边缘-云连续体中的量子计算架构愿景

Alireza Furutanpey[†*]、Johanna Barzen[‡]、Marvin Bechtold[‡]、
Schahram Dustdar[†]、Frank Leymann[‡]、Philipp Raith[†]、Felix Truger[‡]
[†]分布式系统组，维也纳工业大学 first.last@dsg.tuwien.ac.at
[‡]应用系统架构研究所，斯图加特大学 first.last@iaas.uni-stuttgart.de
*通讯作者

arXiv:2305.05238v1 [quant-ph] 9 May 2023

摘要——**目前，量子处理单元（QPU）仅能通过云服务商获取。然而，随着近期技术进展，不久之后，几乎所有地方都有望部署量子处理单元。现有研究尚未借鉴边缘计算领域的研究成果，探索如何利用移动量子处理单元的系统，或混合应用如何从分布式异构资源中获益。因此，本文提出了一种面向边缘-云连续体的量子计算架构。我们讨论将现有经典边缘计算研究扩展至集成量子处理单元的必要性、挑战和解决方案。我们阐述热启动如何支持定义工作流，从而利用遍布计算连续体的分层资源。随后，我们介绍一种采用经典-量子混合神经网络（量子神经网络）的分布式推理引擎，以帮助系统设计人员适配具有复杂需求、异构性最高的应用。我们提出了侧重于经典层划分和量子电路切割的解决方案，以展示如何利用计算连续体中的经典计算和量子计算。为了评估这一愿景的重要性和可行性，我们提供了概念验证，展示如何通过扩展经典划分方法以集成量子电路来提高解的质量。具体而言，我们实现了一种可选用混合量子神经网络预测器的拆分神经网络。结果表明，使用量子神经网络扩展经典方法切实可行，并具有良好前景。**

索引词—**量子计算，边缘计算，计算连续体，拆分计算，电路切割，任务划分，深度神经网络划分，经典-量子混合机器学习，量子神经网络，热启动**

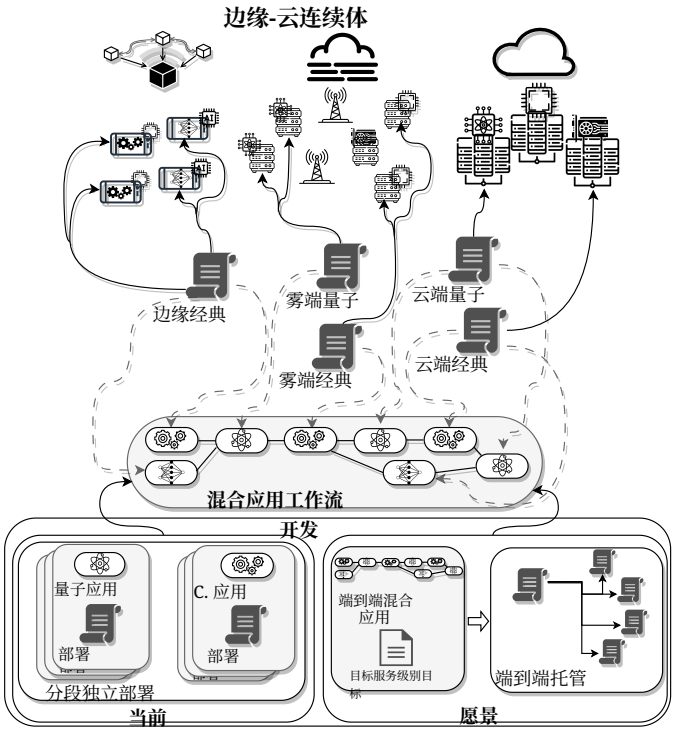

图 1. 分布式经典-Quantum混合平台

I. 引言

含噪中等规模量子（NISQ）计算机容易出现错误，所包含的量子比特数量有限，并且对可成功执行的电路深度施加了限制 [31]。然而，针对 NISQ 设备量身定制的算法

已开始展示量子计算机在多个领域的可行性，涵盖分子模拟 [27]、机器学习 [11] 和优化问题 [14]。显然，为了推动量子算法的研究与开发，并将其用于实际应用，通过引入适当的抽象来提高量子计算机的可访问性是有效的。然而，让研究人员和从业者能够访问量子处理单元（QPU）颇具挑战。由于量子处理单元（QPU）的可用性有限、复杂且成本高昂，目前大众进行量子计算或许只能通过云服务来实现，而云服务能够借助便捷的接口隐藏底层运作机制。

然而，尽管云服务提供商能够降低复杂性和成本，但我们不能一味依赖硬件制造商的努力，通过简化量子处理单元的生产和安装来提高资源的可及性。相反，我们必须借鉴经典计算领域的经验，了解过度依赖集中式云平台会带来哪些影响。除了将敏感数据托付给第三方提供商所产生的隐私相关风险外，云计算范式还存在诸多缺点，例如供应商锁定，以及数据中心容易因中断而成为单点故障。此外，狭隘的云中心视角效率低下，因为它会不加区分地将任务卸载到远程服务器，使客户端附近的宝贵资源（例如边缘、雾）闲置。

边缘-云连续体通过聚合分层架构中的资源，弥补了云计算的局限性

该分布式网络涵盖从受限的边缘设备到云数据中心的各个层级。遗憾的是，经过数十年对集中式架构的依赖，这一转型进展缓慢，半去中心化或完全去中心化的平台仍有待出现 [45]。因此，为避免重蹈经典计算的覆辙，在集中式方案固化之前，帮助研究人员在分布式平台上实现量子应用至关重要。

值得注意的是，随着金刚石量子处理器 [25] 的最新进展，人们有望很快在移动量子设备上部署边缘应用；这类处理器 [37]能够在室温下进行量子计算。因此，可以合理地设想，量子计算机很快就会广泛应用于个人和组织。正如移动设备上的人工智能（AI）加速器催生了依赖计算机视觉（CV）或自然语言处理（NLP）、对性能要求极高的边缘智能应用程序一样，移动量子处理单元的出现也将为新一类智能应用程序铺平道路，例如，这些应用程序可从高效解决复杂优化任务中受益。即便如此，即使采用集中式方法，开发混合应用程序的复杂性仍严重阻碍着量子研究的发展。从业者必须考虑经典和混合组件，选择合适的硬件，并手动管理编排。为此，本文

表 I 现状与我们的愿景概述

类别	当前情况	愿景
可及性	受限、集中式， 整合资源	可选择集中式或分布式
复杂性	需要选择组件（例如，量子 计算机）、多个供应商和框架	异构组件被抽象 为统一的接口
应用程序	受限于云，网络延迟高	不受限制，设备协同
服务保障	需要进行端到端评估 手动拼接应用程序的性能	针对单个混合应用的服务等级目标。 自动化编排资源分配
工作流	必须整合单个经典和 量子服务提供商手动地	可自由组合的工作流 of 量子与经典组件
资源局部性	仅云端	边缘-云连续体

旨在推动边缘侧量子计算研究。我们阐述了分布式混合经典—量子架构面临的关键挑战及可能的解决方案。具体而言，我们重点关注任务划分，以展示借助边缘-云连续体的潜力，并解释系统设计如何将众多经典和量子组件巧妙地交织在一起，组成一个协调统一的整体。最终，我们设想出现一个分布式混合平台，能够自动编排混合应用的端到端流程。届时，正如图 1 所构想、表 I 所概述的那样，实践者便可专注于实现其应用和算法的核心逻辑。

为了具体阐释并加深对本文抽象概念的直观理解，我们以实现一个用于移动增强现实（MAR）的分布式混合平台为贯穿全文的示例。考虑到为满足 MAR 应用程序严苛而复杂的需求而设计的方法能够很好地推广到任意应用程序，且这一方向是一个活跃的研究领域，因此我们的选择是合理的研究领域 [61]。此外，承载 MAR 应用程序的平台必须可靠地执行众多可受益于量子处理单元（QPUs）的智能任务。例如，为骑行者提供安全导航的系统，需要利用分布式摄像头网络 [54]中的目标检测和跟踪模型，帮助骑行者在城市区域安全通行，避免与驶近的车辆相撞。我们将贡献总结如下

- 强调将量子处理单元纳入边缘-云连续体所带来的机遇，以及量子处理单元的独特属性所引发的新挑战。
- 介绍一种分布式混合平台架构，其中量子节点和经典计算节点的组成可灵活变化。
- 通过实现、评估并开源一个可拆分的混合经典—量子神经网络[1]，展示将经典深度神经网络（DNN）的分区方法扩展至量子嵌入的可行性，并提供概念验证（PoC）。

第二节介绍基础背景和相关工作。第三节阐述未来的量子边缘计算（QEC）研究如何借鉴经典边缘计算（CEC）的现有工作，并以人工智能加速器及其在边缘智能中的作用为例进行类比。本文其余部分将自上而下逐步介绍我们构想的平台组件。第四节介绍面向客户端程序员的核心高级无服务器抽象模型。第五节扩展该模型，以提供对热启动的一流支持。第六节聚焦分布式混合推理引擎面临的底层系统挑战。第七节介绍并评估我们用于拆分推理的混合经典—量子深度（量子）神经网络概念验证。第八节总结全文。

II. 背景与相关工作

以云为中心的平台为公众使用经济高效的大规模应用铺平了道路。然而，新兴的边缘-云连续体凸显了集中式架构的缺点。边缘智能等前景广阔的应用范式 [17], 高度依赖边缘-云连续体，并要求对规模庞大且异构的系统进行自主管理。我们认为，量子计算机能够提升现有应用的质量，并为新型应用铺平道路。这些应用能否成功，取决于可用的平台是否能够支持开发者对其进行设计、编写、测试、部署和管理。本节介绍我们架构愿景的基本概念，并概述相关工作。

A. 编排

提供量子计算机访问能力的集中式平台服务，可以与经典应用相结合（即，Amazon 为其量子服务提供基于事件的处理）。因此，从业者和研究人员必须

[1]https://github.com/rezafuru/QuantenSplit

通过组合独立的量子与经典组件来构建混合应用程序。更糟糕的是，他们还必须费心选择不同的量子处理器技术、设备和编译器 [23], [52]。

1）量子应用编排：尽管量子应用由经典组件和量子组件构成，但与经典计算一样，它们也将编排划分为两个不同的流程 [66]。首先，工作流技术负责管理控制流。其次，配置技术负责处理应用组件的部署。因此，我们可以扩展现有系统以支持量子应用，从而降低基础设施的复杂性。Wild 等人提出了 Tosca4Q，它扩展了 Tosca，以支持依赖量子计算机的工作负载 [69]。Weder 等人引入了量子应用档案（QAAs），使编排方法能够将量子应用视为自包含实体 [66]。随后，Leymann 等人提出通过市场扩展 QAAs，并设计了一种协作软件平台架构，以涵盖开发过程 [32]。

