Research Article

OPTICA QUANTUM

Quantum optical reservoir computing powered by boson sampling

Akitada Sakurai,[1,4] Aoi Hayashi,[1,2,3,5] William John Munro,[1,*] and Kae Nemoto[1,6]

[1]Okinawa Institute of Science and Technology Graduate University, Onna-son, Okinawa 904-0495, Japan
[2]School of Multidisciplinary Science, Department of Informatics, SOKENDAI (the Graduate University for Advanced Studies), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
[3]National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
[4]Akitada.Sakurai@oist.jp
[5]aoi.hayashi@oist.jp
[6]kae.nemoto@oist.jp
*bill.munro@oist.jp

Received 5 September 2024; revised 24 April 2025; accepted 7 May 2025; published 28 May 2025

It is well known that boson sampling, a restricted non-universal quantum computation model, enables one to perform tasks that are hard to accomplish with digital computers. Boson sampling is associated with sampling the probability distribution of identical bosons passing through a random interferometer, and its quantum advantage has been demonstrated. It has, however, proven elusive to use such a model for practical applications. In this work, we show that the random interferometer powering boson sampling can be used to generate the complex dynamics necessary for quantum reservoir computing. We use these dynamics to perform various image recognition problems, illustrating the utility of the approach even for modest-size systems.

© 2025 Optica Publishing Group under the terms of the Optica Open Access Publishing Agreement

https://doi.org/10.1364/OPTICAQ.541432

1. INTRODUCTION

Quantum computation has reached an interesting stage in its development, where a computational advantage has been demonstrated over its classical counterpart in various physical systems [1–4]. That advantage is typically associated with sampling a probability distribution generated from some random quantum circuit [5–9]. Boson sampling, a restricted non-universal quantum computation model with #P-hard complexity [10], is one such example where N indistinguishable single photons are injected into an $M(>N^2)$ mode linear-optical interferometer with the task being to generate a sample from the probability distribution of single-photon measurements at the output of the circuit. As such, variants of boson sampling have been used to show a computational advantage over digital computation [2,4]. A critical and timely issue that needs to be resolved is whether non-trivial computational complexity can be used for practical tasks or applications [11].

Quantum reservoir computing (QRC) [12–23], quantum extreme learning machine (QELM) [24,25], quantum extreme reservoir computing (QERC) [26,27], and quantum neuromorphic computing (QNC) [28] are forms of quantum neural network models that are capable of solving hard learning tasks on both classical and quantum input data. Examples include time-series prediction [12–19], system identification, and image recognition and classification [13,21,26]. At the core of these approaches are complex reservoirs formed from the nonlinear nature of quantum mechanical processes or linear reservoirs with nonclassical inputs [19]. Boson sampling networks provide such a complex reservoir; thus, these reservoir computing models provide a route forward for practical applications. This work proposes a new quantum machine learning model using boson sampling to generate its quantum reservoir. The quantum linear optical extreme reservoir computing (QORC) model can effectively solve image classification problems in a regime where the number of modes is much greater than the number of single-photon inputs. It achieves higher classification accuracy rates using single-photon inputs than weak coherent states.

2. QUANTUM LINEAR OPTICAL EXTREME RESERVOIR COMPUTING

Quantum linear optical extreme reservoir computing is the optical analog of qubit-based QERC that uses the boson sampler as its physical reservoir (as illustrated in Fig. 1) [26]. At its core, the QORC device uses N photons and two $M>N^2$ random boson sampling interferometers (or the same one used twice). Those random interferometers [29] and encoding are not optimized depending on the results from the linear classifier. There is no back-propagation.

QORC begins by taking the data space $\mathbb{D}$ of the dataset one wants to use (images in this case) and, if necessary, uses techniques like principal component analysis (PCA) [30,31] to pull the key features out and reduce the large data size into a smaller, more manageable one while at the same time, maintaining its significant patterns and trends. For example, in the hand-written

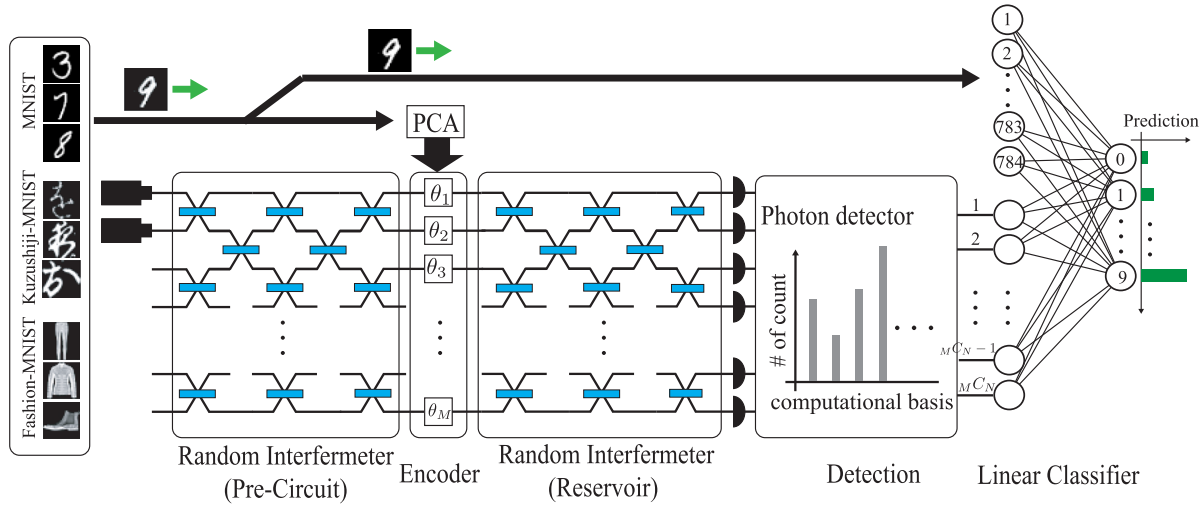

Fig. 1. Schematic diagram of QORC. It comprises several key components, beginning with dimensional reduction techniques (if necessary), such as principal component analysis (PCA), to reduce a large data set into a smaller one while maintaining its significant patterns and trends. This information is represented by the vector $\{\theta_1 - \theta_M\}$. Then, an M mode random interferometer (pre-circuit), which has N single-photon inputs, creates a complex photonic resource state that the Encoder uses to input the smaller data set information θ_i, $i \in \{0, 1, \ldots, M\}$ on to using linear optical elements. The resultant encoded photonic states are then passed through the Reservoir (an M random interferometer, which could be the same one used in the pre-circuit), followed by coincidence detection using bucket detectors at each mode. This is repeated several times to estimate the ratio of counts on a computational basis. That information is then fed into a linear classifier for optimization, and the resulting output classifies the image.

