

## ARTICLE OPEN



## An artificial spiking quantum neuron

Lasse Bjørn Kristensen<sup>1</sup>✉, Matthias Degroote<sup>2,3,4</sup>, Peter Wittek<sup>5,6,7,8</sup>, Alán Aspuru-Guzik<sup>2,3,4,7,9,10</sup> and Nikolaj T. Zinner<sup>1,11</sup>

Artificial spiking neural networks have found applications in areas where the temporal nature of activation offers an advantage, such as time series prediction and signal processing. To improve their efficiency, spiking architectures often run on custom-designed neuromorphic hardware, but, despite their attractive properties, these implementations have been limited to digital systems. We describe an artificial quantum spiking neuron that relies on the dynamical evolution of two easy to implement Hamiltonians and subsequent local measurements. The architecture allows exploiting complex amplitudes and back-action from measurements to influence the input. This approach to learning protocols is advantageous in the case where the input and output of the system are both quantum states. We demonstrate this through the classification of Bell pairs which can be seen as a certification protocol. Stacking the introduced elementary building blocks into larger networks combines the spatiotemporal features of a spiking neural network with the non-local quantum correlations across the graph.

npj Quantum Information (2021)7:59; <https://doi.org/10.1038/s41534-021-00381-7>

## INTRODUCTION

As Moore's law slows down<sup>1</sup>, increased attention has been put towards alternative models for solving computationally hard problems and analyzing the ever growing stream of data<sup>2,3</sup>. One significant example has been the reinvigoration of the field of machine learning: neuromorphic models, inspired by biology, found applications in a large host of fields<sup>4,5</sup>. In parallel, quantum computing has been taking significant steps moving from a scientific curiosity towards a practical technology capable of solving real-world problems<sup>6</sup>. Given the prominence of both fields, it is not surprising that a lot of work has gone into exploring their parallels, and how one may be used to enhance the other. One such synergy has emerged in the field of quantum machine learning<sup>7–9</sup>. Recent results aim to mimic the parametric, teachable structure of a neural network with a sequence of gates on a set of qubits<sup>10–12</sup> or on a set of photonic modes<sup>13</sup>. A subset of these algorithms focus on quintessentially quantum problems: the input to the learning model is a quantum state and so is its output. This scenario is relevant in building and scaling experimental devices and it is often referred to as quantum learning<sup>14,15</sup>.

We take a slightly different approach to quantum learning wherein the structure of the network manifests itself as interactions between qubits in space rather than as gates in a circuit diagram. Specifically, we will present a small toolbox of simple spin models that can be combined into larger networks capable of neuromorphic quantum computation. To illustrate the power of such networks, a small example of such a 'spiking quantum neural network' capable of comparing two Bell states is presented, a task which could have applications in both state preparation and quantum communication. The term 'spiking' refers to the temporal aspect in the functioning of the model during the activation of the neuron, akin to the classical spiking neural networks<sup>16</sup>. As illustrated in the example, a fundamental property of these networks is that they generate entanglement between the inputs and outputs of the network, thus allowing

measurement back-action from standard measurements on the output to influence the state of the input in highly non-trivial ways. The proposed model for spiking quantum neural networks is amenable to implementation in a variety of physical systems, e.g., using superconducting qubits<sup>17–19</sup>.

A key inspiration for the constructions presented in this paper is the advent of dedicated classical hardware for simulating spiking classical neural networks, including implementations from both Intel and IBM<sup>20–23</sup>. These systems emulate the function of a biological spiking neural network through networks of small neuron-like computing resources. They can be roughly grouped into digital systems simulating the dynamics of spiking networks using binary variables and in discrete time-steps<sup>21,22</sup>, and analog circuitry emulating spiking behavior of physical observables in continuous time<sup>23</sup>. The main focus of this paper is to extend the concept of the discretized models to the quantum domain, facilitating the use of purpose-built neuromorphic systems for applications within quantum learning, and providing the models access to quantum resources like entanglement. Additionally, an outlook towards the implementation of continuous-time computational quantum dynamics is also briefly discussed.

Note that several similar proposals for real-space quantum neural networks do exist. One prominent example is within the field of quantum memristors, where a spiking quantum neuron was recently constructed, and networks of these objects were proposed<sup>24</sup>. The central distinction to our scheme is the degrees of freedom under consideration. Whereas our scheme is based on generic qubits, memristive schemes revolve around the dynamics of voltages and currents. This means that the operations of our neural networks below will be more easily interpretable in a quantum computing context in comparison to these dissipative memristive schemes. A more apt comparison may therefore be a recent proposal for implementing a quantum perceptron through adiabatic evolution of an Ising model<sup>25</sup>. Indeed, the interaction implemented in that proposal very closely resembles the

<sup>1</sup>Department of Physics and Astronomy, Aarhus University, DK-8000 Aarhus C, Denmark. <sup>2</sup>Department of Chemistry, University of Toronto, Toronto, ON M5G 1Z8, Canada.

<sup>3</sup>Department of Computer Science, University of Toronto, Toronto, ON M5G 1Z8, Canada. <sup>4</sup>Department of Chemistry and Chemical Biology, Harvard University, Cambridge, MA 02138, USA. <sup>5</sup>Rotman School of Management, University of Toronto, Toronto, ON, Canada. <sup>6</sup>Creative Destruction Lab, Toronto, ON, Canada. <sup>7</sup>Vector Institute for Artificial Intelligence, Toronto, ON M5S 1M1, Canada. <sup>8</sup>Perimeter Institute for Theoretical Physics, Toronto, ON N2L 2Y5, Canada. <sup>9</sup>Zapata Computing Inc., Cambridge, MA 02139, USA.

