

基于码转换的表面码容错通用量子计算实现方法*

权东晓^{1)2)†} 王贞理¹⁾ 程龙祥¹⁾

1) (西安电子科技大学通信工程学院, 西安 710071)

2) (西安电子科技大学, 陕西省量子信息协同创新中心, 西安 710071)

(2025年9月30日收到; 2025年12月9日收到修改稿)

量子计算在处理某类特定问题时展现出巨大潜力,但其易受噪声干扰的特性要求高效的量子纠错方案.表面码是一种重要的量子纠错码,其中二维(2D)表面码易于实现 Clifford 门集,而三维(3D)表面码可横截实现 CCZ 门和 CZ 门,通过两者之间的码转换可高效实现容错通用门集.本文旨在优化 3D 表面码的编码线路设计,并研究其与 2D 表面码之间的码转换机制,以降低容错通用量子计算的资源开销.采用基于稳定子实现的编码线路设计及时隙优化方法,实现了高效的逻辑态编码;进一步提出基于融合与分割操作的码转换方案,并在 Qiskit 平台上对码距为 2 的实例进行仿真验证.结果表明,所设计的编码线路可以在较少时隙内正确编码,码转换方案能够有效实现任意逻辑态在 2D 表面码与 3D 表面码间的相互传输.本方案避免了魔幻态蒸馏过程,为高效实现容错通用量子计算提供了一种新途径.

关键词: 容错通用量子计算, 量子纠错码, 量子表面码, 码转换**DOI:** 10.7498/aps.75.20251344**CSTR:** 32037.14.aps.75.20251344

1 引言

量子计算能够突破传统系统的极限,近年来在理论和实验上取得突飞猛进的发展.中央网络安全和信息化委员会印发《“十四五”国家信息化规划》,其中“量子信息”作为关键前沿领域被提及十余次,充分体现了党和国家对量子科技发展的高度重视.而且近几年随着大数据时代的到来和人工智能领域的高速发展,人们对计算速度和能力的要求越来越高.量子叠加原理使量子信息具有存储量大和可以并行处理等优点,因此量子计算机相比传统计算机拥有指数倍增长的数据处理能力,成为当今最有前景的前沿技术之一^[1,2].

但是量子信息在存储、传输和处理过程中极易出现错误.为发现和纠正量子错误,量子信息学引入了经典信息学中的信道编码体系,即通过信息

状态的重复和纠缠等方法增加冗余,从而通过稳定子测量检测出错误信息并纠正,确保信息准确无误^[3].为了构建这样一个需要经历噪声和错误的量子计算机,需要一种具有容错能力的架构,将量子信息先编码成逻辑量子比特,在逻辑量子比特上进行传输和相应的逻辑操作,在此过程中不断地使用量子纠错来校正计算过程中出现的错误,最终再将逻辑量子比特解码为量子信息比特,这就是量子纠错编解码产生的意义^[4].

2D 表面码 (two-dimensional surface codes) 是一种重要的量子纠错码,其概念在 2003 年首次由 Kitaev^[5] 正式提出.2D 表面码由一种复曲面码 (toric codes) 演变而来^[6,7],其二维近邻耦合量子比特晶格结构、高阈值错误率、低开销以及任意长度逻辑门的特性,使其成为实现大规模容错通用量子计算最有前景的方案之一^[8,9].容错通用量子计算的实现需要结合 Clifford 门以及非 Clifford 门

* 国家自然科学基金 (批准号: 62471350, 62001351) 和陕西省重点研发计划 (批准号: 2019ZDLGY09-02) 资助的课题.

† 通信作者. E-mail: dxquan@xidian.edu.cn

(non-Clifford)^[10], 基于表面码的横截操作、稳定子的测量与校正、拓扑结构的变形、融合和分割操作以及编织操作等可以实现一个完整的 Clifford 门集^[11]. 2025 年 6 月, Tasler 等^[12]提出了一种在 2D 表面码中使用 CCZ 门可以进一步提高表面码错误阈值和逻辑性能的方法, 将物理错误阈值从约 0.66% 提升至约 1.2%. 在 2D 表面码上实现容错通用量子计算的一种途径是借助魔幻态蒸馏技术 (magic state distillation)^[13,14], 虽然目前提出的方法大大缩小了魔幻态蒸馏需要的资源^[15], 但是对于每一个 CCZ 门都需要相同的资源开销.

2019 年, Vasmer 和 Browne^[16]在三维拓扑结构^[17,18]中研究 3D 表面码 (three-dimensional surface codes) 并提出了一种可视化的架构, 该架构在实现非 Clifford 门集中的 CCZ 门时不需要借助魔幻态蒸馏. 2021 年 3 月, Scruby 等^[19]通过 3D 表面码提出了一组完全的有界高度晶格切片, 这些切片保留了完整 3D 表面码的码距和测量误差检测属性. 2023 年, Huang 等^[20]提出了 3D 表面码的 Clifford 变形, 证明了其在无限偏置的泡利噪声下具有 50% 的错误率阈值, 并提升了其在有限偏置下的存储性能. 2025 年 4 月, Lee 等^[21]进一步将 3D 环面码推广到模块化量子位, 提供了更丰富的相结构和激发行为.

综上所述, 2D 表面码易于实现 Clifford 门集, 而 3D 表面码可横截实现 CCZ 门, 如果能够在量子计算过程中适时实现 2D 表面码和 3D 表面码之间的转换, 则可以以 2D 表面码为主, 遇到非 Clifford 门时转换到 3D 表面码实现 CCZ 门, 从而能够不借助魔幻态蒸馏实现容错通用量子计算. 因此本文旨在设计优化 3D 表面码的编码线路, 并研究其与 2D 表面码之间的码转换机制, 以获得更优的量子纠错性能, 为高效实现容错通用量子计算提供可行的途径.

2 基于稳定子实现的 3D 表面码编码线路设计优化

2.1 表面码基础结构

2.1.1 2D 表面码结构

量子 2D 表面码属于稳定子码^[22], 它将量子比特排列在可扩展的二维平面结构上, 包括最早提出

的表面码^[23]以及旋转 45°后形成的更节省物理比特的表面码^[24]. 图 1(a) 是码距为 3 的 2D 表面码, 图 1(b) 是旋转后的 2D 表面码.

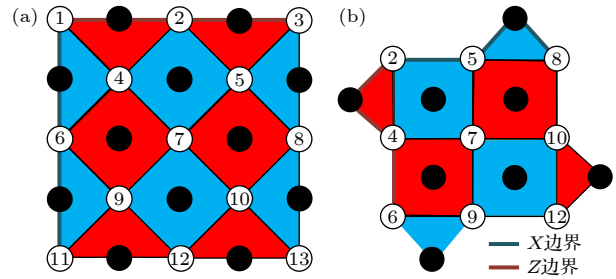


图 1 2D 表面码结构图 (a) 旋转前; (b) 旋转后
Fig. 1. Structure diagram of 2D surface code: (a) Before rotation; (b) after rotation.

量子 2D 表面码将量子比特置于顶点, 晶格面红和蓝二着色, X 稳定子和 Z 稳定子关联于不同颜色的面. 在图 1 中, 白色比特为数据比特, 黑色比特为测量比特, 蓝色区域为 X 型稳定子, 红色区域为 Z 型稳定子. 从图中可以看出不同类型的稳定子间都有 0 个或 2 个, 即偶数个共同的数据比特, 因此它们互相对易, 又因为同类型的稳定子之间也互相对易, 所以所有的稳定子之间均满足对易性; 所有稳定子也满足独立性, 即任何一个稳定子都不能由其他稳定子直积得到. 如果定义红色区域为 X 型稳定子, 蓝色区域为 Z 型稳定子, 则可以形成另外一种表面码.

表面码有两种边界: X 型稳定子所形成的 X 边界, 又称光滑边界; Z 型稳定子所形成的 Z 边界, 又称粗糙边界. 连接两个 X 边界且与所经过的每个 Z 型稳定子有偶数个共同数据比特的连线上的所有数据比特 X 操作的直积就是 X 逻辑操作 \bar{X}_L ; 类似地, 连接两个 Z 边界且与所经过的每个 X 型稳定子有偶数个共同数据比特的连线上的所有数据比特 Z 操作的直积就是 Z 逻辑操作 \bar{Z}_L . 信息比特要选择为 \bar{X}_L 和 \bar{Z}_L 所涉及共同量子比特, 可以看出 \bar{X}_L 和 \bar{Z}_L 与所有的稳定子对易, 两者反对易.

2.1.2 3D 表面码结构

量子 3D 表面码的结构如图 2 所示, 图 2(a) 是可扩展的码距为 d 的 3D 表面码, 图 2(b) 是一个具体的码距为 3 的 3D 表面码. 量子 3D 表面码是 2D 表面码在三维的类比扩展, 其数据量子比特也被放置在 3D 晶格的各个顶点上, 由图中绿色球表示. 量子 3D 表面码的 X 稳定子与晶格的体积相关

联, Z 稳定子与晶格的面关联, 晶格体红、绿、蓝三着色 [16]. 基于该结构可以定义三种量子纠错码, 记为 $SC_{c(c=r,g,b)}$, X 型稳定子是颜色 c 所对应的体结构, 由该体结构所涉及数据量子比特的 X 操作构成; Z 型稳定子由另外两种颜色的共同面标识, 由该平面所涉及数据量子比特的 Z 操作构成. 不同颜色 3D 表面码的 X 和 Z 型稳定子如图 3 所示, 图中红码和蓝码的 X 型稳定子的体结构是相同的, 由相应颜色的立方八面体构成, Z 型稳定子分别由蓝绿面和红绿面构成; 绿码的 X 型稳定子由绿色八面体构成, Z 型稳定子由红蓝面构成. 与 2D 旋

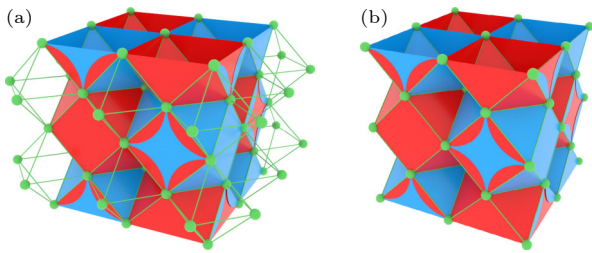


图 2 3D 表面码结构图 (a) 码距为 d 的可扩展 3D 表面码; (b) 码距为 3 的 3D 表面码

Fig. 2. Structure diagram of 3D surface code: (a) Scalable with code distance d ; (b) specific case with code distance 3.

