arXiv:2602.02698v1 [quant-ph] 2 Feb 2026

编译量子正则语言态

Armando Bellante, [1, 2, *] Reinis Irmejs, [1, 2, †] Marta Florido-Llinàs, [1,2] María Cea Fernández, [1,2] Marianna Crupi, [1,2] Matthew Kiser, [3,4] 和 J. Ignacio Cirac[1,2]

[1] *Max-Planck-Institut für Quantenoptik, Hans-Kopfermann-Straße 1, D-85748 Garching, Germany*
[2] *Munich Center for Quantum Science and Technology (MCQST), Schellingstraße 4, D-80799 Munich, Germany*
[3] *TUM School of Natural Sciences, Technical University of Munich, Boltzmannstr. 10, 85748 Garching, Germany*
[4] *IQM Quantum Computers, Georg-Brauchle-Ring 23-25, 80992 Munich, Germany*

量子计算机的量子态制备编译器通常处于两个极端：一类是将目标视为不透明振幅向量的通用程序，另一类则是针对少数已知量子态族的定制构造。我们探究编译器能否改为接受简单且能体现结构的规格说明，同时提供可预测的资源保证。为此，我们设计并实现了一种用于正则语言态（RLS）的量子态制备编译器：它生成由正则描述所接受的比特串构成的均匀叠加态及其补态。用户可以通过以下方式描述目标态：(i) 比特串的有限集合；(ii) 正则表达式；或 (iii) 确定性有限自动机（DFA），并可选择设置补标志。编译器将输入转换为 DFA，对其进行最小化，再将其映射为最优矩阵乘积态（MPS），从而获得一种能够显露并压缩隐藏结构的中间表示（IR）。高效的 DFA 表示和最小化将耗费较大的线性代数计算转化为较简单的自动机操作。正则语言前端与这种 IR 相结合，不仅能为 RLS 提供简洁的规格说明，也能为其补态提供简洁的规格说明；否则，这些补态可能需要规模呈指数增长的量子态描述。据我们所知，现有编译器尚不支持以相同的渐近资源和编译时间制备 RLS 或其补态。我们介绍两种硬件感知后端：SeqRLSP 通过顺序生成，为线性最近邻架构生成线性深度、无辅助量子比特电路；TreeRLSP 则借助树张量网络，在全连接架构上实现对数深度。在理论方面，我们证明了电路深度和门数界，其随系统规模和目标态的最大施密特秩变化；此外，我们还给出编译时间界，展现初始 DFA 表示的优势。我们实现了完整流水线，并在 Dicke态、W 态、随机均匀叠加态和补态上进行评估，将结果与通用、稀疏态及专用基线进行比较。

I. 引言

量子计算机是一种可编程的物理系统，它不仅遵循量子力学定律，还在器件尺度上体现这些定律，使叠加和纠缠变得可控且可利用。这种能力拓展了信息处理方式，有望带来超越传统经典计算机的优势。尤其是在若干问题上，已证实量子计算相较于已知最佳经典算法具有指数级优势，应用范围涵盖通过整数分解和离散对数问题进行密码分析 [1], 、代数数论中的问题 [2], 以及多体物理模拟 [3, 4]。在适当的输入模型下，优化 [5–10], 、统计学 [11, 12], 和机器学习 [13–17]中也会出现多项式乃至超多项式优势。

过去几年间，量子计算机在规模和保真度方面稳步提升 [18, 19], ，因此，越来越需要自动化方法，将用户的高层意图高效映射为低层量子电路，同时优化编译时间和量子资源（电路深度、门数、量子比特数）。在实践中，为这些设备编程类似于现场可编程门阵列（FPGA）的合成工作流。在量子电路模型中，算法是一个紧密的经典—量子循环：经典主机编译量子电路，量子处理器将量子比特初始化为全零态，施加规定的量子门并测量终态，返回采样得到的比特串。主机汇总结果、解读信息，并可根据需要调整下一轮电路，直至任务完成。

许多算法最自然的表述方式，是假设可以查询一个电路，该电路能从全零态制备出目标态：线性系统求解器假设能够制备右端向量 [5]；振幅放大

* armando.bellante@mpq.mpg.de
† reinis.irmejs@mpq.mpg.de

和估计假设存在一个包含并标记“良好”子空间的预言机 [20]；数据驱动方法假设能够访问一个加载电路，该电路会创建一种对数据集进行编码的叠加 [21–23]。

由于量子态制备是许多量子算法的重要组成部分，正如经典编译器会在每次调用周围插入函数序言一样，高效编译这一子程序至关重要。具体而言，宿主需要根据用户提供的描述，合成一个能够制备目标态的量子电路。一般来说，对 N 个量子比特上的量子态进行精确的经典描述，最多需要指定 2^N 个复振幅，这是由量子叠加的本质决定的。随着量子比特数量增加，这会形成一道根本性的障碍。

实践中，目标规格说明有两种方式传递给编译器。第一种方式是用户提供显式支撑集——振幅列表或字典——这仅适用于非零振幅数量为 $\ll 2^N$ 的稀疏量子态，或适用于系统规模较小的情况。在这种情况下，通用例程会将输入视为任意向量，通常无法利用稀疏度以外的规律性，因而生成效率低下的电路。第二种方式是用户识别出该量子态属于一个经过充分研究的态族（例如，GHZ [24], W [25], Dicke [26]），并调用专用例程来合成高效且针对性设计的电路。

This 通用但不考虑结构的方法与量身定制的方案之间的差距，引出了我们的问题： n:

我们能否提出一种简单、紧凑且感知结构的规格说明，无需指定某个特定的量子态族，同时又能编译成资源界可预测且高效的量子电路？

我们的答案介于两者之间，聚焦于一类范围广泛但定义严谨的目标——正则语言态（RLS）：由正则描述识别的长度为 N 的比特串所构成的均匀叠加，以及与之对称的补态——即所有长度为 N 的比特串中，除所描述的比特串之外其余比特串的均匀叠加。在本研究中，我们将讨论范围限定为均匀叠加，而非任意复系数。这样的选择使这一类目标更易于分析，同时仍涵盖许多有代表性的族（GHZ态、W态、Dicke态、固定奇偶性/权重，以及简单的前缀/后缀）[24–26]。

先前研究将正则语言（RL）与矩阵乘积态（MPS）联系起来 [27, 28], ，后者是张量网络（TN）的一种特殊类型 [29], ；另有研究展示了如何通过顺序生成，将给定的MPS实现为量子电路 [30]、树形/并行结构 [31, 32], 或测量与反馈方案 [33]。我们弥合了正则描述与量子态合成之间的差距，将这些彼此独立的事实整合为一套端到端的编译器流水线（图 1）。用户可以提供长度为 N 的字符串的正则描述，作为

— 比特串的集合（或列表，或字典），— 正则表达式（正则表达式），—
接受该语言的确定性有限自动机（确定有限自动机）。

用户也可以选择请求任一描述的补态。如图 1所示，编译器将输入规范化为最小确定有限自动机，再将该确定有限自动机转换为矩阵乘积态，由矩阵乘积态构造量子操作，并将其转换为量子电路。从概念上讲，前端将用户规格映射为 DFA/MPS 中间表示（IR），后端则根据硬件约束选择从矩阵乘积态到电路的实现方式。由此得到一条模块化流水线，它利用形式语言理论、张量网络和矩阵乘积态综合的成果，自动发现结构并优化电路综合。在前端，正则描述提供了一种通用且结构感知的规格说明，而中间表示与后端则确保编译给出可预测且高效的资源界。

每种指定目标态的方式都有其自身优势。首先，用户可以以集合、列表或字典的形式提供非结构化输入，而流水线仍不会将其视为不透明向量：前端会将其转换为确定有限自动机，并放入与其他输入相同的中间表示中，从而使编译器能够揭示其结构。其次，当用户了解所需结构时，可以直接用正则表达式或确定有限自动机加以表达，而无需点明某一特定量子态族：例如，长度为 N 的所有独热字符串上的均匀叠加可以写成一个简单的正则表达式模式，而无需知道“W 态”这一术语。第三，尽管本文聚焦于正则语言，确定有限自动机输入的用途也超出了严格正则族：例如，Motzkin 态源于括号配对字符串的（非正则）戴克语言 [34, 35], ，但对于任意固定长度 N，它们仍可由确定有限自动机简洁地表示。最后，补集是一种原生构造：用户不必列出所有允许的字符串，而可以描述一个较小的禁用语言，并请求其余所有比特串上的均匀叠加。这样，即使量子支撑集呈指数级规模，经典描述依然保持紧凑；而且正如我们将要表明的，这并不会显著增加该量子态的结构复杂度。

中间表示上的优化遍首先对确定有限自动机进行最小化，然后优化矩阵乘积态张量的内部维度，从而揭示目标态的结构。在处理矩阵乘积态之前先对确定有限自动机进行最小化，可以将工作从针对大型张量的高开销线性代数运算转向更简单的自动机操作。这使得

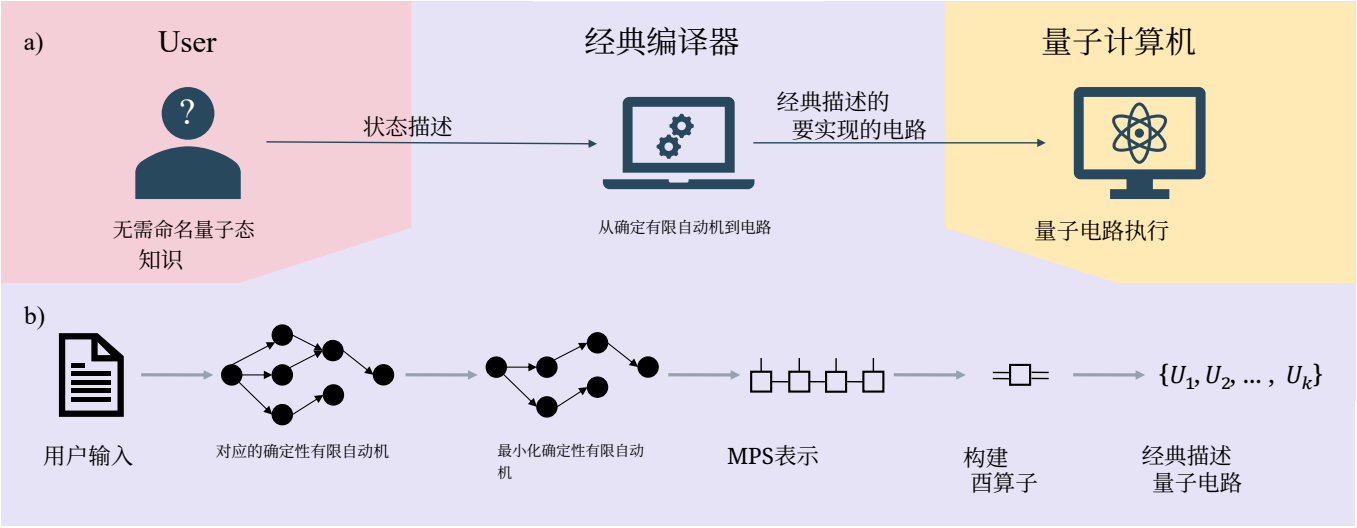


图 1：(a) 一般设定的示意图。用户可能并不了解量子态的名称或结构，但会向经典编译器提供所需量子态的描述，例如字符串集合、正则表达式或确定有限自动机，以及（可选的）补标志。随后，编译器使用与目标硬件兼容的门集生成量子电路，由目标硬件执行生成的指令。(b) 使用确定有限自动机和矩阵乘积态的编译流水线概览。用户输入（正则表达式、确定有限自动机或字符串集合）会被转换为确定有限自动机，再经过最小化并转换为矩阵乘积态。矩阵乘积态会被分解为酉算子，并编译为门指令集，以便在量子计算机上执行。

编译更加高效。更广泛地说，确定有限自动机与矩阵乘积态的对应关系让我们能够从用户层面的规格说明系统地推导出紧凑的矩阵乘积态表示，而不必假设专家已经提供了优质的矩阵乘积态，从而弥合了与以往基于矩阵乘积态的合成工作的差距。

根据中间表示，后端会合成硬件感知电路。本文介绍两种实现：适用于线性最近邻设备的顺序后端 SeqRLSP，其深度为线性且无需辅助量子比特 [30],；以及适用于全连接设备的树形后端 TreeRLSP，它通过并行化矩阵乘积态实现，将深度降至对数级 [31, 32]。

a. 贡献 总而言之，我们的贡献如下：

1. 规格说明。我们为量子态制备引入了一种正则描述（正则表达式/确定有限自动机/有限集合），并支持补集选项。这使用户无需命名族，即可简洁地描述规模庞大且具有结构的支撑集，并且能够自然扩展到非正则族的定长切片。 2. 编译。我们弥合了正则语言、矩阵乘积态和量子电路相关文献之间的鸿沟，提供了一个端到端编译器，其中包括 DFA/MPS 中间表示以及两个硬件感知后端：SeqRLSP 和 TreeRLSP。 3. 资源使用与编译时间。我们根据 N 和目标态的最大施密特秩 χ 给出了明确的量子电路资源界（定义 5、定理 3 和定理 4），并提供了最坏情况编译时间界（定理 1、推论 1）。 4. 补集理论。我们证明，补集与其基础描述具有相同的渐近量子代价，因为其最大施密特秩 χ 最多增加 1（定理 2）；这将量子态的纠缠结构与其补集的纠缠结构联系起来。 5. 评估。我们实现了完整技术栈，并通过与通用、稀疏态和 Dicke/W 专用基线进行基准测试来支持我们的论断，测试对象涵盖 Dicke/W、随机均匀叠加态和补集，并报告量子电路深度、量子门数量、辅助量子比特数量和编译时间。

b. 文档结构 第二节 介绍相关背景，回顾量子计算和电路设计的基本要素、相关编译器、RLS 和矩阵乘积态。我们还回顾了现有的量子态编译器。 第三节介绍我们的方法，并概述编译器的各个步骤。在第四节中，我们讨论编译时

