

Quantum Extreme Reservoir Computation Utilizing Scale-Free Networks

Akitada Sakurai,^{1,2} Marta P. Estarellas,^{2,†} William J. Munro^{2,3} and Kae Nemoto^{1,2,4,*}

¹*School of Multidisciplinary Science, Department of Informatics, SOKENDAI (the Graduate University for Advanced Studies), 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*

²*National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*

³*NTT Basic Research Laboratories & Research Center for Theoretical Quantum Physics, 3-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, Japan*

⁴*Quantum Information Science and Technology Unit, Okinawa Institute of Science and Technology Graduate University, Onna-son, Okinawa 904-0495, Japan*

(Received 5 September 2021; revised 10 March 2022; accepted 2 May 2022; published 23 June 2022)

Today's quantum processors composed of fifty or more qubits have allowed us to enter a computational era where the output results are not easily simulatable on the world's biggest supercomputers. What we have not seen yet, however, is whether or not the complexity achievable with such quantum processors would ever be useful for any practical applications. In the quantum neural network approach, there has been an expectation that the Hilbert space can serve as a computational resource by providing a large feature space; however, it is not understood, nor illustrated in what way the Hilbert space provides that computational power. In this work we introduce a resource-efficient quantum neural network model with a focus on the role of the feature space the quantum processors can give. Then we apply it for classification tasks, showing a quantum advantage in terms of the physical resources needed to realize the neural network. To facilitate the Hilbert space as a large feature space, we utilize scale-free networks that can be generated in a discrete-time crystal model. The virtue of this approach is in both its theoretical simplicity and its practical applicability: our quantum neural network model is simple enough to identify the role of the Hilbert space as the feature space, and, furthermore, can be implemented with current technology. Our model does not require optimization of the quantum processor, and hence once the quantum processor has been set, it can be used for other different classification problems without any changes in the quantum processor itself.

DOI: 10.1103/PhysRevApplied.17.064044

I. INTRODUCTION

The recent realization of quantum processors with fifty plus qubits is undoubtedly a key milestone for the nascent quantum technology field [1–3]. The quantum advantage has been estimated through its complex dynamics these quantum processors provide, benchmarking it against the necessary run time for a classical computer to simulate it. In the field of quantum neural networks (QNNs), it has been believed that the Hilbert space associated with the quantum complex dynamics provides a large and rich feature space for machine intelligence. However, the mechanism to generate complex behaviors alone is not enough to ensure its quantum computational power, as the information we can extract from the quantum processor is limited by the fundamental principles of measurements in quantum mechanics. Hence, showing a way to facilitate

efficient Hilbert space usage for them would be a significant step forward to design QNNs applicable to practical applications.

QNNs have been considered as one of the potential directions to utilize the current quantum processors [including noisy intermediate scale quantum (NISQ) devices]. There have been a number of approaches and definitions for QNNs [4–7]. In the early days of QNNs, the core ideas behind perceptron (a model of artificial neural networks) had been extended to QNNs [8], where the perceptron activation function was replaced by an operator. Despite the simple notion of the quantum perceptron given by Altaisky [8], its implementation remains highly nontrivial. The main focus of the early works on quantum perceptron was to integrate the nonlinearity present in the classical model into quantum mechanics [9–12]. More recently, QNNs have been extensively investigated as a subclass of variational quantum algorithms (VQAs) [13, 14]. VQAs can be modeled with a feature map and a variational model with classical feedforward, some of which have been demonstrated on gate-based quantum computers

*nemoto@nii.ac.jp

†Qilimanjaro Quantum Tech, Carrer dels Comtes de Bell-Lloc, 161, 08014 Barcelona, Spain.

[15,16]. It has been suggested that some quantum neural networks may offer an advantage over classical neural networks [13]; however, the optimization of a parameterized quantum circuit with a classical feedback loop would be difficult to scale [17]. Quantum reservoir computation (QRC) can also be considered as a type of QNN, which was introduced to mainly analyze temporal and sequential data processing with both qubits [18,19] and continuous variables [20]. In those works quantum neural networks are considered in the Hilbert space, whereas another approach uses a network of physical qubits [21], which has been used to construct a universal quantum reservoir computer [22].

In this paper, we propose a QNN model based on extreme machine learning (EML) [23,24], where the random classical network with a Gaussian distribution is replaced by a quantum reservoir. There are many choices to design the quantum reservoir; however, we employ a discrete-time crystal model (shown in Fig. 1). A recent study has shown that this model can generate nontrivial complex networks on the Hilbert space [25], and in particular, for a certain parameter region, it generates scale-free networks indicated in Fig. 1(a). While network science tells us that most of the interesting network properties arise from directed networks [26], it is known that the quantum network does not require direction in its flow [27]; hence, it is enough to set the connections between states to characterize the quantum network. We use these nontrivial networks as a quantum layer of our QNN [indicated in light blue in Fig. 1(d)]. There are two important motivations in this choice: the discrete-time crystal model has been realized and requires only established technology [28]. Its implementation is much simpler than the average quantum gate sequence on NISQ processors and hence it is likely to complete the computation well within the quantum processors coherence time. The quantum equivalence of a random network with a Gaussian distribution is costly. The scale-free networks tend to appear more frequently in Hamiltonian quantum dynamics and the analysis of the network properties in the QNN scenario would give us important insights about what nature of the quantum dynamics on the Hilbert space provides a useful feature space for QNNs.

Our QNN model presents several advantages. First and foremost, the network on the Hilbert space grows exponentially with the number of qubits N ; 50 qubits could generate a network as large as the neural network in a human brain. Second, despite this rapid growth in complexity, as our quantum computation is run by the quantum reservoir, it does not need to be programmed. The parameter setting for the quantum reservoir remains simple, and once it has set to exhibit the scale-free nature, we do not touch the quantum reservoir. Hence, the model is capable of being applied to different computational tasks without any changes to the quantum reservoir. Third, by

eliminating the costly parameter optimization over the variational model in VQAs, we can expect a high feasibility of our model and a significant speed up in the training process. This advantage of the learning speed comes from the nature of the extreme learning machine [23,24].

We start with a detailed description of our model in the next section.

