

**Experimentally realizable continuous-variable quantum neural networks**Shikha Bangar,<sup>1,\*</sup> Leanto Sunny<sup>1,†</sup>, Kübra Yeter-Aydeniz,<sup>2,‡</sup> and George Siopsis<sup>1,§</sup><sup>1</sup>*Department of Physics and Astronomy, The University of Tennessee, Knoxville, Tennessee 37996-1200, USA*<sup>2</sup>*Emerging Technologies and Physical Sciences Department, The MITRE Corporation, 7515 Colshire Drive, McLean, Virginia 22102-7539, USA*

(Received 9 June 2023; revised 8 September 2023; accepted 21 September 2023; published 16 October 2023)

Continuous-variable (CV) quantum computing has shown great potential for building neural network models. These neural networks can have different levels of quantum-classical hybridization depending on the complexity of the problem. Previous work on CV neural network protocols required the implementation of non-Gaussian operators in the network. These operators were used to introduce nonlinearity, an essential feature of neural networks. However, these protocols are hard to execute experimentally. We built a CV hybrid quantum-classical neural network protocol that can be realized experimentally with current photonic quantum hardware. Our protocol uses Gaussian gates only with the addition of ancillary qumodes. We implemented nonlinearity through repeat-until-success measurements on ancillary qumodes. To test our neural network, we studied canonical machine-learning and quantum computer problems in a supervised learning setting—state preparation, curve fitting, and classification problems. We achieved high fidelity in state preparation of single-photon (99.9%), cat (99.8%), and Gottesman-Kitaev-Preskill (93.9%) states, a well-fitted curve in the presence of noise at a cost of less than 1%, and more than 95% accuracy in classification problems. These results bode well for real-world applications of CV quantum neural networks.

DOI: [10.1103/PhysRevA.108.042414](https://doi.org/10.1103/PhysRevA.108.042414)**I. INTRODUCTION**

Continuous-variable (CV) quantum computing (QC) takes advantage of the wave-like properties of particles. For example, it can be realized by photonic quantum hardware manipulating electromagnetic fields. Thus, CV QC is achievable in quantum optics by utilizing continuous quadratures of the quantized electromagnetic field [1], enabling the essential steps in quantum algorithms [preparation, unitary manipulation, and measurement of (entangled) quantum states]. CV quantum algorithms have been developed for various applications, ranging from quantum field theory [2–4] to machine learning [5]. Another interesting application of CV architecture is in calculating the quantum averages using squeezed states. This feature was studied in Ref. [6], where the advantage of quantum averaging compared to a standard arithmetic mean strategy was demonstrated.

It has recently been shown that CV is a good architecture for building quantum neural network (QNN) models on quantum computers [7]. The developed CV QNN architecture has been applied to various practical real-world problems, e.g., function fitting, fraud detection of credit card transactions, image classification of handwritten digits, data encryption and decryption in a secure cryptography algorithm [8], and entangled state detection [9]. In addition to these applications, CV QNN has been utilized as the generator and discriminator

in a CV quantum adversarial network (CV QGAN) [10] to reproduce the data outputs of the calorimeters for data collected in high-energy physics experiments at CERN. The authors of Ref. [11] utilized a quantum convolutional neural network (QCNN) for detecting quantum phase transitions in axial next-nearest-neighbor Ising (ANNNI) models. While the QCNN approach emphasized bolstering out-of-distribution generalization through a partial supervised strategy and showcased resilience against barren plateaus [12], our work diverges by focusing on CV quantum systems. This adaptation is particularly advantageous for scenarios such as optical systems, enabling seamless integration into specific experimental setups, thus streamlining hardware realization.

In the CV QNN model discussed in Ref. [7], a non-Gaussian Kerr gate provides the nonlinearity required for neural networks. However, experimentally realizing these non-Gaussian operations is challenging due to their weakly interacting nature. Therefore, we developed an alternative CV QNN model in which nonlinearities are introduced through measurements on ancillary qumodes following the proposal in Ref. [13] based on repeat-until-success measurements. In this paper, we propose a variational hybrid quantum-classical circuit implementing a neural network that solves well-known problems such as function fitting, state preparation, binary classification, and image recognition. The quantum circuit uses only Gaussian gates that can be implemented with optical elements, such as beam splitters and squeezers. Thus, our CV hybrid neural network can be realized experimentally using current photonic quantum hardware.

Similar measurements on ancillary qumodes as a means towards creating desired quantum states have been considered before. Single- and two-mode quantum gates acting on

\*sbangar@vols.utk.edu

†lsunny@vols.utk.edu

‡kyeteraydeniz@mitre.org

§siopsis@tennessee.edu

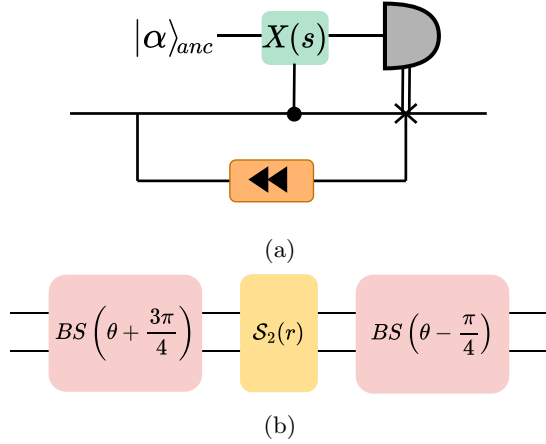


FIG. 1. Two-qumode Gaussian quantum circuit element introducing nonlinearity in a CV QNN. (a) The ancillary qumode, initially in the coherent state  $|\alpha\rangle_{\text{anc}} = \mathcal{D}(\alpha)|0\rangle_{\text{anc}}$ , is entangled with the primary qumode with a controlled displacement  $[CX(s)]$  gate. The primary qumode goes through a feedback loop until the detector on the ancillary qumode clicks [13]. (b) The  $CX(s)$  gate with parameters given in Eq. (1).

photonic qubits were generated with single photon sources, linear optical elements, and measurements of ancillary modes [14]. To generate the cubic phase gate, nonlinear quadrature measurements were implemented using ancillary states, homodyne measurements, and nonlinear feedforwards based on the measurement results [15]. Photon-number-resolving (PNR) measurements were used for the production of multimode Gaussian states, including cat (superpositions of coherent states), ON, Gottesman-Kitaev-Preskill (GKP) [16], NOON [17], and bosonic code states [18]. Cat and GKP states were also created experimentally within a CV cluster state by performing PNR measurements [19].

The structure of this paper is as follows. We begin with a description of the CV neural network in Sec. II, followed by a detailed account of our proposal to introduce nonlinearities that avoids non-Gaussian gates. In Sec. III, we study several problems involving varying degrees of hybridization between quantum and classical neural networks. We conclude with a discussion of the potential of this work in Sec. IV.

## II. THE METHOD

A general CV QNN was discussed in Ref. [7]. It included  $N$ -port linear optical interferometers consisting of beam splitter ( $BS$ ), rotation ( $\mathcal{R}$ ), displacement ( $\mathcal{D}$ ), and squeezing ( $\mathcal{S}$ ) gates. It also featured a non-Gaussian Kerr gate ( $\Phi$ ) introducing nonlinearity to the neural network. In our setup, we have replaced this single-mode gate with a two-mode quantum circuit element consisting of Gaussian gates and a photon detector, which is experimentally feasible with current technology (shown in Fig. 1).

In more detail, our nonlinear circuit element is implemented by adding an ancilla qumode in a coherent state  $|\alpha\rangle = \mathcal{D}(\alpha)|0\rangle_{\text{anc}}$  to the primary qumode, where  $\alpha \in \mathbb{R}$ . The two modes are then entangled by a controlled displacement  $[CX(s)]$  gate that uses the  $q$ -quadrature of the primary mode to

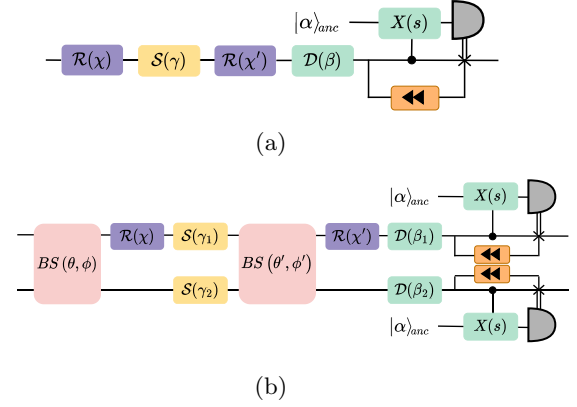


FIG. 2. Detailed CV neural network architecture for (a) single-mode and (b) two-mode layer.

shift the  $q_{\text{anc}}$  quadrature by  $sq$ , where  $s \in \mathbb{R}$ . It is implemented with beam splitters and a two-mode squeezer of parameters

$$\cot 2\theta = \sinh r = -s. \quad (1)$$

If the incoming state of the primary mode is  $|\psi\rangle = \int dq \psi(q) |q\rangle$ , then the final two-mode state is the entangled state

$$\int dq \psi(q) |\alpha + sq\rangle_{\text{anc}} \otimes |q\rangle. \quad (2)$$

Then we use a photon detector to measure the photon number of the ancilla qumode. If the detector clicks, the process is considered successful, and the primary qumode proceeds to the next layer. For small  $\alpha$  and  $s$  parameters, the ancilla qumode decouples and the outgoing primary qumode is in the (unnormalized) state

$$\int dq (\alpha + sq) \psi(q) |q\rangle, \quad (3)$$

showing that we have effectively applied the non-Gaussian gate  $\alpha + sq$ . For larger values of the parameters, the expression for this effective gate is more complicated, but it is still a non-Gaussian gate.

