Medical Imaging Classification with Cold-Atom Reservoir Computing using Auto-Encoders and Surrogate-Driven Training

Nuno Batista
Department of Informatics Engineering
Faculty of Sciences and Technology, University of Coimbra
Coimbra, Portugal
nunomarquesbatista@gmail.com

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—We introduce a quantum-classical hybrid architecture for medical image classification based on neutral atom quantum processors. This approach is designed to address the challenges of medical imaging, with a particular focus on tasks such as polyp detection and classification. By integrating an autoencoder guided by a quantum reservoir, the pipeline learns compact and discriminative representations of image data that are also well-suited for quantum reservoir computing. To overcome the non-differentiability of quantum measurements, we circumvent this 'gradient barrier' by incorporating a differentiable surrogate model that simulates the behaviour of the quantum layer, enabling end-to-end backpropagation. The guided training process jointly optimizes for both image reconstruction and classification accuracy, ensuring that the latent representations are both meaningful and effective for quantum processing. In our implementation, image data is encoded as atom detuning parameters in a Rydberg Hamiltonian, and quantum embeddings are obtained through expectation values. These embeddings are then passed to a linear classifier, enabling faster training and inference compared to deep classical networks. Our experiments show that this method outperforms traditional approaches using PCA or unguided autoencoders. We also conduct ablation studies to evaluate the impact of quantum and training parameters, demonstrating the robustness and flexibility of the proposed pipeline for real-world medical imaging applications, even in the NISQ era.**

***Index Terms*—Reservoir Computing, Quantum-Guided Autoencoding, Neutral Atoms, Autoencoder, Dimensionality Reduction, Quantum Machine Learning, Hybrid Quantum-Classical Algorithms, Medical Image Classification, Quantum Surrogate Models**

I. Introduction

A. Background and Motivation

Advances in medical imaging have significantly improved disease diagnosis and treatment planning. For conditions like colorectal cancer, early detection of polyps through colonoscopy image analysis is critical for reducing mortality [**?**]. Deep learning techniques, especially autoencoders, are widely used to extract compressed, informative features from high-dimensional images for classification and segmentation tasks [**?**]. However, classical neural networks may struggle to capture intricate correlations in complex medical data.

Quantum computing offers novel opportunities for machine learning, particularly through quantum reservoir computing (QRC), where a physical quantum system processes classical inputs into high-dimensional nonlinear embeddings [1], [2]. Recent works show that analog quantum systems, such as neutral-atom platforms, can serve as untrained reservoirs with rich dynamics for temporal and pattern recognition tasks [3], [4]. In hybrid approaches, a classical encoder compresses image data, and a quantum reservoir expands the encoded features into a higher-dimensional space, potentially boosting classification performance.

A major challenge in such hybrid quantum-classical models is the non-differentiability of quantum measurements, which obstructs gradient-based optimization. Additionally, tuning quantum parameters can suffer from barren plateaus, where gradients vanish in high-dimensional Hilbert spaces [5]. To address this, we introduce a classical neural surrogate that emulates the quantum reservoir's input-output behavior. This surrogate enables end-to-end training via backpropagation, while the quantum system remains fixed and non-trainable.

B. Contributions of This Work

We propose a quantum-guided autoencoder architecture that integrates a classical image encoder with a neutral-atom quantum reservoir.

A classical surrogate network of the reservoir itself enables gradient flow through the whole model during training.

The model is evaluated and compared to classical benchmarks on three different datasets:

1) A synthetic dataset of polyp images, generated to simulate realistic medical imaging scenarios.
2) Real image patches extracted from the CVC-ClinicDB dataset, a well-known benchmark for polyp detection.
3) A reduced version of the MNIST dataset, containing only the digits 0 and 1, suitable for binary classification tasks.

Our results illustrate the viability of QRC for real-world medical tasks and offer a scalable path to hybrid quantum-

classical learning, even in the noisy intermediate-scale quantum (NISQ) era.

II. Background

A. Principles of Reservoir Computing

Reservoir computing is a computational framework derived from recurrent neural networks (RNNs). It involves a fixed, high-dimensional dynamical system—the reservoir—that projects input data into a rich feature space. Only the output layer is trained, simplifying the learning process and reducing computational overhead. This approach is particularly effective for time-series prediction and pattern recognition tasks.

