Non-Centralized Quantum Neural Networks for Cell-Free MIMO Systems

Bhaskara Narottama, *Member, IEEE,* Berk Canberk, *Senior Member, IEEE,* Simon L. Cotton, *Fellow, IEEE,* Hyundong Shin, *Fellow, IEEE,* George K. Karagiannidis, *Fellow, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—This paper propose a two-stage quantum neural network (QNN) framework for cell-free multiple-input and multiple-output (MIMO) wireless communication systems. Cell-free MIMO, which has been regarded as a key technology for enhancing the performance of the next-generation wireless communication systems, leverages the collective capability of multiple distributed access points (APs), allowing collaboration between them. However, optimizing cell-free MIMO can pose challenges for centralized optimization schemes. In particular, complexities associated with the joint optimizations of user-transmission assignment and transmission precoding, two factors which are of much importance for determining the quality-of-service, grow with the number of APs and served users. To this end, a unified scheme employing distributed QNNs is used to optimize downlink transmitter-user assignment and transmit precoding with the goal of maximizing the achieved sum rate. Firstly, the cloud processing unit, which holds holistic information about the particular wireless communication network, employs QNN to assign each AP to its designated mobile terminal. Secondly, the edge processing units, which are computed in proximity relative to the AP in order to reduce latency, estimate transmission precoding for their corresponding APs. Moreover, numerical results are presented to showcase the performance of the proposed protocol.**

***Index Terms*—Cell-free MIMO; quantum machine learning; quantum neural networks.**

B. Narottama is with the Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL A1C 5S7, Canada (e-mail: bnarottama@mun.ca).

B. Canberk is with the School of Computing, Engineering and The Build Environment, Edinburgh Napier University, United Kingdom (e-mail: b.canberk@napier.ac.uk).

S. L. Cotton is with the Centre for Wireless Innovation (CWI), School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT3 9DT, Belfast, U.K. (email: simon.cotton@qub.ac.uk).

H. Shin is with the Department of Electronics and Information Convergence Engineering, Kyung Hee University, 1732 Deogyeong-daero, Giheung-gu, Yongin-si, Gyeonggi-do 17104, Republic of Korea (e-mail: hshin@khu.ac.kr).

G. K. Karagiannidis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece and also with Artificial Intelligence Cyber Systems Research Center, Lebanese American University (LAU), Lebanon (e-mail: geokarag@auth.gr).

T. Q. Duong is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, and with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, BT7 1NN Belfast, U.K., and also with the Department of Electronic Engineering, Kyung Hee University, Yongin-si, Gyeonggi-do 17104, South Korea (e-mail: tduong@mun.ca).

This paper was presented in part at IEEE Global Communications Conference, December 2022, Rio de Janeiro, Brazil [1].

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, and in part by the NSERC CREATE program (Grant number 596205-2025). The work of B. Canberk is supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab Project 5239903. The work of S. L. Cotton was supported by the U.K. Engineering and Physical Sciences Research Council (EPSRC) through the EPSRC Hub on All Spectrum Connectivity under Grant EP/X040569/1 and Grant EP/Y037197/1. The work of H. Shin was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) under RS-2025-00556064 and RS-2025-25442355.

Corresponding authors are Trung Q. Duong and Hyundong Shin.

I. Introduction

CELL-FREE multiple-input and multiple-output (MIMO), which allows multiple access points (APs) to be operated in cooperation removing cell boundaries, is regarded as one of the key technologies to enhance the coverage and performance exceeding conventional MIMO networks [2]. Cell-free MIMO allows a number of distributed APs, that are spatially dispersed throughout a particular deployment area, to cooperatively serve a number of users, possibly using the same time and frequency resource [3], [4]. Various studies have demonstrated the benefits of cell-free networks, including flexibility in deployment [5], enhanced data rates [6], delivering effective spatial multiplexing, among other advantages, while its possible adoptions includes distributed remote radio heads (RRHs), low-energy access points, and even cooperation with non-terrestrial networks [4]. To realize their full capability, cell-free networks need to be managed properly, to avoid excessive inter-AP interference, and maximize receivable signal gain [7]. In this regard, the optimizations of transmitter-user assignment and transmission beamforming will be critical for ensuring the optimal performance of cell-free MIMO in 6G [8], [9]. However, As cell-free MIMO networks grow in scale, it will become more challenging to provide analytical-based optimal solutions [2], as we now need to address joint transmissions with multiple APs, where each AP operates a number of multiple antennas [10]. This complicates tracing solutions for cell-free optimizations, typically presented as non-convex problems, since added variables might occasion further analysis. While iterative approaches exist, e.g., those leveraging successive convex approximation [11], this issue persists, especially when optimizing multiple cell-free transmitters [12].

To address this issue, prior studies have introduced various optimization approaches based on artificial intelligence (AI) [9], [13]. AI estimates solutions through learning from acquired data relevant to cell-free systems, e.g., user channel information.[1] Notably, those based on neural networks (NNs) provide non-linear responses given particular inputs and, thereby, have been used to optimize different aspects of cell-free MIMO,

[1]AI's potential in automating and optimizing future wireless operations is evident in various global initiatives, e.g., [14].

e.g., beamforming [15] and power allocations [16]. A typical neural network model consists of a number of different layers, each of which is composed of a number of perceptrons. To provide desired estimations, the parameters of the perceptrons need to be optimized under a training algorithm, using a sufficient amount of data features. Typically, the computational complexity of a classical-based neural network, which is often related to its processing time, grows with the number of layers and neurons. When adopted to optimize wireless systems, these numbers are strongly linked to the numbers of APs, users, and active antennas [17]. As a result, aggressively scaling the size of cell-free networks might result in excessive training durations, introducing significant latencies into the network operation. Unsurprisingly, efforts have been made to reduce the computational complexity of the neural networks used to optimize wireless communications, without significantly compromising their ability to make accurate predictions. For instance, in [18], an attention mechanism was used to reduce the input dimension of the considered NN, that was used to model wideband power amplifiers, while neural network pruning was leveraged in [19], to reduce redundant filters in the convolutional NN that was used for precoding optimization.

Nevertheless, such solutions, while beneficial, are primarily centered on refining existing classical NNs. This motivates us to investigate an alternative computational platform for facilitating NNs for cell-free system optimizations. Previous studies have primarily relied on classical computing, which processes binary bits sequentially, with each bit having a value of either 0 or 1, for executing NNs on such tasks, e.g., [20] and [21]. Quantum computing has recently attracted unprecedented research attention and significant investments [22], [23]. Unlike classical bits, which are restricted to representing either 0 or 1, quantum bits, known as "qubits," can exist in a superposition of both $|0\rangle$ and $|1\rangle$. This property allows a register of ϕ qubits to span 2^{ϕ} computational basis states simultaneously. In contrast, ϕ classical bits can only represent one of those 2^{ϕ} states at a time, highlighting the potential exponential advantage of quantum computing in certain tasks.[2]

Exploiting the computational advantages offered by quantum computations [28], [29], quantum neural network (QNN) –based schemes can be employed to optimize cell-free wireless networks [30]. In contrast with classical NNs, QNNs employ quantum circuits, each of which assembles different quantum operations, to represent a trainable NN. Specifically, each qubit can represent a neuron, from a set of which a layer is assembled. Parameterized quantum logic gates, e.g., Y-rotation gates, therein represent trainable operations, while controlled gates, e.g., controlled-Z gates, facilitate non-linear outputs (Particular usages of these gates are elaborated most thoroughly in subsections IV-A and IV-B.). Even though they are in their infancy, studies have already demonstrated the merits of QNNs, for example facilitating training convergence [31] and learning model expressivity [32], reducing prediction errors [33], and handling large amounts of data, even with noisy quantum processors [34], while others reported their ability to improve prediction performance [28] and reduce complexity [35], which can be attributed to the quantum parallelism and superposition [29], highlighting their utility in improving wireless system performance [36]. Nevertheless, despite its attractive features, the implementations of QNNs for optimizing cell-free systems, as of the time of writing, has not been reported in the literature, as discussed in the following.[3]

A. Related Studies

In order to highlight the significance of employing quantum-based NNs for optimizing cell-free MIMO systems, the following discussion covers some of the key studies in NN and machine learning –based cell-free MIMO optimization, while Table I summarizes their findings. A non-centralized approach to precoding optimization has been demonstrated in [10]. In this work, an actor-critic method with distributed actor networks contributing to an aggregated experience was used to maximize the achieved sum rate. In [39], a graph neural network was used for transmitter assignment in a cell-free network. In [9], an NN-based scheme was used to optimize transmitter-user assignment. In [40], a machine learning –based optimization was utilized to optimize transmit scheduling in cell-free MIMO. In addition, the authors of [41] utilized the deep deterministic policy gradient to optimize transmit power coefficients for the purpose of maximizing the weighted sum of achieved energy efficiency. In [20], a NN was used for transmitter scheduling in uplink cell-free MIMO systems. In [21], a cascaded deep learning method was used for channel interpolation in a cell-free system. However, quantum-based neural networks were not considered in these studies.[4]

Besides, recent studies have exploited quantum-based learning solution to optimize wireless communications. In [44], hybrid quantum-classical NNs were employed to maximize user sum rate, through transmit precoding optimization. The study of [45] develops a hybrid autoencoder architecture to optimize end-to-end communication systems, improving block error rate (BLER). In [46], federated quantum NNs were adopted for satellite ground communications, demonstrating improved learning capabilities. In [47], quantum-based machine learning

[2]We are witnessing rapid progress in the developments of quantum computing backends, platforms, strongly advocating for their adoption in wireless systems: Of note, the number of qubits within the quantum computing platforms grows each year, e.g., IBM's platforms featured 127 qubits in 2021, 433 qubits in 2022, and 1,121 qubits in 2023 [24]. Not to mention, error mitigation approaches have been extensively developed to support their near-term applications [25], e.g., those based upon virtual distillation, zero noise extrapolation, and even, machine learning [26]. Such advances have attracted significant government backing: Among others, the Canadian government has committed around $360 million, as of 2021, through its National Quantum Strategy, while the Taiwanese government has invested roughly $282 million to support quantum technology developments, as of 2022 [27].

[3]In fact, there is a growing interest in adopting quantum-enabled learning for optimizing wireless systems. In particular, [37] uses a variational quantum algorithm within a federated learning workflow, processed over space-air-ground integrated networks (SAGIN), while [38] highlights its versatility across learning paradigms (supervised, unsupervised, and reinforcement learning) for various 6G use cases, such as indoor localization. Nevertheless, this study is an early attempt at adopting quantum-enabled learning for cell-free systems.

[4]Studies have also presented the use of model learning-based approaches to optimize coherent joint transmissions. In particular, the study in [42] employs a graph convolutional network-based scheme to maximize weighted sum rates, while [43] adopts deep Q-network to optimize mode selection for such transmissions. Nonetheless, these studies are confined to classical-valued learning approaches, and this paper addresses this gap by utilizing non-centralized QNNs.

is employed for wireless optimization. In [35], QNN schemes were utilized for user grouping in non-orthogonal multiple access (NOMA). In [48], quantum-based learning was used for ultra-reliable low-latency communications (URLLC). In [49], a quantum-inspired reinforcement learning was utilized for UAV-enabled wireless network. However, cell-free MIMO optimization using QNNs has not considered in the mentioned works. Not to mention, QNNs have demonstrated their applicability across other fields, e.g., healthcare [50] and renewable energy [51]. Lastly, a number of surveys have also covered possible quantum-based approaches that can be applied to wireless applications, as outlined in Table II. In especial, various QNN architectures are discussed in [22], while [52] inquires into the applicability of quantum-based approaches for wireless systems.

B. The Main Contributions

Based on the discussion above, the usage of QNN-based methods for the optimizations of cell-free systems, remains largely unexplored, despite the potential benefits of leveraging QNNs. Hence, in this paper, we propose a non-centralized, multi-stage QNN protocol that maximizes the performance of cell-free systems. The main contributions of this study can be summarized as follows.

- This study proposes the utilization of QNNs to optimize cell-free systems, by specifically using a QNN-based framework for transmitter-user assignment and transmit precoding optimization, in order to maximize the minimum rate among all users, which is a non-convex NP-hard problem, that prohibits tracing optimal solutions, as discussed later in subsection III-B. QNNs and other parameterized quantum models have been shown to offer many benefits, e.g., higher effective dimension, trainability, and expressivity [55], [56], attributed to their ability to map data points to higher-dimensional Hilbert spaces, and have thus been effective across diverse applications, e.g., facilitating reusable rockets' stabilized landing [57], detecting financial fraud [58], and identifying brain tumors [59]. Yet, to the best of the authors' understanding, this is an earlier study that adopts QNNs for cell-free systems. Among the related studies, [44] and [60] employ hybrid quantum-classical NNs and quantum annealing –based approaches, respectively, yet neither considers cell-free systems, and both are focused solely on transmit precoding.
- In addition, to distribute the computational burden, QNN models are processed in a non-centralized manner: the central QNN model is computed by the cloud computing unit, while edge QNN models are processed by their respective edge units.[5] Such an approach allows the cloud to administer network-wide tasks, e.g., assigning different APs to users in a cell-free network, while each edge focuses upon AP-specific problems, e.g., optimizing its associated AP's transmit precoding.[6] While non-centralized QNNs are increasingly relevant for alleviating burden on the central processing unit, particularly amid the growing number of network elements, e.g., in ultra-dense networks, their adoptions in cell-free systems, and even wireless systems in general, is scarce.[7] In particular, [47] and [35] take no account of non-centralized QNNs.
- To analyze how well the prediction model of the proposed QNN-based framework can learn from the training data, the trainability of the considered QNN model is investigated in terms of the gradient norm's upper and lower bounds, given the defined loss function, as well as the model's expressivity.
- Moreover, to analyze the required computational resources to implement the QNN models, the required number of qubits is also investigated. Furthermore, to demonstrate the effectiveness of the proposed QNN framework for optimizing cell-free systems, the numerical results of the proposed QNN-based approach are compared with those of a classical optimization algorithm.[8]

Organization: The rest of this paper is summarized as follows. In Section II, the proposed QNN-based protocol is introduced. Section III describes the assumed cell-free system and the optimization objective. Detailed descriptions of the cloud and edge QNNs can be found in Section IV. Section V presents the numerical results. Section VI concludes this study. *Notation:* The symbols $\mathbb{R}$ and $\mathbb{C}$ represent real and complex numbers, respectively, with $\mathsf{i} = \sqrt{-1}$. The notations $\|\cdot\|_2$ and $|\cdot|$ represent the l2 norm and absolute value, respectively. Conjugate, transpose, and conjugate transpose operations are denoted by the symbols $(\cdot)^*$, $(\cdot)^\mathsf{T}$ and $(\cdot)^\dagger$, respectively. Regarding the quantum-based operation, the Hadamard gate, controlled-X gate, controlled-Y gate, and Z-rotation gate are represented by $\mathbf{H}$, $\mathbf{C}_\mathrm{X}$, $\mathbf{C}_\mathrm{Y}$, and $\mathbf{R}_\mathrm{Z}$, respectively. Pauli operators associated with the X-, Y-, and Z- axes are denoted by σ_x, σ_y, and σ_z, respectively. span$(\cdot)$ and dim$(\cdot)$ convey the span of a set of vectors and the dimension of a particular vector space, respectively.

