Quantum Machine Intelligence (2025) 7:24
https://doi.org/10.1007/s42484-025-00253-9

RESEARCH ARTICLE

# QRLaXAI: quantum representation learning and explainable AI

**Asitha Kottahachchi Kankanamge Don**[1] · **Ibrahim Khalil**[1]

Received: 16 November 2024 / Accepted: 6 February 2025 / Published online: 19 February 2025


**Abstract**

As machine learning grows increasingly complex due to big data and deep learning, model explainability has become essential to fostering user trust. Quantum machine learning (QML) has emerged as a promising field, leveraging quantum computing to enhance classical machine learning methods, particularly through quantum representation learning (QRL). QRL aims to provide more efficient and powerful machine learning capabilities on noisy intermediate-scale quantum (NISQ) devices. However, interpreting QRL models poses significant challenges due to the reliance on quantum gate-based parameterized circuits, which, while analogous to classical neural network layers, operate in the quantum domain. To address these challenges, we propose an explainable QRL framework combining a quantum autoencoder (QAE) with a variational quantum classifier (VQC) and incorporating theoretical and empirical explainability for image data. Our dual approach enhances model interpretability by integrating visual explanations via local interpretable model-agnostic explanations (LIME) and analytical insights using Shapley Additive Explanations (SHAP). These complementary methods provide a deeper understanding of the model's decision-making process based on prediction outcomes. Experimental evaluations on simulators and superconducting quantum hardware validate the effectiveness of the proposed framework for classification tasks, underscoring the importance of explainable representation learning in advancing QML towards more transparent and reliable applications.

**Keywords** Quantum machine learning · Quantum representation learning · Interpretable machine learning · Explainable AI

## 1 Introduction

Machine learning has emerged as a transformative force across various industries, revolutionizing applications in numerous domains. As research advances, machine learning models have grown increasingly complex, enhancing performance and reducing interpretability. These highly intricate models, such as deep neural networks, are often called "black-box" models due to their opacity, making it nearly impossible to understand how they function internally (Bodria et al. 2023). These models are typically built directly from data, and even their developers and data scientists cannot fully explain the inner workings of the reasoning behind specific outputs generated by artificial intelligence (AI) algorithms. Therefore, in addition to focusing on performance, the explainability of machine learning models has gained prominence (Molnar 2020). Explainable artificial intelligence (XAI) seeks to clarify how machine learning models operate, their potential biases, and their overall impact. XAI enhances transparency, fairness, and accuracy, particularly in decision-making processes powered by machine learning (Minh et al. 2022). It is essential for building trust and ensuring responsible AI deployment within organizations (Ahmed et al. 2022). Although a comprehensive review of XAI is beyond the scope of this paper, we refer readers to Minh et al. (2022); Adadi and Berrada (2018), and Ahmed et al. (2022) for further details.

In addition to explainability, another rapidly evolving branch of machine learning is quantum machine learning (QML) (Biamonte et al. 2017; Schuld and Petruccione 2018). Quantum computing has experienced significant advancements in recent years, especially with the rise of noisy intermediate-scale quantum (NISQ) devices (Preskill 2018; Lau et al. 2022). Giant big data companies, pioneers in quantum computing, have made substantial progress in quantum

Ibrahim Khalil contributed equally to this work.

✉ Asitha Kottahachchi Kankanamge Don
asitha.kottahachchi.kankanamge.don@student.rmit.edu.au

Ibrahim Khalil
ibrahim.khalil@rmit.edu.au

[1] School of Computing Technologies, RMIT University, P.O. Box 3000, Melbourne 3000, Victoria, Australia

error correction—a critical factor for achieving fault-tolerant quantum computing and real quantum speedup (Wootton and Loss 2012; Bravyi et al. 2024). Among the various applications of quantum computing, QML has gained attention due to its potential for significant speedup compared to classical machine learning (Schuld and Killoran 2019, 2022; Jerbi et al. 2024).

A key element in QML is the variational quantum classifier (VQC), which utilizes quantum gates to manipulate quantum states. In contrast, a subset of these gates includes trainable variational parameters (Benedetti et al. 2019; Cerezo et al. 2021, 2022). VQCs are the backbone for many QML applications (Schuld et al. 2020; Farhi and Neven 2018). However, VQCs on NISQ devices face several challenges, including the barren plateau issue (Li and Deng 2021), circuit optimization complexity (Jia et al. 2023), and qubit encoding problems (Maheshwari et al. 2022). One of the main challenges in variational quantum classifiers (VQCs) is the requirement of $n$ qubits to encode $n$ features, as each feature must be embedded into a quantum state. This significantly limits the scalability and efficiency of VQCs (Thumwanit et al. 2021; Belis et al. 2021; Yano et al. 2021). This qubit-intensive nature of quantum models makes it particularly challenging to evaluate explainability on simplified real-world datasets, such as benchmark image datasets, due to the high number of features and the associated qubit requirements. To overcome this, quantum representation learning (QRL) has been proposed as an emerging solution in QML research (Frohnert and Nieuwenburg 2024; Rivas et al. 2021; Don and Khalil 2024; Kottahachchi Kankanamge Don et al. 2024), which is the focus of this study.

Despite the potential for significant speedup over classical machine learning algorithms, QML algorithms including QRL models face numerous challenges, including hardware limitations, noise in quantum systems, and the complexity of classical preprocessing (Strobl et al. 2024; Ciliberto et al. 2017; Thakare 2023). These issues often overshadow the focus on speedup and contribute to significant explainability challenges, similar to those encountered in classical deep neural networks (see Fig. 1). The counter-intuitive and highly abstract internal workings of quantum gates and layers in quantum models further exacerbate these challenges (Pira and Ferrie 2024; Heese et al. 2023). Additionally, entanglements within quantum circuits obscure the contributions of individual components to model predictions, underscoring the importance of improving interpretability (Zhang et al. 2024; Rohe et al. 2024). The QRL models consist of several components that contribute to their overall architecture. The first component, the encoder, is a dimensionality reduction quantum circuit that reduces the feature dimension after transforming classical data into quantum states. This encoder circuit comprises a quantum feature map circuit and a variational circuit. The second component is the classifier, which is a VQC that classifies the reduced feature dimensions outputted by the encoder. The classifier also contains a data encoding circuit and a variational circuit but with a different quantum circuit architecture. Both components include a measurement circuit, which converts the quantum states back into classical scalar values. These three components increase the overall complexity and reduce transparency (Cerezo et al. 2021; Hur et al. 2022; Choudhury et al. 2021). And, the first two components make it difficult for quantum machine learning researchers and users of QRL to understand how

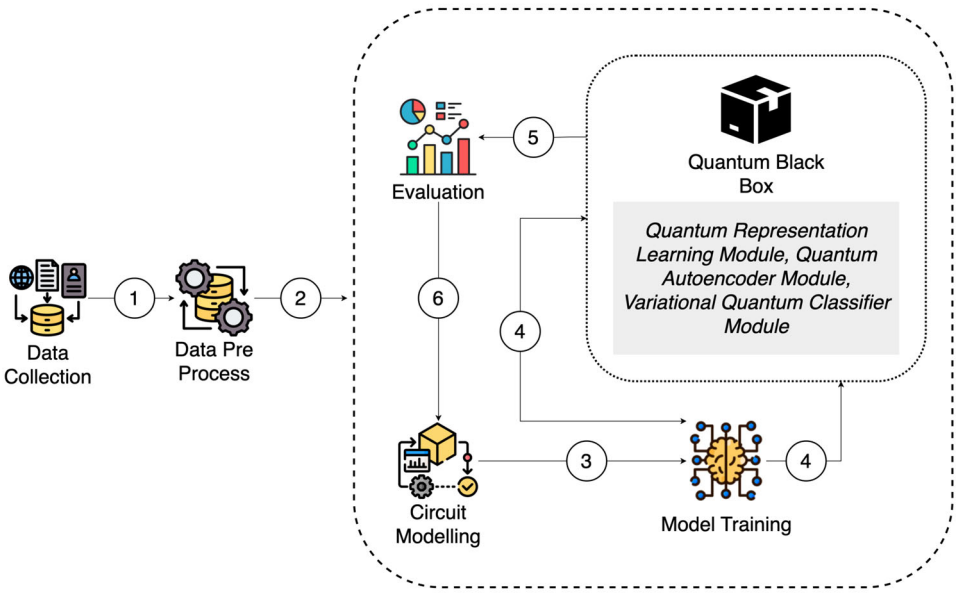


**Fig. 1** QRL black-box model: Steps (1) and (2) involve data collection and preprocessing. In step (3), the data is used to model the quantum circuit, followed by step (4), where the quantum model is trained, forming the quantum black box. Finally, in steps (5) and (6), the model is evaluated and fine-tuned, without incorporating explanations for the predictions

predictions are made based on feature representations and quantum gates. Additionally, the measurement component adds further difficulty, as interpreting how quantum circuits are measured and how results are converted back to scalar values is grounded in quantum mechanics, which is not easily intuitive for many users.

However, addressing all transparency issues related to QRL is not a trivial task. The qubit entanglements within quantum circuits add to the complexity of quantum gate operations. Although the measurement process converts quantum states into scalar values, representing the final prediction results of the quantum machine learning model, explaining how each of the three components within a QRL model contributes to the overall prediction remains a challenge. Previous studies have presented various visualization-based explainable approaches for classical deep neural networks (Choo and Liu 2018; Liu et al. 2017; Mohseni et al. 2021; Saranya and Subhashini 2023). Despite their effectiveness in explaining classical deep learning models, adapting these techniques to quantum machine learning models poses significant challenges due to the fundamentally different internal mechanisms between quantum models and classical deep neural networks. In response to this, prior research has proposed various visualization methods specifically designed for quantum neural networks (Choo and Liu 2018; Pira and Ferrie 2024). These studies introduced visual explainability techniques to enhance understanding of various components, such as data encoding circuits, variational circuits, and measurement processes. For example, Choo and Liu (2018); Pira and Ferrie (2024), and Heese et al. (2023) proposed the use of satellite charts and augmented heatmaps to analyze single-qubit states and expectation value measurements. Although several studies propose visual explainability techniques for quantum neural networks, they do not specifically address the explainability and performance of VQC models using a QRL framework. This gap is especially notable when applying model-agnostic explainability methods to benchmark image datasets under low-qubit constraints.

To address this research gap, we propose *QRLaXAI*, a QRL framework designed to collectively explain its two main components: the quantum feature extractor and the quantum classifier, both optimized for low qubit requirements. The quantum feature extractor transforms and compresses classical benchmark image datasets into low-dimensional quantum feature vectors, enabling efficient processing. The quantum classifier makes predictions based on the compressed quantum states generated by the feature extractor. Explanations are then produced using the original image data, the classically derived quantum states from the feature extractor, and the classifier's predicted probabilities, employing two model-agnostic explainability approaches. The first is a simplified breakdown of local interpretable model-agnostic explanations (LIME), which visually explains the step-by-step results of the QRL model based on its predictions. LIME is a technique used to describe how input features influence the predictions of a machine learning model by identifying the region of an image most strongly associated with a prediction label (Štrumbelj and Kononenko 2014). The simplified LIME breakdown in this study allows users to adjust the internal configurations of LIME to generate more detailed explanations. However, this method primarily provides overall model explainability rather than explanations for individual components. The second approach applies Shapley Additive Explanations (SHAP) to assess the classically derived quantum feature importance of the complete model. SHAP is a game-theoretic approach that explains the output of any machine learning model by assigning feature importance based on contribution to the predictions (Lundberg and Lee 2017).

The remainder of this manuscript is organized as follows: Sect. 2 reviews related works. Section 3 describes the proposed architecture in detail. Section 4 presents experimental results demonstrating the effectiveness of our approach in various applications. Finally, Sect. 5 concludes with a summary and potential directions for future research.

## 2 Related works

In this section, we outline the prerequisites for the proposed *QRLaXAI* method. To this end, we begin by providing a brief overview of classical representation learning, explaining how its explainability is measured and the methods commonly used for this purpose. We then review existing explainability techniques for QML models. Finally, we introduce two state-of-the-art explainability methods, LIME and SHAP, and describe how they are used to measure model explainability.

QRL builds upon classical representation learning, a powerful technique for extracting meaningful representations across various domains, such as computer vision and natural language processing (Chen et al. 2020; He et al. 2020). Classical representation learning methods often excel with fewer model parameters. The state-of-the-art method, supervised contrastive learning (SCL), incorporates a contrastive loss in supervised settings, enabling the network to differentiate between positive and negative pairs and thereby improving its ability to generalize to unseen data (Khosla et al. 2020). In Khosla et al. (2020), variants of the deep neural network were used as encoders, trained using the supervised contrastive loss. During training, 128 projection units were used to produce a 2048-dimensional encoded feature vector, which was then passed through a linear classifier. The study achieved a top-1 accuracy of 81.4% on the ImageNet dataset, surpassing the previously best-reported result for this architecture by 0.8%. Several other studies have also demon-

strated the effectiveness of representation learning, achieving remarkable accuracy (Wang and Qi 2023; Zheng et al. 2021; Hassani and Khasahmadi 2020).

Since classical representation learning has demonstrated promising results, several studies have explored the explainability and interpretability of these models using different techniques. For example, Xu et al. (2022) proposed an SCL-based detection model that utilizes contrast in the representation space to detect novel and unseen deepfake attacks. They further investigate the explainability of the learned features, advocating for score fusion with a deep neural network to enhance variability. Another recent work (Sammani et al. 2024) focused on visual explainability in contrastive learning, which is essential due to its reliance on interacting inputs and data augmentation. This study developed methods for interpreting the model's learning process by analyzing pairs of images. Additionally, the study on self-explaining deep models (Sarkar et al. 2022) introduced an approach that generates concept-based explanations during training, removing the need for post hoc explainability techniques. Despite the advancements, these approaches still exhibit limitations, such as a lack of in-depth feature attribution, over-reliance on task-specific methods, and limited human-centric evaluation and causal explanations. Several studies have proposed model-agnostic explainability methods to explain classical representation learning methods (Zhang et al. 2022; Shahariar et al. 2023; Yuan et al. 2024).