Mathematically, let $u(t) \in \mathbb{R}^m$ be the input at time t, $x(t) \in \mathbb{R}^n$ the reservoir state, and $y(t) \in \mathbb{R}^k$ the output. The reservoir dynamics and output are given by:

$$x(t) = f(W_{in}u(t) + W_{res}x(t-1)) \tag{1}$$

$$y(t) = W_{out}x(t) \tag{2}$$

Where f is a nonlinear activation function, W_{in} and W_{res} are fixed input and reservoir weight matrices, and W_{out} is the trained output weight matrix.

A diagram of a typical reservoir computing architecture is shown in 'Fig.-1'.

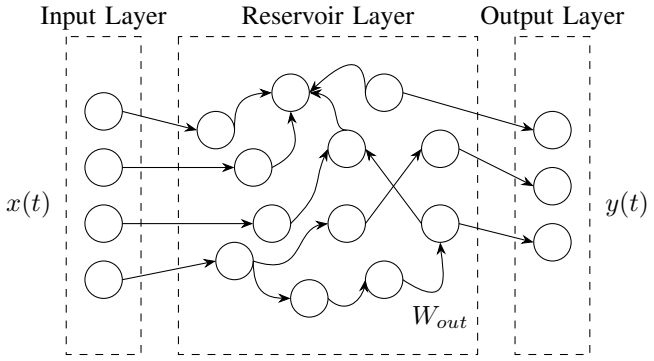

Fig. 1. Reservoir computing architecture showing input nodes, a recurrent reservoir network with internal dynamics, and an output layer.

B. Quantum Reservoir Computing

Quantum Reservoir Computing (QRC) extends the reservoir computing paradigm into the quantum domain. By leveraging quantum systems' inherent properties, such as superposition and entanglement, QRC aims to enhance computational capabilities. Implementations using quantum oscillators have shown promise in solving complex learning tasks, offering advantages over classical counterparts. Notably, large-scale experiments utilizing neutral-atom analog quantum computers have demonstrated the scalability and effectiveness of QRC in various machine learning applications [4]

In QRC, classical input data $u(t)$ is encoded into quantum states $|\psi(t)\rangle$, which evolve under a fixed Hamiltonian H:

$$|\psi(t+1)\rangle = U|\psi(t)\rangle = e^{-iH\Delta t}|\psi(t)\rangle, \tag{3}$$

where U is the unitary evolution operator. Measurements of observables $\hat{O}$ yield outputs:

$$y(t) = \langle\psi(t)|\hat{O}|\psi(t)\rangle. \tag{4}$$

The output weights are trained classically, while the quantum reservoir remains fixed.

C. Reservoir Computing with Neutral Atoms

As quantum Reservoir Computing (QRC) leverages the complex dynamics of quantum systems to process information, extending the classical reservoir computing paradigm into the quantum domain. Neutral atom platforms, particularly those utilizing Rydberg states, have emerged as promising candidates for implementing QRC due to their scalability and controllable interactions.

In the work by M. Kornjača et al. [4], a large-scale, gradient-free QRC algorithm was developed and experimentally implemented on a neutral-atom analog quantum computer. This system achieved competitive performance across various machine learning tasks, including classification and time-series prediction, demonstrating effective learning with increasing system sizes up to 108 qubits.

The dynamics of such neutral atom systems can be described by the Rydberg Hamiltonian, which captures the essential physics of laser-driven interactions among atoms in Rydberg states. Following Kornjača et al. [4] The Hamiltonian for a system of neutral-atoms is given by:

$$\begin{aligned} H(t) = \frac{\Omega(t)}{2}\sum_j \left(|g_j\rangle\langle r_j| + |r_j\rangle\langle g_j|\right) \\ + \sum_{j<k} V_{jk}n_jn_k - \sum_j \left[\Delta_{\mathrm{g}}(t) + \alpha_j\Delta_{\mathrm{l}}(t)\right]n_j, \end{aligned} \tag{5}$$

where $\Omega(t)$ is the global Rabi drive amplitude between a ground state $|g_j\rangle$ and a highly-excited Rydberg state $|r_j\rangle$ of an atom, $n_j = |r_j\rangle\langle r_j|$, $V_{jk} = C/\|r_j - r_k\|^6$ describes the van der Waals interactions between atoms, and the detuning is split into a global term $\Delta_{\mathrm{g}}(t)$ and a site-dependent term $\Delta_{\mathrm{l}}(t)$, with site modulation $\alpha_j \in [0, 1]$.

