

Towards Explainable Quantum AI: Informing the Encoder Selection of Quantum Neural Networks via Visualization

Shaolun Ruan , Feng Liang , Rohan Ramakrishna , Chao Ren ,
Rudai Yan , Qiang Guan , Jiannan Li and Yong Wang

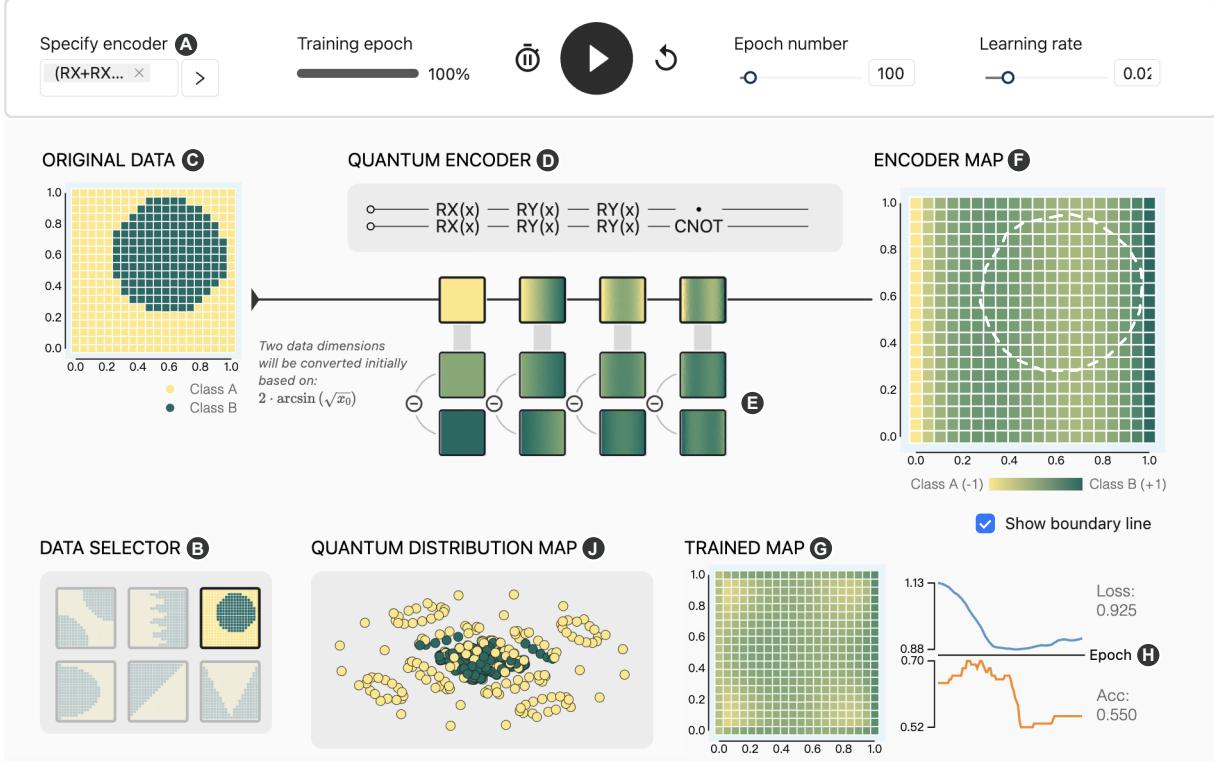


Fig. 1: The interface of *XQAI-Eyes* supports the reasoning and selection of the encoder's quality in QNNs. The Original Data View (C) visualizes the input dataset. The Encoded Data Evolution View (E) illustrates the quantum circuit used for encoding together with the illustration of the encoding process in each step. The Encoder Map View (F) displays the encoded data as a heatmap, facilitating a direct comparison with the original data. The Quantum Distribution Map (J) provides an intuitive representation to show how well the encoder distinguishes the data points with different classes. The Trained Map View (G) visualizes the final learned patterns after training, while the Performance Analysis View (H) uses line charts to depict the training loss and accuracy.

Abstract—Quantum Neural Networks (QNNs) represent a promising fusion of quantum computing and neural network architectures, offering speed-ups and efficient processing of high-dimensional, entangled data. A crucial component of QNNs is the encoder, which maps classical input data into quantum states. However, choosing suitable encoders remains a significant challenge, largely due to the lack of systematic guidance and the trial-and-error nature of current approaches. This process is further impeded by two key challenges: (1) the difficulty in evaluating encoded quantum states prior to training, and (2) the lack of intuitive methods for analyzing an encoder's ability to effectively distinguish data features. To address these issues, we introduce a novel visualization tool, *XQAI-Eyes*, which enables QNN developers to compare classical data features with their corresponding encoded quantum states and to examine the mixed quantum states across different classes. By bridging classical and quantum perspectives, *XQAI-Eyes* facilitates a deeper understanding of how encoders influence QNN performance. Evaluations across diverse datasets and encoder designs demonstrate *XQAI-Eyes*'s potential to support the exploration of the relationship between encoder design and QNN effectiveness, offering a holistic and transparent approach to optimizing quantum encoders. Moreover, domain experts used *XQAI-Eyes* to derive two key practices for quantum encoder selection, grounded in the principles of pattern preservation and feature mapping.

Index Terms—Data visualization, quantum neural network, explainable artificial intelligence (XAI), quantum data encoder.

1 INTRODUCTION

- Shaolun Ruan and Jiannan Li are with Singapore Management University. E-mail: sruan.2021@phdcs.smu.edu.sg, jiannanli@smu.edu.sg
- Feng Liang, Rohan Ramakrishna, Rudai Yan and Yong Wang are with Nanyang Technological University. E-mail:

[feng011|roha0012|rudai011@e.ntu.edu.sg](mailto:{feng011|roha0012|rudai011}@e.ntu.edu.sg), yong-wang@ntu.edu.sg.

Yong Wang is the corresponding author.

• Chao Ren is with KTH Royal Institute of Technology. E-mail: ren0003@e.ntu.edu.sg

• Qiang Guan is with Kent State University. E-mail: qguan@kent.edu.

Quantum computing is an emerging field that harnesses the principles of quantum mechanics to achieve advantages beyond the capabilities of classical computation. Building on this foundation, quantum neural networks (QNNs) integrate quantum operations with neural network architectures to tackle complex optimization problems [9, 19, 25]. Compared to their classical counterparts, QNNs demonstrate the potential for exponential speed-ups and are particularly well-suited for processing high-dimensional and entangled data [1, 10, 40, 51]. Furthermore, by exploiting quantum parallelism, QNNs facilitate more efficient learning and optimization processes [3, 53]. Meanwhile, visualization, which has proven to be a suitable scientific educational tool, has seen an increasing proliferation in research [5, 34, 35, 46] and applications [12, 29, 30] of quantum computing in recent years, significantly enhancing the transparency of black-box quantum algorithms.

Typically, a QNN consists of three basic components: the encoder, the ansatz, and the measurement. As the first component, the encoder plays a crucial role in transforming classical input data into quantum states, enabling the encoded data to be recognized and trained by the subsequent ansatz layer. Due to the fact that different encoders can produce completely different results, the selection and design of the encoder significantly affect the final performance of the QNN [22, 26]. More specifically, the model could achieve optimal performance only if the original data features can be effectively transformed into the corresponding quantum states with the preserved features.

Despite the importance of the QNN encoder, it is still tricky for QNN developers to select the existing encoder templates for their tasks, because of the lack of knowledge to understand how a certain quantum encoder will affect a QNN's performance. Consequently, current practice involves repeatedly implementing different encoders until decent performance is achieved. However, this trial-and-error process has two major problems. Firstly, it is highly time-consuming and labor-intensive to find an appropriate encoder. Secondly, developers may miss the optimal encoder if they deem some commonly-used encoders satisfactory and settle for them too early. Therefore, it is particularly necessary for QNN developers to build a cognitive model and enhance their understanding of encoder selection, enabling them to find the most appropriate encoders based on their informed knowledge. However, there are two significant challenges to address this problem. Firstly, there is rarely work that studied how to **visualize the encoded classical data points** and further inform the developers the correlation between the original data and post-encoded data. Secondly, from the angle of explainable AI, there are no methods available to **intuitively analyze whether the encoder effectively distinguishes the features of different classes**. This lack of transparency in the encoder's performance introduces a significant hurdle for developers.

