arXiv:2603.20127v1 [cs.PL] 20 Mar 2026

量子纠错解码器分析

ABTIN MOLAVI,威斯康星大学麦迪逊分校，美国FERAS SAAD,卡内基梅隆大学，美国AWS ALBARGHOUTHI,威斯康星大学麦迪逊分校，美国

量子纠错（qec）通过将逻辑信息编码到多个物理量子比特中，并定期测量奇偶校验来检测错误，从而使噪声硬件上的计算可靠进行。解码器 是一种经典算法，它利用这些测量结果推断最可能发生了哪种错误，以便系统进行纠正。解码器的准确性——即它做出错误猜测的频率有多低——直接决定了能够可靠执行的量子计算规模。面对众多相互竞争的解码算法，量子纠错系统设计者需要可靠的方法来评估它们。如今，主流方法是使用蒙特卡罗模拟评估解码器。然而，模拟存在若干缺点，例如需要大量样本才能得到低方差估计。

在这项工作中，我们开发了一种用于评估解码器的新型系统分析方法。我们为量子纠错程序的核心语言引入了一种新颖的形式语义，它涵盖了事实上的标准 Stim 量子电路格式，为新兴的容错量子系统设计领域提供了严谨的理论基础。给定一个量子纠错程序和一个解码器，我们可以量化解码器的准确性 以及解码器对物理错误率漂移的鲁棒性 。我们的方法包含两个关键组成部分：（i）对所有可能错误构成的空间进行结构化搜索；（ii）一个约束多项式优化内核。全面的实证评估表明，我们的方法能够胜过模拟，尤其是在低错误率范围内；此外，它还可用于量化解码器在一定物理错误率区间内的鲁棒性。

1 引言

量子计算有望在重要领域超越经典方法，为材料科学、化学、机器学习及其他领域带来突破。然而，这些实际应用通常需要数十亿甚至数万亿次精确操作，而硬件量子比特（qubits）十分脆弱且容易出错。幸运的是，我们可以通过量子纠错（qec）弥合这一差距。

量子纠错通过冗余实现容错，使用多个物理量子比特对每个逻辑量子比特的状态进行编码。通过重复进行奇偶校验测量，可以检测错误而不破坏逻辑态。在没有错误的情况下，每个周期都会产生相同的测量结果。如果某个物理量子比特在两个测量周期之间受到错误影响，则一个或多个测量结果会发生翻转（从 0 变为 1，或从 1 变为 0）。根据测量结果的综合征精确推断发生了哪种错误，这项计算任务称为解码。由于每个综合征都可能对应多种错误模式，解码并非完美过程。相反，解码算法会进行概率推断，力求找出能够解释该综合征的最可能的错误。

当解码失败，且推断出的错误与实际错误不等价时，就会产生逻辑错误。逻辑错误率，也称解码器准确性，决定了纠错量子计算机的可靠性，以及能够准确执行的计算规模。

量子纠错系统依赖高效、准确的解码器来实现可接受的逻辑错误率，因此大量研究工作致力于开发新颖的解码算法。已应用于解码问题的策略多种多样，包括基于 Edmonds 经典花朵收缩算法的方法 [24],贝叶斯网络中的置信传播[25, 35, 36], MaxSAT [4], 以及神经网络 [3, 39]。

作者联系方式：Abtin Molavi，威斯康星大学麦迪逊分校，美国，amolavi@wisc.edu；Feras Saad，卡内基梅隆大学，美国，fsaad@cmu.edu；Aws Albarghouthi，威斯康星大学麦迪逊分校，美国，aws@cs. wisc.edu。

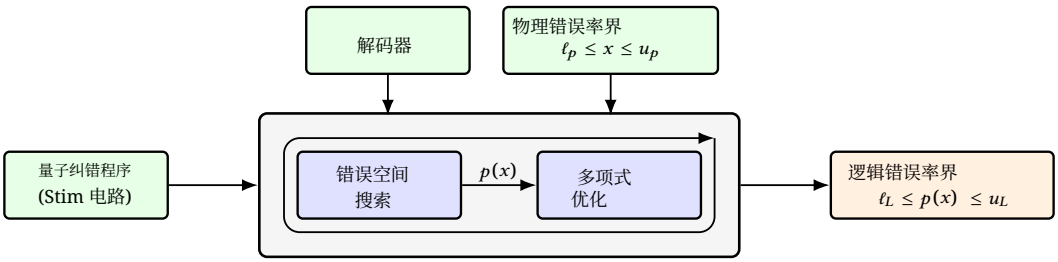


图 1. 我们方法的概览。

为了根据这些选项做出明智的决策，量子纠错系统设计者需要评估不同的解码器，判断解码器部署后预期能够达到的逻辑错误率。**我们的工作旨在提高解码器评估的可靠性**。

通常，通过模拟来估计解码器准确性。一个量子纠错 协议会依据定义的错误模型，以概率方式注入错误进行模拟。收集大量模拟采样轮次后，可以通过计算发生逻辑错误的采样轮次所占比例，经验性地估计解码器准确性。Stim 模拟框架 [17],是一个体现这种解码器评估测试方法的例子。它是一款流行的库，支持高性能并行模拟，并包含简单的错误模型。这种测试策略存在一些重要缺点。

- **极其稀有的事件**。随着设备性能提升，简单采样的效果会越来越差，因为所需样本数量与逻辑错误率成反比。近期的一项估算表明，使用 Shor 算法分解一个 2048 位 RSA 整数，需要逻辑错误率达到 $\approx 10^{-15}$ [18]。通过模拟来可靠估计如此小的数值，需要多到难以承受的采样轮次。
- **建模中的不确定性**。量子计算机所受噪声的模型存在不确定性。校准实验观察到错误率会随时间漂移，也会出现暂时导致错误率激增的突发事件 [7, 21, 33]。在模拟中采用具体数值，无法体现解码器可能需要应对的各种合理运行条件。

枚举式解码器分析。为了解决这些不足，我们提出了一种新的解码器评估方法，如图 1 所示。与重要性抽样类似，我们将事件的概率与其在搜索中的优先级分离。我们系统地枚举错误空间中的事件，每个事件最多访问一次，并通过解析方法计算其发生概率。采用这种方法，我们可以避免将采样轮次浪费在高概率事件上的问题。例如，在现实的物理错误率下，“无错误”事件很可能主导采样结果。对于任何合理的解码器设计，如果没有物理错误，就不会出现逻辑错误，因此这些采样结果几乎无法帮助我们了解不同解码策略之间的差异。对于搜索空间过大、无法通过枚举有效探索的情况，我们还提出了一种混合方法，利用有限的随机采样来估计尚未探索的空间对逻辑错误率的贡献。

量化解码器鲁棒性。我们的方法还能够分析解码器的鲁棒性：即逻辑错误率对物理错误率扰动的敏感性。关键洞见在于，逻辑错误率可以以符号形式表示为一个关于表示物理错误的变量的多项式。计算鲁棒性可归结为对误差多项式进行约束优化。对于每个

物理错误率均被约束在某个区间内的情形，我们能够精确求解这一约束优化问题。实际求解借助一种我们称为偏导数剪枝的技术，它能够在实际问题实例中显著缩小搜索空间。偏导数剪枝利用过近似来分析整个搜索空间内相对于特定变量的偏导数符号。如果可以确定该符号，那么我们便可以选择将该变量固定在某个边界上，从而将搜索空间减半。

一种形式化定义的输入语言。 为定义解码器准确率问题的搜索空间，我们提出量子纠错程序这一概念，用于表示在给定噪声模型下执行量子纠错协议的过程。我们的量子纠错程序抽象对 Stim 仿真框架的输入——Stim 电路语言的核心部分进行了形式化。Stim 电路正逐渐成为容错量子计算高级描述的一种主流编译目标。我们通过给出首个形式语义，为容错量子软件栈奠定了坚实基础。我们还引入了一种扩展，用于定义称为符号量子纠错程序的鲁棒性问题。每个符号量子纠错程序都表示在一族参数化错误模型下执行量子纠错协议，以验证解码器对抗这些模型时的鲁棒性。

评估。 我们通过估算三种最先进解码器在一类主流量子纠错码（旋转表面码）[16]上的准确性和鲁棒性，评估了我们的方法。对于准确性问题，我们将其与 Stim 仿真进行比较，发现我们的方法能够得出更精确的估算，尤其是在低错误率区间。例如，对于一个距离为 5 的表面码问题实例，我们的方法能够以约$3 \cdot 10^6$ 个采样轮次，在 0.1% 的误差范围内确定解码器的逻辑错误率；而要以一个数量级的置信度估算该值，采样所需的轮次是其 100 倍。

贡献。 总之，我们的贡献如下：

- 基于 Stim 电路格式，为量子纠错程序定义了形式语法和语义
- 定义了量子纠错程序的解码器逻辑错误率估计问题和鲁棒性问题，并通过多项式优化对其进行了刻画
- 提出了一种算法，通过系统枚举错误空间来求解逻辑错误率估计问题和鲁棒性问题，并扩展该算法，将有限的采样与枚举相结合
- 提出了一种针对超矩形进行多项式优化的流程，使得我们的算法能够实际应用于一类自然的鲁棒性问题
- 通过实证评估，证明了与基于仿真的方法相比，我们的算法具有良好的效果

2 量子计算与量子纠错基础

本节将概述量子计算和量子纠错所需的基础知识。

量子比特。 量子计算的基本单位称为量子比特，或称为 量子比特。量子比特有两种可能的状态：$|0\rangle$ 和 $|1\rangle$，它们类似于经典比特的 0 和 1。与经典比特不同，这两种状态的任意复线性组合 $\alpha|0\rangle + \beta|1\rangle$也是一种可能的量子比特态，但须满足条件 $|\alpha|^2 + |\beta|^2 = 1$。换句话说，量子比特的状态是一个二维向量空间中的单位向量，该空间定义在 $\mathbb{C}$上。对处于状态 $\alpha|0\rangle + \beta|1\rangle$ 的量子比特进行计算基测量 ，将以概率 $|\alpha|^2$ 得到状态 $|0\rangle$，并以概率$|\beta|^2$得到状态 $|1\rangle$。多量子比特系统是对此的直接扩展。例如，双量子比特

系统是一个四维向量空间中的元素，其基为 $\{|00\rangle\ |01\rangle\ |10\rangle\ |11\rangle\}$。一般而言， n 个量子比特的状态是一个 2^n 维向量空间中的向量。

密度矩阵。 密度矩阵是一种用于表示量子态概率系综的结构。这类系综可能由对多量子比特系统中的一个量子比特进行测量而引入，而测量结果未知。对于我们目前所见的纯量子态 $|\psi\rangle$，密度矩阵 ρ由 $\rho = |\psi\rangle\langle\psi|$给出，其中 $\langle\psi| = |\psi\rangle^\dagger$表示共轭转置。对于量子态系综 $|\psi_i\rangle$，其中每个量子态的概率为 p_i， $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$.