II. THE MODEL

Our quantum extreme reservoir computation (QERC) model consists of two classical layers and one quantum reservoir, as illustrated in Fig. 1(c). The model can be interpreted as a QNN, as illustrated in Fig. 1(d). To start the computation, we first preprocess the classical data $I_{ij}^{(k)}$ for the k th image in order to efficiently encode it to the quantum reservoir [indicated in a green box labeled principle component analysis (PCA) map in Fig. 1(c)]. For the pattern recognition problem in this paper, we employ PCA to extract the most influential elements in each image; however, it should be tailored to the task at hand. The selected set of classical data is then encoded on the initial state of the quantum reservoir. The quantum reservoir converts the initial state for the k th image $|\psi_0^{(k)}\rangle$ to the output state $|\psi_f^{(k)}\rangle$. The output state is measured on the computational basis at the M layer, and then the measurement date is processed in the one-layer classical neural network (ONN). The training is performed only on the ONN.

The quantum reservoir is given as a discrete-time crystal model, and its Hamiltonian $H_\epsilon(t)$ is given by [28,29]

$$\hat{H}_\epsilon(t) = \begin{cases} \hat{H}_1 \equiv \hbar g(1 - \epsilon) \sum_l \sigma_l^x, & 0 \leq t < T_1, \\ \hat{H}_2 \equiv \hbar \sum_{lm} J_{lm}^z \sigma_l^z \sigma_m^z, & T_1 \leq t < T, \end{cases} \quad (1)$$

where $\{\sigma_l^x, \sigma_l^y, \sigma_l^z\}$ are the Pauli operators on the l th qubit, while $J_{lm}^z \equiv J_0/|l - m|^\alpha$ represents the long-range interaction between the l th and m th qubits that takes the form of an approximate power-law decay with a constant exponent α . The layout of the qubits to determine the distance, $|l - m|$, between qubits is a one-dimensional line indicated in Fig. 1(b). Next the parameter g satisfies the condition $2gT_1 = \pi$ such that $\hat{U}_1 = \exp(-i\hat{H}_1 T_1/\hbar)$ realizes a global π pulse around the x axis. We assume that $T_1 = T/2$ for convenience. The parameter ϵ is our rotation error [25]. Because this is a periodic system, the Floquet operator is given by

$$\hat{\mathcal{F}}_\epsilon = \hat{U}_\epsilon(T) = \exp\left(-\frac{i}{\hbar}\hat{H}_2 T_2\right) \exp\left(-\frac{i}{\hbar}\hat{H}_1 T_1\right). \quad (2)$$

The network representation can be obtained from the effective Hamiltonian of the Floquet operator $\hat{H}_\epsilon^{\text{eff}} =$

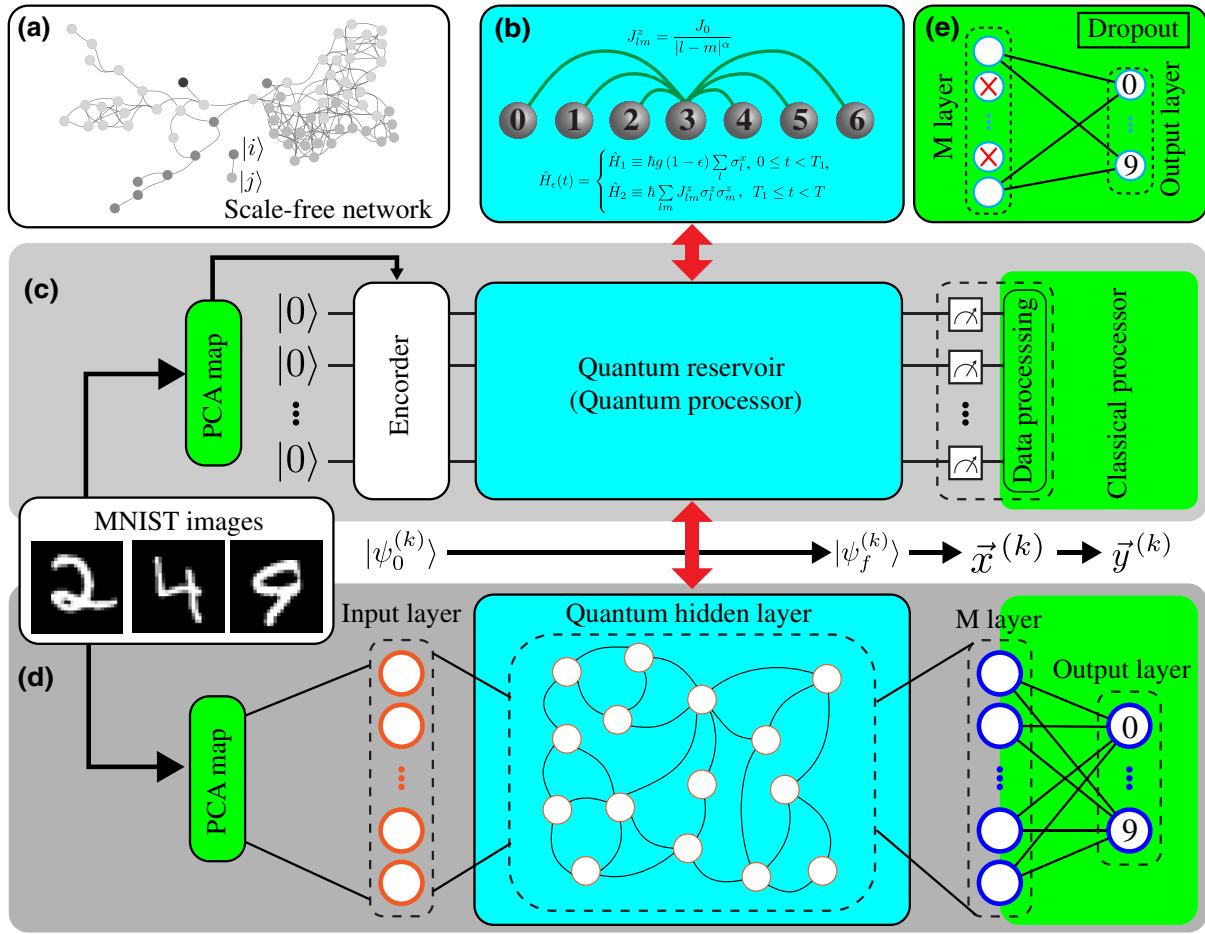


FIG. 1. A schematic illustration of the structure of the quantum reservoir computer. Panel (a) shows an example of a scale-free network represented in Hilbert space. Each dot indicates a computational basis state, while each edge is weighted with the hopping strength between the two states. When the hopping strength is smaller than a threshold given by the percolation rule, there is no edge between the states. Panel (b) indicates a quantum system used to implement the quantum reservoir. Here the model of a discrete-time crystal is used. Panel (c) illustrates the total scheme of our model. The green color indicates classical data processing, whereas light blue is for quantum information processing. Panel (d) explains the model described in (c) as a quantum neural network. There are four network layers: the input layer, the quantum hidden layer, the M layer, and the output layer. The input layer encodes the input data onto the quantum hidden layer (given by the quantum reservoir), while the M layer is to measure the quantum system in the computation basis, converting the quantum data (state) to the classical data. The one-layer neural network (ONN) is responsible for the training. Panel (e) describes the ONN with dropout. Dropout is used when the training accuracy is saturated to unity.

