TensorQ 说明文档

一个基于张量网络的大规模量子线路模拟器

作者：赵先和

组织：中国科学技术大学

时间：6 月 1 日, 2023 年

版本：0.1.0

目录

第 1 章 TensorQ 介绍 **1**

1.1 安装与使用 1

第 2 章 基础原理 **2**

2.1 量子模拟指数墙问题 2

2.2 张量网络基础 2

2.2.1 张量的概念 2

2.2.2 张量的图像表示 2

2.2.3 张量的基本运算 3

2.3 TensorQ 核心技术 3

2.3.1 搜索缩并顺序 3

2.3.2 多振幅复用 5

2.3.3 张量网络切片分解 6

第 3 章 设计说明 **9**

第 4 章 测试效果 **10**

第 5 章 总结报告 **11**

第 1 章 TensorQ 介绍

TensorQ 是一个基于张量网络的量子线路模拟工具包，TensorQ 主要的功能是计算量子线路的振幅，可以选择单振幅、多振幅以及全振幅，可以计算精确振幅也可以计算近似振幅。TensorQ 兼容其他量子线路模拟工具包，例如 cirq 等，TensorQ 本身并不能生成线路，而是用其他工具包生成的线路，TensorQ 能够将量子线路转化成张量网络。另外 TensorQ 工具包提供低计算复杂度、低空间复杂度的张量网络缩并方案，以及相应的缩并运算得到最后的振幅。

TensorQ 在一定程度上解决了指数墙问题，可以在有限内存的计算资源上模拟比特数目更多的量子线路，例如，cirq 模拟 35 比特的量子线路需要 256GiB，而采用 TensorQ 可以在 24GiB 的设备上完成模拟。

我们的联系方式如下：

- GitHub 地址：https://github.com/Xenon-zhao/TensorQ
- 用户 QQ 群：待建设（建议加群）
- 邮件：648124022@qq.com

1.1 安装与使用

在 GitHub 仓库下载 TensorQ 压缩文件'TensorQ-main.zip'，解压压缩文件后。用 win+r 输入 cmd，进入命令行，进入'TensorQ-main' 所在的地址

```
cd <Yourpath>/TensorQ-main
```

其中，<Yourpath>根据你的文件地址修改。然后用下面的命令行进行安装

```
python setup.py install
```

第 2 章 基础原理

这节将介绍 TensorQ 用到的主要技术，将首先介绍量子模拟中遇到的指数墙问题，然后会介绍张量网络的基本概念，以及 TensorQ 如何解决指数墙问题。

2.1 量子模拟指数墙问题

量子计算系统的希尔伯特空间维度 M 随着量子比特的数量 N 指数增加，$M = 2^N$。在经典计算上模拟大规律量子线路，按照 Cirq、MindQuantum 以及 pyQPanda 等工具包的方法，需要存储 M 维的态矢量，这导致这种模拟方法对计算机内存的需求量通常随着比特个数 N 指数增加，经典计算机所能够模拟的量子线路比特数非常有限，这极大地限制了科研工作者对量子计算和量子算法的研究。

表 2.1: 指数墙问题

qubit 数量	空间维度	空间复杂度 (complex128)	存储设备
10	2^{10}	16 K bytes	
20	2^{20}	16 M bytes	
30	2^{30}	16 G bytes	笔记本
40	2^{40}	16 T bytes	计算集群
50	2^{50}	16 P bytes	超算
53	2^{53}	128 P bytes	所有超算的硬盘

2.2 张量网络基础

这一节将介绍张量网络的基础概念，包括张量是什么？张量的图像表示以及张量的基本运算。

2.2.1 张量的概念

张量的定义如下 [张量网络课程-冉世举-首都师范大学]:

- N 阶张量又称 **N 维数列**
- 张量阶数即为张量指标的个数
- 向量、矩阵也称为 1 阶张量、2 阶张量
- 每个指标可取的值的个数，被称为指标的维数
- 张量的“直观”定义：多个指标标记下的一堆数