量子门与电路。 量子比特态通过称为量子门的操作进行转换。量子门是保持范数的状态线性映射。也就是说，量子门可以表示为酉矩阵。一个例子是 X门，它将经典电路中的非门推广为量子门，将 $|0\rangle$映射为 $|1\rangle$，反之亦然。另一个例子是 Z门，它施加相对相位，将$\alpha|0\rangle + \beta|1\rangle$映射为 $\alpha|0\rangle - \beta|1\rangle$。 X门和 Z门属于重要的单量子比特量子门类别，称为*Pauli*门，其中包括 $Y = iZX$作为最后一个成员。

量子门还可以作用于多个量子比特。常见的双量子比特量子门是受控非门。受控非门因其对计算基态的作用而被称为“受控-X”。如果第一个参数为 $|1\rangle$，受控非门就会对第二个参数应用 X。如果第一个参数处于 $|0\rangle$态，则该量子门不会产生任何作用。另一个与此相关的例子是多量子比特泡利算符。一个 n量子比特泡利算符是 n 个单量子比特量子门的张量积，这些量子门要么是泡利门，要么是恒等算子。Quantum量子电路由量子门和测量操作组合而成。

量子纠错。与经典纠错码非常相似，量子纠错码通过冗余编码来降低错误率。单个逻辑量子比特的状态被编码到多个物理量子比特中，并利用对辅助量子比特的测量来检测和纠正错误，同时不会使逻辑态坍缩。

最简单的例子是三量子比特重复码，它将状态 $\alpha|0\rangle + \beta|1\rangle$编码为 $\alpha|000\rangle + \beta|111\rangle$。在该编码中，我们利用对前两个和后两个量子比特进行经典奇偶校验测量的量子类比来检测错误。在量子纠错中，这类奇偶校验测量称为稳定子测量。由于只有量子比特全为0或全为1的基态具有非平凡系数，因此在没有错误的情况下，两次稳定子测量的结果以概率1为0。两次测量的结果若不是00，则表明某个量子比特发生了错误翻转；我们可以将其建模为一次非预期的 X 量子门。稳定子测量的结果称为综合征，因为它们用于推断错误的存在。量子纠错码中的另一种关键奇偶校验测量是逻辑可观测量，即对逻辑量子比特状态执行的测量操作。在三量子比特重复码中，对逻辑态进行计算基测量等价于对任意物理态进行计算基测量（因为测量后所有量子比特都会坍缩到同一个基态），因此我们可以将任意单量子比特测量定义为逻辑可观测量。

三量子比特重复码可以防止比特翻转（错误的 X 量子门），但量子比特也可能受到相位翻转（即错误的 Z 量子门）的影响。完整的量子纠错码还会包含稳定子测量，以检测相位翻转。

3 准确性与鲁棒性问题：概述

本节将确立本文的问题设定，最终给出我们所探讨的两个解码器评估问题的定义：准确性（定义 3.1）和鲁棒性（定义 3.2）。

量子纠错程序。我们首先介绍量子纠错程序，它表示在特定噪声模型下执行纠错协议。量子纠错程序是量子电路的一种简单扩展

```
1  // 应用      比特翻转   噪声    to  量子比特（概率为   0.01）
2  Xerr(0.01)   q[0];
3  Xerr(0.01)   q[2];
4  Xerr(0.01)   q[4];
5
6  // 执行          奇偶校验   校验    测量
7  cx q[0] q[1];
8  cx q[2] q[3];
9  cx q[2] q[1];
10 受控非门 q[4] q[3];
11 测量 m[0]    ← q[1];
12 测量 m[1]    ← q[3];
13 声明综合征 m[0];
14 声明综合征 m[1];
15
16 // 可观测量测量，                    坍缩态
17 测量 m[2]    ← q[0];
18 测量 m[3]    ← q[2];
19 测量 m[4]    ← q[4];
20 声明-可观测量 m[2];
```

（a）量子纠错程序

错误	综合征	可观测量	概率
000	00	0	$(0.99)^3$
111	00	1	$(0.01)^3$
001	01	0	$(0.99)^2(0.01)$
110	01	1	$(0.01)^2(0.99)$
011	10	0	$(0.99)(0.01)^2$
100	10	1	$(0.01)(0.99)^2$
010	11	0	(0。99)(0。01)(0。99)
101	11	1	(0.01)(0.99)(0.01)

(b) 诱导概率分布

图 2. 用于三量子比特重复码存储器实验的量子纠错程序及其诱导概率分布。

，包含概率噪声操作，以及用于标注逻辑可观测量和稳定子测量的注释。图 2a 展示了一个简单的示例。在此程序中，我们在三量子比特重复码中制备逻辑 $|0\rangle$ 态，执行一轮综合征测量，然后在计算基中测量逻辑量子比特。这种先制备基态、执行一轮或多轮纠错，再在该基中进行测量的模式称为存储实验。存储实验是用于验证量子纠错程序抑制错误效果的基础测试。

在示例程序中，我们可以看到量子纠错程序中的四种语句类型。

- **标准电路操作**：量子门（例如第 7–10 行）和测量（例如第 17–19 行）
- **误差通道**：第 2–4 行以 0.01 的概率应用 X 量子门，对应 1% 的比特翻转错误率。
- **综合征声明**：第 13 行和第 14 行定义了综合征比特。这是解码器的输入信息。一般而言，综合征比特定义为一组测量结果的奇偶校验；此处则是对单元素集合进行的平凡奇偶校验。
- **可观测量声明**：第 20 行声明了一个逻辑可观测量，即我们可以对逻辑量子比特的状态执行的测量。解码器的输出是对该比特值的预测。与综合征声明一样，此处的表达式可以是一个或多个经典数据比特的奇偶校验。

量子纠错程序的语义是对所声明的综合征比特和可观测量比特的取值所构成的联合分布。例如，该程序定义的分布如图 2b 所示。在错误列中，我们将施加的噪声操作表示为一个错误比特串。索引 i 处的“1”表示施加了第 i 个概率噪声操作。请注意，综合征和可观测量的结果都是错误的确定性函数。也就是说，量子纠错程序的无错误执行会产生确定性的测量结果，而概率性结果完全由概率噪声操作导致。

可观测量与解码。 在我们的示例程序中，我们（任意地）选择量子比特 0 来定义逻辑可观测量。当量子比特 0 受到 X 错误影响时，结果为状态 $\alpha|100\ +\beta|011$。

我们所选可观测量的值发生了翻转，但只要我们能够检测到发生了翻转，就能恢复所需的测量算符。我们只需对测量结果进行相应的修正，将结果“1”解释为“0”，反之亦然。

一个 解码器 利用综合征测量来预测何时发生了会翻转逻辑可观测量的错误。根据图 2b中各综合征-可观测量对的明确概率，我们可以使用以下简单的解码算法：对于每个综合征，计算两种可观测量结果的概率，并预测概率较高的结果。例如，给定综合征“00”，我们预测“0”，因为 $(0.99)^3 > (0.01)^3$.

解码器给出错误预测称为逻辑错误，而解码的目标是尽量降低逻辑错误率。上一段所述的策略称为最大似然解码，是降低逻辑错误率的最优方法。遗憾的是，对于大规模程序而言，最大似然解码难以执行；对于给定综合征，计算最大似然可观测量结果这一一般问题是 #P-困难 [14, 27]。因此，在实际应用中，解码器采用次优算法，以达到可接受的逻辑错误率。为了评估解码器，我们需要估算其逻辑错误率。

定义 *3.1*（准确性问题）。给定一个包含 n 个综合征比特和 m 个逻辑可观测量的量子纠错程序 P，以及一个解码器 $d\colon \{0,1\}^n \to \{0,1\}^m$，求其逻辑错误率：

$$\mathrm{Err}(P, d) \coloneqq \Pr_{(s,o)\sim P}[d(s) \neq o].$$

鲁棒性。图 2a 中的程序将比特翻转概率硬编码为 0.01，但要精确地对量子器件建模颇具挑战，而且错误率往往会随时间漂移 [7, 21]。为描述这种不确定性，我们可以定义该程序的符号版本，用符号变量替换三个具体概率 x_1，x_2和 x_3。为了表示有效概率，我们要求 $0 \le x_i \le 1$，但通常我们关注的是在窄得多的取值范围内的最坏情况性能。如果先验模型估计比特翻转率为 0.01，我们可以在约束集 $0.009 \le x_i \le 0.011$ 上进行评估，以此对物理错误率建模，并考虑 10% 的不确定性。通过求出符号变量取具体值时可能达到的最大逻辑错误率，我们可以量化解码器在约束集上的鲁棒性。

定义 *3.2*（鲁棒性问题）。给定符号量子纠错程序 $P(x)$、解码器 d 和约束集 R，求最坏情况下的逻辑错误率：

$$\max_{\overline{v}\in R} (\mathrm{Err}(P(\overline{v}), d)).$$

鲁棒性问题是准确性问题的一种推广；当约束集为平凡集合时，准确性问题等同于鲁棒性问题，其中 $|R| = 1$。在第 6.1 节中，我们将介绍如何在非平凡约束集上求解鲁棒性问题。具体而言，我们关注超矩形，它涵盖了上述情形，即每个物理错误率都受限于一个独立定义的区间。

由于可能的综合征数量会随逻辑量子比特的规模呈指数增长，因此无法精确求解准确性与鲁棒性问题。我们着重计算可靠界，以保证所关注的量落在某个区间内。

4 qecPrograms 语言

在本节中，我们将形式化 qec 程序的语言，这些程序表示纠错协议的含噪执行。我们的 qec 程序语言以 Stim [17] 量子线路模拟器为基础，对应于 Stim 电路语言的一个核心子集。

$$P := S; DS := \textbf{跳过} \mid U(q) \mid \textbf{复位}\ \ q \mid \textbf{测量}\ \ m \leftarrow q \mid \textbf{错误}(q)$$
$$\mid S_1;\ S_2 D := \textbf{声明综合征}(m) \mid \textbf{声明可观测量}(m) \mid \overrightarrow{D_1}; D_2 Err :=$$
$$\textbf{退极化1}(p) \mid \textbf{退极化2}(p) \uparrow \textbf{X错误}(p) \mid \textbf{Z错误}(p) \mid \cdot \cdot \overrightarrow{\cdot} p \in [0,1]$$

图 3. 量子纠错程序的语法

4.1 语法

一个 量子纠错 程序是由图中语法生成的直线式语句序列。3。语句可以作用于经典和量子寄存器变量。术语 q 表示一个量子寄存器地址，而 q 表示由此类地址组成的有序列表。类似地， m 表示一个经典寄存器地址，而 m 表示一个有序列表。与传统的量子电路语言（例如 OPENQASM2）类似，可能的语句包括应用量子门、量子比特初始化和测量。在语句 $U(q)$ 中， U 是一个 $2^{|q|}$ 维酉算子。语句**复位** q 将寄存器 q 设为 $|0\rangle$，而**测量** $m \leftarrow q$ 则在计算基下测量 q，并将所得比特存储在经典寄存器地址 m 中。

量子纠错程序还包括专用于纠错的语句。这类语句的第一种是错误信道。错误信道通过从一组可能的噪声算子（其中包括恒等算子 I）中按概率选择一个量子门来施加噪声。通常，错误信道由噪声强度 p 参数化。最简单的错误信道是 Xerr(p) 和 Zerr(p) 信道，它们以概率 p 施加相应的量子门，否则以概率 I 施加其他操作。除了这些基本错误信道外，错误模型通常还包括退极化信道。强度为 p 的单量子比特退极化信道 Depolarize1(p)，以概率 p 从集合 $\{X, Y, Z\}$ 中选择一个量子门（均匀选择，因此每个量子门的概率为 $p/3$），以概率 $(1-p)$ 施加恒等算子。双量子比特退极化信道 Depolarize2(p) 以概率 p 从 15 个非恒等的双量子比特泡利算子中均匀选择一个，并以概率 $(1-p)$ 施加恒等算子。

第二种纠错语句是综合征和可观测量声明，它们定义了解码问题的参数。综合征声明将错误综合征中的一个比特定义为输入经典寄存器列表的奇偶校验，而可观测量声明则以相同方式定义逻辑可观测量。求值量子纠错程序相当于以概率方式将这些声明解析为具体比特。我们不失一般性地假设测量语句采用静态单赋值形式，因此可以将综合征和可观测量声明移至程序末尾。

4.2 语义

我们通过整合先前量子 [41] 和概率 [28, 38]程序定义中的思想，为不含综合征或可观测量声明的量子纠错程序前缀赋予小步语义。具体而言，我们定义程序配置 S， Σ， ρ， Γ，其中

- S 是一个量子纠错程序。
- Σ 是一个随机抽取序列。我们将概率误差通道建模为以随机抽取参数 σ 为自变量的函数。参数 σ 类似于从有限集合中均匀选取的伪随机种子。量子纠错程序中施加的噪声由抽样序列 Σ 决定。
- ρ 是一个密度矩阵，表示从有限集合 $QVar$ 中选取的整个量子变量寄存器当前的量子态。

$$[\textsc{SkipSeq}]\ \frac{}{\langle \textbf{skip}; S, \Sigma, \rho, \Gamma\rangle \to \langle S, \Sigma, \rho, \Gamma\rangle} \qquad [\textsc{Unitary}]\ \frac{\dim(U) = 2^{|\overline{q}|} \qquad \overline{q} \subseteq \mathrm{dom}(\rho)}{\langle U(\overline{q}), \Sigma, \rho, \Gamma\rangle \to \langle \textbf{skip}, \Sigma, U_{\overline{q}}\rho U_{\overline{q}}^\dagger, \Gamma\rangle}$$

$$[\textsc{Reset}]\ \frac{q \in \mathrm{dom}(\rho)}{\langle \textbf{reset}\ q, \Sigma, \rho, \Gamma\rangle \to \langle \textbf{skip}, \Sigma, |0\rangle\langle 0|_q\,\rho\,|0\rangle\langle 0|_q + |0\rangle\langle 1|_q\,\rho\,|1\rangle\langle 0|_q, \Gamma\rangle}$$

$$[\textsc{Measure}]\ \frac{q \in \mathrm{dom}(\rho) \qquad \mathrm{Tr}(P_b\rho) > 0}{\langle \textbf{measure}\ m \leftarrow q, \Sigma, \rho, \Gamma\rangle \to \langle \textbf{skip}, \Sigma, \frac{P_b\rho P_b}{\mathrm{Tr}(P_b\rho)}, \Gamma[m \mapsto b]\rangle}$$

$$[\textsc{Err}]\ \frac{\Sigma = \sigma : \Sigma' \qquad U = Err(\sigma)}{\langle Err(\overline{q}), \Sigma, \rho, \Gamma\rangle \to \langle \textbf{skip}, \Sigma', U_{\overline{q}}\rho U_{\overline{q}}^\dagger, \Gamma\rangle}$$

