

开场白

亲爱的老师、助教与各位同学们大家好。我们小组由我、鲁睿与黄书鸿共同完成了对于量子纠错编码理论及其应用的研究，现在将研究成果做如下汇报。我们的汇报主要有三个主题内容：首先是对于量子纠错理论基础的综述，而后是对 Surface Code 与 Subsystem Code 的具体讨论。在此框架下，分为了如下八个部分。

量子纠错基础

量子噪声定义

量子噪声产生的主要原因是实验室环境下的量子系统并不是一个完美的孤立系统，总是会与环境发生相互作用，在这个大系统中的演化，就表现为实验系统中的噪声。

我们可以假设初始状态下系统和环境之间是没有纠缠的，状态分别表示为 ρ 和 ρ_{env} ，且在系统和环境组成的大系统中经历演化 U ，那么从实验室系统的角度来看， ρ 经历如 1 式的变化：

$$E(\rho) = \text{tr}_{env}[U(\rho \otimes \rho_{env})U^\dagger] \quad (1)$$

由于我们总是可以引入一个额外的系统对 ρ_{env} 进行纯化，并且假设它在纯化后表示状态 $|0\rangle$ ，我们总是可以假设环境的初始状态为 $|0\rangle$ 。所以以上的作用总是可以写作如 2 式的 operator sum 的形式：

$$E(\rho) = \sum_i E_i \rho E_i^\dagger \quad (2)$$

它显然是线性的，若 ρ 为正， $E(\rho)$ 也为正，并且保持 trace 不变。由 $\{E_i\}$ 构成的集合就称之为错误。

所有的有噪声的量子信道都可以用一个保持半正定性和保迹的 operator sum 来表示。在量子纠错的理论中，我们最为关注的是一类称为 Pauli 错误的量子错误，它们作用在单个量子比特上，通常可以被表示为 3 式的形式：

$$E(\rho) = (1 - p_X - p_Y - p_Z)\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z \quad (3)$$

可以验证 3 式符合保半正定性与保迹的性质。这还可以解释为对于该量子比特，有 p_σ 的概率发生 σ 错误。

Shor 码

有一种简单的量子码能针对单量子比特上任意差错进行保护。这种码就是以其发明者命名的 Shor 码，它是量子比特相位翻转码和三量子比特比特翻转码的组合。我们首先用相位翻转码来编码量子比特： $|0\rangle \rightarrow |+++ \rangle, |1\rangle \rightarrow |-- \rangle$ 。其次，我们用三个量子比特比特翻转码来编码这些量子比特中的每个： $|+\rangle$ 编码为 $(|000\rangle + |111\rangle)/\sqrt{2}$ ， $|-\rangle$ 编码为 $(|000\rangle - |111\rangle)/\sqrt{2}$ 。如此编码，得到一个 9 量子比特，其码字如 4a 和 4b 所示：

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}} \\ |1\rangle &\rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}} \end{aligned} \quad (4)$$

Shor 码能对任意一个量子比特上的相位翻转差错和比特翻转差错进行保护。事实上，Shor 码对单量子比特上的保护要比比特翻转和相位翻转差错强大的多——Shor 码可对完全任意的差错进行保护，只要这些差错仅影响单个单量子比特。

具体的证明将在附录中展示，此处不再赘述。

量子纠错理论

量子纠错理论自然地推广了由 Shor 码引发的思想。量子状态通过酉运算被编码为量子纠错码，其形式定义为某个较大 Hilbert 空间中的一个子空间 C 。为方便起见，我们采用符号 P 表示到码空间 C 上的投影算子；且对三量子比特比特翻转码， P 表示如右式。在编码以后，这个码会受到噪声的影响，紧接着执行差错症状测量以检测所出现的差错类型。一旦差错症状确定，恢复运算就会执行，以使量子系统回到这个码的原来状态。

不同的差错症状对应于整个 Hilbert 空间中的保形子空间与正交子空间。这些子空间必是正交的，否则它们就不能被差错症状测量可靠地区分。进而，由于到不同子空间的差错映射必将正交码字映射到正交状态，因此这些不同子空间必为原来码空间的保形版本，这样就能使其从差错中恢复。这个直观的理解基本上就是量子纠错条件的要旨。