$i\hbar/T \log[\hat{\mathcal{F}}_\epsilon]$ and the percolation rule [30]. The Hamiltonian of an N -qubit system in the computational basis $\{|i\rangle\}$, where i is a binary digit from 0 to $2^N - 1$, may be represented as $H = \sum_i \mathcal{E}_i |i\rangle\langle i| + \sum_{i,j (i \neq j)} \mathcal{W}_{ij} |i\rangle\langle j|$, where \mathcal{E}_i and \mathcal{W}_{ij} correspond to the energy of the computational basis $|i\rangle$ and the transition energy between $|i\rangle$ and $|j\rangle$, respectively. The network has an edge between the i th node (state) and j th node (state) with weight \mathcal{W}_{ij} when the percolation condition $|\mathcal{E}_j - \mathcal{E}_i| < |\mathcal{W}_{ij}|$ is satisfied [30]. The percolation rule is to eliminate the off-resonant transitions from the network. When $\epsilon = 0$, a period-2 discrete-time crystal (2T DTC) appears, and the corresponding network is a locally connected network (a set of dimers). As ϵ gets larger, the network starts to grow following a preferential

attachment mechanism and typically forms a scale-free network, and when ϵ reaches a critical region ($\epsilon \sim 0.03$), the network goes through the transition from scale-free to random [25]. In our model, we set ϵ to 0.03, and we evaluate this ansatz of the near optimality by numerically calculating the ϵ dependency in the performance.

We note that it is also possible to implement a Hamiltonian H of a given weighted graph of a scale-free network to design our quantum reservoir. Then the necessary quantum dynamics can be given by $U = e^{-iH\tau/\hbar}$. The only possible problem to this approach is that such Hamiltonians generally involve many-body interactions that are harder to implement. It is equally possible to implement the unitary map U as a gate sequence. Such an approach

would be more easily adaptable to NISQ processors; however, the cost of the gate decomposition of the unitary map and its optimization could remain. Even though it is only necessary to set the quantum reservoir, the depth of the circuit could be problematic for NISQ devices. By contrast, the discrete-time crystal has been experimentally demonstrated [28,31,32], and the computational time we need for the quantum reservoir is less than one half of the typical coherence time of an ion-trap experiment with ten qubits [28]. Hence, our model is a convenient platform to investigate the capabilities and mechanism of the feature space in QNNs. The discrete-time crystal can also be implemented on a gate-based quantum processor by a gate decomposition of the Floquet operator or the entire unitary map for the quantum dynamics [32].

III. PATTERN RECOGNITION WITH QRC

Here we illustrate how our quantum extreme reservoir computer performs pattern recognition. We use the MNIST handwritten digit data set publicly available [33] to benchmark the performance. The data set contains 70 000 (28×28 pixel) images of handwritten digits between 0 and 9. The classical data for each image are first processed by the PCA, as shown in Fig. 2, which allows us to select elements from the largest contribution. With the PCA, the k th image data are represented as a vector $\vec{I}^{(k)} = \sum_{j=1}^{784} \tilde{c}_j \vec{v}_j$, where the j th element is the j th contribution among the 784 elements. Although it is possible to encode all 784 values to the quantum input state, it may require a complicated quantum circuit; instead, we select the $2N$ largest contribution elements to encode to the quantum input state. These $2N$ values are encoded to each qubit by single-qubit rotations only; for the l th qubit, the mapping is $\tilde{c}_l \rightarrow \theta_l$ and $\tilde{c}_{N+l} \rightarrow \phi_l$ and the quantum input state for the l th qubit is

$$|\psi_l\rangle = \cos \frac{\theta_l}{2} |0\rangle + e^{i\phi_l} \sin \frac{\theta_l}{2} |1\rangle. \quad (3)$$

This model of the discrete-time crystal with parameters $J_0 T = 0.06$, $\alpha = 1.51$ shows a scale-free nature around the value $\epsilon \sim 0.03$, as shown by the degree distribution of each network in Fig. 3(b), and as the number of qubits increases, the power-law distribution becomes more prominent [25]. The coupling strength corresponds to the weak coupling regime.

Now in the M layer, we measure the output state $|\psi_f^{(k)}\rangle$ in the computational basis (the single-qubit Z basis) and extract the classical data associated with it. The classical data we obtain here are the population density of the wave function of the final state, and in a real experiment we can statistically obtain such data by repeating the procedure a number of times. In our numerical analysis we calculate the final state numerically using the Floquet operator (2) instead of applying a sampling method. We note that there

is no feedforward in these repeating processes. The distribution of the population density is then renormalized to have an average 0 and variance 1 for the convenience of the classical data analysis on ONN to stabilize the classical optimization. The renormalized distribution $\tilde{x}^{(k)}$ is the output of the M layer.

The output $\tilde{x}^{(k)}$ is then weighted with the weight matrix W of the ONN. We follow a method widely used in pattern recognition (see the appendices for details) to obtain the output of the computation $\tilde{y}^{(k)}$.

IV. PERFORMANCE AND PROPERTIES

We begin by first training our ONN with 60 000 samples using gradient decent, back propagation, and the minibatch method, where we evaluate the accuracy rate for the test samples. To evaluate the optimal value of ϵ , we calculate the ϵ dependency of the performance for both training and testing.

Firstly, we evaluate how the accuracy rate for testing changes with the number of epochs for each value of ϵ . The black line in Fig. 3(a) shows the accuracy rate without the quantum reservoir with the full classical input data (784 PCA elements). We use this case with no quantum reservoir to benchmark the performance that the QERC at least needs to achieve. Otherwise, the quantum reservoir would not give any benefits to the computation.