图 4. 不含综合征和可观测量声明的量子纠错程序前缀的小步语义

- $\Gamma : CV \xrightarrow{\text{算术}} \{0, 1\}$ 是从有限集合 $CVar$ 中抽取的经典寄存器变量的状态

我们定义程序配置之间的迁移关系：

$$\langle S, \Sigma, \rho, \Gamma\rangle \to \langle S', \Sigma', \rho', \Gamma\rangle$$

表示执行的一步。定义该关系的推理规则如图 4 所示。重置规则和酉规则说明这些操作会对底层量子态产生相应的作用。酉规则中的记号 U_q 指的是将 U 提升到全局状态，对相关量子比特应用 U，并在其他量子比特上作用为恒等算子。[1] 重置规则中的投影算子 $|0\,0|_q$ 是类似的提升。测量规则根据测量结果更新量子态，并将测量结果存储在经典存储 Γ 中。在此规则中，P_b 是投影到基态上的投影算子之一：对于 $i = 0$，它是 $|i\,i|_q$；对于 $i = 1$，则是另一个投影算子。请注意，该规则允许非确定性迁移到由任意可能测量结果产生的状态。我们将在讨论良定义程序时再回到这一点。最后，Err规则说明误差通道如何消耗一次随机抽样来选择要应用的酉算子。选择函数 $\mathrm{Err}(\sigma)$ 编码了不同酉算子的相对概率。例如，语句 $\mathrm{Xerr}(0.01)$ 可由函数 $\mathrm{Err} : \{1, \ldots, 100\} \to \{I, X\}$ 建模，其中 $\mathrm{Err}(1) = X$，而 $\mathrm{Err}(x) = I$。

综合征和可观测量声明表示经典寄存器上的简单函数，这些函数对其输入比特取奇偶校验，并返回一个比特。这些语句的组合将结果串接成一对比特串：综合征比特串和可观测量比特串。因此，我们定义指称语义 $[\![\cdot]\!]_D$：$D \times (CVar \to \{0,1\}) \to \{0,1\}^* \times \{0,1\}^*$ on D，即声明集合，如图 5 所示。

完整程序语义。 执行一个 量子纠错程序的结果是一对比特串，分别表示对程序体求值后得到的综合征和可观测量。我们可以将其表示为一个大步判断（图 6），它结合了综合征和可观测量声明的语义，以及图 4中其他语句的转换规则。这里，$\to^*$ 是 $\to$的传递闭包。

[1]我们可以 将 U_q明确写为 $\Pi_q(U \otimes I_{\mathrm{dom}(\rho)\setminus q})\Pi_q^{-1}$，其中 Π_q 是一个置换，可将 q 移至最前面的位置

$$[\![\textbf{declare-syndrome}(\overline{m})]\!]_D(\Gamma) = \left(\bigoplus_{m\in\overline{m}} \Gamma(m), \varepsilon\right)$$

$$[\![\textbf{declare-observable}(\overline{m})]\!]_D(\Gamma) = \left(\varepsilon, \bigoplus_{m\in\overline{m}} \Gamma(m)\right)$$

$$[\![D_1; D_2]\!]_D(\Gamma) = (s_1 \cdot s_2, o_1 \cdot o_2) \quad \text{where } (s_i, o_i) \coloneqq [\![D_i]\!]_D(\Gamma)$$

图 5. 声明的指称语义。“·”运算符表示字符串连接

$$[\textsc{FullProg}]\ \frac{\langle S, \Sigma, \rho, \Gamma\rangle \to^* \langle \textbf{skip}, \Sigma', \rho', \Gamma'\rangle \qquad [\![D]\!]_D(\Gamma') = (s, o)}{\langle S; D, \Sigma, \rho, \Gamma\rangle \Downarrow (s, o)}$$

图 6. 完整量子纠错程序的语义

良定义的综合征和可观测量。 非确定性测量意味着，我们可以编写一个 量子纠错 程序，使得FullProg 允许求值产生多个不同的综合征-可观测量对。然而，对于对应于正确构造的 量子纠错 代码和实验的程序，不会出现这种行为。我们称只有一种可能的最终求值结果的程序为良定义程序（参见定义 4.1），并将研究范围限定为良定义程序。可以使用一种称为稳定子表模拟的技术，在静态阶段强制执行对良定义程序的限制 [1, 22]。

定义 *4.1.* 如果对于每个配置（Σ, ρ, Γ），都存在唯一的（s, o），使得 $P, \Sigma, \rho, \Gamma \Downarrow$（$s, o$），则量子纠错程序 P 定义良好。

错误比特串：识别执行过程。为了系统地评估解码器，我们关注如何隔离量子纠错程序的特定执行路径。量子纠错程序。对于定义良好的程序，这相当于记录误差通道实际应用了哪些量子门。为此，我们定义错误比特串的概念。为简单起见，我们假设所有误差通道都是伯努利信道。也就是说，它们从集合$\{U, I\}$中取值，该集合仅包含一个以某种概率 p 应用的非恒等门（去极化信道可以分解为多个伯努利信道）。因此，一个执行过程由比特串 e 定义，其中 $e_i = 1$ 表示在第 i个误差通道语句处应用非恒等门，而 $e_i = 0$ 表示应用恒等算子。图2b 左栏展示了示例程序的所有错误比特串。

概率语义。 现在，我们将 量子纠错 程序的形式语义与第 3 节中的非形式化概念联系起来：一个量子纠错 程序定义了一个关于综合征-可观测量对的分布。作为过渡，我们定义程序关于错误比特串的分布。设 P 是一个 量子纠错 程序，其错误信道语句为 E_1、……、 E_n。那么我们用 e_Σ 表示由抽样序列 $\Sigma = \sigma_1$、……、 σ_n. 得到的比特串。也就是说， $e_\Sigma = e_1$、……、 e_n 由以下规则定义

$$e_i = \begin{cases} 0 & \text{if } E_i(\sigma_i) = I \\ 1 & \text{otherwise} \end{cases}$$

我们称 Σ 实现 e 如果 $e_\Sigma = e$。使用这种记法，我们可以表示由量子纠错程序 P 和随机抽样序列空间 D 所定义的比特串概率分布。一个比特串的概率就是能够实现它的抽样序列所占的比例：

```
    1 Xerr(x1) q[0]; 2
Xerr(x2) q[2]; 3 Xerr(x3) q
[4]; 4 受控非门 q[0] q[1]; 5
受控非门 q[2] q[3]; 6 受控非
门 q[2] q[1]; 7 受控非门 q
[4] q[3]; 8 测量m[0] ←
q[1]; 9 测量m[1] ← q[3];
10 声明综合征 m[0]; 11 声明综
合征 m[1]; 12 测量m[2] ←
q[0]; 13 测量m[3] ← q[2];
14 测量m[4] ← q[4]; 15 声
明可观测量 m[2];
```

图 7. 三量子比特重复码存储器实验的符号化量子纠错电路。

$$f(e; P, \mathcal{D}) = \frac{|\{\Sigma \in \mathcal{D} : e_\Sigma = e\}|}{|\mathcal{D}|}.$$

由于综合征和可观测量的结果是错误比特串的确定性函数，我们可以将错误比特串上的分布扩展为综合征-可观测量对上的分布。我们首先利用程序语义，显式定义从比特串到综合征-可观测量对的函数。给定一个错误比特串 e，$\mathrm{synd}(e)$ 和 $\mathrm{obs}(e)$ 表示相应执行所产生的综合征和可观测量对，具体形式化定义见定义4.2。

定义 *4.2*。令 P 为包含 n 伯努利错误信道语句的量子纠错程序，令 e 为由抽样序列 Σ_e实现的错误比特串。我们将 $\mathrm{synd}(e) = s$ 和 $\mathrm{obs}(e) = o$ 分别记作满足以下条件的综合征和可观测量：P，Σ_e，$|0\,0|^{\otimes n}$，$\varnothing \Downarrow (s,\ o)$。

我们有一个关于综合征—可观测量对的自然概率分布，即产生该对的比特串的概率：

$$g((s,o); P, \mathcal{D}) = \sum_{e \in B} \chi_{s,o}(e; P) f(e; P, \mathcal{D}),$$

其中，B是 P的错误比特串集合，而$\chi_{s,\ o}(e; P)$是指示函数：当 $\mathrm{synd}(e) = s$ 且 $\mathrm{obs}(e) = o$ 时，其值为 1，否则为 0。

符号量子纠错程序。为了构造解码器鲁棒性问题的实例，我们还需要引入符号量子纠错程序的概念。在符号程序中，具体概率 p 被变量 x所替代。我们将 $P(x)$记作一个符号程序，以表示每个符号程序都定义了一个从概率向量到量子纠错程序的函数。我们有时也会将具体程序写作 $P(v)$，以强调其作为在某一点对符号程序求值的视角。

例如，图2a 中的重复码程序的符号版本如图7所示。该程序对应于程序值函数 $P(x_1,\ x_2,\ x_3)$，而图2a 中的程序是其在 $P(0。01,\ 0。01,\ 0。01)$处的求值结果。

5 概率多项式透镜

现在，我们介绍一类称为误差多项式的多项式，它们为分析和解决准确性与鲁棒性问题提供了一种实用的表述方式。正如我们将要展示的，准确性

问题可归结为计算一个误差多项式，而鲁棒性问题则可归结为对误差多项式进行约束优化。

我们首先计算某个特定错误比特串的概率。注意，位串的概率 $f(e; P,$ D) 可以写成各个错误信道概率的简单表达式。每条错误信道语句都表示一个独立的随机变量，因此，位串的概率就是每个索引处各结果概率的乘积。例如，在我们的三量子比特重复码程序中，有三条错误信道语句，其对应的概率分别为： x_1， x_2 和 x_3。一种可能的错误比特串是“010”，其概率可用表达式 $(1 - x_1)(x_2)(1 - x_3)$ 表示。我们将此表达式称为错误最小项，类比于布尔代数中的最小项。

定义 *5.1*。给定一个错误比特串 $e = e_1 \ldots e_n$，错误最小项 m_e 是以下乘积

$$\prod_{e_i=1} x_i \prod_{e_i=0} (1 - x_i)$$

关于符号变量 x_1、……、 x_n。

由于错误比特串表示的是不相交事件，因此长度为n 的一组错误比特串的概率，等于各个比特串概率之和。注意，这个和是关于变量 x_1、……、 x_n的多项式。

定义 *5.2*。 给定一组错误比特串 L，误差多项式 p_L 是以下各项之和

$$\sum_{e \in L} m_e.$$

例 *5.3*。 考虑我们的贯穿示例，即三比特重复码。最大似然解码器会在汉明重量至少为 2 的比特串集合上产生逻辑错误：{111、110、011、101}。与该集合对应的误差多项式为

$$p_L = x_1x_2x_3 + x_1x_2(1 - x_3) + (1 - x_1)x_2x_3 + x_1(1 - x_2)x_3.$$

注意，对于解码器 d，逻辑错误的概率是一组错误比特串的概率，因此可以用误差多项式来描述。具体而言，它是使 d 产生逻辑错误的错误集合的概率。在例 5.3中，我们求出了表示贯穿示例中最优解码器逻辑错误率的多项式。因此，我们可以用多项式 p_L来描述每个解码器评估问题。

