-1} \cap s_j$上将其与前一个集合中的近似函数 g_{j-1} 进行比较，找到一个与 f 接近的函数 $g_j \in \mathcal{L}_{s_j}$，即：

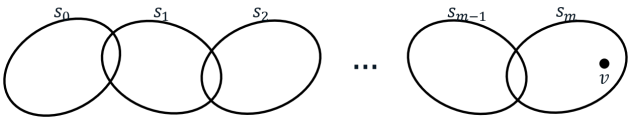

图 1：一条解码路径

1. 我们从 $f = g_0 \in \mathcal{L}_{s_0}$开始。

2. 给定 g_j： $s_j \to \{0,1\}$，我们找到某个 $g_{j+1} \in \mathcal{L}_{s_{j+1}}$，使得 $g_j \approx g_{j+1}$ 在交集 $s_j \cap s_{j+1}$ 上成立。[8]

这里，$g_j \approx g_{j+1}$ 仅表示两个函数之间的汉明距离低于某个阈值。根据三角不等式和基本的归纳法，如果 $f \approx g_j$ 和 $g_j \approx g_{j+1}$ 在 $s_j \cap s_{j+1}$ 上成立，那么 $f \approx g_{j+1}$ 在 $s_j \cap s_{j+1}$ 上也成立。最后，s_{j+1} 的局部采样性质意味着 $f \approx g_{j+1}$ 在所有 s_{j+1} 上成立（以高概率），从而继续沿链向下传播。

需要注意的是，即使 $g_0 = f$ 在所有 s_0 上成立，我们在算法后续步骤中仍只能期望得到近似保证 $f \approx g_j$（因为局部采样永远无法推出完美的全局一致性）。处理这一误差需要稍加斟酌，而上述简单策略实际上行不通（每轮都确定唯一的 g_j 会导致误差在每一步都翻倍[9]）。值得庆幸的是，我们可以通过在每个 s_j 维护一个函数子列表，其中的函数都接近 f，并在步骤 $j+1$ 记录 $\mathcal{L}_{s_{j+1}}$ 中所有接近先前子列表中任一函数的函数所构成的子集（最初仍只从 $g_0 = f$ 开始），来解决这个问题。更仔细的分析表明，子列表中的误差只会线性增长（见第 3.1.2 节）。因此，只要我们能将局部采样误差控制在$1_{\text{多对数}(N)}$以下，那么经过 $m \le$ 多对数(N) 步后，我们的最终解码几乎必然成功。

强显式低拥塞路由 虽然这一策略听起来很有吸引力，但仍存在一个隐藏的关键技术难点。为了高效解码，我们不仅需要任意一对超边之间存在 短路径 s_0，s_m，还需要能够在多对数时间内高效生成这些路径。此外，路径必须足够短（因为误差会随路径长度线性增长），并且必须在以下意义上具有“低拥塞”。即使有抽样保证，交集 $s_j \cap s_{j+1}$ 仍有一定概率无法准确测量 f 与 g_j之间的距离。除了该坏边集合规模较小（这是基于局部采样的假设）这一事实之外，我们无法控制这个坏边集合 s_j，s_{j+1}。因此，我们需要确保输出的路径以较低概率经过任何小规模的边集。

我们通过一种新的算法原语将此形式化，称为强显式（或“局部”）低拥塞路由（定义 3.3）。粗略地说，如果存在一个（随机化）算法 $\mathcal{O}$，对于任意输入顶点对 v，v'，都会输出一条路径$P = (v = v_1, \dots, v_m = v')$，其长度至多为 $m \le \text{polylog}(|V|)$，且运行时间为 $\text{polylog}(|V|)$，并满足对任意 $B \subset E$

$$\Pr_{v,v'}[\mathcal{O}(v,v') \text{ hits } B] \le \text{poly}(m, \Pr[B])$$

其中 v，v' 是来自 G 的独立同分布随机顶点。换句话说，由 $\mathcal{O}$ 输出的路径集合不会集中在任何小的边集上（因此得名“低拥塞”，文献中通常使用这一名称来指代该性质的各种变体）。[10]

因此，我们上述的要求对应于为图 G 构造一个局部路由方案，该图的顶点是超边 S，其边由具有非平凡交集的超边给出。[11]

关键性质总结 综上所述，要使该解码策略奏效，我们需要一个集合系统（V，S），并满足以下三个关键性质：

1. 利用交集上的接近程度推断与 f 的接近程度的局部采样性质。

[8]该算法以顺序方式给出，但可以使用 Savitch 算法 [54]以低深度执行。[9]原因在于，由于你可能无法选择距离 f最近的函数，因此在每一步中都必须将允许的误差阈值加倍，以确保下一步确实能够找到一个接近的函数。[10]实际上，我们给出了具有更强保证的局部路由方案 $\Pr_{v,\ v'}[\mathcal{O}(v,\ v')$ 命中 $B] \le m\Pr[B]$，但列表恢复并不需要这一点。[11]实际上，我们只会根据超图结构纳入这些边的一个自然子集。

2. 用于相交超边的多对数时间低拥塞路由方案（V， S ）

3. 对于高码率与良好局部性：

- $|S| \le N$ poly(log($N, \frac{1}{\varepsilon}$))，以及
- 每个 $s \in S$ 至多应有 poly(log $N, \frac{1}{\varepsilon}$) 个顶点。–

如何构造这样的集合系统?

高维扩展体前来救援 我们最终转向高维扩张器（高维扩展体）理论，它们是其低维同类的强大超图类比物，在近期进展与编码理论突破中发挥了重要作用，其中包括高码率（非局部）列表可解码码 [55], 局部可测试直积码 [56, 57, 29, 30], 乃至高效概率可检验证明 [31]。简而言之，高维扩展体是这样的超图：顶点和超边的邻域本身就是小型扩展图。

作为稀疏且良好连通的超图，高维扩展体在追求高码率局部码时是自然的选择。事实上，早在 [56] 的早期工作中就已观察到，我们所需的局部采样性质的变体在高维扩展体中自然出现。有趣的是，低拥塞路由的一种变体也出现在[30] 关于高效概率可检验证明的近期工作中，但设定相当不同（见第 1.5 节）。特别是， [30]的方案花费多项式时间并处理整个高维扩展体。这对我们的设定毫无用处，因为全部要点是在次线性（实际上甚至是多对数时间）内解码。我们的局部路由方案使用了完全不同的技术，可能对高维扩展体与算法领域具有独立意义。

我们从高维扩展体中为码导出的集合系统，与 [56] 及其他先前工作中的集合系统不同；在那些工作中，(V， S)分别由高维扩展体的顶点和超边构成。相反，在次常数误差情形下，我们依赖二维高维扩展图（即超边为三均匀的高维扩展图），而我们系统中的集合 S并非高维扩展体本身的超边，而是顶点邻域。因此，交集路径 s_0， s_1，……， s_m 实际上对应于高维扩展体底层图中的路径，而不是高维随机游走。这一看似细微的变化至关重要，因为先前工作中使用的自然超边编码无法支持列表恢复所需的[1] 多对数(N)强度局部采样。

最后，我们利用近期谱高维扩展图构造 [58] （对 [59]的“初等”构造的一种变体）所具有的强代数对称性，构造局部低拥塞路由方案。[12] 具体而言，我们将这些高维扩展体上的路由问题，化为在 $SL_d(\mathbb{F}_{q^\kappa})$上为群论字问题给出短解（即利用指定的一组简单生成元构造任意群元素 g ），并给出一个只需多对数(N)步的解法。在此过程中，我们还证明，这些高维扩展体及其相应的路由可以通过小型低深度电路实现。

1.4.2 常数码率的 aLLDC（定理 1.2）

我们的常数码率构造同样使用在适当的直积码上进行置信传播，但由此产生了一个核心问题：由于我们的列表恢复算法本质上需要 [1] 多对数(N)-强度采样，我们必须使用一个度为 polylog(N) 的高维扩展体来实例化我们的框架，这会导致编码码率损失 polylog(N)。[13] 我们能否在一个有界度 对象上进行 polylog(N) 步的信息传播，其中每个顶点仅位于 $O_\varepsilon(1)$ 个超边中?

通过局部码进行列表恢复 受 [56, 60] 的“双重采样器”方法启发（参见第 1.5 节），我们通过引入由超边构成的第三层 U （即“局部码”）来回答这个问题，使得我们能够将 (V, S) 上的列表译码（其度有界）归约为 (V, U) 上的列表恢复（其度可以达到多对数(N)，且不会对码率造成不利影响）。具体而言，我们将找到一个有界度系统 (V, S) 以及一组超边 U，使得 (V, S) 中每个 $u \in U$ 的“局部视图”，即交集 $\{u \cap s\}_{s \in S}$，都具有类似完全超图的结构。利用 [36], 中的思路，可以相对直接地对每个这样的局部码进行近似列表译码，从而模拟每个 $u \in U$ 对应的列表 $\mathcal{L}_u$ 。如果我们能确保 U 对 S 进行“全局采样”，也就是说，几乎所有内码 u 都能看到许多集合

[12]我们指出， [59] 的构造同样适用。我们采用 [58] ，因为它们的局部性质更易于描述。[13]另一个问题是，我们在上一节中使用的简单归约到列表恢复也会导致码率损失 polylog(N) 因子，但或许可以像 [39] 中那样，使用更复杂的基于 AEL 的归约来避开这一问题。由于在解决核心采样问题时，我们已经会在方案中使用一种“无损”的归约到列表恢复，因此没有必要采用这种方法。

s在计算 f 后，我们最终会得到一个定义在多对数(N) 度系统 (V, U) 上的次常数噪声列表恢复问题；我们可以合理地期待将原始传播算法应用于该问题。

即使暂且不考虑存在满足上述条件的三元系统（V、 S、 U），这种方法仍然存在一个问题：由于信息论障碍，我们无法保证列表 $\mathcal{L}_u$ 恰好包含全局函数 f，只能保证其中包含一个函数 f'，该函数与 f 的距离为 δ，其中 δ 与 $1/$码率存在多项式关系。由于我们关注的是保持常数码率，将这些列表代入之前的传播算法后，经过 m步传播， f 与最终选定函数 g_m 之间的归一化距离将达到 $m\delta \gg 1$，使结果完全失去实用价值，因为 $m \geq \Omega(\log(N))$。

为克服这一障碍，我们采用由 [60] 提出的列表处理过程的一个变体，以确保局部列表 $\mathcal{L}_u$ 中的函数彼此良好分离，同时仍包含一个与 f接近的函数。不幸的是，这个预处理步骤代价高昂，并可能导致内部误差 δ增大到 $\delta' = \delta\exp(O(\frac{1}{\varepsilon}))$.[14] 然而，完成初始预处理后，我们将拥有这样的性质：如果 g_k 与 g_{k+1} 的距离为 $2\delta'$，那么它与列表中的其他任何函数的距离都将至少为 $6\delta'$。因此，在传播算法中，我们可以将 g_k 唯一地匹配到下一个与 f 最接近的 g_{k+1}，且完全不会引入额外误差。论证与 [60], 中的论证类似，因此我们在此概述中略去。

容错强显式路由 即便如此，我们仍把一个最终的核心难题掩盖了起来。事实上，出于技术原因，我们无法用这种方式对系统中的所有集合进行预处理，只能处理其中一个较大的常数比例（比如 99%）。这就留下了 1% 的局部码；如果使用这些码，我们的传播仍会产生无法承受的误差。此外，如果只是从 u_0 路由到某个随机的 u_m，我们几乎肯定会遇到一个（实际上会遇到很多个） u_i，其预处理失败，从而破坏我们的传播。为了解决这个问题，我们大幅增强了路由方案，使其能够容忍顶点故障。具体而言，给定一个（足够大的）任意子集 $T \subset V$，顶点容错（“自适应”）路由方案必须输出路径u_0、 u_1、.....、 u_m，使得每个内部节点 $u_i \in T$，并保持低拥塞。这里， T 应理解为预处理成功的 99% 的节点，因此完全通过 T 进行路由，就能让我们执行上文所述的“无损”列表恢复。由于这与先前研究中另一种容错路由概念（例如 [31, 61]）的记号发生冲突，我们将在本文正文中把上述概念称为“子集内部路由”。我们的子集内部路由方案具有很强的通用性，能够在 [58] 的高维扩展体中任意足够大的顶点集或超边集内部高效路由（失败概率至多为某个次常数）。

关键性质总结 总之，要构造这些码，我们现在需要一个集合系统，它（粗略地说）具备之前的所有性质，并且还满足：

1. 一个额外的层 U，使得
   - 与 U 的交集（V, S）是（类似于）完全超图
   - U “样本” S，但发生概率为 $1/\mathrm{polylog}(N)$ 的情况除外（以确保 f 出现在大多数列表中）
2. 一种容错的多对数时间低拥塞路由方案。
3. 为确保常数码率和多对数(N)对数字母表大小：
   - 每个 $v \in V$ 至多位于 O_ε(1)个顶点中
   - 每个 s 包含至多 polylog(N) 个顶点

起初，我们并不清楚是否存在这样一个系统，使得 (V， S) 按照上述意义具有有界度。事实上，无法基于先前工作中使用的、元数不断增加的超边来构造这样的三层系统，因为超常数强度采样意味着超常数度（我们也无法直接使用上一节的顶点邻域编码，因为它也存在类似问题）。

我们通过使用维数 d （即极大集合规模）随 $\frac{1}{\varepsilon}$ 指数增长的高维扩展体来绕过这一障碍，并且有些反常地将消息比特编码在复形的 超边 上，而系统中的集合则对应于 顶点 （即每个顶点记录包含它的每条超边的比特值）。

[14]请注意，这种膨胀意味着我们无法在定理 1.1 的证明中使用这种预处理，因为其中 $\varepsilon \leq 1/\log N$，这样最终又会得到 poly($\delta'^{-1}$) $\leq 1/\mathrm{poly}(N)$ 码率，与先前构造类似。

这种编码确保码率按 $\frac{1}{d}$ 的量级变化，即一个常数。随后，我们将“第三层”集合 $u \in U$ 选取为高维扩展体中规模为多对数($|V|$)的重叠邻域，类似于定理 1.1 中使用的邻域编码，使得强采样直接由高维扩张蕴含，而局部“完备”结构则源于 [58] 的高维扩张器构造的强局部独立性。后一结构在第 6.2 节中讨论，也是使局部路由具备容错能力的关键。

一个有趣的问题是，我们的框架能否应用于 poly($1/\varepsilon$) 维高维扩展体，从而提高相应的码率。

1.5 相关工作

高维扩展体上的码、局部测试与列表译码过去十年间，许多工作致力于发展高维扩张器上纠错码的理论 [56, 62–66, 57, 67–71, 29, 30], ，主要聚焦于局部可测试性，[15] 最终推动了 c^3-LTC 猜想的解决 [28, 26] ，并构造出具有低可靠性的准线性 PCP [31]。这些工作，尤其是处于“低可靠性”测试范畴的工作，与我们的问题相关，因为它们的局部测试往往隐含地依赖列表译码。然而，尽管这些工作中的测试是局部的，相应的列表译码算法却并非如此。具体而言，典型策略是将问题归约为在整个编码上求解一个扩展型唯一游戏实例。由于我们尚不知道如何局部求解唯一游戏，即使是在扩展图上也不行，因此并没有明确的途径使这些算法实现局部化。

文献 [60], 对高维扩展体上的全局列表译码进行了更明确的探讨，其中作者分析了一种自然的直积构造：消息比特位于 k维高维扩展体的顶点上，并对超边 $X(i)$取直积，其中 $i \ll k$满足一定条件。与我们的超图三元组类似，作者的列表译码算法利用了这个“三层”系统（V， $X(i)$ $X(k)$），该系统构成了一种所谓的双重采样器（V， S， U），使得每个 $u \in U$的“局部视图”都是完备的。这也使他们能够在每个 $u \in U$上构造候选列表，然后利用 $X(k)$的全局扩张／重叠结构，通过求解相应的唯一游戏来进行列表译码。后续工作 [76, 77, 71] 利用半定规划和正则性引理大幅改进了这一方法，免去了对双重采样器的需求。

我们在常数码率区域中的超图三元系统，尤其是转向一种内置“局部码”的对象这一思路，显然受到了双重采样器方法的启发；不过，我们的系统需要满足更精细得多的性质（例如，由于需要在保证与 U 相关的各种其他局部和全局采样图具有次常数失败率的同时，使（V， S）保持常数度）。这使我们无法采用高维扩展体上的“自然”超边编码；即便使用我们其余的新工具，该方法似乎至多只能达到 exp($-$ poly log log(N) exp($O(1/\varepsilon)$)) 的码率，与解决本文中任何主要问题所需的参数相去甚远。

路由与概率可检验证明 如上所述，近期有研究 [31], 基于在某些特殊高维扩展体上，对超边直接积编码的局部可测试性（“一致性测试”）的研究进展 [62–64, 29, 30],，首次构造出了低可靠性、准线性规模的概率可检验证明。这也对应着某种规模高效的“困难性放大”，但这一设定中的挑战与我们要解决的截然不同。例如，我们要么处理 极度次常数 ε，要么要求编码具有常数度；而 [31] 处理的是常数 ε，并且可以接受在 N 中损失多对数因子，这使得问题有很大不同。更显然的是，[31] 需要局部测试，而我们需要局部译码。事实证明，这两者本质上截然不同：例如，局部测试实际上要求余边界扩张 [63, 64] （一种非常强的“拓扑”扩张概念）。我们不需要这一性质，但要求我们的复形满足其他各种性质，例如存在局部路由方案。

就此，我们指出 [31] 也使用了一种相关的路由概念，他们称之为“容错路由协议”。同样，这里的挑战相当正交，尽管这些构造共享了部分相同的高维扩展体工具。大致来说，容错路由的目标是构建一种协议，使每个顶点都能向其他每个顶点发送一条消息，即使有一定比例的边出现错误，且没有任何单个顶点进行过多的计算。唯一的其他约束是，该协议应能在 poly(N) 时间内完成计算。这与我们的设定相当不同，在我们的设定中，我们只计算一组

[15]此外，相关地，关于量子低密度奇偶校验码（另见 [72, 73, 26, 74, 75] 等）。[16]我们注意到，这本身是 [56], 中一个思想的变体，该思想使用相同的局部视图结构进行一致性测试。

路径，但我们必须在多对数(N)时间内完成。虽然我们确实需要一个能容忍顶点故障的版本，但我们的方案知道这些错误，这使得该问题与 [31]相当不同。

高码率局部列表译码 过去十年间，关于构造高码率局部（列表）可译码码取得了令人振奋的进展 [78–83]，最终在低噪声区域构造出了 [38] 的次多项式时间码，随后又在列表译码区间取得了进一步改进和扩展 [44, 40, 39] （进一步讨论见附录 C.2）。后续这些工作同样依赖组合距离放大，尤其是所谓的 AEL 构造 [42], ，该构造利用其基码（一种次常数距离重数码）具有列表可恢复性 这一事实，来证明最终码的局部列表可译码性。我们的方法消除了这一限制，并改进了查询复杂度和运行时间，但代价是相对于 ε而言码率显著降低。事实上，在大字母表设置下， [40] 实现了达到信息论容量的次多项式时间局部列表译码（即相对于纠错半径的最大码率）。如果我们的技术能够适用于这一更强的设置，将会非常有趣。

最后值得注意的是，已知某些扩展码的具体实例本身在低噪声区域中具有局部列表可纠错性质，所需时间为 N^{α} （即次线性但并非次多项式时间）。从某种意义上说，我们的结果利用高维扩展图，将其“提升”为在高噪声区域中的次多项式时间解码；不过，尽管二者存在一些相似之处（使用内码、路径），除了这些方面之外，很难看出两者的方法有多大关联。

伪随机生成器从各种平均情况和最坏情况的复杂度理论假设或密码学假设构造伪随机生成器，已有大量研究 [15, 84, 13, 85–89]。就在最近，两项令人振奋的工作 [90] （基于更强的非确定性困难性假设）和 [51]（假设存在单向函数）首次构造出了具有近乎最优（$1+\alpha$）log(N)种子长度的此类 PRG，并由此展示了如何在这些假设下对 BPP 进行去随机化，同时运行时间仅增加二次幅度。[17]

我们的伪随机生成器深受 [51] 的启发。事实上，其构造完全相同，只是将使用（拼接的）Reed-Muller 码替换为我们的 aLLDC 族。通过避免使用 Reed-Muller 码，我们无需为了计算 G 的哪怕一个输入，就先计算初始困难函数的整个真值表；正是这一点导致先前的构造需要二次时间。相反，我们的 aLLDC 只查询 f 的一小部分取值，因此可以在近线性时间内运行。我们注意到，虽然乍看之下，这意味着我们必须从平均情形困难性出发，但由于起始的最坏情形假设针对规模非常大的电路（与 [51] 中的假设相同），我们得以避免这一点；在这一范围内，足够强的平均情形困难性与最坏情形困难性基本等价。我们还注意到，这一改进在以下场景中最为重要：人们只想用 PRG 减少所需的真随机比特数，而非实现完全去随机化。后者需要遍历所有 $(1+\alpha)$ log(N) 个种子，因此总运行时间接近二次；这虽不可避免，但使用 Reed-Muller 码也能做到。

路线图

下面简要介绍本文主体部分的结构。在第 2 节中，我们介绍一般性的预备知识。在第 3 节中，我们引入传播解码和局部路由方案，并利用子空间给出一种简单的次多项式率构造。在第 4 节中，假设超图系统更为稀疏，我们实例化这一框架并证明定理 1.1。在第 5 节中，我们为常数码率码引入超图三元组框架，并在假设存在适当超图系统的条件下证明定理 1.2。在第 6 节中，我们利用高维扩展图，为这两种情形构造所需的超图系统和路由方案。在第 7 节中，我们给出 aLLDC 的查询下界，表明我们构造的码的变体接近信息论最优值。最后，在附录 B 和附录 C 中，我们分别给出 aLLDC 在困难性放大、PRG 和编码理论中的直接应用（并进一步讨论这些问题）。

[17]在略强的假设下， [51] 甚至能给出更精细的结果，其复杂度随 N 对输入规模的依赖关系而变化，并且可以优于二次复杂度。

2 预备知识

2.1 近似局部列表译码码

定义 2.1。 纠错码 $\mathcal{C}$： $\Sigma_1^N \to \Sigma_2^{N'}$ 是一个将消息 $f \in \Sigma_1^N$映射为码字 $w \in \Sigma_2^{N'}$ 的映射。

- 一个局部可计算码 $\mathcal{C}$配备了一个局部编码函数 Enc^f： $[N'] \times [\log |\Sigma_2|] \to \{0,1\}$，该函数可通过预言机访问消息 f： $[N] \to \Sigma_1$。给定输入 $s \in [N']$和索引 $i \in [\log |\Sigma_2|]$时，$Enc^f(s,i) \in \{0,1\}$输出 i 位，即与消息 f对应的码字 $\mathcal{C}(f)_s$的第 s^{th}个坐标的值。

除非另有说明，我们始终将输入字母表 Σ_1 视为规模恒定。两个字符串之间的距离 $dist(w, w')$是它们之间的归一化汉明距离，而 $\mathcal{C}$ 的距离 是任意两个码字之间的最小距离。

在本文中，我们将关注“高噪声”解码区域，在该区域中我们只能保证列表译码；此外，我们还将进一步放宽列表译码的要求，采用近似列表译码：

定义 2.2（近似局部列表可译码码（aLLDC））。 对于码 $\mathcal{C}$： $\Sigma_1^N \to \Sigma_2^{N'}$，令$\mathcal{B}_{1-\varepsilon}(w)$表示与 w的距离不超过 $1-\varepsilon$的码字集合。 $\mathcal{C}$是 δ-近似的、（ε， ℓ， q）-局部列表可解码的，如果存在一个概率图灵机 $\mathcal{A}$，它在输入 1^N时输出（概率）电路 C_1，...， C_ℓ，使得

$$\forall w \in \Sigma_2^{N'}, \forall f \in \mathcal{B}_{1-\varepsilon}(w) : \Pr_{\mathcal{A}}\left[\exists i : \Pr_{j\in[N],r}[C_i^{w,r}(j) = f(j)] \geq 1-\delta\right] \geq \frac{3}{4}.$$

此外，每个 C_i 最多在 q 个位置查询 w 。

我们指出，在正文中，读者经常会看到一个额外参数 δ_{in} 。这是一个强化的概念，它只要求字母表符号之间近似一致；我们仅针对直积码定义了这一概念。完整定义见定义 2.6。

请注意，为了简洁起见，我们也放宽了这样的要求：输出电路列表必须覆盖所有满足 Enc^f 与 w 的距离为 $(1-\epsilon)$-近的 f，改为只需以高概率覆盖每个固定的 f，使得 Enc^f与 w的距离为 $(1-\epsilon)$-近。此改动只是为了方便；在我们的构造中，只需将解码器运行足够多次以放大成功概率，使得我们可以对较小的组合列表大小应用并集界，就总能在此定义与通常覆盖所有函数的“更强”定义之间转换。由此我们还指出，有时需要对电路 C 和生成算法 A的内部随机性给出稍强的置信度保证，即

$$\forall w \in \Sigma_2^n, \forall f \in \mathcal{B}_{1-\varepsilon}(w) : \Pr_{\mathcal{A}}\left[\exists i : \Pr_r[C_i^{w,r} \text{ } \delta\text{-computes } f] \geq 1-\tau'\right] \geq 1-\tau.$$

换言之， C_i 在几乎所有随机性设置下计算 f，而不是对r取期望，并且以高（而非$_{3\,4}$）概率成立。只需两步即可转为这一概念：1）定义新的电路 C_i'，对 C_i的 O（$\log \frac{1}{\tau'}$）次独立运行的输出取多数表决（$\cdot$， r），标准的马尔可夫+切尔诺夫论证保证结果 O（δ）除概率为（$1-\tau'$）的情况外都能计算 f；2）重复运行该算法并输出这些电路 O（$\log \frac{1}{\tau}$）次，以确保其中一个确实能以高概率计算 f。

编码与解码局部性。 如引言中所讨论的，这项工作的关键在于编码和解码的电路复杂度，既涉及电路规模和深度，也涉及编码和解码局部性。我们用 q 表示解码局部性（解码所需的查询次数），用 r 表示编码局部性（每个码坐标所依赖的消息坐标数量）。我们指出，在大多数情况下，仅写出编码时间会更简单。除局部性 r之外，这还包括计算应查询哪 r 个消息比特所需的时间（然后据此计算相应的输出）。必要时，我们会同时记录编码时间和编码局部性，但在可能的情况下，为简便起见只记录前者，因为它给出后者的上界。

2.2 直积与电路复杂度

几乎所有近似局部列表可译码码的构造都是通过所谓的直积 构造建立的。直积码基于底层超图 $H=(V,S)$，其中 $V=[N]$是顶点集， $S\subset\binom{[N]}{r}$ 是一个（r-均匀）集合系统（超边族）。若每个顶点 v 在 S中出现于相同数量的超边中，则该超图是正则的。本工作中用于编码的所有超图都将是正则的。

与 (V,S) 对应的直积码记作 $\mathcal{C}_{(V,S)}$，它将消息 $f:V\to\Sigma_1$ 编码为码字 $\mathsf{Enc}^f:S\to(\Sigma_1)^r$，方法是沿每条超边拼接符号，即：

$$\mathsf{Enc}^f((v_1,\ldots,v_r))=(f(v_1),\ldots,f(v_r))$$

其中，我们为超边 $s=\{v_1,\ldots,v_r\}$ 固定了某个任意顺序。等价地，可以将 Enc^f 看作一族赋值 $\{\mathsf{Enc}^f_s:s\to\Sigma_1\}_{s\in S}$，其中 $\mathsf{Enc}^f_s(v_i)=f(v_i)$。

在本文中，我们考虑配备辅助解码图 G的直积码，该解码图通常由超边上的某种自然交集结构给出。我们将超图及其对应解码图构成的集合称为“超图系统”：

定义 2.3 (超图系统)。一个译码系统(V,S,G) 由一个超图(V,S)以及定义在超边 $G=$ 上的一个“解码图”(S,E)组成

超图系统还为每个 $s\in S$ 自然地配备了一族交图，它们将对我们的解码过程至关重要。给定 $(V、S、G)$ 和 $s\in S$，交图 $G_s=(L、R、E)$ 是一个二部图，其中 $R=s$，而 L 由 s 与其邻居$s'\in G$之间所有可能的交集集合索引，即

$$L\coloneqq\{s\cap s':(s,s')\in E\}$$

G_s 是该系统的包含图。更正式地说，我们通过以下概率实验生成该图的边：

1. 抽样一个邻居 $s'\sim\pi_E(s)$，并记为 $t=s\cap s'\in L$
2. 均匀随机地抽样 $v\in t$，并输出边 $\{v,t\}$。

如果所得的关于 $R=s$ 的边缘分布是均匀的，我们就称 G_s 是“均匀”的。如果 $(V、S)$ 和 G 是正则的，并且所有 G_s 都是均匀的，则称超图系统正则。本文所考虑的所有超图系统都是正则的。

由于 上述定义有些抽象，为了具体起见，我们来考虑一个基本例子 s.

例 2.4 （完全复形）. 设 $V=[N]$ 与 $S=\binom{[N]}{r}$ 为完全超图。解码图的一个“典型”选择是将恰好在一半顶点上相交的超边连接起来（所得图通常称为“约翰逊图”）。给定固定的 $s\in\binom{[N]}{r}$，交图 G_s就是带有 $L=[s]$、$R=\binom{[s]}{r/2}$ 的“包含图”（即 R 中的每个 $r/2$-集都与它所包含的 L 中的每个顶点连通）。这是因为我们通过随机选取 s 的一半作为相交部分，再随机选取 $r/2$ 个新顶点纳入 s'，从而从 s 移动到解码图中的一个相邻 s'，因此与随机邻居的交集恰好是来自 s的 $r/2$ 个随机顶点。

(, S) 的电路实现。基于超图系统的码的复杂度，会随着构造超图和解码图本身的复杂度而变化。具体来说，我们通常需要超图的邻接算子的电路实现（更准确地说，是 (V,S) 的二分包含图的邻接电路）。更明确地说，我们需要以下电路

1. $C_{V\to S}(v,i)$：输出 S 中第 i 个邻居
2. $C_{S\to V}(s,i)$：输出 V 中第 i 个邻居

针对邻居的某种特定实例的排序。这里请注意， v 和 s 分别由 $\{0,1\}^{\log V}$和 $\{0,1\}^{\log S}$中的相应位表示给出。

不那么显而易见的是，我们还需要这样的电路：给定系统中的相邻项（v、s），输出使得 邻居 v 相对于 s 而言是 s，反之亦然。我们称之为“反向索引”电路：

1. $C_{V\to S}^{-1}(v,\ s)$：输出 i，使得 $C_{V\to S}(v,\ i) = s$
2. $C_{S\to V}^{-1}(s,\ v)$：输出 i，使得 $C_{S\to V}(s,\ i) = v$

最后，我们的解码算法还需要一个电路，它接收两条超边 $s,\ s'$ 并计算出它们交集中的一个元素。

1. $C_{S\cap S\to V}(s,\ s',\ i)$：输出第 i 个元素 $v \in s \cap s'$；若 $s \cap s' = \emptyset$，则输出'失败'

编码和解码复杂度：最后，基于以上内容，我们简要地在电路模型中形式化定义直积码的编码和解码。对于 $C_{(V,S)}$，编码电路是一个电路 $C_{Enc}(s,\ i)$，它以 $s \in S$ 的比特名和索引 i 为输入，并在给定消息 f：$V \to \{0,1\}$ 的预言机访问权限后，计算 $f(C_{S\to V}(s,\ i))$，也就是说，计算 f 在 s 的第 i 个邻居上的值。

类似地，前一节讨论的码的解码电路可以预言机的形式访问码字 w（通常将其视为一个电路 C_w）；该电路接收比特表示 $(s,\ i)$，并输出 w 对 s 的第 i 个邻居所声称的值。注意，这使我们能够构造规模仅随码的字母表大小（即索引长度）对数增长的解码电路。换言之，这些解码电路可以选择不读取编码在 $s \in S$ 处的整个字母表符号，而是查询该符号的特定位。除非另有说明，解码电路通常是随机化的。

2.3 列表恢复

我们的列表解码器关键依赖于一项称为列表恢复的中间解码任务。一个列表可恢复码 $\mathcal{C}\colon \Sigma_1^N \to \Sigma_2^{N'}$，其解码器的输入是一个列表，其中对每个 $i \in [N']$，Σ_2 中都有一个包含 ℓ_{in} 个符号的列表，记作 $\mathcal{L} = \{\mathcal{L}_s\}_{s\in[N']}$。注意，如果 $\ell_{in} = 1$，那么 $\mathcal{L}$ 就只是一个码字 $w \in \Sigma_2^{N'}$，与列表译码的情形相同。不过，与列表译码的情形不同，我们主要关注低噪声区域下的列表恢复；在这种情况下，我们认为希望解码的全局函数 f 存在于解码器的几乎每个列表输入中。

现在我们将这一设定形式化。一个 ℓ_{in}-输入列表可恢复码 $\mathcal{C}\colon \Sigma_1^N \to \Sigma_2^{N'}$ 可以预言机访问一个输入列表 $\mathcal{L} = \{\mathcal{L}_s\}_{s\in[N']}$，其中每个 $\mathcal{L}_s \subset \Sigma_2$ 都是一个符号列表，规模至多为 ℓ_{in}。如果除去比例为 η 的列表外，其余列表都包含 $\mathrm{Enc}^f(s)$，我们就称消息 $f \in \Sigma_1^N$ 由 $\mathcal{L}$ 以 $(1-\eta)$-方式计算。我们将这类 f 的集合记为 η-球：

$$\mathcal{B}_\eta(\mathcal{L}) \coloneqq \{f : \mathcal{L}\ (1-\eta)\text{-computes } \mathcal{C}(f)\}$$

如果一种码能够恢复此球内的任意 f，则称其为列表可恢复码：

定义 2.5（近似局部列表可恢复码（aLLRCs））。若存在概率图灵机 $\mathcal{A}$，使其在输入 1^N 时输出（概率）电路 C_1、.....、$C_{\ell_{out}}$，则码 $\mathcal{C}\colon \Sigma_1^N \to \Sigma_2^{N'}$ 称为 δ_{out}-近似、ℓ_{in}-输入（$1-\eta, \ell_{out}, q$）-局部列表可恢复码，当且仅当

$$\forall \mathcal{L} \in (\Sigma_2^{\ell_{in}})^{N'}, \forall f \in \mathcal{B}_\eta(w) : \Pr_{\mathcal{A}}\left[\exists i \in [\ell_{out}] : \Pr_{j\in[N],r}[C_i^{\mathcal{L},r}(j) = f(j)] \geq 1-\delta_{out}\right] \geq \frac{3}{4}.$$

此外，每个 C_i 至多在 q 个位置进行 w 次查询。

我们这样表述上述内容，是为了强调低噪声区域（可以认为 η 很小，例如 1%，甚至是 $o_N(1)$）；不过值得注意的是，当 $\eta = 1-\varepsilon$ 很大（即"高噪声"区域）时，这一定义直接推广了（$\varepsilon, \ell_{out}, q$）-列表译码，后者正是 $\ell_{in} = 1$ 的情形。虽然我们主要关注低噪声区域，但在转向二元码和低局部性情形的级联分析中，也会使用高噪声列表恢复；此时我们写作（$\varepsilon, \ell_{out}, q$）-LLRC，以强调我们处于高噪声情形。

2.4 码级联与双重近似差分隐私码 s

码级联是编码理论中的一种简单范式：将"外"码中的大符号替换为短的"内码"。在我们的构造中，我们将在多个地方关键性地使用码级联，主要是为了从大字母表转向放大困难性和构造 PRG 所需的二元码。

更形式化地说，给定一个“外码” $\mathcal{C}_{\text{out}}$:$\Sigma_{\text{out}}^{k} \to \Sigma^{n_1}$外, 2 及其编码映射 Enc_{out} ，以及一个“内码”$\mathcal{C}_{\text{in}}$: $\{0,1\}^{\log|\Sigma_{\text{out},\ 2}|} \to \sum^{n_2}$内（为简单起见，我们假设输入是二元的，但并非一定如此）及其编码映射 Enc_{in}，级联码 $\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}} : \Sigma_{\text{out}}^{k} \to \Sigma^{n_1 n_2}$内的定义是：用内码 $\mathcal{C}_{\text{in}}$ 的较小字母表对外码 $\mathcal{C}_{\text{out}}$ 的每个符号进行编码。也就是说，对于 $(i,\ j) \in [n_1] \times [n_2]$ 和一条消息 $f \in \Sigma_{\text{out}}^{k}$:

$$\text{Enc}_{\mathcal{C}_{\text{out}} \circ \mathcal{C}_{\text{in}}}(f)_{i,j} = Enc_{\text{in}}(Enc_{\text{out}}(f)_i)_j$$

正如我们将在全文中讨论的那样，码级联通常能保留码的大多数优良性质（至多会有一些小损失），例如局部性和查询复杂度。

2.4.1 双重近似 LLRC

在近似解码中，级联存在一个小问题。级联中的内码只能近似解码时，外码通常接收到的符号只是接近 正确，而非完全正确。这是文献中的一个常见问题 [36], ，也引出了我们最后的核心定义：该定义确保即使有 ε 比例的列表所对应的函数只能近似地 计算出编码后的消息，LLRC/LLDC 仍能解码。为简便起见，我们只针对直积码使用此定义。给定一个 DP 码 $\mathcal{C}_{(V,S)} : \Sigma_1^N \to (\Sigma_1^r)^{N'}$，若存在比例为$(1-\eta)$的块 $s \in [N']$，使得某个函数$g \in \mathcal{L}_s$ 在 δ_{in}意义下与 f 相差不超过 δ_{in}，即 $\Pr_{v\sim s}[g(v) \neq f(v)] \leq \delta_{in}$，也就是说它们除至多 δ_{in} 外均一致，则称输入列表 $\mathcal{L}(1-\eta,\ \delta_{in})$计算出消息 f。我们将这类 f 构成的集合定义为以 $\mathcal{L}$为中心的（近似）$(\eta,\ \delta_{in})$-球:

$$\mathcal{B}_{\eta,\delta_{in}}(\mathcal{L}) \coloneqq \{f \in \Sigma_{in}^N : \Pr_{s\in\Sigma_{out}^{N'}}[\exists g \in \mathcal{L}_s : g \text{ is } \delta_{in}\text{-close to } f|_s] \geq 1-\eta\}$$

稍微滥用一下记号，我们也将能够恢复此类内容的直积码简称为“ f 近似局部列表可解码（恢复）码”，但会包含额外参数 δ_{in}。

定义 2.6 （（双重）近似局部列表可恢复码）。码 $\mathcal{C}$: $\Sigma_1^N \to \Sigma_2^{N'}$ 是一个$(\delta_{in}, \delta_{out})$-近似、 ℓ_{in}-输入、$(1-\eta, \ell_{out}, q)$-*LLRC*，当且仅当存在一个概率图灵机 $\mathcal{A}$，它在输入1^N 后输出（概率）电路 $C_1, \ldots, C_{\ell_{out}}$，使得

$$\forall \mathcal{L} \in (\Sigma_2^{\ell_{in}})^{N'}, \forall f \in \mathcal{B}_{\eta,\delta_{in}}(\mathcal{L}) : \Pr_{\mathcal{A}}\left[\exists i \in [\ell_{out}] : \Pr_{j\in[N],r}[{C_i}^{w,r}(j) = f(j)] \geq 1-\delta_{out}\right] \geq \frac{3}{4}.$$

此外，每个 C_i 查询在至多 q 个位置上进行 $\mathcal{L}$ 。

当 $\ell_{in} = 1$ 且 $\eta = (1-\varepsilon)$ 时（即处于高噪声区域），上述定义就特化为（双重）近似局部列表译码。为方便起见，我们明确给出定义。

定义 2.7 （（双重）近似局部列表可恢复码）。码 $\mathcal{C}$: $\Sigma_1^N \to \Sigma_2^{N'}$ 是一个$(\delta_{in}, \delta_{out})$-近似、$(\varepsilon, \ell_{out}, q)$-*LLDC*，当且仅当存在一个概率图灵机 $\mathcal{A}$，它在输入 1^N 后输出（概率）电路 $C_1, \ldots, C_{\ell_{out}}$，使得

$$\forall w \in \Sigma_2^{N'}, \forall f \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L}) : \Pr_{\mathcal{A}}\left[\exists i \in [\ell_{out}] : \Pr_{j\in[N],r}[{C_i}^{w,r}(j) = f(j)] \geq 1-\delta_{out}\right] \geq \frac{3}{4}.$$

此外，每次 C_i 查询 w 至多涉及 q 个位置。

当 $\delta_{in} = 0$时，我们便得到上文定义的近似解码的标准概念。下文中，我们将这些统称为近似局部列表可恢复码或局部列表可译码码，并仅通过是否包含参数 δ_{in}来区分这一概念与原始定义。

2.5 图与采样器

设 $G = (L, R, E)$ 为一个二部双正则图。我们用 表示 $A \subseteq R$ 的概率

$$\mathop{\mathbb{P}}_{v\in R}[A] \coloneqq \frac{|A|}{|R|}.$$

（分别为 $A \subseteq L$ 将 R 替换为 L，反之亦然）。为简洁起见，有时我们省略 $v \in R$。对于 $A \in R$ 和 $v \in L$, 我们用 表示

$$\mathop{\mathbb{P}}_{u \sim v}[A] := \frac{|A \cap N(v)|}{D},$$

其中 $N(v)$ 是与 v 相邻的顶点集合，而 $D = |N(v)|$ 是左度。

以下图性质对我们的分析至关重要。

定义 2.8 (采样器图)。设 $\beta \geq 0$，且 $G = (L, R, E)$ 是一个二部图。若对每个 $A \subseteq R$ 和每个 $\delta \in [0,1]$, ..., 则称 G 是一个 β-采样器，

$$\mathop{\mathbb{P}}_{v \in L}\left[\left|\mathop{\mathbb{P}}_{u \sim v}[u \in A] - \mathop{\mathbb{P}}_{u \in R}[A]\right| \geq \delta\right] < \frac{\beta}{\delta^2}.$$

以下标准引理指出，任何二部扩展图都会自动成为一个良好的采样器（例如参见 [56]）。设 $\lambda \geq 0$。回顾一下，若一个二部图的归一化邻接算子的第二大特征值为 $\leq \lambda$，则称其为 λ-谱扩展图。[18]

断言 2.9 ([56])。设 $G = (L, R, E)$ 是一个 λ-谱扩展二部图。那么 G 是一个 λ^2-采样器。

2.6 复杂度类与平均情形复杂度

在本小节中，我们介绍有关复杂性类和平均情形复杂度的记号，以便陈述我们的工作对困难性放大和伪随机性的影响。

我们假设读者熟悉以下基本复杂性类:

$$P, NP, BPP, PSPACE, E = TIME(2^{O(n)})$$

。我们用 Size(S, n) 表示可由至多包含 S 个门的电路计算的、作用于 n 比特输入的布尔函数集合（若上下文清楚，也简写为 Size(S)）；用 SizeDepth(n, S, d) 表示可由深度为 d 且至多包含 S 个门的电路计算的布尔函数集合。对于复杂性类 C 和布尔函数 f，我们用 C^f 表示由定义 C 的同类算法计算、但可访问 f 预言机的函数类。

为简单起见，我们只考虑相对于 n 位串上的均匀分布的平均情形复杂度，但大多数结果很容易推广到其他分布。对于类 C，我们用 Heur$_{\delta(n)}[C]$ 表示函数类 f：其中存在一个函数 $g \in C$，使得对于所有 n, $Prob_{x \in \{0,1\}^n}[f(x) \neq g(x)] \leq \delta(n)$；换言之，存在一个属于 C 的启发式算法，能够解决除占 $\delta(n)$ 比例以外的所有实例。如果 C 是由随机化算法定义的类，我们就对输入 x 和算法所用的随机性同时取概率。 $i.o[C]$ 表示这样的函数类：其中的函数在无限多个输入规模上与 C 中某个函数相等。如果 $f \notin i.o[\text{Heur}_{\delta(n)}[C]]$，我们就称 f 对 C 是 $\delta(n)$ 困难的 。

布尔电路是一种非一致计算模型，在这种模型中，每种输入规模都可以对应一个单独且独立的算法。诸如 P 这样的复杂性类则要求所有输入规模都使用同一个算法。遵循 [50, 49], ，我们将 $C/a(n)$定义为函数 f 的“半一致”类：存在一个$g \in C$，使得对于所有 n，都存在一个 z、 $|z| = a(n)$，满足对于所有 x，只要 $|x| = n$，就有 $f(x) = g(x, z)$。如果 C是由随机化算法确定的复杂性类，我们允许 z 成为算法所用随机硬币的函数，因此存在一个使用随机性 r 的相应类型的 g，以及函数 $z(r)$ $|z(r)| = a(n)$，使得对于所有 x，在 r 上以高概率有 $f(x) = g(x, r, z(r))$。对于大多数类 C，我们可以找到相应类型的电路 $Circ$ 以及生成电路的算法，使得 $f \in C/a(n)$当且仅当存在一个随机化算法：给定 n，以概率 $2^{-a(n)}$ 返回一个计算 f的电路 $Circ$ 。

困难性放大旨在表明，有一定难度的问题的存在蕴含着更可靠地困难的问题的存在。一个 $\delta(n)$ $\delta'(n)$ C_1, C_2, C_3 困难性放大构造是一个将 f 映射到 $F \in C_1^f$ 的泛函，使得若 $f \notin$Heur$_{\delta(n)}[C_2]$ 则 $F \notin$ Heur$_{\delta'(n)}[C_3]$。若 C_1 相对于 C_2 和 C_3是一个“小”类，若 C_3 并不比 C_2小很多，并且若 $\delta'(n)$远大于 $\delta(n)$，则这一点是有意义的。

[18]邻接算子是一个 $V \times V$实值矩阵：当且仅当 $\{v, u\}$ 是一条边时， $A(v, u) = \frac{1}{deg(v)}$，否则为 0。谱扩张是一个标准概念，详见 [91] 中的广泛综述。

3 次多项式率 aLLDC

在本节中，我们给出如下具有次多项式码率的 aLLDC 的“热身”构造，它引入了我们框架的核心部分，同时避免了大部分技术困难。

定理 3.1 (次多项式率 aLLDC)。 对于所有足够大的 $N \in \mathbb{N}$ 和 $\varepsilon \leq 2^{-\log(N)}$，存在$N' \geq N$以及一个强显式直积码 $\mathcal{C}: \Sigma^N \to \Sigma^{N'}$，它是一个 $(10^{-6}, \log N, 0.01)$-近似$(\varepsilon, \ell_{out}, q)$-局部列表可解码码，且[19]

1. 码率： $R \geq 2^{-16\sqrt{\log N \log \frac{1}{\varepsilon}}}$
2. 查询复杂度与列表大小： $q,\ \ell_{out} \leq \tilde{O}(\frac{1}{\varepsilon})$

特别地，当 $\varepsilon > N^{-o(1)}$ 时，码率为次多项式 。此外， $\mathcal{C}$ 可局部可编码，时间为$\mathrm{Time}(\mathrm{polylog}(N) \cdot \log(1/\varepsilon))$，且可局部解码，规模为$\mathrm{poly}(\frac{1}{\varepsilon})$，深度为 $O\left(\log \frac{1}{\varepsilon} \cdot \log^2(\log(N))\right)$。

定理 3.1 的表述旨在突出 $\varepsilon \ll 1/\log(N)^{\omega(1)}$ 区间（也就是说，我们省略了 $\log(N)$ 因子，并且只关注次常数 ε）。这对应于将一个函数 $f: \{0,1\}^n \to \{0,1\}$ 放大为 $\mathrm{Enc}^f: \{0,1\}^n \to \Sigma$。前者在超过 99% 的输入上无法由规模为 $\mathrm{Size}(S)$ 的电路计算；后者则难以在多项式时间内计算。这是困难性放大中更有意义的区间。

然而，从编码理论的角度来看，当 ε 为常数（或随 N 衰减得更慢）时，定理 3.1 背后的构造也很有趣。事实上，在这种情况下，该码实现了近乎渐近最优的码率/查询/列表大小权衡，其码率为 $2^{-\tilde{O}(\sqrt{\log(N)})}$，列表大小为 $O(\frac{1}{\varepsilon})$，查询复杂度为 $\tilde{O}\left(\frac{\sqrt{\log(N)}}{\varepsilon}\right)$。根据我们在第 7 节针对（局部可编码）码给出的一般下界，这一结果是紧的。

虽然定理 3.1 实现了次多项式码率，但其字母表和局部性都偏大，这对伪随机生成器与一致性困难度放大的应用而言是个问题。这些问题中的第一个很容易解决，只需与一个足够好的多项式码率二元内码进行级联。具体而言，将定理 3.1 与 [36] 的多项式码率近似局部列表可恢复码级联，以将对数字母表大小降至 $\mathrm{poly}(\log N \cdot \frac{1}{\varepsilon})$，再与任意足够好的二元内码级联（我们使用级联里德-所罗门码[92],，详见附录 A），即可得到如下推论：[21]

推论 3.2 （二元次多项式码率 aLLDC）。 对于所有足够大的 $N \in \mathbb{N}$ 和 $\varepsilon \leq 2^{-\log(N)}$，存在$N' > N$ 以及一个强显式二元线性码 $\mathcal{C}: \{0,1\}^N \to \{0,1\}^{N'}$，它是一个 0.01-近似$(\frac{1}{2} + \varepsilon, \ell_{out}, q)$-*aLLDC*，且具有

1. 码率： $r \leq 2^{-O(\sqrt{\log N \log \frac{1}{\varepsilon}})}$
2. 查询复杂度和列表大小： $q, \ell_{out} \leq \mathrm{poly}(\frac{1}{\varepsilon})$
3. 局部性：$\mathrm{poly}(\frac{1}{\varepsilon})$

此外， $\mathcal{C}$可以在$\mathrm{Time}(\mathrm{poly}(\frac{1}{\varepsilon}))$ 内进行局部编码，并以对数空间一致规模$\mathrm{poly}(\frac{1}{\varepsilon}, \log N)$和深度 $\tilde{O}\left(\log^2\left(\frac{\log(N)}{\varepsilon}\right)\right)$进行局部解码。

在某些应用中，例如在 NP $\cap$ 与 Co-NP [36]等复杂性类中进行困难性放大， [19],或应用于学习时，$\frac{1}{\varepsilon}$ $\mathrm{poly}(\frac{1}{\varepsilon})$ 编码时间并不足够，尤其是当$\varepsilon \ll 1/\mathrm{polylog}(N)$时，因为由此得到的编码函数过于复杂（例如，需要在超多项式多个位置进行计算）。这就是为什么我们需要上述码允许一定程度的 δ_{in} 符号错误。具备这一性质后，通过级联将局部性提升到仅为 $\mathrm{polylog}(N) \cdot \log^2(\frac{1}{\varepsilon})$ 并不太困难（尽管码率会更差）。由于在我们的参数范围内，这一点并不能直接由文献中的结果以黑盒方式推出，因此我们在附录 A 中仔细陈述并证明了这些码的存在性，以构造码率得到改进的 $\mathrm{polylog}(N)$ 码率构造。

[19]下文中， $\tilde{O}$ 表示隐藏多对数 N 因子。[20]我们指出，对于足够小的 δ_{in}，也可以处理一般的 δ_{out}，代价是码率和查询复杂度至多损失一个 $\mathrm{poly}(\delta_{out})$ 因子。[21]请注意，在下文的查询复杂度中，定理 3.1 只读取所查询的每个符号中的 $\mathrm{poly}(\log N, \frac{1}{\varepsilon})$ 比特。我们的算法几乎可以“免费”实现这一点。进一步讨论请参见注 3.6。

构造概述： 在继续之前，我们先简要概述这一构造。从高层次来看，我们使用第 2 节介绍的超图系统框架。这里简要回顾一下。一个超图系统由顶点集 V、超边 S，以及用于建议传播的、定义在 S 上的重叠超边解码图 G 组成。系统 (V, S, G) 会自动诱导出交图 G_s，它们对应于在 G 中， s 的哪些顶点会与来自 s 的随机邻居 s' 取交时被子抽样。

定理 3.1 的构造分为三个部分，每个部分在下文都有对应的小节。第一部分完全是通用的，它是Guruswami和Indyk [43, 53] 如今标准技巧的一个变体，该技巧将高噪声列表译码问题归约为低噪声列表恢复，即如下设定：每条超边 $s \in S$ 看到一个较小的列表 符号，并且我们保证存在一个全局函数 f：$V \to \{0,1\}$，其中几乎所有 超边的列表中都包含 $f|_s$。我们在 aLLDC 设定下给出他们这一归约的一个版本，从而将问题归约为在低噪声区域中构造良好的近似局部列表可恢复码。

构造的第二部分，即主要 部分，则是超图系统 (V, S, G) 上局部列表恢复的一个新框架及其后续实例化。具体而言，我们为满足以下两个关键性质的任意系统构造一个简单的传播式译码器：

1. 所有交图 G_s 都是（非常）良好的采样器，并且
2. G 具有“低拥塞局部路由方案”。

第一条性质不言自明，但第二条需要稍作讨论。路由方案 是一种小型低深度电路，它以 s_{in}， $s_{out} \in S$ 为输入，输出一条连接 s_{in} 和 s_{out}、位于 G 中的短路径 P。粗略地说，当 G 中没有边在过多路径中被重复使用，或者等价地说，当方案中没有一小组边被过多路径经过时，路由方案就具有低拥塞 （我们采用的精确定义见定义 3.3）。从本质上说，我们利用这种路由，将辅助信息从一个随机起始 s_{in} 局部传播到一个随机$s_{out} \ni v$，以便解码顶点 v。如果 G_s 是足够好的采样器，那么图中对传播解码而言“坏”的边就非常少；而低拥塞则保证我们的路由通常会避开这组坏边，并以高概率输出正确的 $f(v)$。

最后一个组成部分是实例化上述框架。具体而言，我们需要找到一族稀疏且显式的超图，其超边具有结构化的重叠（以确保交集采样）、具有良好的“全局”扩张性（以确保存在短路径），并且有一种强显式算法，能够以足够对称的方式，在系统中的任意两个超边之间进行局部路由 G，从而确保低拥塞。令人惊讶的是，我们证明，基于子空间的超图系统可以用一种完全初等的方法实现这一点，且度为次多项式。稍后我们将看到，这如何融入一系列更广泛的构造中；这些构造基于高维扩展图上的邻域编码，[22] 以及如何修改整体框架，以达到多对数码率，甚至常数码率。

3.1 列表恢复与路由

尽管 Guruswami 和 Indyk 的归约到列表恢复在概念上很简单，也是我们构造的“第一步”，但它有各种难以处理的参数，需要进行设置以匹配底层的可列表恢复码。因此，我们不妨先介绍新的列表恢复过程，然后再讨论如何用恰当且相应的参数，将列表译码归约到列表恢复。

我们将本节关于列表恢复的内容分为两个部分。首先，我们正式描述低拥塞路由方案的概念，然后证明，任何具有此类方案（且采样足够良好）的超图系统，都具有一种简单的基于传播的列表恢复算法。

3.1.1 低拥塞路由

如上文简要所述，我们的列表恢复算法的一个核心要求，是能够在解码图 G_s 中局部计算出用于传播辅助信息的路径。我们将特别需要这些路径具备两个主要性质：1）路径应当较短（这对应于查询复杂度）；2）对于 G中的任意小边集 B，我们的路径经过 B 的概率应当仍然很低（这确保

[22]具体而言，本节中的构造源自对一种非常简单的高维扩展体取邻域，这种高维扩展体称为“旗标复形”或“球面建筑”。

我们不会命中任何小的“坏边”子集）。我们将这些保证抽象为“低拥塞路由方案”的概念：

定义 3.3（低拥塞路由方案）。对于任意 $t \in \mathbb{N}$ 和图 $G = (V, E)$，G 上的长度为 t 的路由方案是一种（可能随机化的）算法 $\mathcal{O}$，它以 $v, v' \in G$ 为输入，输出一条连接 v 和 v'、长度至多为 t 的路径 $\mathcal{O}(v, v')$。若对于每个子集 $B \subset E$，概率 $\mathcal{O}$ 命中 B 至多为……，则称 $\mathcal{O}$ 具有单位拥塞

$$\Pr_{v,v',\mathcal{O}}[\exists e \in \mathcal{O}(v, v') : e \in B] \leq t \Pr[B]$$

如定义所示，为例如扩展图构造低拥塞路由方案并非特别困难。真正的困难在于上文提到的第三个关键标准：该方案必须具有局部实现。换句话说，我们希望 $\mathcal{O}$ 能由一个规模为 polylog $|V|$、深度较低（例如 poly log log $|V|$）、对数空间均匀的电路类实现。

我们强调，即使不考虑困难性放大，构造此类局部方案在应用中也至关重要；否则，就无法计算出解码特定顶点 v 所需的查询集。事实上，没有这种实现，我们甚至无法实现次线性时间局部译码，更不用说本工作最终构造的次多项式时间恒定码率局部列表可解码码了。

在本节及后续章节中，我们将了解如何利用子空间（以及随后利用群/多项式）结构，即使在高度稀疏的超图系统上，也能构建此类高效方案。事实上，在后续章节中我们将展示更强的结论——存在局部路由方案，即使图中有 1% 的部分被对抗性地篡改（以一种“可测试”的方式），它们仍然能够正常工作；这是实现常数码率的关键。[23]

3.1.2 超图系统的列表恢复

现在我们将证明，任何具有低拥塞路由方案且其交图是良好采样器的超图系统，都拥有一种简单的列表恢复算法。

定理 3.4。令 $\delta > 0$ *and* $\ell_{in}, t \in \mathbb{N}$。令$(V, S, G)$ 为一个正则超图系统，使得

1. 对于每个 $s \in S$，G_s 都是一个 $\frac{10^{-3}\delta^3}{t\ell_{in}}$-采样器。*2.* G 拥有一个长度为 t 的单位拥塞路由方案 $\mathcal{O}$。

那么，$\mathcal{C}_{(V,S)}$ 是一个$(\delta, 15t\delta)$-近似、ℓ_{in}-输入、$(1 - \frac{10^{-5}}{t}, \ell_{in}, t)$-*LLRC*。

定理 3.4 只陈述了我们所构造的码的组合性质。如前所述，我们的译码算法还必须高效得多，这一点至关重要；也就是说，输出电路应当规模小、深度低，并且能够高效生成。我们证明，只要可以预言机访问超图及路由方案的电路实现（电路实现的定义见第 2.2 节），上述性质便成立：

引理 3.5 (定理 3.4 的电路复杂度)。给定$(C_V, C_S, C_{V^{-1}}, C_{S^{-1}}, C_I)$的电路实现$(V, S)$、输入列表 $C_{\mathcal{L}}$ 的电路实现，以及 $\mathcal{O}$ 的电路实现 $C_{\mathcal{O}}$，所生成的译码电路 $\{C_1, \ldots, C_{\ell_{in}}\}$ 是对数空间一致的

1. 规模：规模$(C_i) \leq \text{poly}(t, \ell_{in}, \delta^{-1})$*2.* 深度：深度$(C_i) \leq O(\log(\ell_{in}) \log(t))$

对 $C_{\mathcal{O}}$和 C_V进行一次预言机调用，并对 C_I、$C_{S^{-1}}$和 $C_{\mathcal{L}}$进行 $\tilde{O}(\frac{t\ell_{in}^2}{\delta})$（并行）预言机调用

我们的主要解码过程由图 2 中的传播算法给出。粗略地说，该过程从超边 s_1 开始，并假设其列表中的某个固定函数 $g \in \mathcal{L}_{s_1}$ 接近我们希望解码的全局 f（实际上，对于每个 g，我们都会输出一个电路）。给定目标 v后，算法抽样一个随机超边 $s_t \ni v$（步骤 1），然后调用路由方案，寻找从 s_1到 s_t 的路径（步骤 2）。现在的目标是基于初始辅助信息 g，在 s_t 的列表中找到一个函数 g_t，使其以高概率接近 f（随后我们可以通过输出 $g_t(v)$ 来进行解码）。

[23]注意，t 路由方案的长度至少必须等于图的直径，因此在我们研究的设置中为多对数(N)。即使算法“知道”哪些顶点是不良的，在这种情况下处理 1% 的错误仍然极具挑战性。

具体通过以下步骤（3）至（4）中描述的迭代过程完成：我们为路径中的每个 s_i 构建一个“良好候选项”列表 L_i。具体而言，从辅助信息 $L_1 = \{g\}$开始，在每一轮中，我们将 L_{i+1} 计算为 s_{i+1} 列表中与上一轮的 L_i 中任意函数接近的所有函数（根据 $s_i \cap s_{i+1}$的交集进行测量）。我们论证，在该过程结束时，L_t 将非空，并且 每个 L_t 中的函数都将以高概率接近 f，因此我们可以任意选择其中一个来解码 v。我们指出，利用 Savitch 算法 [54]，可以在较低深度下执行这一过程。

在附录 F 中，我们分析了一个非常简单的电路 $(\frac{\log \frac{2t\ell_{in}^2}{\delta}}{\delta}, 4\delta)$ 列表距离测试器，它可以预言机访问公共定义域（本例中为 $s_i \cap s_{i+1}$）上的函数列表 L_1, L_2，并输出一个列表 $L_2' \subseteq L_2$，使得以概率 $1 - \frac{\delta}{2t\ell_{in}^2}$：

1. 如果 $f \in L_2$ 与 2δ 中的某个函数距离不超过 L_1，那么 $f \in L_2'$。

2. 如果 $f \in L_2'$，那么存在某个函数 $f' \in L_1$，使得 dist$(f, f') \leq 6\delta$。

电路在定义域中抽样 $O(\ell_{in} \frac{\log \frac{2t\ell_{in}^2}{\delta}}{\delta})$ 个随机点，并查询所有函数在这些点上的取值。我们采用这一过程，避免在每个 s_i 上处理整个符号，因为其规模可能很大（也就是说，这确保我们的解码电路规模不依赖于字母表，并且在级联后查询次数更少）。

列表恢复算法 $Dec(s_1, g, v)$:

* **输入：**

- 一个集合 s_1 及其列表中的一个成员 $g \in \mathcal{L}_{s_1}$
- 一个目标顶点 $v \in V$

*** 输出：** Σ中的一个值。

1. 抽样一个随机的 $s_t \ni v$。

2. 从 $\mathcal{O}$ 中抽样一条随机路径 s_1，s_2、……、s_t。

3. 设置 $L_1 = \{g\} \subseteq \mathcal{L}_{s_1}$。

4. 对于 $i = 2, 3, \cdots, t$ 执行:

   (a) $L_i \leftarrow (\frac{\log \frac{2t\ell_{in}^2}{\delta}}{\delta}\ 4\delta)$-距离测试器$(L_{i-1}, \mathcal{L}_{s_i})$

   (b) 如果 $L_i = \emptyset$，则输出‘失败’。

5. 随机选择一个 $g_t \in L_t$，并输出 $g_t(v)$。

图 2：列表恢复算法

定理 3.4 的证明。 固定一个输入列表 $\mathcal{L}$ 和 $f \in \mathcal{B}_{1-\varepsilon,\delta}(\mathcal{L})$。我们的目标是证明，对于大多数 s_1，都存在一个列表元素 g_j的选择，使得 $Dec(s_1, g_j, \cdot)$ 近似计算 f:

$$\mathop{\mathbb{P}}_{s_1}\left[\exists g \in \mathcal{L}_{s_1} : \mathop{\mathbb{P}}_{v,r}\left[Dec(s_1, g, v) = f(v)\right] \geq 1 - 15t\delta\right] \geq 0.99$$

特别是，输出 削减电路 $\{Dec(s_1, g, \cdot)\}_{g \in \mathcal{L}_{s_1}}$ 即可得到所需的列表恢复保证。

为此，任取输入列表 $\mathcal{L}$ 和 $f \in \mathcal{B}_{1-\varepsilon,\delta}(\mathcal{L})$。我们首先定义几个罕见的「坏事件」，然后论证在这些情况之外，解码能够成功。具体而言，令 B 表示满足以下任一条件的边集 $\{s, s'\}$： $s'' \in \{s, s'\}$

1. **坏列表：** 不存在与 f 距离小于 δ 的 $g'' \in \mathcal{L}_{s''}$

2. **不良采样：**存在某个函数 $g'' \in \mathcal{L}_{s''}$，使得

$$\left|\operatorname*{dist}_{s''}(f, g'') - \operatorname*{dist}_{s \cap s'}(f, g'')\right| > \delta.$$

如果路径中没有任何一条边属于 B，我们就称路径 $(s_1, \ldots, s_t)$ 是良好的。我们断言，只需证明对于大多数 s_1，其通往随机端点 s_t 的预言机路径大多是良好的：

$$\Pr_{s_1}\left[\Pr_{s_t, \mathcal{O}}[\mathcal{O}(s_1, s_t) \text{ is good}] \geq 1 - \frac{\delta}{2}\right] \geq 99\% \tag{1}$$

我们先假设这个断言成立，并完成证明。任取满足内部条件的 s_1。根据假设，存在某个与 f 的距离为 δ 的 $g \in \mathcal{L}_{s_1}$。我们断言，对于任意这样的固定选择，$\mathsf{Dec}(s_1, g, v)$ 成功解码随机 v 对应的 $f(v)$ 的概率至少为 $1 - 15t\delta$。

为说明这一点，先回顾一下：给定输入 v，Dec 会抽取一个随机 $s_t \ni v$，通过 $\mathcal{O}(s_1, s_t)$ 找到通往它的路径，并针对某个有效的 $g_t \in L_t$ 输出$g_t(v)$。因此，我们的目标是给下式一个下界：

$$\Pr_{r,v}[Dec(s_1, g, v) = f(v)] = \Pr_{v \in s_t, \mathcal{O}, g_t}[g_t(v) = f(v)]$$

其中 $g_t \in L_t$ 是均匀随机选取的，且与 v 相互独立。

现在考察右侧概率。注意，我们先均匀随机抽取了 $v \in V$，再抽取一个随机邻居 $s_t \ni v$。由于 (V, S) 是正则的，这等价于先抽取一个随机 $s_t \in S$，再对子采样一个随机顶点 $v \in s_t$。这意味着，只需证明：在 s_t, $\mathcal{O}$ 的随机性下，最终函数 g_t 在 s_t 上以高概率是与 f 的 $O(t\delta)$-近似：

$$\Pr_{s_t, \mathcal{O}}[\forall g_t \in L_t : \operatorname{dist}(g_t, f) \leq 14t\delta] \geq 1 - \delta$$

特别地，该概率，我们在对 $s_t \in S, g_t \in L_t$ 与 $v \in s_t$ i的随机选择下正确输出 $f(v)$ s 至少为$(1-\delta)(1-14t\delta) \leq 1 - 15t\delta$，如所期望。

为证明这一点，我们引入最后一个坏事件，在该事件之外我们的解码器能够成功：即步骤4(a)中的距离测试器出错的事件（回顾一下，我们对交集中的点进行子抽样以测试距离，而不是检查完整符号，以确保我们的译码电路保持较小）。根据构造（以及断言 F.1），该事件发生的概率至多为 $\frac{\delta}{2}$。

现在对路径 $\mathcal{O}(s_1, s_t)$为良好与上述距离测试器在该路径上不出错的事件进行条件化（由并集界，其发生的概率至少为 $1-\delta$）。我们断言，在此条件化下，每一个 $g_t \in L_t$ 在我们从所承诺的、与 f为 δ-接近的 g 出发时，都与 f 为 $14t\delta$-接近。更形式化地说，我们将用归纳法对沿我们路径的每一个 L_i 论证以下两个断言：

如果这些事件都没有发生，那么我们可以论证：

1. L_i 包含一个在 s_i上与 f 的距离为 δ的函数。
2. L_i 中的每个函数都与 f的距离为 $14i\delta$。

根据这些断言，显然 L'_{s_t} 中的任意函数都与 f 的距离为 $14t\delta$，因此它在其 $1 - 14t\delta$ 个顶点上都能给出正确答案，这正是我们需要的。

我们通过对 i进行归纳法来证明这两点。在证明中，我们将用到距离测试器从不出错这一事实；回顾一下，这意味着：(1) 任意两个距离为 4δ的函数都会通过测试；(2) 如果一对函数通过了距离测试，那么它们之间的距离至多为 12δ。

现在，首先注意，根据假设，L_1 的基础情形成立。因此，假设这两项对 L_{i-1} 均成立。我们将证明它们对 L_i 成立。先看第一项，假设存在某个 $g_{i-1} \in L_{i-1}$，它与 f 的距离不超过 δ。根据假设，$\{s_{i-1}, s_i\} \notin B$（第一项），存在某个 $f_i \in L_i$，它与 f 的距离不超过 δ。由于这条边不在 B 中（第二项），

$$\operatorname*{dist}_{s_{i-1} \cap s_i}(g_i, f) \leq \operatorname*{dist}_{s_{i-1}}(g_{i-1}, f) + \delta \leq 2\delta,$$

$$\operatorname*{dist}_{s_{i-1} \cap s_i}(g_i, f) \leq \operatorname*{dist}_{s_i}(g_i, f) + \delta \leq 2\delta.$$

因此 b　　根据三角不等式，$\mathrm{dist}_{s_{i-1}\cap s_i}(g_{i-1},\ g_i) \le 4\delta$。由于测试器没有出错，$g_i \in L$　　$_i$.

