

## ARTICLE OPEN



# Experimental quantum end-to-end learning on a superconducting processor

Xiaoxuan Pan<sup>1,6</sup>, Xi Cao<sup>2,6</sup>, Weiting Wang<sup>1</sup>, Ziyue Hua<sup>1</sup>, Weizhou Cai<sup>1</sup>, Xuegang Li<sup>1</sup>, Haiyan Wang<sup>1</sup>, Jiaqi Hu<sup>2</sup>, Yipu Song<sup>1</sup>, Dong-Ling Deng<sup>1,3,4</sup>, Chang-Ling Zou<sup>4,5</sup>, Re-Bing Wu<sup>2,4</sup> and Luyan Sun<sup>1,4</sup>

Machine learning can be enhanced by a quantum computer via its inherent quantum parallelism. In the pursuit of quantum advantages for machine learning with noisy intermediate-scale quantum devices, it was proposed that the learning model can be designed in an end-to-end fashion, i.e., the quantum ansatz is parameterized by directly manipulable control pulses without circuit design and compilation. Such gate-free models are hardware friendly and can fully exploit limited quantum resources. Here, we report the experimental realization of quantum end-to-end machine learning on a superconducting processor. The trained model can achieve 98% recognition accuracy for two handwritten digits (via two qubits) and 89% for four digits (via three qubits) in the MNIST (Mixed National Institute of Standards and Technology) database. The experimental results exhibit the great potential of quantum end-to-end learning for resolving complex real-world tasks when more qubits are available.

*npj Quantum Information* (2023)9:18; <https://doi.org/10.1038/s41534-023-00685-w>

## INTRODUCTION

Quantum computing<sup>1</sup> holds the promise of revolutionizing the field of machine learning (ML)<sup>2–4</sup>. Powered by quantum Fourier transform and amplitude amplification, potential exponential speed-up has been envisaged for high-dimensional and big-data ML tasks<sup>5–9</sup> using fault-tolerant quantum computers. Even with noisy intermediate-scale quantum (NISQ) devices, quantum advantage is still promising, because the model expressibility can be substantially enhanced by the exponentially large feature space carried by multi-qubit quantum states<sup>10–13</sup>.

To deploy quantum machine learning algorithms on NISQ processors, a key step is to construct a parameterized quantum ansatz that can be trained by a classical optimizer. To date, most quantum ansatzes are realized by quantum neural networks (QNN)<sup>10,14–23</sup> that consist of layers of parameterized quantum gates, and demonstrative experiments have been carried out on classification<sup>24–26</sup>, clustering<sup>27,28</sup>, and generative<sup>29–31</sup> learning tasks. The gate-based QNN ansatz naturally incorporates the theory of quantum circuits, but the learning performance is highly dependent on the architecture design and the mapping of circuits to experimentally operable native gates. A structurally non-optimized QNN cannot fully exploit the limited quantum coherence resource, and this is partially why high learning accuracy is hard to attain on NISQ devices without downsizing the training dataset.

There are certainly much room for performance improvement by using more hardware-efficient quantum ansatzes, e.g., via deep optimization of the circuit architecture<sup>32</sup> and qubit mapping strategies<sup>33</sup>. Recently, a hardware-friendly end-to-end learning scheme (in the sense that the model is trained as a whole instead of being divided into separate modules) is proposed<sup>34</sup> by replacing the gate-based QNN with natural quantum dynamics driven by coherent control pulses. This model requires very little architecture design, system calibration, and no qubit mapping.

One can also jointly train a data encoder that automatically transforms classical data to quantum states via control pulses, and this essentially simplifies the encoding process because the preparation of quantum states according to a hand-designed encoding scheme is no more required. More importantly, the natural control-to-state mapping involved in the encoding process introduces nonlinearity that is crucial for better model expressibility. The idea of applying pulse-based ansatzes has drawn much attention in NISQ applications, e.g., the state preparation of quantum eigensolver<sup>35</sup>, optimization landscape investigation<sup>36</sup>, and cloud-based training<sup>37</sup>.

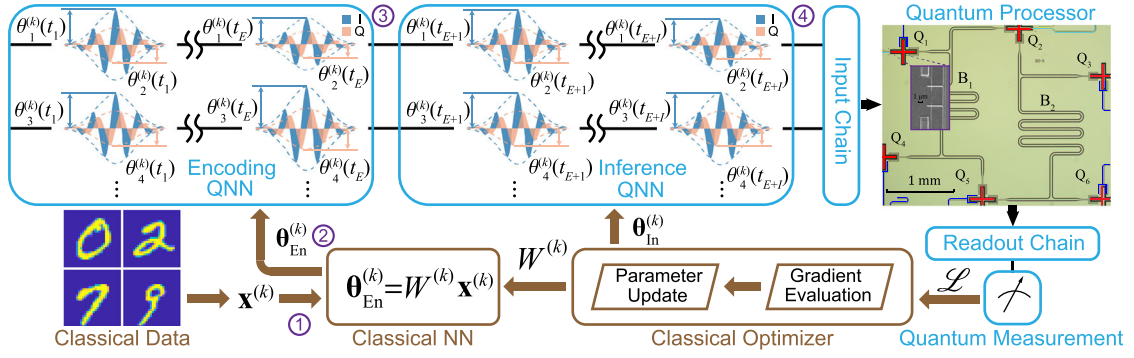
In this paper, we report the experimental demonstration of quantum end-to-end machine learning using a superconducting processor through the recognition of handwritten digits selected from the MNIST (Mixed National Institute of Standards and Technology) dataset<sup>38</sup>. Without downsizing the original 784-pixel images, the end-to-end learning model can be trained to achieve 98% accuracy with two qubits for the 2-digit classification and 89% accuracy with three qubits for the 4-digit task, which are among the best experimental results reported on small-size quantum processors<sup>39</sup>. The demonstrated quantum end-to-end model can be easily scaled up for solving complex real-world learning tasks owing to its inherent hardware friendliness and efficiency.