Various studies on explainable classical representation learning have employed diverse explainability techniques, with two main approaches to creating interpretable machine learning models, as outlined by Molnar (2020). The first approach involves designing an intrinsically interpretable predictive model, such as rule-based algorithms, or utilizing a black-box model with a surrogate model to provide explanations. The second approach is known as post hoc interpretability, which can be categorized into global models that explain the average behavior of the black-box model, and local models that explain individual predictions. Several tools are available for creating post hoc interpretable models, most of which are model-agnostic, meaning they can be applied regardless of the underlying algorithm. According to Molnar (2020), the most common post hoc interpretability techniques include partial dependence plot (PDP), accumulated local effects (ALE), individual conditional expectation (ICE), local interpretable model-agnostic explanations (LIME), and Shapley Additive Explanations (SHAP). PDP is a global, model-agnostic interpretation method that shows how individual features contribute to the prediction results of a black-box model (Hastie et al. 2009). However, PDP becomes less effective when models have more than two features, as the plots become harder to interpret. ALE, another global model-agnostic method, provides an alternative to PDP, but can still be biased when variables are highly correlated (Apley and Zhu 2020). While ALE uses a different methodology, it also struggles to capture complex interactions between multiple features, as it only considers a pair of features at a time.

Unlike PDP and ALE, ICE is a local, model-agnostic interpretation method that, while similar to PDP, plots the contribution for each individual instead of the average contribution. However, ICE shares PDP's limitation of struggling with models that have more than two features; as the number of features increases, ICE explanations become difficult to interpret (Adadi and Berrada 2018). Given that our work deals with image datasets, limiting evaluations to a small number of features is not feasible, leading us to discard both PDP, ALE global interpretable methods and the ICE local interpretable method. Next, LIME, as its name suggests, is a local, model-agnostic interpretation method. According to Ribeiro et al. (2016), LIME works by generating a new dataset consisting of perturbed samples and the corresponding predictions of the underlying model for a new observation. An interpretable model is then fitted to this new dataset, weighted by the proximity of the perturbed samples to the original observation. In the case of image classification tasks, LIME identifies the region of an image most strongly associated with a prediction label. SHAP, another local model-agnostic interpretation method, is based on the Shapley value (Lundberg and Lee 2017). Several studies (Štrumbelj and Kononenko 2014; Shrikumar et al. 2019; Datta et al. 2016; Bach et al. 2015; Lipovetsky and Conklin 2001; Lundberg and Lee 2017) have shown that the Shapley value, derived from game theory, explains the prediction by distributing the "gains" (prediction shift from a baseline) among the input features, showing how each feature contributes positively or negatively to the prediction.

In prior research, LIME and SHAP have demonstrated remarkable effectiveness in explaining classification tasks. For instance, Lin et al. (2021) show that LIME effectively highlights patterns in virus datasets and outperforms six other XAI methods by accurately identifying trigger regions. Similarly, Dieber and Kirrane (2020) show LIME's potential for enhancing the interpretability of tabular machine learning models, as demonstrated through performance assessments and user studies. Furthermore, Jeyakumar et al. (2020) suggest that LIME was preferred for text sentiment classification tasks due to its annotation-based explanations. SHAP, on the other hand, has excelled in explaining models for image and text classification tasks. An empirical evaluation (Hailemariam et al. 2020) comparing LIME and SHAP revealed that SHAP slightly outperforms LIME in terms of identity, stability, and separability across deep neural network models on image classification datasets. Several studies have utilized LIME and SHAP together to interpret black-box models for various machine learning tasks, including image classification (Gaspar et al. (2024); Alabi et al. (2023); Ahmed et al.

(2024); Vimbi et al. (2024); Panati et al. (2022)). Therefore, in this study, we decided to leverage LIME and SHAP-based explainability methods to evaluate the explainability of our quantum-supervised contrastive learning model for image classification tasks.

## 3 Methodology

The *QRLaXAI* method's explainable QRL model consists of a feature extractor and a classifier. The feature extractor is a quantum autoencoder trained with a supervised contrastive learning loss function, while the classifier is a VQC designed for multi-class tasks and optimized using the cross-entropy loss function. After training the feature extractor on image data, its output is used to train the VQC. As shown in Fig. 2, the trained QRL model generates model-agnostic LIME and SHAP-based visual and analytical explanations, including a feature analysis based on its predictions for image data across each dataset.

### 3.1 Quantum model

**Feature extractor** The explainable QRL model consists of two main components: the feature extractor and the classifier. The feature extractor employs a quantum autoencoder architecture (Romero et al. 2017), with a quantum data augmentation module (Chalumuri et al. 2022) as its head and a quantum projection head (Khosla et al. 2020) as its tail. This setup enables the transformation of classical image data into quantum data, leveraging quantum gates and entanglements to capture complex features that classical methods may overlook. The output is a reduced-dimensional feature vector representing quantum-transformed information trained using the supervised contrastive loss function.

The quantum data augmentation module, serving as the head of the feature extractor (see Eq. 9), uses 6 qubits-matching the number of latent qubits in the quantum autoencoder-to encode pixel values through $RZ(\theta)$ rotation gates, where $\theta$ corresponds to pixel values. Initially, all qubits are set to the $|0\rangle$ state, and the $RZ(\theta)$ gates adjust the qubits based on the image data. By encoding image data into quantum states, this module initiates the quantum processing pipeline, which distinguishes this method from purely classical approaches.

The quantum encoder processes these quantum states through multiple layers, applying quantum gates like $RY(\theta)$ and $CX$, which introduce entanglement and non-linearity to the feature space. This operation results in compressed quantum feature representations, highlighting the quantum system's ability to handle high-dimensional data efficiently. It begins with an input layer containing an 8-qubit quantum state, which includes the quantum state, a reference state, an auxiliary qubit, and a classical register for measurement. The next layer is a parameterized circuit (see Eq. 11), which applies alternating $RY(\theta)$ rotations and $CX$ entanglements. In the bottleneck layer, 2 qubits initialized in the $|0\rangle$ state, known as "trash states," are excluded, representing successful compression of the input state. The remaining 6 latent qubits capture the core features of the input.

The quantum data augmentation circuit transforms images, which are then fed into the quantum encoder for further processing. At the end of the quantum autoencoder, the projection head—a simple fully connected quantum neural network layer with 64 units—generates the feature vector. Unlike previous studies (Khosla et al. 2020; Don and Khalil 2024) that use more complex configurations, we retain the projection head during classifier training to reduce computational complexity when generating feature vectors. The quantum autoencoder, data augmentation module, and pro-

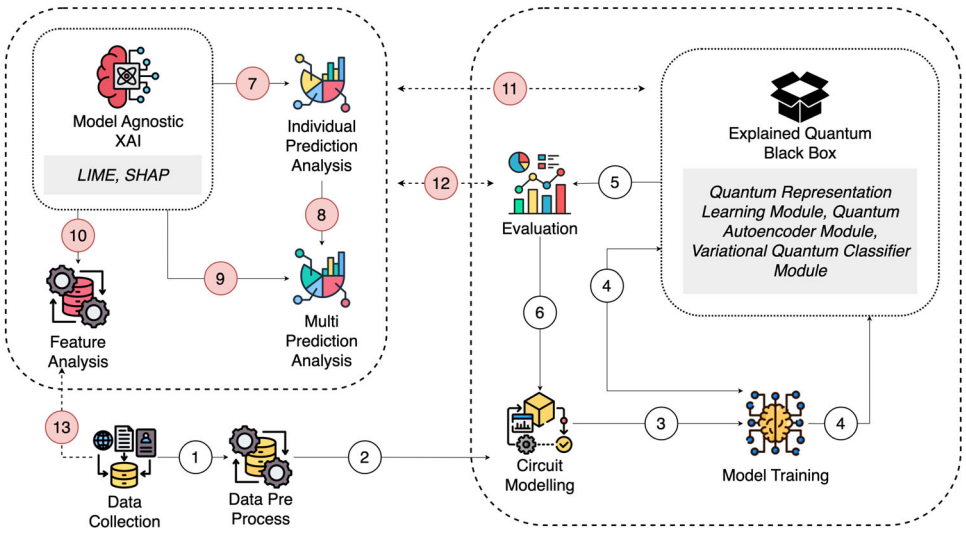


**Fig. 2** *QRLaXAI* explainability framework: Steps (7), (8), and (9) generate model-agnostic LIME and SHAP explanations, including individual and multi-prediction explanations. Step (10) involves feature analysis, using the original data from step (13) and QRL model predictions from step (11). Step (12) fine-tunes the QRL model based on the explanation results and re-evaluates model explanations using the fine-tuned QRL model

jection head are all trained using the supervised contrastive loss function (Khosla et al. 2020), which encourages similar data points to cluster in the representation space while separating dissimilar points, thus enhancing classification performance.

The supervised contrastive loss function is defined as follows:

$$L_S(\omega) = -\sum_{\beta \in K} \log \left( \frac{\exp(Z_i \cdot Z_j / T)}{\sum_{n \in S} \exp(Z_i \cdot Z_n / T)} \right) \tag{1}$$

where $Z_i$ and $Z_j$ are the encoded states in the feature space, $T$ is the temperature scaling parameter, and the loss is computed over positive pairs $\beta$ compared to all instances $S$. The objective is to minimize this loss, bringing positive pairs closer in the feature space while pushing negative pairs further apart.

The optimization process uses the COBYLA optimizer (IBM 2024), a gradient-free algorithm that iteratively refines linear approximations of the objective function, making it suitable for managing complex constraints and tuning quantum parameters.

**Classifier** The second component of the QRL model is the VQC (Farhi and Neven 2018; Schuld et al. 2020), which classifies the features extracted by the feature extractor. By converting quantum-transformed feature vectors into a probability distribution over classes, the VQC directly integrates quantum processing into the classification task. The VQC used in this study employs the Special Unitary Group Degree 2 (SU2) variational circuit (see Eq. 12), a hardware-efficient design with 6 qubits that processes the 64 features produced by the feature extractor.

The VQC is trained using the cross-entropy loss function and optimized with the COBYLA algorithm. The cross-entropy loss function is defined as follows:

$$L_C(\omega) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} p_{ij} \log q_{ij} \tag{2}$$

where $p_{ij}$ represents the true label for sample $i$ in class $j$, $q_{ij}$ is the VQC output for sample $i$ in class $j$, and $N = 50$ and $C = 10$ represent the number of samples and classes, respectively.

## 3.2 Explainability framework

### 3.2.1 LIME breakdown

In this study, we apply the LIME (Ribeiro et al. 2016) method to interpret the QRL model's predictions on individual images. LIME provides local explanations for complex, black-box models by approximating them with interpretable models around specific predictions. As shown in Algorithm 1, given an image $x \in \mathbb{R}^{n \times m}$, where $n \times m$ represents the image's pixel dimensions, LIME generates an explanation by creating perturbed versions of the input image and fitting a locally weighted linear model to approximate the classifier's predictions near the original image.

**Algorithm 1** LIME (Ribeiro et al. 2016).

**function** LIME($f$, $x$, $D$, $K$)
  **Input:**
    $f$: Classifier model
    $x$: Data point (image) to be explained
    $D$: Distribution from which to sample perturbations
    $K$: Number of perturbed samples to generate
  **Output:** Surrogate model $\xi(x)$
  $Z \leftarrow$ Sample $K$ points from $D$ ▷ Perturbed samples generated via binary masking
  **for** $i = 1 \rightarrow K$ **do**
    Generate a binary mask $z_i \in \{0, 1\}^{n \times m}$ ▷ Random pixel removal for perturbation
    Generate perturbed image $x_i' = x \odot z_i$ ▷ Apply the mask
    Compute prediction $g_i = f(x_i')$ ▷ Prediction of $f$ on perturbed sample $x_i'$
  **end for**
  **Compute Locality and Fit Surrogate Model:**
  **for** each perturbed sample $z_i$ **do**
    Compute proximity weight $\pi_x(z_i) = \exp\left(-\frac{d(x,z_i)^2}{\sigma^2}\right)$ ▷ Gaussian kernel for locality with cosine similarity
  **end for**
  Fit a weighted linear regression model $\xi(x)$ using $(z_i, g_i)$ pairs, weighted by $\pi_x(z_i)$ ▷ Locally weighted linear model
  **return** $\xi(x)$
**end function**

The main steps of the LIME methodology are as follows:

**Image segmentation** The first step involves segmenting the image into meaningful regions using the QuickShift algorithm (Vedaldi and Soatto 2008), an edge-preserving image segmentation technique that groups pixels based on color and spatial proximity. QuickShift ensures that perturbations affect coherent regions within the image, enhancing the interpretability of the resulting explanations.

**Perturbation of input image** In this step, we generate a set of perturbed samples by applying random binary masking to the images. For each pixel in the image, a binary mask $z \in \{0, 1\}^{n \times m}$ is created, where a value of 0 masks the pixel and a value of 1 retains it. Applying these masks to the original image $x$ produces a set of perturbed images $X'$, where only a subset of pixels is preserved. This random pixel removal serves as the perturbation strategy (see Eq. 3).

$$X' = \{x \odot z_i : z_i \in \{0, 1\}^{n \times m}, i = 1, \ldots, k\} \tag{3}$$

where $k$ is the number of perturbed samples, and $\odot$ represents element-wise multiplication.

**Prediction on perturbed samples** The classifier $f(x)$ (see Eq. 4) is then applied to each perturbed sample $x' \in X'$, generating a set of predictions. We use the QRL model to compute these predictions, denoted as follows:

$$f(x') = \{f(x'_i) : x'_i \in X', i = 1, \ldots, k\} \tag{4}$$

These predictions provide the basis for fitting an interpretable model.

**Locally weighted linear model** To approximate the behavior of the black-box model $f$ near the original image $x$, we fit a weighted linear regression model $g(z)$ to the predictions. This model is trained on the binary masks $z$ and their corresponding predictions $f(x')$.

The weight of each sample depends on its proximity to the original image, computed using a Gaussian similarity kernel (see Eq. 5). The proximity between a perturbed image and the original image is calculated using cosine similarity, which measures the angle between two non-zero vectors:

$$K(x, x') = \exp\left(-\frac{d(x, x')^2}{\sigma^2}\right) \tag{5}$$

where $d(x, x')$ represents the cosine similarity distance between the original image $x$ and the perturbed sample $x'$, and $\sigma$ is a scaling factor controlling the width of the kernel.

