



OPEN

## Quantum neural networks with multi-qubit potentials

Yue Ban<sup>1</sup>✉, E. Torrontegui<sup>2,3</sup> & J. Casanova<sup>4,5,6</sup>

We propose quantum neural networks that include multi-qubit interactions in the neural potential leading to a reduction of the network depth without losing approximative power. We show that the presence of multi-qubit potentials in the quantum perceptrons enables more efficient information processing tasks such as XOR gate implementation and prime numbers search, while it also provides a depth reduction to construct distinct entangling quantum gates like CNOT, Toffoli, and Fredkin. This simplification in the network architecture paves the way to address the connectivity challenge to scale up a quantum neural network while facilitating its training.

Information is a resource due to its advance and expansion in the digitalization and control<sup>1</sup>. However, programming explicit algorithms with good performance may become unfeasible due to the vertiginous growth in (i) the amount of available information with which classical algorithms have to deal<sup>2</sup>, and (ii) the inherent difficulty of finding efficient algorithms for specific problems<sup>3</sup>. All these limit our current capabilities in information processing tasks. Two alternative approaches that would contest these limitations are machine learning and quantum computing<sup>4,5</sup>.

On the one hand, machine learning (ML) is a branch of artificial intelligence that uses statistical techniques to give computers the ability to progressively learn with input data without being explicitly programmed. ML is based on the generation of a hypothesis that is optimized from sample inputs and re-used to generate new predictions<sup>6</sup>. Thus the algorithms can learn from data and overcome the static program instructions by making data-driven decisions from sample inputs. Among the distinct hypothesis models, neural networks<sup>7</sup> are very extended due to the blooming of deep learning<sup>8,9</sup>. Artificial neural networks are organized in layers and each layer learns new behavior patterns<sup>10</sup>. The computational power of artificial neural networks relies on this architecture where *neurons* in each layer feed signals into other neurons allowing parallel-processed computing<sup>11,12</sup>. In this manner, several calculations can be performed at the same time, and large computational problems can often be divided into smaller ones, which can be then solved simultaneously. The versatility of neural networks to classify complex data relies on the *universal approximation theorem*, which leads artificial neural networks the capacity to approximate any function<sup>13</sup>. As result, they span a broad range of applications such as speech<sup>14</sup> or object recognition<sup>15</sup>, spam filters<sup>16</sup>, vehicle control<sup>17,18</sup>, trajectory prediction<sup>19</sup>, decision making<sup>20</sup>, game-playing<sup>21</sup>, or automated trading systems<sup>22</sup>.

On the other hand, quantum computing represents a different paradigm from classical information processing. Based on an alternative information encoding that exploits the quantum properties of matter, systems that encompass several quantum bits (qubits) are exponentially hard to simulate with classical devices<sup>5</sup> showing that quantum systems do not seem to obey the Church thesis<sup>23</sup>, and consequently they are not polynomially equivalent to classical systems. Then, quantum systems harnessed as computational devices, might be dramatically more powerful than any other classical system<sup>24</sup>. The universality of quantum computing<sup>25</sup> expands a broad range of applications. Some illustrative examples are linear systems solvers<sup>26</sup>, molecule simulators<sup>27</sup>, combinatorial optimizers<sup>28</sup>, black-box<sup>29</sup> and factorization problems<sup>30</sup>, or Hamiltonian simulations<sup>31</sup>. Although a set of single  $N = 1$  and two-qubit  $N = 2$  gates is a universal approximator<sup>32</sup>, larger multi-qubit ( $N > 2$ ) gates may offer a computational advantage that reduces complexity in existing algorithms<sup>33–35</sup>.

Quantum machine learning<sup>36–44</sup> aims for the symbiosis of both paradigms to the mutual reinforcement. To achieve improvements of the machine learning protocols by leveraging quantum resources in comparison to their classical counterparts is the goal for the future. On the other hand, the universality of artificial networks may enhance the accuracy and efficiency of quantum protocols<sup>45–49</sup>. By making analogy to classical neural networks, a quantum neural network (QNN) consists of quantum perceptrons (neurons) possessing nonlinear activation

<sup>1</sup>TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain. <sup>2</sup>Departamento de Física, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés (Madrid), Spain. <sup>3</sup>Instituto de Física Fundamental IFF-CSIC, Calle Serrano 113, 28006 Madrid, Spain. <sup>4</sup>Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain. <sup>5</sup>EHU Quantum Center, University of the Basque Country UPV/EHU, Leioa, Spain. <sup>6</sup>Basque Foundation for Science, IKERBASQUE, Plaza Euskadi 5, 48009 Bilbao, Spain. ✉email: ybanxc@gmail.com

functions in different layers. In the network, the hidden layers in a QNN are the intermediate ones which are composed of quantum perceptrons, each of which is a qubit encoded in an Ising Hamiltonian<sup>44</sup>. By measuring the excitation probability of the reduced eigenstate of such a Hamiltonian, one can get the nonlinear activation function. In this work, we propose an extension of a universal QNN enabling multi-qubit ( $N > 2$ ) interactions that lead to a reduction of the network depth while keeping the approximation power. As a result, achieving a simplification of existing protocols not only requires shorter operation times, but it may also introduce less accumulation of errors due to the reduction in the amount of requested gates. The article is structured as follows: We firstly define that the quantum perceptron is a single-output quantum neuron with multi-qubit interactions, connecting to several-input quantum neurons without hidden layers. The result of nesting several quantum perceptrons is a QNN. We show that a quantum perceptron with multi-qubit interactions can do an XOR gate and prime number search from 3 to 5 bits, improving the performance of approximation power compared to the classical counterpart, as well as compared to standard QNNs (i.e. QNNs that include nested perceptrons without multi-qubit interactions). Then, we show that quantum gates such as CNOT, Toffoli, and Fredkin can be implemented by QNN involving quantum perceptrons with multi-qubit interactions, thus reducing the circuit depth as it is not needed to add hidden layers.