To address the aforementioned challenges, we first proposed a novel quantum circuit measurement method, namely *Encoder Expectation Measurement*, which extracts the expectation value of each original data point and transforms the abstract encoded states into an observable “classical” variable. This method allows users to directly gain knowledge of how different features of data points are encoded within different QNNs. Additionally, to visually highlight how well the encoder distinguishes between different classes of data, we developed a visualization approach called *State Comparison Map* to intuitively provide a clear overview of the mixed levels of different encoded data, enhancing the understanding from a “quantum” perspective by comparing it with the above “classical” method. With the integration of these two methods, we developed an Explainable Quantum Artificial Intelligence (XQAI) visualization system, namely *XQAI-Eyes*, allowing users to freely explore and understand how different encoders capture the data features and, furthermore, affect the model's performance. Users can experiment with various cases (i.e., 6 datasets \times 10 encoders = 60 cases), which offers users a holistic view of how to design an optimal encoder regarding the features of training data. We finally conducted

case studies and post-study interviews with quantum computing experts to extensively evaluate *XQAI-Eyes*. To the best of our knowledge, *XQAI-Eyes* is the first work to visually explain the performance of quantum encoders, bridging the gap between the encoder reasoning and the QNN performance.

The key contributions of this work are summarized as follows:

- We first formulated the design requirements for informing QNN developers of the guidance of encoder selection, based on the semi-structured interviews with QNN developers.
- We then proposed two methods, i.e., Encoder Expectation Measurement and the Quantum Distribution View, from the perspectives of “classical” and “quantum”. These two methods facilitate 1) the comparison between data features and encoded quantum states, and 2) the analysis of the mixed quantum states of different classes, respectively.
- Based on the distilled design requirements and the two methods described above, we developed a visualization tool, *XQAI-Eyes* to enhance QNN developers' understanding and facilitate intuitive insights into how encoders impact QNN model performance.

We made *XQAI-Eyes*^{1,2} public to benefit QNN developers.

2 RELATED WORK

Our research relates to visualization for quantum computing, visualization for deep neural network explainability and performance evaluation of quantum data encoders.

2.1 Visualization for Quantum Computing

Visualization techniques have been used in quantum computing for quantum circuit visualization and quantum state visualization.

Quantum circuit visualization. Recent work focused on utilizing generally-applicable visualization approaches to enhance the explainability of quantum circuits. For example, QuantumEyes [34] introduced an interactive visualization system that enhances the interpretability of quantum circuits by mapping network activations and weight distributions onto intuitive graphical representations. Williams et al. [48] developed a sophisticated quantum simulator framework designed to compute qubit state probabilities; Furthermore, Karafyllidis et al. [15] proposed a matrix-like visualization approach to solve the non-transparency problem for quantum-version Fourier Transform algorithm. Some co-design designs [16] were also proposed to yield the design principles for designing the quantum computing interfaces.

Quantum state visualization. The visualization of quantum states has been explored using various representations of the state vector. The Bloch Sphere representation [6] maps any pure single-qubit state to a unique point on a three-dimensional unit sphere. This fundamental approach has inspired further adaptations; for example, Makela et al. [21] extended the Bloch sphere concept to handle two-qubit states by embedding their joint state space in a higher-dimensional geometric framework. Wille et al. [47] introduced a tree-like design that hierarchically decomposes a state vector into constituent components. In a similar spirit, Zulehner et al. [54] employed decision diagrams to succinctly represent the matrix form of quantum state amplitudes. In addition, the VENUS system proposed by Ruan et al. [36] provides a geometrical representation to explain how these geometric features relate to the underlying quantum behavior.

Despite their effectiveness, none of them could be directly applied to explain the inner workings of quantum neural networks. The latest work, namely Violet [35], was the most relevant study to our work, which opened the black box of QNNs by visualizing the quantum state evolution in each component. However, it did not focus on the encoder part, which was the most critical part of QNNs. Moreover, we move beyond the transparency enhancement of QNN by unveiling the way how different encoders captured the data features and affected the model's performance accordingly.

¹<https://q-encoder-vis.vercel.app/>

²<https://github.com/shaolunruan/QEncoderVis>

2.2 Visualization for Deep Neural Network Explainability

Due to the increasing complexity of data and deep neural network models, various visualization approaches were proposed to understand [7, 14, 23, 39, 44], diagnose [13, 20, 41–43, 50], and improve [37, 49, 50] these models. Specifically, these approaches were categorized into feature-oriented and evolution-oriented visualizations [18]. Feature-oriented visualizations helped understand the features learned by the model and their impact on predictions. For example, Grad-CAM [39] highlighted important areas in input images for convolutional neural network (CNN) predictions. VBridge [7] linked influential features with raw data. M^2 Lens [45] visualized multimodal features, including verbal, acoustic, and visual elements. Neuron-level interpretations were also explored by techniques like NeuroCartography [27]. Evolution-oriented visualizations focused on the training process of the network. Re-VACNN [8] visualized layer activations in real-time during CNN training. CNNComparator [52] used matrix visualization to compare neuron features between model snapshots.

While the above work significantly addressed the non-transparency issues of deep neural networks, they cannot be directly used for explaining quantum neural network due to the quantum-specific properties. For example, the classical systems cannot reveal how the classical data has been transformed during the quantum encoding procedure because the quantum states are actually complex numbers and cannot be visualized.

2.3 Performance Evaluation of Quantum Data Encoder

Three prior works investigated the empirical studies for a better characterization of quantum data encoders. For instance, Rath et al. [31] conducted a broad empirical analysis of quantum encoding methods across classical machine learning models, highlighting the encoding technologies that improved model accuracy and feature representation efficiency. Monnet et al. [24] also used an empirical study to evaluate four different encoding strategies, including the angle encoding, in QNN design. Abuga [2] focused on different encoding techniques and tested their performance on the MNIST dataset. Meanwhile, some works were also conducted to evaluate other aspects of quantum data encoding techniques. Perez et al. [28] introduced the concept of data re-uploading, where encoded data is repeatedly fed through quantum layers to address the challenge that quantum states cannot be copied or duplicated (a principle known as the no-cloning theorem). Hubregtsen et al. [11] proposed quantum metrics like expressibility and entanglement capacity to evaluate how different encoders influenced variational circuit outputs. Lin et al. [17] studied encoding strategies tailored for noisy intermediate-scale quantum (NISQ) hardware and balanced circuit complexity with encoding fidelity for practical implementations. Romero et al. [33] examined how encoding using Pauli matrices enhanced the separability of quantum states, leading to better feature discrimination in supervised learning tasks.

The prior works mainly focused on leveraging different empirical study approaches (e.g., metric evaluation and comparative study) to investigate how to evaluate the quantum data encoders' performance. Despite the popularity, the conventional methods, e.g., the expressibility or end-to-end testing, cannot directly build a mental intuition and further support a better understanding of the encoder by utilizing the human-in-the-loop methods. Our study focuses on this topic via a design study by collaborating with domain users and experts.

3 BACKGROUND

In this section, we introduce some fundamental concepts (i.e., quantum data, QNNs, and quantum data encoders) to help readers better understand the basics of quantum computing and QNNs.

Quantum data. Quantum data refers to information stored in quantum states, which are the fundamental units of quantum information. Unlike classical data represented by bits that can be either 0 or 1, quantum data is represented by qubits that can exist in a superposition of both 0 and 1 states simultaneously. Mathematically, a quantum state $|\psi\rangle$ of a single qubit can be expressed as a linear combination of the basis states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

where α and β are complex numbers such that $|\alpha|^2 + |\beta|^2 = 1$. Another fundamental property of quantum data is entanglement, which occurs when the quantum states of two or more qubits become interconnected such that the state of one qubit cannot be described independently of the states of the other qubits. Entanglement allows for the creation of correlations between qubits that are not possible in classical systems.

In quantum mechanics, the density matrix is a fundamental mathematical framework used to describe the state of a quantum system. For example, consider a two-qubit system, i.e., a Bell state, which is a well-known pure state in the category of quantum states, to illustrate the concept. The density matrix corresponding to this state is given by:

$$\rho = |\Phi^+\rangle\langle\Phi^+| = \frac{1}{2} \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}. \quad (2)$$

This density matrix fully captures the entangled nature of the Bell state, as it encodes the correlations between the two qubits. If the two qubits were in a mixed state, such as a probabilistic combination of $|00\rangle$ and $|11\rangle$, the density matrix would instead represent the weighted sum of these states.