**Explanation generation** The explanation is derived by interpreting the coefficients of the linear model $g(z)$ (see Eq. 6). The magnitude and sign of each coefficient indicate the contribution of each pixel to the classifier's prediction. The weighted linear regression coefficients $\beta_i$ reflect pixel importance:

$$g(z) = \beta_0 + \sum_{i=1}^{n \times m} \beta_i z_i \tag{6}$$

where $\beta_i$ represents the importance of pixel $i$ in the explanation, and $\beta_0$ is the bias term. Pixels with the highest absolute $\beta_i$ values are identified as the most influential in the classifier's decision. By analyzing the linear approximation $g(z)$, we generate a visual explanation highlighting the pixels that most influence the model's prediction. In this study, we select 50 representative samples from each dataset and generate 150 perturbations per sample to compute the visual LIME explanation for each QRL model prediction, as outlined in QRL-LIME (Algorithm 2). This process involves 150 evaluations per sample, providing a clear visual representation of feature importance.

**Algorithm 2** QRL-LIME.

**function** QRL- LIME($f_{\text{QRL}}$, $x$, $D$, $K = 150$)
  **Input:**
    $f_{\text{QRL}}$: Quantum Representation Learning model
    $x$: Data point to be explained
    $D$: Distribution from which to sample perturbations
    $K$: Number of perturbed samples to generate ($K = 150$)
  **Output:** List $\Gamma$ of surrogate models
  $\Gamma \leftarrow$ Empty list to store surrogate models
  **for** $i = 1$ to $K$ **do**
    $D_i \leftarrow$ Generate perturbed quantum data locally around $x$
    $\gamma_i \leftarrow$ LIME($f_{\text{QRL}}$, $x$, $D_i$, $K$) ▷ Apply LIME to generate a surrogate model (see Alg. 1)
    Add $\gamma_i$ to $\Gamma$
  **end for**
  **return** $\Gamma$
**end function**

It is important to note that LIME provides local explanations, meaning the explanation is valid only in the vicinity of the specific instance analyzed. The accuracy of the explanation depends on the number of perturbed samples $k$ and the choice of distance metric in the similarity kernel, both of which warrant further investigation in future studies.

### 3.2.2 Shapley values

In this section, we extend the SHAP method, based on Shapley values (SVs), to the QRL model. SVs are commonly used to assess the importance of each feature in a model's prediction by treating individual features as players in a coalition game (Hart 1989), with the model's output serving as the value function that quantifies each feature's impact.

**Features as players and value function** Given an image $x \in \mathbb{R}^N$, where $N$ is the number of features (or dimensions), each feature $i \in \{1, \ldots, N\}$ is treated as a player in the coalition game. The value function $v(S)$ (see Eq. 7), which evaluates the importance of a coalition of features $S \subseteq \{1, \ldots, N\}$, is determined by the output of the classification model $f$ based on these features:

$$v(S) = v(S; x, f) \tag{7}$$

where $f$ is the classification model of interest. This formulation enables the calculation of each feature's contribution by assessing how the model's output changes when the feature is added or removed from the coalition.

**Computing Shapley values** The Shapley value $\phi_i$ (see Eq. 8) for each feature $i \in \{1, \ldots, N\}$ is computed by averaging the marginal contributions of the feature across all possible coalitions $S$ that exclude feature $i$. The Shapley value for

feature $i$ is defined as follows:

$$\phi_i = \sum_{S \subseteq \{1,\ldots,N\} \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} [v(S \cup \{i\}) - v(S)] \tag{8}$$

where $|S|$ represents the size of the coalition $S$, and $N!$ is the total number of coalitions. This equation ensures that the Shapley values fairly distribute the model's output contribution among all features.

**Approximating Shapley values** Calculating exact Shapley values is computationally expensive due to the factorial number of coalitions that require evaluation. To address this, we use the Kernel SHAP approximation (Covert and Lee 2021), which applies a locally linear surrogate model to approximate the value function, significantly reducing computational complexity. The expectation $E[f(X)|x_S]$, where $X$ represents all features not included in coalition $S$, is estimated by sampling from the dataset. In this study, similar to the LIME approach, we select 50 samples from each dataset to represent feature values and generate 500 perturbation samples to estimate the Shapley values for each prediction. This results in 500 times 50 model evaluations.

### 3.3 Dataset selection and preparation

**Selection** This study utilizes four datasets: MNIST (Deng 2012), FMNIST (Xiao et al. 2017), KMNIST (Clanuwat et al. 2018), and CIFAR10 (Krizhevsky 2009), each containing ten classes. Due to quantum hardware constraints, only a subset of classes and samples from each dataset is selected for experimentation. Additionally, images are resized to a $16 \times 16$ pixel resolution to fit these limitations. A random sampling method is used to select classes and data samples.

- **Class selection:** Each dataset includes 10 classes. During the random selection process, $n$ classes (where $1 < n < 10$) are initially selected for the experiments, with this procedure repeated five times. In each iteration, previously selected classes are excluded to allow new selections. Due to quantum hardware constraints, we do not evaluate all possible class combinations to prevent bias.
- **Data sample selection:** MNIST, FMNIST, and CIFAR10 contain 60,000 samples each, while KMNIST has 70,000 samples. Given the high cost and inefficiency of processing these volumes on quantum hardware, we select a smaller, manageable set of samples for experimentation, ensuring class balance.

**Preparation** To train the QRL model, we create incremental data batches, each containing 50 samples from the relevant classes. Starting with two classes, we add one class at a time until reaching 10 classes, resulting in a total of 9 data batches for each dataset. Our primary evaluations focus on the QRL model trained with the first two-class data batch in each dataset. Specifically, the first two classes are digits 0 and 1 for MNIST, T-shirt/top and Trouser for FMNIST, labels 0 and 1 for KMNIST, and airplane and automobile for CIFAR10. For simplicity, we refer to these two classes as "0" and "1" throughout the study. During the multi-class experiments, we evaluate all 9 QRL models trained across the 10 classes in each dataset.

### 3.4 Experimental setup

To ensure reliable performance, the experimental setup was configured to include both classical counterparts and quantum simulators. These systems were equipped with a 16 GiB NVIDIA T4 Tensor Core GPU to support machine learning inference and graphics-intensive tasks effectively.

For QRL model training, the Amazon Braket IonQ Harmony device (Amazon 2024; IonQ 2024) was utilized. This quantum processor, comprising 11 qubits, operates on a universal gate-model ion-trap architecture. The Universal Gate Set includes quantum gates capable of approximating any unitary transformation to a desired precision. The set consists of Pauli-X, Y, and Z gates, along with the CNOT gate, making it suitable for constructing any quantum circuit. In the ion-trap architecture, qubits are formed using the electronic states of confined ions, which interact via the Coulomb force (Kielpinski et al. 2002). These ions, isolated and trapped to preserve quantum coherence, encode qubit states in their electronic or hyperfine energy levels. Laser beams manipulate the internal energy levels of the trapped ions, enabling single-qubit gate operations. The experiments were implemented and executed using the IBM Qiskit library (Treinish 2023). Access to the IonQ Harmony device was facilitated via the Amazon Braket Provider (Amazon 2024), which allowed the conversion of IBM Qiskit circuits for compatibility with the Harmony platform.

### 3.5 Code and dataset availability

The code used in this study, along with detailed instructions for reproducing the experiments, is available at the following repository: https://github.com/AsithaIndrajith/qrlaxai. All datasets used in this study—MNIST, FMNIST, KMNIST, and CIFAR10—are publicly available and can be accessed through their respective official sources. The repository also includes preprocessing scripts for preparing the datasets as

described in this paper. By providing access to the code and datasets, we aim to ensure transparency and facilitate further exploration and validation of our proposed framework.

## 3.6 Quantum gate explainability analysis

This section provides a theoretical explanation of the QRL model's feature extractor, detailing the structure and function of its quantum circuits and the roles of each gate within them.

### 3.6.1 Feature extractor

The feature extractor in the QRL model includes a quantum data augmentation circuit, a quantum autoencoder (QAE), and a projection head. Together, these components encode classical data into enhanced quantum representations, optimized for quantum-based classification.

**Data augmentation circuit** The data augmentation circuit applies sequences of $SX$ and $R_Z$ rotations to each qubit, entangled through $CNOT$ gates between adjacent qubits. Its structure is given by the following:

$$\mathrm{DA} = \left(\bigotimes_{i=1}^{n} SX \cdot R_Z\left(\frac{\pi}{2}\right) \cdot SX \cdot R_Z\left(\frac{\pi}{2}\right)\right) \cdot \left(\prod_{k=1}^{n-1} \mathrm{CNOT}(q_k, q_{k+1})\right) \tag{9}$$

The $SX$ gate, or the square root of the $X$ (Pauli-X) gate, is applied to each qubit $q_i$, placing it into a superposition and enabling each qubit to represent multiple values simultaneously. Each qubit then undergoes an $R_Z\left(\frac{\pi}{2}\right)$ rotation, which shifts its phase around the $Z$-axis by $\pi/2$, embedding additional information into the quantum state. By combining $SX$ and $R_Z$ operations, the circuit creates a complex representation of the input data. To introduce entanglement, $CNOT$ gates are applied between adjacent qubits $q_k$ (control) and $q_{k+1}$ (target), establishing dependencies that capture interaction patterns and enhance the data-augmented quantum state.

**Autoencoder circuit** The autoencoder circuit combines an amplitude encoding (AmpEnc) layer with a swap test to evaluate reconstruction quality. The circuit is structured as follows:

$$\mathrm{QAE} = (AE) \cdot \mathrm{Barrier} \cdot \left(H(q_{n+1}) \cdot \prod_{j=1}^{m} \mathrm{CSWAP}(q_{n+1}, q_j, q_{j+m}) \cdot H(q_{n+1})\right) \cdot \mathrm{M}(q_{n+1}) \tag{10}$$

In the amplitude encoding layer, each qubit $q_i$ undergoes a rotation around the $Y$-axis using the $R_Y(\theta_i)$ gate, parameterized by $\theta_i$. This transformation maps classical data onto the quantum state by modifying each qubit's probability amplitude, an essential step for encoding data into quantum form. Following this, $CNOT$ gates entangle adjacent qubits, creating correlations that capture complex data relationships.

A barrier separates the encoding layer from the swap test, maintaining a logical separation between the two stages.

The swap test begins with a Hadamard gate, $H(q_{n+1})$, applied to an auxiliary qubit $q_{n+1}$, placing it in superposition to enable controlled swap operations on two sets of qubits. Next, $CSWAP$ gates are applied between pairs of qubits $q_j$ and $q_{j+m}$, with $q_{n+1}$ serving as the control qubit. These gates swap the states of paired qubits only when the auxiliary qubit is in a particular state, allowing a comparison of two quantum states. After the $CSWAP$ operations, a second Hadamard gate, $H(q_{n+1})$, completes the swap test.

The circuit concludes with a measurement (M) on $q_{n+1}$, assessing the similarity between the quantum states generated by the encoding layer, effectively verifying the fidelity of the compressed quantum data.

The AmpEnc circuit (IBM 2024a) in the QAE applies parameterized $R_Y$ rotations and entangling $CNOT$ gates, as described by the following:

$$\mathrm{AmpEnc} = \left(\bigotimes_{i=1}^{n} R_Y(\theta_i)\right) \cdot \left(\prod_{k=1}^{n-1} \mathrm{CNOT}(q_k, q_{k+1})\right) \cdot \left(\bigotimes_{i=1}^{n} R_Y(\theta_{i+n})\right) \tag{11}$$

Each qubit is initialized in $|0\rangle$ and undergoes $R_Y$ rotations followed by $CNOT$ gates, transforming the input into an entangled quantum state. This step encodes classical data while preserving information critical for the autoencoder's compression.

### 3.6.2 Classifier

The VQC in the QRL model uses a SU2 variational circuit (IBM 2024b), which consists of multiple repeated layers. Each layer includes a combination of rotation gates and entangling gates, specifically $R_Y$, $R_Z$, and $CNOT$ gates. These layers enable the VQC to capture complex patterns in quantum states, making it suitable for classification tasks.

The SU2 variational circuit is structured as follows:

$$\mathrm{SU2} = \left(\bigotimes_{i=1}^{n} R_Y(\theta_i) R_Z(\theta_{i+n})\right) \cdot \left(\prod_{k=1}^{n-1} \mathrm{CNOT}(q_k, q_{k+1})\right) \tag{12}$$

The $R_Y$ and $R_Z$ gates are parameterized rotation gates applied to each qubit. The $R_Y(\theta_i)$ gate rotates the qubit state around the $Y$-axis by an angle specified by $\theta_i$, adjusting the quantum representation based on learnable parameters. Following this, the $R_Z(\theta_{i+n})$ gate rotates the qubit state around

the $Z$-axis, using parameters $\theta_{i+n}$. Together, these gates allow each qubit to explore a broader range of orientations on the Bloch sphere, enhancing the model's ability to capture complex data relationships.

After each qubit undergoes $R_Y$ and $R_Z$ rotations, $CNOT$ gates are applied between adjacent qubits, creating entanglement. A $CNOT$ gate is a two-qubit gate where the first qubit acts as the control and the second as the target. If the control qubit is in state $|1\rangle$, the $CNOT$ gate flips the target qubit's state; otherwise, the target remains unchanged. This entanglement establishes dependencies between the states of neighboring qubits, allowing the circuit to capture interactions between data features.

By repeating these layers, the SU2 circuit iteratively refines the quantum state, optimizing it for classification. The adjustable parameters $\theta_i$ are tuned through a training process to minimize classification errors, effectively enabling the circuit to learn from data and perform accurate predictions.

## 4 Experiments and evaluations

### 4.1 Evaluation on MNIST, KMNIST, and FMNIST

We performed three types of experiments on the MNIST, KMNIST, and FMNIST datasets. First, we generated LIME visual explanations to interpret prediction results. Second, we utilized SHAP to analyze prediction explanations across all three datasets. For these two experiments, the QRL model, trained with a standardized data preparation step (see Sect. 3.3), achieved accuracies of 90%, 90%, and 75% on MNIST, FMNIST, and KMNIST, respectively, with explanations provided by LIME and SHAP. Finally, we conducted ablation studies to evaluate the impact of various QRL model components on explainability, examining variations in the variational circuits, the number of quantum gates within these circuits, and the effect of varying class counts on the explanations generated by LIME and SHAP.

