关于量子系统与量子平台的思考

南昌大学
20 级人工智能实验班
陈科
ncukechen@email.ncu.edu.cn

前言

老师，您好！以下是我关于量子系统与量子平台的相关思考。其中包括了：量子系统与量子平台的调研、量子系统与量子平台的开发经历、量子系统与量子平台的心得和想法。

关于量子系统与量子平台，我了解了一些优秀的例子，其中主要学习经典量子模拟、量子编译方面；除此之外，我还了解了一些量子算法。

量子系统与量子平台调研

我将讲述一些背景调研和学习内容，关于量子系统与量子平台方面。首先是现有一些优秀的例子。其次我将分别深入介绍量子模拟器、量子编译器以及一些量子算法。

1. 现有量子系统与量子平台调研

关于量子系统与量子平台方面的调研工作，我主要是关注目前开源的优秀量子系统与量子云平台。

我曾深入学习并使用过以下几种量子系统与量子平台：

- IBM Quantum(Qiskit[1]);
- 本源量子 [2];
- 华为量子平台（MindQuantum[3]）。

此外，我还了解了其他一类似上述的量子系统与量子平台，还包括：

- 微软的 Azure Quantum[4];

- 谷歌的 Quantum AI（Cirq + TensorFlow Quantum）[5];
- 腾讯的 TensorCircuit[6]：对张量网络进行优化，支持自动微分、即时编译、硬件加速和矢量化并行;
- 亚马逊的 Amazon Braket[7];
- 最近由 NVIDIA 与 Quantum Machines 合作推出的 DGX Quantum[8](CUDA Quantum 和 OPX+，未开源)，关键是 NVIDIA CUDA Quantum（已开源）[9]。

这里的量子系统与量子平台的定义，是指不仅仅支持量子电路的编译、模拟，还支持上层量子算法的封装或支持量子云平台的提供。

除此之外，主要参考 [10]，我还了解到一些优秀的量子计算软件包，支持量子电路的运行（感觉可以理解为量子模拟器，封装程度相对较小），如下：

- Simon Benjamin 和牛津大学量子研究中心开发的 QUEST[11]：通过 C++ 实现，致力于高性能、高效率、全振幅;
- 滑铁卢大学量子计算研究所的 Yao.ji[12]：通过 julia 实现，可扩展性和高效性以及;
- NVIDIA 的 cuQuantum SDK[13]：通过 CUDA 实现，Python 调用，对 GPU 进行较多优化，具备高效性。
- 英特尔的 Intel quantum simulator：通过 C++ 实现，Python 调用，向后兼容，优化多核并行;
- 滑铁卢大学量子计算研究所的 Quantum++[14]：通过 C++ 实现，Python 调用，非常低的外部依赖，考虑易用性、高便携性和高性能。
- Nathaniel Johnston 的 QETLAB[15]：通过 MATLAB 实现，优化处理满矩阵、大型稀疏矩阵并利用半定规划技术。
- PyZX[16]：实现了 ZX-calculus 方法。

2. 现有量子系统与量子平台深入分析

我曾尝试配置并使用过 IBM Quantum(Qiskit)、国内本源量子的云平台(Qpanda-2）和华为量子云平台（MindQuantum），因此对它们有一定的了解。在本段中，

我将根据这三个量子系统与量子平台以及中科大量子系统与量子平台的相关资料，对它们进行比较和分析。

以下存在上述四个量子系统与量子平台的基本架构图, 根据以下架构图, 我大概把量子系统与量子平台的开发总结为三个层次:

- 底层计算层：经典硬件上的**量子模拟器** (本源量子称为量子虚拟机 [图3], 在 CPU、GPU、超算硬件架构上运行)；**量子处理器芯片**控制接口的有效实现（超导、光子、离子阱)。

- 系统架构层：**量子编译器的实现**（广义上，可包含量子电路编译、量子程序验证和分析、量子编程语言)；**量子操作系统**（这个词没有明确定义，在这里我所表达的含义是能够对任务、资源进行抽象，并且对抽象的任务与资源进行管理和调度)。

  对于**量子处理器芯片**，考虑到**噪声**的影响：需要实现**自动化的校准**；**基于硬件的量子电路最优编译**（较少的量子门和较小的量子电路深度，从而提高在量子处理器芯片上运行的准确性)；**量子纠错**（通过额外的量子比特对部分量子比特进行纠错)。

  考虑**经典与量子的混合架构**，经典与量子的通信（目前了解到变分算法、测量操作需要较多的经典和量子通信)，这里我没有想清楚，存在以下几个问题：什么情况下需要进行经典和量子之间的通信? 如何进行经典与量子计算之间的通信?