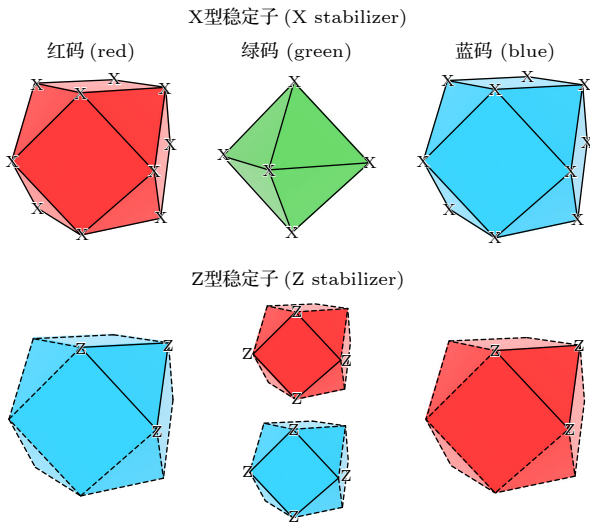


图 3 三种 3D 表面码的 X 和 Z 型稳定子示意图

Fig. 3. Diagram of X and Z stabilizers for three types of 3D surface codes.

转表面码边界处的稳定子类似, 图 2 中边界处增加的弓形稳定子保证所有量子比特均涉及 X 型和 Z 型两种稳定子, 形成连续边界.

为方便判断量子比特数和三种不同码稳定子数量, 按照水平方向将 3D 表面码划分为“棋盘层”和“菱形层”, 分别如图 4(a) 和图 4(b) 所示, 每个顶点处放置数据量子比特. 每个棋盘层都有 d^2 个量子比特, 菱形层实际上是由棋盘层每边中点处的量子比特构成的, 每个菱形层有 $2d(d-1)$ 个量子比特. 码距为 d 的 3D 表面码含有 d 个棋盘层和 $d-1$ 个菱形层, 因此物理量子比特数量为

$$n = d^3 + 2d(d-1)^2 = 3d^3 - 4d^2 + 2d, \quad (1)$$

结合分析图 3 和图 4 可以得到 3D 表面码三种不同颜色码的稳定子数目, 如表 1 所列, 详细分析过程见文献 [16].

与 2D 表面码类似, 3D 表面码也有粗糙边界和光滑边界, 在图 2 中, 前后两个面对应于红码和绿码的 X 型稳定子, 蓝码的 Z 型稳定子, 因此称为红码和绿码的光滑边界, 蓝码的粗糙边界; 类似地, 左右两个面为蓝码和绿码的光滑边界、红码的粗糙边界; 上下两个面为红码和蓝码的光滑边界、绿码的粗糙边界. 这里额外定义立方晶格的颜色边界; c 边界对应于 $SC_{(c)}$ 的粗糙边界 (Z 边界), 对应于 $SC_{(c')}$ 和 $SC_{(c'')}$ 的光滑边界 (X 边界). 根据此定义可以看到, 前后两个面对应于蓝色边界, 左右两个面对应于红色边界, 上下两个面对应于绿色边界. 逻辑 X 操作为连接两对光滑边界的平面上所有量子比特的 X 操作的直积, 逻辑 Z 操作为连接两个

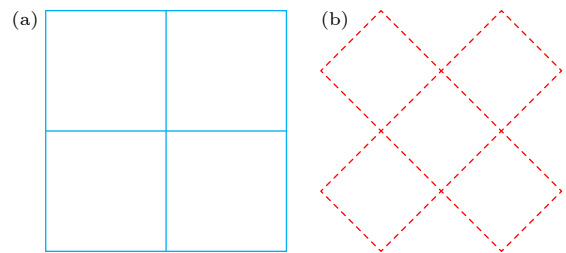


图 4 (a) 棋盘层; (b) 菱形层

Fig. 4. (a) Checkerboard layers; (b) diamond layers.

表 1 3D 表面码中红绿蓝码的稳定子与逻辑量子比特数目

Table 1. Number of stabilizers and logical qubits of rgb codes in the 3D surface code.

编码	X 稳定子数目	Z 稳定子数目	稳定子总数目	逻辑量子比特数 k
SC_r 和 SC_b	$(d-1)(d^2+d)/2$	$(d-1)(5d^2-3d+2)/2$	$3d^3-4d^2+2d-1$	1
SC_g	$d^2(d-1)$	$(d-1)(2d^2-d+1)$	$3d^3-4d^2+2d-1$	1

粗糙面的棱上所有量子比特的 Z 操作直积. 对于红码, 光滑边界为前后面和上下面, 粗糙边界为左右面, 因此逻辑 X 操作可以选择左侧面或者右侧面, 逻辑 Z 操作为连接左右面的棱; 对于绿码, 光滑边界为前后面和左右面, 逻辑 X 操作选择为下底面或者上底面, 逻辑 Z 操作为连接上下面的棱; 对于蓝码, 光滑边界为左右面和上下面, 逻辑 X 操作选择为前面或者后面, 逻辑 Z 操作为连接前后面的棱. 单个堆栈即图 2 中三种 3D 表面码的逻辑操作 \bar{Z}_c 和 \bar{X}_c , 如图 5 所示, 图中红码的逻辑 X 操作选取为右侧面, 蓝码的逻辑 X 操作选取为前面, 绿码的逻辑 X 操作选取为上底面.

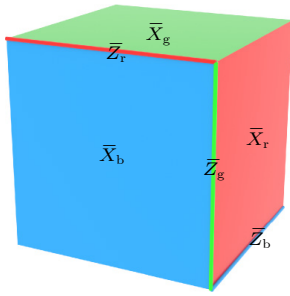


图 5 3D 表面码的逻辑操作示意图

Fig. 5. Diagram of logical operations for the 3D surface code.

2.2 3D 表面码编码线路设计与仿真

2.2.1 编码线路的设计与优化原理

编码线路的实现方法主要有两种: 一种是基于稳定子测量和纠正的编码线路^[25]; 第二种是基于稳定子实现的编码线路^[26]. 由于基于稳定子测量的编码线路在编码过程中会发生随机塌缩, 需要根据测量得到的结果添加必要的纠正操作才能正确编码. 基于稳定子测量的编码线路方法所消耗的资源较多, 所以本文基于稳定子的实现设计编码线路. 编码线路的第一部分是逻辑操作的实现, 首先需要选取 X 逻辑操作和 Z 逻辑操作共同涉及的量子比特作为信息比特, 将辅助物理量子比特初始化为 $|0\rangle$, 然后利用信息比特控制逻辑操作的其他比特. 在此过程中, 若信息比特为 $|0\rangle$, 则此部分量子线路不起作用, 后面将实现 $|0\rangle_L$ 的编码, 若信息比特为 $|1\rangle$, 则相当于对初态进行了 \bar{X}_L 操作, 最终将实现 $|1\rangle_L$ 的编码. 第二部分是利用 H 门和 CNOT 门逐个实现所有 X 型稳定子. 单个稳定子的实现可以首先对稳定子涉及的初态为 $|0\rangle$ 的量子比特进

行 H 操作, 然后利用该比特控制稳定子涉及的其他量子比特以实现稳定子. 此过程可以描述为

$$\begin{aligned} & CN_{ck}CN_{cj}CN_{ci}H_c|0\rangle_c|A\rangle_i|B\rangle_j|C\rangle_k \\ &= \frac{1}{\sqrt{2}}CN_{ck}CN_{cj}CN_{ci}(|0\rangle_c + |1\rangle_c)|A\rangle_i|B\rangle_j|C\rangle_k \\ &= \frac{1}{\sqrt{2}}(|0\rangle_c|A\rangle_i|B\rangle_j|C\rangle_k + |1\rangle_c|\bar{A}\rangle_i|\bar{B}\rangle_j|\bar{C}\rangle_k) \\ &= \frac{1}{\sqrt{2}}(I + X_cX_iX_jX_k)|0\rangle_c|A\rangle_i|B\rangle_j|C\rangle_k, \quad (2) \end{aligned}$$

其中, $|\psi\rangle_i$ 表示第 i 个量子比特, CN_{ci} 表示以第 c 个量子比特为控制比特, 第 i 个量子比特为目标比特的 CNOT 门, H_i 表示对第 i 个量子比特作用 H 门. 依次实现所有的稳定子就可以实现信息比特的编码.

编码线路的优化主要是根据量子门之间的对易性进行量子门顺序的调整, 对整个编码线路进行时隙的划分, 在同一个时隙里放置尽可能多的量子门, 从而减小量子比特的存储错误, 提高整个线路的正确率. 在优化过程中用到的对易性主要是 CNOT 门的对易性, 具体包括: 拥有共同控制比特的两个 CNOT 门是对易的; 拥有共同目标比特的两个 CNOT 门是对易的; 若同一个量子比特在两个 CNOT 门中分别用作控制比特和目标比特, 则这两个门是非对易的.

基于稳定子实现的编码线路会因为 CNOT 门的存在而产生错误的传播, 带来多比特错误, 针对此问题可以采用的解决方案包括: 1) 添加 flag 比特, 通过对 flag 比特的测量判断编码线路在指定位置是否出错^[27,28], 再结合稳定子测量的结果进行纠错从而得到正确的编码^[26]; 2) 由于基于稳定子实现的编码过程消耗的资源较少, 且量子门的精度很高, 编码过程中出错的概率很低, 而且整个编码线路处于量子计算前端, 如果编码过程中允许一定的失败概率, 则可以仅在编码线路之后添加一轮症状测量, 发现有误码时直接舍弃此次编码的逻辑比特, 重置辅助量子比特, 重新编码即可. 本文将采用第 2 种策略设计编码线路.

2.2.2 $[[12, 1, 2]]$ 3D 表面码编码线路的设计与仿真

随着量子计算技术的快速发展, 量子仿真平台成为研究量子算法、验证量子计算理论的重要工具. 目前主流的量子仿真平台有 IBM 公司推出的

Qiskit 仿真平台、Google 公司推出的 Google Cirq 仿真平台、本源量子推出的 OriginQ 仿真平台等。本文选用 Qiskit 平台进行仿真，Qiskit 支持开源 Python 库，支持量子线路的在线构建和仿真，并提供了高性能的量子仿真器 Qiskit Aer。由于 Qiskit 对量子比特数量的限制，下面以码距为 2 的 $[[12, 1, 2]]$ 3D 表面码为例对所设计的编码线路进行仿真。

$[[12, 1, 2]]$ 3D 表面码结构如图 6 所示，包括两个棋盘层和一个菱形层，其中绿色球体表示数据量子比特，编号如图 6 右图所示，稳定子的标识与图 2 一致。根据图 3 中的定义，可以得到 $[[12, 1, 2]]$ 3D 表面码红码的 X 型稳定子为： $X_5 X_6 X_7 X_8 X_9 X_{10} X_{11} X_{12}$ ， $X_1 X_3 X_5$ ， $X_2 X_4 X_7$ ；Z 型稳定子为： $Z_6 Z_9$ ， $Z_6 Z_{10}$ ， $Z_8 Z_{11}$ ， $Z_8 Z_{12}$ ， $Z_2 Z_6 Z_7$ ， $Z_1 Z_5 Z_6$ ， $Z_4 Z_7 Z_8$ ， $Z_3 Z_5 Z_8$ ；X 逻辑操作为 $\bar{X}_r = X_3 X_4 X_8 X_{11} X_{12}$ ；Z 逻辑操作为 $\bar{Z}_r = Z_1 Z_3$ 。基于稳定子实现得到的编码线路如图 7 所示。

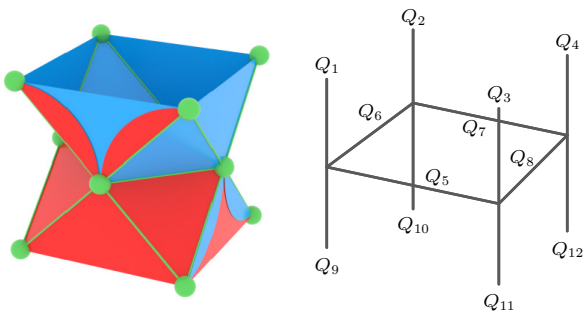


图 6 $[[12, 1, 2]]$ 3D 表面码示意图
Fig. 6. Diagram of the $[[12, 1, 2]]$ 3D surface code.