II. The Proposed Multi-Stage QNN Protocol

In the sequel, we showcase the generic framework of the two-stage QNN framework. The unified protocol of the proposed framework is then illustrated in Fig. 1.

[5]For simplicity, the cloud and the edge computing units will hereinafter be referred to as "the cloud" and "the edge". Furthermore, after the cloud and edge QNNs are trained, solution inferences are executed first on the cloud QNN, followed by the edge QNNs (detailed later in Algorithm 1). Such a two-step approach is intended to minimize large dataset uploading, e.g., that of channel information, sustaining practical deployments. In the future, it is possible to explore dimension reductions of channel information, facilitating iterative optimizations between the cloud and edges.

[6]This approach supports bi-level optimizations, allowing the cloud and edges to pursue distinct objectives [61], discussed further in section II-A. In addition, though designing hybrid quantum-classical NN frameworks is reserved for future work, it also accommodates the possibility of certain edges in the network employing classical NN instead.

[7]Though this study considers the cloud as the central processing unit, its not restrictive, for core processing units of centralized radio access networks (C-RANs) can also fulfill such function, among other alternatives.

[8]The comparative method examines the transmission assignment solution within the solution space (see Appendix C) using a heuristic approach, making it an effective basis for comparison.

Table I: Related Studies

Reference	Employs QNN	Considers Cell-Free MIMO	Takeaways
[10]	-	✓	non-centralized RL for cell-free MIMO precoding
[39]	-	✓	graph NN for cell-free transmitter assignment
[9]	-	✓	NN-based scheme for transmitter-user assignments
[40]	-	✓	ML-based scheme for optimizing cell-free MIMO
[41]	-	✓	DDPG to optimize transmit power in cell-free MIMO
[20]	-	✓	NN for uplink cell-free MIMO
[21]	-	✓	cascaded deep learning for channel prediction in cell-free system
[47]	✓	-	employs quantum ML for wireless optimization
[35]	✓	-	QNN schemes for wireless optimization
[48]	✓	-	quantum-based learning for URLLC
[49]	✓	-	quantum-inspired reinforcement learning
this work	✓	✓	Non-centralized QNNs for transmitter-user assignment and precoding optimizations in cell-free MIMO

Table II: Releted Surveys on QNNs and Quantum Technologies

Reference	Takeaways
[53]	motivation for using quantum-based ML in 6G, covers several Quantum-based ML frameworks for 6G
[23]	covers general quantum-based ML algorithms and popular quantum platforms
[54]	discusses the possibility of using quantum-based scheme for 6G
[52]	examples of using quantum algorithms for wireless communications
[22]	discusses various quantum machine learning architectures

A. Generalized Problem

We assume that the proposed protocol can handle joint optimization problems: $\mathcal{P}_1, \mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$. $\mathcal{P}_1$ is a higher-level problem, which is solved earlier. $\mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$ is a set of a similar lower-level problems that will be solved after $\mathcal{P}_1$. Accordingly, such a problem resembles a tree optimization problem; solving $\mathcal{P}_1$ yields solution a_1, which is subsequently served as a given condition for $\mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$ [62]. Furthermore, the optimization variable for different stages of optimization can be described as follows. The upper-level problem assumes solution vector v_1^*, which is obtained as

$$v_1^* = \arg\max_{v_1} f_1(v_1|x), \tag{1}$$

where f_1 indicates the objective of $\mathcal{P}_1$. Later, Eq. (1) will be manifested as Eq. (8). Subsequently, the solution vector of a particular lower-level problem i, can be presented as follows:

$$v_{2,i}^* = \arg\max_{v_{2,i}} f_{2,i}(v_{2,i}|v_1^*, x_{2,i}), \quad i \in \{1, \cdots, N\}, \tag{2}$$

where $f_{2,i}$ is the objective of the $\mathcal{P}_{2,i}$. Later, Eq. (2) will be applied as Eq. (15). Beyond the problem addressed in this paper, later described in Subsection III-B, $\mathcal{P}_1$ and $\mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$ can also represent the optimization problems across different layers of physical network architectures, including SAGIN [38]. For instance, $\mathcal{P}_1$ of the satellite layer can be addressed first, followed by $\mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$ of the high-altitude platform systems (HAPS) layer. Such an approach also extends to virtual architectures, where $\mathcal{P}_{2,1}, \cdots, \mathcal{P}_{2,N}$ correspond to the problems of distinct network slices, each with its own optimization variables. This study adopts the above concept to handle AP assignment and precoding optimizations. Specifically, the AP assignment problem is presented as the upper-level problem, while the transmit precoding optimizations are designated as lower problems. Following this notion, the problem formulation is described later in Section III-B. The multi-stage workflow allows optimization at the cloud to shape the performance of N_{AP} number of APs, regardless of their number, making it relevant to the anticipated growth in network density, while the non-centralized approach allows those APs to address their local optimization problems, which can be treated as sub-problems, as expressed later in Eq. (15).

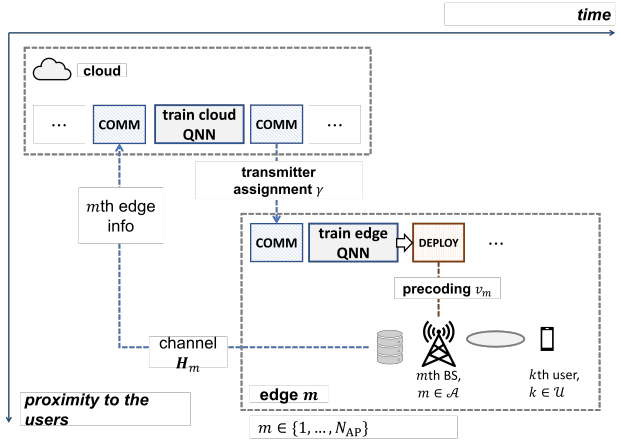

Figure 1: The proposed protocol employing multi-stage QNNs. For global objectives, e.g., to maximize the minimum rate among all users, see Eq. (8), the cloud responsible for network-level optimizations, e.g., transmitter-user assignment, see Eq. (8). Upon receiving the cloud's transmitter-user assignment solution, each edge optimizes the variables of its corresponding AP, e.g., its transmit precoding, to meet local objectives, such as maximizing the sum rate of the users assigned to it, see Eq. (15), aligning with the global objective.

B. The Training and Deployment Phases

The proposed protocol is described in Algorithm 1. The QNN-based learning models for the cloud and the m-th

edge, are employed therein, presented as U^{cloud} and $U^{[m]}$, respectively. QNN models leverage quantum computing, whose benefits are showcased as follows. The foundation of quantum computing, the quantum bit (qubit) can process information as a superposition of the basis states $|0\rangle$ and $|1\rangle$, given by $|\psi\rangle = p_0 |0\rangle + p_1 |1\rangle$, where $|p_0|^2$ and $|p_0|^2$ are pertinent to the probabilities of acquiring state $|0\rangle$ and $|1\rangle$, respectively. Still, the former illustration only considers a single qubit. Given M as the number of qubits with $|0\rangle$ and $|1\rangle$ as the computational basses, we can process 2^M states simultaneously; assuming M qubits, the generalized quantum state can then be expressed as $|\psi\rangle = \sum_{n=0}^{2^M-1} p_n |n\rangle$, where $|n\rangle$ denotes the n-th basis that can be obtained with a probability of $|p_n|^2$.

Different Phases in the Introduced Protocol: Cloud QNN Training: In Algorithm 1, the first phase is intended to train the cloud QNN. To do so, each m-th AP, $m \in \{1, \cdots, N_{\text{edge}}\}$ sends its user channel information, e.g., $\hat{\mathbf{H}}_m$, to the cloud. Next, the cloud trains the cloud QNN using channel information $\hat{\mathbf{H}} = \{\hat{\mathbf{H}}_m\}_{m=1}^{N_{\text{edge}}}$ to optimize its parameter vector θ^{cloud}. Edge QNN Training: The next phase trains the QNNs used to optimize transmit precoding. First, each m-th AP, $m \in \{1, \cdots, N_{\text{edge}}\}$ sends its channel information $\hat{\mathbf{H}}_m$ to the cloud. The cloud then employs optimized U^{cloud} to estimate AP assignment γ. Subsequently, the cloud broadcasts the optimized γ to all N_{edges} edges. Upon receiving γ, the training processes on the m-th edge is performed. Finally, we now have optimized θ^{cloud} and $\theta^{[m]}$ for U^{cloud} and $U^{[m]}$, respectively, $\forall m \in \{1, \cdots, N_{\text{AP}}\}$. Deployment: The deployment phase, which considers trained $U^{\text{cloud}}(\theta^{\text{cloud}})$ and $U^{[m]}(\theta^{[m]})$, $\forall m \in \{1, \cdots, N_{\text{AP}}\}$, can be described as follows. First, each m-th edge transmits channel information $\hat{\mathbf{H}}_m$ to the cloud. Subsequently, using the trained $U^{\text{cloud}}(\theta^{\text{cloud}})$, γ is estimated. Eventually, given trained $U^{[m]}(\theta^{[m]})$, each m-th AP leverages estimated γ and $\hat{\mathbf{H}}_m$ to estimated $\mathbf{v}_m$. Each AP then serves its designated user(s) according to γ, using transmit precoding $\mathbf{v}_m$.

III. The System Model

The study assumes a cell-free wireless communication system comprising of N_{AP} APs, utilized serving the downlink communications for N_{user} users, while the sets of available APs and users can be presented as $\mathcal{A} = \{a_m\}_{m=1}^{N_{\text{AP}}}$ and $\mathcal{U} = \{u_k\}_{k=1}^{N_{\text{user}}}$, respectively, and each AP is fitted with N_{Tx} antennas.[9] In addition, let us assume a cell-free network where $N_{\text{AP}} > N_{\text{user}}$, allowing different APs to serve a single user. Accounting for the low number of antennas in low-power small APs [64], each AP a_m, $m \in \{1, \cdots, N_{\text{AP}}\}$, generates only a single directional beam directed toward a receiving user u_k, $k \in \{1, \cdots, N_{\text{user}}\}$, with its transmit precoding denoted by $\mathbf{v}_m$.

[9]While this paper considers downlink communications, the workflow in Algorithm 1 can be extended to accommodate uplink communications, allowing optimization at both the AP and user levels, including uplink power management [63]. The classical-valued optimization variables in such uplink scenarios can be obtained via quantum measurement. For instance, we may optimize the k-th user's uplink power $p_{\text{UL},k} = P_{\text{maxUL},k} \cdot \varepsilon_{\text{UL},k}$, where $P_{\text{maxUL},k}$ is a constant denoting the k-th user's maximum uplink power, and $\varepsilon_{\text{UL},k} \in [0, 1)$ is the power coefficient obtained from measurement.

Algorithm 1 Multi-Stage QNNs

Input: The acquired channel information, $\{\hat{\mathbf{H}}^{[i]}\}_{i=1}^{N_{\text{data}}}$
Output: The assignment policy, γ, and the transmit precoding, $\{\mathbf{v}_m\}_{m=1}^{N_{\text{edge}}}$
Initialization:
Define the sets of the available APs and users as $\mathcal{A}$ and $\mathcal{U}$, respectively.
Cloud QNN Training:
Given complete channel information of the cell-free network, $\hat{\mathbf{H}} = \{\hat{\mathbf{H}}_m\}_{m=1}^{N_{\text{edge}}}$, the cloud trains the cloud QNN model U^{cloud} (to be described later in Algorithm 2), resulting in its trained parameter set θ^{cloud}.
Edge QNN Training:
for each $t \in \{1, \cdots, N_{\text{iteration}}\}$ **do**
  Each m-th edge, $m \in \{1, \cdots, N_{\text{edge}}\}$, sends its channel information $\hat{\mathbf{H}}_m$ to the cloud.
  As it acquired the complete channel information for the network, $\hat{\mathbf{H}} = \{\hat{\mathbf{H}}_m\}_{m=1}^{N_{\text{edge}}}$, and optimized parameter vector, Θ^{cloud}, the cloud employs U^{cloud} to estimate γ.
  The cloud broadcasts transmitter-user assignment γ to all edges.
  Given γ, each m-th edge trains its learning model, $U^{[m]}$, using its local channel information $\hat{\mathbf{H}}_m$, resulting in $\theta^{[m]}$ (to be described later in Algorithm 3).
end for
Deployment:
As it acquires instantaneous channel information $\hat{\mathbf{H}}$, the cloud employs U^{cloud} with optimized Θ^{cloud} to estimate γ. In turn, accounting for γ and $\hat{\mathbf{H}}_m$, each m-th AP employs $U^{[m]}$ with optimized $\Theta^{[m]}$ to estimate its transmit precoding $\mathbf{v}_m$.

Moreover, the wireless communication channel between a_m and u_k can be expressed as [65]

$$\mathbf{h}_{m,k} = \begin{bmatrix} h_{m,k}^{[1]} & \cdots & h_{m,k}^{[N_{\text{Tx}}]} \end{bmatrix}^{\mathsf{T}} \subseteq \mathbb{C}^{N_{\text{Tx}}}, \tag{3}$$

where $h_{m,k}^{[j]} = 1/\sqrt{N_{\text{path}}} \sum_{n=1}^{N_{\text{path}}} g_{m,k}^{[j]} a(\phi_n)$, $j \in \{1, \cdots, N_{\text{Tx}}\}$, in which $g_{m,k}^{[j]}$ denotes the channel gain between a_m and u_k. $a(\phi_n)$ conveys uniform linear array (ULA) antenna steering: $a(\phi_n) = \{\exp(-\mathbf{i}2\pi\phi_n z)\}_{z\in\mathcal{Z}}$, where $\mathcal{Z} = \{n - 0.5 \cdot (N_{\text{Tx}} - 1)\}_{n\in\{1,\cdots,N_{\text{Tx}}-1\}}$, and ϕ_n marks the departure angle [65]. Moreover, the acquired channel information can be presented as $\hat{\mathbf{H}} = \mathbf{H} + n_{\text{CSI}}$, where n_{CSI} accounts for the imperfection in the channel information [66].