接下来讨论第二项，假设 L_{i-1} 中的所有函数都与 f 14$(i-1)\delta$-接近。和之前一样，因为 $\{s_{i-1}, s_i\} \notin B$，所以对每个 $g_{i-1} \in L_i$

$$\operatorname*{dist}_{s_{i-1}\cap s_i}(g_{i-1}, f) \le \operatorname*{dist}_{s_{i-1}}(g_{i-1}, f) + \delta \le (14(i-1)+1)\delta.$$

由于测试器没有出错，对每个 $g_i \in L_i$，我们还有

$$\operatorname*{dist}_{s_{i-1}\cap s_i}(g_{i-1}, g_i) \le 12\delta.$$

因此，根据三角不等式，

$$\operatorname*{dist}_{s_{i-1}\cap s_i}(g_{i-1}, f) \le \operatorname*{dist}_{s_{i-1}\cap s_i}(g_{i-1}, f_i) + \operatorname*{dist}_{s_{i-1}\cap s_i}(g_{i-1}, g_i) \le (14(i-1)+1)\delta + 12\delta = (14(i-1)+13)\delta.$$

再次利用边 $\{s_{i-1}, s_i\} \notin B$ 这一事实，我们有

$$\operatorname*{dist}_{s_i}(g_i, f) \le \operatorname*{dist}_{s_{i-1}\cap s_i}(g_i, f) + \delta \le 14i\delta.$$

最后，还需证明公式 (1)，即许多 s_1 的坏路径很少。为此，首先注意到，根据马尔可夫不等式，只需证明随机选择 s_1，s_t 和路径 $\mathcal{O}(s_1,\ s_t)$ 时，概率$\mathcal{O}(s_1,\ s_t)$ 命中 B 的概率非常低，例如：

$$\Pr_{s_1, s_t, \mathcal{O}}[\mathcal{O}(s_1, s_t) \text{ hits } B] \le \frac{\delta}{200}.$$

这现在恰好是由低拥塞路由所界定的情形。具体而言，由于 $\mathcal{O}$ 是单位拥塞，只需证明 B 的测度很小，即 $\Pr[B] \le \frac{\delta}{200t}$。这基本上直接由采样以及如下承诺得出：除至多 $10^{-3}\frac{\delta}{t}$ 比例的 $s \in S$ 外，其余都包含一个与 f相距δ的函数。后者通过并集界保证，随机边满足“坏列表”条件的概率至多为 $2^{10^{-3}}_{t}$。前者表明，对任意固定的 s，随机邻居 s'对 s 的列表中任一固定函数发生错误采样的概率至多为 $\frac{10^{-3}\delta}{t\ell_{in}}$。对随机边中 s 与s' 两者的列表作并集界，该事件发生的概率因此至多为 $2^{10^{-3}\delta}_{t}$，最后一次并集界即给出所需的$\Pr[B] \le \frac{\delta}{200t}$。

□

我们简要指出，上述证明有一个良好性质：它对所查询的每个符号仅读取 $O(\frac{1}{\delta}\log\frac{t\ell_{in}^2}{\delta})$ 比特。我们将所读取的比特数（而非完整符号数）称为该解码器的“细粒度查询复杂度”。

注 3.6（细粒度查询复杂度）。 定理 *3.4* 的细粒度查询复杂度至多为$O(\frac{t}{\delta}$ 对数 $\frac{t\ell_{in}^2}{\delta})$。

这在附录 A 中会很有用，届时我们希望构造查询最优串接码，并为读取输入字的每一位付出代价。现在，我们还需要证明所提出的译码算法 Dec$(s,\ g,\ \cdot)$ 的电路复杂度满足所述上界：

引理 *3.5* 的证明。 我们回顾一下，在 $Dec(s_1,\ g,\ \cdot)$ 中作为黑盒使用的各个组件：

1. 该　第 2 节中描述的超图电路实现（C_V，C_S，$C_{V^{-1}}$，$C_{S^{-1}}$，C_I）。　　2
2. 一个实现该路由方案的电路 $C_{\mathcal{O}}$。
3. 一个电路 $C_{\mathcal{L}}$，其输入为 $s \in S$ 和一个邻居 $v \sim s$（形式上，是与该邻居 v 对应的索引 i），并为 $g^{(j)}$ 中的第 j 个成员输出 $\{g^{(j)}(v)\}_{j\in[\ell_{in}]}$。

现在，我们描述如何使用对上述内容的预言机调用来实现我们的 $Dec(s_1,\ g,\ \cdot)$　　：

1. Dec 将输入 $v \in V$（以及一个随机索引 i_v）输入 C_V，以生成 $s_t \ni v$。

2. s_t 被输入 $C_{\mathcal{O}}$（同时附带硬编码的 s_1 和一个额外的随机字符串 $r_{\mathcal{O}}$），以生成路径 s_1，s_1，……，s_t。

3. 对于每条边 $\{s_{i-1}, s_i\}$，使用 $d = O(\frac{\log \frac{t\ell_{in}}{\delta}}{\delta})$ 个 C_I 副本，在 $s_i \cap s_{i-1}$ 中抽样 d-随机点。

4. 使用 $C_{S^{-1}}$ 的 $2d$ 个副本，计算这些抽样顶点分别对应于 s_i 和 s_{i-1} 的哪些邻居。

5. 将这些值输入 $C_{\mathcal{L}}$，以针对每对函数 $g_{i-1} \in \mathcal{L}_{s_{i-1}}, g_i \in \mathcal{L}_{s_i}$ 计算 $g_{i-1}(v)$ $g_i(v)$；并将 $(g_{i-1}(v_1) \ldots, g_{i-1}(v_d)) \oplus (g_i(v_1) \ldots, g_i(v_d))$ 输入一个 4δ-阈值电路，以实现距离测试器。[24] 每个阈值电路都可以用深度为 $O(\log(d))$、规模为 $O(d \log d)$ 的电路实现。

6. 现在，我们实现一个电路，以这些值作为输入，并输出一个随机$g \in L_t$的索引。为此，我们在这些列表上构造如下“邻接图”

(a) 顶点是所有列表 $\mathcal{L}_{s_1}, \mathcal{L}_{s_1}, \ldots, \mathcal{L}_{s_t}$ 中的函数。(b) 边位于连续的 $\mathcal{L}_{s_{i-1}}, \mathcal{L}_{s_i}$ 之间，其中 $g_{i-1} \sim g_i$ 若对应的阈值电路输出 1（即它们被测量为 4δ-接近）

现在可以观察到，函数 $g'' \in \mathcal{L}_{s_t}$ 属于 L_t（如图 2 所述）当且仅当在上述图中存在一条从 $g \in \mathcal{L}_{s_1}$ 到 g'' 的 t-长度路径。因此，我们可以输出 L_t 中函数的（指标），所用电路为 C_{Sav}，该电路实现 Savitch 算法 [54]。在我们的设定中，该图有 t 层，每层有 ℓ_{in} 个顶点，通过复用大量递推计算，Savitch 算法可以以对数空间一致的深度 $O(\log t \log \ell_{in})$ 和规模 $\tilde{O}(\ell_{in}^3 t)$ 实现（大致而言，$\log t$ 个递推关系层中的每一层都需要在来自上一轮次的各次计算之间，取至多 $\ell^2 t$ 路并行的 ℓ 2 位与的或）。

7. 最后，我们随机选取某个 $g'' \in L_t$，并通过一次额外的对 $C_{\mathcal{L}}$ 的查询（以及对 $C_{S^{-1}}$的查询，以找到 s_t 的适当邻居索引，从而查询 $C_{\mathcal{L}}$）输出 $g''(v)$。

因此，我们总共进行了所声称次数的预言机调用，此外还有

$$\mathrm{Depth}(\mathrm{Dec}(s_1, f, \cdot)) \leq \underbrace{O\left(\log\left(\frac{\log(t\ell)}{\delta}\right)\right)}_{\text{Distance Tester}} + \underbrace{O(\log t \log \ell_{in})}_{\text{Savitch's Algorithm}}$$

and

$$\mathrm{Size}(\mathrm{Dec}(s_1, f, \cdot)) \leq \mathrm{poly}(t, \ell_{in}, \delta)$$

□

3.2 从列表译码到列表恢复

我们现在给出Guruswami和Indyk的 [43, 53] 归约到列表恢复的一个简单局部变体。该归约的关键性质是：它从仅有 ε比例的集合包含我们想要计算的信息的含噪设置，转移到一个更干净的设置，其中$(1 - o(1))$比例的集合包含关于我们想要计算的函数的信息（尽管该信息隐藏在一个函数列表中）。这与前一小节所述的列表恢复算法相匹配，该算法要求除 $O(\frac{1}{t})$比例之外的所有码字块，其列表中都有一个函数接近我们想要解码的全局 f。

归约背后的想法极其简单。假设我们有一个定义在集合系统 (V, S) 上的直积码，以及一个正则采样器图 $G' = (L, R, E)$，其中 $L = S$。通过将二者“组合”起来，我们构造出第二个直积码 (V, T)，其中超边 $T = \{t_r\}_{r \in R}$由 R 中的顶点索引，并且在形式上对应于多重集

$$t_r := \biguplus_{s \sim r} s$$

其中，多重集并集是对采样器 G' 中 r 的邻居取并集。换句话说，如果 (V, S) 是 r_1-均匀的，且 G' 是 r_2 右正则的，那么复合码会将消息 $f\colon V \to \Sigma$ 编码为码字 $\mathrm{Enc}^f\colon T \to (\Sigma^{r_1})^{r_2}$；对于每个邻接的 $s \sim r$，在输入 $t_r \in t$ 时，它都会输出符号 $f(s) \in \Sigma^{r_1}$。

[24] 注意，为了实现电路，我们对图 2 中的这一步稍作修改：在整个列表上计算阈值，而不是在当前的 L_{i-1} 上计算。

构造的要点如下。假设现在有一个含噪输入字，它在 $t_r \in T$ 的 ε 比例上与 Enc^f 一致。通过采样，在原始超图系统中随机选取一个 $s \in S$，以高概率至少会邻接这些“良好”符号中的 $\frac{\varepsilon}{2}$ 比例。这意味着，我们只需记下其邻居 t_r 中反复出现的任意符号，就能在原始超边 s 上解码出 f 的候选列表。这就将问题化归为列表恢复情形，此时我们只需对 (V, S) 运行上一节中的算法即可。

下面的引理将上述结果形式化，并将其推广到近似一致度的情形；具体参数取自上一节的列表恢复算法。

引理 3.7. 对任意 $\varepsilon > 0$、$t \in \mathbb{N}$ 和 $\delta \le \frac{10^{-5}}{16t}$，假设 $\mathcal{C}_{(V,S)}$ 是一个 $(\delta, 15t\delta)$-近似、$100\log_{\varepsilon} t$ 输入、$(1 - \frac{10^{-5}}{t}, 100\log_{\varepsilon} t, t)$-LLRC。那么对于任意 $\frac{10^{-5}}{4t\varepsilon^2}$-采样器图 $G' = (S, R, E)$，相应的直积码 (V, T) 是一个 $(\delta^2, 15t\delta)$-近似 $(\varepsilon, 200\log_{\varepsilon} t, 100t\log_{\varepsilon} t)$-局部列表可解码码。

证明。固定一个输入字 $w = \{w_{t_r}\}$ 和一个全局函数 $f \in \mathcal{B}_{1-\varepsilon,\ \delta^2}(w)$，我们希望对其进行解码。我们将从 w（隐式地）构造一个规模为 100 的随机列表集 $\mathcal{L}_S = \{L_s\}_{s \in S^{\log_{\varepsilon} t}}$，使得以 99% 的概率，$f \in C_{1-\frac{10^{-5}}{t},\ \delta}(\mathcal{L}_S)$。给定对这些列表的查询访问后，列表恢复算法会生成一个 ℓ_{out}-电路列表，其中有一个电路保证以 $3/4$ 概率与 f 足够接近，只要上述高概率事件成立。因此，将列表恢复算法运行两次，即可确保以至少 $3/4$ 的概率输出这样的电路，达到所需效果。

现在我们讨论如何构造随机列表族 $\mathcal{L}_S = \{L_s\}_{s \in S}$，以及列表恢复算法如何根据 w 按需计算出 L_s。具体而言，$s \in S$ 对应的列表 L_s 将通过选取 $100\log_{\varepsilon} t$ 个随机邻居 $r \sim s$，并输出 w_{t_r} 在 s 上的值来生成。为确保这能由一个小规模电路实现，我们将对所有 $s \in S$ 使用相同的随机性选择。换句话说，这些指标会预先选定并硬编码到输出电路中，从而确定一个特定的列表族 $\mathcal{L}_S$；随后，解码电路只需对 w 进行 $100\log_{\varepsilon} t$ 次查询，即可确定性地计算出该列表族。

现在我们需要论证，上述选取 $\mathcal{L}_S$ 的过程确实会生成一个列表集合，使得至少以 99% 的概率，$f \in C_{1-\frac{10^{-5}}{t},\ \delta}(\mathcal{L}_S)$。根据马尔可夫不等式，只需证明：对于除比例为 $\frac{10^{-5}}{2t}$ 的 s 之外的所有 s，L_s 以至少 $1 - \frac{10^{-6}}{2t}$ 的概率包含一个与 f 相差不超过 δ 的函数。

为此，定义如下“坏”超边集合 s：它们没有足够多的良好邻居

$$B = \left\{ s \in S : \mathop{\mathbb{P}}_{r \sim s} [\mathrm{dist}(w_{t_r}|_s, f|_s) < \delta] < \frac{\varepsilon}{4} \right\}. \tag{2}$$

我们将证明 $\mathop{\mathbb{P}}_{s}[B] \le \frac{10^{-5}}{2t}$。根据切尔诺夫界，显然，对于任意 $s \notin B$，L_s 以至少 $\frac{10^{-6}}{2t}$ 的概率包含一个与 f 相差不超过 δ 的函数。

为了界定 $\Pr[B]$，我们进一步将其分为两个“坏事件” B_1，B_2，使得 $B \subseteq B_1 \cup B_2$。令 $E \subseteq R$ 为所保证的右侧顶点集合，满足 w_{t_r} 与 f 相差不超过 δ^2。我们将 B_1 定义为所有满足 $\mathop{\mathbb{P}}_{r \sim s}[E] < \frac{\mathbb{P}[E]}{2}$ 的集合 $s \in S$。根据假设，$\Pr[E] \ge \varepsilon$，对 G' 进行采样可保证 $\mathop{\mathbb{P}}_{s \in S}[B_1] < \frac{10^{-5}}{4t}$。令 $B_2 \subseteq S \setminus B_1$ 为所有这样的 $s \in S$：其邻域中至少有占 $r \in R$ 的 $\frac{\mathbb{P}[E]}{2}$ 比例的顶点能够 δ^2-计算 f，但这些良好邻居中至少一半 $r \in E \cap N_G(s)$ 与 f 相差超过 δ。显然，$s \notin B_1 \cup B_2$ 蕴含 $s \notin B$。

为界定 $\Pr[B_2]$，注意对于 G' 的一条随机边 $(s,\ r)$

$$\mathop{\mathbb{P}}_{s \sim r}[r \in E, \mathrm{dist}(w_{t_r}|_s, f_s) > \delta] \le \delta \mathop{\mathbb{P}}_{r}[E].$$

只需先从 R 中均匀抽取 r，再将 $s \sim r$ 视为随机邻居来考察边 $(s,\ r)$，即可立即得到这一结论。具体而言，初始选择恰为 $\Pr[E]$ 的概率是如此；对于任意 $r \in E$，马尔可夫不等式保证，至多有 $s \sim r$ 的 δ 部分满足 $w_{t_r}|_s$ 与 $f|_s$ 的距离至少为 δ。另一方面，根据定义，每个 $s \in B_2$ 至少能看到 $\frac{\mathbb{P}[E]}{4}$ 条这样的边，因此这类 s 至多占 $4\delta \le \frac{10^{-5}}{4t}$ 的比例，正如所需。 □

我们注意到，假设可以预言机访问图 G' 的邻域电路实现，则此过程的输出电路与原始输入的输出具有相同的渐近电路复杂度

列表可恢复码，因为唯一的改动是：1）运行原始码两次；2）通过查询 G' 中一组硬编码的邻居来模拟输入列表。每一步进行距离测试时，“精细化”查询复杂度还需要额外乘以 $O(\frac{1}{\delta}\log\frac{t}{\delta\varepsilon})$。

最后，在继续之前，我们给出两点简单说明。第一点是，引理 3.7 可以很容易地推广到高噪声列表恢复情形，其中我们记为 $\varepsilon = 1-\eta$，而这正是我们的串接码所需要的：

推论 3.8 $\frac{3}{200}\frac{7}{100\ell_{in}}\ell_{in}\ell_{out} \le .$（GI 列表恢复）。引理 . 中的输出码是一个 -输入近似局部列表可恢复码，具有 $\log_{\varepsilon} t$，并且其余参数相同，只要 $\mathcal{C}_{(V,S)}$ 是 $\log_{\varepsilon} t$-输入列表可恢复的。

证明完全相同，其中每个 L_s 都是通过取其选定邻居处列表的并集来构造的。事实上，甚至很容易验证上述结果甚至是一个“单电路”近似局部列表可恢复码（见定义 A.3），因为很容易验证我们可以对每个输出电路使用同一个查询集。这使得我们能够在级联后达到近乎最优的查询复杂度。

其次，我们指出，按目前的表述，引理 3.7 和推论 3.8 有一个不理想的性质：它们的列表大小会随 t 增长，而 t 本身又会随 N 增长。这是我们上面给出的简化证明所造成的结果，并非必要。在第 4 节中，我们将讨论一种简单的剪枝过程，将列表大小改进至渐近最优界 $O(\frac{\ell_{in}}{\varepsilon})$。

3.3 子空间 aLLDC

到目前为止，我们的陈述都是抽象的，即我们只规定了超图系统需要具备哪些性质才能实现列表译码。在本小节中，我们将通过一个基于子空间的简单集合系统补全这一图景；该系统使我们能够实例化引理 3.7 和定理 3.4，从而证明定理 3.1。尽管后续各节中介绍的系统能够实现显著更优的参数，但这里介绍的系统更加简单，并且不要求具备高维扩展图方面的背景知识。

3.3.1 子空间集合系统

设 $\mathfrak{q}$ 为素数幂，$d>1$ 为整数，$\mathbb{F}_{\mathfrak{q}}^{2d}$ 为向量空间。我们的码就是定义在 $\mathbb{F}_{\mathfrak{q}}^{2d}$ 的 $(d-1)$ 维子空间与 $(d+1)$ 维子空间之间的包含图上的一个直积。下面，我们用超图系统的语言对此作出正式表述。

对于 $\mathbb{F}_{\mathfrak{q}}^{2d}$ 中的子空间 v，u，记 $v \le u$，表示 u 是 v 的线性子空间。令 $\mathcal{SS}_{\mathfrak{q},\ d} = (V,\ S)$ 为一个超图，其顶点

$$V = \left\{v \le \mathbb{F}_{\mathfrak{q}}^{2d} : \dim(v) = d-1\right\}$$

是 $\mathbb{F}_{\mathfrak{q}}^{2d}$ 的所有 $d-1$ 维子空间。为了描述超边（以及后面的交图），对于维数大于 $d-1$ 的任意子空间 $u \le \mathbb{F}_{\mathfrak{q}}^{2d}$，记 s_u 为包含于 v 中的所有 $d-1$ 维子空间的集合

$$s_u \coloneqq \{v \in V : v \le u\}.$$

我们将所有这类 s_u 在维数为 $d+1$ 时的情形作为超边：

$$S = \{s_u : \dim(u) = d+1\}$$

换句话说，我们的消息比特位于 $(d-1)$ 维子空间上，而我们的编码由 $(d+1)$ 维子空间索引；这些子空间记录其下方某个 $(d-1)$ 维子空间上包含的每一比特。

我们简要说明该系统（V，S）上直积码的参数。消息长度为 $|V| = N = (1 \pm o(1))\mathfrak{q}^{d^2-d}$。编码长度也为 $|S| = |V| = N$，因为 $\mathbb{F}_{\mathfrak{q}}^{2d}$具有相同数量的（$d-1$）维和（$d+1$）维子空间。$s_u \in S$的集合规模（对应于（对数）字母表）为 $|s_u| = (1 \pm o(1))\mathfrak{q}^{2d-2}$。对于给定的目标消息长度 N 以及 $\varepsilon \le \frac{1}{\mathrm{polylog}(N)}$，大致可以认为取 $\mathfrak{q} \sim \mathrm{poly}(\frac{1}{\varepsilon})$ 和 $d \sim O\left(\sqrt{\log N \log\frac{1}{\varepsilon}}\right)$（精确参数见第 3.3.3 节）。因此，该码的码率为 $2^{\Theta(\sqrt{\log N}\log\frac{1}{\varepsilon})}$，最差情况下为 $2^{\Theta(\sqrt{\log N}\log\log N)}$，其中 ε 为常数。

解码图 我们系统的解码图 G，我们记得它是顶点集$S = \{s_u$ 上的一个图：$\dim(u) = d+1\}$，在 s_u 与 $s_{u'}$ 之间有边，当且仅当 $\dim(u \cap u') = d$。我们注意到该交集等于 $s_u \cap s_{u'} = s_{u\cap u'}$，即所有包含 $u \cap u'$的子空间。容易验证这是一个正则超图系统，且 G 上的平稳分布是均匀的。

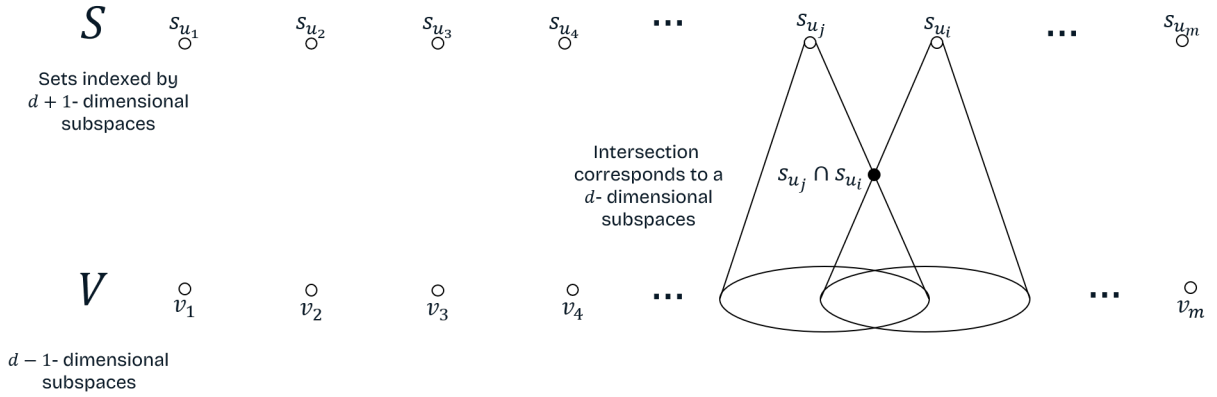

图3：子空间集合系统

交图 我们现在描述交图 $G_{s_u} = (L, R, E)$，并论证它们是良好的采样器。右侧是 s_u 本身中的子空间。左侧由可能的交集索引。我们可以根据与 s_u的交集，将 L 中的顶点划分为集合，即$t_w = \{s_{u'}:$ 对所有维数为 d 的 $w \leq u$， $u \cap u' = w\}$。我们注意到所有 t_w 具有相同的规模，且每一对$s_{u_1}, s_{u_2} \in t_w$ 具有相同的邻居集合，即 $N(t_w) = \{v \leq w:\ \dim(u) = d-1\}$。

命题3.9. 对 于每个 $s_u \in S$，图 G_{s_u} 是一个 $\left(\sqrt{q}^{2}\right)$-扩展图。因此，它也是一个 (q^{4})-采样器 .

这一断言相当标准；类似的变体例如出现在 [36, 引理2.12]中。

证明。 无需分析 G_{s_u}，只需分析将 t_w 中的所有顶点替换为一个邻域为 $N(t_w)$的单一顶点后得到的图。该图正是维数为 d 的子空间与位于 u中、维数为 $d-1$ 的子空间之间的包含图。进而，众所周知，经由将 $w \leq u$ 映射到 $w^{\perp} = \{x \in u$，它同构于直线-平面图：$\forall y \in w, \langle x, y\rangle = 0\}$，其中 $\langle x, y\rangle = \sum_{i=1}^{2d} x_i y_i$是标准双线性形式。最后，直线-平面图是 $\left(\sqrt{q}^{2}\right)$-扩展图是一个标准事实，因此 G_{s_u} 本身是 $\left(\sqrt{q}^{2}\right)$-扩展图，该结果由断言 2.9得出。 □

Guruswami-Indyk采样器 上述构造描述了我们将用于列表恢复的超图系统（V， S， G）。我们还需要一个相应的二部采样器 $G' =$ （S， R， E），其左顶点对应于 S，以便应用 Guruswami-Indyk 归约。幸运的是，在这种情况下，我们手头已经有这样一个扩展图：(V， S) 本身！更准确地说，我们将使用包含图 (S， V)，其左侧为 $(d+1)$ 维子空间，右侧为 $(d-1)$ 维子空间。这就是所谓的（广义）“Grassmann图”，众所周知，它是一个 (q^{1})-扩展图（例如，参见 [93]），因此也是一个 (q^{2})-采样器。选择这个图至多会使对数字母表的平方（因此码率仍为$2^{\Theta(\sqrt{\log N \log \frac{1}{\varepsilon}})}$），不会影响最终码长，因为 $|S| = |V|$；此外，这种选择也很方便，因为我们无需针对新的 G' 实现第二类邻接电路。

3.3.2 低拥塞路由

令 $s_u, s_{u'} \in S$。直观而言，我们的路由方案相当简单：为交集$u \cap u'$选取一个随机基，再分别将该基随机扩展到 u 和 u'，然后通过逐个替换基向量进行路由，从 u 移动到 u'。更形式化地说，考虑以下路由方案：

1. 定 义 $k := \dim(v \cap v')$，并选取一个随机基 $x_{d+1-k}, x_{d+2-k}, \ldots, x_{d+1}$到 $u \cap u$ $'$.

2. 选取 $d-k$随机向量 y_1， y_{k+2}，……， $y_{d-k} \in v$，以将 x_{d+1-k}， x_{d+2-k}，……， x_{d+1}扩展为u的一个基，并类似地选取随机向量 z_1， z_2，……， $z_{d-k} \in u'$用于 u'。

3. 令 $u_1 = u$、 $u_{d+1-k} = u'$，并对 $i \in [d-k]$:

$$u_i = span(z_1, z_2, \ldots, z_i, y_{i+1}, y_{i+2}, \ldots, y_{d-k}) + (v \cap v').$$

4. 输出路径 s_{u_1}， s_{u_2}， ……， $s_{u_{d+1-k}}$。

断言 3.10。上述路由方案具有以下性质:

1. 长度 d

2. 单位拥塞

3. 它可以由规模为 $\mathrm{poly}(d, \mathfrak{q})$ 且深度为 $O(\log^2(d)\log(\mathfrak{q}))$ 的电路实现。

证明。第一条性质由构造直接得出。第二条性质源于我们的路由方案在 G上"均匀分布"。换言之，若我们随机选取起点和终点 s_u 和$s_{u'}$，从该方案中随机选取一条路径 $\mathcal{O}(s_u, s_{u'})$，最后再从路径中随机选取一条边 $(s_{u_i}, s_{u_{i+1}})$，根据对称性不难看出，最终得到的边在 G上均匀分布。因此

$$\Pr_{u,u',\mathcal{O}}[\exists e \in \mathcal{O}(u,u') : e \in B] \le d \Pr_{u,u',\mathcal{O},e\in\mathcal{O}(u,u')}[e \in B] \le d\mathbb{P}[B].$$

电路复杂度的证明留到最后一个小节，我们将在那里为本节中的所有操作构造电路。

□

3.3.3 定理3.1 的证明（模电路）

现在我们准备证明本节的主要结果（不过，我们仍将精确的电路实现留到第 3.3.4 节）。首先，我们断言，只要 $\mathfrak{q}$ 足够大，$\mathcal{SS}_{\mathfrak{q},\,d}$ 就是列表可恢复的。

引理 3.11。对于任意 $\varepsilon > 0$、 $d \in \mathbb{N}$，以及满足 $\mathfrak{q} > 3\cdot 10^8 d^3 \log_{\varepsilon^2} d$ 、$\mathcal{SS}_{\mathfrak{q},\,d}$ 的素数幂 $\mathfrak{q}$，d 是一个 $(\frac{1}{1500d}, 0.01)$-近似的、 $100\log_{\varepsilon} d$-输入的 $(1-\frac{1}{10^5 d}, 100\log_{\varepsilon} d, d)$-*LLRC*，且具有定理 *3.4* 中指定的性质。

证明。 命题3.10 指出，$\mathcal{SS}_{\mathfrak{q},\,d}$ 具有长度为 $\le d$ 的单位拥塞路由方案；结合我们对域大小 $\mathfrak{q}$ 的假设，命题3.9 保证 $\mathcal{SS}_{\mathfrak{q},\,d}$ 的所有交图都是 $\frac{10^{-12}\varepsilon}{d^3}\frac{\varepsilon}{d}\log$-采样器。由定理 3.4 取 $t = d$ 和 $\delta = \frac{1}{1500d}$，即可立即得到该引理。 □

我们现在可以证明定理 3.1。

定理 *3.1* 的证明。我们的目标是构造一个作用于 N 个消息比特上的码，其参数为 $\varepsilon \le \frac{1}{\log^2 N}$。我们将证明，存在一组有效的参数，使得这样的码作用于

$$N' \in \left[N, N \cdot 2^{8\sqrt{\frac{\log(N)}{\log\frac{1}{\varepsilon}}}}\right]$$

任意 $\varepsilon \le \log^2(N)$ 个消息比特。随后，只需用上面稍大一些的码对原始消息进行 $\frac{N'}{N}$ 份编码，就能轻松得到所需的作用于 N 比特的码。[25] 为完整起见，我们在引理 E.2 中说明这一点。

基于此，我们的码将由以下码进行（Guruswami-Indyk）复合得到:

1. 引理 3.11 中所述的列表可恢复码，其参数为 $\mathcal{SS}_{\mathfrak{q},\,d} = (V,\ S)$:
   - 一个素数幂 $\mathfrak{q} \in [\frac{1}{\varepsilon^3}, \frac{2}{\varepsilon^3}]$（例如，在此范围内的第一个 s 对应 $\mathfrak{q} = 2^s$）。
   - 满足 $\left[\frac{2d}{d+1}\right]_{\mathfrak{q}} \ge N$ 的第一个整数 $d > 1$。

2. 所讨论的采样器图 $G' = (S, V, E)$，其定义在 $(d+1)$ 维子空间与 $(d-1)$ 维子空间之间

[25] 更正式地说，$\lfloor \frac{N'}{N} \rfloor$ 个副本加上一个截断的最后副本，以达到恰好为 N' 的规模。

现在让我们更仔细地计算这些参数。回想一下，我们的目标消息比特数是 N，而上述 GI 复合实际产生的消息比特数是 $|V| = \begin{bmatrix} 2d \\ d+1 \end{bmatrix}_{\mathfrak{q}} = (1 \pm o(1))\mathfrak{q}^{d^2-d}$。因此，我们取满足 $(1 \pm o(1))\mathfrak{q}^{d^2-d} \geq N$ 的最小 $d > 1$。特别地，对于足够大的 N，可以验证第一个这样的 d 位于：

$$d \in \left[\sqrt{\frac{\log N}{\log \mathfrak{q}}}, \sqrt{\frac{\log N}{\log \mathfrak{q}}} + 2 \right]$$

因此代入 $d = \sqrt{\log N \log q} + 2$ 的上界后，我们得到 $|V| \leq N\mathfrak{q}^{4d} \leq N2^{8\sqrt{\log(N)} \log \frac{1}{\varepsilon}}$，如所述。

现在还可以看出，对于我们选择的参数，有

1. $\mathfrak{q} > \frac{3 \cdot 10^{12} d^3 \log \frac{1}{\varepsilon^2}}{}^{d}$，以及
2. G' 是一个 $\frac{1}{10^7 d \varepsilon^2}$-采样器图

当 N 足够大时。因此，根据引理 3.11 和引理 3.7，上述内容的 GI 复合立即得到一个（$\frac{1}{1500^2 d^2}$，0.01）-近似 $(\varepsilon, \ell_{out}, q)$-局部列表可解码码，其

$$q = O(\frac{d \log d}{\varepsilon})$$

查询复杂度，以及

$$\ell_{out} \leq O(\frac{\log d}{\varepsilon})$$

列表大小。如前所述，列表大小可以改进为仅 $O(\frac{1}{\varepsilon})$（在列表恢复情形下为 $O(\frac{\ell_{in}}{\varepsilon})$），但我们将此讨论推迟到下一节。代入 $d = \Theta\left(\sqrt{\log N \log \frac{1}{\varepsilon}}\right)$ 后即可得到所述参数。

剩下的工作是界定该码的码率。经过与 $G' = (S, V, E)$ 进行 GI 复合后，块长为 $|V|$，这会使码率带来一个 $2^{-8\sqrt{\log N} \log \frac{1}{\varepsilon}}$ 因子，因为 $|V| \leq N2^{8\sqrt{\log(N)} \log \frac{1}{\varepsilon}}$。由于 (V, S) 是平衡且正则的，对数字母表就是图的度 $\begin{bmatrix} d+1 \\ 2 \end{bmatrix}_q$ 的平方；当 N 足够大时，该值至多为 $(1 + o(1))\mathfrak{q}^{4d-4} \leq 2^{8\sqrt{\log N} \log \frac{1}{\varepsilon}}$。因此综合起来，码率至少为 $2^{-16\sqrt{\log N} \log \frac{1}{\varepsilon}}$，如断言所述。

□

最后，我们简要讨论一般参数 ε 下该码的参数，不再假设 $\varepsilon \leq \frac{1}{2} \log(N)$，因为这一假设仅用于简化码率计算和记号。

注释 3.12 (一般 ε)。 对于任意 $\varepsilon > 0$，上述构造都可以实例化为具有码率 $2^{O(\sqrt{\log(N) \log \log(N)})}$，进行 $q \leq \tilde{O}\left(\sqrt{\log N \log \frac{1}{\varepsilon}}\right)$ 次查询，且列表大小为 $\ell_{out} \leq O(\frac{1}{\varepsilon})$（经过第 *3* 节中所述的列表“剪枝”后）。

3.3.4 电路实现

接下来要做的就是实际设计 (V, S) 的电路实现以及我们的路由方案。从整体上看，此处以及后续各节中的电路实现通常包含两个部分。首先，我们将构造一个电路，它接收图中顶点或超边的“名称”（即其比特表示），并输出该对象更便于处理的“规范表示”。在本例中，我们的图由子空间构成，因此将构造一个电路，输入位串并输出相应子空间更合适的矩阵表示。同样，我们还需要一个“逆电路”，将规范表示映射回其对应的名称。其次，实现这些之后，我们将在这个更合适的规范表示空间中实现路由以及各种邻接电路。最后，只需将这些电路与“比特 $\leftrightarrow$ 规范”表示电路组合起来，即可构造出最终的电路实现。

比特表示 $\leftrightarrow$ 规范表示：我们先从基础电路入手，将比特名称转换为子空间表示。有些出人意料的是，如何实际实现这一转换并非显而易见。例如，人们很容易想到直接将输入比特表示$\{0,1\}^T$读作某个基（针对某种适当的 T选择），但这会遇到编码不唯一的问题。事实上，由于每个子空间都有 $\mathfrak{q}^{O(d^2)}$ 个基，采用这种编码会使我们的码的码率呈多项式级增长；显然，在次多项式范围内，我们无法承受这种增长。

因此，我们应当尝试为每个子空间寻找某种标准表示或基。对此有一个自然的选择：将基（或者说，以基向量为行组成的矩阵）化为简化行阶梯形（RREF）。这样便得到一组便于使用且一一对应的表示（因为两个矩阵具有相同的简化行阶梯形，当且仅当它们的行空间相同）。不过，现在不再显而易见的是，如何构造一个简单的电路，将特定输入 $\{0,1\}^T$ 映射为相应的简化行阶梯形矩阵。

虽然很可能可以高效地构造这样的电路，但我们将稍微取巧，以一种更简单的方式解决这个问题。具体而言，我们不会像本节前文所述那样使用全部子空间，而只使用具有以下非常简单的简化行阶梯形的子空间（对于 $\mathbb{F}_{\mathfrak{q}}^{2d}$中的 k维子空间）：

$$A(M) \;=\; \begin{bmatrix} I_k & M \end{bmatrix}_{k\times 2d} \tag{3}$$

对于任意的 $M \in \mathbb{F}_{\mathfrak{q}}^{k\times 2d-k}$和 $k \in \{d-1, d+1\}$。当然，只需从输入中读出矩阵 M，就能很容易地给出到这种表示的比特映射（至少当 q 是 2 的幂时；从上述形式转换回位表示同样简单）。我们将把具有公式（3）中所示简化行阶梯形的子空间称为“有效”。很容易看出，几乎所有子空间都是有效的，因为有效子空间所占的比例，等于 $\mathbb{F}_{\mathfrak{q}}^{k\times 2d}$中的随机矩阵其初始 $k\times k$ 阶子式满秩的比例：

观察 3.13. 除去 $O(1/\mathfrak{q})$比例的子空间外，其余子空间均为有效子空间。

由于我们的路由方案具有单位拥塞，只要 $\mathfrak{q}$ 充分大于 d，我们就能确保在完整子空间系统上进行路由时，遇到无效子空间的概率可以忽略不计（事实上，我们已在上一节设置了参数，使这一点成立）。这意味着，我们对完整子空间码的分析可以直接移植到上述仅对有效子空间进行编码的情形。

超图邻接电路：有了上述转换，我们现在来回顾一下完整构造中实际需要实现哪些电路。我们讨论列表译码的情形；列表恢复的情形并无不同。经过 GI 复合后，给定一个布尔值电路$\mathcal{C}_w(v,i,j)$作为输入，该电路计算码字 w；它接收一个 $(d-1)$ 维子空间 $v\in V$，以及两个指定邻居 $s\sim v$ 的索引 i,j，随后再指定第二个邻居 $v'\sim s$。注意，码被视为具有由 v 索引的块，字母表则由 i,j 索引。当然，我们尚未定义特定 $v\in V$ 的第“i”个邻居是什么，也未定义反过来的对应关系 $s\in S$。我们将在下文通过类似于上述的简化行阶梯形变换，仔细定义这一点。

暂时把这一点放在一边，现在让我们回顾一下需要用上述电路实现的过程。给定一个目标 v，我们首先抽样一个随机邻居 $s_t \ni v$，然后调用路由预言机 $C_{\mathcal{O}}$（我们稍后将实现它）来计算一条由 $(d+1)$维子空间构成的路径 $s_1,\dots,s_t$。我们的算法通过选取若干随机 $v'\sim s_\ell$，并查询 $\mathcal{C}_w(v',i',\cdot)$ 的值，在每个 s_ℓ 上计算一个局部函数列表，其中 i' 是使得 s_ℓ 为 v' 的第 i' 个邻居的索引。最后，给定这些列表，我们的列表匹配过程抽取若干随机 $v''\in s_\ell\cap s_{\ell+1}$，并在数对 v'_1,v'_2（对应于译码后的列表）之间检查是否 $\mathcal{C}_w(v'_1,i'_1,j'_1)=\mathcal{C}_w(v'_2,i'_2,j'_2)$，其中 j'_z 是使得 v'' 为 v'_z 的第 i'_z 个邻居的第 j'_z 个邻居的索引（对于 $z\in\{1,2\}$）。在匹配过程结束时，我们在 s_t 处选取一个良好列表（由某个邻居 v_t 索引），并输出 $\mathcal{C}_w(v_t,i,j)$，其中 i 是 s_t 在 v_t 中的索引，而 j 是 v 的索引。

现在，撇开上述混乱的记号不谈，我们断言，只需在 (V,S) 的电路实现中实现五个“标准”电路即可。回顾一下，这些电路包括邻接电路和交集电路（采用某种尚待指定的邻居排序）：

1. $C_{V\to S}(v,i)$ 输出第 i 个邻居 $s\sim v$（矩阵形式）
2. $C_{S\to V}(s,\ i)$ 输出第 i 个邻居 $v\sim s$（矩阵形式）

3. $C_{S\cap S\to V}(s,\ s',\ i)$ 输出一个 $(d-1)$ 维子空间 $v\in s\cap s'$（矩阵形式），

以及两个“反向索引”电路：

1. $C^{-1}_{V\to S}(v,s)$ 输出索引 i，使得 $C_V(v,i)=s$2. $C^{-1}_{S\to V}(s,$ $v)$ 输出索引 i，使得 $C_S(s,i)=v$

上文中，我们将 v 和 s 视为以规范行最简阶梯形输入电路（特别是，我们只处理有效子空间），并将 i 视为一个有效邻居的位索引。注意，根据上文，我们可以轻松实现前面描述中用于 GI 复合的“(i,j)-索引”电路：给定 $v'\in V$、一个邻居 $s_\ell\sim v$ 和另一个 $v''\sim s_\ell$，只需运行 $C^{-1}_{V\to S}(v',s_\ell)$ 和 $C^{-1}_{S\to V}(s_\ell,v'')$，就能计算出要查询的 $\mathcal{C}_w(v',\cdot,\cdot)$ 的相应索引。换言之，一旦成功对 (V,S) 进行索引，便可立即对 (V,S) 上的两步游走（对应于完整的 GI编码）进行索引。

现在让我们实现电路（并在此过程中定义适当的邻域排序）。我们将从电路 $C_{S\to V}$ 和 $C^{-1}_{S\to V}$ 开始。

实现 $C_{S\to V}$ 和 $C^{-1}_{S\to V}$： 我们构建这些电路的策略基本上沿用了之前的规范化简行阶梯形方法的一个变体，但操作是在固定的有效（$d+1$）维子空间 s 内进行的。L

et A_s be the canonical RREF form of s. A subspace $v\le s$ is va$^{\text{成立当}}_{\text{且仅当}}$ and only if 可以将其写ite i成一个 $[A]_s(M)\cdot A_s$ 的乘积，其中

$$[A]_s(M)\ =\ \begin{bmatrix} I_{d-1} & M\end{bmatrix}_{d-1\times d+1}. \tag{4}$$

要看出这为何成立，注意一个方向上，形如 $[A]_s(M)\cdot A_s$ 的每个矩阵都是一个包含在 s 中的有效 $d-1$-子空间的不同简化行阶梯形。

对于另一个方向，每个有效的 $v\le s$ 都存在一个基 $x_1,\ x_2,\ \ldots,\ x_{d-1}$，将其投影到前 $d-1$ 个坐标后，得到的就是标准基。这意味着，x_i 等于 A_s 的第 i 行加上 d 和 $d+1$ 行的某个线性组合，而这些行属于 s 内投影的核。因此，逐行写出这些线性组合，就能得到一个矩阵 $[A]_s(M)$，使得 $[A]_s(M)A_s$ 是 v 的简化行阶梯形。

因此，电路 $C_{S\to V}$ 接收 s（如上所表示），以及索引 i，该索引实际上等于 $[A]_s$（M）中的 M，并输出 $[A]_s$（M）A_s。

我们注意到，该电路只覆盖 s 内的有效子空间，但与上面类似，s 中的无效子空间所占比例为 $O(\frac{1}{\mathfrak{q}})$。为了在我们的近似列表恢复框架中处理这一修改，我们只需将缺失的 $O(1/\mathfrak{q})$ 比例的子空间视为近似输入一致性 δ_{in} 的一部分。换言之，我们简单地假设这些条目与希望解码的全局 f 不一致（更正式地说，我们将它们视为始终返回某个固定符号，例如 0）。由于我们已经假设 $\delta_{in}>\Omega(\frac{1}{\mathfrak{q}})$，因此这至多使参数改变一个常数因子。

反过来，为了计算 $C^{-1}_{S\to V}$，给定一对 $v\subset s$ 的全局名称或规范表示，我们只需计算 v 关于 s 的一个基，并将其转换为该基下的简化行阶梯形，这样即可立即得到所需的索引。这些线性代数运算可以由规模为 poly$(d,\ \log\mathfrak{q})$、深度为 $O(\log^2(d)\log\mathfrak{q})$ [94] 的对数空间一致电路执行。我们指出，由于这些电路是均匀生成的，这也给出了所声称的 Time(polylog$(N)\cdot\log\frac{1}{\varepsilon}$) 局部编码。

实现 $C_{V\to S}$ 与 $C^{-1}_{V\to S}$： 现在，我们需要实现从 $(d-1)$ 维子空间到 $(d+1)$ 维子空间的“从左到右”邻接算子电路。幸运的是，我们的图通过将任一侧的子空间 w 映射到其“正交补” $w^\perp$ 而成为“自对偶”的。对于一个子空间 w，其“正交补”

$$w^\perp=\left\{x\in\mathbb{F}_{\mathfrak{q}}^{2d}:\forall y\in w,\langle x,y\rangle=0\right\}$$

其中 $\langle x,\ y\rangle$ 是标准双线性形式（尽管有限域中并没有正交的概念，我们仍称其为“正交补”）。

众所周知，k维子空间的对偶空间是一个 $2d-k$维子空间，并且对偶子空间的对偶空间就是原子空间。此外，当且仅当该对偶空间

相对于最后 $d \mp 1$个坐标是有效的。也就是说，我们可以将其写成如下形式矩阵的行空间

$$A'(M) = \begin{bmatrix} M & I_k \end{bmatrix}_{k \times 2d}$$

这是成立的，因为 $d \pm 1$-子空间有效，当且仅当 $w + E_{d\mp 1} = \mathbb{F}_q^{2d}$，其中 $E_{d\mp 1}$ 是由最后 $d \mp 1$ 个标准向量张成的子空间。当且仅当 $w^\perp \cap E_{d\mp 1}^\perp = \{0\}$时，这一条件成立。子空间 $E_{d\mp 1}^\perp$ 是由前 $d \pm 1$ 个标准向量张成的子空间，因此当且仅当 $w^\perp + E_{d\mp 1}^\perp = \mathbb{F}_q^{2d}$，也当且仅当 $w^\perp$ 相对于最后 $d \mp 1$ 个坐标有效。

因此，要实现 $C_{V\to S}$，我们只需计算正交补空间，求出其相对于最后几个坐标的规范行最简阶梯形，使用 $C_{S\to V}$（相对于最后几个坐标），并求出 $C_{S\to V}$输出的子空间的对偶空间。对于 $C_{V\to S}^{-1}$，我们也进行类似的操作。

计算这些只需计算输入的正交补（并运行上述电路），因此仍可在对数空间一致的规模 poly($d, \log q$)、深度 $O(\log^2(d) \log q)$ 内完成。我们注意到，这（当然）也有与上文相同的局限：它无法编码 v 的邻居中占 $O(\frac{1}{q})$ 比例的一部分，而在我们的算法中，可以将其视为 δ_{in} 误差的一部分。

实现 $C_{S\cap S\to V}$： 回忆一下，$C_{S\cap S\to V}(s, s', i)$ 接收两个处于规范简化行最简阶梯形的矩阵 $\mathbb{F}_q^{(d+1)\times 2d}$，并必须输出一个随机的有效（$d-1$）维子空间 $v \in s \cap s'$ 的简化行阶梯形。不幸的是，两个有效的（$d+1$）维子空间 $s \cap s'$ 的交集可能不包含任何有效的（$d-1$）维子空间 v。我们实际实现的电路会取 $s \cap s'$ 的交集并计算其简化行阶梯形；如果它是一个有效的 d 维子空间，则以 $C_{S\to V}$ 寻找有效子空间的相同方式，在其中找到一个有效子空间（当然，该交集是 d 维的，但除了将 $d+1$ 改为 d 之外，电路的运作方式相同）。我们观察到，只有 $O(\frac{1}{q})$ 条边 $\{s, s'\}$ 的交集无效，因此该电路在占边总数 $O(\frac{1}{q})$ 比例的一部分上失败。对于其余边，它只会映射到所有可能子空间中占 $1 - O(\frac{1}{q})$ 比例的一部分。

这对我们的目的而言已经足够，因为即使我们在额外的 $O(\frac{1}{q})$-比例的边上失败，对该算法的分析基本上仍保持不变；同样地，即使对于未失败的边，我们只覆盖可能子空间中的 $1 - O(\frac{1}{q})$-比例，这也基本上只会在每个传播步骤中额外引入 $O(\delta_{in})$-误差，而我们可以将其吸收。

最后，标准的并行线性代数允许我们在给定 s 和 s' 的矩阵表示的情况下，使用规模为 poly(d,log q)、深度为 $O(\log^2(d) \log q)$ 的对数空间一致电路计算出这样的形式，正如所需。

路由电路 最后，我们需要实现路由电路。回忆一下，我们的路由电路将接收两个 $d+1$ 维子空间 s 和 s' 的简化行阶梯形，其中 $\dim(s \cap s') = k$，并通过以下方式进行路由：

1. 从 $s \cap s'$中随机选取一个基 $x_{d-1-k}, \ldots, x_{d-1}$
2. 独立地将其扩展到 s 和 s'的随机基，($y_1, \ldots, y_{d-2-k}$) 和 ($z_1, z_2, \ldots, z_{d-2-k}$).
3. 对于每个 $i \in [d-k-2]$ 输出子空间

$$v_i = span(z_1, z_2, \ldots, z_i, y_{i+1}, y_{i+2}, \ldots, y_{d-2-1}) + (v \cap v').$$

有了上述所有工具，实现这个路由过程其实相当容易。我们已有 s 和 s' 的简化行阶梯形形式，并且可以如前所述计算出 $s \cap s'$ 的矩阵形式。接下来，只需抽取适当数量的行的随机线性组合，就能得到 $s \cap s'$ 的随机基（及相应的扩展）。[26] 生成这些基向量后，我们便可直接输出上文定义的每个 v_i 的简化行阶梯形表示。同样，所有过程都可以由规模为 poly(d, log q)、深度为 $O(\log^2(d) \log q)$ 的对数空间一致电路完成。最后，如前所述，这个过程有可能输出不属于我们编码的 v_i（即不具有相应简化行阶梯形形式），但由于单位拥塞，这种情况发生的概率至多为 $O(d/q)$，因此可以像处理上述坏事件那样忽略。

[26]同样，这仍有小概率无法得到满秩系统，但这种情况发生的概率至多为 $O(1/q)$，而且只需重复几次便可轻松放大成功率，因此可以放心忽略。

定理 3.1 的电路复杂度 最后，我们可以把所有内容整合起来，证明所声称的电路复杂度界。根据引理 3.5，定理 3.4 至多需要 poly(t, $\frac{1}{\varepsilon}$, δ^{-1}) 次对路由预言机电路实现和超图实现的并行调用（注意，ℓ_{in} 设为 $O(\log_{\varepsilon} t)$），此外还至多需要 poly(t, $\frac{1}{\varepsilon}$, δ^{-1}) 次对 GI 图的并行调用，以模拟输入列表。因此，总规模至多为 poly(t, ℓ_{in}, δ^{-1}) 乘以这些电路实现中的最大规模；它们的规模均为 poly(log(N), log q)，而深度至多为这些实现的最坏情况深度，即 $O(\log(\log_{\varepsilon} N)\log^2(\log(N)))$，如所声称。 ——

4 多对数速率 aLLDC

在本节中，我们将利用上一节中开发的工具，并假设存在具有单位拥塞路由的稀疏超图系统（V、S、G），正式陈述并证明我们的多对数码率构造。我们的主要结果是如下具有多对数码率的（大字母表）aLLDC：

定理 4.1（多对数率 aLLDC）。存在一个普适常数 $C > 0$，使得对于所有足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，都存在 $N' \geq N$ 以及一个强显式直积码 $\mathcal{C}$：$\Sigma_1^N \to \Sigma_2^{N'}$，它是一个$\left(C^{-1}\text{对数}^{10} N,\ 0.01\right)$-近似的 $(\varepsilon,\ \ell_{out},\ q)$-局部列表可解码码，且

1. 码率：$R \geq \text{poly}\left(1/\text{对数}(N)\ \varepsilon\right)$
2. 查询复杂度：$q \leq \text{对数}^5(N)\cdot\varepsilon^{-}\text{poly}\left(\text{对数 对数 } N,\ \text{对数 } \frac{1}{\varepsilon}\right)$
3. 列表大小：$\ell_{out} \leq O\left(\frac{1}{\varepsilon}\right)$

此外，$\mathcal{C}$ 可以在 $\text{Time}(\text{polylog}(N)\log\frac{1}{\varepsilon})$ 内进行局部编码，并可在对数空间一致的规模 $\text{poly}(\frac{1}{\varepsilon}, \log N)$ 和深度 $O\left(\log(\frac{\log N}{\varepsilon})\cdot\log\log N\right)$ 的电路中解码。

我们指出，如果只关心信息论查询界，那么只需使用完全随机路径（通过暴力搜索计算），而非路由，就可以将上述结果改进为 $q \leq \tilde{O}(\frac{\log N}{\varepsilon})$。这与我们在第 7 节中针对此类码给出的查询下界相匹配。

定理 4.1 还满足一种稍强的高噪声 ℓ_{in}-输入、$(\delta_{in}, \delta_{out})$-近似、$(\varepsilon, \ell_{out}, q)$-列表恢复 保证：将上述参数中的 ε 替换为 $\frac{\varepsilon}{\ell_{in}}$ 即可。[27] 通过附录 A 中给出的改进级联方法，可得到以下二元近似局部可解码码：

推论 4.2 (多对数速率二元 aLLDC)。对于所有足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个强显式二元 $\mathcal{C}$：$\{0,1\}^N \to \{0,1\}^{N'}$，它是一个 (0.01)-近似、$(\frac{1}{2}+\varepsilon, \ell_{out}, q)$-局部列表可解码码，且

1. 码率：$R \geq \text{poly}(1/\log(N)\ \varepsilon)$
2. 查询复杂度：$q \leq \text{poly}(\log(N)\ \frac{1}{\varepsilon})$
3. 列表大小：$\ell_{out} \leq \text{polylog}(\frac{1}{\varepsilon})\varepsilon^{-2}$

此外，$\mathcal{C}$ 可以在时间 $\left(\text{poly}(\log(N)\ \frac{1}{\varepsilon})\right)$ 内进行局部编码，并以对数空间一致的规模进行局部解码 $\text{poly}(\frac{1}{\varepsilon}, \log N)$，深度为 $\tilde{O}\left(\log^2\left(\log\frac{N}{\varepsilon}\right)\right)$。[28] ——

如前所述，在某些情况下，poly($\frac{1}{\varepsilon}$) 的编码时间是不够的。在附录 A 中，我们还给出了一种变体，其编码时间为 poly(log N) $\cdot\log^2\frac{1}{\varepsilon}$，查询复杂度为 $\tilde{O}(\frac{1}{\varepsilon^2})$（对于二元 aLLDC 而言，这是可能达到的最佳复杂度 [95]）。其代价是码率会随着 ε 以准多项式级下降（这种权衡

[27] 这是一种通用的转换，只需取输入列表 $\mathcal{L}$，并通过从每个列表中随机选取一个元素来构造一个码字 w 即可。只要该码的码长（编码符号的数量）至少为 $\Omega(\frac{1}{\varepsilon})$，任何在 $\mathcal{L}$ 中以 2ε 计算的码字，在所得的 w 中以 $\frac{\varepsilon}{\ell_{in}}$ 计算的概率就会极高；此时我们便可以运行 aLLDC。细节从略。[28] 这里 $\tilde{O}$ 隐去了 $\log\log(\frac{\log N}{\varepsilon})$ 因子。

是标准的，源于与一个小型完备复形进行级联；进一步讨论见下文推论 A.2）。

为了证明定理 4.1，我们需要一个具有第 3 节所述性质、但度仅为多对数级的超图系统 $(V,\ S,\ G)$。粗略地说，这些系统是通过将我们在相邻子空间上的直积覆盖，替换为 3-部高维扩展图的邻域上的直积构造而成的。[29] 我们将具体构造推迟到第 6 节，在那里我们还会证明某些此类族具有高度高效的局部路由方案。现在，我们先正式陈述所得超图系统的性质，这些性质将用于我们的码。我们将其分为两部分：存在性和电路实现。

命题 4.3 (稀疏超图系统)。 对于足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个正则超图系统 (V, S, G)，具有以下性质：

1. $|V| \in [N, N$ 多项式(log $N, \varepsilon^{-1})$]*2.* (V, S) 是双正则的，左度和右度至多为 多项式(log $N, \varepsilon^{-1})$*3.* G 具有长度为 $t \le O(\log^3(N))$ 的单位拥塞路由方案 $\mathcal{O}$*4.* 对于每个 $s \in S,\ G_s$ 都是一个 $\varepsilon_{15^3 \cdot 10^{11} t^4 \log t}$-采样器

5.（反向）包含图 (S, V) 是一个 $\frac{10^{-5}}{4t\varepsilon^2}$-采样器。

请注意，上述看似复杂的参数集之所以如此设定，是为了获得具有 $\delta_{out} = 0.01$ 码率、且输入列表大小由 Guruswami-Indyk 归约生成的码，这些码来自定理 3.4。

断言 4.4 (电路实现，非正式表述)。 (V, S, G) 以及命题 *4.3* 中所述的 $\mathcal{O}$，都存在对数空间一致的电路实现，其深度为 $O(\log\log(N)\log(\log_{\varepsilon^N}))$，规模为 $\mathrm{poly}(\frac{1}{\varepsilon}, \log N)$。

严格来说，断言 4.4 仅在非正式表述的意义上是非正式的：与我们的子空间电路实现类似，我们构造的电路不会恰好 编码命题 4.3 中的系统 (V, S, G)，而是会带有极其微小的误差（具体而言，我们给出的电路对应于一个略大的超图 (V', S')，其中包含比例为 $\ll \mathrm{poly}(\varepsilon)$ 的额外“退化”节点）。与子空间构造一样，这种误差足够小，不会对证明产生任何实质影响。第 6 节将对此作进一步讨论。

现在我们可以证明定理 4.1 了。证明整体上沿用定理 3.1 的思路，只需用命题 4.3 替换子空间系统。

*定理 4.1 的证明。*我们首先描述该码，其框架与定理 3.1 完全相同，只是在每一步用命题 4.3 替换子空间超图系统。

令 N 为目标消息长度，并令 N' 为大于 N 且满足以下条件的最小整数：存在一个来自命题 4.3 的超图系统 $(V,\ S,\ G)$，使得 $|V| = N'$。已知 $N' \in [N, N\mathrm{poly}(\log(N)\ \frac{1}{\varepsilon})]$。与之前一样，引理 E.2 表明，只需在 N'个消息比特上构造相应的 aLLDC，再通过（截断的）重复编码将原消息编码到该空间中。

现在，我们在 N' 消息比特上的码将是（Guruswami-Indyk）对下列各项的复合：

1. 命题 4.3 中对应于 (V, S, G) 的列表可恢复直接积码 $C_{(V,S)}$，以及
2. 命题 4.3 所保证的采样器（包含）图 $G' = (S, V)$

为了证明这些对象的 GI 复合所声称的编码参数，我们先正式回顾 $C_{(V,S)}$ 在我们的参数设置下的列表恢复保证。令 $t \le O(\log^3(N))$ 为命题 4.3 中承诺的路由方案长度。固定 $\delta = \frac{1}{1500t}$ 和 $\ell_{in} = {}^{100}\log_{\varepsilon}\ {}^{t}$，我们在命题 4.3 中设置的参数恰好使得每个交图 G_s 都是一个 $\frac{\delta^2}{10^3 t}$采样器。因此，根据定理 3.4，$C_{(V,S)}$ 是一个 ℓ_{in} 输入、(δ,0.01)-近似的 $(1 - \frac{10^{-5}}{t}, \ell_{in}, t)$-LLRC。

[29] 熟悉高维扩展体的读者可能会注意到，先前的构造在某种意义上已经是某个高维扩展图上的邻域编码，尽管该扩展图（旗标复形）并不非常稀疏。我们用于一般三部 HDX 的邻域编码则略有不同，因为第 3 节中的简化版本依赖于子空间的特殊性质。

现在可以将上述码输入到带有我们的采样器 G' 的 Guruswami-Indyk 归约中。这样还不能完全达到所需的参数（也就是说，列表大小会过大），但我们之后会处理这一点。命题 4.3 承诺反向包含图 $G' = (S, V)$ 是一个 $\frac{10^{-5}}{4t\varepsilon^2}$ 采样器，因此引理 3.7 立即推出，带有 G' 的复合码是一个 $(\frac{1}{1500^2 t^2}, 0.01)$- 近似的 $(\varepsilon, \frac{200}{\log_\varepsilon t}, \frac{100t}{\log_\varepsilon t})$-LLDC。代入 $t = O(\log^3(N))$ 即可得到所声称的参数，列表大小除外；我们将在下文单独讨论这一点。