#### 4.1.1 LIME visual explanation

**Image segmentation for LIME explanation** The first step in generating LIME explanations involved segmenting the images into coherent regions to facilitate meaningful perturbations. This segmentation was performed using the quick-shift algorithm, an edge-preserving image segmentation technique that groups pixels based on color and spatial proximity. By applying quick-shift, we ensured each segmented region represented a visually consistent part of the image, enhancing the interpretability of the perturbations applied in subsequent steps.

Figures 3, 4, and 5 show two correctly predicted sample images from the 0th and 1st classes in each dataset, illustrating comparisons among the original images, generated superpixels, and segmented images with highlighted boundaries for the QRL model. For each sample, the original image is shown in the first column, the superpixel representation generated by quick-shift appears in the second column, and the segmented image with highlighted boundaries is in the third column. This segmentation approach was applied to 50

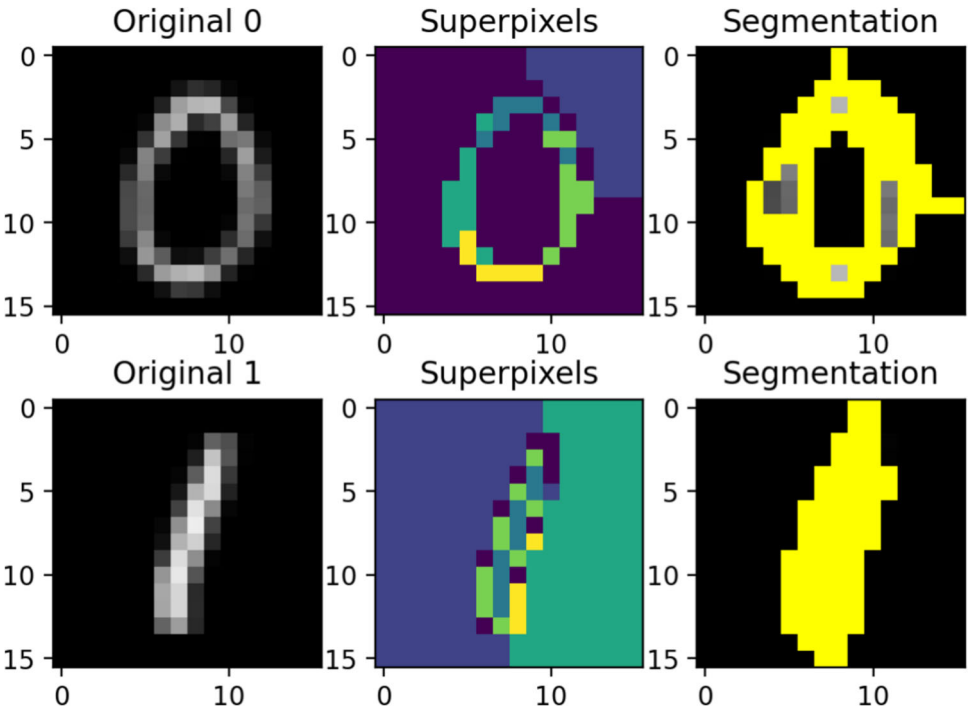


**Fig. 3** MNIST image segmentation: The first column shows the original MNIST image, the second displays the quick-shift-generated superpixel representation, and the third presents the segmented image with highlighted boundaries

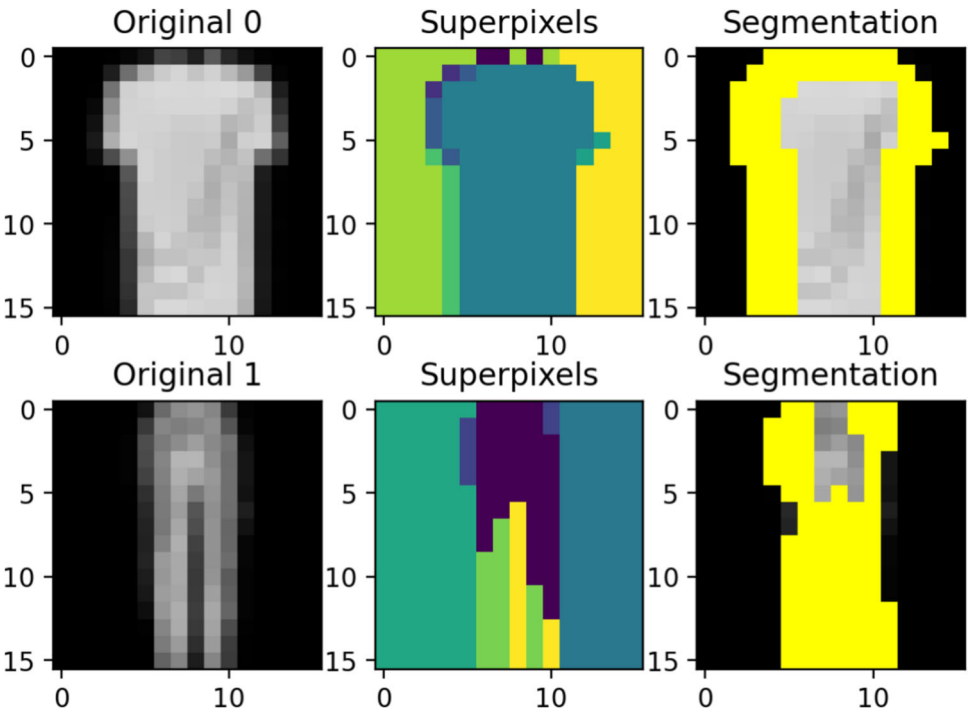


**Fig. 4** FMNIST image segmentation: The first column shows the original FMNIST image, the second displays the quick-shift-generated superpixel representation, and the third presents the segmented image with highlighted boundaries

images in each dataset to evaluate the stability and effectiveness of superpixel generation across a variety of samples.

These initial segmentation results confirm that quick-shift successfully identified distinct, coherent regions within each image, allowing the LIME methodology to apply perturbations while maintaining the contextual integrity of each segmented area. This segmentation is crucial for ensuring that the explanation generated by LIME is both interpretable

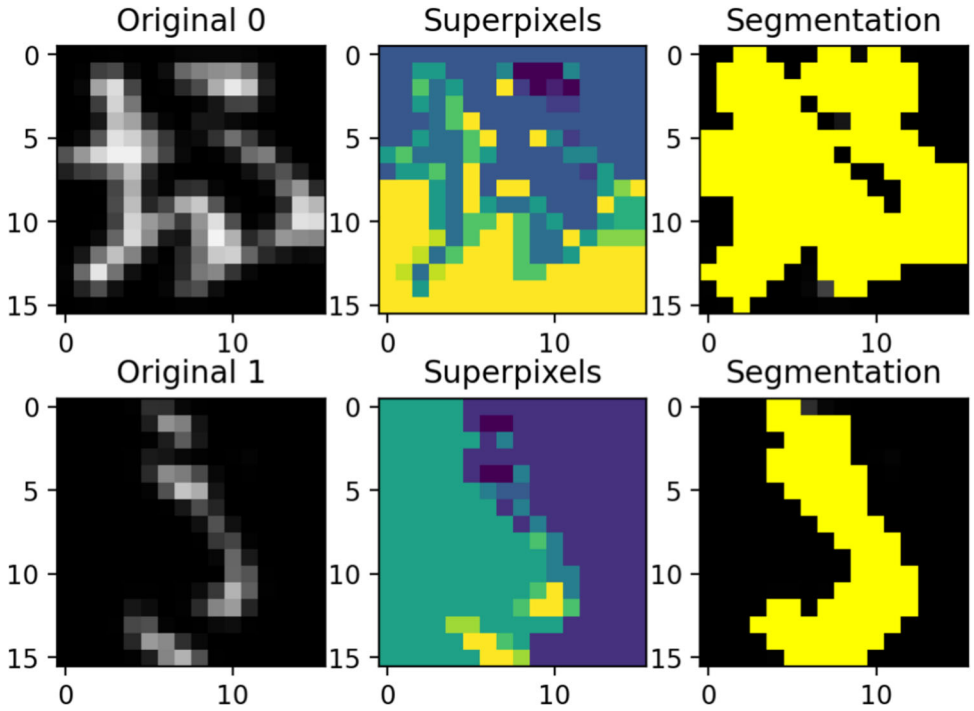


**Fig. 5** KMNIST image segmentation: The first column shows the original KMNIST image, the second displays the quick-shift-generated superpixel representation, and the third presents the segmented image with highlighted boundaries

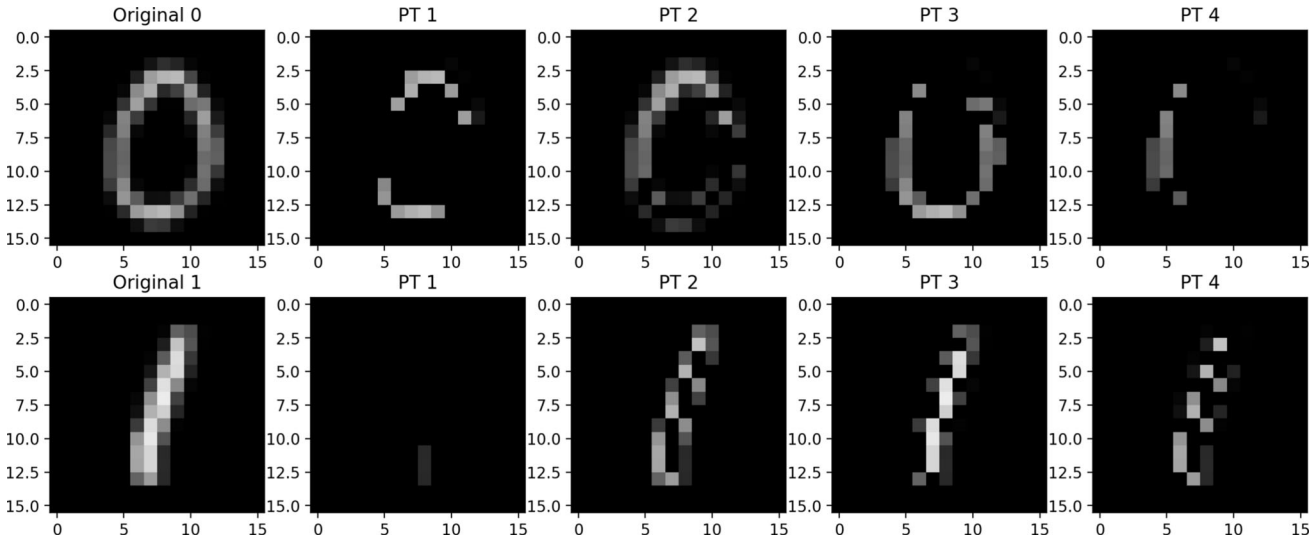


**Fig. 6** MNIST perturbations: The first column shows the original MNIST image, followed by four perturbations

and visually consistent, as perturbations are applied to well-defined areas of the image rather than individual pixels.

**Perturbation of input image for LIME explanation** The second step in generating LIME explanations involved creating a set of perturbed images to facilitate local explanations of model predictions. To achieve this, we applied random binary masking to each image, generating distinct perturbations by selectively masking or retaining certain pixel regions. For each pixel in the image, a binary mask was created, where a value of 0 masked the pixel and a value of 1 retained it. This approach produced perturbed images in which only a subset of pixels is preserved, effectively simulating various occlusion patterns across the image.

In Figs. 6, 7, and 8, we present the same two sample images from each of the three datasets, showing the original image alongside four representative perturbations generated by this method. For each sample image, the original image is shown in the first column, followed by four perturbations in the second through fifth columns. The perturbations illustrate different subsets of visible superpixels, achieved through random binary masking. This random pixel removal constitutes the perturbation strategy, enabling the model to focus on specific image segments while disregarding others.

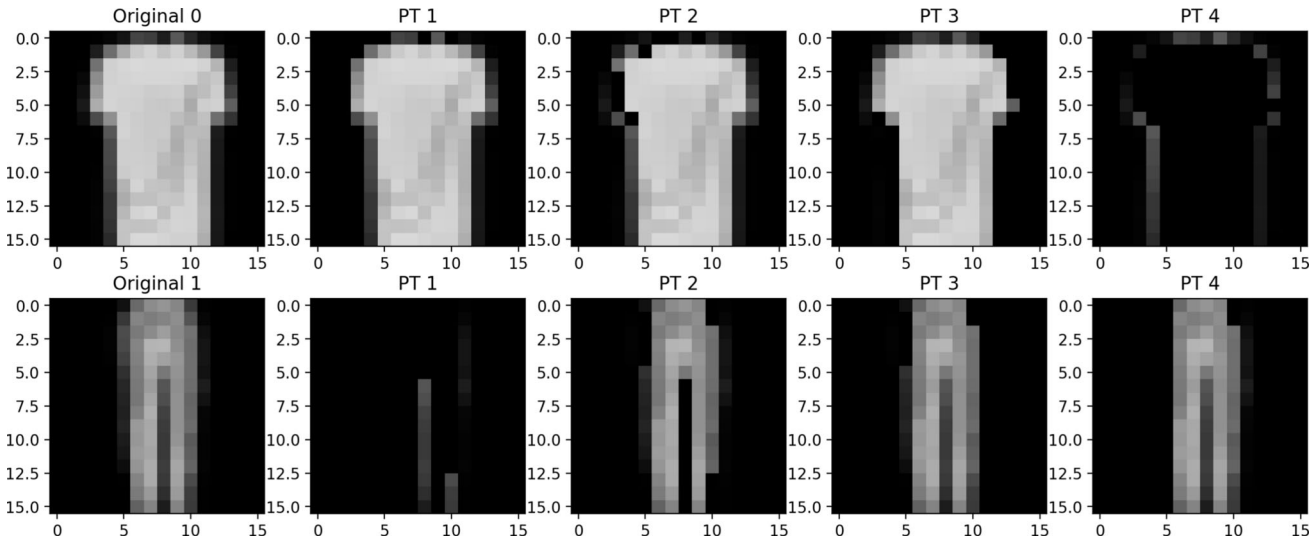


**Fig. 7** FMNIST perturbations: The first column shows the original FMNIST image, followed by four perturbations

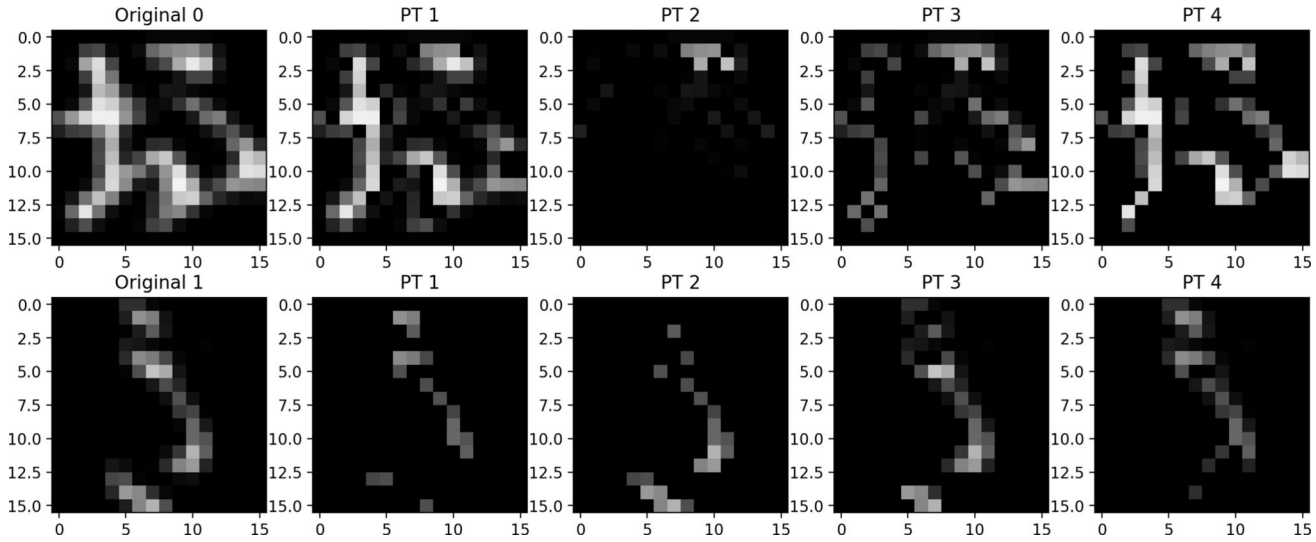


**Fig. 8** KMNIST perturbations: The first column shows the original KMNIST image, followed by four perturbations

During this experiment, 150 perturbations were generated for each sample, though only four are displayed here to provide a visual example.

