Quantum Graph Neural Network for Joint Optimization of Pinching and Fluid Antenna Systems

Okzata Recy*, Bhaskara Narottama*, Simon L. Cotton†, Trung Q. Duong*,†
*Memorial University, Canada, e-mail:{orecy, bnarottama, tduong}@mun.ca
†Queen's University Belfast, U.K., e-mail: {trung.q.duong, simon.cotton}@qub.ac.uk

***Abstract*—This work optimizes a flexible antenna environment, that incorporates both pinching antenna systems (PASS) and fluid antenna systems (FAS) at the transmitter and the user sides. Optimizing PASS and FAS presents a non-trivial challenge, as the wireless network exhibits a highly dynamic topology and an extensive solution search space. To address this, this work proposes a quantum graph neural network (QGNN)-based solution workflow designed to optimize users' quality of service by learning the spatial and structural dependencies between the pinching antenna elements and user ports as a quantum state. The QGNN approach encodes the wireless network topology, encompassing both edges and vertex relationships, into the quantum state space, and performs optimization through a tunable single-qubit rotation gate. Overall, the QGNN-based solution shows a faster loss reduction especially during the early learning episodes compared to the baseline model. The QGNN achieves an average training loss approximately $3.7\times$ lower than that of the typical parameter quantum circuit model during model training. In summary, this work points out the feasibility of applying graph-based, data-driven learning, realized through the quantum graph neural network workflow, to the problem of flexible antenna optimization.**

***Index Terms*—Flexible antennas; graph neural networks; quantum computing; wireless system management.**

I. Introduction

IN RECENT years, the transition from fixed to flexible antenna designs has drawn significant research attention. Fixed antennas are approaching their performance limits, even as the demands of next-generation wireless networks grow substantially: While increasing the number of fixed antennas can improve wireless system performance, it introduces substantial computational burden and physical form factor, such as those required for extremely large-scale massive multiple-input multiple-output (MIMO) systems. Beyond conventional architectures, MIMO systems have evolved to encompass a wide range of reconfigurable electromagnetic structures, notably structurally reconfigurable antenna systems, known as flexible antenna systems (e.g., pinching and fluid antennas), which facilitate dynamic adjustment of antenna position and/or geometry, thereby optimizing signal propagation and channel conditions. By physically adjusting the effective antenna position, these systems introduce an additional spatial degree of freedom without incurring the excessive physical footprint associated with conventional fixed antenna arrays. Studies have explored flexible antenna systems, particularly pinching antenna systems (PASS), in areas such as beamforming optimization [1], physical layer security [2], and channel estimation [3]. Complementing this line of research, studies on fluid antenna systems (FAS) support the development of flexible antenna design (including port selection and antenna design [4], and modeling FAS under various correlation scenarios [5]. As flexible antennas remain in an exploratory phase, the joint implementation of PASS and FAS is largely underexplored, and presents challenges related to optimization complexity [6].

In response to these optimization challenges, data-driven wireless system optimization based on machine learning has become a new approach to anticipating the growing number of network elements, many of which are difficult (or even intractable) to address analytically. For PASS-FAS cases, this notion can be advanced as graph-based learning, by which the dynamic structures of the moving antennas can inherently be represented as topological structures. These topologies consist of vertices and edges, connecting correlated parts to form graphs. In a flexible antenna environment, specifically in a PASS-FAS topology, such a graph-oriented conception reflect the relationship between a base station (BS) and a user equipment (UE). This, in turn, allows the use of a graph-based learning architecture, including graph neural networks (GNNs). For example, the study in [7] utilizes graph-based learning to represent and optimize meta-material absorber designs, while [8] employs a GNN for joint beamforming and antenna selection optimization. However, graph-based machine learning approaches typically employ classical neural networks with a substantial number of parameters and operations, potentially resulting in considerable processing overhead.

Conveniently, the growing maturity of quantum computing, from both hardware and algorithmic perspectives, has made complex optimization via hybrid quantum-classical learning feasible (e.g., [9]). Related thereto, many studies have demonstrated the potential of quantum graph-based learning and optimization across diverse domains, including transportation traffic prediction [10], electrical grid optimization [11], and molecular classification [12]. However, existing literature reveals limited exploration on quantum graph-based learning adoption to FAS optimization, particularly at both the BS and the UE. This work, therefore, introduces a quantum graph neural network (QGNN)-based solution. Its principal contributions, along with the identified research gaps and challenges, are described as follows:

- This work initiates the study of graph-based learning for flexible antenna environments, particularly PASS and FAS at both the BS and the UE. This graph-based learning translates the vertices and edges according to their topological representation, such that each component of the graph effectively depicts the structure of the flexible antenna environment. As far as can be ascertained from the existing literature, this is an early study that uses a learning-based approach to exploits a flexible antenna environment, particularly a PASS-FAS configuration, though a graph representation.[1]
- This work demonstrates graph extraction to the quantum circuit, forming a quantum graph neural network (QGNN). The QGNN captures the graph relation through controlled-unitary operations: The graph's vertices represents PASS-FAS pairs, while the edges represents information extracted from the wireless network, particularly channel state information. Thus, the QGNN encodes not only feature input information, but also network topology from the wireless environment.[2] The QGNN's parameters are trained using the parameter-shift rule.
- The numerical results are presented to validate the effectiveness of the proposed QGNN-based solution. Furthermore, the proposed QGNN-based solution aims to maximize the user data rate through a quantum learning workflow, jointly optimizing the selection of BS's pinching antenna (PA) elements and the user's fluid antenna (FA) ports. The results show that the QGNN exhibits a steeper decline in loss during the early learning episodes and eventually achieve a higher rate, demonstrating capability to capture channel information and the structure of the PASS-FAS environment.