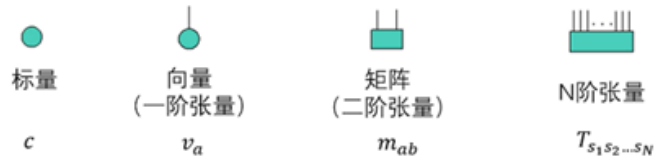
比如张量 $A[n_0][n_1]\cdots[n_{k-1}]$，其指标的个数为 k，称其为 k 阶张量，若指标可取的值的个数为 $n_i = N(0ik-1, N > 0)$，称指标的维数为 N。

一个 qubit 是一个向量，即一阶张量；一个单比特门是一个矩阵，即二阶张量；一个双比特门可以看成一个指标为 4 维的二阶张量也可以看成一个指标维度为 2 的四阶张量，因此量子线路中的所有元素都可以用张量表示。量子线路中的量子比特以及量子纠缠也是天然的可以使用张量网络来描述的 [Matrix product states and projected entangled pair states: Concepts, symmetries, theorems].

2.2.2 张量的图像表示

张量用连接着个腿的圆圈或方块表示，为张量的阶数，如下图所示。

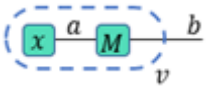
每个腿对应指标，比如一阶张量的腿对应指标 a，二阶张量的腿对应指标 a 和 b。



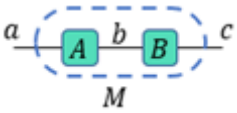
2.2.3 张量的基本运算

1. 向量内积：$c = \sum_a x_a y_a$;

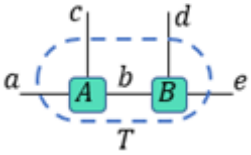
2. 向量乘矩阵：$v_b = \sum_a x_a M_{ab} = xM$;



3. 矩阵乘矩阵：$M_{ac} = \sum_b A_{ab} B_{bc} = AB$;



4. 张量缩并：$T_{acde} = \sum_b A_{acb} B_{bde}$;



2.3 TensorQ 核心技术

这一节将介绍 TensorQ 中的核心技术，包括搜索缩并顺序、多振幅复用以及张量网络切片分解。

2.3.1 搜索缩并顺序

上一节已经说明了，量子线路都可以转化成张量网络，而通过张量计算振幅就是要把网络缩并，而将一个张量网络缩并通常有很多种不同的缩并顺序，往往不同的缩并顺序对应着不同的复杂度 [Hyper-optimized tensor network contraction]，接下来我们通过一个简单的网络缩并来说明这件事。

我们考虑一个如图2.1(a) 所示的张量网络，一共有 5 个张量，用 5 个黑色节点表示。每个节所连接的边表示该张量的指标，两个节点之间相连的边表示两个张量的共有指标，也就是缩并指标。

对于图2.1(a) 所示的张量网络，我们给出两种缩并顺序，把缩并顺序画出树状图的结构，分别如图2.1(b) 和2.1(c) 所示，这种图称为缩并树。缩并树中每一个节点表示一个张量，并且用一组字母表示该张量的指标；每一条边表示张量的缩并路径。五个节点按照缩并树从下到上的方向，两两相遇合并为一个新的节点，代表张量两两缩并产生中间张量，并最终缩并为一个标量。