The gray line is for $\epsilon = 0$, and interestingly we observe that the reduction of the classical information at the input layer does not significantly affect the accuracy rate. This indicates that the nonlinear map between the classical data

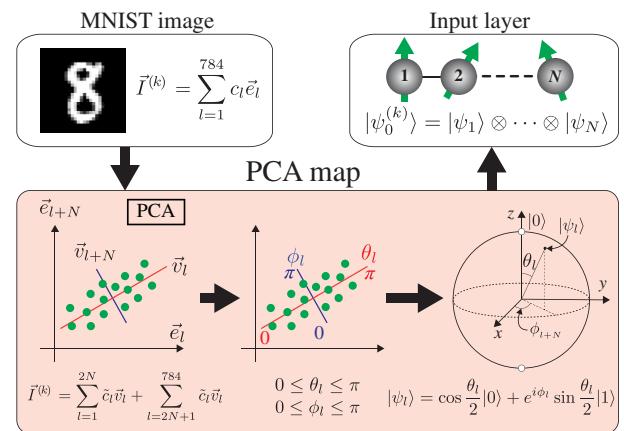


FIG. 2. Schematic diagram detailing how the $2N$ parameters are extracted from a sample image and encoded onto N qubits. The left top box shows that the k th image from the MNIST image set requires 784 parameters to represent the data in full. In the PCA map (the bottom box), the PCA is used to select the $2N$ most influential parameters, and a pair of parameters (c_l, c_{l+N}) is encoded on one qubit by single-qubit rotations. Panel (c) shows the input state of the quantum reservoir.

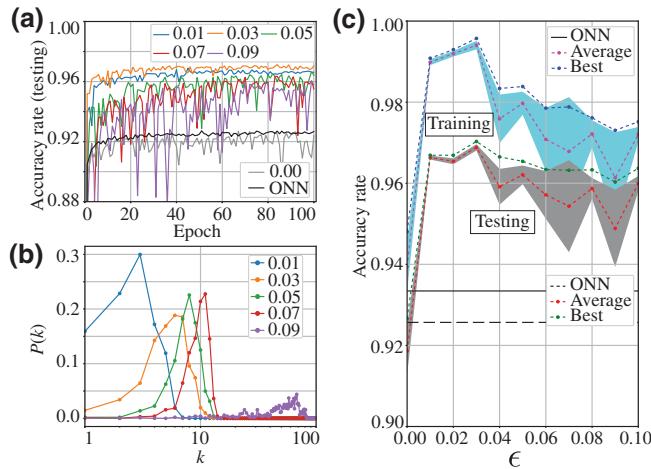


FIG. 3. Performance of the QRC. Plot (a) shows the accuracy rate versus the number of epochs for each value of ϵ . The number of periods for the time evolution of the quantum reservoir n is 50, and the coupling strength $J_0 T = 0.06$. The degree distribution of the network for the effective Hamiltonian for each ϵ is plotted in (b), where k is the degree (the number of links) at each node and $P(k)$ is the number of nodes for the degree of k . To optimize the ONN, we employ stochastic gradient decent with minibatch size 100. Plot (c) shows the average with the associated standard deviation and best accuracy rates for both training and testing. The average and the standard deviation are taken for 90 to 100 epochs. The number of qubits is $N = 11$.

and the quantum state as well as the application of PCA helps to push up the accuracy rate.

In Fig. 3(a), the accuracy rate for testing quickly goes up high for ϵ values 0 to 0.03. The optimality peaks around $\epsilon = 0.03$ and gradually reduces when ϵ gets larger. The degree distributions of the network for the effective Hamiltonian with $\epsilon = 0.01$ to 0.03 exhibit the scale-free nature, as shown in Fig. 3(b). The near optimality of the performance for both the accuracy rate and the speed of learning can be obtained in the regime of the effective Hamiltonian with the scale-free nature. The computational time for the quantum reservoir in this comparison is set to 50 steps. Figure 3(c) shows the ϵ dependency on the accuracy rate for training and testing. The stability of the performance is also better in the region of scale-free networks. The system size of the quantum reservoir is $N = 11$.

Next, we show the dependence of the accuracy rate on the computational time of the quantum reservoir. As the model for the quantum reservoir has a period, we use the number of periods n for the computational time. Figures 4(a-1) and 4(a-2) show the n dependency of the accuracy rate for training (a-1) and testing (a-2). The accuracy rates nearly saturate around $n = 50$, and this number of periods is feasible in comparison to the current ion-trap technology [28].

The scaling of the performance is another important element. Figures 4(b-1) and 4(b-2) show the dependency of

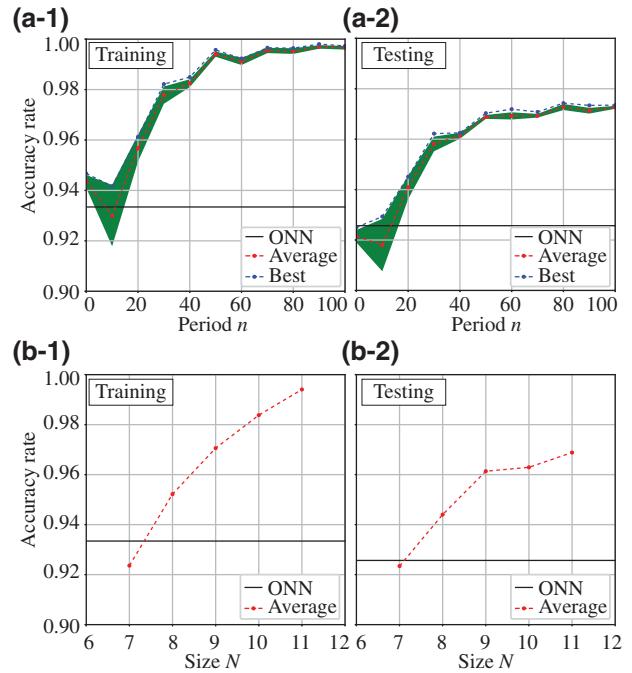


FIG. 4. Performance of the QRC with time (nT) and size (N). The top two figures plot the average with the associated standard deviation and best accuracy rates for training (a-1) and testing (a-2) for the different periods with $N = 11$. The bottom two figures show the size dependency of the accuracy rate for training (b-1) and testing (b-2). The parameters are set as $J_0 T = 0.06$ and $\epsilon = 0.03$. The average and the standard deviation are taken for 90 to 100 epochs.

the accuracy rate on the system size N for the quantum reservoir. With a larger N , we exponentially increase the dimension of the feature space and linearly increase the number of PCA elements to be input. The accuracy rates for both training and testing go up as the system size N increases. In fact, only two extra qubits from $N = 7$ to 9 gives a nearly 4% improvement in the accuracy for testing.