- 用户应用层：**量子通用算法**的实现与研究（Shor、Grover、HHL 等)；**量子变分算法**的实现和研究：(1) 针对特点目标（机器学习、量子优化、化学模拟、分子动力学模拟、量子金融 [图1])，如何搭建一个量子电路? (2) 如何实现更加快速、鲁棒的训练过程? **量子云平台**的实现（通过实现量子的云服务，从而使得更多的人能够使用)。

目前来看，三层之间是紧密结合的。比如：(1) 对于变分量子算法的优化，不仅仅是在变分电路搭建层面，还与量子编译有关。针对量子变分电路，每个训练周期内，电路仅仅更新参数，不改变电路结构，因此可以考虑在变分电路的后续编译过程中，去除冗余的编译过程。(2) 我认为量子系统架构层的实现与底层的技术路线（不同硬件实现：超导、光子、离子阱；不同量子计算模型: 量子电路、量子拓扑、量子图灵机）也关系很大。

过于紧密结合，容易导致开发、优化的难度较高。如何实现分层、实现不同层之间的抽象，实现不同层之间的中间部分？我认为也是一个值得考虑的问题，但我仅仅了解到清华在对硬件与量子模拟算法进行抽象，两层中间，实现量子模拟的优化，只需要写一遍 [17]。

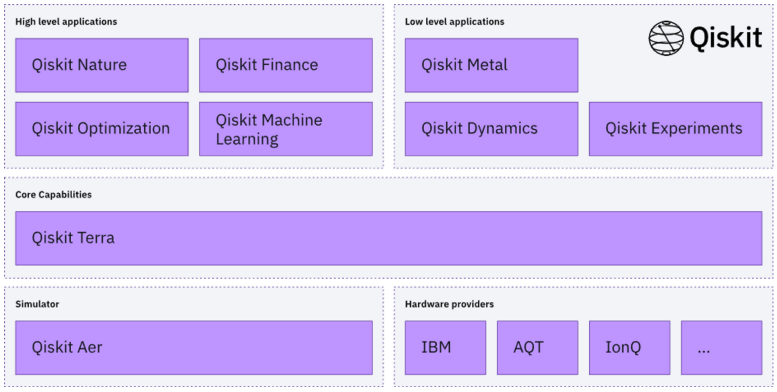

图 1: IBM 的量子系统与量子平台基本架构图（IBM Qiskit）

3. 量子模拟器的调研与分析

由于我没有直接使用过量子处理器芯片的机会，对其工作原理缺乏深入的了解。为此，我主要通过量子模拟器来探索在经典计算架构如何进行量子模拟，并且目前大部分量子计算的学习、研究都是利用理想的量子模拟器或带噪声的量子模拟器。

根据我阅读的关于量子系统、量子平台和量子模拟器的优秀论文 [18][19][20][21][22]，目前量子模拟器的研究主要分为两个方向：

- 设计、实现更加创意、更高效的模拟方法（量子模拟器的模拟方法）；
- 移植到更高性能的硬件架构上，并对其进行特点优化（经典架构新硬件上移植）。

3.1 量子模拟器的模拟方法：

主要是一些物理学家、数学家以及计算机科学家，希望找到一种更加高效的、易于理解、易扩展的量子计算模拟方法。IBM qiskit-aer 量子模拟器，以下部分支持噪声 [23]：