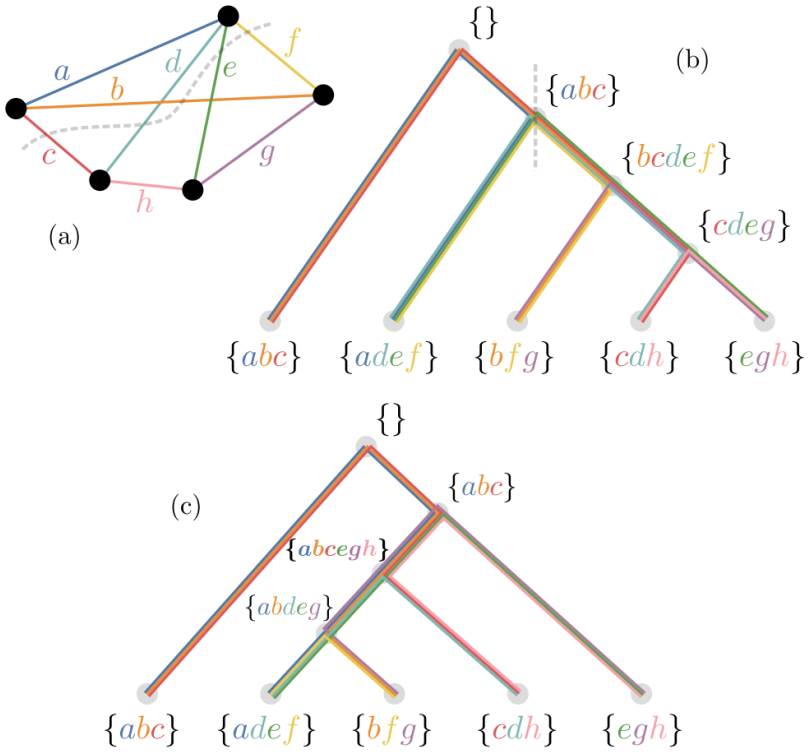


图 2.1: 对于 (a) 所示的图，两个可能的收缩树 (b) 和 (c)，展示了中间张量和缩并。树中的每条边都有一个相对应的张量与子图。张量的大小与指标的数量呈指数关系，沿着边缩并的指标，用独特的颜色表示。树中的每个顶点代表两个张量的成对收缩，以及子图的二分割 (虚线显示了一个例子)。这种两两收缩的计算复杂度通过顶点的指标数量指数怎加。假设每个索引的大小相同，树 (c) 因此具有比树 (b) 更高的最大缩并宽度 (粗体) 和总收缩成本 [Hyper-optimized tensor network contraction]。

接下来，我们计算两种缩并顺序的复杂度。复杂度分为空间复杂度与时间复杂度，空间复杂度就是将整个张量网络缩并所需要的存储空间，这由整个缩并过程中最大的中间张量的大小决定；时间复杂度就是整个张量网络缩并过程中的浮点运行次数，浮点运行包括加减乘除运算，由于乘法运算是缩并的主要运算开销，一般只计算乘法运算次数。我们假设图2.1(a) 中的每个指标维度相同都为 n，因此缩并顺序 (b) 的最大张量具有 5 个指标 bcdef，空间复杂度为 n^5；缩并顺序 (c) 的最大张量有 6 个指标 abcegh，空间复杂度为 n^6。在时间复杂度计算上，每次缩并的乘法运算的次数与独立指标个数成指数关系，缩并顺序 (b) 一个有 4 次缩并，乘法次数依次为 n^5, n^6, n^6, n^3；缩并顺序 (c) 也有 4 次缩并，乘法次数依次为 n^6, n^7, n^6, n^3。可见，缩并顺序 (b) 的空间复杂度和时间复杂度都更小。

对应一个任意的张量网络，寻找复杂度最低的缩并方案至少是一个#P-hard的问题 [[26] S. J. Denny, J. D. Biamonte, D. Jaksch, and S. R. Clark, J. Phys. A Math. Theor. 45, 015309 (2012). [27] L. G. Valiant, Theor. Comput. Sci. 8, 189 (1979).]。TensorQ 采用 arTensor 工具包进行缩并顺序搜索，arTensor 中采用的是随机算法来进行搜索，通常搜索时间越长得到的缩并顺序复杂度越低，arTensor 可以给出缩并顺序相应的复杂度，根据计算资源的能力得到满足需求的缩并顺序即可。

cirq、MindQuantum 等现有工具包模拟量子线路的方法称为薛定谔算法，这种方法存储并计算完整的态矢量 [Simulating the Sycamore quantum supremacy circuits]。薛定谔算法的缩并顺序如图2.2所示，这种算法可以看作是一种特殊的缩并顺序，对于单振幅或者少振幅的情况，这种顺序显然不是张量网络中复杂度较低的顺序。图2.1中的例子可以启发我们发现，通常先将指标个数较少，共有指标较多的张量先缩并能够产生指标个数较少的中间张量，这样更有利于减少缩并的复杂度。将量子线路转化成张量网络后，搜索低复杂度的缩并顺序，这是 TensorQ 优化大规模量子线路模拟的第一步。