复杂度，并利用我们的编译器界定制备 RLS 所需的量子资源，进一步将 RLS 的纠缠结构与其补态联系起来。最后，第五节 介绍数值实验，将我们的编译器与现有方法进行基准测试；第六节 总结全文，并指出未来可能的研究方向。

II. 背景

本节回顾本文所需的基础知识，包括量子计算、RLS、有限状态自动机和矩阵乘积态，以及相关的编译器和量子态制备技术。更深入的介绍可参阅：量子计算方面的尼尔森和庄 [36]，矩阵乘积态和张量网络方面的 Cirac等人 [37] 和 Or´us [29]，以及有限状态自动机和正则语言方面的霍普克罗夫特等人 [38]。

A. 量子计算

a. 量子态 单个$qudit$ 的物理维数为 d，其标准（计算）基向量为$\{|0\rangle,\ldots,|d-1\rangle\}$。这些标签构成一个字母表 Σ，其中包含 $|\Sigma| = d$（对于量子比特，d=2和 $\Sigma = \{0,1\}$）。由 N 个 qudit 组成的寄存器处于希尔伯特空间 $(\mathbb{C}^d)^{\otimes N}$ 中，其基向量是各个 qudit 状态的张量积$|x_1\rangle\otimes\cdots\otimes|x_N\rangle$，由字符串 $x = x_1\cdots x_N \in \Sigma^N$索引；简写为 $|x\rangle$。纯量子态是一个单位向量 $|\psi\rangle = \sum_{x\in\Sigma^N}\alpha_x|x\rangle$，满足 $\sum_x|\alpha_x|^2 = 1$，其中复振幅为 α_x，而全局相位在物理上无关紧要。因此，一个一般的 N-qudit 状态由 d^N 个振幅指定。对于有限集合 $S \subseteq \Sigma^N$，其均匀叠加为$|u_S\rangle = \frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle$；即，$\alpha_x$ 在 S 上为常数，在其外为 0。特别地，对于量子比特，通常将状态写作 $|+\rangle^{\otimes N} = \frac{1}{\sqrt{2^N}}\sum_{x\in\{0,1\}^N}|x\rangle$，它是对所有 N 位比特串的均匀叠加。

b. 幺正演化与量子电路 封闭量子系统在幺正算子作用下演化，即在 $(\mathbb{C}^d)^{\otimes N}$上保持内积的可逆线性映射。对于量子态制备，作用于 N 个量子位的幺正算子 U会将全零态映射到目标量子态 $U|0^N\rangle = |\psi\rangle$。电路模型利用了这样一个事实：任意 N 量子比特的幺正算子都可以精确分解为作用于 1 个或 2 个量子位的更小的幺正算子，这些算子称为量子门。由于硬件只支持固定且有限的门集，而该门集必须具有通用性，因此在合成过程中需要将这些任意量子门高效地近似到指定精度（例如，通过索洛维-基塔耶夫 [39, 40]）。

c. 量子硬件 就本文而言，设备由以下两部分描述：(i) 经校准的原生、有限且通用的门集；(ii) 表示哪些量子位可以直接交互的连接图。两种常见的连接模型是线性最近邻（LNN），其中交互仅发生在相邻格点之间（因此远距离量子门需要进行 SWAP 路由）；以及全对全，其中任意一对量子位都可以直接交互。这些限制会影响电路成本：在 LNN 模型中，路由会增加门数和深度，但通常可以并行执行量子门；而全对全模型有利于降低深度，但可能不支持并行门操作。编译器通过将任意量子门分解为原生门，并采用连接感知路由，将抽象电路转译到硬件上。

d. 资源数量成本模型与经典布尔电路的相同：衡量指标包括关键电路深度、基本门总数和*qudit* 总数。对于量子态制备，N 是 qudit 数量的一个显然下界；因此，我们统计辅助 qudit 的数量（也称为辅助量子比特）。

B. 量子态制备与编译器

下面我们回顾一些具有代表性的方法，这些方法用于量子态制备，涵盖从通用编译器到针对特定结构化量子态族的算法。我们重点关注量子比特方法，设定$d = 2$。

a. 通用编译器 通用量子编程框架（例如$Qiskit$ [41, 42] 和$Qualtran$ [43, 44], ）包含量子态制备模块，可将由振幅向量定义的任意目标态编译为可执行的量子电路。这些方法具有通用性，并能保证实现精确或近似的量子态制备，但目前的实现要么难以识别输入描述中的结构并加以利用，要么在设计时就未考虑这一点。这一不足要求用户设计自定义程序，以获得更优的量子电路。这一局限是实际层面的，而非根本性的；原则上，未来的编译器可以识别此类结构，自动降低资源成本。目前，$Qiskit$ 的复杂度随完整的希尔伯特空间维数增长，因此该方法主要适用于小规模系统或概念验证研究。另一方面，$Qualtran$ 实现了一种较新的技术，该技术引入辅助量子比特，在稀疏态上的表现更好，我们将在下一段讨论这一技术。

b. 稀疏态 仅含少量非零振幅的稀疏量子态 $s \ll 2^N$，其编译效率远高于任意稠密态。这种优势不仅源于其经典描述更精简，只需指定 s 个非零振幅；也源于其量子结构更简单，因此电路的资源成本可随 s 而非完整的希尔伯特空间维数 2^N增长。例如，Qualtran 针对 s-稀疏态 [45, 46], 提供了专用子程序，可按选定精度制备近似量子态，所需量子门数量为 $O(Ns)$，辅助量子比特数量则随系统规模 N线性增长。其他值得注意的例子包括 Gleinig和 Hoefler [47], 提出的一种算法，该算法使用 $O(Ns)$个量子门且不需要辅助量子比特，即可制备 N量子比特、 s-稀疏的量子态。最近，Luo和Li [48] 研究了不使用辅助量子比特、辅助量子比特数量相对于系统规模 N呈次线性，以及辅助量子比特数量不受限等情况。他们发现，在所有情况下，电路规模均为 $O(Ns)$，并可通过多对数因子进一步改进。后续研究探索了稀疏性与其他结构约束的结合。具体而言，de Veras 等人 [49] 和 Farias 等人 [50] 研究了支撑在低汉明重量基向量上的量子态。他们引入双重稀疏性的概念，指振幅数量 s和汉明重量 k均稀疏的量子态，并在不使用辅助量子比特的情况下实现了 $O(sk)$的门复杂度。

这些结果共同表明，利用稀疏度以及（在存在时）额外结构，能够实现远超通用程序能力范围的高效编译，同时使用由振幅构成的通用列表（或字典）来描述所需的量子态。

c. 专用程序 具有组合模式或内在对称性的量子态，通常可以比任意状态高效得多地制备，因为其结构限制了希尔伯特空间中的相关子空间。研究人员利用这些性质，为具有物理意义的状态族设计了高效的量子电路，例如 Motzkin [34]、 W [25] 以及 Dicke [26] 态。在最相关的例子中，Gopalakrishnan [35]利用下推自动机的概念，为 Motzkin 态设计了量子电路。在我们的工作中，我们重点介绍的状态族之一是 Dicke-k 态，即固定汉明重量 k的字符串的均匀叠加态。具体而言，我们将把我们的制备方法与 B¨artschi 和 Eidenbenz [51], 的方法进行比较，后者提出了一种使用受控双量子比特旋转的通用构造，门复杂度为 $O(Nk)$，深度为 $O(N)$，且不使用辅助量子比特。B¨artschi 和 Eidenbenz [52]后来通过利用全连接硬件连接改进了该结果，并实现了对数级深度降低，这与我们的 $TreeRLSP$ 方法类似。这些工作例证了，利用特殊状态族 的结构，能够实现可扩展的硬件感知编译，从而超越通用方法的能力。然而，专用程序的局限性 在于，用户必须显式调用或实现 它们，除非编译器具有有效的用户界面 和测试，从而在用户的描述中自动识别结构 。

C. 正则语言量子态

现在，我们介绍 RLS 的概念，它是本文工作的重点。给定一个字母表 Σ，我们用 Σ^*表示通过连接 Σ中的符号而形成的任意长度字符串的集合。语言 L 是 Σ^*的任意子集。语言通常通过字符串必须满足的规则来定义。具体而言， Σ 上的正则语言是由正则表达式定义的，而正则表达式由以下元素递归构造而成：空集 $\emptyset$ 和空字符串 ε （两者的区别在于，对于任意语言 L， $L\emptyset = \emptyset L = \emptyset$ ，而 $L\varepsilon = \varepsilon L = L$ ），单个字母 $a \in \Sigma$，以及连接（R_1R_2）、并集（$R_1 \cup R_2$）和克莱尼星号（$R_1^* := \varepsilon \cup R_1 \cup R_1R_1 \cup \ldots$）运算。对于 $N \in \mathbb{N}$， Σ^N 表示所有长度为 N的词 w 组成的集合，而任意语言 $L \subseteq \Sigma^N$ 都是有限的，因此也是正则的 [38]。

定义 1 （正则语言态）。 任意正则语言 L 都可映射为一族量子 *RLS*，记作 $\{|L_N\rangle\}_{N\in\mathbb{N}}$，其中每个 $|L_N\rangle$ 都是 L 中所有长度为 N的词 w 的均匀叠加，其物理维数为 $d = |\Sigma|$:

$$|L_N\rangle = \frac{1}{\sqrt{|L \cap \Sigma^N|}} \sum_{w\in L\cap\Sigma^N} |w\rangle. \tag{1}$$

我们先前提到的许多具有物理意义的量子态都是正则语言态，例如 GHZ态 [24] ($0^* \cup 1^* \cup \ldots$)、W态 [25] (0^*10^*)、迪克态 [26] ($0^*(10^*)^k$)，以及任意有限词集合的均匀叠加。

D. 有限状态自动机

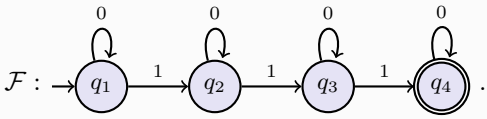
有限状态自动机是一种机器，可以判定给定单词是否属于特定的正则语言。正则语言可以通过正则表达式和有限自动机进行等价描述，这是自动机理论中的一项重要成果，称为克莱尼定理 [53]。这种等价性使正则语言能够以便于算法处理的形式表示，从而实现简洁描述和高效操作。下面给出确定有限自动机的形式化定义。

定义 2 (确定有限自动机**)。** 确定有限自动机定义为 $\mathcal{F} = \langle Q, \Sigma, \delta, r_0, F \rangle$，其中 Q 是内部状态的集合，$r_0 \in Q$ 是初始状态，$F \subseteq Q$ 是接受状态的集合，$\delta:\ Q \times \Sigma \to Q \cup \{\emptyset\}$ 是转移函数。若对于每个单词 $w = x_1 x_2 \ldots x_n \in L$，都存在一个状态序列 $(q_1, \ldots, q_n) \in Q^n$，使得 $q_{i+1} = \delta(q_i, x_{i+1})$ 对于 $i = 0, \ldots, n-1$成立，且 $q_n \in F$成立，则自动机 $\mathcal{F}$接受语言 L。

确定有限自动机通常表示为图，其中量子态是节点，转移是由符号标记的边，初始状态用指向它的箭头标记，而接受状态则用双圆圈表示。要判断一个单词是否属于该语言，确定有限自动机每次读取一个符号，沿着相应的转移前进（如果不存在转移，则进入指定的隐式错误状态 $\emptyset$），并且当且仅当运行结束于接受状态时才接受该单词。

示例 1：汉明重量为 k 的字符串/Dicke-k RLS

考虑恰好包含 3 个 1（汉明重量为 3）的词所构成的正则语言：$0^*(10^*)^3$。这也对应于 Dicke-3 态。两者都可以用这个确定有限自动机高效表示



(2)

限定为固定长度 N 字符串的语言，可以用有向无环图（DAG）识别，这与存储词的前缀树数据结构非常相似。我们可以将这类图描述为具有 $N+1$ 层，并由 N 组边连接相邻层（每组对应一个符号位置）。由此形成一类特定的确定有限自动机，我们在框架的某些部分会用到它们，并称之为有向无环图确定有限自动机。

定义 3 （有向无环 DFA）。*DAG-DFA* 是 $\mathcal{F}_{\mathrm{DAG}} = \langle \{Q^{(i)}\}_{i=0}^{N}, \Sigma, \{\delta^{(i)}\}_{i=1}^{N} \rangle$，其中 $Q^{(0)} = \{q_0\}$ 和 $Q^{(N)} = \{q_N\}$ 分别是唯一的量子态和接受状态，并且每个 $\delta^{(i)}:\ Q^{(i-1)} \times \Sigma \to Q^{(i)} \cup \{\emptyset\}$ 定义了连续层之间的转移。若对于每个单词 $w = x_1 \ldots x_N \in L$，都存在一个状态序列 $(q_0, \ldots, q_N)$，使得 $q_i \in Q^{(i)}$，且对于每个 $i = 1, \ldots, N$，都有 $q_i = \delta^{(i)}(q_{i-1}, x_i)$，则语言 $L \subseteq \Sigma^N$ 被 $\mathcal{F}_{\mathrm{DAG}}$ 接受。

将 DAG-DFA 推广到长度可变的有限语言，也称为确定性无环有限状态自动机（DAFSA）或有向无环词图（DAWG）。将范围限制为恰好包含 N 个符号的词，可确保：（1）量子态之间不会存在跨非连续层的转移；（2）在第 $N+1$ 层存在唯一的终态。