Quantum neural networks. QNNs are neural networks that leverage the principles of quantum computing to perform computations. QNNs consist of three basic components: the encoder, the ansatz, and the measurement. The encoder transforms classical input data into quantum states. The ansatz, also known as the quantum-version model layer, is a parameterized quantum circuit that applies a series of quantum gates to the encoded quantum states. The general form of an ansatz can be represented as: $|\psi(\theta)\rangle = U(\theta)|\phi\rangle$, where $U(\theta)$ is a unitary operator parameterized by a set of parameters θ , and $|\phi\rangle$ is the encoded quantum state. The parameters θ are typically trained parameters that are optimized during the training process to minimize a cost function. Measurement is the process of extracting classical information from quantum states after they have been processed by the ansatz. The measurement outcome is a probability distribution over the possible states. Mathematically, the probability of measuring a state $|x\rangle$ from the quantum state $|\psi(\theta)\rangle$ is given by:

$$P(x) = |\langle x|\psi(\theta)\rangle|^2, \quad (3)$$

where $\langle x|\psi(\theta)\rangle$ is the inner product of the measured state $|x\rangle$ and the quantum state $|\psi(\theta)\rangle$.

Quantum data encoder. The quantum data encoder, also known as quantum state preparation, is a crucial component in QNNs. It transforms classical input data into quantum states that can be processed by the quantum circuit. Essentially, the encoder is formed by a series of quantum gates that map classical data into the quantum state space. Notably, one of the most popular methods today is angle encoding, where the values of classical data determine the rotation angles of quantum gates, such as Pauli rotation gates. This transformation converts a data value into a quantum state representation.

Intuitively, the role of the quantum data encoder can be compared to the input layer of a classical neural network where input features are scaled and passed into neurons for further computation. Similarly, in a QNN, an encoder maps input features into the quantum state space through quantum gate operations (e.g., rotations), allowing the quantum circuit to process information in superposition, giving the network the ability to evaluate many feature combinations simultaneously.

Once the data is encoded into a quantum state, it cannot be directly inspected, because any measurement collapses the state and destroys its quantum properties such as superposition and entanglement. The encoded data is then passed to the ansatz for further processing, and only after the optimization process is completed can the final quantum state be measured—ensuring that quantum properties are preserved throughout the computation.

4 FORMATIVE STUDY

To inform the design of our visualization system *XQAI-Eyes*, we worked closely with domain experts and collected their actual needs in routine tasks. In this section, we first introduce the formative study and report on the design requirements distilled.

4.1 Formative Methods

Following the core stages of the design study as formulated by Sedlmair et al. [38], we designed the formative study as follows:

Participants. The study involved nine domain experts (**P1-P9**) (3 females + 6 males, $age_{avg} = 33.6$, $age_{sd} = 5.8$) from academic institutions across the United States, Singapore, and Sweden. Specifically, **P1-3** are professors, **P4-6** Ph.D. students, and **P7-9** postdoctoral researchers. All participants have research experience in quantum computing and quantum machine learning, with an average of 6.8 years of experience in the field. Specifically, **P1**, **P3**, **P6-8** work on Variational Quantum Circuits (VQCs). **P4-5** specialize in Quantum Reinforcement Learning, and **P2** and **P9** focuses on Quantum Chemistry and Quantum Federated Learning. Among them, **P8** and **P9** are the collaborators of this research project.

Process. To develop a visualization framework that effectively addresses the needs of quantum computing domain experts, we adopted a problem-driven methodology inspired by the design study framework. In five months, we engaged in a structured and iterative collaboration with domain experts to identify challenges, gather requirements and refine our solution through continuous feedback and evaluation.

Learning phase: We conducted one-on-one, semi-structured interviews with all domain experts, aiming to elicit insights about their workflows and challenges regarding the interpretability of quantum encoders. During this phase, we focused on understanding the specific problems faced by the experts in analyzing quantum encoders, as well as identifying gaps in existing tools and methods. The interviews were complemented by an analysis of the experts' real tasks and datasets to ground our understanding in their real-world context.

Design phase: We then synthesized the insights gathered from the interviews and task analysis into a set of initial design requirements. Based on these requirements, we developed a low-fidelity prototype aimed at addressing the core interpretability challenges identified during the interviews to provide interpretable insights into quantum encoders.

Evaluation phase: We further engaged the domain experts in iterative expert reviews of the prototype. These reviews were conducted in think-aloud sessions, where experts were encouraged to freely explore the prototype and articulate their thoughts, concerns, and suggestions. This feedback was instrumental in identifying usability issues, refining the visualization techniques and ensuring that the tool aligned with the experts' workflows. Each iteration of the prototype was followed by targeted refinements, addressing specific concerns raised during the reviews while ensuring that the overall design remained consistent with the identified requirements.

Deployment phase: Finally, we delivered a refined version of the visualization framework to the domain experts for extended use in their tasks. This phase allowed us to validate the usability and effectiveness of the framework in addressing the interpretability challenges of quantum circuits. Feedback from this phase was incorporated into our discussion, the lessons learned documented to inform future work.

4.2 Design Requirements

We collected all feedback from the formative study and compiled five design requirements towards supporting a better understanding of the quantum data encoder.

R1 Visualize the encoded feature map. Most experts (**P2**, **P4**, **P7-9**) strongly agreed that it would be helpful to show the encoded feature map to users, which could directly show how the encoder processed the original dataset by the visual comparison. **P5-6** also suggested the decision boundary between the encoded feature map and the original dataset could highlight how well the encoder distinguishes the different classes.

R2 Support the analysis of the mixture of quantum states. Five experts (**P1**, **P3-5**, **P7**) expected to observe the encoded data from a "quantum" perspective. Specifically, they suggested enabling the analysis of how the quantum states with different classes are mixed, which can directly reflect the quality of the encoding process. The motivation, mentioned by **P7**, is that the quantum-version mixed states could be more accurate than the measured results from the encoder in some cases (i.e., the encoded feature map mentioned in R1) because the quantum states may vary even though the measured values are the same.

R3 Unveil the process of how data points are encoded. According to the feedback from four experts (**P1-2**, **P5-6**), it will be helpful to reflect the process of data encoding, which can inform the users of the functionality of each quantum gate in a data encoder. Moreover, **P2** commented that "*The explanation of the intermediate state in each step means everything for the developers.*"

R4 Enable flexible trials with different encoders and datasets. To better support the understanding process, five experts (**P2**, **P4-6**, **P9**) suggested that the visualization system should not be limited to a fixed example and instead enable flexible experiments with different data encoders along with various datasets. **P5** emphasized that QNN users can only gain the key knowledge of how the encoders capture the features by back-and-forth comparisons.

R5 Bridge the gap with the classical neural network. Five experts (**P2-3**, **P5-7**) appreciated systems that can integrate the conventional visualizations for model performance tracing, e.g., the line charts for visualizing the accuracy changes across epochs. Meanwhile, they all pointed out the importance of supporting the fine-tuning of the hyperparameters (e.g., learning rates), which can connect the dots between the training of classical neural networks. **P2** also encouraged us to enable the observation of the trained feature map after each epoch of the training process has been completed, which could support a direct analysis of the training for each data point.

5 DATASETS

In this paper, we use 2-dimensional data as examples to illustrate the effectiveness of the proposed visualization tool. As depicted in Figure 1C, each cell in the grid represents an individual data point. The x and y axes correspond to the two features of the dataset, while the color of each cell denotes the label of the respective data point. To process this dataset within a quantum neural network framework, we utilized the 2-qubit QNNs, which consist of the following three components: 1) *Encoder*: transforming the 2-dimensional classical data points into quantum states, which can be represented by a 4×4 matrix; 2) *Ansatz*: A parameterized quantum circuit that applies single-qubit Pauli rotation gates to the above encoded states. For all examples integrated in our visualization system, we fixed the ansatz to support the analysis of the encoder's pattern by implementing different encoders.

6 XQAI-EYES

We proposed *XQAI-Eyes*, a visualization tool to facilitate the understanding of how the quantum encoder affects the training of QNNs. In this section, we first introduce the *Encoder Expectation Measurement* to simulate the encoded states before they are fed into the training model. Then, we illustrate the developed visualization system *XQAI-Eyes* together with the justification of the system design. Note that *Encoder Expectation Measurement* and the visualization system are based on the simulator provided by the *Pennylane* [4] framework.

6.1 Encoder Expectation Measurement

In QNNs, directly measuring the quantum states after the encoder layer would immediately terminate the training process, because any measurements will collapse the quantum states, leaving no valid information for subsequent layers (i.e., the ansatz) in the next epoch. To address this challenge and obtain the encoded quantum states between

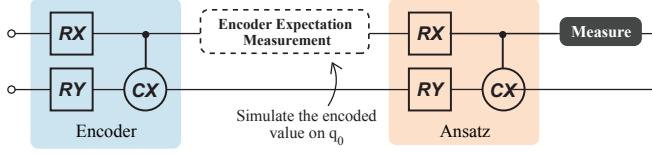


Fig. 2: The illustration of the Encoder Expectation Measurement simulation without directly implementing the measurement gate. We simulate the encoded states on the first qubit, aligning with the configuration of the actual measurement of q_0 .