If there is no click in the detector, the output state of the primary mode is approximately the same as the input state,  $|\psi\rangle$  for small  $\alpha$  and  $s$ , changing slightly for larger values of the parameters (with insertions of factors of  $e^{-|\alpha+sq|^2/2}$ ). We feed the state back into the input port and repeat the process shown in Fig. 1. This loop continues until the detector clicks and the primary qumode can advance (repeat-until-success process [13]). We fix the coherence parameter  $\alpha$  of the ancillary qumode, treating it as a hyperparameter and not a trainable parameter. In our simulations, we set  $\alpha = 1$ .

A detailed neural network architecture for single-mode and two-mode layers is shown in Fig. 2. We create a multiple-layer structure by arranging each layer as a building block of the neural network with the gate variables  $(\theta, \phi, \chi, x, \alpha, \beta)$  being free parameters, collectively denoted by  $\vec{\zeta}$ . We want to find  $\vec{\zeta}$  such that the value of the cost function  $[C(\vec{\zeta})]$  is minimum. This can be done using various optimization techniques available in deep learning (for example, gradient descent, stochastic gradient descent, and, most commonly, the

Adam optimizer). The parameters are updated by the rule

$$\vec{\zeta} \longrightarrow \vec{\zeta} + \eta \nabla_{\vec{\zeta}} C(\vec{\zeta}), \quad (4)$$

where  $\eta$  is the step size, also known as the learning rate. This procedure continues until the model cannot modify the cost function further. Then, the parameters associated with the lowest observed cost function offer the best solution to the given task. Note that the free parameters correspond to a circuit; hence, the circuit is the solution.

In principle, only one layer ( $\mathcal{L}$ ) is sufficient to parametrize every possible unitary affine transformation on  $N$  modes. However, deeper architectures provide increased expressive power, better learning capabilities, and a more efficient representation of complex transformations.

To test our neural network, we studied various machine-learning and quantum computation problems, namely, state preparation, curve fitting, binary classification, and image recognition with varying degrees of hybridization in quantum and classical neural networks, as discussed in Sec. III.

We conducted our simulations using STRAWBERRY FIELDS quantum computing software. The realization of the quantum circuit element implementing nonlinearity depicted in Fig. 1 in a QNN setting turned out to be challenging. To obtain a good approximation to our setup that could be implemented with software available to us, we adjusted the coherent parameter of the ancillary qumode so that a single photon would be detected in that mode with a high success probability. To estimate the number of ancillary qumode measurements (or feedback loops) necessary for the successful application of the quantum circuit element, we utilized BOSONIC QISKIT software [20] for simulating an elementary circuit and tracking the number of measurements required for success in various applications.

Sample results are illustrated in Fig. 3. Using the architecture of the binary classification circuit discussed in Sec. III C as a concrete example, we counted the number of repeated ancillary qumode measurements required for a successful pass of the primary qumodes during forward propagation through the network. We plotted the success rates per layer for setups utilizing various numbers of layers. Evidently, most successful measurements occur at the first photon detection of the ancillary qumode.

Due to the constraints in the STRAWBERRY FIELDS library, we were compelled to perform postselection of a single Fock state during the ancilla measurement step to collapse the wave function, thus effectively implementing measurements by photon-number-resolving detectors. It would also be interesting to simulate photon detectors that cannot resolve photon number and are widely available. This would alter the effective nonlinear operation slightly (depending on the choice of the laser intensity for the ancillary qumodes) but would simplify the experimental setup. To demonstrate that a high success rate can be achieved even without a photon-number-resolving detector, we trained two distinct models that performed multilabel classification on the MNIST handwritten digit data set discussed in Sec. III D. The training and testing loss values for these classical-quantum hybrid models are presented in Fig. 4. Model 1 was designed to successfully select the required state in the ancilla measurement

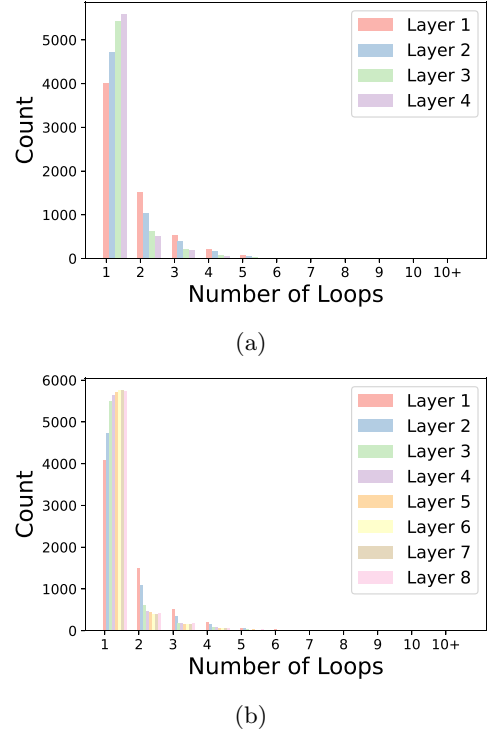


FIG. 3. Number of measurements necessary on the ancillary qumode to achieve successful measurement for each quantum layer of sample hybrid networks for binary classification, consisting of two classical and (a) four or (b) eight quantum layers.

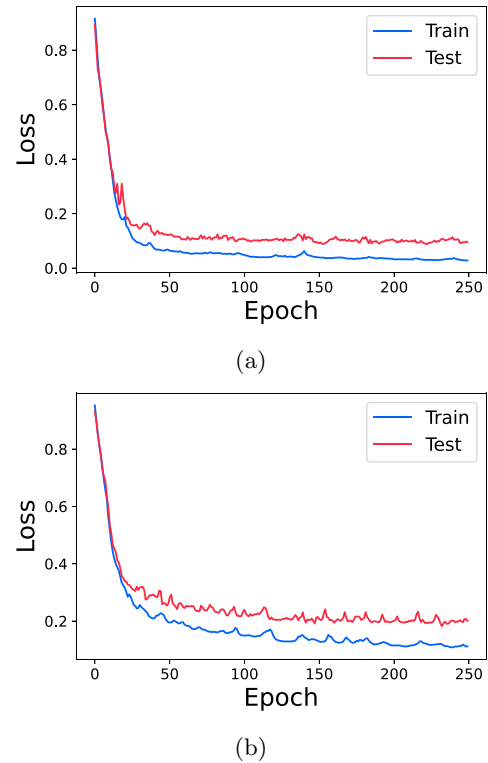


FIG. 4. Training and testing loss of MNIST classification models with successful measurement of the ancilla within (a) one feedback loop (Model 1) and (b) two feedback loops (Model 2).

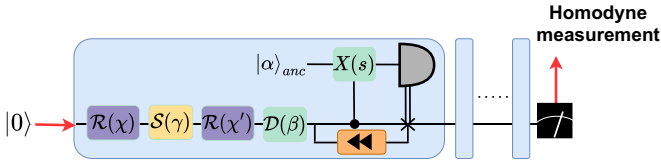


FIG. 5. CV QNN architecture for quantum state preparation. The input to the network is the vacuum state, and at the end, a homodyne measurement is performed.

on the initial attempt, whereas Model 2 was set up to fail the first measurement attempt and succeed on the second measurement. Even though the loss function and classification accuracy experienced a slight decline at the end of training from 97.25% to 96.49%, we were still able to train Model 2 successfully and achieve a reasonably high classification accuracy for the four-class MNIST handwritten digit data set classification.

### III. CASE STUDIES

In this section, we develop CV QNN models for various applications, including quantum state preparation (Sec. III A), curve fitting (Sec. III B), binary classification of fraud and genuine credit card transactions (Sec. III C), and multilabel classification of MNIST handwritten digits (Sec. III D). In order to observe the effect of the classical and quantum neural network layers, we analyze both hybrid quantum-classical and fully quantum layers.

We simulated our CV QNN models using the STRAWBERRY FIELDS software platform [21]. The quantum machine-learning toolbox application is built on top of it with TENSORFLOW features [22]. We used the quantum circuit simulator, optimized the algorithm, and trained the neural network to obtain the desired results.