E. 矩阵乘积态

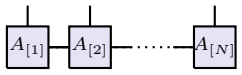
a. 张量网络 张量网络是表示量子多体态的强大工具，它们通过由低秩张量构成的网络来表达高度纠缠的波函数，并在局部编码关联。一个 N-qudit 量子态 $|\psi\rangle$ 是一个秩为 N 的张量，需要 $O(d^N)$ 个振幅；而张量网络将其分解为更小的张量，其内部（键）维度受某个参数 D 限制。当 D 保持有限时，参数数量按 $O(NdD^2)$ 的规模增长，从而为纠缠有限的状态提供指数级精简的描述。

b. 张量网络图示记法。 张量网络可以方便地用图示语言表示，其中张量表示为节点，其指标表示为连接线。每条边对应一个指标，连接两条边表示缩并。标量、向量和矩阵分别是具有零条、一条和两条边的张量；更高阶张量则有多条边。例如，秩为 3 的张量 $A^i_{\alpha\beta}$ 可视为一组矩阵 $\{A^i\}_{i=0}^{d-1}$，带有物理指标 i 和键指标 α，β。

c. 矩阵乘积态。 将这些局部张量沿其键收缩，可得到一种链状结构，称为矩阵乘积态（MPS）。矩阵乘积态已在量子多体物理领域得到广泛发展 [29, 37]，，并支撑了密度矩阵重整化群（DMRG）等典型算法 [54]，，以及一维系统中有能隙物质相分类等重要理论成果 [55, 56]。数据建模领域也广泛使用这类结构，并将其称为 张量列车 [57]。本文中，我们考虑 开放边界条件 （OBC），其中第一个和最后一个张量各只有一个虚指标；也就是说，$D_0 = D_N = 1$ 如下定义。

定义 4（开边界条件矩阵乘积态）。 由一组位点相关矩阵$\{A^i_{[n]}\}_{i=0}^{d-1}$定义 N 个格点上的矩阵乘积态，其中每个 $A^i_{[n]} \in \mathbb{C}^{D_{n-1}\times D_n}$ 对于 $n \in \{1,\dots,N\}$ 和 $D_0 = D_N = 1$。这些矩阵给出量子态

$$|\psi\rangle = \sum_{i_1\dots i_N=0}^{d-1} A^{i_1}_{[1]} A^{i_2}_{[2]} \dots A^{i_N}_{[N]} |i_1 i_2 \dots i_N\rangle \equiv$$



其中 d 是物理维数， D_n 是内部矩阵的键维数。我们称 $D = \max_n(D_n)$为矩阵乘积态的键维数 。

可能存在键维数不同、但表示相同量子态的等价矩阵乘积态。通常，要忠实表示给定量子态，位于站点 n的张量所需的最优（最小）键维数 D_n 等于系统相应二分划上的施密特秩。因此，矩阵乘积态的最优 键维数由最大施密特秩给出。

定义 5（施密特秩）。设 $|\psi\rangle \in (\mathbb{C}^d)^{\otimes N}$，并固定一个切分 $n \in \{1,\dots,N-1\}$，该切分诱导出二分划 $L = \{1,\dots,n\}$和 $R = \{n{+}1,\dots,N\}$。二分划上的施密特秩 $L|R$为 $\chi_n = \mathrm{rank}(\Psi_{(n)})$，其中 $\Psi_{(n)}$是根据切分 n对量子态 $|\psi\rangle$ 的振幅进行重塑后得到的 $d^n \times d^{N-n}$ 矩阵。我们将 $|\psi\rangle$ 的最大施密特秩 记为 $\chi = \max_n(\chi_n)$。

施密特秩可以为纯态的两体纠缠提供判据：跨越切分 $L|R$, $\chi_n = 1$ 当且仅当 $|\psi\rangle$ 是乘积态（无纠缠）时，而 $\chi_n > 1$ 则意味着该切分两侧存在纠缠。此外，对于约化态ρ_L，任意 R´enyi 熵（对于纯态而言，它量化了两体纠缠）都满足 $S_\alpha(\rho_L) \le \log(\chi_n)$；当 $\alpha = 0$ 或非零施密特系数均匀时取等号。我们的量子电路所需的资源将随 N 以及这一内在 量而扩展。

d. 与形式语言的联系。 矩阵乘积态可以用有限自动机进行自然表述 [27, 28], 从而确立张量网络与正则语言之间的对应关系。这种对偶性已被用于加速多体模拟 [28]并研究张量网络的表达能力 [27]。在此，我们利用这种对应关系，将正则描述高效地编译为量子电路。

III. RLS 编译流水线

我们的方法将正则语言与矩阵乘积态领域中相互独立的研究方向衔接并加以拓展，采用模块化设计，由一系列定义明确且可独立优化的步骤组成。图 1总结了该流水线，其步骤如下：

—**用户输入到自动机：** 输入的 RLS 可以指定为正则表达式、确定有限自动机或字符串集合，并转换为有限自动机。— **语言的补态（可选）：** 如果用户提出要求，则通过交换接受状态与非接受状态来对自动机取补。 — **自动机到矩阵乘积态：** 对确定有限自动机进行最小化，并将其映射为矩阵乘积态。 — **矩阵乘积态到等距映射：** 将矩阵乘积态分解为作用于一部分量子比特并描述量子电路的等距映射。 — **等距映射到量子电路**：将等距映射嵌入酉算子并转译为量子电路，同时使用通用门集中的量子门，并考虑连通性。

以下各节将详细介绍每个步骤。我们将针对量子比特的状态调整方法，即将字母表限制为 $\Sigma = \{0,1\}$。不过，同样的方法也很容易扩展到 qudit 的状态和更大的字母表。

A. 用户输入到自动机

用户可以通过三种方式指定量子态：提供明确的字符串集合、使用正则表达式，或直接提供一个接受该语言的确定有限自动机。编译流水线的第一步会将输入转换为确定有限自动机，这是我们中间表示的第一个工具。下面将讨论这三种输入类型。

1. *Set of strings*

当用户通过有限集合（或列表）描述量子态，其中包含 s 个互不相同的字符串 $L = \{w_1, w_2, \ldots, w_s\} \subset \Sigma^N$ 时，我们可以构造一个接受这些字符串的有向无环图确定有限自动机：每个字符串都对应自动机中从初始状态到唯一接受状态的一条路径；通过共享前缀来合并量子态，以保持确定性。此过程需要 $O(sN)$ 次操作。

2. *Regular expression*

当用户通过正则表达式（以及系统规模 N）描述量子态时，我们需要将正则表达式编译为确定有限自动机。这可以通过标准技术完成，例如先构建非确定自动机（非确定有限自动机）[58, 59]再将其确定化 [38, 60, 61]。虽然指定正则表达式可能比枚举整个量子态支撑简洁得多，但将非确定有限自动机确定化可能产生随非确定有限自动机规模呈指数增长的开销，更宽泛地说，也随正则表达式描述规模增长。我们并不预期描述常见物理态的正则表达式会出现这种膨胀，但对于特定或对抗性正则表达式则可能发生。在我们的编译时分析（第四节）中，我们将强调这种可能性，并从确定有限自动机表示开始分析，从而完全避免这一潜在指数开销的风险。

3. *DFA*

在某些情况下，用户可能希望直接以确定有限自动机指定量子态。此功能将为用户提供一种变通方法，从而能够对严格正则语言之外的情形进行高效描述。例如，由括号配对良好的字符串构成的戴克语言是上下文无关、非正则语言的典型例子，并对应于量子莫茨金态 [34, 35]。然而，任何有限语言都是正则的，因此将其限制为长度为 N的字符串时，可以用具有 $O(N)$个状态的确定有限自动机来刻画，每个待闭合的左括号对应一个状态。

定义 6 (D_{init} 和 D_{minDFA})。 我们将 D_{init} 称为确定有限自动机的内部状态数量，或有向无环图确定有限自动机中某一层的最大状态数$\max_{i\in\{0,\ldots,N\}} |Q^{(i)}|$ 。对于最小且等价的自动机，我们将此参数称为D_{minDFA}。一般而言，有 $D_{\text{minDFA}} \le D_{\text{init}}$。

a. 确定有限自动机最小化 每个正则语言都对应唯一的最小确定有限自动机（按状态数量计），除内部状态重新标记外，构成其规范形式 [38]。使用霍普克罗夫特算法 [38, 62]，，可在 $O(D_{\text{init}}\log(D_{\text{init}}))$ 时间内将确定有限自动机最小化；而有向无环图确定有限自动机可在 $O(ND_{\text{init}})$ [63]，内完成最小化，同时保留其结构。最小确定有限自动机将输入规范化，并为流水线的其余部分奠定基础，从而减轻矩阵乘积态规范化的计算负担。

B. 正则语言补态 Sta tes

在某些应用 中，准备一个补RLS 很有用例如，参见 [64, 定义 9] 和 [65]) .

定义 7 (补集 RLS)。 与正则语言 L 相关联的量子补集 *RLS* 族，是与其补正则语言相关联的 $RLS^{\neg}$，$L := \Sigma^* \setminus L$。

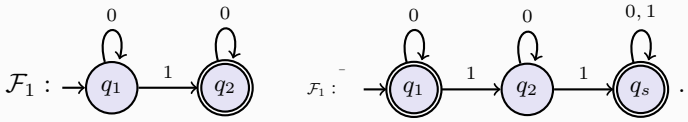
重要的是，如果某个 RLS 有简洁的描述，它的补态也有。在我们的框架中，用户可以使用基础 RLS 描述并设置一个标志来请求合成补态，我们的流水线会高效地进行适配。当前的编译器（*Qiskit* 和 *Qualtran*）不提供此功能。此外，如果基础 RLS 是稀疏的，其补态将具有稠密支撑，因此这些编译器无法通过字符串集合 或字典 描述进行编译。相反，我们会按照上一节的说明，为基础 RLS 创建一个确定有限自动机；如果设置了补标志，则按如下方式修改 DFA 中间表示：

1. 如果自动机是有向无环图确定有限自动机，将补态限制为长度为 N 的字符串（即 $\bar{L} \cap \Sigma^N$）并保留有向无环图确定有限自动机结构会很方便，因为这有助于之后构造更好的矩阵乘积态。这个补态有向无环图确定有限自动机可以在 $O(ND_{\text{minDFA}})$ 步内高效计算。找到第一个存在未定义转移的层（即 $\delta^{(i)}(q_{i-1},\ x_i) = \emptyset$）后，我们在之后的每一层都添加一个专用的汇状态 。所有之前未定义的转移现在都指向下一层的汇状态，而每一层的汇状态则对所有可能的符号转移到后一层的汇状态。移除先前的接受状态及其传入转移，并将最后一层的汇状态设为唯一接受状态。由此得到的有向无环图确定有限自动机识别 $\bar{L} \cap \Sigma^N$。我们再次对其进行最小化；根据构造，最终的 D_{minDFA} 至多增加 1。

2. 如果自动机是一般的确定有限自动机，则可通过添加一个显式的陷阱状态、将未定义的转移重定向至该状态，并将非接受状态改为接受状态、将接受状态改为非接受状态来得到补态。随后进行最小化[7]，可确保所得结果是针对 L 的唯一最小确定有限自动机。即使在这种情况下，最终的 D_{minDFA} 也至多增加 1。

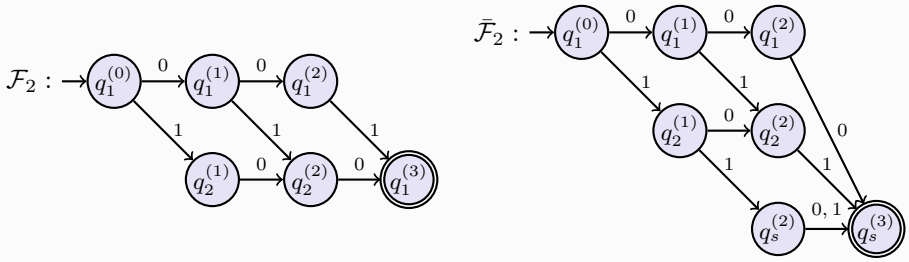
示例 2： W态/独热编码字符串确定有限自动机

让用户指定正则语言 $L = 0^*10^*$，其量子 RLS 对应于 W态，以及其补集L。分别接受这些语言的最小确定有限自动机为：



(3)

如果用户提供长度为 3 的 W态的字符串集合，并要求同时生成补集，我们的流水线将生成以下 4 层最小有向无环图确定有限自动机：



(4)

注意， $D^{\mathcal{F}_1}$最小确定有限自动机 $= D^{\mathcal{F}_2}$最小确定有限自动机 $= 2$ 和 $D^{\bar{\mathcal{F}}_1}$最小确定有限自动机 $= D^{\bar{\mathcal{F}}_2}$最小确定有限自动机 $= 3$。我们的有向无环图确定有限自动机构造方法，即使从字符串集合描述而不是从紧凑且更通用的正则表达式开始，也能恢复出一个较小的 D_{minDFA} 。

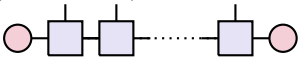
C. 从自动机到矩阵乘积态

编译流水线的下一步需要将最小确定有限自动机映射为矩阵乘积态。直观来说，确定有限自动机的序列结构为MPS表示提供了天然的骨架。确定有限自动机的状态对应键指标，而字母表 Σ 中的输入符号对应物理指标。从某种意义上说，矩阵乘积态编码了确定有限自动机的邻接矩阵：左边界编码初始状态，右边界则投影到接受状态上。

下面我们介绍两种实用的构造方法，分别从最小确定有限自动机或有向无环图确定有限自动机出发。在这两种情况下，矩阵乘积态的键维数均为 $D = D_{\text{minDFA}}$ (定义 6)。

1. *Uniform-bulk construction (DFA)*

这种对应关系产生了一种矩阵乘积态，其所有张量都相同（即具有均匀体区），唯有两个边界向量例外。

引理 1 （确定有限自动机到均匀体区矩阵乘积态 [27]）。 给定一个确定有限自动机 $\mathcal{F} = \langle Q, \Sigma, \delta, I, F\rangle$，它接受正则语言 L，则可以得到与之相关的量子 *RLS* $\{|L_N\rangle\}_{N\in\mathbb{N}}$ 族的矩阵乘积态描述，使得 $|L_N\rangle$:  $= v_l \quad A \quad A^{N\ times} \quad A \quad v_r$，其中边界向量和体区张量定义如下

