

# Deep quantum neural networks on a superconducting processor

Received: 6 December 2022

Accepted: 29 June 2023

Published online: 06 July 2023

 Check for updates

Xiaoxuan Pan <sup>1,5</sup>, Zhide Lu <sup>1,5</sup>, Weiting Wang<sup>1</sup>, Ziyue Hua<sup>1</sup>, Yifang Xu<sup>1</sup>, Weikang Li <sup>1</sup>, Weizhou Cai<sup>1</sup>, Xuegang Li<sup>1</sup>, Haiyan Wang<sup>1</sup>, Yi-Pu Song<sup>1</sup>, Chang-Ling Zou <sup>2,3</sup>, Dong-Ling Deng <sup>1,3,4</sup>  & Luyan Sun <sup>1,3</sup> 

Deep learning and quantum computing have achieved dramatic progresses in recent years. The interplay between these two fast-growing fields gives rise to a new research frontier of quantum machine learning. In this work, we report an experimental demonstration of training deep quantum neural networks via the backpropagation algorithm with a six-qubit programmable superconducting processor. We experimentally perform the forward process of the backpropagation algorithm and classically simulate the backward process. In particular, we show that three-layer deep quantum neural networks can be trained efficiently to learn two-qubit quantum channels with a mean fidelity up to 96.0% and the ground state energy of molecular hydrogen with an accuracy up to 93.3% compared to the theoretical value. In addition, six-layer deep quantum neural networks can be trained in a similar fashion to achieve a mean fidelity up to 94.8% for learning single-qubit quantum channels. Our experimental results indicate that the number of coherent qubits required to maintain does not scale with the depth of the deep quantum neural network, thus providing a valuable guide for quantum machine learning applications with both near-term and future quantum devices.

Machine learning has achieved tremendous success in both commercial applications and scientific research over the past decade. In particular, deep neural networks play a vital role in cracking some notoriously challenging problems, ranging from playing Go<sup>1</sup> to predicting protein structures<sup>2</sup>. They contain multiple hidden layers and are believed to be more powerful in extracting high-level features from data than traditional methods<sup>3,4</sup>. The learning process can be fueled by updating the parameters through gradient descent, where the backpropagation (BP) algorithm enables efficient calculations of gradients via the chain rule<sup>3</sup>.

By harnessing the weirdness of quantum mechanics such as superposition and entanglement, quantum machine learning approaches hold the potential to bring advantages compared with their classical counterpart. In recent years, exciting progress has been made

along this interdisciplinary direction<sup>5–10</sup>. For example, rigorous quantum speedups have been proved in classification models<sup>11</sup> and generative models<sup>12</sup> with complexity-theoretic guarantees. In terms of the expressive power of quantum neural networks, there is also preliminary evidence showing their advantages over the comparable feedforward neural networks<sup>13</sup>. Meanwhile, noteworthy progress has also been made on the experimental side<sup>14–22</sup>. For example, in ref. 14, the authors realize a quantum convolutional neural network on a superconducting quantum processor. In ref. 15, an experimental demonstration of quantum adversarial learning has been reported. Similar to deep classical neural networks with multiple hidden layers, a deep quantum neural network (DQNN) with the layer-by-layer architecture is proposed<sup>23–25</sup>, which can be trained via a quantum analog of the BP algorithm. The word “deep” in the DQNN refers to multiple

<sup>1</sup>Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China. <sup>2</sup>CAS Key Laboratory of Quantum Information, University of Science and Technology of China, Hefei, Anhui 230026, China. <sup>3</sup>Hefei National Laboratory, Hefei 230088, China.

<sup>4</sup>Shanghai Qi Zhi Institute, No. 701 Yunjin Road, Xuhui District, Shanghai 200232, China. <sup>5</sup>These authors contributed equally: Xiaoxuan Pan and Zhide Lu.

 e-mail: [dldeng@tsinghua.edu.cn](mailto:dldeng@tsinghua.edu.cn); [luyansun@tsinghua.edu.cn](mailto:luyansun@tsinghua.edu.cn)

hidden layers, rather than the large depth of quantum circuits. Under this framework, the quantum analog of a perceptron is a general unitary operator acting on qubits from adjacent layers, whose parameters are updated by multiplying the corresponding updating matrix of the perceptron in the training process.

In this paper, we report an experimental demonstration of training DQNNs through the BP algorithm on a programmable superconducting processor with six frequency-tunable transmon qubits. We find that a three-layer DQNN can be efficiently trained to learn a two-qubit target quantum channel with a mean fidelity of up to 96.0% and the ground state energy of molecular hydrogen with an accuracy of up to 93.3% compared to the theoretical prediction. In addition, we also demonstrate that a six-layer DQNN can efficiently learn a one-qubit target quantum channel with a mean fidelity of up to 94.8%. Our approach can carry over to other DQNNs with a larger width and depth straightforwardly, thus paving the way towards large-scale quantum machine learning with potential advantages in practical applications.

