Quantum Reinforcement Learning: An Overview

Gyu Seon Kim[†], Soohyun Park[§], and Joongheon Kim[†]
[†]Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
[§]Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mail: kingdom0545@korea.ac.kr, soohyun.park@sookmyung.ac.kr, joongheon@korea.ac.kr

***Abstract*—Recent research in quantum computing has gained significant momentum, promising transformative advances across various fields, including artificial intelligence (AI). This paper focuses on quantum reinforcement learning (QRL), which utilizes quantum neural network (QNN) to address the inherent challenges in conventional reinforcement learning (RL) frameworks. For a comprehensive examination of QRL, in this paper, the structure of QNN is explored in-depth, including the functional roles of key quantum circuit gates such as the unitary, rotation, controlled-X, and Hadamard gates. This paper also discusses the primary components of QNN, namely state encoding, parameterized quantum circuits (PQC), and measurement processes. Furthermore, the application of IBM qiskit for the visualization of quantum states and circuits is highlighted, providing practical insights into the deployment of quantum principles in RL. Consequently, this paper encapsulates the potential of QRL to revolutionize AI, positioning it as a pivotal area of future research and development in quantum computing.**

I. Introduction

A. Background and Motivation

REINFORCEMENT learning (RL) using conventional artificial neural networks (NNs) has made many advances in various application areas, but it poses several structural limitations, which pose serious constraints, especially in high-dimensional data processing and complex decision-making problems. Firstly, higher-dimensional state, *i.e.*, input of NNs,/action, *i.e.*, output of NNs, space is a major factor that weighs heavily on the performance and training speed of conventional NNs. In conventional RL, as the dimensions of state space and action space increase, the number of parameters the model needs to train increases exponentially, which significantly increases the computational cost. In addition, data sparsity in higher dimensions causes NNs to increase the number of samples needed to train optimal policies. In conclusion, as the agent's action dimension increases, RL using conventional artificial NNs suffers from a *curse of dimensionality*. Secondly, conventional RL still does not solve the problem related to sample effectiveness. Most RL algorithms require large amounts of interaction data for effective training. In real-world applications, generating such large amounts of data can be costly and time-consuming. This poses a major constraint for real-time applications, especially in dynamic and uncertain environments. Thirdly, conventional RL has challenges in dealing with partially observable-Markov decision process (PO-MDP) environments. In many real-world problems, not all state information can be fully observed. Conventional NNs require additional structures, *e.g.*, recurrent NNs, to incorporate such partial observed information, which increases the complexity and computational needs of the model. Quantum reinforcement learning (QRL) is emerging as a promising solution to solve these three problems of conventional RL. Advances in quantum computing technology are providing innovative possibilities in the field of artificial intelligence (AI), especially RL. **Quantum AI** using *quantum neural network* (QNN) utilize basic principles of quantum mechanics, *i.e., superposition, entanglement*, and *quantum tunneling*, to overcome the structural limitations of conventional NNs. Based on these characteristics, Quantum AI can overcome the above three problems. Firstly, QNN can leverage the superposition state of *quantum bit (qubits)* to represent multiple possible states simultaneously. This superposition allows a single qubit to represent multiple states simultaneously, providing a way to effectively represent high-dimensional data with fewer qubits. This significantly reduces the resources required to solve high-dimensional problems, enabling faster and more efficient training processing. Secondly, QNN can solve problem-related to sample effectiveness through entanglement. Entanglement occurs when two or more qubits are in a mutually *dependent* state, allowing information from one qubit to immediately affect another. This property allows QNN to train much deeper and wider with fewer data, and significantly improves sample efficiency. Thirdly, with superposition, QNN provides the ability to infer the state of the entire system using only partial information. This enables the development of RL algorithms that can make overall decisions based on partial information and significantly improves processing efficiency. Due to the characteristics of QNN that distinguish them from conventional NNs, quantum AI can solve a number of problems arising from RL.

Based on the advantages and features of quantum AI for RL, this paper provides a basic *mathematical explanation* and *visualization* of QNNs. In this paper, the structure and principles of QNN for QRL are covered in depth. In particular, the basic structure of QNN is clarified by explaining various quantum circuit gates, *e.g.*, unitary gate, rotation gate, controlled-X Gate, and Hadamard gate, etc. The structure of quantum artificial neural networks covers three main parts, *i.e.*, *i) state encoding*, *ii) parameterized quantum circuits (PQC)*, and *iii) measurement*. Finally, this paper provides a visualization of quantum states in bloch spheres and a visualization of quantum circuits using `IBM qiskit` for smooth understanding. This helps facilitate understanding for several system designers and engineers entering quantum AI.