**Key feature identification in LIME explanations** The third step in the LIME explanation process involved using a locally weighted linear regression model to approximate the behavior of the QRL model around the original image. Each perturbed sample was weighted based on its proximity to the original image using a Gaussian similarity kernel, allowing the model to emphasize features closest to the original image.

Since the model achieved high accuracy on MNIST, FMNIST, and KMNIST, the LIME results indicate that the QRL model effectively learned accurate feature representations for MNIST and FMNIST images. The selected superpixel regions consistently align with visually significant areas in the original images, showing the model's ability to focus on key features contributing to its predictions. However, for KMNIST, the top superpixel regions cover fewer features than in the originals, aligning with the slightly lower accuracy achieved on this dataset. Figure 9 displays the top four superpixel regions identified by LIME across all three datasets for the same two sample images, highlighting the QRL model's robust feature-learning capability. The highlighted regions offer intuitive insights into how specific image areas influence classification decisions.

### 4.1.2 Shapley values explanation

In this experiment, we extended the SHAP methodology to interpret the QRL model's predictions, using Shapley val-

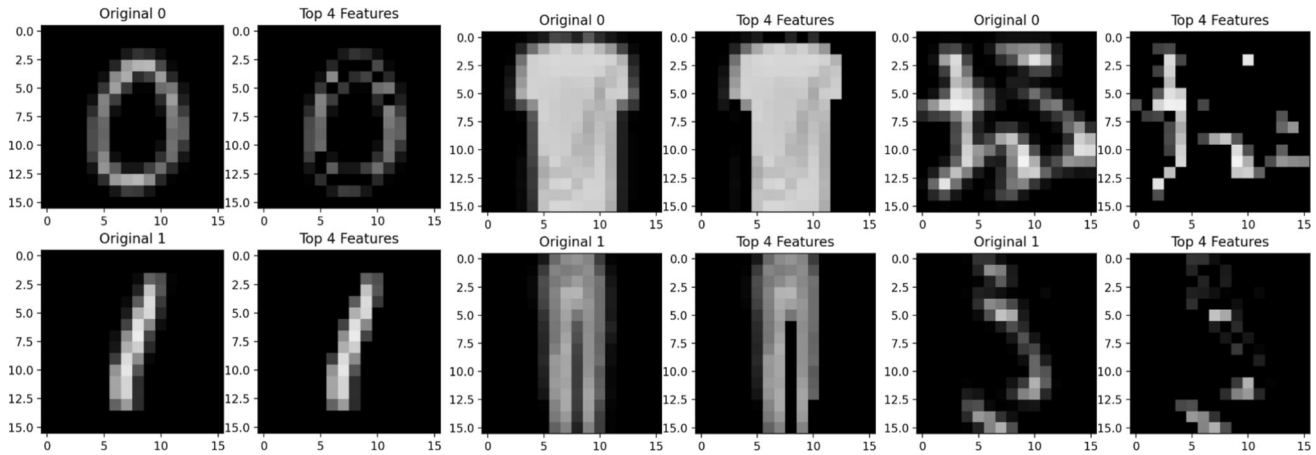


**Fig. 9** MNIST, FMNIST, and KMNIST top 4 features: Displays the original image with the top four superpixel regions identified by LIME for all 3 datasets

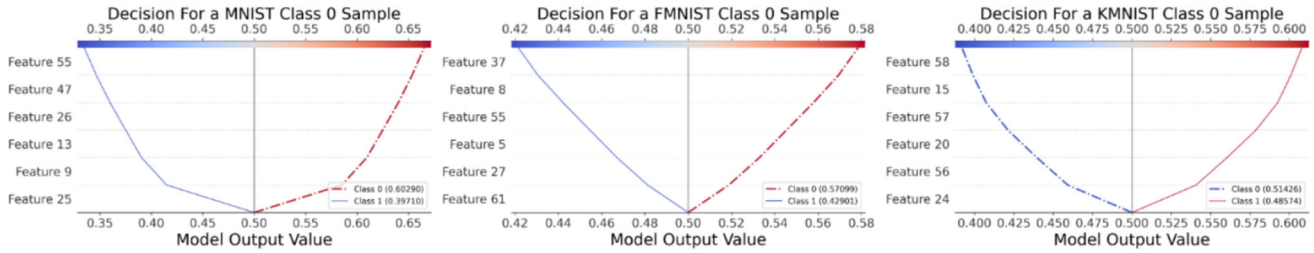


**Fig. 10** Individual SHAP Explanations for MNIST, FMNIST, and KMNIST: the magnitude of each top 5 feature's effect on the class 0 sample prediction in each dataset

ues (SVs) to quantify the contribution of individual features to model outputs, as detailed in Sect. 3.2.2. SHAP values reveal how each feature influences the model's predictions, with the model's output acting as the value function that represents the combined impact of these features. After calculating the SHAP values using KernelSHAP, we visualized the results for individual and multiple predictions to understand the QRL model's decision-making process across the three datasets.

**Individual prediction explanations** For individual samples, we utilized a multi-output decision plot to illustrate the impact of each feature on the model's prediction for a single instance. As shown in Fig. 10, this representation visualizes the SHAP values across multiple classes, indicating how each feature influences the model's output toward or away from each class. The plot structure reveals both the magnitude and direction of each feature's effect on the prediction, facilitating the identification of features that contribute positively or negatively to specific class predictions.

Following a similar approach to the LIME explanations, we selected the same two correctly predicted sample images from the 0th and 1st classes in each dataset and visualized the top 5 features identified using hierarchical clustering based on SHAP values. Figure 10 displays the effect magnitude of each top 5 feature on the class 0 sample prediction across the datasets. The plots illustrate the QRL model's confidence in predicting each sample's class, with the deviation of each prediction from the base value representing the model's confidence level.

**Multi-prediction explanations** We also produced plots for multiple predictions to provide a broader view of how feature contributions vary across different samples, encompassing both correct and incorrect classifications. This representation plots model output values for different samples against feature contributions, offering a comparative perspective on feature influence across samples. The multi-prediction plot uses a distinct ordering along the feature contribution axis to emphasize different aspects of feature importance. It clusters samples by similarity in feature contributions, allowing for the identification of groups where the model exhibits consistent behavior.

We generated SHAP values for 50 samples. Figure 11 illustrates the effect magnitude of each top 5 feature on predictions for 10 samples in each dataset, with dashed lines indicating misclassifications. Although the QRL model achieved high accuracy across all three datasets (see Sect. 4.1), these plots suggest that the model's confidence level is generally moderate, with predictions deviating around the SHAP base value. Additionally, several misclassifications in the MNIST and FMNIST plots show high confidence lev-

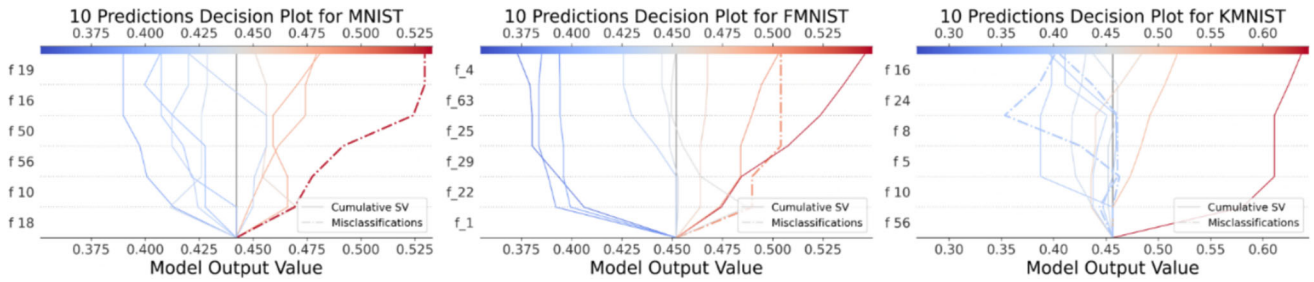


**Fig. 11** Multiple SHAP explanations for MNIST, FMNIST, and KMNIST: the magnitude of each top 5 feature's effect on predictions for 10 samples in each dataset, with misclassifications highlighted

**Table 1** Effect of rotational layers on QRL model accuracy: This demonstrates the impact of different numbers of rotational layers (RY-4 to RY-10) on the accuracy of the QRL model across three datasets: MNIST, FMNIST, and KMNIST

| Dataset | RY-4 | RY-5 | RY-6 | RY-7 | RY-8 | RY-9 | RY-10 |
|---|---|---|---|---|---|---|---|
| Accuracy with rotational layers (%) | | | | | | | |
| MNIST | 80.0 | 90.0 | 80.0 | 85.0 | 75.0 | 85.0 | 90.0 |
| FMNIST | 75.0 | 85.0 | 80.0 | 75.0 | 65.0 | 90.0 | 85.0 |
| KMNIST | 100.0 | 70.0 | 75.0 | 80.0 | 80.0 | 80.0 | 70.0 |

els, pointing to potential improvement areas. This analysis demonstrates that SHAP explanations offer deeper insights into the QRL model's predictions compared to LIME. Consequently, in future experiments, we will continue to utilize multi-prediction SHAP explanations alongside LIME visual explanations, as class-wise SHAP explanations for individual predictions provide limited analytical detail for binary classification problems, where only two classes are compared.

## 4.2 Ablation studies

In the ablation studies, we assessed how different components of the QRL model influence explainability, with a particular focus on the classifier component. We analyzed how variations in the variational circuits, the number of quantum gates within these circuits, and the impact of multi-class classification problems affected the explanations generated by LIME and SHAP.

### 4.2.1 Impact of rotational layers in the QRL model

The classifier, a variational quantum circuit (VQC) (see Sect. 3.1), uses a variational circuit that allows it to learn complex data patterns by iteratively adjusting its parameters, similar to the weight optimization process in classical neural networks. In this experiment, we investigated how increasing the number of quantum gates in the variational circuit-analogous to adding hidden layers in a deep neural network-impacts the accuracy of the QRL model. As detailed in Eq. 12, the SU2 variational circuit employed in this study consists of layers of $R_Y$ and $R_Z$ rotation gates on each qubit, followed by $CNOT$ gates that entangle adjacent qubits. The rotational gate configuration in the QRL model's classifier was initially repeated four times and then gradually increased to ten repetitions to evaluate its impact on performance.

Table 1 shows that increasing the repetitions of rotational gates does not significantly enhance the QRL model's accuracy across the three datasets. The LIME explanations demonstrate that increasing the number of rotational gates up to ten layers allowed the classifier to consistently identify the top 4 superpixel regions in the original images, similar to the results achieved with four layers, as shown in Fig. 12. SHAP results (see Fig. 13) reinforce this finding, showing that additional rotational gates do not boost the QRL model's confidence level in its predictions. Thus, both LIME and SHAP analyses suggest that adding more rotational gates to the QRL model's classifier does not substantially improve accuracy or confidence in correct predictions, though it does increase computational complexity.