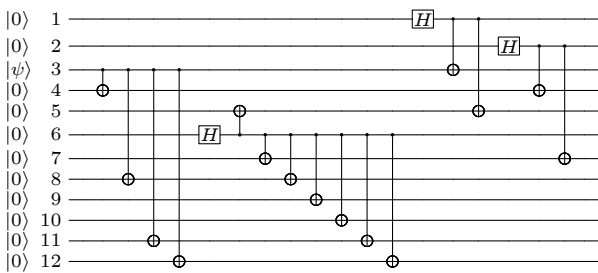


图 7 $[[12, 1, 2]]$ 3D 表面码红码编码线路图
Fig. 7. Quantum encoding circuit for the $[[12, 1, 2]]$ 3D surface code (Red code).

分别设置信息比特的值为 $|0\rangle$ ， $|1\rangle$ ， $|+\rangle$ 和 $|-\rangle$ ，得到 $[[12, 1, 2]]$ 3D 表面码红码的码字结果如附录 A 中表 A1 所列。对整个编码线路进行时隙优化后的线路如图 8 所示，整个线路需要的时隙数由 18 个减小为 8 个，而稳定子的最大权重为 8，即

施加 H 门的量子比特需要与该稳定子涉及的其余 7 个量子比特进行 CNOT 操作，因此这已经是最优的结果。

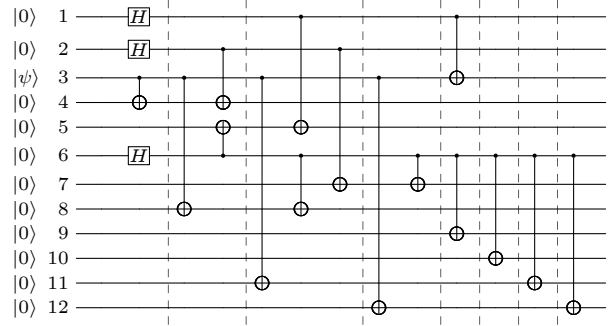


图 8 时隙优化后的 $[[12, 1, 2]]$ 3D 表面码红码编码线路图
Fig. 8. Time-slot optimized $[[12, 1, 2]]$ 3D surface code encoding circuit diagram (Red code).

根据图 3 中的定义，可以得到 $[[12, 1, 2]]$ 3D 表面码绿码的 X 型稳定子为： $X_1 X_5 X_6 X_9$ ， $X_2 X_6 X_7 X_{10}$ ， $X_3 X_5 X_8 X_{11}$ ， $X_4 X_7 X_8 X_{12}$ ；Z 型稳定子为： $Z_1 Z_3 Z_5$ ， $Z_1 Z_2 Z_6$ ， $Z_2 Z_4 Z_7$ ， $Z_3 Z_4 Z_8$ ， $Z_5 Z_9 Z_{11}$ ， $Z_6 Z_9 Z_{10}$ ， $Z_7 Z_{10} Z_{12}$ ；X 逻辑操作为： $\bar{X}_g = X_1 X_2 X_3 X_4$ ；Z 逻辑操作为： $\bar{Z}_g = Z_3 Z_{11}$ 。基于稳定子实现得到的编码线路如图 9 所示，分别设置信息比特的值为 $|0\rangle$ ， $|1\rangle$ ， $|+\rangle$ 和 $|-\rangle$ ，得到 $[[12, 1, 2]]$ 3D 表面码绿码的码字如附录 A 中表 A2 所列。对整个编码线路进行时隙优化后的线路如图 10 所示，需要的时隙数目由 19 个降低为 4 个。同理，由于其稳定子的最大权重为 4，因此这也是最优的结果。此线路时隙优化效果较好的原因是其 X 型稳定子的长度比较均衡。

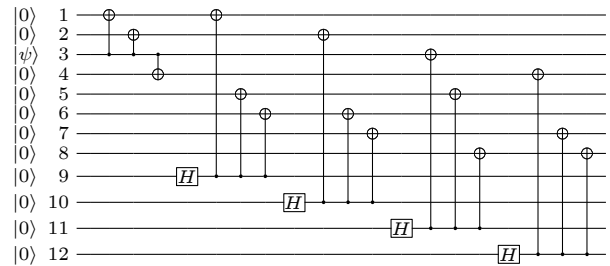


图 9 $[[12, 1, 2]]$ 3D 表面码绿码编码线路图
Fig. 9. Quantum encoding circuit for the $[[12, 1, 2]]$ 3D surface code (Green code).

通过红码和蓝码的优化线路可以看出，优化后的线路在最大稳定子权重个时隙内实现了编码，达到了最优设计。但是当扩展到更大码距时，由于 X 型逻辑操作涉及的量子比特数目为 $O(L^2)$ ，

在 Z 基下编码需要实现 X 型稳定子, 因此需要的时隙数目将受到 X 型逻辑操作长度的限制, 不能达到 $O(L)$ [29,30], 此时可以在 X 基下进行编码, 实现线型的 Z 逻辑操作, 从而逼近理论值.

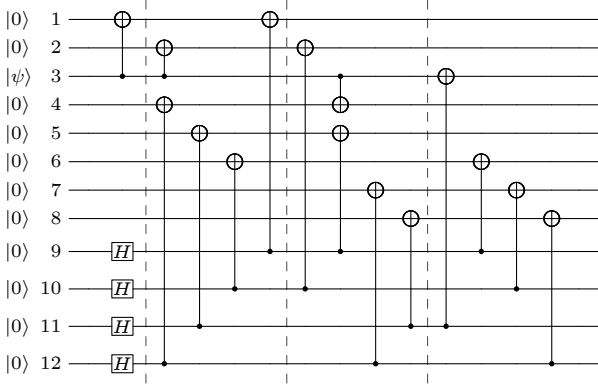


图 10 时隙优化后的 $[[12, 1, 2]]$ 3D 表面码绿码编码线路图
Fig. 10. Time-slot optimized $[[12, 1, 2]]$ 3D surface code encoding circuit diagram (Green code).

由于蓝码和红码的相似性, 可以仅仅改变量子比特的编号得到, 红码和蓝码的量子比特之间的对应关系可以看作: $1 \leftrightarrow 12, 2 \leftrightarrow 10, 3 \leftrightarrow 11, 4 \leftrightarrow 9, 5 \leftrightarrow 8, 6 \leftrightarrow 7$, 这里不再给出编码线路, 分别设置信息比特的值为 $|0\rangle, |1\rangle, |+\rangle$ 和 $|-\rangle$, 得到 $[[12, 1, 2]]$ 3D 表面码蓝码的码字如附录 A 中表 A3 所列.

3 2D 表面码与 3D 表面码之间的转换

3.1 2D 表面码与 3D 表面码逻辑门集性质

2D 表面码的结构与当下主流的量子计算芯片一致, 且相较于 3D 表面码更容易实现, 因此目前实现容错通用量子计算主要基于 2D 表面码. 容错通用门集由 Clifford 门集 $X, Z, S, H, \text{CNOT}, \text{CZ}$ 和非 Clifford 门集 $T, \text{CCNOT}, \text{CCZ}$ 中的至少一个构成 [10], 而基于表面码的横截操作, 稳定子的测量与校正、拓扑结构的变形、融合和分割操作、编织操作等可以实现一个完整的 Clifford 门集 [11]. 目前通常选择实现非 Clifford 门集中的 T 门以在 2D 表面码上实现容错通用量子计算, 而 T 门需要借助魔幻态蒸馏来实现 [31], 对每一个 T 门都需要消耗较多的量子比特和量子门. 3D 表面码对于非 Clifford 门之一的 CCZ 门是横截的 [16], 能够不借助魔幻态蒸馏实现非 Clifford 门, 从而节省资源.

在 CCZ 门中如果令一个控制比特为 1, 则可以在另外一个控制比特和目标比特之间实现 CZ

门. 由此可知控制比特 1 的 X 逻辑操作涉及的量子比特之间的横截 CCZ 门对于 CZ 门的实现起到了实质性作用, 其余的横截 CCZ 门没有起到作用, 因此 CZ 门也能够 3D 表面码中进行简化的横截实现. 设这三种码分别为 $\text{SC}_{(c)}, \text{SC}_{(c')}$ 和 $\text{SC}_{(c'')}$, $c, c', c'' \in \{r, g, b\}$, $c \neq c' \neq c''$, 在 $\text{SC}_{(c)}$ 和 $\text{SC}_{(c')}$ 之间实现横截 CZ 逻辑门的方案是, 对于 $\text{SC}_{(c')}$ 的逻辑 X 操作涉及的比特, 在 $\text{SC}_{(c)}$ 和 $\text{SC}_{(c')}$ 的对应比特之间进行 CZ 操作, 即对图 5 中 $\text{SC}_{(c)}$ 和 $\text{SC}_{(c')}$ 在 c'' 面上的对应比特之间进行横截 CZ 操作即可.

证明 CZ 门的横截性需要用到下面两个引理, 引理的证明过程详见文献 [16].

引理 1 在 3D 表面码结构中, 任意三个来自不同码的 X 稳定子重叠的量子比特数目是偶数, 任意一个码的 X 逻辑操作与分别来自其他两个码的另外两个 X 稳定子有偶数个共同比特.

引理 2 给定有限个相同长度的二进制向量 $\{a_j\}$, 它们和的奇偶性等于各自奇偶性的和. 即假设有 n 个相同长度的二进制向量, 将它们对应位置相加得到新向量的奇偶性和把每个向量的奇偶性相加的结果相同.

CZ 门的横截性证明过程如下:

$$\overline{\text{CZ}}|\alpha\beta\rangle_{cc'} = \sum_{t \in G_c^\alpha; u \in G_{c'}^\beta} (-1)^{|t \odot u|} |t\rangle_c |u\rangle_{c'}, \quad (3)$$

其中 $t \odot u$ 表示对 c'' 面内的比特按位进行二进制乘积, $|t|$ 表示 t 的汉明重量.