## Results

### Deep quantum neural networks

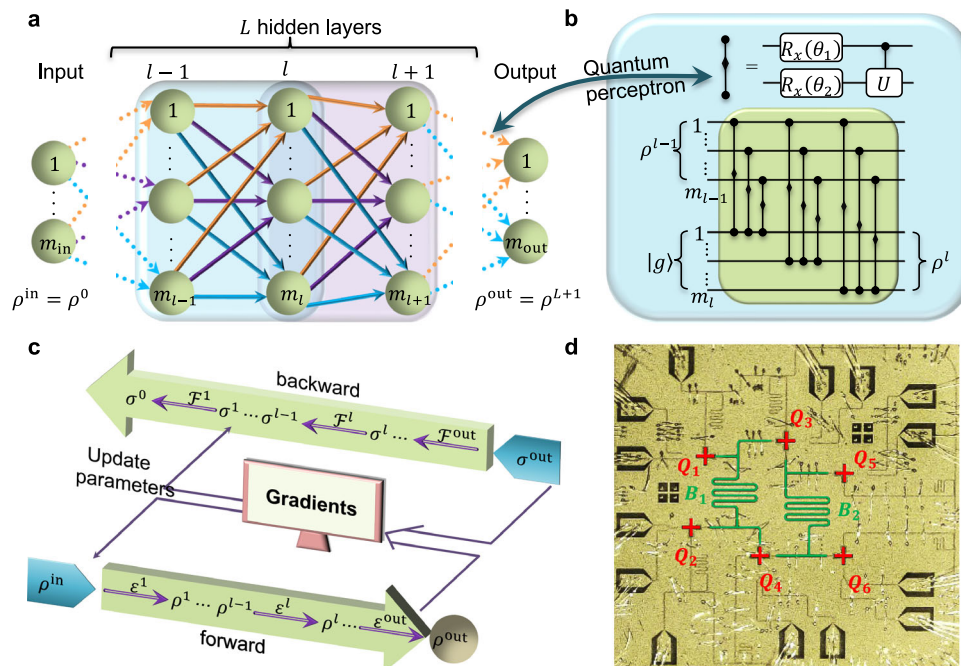
As sketched in Fig. 1a, our DQNN has a layer-by-layer structure, and maps the quantum information layerwise from the input layer state  $\rho^{\text{in}}$ , through  $L$  hidden layers, to the output layer state  $\rho^{\text{out}}$ . Quantum perceptrons are the building blocks of the DQNN, and different types of quantum perceptrons have been experimentally implemented recently<sup>26–28</sup>. DQNNs with the most general form of quantum perceptrons, which can apply generic unitaries on all qubits at adjacent layers, are capable of universal quantum computation<sup>29</sup>. In practice, we usually employ restricted forms of perceptrons for experimental implementations on noisy quantum devices. In this work, a single quantum perceptron is defined as a parameterized quantum circuit applied to the corresponding qubit pair at adjacent layers, which is

shown in Fig. 1b. A sequential combination of the quantum perceptrons constitutes the layerwise operation between adjacent layers. One of the key characteristics of the DQNN is the layer-by-layer quantum state mapping, allowing efficient training via the quantum BP algorithm<sup>23</sup>.

We sketch the general experimental training process in Fig. 1c. When performing the quantum BP algorithm, one only requires the information from adjacent two layers, rather than the full DQNN, to evaluate the gradients with respect to all parameters at these two layers. Such a BP-equipped DQNN bears the following merit: the number of coherent qubits required to maintain does not scale with the depth of the DQNN. This merit makes it possible to realize DQNNs with reduced number of layers of qubits through qubit reusing<sup>23</sup>. The qubits can be reused as follows: after completing the operations between adjacent layers, we can reset qubits in the previous layer to the fiducial product state, and then reuse them as qubits in the subsequent layer. We note that resetting qubits takes extra time in experiments, and this raises additional coherence requirements for qubits in the current layer. By reusing qubits, only two layers of qubits are required to implement a DQNN regardless of its depth. The experimental errors on qubits in the previous layer will affect qubits in the current layer, which will limit the depth of the DQNN in real experiments with noisy devices.

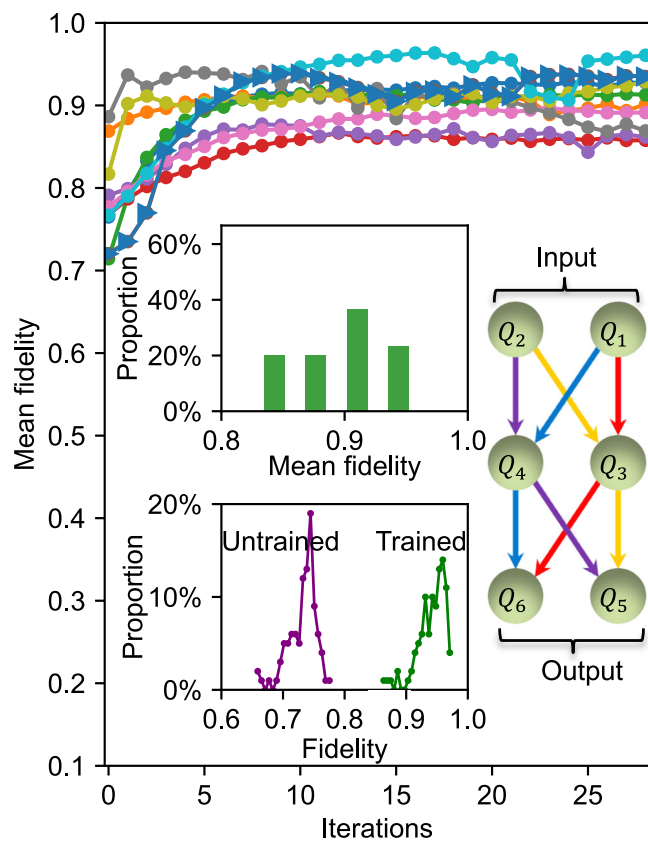
### Experimental setup

Our experiment is carried out on a superconducting quantum processor, which possesses six two-junction and frequency-tunable transmon qubits<sup>30–37</sup>. As photographed in Fig. 1d, the chip is fabricated with the layout of the qubits being purposely and carefully optimized for a layer-by-layer structure. Each transmon qubit is coupled to an individual flux control line, XY control line, and quarter-wavelength readout resonator, respectively. All readout resonators are