不过，在此之前，我们先确认该构造的其余参数。消息长度为N。码长为 $|V| = N' \le N\text{poly}(\log N, \varepsilon^{-1})$。（对数）字母表大小等于 (V, S) 的左度与右度之积；根据命题 4.3，该值也保证至多为 $\text{poly}(\log N, \varepsilon^{-1})$。综合起来，得到码率 $\text{poly}(\frac{1}{\log N}, \varepsilon)$，如所声称。

最后，电路复杂度分析与定理 3.1 中的分析相同，只需使用断言 4.4 所保证的路由电路和超图实现电路，并对相关参数进行相应调整。

改进列表大小： 还需论证，上述 GI 复合实际上具有列表大小 $O(\frac{1}{\varepsilon})$，而非如上所声称的$\frac{1}{200\log_\varepsilon t}$；后者会随着 N 增大，尽管增长得较慢。

为此，我们需要定理 3.4 中列表恢复结论的一个略强变体。回顾一下，定理 3.4 证明：对于任意输入列表 $\mathcal{L}$ 和 $f : V \to \Sigma_1$，如果有比例为 $(1 - \frac{10^{-5}}{t})$ 的 $s \in S$ 存在函数 $g \in \mathcal{L}_s$，使得该函数能以 δ 计算 f，那么图 $2^{15t\delta}$ 中的解码器 $\text{Dec}(s, g, \cdot)$ 就能计算 f，只要

1. s 在公式 (1) 的意义下是“良好”的
2. g 与 f 的距离至多为 δ。

由于我们证明了 99% 的 s 在此意义下是良好的（特别是其中至少包含一个这样的良好起始g），这意味着我们只需随机选取一个 s，并对每个^{-}g 输出电路列表 $\text{Dec}(s, g, \cdot)$，就能以 $\frac{3}{4}$ 的概率解码 f

不同于上述情形，不妨设我们现在处于这样一种情况：随机的 s 不仅在上述意义下有 99% 的概率是“良好”的，而且还 保证其列表中有 $\Omega(\varepsilon)$比例 的元素能够 δ-计算 f。由于我们已在定理 3.4 中证明，从任意这样的 $g \in \mathcal{L}_s$ 出发，都会得到一个能够 $15t\delta$-计算 f 的电路，因此无需为列表中的每个 g 都输出一个电路。特别地，根据切尔诺夫界和并集界，随机选择 s，然后对 $O(\frac{1}{\varepsilon})$ 个随机局部函数 $g \in \mathcal{L}_s$ 输出 $\text{Dec}(s, g, \cdot)$，就足以输出某个能够以 98% 的概率 $15t\delta$-计算 f 的 $\text{Dec}(s, g, \cdot)$。最后请注意，如果我们得到保证：输入字 $\mathcal{L}$ 有 $1 - \frac{10^{-5}}{t}$ 比例的 s，其列表在 $\Omega(\varepsilon)$ 的时间内 δ-计算 f，那么完全相同的证明意味着，随机的 s 有 99% 的概率是“良好”的，且拥有这样的列表。

有了上述讨论作为背景，让我们重新关注 Guruswami-Indyk 复合码。根据上述讨论，只需说明，对于任意 $f \in \mathcal{B}_{1-\varepsilon,\ \delta^2}(\mathcal{L})$，我们在归约中向 $C_{(V,S)}$ 输出一个模拟列表，使得以 99% 的概率，有 $(1 - \frac{10^{-5}}{t})$ 比例的 $s \in S$ 所持列表能够 δ-计算 f，且至少对 $\Omega(\varepsilon)$比例的局部函数成立。值得庆幸的是，这可以直接由引理 3.7 的证明推出。特别是，只需注意，在方程 (2) 下应用的切尔诺夫界立即推出了更强的结论：任意 $s \notin B$ 抽取的样本中，都有 $\Omega(\varepsilon)$ 比例的邻居能够以至少 $\frac{10^{-6}}{2t}$ 的概率 δ-计算 f。回溯马尔可夫界可知，确实有 $1 - \frac{10^{-5}}{t}$ 比例的模拟列表 $\mathcal{L}_s$ 被传递给 $C_{(V,S)}$，并具有所需性质，因此得证。 □

5 常数率 aLLDC

本节介绍我们第二个主要结果——常数码率 aLLDC 的构造与证明。我们首先陈述该结果的“大字母表”版本。

定理 5.1 (常数率 aLLDC)。对于每个 $\varepsilon > 0$，以及无穷多个 $N \in \mathbb{N}$，都存在一个$(\exp(-O(\frac{1}{\varepsilon})), 0.1)$-近似的$(\varepsilon, O(\frac{1}{\varepsilon}), q)$-局部列表可译码码 $\mathcal{C} : \Sigma_1^N \to \Sigma_2^{N'}$，其满足

1. 码率：$R \ge \exp(-O(1/\varepsilon))$
2. 查询复杂度：$q \le \text{poly}(\log(N))\ \exp(O(\frac{1}{\varepsilon}))$

3. 对数字母表与局部性： $r \le \log(N)^{\exp(O(\frac{1}{\varepsilon}))}$

该码可在 $\mathrm{Time}(\mathrm{poly}(\log(N), \exp(\frac{1}{\varepsilon})))$ 内进行局部编码，并可在对数空间一致规模 $(\log N)^{O(\exp(\frac{1}{\varepsilon}))}$ 内完成解码，且深度 $(\tilde{O}(\log\log N \exp(O(\frac{1}{\varepsilon})))$。

当 ε 是一个较小的常数时，上述结果给出一个 $(10^{-10}, 0.1)$-近似 $(\varepsilon, O(\frac{1}{\varepsilon}), \mathrm{polylog}(N))$-局部列表可译码码，具有常数码率，且局部性为 $\mathrm{poly}\log(N)$。同样，在高噪声情形下，可以轻松修改该码，使其具有 ℓ_{in}-列表可恢复性：将上述内容中每个 ε 都替换为 $\frac{\varepsilon}{\ell_{in}}$。将上述构造与任意足够好的二元列表可译码码（例如级联 Reed-Solomon 码[92]）级联，即可得到以下推论。

推论 5.2 (二元常数码率 aLLDC)。对于每个 $\varepsilon > 0$ 以及无穷多个 $N \in \mathbb{N}$，存在一个 $(O(\varepsilon)\ 0.1)$-近似、$(\frac{1}{2} + \varepsilon, O(\frac{1}{\varepsilon^2})\ q)$-局部列表可解码码 $\mathcal{C}$： $\Sigma_1^N \to \Sigma_2^{N'}$ 具有

1. 码率： $R \ge \exp(-O(1/\varepsilon^3))$

2. 局部性： $r \le \log(N)^{\exp(-O(1/\varepsilon^3))}$ *3.* 查询复杂度： $q \le \log(N)^{\exp(O(\frac{1}{\varepsilon^3}))}$

该码可以在 $\mathrm{TIME}(\log(N)^{O(\exp(\frac{1}{\varepsilon^3}))})$ 内进行局部编码，并且可以在对数空间一致电路规模 $(\log N)^{O(\exp(\frac{1}{\varepsilon^3}))}$ 和深度为 $\tilde{O}(\log^2(\log N)\exp(O(\frac{1}{\varepsilon^3})))$ 的情况下完成解码。

我们注意到，上述两个码实际上都可以在对数空间一致电路规模为 $\log(N)^{O(\exp(\frac{1}{\varepsilon^3}))}$ 且深度为多对数 $\log(N)\exp(O(1/\varepsilon^3))$ 的电路中进行局部编码。最后，尽管这些定理是针对常数 $\varepsilon > 0$ 陈述的，但也可以取略微次常数的 ε，例如 $\varepsilon = \frac{1}{\mathrm{polylog}\log N}$。

和之前一样，我们首先概述该构造；与之前的框架相比，它需要进行若干重大改变：

构造概述 在本概述中，我们将把参数 $\varepsilon, \delta_{in}, \delta_{out}$ 等视为常数，与集合系统的规模无关。

回顾第 3 节和第 4 节，我们的论证在很大程度上依赖于超图系统 (V, S, G) 的交图 G_s 是 $\frac{1}{\text{多对数}(N)}$ 采样器。遗憾的是，我们不知道如何构造这样的系统，同时还要求 (V, S) 是有界度的，也就是说，每个顶点至多属于 $O(1)$ 条超边。事实上请注意，$\frac{1}{\text{多对数}(N)}$-采样要求 S 中的集合规模至少为多对数 (N)，因此，如果这样的系统确实存在，它们就必须极度不平衡，且 $|S| \ll |V|$。虽然我们尚不知道如何从形式上排除这类系统，但这种类型的不平衡对象向来很难构造。

相反，我们从 [60] 的全局列表译码算法中汲取灵感来解决这个问题：具体而言，我们将在系统中添加第三层 U，由 V 的子集构成，使得任意 $u \in U$ 的“局部视图”（更准确地说，超图 $(u, \{s \cap u\}_{s \in S})$）本身都是一个良好的直积码（我们将其称为 (V, S, U, G) 的“内码”）。粗略地说，我们需要 (V, S, U, G) 具备三个关键性质：

1. 图 (V, S) 具有<运行方向上的>常数度（因此直积码 $C_{(V,S)}$ 具有常数码率）。
2. 给定任意输入码字 w 到 (V, S) 以及 $f \in \mathcal{B}_{1-\varepsilon, \delta_{in}}(w)$，函数 f 位于除一个次常数比例的内码 $u \in U$ 之外的所有内码的球中。
3. 超图系统 (V, U, G)（我们称之为“外码”）具有一个适用于次常数噪声列表恢复区域的传播译码器（特别地，我们将取 (V, U) 为多对数 (N)-度，以避免原有问题）。

假设我们有一个满足这些要求的系统。那么，位于 $u \in U$ 处的每个“内码”都可以解码一个局部函数列表 $\mathcal{I}_u$，这些函数为 u 中的顶点标记。根据第二个性质，我们可以保证，除比例为 $\frac{1}{\text{多对数}(N)}$ 的这些译码后的列表外，其余列表都包含一个与 f 相差 δ' 的函数；换句话说，即 $f \in \mathcal{B}_{\frac{1}{\text{多对数}(N)}\delta'}(\mathcal{I})$。但这恰好是 (V, U) 上的低噪声列表恢复设置，因此我们可以在 (V, U) 上运行解码器，按要求局部解码 f；为此，我们通过调用内部解码器来查询这些列表 $\mathcal{I}_u$。

构造这样的集合系统需要非平凡的高维扩张器机制。（我们指出，这些机制无法从 [60] 的“双重采样器”中得到，因为双重采样器无法同时满足前两个性质。）

即便先不论构造此类系统的困难，我们还掩盖了一个关键问题：局部码中的每个顶点仅被编码到常数个集合中（这是我们实现全局常数码率所关键依赖的性质），因此我们的解码准确率 δ' 也必须是常数（这实际上是固有的，因为可以迫使这些码中常数比例的顶点只能看到噪声）。这对我们原来的列表恢复方案当然是一个大问题，该方案最终只能对$t =$ 多对数(N)达到准确率 $\delta_{out} \geq t\delta'$ 。

为了处理这一问题，我们在本节提出一种面向超图系统的新列表恢复方法，其输出准确率为 $\delta_{out} = \delta_{in}$ exp$(O(\frac{1}{\varepsilon}))$，从而使我们能够在常数码率区域中实例化外层解码。该技术依赖于两个关键的新想法。第一个想法是对 [60],中“列表剪枝”方法的改编，旨在确保输入到我们列表恢复算法的列表是良好分离的。粗略地说，分离使我们能够在沿解码图行走时唯一地 匹配列表，从而避免误差的传播（只要我们避开由采样误差产生的一小集合坏边）。当然，这多少有些一厢情愿；我们无法归结到所有列表都良好分离的情形。然而，关键的是，我们能够 确保（以付出exp$(O(\frac{1}{\varepsilon}))$的代价）我们系统中很大的常数比例 （例如99%）的集合具有这种无损传播，并且该集合对算法是可识别的。

现在，朴素地看，我们仍然面临一个问题：由于我们的路由方案采用长度为多对数(N)的路径（而且事实上我们要求这些路径良好地覆盖该图），一条随机路径极有可能逸出上文所述的这个“良好”集合 T，从而破坏我们的误差。这就引出了第二个关键思想：子集内部（“顶点容错”）路由。具体而言，我们将证明可以构造局部路由的一个变体：给定对定义域 T任意较大常数比例部分的预言机访问，它能在几乎所有 u, $u' \in T$之间以低拥塞进行局部路由，同时完全停留在 T内部。结合无损传播，即使原始列表具有常数误差，这也足以保证准确解码，并且（连同超图系统本身的存在）是实现常数码率的关键。

本节路线图 下面我们简要概述本节的结构。我们首先在第 5.1 节中形式化超图三元组的设定，以及子集内部路由方案这一新概念。在第5.2节中，我们形式化将要使用的超图三元组的性质。在第5.3节中，我们利用该三元组的内部码结构，给出到低噪声列表恢复的归约。在第5.4节中，我们引入良好分离条件，并在假设可访问子集内部路由方案且输入列表良好分离的前提下，给出一个列表恢复算法。最后，在第5.5节中，我们说明如何从一般的列表恢复归约到良好分离情形。

5.1 形式化设置

正如我们在概述中提到的，我们要求一个具有三层集合的集合系统，这在某种程度上类似于 [60] 中的“双重采样器”，但所需的保证大不相同。令 $(V,\ S,\ U,\ G)$ 为一个集合系统，其中 V 是顶点，$S \subset \binom{V}{r_1}$、$U \subset \binom{V}{r_2}$ 由集合组成。（注意，对于 $r_1,\ r_2$ 没有任何要求。）图 $G = (U,\ E)$ 是一个解码图，其定义方式类似于第 2.2 节中的定义，只不过这里是相对于 U 而定义的。它是一个正则图，其顶点为 U，并且对于该图，当 $u \sim u'$ 时有 $u \cap u' \neq \emptyset$。[30] 交图 G_u 的定义也类似：一侧的顶点是 u 的顶点，另一侧则放置交集 $\{u \cap u' :\ \{u, u'\} \in E\}$。

除了上述内容（它们都类似于我们在基本超图系统中分析过的图）之外，我们还需要几个由 $(V、\ S、\ U、\ G)$ 产生的“新”局部图和全局图。在全局方面，这样的新图只有一个：$(S、\ U)$ 集合层图。这是一个二部图，其顶点一侧为 S，另一侧为 U，其边为所有满足 $s \cap u \neq \emptyset$ 的 $\{s, u\}$。

现在我们转而介绍局部集合系统的定义。将具有 $V_u = u$ 和 $S_u = \{s \cap u$ 的内部集合系统记为 $In_u = (V_u,\ S_u)$: $s \in S,\ s \cap u \neq \emptyset\}$。将 In_u 的局部解码图 记为 G_{In_u}。

[30] 与第 2.2 节中的解码图一样，并非所有重叠的集合之间都有边。图本身将根据我们的系统实例而定。

也就是说，该图中的顶点是 S_u，并且在这种情况下，与上文不同，我们将连接满足 $s \cap s' \neq \emptyset$ 的所有 $s \sim s'$。[31]

如上文所述，我们的构造将仅像之前一样对 (V, S) 取直积。额外的集合层 U 仅用于解码。给定一个从 w 到 $\mathcal{C}_{(V,S)}$的输入字 w，我们以自然方式记在 S_u 上诱导出的局部字为 w_{In_u}。也就是说，对于 S_u 中的每个 $s' = s \cap u$，我们取 $(w_{In_u})_{s'} = w_s|_{s'}$。当然，通过查询 s处的原始字，我们也能够查询字 s'。

类似地，我们也以相同方式定义 $In_s = (V_s, U_s)$ 的集合系统。顶点 $V_s = s$以及这些集合全都是 $U_s = \{u \cap s : u \in U\}$。我们注意到，该集合系统对于解码本身并非必要，但其采样性质对于分析将十分重要。

5.1.1 子集内部路由

如概述中所述，为了使我们的外码在常数码率情形下发挥作用，我们需要在 U 的解码器图上采用一种强化的路由概念，该路由仅通过 $u \in U$，其中的列表具有概述中提到的距离性质（我们将在下文不久将其形式化）。在此，我们将其视为图 G 的任意子集 T（实际上，在该论证中我们无法控制这个子集）。

更形式化地说，我们现在引入"子集内部"路由的概念。它是一种低拥塞路由方案，在图 $G = (U, E)$ 中为顶点 v、v' 之间建立路径，使得 v 和 v' 之间的所有中间顶点都属于某个规模为常数的集合 $T \subseteq U$。由于 T 可以是任意的，因此按此表述，这项任务当然可能无法完成。例如，某个顶点 v 在 T 中可能没有邻居，因此若不离开 T，就无法到达任何 $v' \neq v$。尽管如此，我们将证明，在某些高维扩展体上，只要允许路由方案在选择端点 v、v' 时以 $o(1)$ 的概率失败，并稍微放宽用于避免坏事件的低拥塞概念，这样的方案仍然可行：

定义 *5.3*（子集内部路由）。设 $G = (U, E)$ 是一个图。子集内部路由方案 $\mathcal{O}^T(u, u')$ 获得某个规模为 $\mathbb{P}[T] \geq 0.9$ 的集合 $T \subseteq U$ 的成员资格预言机，并输出 $FAIL$ 或一条路径 $P = (u = u_0, u_1, u_2, \ldots, u_m = u')$，使得 $u_1, u_2, \ldots, u_{m-1} \in T$。若 $\mathcal{O}$ 输出 $FAIL$ 的概率（对 $u, u' \in U$ 进行均匀选择）为 $\leq \frac{1}{\log^4 |U|}$，则称 $\mathcal{O}$ 成功。$\mathcal{O}$ 的长度是 $\mathcal{O}$ 的支撑集中路径的最大长度。若对每个子集 $B \subset E$，$\mathcal{O}$ 命中 B 的概率至多为……，则称 $\mathcal{O}$ 具有低拥塞。

$$\Pr_{u,u',\mathcal{O}}[\mathcal{O}(u,u') \neq FAIL \wedge \exists e \in \mathcal{O}(u,u') : e \in B] \leq 4t \Pr[B]^{1/4} + \frac{1}{\log^4 |U|}$$

我们指出，要求 $\mathbb{P}[T] \geq 0.9$ 主要是出于方便。我们将在满足 $U = L \cup R$的二部图上构造此类方案，而实际所需的条件是 $\mathbb{P}[T \cap L]$，$\mathbb{P}[T \cap R]$ 都不可忽略。同样，常数 4，$1/4$ 和 $\frac{1}{\log^4 |V|}$ 是任意选取的，并为简洁起见固定为我们方案所达到的参数。 ——

5.1.2 （V，S，U，G）的电路实现

现在我们有了一个三部分系统，需要与之对应的电路实现（V，S，U，G），类似于第 2.2 节中仅针对（V，S，G）定义的电路实现。

对于外层解码器，我们需要第 2.2 节中定义的集合系统（V，U）的电路实现，也就是（$C_{V \to U}$，$C_{U \to V}$，$C^{-1}_{V \to U}$，$C^{-1}_{U \to V}$，$C_{U \cap U \to V}$）。

此外，我们还需要针对每个 $u \in U$ 的 In_u 电路实现。我们仍然只使用 $v \in u$ 或 $s \in S_u$ 的全局名称，而不使用它们在 u 中作为第 i 个邻居的标记。我们需要以下电路。

1. 一个电路 $C_{U \to S}(u, i)$，其输出为 u 在 S 中的第 i 个邻居（即第 i 个集合 $s \in S_u$）。

2. 一个电路 $C_{U;S \to V}(u、s、i)$，接收与 u 相交的 $u \in U$、$s \in S$ 以及索引 i 作为输入。它输出 $s' = s \cap u$ 的第 i 个邻居。

[31]原则上可以更一般地定义这些内部图，但在我们构造的系统中，连接所有重叠集合总能得到所需的内部解码器，因此为求简洁，我们将定义固定为这种形式。

3. 一个电路 $C_{U;V\to S}(u、\ v、\ i)$，接收 $u\in U、\ v\in u$ 以及索引 i 作为输入。它输出 v 在 S_u 中的第 i 个邻居。4. 一个电路 $C_{U;S\cap S\to V}(u、\ s、\ s'、\ i)$，当 $u\cap s\cap s'=\emptyset$ 时输出 $u\cap s\cap s'$ 的第 i 个邻居，或输出 $FAIL$。5. 电路 C_U^{-1}；$S\to V$、C_U^{-1}；$V\to S$ 分别输出 i，使得 $C_{U;S\to V}(u、\ s、\ i)=v$，或 $C_{U;V\to S}(u、\ v、\ i)=s$。6. 一个电路 $\mathcal{C}_{\mathcal{O}}$，实现了解码图 G 的子集内部路由方案。

5.2 集合系统

在下面的命题中，我们指出所需集合系统具备的性质。该命题将在第 6 节中证明。参数经过设置，使得我们可以将该系统直接代入下文的解码框架。

命题 5.4。令 $N>0$ 为足够大的整数，并令 $\varepsilon>{}^{1}\log\log N$.[32] 设 $\delta_{in}=\exp(-3\cdot 10^3\frac{1}{\varepsilon})$,$\eta=(\frac{\varepsilon\delta_{in}}{8})^7$ 以及 $\beta=(\frac{\varepsilon\delta_{in}^2}{\log N})^{28}$。存在一个集合系统 (V,S,U,G)，满足以下性质。

1. $|V|\in[N,N\log(N)^{\exp(O(\frac{1}{\varepsilon}))}]$. *2.* 每个 $s\in S$ 和 $u\in U$ 的规模均为 $\le\log(N)^{\exp(O(\frac{1}{\varepsilon}))}$. *3.* 每个顶点都出现在 $\exp(O(\frac{1}{\varepsilon}))$ 个集合中。*4.* 包含图 (V,S)、(V,U)，集合层图 (S,U)，局部图 (V_s,U_s)、(V_u,S_u) 以及对每个 $s\in S$ 和 $u\in U$ 而言的交图 G_u，都是双正则图。*5.* 交图 G_u 是一个 β-采样器。*6.* 图 (S,U) 是一个 β-采样器。*7.* 对每个 $u\in U$，(V_u,S_u) 都是一个 η-采样器。*8.* 对每个 $u\in U$，局部解码图图 G_{In_u} 都是 $(1-O(\eta))|S_u|$ 正则的。*9.* 对每个 $u\in U$ 和 $s'\in S_u$，局部交图 $(G_{In_u})_{s'}$ 都是一个 η-采样器。*10.* 对每个 $s\in S$，图 (V_s,U_s) 都是一个 β-采样器。*11.* 在 G 的解码图上存在一个长度为 $O(\log^3 N\,\mathrm{poly}(\eta^{-1}))$ 的低拥塞子集内部路由方案。我们还将为该集合系统实现第 6.7 节中的电路，并获得以下保证。

断言 5.5（电路实现（非正式））。令（V、S、U、G）为命题 5.4 中的构造。那么，该系统可由对数空间一致、规模为（$\log N$）$^{\exp(O(\frac{1}{\varepsilon}))}$、深度为 $\tilde{O}$（$\log\log N\exp(O(\frac{1}{\varepsilon}))$）的电路实现。此外，集合系统的内部路由电路 $C_{\mathcal{O}^T}$ 可并行向预言机发出 $\exp(O(\frac{1}{\varepsilon}))\ \mathrm{poly}\log N$ 次查询。

与前面章节一样，上述内容仅在非正式意义上成立，因为我们实现这些电路时会引入某个趋于零的小误差，而这不会影响分析（对应于对不可逆矩阵进行编码）。

[32]事实上，对 ε 无需施加任何限制，但在此参数范围之外，集合系统会比 N 大得多。

5.3 到列表恢复的规约（内层解码）

本小节中，我们将介绍另一种归约到低噪声列表恢复的方法，该方法利用了 (V, S, U, G) 和“内码” $In_u = (V_u, S_u)$ 所假设的采样性质。具体而言，我们的主要目标是证明：对于 (V, S) 的任意输入字 w 以及全局函数 $f \in \mathcal{B}_{1-\varepsilon, \delta}(w)$，我们能够近似解码 f 在 u 上的限制，并将结果作为一个小列表 $\mathcal{I}_u$ 的一部分，且这在几乎每个内码 In_u 中都成立（大致上，这将 (V, U) 上的列表恢复问题归约为此问题）。形式化地，我们证明：

引理 5.6。 固定 $\delta_{in}, \varepsilon, \eta, \beta > 0$，并令 (V, S, U, G) 如命题 *5.4* 中所述。令 w 为 $\mathcal{C}_{(V,S)}$和 $f \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(w)$的输入字。则存在一个随机电路 C_{list-u}，满足：

1. 以 $u \in U$、一个索引 $i = 1$、2、…、 $\frac{10}{\varepsilon}$ 和 $v \in V_u$ 为输入，并输出 Σ_1 中的一个符号 。
2. 对于 $1-\beta$ 比例的随机字符串，存在 $1 - \frac{10\beta}{\varepsilon\delta_{in}^2}$ 比例的 $u \in U$，使得存在 i，满足 $C_{list-u}(u, i, \cdot)$ 在 u 上与 f 的距离为 $O(\sqrt[3]{\delta_{in}})$。
3. C_{list-u}向 w 发起 多项式$(\frac{1}{\delta_{in}}, \frac{1}{\varepsilon},$对数 $\frac{1}{\beta})$次查询
4. C_{list-u}规模为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度为 $\tilde{O}(\log^2(\log N)\exp(O(\frac{1}{\varepsilon})))$。

明确地说，列表 $\mathcal{I}_u$由以下函数组成： $\frac{10}{\varepsilon}$函数 $\{C_{list-u}(u, i, \cdot)\}_{i\in[\frac{10}{\varepsilon}]}$。

我们指出，上述内容的细粒度查询复杂度（从列表中查询的比特数，而非完整符号的数量）可以达到 $\tilde{O}(\frac{1}{\varepsilon\delta_{in}^2})$，参见备注 D.1。

引理 5.6 的证明分为三部分。第一部分不出所料，是证明我们的超图三元组中的每个 $In_u = (V_u, S_u)$ 都是良好的 aLLDC。

命题 5.7。 设 ε、$\delta_{in} > 0$ 如命题 *5.4* 中所述，并设 η。设 $u \in U$ 和 In_u 是命题 *5.4* 中所述的集合系统，即

1. 集合系统 (V_u, S_u) 是一个 η-采样器。
2. 对于每个 $s' \in S_u$，局部交图[33] $(G_{In_u})_{s'}$ 都是一个 η-采样器。
3. 局部解码图 G_{In_u} 是 $(1 - O(\eta))|S_u|$ 正则的。

那么， In_u上的直积码是一个 $(\delta_{in}, 13\sqrt[3]{\delta_{in}})$-近似、$(\varepsilon, \frac{10}{\varepsilon}, q = \mathrm{poly}(\frac{1}{\delta_{in}}, \frac{1}{\varepsilon}))$-*LLDC*，其解码所用的对数空间一致电路规模为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，规模和深度为 $\tilde{O}(\log\log N\exp(O(\frac{1}{\varepsilon})))$。

命题 5.7 的证明在技术上较为复杂，但其思路与 [36] 中最初的 aLLDC 分析颇为相似，均依赖于内码的抽样和密度（尽管具体方式略有不同），因此证明推迟至附录 D。不过，我们在此通过图 4 展示该算法，部分原因是为了便于之后引用。

现在请注意，在我们对列表译码的定义中，我们输出的电路只需以概率 $\frac{3}{4}$ 成功解码某个特定函数 f。因此，即便按所写的命题 5.7 来假设，我们也只能保证以常数概率成功对某个特定的 f 进行 δ_{out}-计算。这显然不足以证明引理 5.6，因为该引理要求以高概率解码 f。我们可以通过简单地将解码器重复 $O(\log \frac{1}{\beta})$ 次来放大成功概率，但这会使列表大小膨胀。我们将在下一节看到，这是一个真正的问题，因为为了归约到概述中所讨论的“良好分离”设定，我们将在列表大小上付出指数代价，因此我们确实需要它保持为常数。

所幸，这个问题有一个简单的解决办法（这是引理 5.6 的第二部分）。我们可以运行一个基本的剪枝过程，将列表大小削减到 $O(\frac{1}{\varepsilon})$：

命题 5.8 (列表剪枝（Claim D.2）)。 固定 $\tau > 0$，其参数为 ε、δ_{out}，一个输入字 w_{In_u}，以及$\{C_1, \ldots, C_L\}$通过多次运行解码器 $InDec_{\delta_{in}}$ 的生成算法所得到的一组输出电路。假设图 (V_u, S_u) 是一个 $\frac{\delta_{out}^2\varepsilon^2}{256}$-采样器。存在一个电路 C_{Prune}，它通过预言机访问 $\{C_i\}$并获得输入列表后，输出一个规模为 $|L'| \le O$ 的子集 $L' \subset [L]$($\frac{1}{\varepsilon}$)。以至少 $1-\tau$ 的概率，任何原始的 $C_i \in \mathcal{B}_{1-\varepsilon, \delta_{out}}(\mathcal{L})$ 都与某个 $C_j \in L'$ 相差至多 $6\delta_{out}$。

[33]回想一下，这是这样一个图：一侧是 s中的顶点，另一侧是交集 $s' \cap s$，即与 s相交的那些$s' \in S_u$。

内部解码器 $InDec_{\delta_{in}}(s,\ v)$:

* **输入:**

- 一个集合 $s \in S_u$。
- 一个目标顶点 $v \in V$。
- 对码字 w 的预言机访问

*** 输出:** Σ中的一个值。

1. 对于 $j = 1$，2、~~……~~ $^{4\log_{\varepsilon}\delta_{in}}$: (a) 抽样一个集合 $s'_j \ni v$。(b) 测试以下条件是否成立：i. 集合 s 与 s'_j 相交，且 ii. 如果 $\mathrm{dist}^{s\cap s'_j}(w_s,\ w_{s'_j}) < 4\delta_{in}$。(c) 如果 s'_j 通过测试，则退出循环并输出 $w_{s'_j}(v)$

2. 如果没有任何一个 s' 通过，则输出一个任意符号。

图 4：内部解码器。命题 5.7 可通过对 $s \in S_u$ 进行 $O(\frac{1}{\varepsilon})$ 次随机选择，并输出 $InDec_{\delta_{in}}(s,\cdot)$ 得出。形式化证明见附录 D。

此外，C_{Prune} 是一个对数空间一致电路，规模为 $\mathrm{poly}(|L|, \frac{1}{\varepsilon}, \frac{1}{\delta_{out}}, \log\frac{1}{\tau})$，深度为 $O(\log^2|L| \cdot \log\frac{1}{\delta_{out}\varepsilon\tau})$，对 $\{C_i\}$ 最多进行 $O(\log\frac{L}{\tau}\max\{\varepsilon,\delta\})$ 次预言机调用；它是一个码字 w、生成随机 $s \in S_u$ 和随机 $v \in V_u$ 的电路，以及超图邻接电路。

同样，这个证明并没有特别的启发性（只需查询输入列表，丢弃任何并非由 w 实际计算出的电路，然后找到其余电路的一个小覆盖；码上的组合列表大小界保证了这一点）。证明也推迟至附录 D。

现在，多次运行断言 5.7 以放大成功概率，然后对得到的列表进行剪枝，我们显然可以构造一个类似引理 5.6 中的电路，使其在 f 在 In_u 中全局时，以高概率计算出 f。因此，最后一步是通过采样论证：任何在 $(V,\ S)$ 中全局的 f，也应当通过采样在几乎所有局部码 In_u 中全局。

断言 5.9。任取满足 $\delta \le 4\sqrt{\varepsilon}$ 的 ε、δ、$\beta > 0$，并设 $(V$、S、U、$G)$ 满足

1. $(S$、$U)$ 是一个 β-采样器。*2.* 对每个 $s \in S$，图 $(V_s$、$U_s)$ 都是一个 β-采样器。

那么，对于任意 $f \in \mathcal{B}_{1-\varepsilon,\ \delta}(w)$:

$$\mathop{\mathbb{P}}_{u\in U}\left[f|_u \in \mathcal{B}_{1-\frac{\varepsilon}{2},2\delta}(w_{In_u})\right] \ge 1 - \frac{9\beta}{\delta_{in}^2\varepsilon}$$

这里可以将 β 看作 $^{1}\ \mathrm{polylog}(N)$~~，将 δ~~ 看作上文中 δ_{in} 的某个常数倍。我们指出，对 δ 的条件只是为了计算简便性，对后续论证没有影响，因为无论如何我们都会取 $\delta \le \exp(-1/\varepsilon)$。还要注意，如果满足 $f \in \mathcal{B}_{1-\varepsilon,\ 0}(w)$，那么命题 5.9 可直接由 $(S,\ U)$ 的采样推出。然而，由于 $f \in \mathcal{B}_{1-\varepsilon,\ \delta}(w)$，事先无法排除这样的情况：$\mathrm{dist}_s(g,\ f) < \delta$，但与 u 相交后距离却增大。在这种情况下，$g|_{s\cap u}$ 将无法充分计算 $f|_u$，我们可能无法解码 f。所幸，对 $(V_s,\ U_s)$ 进行采样可确保这种情况很少发生。

命题 *5.9* 的证明。固定 f，并令 $A \subseteq S$ 为满足 w_s 与 f 的距离不超过 δ 的集合。根据定义，$\mathbb{P}[A] \ge \varepsilon$。如果 f 在 In_u 中不是全局的，那么发生了以下事件之一：

1. 要么 $\mathop{\mathbb{P}}_{s\sim u}[A] < \frac{3\mathbb{P}[A]}{4}$，要么

2. 超过 $\frac{\mathbb{P}[A]}{4}$ 个集合 $s \in A$ 邻居 u 与 f 的距离为 w_s δ，但

$$\operatorname*{dist}_{s\cap u}(f, w|_{s\cap u}) > 2\delta.$$

这是因为，如果第一项和第二项都不满足，那么至少有 $\frac{\mathbb{P}[A]}{2} \geq \frac{\varepsilon}{2}$ 比例的 $s \sim u$满足 $\operatorname{dist}^{s\cap u}(f, w|_{s\cap u}) \leq 2\delta$，正如所需。

因此，只需界定这两个事件的概率。注意，根据 (S, U) 之间的全局采样性质，第一个事件的概率至多为 $\frac{16\beta}{\varepsilon^2} \leq \frac{\beta}{\delta^2\varepsilon}$。

为界定第二个事件，对于任意 $s \in A$，称满足 $s \sim u$ 的数对 (s, u) 为误导 数对，若 $\operatorname{dist}(f, w_s) < \delta$，但 $\operatorname{dist}^{s\cap u}(f, w_s|_{s\cap u}) > 2\delta$。根据 (V_s, U_s) 的 β-采样性质，对于每个 $s \in A$，至多有 $\frac{\beta}{\delta^2}$ 个邻居 $u \sim s$ 使得 (s, u) 是误导数对，因此误导数对总数至多为 $\frac{\beta}{\delta^2}$。根据马尔可夫不等式，

$$\mathop{\mathbb{P}}_{u\in U}\left[\mathop{\mathbb{P}}_{s\sim u}[(u,s) \text{ is misleading}] > \frac{\mathbb{P}[A]}{4}\right] \leq \frac{4\mathop{\mathbb{P}}_{u,s}[(u,s) \text{ is misleading}]}{\mathbb{P}[A]}.$$

根据上述内容，分子上界为 $\frac{4\beta}{\delta^2}$，分母则由 ε 限定。

另一方面，如果 u 满足列表中第一项的条件，却违反上述列表中的第二项，那么它会计入 $\mathop{\mathbb{P}}_{u\in U}\left[\mathop{\mathbb{P}}_{s\sim u}[(u, s) \text{具有误导性}] > \frac{\mathbb{P}[A]}{4}\right]$。因此，此类 u 至多有 $\frac{8\beta}{\delta^2\varepsilon}$ 个。将此概率与违反第一项条件的概率相加，即可得到该断言。

□

现在我们可以证明引理 5.6 了。

引理 *5.6* 的证明。任取一个输入字 w，使其属于 (V, S)，并令 $f \in \mathcal{B}_{1-\varepsilon,\ \delta_{in}}(w)$。电路 C_{list-u} 在输入 (u, i, v) 上运行 In_u-内部解码器 $O(\log \frac{1}{\beta})$ 次[34]，并将得到的 $L = O(\frac{\log \frac{1}{\beta}}{\varepsilon})$ 个电路 $C_1, \ldots\ldots, C_L$ 输入到 C_{prune} 中，其失败概率为 $\tau \leq O(\beta^2)$。电路 C_{prune} 输出经过剪枝后保留下来的 $L' \subseteq L$ 个电路的索引。最后，C_{list-u} 使用 i 对输出子集 $L' \subset [L]$ 中的 C_{prune} 进行索引，并生成 $C_{L'(i)}(v)$。[35]

我们需要论证：在 C_{list-u} 的内部随机性上，有 $1-\beta$ 比例的时间存在 $1 - \frac{10\beta}{\varepsilon\delta_{in}^2}$ 比例的 $u \in U$，使得 $C_{list-u}(u, i, \cdot)$ $O(\sqrt[3]{\delta_{in}})$-计算 f。为看出这一点，注意根据断言 5.9，我们保证除 $\frac{9\beta}{\varepsilon\delta_{in}^2}$ 比例的 $u \in U$ 之外，对所有情形都有 $f \in \mathcal{B}_{1-\frac{\varepsilon}{2},2\delta_{in}}(w_{In_u})$。由于我们已经将内部解码器运行了 $O(\log \frac{1}{\beta})$ 次，根据命题 5.8，对每一个这样的“良好” u，我们确信在内部随机性上以至少 $1 - O(\beta^2)$ 的概率确实输出一个确定性电路 $O(\sqrt[3]{\delta_{in}})$-计算 f。因此，根据马尔可夫不等式，在内部随机性上，这些良好 $u \in U$ 中有超过 β 比例未能拥有这样一个索引的概率至多为 β，从而至多留下 $\frac{10\beta}{\varepsilon\delta_{in}^2}$ 比例没有计算 f 的良好索引，正如所需。

电路复杂度 根据命题 5.7，内解码器电路可在规模 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$ 和深度 $\tilde{O}(\log(\log N)\exp(O(\frac{1}{\varepsilon})))$ 内实现。列表剪枝过程对 $\{C_i\}$ 的谕示进行 $\operatorname{poly}(\log N)\exp(O(\frac{1}{\varepsilon}))$ 次并行调用。最后，内部解码器的输出被送入以置信度 $\tau = (\log^6 N \cdot \exp(O(\frac{1}{\varepsilon})))^{-1}$, $L = \operatorname{poly}(\frac{1}{\varepsilon}, \log\log N)$ 和 $\delta_{out} = O(\delta_{in}^{2/3}) = \exp(O(\frac{1}{\varepsilon}))$ 实例化的列表剪枝电路，这不会增加渐近规模，但会将深度增加到 $\tilde{O}(\log^2(\log N)\exp(O(\frac{1}{\varepsilon})))$。

□

[34]请注意，我们必须谨慎处理 C_{list-u} 究竟如何访问此处所述的内部解码器。也就是说，我们不必为每个 u 都在电路中硬编码一个单独的内码（这会使电路规模大幅膨胀）。相反，如下文将很快介绍的，我们只需使用图 4 中 $InDec(s, g_i, \cdot)$ 电路的 L 个副本，并结合我们的电路实现，确保输入（并在该过程中查询）的 s 是 u 的适当邻居。[35]如果某个 L' 小于指标数量，那么对于较大的 i，输出可以是任意值。

5.4 从常数准确率进行列表恢复（外层解码)

上一节为我们提供了一种归约到低噪声列表恢复区域的新方法，即对每个 $u \in U$ 构造列表 $\mathcal{I}_u$，使得任何固定的 (δ, ε)-全局函数 f 在 (V, S) 中都会出现在几乎所有列表中。因此，如第 3 节所述，我们现在只需要一个在 (V, U) 上的列表恢复算法，用以恢复在列表 $\mathcal{I} = \{\mathcal{I}_u\}$ 中为 $(\delta, 1 - {}^{1}\!/_{\mathrm{poly}\log(N)})$-全局的函数。

与之前一样，从较高层次来看，我们针对（V，U）的列表恢复算法将从一个随机外层集合 u 和一个函数 $g \in \mathcal{I}_u$ 开始，该函数与全局函数 f 的距离为 δ_{in}。为了解码 v，我们将路由到一个随机的 $u' \ni v$，并希望通过这一过程找出某个接近 f 的 $g' \in \mathcal{I}_{u'}$，然后输出 $g'(v)$。在定理 3.4 中，我们通过跟踪接近初始 g 的函数列表证明，可以确保最终输出的 g' 与 f 的距离为 $O(\delta_{in}$ 多对数$(N))$。

遗憾的是，正如上文所述，这对于常数码率情形并不足够，因为 δ_{in}（内码的解码准确率）也是常数。[36] 这意味着我们无法承受误差出现接近多对数(N)的乘法膨胀，因此需要一种新的传播解码方法。本节将介绍我们基于强化版子集内部路由概念提出的新方法，其界为 $\delta_{out} = \delta_{in}\exp(O({}^{1}\!/_{\varepsilon}))$。

定理 5.10. 令 $N > 0$, $\delta_0, \beta > 0$ 满足 $\beta < \left({}^{\varepsilon\delta_0}\!/_{\log N} \right)^{28}$。令 (V, U) 为一个满足以下条件的集合系统：

1. V 有 $\leq N^2$ 个顶点。[37]
2. U 具有一个最大长度为 $O(\log^3 N \exp({}^{5}\!/_{\varepsilon}))$ 的子集内部路由方案。
3. 对于每个 $u \in U$，交图 G_u 都是一个 β-采样器。

那么，$\mathcal{C}_{(V,U)}$ 是一个 $(\delta_0, 8^{103/\varepsilon}\delta_0))$-近似、${}^{10}\!/_{\varepsilon}$-输入 $(1 - {}^{1}\!/_{\log^{10} N}, \tilde{O}({}^{1}\!/_{\varepsilon}), \mathrm{poly}\log_{\delta_0} N \exp(O({}^{1}\!/_{\varepsilon})))$-*LLRC*。

请注意，这些略显复杂的参数是固定的，因此我们可以将内码直接代入这一结果。

实现常数 δ_{in} 的列表恢复的第一个关键思路，是对 [60] 中“列表剪枝”方法的改编，以确保输入给我们的列表恢复算法的列表是良好分离的。

定义 5.11。设 $\mathcal{I}_u$ 是定义域 u 上的函数列表。

- 若列表 $L \subset \mathcal{I}_u$ 中的每个函数 $g \in \mathcal{I}_u$ 都与某个函数 $g' \in L$ δ-接近，则称该列表是 δ-稠密的（相对于 $\mathcal{I}_u$）。
- 若对于任意一对不同的函数 g_1、$g_2 \in L$，都有 $\mathrm{dist}(g_1, g_2) > \delta'$，则称列表 $L \subseteq \mathcal{I}_u$ 是 δ'-分离的。
- 若列表 $L \subseteq \mathcal{I}_u$ 是 δ-稠密且 6δ-分离的，则称其为 δ-良好分离 的。

需要说明的是，常数 6 是任意选取的（任何大于 2 的常数都足够；这里固定取 6 是为了简化计算）。

粗略地说，如果每个列表都在相同的尺度下 δ良好分离，那么我们沿着解码图前进时，就能唯一地匹配列表，从而避免每一步引入额外的差错。遗憾的是，虽然我们可以确保每个列表对于某个 δ 都是 δ-良好分离的，但我们不知道如何确保它们在相同的尺度下分离，而这对于此类无损传播至关重要。相反，我们将在下文说明如何确保系统中的集合有一个较大的常数比例（例如 99%）在相同的尺度下良好分离，因此具有无损传播（而且，我们还能以一种使该条件可测试的方式做到这一点）。记 $T \subseteq U$ 为我们能够获得此类良好分离列表的 $u \in U$，问题

[36]这是任何常数码率直积码（甚至其他具有常数码率和局部性的码）所固有的性质。直积码的码率是 (V, S) 中一个顶点的平均度，我们将其记为 d。我们总能找到一个相对大小为 ${}^{1}\!/_{2d}$ 的顶点集 $V' \subseteq V$，使其在 S 中的邻域规模为 $\leq {}^{1}\!/_{2}$。将这样的集合记为 $V' \subseteq V$。现在，若我们取一个码字 $Enc(f)$，并擦除/用噪声替换 V' 邻域中的集合 $Enc(f)$，仍会得到一个与 $Enc(f)$ 距离为 $1/2$ 的码字。然而，在这样的码字中，关于 V' 的所有信息都已丢失，因此我们无法指望恢复 V' 的任何比特（即使采用非局部算法也不行）。尤其是，即使 $\varepsilon = {}^{1}\!/_{2}$，我们也只能使用与 ${}^{1}\!/_{2d}$ 的距离为.....的解码函数。[37]严格来说，我们希望 V 恰好有 N 个顶点。命题 5.4 只能保证其规模近似如此，但 N^2 的界已经绰绰有余。

于是就归结为在 T 内部进行子集路由；根据假设，我们具备这一能力（并将在第 6.4 节中明确构造）。

现在我们可以介绍图 5 中外层解码器的解码电路了。该电路基于以下抽象出的组件：

1. **列表的实现** $\mathcal{I} = \{\mathcal{I}_u\}_{u\in U}$**：** 回想一下，实际上我们从集合 $s \in S$ 上的一个输入字开始，而不是从 $u \in U$ 开始。引理 3.5 中的电路 C_{list-u} 可以对 w 进行内层解码，得到列表 $\mathcal{I}_u$，并模拟对其中函数的查询。我们再次强调，$\{\mathcal{I}_u\}_{u\in U}$ 的重要之处在于：对于原始码字 w 在 S 上以 $(\varepsilon,\ \delta_{in})$ 计算的函数 f，在构造出的列表中有很大一部分（占 U 上列表的 $1 - {}^{1}\!/_{\mathrm{poly}\log(N)}$）包含一个与之接近的函数。我们用列表 $\mathcal{I}$ 来表述外层解码器的定理。计算电路复杂度时，我们再回到实际的电路实现。

2. **列表良分离电路**：如上所述，我们需要一个电路 C_T，它以列表$\mathcal{I}_u$ 和一些 j 作为输入，并输出 $FAIL$ 或一个列表 $L_u \subseteq \mathcal{I}_u$，该列表在 $\mathcal{I}_u$ 内是 $1.168^j\delta_{in}$- 良好分离的。我们将在下一小节证明下面的引理。

> **引理 5.12**。对于每个 $\delta_0 > 0$，都存在一个随机电路 $C_T^{\delta_0}$，它以 u（以及列表 $\mathcal{I}_u$ 的某种描述）和一个索引 $j \in \left\{1, 2, \ldots, {}^{1000}\!/_{\varepsilon}\right\}$ 作为输入，输入规模为 ${}^{10}\!/_{\varepsilon}$。它要么输出 $FAIL$，要么输出一个列表 $L_u \subseteq \mathcal{I}_u$，并满足以下保证。
>
> *(a)* 对于任意 $\mathcal{I}_u$，至多有比例为 0.01 的 $j \in \left\{1, 2, \ldots, {}^{1000}\!/_{\varepsilon}\right\}$，电路会以概率 $> {}^{1}\!/_{\text{多项式对数 } N}$ 输出 $FAIL$。~~*(b)* 如果~~ $C_T^{\delta_0}$ 没有输出 $FAIL$，那么以概率 $1 - {}^{1}\!/_{\text{多项式对数 } N}$，输出列表 $L_u \subseteq \mathcal{I}_u$ 在 $\mathcal{I}_u$ 中是 δ_j'-良好分离的，其中 $\delta_j' = 1.16 \cdot 8^j\delta_0$。
>
> 保证条件中的多项式对数 N 可以是我们选择的任意多项式，特别地，可将其设为 ${}^{1}\!/_{\text{对数}^{10}\ N}$。
>
> 该引理的证明将在下一节给出，同时我们也会分析其电路复杂度。目前，我们先接受这一点，并继续假设可以构造出该电路。

3. **子集内部路由方案**：记为 $\mathcal{O}^T$。明确地说，该路由方案以一对 $u,\ u' \in U$ 作为输入，并应输出一条从 u 到 u' 的路径，使得所有顶点（端点或许除外）都属于集合 T。根据命题 5.4，这样的路由方案存在。这里，集合 $T \subseteq U$是满足以下条件的 u的集合：$C_T^{\delta_0}(u,\ j)$以至多 ${}^{1}\!/_{\text{对数}^{10}(N)}$ 的概率输出失败（依据引理 5.12 的性质 (a)），其中 j 将作为参数提供给解码器（特别地，它将被随机抽样）。形式化地说，当我们在集合 $u'' \in U$上调用 T-成员资格预言机时，我们对其应用 $C_T^{\delta_0}$，并检查它是否失败。

我们的生成算法随机抽取一个 u_0、一个随机的 $j \in [{}^{1000}\!/_{\varepsilon}]$，并输出电路 $\{OutDec_{\delta_0,j_i}(u_0,\ g,\ \cdot)\}_{g\in\mathcal{I}_u}$。现在我们准备证明本小节的主要结果：从子集内部路由方案中进行列表恢复：

定理 *5.10* 的证明。证明与定理 3.4 类似。固定列表 $\mathcal{I}$和 $f \in \mathcal{B}$ ${}^{1}\!/_{\log^6 N},\ \delta_0(\mathcal{I})$。我们的目标是证明，对于大多数 u_0和 $j \in [{}^{1000}\!/_{\varepsilon}]$的取值，都存在$g \in \mathcal{I}_{u_0}$，使得 $OutDec_{\delta_0,\ j}(u_0,\ g,\ \cdot)$ 近似计算 f：

$$\mathop{\mathbb{P}}_{u_0,j}\left[\exists g \in \mathcal{I}_{u_0} : \mathop{\mathbb{P}}_{v,r}\left[OutDec_{\delta_0,j}(u_0,g,v) = f(v)\right] \geq 1 - \delta_1\right] \geq \frac{3}{4}$$

特别地，通过尝试 所有 $g \in \mathcal{I}_{u_0}$，生成算法将以高概率成功。

为此，令 $T = T_j$ 为满足以下条件的 u 的集合：$C_T^{\delta_0}(u,\ j)$ 输出 FAIL 的概率至多为${}^{1}\!/_{\log^{10}(N)}$；并回顾一下，子集内部路由要求$|T| \geq 0.9|U|$。根据引理 5.12 的性质 (a)，对随机 j取期望时，T的规模为 $0.99|U|$，因此根据马尔可夫不等式，$|T| < 0.9|U|$的概率至多为 0。1。

现在，假设我们抽样得到一个良好的 $j$with $|T| \geq 0.9|U|$。与定理 3.4 中一样，我们进一步定义若干罕见的“坏事件”；只要这些情况都不发生，我们的解码器就能成功。具体而言，令 B表示满足以下任一条件的边 $\{u, u'\}$ 集合：

外层解码器 $OutDec_{\delta_0,j}(u,g,v)$:

* **输入:**

- $u \in U$ 与一个函数 $g \in \mathcal{I}_u$。
- $v \in V$。
- 参数 j 与 δ_0。

*** 输出:** Σ_1 中的一个值。

1. 令 $T \subseteq U$ 为所有满足以下条件的 u 的集合: $C_T^{\delta_0}(u,j)$ 不会失败。
2. 抽取一个随机的 $u' \in U$, 使得 $u' \ni v$。
3. 调用 $\mathcal{O}^T(u,\ u')$。如果路由方案本身失败, 则中止并输出一个任意符号。否则, 令 $P = (u = u_0,\ u_1, \ldots\ldots,\ u_t = u')$ 为其输出路径; 对于 $k < t$, 用 L_{u_k} 表示由 $C_T(u_k,\ j)$ 输出的列表。
4. 对于 $k = 1, \ldots, t-1$:
   (a) 找到某个 $g_k \in L_{u_k}$, 使得 $\mathrm{dist}^{u_k \cap u_{k-1}}(g_{k-1},\ g_k) < 3 \cdot \delta'_j$。
   (b) 如果存在多个这样的函数, 则均匀随机地选择其中一个。
5. 找到任意满足 $\mathrm{dist}^{u_t \cap u_{t-1}}(g_{t-1}, g_t) < 3 \cdot \delta'_j$ 的 $g_t \in \mathcal{I}_{u_t}$, 并输出 $g_t(v)$。

图 5: 外层解码器电路算法

1. **坏列表:** 不存在 δ_0-接近于 f 的 $g'' \in \mathcal{I}_{u''}$。
2. **不良采样:** 存在某个函数 $g'' \in \mathcal{I}_{u''}$, 使得

$$\left|\operatorname*{dist}_{u''}(f,g'') - \operatorname*{dist}_{u \cap u'}(f,g'')\right| > \delta_0.$$

若路径中没有边属于 B, 则称路径 $(u_0,u_1,\ldots,u_m)$ 为良好。我们断言, 只需证明大多数 u_0 到随机端点 u_t 的预言机路径大多是良好的:

$$\Pr_{u_0,j}\left[\Pr_{u_t,\mathcal{O}^T}[\mathcal{O}^T(u_0,u_t) \text{ doesn't fail and the path is good}] \geq 1 - \frac{\delta_1}{2}\right] \geq 99\% \tag{5}$$

注意, 内部概率隐式地依赖于 j, 因为 T 是相对于 $C_T(\cdot,\ j)$ 定义的。

我们先假设 (5) 成立, 并完成证明。固定任意满足内部条件的 $u_0,\ j$。根据假设, 存在某个与 f 距离至多为 δ_0 的 $g \in \mathcal{I}_{u_0}$。我们断言, 对于任意这样的固定选择, 在随机 v 上, 概率 $OutDec_{\delta_0,\ j}(u_0,\ g,\ v) = f(v)$ 至少为 $1 - \delta_1$。对良好 j 和 u_0 的选择应用并集界, 即可得到最终所需的解码保证。

为说明这一点, 先回顾: 在输入 v 上, 解码器会随机抽取一个 $u_t \ni v$, 通过 $\mathcal{O}^T(u_0,\ u_t)$ 找到通往它的路径, 并针对某个有效的 $g_t \in L_{u_t} \subseteq \mathcal{I}_t$ 选择输出 $g_t(v)$。因此, 我们的目标是给出下式的下界

$$\Pr_{r,v}[OutDec_{j,\delta_0}(u_0,g,v) = f(v)] = \Pr_{v \in u_t, \mathcal{O}^T, g_t}[g_t(v) = f(v)]$$

其中, $g_t \in L_{u_t}$ 是算法最后一步得到的函数; 我们还记得, $L_{u_t} \subset \mathcal{I}_{u_t}$ 是良分离电路 $C_T(u_t,\ j)$ 的输出列表。需要注意的是, 在等式两边, 如果 $\mathcal{O}^T$ 失败或不存在有效的 g_t 选择, 我们都不将其计入概率。

现在考察右侧概率, 注意我们已均匀随机抽取了 $v \in V$, 然后抽取了一个随机邻居 $u_t \in v$。由于 (V,U) 是正则的, 这等价于先抽取一个随机的 $u_t \in U$, 再子采样一个随机顶点 $v \in u_t$。这意味着, 只需论证在 $u_t, \mathcal{O}^T$ 的随机性下,

最终函数 g_t 在 u_t 上以高概率 δ_1-接近 f。令 $L'_{u_t} \subseteq L_{u_t}$ 为与 g_{t-1} $3\delta_0$ 接近的可能函数。

$$\Pr_{u_t, \mathcal{O}^T}\left[\forall g_t \in L'_{u_t} : \mathrm{dist}(g_t, f) \le \frac{\delta_1}{2}\right] \ge 1 - \frac{\delta_1}{2}$$

特别地，对于 $u_t \in U, g_t \in L'_{u_t}$ 和 $v \in u_t$ 的随机选择，我们正确输出 $f(v)$ 的概率至少为 $(1 - \frac{\delta_1}{2})^2 \le 1 - \delta_1$，符合预期。

现在来证明上述结论。我们引入最后一个坏事件；只要该事件不发生，我们的解码器就能成功：距离测试器出错的事件（回想一下，我们对子抽样交集中的点来测试距离，而不是检查完整符号，以确保解码电路保持较小）。根据构造（以及断言 F.1），该事件发生的概率至多为 $\frac{\delta_1}{4}$。

现在，假设以下条件成立：路由预言机没有失败，$\mathcal{O}^T(u_0, u_t)$ 良好，并且 上述距离测试器在路径上没有出错（根据并集界，该事件发生的概率至少为 $1 - \delta_1$ [38]）。我们断言，在这些条件下，当我们从承诺的、与 f 的距离为 δ_0 的 $g = g_0$ 开始时，每个 $g_t \in L'_{u_t}$ 都与 f 的距离为 $\frac{\delta_1}{2}$。

这正是该证明不同于定理 3.4 之处。根据路径的良好性，对于每个 u_k，列表 $\mathcal{I}_{u_k}$ 中都有一个与 f δ_0接近的函数。根据每个 L_{u_k} 的 δ'_j-良分离性（对于 $k = 1$， 2、.....、$t-1$），存在一个唯一的 函数 $g' \in L_{u_k}$，它与 f $\delta'_j + \delta_0$接近。我们断言，在算法的每一步中所选取的函数都是这个函数。这一点由对 k的归纳法得出。对于 $k = 0$，这是因为根据假设，$g = g_0$ 与 f δ_0接近。现在假设对于 g_{k-1} 结论成立，并证明对于 g_k 也成立。我们的假设是 $\mathrm{dist}^{u_{k-1}}(g_{k-1}, f) < (6^j + 1)\delta_0$。因此，根据边的良好性，在交集

$$\operatorname*{dist}_{u_{k-1} \cap u_k}(g_{k-1}|_{u_{k-1} \cap u_k}, f) < \delta'_j + 2\delta_0.$$

现在，一方面，如果 g_k 是与 f $\delta'_j + \delta_0$接近的函数，那么在交集中也有

$$\operatorname*{dist}_{u_{k-1} \cap u_k}(g_k|_{u_{k-1} \cap u_k}, f) < \delta'_j + 2\delta_0$$

因此，根据三角不等式，

$$\operatorname*{dist}_{u_{k-1} \cap u_k}(g_{k-1}|_{u_{k-1} \cap u_k}, g_k|_{u_{k-1} \cap u_k}) < 2\delta'_j + 2\delta_0 \le 3 \cdot \delta'_j.$$

另一方面，对于任意其他的 $g' \in L_{u_k}$，根据良分离性，它与 f相距 $5\delta'_j - \delta_0$。[39]根据这条边的良好性，在交集中

$$\operatorname*{dist}_{u_{k-1} \cap u_k}(f|_{u_{k-1} \cap u_k}, g'|_{u_{k-1} \cap u_k}) > 5\delta'_j - 2\delta_0.$$

根据（三角不等式的）反向形式，

$$\operatorname*{dist}_{u_{k-1} \cap u_k}(g_{k-1}|_{u_{k-1} \cap u_k}, g'|_{u_{k-1} \cap u_k}) > 5\delta'_j - 2\delta_0 - \operatorname*{dist}_{u_{k-1} \cap u_k}(g_{k-1}|_{u_{k-1} \cap u_k}, f)$$

根据上述界限，这至少为 $5\delta'_j - 2\delta_0 - (\delta'_j + 2\delta_0) = 4\delta'_j - 4\delta_0 > 3 \cdot \delta'_j$。因此，$g_k$ 在第 k 步中只能选择 g_k。

在最后一步中，我们无法保证存在一个能通过测试的单一函数，但通过类似于上述的论证，函数 g_t 将与 f 相距不超过 $3\delta'_j$，而这小于 $\frac{\delta_1}{2}$。

证明的其余部分沿用定理 3.4 的相同步骤。尚需证明方程 (5)，即许多 u_0， j 都只有很少的坏路径。为此，首先注意到，根据马尔可夫不等式，只需证明：随机选择 u_0， u_t 和路径 $\mathcal{O}(u_0,$ $u_t)$ 时， $\mathcal{O}^T(u_0,$ $u_t)$ 失败或命中 B 的概率极低，例如：

$$\Pr_{u_0, u_t, \mathcal{O}^T}[\mathcal{O}^T(u_0, u_t) \text{ fails or hits } B] \le \frac{1}{\log^2 N}.$$

[38]这里我们还利用了上面选取的 j 使失败概率 $\frac{1}{\log^2 N}$.

[39]这只是利用了 事实是， f 与 g_k相距 $\delta'_j + \delta_0$， g_k 和 g' 相距 $6\delta'_j$，以及反向三角不等式 y.

这是由低拥塞路由所界定的情形。具体而言，由于 $\mathcal{O}^T$ 是低拥塞的，只需证明 B 的测度很小，即 $\Pr[B] \leq \frac{1}{\log^9 N}$。这基本上可由采样直接推出，并利用如下保证：一 $u \in U$ 的列表中，除了至多占 $\frac{1}{\log^{10} N}$ 比例的列表外，其余列表都包含一个与 f 相距 δ_0 的函数。一后者保证随机边满足“坏列表”条件的概率至多为 $2\frac{\log}{\log^{10} N}$，这是由并集界得到的。前者则说明，对于任意固定的 u，随机邻居 u' 对 u 的列表中的任意固定函数采样错误的概率至多为 $\frac{\beta}{\delta_{in}^2}$。对随机边两端 s 和 s' 的列表应用并集界，这一事件发生的概率因此至多为 $\frac{2\beta}{\varepsilon\delta_0^2}$；再应用一次并集界，便得到所需的 $\Pr[B] \leq \frac{1}{\log^9 N}$。

□

5.5 归约到良好分离列表

我们现在通过构造我们的列表稀疏化电路 $C_T^{\delta_0}$，给出引理 5.12 的形式化证明 .