A. User Rates

Let P_t be the total transmit power of each AP. Moreover, the transmit signal-to-noise ratio can be presented as $\rho = P_t/\sigma^2$, in which σ^2 is the noise variance. Considering similar P_t for each AP, the downlink rate for u_k served by a_m can be expressed as

$$R_{m\to k} = \log_2\left(1 + \frac{\rho|\hat{\mathbf{h}}_{m,k}^{\mathsf{T}}\mathbf{v}_m|^2}{\rho\sum_{n=1,n\neq m}^{N_{\text{AP}}}\mu_{n,k}|\hat{\mathbf{h}}_{n,k}^{\mathsf{T}}\mathbf{v}_n|^2 + 1}\right), \tag{4}$$

where $\hat{\mathbf{h}}_{m,k} \in \hat{\mathbf{H}}_m$ denotes the acquired channel information corresponds to the links between u_k and a_m, $\mathbf{v}_m$ denotes the transmit precoding of the m-th AP, and $\mathbf{v}_m = \begin{bmatrix} v_m^{[1]} & \cdots & v_m^{[N_{\text{Tx}}]} \end{bmatrix}^{\mathsf{T}} \subseteq \mathbb{C}^{N_{\text{Tx}}}$, with $v_m^{[j]}$ as the j-th entry of $\mathbf{v}_m$. The term $\rho \sum_{n=1, n\neq m}^{N_{\text{AP}}} \mu_{n,k} |\hat{\mathbf{h}}_{n,k}^{\mathsf{T}} \mathbf{v}_n|^2$ indicates interference from all APs other than a_m, with $\mu_{n,k} < 1$ denoting the interference factor from AP a_n to the user u_k.[10] The rate of u_k can then be expressed as

$$R_k = \log_2 \left(1 + \frac{\sum_{m \in \mathcal{A}_m} \rho |\hat{\mathbf{h}}_{m,k}^{\mathsf{T}} \mathbf{v}_m|^2}{\rho \sum_{n \in \mathcal{A}, n \notin \mathcal{A}_m} \mu_{n,k} |\hat{\mathbf{h}}_{n,k}^{\mathsf{T}} \mathbf{v}_n|^2 + 1} \right). \quad (5)$$

In Eq. (5), the term $\sum_{m \in \mathcal{A}_m} \rho |\hat{\mathbf{h}}_{m,k}^{\mathsf{T}} \mathbf{v}_m|^2$ collects the desired signal for the k-th user, transmitted by the APs in $\mathcal{A}_m$ and combined at the receiving user, whereas the term $\rho \sum_{n \in \mathcal{A}, n \notin \mathcal{A}_m} \mu_{n,k} |\hat{\mathbf{h}}_{n,k}^{\mathsf{T}} \mathbf{v}_n|^2$ accounts for the interference from remaining APs outside $\mathcal{A}_m$ [70]. The term $\rho = {}^{P_t}/_{\sigma^2}$ denotes the transmit signal-to-noise ratio, with σ^2 indicating the noise variance.

B. The Objective

The objective of the QNN-based optimization in this study is to maximize the lowest rate among the users as a max-min optimization, which is used to avoid APs only being allocated to the users with a higher channel gain.[11] Given R_k as the rate of the k-th user, $k \in \{1, \cdots, N_{\text{user}}\}$, the optimization problem can be expressed as:

$$\begin{aligned} \underset{\Lambda}{\text{maximize}} \quad & \min_{\{u_k\}_{k=1}^{N_{\text{user}}}} R_k(\Lambda) && \text{(6a)} \\ \text{subject to} \quad & \|\mathbf{v}_m\|_2^2 \leq 1, \forall m, && \text{(6b)} \\ & \varrho_k \geq 1, \forall k, && \text{(6c)} \end{aligned}$$

where $\Lambda = \left\{ \gamma, \{\mathbf{v}_m\}_{m=1}^{N_{\text{AP}}} \right\}$ is the tuple of solution which contains (i) transmitter-user assignment policy γ and (ii) transmit precoding $\{\mathbf{v}_m\}_{m=1}^{N_{\text{AP}}}$; let

$$\gamma \triangleq \begin{bmatrix} \gamma_{1,1} & \cdots & \gamma_{1,N_{\text{user}}} \\ \vdots & \ddots & \vdots \\ \gamma_{N_{\text{AP}},1} & \cdots & \gamma_{N_{\text{AP}},N_{\text{user}}} \end{bmatrix} \in \{0,1\}^{N_{\text{AP}} \times N_{\text{user}}}, \quad (7)$$

where $\gamma_{m,k} = \{0,1\}, \forall m, \forall k$. $\gamma_{m,k} = 1$ indicates that the m-th AP is assigned to the k-th user; otherwise, $\gamma_{m,k} = 0$.[12] The energy normalization constraint is conveyed in Eq. (6c), ensuring compliance with each m-th AP's transmit power limitation. The constraint in Eq. (6c) ensures that each k-th user is assigned at least one AP, ϱ_k marks the number of APs assigned to the user thereof. Further, as the non-convex problem in Eq. (6) involves different optimization variables, we can split it into two classes of sub-problems: the transmitter-user assignment and transmission precoding problems.[13] The solution of the first sub-problem, which is to assign each of the cell-free APs to the corresponding user terminal, is estimated by the cloud QNN model, later discussed in Section IV-A. Subsequently, the solutions of the second sub-problem, aimed to optimize the transmit precoding of each of the cell-free APs, are optimized using the edge QNN models, later defined in Section IV-A.

The complexity of the problem: Eq. (6) can be presented as a non-convex, NP-hard problem. (i) *Non-convexity*: owing to the potential inter-AP interference associated with $R_k(\cdot)$ (see Eq. (4)), Eq. (6) is inherently non-convex, as widely discussed in other studies, e.g., [74]. (ii) *NP-hardness*: The problem of Eq. (6) can be posed as a min-rate utility maximization, e.g., as in [75], and there exists the possibility of reducing this problem as a generalized fractional linear programming problem, to be solved by parametric linear programming. Unfortunately, constraint Eq. (6c) prevents this approach: although common spectral resource is assumed, Eq. (6c) requires that each user be assigned no fewer than one AP. Such a constraint maintains Eq. (6) as a combinatorial problem, with the solution space expanding with N_{user} and N_{AP}. Since this paper assumes $N_{\text{user}} > 2$ and $N_{\text{AP}} > 2$, Eq. (6) remains as an NP-hard problem, precluding obtaining a global optimum via iterative means within polynomial time.

IV. Employing Multi-Stage QNNs

A. Transmitter-User Assignment

The *first sub-problem*, that is a transmitter-user assignment problem, is now discussed as follows. In accordance with Eq. (6), the objective of this phase is to maximize the achieved rate of the user with the minimum rate, which can be expressed as:

$$\begin{aligned} \underset{\gamma}{\text{maximize}} \quad & \min_{\{u_k\}_{k=1}^{N_{\text{user}}}} R_k(\gamma | \mathbf{v}_m) && \text{(8a)} \\ \text{subject to} \quad & \|\mathbf{v}_m\|_2^2 \leq 1, && \text{(8b)} \\ & \varrho_k \geq 1, \forall k, && \text{(8c)} \end{aligned}$$

Nonetheless, this sub-problem focuses solely on optimizing the transmitter-user assignment, without taking into account the

[10] For generality, this study assumes non-coherent transmissions in the system model, which do not necessitate transmitter phase-synchronizations [67]. Importantly, the applicability of the proposed QNN-based approach, presented in Algorithms 2 and 3, extends to coherent transmissions, as well as to combinations of both types [68]. In this way, it serves as suitable future work for high-mobility network environments, e.g., vehicle-to-everything (V2X) communications. For instance, a separate parameterized quantum circuit may be required to yield a time-varying parameter $\vartheta_m^{(\tau)}$, associated with the channel's coherence interval τ. The imperfect non-coherent joint transmission between multiple APs serving the same user can be realized through applying a phase compensation factor. For AP m at time τ, this is given by $\varsigma_m^{(\tau)} = -\vartheta_m^{(\tau)} + \zeta^{(\tau)}$, where $\zeta^{(\tau)} \sim \mathcal{U}[-\theta_{\text{pu}}, \theta_{\text{pu}}]$ represents the channel phase uncertainty bounded by the phases $-\theta_{\text{pu}}$ and θ_{pu}, with $\mathcal{U}[\cdot,\cdot]$ indicating a uniform distribution [69].

[11] The max-min optimization objectives have been considered in the literature to attain rate fairness among N_{user} users. Recent studies such as [63], which optimize transmit power in cell-free systems, adopt learning-based workflows in which parameterized models (e.g., QNNs), once trained, can infer solutions to the max-min optimization, mitigating the need for iterative methods that may exceed the coherence time of wireless channels in the dynamic environments of future wireless networks. They typically involve linear constraints, which, in the context of this paper, can later be observed in (6b), each associated with the m-th AP [71].

[12] The transmitter-user assignment in this paper allows different APs to collaborate, particularly when $|\mathcal{A}| > |\mathcal{U}|$, representing a many-to-one relationship for each user k, as illustrated in Fig. 2 and further reflected in Eq. (5). Such a consideration aligns with studies such as [72], which outlines it as repeated AP-user association, allowing different APs to serve the same user [72, Algorithm 1], and [73], which frames it as a means to facilitate user performance fairness, through maximizing weighted sum rates.

[13] To be precise, Eq. (6) can be perceived as a mixed-integer non-convex problem, due to the presence of binary optimization variables concerning transmitter-user assignment (will be articulated in Eq. (7)).

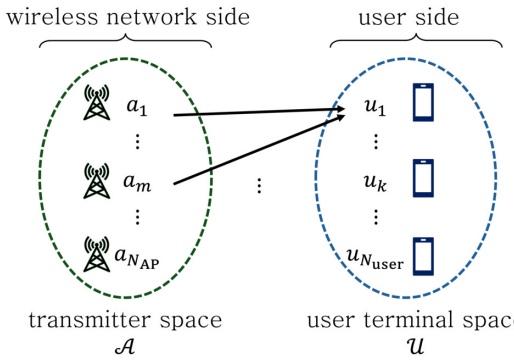

Figure 2: The sub-problem of the transmitter-user assignment, illustrated as a bipartite graph. It can be considered a many-to-one relationship, under the assumption that $|\mathcal{A}| > |\mathcal{U}|$ [76]. Such a relationship reflects the global optimization of transmitter–user assignment (detailed later in Algorithm 2), which is performed prior to the local regression tasks of transmit precoding optimization at each AP (see Algorithm 3).

variability of transmit precoding for each AP, that is assumed to be static during the optimization process. Particularly, maximum-ratio-based transmit precoding, i.e., $\mathbf{v}_m^{\text{MR}} = \hat{\mathbf{h}}_m^* / \|\hat{\mathbf{h}}_m^*\|_2$, $\forall m \in \{1, \cdots, N_{\text{AP}}\}$, can be employed during this optimization phase [77], ensuring that the constraint of Eq. (8b) holds. In complying with the constraint of Eq. (8c), a penalty variable can be invoked, as set forth following Eq. (14).

To solve the sub-problem defined in Eq. (8), the transmitter-user assignment can be formulated as a bipartite graph that links the members of the AP space $\mathcal{A}$ to those of the user space $\mathcal{U}$ in a many-to-one correspondence, as illustrated in Fig. 2, assuming that $|\mathcal{A}| > |\mathcal{U}|$, different APs can serve a particular user terminal. To solve this sub-problem, a quantum-based optimization scheme is presented in Fig. 3.[14] For all k-th user, given channel matrix $\hat{\mathbf{H}}$ as an input feature, the assignment policy output can be expressed as

$$\pi_{\text{assign}} : \hat{\mathbf{H}} \xrightarrow{U_{\text{QNN}}^{\text{cloud}}} \gamma, \tag{9}$$

where $U_{\text{QNN}}^{\text{cloud}}$ is the QNN operation at the cloud. $\hat{\mathbf{H}}$ can be obtained, in practice, through uplink data pilots, where the channel reciprocity, apposite in acquiring downlink channel information, is affected by time-division duplexing (TDD) [78].[15] For that matter, the QNN outputs can be decoded as a feature matrix representing the transmitter-user assignment, which was specified earlier in Eq. (7).

To provide solutions for the aforementioned sub-problem, the proposed QNN-based optimization scheme is presented in Algorithm 2. The designated QNN operation employed in Algorithm 2 is defined below. Different quantum logical gates employed across this paper are discussed in Appendix A.