**Fig. 1 | A schematic of training deep quantum neural networks.** **a** Architecture exhibition of a general DQNN. Information propagates layerwise from the input layer to the output layer. At adjacent two layers, we apply the quantum perceptron in the order according to the exhibited circuit in **b**. A quantum perceptron is realized by applying two single-qubit rotation gates  $R_x(\theta_1)$  and  $R_x(\theta_2)$  (the rotations along the  $x$  axis with variational angles  $\theta_1$  and  $\theta_2$ , respectively) followed by a fixed two-qubit controlled-Phase gate. **c** Illustration of the quantum backpropagation

algorithm. We apply forward channels  $\mathcal{E}$  on  $\rho^{\text{in}}$  and successively obtain  $\{\rho^1, \rho^2 \dots \rho^{\text{out}}\}$ , and apply backward channels  $\mathcal{F}$  to successively obtain  $\{\sigma^{\text{out}}, \sigma^l \dots \sigma^1\}$  in the backward process. These forward and backward terms are used for the gradient evaluation. **d** Exhibition of a quantum processor with six superconducting transmon qubits, which are used to experimentally implement the DQNNs. The transmon qubits ( $Q_1$ – $Q_6$ ) and the bus resonators ( $B_1$  and  $B_2$ ) are marked.



**Fig. 2 | Experimental results for learning a two-qubit quantum channel.** We train the three-layer DQNN<sub>1</sub> with 30 different initial parameters and plot the mean fidelity as a function of training iterations for 10 of them for clarity. The upper left inset shows the distribution of the converged mean fidelities of these 30 different initial parameters. We choose one of the learning curves (marked with triangles), then randomly generate 100 different input quantum states, and test the fidelity between their output states given by the target quantum channel and the trained (untrained) DQNN<sub>1</sub>. In the lower left inset, two curves show the distribution of the fidelities for the trained (untrained) DQNN<sub>1</sub>. The right inset is a schematic illustration of DQNN<sub>1</sub>. At adjacent layers, we apply the quantum perceptrons in the order provided in Supplementary Table 2.

coupled to a common transmission line, which is connected through a Josephson parametric amplifier for high-fidelity single-shot readout of the qubits<sup>38,39</sup>. In order to implement the two-qubit gates in the quantum perceptrons, two separate half-wavelength bus resonators are respectively used to mediate the interactions among the qubits between layers<sup>16,40,41</sup>. The characterized fidelities of single-qubit  $R_x$  gates are above 99.5%, while the average fidelity of the two-qubit gates is 98.4%. The detailed experimental setup and device parameters can be found in Supplementary Note 4.

### Learning a two-qubit quantum channel

The first application of DQNNs is learning a quantum channel. Specifically, we consider using DQNNs to learn a two-qubit target quantum channel. We experimentally implement a three-layer DQNN with two qubits in each layer. This three-layer DQNN is denoted by DQNN<sub>1</sub>. As mentioned above, we employ restricted form of perceptrons due to realistic experimental limitations. In general, the restricted form of perceptrons would reduce the representation power of DQNNs. Therefore, we construct the target quantum channel using the same ansatz as DQNN<sub>1</sub> with randomly generated parameters  $\theta_i$ , so that the target quantum channel would be within the representation range of DQNN<sub>1</sub> (see Methods). Here, we choose  $|00\rangle, |01\rangle, |++\rangle$ , and  $|++i\rangle$  as our input states  $\rho_x^{\text{in}}$ , where the subscript  $x=1, 2, 3, 4$  is the labeling,  $|0\rangle$

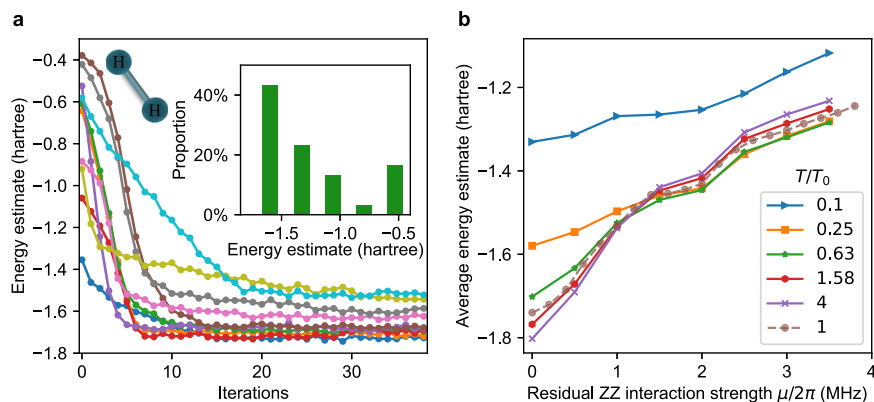
and  $|1\rangle$  are the eigenstates of Pauli Z matrix,  $|+\rangle (|-\rangle)$  is the eigenstate of Pauli X matrix, and  $|+i\rangle$  is the eigenstate of Pauli Y matrix. The four pairs of  $(\rho_x^{\text{in}}, \tau_x^{\text{out}})$  serve as the training dataset, where  $\tau_x^{\text{out}}$  is the corresponding desired output state produced by the target quantum channel. The optimization goal is to maximize the mean fidelity between  $\tau_x^{\text{out}}$  and the measured DQNN output  $\rho_x^{\text{out}}$  averaged over all four input states. The general training procedure goes as follows: (1) Initialization: we randomly choose the initial gate parameters  $\theta$  for all perceptrons in DQNN<sub>1</sub>. (2) Forward process (implemented on our quantum processor): for each training sample  $(\rho_x^{\text{in}}, \tau_x^{\text{out}})$ , we prepare the input layer to  $\rho_x^{\text{in}}$ , then apply layerwise forward channels  $\mathcal{E}^1$  and  $\mathcal{E}^{\text{out}}$ , and extract  $\rho_x^1$  and  $\rho_x^{\text{out}}$  successively by carrying out quantum state tomography<sup>42</sup>. (3) Backward process (implemented on a classical computer): we initialize the output layer to  $\sigma_x^{\text{out}}$ , which is determined by  $\rho_x^{\text{out}}$  and  $\tau_x^{\text{out}}$  (see Supplementary Note 1), and then apply backward channels  $\mathcal{F}^{\text{out}}$  and  $\mathcal{F}^1$  on  $\sigma_x^{\text{out}}$  to successively obtain  $\sigma_x^1$  and  $\sigma_x^0$ . (4) Based on  $\{(\rho_x^{l-1}, \sigma_x^l)\}$ , we evaluate the gradient of the fidelity with respect to all the variational parameters in the adjacent layers  $l-1$  and  $l$ . Then we take the average over the whole training dataset for the final gradient, which is used to update the variational parameters  $\theta$ . (5) Repeat (2), (3), (4) for  $s_0$  rounds. The pseudocode for our algorithm is provided in Supplementary Note 1.

