arXiv:2601.03343v1 [quant-ph] 6 Jan 2026

优化容错猫态制备

Tom Peham, [1, *] Erik Weilandt, [1, †] 和 Robert Wille[1, 2, ‡]
[1]慕尼黑工业大学设计自动化 *кафедра*，德国
[2]慕尼黑量子软件公司，德国

猫态是容错量子计算的重要资源，可作为各种容错原语的构建模块。因此，能够在较大的故障距离下制备高质量猫态至关重要。虽然已有针对较低故障距离或较少量子比特数的优化方法，但更高的故障距离可以通过广义构造实现，不过其电路规模可能并非最优。在本工作中，我们提出一种猫态制备方案：先用低深度电路制备两个猫态，再对其中一个态进行横向 CNOT 和测量。该方案能够容错地制备 w 量子比特猫态，故障距离最高可达 9，深度为 $\lceil \log_2 w \rceil + 1$，且最多使用 $3w - 2$ 个受控非门和 $2w$ 个量子比特。我们指出，这种构造中组合复杂度最高的部分是精确确定横向 CNOT 的连线，并提出三种确定方法：两种基于理论可满足性求解，另一种则是基于局部修复策略的启发式搜索。数值评估表明，与广义构造相比，我们的电路在使用更少资源的同时实现了较高的容错距离。

I. 引言

量子纠错（QEC）和容错量子计算（FTQC）[1–4]为迈向实用规模量子计算开辟了道路；在实用规模量子计算中，需要可靠地对编码量子信息执行数十亿次操作[5]。通过将信息冗余编码到量子纠错码中，并反复提取错误综合征，逻辑量子比特的寿命可以延长到远超物理量子比特所能达到的水平。近期实验进展表明，当前硬件平台已具备容错量子计算所需的能力，例如实时、低于阈值的解码 [6], 编码通用门集 [7], 以及首批编码算法[8]。

要实现大规模容错量子计算，计算的每一步都必须以容错方式实现，从而避免物理错误在系统中失控地传播。与其从整个算法的层面看待容错，通常的做法是将计算分解为子电路或构件 [9], ，并确保每个子电路或构件本身都具备容错能力。若通过解码和纠错迅速消除错误，它们就不会累积，信息也能以较高概率得到保护。从这一视角来看，容错量子计算中的一项核心电路设计任务，是为综合征提取 [10–13], 、辅助态制备 [14–16], 、魔态制备 [17–19], 、码切换 [20, 21], 或无需测量的协议 [22, 23], 等任务设计并优化容错构件。尽管此类电路通常由人工设计，或针对特定编码实例设计，但针对容错电路构造开发通用的自动化编译与优化技术，已开始受到越来越多的关注 [14, 15, 24–27]。

尽管容错技术已有初步演示，最先进的量子硬件仍受到严格的资源限制 [6, 8, 28–30]。这意味着，要在近期实现成功的容错量子计算，就必须仔细优化计算过程中使用的容错电路。即使是频繁重复使用相同子电路的大规模计算，也能从电路优化中显著获益，因为对一个构件的任何优化都会立即带来成倍的资源节省。

本工作聚焦于优化特定资源态的容错制备电路，这类资源态称为“猫态”，是形如 $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes w} + |1\rangle^{\otimes w})$ 的 w 量子比特态（也称为 Greenberg-Horn-Zeilinger态 [31]），或形如 $\frac{1}{\sqrt{2}}(|+\rangle^{\otimes w} + |-\rangle^{\otimes w})$ 的态。这类态是容错量子计算的一种有用资源，因为它们可用于综合征提取 [2, 32, 33], 魔态制备 [17],或容错编码器 [34]。

鉴于这些态在概念上简单且实用，针对这类态的容错电路设计已经得到研究也就不足为奇了。文献 [11] 表明，距离为三的猫态制备可以通过对数级开销实现，但制备具有更高故障距离的猫态时，开销会增加。文献 [26] 给出了一种适用于任意规模猫态的构造，该构造所需深度为对数级，双量子比特门和量子比特数量为线性对数级。另一种方法是采用旗标容错构造，因为任何更高距离的旗标容错综合征测量方案都可以转化为一种容错猫态制备方案，其最坏情况下的深度、双量子比特门数量和量子比特开销均为线性级 [10, 16]。

在本工作中，我们旨在进一步优化容错猫态制备的资源需求。我们提出了一种容错制备猫态的方案：使用两个（大小可能不同的）猫态，将错误从一个态复制到另一个态，测量辅助态，并在测量结果无错误时进行后选择——

* tom.peham@tum.de
† erik.weilandt@tum.de
‡ robert.wille@tum.de

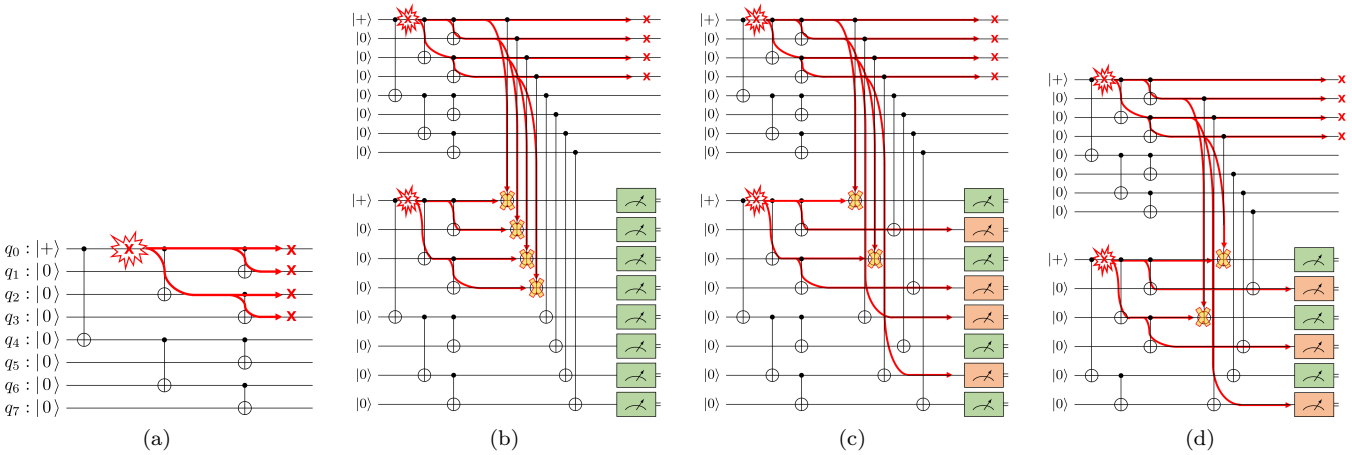


图1. 容错猫态制备。**(a)** 8量子比特猫态的非容错对数深度制备。单个权重为1的错误会传播为制备态上的权重为 4 的错误。**(b)** 通过将数据猫态中的 X 错误复制到辅助猫态，并在 Z 基中进行测量，可以检测数据猫态中的错误。这里并不具备容错性，因为有两个错误未被检测到，并会在数据量子比特上导致权重为 4 的错误。**(c)** 通过改变横向 CNOT 的连接方式，错误检测组件可以针对4个错误实现容错。**(d)** 使用更小的辅助猫态可以减少量子比特开销；具有适当连接方式的横向 CNOT 配合6量子比特猫态即可满足要求。

量结果（见图 1）。该构造以对数深度和线性数量的双量子比特门及量子比特制备猫态。该构造的难点在于找到特定的量子比特置换，以避免错误相互抵消，从而违反制备方案的容错性。

我们的主要贡献如下：

- 我们提出了一种容错猫态制备方案，该方案使用两个低深度猫态、单层横向 CNOT 和后选择。该方案能够以对数级 CNOT深度以及量子比特和双量子比特门方面的线性开销，生成 w量子比特猫态。

- 我们提出了多种组合搜索方法，用于确定合适的横向受控非门以确保容错，包括基于CEGAR的搜索、SAT/SMT形式化以及启发式局部修复策略。

- 我们应用这些方法，合成了适用于最多 49 个量子比特、容错距离为 4，以及最多 19 个量子比特、容错距离为 9 的容错猫态制备电路。

- 我们使用电路级噪声模拟评估了所构造的电路。与现有的通用构造相比，所得电路所需的量子比特和门开销更低，接受率和错误率也有所改善。

本工作的其余部分结构如下。第二节回顾猫态和容错所需的背景知识。在第三节中，我们介绍本文研究的容错猫态制备方案并形式化（部分）横向 CNOT 保持容错性的条件。第四节回顾现有的容错猫态制备构造。在第五节中，我们分析平衡树猫态制备电路的错误传播结构，并展示如何利用该结构限制容错构造的搜索空间，以及高效检查给定候选CNOT的容错性。第六节介绍用于综合容错横向受控非门的组合搜索方法。最后，第七节评估所得电路，并由第八节总结全文。

II. 背景

在本工作中，我们使用 $[i] = \{1,2,\ldots,i\} \subseteq \mathbb{Z}$。泡利矩阵定义如下

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

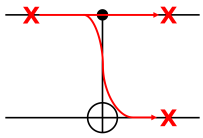
它们共同构成单量子比特泡利群，该群自然扩展为多量子比特上的泡利群。在本工作中，我们考虑纯 X-型或纯 Z- 型泡利算符，即形式为 $\prod_{i=1}^{w} \sigma_i$ 的 w 量子比特算符，其中 $\sigma_i \in \{I, X\}$ 或 $\sigma_i \in \{I, Z\}$。

我们将重点讨论 w 个量子比特 GHZ 态的制备，即形如 $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes w} + |1\rangle^{\otimes w})$ 的态。 $\frac{1}{\sqrt{2}}(|\mp\rangle^{\otimes w} + |-\rangle^{\otimes w})$ 的制备方式与此类似。GHZ 态是由稳定子 $X^{\otimes n}$ 和 $Z_i \otimes Z_{i+1}$，$i \in [w-1]$ 定义的 w 量子比特稳定子态。

