ARTICLE   OPEN

Quantum optical neural networks

Gregory R. Steinbrecher[1], Jonathan P. Olson[2], Dirk Englund [1] and Jacques Carolan[1]

Physically motivated quantum algorithms for specific near-term quantum hardware will likely be the next frontier in quantum information science. Here, we show how many of the features of neural networks for machine learning can naturally be mapped into the quantum optical domain by introducing the quantum optical neural network (QONN). Through numerical simulation and analysis we train the QONN to perform a range of quantum information processing tasks, including newly developed protocols for quantum optical state compression, reinforcement learning, black-box quantum simulation, and one-way quantum repeaters. We consistently demonstrate that our system can generalize from only a small set of training data onto inputs for which it has not been trained. Our results indicate that QONNs are a powerful design tool for quantum optical systems and, leveraging advances in integrated quantum photonics, a promising architecture for next-generation quantum processors.

INTRODUCTION

Deep learning is revolutionizing computing[1] for an ever-increasing range of applications, from natural language processing[2] to particle physics[3] to cancer diagnosis.[4] These advances have been made possible by a combination of algorithmic design[5] and dedicated hardware development.[6] Quantum computing,[7] while more nascent, is experiencing a similar trajectory, with a rapidly closing gap between current hardware and the scale required for practical implementation of quantum algorithms. Error rates on individual quantum bits (qubits) have steadily decreased,[8] and the number and connectivity of qubits have improved,[9] making the so-called noisy intermediate-scale quantum (NISQ) processors[10] capable of tasks too hard for a classical computer a near-term prospect. Experimental progress has been met with algorithmic advances[11] and near-term quantum algorithms have been developed to tackle problems in combinatorics,[12] quantum chemistry,[13] and solid-state physics.[14] However, it is only recently that the potential for quantum processors to accelerate machine learning has been explored.[15]

Quantum machine learning algorithms for universal quantum computers have been proposed[16] and small-scale demonstrations implemented.[17] Relaxing the requirement of universality, quantum machine learning for NISQ processors has emerged as a rapidly advancing field[18–22] that may provide a plausible route towards practical quantum-enhanced machine learning systems. These protocols typically map features of machine learning algorithms (such as hidden layers in a neural network) directly onto a shallow quantum circuits in a platform-independent manner. In contrast, the work presented here leverages features unique to a particular physical platform.

Although the demonstration of an unambiguous quantum advantage in machine learning is an open question,[23] an increasing number of results and heuristic arguments indicate quantum systems are well suited to addressing such computational tasks. First, certain classes of non-universal quantum processor have been shown to sample from probability distributions that, under plausible complexity theoretic conjectures, cannot be sampled from classically.[24] For example, ensembles of non-interacting photons (which is a subclass of the architecture presented here) sample from non-classical distributions even without the optical nonlinearities required for quantum universality.[25,26] Speculatively, this may enable quantum networks, in certain instances, to surpass classical networks in both generative and recognition tasks.

Second, classical machine learning algorithms typically involve many linear algebraic operations. Existing quantum algorithms have already demonstrated theoretical speedups in problems related to many of the most elementary algebraic operations such as Fourier transforms,[27] vector inner products,[28] matrix eigenvalues and eigenvectors,[29] and linear system solving.[30] These techniques may form parts of a toolkit enabling quantum machine learning algorithms. Finally, certain physical systems, such as those studied in quantum chemistry, are naturally encoded by quantum information.[31] Quantum features of these states, such as coherence and entanglement, are naturally exploitable by networks that themselves are inherently quantum. Classical computers on the other hand require an exponential (in, for instance, the number of spin orbitals of a molecule) amount of memory to even encode such states.

In this work, we introduce an architecture that unites the complexity of quantum optical systems with the versatility of neural networks: the quantum optical neural network (QONN). We argue that many of the features that are natural to quantum optics (mode mixing, optical nonlinearity) can directly be mapped to neural networks, and train our system to implement both coherent quantum operations and classical learning tasks, suggesting that our architecture is capable of much of the functionality of both its parent platforms. Moreover, technological advances driven by trends in photonic quantum computing[32] and the microelectronics industry[33] offer a plausible route towards large-scale, high-bandwidth QONNs, all within a CMOS (complementary-metal-oxide-semiconductor)-compatible platform.

[1]Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA and [2]Zapata Computing Inc., 501 Massachusetts Avenue, Cambridge, MA 02139, USA
Correspondence: Jacques Carolan (carolanj@mit.edu)

(a)

A Classical Neural Network

$$\rho_{out} = \max\{0, \rho_{in}\}$$
$$\sigma_{out} = \frac{1}{1 + e^{-(\sigma_{in}+b)}}$$

Linear Transform   Linear Transform   Linear Transform

$\mathbf{W}(\vec{\theta}_1)$ $\mathbf{W}(\vec{\theta}_2)$ $\mathbf{W}(\vec{\theta}_3)$

Input
Layer   Hidden
Layer   Hidden
Layer   Output
Layer

(b)

A Quantum Optical Neural Network

$$\Sigma(\phi) = \sum_{n=0}^{\infty} e^{in(n-1)\phi/2} |n\rangle\langle n|$$

Linear Transform   Linear Transform   Linear Transform

$\mathbf{U}(\vec{\theta}_1)$ $\mathbf{U}(\vec{\theta}_2)$ $\mathbf{U}(\vec{\theta}_3)$