首先，准确性问题等价于在特定点对 p_L 进行求值。

定理 5.4（准确性作为多项式求值）。给定一个量子纠错程序 $P(v)$和一个解码器 d,

$$\mathrm{Err}(P, d) = p_L(\overline{v}),$$

其中 $L = \{e : d(\mathrm{synd}(e)) \neq \mathrm{obs}(e)\}$。

其次，鲁棒性问题等价于对 p_L进行约束最大化。

定理 5.5（鲁棒性作为多项式最大化）。给定一个符号量子纠错程序 P（x）、解码器 d和一个约束集 R,

$$\max_{\overline{v} \in R} (\mathrm{Err}(P(\overline{v}), d)) = \max_{\overline{v} \in R} p_L(\overline{v}),$$

其中 $L = \{e : d(\mathrm{synd}(e)) \neq \mathrm{obs}(e)\}$。

误差多项式的界。 当然，这些问题的多项式表述并不能克服其根本上的难解性。多项式 p_L 包含指数多个项，因此，即使只是在某一点对其求值，也未必可行。因此，我们推导出 p_L 的界，无需构造整个多项式即可计算这些界，只需使用来自已探索错误比特串集合 S 的项。我们的界依赖于以下定理，该定理本质上指出，集合 L 的概率下界由我们已见到的 L 中元素的概率给出，上界则基于最坏情况假设：所有尚未确认属于 L^C的比特串实际上都属于 L。

定理 5.6。 给定任意两个长度为 n 的比特串集合 L 和 S，在集合 $[0,1]^n$上有以下不等式：

$$p_{L\cap S} \leq p_L \leq 1 - p_{S\setminus L}.$$

推论 5.7。 给定任意两个长度为 n 的比特串集合 L 和 S，以及约束集 $R \subseteq [0,1]^n$，有以下不等式：

$$\max_{\overline{x}\in R}(p_{L\cap S}(\overline{x})) \leq \max_{\overline{x}\in R}(p_L(\overline{x})) \leq 1 - \min_{\overline{x}\in R}(p_{S\setminus L}(\overline{x})).$$

6 我们的估计算法

我们的主要算法会系统地探索错误比特串的空间，并利用第 5 节的结果，迭代地优化准确性问题或鲁棒性问题解的界限。我们在算法 1中展示通用的鲁棒性版本。

算法会遍历错误比特串集合，直到穷尽所有比特串或被中断（大多数程序所定义的错误比特串空间都过于庞大，无法完全枚举）。在第 6.2节中，我们将讨论迭代顺序的选择。每枚举出一个新的比特串，就将其加入已见比特串集合（第 5 行）。然后，将其转换为综合征和可观测量对，对该综合征运行解码器，并检查是否存在逻辑错误（第 6 行）。如果结果是逻辑错误，就将该比特串加入集合 L（第 7 行）。最后，通过约束多项式优化，为鲁棒性问题实例的解推导出新的界限（第 8 和第 9 行）。

在准确性问题的特殊情形中，集合 R 是一个单元素集合，因此第 8 行和第 9 行的优化简化为在单个点处计算多项式。对于一般的鲁棒性问题，我们必须在无限域上求解优化问题。在第 6.1 节中，我们将介绍针对无限约束集进行精确优化的方法；这类约束集由每个变量的上界和下界定义。

6.1 超矩形上的优化

在本小节中，我们将提出一种过程，用于在一种特定类型的约束集——超矩形——上最大化多项式。借助这一过程，我们可以解决这样的鲁棒性问题：为每个物理错误率定义一个可能取值区间。我们将通过一系列逐步改进来介绍这一方法。首先，我们证明该问题可在指数时间内判定。接着，我们运用剪枝和简化技术，在不牺牲精确优化的前提下缩小搜索空间。最终，我们提出完整的多项式优化算法，即算法 2。

定义 *6.1*。 超矩形 R 是 $\mathbb{R}^n$ 的一个子集，由一对向量 ℓ，$u \in \mathbb{R}^n$ 描述，满足 $x \in R \Leftarrow\Rightarrow \ell \leq x \leq u$。

算法1 我们用于计算解码器鲁棒性可靠界的算法

过程 bound($P(x)$：符号量子纠错程序，d：解码器，$R \subseteq [0,1]^n$)
  令 B 为 P的错误比特串集合
  将错误比特串的空集合 L 和 S 初始化
  对 $e \in B$ **执行** ▷ 参见第6.2
    $S.add(e)$
    若 $d(\text{synd}(e)) \neq \text{obs}(e)$ **则**
      $L.add(e)$
    下界 $\leftarrow$最大值$_{x\in R}(p_L(x))$ ▷ 参见第6.1
    上界 $\leftarrow 1 -$最小值$_{x\in R}(p_{S\setminus L}(x))$
  返回(下界，上界)

第一个关键观测值是，尽管超矩形包含无限多个点，但极值出现在有限多个顶点之一：每个变量都被设为其上界或下界的点。

定理 6.2。 设 p 是一个关于 n 个变量的误差多项式，R 是一个边界为 (ℓ, u) 的超矩形。那么，p 在 R 上的极值可在一个顶点 x^* 处取得，并且该顶点满足 $x_i^* \in \{\ell_i, u_i\}$对所有$i \in \{1, \ldots, n\}$均成立。

直观而言，极值点位于顶点上，是因为误差多项式是一个多重线性函数，而线性函数在有界区间上的极值会在端点处取得。该定理通过对各顶点进行穷举搜索，立即给出了一个用于求解该优化问题的指数时间算法。鉴于一般的多重线性多项式优化问题是 NP难的，我们不认为存在多项式时间算法。不过，在实践中，我们可以通过对顶点进行离散搜索时采用额外的剪枝，优于蛮力枚举。

偏导数剪枝。 为了避免在每个顶点处都计算误差多项式，我们采用类似于分支定界或 DPLL 的策略。我们逐一遍历每个变量 x_i，并尝试证明最优点必定位于两个平面 $x_i = \ell_i$和 $x_i = u_i$之一上，从而将剩余搜索空间缩小一半。为了获得这样的证明，我们需要确定偏导数 $\frac{\partial p}{\partial x_i}$的符号 —

例*6.3*。 假设我们希望最大化 误差多项式

$$p = x_1(1 - x_2) + (1 - x_1)x_2$$

在约束集 $0.009 \leq x_i \leq 0.011$ 上。我们有 $\frac{\partial p}{\partial x_1} = (1 - x_2) - x_2$，只要 $x_2 < 0.5$，它就为正；而这对整个约束集都成立。因此，将 $x_1 = 0.011$设为 p 时，p 达到最大值。

在例 6.3中，确定偏导数的符号很简单，因为它是一个单变量函数。一般来说，这一步相当于求解另一个多项式最大化问题，只是变量少了一个。我们通过引入无法确定符号的可能性来放宽问题，此时变量保持未赋值。我们通过分别优化各个最小项来计算可靠界，因为对于写成求和形式的多项式 p $\sum_{i=1}^{k} m_i$，我们知道

$$\min p = \min \sum_{i=1}^{k} m_i \geq \sum_{i=1}^{k} \min m_i \qquad \text{and} \qquad \max p = \max \sum_{i=1}^{k} m_i \leq \sum_{i=1}^{k} \max m_i.$$

优化单个最小项很简单，与变量总数无关，因为每个变量恰好只出现一次，要么作为 x_i 因子，要么作为 $(1-x_i)$ 因子。如果整个区间 $[\sum_{i=1}^{k}\min\ m_i,\ \sum_{i=1}^{k}\max\ m_i]$ 的符号相同，我们就可以确定偏导数的符号。

例 *6.4.* 对多项式应用偏导数剪枝 $p = x_1(1-x_2) + (1-x_1)x_2$，根据例 6.3，我们得到以下关于 $\frac{\partial p}{\partial x_1}$ 的界：

$$\min \frac{\partial p}{\partial x_1} \geq \min(1-x_2) + \min(-x_2) = (1-0.011) + (-0.011) = 0.978,$$
$$\max \frac{\partial p}{\partial x_1} \leq \max(1-x_2) + \max(-x_2) = (1-0.009) + (-0.009) = 0.982.$$

由于整个区间 $[0.978, 0.982]$ 都为正数，因此可以得出 $x_1 = 0.011$ 是 p 的最大值。

示例 *6.5.* 举例来说，如果我们无法确定偏导数的符号，假设我们希望最大化多项式

$$p = x_1(1-x_2)x_3 + (1-x_1)x_2(1-x_3).$$

我们有 $\frac{\partial p}{\partial x_1} = (1-x_2)x_3 - x_2(1-x_3)$。对各项分别求界：

$$\min \frac{\partial p}{\partial x_1} \geq \min((1-x_2)x_3) + \min(-x_2(1-x_3)) = -0.002,$$
$$\max \frac{\partial p}{\partial x_1} \leq \max((1-x_2)x_3) + \max(-x_2(1-x_3)) = 0.002.$$

区间 $[-0.002, 0.002]$ 包含正值和负值，因此我们无法确定 $\frac{\partial p}{\partial x_1}$ 的符号，也就无法剪枝搜索空间。

匹配项化简。提高偏导数剪枝精确度的一项重要优化，是化简偏导数中的匹配项。给定一个特定的目标索引 x_i，请注意，仅在 x_i 处不同的两个项对 $\frac{\partial p}{\partial x_i}$ 的贡献大小相同、符号相反，因此会相互抵消。在分别最大化每一项之前，我们先进行代数化简，去除匹配项，以提高偏导数剪枝成功的可能性。

示例 *6.6*。这个示例展示了匹配项化简如何提高精确度。假设我们希望最大化多项式

$$p = x_1(1-x_2)x_3 + (1-x_1)(1-x_2)x_3 + x_1x_2x_3.$$

首先，我们尝试在不进行化简的情况下应用偏导数剪枝。我们计算

$$\frac{\partial p}{\partial x_1} = (1-x_2)x_3 + -(1-x_2)x_3 + x_2x_3.$$

然后，我们分别优化各项：

$$\min \frac{\partial p}{\partial x_1} \geq \min((1-x_2)x_3) + \min(-(1-x_2)x_3) + \min x_2x_3 = -1.9 \cdot 10^{-3},$$
$$\max \frac{\partial p}{\partial x_1} \leq \max((1-x_2)x_3) + \max(-(1-x_2)x_3) + \max x_2x_3 = 2.1 \cdot 10^{-3}.$$

区间 $[-1.9 \cdot 10^{-3}, 2.1 \cdot 10^{-3}]$ 既包含正值也包含负值，因此我们无法确定 $\frac{\partial p}{\partial x_1}$ 的符号，也就无法剪枝搜索空间。另一方面，如果我们简化为

算法 2 误差多项式的分支优化

过程 最大化(p : 误差多项式, ℓ, u)
  设 n 为 p中的变量数
  初始化解向量 $s_1, \ldots, s_n$
  对于 $i \in \{1, \ldots, n\}$ **执行**
    令 p' 为 p 关于 x_i 的简化偏导数。
    (下界, 上界) $\leftarrow$ 逐项界定(p', ℓ, u) ▷ 示例中的过程 6.4
    如果 下界 > 0, **则**
      将 u_i 代入 p 中的 x_i
      $s_i \leftarrow u_i$
    否则, 如果 上界 < 0, **则**
      将 ℓ_i 代入 p 中的 x_i
      $s_i \leftarrow \ell_i$
  令 $j_1, \ldots, j_k$ 为剩余自由变量的索引
  通过穷举找出使 p 最大的 $s_{j_1}, \ldots, s_{j_k}$
  返回 s

$\frac{\partial p}{\partial x_1} = x_2 x_3$, 那么我们会看到

$$\min \frac{\partial p}{\partial x_1} \geq \min x_2 x_3 = 8.1 \cdot 10^{-5} \qquad \text{and} \qquad \max \frac{\partial p}{\partial x_1} \leq \max x_2 x_3 = 1.21 \cdot 10^{-4}.$$

因此我们得出结论, 固定 $x_1 = 0.011$ 可使 p 最大。

完整的多项式最大化算法如算法 2 所示。我们遍历每个变量, 应用偏导数剪枝, 尝试将每个变量固定为两个边界值之一。然后, 我们对剩余搜索空间执行穷举搜索。

定理 6.7. 算法 *2* 会返回给定误差多项式在给定超矩形上的一个最大值点。

6.2 枚举策略

算法 1 中的伪代码没有指定错误比特串空间的遍历顺序。在本小节中, 我们将介绍所考虑的策略。首先介绍核心的简单技术——汉明重量顺序遍历, 然后介绍两种正交扩展。

汉明重量顺序。 我们的核心策略是按照汉明重量从低到高枚举错误比特串。错误比特串的概率会随错误数量增加而呈指数级衰减, 因此大部分概率质量集中在低重量比特串上。因此, 重量较低的比特串能提供更多信息。如果低重量比特串导致逻辑错误, 下界就会显著提高; 如果没有, 上界则会显著提高。