## Results

**Quantum perceptrons with multi-qubit potentials.** A quantum perceptron, or a quantum neuron, is the basic building block of a QNN. It can be constructed as a qubit that presents a nonlinear response to an input potential  $\hat{x}_j$  in the excitation probability. This can be written as the following quantum gate acting on a  $j$ th qubit that encodes the quantum perceptron<sup>44,45</sup>:

$$\hat{U}_j(\hat{x}_j; f)|0_j\rangle = \sqrt{1 - f(\hat{x}_j)}|0_j\rangle + \sqrt{f(\hat{x}_j)}|1_j\rangle, \quad (1)$$

where

$$f(x) = \frac{1}{2} \left( 1 + \frac{x}{\sqrt{1+x^2}} \right). \quad (2)$$

corresponds to a nonlinear function. The transformation in Eq. (1) can be engineered, e.g., by evolving adiabatically the qubit with the Hamiltonian

$$\hat{H} = \frac{1}{2} [\hat{x}_j \hat{\sigma}_j^z + \Omega(t) \hat{\sigma}_j^x] \quad (3)$$

where  $\hat{x}_j$  is the potential exerted by other neurons on the perceptron, and the applied external field  $\Omega(t)$  leads to a tunable energy gap in the dressed-state qubit basis  $|\pm\rangle$ , with  $\hat{\sigma}_j^x|\pm\rangle = \pm|\pm\rangle$ . Typically,  $\hat{x}_j = \sum_{i=1}^k (w_{ji} \hat{\sigma}_i^z) + b_j$ <sup>44,45</sup> which implies that the  $j$ th perceptron is coupled to a number  $k$  of neurons (labelled with  $i$ ) in the previous/input layer via standard spin-spin interactions. The Hamiltonian in Eq. (3) has the following reduced eigenstate (i.e. when the degrees of freedom of any other neuron are traced-out):

$$|\Phi(x_j/\Omega(t))\rangle = \sqrt{1 - f(x_j/\Omega(t))}|0_j\rangle + \sqrt{f(x_j/\Omega(t))}|1_j\rangle, \quad (4)$$

with  $f(x)$ , the excitation probability, in the form of Eq. (2). Specifically, to accomplish  $\hat{U}_j(\hat{x}_j; f)|0_j\rangle$ , one can use a Hadamard gate to firstly get the transformation  $|0_j\rangle \rightarrow |+\rangle_j$  and finally obtain  $|\Psi\rangle = |\Phi(x_j/\Omega(t_f))\rangle$  at a certain time  $t_f$  by evolving the system adiabatically with Hamiltonian (3). With the fixed trajectory always along with the instantaneous eigenstate of the Hamiltonian, one can deduce the external driving  $\Omega(t)$  from the fast quasi-adiabatic passage<sup>44</sup>. In this manner, the nonlinear activation function of the quantum perceptron is encoded in the probability of the excited state  $P_j = f(x_j) = \frac{1}{2}(1 + \langle \hat{\sigma}_j^z \rangle)$  during the adiabatic evolution<sup>44</sup>. In order to speed up the operation of this perceptron, one can also use inverse engineering techniques which directly impose conditions in the wave function evolution at the initial and final time instants, resulting in nonlinear response in the quantum perceptron<sup>45</sup>. Correspondingly, a smoother control  $\Omega(t)$  easily to be implemented experimentally can also be found. In addition, this accelerated activation mechanism by inverse engineering for the quantum neurons would reduce the decoherence and the variation of the input potential induced by neurons in the previous layer leading to enhanced performance.

Now, we introduce a different type of potentials which rely on the possibility to implement multi-qubit interactions. In particular, we consider a potential of the kind

$$\hat{x}_j = \sum_{i=1}^k (w_{ji} \hat{\sigma}_i^z) + w_m \hat{\sigma}_{l_1}^z \cdots \hat{\sigma}_{l_m}^z + b_j. \quad (5)$$

where  $w_m$  is a multi-qubit coefficient marked by the subscript  $m$ , and  $l_p \in [1, 2, \dots, k]$  (namely, the term involving several Pauli matrices includes products of an arbitrary number of neurons in the previous input). For the sake of simplicity in the presentation, there is only a single multi-qubit term in Eq. (5). However, this can include several products of distinct neurons in the input layer (i.e. additional multi-qubit terms). Later we will provide specific examples of these interactions associated to definite problems.

In the following, we show that the multi-qubit potential enables tasks such as (i) Constructing XOR gates at the perceptron level (ii) Searching prime numbers and (iii) Encoding quantum gates. All these are implemented

without hidden layers and/or ancillary qubits, thus, showing the significant role of multi-qubit potentials in the simplification of QNNs.

**XOR gate.** As a classical perceptron is a linear separator, a nonlinear logic gate such as the well-known XOR problem (i.e., the exclusive OR boolean function) requires at least one hidden layer to be implemented in classical neural networks<sup>51</sup>. Now we show that a quantum perceptron with multi-qubit interactions in the neural potential is a nonlinear classifier. In particular, we illustrate the construction of an XOR gate by a single quantum perceptron with multi-qubit interactions. We also show that the lack of hidden layers prevents classical neural networks and standard QNNs to achieve the same task.

As the values of the output neuron 0 and 1 cannot be separated linearly, a classical perceptron with two inputs and one output (i.e. without hidden layers) fails to solve the XOR gate. To show this, we use the standard gradient descent algorithm to train this simple classical perceptron with a sigmoidal activation potential  $f(x)$ , see Eq. (2), and  $x_j = \sum_{i=1}^k w_{ij}s_i + b_j$  with the classical input  $s_i \in \{0, 1\}$  whose cost function is in the form of the mean square value

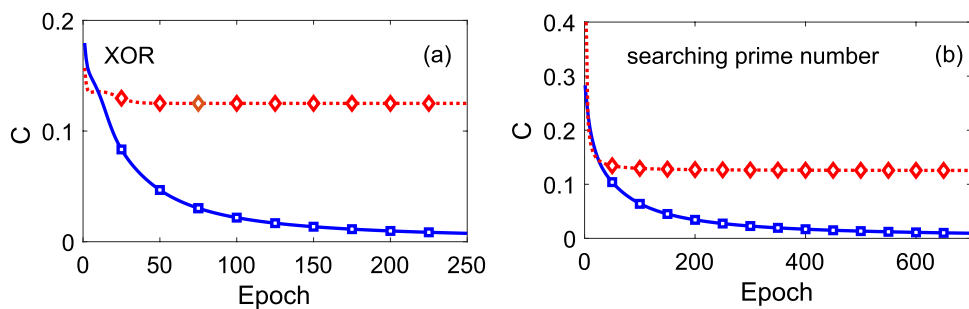
$$C = \frac{1}{2N} \sum_{n=1}^N (y^{(n)} - t^{(n)})^2. \tag{6}$$