任何支撑集大小为奇数的 Z 类型算符都与 $X^{\otimes n}$ 算符反对易，而任意 X 类型

算符都至少与一个 Z 稳定子反对易。因此，可以检测多达 $w-1$ X 个（比特翻转）错误，以及作用于该态的任意奇数个 Z （相位翻转）错误。由于任何双量子比特 Z 错误都是稳定子，任何 Z 错误都等价于一个平凡错误或单量子比特错误的稳定子。

故障是在电路执行过程中发生的错误事件，这些故障可以在电路中传播。 X-故障会沿 CNOT 门的控制端传播，即如果控制量子比特发生翻转，目标量子比特也会翻转。



错误是故障传播到电路末端后的结果。我们将任意 w 量子比特的 X 错误表示为一个 $\mathbb{F}_2^w$ 向量，其中该向量仅在与错误支撑集对应的索引处具有非零项，例如，向量 $[1,0,1]$ 对应于算符 XIX。对于 GHZ 态，任意 X 错误 $e \in \mathbb{F}_2^w$ 都等价于通过翻转 e 中的所有比特得到的错误$\mathbb{1}^{\otimes w}+e$，因为 $X^{\otimes w}$ 是该态的稳定子。猫态上的错误 e （记为 wt e）的<依样保留结构? >权重定义为 e 或 $\mathbb{1}+e$ 中非零项的数量，取两者中较小者。

电路中可能同时发生多个故障，从而导致多个错误。如果两个故障分别导致错误 e_1 和e_2，则合并后的错误为 e_1+e_2。电路中至多发生 t个故障所导致的错误集合记作 $\mathcal{C}$ 上的 $\mathcal{E}_t(\mathcal{C})$，我们称之为电路 $\mathcal{C}$ 的（t-）故障集 。如果上下文已明确所指电路，我们通常只将 t-故障集记作 $\mathcal{E}_t$ 。

电路中的错误可以通过某些不会扰动态的测量来检测。

定义 1. 如果猫态制备方案能够检测到导致最终猫态上权重大于t的错误$e \in \mathcal{E}_t$，且该错误由电路中至多 t 个故障的任意组合引起，那么我们称该方案容错至多 t 个错误，FT^t。

III. 通过横向受控非门验证猫态

由于猫态是 CSS 态，因此可以先将量子比特初始化为 $|+\rangle$，再接上受控非门电 [15, 35]路。具体而言，由于猫态只有一个 X型稳定子，因此将一个量子比特制备为 $|+\rangle$，其余量子比特制备为 $|0\rangle$。接着，受控非门电路必须将初始的 X稳定子传播到其他量子比特上。使用“平衡树”受控非门电路即可在 $\lceil \log_2 w \rceil$ 深度内完成，如图 1a所示。每一层都通过将当前态与数量相同的 $|0\rangle$ 量子比特纠缠，使猫态的规模翻倍。如果 w 不是 2 的幂，最后一层只会纠缠 $2^{\lceil \log_2 w \rceil}-w$ 个量子比特。我们将这种电路结构称为$\mathcal{B}_w$。

一般而言，这种制备甚至不是容错的[1]：电路中控制端上的单个错误会传播为整个子树上的 X错误，如使用电路8制备 量子比特猫态所示$\mathcal{B}_8$，见图 1a。

通常，可以使用辅助猫态检测这类高权重传播错误：通过横向受控非门复制错误，并在 Z 基底下测量辅助猫态。横向受控非门对猫态的作用是平凡的，因为“控制”猫态的 $X^{\otimes w}$ 稳定子会传播为辅助猫态的稳定子；对于 Z 型稳定子，情况也类似。假设辅助猫态没有错误，复制错误会按照该错误的支撑集翻转猫态中的比特。没有错误时，在 Z 基底下测量辅助猫态，结果为比特串 0^w 或1^w。因此，测量结果中任何翻转的比特都表明存在错误。

然而，如果辅助猫态并非无错误，电路中的低权重故障可能导致未被检测到的高权重错误。特别是，如果两个猫态均使用相同的电路结构制备，那么在两个电路中相同位置发生的任意两个故障，都会在横向 CNOT 之后相互抵消。

示例 1。 考虑图1b中的电路。该电路使用电路 $\mathcal{B}_8$制备两个八量子比特猫态。通过横向 CNOT，将前八个量子比特中的错误复制到后八个量子比特。在所示情形中，故障 $X_1X_2X_3X_4$ 被复制到辅助猫态中，但由于在辅助猫态的制备过程中发生了相同的故障，该故障仍未被检测到。由于制备所得猫态上的残余故障权重为4，因此图 1中的猫态制备并非容错[2]。

通过置换横向 CNOT 的连线，可以避免错误抵消。由于对猫态的量子比特进行任意置换都不会改变其状态，因此，带有 $\mathrm{CX}_{i,\ \sigma(w)}$ 且置换 σ 属于对称群$\mathcal{S}_w$ 的任意横向 CNOT，在将错误从一个状态复制到另一个状态的同时，对猫态起到平凡作用。这一事实可以用来使错误检测具备容错性。直观地说，只要能够确保高权重故障在量子比特置换后不会相互抵消，也不会合并为稳定子 $X^{\otimes w}$，所有有问题的错误都将被检测到。

使用横向 CNOT检测猫态中的错误，优点是只需在制备过程中额外增加一层受控非门。如果使用 $\mathcal{B}_w$制备这些态，那么制备所需的 CNOT深度为 $\lceil \log_2 w \rceil + 1$，使用 $3w-2$ 个受控非门，并对 w 个量子比特进行后选择。

示例 2. 按照图 1b连接横向 CNOT

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 5 & 3 & 7 & 2 & 6 & 8 & 4 \end{pmatrix},$$

防止故障相互抵消。事实上，在这种构造中，多达 $t<4$ 个故障的任何组合都不会相互抵消

，从而导致权重大于 t 的错误未被检测到。因此，所得电路是 FT[4]。

通过使用更小的辅助系统，还可以进一步减少受控非门和量子比特开销。如果辅助态具有 $1 < w' < w$ 个量子比特，那么第一条电路和第二条电路中任意 w' 个量子比特之间的横向 CNOT 在该系统上的作用都是平凡的。我们可以根据它如何连接第一条和第二条电路中的量子比特，来刻画这种部分横向 CNOT。具体而言，它由一组 CNOT 对给出

$$\overline{\mathrm{CX}} \subseteq [w] \times [w']$$

其中 $|\overline{\mathrm{CX}}| = w'$，使得任一索引在任一侧出现的次数都不超过一次。等价地，CX 在 w' 个辅助量子比特与 $\overline{w'}$ 个数据量子比特的某个子集之间定义了一个双射。这种 CNOT 关联着一个 映射 $f: \mathbb{F}_2^w \to \mathbb{F}_2^{w'}$，它将 w 个量子比特上的错误映射为 w' 个量子比特上的错误，并在第 i 个单量子比特错误 $\hat{e}_i$ 上定义为

$$f(\hat{e}_i) = \begin{cases} \hat{e}_j \text{ if } (i,j) \in \overline{\mathrm{CX}} \\ \mathbf{0} \text{ else} \end{cases}$$

仍然必须通过恰当地连接两个猫态中的量子比特来避免错误抵消。为了定义这种猫态制备电路具有 FT^t 的含义，我们引入以下定义。

定义 2. 设$\mathcal{C}$,$\mathcal{C}'$ 是分别作用于 w 个量子比特和 $w' \leq w$ 个量子比特的两个猫态制备电路。具有相关映射f: $\mathbb{F}_2^w \to \mathbb{F}_2^{w'}$ 的（部分）横向受控非门 CX 若满足以下条件，则称其为对 $T > 0 \in \mathbb{Z}$ 而言容错的（或 *FT*）：对于所有满足 的 , $t' \geq 0$, 都不存在

$$e \in \mathcal{E}_t(\mathcal{C}), \qquad e' \in \mathcal{E}_{t'}(\mathcal{C}')$$

使得组合错误

$$f(e) + e' \in \{I^{\otimes w'}, X^{\otimes w'}\}$$

而

$$\mathrm{wt}(e) > T.$$

示例 3. 再次考虑FT[4] 的 8量子比特猫态制备电路，如图 1c 所示。事实证明，通过选择部分横向 CNOT，只需一个6量子比特辅助态即可使该构造FT[4]

$$\{(2,3),(3,6),(4,1),(6,5),(7,4),(8,2)\} \tag{1}$$

所得电路如图 1d 所示。 6 个量子比特也是使该构造 FT[4]所需的最小辅助态。

本工作的目标正是利用（部分）横向受控非门构造此类容错猫态制备方案。注意，电路规模仅取决于辅助态的大小。由于我们希望将资源开销降至最低，即深度、受控非门数量和后选择开销，因此要找到可能的量子比特数量最少的辅助量子比特。

在讨论如何解决这一优化问题之前，我们先简要回顾相关的容错猫态制备构造。

IV. 相关工作

表 I. 不同容错^t w量子比特猫态制备方法的资源扩展。

方法	门数 深度		量子比特
本方法	$O(w)$	$\boldsymbol{O(\text{日志 } w)}$	$\boldsymbol{O(w)}$
参考文献 [26]	$O(w \text{ 对数 } w)$	$\boldsymbol{O(\text{对数 } w)}$	$O(w \text{ 对数 } w)$
参考文献 [10, 16]	$\boldsymbol{O(w)}$	$O(w)$	$\boldsymbol{O(w)}$

最近，参考文献 [26] 提出了一种基于 ZX 图上严格保持距离的重写规则的 w-容错猫态制备方案 [36, 37]。该构造通过递归地从 $w/2$ 个量子比特的猫态构造 w 个量子比特的猫态，直至达到 4 个量子比特的基例；对此，采用一种简单构造，使用五个量子比特、五个 CNOT 门和一次测量。递归组合通过测量大小为$w/2$的猫态中成对量子比特之间的 $w/2$ XX个奇偶校验来实现。因此，每个组合步骤需要 $w/2$个额外量子比特、 w 个 CNOT 门，并对 $w/2$次测量进行后选择。总的来说，如果 w 是 2 的幂，该方法需要 $\frac{w}{2}\log_2 w + \frac{w}{4}$ 个量子比特、 $w\log_2 w - \frac{3}{4}w$ 个受控非门，深度为 $2\log_2 w + 1$。如果所需的猫态制备应为 FT^t 且满足 $t < w$，那么要求会更加宽松，因为递归组合中最多只需进行 t 个奇偶校验。