分割搜索。 分割搜索 扩展利用并行性, 更快地发现最初的逻辑错误。请注意, 我们可以并行枚举错误比特串, 这就引入了如何在不同枚举线程之间划分搜索空间的选择。给定 k 个工作线程, 最基本的策略是将工作线程 i 分配给枚举顺序中位置等于 i 模 k的比特串。另一种选择是将工作线程分配到搜索空间的不同区域。从理论上讲, 对于码距为 d 且使用最优解码器的纠错码, 所有逻辑错误都应源于权重大于或等于 $d/2$的错误比特串。因此, 枚举权重低于该值的错误可以改善上界, 但不太可能改善

下界。在分割搜索 策略中，我们指定一半工作线程从权重0开始枚举，另一半从权重 $\lfloor d/2 \rfloor + 1$开始枚举，其中 d 是给定的码距拟设。

局部搜索。 局部搜索扩展基于这样一种直觉：对逻辑错误比特串进行小幅扰动，很可能也会产生逻辑错误。局部搜索策略是自适应的。每当发现一个会导致逻辑错误的比特串时，就会暂停预先规划的汉明重量顺序枚举，转而探索通过施加扰动得到的邻居。其目标是在这些相关比特串也会导致逻辑错误的假设下，更快地提高下界。穷尽邻居集合后，便恢复汉明重量顺序搜索。局部搜索由一组局部移动参数化，这组移动定义了要考虑哪些邻居比特串。在本研究中，我们评估了两种局部移动集合（以及它们的并集）。

第一种局部移动集合是翻转，即翻转比特串中的任意一个位置。采用这一移动集合的动机是，逻辑错误由成簇的物理错误引起，而其他位置上的孤立变化很可能会产生新的逻辑错误比特串。

第二种局部移动集合是移位，它包含比特串的所有循环移位。移位算子 τ 将所有比特向右移动一位（对比特串长度取模）：

$$\tau(e_1 \dots e_n) = e_n e_1 \dots e_{n-1}.$$

移位集合中的局部移动是这些 $n-1$ 偏移量 τ， τ^2、……、 τ^{n-1}.这里的动机是纠错码的平移对称性。量子纠错程序通常包含许多相同量子电路的副本，并将它们应用于不同的量子比特集合。对比特串进行移位的目标，是找到一个比特串，使其对应于同一物理错误从一个副本平移到另一个副本。如果一组物理错误会在第一个副本中导致逻辑错误，那么它很可能也会在其他副本中导致逻辑错误。

7 结合采样与枚举

本节将介绍一种技术，将基于采样的方法的优势融入我们用于解决准确性问题的枚举算法中。单独使用枚举可能难以处理具有大量低概率比特串长尾分布的大规模量子纠错程序。因此，枚举搜索可能会遗漏很大一部分概率质量，导致数量$p_{S \setminus L}(o)$相对较大。

–

在定理 5.6中，通过做出最保守的假设——没有任何未见的错误比特串会导致逻辑错误——我们得到了一个无条件可靠的下界。同样，我们假设所有未见的错误比特串都会导致逻辑错误，从而得到上界。不过，这些界可能相当宽松。在求解准确性问题时，我们可以通过采样，更准确地估计尚未枚举的错误对逻辑错误率的贡献。从未见集合中采样错误比特串，并检查它们是否会导致逻辑错误，可以得到一个经验估计，表示未见错误导致逻辑错误的概率。

假设我们正在求解某个准确性问题，其中错误比特串的长度为 n。令 $S \subset \{0,1\}^n$为已枚举错误比特串的一个子集，并令 $L \subset \{0,1\}^n$ 为会导致逻辑错误的错误比特串。那么，我们可以将逻辑错误率分解为来自S 中元素的贡献，以及来自 S之外元素的贡献：

$$\Pr_{(s,o)\sim P}[d(s) \neq o] = p_L(v) = p_{L \cap S}(v) + p_{L \cap S^c}(v)$$

估计未枚举贡献。 问题在于，我们无法直接计算 $p_{L \cap S^c}(o)$，因为我们不知道 $L \cap S^c$ 中的元素。因此，我们对该值进行统计估计

$p_{L\cap S^c}(v)$。为此，我们将 $p_{L\cap S^c}(v)$ 改写为期望值：

$$p_{L\cap S^c}(v) = \sum_{e\in S^c} \chi_L(e) m_e(v) = p_{S^c}(v) \cdot \mathbb{E}_{e\sim P|e\in S^c}[\chi_L(e)],$$

其中， χ_L 是指示函数，对于 L 中的输入，其值为 1；其他情况下为 0。注意，借助 S，可以直接计算 p_{S^c}，即 $1 - p_S$。令 $\theta = \mathbb{E}_{e\sim P|e\in S^c}[\chi_L(e)]$.。正如符号所示，我们将 θ 视为一个伯努利随机变量的参数，该变量表示“发生逻辑错误”事件。我们可以用经验估计 Θ 替换 θ，从而得到逻辑错误率的估计值：

$$p_L(v) \approx p_{L\cap S}(v) + (1 - p_S(v))\hat{\Theta}.$$

为了得到这样的经验估计，我们从 S^c 中（通过拒绝采样）抽取 N 个样本 s_1，……， s_N，并计算样本均值 $\frac{1}{N}\sum_{i=1}^{N} \chi_L(e_i)$。−

置信区间。到目前为止，我们已经得到了逻辑错误率的点估计。不过，类似于通过枚举得到的可靠上界和下界，我们希望能够以某种方式保证真实逻辑错误率落在某个范围内。由于我们使用的是随机样本，这种保证是概率性的，从而得到一个置信区间。对于置信参数 α，我们可以采用标准方法，根据 Chernoff 界推导出 $1 - \alpha$ 置信区间（例如，见 [29, 第 10 节。1]），其形式化表述见定理7.1。

定理 7.1。 设 $\theta \in [0, 1]$ 是未知参数，且 $X_1, \ldots, X_N$ 是独立同分布的伯努利(θ)变量。设$\hat{\Theta} := \frac{1}{N}\sum_{i=1}^{N} X_i$ 为样本均值。在事件 $0 < \hat{\Theta} < 1$发生时，将随机端点 $L_\alpha \in [0, \hat{\Theta}]$和 $U_\alpha \in [\hat{\Theta}, 1]$ 定义为下列方程的唯一解：

$$KL(\hat{\Theta}||U_\alpha) = KL(\hat{\Theta}||L_\alpha) = \frac{1}{N}\ln\left(\frac{2}{\alpha}\right),$$

其中 $KL(p||q) := p\ln\left(\frac{p}{q}\right) + (1-p)\ln\left(\frac{1-p}{1-q}\right)$是库尔贝克-莱布勒散度。随机区间

$$[L, U] := \begin{cases} \left[0, 1 - (\alpha/2)^{1/N}\right] & (\hat{\Theta} = 0) \\ [L_\alpha, U_\alpha] & (0 < \hat{\Theta} < 1) \\ \left[(\alpha/2)^{1/N}, 1\right] & (\hat{\Theta} = 1) \end{cases}$$

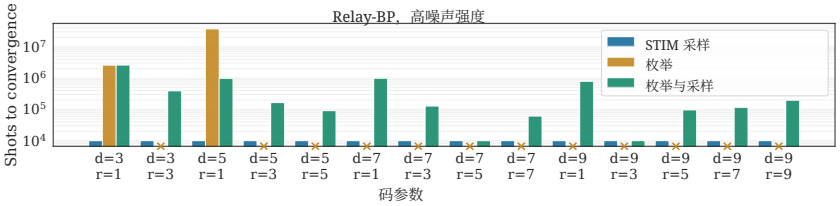
是 $1 - \alpha$ 关于 θ的置信区间，即

$$\Pr_\theta\{\theta \notin [L, U]\} \le \alpha.$$

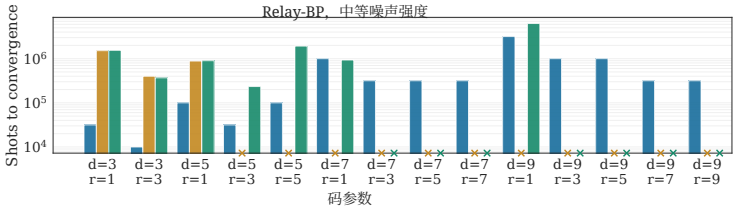
请注意，该定理也适用于通过采样直接估计逻辑错误率。我们将在实验评估中用它推导 Stimsampling 的置信区间。

8 实现与评估

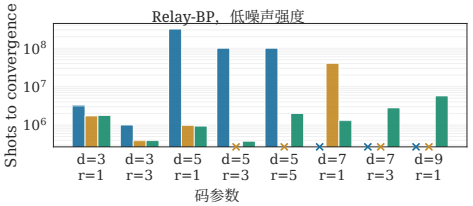
我们将估计算法实现为一个 Python 库。该库以 Stim 电路作为输入。它利用 Stim 的功能将量子电路编译为探测器错误模型，这些模型由一系列伯努利错误信道组成，并附有相应的概率以及对综合征和可观测量的影响。给定一个探测器错误模型，我们可以针对错误比特串 e提取 synd(e) 和 obs(e)。我们的实现采用并行化方式，在 k核机器上运行时会启动 $k - 1$ 个枚举工作线程。每个工作线程负责探索错误比特串空间中的不同区域，对枚举出的每个比特串运行解码器，并将其分类为会导致逻辑错误或不会导致逻辑错误的比特串。



(a) $p = 0.01$



(b) $p = 0.001$



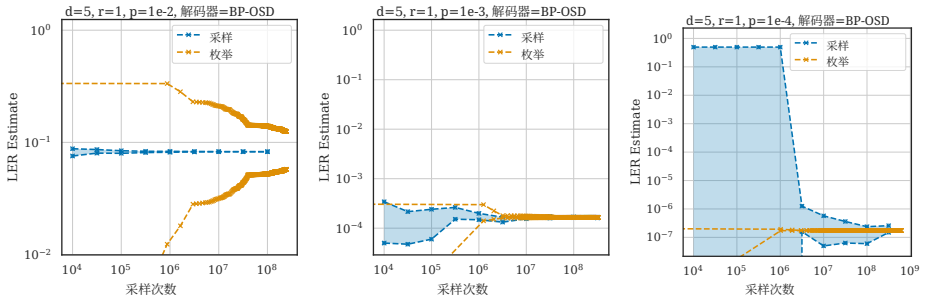
(c) $p = 0.0001$

图 8. 逻辑错误率问题的收敛性。

实验设置。除非另有说明，以下实验条件适用于所有实证评估。我们的搜索和 Stim 采样均分配到一台 AMD EPYCTM 7763 2.45 GHz 处理器的 16 个核心，并通过分布式研究集群使用 32 GB 随机存取存储器。运行 24 小时或达到 10^9 个采样轮次后，我们将终止实验。

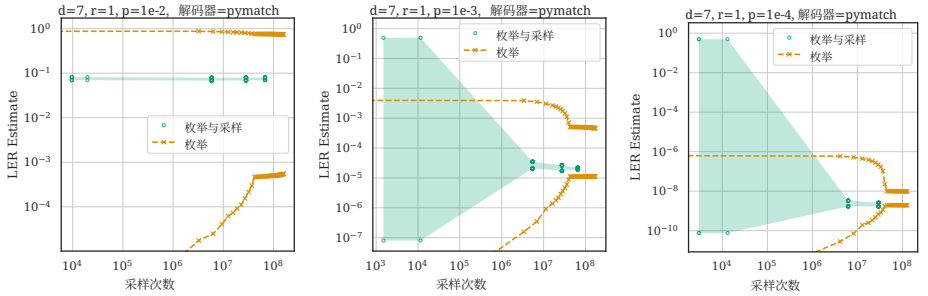
量子纠错程序基准测试。我们将我们的方法应用于一类主要的量子纠错码——旋转表面码的存储实验。通过改变码距 d 和综合征提取轮数 r，我们构建了多个存储实验程序。具体而言，我们考虑 $d \in \{3、5、7、9\}$ 和 $r \in \{1、3、5、7、9：\ r \le d\}$。在所有情况下，我们均使用 si1000噪声模型[19] 。si1000噪声模型由谷歌开发，用于对超导器件中的错误进行逼真建模（“si1000”是 superconducting-inspired 的缩写，表示 1000 纳秒的错误纠正周期）。该模型会在每个量子门、复位和测量操作之前应用退极化错误信道。噪声强度由单个参数 p控制，该参数设定双量子比特门的退极化概率。其他操作的错误率则定义为 p的固定倍数。我们使用 $p = 0.001$ 进行评估，该值大致对应于当前量子硬件的最先进水平，并在两个方向上各取一个数量级：$p = 0.01$ 表示高噪声环境，$p = 0.0001$ 表示近期未来可能实现的改进型硬件。

解码器。 我们评估了代表当前最先进水平的三种不同解码器：pymatch [24], BP-OSD [36], 和 Relay-BP [35]。pymatch 基于将解码问题归约为最小权完美匹配。BP-OSD 和 Relay-BP 基于贝叶斯网络中的置信传播。



(a) p = 0.01 (b) p = 0.001 (c) p = 0.0001

图 9. 比较枚举与采样。每个图表显示同一程序在不同噪声强度下，估计质量随采样轮次变化的情况。



(a) p = 0.01 (b) p = 0.001 (c) p = 0.0001

图 10. 比较单独使用枚举与使用枚举与采样。每幅图都展示了同一程序在不同噪声强度下，估计质量随采样轮次的变化。

研究问题。 我们设计了评估，以回答以下研究问题。 **（RQ1）**对于准确性问题，与随机采样相比，我们的估计策略效率如何？ **（RQ2）**对于一般的鲁棒性问题，我们能够认证哪些界限？ **（RQ3）**改变错误比特串枚举策略会产生什么影响?