遗憾的是，我们的内码生成的列表并非总是自动良好分散（事实上，几乎可以肯定并非如此）。然而，事实证明，我们可以以某种方式对列表进行剪枝，使得在许多“尺度” j 上，我们都能找到 δ'_j-良好分离的子列表 $\delta'_j \approx \delta_{in} 8^i$。这一论证是对 [60]中类似方法的迭代变体；他们证明了至少存在一个这样的尺度。

我们在图 6 中介绍了为此目的设计的剪枝过程，它基于一个称为“距离图”的基本对象。具体而言，给定一个函数列表 L，我们通过分析（经验）(L, δ_0)-距离图来对 L 进行剪枝。该图的顶点集为 L，边集包含所有满足 $\text{dist}(g, g') \leq \delta_0$ 的 $\{g, g'\}$；距离是在定义域的随机子样本上经验测得的。我们指出，严格来说，电路 $C_T^{\delta_0}$ 是一个随机电路，但正如证明中所解释的，其误差为 $\frac{1}{\text{多对数 } N}$，因此我们可以放心地运行它，而不会损害外层解码器成功的概率。

列表稀疏化器 $C_T^{\delta_0}$:

* **输入:**

- 列表 $\mathcal{I}_u$。
- 一个索引 $j \in \left\{1, 2, \ldots, \frac{1000}{\varepsilon}\right\}$。

* **输出:** 要么 $FAIL$，要么一个列表 $L' \subseteq \mathcal{I}_u$。

1. 从定义域中抽样 $O(\log_{\log_{\delta_0} \varepsilon} N)$ 个随机点，并计算每一对 $g, g' \in \mathcal{I}_u$ 之间的经验距离。

2. 设置 $\delta_k = 8^k \delta_0$ 与 $L_0 = \mathcal{I}_u$。计算 G_0，即经验（L_0, δ_0）-距离图。

3. 对于 $k = 1$、2、.....、$\frac{1000}{\varepsilon}$:

   (a) 找到 $L_k \subseteq L_{k-1}$，它是前一轮计算得到的 G_{k-1} 中的极大独立集（如果是 G_0，则是在循环开始之前计算得到的）。（b）计算 G_k，即（L_k, δ_k）-距离图。

4. 如果子集 L_j 是 δ_{j+1}-分离的，则输出 L_j（根据经验距离），否则输出 $FAIL$。

图 6：外层解码器电路算法

我们指出，无论输出如何，该算法都会运行 $\frac{1000}{\varepsilon}$ 轮，但输出只取决于前 j 轮。这样的特殊选择只是为了便于分析以及便于在电路中实现。我们同样可以在 $k = j$ 时停止循环。

引理 *5.12* 的证明。 我们的算法中唯一的随机化部分，是计算列表中任意两个函数之间的距离。从定义域中抽取 $O(\log_{\log_{\delta_0} \varepsilon} N)$ 个随机点，并计算经验

距离。对每一对 $g,\ g' \in L$，由标准的切尔诺夫界和并集界可知，对任意 $\delta_k = \delta_0 8^k$，任意一对 $g,\ g'$ 的 $d(g,\ g') \le 7.9\delta_k$，但经验距离大于 $8\delta_k$ 的概率至多为1 多对数 N。类似地，对于 $d(g,\ g') \ge 8.1\delta_k$，但测得的距离小于 $8\delta_k$ 的情形也是如此。假设这些事件均未发生。

我们观察到，在每一步中，L_k 在 L_{k-1}中都是 $8.1\delta_{k-1}$-稠密的。这是因为，如果存在一个函数 $f \in L_{k-1}$，它与 L_k中的任何函数的距离都大于 $8.1\delta_{k-1}$，那么根据上述条件化，距离图 G_k中将不会出现连接 f 和 L_{k-1} 的边，这与 L_{k-1}的极大性矛盾。利用三角不等式，对于 $f_0 \in L_0$ 中的每个函数，都存在 $f_1 \in L_1$ 中的一个函数与之距离为 $8.1\delta_0$。存在一个函数 $f_2 \in L_2$ 与 f_1 的距离为 $8.1\delta_1$，以此类推。因此，通过三角不等式可得，对于每个 $f_0 \in L_0$，都存在一个 $f_k \in L_k$ 使得

$$\operatorname{dist}(f_0, f_k) \le \sum_{m=0}^{k-1} \operatorname{dist}(f_m, f_{m+1}) \le 8.1\delta_0 \sum_{m=0}^{k-1} 8^m \le \frac{8.1}{7} 8^{k-1}\delta_0.$$

特别地，L_j 在 L_0中是 $1.16\delta_{j-1}$-稠密的。

接下来我们注意到，如果输出了 L_j，那么它们之间是 $7.9\delta_j$-分离的。事实上，如果输出了 L_j，那么 L_j 中任意两个函数的经验距离都是 $\ge 8\delta_j$，因此根据我们所条件化的事件，它们的实际距离至少为 $7.9\delta_j$。特别地，对于 $\delta'_j = 1.16 \cdot 8^j\delta_0$，如果输出了 L_j，那么它是 δ'_j-良好分离的。

最后请注意，如果没有输出 L_j，那么这意味着 $L_{j+1} \ne L_j$，特别是意味着 $|L_{j+1}| \le |L_j| - 1$。注意，在经历 $\frac{10}{\varepsilon} - 1$ 次失败后，列表就只剩下一个元素，在这种情况下我们总能成功。因此，对于任意给定的列表，失败的索引至多占 0.1 的比例。

□

电路复杂度 最后，我们概述一个标准事实：上述算法可以用对数空间一致、规模和深度均为.....的电路实现。

断言 5.13. 电路 $C_T^{\delta_0}$可由规模为 $\operatorname{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta_0}, \log\log N)$、深度为 $O(\frac{1}{\varepsilon}(\log^2 \log_{\varepsilon\delta_0^{\ N}}))$ 的电路实现，并且对列表电路 C_{list-u}和点电路 $C_{U\to V}$的调用次数至多为 $O(^{\log}{}_{\log_{\delta_0^{\ \varepsilon^N}}})$。

证明。 预言机调用的次数可从描述中直接得出。算法的第一步并行调用采样预言机。随后，算法针对 $\mathcal{I}_u$ 中的每一对函数执行并行两两距离测试（在后续迭代过程中将测试的每个尺度上）。这相当于调用$O(|\mathcal{I}_u|)^2$ 个相等性电路，以及 $O(|\mathcal{I}_u|)^3$ 个阈值电路（因为共有 $O(|\mathcal{I}_u|)$轮迭代，且有 $\binom{|\mathcal{I}_u|}{2}$ 对不同函数）。

现在，对于算法的迭代部分，只需有一个电路，它以列表 $L_{k-1} \in \{0,1\}^{|\mathcal{I}_u|}$ 当前的指示量为输入，并输出 L_k的指示量，以及一个比特，用于指示算法在该轮是否输出失败。注意，也可以假设该电路能够访问该轮的距离图，因为它知道顶点集 L_{k-1}，而距离阈值计算会给出边。因此，只需在该输入图上计算一个极大独立集；根据 Luby [96]的结果，该问题属于 NC^2。要检查是否输出失败，算法只需检查输出的极大独立集的指示量是否与原始输入相等。

上述所有过程都在采样点的定义域上以 NC^2 进行。依次连接每个 $O(|\mathcal{I}_u|)$ 规模对应的电路，即可得到所需的均匀规模和深度。 □

注5.14。 如果可以接受将规模膨胀到$\exp(O(\frac{1}{\varepsilon^2}))$，而不是$\exp(O(\frac{1}{\varepsilon}))$，则可以采用一种更简单的算法：对于输入 $j \le \exp(O(\frac{1}{\varepsilon^2}))$，只需在$(\mathcal{I}_u, 5^j\delta_0)$-距离图中找到一个极大独立集。对于每个 j，该算法都会输出一个极大独立集，使得任意一对函数 $f, f' \in \mathcal{I}_u$ 的距离都不在 $5^j\delta_0$ 和 $5^{j+1}\delta_0$ 之间；这样的集合是 $6^j\delta_0$-稠密的，同时也是 $6^{j+1}\delta_0$-分离的。由于 L 中函数对之间可能存在 $O(\frac{1}{\varepsilon^2})$ 种不同距离，通过允许 $O(\frac{1}{\varepsilon^2})$ 种可能的索引j，我们可以获得较大的常数成功率。由于后文不会用到这一点，我们省略细节，只指出采用此过程的一个小优点是，它降低了电路深度对 ε 的依赖。

5.6 将所有内容整合起来（模电路）

现在我们可以证明定理 5.1（电路复杂度分析留到下一小节）。固定 $\varepsilon > 0$ 和 $N \in \mathbb{N}$。我们取命题 5.4 中保证存在的集合系统 (V, S, U, G)。定理陈述中的参数β, η经过选择，足够小，以便使用本节中的所有断言和引理。所得的码是直积 $\mathcal{C}_{(V,S)}$。超边集的规模为 $\log(N)^{\exp(O(\frac{1}{\varepsilon}))}$，因此局部性和字母表大小可通过直接计算得出。由于 (V, S) 是双正则的，该码的码率就是每个顶点所包含的集合数的倒数，即 $\exp(-O(\frac{1}{\varepsilon}))$。

下面描述该解码器：

1. 我们设置 $\delta_0 = O(\sqrt[3]{\delta_{in}})$。2. 我们使用外层解码器的生成算法，将通往起始集合 u 的列表索引视为 $C_{list-u}(u, i, \cdot)$ 的输入。3. 我们调用外层解码器电路；每当调用一个内层列表 $g_j \in \mathcal{I}_{u'}$ 时，就通过 $C_{list-u}(u', j, \cdot)$ 对其进行模拟。

固定 $f \in \mathcal{B}_{1-\varepsilon,\ \delta_{in}}(w)$。根据断言 5.9 $f \in \mathcal{B}_{1-\frac{\varepsilon}{2},\ 2\delta_{in}}(w_{In_u})$ 对除 $1210\log(N)$ 个 $u \in U$ 之外的所有情况都成立。因此，根据引理 5.6，我们可以找到一个规模为 $< \frac{10}{\varepsilon}$ 的列表 $\mathcal{I}_u$，其中包含某个函数，该函数与 f 的距离不超过 $\delta_0 = O(\sqrt[3]{\delta_{in}})$。也就是说，$f \in \mathcal{B}_{1\log^{10}(N)\delta_0}(\mathcal{I})$。因此，我们可以使用定理 5.10 中的外层解码器，恢复一个与 f 的距离不超过 $\delta_{out} = \exp(102\frac{1}{\varepsilon})\delta_0$ 的函数（根据我们的参数设置，该值小于 0.01）。

最后，对于我们选定的参数：

1. 该路由方案查询由 $\log^3(N)\exp(O(\frac{1}{\varepsilon}))$ 个顶点组成的列表。

2. 为准备每个列表，内部解码器会被实例化 $O(^{\log}_{\log(N)\varepsilon\delta_{in}})$ 次；每次实例化时，C_{list-u} 会查询 $O(^{\log}_{\log(N)\varepsilon\delta_{in}})$ 个点。

3. 内部解码器的每次查询都需要从原始列表 $\mathcal{L}$ 中取出 $\mathrm{poly}(\frac{1}{\varepsilon}, \frac{1}{\delta_{in}})$ 个点。

4. 除列表构造外，外层解码器还会测试 $O(\log^3(N) \cdot \frac{1}{\varepsilon})$ 对函数之间的距离，精度达到 $\geq \frac{1}{\delta_0}$。这总共最多需要 $O(\log^3(N) \cdot \frac{1}{\varepsilon\delta_{in}})$ 个点。

因此，总查询复杂度确实为 $\log^3(N)\exp(O(\frac{1}{\varepsilon}))\,\mathrm{poly}(\log\log(N))$。

5.7 电路实现

要实现外层解码器的电路，我们执行以下步骤。

1. 为了进行路由，我们需要调用 $C^T_{\mathcal{C}}(u, u')$。为此，我们还需要调用 $\mathrm{poly}\log(N)\exp(O(\frac{1}{\varepsilon}))$ 次 C_{list-u}，以实现 T 的预言机。幸运的是，根据断言 5.5，我们可以并行进行所有调用。因此，这一步可以用规模为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$、深度为 $\tilde{O}(\exp(O(\frac{1}{\varepsilon}))\log^2\log N$ 的电路实现。2. 随后，我们需要计算列表 u' 中哪个最终 g_m 最接近函数 f。与定理 3.4 的外层解码器类似，我们可以用 Savitch 算法完成此操作，其规模为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$、深度为 $\tilde{O}(\exp(O(\frac{1}{\varepsilon}))\log^2\log N$。由于其与引理 3.5 中的方法类似，我们省略细节。

总而言之，定理 5.1 的解码器的电路实现规模为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度为 $\tilde{O}(\exp(O(\frac{1}{\varepsilon}))\log^2\log N$。

6 主要构造：KMS 码

在本节中，我们构造命题 4.3 和命题 5.4 中的集合系统。这些系统源自一种称为局部谱扩张器 的高维扩张器，具体而言，源自 [58] Kaufman 和 Oppenheim [59]提出的优雅的陪集复形高维扩展体的一种变体。

路线图 本节安排如下。第 6.1 节介绍高维扩展体的一些预备知识。第 6.2 节介绍一类称为 Kac-Moody-Steinberg 复形（KMS复形）的高维扩展体，它们由 [58], 构造并进行了分析；第 6.3 节讨论其局部结构。第 6.4 节利用这一结构构造（子集内部）局部路由方案。第 6.5 节定义命题 4.3 和命题 5.4 中的集合系统；第 6.6 节则完整证明命题 4.3 和命题 5.4。最后，在第 6.7 节中，我们构造实现 KMS 复形中邻域关系的电路，并给出这些集合系统实现的电路复杂度上界。

6.1 高维扩展图的预备知识

下面我们仅简要介绍满足本文需求的高维扩展体。若想了解更全面的背景，读者可参阅 [93]及其中的参考文献。

单纯复形 一个 d维单纯复形 X 由一个（$d+1$）均匀超图$X(d)$及其向下闭包组成

$$X = X(-1) \cup X(0) \cup \ldots \cup X(d),$$

其中，$X(i) \subseteq \binom{[N]}{i}$称为"$i$-面"或"$(i+1)$-集合"，它们是所有包含于 $X(d)$某个超边中的（$i+1$）规模子集，而 $X(-1) = \{\emptyset\}$。给定一个面 $t \in X(i)$，t 的链接是通过局限于包含 t的面而诱导出的子复形，即 $X_t := \{s \setminus t \in X : s \in X, s \supseteq t\}$。若每个链接 X_t 的基图（其顶点为 $X_t(0)$，边为 $X_t(1)$）均连通，则称 X 连通。包含图（$X(k)$ $X(i)$）是 X 的 k-面与 i-面之间的二部图，其中边由包含关系确定。相关的交换图（或交换游走）（$X(k)$ $X(i)$）连接 X 的 k-集合与 i-集合，当且仅当其不交并$s \bigsqcup t \in X(k+i+1)$时，$s \sim t$。对于 $k < d$，单纯复形 X 的 k-骨架为 $X^{\leq k} = X(-1) \cup X(0) \cup \ldots \cup X(k)$。

分部复形若一个 $(d-1)$ 维复形的顶点可以划分为 d 个**部分**，使得每个顶层面都从每个分量中各取一个部分，则称该复形为分部复形。我们用 $X[1], X[2], ..., X[d] \subseteq X(0)$ 表示各个部分（我们强调，方括号表示不同的顶点部分）。对于一个索引集 $C = \{i_1, i_2, \ldots, i_j\} \subseteq \{1, 2, \ldots, d\}$，我们也用 $X[C] \subseteq X(j-1)$ 表示所有满足 $v_1 \in X[i_1]$，$v_2 \in X[i_2]$,...,$v_j \in X[i_j]$ （次序可变）的 $(j-1)$-面 $\{v_1, v_2, \ldots, v_j\} \in X(j-1)$。我们称这些面为 C-颜色面。对于一个面 t，我们用 $color(t)$ 表示满足 $t \in X[C]$的索引集 C。我们还记 $X^C = \bigcup_{C' \subseteq C} X[C']$。如果包含集合 $t \in X$ 的顶层面数量仅取决于 t 的颜色，则称该分部复形为**正则**。若 X 且其所有链接均为正则，则称 X 为**完全正则**。

有时我们也会考虑彩色交换图（$X[I]$, $X[J]$），它是上述交换图的子图，只包含 $X[I]$ 和 $X[J]$中的顶点。

高维扩展图回顾一下，如果图 G 的归一化随机游走算子的第二大特征值为 $\leq \lambda$，则称 G 是一个 λ-扩展图。归一化邻接算子是 $V \times V$实值矩阵

$$A(v,u) = \begin{cases} \frac{1}{deg(v)} & \{v,u\} \in E \\ 0 & \text{otherwise} \end{cases}.$$

为简单起见，我们仅对完全正则多部复形定义高维扩展图。我们称 X 是一个 λ-高维扩展图，如果对于每个 $t \in X^{\leq d-2}$ 以及每个 i，$j \notin color(t)$，$X_t[i]$ 与 $X_t[j]$ 之间的二部图都是一个 λ-扩展图。[40]

6.1.1 高维扩展体的定理与事实

这里简要介绍一些关于高维扩展体的事实。

[40]即使 X_t 不是完全正则，这仍然是其定义，只不过在这种情况下，我们需要为该图的边赋予权重。

扩张　　包含图和交换图 包含图和交换图是良好的扩展图　　s.

定理 6.1 ([24])。 设 X 是一个 d维的 $\frac{1}{2d}$单侧高维扩展体。则包含图 $(X(k)\ X(i))$ 是一个 $O(\sqrt{\frac{i}{k}})$单侧扩展图。

我们注意到，一般来说，人们不期望得到比这个界更好的结果，因为形如 $\{s \in X(k)\}\, v_0 \in s \subseteq X(k)$ 的集合在包含图中的扩张性不会超过 $\approx \sqrt{\frac{i}{k}}$，但交换图的扩张性确实更好 [57, 97–99]:

定理 6.2。 设 X 是一个 d部 λ单侧高维扩展体。则对于每一对规模分别为 i、k 的 I、$J \subseteq \{1, 2, \ldots, d\}$，交换图 $(X[I], X[J])$ 是一个$(i+k)\lambda$单侧扩展图。

6.2 陪集与 KMS 复形

在本小节中，我们给出 [58], 中所述 KMS 复形的一种实例化，它们是 [59]中复形的一种变体。

KMS 复形是陪集复形。也就是说，它们由某个有限群 G 和子群K_1、K_2、.....、K_d参数化。陪集复形 $X = X(G; K_1、K_2、.....、K_d)$ 的顶点是陪集

$$X(0) = \bigcup_{i=1}^{d} gK_i$$

而顶层集合为

$$X(d-1) = \{\{gK_1, gK_2, \ldots, gK_d\} : g \in G\}.$$

等价地，这些都是满足 $\bigcap_{i=1}^{d} g_i K_i \neq \emptyset$ 的集合 $\{g_1K_1, g_2K_2, \ldots, g_dK_d\}$。可以验证，中间层集合 $X(i)$ 在每个颜色集合中也有类似的描述。也就是说，对于 $C = \{i_1, i_2, \ldots, i_j\}$，$X[C]$ 是$\bigcap_{i\in C} g_i K_i \neq \emptyset$ 的集合。

经典结果表明，该映射

$$\{g_1K_{i_1}, g_2K_{i_2}, \ldots, g_rK_{i_r}\} \in X[C] \longmapsto \bigcap_{i\in C} g_iK_i$$

是从面到 $K_C := \bigcap_{i\in C} K_i$ 的陪集的双射。此外，面 s，$t \in X$ 满足 $s \subseteq t$ 当且仅当 $t \cong gK_{C_1}$，$s \cong g'K_{C_2}$ 使得 $C_1 \subseteq C_2$ 且 $gK_{C_1} \supseteq g'K_{C_2}$。

我们使用的复形基于群 $G = SL_d(R)$，即行列式为 1 的 $d \times d$ 矩阵，其中 $R = \mathbb{F}_q[t]/\phi(t) \cong \mathbb{F}_{q^\kappa}$，也就是说，$\phi(t)$ 是一个 κ 度的不可约多项式。

我们的子群 K_i 由初等幺幂矩阵生成。

定义 6.3（初等幺幂矩阵）。 对于满足 $i \neq j$ 且 $r \in R$ 的 $i, j \in \{1, 2, \ldots, d\}$，令 $e_{i,\,j}(r)$ 表示 G 中的幺幂初等矩阵，使得 r 出现在 (i, j) 项中，且所有其他非对角线项均为零；也就是说，它是一个在 (i, j) 项替换为 r 的单位矩阵。

定义 6.4 (子群 K_i)。 对每个 $i \in \{1, 2, \ldots, d-1\}$，令 K_i 为

$$K_i = \langle \{e_{j,j+1}(a) : j \in [d] \setminus \{i\}\,,\, a \in \mathbb{F}_q\} \cup \{e_{d,1}(at) : a \in \mathbb{F}_q\} \rangle.$$

此外，令

$$K_d = \langle \{e_{j,j+1}(a) : j \in [d-1], a \in \mathbb{F}_q\} \rangle.$$

注意到（例如参见 [58]），这些群都同构，且 K_d 只是对角线上为 1、对角线上方的元素属于域 $\mathbb{F}_q$ 的上三角矩阵（变量 t 不出现）。特别地，每个 $|K_i| = q^{\binom{d}{2}}$，因此 $|X(0)| = dq^{\binom{d}{2}}|SL_d(R)| = q^{\Omega(d^2\kappa)}$。在 [100, 定理 3.6], 中，作者证明这些复形是连通的，并且具有强显式性。不幸的是，他们证明具有强显式性的陪集比特表示存在很大的冗余（即比特长度远大于 $\log |X(0)|$），因此无法用于我们的情形（参见第 3.3.4 节中的讨论）。在第 6.7 节中，我们将展示如何用小型、低深度电路实现这些复形，且冗余可以忽略不计。

定义 6.5. 令 q 为素数幂，d、$\kappa > 0$ 为整数。将复形 $\mathfrak{X} = \mathfrak{X}^{q,d,\kappa}$ 定义为使用上述群和子群构造的陪集复形。

6.3 $\mathfrak{X}$的链接

正如前面所提到的，这些复形是高维扩展图。[41]

引理 6.6 ([100], [58])。 复形 $\mathfrak{X}$ 是一个 $\sqrt[10]{\mathfrak{q}}$高维扩展图。此外，每个顶点至多与 $\leq \mathfrak{q}^{\Theta(d^2)}$ 个最高维面相连，且与 κ无关。

我们可以利用下面这个称为“对置复形”的复形，给出链接的一种非常显式的构造。[42]

固定 $\mathcal{V} = \mathbb{F}_{\mathfrak{q}}^d$，并设 $e_1, e_2, \ldots, e_d$为标准基。对于 $j = 1, 2, \ldots, d-1$，令 $E_j = span\{e_1, ..., e_j\}$。对置复形 P 是一个（$d-1$）分部复形，其顶点为

$$P[i] = \{W \subseteq \mathcal{V} : W \text{ is a linear subspace, } \dim(W) = i, \text{ and } W + E_{d-i} = \mathcal{V}\}.$$

边是满足以下条件的数对 $\{W, W'\}$：经过重排后， $W \subset W'$、 i-面是由 i个子空间构成的嵌套序列：$\{W_1, W_2, \ldots, W_i\}$，其中 $W_1 \subset W_2 ...$在重排后成立；最高维面则由完全旗给出（同样允许重排）：

$$P(d-1) = \{\{W_1, W_2, W_3, \ldots, W_{d-1}\} : W_1 \subset W_2 \subset \cdots \subset W_{d-1}\}.$$

断言 6.7 （[102], 参见 [58, 103]）。 对于每个 $gK_i \in \mathfrak{X}$（0），链接 $\mathfrak{X}_{gK_i}$ 与对置复形 P同构。此外，在此同构下，部分 $\mathfrak{X}_{gK_i}[j]$ 被映射到 $P[j-i]$ （模 $d-1$）。

我们还注意到，对置复形可以上下翻转，也就是说，

断言 6.8。 存在一个 P的自同构，它将 $P[i]$ 映射到 $P[d-i]$，反之亦然 .

证明。 首先注意，在 P 的定义中，标准基的选取是任意的；事实上，我们可以使用任意有序基，并得到一个同构的复形。我们将描述一个 P 的自同构 ϕ，它将 $\mathcal{V}$的子空间映射到 $\mathcal{V}$的子空间。该自同构将 $W \in P[i]$映射到 ϕ（W） $\in P[d-i]$；对于每个 ~ ~$W \subseteq \mathcal{V}$，当且仅当 $W + E_{d-i} = V$（W） $+ E_i = \mathcal{V}$时， ϕ成立，其中 E_i经过适当定义。

我们先描述这个自同构。对于向量 $v = (v_1, v_2, \ldots, v_d) \in \mathbb{F}_{\mathfrak{q}}^d$，令 $v^{\perp}$ 为线性泛函 $v^{\perp}(u_1, u_2, \ldots, u_d)$ $= v_1u_1 + v_2u_2 + \cdots + v_du_d$。对于任意子空间 $W \subseteq V$，令 $\phi(W) = \{v \in \mathcal{V} : \forall w \in W, w^{\perp}(v) = 0\}$。注意，$\dim(\phi(W))$ $= d - dim(W)$（固定从 W 到 $w_1, ...w_i$ 的一组基后，方程 $w_j^{\perp}(v) = 0$ 是线性无关方程。这些方程的零空间恰好是 $\phi(W)$）。 ~现在取 $f_1 = e_d, f_2 = e_{d-1}, \ldots, f_{d-1} = e_2, f_d = e_1$，并令 $E_i = span(f_1, f_2, \ldots, f_i)$。可以验证 ~ ~， $\phi(E_i) = E_{d-i}$。我们断言， $W + E_{d-i} = \mathcal{V}$ 当且仅当 $\phi(W)$ $+ E_i = \mathcal{V}$。实际上，这等价于~ ~证明 $W + E_{d-i} = \mathcal{V}$ 当且仅当 $\phi(W)$ $\cap E_i = \{0\}$ （因为 $\dim(\phi(W) + E_i) = d - \dim(\phi(W) \cap E_i)$）。确实， $W + E_{d-i} = V$ 当且仅当 $\phi(W + E_i) = \{0\}$。根据 $W \cup E_i$ 张成 $W + E_i$这一事实，~ $\phi(W + E_i) = \phi(W)$ $\cap \phi(E_i) = \phi(W) + E_i$，因此断言得证。 □

由最后一个断言可直接得到以下推论，因此省略其证明。

推论 6.9. 对于每个 $gK_i \in \mathfrak{X}(0)$，链接 $\mathfrak{X}_{gK_i}$ 与对置复形 P同构。此外，在此同构下，部分 $\mathfrak{X}_{gK_i}[j]$ 被映射到 $P[d-j+i]$ （模 $d-1$）。

最后，我们得到以下一般性的对称性保证。

断言 6.10. 设 C_1、 $C_2 \subseteq [d]$ 是两个颜色集，使得循环 $\phi : [d] \to [d]$存在一个同构，该同构将 $\phi(C_1) = C_2$ 映射到（即，将 $\phi(x) = \pm x + c$ 模 d映射到）某个 $c \in [d]$。那么对于每个 $x \in \mathfrak{X}[C_1]$ 和$y \in \mathfrak{X}[C_2]$，都存在一个同构，使得对于每种颜色 $C \subseteq [d]$， $\mathfrak{X}_x[C]$ 被映射到 $\mathfrak{X}_y[\phi(C)]$。

证明。 如果 C_1 和 C_2 是单元素集，则结论直接由断言 6.7 和推论 6.9 得出。否则，写作$C_1 = \{i\} \cup C_1'$ 和 $C_2 = \{i'\} \cup C_2'$，其中 $i' = \phi(i)$。取 $x = \{x_1\} \cup x' \in \mathfrak{X}[C_1]$，使得 $x \in \mathfrak{X}[i]$ 且$x' \in \mathfrak{X}[C_1']$。令 $y_1 \in \mathfrak{X}[i']$，根据关于单元素集的断言，存在一个从 $\mathfrak{X}_{x_1}$ 到 $\mathfrak{X}_{y_1}$ 的同构，将 C_1' 映射到 $\phi(C_1') = C_2'$。特别地，集合 $x' \mapsto y'$ 对应某个 $y' \in \mathfrak{X}[C_2']$。此外，这是一个复形的同构，因此链路 $\mathfrak{X}_x = (\mathfrak{X}_{x_1})_{x'} \cong (\mathfrak{X}_{y_1})$ $_{y'} = \mathfrak{X}_y$ 也由这同一个同构得到。断言得证。 □

[41]严格来说，由于局部谱扩张的概念略有不同，这个引理使用了 [57] 中的一些标准工具来在两者之间转换，但我们省略细节。[101]中给出了明确的转换。[42]事实上， [102] 将对置复形定义为一种更一般的复形，但我们将只用“对置复形”这一名称来指代我们所描述的复形。

6.4 路由与 KMS 复形

在本小节中，我们将为这个复形建立一个路由方案，进而得到命题 4.3 和命题 5.4 中的路由方案。思路如下。我们首先为两个不相交颜色 $\mathfrak{X}[C_1]$ 和 $\mathfrak{X}[C_2]$之间的交换游走图找到一个确定性路由方案。然后，我们将利用群 $SL_d(\mathbb{F}_\mathfrak{q})$作用在该复形上这一事实，以及该作用对于$\mathfrak{X}[C_1]$ 和 $\mathfrak{X}[C_2]$之间的边是边传递的这一性质。我们将利用该群对路由方案充分随机化，使其具有低拥塞。

在最后一步中，我们还将给出一个 $\mathfrak{X}[C_1], \mathfrak{X}[C_2]$ 上的子集内部路由方案。该方案以某个较大 $T \subseteq \mathfrak{X}[C_1] \cup \mathfrak{X}[C_2]$上的成员资格预言机为输入，并为大多数 s_1, s_2 生成一条路径，使路径上的所有中间顶点都属于 T。此方案还将具有低拥塞，这对定理 5.1 至关重要。

6.4.1 一个确定性路由方案

我们首先设计一种确定性路由算法，用于连接任意两个 $\mathfrak{X}[C_1], \mathfrak{X}[C_2]$ ，且它们的颜色C_1, C_2 互不相交。

回顾一下，对于每种颜色 $C \subseteq [d]$， $\mathfrak{X}[C]$ 的面可以视为 $K_C = \bigcap_{i \in C} K_i$ 的陪集，因为每个$s \in \mathfrak{X}[C]$ 都等于某个 $s = \{gK_i\}_{i \in C}$ ，而且我们可以将其视为 gK_C ；后者是 s 中所有顶点的公共交集（形式化证明可由定义直接得出，因此省略）。由于$s_1 \in \mathfrak{X}[C_1], s_2 \in \mathfrak{X}[C_2]$ 连通当且仅当 $s_1 \cup s_2 \in \mathfrak{X}[C_1 \cup C_2]$，我们也可以将交换游走图中的边视为 $gK_{C_1 \cup C_2}$。

断言 6.11 （邻域结构）**.** 固定互不相交的 C_1、$C_2 \subseteq [d]$ 。陪集 $gK_{C_1} \in \mathfrak{X}[C_1]$ 的 C_2-邻域是如下集合：

$$\{ghK_{C_2} : h \in K_{C_1}\}.$$

注 6.12. 鉴于该断言，寻找一条从（例如） AK_{C_1} 到 BK_{C_2}、长度为 $\leq t$ 的路径这一问题，可归结为寻找 $A^{-1}B = h_1 h_2 \dots h_{t-1}$的一个分解，使得 $h_{2k+1} \in K_{C_1}$与 $h_{2k} \in K_{C_2}$（当然，其中一些可以是单位元），因为我们可以取如下形式的路径

$$AK_{C_1}, Ah_1K_{C_2}, Ah_1h_2K_{C_1}, \dots, Ah_1h_2 \dots h_{t-1}K_{C_x} = BK_{C_x}$$

然后，若 $K_{C_x} = K_{C_2}$ ，则我们完成；否则，我们只需从 BK_{C_1} 走到 BK_{C_2}作为最后一步。为完整起见，我们来证明该断言。

断言 *6.11* 的证明. gK_{C_1} 的邻居是所有满足 $gK_{C_1} \cap g'K_{C_2} \neq \emptyset$的 $g'K_{C_2}$。显然，若对 $h \in K_{C_1}$ 有 $g' = gh$，则交集非空。反之，若交集包含某个元素 g'' ，则由 gK_{C_1}的定义，对某个 $h \in K_{C_1}$有 $g'' = gh$ ，因此 $g'K_{C_2} = g''K_{C_2} = ghK_{C_2}$。 □

在我们的图中，我们总能走出短步，这些步对应于生成子群的初等矩阵。

推论 6.13。 设 $e_{j,j+1}(r)$ 是子群定义中的某个生成元 $K_1, K_2, \dots, K_d$ （即，$r \in \mathbb{F}_\mathfrak{q}$ 当且仅当 $j \neq d$ ，且 $r = r't$ 当且仅当 $r' \in \mathbb{F}_\mathfrak{q}$ ）。那么对于任意不相交的 $C_1, C_2 \subseteq [d]$，以及任意陪集 gK_{C_1}，存在一条长度为 ≤ 2 的路径，从 gK_{C_1} 到 $ge_{j,j+1}(r)K_{C_1}$。

证明。 回想一下，颜色会告诉我们子群中缺少哪些生成元。注意， $C_1 \cap C_2 = \emptyset$，因此 j 不可能同时属于两个颜色集合，这意味着形如 $e_{j,j+1}(r)$ 的矩阵必定属于子群 K_{C_1}, K_{C_2}之一。如果 $j \notin C_1$ ，那么 $e_{j,j+1}(r) \in K_{C_1}$ ，接着 $gK_{C_1} = ge_{j,j+1}(r)K_{C_1}$，因此严格来说，从该陪集到自身存在一条长度为 0 的路径。否则 $j \notin K_{C_2}$ ，我们可以沿着 $(gK_{C_1}, gK_{C_2}, ge_{j,j+1}(r)K_{C_1})$ 行进，因为 $e_{j,j+1}(r) \in K_{C_2}$。 □

引理 6.14 (路由方案**)**。 设 C_1、$C_2 \subseteq [d]$ 是两个不相交的颜色集。那么， $\mathfrak{X}[C_1] = \{gK_{C_1}\}_{g \in G}$ 与 $\mathfrak{X}[C_2] = \{gK_{C_2}\}_{g \in G}$ 之间的交换游走图存在一个确定性路由方案 $\mathfrak{X}$。路径的最大长度为 $O(\mathrm{poly}(d)\kappa^{\log_2 5})$。

证明。 要为每一对顶点构造一条路径，只需给出一条从陪集（顶点）K_{C_1} 到 $\mathfrak{x}[C_1]$ 或 $\mathfrak{x}[C_2]$中每个顶点的路径。于是，要从任意一个 AK_{C_i} 到达 BK_{C_j}，只需经由 K_{C_1}连接它们各自的路径即可。此外，请注意，如果我们有一条长度为 t 的路径从 K_{C_1} 到 AK_{C_1}，那么由于 $AK_{C_1} \sim AK_{C_2}$，我们也有一条长度为 $t+1$ 的路径到达 AK_{C_2}。因此，只需为任意 $A \in SL_d(R)$构造一条从 K_{C_1} 到AK_{C_1} 的路径。从现在起，固定 $A \in SL_d(R)$。

现在根据注 6.12 和推论 6.13，只需将 $A$to $O(\text{poly}(d)\kappa^5)$ 生成元分解，即表示为 $O(\text{poly}(d)\kappa^5)$ 初等矩阵的乘积$e_{j,\ j+1}(r)$，如子群定义中所述。

已知每个 $A \in SL_d(R)$ 都是 $O(d^2)$ 初等幺幂矩阵的乘积[43] $e_{i,\ j}(r)$，因此只需在假设 $A = e_{i,\ j}(r)$ 的情况下，找到这样的分解，其中 $i,\ j$ 且 $r \in R$。事实上，我们断言，只需证明 r 是 $r = t \cdot r'(t)$ 这种形式即可，也就是说，只需考虑没有常数项的多项式。原因如下。假设我们知道如何对每个形如 $r = tr'(t)$ 的多项式，将 $e_{i,\ j}(r)$ 写成这种形式。由于我们是在 $\mathbb{F}_{\mathfrak{q}}[t]/(\phi(t))$ 上而不是在 $\mathbb{F}_{\mathfrak{q}}[t]$ 上运算，这意味着 $\phi(t) \equiv 0$ 在我们的环中。因此，由于这个多项式不可约，这意味着 $\phi(0) \neq 0$，特别是它有非零常数项。因此，如果 $\phi(t) = \sum_{j=0}^{\kappa} \alpha_j t^j$，那么

$$1 = -t(\sum_{j=1}^{\kappa} \frac{\alpha_j}{\alpha_0} t^{j-1}).$$

这意味着，写下 $e_{i,j}(t(\sum_{j=1}^{\kappa} \frac{\alpha_j}{\alpha_0} t^{j-1}))$ 实际上就是写下一个常数。当然，对于任何其他带有常数项 $q(t) = a + tq'(t)$ 的多项式，我们只需写出该多项式

$$t \cdot a(\sum_{j=1}^{\kappa} \frac{\alpha_j}{\alpha_0} t^{j-1}) + tq'(t)$$

它与之等价。因此，如果我们能写出任意形如 $tr'(t)$ 的多项式，我们也能写出 R.[44] 中的所有多项式。

现在我们对 $\deg(r')$ 使用归纳法。令 $T(m)$ 表示从 $K_{C_1} \cup K_{C_2}$ 中取出元素来写出 tr'，且满足 $\deg(r') \leq m$ 时所需元素的最大数量。

回顾以下可轻松验证的公式：

1. 对于任意 i, j 和 $r_1, r_2 \in R$

$$e_{i,j}(r_1)e_{i,j}(r_2) = e_{i,j}(r_1 + r_2) \tag{6}$$

2. 对于任意三个互不相同的 i, j, ℓ 和 $r_1, r_2 \in R$

$$[e_{i,\ell}(r_1), e_{\ell,j}(r_2)] = e_{i,j}(r_1 \cdot r_2) \tag{7}$$

其中方括号表示交换子 $[g,h] = ghg^{-1}h^{-1}$

让我们进行归纳步骤，证明 $T(2^m) \leq 5T(2^{m-1})$。接下来，我们通过一些繁琐的计算证明 $T(1) = O(d)$。求解递推关系可得 $T(\kappa) \leq O(d\kappa^{\log_2 5})$，从而证明该断言。设 r' 是一个度为 $\leq 2^m$ 的多项式。我们可以写成 $r' = t^{2^{m-1}} r_1 + r_2$，其中 r_1, r_2 的度至多为 2^{m-1}（显然 $t^{2^{m-1}}$ 的度也为 2^{m-1}）。因此 $e_{i,j}(tr') = [e_{i,\ell}(tr_1)\, e_{\ell,j}(t^{2^{i-1}})]e_{i,\ell}(tr_2)$。该公式涉及 5 次初等矩阵的乘法，这些初等矩阵的非零元素的度为 $\leq 2^{m-1}$，因此 $T(2^m) \leq 5T(2^{m-1})$。

因此，只需对 $T(1)$ 进行界定，或者等价地证明每个 $e_{i,j}(\alpha t + \beta t^2)$ 都可以表示为 $O(d)$ 个 $K_{C_1} \cup K_{C_2}$ 中的元素。根据上述加法公式，只需能够表示 $e_{i,j}(\alpha t)$ 和 $e_{i,j}(\beta t^2)$。此外，如果我们能够对每个 i, j 表示 $e_{i,j}(\alpha t)$，那么根据上述乘法公式，$e_{i,j}(\beta t^2) = [e_{i,\ell}(t)e_{\ell,j}(\beta t)]$。因此，实际上只需证明对于任意 $i \neq j$，我们都能表示 $e_{i,j}(\alpha t)$。

W现在我们分以下几种情况讨论：$i > j$、$i < j < d$ 和 $i < j = d$。根据（7），对于任意 $i > j$

$$e_{i,j}(\alpha t) = [\dots[[e_{i,i+1}(p_i)e_{i+1,i+2}(p_{i+1})], e_{i+2,i+3}(p_{i+2})], \dots], e_{j-1,j}(p_{j-1})]$$

其中:

[43]确实，这只是高斯消元。[44]由于实际上对于任意 $i < j \neq d$，我们都有 $e_{i,j}(\alpha)$，其中 $\alpha \in \mathbb{F}_{\mathfrak{q}}$ 为常数，因此这似乎过于复杂。然而，要使用这些生成元得到矩阵 $e_{i,d}(\alpha)$，我们实际上需要使用一个次数很高的多项式。

1. 上述索引应按模 d 取。
2. $p_\ell = \begin{cases} \alpha t & \ell = d \\ 1 & o.w. \end{cases}$。注意，由于 $i > j$，我们确实在某个时刻到达了 $\ell = d$。

这使得我们在 $i > j$ 时，只需进行 $O(d)$ 次群乘法，就能得到 $e_{i,j}(\alpha t)$。注意，根据类似的公式，对于任意 $i < j$，我们都可以写出 $e_{i,j}(1)$（包括 $j = d$）。因此，对于 $i < j < d$，我们可以写出 $e_{i,j}(\alpha t) = [e_{i,d}(1)e_{d,j}(\alpha t)]$。最后一种情形是 $e_{i,d}(\alpha t)$，它等于 $[e_{i,\ell}(\alpha t)e_{\ell,d}(1)]$，其中 $i \neq \ell,\ d$（同样，我们使用前面几种情形来表示）。引理得证。

□

6.4.2 从确定性路由到低拥塞路由

注意，本小节适用于任意陪集复形，并未使用 $SL_d(R)$ 及其子群的任何特定性质。

设 X 是关于群 Γ 及子群 $K_1, K_2, \ldots, K$ 的陪集复形 $_d$.

断言 6.15 (Γ 的传递性)。群 Γ 在 X 的顶层面上具有传递作用，更一般地，它在任意面的颜色集上也具有传递作用。该作用由 $h \cdot gK_i \mapsto hgK_i$ 给出。因此，它也作用于任意两个不相交颜色 $X[C_1]$、$X[C_2]$ 之间的交换游走图，并且在该图上边传递。集合的颜色在此作用下保持不变。

这一断言的证明只需追溯定义，因此略去。

设 G 是一个二部图，且群 Γ 在该图上作用，并保持左右两侧不变。建议读者思考 $\Gamma = SL_d(R)$ 以及 G 在 $\mathfrak{X}[C_1]$ 和 $\mathfrak{X}[C_2]$ 之间的交换游走图。

对于路径 $P = (u_0, u_1, \ldots, u_m)$ 和 $g \in \Gamma$，我们用 $g.P = (g.u_0, g.u_1, \ldots, g.u_m)$ 表示。

对于一个路由方案 $\mathcal{O}$ 以及作用于图上的群 Γ，我们考虑由下式定义的随机路由方案 $\mathcal{O}^\Gamma$：

1. 输入：一个顶点有序对（$v,\ v'$）。
2. 输出：一条从 v 到 v'的路径 P。
3. 算法：

   (a) 均匀随机地选择一个 $g \in \Gamma$。(b) 获取一条从 $g^{-1}v$ 到 $g^{-1}v'$的路径 $P_0 = \mathcal{O}(g^{-1}v,\ g^{-1}v')$。(c) 输出 $P = g.P_0$。

断言 6.16。路由方案 $\mathcal{O}^\Gamma$ 具有单位拥塞。

证明。令 t 为 $\mathcal{O}^\Gamma$的长度。固定任意边子集 B。我们需要界定概率$_{v\ ,\ v'\mathcal{O}^\Gamma}\left[\exists e \in \mathcal{O}^\Gamma(v,\ v') :\ e \in B\right]$。显然，这个概率不超过概率 $^t{}_{v\ ,\ v'\mathcal{O}^\Gamma,\ e}[e \in B]$，其中 e 是从 $\mathcal{O}^\Gamma(v,\ v')$ 的输出中均匀随机选取的。

此实验中的随机性来自 $v,\ v' \in V$、$P_0 \sim \mathcal{O}(v,\ v')$（如果 $\mathcal{O}$ 本身是随机的，则也包括其随机性）、$g \in \Gamma$以及 $i \leq |P_0|$（其中 e 是 P_0中的第 i 条边）。我们断言，对于 $u = g^{-1}v,\ u' = g^{-1}v'$、$P_0$ 和 i的任意固定取值，由 $\mathcal{O}^\Gamma$ 选出的元素 $g \in \Gamma$ 仍然服从均匀分布。注意，如果这一点成立，那么根据边传递性，P_0中作为第 i 条边的任意 e 都在所有边上均匀分布，因此断言成立，因为$_{v\ \ ,\ v'\mathcal{O}^\Gamma,\ e}[e \in B] = \mathbb{P}[B]$。

固定任意 $g_0, g_1 \in \Gamma$。我们想证明，对于上文所述的任意固定 u, u'

$$\Pr_{g,v,v'}[g = g_0 \mid g^{-1}.v = u, g^{-1}v' = u'] = \Pr_{g,v,v'}[g = g_1 \mid g^{-1}.v = u, g^{-1}v' = u']$$

（注意， $\mathcal{O}$ 的随机性与 i 的选择相互独立，因此我们已将它们从概率中移除。）观察左侧可得

$$\begin{aligned}\Pr_{g,v,v'}[g=g_0 \mid g^{-1}.v=u, g^{-1}v'=u'] &= \Pr_{g,v,v'}[g=g_0 \mid v=g.u, v'=g.u'] \\ &= \Pr_{g,v,v'}[v=g.u, v'=g.u' \mid g=g_0]\frac{\Pr_{g,v,v'}[g=g_0]}{\Pr_{g,v,v'}[v=g.u, v'=g.u']} \\ &= \frac{\frac{1}{|V|^2}\cdot\frac{1}{\Gamma}}{\Pr_{g,v,v'}[v=g.u, v'=g.u']}\end{aligned}$$

这里我们使用了贝叶斯法则，最后一个等号则是因为：即使在第一个概率中固定了 g_0， v 和 v' 仍是从 V 中独立同分布抽取的随机样本，因此它们分别为 $g_0.u$ 和 $g_0.u'$ 的概率为 $\frac{1}{|V|}$，因为这些只是图中的固定顶点。分母 $\Pr_{g,v,v'}[v=g.u, v'=g.u']$ 恰好等于

$$\sum_{g_0\in\Gamma}\mathop{\mathbb{P}}_{g,v,v'}[g=g_0]\Pr_{g,v,v'}[v=g.u, v'=g.u' \mid g=g_0] = \frac{1}{|V|^2}$$

根据上述结果，这恰好等于 $\frac{1}{|\Gamma|}$，且与 g_0无关。 □

现在我们得到了针对 $\mathfrak{X}$的主要路由方案。

推论 6.17。 对于任意互不相交的 C_1、 $C_2 \subseteq [d]$， $\mathfrak{X}[C_1]$ 与 $\mathfrak{X}[C_2]$之间的交换游走图具有一个长度为 $O(\mathrm{poly}(d)\kappa^{\log_2 5})$ 且拥塞为单位的路由方案。

6.4.3 子集内部路由

现在我们转而证明，交换图在 $\mathfrak{X}$ 上的一种特殊情形具有子集内部低拥塞路由方案；最终，我们将对其进行实例化，以得到命题 5.4 所需的相应路由方案。

设 $C \subseteq [d]$。我们称 $C' \subseteq [d]$ 分离 C，如果 $C = \{i_1 < i_2 < \cdots < i_j\}$ 满足：对于每个 $\ell = 1$、2、.....、$j-1$，都有 $C' \cap (i_\ell < i_{\ell+1}) \neq \emptyset$；此外， $C' \cap [1, i_1) \cup (i_j, d] = \emptyset$。换句话说，如果按显而易见的方式将$C$ 和 C' 的索引嵌入到 d-循环上，那么 C 的任意两个索引之间都由 C'的一个索引隔开。我们称C 是可分离的，如果存在 C' 分离 C。我们称 C 是双可分离的，如果存在两个不相交的C'、 C'' 分离 C。

断言 6.18。 设 C_1、 $C_2 \subseteq [d]$，并设 $C' \subseteq [d]$ 分离 $C_1 \cup C_2$。那么对于任意 $s \in \mathfrak{X}[C']$， $\mathfrak{X}_s[C_1]$、 $\mathfrak{X}_s[C_2]$ 之间的交换游走图是一个完全多部图。

证明。 令 $s \in \mathfrak{X}[C']$，并写作 $s = v_1 \cup s'$，其中 $v_1 \notin s'$。链接 $\mathfrak{X}_s = (\mathfrak{X}_{v_1})_{s'}$。然而， $\mathfrak{X}_{v_1}$ 是对置复形，而分离性意味着，对于任意两个颜色 $C_1 \cup C_2$，其间都存在某个颜色 $i \in C'$。我们将 $\mathfrak{X}_{v_1}$ 中的顶点视为子空间，将 $\mathfrak{X}[C_i]$ 中的面视为子空间旗。这意味着，对于任意两个面 $t_1 \in \mathfrak{X}_s[C_1]$， $t_2 \in \mathfrak{X}_s[C_2]$以及任意两个子空间 $u_1 \in t_1$， $u_2 \in t_2$，根据对置复形中链接的定义，存在一个子空间 $w \in s'$，使得 $u_1 \subset w \subset u_2$ （反之亦然）。这尤其意味着，每一对这样的元素 t_1， t_2 的并都是一个旗，因此这是一个完全二部图。

□

命题 6.19. 设 C_1 和 C_2 满足 $C_1 \cup C_2$ 是双可分离的。那么，交换游走图 $(\mathfrak{X}[C_1], \mathfrak{X}[C_2])$ 存在一个成功的 子集内部路由方案 $\mathcal{O}^T(v, v')$，并具有以下性质：

1. 对于 T，它向成员资格预言机发起的查询不超过 $O(\mathrm{poly}(d)\kappa \log\log N)$ 次。

2. 最大路径长度为 $t = O(\mathrm{poly}(d)\kappa^{\log_2 5})$。

3. 它具有低拥塞。

下面介绍路由算法。设 C_3 和 C_4 是不相交颜色，它们将 $C_1 \cup C_2$分开。为简单起见，我们只描述两个顶点具有相同颜色 $v, v' \in \mathfrak{X}[C_1]$ 的情形（其他情形的描述类似，但索引会变得复杂）。

T中的路由算法:

*** 输入:** 一对 $v, v' \in \mathfrak{X}[C_1]$。

*** 输出:** 交换游走图中从 v 到 v'、连接 $\mathfrak{X}[C_1]$ 与 $\mathfrak{X}[C_2]$ 的一条路径。

1. 抽样 $u \in \mathfrak{X}_v[C_3]$ 和 $u' \in \mathfrak{X}_{v'}[C_3]$。

2. 根据推论 6.17 中对 $\mathfrak{X}[C_3]$ 和 $\mathfrak{X}[C_4]$ 从 u 和 u' 进行路由的保证，调用路由过程，得到路径$P_0 =($ $u = u_0,\ u_1,\ u_2,\ \ldots,\ u_m = u')$。

3. 将 $v_0 = v,\ v_{2m} = v'$记为。对于 $j = 1,\ 2,\ \ldots,\ 2m-1$，我们按如下方式构造路径 $(v_0,\ v_1,\ v_2,\ \ldots,\ v_{2m})$。

- 如果 j 是偶数，则均匀随机选择 $v_j \in \mathfrak{X}_{u_j}[C_1] \cap T$。
- 如果 j 为奇数，则均匀随机选择 $v_j \in \mathfrak{X}_{\{u_{j-1},u_j\}}[C_2] \cap T$。

在这两种情况下，选择均通过拒绝采样完成：对于每个抽样得到的 v_j，我们查询它是否属于 T

如果在抽取 100 log log N个样本后仍未找到 $v \in T$，则中止并输出 $FAIL$。

4. 输出（$v_0,\ v_1,\ \ldots\ldots,\ v_{2m}$）。

图 7: $\mathfrak{X}[C_1] \cup \mathfrak{X}[C_2]$中的路由

证明。 我们首先证明路由是成功的，即输出 $FAIL$ 的概率被 $1_{\log^4 |V|}$所界定。我们注意到以下事实。

1. $\mathbb{P}_{s\in\mathfrak{X}[C_1]\cup\mathfrak{X}[C_2]}[T|col(s) = C_1] \geq 0.8$，类似地，在以 $col(s) = C_2$为条件时也成立。2. $\mathfrak{X}[C_3]$ 与 $\mathfrak{X}[C_1]$ 之间的交换游走图是一个 $\frac{10d}{\sqrt{q}}$-扩展图。因此，$s \in \mathfrak{X}[C_3]$ 具有该 $\mathbb{P}_{s'\in\mathfrak{X}_s[C_1]}[s \in T] < 0.75$ 的概率为 $O(\frac{d^2}{q})$。
3. 上一条对所有数对 $(\mathfrak{X}[C_3],\ \mathfrak{X}[C_2])$、$(\mathfrak{X}[C_4],\ \mathfrak{X}[C_1])$、$(\mathfrak{X}[\overline{C_4}],\ \mathfrak{X}[C_2])$ 也成立；对于一侧同时具有两种颜色 $C_3 \cup C_4$的数对也成立：$(\mathfrak{X}[C_3 \cup C_4],\ \mathfrak{X}[C_1])$、$\overline{(\mathfrak{X}[C_3 \cup C_4],\ \mathfrak{X}[C_2])}$。

根据以上内容，面 s在 $\mathfrak{X}[C_3],\ \mathfrak{X}[C_4]$或 $\mathfrak{X}[C_3 \cup C_4]$ 中具有 $\mathbb{P}_{s'\in\mathfrak{X}_s[C_1]}[s \in T] < 0.75$（或 $\mathbb{P}_{s'\in\mathfrak{X}_s[C_2]}[s \in T] < 0.75$) 为 $O(\frac{d^2}{q})$。注意，当 v, v'独立且均匀抽样时，u, u'（算法中 P_0 的端点）也独立且均匀。因此，由于路由方案在 $\mathfrak{X}[C_3], \mathfrak{X}[C_4]$上具有单位拥塞，路径 P_0 中任意边或顶点满足此性质的概率至多为 $\frac{3}{N} < \frac{1}{2^{\log^4 N}}$。如果对于每个这样的 $s = u_j$ 和 $s = \{u_j, u_{j+1}\}$，T 位于每一侧的概率至少为 0.75，那么在抽样 100 log log N 次后，未能从 s 的链接中的 T 抽到顶点的概率为 $< \frac{1}{\log^4 N}$。因此，路由成功。

现在我们开始证明所声称的三个结论。事实上，前两个性质可直接由算法的定义以及推论 6.17 中路由方案的路径最大长度得出。至于最后一个性质，设 B 为边的一个子集。

在界定 B 的概率之前，我们先为可能的边 $e = \{u_j, u_{j+1}\}$ 定义三个"坏事件"。界定这些坏事件后，我们将界定在这些事件未发生时采样 B 的概率:

1. The density of T inside $\mathfrak{X}_{u_j}[C_1]$ or $\mathfrak{X}_{\{u_j,u_{j+1}\}}[C_2]$ is less than 0.75.
2. $\mathfrak{X}_{u_j}$ 的链接在 B中有超过 $\mathbb{P}[B]^{1/2}$ 条边。

3. $\mathfrak{x}_{u_j}$的链接在 B中的边少于 $\mathbb{P}[B]^{1/2}$条，但 $\mathfrak{x}_{u_j}[C_1]$与 $\mathfrak{x}_{\{u_j,u_{j+1}\}}[C_2]$之间的图在 B中的边超过 $\mathbb{P}[B]^{1/4}$ 条。

当一条边被均匀随机采样时，第一项以概率 $< O(\frac{d}{\mathfrak{q}})$ 发生；特别地，由于我们将 $\mathfrak{q}$ 取得足够大，它发生的概率小于 $2t^{\frac{1}{\log^4|V|}}$。

由马尔可夫不等式，第二个事件发生的概率为 $\mathbb{P}[B]^{1/2}$。此外，对于任意其链接在 B 中少于 $\mathbb{P}[B]^{1/2}$ 条边的 $\mathfrak{x}_{u_j}$，使得第三个事件发生的 u_{j+1} 所占比例同样至多为 $\mathbb{P}[B]^{1/4}$（同样由马尔可夫不等式得出；再次强调，以上均是在边被均匀随机采样时）。

将这些坏边记为 R。到目前为止，我们仅在均匀随机采样一条边时给出了这些事件的上界。然而请注意，我们所使用的内部路由具有单位拥塞，因此我们碰到这样一条坏边的概率以 $t\mathbb{P}[R]$ 为上界（其中 e 是交换游走中的一条随机边）。该概率至多为 $e^{2t\mathbb{P}[B]^{1/4}}+2^{\frac{1}{\log^4|V|}}$。

如前所述，输出路径中的某条边命中边集 B 的概率为$\leq t\mathbb{P}_{v\,v',,e\in\mathcal{O}^T(v,v')}[B]$。假设路径也不在 R 中时，同样成立。因此，为了得到所需的拥塞，我们需要证明$\mathbb{P}_{v\,v',,e\in\mathcal{O}^T(v,v')}\left[B\text{ 和 }\mathcal{O}^T(v,v')\text{ 不会命中 }R\right]\leq\mathbb{P}[B]^{1/4}$，因为:

$$\begin{aligned}
&\mathbb{P}_{v,v',\mathcal{O}^T(v,v')}\left[\exists e\in\mathcal{O}^T(v,v')\ e\in B\right]\\
&\qquad\leq\mathbb{P}\left[\mathcal{O}^T(v,v')\text{ hits }R\right]+t\,\mathbb{P}_{v,v',e\in\mathcal{O}^T(v,v')}\left[B\text{ and }\mathcal{O}^T(v,v')\text{ doesn't hit }R\right]\\
&\qquad\leq\frac{1}{2\log^4|V|}+2t\mathbb{P}[B]^{1/4}+t\,\mathbb{P}_{v,v',e\in\mathcal{O}^T(v,v')}\left[B\text{ and }\mathcal{O}^T(v,v')\text{ doesn't hit }R\right].
\end{aligned}$$

注意，我们可以先在 $\mathcal{O}^T(v,v')$ 中随机选一条边 $\{u_j,u_{j+1}\}\in P_0$，再将边设为 $e=\{v_j,v_{j+1}\}$，从而选出一条随机边。我们还注意到，这是一条连接 $\mathfrak{x}_{u_j}[C_1]\cap T$ 和 $\mathfrak{x}_{\{u_j,u_{j+1}\}}[C_2]\cap T$ 的随机边。

如果 $e\notin R$，那么根据选择一条边等同于选择一对随机顶点（其中一个位于顶点 u_j 的链接中，另一个位于 $\{u_j,u_{j+1}\}$的链接中）这一事实，$T\times T$ 中连接 $\mathfrak{x}_{u_j}[C_1]$ 和 $\mathfrak{x}_{\{u_j,u_{j+1}\}}[C_2]$ 的边的比例至多为 $0.75^2\geq0.5$。此外，B 中一条边出现的概率至多为$\mathbb{P}[B]^{1/4}$，因此即使在边连接 T 中端点的条件下，该概率也至多为 $2\mathbb{P}[B]^{1/4}$。因此，选择一条位于$\mathbb{P}_{v\,v',,e\in\mathcal{O}^T(v,v')}\left[B\text{ 且 }\mathcal{O}^T(v,v')\text{ 不会命中 }R\right]$ 的边的概率至多为$2t\mathbb{P}[B]^{1/4}$.

□

注释 6.20。有人可能会问，为什么在偶数步中，我们直接从 $\mathfrak{x}_{\{u_{j-1},u_j\}}$ 中的一个面走到$\mathfrak{x}_{u_j}$ 中的一个面，而不是走到 $\mathfrak{x}_{\{u_j,u_{j+1}\}}$中的一个面。如果我们这样做，那么第三个"坏事件"就会取决于三个顶点 u_{j-1}、u_j、u_{j+1}，而不仅仅取决于一条边。虽然我们的路由方案具有单位拥塞，但我们并未证明关于边对上的事件有任何保证。

6.5 从 $\mathfrak{x}$产生的集合系统

我们首先介绍命题 4.3 和命题 5.4 中的集合系统

集合系统 $(V,\ S,\ G)$ 用于命题 4.3 对于这个构造，令 $d=3$和 $\mathfrak{q}>C\frac{1}{\varepsilon}\log^{10}N\log\log N$为某个绝对常数 $C>0$的素数幂[45]。我们对这些参数取 $\mathfrak{x}$ $\mathfrak{q}$，d和 $\kappa\to\infty$。命题 4.3 中的集合系统为:

- **顶点集** V。

$$V=\mathfrak{x}[1].$$

[45]出于实现方面的考虑，我们取 $\mathfrak{q}=2^z$，其中 z是一个足够大的整数。

- **集合** S。
$$S = \{\mathfrak{X}_s[1] : s \in \mathfrak{X}[2] \cup \mathfrak{X}[3]\}.$$
换言之，我们将 $\mathfrak{X}[1]$ 中每个顶点 $s \in \mathfrak{X}[2] \cup \mathfrak{X}[3]$的邻域作为集合。为简洁起见，写作 $v \in s$ 时，实际指的是 $v \in \mathfrak{X}_s[1]$。

- **解码图** G。 这是 $\mathfrak{X}[2]$ 与 $\mathfrak{X}[3]$之间的图。

我们也来具体说明命题 4.3 中需要分析的一些图：

- **图** (V, S)。 这是 $\mathfrak{X}[1]$ 与 $\mathfrak{X}[2] \cup \mathfrak{X}[3]$之间的图。

- **图** G_s。 对于 $s \in \mathfrak{X}[2]$ （分别对于 $s \in \mathfrak{X}[3]$），交图是 s的链接，也就是 $\mathfrak{X}_s[3]$ （分别对于 $\mathfrak{X}_s[2]$）与 $\mathfrak{X}_s[1]$之间的图。这直接源于这样一个事实：若 $s \sim s'$ 属于 G，那么它们的交集就是 $\mathfrak{X}_s[1] \cap \mathfrak{X}_{s'}[1]$。根据定义可验证，这等于 $\mathfrak{X}_{s,\ s'}[1]$ 即 s' 在 s的链接中的邻域。

命题 5.4 的集合系统 设 $\eta, \beta > 0$ 如命题 5.4 所述。令 $d = \frac{14}{\eta}$，不失一般性，假设它能被 7整除。令 $\mathfrak{q} > 4\eta^2\beta^2$ 为一个素数幂。设 $\mathfrak{X}$ 为参数为 $d, \mathfrak{q}$和 $\kappa \to \infty$的 KMS 复形。[46] 令 $C = \{7, 14, \ldots, d\}$，对于 $i \in \{1, 2, \ldots, 6\}$，我们用$C_{+i} = \{i, 7+i, \ldots, d-7+i\}$表示。

1. **顶点集** V。
$$V = \mathfrak{X}[C],$$
2. **集合** S。 对于顶点 $s \in \bigcup_{i \in C} \mathfrak{X}[i]$，定义 $N_{\mathfrak{X}}(s) = \{v \in \mathfrak{X}[C] : s \in v\}$。集合
$$S = \left\{ N_{\mathfrak{X}}(s) : s \in \bigcup_{i \in C} \mathfrak{X}[i] \right\}.$$
这可能有些令人困惑，因此我们明确地说明一下。对于每个顶点 s 属于 $\mathfrak{X}$ 且颜色属于 C的顶点，我们定义集合 $N(s)$，它由所有包含顶点 s 的面组成（这些面是 V中的元素）。这与典型编码的做法相反；典型编码会将消息比特放在 $\mathfrak{X}$的顶点上，并对面取直积。我们使用反向图，是因为每个顶点（在 V 中）恰好包含于 $d = O$ 个（ S 中的）集合中。这意味着直积码的码率为 $\frac{1}{d} = \Omega(\eta)$。另一方面，我们的复形不是有界度的（这是为了确保足够强的采样），因此常规编码不会是常数码率。此后，当我们写 $s \in S$ 时，实际指的是 $N_{\mathfrak{X}}(s) \in S$。
3. **集合** U。 这些是链接 $\mathfrak{X}_u[C]$ 对应的 $u \in \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$。也就是说，
$$U = \{\mathfrak{X}_u[C] : u \in \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]\}.$$
与之前一样，为简便起见，我们用 $v \in u$ 代替 $v \in \mathfrak{X}_u[C]$ 。集合系统 $In_u = (V_u, S_u)$ 由 $V_u = X_u[C] = u$组成。这些集合由顶点 $s \in \bigcup_{u \in C} \mathfrak{X}_u[i]$索引。集合 s 实际上是
$$N_{\mathfrak{X}_u}(s) \coloneqq \{v \in \mathfrak{X}_u[C] : v \in s\}.$$
换言之，它们是包含 s 、位于 u的链接中的所有颜色为 C 的面。细心的读者会注意到，定义这些集合的另一种方式是直接将集合 s 与 u求交，即
$$N_{\mathfrak{X}}(s) \cap \mathfrak{X}_u[C] = N_{\mathfrak{X}_u}(s).$$
4. **解码图** G。 这是 $\mathfrak{X}[C_{+2}]$ 与 $\mathfrak{X}[C_{+5}$之间的交换游走图].