下面分析 $(-1)^{|t \odot u|}$ 的值. 将上述 $t \odot u$ 展开:

$$\begin{aligned} t \odot u &= (\alpha X_{(c)} + t') \odot (\beta X_{(c')} + u') \\ &= \alpha\beta (X_{(c)} \odot X_{(c')}) + \alpha (X_{(c)} \odot u') \\ &\quad + \beta (t' \odot X_{(c')}) + (t' \odot u'), \end{aligned} \quad (4)$$

其中 $t' \in G_{(c)}^0, u' \in G_{(c')}^0$.

对于 $|00\rangle$ 态, 对应 $(t' \odot u')$, 其汉明重量为 c'' 面上 $\text{SC}_{(c)}$ 和 $\text{SC}_{(c')}$ 的 X 稳定子重叠的量子比特数目, 由引理 1 可知, 重叠的数目总是偶数, 因此 $(-1)^{|t \odot u|} = 1$. 对于 $|01\rangle$ 态, 其指数为 $(t' \odot X_{(c')}) + (t' \odot u')$, 对于 $t' \odot X_{(c')}$, 其含义为 c'' 面和 c' 面的共同棱, 即 Z_c 与 $\text{SC}_{(c)}$ 的 X 稳定子重叠的数目, 为偶数. 由引理 2 可知 $(-1)^{|t \odot u|} = 1$, 对于 $|10\rangle$ 态也类似. 对于 $|11\rangle$ 态, 其指数取决于 $(X_{(c)} \odot X_{(c')})$ 的奇偶性, 由图 5 可以看出三种码的 X 逻辑操作在立方体的一个顶点重合, 因此 $|(X_{(c)} \odot X_{(c')})| = 1$, 由引理 2 得: $(-1)^{|t \odot u|} = -1$.

综上, 通过部分横截操作可以实现逻辑 CZ 门. 且 3D 表面码属于 CSS 码, 因此同色 3D 表面码间的 CNOT 门是横截的^[32-34].

3.2 基于码转换的表面码容错通用量子计算实现方案

在 3D 表面码中能够横截实现 X, Z 门, 在同色 3D 表面码之间能够横截实现 CNOT 门, 在异色 3D 表面码之间能够横截实现 CCZ, CZ 门. 利用 2D 表面码能够较容易地实现 X, Z, H, CNOT 门, 且 2D 表面码的量子比特数目比 3D 表面码少, 若能实现 2D 表面码和 3D 表面码间的转换, 则可以以 2D 表面码为主, 遇到 CCZ 门时转到 3D 表面码实现, 从而能够不借助魔幻态蒸馏实现容错通用量子计算.

基于码转换的表面码容错通用量子计算流程如图 11 所示. 第一步是逻辑层面量子线路的优化, 基于 X, Z, H, CNOT, CZ, CCZ 来实现量子线路, 一个额外的优化目标是 CCZ 门和 H 门、异色 3D 表面码间的 CNOT 门不要频繁地交替出现, 这样可以

避免在两种码之间频繁地转换. 如果量子线路要执行的是 CCZ 门, 则将该量子线路涉及三个逻辑量子态分别转换为 3D 表面码的红码、蓝码和绿码, 继而可以横截实现 CCZ 门; 整个计算实现的流程以 2D 表面码为主, 但是转换到 3D 表面码后, 如果后续量子门是在 3D 表面码中横截实现的量子门, 比如逻辑 X, Z, CZ 门和同色 3D 表面码间的 CNOT 门时则继续在 3D 表面码中进行, 从而避免频繁地在两种码字之间转换. 整个计算过程完成后, 按照量子比特所处的编码空间进行纠错和解码恢复出量子信息.

4 基于融合与分割的 2D 表面码和 3D 表面码间码转换方案

在基于边界的表面码中进行融合与分割的方法由 Horsman 等^[24]在 2012 年提出, 本节首先简单回顾粗糙融合、光滑融合和粗糙分割、光滑分割的原理, 然后基于融合与分割方法提出 2D 表面码与 3D 表面码间码转换的方案并进行仿真验证.

4.1 融合与分割操作

融合操作的作用是将相对独立的两个表面码进行合并, 融合成为一个表面码. 融合操作的实现, 是通过测量两个独立表面码结合后边界上新产生的稳定子来完成的. 根据融合边界的不同, 可以分为粗糙融合和光滑融合, 如图 12 所示. 粗糙融合指的是将两个独立表面码的粗糙边界 (Z 边界) 进行融合操作, 如图 12(a) 所示. 将辅助量子比特 Q_1, Q_2, Q_3 设置为 $|0\rangle$ 态, 测量新增加的 4 个 X 型稳定子, 得到一个新的表面码. 根据四个 X 型稳定子测量结果的乘积, 可以得到新的表面码的逻辑 $|0\rangle$ 态和 $|1\rangle$ 态, 融合后的逻辑 X 操作与融合前表面码的逻辑 X 操作一致, 即 $\bar{X}_{\text{new}} = \bar{X}_{L1} = \bar{X}_{L2}$; 逻辑 Z 操作是融合前表面码的逻辑 Z 操作的直积, 即 $\bar{Z}_{\text{new}} = \bar{Z}_{L1} \otimes \bar{Z}_{L2}$.

光滑融合指的是将两个独立表面码的光滑边界 (X 边界) 进行融合操作, 如图 12(b) 所示. 将辅助量子比特 Q_1, Q_2, Q_3 设置为 $|+\rangle$ 态, 测量新增加的 4 个 Z 型稳定子, 得到一个新的表面码. 根据四个 Z 型稳定子测量结果的乘积, 可以得到新的表面码的逻辑 $|+\rangle$ 态和 $|-\rangle$ 态, 融合后的逻辑 Z 操作与融合前表面码的逻辑 Z 操作一致, 即 $\bar{Z}_{\text{new}} =$

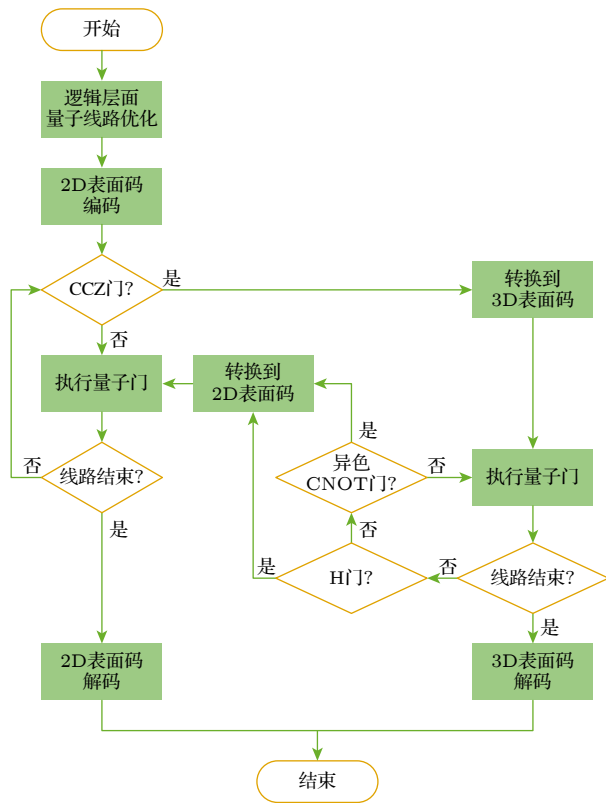


图 11 基于码转换的表面码容错通用量子计算流程图
Fig. 11. Flowchart of code-conversion-based surface code fault-tolerant universal quantum computation.

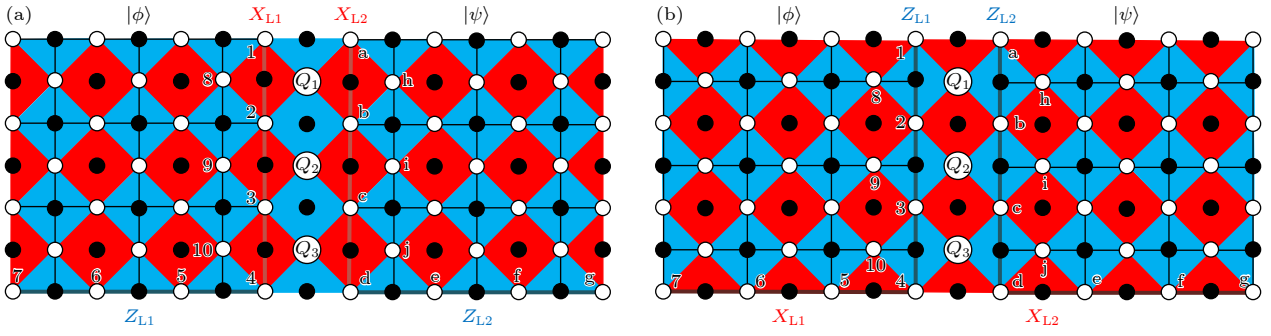


图 12 表面码的融合操作 (a) 粗糙融合; (b) 光滑融合

Fig. 12. Merging operations in surface codes: (a) Rough merging; (b) smooth merging.

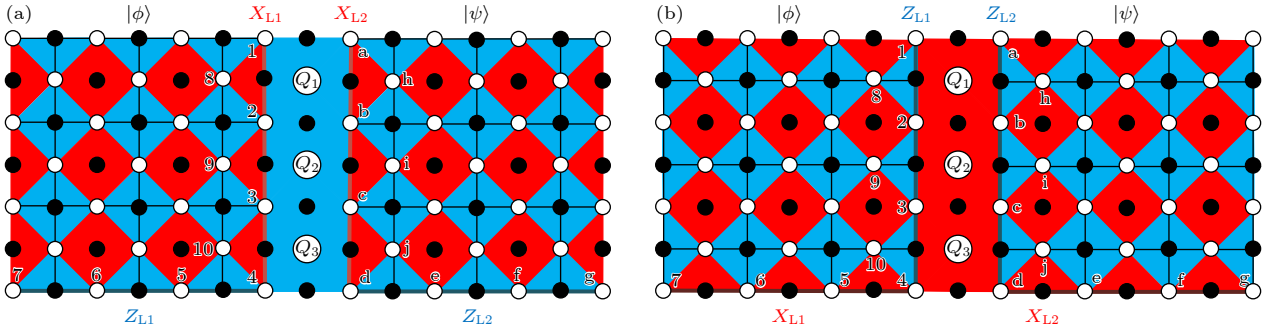


图 13 表面码的分割操作 (a) 粗糙分割; (b) 光滑分割

Fig. 13. Splitting operations in surface codes: (a) Rough splitting; (b) smooth splitting.

$\bar{Z}_{L1} = \bar{Z}_{L2}$; 逻辑 X 操作是融合前表面码的逻辑 X 操作的直积, 即 $\bar{X}_{\text{new}} = \bar{X}_{L1} \otimes \bar{X}_{L2}$. 不论哪种融合方式, 新的表面码中都包含原有表面码幅度的信息和稳定子测量结果乘积的信息.

