

# Benchmarking Quantum Convolutional Neural Networks for Classification and Data Compression Tasks

Jun Yong Khoo,<sup>1</sup> Chee Kwan Gan,<sup>1</sup> Wenjun Ding,<sup>1</sup> Stefano Carrazza,<sup>2,3,4,5</sup> Jun Ye,<sup>1</sup> and Jian Feng Kong<sup>1,\*</sup>

<sup>1</sup>*Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A\*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Republic of Singapore*

<sup>2</sup>*CERN, Theoretical Physics Department, CH-1211 Geneva 23, Switzerland*

<sup>3</sup>*TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano, Italy*

<sup>4</sup>*INFN, Sezione di Milano, I-20133 Milan, Italy*

<sup>5</sup>*Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE*

Quantum Convolutional Neural Networks (QCNNs) have emerged as promising models for quantum machine learning tasks, including classification and data compression. This paper investigates the performance of QCNNs in comparison to the hardware-efficient ansatz (HEA) for classifying the phases of quantum ground states of the transverse field Ising model and the XXZ model. Various system sizes, including 4, 8, and 16 qubits, through simulation were examined. Additionally, QCNN and HEA-based autoencoders were implemented to assess their capabilities in compressing quantum states. The results show that QCNN with RY gates can be trained faster due to fewer trainable parameters while matching the performance of HEAs.

## I. INTRODUCTION

Quantum machine learning (QML) leverages quantum computing's principles to enhance classical machine learning algorithms' performance and efficiency. Among various QML models, Quantum Convolutional Neural Networks (QCNNs) have shown significant potential in classification tasks, including both quantum [1] and classical data [2]. Inspired by classical convolutional neural networks, QCNNs exploit quantum parallelism and entanglement to process quantum data with short range entanglement structure effectively, and avoids the notorious barren plateau problem by construction due to its logarithmic circuit depth [3].

In this study, we seek to demonstrate the extent of its efficiency and advantage over the hardware-efficient ansatz (HEA), by comparing their performances on two primary machine learning tasks: phase classification and compression of quantum data. The ground states of transverse field Ising (TFI) model and the XXZ model serve as benchmarks for phase classification, while autoencoder architectures based on QCNN and HEA are evaluated for their compression capabilities of TFI ground states. We also investigate using quantum simulation the impact of different system sizes, *i.e.* number of qubits, and classical optimizers on the models' performance.

## II. METHODOLOGY

### A. Models

**Hardware-Efficient Ansatz (HEA)** is a versatile quantum circuit design that can be adapted to various QML tasks. It typically consists of alternating layers of single-qubit rotations and entangling gates, tailored to the hardware's connectivity. Our preliminary benchmarks showed that for these datasets, the HEA with

RY gates obtained similar performances compared to the more general HEA with RX-RZ-RX gates but required only a fraction of the training time. Therefore, we focused on the HEA with RY gates for more detailed benchmark studies against QCNNs.

**Quantum Convolutional Neural Networks (QCNNs)** extend the concept of classical CNNs to quantum data. They consist of layers of quantum gates that perform convolution and pooling operations on quantum states. Inspired from the competitive performance of HEA with RY gates and together with the observation that these quantum states are real-valued, we hypothesized that real-valued quantum classifiers that are sufficiently expressible would be able to perform the classification task well. To test this hypothesis, we propose and investigate real variants of the original (complex-valued) QCNN with different levels of expressibility and inversely, the number of trainable parameters, leading to trade-offs between performance and training times.

### B. QML Tasks

**Phase Classification.** The phase classification task involves distinguishing between different quantum phases of the ground states of the TFI and XXZ model,

$$\mathcal{H}_{\text{TFI}} = - \sum_{j=1}^{N-1} \sigma_j^z \sigma_{j+1}^z - h \sum_{j=1}^N \sigma_j^x, \quad (1)$$

$$\mathcal{H}_{\text{XXZ}} = - \sum_{j=1}^{N-1} (\sigma_j^x \sigma_{j+1}^x + \sigma_j^y \sigma_{j+1}^y + h \sigma_j^z \sigma_{j+1}^z). \quad (2)$$

We adopt the conventional quantum classifier setup, in which the test state  $|\psi_{\text{test}}\rangle$  with label  $l = \pm 1$ , is passed into the trained model  $U(\theta)$  with optimized parameters  $\theta$ . The label predicted by the model  $p$ , is given by  $p = \text{sign} \langle \psi_{\text{test}} | U^\dagger(\theta) Z_0 U(\theta) | \psi_{\text{test}} \rangle$ . We use the mean squared error between  $l$  and  $p$  over the training dataset as the cost function (see Fig. 1).