### 4.2.2 Effect of variational circuit architecture

Since increasing the rotation layer repetitions ($R_Y$) in the QRL model's classifier showed no clear impact on accuracy, and the explainability studies produced similar explanations, we further analyzed the effect of different variational circuit architectures for the classifier. Specifically, we examined the SU2 (see Eq. 12) variational circuit architecture used throughout the study, along with the Amplitude encoding cir-

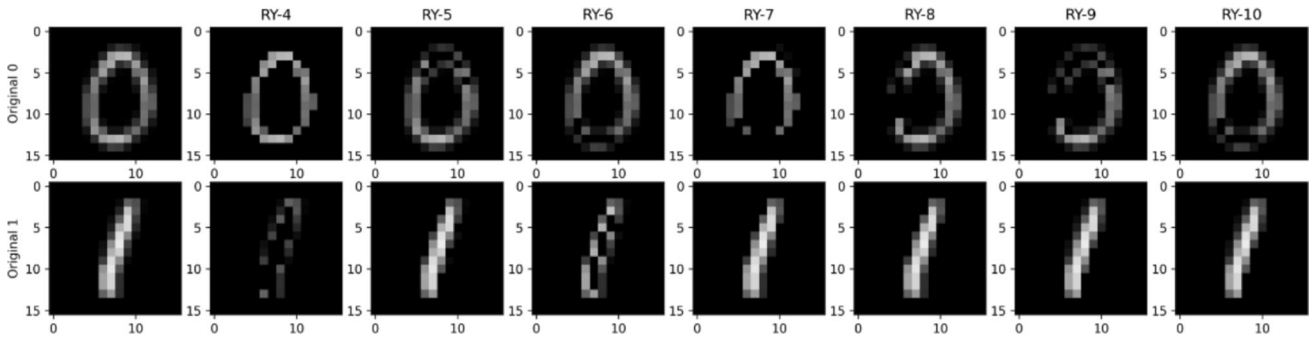


**Fig. 12** MNIST top 4 features with rotational layers: The first column displays the original MNIST image, while the remaining columns illustrate the behavior of the top 4 features as rotational layers are incrementally added in the classifier

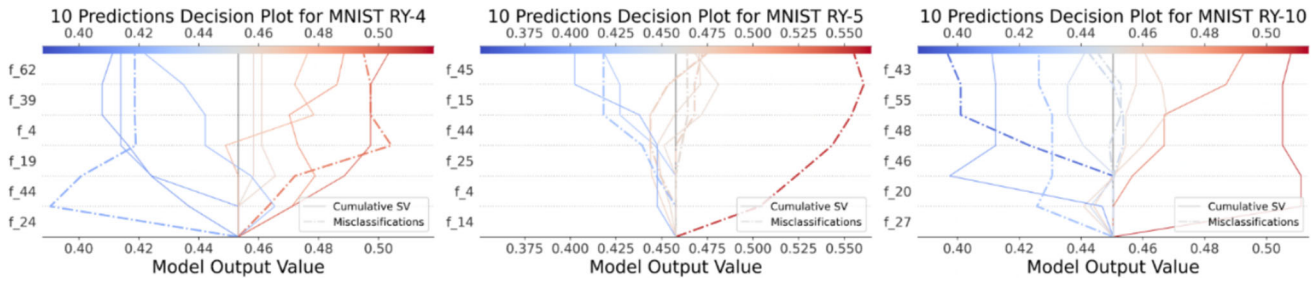


**Fig. 13** Multiple SHAP explanations for MNIST with rotational layers: This visualization shows the magnitude of each top 5 feature's effect on 10 predictions for rotational layer counts of 4, 5, and 10

cuit (see Eq. 11) and the local interactions circuit (TL) (IBM 2024).

The TL circuit (see Eq. 13) employs alternating layers of $R_Y$ and $R_Z$ rotations, interspersed with $CNOT$ entanglement between adjacent qubits in a linear configuration.

$$\mathrm{TL} = \left(\bigotimes_{i=1}^{n} R_Y(\theta_i) R_Z(\theta_{i+n})\right) \cdot \left(\prod_{k=1}^{n-1} \mathrm{CNOT}(q_k, q_{k+1})\right) \cdot \left(\bigotimes_{i=1}^{n} R_Y(\theta_{i+2n}) R_Z(\theta_{i+3n})\right) \tag{13}$$

The QRL model's classifiers, based on SU2, AE, and TL variational circuits, achieved the following accuracies: for MNIST, the SU2, AE, and TL circuits reached accuracies of 90%, 80%, and 95%, respectively, with TL exhibiting the highest performance. On FMNIST, the accuracies were 90% for SU2, 75% for AE, and 80% for TL, with SU2 performing best. Finally, for KMNIST, SU2 achieved an accuracy of 75%, outperforming AE and TL, which scored 60% and 55%, respectively.

While the QRL model demonstrated relatively high accuracies across all datasets, the results indicate that varying the circuit architecture (SU2, AE, TL) does not significantly improve the overall accuracy of the QRL model. This finding is consistent with the limited effect observed when increasing the number of rotation layers (see Sect. 4.2.1). This observation is further supported by the LIME and SHAP explanations, as shown in Fig. 14. The LIME explanations for MNIST revealed that the AE and TL circuits captured most of the top superpixel regions similarly to the SU2 variational circuit used in the study. SHAP explanations also

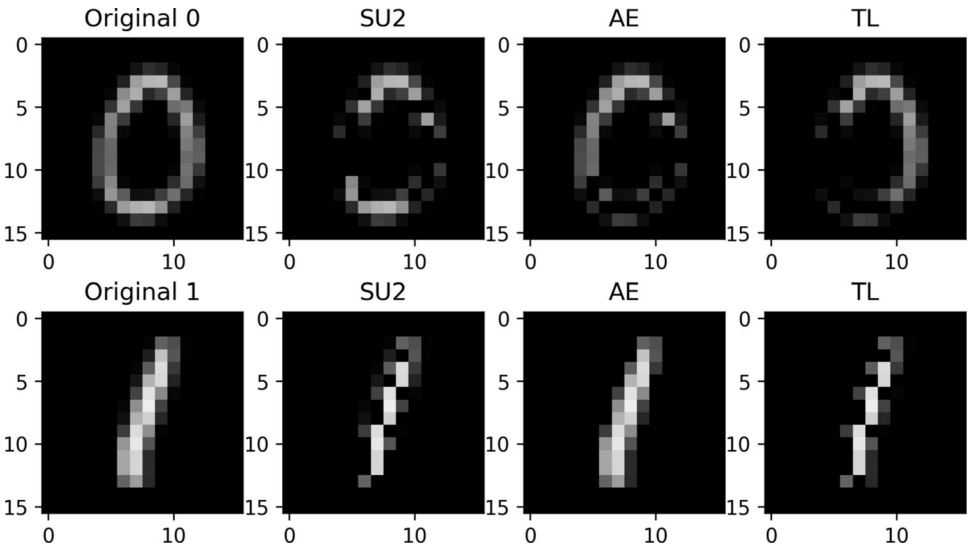


**Fig. 14** Top 4 features of MNIST with different variational circuit architectures: The first column shows the original MNIST image, while the remaining columns illustrate the behavior of the top 4 features for SU2, AE, and TL circuits integrated into the classifier

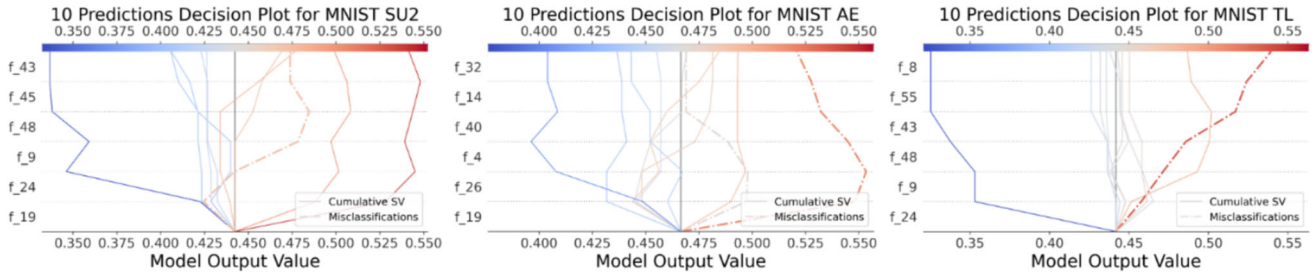


**Fig. 15** Multiple SHAP explanations for MNIST with different variational circuit architectures: This visualization shows the magnitude of each top 5 feature's impact on 10 predictions for SU2, AE, and TL circuits within the classifier

verified this result, although the AE and TL circuits exhibited higher-confidence misclassifications compared to the SU2 circuit (see Fig. 15).

#### 4.2.3 Multi-class classification

In addition to analyzing the effects of rotational layers on QRL model explainability (see Sect. 4.2.1) and the impact of variational circuit architecture (see Sect. 4.2.2), both of which focused on QRL model components in binary classification, we conducted experiments to assess the explainability of the QRL model in a multi-class setting. In this experiment, we incrementally trained the QRL model on 2 to 10 classes, following the order of the original class labels and using 50 samples per class across all three datasets.

As shown in Table 2, the accuracy of the QRL model begins to degrade after three classes. This reduction in accuracy results from the increased complexity of encoding multi-class data, which necessitates more advanced quantum state preparation and additional qubits within the VQC. Consequently, this added complexity can lead to greater inaccuracies due to the noise present in NISQ devices.

The LIME explanations reveal that as the QRL model's feature complexity increases with additional classes, the top superpixel regions identified by LIME progressively reduce in size. In Fig. 16, we observe a clear reduction in the intensity of the top 4 superpixel regions in class 1, a trend similarly observed in the other datasets.

In the SHAP explanations (see Fig. 17), the initial plot shows the QRL model's prediction confidence, with a distinct separation between correct predictions and the SHAP base value, indicating high confidence. In the subsequent plot, however, the SHAP values for predictions are closer to the SHAP base value, reflecting decreased confidence in correct predictions and an increase in misclassification instances. The final plot indicates that the QRL model often predicts misclassifications with higher confidence than correct classifications.

### 4.3 Evaluation on CIFAR10

We extended our experiments to assess the explainability of QRL models on a more complex dataset, CIFAR10. In this experiment, we conducted a binary classification task as outlined in Sect. 3.3. The QRL model achieved an accuracy of 65% on CIFAR10. We then applied both LIME and SHAP to interpret the trained QRL model.

In the LIME explanations, interpreting the top 4 superpixel regions identified proved challenging. Based on the results, it appears that the QRL model did not effectively capture the most important regions of the original image. In Fig. 18, despite applying the correct perturbations, LIME struggled to identify the key pixel regions using the QRL model.

The SHAP explanations, shown in Fig. 19, further illustrate this finding. The plots, which display SHAP values for 10, 20, and 50 predictions, indicate an increase in the

**Table 2** QRL model's accuracy across classes: This illustrates the accuracy of classification across increasing numbers of classes (2 to 10) for three datasets: MNIST, FMNIST, and KMNIST

| Dataset | 2-C | 3-C | 4-C | 5-C | 6-C | 7-C | 8-C | 9-C | 10-C |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy across classes 2 to 10 (%) | | | | | | | | | |
| MNIST | 90.0 | 67.0 | 43.0 | 30.0 | 23.0 | 19.0 | 25.0 | 21.0 | 19.0 |
| FMNIST | 90.0 | 57.0 | 30.0 | 18.0 | 17.0 | 13.0 | 13.0 | 16.0 | 13.0 |
| KMNIST | 75.0 | 43.0 | 23.0 | 18.0 | 15.0 | 21.0 | 15.0 | 10.0 | 15.0 |

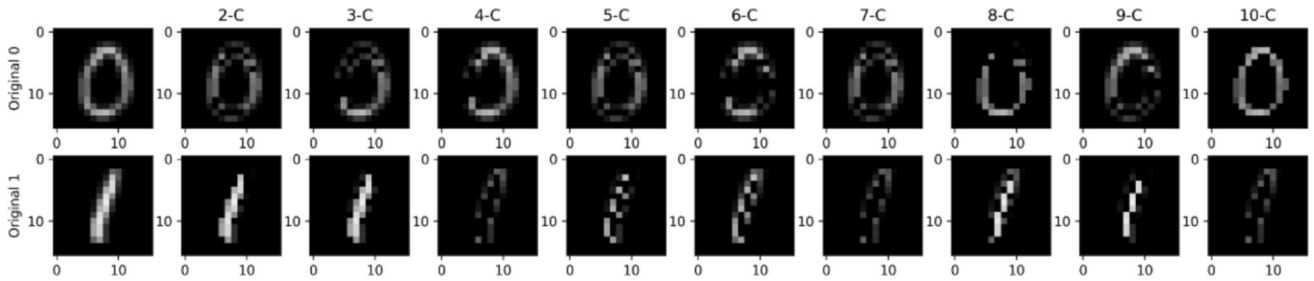


**Fig. 16** MNIST top 4 features for multi-class classification: The first column displays the original MNIST image, while the remaining columns show the behavior of the top 4 features across classes 2 to 10

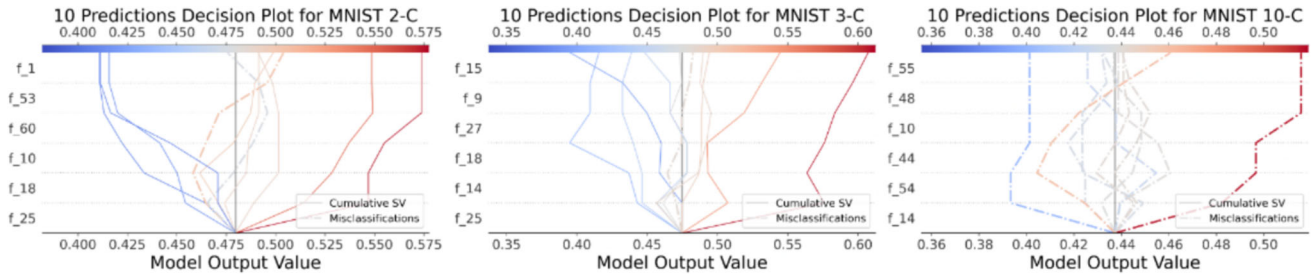


**Fig. 17** SHAP explanations for MNIST in multi-class classification: This visualization shows the impact of the top 5 features on predictions for classes 2, 3, and 10