分割操作相当于融合操作的逆过程, 通过在某一测量基下对单个量子比特进行测量从而消除正交基下的稳定子, 最终正交基下的联合逻辑操作可以视为分割形成的两个表面码的稳定子. 与融合操作类似, 分割操作根据分割后新生成的边界类型分为粗糙分割和光滑分割. 粗糙分割如图 13(a) 所示, 通过对 Q_1 , Q_2 和 Q_3 在 Z 基下进行测量和纠正, 每次测量过程中会使得与测量比特相连的 2 个 Z 型稳定子由四端变为三端, 一个 X 型稳定子消失, 最终形成了图 13(a) 中蓝色大长方形所示的 X 型稳定子, 实际上是左右两个表面码 X 逻辑操作的直积态, 即 $\bar{X}_{L1} \otimes \bar{X}_{L2}$ 是分割操作后的表面码的稳定子, 最终得到的量子状态为

$$\alpha|+\rangle_L + \beta|-\rangle_L = \alpha|++\rangle_{L1L2} + \beta|--\rangle_{L1L2}. \quad (5)$$

可以看到, 粗糙分割操作会形成两个表面码在 X 基下的纠缠态. 如果对其中的一个量子比特在 Z 基下进行逻辑测量, 必要的时候添加 X 逻辑操作, 则得到了携带相同信息的小表面码, 可以理

解为实现了大表面码向小表面码的转换. 进行分割操作后, X 逻辑操作长度不变, Z 逻辑操作被分割成两部分.

同理, 光滑分割如图 13(b) 所示, 通过对 Q_1 , Q_2 和 Q_3 在 X 基下进行测量和纠正, 最终形成了图 13(b) 中红色大长方形所示的 Z 型稳定子, 即 $\bar{Z}_{L1} \otimes \bar{Z}_{L2}$ 是分割操作后的表面码的稳定子, 最终得到的量子状态为

$$\gamma|0\rangle_L + \delta|1\rangle_L = \gamma|00\rangle_{L1L2} + \delta|11\rangle_{L1L2}. \quad (6)$$

同样, 光滑分割操作会形成两个表面码在 Z 基下的纠缠态. 从图形和表达式都可以看出, 在光滑分割过程中, Z 逻辑操作不变, X 逻辑操作被分割成两部分.

4.2 基于融合与分割的码转换实现方法

2D 表面码的融合与分割操作可以扩展到 2D 表面码和 3D 表面码之间. 融合与分割操作实际上是对两个表面码进行逻辑层面的联合测量, 在 3D 表面码中 X 逻辑操作是面结构, 进行联合测量涉及的比特数目较多, 而 Z 逻辑操作是线结构, 因此可以基于光滑融合与分割实现 2D 表面码和 3D 表面码之间的转换.

利用光滑融合与光滑分割的方法实现 2D 表面码和 3D 表面码间转换的码间拼接如图 14 所示, 图中绿色的球表示数据量子比特, 黑色的球表示辅助量子比特, 红色网格表示新增加的 Z 型稳定子, 蓝色网格表示 2D 表面码和 3D 表面码原 X 型稳定子的扩展部分, 通过辅助比特的初始化, 扩展后的 X 型稳定子会自动满足. 2D 表面码与 3D 表面码蓝码间码转换的码间拼接如图 14(a) 所示, 2D 表面码与 3D 表面码绿码间码转换的码间拼接如图 14(b) 所示. 由于红码和蓝码的相似性, 2D 表面码与红码间码转换的码间拼接与图 14(a) 类似, 只需更改 2D 表面码的位置, 将其与 3D 表面码红码的逻辑 Z 操作置于同一平面即可. 下面以 2D 表面码转换到 3D 表面码为例来描述主要步骤.

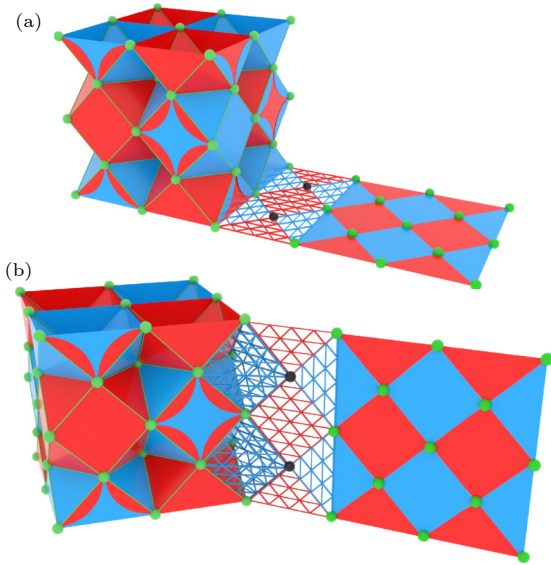


图 14 2D 表面码与 3D 表面码间码转换的拼接方式 (a) 3D 表面码蓝码与 2D 表面码的拼接; (b) 3D 表面码绿码与 2D 表面码的拼接

Fig. 14. Splicing scheme for inter-code conversion between 2D and 3D surface codes: (a) Splicing between 3D surface blue code and 2D surface code; (b) splicing between 3D surface green code and 2D surface code.

第一步, 设 2D 表面码的逻辑量子态为 $|\bar{\psi}\rangle$, 将 3D 表面码的信息比特设为 $|+\rangle$, 并将其编码为逻辑量子比特 $|\bar{\Gamma}\rangle$, 其中

$$|\bar{\psi}\rangle = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle, \quad |\bar{\Gamma}\rangle = \frac{1}{\sqrt{2}} (|\bar{0}\rangle + |\bar{1}\rangle). \quad (7)$$

第二步, 按照图 14 所示的连接方式, 将辅助量子比特设置为 $|+\rangle$ 态, 对新增加的 Z 型稳定子进行测量, 并且对 $|\bar{\Gamma}\rangle$ 涉及的物理比特添加纠正操作使得所有的稳定子得到满足, 然后对辅助量子比特

在 X 基下进行测量, 对其涉及的 X 型稳定子进行纠错操作, 从而得到纠缠态:

$$|\bar{\psi}'\rangle = \alpha |\bar{00}\rangle + \beta |\bar{11}\rangle. \quad (8)$$

第三步, 在 X 基下对 2D 表面码进行逻辑测量, 不同的测量结果 M 对应的逻辑态为

$$M = 0: |\bar{\psi}\rangle'' = \alpha |\bar{0}\rangle + \beta |\bar{1}\rangle,$$

$$M = 1: |\bar{\psi}\rangle'' = \alpha |\bar{0}\rangle - \beta |\bar{1}\rangle. \quad (9)$$

第四步, 根据测量结果对 3D 表面码添加逻辑 Z 纠正操作. $M = 0$, 即 2D 表面码塌缩为 $|\bar{+}\rangle$ 态时无需进行操作; $M = 1$, 即 2D 表面码塌缩为 $|\bar{-}\rangle$ 态时对 3D 表面码添加逻辑 Z 纠正操作即可得到相应逻辑量子态的 3D 表面码.

将 3D 表面码转换到 2D 表面码时, 步骤与上述过程类似, 不同之处在于: 将 2D 表面码编码为 $|\bar{+}\rangle$, 通过融合与分割操作得到纠缠态后, 对 3D 表面码在 X 基下进行逻辑测量, 并根据测量结果对 2D 表面码进行纠正即可.

4.3 基于融合与分割的码转换实现线路设计及仿真

由于 Qiskit 对量子比特数量的限制, 下面以 $[[12, 1, 2]]$ 3D 表面码蓝码和 $[[5, 1, 2]]$ 2D 表面码之间的转换为例进行线路设计及仿真. 两个表面码的连接方式以及量子比特编号如图 15 所示, 图中绿色球表示数据量子比特, 黑色的 13 号球表示辅助量子比特, 红色网格表示新增加的待测量的 Z 型稳定子, 蓝色网格表示 2D 表面码和 3D 表面码原 X 型稳定子的扩展部分, 联合测量的逻辑 Z 操作分别为: $\bar{Z}_{L1} = Z_{11}Z_{12}$, $\bar{Z}_{L2} = Z_{14}Z_{15}$.

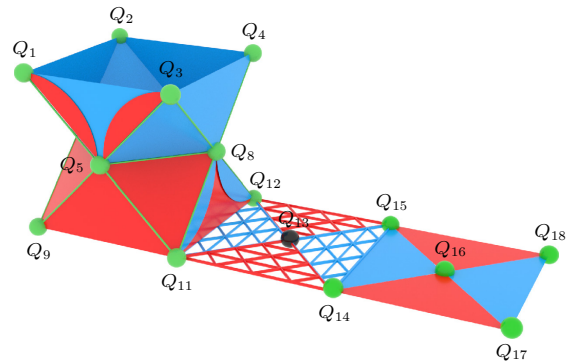


图 15 $[[5, 1, 2]]$ 2D 表面码与 $[[12, 1, 2]]$ 3D 表面码蓝码间码转换拼接示意图

Fig. 15. Diagram of code conversion between $[[5, 1, 2]]$ 2D and $[[12, 1, 2]]$ 3D surface codes.

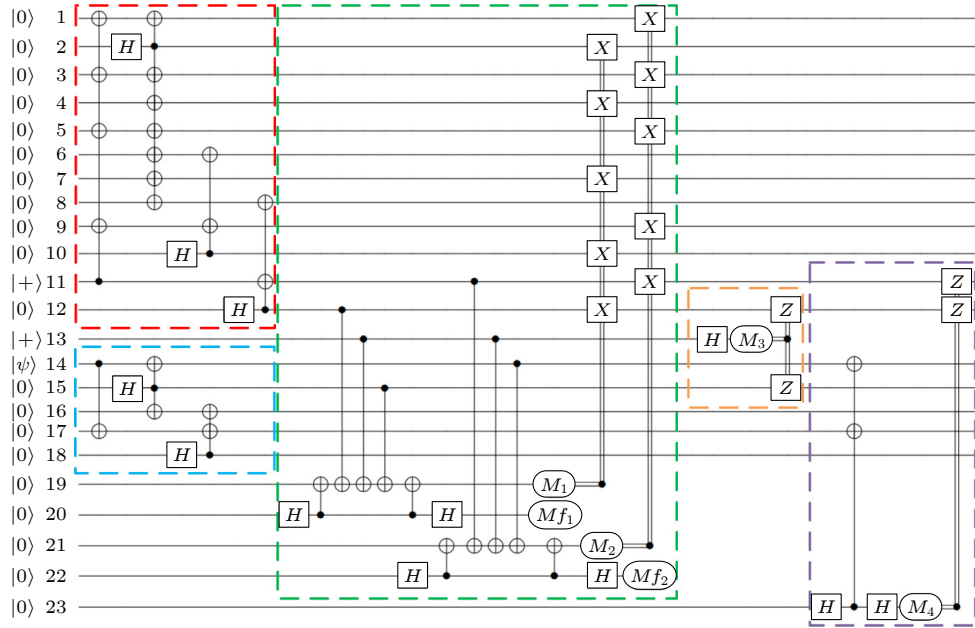


图 16 2D 表面码向 3D 表面码转换线路图

Fig. 16. Circuit diagram for 2D to 3D surface code conversion.