However, besides a sigmoid function, neurons in other forms of nonlinear activation functions, such as a ReLU function, can also create a neural network to learn the XOR gate. Here,  $N = 4$  determines the four possible examples 00, 10, 01, 11, while  $y^{(n)}$  and  $t^{(n)}$  are the output and the target respectively, for the  $n$ th example. For simplicity, the subscript  $j$  labelling one perceptron will be neglected in the following text. During the training, the parameters of the classical perceptron are updated after each epoch as

$$\begin{aligned} \tilde{w}_i &= w_i - \eta \frac{\partial C}{\partial w_i} = w_i - \frac{\eta}{N} \sum_{n=1}^N (y^{(n)} - t^{(n)}) f'(x), \\ \tilde{b} &= b - \eta \frac{\partial C}{\partial b} = b - \frac{\eta}{N} \sum_{i=1}^N (y^{(n)} - t^{(n)}) f'(x), \end{aligned} \tag{7}$$

with the learning rate  $\eta$ . Any other nonlinear function  $g(x)$ , similar to  $f(x)$ , can be applied to train this perceptron in order to obtain the same approximation power. As it is shown in Fig. 1a, the cost function value of this classical perceptron sticks in  $C = 0.125$  (dotted-red line with superimposed diamonds for a better identification). As a classical perceptron can only converge on linearly separable data, it is not able to imitate the XOR function. In order to complete an XOR, a hidden layer with two neurons is needed. These two neurons can be regarded to perform an OR and a NAND gate.

To implement an XOR gate with a quantum perceptron without the multi-qubit term in Eq. (5), this is by using the potential  $\hat{x} = w_1 \hat{\sigma}_1^z + w_2 \hat{\sigma}_2^z + b$ , one can follow the procedure described in Ref.<sup>44</sup>. To encode an XOR gate, the neural potential is derived from the four basis states  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ , which play the role of the four examples of the input in the XOR gate (namely, 00, 10, 01, 11). In this case, as the input values of the XOR gate are bits, one can transform them into the measurement value of  $\hat{\sigma}^z$  of the input qubits for the perceptron, i.e.  $\sigma_{in}^z = \frac{1}{2}(1 + \langle \hat{\sigma}^z \rangle)$ . Therefore, the input values 0 and 1 refer to the input states as the ground state  $|0\rangle$  ( $\langle \hat{\sigma}^z \rangle = -1$ ) and the excited state  $|1\rangle$  ( $\langle \hat{\sigma}^z \rangle = 1$ ), respectively. Such a quantum perceptron without hidden layers and with only two-qubit interactions is equivalent to a classical perceptron, from the point of view of the following training process. Aiming at training it with the gradient descent method, one has to update the weights and bias as



**Figure 1.** In order to (a) encode an XOR gate and to (b) search prime numbers among the integers from 0 to 7, we show the value of the cost function at different epochs, using a multi-qubit interaction perceptron (solid blue) and a classical perceptron (dotted red). The latter is equivalent to a quantum perceptron with only two-qubit terms, which is shown from the comparison of the weights and bias during the training process, see Eqs. (7)–(9), where the learning rate is  $\eta = 1.5$ . The transfer function in Eq. (2) is used for both cases. In the quantum perceptron, this function indicates that the system evolves adiabatically.

$$\begin{aligned}\tilde{w}_i &= w_i - \frac{\eta}{N} \sum_{n=1}^N (y^{(n)} - t^{(n)}) \frac{\partial y^{(n)}}{\partial x} \frac{\partial x}{\partial w_i}, \\ \tilde{b} &= b - \frac{\eta}{N} \sum_{i=1}^N (y^{(n)} - t^{(n)}) \frac{\partial y^{(n)}}{\partial x},\end{aligned}\quad (8)$$

where

$$\begin{aligned}\frac{\partial y^{(n)}}{\partial x} &= \frac{1}{2} \left( \left\langle \frac{\partial \Psi}{\partial x} \middle| \hat{\sigma}_z \middle| \Psi \right\rangle + \left\langle \Psi \middle| \hat{\sigma}_z \middle| \frac{\partial \Psi}{\partial x} \right\rangle \right) \\ &= f'(x)\end{aligned}\quad (9)$$

and  $|\Psi\rangle = \hat{U}(\hat{x}, f)|0\rangle$  is the solution to the Schrödinger equation driven by the Hamiltonian (3) with  $\hat{x} = w_1 \hat{\sigma}_1^z + w_2 \hat{\sigma}_2^z + b$ . In the previous equations we can see that weights and bias are obtained in the same way as their classical counterparts. For that, one has to compare Eq. (7) with Eqs. (8) and (9) provided that  $\frac{\partial x}{\partial w_i} = \hat{\sigma}_i^z \rightarrow s_i$ . This indicates that a single quantum perceptron with two-qubit interactions and the basis states  $|00\rangle, |01\rangle, |10\rangle, |11\rangle$  as input is equivalent to a classical perceptron.

In order to implement the XOR gate with this quantum perceptron, one would need to perform two adiabatic passages with different controls  $\Omega(t)$ , as well as to use different neural potentials in each passage by appropriately changing the weights and biases<sup>44</sup>. Equivalently, one could also do the XOR gate by including one hidden layer with two additional quantum neurons and the application of a single  $\Omega(t)$ . It is worth mentioning that Eq. (9) holds only for a quantum perceptron instead of a QNN with hidden layers. If a QNN has more layers, one only needs to measure the output qubit value

$$y = P(x_{\text{out}}) = \frac{1}{2} (1 + \langle \Psi | \hat{\sigma}_{\text{out}}^z | \Psi \rangle) \quad (10)$$

instead of the intermediate ones, i.e.,  $|\Psi\rangle = \hat{U}_{\text{tot}}|0\rangle$  with  $\hat{U}_{\text{tot}} = \prod_{j=1}^M \hat{U}_j$  where  $M$  is the total number of quantum perceptrons in a QNN. Otherwise, shot noise is introduced by measuring the neurons in the hidden layer.

This procedure to do the XOR gate can be simplified when considering a quantum perceptron with multi-qubit interactions. We can find the output of the quantum perceptron by using Eq. (10), where  $|\Psi\rangle = \hat{U}(\hat{x}; f)|0\rangle$ . In the unitary transformation implemented by the quantum perceptron gate in the Heisenberg picture

$$\hat{U}^\dagger \hat{\sigma}^z \hat{U} = [1 - 2f(\hat{x})] \hat{\sigma}^z + 2\sqrt{f(\hat{x})[1 - f(\hat{x})]} \hat{\sigma}^x, \quad (11)$$

leads to  $y = P(x) = f(x)$ , where  $y = 1$ , if  $x > 1/2$ ; and  $y = 0$ , if  $x \leq 1/2$ . In particular, we explore the following multi-qubit potential

$$\hat{x} = w_1 \hat{\sigma}_1^z + w_2 \hat{\sigma}_2^z + b + w_m \hat{\sigma}_1^z \hat{\sigma}_2^z. \quad (12)$$