另一种值得注意的构造基于标志故障容错测量 [10]。基于标志的构造通过检测作用于辅助量子比特的成对受控非门之间的错误来工作；在没有错误时，这些受控非门对系统的作用是平凡的。此方法需要使用深度为 $w-1$ 的电路初始化辅助量子比特，其中初始化为 $|+\rangle$ 的量子比特充当所有受控非门的控制量子比特。最近，参考文献 [16] 提出了一种寻找最优错误检测标志组件的方法。虽然该构造是在任意 CSS 码的态制备背景下提出的，但也可用于猫态制备。作者成功找到了可容忍多达 $t = 5$ 个故障、涉及多达 29 个量子比特的电路。因此，最多可实现 10 量子比特猫态的完全容错初始化（因为我们希望防护多达 $t = 5$ 个故障）。在这种情况下，最优构造需要 16 个量子比特（10 个用于猫态，6 个用于标志位）、21 个受控非门（9 个用于猫态制备，12 个用于标志位构造），且具有线性深度。与所提出的横向构造相比，该构造针对$w = 10$ 需要后选择的测量次数更少，但深度开销要高得多。此外，我们旨在

构造更大的猫态，这需要超越 $t = 5$ 的容错能力。参考文献 [10] 还给出了针对这种基于标志的线性猫态制备的一般 w 标志位构造，需要 $2w - 1$ 个带复位的量子比特（ $3w - 1$ 个不带复位的量子比特）， $3w - 3$ 个受控非门，深度为 $7w - 15$ （对于更大的w）。这意味着，除线性深度开销外，这种一般构造几乎与横向猫态制备相当。

资源扩展的简要比较见表 I。这表明，在渐近意义上，使用横向验证制备猫态，在所有考察的指标上都最为有利。在第七节 中我们将看到，这不仅是渐近优势；即使在量子比特数很少的情况下，所提出的猫态制备方法在几乎所有情形下所需资源也少于参考文献 [26]中的递归构造。不过，这些改进的前提是，此类t-FT 置换确实可以构造出来。

使用多个状态副本和横向操作进行容错态制备，是一种广为人知的构造 [38–40]。事实上，我们的研究设定与参考文献 [39] 和参考文献 [38] 非常相似；这些工作通过构造多个状态副本并寻找合适的横向 CNOT 置换，实现了量子 Reed–Muller 码和 Golay 码的逻辑基态的容错制备。这些工作所研究的编码并不允许量子比特进行所有可能的置换，限制更多。此外，与我们的情况不同，这些构造中的辅助态是固定的，因为无法使用更小的状态进行验证。这两项工作中的构造仅适用于距离为 7 和 9 的编码，其故障距离分别为 3 和 4。我们的目标是构造具有很高故障距离且尽可能少用量子比特的电路。为此，我们需要开发高效的组合搜索算法，以构造所需的猫态制备电路。在接下来的章节中，我们将概述如何实现这一目标。

V. 构造容错猫态制备电路

给定两个猫态制备电路$\mathcal{C}$、$\mathcal{C}'$ ，其规模为w， w' 且 $w' \leq w$，任务是构造一个部分横向 CNOT，使其满足定义 2所述的容错要求。由于所有 w' 辅助态的量子比特在此构造中均作为目标，因此横向 CNOT 可由以下内容刻画：

- 在数据猫态上选择 $w\prime$ 个控制量子比特，以及

•对辅助猫态上的目标量子比特进行置换。

只有这两种选择的某些组合能够避免导致容错失效的错误抵消。由于可能的故障组合数量会随着故障距离迅速增加，而且联合搜索空间的大小为 $\binom{w}{w'}w'!$，极其庞大，因此我们依靠制备电路的结构特性来指导构造。

A. 制备电路$\mathcal{B}_w$的故障结构

使用平衡树制备电路$\mathcal{B}_w$ 的一项关键优势是其故障结构高度规整。在$\mathcal{B}_w$ 中，控制操作之前发生的任何单个 X 错误都会传播到其下方的整个二叉子树。例如，对于 $w = 8$，该电路的 1-故障集为

$$\begin{aligned}\mathcal{E}_1 = \{&X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8,\\ &X_1X_2, X_3X_4, X_5X_6, X_7X_8,\\ &X_1X_2X_3X_4, X_5X_6X_7X_8\},\end{aligned}$$

一般而言，$\mathcal{E}_1(\mathcal{B}_w)$由所有支撑在 $[w]$的二进区间上的 X-错误组成，因此其大小为 $2w - 2$。高阶故障集$\mathcal{E}_t(\mathcal{B}_w)$则由这些错误模式经过 $\mathbb{F}_2$相加而形成。由于结构规则，许多此类组合会彼此重合，从而避免在内存中存储大量冗余元素。尽管如此， t-故障集的大小仍会随 t呈指数增长。

故障集结构至关重要，原因有二：

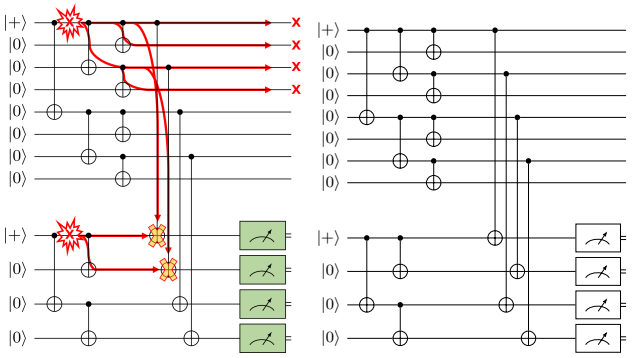
1. 它使我们能够推断哪些错误可以传播到辅助量子比特并在那里相互抵消；以及2. 它能够高效地检查候选部分横向 CNOT 是否具备容错性。

B. 确定 w' 并选择控制量子比特

下文中，我们假设$\mathcal{C}$ 和$\mathcal{C}'$ 均使用平衡树电路$\mathcal{B}_w$ （或 $\mathcal{B}_{w'}$）实现，如第三节所述。在搜索辅助目标量子比特的排列之前，我们必须确定所需辅助量子比特的数量，以及哪些数据量子比特应作为控制项。一般而言，这很难确定，但我们可以利用可能故障的结构缩小搜索范围。

我们先考虑最简单的情况，即构造容错[2] 猫态制备电路。如果辅助量子比特少于 $w/2$ 个，那么在数据电路最后一层的同一个受控非门中参与运算的两个量子比特将不会与辅助量子比特相连，从而使某些权重为2 的传播错误完全无法检测。因此 $w' \geq w/2$ 是必要的，事实上也已经足够。若有 $w/2$ 个控制项，则每个大小为2 的子树都与辅助量子比特相连，并且只要确保整个子树不会映射到整个子树，就能避免图 2中所示的抵消现象。

子树的大小也可用于确定所需辅助态的最小尺寸。例如，考虑一个包含四个量子比特的子树，这意味着一个错误就可能在



（a）非容错[2] 置换。两个故障导致重量为四的数据故障。

（b）容错[2] 置换。

图 2. 容错[2] 猫态制备。

数据量子比特上导致一个权重为 4 的错误。如果只有一个或两个量子比特与辅助态相连，那么最多只需两个单量子比特错误，就足以抵消辅助态上的复制错误。这完全与目标置换无关。因此，要实现容错[3]，，这些数据量子比特中至少有三个必须与辅助态相连。由于大小为四的子树也必须满足这一条件，我们知道容错[3] 猫态制备要求辅助态的大小至少为 $(3/4)\,w$。

这一论证也可以直接推广到更大的子树。如果有一个大小为八的子树，那么容错[7] 制备要求其中至少 7 个量子比特与辅助态相连，因此，容错[7] 制备（足够大的）猫态至少需要一个 $(7/8)w$ 量子比特辅助量子比特。

考虑传播的辅助故障后，可能需要稍大的辅助量子比特规模，但这些下界大大简化了对辅助量子比特规模和控制量子比特可能取值的搜索。

C. 检查固定控制项集合的容错性

选定候选辅助量子比特规模 w' 和一组控制位置后，剩下的任务是寻找 w' 个辅助量子比特的一种置换，使其得到容错的 t 横向 CNOT。为了高效搜索各种置换，我们需要一种快速的容错性检查方法。

令 $\pi : \mathbb{F}_2^w \to \mathbb{F}_2^{w'}$ 表示错误在选定的控制量子比特上的投影。对于每个 k，我们预先计算投影故障集

$$\mathcal{P}_k = \pi(\mathcal{E}_k).$$

$\mathcal{E}_k$ 中的许多错误可能具有相同的投影，因此对于每种投影模式 $e' \in \mathcal{P}_k$，我们只保留一个权重最大的代表

$$\mathcal{R}_k = \{\arg\max_{e \in \pi^{-1}(e')} \mathrm{wt}(e) \mid e' \in \mathcal{P}_k\}.$$

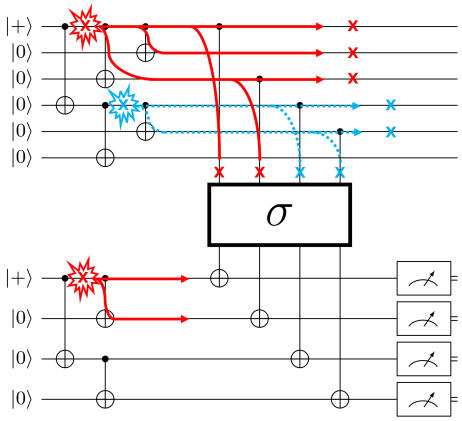


图 3. 容错性检查：预先计算辅助量子比特上的错误能够与数据量子比特上的最坏情况错误相抵消的情形，从而根据置换 σ 快速检查容错性。

这些代表恰好涵盖了可能损害容错能力的错误模式。仅存储 $\mathcal{R}_k$可显著降低内存开销，因为$\mathcal{E}_k$ 中的许多错误投影结果相同，而我们只保留每种投影对应的最坏情况原像。