Fock
Input State   Single-Site
Nonlinearities   Single-Site
Nonlinearities   Photon Counting
Detectors
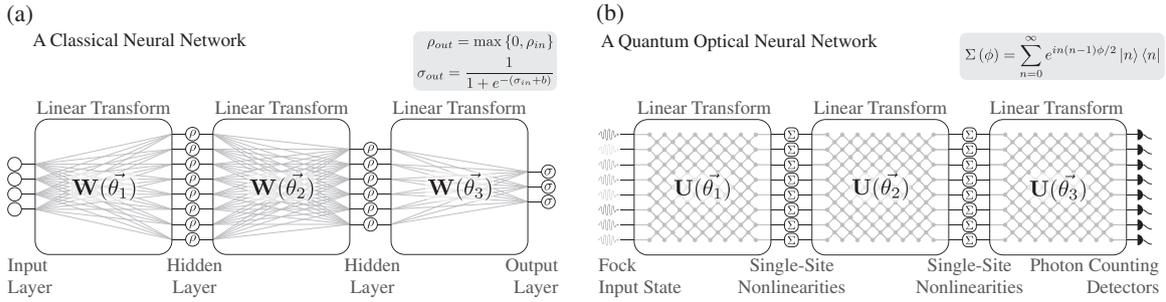
Fig. 1 Quantum optical neural network (QONN). **a** An example of a classical neural network architecture. Hidden layers are rectified linear units (ReLUs) and the output neuron uses a sigmoid activation function to map the output into the range (0, 1). **b** An example of our quantum optical neural network (QONN) architecture. Inputs are single photon Fock states. The single-site nonlinearities are given a Kerr-type interaction applying a phase quadratic in the number of photons. Readout is given by photon-number-resolving detectors, which measure the photon number at each output mode

Through numerical simulation and analysis, we apply our architecture to a number of key quantum information science protocols. We benchmark the QONN by designing quantum optical gates where circuit decompositions are already known. Next, we show that our system can learn to simulate other quantum systems using only a limited set of input/output state pairs, generalizing what it learns to previously unseen inputs. We demonstrate this learning on both Ising and Bose–Hubbard Hamiltonians. We then introduce and test a new quantum optical autoencoder protocol for compression of quantum information, with applications in quantum communications and quantum networks. This again relies on the ability to train our systems using a subset of possible inputs. Next, we apply our system to a classical machine learning controls task, balancing an inverted pendulum, by a reinforcement learning approach. Finally, we train the QONN to implement a one-way quantum repeater, whose physical implementation was, until now, unknown. Our results may find application both as an important technique for designing next-generation quantum optical systems and as a versatile experimental platform for near-term optical quantum information processing and machine learning. Moreover, machine learning protocols for NISQ processors typically operate on quantum states for which there are no clear classical analog. Similarly, the QONN may be able to perform inference on quantum optical states, such as those generated by molecular systems[34] or states within a quantum network.[35]

In prototypical neural networks (see Fig. 1a) an input vector $\vec{x} \in \mathbb{R}^n$ is passed through multiple layers of: (1) linear transformation, that is, a matrix multiplication $W(\theta_i).\vec{x}$ parameterized by weights θ_i at layer i, and (2) nonlinear operations $\sigma(\vec{x})$, which are single-site nonlinear functions sometimes parameterized by biases $\vec{b}_i$ (typically referred to as the perceptron or neuron, see Fig. 1a, inset for two examples: the rectifying neuron and the sigmoid neuron). The goal of the neural network is to optimize the parameter sets $\{\theta_i\}$ and $\{b_i\}$ to realize a particular input–output function $f(\vec{x}) = y$. The power of neural networks lies in the fact that when trained over a large data set $\{\vec{x}_i\}$, this often highly nonlinear functional relationship is generalizable to a large vector set to which the network was not exposed during training. For example, in the context of cancer diagnosis, the input vectors may be gray-scale values of pixels of an image of a cell, and the output may be a two-dimensional vector that corresponds to the binary label of the cell as either a benign or malignant.[36] Once the network is trained, it may categorize with high probability new, unlabelled, images of cells as either "benign" or "malignant".
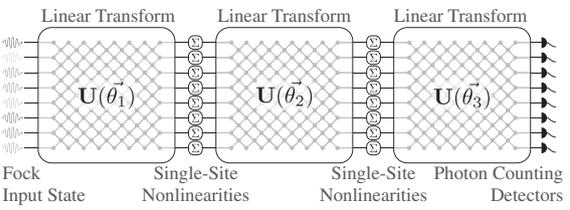
RESULTS

Architecture

As shown in Fig. 1b, input data to our QONN is encoded as photonic Fock states i_j (corresponding to i photons in the jth optical mode), which for n photons in m modes is described by a $\binom{n+m-1}{m}$-dimensional complex vector of unit magnitude. As we will show, leveraging the full Fock space may be advantageous for training certain classes of QONN. The linear circuit is described by an m-mode linear optical unitary $U(\vec{\theta})$ parameterized by a vector $\vec{\theta}$ of $m(m-1)$ phases shifts $\theta_i \in (0, 2\pi)$ via the encoding of Reck et al.[37] The nonlinear layer Σ comprises single-mode Kerr interactions in the monochromatic approximation, applying a phase that is quadratic in the number of photons present.[38] For a given interaction strength ϕ, this unitary can be expressed as $\Sigma(\phi) = \sum_{n=0}^{\infty} e^{in(n-1)\phi/2}|n\rangle\langle n|$. The full system comprising N layers is therefore

$$S(\vec{\Theta}) = \prod_{i}^{N} \Sigma(\phi).U(\vec{\theta}_i), \tag{1}$$

where $\vec{\Theta}$ is a $Nm(m-1)$-dimensional vector and the strength of the nonlinearity is typically fixed as $\phi = \pi$. Finally, photon-number-resolving detectors are used to measure the photon number at each output. We consider number resolution without loss of generality as the so-called threshold detectors (vacuum, or not) can always be made non-deterministically number resolving via beamsplitters and multiple detectors.[39] We use the results of this measurement, along with a training set of K desired input/output pairs $\{|\psi_{in}^i\rangle \to |\psi_{out}^i\rangle\}_{i=1}^{K}$, to construct a cost function

$$C(\vec{\Theta}) = 1 - 1/K \sum_{i=1}^{K} \left| \langle\psi_{out}^i | S(\vec{\Theta}) |\psi_{in}^i\rangle \right|^2 \tag{2}$$

that is variationally minimized over $\vec{\Theta}$ to find a target transformation (up to an unobservable global phase). In the Supplementary Information S2 we show that the QONN architecture is also capable of implementing classical optical neural networks,[40] and may therefore benefit from advances in this rapidly growing field.[41]