## RESULTS

### Preliminaries

The basic idea of end-to-end quantum learning is to parameterize the quantum ansatz by physical control pulses that are usually applied to implement abstract quantum gates in variational quantum classifiers. In this way, a feedforward QNN can be constructed by the control-driven evolution of the quantum state

<sup>1</sup>Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China. <sup>2</sup>Center for Intelligent and Networked Systems, Department of Automation, Tsinghua University, Beijing 100084, China. <sup>3</sup>Shanghai Qi Zhi Institute, No. 701 Yunjin Road, Xuhui District, Shanghai 200232, China. <sup>4</sup>Heifei National Laboratory, Hefei 230088, China. <sup>5</sup>Key Laboratory of Quantum Information, CAS, University of Science and Technology of China, Hefei, Anhui 230026, China. <sup>6</sup>These authors contributed equally: Xiaoxuan Pan, Xi Cao. ✉email: rbwu@tsinghua.edu.cn; luyansun@tsinghua.edu.cn



**Fig. 1 The training protocol of the quantum end-to-end learning framework.** In the  $k$ -th iteration, a randomly selected image of a handwritten digit in the MNIST dataset is converted to a vector  $\mathbf{x}^{(k)}$  and then transformed by a matrix  $W^{(k)}$  to the control variables  $\boldsymbol{\theta}_{\text{En}}^{(k)}$  that steer the quantum state to  $|\psi^{(k)}(t_E)\rangle$  of the qubits in the QNN. This process encodes  $\mathbf{x}^{(k)}$  to  $|\psi^{(k)}\rangle$ . Subsequent inference control pulses  $\boldsymbol{\theta}_{\text{In}}^{(k)}$  are applied to drive  $|\psi^{(k)}(t_E)\rangle$  to  $|\psi^{(k)}(t_{E+1})\rangle$  that is to be measured. The parameters in  $W^{(k)}$  and  $\boldsymbol{\theta}_{\text{In}}^{(k)}$  are updated for the next iteration according to the loss function  $\mathcal{L}$  and its gradient obtained from the measurement. The circled numbers represent the specific points in the data flow and the corresponding learning performances are shown in Fig. 4. The top right is the false-colored optical image of the six-qubit device used in our experiment.

$|\psi(t)\rangle$ , as follows<sup>40</sup>:

$$\frac{d|\psi(t)\rangle}{dt} = -\frac{i}{\hbar} \left[ H_0 + \sum_{m=1}^r \theta_m(t) H_m \right] |\psi(t)\rangle, \quad (1)$$

where  $H_0$  is the static Hamiltonian which involves the coupling between different qubits, and  $r$  is the number of pulsed control functions/channels in the quantum processor. For example, if there are  $M$  qubits for the QNN and each qubit is dictated by  $c$  control functions (e.g., flux bias or microwave driving), we have  $r = c \times M$ . Here,  $H_m$  is the control Hamiltonian associated with the  $m$ -th control pulse that contains  $n$  sub-pulses over  $n$  sampling periods. The  $j$ -th sub-pulse is parameterized by  $\theta_m(t_j)$ , and hence we denote the  $m$ -th control pulse by  $\boldsymbol{\theta}_m = [\theta_m(t_1), \theta_m(t_2), \dots, \theta_m(t_n)]$ . The evolution of the quantum system under all  $n$ -th control sub-pulses constructs the  $n$ -th layer of the QNN.

We illustrate the quantum end-to-end learning with a classification task based on the MNIST dataset. As shown in Fig. 1, an image of a handwritten digit is randomly selected from the training dataset  $\mathcal{D}$ . In the  $k$ -th iteration, the sampled image is converted to a  $d = 784$  dimensional vector  $\mathbf{x}^{(k)}$ , and  $y^{(k)}$  is the corresponding label. The input data  $\mathbf{x}^{(k)}$  is transformed by a matrix  $W^{(k)}$  to the control variables  $\boldsymbol{\theta}_{\text{En}}^{(k)} = W^{(k)} \mathbf{x}^{(k)}$ . This constructs a classical encoding block with  $r$  channels and  $E$  sub-pulses per channel:  $\boldsymbol{\theta}_{\text{En}}^{(k)} = [\theta_1^{(k)}(t_1), \dots, \theta_1^{(k)}(t_E), \dots, \theta_r^{(k)}(t_1), \dots, \theta_r^{(k)}(t_E)]$ . The generated control pulses  $\boldsymbol{\theta}_{\text{En}}^{(k)}$  then automatically encode  $\mathbf{x}^{(k)}$  to the quantum state  $|\psi^{(k)}(t_E)\rangle$  via the natural quantum state evolution of Eq. (1).