In Fig. 2, we randomly choose 30 different initial parameters  $\theta$ , and then train DQNN<sub>1</sub> to learn the same target quantum channel. We observe that DQNN<sub>1</sub> converges quickly during the training process, with the highest fidelity above 96%. To benchmark the performance of DQNN<sub>1</sub>, we carry out a classical simulation of the training in Supplementary Note 3, where we train DQNN<sub>1</sub> to learn a target quantum channel without considering any experimental imperfections. The numerical results show that the average converged mean fidelity for 50 different initial parameters is above 98%. Compared with the numerical simulation results, the deviation of the final converged fidelities is due to experimental imperfections, including qubit decoherence and residual ZZ interactions between qubits<sup>43-45</sup>. In the upper left inset of Fig. 2, we show the distribution for all the converged fidelities from these 30 repeated experiments. We expect that the distribution will concentrate to a higher fidelity for improved performance of the quantum processor.

To evaluate the performance of DQNN<sub>1</sub>, we choose one training process from the 30 experiments, and refer the DQNN<sub>1</sub> with parameters corresponding to the ending (starting) iteration of the training curve as the trained (untrained) DQNN<sub>1</sub>. We generate other 100 different input quantum states and experimentally measure their corresponding output states produced by the trained (untrained) DQNN<sub>1</sub>. We test the fidelity between output states given by the target channel and the trained (untrained) DQNN<sub>1</sub>. As shown in the lower left inset of Fig. 2, for the trained DQNN<sub>1</sub>, 43% of the fidelities exceed 0.95 and 95% of the fidelities are higher than 0.9, which separate away from the distribution of the results of the untrained DQNN<sub>1</sub>. This contrast illustrates the effectiveness of the training process of DQNN<sub>1</sub>.

### Learning the ground state energy

DQNNs also provide a complementary approach to solving quantum chemistry problems. Here, we apply DQNNs to learn the ground state energy of a given Hamiltonian  $H$  as an example. The optimization goal is to minimize the energy estimate  $\text{tr}(\rho^{\text{out}}H)$  for the output state of the DQNN. We aim to learn the ground state energy of the molecular hydrogen Hamiltonian<sup>46</sup>. By exploiting the Bravyi-Kitaev transformation and certain symmetry, the Hamiltonian of molecular hydrogen can be reduced to the effective Hamiltonian acting on two qubits:  $\hat{H}_{\text{BK}} = g_0\mathbb{1} + g_1Z_0 + g_2Z_1 + g_3Z_0Z_1 + g_4Y_0Y_1 + g_5X_0X_1$ , where  $X_i, Y_i, Z_i$  are Pauli operators on the  $i$ -th qubit, and coefficients  $g_j$  ( $j=0, \dots, 5$ ) depend on the fixed bond length of molecular hydrogen. We consider the bond length 0.075 nm in this work and the corresponding coefficients  $g_i$  can be found in ref. 46.



**Fig. 3 | Experimental and numerical results for learning the ground state energy of molecular hydrogen.** **a** Experimental energy estimate at each iteration during the learning process for different initial parameters. The inset displays the distribution of converged energy estimates of 30 different initial parameters. **b** Numerical results for the mean energy estimate with different coherence time  $T$

and residual ZZ interaction strength  $\mu$  between qubits. Specifically, we assume the same  $\mu$  between all neighboring qubits and adjust the coherence time by the same ratio ( $T/T_0$ ) for all qubits, where  $T$  and  $T_0$  are the coherence times in the numerical simulation and the experiment, respectively.

We use DQNN<sub>1</sub> again as the variational ansatz to learn the ground state of molecular hydrogen with the following procedure, similar to the previous one of learning a quantum channel: (1) Initialization: we prepare the input layer to the fiducial product state  $|00\rangle$ , and randomly generate initial gate parameters  $\theta$  for DQNN<sub>1</sub>. (2) In the forward process (implemented on the quantum processor), we apply forward channels  $\mathcal{E}^1$  and  $\mathcal{E}^{\text{out}}$  in succession, and extract quantum states of the hidden layer ( $\rho^1$ ) and the output layer ( $\rho^{\text{out}}$ ) by quantum state tomography. (3) In the backward process (implemented on a classical computer), we initialize the quantum state of the output layer to  $\sigma^{\text{out}}$ , and then obtain  $\sigma^1$  and  $\sigma^0$  after successively applying backward channels  $\mathcal{F}^{\text{out}}$  and  $\mathcal{F}^1$  on  $\sigma^{\text{out}}$ . (4) Based on  $\{(\rho^{l-1}, \sigma^l)\}$ , we calculate the gradient of the energy estimate with respect to all the variational parameters in the adjacent layers  $l-1$  and  $l$ , and then update all gate parameters in DQNN<sub>1</sub>. (5) Repeat (2), (3), (4) for  $s_0$  rounds. The pseudocode for our algorithm is provided in Supplementary Note 1.