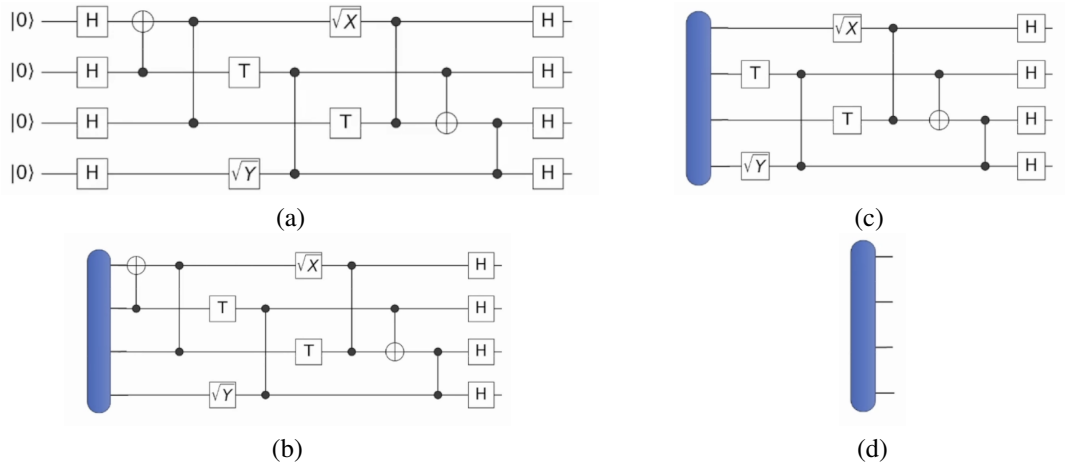


图 2.2: 薛定谔算法将量子线路按照 (a), (b), (c), (d) 的顺序进行缩并。

2.3.2 多振幅复用

对量子线路的模拟通常分为三种，单振幅模拟，多振幅模拟，全振幅模拟，分别表示模拟结果输出末态单个，多个，全部振幅。对于单振幅模拟，张量网络方法比薛定谔算法复杂度低很多，因为线路两端是相互独立的 2N 个二维向量，N 是比特数量，都是一些非常小的一阶张量，一阶张量与其他张量缩并后只会将另外一个张量的阶数也降低，因此张量网络方法很容易找到复杂度比薛定谔算法低的缩并方案。对于全振幅模拟，张量网络方法相比薛定谔算法就没有优势了，因为 N 个比特总共有 2^N 个振幅，当 N 很大时这么多振幅是能难存储的，全振幅模拟的指数墙困难没有办法解决。

不过，在大规模量子线路的研究中，通常关注的都只是一部分振幅，即对于 2^N 个振幅，真正需要计算的只有 m 个振幅，通常 $m << 2^N$，这种情况张量网络方法是可以模拟的。在多振幅的模拟中，又有两种情况，关联多振幅模拟与无关联多振幅模拟，关联多振幅是指，多个振幅对于的比特串的构形是相互关联的，例如 5 个 qubit 中，固定第一、二个 qubit 为 0，那么一共有 8 种比特串，分别是’00000’, ’00001’, ’00010’, ’00011’, ’00100’, ’00101’, ’00110’, ’00111’。这种情况下，第一、二个比特的末态仍然是二维向量，只有一个指标，而另外三个比特的模拟由于会出现两种情况，因此阶数增加，变成二阶张量。而无关联多振幅的情况更加复杂，它是指多个振幅的比特串构形是任意的，相互之间没有关联，比特任意写成 8 个比特，可以是’01000’, ’10001’, ’11010’, ’00011’, ’00100’, ’11101’, ’10110’, ’01111’，可以发现这 8 个比特串中，每个 qubit 都既出现了’0’ 也出现了’1’。