In this case, the weights  $w_1$ ,  $w_2$  and  $b$  are updated as in Eq. (8), while the updating formula for the weight of the  $w_m \hat{\sigma}_1^z \hat{\sigma}_2^z$  term is

$$\tilde{w}_m = w_m - \frac{\eta}{N} \sum_{n=1}^N (y^{(n)} - t^{(n)}) \frac{\partial y^{(n)}}{\partial x} \hat{\sigma}_1^z \hat{\sigma}_2^z. \quad (13)$$

Considering the construction of an XOR gate from the equations

$$0 \times w_1 + 0 \times w_2 + b \leq \frac{1}{2} \Leftrightarrow b - \frac{1}{2} \leq 0, \quad (14)$$

$$0 \times w_1 + 1 \times w_2 + b > \frac{1}{2} \Leftrightarrow b - \frac{1}{2} > -w_2, \quad (15)$$

$$1 \times w_1 + 0 \times w_2 + b > \frac{1}{2} \Leftrightarrow b - \frac{1}{2} > -w_1, \quad (16)$$

$$1 \times w_1 + 1 \times w_2 + b + w_m \leq \frac{1}{2} \Leftrightarrow b - \frac{1}{2} + w_m \leq -w_1 - w_2, \quad (17)$$

we find that Eqs. (15)–(17) are contradictory, if the multi-qubit interaction term does not exist. However, one can always find appropriate values for  $w_m$  to satisfy the above inequalities. The existence of  $w_m \hat{\sigma}_1^z \hat{\sigma}_2^z$  in the neural potential enables the quantum perceptron to construct an XOR gate as a nonlinear separator. We test the cost function value (see, Eq. (6)) for our quantum perceptron and find that  $C < 1\%$  occurs at epoch = 197 as shown in Fig. 1a (solid-blue line with squares). From the numerical calculation, we can see that the cost function value continues to decrease. On the contrary the classical and qubit-qubit interaction perceptron are not able to produce an XOR gate, see plateau behavior of the cost function (dotted-red) that remains constant after the epoch ~ 50.

**Searching prime numbers.** In Ref.<sup>44</sup>, a specific example of two, three, and four perceptrons per layer was illustrated to detect prime numbers from 0 to  $2^i - 1$  where  $i$  ranges from  $i = 3$  to  $i = 7$  bits. In particular, it was shown that a QNN with two perceptrons –one in the hidden layer and the other one as the output– can classify prime numbers from 0 to 7 (3 bits). This exemplifies the better performance of QNNs compared with classical ones where a hidden layer with two neurons are necessary to accomplish the same task. A scheme of these networks with 3 bits in input is shown figure in Fig. 2a,b, while details regarding QNN-training to search prime numbers are in “Methods”.

Now, we introduce the multi-qubit term into the neural potential and find that the same task is achieved by a QNN without hidden layers, i.e. at the single quantum perceptron level. To search a prime number for 3 bits (Truth table listed in Fig. 2c) using a single quantum perceptron we consider the potential

$$\hat{x} = w_1\hat{\sigma}_1^z + w_2\hat{\sigma}_2^z + w_3\hat{\sigma}_3^z + b + w_m\hat{\sigma}_2^z\hat{\sigma}_3^z. \tag{18}$$

As we demonstrate later, adding the multi-qubit term  $w_m\hat{\sigma}_2^z\hat{\sigma}_3^z$  is enough to fulfill the prime numbers searching task. In this respect, one can include additional multi-qubit terms in the neural potential. However, we use the simplest example in Eq. (18) to simplify the network and training process.

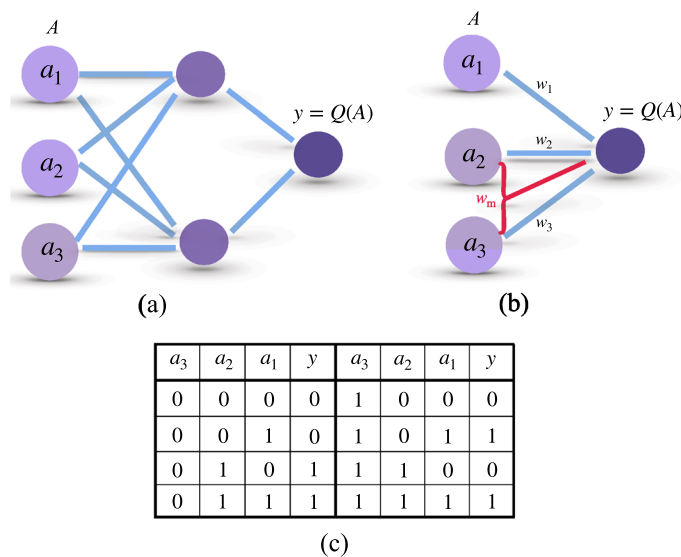
We note that when the input of a quantum perceptron that presents only two-qubit interactions are the basis states, this is equivalent to the classical perceptron as shown in Eqs. (7)–(9). For this reason, now we compare the cost function of a classical perceptron with the cost function of a quantum perceptron using the potential in Eq. (18) for the specific purpose of searching prime numbers from 0 to 7. As it is shown in Fig. 1b the quantum perceptron achieves  $C < 1\%$  at epoch = 667 (solid-blue with squares), while the  $C$  value saturates to 0.126 in the classical counterpart (dotted-red with diamonds). This indicates that our perceptron achieves the searching task without hidden layers. We have also verified the ability of the quantum perceptron by minimizing the cost function to an acceptable error for searching prime numbers for 4 bits with the potential

$$\hat{x} = w_1\hat{\sigma}_1^z + w_2\hat{\sigma}_2^z + w_3\hat{\sigma}_3^z + w_4\hat{\sigma}_4^z + b + w_m\hat{\sigma}_2^z\hat{\sigma}_3^z, \tag{19}$$

and for 5 bits with

$$\hat{x} = w_1\hat{\sigma}_1^z + w_2\hat{\sigma}_2^z + w_3\hat{\sigma}_3^z + w_4\hat{\sigma}_4^z + w_5\hat{\sigma}_5^z + b + w_m\hat{\sigma}_2^z\hat{\sigma}_3^z. \tag{20}$$

The multi-qubit terms in the above two neural potentials for 4 and 5 bits are chosen due to the fact that the cost function result in  $C < 1\%$  at epoch = 1692 for Eq. (19) and epoch = 1698 for Eq. (20) with the consideration of adopting the minimal number of multi-qubit terms. Further training the above quantum perceptrons presents that the value of  $C$  continue to decrease, proving the success to achieve these tasks.