<sup>10</sup>Canadian Institute for Advanced Research, Toronto, ON M5G 1Z8, Canada. <sup>11</sup>Aarhus Institute of Advanced Study, Aarhus University, DK-8000 Aarhus C, Denmark.

✉email: lbk@phys.au.dk

operation of the first spin-model presented below. However, the nature of the adiabatic protocol makes the path towards connecting such building blocks into a larger dynamical network unclear. Neither case fully captures the properties of the spiking quantum neural network proposed here.

## RESULTS

### Defining the building blocks

The first step towards a neuromorphic quantum spin model is the construction of neuron-like building blocks. In other words, we need objects capable of sensing the state of a multi-spin input state and encoding information about relevant properties of this input into the state of an output spin. Additionally, we will require that this operation does not disturb the state of the input. This additional property is partially motivated by a similar property of the neurons used in e.g., classical feed forward networks, which similarly only exerts influence on the state of the network through their output. Furthermore, we show below that the preservation of inputs allow for interesting and non-trivial effects of the entanglement between the generated output and the preserved input. The cost of this preservation is a set of restrictions on each building block, as described in more detail below.

Inspired by the way classical neurons activate based on a (weighted) sum of their inputs, the first building block will be one that flips the state of its output spin, depending on how many of the input spins are in the ‘active’  $|\uparrow\rangle$ -state. The second building block, on the other hand, measures relative phases of components in the computational (i.e., the  $\sigma^z$ ) basis, and thus has no classical analog.

### Neuron 1: Counting excitations

In analogy to the thresholding behavior of classical spiking neurons, we start by constructing a spin system that is capable of detecting the number of excitations (i.e., the number of inputs in the state  $|\uparrow\rangle$ ) and exciting its output spin conditional on this information. As shown in Supplementary Note 1, this behavior can be implemented using dynamical evolution driven by the Hamiltonian

$$H_{\text{Exc}} = \frac{1}{2}(\sigma_1^x \sigma_2^x + \sigma_1^y \sigma_2^y + \sigma_1^z \sigma_2^z) + \beta \sigma_2^z \sigma_3^z + A \cos\left(\frac{2\beta}{\hbar} t\right) \sigma_3^x, \quad (1)$$

assuming the evolution is run for a time  $\tau = \pi\hbar/A$  and that the Hamiltonian fulfills some restrictions on the interaction strengths  $J, \beta, A$  that will be discussed below. In this model, we label spins 1 and 2 as the input, and spin 3 as the output. The intuition behind this model is that the Heisenberg interaction between the inputs sets up energy differences among the four possible Bell state of the inputs. Through the  $\sigma_2^z \sigma_3^z$  coupling to the output, these differences then influence the energy cost of flipping the output spin, resulting in the cosine drive on the output only being resonant when the input qubits are in certain states. The result is that the driving induces flips in the output qubit if and only if the input is in a Bell state with an even number of excitations. Since the conditionality is a resonance/off-resonance effect, the detuning of the undesired transitions needs to be much larger than the strength of the driving, which leads to the criterion

$$\Delta_{\pm} = \left| 2\beta \pm 2\sqrt{J^2 + \beta^2} \right| \gg A, \quad (2)$$

which is naturally fulfilled whenever the driving-strength  $A$  is much smaller than the chain interaction strength  $J$ .

Due to dynamical phases, the requirement that superpositions of input states should be preserved adds two additional constraints for the parameters of the model. Specifically, conservation of relative phases within the subspaces of inputs

that either flips (“above threshold”) or does not flip (“below threshold”) the output yields

$$\begin{aligned} \beta &= kA & k &\in \mathbb{Z} \\ J &= \pm \sqrt{l^2 - k^2} A & l &\in \mathbb{Z}. \end{aligned} \quad (3)$$

If these constraints are fulfilled, the only non-trivial phase will be a coherent phase between the above-threshold and below-threshold subspaces of  $-i(-1)^{k+l} \exp(-iJt/\hbar)$ , where  $t$  is the time elapsed during detection. Since these two subspaces are now distinguished by the state of the output, correcting for this phase is just a matter of performing the corresponding phase-gate on the output qubit. In the special case where  $\sqrt{l^2 - k^2}$  is an integer, this reduces to performing a  $\pi/2$ -rotation of the output about the  $z$ -axis (see Supplementary Note 1 for details).

When combined with this subsequent unconditional phase gate, the dynamical evolution induced by the Hamiltonian in (1) is to coherently detect the parity of the number of excitations of the input, and to encode this information in the output spin, i.e., in conventional Bell-state notation:

$$\begin{aligned} |\Psi^{\pm}\rangle|\downarrow\rangle &\longrightarrow |\Psi^{\pm}\rangle|\downarrow\rangle \\ |\Phi^{\pm}\rangle|\downarrow\rangle &\longrightarrow |\Phi^{\pm}\rangle|\uparrow\rangle, \end{aligned} \quad (4)$$

where the first ket denotes the state of the inputs and the second ket the state of the output. The output is either fully excited or not excited at all by the evolution—hence we refer to this structure as a spiking quantum neuron, in analogy to similar objects from classical computing. Sample simulations showing the dynamics of the spiking process are shown in Fig. 1. Note that the dynamics implementing this behavior is linear unitary time evolution, thus the effect on general 2-qubit inputs follows from expressing the input in the Bell-state basis and applying the rules of (4). For the neuron-parameters  $l=17$  and  $k=8$ , the corresponding operation is performed with an average fidelity of 99.98% when averaged over all possible 2-qubit inputs.