image classification, the original data space for one sample is $\mathbb{D} = \mathbb{R}^{28\times28}$, but after the PCA and standardization, it becomes $\mathbb{D}_{\text{PCA}} = \mathbb{R}^M$. The standardization is performed to fit each principal component to angles ranging from 0 to π. Then, data in the small set is represented by θ_i, $i \in \{0, 1, \ldots, M\}$. Following this, N single photons are input into an M-mode random interferometer—called the pre-circuit. This is used to create a complex photonic resource state $|\hat{\Psi}_{\text{Pre}}\rangle = \hat{U}_{\text{Pre}}|\text{In}\rangle$, where $\hat{U}_{\text{Pre}}$ represents an $M \times M$ Haar random unitary while $|\text{In}\rangle = \prod_{i=1}^{N} \hat{a}_i^\dagger|0\rangle$ is the input state to that interferometer with $\hat{a}_l^\dagger$ being the ith mode creation operator. Next, the data elements θ_i must be encoded on the resource state $|\hat{\Psi}_{\text{Pre}}\rangle$. We can now encode our data into this state. There are several choices, including phase and/or amplitude encoding. Here, we will focus on a phase encoding using the rotational operator $\hat{R}(\theta)$ with an angle θ that depends on the input. The encoded state expressed by

$$|\Psi(\theta_{l=1,\ldots,M})\rangle_{\text{R}} = \prod_{i}^{M} \hat{R}(\theta_i)\hat{U}_{\text{Pre}}|\text{In}\rangle \quad (1)$$

is then fed through the boson sampling random interferometer defined by $\hat{U}_{\text{BS}}$, resulting in the output state $|\text{Out}\rangle = \hat{U}_{\text{BS}}|\Psi(\theta)\rangle_{\text{R}}$.

To extract the data from this optical circuit, we use bucket detection to record the number of events per mode and post-select on measuring N detector clicks to sample the coincidence rate distribution in the computational basis. The dimensionality of this distribution ${}_M C_N = M!/(N!(M-N)!)$ is much larger than the number of modes M. Next, the sampled coincidence rate distribution can be regularized to give a probability distribution with a mean value of zero and a variance of one. The resulting estimated probability distribution is then fed into our linear classifier that uses the single-perceptron model with multiple classes [32]. In the original QERC, the standardized distribution is an input for the linear classifier. In contrast, the QORC's linear classifier also accesses the original images to compensate for the information loss at PCA. In other words, QORC can, in principle, solve linear separation problems in the same way as a linear classifier [33], and the quantum part is used to expand the feature space to solve the nonlinear separation task. Consider an original input $\vec{x}$ and output from the reservoir, $\vec{u}(\vec{x})$, the input for the linear classifier is given by $\vec{y} = (\vec{x}, \vec{u}(\vec{x}))$. Our linear classifier is optimized with the AdaGrad algorithm [34] with a learning rate $\eta = 0.05$ (batch size = 100) and 100 epochs.

3. SIMULATION RESULTS

Our QORC model is ideally suited to the classification of images. As such, we will explore its performance for three well-known datasets of increasing hardness, that is, MNIST [35], Kuzusiji-MIST (K-MNIST) [36], and Fashion-MNIST [37]. However, we need to choose the number of single photons N and the number of modes M that will be used, noting that we ideally want to keep $M > N^2$. In Table 1, we summarize our performance on those three datasets for three photons with varying numbers of modes. Further, as a comparison, we also present the shallow machine learning models: Linear SVC (L-SVC) and the SVC with radial basis function(RBF) kernel [38,39]. In Fig. 2(a), we also compare our model with the Random Fourier Features (RFF), which is known as the fast approximate model of the RBF. Here, we note that to be a fair comparison between RFF and QORC, we used the modified model whose features consist of the random Fourier features and the image as similar to the QORC. The details of datasets and other models are discussed in Appendix A.

What becomes clear from our results is that as the system size increases, QORC's performance improves and exceeds L-SVC in all datasets. This indicates that the optical network effectively expands the feature space in high-dimensional spaces, leading to better performance than a linear classifier. As shown in Fig. 2(a), our QORC model behaves like the RFF. This indicates that our