We distinguish between two approaches to training: in situ and in silico. The in situ approach directly optimizes the quantum optical processor and measurements are made via single photon detectors at the end of the circuit. One aim is to optimize figures of merit that can be estimated with a number of measurements that scales polynomially with the photon number (as opposed to full quantum process tomography).[42] If the target state is
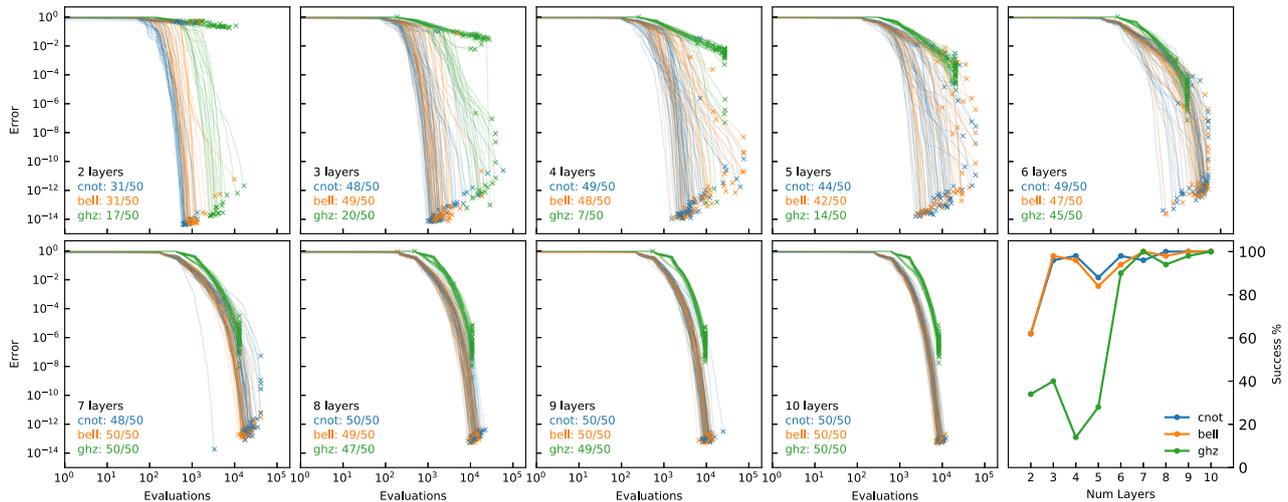
Fig. 2 Benchmarking results. The first nine figures show 50 training runs for each of three representative optical quantum computing tasks: performing a controlled NOT (CNOT) gate, separating/generating Bell states, and generating GHZ states. Evaluation number is defined as the number of updates of $\vec{\Theta}$. At low layer depth, the optimizations frequently fail to converge to an optimal value (we defined an error $<10^{-4}$ as "success"), terminating at relatively large errors. This behavior gets worse as we add layers, out to five layers, at which point it undergoes a rapid reversal, with the training essentially always succeeding at layer depths of seven or more. This is shown in the final figure, where success percentage is plotted against the number of layers for each of the three tasks. The non-monotonic behavior is due to the large variance in final costs at low layer number. In Supplementary Information S1 we plot layer number against median error, recovering the expected monotonic behavior

accessible, the overlap can be estimated with the addition of a controlled-SWAP operation, which is related to the Hong–Ou–Mandel effect in quantum optics.[43] Efficient fidelity proxies provide another route towards estimating salient features of quantum states without reconstruction of the full density matrix.[44] Moreover, the in situ approach may enable a form of error mitigation by routing quantum information around faulty hardware.[45] In contrast, the in silico approach simulates the QONN on a digital classical computer and keeps track of the full quantum state internal to the system. Simulations will therefore be limited in scale, but may help guide the design of, say, quantum gates where the optimal decomposition is not already known, or as an ansatz for the in situ approach. In the Methods, we detail the computational techniques used in this work.

Hardware implementation

A number of the key components of QONNs are readily implementable using state-of-the-art integrated quantum photonics. First, matrix multiplication can be realized across optical modes (where each mode contains a complex electric field component) via arrays of beamsplitters and programmable phase shifts.[37,46] In the lossless case, an n-mode optical circuit comprising $n(n-1)$ component implements an arbitrary $n \times n$ single particle unitary operation (which can also be used for classical neural networks),[47] and a n-dimensional non-unitary operation can always be embedded across a $2n$-mode optical circuit.[48] Advances in integrated optics[49] have enabled the implementation of such circuits for applications in quantum computation,[50] quantum simulation,[51] and classical optical neural networks.[40] Second, optical nonlinearities are a core component of many classical[52] and quantum[53] optical computing architectures. Single photon coherent nonlinearities can be implemented via measurement,[54] interaction with three-level atoms[55] or superconducting materials,[56] and through all-optical phenomena such as the Kerr effect.[57] Notably, promising progress has been made towards solid-state waveguide-based nonlinearities.[58] Third, superconducting nanowire single photon detectors (SNSPDs) enable ultra-efficient single photon readout, either via low-loss out-coupling[59] to a dedicated high-efficiency detection system[60]

or through the direct integration of SNSPDs on chip.[61] Moreover, advances in electronic readout have made it is possible to scale SNSPDs across many channels and with photon-number resolution.[62] While incorporating these technologies into a single scalable system is an outstanding challenge, hybrid integration techniques provide a path towards combining otherwise incompatible material platforms.[63]

In this work, we focus on discrete variable QONNs due to the maturity of the technology platform, but note that continuous variable implementations are also promising.[64] Our discrete variable architecture can naturally be mapped to other platforms that manipulate bosonic modes such as ultracold atoms,[65] superconducting cavities,[66] or phononic modes in trapped ions.[67] In each of these platforms, significant progress has been made towards reconfigurable linear mode transformations[68] to compliment pre-existing ultra-strong nonlinearities, thus making bosonic quantum simulators excellent candidates for near-term QONNs.

Benchmarking

As a first step in validating our architecture, we ensure it can learn elementary quantum tasks such as quantum state preparation, measurement, and quantum gates. We chose Bell-state projection/generation, Greenberger–Horne–Zeilinger (GHZ) state generation, and the implementation of the controlled NOT (CNOT) gate as representative of typical optical quantum information tasks. As described in the Methods section, in each of these cases the training set represents the full basis set for the quantum operation of interest, and successful training tells us something about the expressivity of our architecture.