#### A. State preparation

The CV QNN model for quantum state preparation trains a quantum circuit to generate a target quantum state. To this end, we provide a canonical input state  $|\Psi_0\rangle$  and target output state  $|\Psi_t\rangle$  and aim to find out the circuit  $U$  (a unitary transformation) such that

$$|\Psi_t\rangle = U |\Psi_0\rangle. \quad (5)$$

For simplicity, we fixed the input state to be the vacuum,  $|\Psi_0\rangle = |0\rangle$ . We considered a basic single-mode architecture of a quantum neural network with a fixed number of layers, as shown in Fig. 5. As described earlier, our goal was to find parameters  $\vec{\zeta}$  determining a unitary  $U(\vec{\zeta})$  such that  $U(\vec{\zeta})|0\rangle = |\Psi_t\rangle$ , or equivalently,  $|\langle\Psi_t|U(\vec{\zeta})|0\rangle| = 1$ . The fidelity of the output states is defined in terms of the overlap between the target and optimized states as

$$F = |\langle\Psi_t|U(\vec{\zeta})|0\rangle|^2. \quad (6)$$

Then, the cost function to be minimized can be defined in terms of the fidelity as

$$C = |\sqrt{F} - 1| = ||\langle\Psi_t|U(\vec{\zeta})|0\rangle| - 1|. \quad (7)$$

It should be noted that a slightly different definition of the cost function has been used before,  $C = |\langle\Psi_t|U(\vec{\zeta})|0\rangle - 1|$ , which involves the phase of the output state. Including the phase adds flexibility to the search space determining the unitary  $U(\vec{\zeta})$  which is lost if our definition (7) is used. However, this turned out not be an impediment in all cases we considered. The advantage of using the definition (7) is that the cost function can be deduced experimentally through the Wigner function by performing homodyne measurement on the output state at each optimization step.

In detail, to obtain  $C$ , we perform homodyne tomography on the output state  $U(\vec{\zeta})|0\rangle$ . We measure the quadrature  $X_\phi = \frac{1}{2}(e^{i\phi}a^\dagger + e^{-i\phi}a)$ , and obtain a series of output pairs  $(\phi_k, x_k)$  ( $k = 1, \dots, N$ ). They allow us to obtain an estimate of the Wigner function of the final state,  $W_{U(\vec{\zeta})|0}(q, p)$ . By comparing with the Wigner function of the desired state,  $W_{|\Psi_t\rangle}(q, p)$ , we deduce the cost function  $C$  from the fidelity (overlap)

$$F = 2\pi \int dq dp W_{U(\vec{\zeta})|0}(q, p) W_{|\Psi_t\rangle}(q, p). \quad (8)$$

To test the performance of the QNN, we prepared two different states, the single-photon state  $|\Psi_t\rangle = |1\rangle = a^\dagger|0\rangle$  and the cat state

$$|\text{cat}, \theta\rangle = \frac{1}{\sqrt{2}}[\mathcal{D}(\alpha_0) + e^{i\theta}\mathcal{D}(-\alpha_0)]|0\rangle, \quad (9)$$

where  $\theta, \alpha_0 \in \mathbb{R}$ .

Since we were using STRAWBERRY FIELDS software, we could quickly get the Wigner function after performing the homodyne measurement. However, these Wigner functions are incompatible with the commonly used efficient TENSORFLOW framework to perform the optimization. Hence, we used the PYTHON library Scipy [23] to optimize cost, as Scipy provided the Nelder-Mead optimization technique [24]. This technique is nongradient based and designed for high-dimensional minimization, which worked best for our purposes as our goal was to minimize the cost built from two-dimensional (2D) Wigner functions obtained via homodyne detection.

We noticed that just in a few steps, the model started to learn the state. For best results, we ran the model with a different number of layers, as shown in Fig. 6. In these simulations, we did not include any possible quantum hardware errors. A small number of layers yields a higher cost, but fewer layers require fewer gates and lead to fewer errors due to quantum hardware imperfections. Ignoring such errors, as the number of layers increases, the cost is lowered. Notice that the cost starts to increase again beyond a certain number of layers due to overfitting, e.g., for more than six layers in the case of the single-photon state. There is an optimum number of layers which would be important to determine by including a realistic model of quantum hardware.

For the single-photon state, six quantum layers gave the best results with a fidelity of 99.9%. The cost achieved after optimization was 0.008 after 5000 steps. We used a cutoff dimension of 6, and the maximum number of steps was fixed at 5000. It should be noted that in the Nelder-Mead optimization, the number of steps is determined by the difference between two consecutive cost values. This difference is treated as another hyperparameter during the training process. The result of comparing other numbers of layers is shown in



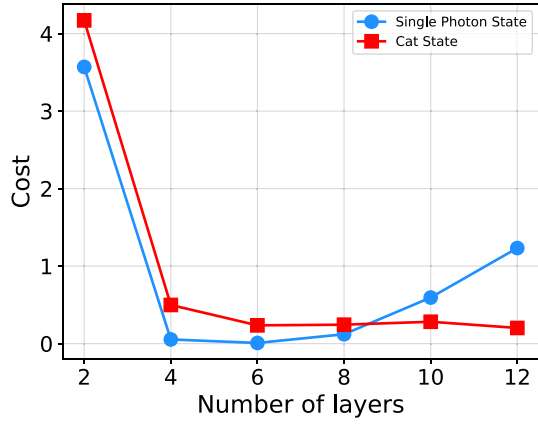


FIG. 6. Value of cost function for different number of layers for preparation of single-photon (blue circles) and cat (red squares) states. Values averaged over five independent runs. Cutoff dimension was set to 6 (10) for the single-photon (cat) state.

Fig. 7. We plotted the 2D Wigner function for two and six layers to demonstrate the importance of the optimal number of layers. We also showed the one-dimensional (1D) Wigner function for different quantum layers. We integrated the 2D Wigner function over the momentum using the Scipy library to obtain the 1D Wigner function. The plot shows that the results improve up to six quantum layers, and beyond that, they worsen due to overfitting.

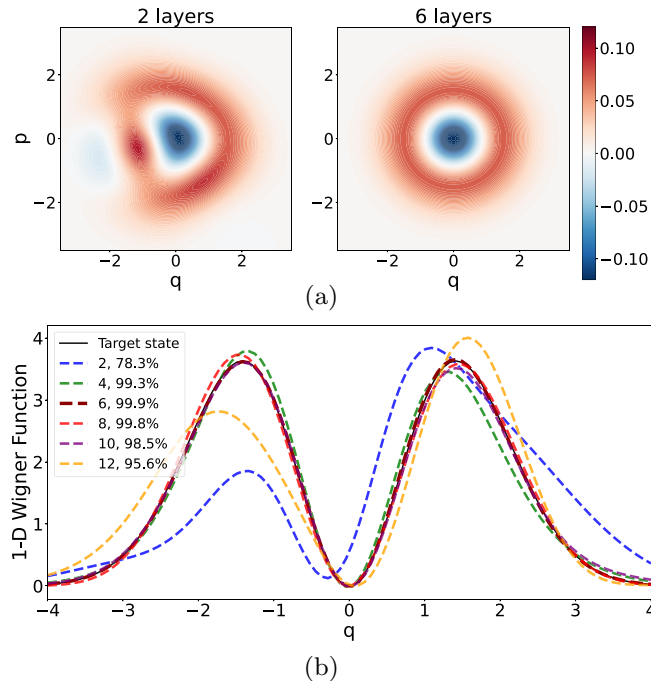


FIG. 7. Study of the effect of changing the number of layers of the QNN for the preparation of a single-photon state. (a) 2D Wigner function comparison for two and six quantum layers, with fidelities 78.3% and 99.9%, respectively. (b) 1D Wigner function for various numbers of quantum layers. Six quantum layers produce the best fidelity. The maximum number of steps was kept fixed at 5000 and the cutoff dimension at 6.

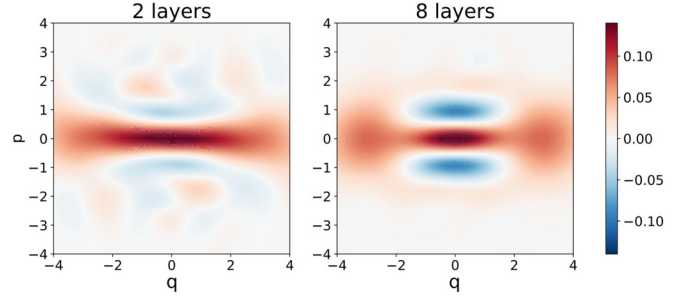


FIG. 8. Preparation of an even cat state [Eq. (9) with  $\alpha_0 = 1.5$ ,  $\theta = 0$ ]. 2D Wigner function comparison for two and eight quantum layers with fidelity 79% and 99.8%, respectively. A cutoff dimension of 10 was used in both cases. Two (eight) layers were optimized in 2000 (9800) steps, resulting in a cost of 2.64 (0.03).

Turning to state preparation of the cat state (9), which is a superposition of two coherent states, we concentrated on the even cat state with  $\alpha_0 = 1.5$  and  $\theta = 0$ . As the cat state is more complicated than the single-photon state, we had to increase the cutoff dimension to 10. We achieved high fidelity with eight quantum layers. The comparison of the 2D Wigner functions of two and eight layers is shown in Fig. 8. We obtained excellent fidelity of 99.8% with eight layers, and after training, the cost was minimized to the value of 0.03 after 9800 steps. For comparison, with two layers, we obtained fidelity of 79% with cost at 2.64 after 2000 steps.