B. Contributions

- This paper summarizes the limitations of RL using classical NNs and proposes the QNN-based QRL that can solve these problems. A novel NN architecture is introduced to solve the problems that existing algorithms had.
- This paper describes the mathematical representations of various quantum circuit gates used in QNNs. Furthermore, the basic structure of QNNs and the role of each structure are introduced. These are essential for understanding quantum circuit operations.
- A visualization of quantum states represented by Bloch spheres and quantum circuits is presented. In addition, how quantum states through multiple quantum circuit gates are finally converted into probabilities in the measurement phase is also visually represented.

II. Preliminaries

A. Related Work

Application problems such as image classification are addressed using QNN instead of traditional convolutional neural network (CNN) [1]. Moreover, utilizing QNNs in image classification problems requires as many qubits as pixels, while using QCNN requires only as many qubits as the filter size, which is advantageous in the noisy intermediate-scale quantum (NISQ) era, where the number of qubits is limited [2]. In addition, a study is conducted on federated learning (FL) using QNN rather than conventional NN [3]. Finally, from the perspective of quantum AI, QNN is also utilized for RL and communication systems [4]. RL and algorithms that maximize specific values over time are also widely used in aerial networks [5]–[8]. When designing an aerial access network (AAN) through multiple UAVs, it is important to effectively optimize the communication systems of multiple UAVs [9]. In this environment, QNN can be used to solve problems with QRL. Furthermore, research has been conducted on rocket weight reduction using fewer training parameters in the landing scenario of a reusable rocket [10]. In the AAN environment where the agent's action dimension is very large, quanutm multi-agent reinforcement learning (QMARL) successfully reduced the ground stations (GSs)'s action dimension to a logarithmic scale and learned [11].

B. Convolution Reinforcement Learning

As shown in Fig. 1, RL is a domain within the field of machine learning that focuses on enabling agents to learn optimal behaviors through interactions with an environment. In this paradigm, an agent observes the state of the environment and makes decisions about actions based on these observations. Each action taken by the agent impacts the environment, which in turn provides new states and rewards to the agent. The rewards serve as feedback, guiding the agent in adjusting its actions and improving its policy to maximize future rewards. The primary goal of RL is for the agent to discover the optimal policy that yields the highest cumulative reward through trial and error.

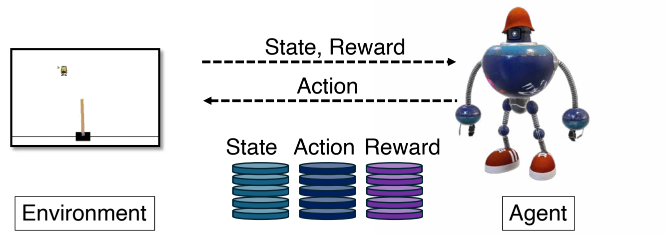
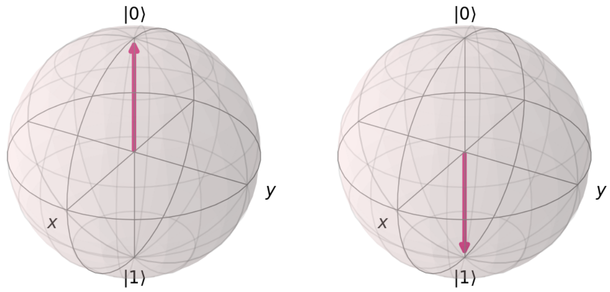


Fig. 1: The concept and components of reinforcement learning

C. Advantages of Quantum Neural Networks

Fast Training Convergence. In QNN, the training process differs from conventional methods by utilizing the parameter shift rule instead of the backpropagation technique used in classical NNs. This rule offers a simpler and more direct approach to policy gradient estimation, which can significantly speed up the training process. This acceleration is particularly beneficial for applications involving cube-satellites (CubeSats)/unmanned aerial vehicles (UAVs) communication systems, where processing power and time are constrained. The ability to efficiently train multiple QNN models without extensive time commitments is crucial, given the limited operational timelines and computational resources available in large-scale, globally distributed network environments. Consequently, the increased training speed facilitated by the parameter shift rule in QMARL is not only advantageous but also essential for optimizing the functionality and performance of globally distributed networks.