Our model does not have disorder among the qubits; however, it might be more realistic to have a small disorder among the qubits in a real system. The scale-free nature of the effective Hamiltonian is usually evaluated statistically over the disorder distribution. Our system is limited in size, and hence this statistical property has fluctuations; some instances show a power-law distribution clearer than other instances. In particular, the case of no disorder can be slightly special. When the system size N is even, the system could acquire an additional symmetry and the degree distribution is less power law than in the case of an odd number N . Although we can see the slight slowdown in the accuracy rate increase in Fig. 4(b-2), the numerical calculations show that the effectiveness of the quantum reservoir is stable when the disorder is small and is not sensitive to the degree distribution of each instance as long as the parameter ϵ is chosen from the right parameter regime.

V. OVERRFITTING AND DROPOUT

One of the most common problems with machine learning is overfitting, and our model is no exception. The gap between the accuracy rates for the training [Fig. 4(a-1)] and the testing [Fig. 4(a-2)] gets larger as the learning progresses, and the accuracy rate for training in Fig. 4(b-1) almost reaches unity at $N = 11$. This suggests overfitting. To capture this more clearly, we plot the difference between the accuracy rates for training and testing in Fig. 5(a). Here we observe that the blue line does not saturate. To address this, we introduce dropout to the ONN. Dropout is one of the techniques developed in classical machine learning to circumvent overfitting by randomly erasing information in neural networks, as shown in Fig. 1(d). It is not commonly used due to the computational cost associated with the technique [34]; however, in our case the cost of dropout is minimal due to the structure of the classical network (ONN). Figure 5(a) shows the accuracy gap between the training and testing for four different degrees of dropout ranging from 0%–15%. When dropout is introduced, overfitting is significantly suppressed.

Now we reevaluate the accuracy rates for the QERC model. Figures 5(b) and 5(c) show that we can achieve the best performance for testing with 5% dropout for $N = 11$, while Fig. 5(c) shows the accuracy rate for testing samples with a 5% dropout for different values of ϵ . The performance of the QERC model slightly increases with dropout, with a maximum accuracy rate for testing of 97.1% (average) [97.3% (best)], and hence dropout successfully suppresses the accuracy rate for training and increases it for testing for a larger N where the overfitting is more severe.

It is interesting to have a comparison between the QERC performance and its classical counterpart. The best comparison would be a test for the approaches that put both on an equal footing; however, it is rather unpractical to compare on an equal footing all aspects of the computational models. Here, we present a simple comparison between the QERC and a classical ELM with the random network as large as the network on the Hilbert space of the quantum reservoir, so that their learning times (optimization times) remain the same. The training and testing accuracies of the classical ELM are about 100% and 95.9%, respectively. Here, we optimize it with minibatch 100 and 200 epochs. This indicates that QERC has a higher performance in the testing by about 1%. Interestingly, its overfitting is smaller than that of the classical ELM, and there is an exponential compression in the physical resources (the number of bits or qubits), though the computational time for the quantum reservoir is not discussed here.

It is also useful to provide a brief comparison of the QERC with quantum classifiers using qubits. There are a number of approaches for quantum classifiers; however,

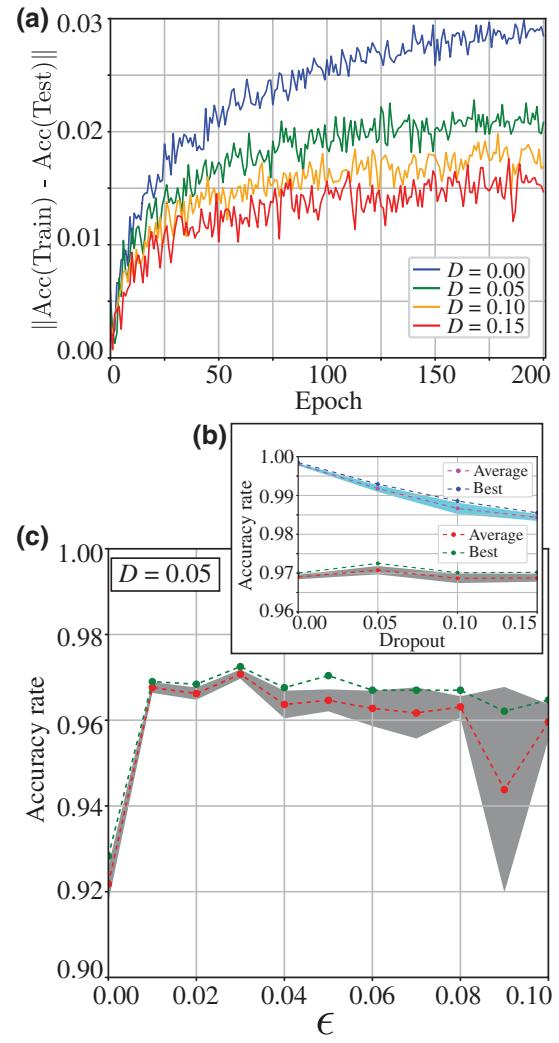


FIG. 5. The effect of dropout on the accuracy of the QERC. Panel (a) shows the accuracy rate difference between training and testing for each dropout. The difference for all nonzero dropout rates saturates as the learning proceeds, which indicates that overfitting can be suppressed by dropout. Panel (b) shows the accuracy rates for training (top lines) and testing (bottom lines) to choose the best rate for dropout, which is $D = 0.05$ (5%) in this case. Overfitting can be further suppressed by increasing D ; however, the overall performance starts to drop for large D . Panel (c) shows the ϵ dependency on the accuracy rate for testing with 5% dropout. All other parameters are the same as in Fig. 3. We increase the number of epochs to 200 in order to obtain enough network realizations with dropout. We evaluate the average and the best accuracy from 190 to 200 epochs.

for this comparison, we choose the classifier based on tensor networks [35], which classifies the ten handwritten digits using the MNIST dataset. This approach uses over 700 qubits to achieve testing accuracies of 95.5% and 97.0% [35,36]. This was a reason for the common belief

that a relatively large number of qubits would be necessary for a quantum classifier for the digits. Our QERC uses a significantly smaller number of qubits to classify the ten handwritten digits and is highly compatible with current NISQ technology.