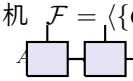
$$v_l := \sum_{i\in I}\langle i|, \qquad i \xrightarrow{x} A \xrightarrow{} j := \begin{cases} 1 & if\ j = \delta(i,x), \\ 0 & otherwise, \end{cases} \qquad v_r := \sum_{f\in F}|f\rangle. \tag{5}$$

该矩阵乘积态的键维数为 $D = |Q|$。

换言之，体区中的每个张量 $A^x_{ij} \equiv \{A^x\}_{x\in\Sigma}$ 都编码了确定有限自动机图的整个邻接矩阵，其中每个矩阵 A^x 编码由符号 x 标记的边。

2. *Non-uniform construction (DAG-DFA)*

利用引理 1 [27], ，所得矩阵乘积态的键维数将随内部状态总数增长。然而，可以利用有向无环图确定有限自动机的分层结构，得到一种不同的非均匀 MPS 构造 其键维数等于每层中的最大节点数，并保持 $D = D_{\text{minDFA}}$。

引理 2 （有向无环图确定有限自动机到非均匀矩阵乘积态）。 给定一个接受有限语言 $L \subseteq \Sigma^N$ 的有向无环图确定有限自动机 $\mathcal{F} = \langle \{Q^{(i)}\}_{i=0}^N, \Sigma, \{\delta^{(i)}\}_{i=0}^N \rangle$，可以得到与之关联的量子 *RLS* $\{|L_N\rangle\}_{N\in\mathbb{N}}$ 的矩阵乘积态描述，使得$|L_N\rangle$：$= A_{[1]}$ A  $A_{[N]}$,  中张量定义为

$$\begin{array}{c} x \\ i \text{—}\boxed{A_{[n]}}\text{—} j \end{array} = \begin{cases} 1 & \textit{if } j = \delta^{(n)}(i,x), \\ 0 & \textit{otherwise.} \end{cases} \tag{6}$$

每个 $A^x_{[n]}$ 是 $D_{n-1} \times D_n$，其中 $D_n = |Q^{(n)}|$。该矩阵乘积态的键维数为 $D =$ 最大值$_n$（$|Q^{(n)}|$） .

证明。 正确性可由 RLS（1）、矩阵乘积态（4）和有向无环图确定有限自动机（3）的定义直接得出。特别地，注意对于任意单词 $w = x_1 \dots x_N$，确定有限自动机存在一条接受路径（$q_0, \dots, q_N$）当且仅当 $A^{x_1}_{[1]} A^{x_2}_{[2]} \dots A^{x_N}_{[N]}$ =1，或者等价地， $A^{x_n}_{[n]} = 1$， $\forall n \in \{1, \dots, N\}$。事实上，根据（6），每个 $A^{x_n}_{[n]}$ 都可以看作图的示性函数 $\Sigma \times Q^{(n-1)} \times Q^{(n)} \to \{0,1\}$。或者等价地， $A^x_{[n]}$ 编码了有向无环图确定有限自动机中层 $n-1$ 与 n 之间、针对符号 x的边所构成的二分图的邻接矩阵。这也意味着 $D_n = |Q^{(n)}|$。 □

例 3： W-态/独热编码字符串矩阵乘积态

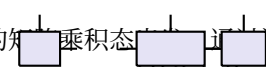
将引理 1 的构造应用于确定有限自动机 $\mathcal{F}_1$ (3) 可得到以下矩阵乘积态：

$$\begin{cases} \begin{array}{c} 0 \\ \text{—}\boxed{A}\text{—} \end{array} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & \begin{array}{c} 1 \\ \text{—}\boxed{A}\text{—} \end{array} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \\ \text{—}v_l\text{—} = (1,0), & \text{—}v_r\text{—} = (0,1)^T. \end{cases} \tag{7}$$

对有向无环图确定有限自动机 引理 2 应用 $\mathcal{F}_2$ (4)，可得到一个等价的矩阵乘积态，其键维数相同 $D = 2$。 从 $\mathcal{F}_1$ 和 $\mathcal{F}_2$ 出发，可以得到补态的矩阵乘积态。

D. 矩阵乘积态到等距映射

此步骤标志着后端部分的开始，因为剩下的工作是将 MPS 中间表示综合为量子电路。我们概述两种方法：一种顺序 生成方案 [30] ，称为 SeqRLSP，适用于线性最近邻连接；另一种树形/并行方案 [31, 32]，称为 TreeRLSP，最适用于全连接设备。这两种方法都先将矩阵乘积态转换为等距映射网络。具体而言，若满足 $\sum_x A^{x\dagger}_{[n]} A^x_{[n]} = I_{D_n}$，则局部张量 $A^x_{[n]}$ 是左等距的；若满足 $\sum_x A^x_{[n]} A^{x\dagger}_{[n]} = I_{D_{n-1}}$，则是右等距的。

我们从一个键维数为 D_{minDFA}、形式为 $A_{[1]} \quad \dots \quad A_{[N-1]} \quad A_{[N]}$ 的矩阵乘积态  设置 引理 1 中的 $A^x_{[1]} = \langle v_l | A^x$， $A^x_{[N]} = A^x | v_r \rangle$，以及 $A^x_{[i]} = A^x$ 对每个 $i \in \{2, \dots, N-1\}$，对其均匀体区构造进行调整。等距化过程取决于所选方案。

1. *Sequential SVD pass (SeqRLSP)*

在顺序方法中，我们沿张量链进行两轮奇异值分解扫描，将矩阵乘积态转换为规范形式：先从右向左，再从左向右，如 图 2 所示。这一双向扫描的奇异值分解过程有两个作用：(1) 生成等距映射；(2) 消除矩阵乘积态张量中的冗余维度，使局部键维数与各个切分处的施密特秩 χ_n 相匹配。

由于 χ_n 是通过重排 $|\psi\rangle$ 在第 n 个切分处的振幅得到的 $2^n \times 2^{N-n}$ 矩阵 $\Psi^{(n)}$ 的秩，因此我们有 $\chi_n \leq 2^{\min\{n, N-n\}}$。这一性质保证了边界附近的键维数较小，

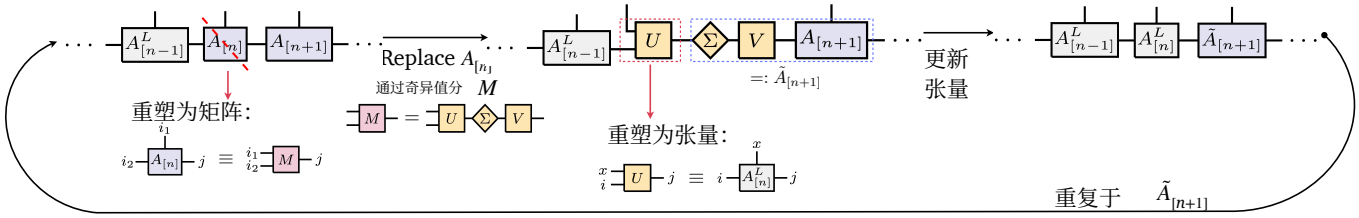


图 2：从左到右执行顺序 SVD 的示意图。以类似方式，从相反边界镜像执行该过程，即可完成从右到左的 SVD 扫描。所得的每个张量 $A^L_{[n]}$ 都是大小为 $\chi_{n-1} \times \chi_n$的等距算子，其中 $\chi_n \leq \min\{\chi, 2^n, 2^{N-n}\}$，适用于 $n \in \{1, N-1\}$。有 $D_0 = D_N = 1$。

从而支持量子比特复用，且除系统规模外无需额外辅助量子比特（见 图4a）。总体而言，该规范矩阵乘积态的键维数是最优的，等于最大施密特秩 $D = \chi$。这一遍消除了 D_{minDFA}与 χ之间的任何差异。

该过程总共需要执行 $2N$ 次 SVD。由于计算一个 $m \times n$ 矩阵的 SVD 的代价为$O(\min(mn^2,\ m^2n))$，且算法过程中各中间维数均不超过 D_{minDFA}，因此此遍历的总计算代价为 $O(ND^3_{\text{minDFA}})$。

2. *Tree-based SVD pass (TreeRLSP)*

在基于树的方法中，通过上述顺序 SVD 遍历得到的规范矩阵乘积态会经过后处理，转化为具有 $O(\log N)$ 层和 $O(N)$ 个等距映射的树状张量网络 [31, 32, 66]，具体采用逐层的分层 SVD 方案（图 3）。在每一步中，树中最上方的垂直键承载物理维数（量子比特为 2），水平键的维数不超过 χ，其余垂直键的维数均不超过 χ^2。将两个张量重塑为矩阵 $M_{[n]}$时，两个上方垂直键的维数各自不超过 χ^2，两个下方垂直键的维数各自不超过 χ。因此，该遍历需要对至多为 $\chi^4 \times \chi^2$大小的矩阵执行 $O(N)$ 次 SVD，总运行时间为 $O(N\chi^8)$。

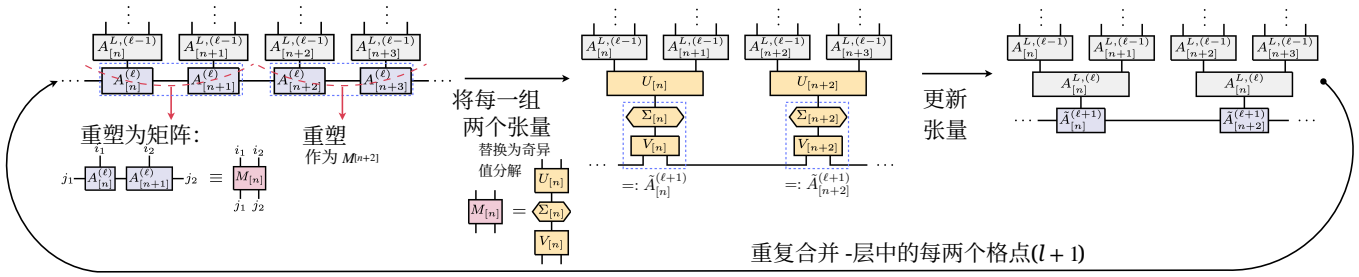
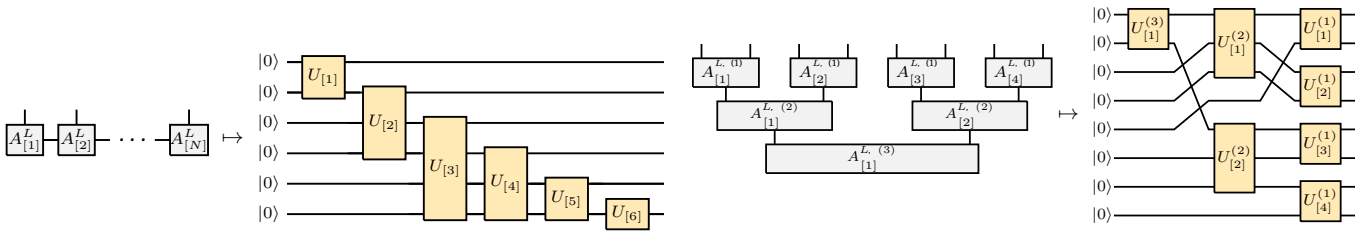


图 3：第 ℓ 层基于树的奇异值分解过程示意图。树中最上方的竖直键承载物理维数（量子比特为 2），水平键的维数不超过 χ，其余竖直键的维数均不超过 χ^2。当某一层中的张量数量为奇数时，未配对的张量会传递到下一层。

E. 等距映射到量子电路

在上一步中，我们得到了目标 RLS 的 TN 描述。最后一步的挑战是将 TN 描述中的等距张量转换为量子电路操作，并根据目标硬件设备的通用量子门集及其连通性，对这些操作进行转译。

首先，由于我们的目标是创建一个作用于量子比特的量子电路，因此需要对等距张量进行适当填充，使每个后续的键维数至少增至 d 的幂，从而量子电路操作的映射见（8）和图4，它们说明了张量网络拓扑如何影响所需的量子目标连通性。SeqRLSP 可原生映射到线性最近邻，而 TreeRLSP 中的操作作用于几何非局域量子比特，这也说明了它为何更适合全连接设备。



(a) SeqRLSP；$(\chi_0, \ldots, \chi_6) = (1, 2, 4, 8, 4, 2, 1)$。

(b) TreeRLSP；$(\chi_0, \ldots\ldots, \chi_8) = (1, 2, 2, 2, 2, 2, 2, 2, 1)$。

图 4：TN 与量子电路操作之间的可视化映射。(a) SeqRLSP 的映射。(b) TreeRLSP 的映射。

第三，公式（8）中的量子操作通过合并输入腿和输出腿，从四维张量形式转换为等距矩阵（二维）。可以看出，SeqRLSP 的等距矩阵最大尺寸为 $O(2\chi \times \chi)$，而 TreeRLSP 的最大尺寸为 $O(\chi^4 \times \chi^2)$。这些等距矩阵描述了一种量子电路操作：将一个 m 量子比特态以及 $n-m$ 个初始化为 $|0\rangle$的量子比特映射为另一个 n 量子比特态，$n \geq m$。

$$\text{SeqRLSP}: \quad \begin{array}{l}\alpha \\ \langle 0| \end{array} \boxed{U_{[n]}} \begin{array}{l} i \\ \beta \end{array} \;=\; \overset{i}{\underset{\alpha}{\boxed{A^{L}_{[n]}}}}\; \beta\,, \qquad \text{TreeRLSP}: \quad \begin{array}{l}\alpha \\ \langle 0| \end{array} \boxed{U^{(\ell)}_{[n]}} \begin{array}{l} \beta \\ \gamma \end{array} \;=\; \alpha \boxed{A^{L,(\ell)}_{[n]}} \begin{array}{l} \beta \\ \gamma \end{array}. \tag{8}$$