量子纠错假定

对于量子纠错理论，我们仅只作两个很宽泛的假定：噪声由量子运算 ε 所描述，整个纠错方法由我们称之为纠错运算的一个保迹量子运算 \mathcal{R} 承担。这个纠错运算把我们上面称为差错检测和恢复的两个步骤合并。为确保纠错是成功的，我们要求对任何状态 ρ ，其支撑集位于码空间 C 中，有 5 式的关系：

$$(\mathcal{R} \circ \varepsilon)(\rho) \propto \rho \quad (5)$$

量子纠错条件是一个简单方程组，它们可被检验以确定量子纠错码是否能对抗特殊类型的噪声 ε 。我们将应用这些条件来构造大量的量子码，并将研究量子纠错码的一些普遍性质。

量子纠错条件

至于什么样的量子噪声能够得到纠正。令 C 为一个量子码， P 为到 C 的投影算子。设 ε 为具有运算元 $\{E_i\}$ 的量子运算。则纠正 C 上 ε 的纠错运算 R 存在的充分必要条件为，对某个复数 Hermite 矩阵 α 存在 6 式所示的关系：

$$PE_i^\dagger E_j P = \alpha_{ij} P \quad (6)$$

我们称运算元 $\{E_i\}$ 为噪声 ε 的差错，且如果这样一个 \mathfrak{R} 存在，我们就说 $\{E_i\}$ 组成一个可纠正的差错集合。

差错离散化

我们已经讨论了针对一种特定噪声过程的量子信息的保护。但是，一般来说，我们并不准确知道量子系统遭受的是什么噪声。如果特定码 C 和纠错运算 \mathcal{R} 能被用于针对全部类型噪声过程的保护，那么这将会非常有用。幸运的是，量子纠错条件很容易用于严格地提供这类保护。

定理：设 C 为量子码， \mathcal{R} 为符合量子纠错条件的纠错运算，用以从具有算子元 $\{E_i\}$ 的噪声过程中恢复。设 \mathcal{F} 为具有运算元 $\{F_j\}$ 的量子运算，运算元 $\{F_j\}$ 为 E_i 的线性组合，即对某个复数矩阵 m_{ji} 有 $F_j = \sum_i m_{ji} E_i$ 。那么，纠错运算 \mathcal{R} 也可对码 C 上的噪声过程 \mathcal{F} 的作用来进行纠正。

由此，一噪声过程 ε ，其运算元由这些差错算子 $\{E_i\}$ 的线性组合而成，都将通过恢复运算 \mathcal{R} 被纠正。

按此观点，设 ε 为作用于单量子比特上的量子运算。那么，其每个运算元 $\{E_i\}$ 都可以被写成为 Pauli 矩阵 $\sigma_0, \sigma_1, \sigma_2, \sigma_3$ 的线性组合。

量子码构建

CSS 码

有了纠错的理论，我们来讨论传统的量子纠错码大类，第一个例子是 Calderbank-Shor-Steane 编码，通常被简称为 CSS 码，这是以码发明者姓名的首字母所命名的。CSS 码是更为一般类稳定子码的一个重要子类。

设 C_1 和 C_2 为 $[n, k_1]$ 和 $[n, k_2]$ 经典线性码，使有 C_2 的编码空间是 C_1 的子空间且 C_1 和 C_2 互为对偶空间，两者可纠正 $k_1 - k_2$ 个差错。通过下面的构造，我们将要定义能纠正 $k_1 - k_2$ 个量子比特上差错的一个 $[n, k_1 - k_2]$ 量子码 CSS(C_1, C_2)。即 C_2 上 C_1 的 CSS 码。设 $x \in C_1$ 且为 C_1 中的任一码字，那么，我们就定义量子状态 $|x + C_2\rangle$ 如 9 式所示：

$$|x + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle \quad (7)$$

对于 CSS 编码的具体阐释也在附录中写道。

稳定子码

在开始 Surface Code 的讨论前，我们需要定义稳定子与稳定子码。设 S 为 G_n 的一个子群，定义 V_S 为由 S 的每个元所固定的 n 量子比特状态的集合。进而， V_S 为由 S 所稳定的向量空间， S 被称为空间 V_S 的稳定子，因为 V_S 的每个元为在 S 中元的作用下是稳定的。