An 个错误 $e \in \mathcal{R}_k$ 违反容错性 precis 只有当

- 少于 $\mathrm{wt}(e) - k - 1$ 个故障出现在$\mathcal{C}_\prime$ 中时，才足以抵消 $\pi(e)$，并且
- 仍允许发生这么多个故障，即 $\mathrm{wt}(e) - k - 1 \le t - k$。

这意味着，如果至多有 $\pi(e)$ 个故障可以抵消，则容错会遭到破坏

$$h_k(e) := \min(t - k,\ \mathrm{wt}(e) - k - 1)$$

辅助量子比特上的故障。

因此，我们预先计算每个 $e \in \mathcal{R}_k$ 对应的坏映像集合：

$$\mathrm{Bad}(e) = \{e' \in \bigcup_{k'=1}^{h_k(e)} \mathcal{E}_{k'}(\mathcal{C}') \mid \mathrm{wt}(e') = \mathrm{wt}(e)\},$$

哪些是辅助态中的错误，它们可能会在置换下抵消投影后的数据错误。

给定此集合，检查置换 σ 是否容错^t ，归结为验证

$$\sigma(\pi(e)) \notin \mathrm{Bad}(e) \quad \text{for all } k < t \text{ and all } e \in \mathcal{R}_k.$$

只要我们能够存储每个 k 对应的集合$\mathcal{R}_k$ ，以及这些集合中每个错误对应的 Bad(e)，就可以针对每个错误高效地检查置换是否容错。

示例 4。考虑图3所示的 $w = 6$ 量子比特猫态的制备图3。这里，量子比特 0、2、3和4 已被选为控制量子比特。该故障

该6量子比特制备电路的集合（相差一个稳定子）为

$$\{[0,0,0,0,0,1],[0,0,0,0,1,0],[0,0,0,1,0,0],\\ [0,0,1,0,0,0],[0,1,0,0,0,0],[1,0,0,0,0,0],\\ [1,1,0,0,0,0],[0,0,0,1,1,0],[1,1,1,0,0,0]\}.$$

将其投影到控制项上，得到投影故障集

$$\{[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1],[1,1,0,0]\}.$$

这个投影集中的唯一高权重故障是$[1,1,0,0],$，它是$[0,0,0,1,1,0]$（在图3中以蓝色绘制）的投影，或是$[1,1,1,0,0,0]$（在图3中以红色绘制）的投影。这意味着在这种情况下，$\mathcal{R}_1$ 仅仅是 $\{[1,1,1,0,0,0]\}$。$[1,1,1,0,0,0]$ 的投影权重为二，因此，坏映像是辅助量子比特制备电路的单量子比特故障集中的所有权重为二的故障：

$$\mathrm{Bad}([1,1,1,0,0,0]) = \{[1,1,0,0],[0,0,1,1]\}.$$

这样，我们就可以快速检查使制备过程满足 FT^2 的置换 σ 候选项。例如，置换

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \end{pmatrix}$$

不起作用，因为 σ 将 $[1,1,1,0,0,0]$ 投影到$[1,1,0,0] \in \mathrm{Bad}([1,1,1,0,0,0])$。另一方面

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 2 & 1 & 3 \end{pmatrix}$$

确实起作用。

有了缩小控制项选择范围的方法，以及一种高效检验候选置换容错性的方法，现在我们来着手实际构造 FT^t 部分横向受控非门。

VI. 组合搜索方法

虽然我们特别关注采用平衡二叉树结构的容错猫态制备电路，但本节介绍的方法适用于任意猫态制备电路 $\mathcal{C}$，$\mathcal{C}'$。所需的唯一信息是两个猫态制备电路的单量子比特故障集$\mathcal{E}_1(\mathcal{C})$ 和$\mathcal{E}_1(\mathcal{C}')$，以及目标故障距离 t。

虽然电路结构$\mathcal{B}_w$ 具有酉猫态制备电路所能达到的最低深度，但额外的约束（例如硬件约束）可能会要求采用其他制备电路。事实上，任意量子比特二叉树都可以转化为猫态制备电路，其单量子比特故障集可直接由真子树集合给出。

基于这些单量子比特故障集合，我们可以尝试搜索容错部分CNOT门。

A. 使用可满足性模理论直接搜索目标排列

对于固定的 w' 量子比特数据猫态中的控制位置，任务是确定这些控制项如何连接到 w量子比特辅助猫态。我们将此问题编码为一个关于整数变量的可满足实例，并使用 SMT 求解器 w'求解 [41, 42]。

线路连接关系由符号排列$\sigma = (\sigma_1, \ldots, \sigma_{w'})$表示，其中每个变量 σ_i 的取值属于$[w']$，并指定第 i个控制量子比特所指向的辅助量子比特。为确保每个辅助量子比特恰好被指向一次，元组$(\sigma_1, \ldots, \sigma_{w'})$被约束为一个双射：

$$\bigwedge_{i\in[w']} \bigwedge_{j\in[w']\setminus\{i\}} \sigma_i \neq \sigma_j.$$

现在我们来说明控制量子比特上的投影错误模式如何在这样的置换中传播。任何投影误差都表示为比特串 $e \in \{0,1\}^{w'}$，其支撑集为 $S(e) = \{i \mid e_i = 1\}$，权重为$|e|$。给定排列变量$\sigma$的一个赋值后，辅助量子比特上的诱导错误是另一个比特串$e^\sigma \in \{0,1\}^{w'}$，其各项定义如下：

$$e^\sigma_j \quad \Leftrightarrow \quad \bigvee_{i\in S(e)} (\sigma_i = j), \qquad j = 1, \ldots, w'.$$

也就是说，只要 e 的支撑集中至少有一个控制量子比特连接到第 j 个辅助量子比特，该辅助量子比特就会受到影响。由于 σ 是一个置换，这种映射保持汉明重量：

$$\sum_{j=1}^{w'} e^\sigma_j = |e|.$$

对于每个错误 e，令 $\mathrm{Bad}(e) \subseteq \{0,1\}^{w'}$ 表示第 V C 节中定义的坏映像集合，其中每个映像的汉明重量均为 $|e|$。对于每个 $e' \in \mathrm{Bad}(e)$，我们添加子句

$$\bigvee_{j=1}^{w'} (e'_j \neq e^\sigma_j),$$

该子句禁止求解器将 e 映射到 $\mathrm{Bad}(e)$ 中的任何映像。

这种编码的缺点是，必须预先编码所有被禁止的传播，即使只需考虑其中一小部分传播故障，就足以约束置换的选择。此外，这种编码特定于控制量子比特的选择，因此可能需要针对不同的控制量子比特选择求解多个实例。

B. 使用 CEGAR 同时搜索控制量子比特和目标排列

现在，我们将介绍一种结合的精确方法，它同时搜索从 w' 数据量子比特中选择 w' 控制量子比特的方案

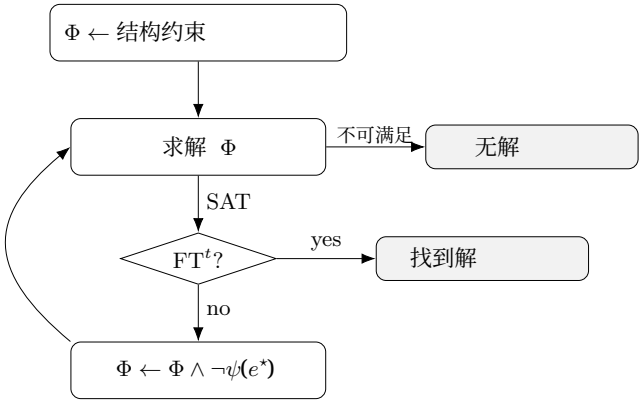


图 4. 使用 CEGAR 为猫态制备构造 FT^t 部分横向受控非门。初始 SAT 公式 Φ通过迭代添加阻塞子句来细化，这些子句用于排除反例，直至判定其可满足性。

以及连接到 w' 辅助量子比特的布线方式，而无需预先编码所有故障传播。由于由此产生的搜索空间极其庞大，我们采用反例引导的抽象精化（CEGAR）策略。

CEGAR [43] 是一种迭代方法：首先求解问题的粗略抽象，然后检查候选解是否满足完整规范。如果发现反例，则利用它细化抽象，添加能够排除违规行为的约束。这个求解–检查–细化循环会不断重复，直到找到有效解或证明其不可满足。

在我们的设定中，初始抽象仅包含结构约束：

- 恰好选择 w 个数据量子比特作为控制项，并且
- 所选控制项与 w个辅助量子比特之间存在双射连接。

对于每个数据量子比特 $q \in [w]$，我们引入一个布尔变量 ctrl_q，用于指示是否将 q 选为控制项；再引入一个整数变量 $\mathrm{trgt}_q \in \{1,\dots,w'\}$，用于指定 q 在 $\mathrm{ctrl}_q = 1$时所指向的辅助量子比特。通过以下约束确保得到有效的部分横向 CNOT：

$$\sum_{q=1}^{w} \mathrm{ctrl}_q = w',$$

从而确保恰好选择 w' 个控制项，并且

$$\bigwedge_{u<v} \big(\mathrm{ctrl}_u = 0 \ \vee \ \mathrm{ctrl}_v = 0 \ \vee \ \mathrm{trgt}_u \neq \mathrm{trgt}_v\big),$$

从而强制要求任意两个选定的控制量子比特不能连接到同一个辅助目标量子比特。

给定一个候选赋值（ctrl，trgt），我们构造相应的部分横向 CNOT CX 及其诱导错误映射 f，并检查所得构造是否为 FT^t。如果未发现违规，搜索将终止，并得到一个 FT^t 部分横向 CNOT。

如果检查器识别出一个反例，即一个投影误差 e 其诱导映像为 $f(e) \in \mathrm{Bad}(e)$[1]在当前赋值 (ctrl,trgt)下，我们通过添加一个阻塞子句来细化抽象，从而排除此特定赋值。

具体而言，对于每个辅助量子比特索引 $j = 1,\dots,w'$，我们引入布尔公式

$$b_j(e) \ := \bigvee_{\substack{q\in[w]\\ \mathrm{ctrl}_q=1,\, e_q=1}} \big(\mathrm{trgt}_q = j\big),$$