第四，给定一个 $2^n \times 2^m$ 等距矩阵，我们使用内置的 *Qiskit* 转译方法 [41, 42] 将其映射为基于所选通用门集的电路，其门（受控非门）成本 $O(2^{m+n})$，并使用经典预处理时间 $O(n2^{2m+n})$ [67]。这为 SeqRLSP 和 TreeRLSP 中实现单个操作分别带来了门成本 $O(\chi^2)$和 $O(\chi^6)$。由于两个流水线 SeqRLSP 和 TreeRLSP 中均有 $O(N)$个此类等距映射，因此门总数会产生 N的倍数因子。一旦电路用填充后的等距矩阵表示，就可以使用任何标准转译器完成合成，使我们的流水线能够灵活适应这一阶段的改进。这也是我们的编译器相较于专用例程（例如 Dicke/W）可能产生额外开销的地方：我们不会详细分析所得的等距映射，也不会针对基本操作优化矩阵乘积态的处理流程，这可能会掩盖可利用的结构，从而错失进一步降低资源的机会。此类针对性优化超出了本文的范围。

IV. 编译成本与电路保证

在本节中，我们回顾、讨论并从理论上界定我们流水线的最坏情况性能，涵盖编译时间与量子资源两个方面。在此过程中，我们论证了设计选择。此外，我们将 RLS 的施密特秩与其补态的施密特秩紧密联系起来，关联它们的纠缠，并表明我们的方法能够以相同的标度合成二者。表 I 总结了两种后端——SeqRLSP 与 TreeRLSP——的标度。

a. 编译成本 我们研究编译时间性能如何作为系统规模 N、初始确定有限自动机状态数 $|Q|$以及最小确定有限自动机规模 D_{minDFA} （定义6）的函数。

构建自动机的前端成本取决于用户描述：（1）对于一个字符串集合，其规模为 s，有向无环图确定有限自动机的构造成本为 $O(Ns)$；（2）对于正则表达式，确定化相对于描述规模在最坏情况下可能呈指数级，不过正则表达式到确定有限自动机的流水线在实践中通常非常有效；（3）对于用户提供的确定有限自动机，编译器直接进行最小化。

从自动机出发，我们可以如下界定编译时间。

定理 1（自动机的编译时间）。 设目标 *RLS* 由一个具有 $|Q|$ 个内部状态的确定有限自动机表示，并设 D_{minDFA} 如 定义 6所述。那么，对于 *SeqRLSP*，电路综合的开销为 $O\left(|Q|\log(|Q|) + ND^3_{\text{minDFA}}\right)$；对于 *TreeRLSP*，开销为 $O\left(|Q|\log(|Q|) + N\left(D^3_{\text{minDFA}} + \chi^8\right)\right)$，相差至多 $O(\log(\chi))$ 倍。对于有向无环图确定有限自动机，可以去掉 $\log(|Q|)$ 因子。

证明。处理可选的补标志需要 $O(|Q|)$（第 7 节）。Hopcroft 最小化算法的运行时间为 $O(|Q|\log(|Q|)$，在无环（有向无环图确定有限自动机）情形下，$\log(|Q|)$ 因子可省去（第三节 A）。根据引理 1 和 2 构造矩阵乘积态，所需时间与最小确定有限自动机的量子态数成正比，其上界为 $O(ND_{\text{minDFA}})$。顺序SVD遍历的开销为 $O(ND^3_{\text{minDFA}})$（第三节 D 1），树网络的开销为 $O(N\chi^8)$（第三节 D 2）。最后，等距映射综合对于 SeqRLSP 的开销为 $O(N\chi^3)$，对于 TreeRLSP 的开销为 $O(N\chi^8)$，忽略 $\log(\chi)$ 因子（第III.E节）。

□

结合一个字符串集合输入，我们可以刻画该流水线的端到端复杂度。

推论 1 （从字符串集合编译的开销）。 设目标 *RLS* 由一个基数为s的字符串集合 指定。端到端电路合成使用有向无环图确定有限自动机，*SeqRLSP* 的开销为 $O\left(N\left(s+D_{\mathrm{minDFA}}^{3}\right)\right.$，*TreeRLSP* 的开销为 $O\left(N\left(s+D_{\mathrm{minDFA}}^{3}+\chi^{8}\right)\right.$。

证明。 有向无环图确定有限自动机可在 $O(Ns)$ 时间内构造，且具有同样多的内部状态。 □

这些界限凸显了确定有限自动机最小化的重要性。无论如何，顺序SVD遍历都能恢复键维数为 χ 的最优MPS；但跳过最小化会在 ND^3 项中将 D_{minDFA} 替换为 D_{init}。存在这样的例子： D_{minDFA} 对于 N是常数，但 D_{init} 并非常数。对于字符串集合 输入，这一优势尤为明显：例如， N量子比特的 W态可由 $s=N$个字符串表示，从而得到具有 $D_{\mathrm{init}}=N$的有向无环图确定有限自动机，但 $D_{\mathrm{minDFA}}=2$（见示例 1）。因此，确定有限自动机最小化将 N^3 项替换为一个常数。随着支持集增大，这一差距也会扩大（例如，Dicke-k： $\binom{N}{k}$ 与 $k+1$），因为我们发现 D_{minDFA} 能很好地近似 χ。

最后，对于字符串集合输入，人们或许会想知道为何要使用 DFA 中间表示。原因在于，它能提供从稀疏描述高效构造初始MPS的方法。事实上，教科书式TN过程 [54, 第 4.1.3 节，图5] 要求执行 N 对尺寸最大为 $2^{N/2}\times 2^{N/2}$的矩阵进行精确奇异值分解，即使对于稀疏描述，也会造成指数级开销。

b. 补态*Schmidt*秩在给出电路保证和资源界之前，我们先说明 RLS 及其补态的 Schmidt 秩之间的关系。尽管二者的支撑集大小可能相差呈指数级，但它们的两体纠缠紧密相关，这意味着无论合成哪种量子态，所需的量子资源基本相同。

定理 2（补态RLS的Schmidt秩）。对于每个切分，*RLS* 与其补态的 *Schmidt* 秩最多相差一。等价地， $|\chi_n-\bar{\chi}_n|\leq 1$。

证明。 令 $\chi_n,\bar{\chi}_n$ 分别表示 RLS 及其补态在第 n 个切分处的施密特秩。对于每个 N，补集 RLS 可写为 $|\bar{L}_N\rangle=|+\rangle^{\otimes N}-|L_N\rangle$。因此，其在第 n 个切分处重塑后的振幅矩阵为 $\Psi^{(n)}_{|\bar{L}_N\rangle}=\Psi^{(n)}_{|+\rangle^{\otimes N}}-\Psi^{(n)}_{|L_N\rangle}$，而 $\bar{\chi}_n$是该矩阵的秩。由于 $\mathrm{rank}(\Psi^{(n)}_{|L_N\rangle})=\chi_n$且 $\mathrm{rank}(\Psi^{(n)}_{|+\rangle^{\otimes N}})=1$，再结合任意矩阵 A,B均满足 $|\mathrm{rank}(A)-\mathrm{rank}(B)|\leq \mathrm{rank}(A-B)\leq \mathrm{rank}(A)+\mathrm{rank}(B)$，结论得证。 □

c. 电路保证与资源数量 最后，我们讨论两个后端所需的最坏情况下的量子资源。对于 SeqRLSP，总成本主要由 $O(N)$ 层酉算子决定，每层最多作用于 $\lceil\log_2(\chi)\rceil+1$ 个相邻量子比特。

定理 3 (SeqRLSP 资源)。 设 N 为系统规模，且 *RLS* 的最大施密特秩为 χ。*SeqRLSP* 流程生成一个 *LNN* 电路，在无需辅助量子比特的情况下，使用 $O(\chi^2N)$ 深度和基本门制备 *RLS*（或其补态）。

证明。 电路的正确性源于 RLS 的定义（定义 1）、DFA-MPS 等价性（引理 1和 2）以及 MPS 顺序制备协议 [30], ，详见第三节 D。在 SeqRLSP 中，电路由一系列 $O(N)$ 个最近邻等距算子组成，每个等距算子的规模为 $O(\chi)$（第三节 D）。每个等距算子都可以在 LNN 架构上分解为 $O(\chi^2)$ 个基本门（第三节 E）。由于这些等距算子沿链依次应用，门总数和深度均为 $O(\chi^2N)$。根据定理 2，补态在每个切分处的施密特秩至多相差一，因此相同的资源界同样适用。 □

对于 TreeRLSP，正确性证明与定理 3相同，只需将顺序方案替换为树形方案（第三节 D 2和 [31, 32]）。该树有 $O(\log_2 N)$ 层，每层的等距映射数量是上一层的一半，因此总共有$O(N)$ 个等距映射。最坏情况下，一个等距映射作用于 $n=\lceil\log_2(\chi^4)\rceil$ 个量子比特，其维数为 $\chi^4\times\chi^2$(第三节 D 2)。由于这些等距映射可能耦合非相邻量子比特，因此该后端更适用于囚禁离子 [68] 或中性原子 [18]等全对全平台。我们在下述定理中汇总这些考量。

定理 4 (TreeRLSP 资源)。 设 N 为系统规模，且该正则语言的最大施密特秩为 χ。*TreeRLSP* 流水线生成一个全连接电路，用于制备该正则语言（或其补态），所需深度为 $O(\chi^6\log(N))$，基本门数量为 $O(\chi^6N)$，且无需辅助量子比特。

证明。 推导过程与定理 3等价，并结合上述考量即可。 □

方法	总门数	深度	连通性	编译时间（确定有限自动机输入）	编译时间（ s 字符串集合）
SeqRLSP	$O(\chi^2 N)$	$O\left(\chi^2 N\right)$	局部	$O\left(\lvert Q\rvert\text{对数}(\lvert Q\rvert) + N D^3_{\text{minDFA}}\right)$	$O(N(s + D^3_{\text{minDFA}}))$
TreeRLSP	$O(\chi^6 N)$	$O\left(\chi^6 \log(N)\right)$	全对全	$O\left(\lvert Q\rvert\log(\lvert Q\rvert) + N(D^3_{\text{minDFA}} + \chi^8)\right)$	$O(N(s + D^3_{\text{minDFA}} + \chi^8))$

表 I：对于一个正则语言及其补态，SeqRLSP 和 TreeRLSP 的资源与编译时间扩展。 N 是系统规模， χ 是最大施密特秩（定义 5）， $|Q|$是初始自动机状态数，并且 如定义 6所示。两种方法均不使用辅助量子比特。“编译时间（DFA 输入）”不包括 DFA 实例化；“编译时间（字符串 s集合）”为端到端时间。

V. 数值实验

本节给出数值结果，以刻画我们的编译策略所带来的优势，以及两个后端 SeqRLSP 和 TreeRLSP 的资源需求。我们将它们与文献中广泛使用的方法进行基准测试，并比较资源数量的标准评价指标：电路深度、量子门总数，以及生成电路描述所需的经典编译时间。

编译时间使用 Intel(R) Xeon(R) Platinum 8360Y CPU @ 2.40GHz（Ice-lake）的 72 个核心进行测量，该处理器配备 512GB 随机存取存储器（限制为 100GB ）。所有数值实验均使用 Python[1] 实现。对于 RLS 优化，我们使用 Pyformlang 软件包 [69], ；对于矩阵乘积态计算，我们使用 NumPy [70]。在以深度和门数报告成本时，我们使用原生的 *Qiskit* 函数，将给定的任意酉算子（或等距算子）转译为通用门集 [中的量子门：受控非门、 RZ(θ) $\sqrt{\text{X}}$ 、 X $\mathbb{I}$]。将矩阵转译为量子电路的成本计入编译时间。使用字符串集合 （列表或字典）描述时，我们不计入用户创建列表所需的时间。

我们进行基准比较的方法，是在流行的量子计算框架中实现、并在文献中讨论过的成熟态制备方法（参见第二节 第二节IIB）。我们将其总结如下：

- **Qiskit** [41, 42]: 这种通用方法的复杂度随系统规模 N呈指数增长；因此，我们将基准测试限制在最多包含 $N \leq 16$ 个量子比特的系统上。
- **Qualtran** [45, 46]: 一种近似稀疏态制备方法，其辅助量子比特数量随 N增长，且其能力本质上仅限于 $N \leq 32$。整个过程中，目标相对精度固定为 2^{-3} 。
- **Gleinig and Hoefler** [47]: 一种无需辅助量子比特的精确稀疏态制备算法。
- **B¨artschi and Eidenbenz** [51]: 一种专用程序，用于在线性最近邻硬件上确定性地制备Dicke态。

A. Dicke态

Dicke态 [26] 是一类可在我们的框架内高效制备的量子态。Dicke-k 态是所有长度为 N 且汉明重量为 k的比特串的均匀叠加。

在图5中，我们展示了使用 $k = 3$ 在不同系统规模 N下制备Dicke态的结果。我们将SeqRLSP和TreeRLSP与Qiskit [41], 、Qualtran [45, 46], 、Gleinig和Hoefler [47], 以及B¨artschi和Eidenbenz[51]进行比较。尽管该量子态的稀疏度为 $\binom{N}{k}$，其标度为 $O(N^{\min(k,N-k)})$，但它具有键维数为 $\chi = k + 1$ 的精确 MPS 表示。这种关于 k 的幂律标度给稀疏态制备方法带来了挑战，因为稀疏度会随 N迅速增长。如图5(c)所示，对于 $N \approx 20$的中等规模系统，稀疏方法的编译时间超过了100秒。B¨artschi和Eidenbenz [51] 专门开发的先进Dicke态编译器，可在 $O(N)$深度和 $O(kN)$个量子门内，利用线性最近邻连接制备该量子态。SeqRLSP编译器在 N中实现了相同的标度，总门数为 $O(k^2 N)$，因为 $\chi = k + 1$ （见表 I）。