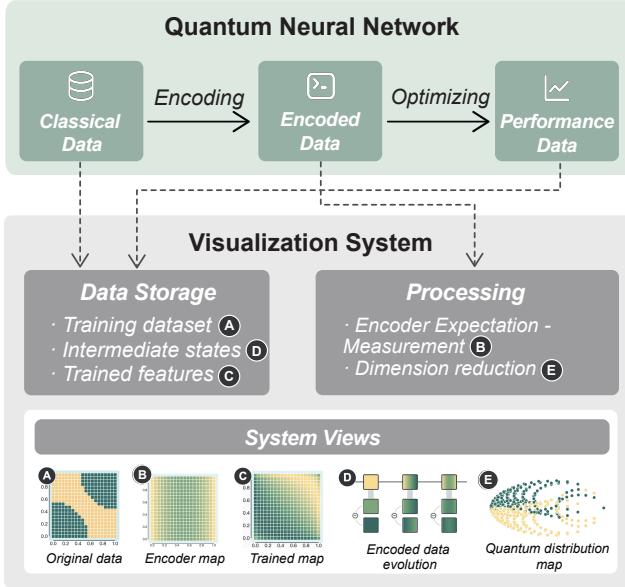


Fig. 3: The system of XQAI-Eyes contains three modules: a Data Storage module, a Processing module, and a System Views module. The Encoded Data will be extracted from the QNN training process and then delivered to the Processing module for the generation of the two views, i.e., the Encoder map and the State Comparison Map.

the encoder and ansatz layers, we propose a novel approach *Encoder Expectation Measurement* to calculate the encoder expectation values.

To achieve this, we first extract the quantum system states by setting the flag after the encoder layer ends, which provides the density matrix of each quantum state, as shown in Equation 2. This step allows us to capture the encoded quantum state without directly measuring it. As shown in Figure 2, since the first qubit (q_0) is measured as the output of the quantum circuit in every training epoch, the probabilities of q_0 being in the states $|0\rangle$ and $|1\rangle$ are calculated, denoted as $\text{Prob}(q_0 = 0)$ and $\text{Prob}(q_0 = 1)$, respectively. So, the expectation value could be $\text{Prob}(q_0 = 0) - \text{Prob}(q_0 = 1)$. Given the method of single-qubit probability calculation, the expectation value of the encoded quantum state at the encoder output can be computed as:

$$E_{\text{encoder}} = \sum_{i \in \{0,1\}} \text{Pr}(|0i\rangle) - \sum_{j \in \{0,1\}} \text{Pr}(|1j\rangle), \quad (4)$$

where E_{encoder} represents the encoded value of the corresponding classical data point. This expectation value is then used as the encoded representation of the classical data point, ensuring that the encoded quantum state is preserved and passed to the ansatz layer without collapsing the quantum circuit.

6.2 Visual Interface

In this subsection, we illustrate the visualization system XQAI-Eyes in terms of the implemented views, followed by the user interactions and

design justifications. Figure 3 shows the architecture of XQAI-Eyes and how it integrates with the external QNNs.

6.2.1 Classical Data Representation

To provide users with an intuitive understanding of the original data and facilitate comparisons with the encoded quantum states, we developed the “Original Dataset View”. Shown in Figure 4(A), this view visualizes the original two-dimensional input before it is encoded into quantum states. Specifically, each data point is represented as a cell, where the horizontal and vertical positions correspond to the two dimensions of the dataset. The data points are color-coded to indicate their respective classes. To standardize the encoding process, we assign numerical values of -1 and $+1$ to the two classes, respectively. Note that for all datasets implemented in the system, the horizontal and vertical features are evenly distributed within the range of 0 to 1 .

6.2.2 Encoded Data Representation

We present the Encoder Map View to provide users with a direct visualization of the quantum states after the encoding process, enabling them to observe and analyze the patterns generated by the encoder. As described in Section 2, we leverage the Encoder Expectation Measurement to compute the encoded quantum states by mapping it back to a scalar value. This scalar value serves as a bridge between the complex quantum states and an interpretable visual representation, allowing users to intuitively understand the encoding process. Then, based on the scalar values, the Encoded Map View visualizes all encoded data points in a heatmap-like view, as shown in Figure 4(B). The principles of the visual design of the above Encoded Map are consistent with the Original Dataset View. The key difference lies in the color encoding, where the color of each cell in the Encoded Map represents the expectation value of the encoded quantum states for the corresponding data point, ranging from -1 to 1 .

To further support the rationale behind the Encoded Map, we developed a visualization called the Encoded Data Evolution View, as shown in Figure 4(D). It illustrates how the encoded map is generated during the encoding process before the encoder is finalized. The first row of heatmaps in this view depicts the intermediate states of the encoder at each step of the encoding process, providing users with insights into how the encoding evolves over time. Moreover, to explain the intermediate quantum states in greater detail, the second and third rows of the Encoded Data Evolution View visualize the probabilities of qubit q_0 being in states $|0\rangle$ and $|1\rangle$, respectively, aligning with Equation 4.

6.2.3 State Comparison Map

To better portray the encoded data, we presented another visualization called “State Comparison Map” (Figure 4(E)), a two-dimensional scatter plot designed to illustrate how well the different-label data points are encoded, which can intuitively reflect encoder’s ability to distinguish between different classes of data.

The system captures intermediate quantum states at designated points, which are placed at key stages. For each data point, the extracted state provides a complex state vector, which is subsequently converted into a density matrix representation. Given that even a small number of qubits can result in high-dimensional density matrices, i.e., the density matrix with the size of $2^N \times 2^N$ for N -qubit case, as illustrated in Equation 2. We then employ dimensionality reduction techniques to create a more interpretable representation. The density matrix is first flattened into a single numeric array by Principal Component Analysis (PCA) to reduce these high-dimensional arrays into 2D coordinates. Unlike nonlinear methods such as t-SNE, we chose PCA because it provides a deterministic projection that preserves the global variance structure of the encoded quantum states and also allows a direct linear mapping from the high-dimensional density matrix to two principal axes, aligning with the system’s goal of enabling transparent and reproducible visual reasoning. PCA identifies the directions (i.e., principal components) along which the data exhibits the greatest variance and projects each flattened density matrix onto these components. This processing stage can capture the most significant features of the

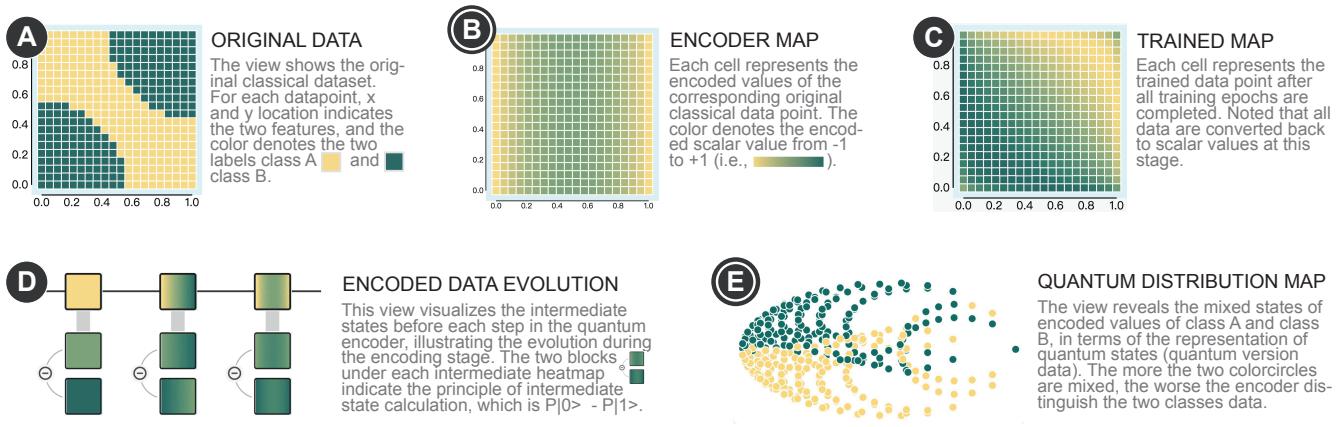


Fig. 4: All views implemented in *XQAI-Eyes*. (A) The visualization of the original 2-dimensional training dataset. (B) The Encoded Map view supports the direct inspection of the encoded classical data, where each cell represents the expectation value of the quantum states. (C) The Trained Map view shows the features of the trained dataset after the training process is finished. (D) The Encoded Data Evolution view shows the evolution before the all data points are completely encoded, where each group of heatmap indicates the states before each step in the encoding process. (E) The State Comparison Map view denotes the mixed states of the two-class data points, assessing the encoder’s performance to distinguish between the features of two-class data points. Note that the work shown in sub-figures (B) and (E) are the key contributions of our paper.

quantum states while ensuring the data remains visually interpretable. Each projected sample is paired with its original class label (e.g., +1 or -1). Once the PCA-generated coordinates are paired with their class labels, the system visualizes the data as a two-dimensional scatter plot. As shown in Figure 4E, each data point is positioned according to its corresponding PCA coordinates, with its class label determining its color, allowing users to visually assess the separability of different classes that varies significantly depending on the encoder design.