Notations. The following notation is adopted throughout this work: Bold-uppercase and lowercase letters represent matrices and vectors, respectively. The symbols $(\cdot)^{\mathsf{H}}$ and $(\cdot)^{\mathsf{T}}$ represent Hermitian and transpose matrices, respectively. The sets of complex numbers and positive integers are denoted as $\mathbb{C}$ and $\mathbb{Z}_{+}$, respectively. Assuming an arbitrary complex number b, its complex conjugate is given by b^{*}. The Kronecker product is indicated by the symbol $\otimes$. $\|\cdot\|_2$ represents the Euclidean norm.

II. Wireless PASS and FAS Environments

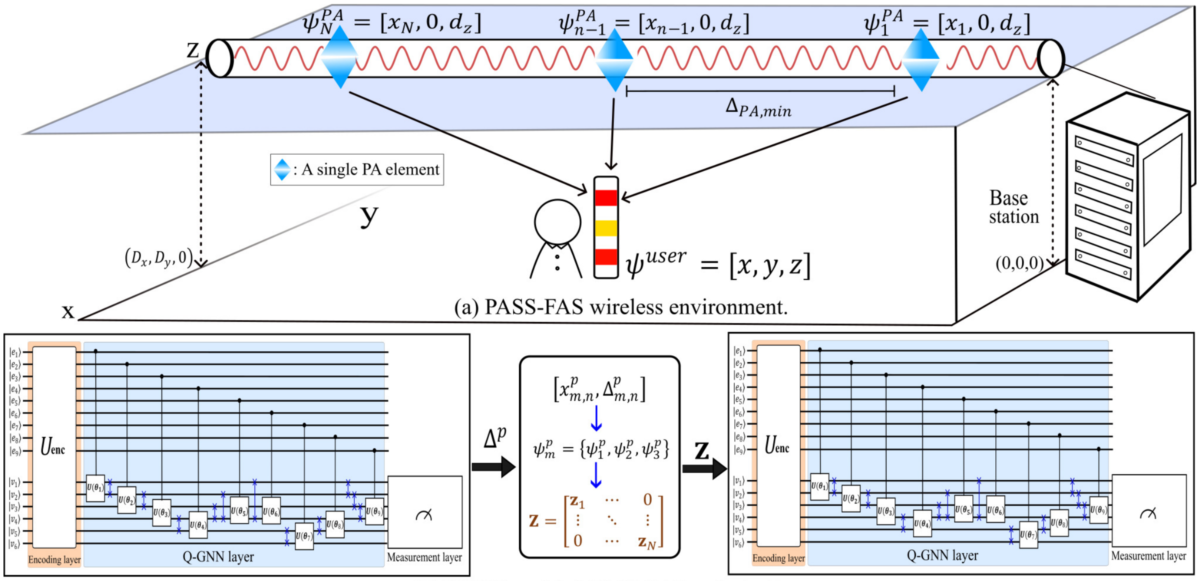
The following section provides a detailed description of the wireless environment for PASS and FAS networks. Let us consider downlink communication links with a BS's PASS waveguide mounted with N PA elements and a UE with M ports, operating at carrier frequency of $f = 28$ GHz, as shown in Fig. 1 (a). This work focuses on indoor PASS deployment, particularly ceiling installation. To be specific, we consider the wireless environment in a square horizontal plane, x, y, with width $D_x = 10$ m and length $D_y = 10$ m. The waveguide is aligned with the x axis and installed at a height of $D_z = 3$ m above the floor level. In addition, each of PA elements can move along the waveguide within the said two-dimensional (2D) region, according to the current user location[3]. We aim to identify the suitable position of each PA element (typically closer to the user) so that each element exhibits phase-shift delays relative to its neighboring elements. The user, at each time step, is assumed to be distributed according to the bivariate uniform distribution within the coordinates $\psi_t^{\text{user}} = [x_t^{\text{user}}, y_t^{\text{user}}, z_t^{\text{user}}]$, where $x_t^{\text{user}} = [0, D_x]$ and $y_t^{\text{user}} = [0, D_y]$. The UE height remains fixed at $z_t^{\text{user}} = 0.5$ m. The system also carefully maintains equal transmit power across all PA elements, so that the attenuation along the waveguide is neglectable.