图 2: 华为的量子系统与量子平台基本架构图（MindQuantum）

图 3: 本源量子的量子系统与量子平台基本架构图

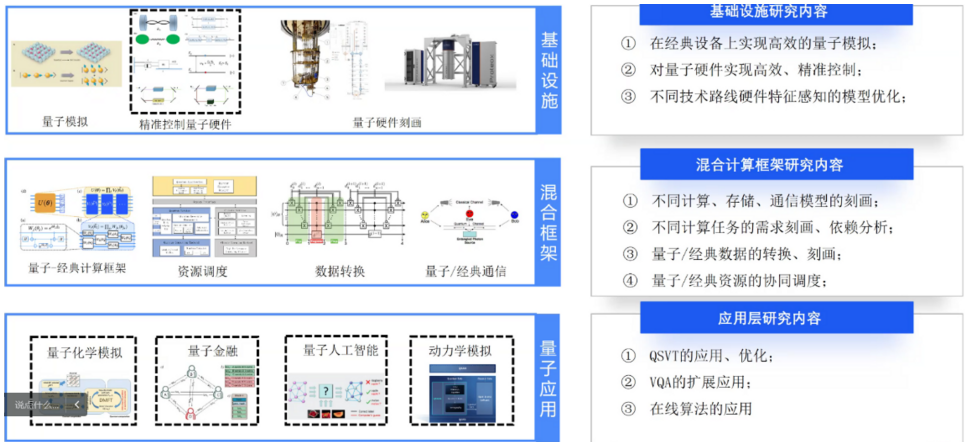

图 4: USTC 计划开发的量子系统与量子平台基本架构图

- 状态向量模拟;
- 密度矩阵模拟;
- 稳定器（Clifford stabilizer）;
- 扩展稳定器（Clifford + T extended stabilizer）;
- 矩阵乘积态（matrix_product_state）;
- 酉矩阵模拟;
- 超级算子矩阵（superop）;
- 张量网络模拟;[24]
- **脉冲模拟** [25];

本源量子 QPanda-2 的量子虚拟机，以下部分支持噪声 [2]:

- 状态向量模拟;
- 密度矩阵模拟;
- 张量网络模拟;

华为 MindQuantum，之前使用苏黎世联邦理工学院开发的 ProjectQ 量子模拟器，近期开发了自己的量子模拟器 [3]:

- 状态向量模拟；

除此之外，还有一些基于图语言的模拟，ZX-calculus[22] 比较流行，ZX-calculus 是一种超越电路图的图形语言 [26]。号称将用于下一代量子软件，因为它不仅仅能实现直观的可视化，还可以用于量子电路的优化 [27]。

目前基于张量网络的大规模、部分振幅模拟，也是针对于大规模量子电路模拟的一项比较流行的技术。之前了解到是张潘老师组在推进中 [28]，是基于张量网络的高效模拟谷歌悬铃木量子电路的方法。

3.2 **经典架构新硬件上移植：**

目前大部分是研究计算机学科的人讨论，目前主要是将其移植到 GPU、超级计算机、FPGA 上。GPU 上移植：

- Qiksit 中，实现了部分方法的 GPU 支持，比如状态向量、密度矩阵、张量网络等 [23]。
- 国内华为 MindQuantum 支持状态向量方的 GPU 版本 [3]。
- NVIDIA 作为 GPU 的研发公司，其组织开发了 CUDA Quantum[9]，号称是第一个 GPU 搭建的量子系统。
- 国内清华翟季东老师组，HyQuas 针对了 GPU 上的量子状态向量进行优化。

超级计算机上的移植（高性能计算集群）：

- 国内刘勇组，神威大规模随机量子电路并行计算模拟。获得 2021 年戈登贝尔奖，通过太湖之光的超算解决悬铃木量子电路的经典模拟问题。
- 本源的高性能计算集群云服务。

关于 FPGA 上移植，我最初是在会议“CCF2022 经典计算机上的量子模拟器：挑战与瓶颈在哪里?”中了解到该想法，还未看到具体的实现，考虑与熟悉 FPGA 的同学一起研究一下。

4. 量子编译器的调研与分析

关于量子编译器，我在一年之前进行了一些简单的学习和调研，主要讨论针对**量子处理器芯片的编译**，具体调研内容如下：针对量子电路在真实的量子模拟器上运行的量子编译器，目前主要分为以下步骤 [29]：

1. 量子门替换（replace），转化为 DAG 图（unroll）。
2. Qubit Mapping，逻辑比特映射到物理比特。
3. Routing（Swap Mapping），添加 SWAP 门，使得 CNOT 门可执行。
4. Optimize，量子电路的优化。

关于 Qubit Mapping，如果量子电路占用比特较少，支持量子比特数较多，还可实现多个量子程序同时映射 [30]。常见 Qubit Mapping 的方法有：

- 随机 Mapping：不进行特殊的 Mapping 处理。
- 利用量子电路的可逆性，两次映射得到较好的量子初始映射。[31]
- 基于图算法，社区检测算法 [32] 寻找合适的映射。
- IBM Qiskit 中支持的 TrivialLayout、DenseLayout、NoiseAdaptiveLayout[33]、Custom Layout。（去年了解，未更新）