For example, as indicated by Dataset 1 and State Comparison Map 1 in Figure 5, when the encoder effectively separates the classes, the scatter plot displays distinct clusters, with each cluster corresponding to a specific class. Such clear separation indicates that the encoding process preserves meaningful feature distinctions in the quantum state space, leading to a decent accuracy of 95%. Conversely, for the poorly-encoded data like Dataset 2 in Figure 5, if the scatter plot shows extensive mixed patterns between multiple classes, the encoder fails to promote sufficient differentiation. In such cases, the prediction accuracy is usually very low (55% in this case) due to the mismatch of the original dataset and corresponding Encoder Map.

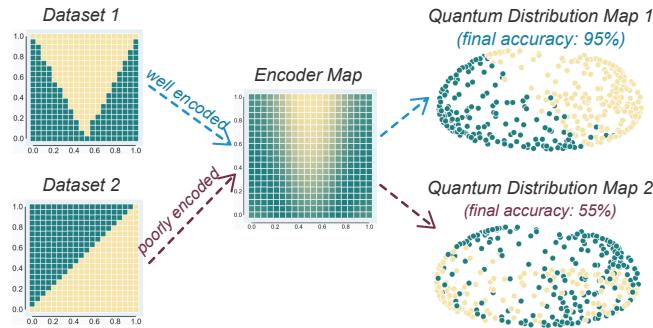


Fig. 5: The State Comparison Map compares two scenarios. Top: a well-encoded scenario with an accuracy of 95%, where distinct clusters corresponding to different classes are clearly visible; Bottom: a poorly-encoded scenario with an accuracy of 55%, where overlapping clusters indicate insufficient class differentiation.

6.2.4 Model Performance Analysis

To bridge the gap between quantum and classical neural networks training, we implemented a widely used approach to visualize the changes in

performance throughout the training process. As shown in Figure 1H, we utilized line charts to depict the training loss and accuracy across each epoch of the QNN. Specifically, the blue line represents the training loss, while the orange line illustrates the accuracy. Furthermore, to better explain the accuracy, we developed another view to directly visualize the trained dataset and the resulting classification. As shown in Figure 4C, this view presents the learned patterns after the training process is finalized. Each cell in this visualization corresponds to a data point, with its color representing the predicted value for that point. This design allows users to compare the original dataset with the trained dataset, offering a direct understanding of how well the model has captured the underlying patterns.

6.2.5 User Interactions

We also developed various user interactions to support the flexible exploration of *XQAI-Eyes* in terms of the QNN model training and the pattern comparison of original data and encoded data.

Encoder selection. To enable users to explore a wide range of encoder options, we implemented an encoder selection feature. As shown in Figure 1A, users can click the arrows in the component named “Specify Encoder” to open a side-selection panel, allowing users to freely select from a set of pre-configured encoder circuits. Specifically, we prepared a total of 10 2-qubit encoder circuits, incorporating various combinations of Pauli Rotation Gates and Control Gates.

Model configuration. Users can also control the training process through a control panel. For example, the panel allows users to configure key hyperparameters such as the number of training iterations and the learning rate. These parameters can be adjusted either by directly inputting the desired values or by dragging the slider handles to fine-tune the settings. Additionally, the control panel includes buttons that enable users to pause or resume the training process at any time, offering greater flexibility in managing the execution of the QNN model. This interactive control mechanism ensures that users can experiment with different training setups and transparentize their effects on the QNN model’s behavior.

6.2.6 Design Justification

Based on the interactive test of the deployment with the participants during the formative study, we carefully refined *XQAI-Eyes* by receiving valuable feedback. Below, we outline the key design decisions and their justifications:

Cell shape. Initially, we used circular shapes to represent each data point in the heatmap. However, the gaps between the circles made it

difficult to inspect the overall patterns, as the empty spaces diluted the visual continuity of the heatmap. To mitigate this issue, we switched to square cells, which can seamlessly tile without gaps, allowing a clearer comparison of the two colors representing different classes.

Color distribution. We initially employed a color range with three key points (e.g., $-1 \rightarrow +1$), where white was used as the middle point, but this approach caused the colors near the white midpoint to appear too faint. In response, we changed the color distribution to use a two-keypoint range by eliminating the white midpoint.

Circuit display. Feedback from the experts highlighted the importance of aligning the visualization of encoded data with the actual encoder circuit used, which provides a direct reference to the encoder's structure, so we implemented the encoder circuit diagram in *XQAI-Eyes*.

Color palette selection. Since the encoded data generated by the Encoder Expectation Measurement often consists of decimal values, the palette needed to clearly differentiate between the two classes while preserving subtle variations in the data. We experimented with a number of palettes (candidates:                      <img alt="color palette 330" data-bbox="18968 298

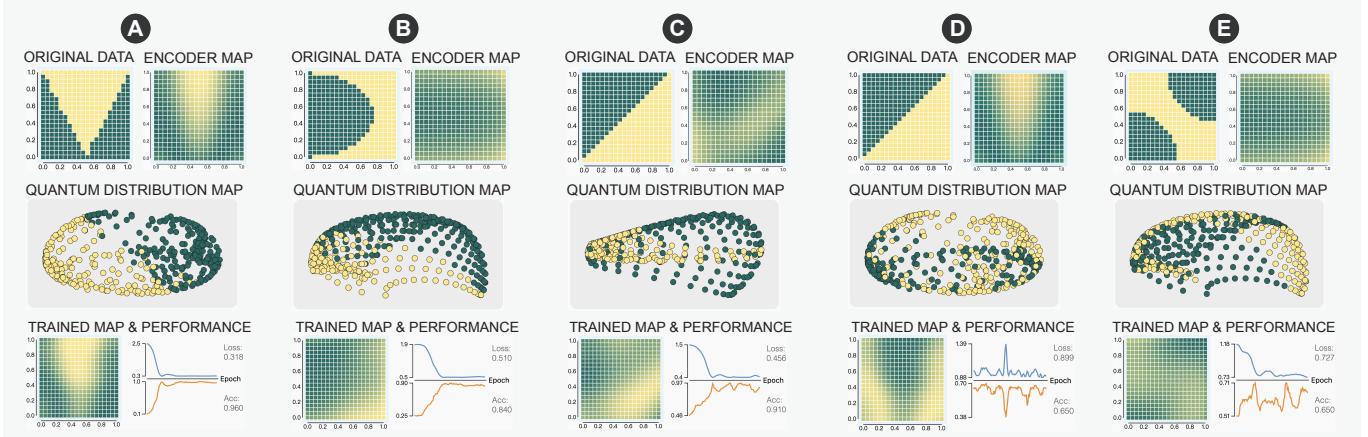


Fig. 6: The results in different usage scenarios, with the encoding performance varying from well-coded to poorly-encoded levels. (A)(B) The Encoder Map and Quantum Distribution Map show the encoder encodes the original dataset well, resulting in good accuracy. (C) The Encoder Map shows the encoder mapped the features close to the original datasets, but the Quantum Distribution Map shows the encoder still cannot separate different classes apart. (D)(E) The Encoder Map and Quantum Distribution Map perform badly in encoding the original datasets, both leading to bad accuracy.

in Figure 6(A). They noticed that the encoded states look very similar to the sixth dataset. So, they chose this combination to validate their hypothesis. After launching the training and inspecting the Quantum Distribution Map, P2 reported: “*The Encoded Map view shows the encoder is good enough for this dataset, which reduces the training burden for the ansatz because this encoder has successfully simplified the classification task. The Quantum Distribution Map confirms this, as the two data points are well-separated, showing that the encoding process effectively captured the features of the two classes. As a result, the model accuracy shows excellent performance achieving 96%, with a fast convergence curve and minimal fluctuation.*”

The participants continued their exploration by selecting another dataset. As shown in Figure 6(B), they found that the Encoder Map closely resembled the dataset. However, the Quantum Distribution Map was not as good as in the previous example, with some overlapping and disorganized data points at the boundaries. After starting training, the results showed that, although the accuracy was still decent, the convergence process was longer compared to Scenario A. P2 commented: “*This roughly confirms our hypothesis, but I would like to observe the behavior of some other bad scenarios.*”

Next, they switched to the dataset shown in Figure 6(C). This scenario was similar to Scenario B, but the Quantum Distribution Map was more messy, with many data points clustered around the middle horizontal region. P3 reported: “*That’s why the line is more fluctuated, that’s very interesting.*”

Curious about the performance when the dataset and encoder map were completely different, as shown in Figure 6(D), they kept the dataset from Scenario C but switched to the encoder used in Scenario A. After the comparison, they observed that under this scenario, with a poor Encoder Map and a messy Quantum Distribution Map, the model struggled to converge. The accuracy curve showed significant fluctuations, and even after 100 epochs, the accuracy performance only reached approximately 65%. A similar pattern was observed in Figure 6(E), where a poor encoder and distribution map resulted in a prolonged convergence process and a highly fluctuating curve.