4.3.1 2D 表面码向 3D 表面码转换

参考 4.2 节所述步骤, 首先将辅助数据量子比特 Q_{13} 初始化为 $|+\rangle$ 态, 3D 表面码与 2D 表面码边界处的三端 X 型稳定子 $X_8X_{11}X_{12}$, $X_{14}X_{15}X_{16}$ 都变成了自动满足的四端稳定子 $X_8X_{11}X_{12}X_{13}$, $X_{13}X_{14}X_{15}X_{16}$; 第二步, 依次对两个表面码之间新生成的 Z 型稳定子 $Z_{12}Z_{13}Z_{15}$, $Z_{11}Z_{13}Z_{14}$ 进行测量, 测量结果分别记为 M_1 , M_2 , M_1 对应的纠正操作为 $X_{12}X_7X_{10}X_2X_4$, M_2 对应的纠正操作为 $X_{11}X_5X_9X_1X_3$, 施加纠正操作后实现光滑融合; 然后在 X 基下对辅助量子比特 Q_{13} 进行测量, 若测量结果为 1, 则添加 $Z_{12}Z_{15}$ 纠正操作使得三端 X 型稳定子 $X_8X_{11}X_{12}$, $X_{14}X_{15}X_{16}$ 得到满足, 以实现光滑分割操作得到纠缠态; 第三步, 在 X 基下对 2D 表面码进行逻辑测量; 第四步, 若测量结果为 1, 则添加逻辑 Z 纠正操作 $Z_{11}Z_{12}$, 实现 2D 表面码与 3D 表面码间的码转换. 整个编码线路如图 16 所示. 红色虚线框内对应的是编码实现 $[[12, 1, 2]]$ 3D 表面码蓝码, 蓝色虚线框内对应的是编码实现 $[[5, 1, 2]]$ 2D 表面码; 绿色虚线框内对应的是测量两个表面码之间新生成的 Z 型稳定子并进行对应纠正操作; 橙色虚线框内对应的是测量辅助量子比特并根据测量结果进行对应纠正操作; 紫色虚线框内对应的是对 2D 表面码进行逻辑测量并根据测量结果进行对应纠正操作.

分别设置 2D 表面码的信息比特 Q_{14} 为 $|0\rangle$, $|1\rangle$, $|+\rangle$ 和 $|-\rangle$ 对所设计的码转换线路进行仿真验证, 运行编码线路, 将得到的码字结果与附录 A 中表 A3 $[[12, 1, 2]]$ 3D 表面码蓝码的码字进行对照, 结果表明 2D 表面码编码的四种量子态 $|0\rangle$, $|1\rangle$, $|+\rangle$ 和 $|-\rangle$ 通过码转换线路传输到了 3D 表面码蓝码中. 当设置 $[[5, 1, 2]]$ 2D 表面码的信息比特为 $0.6|0\rangle + 0.8|1\rangle$ 时, 将线路运行 102400 次并统计运行结果, 得到 M_1, M_2, M_3, M_4 的测量结果如图 17 所示, 得到 3D 表面码码字概率分布直方图如图 18 所示, 可以看出 M_1, M_2, M_3, M_4 是随机塌缩的且得到的码字结果概率分布符合预期, 成功实现了任意量子态 2D 表面码向 3D 表面码的转换.

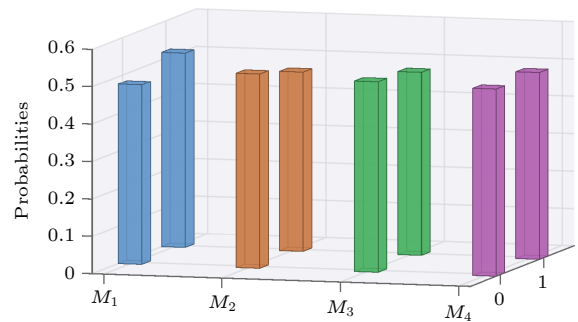


图 17 测量结果 M_1, M_2, M_3, M_4 的概率分布图

Fig. 17. Probability distribution diagram for M_1, M_2, M_3, M_4 .

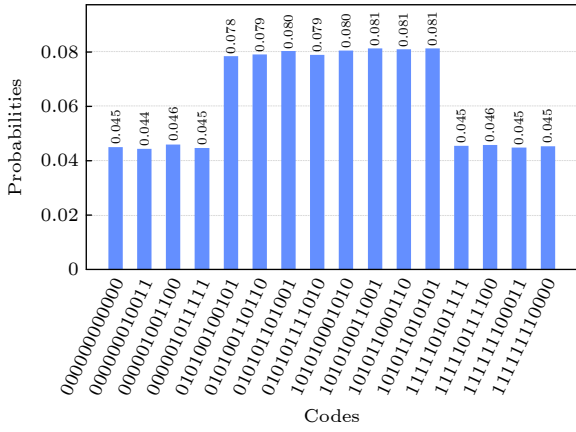


图 18 2D 表面码 $0.6|\bar{0}\rangle + 0.8|\bar{1}\rangle$ 转换为 3D 表面码的概率分布直方图
 Fig. 18. Probability distribution diagram of the inter-code conversion from a 2D surface code state $0.6|\bar{0}\rangle + 0.8|\bar{1}\rangle$ to a 3D surface code.

4.3.2 3D 表面码向 2D 表面码转换

3D 表面码向 2D 表面码转换的编码线路如图 19 所示, 图中不同颜色虚线框内编码线路对应实现的功能与 2D 表面码向 3D 表面码转换相同. 不同的是, 3D 表面码的信息比特 Q_{11} 可以初始化为任意量子态 $\alpha|0\rangle + \beta|1\rangle$, 2D 表面码的信息比特 Q_{14} 和辅助数据量子比特 Q_{13} 均初始化为 $|+\rangle$ 态; 绿色虚线框内 M_1 对应的纠正操作变为 $X_{15}X_{18}$, M_2 对应的纠正操作变为 $X_{14}X_{17}$; 紫色虚线框内对 3D 表面码进行逻辑操作 $X_1X_3X_5X_9X_{11}$ 的测量使之塌缩到 $|+\rangle$ 或者 $|-\rangle$ 态; M_4 测量结果为 1 时

对 2D 表面码添加逻辑 Z 纠正操作 $Z_{14}Z_{15}$.

为方便结果验证, 这里先给出 $[[5, 1, 2]]$ 2D 表面码 $|\bar{0}\rangle$ 和 $|\bar{1}\rangle$ 的码字, 如表 2 所列.

表 2 $[[5, 1, 2]]$ 2D 表面码码字
 Table 2. Codeword in the $[[5, 1, 2]]$ 2D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{0}\rangle$	$ \bar{1}\rangle$
+ 00000)	+ 01001)	+ 00111)	+ 01110)
+ 11011)	+ 10010)	+ 11100)	+ 10101)

分别设置 $[[12, 1, 2]]$ 3D 表面码蓝码的信息量子位为 $|0\rangle$, $|1\rangle$, $|+\rangle$ 和 $|-\rangle$, 运行编码线路得到的码字结果如表 3 所列. 将结果与表 2 进行对照, 易验证所设计的量子线路成功将 3D 表面码蓝码编码的四种量子态 $|\bar{0}\rangle$, $|\bar{1}\rangle$, $|\bar{+}\rangle$ 和 $|\bar{-}\rangle$ 传输到了 2D 表面码中.

表 3 3D 表面码向 2D 表面码转换结果

Table 3. Conversion results from 3D surface code to 2D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{+}\rangle$	$ \bar{-}\rangle$
+ 00000)	+ 01001)	+ 00000)	+ 00000)
+ 00111)	+ 01110)	+ 00111)	+ 00111)
+ 11011)	+ 10010)	+ 01001)	- 01001)
+ 11100)	+ 10101)	+ 01110)	- 01110)
		+ 10010)	- 10010)
		+ 10101)	- 10101)
		+ 11011)	+ 11011)
		+ 11100)	+ 11100)

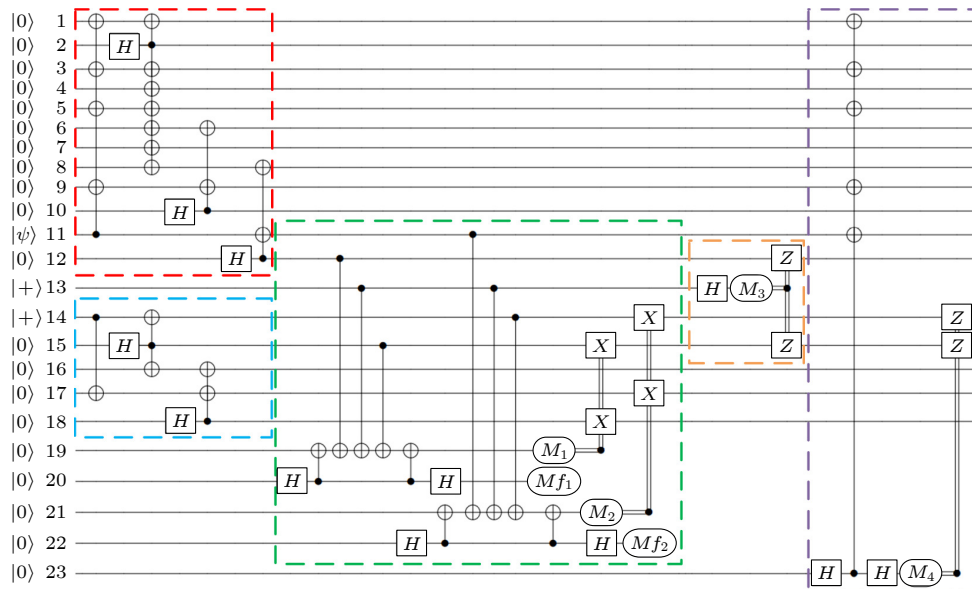


图 19 3D 表面码向 2D 表面码转换线路图

Fig. 19. Circuit diagram for 3D to 2D surface code conversion.