Subsequent inference control pulses  $\boldsymbol{\theta}_{\text{In}}^{(k)}$ , which have the same form as  $\boldsymbol{\theta}_{\text{En}}^{(k)}$  but consist of  $l$  sub-pulses in each channel, are then applied to induce the quantum evolution from the encoded quantum state  $|\psi^{(k)}(t_E)\rangle$ . The inference controls are introduced for improving the classification performance. Finally, the end-time quantum state  $|\psi^{(k)}(t_{E+1})\rangle$  is measured under the appropriate experiment-available positive operator  $O^{(k)}$  according to the classical label  $y^{(k)}$ , which gives the conditional probability (or confidence) of obtaining  $y^{(k)}$  for a given input  $\mathbf{x}^{(k)}$

$$P(y^{(k)} | \mathbf{x}^{(k)}, W^{(k)}, \boldsymbol{\theta}_{\text{In}}^{(k)}) = \langle \psi^{(k)}(t_{E+1}) | O^{(k)} | \psi^{(k)}(t_{E+1}) \rangle. \quad (2)$$

The corresponding loss function is defined as

$$\mathcal{L} \left[ W^{(k)}, \boldsymbol{\theta}_{\text{In}}^{(k)}, \left\{ \mathbf{x}_\ell^{(k)} \right\} \right] = 1 - \frac{1}{b} \sum_{\ell=1}^b P(y^{(k)} | \mathbf{x}_\ell^{(k)}, W^{(k)}, \boldsymbol{\theta}_{\text{In}}^{(k)}). \quad (3)$$

In the experiment, we select a batch of  $b$  samples in each iteration to reduce the fluctuation of  $\mathcal{L}$  for faster convergence of the learning process. The gradient of the loss function  $\mathcal{L}$  with respect to the encoding control  $\boldsymbol{\theta}_{\text{En}}^{(k)}$  and the inference control  $\boldsymbol{\theta}_{\text{In}}^{(k)}$  can be evaluated with the finite difference method by making a small change of the each control parameter  $\theta_i^{(k)}(t_j)$ <sup>41</sup>. The gradient of  $\mathcal{L}$  with respect to  $W^{(k)}$  can be derived from the gradient of  $\mathcal{L}$  with respect to  $\boldsymbol{\theta}_{\text{En}}^{(k)}$ <sup>34</sup>. Therefore, we can apply the widely used stochastic gradient-descent algorithm in machine learning to update  $W^{(k)}$  and  $\boldsymbol{\theta}_{\text{In}}^{(k)}$  by minimizing  $\mathcal{L}$  on the training dataset  $\mathcal{D}$  (see Methods for details of the algorithms)<sup>42</sup>. Once the model is well trained, one can use fresh samples from a testing dataset to examine the recognition performance of the handwritten digits.

## Experiments and simulations

The end-to-end model is demonstrated in a superconducting processor, as shown in Fig. 1. All qubits take the form of the flux-tunable Xmon geometry and are driven with inductively coupled flux bias lines and capacitively coupled RF control lines<sup>43–45</sup>. Among the six qubits,  $Q_1, Q_2, Q_4, Q_5$  are dispersively coupled to a half-wavelength coplanar cavity  $B_1$ , and  $Q_2, Q_3, Q_5, Q_6$  are dispersively coupled to another cavity  $B_2$ . Each qubit is dispersively coupled to a quarter-wavelength readout resonator for a high-fidelity single-shot readout and all the resonators are coupled to a common transmission line for multiplexed readouts (see details of the experimental setup in Supplementary Notes I ~ III). The qubits that are not relevant to the QNN is biased far away and can be ignored from the system Hamiltonian, therefore, the static Hamiltonian of the QNN can be written in the interaction picture as

$$H_0/\hbar = \sum_{q \neq p} J_{qp} (a_q^\dagger a_p + a_p^\dagger a_q) - \sum_{q=1}^M \frac{E_{C,q}}{2} a_q^\dagger a_q^\dagger a_q a_q, \quad (4)$$

where  $J_{qp}$  is the coupling strength between the  $p$ -th and  $q$ -th qubits mediated by the bus cavity,  $E_{C,q}$  denotes the qubit anharmonicity, and  $a_q$  is the annihilation operator of the  $q$ -th qubit.

Throughout this work, we set the encoding block with  $E = 2$  layers followed by an inference block with  $l = 2$  layers. As shown in Fig. 1, for the  $q$ -th qubit in the  $n$ -th ( $n = 1, 2, 3, 4$ ) layer of the QNN, there are  $c = 2$  control parameters  $\theta_{2q-1}(t_n)$  and  $\theta_{2q}(t_n)$ , which are associated with the control Hamiltonians  $H_{2q-1} = (a_q + a_q^\dagger)/2$  (rotation along the  $x$ -axis of the Bloch sphere) and

$H_{2q} = i(a_q - a_q^\dagger)/2$  (rotation along the  $y$ -axis of the Bloch sphere), respectively. The control parameters are the variable amplitudes of the Gaussian envelopes of two resonant microwave sub-pulses, each of which has a fixed width of  $4\sigma = 40$  ns. All the quantum controls in the same time interval are exerted simultaneously. For an  $N$ -digit classification task, we take  $M = \lceil \log_2 N \rceil + 1$  qubits for the QNN: the classification results are mapped to the computation bases of the first  $\lceil \log_2 N \rceil$  qubits (label qubits) by the majority vote of the collective measurement performed on label qubits, while one additional qubit is introduced for a better expressibility of the model. Therefore, the QNN in our experiment involves totally  $cM(E + I) = 8M$  control parameters.