Conclusion. The experts were excited about the findings in the exploration. After a short discussion, they concluded that, while the visualization system was highly effective, it also validated two key findings in their research. They summarized the practice as follows: If the objective is to optimize QNN model performance, two strategies can be considered regarding the quantum encoder. The first strategy involves designing the encoder to make feature representations closely aligned with the characteristics of the training dataset, thereby reduc-

ing the complexity of the subsequent optimization task—effectively shortening the optimization trajectory. The second strategy, as shown in the Quantum Distribution Map, is to encode data points such that features from different classes become clearly distinguishable, thereby resulting in smoother convergence during model training. However, the ultimate optimal model performance still relies on the combined effect of the quantum encoder and the ansatz, but the ansatz design would be a different story, which is beyond the scope of this work.

7.3 Expert Interview

We conducted post-study interviews with the domain experts to gather qualitative feedback on the effectiveness, usability, workflow and visual design of *XQAI-Eyes*.

Effectiveness. The experts were asked about how effective *XQAI-Eyes* is on understanding the correlation between encoders and model performance, aligning with the design requirements R1-4. The experts provide positive feedback regarding the system’s ability to uncover hidden insights about the quality of encoders, which was barely studied in the previous quantum machine learning area. P2 highlighted, “*From the perspective of our findings, I believe XQAI-Eyes does an excellent job of revealing hidden insights about the encoder’s quality,*” because previously, developers of VQCs were essentially performing blind executions, but this system truly “visualizes” the diagnostic capabilities of encoders. P1 emphasized the innovation of the Encoder Expectation Measurement within the quantum computing field, stating, “*From a quantum computing perspective, Encoder Expectation Measurement itself is a significant novel approach in the area of explainable quantum neural networks. Since explainability in QNNs is still an under-explored research direction, this contribution is impactful.*” P3 also praised the Quantum Distribution Map, commenting, “*I’m still surprised by the intuitive visualization of the Quantum Distribution Map. This design makes the encoder’s performance immediately clear, showing how far the encoded states are from the maximum mixed states based on the distribution of data points. This is absolutely critical.*”

Usability. The system’s usability was highly regarded, especially for domain experts without a visualization background. P3 remarked, “*In general, this visualization system is very user-friendly for domain experts without a visualization background. We often use heatmaps to analyze performance, so the heatmaps in this system are simple and practical for us. Compared to other tailored visualization systems we have used before, they often confused us, while XQAI-Eyes feels like a very practical dashboard. I could figure out how to use it even without a system introduction.*” P2 appreciated the system’s support for free exploration, noting that the system allows users to freely explore and compare various encoder templates and datasets, enabling

a comprehensive understanding in encoders. Additionally, **P2** praised the clear tutorial at the bottom of the interface, stating that the tutorial provides a concise introduction to the interface and includes helpful explanations of QNNs and encoders, making it especially friendly for novice students.

Workflow. The system’s workflow was described as intuitive and logical, guiding users seamlessly through the analysis process. **P1** commented that the system’s layout and transitions effectively guide user interactions, allowing users to understand how to interact with the system without needing a tutorial. He also noted that the current workflow—starting with training and then analyzing the encoder’s quality and performance through the Encoder Map and Quantum Distribution Map—is both reasonable and effective. Furthermore, **P1** suggested an enhancement that the Quantum Distribution Map could be used to pre-test encoders that perform well for specific types of datasets. Such encoders could then serve as templates for similar datasets in the future.

Visual Design and Interactions. The visual design and interactions of *XQAI-Eyes* received positive feedback from the experts. **P2** commended the intuitive representation of the Quantum Distribution Map, saying that the Quantum Distribution Map clearly visualizes the degree of confusion in the encoded states. Additionally, the design and color choices for all the heatmaps are excellent, making it easy to analyze encoded and learned patterns. **P3** also highlighted the value of the Encoded Data Evolution view and reported that it provides an in-depth explanation of how different gates impact the dataset from the perspective of gate implementation. It is incredibly helpful for understanding the role of different rotation gates and offers valuable knowledge for designing encoders in the future. Finally, **P3** praised the smooth interactions within the system, particularly during the training process, “*The interactions are very smooth, especially the ability to pause training at any epoch to inspect the intermediate states. This functionality is extremely practical.*”

Suggestions. Besides the positive feedback, several participants also offered constructive suggestions. First, experts emphasized the need to involve a larger community of researchers to contribute to the system. This expansion could facilitate the identification and design of new typical encoders tailored for datasets with specific features. By integrating more diverse encoders, the system could broaden the community’s knowledge base, leveraging intuitive visual representations to drive further advancements. Second, to support deeper analysis, the system could allow users to trace specific data points of interest. For example, incorporating brushing interactions in the Trained Map View would enable backward-analysis for poorly-encoded or mixed data points.

8 DISCUSSION AND FUTURE WORK

We further discuss four aspects in our research.

Explainable Quantum Artificial Intelligence. Explainable artificial intelligence (XAI) has emerged as an active area of research in classical machine learning, aiming to make AI systems more transparent, interpretable, and trustworthy. In the context of quantum artificial intelligence, the need for explainability is even more pressing due to the inherent complexity and non-intuitive nature of quantum mechanics. According to our expert interview, they all agreed that this paper represents a foundational step toward the emerging field of XQAI by introducing a method to calculate and visualize the encoder expectation value. The methodology proposed in this paper aligns with the goals of XQAI by addressing the “black-box” nature of quantum encoders. In the future, as the field of quantum computing continues to advance, XQAI will play an important role in ensuring that quantum systems are not only powerful but also interpretable.

Potential Impacts. The proposed approach for calculating and visualizing the encoder expectation value demonstrates significant potential for advancing the understanding and usability of QNNs across multiple domains. First, the approach outlined in this paper has the potential in educational scenarios, particularly for lecturers and instructors aiming to teach quantum computing and quantum neural networks. Second, for novice developers working on quantum machine learning models, the proposed approach offers an interpretable method for understanding the encoders. Furthermore, the ability to visualize the encoded

states helps identify potential issues or inefficiencies in the encoding process. Furthermore, the proposed method also has implications for quantum circuit designers. By providing an intuitive understanding of the encoder’s performance, this approach facilitates the identification of optimal encoding strategies for specific applications. Additionally, the visualization techniques introduced in this paper provide a novel way to analyze and refine quantum circuits, helping designers intuitively explore the relationship between the encoder and model performance.

Scalability and Generalizability. In this paper, we demonstrated the proposed method using a two-dimensional dataset example, which allowed us to visualize the quantum states in a two-dimensional heatmap-like representation. Our paper focuses on using the 2-dimensional dataset to build a cognitive model for QNN developers, because 2D datasets are straightforward and users can understand without much time and efforts, which is suitable for enhancing the explainability of quantum encoders. However, for more complicated real-world datasets, our system, especially our major contributions “Encoded Data Representation” and “State Comparison Map”, can be generalized by incorporating dimensionality reduction techniques, which can reduce the high-dimensional quantum states into a lower-dimensional representation, which can suit *XQAI-Eyes* while preserving the essential features of the data. Furthermore, while the primary focus of this work is on analyzing and interpreting quantum encoders, the proposed methodology has the potential to be generalized to the optimization process of QNNs, e.g., ansatz layers. For example, the presence of barren plateaus and vanishing gradients in quantum optimization could be better solved by the proposed method.

Future Work. The current system operates on static datasets and pre-defined quantum circuits. Future work could focus on integrating adaptive learning algorithms into the system, allowing it to dynamically adjust its parameters in response to changes in the input data or circuit configurations. Moreover, the system is tailored for specific tasks at the moment, such as analyzing quantum encoders. Future work could extend its functionality by enabling it to handle diverse quantum machine learning tasks, such as classification, regression, and clustering, within a single framework. Additionally, future work can be further applied to quantum federated learning [32] within a distributed framework.