The baseband signal is modulated and fed into the corresponding waveguide via a dedicated radio-frequency (RF) chain. The transmitted signal at the n-th PA elements can then be written as $\mathbf{s}_n = \mathbf{Z}_n \cdot \mathbf{v} \cdot x \in \mathbb{C}^{N \times 1}$, where x denotes the transmitted symbol at particular n port, $\mathbf{v} \in \mathbb{C}^{N \times 1}$ represents the transmit beamforming vector, and $\mathbf{Z} = \text{diag}(\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N) \in \mathbb{C}^{N \times N}$ indicates the in-waveguide channel response from the feed point to the each PA element, with each entry given by $\mathbf{z}_{m,n} = e^{\frac{-\mathrm{j}2\pi\epsilon_{\text{ER}}}{\lambda}} \left\|\psi_{n=0}^{\text{PA}} - \psi_n^{\text{PA}}\right\| \in \mathbb{C}$. Here, the elements of $\mathbf{Z}$, each of which denoted by $\mathbf{z}_{m,n}$, corresponds to a phase shift delay (θ_g) caused by the propagation distance between the feed point (indicated by the index $n = 0$) and the (m,n)-th PA elements. The wavelength is defined as $\lambda = c/f$, given the light of speed $c = 3 \times 10^8$ m/s, while ϵ_{ER} denotes the effective reactive index, approximated as 1.4. Therefore, the channel coefficient between a particular PA element, located at ψ_n^{PA}, and its corresponding m-th FA

[1] In comparison, many studies (e.g., [13], [14]) investigate learning-based optimization on the BS side, while others (e.g., [15], [16]) address the user side.

[2] There are pertinent studies that have explored graph-based learning: The study in [17] utilizes the classical GNN with continual learning to solve problems in a dynamic environment. [18] presents robust controllability learning via generating a network through spatial GNN to maintain high precision between the training and testing datasets. [19] employs deep graph reinforcement learning to achieve an improved secrecy rate in physical-layer security. Despite these advancements, the adoption of quantum graph-based learning remains unexplored.

[3] This work considers slow-fading channels, such that adjusting antenna positions occurs significantly faster than the channel coherence time. Other deployment scenarios (e.g. outdoor building) and alternative frequency operations (e.g., millimeter wave and sub-terahertz) can also be accommodated for future studies.



$\psi_N^{PA} = [x_N, 0, d_z]$
$\psi_{n-1}^{PA} = [x_{n-1}, 0, d_z]$
$\psi_1^{PA} = [x_1, 0, d_z]$
z
$\Delta_{PA,min}$
: A single PA element
y
Base station
$(D_x, D_y, 0)$
$\psi^{user} = [x, y, z]$
(0,0,0)
x
U_{enc}
Encoding layer
Q-GNN layer
Measurement layer
Δ^p
$[x_{m,n}^p, \Delta_{m,n}^p]$
$\psi_m^p = \{\psi_1^p, \psi_2^p, \psi_3^p\}$
$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{z}_N \end{bmatrix}$
Z

(a) PASS-FAS wireless environment.

(b) QGNN-enabled PASS-FAS optimization.

Fig. 1: An Illustration of (a) the considered PASS-FAS environment, where N movable PA elements are deployed along a waveguide and (b) the QGNN architecture for joint optimization of PA beamforming and port selection via a quantum learning workflow.

port, located at ψ_m^{FA}, can be written as

$$h_{m,n} = \sqrt{\eta}\frac{e^{-\mathrm{j}\frac{2\pi}{\lambda}\|\psi_m^{\text{FA}}-\psi_n^{\text{PA}}\|}}{\|\psi_m^{\text{FA}}-\psi_n^{\text{PA}}\|}, \tag{1}$$

where the path loss is $\eta = \left(\frac{\lambda}{4\pi}\right)^2$. Then, concatenating all the channel coefficients, each of which denoted by $h_{m,n}$, forms the channel matrix $\mathbf{H} \in \mathbb{C}^{M\times N}$.

PASS-FAS Optimization. Each PA elements transmit its modulated symbol through the corresponding RF chain, while each FAS port receives its own signal. Then, the achieved user rate serves as a key indicator of the PASS-FAS performance, with the achievable system sum rate formulated as

$$\mathsf{R} = \sum_{m=1}^{M} \log_2\left(1 + \frac{|\mathbf{h}_m^{\mathsf{H}}\,\mathbf{Z}\,\mathbf{p}_m|^2}{\sigma_m^2}\right), \tag{2}$$

where $\mathbf{h}_m$ is the channel vector from all PA elements to the m-th FAS user port, $\mathbf{p}_m$ denotes the selected m-th FAS port, and σ_m^2 represents the noise variance. The objective is to maximize R by jointly optimizing the PASS beamforming matrix $\mathbf{Z}$ and port selection $\mathbf{P}$, which can be expressed as

$$\underset{\mathbf{Z},\mathbf{P}}{\text{maximize}} \quad \mathsf{R} \tag{3a}$$

$$\text{subject to} \quad 0 \le x_{\text{PA},n,j} \le L_{\text{WG}} \quad \forall n, \tag{3b}$$

$$\triangle_{\text{PA},n} \ge \triangle_{\text{PA,min}} \quad \forall n, \tag{3c}$$

$$p_m \in \mathcal{P}_{\text{opt}}, \quad \forall m, \tag{3d}$$

$$\|\mathbf{Z}\mathbf{V}\|^2 \le P_{\text{max}}. \tag{3e}$$

These constraints ensure the following: (i) Each PA element remains within the waveguide length L_{WG}, as conveyed in (3b). (ii) A minimum spacing of $\Delta_{\text{PA,min}} = \lambda/\epsilon_{\text{ER}}$ is maintained, as described in (3c). (iii) The selected ports must belong to the feasible set of $\mathcal{P}_{\text{opt}}$, as specified in (3d). (iv) The total transmit power is limited according to (3e).