2）量子平台：多种云服务提供了以服务形式使用量子计算的途径。然而，要将托管量子服务无缝集成到经典应用中仍然颇具挑战。

Garcia-Alonso 等人 [23]介绍了其量子 API 网关的概念验证实现，该网关可为 Amazon Braket 推荐运行给定量子应用的目标量子计算机。此外，Beisel 等人 [6] 提出了*Quokka*，这是一种用于建模和部署量子工作流的基于微服务的框架。作者提出了一组微服务，用于建模基于变分量子算法（*VQAs*） [12]的典型量子工作流。该工作流包括量子电路生成、执行、误差缓解、目标函数评估和参数优化。这种方法将预处理、执行和后处理的各个部分完全解耦，从而支持灵活地定义工作流。此外，Salm 等人 [56] 提出了一种概念，可自动处理量子算法分析和量子计算机选择。Grossi 等人 [28] 构建了一个受无服务器计算启发的原型平台，量子开发者可通过该平台部署其应用。他们采用了一个专注于队列管理和结果检索的调度器。Leymann 等人 [32] 提出了一种面向量子应用的协作软件平台架构，通过量子应用市场涵盖开发过程和部署环节。

迄今为止，这些系统展示了平台如何促进协作、改进应用程序开发并简化部署环节（例如，动态选择量子计算机）。然而，它们仅考虑了云系统，且重点关注软件开发。我们认为，这些领域的进展与我们的工作相辅相成，能够丰富我们所设想的混合应用平台的开发环节。例如，采用 Tosca4Q [69] 对无服务器应用建模 [70] ，可以让我们设想的基于无服务器架构的平台更易于使用。尽管如此，要让量子应用触手可及，仍需要一个能够管理边缘云计算连续体中的混合应用的平台。

B. 无服务器边缘计算

边缘云应用（例如边缘智能）的一个关键问题是边缘云应用的自主编排 [17]。在这些大规模的地理分布式基础设施中，手动管理不可行，因此需要一个能够自主管理应用部署的平台。

无服务器边缘计算是无服务器计算的自然延伸，它抽象底层基础设施，并在整个系统中透明地将打包为函数的应用程序进行部署 [3], [45]。我们认为，自主管理以及简化的应用开发与部署，能够推动新兴的量子计算范式。Nguyen 等人 [47] 提出了一个整体式无服务器平台，支持经典、Quantum 和混合应用程序。与之相反，我们设想一个贯穿边缘-云计算连续体的平台，能够跨异构基础设施管理应用部署。正如第四节将讨论的那样，复杂性增加源于组合应用，以及精细而复杂的监控。

C. 任务划分

经典边缘计算与量子计算中的任务划分是截然不同的研究领域，分别解决彼此正交的问题。尽管如此，它们在将任务划分为可由隔离计算节点执行的子任务这一点上有着共同的理念。

尽管我们可以借鉴现有的量子计算划分方法，辅助调度器制定放置策略，但在将其推广到经典方法之前，必须考虑量子处理单元的独特属性。

经典边缘计算中的分区旨在分配负载，以提高资源效率 [34]。在量子计算中，为了让近期应用能够应对 NISQ 设备的局限性，需要在经典计算节点与量子计算节点之间划分任务 [31]，；而最常见的经典量子混合划分模式会为各个组件分配固定角色 [67]。此外，量子计算的模式通常是为执行某一类特定算法而设计的，并未考虑量子算法仅仅是多个子任务之一的应用场景。

相比之下，随着量子计算的局限性逐渐减小，平台应能够适应新兴模式。例如，IBM 于 2022年 宣布的 433 量子比特量子处理单元很快将被 1’121 量子比特系统取代 [18]。此外，对于近期和长期的量子处理单元，平台应根据目标服务级别目标、内部条件（例如负载）和外部条件（例如带宽），动态调整量子计算节点与经典计算节点之间的工作负载。

1) 变分量子算法：变分量子算法 *(VQA)* 是在经典计算机上优化量子电路参数的通用框架 [12]。根据目标任务，我们可以推导出更具体的算法，例如 变分量子本征求解器，用于近似矩阵的最低特征值 [62]

或量子近似优化算法（量子近似优化算法）来近似求解组合优化问题 [59]。另一个值得注意的变分量子算法实例是量子神经网络，旨在通过希尔伯特空间中的嵌入来改进经典神经网络的表示。请注意，在文献中，变分量子算法、量子机器学习（QML）和量子神经网络之间的区别并不明确；因此，为了清晰起见，我们将量子神经网络称为针对典型机器学习任务（例如特征提取、回归或分类）构建和训练的模型。

2）热启动：术语热启动由于其在经典和量子计算中的广泛使用而存在歧义。例如，它可能指用于减少机器学习和优化中资源使用情况的技术 [2]。相比之下，在经典无服务器计算中，热启动通常与准备执行环境的方法有关，例如重新利用运行中的容器 [38]。本文将热启动视为一种通用策略，即在辅助设备上部分计算或准备量子算法的输出。值得注意的是，热启动方法并不局限于经典到量子，也可以是量子到量子或量子到经典。

经典-量子混合系统可以受益于各种利用先前获得的解、近似结果或训练模型的热启动方法 [64]。例如，基于以下假设：使用变分量子算法求解相似问题实例时，其最优变分参数彼此接近，因此可以在不同实例之间传递参数，将其作为热启动的初始点，并在此基础上进行改进 [22]。此外，还可以利用由高效经典算法廉价生成的近似结果，为量子电路初始化量子态，使其偏向潜在解，而不是从中性的初始态开始 [19]。另一方面，量子神经网络可以通过迁移学习受益于预训练模型，即调整一个针对一般任务训练的经典或混合模型，并进一步训练它以处理相似或更具体的子任务 [39]。显然，由于这些热启动方法包含一个用于提取信息的源算法，以及一个借助这些信息得到增强的目标算法，热启动可能揭示出在计算连续体中分配经典和量子计算工作量的潜在方式。

3) 深度和宽度方向的 *D(Q)NN*划分：深度方向划分是指在垂直层之间拆分大型深度神经网络，因此会形成存在顺序依赖关系的分区，且不会提高并行化程度。在经典计算中，深度方向划分方法通常旨在让多个设备能够跨计算连续体处理请求，从而提高资源效率 [20]。在量子计算中，深度方向划分可能是指堆叠量子神经网络电路，以在训练期间缓解级联门错误和累积噪声 [9]。

相反，宽度方向划分通过横向拆分一层或电路来促进并行化。量子电路切割通过将较大的电路划分为若干较小的子电路，解决 NISQ 设备的一些限制 [4]。经典的宽度方向深度神经网络划分并不常见，因为人工智能加速器会并行执行各层。不过，在无法使用服务器级硬件、限制严格的环境中，跨设备并行计算卷积层的滤波器仍是可行的方法 [71]。

我们认为，经典深度方向划分和量子宽度方向划分是我们所设想的平台中一个关键组件的可行方法，并将在第六节详细介绍。

III. 量子计算与边缘智能

我们认为，量子边缘计算（QEC）的研究应直接拓展经典边缘计算（CEC），而不是促成彼此孤立的研究群体。具体而言，我们发现，未来的 QEC 研究可以从边缘智能（EI）的现有研究中汲取宝贵经验。智能任务通常指解决那些采用经典控制结构的程序无法高效计算或无法以足够精度计算的问题。边缘智能利用面向受限设备（例如人工智能加速器）的专用硬件进展，借助现代人工智能方法（例如深度神经网络）来推动此类任务的计算。同样，QEC 可以利用（移动设备）量子处理单元的进展，高效地计算众多任务（例如优化问题），并实现更高精度。

为确定我们可以借鉴哪些经验，本节将高能效人工智能加速器的兴起与集成量子处理单元所面临的挑战进行对比。需要注意的是，尽管“人工智能”有更广泛的定义，但边缘计算领域的文献通常将人工智能视为经典机器学习和深度学习的同义词。为避免歧义，我们将遵循这一惯例。此外，为简洁起见，我们将能够高效执行深度神经网络供应商专用编译计算图的一大类适用于移动设备的芯片统称为张量处理器（TPU） [33]。

A. 移动量子处理单元与张量处理单元的相似之处

量子处理单元能够高效解决一些经典计算机难以处理的问题（例如整数分解和离散对数 [60]）。值得注意的是，量子处理单元可以利用量子纠缠和叠加，这两种特性是量子计算所独有的。相较之下，尽管添加张量处理器在形式上无法与量子处理单元所能带来的加速和精度提升相提并论，但在实践中，张量处理单元能显著加速深度神经网络擅长处理的复杂任务。

为简洁起见，我们将专为资源受限设备（例如物联网设备）设计的深度神经网络，或经量化、剪枝或知识蒸馏方法压缩的深度神经网络 [16], [26] 称为“轻量级”深度神经网络。类似地，我们将设计用于比最先进量子处理单元能力更受限的量子处理单元的电路称为“浅层”或“窄型”电路。这里的参照标准是相对的，其依据是这样一种假设：资源受限设备中的集成芯片，其容量低于当代服务器级硬件中的芯片。正如张量处理单元是用于轻量级深度神经网络 [13] 、以解决计算机视觉和自然语言处理问题的加速器一样，移动量子处理单元将使受限设备能够运行浅层量子电路来解决优化问题。

总而言之，将量子处理单元视为针对一组正交智能任务（即经典手工编写程序难以实际处理的任务）的加速器，是合理的。尽管如此，在短期内，量子处理单元和张量处理单元也能在非正交任务上相互补充。例如，量子神经网络的主要动机与量子计算的典型优势有着有趣的不同。由于深度神经网络的训练和推理高度可并行化，现代经典硬件已经能够高效地处理这些任务。相比之下，量子神经网络可以利用希尔伯特空间的性质来寻找更好的表示 [1]。此外，研究表明，在训练数据中利用纠缠可以减轻（经典）无免费午餐定理对可学习性施加的限制 [58]。然而，尤其是对于高维数据（例如图像），构建和训练量子神经网络颇具挑战，因为相关方法必须将高维表示映射到数量较少的量子比特上。因此，目前混合经典-量子神经网络展现出更有希望的成果。与纯量子神经网络和其他变分量子算法不同，混合量子神经网络由参数化的经典组件和量子组件构成，并会先应用经典层来寻找合适的表示，再将特征编码为量子态并传入量子电路 [36], [39]。

B. 面向边缘与运行于边缘的 *Quantum*

在其开创性研究中，Denget al. [17] 提出将边缘智能区分为*边缘侧 AI* 和*边缘上的 AI*。借鉴他们的研究，我们建议将未来关于量子边缘计算的研究划分为运行于边缘的 Quantum 和面向边缘的 Quantum。

1) 运行于边缘的 *Quantum*（*QoE*）：探讨如何为依赖量子处理单元的分布式应用构建系统所面临的挑战。此类研究可以聚焦于将现有量子系统集成到计算连续体中，使分层网络中的经典资源能够帮助减少量子处理单元的等待时间和负载。或者，也可以着眼于可泛化的方法，通过量子处理单元提升现有经典系统的解的质量。第五节和第六节将分别介绍这两个子类别的构想方法，以推动这一方向的发展。

2) 面向边缘的 *Quantum*（*QfE*）：聚焦于解决边缘计算中的优化问题。与 QoE 不同，它并非与经典 AI 的对应方向互补。相反，量子研究人员应致力于超越 AI 方法。现有的经典方法可用于热启动
量子优化。此外，它们还可作为有价值的基准，标示出量子算法在实际应用前必须跨越的门槛。值得注意的是，研究表明，对于某些问题实例，量子算法的表现优于经典优化方法，而且在运行算法之前，就能以较高概率识别出这些实例 [43]。

IV. 分布式混合计算平台的架构

本节介绍图2所示的愿景架构。我们首先提供边缘-云连续体的高层模型，并将其与我们的示例联系起来：一个旨在实现我们架构的 MAR平台。然后，我们描述各个组件及客户端与它们的交互方式，以便后续各节可以引用该架构。我们并不声称这些组件