关于 Routing，在确定初始映射的情况下，需要考虑添加较少的 SWAP 门，使得 CNOT 门可执行。

- 2022 年之前 Qiskit 中的实现了三种 Routing 算法 BasicSwap、LookaheadSwap、StochasticSwap[29]，基本思想应该都是基于贪心的。
- 基于启发式算法的 Swap 算法（SABRE）[31]。
- 冯元老师，22 年发表的基于蒙特卡洛树搜索（MCTS）[34]，一种强化学习的方法。

关于量子编译的优化，了解不多，目前了解到两篇文章：

- 基于 ZX-Calculus 的全新量子电路优化方法。我们将量子电路解释为 ZX 图，它提供了一种灵活的低级语言来以图形方式描述量子计算 [27]。
- ASPLOS22 的一篇文章 Paulihedral: A Generalized Block-Wise Compiler Optimization Framework for Quantum Simulation Kernels[35]，引入 Pauli 中间表示，实现更加深入的优化。可以参考 [图5]，大致了解 Paulihedral 编译器与传统编译器的区别。

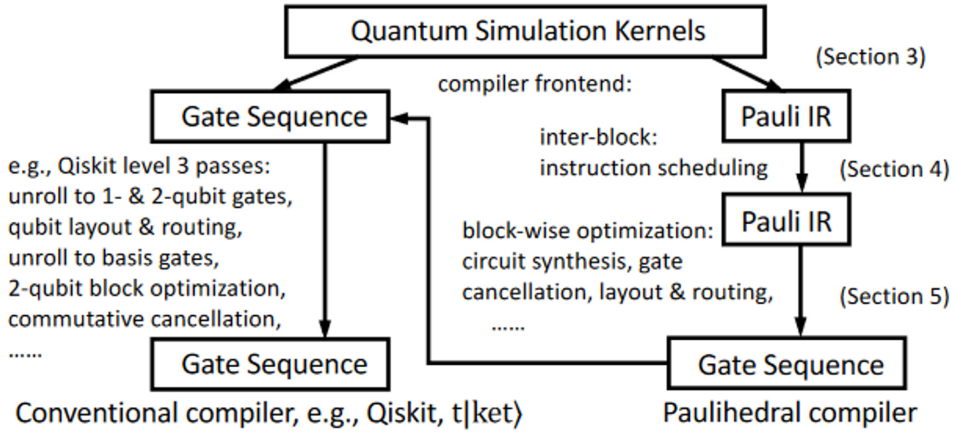

Figure 1: Paulihedral vs conventional compilers

图 5: Paulihedral 编译器与传统编译器的区别 [35]

5. 量子算法的调研

量子算法，目前没有太深入的调研。首先希望把计算层、系统层问题解决，后续再进行算法层的深入学习与开发，目前主要是进行简单使用和学习：

- 在量子系统与平台上搭建量子算法，包括 IBM Qiskit、华为的 MindQuantum、本源量子的 Qpanda-2。
- 量子神经网络的贫瘠高原问题。[36]。
- 如何搭建量子变分算法？通过启发式搜索的方法，针对特定问题，去搜索最佳的量子变分电路 [37]。

量子系统与量子平台的开发经历

我没有系统地参与过量子系统和量子平台的开发，但我有阅读 qiskit-aer 量子模拟器 [23] 的源码和实现量子编译论文 [32] 中一个小模块的经验。

通过阅读 qiskit-aer 源码，我了解了基本的开发流程：

- 项目语言：qiskit-aer 采用 C++ 和 Python 两种语言编写，其中 C++ 负责核心功能的实现，保证高性能，Python 负责接口调用和脚本编写，便于与其他模块交互，二者通过 Pybind11 库进行连接。（qiskit-terra 采用 Rust 和 Python 编写，Rust 具有较好的内存管理机制并且性能与 C++ 相当）
- 项目构建：qiskit-aer 通过 skbuild 控制 CMake 和 pyproject.toml 实现 C++ 和 Python 项目的构建。
- 并行实现：qiskit-aer 通过 OpenMP 实现多线程并行，GPU 的实现需要 CUDA 编程。我在准备超算比赛中学习过 OpenMP 和 CUDA 的并行编程，并且目前在不断地提升我的并行编程能力。