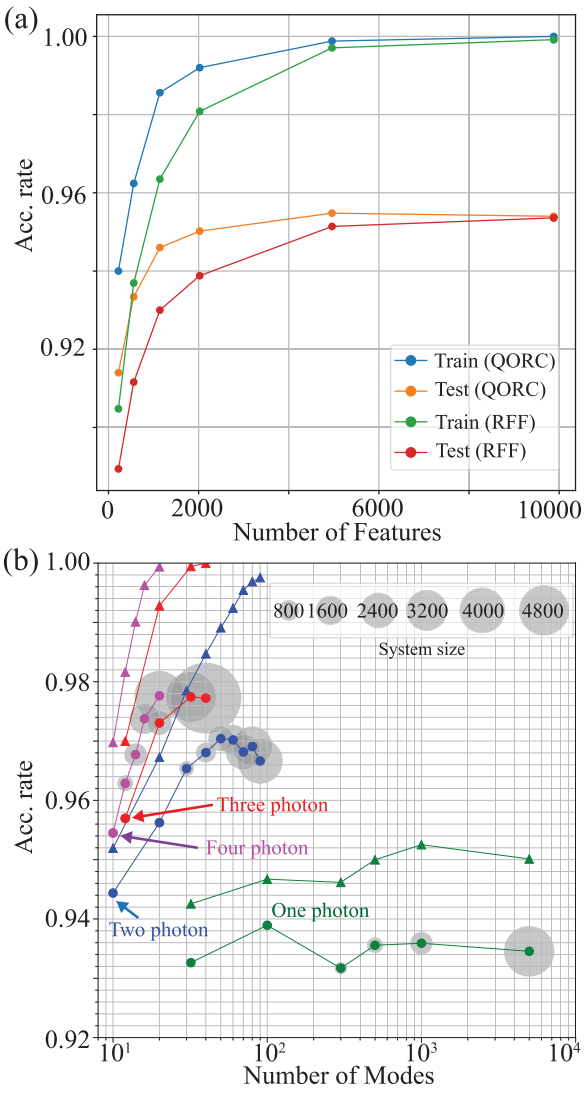

Fig. 2. QORC performance in solving the MNIST dataset. (a) Performance versus the different number of features. The blue and orange lines represent QORC with three photons. The green and red lines represent the RFF. To calculate the RFF, we used a normal distribution having $\sigma = 10$. (b) Performance versus different system sizes. The green, blue, red, and purple represent the cases of one, two, three, and four photons, respectively. The triangles/dots indicate the training/testing results, respectively. The light-shaded circles on the testing lines represent the boson sampler's output dimension (system size).

model works as an approximate model of the kernel model, similar to the RFF. Our model's performance is comparable to other shallow machine learning models in all datasets, with results approaching those of the RBF kernel model, which has an infinite number of neurons in its hidden layer, but does not appear in the actual calculation.

We summarize the insights obtained from comparing QORC, RBF, and RFF. The RBF kernel-based SVM achieves high test performance through margin maximization. However, given a training dataset of size D, its optimization cost scales as $O(D^3)$, and its inference cost as $O(D)$ [38]. As a result, even for datasets of the scale of MNIST, the computational cost becomes prohibitively high, making it unsuitable for modern big data

Table 1. Classification Performance for Different Datasets of Increasing Hardness (MNIST, K-MNIST, and Fashion-MNIST) for (*N*Photons, *M*Modes) Using the Same Boson SamplingLinear Optical Circuit[a]

(N,M)	Dataset		
	MNIST	K-MNIST	Fashion-MNIST
(3,12)	0.9700 ± 0.0002	0.9310 ± 0.0003	0.8938 ± 0.0005
	0.9569 ± 0.0005	0.8176 ± 0.0011	0.8662 ± 0.0006
(3,16)	0.9853 ± 0.0003	0.9541 ± 0.0003	0.9050 ± 0.0004
	0.9688 ± 0.0004	0.8479 ± 0.0014	0.8750 ± 0.0010
(3,20)	0.9896 ± 0.0002	0.9663 ± 0.0004	0.9097 ± 0.0006
	0.9705 ± 0.0003	0.8643 ± 0.0010	0.8766 ± 0.0019
(3,24)	0.9953 ± 0.0002	0.9795 ± 0.0007	0.9173 ± 0.0003
	0.9753 ± 0.0007	0.8857 ± 0.0016	0.8801 ± 0.0020
(3,32)	0.9996 ± 0.0001	0.9932 ± 0.0003	0.9297 ± 0.0009
	0.9784 ± 0.0004	0.8991 ± 0.0011	0.8826 ± 0.0025
(3,40)	1.0000 ± 0.000	0.9978 ± 0.0002	0.9383 ± 0.0008
	0.9783 ± 0.0005	0.9013 ± 0.0016	0.8769 ± 0.0055
L-SVC	0.9262	0.835	0.873
	0.918	0.676	0.839
RFF	1.0000	0.9954 ± 0.0002	0.9337 ± 0.0006
	0.9713 ± 0.0002	0.8438 ± 0.0011	0.8752 ± 0.0020
RBF	0.999	1.000	0.972
	0.984	0.929	0.900

[a]The upper value represents training accuracy, and the lower value represents testing accuracy. In the QORC, since we used the perceptron model to see the optimization stability, we show the average (standard deviation) taken from 90 to 100 epochs. The L-SVC, RFF, and RBF models are also presented for comparison. Here, the RFF has 2024 features like the QORC with 24 modes. For the QORC simulation, we used the ideal probability distribution. In other words, sampling effects are not included.

analysis. In contrast, RFF approximates the RBF kernel, maintaining comparable performance while reducing optimization and inference costs to $O(D)$ and $O(1)$, respectively. However, RFF requires the generation of large random matrices to map data into a high-dimensional space. QORC, however, leverages the non-trivial probability distribution of boson sampling to achieve similar performance with significantly smaller optical circuits relative to the number of required feature mappings. As will be discussed later, even when considering the required number of shots, the computational cost of QORC remains lower than that of an equivalently scaled RFF.

So far, we have only explored several N, M cases, and so in Fig. 2(b), we present the impact of the number of modes and photons on performance.

The results show that increasing the number of photons can reduce the number of modes required to achieve the same level of performance observed in cases with fewer photons. This performance gain in a larger system is due to the growth of its Hilbert space (feature space) rather than adding more PCA components, as we have used the same number of components and modes for our analysis.

The higher-order components of the PCA used in our model include low-frequency components where image noise effects are prominent. In machine learning, training with noisy data increases the risk of overfitting. Thus, in our model, encoding components above a certain order into the optical circuit is also expected to result in overfitting due to noise effects. In fact, in the two-photon case, overfitting starts to occur when more than 50 modes (up to 50 components) are used. However, this

overfitting can be reduced by excluding the low-frequency components. Specifically, when encoding up to 45 components in the two-photon, 90-mode case, the test performance recovers from approximately 96.7% (in the overfitting scenario) to 97.4%.