III. Utilizing a Quantum Graph Neural Network

This section discusses the integration of the PASS-FAS environment with the hybrid quantum-classical machine learning workflow, particularly the QGNN learning workflow, developed to address PASS-FAS optimization problems. The key idea of the QGNN is to represent information and relationships among wireless network entities as a graph structure. In this context, the wireless PASS and FAS environment is modeled as a graph of $G = (\mathcal{V}, \mathcal{E})$, where the vertices and edges capture information and connections within the environment. Throughout this work, we represent the mutual interactions between PASS and FAS, such as channel coefficients, as edges, while the individual PASS and FAS elements are represented by vertices. For a concrete example, let the set of PASS and FAS elements be $\mathcal{V} = \{v_1, v_2, \cdots, v_{m+n}\}$, where $v_1 \cdots v_m$ correspond to PA elements and $v_{m+1} \cdots v_{m+n}$ correspond to FAS ports. Then, the adjacent matrix capturing the PASS-FAS topology is presented by an edge-weight matrix $\mathcal{E} \in \mathbb{C}^{J\times J}$, where $J = M \times N$ corresponds to the number of network elements within the graph. Each element of $\mathcal{E}$ is defined as

$$\mathcal{E}_{i,j} = \begin{cases} e_{(i-1)M+(j-M)}, & \text{if } i \le M,\ j > M, \\ e_{(j-1)M+(i-M)}, & \text{if } i > M,\ j \le M, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where e_k denotes the channel coefficient representing the interaction between the n-th PA elements with m-th FAS ports, with $\{e_k\} \in \mathbf{H}^{M\times N}$.

State Preparation and Encoding Process. As the information is intially in the classical domain, it needs to be prepared for processing in the quantum domain. To do so, we prepare two quantum registers. The first, the edge register $|e\rangle$, encodes the information from edges into a quantum state, and requires $M \times N$ qubits. The second, the vertex register $|v\rangle$, requires $M+N$ qubits and translates the classical information from the vertices into quantum states. We applies angle encoding (e.g., angular rotations around y- and z-axes), to both the edge and vertex registers, to transform the classical data into quantum bits in Hilbert space. Without loss of generality, the encoding process in the edge circuit uses channel coefficient h as a feature input. Its real and imaginary parts are separated, and mapped into quantum angle rotations, followed by X gates to activate all qubits in this register[4]. The overall operation for the encoding process can be expressed as follows.

$$\mathsf{U}_{\text{enc}} = \bigotimes_{m=1}^{M} \bigotimes_{n=1}^{N} \mathsf{X}_{m,n,t} \, \mathsf{R}_z(\mathfrak{Im}(h_{m,n,t})) \, \mathsf{R}_y(\mathfrak{Re}(h_{m,n,t})), \tag{5}$$

where $\mathsf{R}_y(\mathfrak{Re}(h_{m,n,t}))$ and $\mathsf{R}_z(\mathfrak{Im}(h_{m,n,t}))$ denote rotations around the y- and z-axes of the Bloch sphere, corresponding to the real and imaginary parts of $h_{m,n,t} \in \mathbb{C}$, respectively. Similar to the edge circuit, we apply angle encoding with an identical value to prepare the state in the vertex circuit. These identical values in the vertex circuit represent similar entities (e.g., PA waveguide elements and FAS ports), and symmetry in their initial state avoids introducing bias. All PA waveguides and FAS port pairs share the same physical configuration and parameter (each pair has one encoded edge information). By encoding them identically, all vertices start from a common reference state in the Hilbert space, such that any subsequence differences in their states arise solely from the learned message passing and graph-embedding through the QGNN circuit.