We also prepared a realistic GKP state [16]. Ideal GKP states are linear combinations of an infinite number of eigenstates of the  $q$ -quadrature. We concentrated on the state

$$|\text{GKP}\rangle_{\text{ideal}} = \sum_{n=-\infty}^{\infty} |q = 2n\sqrt{\pi\hbar}\rangle. \quad (10)$$

However, such states are not normalizable and impossible to create experimentally because they have infinite energy and each component would require an infinite amount of squeezing. For a realistic case, we applied an energy cutoff and defined the realistic GKP state [25]

$$|\text{GKP}\rangle_{\text{real}} = e^{-\epsilon a^\dagger a} |\text{GKP}\rangle_{\text{ideal}}. \quad (11)$$

We chose  $\epsilon = 0.1$ . Since this state is more complex than a cat state, we had to increase the cutoff dimension even further to 15 and employ 15 layers. After 15 000 steps, we achieved a fidelity of 93.9% at a cost of 1.1. The comparison between 10 and 15 layers is shown in Fig. 9.

Comparing with the fidelity of single-photon and cat states, it should be noted that the cost for the number of layers depicted in Fig. 6 is larger than 4. To lower the cost to levels comparable with single-photon and cat states, more than 12 layers are needed. Due to computational resource limitations, we could only go up to 15 layers. Considering more layers will improve the results for GKP state preparation as the number of learning parameters would increase. Another critical factor affecting the complexity of the GKP state preparation is the decay parameter ( $\epsilon$ ). The smaller this parameter is, the closer we are to the ideal GKP state. Our computational resources allowed us to go as low as  $\epsilon = 0.1$ .

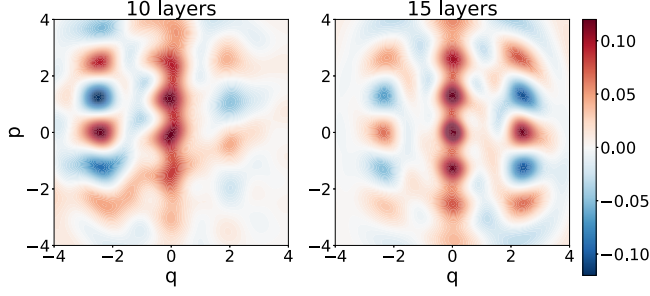


FIG. 9. Preparation of a realistic GKP state [Eq. (11)] with  $\epsilon = 0.1$ . 2D Wigner function comparison for 10 and 15 quantum layers with fidelity 70.6% and 93.9%, respectively. A cutoff dimension of 15 was used in both cases. Ten (15) layers were optimized in 18 000 (15 000) steps, resulting in a cost of 4.87 (1.1).

To understand the effects of loss on state preparation, we applied a loss channel operation to every qumode at the completion of each quantum layer. This operation connects a given qumode, created by  $a^\dagger$ , with an environmental qumode, created by  $b^\dagger$ , which is initially in the vacuum state. At the end of the operation, the latter is traced out. The corresponding transformation is modeled by a beam splitting operation,

$$a \mapsto \sqrt{T}a + \sqrt{1-T}b, \quad (12)$$

where  $T$  represents the energy transmissivity. The effect of this operation, leading to a decrease in overall fidelity due to the introduction of loss in our simulations, is depicted in Fig. 10. The average fidelity remains notably high for some prepared states under a realistic loss scenario with  $T = 0.9$ . As shown in Fig. 11, the single photon state exhibits exceptional performance, achieving an average fidelity of 92%, down from peak fidelity of 96% at  $T = 1$  (no loss). Cat states drop to an average fidelity of 74%, down from a peak of 88% in the ideal case. The GKP state exhibits the lowest performance, with average fidelity at  $T = 0.9$  dropping to 69%, showing that a lot more layers are needed for a decent performance.

### B. Curve fitting

Next, we build a CV QNN in a supervised learning setting to learn the relationship between input ( $x$ ) and output [ $f(x)$ ], also known as curve fitting. It is an essential part of

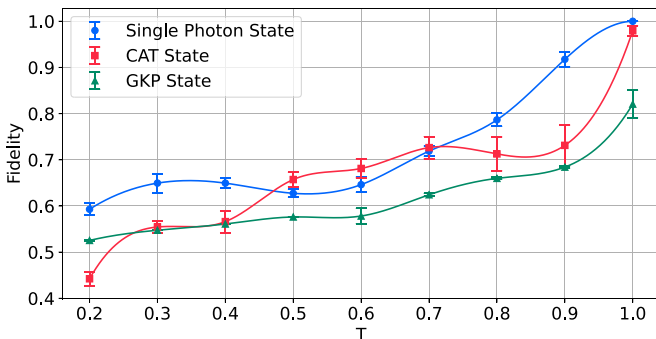


FIG. 10. Impact of transmissivity ( $T$ ) on fidelity for single-photon, cat, and GKP states.

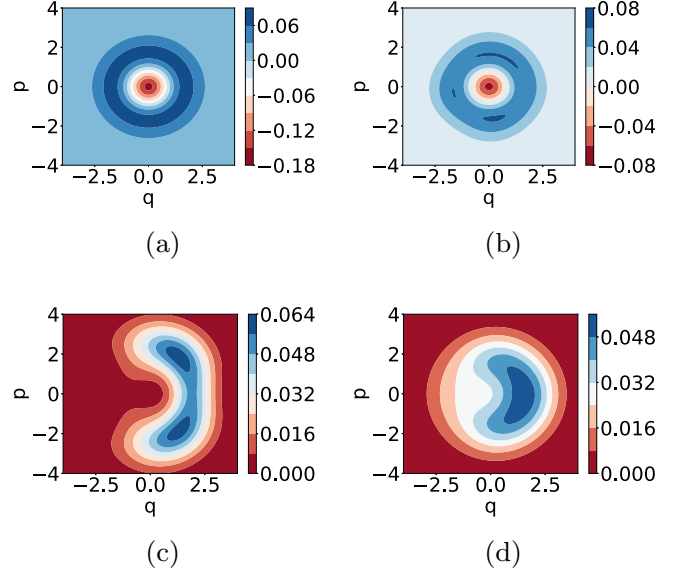


FIG. 11. Wigner functions of the single-photon state at transmissivity ( $T$ ) values of (a)  $T = 1$ , (b)  $T = 0.9$ , (c)  $T = 0.5$ , and (d)  $T = 0.2$ , respectively.  $T = 1$  corresponds to the noiseless case.

data analysis and a classic machine-learning problem. The architecture for our CV QNN is shown in Fig. 12. We encoded the classical input  $x$ , sampled from a noisy function  $f(x)$ , as the coherence parameter of the input qumode,  $|x\rangle = \mathcal{D}(x)|0\rangle$ . The objective was to train the CV QNN to generate output states  $|\psi_x\rangle$  that have an expectation value of the quadrature  $q$  close to  $f(x)$  [i.e.,  $\langle\psi_x|q|\psi_x\rangle \approx f(x)$ ] for a given input  $x$ . We studied the noisy sine function. The data were prepared as  $f(x) = f_0(x) + \Delta f$ , where  $\Delta f$  is a normal distribution with zero mean and standard deviation  $\epsilon$ . The parameter  $\epsilon$  determines the amount of error present in the training data. We chose the noisy sine function with  $f_0(x) = \sin x$  in the range of  $x \in (-2, 2)$ . We used six quantum layers in this process. The training was done on 1000 steps with a Hilbert-space cutoff dimension of 6. The training and test data were prepared as tuples  $[x_i, f(x_i)]$ , and  $x_i$  was chosen uniformly at random in the chosen interval. For training, we chose the cost function to be the mean square error (MSE) value between the circuit outputs and the desired function values,

$$C = \frac{1}{N} \sum_{i=1}^N [f(x_i) - \langle\psi_{x_i}|q|\psi_{x_i}\rangle]^2. \quad (13)$$

To learn about the performance of CV QNN, we studied how the cost would change if we increased the number of layers. This helped us determine the optimum number of

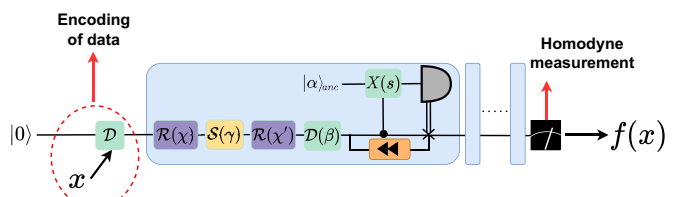


FIG. 12. CV QNN architecture for curve fitting.

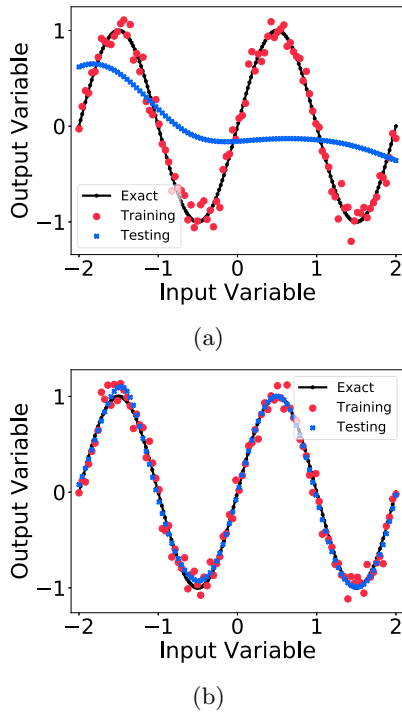


FIG. 13. The effect of changing the number of layers on curve fitting of a noisy sine function: (a) two layers, cost = 0.318, and (b) six layers, cost = 0.008. 100 data points were used for training and testing in 1000 steps with error parameter  $\epsilon = 0.1$  and cutoff dimension 6.

layers required for the desired results. We started with a single quantum layer and increased the number of quantum layers up to ten. We kept the number of steps fixed at 1000 with 100 data points and a cutoff dimension of 6. We used the Adam optimizer to minimize the cost function value. Some interesting results of how testing data behaved with a changing number of layers are shown in Fig. 13. The final result of the study is summarized in the cost vs the number of layers in Fig. 14. We found that the cost function value decreases as we increase the number of layers, making the curve fit better. However, this improvement saturates around six quantum layers. One also has to keep in mind that more layers corresponds to a greater number of training parameters. Hence, finding a number that yields good results and keeps the training parameters manageable is important. In the case considered here, optimal results were obtained with six layers.