Definition 1 (The QNN operation in the cloud)**.** With θ^{cloud} representing the set of its parameters, the QNN operation, which is processed in the cloud and used to optimize transmitter-user assignment, can be expressed as

$$U^{\text{cloud}} = U_{\text{connect}}^{\text{cloud}}(\theta^{\text{cloud}})\, U_{\text{encode}}^{\text{cloud}}(\hat{\mathbf{H}}), \tag{10}$$

where $U_{\text{encode}}^{\text{cloud}}$ and $U_{\text{connect}}^{\text{cloud}}$ are the encoding and connection operations, respectively. Moreover, the initial qubit preparation can be expressed as $|0\rangle^{\otimes(N_{\text{layer}} \times N_{\text{neuron}})}$, where N_{layer} is the number of layers and N_{neuron} is the number of neurons per layer. The encoding operation, used to embed classical information into Hilbert spaces, can be thus articulated [79]:

$$U_{\text{encode}}^{\text{cloud}}(\hat{\mathbf{H}}) \triangleq \bigotimes_{m=1}^{N_{\text{AP}}} \Big(\underbrace{\prod_{k=1}^{N_{\text{user}}} \mathbf{R}_{\text{Z}}\big(\hat{\mathbf{h}}_{m,k}; \big|q_{1,n}^{\text{cloud}}\big\rangle\big)\mathbf{H}\big(\big|q_{1,n}^{\text{cloud}}\big\rangle\big)}_{\text{for } a_m} \Big), \tag{11}$$

where $\left|q_{l,n}^{\text{cloud}}\right\rangle$ indicates a particular qubit designated for the n-th neuron of the l-th layer, $l \in \{1, \cdots, N_{\text{layer}}\}$, $n \in \{1, \cdots, N_{\text{neuron},l}\}$, and the term $\mathbf{R}_{\text{Z}}\big(\hat{\mathbf{h}}_{m,k}; \big|q_{1,n}^{\text{cloud}}\big\rangle\big)\mathbf{H}\big(\big|q_{1,n}^{\text{cloud}}\big\rangle\big)$ is used to encode the channel information related to a particular AP a_m, $m \in \{m, \cdots, N_{\text{AP}}\}$. Subsequently, the main QNN operation,

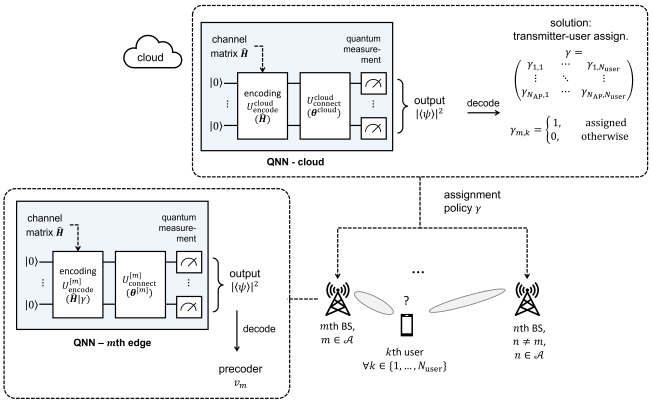

Figure 3: The proposed QNN-based optimization framework for the joint optimizations of transmitter-user assignment and transmit precoding in cell-free system. The framework utilizes non-centralized QNN models, which are processed by cloud and edge processing units to estimate the solutions for different sub-problems. The cloud QNN: It encodes channel information, originally as classical information, into a quantum circuit via an encoding operation, as in Eq. (11). Its parameterized operation allows it to be trained, as in Algorithm 2. Its output is then translated into a transmitter-user assignment solution. The edge QNN: It encodes its local channel channel information, relevant for the corresponding m-th AP, as in Eq. (19). Its parameterized operation is to be trained as in Algorithm 3. The output thence yields transmit precoding solution for the m-th AP.

[14]To ease implementation, quantum-based computations can be conducted using platforms and libraries currently available, e.g., IBM Qiskit [24].

[15]The upperbound of the cross-correlation effect of the pilot contamination can be analyzed as follows [20]: Let us assume uplink (UL) pilot training transmitted through time division duplexing (TDD). Given K orthogonal pilot signals, each k-th signal can be expressed as $\sqrt{N_{\text{pilot}}}\xi_k \in \mathbb{C}^{N_{\text{pilot}} \times 1}$. Accordingly, the pilot contamination correlation coefficient between users k and l is given by $\xi_{k,l} = |\xi_k^{\text{H}} \cdot \xi_l|^2$. The term $\xi_{k,l}$ is upperbounded by $\xi_{k,l} \leq \frac{\epsilon}{(\max_{1 \leq m \leq N_{\text{AP}}} \{\frac{\mathbf{a}_{l,m}}{\mathbf{a}_{k,m}}\})^2}$, where $\mathbf{a}_k \in \mathbb{Z}^{N_{\text{AP}} \times 1}$ and $\mathbf{a}_l \in \mathbb{Z}^{N_{\text{AP}} \times 1}$ are the matrix comprising of the pathloss values of the links between N_{AP} APs and user k and l, respectively.

Algorithm 2 Transmitter-User Assignment

Input: The acquired channel information $\left\{\hat{\mathbf{H}}^{[i]}\right\}_{i=1}^{N_{\text{data}}}$.
Output: The assignment policy, γ.
Initialization:
1: Define the sets of the available APs and users as $\mathcal{A}$ and $\mathcal{U}$, respectively.
2: Prepare the qubits, by initializing the state as $|0\rangle$ for each of the qubits.
Training:
3: **for** each $i_{\text{train}} \in \{1, \cdots, N_{\text{train}}\}$ **do**
4: Considering $\hat{\mathbf{H}}^{[i]}$ as the input, employs U^{cloud} in Eq. (10).
5: Decode the output of U^{cloud}, and then obtain assignment policy γ, as in Eq. (9).
6: Obtain $Q_{\text{assign}}(\gamma|\hat{\mathbf{H}}^{[i]}, \mathbf{v}_m^{\text{MR}})$ and $\Phi_{\text{assign}}(\hat{\mathbf{H}}^{[i]})$ (as in Eq. (14)).
7: Considering the training loss $L_{\text{assign}}(\hat{\mathbf{H}}^{[i]})$ as in Eq. (13), calculate $\nabla L_{\text{assign}}(\hat{\mathbf{H}}^{[i]})$ (as discussed in Appendix B). Update parameter as $\theta^{\text{cloud}} \leftarrow \theta^{\text{cloud}} - \mu \nabla L_{\text{assign}}(\hat{\mathbf{H}}^{[i]})$. The Rotosolve approach, as discussed in Appendix B, can also be employed.
8: **end for**

which is comprised of different neurons in different layers, can then be expressed as [80]

$$U_{\text{connect}}^{\text{cloud}}(\theta^{\text{cloud}}) \triangleq \bigotimes_{l=1}^{N_{\text{layer}}^{\text{cloud}}-1} \prod_{n=1}^{N_{\text{neuron},l}^{\text{cloud}}-1} \underbrace{\mathbf{C}_{\text{Z}}\left(\left|q_{l,n+1}^{\text{cloud}}\right\rangle; \left|q_{l,n}^{\text{cloud}}\right\rangle\right)}_{\text{connect neuron}}$$
$$\bigotimes_{l=1}^{N_{\text{layer}}^{\text{cloud}}-1} \underbrace{\mathbf{C}_{\text{X}}\left(\left|q_{l+1,1}^{\text{cloud}}\right\rangle; \left|q_{l,N_{\text{neuron},l}^{\text{cloud}}}^{\text{cloud}}\right\rangle\right)}_{\text{connect layer}}$$
$$\bigotimes_{l=1}^{N_{\text{layer}}^{\text{cloud}}} \bigotimes_{n=1}^{N_{\text{neuron},l}^{\text{cloud}}} \mathbf{R}_{\text{Y}}\left(\theta_{l,n}^{\text{cloud}}\right), \tag{12}$$

where $\theta_{l,n}^{\text{cloud}} \in \theta^{\text{cloud}}$ indicates the vector of weight parameters for the n-th neuron, l-th layer. The terms of $\mathbf{C}_{\text{Z}}\left(\left|q_{l,n+1}^{\text{cloud}}\right\rangle; \left|q_{l,n}^{\text{cloud}}\right\rangle\right)$ and $\mathbf{C}_{\text{X}}\left(\left|q_{l+1,1}^{\text{cloud}}\right\rangle; \left|q_{l,N_{\text{neuron},l}^{\text{cloud}}}^{\text{cloud}}\right\rangle\right)$ respectively indicate the operations of Z- and X-rotations employed to connect different neurons and layers. For instance, $\mathbf{C}_{\text{Z}}\left(\left|q_{l,n+1}^{\text{cloud}}\right\rangle; \left|q_{l,n}^{\text{cloud}}\right\rangle\right)$ applies Z-rotation on $\left|q_{l,n+1}^{\text{cloud}}\right\rangle$, according to the state $\left|q_{l,n}^{\text{cloud}}\right\rangle$ (see Appendix A).

Furthermore, to obtain classical values as the output of the cloud quantum circuit, we can perform quantum measurements on the output state of $U_{\text{connect}}^{\text{cloud}}$, as illustrated in Fig. 3. After the measurement, each classical-valued output can be defined as $o_m^{\text{cloud}} \in [0, 1]$, $\forall m \in \{1, \cdots, N_{\text{AP}}\}$, and the transmitter-user solution for m-th AP can be obtained as: $\hat{o}_m = \lfloor o_m^{\text{cloud}} N_{\text{user}} \rceil$, whereas $\hat{o}_m$ indicates the index of the user that will be served by the m-th AP. For example, if $\hat{o}_1 = 2$, the first user will be served by the second AP. In addition, the value of the training loss, which quantifies the performance of the cloud QNN model, can be defined as the difference between the performance achieved by employing the quantum model and a reference point representing the target performance, which can be expressed as

$$\text{L}_{\text{assign}}(\gamma|\hat{\mathbf{H}}) = \left\| Q_{\text{assign}}(\gamma|\hat{\mathbf{H}}, \mathbf{v}_m^{\text{MR}}) - \Phi_{\text{assign}}(\hat{\mathbf{H}}) \right\|^2, \tag{13}$$

while the reward can be formulated as

$$Q_{\text{assign}}\left(\gamma|\hat{\mathbf{H}}, \mathbf{v}_m^{\text{MR}}\right) = -\min_{\{u_k\}_{k=1}^{N_{\text{user}}}} R_k\left(\gamma|\mathbf{v}_m^{\text{MR}}\right). \tag{14}$$

For the aforementioned reference point, we assume that each j-th transmission link is physically separated from the other links and does not interfere with them, allowing them to be treated as orthogonal to each other. In this case, we use $\Phi_{\text{assign}}(\hat{\mathbf{H}}) = -\sum_{m=1}^{N_{\text{AP}}} \sum_{i=1}^{N_\lambda^{[m]}} \log_2\left(1 + \frac{\lambda_i^{[m]}}{N_\lambda^{[m]}}\rho\right)$, where $\{\lambda_i^{[m]}\}_{i=1}^{N_{\text{Tx}}}$ indicates the array of eigenvalues of $\hat{\mathbf{h}}_{m,k}$, as the target performance to represent the ideal capacity [81], [82], given the acquired channel information $\hat{\mathbf{H}}$.[16] In order to meet the constraint in Eq. (8c), let us introduce a penalty variable r_{penalty} to the reward function, to encourage the cloud to always assign an AP to each user: Specifically, when no AP is assigned to the k-th user, $R_k\left(\gamma|\mathbf{v}_m^{\text{MR}}\right) = -r_{\text{penalty}}$ is assumed instead of $R_k\left(\gamma|\mathbf{v}_m^{\text{MR}}\right) = 0$.

B. Transmit Precoding Optimization

The solution for the *second sub-problem*, which is defined as the precoding optimization of each AP a_m, $\forall m \in \{m, \cdots, N_{\text{AP}}\}$, is estimated by each of the edge QNN models. Given assignment solution $\gamma_m \in \gamma$ and assigned user k, the precoding optimization of the m-th AP can be expressed as

$$\underset{\mathbf{v}_m}{\text{maximize}} \quad R_{m\to k}(\mathbf{v}_m|\gamma_m) \tag{15a}$$
$$\text{subject to} \quad \|\mathbf{v}_m\|_2^2 \le 1. \tag{15b}$$

The constraint of Eq. (6c) is relaxed in Eq. (15), as satisfying Eq. (6c) is the cloud's responsibility, and each edge optimize its corresponding AP according to the cloud's optimized γ_m. For each m-th AP, given channel information and assignment policy coupled as input features $\{\hat{\mathbf{h}}_m, \gamma_m\}$, the precoding policy for AP a_m can be expressed as $\pi_{\text{precode}} : \{\hat{\mathbf{h}}_m, \gamma_m\} \xrightarrow{U_{\text{QNN}}^{[m]}} \mathbf{v}_m$, where the precoding elements can be presented as $\mathbf{v}_m = \{v_m^{[1]}, \cdots, v_m^{[N_{\text{Tx}}]}\} \in \mathbb{C}^{N_{\text{Tx}}}$. In addition, to ensure that the constraint of Eq. (15b) is met, let us consider the normalization of $\mathbf{v}_m$, as so articulated: $\mathbf{v}_m \leftarrow {}^{\mathbf{v}_m}/_{\|\mathbf{v}_m\|^2}$. As we aim to maximize the sum rate in Eq. (15), the reward for each m-th AP can be expressed as

$$Q_{\text{precode}}^{[m]}(\mathbf{v}_m|\hat{\mathbf{h}}_m, \gamma) = -R_{m\to k}(\mathbf{v}_m|\gamma). \tag{16}$$

Moreover, the pertinent training loss can be expressed as

$$\text{L}_{\text{precode}}(\pi_{\text{precode}}|\hat{\mathbf{h}}_m) = \|Q_{\text{precode}}^{[m]}(\mathbf{v}_m|\hat{\mathbf{h}}_m, \gamma) - \Phi_{\text{precode}}(\hat{\mathbf{h}}_m)\|^2, \tag{17}$$

[16]The use of target values, $\Phi_{\text{assign}}(\cdot)$ in Eq. (13) and $\Phi_{\text{precode}}(\cdot)$ in Eq. (17), within this paper, is intended to facilitate training convergence. They represent ideal performance levels, and thus, the training losses indicate the deviation of achieved performance from the ideals. This can be extended to other wireless network scenarios as well, for $\Phi_{\text{assign}}(\cdot)$ and $\Phi_{\text{precode}}(\cdot)$, in particular, can be conveniently substituted with upper-bound performance indicators.

where $\Phi_{\text{precode}}(\hat{\mathbf{h}}_m)$ denotes the target reward for AP m, which is defined as

$$\Phi_{\text{precode}}(\hat{\mathbf{h}}_m) = -\sum_{i=1}^{N_\lambda^{[m]}} \log_2\Big(1+\frac{\lambda_i^{[m]}}{N_\lambda^{[m]}}\rho\Big). \tag{18}$$