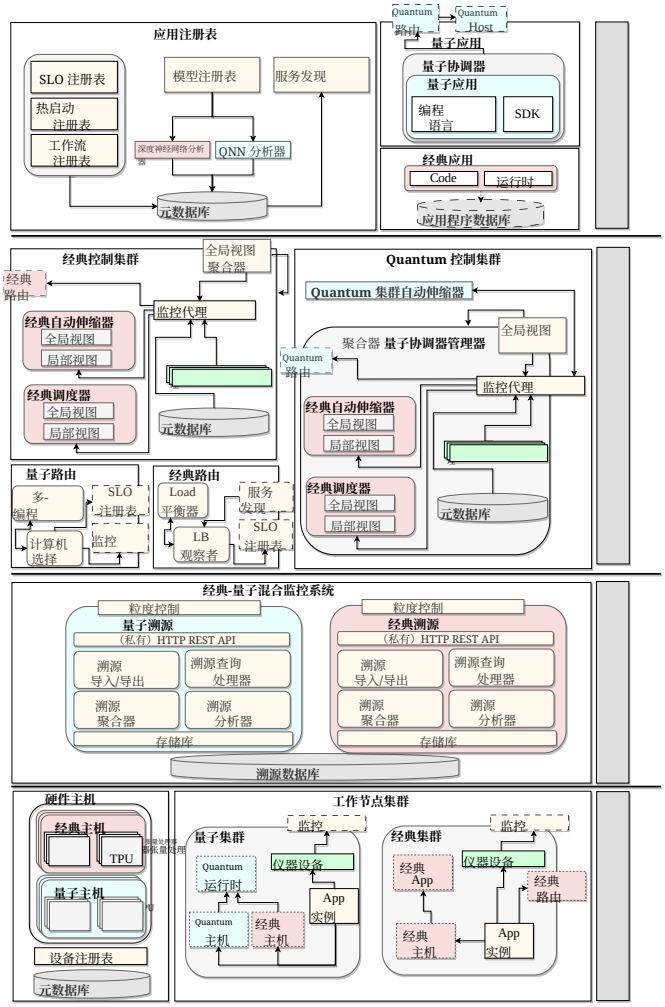

图 2. 我们构想的架构

我们在架构中提出的内容足以构成一个混合平台。相反，我们认为它为未来的工作奠定了基础，并为推进量子计算和边缘计算开辟了新的研究方向。

A. 资源与场景构建

我们的概念性 MAR 平台区分系统设计人员、客户端程序员和客户端。系统设计人员寻求构建平台的方法，使客户端程序员能够部署应用程序或实验。客户端程序员编写并部署 MAR 应用程序，且需求各不相同。客户端是通过发送请求来访问应用程序的设备。我们的目标是构建一种架构，通过无条件地 降低客户端程序员的复杂度（即使增加系统设计人员和平台提供商的负担），从而最大限度地加快量子应用和算法的发展。

1）服务级别目标与协议：借助服务等级目标，客户端可以根据应用程序的需求，指定平台应当满足的可量化指标。通常，提供商会将服务等级目标打包为服务级别协议（SLA）。不过，我们无需区分服务等级目标与服务级别协议，因为后者只是可以根据前者构建的协议。即使在纯经典的无服务器（边缘）云系统中，保证实现服务等级目标也很困难 [44], [53], [57]。下文提出一些解决方案，以构想一种平台架构愿景，并进一步考虑集成量子处理单元。第三节强调，应扩展现有方法以适配量子处理单元，而非构想不了解经典边缘计算近期进展的新方法。因此，我们从系统视角出发，借鉴经典计算领域正在进行的研究，旨在扩展现有最先进的经典方法，而非与之竞争。在本文余下部分中，如果某个组件或服务能够访问 SLO 注册表并实现相应的决策机制，我们就称其具备 *SLO* 感知能力 。

2）边缘-云连续体上的资源假设：资源分布在三个领域：边缘、雾和云。对于边缘领域和雾域，目前尚无普遍认可的定义。因此，在不影响架构实用性的前提下，我们尽可能采用宽泛的定义。就我们的目的而言，将雾域定义为由集群计算节点组成、且比最近的云数据中心更靠近客户端设备的区域，已足够。边缘领域由客户端及附近具备 SLO 感知能力的负载均衡器 [8] 组成。表 II 总结了各领域的特性，以及 MAR平台可访问的资源。特别是对于量子处理单元，我们指的是当前或预计近期推出的设备的特性。尽管 QPU 最终成熟后或许能像经典硬件一样易于获取，但系统设计人员仍需应对 QPU 近期的局限性。我们区分移动设备与

表 II 各领域的资源概览

领域	主要成本	Type	级别	容量	显著局限性
Edge	能量 - 客户端 可能无法进行热启动	CPU	移动设备	Low	异构性
		TPU	移动设备	非常低	轻量级深度神经网络
Fog	负载 - 请求可能 需要路由至 云。	QPU	移动设备	Low	电路宽度/深度
		CPU	服务器	High	中断
		GPU	服务器	High	中断
云	通信 - 远程数据中心 产生高延迟	QPU	服务器	中	可用性
		CPU	服务器	无限	干扰
		GPU	服务器	无限	干扰

以及服务器级硬件。前者只能执行后者的一部分。容量指模型的大小以及我们可以按需启动多少个实例。无限意味着我们可以根据需要托管任意多个最大的现成模型实例。高 意味着模型大小与实例数量之间存在一定权衡，当并发客户端请求数量超过特定阈值时，可用容量可能会耗尽。中 和 低 则表示硬件还受到其他限制，例如电路深度和数量的限制。

NISQ 设备上的量子比特数量限制了量子神经网络的规模。在我们贯穿全文的示例中，平台提供商计划在多个主要城市附近部署量子处理单元。因此，他们购买了大量移动级量子处理器（例如 Quantum Brilliance [37] 或 SaxonQ [25] 宣布推出的基于金刚石的量子处理器），并预计这些处理器的容量足以满足大多数模型的需求。由于他们无法在每个受支持的位置都部署服务器级 QPU（例如 IBM 计划推出的 1’121 量子比特系统 [18]），需要运行更大电路的请求要么转发至远程云数据中心，要么通过分区方法处理，我们将在第六节详细阐述这些方法。最后，我们将边缘张量处理单元的容量标为“非常低”，因为它们通常只能在内存中容纳一个轻量级深度神经网络。与之相反，在雾和云领域，我们可以在虚拟化图形处理器上运行同一深度神经网络的多个实例。资源的一项显著限制未必仅存在于某个单一领域，但系统设计人员应特别关注该领域中的这一问题。异构性指的是具有不同限制的众多芯片架构。轻量级深度神经网络和电路宽度/深度指的是张量处理单元和多量子处理单元在运行更大模型时受到的限制。干扰指的是向多个客户端提供虚拟化资源所导致的性能下降 [46]。

B. 架构平面与组件

该架构根据四个平面的功能以及它们与其他组件、客户端或客户端程序员的预期交互方式，对其进行了区分。每个平面都会公开私有或公共 API。私有 API 仅供内部组件访问，而公共 API 则可供外部实体（即客户端和客户端程序员）访问。

1) 执行平面： 应用实例在执行平面上运行。该平面由公共应用程序接口、硬件主机和工作集群组成。公共应用程序接口使客户端能够通过接入点与应用交互。

硬件主机根据所支持的应用类型进一步细分为经典计算节点和量子计算节点。量子主机还会按等级进一步细分，即 QPU 属于服务器级，MQPU 属于移动设备级。经典计算主机还会进一步细分，以支持人工智能工作负载。每个节点都由设备注册表 登记；设备注册表会关联描述节点容量的数据（例如内存大小、VRAM、量子比特数），并将其存储到元数据 存储中。元数据存储是一种高可用的键值存储（例如 Kubernetes 中使用的 ETCD），供其他组件访问。尤其是在计算连续体中，节点可以随意加入或离开系统，因此元数据必须具备高可用性且保持最新，才能准确衡量系统的总体容量。

工作集群 由至少一个硬件主机组成，代表可能依赖一个或多个硬件节点的应用环境。由于量子环境和经典环境彼此独立，因此有必要区分经典工作节点集群和量子工作集群。显然，量子应用需要经典工作器来执行辅助任务（例如预处理、测量）。一旦

工作节点完成任务后，经典工作节点负责转发结果。对于将任务分区到多个设备上的分布式应用，结果会转发到另一个工作节点集群。对于单体应用或分布式应用的最后一个分区，结果会转发给客户端。监测工具通过访问 ProvenancePlane 的私有 API，向监控系统发送遥测数据。

2）溯源平面：溯源平面通过高度可用的分布式溯源数据库，实现实时监控数据的存储与共享。它由量子溯源系统和经典溯源系统组成，并提供私有应用程序接口，用于访问和存储遥测数据。设计系统时，必须考虑量子处理单元的内在特性，才能为量子工作负载构建可靠且可预测的系统。值得注意的是，错误率会随量子处理单元的当前状态而变化；因此，对于具有解质量目标的服务级别目标而言，仅收集与量子和混合应用程序相关的经典数据（例如负载）并不足够。平台的经典监控会在应用旁部署监测工具（执行平面），以收集工作负载趋势以及函数实例的资源使用情况（例如负载、CPU、图形处理器、输入/输出）的遥测数据。与此同时，量子溯源系统会收集量子处理单元的状态信息，以分析错误，例如量子电路、量子处理单元、编译和执行的相关属性，正如 Weder 等人所提出的 [65]。平台可利用溯源数据缓解错误，并帮助调度器通过评估当前预期的解质量来维持服务级别目标。支持混合应用的平台开户应将量子溯源与经典监测工具集成，构建统一的监控系统，从而简化对各种异构设备的访问。

监控系统的目标是收集通知调度器以维护服务等级目标所必需的最少数据。构想混合系统并非易事，因为即使是经典边缘云和混合系统的监控，要找到平衡也已经颇具挑战 [24], [53]。通过激进地收集数据而将监控简单化的系统，可能会减轻调度器的任务，但也可能导致整个边缘-云连续体中的资源拥塞 [24]。相反，收集不足的遥测数据将显著降低系统的可扩展性，因为系统无法适当地路由请求或扩展和收缩资源。我们认为，一个粒度机制 能够根据当前工作负载的特征调整 Quantum 和经典监控数据的频率，这是未来工作必须解决的主要挑战之一，只有这样，混合平台才有可能出现。然而，监控本身仅仅是一个前提条件。以下内容将介绍我们的架构如何基于可用监控数据支持资源效率和弹性。

3）弹性平面：弹性平面是系统的核心组成部分，负责根据客户端的服务级别目标、元数据和监控数据，决定如何分配资源和路由请求。控制集群分为量子控制集群和经典控制集群。前者管理量子应用实例（例如，扩展量子协调器），后者管理经典应用实例（例如，水平扩展应用程序）。每个控制集群都管理一组工作节点集群（见执行平面），并根据该集群可控制的应用实例动态扩缩这些工作节点集群。监控代理负责监控工作节点集群，并将数据传递给控制集群的监控代理。监控代理会将数据分发到各个组件。经典自动伸缩器和经典调度器负责维护控制集群状态的局部视图和全局视图。量子控制集群中也设有经典自动伸缩器和调度器，因为量子协调器应用（见控制平面）属于经典应用。量子协调器管理器负责监督经典自动伸缩器和调度器，以扩缩并放置量子协调器应用（见控制平面）实例。