当设置 3D 表面码蓝码的信息比特为 $0.6|\bar{0}\rangle + 0.8|\bar{1}\rangle$ 时, 将线路运行 102400 次并统计运行结果, 得到的码字概率分布图如图 20 所示, 可以看到码字的概率分布符合预期, 成功实现了任意逻辑量子态在 2D 表面码与 3D 表面码之间的码转换。

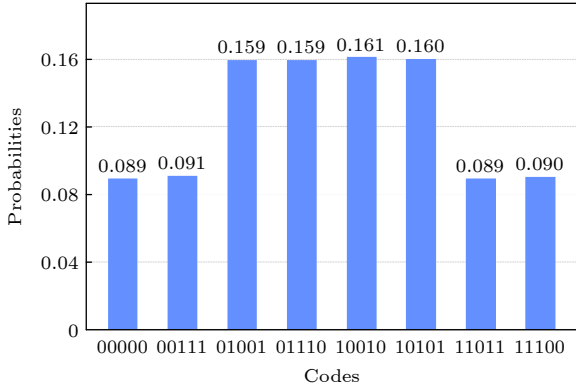


图 20 3D 表面码 $0.6|\bar{0}\rangle + 0.8|\bar{1}\rangle$ 转换为 2D 表面码的概率分布直方图
 Fig. 20. Histogram of the probability distribution for the input $0.6|\bar{0}\rangle + 0.8|\bar{1}\rangle$.

5 讨论

对于码距为 d 的表面码, 转换到 3D 表面码进行 CCZ 门需要的量子比特数目为 $9d^3 - 6d^2 + 6d + 3$, 文献 [13] 提出的基于魔幻态工厂实现 CCZ 门需要的量子比特数目为 $72d^2$, 当码距小于 9 时, 转换到 3D 表面码可以节省物理比特. 文献 [15] 进一步改进了魔幻态蒸馏方案, 资源略有减少, 随着魔幻态蒸馏方案性能的提升, 基于码转换实现 CCZ 门的优势码距可能会进一步减小. 这个比较是针对一次 CCZ 门的实现进行的, 在进行量子计算时, 如果利用魔幻态蒸馏的方法, 对于任意一个 CCZ 门都需要 $72d^2$ 个物理比特, 但是转换到 3D 表面码以后, 后续的 CCZ 门都可以横截执行, 不需要消耗新的物理比特. 因此在实际中可以结合线路的优化设计尽量在一个码字空间执行更多的量子门, 从而避免频繁码字转换, 提升整个线路的效率.

容错性是量子计算关注的至关重要的问题. 利用“猫态”能够进行容错的稳定子测量 [35-37], 但消耗的量子比特数目较多, 图 16 和图 19 中稳定子的测量均引入了标记比特 Mf_1 或者 Mf_2 来检测稳定子测量过程中辅助比特是否发生了 Z 错误 [26-28].

如果某一标记比特的测量结果为 1, 则转换线路之后进行稳定子测量和纠正时需结合标记比特准确判断错误类型. 逻辑操作测量过程中如果辅助比特发生了 X 错误, 则只会给原始码块带来错误, 不会影响逻辑测量的结果和纠正操作, 从而不需要引入标记比特检测此错误. 受仿真平台限制, 本文仅对码距为 2 的编码实例进行了验证, 该码距并不具备实际纠错能力, 因此未对噪声环境下的性能进行测试. 若计算资源允许, 可进一步仿真码距为 3 及含噪声情况下的码转换性能. 此外, 本文仅探讨了表面码之间的转换, 其中 H 门的实现仍需借助辅助操作, 而 Steane 码 [38] 和 Reed-Muller 码分别具备横截 H 门和横截 T 门 [39-41], 后续研究可探索更多类型量子纠错码之间的转换机制, 以简化容错通用门集的实现. 目前, 3D 表面码的物理实现仍面临三维比特阵列制备与控制复杂度高等挑战. 未来工作可致力于高码距含噪编码线路的设计, 以及跨码型转换机制的深入研究, 推动容错通用量子计算的实际应用.

6 结论

本研究系统探讨了 3D 表面码的编码线路设计, 逻辑门集性质及其与 2D 表面码之间的转换方案, 主要获得以下结论: 首先, 基于稳定子实现的方法设计了不同颜色 3D 表面码的编码线路, 并通过时隙优化有效降低了门序列深度, 能够提升噪声环境下的编码正确率, 进而实现高效逻辑态制备. 其次, 结合 3D 表面码横截 CCZ 门的性质理论证明了其 CZ 门的横截性质. 最后, 基于融合与分割操作提出了 2D 表面码和 3D 表面码之间的码转换方案, 并完成了该方案的量子线路设计与仿真验证. 结果表明, 本研究成功实现了 2D 表面码与 3D 表面码间任意逻辑态的转换, 提出了可在以 2D 表面码为主的计算流程中动态引入 3D 表面码实现非 Clifford 门的容错通用量子计算实现方案, 在线路优化设计时, 尽量使得能在同一个码字空间执行的量子门挨在一起, 从而避免频繁码字转换, 提升整体效率. 然而, 本研究尚未考虑该方案在噪声环境下的性能、容错机制和物理实现等, 未来工作可围绕含噪性能仿真及跨类型量子纠错码转换等方面展开, 进一步推动容错通用量子计算发展.

附录 A

表 A1 $[[12, 1, 2]]$ 3D 表面码红码字
Table A1. Red codeword in the $[[12, 1, 2]]$ 3D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{\tau}\rangle$	$ \bar{\tau}'\rangle$
+ 000000000000⟩	+ 001100010011⟩	+ 000000000000⟩	+ 000000000000⟩
+ 000011111111⟩	+ 001111101100⟩	+ 000011111111⟩	+ 000011111111⟩
+ 010100100000⟩	+ 011000110011⟩	+ 001100010011⟩	- 001100010011⟩
+ 010111011111⟩	+ 011011001100⟩	+ 001111101100⟩	- 001111101100⟩
+ 101001111111⟩	+ 100101101100⟩	+ 010100100000⟩	+ 010100100000⟩
+ 101010000000⟩	+ 100110010011⟩	+ 010111011111⟩	+ 010111011111⟩
+ 111101011111⟩	+ 110001001100⟩	+ 011000110011⟩	- 011000110011⟩
+ 111110100000⟩	+ 110010110011⟩	+ 011011001100⟩	- 011011001100⟩
		+ 100101101100⟩	- 100101101100⟩
		+ 100110010011⟩	- 100110010011⟩
		+ 101001111111⟩	+ 101001111111⟩
		+ 101010000000⟩	+ 101010000000⟩
		+ 110001001100⟩	- 110001001100⟩
		+ 110010110011⟩	- 110010110011⟩
		+ 111101011111⟩	+ 111101011111⟩
		+ 111110100000⟩	+ 111110100000⟩

表 A2 $[[12, 1, 2]]$ 3D 表面码绿码字
Table A2. Green codeword in the $[[12, 1, 2]]$ 3D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{\tau}\rangle$	$ \bar{\tau}'\rangle$
+ 000000000000⟩	+ 000000011111⟩	+ 000000000000⟩	+ 000000000000⟩
+ 000100110001⟩	+ 000100111110⟩	+ 000000001111⟩	- 000000001111⟩
+ 001010010010⟩	+ 001010011101⟩	+ 000100110001⟩	+ 000100110001⟩
+ 001110100011⟩	+ 001110101100⟩	+ 000100111110⟩	- 000100111110⟩
+ 010001100100⟩	+ 010001101011⟩	+ 001010010010⟩	+ 001010010010⟩
+ 010101010101⟩	+ 010101011010⟩	+ 001010011101⟩	- 001010011101⟩
+ 011011110110⟩	+ 011011110011⟩	+ 001110100011⟩	+ 001110100011⟩
+ 011111000111⟩	+ 011111001000⟩	+ 001110101100⟩	- 001110101100⟩
+ 100011001000⟩	+ 100011000111⟩	+ 010001100100⟩	+ 010001100100⟩
+ 100111111001⟩	+ 100111110110⟩	+ 010001101011⟩	- 010001101011⟩
+ 101001011010⟩	+ 101001010101⟩	+ 010101010101⟩	+ 010101010101⟩
+ 101101101011⟩	+ 101101100100⟩	+ 010101011010⟩	- 010101011010⟩
+ 110010101100⟩	+ 110010100011⟩	+ 011011110110⟩	+ 011011110110⟩
+ 110110011101⟩	+ 110110010010⟩	+ 011011111001⟩	- 011011111001⟩
+ 111000111110⟩	+ 111000110001⟩	+ 011111000111⟩	+ 011111000111⟩
+ 111100001111⟩	+ 111100000000⟩	+ 011111001000⟩	- 011111001000⟩
		+ 100011000111⟩	- 100011000111⟩
		+ 100011001000⟩	+ 100011001000⟩
		+ 100111110110⟩	- 100111110110⟩
		+ 100111111001⟩	+ 100111111001⟩
		+ 101001010101⟩	- 101001010101⟩
		+ 101001011010⟩	+ 101001011010⟩

表 A2 (续) $[[12, 1, 2]]$ 3D 表面码绿码码字
Table A2 (continued). Green codeword in the $[[12, 1, 2]]$ 3D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{\oplus}\rangle$	$ \bar{\ominus}\rangle$
		+ 101101100100⟩	- 101101100100⟩
		+ 101101101011⟩	+ 101101101011⟩
		+ 110010100011⟩	- 110010100011⟩
		+ 110010101100⟩	+ 110010101100⟩
		+ 110110010010⟩	- 110110010010⟩
		+ 110110011101⟩	+ 110110011101⟩
		+ 111000110001⟩	- 111000110001⟩
		+ 111000111110⟩	+ 111000111110⟩
		+ 111100000000⟩	- 111100000000⟩
		+ 111100001111⟩	+ 111100001111⟩

表 A3 $[[12, 1, 2]]$ 3D 表面码蓝码码字
Table A3. Blue codeword in the $[[12, 1, 2]]$ 3D surface code.