Definition 2 (The QNN operation pertinent to each m-th AP)**.** The QNN operation employed for optimizing transmit precoding of each m-th AP can be expressed as

$$U^{[m]} \triangleq U^{[m]}_{\text{connect}}(\theta^{[m]})U^{[m]}_{\text{encode}}(\hat{\mathbf{H}}) \tag{19}$$

where $U^{[m]}_{\text{encode}}$ and $U^{[m]}_{\text{connect}}$ are the encoding and connection operations, respectively. To execute $U^{[m]}$, the qubits can be initialized as $|0\rangle^{\otimes N_{\text{user}}}$. As in [28], the encoding operation for the m-th AP can be presented as $U^{[m]}_{\text{encode}} = \bigotimes_{k=1}^{N_{\text{user}}} \mathbf{R}_{\text{Z}}\big(\hat{\mathbf{h}}_{m,k}; \Big|q_k^{[m]}\Big\rangle\big)\mathbf{H}\big(\Big|q_k^{[m]}\Big\rangle\big)$. Akin to the operations in [28] and [80], the main QNN operation for the m-th AP can be expressed as

$$\begin{aligned}
U^{[m]}_{\text{connect}}(\theta^{[m]}) \triangleq\ & \mathbf{R}_{\text{Y}}\big(\theta_1^{[\text{m}]}; \Big|q_1^{[m]}\Big\rangle\big)\prod_{k=2}^{N_{\text{user}}}\mathbf{C}_{\text{Z}}\big(\Big|q_k^{[m]}\Big\rangle; \Big|q_{k-1}^{[m]}\Big\rangle\big) \\
& \bigotimes_{k=1}^{N_{\text{user}}}\mathbf{R}_{\text{Y}}\big(\Big|q_k^{[m]}\Big\rangle\big)\mathbf{C}_{\text{Z}}\big(\Big|q_1^{[m]}\Big\rangle; \Big|q_{N_{\text{user}}}^{[m]}\Big\rangle\big) \\
& \prod_{k=2}^{N_{\text{user}}}\mathbf{C}_{\text{Z}}\big(\Big|q_k^{[m]}\Big\rangle; \Big|q_{k-1}^{[m]}\Big\rangle\big)\prod_{k=2}^{N_{\text{user}}}\Big\{\mathbf{C}_{\text{Z}}\big(\Big|q_k^{[m]}\Big\rangle; \Big|q_{k-1}^{[m]}\Big\rangle\big) \\
& \mathbf{R}_{\text{Y}}\big(\theta_k^{[\text{m}]}; \Big|q_k^{[m]}\Big\rangle\big)\mathbf{C}_{\text{Z}}\big(\Big|q_k^{[m]}\Big\rangle; \Big|q_{k-1}^{[m]}\Big\rangle\big)\Big\},
\end{aligned} \tag{20}$$

where $\theta_k^{[m]} \in \theta^{[m]}$ is the weight parameter for the k-th input embedding, $k \in \{1,\cdots,N_{\text{user}}\}$. It is worth noting that, in contrast to the operation of $U^{\text{cloud}}_{\text{connect}}$, the qubits are assigned depending on the number of input entries of the said operation, as defined in Eq. (20).

C. The Algorithms for Transmitter-User Assignment and Transmit Precoding Optimization

After γ is obtained, each m-th edge executes Algorithm 3 to estimate precoding. The number of data training samples and episodes are presented as N_{data} and N_{epoch}. As Algorithm 2 and Algorithm 3 employ stochastic gradient descent, the number of training iterations is given as $N_{\text{train}} = N_{\text{data}}N_{\text{epoch}}$. The learning step size is denoted as μ.

D. Complexity Analysis and the Number of Required Qubits

To facilitate the analysis, we evaluate the complexities of Algorithm 2 and Algorithm 3 based on the number of gradient calculations. In particular, we use the symbol $\mathcal{G}$ to represent a single gradient calculation. Based on this notation, we can determine the complexities of Algorithm 2 and Algorithm 3 as $N_{\text{iter}}\mathcal{G}$ and $N_{\text{AP}}N_{\text{iter}}\mathcal{G}$, respectively. The number of required qubits for the QNNs under consideration can be analyzed as follows.

Lemma 1. *To execute the quantum-based operations $U^{cloud}_{connect}$ requires $\big((N^{cloud}_{layer}-1)+(N^{cloud}_{layer}-1)+N^{cloud}_{layer}N^{cloud}_{neuron}\big)$, while $U^{[m]}_{connect}$ requires N_{user} qubits.*

Algorithm 3 Transmit Precoding Optimizations for m-th AP

Input: The channel information, and the assignment policy γ, obtained from Algorithm 2.

Output: The transmit precoding $\mathbf{v}_m$.

Initialization:

1: Prepare the qubits, by initializing the state as $|0\rangle$ for each of the qubits.

Training:

2: **for** each $i_{\text{train}} \in \{1,\cdots,N_{\text{train}}\}$ **do**

3: Considering $\hat{\mathbf{h}}_m$ as input, employs $U^{[m]}$ in Eq. (19).

4: Decode the output of $U^{[m]}$, to obtain the assignment policy $\mathbf{v}_m$.

5: Given the assignment policy γ (see Algorithm 2), obtain $Q^{[m]}_{\text{precode}}(\mathbf{v}_m|\hat{\mathbf{h}}_m,\gamma)$ and $\Phi_{\text{precode}}(\hat{\mathbf{h}}_m)$ (as in Eqs. (16) and (18)). Subsequently, calculate the training loss L_{precode} in Eq. (17).

6: Calculate $\nabla^{[m]}_\theta L_{\text{precode}}$ (as in Appendix B). Update the edge QNN's parameter as per $\theta^{[m]} \leftarrow \theta^{[m]} - \mu\nabla^{[m]}_\theta L_{\text{precode}}$.

7: **end for**

Proof. As expressed in Eq. (12), $U^{\text{cloud}}_{\text{connect}}$ employs $N^{\text{cloud}}_{\text{layer}}-1$ qubits to connect layers and $N^{\text{cloud}}_{\text{neuron}}-1$ to connect neurons. Moreover, $N^{\text{cloud}}_{\text{layer}}N^{\text{cloud}}_{\text{neuron}}$ qubits are needed for weight parameters. In addition, the cloud's encoding operation $U^{\text{cloud}}_{\text{encode}}$ requires N_{AP} qubits. Therefore, $U^{\text{cloud}}_{\text{connect}}$ employs $\big((N^{\text{cloud}}_{\text{layer}}-1)+(N^{\text{cloud}}_{\text{layer}}-1)+N^{\text{cloud}}_{\text{layer}}N^{\text{cloud}}_{\text{neuron}}\big)$ qubits. On the other hand, as presented in Eq. (20), each m-th edge processing on $U^{[m]}_{\text{connect}}$ involves N_{user} connected qubits for the purpose of entanglement. In addition, for the operation of variational gates with the parameters acting as the weights, N_{user} qubits are required. Since the mentioned operations involved in $U^{[m]}_{\text{connect}}$ are performed using the same qubits, the whole operation of $U^{[m]}_{\text{connect}}$ requires N_{user} qubits. ∎

With respect to Eq. (12), the circuit depth, another computational complexity indicator, can be analyzed as follows. To facilitate the analysis, let us assume a uniform number of neurons across all layers, given by $N^{\text{cloud}}_{\text{neuron}} = N^{\text{cloud}}_{\text{neuron},1} = \cdots = N^{\text{cloud}}_{\text{neuron},l} = \cdots = N^{\text{cloud}}_{\text{neuron},N^{\text{cloud}}_{\text{layer}}}$. The initial component of $U^{\text{cloud}}_{\text{connect}}$, i.e., $\bigotimes_{l=1}^{N^{\text{cloud}}_{\text{layer}}-1}\prod_{n=1}^{N^{\text{cloud}}_{\text{neuron},l}-1}\mathbf{C}_{\text{Z}}\Big(\Big|q^{\text{cloud}}_{l,n+1}\Big\rangle; \Big|q^{\text{cloud}}_{l,n}\Big\rangle\Big)$, yields the circuit depth of $\mathcal{O}(N^{\text{cloud}}_{\text{neuron}})$, since we may execute $N^{\text{cloud}}_{\text{neuron}}-1$ instances of the operation $\prod_{n=1}^{N^{\text{cloud}}_{\text{neuron},l}-1}\mathbf{C}_{\text{Z}}\Big(\Big|q^{\text{cloud}}_{l,n+1}\Big\rangle; \Big|q^{\text{cloud}}_{l,n}\Big\rangle\Big)$ in parallel.[17] Likewise, the remaining two terms, i.e., $\bigotimes_{l=1}^{N^{\text{cloud}}_{\text{layer}}-1}\mathbf{C}_{\text{X}}\Big(\Big|q^{\text{cloud}}_{l+1,1}\Big\rangle; \Big|q^{\text{cloud}}_{l,N^{\text{cloud}}_{\text{neuron},l}}\Big\rangle\Big)$ and $\bigotimes_{l=1}^{N^{\text{cloud}}_{\text{layer}}}\Big\{\bigotimes_{n=1}^{N^{\text{cloud}}_{\text{neuron},l}}\mathbf{R}_{\text{Y}}\big(\theta^{\text{cloud}}_{l,n}\big)\Big\}$, each contribute only $\mathcal{O}(1)$ to the circuit depth. Overall, $U^{\text{cloud}}_{\text{connect}}$ manifests a circuit depth of $\mathcal{O}(N^{\text{cloud}}_{\text{neuron}})$. For reference, a typical classical neural network has a computational complexity of $\mathcal{O}(N^{\text{class}}_{\text{neuron}}(N^{\text{class}}_{\text{layer}})^2)$, where

[17]For generality and consistency with Eq. (12), this analysis assumes serial execution of controlled-Z gates within each layer. These gates may be scheduled in two sequential rounds, effectively reducing the circuit depth per layer to $\mathcal{O}(1)$.

$N_{\text{neuron}}^{\text{class}}$ and $N_{\text{layer}}^{\text{class}}$ are the number of neurons and layers, respectively, assuming an equal number of neurons per layer [83]. The parameter complexity of the cloud QNN presented in Eq. (12) can be analyzed as follows. As for $U_{\text{connect}}^{\text{cloud}}$, only the last term $\bigotimes_{l=1}^{N_{\text{layer}}^{\text{cloud}}} \left\{ \bigotimes_{n=1}^{N_{\text{neuron},l}^{\text{cloud}}} \mathbf{R}_{\text{Y}}\left(\theta_{l,n}^{\text{cloud}}\right) \right\}$ contains trainable parameters, particularly corresponding to the rotation operation around the Y-axis $\mathbf{R}_{\text{Y}}(\cdot)$. The other parts of (14), e.g., $\bigotimes_{l=1}^{N_{\text{layer}}^{\text{cloud}}-1} \mathbf{C}_{\text{X}} \left(\left| q_{l+1,1}^{\text{cloud}} \right\rangle ; \left| q_{l,N_{\text{neuron},l}^{\text{cloud}}}^{\text{cloud}} \right\rangle \right)$, only execute controllable gates like $\mathbf{C}_{\text{X}}(\cdot)$. Hence, $U_{\text{connect}}^{\text{cloud}}$ involves $\mathcal{O}(N_{\text{layer}}^{\text{cloud}} N_{\text{neuron}}^{\text{cloud}})$ parameters, assuming that $N_{\text{neuron}}^{\text{cloud}} = N_{\text{neuron},1}^{\text{cloud}} = \cdots = N_{\text{neuron},l}^{\text{cloud}} = \cdots = N_{\text{neuron},N_{\text{layer}}^{\text{cloud}}}^{\text{cloud}}$. Such a parameter complexity is comparable to existing classical learning models. For perspective, a full-batch graph NN and a graph convolutional network with variance reduction [84] both operate on $\mathcal{O}(N_{\text{layer}} N_{\text{nodes}} N_{\text{feat}})$ parameters, where N_{layer}, N_{nodes}, and N_{feat} represent the numbers of layers, nodes (or vertices), and each node's (or vertex's) features, respectively [85].

Further, the trainability of the cloud QNN can be expressed as follows. The norm of the gradient for the cloud QNN has bounds expressed as:

$$\begin{aligned} &\frac{1}{8}\left\{\text{Tr}[x_{\text{input}}]^2 + \text{Tr}[\sigma_z x_{\text{input}} \sigma_z]^2\right\} \\ &\quad \leq \mathbb{E}_{\theta^{\text{cloud}}}\left(\left\| \nabla \theta^{\text{cloud}} \hat{\mathbf{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}}) \right\|^2 \right) \leq N_{\text{layer}} N_{\text{neuron}}, \end{aligned} \tag{21}$$

where x_{input} denotes the input data, and σ_z represents the Pauli Z operator. The proof for these bounds can be found in Appendix D.

V. Results

In this section, the performance of the proposed method is investigated. Unless otherwise stated, the simulation and training parameters are presented in Table III.[18] In addition, in relation to the channel model articulated in Eq. (3), the Rayleigh channel gain is assumed, expressed as $g_{m,k}^{[j]} \sim \mathcal{CN}(0, d_{m,k}^{-\kappa})$, $\forall m$, $\forall k$, $\forall j$, in which κ represents the pathloss exponent and the normalized distance between m-th AP, and k-th user is denoted as $d_{m,k} \in [0,1]$ [88]. Further, we focus our study on employing non-centralized QNNs (Eqs. (10) and (19)) and the corresponding algorithm design (Algorithms 1, 2, and 3). As as in pertinent studies concerning model learning (including [89] and [44]), we accordingly assume that the APs, as parts of the radio access network, can access samples of CSI (channel state information), which can be obtained, for example, via pilot transmissions. The simulation was carried out as follows. First, Algorithm 2 was employed to estimate transmitter-user assignment solution (refer to Section IV-A). Imperfections in the channel information are accounted for in $\hat{\mathbf{H}}$, as discussed after Eq. 3. Second, Algorithm 3 optimized the precoding of each m-th AP, $m \in \{1, \cdots, N_{\text{AP}}\}$ (see Section IV-B). Finally, the achieved sum rates of the users were determined. During the training, a descending learning rate is assumed. In particular, the learning step at the i_{train}-th training iteration can be expressed as $\mu_{i_{\text{train}}} = \mu / \sqrt{i_{\text{episode}}+1}$. In addition, following the assumption in [90], the transmit SNR is known during training. A Monte-Carlo simulation with 10^3 trials was performed. Fig. 4 exhibits the achieved sum rates. Moreover, for the sake of comparison, an iterative search method, which searches for the highest possible transmitter-user assignment with maximum received SINR, is described in Appendix C. It is worth noticing that the search space of the search method in Appendix C, defined as the set of APs and users, i.e., $\mathcal{A}$ and $\mathcal{U}$, respectively, grows with the number of APs and users. In addition, the factor μ_d in Fig. 4 corresponds to the interference from neighboring APs (see Eqs. (4) and (5)). To maintain fair comparisons, let $\mu_d = \mu_{n,k}, \forall n, \forall k$. Notably, the proposed QNN-based framework outperforms the iterative search method, even under conditions of high interference among different APs.