再次，让我们明确写出命题 5.4 中的图。

[46]同样，出于电路方面的原因，最简单的做法是假设 $\mathfrak{q}$ 是 2 的幂。

• **图** (V, S)。

$$V = \mathfrak{X}[C]$$
$$S = \mathfrak{X}^C(0) = \bigcup_{i \in C} \mathfrak{X}[i]$$

且当且仅当 $v \sim s$ 如果对应于 v 的面包含与 s 对应的 $\mathfrak{X}^C(0)$ 中的顶点（即包含关系反转）。

• **图** (V, U)。

$$U = \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$$
$$V = \mathfrak{X}[C]$$

且当且仅当 $\{v, u\}$ 是交换游走图中的一条边（位于 C 和 C_{+i} 之间，后者是 u 的颜色）。

• **图** (S, U)。

$$U = \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$$
$$S = \mathfrak{X}^C(0)$$

且当且仅当 $\{s, u\}$ 是交换游走图中的一条边（位于 u 的颜色和 s 的颜色之间）。

我们还有局部图。

• **图** G_u。令 $u \in \mathfrak{X}[C_{+i}]$ 表示 $i \in \{2, 5\}$，并令 $j \in \{2, 5\}$，$j \neq i$ 表示另一种颜色。与命题 4.3 中的图类似，此图是如下二部图 (L, R, E)。

$$L = u = \mathfrak{X}_u^C(0)$$
$$R = \mathfrak{X}_u[C_{+j}]$$

且当且仅当 $u' \sim v$ 属于交换游走图时，$\{u', v\} \in E$ 成立。解释与命题 4.3 中的相同。

• **图** (V_s, U_s)。此图的一侧包含 $N_{\mathfrak{X}}(s)$，即包含 s 的面 $\mathfrak{X}[C]$。另一侧则包含 $\mathfrak{X}_s[C_{+2}] \cup \mathfrak{X}_s[C_{+5}]$。当且仅当 $\{v, u\}$ 是一条交换游走边时，$u \sim v$ 之间有一条边。注意，我们可以将 $N_{\mathfrak{X}}(s) \cong \mathfrak{X}_s[C \setminus \{i\}]$ 识别为 $s \in \mathfrak{X}[i]$（且此识别为 $v \in N_{\mathfrak{X}}(s) \mapsto v \setminus \{s\}$）。因此，根据定义，当且仅当 $v \setminus \{s\} \sim u$ 属于 $\mathfrak{X}_s$ 的链接中的交换游走时，$v \sim u$ 成立。因此，此图同构于图 (L, R, E)，其中

$$L = \mathfrak{X}_s[C_{+2}] \cup \mathfrak{X}_s[C_{+5}]$$
$$R = \mathfrak{X}_s[C \setminus \{i\}]$$

并且 $u \sim v$ 是 $\mathfrak{X}_s$中的交换游走边。

• **图** (V_u, S_u)。这是链接中的（反向）包含图，即

$$V_u = \mathfrak{X}_u[C]$$
$$S_u = \mathfrak{X}_u^C(0)$$

并且 $v \sim s$ 当且仅当 $\mathfrak{X}_u[C]$ 中与 v 对应的面包含 $\mathfrak{X}_u^C(0)$中与 s对应的顶点时成立。

• 图 $(G_{In_u})_s$。固定 $u \in U \cong \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$ 和 $s \in S_u \cong \mathfrak{X}^C(0)$。$(G_{In_u})_s$ 的局部图是一个图 (L, R, E)，其中

$$L = s = \{v \in \mathfrak{X}_u[C] : v \text{ contain } s \text{ as a face}\}$$
$$R = \left\{s' \in \mathfrak{X}^C(0) : s \cap s' \neq \emptyset\right\}$$

且边由包含关系确定。

6.6 命题 4.3 与命题 5.4 的证明

定理 （命题 4.3 的重述）。 对于足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个正则超图系统 （V、 S、 G），具有以下性质：

1. $|V| \in [N, N\mathrm{poly}(\log N, \varepsilon^{-1})]$*2.* (V, S) 是双正则的，左度和右度至多为 $\mathrm{poly}(\log N, \varepsilon^{-1})$*3.* G 具有一个长度为 $t \leq O(\log^3(N))$ 的单位拥塞路由方案 *O4.* 对于每个 $s \in S$， G_s 是一个 $\frac{\varepsilon}{15^3 \cdot 10^{11} t^4 \log t}$-采样器 *5.* （反向）包含图 (S, V) 是一个 $\frac{10^{-5}}{4t\varepsilon^2}$-采样器。

命题 *4.3* 的证明。 让我们逐项进行。

1. 回想一下， $\mathfrak{q} = \mathrm{poly}\log(N, \frac{1}{\varepsilon})$。 $V = \mathfrak{X}[1]$ 中的顶点数量是陪集的数量，即 $|SL_d(\mathbb{F}_{\mathfrak{q}^\kappa})|$。 $|K_1| = \mathfrak{q}^3$ 的规模以及 $|SL_3(\mathbb{F}_{\mathfrak{q}^\kappa})| = c\mathfrak{q}^8$ 的规模对于某个 $c \in (0.99, 1)$。这等于某个 $c \in (0.99^{|K_1|}, 1)$ 对应的 $c\mathfrak{q}^{8\kappa-3}$。将 κ 设为 $\frac{3}{8} + \log_{\mathfrak{q}} 8$ 向上取整后的整数值。那么显然 $|V| \in [N, N\mathfrak{q}^8]$，或换言之 $[N, N\mathrm{poly}(\log N, \frac{1}{\varepsilon})]$。

2. 回想一下，(V, S)之间的图就是 $\mathfrak{X}[1]$ 与 $\mathfrak{X}[2] \cup \mathfrak{X}[3]$之间的图。该图确实是双正则的。为此，只需注意到复形在顶层面上有传递作用，因此所有同色顶点的邻域规模都相同，这一点显而易见。特别地，每个 $v \in V = \mathfrak{X}[1]$都出现在 $\mathfrak{X}[2] \cup \mathfrak{X}[3]$中相同数量的集合邻域里。反过来，回想一下，根据第 6.3 节，我们可以将颜色 2 顶点的链接中的颜色 1 顶点与对偶复形中的直线对应起来；根据对偶性，颜色 3 的情形也一样。因此，任意 $s \in \mathfrak{X}[2] \cup \mathfrak{X}[3]$的度都相同。

3. 推论 6.17 中 $\mathfrak{X}[2]$ 与 $\mathfrak{X}[3]$ 之间的路由方案，是一个长度为 $t = O(\kappa^{\log_2 5}) \leq O(\log^3 N)$、具有单位拥塞的路由方案。

4. 回顾一下，对于每个 $s \in S$，图 G_s 是 $\mathfrak{X}_s$的链接。根据引理 6.6，这是一个 $\frac{10}{\sqrt{\mathfrak{q}}}$-扩展图。根据断言 2.9，这些也都是 $\frac{100}{\mathfrak{q}}$-采样器。因此，根据 $\mathfrak{q}$的选择，该链接是一个足够好的采样器。

5. 以下是关于扩展图的一个标准断言，我们也将在命题 5.4 的证明中多次用到。

断言 6.21. 设 $G = (L, R, E)$ 是一个双正则图。假设存在一个划分 $L = L_1 \cup L_2 \cup \cdots \cup L_m$和 $R = R_1 \cup R_2 \cup \cdots \cup R_{m'}$，使得每个 L_i 与 R_j 之间的导出二分图都是双正则的，并且彼此同构。那么，如果 L_1 与 R_1 之间的图是一个 β'-扩展图，则可推出G 是一个 β'-扩展图。

我们可以利用这一断言，将 $L = L_1 = \mathfrak{X}[1]$ 与 $R = R_2 \cup R_3$ 等同起来，其中 $R_2 = \mathfrak{X}[2]$ 与 $R_3 = \mathfrak{X}[3]$。$\mathfrak{X}[1]$ 与 $\mathfrak{X}[i]$ 之间的图分别是 $i \in \{2, 3\}$的 $\frac{10}{\sqrt{\mathfrak{q}}}$-扩展图。根据定义，还可以观察到它们同样是具有相同正则性的双正则图。因此，图 (V, S) 是一个$\frac{10}{\sqrt{\mathfrak{q}}}$-扩展图。采样由对 $\mathfrak{q}$ 的选取以及断言 2.9 得出。

□

命题 （命题 5.4 的重述）**.** 设 $N > 0$ 为一个足够大的整数，并设 $\varepsilon > \frac{1}{\log\log N}$。[47]令 $\delta_{in} = \exp(-3 \cdot 10^3 \frac{1}{\varepsilon})$， $\eta = (\frac{\varepsilon\delta_{in}}{8})^7$ 与 $\beta = \frac{\varepsilon\delta_{in}^2}{\log N})^{28}$。存在一个集合系统 (V, S, U, G)，具有以下性质。

1. $|V| \in [N, N\log(N)^{\exp(O(\frac{1}{\varepsilon}))}]$.

[47]事实上，对 ε 没有任何约束是必要的，但在这组参数范围之外，集合系统会比 N大得多。

2. 每个 $s \in S$ 与 $u \in U$ 的规模为 $\leq \log(N)^{\exp(O(\frac{1}{\varepsilon}))}$。

3. 每个顶点都出现在 $\exp(O(\frac{1}{\varepsilon}))$ 个集合中。

4. 包含图 (V, S)、(V, U)，集合层图 (S, U) 以及局部图 (V_s, U_s)、(V_u, S_u) 和交图 G_u 对每个 $s \in S$ 和 $u \in U$ 而言，都是双正则图。

5. 交图 G_u 是一个 β-采样器。

6. 图 (S, U) 是一个 β-采样器。

7. 对每个 $u \in U$，(V_u, S_u) 都是一个 η-采样器。

8. 对每个 $u \in U$，局部解码图图 G_{In_u} 是 $(1 - O(\eta))|S_u|$ 正则的。

9. 对于每个 $u \in U$ 和 $s' \in S_u$，局部交图 $(G_{In_u})_{s'}$ 都是一个 η-采样器。

10. 对于每个 $s \in S$，图 (V_s, U_s) 都是一个 β-采样器。

11. 在 G 的解码图上存在一个长度为 $O(\log^3 N \operatorname{poly}(\eta^{-1}))$ 的低拥塞子集内部路由方案。

命题 *5.4* 的证明。我们再逐项检查这份待办清单。

1. 顶点数为 $|\mathfrak{X}[C]|$，其中 $C = \{7, 14, 21, \ldots, d\}$。这就是陪集的数量，即 $\frac{|SL_d(\mathbb{F}_{\mathfrak{q}^\kappa})|}{|\mathfrak{X}[C]|}$。换句话说，$\mathfrak{q}$ 的 $\kappa(d^2-1)$ 次方为 $-3d$。将 $\kappa = \log_N \log \mathfrak{q} + 3d$ 向上取整到最近的整数后，我们得到 $|V| \in [N, N\mathfrak{q}^{O(d^2)}]$。由于 $\mathfrak{q}^{O(d^2)} = \log(N)^{\exp(O(\frac{1}{\varepsilon}))}$，第一项显然成立。

2. 让我们对 $s \in S$（类似地，对 $u \in U$）的规模进行界定。回顾一下，集合 s 由顶点 $s \in \mathfrak{X}(0)$ 索引，其颜色属于 C。该集合实际上是 $N_{\mathfrak{X}}(s)$，即包含 s、颜色为 C 的面。注意，对于 $s \in \mathfrak{X}[i]$，集合 $N_{\mathfrak{X}}(s) = \{v \cup \{s\} : v \in \mathfrak{X}_s[C \setminus \{i\}]\}$。因此，要界定集合 s 的规模，只需对任意颜色 C' 下的 $\mathfrak{X}_s[C']$ 的规模进行界定。回顾一下，这就是子群 K_i 中陪集 $K_{C'}$ 的数量。显然，该数量不超过 $|K_i|$ 本身，也就是 $\mathfrak{q}^{O(d^2)}$。因此，根据我们对 $d = \exp(O(\frac{1}{\varepsilon}))$ 和 $\mathfrak{q} = \operatorname{poly}(\log(N), \exp(O(\frac{1}{\varepsilon})))$ 的选择，即可得到正确的界。

   对于 $|\mathfrak{X}_u[C]| \leq |K_{C_{+i}}|$（针对 $i \in \{1, 6\}$），也有类似的界。它至多为 K_i 的规模，因为向集合中添加顶点只会使链接变小，因此上面的界仍然成立。

3. 每个顶点实际上都是一个面 $v \in \mathfrak{X}[C]$。它包含在 $\frac{d}{7} = \exp(O(\frac{1}{\varepsilon}))$ 个集合中，因为它恰好包含 $\frac{d}{7}$ 个来自 $\mathfrak{X}$ 的顶点。

4. 让我们分别考察这五个图，并多次使用断言 6.10：

   (a) 图 (V, S) 是 $\mathfrak{X}[C]$ 与 $\mathfrak{X}^C(0) = \bigcup_{i \in C} \mathfrak{X}[i]$ 之间的包含图。我们已经注意到，$\mathfrak{X}[C]$ 中的任意面都包含相同数量的顶点。反过来，我们再次利用以下事实：对于 $s \in \mathfrak{X}[i]$，$|N_{\mathfrak{X}}(s)| = |\mathfrak{X}_s[C \setminus \{i\}]|$。因此，我们证明 $|\mathfrak{X}_s[C \setminus \{i\}]|$ 与 i 无关。为此，我们使用断言 6.10。对于任意 $i_1, i_2 \in C$，同构 $j \overset{\psi}{\mapsto} j + (i_2 - i_1)$（模 d）是循环的同构。因此，根据断言 6.10，对于任意 $s \in \mathfrak{X}[i_1], s' \in \mathfrak{X}[i_2]$，链接 $\mathfrak{X}_s \cong \mathfrak{X}_{s'}$ 由一个将 $\mathfrak{X}_s[C] \mapsto \mathfrak{X}_{s'}[\psi(C)]$ 映射到的同构给出。由于 $C = \psi(C)$，正则性得证。

   (b) 根据类似的考虑，图 (V, U) 是双正则的。所有顶点的颜色都相同，且 $v \in u$ 当且仅当 $u \in \mathfrak{X}_v[C_{+i}]$。由于所有 $v \in V$ 的颜色集合都相同，它们的链接彼此同构（回想一下，$\mathfrak{X}$ 上存在一个群作用，它在每个 $\mathfrak{X}[C']$ 上都是传递的）。至于 $u \in U$，回想一下，它们由 $\mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$ 索引，而 u 的顶点集实际上是 $\mathfrak{X}_u[C]$。因此，我们需要证明 $|\mathfrak{X}_u[C]|$ 的规模不取决于 $u \in \mathfrak{X}[C_{+2}]$ 还是 $u \in \mathfrak{X}[C_{+5}]$。与上面的条目类似，这可由循环的同构 $j \overset{\psi}{\mapsto} -j$（模 d）得出，该同构将 $\psi(C_{+2})$ 映射到 $= C_{+5}$，并将 $\psi(C)$ 映射到 $= C$。因此，根据断言 6.10，任意 $u \in \mathfrak{X}[C_{+2}]$ 的链接都与任意 $u' \in \mathfrak{X}[C_{+5}]$ 的链接同构，且该同构将 $\mathfrak{X}_u[C] \mapsto \mathfrak{X}_{u'}[C]$ 映射到相应对象，由此可得正则性。

(c) 集合层图实际上是 $U \cong \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$ 与 $S \cong \mathfrak{X}^C(0)$ 之间的图，其中存在边 $\{u, s\}$ 当且仅当 $u \cup s \in \mathfrak{X}$。(*b*) 中的同构还将表明，任意两个 u，$u' \in U$ 的度都相同（无论它们的颜色如何）。(*a*) 中的同构证明了 s，$s' \in S$也具有相同的性质。

(d) 固定 $s \in S$。识别 $s \in \mathfrak{X}[i]$后，局部图（V_s，U_s）在 $\mathfrak{X}[C]$ 中包含 s的集合，以及通过交换游走与 s 连通的$u \in \mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$。当且仅当 $v \in u$ 当且仅当交换游走中的 $u \sim v$ 时，边为 $u \sim v$。这与带有交换游走边的 $\mathfrak{X}_s[C \setminus \{i\}]$和 $\mathfrak{X}_s[C_{+2}] \cup \mathfrak{X}_s[C_{+5}]$ 之间的图同构。V_s 侧的正则性源于这里只有一种颜色。为证明 U_s 的正则性，注意到对于每个 $i \in C$，都存在一个循环的同构映射，将 C_{+2} 映射到 C_{+5}，并将 C 和 i 映射到其自身；这就是 $j \mapsto -j + 2i$（模 d）。这证明了存在一个将 $\mathfrak{X}_s[C_{+2}]$ 映射到 $\mathfrak{X}_s[C_{+5}]$的该图同构。

(e) 对于 (V_u，S_u)，我们注意到，这是集合 $V_u \cong \mathfrak{X}_u[C]$和 $S_u \cong \mathfrak{X}_u^C(0)$之间的包含图。$V_u$ 侧的正则性是因为 $\mathfrak{X}_u[C]$ 中的每个面都包含相同数量的顶点。S_u 侧的正则性则源于项 (*a*) 中循环的同一同构。细节从略。

(f) 我们来描述 G_u 关于 $u \in \mathfrak{X}[C_{+2}]$ 的情形（另一种情形相同，只需交换$2 \leftrightarrow 5$）。该图的集合以 $u' \in \mathfrak{X}_u[C_{+5}]$ 为索引，顶点则是 $\mathfrak{X}_u^C(0)$。每个 u'都与 u 在顶点 $\mathfrak{X}_{u \cup u'}^C(0)$ 处相交。因此，这就是两侧之间由交换游走边连接而成的图。通过与上文类似的论证可知，该图是双正则的，证明从略。

5. 我们在上面对集合系统的描述中看到，图 G_u 的一侧是 $\mathfrak{X}_u[C_{+j}]$（对于$j \in \{2, 5\}$），另一侧是 $\mathfrak{X}_u^C(0)$，边则是 u的链接中的交换游走图边。对于每个 $i \in C$，$\mathfrak{X}_u[C_{+j}]$ 与 $\mathfrak{X}_u[i]$ 之间的图是一个 $\frac{10d}{\sqrt{\mathfrak{q}}}$-扩展图，这是由定理 6.2 得出的。通过再次应用断言 6.10，还可知不同部分具有相同的正则性。因此，根据断言 6.21，该图是一个$10d$

$\sqrt{\mathfrak{q}}$-扩展图。根据断言 2.9 以及 $\mathfrak{q}$ 的选择，它是一个 β-采样器。

6. 回顾一下，图 (S，U) 是交换游走图，其中一侧是 $\mathfrak{X}^C(0)$，另一侧是$\mathfrak{X}[C_{+2}] \cup \mathfrak{X}[C_{+5}]$。这一项与上面类似。注意，对于任意选择的部分 $i \in C$ 和$C' \in \{C_{+2}, C_{+5}\}$，$\mathfrak{X}[i]$ 和 $\mathfrak{X}[C']$ 之间的交换游走图由断言 6.10 可知是同构的。它们也是 $\beta' = {}^{O}{}_{(d)\sqrt{\mathfrak{q}}}$-扩展图。断言 6.21 与断言 2.9 一起推出，图 (S，U) 是一个 $\underline{{}_{O(d^2)}{}_{\mathfrak{q}}} = \beta$-采样器。

7. 如上所见，图 (V_u，S_u) 是 $u \in U \cong \mathfrak{X}[C_{+1}] \cup \mathfrak{X}[C_{+6}]$ 的链接中 $\mathfrak{X}_u[C]$ 与 $\mathfrak{X}_u^C(0)$ 之间的包含图。根据定理 6.1，这是一个 $O(\frac{1}{|C|})$-扩展图。由于 $\frac{1}{|C|} = \Omega(\eta)$，该项即得。

8. 回忆一下，局部解码图 G_{In_u} 是这样一张图：当且仅当 $s \cap s' \neq \emptyset$时，我们将 $s \sim s'$ 相连。回忆 $S_u \cong \mathfrak{X}_u^C(0)$。注意，集合 C_{+i} 分离 C。因此，由断言 6.18 可知，链接 X_u^C是一个完全 $|C|$-部图。特别地，$s \cap s' \neq \{0\}$（即存在一个包含 $\{s, s'\}$的 C-面）当且仅当 s 和 s' 的颜色不同。因此，s 与除去占比为 $\frac{1}{|C|}$的顶点外的所有顶点相连，故是 $(1 - O(\eta))|S_u|$-正则的。

9. 设 i 是 s的颜色。与上文类似，根据这张图的描述，它正是 $\mathfrak{X}_{u \cup \{s\}}[C \setminus \{i\}]$ 与 $\mathfrak{X}^{C \setminus \{i\}}(0)$之间的包含图。根据定理 6.1 和断言 2.9，这是一个 $O(\frac{1}{|C|-1}) \leq \eta$-采样器。

10. 如上所见，图 (V_s，U_s) 与 $\mathfrak{X}_s[C]$ 和 $\mathfrak{X}_s[C_{+2}] \cup \mathfrak{X}_s[C_{+5}]$之间的图同构，当且仅当 $v \sim u$ 在交换游走中时，两者之间存在一条边。交换游走图是一个 $\frac{10d}{\sqrt{\mathfrak{q}}}$-扩展图，因此根据断言 6.21，它也是一个$\frac{10d}{\sqrt{\mathfrak{q}}}$-扩展图。根据断言 2.9，可得所需的采样。

11. 回忆一下，解码图 G 是 $\mathfrak{X}[C_{+2}]$ 与 $\mathfrak{X}[C_{+5}]$之间的交换游走图。颜色集合 $C_{+2} \cup C_{+5}$ 被集合 $D_1 = C_{+1} \cup C_{+4}$ 和 $D_2 = C_{+3} \cup C_{+6}$ 双重分离。因此，根据命题 6.19，存在一个具备所需性质的子集内部路由方案；注意，其长度上界为 $O\left(\kappa^{\log_2 5} \text{poly}\left(\frac{1}{\eta}\right)\right) \leq O\left(\log^3(N) \exp\left(O\left(\frac{1}{\varepsilon}\right)\right)\right)$。

□

6.7 高维扩展体电路实现

现在，我们介绍若干电路，它们是为基于 HDX 的码构造编码器和解码器所必需的。从宏观上看，我们需要对特殊线性群 $SL_d(R)$ 的某些陪集进行编码。自然的做法是直接使用所有矩阵的集合进行编码，并将给定的输入矩阵映射到其对应的陪集。遗憾的是，与子空间情形（见第 3.3.4 节）一样，这种编码存在过多冗余；我们需要一种对陪集进行编码的方法，使其比特表示的长度确实等于陪集总数的对数（误差趋于零）。下文将看到，基本上可以通过高斯消元的变体来实现这一点（不过记号相当繁琐）。利用陪集结构，我们将为每个陪集定义一种标准矩阵形式：通过该陪集中允许的列操作，依次将所有“自由”系数置零，从而得到精确表示；不可逆矩阵带来的误差趋于零。

对于矩阵 A，我们用 $col(A,\ j)$ 表示它的第 j 列。

6.7.1 关于陪集的另一种观点

我们的码的编码器和解码器电路需要将 $\mathfrak{X}[C]$ 中的一个顶点作为输入，它是某个 K_i 的陪集；或者将一个面作为输入，它是集合 $S \subseteq [d]$ 对应的 K_S 的某个陪集。在内部，我们的电路只会使用一个矩阵和一个索引 i 来表示陪集。不幸的是，和第 3.3.4 节中的子空间情形一样，我们的电路不能以这种方式接收复形的顶点或面作为输入。原因在于，矩阵的表示包含了太多重复项和无效矩阵（重复项是指许多矩阵表示同一个陪集；无效矩阵是指不属于 $SL_d(R)$ 的矩阵）。如果尝试使用朴素表示，这一事实会对实际得到的码率造成不利影响。因此，我们必须构造一种格式，以大约 $\log([SL_d(R) : K_i])$ 比特（分别为 $\log([SL_d(R) : K_S])$ 比特）来表示 K_i 或 K_S 的陪集。

接下来我们要做的就是这个。

子群 K_i 可描述如下。子群 K_d 是由元素属于 $\mathbb{F}_\mathfrak{q}$ 的所有上三角矩阵构成的集合。对于 $i < d$，子群 K_i 是由满足以下条件的矩阵构成的集合：

1. 对角线上的元素均为 1。
2. 在索引 $I_i^u = \{(j,k) : i+1 \le j < k \le d\} \cup \{(j,k) : 1 \le j < k \le i\}$ 处的元素属于 $\mathbb{F}_\mathfrak{q}$。
3. 在索引 $I_i^\ell = \{(j,k) : 1 \le k < i+1 \le j \le d\}$ 处的元素属于 $t \cdot \mathbb{F}_\mathfrak{q}$。
4. 其余元素均为零。

$$K_2 : \begin{pmatrix} 1 & a_4 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ a_5 t & a_6 t & 1 & a_1 & a_2 \\ a_7 t & a_8 t & 0 & 1 & a_3 \\ a_9 t & a_{10} t & 0 & 0 & 1 \end{pmatrix}$$

图 8：K_2 在 $SL_5(R)$ 中。$I_2^u = \{(3,4),(3,5),(4,5)\} \cup \{1,2\}$。$I_2^\ell = \{3,4,5\} \times \{1,2\}$。

示例见图 8。事实上，对于 K_d，我们将 $i = 0$ 代入 I_i^u，I_i^ℓ 的定义中，就能得到相同的描述。观察图 9，我们注意到，很容易找到非空 $J \subseteq [d]$ 的交集 $K_J = \bigcap_{j \in J} K_j$。这正是具有以下性质的子群：

1. 对角线上为 1。2. 索引 $I_J^u = \bigcap_{j \in J} I_j^u$ 中的 $\mathbb{F}_\mathfrak{q}$ 元素。3. 索引 $I_J^\ell = \bigcap_{j \in J} I_j^\ell$ 中的 $t \cdot \mathbb{F}_\mathfrak{q}$ 元素。4. 其余项均为零。

$$K_4:\begin{pmatrix}1&a_1&a_2&a_3&0\\0&1&a_4&a_5&0\\0&0&1&a_6&0\\0&0&0&1&0\\a_7t&a_8t&a_9t&a_{10}t&1\end{pmatrix},\ K_{\{2,4\}}:\begin{pmatrix}1&a_1&0&0&0\\0&1&0&0&0\\0&0&1&a_2&0\\0&0&0&1&0\\a_3t&a_4t&0&0&1\end{pmatrix}$$

图 9： K_4 与 $K_{\{2,3\}}$ 在 $SL_5(R)$ 中。

最后，我们注意到，这些群也由以下元素生成：

$$Gen(J)=\{e_{j,k}(\alpha):(j,k)\in I_J^u,\alpha\in\mathbb{F}_{\mathfrak{q}}\}\cup\left\{e_{j,k}(\alpha t):(j,k)\in I_J^\ell,\alpha\in\mathbb{F}_{\mathfrak{q}}\right\}.$$

有鉴于此，将矩阵 A 与矩阵 $B\in Gen(J)$ 右乘，对应于执行以下某个初等列变换

$$col(A,k)\leftarrow col(A,k)+\alpha\cdot col(A,j) \tag{8}$$

如果 $(j,k)\in I_J^u$，或者

$$col(A,k)\leftarrow col(A,k)+\alpha t\cdot col(A,j) \tag{9}$$

如果 $(j,\ k)\in I_{J^\circ}^\ell$。

因此 $A'\in A$ K_J 当且仅当可以通过上述初等列变换从 A 到达 A' s.

6.7.2 常数码率构造的规范表示

设 (V,S,U) 是命题 5.4 中所构造族中的一个集合系统。我们现在为 $v\in V$ 构造典范代表元。回想一下，在该构造中，我们取某个 $d=7d'$，以及维数为 d 的 KMS 复形中的 $V=\mathfrak{X}[C]$ （对于 $C=\{7,14,\ldots,d\}$）。顶点 $v\in V$ 是 K_C 的一个陪集，因此我们先利用上述观察来描述 K_C。K_C 中的矩阵是上三角矩阵（即 $I_C^\ell=\emptyset$），并且

$$I_C^u=\cup_{k=0}^{d'-1}\{(i,j):7k+1\le i<j\le 7k+7\}.$$

因此，根据 (8) 中运算所给出的刻画，两个矩阵 $A,A'\in SL_d(R)$ 属于 K_C 的同一个陪集，当且仅当对每一个 $j=7k+r$ （其中 $r\in[6]$），

$$col(A',j)=col(A,j)+\sum_{1\le r'<r}\alpha_{r'}\cdot col(A,7k+r'),$$

其中 $\alpha_{r'}\in\mathbb{F}_{\mathfrak{q}}$。特别地，由 $j\equiv_7 1$ 索引的每一列在 A 与 A' 之间完全相同。由 $j\equiv_7 2$ 索引的每一列满足：差 $col(A,j)-col(A',j)$ 属于 $\mathbb{F}_{\mathfrak{q}}\cdot col(A,j-1)$，依此类推。

有鉴于此，我们将把 AK_C 构造为一种规范表示，即构造一个特定的矩阵 $A'\in AK_C$，并基于上述结构，通过某种高斯消元尽可能多地将系数置零。这样便能消除多余的自由度，使编码的比特长度确实大致为 $\log([SL_d(R):K_C])$。

为此，对于矩阵 A，令 $\ell_j(A)$ 表示 j列中的第一个非零行。令 $z_j(A)$ 表示多项式$A_{\ell_j(A)\ j}$的度。从高层次来看，规范表示是一种如下形式：对于任意 $7k+1\le j<j'<7(k+1)$，通过从 $col(A,j')$ 中减去 $col(A,j)$ 的若干倍，将 $A_{\ell_j(A)\ j'}$ 处的 $t^{z_j(A)}$ 的系数置零。定义如下。

1. 由 $j\equiv_7 1$ 索引的列为 $col(A',j)=col(A,j)$。

2. 对于索引为 $j\equiv_7 2$ 的列，我们要求如下： $col(A',\ j)=col(A,\ j)+\alpha\cdot col(A,\ j-1)$，并且要求多项式 $A'_{\ell_{j-1}(A')\ j}$ 中单项式 $t^{z_{j-1}(A')}$ 的系数为 0。注意，我们总能通过从 A 中加上 $col(A,\ j-1)$ 的适当倍数来得到这样的矩阵。

3. 对于索引为 $j \equiv_7 3$ 的列，我们要求如下： $col(A', j) = col(A, j) + \alpha \cdot col(A, j-2) + \beta \cdot col(A, j-1)$，并且满足以下约束：(1) 多项式 $A'_{\ell_{j-2}(A')\ j}$ 中单项式 $t^{z_{j-2}(A')}$ 的系数为 0；(2) 多项式 $A'_{\ell_{j-1}(A')\ j}$ 中单项式 $t^{z_{j-1}(A')}$ 的系数为 0。注意，给定一个已经满足此列表前两项的矩阵 $A'' \in AK_C$，我们也可以将其变为满足第三项的矩阵。我们只需将 $col(A'', j-2)$ 的一个倍数加上去，使 $A'_{\ell_{j-2}(A')\ j}$ 中 $t^{z_{j-2}(A'')}$ 的系数变为 0；然后再将 $col(A'', j-1)$ 的一个倍数加上去，使 $A'_{\ell_{j-1}(A')\ j}$ 中 $t^{z_{j-1}(A'')}$ 的系数变为 0。注意，由于 A'' 满足此列表的第二项，加上 $col(A'', j-1)$ 的倍数不会影响 $A'_{\ell_{j-2}(A')\ j}$ 中 $t^{z_{j-2}(A'')}$ 的系数。

4. 一般而言，典范代表元的定义如下。

定义 6.22. 我们称 $A' \in AK_C$ 是标准的，如果对于每个 $r \in \{2, \ldots, 7\}$、列 $j \equiv_r 7$ 以及每个 $r' \in [1, r-1]$，$A'_{\ell_{j-r'(A')\ j}}$ 中 $t^{z_{j-r'(A')}}$ 的系数都为0。

我们称一列 在 $A'' \in AK_C$ 中是标准的，如果它满足上述条件（即使其他列不满足）).

命题 6.23. 对于每个陪集 AK_C，都存在唯一的标准 $A' \in AK_C$。

这个断言的证明与高斯消元和简化行阶梯形类似。我们逐列处理，并将前面各列的倍数加到当前列上，以将特定条目化为零。与高斯消元一样，我们按一定顺序进行列操作，确保后续操作不会干扰先前将条目化为零的结果。

命题 *6.23* 的证明。 我们分别证明存在性和唯一性，先从存在性开始。令 $A = A^1$。我们通过构造矩阵序列 A^2、 A^3、.....、 A^7 来构造矩阵 $A' = A^7$，使得矩阵A^r 对每个 $r' \le r$都满足典范性要求。每个 A^r 都将通过对 $j \equiv_r 7$ 的列进行列操作得到（在第 r 步中，其余列保持不变）。

给定 A^{r-1} 和一列 $j = 7k + r$，对于每个 $r' = 1$，2、.....、 $r-1$，我们:

1. 将 b 设为 $A^{r-1}_{\ell_{7k+r'}(A^{r-1})\ 7k+r'}$ 中 $t^{\ell_{7k+r'}(A^{r-1})}$ 的系数。

2. 将 a 设为 $A^{r-1}_{\ell_{7k+r'}(A^{r-1})\ j}$ 中 $t^{\ell_{7k+r'}(A^{r-1})}$ 的系数。

3. 如果 $a \neq 0$，则执行 $col(A^r, j) \leftarrow col(A^{r-1}, j) = \frac{a}{b} col(A^{r-1}, 7k + r')$。

注意，如果 $r' < r''$，那么 $A^{r-1}_{\ell_{7k+r'}(A^{r-1})\ 7k+r''}$ 中 $t^{\ell_{7k+r'}(A^{r-1})}$ 的系数已经设为 0，因此，在考虑第 $7k + r'$ 列时，一旦第 j 列中 $t^{\ell_{7k+r'}(A^{r-1})}$ 的系数被设为零，后续的列操作就不会改变它。因此， A^r 满足必要的规范性要求。

关于唯一性，反设存在两个不同的 $A', A'' \in AK_C$，它们具有标准形。令 $j = 7k + r$ 为它们不同的第一列。根据上述内容，这意味着 $col(A', j) - col(A'', j)$ 属于 $col(A', 7k + r')$ 对于 $r' \in \{1, 2, \ldots, r-1\}$ 的张成空间。令 $7k + r''$ 为 $col(A', j) - col(A'', j)$ 的线性组合中系数非零的最小列（回想一下，这些列线性无关，因此这样的组合确实唯一）。接着注意，$A'_{\ell_{7k+r''}, 7k+r''}$ 中 $t^{\ell_{7k+r''}(A')}$ 的系数非零，但对于所有满足 $r'' < r' < r$ 的 $A'_{\ell_{7k+r''}, 7k+r'}$，该系数均为零。因此， $col(A', j) - col(A'', j)$ 中 $t^{\ell_{7k+r''}(A')}$ 的系数非零，这与 A' 或 A'' 的标准性矛盾。 □

既然我们已经确立了什么是规范表示，就可以构造一个电路：它获取某个陪集的规范表示的短比特描述，并构造该陪集的规范表示。该电路执行以下操作：

1. 每一列 $j \equiv_1 7$ 都直接编码为 $\mathbb{F}_q[t]/\phi(t) \cong \mathbb{F}_q^\kappa$中的一列 d 元素。电路只需读取这些列的完整描述，就能重建它们。2. 形如 $j = 7k + r < d$ 的列使用 $\mathbb{F}_q$中的 $\kappa d - (r-1)$ 元素进行编码，其中有 r 个条目对应于 $t^{\ell_{7k+r'}(A)}$ 的系数（对于 $r' < r$），这些系数被设为零，因此不会被编码。电路逐列重建这些列，并利用其对 $7k + r' < 7k + r$ 中首项系数的了解，确定哪些系数应留空。

3. 最后，末列使用 $\mathbb{F}_{\mathfrak{q}}$中的 $\kappa d-\kappa-7$ 个元素进行编码。末列不提供系数的精确描述，而是编码一个线性方程组的解，该方程组要求前 6 行中的首项系数为零且 整个矩阵的行列式为 1。注意，行列式为 1 的方程确实是关于末列各项的线性方程（例如，可以利用末列以及其余列的子式来计算行列式）。

注意，如果输入电路的前 $d-1$ 列线性无关，那么最后一个电路中的线性方程确实有规模为 $\mathbb{F}_{\mathfrak{q}}^{\kappa d-\kappa-7}$ 的解空间。这是因为我们可以取任意一个与前 $d-1$ 列线性无关的列，构造出行列式为任意值的矩阵，再使用上述算法对其进行“规范化”，而不改变行列式（这说明，关于行列式的约束确实对应 $\mathfrak{q}^{\kappa}$ 个陪集）。

我们来计算表示的长度与陪集数量的对数。规模 $|K_C|=\mathfrak{q}^{21d'}=\mathfrak{q}^{3d}$。 $SL_d(R)$ 的规模为 $=\mathfrak{q}^{\kappa(d^2-1)}(1-O(\frac{1}{\mathfrak{q}^{\kappa}}))$。因此，$\log[SL_d(R):K_C]=(\kappa d^2-3d-\kappa)\log\mathfrak{q}-O(\mathfrak{q}^{-\kappa})$。另一方面，我们的描述长度为 $\frac{d}{7}\sum_{r=0}^{6}(\kappa d-r)-\kappa=\kappa d^2-3d-\kappa$，与之相符，差别仅在于趋于零的 $o(1)$ 项；该项源自我们对一个很小的不可逆矩阵集合的编码（在前面几节的论证中可以安全地忽略它）。

6.7.3 陪集的标准表示 K_i

我们的构造还需要构造顶点 AK_i 的陪集的标准表示。对于 K_d 的陪集，其规范矩阵与上文 K_C 的矩阵相似，只不过我们不将索引划分为长度为 7 的区间，而是将所有列的倍数加到第 j 列上，以消去首项系数 $j'<j$。

然而，对于 $i<d$ 的 K_i 陪集，会出现一个问题。在这些陪集中，不变的是第 $(i+1)$ 列（而不是像 K_d 中那样的第一列）。第 $(i+2)$ 列可以加上 $col(A,\ i+1)$ 的倍数，依此类推。然而，当我们回到第 1 列到第 i 列时，规则就变了。原因是现在我们还要进行如式 (9) 所示的列操作，即将某些列的 t 倍加到其他列上。为了解决这个问题，我们使用不同的基来写出第 1、2、.....、$i-1$ 列中的多项式，使其适合加上 t 的倍数。详情如下。

我们用 $E=(1,t,t^2,\ldots,t^{\kappa-1})$ 表示我们所使用的多项式环的一组基。我们用 $\tilde{E}=(t,t^2,\ldots,t^{\kappa})$ 表示另一组基。我们将 t^{κ} 解释为多项式 $t^{\kappa}-\phi(t)$。[48] 注意，映射 $f(t)\mapsto tf(t)$ 将多项式 $\sum_{j=0}^{\kappa-1}\beta_j t^j$（即 β_j 是基 E 中第 $(j+1)$ 个向量的系数）映射为多项式 $\sum_{j=1}^{\kappa}\beta_j t^j$。也就是说，此时 β_j 是基 $\tilde{E}$ 中第 j 个元素的系数。

要在 AK_C 的标准形与 AK_i 的标准形之间建立类比，关键是将 $i+1\le j\le d$ 的列视为以基 E 表示的多项式，并将 $1\le j\le i$~ 的列视为以基 E 表示的多项式。这样，当我们将满足 $j\le i$ 的列中的系数置零时，就会将相对于 E 的系数~ 置零；这与我们乘以 t 列 $j\ge i+1$ 的做法正好相配。

有鉴于此，考虑如下关于规范表示的定义。对于矩阵 A 与$j\in[d]$，令 $\ell_j(A)$ 表示第 j 列中第一个非零行。令 $z_j(A)$ 表示 $A_{\ell_j,j}$ 中多项式的度，并令 $\tilde{z}_j(A)$ 表示在基 $t,t^2,\ldots,t^{\kappa}$ 下， $\ell_j(A)$ 处多项式的度。遗憾的是，尽管这里的想法很简单，记号却相当繁琐；为清楚起见，我们需要分别给出 $1\le i\le d-2$ 以及“边界情况” $i=d-1$ 和 $i=d$ 的定义。我们先从前者开始:

定义 6.24 (标准形)。 设 $i\le d-2$，且 AK_i 是一个陪集。若矩阵 $A'\in AK_i$ 满足以下条件，我们称其为标准的

1. 对每个 $j\in[i+2,d]$和 $i+1\le j'<j$， $A'_{\ell_{j'}(A'),j}$ 中的多项式在 $t^{z_{j'}(A')}$ 的系数为零。

2. 对于每个 $j\in[1,i]$和 $1\le j'<j$， $A'_{\ell_{j'}(A'),j}$ 中的多项式在基 E 下展开时， $t^{\tilde{z}_{j'}(A')}$ 的系数为零。

[48]假设我们用来取商的多项式 $\phi(t)$ 是首一的，那么这确实是一个度为 $\le\kappa-1$ 的多项式。此外，由于 $\phi(t)$ 不可约，这确实构成一个基: $\phi(t)$ 在 0 处没有根，因此 $t^{\kappa}-\phi(t)$ 是唯一一个带有常数项的多项式。

3. 对于每个 $j \in [1,i]$ 和 $i+1 \le j' \le d$，$A'_{\ell_{j'}(A')\,j}$ 中的多项式在基 E 下展开时，$t^{z_{j'}(A')+1}$ 的系数为零。

遗憾的是，当 $i \ge d-1$ 时，该定义可以有不同的解释，因为定义中出现的 $i+2$ 大于 d。这两种情形下的正确定义为：

定义 6.25（标准形 - 边情形模 d）。对于 $i = d-1$，只应出现第 2 项和第 3 项情形，也就是说，我们仅对列 $j \in [1, d-1]$ 有约束。因此应读作：

1. 无第 1 项。*2.* 对每个 $j \in [1, d-1]$ 和 $1 \le j' < j$，$A'_{\ell_{j'}(A')\,j}$ 中的多项式在基 E 下当~写出时，对 $t^{\tilde{z}_{j'}(A')}$ 没有系数。

3. 对每个 $j \in [1, d-1]$ 和 $i+1 \le j' \le d$，$A'_{\ell_{j'}(A')\,j}$ 中的多项式在基 E 下写出时，对 $t^{z_{j'}(A')+1\tilde{}}$ 没有系数。

仅显示第 $i = d$ 项，而不是 $j \in [i+2, d]$，应为 $j \in [2,d]$。应读作：

1. 对于每个 $j \in [2,d]$ 和 $1 \le j' < j$，A' 中的多项式 $\ell_{j'(A')\,j}$ 没有 $t^{z_{j'}(A')}$ 的系数。

2. 没有第 2 项。

3. 无第 *3* 项。

断言 6.26. 存在唯一的标准矩阵 $A' \in AK_i$。

断言 *6.26* 的证明. 我们证明其存在性。唯一性的证明与命题 6.23 中的相同，因此留给读者。存在性的证明只需应用一种类似于高斯消元的算法。我们注意到，对列 $i+2 \le j \le d$ 所能进行的操作是将列 $i+1 \le j' < j$的 $\mathbb{F}_{\mathfrak{q}}$ 倍加上去。对于列 $1 \le j < i$，我们可以将列 $1 \le j' < j$ 的倍数加上去，或者将列 $i \le j' \le d$的 $t\mathbb{F}_{\mathfrak{q}}$ 倍加上去。

因此，我们的算法将从一列依次处理到下一列，从 $j = i+1$ 开始，直到 $j = d$，然后回到 $j = 1$ 并继续到 $j = i$。为方便起见，我们将此顺序记为 $i+1 \prec i+2 \prec \cdots \prec d \prec 1 \prec 2 \prec \cdots \prec i$，并在下文中使用。

在每一步中，算法都有矩阵 A^j，并且只使用索引为 $j' \prec j$ 的列，对 $col(A^j,\ j)$ 执行列操作，也就是说，这些列已在之前的轮次中修改过（当然，算法中不会修改的第 i 列除外）。也就是说，除了第 j 列之外，A^{j+1} 在所有列中都等于 A^j，并且

$$col(A^{j+1}, j) = col(A^j, j) + \sum_{j' \prec j} \alpha_{j'} \cdot col(A^j, j')$$

wh其中，当 $j \ge i+2$ 时为 $\alpha_{j'} \in \mathbb{F}_{\mathfrak{q}}$，或者当 $j,\ j' \le i$ 时为 $\alpha_{j'} \in \mathbb{F}_{\mathfrak{q}}$，其他情况下则为某个 $\alpha_{j'} \in \mathbb{F}_{\mathfrak{q}}$ 对应的 $\alpha_{j'} = \alpha t$.

我们按照 $\prec$ 的顺序进行归纳法证明：可以找到一个线性组合，使第 j 列中断言所指定的系数归零。先来看基础情形：

1. 如果 $i < d-1$：在索引 $A^{i+2}_{i+1,\ \ell_{i+1}(A^{i+2})}$ 的第一轮中，有一个度为 $z_{i+1}(A^{i+2})$ 的多项式，因此，如果 $col(A^{i+2},\ i+2)$ 有某个非零系数 $t^{z_{i+1}(A^{i+2})}$，我们可以加上 $col(A^{i+2},\ i+1)$ 的一个域倍数，使该系数归零。

2. 若 $i = d$：则此处与前一种情况的论证相同，只需分别将 $i+1, i+2$ 替换为 1,2。

3. 若 $i = d-1$：则我们进行类似的操作。在索引 $A^1_{d,\ \ell_d(A^1)}$ 处，有一个度为 $z_d(A^1)$ 的多项式。因此，若将 $col(A^1,\ d)$ 乘以 t，那么将其写成基 $\tilde{E}$ 时，所得多项式的度为 $z_d(A^1)+1$。因此，若 $col(A^1,\ 1)$ 有某个非零系数 $t^{z_d(A^d+1)}$（根据基 $\tilde{E}$），我们可以将 $col(A^1,\ d)$ 的一个域倍数加上去，从而将 $col(A^{d-1},\ 1)$ 中 $t^{z_d(A^d)}$ 的系数消为零。

现在讨论归纳步骤，我们也分情况讨论 $i+2 \le j \le d$ 和 $1 \le j \le i$。先从前一种情况开始。根据归纳假设，在 A^j 中，对于每个 $k \prec k' \prec j$， $col(A^j, k')$ 中的第 $\ell_k(A^j)$ 行都是一个多项式，其中 $t^{z_k}(A^j)$ 的系数为零。因此，我们可以遍历所有 $k = i, i+1, \ldots, j-1$，并将 $col(A^j, k)$ 的倍数加到 $col(A^j, j)$ 上，以便将 $A^j_{\ell_k}(A^j)_j$ 中的 $t^{z_k}(A^j)$ 消为零。一旦我们将 $A^j_{\ell_k}(A^j)_j$ 中的 $t^{z_k}(A^j)$ 消为零，再为 $k' \succ k$ 添加任意 $col(A^j, k')$，都能保持 $t^{z_k}(A^j)$ 的系数为零。我们还注意到，由于这些列线性无关，经过这些操作后， $col(A^{j+1}, j) \neq 0$，因此 $\ell_j(A^{j+1})$ $z_j(A^{j+1})$ 仍然有定义。

现在我们也来讨论 $1 \le j \le i$的情况。与上面类似，我们断言，如果考虑向量 $t \cdot col(A^j,\ i)$ $t \cdot col(A^j,\ i+1) \ldots,$ $t \cdot col(A^j,\ d)$ 和 $col(A^j,\ 1)$ $col(A^j,\ 2) \ldots, col(A^j,\ j-1)$，那么对于任意$k \prec k' \prec j$，若将 $col(A^j,\ k')$ 按照 $\tilde{E}$ 展开，则：

1. 若 $i \ge k,\ k'$，那么 $t \cdot col(A^j,\ k')$ 中的 $\ell_k(A^j)$ 行是一个多项式，其中 $t^{z_k(A^j)+1}$ 的系数为零。2. 若 $i \ge k$ 且 $k' < i$，那么 $col(A^j,\ k')$ 中的 $\ell_k(A^j)$ 行是一个多项式，其中 $t^{z_k(A^j)+1}$的系数为零。3. 若 $k,$ $k' < i$，那么 $col(A^j,\ k')$ 中的 $\ell_k(A^j)$ 行是一个多项式，其中 $t^{\tilde{z}_k(A^j)}$的系数为零。

因此，我们可以遍历所有 $k = i, i+1, \ldots, j-1$，并将 $t \cdot col(A^j, k)$ 的倍数（若 $k \ge i$）以及 $col(A^j, k)$ 的倍数（若 $k < i$）加到 $col(A^j, j)$ 上，以便将 $A^j_{\ell_k}(A^j)_j$ 中的 $t^{z_k}(A^j)+1$ （若 $k \ge i$）或 $t^{\tilde{z}_k}(A^j)$（若 $k < i$）归零。一旦我们将 $A^j_{\ell_k}(A^j)_j$ 中 k 的相关系数归零，再次添加任意 $col(A^j, k')$（或 $t \cdot col(A^j, k')$）仍会使 k 的系数保持为零。

□

现在，我们可以用编码 AK_C 的陪集时相同的方法对陪集 AK_i 进行编码。

1. 显式编码第 $(i+1)$ 列。2. 对于 $j = i+2, i+3, \ldots, d$，我们使用 $\mathbb{F}_\mathfrak{q}$ 中的 $\kappa d - (j-i-1)$ 个元素编码第 j 列。给定列 $i+1, i+2, \ldots, j-1$，我们解码第 j 列；其中，对于所有 $i+1 \le k < j$，我们将 $A_{\ell_k}(A)_j$ 中 $t^{z_k}(A)$ 的系数归零。我们注意到，只有在构造出前面的列之后，才能确定这些系数。3. 对于 $j = 1$、2、$\ldots$、$i-1$，我们使用 $\mathbb{F}_\mathfrak{q}$ 中的 $\kappa d - (d-i+j-1)$ 个元素进行编码。现在，我们的元素表示相对于基 $\tilde{E}$ 的系数。对于所有 $i+1 \le k \le d$，我们将 $A_{\ell_k}(A)_j$ 中 $t^{z_k}(A)+1$ 的系数归零，并将所有 $1 \le k < j$ 对应的 $t^{\tilde{z}_k}(A)$ 的系数归零。4. 对于最后一列，我们使用 $\mathbb{F}_\mathfrak{q}$ 中的 $\kappa d - d$ 个元素。现在，我们的元素指定了一个线性方程的解，其中所有相应的系数均已归零，且行列式为 1。

我们注意到，对于 $i = d-1,\ d$ 同样的编码操作也成立：在 $i = d-1$ 中我们跳过步骤 2，在 $i = d$中我们跳过步骤 3。最后，类似于 K_C 情形的计算表明，上文生成的比特表示长度与 $\log([SL_d(R) : K_i])$ 相差不超过 $O(\mathfrak{q}^{-\kappa})$。下文会稍微不拘形式地说电路接收 $\log(|SL_d(R)|/|K_i|)$ 比特作为输入，忽略这一微小差异；我们注意到，对于常数 d，该差异在 $|SL_d(R)|/|K_i|$ 中是逆多项式级别的小，因此在我们的所有应用中都可以接受。

推论 6.27 （典范代表元的电路复杂度）。

1. 一个电路 $C_{input \to mat}$，其输入为 $\log(|SL_d(R)|/|K_C|)$ 比特（分别为 $\log(|SL_d(R)|/|K_i|)$ 比特），并输出与这些比特对应的典范代表元 AK_C （分别为 AK_i）。*2.* 一个电路 $C_{mat \to can}$，其输入为矩阵 A，并输出典范矩阵 $A' \in AK_C$ （分别为 AK_i）。*3.* 一个电路 $C_{mat \to input}$，其输入为典范矩阵 $A' \in AK_C$ （分别为 AK_i），并以输入形式输出该矩阵。

这三个电路都是 可由对数空间一致规模为 $O(d^3\kappa^3 \text{对数}^2\,\mathfrak{q})$ 且深度为 $O(d^2 \text{对数}^2(d\kappa) \text{对数}\ \text{对数}\ \mathfrak{q}$的电路实现).

证明。 全文中，我们对 $\mathbb{F}_{\mathfrak{q}}$ 中的元素进行加法、乘法、求逆和比较。这些运算总是可以由对数空间一致、深度为 $O(\text{对数 对数 }\mathfrak{q})$、规模为 $O(\text{对数}^2\,\mathfrak{q})$ 的电路实现。因此，我们在假设可用带有域运算门的电路完成这些运算的前提下统计规模和深度，最后只需将深度乘以 $O(\text{对数 对数 }\mathfrak{q})$，将规模乘以 $O(\text{对数}^2\,\mathfrak{q})$。

1. 上述算法以常数深度和 $O(d\kappa)$ 规模计算第一列，因为它只需复制输入。接下来，对 $d-2$ 列进行循环，在每次迭代中，我们在 $d\kappa$ 个元素中找到第一个非零元素（可能需要先进行从 E 到 $\tilde{E}$ 的基变换，耗费 κ^2 次操作和 $\log^2\kappa$ 深度），并标记后续列，使其在复制输出时不使用此寄存器。对于最后一列，我们需要求解一个含有 $d\kappa$ 个变量（以及 $d\kappa$ 个方程）的线性方程组。这可以在规模 $(d\kappa)^3$ 和深度 $\log^2(d\kappa)$ 内完成。总的来说，规模为 $O(d^3\kappa^3\log^2\mathfrak{q})$，深度为 $O(d^2\log^2(d\kappa)\log\log\mathfrak{q})$。

2. 这是通过所述的顺序算法完成的，该算法将任意 $A'\in AK_i$（或 AK_C）变换为标准形。我们注意到，从 E 到 $\tilde{E}$ 的基变换可以在深度 $O(\log\kappa+\log\log\mathfrak{q})$ 内完成。[49] 所有这些也可以在规模 $O(d^3\kappa^2\log^2\mathfrak{q})$ 和深度 $O(d^2\log^2(d\kappa)\log\log\mathfrak{q})$ 内完成。

3. 这与第二个电路完全相同，只是多了一层，用于输出矩阵的所有系数（零置系数除外）。深度和规模在渐近意义下保持不变。

□

备注 6.28. 我们注意到，在我们的两种码构造中，集合 S 均由多个不同 K_i的陪集索引；具体而言，在常数码率情形下，是所有 K_i 的陪集（对所有 $i\in C$ 而言），而在多对数码率情形下，则是所有 K_2 和K_3 的陪集。为了编码这一族集合，我们可以将索引 i 以及 K_i的某个陪集的位表示作为输入，然后使用 i 索引到上文定义的相应 K_i专用电路 $C_{input\to mat}$中。通过适当选择 d，我们可以确保需要编码的不同陪集类数量始终是 *2* 的幂，因此不会增加集合位表示中的任何冗余。这最多会使电路规模增加 d 倍，且不会对深度产生渐近影响。

6.7.4 KMS 复形元素的电路复杂度

下面我们讨论第 2.2 节和第 5.1 节中电路的实现，以证明断言 4.4 和断言 5.5。

对于一个 陪集 AK_J，我们将 $lex(AK_J)$ 定义为 AK_J 中按字典序排列的第一个矩阵 .

命题 6.29。 存在一个电路，以 $A'\in AK_J$（作为矩阵）为输入，并输出 $lex(AK_J)$，其规模为 $O(|K_J|d^3\kappa\log\mathfrak{q}) = \mathfrak{q}^{O(d^2)}d^3\kappa\log\mathfrak{q}$，深度为 $O((\log\log\mathfrak{q}+\log d+\log\kappa)\log|K_J|) = O((\log\log\mathfrak{q}+\log d+\log\kappa)d^2\log\mathfrak{q})$。

证明。 该电路将 A' 与 K_J 中的所有矩阵并行相乘（找出整个陪集 $A'K_J$），然后通过锦标赛树比较它们。第一步是并行完成的，因此深度等于域 $\mathbb{F}_{\mathfrak{q}^\kappa}$ 上矩阵乘法的深度，即规模为 $|K_J|d^2\kappa\log\mathfrak{q}$，深度为 $\log^2(d)(\log\kappa+\log\log\mathfrak{q})$。锦标赛树的规模为 $|K_J|\cdot(d^2\kappa\log\mathfrak{q})$（即矩阵数量乘以矩阵作为位串的长度）。其深度为 $\log|K_J|\cdot\log(d^2\kappa\log\mathfrak{q})$。断言得证。 □

断言 6.30。 固定 $\mathfrak{q}$、κ、d，令 $\mathfrak{X}$ 为由这些参数确定的复形。则

1. 对于每一对颜色 $C_1\supseteq C_2$，存在一个电路 $C_{C_1\searrow C_2}$，其输入是面 $s=AK_{C_1}$（作为矩阵）的一个代表元 A'。它输出颜色 C_2的面 $t\subseteq s$ 的一个代表元。该电路可用常数规模和深度实现。*2.* 对于每一对颜色 $C_1\subseteq C_2$，存在一个电路 $C_{C_1\nearrow C_2}$，其输入是面$s=AK_{C_1}$（作为矩阵）的一个代表元 A' 和一个索引 i。它输出颜色 C_2的第 i 个面 $t\supseteq s$ 的一个代表元。该电路可用规模 $\mathfrak{q}^{O(d^2)}d^3\kappa\log\mathfrak{q}$和深度 $O((\log\log\mathfrak{q}+\log d+\log\kappa)d^2\log\mathfrak{q})$实现。

[49]这是一次方便的基变换，因为唯一不同的单项式是从 1 变为 t^d。因此，在一个方向上，如果多项式的常数项为 a，我们向多项式中加上 $\phi^{-a}(0)\phi(t)$。在另一个方向上，我们在多项式中将 at^d 替换为 $a(t^d-\phi(t))$，然后合并各项。

3. 对于每一对不相交颜色 C_1, C_2，存在一个电路 $C_{C_1 \to C_2}$，其输入是面 $s = AK_{C_1}$（作为矩阵）的一个代表元 A'和一个索引 i。它输出 $\mathfrak{X}[C_1]$ 与 $\mathfrak{X}[C_2]$ 交换游走中的第 i 个邻居$t \sim s$ 的一个代表元。该电路可用规模 $\mathfrak{q}^{O(d^2)} d^3 \kappa \log \mathfrak{q}$ 和深度$O((\log\log \mathfrak{q} + \log d + \log \kappa) d^2 \log \mathfrak{q})$实现。

$\mathfrak{X}$中的面的链接也同样如此。

证明。第一个电路是平凡的：它只输出输入。回想一下，如果 $C_1 \supseteq C_2$，那么 $K_{C_1} \subseteq K_{C_2}$。此外，$t \subseteq s$ 当且仅当 t 的陪集包含 s 的陪集。因此，包含$AK_{C_1} = A'K_{C_1}$ 的 K_{C_2} 的唯一可能陪集是 $A'K_{C_2}$。

第二个电路 C_2内。我们 需要更加仔细地处理。令 B_1、B_2、.....、B_m 为 K 的不同陪集的代表元执行以下步骤：

1. 我们找到 $A'' = lex(AK_{C_1})$。

2. 我们将 $A''B_i$ 相乘并输出结果。

这样做可行的原因如下。首先，由于我们找到 $lex(AK_{C_1})$，结果与代表元A'无关。其次，我们注意到 $t \supseteq s$ 当且仅当陪集 $B'K_{C_2}$ 包含于 AK_{C_1}中。因此，我们需要输出这样一个陪集。我们注意到，$A''B_iK_{C_2}$ 包含于 AK_{C_1} 中当且仅当 $B_iK_{C_2} \subseteq A''^{-1}A_{C_1}$。由于 A 和A'' 属于同一个陪集，$A''^{-1}A \in K_{C_1}$，因此我们给出了正确答案。我们注意到，所有 B_i 都代表不同陪集这一事实，也意味着 $A''B_i$ 代表不同陪集。最后，电路的规模等于求出 $lex(AK_{C_1})$ 的电路规模，加上代表元的数量乘以矩阵乘法电路的规模。第一步的规模开销最大，根据命题 6.29，为 $\mathfrak{q}^{O(d^2)} d^3 \kappa \log \mathfrak{q}$。深度同样主要由第一步决定，即 $O((\log\log \mathfrak{q} + \log d + \log \kappa) d^2 \log \mathfrak{q})$。

给定第二个电路，第三个电路同样是平凡的。从 $s \in \mathfrak{X}[C_1]$ 出发，我们使用第二个电路得到面 $s \cup t \in \mathfrak{X}[C_1 \cup C_2]$，然后输出 $t \in \mathfrak{X}[C_2]$（使用第一个电路）。规模和深度与第二个电路相同。

最后，我们指出，处于 $\mathfrak{X}_x$ 而非 $\mathfrak{X}$ 并不会造成实质差异，因为如果 $x \in \mathfrak{X}[C_3]$，我们总可以改用 $s \cup x \in \mathfrak{X}[C_1 \cup C_3]$ 而非 $s \in \mathfrak{X}_x[C_1]$，并在前两个电路中将另一种颜色设为 $C_2 \cup C_3$，在第三个电路中设为 C_2。□

现在，我们可以在第 2.2 节和第 5.1 节中实现主要电路了。

6.7.5 命题 4.3 的电路

下面的解释证明了断言 4.4。

这些电路 $C_{V \to S}$ **和** $C_{S \to V}$ 它们只是这样的电路：输入来自 $V = \mathfrak{X}[1]$（分别是 $S = \mathfrak{X}[2] \cup \mathfrak{X}[3]$）的一个代表元，并输出一个颜色为 2 或 3（分别为 1）的邻居。除了指定该邻居应具有颜色 2 或 3 的位置（在 $C_{V \to S}$ 中）外，这就是断言 6.30 中的第三个电路。特别是，它们可以用规模为 $\mathfrak{q}^8 \kappa \log \mathfrak{q}$、深度为 $O(\log \mathfrak{q}(\log \log \mathfrak{q} + \log \kappa))$ 的电路实现。代入参数后，规模为 $\mathrm{poly}(\frac{1}{\varepsilon}, \log N)$，深度为 $O(\log\log N \log_{\frac{1}{\varepsilon}} N)$。

这些电路 $C^{-1}_{V \to S}$，$C^{-1}_{S \to V}$在这个电路中，我们得到 $A' \in AK_i$ 的代表元以及一个相邻陪集BK_j，并且需要找到 i，使得 $C_{V \to S}(A', i) \in BK_j$。我们首先找到 $lex(AK_i)$。然后我们知道存在一个不同的 i，使得 $lex(AK_i) \cdot B_i \in BK_j$；当且仅当 $lex(AK_i) B_i B^{-1}$ 属于 K_j 时，这一条件成立。我们通过测试所有代表元 B_i 来找到它。这样做代价很高，但也不超过寻找 $lex(AK_i)$ 的代价，因此可以用规模为 $\mathfrak{q}^8 d^3 \kappa \log \mathfrak{q}$、深度为 $O(d^2 \log \mathfrak{q}(\log \log \mathfrak{q} + \log d + \log \kappa))$ 的电路实现。代入参数后，规模为 $\mathrm{poly}(\frac{1}{\varepsilon}, \log N)$，深度为 $O(\log\log N \log_{\frac{1}{\varepsilon}} N)$。

该电路 $C_{S \cap S \to V}$。该电路获取 i 和两个相邻的 s，s' 分别表示为 $A_2 \in AK_2$ 和 $A_3 \in A'K_3$。如果我们知道某个 $A'' \in AK_2 \cap A'K_3$，那么就可以直接使用第三个电路来计算 A''的一个邻居。为了找到这个代表元，我们只需在 AK_2中进行穷举搜索；与字典序搜索一样，该搜索可实现为规模 $\mathfrak{q}^8 \kappa \log \mathfrak{q}$、深度 $O(\log \mathfrak{q}(\log \log \mathfrak{q} + \log \kappa))$的电路。代入参数后，其规模为$\mathrm{poly}(\frac{1}{\varepsilon}, \log N)$，深度为 $O(\log\log N \log_{\frac{1}{\varepsilon}} N)$。

该路由电路 $C_{\mathcal{O}}$。该电路获取 AK_i，$A'K_j$ 并需要计算 A 与 A'之间的路径。我们改为计算单位元与 $B = A^{-1}A'$ 之间的路径，然后只需乘以 A。我们首先执行高斯消元，并找出如何将 B 表示为 $O(1)$-幺幂初等矩阵

$$B = e_{i_1,j_1}(r_1)e_{i_2,j_2}(r_2)\dots e_{i_k,j_k}(r_k)$$

然后通过第 6.4 节中的路由方案分别分解这些部分。请注意，第 6.4 节中的算法是递归的。实现它所需的规模就是递归树的规模，而深度就是树的深度。因此，所需规模为 $O(\kappa^{\log_2 5})$ 次矩阵乘法，加上递归中的 $O(\kappa^{\log_2 5})$ 次计算。考虑域运算后，规模为 $O(\kappa^3 \log^2 \mathfrak{q})$，深度为 $O(\log\ \kappa \log\log\ \mathfrak{q})$。代入参数后，规模为 $\mathrm{poly}(\log\ \frac{1}{\varepsilon}, \log\ N)$，深度为 $O(\log\log\ N \log_{\log_\varepsilon N})$。

访问解码器要进行解码，我们需要查询含噪码字或列表。为此，我们需要将要查询的相关矩阵表示转换为输入比特，这可以通过推论 6.27 完成。其规模为 $O(\kappa^2 \log^2 \mathfrak{q})$，深度为 $O(\log^2 \kappa + \log\log\ \mathfrak{q})$。代入参数后，得到 $O((\log\log\ N)^2)$。

编码注意，用于编码的电路只接收一对 $s,\ i$（作为输入比特，而不是矩阵），以及对函数 f 的预言机访问，并必须输出 $f(v)$，其中 v 是 s 的第 i 个邻居。如果按照上文所述的方法来做，我们会通过推论 6.27 计算出 s 的规范表示，A，调用 $C_{S\to V}(A,\ i) = B$，并输出 $f(B)$。这可以在 $\mathrm{poly}(\frac{1}{\varepsilon}, \log\ N)$ 时间内完成（因为这是电路规模的上界）。

然而，这对我们的目的来说还不完全够，因为我们声称我们的码具有 $\mathrm{polylog}(N)\log\ \frac{1}{\varepsilon}$ 的编码时间（事实上，这种类型的编码局部性对于我们在低复杂度困难性放大中使用的强局部可计算码是必要的）。为了达到这一要求，我们在第 6.7.7 节中设计了另一种编码电路（以及解码器访问方式）。这些电路满足所需的电路规模界。

6.7.6 命题 5.4 的电路

下面转到断言 5.5。

这些电路 $C_{V\to U}, C_{U\to V}, C^{-1}_{V\to U}, C^{-1}_{U\to V}, C_{U\cap U\to V}, C_{U\to S}$的实现方式与上一小节中断言 4.4 的电路相同。这些操作的规模和深度由求字典序最小值所主导。根据命题 6.29，这一步的规模为 $\mathfrak{q}^{O(d^2)} d^4 \kappa \log\ \mathfrak{q}$，深度为 $O(d^2 \log\ \mathfrak{q}\ (\log\log\ \mathfrak{q} + \log\ d + \log\ \kappa)$。代入参数后，其规模为 $(\log\ N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度为 $\tilde{O}(\log\log\ N \exp(O(\frac{1}{\varepsilon})))$。

这些电路 $C_{U;S\to V}, C_{U;V\to S}, C^{-1}_{U;S\to V}, C^{-1}_{U;V\to S}, C_{U;S\cap S\to V}$. 这些是“内部电路”：我们先指定 u，然后需要在 In_u 中执行集合系统操作。幸运的是，对于任意 $u \in U$，这仅意味着所有操作都在 $\mathfrak{X}_u$ 中执行，因此这些操作继承了相应全局操作的相同电路复杂度。例如，对于全局电路 $C_{S\to V}$（它只需为编码而实现，解码过程并不需要它），我们会取某个 $s \in \mathfrak{X}^C(0)$，并输出某个 $v \in s$（换言之，就是一个表示 C-面且包含作为 $\mathfrak{X}^C(0)$ 中一个顶点的 s 的矩阵）。对于局部电路 $C_{U;S\to V}$，我们只需对链接 $\mathfrak{X}_u$执行相同操作。利用断言 6.30 的相同方法即可实现，只不过是在 u 的链接中进行。因此，这些电路也可以实现为规模 $(\log\ N)^{\exp(O(\frac{1}{\varepsilon}))}$、深度为 $\tilde{O}(\log\log\ N \exp(O(\frac{1}{\varepsilon})))$。