在全连接架构上，制备开销可以降至 $O(k \log N)$ [52]。同样，TreeRLSP 方法实现了相同的 N-阶扩展，但其 k-依赖性较不理想，电路深度为 $O(k^6 \log N)$。两种情况下 k-阶扩展未达最优，可能是因为使用了通用电路分解后端（见 第三节E）。

[1] 我们的代码位于 https://github.com/reinisirmejs/RLSComp。

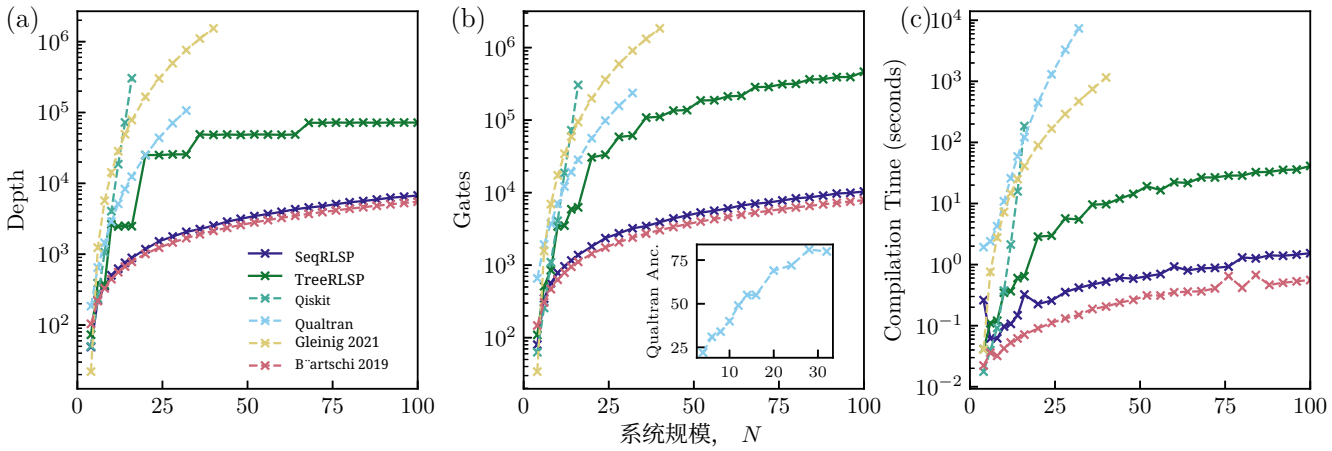


图5：不同系统规模下的Dicke态制备成本−3 N。我们比较SeqRLSP、TreeRLSP、*Qiskit*、*Qualtran*、Gleinig和Hoefler [47], 以及B¨artschi和Eidenbenz [51]。（a）量子电路深度。（b）门总数。插图显示了*Qualtran*使用的辅助量子比特数量。（c）编译时间（秒）。

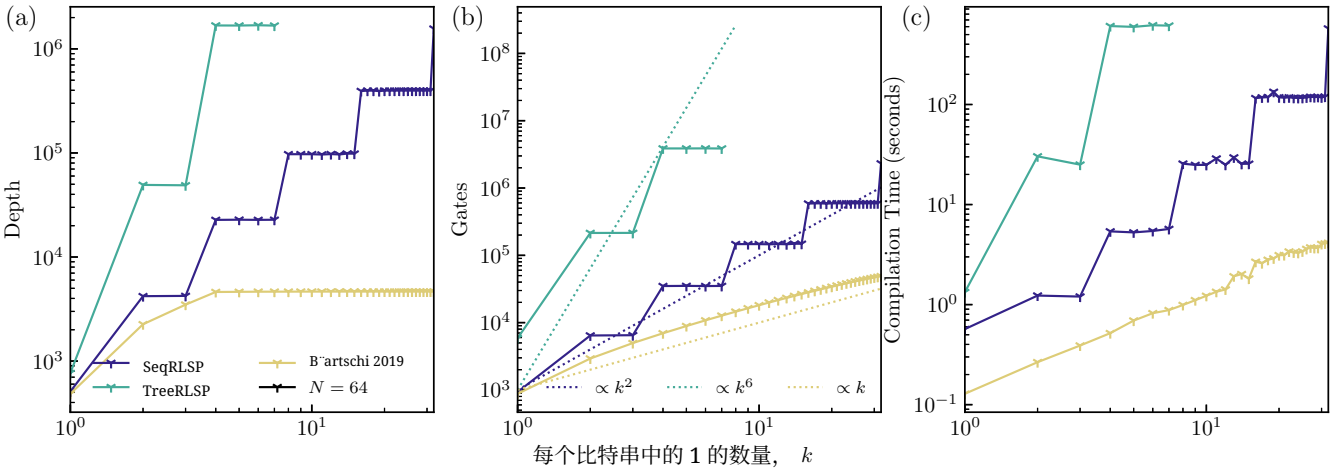


图6：在 $N = 64$ 个量子比特上制备−k Dicke态的成本随汉明重量 k的变化。我们比较SeqRLSP、TreeRLSP和B¨artschi和Eidenbenz [51]。（a）量子电路深度。（b）门总数。虚线趋势线显示预期的理论 k标度。（c）编译时间（秒）。

我们观察到，SeqRLSP 与 B¨artschi 和 Eidenbenz [51] 在 N 上呈现相同的扩展规律，其中 B¨artschi 和 Eidenbenz [51] 在电路深度、门数和编译时间方面仅以较小的常数优势胜过 SeqRLSP。这表明我们通用的 SeqRLSP 方法十分有效，其性能与专为 Dicke态 设计的方法非常接近。TreeRLSP 起初开销较高，但其 $O(\log N)$ 深度扩展最终将优于 SeqRLSP 的 $O(N)$ 扩展规律。我们将在 第V.B节 中进一步展示这一现象，并分析 W态制备。

在图6中，我们给出了系统规模为 $N = 64$ 且汉明重量 k变化时的Dicke态制备结果。我们比较了 k标度下的SeqRLSP和TreeRLSP与B¨artschi和Eidenbenz [51]。在图6（b）中，我们通过数值验证了 k标度。SeqRLSP和B¨artschi和Eidenbenz [51] 与理论结果较为吻合；而对于TreeRLSP，观测到的标度似乎优于预期的 k^6；不过，由于等距映射的维度很大，进一步验证受到限制（$k+1$）$^4 \times (k+1)^2$, , 其维度超出了*Qiskit*的分解能力。

图6中SeqRLSP 和 TreeRLSP 的一个显著特点是深度和门总数呈现阶梯结构。这是因为等距映射会填充至不小于其维度的最小二次幂（参见第三节）。因此，每当 $\chi = k+1$ 超过前一个二次幂时，电路深度和门数就会突然跃升，在 $k = \{2, 4, 8, \dots\}$处形成阶梯。

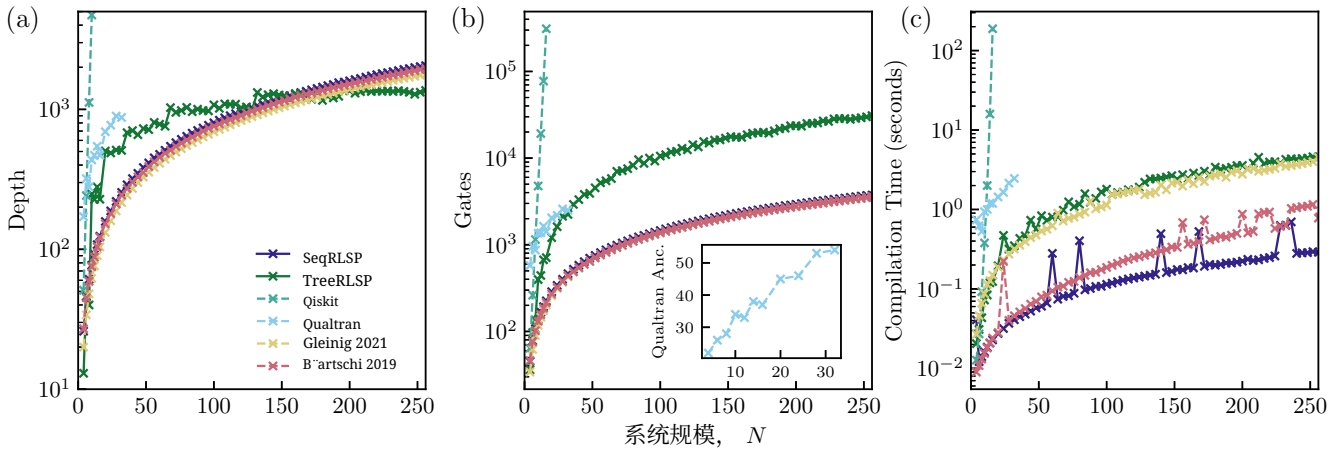


图7：不同系统规模下 W态的制备成本 N。我们比较 SeqRLSP、TreeRLSP、*Qiskit*、*Qualtran*、Gleinig 和 Hoefler [47], 以及 B¨artschi 和 Eidenbenz [51]。（a）量子电路深度。（b）门总数。插图显示了 *Qualtran* 使用的辅助量子比特数量。（c）编译时间（秒）。

B. W态

W态是 Dicke态的一种特殊情况，其中 $k = 1$ [25]。对于规模为 N 的系统，它是所有 N 独热态的均匀叠加。在图7中，我们报告了不同系统规模 N 下的 W态制备数值结果，最大达到 $N = 256$。我们将 SeqRLSP 和 TreeRLSP 与 Qiskit [41], 、Qualtran [45, 46], 、Gleinig 和 Hoefler [47],以及 B¨artschi 和 Eidenbenz [51]进行比较。W态的低稀疏度使 Gleinig 和 Hoefler [47] 能够处理比图5中更大的 N。在图7中，我们观察到 SeqRLSP、Gleinig 和 Hoefler [47], 以及 B¨artschi 和 Eidenbenz [51], 的电路深度和门总数非常相近，而 SeqRLSP 始终实现最低的编译时间，尽管所有方法都能在几秒内完成。对于 TreeRLSP，其 $O(\log N)$ 深度扩展最终会在约 $N \approx 150$ 时超过 SeqRLSP，这凸显了它在超大系统规模下的优势。

C. 补态

如前所述，补RLS尤其适用于我们的框架，因为它们可以高效地表示为 RLS。特别是，如果一个量子态 $|\Psi\rangle$ 可以表示为键维数为 χ 的矩阵乘积态，$|\Psi_c\rangle = |+\rangle^{\otimes N} - |\Psi\rangle$ 则其补态 $\chi + 1$ 可以表示为键维数至多为 定理 2（见）。据我们所知，目前没有专门的编译器能够高效生成那些原本容易制备的量子态的补态。在图8中，我们报告了制备$k = 2$ 的 Dicke态补态的数值结果。我们将 SeqRLSP 和 TreeRLSP 与 *Qiskit* [41], *Qualtran* [45, 46], 以及 Gleinig 和 Hoefler [47]进行比较。由于 Dicke-2 态的补态并非稀疏态，因此 *Qualtran* 和 Gleinig 和 Hoefler [47] 等稀疏态制备方法在系统规模增大时， N 量子资源和编译时间都将呈指数级增长，因此仅适用于小规模系统 N。相比之下，SeqRLSP 和 TreeRLSP 仍然高效，并且是唯一能够处理 $N > 20$ 此任务的方法。

D. 随机均匀叠加

制备字符串的随机均匀叠加，正是稀疏态制备算法 [45, 47] 所针对的任务。相比之下，我们的 SeqRLSP 与 TreeRLSP 方法经过优化，旨在利用底层结构进行压缩，而这一优势在随机态中并不存在。因此，本小节考察我们算法的最坏情况性能。具体而言，在缺乏结构的情况下，表示 s个随机比特串的叠加需要键维数为 $\chi = s$ 的矩阵乘积态。

在图 9中，我们报告了为系统规模 $N = 10$ 和 $N = 20$制备随机叠加态的成本。正如预期，最大的 MPS 键维数通常随 $\chi = s$增长。对于 $N = 10$，在 $s \approx 32$之后会出现一个平台。这种行为源于：对于具有 N 个格点的 MPS，由于 MPS 的阶梯式结构，最大键维数受 $\chi = \min(2^{\lfloor N/2 \rfloor}, s)$限制。当 s时，电路深度和门数会出现阶梯式增长。

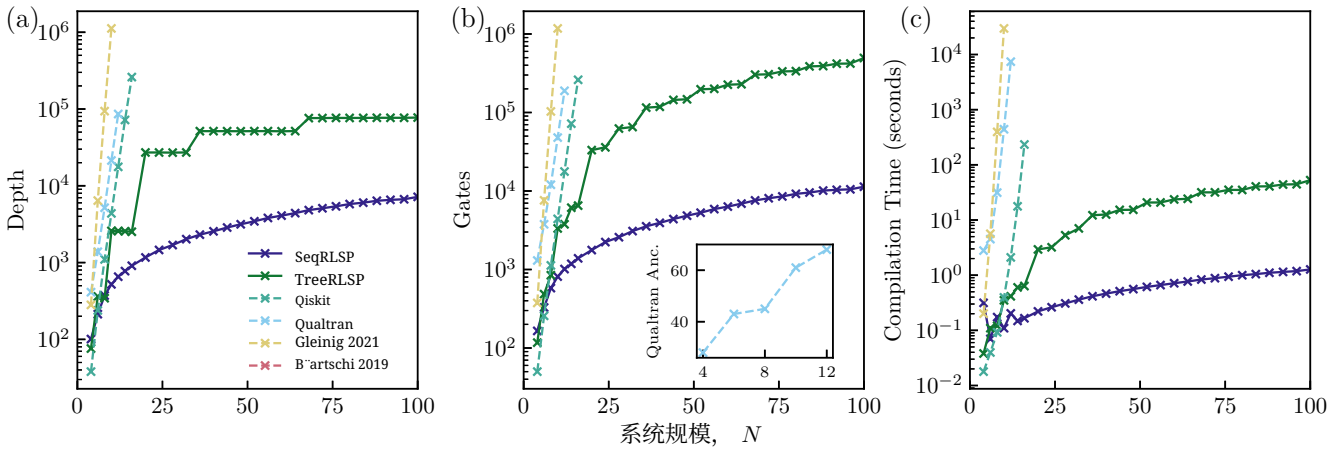


图8：不同系统规模下 Dicke态补态的制备成本−2 。 N我们比较 SeqRLSP、TreeRLSP、*Qiskit*、*Qualtran*、Gleinig和Hoefler [47], 以及 B¨artschi和Eidenbenz [51]。(a) 量子电路深度。(b) 门总数。插图显示了*Qualtran*使用的辅助量子比特数。(c) 编译时间（秒）。