It is critical to explore the sampling cost of QORC as it has a major impact on its feasibility. Our model uses the probability distribution as the output from the physical reservoir, and to reconstruct it, a certain amount of sampling is required. We perform a numerical simulation with the results shown in Fig. 3(a) to see how the performance changes with the number of samples. It shows that the performance convergence speeds are prolonged, meaning many samples are required to achieve performance close to the performance obtained using theoretical distributions. However, our results show that the convergence tendencies are the same, although their size changes. Increasing the number of modes makes it possible to reduce the sampling cost and achieve target accuracy.

To investigate how the number of shots of the QORC scales with the Hilbert space dimension of the quantum system used, we evaluate the absolute difference between the test performance based on the theoretical probability distribution and that based on the empirical distribution constructed from finite shots. Specifically, we plot this absolute difference against a rescaled axis using the number of shots and the dimension of the probability distribution after the post-selection. The number of shots N_s follows the general estimation scale of $1/\sqrt{N_s}$, while the dimensionality of Hilbert space is considered under scaling power α. We compared several α, and found that when $\alpha = 1/4$, all data points are along the same curve. From these results, we conclude that when the absolute difference between the test performance of the theoretical and empirical distributions is fixed, the required number of shots scales as $\sqrt{(N_b)}$. Furthermore, we have observed the same tendency in other datasets as well.

The fact that the scale of the number of samples is not the dimension of the Hilbert space but its square root is an interesting result given previous studies of QRC and QELM [40]. In prior studies, the estimator the model requires is the expected value of the observables. Adapting this to our model, the estimator is a probability distribution (vector). Therefore, even intuitively, it would seem that we would need a sample size of approximately the dimension of the Hilbert space. However, the estimator we need for our model is not a probability distribution, but a similarity (scalar quantity), or inner product of probability distributions, by viewing our model as a kernel model. This inner product is not explicitly dependent on the dimension of the Hilbert space, so the number of samples required may differ from that in the case of probability vectors. Our results will improve our understanding of the required samples in QRC and QELM models.

Finally, we investigate the overall optimization and inference costs with the QORC using N photons and M modes, while L is the image size. In QORC, we use PCA for the dimensional reduction from L to M. The optimization cost (dominant cost) of PCA using randomized SVD is $O(DLM)$ [41] for training, where we used $M \ll \min\{L, D\}$ and D is the (training) dataset size, while the cost of the dimensional reduction for the testing is $O(LM)$. We consider the total number of gates involved in the quantum process to evaluate the computational cost of the linear optical circuit. Each shot through the circuit requires $O(M^2)$ operations. As the number of shots required scales as $O(\sqrt{N_b})$, the overall computational cost of the linear optical component is $O(M^2\sqrt{N_b})$ per image. Then, the total number of features for the QORC is given by $N_f = {}^M C_N + L$. The cost of the linear

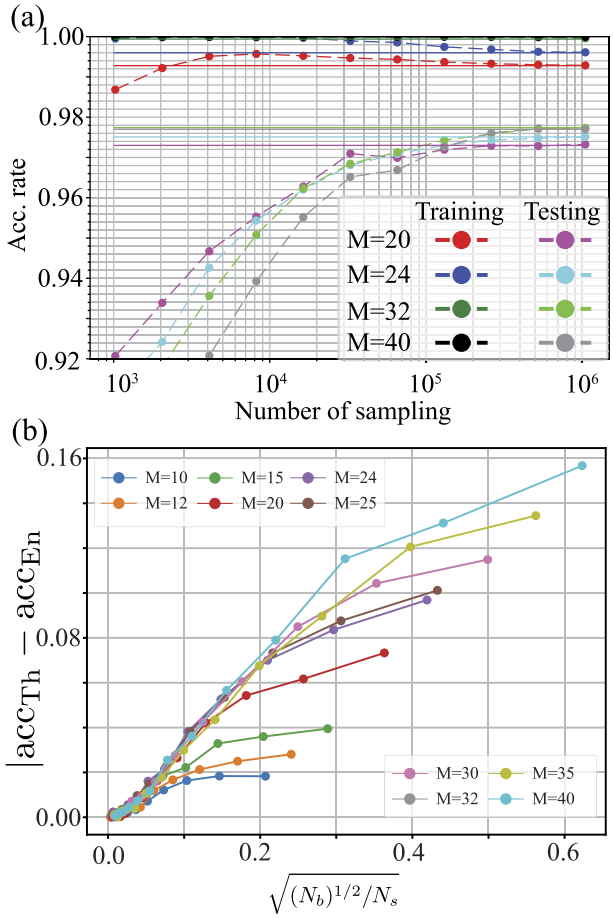

Fig. 3. (a) Plot of the accuracy rate versus the QORC sampling cost. The colored horizontal lines show the training and testing accuracies using theoretical probability distributions. Performance evaluation is conducted using the final accuracy instead of averaging over epochs for a realistic implementation, as shown in Table 1. Furthermore, to mitigate overfitting caused by noise from finite shots, the number of optimization epochs is set to 50, applying to both theoretical and empirical ones. (b) Plot of the absolute difference between the test performance using the theoretical probability distribution and the test performance using the empirical distribution (constructed from shots) versus $\sqrt{N_b^{1/2}/N_s}$ with N_s being the number of samples and N_b the Hilbert space dimension.

classifier is similar to that of RFF. Therefore, the total optimization cost is $O\left(D(LM + M^2\sqrt{N_b} + N_f T)\right)$, and the inference cost is $O\left(LM + M^2\sqrt{N_b} + N_f\right)$ per image, where T is the number of epochs.