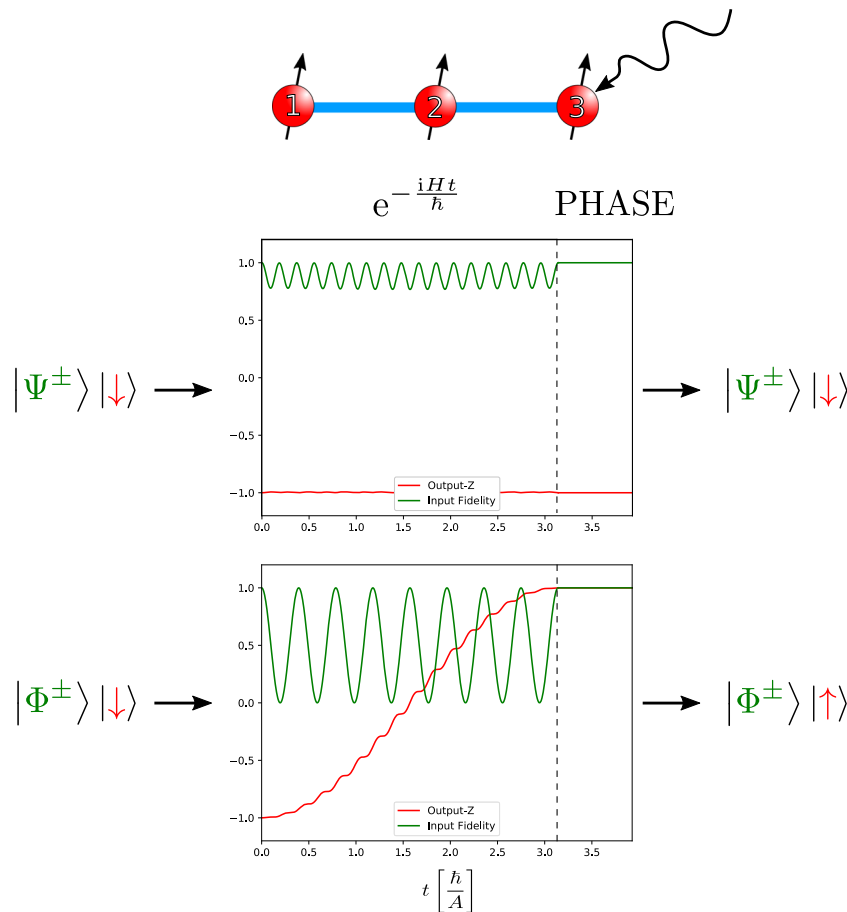
### Neuron 2: Detecting phases

While neuron 1 is a fully quantum mechanical object, capable of coherently treating superpositions in the inputs, the property that it detects—the number of excitations in the input—would be similarly well-defined for a classical neuron participating in a classical digital computation. However, the state of the two input qubits will also be characterized by properties that have no classical analog, such as the relative phases of terms in a superposition state. The goal of the second neuron is to be able to detect these relative phases of states in the computational basis. Specifically, it aims to distinguish the states  $\{|\Psi^+\rangle, |\Phi^+\rangle\}$  from the states  $\{|\Psi^-\rangle, |\Phi^-\rangle\}$ . Combining this detection-capability with the capabilities of the excitation-counting neuron of the previous section (exemplified by (4)) allows complete discrimination between the four Bell states.

The operational principle of the phase-detection neuron is similar to that of the excitation-detection neuron: it relies on a combination of single-qubit gates and the unitary time-evolution generated by a Hamiltonian of the form:

$$H_{\text{phase}} = \frac{J}{2}(\sigma_1^x \sigma_2^x + \sigma_1^y \sigma_2^y + \sigma_1^z \sigma_2^z) + \delta \sigma_2^x \sigma_3^x + B \sigma_3^z. \quad (5)$$

As shown in Supplementary Note 2, running the dynamics of this Hamiltonian for a time  $\tau = \pi\hbar/2B$  performs a  $(-iZ)$ -gate on the output qubit if and only if the state of input qubits are in the subspace spanned by  $|\Psi^-\rangle$  and  $|\Phi^-\rangle$ . Thus by conjugating this operation with Hadamard gates on the output qubit and correcting for the  $-i$ -phase using a phase-gate (see Fig. 2) yields



**Fig. 1 Dynamics of Neuron 1.** Schematic depiction of the excitation-number detecting neuron (top) and plots of the time-evolution of the output-qubit state (red) and the overlap between the state of the input qubits and their initial state (green) for the four Bell states during the operation of the neuron. As illustrated in these plots, the state of the output is either flipped or not depending on whether the input contains an odd (middle) or even (bottom) number of excitations. In contrast, the input qubits return to their initial states in all four cases. Parameters used are  $l = 17$ ,  $k = 8$ , which yields an average operational fidelity of 99.98% in the absence of noise.

the desired phase-detection operation:

$$\begin{aligned}
 |\Psi^+\rangle|\downarrow\rangle &\rightarrow |\Psi^+\rangle|\downarrow\rangle \\
 |\Phi^+\rangle|\downarrow\rangle &\rightarrow |\Phi^+\rangle|\downarrow\rangle \\
 |\Psi^-\rangle|\downarrow\rangle &\rightarrow |\Psi^-\rangle|\uparrow\rangle \\
 |\Phi^-\rangle|\downarrow\rangle &\rightarrow |\Phi^-\rangle|\uparrow\rangle.
 \end{aligned} \tag{6}$$

The fundamental principle of operation is identical to the one of the excitation-counting neuron, in that the Hamiltonian once again contains three terms: a Heisenberg interaction to set up an energy spectrum that distinguishes the Bell states, an interaction that tunes the energy of the output qubit (i.e., qubit 3) dependent on the state of the inputs, and a single-qubit operator attempting to change the state of the output and succeeding if and only if the driving related to this term matches the energy cost of flipping the output. The only difference is that the interaction term now sets up an energy splitting between the spin states  $|\pm\rangle = \frac{1}{\sqrt{2}}(|\downarrow\rangle \pm |\uparrow\rangle)$  rather than the states  $|\uparrow\rangle/|\downarrow\rangle$ , hence the need for Hadamard gates to convert between the two bases. Note that all of these operations are again linear, thus (6) specifies the operation also for general 2-qubit inputs. For the neuron-parameters  $n=82$  and  $m=3$ , the corresponding phase-detection operation is performed with an average fidelity of 99.07% when averaged over all possible 2-qubit inputs, with higher values achievable through small adjustments (see Supplementary Note 2 for details).

