

# 1 Noisy random quantum circuits and its classical simulation

## 1.1 Introduction

在一些问题上，量子算法已经展示了相比于经典方法的优越性，比如量子采样。但是，出于实验结果是渐进的、实验在有限系统上进行、局部噪声的存在等原因，对实验的概率分布进行近似十分困难。因此在对实验结果的验证过程中，最直接的方式是将实验结果与经典模拟进行比较。

## 1.2 Random quantum circuits sampling

证明量子采样问题不能被经典计算机高效模拟。

## 1.3 Classical sampling of random quantum circuits

三种经典采样算法：

- Perfect sampling: 给定 bit 串  $x = (x_1, x_2, \dots, x_n)$  和理想概率  $p(x) = p(x_1, x_2, \dots, x_n)$ ，可以使用边缘分布对  $x$  中 bit 进行逐个采样。首先计算出第一个 bit 的边缘分布  $p(x_1) = \int p(x_1, x_2, \dots, x_n) dx_2 \dots dx_n$  并根据该分布对  $x_1$  进行采样。之后，将  $\bar{x}_1$  固定在采样结果中，并计算第二个 bit  $p(x_2) = \int p(\bar{x}_1, x_2, \dots, x_n) dx_3 \dots dx_n / \int p(\bar{x}_1, x_2, \dots, x_n) dx_2 \dots dx_n$ 。并依次继续。当所有 bit 都确定后，便成功采样  $x$ 。Perfect sampling 始终能给出理想分布的采样，但是在大多数情况下，它不像其它那些只需要  $p(x_1, \dots, x_n)$  的经典采样方法高效，因为计算边缘分布通常需要消耗更多时间。
- Markov chain Monte Carlo (MCMC): 在每个采样步骤中只需要单振幅。MCMC 通过构建一个马尔科夫链来实现采样。首先需要有一个非负函数  $q(y|x)$  满足  $\sum_y qy|x = 1$ 。 $q(y|x)$  的精确形式是随机的，但它会影响 MCMC 的收敛速度。在  $q(y|x)$  确定后，如果出于马尔科夫链的  $x^{[i]}$  态，根据以下算法跳转到  $x^{[i+1]}$  态：
  1. 根据分布  $q(y|x^{[i]})$  随机选择  $y$
  2. 在  $[0, 1]$  生成一个随机数  $r$

3. 如果  $r < \frac{p(y)q(x^{[i]}|y)}{p(x^{[i]})q(y|x^{[i]})}$ , 跳转到  $x^{[i+1]} = y$ ; 否则停留在  $x^{[i+1]} = x^{[i]}$

最终  $S = \{x^{[0]}, x^{[1]}, \dots\}$  是采样结果。经过足够的采样后, X 的分布在一定条件下会收敛到理想分布。尽管 MCMC 相比与 perfect sampling 计算量较低, 但当配置空间中高可能性区域是弱联通时, S 的分布可能收敛很慢。

- Rejection sampling: 只需要单振幅计算。在对分布  $p(x)$  进行采样前, 首先构建一个更适合经典模拟的分布  $q(x)$  满足  $p(x) \leq Mq(x)$ 。M 可以被定义为  $p(x)/q(x)$  的最大值。在分布  $q(x)$  下生成多个样本 S。对每个  $x \in S$ , 在  $[0, 1]$  平均生成一个随机数 r。如果  $r < p(x)/[Mq(x)]$ , 接受样本 x; 否则拒绝 x。易得被接受的样本集合满足分布 p(x)。平均而言, 需要计算每个接受样本的 M 个概率 p(x)。Rejection sampling 的主要缺点是, 对于高维空间, M 可以是非常大的。

MCMC 和 Rejection sampling 算法在很大程度上依赖于一个 bit 串的概率的计算, 或等效于一个位串的振幅。

对于振幅计算的基础经典算法:

- 薛定谔算法: 使用长度为  $2^n$  的向量存储量子态, 并在每次应用量子门后进行更新。对于一个有 m 个量子门 n 个 qubits 的线路  $U = U_1U_2 \dots U_m$ , 其中  $U_i$  为通用集中单 qubit 或双 qubit 门: 空间复杂度:  $O(2^n)$ ; 应用量子门并更新量子态的时间:  $O(2^n)$ (假定每个量子门只应用到  $O(1)$  个 qubit 上); 模拟整个量子线路 U 演化的总时间:  $O(m2^n)$ 。薛定谔算法计算完整的输出态, 需要指数级的存储空间。
- 费曼算法: 计算单振幅  $\langle x|U|0^n \rangle$ , 只需要多项式级存储空间。计算单振幅的时间复杂度为  $O(m4^m)$ , 空间复杂度为  $O(n+m)$ 。在  $m \gg n$  情况下, 费曼算法耗时远超薛定谔算法。
- 薛定谔-费曼算法: 在费曼算法基础上, 减少耗时, 同时保持空间复杂度在多项式级。其主要思想是分治。对于深度为 t 的量子线路, 分为两个深度为  $t/2$  的线路, 并再依次划分。最终时间复杂度为  $O(n(2t)^{n+1})$ , 空间复杂度为  $O(n \log t)$ 。该算法也可按 qubits 划分量子线路。

以上算法虽简单, 但都不高效。在加入张量网络 (TN) 后, 算法的计算复杂度得到了极大的降低。同时, TN 也能运用到其他随机量子线路的经典采样方法中。

一个  $\prod_i$  维的复张量  $T$  是在线性空间  $\otimes_i C^{D_i}$  中的一个元素，可以在一组基上展开：

$$T = \sum_{s_1, \dots, s_n} T_{s_1, \dots, s_n} e_{s_1}^{[1]} \otimes \dots \otimes e_{s_n}^{[n]}$$

其中  $\{e_{s_i}^{[i]} | 1 \leq s_i \leq D_i\}$  是  $C^{D_i}$  的一个基。张量  $T$  也可以被图形化表示，如图 1 所示。

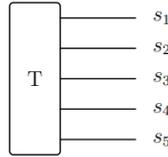


图 1: 图形化表示张量

一般来说，张量网络是张量的集合，以及它们是如何收缩在一起形成最终张量的。在图形表示法中，每个张量由一个节点表示，其指数由从该节点伸出的腿表示。如果要收缩张量的两个索引，就在图形表示中把相应的两条腿连接起来。例如，对于两个张量

$$C_{bef} = \sum_{ac} A_{abc} B_{aefc}$$

可以被图形化表示，如图 2 所示。

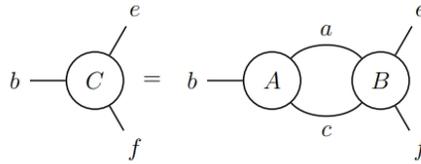


图 2: 张量图形化表示

在一个无噪声环境中，一个  $n$  qubits 系统的初态是一个  $n$  阶复张量，通常被表示为直积态

$$|\Psi\rangle = \otimes_i [\sum_{o_i} \Psi_{o_i}^{[i]} |o_i\rangle]$$

其中  $o$  表示输出的物理索引，可以图形化表示为图 3。

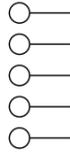


图 3: 物理索引图形表示

在无噪声系统中，一个  $m$  qubit 的酉门  $U$  是一个  $2m$  阶的酉张量

$$U = \sum_{o_1, \dots, o_m, i_1, \dots, i_m} U_{o_1, \dots, o_m, i_1, \dots, i_m} |o_1, \dots, o_m\rangle \langle i_1, \dots, i_m|$$

其中  $o$  是输出的物理索引， $i$  是输入的物理索引，可以图形化表示为图 4。



图 4: 量子门图形表示

由此可得，将量子线路的初态与各量子门相连，即可得整个量子线路的张量图形化表示，如图 5 所示。如果存在噪声，可以同样地构造一个张量网络，反映系统的密度矩阵下的量子线路的演变。我们只需要将物理索引加倍，用一个混合态代替初始态，用一个有噪声的量子通道代替每个门，如图 6 所示。