Quantum Graph Neural Network. Having discussed the encoding layer U_{enc}, we now focus on the QGNN layer, detailing its design and operation. In this layer, we introduce the controlled-unitary operation parameterized by a learnable variable, denoted by $\mathsf{CU}(\theta)$. Unlike the trainable layers in typical parameterized quantum circuit (PQC) design, the control qubit of $\mathsf{CU}(\theta)$ resides in the edge circuit, while the corresponding unitary operation, incorporating two qubits, is applied within the vertex circuit (u_m, v_n). These particular operations extract the information from the wireless topology elements in (4), forming a quantum graph embedding. The parameterized controlled-unitary operation can be compactly written as

$$\mathsf{U}_{\text{QGNN}}(\theta) = \prod_{m=1}^{M} \prod_{n=1}^{N} \mathsf{CU}_{e_{m,n}(u_m,v_n)}(\theta_{m,n}), \tag{6}$$

with each gate defined as

$$\begin{aligned} \mathsf{CU}_{e_{m,n}(u_m,v_n)}(\theta_{m,n}) = |0\rangle\langle 0|_{e_{m,n}} \otimes I_{u_m,v_n} + \\ |1\rangle\langle 1|_{e_{m,n}} \otimes \left(\mathsf{CZ}_{u_m \to v_n} \, \mathsf{R}_y(\theta_{m,n}) \, \mathsf{CZ}_{v_n \to u_m}\right). \end{aligned} \tag{7}$$

Once information is transferred through the quantum domain, the qubits cannot be interrupted during computation. To ensure that the CU gates preserve the corresponding u_m and v_n, representing the PA waveguide and the FAS user port, the *SWAP* gates are included within the QGNN layer[5].

Quantum Circuit Measurements. Following the QGNN layer, the measurement process is applied at the end of the circuit. We observe the particular information in the circuit by projecting it onto an observable operator $\mathbb{M}$, resulting in classical outcomes that can be translated into optimization variables, such as $\mathbf{Z}$ and $\mathbf{P}$. To accomplish this $M \cdot N/2 = 3$ chosen qubits associated with the PA waveguides are observed (measured) in the vertex register: The expectation value of measuring a specific $\mathbb{M}$ can be defined using the trace operator $\mathsf{Tr}(\cdot)$, as in

$$\mathsf{Tr}\left(\mathbb{M} \cdot \mathsf{U}_{\text{QGNN}} \cdot |\psi_{\text{enc}}\rangle \langle\psi_{\text{enc}}| \cdot \mathsf{U}_{\text{QGNN}}^{\dagger}\right). \tag{8}$$

Here, $|\psi_{\text{enc}}\rangle$ encodes the PAS-FAS graph structure, and U_{QGNN} processes this information according to the graph topology. Then, by applying the measurement operator $\mathbb{M}$ along the z-axis to the i-th encoded output state $|\psi_{\text{enc}}\rangle$, the information is projected onto the computation basis $|0\rangle$. Finally, the predicted measurement outcome of the QGNN is obtained by taking the trace of the resulting density operator, $\mathsf{U}_{\text{QGNN}} \cdot |\psi_{\text{enc}}\rangle \langle\psi_{\text{enc}}| \cdot \mathsf{U}_{\text{QGNN}}^{\dagger}$, considering the measurement operator $\mathbb{M}$. Furthermore, this work uses an unsupervised approach to perform a hybrid quantum machine learning workflow. The loss function $\mathcal{L}_n^{\text{PA}}\left(\theta_i^{(t)}, \mathbf{Z}, \mathbf{P}; \mathbf{H}^{(t)}\right)$, then, is defined as the negative achieved user rate pertinent to (3). As a result, minimizing training loss aligns with maximizing the objective function in (3). Since the mobility of the FAS user is assumed to be limited, such an unsupervised approach remains effective.

Output of the QGNN. As shown in Fig. 1 (b), the learning workflow operates in two stages of execution: in the first stage, the PA beamforming matrix $\mathbf{Z}$ is obtained, and in the second stage, the port selection $\mathbf{P}$ is determined at the UE side. Consequently, the QGNN produces the following optimization variables (as formulated in (3)):

1) *With respect to PA elements:* The position matrix $\mathbf{\Psi}_t$ represents the coordinates of N PA elements deployed along a single waveguide and is expressed as

$$\mathbf{\Psi_t} = \left[\psi_{1,t} \, \cdots \, \psi_{N,t}\right]^{\mathsf{T}}, \tag{9}$$

$$\psi_{n,t}^{\text{PA}} = [x_{n,t}^{\text{PA}}, y_{n,t}^{\text{PA}}, z_{n,t}^{\text{PA}}] \in \mathbb{R}^3. \tag{10}$$

Here, $\mathbf{\Psi}_t$ contains the three-dimensional coordinates of all PA elements, at time t. Since all PA elements are mounted on the same waveguide, the coordinates along y- and z- axes

[4] As the initial state of quantum register is $|0\rangle$, the X gates preserve the encoded channel information while transforming the qubit basis. The X operator, given by $\mathsf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, effectively changes the computational basis states from $\mathsf{X}\,|0\rangle \to |1\rangle$, or vice versa. By doing so, each edge qubit ensures that only active PA–FAS links can control the subsequent unitary operations applied to the vertex register.

