ENFORCING EXACT PERMUTATION AND ROTATIONAL SYMMETRIES IN THE APPLICATION OF QUANTUM NEURAL NETWORK ON POINT CLOUD DATASETS

Zhelun Li, Lento Nagano, Koji Terashi
International Center for Elementary Particle Physics (ICEPP),
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
E-mail address: `zhelunli@icepp.s.u-tokyo.ac.jp`

Keywords Quantum machine learning · quantum neural networks · equivariant quantum networks

arXiv link: `https://arxiv.org/abs/2405.11150`

1 Introduction

The study of equivariant Quantum Neural Networks (QNNs) has attracted much attention in the quantum computing community. Several recent papers have explored the possibility of encoding problem-specific symmetries into the QNNs. A well-known example is the Quantum Convolutional Neural Networks (QCNNs), which realizes the translational symmetry in the QNN architecture by introducing quantum convolution layers made of quasi-local unitary transformations applied in a translationally invariant fashion [1]. Despite the success in equivariant QNN, the realization of arbitrary symmetries, especially continuous symmetries, in QNN is still considered a challenging task. Since the convolution-based method requires the model to perform convolutions across all group elements, its application in datasets with continuous symmetry inevitably runs into difficulties induced by the infinite number of elements in continuous groups. Furthermore, the traditional convolution-based treatment in classical ML requires the use of complicated convolutional layers targeting a discretized approximation, often realized by Fourier transforms, of the continuous symmetry. It is still unclear how a QNN could replicate these architectures using hardware-efficient ansatz consisting of single-qubit rotations and 2-qubit entanglements.

In this work, we aim to design a new framework of equivariant QNNs that is compatible with near-term quantum machines. We take an alternative approach to enforce exact rotational symmetries in point cloud datasets, which consist of collections of data points in arbitrary dimension spaces. To achieve the exact equivariance, we invoke Weyl's work [2], which asserted that all invariant functions of the orthogonal groups $O(N)$ with vector inputs could be expressed as functions solely dependent on the group-invariant inner products of these vectors. This key insight enables us to perform simple data preprocessing to obtain $O(N)$-invariant inputs to a network that is therefore guaranteed to be exactly $O(N)$-invariant at the input level. This approach was first developed by Villar et al. [3] who proved that this simple method involving only pair-wise scalars could significantly improve the performance of NNs. After taking the inner products, we proceed to use the twirling method to achieve permutation equivariance in our QNN. The network with both rotational and permutation symmetries is numerically benchmarked in the tasks of 2D image classifications and high-energy particle identifications against a baseline QNN without any symmetry.

2 Quantum neural network architectures

For a typical point cloud dataset, each input data consists of a series of vector points, the value of which indicates the geometrical positions of objects in space. In this paper, we denote a vector in the point cloud by $\vec{p}$. To use these inputs for our baseline QNN, we first flatten the input into a one-dimensional array, $\{p_0^0, p_0^1, p_1^0, p_1^1, ..., p_n^0, p_n^1\}$, and then encode these values in qubits using Z feature maps.

For the symmetric QNN, the rotational symmetry is guaranteed by Weyl's theorem [2], which states that for any rotationally invariant $I(p_1, ..., p_N)$, there always exists an equivalent rotationally invariant function $F(\{p_i \cdot p_j\})$ with pair-wise inner products as inputs:

$$I(p_1, ..., p_N) = F(\{p_i \cdot p_j\}_{i,j}). \tag{1}$$

The input to the symmetric QNN is then taken to be the inner products of point cloud vectors, which are intrinsic rotational invariants. To introduce the permutation equivariance, we use the twirling formula [4], which symmetrizes circuit blocks with respect to the group in the induced representation. The twirling formula requires the circuit to loop through all possible permutations of the quantum gates to guarantee the permutation equivariance.

Combining the twirling formula with the inner product inputs, we arrive at the fully symmetric QNN shown in Figure 1. The symmetric model uses much fewer variables compared to the baseline model and is thus less prone to the barren plateau phenomenon. The computation of inner products can be easily extended to other variants of the rotational group like $SO(1, 3)$, which is particularly important in the field of high-energy physics.