We also studied how the noise present in data affected our results by varying the error parameter  $\epsilon$  discussed above. We kept the number of steps fixed at 1000 with 100 data points, a cutoff dimension of 6, and six quantum layers. The results of  $\epsilon = 0.2, 0.5$  are shown in Fig. 15. The value of cost increases from 0.037 to 0.232 as we increase the error from  $\epsilon = 0.2$  to  $\epsilon = 0.5$ , and the fitting worsens as we increase the noise; this is expected as the model is training on noisy data. Although the fitting is getting worse with increased noise, the CV QNN still performs well in learning the shape of the sine function. The complete study of the dependence of the cost function on data noise ( $\epsilon$ ) is shown in Fig. 16.

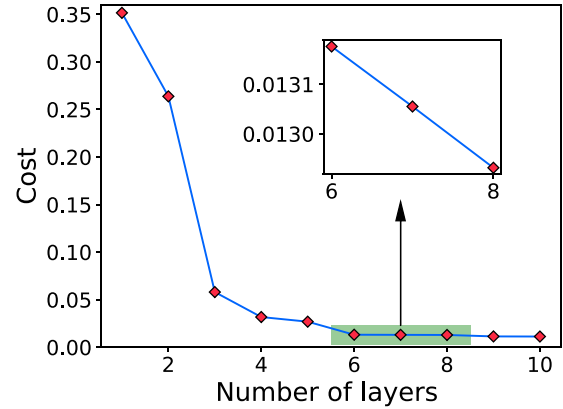


FIG. 14. Cost function values vs the number of layers plot obtained after performing learning on our CV quantum neural network model for fitting the noisy sine function curve. Each data point in the scatter plot corresponds to the five independent runs average. The training was done for 1000 steps with 100 data points (with  $\epsilon = 0.1$ ) and a cutoff dimension of 6.

### C. Binary classification

For the third problem, we constructed a CV quantum-classical hybrid neural network to demonstrate its effectiveness in detecting fraudulent transactions on credit card purchase data. This is a binary classification problem, a canonical problem in machine learning. The main reason for

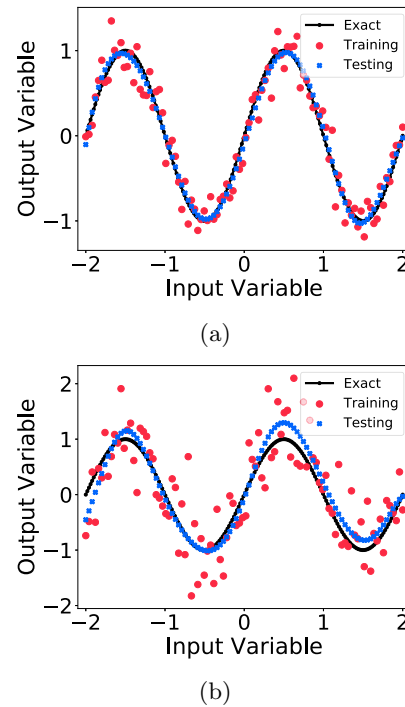


FIG. 15. The effect of increasing the error in the data on curve fitting of a noisy sine function: (a)  $\epsilon = 0.2$  and cost = 0.037, and (b)  $\epsilon = 0.5$  and cost = 0.232. 100 data points were used for training and testing in 1000 steps with cutoff dimension 6 using six quantum layers.

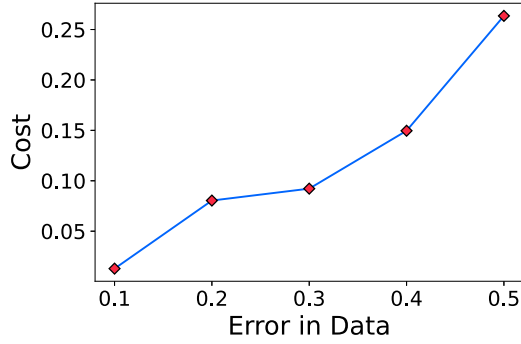


FIG. 16. Plot of cost function value vs noise in data determined by the parameter  $\epsilon$ , obtained after learning on a QNN to fit the noisy sine function curve. Each data point in the scatter corresponds to the average of five independent runs. The training was done for 1000 steps with 100 data points, a cutoff dimension of 6, and using six quantum layers.

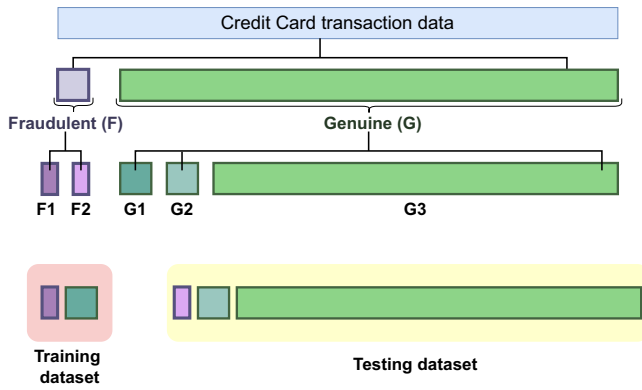


FIG. 17. The description of data preparation done for the binary classification problem. Note that the smaller parts of fraudulent and genuine transactions are denoted by the letters F1 and F2 and G1 and G2, respectively. The size of G1 and G2 are equal but three times more than F1 and F2 because of 3:1 undersampling. In the end, the training data set consists of F1 and G1, and the testing data set consists of F2, G2, and G3.

including the classical layers in this problem is that we want to encode the data with the help of classical layers.

The credit card transaction data is taken from Kaggle [26], a publicly available database. Each transaction was flagged as either genuine or fraudulent with 28 features. Only 0.172% of transactions were fraudulent out of a total number of 284 807.

First, we split the data into training and testing parts. In the training data set, we undersampled the genuine transactions by selecting them randomly and ensuring that the genuine-to-fraudulent transaction ratio was 3:1. All the remaining genuine transactions were added to the test data set. This data preparation is explained in detail in Fig. 17.

The network architecture is shown in Fig. 18. Four fully connected feed-forward classical layers are followed by five quantum layers with four modes (two are ancillary qumodes). The credit card data is fed into the first classical layer of size 10, followed by two hidden layers of the same size. The last classical layer of size 12 controls the gate parameters in the first quantum input layer. This layer marks the beginning of the quantum part of the neural network. And because we are letting the last classical layer control the gate parameters, the quantum layer is the encoding layer. We start with four vacuum qumodes. These layers contain two single-mode squeezing gates  $S$ , one interferometer gate  $U$ , two displacement gates  $D$ , and two CX gates, which provide nonlinearity through measurement on the ancilla qumode. At the end of the encoding quantum layer, a photon number measurement is performed on the two ancillary qumodes. The two primary qumodes are allowed to advance when the detectors on the ancillary qumodes click. If they do not click, the main qumodes are fed back into the quantum circuit, as shown in Fig. 18. The feedback loops are repeated until the detectors on the ancillary qumodes click, thus implementing the desired nonlinearity in the CV quantum neural network [13]. We repeat this process for four more hidden layers. Finally, we measure the photon number on the two output primary qumodes that emerge after the last quantum output layer. If we find the photon in the first qumode, we call it a genuine transaction; if we find it in the second qumode, it is a fraudulent transaction.