关于一些基础的计算机语言（C++、Python）、版本同步工具 Git（github）、编译环境配置、Linux 等基本的开发能力，我都已经掌握。

我认为我有能力编写其中一个模块，但是目前可能无法独立完成一个层级的项目工作量，因此目前也在阅读别人的代码，学习别人是如何编写一个大型项目的代码结构和风格的。

量子系统与量子平台的心得、想法

在调研和学习的过程中，我深刻地认识到了实现一个量子系统和量子平台的复杂性和挑战性，这需要具备扎实的物理、数学、计算机等方面的知识。

在了解了许多的量子系统和量子平台之后，我有一个疑问，关于中科大建设的量子系统和量子平台：目前市场上已经存在许多的量子系统和量子平台，那么中科大的量子系统和量子平台有什么不同点或者说优势呢?

我非常希望能参与到这个量子系统和量子平台的讨论和开发中，但我可能需要一些时间来学习和了解具体的开发情况（因为我去年只听过一次中科大关于量子系统和量子平台的介绍）。我知道底层设计非常困难，但我对从底层设计入手更感兴趣，这样才能深入理解量子系统和量子平台的工作原理。

例如，我可以参与设计和优化量子模拟器和量子编译器，或者在人手不足的情况下，我也愿意设计量子处理器芯片的控制接口，甚至协助设计量子操作系统。如果在时间有限（学习周期不足）和项目完成期望较高的情况下，我也愿意承担一些非创造性的工作，例如量子算法的封装和性能测量等。

综上所述，这是我在大约一年多的时间里对量子系统和量子平台的积累和调研，以及一些个人的初步想法。

参考文献

[1] Q. contributors, "Qiskit." https://qiskit.org/, 2023.

[2] O. Q. contributors, "Origin quantum." https://originqc.com.cn/, 2023.

[3] M. contributors, "Mindquantum: A new generation of general-purpose quantum computing framework developed based on shengsi mindspore open source deep learning platform." https://gitee.com/mindspore/mindquantum, 2023.

[4] A. contributors, "Azure quantum cloud service: Get innovative quantum hardware, software, and solutions in a single cloud service." https://azure.microsoft.com/en-us/products/quantum, 2023.

[5] G. contributors, "Google quantum ai is advancing the state of the art of quantum computing and developing the tools for researchers to operate beyond classical capabilities.." https://azure.microsoft.com/en-us/products/quantum, 2023.

[6] S.-X. Zhang, J. Allcock, Z.-Q. Wan, S. Liu, J. Sun, H. Yu, X.-H. Yang, J. Qiu, Z. Ye, Y.-Q. Chen, C.-K. Lee, Y.-C. Zheng, S.-K. Jian, H. Yao, C.-Y. Hsieh, and S. Zhang, "TensorCircuit: a Quantum Software Framework for the NISQ Era," Quantum, vol. 7, p. 912, Feb. 2023.

[7] A. contributors, "Amazon braket: A fully managed quantum computing service designed to help speed up scientific research and software development for quantum computing." https://nvidianews.nvidia.com/news/nvidia-announces-new-system-for-accelerated-quantum-classical-computing, 2023.

[8] A. contributors, "Nvidia announces new system for accelerated quantum-classical computing." https://nvidianews.nvidia.com/news/nvidia-announces-new-system-for-accelerated-quantum-classical-computing, 2023.

[9] NVIDIA CUDA Quantum contributors, "Nvidia cuda quantum: The platform for hybrid quantum-classical computing." https://developer.nvidia.com/cuda-quantum, 2023.

[10] Qiskit contributors, "Top 63 quantum computer simulators for 2022." file:///C:/Users/86178/Zotero/storage/AXPY7J7W/top-63-quantum-computer-simulators-for-2022.html, 6 2022.

[11] QuEST contributors, "Quest: a high performance simulator of quantum circuits, state-vectors and density matrices." https://github.com/QuEST-Kit/QuEST, 2023.

[12] X.-Z. Luo, J.-G. Liu, P. Zhang, and L. Wang, "Yao.jl: Extensible, efficient framework for quantum algorithm design," arXiv preprint arXiv:1912.10877, 2019.

[13] NVIDIA cuQuantum contributors, "Nvidia cuquantum sdk is a high-performance library for quantum information science and beyond." https://developer.nvidia.com/cuda-quantum, 2023.