VI. CONCLUSION

We have shown that our proposed quantum extreme reservoir computation model presents a quantum advantage for classification tasks. Our model is significantly simpler by eliminating the necessity of quantum programming and quantum or classical feedforward to the quantum processor. We have also shown that the quantum reservoir can be implemented using a DTC, the realization of which has already been demonstrated experimentally in many physical systems. The coherence time achievable with the current ion-trap technology is much longer than the necessary coherence time for this model [28]. The quantum gain is visible with the quantum reservoir formed by a system as small as $N = 8$, which is feasible for a number of different physical realizations.

Our model is applicable to other classification problems. In particular, the quantum reservoir remains the same when running different tasks. Thus, the quantum reservoir can be used as an all-purpose accelerator or booster for numerous classification problems. To illustrate this capability, we show that our model with exactly the same quantum reservoir with $N = 8$ can be used to solve a typical nonlinear classification problem in Appendix E.

Our analysis shows that the complexity on the feature space is crucial for QNNs, and it suggests that the variational model is not necessarily important, which to some extent agrees with the recent observation of the role of the feature map in VQAs [13]. The results of this paper provide a route to investigate the properties of quantum dynamics to gain quantum advantage for computing tasks, and our scheme has paved the way for a practical quantum computation platform.

ACKNOWLEDGMENTS

We thank Victor M. Bastidas and Aoi Hayashi for valuable discussions. This work is supported by the MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) under Grant No. JPMXS0118069605.

APPENDIX A: PRINCIPLE COMPONENT ANALYSIS

In our model, PCA is used to effectively encode the classical information to the quantum hidden layer, as shown in Fig. 2. The k th MNIST image is converted to the vector $\vec{I}^{(k)}$. The basis vectors $\{\vec{v}_j\}_{j=1,\dots,784}$ for this vector representation need to be optimized with the sample data. Once the basis vector set has been optimized with the sample

data, the same set is used to represent the test images. The conversion $\tilde{c}_l \rightarrow \theta_l$ or ϕ_l is given by

$$\theta_l \text{ or } \phi_l = \frac{\pi(\tilde{c}_l - \min[\tilde{c}_l^{(\text{train})}])}{\max[\tilde{c}_l^{(\text{train})}] - \min[\tilde{c}_l^{(\text{train})}]}, \quad (\text{A1})$$

where $\max[\tilde{c}_l^{(\text{train})}]$ and $\min[\tilde{c}_l^{(\text{train})}]$ mean the maximum and minimum values of \tilde{c}_l across all the training samples, respectively. Here, we note that, for the case where θ_l or ϕ_l goes beyond the range $[0, \pi]$ with testing samples, we truncate the value. Because of the optimization of the basis vectors, the loss of information by the reduction from 784 elements to the $2N$ highest contribution elements is minimized.

APPENDIX B: ONN WITH THE M LAYER AND THE OUTPUT LAYER

The ONN is a one-layer neural network of the M layer with m active neurons and the output layer with n active neurons; the ONN can then be optimized. Thus, the dimension of its weight matrix is $m \times n$. The number of neurons in the M layer can in principle be as large as $m = 2^N$ (where N is the number of qubits); however, when we apply dropout, we randomly eliminate the active neurons in this layer, as illustrated in Fig. 1(d). Only when we obtain the black line in Fig. 3(a) for the case without the quantum reservoir and the measurement in the M layer do we use the full PCA elements with size $m = 784$.

At the output layer, the number of neurons is n , which is the same as the number of classes of task. In the case with the MNIST dataset, there are ten neurons ($n = 10$), each corresponding to a digit to be recognized. At the output layer the effects of the weight matrix is summarized and a shift \vec{B} is applied as

$$u_l = \sum_{i=1}^m x_i \times w_{i,l} + B_l, \quad (\text{B1})$$

where x_i is the i th element of the output of the M layer \vec{x} and b_l is the l th element of the shift \vec{B} . As our computational task is pattern recognition, we insert the activation function f to convert the data \vec{u} to \vec{y} as $\vec{y} = f(\vec{u})$.

For the activation function, we employ the soft-max function [37,38], which is widely used in classification problems [37,38]:

$$y_l = f(x_l) = \frac{\exp(x_l)}{\sum_k \exp(x_k)}. \quad (\text{B2})$$

We define the lost function L_k for the k th sample by the cross entropy as

$$L_k = - \sum_l^n t_l^{(k)} \log(y_l^{(k)}), \quad (\text{B3})$$

to evaluate the learning progress. Here $\vec{y}^{(k)}$ is the output vector, while $\vec{t}^{(k)}$ is a one-hot vector, that is, a basis unit vector of the ten-dimensional vector space. The one-hot vectors represent the correct result and are hence the reference for the evaluation of the output of the ONN. Then we apply gradient descent, back propagation, and the minibatch method to optimize the ONN.

APPENDIX C: GRADIENT DESCENT AND BACK PROPAGATION

The weight matrix (w_{ij}) and the shift vector (b_i) are optimized using the gradient descent method, through

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} - \eta \frac{\partial L_l}{\partial w_{ij}}, \quad (C1)$$

$$b_i^{(n+1)} = b_i^{(n)} - \eta \frac{\partial L_l}{\partial b_i}, \quad (C2)$$

where η is a learning rate. To calculate the derivatives, we use the back propagation method. Applying the chain rule with Eqs. (B1)–(B3), we have

$$\frac{\partial L_l}{\partial w_{ij}} = x_{l,i}(y_j^{(l)} - t_j^{(l)}), \quad (C3)$$

$$\frac{\partial L_l}{\partial b_i} = (y_j^{(l)} - t_j^{(l)}). \quad (C4)$$

APPENDIX D: MINIBATCH METHOD

We use the minibatch method to reduce the computational cost and to avoid local minima. In this method, the loss function is now the average of the loss function for each sample and is given by

$$L = \frac{1}{M} \sum_{l=1}^M L_l, \quad (D1)$$

where M is the batch size and L_l is the cost function for the l th sample. The hidden layer can be written as the matrix

$$X = \begin{pmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_M^T \end{pmatrix}. \quad (D2)$$

In this method, the derivative for the weight matrix is given in matrix representation by

$$\frac{\partial L}{\partial w_{ij}} = \frac{1}{M} X^T \cdot \begin{pmatrix} \vec{y}^{(1)} - \vec{t}^{(1)} \\ \vdots \\ \vec{y}^{(M)} - \vec{t}^{(M)} \end{pmatrix}. \quad (D3)$$

The derivative for the bias vector is then a sum of each derivative:

$$\frac{\partial L}{\partial b_i} = \frac{1}{M} \sum_l^M \frac{\partial L_l}{\partial b_i}. \quad (D4)$$

APPENDIX E: ISOTROPIC GAUSSIAN SAMPLES

Our previous results demonstrated the classification task of the MNIST dataset using quantum extreme reservoir computation. We employ the 2T-DTC model as the Floquet quantum system and show that the classification performance is higher in the parameter regime where the DTC melts, and therefore the network is in the scale-free regime.