越过下一个二的幂次，这一点对于 $N = 20$ 尤为明显。

值得注意的是，对于 $N = 10$ 和 $s \approx 1024$，当接近所有比特串的均匀叠加（即乘积态）时，SeqRLSP 会识别出其底层（补态）结构，使深度和门数骤然下降。图 9图 9 (a) 中的插图进一步表明，在非常大的 s下，阶梯状结构会再次出现，这表明我们的方法能够有效检测何时可以压缩所得量子态，即将其表示为（$N - s$）个随机字符串的补态。相比之下，稀疏态方法无法利用这一结构：*Qualtran* 无法识别（$N = 10,\ s = 1024$）乘积态，而 Gleinig和 Hoefler [47] 在运行时间超过10^5 秒后仍未收敛。

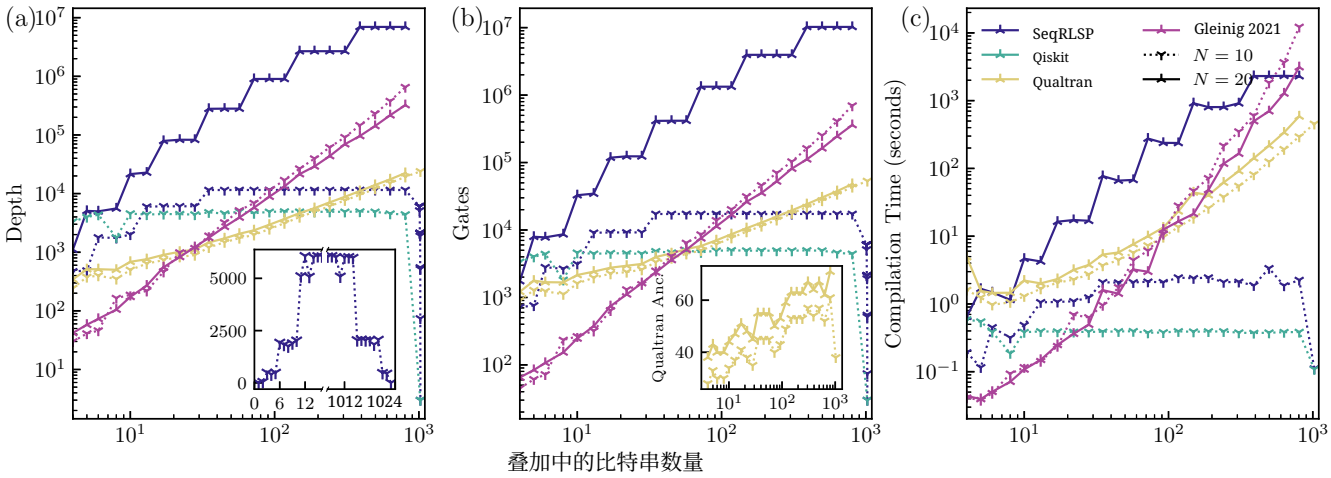


图 9：由 s 个随机比特串的均匀叠加构成的 $N = 10$ 和 $N = 20$ 量子比特量子态的制备成本。我们比较 SeqRLSP、*Qiskit*、*Qualtran* 和 Gleinig 和 Hoefler [47]。(a) 量子电路深度。插图展示了 SeqRLSP 在较小和较大 s 时的表现。(b) 门总数。插图展示了 *Qualtran* 使用的辅助量子比特数量。(c) 编译时间（秒）。

VI. 结论

我们开发了一种量子电路编译器流水线，介于通用但不考虑结构的量子态制备例程与定制的、针对特定族的构造之间。我们的目标是 RLS 及其补集——均匀叠加态，

其中包含给定正则语言所接受的字符串的位置，或在补集情形下被拒绝的字符串的位置。该流水线将基于正则语言的前端与 DFA/MPS 中间表示相结合，使用户能够通过正则表达式、DFA 或有限字符串集来指定目标态，而无需预先识别某个命名的量子态族。

基于这一中间表示，我们提出了两种硬件感知后端：SeqRLSP 和 TreeRLSP。SeqRLSP 面向线性最近邻硬件，在系统规模上实现线性深度；TreeRLSP 则利用全对全连通性，将深度降至系统规模的对数级。在这两种方案中，DFA/MPS 中间表示都至关重要：通过最小化确定有限自动机，并将矩阵乘积态转换为等距网络，编译器便能在存在可压缩性时检测并加以利用。确定有限自动机中间表示为从用户级描述到矩阵乘积态构造提供了一条快速高效的路径，弥合了与矩阵乘积态综合策略之间的差距；后者通常假定已有一个良好的矩阵乘积态。

我们的数值实验验证了理论资源界，并表明该编译器在制备迪克态时能够接近专用方法的性能，同时还能在通用稀疏态制备例程无法处理的情形下，从非结构化描述中识别并利用结构。此外，实验还展现了补集描述和高效制备的优势，这是我们方法一项全新的特性。

我们方法的一项关键设计优势是模块化。DFA/MPS 中间表示将面向用户的规格说明与后端实现清晰分离，因此可以集成正则表达式转确定有限自动机工具、确定有限自动机最小化、矩阵乘积态综合或等距分解方面的改进，而无需更改前端。未来的工作包括将我们的方法从均匀 RLS 扩展到具有任意复值振幅的状态，并探索更先进的矩阵乘积态到量子电路的映射——例如测量与反馈方案，或更直接地合成单量子比特和双量子比特门序列——以进一步改善其相对于 χ 的渐近尺度，并更贴近专门构造。

致谢

这项工作部分是在 MPQ 理论黑客松期间完成的。作者感谢 Khurshed P. Fitter 在研究初期进行的讨论。A.B. 感谢 Davide Conficconi 提供宝贵的反馈并参与富有帮助的讨论。MPQ 的工作获得德国联邦教育与研究部（BMBF）通过资助项目 ALMANAQC（项目编号 13N17236）的支持，该项目隶属于“量子系统”研究计划。此项研究还部分获得 T HEQUCO 的资助，作为慕尼黑量子谷项目的一部分；慕尼黑量子谷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。

[1] P.W. Shor。量子计算算法：离散对数与因数分解。载于 *Proceedings 35th Annual Symposium on Foundations of Computer Science* 124 134, 1994 10 1109 1994 365700 ，页码 – 。doi: 。/SFCS. 。[2] Sean Hallgren。用于佩尔方程和主理想问题的多项式时间量子算法。 *J. ACM*，54(1)，2007年3月。国际标准连续出版物号 0004-5411。doi:10.1145/1206035.1206039。[3] Seth Lloyd。通用量子模拟器。 *Science*，273(5278):1073–1078，1996年。doi:10.1126/science.273.5278.1073。URL https://www.science.org/doi/abs/10.1126/science.273.5278.1073。[4] Rolando D. Somma、Robbie King、Robin Kothari、Thomas E. O’Brien 和 Ryan Babbush。影子哈密顿量模拟。*Nature Communications*，16(1):2690，2025年。doi:10.1038/s41467-025-57451-z。[5] Aram W. Harrow、Avinatan Hassidim 和 Seth Lloyd。求解线性方程组的量子算法。 *Phys. Rev.* , : 10月。doi:。/*Lett.* 103 150502, 2009 10 1103 103 150502 PhysRevLett. 。[6] Shantanav Chakraborty、Aditya Morolia 和 Anurudh Peduri。量子正则最小二乘。 *Quantum*，7:988，2023年4月。国际标准连续出版物号 2521-327X。doi:10.22331/q-2023-04-27-988。[7] Yanlin Chen 和 Ronald de Wolf。带范数约束的线性回归量子算法与下界。*Leibniz Int. Proc. Inf.*，261:38:1–38:21，2023年。doi:10.4230/LIPIcs.ICALP.2023.38。[8] Armando Bellante 和 Stefano Zanero。量子匹配追踪：一种用于稀疏表示的量子算法。*Phys. Rev. A*，105:022414，2022年2月。doi:10.1103/PhysRevA.105.022414。[9] Amira Abbas、Andris Ambainis、Brandon Augustino、Andreas B¨artschi、Harry Buhrman、Carleton Coffrin、Giorgio Cor- tiana、Vedran Dunjko、Daniel J. Egger、Bruce G. Elmegreen 等。量子优化中的挑战与机遇。*Nature Reviews Physics*，6(12):718–735，2024年10月。国际标准连续出版物号 2522-5820。doi:10.1038/s42254-024-00770-9。[10] Stephen P. Jordan、Noah Shutty、Mary Wootters、Adam Zalcman、Alexander Schmidhuber、Robbie King、Sergei V. Isakov、Tanuj Khattar 和 Ryan Babbush。基于解码量子干涉测量的优化。 *Nature*，646(8086):831–836，2025年。doi:10.1038/s41586-025-09527-5。[11] Ashley Montanaro。蒙特卡罗方法的量子加速。 *Proceedings of the Royal Society A: Mathematical Physical* , , ( ): 9月。国际 *and Engineering Sciences* 471 2181 20150301, 2015 1471 2946 10 1098 2015 0301 标准连续出版物号 -。doi:。/rspa. 。[12] Arjan Cornelissen、Yassine Hamoudi 和 Sofiene Jerbi。用于多变量均值估计的近最优量子算法。载于 *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*，STOC ’22，页码 33–43。ACM，2022年6月。doi:10.1145/3519935.3520045。

[13] Patrick Rebentrost、Masoud Mohseni 和 Seth Lloyd。用于大数据分类的量子支持向量机。 *Phys.* , : 9月。doi: 。/ *Rev. Lett.* 113 130503, 2014 10 1103 113 130503 PhysRevLett. 。 。[14] Iordanis Kerenidis 和 Anupam Prakash。量子推荐系统。载于 Christos H. Papadimitriou 编, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)* 67 *Leibniz International Proceedings in* , 第 卷, 页码: –: 德国达格施图尔, 。达格施图尔·莱布尼茨信息学中心。国际标准书号 978-3-95977-029-3。doi:10.4230/LIPIcs.ITCS.2017.49。[15] Jacob Biamonte、Peter Wittek、Nicola Pancotti、Patrick Rebentrost、Nathan Wiebe 和 Seth Lloyd。量子机器学习。 *Nature*, 549(7671):195–202, 2017年9月。国际标准连续出版物号 1476-4687。doi:10.1038/nature23474。[16] Yuxuan Du、Xinbiao Wang、Naixu Guo、Zhan Yu、Yang Qian、Kaining Zhang、Min-Hsiu Hsieh、Patrick Rebentrost 和 Dacheng Tao。量子机器学习: 面向机器学习从业者和研究人员的实践教程。 *arXiv preprint arxiv:2502.01146* 2025 10 48550 2502 01146 , 。doi: 。/arXiv. 。 。