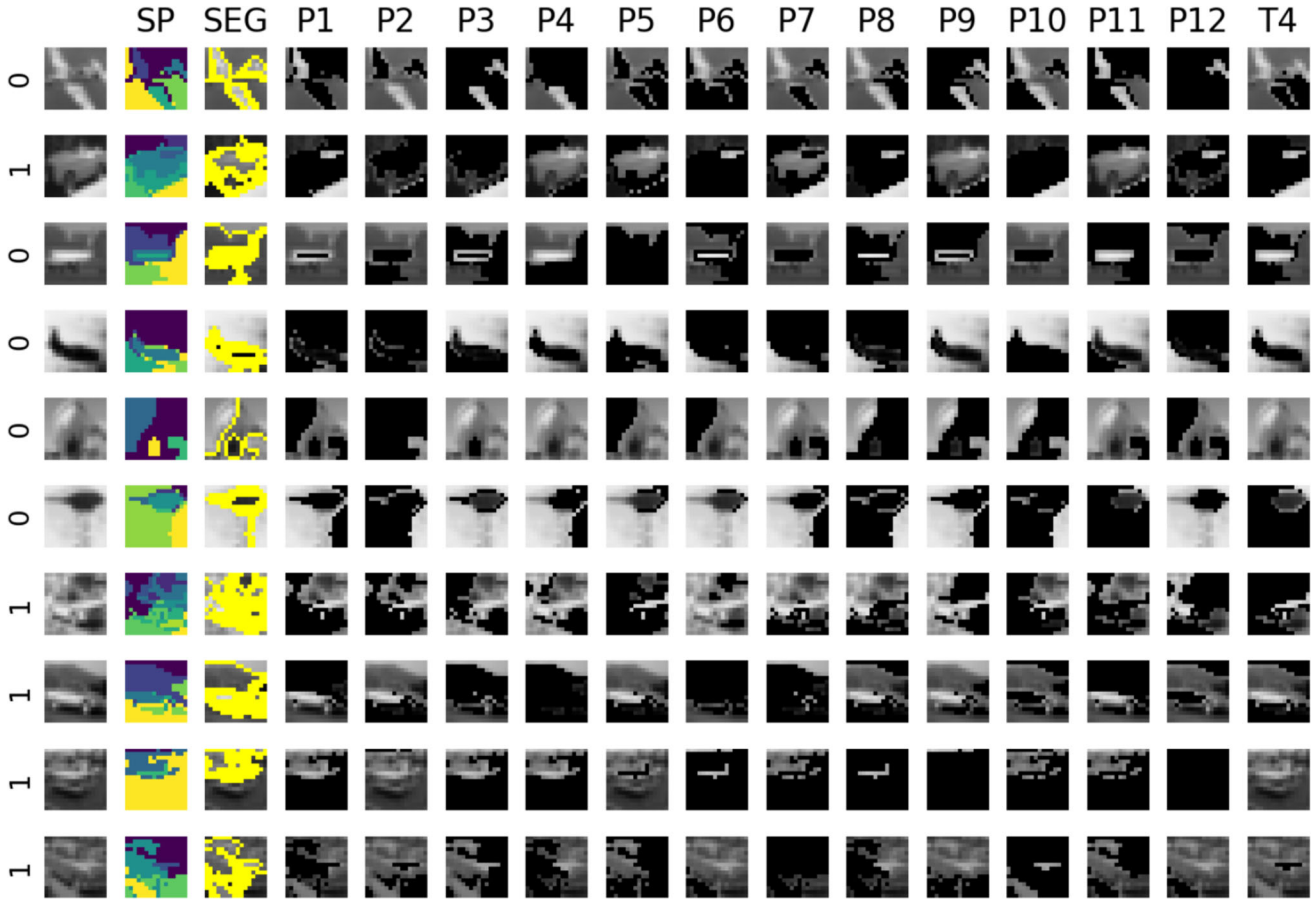


**Fig. 18** Top 4 features in CIFAR10 for binary classification: The first column displays the original CIFAR10 image, the second column shows the superpixels, the third column shows the segmentation, columns 4 to 15 show the 12 perturbations generated, and the last column highlights the top 4 features identified by LIME

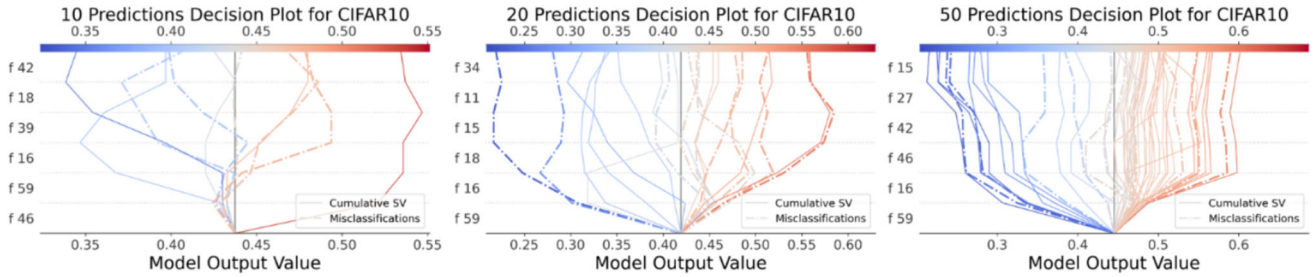


**Fig. 19** SHAP explanations for binary classification: From left to right, this visualization shows the effect magnitude of each top 5 feature on 10, 20, and 50 predictions

model's confidence in misclassified predictions. The QRL model exhibited higher confidence in misclassifications than in true predictions, despite achieving an accuracy of 65%.

## 5 Conclusion

In this study, we introduced QRLaXAI, an explainable framework for quantum representation learning (QRL) that integrates local interpretable model-agnostic explanations (LIME) and Shapley Additive Explanations (SHAP) to enhance the interpretability of variational quantum machine learning models. The framework is designed to operate under low qubit requirements and is evaluated on benchmark image datasets. Our results demonstrate the feasibility of explaining variational quantum classifiers (VQCs) through the combined use of QRL, LIME, and SHAP. We validated QRLaXAI's effectiveness through extensive experiments on MNIST, FMNIST, KMNIST, and CIFAR10 datasets, evaluating the model's explainability across binary and multi-class tasks. The results provided insightful visual and analytical interpretations, making the decision-making process of QRL models more transparent. However, the findings revealed limitations when applied to more complex datasets, such as CIFAR10, where interpretability and accuracy were impacted. Ablation studies further highlighted the sensitivity of QRL model interpretability to quantum circuit design, particularly the number of rotational layers and variational circuit architecture. By processing and explaining quantum-transformed data derived from real-world datasets, QRLaXAI establishes a robust baseline for advancing quantum machine learning explainability research.

Despite these advancements, several limitations remain. This study does not compare QRL models against classical machine learning models, as the primary focus is on validating the reliability and explainability of QRL models. While such comparisons are valuable for understanding the unique advantages of QML interpretability, they are beyond the current scope and will be addressed in future work. We also acknowledge the inherent sensitivity of LIME and SHAP explanations to factors such as data perturbations, model variability, and sampling strategies, which can introduce uncertainties in the generated explanations. These sensitivities are further influenced by quantum noise and gate variability in quantum circuits. Although this study provides initial insights, a deeper statistical analysis, including the use of confidence intervals, uncertainty quantification, and stabilization techniques, is required to fully evaluate the robustness of these explainability methods in quantum contexts.

Future work will focus on several key directions to address these limitations and advance the field of quantum interpretability. These include exploring circuit-level and gate-level explainability to identify the contributions of individual quantum gates and layers, particularly concerning quantum phenomena such as entanglement and superposition. Additionally, we plan to evaluate QRLaXAI on pure quantum datasets, such as the Quantum Chemistry Dataset (QCD), VQE Molecule Dataset, and QAOA Results, to distinguish its capabilities from classical approaches. Comparative analyses with classical machine learning models will be conducted to assess the interpretability advantages of QML and its practical utility. Furthermore, the framework will be tested on real quantum hardware to evaluate its reliability under hardware-specific constraints, such as noise and decoherence. Finally, we aim to enhance the robustness of LIME and SHAP explanations by incorporating advanced statistical tools to quantify and mitigate uncertainties, ensuring more reliable interpretability even in noisy quantum environments.

**Acknowledgements** We would like to extend our acknowledgment to Robert Shen from RACE (RMIT AWS Cloud Supercomputing Hub) and Jeff Paine from Amazon Web Services (AWS) for their invaluable provision of computing resources.

**Author contribution** A.K.K.D. (Asitha Kottahachchi Kankanamge Don) conceptualized and developed the methodology, conducted all experiments, performed data analysis, and wrote the main manuscript text. I.K. (Ibrahim Khalil) provided guidance, supervision, and critical feedback throughout the research and manuscript preparation. All authors reviewed and approved the final manuscript.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions. This work is supported by the Australian Research Council Discovery Project (DP210102761).

**Data availability** No datasets were generated or analyzed during the current study.

## Declarations

**Conflict of interest** The authors declare no competing interests.



## References

(2024) Amazon: Qiskit provider for Amazon Braket. https://aws.amazon.com/blogs/quantum-computing/introducing-the-qiskit-provider-for-amazon-braket/. Accessed 12 Jan 2024

(2024) Amazon: quantum cloud computing service - Amazon Braket - AWS. https://aws.amazon.com/braket/. Accessed 14 Sept 2024

(2024) IBM: COBYLA. https://docs.quantum.ibm.com/api/qiskit/0.26/qiskit.algorithms.optimizers.COBYLA. Accessed 19 Feb 2024

(2024) IBM: TwoLocal. https://docs.quantum.ibm.com/api/qiskit/docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.realamplitudes. Accessed 15 Nov 2024

(2024) IonQ: IonQ harmony. https://ionq.com/quantum-systems/harmony. Accessed 29 Sept 2023

(2024a) IBM: RealAmplitudes. https://docs.quantum.ibm.com/api/qiskit/docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.realamplitudes. Accessed 15 Nov 2024

(2024b) IBM: EfficientSU2. https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.EfficientSU2. Accessed 12 Jan 2024

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052. Accessed 10 Jun 2024

Ahmed I, Jeon G, Piccialli F (2022) From artificial intelligence to explainable artificial intelligence in Industry 4.0: a survey on what, how, and where. IEEE Trans Ind Inf 18(8):5031–5042. https://doi.org/10.1109/TII.2022.3146552. Accessed 10 Jun 2024

Ahmed S, Shamim Kaiser M, Hossain MS, Andersson K (2024) A comparative analysis of LIME and SHAP interpreters with explainable ML-based diabetes predictions. IEEE Access 1–1. https://doi.org/10.1109/ACCESS.2024.3422319. Accessed 13 Oct 2024

Alabi RO, Elmusrati M, Leivo I, Almangush A, Mäkitie AA (2023) Machine learning explainability in nasopharyngeal cancer survival using LIME and SHAP. 13(1):8984. Nature Publishing Group. https://doi.org/10.1038/s41598-023-35795-0. Accessed 13 Oct 2024

Apley DW, Zhu J (2020) Visualizing Eff predictor variables black box supervised learn models 82(4):1059–1086. https://doi.org/10.1111/rssb.12377. Accessed 13 Oct 2024

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W (2015) Pixel-wise explanations non-linear classifier decis layer-wise relevance propagation. 10(7):0130140. https://doi.org/10.1371/journal.pone.0130140. Public Library of Science. Accessed 10 Oct 2024

Belis V, González-Castillo S, Reissel C, Vallecorsa S, Combarro EF, Dissertori G, Reiter F (2021) Higgs Anal Quant Classifiers 251:03070. EDP Sciences. https://doi.org/10.1051/epjconf/202125103070. Accessed 17 Oct 2024

Benedetti M, Lloyd E, Sack S, Fiorentini M (2019) Parameterized Quant Circ Mach Learn Models 4(4). IOP Publishing. https://doi.org/10.1088/2058-9565/ab4eb5. Accessed 06 Oct 2024

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature Publishing. Group. 549(7671):195–202. https://doi.org/10.1038/nature23474. Accessed 10 Jun 2024

Bodria F, Giannotti F, Guidotti R, Naretto F, Pedreschi D, Rinzivillo S (2023) Benchmarking and survey of explanation methods for black box models 37(5):1719–1778. https://doi.org/10.1007/s10618-023-00933-9. Accessed 10 Jun 2024

Bravyi S, Cross AW, Gambetta JM, Maslov D, Rall P, Yoder TJ (2024) High-threshold and low-overhead fault-tolerant quantum memory. 627(8005):778–782. Nature Publishing Group. https://doi.org/10.1038/s41586-024-07107-7. Accessed 06 Oct 2024

Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational Quant Algorithm 3(9):625–644. Nature Publishing Group. https://doi.org/10.1038/s42254-021-00348-9. Accessed 06 Oct 2024

Cerezo M, Verdon G, Huang HY, Cincio L, Coles PJ (2022) Challenges Oppor Quant Mach Learn 2(9):567–576. https://doi.org/10.1038/s43588-022-00311-3. Accessed 06 Oct 2024

Chalumuri A, Kune R, Kannan S, Manoj BS (2022) Quantum–classical image processing for scene classification. 6(6):1–4. https://doi.org/10.1109/LSENS.2022.3173253 . Accessed 24 Sept 2023

Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: Proceedings of the 37th international conference on machine learning, PMLR, ???, pp 1597–1607. https://proceedings.mlr.press/v119/chen20j.html Accessed 10 Aug 2023

Choo J, Liu S (2018) Visual analytics for explainable deep learning. IEEE Comput Graph Appl 38(4):84–92. https://doi.org/10.1109/MCG.2018.042731661. Accessed 08 Oct 2024

Choudhury S, Dutta A, Ray D (2021) Chaos and complexity from quantum neural network. A study with diffusion metric in machine learning. 2021(4):138. https://doi.org/10.1007/JHEP04(2021)138. Accessed 08 Oct 2024

Ciliberto C, Herbster M, Ialongo AD, Pontil M, Rocchetto A, Severini S, Wossnig L (2017) Quantum machine learning: a classical perspective. Proc Math Phys Eng Sci 474. https://doi.org/10.1098/rspa.2017.0551

Clanuwat T, Bober-Irizar M, Kitamoto A, Lamb A, Yamamoto K, Ha D (2018) Deep learning for classical Japanese literature. https://doi.org/10.48550/arXiv.1812.01718. Accessed 16 Nov 2024

Covert I, Lee SI (2021) Improving KernelSHAP: practical Shapley Value estimation using linear regression. In: Proceedings of The 24th international conference on artificial intelligence and statistics, PMLR, ???, pp 3457–3465. ISSN: 2640-3498. https://proceedings.mlr.press/v130/covert21a.html. Accessed 15 Nov 2024

Datta A, Sen S, Zick Y (2016) Algorithmic transparency via quantitative input influence: theory and experiments with learning systems. In: 2016 IEEE symposium on security and privacy (SP), pp

598–617. https://doi.org/10.1109/SP.2016.42. ISSN: 2375-1207. https://ieeexplore.ieee.org/document/7546525. Accessed 10 Oct 2024

Deng L (2012) The MNIST database of handwritten digit images for machine learning research [best of the web]. 29(6):141–142. https://doi.org/10.1109/MSP.2012.2211477. Accessed 24 Sept 2023

Dieber J, Kirrane S (2020) Why model why? Assessing the strengths and limitations of LIME. Accessed 13 Oct 2024

Don AKK, Khalil I (2024) Q-SupCon: quantum-enhanced supervised contrastive learning architecture within the representation learning framework. https://doi.org/10.1145/3660647 . Accessed 13 Oct 2024