We train DQNN<sub>1</sub> with 30 different initial parameters and show our experimental results in Fig. 3a. We observe that DQNN<sub>1</sub> converges within 20 iterations. The lowest ansatz energy estimate reaches below  $-1.727$  (hartree) in the learning process, with an accuracy up to 93.3% compared to the theoretical value of the ground state energy  $-1.851$  (hartree). This shows the good performance of DQNN<sub>1</sub> and the accuracy of our experimental system control. The inset of Fig. 3a shows the distribution of all the converged energy from these 30 repeated experiments with different initial parameters, six of which have accuracy above 90%.

To numerically investigate the effects of experimental imperfections on training DQNNs, we consider two possible sources: decoherence of qubits and residual ZZ interactions between qubits (see Methods for details). Under different residual ZZ interaction strength  $\mu$  and different decoherence time  $T$ , we numerically train the DQNN with 30 different initial parameters to learn the ground state energy of the molecular hydrogen. We find that for four of these initial parameters DQNN<sub>1</sub> converges to local minima instead of the global minimum. Excluding these abnormal instances with local minima, we plot the average energy estimate as a function of  $\mu$  with different  $T/T_0$  in Fig. 3b, where  $T_0$  is the experimentally measured qubit coherence time. The numerical results show that, when there is no residual ZZ interaction, the decoherence of the qubits at  $T/T_0 = 1$  degrades the accuracy of the average energy estimate by 6% to  $-1.74$  (the leftmost point of the dotted line). At  $\mu/2\pi \approx 1$  MHz (close to our experimental characterization) with  $T/T_0 = 1$ , the average energy estimate is  $-1.53$ , which is about 17% higher than the theoretical value and is comparable to the experimental values shown in Fig. 3a. Apparently, such a large  $\mu$

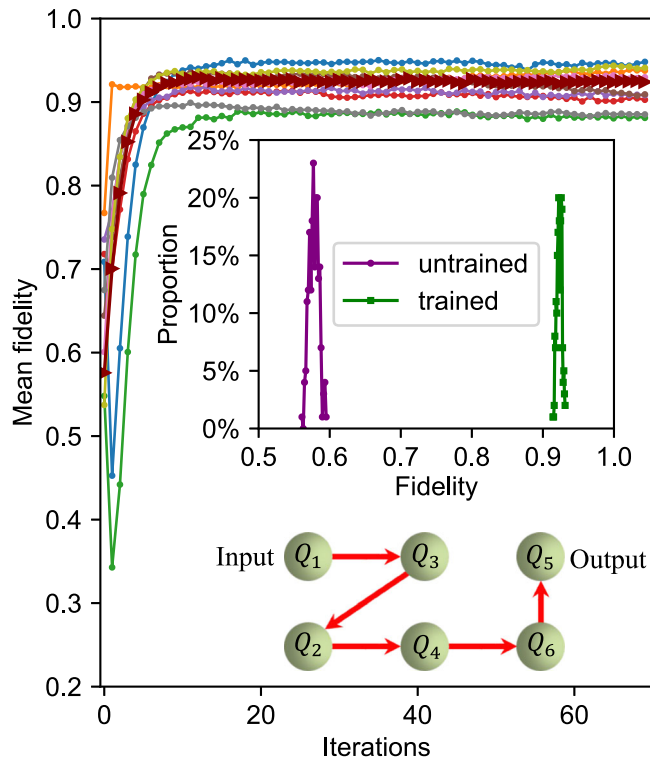
dominantly limits the training performance, while the variation of the coherence time has a minor effect. This is anticipated given the fact that the total running time of the DQNN (1.2  $\mu\text{s}$ ) is significantly shorter than the average characteristic coherence time of the qubits (7.5  $\mu\text{s}$ ). These experimental imperfections can be suppressed after introducing advanced technologies in the design and fabrication of better superconducting quantum circuits, such as tunable couplers<sup>47–49</sup> and tantalum-based qubits<sup>50,51</sup>.

### Learning a one-qubit quantum channel

To further illustrate the efficiency of the quantum BP algorithm, we construct another DQNN with four hidden layers (denoted as DQNN<sub>2</sub>) by rearranging our six-qubit quantum processor into a six-layer structure, with one-qubit respectively in each layer. We focus on the task of learning a one-qubit target quantum channel, which is constructed using the same ansatz as DQNN<sub>2</sub> with randomly generated parameters (see Methods). We choose  $|0\rangle, |1\rangle, |-\rangle$  as our input states and compare the measured output states of DQNN<sub>2</sub> with the desired ones from the target single-qubit quantum channel. The general training procedure is similar as in training DQNN<sub>1</sub> discussed above. Our experimental results are summarized in Fig. 4, which shows the learning curves for 10 different initial parameters. We find that DQNN<sub>2</sub> can learn the target quantum channel with a mean fidelity up to 94.8%. We notice that the variance among the converged mean fidelity in DQNN<sub>2</sub> is smaller than that for DQNN<sub>1</sub>, which may be attributed to the smaller total circuit depth and thus less error accumulation due to experimental imperfections. To study the learning performance, we choose one of these learning curves (marked in triangles), and refer DQNN<sub>2</sub> with parameters corresponding to the ending (starting) iteration of the learning curve as the trained (untrained) DQNN<sub>2</sub>. We then use other 100 different input quantum states to test the trained and untrained DQNN<sub>2</sub> by measuring the fidelities between the experimental output states and the corresponding desired ones given by the target quantum channel. As shown in the upper inset of Fig. 4, the fidelity distribution concentrates around 0.92 for the trained DQNN<sub>2</sub>, which stands in stark contrast to that of the untrained DQNN<sub>2</sub> and thus indicates a good performance after training.