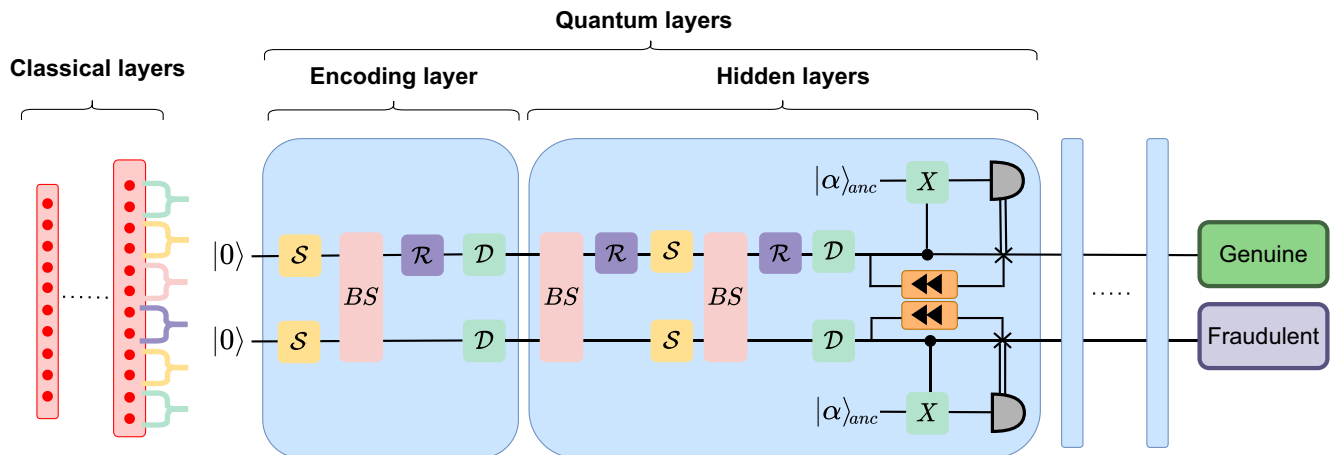
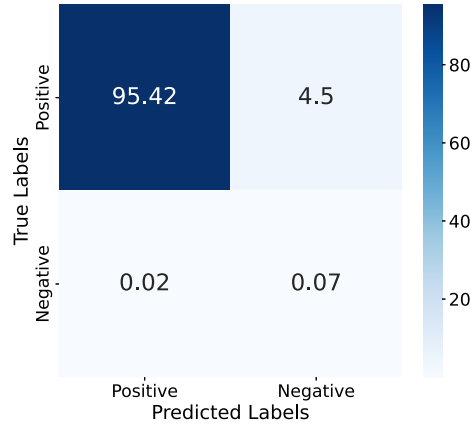
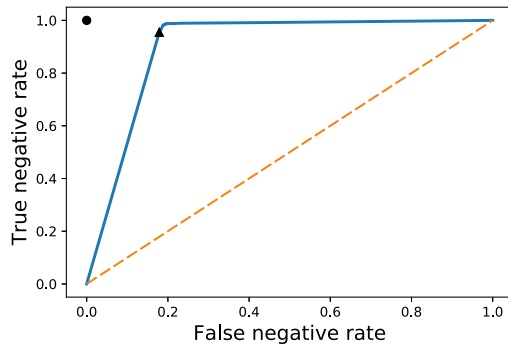


FIG. 18. CV hybrid NN for fraudulent transaction detection in credit card data. The parameters of each gate in the encoding layer are obtained from the values of the last classical layer.





(a)



(b)

FIG. 19. Results of fraud detection in credit card transactions: (a) confusion matrix showing accuracy of 95.48%, and (b) ROC curve with area under it 0.9. The CV hybrid NN had two classical and four quantum hidden layers.

The training was performed using the Adam optimizer with a batch size 24. We minimized the cost function defined by

$$C = \sum_{i \in \text{data}} (1 - p_i)^2, \quad (14)$$

where  $p_i$  is the probability of detecting a photon for input  $i$  in the correct mode. We used a cutoff dimension of 8 in each mode for 10 000 batches. Once the model was trained, we tested it by choosing a threshold probability closest to the optimal receiver operating characteristic (ROC) required for a transaction to be classified as genuine.

There are various ways to measure the accuracy of binary classification problems; for this case, we used a confusion matrix and ROC curve. The confusion matrix is a table containing four actual and predicted value combinations. The four quadrants of the confusion matrix contain results for true positive (TP), false negative (FN), false positive (FP), and true negative (TN) events. Various quantities of interest can be deduced from the confusion matrix, such as accuracy, precision, recall, and specificity. We calculated accuracy using the matrix

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (15)$$

The confusion matrix is shown in Fig. 19. The accuracy of the model calculated from the confusion matrix came out to be

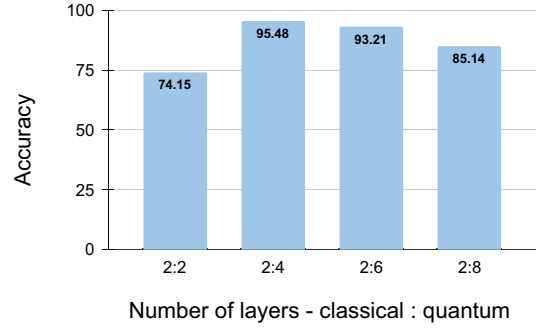


FIG. 20. Plot showing the effect of changing the number of classical and quantum layers in a hybrid CV NN on the accuracy of the results for binary classification.

95.48%. The TP quadrant shows that the model predicts the genuine data correctly more than 95% of the time. The number in the second quadrant representing FN events appears to be high. However, credit card companies can alert their users about such transactions, and by verifying these transactions, the FN rate can be brought down. The essential quadrant to consider is the third one that represents FP events, i.e., the fraudulent transactions that are incorrectly predicted as genuine. Fortunately, this number is very low for the trained model. Also, the testing data set (which was used to plot the confusion matrix) contains a tiny number of fraudulent transactions, all identified almost correctly, as the percentage of fraudulent transactions in the testing data set matches the fourth quadrant.

The ROC curve is an essential tool in the analysis of a classification model. It shows the trade-off between the TP and FP rates at different threshold values. It can help determine the optimal threshold for a given model and aid in evaluating the model's overall performance. The solid black circle in Fig. 19(b) represents the ideal binary classifier, at TN rate of 1 and FN rate of 0, whereas the solid black triangle is the closest point to the optimal within the chosen threshold. The area under the curve (AUC) is 0.90, which is close to the ideal value of 1. The AUC value is a good measure of the separability of the data being classified.

It should be pointed out that the number of features of the credit card transaction data we are interested in is equal to the number of parameters in the quantum circuit used for data encoding. Therefore, we investigated the role of the classical layer in this hybrid quantum-classical neural network architecture. We kept the classical layers constant at 2 and changed the quantum layer at an increment of 2. The results are plotted as accuracy calculated from the confusion matrices vs different layers in Fig. 20. There is an optimal number of hybrid layers scenario which give out the best results. When we use two classical and two quantum layers, the accuracy is around 74%, meaning there are not enough layers for learning. However, when we increased the quantum layers to eight, the accuracy went slightly down, indicating the overtraining for a simple case of binary classification. We also understand that hyperparameters play a major role in training, but we found this for the set of hyperparameters we chose.

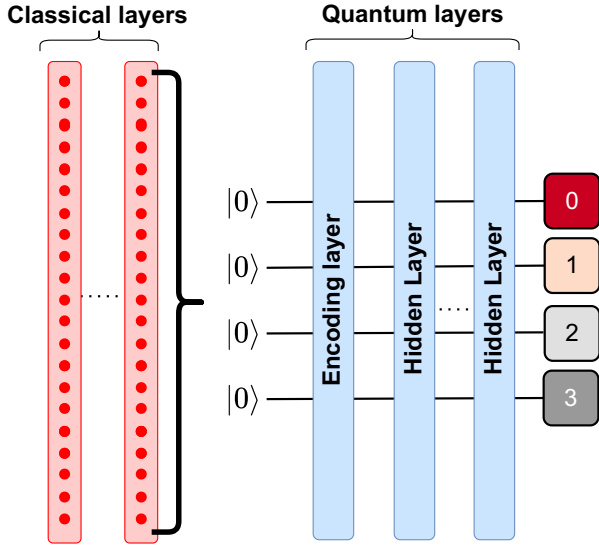


FIG. 21. CV hybrid NN architecture used for MNIST image classification.

#### D. Multilabel classification

Extending the results for binary classification, we developed a CV quantum-classical hybrid NN to classify MNIST handwritten digits [27] into their respective classes. The MNIST data set comprises 60 000 training images and 10 000 testing images, each normalized to  $28 \times 28$  pixels in size and grayscale in color. Each data point is labeled with the corresponding digit (0–9). Our current hardware limitations allowed our model to train and classify images up to four classes (0–3).

The network architecture we used is illustrated in Fig. 21 with details of the encoding layer in Fig. 22(a) and the quantum layers in Fig. 22(b). The network consists of fully connected feed-forward classical neural network layers that take the input data, feeding into an encoding quantum layer, followed by regular quantum layers that can be repeated as needed. Each quantum layer comprises several primary qumodes, each representing a class of the MNIST data set, as well as ancillary qumodes that implement nonlinearity.

During training, we calculated the probability or accuracy of classification by considering the overlap of the final state of the circuit, which comprises only the primary qumodes, with the one-hot encoded ground truth value of the training data sample. We counted all probabilities corresponding to nonzero values of the Fock number of the correct class towards the accuracy, indicating a “click” or “nonclick” on a detector. The loss was then calculated using Eq. (14). For training, we used the Adam optimizer with a batch size of 16 and a decaying learning rate beginning at 0.001 and decreasing by a factor of 0.9 every 5000 steps. During validation or testing runs, we interpreted the probabilities corresponding to each primary qumode as logits. The predicted class was determined by selecting the logit with the highest value.