全振幅、单振幅、关联多振幅、无关联多振幅的模拟，通过下面的图2.3说明了。

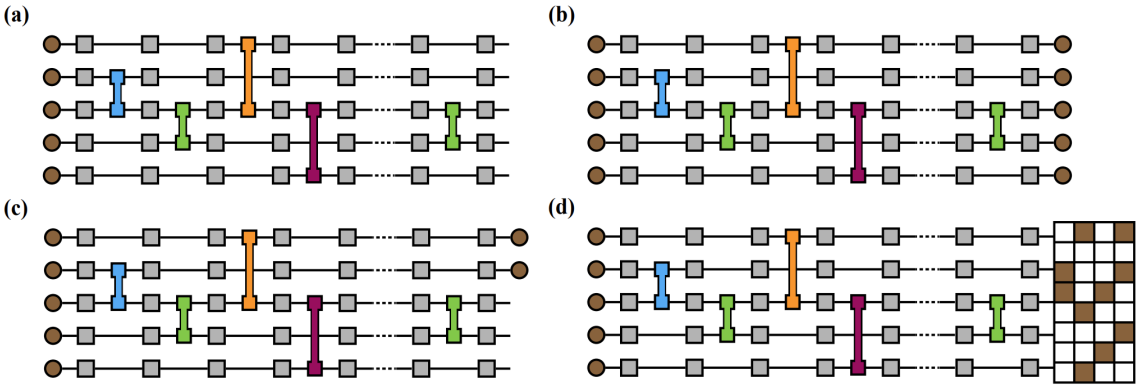


图 2.3: 四种不同的量子线路模拟类型，分别是 (a) 全振幅模拟，(b) 单振幅模拟，(c) 关联多振幅模拟，(d) 无关联多振幅模拟 [Solving the Sampling Problem of the Sycamore Quantum Circuits]

无关联多振幅的情况比关联多振幅的情况更加复杂，而且也更加重要，因为关联多振幅只计算了希尔伯特空间中，部分相互关联的振幅，而无关联多振幅才是希尔伯特空间中随机的振幅，更能代表希尔伯特全空间的

性质。但是，无光联振幅中，每个比特都出现了两种情况，是不是要把所有比特的末态都变成二阶张量呢？这是不可行的，如果所有末态比特都变成二阶张量，将会使这个模拟与全振幅模拟一模一样，所以关联多振幅的模拟变成了一个非常重要且有具有挑战的问题。

另外一个缩并顺序搜索的软件包 cotengra[https://github.com/jcmgray/cotengra] 也提供了张量网络的缩并顺序搜索功能，不过它只针对单振幅和全振幅的缩并，并没有优化无关联多振幅的缩并。TensorQ 的缩并模块采用 arTensor 软件包，针对无关联多振幅模拟的情况，采用了多振幅复用的算法，又称为稀疏态模拟算法（sparse-state simulation）[Solving the Sampling Problem of the Sycamore Quantum Circuits]，来减低复杂度。多振幅复用算法的原理是，根据需要计算振幅的目标比特串，动态地调整末态比特的指标维度。

如图2.4所示，考虑在 3 比特的线路中计算'111', '010', '000' 三个比特串的振幅，在图2.4(a) 中，张量网络初始状态下，三个末态比特都是二阶张量，但是当二、三号比特缩并时，二、三比特共同末态的张量不会计算和保留全部的维度，而是根据二、三号比特的目标比特串有选择地计算和保留部分分量，在这个例子中，二、三比特的目标比特串只需要'11', '10', '00' 三个分量的值，而不需要'01' 分量的值，因此将这个缩并过程中的复杂度降低为多振幅复用前的 3/4。类似的，当下一步缩并，三个比特全部缩并在一起时，也只选择了'111', '010', '000' 分量进行计算与存储，这将复杂度降低为了多振幅复用前的 3/8。

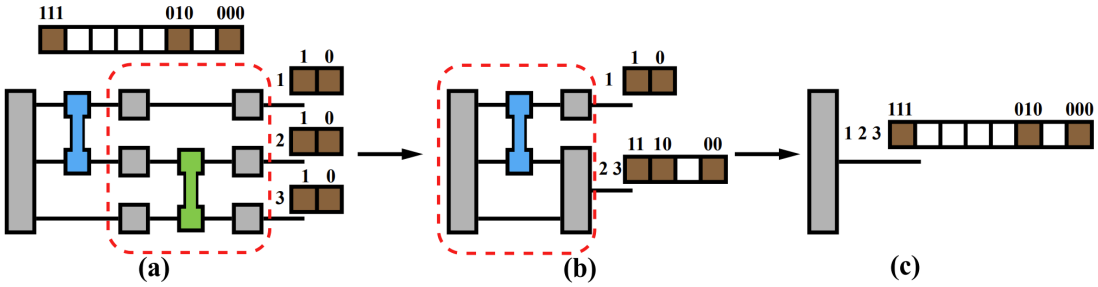


图 2.4: 无关联多振幅复用算法示意图。[Solving the Sampling Problem of the Sycamore Quantum Circuits]