该电路 $C^T_{\mathcal{O}}$。其复杂度基本上等于上一节中 $C_{\mathcal{O}}$ 的复杂度，再加上在采样 T 时进行试错所需的 $\mathrm{poly}(\log\ N)\exp(\frac{1}{\varepsilon})$ 深度和规模。代入参数后，其规模仍为 $(\log\ N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度为 $\tilde{O}(\log\log\ N \exp(O(\frac{1}{\varepsilon})))$。此外，按照该算法可以看出，它会向 T-预言机发出 $\exp(O(\frac{1}{\varepsilon}))\ \mathrm{poly}(\log\ N)$ 次查询，并且在完成 u-路径 P_0 的采样后，可以并行执行所有这些查询。

编码 编码的方式与上一小节完全相同。找到标准形所需的时间为 $\mathrm{poly}(\log\ N)\exp(\frac{1}{\varepsilon})$，同样，应用电路 $C_{S\to V}$ 所需的时间为 $\mathrm{poly}(\log\ N)\exp(\frac{1}{\varepsilon})$。根据推论 6.27 和断言 6.30，电路的深度为 $\exp(O(\frac{1}{\varepsilon}))\log^2(\log\ N)$。

访问解码器（解码） 与上文一样，我们通过找到标准形，然后将其转换回输入比特来实现。这同样需要规模 $(\log\ N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度为 $\tilde{O}(\log\log\ N\exp(O(\ \frac{1}{\varepsilon})))$。

6.7.7 命题 4.3 的改进编码和解码

为结束本节，我们为命题 4.3 构造改进的编码和解码（访问解码器）电路。为此，让我们显式写出此情形下所用的子群，其中 $d=3$。我们不使用集合记号，而只写出相应子群中矩阵的格式，其中 $\alpha,\beta,\gamma\in\mathbb{F}_{\mathfrak{q}}$。

$$K_1:\begin{pmatrix}1&0&0\\ \beta t&1&\alpha\\ \gamma t&0&1\end{pmatrix},\ K_2:\begin{pmatrix}1&\alpha&0\\ 0&1&0\\ \beta t&\gamma t&1\end{pmatrix},\ K_3:\begin{pmatrix}1&\alpha&\beta\\ 0&1&\gamma\\ 0&0&1\end{pmatrix}$$

以及这些交集

$$K_{\{1,2\}}:\begin{pmatrix}1&0&0\\ 0&1&0\\ \alpha t&0&1\end{pmatrix},\ K_{\{1,3\}}:\begin{pmatrix}1&0&0\\ 0&1&\alpha\\ 0&0&1\end{pmatrix},\ K_{\{2,3\}}:\begin{pmatrix}1&\alpha&0\\ 0&1&0\\ 0&0&1\end{pmatrix}.$$

利用这些格式以及 $x\in YK$ 当且仅当 $XY^{-1}\in K$ 这一事实，容易验证下列矩阵是 K_3 内 $K_{\{1,3\}}$ 的互异代表元。

$$A_{13\subseteq 3}(\beta,\gamma)\coloneqq\begin{pmatrix}1&\beta&\gamma\\ 0&1&0\\ 0&0&1\end{pmatrix}.$$

由于这样的矩阵共有 $\mathfrak{q}^2=\frac{|K_3|}{|K_{\{1,3\}}|}$ 个，我们得出结论： K_3 内的每一个陪集 $AK_{\{1,3\}}$ 都有一个这种形式的代表元。[50]

类似地，对每一对不同的 $k,j\in\{1,2,3\}$，我们定义 $A_{kj\subseteq j}$，使得包含于子群 K_j 中的 $K_k\cap K_j$ 的每一个陪集都有唯一一个这种形式的矩阵。它们是，

$$A_{23\subseteq 3}\coloneqq\begin{pmatrix}1&0&\beta\\ 0&1&\gamma\\ 0&0&1\end{pmatrix},\ A_{23\subseteq 2}\coloneqq\begin{pmatrix}1&0&0\\ 0&1&0\\ \beta t&\gamma t&1\end{pmatrix},\ A_{13\subseteq 1}\coloneqq\begin{pmatrix}1&0&0\\ \beta t&1&0\\ \gamma t&0&1\end{pmatrix}$$

and

$$A_{12\subseteq 2}\coloneqq\begin{pmatrix}1&\beta&0\\ 0&1&0\\ 0&\gamma t&1\end{pmatrix},\ A_{12\subseteq 1}\coloneqq\begin{pmatrix}1&0&0\\ \beta t&1&\gamma\\ 0&0&1\end{pmatrix}$$

因此，对于 $d=3$ 和颜色 $C_1=\{j\}$， $C_2=\{k,j\}$，我们可以实现 $C_{C_1\nearrow C_2}(A,\ i)$，其中 $i=(\beta,\ \gamma)$，其中电路：

1. 找到 规范表示 $A'=can(AK_j)$。
2. 输出 $A'\cdot A_{kj\subseteq j}(\beta,\ \gamma)$。

令 $s=AK_j$。和之前一样，由于这些是 $K_{\{k,j\}}\leq K_j$ 的陪集的不同规范表示，这确实给出了 $K_{\{k,j\}}$ 的所有相邻陪集 $t\supseteq s$ （因为它输出了所有满足 $B_i\in K_j$ 的陪集 $A'B_iK_{\{k,j\}}$ ）。根据推论 6.27，该电路的规模为 $\tilde{O}(\log^2 N\log^2\frac{1}{\varepsilon})$，深度为 $O(\log^2 N+\log\ \frac{1}{\varepsilon})$。[51] 对于 $C_1=\{k\},\ C_2=\{j\}$，电路 $C_{C_1\to C_2}$ 也具有相同性质，因此我们可以用规模为 $\tilde{O}(\log^2 N\log^2\frac{1}{\varepsilon})$、深度为 $O(\log^2 N+\log\ \frac{1}{\varepsilon})$ 的电路实现 $C_{V\to S}$, $C_{S\to V}$，进而意味着编码可以在此时间内完成，且具有相应的电路复杂度。

我们也来展示如何实现**解码**电路（即访问解码器），换句话说，实现 $C_{S\to V}^{-1}$ 的电路。求逆 $C_{S\to V}$ 归结为以下问题。给定一个“集合”

[50]事实上，如果我们要为这些子群定义标准形，那么这就是标准形。[51]假设 $\kappa\leq\log\ N$ 和 $\mathfrak{q}=\mathrm{poly}\log N\,\mathrm{poly}(\varepsilon)$。

代表元 $A' \in AK_j$ 和邻接"顶点"的代表元 $B' \in BK_1$，求矩阵 $A_{1j\subseteq j}(\beta,\ \gamma)$，使得 $can(AK_j) \cdot A_{1j\subseteq j}(\beta,\ \gamma) \in BK_1$。原因如下：$C_{S\to V}$ 接收 AK_j 的一个代表元（由一个比特向电路指示 $j \in \{2,3\}$），以及某个 $i = (\beta,\ \gamma)$，并输出邻接 K_1 陪集的代表元 $can(AK_j) \cdot A_{1j\subseteq 1}(\beta,\ \gamma)$。

因此，要找到这样的 β, γ，我们只需：

1. 计算 $A_c = can(AK_j)$ 和 $B_c = can(BK_1)$ 2. 将 $B_c^{-1}A_c$相乘。 3. 找到 β, γ，使得 $(B_c^{-1}A_c) \cdot A_{1j\subset j}(\beta,\gamma) \in K_1$，并输出 $i = (\beta,\gamma)$。

最后一步会得到正确的 i，因为 $A_c \cdot A_{1j\subset j}(\beta,\gamma) \in B_cK_1 = BK_1$ 当且仅当 $B_c^{-1}A_c \cdot A_{1j\subset j}(\beta,\gamma) \in K_1$。

最后一步只需通过求解一组线性等式即可完成：显然，$B_c^{-1}A_c \cdot A_{1j\subset j}(\beta,\gamma)$ 的各项关于 β,γ 都是线性的。此外，K_1 是 $\mathbb{F}_{\mathfrak{q}}$ 上的一个仿射子空间，因此很容易将其转化为关于 $O(\kappa)$ 个变量、在 $\mathbb{F}_{\mathfrak{q}}$上的方程。

因此，设置 $\mathfrak{q} = \text{poly}(\log \frac{N}{\varepsilon})$ 和 $\kappa \le \log N$（以及 $d = 3$）后，第一步的规模为 $\text{poly}(\log N, \log \frac{1}{\varepsilon})$，深度为 $O(\log N(\log\log N + \log\log \frac{1}{\varepsilon}))$。第二步和第三步的规模为 $\text{poly}(\log N, \log \frac{1}{\varepsilon})$，深度为 $O(\log^2 N(\log\log N + \log\log \frac{1}{\varepsilon}))$。因此，总体规模为 $\text{poly}(\log N, \log \frac{1}{\varepsilon})$，深度为 $O(\log^2 N(\log\log N + \log\log \frac{1}{\varepsilon}))$

7 下界

本节证明，任何高码率（弱局部可计算）的 aLLDC 都必须在 N 中进行对数级别的查询。特别地，我们针对任意此类码，关于码率 R、容错能力 ε 和定义域大小 N，证明如下紧下界：

定理 7.1。 存在普适常数 c_1、$c_2 > 0$，使得以下结论成立。对于任意 $N \in \mathbb{N}$、$r \le N^{0.99}$ $S \subset \binom{[N]}{r}$ $\mathcal{C}_{([N],S)}$ .01 ϵ ℓ_{out} q $\ell_{out} \le$ N^{c_1} 和，如果是一个 -近似的 (，，)-*LLDC*，且列表大小为 $\exp()$，那么对于任意 $N^{-c_2} \le \varepsilon \le \frac{1}{4}$，$\mathcal{C}_{([N],S)}$ 的查询复杂度至少为

$$q \ge \Omega\left(\frac{\log N}{\varepsilon \log \frac{2}{R}}\right)$$

其中我们回顾一下，~~$R =$~~ $\frac{N}{|S|r}$ 是码的码率。

我们指出，假设 $\varepsilon \le \frac{1}{4}$并非论证所必需，可以将其替换为任意严格小于$_{12}$的常数。类似地，$r \le N^{0.99}$ 可以替换为任意 $r \le N^{1-\alpha}$，其中 $\alpha > 0$，代价是相应地减小列表大小中的常数 c_1。当 $R \ge 2$时，该界是平凡的（或未定义）；但在这种情况下，不存在这样的码（参见定义 7.3 之前的讨论）。最后，我们指出，该界适用于任意消息字母表。

由于任何具有编码局部性 r 的码都可以做成具有类似参数的直积码，将 R 替换为 $\frac{N}{|S|r}$ 后，上述结论对任何 r-局部可编码码都成立。[53] 作为直接推论，我们得到：任何码率与局部性均为 $\text{poly}(\log N, \frac{1}{\varepsilon})$ 的码，必须至少进行 $\tilde{\Omega}(\frac{\log N}{\varepsilon})$ 次查询，这与我们在第 4 节中的信息论上界相符，至多相差一个 ~~log log~~ N因子。[54]类似地，任何码率与局部性均为 $2^{\sqrt{\log N}}$的码必须进行 $\Omega(\frac{\sqrt{\log N}}{\varepsilon})$ 次查询，这与我们简单的子空间构造相符。

[52]也就是说，从 K_1 的矩阵中减去单位矩阵，得到的是一个线性子空间。[53]给定一个 r-局部可编码码 C，相应的直积码 C_{DP} 会在每个块中记录原始码 C 所依赖的消息的 r 值。为了解码码字 w，解码器通过读取任意代码块 w_s 中的 r比特，并据此计算出 C 处相应的编码符号，来模拟原始码 C 上的码字 $\tilde{w}$。如果 w ε计算出一个消息 f，那么 $\tilde{w}$也能做到，因此我们只需通过这种方式模拟 C的解码器，即可进行局部解码。[54]回顾一下，我们在第 4 节中的主要结果需要多对数$\frac{N}{\varepsilon}$ 次查询；但如果不考虑电路规模，只需采用暴力搜索找到的长度为 $\log(N)$ 的随机路径，而不是使用较长的强显式路由方案，就能轻松将其减少到 $\tilde{O}(\frac{\log(N)}{\varepsilon})$。

证明直觉 我们简要给出该下界背后的一些直觉。其核心在于，（近似）局部列表译码的关键困难在于：可能存在两个（或更多）与输入字 w相关的全局消息 f_1 和 f_2，但我们仅能通过查询获得目标周围的一个小局部视图 来区分它们。特别地，由于对任意给定的目标顶点 v，我们可能有 $f_1(v) \neq f_2(v)$，因此需要某种全局一致的方法，来确定 w 中围绕 v 的哪些编码符号对应于 f_1，哪些对应于 f_2。我们本质上论证：只有当解码器进行足够多的查询，将 v “链接”到码中良好比例的其他顶点，从而在全局上强制这种一致性时，这才是可能的；这大约需要 $q \geq \log N \log \frac{1}{R}$ 次查询（因为码率 R 控制了我们用来将 v 连接到任何其他 v' 的包含图的度）。在我们的下界中，这对应于我们将每个 v 链接到某个全局起始辅助信息 $s_0 \in S$ （从而经由 s_0彼此链接到 v' ）这一事实。

下界中额外的 $-\frac{1}{\varepsilon}$ 仅仅源于这样一个事实：对手还可以让输入码字中除 $O(\varepsilon)$ 比例以外的部分都受到噪声污染（然后将剩余部分随机分配到 f_1 和 f_2之间）。因此，解码器平均需要进行 $\frac{1}{\varepsilon}$ 次查询，才能获得一个关于 f_1的“有信息量”的查询，这就带来了额外的 $\frac{1}{\varepsilon}$ 乘法因子。

7.1 定理7.1的证明

在开始证明之前，我们需要先介绍一些相关的定义和对象。首先回顾直积码的标准包含图以及一些相关记号。

定义 7.2. 设 $\mathcal{C}_{([N],S)}$ 是一个直积码，其中 $S \subset \binom{[N]}{r}$。我们定义二部图 G^{Enc}，其中每个左顶点 $v \in [N]$ 对应一个消息坐标，每个右顶点 $s \in S$对应一个编码符号（我们称之为“码块”）。每个 $s \in S$ 都与组成该码块的 r左顶点相连。设 G^{Enc}中的平均左度为 $d_{\text{avg}} = |S|/N$。对于 $s \in S$，我们沿用前文中的略微滥用记号的方式，将 s 既表示该图中的一个右顶点，也表示 $\text{Nbr}(s)$，即与 s相 *incident* 的左顶点集合 $v \in [N]$；具体含义由上下文决定。对于 $v \in [N]$，我们用 $\text{Nbr}(v)$ 表示 u的右邻居集合，即所有包含 v的集合 $s \in S$。

继续之前，我们先进行几项不失一般性的简化。首先，我们注意到可以假设包含图的最大左度至多为 $100d_{avg}$。具体而言，给定一个参数如上的直积编码，根据马尔可夫不等式，G^{Enc} 中度大于 $100d_{\text{avg}}$ 的左顶点至多有 $N/100$ 个。本节余下部分中，我们稍微滥用一下记号，用 $[N]$ 表示原图 G^{Enc} 中度至多为 $d = 100d_{\text{avg}}$ 的 $\Omega(N)$ 个左顶点组成的集合。这一区别对下文的证明基本没有影响。

我们还可以假设 $q \geq \frac{1}{100\varepsilon}$，因为在所述范围内，针对适当选取的 c_1，c_2，以及任意固定常数 $C > 0$，[104] 中已经证明了这种形式的下界。否则，考虑通过选取 S中任意 ε 比例的部分来编码某个固定消息 f：$[N] \to \{0,1\}$，从而得到码字 w。由于 $r \leq N^{0.99}$，这意味着有 $\exp(O(N))$ 个 0.01-远消息 f 会产生相同的编码（即满足 $f \in \mathcal{B}_{1-\varepsilon}(w)$ 的消息），因此不存在具有所声称参数的这种码。类似地，我们也可以假设码率 $R \leq \frac{3}{2}$。

完成这些修改后，我们就可以介绍受损码字上的分布 $w \in (\{0,1\}^r)^S$，该分布用于证明我们的下界：[55]

定义 7.3。令 $\mathcal{D}^{\mathsf{Enc}}$ 为由以下采样过程生成的受损码字上的分布 $w \in (\{0,1\}^r)^{|S|}$：

$\mathcal{D}^{\mathsf{Enc}}$的采样过程

1. 随机选取消息 $\mathcal{F} = \{f_1, f_2\}$, $f_i \in \{0,1\}^N$。*2.* 选取一个测度为 4ε 的随机子集 $D \subset S$。*3.* 随机选取 $b \in \{1,2\}^{|S|}$。

[55]如前所述，我们的界适用于任意输入字母表。对于一般的 Σ，只需将其中两个符号视为 $\{0,1\}$，并在构造中忽略其余符号。

4. 对于每个 $s \in S$

(a) 对每个 $s \in D$：令 $w_s = f_{b_s}|_s$

(b) 对每个 $s \notin D$：令 $w_s = 1^r$

换句话说，我们在顶点集上随机选取两个函数 f_1，f_2，并令 w 在随机选取的码块中随机的 4ε 比例上对其中一个函数进行编码（否则就直接给出全 1 函数，我们将其视为不包含关于 f_i的信息）。注意，由于我们可以假设 $|S|$ 不会太小，因此结果码字 w 以极高概率会 ε-计算出 f_1 和 f_2。另一方面，我们将证明，任何以良好概率局部解码这些函数的 aLLDC，都必须进行大量查询（否则其列表大小将呈指数级增长）。

我们通过 Alice 和 Bob 之间的多轮博弈将其形式化。我们将证明（引理 7.4），一方面，如果存在一个好得不真实的直积码 $\mathcal{C}_{([N],S)}$，那么 Bob 就能以非可忽略的概率赢得博弈中的大多数轮次（具体而言，概率约为 Ω($\frac{1}{\ell_{out}}$)）。另一方面，我们将证明（引理 7.5），如果 Bob 受到 $\mathcal{C}_{([N],S)}$的局部性条件限制，那么除非发生概率为可忽略的 exp(−Ω($N^{0.005}$)) 的情况，否则他会输掉博弈中的大多数轮次。

下面的图 10 描述了 Alice（对手）与 Bob（猜测者）之间的博弈。该博弈有两个参数：轮次数 $T = N^{0.005}$ 和查询复杂度 q，后者控制 Bob 每轮发起的查询数量（模仿 aLLDC）。不失一般性，我们假设 N 能被 1000 整除。[56]

Alice/Bob（T，q）博弈：

1. 使用公共随机性（Alice 和 Bob 都能看到）：随机将 $[N]$ 划分为块 $\mathcal{B} = \{B_1, \ldots, B_{N/1000}\}$，每块大小为 1000。随机选取 $b \in \{1,2\}^{|S|}$。

2. 使用私有随机性:Alice 随机选取测度为 4ε 的 $D \subset S$、$\mathcal{F} := \{f_1, f_2\}$ 和 $f^* \in \mathcal{F}$.

3. 对于轮次 $i = 1$、.....、T:

(a) Alice 随机选择 $B_{u_i} \in \mathcal{B}$，并将 B_{u_i} 连同部分限制集合 $\mathcal{F}|_{B_{u_i}}$: $= \{f_1|_{B_{u_i}}, f_2|_{B_{u_i}}\}$ 按字典序发送给 Bob。(b) 最多进行 $\mathrm{II} = 1$、.....、$1000q$ 轮迭代：(b.1) Bob 选择码查询 $s \in S$，并将查询 s 发送给 Alice。(b.2) 如果 $s \in D$，Alice 揭示 s 所涉及的块：令 Blk(s): $= \{B_i \in \mathcal{B} \mid B_i \cap s \neq \emptyset\}$Alice 将部分限制集合（按字典序排列）$\mathcal{F}|_{\mathsf{Blk}(s)}$: $= \{f_1|_{\mathsf{Blk}(s)}, f_2|_{\mathsf{Blk}(s)}\}$ 发送给 Bob。(b.3) 如果 $s \notin D$，Alice 回复“失败”:

(c) Bob 从集合 $\mathcal{F}|_{B_{u_i}}$（来自 (a)）中选择一个限制

图 10：Alice 与 Bob 之间的博弈

如果 Bob 在博弈第 i 轮的最后一步 (c) 中选择的限制等于 $f^*|_{B_{u_i}}$，则称 Bob 赢得该轮次。

图 10 中博弈的一次随机博弈过程会在数对 $(\mathcal{F}, b)$ 上诱导出一个分布，其中 $\mathcal{F}$ 和 b 均为随机选取。进而，每个随机数对 $(\mathcal{F}, b)$ 都会诱导出一个码字向量 $w \in (\{0,1\}^r)^{|S|}$：当 $s \in D$ 时令 $w_s := f_{b_s}|_s$，当 $s \notin D$ 时令 $w_s = 1^r$。由随机

[56]我们可以舍弃顶点，以达到下一个 1000的倍数；当 N 足够大时，这对论证的影响可以忽略不计。

选取 $(\mathcal{F}, b)$ 所诱导的分布恰好就是上文定义的分布 $\mathcal{D}^{\mathsf{Enc}}$，因此从现在起，我们将用 $w \sim \mathcal{D}^{\mathsf{Enc}}$ 表示由博弈过程诱导出的一个随机 w。

定理 7.1 的证明分为两部分。我们首先证明，如果存在一个良好的 aLLDC，那么 Bob 在博弈中以不可忽略的概率赢得 75% 的轮次，该概率随码的列表大小变化。

引理 7.4. 假设 $\mathcal{C}_{([N],S)}$ 是一个0.01-近似$(\varepsilon, \ell_{out}, q)$-*aLLDC*。对于每个 $T \in \mathbb{N}$，都存在一种 *Bob* 的策略，使得 *Bob* 以至少$\Omega(\frac{1}{\ell_{out}})$的概率赢得$(T, q)$-博弈的至少$\frac{3}{4}T$轮。

另一方面，我们证明，当 q 较小时，Bob 的任何随机策略赢得超过 $0.6T$轮的概率都至多是指数级小的。

引理 7.5. 存在一个普适常数 $C' > 0$，使得对于 $T = N^{0.005}$和 $\frac{1}{100\varepsilon} \le q \le C' \log \varepsilon \, {}^{N}_{\log \frac{2}{R}}$，*Bob* 的任何（可能是随机的）策略赢得$(T, q)$-博弈超过 $0.6T$轮的概率至多为$3\exp(-N^{0.005}/100)$。

综合这些引理，可立即推出定理 7.1。

定理 *7.1* 的证明。为得出矛盾，假设存在一个 0.01-近似的 $(\varepsilon, q, \ell_{out})$-aLLDC，其参数为 $q \le C' \log \varepsilon \, {}^{N}_{\log \frac{2}{R}}$、$\ell_{out} \le \exp(N^{c_1})$，且$N^{c_2} \le \varepsilon \le \frac{1}{4}$对任意一对 $c_1 < c_2$都足够小，从而满足 [104, 定理 1.4]（前述 $q \ge \frac{1}{100\varepsilon}$ 下界）的陈述以及 C'引理 7.5 中的常数。根据引理 7.4，对任意 T，Bob 因此有一种策略，能以概率 $P = \Omega(\exp(-N^{c_1}))$ 赢得 (T, q)-博弈的 $\frac{3}{4}T$轮次。若 $q < \frac{1}{100\varepsilon}$，则由 [104] 即告完成。否则，令 $T = N^{0.005}$，引理 7.5 指出，Bob 没有任何策略能在 (T, q)-博弈中于超过 $0.6T$轮次上以大于 $3\exp(-N^{0.005}/100) < P$的概率获胜，只要 N充分大且 $c_1 < 0.005$，从而得到所需的矛盾。 □

7.1.1 引理 7.4 的证明

我们首先证明：在能够访问所声称的 aLLDC 的条件下，Bob 获胜概率的下界。

引理 *7.4* 的证明。设 $\mathcal{C}_{([N],S)}$ 为假定存在的直积码，并设 $\mathcal{A}$ 为随机化 aLLDC 解码算法。则 Bob 在该博弈中的策略如下。回顾标准放大（见第 2.1 节），我们可以在渐近意义上不改变该码参数的前提下假定，解码器输出电路列表 $C_1, \ldots, C_{\ell_{out}}$，使得对任意码字 w与 $f \in \mathcal{B}_{1-\varepsilon}(w)$，以 99% 的概率，某个$C_i$ 确定性地 0.1-计算 f（特别地，$\mathcal{A}$ 本身在多数放大之后随机固定了原始 C_i 的内部随机性）。

Bob 的策略： 首先，Bob 运行 $\mathcal{A}$ 以得到随机化电路 $C_1, \ldots, C_{\ell_{out}}$，并随机选择其中一个电路 C。

在博弈开始时，Alice 和 Bob 随机选择将 $[N]$ 划分为 $\frac{N}{1000}$ 个块，$\mathcal{B} = \{B_1, \ldots, B_{N/1000}\}$每个块的大小为 1000，并选择一个随机向量 $b = b_1$、.....、$b_{|S|}$，其中每个 $b_i \in \{1, 2\}$。每一轮中，Bob 的策略如下。在第 $i \in [T]$轮的步骤 3(a) 中，Alice 向 Bob 发送一个要解码的挑战块 $B_{u_i} \in \mathcal{B}$，并发送部分限制的集合（按字典序排列）$\{\rho_1 = f_1|_{B_{u_i}}, \rho_2 = f_2|_{B_{u_i}}\}$。现在，对于每个 $v \in B_{u_i}$，Bob 在输入 v 上模拟电路 C。每当 $C(v)$ 发起码字坐标查询 $s \in S$时，他就在步骤 (b.1) 中向 Alice 发送查询 s，Alice 则在步骤 (b.2) 中返回部分限制的集合（按字典序排列）$\{\sigma_1 = f_1|_{\mathsf{Blk}(s)}, \sigma_2 = f_2|_{\mathsf{Blk}(s)}\}$；若 $s \notin D$，则返回“失败”。这些限制对应于每个 f_i 在 s所涉及的块上的取值。随后，Bob 使用 b 计算对查询 w_s 的回答：在前一种情况下令 $w_s = f_{b_s}|_s$，在 Alice 回复“失败”时令 $w_s = 1^r$。Bob 以这种方式继续，直到完成对 C 在输入 v上的整个计算的模拟。注意，由于直积码的局部性，$C(v)$ 至多发起 q 次查询，因此他可以通过执行 q 次内循环 (b) 来模拟 $C(v)$ 的计算。由于他会对每个 $v \in B_{u_i}$模拟 C，因此在第 i轮中，他总共至多执行 $1000q$ 次内循环。

完成第 i 轮的 (b) 循环后，Bob 已为所有 $v \in B_{u_i}$ 模拟了 $C(v)$，从而生成一个字符串 $\rho' \in \{0,1\}^{1000}$。在第 i 轮的最后一步 (c) 中，他将该字符串 ρ' 与 Alice 在步骤 3(a) 中给出的字符串 ρ_1 和 ρ_2 进行比较，并输出与 ρ' 的汉明距离最近的字符串 ρ_i（若出现并列，则任意选择）。

现在我们证明，上文定义的 Bob 的策略以非可忽略概率赢得博弈的大多数轮次。为此，我们先定义一些有助于 Bob 赢得博弈的性质。

定义 7.6（非典型回合）。如果一局博弈的第 $i \in [T]$ 轮中，$f_1|_{B_{u_i}}$ 与 $f_2|_{B_{u_i}}$ 的归一化汉明距离小于 04，则称该轮为非典型轮。

定义 7.7（幸运）。如果以下事件均成立，则称这局博弈对 *Bob* 而言是幸运的：

(1) 由 $(\mathcal{F}、b)$ 引发的 $w \sim \mathcal{D}^{Enc}$ 随机选择满足 $\bar{f}^* \in \mathcal{B}_{1-\epsilon}(w)$。*(2)* 非典型轮的数量少于 $T/100$。*(3) Bob* 从 *aLLDC* 中选出的随机电路与 *Alice* 选择的 $f^* \in \mathcal{F}$ 的距离小于 01。

引理 7.8。假设存在具有预先指定参数的 *aLLDC*，则一局博弈对 *Bob* 而言是幸运的概率为 $\Omega(\frac{1}{\ell_{out}})$，

上述引理的证明使用以下断言。

断言 7.9. 在 $\mathcal{D}^{Enc}$ 上，$f^* \in \mathcal{B}_{1-\epsilon}(w)$ 的概率至少为 $1\,200$。

证明. 固定 Alice 对 f^* 的选择以及测度为 4ε的子采样域 $D \subset S$。对于每个 $s \in D$，$w_s = f^*|_s$ 的独立概率为 50%，因此 f^* 与 w 的期望一致度为 2ε。切尔诺夫界保证，当 $|S| \geq \Omega(\frac{1}{\varepsilon})$ 足够大时，$f^* \notin \mathcal{B}_{1-\varepsilon}(w)$ 的概率至多为 $\exp(-\Omega(\varepsilon|S|)) \leq \frac{1}{200}$。 □

断言 7.10. 对于任意足够大的 $T \in \mathbb{N}$ 和足够小的 ϵ，非典型回合数超过 $T/100$ 的概率至多为 $1\,200$。

证明. 我们论证博弈中任一特定轮次 i 为非典型的概率很小，因此通过对 T 个独立块应用切尔诺夫界，几乎所有轮次都以高概率为典型。

对于任意固定的轮次 i，根据切尔诺夫界，$dist(f_1|_{B_{u_i}}, f_2|_{B_{u_i}}) < 0.4$ 的概率至多为 e^{-10}，因为 f_1，f_2 之间的期望距离至多为$1\ 2$（对于更大的字母表，这个值可能更大），且$|B_{u_i}| = 1000$ 比特是均匀随机选取的。现在，由于博弈中的每个 T 块彼此独立，且非典型回合的期望数量至多为 $e^{-10}T$，根据第二个切尔诺夫界，非典型回合超过 $T/100$ 个的概率至多为$1\ 200$，只要 N 足够大。 □

现在我们可以证明引理 7.8 了。

引理 *7.8* 的证明。根据断言 7.9 和断言 7.10，事件（1）和（2）以 99% 的概率发生。

在这些事件发生的条件下，Bob 对（放大后的）aLLDC 的运行会输出一个电路列表 $\{C_1, \ldots, C_{\ell_{out}}\}$，其中某个 C_i 与 f^* 的距离为 0.1 以内的概率为 99%。由于 Bob 均匀随机地从这些电路中选取一个，选中 C_i 的概率为 $\frac{1}{\ell_{out}}$，因此博弈对 Bob 来说是幸运的概率至少为 $\frac{1}{2\ell_{out}}$。 □

现在我们可以完成引理 7.4 的证明。首先，我们说明，当博弈对 Bob 来说是幸运的时，Bob 以 90% 的概率赢得任意一个固定的典型轮。假设这一点成立，根据马尔可夫不等式即可得出结论：由于博弈幸运时 99% 的轮次都是典型轮，Bob 预期至少赢得 $0.8T$，而他赢得少于 $\frac{3}{4}T$ 轮的概率至多为 0。8。由于 Bob 幸运的概率为 $\Omega(\frac{1}{\ell_{out}})$，因此他以至少 $\Omega(\frac{1}{\ell_{out}})$ 的概率赢得 $\frac{3}{4}T$ 轮，正如所需。

剩下要证明的是，在一次幸运运行中，Bob 以 90% 的概率赢得一个固定的典型轮次。为此，注意到只要 Bob 选择的电路在该块上与 Alice 选择的 f^* 相差不超过 0.15，Bob 就会自动获胜；因为根据假设，$\{f_1, f_2\}$ 相差至少 0.4，所以此时 f^* 是距离更近的唯一函数。不对典型性进行条件化（但仍假设 Bob 的电路在全局上与 f^* 相差不超过 0.1）时，根据切尔诺夫界，它在给定块 B_{u_i} 上与 f^* 相差不超过 0.15 的概率至少为 $1 - e^{-8}$。另一方面，我们已在上文证明，B_{u_i} 是典型的概率至少为 $1 - e^{-10}$（并且该事件与 Bob 对电路的选择相互独立，因此在进行这种条件化后仍然成立）。因此，即使在典型性的条件下，Bob 的电路在 B_{u_i} 上与 f^* 相差不超过 0.15 的概率仍至少为 90%，证明完毕。

□

7.1.2 引理 7.5 的证明

现在我们需要通过反证法证明：当 q 小于指定参数时，除非发生可忽略概率的事件，否则 Bob 赢得的轮次不会超过 $0.6T$。

标注版 Alice/Bob 游戏：

1. 使用公共随机性（Alice 和 Bob 都能看到）：将 $[N]$ 随机划分为若干块 $\mathcal{B} = \{B_1, \ldots, B_{N/1000}\}$，每块大小为 b。随机选取 $b \in \{1,2\}^{|S|}$。设 $H = \emptyset$2. 使用私有随机性：Alice 随机选取测度为 4ε 的 $D \subset S$、$\mathcal{F} := \{f_1, f_2\}$ 和 $f^* \in \mathcal{F}$。3. 对于轮次 $i = 1, \ldots, T$：(a) Alice 随机选取 $B_{u_i} \in \mathcal{B}$，并将 B_{u_i} 连同部分限制集合 $\mathcal{F}|_{B_{u_i}} := \{f_1|_{B_{u_i}}, f_2|_{B_{u_i}}\}$ 一起发送给 Bob，按字典序排列。将 B_{u_i} 加入 H。(b) 至多进行 $\mathrm{II} = 1, \ldots, 1000q$ 次迭代：(b.1) Bob 选择码查询 $s \in S$。(b.2) 如果 $s \in D$，Alice 揭示被 s 触及的块：令 Blk(s) $:= \{B_i \in \mathcal{B} \mid B_i \cap s \neq \emptyset\}$。Alice 将部分限制集合（按字典序排列）发送给 Bob：$\mathcal{F}|_{\mathrm{Blk}(s)} := \{f_1|_{\mathrm{Blk}(s)}, f_2|_{\mathrm{Blk}(s)}\}$。对于所有 $B_i \in$Blk(s)：将 B_i 加入 H。(b.3) 如果 $s \notin D$，Alice 回复“失败”。(c) Bob 从集合 $\mathcal{F}|_{B_{u_i}}$（来自 (a)）中选择一个限制

图 11：Lazy Alice 和 Bob 之间的标注博弈

该下界背后的高层直觉在于：该博弈的每一轮次都是局部的，即 Bob 在任一给定轮次中所见的消息块数量很少，因为 Bob 对每个挑战仅做 $1000q$ 次查询，且每次查询只触及 r 个消息比特。设定 $T = N^{0.005}$，只要 q 与 r 很小，我们将论证：第 i 轮次中的挑战块 B_{u_i} 很可能“远离”任何先前被查询的信息——也就是说，即便做 $1000q$ 次查询，Bob 也无法把挑战块 B_{u_i} “链接”到任何先前信息。因此，Bob 无法将其答案与先前轮次相关联，最多只能猜测 B_{u_i} 上的 f_1 或 f_2 是否为 f^*，从而独立地有 $1\ 2$ 的机会输掉该轮次。只要几乎所有轮次都满足这一性质，基本的集中不等式便意味着：除了以 exp($-O(T)$) 的概率之外，Bob 无法赢得显著超过半数的轮次，从而给出关于 Bob 成功的所需上界。

引理 *7.5* 的证明。我们将证明，对于 Bob 的任意策略，他在博弈中获胜超过 $0.6T$ 轮的概率至多为 3 exp($-N^{0.005}/100$)。为了分析整个博弈过程中各挑战块之间的依赖关系，我们将用一个关联图 $G = (V,\ E)$为博弈添加标注：图中每个挑战块 $B_i \in \mathcal{B}$对应一个顶点 v_i；若存在一个代码块 $s \in S$，其顶点同时与块 B_i 和 B_j 相交，则这两个块之间有一条边。此外，在博弈的每一轮中，我们还会维护一个不断变化的顶点诱导子图 $H \subset G$，用于追踪哪些块已通过 Bob 的查询揭示。通常，我们只用定义该顶点诱导子图的顶点／块来指代它。

粗略地说，我们将论证子图 H 的非连通分量在博弈中是独立的（即 Alice 的秘密 f^* 在 H 的任一固定连通分量上的取值不依赖于它在其他分量上的取值）。然后我们论证，在大多数轮次中，由于图 G 是低度的，Alice 的随机挑战块 B_{u_i} 将离 H 非常远，特别是 Bob 将没有足够的查询来把 B_{u_i} 连通到 H 的任何分量。因此，f^* 在 B_{u_i} 上的取值将

独立于所有先前轮次，因此 Bob 所能做的不会好过单纯猜测，并以概率$1\ 2$ 独立于他在先前轮次中的胜负而失败。标准集中界进而保证，除了以 exp(−$O(T)$) 的概率之外，Bob 无法赢得远多于 $\frac{T}{2}$ 个轮次。关于标注博弈的描述见图 11。

更形式地说，在博弈开始时，顶点诱导子图 H 为空。在每一轮次中，Alice 选取一个随机挑战块 B_{u_i}，并将其加入 H，而 Bob 最多进行 $1000q$ 次查询 $s \in S$。对于每个被查询的 $s \in D$，我们将每一个与 s 相交的块 B_j（记作 Blk(s)）加入 H。为方便起见，我们用 H_i 表示第 i轮次结束时当前的顶点诱导子图 H，并对一个块 $B \in \mathcal{B}$ 用$H_i(B)$ 表示 H_i 中包含 B的连通分量。

我们将证明，在大多数轮次中，挑战块 $\mathcal{B}_{u_i}$ 与 H_{i-1}的距离至少为 $8000\varepsilon q$，并且 Bob 不太可能进行这么多次"有信息量的"查询（即 D中的查询）。因此，Bob 无法将该块与任何先前的挑战链接起来，并且在该轮次中有独立的$1\ 2$失败概率。

定义 7.11（Alice 的典型与幸运）。 我们称博弈的第 i 轮次为典型，如果

- *Bob* 在 D 中查询的集合不超过 $8000\varepsilon q$ 个，并且
- B_{u_i} 与 H_{i-1} 的距离小于 $7000\varepsilon q$，

如果一局博弈中非典型轮次不超过 $\frac{T}{50}$ 轮，我们就称该局博弈对 *Alice* 来说是幸运的。

我们将 Bob 在整个博弈中的胜负序列称为他的胜负记录。现在我们说明，Bob 赢得一轮典型轮次的概率为$1\ 2$，且与截至该时刻的胜负记录无关：

命题 7.12。 *Bob* 赢得一轮典型轮次的概率为 $1/2$，且与他之前的胜负记录无关。

证明。 根据定义，每次查询 $s \in D$ 都会向子图 H中添加一个团，而每次查询 $s \notin D$ 都不会添加任何内容。因此，如果 B_{u_i} 与 H_{i-1} 的距离至少为 $8000q\varepsilon$，且 Bob 在 D 中至多进行 $7000q\varepsilon$ 次查询，则 $\mathcal{B}_{u_i}$在 H_i 中必定与每个先前的挑战 $\mathcal{B}_{u_j}$ 不连通（回忆一下，根据构造， H_{i-1} 包含每个先前的挑战块）。

尚需说明：如果 $\mathcal{B}_{u_i}$ 在子图 H_i 中与所有先前的挑战块都不连通，那么 Bob 获胜的概率$1\ 2$ 与他此前的胜负记录无关。粗略地说，这是显然的，因为 f^*（Alice 的秘密函数）在连通分量 $H_i(B_{u_i})$ 中的消息比特上的取值，与它在所有其他连通分量上的取值完全独立。

形式化地说，固定 Alice 对 $f^* = f_1$ 或 $f^* = f_2$ 的选择（设为 f_1），并固定任意可能的

$$\mathcal{F}|_{H_i(B_{u_i})} = \{f_1|_{H_i(B_{u_i})}, f_2|_{H_i(B_{u_i})}\}$$

Alice 在第 i 轮向 Bob 发送[57]。注意，这并未确定 f_1, f_2 本身的取值；它们等概率地是 $\mathcal{F}|_{H_i(B_{u_i})}$ 中的两个函数之一。Bob 必须判断 f^* 是这个按字典序排列的集合中的第一个函数还是第二个函数。

关键在于观察到，对 Bob 的任何先前胜负记录进行条件化，都不会进一步影响 Alice 对 f_1, f_2 在 H_i(B_{u_i}) 上的选择——换言之，在 Alice 的随机性下， f^* 是 $\mathcal{F}|_{H_i(B_{u_i})}$ 中两个固定函数之一的概率仍恰好为$1\ 2$，因此无论 Bob 采用什么策略，他获胜的概率都是$1\ 2$，且与此前的胜负记录无关。这直接源于：根据定义，固定一份胜负记录就固定了 f_1, f_2 在每个先前挑战块上的取值。这些信息可能会影响 $\mathcal{F}|_B$ 在通过 Bob 的查询与 B_{u_j} 连通的块中是哪一个函数，但不会影响 f_1 在 $H_i(B_{u_j})$ 的消息比特上的取值，因为它与其他所有连通分量都不连通（也就是说，Bob 从不查询同时包含 $H_i(B_{u_j})$ 和 $H_i(B_{u_i})$ 中消息比特的任何 s，所以固定 f_1 在前者上的取值，不会影响 f_1 在后者上的取值）。因此，在任何此类条件化下， f_1 的两种可能取值仍然都成立，并且在 Alice 的随机性下，各自以$1\ 2$ 的概率独立选出，正如所需。

□

现在我们论证，除可忽略概率外，该博弈对 Alice 来说是幸运的。

命题 7.13. 除概率至多为 $2\exp(-T/3)$ 外，该博弈对 *Alice* 来说是幸运的。

[57]更准确地说，Alice 分别给出每个 Blk(s) 的值，但如果 Bob 得到的是聚合值，对他来说只会更容易。

证明。 我们需要证明，超过 $\frac{T}{50}$ 轮为非典型的概率至多为 2 exp(−T/3)。首先注意，根据切尔诺夫界（无放回），在 Alice 对 $D \subset S$ 的选择下，Bob 至多进行 $7000\varepsilon q$ $s \in D$ 次查询的概率至少为 $1 - \exp(-T/3)$。以此事件为条件。我们断言，除以下概率外，每一轮都是典型的：

$$\frac{1000T(1+8000\varepsilon qr)(10^5/R)^{8000\varepsilon q}}{N} + \exp(-1000\varepsilon q), \tag{10}$$

即使在对 Alice 和 Bob 先前各轮的选择进行条件化之后也是如此。由于我们假设 $q \geq \frac{1}{100\varepsilon}$，后一个项至多为 $\frac{1}{1200}$。另一方面，假设 $q \leq C' \frac{N}{\log \frac{2}{R}}$ 对足够小的 C' 成立，并代入 $T = N^{.005}$ 和 $r \leq N^{0.99}$，前一个项有如下界：

$$\begin{aligned}\frac{1000T(1+8000\varepsilon qr)(10^5/R)^{8000\varepsilon q}}{N} &\leq O\left(N^{-0.005}\log N(10^5 R^{-1})^{8000C'\frac{\log(N)}{\log\frac{2}{R}}}\right)\\ &\leq O(N^{C''-0.005}\log N)\end{aligned}$$

通过将 C' 取得足够小，我们可以使 $C'' > 0$ 任意小。取 $C'' < 0.005$，可得整个表达式有 poly($1/N$) 的上界；特别地，当 N 足够大时，其上界为 $\frac{1}{200}$。因此，当 N 足够大时，方程 (10) 总是以 $\frac{1}{100}$ 为上界。因此，非典型回合的总数随机占优于 Bin($T, \frac{1}{100}$)，而切尔诺夫界（以及并集界）给出了所需结果。

为证明方程 (10)，注意到根据我们的条件化，在博弈中的任意时刻，H 中的顶点数至多为 $T(1+8000\varepsilon qr)$，也就是说，包括 D 中的查询（每个查询最多可向 H 添加 r 个块）以及挑战块本身。另一方面，我们的图的最大度至多为 $1000dr = 10^5 R^{-1}$，因为每个块 B 有 1000 个消息比特，每个比特至多关联 d 个集合 $s \in S$，而每个 s 分别至多关联 r 个块。因此，G 中距离 H_i 为 $8000\varepsilon q$ 的顶点总数至多为

$$T(1+8000\varepsilon qr)(10^5 R^{-1})^{8000\varepsilon q}.$$

挑战块 B_i 被均匀随机选取，从而得到落在 H_{i-1} 附近的期望概率。与上文类似，切尔诺夫界也保证在每一轮次中，Bob 选取 D 中超过$7000\varepsilon q$ 个集合的概率至多为 exp(−1000εq)，符合预期。

□

证明现在基本上是直接的。Alice 是幸运的，除非以 2 exp(−T/3) 的概率例外；在此情况下，博弈中除 $\frac{1}{50}$ 比例之外的所有轮次都是典型的。在此事件条件下，Bob 赢得任意典型轮的概率为 $\frac{1}{2}$，且与先前轮次无关。因此，Bob 赢得的轮次总数被 $\frac{T}{50} + \frac{49}{50}\text{Bin}(\frac{1}{2}, \frac{49}{50}T)$ 随机占优，而切尔诺夫界表明 Bob 赢得超过 $0.6T$ 轮次的概率至多为 exp(−T/100)。对这些事件取并集界即完成证明。

□

致谢

我们非常感谢 Shashank Srivastava 就列表译码和列表恢复进行的多次有益讨论，包括向我们介绍 Guruswami 和 Indyk 的相关著作；也感谢 Swastik Kopparty、Noga Ron-Zewi 和 Mary Wootters 与我们讨论现代并行和局部译码算法。我们还感谢 Irit Dinur、Avi Wigderson 和 Ronen Shaltiel 就高维扩展体和困难性放大进行的有益讨论，并感谢 Izhar Oppenheim 向我们介绍 Kaufman-Oppenheim 高维扩展体构造的 KMS 变体。最后，感谢 Gil Melnik 协助制作本文中的各种图示。

8 参考文献

[1] Peter Elias。噪声信道中的列表译码，1957年。

[2] John M Wozencraft。列表译码。季度进展报告，48:90–95，1958年。

[3] Madhu Sudan。《超越纠错界的 Reed–Solomon 码解码》。复杂性期刊，13(1):180–193，1997年。

[4] Russell Impagliazzo。《有一定难度问题的硬核分布》。载于*IEEE 第36*届计算机科学基础年会论文集，第538–545页。IEEE，1995年。[5] Ryan O'Donnell。《NP 内的困难性放大》。载于第三十四届 *ACM* 计算理论研讨会论文集，第751–760页，2002年。[6] Ran Raz 和 Shmuel Safra。《次常数错误概率的低次测试，以及 NP 的次常数错误概率 PCP 刻画》。载于第二十九届 *ACM* 计算理论研讨会论文集，第475–484页，1997年。[7] Sanjeev Arora 和 Madhu Sudan。《改进的低次测试及其应用》。载于第二十九届 *ACM* 计算理论研讨会论文集，第485–495页，1997年。[8] Johan Håstad。《一些最优不可近似性结果》。*ACM*会刊（*JACM*），48(4):798–859，2001年。[9] Jin-Yi Cai、Aduri Pavan 和 D Sivakumar。《关于永久式的困难性》。载于计算机科学理论方面年度研讨会，第90–99页。Springer，1999年。[10] Shuichi Hirahara 和 Nobutaka Shimizu。《困难性自放大：简化、优化与统一》。载于第*55*届 *ACM* 计算理论研讨会论文集，第70–83页，2023年。[11] Luca Trevisan 等。《提取器与伪随机生成器》。*ACM*期刊，48(4):860–879，2001年。[12] Venkatesan Guruswami、Christopher Umans 和 Salil Vadhan。《由 Parvaresh–Vardy 码构造的非平衡扩展图和随机性提取器》。*ACM*会刊（*J ACM*），56(4):1–34，2009年。[13] Russell Impagliazzo 和 Avi Wigderson。《若 E 需要指数规模电路，则 P=BPP：去随机化异或引理》。载于第二十九届 *ACM* 计算理论研讨会论文集，第220–229页，1997年。[14] Madhu Sudan、Luca Trevisan 和 Salil Vadhan。《不使用异或引理的伪随机生成器》。载于第三十一届 *ACM* 计算理论研讨会论文集，第537–546页，1999年。[15] Andrew C Yao。《陷门函数的理论与应用》。载于第*23*届计算机科学基础年会（*SFCS 1982*），第80–91页。IEEE，1982年。[16] Leonid A Levin。《单向函数与伪随机生成器》。载于第十七届 *ACM* 计算理论研讨会论文集，第363–365页，1985年。[17] 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* 计算理论研讨会论文集，第25–32页，1989年。[18] Benny Chor、Eyal Kushilevitz、Oded Goldreich 和 Madhu Sudan。《隐私信息检索》。*ACM*会刊（*JACM*），45(6):965–981，1998年。[19] Marco L Carmosino、Russell Impagliazzo、Valentine Kabanets 和 Antonina Kolokolova。《从自然证明中学习算法》。载于第*31*届计算复杂性会议（*CCC 2016*），第10–1页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2016年。[20] Andrej Bogdanov、Manuel Sabin 和 Prashant Nalini Vasudevan。《异或码与稀疏含噪学习奇偶性》。载于第三十届 *ACM-SIAM* 离散算法研讨会论文集，第986–1004页。SIAM，2019年。[21] Guy Blanc、Yuhao Huang、Tal Malkin 和 Rocco A. Servedio。《恶劣噪声真的比恶意噪声更难吗？》载于第*37*届 *ACM-SIAM* 离散算法研讨会（*SODA 2026*）论文集，2026年。即将发表。[22] Victor Vasilievich Zyablov 和 Mark Semenovich Pinsker。《擦除信道中传输用低密度码的译码复杂度》。*Problemy Peredachi Informatsii*，10(1):15–28，1974年。

[23] Mi chael Sipser an d Daniel A S pielman. Expander co des. *IEEE t ransac ti ons on Information Th eory*, 42(6):1710–1722，2002年。[24] Tali Kaufman 和 Izhar Oppenheim。高阶随机游走：超越谱隙。*Combinatorica*，第1–37页，2020年。[25] Nima Anari、Kuikui Liu、Shayan Oveis Gharan 和 Cynthia Vinzant。对数凹多项式 II：高维游走与用于计算拟阵基数的全多项式时间近似方案。在第*51*届 *ACM SIGACT* 计算理论研讨会论文集中，第1–12页，2019年。[26] Pavel Panteleev 和 Gleb Kalachev。渐近良好的量子码与局部可测试经典低密度奇偶校验码。*arXiv*预印本 *arXiv:2111.03654*，2021年。[27] Anurag Anshu、Nikolas P Breuckmann 和 Chinmay Nirkhe。由良好量子码构造的非局域至平凡态哈密顿量。在第*55*届 *ACM* 计算理论研讨会论文集中，第1090–1096页，2023年。[28] Irit Dinur、Shai Evra、Ron Livne、Alexander Lubotzky 和 Shahar Mozes。具有常数码率、距离和局部性的局部可测试码。*arXiv*预印本 *arXiv:2111.04808*，2021年。[29] Yotam Dikstein、Irit Dinur 和 Alexander Lubotzky。通过有界度辛高维扩展体实现的低接受率一致性测试。*arXiv*预印本 *arXiv:2402.01078*，2024年。[30] Mitali Bafna、Noam Lifshitz 和 Dor Minzer。具有小可靠性的常数度直积测试器。*arXiv*预印本 *arXiv:2402.00850*，2024年。[31] Mitali Bafna、Dor Minzer 和 Nikhil Vyas。基于高维扩展体构造的小可靠性准线性规模概率可检验证明。*arXiv*预印本 *arXiv:2407.12762*，2024年。[32] Jun-Ting Hsieh、Alexander Lubotzky、Sidhanth Mohanty、Assaf Reiner 和 Rachel Yun Zhang。显式无损顶点扩展图。*arXiv*预印本 *arXiv:2504.15087*，2025年。[33] Luca Trevisan。使用异或引理进行列表译码。在第*44*届 *IEEE* 计算机科学基础研讨会论文集，*2003*年中，第126–135页。IEEE，2003年。[34] Russell Impagliazzo。困难性即随机性：通用去随机化综述。*arXiv*预印本 *cs/0304040*，2003年。[35] Venkatesan Guruswami 和 Valentine Kabanets。通过空间高效直积实现困难性放大。在拉丁美洲理论计算机科学研讨会中，第556–568页。Springer，2006年。[36] Russell Impagliazzo、Ragesh Jaiswal、Valentine Kabanets 和 Avi Wigderson。一致直积定理：简化、优化与去随机化。在第*40*届 *ACM* 计算理论研讨会论文集中，第579–588页，2008年。[37] Russell Impagliazzo、Ragesh Jaiswal 和 Valentine Kabanets。直积码的近似列表译码与一致性困难性放大。*SIAM* 计算学杂志，39(2):564–605，2009年。[38] Swastik Kopparty、Or Meir、Noga Ron-Zewi 和 Shubhangi Saraf。具有次多项式查询复杂度的高码率局部可纠错码与局部可测试码。*ACM*会刊（*JACM*），64(2):1–42，2017年。[39] Sivakanth Gopi、Swastik Kopparty、Rafael Oliveira、Noga Ron-Zewi 和 Shubhangi Saraf。逼近 Gilbert–Varshamov 界的局部可测试码与局部可纠错码。*IEEE* 信息论汇刊，64(8):5813–5831，2018年。[40] Swastik Kopparty、Noga Ron-Zewi、Shubhangi Saraf 和 Mary Wootters。折叠 Reed-Solomon 码与重数码的改进解码。在*2018*年*IEEE*第*59*届计算机科学基础年会（*FOCS*）中，第212–223页。IEEE，2018年。

[41] Noga Alon、Jehoshua Bruck、Joseph Naor、Moni Naor 和 Ron M Roth。通过伪随机图构造渐近良好的低码率纠错码。*IEEE* 信息论汇刊，38(2):509–516，2002年。[42] Noga Alon、Jeff Edmonds 和 Michael Luby。具有近乎最优恢复能力的线性时间擦除码。在*IEEE* 第*36*届计算机科学基础年会论文集中，第512–519页。IEEE，1995年。[43] Venkatesan Guruswami 和 Piotr Indyk。可在线性时间内编码和列表解码的码。在第*35*届 *ACM* 计算理论年会论文集中，第126–135页，2003年。[44] Brett Hemenway、Noga Ron-Zewi 和 Mary Wootters。高码率张量码的局部列表恢复及其应用。*SIAM* 计算学杂志，49(4):FOCS17–157，2019年。[45] Fernando Granha Jeronimo、Tushant Mittal、Shashank Srivastava 和 Madhur Tulsiani。使用扩展图构造逼近广义 Singleton 界的显式码。在第*57*届 *ACM* 计算理论研讨会论文集中，第843–854页，2025年。[46] Shashank Srivastava 和 Madhur Tulsiani。在线性近似时间内将基于扩展子的码列表译码至容量。*arXiv*预印本 *arXiv:2504.20333*，2025年。[47] Robert Gallager。低密度奇偶校验码。*IRE* 信息论汇刊，8(1):21–28，2003年。[48] Jonathan Katz 和 Luca Trevisan。关于纠错码局部解码过程的效率。在第*32*届*ACM*计算理论年会论文集中，第80–86页，2000年。[49] Richard M Karp 和 Richard J Lipton。非一致复杂性类与一致复杂性类之间的一些联系。在第*12*届 *ACM* 计算理论年会论文集中，第302–309页，1980年。[50] Luca Trevisan 和 Salil Vadhan。通过一致归约研究伪随机性与平均情形复杂度。计算复杂性，16(4):331–364，2007年。[51] Lijie Chen 和 Roei Tell。基于极难函数的简单快速去随机化：以几乎零成本消除随机性。在第*53*届 *ACM SIGACT* 计算理论年会中，第283–291页，2021年。[52] Ryan Williams。强指数时间假设分离与 Merlin 和 Arthur：批量求值的短非交互式证明。*arXiv*预印本 *arXiv:1601.04743*，2016年。[53] Venkatesan Guruswami 和 Piotr Indyk。高效可解码码的基于扩展子的构造。在第*42*届 *IEEE* 计算机科学基础研讨会论文集中，第658–667页。IEEE，2001年。[54] Walter J. Savitch。非确定性与确定性带复杂度之间的关系。计算机与系统科学杂志，4(2):177–192，1970年。[55] Irit Dinur、Prahladh Harsha、Tali Kaufman、Inbal Livni Navon 和 Amnon Ta-Shma。使用双重采样器进行列表译码。*SIAM* 计算学杂志，50(2):301–349，2021年。[56] Irit Dinur 和 Tali Kaufman。高维扩展图蕴含一致性扩展子。在*2017*年 *IEEE* 第*58*届计算机科学基础年会（*FOCS*）中，第974–985页。IEEE，2017年。[57] Yotam Dikstein 和 Irit Dinur。分层集合系统上的一致性测试定理。在*2019*年 *IEEE* 第*60*届计算机科学基础年会（*FOCS*）中，第1495–1524页。IEEE，2019年。[58] Laura Grave de Peralta 和 Inga Valentiner-Branth。由 Kac–Moody–Steinberg 群构造的高维扩展图。欧洲组合学杂志，126:104131，2025年。

[59] Tali Kaufman 和 Izhar Oppenheim. 新型局部谱高维扩张器的构造. 载于第*50*届 *ACM SIGACT* 计算理论年会论文集，第773–786页，2018年.[60] Irit Dinur、Prahladh Harsha、Tali Kaufman、Inbal Livni Navon 和 Amnon Ta Shma. 使用双重采样器进行列表译码. 载于第*30*届 *ACM-SIAM* 离散算法研讨会论文集，第2134–2153页. SIAM，2019年.[61] Mitali Bafna 和 Dor Minzer. 用于几乎处处可靠传输的常数度网络. 载于第*57*届 *ACM* 计算理论年会论文集，第1319–1328页，2025年.[62] Roy Gotlib 和 Tali Kaufman. 从余边界扩张得到列表一致性扩张. *arXiv* 预印本 *arXiv:2210.15714*，2022年.[63] Yotam Dikstein 和 Irit Dinur. 小可靠性参数区域内高维扩展图的一致性定理：覆盖的作用. *arXiv* 预印本 *arXiv:2308.09582*，2023年.[64] Mitali Bafna 和 Dor Minzer. 通过余边界扩张刻画直积测试. 载于第*56*届 *ACM* 计算理论年会论文集，第 1978–1989 页，2024 年 .[65] Tali Kaufman 和 Alexander Lubotzky. 边传递 Ramanujan 图与高度对称的 LDPC 优码. *arXiv* 预印本 *arXiv:1108.2960*，2011年.[66] Tali Kaufman 和 Alexander Lubotzky. 高维扩展图与性质测试. 载于第*5*届理论计算机科学创新会议论文集，第501–506页，2014年.[67] Yotam Dikstein、Irit Dinur、Prahladh Harsha 和 Noga Ron-Zewi. 通过高维扩展图构造局部可测试码. *arXiv* 预印本 *arXiv:2005.01045*，2020年.[68] Irit Dinur、Siqi Liu 和 Rachel Yun Zhang. 高维扩展图上的新码. *arXiv* 预印本 *arXiv:2308.15563*，2023年.[69] Nikolas P Breuckmann 和 Jens N Eberhardt. 平衡积量子码. *IEEE* 信息论汇刊，67(10):6653–6674，2021年.[70] Uriya A First 和 Tali Kaufman. 关于由高维扩展图上的层构造的良好二查询局部可测试码. *arXiv* 预印本 *arXiv:2208.01778*，2022年.[71] Yotam Dikstein 和 Max Hopkins. 高维扩展体上的 Chernoff 界与反向超收缩性. *arXiv* 预印本 *arXiv:2404.10961*，2024年.[72] Shai Evra、Tali Kaufman 和 Gilles Zémor. 利用高维扩展图构造突破平方根距离屏障的可译码量子 LDPC 码. 载于*2020*年*IEEE*第*61*届计算机科学基础年会（*FOCS*），美国北卡罗来纳州达勒姆，*2020*年*11*月*16*日至*19*日，第218–227页，2020年.[73] Tali Kaufman 和 Ran J. Tessler. 由张量构造的新余边扩张器蕴含距离为 $\Omega(\sqrt{n}\log^k n)$ 的显式量子 LDPC 码. 载于*STOC '21*：第*53*届 *ACM SIGACT* 计算理论年会，线上会议，意大利，*2021*年*6*月*21*日至*25*日，第1317–1329页，2021年.[74] Anthony Leverrier 和 Gilles Zémor. 量子 Tanner 码. 载于*2022*年 *IEEE* 第*63*届计算机科学基础年会（*FOCS*），第872–883页. IEEE，2022年.[75] Kyle Gulshen 和 Tali Kaufman. 具有横向门的量子 Tanner 色码. *arXiv* 预印本 *arXiv:2510.07864*，2025年.[76] Vedat Levi Alev、Fernando Granha Jeronimo、Dylan Quintana、Shashank Srivastava 和 Madhur Tulsiani. 直和码的列表译码. 载于第*14*届 *ACM-SIAM* 离散算法研讨会论文集，第1412–1425页. SIAM，2020年.

[77] Fernando Granha Jeronimo、Shashank Srivastava 和 Madhur Tulsiani. 通过可分割正则性在近线性时间内译码 Ta-Shma 码. 载于第*53*届 *ACM SIGACT* 计算理论年会论文集，第1527–1536页，2021年.
[78] Swastik Kopparty 和 Shubhangi Saraf. 高错误率下随机线性码的局部列表译码与测试. 载于第*42*届 *ACM*计算理论研讨会论文集，第417–426页，2010年.[79] Swastik Kopparty、Shubhangi Saraf 和 Sergey Yekhanin. 具有次线性时间译码的高码率码.*ACM*会刊（*JACM*），61(5):1–20，2014年.[80] Alan Guo、Swastik Kopparty 和 Madhu Sudan. 通过提升构造的新型仿射不变码. 载于第*4*届理论计算机科学创新会议论文集，第529–540页，2013年.[81] Alan Guo. 通过提升构造高码率局部可纠正码. *IEEE* 信息论汇刊，62(12):6672–6682，2015年.[82] Alan Guo 和 Swastik Kopparty. 提升码的列表译码算法. *IEEE* 信息论汇刊，62(5):2719–2725，2016年.[83] Brett Hemenway、Rafail Ostrovsky 和 Mary Wootters. 扩展码的局部可纠正性.信息与计算，243:178–190，2015年.[84] 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.[85] Manuel Blum 和 Silvio Micali. 如何生成密码学强伪随机比特序列. 载于为密码学奠定坚实基础：论 *Shafi Goldwasser* 和 *Silvio Micali* 的工作，第227–240页. IEEE，2019年.[86] L aszl o Babai、Lance Fortnow、Noam Nisan 和 Avi Wigderson. 除非 EXPTIME 具有可发表证明，否则 BPP 可由次指数时间模拟. 计算复杂性，3(4):307–318，1993年.[87] Johan Håstad、Russell Impagliazzo、Leonid A Levin 和 Michael Luby. 从任意单向函数构造伪随机生成器. *SIAM* 计算学杂志，28(4):1364–1396，1999年.[88] Christopher Umans. 适用于所有困难度的伪随机生成器. 载于第*34*届 *ACM* 计算理论年会论文集，第627–634页，2002年.[89] Ronen Shaltiel 和 Christopher Umans. 适用于所有最小熵的简单提取器与一种新的伪随机生成器. *ACM*会刊（*JACM*），52(2):172–216，2005年.[90] Dean Doron、Dana Moshkovitz、Justin Oh 和 David Zuckerman. 从困难性中获得近乎最优的伪随机性. *ACM*期刊，69(6):1–55，2022年.[91] Shlomo Hoory、Nathan Linial 和 Avi Wigderson. 扩展图及其应用. 美国数学会通报，43(4):439–561，2006年.[92] Venkatesan Guruswami 和 Madhu Sudan. 某些级联码的列表译码算法. 载于第*32*届*ACM*计算理论年会论文集，第181–190页，2000年.[93] Yotam Dikstein、Irit Dinur、Yuval Filmus 和 Prahladh Harsha. 高维扩展图上的布尔函数分析. 载于近似、随机化与组合优化：算法与技术（*APPROX/RANDOM 2018*）. 达格斯图尔-莱布尼茨信息学中心，2018年.[94] Laszlo Csanky. 快速并行矩阵求逆算法. 载于第*16*届计算机科学基础年会（*SFCS 1975*），第11–12页. IEEE，1975年.[95] Aryeh Grinberg、Ronen Shaltiel 和 Emanuele Viola. 带辅助信息的自适应过程下的不可区分性，以及困难性放大证明的下界. 载于*2018*年*IEEE*第*59*届计算机科学基础年会（*FOCS*），第956–966页. IEEE，2018年.