8.1 (RQ1) 准确性

为回答 (RQ1)，我们将自己提出的方法与 Stim 仿真进行了比较。Stim 仿真是目前应用最广泛的、用于解决准确性问题的基于采样的方法。我们使用第 7 节中的双侧 Chernoff 界方法，将 Stim 返回的逻辑错误率点估计转换为置信区间。对于 Stim 采样基线，以及我们通过结合枚举与采样得到的概率界，我们固定 $\alpha = 0.01$，从而使实际逻辑错误率有 99% 的概率落在界定的区间内。

图 8 总结了不同方法在表面码电路上运行 Relay-BP 解码器时的效率。“枚举”指我们提出的基础算法（算法 1），“枚举与采样”则使用第 7 节的方法，通过 10,000 个随机样本获得概率界。对于每种算法，我们绘制了将逻辑错误率估计到一个数量级范围内所需的采样轮次。也就是说，我们找出满足下式的最小采样轮次：

$u/\ell \leq \sqrt{10}$，其中 u 和 ℓ 分别是算法给出的上界和下界。需要强调的是，这种直接比较并非完全公平，因为 Stim 采样和枚举与采样只能给出概率界，而枚举界则无条件可靠。‘x’表示相应算法未能在时间和内存限制内达到收敛判据。如果没有任何算法达到该条件，则完全不纳入该基准测试。我们按噪声强度将数据分图展示，以突出相对性能和绝对性能中的关键规律。

比较枚举与采样。降低噪声强度对随机采样的影响，与对我们采用的两种基于枚举的方法的影响恰好相反。对于采样而言，降低噪声强度会大幅增加达到收敛所需的采样轮次。这符合我们的直觉：物理错误率降低，逻辑错误率也随之降低。因此，我们要估计的量更小，需要更多样本。另一方面，噪声强度较低时，枚举的表现更好。噪声强度降低不会影响我们通过枚举能够探索的错误比特串数量，因为事后计算比特串概率所需的运行时间不取决于概率本身。主要影响在于，更多概率质量集中在少数汉明重量较低的字符串上。

相反的缩放关系使不同噪声水平下的相对效率呈现出不同情况。我们发现，在高噪声强度下，Stim 分析每个程序中的Relay-BP 解码器的效率至少高出一个数量级，而在最低噪声强度下，基于枚举的方法效率更高。我们通过在图 9中固定一个量子电路来考察这种关系。这些图均展示了逻辑错误率界限随时间的变化。解码器 BP-OSD 和程序（$d = 5$, $r = 1$ 表面码）保持不变，但噪声强度有所变化。我们预计，随着量子硬件不断改进，物理错误率会随时间降低，因此这些结果表明，我们的方法与未来的量子系统设计息息相关。

比较枚举与采样和枚举。数据还支持这样一种观点：“枚举与采样”方法比单纯枚举更有价值。例如，图8a显示，在高噪声强度下，单独使用枚举仅能在 2 个基准测试上收敛，而枚举与采样能够利用这一情况下采样的优势，在全部 14 个程序上收敛。我们通过在图 10中固定一个基准测试来突出这一效果。

总结：在最高噪声级别下，采样比枚举用更少的采样轮次即可收敛，但在最低噪声级别下则恰好相反。将枚举与采样结合起来，可以普遍减少所需的采样轮次，有时效果非常显著，但代价是将无条件界限换成概率保证。

8.2 (RQ2) 鲁棒性

接下来，我们将评估该方法在鲁棒性问题上的可扩展性。对于每个基准实例，我们构造一个不确定性为 10% 的鲁棒性实例。也就是说，给定一个具有伯努利信道参数 v 的具体程序，我们在符号程序中将 x 限制在超矩形 $0.9 \cdot v \leq x \leq 1.1 \cdot v$ 内。

图 11 以类似图 8 的方式总结了结果。不过请注意，这里柱状图按解码器分组，因为我们只有一种鲁棒性算法需要评估。一般的鲁棒性问题比准确性问题这一退化情形更难求解，这并不令人意外。在我们的实验中，结果表明，我们的算法只能为相对较小的程序证明紧致的鲁棒性界。尽管如此，在小规模实例上验证解码器鲁棒性，仍有助于增强我们对其在更大规模实例上性能的信心。请注意，即使这些小型程序，也会在庞大的搜索空间上引出约束优化问题。例如，

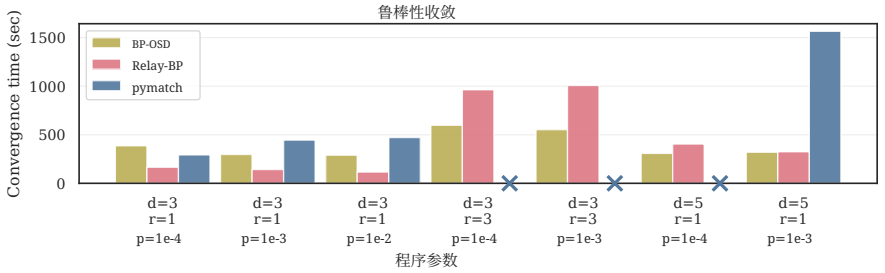
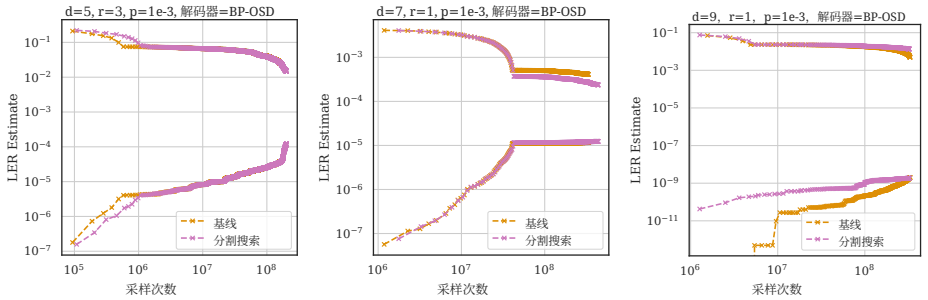


图 11。鲁棒性问题的收敛。



(a) d = 5 (b) d = 7 (c) d = 9

图 12。E 评估分割搜索策略。每个图展示了一个具有不同码距的程序 .

这些 $d = 3,\ r = 3$ 量子电路包含 286 条错误信道语句，因此最大化问题涉及 286 个变量。一组 2^{286} 可能的变量赋值远非蛮力枚举所能遍历，因此求解这一实例证明了偏导数剪枝的有效性。

相对解码器鲁棒性。 解码器鲁棒性为比较不同解码器提供了一个新的角度。在我们的评估中，我们发现 RELAY-BP 解码器在存在错误率不确定性时，表现不如另外两种解码器。在图 11 所示的 6 个程序中，RELAY-BP 解码器的准确率值与最坏情况鲁棒性值之间的差距最大，达到 28.6%。BP-OSD 和 PYMATCH 解码器的差距分别为 21.6% 和 21.7%。对于一个量子纠错 程序（$d = 3,\ r = 3,\ p = 0.001$），如果关注鲁棒性而非准确性，我们甚至会看到解码器的相对排序发生变化。 RELAY-BP 解码器的准确性更高，但鲁棒性不如 BP-OSD 解码器。

总结： 我们的算法能够量化多个基准测试中的鲁棒性，其中规模最大的是 $d = 3,\ r = 3$ 表面码电路，这证明了我们的方法能够解决超矩形上的鲁棒性问题。鲁棒性分析结果表明，relay-bp 解码器的鲁棒性可能不如 pymatch 和 BP-OSD。

8.3（RQ3）枚举策略

最后，我们研究枚举错误比特串时所选策略对算法性能的影响。我们重点关注准确性问题，以便将枚举与求解过程区分开来。

	分数改进	几何平均命中率
枚举		
FLIP	13/25（0.52）	1.05
移位	24/25 (0.96)	0.73
移位 ∪ 翻转	18/25 (0.72)	0.84
枚举与采样		
FLIP	46/81 (0.57)	1.08
移位	55/81 (0.68)	0.89
移位 ∪ 翻转	53/81 (0.65)	0.97

图 13. 将局部搜索的不同变体与汉明重量顺序基线进行比较。基准测试的采样次数比，是局部移动策略达到收敛所需的采样次数除以基线搜索所需的采样次数。

分割搜索。我们首先评估“分割搜索”策略。回顾一下，该策略将一组枚举工作线程分成两半，其中一半从重量为 $\lfloor d/2 \rfloor + 1$ 的错误开始搜索。我们在图12中，将分割搜索策略与按汉明重量进行的标准枚举进行比较，使用了三个具有代表性的示例程序。

总体而言，我们发现，“分割搜索”策略与标准枚举在下界的紧致程度上早期存在差异，但随着采样次数增加，二者往往会趋于收敛。在采样次数有限的情况下，哪种方法更优取决于程序。对于码距为 5 的示例，导致解码器产生逻辑错误的比特串集中度较高，但这无法抵消被访问比特串的概率较低这一因素。然而，对于更大的码距，向前“跳跃”至高汉明重量错误比特串空间是有效的。在码距为 9 的程序中，标准枚举达到分割搜索方法首次记录的下界所需的采样次数约为 $6\times$ 倍，因为它花费了大量精力枚举不会导致逻辑错误的低重量错误比特串。

局部搜索。 接下来，我们评估局部移动扩展的影响。图 13. 展示了在整个基准测试套件上与基线枚举的比较。对于每种局部移动集合，我们记录两项汇总统计数据。第一列是“改进比例”，统计局部移动策略收敛所需的采样轮次少于基线的基准测试（程序、解码器对）数量。第二列反映差异的幅度。采样次数比定义为局部移动策略收敛所需的采样次数除以基线收敛所需的采样次数。因此，采样次数比小于 1 表示性能更好。我们计算所有基准测试的采样次数比的几何平均数。这两个表分别列出枚举界和概率枚举与采样界。

对于这两种界限，我们都发现移位策略优于基线。不过，在枚举界限中，这种提升更加显著。除一个基准测试外，其余基准测试的收敛速度均有所提升，平均提升 27%；相比之下，枚举与采样仅在 68% 的基准测试中实现更快收敛，平均提升 16%。另一方面，翻转策略的表现大致与基线持平，在略多于一半的基准测试中有所改善，但平均采样次数比大于 1，表明在基线表现更好的情况下，差距更大。两种策略结合后的结果居于两者之间。

总结： 总体而言，其他枚举策略的效果参差不齐。拆分搜索策略会在低采样次数区间产生不同表现：在低距离基准测试中，其表现不如标准的按汉明重量排序枚举；而在高距离基准测试中则表现更好。在局部搜索方面，翻转策略的表现与基线大致相当，而移位策略似乎略为有效。