局部视图 包含有关工作节点集群的细粒度信息（例如，每秒的 CPU 使用率）。相比之下，全局视图 包含有关相邻控制集群的粗粒度信息（例如，一小时内的平均 CPU 使用率）。粗粒度数据由全局视图聚合器收集的细粒度局部数据组成，该聚合器会定期发布摘要，即全局视图由交换的局部视图摘要构成。

控制集群的消息传递拓扑和消息代理在一定程度上解决了监控数据的粒度控制问题，并通过适当分配有限资源，实现了能够扩展整个系统的弹性控制机制。自动伸缩器和调度器的实现可以互换，系统设计人员可以尝试各种方法。*Quantum* 集群自动伸缩器受到 Tamiru 等人和 Gandhi 等人的工作启发，是一种具备 SLO 感知能力的集群自动伸缩器，能够从工作节点集群中添加和移除量子主机，以处理传入的工作负载。

经典路由受到 Raith 等人的工作启发，由一个负载均衡器和一个负载均衡器（*LB*）监视器组成。负载均衡器是一个高吞吐、低开销的组件，可将传入请求重新定向到应用实例或其他集群（例如，因为没有正在运行的应用实例）。负载均衡器监视器具备 SLO 感知能力，并定期刷新负载均衡器的状态，以更新决策机制。

量子路由组件有别于经典路由，因为它考虑了额外的挑战，以提高量子主机的资源效率。量子计算选择 对于边缘-云计算连续体尤为重要，因为它引入了更多异构性。选择方法是另一个可自由互换的组件。例如，系统设计人员可以选择采用 Quetschlich 等人提出的方法 [52] ，并在找到更合适的替代方案后将其替换。量子计算中的多程序设计 与经典计算中的虚拟化 具有类似的作用，即允许多个电路共享量子计算机的资源 [15]。然而，与经典计算不同，在经典计算中我们可以直接选择现有的成熟虚拟化方法，而多程序设计.....

涉及的因素更多，因此在编码过程中应一并考虑 [48]，并会影响量子计算机选择。

4) 控制平面：控制平面向客户端程序员提供应用程序接口，以便部署和管理应用程序。公共API应允许客户端程序员将混合应用定义为工作流，而无需分别部署经典部分和量子部分。平台会在注册过程中分析工作流，并将其划分为经典应用和量子应用。应用注册表封装了多个注册表，用于存储应用服务、SLO 目标和参数化模型。模型注册表类似于模型动物园（例如 Torch 图像模型 [68]）。不过，这里的模型既可以是经典神经网络，也可以是量子电路。此外，分析器会为每个模型关联静态元数据（例如参数数量、电路深度和层类型）。分析器元数据为调度器和自动扩缩容器提供有价值的信息，帮助它们更准确地预测资源使用情况（例如，Swin-Tiny 的资源使用量会高于 Swin-Base [35]）。我们的架构将热启动作为一等功能予以支持，第五节将对此进行详细说明。服务发现使 Quantum 路由和经典路由能够定位正在运行的应用实例。

经典应用由应用代码和一个运行时组成，该运行时会在创建应用实例时执行代码。该架构不对运行时作任何假设，即运行时可以是从容器到 WebAssembly 模块的各种形式。应用还可以选择配置任意的应用数据库（例如关系型数据库、面向对象数据库）。

量子应用更为复杂，因为它需要量子资源与经典资源协同工作。量子协调器包含由客户端程序员实现的量子应用。量子应用使用特定的编程语言和量子计算软件开发工具包编写。量子协调器负责协调量子应用（即量子电路）的执行。量子应用需要具备路由能力，以连接其经典部分与量子部分：经典部分负责对输入和输出进行预处理和后处理，而量子部分则是执行量子电路的量子处理器。除此之外，量子协调器还需要具备量子路由能力，以选择合适的量子主机，或执行多程序设计来提高量子处理单元利用效率。

V. 边缘侧的热启动

热启动与分布式混合平台的目标一致，即有助于利用计算连续体中的资源。理想情况下，应用程序可以在将输入传递给远程服务器之前对其进行预处理。在量子计算的背景下，热启动方法可分为经典到量子（C2Q）、量子到量子（Q2Q）或量子到经典（Q2C） [64]。每个类别都有相应的输入和输出格式。例如，C2Q 接受经典输入，而输出格式应适用于量子算法。

然而，我们认为系统设计人员必须进一步细分这些类别，以纳入神经输入和输出格式，例如神经到量子（N2Q）或经典到神经（C2N），原因有二。首先，神经方法依赖于人工智能加速器，而这些加速器可能并不存在，也可能无法降低能耗；也就是说，调度器必须了解硬件属性才能达到服务级别目标。其次，尽管经典深度神经网络的输出是经典的，但网络权重可能经过调整，以适配针对特定算法类别提取的特征。

A. 热启动的当前与未来作用

目前，热启动的一个常见动机是降低对量子处理器时间的依赖，因为其成本高且可用性有限 [64]。然而，我们强调，热启动的重要性在于通过结合经典算法和量子算法来提高解的质量。此外，一旦量子处理单元发展到可以完全不依赖经典计算的程度，研究重点便可转向 Q2Q 热启动，使较小的客户端设备能够部分卸载量子算法，以提高资源效率。例如，热启动可以将经典算法的性能保证嵌入量子算法中，并减少（量子）机器学习所需的训练数据量。

热启动的一个缺点是，它会增加应用的复杂性，因为会引入更多需要管理的组件。因此，提供便捷的接口至关重要，该接口应将热启动作为一等公民予以支持，从而进一步将复杂性从客户端程序员转移到平台提供商。值得注意的是，热启动具有可链式组合的特性。因此，如果平台提供的接口能够体现网络中设备容量的层级属性，我们就可以在边缘-云连续体中自然地集成热启动方法。

以下描述了具备感知能力的分层热启动编程模型的需求和拟议的解决方案。分层是指热启动方法如何在一个流水线 中进行组合，该流水线与其资源使用需求相匹配。所提出的接口不依赖于任何有关 QPU 的可用性和局限性的假设，即将流水线中的每种方法视为可互换的构建模块。当前可用 QPU 的发展进展通过向客户端程序员告知其计划的热启动流水线的可行性，反映在客户端程序员的使用过程中。例如，在多量子处理单元（MQPUs）在智能手机等终端设备中得到广泛采用之前，平台可以禁用对网络边缘 Q2Q 热启动的支持。

B. 分层热启动

在我们的运行示例中，MAR平台旨在通过分层热启动提高资源效率。城市规模应用的负载会随日期和时间大幅波动。通过分层串联热启动，可以利用边缘节点和雾节点的闲置计算资源，例如减少向云端卸载。

热启动是一个广义术语，涵盖多种方法类别 [64]。挑战在于设计一种足够灵活、同时又便于使用的接口，避免向客户端暴露底层细节，例如手动选择设备和

回退机制。若接口要求客户端程序员定义执行的每个步骤、指定各部分部署在哪个节点，并为流水线中的每种方法手动配置量子执行（例如设备、编译器），那么该接口将具有最大的灵活性。考虑到计算连续体的异构性，不限制配置空间的接口尤其不可取。具体而言，为某种方法仅提供单一配置并不足够，而且无法保证边缘或雾环境中存在特定的设备配置。反过来，若将客户端程序员限定在预先实现的解决方案列表中，也会适得其反，因为这会阻碍创新；也就是说，至少应允许他们（可选地）提供自己的热启动方法。

总而言之，平台的职责是构建基础设施，并提供足够的抽象来访问资源。专用于热启动的接口应允许客户端定义可组合工作流，以处理热启动流水线；也就是说，客户端程序员可以通过图 2 所示架构中的控制平面，注册热启动的预处理步骤，这些步骤可在部署于客户端设备或雾节点上的中央处理器、张量处理器或量子处理单元上运行。

请参考图3中的示例。三个客户端使用控制平面的公共API执行同一个应用程序。客户端程序员为其应用程序定义了热启动流水线。因此，流水线及其方法被放置在热启动注册表中，系统则根据分析器元数据决定在计算连续体中的何处放置模型。然而，客户端要求的目标质量各不相同；因此，在将任务发送给量子云服务商之前，平台会采用不同强度的热启动。强度指的是量子算法对经过热启动方法处理的输入进行处理后，预期达到的解的质量。在图3中，客户端 C1 无法实现

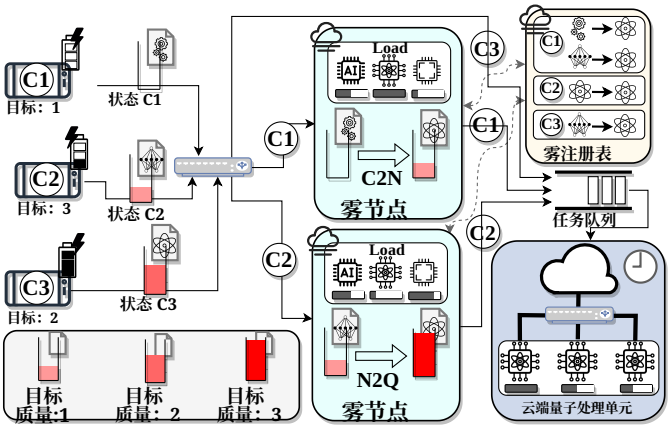

图3. 分层热启动的高级序列

由于能量约束（以低电量符号表示），客户端不会在边缘进行任何预处理，因此会以其原始的经典表示转发输入。由于 C1 在雾节点至少登记了一项预处理步骤，负载均衡器会将请求路由至一个雾节点，应用映射到经典到量子热启动的步骤。相反，如果 C1 具备用于为目标神经网络预处理其输入的资源，雾节点就会接管其输出的进一步处理，以热启动量子任务，从而形成经典到神经网络热启动和神经网络到量子热启动的链路。客户端 C2 可以执行资源感知的预处理，以实现经典到神经网络热启动。不过，由于其无法达到所需的目标质量，请求会被路由至一个具有可用量子处理单元的雾节点，以执行进一步预处理，实现神经网络到量子热启动。由于 C3 具备足够的机载资源，可在不依赖雾节点的情况下将任务预处理至目标质量，客户端 C3 的请求会由负载均衡器直接路由至云端。

VI. 用于经典-量子混合神经网络的分布式推理引擎

前两节介绍了该架构的高层概念，本节则聚焦于面向平台设计者的底层系统设计，以及客户端程序员如何实现分布式混合应用。我们将扩展贯穿全文的示例，并讨论MAR平台如何支持基于经典-量子混合神经网络的推理引擎。

（分布式）量子神经网络是深度神经网络推理之外一种具有代表性的应用，能够独特地说明：随着量子处理单元的限制逐步消除，分布式平台如何进行适应，原因有三。第一，混合量子神经网络由参数化经典节点和量子节点组成。相比之下，其他变分量子算法通常仅使用未参数化的经典组件进行预处理/后处理和参数优化。对于混合量子神经网络而言，还存在一些没有静态预定义角色分配的额外组件；也就是说，我们可以表示系统如何根据量子处理单元的可用性和发展，逐步以量子节点取代经典节点。第二，我们可以按层水平划分神经网络，或垂直切分单独的层，并将每个分区视为一个独立的计算图。这样，一个简单的推理请求（例如图像分类）就可以模拟复杂任务的行为：系统必须协调众多经典和量子子任务，才能计算出单一结果。第三，深度神经网络划分和协同推理是经典边缘计算中较为成熟的研究领域，它们考虑了经典硬件的异构性，以及边缘、雾和云节点之间的资源不对称性 [41]。因此，我们可以直接扩展现有工作，以确定量子处理单元是否能够提升经典方法的质量，而不局限于那些经典计算机无法处理的正交问题。