9 CONCLUSION

In this work, we identified key challenges for improving the explainability of quantum encoders. Following the distilled design requirements collected from the formative study, we developed *XQAI-Eyes*, a visualization system designed to enhance the understanding of quantum encoders and their role in QNNs. To obtain the encoded data before the optimization process, we introduced *Encoder Expectation Measurement*, which can extract and convert the intermediate encoded states into scalar values. Furthermore, from the perspective of quantum, we presented the approach to intuitively analyze whether the encoder effectively distinguishes the features of different classes. We also conducted case studies and expert interviews with domain users, whose feedback highlights the effectiveness and usability of *XQAI-Eyes* in addressing the challenges of encoders’ explainability.

Two key takeaways from the evaluation of *XQAI-Eyes* are: First, designing encoders whose features closely align with the training dataset can reduce the optimization burden, effectively simplifying the learning process. This finding aligns with the insights collected from the analysis of the Encoder Map View. Second, encoding data points in a way that ensures the distinguishability of data points with different classes, as visualized by the Quantum Distribution Map, leads to smoother convergence and better performance.

ACKNOWLEDGMENTS

This project is supported by the Ministry of Education, Singapore, under Academic Research Fund Tier 2 (Proposal ID: T2EP202220049). Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the Ministry of Education, Singapore.

REFERENCES

- [1] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021. [2](#)
- [2] L. S. Abuga. *Comparative Analysis of Quantum Encoding Methods in Quantum Neural Networks for Image Classification on the MNIST Dataset*. PhD thesis, University of Nairobi, 2023. [3](#)
- [3] M. Afane, G. Ebbrecht, Y. Wang, J. Chen, and J. Farooq. Atp: Adaptive threshold pruning for efficient data encoding in quantum neural networks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 20427–20436, 2025. [2](#)
- [4] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. D. Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill, A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O’Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran. PennyLane: Automatic differentiation of hybrid quantum-classical computations, 2022. [4](#)
- [5] E. W. Bethel, M. G. Amankwah, J. Balewski, R. Van Beeumen, D. Camps, D. Huang, and T. Perciano. Quantum computing and visualization: A disruptive technological change ahead. *IEEE Computer Graphics and Applications*, 43(6):101–111, 2023. [2](#)
- [6] F. Bloch. Nuclear induction. *Physical Review*, 70(7-8):460, 1946. doi: [10.1103/PhysRev.70.460](#) [2](#)
- [7] F. Cheng, D. Liu, F. Du, Y. Lin, A. Zytek, H. Li, H. Qu, and K. Veeramachaneni. VBridge: Connecting the dots between features and data to explain healthcare models. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):378–388, 2021. [3](#)
- [8] S. Chung, C. Park, S. Suh, K. Kang, J. Choo, and B. C. Kwon. Re-vacnn: Steering convolutional neural network via real-time visual analytics. In *Proceedings of Future of Interactive Learning Machines Workshop at the 30th Annual Conference on Neural Information Processing Systems*, 2016. [3](#)
- [9] A. J. da Silva and R. L. F. de Oliveira. Neural networks architecture evaluation in a quantum computer. In *Proceedings of 2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pp. 163–168, 2017. doi: [10.1109/BRACIS.2017.83](#) [2](#)
- [10] M. Heidari, A. Grama, and W. Szpankowski. Toward physically realizable quantum neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 6902–6909, 2022. [2](#)
- [11] T. Hubregtsen, J. Pichlmeier, P. Stecher, and K. Bertels. Evaluation of parameterized quantum circuits: on the relation between classification accuracy, expressibility, and entangling capability. *Quantum Machine Intelligence*, 3:1–19, 2021. [3](#)
- [12] IBM Quantum: An Online Interactive and Web-based Graphical Interface for Quantum Circuits. <https://quantum.ibm.com/>, Access Date: 2025-2-11. [2](#)
- [13] Z. Jin, Y. Wang, Q. Wang, Y. Ming, T. Ma, and H. Qu. Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 2022. [3](#)
- [14] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau. Activis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):88–97, 2017. doi: [10.1109/TVCG.2017.2744718](#) [3](#)
- [15] I. G. Karafyllidis. Visualization of the quantum fourier transform using a quantum computer simulator. *Quantum Information Processing*, 2:271–288, 2003. [2](#)
- [16] H. Kim, M. J. Jeng, and K. N. Smith. Toward human-quantum computer interaction: Interface techniques for usable quantum computing. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI ’25, article no. 580, 18 pages. Association for Computing Machinery, New York, NY, USA, 2025. doi: [10.1145/3706598.3713370](#) [2](#)
- [17] Y.-J. Lin and T. S. Chang. Data and hardware efficient design for convolutional neural network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(5):1642–1651, 2017. [3](#)
- [18] D. Liu, W. Cui, K. Jin, Y. Guo, and H. Qu. Deeptracker: Visualizing the training process of convolutional neural networks. *ACM Transactions on Intelligent Systems and Technology*, 10(1), article no. 6, 25 pages, nov 2018. doi: [10.1145/3200489](#) [3](#)
- [19] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang. Hybrid quantum-classical convolutional neural networks. *Science China Physics, Mechanics & Astronomy*, 64(9):290311, 2021. [2](#)
- [20] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu. Towards better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):91–100, 2016. [3](#)
- [21] H. Mäkelä and A. Messina. N-qubit states as points on the bloch sphere. *Physica Scripta*, 2010(T140):014054, 2010. doi: [10.1088/0031-8949/2010/T140/014054](#) [2](#)
- [22] D. Mattern, D. Martyniuk, H. Willems, F. Bergmann, and A. Paschke. Variational quanvolutional neural networks with enhanced image encoding, 2021. [2](#)
- [23] Y. Ming, H. Qu, and E. Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):342–352, 2018. doi: [10.1109/TVCG.2018.2864812](#) [3](#)
- [24] M. Monnet, N. Chaabani, T.-A. Drăgan, B. Schachtner, and J. M. Lorenz. Understanding the effects of data encoding on quantum-classical convolutional neural networks. In *Proceedings of 2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 01, pp. 1436–1446, 2024. doi: [10.1109/QCE60285.2024.00170](#) [3](#)
- [25] K. Nakaji, H. Tezuka, and N. Yamamoto. Quantum-classical hybrid neural networks in the neural tangent kernel regime. *Quantum Science and Technology*, 9(1):015022, Dec. 2023. doi: [10.1088/2058-9565/ad133e](#) [2](#)
- [26] T. Nguyen, I. Paik, Y. Watanobe, and T. C. Thang. An evaluation of hardware-efficient quantum neural networks for image data classification. *Electronics*, 11(3):437, 2022. [2](#)
- [27] H. Park, N. Das, R. Duggal, A. P. Wright, O. Shaikh, F. Hohman, and D. H. Polo Chau. Neurocartography: Scalable automatic visual summarization of concepts in deep neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):813–823, 2022. doi: [10.1109/TVCG.2021.3114858](#) [3](#)
- [28] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020. [3](#)
- [29] Q-Sphere: Visualization of Quantum States with Phase. https://deeplearninguniversity.com/qiskit/qiskit-q-sphere/#google_vignette, Access Date: 2025-2-11. [2](#)
- [30] Quirk: A Drag-and-drop Quantum Circuit Simulator. <https://algassert.com/quirk>, Access Date: 2025-2-11. [2](#)
- [31] M. Rath and H. Date. Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology*, 11(1):72, 2024. [3](#)
- [32] C. Ren, R. Yan, H. Zhu, H. Yu, M. Xu, Y. Shen, Y. Xu, M. Xiao, Z. Y. Dong, M. Skoglund, et al. Towards quantum federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.
- [33] J. Romero, J. P. Olson, and A. Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017. [3](#)
- [34] S. Ruan, Q. Guan, P. Griffin, Y. Mao, and Y. Wang. Quantumeyes: Towards better interpretability of quantum circuits. *IEEE Transactions on Visualization & Computer Graphics*, (01):1–13, 2023. [2](#)
- [35] S. Ruan, Z. Liang, Q. Guan, P. Griffin, X. Wen, Y. Lin, and Y. Wang. Violet: Visual analytics for explainable quantum neural networks. *IEEE Transactions on Visualization and Computer Graphics*, 2024. [2](#)
- [36] S. Ruan, R. Yuan, Q. Guan, Y. Lin, Y. Mao, W. Jiang, Z. Wang, W. Xu, and Y. Wang. VENUS: A Geometrical Representation for Quantum State Visualization. In *Computer Graphics Forum*, vol. 42, pp. 247–258. Wiley Online Library, 2023. doi: [10.1111/cgf.14827](#) [2](#)
- [37] B. Schneider, D. Jäckle, F. Stoffel, A. Diehl, J. Fuchs, and D. Keim. Integrating data and model space in ensemble learning by visual analytics. *IEEE Transactions on Big Data*, 7(3):483–496, 2018. [3](#)
- [38] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE transactions on visualization and computer graphics*, 18(12):2431–2440, 2012. [4](#)
- [39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017. [3](#)
- [40] F. Shen and J. Liu. Quantum fourier convolutional network. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(1):1–14, 2023. [2](#)