**Figure 2.** (a) Schematic configuration of a classical neural network with one hidden layer of two neurons for the task to search prime numbers of the input number  $A = (a_1, a_2, a_3) \in \{0, 1, \dots, 7\}$  with 3 bits. After well trained, the network gives the output  $y = Q(A)$ . (b) The same task can be achieved by a QNN which has multi-qubit interactions in the neural potential (red junction involving second and third neurons) without hidden layers. (c) Truth table for prime-number search with 3 input bits for a QNN. The input values  $A = (a_1, a_2, a_3)$  are the binary numbers for the integers from 0 to 7. The output value  $y = Q(A)$  is 1 for prime numbers and 0 for non-prime ones.

**Quantum gates.** Now we show that one can use a QNN with multi-qubit interactions to construct quantum gates such as CNOT, Toffoli, and Fredkin gates without the necessity of hidden layers. In comparison, one can demonstrate that a QNN with only two-qubit interactions cannot construct the above mentioned quantum gates in the same conditions. This will be shown later. As quantum gates are reversible, the number of input and output qubits of quantum gates should be equal. Hence, the number of the neurons in the input equals that in the output. Correspondingly, the cost function is changed into

$$C = \frac{1}{2Nk} \sum_{n=1}^N \sum_{i=1}^k (y_i^{(n)} - t_i^{(n)})^2, \tag{21}$$

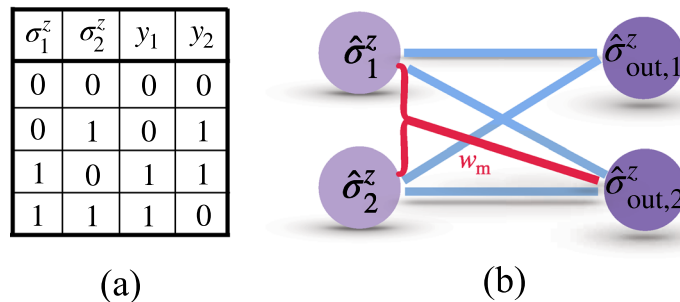
where  $k$  is the number of quantum perceptrons.

For CNOT gate where  $k = 2$ , the truth table and the schematic configuration of the QNN is shown in Fig. 3a,b respectively, where each perceptron (output neuron) gives the output value  $y_i = \frac{1}{2}(1 + \langle \hat{\sigma}_{out,i}^z \rangle)$ . The first perceptron should have the same value as the first input neuron, while the second one aims to achieve the same value as the XOR output of two input neurons, i.e.  $\hat{\sigma}_{out,2}^z = \hat{\sigma}_1^z \oplus \hat{\sigma}_2^z$ . The neural potential of the second perceptron is

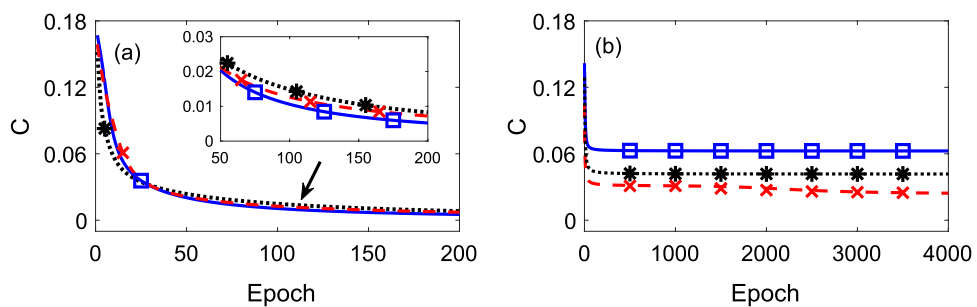
$$\hat{x}_2 = w_{21}\hat{\sigma}_1^z + w_{22}\hat{\sigma}_2^z + b_2 + w_m\hat{\sigma}_1^z\hat{\sigma}_2^z, \tag{22}$$

and the first perceptron has two-qubit interaction terms. We find that this is the most concise choice, as it requires least connectivity. We obtain the cost function value (using Eq. (21))  $C < 1\%$  at epoch = 116 and  $C \rightarrow 0$  at a larger epoch, as shown in Fig. 4a (solid-blue with squares) indicating the satisfaction to train a CNOT gate well. On the contrary, training such a CNOT gate by a QNN with only two-qubit interaction terms leads to a  $C$  tending to 0.0625 at large epoch, see Fig. 4b. This indicates that a standard QNN cannot encode the CNOT without hidden layers.

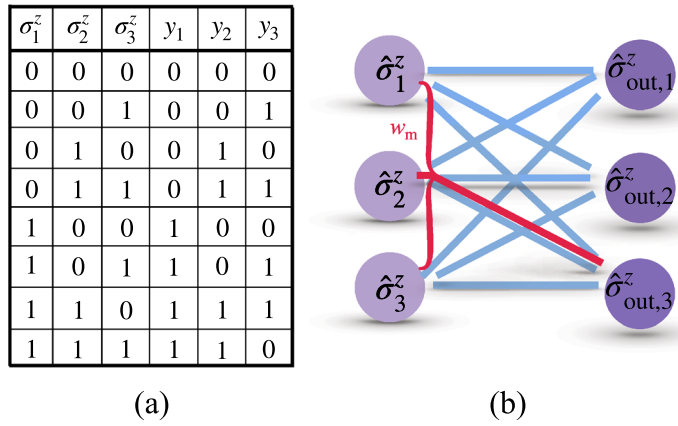
A QNN with multi-qubit terms also works for constructing a Toffoli gate without hidden layers. Truth table, and QNN for Toffoli are shown in Fig. 5a,b. Known as Controlled-Controlled-Not gate, a Toffoli gate has 3-qubit inputs and outputs. In the QNN, the outputs of the first two qubits should be the same value as their inputs  $\hat{\sigma}_{out,i}^z = \hat{\sigma}_i^z$  ( $i = 1, 2$ ), while the output of the third qubit aims at  $\hat{\sigma}_{out,3}^z = \hat{\sigma}_3^z \oplus \hat{\sigma}_1^z\hat{\sigma}_2^z$ . In this case, we propose the following neural potential of the third perceptron



**Figure 3.** Truth table of a CNOT gate (a) constructed by a QNN illustrated in a schematic configuration (b) with the multi-qubit interaction term  $w_m\hat{\sigma}_1^z\hat{\sigma}_2^z$ .



**Figure 4.** Cost function at different epoch for (a) QNN with the multi-qubit term and (b) QNN with only two-qubit interactions. In particular the solid-blue line is the cost function obtained to construct a CNOT gate, a Toffoli gate (dashed-red line) and a Fredkin gate (dotted-black line). Illustrated in the inset of (a), the cost function values reach  $C < 1\%$  at 116th, 140th, 162th epoch for the above three quantum gates, respectively, indicating that these gates can be constructed by QNNs with multi-qubit interaction without hidden layers. In contrast,  $C$  values go to their respective plateau by quantum perceptrons with two-qubit interactions.