It is important to establish the cost of QORC compared with the cost of RFF having the same number of features, such as N_f. We note that we consider the modified RFF discussed in Appendix A. The computation cost of the feature map per image is $O\left((N_f - L)L\right) = O\left({}^M C_N L\right)$. The optimization cost for the linear classifier is $O\left(DN_f T\right)$. Thus, the total optimization cost is $O\left(D({}^M C_N L + N_f T)\right)$, and the inference cost is $O\left(N_f + {}^M C_N L\right)$ per image. Here, we assume that the order of the total number of epochs is the same as that of the QORC. In the comparison, we assume that when N_f is sufficiently large, the linear classifier determines the dominant scaling of inference cost in both methods, resulting in $O(N_f)$. We further assume $N_b \approx N_f$ as the

image size is fixed and N_b is sufficiently large. Thus, no clear advantage of using a quantum system is observed at this stage. However, in random feature models that use fixed features, the feature computation cost—one of the significant computational bottlenecks—has been improved from $O(N_f)$ to $O(\sqrt{N_f})$, achieving a quadratic speedup. This suggests that for sufficiently large N_f, the benefit of using a quantum system becomes apparent.

Our results suggest a potential avenue for reducing computational costs, particularly in the context of the input dimensionality of the linear classifier. Currently, the overall computational cost scales with N_f, as the input to the classifier consists of N_f features. Since the quantum model encodes feature representations as probability distributions, this implies that each image contains N_f feature components. However, given that the number of measurement shots is proportional to the square root of the number of features, the effective number of contributing features may also be of the order of $\sqrt{N_f}$.

In classical two-layer neural networks, the ReLU activation function suppresses negative inputs by setting them to zero, effectively blocking the flow of information. This results in a computational cost reduction, as the subsequent layer only needs to process the activated neurons. However, a key limitation of this approach is that the locations of the non-activations are unknown in advance, necessitating computations for all $\sqrt{N_f}$ features in the preceding layer.

Our findings indicate that, with an appropriately designed post-processing method leveraging the probability distribution of the quantum system, it may be possible to reduce the effective number of features input to the linear classifier to approximately $\sqrt{N_f}$. This suggests a promising direction for future research in optimizing the computational efficiency of quantum-enhanced machine learning models.

Recently, it has been reported that in random feature models such as RFF, when the number of features exceeds the size of the training dataset, generalization performance improves, in contrast to the predictions of the bias-variance theory regarding overfitting [39]. This phenomenon, known as double descent, has led to reconsidering the relationship between model complexity and generalization performance in machine learning [42,43]. Following this discovery, two-layer neural networks, which had previously received little attention due to their inferior performance compared with convolutional neural networks (CNNs), are now being re-examined in theoretical and experimental contexts [39,43,44]. Our proposed QORC maintains performance comparable to RFF while reducing the feature computation cost to a square root scaling. This characteristic suggests that our approach has the potential to contribute significantly to future large-scale model experiments and theoretical analyses.

4. COMPARISON WITH COHERENT STATES

It is critical to provide a performance comparison to understand whether quantum effects using single photons provide our observed advantage. Using post-selection techniques, we replace our single photons with coherent states [45,46]. In that case, our input state would be $|\mathrm{In}\rangle = \prod_{l=1}^{N} \hat{D}_l(\alpha)|0\rangle$, where $\hat{D}_l(\alpha)$ is a displacement of size α at the lth mode. After passing through the phase encoding step and the boson sampler, it is well known that our output state $|\mathrm{Out}\rangle_C$ is simply a tensor product of coherent states in the different modes. No entanglement is generated during the entire process. It is thus interesting to determine the performance of such a scheme.

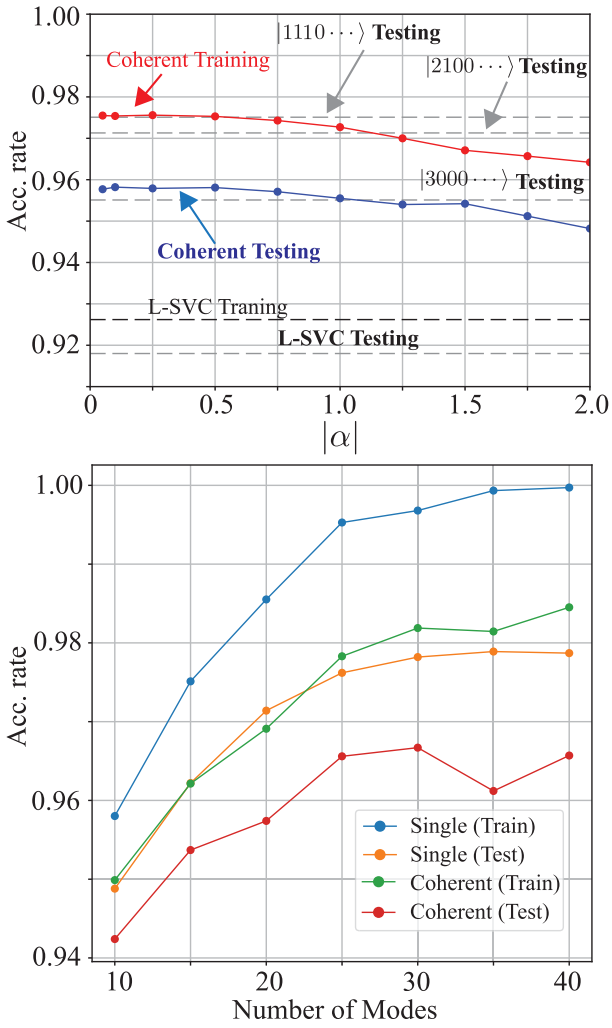

Fig. 4. QORC performance comparison between coherent and single-photon sources. (a) QORC performance using initial coherent states with amplitude α in the (3,24) arrangement. For all α, we use the same random interferometer in both the pre-circuit and the reservoir. The red and blue lines represent the training and testing results. We plot the performance of the L-SVC. We also plot the testing performance of the single-photon sources $|111\cdots0\rangle$, $|300\cdots0\rangle$, and $|210\cdots0\rangle$. (b) Comparison of the performance of both sources with different modes. Here, the number of photons is three.

Figure 4 displays the coherent input state's performance results versus amplitude α. We immediately observe that the coherent state case is always worse than the single-photon situation, but always better than using the linear classifier alone. That shows the coherent state is useful. Further, a significant decrease in the coherent state case performance occurs for $\alpha \geq 1$ due to higher photon number states in the coherent state and our post-selection based on bucket detection.