[96] Michael Luby。极大独立集问题的一个简单并行算法。载于第十七届 ACM 计算理论年会论文集，第1–10页，1985年。[97] Vedat Levi Alev、Fernando Granha Jeronimo 与 Madhur Tulsiani。高维扩展图上约束满足问题的近似。载于2019年 IEEE 第60届计算机科学基础年会（FOCS），第180–201页。IEEE，2019年。[98] Tom Gur、Noam Lifshitz 与 Siqi Liu。高维扩张器上的超收缩性。arXiv 预印本 arXiv:2111.09375，2021年。[99] Vedat Levi Alev 与 Ori Parzanchevski。顺序扫描与高维扩张。arXiv 预印本 arXiv:2312.02089，2023年。[100] Ryan O'Donnell 与 Kevin Pratt。由 Chevalley群构造的高维扩展图。载于第37届计算复杂性会议，2022年。[101] Max Hopkins。高维扩展体上的超收缩性 II：对称化与 q-范数。载于第57届 ACM 计算理论年会论文集，第96–103页，2025年。[102] Peter Abramenko。孪生建筑及其在 S-算术群中的应用。Springer，2006年。[103] Izhar Oppenheim 与 Inga Valentiner-Branth。由 KMS群得到的新余循环高维扩张图。arXiv 预印本 arXiv:2504.05823，2025年。[104] Sergei Artemenko 与 Ronen Shaltiel。表明困难性放大的非均匀与自适应归约的查询复杂度下界。《计算复杂性》，23(1):43–83，2014年。[105] Michael Alekhnovich。多项式上的线性丢番图方程与里德-所罗门码的软判决译码。载于第43届 IEEE 计算机科学基础年会，2002年。论文集，第439–448页。IEEE，2002年。[106] Avraham Ben-Aroya、Klim Efremenko 与 Amnon Ta-Shma。关于放大局部可解码码错误容忍度的注记。载于计算复杂性电子研讨会（ECCC），TR10-134，2010年。[107] Luca Trevisan 与 Madhur Tulsiani（记录）。第8讲：Goldreich–Levin定理：学习线性函数。CS294 讲义：伪随机性与组合构造，2005年9月。[108] Geoffrey Mon、Dana Moshkovitz 与 Justin Oh。具有常数查询复杂度且码率近乎最优的近似局部可译码码。载于2024年 IEEE 国际信息论研讨会（ISIT），第2838–2843页。IEEE，2024年。[109] Sanjeev Arora 与 Shmuel Safra。概率可检验证明：NP 的一种新刻画。《ACM会刊（JACM）》，45(1):70–122，1998年。[110] Sergey Yekhanin 等。局部可解码码。《理论计算机科学基础与趋势®》，6(3):139–255，2012年。[111] Salil P Vadhan 等。伪随机性。《理论计算机科学基础与趋势®》，7(1–3):1–336，2012年。[112] Roei Tell。线性阈值电路的量词化去随机化。载于第50届 ACM SIGACT 计算理论研讨会论文集，第855–865页，2018年。[113] Victor Vasilievich Zyablov 与 Mark Semenovich Pinsker。Gallager低密度码纠错复杂度的估计。Problemy Peredachi Informatsii，11(1):23–36，1975年。[114] Rohan Goyal、Prahladh Harsha、Mrinal Kumar 与 Ashutosh Shankar。一元重数码与折叠里德-所罗门码的快速列表译码。载于2024年 IEEE 第65届计算机科学基础年会（FOCS），第328–343页。IEEE，2024年。

[115] Swastik Kopparty。重数码的列表译码。《计算理论》，11(1):149–182，2015年。[116] Swastik Kopparty、Nicolas Resch、Noga Ron-Zewi、Shubhangi Saraf 与 Shashwat Silas。论高码率张量码的列表恢复。《IEEE 信息论汇刊》，67(1):296–316，2020年。[117] Daniel A Spielman。线性时间可编码且可译码的纠错码。载于第二十七届 ACM 计算理论年会论文集，第388–397页，1995年。[118] Alexander Lubotzky、Ralph Phillips 与 Peter Sarnak。拉马努金图。Combinatorica，8(3):261–277，1988年。[119] Moshe Morgenstern。对每个素数幂 q 的 $q+1$ 正则拉马努金图的存在性与显式构造。《组合论杂志》B辑，62(1):44–62，1994年。[120] Emanuele Viola 与 Avi Wigderson。局部扩张图。《计算复杂性》，27(2):225–244，2018年。[121] Oded Goldreich、Ronitt Rubinfeld 与 Madhu Sudan。通过查询学习多项式：高噪声情形。《SIAM离散数学期刊》，13(4):535–570，2000年。[122] Venkatesan Guruswami 等。列表译码中的算法结果。《理论计算机科学基础与趋势®》，2(2):107–195，2007年。[123] Ronitt Rubinfeld 与 Madhu Sudan。多项式的稳健刻画及其在程序测试中的应用。《SIAM 计算学杂志》，25(2):252–271，1996年。[124] Sergey Yekhanin。迈向次指数长度的三查询局部可译码。《ACM会刊（JACM）》，55(1):1–16，2008年。[125] Klim Efremenko。次指数长度的三查询局部可译码。载于第四十一届 ACM 计算理论年会论文集，第39–44页，2009年。[126] Tali Kaufman 与 Madhu Sudan。稀疏随机线性码是局部可译码且可测试的。载于第48届 IEEE 计算机科学基础年会（FOCS'07），第590–600页。IEEE，2007年。[127] Eli Ben-Sasson、Oded Goldreich、Prahladh Harsha、Madhu Sudan 与 Salil Vadhan。邻近性稳健概率可检验证明、更短的概率可检验证明及其在编码中的应用。载于第三十六届 ACM 计算理论年会论文集，第1–10页，2004年。[128] Jeremiah Blocki、Venkata Gandikota、Elena Grigorescu 与 Samson Zhou。简报：计算受限信道中的松弛局部可纠正码。载于第45届自动机、语言与编程国际学术讨论会（ICALP 2018），第106–1页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2018年。[129] Tom Gur、Govind Ramnarayan 与 Ron Rothblum。松弛局部可纠正码。《计算理论》，16(1):1–68，2020年。[130] Tom Gur 与 Oded Lachish。论松弛局部译码算法的能力。《SIAM 计算学杂志》，50(2):788–813，2021年。[131] Gil Cohen 与 Tal Yankovitz。松弛局部可译且可纠正码：超越张量化。载于2022年 IEEE 第63届计算机科学基础年会（FOCS），第24–35页。IEEE，2022年。[132] Guy Goldberg。线性松弛局部可译且可纠正码不需要自适应性与双侧误差。载于第51届自动机、语言与编程国际学术讨论会（ICALP 2024），第74–1页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2024年。[133] Vinayak M Kumar 与 Geoffrey Mon。由局部测试得到的松弛局部可纠正性。载于第56届 ACM 计算理论年会论文集，第1585–1593页，2024年。

[134] Gil Cohen 和 Tal Yankovitz。具有 (log n)ˆ(2+ o (1)) 次查询的渐近良好 RLCC。载于第 *39* 届计算复杂性会议（*CCC 2024*），第 8–1 页。Schloss Dagstuhl–Leibniz-Zentrum für Informatik，2024年。

A 字母表与局部性归约

在正文中，我们构造了码率良好、查询复杂度低的码，但其（对数）字母表较大，且局部编码时间随 poly(ε^{-1}, log(N)) 增长。本节中，我们使用标准级联方法的变体，将这些码转化为二元、局部可计算的码。我们研究两种情形。首先，如果只需实现二元码，并且可以接受 poly(ε^{-1}，log(N)) 的局部性，那么只需与几乎任意一种具有 RNC^2 解码能力的良好列表可译码码进行级联（例如，经典上由级联里德-所罗门码 [92, 105]得到的那些码）。由此可得第 4 节中的以下推论。为方便起见，我们在此重述，并将在本节末尾证明：

推论 A.1（二元 aLLDC）。 对于所有足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个显式二元码 $\mathcal{C}$：$\{0,1\}^N \to \{0,1\}^{N'}$，其是 （0.01）-近似的 （$\frac{1}{2}+\varepsilon, \ell_{out}, \bar{q}$）-列表可译码码，且

1. 码率： $R \geq$ 多项式（$\frac{1}{\log(N)\ \varepsilon}$）
2. 查询复杂度： $q \leq$ 多项式（log（N） $\frac{1}{\varepsilon}$）
3. 列表大小： $\ell_{out} \leq$ 多对数（$\frac{1}{\varepsilon}$）$\leq^2$

此外， $\mathcal{C}$ 可以在 *TIME*（多项式（log（N） $\frac{1}{\varepsilon}$））内进行局部编码，并由规模为多项式（$\frac{1}{\varepsilon}$、$\log N$）且深度为 $\tilde{O}\left(\log^2\left(\frac{\log N}{\varepsilon}\right)\right)$的对数空间一致电路进行解码。[58]

次多项式（推论 3.2）和常数（推论 5.2）区间中的二元码的证明本质上完全相同（不过，对于前者，首先要与 [36] 的去随机化子空间码进行级联，以从编码局部性中去除 $2^{\sqrt{\log(N)}}$ 项。），因此我们略去证明。

本节的第二个目标是通过级联构造二元码，使其能够在 $\frac{1}{\varepsilon}$ 中以多对数时间 进行局部计算。这对于在 P^{NP} 等自然复杂度类中实现指数困难性放大至关重要。实现这一目标所需的思路大多仍是标准的，但由于我们对 δ_{in} 的要求很高，并且要达到最优查询复杂度和列表大小，因此需要做一些调整：

推论 A.2 (局部可计算的二元 aLLDC)。 对于所有充分大的 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，存在一个显式二元码 $\mathcal{C}$：$\{0,1\}^N \to \{0,1\}^{N'}$，它是0.01-近似($\frac{1}{2}+\varepsilon, \ell_{out}, q$)-列表可译码的，且满足

1. 码率： $R \geq \frac{1}{\text{多对数}\ N} \cdot 2^{-O(\log^2(1/\varepsilon))}$
2. 查询复杂度： $q \leq$ 多对数(N, $\frac{1}{\varepsilon}$) $\varepsilon^2$3. 列表大小： $\ell \leq$ 多对数 $\frac{\frac{1}{\varepsilon}}{\varepsilon^2}$

此外， $\mathcal{C}$ 可以在时间（多项式对数(N) $\cdot$ 对数^2 $\frac{1}{\varepsilon}$)) 内进行局部编码，并由规模为多项式($\frac{1}{\varepsilon}$, 对数 N)、深度为 $O\left(\text{对数}\left(\frac{\text{对数}\ N}{\varepsilon}\right)\right.$ 对数对数(N)$\Big)$的对数空间一致电路进行列表解码。

在继续之前，有几点值得说明。首先，从积极的一面看，推论 A.2 的查询复杂度关于 ε 与 [95] 的依赖基本上是最优的。然而，码率却大幅下降了，从多项式降至准多项式。这是均匀困难性放大中令人遗憾但普遍存在的障碍（见 [36]），并非本文所要解决的问题。事实上，这个问题与实现对 N 的近线性依赖基本正交：如果已知关于 ε 依赖更优、且对 N 甚至只有次指数依赖的均匀 aLLDC，我们便可在自己的框架中直接使用它们，以类似的方式改善对 ε 的依赖，同时保持对 N 的近线性依赖。

[58]这里的 $\tilde{O}$ 隐藏着对数对数($\frac{\log N}{\varepsilon}$) 因子。

证明概述 在继续之前，我们先简要概述局部可计算情形下的一般证明策略。从高层次来看，基本思路与标准的码级联相同：我们只需将原始码中的每个大符号替换为其自身的内层 aLLDC，使得每当“外码”想要查询某个符号（的一部分）时，就调用（小字母表的）内码来解码出（可能正确的）值。 [36] 采用了这一策略，通过与一个小型完备复形复合，降低其子空间构造的字母表大小和局部性。

不幸的是，该方法无法直接用于证明推论 A.2，原因有二。首先，回顾第 3.1 节可知，出于技术原因，我们需要内码解码到精度 $\delta_{out} \ll {}^{1}\log(N)$，因此特别地还必须具有 $\delta_{in} \ll {}^{1}\log(N)$。若借助直积内码来做到这一点，就需要使用规模至少为 $\overline{\log}(N)$ 的集合，这实际上使字母表缩减失去意义（更具体地说，其规模过大，以至于无法再与 Hadamard 级联）。有鉴于此，我们需要一种内码，使字母表大小与该 δ 无关。

幸运的是，一旦脱离直积范式，构造这样的码并不太困难。具体而言，暂且略去一些细节，这可以通过复合两种已知码的变体来实现：一种二元直和码，可从 ε-一致度解码到 99%-一致度；以及第二种基于困难性放大方法的二元非直积 码，可从 99%-一致度解码到$1-\delta_{out}$ 一致度。尽管后者仍需要编码局部性 $O({}^{1}_{\delta_{out}})$，但关键在于我们能够在整个过程中保持二元字母表。

第二个主要问题是，已知的 aLLDC 串接定理是有损的，会在查询和列表大小上造成较大的多项式损失。我们通过引入一种新的强化近似局部列表可恢复码模型来处理这一问题，称之为“单电路”模型（其本质是允许在电路之间复用查询），并进一步收紧标准串接论证，以支持在多个潜在误差尺度上进行解码，从而优化列表大小和查询复杂度。

A.1 单电路（高噪声）列表恢复

在正文中，我们通常在低噪声区域中考虑列表恢复：我们将高噪声下的列表译码归约为列表恢复，其中几乎每个索引都包含所需的函数 f。在级联中，我们还需要处理高噪声 列表恢复情形，因为只有一小部分内码会具有有效的列表译码。因此，运行这些内码会向另一个码生成高噪声列表恢复输入。此前，我们将列表恢复写成记号 $(1-\eta, \ell_{out}, q)$-LLRC，以强调低噪声区域。现在，我们将改为直接写成 $(\varepsilon, \ell_{out}, q)$-LLRC，以匹配我们高噪声局部可解码码定义中的记号；该记号仍然表示我们应当解码任意 $f \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L})$。

尽管上述内容在编码文献中已是标准做法，但为了在级联下实现最优参数，我们还需要一种新的、强化的列表译码/恢复表述，称为单电路解码。非正式地说，单电路近似局部列表可恢复码是一种近似局部列表可恢复码：它不输出一个由 ℓ_{out} 个电路组成的列表（其中一个电路被保证以良好概率解码目标消息 f），而是输出一个单个电路，其输出属于 $\Sigma_1^{\ell_{out}}$，并且输出中的某个索引被保证以良好概率计算 f。

定义 A.3（单电路近似局部列表可恢复码）。设 ε、δ、>0 和 ℓ_{in}、ℓ_{out}、$q \in \mathbb{N}$。设 $\mathcal{C}$： $\Sigma_1^m \to \Sigma_2^n$ 为一个码。若存在概率图灵机 $\mathcal{A}$，使其在输入 1^k 时输出一个（概率）电路 $C^{\mathcal{L}}$： $[m] \to \Sigma_1^{\ell_{out}}$，并且对于任意固定的 $f \in \mathcal{B}_{1-\varepsilon,\ \delta_{in}}(\mathcal{L})$，存在一个索引 $i \in [\ell_{out}]$，使得 $C^{\mathcal{L}}$ δ_{out}的第 i个输出计算 f 的概率良好，则称 $\mathcal{C}$ 为一个（δ_{in}、δ_{out}）-近似、 ℓ_{in}-输入、（ε、ℓ_{out}、q）-单电路近似局部列表可恢复码：

$$\forall f \in \mathcal{B}_{1-\varepsilon,\delta}(\mathcal{L}) : \Pr_{\mathcal{A}}\left[\exists i \in [\ell_{out}] : \Pr_{r,x\in[k]}\left[C^{\mathcal{L}}(x)_i = f(x)\right] \geq 1-\delta_{out}\right] \geq \frac{3}{4}$$

这里有几点值得说明。首先，为什么这是一个有用的定义？关键思路如下：在标准级联中，外码会调用内码来生成一个电路列表，而它通常需要检查列表中的每个电路，找出一个与目标函数一致的电路（否则就随机挑选一个，并寄希望于它是良好的）。如果这些电路各自使用独立的查询，这意味着外码必须为列表中的每个电路支付q_{inner} 次查询，使复杂度增加列表大小倍。另一方面，如果只有一个电路（它在某种意义上重复使用这些查询来计算所有列表元素），那么外部算法只需支付 q_{inner}，从而节省列表大小这一倍数。

最后，当输入列表大小为 1 时，我们将上述码称为单电路近似局部列表可译码码；当 δ_{in}，$\delta_{out}=0$ 则称为单电路 LLDC。

A.2 单电路 aLLRC 级联

现在，我们给出一个关于（单电路）局部列表可恢复码的近乎最优级联定理。回顾一下，两个码 $\mathcal{C}_1$ 和 $\mathcal{C}_2$ 的级联 通过先应用 $\mathcal{C}_1$，再用 $\mathcal{C}_2$对每个符号进行编码，来编码消息 x 。众所周知，LLRC 可以进行级联，但查询复杂度会损失多项式因子 [36]。下面，我们将说明如何避免这种损失。

为了陈述结果，我们需要使用细粒度查询复杂度，这在注 3.6 中讨论过。回顾一下，对于直积码，它是解码器在单个 $(\Sigma_1)^r$ 编码符号中查询的 Σ_1 符号总数（回想一下，我们的解码器经常对子符号进行子抽样，并且只读取其中少数几个比特）。我们将用大写的 Q 表示细粒度查询复杂度，而不是小写的 q ，以便与前文区分。

命题 A.4 (单电路 LLRC 级联)。 固定常数 $c \in [0,1)$、错误参数 $\varepsilon > 0$、输入列表大小 $\ell'_{in} \in \mathbb{N}$，并且对于每个 $t \in [1+\log \frac{1}{\varepsilon}]$，固定细粒度查询复杂度 $Q^{(t)}, Q'^{(t)}$和输出列表大小 $\ell'^{(t)}_{in}, \ell^{(t)}_{out} \in \mathbb{N}$。设 $\mathcal{C}_1$ 和 $\mathcal{C}_2$ 是码，使得对于每个 t:

1. 外码 码： $\mathcal{C}_1$： $\Sigma_1^k \to \Sigma_2^{n_1}$是一个 $(\delta_{in}, \delta_{out})$-近似，$(\ell'^{(t)}_{out} \log \frac{1}{\varepsilon})$-输入，$(2^{-t} \log \frac{1}{\bar{\varepsilon}} \ell^{(t)}_{out}, Q^{(t)})$-*LLRC*

2. 内码： $\mathcal{C}_2$： $\{0,1\}^{\log|\Sigma_2|} \to \Sigma_3^{n_2}$是一个$(\delta'_{in}, \delta_{in}/2)$-近似，$(\ell'_{in})$-输入$(c + 2^{t-2}\varepsilon, \ell'^{(t)}_{out}, Q'^{(t)})$- *LLRC*

那么 $\mathcal{C}_1 \circ \mathcal{C}_2$： $\Sigma_1^k \to \Sigma_3^{[n_1]\times[n_2]}$ 是一个$(\delta'_{in}, \delta_{out})$-近似，$\ell'_{in}$-输入，$(c+\varepsilon, q, \ell_{out})$-*LLRC*，且

1. 查询复杂度： $q \le O\left(\log(Q^{(T)}) \sum_{t \in [\log \frac{1}{\varepsilon}+1]} Q'^{(t)} Q^{(t)}\right)$ *2.*

列表大小： $\ell_{out} \le \sum_{t \in [\log \frac{1}{\varepsilon}+1]} \ell^{(t)}_{out}$

其中 $T = 1 + \log \frac{1}{\varepsilon}$。[59]

证明。 固定一个输入字 $\mathcal{L} \in ((\Sigma_2)^{\ell'_{in}})^{[n_1]\times[n_2]}$ 以及级联码的一个函数 $f \in \mathcal{B}_{1-\varepsilon,\delta'_{in}}(\mathcal{L})$。对于每个 $i \in [n_1]$，令 $\mu(i)$ 表示满足条件 $\mathcal{L}_{i,j}$ δ'_{in}-计算 f 的 $j \in [n_2]$ 所占比例。在 [36], 中，作者通过马尔可夫不等式观察到，必定有 $\Omega(\varepsilon)$ 比例的指标 $i \in [n_1]$ 满足 $\mu(i) \ge c + \Omega(\varepsilon)$。我们证明了这一观察的一个强化版本，从而可以改进该界：忽略对数因子，存在某个 t，使得至少有 2^{-t} 比例的 i 满足 $\mu(i) \ge c + 2^t\varepsilon$。因此，内码、外码（或两者）都可以在假设输入 $\varepsilon^{(t)}$ 大得多的情况下运行。

命题 A.5。 存在 $t \in [1+\log \frac{1}{\varepsilon}]$，使得至少有 $\frac{2^{-t+1}}{3}(\log \frac{1}{\varepsilon})$比例的 $i \in [n_1]$满足 $\mu(i) \ge c + 2^{t-2}\varepsilon$。

我们将推迟证明该断言，因为它只是一个简单的分桶马尔可夫论证。现在请注意，虽然在不知道 f 的情况下我们无法显式找到这个 t，但由于必然存在一个良好的 t，我们将简单地对每个 t分别运行内层和外层解码器。我们的最终输出电路将把这些电路并行连接起来（更准确地说，是连接这些电路的成功放大版本），这意味着我们需要付出查询代价和列表大小之和。

形式化地，令 $T = \lceil \log \frac{1}{\varepsilon} + 1 \rceil$，并固定满足上述保证的 t'；如果至少有 $c + \frac{2^t\varepsilon}{4}$-比例的符号 $w_{i,\,j}$ δ'_{in}-计算 $\mathcal{C}_1 \circ \mathcal{C}_2(f)$, 则称索引 $i \in [n_1]$ 为“良好”索引。根据假设，在良好索引上，我们的内码 $\mathcal{C}_2$ （在级别 t'运行）输出一个单电路 $\mathcal{C}_i$，其输出索引中有一个（例如第 ℓ个） $\ell'^{(t)}_{out}$ $\delta_{in}/2$-计算f。通过对每个 $\mathcal{C}_i$ O进行 $Q^{(T)}$ 次多数表决，我们可以确保第 ℓ个索引确定性地δ_{in}-计算 f；对于 r 的随机固定选择，该事件以至少 $1 - O(\frac{1}{Q^{(T)}})$ 的概率发生。

现在，我们对每个 t 运行外码，并使用这些内电路。选取一个随机字符串 r 供内部电路使用。对于“良好”的 t，我们的外码至多查询 $Q^{(t)} \le Q^{(T)}$ 个内电路，因此根据并集界，每个（经过放大后的）“良好”内电路 $\mathcal{C}_i(\cdot, r)$都具有一个输出索引 $O(\delta_{in})$、可计算 f的概率至少为（例如）99%。

[59]为简便起见，我们假设 $1 + \log \frac{1}{\varepsilon}$为整数，且 $Q^{(t)}$ 随 t 递增，这通常都是如此。否则，可以将下方的 $\log Q^{(1+\log \frac{1}{\varepsilon})}$ 替换为 $\max_t \log Q^{(t)}$。

以该事件为条件，我们恰好处于近似列表恢复的情形。换言之，我们的外码 $\mathcal{C}_1$ 在每个索引上都具有对内层码字列表的（模拟）查询访问（由每个 $\mathcal{C}_i$的输出索引给出），其中有 $c+2^{t-2}\varepsilon$的比例包含一个可对 f 进行 δ_{in}-计算的函数。因此，我们的外码对内 $\mathcal{C}_i$ 进行总计$Q^{(t)}$ 次查询，并生成一个单电路 $\ell_{out}^{(t)}$维输出，其中某个索引δ_{out}以至少 $1/2$的概率对 f 进行计算。将这一过程独立重复 $O(1)$ 次（并行连接输出电路），即可达到所需的 $3/4$ 成功率。查询复杂度和列表大小可直接由构造得出。

尚待证明命题 A.5。与先前的工作一样，关键观察很简单：根据假设，我们有全局期望 $\mathbb{E}_{i\in[n_1]}[\mu(i)] = c+\varepsilon$，因为 μ 是在随机 $j\in[n_2]$上取平均。反设对于每个 t，满足 $\mu(i)\in[c+2^{t-3}\varepsilon, c+2^{t-2}\varepsilon]$ 的 $i\in[n_1]$ 所占的比例都小于 $\frac{2^{-t+1}}{3}$ 对数 $\frac{1}{\varepsilon}$。于是：

$$\begin{aligned}\mathbb{E}[\mu(i)] &= \sum_{a\in[0,1]} \Pr[\mu(i)=a]a \\ &\le c+\frac{\varepsilon}{2} + \sum_{t\in[1+\log\frac{1}{\varepsilon}]} \Pr\left[\mu(i)\in[c+2^{t-2}\varepsilon, c+2^{t-1}\varepsilon]\right]2^{t-1}\varepsilon \\ &\le c+\frac{\varepsilon}{2} + \sum_{t\in[1+\log\frac{1}{\varepsilon}]} \frac{2^{-t+1}}{3(1+\log\frac{1}{\varepsilon})}2^{t-1}\varepsilon \\ &< c+\varepsilon\end{aligned}$$

从而得到所需的矛盾。特别地，必然存在某个 t，其中满足 $\mu(i)\ge c+2^{t-2}\varepsilon$ 的 $i\in[n_1]$所占比例为 $> \frac{2^{-t+2}}{3(1+}$（对数 $\frac{1}{\varepsilon}$），如所需。 □

我们指出，假设 $\varepsilon \ge \text{poly}(\frac{1}{n_1})$ 足够大，则最终查询复杂度中的对数 $Q^{(t)}$ 项可以去除。

A.3 多项式规模、低错误率 aLLDC

In 在本小节中，我们构造局部可计算 aLLDC 的完整内码：

命题 A.6（内码）。对于所有足够大的 $N\in\mathbb{N}$、$\delta_{out}\ge\Omega(\frac{1}{\text{poly}(N)})$ 以及 $\varepsilon>0$，存在一个二元码 $\mathcal{C}$：$\{0,1\}^N\to\{0,1\}^{N'}$，它是 δ_{out}-近似的 $(\varepsilon,\ell_{out},q)$-单电路 *LLDC*，具有

1. 码率：$R\ge\text{poly}(N^{-1})\cdot 2^{-O(\log^2\frac{1}{\varepsilon})}$ *2.* 查询复杂度：$q\le$多对数$(\frac{1}{\delta_{out}},\frac{1}{\varepsilon})\varepsilon^2$ *3.* 列表大小：$\ell_{out}\le$多对数$(\frac{1}{\varepsilon})\varepsilon^2$ *4.* 局部性：$\text{poly}(\delta_{out}^{-1})\log(\frac{1}{\varepsilon})$

此外 $\mathcal{C}$可在 *TIME*(多对数$(N)\log^2(1/\varepsilon)$) 内局部可编码，并可由对数空间一致电路解码，其规模为 $\text{poly}(\log(N)\frac{1}{\varepsilon})$，深度为 $O(\log^2(\log_{\delta_{out}}N)+\log\frac{1}{\varepsilon})$

命题 A.6 是两个码的复合。一个局部性高效二元近似局部可解码码：

命题 *A.7*（唯一译码范围内的放大）。对于所有足够大的 $N\in\mathbb{N}$，$\delta_{out}\ge\Omega(\frac{1}{\text{poly}(N)})$，以及每个 $N'\ge \text{poly}(\frac{1}{\delta_{out}})N$，都存在一个二元码 $\mathcal{C}:\{0,1\}^N\to\{0,1\}^{N'}$，它是一个 δ_{out}-近似、$(0.9,q)$-查询局部可解码码，适用于 $q\le\text{polylog}(\log_{\delta_{out}}N)$。此外，$\mathcal{C}$是局部可编码的，所需时间为 $\text{Time}(\text{poly}(\log(N)\frac{1}{\delta_{out}}))$，并且可以由规模为 $\text{poly}(\log(N)\frac{1}{\delta_{out}})$、深度为 $O(\log^2(\log_{\delta_{out}}N))$ 的对数空间一致电路解码。

以及处于列表译码区间的二元直积码，来自 [36, 106]:

命题 A.8 (多项式码率二元 aLLDC). 设 p 是素数幂，$\varepsilon \geq \Omega(1/p)$，且对于任意 $m \geq 10$，$N = \mathbb{F}_p^m$。存在一个二元码 $\mathcal{C}$：$\{0,1\}^N \to \{0,1\}^{N'}$，它是一个 0.001-近似的、$(\varepsilon, \ell_{out}, q)$-单电路局部列表可解码码，且

1. 码率：$R \geq$ 多项式$(N^{-1}) \cdot 2^{-O(\log^2 \frac{1}{\varepsilon})}$ *2.* 查询复杂度：$q \leq \tilde{O}\left(\frac{1}{\varepsilon^2}\right)$ *3.* 列表大小：$\ell_{out} \leq \tilde{O}(\frac{1}{\varepsilon^2})$ *4.* 局部性：$O($对数$\frac{1}{\varepsilon})$

此外，$\mathcal{C}$ 可在时间(多对数$(N) \cdot$ 对数$^2(1/\varepsilon))$ 内局部可编码，并且可由规模为多项式(对数$(N)\ \frac{1}{\varepsilon}$)、深度为 O(对数$(\frac{1}{\varepsilon}))$的对数空间一致电路解码。

我们将在下面的不同小节中分别介绍这些内容。命题 A.6 可由二者直接推出。

命题 *A.6* 的证明。所述码通过组合命题 A.8（$\mathcal{C}_2$）和命题 A.7（$\mathcal{C}_1$）的适当参数设置得到。具体而言，令 p 为大于 $\Omega(\varepsilon^{-3/4})$ 的最小素数幂。我们可以将命题 A.7 实例化为适用于 $m \geq 10$、$N' \leq N \cdot$和 $N, \frac{1}{\delta_{out}}, \frac{1}{\varepsilon}$的 $[N' = p^m, N]$码。这为与命题 A.8（经常数次成功放大）组合提供了合适的参数；命题 A.8 要求消息长度为这样的素数幂（即取编码 $\mathcal{C}_2(\mathcal{C}_1(f))$）。

现在，通过成功放大（见定义 2.2 之后的讨论），给定复合码的一个码字 w，我们可以假设 $\mathcal{C}_2$的列表译码器（在随机性的选择上）以98%的概率产生一个确定性电路，其某个输出对 $\mathcal{C}_1(f)$进行0.1-计算。在该事件条件下，我们可以将此次解码视为我们针对 $\mathcal{C}_1$的唯一译码器的输入。该过程（再次经过适当放大）以99%的概率产生一个与 f 相距 $1 - \delta_{out}$ 的函数，正如所需（对成功概率取并集界后显然大于所需的$_{3\ 4}$）。

查询复杂度、码率、编码时间与电路复杂度均可由构造直接得到。具体而言，在复合中，查询次数与码率是组成码的乘积，电路规模至多按各电路规模与查询复杂度之积增长，深度可加，编码局部性为乘积，且编码时间至多为内码时间加上外码时间与局部性的乘积。

□

A.3.1 列表译码体制（命题 A.8）

在本小节中，我们证明命题 A.8。如上所述，此码由三个针对单电路列表恢复设置改造的标准 aLLDC 拼接而成：[36] 的子空间码、完全复形和哈达玛码。事实上，此码已隐含于 [36]，并在 [106] 中明确使用——唯一的新意是通过命题 A.4 改进了查询复杂度和列表大小的分析。

我们首先陈述 [36] 的子空间码在单电路近似局部列表可恢复码模型中的保证：

定理 A.9（子空间码 [36]）。对于任意 $\varepsilon > 0$，令 $p \geq \Omega(\frac{1}{\varepsilon^{3/4}})$ 为一个素数幂，$m \geq 10$、$V = \mathbb{F}_p^m$，且 S 为 V 的 8 维子空间集合。则直积码 $C_{(V,S)}$ 是一个（0.0005，0.001）-近似的、具有 ℓ_{in} 个输入的（ε、ℓ_{out}、q）单电路 *LLRC*，其

1. 码率：$R \geq \frac{1}{\text{多项式}(|V|)}$ *2.* 查询复杂度：$q \leq O(\frac{1}{\varepsilon^2})$ *3.* 列表大小：$\ell_{out} \leq O(\frac{\ell_{in}}{\varepsilon})$ *4.* 局部性：$r \leq O(p^8)$

此外，$C_{(V,S)}$ 可在时间$(O(\log \frac{N}{\varepsilon}))$ 内进行局部编码，并可在对数空间一致规模为 $O(\text{poly}(\log(N)\ \frac{1}{\varepsilon}))$、深度为 $O(\log(\frac{1}{\varepsilon}))$的电路中解码。

上述结果与 [36] 所述结果的唯一区别在于，必须从输入列表中的每个函数开始运行其译码算法。[60] 查询可以在列表元素之间重复使用，但朴素地说，无法在他们用作辅助信息的 $O(1/\varepsilon)$ 个不同起始点之间重复使用，因此复杂度为 $O(1/\varepsilon^2)$。值得指出的是，通过进一步修改他们的算法，实际上可以将其改进为 $q \leq O(\frac{1}{\varepsilon})$，但这并不会改善级联码最终的查询复杂度，因此我们省略这一点。

该子空间码仍具有相当大的对数字母表（规模为 $\mathrm{poly}(\varepsilon^{-1})$），但关键在于它与顶点数量无关 $N = q^m$。因此，我们可以将其与完全复形码串接，而不会导致 N 大幅膨胀：

定理 A.10（完全复形码 [36]）。存在 $C > 0$，使得对于任意 ε、$\delta > 0$、$\ell_{in} \in \mathbb{N}$、$d \geq C \ln \frac{\ell_{in}}{\varepsilon}$，以及 $N \geq d$，完全复形 $(\Gamma, \binom{[N]}{d})$ 上的直积码 $\mathcal{C}$ 是一个 0.0005-近似、ℓ_{in}-输入的 $(\varepsilon, \ell_{out}, q)$-单电路近似局部列表可恢复码，其

1. 查询复杂度：$q \leq \tilde{O}(\frac{1}{\varepsilon^2})$
2. 列表大小：$\ell_{out} \leq \tilde{O}(\frac{\ell_{in}}{\varepsilon})$
3. 局部性：$O(\log \frac{\ell_{in}}{\varepsilon})$

此外，$\mathcal{C}$ 可在时间 $(O(\log N \log \frac{1}{\varepsilon}))$ 内进行局部编码，并可在对数空间一致电路规模 $\mathrm{poly}(\log(N) \frac{1}{\varepsilon})$ 和深度 $O(\log \frac{1}{\varepsilon})$ 下进行局部解码。

上述内容需要进行与子空间码部分所讨论的完全相同的修改。

最后，我们需要从完全复形所提供的小字母表设定转到二元设定。经典做法是使用哈达玛码，我们回顾一下，它将一个字符串 $x \in \{0,1\}^k$ 映射为 $Had(x)_\gamma := \langle x, \gamma \rangle$。Goldreich和Levin [17] 提出了一个低复杂度的局部列表译码算法，可将哈达玛码的译码距离任意逼近 $1/2$。事实上，Goldreich和Levin 的分析已经给出了如下形式的单电路 LLDC（即查询位置不依赖于辅助信息，因此可以重复使用）：[61]

定理 A.11 (哈达玛码 [17])。对于每个 $N \in \mathbb{N}$ 和 $\varepsilon > 0$，哈达玛码 $\mathcal{C}_{Had}: \{0,1\}^N \to \{0,1\}^{2^N}$ 是一个具有以下性质的 $(\frac{1}{2} + \varepsilon, \ell_{out}, q)$-单电路 *LLDC*

1. 查询复杂度：$q \leq O(\frac{1}{\varepsilon^2})$
2. 列表大小：$\ell \leq O(\frac{1}{\varepsilon^2})$

此外，$\mathcal{C}_{Had}$ 可以在时间 $(O(N))$ 内进行局部编码，并可由规模为 $\mathrm{poly}(N, \frac{1}{\varepsilon})$、深度为 $O(\log \frac{1}{\varepsilon})$ 的对数空间一致电路进行局部解码。

哈达玛码的主要缺点当然是其指数级码长，但这并不是问题，因为与完全复形级联后，我们将把它应用于大小为 $N = O(\log \frac{1}{\varepsilon})$ 的块。将这三种码级联起来，即可得到命题 A.8 所需的多项式码率二元近似局部列表可恢复码：

命题 *A.8* 的证明。结合命题 A.4、定理 A.9、定理 A.10 和定理 A.11，即可立即得到证明。具体而言，我们采用从子空间码开始，接着是完全复形，最后是哈达玛码的三重串接。更明确地说，根据命题 A.4，将定义在 N 个顶点上的子空间码与维数为 $O(\log \frac{1}{\varepsilon})$ 的完全复形码级联，得到一个 0.001-近似、ℓ_{in}-输入的 $(\varepsilon, \ell_{out}, q)$-LLRC，其

$$q \leq \tilde{O}(\frac{1}{\varepsilon^2}) \qquad \text{and} \qquad \ell_{out} \leq \tilde{O}(\frac{\ell_{in}}{\varepsilon})$$

再次应用命题 A.4，将其与哈达玛码串接，可得到一个 0.001-近似的 $(\varepsilon, \ell_{out}, q)$-局部列表可解码码，其

$$q \leq \tilde{O}(\frac{1}{\varepsilon^2}) \qquad \text{and} \qquad \ell_{out} \leq \tilde{O}(\frac{1}{\varepsilon^2})$$

[60] 严格来说，这还要求设置采样参数，以便我们能对列表应用并集界，但这只是一个相当低阶的因子。[61] 关于该结果的阐述，请读者参阅 [107]；下文所声称的版本在其中尤为清晰。

局部性、编码和译码复杂度可直接从构造（即码级联）得出。局部性由完全复形步骤继承。编码时间的界可通过追溯各组成码的编码时间和局部性得出，同时注意到我们的子空间复形、完全复形和哈达玛码的消息长度分别为 $N_{subspace} = \text{poly}(\log\ N, \frac{1}{\varepsilon})$、 $N_{complete} = \text{poly}(1/\varepsilon)$和 $N_{Had} = O(\log\ \frac{1}{\varepsilon})$。由此得到的电路复杂度也可用类似方式得出。 □

A.3.2 唯一译码区间（命题 A.7）

在本小节中，我们证明命题 A.7：一种兼具局部性和查询效率的唯一解码过程，可将一个以 f 99% percent 的概率计算正确的电路放大为一个除误差为 δ_{out} 外以 f 计算正确的电路。这是较简单的「唯一译码区间」，在困难性放大文献（例如 [35]）和编码理论 [108] 中都已得到充分研究。我们将采用 Guruswami 和 Kabanets [35] 提出的一种简单构造的变体；该构造通过扩展图游走上的直积对 f 进行编码，然后用纠错码进行「扁平化」。

为避免局部性和查询复杂度额外损失 poly log(N) 因子，我们不使用扩展图游走，而是在第 6 节的 KMS 扩展图的邻域上取直积，并使用局部可解码码进行「扁平化」。为处理任意码长，我们按照引理 E.2 重复输出，这与我们先前构造任意消息长度的码时采用的方法相同。

在给出的扩展图

证明之前，我们先正式陈述这两个主要组成部分。首先回顾第 6 节中满足所需参数范围的对数空间一致构造第 6 节中处于所需参数范围的扩展图：

推论 A.12 (对数空间采样器)。 存在 $C > 0$，使得以下结论成立。任取 $N \in \mathbb{N}, \beta \in (0,\ 1)$。存在 $N' \in [N,\ \ ^{C_{\text{poly}(\beta)}}N]$，以及一个强显式的、$\text{poly}(\beta^{-1})$-正则 β-采样器图 $G_{KMS} = ([N'], [N'], E)$。此外，$G_{KMS}$ 具有规模为 $\text{poly}(\log(N)\ \beta^{-1})$、深度为 $O(\log\log(N) \cdot \log\frac{1}{\beta})$ 的对数空间一致电路实现。

具体而言，上述结论可由对命题 4.3 中超图系统底层的高维扩展体内 $(\mathfrak{X}[1], \mathfrak{X}[2])$ 上的着色交换游走进行抽样得到，其中 $\mathfrak{q} = \text{poly}(\beta)$，即二维 KMS 复形。

我们的第二个组成部分是一种标准构造：将 Reed-Muller 与任意良好的纠错码级联（例如参见 [109, 110]），即可得到多对数查询二元局部可解码码。

定理 A.13 (多对数查询局部可解码码)。 对于每个足够大的 $N \in \mathbb{N}$，以及满足 $2^{O(\log^{2/3}}$ 的倍数条件的 $N' \geq N^{2+o(1)_{(N))}}$，存在一个显式二元码 $\mathcal{C}_N$：$\{0,1\}^N \to \{0,1\}^{N'}$，它是 $(0.99, O(\log^2(N)))$ 局部可解码码，且可在 *TIME*$(\text{poly}(N))$ 内局部编码，并可在规模为 $O(\text{poly}(N))$、深度为 $O(\log^2(N))$ 的对数空间一致电路中局部解码。

现在我们简要概述构造本身。正如前面所述，我们的编码由两个主要部分组成，并在两部分外侧加入重复码，以确保每部分的码长和消息长度相匹配。具体而言，从 $x \in \{0,1\}^k$ 中的消息开始，我们首先重复编码 x，使其长度与采样器图 G_{KMS} 的规模相匹配；然后将每个符号与所需的局部可解码码级联，并重复编码结果，以达到最终所需的码长。

命题 *A.7* 的证明。只需证明，将首个 G_{KMS}-直积编码与 $\beta = O(\delta_{out})$（因此度为 $\text{poly}(1/\delta_{out})$）、顶点规模为 $N' \in [N, \text{poly}(\delta_{out}^{-1})N]$，以及定理 A.13 中的局部可译码相级联后，得到的是一个 δ_{out}-近似 $(0.99, q)$-查询局部可解码码。由引理 E.2 可知，完整码（带有初始和最终重复编码）就是所需的近似局部可解码码。将前述码记为 $\mathcal{C}$。

固定一个码字 w 和目标函数 $f \in \mathcal{B}_{0.01}(w)$。我们逐层解码 f。根据马尔可夫不等式，LDC 的“块”（即 LDC 对首层直积码中符号的编码）中，坐标有超过 20% 被翻转的比例至多为 $1/10$。因此，在 90% 的块上（也就是该层中 90% 的消息符号上），LDC 解码以 $2/3$ 的概率（对 LDC 的随机性取概率）正确。

现在，在上述解码器成功的条件下，根据采样可直接得出，除占比为 a$O(\delta_{out})$ 的消息位外，其余消息位都至少有 80% 的“良好”块（即错误少于 20% 的块）。查询这样的块，得到正确比特的概率至少为 $2/3$，因此根据切尔诺夫界，对包含该比特的 $O(\log(\frac{1}{\delta_{out}}))$ 个随机编码进行多数解码，除概率为 $1 - O(\delta_{out})$ 的情况外，都能解出正确比特。

总而言之，对失败事件以及占比为 $O(\delta_{out})$、未能看到足够多良好 LDC 块的“坏”顶点使用并集界后，对于随机消息位以及算法的内部随机性，我们的解码成功概率至少为 $1 - O(\delta_{out})$，符合预期。接下来需要分析该算法的各项参数，我们将在下文分别进行分析：

码长 根据引理 E.2，将初始符号从 N 重复到下一个最小的顶点规模 G_{KMS}，码长至多增加 $\mathrm{poly}(\delta_{out})$ 倍。直积编码使对数字母表增加 $\mathrm{poly}(\delta_{out})$，随后使用局部可解码码对每个符号进行编码，使每个块的长度加倍，从而再增加 $\mathrm{poly}(\delta_{out})$ 倍。最后，通过重复编码将码长调整为任意所需的 $N' \geq N\,\mathrm{poly}(\delta_{out}^{-1})$。

查询复杂度 基础解码器进行 $O(\log(\frac{1}{\delta_{out}}))$ 次查询，每次查询都调用内部局部可解码码，而后者进行 $O(\log^2 \frac{1}{\delta_{out}})$ 次查询。重复码每个进行 $O(1)$ 次查询，因此总查询复杂度为 $O(\log^3 \frac{1}{\delta_{out}})$。

编码时间 每个编码位仅依赖于 G_{KMS} 直积码的一个“块”（其度为 $\mathrm{poly}(\delta_{out}^{-1})$），然后再由局部可解码码进行编码。局部可解码码的编码耗时为 $\mathrm{poly}(\delta_{out})$。在局部可解码码层中计算每个消息位，实质上只需计算该块的邻居上的原始消息；通过我们对 KMS 的实现，这可以在 $\mathrm{poly}(\log(N)\ \delta_{out}^{-1})$ 时间内完成。重复编码仅需 $O(\log(N))$ 时间。注意，码的局部性由编码时间给出上界（当然，实际上它更小，恰好是 G_{KMS} 的度，但我们不会用到这一点）。

译码复杂度 我们的解码过程需要：1）在 G_{KMS} 中采样 $O(\log \frac{1}{\delta})$ 个随机邻居并进行多数表决；2）通过运行相应的 LDC 对这些邻居进行解码；3）对重复码进行解码。上述每个过程都可以按需在对数空间一致、规模为 $\mathrm{poly}(\log(N)\ \delta_{out}^{-1})$、深度为 $O(\log^2(\frac{\log(N)}{\delta_{out}}))$ 的条件下完成。

□

A.4 将一切整合起来

我们基本上已经准备好证明本节的主要结果。所欠缺的只是以下这种标准的列表可译码码构造，它用于我们的大局部性结果。该构造本身很简单：将 Reed-Solomon 码与任意一种低于 Johnson 列表译码半径的良好二元内码进行级联，再填充至所需的任意消息长度：

定理 A.14 (二元列表可解码码（例如参见 [3, 92, 111]))。对于足够大的 $N \in \mathbb{N}$ 和 $\varepsilon > \frac{1}{\mathrm{poly}(N)}$，存在一个二元线性码 $\mathcal{C}\colon \{0,1\}^N \to \{0,1\}^{N'}$，其码率为 $\mathrm{poly}(\varepsilon)$，且该码是 $(\varepsilon, O(\frac{1}{\varepsilon^2}))$-列表可译码的。此外，该码可在 $\mathit{TIME}(\mathrm{poly}(N, \frac{1}{\varepsilon}))$ 内编码，并可在对数空间一致、规模为 $O(\mathrm{poly}(N, \frac{1}{\varepsilon}))$、深度为 $\tilde{O}(\log^2 N)$ 的条件下解码。[62]

现在我们可以证明本节的两个主要结论了。

推论 *A.1* 和推论 *A.2* 的证明。我们先证明前者。回顾定理 4.1（当 $\delta \leq O(\frac{1}{\text{对数}^5(N)})$ 足够小时）是一个 $(O(\frac{1}{\text{对数}^5(N)}), 0.01)$-近似的、$\ell_{in}$-输入的 $(\varepsilon, O(\frac{\ell_{in}}{\varepsilon}) \tilde{O}(\text{对数}^5(N)\varepsilon^2))$-LLRC，适用于任意 $\varepsilon > 0$，其对数字母表大小为 $\mathrm{poly}(\frac{\log(N)}{\varepsilon})$（注意，只需将 q 多项式增大，我们也可以令这个多项式任意大）。因此，我们可以与定理 A.14 进行级联，而命题 A.4 表明最终的码具有所需的列表译码参数。码率、编码和译码复杂度可由级联直接得到（注意，内码定理 A.14 中的消息长度‘N’设为对数字母表大小 $\mathrm{poly}(\frac{\log(N)}{\varepsilon})$）。

后一个结果的证明类似。唯一的变化是，我们在参数设置相同的情况下，与命题 A.6 进行级联。列表译码参数也可类似地得到。译码电路深度是相加的，而原外码的深度 $O(\log(\log_{\varepsilon} N) \log\log N)$ 占主导。电路规模至多为所有电路规模与查询次数的乘积，该值仍为 $\mathrm{poly}(\log(N)\ \frac{1}{\varepsilon})$。

[62] 这里 $\tilde{O}$ 隐藏了 $\mathrm{loglog}(N)$ 因子。我们指出，深度界是广为流传的结果。它并未在 [3] 中明确陈述，但译码算法仅使用域大小为 $O(N)$ 上的求根与线性方程组求解，而这些处于对数空间一致的 $\tilde{O}(\log^2(N))$-深度。

最后，编码时间至多为内码的时间，加上内码的局部性乘以外码的编码时间；后者至多为 $\text{polylog}(N) \cdot \log^2 \frac{1}{\varepsilon}$，正如所需。 □

最后，我们指出，推论 5.2 可由完全相同的论证推出，因为与定理 A.14 级联只会使码率损失常数因子。

B 困难性放大

本节将介绍我们的结果对困难性放大的若干推论，即如何从偶尔困难问题构造可靠困难问题。困难性放大对于定量理解复杂度很重要，同时也对密码学和伪随机性至关重要，因为可靠的困难性在这些领域中很有用。

我们提出的码的不同版本，分别对应于在不同设置下、用于不同目的的困难性放大。其联系在于将略难函数视为待发送的消息，而可靠困难函数则是对该消息的编码。给定这个码的一个噪声很大的版本（即一种仅比随机猜测略好一些的问题求解算法），我们可以解码出消息的一个近似版本，也就是说，得到一种能够解决原问题几乎所有实例的算法。由于这与原问题的假定困难性相矛盾，所构造的函数甚至不可能被以略优于猜测的准确率求解。

通常，我们希望考察复杂性类内部的困难性，因此希望可靠困难问题的复杂度与有时困难的问题基本相同。这对应于具有非常高效的局部编码算法的码。在其他情形下，我们假设有些困难问题的最坏情况难度是指数级的，并希望可靠困难问题仍然是指数级的，且指数几乎相同。这对应于其译码算法的复杂度次线性于消息长度、但仍是消息长度的某个幂次的码。如果我们希望保留问题精确的定量难度（作为输入规模的函数），那么构造出的函数的输入规模必须与原问题的输入规模非常接近。这对应于具有（相对）高码率的码。对于非一致模型，精确的列表大小通常不那么重要；但对于一致模型，这一点至关重要。我们的码在码率、局部性、列表大小和译码复杂度之间具有不同的权衡，因此会带来不同的困难性放大结果。下文中，我们只介绍这些码的一些推论，并着重于简洁而非最优性。

B.1 复杂性类内的困难性放大

我们将首先考虑一种困难性放大，它保持一个有时困难的问题的复杂度。这将为我们提供在相对较小的复杂性类（例如$NP \cap co-NP$或 P^{NP}）内可行的细粒度时间/错误权衡，这些复杂性类在多项式时间图灵归约下封闭。我们将给出一致模型与非一致模型的版本，但二者都将直接由相同的编码与译码算法得出。

定理 B.1 （一致的复杂性保持型硬度放大）。 设 $T(n)$ 为时间复杂度，$1 > \varepsilon(n) \in 2^{-o(\sqrt{n})}$，与 A 为一个非负整数。设

$$f : \{0,1\}^n \to \{0,1\},$$

使得

$$f \notin Heur_{\frac{1}{n^A}} - BPTIME(T(n))/O(\log(n/\varepsilon))).$$

则存在长度 $n' = n(1 + o(1))$ 与一个 F: $\{0,1\}^{n'} \to \{0,1\}$ 使得 $F \in P^f$ 与 $F \notin Heur_{\frac{1}{2}-\varepsilon(n)} - BPTIME($ $T(n)$ poly$(\frac{\varepsilon}{(n)n}))$

定理 B.2 (非一致保持复杂度的困难性放大)。 设 $S(n)$ 为一个电路规模函数， $1 > \varepsilon(n) \in 2^{-o(\sqrt{n})}$, A 为非负整数。设

$$f : \{0,1\}^n \to \{0,1\}$$

使得

$$f \notin Heur_{\frac{1}{n^A}} - Size(S(n), n).$$

那么存在一个长度为 $n' = n(1 + o(1))$ 以及一个 F: $\{0,1\}^{n'} \to \{0,1\}$ 使得 $F \in P^f$ 且 $F \notin Heur_{\frac{1}{2}-\varepsilon(n)} - Size(S(n)$ 多项式$(\frac{\varepsilon(n)}{n}))$

我们通过一个相同的构造来证明这两个定理。首先，令 $N = 2^n$ 并将 f 视为一条 N 比特消息，其中第 x 位为 $f(x)$。我们使用来自命题 A.7 的码 C_1 （使用 $\delta_{out} = n^{-A}$ 和 $N = 2^n$）构造一个新的函数 f_1，其输入为 $n_1 = n + O(\log\ n)$ 比特，其中 $f_1(y)$ 是 $C_1(f)$ 的第 y 个输出比特。C_1 具有以下性质：

1. $|C_1(f)| = N_1 = N/$ 多项式$(1/\delta_{out})$，因此 $n_1 = \log\ N_1 = n + O(\log\ n)$
2. $C_1(f)$ 的每一位都可以通过对 f 的查询，在 $\text{poly}(\log\ N, 1/\delta_{out}) = \text{poly}(n)$ 时间内计算得到。因此，$f_1 \in P^f$。
3. f 可以从任意与$C_1(f)$具有一致度 $.9N'$ 的受损消息中被 δ_{out}-近似唯一解码，且解码得到的消息表示为一个规模为

$$\text{poly}(\log N, 1/\delta_{out}) = \text{poly}(n)$$

通过对受损码字的查询。

那么如果

$$f_1 \in Heur_{.1} - Size(S_1(n), n_1),$$

存在一个规模为 $S_1(n)$的电路 $Circ_1$，它在 $.9N_1$ 个比特位置 y上与 $C(f)$一致。使用上述译码算法可得到一个电路 $Circ$，它以 $Circ_1$ 为子程序，其余部分规模为$\text{poly}(n)$，并且除 $\delta_{out}N$ 个输入 x外均与 f 一致。由于 $Circ_1$ 的规模为 $S_1(n)$， $Circ$ 的规模将为 $S_1(n)\ \text{poly}(n)$。因此，假设 $f \notin Heur_{\delta_{out}} - Size(S_1\ \text{poly}(n)\ n)$，可得 $f_1 \notin Heur_{.1} - Size(S_1(n)\ n_1)$。

然后，我们应用推论 A.2 中的列表可译码码 C_2，得到一个函数 F，其中 $F(z)$ 是 $z(f_1)$ 的第 z 个比特。该码具有以下性质：

1. 编码后消息的长度为

$$|C_2(f_1)| = N_2 = N_1\ \text{poly}(\log N_1) 2^{O(\log^2 1/\varepsilon)}.$$

因此， F 的输入规模为 $n_2 = \log\ N_2 = n_1 + O(\log n_1) + O(\log^2 1/\varepsilon) = n + O(\log\ n) + O(\log^2 1/\varepsilon)$。由于 $\varepsilon \in 2^{-o(n^{1/2})}$， $n_2 = n + o(n)$。

2. $C_2(f_1)$ 的每个输出比特都可以在 $\text{poly}\log(N_1) \cdot \log^2 1/\varepsilon = \text{poly}(n)$ 时间内通过查询 f_1 的比特来计算。因此，$F \in P^{f_1} \subseteq P^f$。

3. 对于任意与 F 在少于 $\frac{1}{2} - \varepsilon$ 比例的比特位置上不同的电路 $Circ_2$，我们可以在 $\text{poly}\log(N_2) \cdot \text{poly}(1/\varepsilon)$ 时间内计算出一个由 $\text{poly}(1/\varepsilon)$ 个电路组成的列表 $Circ_{1,\ i}$，其中一个电路与f_1 的一致度为 。9。

如果 $F \in Heur_{\frac{1}{2}-\varepsilon} - Size(S_2(n)\ n_2)$ 可通过电路 C_2 实现，那么解码器使用 $Circ_2$ 作为受损消息时输出的列表中，电路在所有输入中除 .1 的比例之外的输入上的最佳输出应为一个 $Circ_1 \in Size(S_2(n)\ \text{poly}(n/\varepsilon))$，计算 f_1。因此 $f_1 \in Heur_{.1} - Size(S_2(n)\ \text{poly}(n/\varepsilon)\ n_1)$。根据上文所述，这意味着 $f \in Heur_{\delta(n)} - Size(S_2(n)\ \text{poly}(n)\ \cdot \text{poly}(n/\varepsilon)\ n)$，对于某个 $S_2(n) < S(n)poly(\varepsilon/n)$ 而言产生矛盾。

对于定理 B.1，下面对 $Heur_{\delta(n)} - BPTIME(T(n))/a(n)$ 的部分刻画将会很方便：

引理 B.3。如果 $f \in Heur_{\delta(n)} - BPTIME(T(n))/a(n)$，则存在一个随机化算法，给定 n，在 $T(n+a(n)$ $poly(n)$时间内返回一个规模为 $S(n)$的电路 $Circ_{guess}$，使得以至少 $1/2^{-a(n)+1}$ 的概率， $Circ_{guess}$ 在除 $2\delta(n)$比例的输入之外的所有输入上都与 f 一致

证明。令 $A(n, r, advice)$ 为证明 $f \in Heur_{\delta(n)} - BPTIME(T(n))/a(n)$ 的随机化算法。该随机化算法选取一个随机的 r_0 和一个长度为 $a(n)$ 的随机字符串 $advice_0$，并返回一个模拟 $Circ_{guess}(x) = A(x, r_0, advice_0)$ 的电路。以概率 $1/2^{a(n)}$，$advice_0 = advice(r_0)$，且在此事件发生的条件下，根据马尔可夫不等式，$A(x, r_0, advice_0)$ 与 f 的差异超过预期输入数量两倍的概率至多为 $1/2$。 □

下面给出一个部分逆命题：

引理 B.4. 如果存在一个运行时间为 $T(n)$ 的随机化算法，能够生成电路 $Circ$，使得以概率$\alpha(n)$，$Circ(x)$与 $f(x)$在至多占输入总数 $\delta(n)$的输入上不同，那么 $f \in Heur_{2\delta(n)} - BPTIME(T(n) + \text{poly}(n))/O(\log 1/\alpha + \log 1/\delta)$。

证明。考虑一个两两独立的随机字符串序列 $r_1, r_2, ...r_K$，使得给定 i后，可以在多项式时间内计算出 r_i。（例如，令 $r_i = a * i + b$ 取自有限域中大小为 $2^{|r|}$的元素 a 和 b，并将每个 i 解释为不同的域元素。）根据切比雪夫不等式，没有 r_i 生成具有所需性质的电路 $Circ_i$ 的概率至多为 $\frac{1}{\alpha K}$。选取 $K = \frac{1}{\alpha\delta}$，可使该概率至多为 δ。若随机选取用于生成该序列的随机性，并将满足此条件的一个 i 用作辅助信息，则除以概率 δ发生的情况外，所生成电路的条件误差至多为 δ。以概率 δ，误差可能为 1，因此总体期望误差至多为 2δ。这说明$f \in Heur_{2\delta} - BPTIME(T(n)\,\text{poly}(n))/(\log 1/\alpha(n) + \log 1/\delta(n))$ □

回到定理 B.1，我们使用相同的构造，其中 $f_1 = C_1(f)$ 和 $F = C_2(f_1)$。由于这是相同的构造，$F \in P^f$ 和 $n_2 = n(1 + o(1))$ 也与之前一样。

假设 $F \in Heur_{\frac{1}{2}-2\varepsilon} - BPTIME(T(n))$。我们想要证明

$$f \in Heur_{2\delta(n)} - BPTIME(T(n)\,\text{poly}(n/\varepsilon)/O(\log 1/\varepsilon + \log n).$$

根据引理，只需给出一个在此时间内运行的算法，使其以 $\text{poly}(\varepsilon)$ 的概率生成一个电路 $Circ$，该电路在除 $\delta(n)$ 比例的输入之外，均能计算 f。我们只需对非一致情形中的几个步骤稍作修改。

首先，我们为随机启发式算法 $A(x, r)$ 选择一条随机带并将其固定。该启发式算法使用这条固定的随机带时，以至少 ε 的概率在除 $\frac{1}{2} - \varepsilon$ 比例的输入之外，均能计算 F。我们将此算法称为$-A_2$。然后，我们把 C_2 的解码器应用于作为受损消息的 A_2。这会得到一个由 $\text{poly}(1/\varepsilon)$ 个电路组成的族，其规模为 $T(n)\,\text{poly}(n/\varepsilon)$，其中一个电路以至多 。1 个差错计算 f_1。我们随机猜测应从该电路列表中保留哪一个作为 $Circ_1$。然后，我们将 $C_1(f)$ 的解码器应用于 $Circ_1$，以得到一个电路 $Circ$。如果$Circ_1$ 正确，那么 $Circ$ 在除 δ_{out} 比例的输入之外，均与 f 一致。

固定正确随机带的成功概率为 ε，乘以选中正确列表元素的概率 $\text{poly}(\varepsilon)$，总体成功概率为 $\text{poly}(\varepsilon)$。刻画带辅助信息的随机启发式类的引理由此完成证明。

B.2 保持计算近似时间的困难性放大

将困难问题用于伪随机生成器时，我们通常不关心确保新的可靠困难函数属于某个较小的复杂性类，只关心能否相对高效地计算其真值表。事实上，已知可以从最坏情况困难问题构造可靠困难问题（例如 [86, 13, 14]），这些构造隐式或显式地使用可精确解码的 LLDC。问题在于，这些码无法局部计算。这里，我们给出另一种权衡：基于平均情况难度的构造能够精确保留复杂度，而基于最坏情况难度的构造则无法保留复杂度，除非考虑 E 这类大型复杂性类。与前一小节中的结果相比，这种方法的优势在于，我们可以从非常弱的平均情况假设出发，将难度放大到 $1/2 - \varepsilon$，使其适用于 $\varepsilon = 2^{-\Omega(n)}$，同时保持输入规模较小；而先前的结果则要求$_{\varepsilon} = 2^{-o(n^{1/2})}$。

我们稍后将遵循[51]，利用这一结果给出可高效计算的伪随机生成器。由于这需要非一致下界，因此我们在此仅给出针对非一致复杂度的困难性放大。

定理 B.5. 存在一个 $\alpha > 0$，使得下述结论成立。设 $n \le S < 2^n$，设 $\frac{1}{n} > \varepsilon > 2^{-\alpha n}$ 和 $f \notin Heur_{\varepsilon} - Size(S, n)$。那么，存在布尔函数 $F \in TIME^f(\mathrm{poly}(n, 1/\varepsilon))$，其输入规模为 $n' = n + O(\log 1/\varepsilon)$，使得 $F \notin Heur_{1/2-\varepsilon} - Size(S\,\mathrm{poly}(\varepsilon)\ n')$

证明：该证明与前一小节中的证明类似，不同之处在于我们将命题 A.7 中的码 C_1 与 $\delta_{out} = \varepsilon$ 以及来自定理 1.1 和推论 4.2 的码 C_3 进行复合，分别得到 f_1 在 $n_1 = n + O(\log n)$ 比特输入上以及 F 在 n' 比特输入上。

码 C_1 的性质如下：

1. $|C_1(f)| = N_1 = N/$ 多项式$(1/\delta_{out})$，因此 $n_1 =$ 对数 $N_1 = n + O($对数 $1/\varepsilon)$

2. $C_1(f)$ 的每个比特都可通过对 f的查询在时间 poly(log N) $1/\delta_{out})$ $=$poly(n/ε) $=$poly$(1/\varepsilon)$ 内计算。因此，$f_1 \in TIME^f$(poly(n/ε))。

3. f 可被 $\delta_{out} = \varepsilon$ 唯一解码，其来源为任意具有一致度 $.9N_1$、且相对于 $C_1(f)$ 的受损消息，并且解码后的消息表示为一个规模为

$$\mathrm{poly}(\log N, 1/\delta_{out}) = \mathrm{poly}(1/\varepsilon)$$

通过查询访问受损码字。

因此可得 $f_1 \in TIME^f$(poly$(1/\varepsilon)$) 和 $f_1 \notin Heur_{.1} - Size(S \cdot$poly$(\varepsilon)\ n_1)$。

We 接着我们使用码 C_3从 f_1 构造 F。码 C_3 具有以下性质 :

1. 码率为 poly$(1/\log\ N_1, \varepsilon)$。因此，定义域大小为 $N_3 = N_1$ poly$(n, 1/\varepsilon)$ $= N_1$ poly$(1/\varepsilon)$，所以 $n_3 =$log $N_3 = n_1 + O(\log 1/\varepsilon)$ $= n + O(\log 1/\varepsilon)$。

2. 编码时间为 poly(log $N_1, 1/\varepsilon)$ $=$poly$(1/\varepsilon)$，使用对消息比特的查询访问。因此，$F \in TIME^{f_1}$(poly$(1/\varepsilon)$) $\subseteq TIME^f$(poly$(1/\varepsilon)$)

3. 列表译码电路规模为 poly(log $N_3, 1/\varepsilon)$ $=$poly$(1/\varepsilon)$。因此，给定一个规模为 S_3、计算 F且错误率为 $1/2 - \varepsilon$的电路，存在一个由此列表中的元素构成、总规模为 S_3poly$(1/\varepsilon)$ 的电路，该电路计算 f_1，错误率至多为.1

因此，假设通过某个电路 $Circ_3$ 满足 $F \in Heur_{1/2-\varepsilon} - Size(S_3,\ n_3)$，则译码列表中存在一个以 $Circ_3$ 作为受损消息的电路 $Circ_1$，它在除 。1 比例的输入之外的所有输入上计算 f_1，且规模为 $S_1 = S_3$poly$(1/\varepsilon)$。对 $Circ_1$ 运行命题 A.7 中的近似译码算法，将返回一个规模为 S_1poly$(1/\varepsilon)$ 的电路 $Circ$，在除 ε 比例的实例之外的所有实例上计算 f。除非 $S_3 < S$ poly(ε)，否则这将与关于 f 的既定困难性相矛盾。

应用该定理时，一组特别有趣的取值是：对于某个足够小的 $\alpha > 0$，$\varepsilon = 2^{-\alpha n}$ 且 $S = 2^{(1-\alpha)n}$。在这种情况下，平均情形假设等价于最坏情形假设：