如果群 G 的每个元都可被写为序列 g_1, \dots, g_l 的元的乘积，群 G 中的一组元 g_1, \dots, g_l 是 G 的生成元，写为 $G = \langle g_1, \dots, g_l \rangle$ 。大小为 $|G|$ 的一个群 G 具有最多 $\log(|G|)$ 个的一组生成元。进而，为看清一个特定的向量可用群 S 来稳定，我们只需要检验向量可用 S 的生成元稳定。

并非 Pauli 群的任一子群 S 都可被用作非平凡向量空间的稳定子。为使 S 稳定一个非平凡向量空间 V_S , S 必须满足两个必要条件：

1. S 的元可对易；
2. $-I$ 不是 S 的一个元。

Surface code 的构造

接下来，我们对稳定子码的一个变种 Surface code。Surface code 是一种量子纠错码，其构造如图所示，其中白色圆点表示 data qubit，黑色圆点表示 measure qubit，measure qubit 又分为两种，用 X 和 Z 表示，它们与周围的 4 个 data qubit 连接（在边界处的是 3 个）

这些测量对应的可观测量就是该稳定子码对应的稳定子群 S 的生成元，但是符号尚未确定。一般我们不会事先确定这些生成元的符号，而是经过多轮测量，等到其测量结果稳定后，根据测量结果决定。

单比特错误

我们分别阐述 Surface Code 的纠错效果，首先考虑单比特的 Pauli 错误。

若在某一个 data qubit 上出现了 X 错误，则与其相邻的两个 Z -measure qubit 的测量结果会发生翻转，十分容易据此定位错误。

如果是某个 measure qubit 上出现 X 错误，注意到这样的错误最多影响一轮测量的一个测量结果，很容易与 data qubit 上的 X 错误区分。

对于 Z 错误，也可以如此分析。

因此，Surface code 可以有效抵抗单比特错误。

对于多比特错误，在测量结果上表现为多个测量位置的翻转，暂且假定测量过程无误，我们可以通过搜索找出匹配这种错误症状的 Pauli 错误的组合，然后假定它是 weight 最小的一个，这是因为一般来说错误发生的概率都较小。

定义 X_L 和 Z_L 算子

接下来我们阐述 X_L 和 Z_L 算子的定义。沿着上图蓝（红）线，在经过的 data qubit 上作用 X （ Z ），定义了 X_L （ Z_L ）算子。

这样定义的逻辑算子在乘上一个 S 中的元素的意义下唯一，例如上图沿着紫线定义的 X_L' 算子，就可以看作是 X_L 乘上一个 S 中的元素，忽略符号。

基于缺陷的 Surface code

传统的 Surface Code 之外，我们可以关闭两个 $X(Z)$ -measure qubit 来为 Surface code 带来额外的自由度，并在该自由度下定义逻辑量子比特，这种方法称为缺陷，按图索骥就能定义该编码下的逻辑算子。

这么做的优点有：算子的定义不依赖平面边界，原来需要有两个 X -边界和两个 Z -边界；减少了逻辑算子实际作用在 physical qubit 上的算子数量；方便进行测量和初始化；可以通过 topological braid transformation 来定义 $CNOT$ 操作。

缺点是：纠错码的距离 d 减小了，仅为 4。4 个 data qubit 上的 Z 错误可能就无法通过测量分辨，然而这一点是受限于缺陷的大小，如果我们关闭更多的 measure qubit，制造更大的缺陷，就可以提高 d 。

移动缺陷

除此之外，缺陷还可以进行移动。如图所示的（a）到（b），先关闭一个 measure qubit，进行一轮测量，注意这里也需要测量由于断开连接而被孤立的那个 data qubit，新的逻辑算子按照 8 式 a, b 进行的变化。