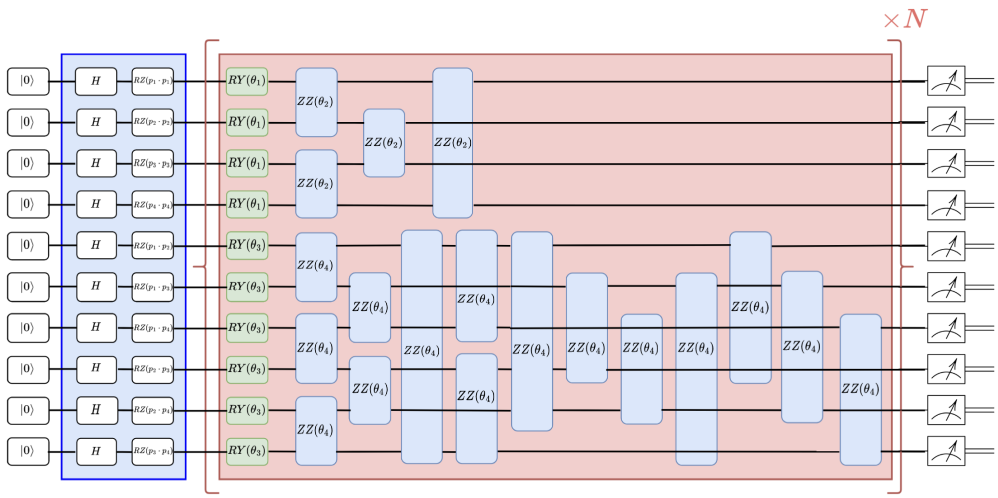


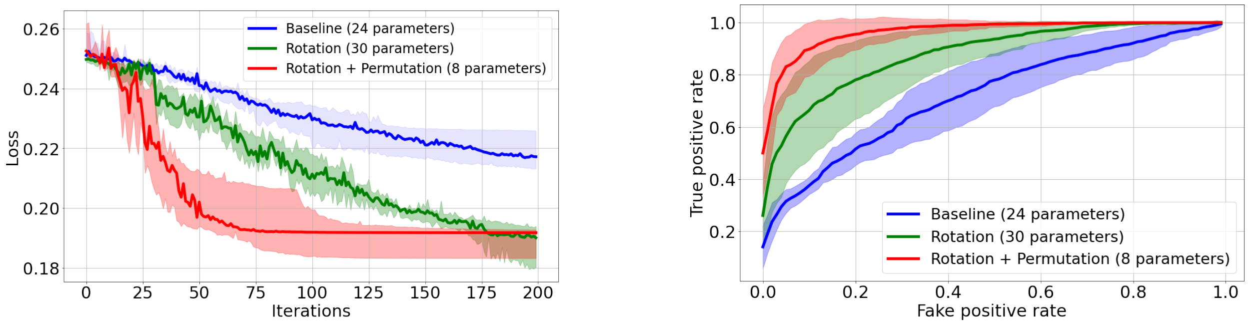
Figure 1: An example of a QNN setup with both rotational and permutation symmetry of four input vectors. The rotational symmetry is enforced by having pair-wise inner products as inputs, which requires $4 \cdot (4 + 1)/2 = 10$ qubits in the encoding process: $\{p_1 \cdot p_1\}, ..., \{p_3 \cdot p_4\}$. Each twirled operator is required to have the same parameter by the twirling method to achieve equivariance. Each ZZ gate takes exactly two inputs. If a horizontal line goes through a ZZ gate, it implies that the corresponding qubit is not one of the ZZ gate's inputs.

3 Results

In our paper, we consider two main tasks to benchmark our symmetric QNN model: the 2D image classification and the high-energy particle identification. Other than the fully symmetric model, We consider a hardware-efficient baseline model with no symmetry and a symmetric model with only rotational symmetry.

For the 2D image classification, we first create a dataset with rotated images of two similar geometrical objects: the square and the triangle. We consider various transformations to be applied to these templates, including translation, resizing, rotation, data point shuffling, and smearing. Overall, we created $1,600$ different images, each of which is a template undergone all five random transformations listed above. In this dataset, $1,200$ images are used as training data while the other 400 are used as testing data to verify performances. The loss functions of all three models during the training process are shown in Figure 2a. The Receiver Operating Characteristic (ROC) curves in the test dataset are shown in Figure 2b.