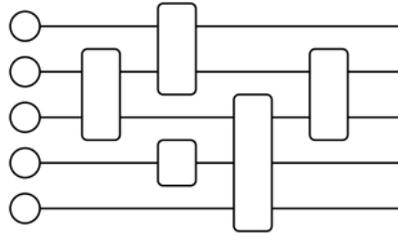


图 5: 无噪声情况下张量网络

每个含噪声的量子通道可以表示为 Kraus operators 的形式：

$$U_{i\bar{i}o\bar{o}} = \sum_s K_{ios} K_{i\bar{o}s}^*$$

其中  $i$  和  $\bar{i}$  分别是在状态空间和对偶空间中的输入物理索引，其中  $o$  和  $\bar{o}$

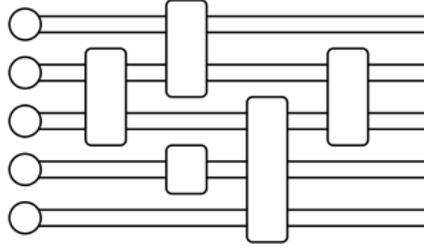


图 6: 有噪声情况下张量网络

分别是在状态空间和对偶空间中的输出物理索引， $s$  是噪声索引。可以表示为图 7。

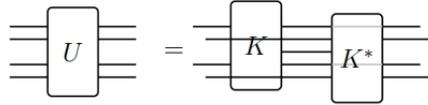


图 7: 含噪量子通道图表示

在使用张量网络对量子线路进行表示后，便可以将显示的仿真任务转化为张量网络算法。对于物理基础上的采样问题，可以使用蒙特卡洛采样或 Rejection sampling 进行单振幅计算；也可以使用 Perfect sampling 计算约化密度矩阵。对于计算观测量期望的任务，比如 VQA，也可以使用张量网络。

以上大部分的计算都需要进行张量网络的收缩。最朴素的精确压缩方法是用一棵压缩树直接压缩整个网络。收缩树是一种二叉树，其叶子与网络中的张量有 1-1 对应关系。对于收缩树，可以通过它的两个子节点的收缩来计算每个节点上的张量。根节点上的张量就是整个张量网络的收缩。图 8 是一个收缩树的例子。

已经证明有树状结构的张量网络的收缩具有多项式时间复杂度。然而，在一般情况下，根据网络的大小，被认为有指数级的时间复杂度。收缩的时间和空间复杂度受收缩树和张量网络结构的影响很大。目前应用最广泛的收缩树搜索算法是基于图划分的算法：

1. 将整个张量网络分成  $n \geq 2$  个具有固定不均匀性的部分，使得切割 (由连接不同部分的张量的键的维数的乘积定义) 最小

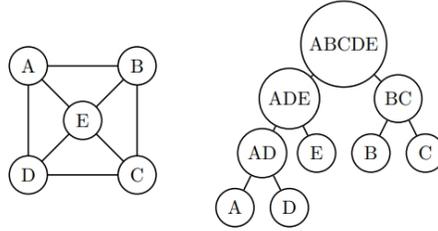


图 8: 张量网络收缩树

2. 对于每个部分，如果它的大小大于一个限制，递归地执行整个算法来找到它的收缩树。否则通过精确的方法获得该部分的收缩树
3. 找出收缩这  $n$  个部分所产生的  $n$  个张量的最佳方法。从而得到了整个张量网络的收缩树

常见的图划分算法包括 Kahypar、hMetis、PaToH、Zotan、HYPE。也可以使用结构化的张量网络（如 MPS 或 PEPS）来表示对应量子线路的张量网络。

量子器件的计算过程中存在噪声，导致输出分布的最终保真度是相对较低的。在这种情况下，经典的模拟方法不需要从有噪声的随机量子电路中准确地采样，甚至不需要直接采样，经典计算机产生的样本与有噪声的量子器件具有相同的质量就足够。因此可以通过将仿真的准确性与仿真成本进行权衡来加快经典仿真的速度。

一种以保真度  $F$  从输出分布中采样的方法是直接混合采样：

$$p_{noise}(x) = F p_{ideal}(x) + (1 - F) \frac{1}{2^n}$$

因此，如果需要生成  $M$  个样本，只需要从理想分布中生成  $M \cdot F$  个样本，剩余的直接从均匀分布中生成，从而实现  $1/F$  的加速比。

另一种方法 [?] 通过只对费曼路径的一小部分加权和来换取仿真成本，这也可以达到  $1/F$  加速比。

这种方式也可以运用到基于张量网络的经典模拟上。在 [?] 中，通过打破连接张量的  $m$  条边来对表示随机量子线路的张量网络进行简化。通过插入  $|0\rangle\langle 0| = \frac{1}{2}(I + \sigma_z)$  的方式来打破一条边，等同于提高  $1/2$  的错误率。因此打破  $m$  条边，最终态的保真度  $\sim 1/2^m$ 。