**Figure 5.** Truth table of a Toffoli gate (a) constructed by a QNN illustrated in a schematic configuration (b) with the multi-qubit interaction term  $w_m \hat{\sigma}_1^z \hat{\sigma}_2^z \hat{\sigma}_3^z$ .

$$\hat{x}_3 = w_{31} \hat{\sigma}_1^z + w_{32} \hat{\sigma}_2^z + w_{33} \hat{\sigma}_3^z + b_3 + w_m \hat{\sigma}_1^z \hat{\sigma}_2^z \hat{\sigma}_3^z, \tag{23}$$

with a minimal number of multi-qubit terms. Demonstrated in Fig. 4a,b, the value  $C < 1\%$  (dashed-red line with crosses) occurs at epoch = 140 and goes to 0 at large epoch during the training of a Toffoli gate. In close similarity with the previous case, a standard QNN without hidden layers fails to achieve the Toffoli gate, as  $C$  saturates at 0.0238.

Another example that can be successfully constructed by our QNNs including multi-qubit interactions without hidden layers is the Fredkin gate (also known as Controlled-SWAP gate) whose truth table and schematic configuration are in Fig. 6a,b. In this case, the adopted potential of the third perceptron is

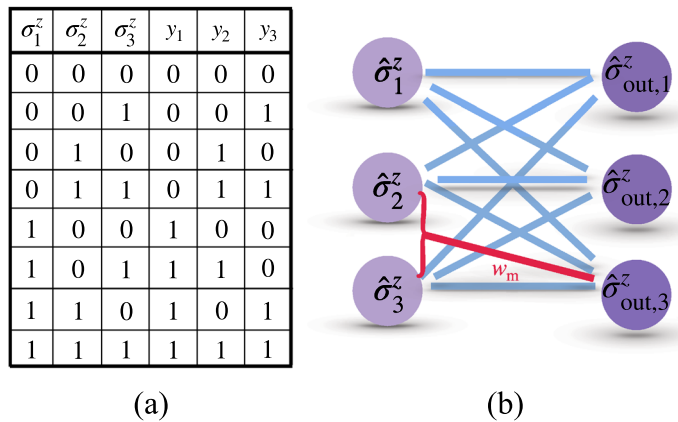
$$\hat{x}_3 = w_{31} \hat{\sigma}_1^z + w_{32} \hat{\sigma}_2^z + w_{33} \hat{\sigma}_3^z \tag{24}$$

$$+ b_3 + w_m \hat{\sigma}_2^z \hat{\sigma}_3^z, \tag{25}$$

According to the numerical calculation, the cost function value  $C$  tends to zero, shown in Fig. 4a (dotted-black line with asterisks), indicating that these gates can be constructed by quantum perceptrons without hidden layers. Again,  $C$  saturates at the value 0.0417 at large epoch for a standard QNN without hidden layers, see Fig. 4b.

### Discussions

We have shown different applications of QNNs which possess multi-qubit interactions. Meanwhile, the performance is enhanced compared to the same topology of a QNN without multi-qubit interactions. The multi-qubit interaction terms induce connectivities among quantum perceptrons that deviate from the current network paradigm of additive activations. It is due to the multi-qubit terms in the potentials that one can avoid the



**Figure 6.** Truth table of a Fredkin gate (a) constructed by a QNN illustrated in a schematic configuration (b) with the multi-qubit interaction term  $w_m \hat{\sigma}_2^z \hat{\sigma}_3^z$ .

presence of some hidden layer without sacrificing approximative power. Such architecture allows us to address the connectivity challenge in scaling up QNNs. Moreover, the simple configuration helps to control the efficiency of the training processes. During the training process of all the examples shown above, the activation function (Eq. 2) based on the adiabatic evolution of the system is used. Instead, one may use *shortcuts to adiabaticity* to accelerate the formation of the activation function in physical registers<sup>45</sup>.

The Hamiltonian of the  $j$ th perceptron given by Eq. (3) where the neural potential expressed by Eq. (5) corresponds to the linear Ising Hamiltonian with higher order interactions. Such a model is present in distinct quantum platforms<sup>53–58</sup> experimentally and theoretically, although the developments in different platforms vary. An optical Ising machine hosting adjustable four-body interaction with all-to-all connections over a large number of spins was experimentally demonstrated in Ref.<sup>54</sup>. Moreover, the four-body interactions are designed to be realized via superconducting quantum interference devices (SQUIDS)<sup>55</sup>, while a single shot method for executing an *i*-Toffoli gate which is a three-qubit gate with two control and one target qubits was proposed in Ref.<sup>56</sup>, with the application of currently existing superconducting hardware. Using resonant microwave-mediated interactions between distant electron spins to implement multi-qubit potentials<sup>57</sup> marks an important milestone for all-to-all qubit connectivity and scalability in silicon-based quantum circuits. Quantum evolutions governed by terms involving up to six-qubit interactions has already been implemented in trapped-ion systems<sup>58</sup> where the value of qubit-qubit interactions can be tuned, allowing the same architecture to be used to implement different types of gates and leading to a variety of quantum gates in the system. Each of the examples considered above corresponds to a particular case of multi-qubit interaction and its physical implementation would depend on the specific considered platform. The development in the quantum hardware highlights the potential for practical implementations of multi-qubit potentials in QNNs. The promotion of quantum hardware and QNN protocols in respective fields will boost mutual developments.

## Methods

**Hamiltonian and training of the neural networks.** In the article, we propose a QNN with multi-qubit interactions where the Hamiltonian Eq. (3) of one perceptron has the neural potential expressed in Eq. (5). In preparation for a QNN to demonstrate advantages in comparison with classical counterparts, we need to develop concepts to improve scalability of QNNs where controlling the network depth becomes crucial. One main objective to develop a QNN is to therefore minimize the depth without sacrificing approximative power. To this end, we explore deviations from the current network paradigm of additive activations and include multi-qubit interactions in the neural potential leading to a reduction of the network depth. In all the calculations, we use the standard gradient descent to train the neural networks with the cost function Eq. (21). Correspondingly, the weight of multi-qubit interactions term  $w_m \hat{\sigma}_1^z \cdots \hat{\sigma}_n^z$  in the neural potential Eq. (5) should be updated as

$$\tilde{w}_m = w_m - \frac{\eta}{N} \sum_{n=1}^N \left( y^{(n)} - t^{(n)} \right) \frac{\partial y^{(n)}}{\partial x} \hat{\sigma}_1^z \cdots \hat{\sigma}_n^z. \quad (26)$$

**Training for prime numbers search.** To train a QNN for the task of searching prime numbers, a set of  $2^i$  pairs for  $i$  bits containing the input and output values  $\{A^{(n)}, y^{(n)}\}_{n=1}^{2^i}$  is taken, where the inputs are binary numbers  $A^{(n)} = (a_1, a_2, \dots, a_i)$  corresponding to the integers belonging to the set  $\in \{0, 1, \dots, 2^i - 1\}$ . As the targets are  $t \in \{0, 1\}$ , the output of the network  $y^n = Q(A^{(n)}) = 1$ , if and only if the input number is prime.