[5] Theoretically, a *SWAP* gate allows the qubits to be interchanged without altering the hardware. For example, SWAP$(|u_m, v_n\rangle) \to |v_n, u_m\rangle$. To this purpose, the *swap* gate is used to exchange the positions of qubits according to their corresponding PA waveguide and FAS ports.

remain fixed, while only x-axis coordinates are optimized for PA beamforming. According to the constraints defined in Eqs. (3b) and (3c), each of PA elements is placed within the physical length of its waveguide while maintaining a minimum distance from its neighboring elements. To satisfy these requirements, an auxiliary vector is constructed as $\tilde{x}^{\mathrm{PA}}_{t} = [\tilde{x}^{\mathrm{PA}}_{1,t}, \tilde{x}^{\mathrm{PA}}_{2,t}, \cdots, \tilde{x}^{\mathrm{PA}}_{N,t}]$, where the n-th entry is given by

$$\tilde{x}^{\mathrm{PA}}_{n,t} = \tilde{x}^{\mathrm{PA}}_{1,t} + \sum_{a=2}^{n} \left(\tilde{\Delta}^{\mathrm{PA}}_{a,t}\right), \tag{11}$$

where $\tilde{\Delta}^{\mathrm{PA}}_{a,t} = \max\left[\Delta_{\mathrm{PA,min}}, b_{a,t}\frac{L_{\mathrm{WG}}}{N}\right]$ and $b_{a,t} \in [0,1]$ is obtained from the QGNN output associated with the PA elements. These terms ensure that both the length and spacing constraints are satisfied. By optimizing the PA coordinates, the BS beamforming matrix at time t is obtained as one of the optimization variables in (3), expressed as $\mathbf{Z}_t = \mathrm{diag}\left[\mathbf{z}_{1,t}, \mathbf{z}_{2,t}, \cdots, \mathbf{z}_{N,t}\right]$, where $\mathbf{z}_{n,t}$ denotes the steering vector of the n-th PA element, given by $\mathbf{z}_n = e^{\frac{-j2\pi\epsilon_{\mathrm{ER}}}{\lambda}} \left\|\psi^{\mathrm{PA}}_{0,t} - \psi^{\mathrm{PA}}_{n,t}\right\|$.

2) *With respect to FAS ports.* The port selection vector at time t is represented as $\mathbf{P}_t = [p_{1,t}, p_{2,t}, \cdots, p_{M,t}]^{\mathsf{T}} \in \mathbb{Z}_{+}^{(M-1)\times 1}$, where each $p_{m,t} \in \mathbb{Z}_{+}$ denotes the selected FAS port index associated with the n-th PA element. According to the selection port constraint in (3d), the optimized port index for each PA element choose from a discrete set of available ports, denoted by $\mathcal{P}_{\mathrm{opt}} = \{1, 2, \cdots, M_{\mathrm{port}}\} \in \mathbb{Z}_{+}^{M\times 1}$, where M_{port} indicates the total number of available ports indices. To satisfy this constraint, the QGNN output corresponding to the m-th FAS port, $b_{m,t} \in [0,1]$, is quantized to the nearest entry of $\mathcal{P}_{\mathrm{opt}}$ using a scaling function $\pi(\cdot)$, defined as

$$\begin{aligned} &p_{m,t} = \left\{\pi\left(V_{\mathrm{min}}\right), \pi\left(V_{\mathrm{max}}\right)\right\}, \\ &\pi(v) = \arg\min |v \cdot M_{\mathrm{port}} - p_{\mathrm{opt},u}|, \forall v \in \{V_{\mathrm{min}}, V_{\mathrm{max}}\}. \end{aligned} \tag{12}$$

Here, $V_{\mathrm{min}} = \min_m b_{m,t}$ and $V_{\mathrm{max}} = \max_m b_{m,t}$ define the lower and upper bound of the QGNN outputs at time t, respectively. The function $\pi(v)$ maps each continuous-valued QGNN output variable into a valid discrete of FAS port within the set of $\mathcal{P}_{\mathrm{opt}}$.

IV. Performance Analysis

The following section demonstrates a QGNN-based solution for the joint optimization of flexible antenna systems. We consider a setup comprising of a PASS at the BS and a FAS at the UE, aiming to maximize the user rate. This involves an indoor ceiling-mounted waveguide equipped with $N = 3$ PA elements. All PA elements are positioned at a ceiling height of $z_{\mathrm{PA}} = 3$ m, with a minimum spacing of $\Delta_{\mathrm{PA,min}}$. The UE, located at 0.5 m above the floor level, operates an FAS with M ports. The UE randomly located within a square region of $D_x \times D_y$ m^2, following a bivariate uniform distribution. The system operates at a carrier frequency of 28 GHz. For initialization, the PA elements are placed along the x-axis at $[0, 1, 2]$ m and are iteratively updated during the training process. The QGNN requires as input both the channel information (encoded in the edge circuit qubits, as the state $|e\rangle$) and the flexible antenna's topology (encoded in the vertex circuit qubits, as the state $|v\rangle$), represented as a graph. The QGNN circuit is processed using the IBM Qiskit `AerSimulator` backend, and eventually measured with $N_{\mathrm{shots}} = 1024$ shots. The QGNN model is trained using a gradient descent approach with the parameter-shift rule. The learning rate decays according to $\gamma_{t+1} = \frac{\gamma_t}{\sqrt{t+1}}$.