We trained QONNs of increasing layer depth from $N = 2 \rightarrow 10$ with $\phi = \pi$. As shown in Fig. 2, at short layer depths the optimization frequently terminates early, finding a non-optimal local minima. We observe similar behavior for all of the studied tasks. Most notable here is the behavior of the optimization as the layer count increases: just like a classical neural network, as we increase the layer depth, it becomes consistently easy to find a local minimum that is close to the global minimum. This demonstrates the utility of deep networks: while a single layer may be *sufficient* to implement, for exampe, a CNOT gate, with

(a)

(b)
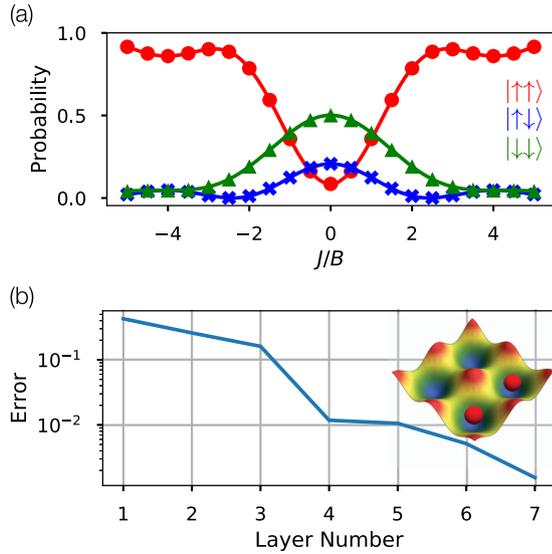
Fig. 3 Quantum optical neural networks (QONNs) for Hamiltonian simulation. **a** Ising model. A three-layer QONN is trained for a range of interaction strengths $J/B \in [-5, 5]$ and the probability for particular output spin configuration is plotted (points) given the $|\uparrow\uparrow\rangle$ initialization state. The expected evolution is plotted alongside (lines). Critically, during the training process our QONN was never exposed to the initialization state. **b** Bose–Hubbard model. Number of layers required to reach a particular test error for the simulation of a (2,4) strongly interacting $U/t_{hop} = 20$ Bose–Hubbard Hamiltonian (schematic shown in inset) with $t = 1$. Training is performed 20 times for each layer depth, and the lowest test error is recorded. The single-layer system gives a mean error in the test set of 42% and seven layers yields an error of 0.1%

deep networks we can reliably discover a configuration that yields the correct operation. For more complex operations, where the small-layer-number implementation may be difficult to find or simply not exist, this gives hope that we can still reliably train a deep network to perform the task. While the inputs and desired outputs are restricted to the dual-rail basis, we have verified that at intermediate layers the joint state of the photons span the entire Fock space, which is a unique feature of photonic systems. In the Supplementary Information S3 we examine the trade-off between nonlinear interaction strength and gate fidelity.

Hamiltonian simulation
While the results thus far benchmark the training of the QONN, a critical feature of any learning system is that it can generalize to states on which it has not been trained. To assess generalization, we apply the QONN to the task of quantum simulation, whereby a well-controlled system in the laboratory $S(\vec{\Theta})$ is programmed over parameters $\vec{\Theta}$ to mimic the evolution of a quantum system of interest described by the Hamiltonian $\hat{H}$. In particular, we train our QONN on K sets of input/output states $\{|\psi_{in}^i\rangle\} \{|\psi_{out}^i\rangle\}$ related by the Hamiltonian of interest $|\psi_{out}^i\rangle = \exp(-i\hat{H}t)|\psi_{in}^i\rangle$, and test it on new states which it has not been exposed to.

As a first test we look at the Ising model (see Methods section), which is optically implemented via a dual-rail encoding with $m = 2n$, where $|\uparrow\rangle \equiv |10\rangle_{12}$ and $|\downarrow\rangle \equiv |01\rangle_{12}$. For the $n = 2$ spin case, we train the QONN on a training set of 20 random two-photon states and test it on 50 different states. We empirically determine that for a wide range of J/B values (with $t = 1$), a three-layer QONN reliably converges to an optimum. In Fig. 3a we vary the interaction strength J/B and plot the probability of finding a particular spin configuration given an initialization state $|\uparrow\uparrow\rangle$. Critically, this input state is not in set of states for which the QONN was trained. We

also train our QONN for the $n = 3$ spin case, reaching an average test error of 10.1%. This higher error in the larger system motivates the need for advanced training methods such as backpropagation[69] or layer-wise training approaches[70] to efficiently train deeper QONN.

Finally, we look at a Hamiltonian more natural for photons in optical modes, the Bose–Hubbard model, see Methods section for further details. Now, the (n, m) configuration of bosons to be simulated is naturally mapped to an n-photon m-mode photonic system.

To benchmark our system, we look at the number of layers required to express a (2, 4) strongly interacting Bose–Hubbard model on a square lattice (Fig. 3b, inset). Figure 3b shows that increasing the number of layers reduces the error on the test set, suggesting that deeper networks can express a richer class of quantum functions (i.e. Hamiltonians); a concept familiar in classical deep neural networks.[71] Choosing five layers to give a reasonable trade-off between error (~1%) and computational tractability, we vary the interaction strength in the the range $U/t_{hop} \in [-20, 20]$. Across all numerical simulations we achieve a mean test error of 2.9 ± 1.3% (error given by the standard deviation in 22 simulations).

While our analysis has focused on Hamiltonians that exist in nature, the approach itself is very general: mimicking input–output configurations given access to a reduced set of input–output pairs from some family of quantum states. This may find application in learning representations of quantum systems where circuit decompositions are unknown, or finding compiled implementations of known circuits.

Quantum optical autoencoder
Photons play a critical role in virtually all quantum communication and quantum networking protocols, either as information carriers themselves or to mediate interactions between long-lived atomic memories.[72] However, such schemes are exponentially sensitive to loss: given a channel transmissivity η and the number of photons n required to encode a message, the probability of successful transmission scales as η^n. Reducing the photon number while maintaining the information content, therefore, exponentially increases the communication rate. In the following we use the QONN as a quantum autoencoder to learn a compressed representation of quantum states. This compressed representation could be used, for example, to more efficiently and reliably exchange information between physically separated quantum nodes.[35]

Quantum autoencoders have been proposed as a general technique for encoding, or compressing, a family of states on n qubits to a lower-dimensional k-qubit manifold called the latent space.[73,74] Similar to classical autoencoders, a quantum autoencoder learns to generalize from a small training set T and is able to compress states from the family that it has not previously seen. As well as applications in quantum communication and quantum memory, it has recently been proposed as a subroutine to augment variational algorithms in finding more efficient device-specific ansatzes.[75] In contrast, the quantum optical autoencoder encodes input states in the Fock basis. Moreover, even if optical input states are encoded in the dual-rail qubit basis, the autoencoder may learn a compression onto a non-computational Fock basis latent space.