By initializing the system in a specific state and allowing it to evolve under this Hamiltonian, the resulting quantum state encodes information about the input data. Measurements of observables on this state yield outputs that can be used for tasks such as classification or prediction, with only the final readout layer requiring training.

D. Dimensionality Reduction for Image Data

Dimensionality reduction is a crucial preprocessing step in machine learning and data analysis, aiming to reduce the number of input variables in a dataset while preserving as much information as possible. This process enhances computational efficiency, mitigates the curse of dimensionality, and facilitates data visualization

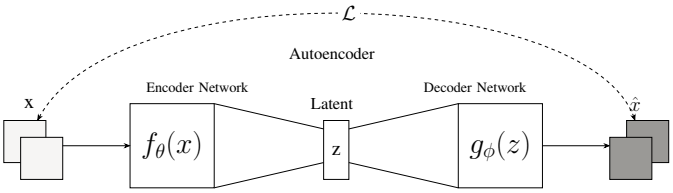

Fig. 2. Autoencoder architecture. The encoder network $f_\theta(x)$ compresses the input x into a latent representation z, while the decoder network $g_\phi(z)$ reconstructs the input from the latent representation, producing $\hat{x}$. The reconstruction loss $\mathcal{L}$ measures the compresses the input x into a latent representation z, while the decoder network $g_\phi(z)$ reconstructs the input from the latent representation, producing $\hat{x}$.

1) Principal Component Analysis: Principal Component Analysis (PCA) [6] is a linear dimensionality reduction technique that transforms a set of correlated variables into a set of uncorrelated variables called principal components. The goal is to capture the maximum variance in the data with the fewest number of components.

PCA is effective for datasets where the principal components align with the directions of maximum variance, but it may not capture complex, nonlinear relationships in the data [7].

2) Autoencoder Architectures: Autoencoders are a class of artificial neural networks designed to learn efficient codings of input data in an unsupervised manner. They consist of two main parts: an encoder that compresses the input into a latent-space representation, and a decoder that reconstructs the input from this representation.

Given an input $x \in \mathbb{R}^d$, the encoder maps x to a latent representation $z \in \mathbb{R}^k$ (where $k < d$):

$$z = f_\theta(x) \tag{6}$$

The decoder then reconstructs the input:

$$\hat{x} = g_\phi(z) \tag{7}$$

The network is trained to minimize the reconstruction loss:

$$\mathcal{L}(x, \hat{x}) = \|x - \hat{x}\|^2 \tag{8}$$

This typical architecture of an autoencoder is illustrated in 'Fig.-2'.

Autoencoders can capture complex, nonlinear relationships in the data, making them suitable for tasks like image compression, denoising, and anomaly detection [**?**], [8].

3) Quantum-Guided Autoencoding: Quantum-Guided Autoencoders integrate quantum computing principles into the autoencoder framework to leverage quantum advantages in processing and representing data. These models aim to perform dimensionality reduction and classification within a single architecture, enhancing performance on complex datasets.

In the Quantum-Guided Autoencoder (QGAE) model, a classical encoder first reduced the dimensionality of the input data. The compressed data is then processed by a parametrized quantum circuit, which acts as the decoder and classifier. The quantum circuit transforms the input state $|\psi_{\text{in}}\rangle$ into an output state $|\psi_{\text{out}}\rangle$ using a unitary operation

$$|\psi_{\text{out}}\rangle = U(\theta)|\psi_{\text{in}}\rangle \tag{9}$$

Measurements on $|\psi_{\text{out}}\rangle$ yield the final classification result. The parameters θ are optimized to minimize a loss function that combines reconstruction error and classification accuracy.