$ \bar{0}\rangle$	$ \bar{1}\rangle$	$ \bar{\oplus}\rangle$	$ \bar{\ominus}\rangle$
+ 000000000000⟩	+ 010100100101⟩	+ 000000000000⟩	+ 000000000000⟩
+ 000000010011⟩	+ 010100110110⟩	+ 000000010011⟩	+ 000000010011⟩
+ 000001001100⟩	+ 010101101001⟩	+ 000001001100⟩	+ 000001001100⟩
+ 000001011111⟩	+ 010101111010⟩	+ 000001011111⟩	+ 000001011111⟩
+ 111110101111⟩	+ 101010001010⟩	+ 1010100100101⟩	- 010100100101⟩
+ 111110111100⟩	+ 101010011001⟩	+ 010100110110⟩	- 010100110110⟩
+ 111111000011⟩	+ 101011000110⟩	+ 010101101001⟩	- 010101101001⟩
+ 111111100000⟩	+ 101011010101⟩	+ 010101111010⟩	- 010101111010⟩
		+ 101010001010⟩	- 101010001010⟩
		+ 101010011001⟩	- 101010011001⟩
		+ 101011000110⟩	- 101011000110⟩
		+ 101011010101⟩	- 101011010101⟩
		+ 111110101111⟩	+ 111110101111⟩
		+ 111110111100⟩	+ 111110111100⟩
		+ 111111000011⟩	+ 111111000011⟩
		+ 111111100000⟩	+ 111111100000⟩

参考文献

[1] Wang M H, Hao S H, Qin Z Z, Su X L 2022 *Acta Phys. Sin.* **71** 160305 [王美红, 郝树宏, 秦忠忠, 苏晓龙 2022 物理学报 **71** 160305]

[2] Zhou W H, Wang Y, Weng W K, Jin X M 2022 *Acta Phys. Sin.* **71** 240302 [周文豪, 王耀, 翁文康, 金贤敏 2022 物理学报 **71** 240302]

[3] Zhang S H, Zhang X D, Li L Z 2021 *Acta Phys. Sin.* **70** 210301 [张诗豪, 张向东, 李绿周 2021 物理学报 **70** 210301]

[4] Fan H 2023 *Acta Phys. Sin.* **72** 070303 [范桁 2023 物理学报 **72** 070303]

[5] Kitaev A Y 2003 *Ann. Phys.* **303** 2

[6] Bravyi S B, Kitaev A Y 1998 arXiv: 9811052 [quant-ph]

[7] Freedman M H, Meyer D A 2001 *Found. Comput. Math.* **1** 325

[8] Raussendorf R, Harrington J 2007 *Phys. Rev. Lett.* **98** 190504

[9] Fowler A G, Stephens A M, Groszkowski P 2009 *Phys. Rev. A* **80** 052312

[10] Gottesman D 1998 arXiv: 9807006 [quant-ph]

[11] Brown B J, Laubscher K, Kesselring M S, Wootton J R 2017 *Phys. Rev. X* **7** 021029

[12] Tasler S, Old J, Heunisch L, Feulner V, Eckstein T, Müller M, Hartmann M J 2025 arXiv: 2506.09028 [quant-ph]

[13] Gidney C, Fowler A G 2019 *Quantum* **3** 135

[14] Fowler A G, Gidney C 2019 arXiv: 1808.06709 [quant-ph]

[15] Fazio N, Webster M, Cai Z 2025 arXiv: 2501.10291 [quant-ph]

[16] Vasmer M, Browne D E 2019 *Phys. Rev. A* **100** 012312

[17] Bombín H, Martin Delgado M A 2007 *Phys. Rev. Lett.* **98** 160502

[18] Castelnuovo C, Chamon C 2008 *Phys. Rev. B* **78** 155120

[19] Scruby T R, Browne D E, Webster P, Vasmer M 2022 *Quantum* **6** 721

[20] Huang E, Pesah A, Chubb C T, Vasmer M, Dua A 2023 *PRX Quantum* **4** 030338

- [21] Lee C, Hu Y, Cho G Y, Watanabe H 2025 *Phys. Rev. B* **112** 155136
- [22] Gottesman D 1997 arXiv: 9705052 [quant-ph]
- [23] Dennis E, Kitaev A, Landahl A, Preskill J 2002 *J. Math. Phys.* **43** 4452
- [24] Horsman D, Fowler A G, Devitt S, Meter R V 2012 *New J. Phys.* **14** 123011
- [25] Fowler A G, Mariantoni M, Martinis J M, Cleland A N 2012 *Phys. Rev. A* **86** 032324
- [26] Quan D X, Liu C S, Lv X J, Pei C X 2022 *Entropy* **24** 1107
- [27] Chamberland C, Beverland M E 2018 *Quantum* **2** 53
- [28] Tansuwannont T, Chamberland C, Leung D 2020 *Phys. Rev. A* **101** 012342
- [29] Bravyi S, Hastings M B, Verstraete F 2006 *Phys. Rev. Lett.* **97** 050401
- [30] Higgott O, Wilson M, Hefford J, Dborin J, Hanif F, Burton S, Browne D E 2021 *Quantum* **5** 517
- [31] Bravyi S, Kitaev A 2005 *Phys. Rev. A* **71** 022316
- [32] Shor P 1996 *Proceedings of 37th Conference on Foundations of Computer Science* (Burlington, USA, October 14–16, 1996) pp56–65
- [33] Wang C, Harrington J, Preskill J 2003 *Ann. Phys.* **303** 31
- [34] Poirson C, Roffe J, Booth R I 2025 arXiv: 2505.01370 [quant-ph]
- [35] Darmawan A S, Poulin D 2018 *Phys. Rev. E* **97** 051302
- [36] Divincenzo D P, Aliferis P 2007 *Phys. Rev. Lett.* **98** 020501
- [37] Stephens A M 2014 arXiv: 1402.3037 [quant-ph]
- [38] Steane A 1996 *Phys. Rev. Lett.* **77** 793
- [39] Zeng B, Cross A, Chuang I L 2011 *IEEE Trans. Inf. Theory* **57** 6272
- [40] Anderson J T, Duclos-Cianci G, Poulin D 2014 *Phys. Rev. Lett.* **113** 080501
- [41] Campbell E T, Terhal B M, Vuillot C 2017 *Nature* **549** 172

Fault-tolerant universal quantum computation via surface code conversion *

QUAN Dongxiao ^{1)2)†} WANG Zhenli ¹⁾ CHENG Longxiang ¹⁾

¹⁾ (School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)

²⁾ (Collaborative Innovation Center of Quantum Information of Shaanxi Province, Xidian University, Xi'an 710071, China)

(Received 30 September 2025; revised manuscript received 9 December 2025)

Abstract

Quantum superposition enables the inherent parallelism of quantum computing, which demonstrates great potential for solving specific problems, but its practical implementation is hindered by high susceptibility to noise. Therefore, encoding logical qubits with quantum error-correcting codes and implementing universal quantum operations on these logical qubits is a feasible way toward fault-tolerant quantum computing. Because its stabilizers involve only neighboring qubits, the surface code achieves a high fault-tolerance threshold, making it a leading class of topological quantum error-correcting codes. The 2-dimensional (2D) surface code is well-suited for implementing Clifford-group gates, whereas the 3D surface code natively supports transversal non-Clifford gates such as CCZ and CZ. In this work, to fully leverage the transversal gates of both 2D and 3D surface codes, we present a scheme for implementing fault-tolerant quantum computation via code conversion.

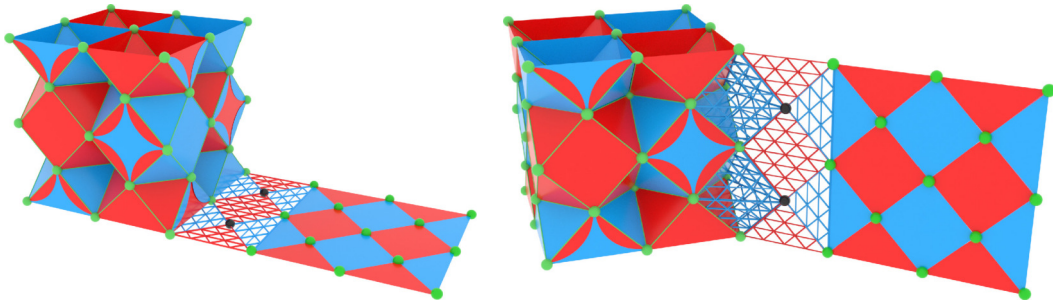
By studying the structure and stabilizers of the 3D surface code, this paper designs encoding circuits through stabilizer implementation, followed by time-slot optimization to reduce circuit depth and improve fidelity. Simulation results confirm the correctness and superiority of the encoding circuit design. We demonstrate the realization of a transversal CZ gate and, building on the respective transversal gates of the two surface codes, propose a fault-tolerant universal quantum computation framework based on code conversion. During quantum circuit design, frequent alternation between CCZ gates, H gates, and CNOT gates across differently-colored surface codes should be avoided. Throughout fault-tolerant computation, quantum operations should be executed as extensively as possible within a single surface code. A switch to another surface code should only occur when encountering an operation that cannot be executed transversally. Furthermore, we

* Project supported by the National Natural Science Foundation of China (Grant Nos. 62471350, 62001351) and the Key Research and Development Program of Shaanxi Province, China (Grant No. 2019ZDLGY09-02).

† Corresponding author. E-mail: dxquan@xidian.edu.cn

present a code conversion scheme that enables dynamic switching between 2D and 3D encodings during computation. This scheme, based on lattice merging and splitting operations, is illustrated with a code distance of 3. In this process, two surface codes are first merged through the initialization of data qubits along with the measurement and correction of auxiliary stabilizers. Subsequently, the logical information is transferred to the target surface code via logical measurements performed on the original surface code and correcting operations applied to the target surface code.

Limited by the qubit availability in the Qiskit platform, the proposed schemes are validated through simulations of distance-2 codes, which successfully demonstrate correct encoding and bidirectional conversion between 2D and 3D logical states. Our results indicate that the hybrid approach leverages the efficiency of 2D codes for Clifford operations and the transversal capabilities of 3D codes for non-Clifford gates. This work provides a new pathway toward fault-tolerant universal quantum computation.



Keywords: fault-tolerant universal quantum computation, quantum error correction code, quantum surface code, code conversion

DOI: [10.7498/aps.75.20251344](https://doi.org/10.7498/aps.75.20251344)

CSTR: [32037.14.aps.75.20251344](https://cstr.org/cstr/32037.14.aps.75.20251344)



基于码转换的表面码容错通用量子计算实现方法

权东晓 王贞理 程龙祥

Fault-tolerant universal quantum computation via surface code conversion

QUAN Dongxiao WANG Zhenli CHENG Longxiang

引用信息 Citation: *Acta Physica Sinica*, 75, 080601 (2026) DOI: 10.7498/aps.75.20251344

CSTR: 32037.14.aps.75.20251344

在线阅读 View online: <https://doi.org/10.7498/aps.75.20251344>

当期内容 View table of contents: <https://wulixb.iphy.ac.cn>

您可能感兴趣的其他文章

Articles you may be interested in

多逻辑比特表面码结构设计及其逻辑CNOT门实现

Structure design and logical CNOT implementation of multi-logical-qubits surface code

物理学报. 2024, 73(4): 040304 <https://doi.org/10.7498/aps.73.20231138>

通用量子计算模型: 一个资源理论的视角

Universal quantum computing models: a perspective of resource theory

物理学报. 2024, 73(22): 220302 <https://doi.org/10.7498/aps.73.20240893>

量子计算纠错取得突破性进展

Breakthrough of error correction in quantum computing

物理学报. 2023, 72(7): 070303 <https://doi.org/10.7498/aps.72.20230330>

连续变量量子计算和量子纠错研究进展

Research advances in continuous-variable quantum computation and quantum error correction

物理学报. 2022, 71(16): 160305 <https://doi.org/10.7498/aps.71.20220635>

基于超导量子系统的量子纠错研究进展

Advances in quantum error correction based on superconducting quantum systems

物理学报. 2022, 71(24): 240305 <https://doi.org/10.7498/aps.71.20221824>

离子阱量子计算规模化的研究进展

Research progress of ion trap quantum computing

物理学报. 2023, 72(23): 230302 <https://doi.org/10.7498/aps.72.20231128>