As the operation of the phase-detection neuron also relies on resonance/off-resonance effects, a restriction of similar to (2) is present. Specifically, we require that

$$\Delta = 2\delta \gg B. \tag{7}$$

Additionally, the requirement that the state of the inputs should not be distorted by the operation of the neuron yields the requirement that the ratios between  $J$ ,  $\delta$  and  $B$  should fulfill:

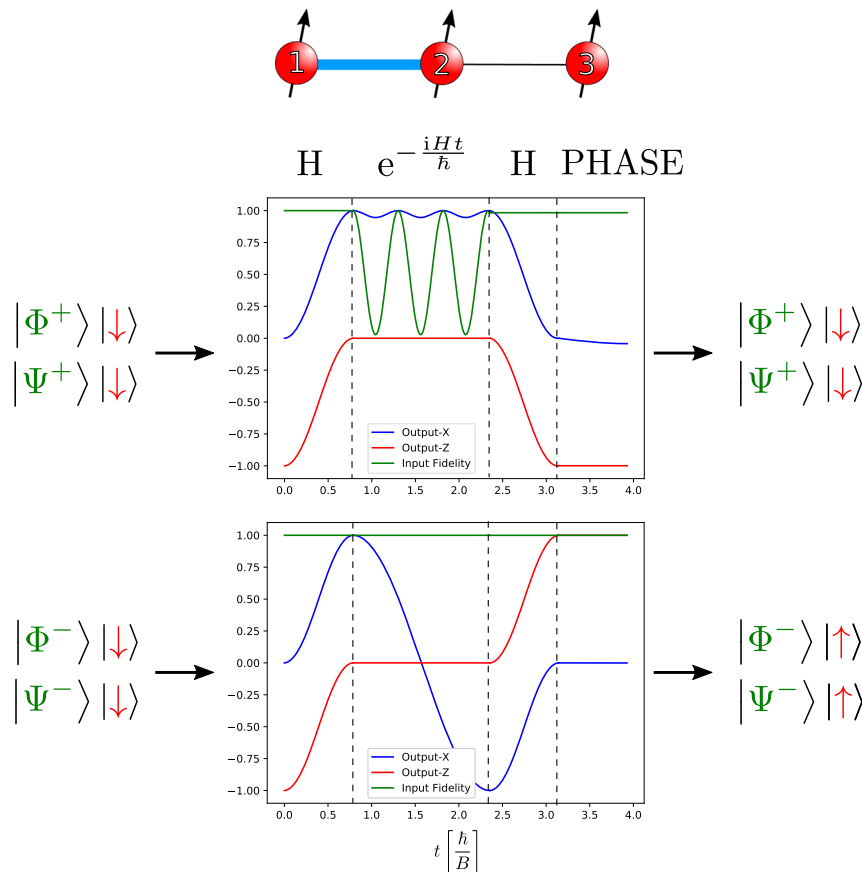
$$\begin{aligned}
 J &= 2nB \quad n \in \mathbb{Z} \\
 \delta &= 2mB \quad m \in \mathbb{Z}.
 \end{aligned} \tag{8}$$

with  $n \gg m$ .

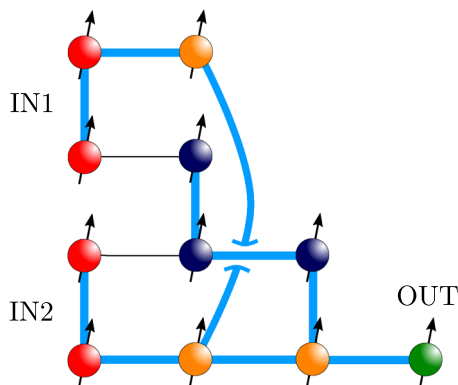
### State comparison network

Having introduced a set of computational building blocks above, we now aim to illustrate how these can be combined into larger networks in order to solve computation- and classification-problems. Specifically, we will illustrate how a network of these objects allows one to compare pairs of Bell states to determine if they are the same Bell state. As detailed below, such a network could play a central role in the certification of Bell-pair sources and quantum channels, and may also have potential applications for machine learning and state preparation tasks.

The basic structure of the proposed network is depicted in Fig. 3, and consists of three layers. The first layer constitutes the input to the system. It is in these four qubits that the two Bell states to be compared are stored. Each pair is used as an input to both of the



**Fig. 2 Dynamics of Neuron 2.** Schematic depiction of the phase-detecting neuron (top) and plots of the time-evolution of the X-component (blue) and Z-component (red) of the output-qubit state as well as the overlap between the state of the input qubits and their initial state (green) for the four Bell states during the operation of the neuron. As illustrated in these plots, the state of the output is either flipped or not depending on whether the input contains a positive (top) or negative (bottom) relative phase. In contrast, the input qubits return to their initial states in all four cases. Parameters used are  $n = 82$ ,  $m = 3$ , which yields an average operational fidelity of 99.07% in the absence of noise. As detailed in Supplementary Note 2, small adjustments around these values can yield a slightly higher fidelity, in this case 99.58%.