Action Dimension Reduction for Efficient Qubit Utilization. QMARL significantly reduces the action dimensions of GSs and the number of qubits required by employing *basis* measurements instead of Pauli-Z measurements. This approach effectively overcomes the limitations associated with the restricted number of qubits available in the NISQ era. In multi-agent reinforcement learning (MARL) environments, the total number of potential actions can increase dramatically, necessitating a proportional rise in the number of qubits. However, in the NISQ era, increasing the number of qubits exacerbates the issue of quantum noise. Moreover, when using Pauli-Z measurements and classical MARL, GSs face the *curse of dimensionality*, making training convergence challenging as the action dimension of GSs expands exponentially. To tackle these challenges, a novel QMARL-based scheduler has been developed. This scheduler leverages basis measurements to significantly decrease the required number of qubits and action dimensions. Unlike Pauli-Z measurements, which assess *individual* qubits against the two computational bases ($|0\rangle$ or $|1\rangle$), basis measurements evaluate the *entire* quantum system across all possible basis states. This approach reduces qubit complexity *logarithmically* in relation to the number of possible actions for GSs. These features are particularly beneficial for environments that cannot utilize a large number



(a) $|0\rangle$ in Bloch sphere. (b) $|1\rangle$ in Bloch sphere.

Fig. 2: Quantum states in Bloch sphere.

of qubits and require high action dimensions for GSs, such as large-scale, globally distributed networks. This methodology not only mitigates the challenges posed by quantum noise but also improves the practical deployment and scalability of quantum-based systems in complex distributed network environments.

III. Quantum Neural Network

A. Characteristics of Quantum Computing

In QNN, unlike in classical NNs, basic training tasks utilize units other than bits, which is a critical aspect. In quantum computing, the fundamental units of memory are quantum bits, or qubits. To highlight the difference between qubits and classical bits, consider that a register of $\mathcal{Q}$ classical bits can exist in $2^{\mathcal{Q}}$ possible states. Each state can be represented by a vector of length $2^{\mathcal{Q}}$, where one element is 1 and all others are 0. In quantum mechanics, a quantum state with $\mathcal{Q}$ qubits is represented as a complex vector with $2^{\mathcal{Q}}$ dimensions. This representation allows a quantum state to exist as a superposition of multiple states simultaneously, unlike classical systems where states are typically represented by a single non-zero element. This phenomenon is referred to as quantum superposition. As shown in Fig. 2, in quantum systems, qubits are represented in two fundamental states using bra-ket notation: $|0\rangle := \begin{bmatrix}1\\0\end{bmatrix}, |1\rangle := \begin{bmatrix}0\\1\end{bmatrix}$, Additionally, a normalized two-dimensional complex vector represents a single qubit state as follows: $|\psi\rangle = \mathfrak{A}\,|0\rangle + \mathfrak{B}\,|1\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$, where $\mathfrak{A}$ and $\mathfrak{B}$ are complex probability amplitudes associated with the states $|0\rangle$ and $|1\rangle$, respectively. These amplitudes must satisfy the condition, *i.e.*, $\|\mathfrak{A}\|^2 + \|\mathfrak{B}\|^2 = 1$. As shown in Fig. 2, QNN computations are performed within the Bloch sphere, which represents the 3D quantum state space, or Hilbert space. When depicted within the Bloch sphere, a quantum state can be geometrically represented as: $|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle$, where ϕ and θ denotes the relative phase and parameter that defines the probabilities of measuring the states, *i.e.*, $|0\rangle$ and $|1\rangle$, respectively. These parameters adhere to the inequalities $0 \le \theta \le \pi$, $0 \le \phi < 2\pi$. For a system with $\mathfrak{Q}$ qubits, the quantum states are represented in the Hilbert space as: $|\psi\rangle = \sum_{l=0}^{2^{\mathfrak{Q}}-1} \omega_l\,|l\rangle$, where ψ, ω_l, and $|l\rangle$ denotes the quantum state, probability amplitude, and l-th basis state of the $\mathfrak{Q}$-qubit system. The probability amplitudes shall satisfy the following conditions, *i.e.*, $\sum_{l=0}^{2^{\mathfrak{Q}}-1} |\omega_l|^2 = 1$. Furthermore, two quantum states, such as $|\psi\rangle$ and $|\phi\rangle$, can be jointly represented using the tensor product for multi-qubit states as $|\psi\rangle \otimes |\phi\rangle$. However, due to the phenomenon of quantum entanglement, a multi-qubit state $|\psi\rangle$ cannot always be decomposed into tensor products of simpler systems.