[17] Laura Lewis、Dar Gilboa 和 Jarrod R. McClean。利用自然数据分布学习浅层神经网络的量子优势。 *arxiv preprint arxiv:2503.20879*, 2025年。doi:10.48550/arXiv.2503.20879。[18] Dolev Bluvstein、Simon J. Evered、Alexandra A. Geim、Sophie H. Li、Hengyun Zhou、Tom Manovitz、Sepehr Ebadi、Madelyn Cain、Marcin Kalinowski、Dominik Hangleiter、J. Pablo Bonilla Ataides、Nishad Maskara、Iris Cong、Xun Gao、Pedro Sales Rodriguez、Thomas Karolyshyn、Giulia Semeghini、Michael J. Gullans、Markus Greiner、Vladan Vuleti´c 和 Mikhail D. Lukin。基于可重构原子阵列的逻辑量子处理器。 *Nature*, 626(7997):58–65, 2023年12月。国际标准连续出版物号 1476-4687。doi:10.1038/s41586-023-06927-3。[19] Rajeev Acharya、Dmitry A. Abanin、Laleh Aghababaie-Beni、Igor Aleiner、Trond I. Andersen、Markus Ansmann、Frank Arute、Kunal Arya、Abraham Asfaw、Nikita Astrakhantsev 等。低于表面码阈值的量子纠错。 *Nature*, 638(8052):920–926, 2025年。doi:10.1038/s41586-024-08449-y。[20] Gilles Brassard、Peter Høyer、Michele Mosca 和 Alain Tapp。量子振幅放大与估计。*Computation and Information* 53 74, 2002 0271 4132 10 1090 305 05215 *Quantum* , 页码 –。国际标准连续出版物号 -。doi: 。/conm/ / 。[21] Iordanis Kerenidis、Jonas Landman、Alessandro Luongo 和 Anupam Prakash。 *q-means: a quantum algorithm for unsupervised machine learning* 2019 10 48550 1812 03584 。Curran Associates Inc., 美国纽约州雷德胡克, 。doi: 。/arXiv. 。 。[22] Armando Bellante、Alessandro Luongo 和 Stefano Zanero。用于数据表示与分析的基于奇异值分解的量子算法。 *Quantum Machine Intelligence*, 4(2):20, 2022。doi:10.1007/s42484-022-00076-y。[23] Iordanis Kerenidis 和 Alessandro Luongo。利用量子慢特征分析对 MNIST 数据集进行分类。 *Phys.* , : 6月。doi: 。/ *Rev. A* 101 062327, 2020 10 1103 101 062327 PhysRevA. 。 。[24] Daniel M. Greenberger、Michael A. Horne 和 Anton Zeilinger。 *Going Beyond Bell's Theorem*, 第69–72页。施普林格·荷兰, 多德雷赫特, 1989年。国际标准书号 978-94-017-0849-4。doi:10.1007/978-94-017-0849-4˙10。[25] W. D¨ur、G. Vidal 和 J. I. Cirac。三个量子比特可以以两种互不等价的方式纠缠。 *Phys. Rev. A*, 62:062314, 2000年11月。doi:10.1103/PhysRevA.62.062314。[26] R. H. Dicke。自发辐射过程中的相干性。 *Phys. Rev.*, 93:99–110, 1954年1月。doi:10.1103/PhysRev.93.99。[27] Marta Florido-Llin`as, ´Alvaro M Alhambra、David P´erez-Garc´ıa 和 J Ignacio Cirac。正则语言量子态。*arXiv preprint arXiv:2407.17641*, 2024。doi:10.48550/arXiv.2407.17641。[28] Gregory M. Crosswhite 和 Dave Bacon。用于矩阵乘积算法缓存的有限自动机。 *Phys. Rev. A*, 78: 012356, 2008年7月。doi:10.1103/PhysRevA.78.012356。[29] Rom´an Or´us。张量网络实用导论: 矩阵乘积态与投影纠缠对态。*Annals of Physics*, 349:117–158, 2014。国际标准连续出版物号 0003-4916。doi:10.1016/j.aop.2014.06.013。[30] C. Sch¨on、E. Solano、F. Verstraete、J. I. Cirac 和 M. M. Wolf。纠缠多量子比特态的顺序生成。 *Phys. Rev. Lett.*, 95:110503, 2005年9月。doi:10.1103/PhysRevLett.95.110503。[31] J Ignacio Cirac 和 Frank Verstraete。自旋链与晶格中的重整化和张量积态。*Physics A: Mathematical and Theoretical* 42 50 504004, 2009 10 1088 1751 8113 42 50 504004 *Journal of* , ( ): 12月。doi: 。/ - / / / 。[32] Daniel Malz、Georgios Styliaris、Zhi-Yuan Wei 和 J. Ignacio Cirac。利用对数深度量子电路制备矩阵乘积态。 *Phys. Rev. Lett.*, 132:040404, 2024年1月。doi:10.1103/PhysRevLett.132.040404。[33] Lorenzo Piroli、Georgios Styliaris 和 J. Ignacio Cirac。利用量子电路和测量近似多体量子态。 *Phys. Rev. Lett.*, 133:230401, 2024年12月。doi:10.1103/PhysRevLett.133.230401。[34] Ramis Movassagh 和 Peter W. Shor。局域系统中的超临界纠缠: 量子物质面积律的反例。*Proceedings of the National Academy of Sciences*, 113(47):13278–13282, 2016年。doi:10.1073/pnas.1605716113。[35] Sarang Gopalakrishnan。作为高度纠缠态序贯生成器的下推自动机。 *Journal of Physics A: Mathematical and Theoretical* 58 5 055301, 2025 10 1088 1751 8121 9 , ( ): 。doi: 。/ - /adab f。

[36] Michael A. Nielsen 和 Isaac L. Chuang。 *Quantum Computation and Quantum Information*。剑桥大学出版社, 2012年6月。国际标准书号 978-0-521-63503-5。doi:10.1017/cbo9780511976667。[37] J. Ignacio Cirac、David P´erez-Garc´ıa、Norbert Schuch 和 Frank Verstraete。矩阵乘积态与投影纠缠对态: 概念、对称性与定理。 *Rev. Mod. Phys.*, 93:045003, 2021年12月。doi:10.1103/RevModPhys.93.045003。[38] John E. Hopcroft、Rajeev Motwani 和 Jeffrey D. Ullman。《自动机理论、语言与计算导论》, 第2版。 *SIGACT News*, 32(1):60–65, 2001年3月。国际标准连续出版物号 0163-5700。doi:10.1145/568438.568455。[39] A Yu Kitaev。量子计算: 算法与纠错。 *Russian Mathematical Surveys*, 52(6):1191, 1997年12月。doi:10.1070/RM1997v052n06ABEH002155。[40] Christopher M. Dawson 和 Michael A. Nielsen。Solovay-Kitaev算法。 *Quant. Inf. Comput.*, 6:081–095, 2006年。doi:10.26421/QIC6.1-6。

[41] Ali Javadi-Abhari、Matthew Treinish、Kevin Krsulich、Christopher J. Wood、Jake Lishman、Julien Gacon、Simon Martiel、Paul D. Nation、Lev S. Bishop、Andrew W. Cross、Blake R. Johnson 和 Jay M. Gambetta。使用 Qiskit 进行量子计算。 *arXiv preprint arXiv:2405.08810*，2024年。doi:10.48550/arXiv.2405.08810。[42] Raban Iten、Roger Colbeck、Ivan Kukuljan、Jonathan Home 和 Matthias Christandl。用于等距映射的量子电路。*Phys. Rev. A*，93:032318，2016年3月。doi:10.1103/PhysRevA.93.032318。[43] Dominic W. Berry、Craig Gidney、Mario Motta、Jarrod R. McClean 和 Ryan Babbush。利用稀疏性和低秩分解对任意基底量子化学进行量子比特化。 *Quantum*，3:208，2019年12月。国际标准连续出版物号 2521-327X。doi:10.22331/q-2019-12-02-208。[44] Ryan Babbush、Craig Gidney、Dominic W. Berry、Nathan Wiebe、Jarrod McClean、Alexandru Paler、Austin Fowler 和 Hartmut Neven。以线性 t 复杂度在量子电路中编码电子能谱。 *Physical Review X*，8(4)，2018年10月。国际标准连续出版物号 2160-3308。doi:10.1103/physrevx.8.041015。[45] Matthew P. Harrigan、Tanuj Khattar、Charles Yuan、Anurudh Peduri、Noureldin Yosri、Fionn D. Malone、Ryan Babbush 和 Nicholas C. Rubin。使用 Qualtran 表达和分析量子算法。 *arXiv preprint arXiv:2409.04643*，2024 2409 04643。网址：https://arxiv.org/abs/。。[46] Martin Plesch 和 ˇ Caslav Brukner。利用通用门分解进行量子态制备。 *Phys. Rev. A*，83: 032302，2011年3月。doi:10.1103/PhysRevA.83.032302。[47] Niels Gleinig 和 Torsten Hoefler。一种高效的稀疏量子态制备算法。载于 *2021 58th ACM/IEEE Design Automation Conference (DAC)* 433 438 2021 10 1109 18074 2021 9586240 ，页码 –。IEEE，。doi:。/DAC 。。。[48] Jingquan Luo 和 Lvzhou Li。稀疏量子态制备的电路复杂度。 *arXiv preprint arXiv:2406.16142v1*，2024 2406 16142 1。网址：https://arxiv.org/abs/。v。[49] Tiago M. L. de Veras、Leon D. da Silva 和 Adenilton J. da Silva。双重稀疏量子态制备。 *Quantum Information Processing* 21 6 204, 2022 10 1007 11128 022 03549 ，( ):。doi:。/s - - -y。[50] Renato M.S. Farias、Thiago O. Maciel、Giancarlo Camilo、Ruge Lin、Sergi Ramos-Calderer 和 Leandro Aolita。固定汉明重量子空间的量子编码器。*Phys. Rev. Appl.*，23:044014，2025年4月。doi: 10.1103/PhysRevApplied.23.044014。[51] Andreas B¨artschi 和 Stephan Eidenbenz。 *Deterministic Preparation of Dicke States*，第126–139页。Springer国际出版公司，2019年。国际标准书号 9783030250270。doi:10.1007/978-3-030-25027-0˙9。[52] Andreas B¨artschi 和 Stephan Eidenbenz。用于 Dicke态制备的浅层电路。载于 *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* 87 96 2022 ，第 – 页。IEEE，9月 。doi: 10.1109/qce53715.2022.00027。[53] S. C. Kleene。 *Representation of Events in Nerve Nets and Finite Automata*，第3–42页。普林斯顿大学出版社，普林斯顿，1956年。国际标准书号 9781400882618。doi:10.1515/9781400882618-002。[54] Ulrich Schollw¨ock。矩阵乘积态时代的密度矩阵重整化群。 *Annals of Physics*，326 (1):96–192，2011年。国际标准连续出版物号 0003-4916。doi:10.1016/j.aop.2010.09.012。2011年1月特刊。[55] Xie Chen、Zheng-Cheng Gu 和 Xiao-Gang Wen。一维自旋系统中有能隙对称相的分类。 *Phys. Rev. B*，83:035107，2011年1月。doi:10.1103/PhysRevB.83.035107。[56] Norbert Schuch、David P´erez-Garc´ıa 和 Ignacio Cirac。利用矩阵乘积态和投影纠缠对态对量子相进行分类。 *Phys. Rev. B*，84:165139，2011年10月。doi:10.1103/PhysRevB.84.165139。[57] I. V. Oseledets。张量列车分解。*SIAM Journal on Scientific Computing*，33(5):2295–2317，2011年。doi: 10.1137/090752286。[58] Ken Thompson。编程技术：正则表达式搜索算法。 *Commun. ACM*，11(6):419–422，1968年6月。国际标准连续出版物号 0001-0782。doi:10.1145/363347.363387。[59] V M Glushkov。自动机抽象理论。*Russian Mathematical Surveys*，16(5):1，1961年10月。doi: 10.1070/RM1961v016n05ABEH004112。[60] M. O. Rabin 和 D. Scott。有限自动机及其判定问题。 *IBM Journal of Research and Development*，3 (2):114–125，1959年。doi:10.1147/rd.32.0114。[61] V Aho Alfred、S Lam Monica 和 D Ullman Jeffrey。 *Compilers principles, techniques & tools*。培生教育，2007年。[62] John Hopcroft。一种用于有限自动机状态最小化的 n log n 算法。载于 Zvi Kohavi 和 Azaria Paz 编，*Theory of Machines and Computations*，第189–196页。学术出版社，1971年。国际标准书号 978-0-12-417750-5。doi:10.1016/B978- 0-12-417750-5.50022-1。[63] Dominique Revuz。在线性时间内最小化无环确定有限自动机。 *Theoretical Computer Science*，92(1): 181–189，1992年。国际标准连续出版物号 0304-3975。doi:10.1016/0304-3975(92)90142-3。[64] Armando Bellante、Stefano Vanerio 和 Stefano Zanero。量子稀疏恢复与量子正交匹配追踪。 *arXiv preprint arXiv:2510.06925*，2025年。网址：https://arxiv.org/abs/2510.06925。[65] Marcello Benedetti、Harry Buhrman 和 Jordi Weggemans。补集采样：在样本复杂度方面具备可证明、可验证且适用于含噪中型量子的量子优势。 *arXiv preprint arXiv:2502.08721*，2025年。网址：https://arxiv.org/abs/2502. 08721。[66] Y.-Y. Shi、L.-M. Duan 和 G. Vidal。利用树张量网络对量子多体系统进行经典模拟。*Phys. Rev. A*，74:022320，2006年8月。doi:10.1103/PhysRevA.74.022320。[67] Raban Iten、Oliver Reardon-Smith、Emanuel Malvetti、Luca Mondada、Gabrielle Pauvert、Ethan Redmond、Ravjot Singh Kohli 和 Roger Colbeck。universalqcompiler 简介。 *arXiv preprint arXiv:2502.08721*，2021 年。网址：https: // arxiv.org/abs/1904.01072。[68] Minzhao Liu、Ruslan Shaydulin、Pradeep Niroula、Matthew DeCross、Shih-Han Hung、Wen Yu Kon、Enrique Cervero- Mart´ín、Kaushik Chakraborty、Omar Amer、Scott Aaronson、Atithi Acharya、Yuri Alexeev、K. Jordan Berg、Shouvanik

Chakrabarti、Florian J. Curchod、Joan M. Dreiling、Neal Erickson、Cameron Foltz、Michael Foss-Feig、David Hayes、Travis S. Humble、Niraj Kumar、Jeffrey Larson、Danylo Lykov、Michael Mills、Steven A. Moses、Brian Neyenhuis、Shaltiel Eloul、Peter Siegfried、James Walker、Charles Lim 和 Marco Pistoia。使用囚禁离子量子处理器实现认证随机性。 *Nature*，640(8058):343–348，2025年3月。国际标准连续出版物号 1476-4687。doi:10.1038/s41586-025-08737-1。[69] Julien Romero。Pyformlang：用于形式语言处理的教学库，载于 *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* 21，576 582，2021，SIGCSE ' 页码 – 纽约，纽约州，美国，。计算机协会。国际标准书号 9781450380621。doi:10.1145/3408877.3432464。[70] Charles R. Harris、K. Jarrod Millman、St´efan J. van der Walt、Ralf Gommers、Pauli Virtanen、David Cournapeau、Eric Wieser、Julian Taylor、Sebastian Berg、Nathaniel J. Smith、Robert Kern、Matti Picus、Stephan Hoyer、Marten H. van Kerkwijk、Matthew Brett、Allan Haldane、Jaime Fern´andez del R´ıo、Mark Wiebe、Pearu Peterson、Pierre G´erard-Marchant、Kevin Sheppard、Tyler Reddy、Warren Weckesser、Hameer Abbasi、Christoph Gohlke 和 Travis E. Oliphant。使用 NumPy 进行数组编程。 *Nature*，585(7825):357–362，2020年9月。doi:10.1038/s41586-020-2649-2。