**Data Compression.** For the data compression task, we design quantum autoencoders using QCNN and HEA

\* kongjf@ihpc.a-star.edu.sg

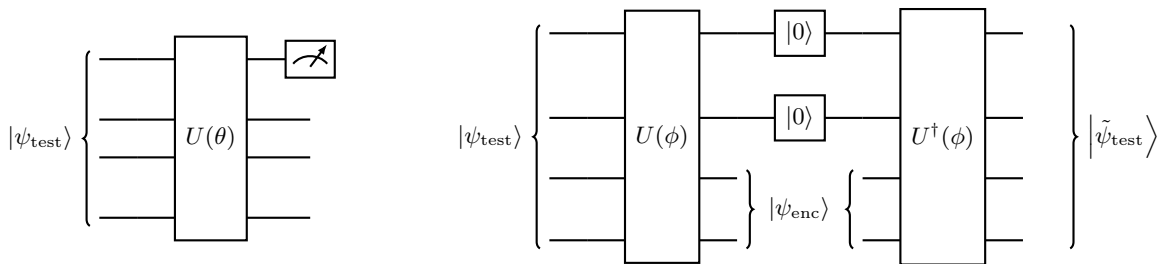


FIG. 1: Framework of quantum phase classification (left) and data compression (right) for 4 qubits.

architectures. These quantum autoencoders [4, 5] aim to compress input quantum states into lower-dimensional representations and then reconstruct the original states with minimal loss. Our setup for each autoencoder involves an encoding phase and a decoding phase. In the former, the test state  $|\psi_{\text{test}}\rangle$  is passed into the trained encoder  $U(\phi)$  with optimized parameters  $\phi$ , and finally applying reset gates to the qubits to be discarded to obtain the encoded state  $|\psi_{\text{enc}}\rangle$  of the remaining qubits. In the latter, the  $|\psi_{\text{enc}}\rangle$  is passed with the  $|0\rangle$  states of the discarded qubits into the trained decoder  $U^\dagger(\phi)$  to obtain the decoded test state  $|\tilde{\psi}_{\text{test}}\rangle$ . The quality of the encoding, can then be determined by the overlap  $|\langle \tilde{\psi}_{\text{test}} | \psi_{\text{test}} \rangle|^2$  (see Fig. 1).

In training of the autoencoder, only the encoder part needs to be trained, as the decoder circuit is the inverse of the encoder. The encoder is trained by minimizing the following cost function:

$$C = \frac{1}{2} \left( - \sum_{i \in \mathcal{N}_d} Z_i + n_d \right), \quad (3)$$

where  $\mathcal{N}_d$  and  $n_d = |\mathcal{N}_d|$  is the set and number of discarded qubits respectively. Its global minimum occurs when the output state is  $|0\rangle^{\otimes n_d} \otimes |\psi_{\text{enc}}\rangle$ , indicating that the discarded qubits can be successfully set to  $|0\rangle^{\otimes n_d}$  and  $|\psi_{\text{test}}\rangle$  can be compressed to  $|\psi_{\text{enc}}\rangle$ .

### C. Simulation Setup

For the classification task, we conducted experiments with system sizes of 4, 8, and 16 qubits. The training was done using the Qibo library [6] on Nvidia A100 GPUs. For 4- and 8-qubit systems, we generated labelled datasets of groundstates by performing exact diagonalization to solve the TFI and XXZ models, Eqs. (1) and (2) respectively, for a range of values of  $h$  across their respective phase boundaries. For the 16-qubit case, we used optimized variational circuits provided in the tensorflow-quantum repository [7].

For the data compression task, five input states of 4 and 8 qubits corresponding to the ground states of the TFI model, across the phase boundary of  $h = 1$ , were used. QCNN architecture is naturally suited for data compression as each successive layers of QCNN downscales the number of qubits by half, thus  $n_d = N(1 - 1/2^l)$ , there  $l$  is the number of QCNN layers. For

HEA, there are no fixed requirement on downscaling, and hence  $n_d$  can be varied from 1 to  $N - 1$ . With the trained encoder  $U(\tilde{\phi})$ , the decoder can be constructed by the inverse of the encoder circuit  $U^\dagger(\tilde{\phi})$ .

## III. RESULTS

Our results in Fig. 2 indicate that QCNN with RY gates showed competitive performance compared to the best performing HEA, with the added benefit of a dramatic reduction in training times due to substantially fewer trainable parameters. For the same number of trainable parameters therefore, QCNN with RY gates is the most efficient architecture for classifying these quantum ground states. The relatively poorer performance of the more expressible QCNN variants can be attributed to trainability issues. HEAs on the other hand, show an increasing performance with expressibility (i.e. number of layers), as well as a corresponding increasing running time per training data (approximately linearly with number of layers). A more detailed analysis was also performed by generating the receiver operating characteristic curve and computing the area under curve.

The autoencoder architectures were assessed on their ability to compress and reconstruct quantum states. Firstly, we note that the generally high reconstruction fidelities imply that all the evaluated models are capable of compressing TFIM ground states to a smaller number of qubits. However, QCNNs demonstrated faster training convergence, due to the lower number of trainable parameters in the QCNN models. This demonstrates one key advantage of the QCNN architecture for this task — lower number of trainable parameters leading to faster convergence and lesser trainability issues, with minimal tradeoff in compression capability.