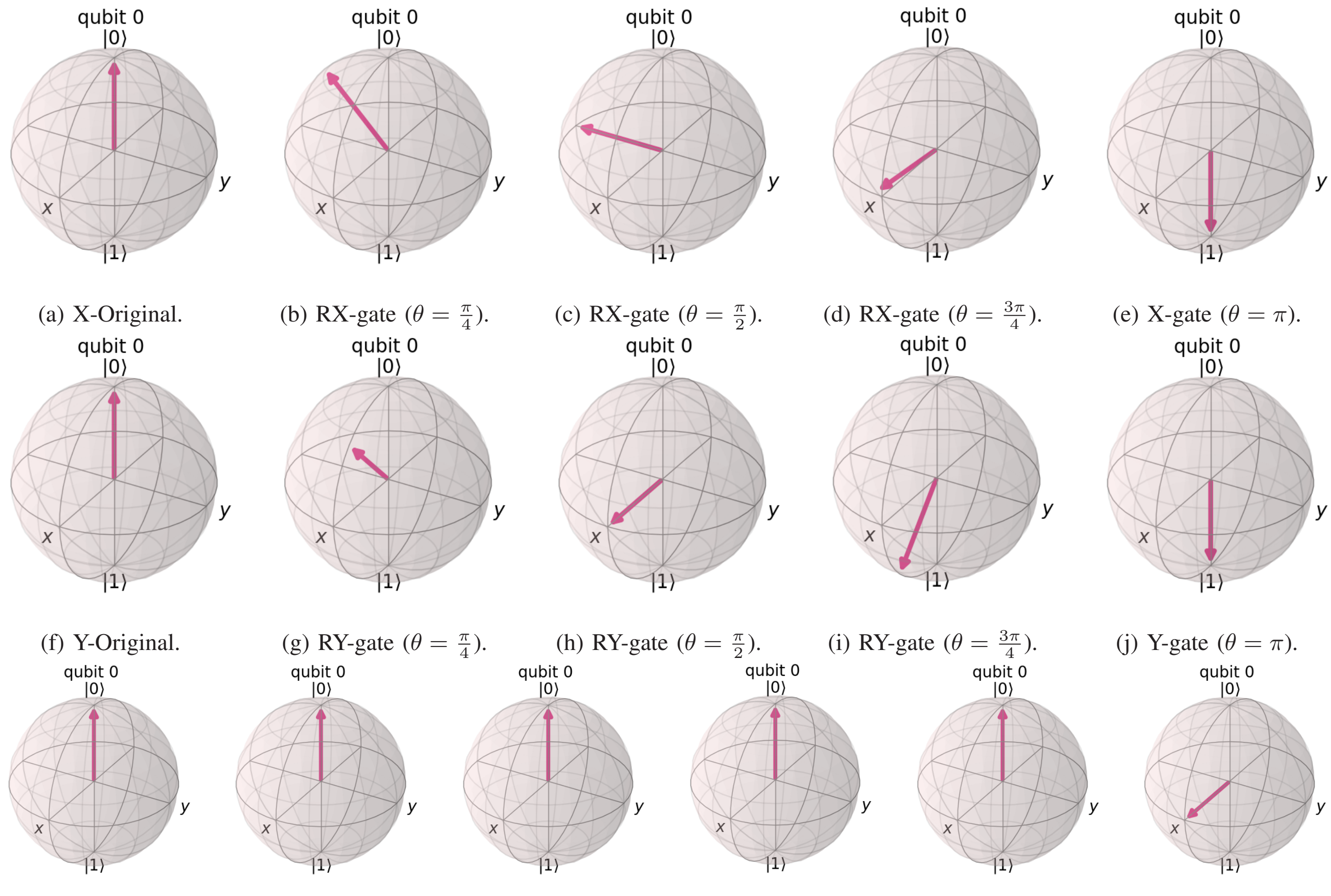
B. Quantum Circuit Gates

Quantum circuit gates are used as basic computational units in quantum computing and perform similar functions to logic gates of classical computers. These gates manipulate the state of qubits, processing information and perform calculations. Quantum circuit gates operate by exploiting properties of quantum mechanics, such as quantum superposition and entanglement, providing the possibility for quantum computers to more efficiently solve complex problems.