- [41] D. Sun, Z. Feng, Y. Chen, Y. Wang, J. Zeng, M. Yuan, T.-C. Pong, and H. Qu. Dfseer: A visual analytics approach to facilitate model selection for demand forecasting. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020. 3
- [42] A. Tyagi, C. Xie, and K. Mueller. Nas-navigator: Visual steering for explainable one-shot deep neural network synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 19(12), 3
- [43] Q. Wang, Y. Ming, Z. Jin, Q. Shen, D. Liu, M. J. Smith, K. Veeramachaneni, and H. Qu. Atmseer: Increasing transparency and controllability in automated machine learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019. 3
- [44] Q. Wang, Z. Xu, Z. Chen, Y. Wang, S. Liu, and H. Qu. Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1470–1480, 2020. 3
- [45] X. Wang, J. He, Z. Jin, M. Yang, Y. Wang, and H. Qu. M2Lens: Visualizing and explaining multimodal models for sentiment analysis. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):802–812, 2021. 3
- [46] Z. Wen, Y. Liu, S. Tan, J. Chen, M. Zhu, D. Han, J. Yin, M. Xu, and W. Chen. Quantivine: A visualization approach for large-scale quantum circuit representation and analysis. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 2
- [47] R. Wille, L. Burgholzer, and M. Artner. Visualizing decision diagrams for quantum computing (special session summary). In *Proceedings of 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 768–773. IEEE, 2021. doi: [10.23919/DATEx51398.2021.9474236](https://doi.org/10.23919/DATEx51398.2021.9474236) 2
- [48] M. M. Williams. *QCVis: A quantum circuit visualization and education platform for novices*. PhD thesis, Harvard University, 2021. 2
- [49] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *Proceedings of 2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 57–68. IEEE, 2019. 3
- [50] W. Yang, X. Ye, X. Zhang, L. Xiao, J. Xia, Z. Wang, J. Zhu, H. Pfister, and S. Liu. Diagnosing ensemble few-shot classifiers. *IEEE Transactions on Visualization and Computer Graphics*, 28(9):3292–3306, 2022. 3
- [51] W. J. Yun, J. Park, and J. Kim. Quantum multi-agent meta reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 11087–11095, 2023. 2
- [52] H. Zeng, H. Haleem, X. Plantaz, N. Cao, and H. Qu. Cnncomparator: Comparative analytics of convolutional neural networks, 2017. 3
- [53] J. Zhang, Y. Zhang, and Y. Zhou. Quantum-inspired spectral-spatial pyramid network for hyperspectral image classification. In *Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9925–9934, 2023. doi: [10.1109/CVPR52729.2023.00957](https://doi.org/10.1109/CVPR52729.2023.00957) 2
- [54] A. Zulehner, S. Hillmich, and R. Wille. How to efficiently handle complex values? implementing decision diagrams for quantum computing. In *Proceedings of 2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 1–7, 2019. doi: [10.1109/ICCAD45719.2019.8942057](https://doi.org/10.1109/ICCAD45719.2019.8942057) 2

A INTERVIEWS IN THE FORMATIVE STUDY

To ensure the design of our visualization system effectively meets the needs of domain experts in quantum computing, we conducted a structured formative study aimed at uncovering key challenges and opportunities in analyzing quantum data encoders. This study focused on gathering insights into workflows, interpretability issues, and usability concerns associated with quantum data encoders.

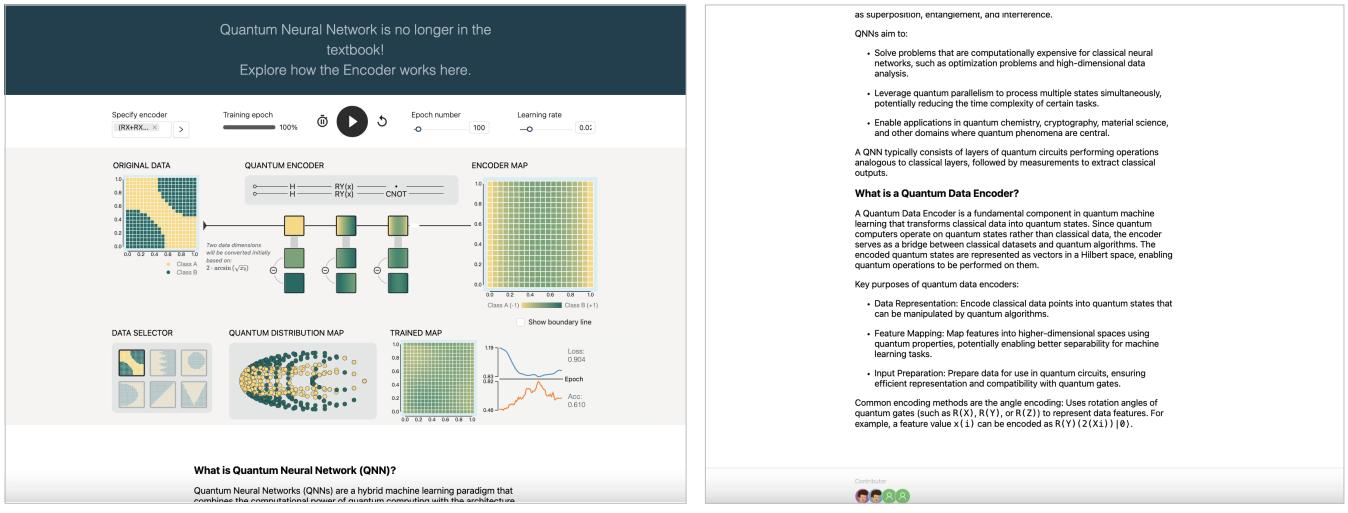
The interviews provided a comprehensive understanding of how experts approach encoder design, evaluate encoded feature maps, and analyze the intermediate quantum states during the encoding process. These discussions highlighted the importance of bridging the gap between classical and quantum neural networks, supporting flexible experimentation with diverse datasets, and enabling intuitive visualizations for performance analysis and debugging. Table 1 presents the key questions designed to elicit feedback from participants, focusing on their expectations for visualization systems that enhance interpretability, usability, and adaptability in quantum data encoding workflows.

Table 1: The pre-defined questions used in the preliminary study for the session of the design requirement collection.

Q1	What aspects of Quantum Neural Networks do you find most challenging to understand or implement?
Q2	How do you currently approach designing or selecting encoders for QNNs?
Q3	What types of visual aids would help you better analyze the performance of QNNs and encoders?
Q4	How do you interpret the outputs of QNNs, and what tools or methods do you use in this process?
Q5	What features should a visualization system include to make encoder designs more accessible and intuitive?
Q6	In what ways can visualization tools improve your ability to debug or optimize QNN architectures?
Q7	What strategies or features can reduce the learning curve for domain users working with QNNs and encoders?
Q8	How can visualization systems enhance the interpretability of QNNs for both researchers and practitioners?

B SYSTEM INTERFACE

At the forefront of the interface lies a prominent header, designed to capture the user's attention and provide context for the exploration of quantum neural networks and data encoding mechanisms. The central interface comprises several interconnected components that collectively illustrate the process of quantum data encoding and neural network training. These components are arranged to provide a logical flow of information, beginning with the presentation of classical data and culminating in the visualization of trained quantum neural network outputs. The interface integrates an interactive control panel at its uppermost section, enabling users to dynamically adjust critical parameters such as the encoding method, training epochs, learning rate, and the search space for optimization. To ensure users have a foundational understanding of the underlying principles, the system incorporates a comprehensive tutorial on quantum neural networks (QNNs) and quantum data encoders. This tutorial is designed to bridge the gap between classical machine learning concepts and quantum computing paradigms, enabling users to grasp the unique advantages and challenges associated with quantum neural networks. A specific focus is placed on angle encoding, a widely used method for quantum data representation. This technique utilizes the rotation angles of quantum gates, such as R_x , R_y , or R_z , to encode features of classical data. For instance, a feature value x_i can be represented as $R_y(2x_i)|0\rangle$, allowing for efficient integration into quantum circuits.



Part 1

Part 2

Fig. 7: The screenshot of the visualization system, which consists of three components, i.e., the header, the system interface, and the tutorial.