In addition, the training losses of the QNNs processed by the cloud, pertinent to transmitter-user assignment, and the edges, pertinent to precoding optimizations, are shown in Fig. 5 and Fig. 6, respectively. As portrayed in Fig. 6, it is evident that the edge QNNs converge as early as the first episode. This expedited convergence reduces learning latency, as transmit precoding can be optimized without extended training iterations, facilitating operations. As shown in the figure, the average training loss values of edge QNN decreases by about 71.9% as early as the first episode, and remain stable thereafter. For example, between the second and the tenth episode, the average training loss varies by around 1.9%. These trends reflects progress toward approaching local minima, as each edge QNN independently optimizes the transmit precoding of its corresponding AP. Regarding Fig. 5, it is noteworthy that cloud needs to handle transmitter-user assignment for all users and AP spaces, and the vast number of combinations are mostly responsible for the alterations in training loss. Nonetheless, it is evident that the training loss follows a converging tendency. Our analysis attributes such fluctuations to the combinatorial nature of the solution space of the transmitter-user assignment managed by the cloud. Unfortunately, the inherent complexity of such combinatorial problems is well recognized across studies, e.g., in the user pairing problem studied in [91]. To add to this, the channel matrix $\hat{\mathbf{H}}$ is regenerated at each iteration, significantly affecting the user rate, albeit the algorithm's efforts to identify solutions. These fluctuations are much akin to those observed in reinforcement learning studies, including [92] and [91]. Further in optimizing AP transmit precoding, the proposed two-stage approach in Algorithm 1 requires fewer quantum measurements, thereby reducing resource demand. Since each m-th AP already knows its assigned users from

[18]While the limited number of currently available qubits restricts the network scale of this paper's simulation, studies like [86] promote the use of hybrid architectures. In future work, a classical-quantum algorithm sequence may be used to reduce the dimensionality of the CSI before feeding it into the quantum circuit. For instance, [87] employs classical convolutional neural network models to flatten and reshape the high-dimensional input data.

Table III: Parameters

Parameter	Value	Parameter	Value
N_{user}	3	μ	0.01
κ	2.3	N_{data}	100
$\mu_{n,k}$	0.1	N_{epoch}	100
N_{AP}	4	r_{penalty}	-10
N_{Tx}	2	ϱ	$\pi/2$

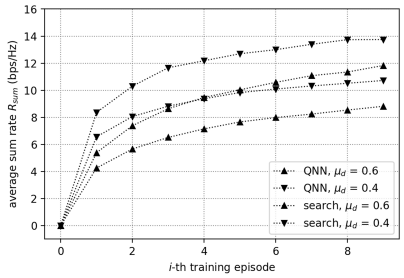

Figure 4: The achieved user sum rate.

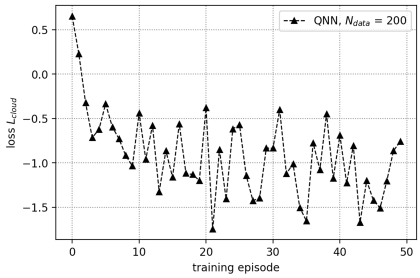

Figure 5: The training loss of the cloud QNN, pertinent to transmitter-user assignment (see Eq. (13)).

the previous optimization stage in Algorithm 2, each edge QNN only needs to yield N_{Tx} output entries, matching the dimensionality of the transmit precoding. In contrast, a centralized workflow employing a typical variational quantum circuit (VQC) architecture, as in [93] and [94], would need to produce $N_{\text{AP}}N_{\text{Tx}}$ entries for the same precoding optimization task, burdening the cloud's computational resources. These results align with the expected performance of QNNs capability to provide suitable solutions. Besides their trainability articulated earlier in Eq. (21), the expressivity of the QNN models is also investigated, as discussed thus.

Lemma 2. *The expressivity, $\mathcal{Y}$, of the QNN models of interest, particularly the cloud QNN, can be determined as the dimension of its dynamic Lie algebra $\mathfrak{a}$, that is, $\mathcal{Y} = dim(\mathfrak{a})$.*

Proof. In regard to Eqs. (10), (11), and (12), it is plausible to frame the cloud QNN model as a generalized generator $\mathcal{P}$, comprised of concurrent Pauli operators, i.e., $\mathrm{i}\mathcal{P} = \{\mathrm{i}\mathbf{O}_j\}_{j=1}^{J}$, where $\mathbf{O}_j$ is a Pauli operator with φ qubits, and where J denotes the depth of $\mathbf{O}$.[19] With $\langle\cdot\rangle_{\text{Lie}}$ indicating a Lie closure, the dynamic Lie algebra pertinent to the model can be expressed as $\mathfrak{a} = \text{span}(\langle\mathcal{P}\rangle_{\text{Lie}})$. Accounting for the analysis in [96], the expressivity of the model can thereby be described as $\mathcal{Y} = \dim(\mathfrak{a})$. ■

VI. Conclusion

Motivated by the rapid advancement of quantum computing, this study explored the possibilities of leveraging a QNN-based optimization scheme, to maximize the achieved sum information rate in cell-free MIMO systems. Specifically, the QNN-based framework is realized as a two-stage, non-centralized optimization process to handle joint optimization problems. First, a QNN model, processed by the cloud processing unit, is utilized to optimize transmitter-user assignment, given the complete channel information of the network. Afterwards, the transmitter-user assignment decision is forwarded to the edge processing units, which are each connected to an AP. Second, to optimize the transmit precoding of each AP, local QNN models are used. These models are computed by the edge processing units, and each one takes advantage of local channel information obtained by the corresponding AP. In addition, this work provided detailed descriptions of the specific quantum operations used for the proposed QNN-based scheme. Further, the trainability, loss, and performance results of the proposed QNN-based scheme were also investigated. Possible future directions for this work include applying quantum-based optimizations to estimate and optimize the parameters of the heterogeneous wireless communication networks [97]. The maximization or minimization of other performance metrics, e.g., minimizing the communication outage [98], can be pursued as optimization objectives. Additional optimization variables, including transmit power management, can be considered. QNNs can also be used in multi-agent settings, as demonstrated in [99] in which they are used to optimize multiple un-crewed aerial vehicles (UAVs) as a system. It also facilitates interplay between wireless systems and other verticals, such as traffic, railway, harbor, and/or airport management.

Though quantum computing remains in its infancy, the non-centralized workflow presented in this paper resonates with distributed quantum computing directives [100]. Subsequent studies may thus attend to operational aspects, particularly packet buffer management and entanglement distribution [101].

[19]Here, the controlled gates are not explicitly discussed, as with [95], as their relevance pertains primarily to contributing to the non-linearity of the outputs.

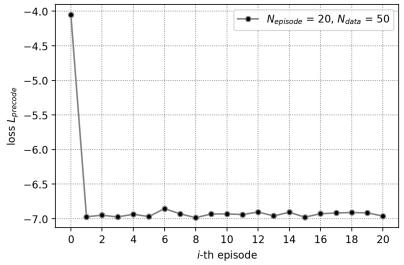

Figure 6: The average training loss of edge QNN pertinent to optimizing transmit precoding, taking on optimized assignment policy γ (see Eq. (17)).

Appendix A
Employed Quantum Logic Gates

The gates ensuing are used in the cloud and edge QNN operations across this paper. Hadamard gate: To introduce superposition into a single qubit, especially when its prior state is either $|0\rangle$ or $|1\rangle$. It can be conveyed as $\mathbf{H} \equiv \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$. Controlled-Z gate: To induce Pauli-Z operation on the target qubit $|q_a\rangle$, in reference to the state of

the control qubit $|q_b\rangle$. It is given by $\mathbf{C}_{\mathrm{Z}}(|q_a\rangle\,|\,|q_b\rangle) \equiv \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$. Controlled-X gate: To induce Pauli-X operation on the target qubit $|q_a\rangle$, given the state of the control qubit $|q_b\rangle$. It is given by $\mathbf{C}_{\mathrm{X}}(|q_a\rangle\,|\,|q_b\rangle) \equiv \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$. A quantum rotation gate around the Z-axis: To introduce a rotation along the Z-axis, to a single qubit, given the angle of rotation ϖ. It can be articulated as $\mathbf{R}_{\mathrm{Z}}(\varpi) \equiv \begin{bmatrix}e^{-i\frac{\varpi}{2}}&0\\0&e^{i\frac{\varpi}{2}}\end{bmatrix}$. A quantum rotation gate around the Y-axis: To apply a rotation along the Y-axis to a particular qubit, given the angle of rotation ϖ. It can be written as $\mathbf{R}_{\mathrm{Y}}(\varpi) \equiv \begin{bmatrix}\cos\left(\frac{\varpi}{2}\right)&-\sin\left(\frac{\varpi}{2}\right)\\\sin\left(\frac{\varpi}{2}\right)&\cos\left(\frac{\varpi}{2}\right)\end{bmatrix}$.

Appendix B
The Employed QNN Parameter Optimization Approach

As the QNN models take advantage of the gradient descent parameter optimization in Algorithm 2 and Algorithm 3, the gradient calculation can be described as follows. The gradient of $L \in \{L_{\text{assign}}, L_{\text{precode}}\}$, given parameter θ, calculated by leveraging parameter-shift rule [102], is given by $\nabla_\theta L(\theta) = {}^1\!/_{2\sin(\varrho)}\big(L(\theta+\varrho) - L(\theta-\varrho)\big)$, where ϱ is the shift parameter. Moreover, a gradient-free approach can also be employed. Let $\theta_i^{[k]}$ be the i-th element of the parameter vector $\theta^{[k]}$, where $k \in \{\text{cloud}, m\}$. As an effort to acquire optimized parameter $\hat{\theta}_i^{[k]} = \arg\min_{\theta_i^{[k]}} L$, the i-th parameter can be optimized using a Rotosolve-based approach as $\hat{\theta}_i^{[k]} = -\pi/2 - \arctan\left(\frac{2L_0 - L_+ - L_-}{L_+ - L_-}\right)$ where $L_0 = L(\theta_i^{[k]} = 0)$, $L_+ = L(\theta_i^{[k]} = \pi/2)$, and $L_- = L(\theta_i^{[k]} = -\pi/2)$, accordingly [103].

Appendix C
The Search-Based Method

For comparison, a search-based method is presented in Algorithm 4. The received signal-to-noise-ratio of the k-th user served by the j-th AP is given as $\Gamma_{j\to k} = \frac{\rho|\hat{\mathbf{h}}_{m,k}^T v_m|^2}{\rho\sum_{n=1,n\neq m}^{N_{\text{AP}}}\mu_{n,k}|\hat{\mathbf{h}}_{n,k}^T v_n|^2+1}$. For each iteration, Algorithm 4 selects the j-th AP which yields highest $\Gamma_{j\to k}$, for each k-th user. Here, $\gamma_{k,j}$ is the $\{k,j\}$-th element of the transmitter-user assignment matrix γ in Eq. (7). Eventually, the maximum-ratio-based transmit vector, i.e., $\mathbf{v}_m^{\text{MR}} = \frac{\mathbf{h}_m^*}{\|\mathbf{h}_m^*\|_2}$, $\forall m \in \{1,\cdots,N_{\text{AP}}\}$, is considered for each AP.

Appendix D
On the Trainability of the Cloud QNN

The bounds for the QNN operation utilized in transmitter-user assignment can be presented as follows. To facilitate the analysis, let us consider the normalized loss as

$$\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}}) = \left\|\frac{Q_{\text{assign}}(\gamma|\hat{\mathbf{H}},\mathbf{v}_m^{\text{MR}}) - \Phi_{\text{assign}}(\hat{\mathbf{H}})}{\Phi_{\text{assign}}(\hat{\mathbf{H}})}\right\|^2, \quad (22)$$

Algorithm 4 The Search-Based Method

Input: The acquired channel information $\left\{\hat{\mathbf{H}}^{[i]}\right\}_{i=1}^{N_{\text{data}}}$.
Output: The assignment policy γ.
Initialization:
1: Define the available APs and users as $\mathcal{A}$ and $\mathcal{U}$, respectively.
2: Set $\mathcal{S} \leftarrow \emptyset$.
Iteration:
3: **while** $|\mathcal{S}| \leq N_{\text{AP}}$ **do**
4: **for** $k \in \{1,\cdots,N_{\text{user}}\}$ **do**
5: Calculate $\Gamma_{j\to k}$, $\forall j \in \{1,\cdots,N_{\text{AP}}\}$, $j \notin \mathcal{S}$
6: Define $j^* = \arg\max_j(\gamma_{j\to k})$, $j \in \{1,\cdots,N_{\text{AP}}\}$. Set $j^* \cup \mathcal{S}$. Set $\gamma_{k,j^*} = 1$.
7: **end for**
8: **end while**

so that $\hat{\mathrm{L}}_{\text{assign}} \in [0,1]$.

Lemma 3. *The bounds of the gradient norm for the cloud QNN can be analyzed as* $\frac{1}{8}\left\{\mathrm{Tr}[x_{input}]^2 + \mathrm{Tr}[\sigma_z x_{input}\sigma_z]^2\right\} \leq \mathbb{E}_{\theta^{cloud}}\left(\left\|\nabla_{\theta^{cloud}}\hat{L}_{assign}(\gamma|\hat{\boldsymbol{H}})\right\|^2\right) \leq N_{layer}N_{neuron}$.