### Discussion

In this work, we experimentally perform the forward process while implementing the backward process on a classical computer. For the task of learning a target quantum channel, we note that the backward process can also be implemented with a quantum device in principle (see Supplementary Note 1 for an experimental proposal). Yet, it is



**Fig. 4 | Experimental results for learning a one-qubit quantum channel.** The mean fidelity of training the six-layer DQNN<sub>2</sub> is plotted as the function of training iterations for different initial parameters. We randomly generate 100 different single-qubit states, and evaluate the fidelities between their output states produced by DQNN<sub>2</sub> and their desired output states given by the target quantum channel. The upper inset displays the distribution in two cases: a well-trained DQNN<sub>2</sub> (marked by squares) and an untrained (marked by circles) DQNN<sub>2</sub>, both are defined with the learning curve marked in triangles. The lower inset is a schematic illustration of DQNN<sub>2</sub>, where we apply the perceptrons in the order indicated by the direction of the arrows.

more challenging to achieve the required experimental accuracy for backward channels than that for forward channels. In the future, we expect an experimental implementation of the backward channel on a quantum processor with better performance.

The efficiency of the quantum BP algorithm can be measured in terms of the required number of copies for each training data per training iteration (see Supplementary Note 2 for detailed discussions). For the training with the quantum BP algorithm, this number scales exponentially with the number of qubits in the hidden layers and the output layer. This is due to the fact that an exponential number of measurements are required for quantum state tomography and the experimental implementation of backward channels. Compared with the training not utilizing the quantum BP algorithm, we find that the use of the quantum BP algorithm can enhance the training efficiency in some cases where DQNNs have narrow hidden layers. Such DQNNs can be used as quantum autoencoders to compress and denoise quantum data<sup>52</sup>.

In summary, we have demonstrated the training of deep quantum neural networks on a six-qubit programmable superconducting quantum processor. We experimentally exhibit its intriguing ability to learn quantum channels and learn the ground state energy of a given Hamiltonian. The quantum BP algorithm demonstrated in our experiments can be directly applied to DQNNs with extended widths and depths. With further improvements in experimental conditions, quantum perceptrons with enhanced expressive power are expected to be constructed with deeper circuits, which allows DQNNs to tackle more challenging tasks in the future.

## Methods

### Framework

We consider a DQNN that includes  $L$  hidden layers with a total number  $m_l$  of qubits in layer  $l$ . The qubits in two adjacent layers are connected with quantum perceptrons and each perceptron consists of two single-qubit rotation gates  $R_x(\theta_1)$  and  $R_x(\theta_2)$  along the  $x$  axis with variational angles  $\theta_1$  and  $\theta_2$ , respectively, followed by a fixed two-qubit controlled-Phase gate. The unitary of the quantum perceptron that acts on the  $i$ -th qubit at layer  $l-1$  and the  $j$ -th qubit at layer  $l$  in the DQNN is written as  $U_{(i,j)}^l(\theta_{(i,j),1}^l, \theta_{(i,j),2}^l)$ . Then the unitary product of all quantum perceptrons acting on the qubits in layers  $l-1$  and  $l$  is denoted as  $U^l = \prod_{j=1}^{m_l} \prod_{i=1}^{m_{l-1}} U_{(i,j)}^l$ . The DQNN acts on the input state  $\rho^{\text{in}}$  and produces the output state  $\rho^{\text{out}}$  according to

$$\rho^{\text{out}} \equiv \text{tr}_{\text{in, hid}}(\mathcal{U}(\rho^{\text{in}} \otimes |0 \dots 0\rangle_{\text{hid, out}} \langle 0 \dots 0|) \mathcal{U}^\dagger), \quad (1)$$

where  $\mathcal{U} \equiv U^{\text{out}} U^L U^{L-1} \dots U^1$  is the unitary of the DQNN, and all qubits in the hidden layers and the output layer are initialized to a fiducial product state  $|0 \dots 0\rangle$ . The characteristic of the layer-by-layer architecture enables  $\rho^{\text{out}}$  to be expressed as a series of maps on  $\rho^{\text{in}}$ :

$$\rho^{\text{out}} = \mathcal{E}^{\text{out}}(\mathcal{E}^L(\dots \mathcal{E}^2(\mathcal{E}^1(\rho^{\text{in}})) \dots)), \quad (2)$$

where  $\mathcal{E}^l(\rho^{l-1}) \equiv \text{tr}_{l-1}(U^l(\rho^{l-1} \otimes |0 \dots 0\rangle_l \langle 0 \dots 0|) U^{l\dagger})$  is the forward quantum channel.

In the Supplementary Information, we prove that for the two machine learning tasks in our work, the derivative of the mean fidelity or the energy estimate with respect to  $\theta_{(i,j),k}^l$  can be calculated with the information of layers  $l-1$  and  $l$ , which can be written as  $G(\boldsymbol{\theta}^l, \rho^{l-1}, \sigma^l)$  with  $\boldsymbol{\theta}^l$  incorporating all parameters in layers  $l-1$  and  $l$ . We note that  $\rho^{l-1} = \mathcal{E}^{l-1}(\dots \mathcal{E}^2(\mathcal{E}^1(\rho^{\text{in}})) \dots)$  refers to the quantum state in layer  $l-1$  in the forward process, and  $\sigma^l = \mathcal{F}^{l+1}(\dots \mathcal{F}^{\text{out}}(\dots))$  represents the backward term in layer  $l$  with  $\mathcal{F}^l$  being the adjoint channel of  $\mathcal{E}^l$ . The backward channel  $\mathcal{F}^l$  applies on the backward term  $\sigma^l$  and produces  $\sigma^{l-1}$  according to  $\sigma^{l-1} = \mathcal{F}^l(\sigma^l) = \text{tr}_l((\mathbb{I}_{l-1} \otimes |0\rangle_l \langle 0|) U^l (\mathbb{I}_{l-1} \otimes \sigma^l) U^{l\dagger})$ .