9 相关工作

量子编程语言。 如前所述，我们对 量子纠错 程序的定义，是对 Stim 电路格式的直接形式化 [17]。量子编程语言的形式化定义由来已久，这些语言通常以程序验证为设计目标。我们用于 量子纠错 程序的语言与 **qwhile** 量子程序语言[41], 密切相关；该语言提出时附有指称语义和相对完备的霍尔式证明系统。后续研究在 Roqc [43] 和 Isabelle/HOL [30] 证明助手中对 **qwhile** 的语义进行了机械化。其他值得关注的形式化量子编程语言还包括 *Q*WIRE [31], SQIR [23], 和 QBRICKS [8]。与这些语言相比，我们的语言更为简单，因为它不支持无界循环或经典控制流。另一方面，概率误差通道以及综合征和可观测量声明，是我们在形式语义中所研究的纠错场景所特有的功能。

概率编程语言。概率编程语言是一个丰富的研究领域 [2, 9–11, 20, 28, 38]。与概率编程的许多其他应用一样，准确性与鲁棒性问题是针对我们程序的概率推断任务。我们的设定与典型情况的一个根本区别在于，我们假设只能黑箱访问解码器本身，因为解码算法过于复杂，无法直接嵌入概率编程语言。因此，我们无法直接应用加权模型计数等符号技术 [9, 34]。

解码器准确性的启发式方法。与本研究同期提出了另一种解决直接蒙特卡罗采样采样次数效率低下问题的方法。Beverland 等人 [5]提出了两种主要技术。一种是多种子分裂方法（基于先前的研究[6, 32]）。分裂方法是一种用于模拟稀有事件的标准蒙特卡罗算法 [37]，其基础是多层马尔可夫链蒙特卡罗采样。在此背景下，该方法利用较高噪声强度下的高置信度估计，外推低噪声强度下的逻辑错误率。另一种技术是对失败谱进行曲线拟合；失败谱表示在每个汉明重量下会导致逻辑错误的错误比特串所占比例。给定失败谱的闭式表达式，他们通过对每个汉明重量的贡献求和，推导出逻辑错误率。Ye 和 Palsberg [40] 也开发了类似的曲线拟合框架。我们的工作在以下方面有所不同。

（1）我们不对物理错误模型作任何假设。曲线拟合方法只能原生地对物理错误的均匀概率进行建模。（2）我们不对逻辑错误的结构作任何假设。划分方法假设会导致逻辑错误的错误比特串集合在单比特翻转下是连通的，而曲线拟合则通过选择平滑的失效谱假设形式增加了一项建模假设。（3）我们可以表述并解决一般性的鲁棒性问题。

解析与符号量子纠错 验证。 手工构造的阈值存在性证明 [15, 42]已被应用于特定的解码算法和纠错码，以证明有限阈值的存在：当码距增加时，逻辑错误率渐近趋于 0 的物理错误率。它与我们的黑箱自动数值分析互为补充。另一项

工作将符号技术和 SMT 求解应用于验证 量子纠错 协议 [13, 26]。这里的重点是验证 量子纠错 程序本身，假设解码器完美，并检查是否存在（或验证不存在）会导致逻辑错误的小规模物理错误集合。

10 结论

在这项工作中，我们提出了一种新的途径，通过为量子纠错程序提供形式语法和语义，并将问题归约为多项式优化，从而对解码器的行为进行形式化推理。我们的视角首次为量子纠错解码器定义了鲁棒性问题。未来工作的一个有前景的方向，是在我们初步探索的基础上进一步研究鲁棒性问题。例如，开发能够处理超出超矩形范围的约束集的鲁棒性问题求解技术，或开发具有鲁棒性保证的解码算法，借鉴机器学习中的鲁棒训练思路，都会很有价值 [12]。

数据可用性声明

我们的论文附有一项软件制品，其中包含基准测试、算法实现，以及用于运行实验和生成图表的脚本。我们将提交打包在 Docker 镜像中的该制品，以供评估。初步版本可通过此网址获取：https://anonymous.4open.science/r/analyzing-decoders-8034/。

参考文献

[1] Scott Aaronson 和 Daniel Gottesman。2004年。改进稳定子电路的模拟。《物理评论 A》70（2004年11月），052328 5 10 1103 70 052328 。期号 。doi: 。/PhysRevA. 。Aws Albarghouthi、Loris D'Antoni、Samuel [2] Drews 和 Aditya V. Nori。2017年。FairSquare：程序公平性的概率验证。《ACM编程语言会议录》1，OOPSLA，文章 80（2017年10月），30 页。doi:10.1145/3133904[3] Johannes Bausch、Andrew Senior、Francisco Heras、Thomas Edlich、Alex Davies、Michael Newman、Cody Jones、Kevin Satzinger、Yuezhen Niu、Sam Blackwell、George Holland、Dvir Kafri、Juan Atalaya、Craig Gidney、Demis Hassabis、Sergio Boixo、Hartmut Neven 和 Pushmeet Kohli。2024年。为量子处理器学习高准确度错误解码。《自然》 635 11 2024 834 840 10 1038 41586 024 08148 8 ( )，– 。doi: 。/s - - - Lucas Berent、Lukas Burgholzer、Peter-Jan [4] H.S. Derks、Jens Eisert 和 Robert Wille。2024年。使用 MaxSAT 解码量子颜色码。Quantum 8 2024 1506 10 22331 2024 10 23 1506 （2024年10月），。doi: 。/q- - - - Michael E. [5] Beverland、Malcolm Carroll、Andrew W. Cross 和 Theodore J. Yoder。2025年。快速失败：探测量子纠错中稀有事件的技术。arXiv: 2511 15177 [ ] 2511 15177 。量子物理 [6] https://arxiv.org/abs/ 。Sergey Bravyi 和 2013 Alexander Vargo。。量子纠错中稀有事件的模拟。《物理评论 A》 88, 6 2013 10 1103 88 062308 （2024年12月）。doi: 。/physreva. 。M. Carroll、S. Rosenblatt、P. Jurcevic、[7] I. Lauer 和 A. Kandala。2022年。超导量子比特弛豫时间的动态。《npj 量子信息》8，1（2022年），132。doi:10.1038/s41534-022-00643-y [8] Christophe Chareton、Sébastien Bardin、François Bobot、Valentin Perrelle 和 Benoît Valiron。2021年。电路构建量子程序的自动化演绎验证框架。载于《编程语言与系统：第30届欧洲编程研讨会，ESOP 2021，作为2021年欧洲软件理论与实践联合会议 ETAPS 的一部分举行，卢森堡卢森堡市，2021年3月27日至4月1日，论文集》（卢森堡卢森堡市）。施普林格出版社，柏林，海德堡，148 177 10 1007 978 3 030 72019 3_6 – 。doi: 。/ - - - - Christophe [9] Chareton、Sébastien Bardin、François Bobot、Valentin Perrelle 和 Benoît Valiron。。2021 电路构建量子程序的自动化演绎验证框架。载于《编程语言与系统：第30届欧洲编程研讨会，ESOP 2021，作为2021年欧洲软件理论与实践联合会议 ETAPS 的一部分举行，卢森堡卢森堡市，2021年3月27日至4月1日，论文集》（卢森堡卢森堡市）。施普林格出版社，柏林，海德堡，148 177 10 1007 978 3 030 72019 3_6 – 。doi: 。/ - - - - Marco F. [10] Cusumano-Towner、Feras A. Saad、Alexander K. Lew 和 Vikash K. Mansinghka。。2019 Gen：一种具备可编程推断功能的通用概率编程系统。载于《第40届 ACM SIGPLAN 编程设计与实现大会论文集》。美国计算机协会，纽约，NY，221–236。doi:10.1145/3314221.3314642[11] Luc De Raedt、Angelika Kimmig 和 Hannu Toivonen。2007年。Problog：一种概率 Prolog 及其在链接发现中的应用。载于《第二十届人工智能国际联合会议论文集》（印度海得拉巴）

（IJCAI'07）。Morgan Kaufmann Publishers Inc.，美国加利福尼亚州旧金山，2468–2473。[12] John C. Duchi 和 Hongseok Namkoong。2021年。通过分布鲁棒优化学习具有一致性能的模型。《统计年刊》 49, 3 6 2021 1378 1406 10 1214 20 2004 ( )，–。doi: 。/-AOS Wang Fang 和 Mingsheng Ying。。2024 [13] 量子纠错程序的符号执行。《ACM 编程语言会议录》 8, 2024 1040 1065 10 1145 3656419 编程语言设计与实现会议（2024年6月），–。doi: 。/ Alex Fischer 和 Akimasa Miyake。。2024 [14] 泡利噪声下表面码解码的困难性结果。Quantum 8 （2024 年 10 月 2024 1511 10 22331 2024 10 28 1511），。doi: 。/q- - - Austin G. Fowler。。匹配的有限表面码阈值证明。《物理评论快报》 109（2012年11月），180502。期号 18 10 1103 109 180502。doi: 。/PhysRevLett. 。Austin G. Fowler、Matteo Mariantoni、John M. Martinis 和 Andrew N. Cleland。2012年。表面码：迈向实用的大规模量子计算。《物理评论 A》 86 2012 032324 3 10 1103 86 032324 （9月），。期号 。doi: 。/ PhysRevA. 。Craig Gidney。。Stim：一种快速稳定子电路模拟器。Quantum（7月），。doi: 。/q- - - 06 497 - Craig Gidney。。如何用少于一百万个含噪量子比特分解 RSA 整数。arXiv: 。量子物理] 2505 15917 [2505 15917 https://arxiv.org/abs/ 。Craig Gidney、Michael Newman、Peter Brooks 和 Cody Jones。2025年 16，1 14 2025 4498 10 1038 41467 025 59714。耦合表面码。《自然·通讯》（5月），。doi: 。/s - - [19] 1 Noah D. Goodman、Vikash K. Mansinghka、Daniel Roy、Keith Bonawitz 和 Joshua B. Tenenbaum。2008年。Church：一种用于生成模型的语言。载于《第二十四届人工智能不确定性会议论文集》（芬兰赫尔辛基）（UAI'08）。AUAI出版社，阿灵顿，弗吉尼亚州，美国，220–229。[21] Google Quantum AI 及合作者。2024年。低于表面码阈值的量子纠错。《自然》（2024年12月）。doi:10.1038/s41586-024-08449-y [22] 1997 5 1997 9705052 Daniel Gottesman。。稳定子码与量子纠错。（）。arXiv:quant-ph/ Kesha Hietala、Robert Rand、Shih-Han Hung、Xiaodi Wu 和 Michael Hicks。。用于量子电路的经验证优化器。《ACM编程语言会议录》 5, 37 2021 29 10 1145 3434318 编程语言原理会议，文章（1月），页。doi: 。/ Oscar Higgott。。PyMatching：一个使用最小权完美匹配解码量子码的 Python软件包。arXiv: 2105 13082 [ ] 2105 13082 。量子物理 https://arxiv.org/abs/ 。Timo Hillmann、Lucas Berent、Armanda Ottaviano Quintavalle、Jens Eisert、Robert Wille 和 Joschka Roffe。2024年。量子低密度奇偶校验码的局部化统计解码。《自然·通讯》16（2024年）。https: //api.semanticscholar.org/CorpusID:[26] Qifan Huang、Li Zhou、Wang Fang、Mengyu Zhao 和 Mingsheng Ying。2025年。量子纠错程序的高效形式化验证。《ACM编程语言会议录》 9, 190 2025 26 10 1145 3729293 编程语言设计与实现会议，文章（6月），页。doi: 。/ Pavithran Iyer 和 David Poulin。。量子稳定子码解码的困难性。《IEEE信息论汇刊》（2024年9月 2015 5209 5223 10 1109 2015 2422294），–。doi: 。/TIT. 。Dexter Kozen。。概率程序语义。《计算机与系统科学杂志》 22，3（1981年6月），328–350。doi:10.1016/ 0022 0000 81 90036 2 -( )- Tor Lattimore 和 Csaba Szepesvári。2020年。《赌博机算法》。剑桥大学出版社。[30] Junyi Liu、Bohua Zhan、Shuling Wang、Shenggang Ying、Tao Liu、Yangjia Li、Mingsheng Ying 和 Naijun Zhan。2019年。使用量子霍尔逻辑对量子算法进行形式化验证。载于《计算机辅助验证》，Isil Dillig 和 Serdar Tasiran（编）。施普林格国际出版公司，尚，187–207。[31] Junyi Liu、Bohua Zhan、Shuling Wang、Shenggang Ying、Tao Liu、Yangjia Li、Mingsheng Ying 和 Naijun Zhan。2019年。使用量子霍尔逻辑对量子算法进行形式化验证。载于《计算机辅助验证》，Isil Dillig 和 Serdar Tasiran（编）。施普林格国际出版公司，尚，187–207。[32] Carolyn Mayer、Anand Ganti、Uzoma Onunkwo、Tzvetan Metodi、Benjamin Anker 和 Jacek Skryzalin。2025年。量子纠错电路的稀有事件模拟。arXiv: 2509 13678 [ ] 2509 13678 。量子物理 https://arxiv.org/abs/ 。Michael Mills、Jonathan Sedlacek、Tim Peterson、Sara Campbell、Jacob Johansen、Joan Dreiling 和 David Francois。2022年。太阳对囚禁离子量子计算机的影响。载于《2022年 IEEE量子计算与工程国际会议（量子计算与工程会议）》。 879 881 10 1109 53715 2022 00151 –。doi: 。/QCE 。。Cameron Moy、Jack Czenszak、John M. Li、Brianna Marshall 和 Steven Holtzen。2025年。Roulette：一种表达力强、精确且高效的离散概率编程语言。《ACM编程语言会议录》9，编程语言设计与实现会议，文章 231（2025年6月 2025 25 10 1145 3729334），页。doi: 。/ Tristan Müller、Thomas Alexander、Michael E. Beverland、Markus Bühler、Blake R. Johnson、Thilo Maurer 和 Drew Vandeth。2025年。改进的置信传播足以实现量子存储器的实时解码。arXiv: 2506 01779 [ ] 2506 01779 。量子物理 https://arxiv.org/abs/ 。Pavel Panteleev 和 Gleb Kalachev。。具有良好有限长度性能的简并量子低密度奇偶校验码。Quantum 5 2021 585 10 22331 2021 11 22 585（11月），。doi: 。/q- - - Gerardo Rubino 和 Bruno Tuffin。。使用蒙特卡罗方法进行稀有事件模拟。Wiley出版社。