当且仅当该投影误差在辅助量子比特 j 上具有支撑时，该公式的值为真。为防止该投影再次出现，我们添加子句

$$\bigvee_{j=1}^{w'} \big(f(e)_j \neq b_j(e)\big),$$

从而排除当前赋值。

随着发现新的反例，这一细化过程会不断累积阻塞子句，逐步缩小搜索空间，直到找到 FT^t 部分横向 CNOT，或确定其不可满足性。

这种 CEGAR 表述的优势在于，它无需预先编码所有禁用投影，而是仅学习消除已观察到的反例所需的约束。不过，在某些情况下，可能需要大量细化才能实现收敛。在这类情况下，固定控制量子比特的选择并直接搜索容错布线可能更有优势，正如上一节所讨论的那样。

C. 启发式局部搜索

除了精确的 SMT 编码之外，我们还采用一种启发式局部搜索策略，直接在辅助量子比特排列的搜索空间中运行。请记住，对于固定的 w' 控制位置，任务是找到 w' 辅助量子比特的一种置换，使得任何错误 e 都不会被投影为 $\mathrm{Bad}(e)$ 中的错误。

该过程从对 w' 辅助量子比特位置进行随机排列开始。对于给定的排列，我们会检查任意投影误差的置换是否属于预计算的坏映像 $\mathrm{Bad}(e)$。如果存在这样的情况 e，我们会尝试调整排列，使 e不再构成违规。这可以通过交换

[1] 由于控制量子比特的选择与布线是联合确定的，因此无法完全预先计算 $\mathrm{Bad}(e)$。实践中，按需计算并缓存这些集合效果很好，因为通常能较早找到有效的控制选择。

算法 1： 用于容错的启发式搜索部分 CNOT

输入： 作用于 w 量子比特的故障集$\mathcal{E}$，禁用像对于所有 $e \in \mathcal{E}$，Bad(e)，最大迭代次数 N_{iter}，一个投影 $\pi : \mathbb{F}_2^w \to \mathbb{F}_2^{w'}$

输出： 置换 σ，使整体 CNOT FT^t 或失败

$\sigma \leftarrow$ 随机排列;
对 $k = 1$ **到** N_{iter} **do**
  查找 $e \in \mathcal{E}$ 使得 $\sigma(\pi(e)) \in$Bad(e);
  如果 没有这样的 e**则**
    返回 σ;
  ones $\leftarrow$ supp($\pi(e)$);
  零 $\leftarrow [w']\setminus$ supp($\pi(e)$);
  当 $\sigma(\pi(e)) \in$ Bad(e) **执行**
    随机选择 $i \in$ ones;
    随机选择 $j \in$zeros;
    $\sigma \leftarrow \sigma \circ (i\,j)$;
返回失败;

e 支撑集中的量子比特与支撑集外的量子比特来实现。请注意，这并不能保证算法收敛，但在实践中，通常能迅速消除冲突。如果在尝试消除违规时搜索陷入停滞，则会偶尔执行随机置换，希望打破搜索循环。

该算法的伪代码见 算法 1。这一随机修复过程会持续进行，直到找到有效置换或达到最大迭代次数。虽然该方法无法保证完备性——可能存在它未能发现的有效置换——但在系统规模适中的情况下，它在实践中往往能够成功。

这种方法的一个主要挑战是，修复某个错误 e并不能保证此前修复的冲突仍然得到满足。换句话说，消除 $f(e) \in \text{Bad}(e)$ 的交换，可能会给另一个错误 e'引入新的违规。我们的方法受到 Moser 和 Tardos 的重采样算法 [44],的启发。该算法最初是作为 Lovász 局部引理的构造性证明提出的；该引理描述了在什么条件下，针对这类罕见组合对象的搜索过程能够成功。尽管该引理的形式化条件在我们的场景中并不成立，但同样的随机修复思路在实践中往往行之有效。反复交换并偶尔进行随机扰动，通常能在迭代次数耗尽之前将排列调整为全局有效的配置。

D. 三种搜索方法的比较

表 II 概述了所提出的每种方法的优点和缺点。CEGAR 方法最为通用，因为它不预先假定横向 CNOT 的控制量子比特选择。它可以证明某个

表 II. 三种搜索策略的比较。

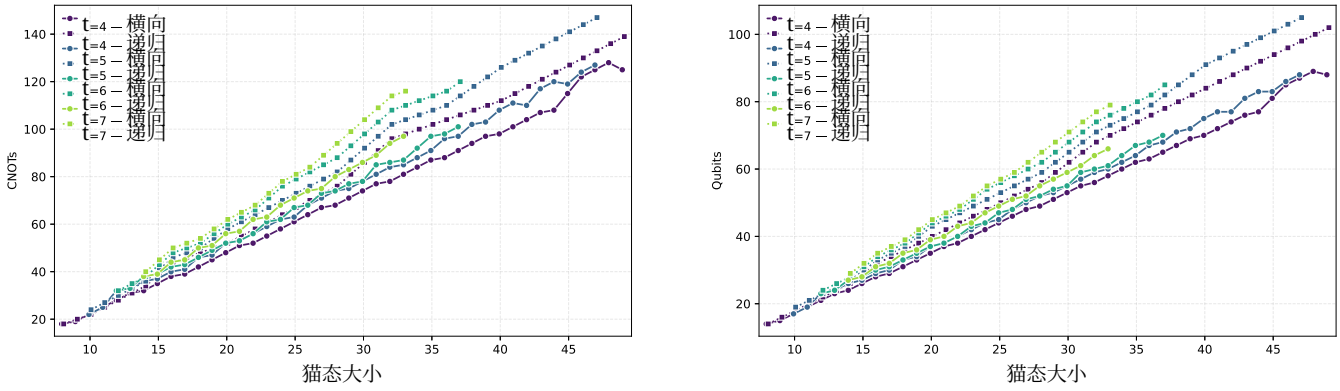
方法（范围）摘要	
SAT/SMT 置换	**优点：** 精确；通常能快速证明不可满足；禁止错误的直接编码。 **缺点：** 必须针对许多控制项重复执行控制集；通常在 SAT 实例上速度最慢。
CEGAR 控制项 + 置换	**优点：** 可证明可满足/不可满足；细化可保持公式规模较小。 **缺点：** 可能需要多次细化才能p证明不可满足。
局部搜索 置换	**优点：** 速度快；通常能找到有效的置换在实践中很快找到。 **缺点：** 启发式方法；不具备完备性，也不保证判断为不可满足g保证。

给定 w'的容错部分横截 CNOT 是否存在，并且通常比直接 SMT 编码更快地求解 SAT 实例。另一方面，由于所需迭代次数较多，不可满足实例通常求解速度较慢。采用固定控制项的 SAT/SMT 编码适用范围较窄，但如果一个实例是不可满足的，这种方法在实践中通常能够非常快速地证明这一点。如果能够缩小不同控制量子比特选择的数量范围，这种直接编码对于不可满足实例会非常有效。局部搜索启发式方法迄今为止速度最快，而且通常能够迅速成功，但它无法证明不可满足性，即使存在容错 CNOT，也可能无法找到。

在实践中，我们将这些方法结合起来。求解器首先尝试联合 CEGAR 搜索；只有在搜索超时的情况下，才会遍历控制选择，并结合局部搜索应用 SAT/SMT 编码。由于直接 SMT 编码能够快速处理不可满足的实例，我们用它快速筛除无效的控制项选择，从而避免在实例不可满足时，启发式搜索仍不必要地遍历排列。这种分层策略兼顾了通用性、精确性和速度：首先尝试能力最强的方法，而在精确方法耗尽时间后，则将更快但可靠性较低的方法作为后备方案。

七、评估

针对FT^t 横向受控非门的搜索算法已使用 Python 实现，并用于构造最多包含 49 个量子比特、故障距离最高达9的容错猫态制备方案。本节将展示所提方案的资源开销和电路级模拟结果，并与文献 [26]中的通用递归猫态制备构造进行比较。Python 实现及所有生成的电路均可通过开源 MQT QECC 仓库获取；该仓库隶属于慕尼黑量子工具包 [45]。



(a) 受控非门数量 (b) 量子比特数量

图 5. 不同猫态大小和故障距离下猫态制备电路的电路指标比较。“横向”对应所提出的构造，“递归”对应参考文献 [26] 中的构造。

A. 仿真设置

这些电路是在运行 Ubuntu 22.04 的不同机器上生成的，所需内存最多为 128GB。

使用开源稳定子模拟器 Stim 对电路进行仿真 [46]，采用以下由单个噪声参数控制的噪声模型 p [2]：

- 所有双量子比特门之后均施加强度为 p 的双量子比特退极化噪声。
- 所有初始化均以概率 $2/3$ p发生翻转。
- 所有测量均以概率 $2/3$ p发生翻转。

对于所提出的横向验证方案，我们模拟整个电路，并在辅助量子比特的全 0 或全-1 Z-基测量结果上进行后选择。如果辅助量子比特上未检测到错误，则无故障地测量数据，以获得比特串 $b \in \{0,1\}^w$。设 n 为模拟的总采样次数，n_{acc} 为接受的采样次数，B 为接受采样所测得的数据比特串集合。根据这些数据，我们估计接受率 R_{acc} 以及数据上有 k 个比特发生翻转的概率 P_k，其表达式为

$$R_{\text{acc}} = \frac{n_{\text{acc}}}{n}$$
$$P_k = \sum_{b \in B} [\![(\sum_{i \in w} b_i) = k]\!],$$