Proof. The upper bound can be analyzed as follows. Considering the normalization in Eq. (22), we can infer that $\left(\nabla_{\theta_{l,n}^{\text{cloud}}}\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})\right)^2 \leq 1$, where $\nabla_{\theta_{l,n}^{\text{cloud}}}\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}}) \triangleq {}^{\delta\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})}\!/_{\delta\theta_{l,n}^{\text{cloud}}}$ [104]. Moreover, the norm of the said loss gradient can be expressed as

$$\left\|\nabla_{\theta^{\text{cloud}}}\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})\right\|^2 = \sum_{l=1}^{N_{\text{layer}}}\sum_{n=1}^{N_{\text{neuron}}}\left(\frac{\delta\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})}{\delta\theta_{l,n}^{\text{cloud}}}\right)^2, \quad (23)$$

so that $\left\|\nabla_{\theta^{\text{cloud}}}\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})\right\|^2 \leq N_{\text{layer}}N_{\text{neuron}}$, where $N_{\text{neuron}} = N_{\text{AP}}$ can be considered. The lower bound can be investigated as follows. The expectation of the gradient norm in Eq (23), w.r.t. the parameter vector θ^{cloud}, $\theta^{\text{cloud}} \in [0,2\pi]^{N_{\text{layer}}N_{\text{neuron}}}$ is given by

$$\begin{aligned}&\mathbb{E}_{\theta^{\text{cloud}}}\left[\left\|\nabla_{\theta^{\text{cloud}}}\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})\right\|^2\right]\\&\quad= \sum_{l=1}^{N_{\text{layer}}}\sum_{n=1}^{N_{\text{neuron}}}\mathbb{E}_{\theta_{l,n}^{\text{cloud}}}\left[\left(\frac{\delta\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})}{\delta\theta_{l,n}^{\text{cloud}}}\right)^2\right]\\&\quad\geq \sum_{n=1}^{N_{\text{neuron}}}\mathbb{E}_{\theta_{l,1}^{\text{cloud}}}\left[\left(\frac{\delta\hat{\mathrm{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})}{\delta\theta_{l,1}^{\text{cloud}}}\right)^2\right],\end{aligned} \quad (24)$$

where x_{input} denotes the input data. In Eq. (24), the expectation of the gradient norm of the first layer is considered as a single-neuron operation, as it shall require the least amount of qubits (only one qubit required) [104]. To aid further analysis, let us represent the loss function pertinent to the cloud QNN, f^{cloud},

as:

$$\begin{aligned} f^{\text{cloud}} &= 1/2 - 1/2\,\mathrm{Tr}\left[\sigma_{(3,3,\cdots,3)}\cdot U^{\text{cloud}}\cdot x_{\text{input}}\cdot U^{\text{cloud}\dagger}\right], \\ &= 1/2 - 1/2\,\mathrm{Tr}\Big[\sigma_{(3,3,\cdots,3)}\cdot U^{\text{cloud}}_{\text{connect},N_{\text{layer}}+1}U^{\text{cloud}}_{\text{weight},N_{\text{layer}}} \\ &\quad \cdots U^{\text{cloud}}_{\text{weight},1}U^{\text{cloud}}_{\text{connect},1}\cdot x_{\text{input}}\cdot U^{\text{cloud}}_{\text{connect},1}{}^{\dagger}U^{\text{cloud}}_{\text{weight},1}{}^{\dagger}\cdots \\ &\quad U^{\text{cloud}}_{\text{weight},N_{\text{layer}}}{}^{\dagger}U^{\text{cloud}}_{\text{connect},N_{\text{layer}}+1}{}^{\dagger}\Big]. \end{aligned} \tag{25}$$

Its partial derivation pertinent to a particular l-th layer, w.r.t. $\theta^{\text{cloud}}_{l,1}$, can be obtained as

$$\begin{aligned} D_l \triangleq \frac{\delta f^{\text{cloud}}}{\delta\theta^{\text{cloud}}_{l,1}} &= \mathrm{Tr}\Big[\sigma_{(3,3,\cdots,3)}\cdot U^{\text{cloud}}_{\text{connect},N_{\text{layer}}+1}\cdots \\ &\quad W_l U^{\text{cloud}}_{\text{connect},l}\cdot\cdots\cdot x_{\text{input}}\cdot\cdots\cdot U^{\text{cloud}}_{\text{connect},l}{}^{\dagger}U^{\text{cloud}}_{\text{weight},l}{}^{\dagger}\cdots \\ &\quad U^{\text{cloud}}_{\text{connect},N_{\text{layer}}+1}{}^{\dagger}\Big]. \end{aligned} \tag{26}$$

When applying the parameter shift rule, a typical approach during QNN training, to the l-th layer, the following arbitrary variable W_l can be applied $W_l \triangleq \frac{U^{\text{cloud}}_{\text{weight},l}}{\delta\theta^{\text{cloud}}_{l,1}} = \frac{1}{2}\Big\{U^{\text{cloud}}_{\text{weight},l}(\theta^{\text{cloud}}_{l,1} - \varrho) - U^{\text{cloud}}_{\text{weight},l}(\theta^{\text{cloud}}_{l,1} + \varrho)\Big\}$, where ϱ denotes the shifting parameter, e.g., $\varrho = \pi/4$. In the least demanding scenario, the cloud QNN may served a single edge, where employing a single layer with a single neuron suffices ($N_{\text{neuron}} = 1$ and $N_{\text{layer}} = 1$). For this scenario, Eq. (25) can be represented as

$$f^{\text{cloud}}_{1,1} = \frac{1}{2} - \frac{1}{2}\,\mathrm{Tr}\left[\sigma_{(z)}\cdot \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})\cdot m_{\text{input}}\cdot \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})^{\dagger}\right], \tag{27}$$

where $m_{\text{input}} = |x_{\text{input}}\rangle\langle x_{\text{input}}|$ is the operator for input x_{input}. The derivation of Eq. (27), w.r.t. $\theta^{\text{cloud}}_{1,1}$, is given by

$$\frac{\delta f^{\text{cloud}}_{1,1}}{\delta\theta^{\text{cloud}}_{1,1}} = \mathrm{Tr}\left[\sigma_{(z)}\cdot\frac{\delta \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})}{\delta\theta^{\text{cloud}}_{1,1}}\cdot m_{\text{input}}\cdot \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})^{\dagger}\right]. \tag{28}$$

Subsequently, the expectation of the norm of the derivation in Eq. (28) is given by

$$\begin{aligned} \mathbb{E}_{\theta^{\text{cloud}}}\left[\left(\frac{\delta f^{\text{cloud}}_{1,1}}{\delta\theta^{\text{cloud}}_{1,1}}\right)^2\right] &\triangleq \mathbb{E}_{\theta^{\text{cloud}}_{1,1}}\left[\left(\frac{\delta f^{\text{cloud}}_{1,1}}{\delta\theta^{\text{cloud}}_{1,1}}\right)^2\right] \\ &= \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} A\, d\theta^{\text{cloud}}_{1,1}, \end{aligned} \tag{29}$$

where $A \triangleq \mathrm{Tr}\left[\sigma_{(z)}\cdot\frac{\delta \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})}{\delta\theta^{\text{cloud}}_{1,1}}\cdot m_{\text{input}}\cdot \mathrm{R}_y(\theta^{\text{cloud}}_{1,1})^{\dagger}\right]^2$. Considering $e^{-i\theta\sigma_i} = I\cos(\theta) - i\sigma_i\sin(\theta)$, for any parameter θ, Eq. (29) can also be articulated as

$$\begin{aligned} \mathbb{E}_{\theta^{\text{cloud}}}\left[\frac{\delta f^{\text{cloud}}_{1,1}}{\delta\theta^{\text{cloud}}_{1,1}}\right] &= \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} d\theta^{\text{cloud}}_{1,1} A \\ &= \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} d\theta^{\text{cloud}}_{1,1}\left(\cos^2(\theta^{\text{cloud}}_{1,1})\right)^2\mathrm{Tr}[m_{\text{input}}]^2 \\ &\quad - \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} d\theta^{\text{cloud}}_{1,1}\left(\sin\left(\theta^{\text{cloud}}_{1,1}\right)\cos\left(\theta^{\text{cloud}}_{1,1}\right)\right)^2\mathrm{Tr}[\sigma_z m_{\text{input}}]^2 \\ &\quad + \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} d\theta^{\text{cloud}}_{1,1}\left(\sin\left(\theta^{\text{cloud}}_{1,1}\right)\cos\left(\theta^{\text{cloud}}_{1,1}\right)\right)^2\mathrm{Tr}[\sigma_z m_{\text{input}}]^2 \\ &\quad + \frac{1}{2\pi}\int_0^{\frac{1}{2\pi}} d\theta^{\text{cloud}}_{1,1}\left(\sin^2(\theta^{\text{cloud}}_{1,1})\right)^2\mathrm{Tr}[\sigma_z m_{\text{input}}\sigma_z]^2 \\ &= \frac{1}{8}\,\mathrm{Tr}[x_{\text{input}}]^2 + \frac{1}{8}\,\mathrm{Tr}[\sigma_z m_{\text{input}}\sigma_z]^2. \end{aligned} \tag{30}$$

Therefore, the lower bound can be expressed as $\mathbb{E}_{\theta^{\text{cloud}}}\left(\left\|\nabla_{\theta^{\text{cloud}}}\hat{\mathbf{L}}_{\text{assign}}(\gamma|\hat{\mathbf{H}})\right\|^2\right) \geq \frac{1}{8}\Big\{\mathrm{Tr}[x_{\text{input}}]^2 + \mathrm{Tr}[\sigma_z x_{\text{input}}\sigma_z]^2\Big\}$. ■

REFERENCES

[1] B. Narottama and T. Q. Duong, "Quantum neural networks for optimal resource allocation in cell-free MIMO systems," in *Proc. 2022 IEEE Global Commun. Conf.*, 2022, pp. 2444–2449.

[2] E. Björnson and L. Sanguinetti, "Scalable cell-free massive MIMO systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4247–4261, Jul. 2020.

[3] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10 325–10 329, Oct. 2019.

[4] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "User-centric cell-free massive MIMO networks: A survey of opportunities, challenges and solutions," *IEEE Commun. Surv. Tutor.*, vol. 24, no. 1, pp. 611–652, first quarter 2022.

[5] J. Zhang, S. Chen, Y. Lin, J. Zheng, B. Ai, and L. Hanzo, "Cell-free massive MIMO: A new next-generation paradigm," *IEEE Access*, vol. 7, pp. 99 878–99 888, Jul. 2019.

[6] S. Chen, J. Zhang, J. Zhang, E. Björnson, and B. Ai, "A survey on user-centric cell-free massive MIMO systems," *Digital Commun. Netw.*, vol. 8, no. 5, pp. 695–719, Oct. 2021.

[7] M. Zaher, T. Demir, E. Björnson, and M. Petrova, "Learning-based downlink power allocation in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, pp. 174–188, Jan. 2022.

[8] H. Kim, T. Cho, J. Lee, W. Shin, and H. V. Poor, "Optimized shallow neural networks for sum-rate maximization in energy harvesting downlink multiuser NOMA systems," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 982–997, Apr. 2021.

[9] Y. Al-Eryani, M. Akrout, and E. Hossain, "Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1028–1042, Apr. 2021.

[10] F. Fredj, Y. Al-Eryani, S. Maghsudi, M. Akrout, and E. Hossain, "Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning," *IEEE Trans. Cognitive Commun. Netw.*, vol. 8, no. 2, pp. 1186–1201, Jun. 2022.

[11] M. Farooq, H. Q. Ngo, E.-K. Hong, and L.-N. Tran, "Utility maximization for large-scale cell-free massive MIMO downlink," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 7050–7062, Oct. 2021.

[12] Z. Liu, J. Zhang, Z. Liu, H. Du, Z. Wang, D. Niyato, M. Guizani, and B. Ai, "Cell-free XL-MIMO meets multi-agent reinforcement learning: Architectures, challenges, and future directions," *IEEE Wireless Commun.*, vol. 31, no. 4, pp. 155–162, Aug. 2024.

[13] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[14] EU Horizon 2020. Artificial intelligence aided D-band network for 5G long term evolution. https://cordis.europa.eu/project/id/871464. Accessed: Nov. 2024.

[15] H. Hojatian, J. Nadal, J.-F. Frigon, and F. Leduc-Primeau, "Decentralized beamforming for cell-free massive MIMO with unsupervised learning," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1042–1046, May 2022.

[16] M. Zaher, T. Demir, E. Björnson, and M. Petrova, "Learning-based downlink power allocation in cell-free massive MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 22, no. 1, pp. 174–188, Jan. 2023.

[17] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE Trans. Neural Netw. Learning Sys.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014.

[18] Z. Liu, X. Hu, T. Liu, X. Li, W. Wang, and F. M. Ghannouchi, "Attention-based deep neural network behavioral model for wideband wireless power amplifiers," *IEEE Microwave, Wireless Components Lett.*, vol. 30, no. 1, pp. 82–85, Jan. 2020.

[19] M. Zhang, J. Gao, and C. Zhong, "A deep learning-based framework for low complexity multiuser MIMO precoding design," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 11 193–11 206, Dec. 2022.

[20] M. Guenach, A. A. Gorji, and A. Bourdoux, "A deep neural architecture for real-time access point scheduling in uplink cell-free massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1529–1541, Mar. 2022.

[21] N. Athreya, V. Raj, and S. Kalyani, "Beyond 5G: Leveraging cell free TDD massive MIMO using cascaded deep learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 9, pp. 1533–1537, Sep. 2020.

[22] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, "Systematic literature review: Quantum machine learning and its applications," *Comput. Sci. Rev.*, vol. 51, p. 100619, Feb. 2024.

[23] W. O'Quinn and S. Mao, "Quantum machine learning: Recent advances and outlook," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 126–131, Jun. 2020.

[24] H. Abraham, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, G. Alexandrowics, E. Arbel, A. Asfaw, C. Azaustre, P. Barkoutsos, G. Barron *et al.*, "Qiskit: An open-source framework for quantum computing," 2019. [Online]. Available: https://zenodo.org/records/2562111

[25] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Modern Phy.*, vol. 95, no. 4, p. 045005, Dec. 2023.

[26] H. Liao, D. S. Wang, I. Sitdikov, C. Salcedo, A. Seif, and Z. K. Minev, "Machine learning for practical quantum error mitigation," *Nature Machine Intell.*, pp. 1–9, Nov. 2024.

[27] Qureca. Overview on quantum initiatives worldwide – update mid 2021. https://www.qureca.com/overview-on-quantum-initiatives-worldwide-update-mid-2021. Accessed: Nov. 2024.

[28] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nat. Comput. Sci.*, vol. 1, no. 6, pp. 403–409, Jun. 2021.

[29] A. K. Singh, D. Saxena, J. Kumar, and V. Gupta, "A quantum approach towards the adaptive prediction of cloud workloads," *IEEE Trans. Parallel, Distributed Sys.*, vol. 32, no. 12, pp. 2893–2905, Dec. 2021.

[30] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46 317–46 350, Apr. 2019.

[31] N. H. Nguyen, E. C. Behrman, M. A. Moustafa, and J. E. Steck, "Benchmarking neural networks for quantum computations," *IEEE Trans. Neural Netw. Learning Sys.*, vol. 31, no. 7, pp. 2522–2531, Jul. 2020.

[32] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "Expressive power of parametrized quantum circuits," *Phy. Rev. Research*, vol. 2, no. 3, p. 033125, Jul. 2020.

[33] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," *Nature commun.*, vol. 12, no. 1, p. 2631, May 2021.