## Data availability

The datasets used and analysed during the current study available from the corresponding author on reasonable request.

Received: 11 November 2022; Accepted: 25 May 2023

Published online: 05 June 2023

## References

- Castells, M. *The Information Age: Economy, Society and Culture* (Blackwell, Oxford, 1996).
- Walter, C. Kryder's Law. *Sci. Am.* **293**, 32–33. <https://doi.org/10.1038/scientificamerican0805-32> (2005).
- Kolmogorov, A. On Tables of Random Numbers. *Sankhya Ser. A.* **25**, 369–376 (1963).
- Samuel, A. L. Some studies in machine learning using the game of Checkers. *IBM J. Res. Dev.* **3**, 210–229. <https://doi.org/10.1147/rd.33.0210> (1959).
- Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488. <https://doi.org/10.1007/BF02650179> (1982).
- Hebb, D. *The organization of behavior* (Wiley, New York, 1949).
- McCulloch, W. S. & Pitts, W. A. Logical calculus of ideas immanent in nervous activity. *Bull. Math. Biol.* **5**, 115–133. <https://doi.org/10.1007/BF02478259> (1943).
- Oh, K. S. & Jung, K. GPU implementation of neural networks. *Pattern Recogn.* **37**, 1311–1314. <https://doi.org/10.1016/j.patcog.2004.01.013> (2004).
- LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444. <https://doi.org/10.1038/nature14539> (2015).
- Kleene, S. C. Representation of events in nerve nets and finite automata. *Ann. Math. Stud.* **34** (Princeton University Press, 1956). <https://doi.org/10.1515/9781400882618-002>
- Rosenblatt, F. The perceptron—A perceiving and recognizing automaton. Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory, (1957).
- Hopfield, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci.* **81**, 3088–3092. <https://doi.org/10.1073/pnas.81.10.3088> (1984).
- Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **2**, 303–314. <https://doi.org/10.1007/BF02551274> (1989).