**Fig. 3 Bell-state comparison network.** Schematic depiction of network for comparing pairs of Bell states. The left-most layer marked in red constitute the inputs to the network, with subsequent layers extracting and comparing information about either the phase (blue) or the excitation parity (orange) of the input states, with the result of the comparison stored in the green output qubit.

types of neurons detailed above, with the output stored in two pairs of qubits in the second layer. In this way, sequentially running each of the two neuron operations extracts both the excitation-number parity and the relative phase of superpositions in the inputs and encodes it into the second layer. In other words,

this layer ends up containing exactly the information needed to distinguish among the four Bell states. State comparison therefore boils down to detecting if the information extracted from one input matches that extracted from the other input. Since detecting if two qubits are in the same state (i.e., both  $|\downarrow\rangle$  or both  $|\uparrow\rangle$ ) boils down to checking the number of excitations modulo two, this comparison can be done using the excitation-counting neuron defined above. The third layer thus encodes two bits of information: whether the excitation parity of the two inputs match, and whether the relative phases match. Detecting if the two inputs were the same Bell state is thus a matter of detecting if both of these bits are in the  $|\uparrow\rangle$ -state. This detection can be done using similar methods to those of the excitation-detection neuron—see Supplementary Note 3 for details.

The result of the manipulations is that the output is put into the  $|\uparrow\rangle$ -state if the two Bell-state inputs were identical and  $|\downarrow\rangle$  otherwise. Since all of the operations are achieved through linear, unitary dynamics, the behavior for superposition inputs follow from this rule and linearity. This also implies that the network cannot compare arbitrary states, as is indeed prohibited by the no-cloning theorem of quantum mechanics. For instance, two identical inputs in a superposition in the Bell basis may return either  $|\uparrow\rangle$  or  $|\downarrow\rangle$  as output:

$$\begin{aligned} & \frac{1}{2}(|\Psi^+\rangle + |\Phi^-\rangle)_{\text{IN1}}(|\Psi^+\rangle + |\Phi^-\rangle)_{\text{IN2}}|\downarrow\rangle_{\text{OUT}} \\ \longrightarrow & \frac{1}{2}(|\Psi^+\rangle_{\text{IN1}}|\Phi^-\rangle_{\text{IN2}} + |\Phi^-\rangle_{\text{IN1}}|\Psi^+\rangle_{\text{IN2}})|\downarrow\rangle_{\text{OUT}} \\ & + \frac{1}{2}(|\Phi^-\rangle_{\text{IN1}}|\Phi^-\rangle_{\text{IN2}} + |\Psi^+\rangle_{\text{IN1}}|\Psi^+\rangle_{\text{IN2}})|\uparrow\rangle_{\text{OUT}}. \end{aligned} \quad (9)$$

This illustrates an interesting property of quantum neural networks, namely that entanglement between inputs and outputs means that measurements of the outputs may have dramatic effects on the state of the inputs. In this case, a measurement of  $|\uparrow\rangle$  in the output will always project the input qubits into the states corresponding to this output, i.e., to identical Bell-state pairs. In this sense, a classifying quantum network like the one above will simultaneously be a projector onto the spaces corresponding to the states it is build to classify—a property that might prove helpful in, for instance, state preparation schemes. Note that this simple interpretation of measurement back-action follows from the fact that no other perturbations on the input state have been performed by the network during the computation, and hence from the corresponding non-disturbance requirement applied to each of the neurons.

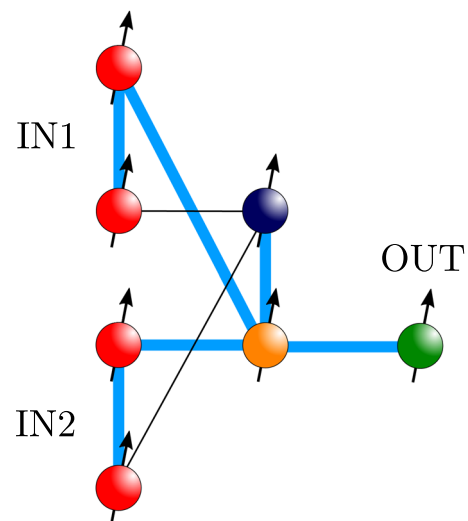
A more concrete application of the network above is the certification of quantum channels and Bell-state sources. The ability to determine the reliability of resources such as Bell-state sources and quantum channels would be a practical benefit in many quantum communication and quantum cryptography applications. This is an active area of research: for instance, device-independent self-testing through Bell inequalities works for certain multipartite entangled states<sup>26</sup>, or quantum template matching for the case where we have two possible template states<sup>27,28</sup>. Since the device presented above allows for the comparison of unknown systems with known-good ones, it is ideally suited to this kind of certification task.

Finally, it is worth noting that the comparative nature of the network means that the output of the network defines a kernel between 2-qubit quantum states. Specifically, given two inputs represented by amplitudes  $\{a_i\}$  and  $\{b_i\}$  in the Bell-state basis, the probability of measuring  $|\uparrow\rangle$  in the output will be given by

$$P_{|\uparrow\rangle} = \sum_i |a_i|^2 |b_i|^2 \quad (10)$$

which bears a strong resemblance to the classical “expected likelihood” kernel<sup>29</sup>. Considering this, comparison networks like the one above may also find applications within kernel-based quantum machine learning approaches<sup>30,31</sup>.