We perform the 2-digit ('0' and '2') classification task ( $N = 2$ ) with  $Q_3$  and  $Q_5$  ( $M = 2$ ). The working frequencies are 6.08 GHz and 6.45 GHz, respectively, which are also the flux sweet-spots of the two qubits. The effective coupling strength  $J_{35}/2\pi = 4.11$  MHz. We take  $Q_5$  as the label qubit and assign the classification result to be '0' or '2' if the respective probability of measuring  $|g\rangle$  or  $|e\rangle$  state is larger.

The end-to-end model is initialized with  $W = W_0$  and  $\theta_{\text{in}} = \theta_0$ , where all elements of  $W_0$  are  $10^{-5}$  and each element of  $\theta_0$  is tuned to induce a  $\pi/4$  rotation of the respective qubit. The parameter update is realized as follows. Firstly, we obtain the loss function  $\mathcal{L}$  according to Eq. (3) by measuring  $Q_5$ . We perturb each control parameter in the control set  $\{\theta_{\text{Env}}, \theta_{\text{in}}\}$  and obtain the corresponding gradient of  $\mathcal{L}$ . The  $\mathcal{L}$  and its gradient averaged over a batch of two training samples ( $b = 2$ ) are sent to a classical Adam optimizer<sup>46</sup> for updating  $W$  and  $\theta_{\text{in}}$ . All control parameters are linearly scaled to the digital-to-analog converter level of a Tektronix arbitrary waveform generator 70002A, working with a sampling rate of 25 GHz, to generate the resonant RF pulses directly. The control pulses composed of in-phase and quadrature components are sent to each qubit with the corresponding RF control line. To obtain the classification result, we repeat the procedure and measure the label qubit for 5000 times.

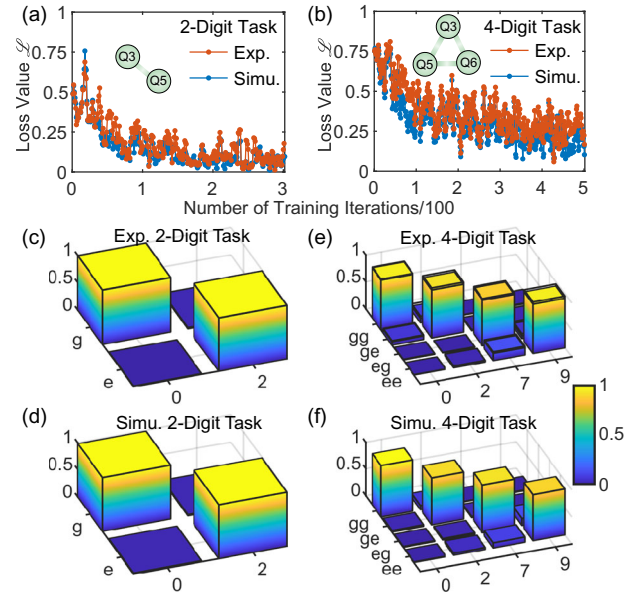
In the 4-digit ('0', '2', '7', and '9') classification task ( $N = 4$ ), we take  $Q_3$ ,  $Q_5$ , and  $Q_6$  ( $M = 3$ ) to construct the QNN, whose working frequencies are 6.08 GHz, 6.45 GHz, and 6.19 GHz, respectively.  $Q_3$  and  $Q_5$  are measured for the classification output. The target digits correspond to the four computational bases spanned by the two label qubits. The training procedure and algorithms are the same as those for the  $N = 2$  task.

The typical training process is shown in Fig. 2a, b. For better clarity, the curves are smoothed out by averaging each data point from its neighboring four ones. For the 2-digit (4-digit) classification task, the experimental loss function  $\mathcal{L}$  converges to 0.14 (0.22) in 300 (500) iterations. The training loss can potentially be reduced by increasing the depth  $E$  of the encoding block<sup>34</sup>. For comparison, numerical simulations are also performed with the calibrated system Hamiltonian, the same batches of training samples, and the same parameter update algorithms. As shown in Fig. 2a, b, the simulations match the experiments well. The small deviation of the experimental data may attribute to the simplified modeling of high-order couplings between the qubits and the control pulses<sup>47</sup>, as well as the system parameter drifting.

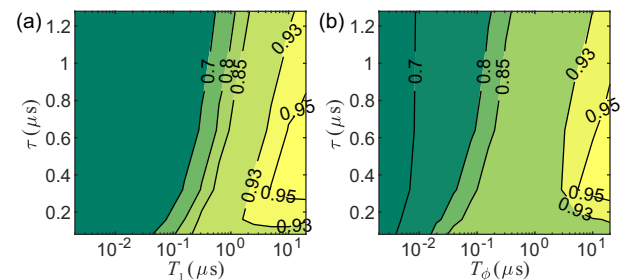
To examine the performance of the end-to-end learning, we experimentally test the generalizability of the trained end-to-end model with fresh testing samples (1000 for each digit), and count the frequencies of assigning these samples to different digits (see Fig. 2c–f). The measured overall accuracies (i.e., the proportion of samples that are correctly classified) are  $(98.6 \pm 0.1)\%$  for the 2-digit task and  $(89.4 \pm 1.5)\%$  for the 4-digit task, respectively, which are consistent with the simulation results (98.2% and 88.9%, respectively) based on the experimentally identified Hamiltonian.