This approach has demonstrated improved performance over traditional methods in tasks such as identifying the Higgs boson in particle collision data, showcasing the potential of quantum-guided models in handling high-dimensional, complex datasets [9].

III. Methodology

A. System Architecture Overview

Our proposed pipeline integrates classical and quantum components for dimensionality reduction and classification. A classical autoencoder compresses high-dimensional image data into a lower-dimensional latent space. This latent representation is encoded into a Rydberg atom chain, serving as input to a quantum reservoir. The reservoir's dynamics, governed by the Rydberg Hamiltonian, produce quantum embeddings through measurements of quantum observables.

To facilitate end-to-end training, we employ a differentiable surrogate model that approximates the quantum reservoir's behavior, enabling gradient flow from the classification loss back to the autoencoder. The surrogate's outputs feed into a linear classifier trained to map quantum embeddings to class labels.

The autoencoder is trained to minimize a combined loss function that interpolates between reconstruction loss and classification loss, promoting representations that are both informative for input reconstruction and effective for quantum reservoir computing.

Once trained, the encoder compresses data into the latent space, which is then processed by the actual quantum reservoir. The resulting quantum embeddings are used to train a linear classifier for final predictions.

The overall architecture is illustrated in 'Fig.-III-A'.

B. Quantum Guided Autoencoder

The Quantum Guided Autoencoder (QGA) combines classical preprocessing with quantum processing to leverage the strengths of both paradigms. The classical encoder reduces the dimensionality of the input data, facilitating efficient quantum processing. The quantum circuit then processes the compressed data, aiming to reconstruct the original input and perform classification simultaneously.

1) Loss Function Design: The QGA is trained to minimize a composite loss function that balances reconstruction fidelity and classification accuracy:

$$\mathcal{L} = (1 - \lambda) \cdot \mathcal{L}_R + \lambda \cdot \mathcal{L}_C, \tag{10}$$

where:

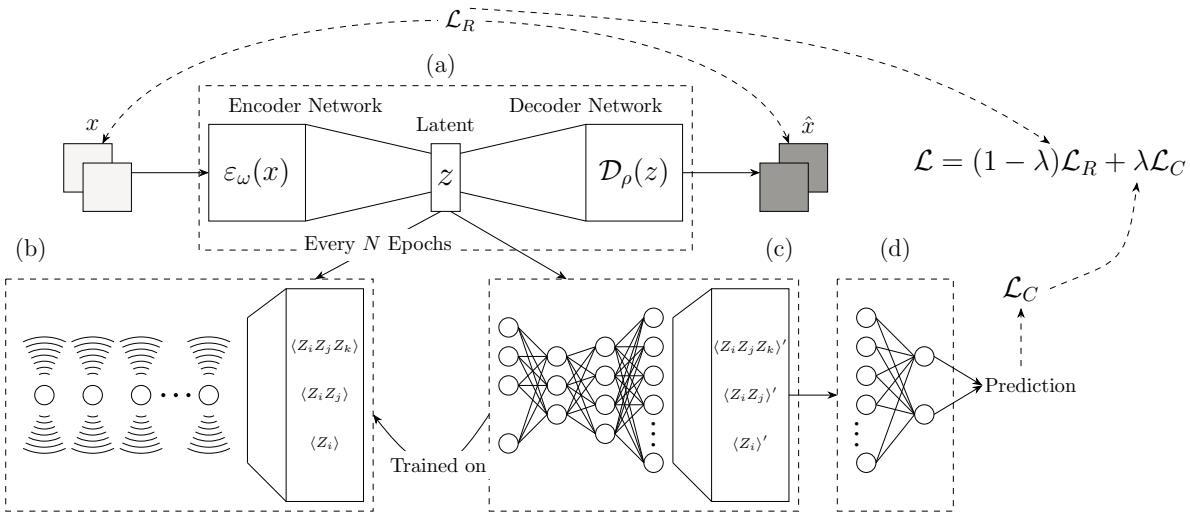

Fig. 3. Overview of the Quantum-Guided Autoencoder Reservoir Computing System.