14. Dahl, G. E., Yu, D., Deng, L. & Acero, A. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Trans. Audio Speech Lang. Process.* **20**, 30–42. <https://doi.org/10.1109/TASL.2011.2134090> (2012).
15. Hinton, G. E., Osindero, S. & Teh, Y. A. Fast learning algorithm for deep belief nets. *Neural Comput.* **18**, 1527–1554. <https://doi.org/10.1162/neco.2006.18.7.1527> (2006).
16. Dada, E. G. *et al.* Machine learning for email spam filtering: Review, approaches and open research problems. *Heliyon* **5**, e01802. <https://doi.org/10.1016/j.heliyon.2019.e01802> (2019).
17. Buehler, M., Iagnemma, K. & Singh, S. The DARPA urbanchallenge: autonomous vehicles in city traffic (Springer U.S., 2009).
18. Devi, S., Malarvezhi, P., Dayana, R. & Vadivukkarasi, K. A comprehensive survey on autonomous driving cars: A perspective view. *Wirel. Pers. Commun.* **114**, 2121–2133. <https://doi.org/10.1007/s11277-020-07468-y> (2020).
19. Valsamis, A., Tserpes, K., Zissis, D., Anagnostopoulos, D. & Varvarigou, T. Employing traditional machine learning algorithms for big data streams analysis: The case of object trajectory prediction. *J. Syst. Softw.* **127**, 249–257 (2017). <https://doi.org/10.1016/j.jss.2016.06.016>
20. Kashyap, P. *Machine learning for decision makers: Cognitive computing fundamentals for better decision making* (APress, U. S., 2018).
21. Fürnkranz, J. *Machine learning and game playing, encyclopedia of machine learning* (Springer U.S., 2010).
22. Huang, B., Huan, Y., Xu, L. D., Zheng, L. & Zou, Z. Automated trading systems statistical and machine learning methods and hardware implementation: A survey. *Enterp. Inf. Syst.* **13**, 132–144. <https://doi.org/10.1080/17517575.2018.1493145> (2019).
23. Kleene, S. C.  $\lambda$ -definability and recursiveness. *Duke Math. J.* **2**, 340–353. <https://doi.org/10.1215/S0012-7094-36-00227-2> (1936).
24. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510. <https://doi.org/10.1038/s41586-019-1666-5> (2019).
25. Kitaev, A. Y. Quantum computations: Algorithms and error correction. *Russ. Math. Surv.* **52**, 1191–1249. <https://doi.org/10.1070/RM1997v052n06ABEH002155> (1997).
26. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502. <https://doi.org/10.1103/PhysRevLett.103.150502> (2009).
27. Peruzzo, A. *et al.* A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213. <https://doi.org/10.1038/ncomms5213> (2014).
28. E. Farhi, J. Goldstone, & S. A. Gutmann, quantum approximate optimization algorithm. [arXiv:1411.4028](https://arxiv.org/abs/1411.4028) (2014).
29. Grover, L. K. A fast quantum mechanical algorithm for database search. *Proceedings, 28th Annual ACM Symposium on the Theory of Computing*, 212–219 (1996). <https://doi.org/10.1145/237814.237866>
30. Shor, P. W. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134 (1994). <https://doi.org/10.1109/SECS.1994.365700>
31. Lloyd, S. Universal quantum simulators. *Science* **273**, 1073. <https://doi.org/10.1126/science.273.5278.1073> (1996).
32. Nielsen, M. A., & Chuang, I. *Quantum computation and quantum information* (Cambridge University Press, New York, U.S., 2011).
33. Möttönen, M., Vartiainen, J. J., Bergholm, V. & Salomaa, M. M. Quantum circuits for general multiqubit gates. *Phys. Rev. Lett.* **93**, 130502. <https://doi.org/10.1103/PhysRevLett.93.130502> (2004).
34. Reagor, M., Osborn, C. B., Tezak, N., *et al.* Demonstration of universal parametric entangling gates on a multi-qubit lattice. *Sci. Adv.* **4**, eaao3603 (2018). <https://doi.org/10.1126/sciadv.aao3603>
35. Bækkegaard, T. *et al.* Realization of efficient quantum gates with a superconducting qubit-qutrit circuit. *Sci. Rep.* **9**, 13389. <https://doi.org/10.1038/s41598-019-49657-1> (2019).
36. Kak, S. C. Quantum neural computing. *Adv. Imag. Elect. Phys.* **94**, 259–313. [https://doi.org/10.1016/S1076-5670\(08\)70147-2](https://doi.org/10.1016/S1076-5670(08)70147-2) (1995).
37. Schuld, M., Sinaysky, I. & Petruccione, F. The quest for a Quantum Neural Network. *Quan. Inf. Proc.* **13**, 2567–2586. <https://doi.org/10.1007/s11128-014-0809-8> (2014).
38. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202. <https://doi.org/10.1038/nature23474> (2017).
39. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. [arXiv:1802.06002](https://arxiv.org/abs/1802.06002) (2018).
40. Schuld, M. & Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **122**, 040504. <https://doi.org/10.1103/PhysRevLett.122.040504> (2019).
41. Schuld, M., Bocharov, A., Svore, K. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308. <https://doi.org/10.1103/PhysRevA.101.032308> (2020).
42. Cao, Y., Guerreschi, G., & Aspuru-Guzik, A. Quantum neuron: An elementary building block for machine learning on quantum computers. [arXiv:1711.11240](https://arxiv.org/abs/1711.11240) (2017).
43. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **4**, 226 (2020). <https://doi.org/10.22331/q-2020-02-06-226>
44. Torrontegui, E. & García-Ripoll, J. J. Unitary quantum perceptron as efficient universal approximator. *EPL* **125**, 30004. <https://doi.org/10.1209/0295-5075/125/30004> (2019).
45. Ban, Y., Chen, X., Torrontegui, E., Solano, E., & Casanova, J. Speeding up quantum perceptron via shortcuts to adiabaticity. *Sci. Rep.* **11**, 5783 (2021). <https://doi.org/10.1038/s41598-021-85208-3>
46. Carrasquilla, J. & Melko, R. G. Machine learning phases of matter. *Nat. Phys.* **13**, 431–434. <https://doi.org/10.1038/nphys4035> (2017).
47. Deng, D.-L., Li, X., & Das Sarma, S. Quantum entanglement in neural network states. *Phys. Rev. X* **7**, 021021 (2017). <https://doi.org/10.1103/PhysRevX.7.021021>
48. Torlai, G. *et al.* Neural-network quantum state tomography. *Nat. Phys.* **14**, 447–450. <https://doi.org/10.1038/s41567-018-0048-5> (2018).
49. Aharon, N. *et al.* NV center based nano-NMR enhanced by deep learning. *Sci. Rep.* **9**, 17802. <https://doi.org/10.1038/s41598-019-54119-9> (2019).
50. Ban, Y., Echanobe, J., Ding, Y., Puebla, R. & Casanova, J. Neural-network-based parameter estimation for quantum detection. *Quant. Sci. Technol.* **6**, 045012. <https://doi.org/10.1088/2058-9565/ac16ed> (2021).
51. Minsky, M. & Papert, S. A. *Perceptrons: An introduction to computational geometry* (MIT Press, Cambridge, MA, USA, 2017).
52. Nielsen, M. A., & Chuang, I. L. *Quantum computation and quantum information* (Cambridge university press, 2010).
53. Donskaya, I. S. Higher-order interactions in the linear rising model. *Theor. Math. Phys.* **74**, 324–328. <https://doi.org/10.1007/BF01016629> (1988).
54. Kumar, S., Zhang, H. & Huang, Y. P. Large-scale Ising emulation with four body interaction and all-to-all connections. *Commun. Phys.* **3**, 108. <https://doi.org/10.1038/s42005-020-0376-5> (2020).
55. Sameti, M., Potočník, A., Browne, D. E., Wallraff, A. & Hartmann, M. J. Superconducting quantum simulator for topological order and the toric code. *Phys. Rev. A* **95**, 042330. <https://doi.org/10.1103/PhysRevA.95.042330> (2017).
56. Baker, A. J. *et al.* Single shot i-Toffoli gate in dispersively coupled superconducting qubits editors-pick. *Appl. Phys. Lett.* **120**, 054002. <https://doi.org/10.1063/5.0077443> (2022).
57. Borjans, F., Croot, X. G., Mi, X., Gullans, M. J. & Petta, J. R. Resonant microwave-mediated interactions between distant electron spins. *Nature* **577**, 195–198. <https://doi.org/10.1038/s41586-019-1867-y> (2020).
58. Lanyon, B. P., Hempel, C., Nigg, D., Müller, M., Gerritsma, R. Z., Fähringer, Schindler, P., Barreiro, J. T., Rambach, M., Kirchmair, G., Hennrich, M., Zoller, P., Blatt, R., & Roos, C. F. Universal digital quantum simulation with trapped ions. *Science* **334**, 57–61 (2011). <https://doi.org/10.1126/science.1208001>

## Acknowledgements

We acknowledge financial support from Spanish Government via PGC2018-095113-B-I00 (MCIU/AEI/FEDER, UE), Basque Government via IT986-16, as well as from QMiCS (820505) and OpenSuperQ (820363) of the EU Flagship on Quantum Technologies, and the EU FET Open Grant Quomorphic (828826). Y. B. acknowledges the CDTI within the Misiones 2021 program and the Ministry of Science and Innovation under the Recovery, Transformation and Resilience Plan—Next Generation EU under the project “CUCO: Quantum Computing and its Application to Strategic Industries”. E. T. acknowledges the Ramón y Cajal (RYC2020-030060-I) research fellowship, the Spanish Government via the project PID2021-126694NA-C22 (MCIU/AEI/FEDER, EU), CSIC Research Platform PTI-001, CAM/FEDER Project No. S2018/TCS-4342 (QUITEMAD-CM), and by Comunidad de Madrid-EPUC3M14. J. C. acknowledges the Ramón y Cajal (RYC2018-025197-I) research fellowship, the financial support from Spanish Government via EUR2020-112117 and Nanoscale NMR and complex systems (PID2021-126694NB-C21) projects, the ELKARTEK project Dispositivos en Tecnologías Cuánticas (KK-2022/00062), and the Basque Government grant IT1470-22.

## Author contributions

Y.B. developed the theoretical formalism, performed the analytic calculations and the numerical simulations. E.T. verified the method. J.C. supervised the project. All the authors contributed to the final version of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to Y.B.

**Reprints and permissions information** is available at [www.nature.com/reprints](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