1) Unitary Gate: A unitary matrix is an important mathematical tool used to transform the state of qubits in quantum computing. All quantum gates apply a unitary transformation to change the state of the qubit. A unitary gate has the property of rotating the state to another state while preserving the overall probability of the quantum state. Unitary matrices define the way quantum circuit gates behave in quantum states, and these matrices have several important mathematical properties which is expressed as, $U^T U = U U^T = I$, where U^T and I stand for conjugate transpose of matrix U and identity matrix, respectively. In the realm of quantum computing, the significance of unitary matrices—specifically U_1, U_2, and U_3 gates—cannot be overstated. These gates form the backbone of operations on qubits within IBM's quantum computing systems, providing the necessary transformations for executing complex quantum algorithms. The U_1 gate is a single-parameter quantum gate that applies a phase shift around the Z-axis of the Bloch sphere. It is defined mathematically by the following matrix: $U_1(\lambda) = \begin{bmatrix}1 & 0\\0 & e^{i\lambda}\end{bmatrix}$. This matrix operation does not alter the probability amplitudes of the qubit states but instead introduces a phase difference of λ between the basis states. The U_1 gate is crucial for phase kickback in algorithms like the quantum phase estimation and is often utilized for its low computational overhead in circuit implementations. The U_2 gate extends the functionality of the U_1 gate by incorporating an additional parameter, providing a richer set of transformations. It is expressed as,

$$U_2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{bmatrix}1 & -e^{-i\lambda}\\ e^{i\phi} & e^{i(\phi+\lambda)}\end{bmatrix}, \tag{1}$$

where λ and ϕ are constants that tune the phase of the state $|1\rangle$ in the angular superposition and those that regulate the phase change introduced when mapped to the superposition in the basis state $|0\rangle$. The U_2 gate effects a transformation that is fundamentally a generalization of the Hadamard gate, combining phase shifts and superposition. It enables more



(a) X-Original. (b) RX-gate ($\theta = \frac{\pi}{4}$). (c) RX-gate ($\theta = \frac{\pi}{2}$). (d) RX-gate ($\theta = \frac{3\pi}{4}$). (e) X-gate ($\theta = \pi$).

(f) Y-Original. (g) RY-gate ($\theta = \frac{\pi}{4}$). (h) RY-gate ($\theta = \frac{\pi}{2}$). (i) RY-gate ($\theta = \frac{3\pi}{4}$). (j) Y-gate ($\theta = \pi$).

(k) Z-Original. (l) RZ-gate ($\theta = \frac{\pi}{4}$). (m) RZ-gate ($\theta = \frac{\pi}{2}$). (n) RZ-gate ($\theta = \frac{3\pi}{4}$). (o) Z-gate ($\theta = \pi$). (p) Hadamard-gate

Fig. 3: Quantum states with various gates on Bloch sphere.

complex state preparations and transformations, making it suitable for diverse quantum operations and entanglement creation. The U_3 gate is the most general form of a single-qubit unitary transformation, incorporating three parameters. It is capable of achieving any rotation on the Bloch sphere and is expressed by the matrix:

$$U_3(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\theta/2) & -e^{-i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{bmatrix}, \tag{2}$$

where θ is a constant that rotates the qubit around the x-axis of the Bloch sphere. This gate combines arbitrary rotations around the Bloch sphere, encompassing and generalizing the effects of both U_1 and U_2 gates. The U_3 gate's ability to perform any conceivable single-qubit gate operation makes it a cornerstone of quantum circuit design. These unitary gates U_1, U_2, and U_3 are essential for designing complex quantum circuits and algorithms. Their implementation on IBM's quantum platforms provides quantum programmers with versatile tools for manipulating qubit states, facilitating quantum computation's unique advantages over classical computing. Each gate, by manipulating phases and amplitudes, allows for the precise control necessary in quantum computation, ensuring that quantum circuits can perform a wide variety of algorithmic tasks effectively.