One thing to note is that the design of the structure in Fig. 3 is motivated by a desire for a one-step forward propagation of information between the layers of qubits, in analogy to the propagation of information in artificial classical neural networks. If each layer is allowed to probe the preceding layer more than once, a significant reduction in qubit overhead is possible. An example of this is the network depicted on Fig. 4. This network performs the same operation as the network in Fig. 3 while omitting the entire second layer. This is achieved by using the same qubit as target for both of the neurons detecting a given property. In this way, a shared property between the two input pairs means the two sequential detections result in an even number of flips to the corresponding qubit of the middle layer – either 0 or 2. On the other hand, non-identical properties will instead lead to precisely one flip. Thus the initial state of a qubit in the middle layer is preserved if and only if the property that it detects is identical between the two input pairs. The state of the output qubit is then determined by performing a flip if and only if both qubits in the middle layer have remained in their initial  $|\downarrow\rangle$ -state. In practice, this can be done using a similar generalization to the final step as the one used in the larger network – See Supplementary Note 3 for details. Thus allowing multiple sequential probings of the inputs by the middle layer allows the qubit-count to be reduced by 4, though at the expense of a slight increase in the complexity of the protocol for forward propagation of information as well as an increase in the maximum number of connections required by a qubit from 3 to 4.



**Fig. 4 Reduced Bell-state comparison network.** Schematic depiction of a network for comparing pairs of Bell states employing fewer qubits to do so than the one presented in the main text. The left-most layer marked in red constitute the inputs to the network, with subsequent layers extracting and comparing information about either the phase (blue) or the excitation parity (orange) of the input states, with the result of the comparison stored in the green output qubit.

## DISCUSSION

We have presented a set of building blocks that detects properties of two-qubit inputs and encodes these properties in a binary and coherent way into the state of an output qubit. To illustrate the power of such spiking quantum neurons, we have presented a network of these building blocks capable of identifying if two Bell states are identical or not, and argued the usefulness of such comparison networks for quantum certification tasks within quantum communication and quantum cryptography. Additionally, we have seen how the entanglement of the inputs and output results in highly non-trivial effects on the inputs when a measurement is performed on the output of the network.

From the considerations above, several interesting questions arise. A main question might be how to scale up structures made from these and similar building blocks into larger networks capable of performing more complex quantum processing tasks. Concerning scaling, it seems reasonable to expect that the kind of intuitive reasoning behind the operation of the network presented here will become ever more challenging. As a result, it might be fruitful to take inspiration from the field of classical neural networks and design quantum networks whose operation depend on parameters. In this way, one can then adjust these parameters to make the network perform a certain task, in a way analogous to how both classical and quantum neural networks are trained. Since the Hamiltonians responsible for the operation of the neurons already contain a number of parameters, the architecture presented in this paper seems well-suited to such an approach. An interesting avenue of further research is therefore the generalization of the dynamical models presented here to tunable models capable of detecting other structures in multi-qubit inputs than the two-qubit Bell-state properties detected here, or to models capable of solving other relevant quantum-computing problems, for instance parity detection for error correction protocols.

A central challenge in such a learning-based approach would be how best to optimize the parameters of the model, and how to identify the class of operations that can be implemented by a given model. Much work is currently being dedicated to these questions in the context of variational algorithms in gate-based quantum computation<sup>10,32–35</sup>. However, for the dynamical models

described in this paper, these problems are best framed within the field of optimal control theory, where a number of methods and results already exist on the optimization of pulses and parameters for quantum models<sup>36,37</sup>. Furthermore, advanced methods such as genetic algorithms<sup>38</sup> and reinforcement-learning<sup>39</sup> have recently begun to garner interest within this field. Thus the intersection of quantum optimal control and quantum machine learning seems a fertile avenue of further research, as recently pointed out in<sup>36</sup>. Nevertheless, optimization of the parameters of the model is in general expected to be a hard problem, although the difficulty compared to the corresponding optimizations currently faced by variational algorithms in gate-based quantum computers is believed by the authors to be an open question.

Another possible avenue of research towards improving the schemes presented here is to reduce the complexity of operation related to having to turn interactions between the different layers on and off by instead employing autonomous methods similar to those already used within the field of quantum error correction. Using such methods to perform quantum operations in a coherent way can be highly non-trivial, but for networks like the one described above where the latter stages of the network are essentially classical processing, strict coherences should not be needed for the network to operate, thus lowering the bar for autonomous implementations of similar networks. Furthermore, a faithful reproduction of the spiking action of biological neurons necessarily requires non-linearities and non-unitary reset. Thus engineered decoherence would also be an essential resource if more closely reproducing the continuous-time dynamical behavior of classical spiking neurons is the goal.

Finally, tapping into the temporality of the neurons presented above also holds great promise. Indeed, it has already been shown that the temporal behavior of comparatively simpler networks of spins allows for universal quantum computation<sup>40</sup>. Thus we believe that augmenting the neurons of this paper with less constrained and clock-like dynamics combined with tunable, teachable behavior and perhaps partial autonomy would be a promising route towards a neuromorphic architecture capable of solving complicated and interesting problems within quantum learning. Additionally, this approach will further distinguish the spiking quantum neural networks from conventional gate-based approaches, both through temporality and through increased complexity. Indeed, while all neuron operations presented here can be implemented on gate-based quantum computers at a cost of between two and six 2-qubit gates, it is unlikely that the same will hold true for the operation of larger, coupled, parametrized networks. This expectation mirrors the expected advantage of classical special-purpose hardware for neuromorphic computing—it does not perform computations that a general-purpose processor could not perform, but may do so faster and more efficiently. In a similar manner, the operations performed here could be emulated on either gate-based quantum computers or (universal) annealing architectures, but would require some overhead, for instance in the form of requiring multiple control pulses to emulate the single-pulse evolution of the excitation-counting neuron. Conversely, while we expect universal computation with models similar to those presented here to be possible—assuming either sufficiently large networks, like in<sup>40</sup>, or sufficiently reconfigurable couplings—we do not expect such a construction to be an efficient architecture for arbitrary classes of algorithms. In other words, it seems likely that a spiking quantum neural network will tend to naturally implement different operations, and therefore tackle different classes of problems, compared to annealing-based or gate-based algorithms, thus potentially making it a valuable additional tool for quantum machine learning.