The fundamental question is whether our quantum extreme reservoir computation model can be applied to other classification problems. Since our model has an optimized one-layer classical neural network, it is guaranteed to solve any linear classification problem. However, it is not obvious whether it can solve other nonlinear classification problems. Given its extreme machine learning nature, it does seem possible—especially given the complexity of the huge networks provided by quantum systems. To reveal the relationship between the time evolution of a quantum system and its classification ability for nonlinear problems, we use isotropic Gaussian samples with two classes (red and blue) in a two-dimensional space, as shown in Fig. 6(a).

In our quantum extreme reservoir computation model the initial state of the quantum system is determined by only $2N$ parameters, where N is the total number of qubits in the system. In the current two-class Gaussian distribution example, each sample point is determined by two values and so can be fully encoded. For the k th point $\vec{I}^{(k)} = (I_1^{(k)}, I_2^{(k)})^T$, we can determine our two angles ($\theta^{(k)}, \phi^{(k)}$) as

$$\theta^{(k)} \text{ or } \phi^{(k)} = \frac{\pi(I_{1 \text{ or } 2}^{(k)} - \min[\mathbf{I}]_{1 \text{ or } 2})}{(\max[\mathbf{I}]_{1 \text{ or } 2} - \min[\mathbf{I}]_{1 \text{ or } 2})}. \quad (E1)$$

Here, since we are considering supervised learning, we assume that we have a prior training dataset, $\mathbf{I} = (\vec{I}^{(1)}, \dots, \vec{I}^{(k)}, \dots)$, in advance. Thus, $\min[\mathbf{I}]_{1 \text{ or } 2}$ and $\max[\mathbf{I}]_{1 \text{ or } 2}$ mean minimum and maximum values of each component ($I_1^{(k)}, I_2^{(k)}$) across all training data, respectively. Thus, one can encode the k th point into the qubit defined in Eq. (3), where θ_k and ϕ_k depend on that point. Then, the initial state of N qubits for the point ($\vec{I}^{(k)}$) is given by

$$|\psi(\vec{I}^{(k)})\rangle_i = |\psi\rangle_1 \otimes |+\rangle_2 \otimes \cdots \otimes |+\rangle_N. \quad (E2)$$

with $|+\rangle_i = (|0\rangle + |1\rangle)/\sqrt{2}$.

We use the 2T-DTC model used in the main text as a quantum system to classify the sample points of this

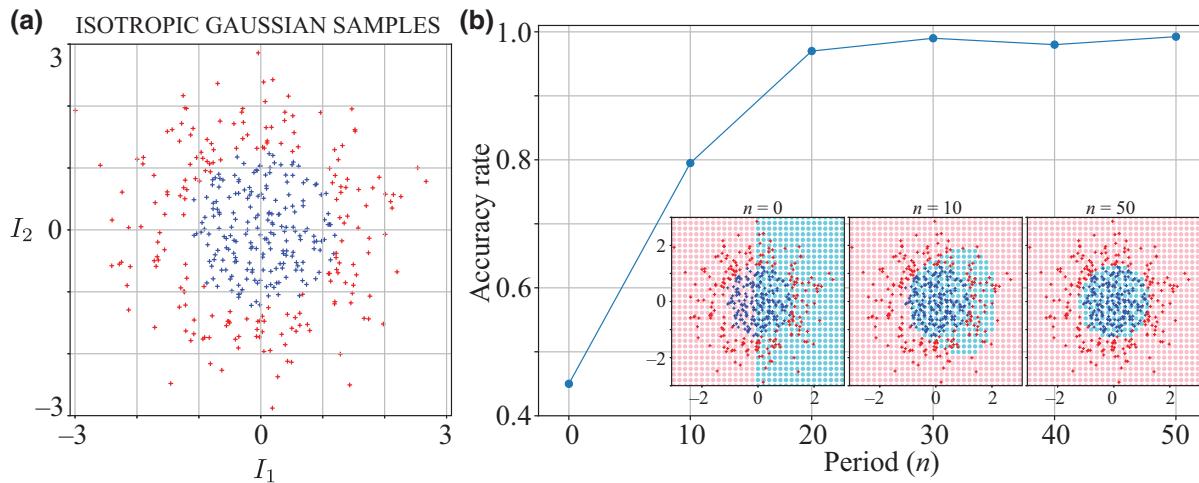


FIG. 6. Isotropic Gaussian samples in two-dimensional space and quantum extreme reservoir computation performance with period (nT). Panel (a) shows the isotropic Gaussian samples with two labeled classes, red and blue. Panel (b) shows the accuracy rate of the classification for training samples. The insets show the results of the classification for different periods $nT = 0$, $nT = 10$, and $nT = 50$ with 2T-DTC parameters $N = 8$, $J_0 T = 0.06$, $\epsilon = 0.02$, and $\alpha = 1.51$.

distribution. Figure 6(b) shows the classification accuracy versus period. Similar to the MNIST results, we see that the quantum system must be sufficiently time evolved to obtain a neural network with high expressive power. For example, at $nT = 0$, the representativeness of the QR network is poor, and only linear separation is possible. At $nT = 50$, our accuracy is raised to near 100%. For the testing accuracy, we instead generate the boundary maps from the trained network, as shown in the insets of Fig. 6(b). As the period increases, the results precisely reflect the Gaussian distribution from the training.