A. 系统挑战

我们考虑表 II 中总结的领域属性，以描述系统设计人员在实现我们的架构时可能面临的挑战。

与常规业务逻辑不同，智能任务（如图像分类、目标检测）依赖专用硬件，即无论我们是否纳入量子处理单元，该平台

必须将推理请求视为具有独特特征的工作负载。

1) 波动性：调度器必须动态适应两种波动来源。首先，雾域中的中断十分频繁。此外，与云不同，经典雾计算资源无法无缝地进行水平扩展，即请求可能必须被路由到云端。其次，雾端和云端的量子处理单元可能十分稀缺，并且根据其当前状态，可能无法达到目标解质量服务等级目标。

2）设备异构性：尽管经典组件的异构性挑战与量子处理单元的集成仅有间接关联，但仍需适当考虑众多加速器。编译器会将经典深度神经网络映射为计算图，而不同厂商对运算的支持各不相同，从而限制了可用的层类型和激活函数 [33]。

3) 任务链和带宽消耗：为了充分利用计算连续体中的资源，我们需要采用一些方法，将部分计算负载卸载到客户端加速器上。然而，移动设备通常只能在内存中容纳一个网络，而从存储中换出深度神经网络权重会带来显著开销。因此，发送不同任务后续推理请求的延迟敏感型应用必须进行卸载，导致宝贵的资源闲置。此外，当众多客户端通过传输高维图像数据争夺有限带宽时，有限的带宽不可避免地会导致响应延迟波动。

4) 可选量子嵌入： 尽管 QPU 的可用性正在稳步提升，但目前客户端无法期待云中经典资源实现同样灵活的扩展，即根据负载情况，使用量子层可能无法达到延迟服务级别目标。因此，推理引擎应具备足够的灵活性，能够跳过为近期和中期设备计算量子嵌入。

5) 利用移动量子设备： 预计金刚石量子加速器很快将成熟到足以投入商业使用 [25], [37]。然而，无论 MQPUs 如何改进，与经典硬件类似，我们假设服务器级硬件将始终优于其移动端对应设备。挑战在于构想出能够利用 MQPUs 优势的方法，同时最好不牺牲解的质量。

B. 解决方案

为支持多个任务和可配置的解的质量，最简单的方案是针对每种变体分别使用一个深度神经网络。这种方式难以维护且缺乏灵活性，迫使客户端程序员针对每项任务实现并重新部署完整的架构。更糟糕的是，它没有解决挑战 3)，即依赖多个任务的应用程序无法在不显著增加延迟开销的情况下利用客户端资源。相反，我们将说明分区方法如何促成可组合的架构，并自然定义出可部署的小型应用程序。

1）经典拆分计算：深度方向拆分计算解决挑战 2)-4)，并将层分组为多个分区。每个分区都是一个特征提取器，依次应用这些分区是我们在第五节中介绍的一种特殊形式的分层热启动。平台可以将预训练深度神经网络纳入控制平面的热启动注册表。此外，客户端程序员可以根据自身需求注册模块。例如，边缘设备可以选择执行初步特征压缩，而雾节点可以根据解的质量和延迟目标应用小型或中型特征提取器。

为解决挑战 2），我们需要一种适用于受限终端设备（例如智能手机）的编码器，其运算应得到各 AI 加速器厂商的广泛支持。为解决挑战 3），编码器应执行初步特征提取，并针对多个下游任务找到充分统计量的最小表示，以减少带宽消耗。随后，服务器可根据配置的延迟和准确率服务级目标，选择可互换的深度神经网络来进一步提取特征。最后，量子神经网络层应嵌入这些特征，以期在将其传递给分类器之前改善表示。不过，为解决挑战 4），即应对量子处理单元可用性有限的问题，量子神经网络应为可选项。

2）量子电路切割：量子电路切割旨在解决挑战 5）。其思路是通过策略性地切割电路线路 [10],[49] 和门 [4], [42]，将需要较多量子比特宽度的大型电路切分为所需量子比特较少的较小子电路 [4]。图 4 展示了一个示例，其中一条线路和两个门被切割。线路切割通过多次测量不同的可观测量 O_i ，并随后将量子比特重新初始化为状态 $|\psi_i\rangle$，从而分离电路线路；而门切割则以不同组合的单量子比特门替代双量子比特门。图 4 中堆叠的子电路表示为每个子电路生成多个变体。

宽度方向分区使每个子电路实例都能在较小且可能更易获取的量子设备上单独执行。执行完这些子电路后，会通过经典后处理程序重新组合各个子电路的执行结果，最终将原始电路的输出重构为子电路结果的线性组合。

这种方法无需量子通信，即可将量子电路计算分布到多个量子处理单元上。因此，量子电路切割让我们有机会利用边缘端的多个较小型多量子处理单元，从而计算规模更大的量子电路。此外，这种方法还能更灵活地将量子电路部署到计算连续体中的各类资源上。另外，随着多量子处理单元广泛普及，我们可以利用它们并行执行子电路。这种并行化能够减少每次切割所带来的开销，显著提升计算效率和可扩展性。

3) 推理流程：为了直观地展示我们提出的方法如何满足不同需求的客户端请求，请考虑图5中的推理请求流程。

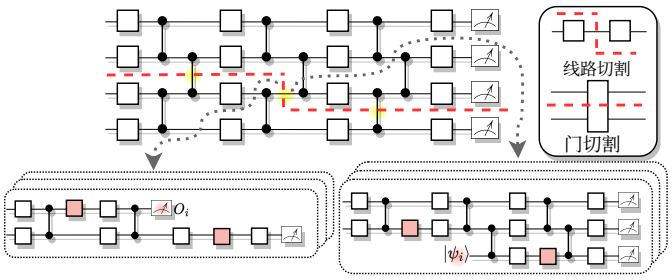

图 4. 电路切割基础

为简单起见，我们将示例限制为图像分类。然而，图像分类任务可能各不相同。例如，一个客户端对博物馆中的艺术品进行分类，以通过虚拟导览获取描述。同时，另一个客户端则希望获取当地动物群的描述。配备人工智能加速器的客户端会应用神经压缩方法进行初步特征提取。实线黑线表示域内请求流。虚线红线表示域间数据传输。虚线灰线表示由跨越所有域的运行时在推理请求附近收集的遥测与溯源数据。我们的架构中的弹性平面对运行时进行了详细说明，如图2所示；该平面收集遥测数据，并定期更新具备 SLO 感知能力的负载均衡器，使其根据每个参与节点的状态和客户端配置做出明智决策。虚线黑线表示一条替代路径，即每个请求只会流向其中一个选项。

边缘负载均衡器会将请求路由至雾计算集群的负载均衡器。根据负载和客户端要求，请求会被路由至雾计算 GPU 节点或云端。完成特征提取后，负载均衡器会决定是否在分类前将经典嵌入传递给量子神经网络。量子神经网络可以在量子雾计算节点上执行，也可以发送至远程云服务提供商（例如，出于隐私或可用性的考虑）。然而，根据我们的假设，多量子处理单元受到的限制比云服务提供商提供的服务器级 QPU 更多。因此，电路切割方法可以帮助多量子处理单元达到目标解的质量。分类标签作为响应发送给客户端后，请求即告结束。

第七节将详细介绍深度神经网络架构，并展示我们所提出方法的可行性。解决挑战 1)、5)并引入运行时组件（例如具备 SLO 感知能力的调度器、部署机制）也是有前景的研究方向，但不在本文的研究范围内。尽管如此，以下内容将说明平台如何切实地准备和部署神经网络组件。

4) 应用准备与部署：深度神经网络（DNN）的各项运算构成一个计算图。此外，分区方法可自然地将单体深度神经网络划分为彼此可连接的顶点。这些顶点表示粗粒度的

粒度经典层或量子神经网络电路，且一个或多个连续顶点构成一个纵向分区部署单元（例如，一个容器）。或者，我们可以进一步对

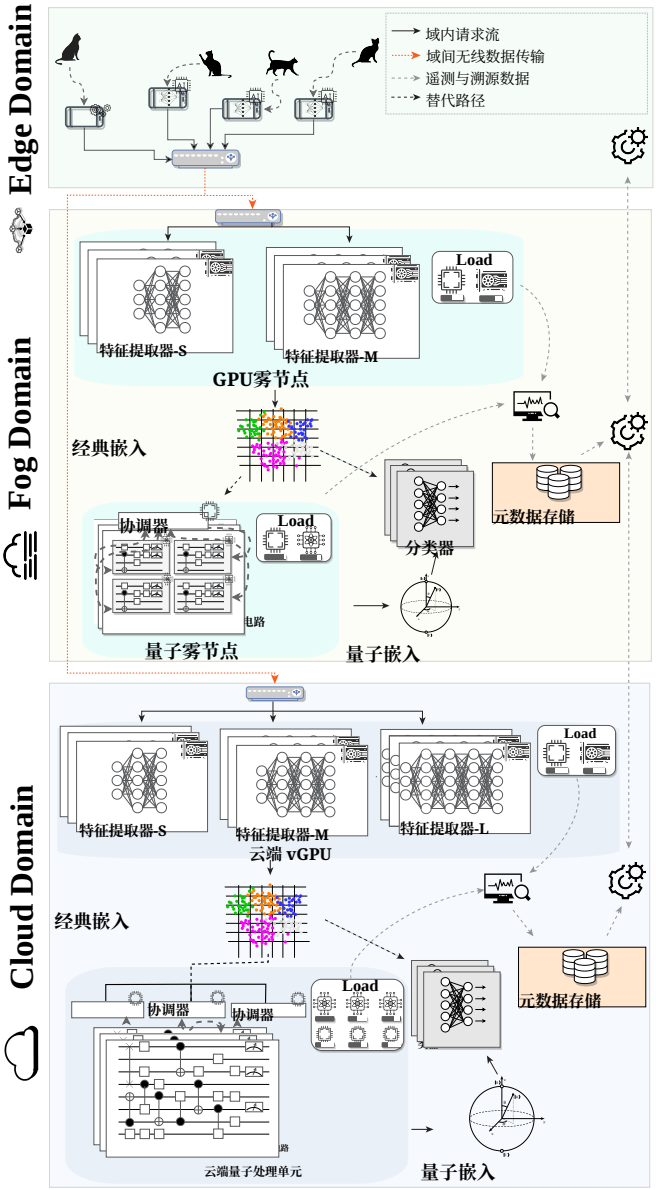

图 5. 推理引擎请求流

一个顶点进行分区，以创建一个横向分区部署单元（例如，通过电路切割）。值得注意的是，纵向方法定义了隔离计算节点，我们可以将其与横向方法透明地结合起来；也就是说，从外部视角来看，适当的抽象可以将切割电路呈现为单个粗粒度层。