As a choice of a family of states, and one which is relevant to quantum chemistry on NISQ processors, we consider the set of ground states of molecular hydrogen, H_2, in the STO-3G minimal basis set,[76] mapped from their fermionic representation into qubits via the Jordan–Wigner transformation.[77] Ground states in this qubit basis have the form $|\psi(i)\rangle = \alpha(i)|0011\rangle_L + \beta(i)|1100\rangle_L$, where i is the bond length of the ground state. The qubits themselves are represented in a dual-rail encoding thus the
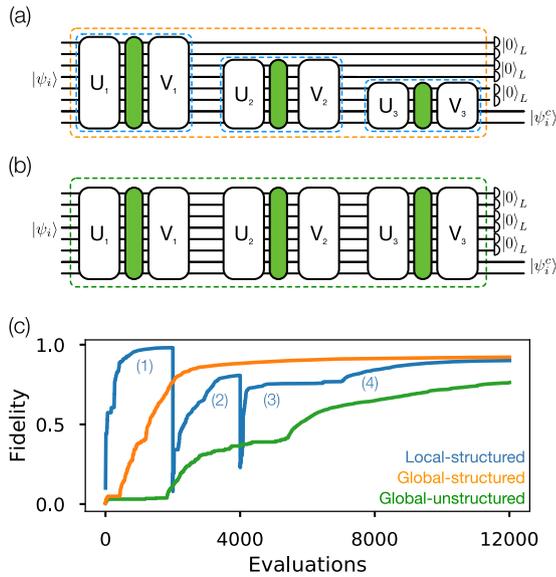
Fig. 4 Quantum optical autoencoder. a, b Schematics of the quantum optical neural network (QONN) architectures corresponding to each of the three training strategies. While the architecture of the globally structured (**a**, orange) and globally unstructured (**b**, green) optimizations remained the same throughout the entire optimization, the locally structured approach (**a**, blue) optimized the parameters of (1) U_1 and V_1 first (with the nonlinear layer shown in green), before moving on to (2) U_2 and V_2, and in the third phase (3) U_3 and V_3. The final refinement step of the iterative approach (4) considered all parameters in the optimization, similar to the global strategy. **c** A plot of the fidelities of the reference states achieved by the different training strategies to compress ground states of molecular hydrogen. While the global (orange) and unstructured (orange) optimizations included all three reference qubits from the start, the large drops in fidelity for the iterative procedure (blue) are due to including increasingly more reference states in the optimization. The global and iterative methods converge to a fidelity of 92.2% and 90.0%, respectively, and unstructured achieved 76.2%

network consists of $n = 4$ photons in $m = 8$ optical modes. We note that the set of states $\{|\psi_i\rangle\}$ are no longer related by a single unitary transformation as in previous sections.

The goal of the quantum optical autoencoder S is for all states in the training set $|\psi_i\rangle \in K$, satisfying

$$S|\psi_i\rangle = |000\rangle_L |\psi_i^C\rangle \qquad (3)$$

for some two-mode state $|\psi_i^C\rangle$ in the latent space. The quantum autoencoder can therefore be seen as an algorithm that systematically disentangles $n - k$ qubits from the set of input states and sets them to a fixed reference state (e.g., $|0\rangle_L^{\otimes n-k}$). For this reason, the fidelity of the reference state will be used a proxy for the fidelity of the decoded state.

To train a quantum autoencoder one should choose a circuit architecture with general enough operations to compress the input states, but few enough parameters to train the network efficiently. As shown in Fig. 4a, b, we test three training schemes for the QONN autoencoder: (1) locally structured training (Fig. 4a, blue): sequentially optimizing two-layer QONNs to disentangle a single qubit at each stage, where each subsequent stage acts only on a reduced qubit subspace. This approach is followed by a final global refinement step after all layers have been individually trained; (2) globally structured training (Fig. 4a, orange): where the above layer structure is trained simultaneously rather than sequentially; and (3) globally unstructured training (Fig. 4b, green): where a six-layer system acting on all four qubits is trained.

The optimization was performed using an implementation of MLSL[78] (also available in the NLopt library), which is a global optimization algorithm that explores the cost function landscape with a sequence of local optimizations (in this case BOBYQA) from carefully chosen starting points, using a heuristic to avoid local optima that have already been found. Our training states are the set of four ground states of H_2 corresponding to bond lengths of 0.5, 1.0, 1.5, and 2.0 Å. Both the global and iterative optimizations performed comparably. However, we note that the the iterative approach could potentially be made more efficient if more stringent convergence criteria were introduced. The unstructured optimization achieved a lower fidelity; however, it is unclear from our data whether the iterative approach would have better scaling or accuracy than the global optimization in an asymptotic setting.

Quantum reinforcement learning

To demonstrate the utility of QONNs for classical machine learning tasks, and to show that they continue to generalize in that setting, we examine a standard reinforcement learning problem: that of trying to balance an inverted pendulum.[79] Classical deep reinforcement learning uses a policy network, that is, a network that takes an observation vector as input and outputs a probability distribution over the space of allowed actions. This probability vector is then sampled to choose an action, a new observation is taken, and the process repeats. As the output from a QONN is inherently a probability distribution, policy networks are a natural application. See Methods for further details.

We simulate a cart moving on a one-dimensional frictionless track, with a pole on a hinge attached to its top (see Fig. 5c, inset). At the beginning of the simulation, the cart is initialized to a random position, with the pole at a random angle. At each time step, the neural network receives four values, the position of the cart x, its velocity $\dot{x}$, the angle of the pole with respect to the track θ, and the time derivative of that angle $\dot{\theta}$. From those four values, it determines whether to apply a force of unit magnitude either in the $+x$ or $-x$ directions; those are the only two options. Each run of the simulation continues until a boundary condition in x, θ, or t ($t_{max} = 300$) is reached (i.e., the cart runs into the edge of the track or the pole falls over). The number of time steps before failure is the fitness of that run; we want to make this as large as possible.