The performance of the model also relies on the amount of entanglement gained in the quantum state. When the number of QNN layers is fixed, the quantum state gets more entangled under longer pulse length  $\tau$  (includes all the  $E + I = 4$  sub-pulses in both the encoding and the inference blocks), but coherence may be

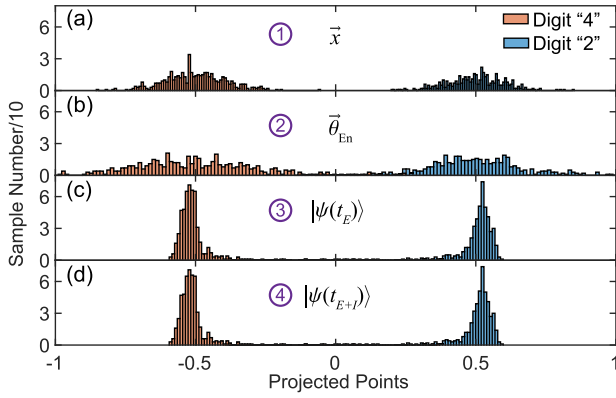
lost in the prolonged control time duration due to the inevitable decoherence. We use the experimentally calibrated parameters to simulate the 2-digit classification process under different  $\tau$  and different coherence times  $T_1$  and  $T_\phi$  of the qubits (see simulation details in Supplementary Notes: The Numerical Model of the System). As shown in Fig. 3, the average confidence  $1 - \mathcal{L}$  varies little with  $\tau$  when  $T_1$  or  $T_\phi$  is sufficiently small because the coherent control is overwhelmed by the strong decoherence. For larger  $T_1$  or  $T_\phi$  (e.g.,  $T_1 = 20 \mu\text{s}$ ), the average confidence initially increases with  $\tau$ , but then decreases after reaching the peak. This trend clearly indicates the trade-off between the gained entanglement and the lost coherence, and thus  $\tau$  as well as the number of layers should be optimally chosen for the best balance.



**Fig. 2 Results of training the quantum end-to-end model.** **a, b** The typical training process of the end-to-end model. For better clarity, all data points are averaged over the neighboring four points. **c–f** The classification performance of the trained model. The horizontal labels show the digits to be classified, while the vertical labels show the majority vote of the computational basis measurement results. The hollow bars (nearly invisible) in the experimental results (**c, e**) correspond to the standard deviation of multiple repeated measurements. **c, d** Experimental and simulation results for the 2-digit classification task, respectively. The averaged accuracies for the classification are  $0.986 \pm 0.001$  and  $0.982$  in the experiment and the simulation, respectively. **e, f** The 4-digit classification of the QNN. The averaged accuracies are  $0.894 \pm 0.015$  and  $0.889$  in the experiment and the simulation, respectively.



**Fig. 3 The error analysis of the average confidence.** **a** The simulated average confidence as a function of the qubit relaxation time  $T_1$  and the pulse length  $\tau$ , while the qubit pure dephasing time  $T_\phi$  is infinite; **b** The simulated average confidence as a function of  $T_\phi$  and  $\tau$ , while  $T_1$  is infinite.



**Fig. 4** The projected points by using LDA, where the distance between the projected cluster centers is normalized. The positions of ①~④ along the data flow are shown in Fig. 1. **a** The projected points of the original 784-dimensional handwritten digits. **b** The projected points of the control pulses for encoding the classical data to the quantum states. **c** The projected points of the quantum states after the encoding block. **d** The projected points of the final quantum states.

The end-to-end learning scheme provides a seamless combination of quantum and classical computers through the joint training of the pulse-based QNN and the classical data encoder  $W$ . To understand their respective roles in the classification, we check how the data distribution varies along the flow  $\mathbf{x} \rightarrow \boldsymbol{\theta}_{\text{En}} \rightarrow |\psi(t_E)\rangle \rightarrow |\psi(t_{E+I})\rangle \rightarrow y$  (see ①~④ in Fig. 1) in the 2-digit classification process. To facilitate the analysis, we use the Linear Discriminant Analysis (LDA)<sup>48</sup> that projects high-dimensional data vectors into two clusters of points distributed on an optimally chosen line (see details in the Supplementary Notes: Linear Discriminant Analysis). The LDA makes it easier to visualize and compare data distributions whose dimensionalities are different.

The projected clusters are plotted in Fig. 4. In each sub-figure, the distance between the centers of the two clusters is normalized, and hence we can quantify the classifiability by their standard deviations (i.e., the narrowness of distribution). As can be seen in Fig. 4a, b, the classical data encoder  $W$  effectively reduces the original 784-dimensional vector  $\mathbf{x}$  to a 8-dimensional vector of control variables  $\boldsymbol{\theta}_{\text{En}}$ , but the standard deviation is increased from 0.1658 for the original dataset to 0.2903 for the transformed control pulses. Then, the control-to-state mapping, which is both nonlinear and quantum, sharply reduces the standard deviation to 0.0919 for the encoded quantum state (Fig. 4c), while the following quantum inference block does not make further improvement (Fig. 4d). These results indicate that the classical data encoder is responsible for the compression of the high-dimensional input data, while the classification is mainly accomplished by the QNN.