Farhi E, Neven H (2018) Classification with quantum neural networks on near term processors. https://doi.org/10.48550/arXiv.1802.06002. Accessed 06 Oct 2024

Frohnert F, Nieuwenburg EV (2024) Explainable Represent Learn small Quant S 5(1). IOP Publishing. https://doi.org/10.1088/2632-2153/ad16a0. Accessed 17 Oct 2024

Gaspar D, Silva P, Silva C (2024) Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron. IEEE Access 12:30164–30175. https://doi.org/10.1109/ACCESS.2024.3368377. Accessed 13 Oct 2024

Hailemariam Y, Yazdinejad A, Parizi RM, Srivastava G, Dehghantanha A (2020) An empirical evaluation of AI deep explainable tools. In: 2020 IEEE globecom workshops (GC Wkshps, pp 1–6. https://doi.org/10.1109/GCWkshps50303.2020.9367541. Accessed 13 Oct 2024

Hart S (1989) Shapley value. In: Eatwell J, Milgate M, Newman P (eds) Game Theory, Palgrave Macmillan UK, ???, pp 210–216. https://doi.org/10.1007/978-1-349-20181-5sps25. Accessed 12 Oct 2024

Hassani K, Khasahmadi AH (2020) Contrastive multi-view representation learning on graphs. In: Proceedings of the 37th international conference on machine learning. ICML'20, JMLR.org, ???, vol 119, pp 4116–4126

Hastie T, Tibshirani R, Friedman J (2009) Boosting and additive trees, Springer, New York, NY, pp 337–387. https://doi.org/10.1007/978-0-387-84858-7sps10

Heese R, Gerlach TT, Mücke S, Müller S, Jakobs M, Piatkowski N (2023) Explaining quantum circuits with shapley values: Towards explainable quantum machine learning. https://doi.org/10.48550/arXiv.2301.09138, https://publica.fraunhofer.de/handle/publica/445397

He K, Fan H, Wu Y, Xie S, Girshick R (2020) Momentum contrast for unsupervised visual representation learning. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), IEEE, ???, pp 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975. https://ieeexplore.ieee.org/document/9157636/. Accessed 10 Aug 2023

Hur T, Kim L, Park DK (2022) Quant Convolutional Neural Netw Class Data Classif 4(1):3. https://doi.org/10.1007/s42484-021-00061-x. Accessed 08 Oct 2024

Jerbi S, Gyurik C, Marshall SC, Molteni R, Dunjko V (2024) Shadows Quant Mach Learn 15(1):5676. Nature Publishing Group. https://doi.org/10.1038/s41467-024-49877-8. Accessed 13 Nov 2024

Jeyakumar JV, Noor J, Cheng YH, Garcia L, Srivastava M (2020) How can i explain this to you? an empirical study of deep neural network explanation methods. In: Advances in neural information processing systems, Curran Associates, Inc., ???, vol 33. pp 4211–4222. https://proceedings.neurips.cc/paperspsfiles/paper/2020/hash/2c29d89cc56cdb191c60db2f0bae796b-Abstract.html. Accessed 13 Oct 2024

Jia R, Yang G, Nie M, Liu Y, Zhang M (2023) Automatic optimization of variational quantum algorithm-based classifiers. In: Proceedings of the 2022 5th international conference on artificial intelligence and pattern recognition. AIPR '22, pp 1–7. Association for computing machinery, New York, NY, USA. https://doi.org/10.1145/3573942.3573943

Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, Maschinot A, Liu C, Krishnan D (2020) Supervised contrastive learning. In: Advances in neural information processing systems, Curran Associates, Inc., ???, vol 33. pp 18661–18673. https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html. Accessed 10 Aug 2023

Kielpinski D, Monroe C, Wineland DJ (2002) Architecture for a large-scale ion-trap quantum computer. 417(6890):709–711 (2002). Nature Publishing Group. https://doi.org/10.1038/nature00784. Accessed 12 Jan 2024

Kottahachchi Kankanamge Don A, Khalil I, Atiquzzaman M (2024) A fusion of supervised contrastive learning and variational quantum classifiers. IEEE Trans Consum Electr 70(1):770–779. https://doi.org/10.1109/TCE.2024.3351649. Accessed 13 Oct 2024

Krizhevsky A (2009) Learning multiple layers of features from tiny images, 32–33

Lau JWZ, Lim KH, Shrotriya H, Kwek LC (2022) NISQ computing: where are we and where do we go? 32(1):27. https://doi.org/10.1007/s43673-022-00058-z. Accessed 10 Jun 2024

Li W, Deng DL (2021) Recent advances for quantum classifiers. Sci Chin Phys Mech Astron 65(2):220301. https://doi.org/10.1007/s11433-021-1793-6

Lin YS, Lee WC, Celik ZB (2021) What do you see?: Evaluation of explainable artificial intelligence (XAI) interpretability through neural backdoors. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, ACM, ???, pp 1027–1035. https://doi.org/10.1145/3447548.3467213. Accessed 13 Oct 2024

Lipovetsky S, Conklin M (2001) Anal Regression Game Theory Approach 17(4):319–330. https://doi.org/10.1002/asmb.446

Liu S, Wang X, Liu M, Zhu J (2017) Towards Better Anal Mach Learn Models Vis Anal Perspect 1(1):48–56. https://doi.org/10.1016/j.visinf.2017.01.006. Accessed 08 Oct 2024

Lundberg SM, Lee SI (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems, vol 30. Curran Associates, Inc., ??? (2017). https://papers.nips.cc/paperspsfiles/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html. Accessed 10 Oct 2024

Maheshwari D, Sierra-Sosa D, Garcia-Zapirain B (2022) Variational quantum classifier for binary classification: Real vs synthetic dataset. IEEE Access 10:3705–3715. https://doi.org/10.1109/ACCESS.2021.3139323. Accessed 17 Oct 2024

Minh D, Wang HX, Li YF, Nguyen TN (2022) Explainable Artif Intell Compr Rev 55(5):3503–3568. https://doi.org/10.1007/s10462-021-10088-y. Accessed 10 Jun 2024

Mohseni S, Zarei N, Ragan ED (2021) A multidisciplinary survey and framework for design and evaluation of explainable ai systems. 11(3–4). https://doi.org/10.1145/3387166

Molnar C (2020) Interpretable Machine Learning. Lulu.com, ???. Google-Books-ID: jBm3DwAAQBAJ

Panati C, Wagner S, Brüggenwirth S (2022) Feature relevance evaluation using grad-CAM, LIME and SHAP for deep learning SAR data classification. In: 2022 23rd international radar symposium (IRS), pp 457–462. https://doi.org/10.23919/IRS54158.2022.9904989. ISSN: 2155-5753

Pira L, Ferrie C (2024) Interpretability Quant Neural Netw 6(2):52. https://doi.org/10.1007/s42484-024-00191-y. Accessed 07 Oct 2024

Pira L, Ferrie C (2024) Interpretability Quant Neural Netw 6(2):52. https://doi.org/10.1007/s42484-024-00191-y. Accessed 08 Oct 2024

Preskill J (2018) Quantum computing in the NISQ era and beyond. Quant 2:79. https://doi.org/10.22331/q-2018-08-06-79

Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '16, Association for Computing Machinery, ???, pp 1135–1144. https://doi.org/10.1145/2939672.2939778. Accessed 13 Nov 2024

Rivas P, Zhao L, Orduz J (2021) Hybrid quantum variational autoencoders for representation learning. In: 2021 international conference on computational science and computational intelligence (CSCI), pp 52–57. https://doi.org/10.1109/CSCI54926.2021.00085, https://ieeexplore.ieee.org/document/9799154. Accessed 17 Oct 2024

Rohe T, Schuman D, Nusslein J, Sunkel L, Stein J, Linnhoff-Popien C (2024) The questionable influence of entanglement in quantum optimisation algorithms. https://doi.org/10.48550/arXiv.2407.17204

Romero J, Olson JP, Aspuru-Guzik A (2017) Quantum autoencoders for efficient compression of quantum data. 2(4):045001. IOP Publishing. https://doi.org/10.1088/2058-9565/aa8072. Accessed 14 Nov 2024

Sammani F, Joukovsky B, Deligiannis N (2024) Visualizing and understanding contrastive learning. IEEE Trans Image Process 33:541–555. https://doi.org/10.1109/TIP.2023.3346295. Accessed 12 Oct 2024

Saranya A, Subhashini R (2023) A Syst Rev Explainable Artif Intell Models Appl Recent Develop Future Trends 7. https://doi.org/10.1016/j.dajour.2023.100230. Accessed 08 Oct 2024

Sarkar A, Vijaykeerthy D, Sarkar A, Balasubramanian VN (2022) A framework for learning ante-hoc explainable models via concepts, IEEE Computer Society, ???, pp 10276–10285. https://doi.org/10.1109/CVPR52688.2022.01004. https://www.computer.org/csdl/proceedings-article/cvpr/2022/694600k0276/1H1l4U6Bu3m. Accessed 13 Nov 2024

Schuld M, Bocharov A, Svore KM, Wiebe N (2020) Circ-centric Quant Classifiers 101(3). American Physical Society. https://doi.org/10.1103/PhysRevA.101.032308. Accessed 06 Oct 2024

Schuld M, Killoran N (2019) Quant Mach Learn Feature Hilbert Spaces 122(4). American Physical Society. https://doi.org/10.1103/PhysRevLett.122.040504. Accessed 06 Oct 2024

Schuld M, Killoran N (2022) Is quantum advantage the right goal for quantum machine learning? 3(3):030101. American Physical Society. https://doi.org/10.1103/PRXQuantum.3.030101. Accessed 06 Oct 2024

Schuld M, Petruccione F (2018) Supervised learning with quantum computers. Quant Sci Technol. Springer, ???. https://doi.org/10.1007/978-3-319-96424-9. Accessed 10 Jun 2024

Shahariar GM, Hasan T, Iqbal A, Uddin G (2023) Contrastive learning for API aspect analysis. In: 2023 38th IEEE/ACM international conference on automated software engineering (ASE), pp 637–648. https://doi.org/10.1109/ASE56229.2023.00064. ISSN: 2643-1572. https://ieeexplore.ieee.org/abstract/document/10298556 Accessed 17 Oct 2024

Shrikumar A, Greenside P, Kundaje A (2019) Learning important features through propagating activation differences. https://doi.org/10.48550/arXiv.1704.02685. Accessed 10 Oct 2024

Strobl M, Kuehn E, Fischer M, Streit A (2024) Improving noisy hybrid quantum graph neural networks for particle decay tree reconstruction. EPJ Web Conf. https://doi.org/10.1051/epjconf/202429512004

Štrumbelj E, Kononenko I (2014) Explaining Prediction Models Individ Predictions Feature contributions. 41(3):647–665. https://doi.org/10.1007/s10115-013-0679-x. Accessed 10 Oct 2024

Thakare PM (2023) Bridging the gap between quantum computing and artificial intelligence. Int J Sci Res Eng Manag. https://doi.org/10.55041/ijsrem27848

Thumwanit N, Lortaraprasert C, Raymond R (2021) Invited: Trainable discrete feature embeddings for quantum machine learning. In: 2021 58th ACM/IEEE design automation conference (DAC), pp 1352–1355. https://doi.org/10.1109/DAC18074.2021.9586190. Accessed 17 Oct 2024

Treinish M (2023) Qiskit 0.44.0. Zenodo. https://doi.org/10.5281/ZENODO.2573505. Accessed 09 Aug 2023

Vedaldi A, Soatto S (2008) Quick shift and kernel methods for mode seeking. In: Forsyth D, Torr P, Zisserman A (eds) Computer Vision – ECCV 2008, Springer, ???, pp 705–718. https://doi.org/10.1007/978-3-540-88693-8sps52

Vimbi V, Shaffi N, Mahmud M (2024) Interpreting Artif Intell Models syst Rev ApplLIME and SHAP Alzheimer's Dis Detect 11(1):10. https://doi.org/10.1186/s40708-024-00222-1. Accessed 13 Oct 2024

Wang X, Qi GJ (2023) Contrastive learning with stronger augmentations. IEEE Trans Patt Anal Mach Intell 45(5):5549–5560. https://doi.org/10.1109/TPAMI.2022.3203630. Accessed 17 Oct 2024

Wootton JR, Loss D (2012) High Threshold Error Correct Surf Code 109(16). American Physical Society. https://doi.org/10.1103/PhysRevLett.109.160503. Accessed 15 Nov 2024

Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. https://doi.org/10.48550/arXiv.1708.07747. Accessed 24 Sept 2023

Xu Y, Raja K, Pedersen M (2022) Supervised contrastive learning for generalizable and explainable DeepFakes detection. In: 2022 IEEE/CVF winter conference on applications of computer vision workshops (WACVW), pp 379–389. https://doi.org/10.1109/WACVW54805.2022.00044. ISSN: 2690-621X. https://ieeexplore.ieee.org/document/9707568. Accessed 12 Oct 2024

Yano H, Suzuki Y, Itoh KM, Raymond R, Yamamoto N (2021) Efficient discrete feature encoding for variational quantum classifier. IEEE Trans Quant Eng 2:1–14. https://doi.org/10.1109/TQE.2021.3103050. Accessed 17 Oct 2024

Yuan X, Tian Y, Zhang C, Ye Y, Chawla NV, Zhang C (2024) Graph cross supervised learning via generalized knowledge. In: Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. KDD '24, Association for Computing Machinery, ???, pp 4083–4094. https://doi.org/10.1145/3637528.3671830. Accessed 17 Oct 2024

Zhang Z, Jang J, Trabelsi C, Li R, Sanner S, Jeong Y, Shim D (2022) ExCon: Explanation-driven supervised contrastive learning for image classification. https://doi.org/10.48550/arXiv.2111.14271. Accessed 13 Nov 2024

Zhang S, Zhou Y, Qin Z, Li R, Du C, Xiao Z, Zhang Y (2024) Machine-learning insights on entanglement-trainability correlation of parameterized quantum circuits

Zheng M, Wang F, You S, Qian C, Zhang C, Wang X, Xu C (2021) Weakly supervised contrastive learning. In: 2021 IEEE/CVF international conference on computer vision (ICCV), pp 10022–10031. https://doi.org/10.1109/ICCV48922.2021.00989. ISSN: 2380-7504. https://ieeexplore.ieee.org/document/9710997 Accessed 17 Oct 2024