因此在一般的无关联多振幅张量网络缩并中，多振幅复用算法将整体的复杂度大大降低，特别是，避免了像全振幅模拟一样需要保存所有的态矢量分量，解决了多振幅模拟中的指数墙问题。

2.3.3 张量网络切片分解

在大规模量子线路模拟的过程中，当比特数目很多，需要计算的振幅数量很多的时候，通过优化缩并顺序以及多振幅复用两种优化算法还是不能很好地解决空间复杂度过大的问题，虽然优化后的空间复杂度会远低于薛定谔算法的复杂度，但往往中间张量仍然需要大量的存储空间。TensorQ 提供了另外一种算法，手动限制张量的空间复杂度，这种算法是张量网络切片分解（slicing）[Solving the Sampling Problem of the Sycamore Quantum Circuits]。

切片分解算法的需要用到如图2.5所示的张量网络性质，在一个包含 4 个张量的网络中，我们可以将网络中的指标 k 固定，指标 k 每一种不同的取值都会形成一个不同的子网络，等式左边表示原始张量网络的缩并结果，等式右边表示不同 k 值的子张量网络缩并后再将结果求和。我们可以证明图2.5中的等式是成立的，在等式左边的缩并过程中，选择 k 指标最后进行缩并，先将 i, j, m, l 指标缩并，再将 k 指标缩并，也就是对 k 指标求和；这和等式右边表示的先缩并子网络再求和是完全一样的。

图2.5的等式说明，我们可以保证得到相同结果的条件下，降低张量缩并的空间复杂度。因为，在图2.5等式左边的张量缩并过程中，最大张量是三阶张量（默认每个指标的维度相同，因为在量子线路的模拟过程中，每个指标的维度都是 2，因此空间复杂度只需要考虑张量阶数即可），而等式右边的缩并过程中，最大张量是二阶张量，其中需要注意指标 k 是固定的，不会贡献张量的阶数。推广到更一般的张量网络缩并过程，TensorQ 中通过设置最大张量阶数，限制了最大空间复杂度，在搜索缩并顺序的过程中会自动地尝试将多个指标进行切片，保证缩并过程不破坏最大空间复杂度的限制。将 m 个维度是 2 的指标切片，意味着切片指标有 2^m 种取值，因此

原始张量网络会生成 2^m 个子网络，每个子网络的缩并都不会超过空间复杂度上限，将子网络的缩并结果求和就能得到原始张量网络的缩并结果。

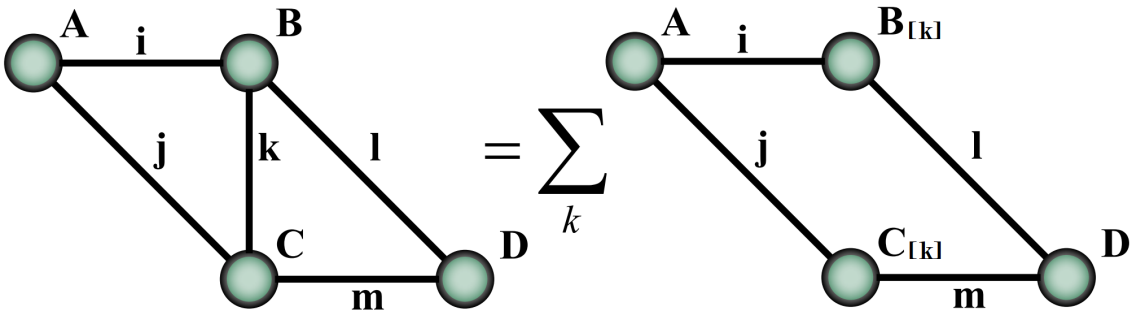


图 2.5: 切片算法示意图。将指标 k 进行切片，也就是将等式左边的原始张量网络的缩并变成了等式右边多个对应不同 k 值的子任务缩并的求和 [Solving the Sampling Problem of the Sycamore Quantum Circuits]