## DISCUSSION

To conclude, our proof-of-principle experiment has clearly demonstrated the feasibility of the end-to-end quantum machine learning framework. The pulse-based QNN is experimentally easy to implement and scale up. Through joint training of the classical encoder and the QNN, high-precision classification is achieved for MNIST digits without downsizing the dataset. Our experiment indicates that the limited quantum resources on NISQ devices can be more efficiently exploited than pure gate-based quantum models.

It should be noted that no quantum advantage is claimed here over classical ML algorithms, which is still an ongoing pursuit for

all types of variational quantum algorithms. When more qubits are available and the noise level is sufficiently low, quantum advantage may be approached owing to the enhanced expressive power by the exponentially scaled quantum Hilbert space. We expect that, with more elaborately designed training algorithms, the framework of quantum end-to-end learning can be applied to more complicated real-world ML applications (e.g., unsupervised and generative learning).

## METHODS

### Algorithm for the classical optimizer

The algorithm for evaluating the loss function and the calculation of the gradient is shown in Algorithm 1. The Adam optimizer that is used in our experiment is shown in Algorithm 2.

**Algorithm 1.** Calculate the loss function and the gradients with respect to the model parameters

**Input**  $W, \boldsymbol{\theta}_{\text{In}}$ , Batch of training data  $\{\mathbf{x}^{(k)}, y^{(k)}\}$

$P_{\text{out}}, \mathbf{g}_W, \mathbf{g}_{\text{In}} \leftarrow \mathbf{0}$

$\delta \in \mathbb{R}^+$

**for**  $\ell = 1 : b$  **do**

$P_0 = P(y_\ell^{(k)} | \mathbf{x}_\ell^{(k)}, W, \boldsymbol{\theta}_{\text{In}})$

$\boldsymbol{\theta}_{\text{En}} = W * \mathbf{x}_\ell^{(k)}$

**for**  $i = 1 : \text{Length}(\boldsymbol{\theta}_{\text{En}})$  **do**

$\boldsymbol{\theta}_1 = [\boldsymbol{\theta}_{\text{En}}(1), \dots, \boldsymbol{\theta}_{\text{En}}(i) + \delta, \dots]$

$\mathbf{g}_1(i) = (P(y_\ell^{(k)} | \mathbf{x}_\ell^{(k)}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_{\text{In}}) - P_0) / \delta$

**end for**

**for**  $i = 1 : \text{Length}(\boldsymbol{\theta}_{\text{In}})$  **do**

$\boldsymbol{\theta}_2 = [\boldsymbol{\theta}_{\text{In}}(1), \dots, \boldsymbol{\theta}_{\text{In}}(i) + \delta, \dots]$

$\mathbf{g}_2(i) = (P(y_\ell^{(k)} | \mathbf{x}_\ell^{(k)}, W, \boldsymbol{\theta}_2) - P_0) / \delta$

**end for**

$\mathbf{g}_W = \mathbf{g}_W - \mathbf{g}_1 * \mathbf{x}_\ell^{(k)\top} / b$

$\mathbf{g}_{\text{In}} = \mathbf{g}_{\text{In}} - \mathbf{g}_2 / b$

$P_{\text{out}} = P_{\text{out}} + P_0 / b$

**end for**

**return**  $P_{\text{out}}, \mathbf{g}_W, \mathbf{g}_{\text{In}}$

**Algorithm 2.** Adam optimizer

**Input** Training dataset  $\{\mathbf{x}, y\}$

$\mathbf{s}_W, \mathbf{r}_W, \mathbf{s}_{\text{In}}, \mathbf{r}_{\text{In}} \leftarrow \mathbf{0}$

$W, \boldsymbol{\theta}_{\text{In}} \leftarrow W_0, \boldsymbol{\theta}_{\text{In},0}$

$\text{lr} = 10^{-3}, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$

**for**  $k = 1 : N$  **do**

$\mathbf{g}_W, \mathbf{g}_{\text{In}} = \text{Algorithm1}(W, \boldsymbol{\theta}_{\text{In}}, \{\mathbf{x}^{(k)}, y^{(k)}\})$

$\mathbf{s}_W = \beta_1 * \mathbf{s}_W + (1 - \beta_1) * \mathbf{g}_W$

$\mathbf{r}_W = \beta_2 * \mathbf{r}_W + (1 - \beta_2) * \mathbf{g}_W * \mathbf{g}_W$

$\mathbf{s}_1 = \mathbf{s}_W / (1 - \beta_1^k)$

$\mathbf{r}_1 = \mathbf{r}_W / (1 - \beta_2^k)$

$W = W - \text{lr} * \mathbf{s}_1 ./ (\sqrt{\mathbf{r}_1} + \epsilon)$

$\mathbf{s}_{\text{In}} = \beta_1 * \mathbf{s}_{\text{In}} + (1 - \beta_1) * \mathbf{g}_{\text{In}}$

$\mathbf{r}_{\text{In}} = \beta_2 * \mathbf{r}_{\text{In}} + (1 - \beta_2) * \mathbf{g}_{\text{In}} * \mathbf{g}_{\text{In}}$