其中，$b_i \in \{0,1\}$ 是比特串 b 的第 i位，而$[\![\cdot]\!]$ 表示艾弗森括号，它将布尔值解释为整数。

递归猫态构造的模拟以类似的方式进行。

[2] 虽然只有比特翻转噪声与我们的研究相关，但我们采用电路级噪声模拟中常用的标准噪声模型，以便更容易将我们的模拟结果与其他设置进行比较。

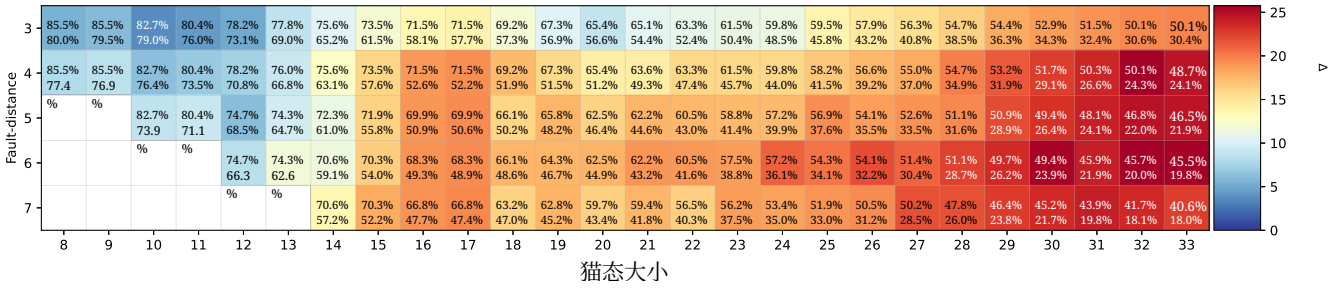
B. 讨论

图中展示了根据容错地制备特定大小的数据猫态所需的 CNOT门数量和量子比特数量，对生成的电路进行的比较图 5。我们观察到，在这两个方面，所提出的横向构造都比递归构造使用更少的资源。对于更大的猫态大小，最多可以节省 20 个 CNOT 门和量子比特。

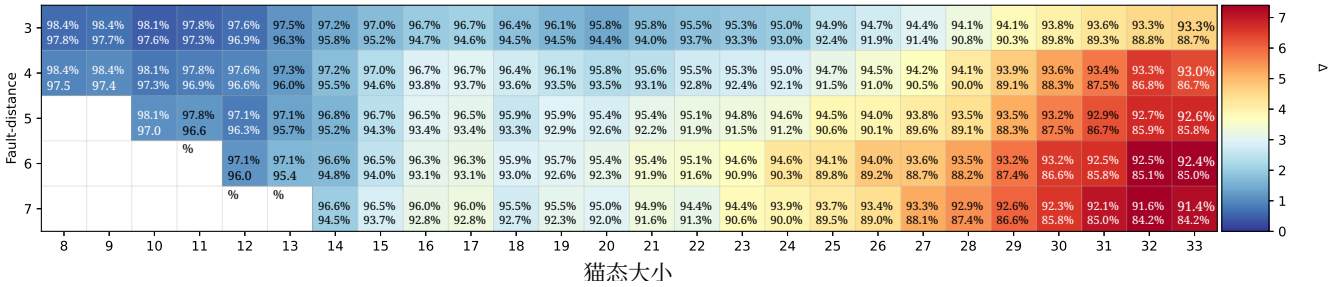
这也体现为接受率的提升，如图 6所示，其中展示了物理错误率为1%（图 6a）和0.1% （图 6b）时模拟运行的接受率。对于1% 这一较高的物理错误率，较高权重猫态的接受率差异最高可达25%；而即使物理错误率为6% ，较高权重猫态的接受率差异仍可高达0.1%。对于较低的错误率，这一差异则不太显著。

与接受率相比，在后选择后数据猫态上的错误差异更加明显。请考虑图 7中的错误率图，该图展示了FT^3猫态在错误率为 $p = 1\%$时的模拟错误率。由于这些电路的构造仅针对3 个错误具有容错性，我们预计观测到4 个错误的概率大致与观测到3个错误的概率相同。这也正是我们在递归构造的模拟中看到的结果。然而，横向验证的表现实际上要好得多。我们观察到 P_4 始终约比 P_3 低一个数量级。尽管对于较小的猫态， P_5 并没有比 P_4 低一个数量级，但与递归构造相比，其缩放表现仍相当有利。对于FT^4 （图 8）猫态，也可以观察到相同的现象：与横向构造相比，递归构造中出现5 次比特翻转的可能性更高。

这是因为，虽然横向构造无法保证检测出 FT{v1} 情况下所有有问题的

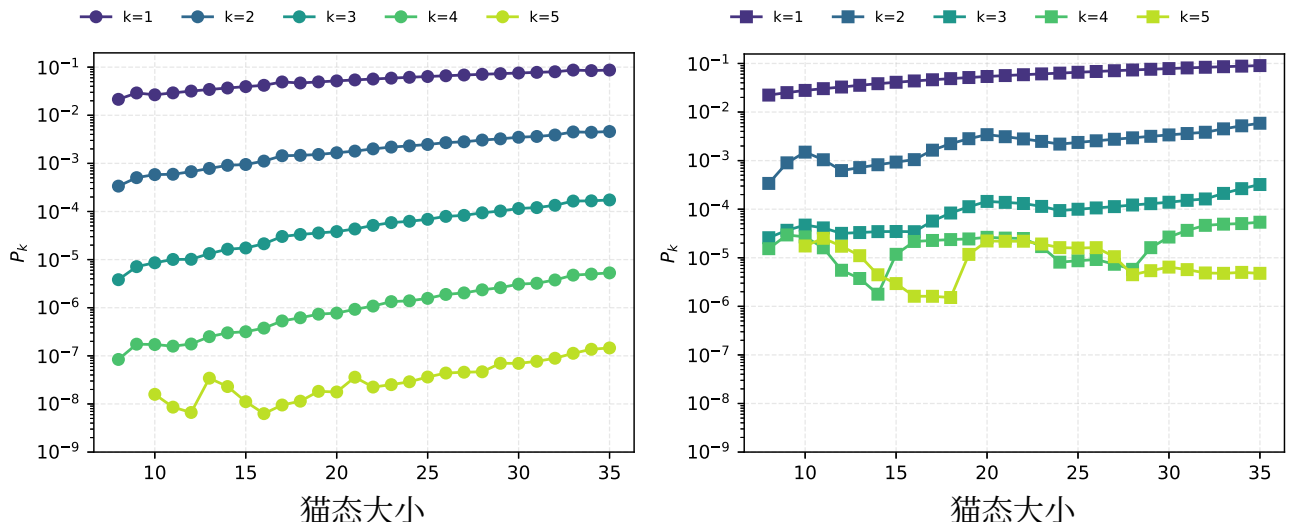


(a) 使用物理错误率为 $p = 1\%$ 进行模拟的接受率



(b) 使用物理错误率为 $p = 0.1\%$ 进行模拟的接受率

图 6. 猫态制备接受率的比较。“横向”对应所提出的构造，“递归”对应参考文献 [26] 中的构造。每个单元格分别显示横向（**上方**）和递归（**下方**）构造的绝对接受率。色调表示两种构造的接受率差异。



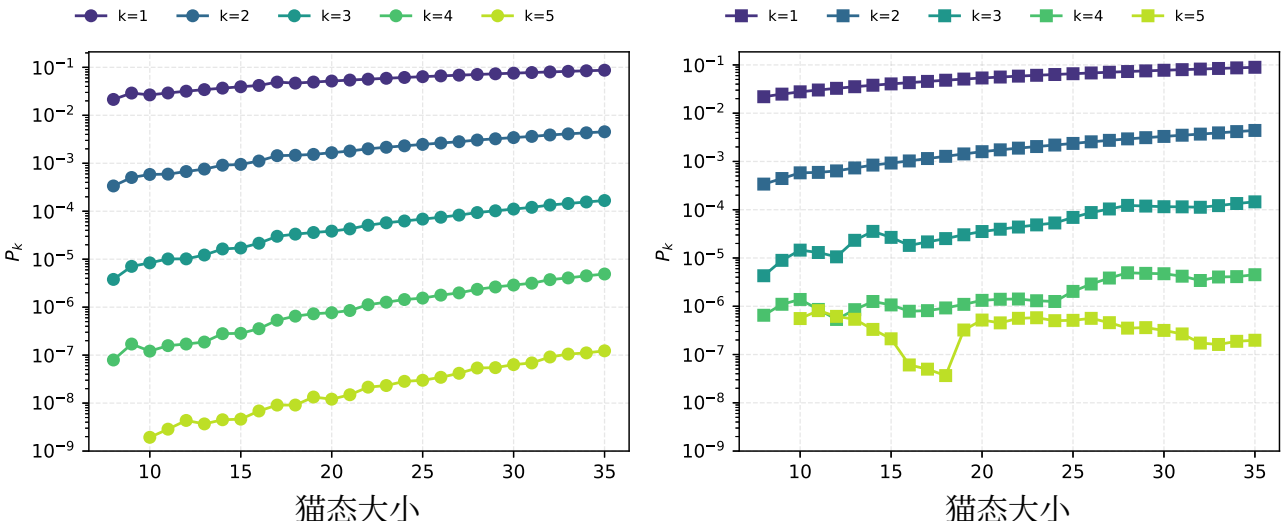
(a) 横向（本工作）　　(b) 递归 [26]

图7. FT^3 猫态制备在 $p = 1\%$时的错误率。 P_k 表示后选择后在数据上测得 k 次比特翻转的概率。

4 个错误组合，但只有极少数组合能够逃过验证。递归构造在使用 ZZ 测量将各个状态组合起来时，只检查每个状态的 3 个量子比特，因此可能会漏掉其他量子比特上的错误。

因此，即使在接受率方面的开销节省对于较小的故障距离，深度、量子比特数或受控非门的开销节省并不显著，横向构造仍可能提供质量更高的猫态。

构造的电路详情见附录 A。



（a）横向（本工作）　（b）递归 [26]

图8。在 $p = 1\%$条件下容错[4] 猫态制备的错误率。 P_k 表示后选择后在数据上测得 k 比特翻转的概率。

VIII. 结论

本工作提出了一种容错构造猫态的方法，该方法使用两个猫态、一个横向 CNOT 门和后选择。我们证明，这种构造的容错性取决于两个猫态与横向 CNOT 门之间的精确关联，并提出了三种寻找容错 CNOT 的不同方法。对于我们能够解决这一组合搜索问题的参数范围，我们证明，与参考文献 [26] 中的对数深度构造相比，本构造在深度、量子比特数量和 CNOT 开销方面表现更优。此外，电路级噪声模拟表明，与参考文献 [26]相比，本构造具有更高的接受率和更低的错误率。

虽然所构造的电路本身可能应用于某些 Shor 式综合征提取方案 [2] ，或作为其他量子纠错方案 [33], 的资源，但所提出的搜索方法或许也可用于寻找其他容错构造。尤其是，CEGAR 搜索可能对其他“暴力”容错构造有所帮助，因为它不需要预先对所有约束进行编码。

尽管本工作的研究方法与文献 [26] 截然不同——前者以大量计算取代了后者形式上的优雅——但二者相辅相成，因为本工作构造的电路可作为进一步开展容错构造的起点，用于容错重写。例如，为了构造更大的猫态，可以将本文所提出构造的优化电路作为文献中递归构造的一部分。

[26].