As shown in Fig. 5a, b we demonstrate training using two different input encodings. First, we directly encode the four observation values x, $\dot{x}$, θ, and $\dot{\theta}$ onto four dual-rail qubits. Second, we encode these four values onto a uniform quantum state over two qubits, a type of quantum random access memory (QRAM) encoding. While it is unknown in general how to efficiently encode a given state into a QRAM, this numerical simulation demonstrates that these networks are capable of learning from general, highly entangled, quantum states, not just those with direct classical analogs.

Both encodings are performed by first compressing each of the four observation variables into $\gamma_j \in [0, \pi/2]$ ($j \in \{1, ..., 4\}$). For the direct encoding, each qubit q_j is set to $\cos(\gamma_j)|0\rangle_L + \sin(\gamma_j)|1\rangle_L$. For the QRAM encoding, the state over the two input qubits is set to $\frac{1}{4}\left(e^{i\gamma_1}|00\rangle_L + e^{i\gamma_2}|01\rangle_L + e^{i\gamma_3}|10\rangle_L + e^{i\gamma_4}|11\rangle_L\right)$. Finally, the QRAM encoding is given an ancilla qubit to act as phase reference.

We use this qubit encoding only for ease of encoding; after this point, we no longer regard the photons as qubits and simply measure the output state, potentially increasing the computational power of the system. In both systems, we picked the arbitrary measure of "number of photons in mode 1" vs. "number of photons in mode 2": if the number of photons in the first mode exceeds the number in the second mode, we apply a force in the $-x$ direction; otherwise, we apply a force in the $+x$ direction. Finally, we train these networks using an evolutionary strategies method.[80]
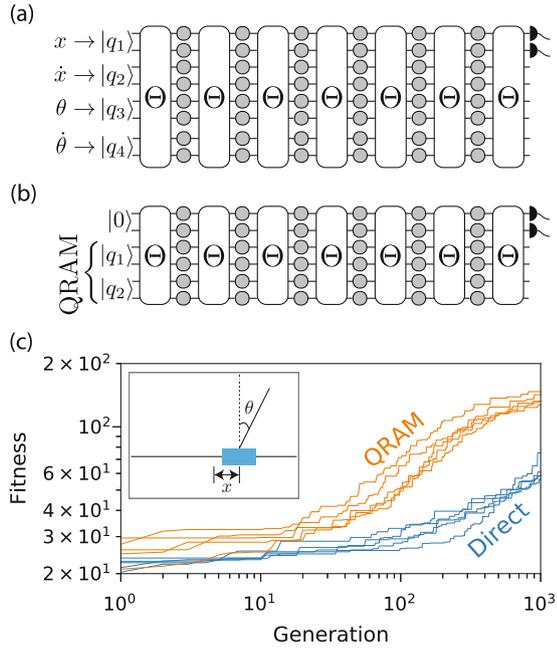
Fig. 5 Quantum reinforcement learning. a Architecture for the directly encoded reinforcement learning network. Each observation variable (x, $\dot{x}$, θ, and $\dot{\theta}$) was mapped to a phase $\gamma \in [0, \pi/2]$ and the corresponding dual-rail-encoded input qubit was set to $\sin(\gamma)0_L + \cos(\gamma)1_L$. Each Θ layer is an independent arbitrary unitary transformation; the gray boxes represent single-site nonlinearities. **b** Architecture for the quantum random access memory (QRAM-) encoded reinforcement learning network. The observation values were mapped to phases as in the direct architecture, which were then encoded into the QRAM (see text). **c** Fitness vs. training generation curves for five different training runs of each type of the reinforcement learning QONN. A higher fitness corresponds to a network that was able to keep the pole upright and the cart within the bounds for more time. The direct encoding requires more parameters and hence is slower to train. Inset: The problem we are trying to solve, a cart on a bounded one-dimensional track with an inverted pendulum on the top

In Fig. 5c we show the results of five training cycles (each with different starting conditions) using a six-layer QONN. For each cycle, we use a batch size of 100 to determine the approximate gradient, and average the fitness over 80 distinct runs of the network at each $\vec{\Theta}$ we evaluate. Hyperparameters (layer depth, batch size, and averaging group) were tuned using linear sweeps. These values apply to both the direct and QRAM encodings. Fitness increases with training generation, meaning the QONN consistently learns to balance the pole for longer times as generation increases; that is, it generalizes examples it has previously seen to new instances of the problem.

To cross-check our performance we trained equivalently sized classical networks, that is, four-neuron, six-layer networks with constant width. Hidden layers had ReLu neurons, while the final layer was a single sigmoid neuron to generate a probability $p \in 0, 1)$ of applying force in the $-x$ direction. We used the same training strategy for the classical networks as for the QONNs and observed a comparable performance, with a mean fitness after 1000 generations in the classical case of 37.1 compared with 61.9 for the directly encoded QONN and 136.1 for the QRAM encoded QONN. The direct encoding took ~5000 generations to reach a comparable fitness as the QRAM. Both networks can likely be optimized, and one should be cautious in directly comparing the classical and quantum results. Notwithstanding, this exploratory work demonstrates that quantum systems can learn on physically

relevant data, and future directions will seek to leverage uniquely quantum properties such as superposition for batch learning.[81]

One-way quantum repeaters

Finally, to demonstrate both the flexibility of the QONN platform and the advantages of co-designing both the architecture and the physical platform, we demonstrate the realization of one-way quantum repeaters. The goal of a one-way quantum repeater is equivalent to that of forward error correction in classical communications: to distribute the information over several symbols in such a way that even if errors occur, the original information can still be recovered. In quantum optics the primary error mechanism is loss; therefore, one should encode a single qubit of information across n photons such that if $m \leq k$ photons are lost (for a k-loss tolerant code), the state can be repaired without round trip communications between the sender and the receiver (see Fig. 6a). Loss correction techniques are critical both for quantum communications over distance[82] and protecting qubits in photonic quantum computing schemes.[83]

In this work, we focus on a recent proposal for unitary one-way quantum repeaters, which do not require measurements or quantum memories.[84] While it can be shown that Hamiltonians for one-way repeaters exist, the question of how to realize these with physical components remains open. Here we train the QONN architecture to implement such quantum repeater schemes, demonstrating the utility of physically realizable variational quantum architectures.