推论 B.6。存在一个 $\alpha > 0$，使得对所有 $\alpha < \alpha_0$，以下结论成立。令 $f \notin Size(2^{(1-\alpha)n}, n)$。那么存在定义在规模为 $n' = n(1 + O(\alpha))$ 的输入上的布尔函数 $F \in TIME^f(2^{O(\alpha)n})$，使得 $F \notin Heur_{1/2-\varepsilon} - Size(2^{(1-O(\alpha))n'}, n')$

证明：设 $\varepsilon = 2^{\alpha n}/n^2$。如果 $f \in Heur_{\varepsilon} - Size(2^{(1-\alpha)n}/2,\ n)$，那么作为其见证的电路至多会产生 $2^{(1-\alpha)n}/n^2$ 个错误，这些错误可以通过一个规模为 $2^{(1-\alpha)n}/2$ 的电路予以纠正，从而得到一个规模为 $2^{(1-\alpha)n}$、能够精确计算该函数的电路。（实际上，使用更复杂的电路构造可以做得更好，但在这里不会有太大区别。）

现在，将前一个定理应用于 f，并使用这个 ε 和 $S = 2^{(1-\alpha)n}/2$，即可得到一个函数

$$F \in TIME^f(\mathrm{poly}(1/\varepsilon)) = TIME^f(2^{O(\alpha n)})$$

在输入规模 $n' = n+O($log $1/\varepsilon)$ $= n(1+O(\alpha))$ 下，使得 $F \notin Heur_{1/2-\varepsilon} - Size(S2^{-O(\alpha n)}, n')$ $= Heur_{1/2-2^{-\alpha n}} - Size(2^{(1-O(\alpha))n'}, n')$。

B.3 伪随机生成器的应用

这里，我们在构造伪随机生成器时使用了 [51], 提出的困难性放大引理，该构造将 [84] 的 PRG 构造与 [85, 15, 87] 的密码学伪随机生成器构造结合起来。

伪随机生成器是一个函数 $G\colon \{0,1\}^r \to \{0,1\}^m$，满足 $r < m$。如果对任意电路 $Circ \in Size(S,\ m)$，都有 $|\Pr_z[Circ(G(z)) = 1] - \Pr_R[Circ(R) = 1]| < \gamma$，其中 R是 $\{0,1\}^m$上的均匀分布，那么它以 γ-欺骗规模 S。

与 [51] 中使用的码相比，使用我们的码的优势在于，我们可以得到一个速度非常快的生成器 G。

Th定理 B.7。 存在一个 α_1，使得对每个 $\alpha_0 < \alpha_1$，以下结论均成立。假设 e

1. 存在能够抵抗多项式规模电路的单向函数

2. $f \in TIME(2^n)$ 不属于 $i.o. - Size(2^{n(1-\alpha_0)}\ n)$

那么，对所有足够大的 S，都存在一个$(S, 1/4)$-伪随机生成器

$$G : \{0,1\}^r \to \{0,1\}^S$$

其种子长度为 $r = (1 + O(\alpha_0))\log\ S$ 比特，且可在时间 $S^{1+O(\alpha_0)}$内计算

证明。 证明是标准的，与 [51], 中的证明完全相同，只需用推论 B.6 替换其初始放大步骤（等价地，用我们的码替换其对 Reed-Muller $+$Hadamard 的使用）。下面给出证明概要。

我们首先应用推论 B. 6 $TIME\ 2^{(1+O(\alpha_0\ \ n}$ $F \notin$，得到一个可在时间 ()) 内计算 $Heur_{1/2-2^{-\alpha_0 n}} - Size\ 2^{(1-O(\alpha_0\ \ n'}\ n' \qquad n' = n\ 1 + O\ \alpha$ 的函数，使得 ()), 对于 (( )) 成立。

若集合系统 $S_1, \ldots, S_m$ 由 $S_i \subset \{1, \ldots R\}$ 个集合组成，则称其为 (R, n', I, m)-NW 设计，若对于所有 i，都有 $|S_i| = n'$，且对于所有 $j \neq i$，都有$|S_I \cap S_J| \leq I$ [84] $R = O\ n' \qquad I = n'/4 \qquad m = 2^{\Omega(n'}$ 。给出了任意满足 ( ) 且满足) 的 NW 设计的构造。隐含地， [112] 描述了如何将此类设计转换为 $R = n(1 + O(\beta))$ $I = n'(1 - \beta)$ 设计。只需将 R 个元素划分为 $n(1 - 2\beta)$ 个固定元素 $Fixed$ 和一个包含 $O(\beta n)$ 个可变元素的集合。在可变元素上取一个 $n'' = 2\beta n, R' = O(n'')$ $= O(\beta n)$ $I' = 1/4n'' < \beta n$ 设计， $S'_1, ..S'_m$。 定义$S_i = S'_i \cup Fixed$。于是 $S_i \cap S_j = Fixed \cup (S'_i \cap S'_j)$，因此规模至多为 $|Fixed| + I' = n($ $1-2\beta n)+\beta n = n(1-\beta)$

我们的第一个生成器选择 $n' = (1 + O(\alpha_0)\ \log(S)$，并使用具有该设计的 NW 生成器，产生一系列 $m = 2^{\Omega(\alpha_0)n}$ 输入到 F，其规模为 n'，并将结果拼接起来。该生成器可在时间 $m * 2^{n'(1+\alpha_0)} = S^{1+O(\alpha_0)}$ 时间内计算。根据 [84],，该生成器的任何规模为 S 的区分电路都会产生一个针对 F 的、规模为 $S + O(2^I m)$ 的近似 $1/2 - \Omega(1/m)$ 电路，从而与 F 的困难性相矛盾。继[51],之后，我们再应用一个近线性时间的密码学伪随机生成器，将拉伸增加到 S。 □

C 编码理论中的应用

继 Trevisan [33], 之后，在本节中，我们从将 aLLDC 视为一种保持复杂度的距离放大工具的角度给出应用。这如今已是文献中非常标准的方法，也是许多列表译码算法的核心（参见例如 [106, 44, 77, 46]）。我们使用该策略，给出首个可在对数深度 （RNC^1）中进行列表译码的优良码族，以及具有最先进时间与查询复杂度的恒定码率近线性列表可译码码的新族。

我们指出，第一类码也可以构造成低密度奇偶校验码，或者能够在 NC^1 中编码。即使在唯一译码区间内，同时实现这三个性质（低密度奇偶校验码的对数深度编码和解码）仍然是一个悬而未决的问题——如果这样的码存在，我们将通过相同的方法立即得到列表译码的对应结果。

C.1 RNC^1 中的列表译码

并行计算主导着现代计算。纠错码也不例外，并且拥有悠久的并行化历史，可追溯到 Gallager 1960年的突破性论文 [47] 引入低密度奇偶校验 （LDPC）码，以及 Zyablov 和 Pinsker 的 [113] 相应低深度译码器。我们简要回顾一些背景。线性码 C 总是可以表示为相应“奇偶校验”矩阵 H的核。若该矩阵的行具有恒定汉明重量，则称该码为 LDPC。

在早期有影响力的工作中，Zyablov 和 Pinsker [113] 证明 Gallager 的随机 LDPC 码可由近线性规模电路并行解码。具体而言，他们给出了一种简单的对数深度置信传播译码器，对“大多数”随机 LDPC 码有效，并且至今仍是编码中非常标准的方法。在一项里程碑式的结果中，Sipser 和 Spielman [23] 著名地构造了 Gallager 码的显式 例子，称为扩展码，并配有相应的对数深度与线性时间译码器。他们的码以及简单的置信传播方案（其本身基于 Zyablov-Pinsker [22]）此后在编码理论中影响巨大，并且在某种意义上是组合译码算法中的“黄金标准”。

令人惊讶的是，在高噪声区域，对并行化的理解要少得多。具体而言，尽管已有大量卓有成效的研究，构造出了具有线性 [43] 或近线性时间全局列表译码[105, 114] 以及亚线性时间局部列表译码 [115, 115, 82, 44, 116], 的列表可译码码，但目前还没有任何优良码（甚至是存在性结果！）能够在优于 $\log^2(N)$ 深度的条件下实现列表译码。事实上，对于已知码的所有列表译码算法，要么采用代数方法，要么采用谱方法，要么依赖半定规划，因此据我们目前所知，它们至多属于 NC^2。

与 Zyablov-Pinsker [113] 和 Sipser-Spielman [23], 类似，我们的 aLLDC 完全基于消息传递，并具有低深度译码器。因此，将我们的码与任意对数深度唯一译码器复合，便立即得到了首批具有（随机化）对数深度译码器的优良列表可译码码构造。

在此定义列表译码模型时存在一些细微之处。我们将采用以下强列表译码（概率性）概念。若存在一个列表C_1，.....，$C_{\ell_{out}}$，其中的电路是对数空间一致、对数深度的随机化电路，使得对于任意 w，关于 r的内部随机性，C_i使列表 $\{C_1^w(\cdot,\ r_1)\text{.....},\ C_{\ell_{out}}^w(\cdot,\ r_{\ell_{out}})\}$ 包含 $\mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$ 中的每个函数的概率至少为 $_{3\,4}$，则称 $\mathcal{C}$在 RNC^1 中是 $(\frac{1}{2}+\varepsilon,\ \ell_{out})$-列表可译码的。注意，给定一个固定的 w后，只需在独立随机性下多次运行该电路族，就可以轻松地放大成功概率，并在 NC^1 中将上述列表剪枝至其组合列表大小（通常为 $O(\frac{1}{\varepsilon^2})$）。不过，下面我们仍将列出未经剪枝的列表大小；该大小与码字 w无关。这一模型的优势在于，只需一次生成这个稍大的随机化电路列表，而不必每次收到新的码字 w 时都重新生成。

定理 C.1（RNC 中的列表译码[1]（常数码率区域））。存在一个普适常数 $C>0$，使得对于任意 $\varepsilon>0$ 以及无穷多个 $N\in\mathbb{N}$，存在一个显式线性码 $\mathcal{C}_N\colon\ \{0,1\}^N\to\{0,1\}^{N'}$，该码是 $(\frac{1}{2}+\varepsilon,\ \ell_{out})$-列表可译码的，并且

1. ***Rate:*** $R\geq\exp(-O(1/\varepsilon^3))$
2. ***List Size:*** $\ell_{out}\leq\frac{\mathrm{polylog}(\frac{1}{\varepsilon})}{\varepsilon^2}$

此外，$\mathcal{C}_N$ 可在 ***RNC***^{1} 中使用规模为 $N\log(N)^{\exp(O(1/\varepsilon^3))}$ 的电路进行列表译码。

根据具体实例化方式，这些码可以构造为 LDPC（Siper-Spielman [23]），或者可在 NC^1 中进行编码（Spielman [117]）。

对于较小的 $\varepsilon\leq\frac{1}{\log N}$，我们可以采用多对数码率编码，从而得到以下改进的构造：

定理 C.2（NC 中的列表译码[1]（次常数错误范围））。对于无穷多个 $N\in\mathbb{N}$ 以及任意 $\varepsilon\leq\frac{1}{\log N}$，存在一个显式线性码 $\mathcal{C}_N\colon\ \{0,1\}^N\to\{0,1\}^{N'}$，该码是 $(\frac{1}{2}+\varepsilon,\ \ell_{out})$-列表可译码的，并且

1. 码率：$R\geq$多项式(ε)
2. 列表大小：$\ell_{out}\leq\frac{\mathrm{polylog}(\frac{1}{\varepsilon})}{\varepsilon^2}$

此外，只要满足 $\varepsilon\geq\exp(-\tilde{O}(\sqrt{\log N}))$，这些码就可以用规模为 $N\mathrm{poly}(\log(N)\ \frac{1}{\varepsilon})$ 的电路在 ***RNC***^{1} 中解码。

我们注意到，通过将原始的大字母表 aLLDC 与任意一个可从 $1-\varepsilon$ 个差错中列表译码的内码进行串接，也可以实现从 $1-\varepsilon$ 个差错中进行列表译码，代价是使用更大的字母表。为简单起见，我们这里只给出二元情形。

如前所述，将我们的码与适当的基础码相结合，即 Sipser-Spielman [23]（用于低密度奇偶校验性质）或 Spielman [117]（用于快速编码），即可基本上直接得到这些结果的证明：

定理 *C.1* 和定理 *C.2* 的证明。 我们只证明前者。后者可用类似方法证明。我们的构造将由两个码复合而成

1. 具有 $\mathcal{C}_{unique}$ 的线性二元基码
   - 常数码率和距离
   - 一个 NC^1、规模为 N polylog(N) 的解码器，能够从常数 $c > 0$ 错误比例中解码，并且
   - 要么可在 NC^1 [117], 中编码，要么是 LDPC [23]
2. 我们的常数码率线性二元 aLLDC C_{apx} ，来自推论 5.2

更具体地，固定任意 $N \in \mathbb{N}$ ，使得我们的常数码率族推论 5.2 中存在一个在 N 个消息比特上的码，该码是 $\frac{c}{100}$-近似（ $\frac{1}{2} + \varepsilon$， $\tilde{O}(1/\varepsilon^2)$）-列表可译码的（回顾一下，将 δ_{out} 进行常数倍放大始终很容易；为简单起见，在推论 5.2 中其形式化取值仅为 $1/10$）。注意，由于这些解码电路被承诺为对数空间一致、规模为 log(N)$^{2^{O(1/\varepsilon^3)}}$、深度为 O_ε(poly log log(N))，因此，只需对每个消息位并行运行解码器，就能得到一个用于全局（近似）解码的 NC^1 算法，其规模为 $N \log(N)^{2^{O(1/\varepsilon^3)}}$ ，正如该断言中所述。最后请注意，我们码的生成算法与 w 无关。在正文中，它唯一的任务是选择一个随机的初始超边（以及可能的列表元素）以进行传播。这可以由电路自身在内部完成，因此这些电路确实只需生成一次，之后即可适用于所有输入码字 w。[63]

现在，利用任意对数空间构造的常数度扩张图（例如 [118–120] 或我们构造的 KMS 复形的 1-骨架）， [23] 和 [117] 的扩张码框架分别意味着存在二进制线性基码 $\mathcal{C}$ ，其码长为 N ，并且具有低密度奇偶校验性质或 NC^1 编码。[64]我们的码将通过以下方式给出：先用 $\mathcal{C}_{unique}$对消息编码，再用我们的常数码率 aLLDC C_{apx}对输出进行编码。将这个复合码记为 $\mathcal{C} = C_{apx}(\mathcal{C}_{unique})$。注意，所声称的码率和线性性质可直接由构造得出。

现在我们证明，这个复合码确实能在 NC^1 中从比例为 $\frac{1}{2} - \varepsilon$ 的错误中进行列表译码。首先，根据标准的直积和级联分析，注意到可以使复合码的距离远大于 $\geq \frac{1}{2} - O(\varepsilon^2)$。因此，Johnson 界（例如参见 [121, 122]）保证该码的组合列表大小至多为 $O(\frac{1}{\varepsilon^2})$。

现在，固定一个输入字 w 以及任意 $f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$。运行我们的 aLLDC 解码器，对于 $\mathcal{C}_{apx}$ 输出以至少 $3/4$ 的概率（可能具有随机性）得到 NC^1 电路 C_1，……， $C_{\ell_{out}}$ 其中一个 $\frac{c}{100}$-计算$\mathcal{C}_{unique}(f)$。对于每个 C_i 固定一个内部随机性的随机设置，则有一个输出电路以至少 $2/3$ 的概率确定性地 c-计算 $\mathcal{C}_{unique}(f)$。最后，由于 $\mathcal{C}$ 的组合列表大小至多为 $O(1/\varepsilon^2)$，将上述过程重复 $O(\log(1/\varepsilon))$ 次，可确保每个 ε-计算的函数 f 以较高的常数概率（例如 99%）被列表中的某个电路 c-计算。

现在，我们可以将这些电路并行馈送给用于 $\mathcal{C}_{unique}$的解码器，该解码器也属于 NC^1。根据唯一译码保证，假设上一步成功，该解码器将输出一个稍大的 $\tilde{O}(1/\varepsilon^2)$函数列表，并保证其中包含每个 $f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$。如果需要，还可以在 NC^{1}中通过移除重复项并检查其与码字 w 的一致度，将该列表剪枝至正确的 $O(1/\varepsilon^2)$ 大小（尽管这是一项“w-特定”的操作，因此如前所述，形式模型中的列表大小仍为$\frac{\text{多对数}}{\varepsilon^2}^{\frac{1}{\varepsilon}}$。

最后，我们指出，若将基码取为 Sipser-Spielman 的 LDPC 扩展码 [23], ，则只要我们在推论 5.2 中与之级联的内码是 LDPC 且系统化的（后者使得直积等式约束具有较短的校验方程），整个构造就是 LDPC 的。类似地，若采用 Spielman 码，则基码与 aLLDC 的编码均属于 NC^1, ，因此复合码的编码也属于 NC^1 ，符合预期。 □

C.2 具有次多项式时间解码的新 LLCC

关于局部（列表）可译码与可纠错码的构造，已有极为广泛的文献，范围从经典代数方法（如 Reed-Muller [123, 7, 14], ）到结合代数与组合

[63]在困难性放大中，这并不合理，因为此时不存在计算原函数的单一固定电路。然而在列表译码中，我们并不在意同一电路在不同运行中是否输出不同的有效消息。[64]请注意，尽管此类扩展图未必对所有 N 都存在，但它们存在于任意目标 N 的常数因子范围内，因此我们可以直接将码长填充至所需的 N ，同时保持常数码率与解码半径。

方法，如匹配向量码 [124, 125], ，乃至纯组合/随机构造（例如稀疏随机线性码） [126, 78]。传统上，大多数局部可译码的构造仅在次常数码率区域有效。这在一定程度上是因为该区域作为复杂度和密码学中的原语用途强大，同时也是因为在现代编码理论、数据存储和算法设计中备受关注的高码率区间严重缺乏工具。事实上，过去20年来日益增多的工作证明了这一点：这些工作构造了具有高码率的松弛 版本局部可译码 [127–132] ，以及相关的对常数码率局部可测试 码 [28, 26],的探索，并最终在近期工作中达到顶点，实现了较弱的松弛局部译码，具有常数码率且仅需多对数查询 [133, 134]。

回到标准的局部可解码码，过去 15 年来，高码率区间也迎来了一次复兴。多年来，已知的唯一可进行次线性时间解码的码只有 Reed-Muller 码（在某些参数设置下，它们可以实现 k^{α} 时间解码，码率为 $\alpha^{\Omega(1/\alpha)}$ ），而自 2010年左右开始的多项研究成功实现了码率高得多（甚至接近 1）的 k^{α} 时间唯一译码，其中包括重数码 [79], 、提升码 [80, 81],、扩展码 [83], 和张量码 [44]。

在现代数据存储等应用中，次线性时间解码远远不够；人们确实希望能有次多项式时间、甚至是已知适用于次常数码率码的多对数时间译码器。Kopparty、Meir、Ron-Zewi 和 Saraf [38], 首次在常数码率区域实现了这类码：他们将重数码与经典的组合放大技术 [42] 相结合，实现了次多项式时间局部译码器。在 [39, 40], 中，作者指出重数码还具有列表可恢复性，因此能够将次多项式时间解码扩展到直至容量的列表译码范围，但查询复杂度也因此显著增加，从 $\exp(\sqrt{\log\ N})$ 增至 $\exp(\log^{3/4}(N))$。利用我们的码，我们构造了新型常数码率局部列表可解码码，实现了前者的性能，不过我们的码距离容量还很远。

更一般地，我们下文采用的方法适用于任意基础局部可解码码，特别是，它推出了第 1.3 节中所述的断言：将局部可解码码放大为局部列表可解码码时，时间和查询复杂度至多产生多项式膨胀（忽略对 ε 的依赖），且保持复杂度。因此，构造运行时间为 $\mathrm{polylog}(N)$ 的局部列表可解码码，唯一的障碍就在唯一译码区间，而这仍是一个长期悬而未决的问题。

在给出主要结果之前，我们先简要介绍本工作中的码所满足的更强局部纠错概念，因为此前尚未在本文中正式定义过。为更贴合这一主题的编码文献，本节将不再采用解码电路的概念，而改用经典的有时间界限的解码方式，即使用能够查询访问码字 w 的预言机（图灵）机。我们也将改回更标准的保证：输出列表应覆盖输入字 w 附近的每个函数，而不是以高概率覆盖每个固定的此类函数。现在我们给出局部列表可纠错码的标准定义。如前所述，这类码能够恢复特定索引处最近的码字的值，而非消息的值：

定义 C.3 （局部列表可纠错码（LLCC））。 **</style}一个码：** $\{0,1\}^k \to \{0,1\}^n$ 若存在一个概率算法 A ，可生成随机化算法M_1、.....、$M_{\ell_{out}}$，使得对所有 $w \in \{0,1\}^n$：则称其为在时间 T 内 $(\frac{1}{2}+\varepsilon, q, \ell_{out})$ -局部列表可纠错。

$$\Pr_A\left[\forall f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)\ \exists i, \forall j \in [n]:\ \Pr[M_i{}^{w,r}(j) = Enc(f)_j] \geq \frac{3}{4}\right] \geq \frac{3}{4}.$$

此外，每个 M_i 查询 w 最多涉及 q 个位置，并且 M_i 以及生成 M_i 描述的算法 A ，都在时间 T 内运行。

如果 LLCC 是系统化的（或者以其他方式在编码中直接编码消息），它便自动具有局部（列表）解码能力。我们的码满足这一点，因此我们将只关注局部纠错。本节的主要结果是以下具有次多项式时间解码能力的 LLCC

定理 C.4 （具有次多项式时间解码器的二进制局部列表可纠错码）。 对于每个 $\varepsilon > 0$ 以及无限多个 $N \in \mathbb{N}$，都存在一个显式的二进制线性 $(\frac{1}{2}+\varepsilon, q, \ell_{out})$ -*LLCC* 无限族，满足

1. 码率： $R \geq \exp(-O(1/\varepsilon^3))$

2. 查询与时间复杂度： $q,\ T \leq$ 指数 $\left(O\left(\sqrt{\text{对数}\ N \cdot \text{对数对数}\ N}\right)\right)$

3. 列表大小： $\ell_{out} \leq$ 多对数$(\frac{1}{\varepsilon})\varepsilon^2$

请注意，为简洁起见，上文聚焦于二元情形。我们也可以轻松地构造具有相同参数、可从半径 $1-\varepsilon$ 进行解码的大字母表码：只需将我们的大字母表近似局部列表可译码码与该参数范围内任意一个足够良好的列表可译码码串接即可（而不是像这里这样与二元码串接）。上述码也具有相同参数的局部列表可解码性。

定理 C.4 是由推论 5.2 和 [38, 定理 1.1]直接得出的推论。 :

定理 C.5 ([38, 定理 1.1])。 对于每个 $r \in (0,1)$，存在 $\delta > 0$，且对于足够大的 $N \in \mathbb{N}$，存在一个显式二进制线性码 C_N，满足：

1. C_N 具有块 k 长度 $N' \in [N, N2^{O(\sqrt{\log N \cdot \text{对数对数 } N})}]$，码率至少为 r，且相对距离至少为 δ，
2. C_N 可从比例为 $\delta/2$ 的错误中局部纠正，运行时间至多为 $\exp\big(O(\sqrt{\log\ N \cdot \log\log\ N})\big)$

我们指出，关于 N' 规模的陈述并未在 [79], 中明确给出，但只要考察其构造的参数便可立即得出。

我们即将开始证明定理 C.4。证明基本上如预期进行，不过有一个细节：在两种情形下，我们都有适用于无穷多个但并非所有 N的码，因此必须谨慎确保消息长度与码长适合进行复合。事实上，为了处理这一点，我们需要下面这个推论 5.2 的变体，以处理超常数规模的消息字母表。

推论 C.6 (具有大输入字母表的二元 aLLDC)。 存在一个普适常数 $C > 0$，使得对于所有足够大的 $M \in \mathbb{N}$ 以及满足 $^{C}{}_{\log\log\ M} < \varepsilon < 0$ 的情形，~~都存在一个~~ $N \in [M, M\log(M)^{2^{O(1/\varepsilon^3)}}]$，以及一个显式二元线性码 $\mathcal{C}$：$\Sigma_{in}^N \to \{0,1\}^{N'}$，它是一个 0.1- 近似的 $(\frac{1}{2}+\varepsilon, \tilde{O}(\frac{1}{\varepsilon^2})\ q)$- 局部列表可解码码，~~满足~~

1. 码率： $R \geq 2^{-O(1/\varepsilon^3)}$

2. 时间与查询复杂度： $q \leq \log(\Sigma_{in})\log(N)^{2^{O(1/\varepsilon^3)}}$

证明完全相同，只需考虑读取输入符号并计算其相等性所增加的时间，以及对最终输出对数字母表大小 $\log\ \Sigma_{in} \cdot \log(N)^{2^{O(1/\varepsilon^3)}}$ 进行级联所需的时间（注意，现在我们不处于电路模型中，只要求级联中使用的内层列表可译码码具有多项式时间编码和解码）。现在我们准备证明良好的次多项式时间局部列表可译码码的存在性：

定理 *C.4* 的证明。 固定某个常数 $r \in (0,1)$ 以及相应的 $\delta_{out} = \frac{\delta}{2}$，其中 $\delta_{out} = \frac{\delta}{2}$ 对应于定理 C.5 中码族的唯一局部译码半径。令 $N \in \mathbb{N}$ 为任意一个整数，使得存在一个 $\frac{\delta}{200}$-近似 $(\frac{1}{2}+\varepsilon, \tilde{O}(\frac{1}{\varepsilon^2}), \text{poly}(\frac{\log N}{\varepsilon}))$-局部列表可解码码 $\mathcal{C}_{apx}$，其码率为 $\exp(-O(1/\varepsilon^3))$，~~如~~推论 5.2 所述。[65]——

现在，定理 C.4 保证存在一个二元码 $\mathcal{C}_{unique}$，其码长为 $N' \in [N, N2^{O(\sqrt{\log N}\log\log\ N)}]$。直接来看，我们无法将 $\mathcal{C}_{unique}$ 与 $\mathcal{C}_{apx}$ 复合，因为前者的码长 N' 与后者的消息长度 N 不匹配。幸运的是，有一种标准技巧，可以基于 $\mathcal{C}_{unique}$ 构造一个更大字母表 的码 $\mathcal{C}_{unique}^{fold}$，使其码长恰好为 N'。具体而言，我们只需将该码折叠，把每个由 $\lceil \frac{N'}{N} \rceil$ 比特组成的块视为一个字母表符号（即“打包”这些符号），并在最后一块填充 0，使其达到相同的字母表。因此，最终的码就是复合码 $\mathcal{C} = \mathcal{C}_{apx}(\mathcal{C}_{unique}^{fold})$。

现在我们给出针对 $\mathcal{C}$ 的局部纠错算法。和往常一样，固定一个输入字 w 和一个函数 $f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$。运行针对 $\mathcal{C}_{apx}$ 的局部列表译码算法，我们得到（随机化的）预言机机器 M_1、.....、$M_{\ell_{out}}$（用于 $\ell_{out} \leq \tilde{O}(1/\varepsilon^2)$），使得以至少 $3/4$ 的概率，这些机器中至少有一~~台能够在~~ $2^{O(\sqrt{\log(N)}\log\log(N))}$ 时间内，通过访问 $\overline{w}$ 的预言机来计算 $\mathcal{C}_{unique}^{fold}(f)$。根据马尔可夫不等式，这些机器的内部随机性取随机值时，因此有 99% 的概率能够计算 $\mathcal{C}_{unique}^{fold}(f)$。注意，由于根据 Johnson 界，我们的复合码的组合列表大小上界至多为 $O(1/\varepsilon^2)$（如定理 C.1 的证明所示），将此过程重复 $O(\log(1/\varepsilon))$ 次，可确保以高概率，每个 $f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$ 都有一个能计算它的 M_i。

[65]再次注意，只需调整构造中的常数，或与适当的扩展图复合，就能轻松实现任意常数 $\delta/200$ 近似解码。我们在定理陈述中设定 $\delta_{out} = 0.01$ 只是为了具体说明。

现在，假设给定一个我们想要纠正的码字索引（s，i），其中 s 表示原始码直积 $\mathcal{C}_{(V,S)}$ 中的超边，而 i 是级联内码的索引。要纠正（s，i），只需正确计算位置 s 处（级联前）的大字母表编码，然后用内码对其编码并输出第 i 位。注意，根据内码具有多项式时间解码器这一假设，后一步最多需要 $2^{O(\sqrt{\log(N)}\log\log(N))}$ 时间。

现在，由于 $C_{(V,S)}$ 是直积码，要计算 s 处的完整符号，显然只需正确计算每个 $v \in s$ 处的编码值 $\mathcal{C}_{unique}^{fold}(f)$（每个 $v \in s$ 本身都是一个包含来自 $\mathcal{C}_{unique}^{fold}(f)$ 的 $2^{O(\sqrt{\log(N)}\log\log(N)}$ 个编码比特的“块”）。因此，如果 M_i 是一个预言机机器，它能以 $\frac{\delta}{2}$ 计算 $\mathcal{C}_{unique}(f)$，我们就可以对每个 $v \in s$ 所对应块中的每个编码比特运行 $\mathcal{C}_{unique}$ 的唯一纠错算法。局部纠错器保证以 $\frac{3}{4}$ 的概率输出正确比特，因此，对 $O(\log(N))$ 次重复编码并取多数答案，就足以确保以（比如）99% 的概率正确编码每个 $v \in s$ 对应的整个块。综上，我们的最终列表解码器只需输出解码器列表：对上文得到的每个预言机机器 M_i，都将 $\mathcal{C}_{unique}$ 应用于该机器；每个解码器最多进行 $2^{O(\sqrt{\log(N)}\log\log(N))}$ 次查询。将上述过程重复常数次后，这确实能以至少 $3/4$ 的概率对每个 $f \in \mathcal{B}_{\frac{1}{2}-\varepsilon}(w)$ 进行局部纠错，如所需。

□

D 内码分析

在本附录中，我们证明命题 5.7 和命题 5.8，即针对常数码率码的内部解码器和剪枝过程。为方便起见，我们在此重述算法和相关断言。正如正文中所述，这里给出的算法与 [36] 中针对稠密系统的原始 aLLDC 非常相似。目前，我们固定 In_u、输入字 $w_{In_u} = \{w_s\}_{s \in S_u}$ 和参数 ε，$\delta_{in} > 0$。解码电路定义见图 12。

内部解码器 $InDec_{\delta_{in}}(s, v)$:

* **输入：**
  - 一个集合 $s \in S_u$。
  - 一个目标顶点 $v \in V$。
* **输出：** Σ 中的一个值。

1. 对于 $j = 1, 2, \ldots, \frac{4\log\frac{1}{\delta_{in}}}{\varepsilon}$:
   (a) 抽取一个集合 $s'_j \ni v$。
   (b) 测试是否：
      i. 集合 s 与 s'_j 相交，and
      ii. 如果 $\mathrm{dist}^{s \cap s'_j}(w_s, w_{s'_j}) < 4\delta_{in}$。
   (c) 退出循环并输出 $g_j(v)$。
2. 如果没有任何一个 s' 通过，则输出任意符号。

图 12：内部解码器

正如预期，生成算法抽样 $\frac{10}{\varepsilon}$ 集合 $s \in S_u$ 并输出计算 $InDec_{\delta_{in}}(s, \cdot)$ 的电路。

断言（断言 5.7 的重述）。设 ε、$\delta_{in} > 0$，并设 η 如命题 *5.4* 中所述。设 $u \in U$ 和 In_u 为命题 *5.4* 中的集合系统，即

1. 集合系统 (V_u, S_u) 是一个 η-采样器。

2. 对于每个 $s' \in S_u$，局部交图[66] $(G_{In_u})_{s'}$ 是一个 η-采样器。

3. 局部解码图 G_{In_u} 是$(1-O(\eta))|S_u|$ 正则的。

那么， In_u 上的直积码是$(\delta_{in}, 13\sqrt[3]{\delta_{in}})$-近似的$(\varepsilon, \frac{10}{\varepsilon}, \overline{q = \mathrm{poly}(\frac{1}{\delta_{in}}, \frac{1}{\varepsilon})})$-局部列表可解码码，其解码采用对数空间一致规模$(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，规模和深度为 $\tilde{O}(\log\log N \exp(O(\frac{1}{\varepsilon})))$。

备注 D.1 (距离测试与细粒度查询复杂度)。 我们通过附录 F 中的距离测试器，在步骤(b) 执行距离测试，而无需查询整个交集。特别是，我们可以通过概率方式实现这一点：在 $s \cap s'$ 中查询 $m = \log\left(\frac{100}{\delta_{in}\varepsilon}\log\frac{1}{\delta_{in}}\right)\frac{1}{\delta_{in}^2}$ 个点，并以置信度 $1-\delta_{in}$ 对所有被检验的函数，将经验距离计算至精度 δ_{in}^2。因此，命题 *5.7* 的查询复杂度中应出现的表达式是$4\log\frac{1}{\varepsilon}\frac{1}{\delta_{in}}\cdot\log\left(\frac{100}{\delta_{in}\varepsilon}\log\frac{1}{\delta_{in}}\right)\frac{1}{\delta_{in}^2}$

在命题 *5.7* 的证明中，为简便起见，我们将直接假设该测试以确定性方式通过。我们请细心的读者参阅定理 *3.4* 的证明，了解一般情形是如何处理的。

我们说明，我们没有尝试优化查询复杂度和输出近似参数 $13\sqrt[3]{\delta_{in}}$，因为它大致设为 $\approx \exp(-\frac{1}{\varepsilon})$，而外层解码器无论如何都会将其缩小 $\approx \exp(\frac{1}{\varepsilon})$ 倍。更仔细地处理，很可能可以使 δ_{in} 与 δ_{out} 之间呈线性关系，参见例如 [36]。此外，我们指出，如有需要，可以让生成算法预先固定输出电路的随机性，从而对输出电路进行去随机化。

命题 *5.7* 的证明。 在证明中，我们将用到这样一个事实： $In_u = (V_u, S_u)$ 本身以及交图 $(G_{In_u})_s$ 都是 η-采样器。特别地，在命题的陈述中，我们选取足够小的 $\eta > 0$，使得在每个 η 作为乘法因子出现的不等式中，所得项都严格小于不等式另一侧的项。即使再开平方，这一点仍然成立。例如，如果我们写出形如 $\sqrt{\eta\delta_{in}^2} \le \varepsilon^3$ 的表达式，这是因为 η 相对于 δ_{in}，ε 被设得足够小，从而确保这一点。为清楚起见，我们在此说明一次，而不在正文中反复验证。

查询复杂度和列表大小可由该构造直接得出。我们必须证明，对于任意函数 $f \in \mathcal{B}_{1-\varepsilon,\delta}(w_{In_u})$，我们以概率 $3/4$ 输出一个与 f 为 $13\sqrt[3]{\delta_{in}}$-接近的电路。

令 $A = \{s \in S_u : \mathrm{dist}(w_s, f) \le \delta_{in}\}$，并注意由假设可知 $\mathbb{P}[A] \ge \varepsilon$。我们希望论证存在一个概率为 $\mathbb{P}[A'] \ge \frac{\varepsilon}{2}$ 的子集 $A' \subseteq A$，使得对每个 $s \in A'$，从 s 进行的解码都以高概率成功

$$\mathop{\mathbb{P}}_{v\in V_u, r}[InDec_{\delta_{in}}(s,v) = f(v)] \ge 1 - 13\sqrt[3]{\delta_{in}} \tag{11}$$

这里 r 是解码电路所使用的随机字符串。若此成立，则我们已证明所需结论，因为生成算法抽样了足够多的 s，从而以大于 $3/4$ 的概率命中 $s \in A'$。

如同 in 定理 3.4，为定义 A'，若对任一 $s'' \in \{s, s'\}$，我们将边 $\{s, s'\}$ 称为“坏的”

$$\left|\mathop{\mathrm{dist}}_{s''}(f, w_{s''}) - \mathop{\mathrm{dist}}_{s\cap s'}(f, w_{s''}|_{s\cap s'})\right| \ge \delta_{in}.$$

回忆一下，图 $(G_{In_u})_{s''}$ 是一个 η-采样器。坏边恰好是那些交集 $s \cap s'$，它们对 s'' 中顶点所占比例的抽样有 δ_{in} 误差，其中 $f \ne w_s$；因此，根据采样保证，对于每个 s'，这类交集至多有 $\frac{\eta}{\delta_{in}^2}$ 个。由此，整体上至多有 $2\cdot\frac{\eta}{\delta_{in}^2}$ 比例的“坏边”，而马尔可夫不等式表明，至多有 $\sqrt{\frac{2\eta}{\delta_{in}^2}}$ 个集合 s 与超过 $\sqrt{\frac{2\eta}{\delta_{in}^2}}$ 比例的坏边相连。根据 η的选择，这一比例小于集合的 ε^3。因此我们定义

$$A' = \left\{ s \in A : s \text{ is connected to less than } \sqrt{\frac{2\eta}{\delta_{in}^2}} \text{ bad edges} \right\}.$$

当然， $\mathbb{P}[A'] \ge \frac{\varepsilon}{2}$，特别是我们的生成算法以 ≥ 0.9 的概率命中 A'。

[66]回忆一下，这就是这样一张图：一侧是 s中的顶点，另一侧是与 s相交的 $s' \in S_u$ 对应的交集 $s' \cap s$。

仍然　　证明对于任意 $s \in A'$，$InDec_{\delta_{in}}(s,\cdot)$ 以 $1-13\sqrt[3]{\delta_{in}}$ 的概率计算出 $f(v)$　　.

固定 $s \in A'$。我们将定义一个顶点集合 $V' \subseteq V$，并证明 (1) $\mathbb{P}[V'] < 5\sqrt[3]{\delta_{in}}$，以及 (2) 若 $v \notin V'$，则

$$\mathop{\mathbb{P}}_{r}[InDec_{\delta_{in}}(s,v) = f(v)] \geq 1 - 8\sqrt[3]{\delta_{in}}. \tag{12}$$

这将证明 (11)，因为未能正确解码的概率以 $\mathbb{P}[V'] + 8\sqrt[3]{\delta_{in}} \leq 13\sqrt[3]{\delta_{in}}$ 为上界。

定义 V'： 令 M 为在步骤 (2) 中通过距离测试的集合 $s' \in S_u$（对于某个如上所定义的固定 $s \in A'$；特别地，这是 s 的邻居的一个子集）。集合 V' 将由所有满足以下至少一个条件的顶点 $v \in V$ 组成：

1. $\mathop{\mathbb{P}}_{s' \in S_u}[M | s' \ni v] < \frac{\varepsilon}{2}$。2. $\mathop{\mathbb{P}}_{s' \in S_u}[w_{s'}(v) \neq f(v) | s' \in M,\ s' \ni v] > 7\sqrt[3]{\delta_{in}}$。明确地说，这是译码算法在 v 上输出错误的概率（假设算法没有在 M 中找不到集合）。

首先确认，如断言所说，如果 $v \notin V'$，那么 (12) 确实成立。

$v \notin V' \Rightarrow$ (12) 成立： 如果

$$\mathop{\mathbb{P}}_{s' \in S_u}[M | s' \ni v] \geq \frac{\varepsilon}{2},$$

$4\log$，那么我们选中集合 $s' \in M$ 的概率为 $\geq 1 - (1 - \frac{\varepsilon}{2})^{\delta_{in}} \geq \frac{1}{1-} \sqrt[3]{\delta_{in}}$。发生这种情况时，在给定 $s' \ni v$ 的条件下，集合 $s' \in M$ 是均匀的。因此，根据第二项，在退出循环时选中的 s' 以概率 $\leq 7\sqrt[3]{\delta_{in}}$ 满足 $w_{s'}(v) \neq f(v)$，于是

$$\mathop{\mathbb{P}}_{r}[InDec_{\delta_{in}}(s,v) = f(v)] \geq (1 - \sqrt[3]{\delta_{in}})(1 - 7\sqrt[3]{\delta_{in}}) \geq 1 - 8\sqrt[3]{\delta_{in}}.$$

界定 $\mathbb{P}[V'] < 5\sqrt{\delta_{in}}$： 我们将证明，定义 V' 的两个条件各自仅以概率 $O(\sqrt{\delta_{in}})$ 被违反。粗略地说，我们的证明策略是说明上文定义的集合 M 大致就是能够计算 $O(\delta_i n)$ 的 s' f：

$$M \approx \left\{ s' : \mathrm{dist}_{s'}(w_{s'}, f) \leq O(\delta_{in}) \right\}.$$

接下来我们将论证 $\mathbb{P}[M] \approx \mathbb{P}[A] \geq \frac{3\varepsilon}{4}$，因此通过采样，大多数顶点都具有 $\mathop{\mathbb{P}}_{s' \in S_u}[M | s' \ni v] \geq \frac{\varepsilon}{2}$，从而证明违反第一项的顶点所占比例为 $\sqrt{\delta_{in}}$。为处理第二项，我们将论证，由于大多数 $s' \in M$ 具有 $\mathrm{dist}(w_{s'}, f) = O(\delta_{in})$，解码在平均意义下能够成功，因此根据马尔可夫论证，除去 $\sqrt[3]{\delta_{in}}$ 比例的顶点外，其余所有顶点都必须违反 V' 定义中的第二项。具体细节如下。

我们首先论证，大多数满足 $w_s \approx f$ 的集合 s' 都属于 M；反过来，$s' \in M$ 意味着 w_s 接近 f。形式化地，

1. **每个 $s' \in A$ 邻居 s 满足 $\{s, s'\}$ 不是坏边的条件时，都属于 M：** 根据假设，二者均满足距离 $s(w_s, f)$，距离 $s'(w_{s'}, f) \leq \delta_{in}$。边 $\{s, s'\}$ 不是坏边，因此

$$\mathop{\mathrm{dist}}_{s \cap s'}(w_{s'}|_{s \cap s'}, f), \mathop{\mathrm{dist}}_{s \cap s'}(w_s|_{s \cap s'}, f) < 2\delta_{in},$$

但根据三角不等式

$$\mathop{\mathrm{dist}}_{s \cap s'}(g|_{s \cap s'}, g'|_{s \cap s'}) < 4\delta_{in}$$

因此 s' 通过了步骤 (2) 中的测试。

2. 如果 $s' \in M$ **且** $\{s, s'\}$ **不是坏边，则** $\mathsf{dist}(w_{s'}, f) < 6\delta_{in}$**：** 由于 $\mathsf{dist}_s(f, w_s) < \delta_{in}$ 且 $\{s, s'\}$ 不是坏边，我们有 $\mathsf{dist}^{s\cap s'}(f, w_s|_{s\cap s'}) < 2\delta_{in}$。另一方面，由于 s' 通过了测试，根据三角不等式：

$$\operatorname*{dist}_{s\cap s'}(f, g'|_{s\cap s'}) < \operatorname*{dist}_{s\cap s'}(f, g|_{s\cap s'}) + \operatorname*{dist}_{s\cap s'}(g, g') \le 6\delta_{in},$$

并且由于这条边不是坏边，我们有 $\mathsf{dist}_{s'}(f, g') < 7\delta_{in}$。

我们将使用 这些断言来界定 V 的定义中第 1 项和第 2 项的概率 ′.

第 1 项的证明 上面的第一个断言意味着

$$\mathbb{P}[M] \ge \mathbb{P}[A] - \mathbb{P}_{s'}[s' \text{ is not a neighbor of } s] - \mathbb{P}_{s'}[\{s', s\} \text{ is bad}].$$

回想一下， $\mathbb{P}[A] \ge \varepsilon$。根据关于密度的假设， $\mathbb{P}_{s'}[s' \text{不是 } s] \le \eta$ 的邻居；根据关于 s的假设， $\mathbb{P}_{s'}[\{s', s\}$ 是坏的$] \le \sqrt{\frac{2\eta}{\delta_{in}^2}}$。因此，根据关于度的假设 d， $\mathbb{P}[M] > \frac{3\varepsilon}{4}$。通过在$V_u$和 S_u之间进行采样，满足 $\mathbb{P}[M|s' \ni v] < \frac{\varepsilon}{2}$的顶点少于 $\frac{4\eta}{\varepsilon^2} \le \sqrt[3]{\delta_{in}}$个；这证明在 V'的定义中，只有 $\sqrt[3]{\delta_{in}}$个顶点违反第 1 项。 —

第 2 项的证明： 现在我们说明，满足第 2 项的 $v \in V$所占比例至多为 $4\sqrt[3]{\delta_{in}}$。为此，如果 $s' \in M$，但 $w_{s'}(v) \ne f(v)$，就称数对（v， s'）是误导对。我们断言，误导对所占比例至多为 $8\delta_{in}\mathbb{P}[M]$。为此，注意到

$$\Pr_{v\in s'}[(v, s') \text{ is misleading}] = \Pr[M] \cdot \Pr_{v\in s'}[w_{s'}(v) \ne f(v) \mid s' \in M] = \Pr[M] \cdot \Pr_{v\in s'}[dist(w_{s'}, f) \mid s' \in M].$$

另一方面，如果 $(s,\ s')$ 不是坏边，则有 $dist(w_{s'},\ f) \le 6\delta_{in}$，因此

$$\Pr_{v\in s'}[(v, s') \text{ is misleading}] \le \Pr[M] \cdot (6\delta_{in} + \Pr_{s'}[(s, s') \text{ is bad} \mid s' \in M]) \le 8\delta_{in} \Pr[M]$$

其中，在最后一个不等式中，我们利用了以下事实：根据假设，$\Pr[M] \ge \frac{3\varepsilon}{4}$ 至多有 a $\sqrt{\frac{2\eta}{\delta_{in}^2}}$ fraction 的 $(s',\ s)$ 是坏的，并且 η 足够小。

现在，回想一下，如果第二项不成立，将不等式改写后可得

$$\Pr_{s'}[w_{s'}(v) \ne f(v) \wedge s' \in M \mid s' \ni v] \ge 7\sqrt[3]{\delta_{in}} \Pr_{s'}[M \mid s' \ni v].$$

那么，只要 $\Pr_{s'}[M \mid s' \ni v] \ge \Pr_2^{[M]}$（通过采样以及 $\mathbb{P}[M] \ge \frac{3\varepsilon}{4}$ 这一事实，除概率为 $\frac{4\eta}{\mathbb{P}[M]^2} < \sqrt{\delta_{in}}$ 的情况外，这一点都会成立），我们就有

$$\Pr_{s'}[(v, s') \text{ is misleading } | s' \ni v] \ge \frac{7}{2}\sqrt[3]{\delta_{in}} \Pr[M].$$

但能够满足上述条件的 v 所占比例至多为

$$\frac{\mathbb{P}_{v\in s'}[(v, s') \text{ is misleading}]}{\frac{7}{2}\sqrt[3]{\delta_{in}}\mathbb{P}[M]} \le 3\delta_{in}^{2/3}$$

根据 Mark ov 的不等式。因此，综上所述，项目 2 发生的概率至多为 $4\sqrt[3]{\delta}$，正如所期望的那样 d. □

电路实现 仔细检查内部解码器后，我们发现它会进行 poly($\frac{1}{\delta_{in}}, \frac{1}{\varepsilon}$) 次调用：

1. 电路 $C_{U;V\to S}$ 在步骤 (a) 中，使用随机索引 i 抽取一个随机 $s' \ni v$。2. 电路 $C_{U;S\cap S\to V}$（通过抽取随机点来评估距离）在步骤 (c) 中运行。该电路还会告知集合 s，s' 是否实际相交。3. 电路 $C_U^{-1}{}_{;\,S\to V}$，以便我们找到查询码字 w_{In_u} 时所用的正确索引。4. 一个解码电路，它将集合的内层表示 s 映射为函数列表电路的输入比特（或在列表译码情形下映射为含噪码字）。

根据断言 5.5，在我们的参数范围内，这些电路的规模均可实现为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，深度均可实现为 $\tilde{O}(\log\log N \exp(O(\frac{1}{\varepsilon})))$。

因此，电路规模仍为 $(\log N)^{\exp(O(\frac{1}{\varepsilon}))}$，电路深度为 $\tilde{O}(\log\log N\exp(O(\frac{1}{\varepsilon})))$。

D.1 列表剪枝

现在，我们将上述普通解码器转化为所需的高置信度解码器。具体来说，通过多次调用内码的生成算法 In_u，我们可以修改内部解码器，使其以概率 $1-\tau$解码一个接近 f的函数，对任意 $\tau>0$均成立（也就是说，只需调用 $\log\frac{1}{\tau}$ 次即可）。这会将列表规模从 $O(\frac{1}{\varepsilon})$ 增加到 $O(\log\frac{1}{\tau}\cdot\frac{1}{\varepsilon})$，看似影响不大，但实际上会破坏我们对“良好分离”列表的归约，因为其参数取决于输入列表大小。因此，为了让内码中的列表规模保持“常数”，我们采用正文中描述的简单电路 C_{Prune}，将大列表剪枝为规模为 $O(\frac{1}{\varepsilon})$ 的小列表，同时确保不会移除近似计算 f 的电路（更准确地说，不会移除任何在 In_u内全局的函数）。为方便起见，我们在此重述该断言，并给出其形式化证明。

断言 D.2 (列表剪枝)。 固定 $\tau>0$，参数为 ε、δ_{out}，输入字为 w_{In_u}，以及 $\{C_1,\dots,C_L\}$：通过运行解码器 $InDec_{\delta_{in}}$ 的生成算法（可能多次）得到的一组输出电路。假设图 (V_u,S_u) 是一个 $\frac{\delta_{out}^2\varepsilon^2}{256}$-采样器。存在一个电路 C_{Prune}，它通过预言机访问 $\{C_i\}$ 并接收输入列表，输出一个规模为 $|L'|\le O(\frac{1}{\varepsilon})$的子集 $L'\subset[L]$。以至少 $1-\tau$ 的概率，任意原始 $C_i\in\mathcal{B}_{1-\varepsilon,\delta_{out}}(\mathcal{L})$ 都与某个 $C_j\in L'$ 是 $6\delta_{out}$-接近的。

此外，C_{Prune} 的对数空间一致规模为 $\mathrm{poly}(|L|,\frac{1}{\varepsilon},\frac{1}{\delta_{out}},\log\frac{1}{\tau})$，深度为 $O(\log^2|L|\cdot\log\frac{1}{\delta_{out}\varepsilon\tau})$，对 $\{C_i\}$ 的预言机调用至多为 $O(\log\frac{L}{\tau}\max\{\varepsilon,\delta\})$ 次；它还接收码字 w、生成随机 $s\in S_u$ 和随机$v\in V_u$的电路，以及超图邻接电路。

我们注意到，后文实际使用的 δ_{out} 是 $O(\delta_{in}^{2/3})$，即我们在前一小节中从内部解码器得到的输出近似参数。

证明。 我们分两步进行。首先，通过经验检验解码出的电路与全局输入列表的距离，移除其中任何与全局输入列表不一致的电路。具体而言，在定义域中抽取 $O(\frac{\log\frac{L}{\tau}}{\varepsilon})$ 条超边，并对每条超边 s 子抽样 $O(\frac{\log\frac{L}{\tau}}{\delta_{out}})$ 个随机 $v\in s$。对于每个电路 C_i，如果它在定义域中超过 $1-\varepsilon/2$ 的比例上与 w_{In_u} 的距离大于 $2\delta_{out}$，我们就将其从列表中移除。根据标准的切尔诺夫界和并集界，剩余的电路集合（记为 L''）以至少 $1-\tau/2$ 的概率满足以下性质：

1. 任何由 w_{In_u} 计算得到的 $(\varepsilon,\ \delta_{out})$- C_i 都保留在列表中
2. 列表中的每个函数都属于 $\mathcal{B}_{1-\varepsilon/4,\ 4\delta_{out}}(w_{In_u})$。

现在根据断言 E.1、第二条性质，以及 $G_{In_u}=(V_u,S_u)$ 是一个 $\frac{\delta_{out}^2\varepsilon^2}{256}$-采样器这一事实，我们可以保证 L'' 中任意一组两两 δ-分离的电路的规模至多为 $\frac{2}{\varepsilon}$。为了找到 L'' 的所需覆盖，考虑如下自然的“经验距离图”。给定函数子集 $L'\subset L''$ 与距离 δ'，(L',δ')-距离图就是以 L' 为顶点集的图，其中 $g,g'\in L'$ 有一条边当且仅当

它们之间的经验距离至多为 δ'（相对于第一步所取的样本进行度量）。注意，以至少 $1-\tau/2$ 的概率，所有经验距离都落在函数之间真实距离的 δ_{out} 范围内。

现在，考虑基于从 V 中抽取的 $O(\frac{\log \frac{L}{\tau}}{\delta_{out}})$ 个采样点所计算的经验距离构造的 $(5\delta_{out},\ L'')$-距离图的一个极大独立集 L'。假设经验估计在上述意义下是良好的，则 L' 是 $\mathcal{B}_{1-\varepsilon/4,\ 4\delta_{out}}(w_{In_u})$ 的 $4\delta_{out}$-分离子集，因此 $|L'| \le \frac{2}{\varepsilon}$。另一方面，每个 $C' \in L''$ 到 L'' 的真实距离都不超过 $6\delta_{out}$。由于 L'' 包含列表中每个原始的 $C_i \in \mathcal{B}_{1-\varepsilon,\ \delta_{in}}$，所以结论成立。

电路复杂度 最后，我们简要说明上述算法可以按所声称的对数空间一致电路规模和深度来实现。预言机调用的次数可直接由描述得出。首先，它并行地对集合进行抽样，然后并行地对其中的点进行抽样。接着，它测试与我们接近的集合所占的比例，并使用阈值函数来决定哪些索引进入 L''。这种电路的规模是 $\mathrm{poly}(|L|, \frac{1}{\varepsilon},\ \frac{1}{\delta_{out}},\ \log \frac{1}{\tau})$，深度为 $O(\log \log_{\frac{1}{\varepsilon}} \frac{L}{\tau})$。

算法的第二步并行调用采样预言机，然后对 L'' 中的每一对函数运行并行两两距离测试。这一步的深度和规模也与前一步相似。

计算 L' 所需的额外步骤是在距离图上计算一个极大独立集；众所周知，这属于 NC^2 [96]。特别地，以上所有过程在采样点的定义域上都属于 NC^2。

□

E 一些近似列表解码引理

在本小节中，我们介绍近似局部列表译码/恢复中用到的若干基本事实。其中大多数至少在某种意义上隐含于先前文献中，或为该领域所熟知；但据我们所知，这些事实尚未见诸文献，因此我们在此列出，以求完整。

E.1 采样器上的组合列表大小

下面给出一个标准论证的变体，该论证表明采样器图上的直积编码具有较小的组合列表大小。

断言 E.1。固定 $\delta_{in}, \varepsilon > 0$, $\ell_{in} \in \mathbb{N}$, $\beta \le \frac{\delta_{in}^2 \varepsilon^2}{4\ell_{in}}$ 和 $\delta \ge \delta_{in}$。设 (V, S) 是一个超图，其包含图是一个 β-采样器。设 $\mathcal{C}_{(V,S)}$ 为相应的直积码，$\mathcal{L} = \{L_s\}$ 为一个基数为 $\le \ell_{in}$ 的列表集合。设 $W \subseteq \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L})$ 是一个极大集合，使得任意 $f_1, f_2 \in W$ 彼此之间的距离都至少为 δ。那么

1. $|W| \le 2^{\frac{\ell_{in}}{\varepsilon}+1}$ 并且
2. 对每个 $f \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L})$，都存在 $f' \in W$，使得 $\mathrm{dist}(f, f') \le \delta$。

断言 *E.1* 的证明。第二项由极大性以及假设 $\delta \ge \delta_{in}$ 自动成立。若存在 $f' \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L})$，其与每个 $f \in W$ 的距离都至少为 δ，那么可以将 f' 加入 W，同时保持两两之间的 δ-距离，这与极大性矛盾。

对于第一项，只需考虑情形 $\delta = \delta_{in}$。反设存在 $2^{\frac{\ell_{in}}{\varepsilon}+1}$ 个互不相同的 δ_{in}-分离函数 $f_1, f_2, \ldots\ldots, f_{2^{\frac{\ell}{\varepsilon}+1}} \in W$。令 $A_i \subseteq S$ 为满足如下条件的 $s \in S$：f_i 与 L_s 中的某个函数 δ_{in}-接近，并且对所有 $j \ne i$，$\mathrm{dist}_s(f_i|_s,\ f_j|_s) > 2\delta_{in}$。由我们的抽样保证，我们有 $\mathbb{P}[A_i] \ge \varepsilon - 2^{\frac{\ell_{in}}{\varepsilon}} \frac{\beta}{\delta_{in}^2} \ge \frac{\varepsilon}{2}$。

现在，我们断言，由平均论证，必定存在某个 s，它至少是这些 A_i 中的 $\ell_{in}+1$，因为

$$\mathbb{E}\left[\sum_{i=1}^{2^{\frac{\ell_{in}}{\varepsilon}+1}} 1_{A_i}\right] \ge \left(2\frac{\ell_{in}}{\varepsilon}+1\right)\cdot\frac{\varepsilon}{2} > \ell_{in}$$

这当然是不可能的，因为对于每个 $s \in A_i$，如果 $\mathrm{dist}(f_i, g) \le \delta_{in}$，那么对于任意 f_j，$\mathrm{dist}(f_j, g) > \delta_{in}$；否则根据不等式 $\mathrm{dist}_s(f_i|_s, f_j|_s) \le 2\delta_{in}$。因此 f_j, f_i 不可能与

输入列表中的相同元素在 $s \in A_i, A_j$ 处很接近；特别地，没有任何 s 能属于超过 ℓ_{in} 个 A_i，由此得到所需的矛盾。

□

E.2 从许多到全部消息长度

在本小节中，我们描述一种简单的重复过程：它取任意一种适用于许多消息大小的近似局部列表可恢复码构造，并将其变换为一种适用于全部消息大小的构造，至多只造成码率损失。

具体而言，假设我们有一种 $(\delta_{in}, \delta_{out})$-近似 ℓ_{in}-输入、$(\varepsilon, \ell_{out}, q)$-近似局部列表可恢复码的构造，它可以在消息大小 $s(k)$ 上实例化，其中 $s(k) \in \mathbb{N}$ 是某个递增的规模函数。我们断言，这立即蕴含一个适用于全部 消息长度的近似局部列表可恢复码族，使得介于 $s(k)$与$s(k+1)$之间的消息长度的码率损失一个因子 $\frac{s(k)}{s(k+1)}$。实现这一点的码族极其简单：只需重复输入消息，直到达到规模 $s(k+1)$（截断最后一次重复）。换言之，最终的码只是截断重复码与原始码的复合。要解码一个消息位，我们只需随机选取其中一次重复，并通过原始码对其进行解码。这意味着所得码保持其局部性与查询复杂度（并且若原始码是直积码，则所得码事实上仍是直积码，尽管可能带有多重集）。

上述内容可由以下简单引理形式化描述，该引理展示了如何将一个码 $C_1 : \Sigma_1^k \to \Sigma_2^n$ 转换为一个码 $C_2 : \Sigma_1^{k'} \to \Sigma_2^n$，其消息比特更少。令 $Rep_{k',k}$ 为上述截断重复码。于是：

引理 E.2。若 $C : \Sigma_1^k \to \Sigma_2^n$ 是一个$(\delta_{in}, \delta_{out})$-近似的、$\ell_{in}$-输入的$(\varepsilon, q, \ell_{out})$-近似局部列表可恢复码，则对任意 $k' \le k$，复合 $C(Rep_{k',k}) : \Sigma_1^{k'} \to \Sigma_2^n$ 是一个$(\delta_{in}, O(\delta_{out}))$-近似的、$\ell_{in}$-输入的$(\varepsilon, O(q)\ O(\ell_{out}))$-近似局部列表可恢复码

证明。令 C' 表示 $C(Rep_{k',k})$。固定一个输入列表 $\mathcal{L}$ 和函数 $f \in \mathcal{B}_{1-\varepsilon,\delta_{in}}(\mathcal{L})$。根据假设，运行原始码 C 会输出一个由（可能经过随机化的）电路组成的列表 $C_1, \ldots, C_{\ell_{out}}$，这些电路的输入为 k 比特，并且其中一个电路 δ_{out} 以至少 $3/4$ 的概率计算出 $Rep_{k',k}(f)$。如第 2 节所述，将这一过程重复 $O(1)$ 次，并将每个 C_i 替换为 $O(1)$-多数版本后，我们可以确保以 99% 的概率，列表中存在一个电路 C_j，使得对于其随机字符串的 99% 种取值，C_j 都能确定性地计算出 $f(\delta_{out})$。

现在，条件化于存在这样的 C_j，以及选择随机字符串 r 使得 C_j^r 确定性地$O(\delta_{out})$-计算 $Rep_{k',k}(f)$。我们现在使用以下基本解码方法：抽样一个随机消息位 $x \in [k']$、一个 $x \in [k]$ 的随机编码位置，并输出该编码位。如果每个消息位都被编码相同次数，那么该方法显然只会以 $O(\delta_{out})$ 的概率失败。然而，由于截断，消息位上的结果分布可能略有偏斜，但即使在最坏情况下，该算法查询任意特定编码位的概率至多是原来的两倍。这意味着误差至多翻倍，因此即使在上述条件化下，最终的失败概率仍至多为 $O(\delta_{out})$，正如所需。

□

F 测试函数之间的距离

我们将经常需要测试定义域 s 上函数 f，g 之间的距离。我们将使用以下基本的查询高效过程：以下断言由标准的切尔诺夫界推出，其证明从略。

断言 F.1. 设 $\alpha, \gamma, \delta > 0$，且 $p \ge \frac{5}{\alpha} \log \frac{1}{\gamma}$。那么，该 (p, δ)-距离测试器具有以下保证。

1. 如果 $\mathrm{dist}(f, g) \le \delta - \alpha$，那么 $\mathbb{P}_{v_1, v_2, \ldots, v_p}[(p, \delta)\text{-距离}(f,g)\text{=为真}] \ge 1 - \gamma$。
2. 如果 $\mathrm{dist}(f, g) \ge \delta + \alpha$，那么 $\mathbb{P}_{v_1, v_2, \ldots, v_p}[(p, \delta)\text{-距离}(f,g)\text{=为假}] \ge 1 - \gamma$。

至于电路复杂度，我们有如下保证：

$(p,\ \delta)$-距离测试器：

* **输入：** 函数 $f,\ g:\ s \to \Sigma$。

* **输出：** 真/假。

1. 抽样 p i.i.d 随机点 $v_1,\ v_2$、...、$v_p \in V$。2. 如果 $\frac{|i:\ f(v_i)=g(v_i)|}{p} \leq \delta$ 输出真，否则输出假。

图13：距离测试器

断言F.2。假设可以对 $f, g:\ s \to \Sigma$进行预言机访问，并且可以对一个在 s 中输出随机点的电路进行预言机访问，则可以使用 $O(\log p)$-深度和 $O(p\log|V|)$-规模实现一个电路 (p, δ)-距离测试器。它对 f, g 进行$2p$次调用。

证明。我们抽样所有 p 个点。对于每个点，我们分别通过一个常数规模电路[67]测试等式 $f(v) = g(v)$。注意，规模为 $p\log|V|$，主要是因为每个点 v 的标识符规模为 $\log|V|$。此后，我们在 p个点上实现一个 δp-阈值电路，根据标准计算，其规模为 $O(p)$，深度为 $O(\log p)$。 □

最后，下面这个直接的推广在后文中是必要的。设 L_1, L_2 为同一定义域 s 上的两个函数列表。下面的算法告诉我们哪些函数 $g \in L_2$ 接近 L_1 中的某个函数。

(p, δ)-距离测试器：

* **输入：** 集合 s 上的函数列表 $L_1 = \{f_i\}$ 和 $L_2 = \{g_i\}$。

* **输出：** 函数列表 $L_{out} \subset L_2$1. 抽样 p 个独立同分布随机点 $v_1, v_2, \ldots, v_p \in V$。2. 令 $L_{out} \leftarrow \emptyset$3. 对于每个 $g_i \in L_2$4. 返回 L_{out}

- If $\exists f_j \in L_1$ s.t. $\frac{|i:f(v_i)=g(v_i)|}{d} \leq 2\delta$, $L_{out} \leftarrow L_{out} \cup g_i$

图14：距离测试器

下面的断言可由断言 F.1和断言F.2直接得出。

断言F.3. 设 $\delta > 0$ 和 $\gamma > 0$，并设 $p \geq \frac{12\log\frac{|L_1||L_2|}{\gamma}}{\delta}$。图*14*中的算法具有如下保证。

1. 若 $\mathrm{dist}(g, L_1) \leq \delta$，则 $\mathbb{P}_{v_1,\ v_2\ldots,\ v_p}[g \in L_{out}] \geq 1-\gamma$。*2.* 若 $\mathrm{dist}(g, L_1) \geq \delta + \alpha$，则 $\mathbb{P}_{v_1,\ v_2\ldots,\ v_p}[g \notin L_{out}] \geq 1-\gamma$。

图*14*可用对数空间一致方式实现，其深度为 $O(\log p + \log|L_1||L_2|)$，规模为 $O(p(|L_1||L_2| + (|L_1| + |L_2|)\log|V|))$，并需要调用这些函数 $p(|L_1| + |L_2|)$ 次。

[67]严格来说，我们假设字母表是常数。若并非如此，则深度还需增加一个 $\log|\Sigma|$ 的加法因子，规模也会乘以 $\log|\Sigma|$。