Additionally, we evaluated the classification performance of three-photon detection using a coherent light source across different mode numbers. As shown in Fig. 4(b), increasing the mode number up to 16 leads to a monotonic improvement in accuracy, reaching 96.7%. However, beyond this point, further

increasing the mode number results in only minor fluctuations, with overall accuracy remaining nearly constant. Evidently, the performance achieved with a coherent light source is lower than that obtained with a single-photon source and does not reach the accuracy level of classical RFF. Furthermore, we analyzed the case of four-photon detection using a coherent light source. Despite the increased Hilbert space dimension, the test accuracy remains lower than that of the single-photon source and saturates at 96%, which is even worse than the three-photon case. These results indicate that incorporating a coherent light source into a random interferometer does not produce a feature mapping comparable to RFF. To achieve a similar level of performance, a single-photon source is necessary.

The changes in performance between single photons and coherent states arise from the difference in the feature space size (the network for phase scrambling). This can be seen in the final state before the measurement. Here, let us consider an $M \times M$ optical circuit ($M \times M$ unitary matrix $\hat{U}$ having an element $U_{i,j}$ at the ith row and jth column). For an initial three single-photon input state, the final output state after all optical circuits is given by

$$|\mathrm{Out}\rangle = \sum_{l_1,l_2,l_3=1}^{M} W_{l_1,l_2,l_3} \hat{a}^\dagger_{l_1} \hat{a}^\dagger_{l_2} \hat{a}^\dagger_{l_3} |0\rangle, \tag{2}$$

where

$$W_{l_1,l_2,l_3} = \sum_{j_1,j_2,j_3=1}^{M} U_{1,j_1} U_{2,j_2} U_{3,j_3} U_{j_1,l_1} \times U_{j_2,l_2} U_{j_3,l_3} e^{i(\theta_{j_1}+\theta_{j_2}+\theta_{j_3})} \tag{3}$$

with $\theta_1 \ldots \theta_M$'s encoding the phase information from the PCA. Now, Eq. (3) corresponds to our feature maps and highlights how boson sampling with its #P-hard complexity can generate a non-trivial one with an enormous combination of phases and their scrambling. The determination of W_{l_1,l_2,l_3} corresponds to a permanent calculation.

However, using coherent state inputs results in the final state

$$|\mathrm{Out}\rangle_C = \prod_{l=1}^{M} \hat{D}\left(\alpha \sum_{j=1}^{M} \sum_{i=1}^{3} U_{j,l} U_{i,j} e^{i\theta_j}\right) |0\rangle. \tag{4}$$

Although the coherent state's product form makes the probability distribution's size equivalent to that of single photons, the dimensionality of the scrambling network is limited to the number of modes (M). Thus, single photons can compute vast combinations and complex mixtures of classical information. In contrast, coherent light is constrained to a much smaller space, resulting in significantly different feature mappings for image classification.

Another interesting case arises when we consider an n photon state input only on a single mode (as shown in Fig. 4 for $n = 3$). We immediately observe that the performance is worse than the three single-photon case and, in fact, has its performance close to the coherent state case. This is not unexpected, however, as the $n = 3$ photon state input looks quite similar to a coherent state input on a single mode when post-selection is included. That hints at the complexity of the feature maps with an $|111\rangle$ input being quite different from the $|300\rangle$ case.

5. CONCLUDING DISCUSSION

Boson samplers, which are associated with sampling the probability distribution of identical bosons passing through a random interferometer, are known to be extremely hard to accomplish with classical computers. These random interferometers, which use nonclassical light, generate complex quantum dynamics that is likely useful for other quantum tasks. This work shows how complex dynamics can be used to generate the quantum reservoir at the core of various computing models, including QRC, QERC, and QNC. We use the dynamics to perform various image recognition problems, illustrating the approach's utility even for a few-photon systems. In particular, we show that QORC outperforms similar approaches using coherent light inputs and linear support vector classifiers. As the size of these QORCs increases, they should be able to perform classification tasks that are challenging in today's machine learning world. Finally, our boson sampling here has been based on single-photon schemes but should also work with other encodings.

APPENDIX A: DATASET AND CLASSICAL MODELS

To benchmark our model, we solve hand-written digit images known as the MNIST dataset [35], in which each image has 28×28 pixels, 60,000 images, and 10,000 images for training and testing. The dataset is widely used as a benchmark for machine learning models, including physical neural networks such as reservoir computing. Additionally, we consider two more datasets, Kuzusiji-MNIST (K-MNIST) [36] and Fashion-MNIST [37] datasets, in which images are of Japanese characters and clothes, respectively. Both datasets have ten classes, and 60,000 images and 10,000 images for training and testing, the same as the MNIST. As the pre-processing, we apply the Min-Max normalization to all datasets [47]. It is assumed that the difficulty among the three datasets is [36,37]: MNIST < K-MNIST < Fashion-MNIST.

Several classical models have been proposed to solve image classification tasks. For instance, the convolutional neural network (CNN) [48,49] and Vision Transformer [50] are known as state-of-the-art image classifiers. It is still a goal to achieve the same level as quantum models, but so far, comparing our models with them is unfair for the following reasons. First, they have a much more complicated network structure to catch image features and more parameters to optimize. In contrast, our model needs to be more complex. Second, they require pre-training and fine-tuning. The pre-training is done with big data and is not used in our model.