- $\mathcal{L}_R = \|x - \hat{x}\|^2$ is the reconstruction loss, measuring the mean squared error between the input x and its reconstruction $\hat{x}$.
- $\mathcal{L}_C = -\sum_i y_i \log(\hat{y}_i)$ is the classification loss, computed as the cross-entropy between the true labels y_i and the predicted probabilities $\hat{y}_i$.
- $0 < \lambda < 1$ is a hyperparameter that controls the trade-off between reconstruction and classification objectives.

C. The Gradient Barrier Problem

Hybrid quantum-classical models, such as the Quantum Guided Autoencoder (QGAE), aim to leverage quantum computational advantages within classical machine learning frameworks. However, a significant challenge arises due to the non-differentiable nature of quantum operations, commonly referred to as the 'gradient barrier'.

In classical neural networks, training relies on backpropagation, which requires the computation of gradients through all components of the network. In hybrid models, the inclusion of quantum layers introduces operations that are inherently non-differentiable:

- **Quantum State Preparation**: Encoding classical data into quantum states often involves operations that are not smoothly differentiable with respect to the input data.
- **Quantum Measurements**: Observing quantum states collapses them, introducing stochasticity and breaking the deterministic gradient flow required for backpropagation.

These aspects hinder the direct application of gradient-based optimization methods across the quantum-classical boundary, obstructing end-to-end training of hybrid models.

D. Surrogate Modeling for Quantum Layers

To circumvent the gradient barrier, surrogate models have been proposed. These are classical, differentiable models trained to approximate the input-output behavior of quantum circuits. By replacing the non-differentiable quantum components with their surrogate counterparts during training, gradients can be propagated through the entire network, enabling end-to-end optimization.

Recent studies have demonstrated the efficacy of surrogate models in mitigating the effects of barren plateaus and facilitating the training of hybrid quantum-classical models. These approaches allow for the practical implementation of quantum-enhanced machine learning models by leveraging classical optimization techniques while still capturing quantum computational advantages [10]

1) Architecture: In our pipeline, the surrogate model is implemented as a feedforward neural network designed to learn the mapping from the autoencoder's latent space to the quantum embeddings produced by the quantum reservoir.

Its architecture comprises multiple fully connected layers with nonlinear activation functions, facilitating the approximation of the complex, non-linear transformations inherent in quantum dynamics.

This approach draws inspiration from recent studies demonstrating the effectiveness of surrogate models in approximating quantum circuit behaviors, thereby facilitating efficient training of hybrid models. [10], [11]

2) Training Procedure:

- Data Input: Feed a batch of latent data z from the autoencoder into the real quantum reservoir, recording the resulting quantum embeddings.
- Surrogate Training: Train the surrogate model using the recorded quantum embeddings as targets, minimizing the loss function:

$$\mathcal{L}_{\text{surrogate}} = \|\hat{y} - y\|^2, \tag{11}$$

where $\hat{y}$ are the surrogate model's predictions and y are the actual quantum embeddings.

- Regular Updates: Periodically update the surrogate model during training to insure it accurately reflects the quantum reservoir's behavior. This can be done after a fixed number of epochs or when the surrogate's performance on a validation set reaches a certain threshold.

3) Gradient Flow Through Surrogate Models: By integrating the surrogate model into the training pipeline, we establish a continuous computational graph from the output layer back to the input layer, enabling the use of gradient-based optimization techniques. The surrogate model serves as a differentiable proxy for the quantum reservoir, allowing the classification loss to influence the autoencoder's parameters effectively.

As a results, we not only achieve end-to-end training of the hybrid model but also makes it computationally efficient, as we reduce the need for frequent quantum measurements.

E. Rydberg Hamiltonian and Quantum Dynamics

The quantum reservoir in our architecture is implemented using a neutral-atom system, specifically utilizing Rydberg atoms to create a programmable quantum reservoir. The dynamics of the system are governed by the Rydberg Hamiltonian, which describes the interactions between atoms in Rydberg states 5.