2) Rotation Gate: In quantum circuits, the rotation gates $\mathcal{R}_x$, $\mathcal{R}_y$, and $\mathcal{R}_z$ play critical roles by enabling precise rotations of qubits around the X, Y, and Z axes of the Bloch sphere, respectively. These gates are fundamental in manipulating qubit states for desired quantum operations and are essential for the implementation of complex quantum algorithms. As shown in Figs. 3(a)-(e), the $\mathcal{R}_x$ gate performs a rotation around the X-axis of the Bloch sphere. It is parametrized by θ, which represents the angle of rotation. The unitary matrix representation of the $\mathcal{R}_X$ gate is given by:

$$\mathcal{R}_X(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \tag{3}$$

This matrix effectively mixes the amplitudes of the computational basis states $|0\rangle$ and $|1\rangle$ with a phase shift introduced by the imaginary unit i, indicating the quantum nature of the operation. The terms $\cos(\theta/2)$ and $\sin(\theta/2)$ determine how much the state vector is rotated, reflecting the probabilistic outcomes upon measurement. Similarly, as shown in Figs. 3(f)-(j), the $\mathcal{R}_Y$ gate rotates a qubit around the Y-axis. Like the $\mathcal{R}_X$ gate, it uses the angle θ to define the extent of the rotation.

The unitary matrix for the $\mathcal{R}_Y$ gate is:

$$\mathcal{R}_Y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \tag{4}$$

Unlike the $\mathcal{R}_X$ gate, the $\mathcal{R}_Y$ gate involves real coefficients, which simplifies its physical interpretation: it rotates the state vector in the real plane of the Bloch sphere without introducing complex phase factors, directly altering the probabilities associated with the computational basis states. As shown in Figs. 3(k)-(o), the $\mathcal{R}_Z$ gate focuses on altering the phase of the qubit states around the Z-axis. This gate does not affect the amplitudes of the quantum states but imparts a relative phase between them. The unitary matrix for the $\mathcal{R}_Z$ gate is expressed as,

$$\mathcal{R}_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{5}$$

This matrix applies a phase shift of $-\theta/2$ to the $|0\rangle$ state and $\theta/2$ to the $|1\rangle$ state, effectively modifying how these states interfere with each other without changing their individual probabilities. These rotation gates are crucial in quantum computing as they enable the precise control necessary for tasks such as quantum state preparation, quantum error correction, and the implementation of quantum logic gates. By adjusting the parameters of these gates, one can tailor the quantum circuit to perform a wide range of operations, from simple quantum state manipulations to the execution of complex algorithms that leverage quantum mechanical principles like superposition and entanglement.

3) Controlled-X Gate: The Controlled-X Gate, often referred to as the CNOT (Controlled NOT) gate, is a pivotal two-qubit gate within quantum computing that performs a conditional operation based on the state of the control qubit. The CNOT gate is essential for creating entanglement between qubits, which is fundamental for quantum teleportation, superdense coding, and various quantum algorithms, including quantum error correction and quantum logic circuits. The Controlled-X Gate operates on a pair of qubits: one acts as the control and the other as the target. The operation of the CNOT gate depends on the state of the control qubit: If the control qubit is in the state $|0\rangle$, the gate leaves the state of the target qubit unchanged. If the control qubit is in the state $|1\rangle$, the gate applies the X operation (equivalent to a classical NOT operation) to the target qubit, flipping its state from $|0\rangle$ to $|1\rangle$ or from $|1\rangle$ to $|0\rangle$. The unitary matrix representation of the CNOT gate is expressed as,

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

The CNOT gate's ability to conditionally alter the state of qubits based on another qubit's state makes it a critical component for implementing logical operations in quantum circuits. Its role in generating entanglement is also crucial, as entangled states are key resources in nearly all quantum communication and computation protocols. Furthermore, the simplicity of its operation, combined with its powerful implications in state manipulation and entanglement, underscores its utility in the broader framework of quantum information processing.