It is important to compare our model with classical models with similar structures and computational resources. Our model has a linear classifier at the end, so we consider the linear support vector classifier (L-SVC) [51]. Linear SVC is a Support Vector Machine (SVM) type that performs classification on linearly separable data. We introduce a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is an input and $y_i \in \{-1, +1\}$ is a class label. Linear SVC aims to find a hyperplane that separates the classes with the maximum margin. This is achieved by solving the following optimization problem:

$$\min_{\mathbf{w},b,\eta} \frac{1}{2}\mathbf{W}^\top \mathbf{W} + C \sum_{i=1}^{n} \xi_i \tag{A1}$$

subject to the constraints given by

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0. \tag{A2}$$

Here, $\mathbf{w}$ is the weight vector, b is the bias, ξ_i are slack variables, and C is a hyperparameter that controls the trade-off between maximizing the margin and minimizing classification errors. Linear SVC is particularly effective in high-dimensional feature spaces and is widely used in applications such as text classification and face recognition. Although SVMs are fundamentally binary classifiers, Linear SVC can be extended to handle multiclass classification. In our calculation, we used the one-versus-rest method. In this method, for K classes, we train K binary classifiers, where each classifier distinguishes one class from the rest. The final prediction is made by choosing the class whose classifier outputs the highest confidence score.

Next, as our model is similar to the kernel model, we also compare our models with Kernel SVC. We use the radial basis function (RBF) [38,39] kernel. In the RBF kernel, the kernel function is defined by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right). \tag{A3}$$

This allows the SVC to construct a nonlinear decision boundary in the original feature space by implicitly mapping the data into a higher-dimensional space. Here, γ controls the influence of a single training example. Low values imply 'far' influence, high values imply 'close' influence. Using the RBF kernel, SVC can effectively handle complex, nonlinear classification problems. Now, using the kernel function, the optimization problem can be formulated in its dual form, as follows:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{A4}$$

subject to the constraints given by

$$0 \le \alpha_i \le C, \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{A5}$$

Here, C controls the regularization parameter. Additionally, the decision function for a new input $\mathbf{x}$ is given by

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{A6}$$

Only those training points with $\alpha_i > 0$ contribute to the decision function. In our calculation, to perform the Linear SVC and RBF kernel, we used the Scikit-Learn [51] and chose the regularization parameter $C = 0$ and $C = 10$ in the Linear SVC and the RBF kernel, respectively. Additionally, γ for the RBC is chosen from "scale" in Scikit-learn.

In our numerical simulations, we employed a cosine-based implementation of Random Fourier Features (RFF), a widely used approach in the literature. Given an input x, the intermediate feature representation $z(x)$ is defined using a randomly sampled weight matrix $W \sim \mathcal{N}(0, \sigma^2)$ and a random bias $b \sim \text{Uni}[0, 2\pi]$, as follows:

$$z(x) = \cos(Wx + b). \tag{A7}$$

We carefully matched the feature space dimensions to ensure a fair comparison with the quantum optical reservoir computing (QORC) model. The output dimension of RFF was set equal to the dimensionality of the probability distribution obtained from QORC. Since QORC inherently generates a high-dimensional feature representation, achieving an equivalent feature space in RFF required using a large matrix W.

Furthermore, to maintain consistency with QORC, we incorporated the original image information into the feature representation. As a result, the final transformed input to the linear classifier was given by $[x, z(x)]$. Additionally, unlike standard RFF implementations, we applied a normalization step to enhance the stability of the optimization process. This normalization was also employed in QORC to ensure a robust training procedure. The final transformed representation was then fed into a linear classifier, which was subsequently optimized.

As shown in Table 1, our numerical analysis shows the QORC efficiently approximates the RBF with a finite-dimensional hidden layer, similar to the RFF [38,39]. Although the RBF is considered state-of-the-art in shallow machine learning, the computational cost of RBF scales with the cubic of the dataset size (for example, $60,000^3$ in MNIST) [52–54], making it difficult to handle big data ($10^6 \sim 10^9$ in order), while our optimization cost scales linearly with the dataset size.

Funding. MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) (JPMXS0118069605); COI- NEXT (JPMJPF2221); Japan Society for the Promotion of Science Kakenhi (21H04880); Japan's Council for Science, Technology and Innovation SIP Program (JPJ012367).

Acknowledgment. We thank S. Nishio, H. L. Nourse, and V. M. Bastidas for the useful discussion.

Disclosures. The authors declare no conflicts of interest.

Data availability. Data underlying the results presented in this paper are available from the authors upon reasonable request.

REFERENCES

1. F. Arute, K. Arya, R. Babbush, *et al.*, "Quantum supremacy using a programmable superconducting processor," Nature **574**, 505–510 (2019).
2. H.-S. Zhong, H. Wang, Y.-H. Deng, *et al.*, "Quantum computational advantage using photons," Science **370**, 1460–1463 (2020).
3. Y. Wu, W.-S. Bao, S. Cao, *et al.*, "Strong quantum computational advantage using a superconducting quantum processor," Phys. Rev. Lett. **127**, 180501 (2021).
4. L. S. Madsen, F. Laudenbach, M. F. Askarani, *et al.*, "Quantum computational advantage with a programmable photonic processor," Nature **606**, 75–81 (2022).
5. S. Aaronson and L. Chen, "Complexity-theoretic foundations of quantum supremacy experiments," arXiv (2016).
6. M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Average-case complexity versus approximate simulation of commuting quantum computations," Phys. Rev. Lett. **117**, 080501 (2016).
7. T. Morimae, "Hardness of classically sampling the one-clean-qubit model with constant total variation distance error," Phys. Rev. A **96**, 040302 (2017).
8. K. Fujii, H. Kobayashi, T. Morimae, *et al.*, "Impossibility of classically simulating one-clean-qubit model with multiplicative error," Phys. Rev. Lett. **120**, 200502 (2018).
9. Y. Takahashi and S. Tani, "Power of uninitialized qubits in shallow quantum circuits," Theor. Comput. Sci. **851**, 129–153 (2021).
10. S. Aaronson and A. Arkhipov, "The computational complexity of linear optics," in *STOC '11: Proceedings of the forty-third annual ACM symposium on Theory of computing* (2011), pp. 333–342.
11. J. Huh, G. G. Guerreschi, B. Peropadre, *et al.*, "Boson sampling for molecular vibronic spectra," Nat. Photonics **9**, 615–620 (2015).
12. K. Nakajima, K. Fujii, M. Negoro, *et al.*, "Boosting computational power through spatial multiplexing in quantum reservoir computing," Phys. Rev. Appl. **11**, 034021 (2019).
13. K. Fujii and K. Nakajima, "Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices," arXiv (2020).