We consider the two-mode code

$$
\begin{aligned}
|0\rangle_L &\equiv (|40\rangle_{12} + |04\rangle_{12})/\sqrt{2}, \\
|1\rangle_L &\equiv |22\rangle_{12},
\end{aligned}
\tag{4}
$$

which is robust against single photon loss. It can be shown that for an input state $|\psi\rangle_L = a|0\rangle_L + \beta|1\rangle_L$, the loss of a single photon can be corrected by a system $\hat{S}$, which coherently performs the map

$$
\hat{S}|30\rangle_{12} = (|40\rangle_{12} + |04\rangle_{12})/\sqrt{2},
\tag{5}
$$

$$
\hat{S}|03\rangle_{12} = (|40\rangle_{12} + |04\rangle_{12})/\sqrt{2},
\tag{6}
$$

$$
\hat{S}|12\rangle_{12} = |22\rangle_{12},
\tag{7}
$$

$$
\hat{S}|21\rangle_{12} = 21|21\rangle_{12}.
\tag{8}
$$

Mathematically, $\hat{S}[\hat{a}_1\rho\hat{a}_1^\dagger]\hat{S}^\dagger = \rho$ and $\hat{S}[\hat{a}_2\rho\hat{a}_2^\dagger]\hat{S}^\dagger = \rho$, where $\rho = |\psi\rangle_{LL}\langle\psi|$.

By photon-number preservation, $\hat{S}$ cannot be unitary on two modes, but $\hat{S}$ can be realized as a unitary with additional ancilla. To train the QONN to implement this mapping, we do the following: let $\{|\psi_i\rangle_L\}_i$ be the set of states $\{|0\rangle_L, |1\rangle_L, (|0\rangle_L + |1\rangle_L)/\sqrt{2}, (|0\rangle_L - |1\rangle_L)/\sqrt{2}, (|0\rangle_L + i|1\rangle_L)/\sqrt{2}, (|0\rangle_L - i|1\rangle_L)/\sqrt{2}.$, and $\sigma_{i,j} = \hat{a}_j\rho_i\hat{a}_j^\dagger$. The action of $\hat{S}$ on the computational (non-ancilla) modes with single photon loss is given by

$$
\sigma_{i,j}^{(out)} = \mathrm{Tr}_A[\hat{S}(\sigma_{i,j} \otimes \rho_A)\hat{S}^\dagger],
\tag{9}
$$

where ρ_A is the input ancilla state. In the lossless case the output is given by

$$
\rho_i^{(out)} = \mathrm{Tr}_A[\hat{S}(\rho_i \otimes \rho_A)\hat{S}^\dagger].
\tag{10}
$$

The desired system should be able to correct all inputs that have single photon-loss error, and also leave the input undisturbed if there is no photon loss. This corresponds to the map $\sigma_{i,j}^{(out)} = \rho_i^{(out)} = \rho_i \forall i, j$.

Numerically, we calculate a cost function that quantifies the average distance (given by the Hilbert–Schmidt inner product $\mathrm{Tr}[A^\dagger B]$) between the six photon subtracted states and non-photon subtracted states, and variationally optimize the QONN.

(a)

(b)
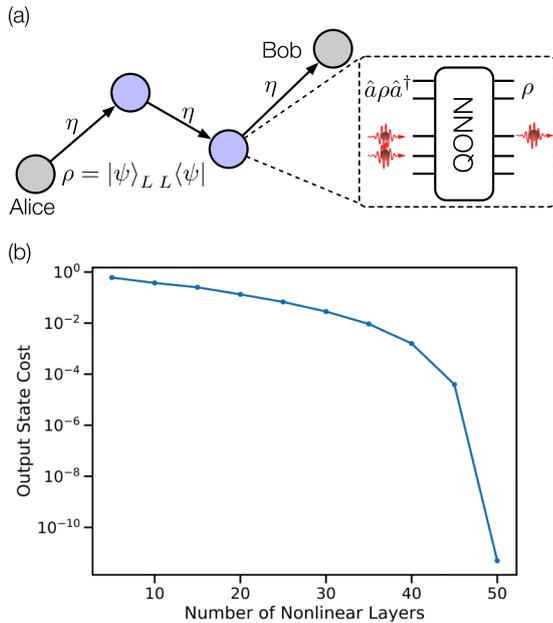
Fig. 6 Learning one-way quantum repeaters. **a** One-way quantum repeaters (shown in blue) are used to correct photon loss on logically encoded qubits $|\psi\rangle_L$ sent through a lossy channel with transmissivity η. The quantum optical neural network (QONN) (inset) can be trained to implement such repeater with the addition of ancillary photons and modes. **b** Numerical simulation results of a $(m, n) = (4, 2)$ code, which corrects single photon loss. The output fidelity for a given number of layers is plotted, reaching numerical accuracy at 50 layers

Due to the complexity of the system a backpropagation method was developed and gradient-based optimization methods were used, to achieve efficient and accurate training. Figure 6b plots the average fidelity of the output states against the number of nonlinear layers, reaching numerical precision at 50 layers. In conclusion, the QONN yields an explicit optical construction of a one-way quantum repeater, which was otherwise unknown. We therefore anticipate other physically motivated variational architectures to yield insights which platform-independent approaches cannot.

DISCUSSION

We have proposed an architecture for near-term quantum optical systems that maps many of the auspicious features of classical neural networks onto the quantum domain. Through numerical simulation and analysis we have applied our QONN to a broad range of quantum information processing tasks, including newly developed protocols such as quantum optical state compression for quantum networking and black-box quantum simulation. Experimentally, advances in integrated photonics and nanofabrication have enabled monolithically integrated circuits with many thousands of optoelectronic components.[85] The architecture we present is not limited to the integration of systems with strong single photon nonlinearities and we anticipate our approach will serve as a natural intermediate step[10] towards large-scale photonic quantum technologies. In this intermediate regime, the QONN may learn practical quantum operations with weak or noisy nonlinearities, which are otherwise unsuitable for fault-tolerant quantum computing.[86] The effect of such noise is an important subject for future work.