$\mathbf{s}_2 = \mathbf{s}_{\text{In}} / (1 - \beta_1^k)$

$\mathbf{r}_2 = \mathbf{r}_{\text{In}} / (1 - \beta_2^k)$

$\boldsymbol{\theta}_{\text{In}} = \boldsymbol{\theta}_{\text{In}} - \text{lr} * \mathbf{s}_2 ./ (\sqrt{\mathbf{r}_2} + \epsilon)$

**end for**

**return**  $W, \boldsymbol{\theta}_{\text{In}}$

The gradient of the loss function  $\mathcal{L}$  for parameter update in each iteration is obtained by averaging the gradients of the conditional probability  $P$  over a batch of randomly selected input samples ( $b=2$ ). This can reduce the fluctuation of  $\mathcal{L}$  for faster convergence in the learning process. For the inference block, the gradient  $\mathbf{g}_{in}$  of  $\mathcal{L}$  with respect to each parameter in  $\boldsymbol{\theta}_{in}$  is directly obtained by rerunning the experiment with a small change in  $\boldsymbol{\theta}_{in}$  and calculating the difference of  $P$ . As for the encoding block, the gradient  $\mathbf{g}_W$  with respect to the elements of the classical matrix  $W$  is needed. To reduce the experimental cost, we can equivalently calculate  $\mathbf{g}_W$  from the outer product between the measured gradient  $\mathbf{g}_1$  with respect to the encoding controls and the input data vector.

Once the gradients with respect to the model parameters are obtained, we take Adaptive Moment Estimation (Adam)<sup>46</sup> update algorithm to update the corresponding parameters. The Adam algorithm is popular for its efficiency and stability in stochastic optimization of learning problems. In the Adam algorithm,  $\text{lr}, \beta_1, \beta_2, \epsilon$  refer to algorithm configuration parameters and are chosen as default empirical values. The intermediate parameters  $\mathbf{s}_W, \mathbf{s}_{in}, \mathbf{r}_W, \mathbf{r}_{in}$  will be passed to the next iteration so as to adaptively control the parameter updating rate.

## DATA AVAILABILITY

All data that is relevant with the figures of the article can be found in <https://github.com/zuozhu2424/Experimental-Quantum-End-to-End-Learning-on-a-Superconducting-Processor>.

## CODE AVAILABILITY

The code used for simulations is available from the corresponding authors upon reasonable request.

Received: 12 April 2022; Accepted: 11 February 2023;

Published online: 01 March 2023

## REFERENCES

- Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2000).
- Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
- Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep. Prog. Phys.* **81**, 074001 (2018).
- Sarma, S. D., Deng, D.-L. & Duan, L. Machine learning meets quantum physics. *Phys. Today* **72**, 48 (2019).
- Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014).
- Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **113**, 130503 (2014).
- Dunjko, V., Taylor, J. M. & Briegel, H. J. Quantum-enhanced machine learning. *Phys. Rev. Lett.* **117**, 130501 (2016).
- Gao, X., Zhang, Z.-Y. & Duan, L.-M. A quantum machine learning algorithm based on generative models. *Sci. Adv.* **4**, eaat9004 (2018).
- Liu, Y., Arunachalam, S. & Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *Nat. Phys.* **17**, 1013 (2021).
- Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
- Schuld, M. & Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
- Gao, X., Anschuetz, E. R., Wang, S.-T., Cirac, J. I. & Lukin, M. D. Enhancing generative models via quantum correlations. *Phys. Rev. X* **12**, 021037 (2022).
- Du, Y., Hsieh, M.-H., Liu, T. & Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* **2**, 033125 (2020).
- Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001 (2019).
- Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020).
- Cong, I., Choi, S. & Lukin, M. D. Quantum convolutional neural networks. *Nat. Phys.* **15**, 1273–1278 (2019).

- Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at <https://doi.org/10.48550/arXiv.1802.06002> (2018).
- Wei, S., Chen, Y., Zhou, Z. & Long, G. A quantum convolutional neural network on NISQ devices. *AAPPS Bulletin* **32**, 2 (2022).
- Houssein, E. H., Abohashima, Z., Elhoseny, M. & Mohamed, W. M. Hybrid quantum-classical convolutional neural network model for covid-19 prediction using chest x-ray images. *J. Comput. Des. Eng.* **9**, 343–363 (2022).
- Rudolph, M. S. et al. Generation of high-resolution handwritten digits with an ion-trap quantum computer. *Phys. Rev. X* **12**, 031010 (2022).
- Zeng, J., Wu, Y., Liu, J.-G., Wang, L. & Hu, J. Learning and inference on generative adversarial quantum circuits. *Phys. Rev. A* **99**, 052306 (2019).
- Jerbi, S., Gyurik, C., Marshall, S., Briegel, H. & Dunjko, V. Parametrized quantum policies for reinforcement learning. *NIPS34*. <https://proceedings.neurips.cc/paper/2021/hash/eec96a7f788e88184c0e713456026f3f-Abstract.html> (2021).
- Li, W. & Deng, D.-L. Recent advances for quantum classifiers. *Sci. China Phys. Mech. Astronomy* **65**, 1–23 (2022).
- Tacchino, F., Macchiavello, C., Gerace, D. & Bajoni, D. An artificial neuron implemented on an actual quantum processor. *npj Quantum Inf.* **5**, 26 (2019).
- Cai, X.-D. et al. Entanglement-based machine learning on a quantum computer. *Phys. Rev. Lett.* **114**, 110504 (2015).
- Johri, S. et al. Nearest centroid classification on a trapped ion quantum computer. *npj Quantum Inf.* **7**, 122 (2021).
- Ouyang, X.-L. et al. Experimental demonstration of quantum-enhanced machine learning in a nitrogen-vacancy-center system. *Phys. Rev. A* **101**, 012307 (2020).
- Li, Z., Liu, X., Xu, N. & Du, J. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett.* **114**, 140504 (2015).
- Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Inf.* **5**, 103 (2019).
- Hu, L. et al. Quantum generative adversarial learning in a superconducting quantum circuit. *Sci. Adv.* **5**, eaav2761 (2019).
- Zhu, D. et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **5**, eaaw9918 (2019).
- Ostaszewski, M., Trenkwalder, L., Masarczyk, W., Scerri, E. & Dunjko, V. Reinforcement learning for optimization of variational quantum circuit architectures. *NIPS34*. <https://proceedings.neurips.cc/paper/2021/hash/9724412729185d53a2e3e7f889d9f057-Abstract.html> (2021).
- Huang, H.-Y. et al. Power of data in quantum machine learning. *Nat. Commun.* **12**, 2631 (2021).
- Wu, R.-B., Cao, X., Xie, P. & Liu, Y.-x. End-to-end quantum machine learning implemented with controlled quantum dynamics. *Phys. Rev. Appl.* **14**, 064020 (2020).
- Meitei, O. R. et al. Gate-free state preparation for fast variational quantum eigensolver simulations. *npj Quantum Inf.* **7**, 155 (2021).
- Magann, A. B. et al. From pulses to circuits and back again: a quantum optimal control perspective on variational quantum algorithms. *PRX Quantum* **2**, 010101 (2021).
- Liang, Z. et al. PAN: Pulse Ansatz on NISQ Machines. Preprint at <http://arxiv.org/abs/2208.01215> (2022).
- LeCun, Y., Cortes, C. & Burges, C. J. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/> (2010).
- Wang, K., Xiao, L., Yi, W., Ran, S.-J. & Xue, P. Experimental realization of a quantum image classifier via tensor-network-based machine learning. *Photonics Res.* **9**, 2332–2340 (2021).
- Rivas, A. & Huelga, S. F. *Open quantum systems* Vol. 10 (Springer, 2012).
- Li, J. General explicit difference formulas for numerical differentiation. *J. Comput. Appl. Math.* **183**, 29–52 (2005).
- Friedman, J. H. Stochastic gradient boosting. *Comput. Stat. Data. An.* **38**, 367–378 (2002).
- Barends, R. et al. Coherent Josephson Qubit suitable for scalable quantum integrated circuits. *Phys. Rev. Lett.* **111**, 080502 (2013).
- Li, X. et al. Perfect quantum state transfer in a superconducting qubit chain with parametrically tunable couplings. *Phys. Rev. Appl.* **10**, 054009 (2018).
- Cai, W. et al. Observation of topological magnon insulator states in a superconducting circuit. *Phys. Rev. Lett.* **123**, 080501 (2019).
- Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).
- Motzoi, F., Gambetta, J. M., Rebentrost, P. & Wilhelm, F. K. Simple pulses for elimination of leakage in weakly nonlinear qubits. *Phys. Rev. Lett.* **103**, 110501 (2009).
- Venables, W. N. & Ripley, B. D. *Modern applied statistics with S-PLUS* (Springer Sci. & Bus. Med., 2013). <https://link.springer.com/book/10.1007/978-1-4757-3121-7>.

## ACKNOWLEDGEMENTS

This work was supported by National Key Research and Development Program of China (Grants No. 2017YFA0304303 and 2017YFA0304304), the National Natural

Science Foundation of China (Grants No. 92165209, No. 61833010, No. 62173201, No. 11925404, No. 11874235, No. 11874342, No. 11922411, No. 12061131011, No. 12075128), Key-Area Research and Development Program of Guangdong Province (Grant No. 2020B0303030001), Anhui Initiative in Quantum Information Technologies (AHY130200), China Postdoctoral Science Foundation (BX2021167), and Grant No. 2019GQG1024 from the Institute for Guo Qiang, Tsinghua University. D.-L.D. also acknowledges additional support from the Shanghai Qi Zhi Institute.

### AUTHOR CONTRIBUTIONS

X.P. and X.C. are co-first authors. X.P. performed the experiment and analyzed the data with the assistance of X.C. R.B.W. proposed the experiment. R.B.W., C.L.Z., and D.L.D. provided theoretical support. L.S. directed the project. X.P., X.C., and J.H. performed the numerical simulations. W.C. fabricated the J.P.A. W.W. and X.P. designed the devices. X.P. fabricated the devices with the assistance of W.W., H.W., and Y.P.S. Z.H., W.C., and X.L. provided further experimental support. X.P., X.C., R.B.W., and L.S. wrote the manuscript with feedback from all authors.

### COMPETING INTERESTS

The authors declare no competing interests.

### ETHICS APPROVAL

All authors have agreed to all manuscript contents, the author list and its order and the author contribution statements. Any changes to the author list after submission will be subject to approval by all authors.

### ADDITIONAL INFORMATION

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41534-023-00685-w>.

**Correspondence** and requests for materials should be addressed to Re-Bing Wu or Luyan Sun.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023