致谢

作者感谢 Lucas Berent 和 Ludwig Schmid 对稿件提出的宝贵意见。

作者感谢欧洲研究委员会（ERC）通过欧盟地平线2020研究与创新计划（拨款协议编号：101001318）以及 Millenion（拨款协议编号：101114305）提供的资助。本工作是慕尼黑量子谷项目的一部分，该项目由巴伐利亚州政府通过巴伐利亚高科技议程Plus提供资金支持。本工作还获得德国研究联合会（DFG，德国研究基金会，编号：563402549）的资助。

[1] D. Gottesman，容错量子计算理论，Phys. Rev. A **57**, 127（1998）。[2] P. Shor，容错量子计算，载于第37届计算机科学基础会议论文集（1996），第56–65页。[3] E. Dennis、A. Kitaev、A. Landahl 和 J. Preskill，拓扑量子存储器，《数学物理学杂志》**43**，4452（2002）。

[4] M. A. Nielsen 和 I. L. Chuang，《量子计算与量子信息》（剑桥大学出版社，2010）。[5] C. Gidney，如何用不到一百万个含噪量子比特分解 2048 位 RSA 整数（ 2025 2505 15917 ），arXiv: . [ ] 量子物理。[6] Google Quantum AI 和合作者，低于表面码阈值的量子错误

纠错，Nature 638, 920（2025）。[7] I. Pogorelov、F. Butt、L. Postler、C. D. Marciniak、P. Schindler、M. Müller 和 T. Monz，实验性容错码切换（2024），arXiv:2403.13732。[8] B. W. Reichardt、A. Paetznick、D. Aasen、I. Basov、J. M. Bello-Rivas、P. Bonderson、R. Chao、W. van Dam、M. B. Hastings、A. Paz、M. P. da Silva、A. Sundaram、K. M. Svore、A. Vaschillo、Z. Wang、M. Zanner、W. B. Cairncross、C.-A. Chen、D. Crow、H. Kim、J. M. Kin- dem、J. King、M. McDonald、M. A. Norcia、A. Ryou、M. Stone、L. Wadleigh、K. Barnes、P. Battaglino、T. C. Bohdanowicz、G. Booth、A. Brown、M. O. Brown、K. Cassella、R. Coxe、J. M. Epstein、M. Feldkamp、C. Griger、E. Halperin、A. Heinz、F. Hummel、M. Jaffe、A. M. W. Jones、E. Kapit、K. Kotru、J. Lauigan、M. Li、J. Marjanovic、E. Megidish、M. Meredith、R. Morshead、J. A. Muniz、S. Narayanaswami、C. Nishiguchi、T. Paule、K. A. Pawlak、K. L. Pudenz、D. R. Pérez、J. Simon、A. Smull、D. Stack、M. Urbanek、R. J. M. van de Veer- donk、Z. Vendeiro、R. T. Weverka、T. Wilkason、T.-Y. Wu、X. Xie、E. Zalys-Geller、X. Zhang 和 B. J. Bloom，利用中性原子量子处理器展示逻辑计算（2024），arXiv:2411.11822 [量子物理]。[9] D. Gottesman，《量子纠错与容错量子计算导论》（2009），arXiv:0904.2557 [量子物理]。[10] C. Chamberland 和 M. E. Beverland，适用于任意距离码的标志辅助容错纠错，Quantum 2, 53（2018）。[11] P. Prabhu 和 B. W. Reichardt，以更少量子比特实现容错综合征提取和猫态制备，Quantum 7, 1154（2023），arXiv:2108.02184 [量子物理]。[12] P.-H. Liou 和 C.-Y. Lai，利用多功能标志共享综合征提取电路降低量子纠错开销（2024），arXiv:2407.00607 [量子物理]。[13] P.-J. H. S. Derks、A. Townsend-Teague、A. G. Burchards 和 J. Eisert，使用检测器错误模型设计容错电路（2024），arXiv:2407.13826 [量子物理]。[14] R. Zen、J. Olle、L. Colmenarez、M. Puviani、M. Müller 和 F. Marquardt，利用强化学习发现用于容错逻辑态制备的量子电路，《物理评论X》15, 041012（2025）。[15] T. Peham、L. Schmid、L. Berent、M. Müller 和 R. Wille，自动化合成量子纠错码的容错态制备电路，PRX Quantum 6, 020330（2025）。[16] D. Forlivesi 和 D. Amaro，源于标志：一种用于 CSS 码的模块化容错制备方案（2025），arXiv:2508.14200 [量子物理]。[17] C. Chamberland 和 K. Noh，利用冗余辅助比特编码和标志量子比特，以极低开销实现容错魔态制备，《npj量子信息》6, 1（2020）。[18] C. Gidney、N. Shutty 和 C. Jones，魔态培育：以低于 CNOT 门的成本生成 T 态（2024），arXiv:2409.17595。[19] K. Sahay、P.-K. Tsai、K. Chang、Q. Su、T. B. Smith、S. Singh 和 S. Puri，折叠横向表面码培育（2025），arXiv:2509.05212 [量子物理]。[20] F. Butt、S. Heußen、M. Rispler 和 M. Müller，面向近期量子处理器的容错码切换协议，PRX Quantum 5, 020345（2024）。[21] S. Heußen 和 J. Hilder，利用单向横向 CNOT 门实现高效容错码切换

tum 9, 1846 (2025). [22] S. Heußen, D. F. Locher, and M. Müller, 近端设备中的无测量容错量子纠错，PRX Quantum 5, 010333 (2024). [23] H. Goto, Y. Ho, and T. Kanao, 一维量子比特阵列中距离三九量子比特表面码的无测量容错逻辑零态编码，Phys. Rev. Res. 5, 043137 (2023). [24] L. Schmid, T. Peham, L. Berent, M. Müller, and R. Wille, 近端量子纠错的确定性容错态制备：使用布尔可满足性进行自动综合（ 2025 2501 05527 ），[ ] arXiv: . 量子物理 . [25] N. Shutty and C. Chamberland, 使用理论模可满足性求解器解码合并的色码-表面码并寻找容错Clifford电路，Phys. Rev. Applied 18, 014072 (2022). [26] B. Rodatz, B. Poór, and A. Kissinger, 通过构造实现容错 (2025), arXiv:2506.17181 [量子物理]. [27] B. Rodatz, B. Poór, and A. Kissinger, 通过保距离重写将稳定子码Floquet化 (2024), arXiv:2410.17240 [量子物理]. [28] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. B. Ataides, N. Maskara, I. Cong, X. Gao, P. S. Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletic, and M. D. Lukin, 基于可重构原子阵列的逻辑量子处理器，Nature 626, 58 (2024). [29] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Steane量子纠错的容错演示，PRX Quantum 5, 030326 (2024). [30] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feld- ker, M. Meth, C. D. Marciniak, R. Stricker, M. Ring- bauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, 容错通用量子门操作的演示，Nature 605, 675 (2022). [31] D. M. Greenberger, M. A. Horne, and A. Zeilinger, 超越贝尔定理 (2007), arXiv:0712.0921. [32] T. Tansuwannont, B. Pato, and K. R. Brown, Shor式纠错的自适应综合征测量，Quantum 7, 1075 (2023). [33] G. Escobar-Arrieta and M. Gutiérrez, 使用Steane综合征提取方法提升Bacon-Shor码的性能，Phys. Rev. A 111, 032427 (2025). [34] A. K. Sharma and S. S. Garani, 容错量子低密度奇偶校验编码器 (2024), arXiv:2405.07242 [量子物理]. [35] A. Kissinger, 无相位ZX图是CSS码（……或者如何以图形方式理解表面码）(2022), arXiv:2204.14038 [量子物理]. [36] B. Coecke and A. Kissinger, 量子过程图示，载于《Diagrammatic Represent. Inference》，由P. Chapman, G. Stapleton, A. Moktefi, S. Perez-Kriz, and F. Bellucci编 (2018). [37] J. van de Wetering, 面向量子计算机科学家的ZX演算 (2020), arXiv:2012.13966. [38] A. Paetznick and B. W. Reichardt, 23量子比特Golay码的容错辅助态制备与噪声阈值下界 (2013), arXiv:1106.2190 [量子物理]. [39] A. Gong and J. M. Renes, 量子Reed-Muller码的计算及其映射到二维原子阵列 (2024), arXiv:2410.23263 [量子物理].