### Generating random input quantum states

To evaluate the learning performance in the task of learning a target quantum channel, we need to generate many different input quantum states and test the fidelity between their output states produced by DQNN<sub>1</sub> and their desired output states given by the target quantum channel.

For the task of learning a two-qubit quantum channel, we generate these input quantum states by separately applying single-qubit rotation gates  $R_{a_1}(\Omega_1) \otimes R_{a_2}(\Omega_2)$  on the two qubits initialized in  $|00\rangle$ . Here each rotation gate has a random rotation axis  $a_i$  in the  $x$ - $y$  plane and a random rotation angle  $\Omega_i$ .

For the task of learning a one-qubit quantum channel, we generate the input quantum states by applying single-qubit rotation gates  $R_b(\Phi)$  on the input qubit initialized in  $|0\rangle$  with a random rotation axis  $b$  in the  $x$ - $y$  plane and a random rotation angle  $\Phi$ .

### The target quantum channels

The target quantum channels are constructed using the DQNN ansatz with randomly chosen parameters  $\boldsymbol{\theta}_r$ . For DQNN<sub>1</sub>, the schematic illustration is shown in the right inset of Fig. 2. The input state is encoded in qubits  $Q_1$  and  $Q_2$  (input layer). After completing all the layerwise quantum perceptrons, the output state is obtained from qubits  $Q_5$  and  $Q_6$  (output layer). Each quantum perceptron is applied with randomly chosen single-qubit rotation angles, which are provided in the third column of Supplementary Table 2. The target quantum channel for DQNN<sub>2</sub> is constructed in a similar way, with the corresponding parameters (randomly chosen) also provided in Supplementary Table 2.

## Numerical simulations

We numerically consider two sources for experimental imperfections: decoherence of qubits and residual ZZ interactions between qubits. In our simulation, we consider the effects of the residual ZZ interactions only during the implementation of rotation gates for simplicity. Specifically,  $R_{x,i}(\theta)$  (rotation gate  $R_x$  on the  $i$ -th qubit) is realized by the evolution of a time-dependent Hamiltonian of the form  $H_0(t) = \theta(a_i^\dagger + a_i)h(t)/2$ , where  $a_i$  ( $a_i^\dagger$ ) is the corresponding annihilation (creation) operator, and  $h(t)$  is the time-dependent driving strength given as a Gaussian function  $h(t) = Ae^{-(t/\delta)^2}$ . We set the evolution time of the rotation gate as  $\Delta t = 40$  ns and  $\delta = 10$  ns. The unwanted residual ZZ interactions between qubits are considered by adding a Hamiltonian of the form  $H_1(t) = \sum_{\langle i,j \rangle} \mu_{i,j} a_i^\dagger a_i a_j^\dagger a_j$  to  $H_0(t)$ , where  $\langle i,j \rangle$  denotes the interaction between nearest-neighbor qubits, and  $\mu_{i,j}$  is the interaction strength between the  $i$ -th and  $j$ -th qubits. The qubit decoherence is considered by adding the relaxation term  $\sqrt{1/T_{1,i}} a_i$  and the pure dephasing term  $\sqrt{1/T_{\phi,i}} a_i^\dagger a_i$  as the collapse operators to the evolution. Here,  $T_{1,i}$  and  $T_{\phi,i}$  are the energy relaxation time and the pure dephasing time of the  $i$ -th qubit, respectively.

## Data availability

The data generated in this study have been deposited in the Figshare database under accession code <https://doi.org/10.6084/m9.figshare.22802501><sup>53</sup>.

## Code availability

The codes for numerical simulations are available at <https://zenodo.org/badge/latestdoi/552193072><sup>54</sup>.