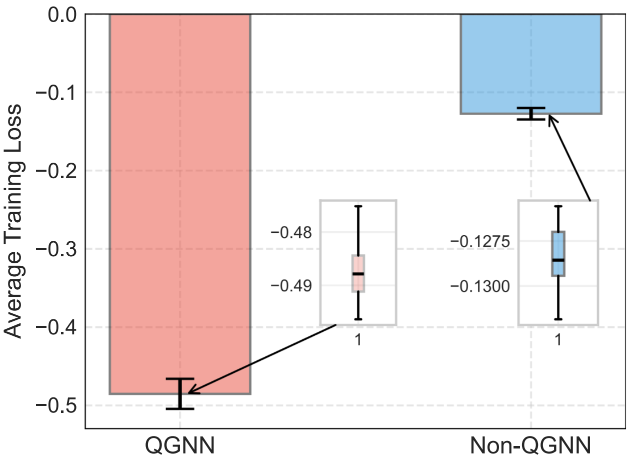
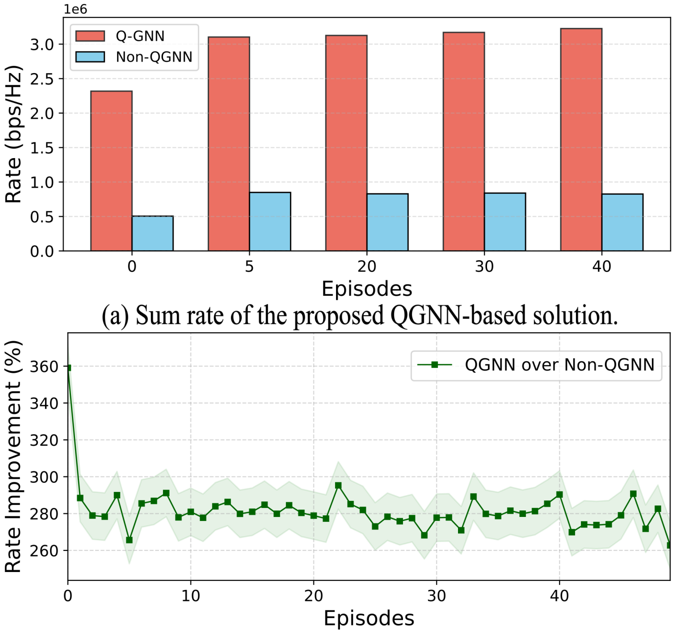


Fig. 2: Training loss performance of the proposed QGNN-based solution.

The effectiveness of the QGNN training process is shown in Fig. 2, as the average training loss values. The "`non-QGNN`" model is implemented as a PQC baseline without any graph embedding mechanism, with the typical circuit shown in [20, Fig. 5].[6] The QGNN achieves a markedly lower loss value, converging around -0.485, whereas the `non-QGNN` model reaches only -0.129 on average. Such a substantial difference of approximately 0.36 in the final loss values further highlights the improved trainability of the QGNN model. These performance improvements arise from the QGNN's ability to leverage the graph-structured representation of the PASS-FAS environment, effectively learning both the channel information and the PA-FAS connections. By embedding these structural correlations through its edge–vertex circuits, the QGNN can better capture the non-linear mapping from the physical configuration. In contrast, the `non-QGNN` relies solely on the channel feature input to learn the parameterized rotations approximating the optimization landscape. Consequently, it lacks the inductive bias provided by the structure of the PASS-FAS system, resulting in slower convergence and higher loss.

Furthermore, the reduction in training loss consistently translates into a steady improvement in system performance, particularly in the achieved rates, defined in (3). Fig. 3 presents the achieved rate performance and the corresponding rate improvement of the proposed QGNN-based solution. As observed in Fig. 3 (a), the QGNN attains rates of approximately 3×10^6 bps/Hz, while the `non-QGNN` baseline achieves only

[6] The number of gates and quantum operations in the baseline model is kept identical to those employed in the QGNN for a fair performance comparison. The baseline model is adjusted to include double-layered entanglement blocks to match the circuit of the QGNN.



(a) Sum rate of the proposed QGNN-based solution.

(b) Rate improvement of QGNN over Non-QGNN.

Fig. 3: The performance of the QGNN-based solution, as indicated by: (a) the achieved user rates and (b) the percentage rate improvement achieved over the Non-QGNN model.

around 0.8×10^6 bps/Hz, and remains stable across the learning iterations. This significant gap underscores the efficacy of the proposed QGNN workflow, as further shown in Fig. 3 (b), where the QGNN consistently maintains an improvement factor of about 3.7× over the baseline model. During the early episodes, slight fluctuations are observed, corresponding to the model's exploration phase as it optimizes the learnable parameters $\mathsf{CU}(\theta)$ parameters in (7). Nevertheless, the shaded area around the line in Fig. 3 (b) shows a narrow variance, indicating strong training stability. These results confirm the feasibility of the QGNN to effectively capture the flexible antenna topology, particularly in the PASS-FAS environment, thereby improving the wireless communication performance.