另外需要强调的是，切片算法虽然能降低缩并的空间复杂度，但并不能降低整体计算过程的时间复杂度，往往还会导致时间复杂度更高，这称之为切片额外开销（slicing overhead）[Lifetime-based Method for Quantum Simulation on a New Sunway Supercomputer]。我们仍然采用图2.5中的例子进行说明，我们假设所有的指标维度是 2，可以通过枚举证明原始张量网络复杂度最低的缩并顺序是：$[A, B], [AB, C], [ABC, D]$，时间复杂度为：$2^4+2^4+2^2$；而如果我们采用图2.6所示的，对 i 指标切片，采用复杂度最低的缩并顺序：$[A, C], [AC, B], [ACB, D]$，总空间复杂度为：$2*(2^3+2^3+2^2)$；可以发现总的乘法运算增加了 4 次。对于更大的张量网络，切片指标更多的情况，时间复杂度增加得更多。因此切片算法只是空间复杂度超过设备存储范围的权宜之策，通常切片数量越少，时间复杂度越低，建议在 TensorQ 中将空间复杂度限制为设备内存的最大值。

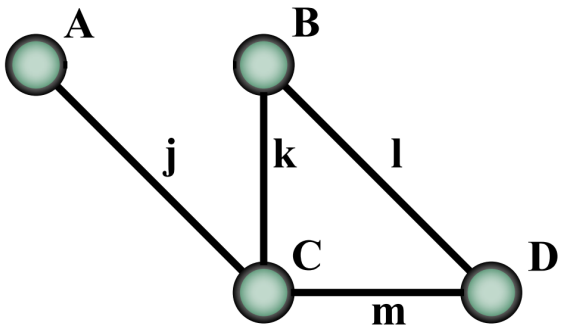


图 2.6: 另外一种切片方案示意图。对指标 i 进行切片，将使整体的时间复杂度更高。

值得一提的是，切片算法还要另外一个作用，在量子纠缠较高的线路中通常纠缠熵较大，纠缠谱是均匀分布的 [Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)]，类似图2.7中的 $|\phi_{random}\rangle$ 态。对于这种态，每一个子网络贡献的结果在统计意义上的等价的，也就是每一个子网络会贡献同样多的保真度 [Solving the Sampling Problem of the Sycamore Quantum Circuits]。如果一个有 K 个子网络，将所有子网络的结果相加可以得到原始网络的精确结果，如果只计算并求和 t($t < K$) 个子网络的结果，将得到原始网络的近似结果，保真度为 $f = t/K$，其中 t 个子网络是任意选择的。这个性质的作用是，当我们不需要模拟量子线路的精确振幅时，可以用这种方法模拟近似振幅，并且时间复杂度相对精确模拟而言降低了 $1/f$ 倍。

因此 TensorQ 通过张量网络切片算法，提供了一个限制空间复杂度方案，虽然这种算法会有用时间复杂度提高的代价，但是它能够模拟切片前会超过设备内存能力的量子线路，完成了之前无法完成的任务。

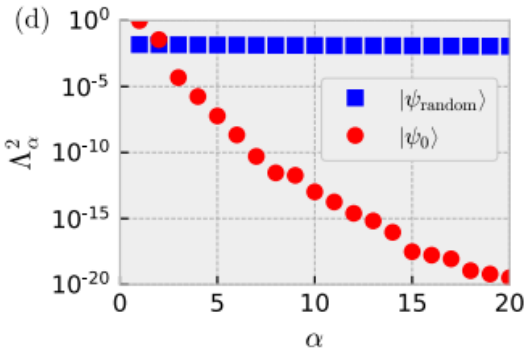


图 2.7: 横场 Ising 模型的两种态的纠缠谱。α 是施密特分解正交项的序号，Λ_α^2 是纠缠谱第 α 项的值，$|\phi_{random}\rangle$ 是自旋随机构形态，$|\psi_0\rangle$ 是基态 [Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy)]。

第 3 章 设计说明

第 4 章 测试效果

第 5 章 总结报告