To assess the versatility of our model, we performed multiple experiments, exploring different configurations of classical and quantum layers in the hybrid model. We considered a range of ratios, starting from one classical layer

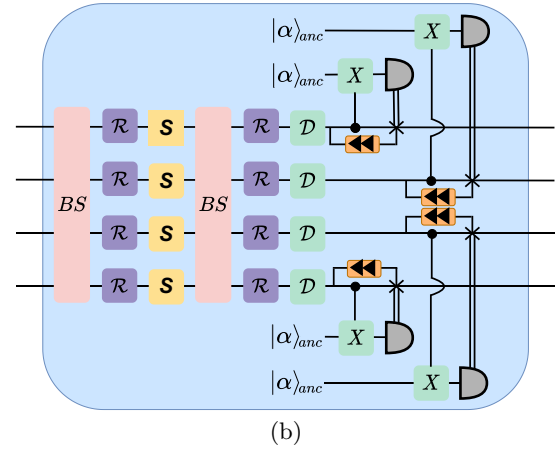
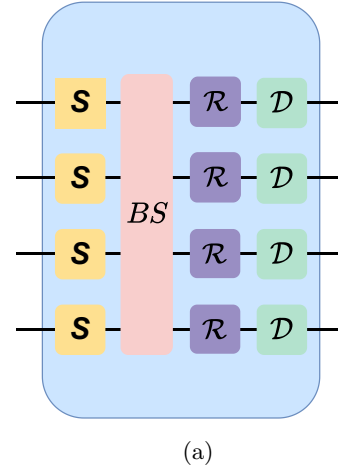


FIG. 22. MNIST hybrid NN. (a) Encoding layer of MNIST hybrid neural network that embeds the output of the classical layers into the quantum circuit. (b) Detailed architecture of the quantum layer.

and five quantum layers (1:5) up to five classical layers and one quantum layer (5:1). In total, we examined eight different layer ratios, including ratios such as 2:2, 2:6, and 2:8. The classical layers comprised 128 nodes each, except for the final layer (or only layer in the case of a single classical layer model), which had the same number of nodes as required by the quantum encoding layer. The latter is determined by the formula  $7p - 2$ , where  $p$  denotes the number of classes or primary qumodes.

The loss function values over the epochs for select ratios of classical and quantum layers are displayed in Fig. 23. All of our models converged within 100 epochs and demonstrated successful training and testing, achieving testing accuracies of  $96.47\% \pm 0.86\%$ . Interestingly, we did not observe a significant impact on changing the number of classical and quantum layers. Nevertheless, our hybrid networks demonstrated high levels of accuracy for the four-class MNIST classification problem.

#### IV. CONCLUSION

We proposed CV QNN models that can be realized experimentally. We introduced a quantum circuit element that

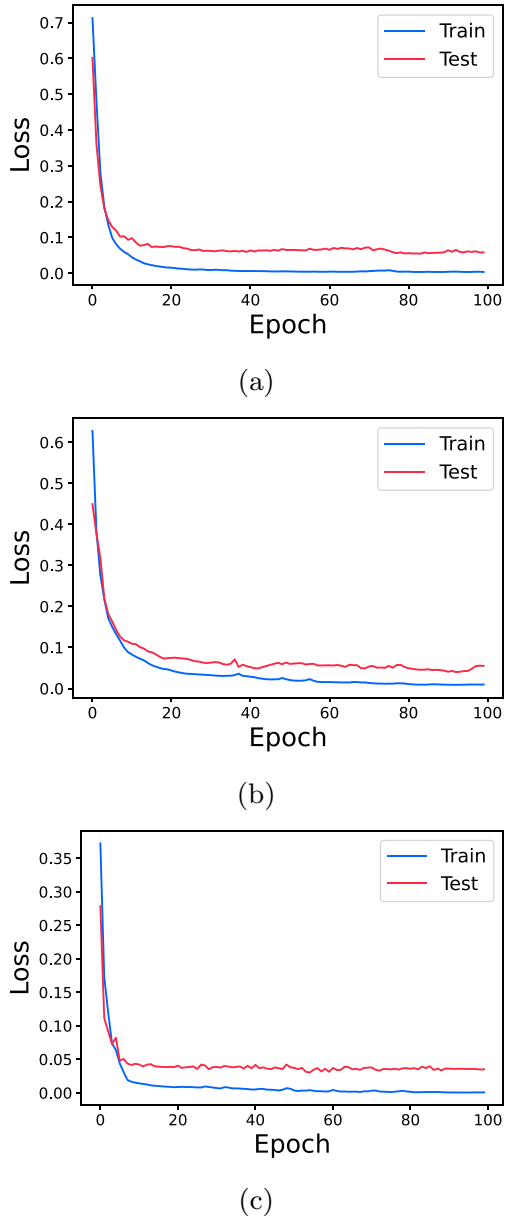


FIG. 23. Training and testing loss plots for select ratios of classical and quantum layers, networks (a) 1:5, (b) 3:3, and (c) 5:1, with the first (second) number indicating the number of classical (quantum) layers.

involves an ancillary qumode with a controlled-X, i.e., CX, gate on the primary qumode. For a good success rate, it relies on repeat-until-success measurements of the photon number of ancillary qumodes [13]. It offers a simple and feasible solution to introducing nonlinearity using current photonic quantum hardware, considering the high complexity of implementing non-Gaussian operators experimentally. Our study demonstrated that our experimentally realizable circuit element could efficiently solve many machine-learning and quantum computation problems.

It would be interesting to test our CV quantum algorithms on photonic quantum hardware. Unfortunately, currently available cloud-accessed photonic quantum devices

have limitations that prevent implementation of the algorithms discussed here. A prominent example is Xanadu's cloud-accessed Borealis quantum chip that includes squeezers, beam splitters, and interferometers. While the parameters of the interferometers can be modified, the parameters of the squeezers are fixed. This prevents implementation of our CV QNN which requires squeezing parameters that can be updated. These limitations on current photonic quantum hardware highlight the importance of codesign of quantum hardware and software. Codesign is critical in overcoming hardware limitations and addressing efficient use of limited resources. We are exploring the possibility of collaborating with a national laboratory or university partner with photonic quantum computing capabilities in order to experimentally test our model.

We created a CV quantum circuit that can prepare a single-photon state with 99.9% fidelity, a cat state with 99.8% fidelity, and a GKP state of fidelity 93.9%. In Ref. [28], they performed the state preparation using CV QNNs and use the Kerr gate for nonlinearity, making the entire process experimentally hard to achieve. However, our CV QNN can prepare the states in a way that can be realized experimentally. Although we could not achieve high fidelity for the GKP state because of the high computational requirements, we would need a higher cutoff dimension with more layers and optimization steps, leading to numerous training parameters for preparing such a complex state. However, it can be done, in principle, using high computational resources. GKP states could be a key factor in creating a scalable photonic fault-tolerant quantum computer [29]. We delved into the impact of noise, particularly loss, on quantum state preparation and discerned that different states respond variably to noise, with an overall fidelity reduction in all cases. Single-photon and cat state preparations performed well under realistic loss, whereas GKP states exhibited low performance under the influence of loss. Looking ahead, it is essential to investigate strategies to improve the fidelity of GKP states in the presence of loss. This could involve exploring novel quantum error correction techniques to mitigate the impact of loss and extending the study to other types of noise and losses, thus broadening our understanding of quantum state preparation in noisy environments. For example, it is worth exploring using the continuous-variable quantum erasure-correcting code introduced in Ref. [30], which is based on experimentally realizable linear optical elements.

We developed models (from CV QNNs) capable of fitting functions using noisy data sets. We thoroughly analyzed the layers and noise effects on the curve fitting. As we increased the noise five times, the accuracy only went down by 24%. Our model could still learn and accurately reproduce the curve's shape. This insight is valuable as our model can perform well even with noisy data sets. We encountered some challenges while attempting to approximate complex functions using our hybrid model for curve fitting. We discovered that these functions required a higher cutoff dimension and more layers to achieve a lower cost. This, in turn, necessitated additional computational resources for fitting more complex functions. Despite the issues, we successfully performed curve fitting on various functions such as sine decay. It would be interesting to further study how quickly the QNN

learns compared to its classical counterpart based on the complexity of the function and available data points for training. For the sine function we studied, the quantum and classical circuits reach the same accuracy after training, although there has been some work showing that with few training data, the quantum circuits could learn faster [31]. They do not provide an advantage over classical machine learning, but it has been argued that quantum circuits can outperform their classical counterparts under certain assumptions. The importance of data in quantum machine learning has also been studied; see, e.g., Ref. [32], where it was discussed how one could achieve quantum advantage based on chosen data and other machine-learning techniques.

The binary classification problem achieved high classification accuracy, with an AUC score of 0.9 and accuracy of more than 95% on a highly unbalanced data set. Similar work has been done to study fraud detection in credit card transactions using the quantum support vector machine [33]. They used a quantum-classical method to select the best features for the training process. They also focused on the importance of using quantum machine learning in selecting these features to improve the model's accuracy, which complements the classical approach in finance.