接着从（b）到（c），打开上面的一个 measure qubit，再进行一轮测量，则逻辑算子按照 9 式进行变化。

新得到的逻辑算子可能在符号上与原来的逻辑算子不一样，这取决于测量的结果，记原来的逻辑量子态在新编码下的表示为 $|\psi'\rangle$ ，对 X_6 和（b）中 Z_L^c 的测量结果分别为 P_X 和 P_Z ，则变换的结果如 10 式所示。

这样的移动自然可以推广到多比特上，只需要像单比特一样关闭路径上的 measure qubit，进行一轮测量，然后再打开即可。

$CNOT$ 的构造

此部分最后阐述 $CNOT$ gate 的模拟。

一个平面上有着 X 和 Z 缺陷表示的两个 logical qubit。可以利用上面介绍的移动缺陷方法，分成两步移动，让一个 Z -缺陷绕着一个 X -缺陷移动一圈，这种操作等效于 $CNOT$ 。这可以由该操作在逻辑算子上的作用来验证，我们发现这个效果等同于一个 $CNOT$ gate 的共轭作用。

子系统编码

简介

在上面提到的稳定子编码框架下，可以扩展到子系统编码的形式，将量子系统作如下分解，希尔伯特 H 拆为 C 以及 C 的正交补， C 继续拆为 A 和 B 的张量积，而信息只需存储在 A 上，从而忽略 B 的错误， C 即为上述提到的稳定子描述。

该方法将信息存储和错误纠正实现了部分分解，拓宽了量子纠错的研究。

构造

至于子系统编码的具体构造，只需要恰当地选择 n 比特空间上的泡利算子集合，满足 11 式。选择 k 生成相应的稳定子群，再次选择 q 将该群进行分离，得到逻辑群和测量群，从而将 C 空间分离，通过 12 式至 14 式，我们可以看到这种具体的分离式可实现的。

纠错条件

与之前的经典量子纠错条件不同，子系统纠错编码由于只需要在其中一个子系统中进行编码，相应的修正条件为此表格所示。

Bacon-Shor 编码

我们以 Bacon-Shor 编码为例来理解量子纠错，先将二维晶格上泡利错误的情况映射到矩阵上，具体形式见 15 式与 16 式。

算子构造

然后在该二维晶格上构造相应算子，其数学表达式如图所示。

为方便理解，我们做了示意图：

图一代表每列有偶数个泡利 X 算符，每行有偶数个 Z 算符，在交叉的地方由于 $XZ = -iY$ ，则用 Y 来表示。

图二代表全为 X 算符的行个数为偶数个，全为 Z 算符的列个数为偶数个。

图三代表全为 X 算符的行个数为奇数个，全为 Z 算符的列个数为奇数个。

二维晶格子系统分离

由于 S 算子集合可交换的，可以将大希尔伯特空间进行分解，又由于 T 和 L 中的算子满足对易关系，又可继续分解。

我们只需要将信息编码于后者，与 T 算子的错误由于不在编码范围内，就不用考虑。我们观察 S 元素的组成形式，其相当于转化为两个一维重复编码的问题，并且在 S 的基础之上，可以构造相应的恢复操作，达到子系统纠错编码的目的。

三维晶格子系统

对于三维晶格的情况，与二维类似，在三维层面上进行修改，使用相同手段构造出相应的算子群。值得一提的是，该方法可以通过在一个立方体晶格上定义哈密顿量实现。

抗噪性能

在子系统编码提出之后，相关实验也有所推进。

在假定环境为随机噪声的情形下，对于最近两比特测量，能够得到对抗随机噪声的更低量子准确极限。相关实验借助了蒙特卡洛模拟方法验证了这一结果。

同时，在假定环境为非随机噪声的情形下，相关理论推导表明，如果泡利错误 X 和 Z 有偏向，当两者的差距达到 50 倍以上的时候，容错率能达到当今经典计算机的效果。