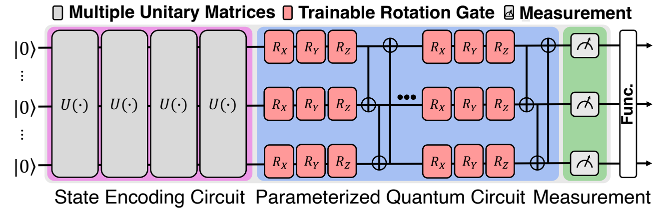


Fig. 4: Three components of the quantum neural network

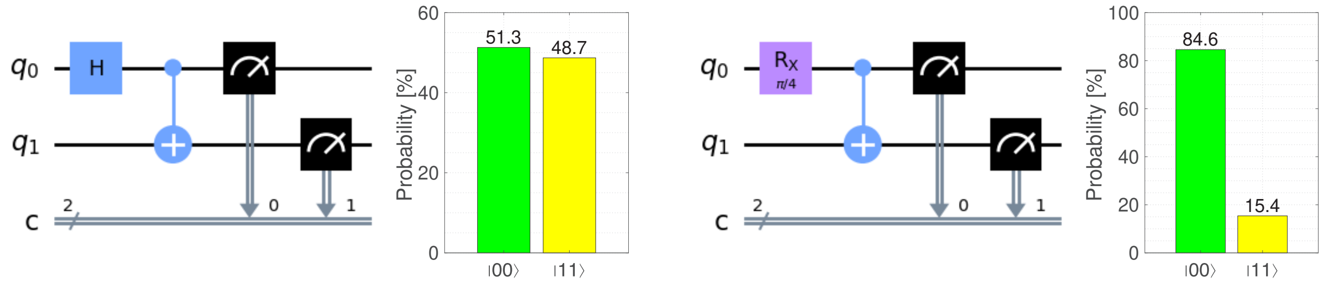
4) Hadamard Gate: The Hadamard Gate acts on a single qubit and has the effect of putting the qubit into a superposition of its basis states. As shown in Fig. 3(p), it performs a rotation of π about the axis that lies halfway between the X-axis and Z-axis on the Bloch sphere. This transformation makes the Hadamard Gate critical for tasks that require the creation of superposition as a preliminary step in quantum computations. The matrix representation of the Hadamard Gate is given by: $H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ The action of the Hadamard Gate on the standard computational basis states $|0\rangle$ and $|1\rangle$ can be described as follows: When applied to $|0\rangle$, the Hadamard Gate transforms it into a state of equal superposition of $|0\rangle$ and $|1\rangle$, often written as, $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) = |+\rangle$. When applied to $|1\rangle$, it results in a state of equal superposition with a relative phase of π (or a negative sign) between $|0\rangle$ and $|1\rangle$, *i.e.*, $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = |-\rangle$.

IV. Quantum Reinforcement Learning

As shown in Fig. 4, QNN is divided into three stages, each of which is *i) state encoding*, *ii) PQC*, and *iii) measurement*.

State Encoding. In QRL, state encoding is the process by which the states of the environment, often represented as vectors in classical reinforcement learning, are encoded into quantum states that can be processed by a quantum computer. Effective state encoding is vital for harnessing the quantum advantage, as it directly impacts the ability of the quantum system to represent and process complex environments. Common methods of quantum state encoding include: *i) Basis encoding :* Directly maps discrete environmental states onto the computational basis of qubits. Each qubit represents a binary state, thus allowing an exponential reduction in the number of qubits required compared to the classical representation. *ii) Amplitude encoding :* Utilizes the amplitudes of a quantum state to encode probabilistic information about possible environmental states, offering a compact representation that can encapsulate a rich set of data with fewer qubits.

Parameterized Quantum Circuits. PQC serve as the functional backbone of QANNs in QRL, analogous to the network of neurons and synapses in classical neural networks. PQCs consist of quantum gates whose parameters are adjustable and



(a) Quantum circuit (Bell state). (b) Probability. (c) Quantum circuit ($\mathcal{R}_X$ + CNOT gate). (d) Probability.

Fig. 5: Quantum circuit and probability in the measurement stage.

optimized through the learning process. In QRL, these circuits are designed to evolve an encoded quantum state into a new state that encodes the policy or value functions associated with RL tasks. Parameters in PQCs are akin to the weights in classical neural networks and are optimized based on the feedback from the environment to improve policy decisions. Furthermore, include rotation gates and entangling gates that manipulate the quantum state. The choice and configuration of these gates influence the training capabilities and efficiency of the QRL model.