14. Y. Suzuki, Q. Gao, K. C. Pradel, *et al.*, "Natural quantum reservoir computing for temporal information processing," Sci. Rep. **12**, 1353 (2022).
15. R. A. Bravo, K. Najafi, X. Gao, *et al.*, "Quantum reservoir computing using arrays of Rydberg atoms," PRX Quantum **3**, 030325 (2022).
16. P. Pfeffer, F. Heyder, and J. Schumacher, "Hybrid quantum-classical reservoir computing of thermal convection flow," Phys. Rev. Res. **4**, 033176 (2022).
17. P. Mujal, R. Martínez-Peña, G. L. Giorgi, *et al.*, "Time-series quantum reservoir computing with weak and projective measurements," npj Quantum Inf **9**, 16 (2023).
18. J. García-Beni, G. L. Giorgi, M. C. Soriano, *et al.*, "Scalable photonic platform for real-time quantum reservoir computing," Phys. Rev. Appl. **20**, 014051 (2023).
19. R. Martínez-Peña, J. Nokkala, G. L. Giorgi, *et al.*, "Information processing capacity of spin-based quantum reservoir computing systems," Cognit. Comput. **15**, 1440–1451 (2023).
20. S. Ghosh, A. Opala, M. Matuszewski, *et al.*, "Quantum reservoir processing," npj Quantum Inf. **5**, 35 (2019).
21. L. Domingo, G. Carlo, and F. Borondo, "Optimal quantum reservoir computing for the noisy intermediate-scale quantum era," Phys. Rev. E **106**, L043301 (2022).
22. R. Martínez-Peña, G. L. Giorgi, J. Nokkala, *et al.*, "Dynamical phase transitions in quantum reservoir computing," Phys. Rev. Lett. **127**, 100502 (2021).
23. W. Xia, J. Zou, X. Qiu, *et al.*, "The reservoir learning power across quantum many-body localization transition," Front. Phys. **17**, 33506 (2022).
24. A. Suprano, D. Zia, L. Innocenti, *et al.*, "Experimental property reconstruction in a photonic quantum extreme learning machine," Phys. Rev. Lett. **132**, 160802 (2024).
25. A. De Lorenzis, M. Casado, M. Estarellas, *et al.*, "Harnessing quantum extreme learning machines for image classification," Phys. Rev. Appl. **23**, 044024 (2025).
26. A. Sakurai, M. P. Estarellas, W. J. Munro, *et al.*, "Quantum extreme reservoir computation utilizing scale-free networks," Phys. Rev. Appl. **17**, 064044 (2022).
27. A. Hayashi, A. Sakurai, S. Nishio, *et al.*, "Impact of the form of weighted networks on the quantum extreme reservoir computation," Phys. Rev. A **108**, 042609 (2023).
28. D. Marković and J. Grollier, "Quantum neuromorphic computing," Appl. Phys. Lett. **117**, 150501 (2020).
29. W. R. Clements, P. C. Humphreys, B. J. Metcalf, *et al.*, "Optimal design for universal multiport interferometers," Optica **3**, 1460–1465 (2016).
30. K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**, 559–572 (1901).
31. H. Hotelling, "Analysis of a complex of statistical variables into principal components," J. Educ. Psychol. **24**, 417 (1933).
32. S. Haykin, *Neural Networks and Learning Machines*, 3 ed. (Pearson Education, Inc., McMaster University, 2009).
33. A. Novikoff, "On convergence proofs for perceptrons," presented at the Symposium on Mathematical Theory of Automata (Brooklyn, New York, April 24-26 1962).
34. J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," J. Mach. Learn. Res. **12**, 2121–2159 (2011).
35. L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," IEEE Signal Process. Mag. **29**, 141–142 (2012).
36. T. Clanuwat, M. Bober-Irizar, A. Kitamoto, *et al.*, "Deep learning for classical Japanese literature," arXiv (2018).
37. H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," arXiv (2017).
38. A. Rahimi and B. Recht, "Random features for large-scale kernel machines" (2007).
39. M. Belkin, D. Hsu, S. Ma, *et al.*, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," Proc. Natl. Acad. Sci. U. S. A. **116**, 15849–15854 (2019).
40. W. Xiong, G. Facelli, M. Sahebi, *et al.*, "On fundamental aspects of quantum extreme learning machines," Quantum Mach. Intell. **7**, 20 (2025).
41. N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions," SIAM Rev. **53**, 217–288 (2011).
42. P. Nakkiran, G. Kaplun, Y. Bansal, *et al.*, "Deep double descent: where bigger models and more data hurt," arXiv (2019).
43. Z. Liao, R. Couillet, and M. W. Mahoney, "A random matrix analysis of random Fourier features: beyond the Gaussian kernel, a precise phase transition, and the corresponding double descent," J. Stat. Mech. **2021**, 124006 (2021).
44. T. Hastie, A. Montanari, S. Rosset, *et al.*, "Surprises in high-dimensional ridgeless least squares interpolation," arXiv (2020).
45. M. Nakajima, K. Tanaka, and T. Hashimoto, "Scalable reservoir computing on coherent linear photonic processor," Commun. Phys. **4**, 20 (2021).
46. C. Ma, J. V. Kerrebrouck, H. Deng, *et al.*, "Integrated photonic reservoir computing with an all-optical readout," Opt. Express **31**, 34843–34854 (2023).
47. S. G. K. Patro and K. K. Sahu, "Normalization: a preprocessing stage," arXiv (2015).
48. Y. LeCun, B. Boser, J. S. Denker, *et al.*, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541–551 (1989).
49. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv Neural Inf Process Syst. **25**, 12 (2012).
50. A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: transformers for image recognition at scale," arXiv (2020).
51. F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: machine learning in Python," J. Mach. Learn Res. **12**, 2825–2830 (2011).
52. B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, 2002).
53. K. Koutroumbas and S. Theodoridis, *Pattern Recognition* (2008).
54. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, 2004).