- [1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, and D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, *Nature (London)* **574**, 505 (2019).
- [2] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, and Y. Hu, *et al.*, Quantum computational advantage using photons, *Science* **370**, 1460 (2020).
- [3] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, and S. Guo, *et al.*, Quantum walks on a programmable two-dimensional 62-qubit superconducting processor, *Science* **372**, 948 (2021).
- [4] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, *Quant. Inf. Process.* **13**, 2567 (2014).
- [5] A. A. Ezhov and D. Ventura, *Quantum Neural Networks* (Physica-Verlag HD, Heidelberg, 2000), 213.
- [6] S. Ghosh, K. Nakajima, T. Krisnanda, K. Fujii, and Timothy C. H. Liew, Quantum neuromorphic computing with reservoir computing networks, *Adv. Quantum Technol.* **4**, 2100053 (2021).
- [7] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Opportunities in quantum reservoir computing and extreme learning machines, *Adv. Quantum Technol.* **4**, 2100027 (2021).
- [8] M. V. Altaisky, Quantum neural network, [ArXiv:quant-ph/0107012](https://arxiv.org/abs/quant-ph/0107012).
- [9] S. Gupta and R. Zia, Quantum neural networks, *J. Comput. Syst. Sci.* **63**, 355 (2001).
- [10] M. Panella and G. Martinelli, Neural networks with quantum architecture and quantum learning, *Int. J. Circuit Theory Appl.* **39**, 61 (2011).
- [11] A. Sagheer and M. Zidan, Autonomous quantum perceptron neural network, [ArXiv:1312.4149](https://arxiv.org/abs/1312.4149).
- [12] R. Zhou and Q. Ding, Quantum M-P neural network, *Int. J. Theor. Phys.* **46**, 3209 (2007).
- [13] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, *Nat. Comput. Sci.* **1**, 403 (2021).
- [14] M. Noori, S. S. Vedaie, I. Singh, D. Crawford, J. S. Oberoi, B. C. Sanders, and E. Zahedinejad, Analog-Quantum Feature Mapping for Machine-Learning Applications, *Phys. Rev. Appl.* **14**, 034034 (2020).
- [15] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum enhanced feature spaces, *Nature* **567**, 209 (2019).
- [16] Y. Xia, W. Li, Q. Zhuang, and Z. Zhang, Quantum-enhanced data classification with a variational entangled sensor network, *Phys. Rev. X* **11**, 021047 (2021).
- [17] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [18] K. Fujii and K. Nakajima, Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning, *Phys. Rev. Appl.* **8**, 024030 (2017).
- [19] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, Dynamical Phase Transitions in Quantum Reservoir Computing, *Phys. Rev. Lett.* **127**, 100502 (2021).

- [20] L. C. G. Govia, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, and T. A. Ohki, Quantum reservoir computing with a single nonlinear oscillator, *Phys. Rev. Res.* **3**, 013077 (2021).
- [21] R. Martínez-Peña, J. Nokkala, G. L. Giorgi, R. Zambrini, and M. C. Soriano, Information processing capacity of spin-based quantum reservoir computing systems, *Cogn. Comput.* (2020).
- [22] S. Ghosh, T. Krisnanda, T. Peterek, and T. C. H. Liew, Realising and compressing quantum circuits with quantum reservoir computing, *Commun. Phys.* **4**, 105 (2021).
- [23] G. B. Huang, Q. Y. Zhu, and C. K. Siew, Extreme learning machine: Theory and applications, *Neurocomputing* **70**, 489 (2006).
- [24] I. R. Rodrigues, S. R. daSilva Neto, J. Kelner, D. Sadok, and P. T. Endo, Convolutional extreme learning machines: A systematic review, *Informatics* **8**, 33 (2021).
- [25] M. P. Estarellas, T. Osada, V. M. Bastidas, B. Renoust, K. Sanaka, W. J. Munro, and K. Nemoto, Simulating complex quantum networks with time crystals, *Sci. Adv.* **6**, eaay8892 (2020).
- [26] L. Kuśmierz, S. Ogawa, and T. Toyoizumi, Edge of Chaos and Scale-Free Avalanches in Neural Networks with Heavy-Tailed Synaptic Disorder, *Phys. Rev. Lett.* **125**, 028101 (2020).
- [27] C. Uchiyama, W. J. Munro, and K. Nemoto, Environmental engineering for quantum energy transport, *npj Quantum Inf.* **4**, 33 (2018).
- [28] J. Zhang, P. W. Hess, A. Kyprianidis, P. Becker, A. Lee, J. Smith, G. Pagano, I.-D. Potirniche, A. C. Potter, A. Vishwanath, N. Y. Yao, and C. Monroe, Observation of a discrete time crystal, *Nature (London)* **543**, 217 (2017).
- [29] N. Y. Yao, A. C. Potter, I.-D. Potirniche, and A. Vishwanath, Discrete Time Crystals: Rigidity, Criticality, and Realizations, *Phys. Rev. Lett.* **118**, 030401 (2017).
- [30] V. M. Bastidas, B. Renoust, K. Nemoto, and W. J. Munro, Ergodic-localized junctions in periodically driven systems, *Phys. Rev. B* **98**, 224307 (2018).
- [31] S. Choi, J. Choi, R. Landig, G. Kucsko, H. Zhou, J. Isoya, F. Jelezko, S. Onoda, H. Sumiya, V. Khemani, C. von Keyserlingk, N. Y. Yao, E. Demler, and M. D. Lukin, Observation of discrete time-crystalline order in a disordered dipolar many-body system, *Nature (London)* **543**, 221 (2017).
- [32] P. Frey and S. Rachel, Realization of a discrete time crystal on 57 qubits of a quantum computer, realization of a discrete time crystal on 57 qubits of a quantum computer, *ArXiv:2105.06632*.
- [33] L. LeCun, C. Cortes, and C. J. Burges, *The MNIST database of handwritten digits*, <http://yann.lecun.com/exdb/mnist/index.html>.
- [34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* **15**, 1929 (2014).
- [35] J. Martyn, G. Vidal, C. Roberts, and S. Leichenauer, Entanglement and tensor networks for supervised image classification, entanglement and tensor networks for supervised image classification, *ArXiv:2007.06082*.
- [36] S. Cheng, L. Wang, and P. Zhang, Supervised learning with projected entangled pair states, *Phys. Rev. B* **103**, 125117 (2021).
- [37] S. Haykin, *Neural Networks And Learning Machines* (Pearson, London, UK, 2010), 3rd ed.
- [38] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning* (MIT Press, Cambridge, MA, USA, 2016).