Measurement. Measurement in QRL translates the quantum states, manipulated and evolved through PQCs, back into classical information that dictates the actions to be taken in the environment. This component is crucial for realizing the outputs of quantum computations in a form that can be practically applied to make decisions. Upon measurement, the quantum state collapses to one of the basis states, with the probabilities influenced by the quantum computations. The result of this measurement is interpreted as an action or a set of actions within the reinforcement learning framework. The choice of measurement basis can affect the policy performance and needs to be aligned with the QRL strategy.

V. Quantum Circuits and Measurements

Fig. 5(a) shows a quantum circuit for the Bell state. In quantum artificial neural networks, Bell states are one of the prime examples of quantum entanglement, indicating a strong quantum correlation between two qubits. Because the first qubit passes through the Hadamard gate and becomes superimposed, the probabilities of $|00\rangle$ and $|11\rangle$ come out close to half-and-half (Fig. 5(b)). On the other hand, as shown in Fig. 5(c), if the first qubit passes through the $\mathcal{R}_X$ gate with $\pi/4$, the quantum state is closer to $|0\rangle$ than $|1\rangle$ before measurement, thus $|00\rangle$ is greater than the probability of $|11\rangle$.

VI. Concluding Remarks

QRL leverages quantum principles like superposition and entanglement to overcome limitations faced by classical reinforcement learning systems, particularly in high-dimensional data processing and partially observable environments. In this paper, the various quantum circuit gates used in such QRL, the structure of QNN, and the visualization of quantum states are addressed.

Acknowledgements

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government [MSIT (Ministry of Science and ICT (Information and Communications Technology))] (RS-2024-00439803, SW Star Lab) for Quantum AI Empowered Second-Life Platform Technology. J. Kim is the corresponding author of this paper.

References

[1] H. Baek, W. J. Yun, S. Park, and J. Kim, "Stereoscopic scalable quantum convolutional neural networks," *Neural Networks*, vol. 165, pp. 860–867, August 2023.

[2] H. Baek, S. Park, and J. Kim, "Logarithmic dimension reduction for quantum neural networks," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, October 2023, pp. 3738–3742.

[3] M. Chehimi and W. Saad, "Quantum federated learning with quantum data," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, May 2022, pp. 8617–8621.

[4] S. Park, G. S. Kim, H. Zhu, and J. Kim, "Quantum multi-agent reinforcement learning is all you need: Coordinated global access in integrated tn/ntn cube-satellite networks," *IEEE Communications Magazine*, October 2024.

[5] G. S. Kim, H. Lee, S. Park, and J. Kim, "Joint frame rate adaptation and object recognition model selection for stabilized unmanned aerial vehicle surveillance," *ETRI Journal*, vol. 45, no. 5, pp. 811–821, October 2023.

[6] S. Park, G. S. Kim, S. Jung, and J. Kim, "Markov decision policies for distributed angular routing in leo mobile satellite constellation networks," *IEEE Internet of Things Journal*, 2024 (Early Access).

[7] C. Park, G. S. Kim, S. Park, S. Jung, and J. Kim, "Multi-agent reinforcement learning for cooperative air transportation services in city-wide autonomous urban air mobility," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 4016–4030, August 2023.

[8] C. Park, S. Park, G. S. Kim, S. Jung, J.-H. Kim, and J. Kim, "Multi-agent deep reinforcement learning for efficient passenger delivery in urban air mobility," in *ICC 2023 - IEEE International Conference on Communications*, Rome, Italy, May 2023, pp. 5689–5694.

[9] G. S. Kim, H. Lee, C. Park, S. Jung, J.-H. Kim, and J. Kim, "Hybrid mac for military uav networks," in *IEEE International Conference on Information Networking (ICOIN)*, Bangkok, Thailand, January 2023, pp. 656–659.

[10] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 12 252–12 257, August 2024.

[11] G. S. Kim, Y. Cho, J. Chung, S. Park, S. Jung, Z. Han, and J. Kim, "Quantum multi-agent reinforcement learning for cooperative mobile access in space-air-ground integrated networks," *arXiv preprint arXiv:2406.16994*, 2024.