## References

- Silver, D. et al. Mastering the game of Go without human knowledge. *Nature* **550**, 354 (2017).
- Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583 (2021).
- Goodfellow, I., Bengio, Y., and Courville, A. Deep learning (The MIT Press, Cambridge, 2016). <https://www.deeplearningbook.org/>
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436 (2015).
- Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195 (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).
- Das Sarma, S., Deng, D.-L. & Duan, L.-M. Machine learning meets quantum physics. *Phys. Today* **72**, 48 (2019).
- Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L. & Coles, P. J. Challenges and opportunities in quantum machine learning. *Nat. Comput. Sci.* **2**, 567 (2022).
- Dawid, A. et al. Modern applications of machine learning in quantum sciences. *arXiv* <http://arxiv.org/abs/2204.04198> (2022).
- Huang, H.-Y., Kueng, R. & Preskill, J. Information-theoretic bounds on quantum advantage in machine learning. *Phys. Rev. Lett.* **126**, 190505 (2021).
- Liu, Y., Arunachalam, S. & Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *Nat. Phys.* **17**, 1013 (2021).
- Gao, X., Zhang, Z.-Y. & Duan, L.-M. A quantum machine learning algorithm based on generative models. *Sci. Adv.* **4**, eaat9004 (2018).
- Abbas, A. et al. The power of quantum neural networks. *Nat. Comput. Sci.* **1**, 403 (2021).
- Herrmann, J. et al. Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases. *Nat. Commun.* **13**, 4144 (2022).
- Ren, W. et al. Experimental quantum adversarial learning with programmable superconducting qubits. *Nat. Comput. Sci.* **2**, 711 (2022).
- Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209 (2019).
- Hu, L. et al. Quantum generative adversarial learning in a superconducting quantum circuit. *Sci. Adv.* **5**, eaav2761 (2019).
- Blank, C., Park, D. K., Rhee, J.-K. K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quantum Inf.* **6**, 1 (2020).
- Li, Z., Liu, X., Xu, N. & Du, J. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett.* **114**, 140504 (2015).
- Zhu, D. et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **5**, eaaw9918 (2019).
- Gong, M. et al. Quantum neuronal sensing of quantum many-body states on a 61-qubit programmable superconducting processor. *Sci. Bull.* **68**, 906 (2023).
- Huang, H.-Y. et al. Quantum advantage in learning from experiments. *Science* **376**, 1182 (2022).
- Beer, K. et al. Training deep quantum neural networks. *Nat. Commun.* **11**, 808 (2020).
- Liu, Z., Duan, L.-M. & Deng, D.-L. Solving quantum master equations with deep quantum neural networks. *Phys. Rev. Res.* **4**, 013097 (2022).
- Wilkinson, S. A. and Hartmann, M. J. Evaluating the performance of sigmoid quantum perceptrons in quantum neural networks. <http://arxiv.org/abs/2208.06198> (2022).
- Huber, P. et al. Realization of a quantum perceptron gate with trapped ions. <http://arxiv.org/abs/2111.08977> (2021).
- Moreira, M. S. et al. Realization of a quantum neural network using repeat-until-success circuits in a superconducting quantum processor. <http://arxiv.org/abs/2212.10742> (2022).
- Pechal, M. et al. Direct implementation of a perceptron in superconducting circuit quantum hardware. *Phys. Rev. Res.* **4**, 033190 (2022).
- Beer, K. Quantum neural networks. <http://arxiv.org/abs/2205.08154> (2022).
- Koch, J. et al. Charge insensitive qubit design from optimizing the cooper-pair box. *Phys. Rev. A* **76**, 042319 (2007).
- 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).
- Kono, S. et al. Breaking the trade-off between fast control and long lifetime of a superconducting qubit. *Nat. Commun.* **11**, 3683 (2020).
- Carusotto, I. et al. Photonic materials in circuit quantum electrodynamics. *Nat. Phys.* **16**, 268 (2020).
- Negirneac, V. et al. High-fidelity controlled-Z gate with maximal intermediate leakage operating at the speed limit in a superconducting quantum processor. *Phys. Rev. Lett.* **126**, 220502 (2021).
- Blais, A., Grimsmo, A. L., Girvin, S. M. & Wallraff, A. Circuit quantum electrodynamics. *Rev. Mod. Phys.* **93**, 025005 (2021).
- Roy, T. et al. Broadband parametric amplification with impedance engineering: beyond the gain-bandwidth product. *Appl. Phys. Lett.* **107**, 262601 (2015).
- Murch, K. W., Weber, S. J., Macklin, C. & Siddiqi, I. Observing single quantum trajectories of a superconducting quantum bit. *Nature* **502**, 211 (2013).
- Majer, J. et al. Coupling superconducting qubits via a cavity bus. *Nature* **449**, 443 (2007).
- Song, C. et al. Generation of multicomponent atomic schrodinger cat states of up to 20 qubits. *Science* **365**, 574 (2019).

42. Nielsen, M. A., and Chuang, I. L., Quantum computation and quantum information (Cambridge University Press, Cambridge, 2010) <https://doi.org/10.1017/CBO9780511976667>.
43. Zhao, P. et al. High-contrast zz interaction using superconducting qubits with opposite-sign anharmonicity. *Phys. Rev. Lett.* **125**, 200503 (2020).
44. Ku, J. et al. Suppression of unwanted zz interactions in a hybrid two-qubit system. *Phys. Rev. Lett.* **125**, 200504 (2020).
45. Kandala, A. et al. Demonstration of a high-fidelity cnot gate for fixed-frequency transmons with engineered zz suppression. *Phys. Rev. Lett.* **127**, 130501 (2021).
46. O'Malley, P. J. J. et al. Scalable quantum simulation of molecular energies. *Phys. Rev. X* **6**, 031007 (2016).
47. Li, X. et al. Tunable coupler for realizing a controlled-phase gate with dynamically decoupled regime in a superconducting circuit. *Phys. Rev. Appl.* **14**, 024070 (2020).
48. Collodo, M. C. et al. Implementation of conditional phase gates based on tunable zz interactions. *Phys. Rev. Lett.* **125**, 240502 (2020).
49. Sung, Y. et al. Realization of high-fidelity cz and zz-free iswap gates with a tunable coupler. *Phys. Rev. X* **11**, 021058 (2021).
50. Place, A. P. M. et al. New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds. *Nat. Commun.* **12**, 1779 (2021).
51. Wang, C. et al. Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds. *npj Quantum Inf.* **8**, 3 (2022).
52. Bondarenko, D. & Feldmann, P. Quantum autoencoders to denoise quantum data. *Phys. Rev. Lett.* **124**, 130502 (2020).
53. Pan, X. and Lu, Z. Deep quantum neural networks on a superconducting processor. Figshare Dataset. <https://doi.org/10.6084/m9.figshare.22802501> (2023).
54. Pan, X. and Lu, Z. Deep quantum neural networks on a superconducting processor. Zenodo Database. <https://zenodo.org/badge/latestdoi/552193072> (2023).

## Acknowledgements

We thank Wenjie Jiang for the helpful discussions. We acknowledge the support of the National Natural Science Foundation of China (Grants No. 92165209, No. 11925404, No. 11874235, No. 11874342, No. 11922411, No. 12061131011, No. T2225008, No. 12075128), the National Key Research and Development Program of China (Grants No. 2017YFA0304303), 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. carried out the experiments and analyzed the data with the assistance of Z.H. and Y.X.; L.S. directed the experiments; Z.L. formalized the theoretical framework and performed the numerical simulations under the supervision of D.-L.D.; W.L. and C.-L.Z. provided theoretical support; W.C. fabricated the parametric amplifier; 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., Z.L., W.L., C.-L.Z., D.-L.D., and L.S. wrote the manuscript with feedback from all authors.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-023-39785-8>.

**Correspondence** and requests for materials should be addressed to Dong-Ling Deng or Luyan Sun.

**Peer review information** *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

**Reprints and permissions 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 licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence 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 licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023