For the high energy particle identification task, we use the Higgs boson decay as the signal process, where a Higgs boson decays into four final state leptons through the ZZ^* channel: $H \to ZZ^* \to \ell^\pm \ell^\mp \ell^\pm \ell^\mp$. The background consists of four lepton events from various Standard Model processes excluding Higgs boson production. The task is to discriminate the signal process against the background. A total number of $6,000$ particle decay events are used in the training, while $2,000$ events are used for testing performances. For the baseline model, we use the four momenta of the final state leptons as inputs. To enforce rotational symmetry, we acknowledge that the $SO(1, 3)$ Lorentz symmetry also satisfies the theorem given in Equation 1 and thus use the pair-wise inner products between final state particles as inputs. We implement an ad-hoc nonlinearity function in the postprocessing to accommodate the nonlinear nature of the



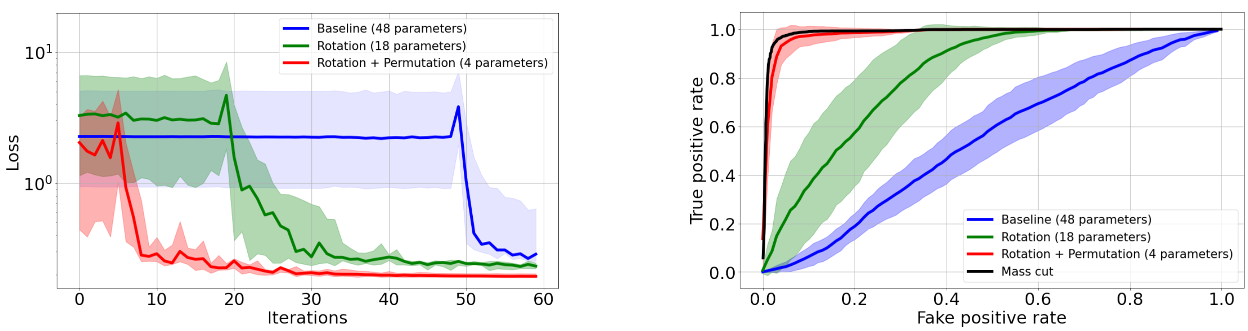
(a) Loss function vs iterations.

(b) ROC curve.

Figure 2: The Loss function during the training process and the ROC curve for various QNN models in the 2D image classification task. The loss values shown in the figure are the median values of all ten initializations, whereas the error bands are computed at 25% and 75% quantiles. The ROC curves are obtained by averaging the training results and the error bands are obtained by taking the standard deviations.

problem. Again, the training losses for all three models are shown in Figure 3a, while the ROC curves are shown in Figure 3b.

The symmetric model is observed to have faster converging speeds in the training processes of both tasks, as shown in Figure 2 and Figure 3. In terms of performances, the symmetric model yields Area-Under-Curve (AUC) values of 0.966 ± 0.030 and 0.988 ± 0.001 for the image classification and particle identification tasks respectively, while those of the baseline model are 0.720 ± 0.060 and 0.544 ± 0.060.



(a) Loss function vs iterations.

(b) ROC curve.

Figure 3: The Loss function during the training process and the ROC curve for various QNN models in the particle decay identification task. The curves are computed similarly as those shown in Figure 2. In the ROC curve, the QNN results are compared to the performance of the cut on 4-lepton invariant mass, which is regarded as the theoretical upper limit of performance.

References

[1] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, August 2019.

[2] H. Weyl. *The Classical Groups: Their Invariants and Representations*. Princeton landmarks in mathematics and physics. Princeton University Press, 1946.

[3] Soledad Villar, David W. Hogg, Kate Storey-Fisher, Weichi Yao, and Ben Blum-Smith. Scalars are universal: Equivariant machine learning, structured like classical physics. In *Neural Information Processing Systems*, 2021.

[4] Louis Schatzki, Martín Larocca, Quynh Nguyen, Frédéric Sauvage, and Marco Cerezo. Theoretical guarantees for permutation-equivariant quantum neural networks. *npj Quantum Information*, 10, 01 2024.