1) Data Encoding Schemes: The input data is encoded into the quantum system by mapping the latent representations from the autoencoder to the detuning parameters of the Rydberg Hamiltonian. This encoding process involves adjusting the detuning parameters $\Delta_l(t)$ for each atom in the chain, allowing the system to represent the input data in a way that is compatible with the quantum dynamics of the reservoir.

2) Quantum Readout Methods: The quantum reservoir's output is obtained by probing, in successive time steps, the state of the atoms after the systems evolution up to a certain time t.

The readout is performed by measuring specific observables, such as:

- **Single-atom Measurements**: $\langle Z_i \rangle$ for each atom i, where Z_i is the Pauli-Z operator.
- **Two-atom Correlations**: $\langle Z_i Z_j \rangle$ for pairs of atoms (i, j), capturing correlations between their states.
- **Three-atom Correlations**: $\langle Z_i Z_j Z_k \rangle$ for triplets of atoms (i, j, k), providing higher-order correlations.

IV. Experimental Setup

A. Datasets

We evaluate our proposed architecture on three primary datasets:

1) **Synthetic Polyp Dataset:** A synthetic dataset of polyp images generated to simulate realistic medical imaging scenarios.
2) **CVC-ClinicDB Dataset:** Real image patches extracted from the CVC-ClinicDB dataset, a well-known benchmark for polyp detection. This dataset provides a diverse set of polyp images with varying characteristics, allowing for robust evaluation of classification performance.
3) **MNIST Dataset:** A reduced version of the MNIST dataset, containing only the digits 0 and 1, suitable for binary classification tasks. This dataset serves as a controlled environment to assess the model's performance on simpler classification tasks.

B. Feature Reduction Methods

We implemented and compared three distinct feature reduction approaches:

1) Principal Component Analysis (PCA): PCA was employed as a baseline linear dimensionality reduction technique. We extracted the top-k principal components (typically 4-12) from the flattened image data, which were then used as inputs to both classical and quantum-enhanced classifiers.

2) Autoencoder: We implemented a neural network-based autoencoder with the following architecture:

- An encoder network that compresses the input to a lower-dimensional latent space.
- A decoder network that reconstructs the original input from the latent representation
- Batch normalization and dropout layers to improve training stability and prevent overfitting.

The autoencoder was trained to minimize reconstruction loss using the Adam optimizer with learning rates between 0.001—0.01 and weight decay for regularization.

3) Quantum-Guided Autoencoder: We introduced a quantum-guided autoencoder that incorporates feedback from the quantum reservoir during training. This approach combines:

- A standard autoencoder architecture
- A linear classifier that maps quantum embeddings to class labels.
- A classification loss computed on quantum embeddings
- A weighting parameter λ to control the trade-off between reconstruction and classification objectives 10.
- A differentiable surrogate model that approximates the quantum reservoir's behavior, enabling end-to-end back-propagation through the entire pipeline. The frequency of quantum updates is controlled by a hyperparameter, allowing for flexible integration of quantum dynamics into the training process.

4) Quantum Reservoir Computing Layer: Our quantum reservoir computing implementation uses a Rydberg atom simulator with the following parameterizable components:

- **Atom Chain Length:** The number of atoms in the chain, which can be adjusted to control the reservoir's capacity and complexity.
- **Rabi Frequency:** The global Rabi drive amplitude, which influences the strength of interactions between atoms.
- **Detuning Parameters:** Site-dependent detuning parameters that encode the input data.
- **Measurement Strategy:** The specific observables measured to obtain quantum embeddings.
- **Encoding Scale:** The scale of the input data encoding, which can be adjusted to optimize the quantum reservoir's response to the input data.

- **Time steps:** The number of time steps for which the quantum reservoir evolves, allowing for the capture of temporal dynamics in the data.

5) Classical Network Architectures: After the traning process of the quantum-guided autoencoder, the following quantum embeddings are passed to a linear classifier. The classifier is a simple feedforward neural network with the following architecture:

- A single hidden layer with a ReLU activation function.
- Cross-entropy loss for training, optimized using the Adam optimizer.