[40] R. Takagi, T. J. Yoder, and I. L. Chuang，编码三量子比特门的错误率与资源开销，Phys. Rev. A **96**, 042302 (2017)。[41] A. Biere, M. Heule, Hans van Maaren, and T. Walsh,《可满足性手册》(IOS Press, 2009)。[42] Leonardo de Moura and N. Bjørner，Z3：一种高效的SMT 求解器，载于工具、算法与系统分析构造(施普林格, 2008) 第337–340页。[43] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith，符号模型检测的反例引导抽象精化，《ACM 杂志》**50**, 752 (2003)。[44] R. A. Moser and G. Tardos，一般洛瓦兹局部引理的构造性证明，《ACM 杂志》**57**, 11:1 (2010)。[45] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schönberger, Y. Stade, and L. Burgholzer，MQT手册：量子计算设计自动化工具与软件概要 (2024), arXiv:2405.17543。[46] C. Gidney，Stim：快速稳定子电路模拟器，Quan- tum **5**, 497 (2021)。

附录 A：完整的电路构造列表

表 III：物理错误率为 $p = 0.1\%$ 时 FT[3] 猫态制备协议的比较。

w	本工作				递归 [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
8	98.44	5	18	14	97.78	7	16	13	0.66	-2	2	1
9			19				18				1	0
10	98.11	6	22	17	97.64	7	20	17	0.47	-1	2	0
11			25				23				2	0
12	97.56	6	28	21	96.90	7	26	21	0.66	-1	2	0
13			29				29				0	-1
14	97.24	6	32	24	95.78	8	32	25	1.46	-2	0	-1
15			35				35				0	-1
16	96.70	6	38	28	94.68	8	38	29	2.02	-2	0	-1
17			39				40				-1	·
18	96.37	7	42	31	94.54	8	42	33	1.83	-1	0	-2
19			45				44				1	-2
20	95.84	7	48	35	94.40	8	46	37	1.43	-1	2	-2
21			49				49				0	-3
22	95.52	7	52	38	93.69	8	52	41	1.83	-1	0	-3
23			55				55				0	-3
24	94.98	7	58	42	92.98	8	58	45	2.01	-1	0	-3
25			59				61				-2	·
26	94.67	7	62	45	91.91	9	64	49	2.76	-2	-2	-4
27			65				67				-2	·
28	94.14	7	68	49	90.85	9	70	53	3.29	-2	-2	-4
29			69				73				-4	·
30	93.83	7	72	52	89.80	9	76	57	4.03	-2	-4	-5
31			75				79				-4	·
32	93.30	7	78	56	88.76	9	82	61	4.54	-2	-4	-5
33			79				84				-5	·
34	92.99	8	82	59	88.63	9	86	65	4.36	-1	-4	-6
35			85				88				-3	·

表 IV：物理错误率为 $p = 0.1\%$ 时的容错[4] 猫态制备方案对比。

w	本工作					递归 [26]					Δ				
	R_A	d	CX	Q		R_A	d	CX		Q	R_A	d	CX	Q	
8	98.44	5	18	14		97.46	8	18		14	0.98	-3	0		0
9			19				8	20		16			-1		-1
10	98.11	6	22	17		97.32	8	22		18	0.79	-2	0		-1
11			25				8	25		20			0		-1
12	97.56	6	28	21		96.58	8	28		22	0.98	-2	0		-1
13			31				8	31		24			0		-1
14	97.24	6	32	24		95.47	8	34		26	1.77	-2	-2		-2
15			35				9	39		29			-4		-3
16	96.70	6	38	28		93.75	9	44		32	2.94	-3	-6		-4
17			39				9	46		34			-7		-5
18	96.37	7	42	31		93.62	9	48		36	2.76	-2	-6		-5
19			45				9	50		38			-5		-5
20	95.84	7	48	35		93.48	9	52		40	2.36	-2	-4		-5
21			51				9	55		42			-4		-5
22	95.52	7	52	38		92.77	9	58		44	2.74	-2	-6		-6
23			55				9	61		46			-6		-6
24	94.99	7	58	42		92.07	9	64		48	2.92	-2	-6		-6
25			61				9	67		50			-6		-6
26	94.45	7	64	46		91.01	9	70		52	3.44	-2	-6		-6
27			67				9	73		54			-6		-6
28	94.14	7	68	49		89.96	9	76		56	4.18	-2	-8		-7
29			71					81		59					-8
30	93.62	7	74	53		88.34	10	86		62	5.27	-3	-12		-9
31			77					91		65					
32	93.30	7	78	56		86.76	10	96		68	6.54	-3	-18	-12	
33			81					98		70					
34	92.78	8	84	60		86.63	10	100		72	6.15	-2	-16	-12	
35			87							74					
36	92.47	8	88	63		86.50	10	104		76	5.97	-2	-16	-13	
37			91							78					
38	91.96	8	94	67		86.38	10	108		80	5.58	-2	-14	-13	
39			97							82					
40	91.65	8	98	70		86.25	10	112		84	5.40	-2	-14	-14	
41										86					
42	91.14	8	104	74		85.60	10	118		88	5.54	-2	-14	-14	
43										90					
44	90.84	8	108	77		84.95	10	124		92	5.89	-2	-16	-15	
45										94					
46	89.51	8	122	85		84.30	10	130		96	5.21	-2		-8	-11
47										98				-8	-
48	89.01	8	128	89		83.67	10	136	100	5.35	-2			-8	-11
49															

表 V：物理错误率为 $p = 0.1\%$ 时 FT[5] 猫态制备协议的比较。

w	本工作				递归 [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
10	98.11	6	22	17	97.00	8	24	19	1.11	-2	-2	-2
11			25			8	27	21			-2	-2
12	97.12	6	32	23	96.27	8	30	23	0.86	-2	2	0
13			33			9	33	25			0	-1
14	96.80	6	36	26	95.16	9	36	27	1.64	-3	0	-1
15			37			9	41	30			-4	-3
16	96.48	6	40	29	93.45	9	46	33	3.03	-3	-6	-4
17			41			9	48	35			-7	-5
18	95.94	7	46	33	93.31	9	50	37	2.63	-2	-4	-4
19			47			9	54	40			-7	-6
20	95.40	7	52	37	92.57	9	58	43	2.84	-2	-6	-6
21			53			9	61	45			-8	-7
22	95.09	7	56	40	91.87	9	64	47	3.22	-2	-8	-7
23			59			9	67	49			-8	-7
24	94.55	7	62	44	91.17	9	70	51	3.38	-2	-8	-7
25			63				73	53				-8
26	94.03	7	68	48	90.12	10	76	55	3.90	-3	-8	-7
27			71				79	57			-8	-7
28	93.50	7	74	52	89.08	10	82	59	4.42	-3	-8	-7
29			75				87	62				-9
30	93.19	7	78	55	87.48	10	92	65	5.71	-3	-14	-10
31			81				97	68				
32	92.67	7	84	59	85.91	10	102	71	6.76	-3	-18	-12
33			85					73				
34	92.36	8	88	62	85.78	10	106	75	6.58	-2	-18	-13
35			91					77				
36	91.64	8	96	67	85.66	10	110	79	5.98	-2	-14	-12
37			97					82				
38	91.13	8	102	71	84.98	10	118	85	6.15	-2	-16	-14
39								88				
40	90.62	8	108	75	84.30	10	126	91	6.32	-2	-18	-16
41								93				
42	90.52	8	110	77	83.66	10	132	95	6.86	-2	-22	-18
43								97				
44	89.61	8	120	83	83.03	10	138	99	6.58	-2	-18	-16
45												
46	89.31	8	124	86	82.40	10	144	103	6.91	-2	-20	-17
47												

表 VI：物理错误率为 $p = 0.1\%$ 时，FT[6] 猫态制备协议的比较。

w	本工作				递归 [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
12	97.12	6	32	23	95.95	8	32	24	1.17	-2	0	-1
13			33			9	35				-2	-2
14	96.58	6	38	27	94.85	9	38	28	1.73	-3	0	-1
15			39			9	43				-4	-3
16	96.26	6	42	30	93.14	9	48	34	3.11	-3	-6	-4
17			43			9	50				-7	-5
18	95.94	7	46	33	93.01	9	52	38	2.93	-2	-6	-5
19			49			9	56				-7	-6
20	95.40	7	52	37	92.27	9	60	44	3.14	-2	-8	-7
21			53			9	63					-8
22	95.09	7	56	40	91.57	9	66	48	3.52	-2	-10	-8
23			61			9	71					-8
24	94.55	7	62	44	90.28	9	76	54	4.27	-2	-14	-10
25			67				79					-9
26	94.03	7	68	48	89.24	10	82	58	4.78	-3	-14	-10
27			73				85					-9
28	93.50	7	74	52	88.21	10	88	62	5.29	-3	-14	-10
29			77				93					
30	93.19	7	78	55	86.63	10	98	68	6.56	-3	-20	-13
31			85									
32	92.46	7	86	60	85.07	10	108	74	7.38	-3	-22	-14
33			87									
34	91.95	8	92	64	84.95	10	112	78	7.00	-2	-20	-14
35			97									
36	91.43	8	98	68	84.82	10	116	82	6.61	-2	-18	-14
37												

表 VII：物理错误率为 $p = 0.1\%$ 时 FT[7] 猫态制备协议的比较。

w	本工作				递归 [26]				Δ		
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX Q
14	96.58	6	38	27	94.54	9	40	29	2.04	-3	-2 -2
15			39			9	45				-6 -4
16	96.04	6	44	31	92.84	9	50	35	3.20	-3	-6 -4
17			45			9	52				-7 -5
18	95.50	7	50	35	92.70	9	54	39	2.80	-2	-4 -4
19			51			9	58				-7 -6
20	94.97	7	56	39	91.97	9	62	45	3.00	-2	-6 -6
21			57			9	65				-8 -7
22	94.44	7	62	43	91.27	9	68	49	3.17	-2	-6 -6
23			63			9	73				-8
24	93.91	7	68	47	89.99	9	78	55	3.93	-2	-10 -8
25			71				81				-8
26	93.39	7	74	51	88.95	10	84	59	4.44	-3	-10 -8
27			75				89				
28	92.87	7	80	55	87.35	10	94	65	5.51	-3	-14 -10
29			83				99				
30	92.35	7	86	59	85.78	10	104	71	6.57	-3	-18 -12
31			89								
32	91.63	7	94	64	84.24	10	114	77	7.38	-3	-20 -13
33			97								

表 VIII：物理错误率为 $p = 0.1\%$ 时 FT[8] 猫态制备方案的比较。

w	本工作				递归 [26]				Δ		
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX Q
16	96.04	6	44	31	92.54	10	52	36	3.50	-4	-8 -5
17			45			9	54				-9 -
18	95.50	7	50	35	92.40	9	56	40	3.10	-2	-6 -5
19			53			9	60				-7 -
20	94.76	7	58	40	91.66	9	64	46	3.09	-2	-6 -6

表 IX：物理错误率为 $p = 0.1\%$ 时的容错[9] 猫态制备协议比较。

w	本工作				递归 [26]				Δ		
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX Q
18	95.50	7	50	35	92.10	10	58	41	3.40	-3	-8 -6
19			53				62				-9 -