[34] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, "Machine learning of high dimensional data on a noisy quantum processor," *npj Quantum Info.*, vol. 7, no. 1, p. 161, Nov. 2021.

[35] B. Narottama and S. Y. Shin, "Quantum neural networks for resource allocation in wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 1103–1116, Feb. 2022.

[36] B. Narottama, T. Jamaluddin, and S. Y. Shin, "Quantum neural network with parallel training for wireless resource optimization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5835–5847, May. 2024.

[37] T. Wang, P. Li, Y. Wu, L. Qian, Z. Su, and R. Lu, "Quantum-empowered federated learning in space-air-ground integrated networks," *IEEE Netw.*, vol. 38, no. 1, pp. 96–103, Jan. 2024.

[38] A. Farouk, N. A. Abuali, and S. Mumtaz, "Quantum-computing-based channel and signal modeling for 6G wireless systems," *IEEE Commun. Mag.*, vol. 62, no. 2, pp. 64–70, Feb. 2024.

[39] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," in *Proc. IEEE Global Commun. Conf.*, 2021, pp. 01–06.

[40] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[41] A. Kumar, V. Tentu, D. N. Amudala, and R. Budhiraja, "WSEE optimization of cell-free mMIMO uplink using deep deterministic policy gradient," *IEEE Commun. Lett.*, vol. 27, no. 1, pp. 219–223, Jan. 2023.

[42] Z. Liu, J. Zhang, B. Xu, D. W. K. Ng, A. Nallanathan, and B. Ai, "GCN-based low-complexity downlink beamforming for cell-free massive MIMO systems with partially coherent joint transmission," *IEEE Trans. Wireless Commun.*, 2025, [Early Access].

[43] S. Li, H.-C. Yang, and F. Hu, "Joint transmission mode selection and scheduling for AoI minimization in NOMA-capable WP-IoT networks: A deep transfer learning solution," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5805–5816, Aug. 2025.

[44] J. Zhang, G. Zheng, T. Koike-Akino, K.-K. Wong, and F. A. Burton, "Hybrid quantum-classical neural networks for downlink beamforming optimization," *IEEE Trans. Wireless Commun.*, vol. 23, no. 11, pp. 16 498–16 512, Nov. 2024.

[45] B. Zhang, G. Zheng, and N. V. Huynh, "A hybrid quantum-classical autoencoder framework for end-to-end communication systems," *IEEE Wireless Commun. Lett.*, vol. 14, no. 3, pp. 806–810, Mar. 2025.

[46] S. Park, S. Jung, and J. Kim, "Dynamic quantum federated learning for satellite-ground integrated systems using slimmable quantum neural networks," *IEEE Access*, vol. 12, pp. 58 239–58 247, Apr. 2024.

[47] A. Jaiswal, S. Kumar, O. Kaiwartya, P. K. Kashyap, E. Kanjo, N. Kumar, and H. Song, "Quantum learning-enabled green communication for next-generation wireless systems," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 3, pp. 1015–1028, Sep. 2021.

[48] D. Xu and P. Ren, "Quantum learning based nonrandom superimposed coding for secure wireless access in 5G URLLC," *IEEE Trans. Info. Forensics Security*, vol. 16, pp. 2429–2444, Feb. 2021.

[49] Y. Li, A. H. Aghvami, and D. Dong, "Intelligent trajectory planning in UAV-mounted wireless networks: A quantum-inspired reinforcement learning perspective," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1994–1998, Sep. 2021.

[50] Z. Qu, W. Shi, B. Liu, D. Gupta, and P. Tiwari, "IoMT-based smart healthcare detection system driven by quantum blockchain and quantum neural network," *IEEE J. Biomedical, Health Info.*, vol. 28, no. 6, pp. 3317–3328, Jun. 2024.

[51] Y.-Y. Hong, D. Josh Domingo Lopez, and Y.-Y. Wang, "Solar irradiance forecasting using a hybrid quantum neural network: A comparison on GPU-based workflow development platforms," *IEEE Access*, vol. 12, pp. 145 079–145 094, Oct. 2024.

[52] P. Botsinis, D. Alanis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum search algorithms for wireless communications," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1209–1242, second quarter 2019.

[53] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46 317–46 350, Apr. 2019.

[54] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "6G wireless systems: A vision, architectural elements, and future directions," *IEEE Access*, vol. 8, pp. 147 029–147 044, Aug. 2020.

[55] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Comput. Sci.*, vol. 1, no. 6, pp. 403–409, Jun. 2021.

[56] Y. Yang, Z. Zhang, A. Wang, X. Xu, X. Wang, and Y. Li, "Maximizing quantum-computing expressive power through randomized circuits," *Phy. Rev. Research*, vol. 6, no. 2, p. 023098, Apr. 2024.

[57] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Trans. Veh. Technol.*, Aug. 2024.

[58] A. Tudisco, D. Volpe, G. Ranieri, G. Curato, D. Ricossa, M. Graziano, and D. Corbelletto, "Evaluating the computational advantages of the variational quantum circuit model in financial fraud detection," *IEEE Access*, vol. 12, pp. 102 918–102 940, Jul. 2024.

[59] D. Konar, S. Bhattacharyya, B. K. Panigrahi, and E. C. Behrman, "Qutrit-inspired fully self-supervised shallow quantum learning network for brain tumor segmentation," *IEEE Trans. Neural Netw. Learning Sys.*, vol. 33, no. 11, pp. 6331–6345, Nov. 2021.

[60] S. Kasi, A. K. Singh, D. Venturelli, and K. Jamieson, "Quantum annealing for large MIMO downlink vector perturbation precoding," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[61] Y. Zhang, P. Khanduri, I. Tsaknakis, Y. Yao, M. Hong, and S. Liu, "An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning," *IEEE Signal Process. Mag.*, vol. 41, no. 1, pp. 38–59, Apr. 2024.

[62] A. Thebelt, C. Tsay, R. M. Lee, N. Sudermann-Merx, D. Walz, T. Tranter, and R. Misener, "Multi-objective constrained optimization for energy applications via tree ensembles," *Appl. Energy*, vol. 306, p. 118061, Jan. 2022.

[63] I. Chafaa, G. Bacci, and L. Sanguinetti, "Transformer-based power optimization for max-min fairness in cell-free massive MIMO," *IEEE Wireless Commun. Lett.*, May 2025.

[64] B. M. Lee, "Cell-free massive MIMO for massive low-power internet of things networks," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6520–6535, May 2022.

[65] R. Jiao, L. Dai, W. Wang, F. Lyu, N. Cheng, and X. Shen, "Max-min fairness for beamspace MIMO-NOMA: From single-beam to multi-beam," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 739–752, Feb. 2022.

[66] J. Zhao, X. Yue, S. Kang, and W. Tang, "Joint effects of imperfect CSI and SIC on NOMA based satellite-terrestrial systems," *IEEE Access*, vol. 9, pp. 12 545–12 554, Jan. 2021.

[67] J. Zheng, Z. Zhao, J. Zhang, J. Cheng, and V. C. M. Leung, "Performance analysis of cell-free massive MIMO systems with asynchronous reception," in *Proc. IEEE Globecom Workshops*, 2022, pp. 190–195.

[68] R. P. Antonioli, I. M. Braga, G. Fodor, Y. C. Silva, and W. C. Freitas, "Mixed coherent and non-coherent transmission for multi-CPU cell-free systems," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 1068–1073.

[69] C. D'Andrea, T. Foggi, A. Piemontese, A. Ugolini, S. Buzzi, and G. Colavolpe, "Coherent vs. non-coherent joint transmission in cell-free user-centric non-terrestrial wireless networks," in *Proc. IEEE Int. Workshop Sig. Process. Advan. Wireless Commun.*, 2024, pp. 636–640.

[70] A. F. Molisch, M. Z. Win, Y.-S. Choi, and J. H. Winters, "Capacity of MIMO systems with antenna selection," *IEEE Trans. Wireless Commun.*, vol. 4, no. 4, pp. 1759–1772, Jul. 2005.

[71] O. T. Demir, E. Björnson, and L. Sanguinetti, *Foundations of User-Centric Cell-Free Massive MIMO*, 2021.

[72] M. Sarker and A. O. Fapojuwo, "Access point-user association and auction algorithm-based pilot assignment schemes for cell-free massive MIMO systems," *IEEE Sys. J.*, vol. 17, no. 3, pp. 4301–4312, Sep. 2023.

[73] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. Srinivas, "Resource allocation and scheduling in non-coherent user-centric cell-free MIMO," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[74] H. A. Ammar, R. Adve, S. Shahbazpanahi, G. Boudreau, and K. V. Srinivas, "Downlink resource allocation in multiuser cell-free MIMO networks with user-centric clustering," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1482–1497, Mar. 2022.

[75] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Sig. Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[76] F. Sun, V. O. Li, and Z. Diao, "Modified bipartite matching for multiobjective optimization: Application to antenna assignments in MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1349–1355, Mar. 2009.

[77] Z. Wang, Y. Cao, D. Zhang, X. Hua, P. Gao, and T. Jiang, "User selection for MIMO downlink with digital and hybrid maximum ratio transmission," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 11 101–11 105, Oct. 2021.

[78] T. T. Vu, D. T. Ngo, N. H. Tran, H. Q. Ngo, M. N. Dao, and R. H. Middleton, "Cell-free massive MIMO for wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6377–6392, Oct. 2020.

[79] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Phy. Rev. A*, vol. 102, no. 3, p. 032420, Sep. 2020.

[80] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Inf.*, vol. 5, no. 103, pp. 1–9, Nov. 2019.

[81] J.-Y. Ko and Y.-H. Lee, "Random beamforming in spatially correlated multiuser MISO channels," in *Proc. IEEE Veh. Tech. Conf.*, Calgary, Canada, Sep. 2008, pp. 1–5.

[82] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[83] M. Bianchini and F. Scarselli, "On the complexity of neural network classifiers: A comparison between shallow and deep architectures," *IEEE Trans. Neural Networks, Learning Sys.*, vol. 25, no. 8, pp. 1553–1565, Aug. 2014.

[84] J. Chen, J. Zhu, and L. Song, "Stochastic training of graph convolutional networks with variance reduction," in *Proc. Int. Conf. Machine Learning*. PMLR, 2018, pp. 942–950.

[85] G. Li, M. Müller, B. Ghanem, and V. Koltun, "Training graph neural networks with 1000 layers," in *Proc. Int. Conf. Machine Learning*. PMLR, 2021, pp. 6437–6449.

[86] M. Alam and S. Ghosh, "DeepQMLP: A scalable quantum-classical hybrid deep neural network architecture for classification," in *2022 Int. Conf. VLSI Design and Int. Conf. Embedded Sys.*, 2022, pp. 275–280.

[87] N. Q. T. Thoong, A. A. Cheema, S. R. Khosravirad, O. A. Dobre, and T. Q. Duong, "Channel estimation for reconfigurable intelligent surface-aided 6G NOMA systems using CNN-based quantum LSTM model," in *Proc. IEEE Vehi. Technol. Conf.*, 2024, pp. 1–5.

[88] M. B. Shahab, M. F. Kader, and S. Y. Shin, "A virtual user pairing scheme to optimally utilize the spectrum of unpaired users in non-orthogonal multiple access," *IEEE Signal Process. Lett.*, vol. 23, no. 12, pp. 1766–1770, Dec. 2016.

[89] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020.

[90] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10 325–10 329, Oct. 2019.

[91] F. Jiang, Z. Gu, C. Sun, and R. Ma, "Dynamic user pairing and power allocation for NOMA with deep reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–6.

[92] F. Fredj, Y. Al-Eryani, S. Maghsudi, M. Akrout, and E. Hossain, "Distributed beamforming techniques for cell-free wireless networks using deep reinforcement learning," *IEEE Trans. Cognitive Commun., Netw.*, vol. 8, no. 2, pp. 1186–1201, Jun. 2022.

[93] Z. He, B. Peng, Y. Alexeev, and Z. Zhang, "Distributionally robust variational quantum algorithms with shifted noise," *IEEE Trans. Quantum Engineer.*, vol. 5, pp. 1–12, Jun. 2024.

[94] T. Winker, S. Groppe, V. Uotila, Z. Yan, J. Lu, M. Franz, and W. Mauerer, "Quantum machine learning: Foundation, new techniques, and opportunities for database research," in *Proc. Companion 2023 Internat. Conf. Manag. of Data*, 2023, pp. 45–52.

[95] K. Chinzei, S. Yamano, Q. H. Tran, Y. Endo, and H. Oshima, "Trade-off between gradient measurement efficiency and expressivity in deep quantum neural networks," *npj Quantum Info.*, vol. 11, no. 1, p. 79, 2025.

[96] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, "Diagnosing barren plateaus with tools from quantum optimal control," *Quantum*, vol. 6, p. 824, Sep. 2022.

[97] A. Liu and V. K. N. Lau, "Joint BS-user association, power allocation, and user-side interference cancellation in cell-free heterogeneous networks," *IEEE Trans. Sig. Process.*, vol. 65, no. 2, pp. 335–345, Jan. 2017.

[98] Y. Al-Eryani, M. Akrout, and E. Hossain, "Multiple access in cell-free networks: Outage performance, dynamic clustering, and deep reinforcement learning-based design," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1028–1042, Apr. 2021.

[99] C. Park, W. J. Yun, J. P. Kim, T. K. Rodrigues, S. Park, S. Jung, and J. Kim, "Quantum multiagent actor–critic networks for cooperative mobile access in multi-UAV systems," *IEEE Internet Things J.*, vol. 10, no. 22, pp. 20 033–20 048, Nov. 2023.

[100] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, "Distributed quantum computing: a survey," *Comput. Netw.*, vol. 254, p. 110672, Dec. 2024.

[101] A. S. Cacciapuoti, J. Illiano, M. Viscardi, and M. Caleffi, "Multipartite entanglement distribution in the quantum internet: Knowing when to stop!" *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 6, pp. 6041–6058, Dec. 2024.

[102] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, no. 3, p. 032309, Sep. 2018.

[103] M. Ostaszewski, E. Grant, and M. Benedetti, "Structure optimization for parameterized quantum circuits," *Quantum*, vol. 5, p. 391, Jan. 2021.

[104] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, "Toward trainability of quantum neural networks," *arXiv preprint arXiv:2011.06258*, Nov. 2020, DOI:10.48550/arXiv.2011.06258.