Image recognition done on the MNIST classification model can classify handwritten digits with up to 97% accuracy. We did not observe a significant change in the results as we varied the mix of classical and quantum layers. Hence, future research could investigate the efficacy of quantum layers in hybrid neural networks and quantum neural networks in general. Quantum computation has the potential to provide exponential speedup over classical computation for certain problems, and quantum layers can exploit this speedup to perform computations more efficiently than classical layers for tasks that can benefit from quantum algorithms. Quantum layers can also leverage the properties of superposition and entanglement to process and represent information in ways that are not possible with classical layers. Quantum layers can also be utilized as nonlinear feature mappings that are challenging for classical layers. Quantum feature maps can be used to transform input data into higher dimensions for richer representation to later be utilized by classical neural networks, leading to potentially more accurate classification. Moreover, even though CV QGANs (quantum generative adversarial networks) have been previously studied (see, e.g., Ref. [10]), it would be interesting to explore the performance of CV QGANs utilizing our proposed experimentally feasible setup. The proposed prototype of QGANs in Ref. [10] requires the use of non-Gaussian gates within its quantum layers for both the quantum generator and the quantum discriminator. Since our prescribed quantum layer only requires Gaussian gates, we can simulate the effectiveness of an experimentally viable QGAN.

An important issue to address is benchmarking the performance of QNNs and identifying advantages over their classical counterparts. This has been addressed in the case of discrete-variable (DV) quantum computing. The authors of Ref. [34] compared the expressibility of classical and quantum neural networks by calculating effective dimensions for different cases. They showed that QNNs have higher effective dimensions and train faster than their classical counterparts. They used the Fisher information spectrum to demonstrate the resilience of QNNs in terms of barren plateaus and the problem of vanishing gradients. It is important to perform a similar study with CV QNNs and benchmark the various models considered here in terms of an appropriate measure, such as effective dimension. A related study has been performed on barren plateaus in bosonic variational circuits [35]. They used an energy-dependent circuit to prepare Gaussian and number states. It would be interesting to extend this to other problems in quantum machine learning by calculating effective dimensions for different models and comparing them with classical as well as DV quantum models. We hope to report on progress on these issues in the near future.

While we have concentrated on quantum optical systems, there are other promising candidates for CV QC. One such platform is based on microwave quantum systems, also known as circuit QED (cQED), in which nonlinearities are introduced through Josephson junctions [36]. They have been shown to lead to superpositions of Fock states [37], cat states [38,39], and GKP states [40]. Through a superconducting nonlinear asymmetric inductive element (SNAIL), a universal set of gates can be implemented for CV QC [41]. Experimental challenges remain, as high resonator quality factors [ $\mathcal{O}(10^5)$ ] are needed [42], as well as  $\mathcal{O}(\text{ns})$  pulse synthesis resolution [43]. Nevertheless, it is an interesting avenue to explore for the CV QNNs discussed here.

In conclusion, with our proposed CV quantum algorithm, we have obtained promising results in solving a wide range of machine-learning problems. The nonlinear quantum circuit element we introduced, which was based on an earlier proposal for universal CV quantum computing [13], offers an experimentally feasible solution to introducing nonlinearity using current photonic quantum hardware, avoiding the high complexity in experimentally realizing non-Gaussian operators.

## ACKNOWLEDGMENTS

This research was funded by the National Science Foundation under Award No. DGE-2152168. A portion of the computation for this work was performed on the University of Tennessee Infrastructure for Scientific Applications and Advanced Computing (ISAAC) computational resources. K.Y.A. was supported by MITRE's Quantum Horizon Program.

- [1] S. L. Braunstein and P. van Loock, *Rev. Mod. Phys.* **77**, 513 (2005).
- [2] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook, *Phys. Rev. A* **92**, 063825 (2015).
- [3] K. Yeter-Aydeniz and G. Siopsis, *Phys. Rev. D* **97**, 036004 (2018).

- [4] K. Yeter-Aydeniz, E. Moschandreou, and G. Siopsis, *Phys. Rev. A* **105**, 012412 (2022).
- [5] H.-K. Lau, R. Pooser, G. Siopsis, and C. Weedbrook, *Phys. Rev. Lett.* **118**, 080501 (2017).
- [6] M. Lassen, L. S. Madsen, M. Sabuncu, R. Filip, and U. L. Andersen, *Phys. Rev. A* **82**, 021801(R) (2010).



- [7] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, *Phys. Rev. Res.* **1**, 033063 (2019).
- [8] J. Shi, S. Chen, Y. Lu, Y. Feng, R. Shi, Y. Yang, and J. Li, *Sci. Rep.* **10**, 2107 (2020).
- [9] P.-H. Qiu, X.-G. Chen, and Y.-W. Shi, *IEEE Access* **7**, 94310 (2019).
- [10] S. Y. Chang, S. Vallecorsa, E. F. Combarro, and F. Carminati, *arXiv:2101.11132*.
- [11] S. Monaco, O. Kiss, A. Mandarino, S. Vallecorsa, and M. Grossi, *Phys. Rev. B* **107**, L081105 (2023).
- [12] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, *Phys. Rev. X* **11**, 041011 (2021).
- [13] K. Marshall, R. Pooser, G. Siopsis, and C. Weedbrook, *Phys. Rev. A* **91**, 032321 (2015).
- [14] S. Scheel, K. Nemoto, W. J. Munro, and P. L. Knight, *Phys. Rev. A* **68**, 032310 (2003).
- [15] S. Konno, A. Sakaguchi, W. Asavanant, H. Ogawa, M. Kobayashi, P. Marek, R. Filip, J.-I. Yoshikawa, and A. Furusawa, *Phys. Rev. Appl.* **15**, 024024 (2021).
- [16] D. Gottesman, A. Kitaev, and J. Preskill, *Phys. Rev. A* **64**, 012310 (2001).
- [17] P. Kok, H. Lee, and J. P. Dowling, *Phys. Rev. A* **65**, 052104 (2002).
- [18] D. Su, C. R. Myers, and K. K. Sabapathy, *Phys. Rev. A* **100**, 052301 (2019).
- [19] M. Eaton, C. González-Arciniegas, R. N. Alexander, N. C. Menicucci, and O. Pfister, *Quantum* **6**, 769 (2022).
- [20] T. J. Stavenger, E. Crane, K. C. Smith, C. T. Kang, S. M. Girvin, and N. Wiebe, in *Proceedings of the 2022 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, New York, 2022), pp. 1–8.
- [21] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, *Quantum* **3**, 129 (2019).
- [22] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, *arXiv:1603.04467*.
- [23] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey *et al.*, *Nat. Methods* **17**, 261 (2020).
- [24] F. Gao and L. Han, *Comput. Optim. Appl.* **51**, 259 (2012).
- [25] Y. Shi, C. Chamberland, and A. Cross, *New J. Phys.* **21**, 093007 (2019).
- [26] E. A. Bachmann, Janio Martinez, Kaggle (2018), <https://www.kaggle.com/mlg-ulb/creditcardfraud>.
- [27] L. Deng, *IEEE Signal Process. Mag.* **29**, 141 (2012).
- [28] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, *Quantum Sci. Technol.* **4**, 024004 (2019).
- [29] J. E. Bourassa, R. N. Alexander, M. Vasmer, A. Patil, I. Tzitrin, T. Matsuura, D. Su, B. Q. Baragiola, S. Guha, G. Dauphinais, K. K. Sabapathy, N. C. Menicucci, and I. Dhand, *Quantum* **5**, 392 (2021).
- [30] M. Lassen, M. Sabuncu, A. Huck, J. Niset, G. Leuchs, N. J. Cerf, and U. L. Andersen, *Nat. Photonics* **4**, 700 (2010).
- [31] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, *Nat. Commun.* **13**, 4919 (2022).
- [32] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, *Nat. Commun.* **12**, 2631 (2021).
- [33] M. Grossi, N. Ibrahim, V. Radescu, R. Loredó, K. Voigt, C. Von Altrock, and A. Rudnik, *IEEE Trans. Quantum Eng.* **3**, 1 (2022).
- [34] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, *Nat. Comput. Sci.* **1**, 403 (2021).
- [35] B. Zhang and Q. Zhuang, *arXiv:2305.01799*.
- [36] X. Gu, A. F. Kockum, A. Miranowicz, Y. X. Liu, and F. Nori, *Phys. Rep.* **718-719**, 1 (2017).
- [37] M. Hofheinz, H. Wang, M. Ansmann, R. C. Bialczak, E. Lucero, M. Neeley, A. D. O'Connell, D. Sank, J. Wenner, J. M. Martinis, and A. N. Cleland, *Nature (London)* **459**, 546 (2009).
- [38] B. Vlastakis, G. Kirchmair, Z. Leghtas, S. E. Nigg, L. Frunzio, S. M. Girvin, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, *Science* **342**, 607 (2013).
- [39] R. Lescanne, M. Villiers, T. Peronnin, A. Sarlette, M. Delbecq, B. Huard, T. Kontos, M. Mirrahimi, and Z. Leghtas, *Nat. Phys.* **16**, 509 (2020).
- [40] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf, L. Frunzio, M. Mirrahimi, and M. H. Devoret, *Nature (London)* **584**, 368 (2020).
- [41] T. Hillmann, F. Quijandría, G. Johansson, A. Ferraro, S. Gasparinetti, and G. Ferrini, *Phys. Rev. Lett.* **125**, 160501 (2020).
- [42] A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, and M. H. Devoret, *Nature (London)* **584**, 205 (2020).
- [43] M. Sandberg, C. M. Wilson, F. Persson, T. Bauch, G. Johansson, V. Shumeiko, T. Duty, and P. Delsing, *Appl. Phys. Lett.* **92**, 203501 (2008).