## IV. DISCUSSION AND CONCLUSION

This study highlights the effectiveness of QCNNs, particularly those limited to RY gates, in quantum phase classification and data compression tasks. Consistent across both tasks, we find that QCNNs not only achieve similar performance to HEAs but also benefit from shorter training times due to their simpler structure. The choice of classical optimizer significantly impacts the training efficiency and final model performance. Furthermore, during our simulations we observed that Powell is one of the most reliable optimizers for these tasks. The simulation results obtained will subsequently be compared against implementation on hardware using the QiboLab software library [8].

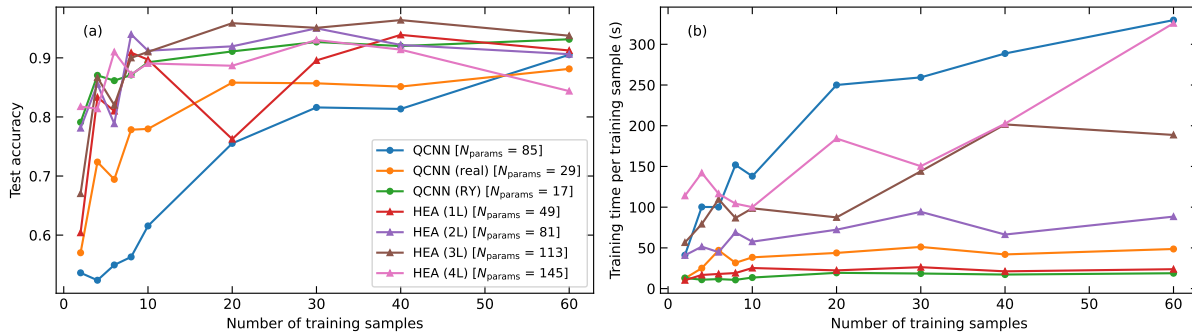


FIG. 2: The results of (a) test accuracy for 16 qubits as a function of number of training samples, with their respective training time per training sample shown in (b). In the case of HEA,  $nL$  denotes  $n$  layers. The number of parameters ( $N_{\text{params}}$ ) for each model are shown as well.

Best Models	Test Accuracy	Training Time / sample	$N_{\text{params}}$
QCNN (RY)	0.931	18.7s	17
HEA (3L)	0.938	188.7s	113

TABLE I: Performance metrics for best QCNN and VQC models for  $N_{\text{params}} = 60$ .

- 
- [1] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, December 2019.
  - [2] Tak Hur, Leeseok Kim, and Daniel K Park. Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, 4(1):3, February 2022.
  - [3] Arthur Pesah, M. Cerezo, Samson Wang, Tyler Volkoff, Andrew T. Sornborger, and Patrick J. Coles. Absence of barren plateaus in quantum convolutional neural networks. *Phys. Rev. X*, 11:041011, Oct 2021.
  - [4] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, aug 2017.
  - [5] Carlos Bravo-Prieto. Quantum autoencoders with enhanced data encoding. *Machine Learning: Science and Technology*, 2(3):035028, jul 2021.
  - [6] Stavros Efthymiou, Sergi Ramos-Calderer, Carlos Bravo-Prieto, Adrián Pérez-Salinas, Diego García-Martín, Artur Garcia-Saez, José Ignacio Latorre, and Stefano Carrazza. Qibo: a framework for quantum simulation with hardware acceleration. *Quantum Science and Technology*, 7(1):015018, dec 2021.
  - [7] Michael Broughton, Guillaume Verdon, Trevor McCourt, Antonio J. Martinez, Jae Hyeon Yoo, Sergei V. Isakov, Philip Massey, Ramin Halavati, Murphy Yuezhen Niu, Alexander Zlokapa, Evan Peters, Owen Lockwood, Andrea Skolik, Sofiene Jerbi, Vedran Dunjko, Martin Leib, Michael Streif, David Von Dollen, Hongxiang Chen, Shuxiang Cao, Roeland Wiersema, Hsin-Yuan Huang, Jarrod R. McClean, Ryan Babbush, Sergio Boixo, Dave Bacon, Alan K. Ho, Hartmut Neven, and Masoud Mohseni. Tensorflow quantum: A software framework for quantum machine learning, 2021.
  - [8] Stavros Efthymiou, Alvaro Orgaz-Fuertes, Rodolfo Carobene, Juan Cereijo, Andrea Pasquale, Sergi Ramos-Calderer, Simone Bordoni, David Fuentes-Ruiz, Alessandro Candido, Edoardo Pedicillo, Matteo Robbiati, Yuanzheng Paul Tan, Jadwiga Wilkens, Ingo Roth, José Ignacio Latorre, and Stefano Carrazza. Qibolab: an open-source hybrid quantum operating system. *Quantum*, 8:1247, February 2024.