客户端程序员可以通过注释向平台提供提示，但应用应当能够作为单个（单体）工作流进行部署。在生成 Quantum 和经典应用实例之前，创建部署单元是我们架构中控制平面的职责。从客户端程序员的角度来看，他们部署的是一个单一应用。然而，运行时系统应当知晓该应用被拆分为

多个部分。图 6 展示了一个粗粒度层计算图的示例。一个粗粒度

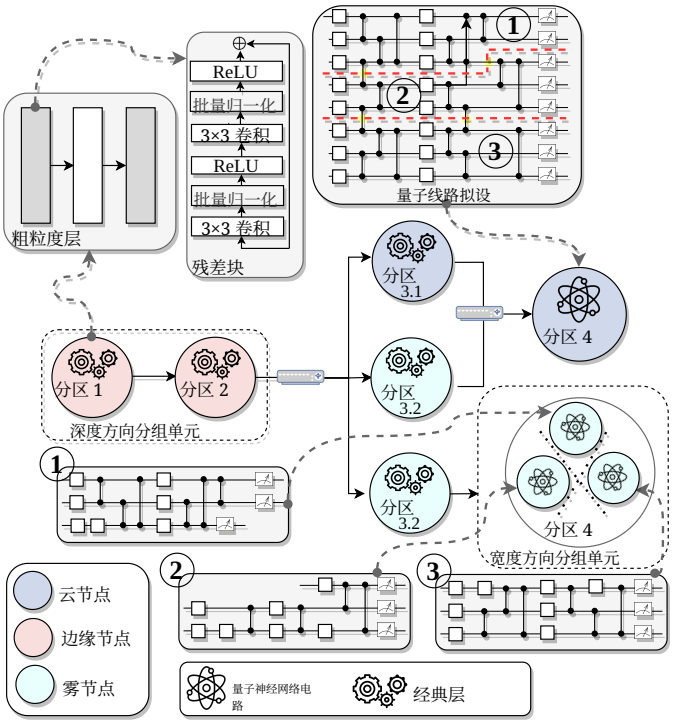

图 6. 粗粒度部署单元

层（递归地）由更细粒度的层组成。节点按处理顺序进行编号，子索引则表示分支路径。具有相同索引和子索引的节点表示部署在不同节点上的同一分区。分区 1–2 按深度方向进行分组，并作为一个整体部署在边缘设备上。分区 3.2 部署在云节点和雾节点上，而分区 3.1 则是一个不同的模型，仅部署在云节点上。例如，分区 3.1 和 3.2 可以分别对应图 5 中的特征提取器 L 和 M。具备 SLO 感知能力的负载均衡器将分区 2 的输出路由至分区 3 的某个变体。最后，分区 3 的输出被传递至分区 4 的某个实例。分区 4 部署在雾节点和云节点上。然而，它是一个云端量子处理器（QPU）能够执行的量子神经网络电路，但对于雾节点上资源受限的多量子处理单元（MQPUs），必须沿宽度方向进行分区（即按照第六节 B 部分所述，将其切分为多个子电路）。

VII. 采用经典-量子混合预测器的拆分推理

可以说，证明我们能够将经典深度神经网络划分方法扩展到经典—量子深度量子神经网络，是构建分布式混合推理引擎的必要前提。因此，为了展示我们构想的可行性，本节通过一种可分区神经网络架构，探讨第六节中的挑战 2)–4)（以及部分挑战 1)）；该架构允许弹性平面的运行时自由决定采用混合预测器还是经典预测器。

为鼓励后续研究，我们开放了一个存储库，其中提供了一个便捷框架，可用于添加采用新电路或模型实现的可配置实验。

A. 问题描述

正如第三节所倡导的，我们将一种最初为经典边缘计算设计的方法扩展到量子边缘计算，而不是忽视已有工作。具体而言，我们引入*QuantenSplit*，这是对拆分推理方法*FrankenSplit* [21]结合深度方向深度神经网络划分所做的简单修改，用于支持采用量子神经网络进行图像分类的量子嵌入。

我们选择 FrankenSplit，是因为它不局限于单一的头部-尾部网络对，因此，当后续两个请求需要不同的头部网络时，客户端设备无需更换头部权重。此外，FrankenSplit 通过变分特征压缩模型将本地资源集中用于降低带宽，即扩展该方法以支持量子神经网络可以应对第六节A部分中的挑战 3）和 4）。该方法借鉴信息瓶颈（IB） [63] 原理，在不牺牲预测能力的情况下，实现了远高于人工设计编解码器的压缩率。

我们省略形式化的概念细节，转而参考原始工作 [21]。在此，只需将 FrankenSplit 视为一个用于训练变分自编码器的框架，该自编码器在压缩过程中对所丢弃的信号具有特别强的选择性。值得注意的是，可以将用于多个下游任务的不同（拆分式）神经网络连接到自编码器上。换句话说，它允许平台向所有参与客户端部署通用编码器，而无需考虑它们各自的推理请求。

为了确定 FrankenSplit 是否能够推广到混合经典-量子神经网络预测器，我们回答以下问题：经典通用编码器高度压缩的特征能否推广到使用量子神经网络的下游任务？借此，我们旨在展示第四节所设想平台的实用性，并为第六节讨论的分布式推理引擎的可行性提供证据。

B. 方法论

为回答上述问题，我们必须证明可以将混合预测器和经典预测器连接到一对自动编码器-骨干网络。我们考虑四个数据集，共得到八个预测器；也就是说，对于每个数据集，我们都将一个混合分类模型和一个经典分类模型连接到基础网络。

1) 压缩和特征提取： 图7展示了模块化神经网络的架构。骨干网络是一个预训练拆分式经典神经网络，可进一步提取特征。可以连接多个不同大小（或架构）的骨干网络，使网络能够部署，如图5所示。不过，就我们的目的而言，只需将多个（混合）分类模型连接到单个预训练骨干网络即可。编码器仅依赖广泛支持的运算，以处理边缘领域中的设备异构性。括号中的数字表示阶段深度。解码器块可平滑处理

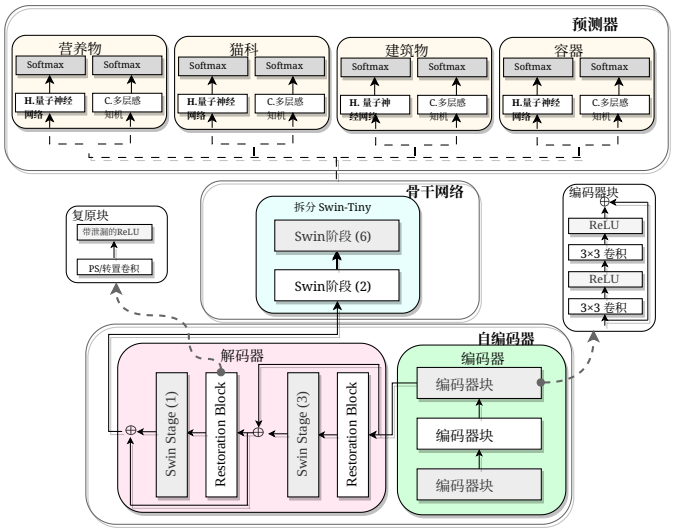

图 7. QuantenSplit网络架构

量化后的特征，并对特征进行变换以适配骨干网络。 [35], 描述了拆分后的 Swin-Tiny 骨干网络，但舍弃了前两个阶段。与原始工作不同的是，我们在经典分类模型之外，还添加了混合分类模型。

2) 分类模型： 经典分类器是一个简单的双层多层感知机（MLP）。图 8 展示了由四个量子比特和多个层构成的量子神经网络的 Ansatz。混合

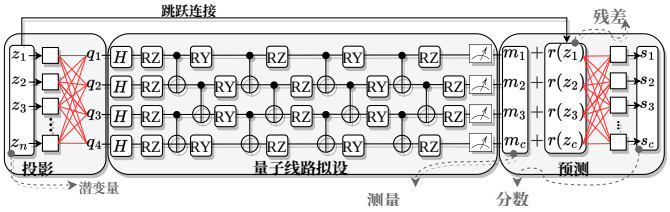

图 8. 具有四个量子比特和层的混合量子神经网络

QNN 模型以 n 维实值特征向量 $\mathbb{Z}^n$ 作为输入，并将其经典地投影为一个维度等于量子比特数量的向量。随后，它将这些特征作为输入传递给 Ansatz。无论电路深度如何，它都会先应用 Hadamard H 门和参数化的 Z旋转 RZ，将特征嵌入量子节点中。接下来，它会反复应用由可训练变分层组成的序列。每层由成对（移位）的 C-NOT 门组成，后接交替的参数化 Y旋转或 Z旋转，即包含 Y旋转的层后面接着包含 Z旋转的层。层数是由电路深度决定的超参数。最后，它将测量结果传递给全连接层，以输出各类别的得分。该跳跃连接的设计灵感来自经典残差神经网络 [29]。是否添加跳跃连接可配置为一个超参数。

C. 训练与实现

我们首先根据 FrankenSplit 的基线目标函数，分别在 1.28 Ima- geNet [55] 训练样本上训练自动编码器。然后，我们冻结参数，并使用交叉熵损失函数依次训练每个附加的分类模型。我们使用 Adam 优化 [30]来训练自动编码器和所有分类模型。采用宽度方向分区方法（例如电路切割）不在本文的研究范围内。有关训练参数的详细说明，请参阅随附存储库中的配置文件。

我们未对超参数调优或优化器相关实验进行穷尽性研究。我们使用 PyTorch [51] 和 CompressAI [5] 实现模型。骨干网络和预训练权重来自 PyTorch 图像模型 [68]。对于量子电路的数值模拟，我们使用 Pennylane [7]。为确保可复现性并便于后续工作，我们扩展了 torchdistill [40]。

D. 评估

我们的实验是在小规模模拟中进行的，类别数远少于原始工作。因此，为了从结果中得出有意义的见解，我们将经典模型的维度设为与其混合模型对应项的量子比特数相同；也就是说，多层感知机首先将高维骨干网络特征投影到低维经典嵌入。例如，在一个包含四个量子比特的实验中，我们将混合预测器与经典基线预测器进行比较，其中多层感知机的第一层将骨干网络特征投影到四维表示。为了模拟第五节中客户端请求的场景，

表 III ILSVRC 的训练集和测试集

Task	类别	训练样本	测试样本
营养物	10	13’000	500
猫科	13	16’900	650
建筑物	14	18’200	700
容器	23	29’900	1150

为了在不同环境中进行推理，我们从 ILSVRC 分类任务的 1000 个标签中创建了四个主题各异的数据集。表 III 总结了每个数据集，它们分别代表需要单独预测器的不同位置。随附的存储库包含用于重新创建这些数据集的脚本和说明。

我们以经典预测器为基线，分别使用 4、6、8 和 10 个量子比特进行实验。电路深度固定为 8。我们还进行了深度可变的补充实验，发现增加深度带来的准确率提升逐渐减小。表 IV 总结了我们的结果。图 9 和图 10 展示了在各数据集上，不带和带跳跃连接的混合模型与其经典对应模型的比较情况。相较于经典