## METHODS

### Simulations of the dynamics

The plots depicted on Figs. 1 and 2 were generated by numerically simulating the dynamics of the Hamiltonians (1) and (6), respectively, using the Python toolbox QuTiP<sup>41</sup>. Specifically, the system was initialized in states of the form

$$\begin{aligned} |\Psi_0\rangle &= |\text{Input}\rangle|\downarrow\rangle \\ |\text{Input}\rangle &\in \{|\Phi^\pm\rangle, |\Psi^\pm\rangle\}, \end{aligned} \quad (11)$$

where the two kets in the notation specifies the state of the input qubits (qubits 1 and 2, first ket) and the output qubit (qubit 3, second ket) separately. The built-in numerical solver of the QuTiP library was then used to find the trajectories resulting from the time-evolution of each of these states:

$$|\Psi(t)\rangle = e^{-iHt}|\Psi_0\rangle. \quad (12)$$

Using these trajectories, the evolution of the quantities depicted in the plot could then be calculated, including the expectation values related to the output qubit:

$$\begin{aligned} \text{Output-X} &: \langle\Psi(t)|\sigma_3^x|\Psi(t)\rangle \\ \text{Output-Z} &: \langle\Psi(t)|\sigma_3^z|\Psi(t)\rangle \end{aligned} \quad (13)$$

and the overlap between the current state of the inputs and their initial state:

$$\text{Input-Fidelity} : \langle\Psi(t)|(|\Psi_0\rangle\langle\Psi_0| + \sigma_3^z|\Psi_0\rangle\langle\Psi_0|\sigma_3^z)|\Psi(t)\rangle. \quad (14)$$

Note that the effect of the second term in this expectation-value is to trace out the dependence on the state of the output. The phase- and Hadamard-gates required for the operation of the neurons were implemented by evolving the system using Hamiltonians of the form:

$$\begin{aligned} H_{\text{Phase}} &= \begin{cases} A\sigma_3^z & \text{Exc. Neuron} \\ B\sigma_3^z & \text{Phase Neuron} \end{cases} \\ H_{\text{Had}} &= \sqrt{2}B(\sigma_3^x + \sigma_3^z) \end{aligned} \quad (15)$$

for a sufficient amount of time to implement the operation, i.e.:

$$\begin{aligned} \tau_{\text{Phase}} &= \begin{cases} \frac{\pi}{4A} & \text{Exc. Neuron} \\ \frac{\pi}{4B} & \text{Phase Neuron} \end{cases} \\ \tau_{\text{Had}} &= \frac{\pi}{4B}. \end{aligned} \quad (16)$$

Note that all simulations were performed without the simulation of noise and decoherence.

### Computation of the average fidelity

In order to quantify the performance of the neurons, the simulations of the dynamics were combined with tools based on<sup>42</sup> for calculating the average fidelity of operations, thus allowing the operations implemented by the neurons to be compared to the idealized operations defined in Eqs. (4) and (6). Each of the fidelities was calculated within the subspace consisting of the 6 states appearing in the corresponding definition, with the following additions made in order to fully specify the desired effects of the neurons within this subspace:

$$\begin{aligned} |\Psi^\pm\rangle|1\rangle &\rightarrow -i|\Psi^\pm\rangle|0\rangle && \text{Exc. Neuron} \\ |\Phi^-\rangle|1\rangle &\rightarrow i(-1)^{n+1}|\Phi^-\rangle|0\rangle && \text{Phase Neuron} \\ |\Psi^-\rangle|1\rangle &\rightarrow i(-1)^{n+1}|\Psi^-\rangle|0\rangle && \text{Phase Neuron.} \end{aligned} \quad (17)$$

In other words, the average fidelity was computed by comparing the implemented operation  $U_{\text{Neur}}$  to this idealized operation  $U_{\text{ideal}}$  and then averaging over a uniformly distributed ensemble over the space  $\mathcal{D}$  spanned by the 6 states used in the definition of the gate:

$$F_{\text{av}}(U_{\text{Neur}}) = \int_{\mathcal{D}} d\psi \left| \langle\psi|U_{\text{ideal}}^\dagger U_{\text{Neur}}|\psi\rangle \right|^2. \quad (18)$$

Note that the 6 states only enters in the ensemble of initial states—the simulations employed the full Hilberspace, and infidelity from leakage out of the 6-state space is fully accounted for by the fidelity-metric. In the specific models presented here, this leakage additionally turned out to be relatively negligible, on the order of  $1 \cdot 10^{-4}$ .

## DATA AVAILABILITY

The code and datasets used in the current study are available from the corresponding author upon request.