V. Conclusion and Future Works

This work presents a QGNN-based workflow to jointly optimize flexible antennas at both the BS and the UE. The proposed solution enables the QGNN to maximize user rates by adaptively optimizing PASS and FAS-related parameters. Such optimization relies not solely on channel feature information, but also on a graph-based representation that captures the relationships between PA elements, at the BS, and FAS ports, at the UE. As an initial effort toward graph-based optimization of flexible antenna systems using a QGNN, this work opens the possibilities for quantum graph learning, particularly for joint PASS-FASS optimization. Future extensions may include dynamic graph representations as an adjustable circuit (e.g., [21]), enabling QGNNs to dynamically adapt its vertex structure and circuit configuration in response to changes in flexible antenna topology, thus supporting more robust and intelligent implementations of future flexible antenna systems.

References

[1] Z. Wang, C. Ouyang, X. Mu, Y. Liu, and Z. Ding, "Modeling and beamforming optimization for pinching-antenna systems," *IEEE Trans. Commun.*, Oct. 2025.

[2] Y. Zhong *et al.*, "Physical layer security for pinching-antenna systems via index and directional modulation," *IEEE Commun. Lett.*, Oct. 2025.

[3] J. Xiao, J. Wang, and Y. Liu, "Channel estimation for pinching-antenna systems (PASS)," *IEEE Commun. Lett.*, May 2025.

[4] L. Zhang, Y. Zhao, H. Yang, G. Liang, and J. Hu, "Energy-efficient port selection and beamforming design for integrated data and energy transfer assisted by fluid antennas," *IEEE J. Select. Areas Commun.*, Sep. 2025.

[5] J. O. Martinez, B. G. Guzman, and A. G. Armada, "Correlation and drop velocity in fluid antenna systems: Modeling and performance," *IEEE J. Select. Areas Commun.*, Oct. 2025.

[6] T. Liao, W. Guo, H. He, S. Song, J. Zhang, and K. B. Letaief, "Joint beamforming and antenna position optimization for fluid antenna-assisted MU-MIMO networks," *IEEE J. Select. Areas Commun.*, 2025, [Early Access].

[7] W. Xiang *et al.*, "Fast prediction of quasi-periodic array using dynamical graph convolutional neural networks," *IEEE Antennas, Wireless Propag. Lett.*, vol. 21, no. 5, pp. 893–897, May. 2022.

[8] S. Shrestha, X. Fu, and M. Hong, "Optimal solutions for joint beamforming and antenna selection: From branch and bound to graph neural imitation learning," *IEEE Trans. Sign. Process.*, vol. 71, pp. 831–846, Feb. 2023.

[9] 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.

[10] Z. Qu, X. Liu, and M. Zheng, "Temporal-spatial quantum graph convolutional neural network based on Schrödinger approach for traffic congestion prediction," *IEEE Trans. Intell. Transport. Sys.*, vol. 24, no. 8, pp. 8677–8686, Aug. 2023.

[11] G. Colucci, S. v. d. Linde, and F. Phillipson, "Power network optimization: A quantum approach," *IEEE Access*, vol. 11, pp. 98 926–98 938, Sep. 2023.

[12] D. Roosan *et al.*, "Variational quantum circuits for molecular classification using graph neural network," in *Proc. Int. Conf. Quantum Commun., Network., Computing (QCNC)*, Apr. 2025, pp. 432–436.

[13] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, 2019.

[14] G. Chen, S. He, Z. An, Y. Huang, and L. Yang, "A deep learning method: QoS-aware joint AP clustering and beamforming design for cell-free networks," *IEEE Trans. Commun.*, vol. 71, no. 12, pp. 7023–7038, 2023.

[15] J. Zhang, H. Wang, H. Xie, and J. Qian, "User-side retraining-free learning for high-precision 5G positioning," *IEEE Trans. Commun.*, 2025.

[16] I. M. Braga, R. P. Antonioli, G. Fodor, Y. C. Silva, and W. C. Freitas, "Decentralized joint pilot and data power control based on deep reinforcement learning for the uplink of cell-free systems," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 957–972, 2022.

[17] H. Zhou, W. Xia, G. Zheng, and H. Zhu, "Graph neural network-based continual learning for resource allocation in dynamic wireless environments," *IEEE Trans. Veh. Technol.*, 2025, [Early Access].

[18] Y. Zhang, J. Ding, and X. Li, "Network controllability robustness learning via spatial graph neural networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 4045–4058, Apr. 2024.

[19] X. Tang, K. Zhao, C. Shen, Q. Du, Y. Wang, D. Niyato, and Z. Han, "Deep graph reinforcement learning for UAV-enabled multi-user secure communications," *IEEE Trans. Mobile Comput.*, Apr. 2025.

[20] S. S. Hassan *et al.*, "Quantum machine learning for 6G space-air-ground integrated networks: A comprehensive tutorial and survey," *IEEE Commun. Surv. Tutor.*, Sep. 2025, [Early Access].

[21] M. Zhao, H. Peng, and L. Li, "Multivariate time-series anomaly detection based on dynamic graph neural networks and self-distillation in industrial internet of things," *IEEE Internet Things J.*, vol. 12, no. 9, pp. 12 181–12 192, May 2025.