表 IV 经典、混合、带残差的混合模型的 Top-1 错误率

	量子比特	经典模型 Top-1 错误率（%）	混合模型 Top-1 错误率（%）	Top-1 错误率 H. Res. (%)
营养物	4	13.00	37.13	12.11
	6	11.73	25.20	10.40
	8	11.30	16.90	10.07
	10	10.69	14.80	9.58
猫科	4	19.82	31.57	19.05
	6	17.60	29.07	16.77
	8	16.77	18.77	15.85
	10	16.56	18.31	15.31
建筑物	4	9.29	31.57	8.57
	6	7.26	14.86	6.86
	8	6.14	10.57	5.71
	10	5.74	9.00	5.27
容器	4	32.43	62.00	30.69
	6	27.82	48.52	26.00
	8	25.48	31.56	24.96
	10	24.26	25.87	23.91

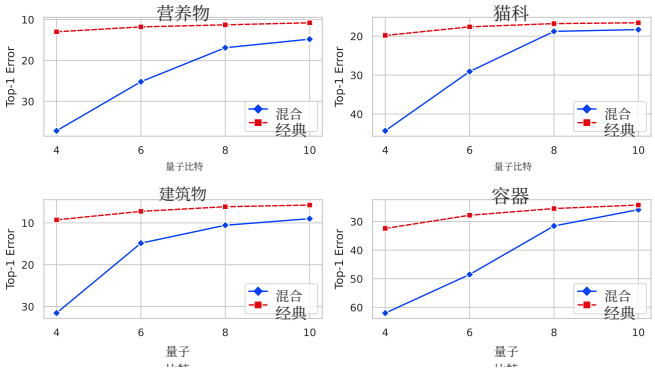

图 9. 不含跳跃连接的混合量子神经网络

基线，随着量子比特数量增加，不带跳跃连接的混合量子神经网络逐渐达到与经典基线相近、但仍较差的 Top-1 错误率。使用 2 个和 4 个量子比特时，Top-1 错误率约高出 20%–30%；而随着量子比特数量增加，差距缩小至 2%–5%。有趣的是，带跳跃连接的混合量子神经网络在所有量子比特数量下都 consistently 优于经典基线。尽管跳跃连接的设计初衷是

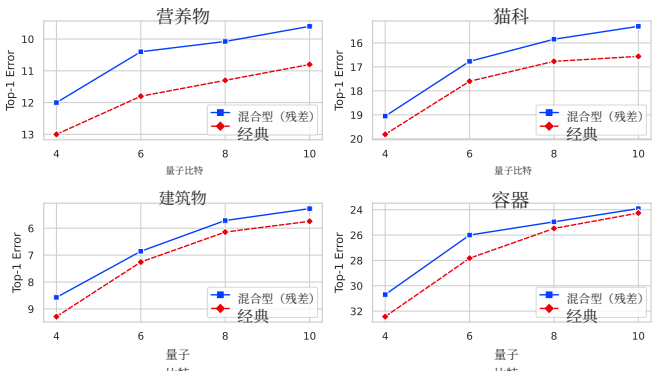

图 10. 带有跳跃连接的混合量子神经网络

经典残差网络旨在缓解超深神经网络的准确率饱和问题，本质上学习的是残差。

函数。考虑到自编码器-骨干网络对已经对输入数据进行了大量处理，我们假设，对于某些实例，QNN 能够为经典特征找到更合适的表示。相反，对于已经得到充分处理、适合分类的样本，QNN 可能会降低性能。随着量子比特数量的增加，QNN 缩小性能差距的现象与我们的假设一致。对于量子比特数量较少的情况，不带跳跃连接的模型无法找到与初始输入同样好的表示。相反，带有跳跃连接的 QNN 能够学习残差函数，并在量子嵌入发挥作用的样本上获得性能提升。

跳跃连接是我们首次基于直觉尝试提供实证证据，初步结果似乎颇有希望。不过，我们提醒读者，QuantenSplit 仅作为概念验证，用于判断我们的构想是否可行。评估策略还不够全面，因此不应将我们的结果视为定论。此外，即便加入跳跃连接，在所有任务的模拟中，混合模型在无噪声 设备上的表现也仅略胜于经典模型。我们没有尝试更适用于量子神经网络的优化算法，也没有投入大量精力设计合适的电路。此外，骨干网络是经典的，也就是说，提取出的特征可能偏向于经典预测器。未来的工作可以尝试更复杂的方法，将低维量子比特映射到高维特征空间，从而显著改善我们的结果 [50]。此外，一旦训练大型量子神经网络提取器成为可行之举，研究能否训练经典压缩模型，为量子嵌入找到更合适的表示，将会很有意义。

VIII. 结论

本文提出了将量子计算融入边缘-云连续体的愿景。我们总结了与本文工作相关的量子计算和经典计算现有文献，随后阐述了扩展现有经典系统以支持边缘的重要性。然后，我们介绍了一种混合经典-量子平台架构，并确定了集成量子处理单元所面临的关键挑战。我们致力于通过热启动和人工智能推理，促进分布式智能任务中量子应用程序的研究工作。最后，我们扩展了一种经典拆分推理方法，使其能够选择性地支持混合量子神经网络。我们的结果为这一愿景的可行性提供了实证依据，并表明我们的构想是值得研究的方向。

致谢

我们谨向 Fabian B¨uhler、Benjamin Weder、Vladimir Yussupov 和 Florian Kowarsch 致以谢意，感谢他们提供的宝贵意见。本研究部分由 BMWK 项目*PlanQK* (01MK20005N)、*EniQmA* (01MQ22007B) 和*Se-QuenC* (01MQ22009B) 资助。

参考文献

[1] Amira Abbas、David Sutter、Christa Zoufal、Aur´elien Lucchi、Alessio Figalli 和 Stefan Woerner。量子神经网络的力量。《自然·计算科学》，1(6):403–409，2021年。[2] Jordan Ash 和 Ryan P Adams。关于神经网络训练热启动。神经信息处理系统进展，33:3884–3894，2020年。[3] Mohammad S Aslanpour、Adel N Toosi、Claudio Cicconetti、Bahman Javadi、Peter Sbarski、Davide Taibi、Marcos Assuncao、Sukhpal Singh Gill、Raj Gaire 和 Schahram Dustdar。无服务器边缘计算：愿景与挑战。载于《2021年澳大拉西亚计算机科学周多会议》，第1–10页，2021年。[4] Marvin Bechtold、Johanna Barzen、Frank Leymann、Alexander Mandl、Julian Obst、Felix Truger 和 Benjamin Weder。研究电路切割对 NISQ 设备上最大割问题的量子近似优化算法的影响。arXiv 预印本 arXiv:2302.01792，2023年。[5] Jean B´egaint、Fabien Racap´e、Simon Feltman 和 Akshay Pushparaja。Compressai：用于端到端压缩研究的 PyTorch 库与评估平台。arXiv 预印本 arXiv:2011.03029，2020年。[6] Martin Beisel、Johanna Barzen、Simon Garhofer、Frank Leymann、Felix Truger、Benjamin Weder 和 Vladimir Yussupov。Quokka：用于基于工作流执行变分量子算法的服务生态系统。载于《面向服务的计算——ICSOC 2022 研讨会：ASOCA、AI-PA、FMCIoT、WESOACS 2022，西班牙塞维利亚，2022年11月29日至12月2日，论文集》，第369–373页。Springer，2023年。[7] Ville Bergholm、Josh Izaac、Maria Schuld、Christian Gogolin、Shahnawaz Ahmed、Vishnu Ajith、M Sohaib Alam、Guillermo Alonso-Linaje、B AkashNarayanan、Ali Asadi 等。Pennylane：混合量子经典计算的自动微分。arXiv 预印本 arXiv:1811.04968，2018年。[8] Kirill L Bogdanov、Waleed Reda、Gerald Q Maguire Jr、Dejan Kosti´c 和 Marco Canini。地理分布式存储系统的快速精准负载均衡。载于《ACM 云计算研讨会论文集》，第386–400页，2018年。[9] Dmytro Bondarenko 和 Polina Feldmann。用于量子数据去噪的量子自编码器。Physical Review Letters，124(13):130502，2020年。[10] Lukas Brenner、Christophe Piveteau 和 David Sutter。利用经典通信实现最优线路切割。2023年2月。[11] M Cerezo、Guillaume Verdon、Hsin-Yuan Huang、Lukasz Cincio 和 Patrick J Coles。量子机器学习的挑战与机遇。《自然·计算科学》，2(9):567–576，2022年。[12] Marco Cerezo、Andrew Arrasmith、Ryan Babbush、Simon C Benjamin、Suguru Endo、Keisuke Fujii、Jarrod R McClean、Kosuke Mitarai、Xiao Yuan、Lukasz Cincio 等。变分量子算法。《自然·物理评论》，3(9):625–644，2021年。[13] Yinpeng Chen、Xiyang Dai、Dongdong Chen、Mengchen Liu、Xiaoyi Dong、Lu Yuan 和 Zicheng Liu。Mobile-former：连接 MobileNet 与 Transformer。载于《IEEE/CVF 计算机视觉与模式识别会议论文集》，第5270–5279页，2022年。[14] Jaeho Choi 和 Joongheon Kim。量子近似优化算法（QAOA）教程：基础与应用。载于《2019年国际信息与通信技术融合会议（ICTC）》，第138–142页。IEEE，2019年。[15] Poulami Das、Swamit S Tannu、Prashant J Nair 和 Moinuddin Qureshi。多程序量子计算机的可行性探讨。载于《第52届IEEE/ACM 微体系结构国际研讨会年会论文集》，第291–303页，2019年。[16] Lei Deng、Guoqi Li、Song Han、Luping Shi 和 Yuan Xie。神经网络的模型压缩与硬件加速：全面综述。《Proceedings of the IEEE》，108(4):485–532，2020年。[17] Shuiguang Deng、Hailiang Zhao、Weijia Fang、Jianwei Yin、Schahram Dustdar 和 Albert Y Zomaya。边缘智能：边缘计算与人工智能的交汇。《IEEE 物联网期刊》，7(8):7457–7469，2020年。[18] Jay DGambetta。IBM 的量子技术扩展路线图。https://research.ibm.com/blog/ibm-quantum-roadmap。（访问日期：05.05.2023）。[19] Daniel J Egger、Jakub Mareˇcek 和 Stefan Woerner。量子优化的热启动。Quantum，5:479，2021年。[20] Alireza Furutanpey 和 Schahram Dustdar。自适应协同推理：迈向分布式智能系统的无妥协框架。2022年。