Received: 16 March 2020; Accepted: 26 January 2021;

Published online: 12 April 2021

## REFERENCES

- Waldrop, M. M. The chips are down for Moore's law. *Nature* **530**, 144–147 (2016).
- Gantz, J. & Reinsel, D. The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. *IDC iView IDC Anal. Future* **2007**, 1–16 (2012).
- Hashem, I. A. T. et al. The rise of "big data" on cloud computing: review and open research issues. *Inf. Syst.* **47**, 98–115 (2015).
- Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inform. Process. Syst.* **25**, 1097–1105 (2012).
- Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to sequence learning with neural networks. *Adv. Neural Inform. Process. Syst.* **27**, 3104–3112 (2014).
- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
- Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
- Dunjko, V. & Briegel, H. J. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep. Prog. Phys.* **81**, 074001 (2018).
- Kapoor, A., Wiebe, N. & Svore, K. Quantum perceptron models. *Adv. Neural Inform. Process. Syst.* **29**, 3999–4007 (2016).
- Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020).
- Killoran, N. et al. Continuous-variable quantum neural networks. *Phys. Rev. Research* **1**, 033063 (2019).
- Tacchino, F., Macchiavello, C., Gerace, D. & Bajoni, D. An artificial neuron implemented on an actual quantum processor. *npj Quantum Inf.* **5**, 26 (2019).
- Steinbrecher, G. R., Olson, J. P., Englund, D. & Carolan, J. Quantum optical neural networks. *npj Quantum Inf.* **5**, 60 (2019).
- Monràs, A., Sentís, G. & Wittek, P. Inductive supervised quantum learning. *Phys. Rev. Lett.* **118**, 190503 (2017).
- Albarrán-Arriagada, F., Retamal, J. C., Solano, E. & Lamata, L. Measurement-based adaptation protocol with quantum reinforcement learning. *Phys. Rev. A* **98**, 042315 (2018).
- Maass, W. Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* **10**, 1659–1671 (1997).
- Kjaergaard, M. et al. Superconducting qubits: current state of play. *Annu. Rev. Condens. Matter Phys.* **11**, 369–395 (2020).
- Kounalakis, M., Dickel, C., Bruno, A., Langford, N. K. & Steele, G. A. Tuneable hopping and nonlinear cross-Kerr interactions in a high-coherence superconducting circuit. *npj Quantum Inf.* **4**, 38 (2018).
- Wallraff, A. et al. Sideband transitions and two-tone spectroscopy of a superconducting qubit strongly coupled to an on-chip cavity. *Phys. Rev. Lett.* **99**, 050501 (2007).
- Roy, K., Jaiswal, A. & Panda, P. Towards spike-based machine intelligence with neuromorphic computing. *Nature* **575**, 607–617 (2019).
- Merolla, P. A. et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* **345**, 668–673 (2014).
- Davies, M. et al. Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018).
- Benjamin, B. V. et al. Neurogrid: a mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* **102**, 699–716 (2014).
- Gonzalez-Raya, T., Solano, E. & Sanz, M. Quantized three-ion-channel neuron model for neural action potentials. *Quantum* **4**, 224 (2020).
- Torrontegui, E. & García-Ripoll, J. J. Unitary quantum perceptron as efficient universal approximator. *EPL* **125**, 30004 (2019).
- Šupić, I., Coladangelo, A., Augusiak, R. & Acín, A. A simple approach to self-testing multipartite entangled states. *New J. Phys.* **20**, 083041 (2017).
- Sasaki, M., Carlini, A. & Jozsa, R. Quantum template matching. *Phys. Rev. A* **64**, 022317 (2001).
- Sentís, G., Calsamiglia, J., Muñoz-Tapia, R. & Bagan, E. Quantum learning without quantum memory. *Sci. Rep.* **2**, 708 (2012).
- Jebara, T., Kondor, R. & Howard, A. Probability product kernels. *J. Mach. Learn. Res.* **5**, 819–844 (2004).
- Schuld, M. & Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
- Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
- Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).

- McClellan, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
- Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at <https://arxiv.org/abs/1802.06002> (2018).
- Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at <https://arxiv.org/abs/1411.4028> (2014).
- Magann, A. B. et al. From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms. *PRX Quantum* **2**, 010101 (2021).
- Yang, X.-d. et al. Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses. *Phys. Rev. A* **102**, 062605 (2020).
- Mortimer, L., Estarellas, M. P., Spiller, T. P. & D'Amico, I. Evolutionary computation for adaptive quantum device design. Preprint at <https://arxiv.org/abs/2009.01706> (2020).
- Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *npj Quantum Inf.* **5**, 33 (2019).
- Childs, A. M., Gosset, D. & Webb, Z. Universal computation by multiparticle quantum walk. *Science* **339**, 791–794 (2013).
- Johansson, J. R., Nation, P. D. & Nori, F. Qutip: an open-source python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **183**, 1760–1772 (2012).
- Nielsen, M. A. A simple formula for the average gate fidelity of a quantum dynamical operation. *Phys. Lett. A* **303**, 249–252 (2002).

## ACKNOWLEDGEMENTS

L.B.K. and N.T.Z. acknowledge funding from the Carlsberg Foundation and the Danish Council for Independent Research (DFF-FNU). M.D. acknowledges support by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Quantum Algorithms Teams Program. A.A.-G. acknowledges support from the Army Research Office under Award No. W911NF-15-1-0256 and the Vannevar Bush Faculty Fellowship program sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering (Award number ONR 00014-16-1-2008). A.A.-G. also acknowledges generous support from Anders G. Frøseth and from the Canada 150 Research Chair Program.

## AUTHOR CONTRIBUTIONS

N.T.Z. and L.B.K. formulated the initial goal of the research, and M.D., P.W. and A.A.-G. subsequently participated in a further refinement of the scope and focus of the research. The bulk of the analytical and numerical work of the study as well as the creation of the initial draft was performed by L.B.K., with further significant contributions to the manuscript by M.D. and P.W. and crucial revisions by A.A.-G. and N.T.Z.

## 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/s41534-021-00381-7>.

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

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

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



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

© The Author(s) 2021