C. Comparison Methods

We benchmarked our model against some classical methods, including:

- **PCA + Linear Classifier:** A linear classifier trained on the PCA-reduced features.
- **Autoencoder + Linear Classifier:** A linear classifier trained on the latent representations from the autoencoder.
- **PCA + Neural Network:** A fully connected neural network trained on the PCA-reduced features.
- **Autoencoder + Neural Network:** A fully connected neural network trained on the latent representations from the autoencoder.

D. Performance Metrics

We evaluated the performance of our model using the following metrics:

- **Classification Accuracy:** The percentage of correctly classified instances in the test set.
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced measure of classification performance.
- **Confusion Matrix:** A matrix summarizing the classification results, showing true positives, false positives, true negatives, and false negatives.

E. Parameter Sweep Strategy

To optimize the performance of our quantum-guided autoencoder, we conducted a systematic parameter sweep over the following hyperparameters:

- **Guided Lambda Parameter (λ):** This parameter controls the trade-off between reconstruction and classification objectives in the loss function 10. We tested values ranging from 0.1 to 0.9 in increments of 0.1.
- **Quantum Update Frequency:** The frequency at which the quantum reservoir is updated during training. We experimented with update frequencies of 1, 5, and 10 epochs.
- **Quantum Parameters:** Including the number of atoms in the chain, Rabi frequency, and detuning parameters. We varied these parameters to assess their impact on classification performance.
- **Batch Size:**
- **Learning Rate:**

V. Results and Discussion

A. Classification Performance Comparison

B. Ablation Studies

1) Impact of Guided Lambda Parameter:

2) Effect of Quantum Update Frequency:

3) Influence of Quantum Parameters:

C. Dimensionality Reduction Comparison

D. Surrogate Model Fidelity Analysis

E. Generalization to Unseen Data

VI. Limitations and Future Work

A. Current Limitations

B. Potential Extensions

C. Hardware Implementation Considerations

VII. Conclusion

References

[1] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, Jul. 2019.

[2] K. Fujii and K. Nakajima, "Harnessing disordered quantum dynamics for machine learning," *Physical Review Applied*, vol. 8, no. 2, p. 024030, Aug. 2017.

[3] L. Domingo, G. Carlo, and F. Borondo, "Optimal quantum reservoir computing for the NISQ era," May 2022.

[4] M. Kornjavca, H.-Y. Hu, C. Zhao, J. Wurtz, P. Weinberg, M. Hamdan, A. Zhdanov, S. Cantu, H. Zhou, R. A. Bravo, K. Bagnall, J. Basham, J. Campo, A. Choukri, R. DeAngelo, P. Frederick, D. Haines, J. Hammett, N. Hsu, M.-G. Hu, F. Huber, P. N. Jepsen, N. Jia, T. Karolyshyn, M. Kwon, J. Long, J. Lopatin, A. Lukin, T. Macri, O. Markovi'c, L. A. Mart'inez-Mart'inez, X. Meng, E. Ostroumov, D. Paquette, J. M. Robinson, P. S. Rodriguez, A. Singh, N. Sinha, H. Thoreen, N. Wan, D. Waxman-Lenz, T. Wong, K.-H. Wu, P. L. S. Lopes, Y. Boger, N. Gemelke, T. Kitagawa, A. Keesling, X. Gao, A. Bylinskii, S. Yelin, F. Liu, and S.-T. Wang, "Large-scale quantum reservoir learning with an analog quantum computer," Jul. 2024.

[5] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 1, p. 4812, Nov. 2018.

[6] J. Shlens, "A Tutorial on Principal Component Analysis," Apr. 2014.

[7] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, Apr. 2016.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[9] V. Belis, P. Odagiu, M. Grossi, F. Reiter, G. Dissertori, and S. Vallecorsa, "Guided Quantum Compression for High Dimensional Data Classification," *Machine Learning: Science and Technology*, vol. 5, no. 3, p. 035010, Sep. 2024.

[10] Y. Xie, "Quantum Surrogate-Driven Image Classifier: A Gradient-Free Approach to Avoid Barren Plateaus," May 2025.

[11] F. J. Schreiber, J. Eisert, and J. J. Meyer, "Classical surrogates for quantum learning models," *Physical Review Letters*, vol. 131, no. 10, p. 100803, Sep. 2023.

CONTENTS