[21] Alireza Furutanpey、Philipp Raith 和 Schahram Dustdar。Frankensplit：基于显著性引导的神经特征压缩与浅层变分瓶颈注入。arXiv预印本 arXiv:2302.10681，2023年。[22] Alexey Galda、Xiaoyuan Liu、Danylo Lykov、Yuri Alexeev 和 Ilya Safro。随机图之间最优量子近似优化算法参数的可迁移性。载于《2021年IEEE量子计算与工程国际会议（QCE）》，第171–180页。IEEE，2021年。[23] Jose Garcia-Alonso、Javier Rojo、David Valencia、Enrique Moguel、Javier Berrocal 和 Juan Manuel Murillo。通过量子 API 网关实现量子软件即服务。IEEE Internet Computing，26(1):34–41，2021年。[24] Zacharias Georgiou、Moysis Symeonides、Demetris Trihinas、George Pallis 和 Marios D Dikaiakos。Streamsight：用于边缘计算中流式分析的查询驱动框架。载于《2018年IEEE/ACM第11届实用与云计算国际会议（UCC）》，第143–152页。IEEE，2018年。[25] DSaxonQ GmbH。SaxonQ Technology：金刚石中的量子比特。https://saxonq.com/technology。（访问日期：05.05.2023）。[26] Jianping Gou、Baosheng Yu、Stephen J Maybank 和 Dacheng Tao。知识蒸馏：综述。International Journal of Computer Vision，129:1789–1819，2021年。[27] Harper R Grimsley、Sophia E Economou、Edwin Barnes 和 Nicholas J Mayhall。一种用于在量子计算机上进行精确分子模拟的自适应变分算法。Nature Communications，10 3007，2019 1( ): 。[28] Michele Grossi、Luca Crippa、Antonello Aita、Giacomo Bartoli、Vito Sammarco、Eleonora Picca、N Said、Filippo Tramonto 和 Federico Mattei。用于量子计算的无服务器云集成。arXiv预印本 arXiv:2107.02007，2021年。[29] Kaiming He、Xiangyu Zhang、Shaoqing Ren 和 Jian Sun。用于图像识别的深度残差学习。载于《IEEE计算机视觉与模式识别会议论文集》，第770 778页，2016年 – 。[30] Diederik P Kingma 和 Jimmy Ba。Adam：一种随机优化方法。arXiv预印本 arXiv:1412.6980，2014年。[31] Frank Leymann 和 Johanna Barzen。关于NISQ时代基于量子门的量子算法的残酷真相。Quantum Science and Technology，5(4):044007，2020年。[32] Frank Leymann、Johanna Barzen、Michael Falkenthal、Daniel Vietz、Benjamin Weder 和 Karoline Wild。云端量子计算：应用潜力与研究机遇。arXiv预印本 arXiv:2003 06256，2020年 . .[33] Wenbin Li 和 Matthieu Liewig。边缘环境人工智能加速器综述。载于《信息系统与技术的趋势与创新：第2卷 8》，第35–44页。Springer，2020年。[34] Li Lin、Xiaofei Liao、Hai Jin 和 Peng Li。面向边缘计算的计算卸载。Proceedings of the IEEE，107 1607，2019年 8 1584( ): – 。[35] Ze Liu、Yutong Lin、Yue Cao、Han Hu、Yixuan Wei、Zheng Zhang、Stephen Lin 和 Baining Guo。Swin Transformer：采用移位窗口的分层视觉Transformer。载于《IEEE/CVF计算机视觉国际会议论文集》，第10012–10022页，2021年。[36] Seth Lloyd、Maria Schuld、Aroosa Ij az、Josh Izaac 和 Nathan Killoran。用于机器学习的量子嵌入。arXiv预印本 arXiv:2001.03622，2020年。[37] Quantum Brilliance Pty Ltd。Quantum Brilliance 硬件技术应用。https://quantumbrilliance.com/quantum-technology-application，2023年。访问日期：05.05.2023。[38] Johannes Manner、Martin Endreß、Tobias Heckel 和 Guido Wirtz。影响函数即服务冷启动的因素。载于《2018年IEEE/ACM实用与云计算国际会议伴会（UCC Companion）》，第181–188页。IEEE，2018年。[39] Andrea Mari、Thomas R Bromley、Josh Izaac、Maria Schuld 和 Nathan Killoran。经典-量子混合神经网络中的迁移学习。Quantum，4:340，2020年。[40] Yoshitomo Matsubara。torchdistill：用于知识蒸馏的模块化配置驱动框架。载于《可复现模式识别研究：第三届国际研讨会，RRPR 2021，线上会议，2021年1月11日，修订精选论文》，第24–44页。Springer，2021年。[41] Yoshitomo Matsubara、Marco Levorato 和 Francesco Restuccia。深度学习应用的拆分计算与提前退出：综述与研究挑战。ACM Computing Surveys，55(5):1–30，2022年。

[42] Kosuke Mitarai 和 Keisuke Fujii。通过对单量子比特操作进行采样来构造虚拟双量子比特门。New Journal of Physics，23(2):023021，2021年2月。[43] Charles Moussa、Henri Calandra 和 Vedran Dunjko。要不要使用量子计算：迈向近期量子优化中的算法选择。Quantum Science and Technology，5(4):044009，2020年10月。[44] Stefan Nastic、Andrea Morichetta、Thomas Pusztai、Schahram Dustdar、Xiaoning Ding、Deepak Vij 和 Ying Xiong。Sloc：面向下一代云计算的服务级别目标。IEEE Internet Computing，24(3):39–50，2020年。[45] Stefan Nastic、Philipp Raith、Alireza Furutanpey、Thomas Pusztai 和 Schahram Dustdar。面向边缘与云的无服务器计算架构。载于2022年 IEEE第四届认知机器智能国际会议（CogMI），第1–12页。IEEE，2022年。[46] Ripal Nathuji、Aman Kansal 和 Alireza Ghaffarkhah。Q-clouds：管理服务质量感知云中的性能干扰效应。载于第五届欧洲计算机系统会议论文集，第237–250页，2010年。[47] Hoa T Nguyen、Muhammad Usman 和 Rajkumar Buyya。Qfaas：面向量子计算的无服务器函数即服务框架。arXiv预印本 arXiv:2205.14845，2022年。[48] Yasuhiro Ohkura、Takahiko Satoh 和 Rodney Van Meter。在当前及近期的 NISQ系统上同时执行量子电路。IEEE Transactions on Quantum Engineering，3:1–10，2022年。[49] Tianyi Peng、Aram Harrow、Maris Ozols 和 Xiaodi Wu。在小型量子计算机上模拟大型量子电路。Physical Review Letters，125:150504，2019年3月。[50] Evan Peters、Jo˜ao Caldeira、Alan Ho、Stefan Leichenauer、Masoud Mohseni、Hartmut Neven、Panagiotis Spentzouris、Doug Strain 和 Gabriel N Perdue。在含噪量子处理器上对高维数据进行机器学习。npj Quantum Information，7(1):161，2021年。[51] PyTorch中的自动微分。PyTorch，2018年。[52] Nils Quetschlich、Lukas Burgholzer 和 Robert Wille。预测良好的量子电路编译选项。arXiv预印本 arXiv:2210 08027，2022年。。[53] Philipp Raith、Thomas Rausch、Schahram Dustdar、Fabiana Rossi、Va-leria Cardellini 和 Rajiv Ranjan。跨边缘-云连续体的移动性感知无服务器函数适配。载于2022年 IEEE/ACM第十五届效用计算与云计算国际会议（UCC），第123–132页，2022年。[54] Thomas Rausch、Waldemar Hummer、Christian Stippel、Silvio Vasiljevic、Carmine Elvezio、Schahram Dustdar 和 Katharina Kr¨osl。迈向智慧城市规模的认知辅助应用平台。载于2021年 IEEE虚拟现实与三维用户界面摘要及研讨会会议（VRW），第330–335页。IEEE，2021年。[55]Olga Russakovsky、Jia Deng、Hao Su、Jonathan Krause、Sanjeev Satheesh、Sean Ma、Zhiheng Huang、Andrej Karpathy、Aditya Khosla、Michael Bernstein 等。ImageNet大规模视觉识别挑战赛。International Journal of Computer Vision，115 252，2015年3月 211( ): – 。[56] Marie Salm、Johanna Barzen、Uwe Breitenb¨ucher、Frank Leymann、Benjamin Weder 和 Karoline Wild。NISQ分析器：自动为量子算法选择量子计算机。载于《面向服务的计算：第14届面向服务计算研讨会与暑期学校，SummerSOC 2020》，希腊克里特岛，2020年9月13日至19日，第14卷，第66–85页。Springer，2020年。[57] Wonik Seo、Sanghoon Cha、Yeonjae Kim、Jaehyuk Huh 和 Jongse Park。面向边缘平台异构处理器的服务级别目标感知推理调度器。ACM Transactions on Architecture and Code Optimization（TACO），18(4):1–26，2021年。[58] Kunal Sharma、M. Cerezo、Zo¨e Holmes、Lukasz Cincio、Andrew Sorn-borger 和 Patrick J. Coles。针对纠缠数据集重新表述无免费午餐定理。Physical Review Letters，128:070501，2022年2月。[59] Ruslan Shaydulin 和 Yuri Alexeev。评估量子近似优化算法：案例研究。载于2019年第十届国际绿色可持续计算会议（IGSC），第1–6页。IEEE，2019年。[60] P.W. Shor。量子计算算法：离散对数与因数分解。载于第三十五届计算机科学基础年度研讨会论文集，第124–134页，1994年。[61] Yushan Siriwardhana、Pawani Porambage、Madhusanka Liyanage 和 Mika Ylianttila。基于5G移动边缘计算的移动增强现实综述：架构、应用程序与技术方面。IEEE Communications Surveys & Tutorials，23(2):1160–1192，2021年。

[62] Jules Tilly、Hongxiang Chen、Shuxiang Cao、Dario Picozzi、Kanav Setia、Ying Li、Edward Grant、Leonard Wossnig、Ivan Rungger、George H Booth 等。变分量子本征求解器：方法与最佳实践综述。*Physics Reports*，986:1–128，2022年。[63] Naftali Tishby、Fernando C. Pereira 和 William Bialek。信息瓶颈方法，2000年。[64] Felix Truger、Johanna Barzen、Marvin Bechtold、Martin Beisel、Frank Leymann、Alexander Mandl 和 Vladimir Yussupov。热启动与量子计算：系统性映射研究。*arXiv* 预印本 *arXiv:2303.06133*，2023年。[65] Benjamin Weder、Johanna Barzen、Frank Leymann、Marie Salm 和 Karoline Wild。Qprov：量子计算溯源系统。*IETQuantumCommunication*，2(4):171–181，2021年。[66] Benjamin Weder、Johanna Barzen、Frank Leymann 和 Michael Zimmermann。混合量子应用需要叠加态中的两种编排：软件架构视角。载于《*18*[*l*]*IEEE* 国际 *Web* 服务会议（*ICWS 2021*）论文集，第1–13页。IEEE，2021年。[67] Manuela Weigold、Johanna Barzen、Frank Leymann 和 Daniel Vietz。混合量子算法模式。载于《面向服务的计算：第*15*届研讨会与暑期学校，*SummerSOC 2021*，线上会议，*2021*年*9*月*13*日至*17*日，论文集*15*》，第34–51页。Springer，2021年。[68] Ross Wightman。Torch 图像模型。https://github.com/rwightman/pytorch-image-models，2019年。[69] Karoline Wild、Uwe Breitenb¨ucher、Lukas Harzenetter、Frank Leymann、Daniel Vietz 和 Michael Zimmermann。Tosca4qc：使用 TOSCA 自动化量子应用的部署与编排的两种建模风格。载于*2020*年*IEEE*第*24*届国际企业分布式对象计算会议（*EDOC*），第125–134页。IEEE，2020年。[70] Michael Wurster、Uwe Breitenb¨ucher、K´alm´an K´epes、Frank Leymann 和 Vladimir Yussupov。使用 TOSCA 对无服务器应用进行建模和自动化部署。载于*2018*年*IEEE*第*11*届面向服务计算与应用会议（*SOCA*），第73–80页。IEEE，2018年。[71] Zhuoran Zhao、Kamyar Mirzazad Barijough 和 Andreas Gerstlauer。DeepThings：资源受限物联网边缘集群上的分布式自适应深度学习推理。*IEEE*集成电路与系统计算机辅助设计汇刊，37(11):2348–2359，2018年。