[38] Adrian Sampson、Pavel Panchekha、Todd Mytkowicz、Kathryn S. McKinley、Dan Grossman 和 Luis Ceze。2014年。表达和验证概率断言。载于 第*35*届美国计算机协会 *SIGPLAN* 编程语言设计与实现会议（*PLDI '14*）论文集。美国计算机协会，纽约，纽约州，美国，112–122页。 doi:10.1145/2594291.2594294[39] Andrew W. Senior、Thomas Edlich、Francisco J. H. Heras、Lei M. Zhang、Oscar Higgott、James S. Spencer、Taylor Applebaum、Sam Blackwell、Justin Ledford、Akvil˙e Žemgulyt˙e、Augustin Žídek、Noah Shutty、Andrew Cowie、Yin Li、George Holland、Peter Brooks、Charlie Beattie、Michael Newman、Alex Davies、Cody Jones、Sergio Boixo、Hartmut Neven、Pushmeet Kohli 和 Johannes Bausch。2025年。面向拓扑量子码的可扩展实时神经解码器。arXiv: 2512 07737 [ ] 2512 07737 。量子物理 https://arxiv.org/abs/ 。John Zhuoyang Ye 和 Jens Palsberg。[40] 。量子纠错的可扩展测试。arXiv：2602.04921 [量子物理] 2026 https://arxiv.org/abs/ 2602 04921 。Mingsheng Ying。 。量子程序的 Floyd–Hoare逻辑。《*ACM*编程语言理论与系统汇刊》33，6，[41]文章19（1月 2012 2012 49 10 1145 2049706 2049708 ），页数。doi: 。 / 。Satoshi Yoshida、Ethan Lake 和 Hayata Yamasaki。[42]2026年。并查集解码器有限阈值的证明。arXiv: 2602 20238 [ ] 2602 20238 。量子物理https://arxiv.org/abs/ 。Li Zhou、Gilles Barthe、Pierre-Yves Strub、Junyi Liu 和 Mingsheng Ying。[43]2023年。CoqQ：量子程序的基础验证。《*ACM*编程语言会议录》7，编程语言原理会议，文章29（2023年1月），33页。doi:10.1145/3571222

证明 A

定理5.4。该定理的证明在符号表达上较为密集，但从概念上讲，只是对第5节中所阐述推理的直接形式化。我们从单个错误比特串的概率开始。

引理 A.1。对于量子纠错程序 $P(v)$ 和错误比特串 e，有 $f(e;\ P,\ \mathcal{D}) = m_e(v)$

证明。首先，回顾一下，随机抽样空间 δ 的选择恰好使其能够对程序 $P(v_1,\ \ldots\ldots,\ v_i)$ 中各个伯努利错误信道的概率进行建模，也就是说，对于每个参数 v_i:

$$\frac{|\sigma \in \delta : E_i(\sigma_i) \neq I|}{|\delta|} = v_i.$$

由此可得

$$\frac{|\sigma \in \delta : E_i(\sigma_i) = I|}{|\delta|} = 1 - v_i.$$

接下来请注意，我们可以将错误比特串的概率分解为各个错误信道的概率。

$$f(e; P, \mathcal{D}) = \frac{|\{\Sigma \in \mathcal{D} : e_\Sigma = e\}|}{|\mathcal{D}|} = \prod_{e_i=1} \frac{|\sigma \in \delta : E_i(\sigma_i) \neq I|}{|\delta|} \prod_{e_i=0} \frac{|\sigma \in \delta : E_i(\sigma_i) = I|}{|\delta|}.$$

这里，D 与之前一样，指的是 n 抽取序列的空间，D $= \delta^n$. 结合这些事实，我们得到

$$f(e; P, \mathcal{D}) = \prod_{e_i=1} v_i \prod_{e_i=0} (1 - v_i).$$

但这恰好是在点 v 处对 e 的错误最小项求值，从而完成证明。□

有了这个引理，我们就可以继续证明该定理。从定义 3.1 出发，我们将准确性改写为关于错误比特串的概率

$$\mathrm{Err}(P, d) = \Pr_{(s,o)\sim P}[d(s) \neq o] = \Pr_{e\sim P}[d(\mathrm{synd}(e)) \neq \mathrm{obs}(e)] = \sum_{e\in L} f(e; P, \mathcal{D})$$

然后，我们应用该引理，得到等式

$$\mathrm{Err}(P, d) = \sum_{e\in L} m_e(\overline{v}).$$

这里右侧是 $p_L(v)$ 的定义，因此证明完毕。

定理 5.5。这直接由定理5.4得出。

定理 5.6。注意到，误差多项式在这个空间上为正（这与其作为概率的解释一致）。因此，通过注意到 $p_L = p_{L\cap S} + p_{L\cap S^c} \geq p_{L\cap S}$，我们得到第一个不等式。对于第二个不等式，我们还利用了这样一个事实：$p_L = 1 - p_{L^c}$，然后应用相同的推理：$p_{L^c} = p_{S\setminus L} + p_{S^c\setminus L} \geq p_{S\setminus L}$，这意味着 $p_L = 1 - p_{L^c} \leq 1 - p_{S\setminus L}$。

推论 5.7。由定理 5.6 以及简单的观测值可得：$\max_{x\in X}(1 - x)$ $=1 - \min_{x\in X}(x)$。

定理 6.2。 设 r是超矩形的一个内点，因此存在某个 i使得 $r_i \notin \{\ell_i, u_i\}$。 考虑在 r处的偏导数 $\frac{\partial p}{\partial x_i} = a(1 - x_i) + b(x_i)$。如果 $a > b$，则令 r'等于 r，但 $r_i = \ell_i$，并注意到 $p(r') > p(r)$。 否则，选择 $r'_i = u_i$，且我们有 $p(r') \geq p(r)$。 反复应用这一论证可知，通过内点得到的 p的任何取值，至少都能在某个角点处达到。

定理 6.7. 设 v为通过此过程得到的顶点， v^* 为达到全局最优值的顶点。首先请注意，v^* 在以下索引处不可能与 v不同： i对于这些索引，我们已证明在整个集合上斜率为正或负。若确实不同，则特别地， $\frac{\partial p}{\partial x_i}(v^*)$ 会具有相应的符号，因此我们可以为 x_i 选择另一个值，从而增大 p的值。然后，由于我们对剩余变量进行了穷举搜索， v 所选的值必定不逊于任何其他选择，包括 v^*所选的值。

定理 7.1. 我们从 Chernoff 界的 Kullback-Leibler 表述开始：

$$\Pr_\theta[\hat{\Theta} \geq u] \leq e^{-N \cdot KL(u||\theta)} \qquad (u \in [\theta, 1]) \tag{1}$$

$$\Pr_\theta[\hat{\Theta} \leq u] \leq e^{-N \cdot KL(u||\theta)} \qquad (u \in [0, \theta]) \tag{2}$$

$$\text{where } KL(p||q) = p \ln\left(\frac{p}{q}\right) + (1-p)\ln\left(\frac{1-p}{1-q}\right).$$

我们的目标是对这个界求逆，从而构造一个基于点估计量 Θ 的 $1-\alpha$置信区间 $[L, U]$，用于估计 θ。

注意，对于固定的 θ，函数 $u \mapsto KL(u||\theta)$ 在区间 $(0, \theta]$ 上严格递减，在 $[\theta, 1)$ 上严格递增，并且在 θ 处取值为 0。因此，对于任意 $t > 0$，存在唯一的值 $u_t^- \in [0, \theta]$ 和 $u_t^+ \in [\theta, 1]$，使得

$$N \cdot KL(u_t^-||\theta) = t, \qquad N \cdot KL(u_t^+||\theta) = t. \tag{3}$$

此外，该函数的单调性意味着

$$\left\{N \cdot KL(\hat{\Theta}||\theta) \geq t\right\} \subseteq \left\{\hat{\Theta} \leq u_t^-\right\} \cup \left\{\hat{\Theta} \geq u_t^+\right\}. \tag{4}$$

因此

$$\begin{aligned}
\Pr_\theta\left[N \cdot KL(\hat{\Theta}||\theta) \geq t\right] &\leq \Pr_\theta\left[\left\{\hat{\Theta} \leq u_t^-\right\} \cup \left\{\hat{\Theta} \geq u_t^+\right\}\right] && \text{(Eq. (4) and monotonicity of measure)} \\
&\leq \Pr_\theta\left[\hat{\Theta} \leq u_t^-\right] + \Pr\left[\hat{\Theta} \geq u_t^+\right] && \text{(union bound)} \\
&\leq e^{-N \cdot KL(u_t^-||\theta)} + e^{-N \cdot KL(u_t^+||\theta)} && \text{(Eqs. (1) and (2))} \\
&= 2e^{-t} && \text{(Eq. (3)).}
\end{aligned}$$

选择 $t_\alpha := \ln(2/\alpha)$，使得 $2e^{-t_\alpha} = \alpha$。那么

$$\Pr_\theta\left[N \cdot KL(\hat{\Theta}||\theta) \leq t_\alpha\right] \geq 1 - \alpha.$$

此配置给出了一个 $1-\alpha$置信区间 $[L, U]$，其中

$$L := \inf\left\{u \in [0, \hat{\Theta}] : KL(\hat{\Theta}||u) \leq \frac{1}{N}(\ln(2/\alpha))\right\},$$

$$U := \sup\left\{u \in [\hat{\Theta}, 1] : KL(\hat{\Theta}||u) \leq \frac{1}{N}(\ln(2/\alpha))\right\}.$$

在事件 $0 < \hat{\Theta} < 1$上， $u \mapsto KL(\hat{\Theta}||u)$函数的单调性保证 L 和 $U\hat{}$是达到等号的唯一取值。如果 $\Theta = 0$ ，那么 $L = 0$ 和

$$KL(0||u) = \ln(1/(1-u)) = 1/N \ln(2/\alpha) \implies U = 1 - (\alpha/2)^{1/N}. \tag{5}$$

$\hat{}$若 $\Theta = 1$ 则 $U = 1$ 且

$$KL(1||u) = \ln(1/u) = 1/N \ln(2/\alpha) \implies L = (\alpha/2)^{1/N}. \tag{6}$$