[14] V. Gheorghiu, "Quantum++: A modern c++ quantum computing library," PloS one, vol. 13, no. 12, p. e0208073, 2018.

[15] N. Johnston, "QETLAB: A MATLAB toolbox for quantum entanglement, version 0.9." http://qetlab.com, Jan 2016.

[16] A. Kissinger and J. van de Wetering, "PyZX: Large Scale Automated Diagrammatic Reasoning," in Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019 (B. Coecke and M. Leifer, eds.), vol. 318 of Electronic Proceedings in Theoretical Computer Science, pp. 229–241, Open Publishing Association, 2020.

[17] C. Zhang, H. Wang, Z. Ma, L. Xie, Z. Song, and J. Zhai, "Uniq: A unified programming model for efficient quantum circuit simulation," in 2022 SC22: International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp. 692–707, IEEE Computer Society, 2022.

[18] G. F. Viamontes, I. L. Markov, and J. P. Hayes, Quantum circuit simulation, vol. 410. Springer, 2009.

[19] R. Wille, L. Burgholzer, S. Hillmich, T. Grurl, A. Ploier, and T. Peham, "The basis of design tools for quantum computing: Arrays, decision diagrams, tensor networks, and zx-calculus," in Proceedings of the 59th ACM/IEEE Design

Automation Conference, DAC '22, (New York, NY, USA), p. 1367–1370, Association for Computing Machinery, 2022.

[20] A. Zulehner and R. Wille, Introducing design automation for quantum computing, vol. 11. Springer, 2020.

[21] Y. Ding and F. Chong, Quantum Computer Systems: Research for Noisy Intermediate-scale Quantum Computers. Synthesis Lectures on Computer Architecture Series, Morgan & Claypool Publishers, 2020.

[22] The ZX-calculus contributors, "The zx-calculus." https://zxcalculus.com/, 2023.

[23] Qiskit contributors, "Qiskit aer," 2023.

[24] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," Annals of Physics, vol. 349, pp. 117–158, oct 2014.

[25] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, "Qiskit pulse: Programming quantum computers through the cloud with pulses," Quantum Science and Technology, vol. 5, p. 044006, Aug. 2020.

[26] R. Vilmart, "A near-optimal axiomatisation of zx-calculus for pure qubit quantum mechanics," 2018.

[27] R. Duncan, A. Kissinger, S. Perdrix, and J. Van De Wetering, "Graph-theoretic simplification of quantum circuits with the zx-calculus," Quantum, vol. 4, p. 279, 2020.

[28] F. Pan and P. Zhang, "Simulation of quantum circuits using the big-batch tensor network method," Phys. Rev. Lett., vol. 128, p. 030501, Jan 2022.

[29] M. Treinish, "Building a Compiler for Quantum Computers - Presentation," Mar. 2021.

[30] P. Das, S. S. Tannu, P. J. Nair, and M. Qureshi, "A Case for Multi-Programming Quantum Computers," in Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture, MICRO '52, (New York, NY, USA), pp. 291–303, Association for Computing Machinery, Oct. 2019.

[31] G. Li, Y. Ding, and Y. Xie, "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices," in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '19, (New York, NY, USA), pp. 1001–1014, Association for Computing Machinery, Apr. 2019.

[32] L. Liu and X. Dou, "Qucloud: A new qubit mapping mechanism for multi-programming quantum computing in cloud environment," 03 2021.

[33] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, "Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers," in Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '19, (New York, NY, USA), pp. 1015–1029, Association for Computing Machinery, Apr. 2019.

[34] X. Zhou, Y. Feng, and S. Li, "Quantum circuit transformation: A monte carlo tree search framework," ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 27, no. 6, pp. 1–27, 2022.

[35] G. Li, A. Wu, Y. Shi, A. Javadi-Abhari, Y. Ding, and Y. Xie, "Paulihedral: A generalized block-wise compiler optimization framework for Quantum simulation kernels," in Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '22, (New York, NY, USA), pp. 554–569, Association for Computing Machinery, Feb. 2022.

[36] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," Nature Communications, vol. 9, p. 4812, Nov. 2018.

[37] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, "QuantumNAS: Noise-Adaptive Search for Robust Quantum Circuits," in 2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA), pp. 692–708, Apr. 2022.