Future work will likely focus on loss correction techniques, which are also possible in an all-optical context.[87] Additionally, classical neural networks have benefitted greatly from improved

training techniques. For example, in transfer learning, a trained network has its final layer or two removed and new layers added, which are then trained for an entirely new application.[88] Future work will explore whether similar techniques may be used to more efficiently solve new problems in the QONN architecture, and whether different architectures such as generative adversarial networks can be applied to quantum optics.[89] Together, our results point towards both a powerful simulation tool for the design of next-generation quantum optical systems and a versatile experimental platform for near-term optical quantum information processing and machine learning.

METHODS

Computational techniques
The quantum optics simulations in this work were performed with custom, optimized code written in Python, with performance-sensitive sections translated to Cython. The Numba library was used to GPU accelerate some large operations. The most computationally intensive step was the calculation of the multi-photon unitary transform ($U(\vec{\theta}_i)$ in Eq. 1) from the single photon unitary. The multi-photon unitary has $\binom{n + m - 1}{n}^2$ entries, each of which requires calculating the permanent of an $n \times n$ matrix.[90]

As with classical neural networks, different optimization algorithms perform better for different tasks. We rely on gradient-free optimization techniques that optimize an objective function without an explicitly defined derivative (or one based on finite difference methods), as computing and backpropagating the gradient through the system likely requires knowledge of the internal quantum state of the system, preventing efficient in situ training. While this might be acceptable for designing small systems in simulation (say, designing quantum gates), it does not allow for systems to be variationally trained in situ. We empirically determined that the BOBYQA algorithm[91] performs well for most applications in terms of speed and accuracy for our QONN, and is available in the NLopt library.[92] We note that calculation of such a gradient is possible with classical optical neural networks.[93] For the quantum reinforcement learning simulations, we used our own implementation of evolutionary strategies.[80] At each stage evolution strategies takes a vector parameterizing the network, generates a population of new vectors by repeatedly perturbing the vector with gaussian noise, and then calculates a fitness for each perturbed vector. The new vector is then the fitness-weighted average of all the perturbed vectors. Evolution strategies does not require backpropagation, in comparison to strategies based on Markov decision processes, making it more suitable for quantum applications.

Hardware and libraries
The computer used to perform these simulations is a custom-built workstation with a 12-core Intel Core i7-5820K and 64 GB of RAM. The GPU used was an Nvidia Tesla K40. Relevant software versions are: Ubuntu 16.04 LTS, Linux 4.13.0-39-generic #44 16.04.1-Ubuntu SMP, Python 2.7.12, NumPy 1.14.1, NLopt 2.4.2, Cython 0.27.3, and Numba 0.40.0.

Benchmarking training
The training set for the Bell-state projector is the full set of Bell states $\{|\psi_{in}^i\rangle\} = \{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ encoded as dual-rail qubits. Our goal is to map these to a set of states distinguishable by single photon detectors, thus we opt for a binary encoding $\{|\psi_{out}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0110\rangle, |0101\rangle\}$. A system designed to perform this map can then be run in reverse to generate Bell states from input Fock states. The CNOT gate uses a full input–output basis set with $\{|\psi_{in}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0110\rangle, |0101\rangle\}$ and $\{|\psi_{out}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0101\rangle, |0110\rangle\}$. For the GHZ generator we select just a single input–output configuration $\{|\psi_{in}^i\rangle\} = \{|101010\rangle\}$ and $\{|\psi_{out}^i\rangle\} = \{(|101010\rangle + |010101\rangle)/\sqrt{2}\}$.

Simulated Hamiltonians

The Ising model we simulate is described by the Hamiltonian

$$H_{ising} = B \sum_i \hat{X}_i + J \sum_{\langle i,j \rangle} \hat{Z}_i \otimes \hat{Z}_j, \tag{11}$$

where B represents the interaction of each spin with a magnetic field in the x direction, and J is the interaction strength between spins in an orthogonal direction. The Bose–Hubbard model we simulate is described by the Hamiltonian

$$\hat{H}_{BH} = \omega \sum_i \hat{b}_i^\dagger \hat{b}_i - t_{hop} \sum_{\langle i,j \rangle} \hat{b}_i^\dagger \hat{b}_j + U/2 \sum_i \hat{n}(\hat{n}_i - 1), \tag{12}$$

where $\hat{b}_i^\dagger$ $(\hat{b}_i)$ represents the creation (annihilation) operator in mode i, $\hat{n}_i$ the number operator and ω, t_{hop}, and U the on-site potential, the hopping amplitude, and the on-site interaction strength, respectively.

DATA AVAILABILITY

The data sets generated during and or analyzed during the current study are available from the corresponding author on reasonable request. The QONN codebase is available at https://github.com/steinbrecher/bosonic.

AUTHOR CONTRIBUTIONS

G.R.S., D.E. and J.C. conceived of the project. The core codebase was built by G.R.S. with contributions from J.C. Numerical experiments were performed by G.R.S., J.P.O. and J.C. All authors contribute to the writing of the manuscript. D.E. and J.C. supervised the project.

ADDITIONAL INFORMATION

Supplementary Information accompanies the paper on the *npj Quantum Information* website (https://doi.org/10.1038/s41534-019-0174-7).

Competing interests: The authors declare no competing interests.

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

REFERENCES

1. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
2. Wu, Y. et al. Google's neural machine translation system: bridging the gap between human and machine translation. Preprint at arXiv:1609.08144v2 (2016).
3. Radovic, A. et al. Machine learning at the energy and intensity frontiers of particle physics. *Nature* **560**, 41–48 (2018).
4. Capper, D. et al. DNA methylation-based classification of central nervous system tumours. *Nature* **555**, 469–474 (2018).
5. Glorot, X., Bordes, A. & Bengio, Y. Deep sparse rectifier neural networks. In *Proc. 14th International Conference on Artificial Intelligence and Statistics* (eds Gordon, G., Dunson, D. & Dudík, M.) 315–323 (PMLR, Fort Lauderdale, FL, USA, 2011).
6. Sze, V., Chen, Y.-H., Yang, T.-J. & Emer, J. S. Efficient processing of deep neural networks: a tutorial and survey. *Proc. IEEE* **105**, 2295–2329 (2017).
7. Nielsen, M. & Chuang, I. *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
8. Barends, R. et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500–503 (2014).
9. Bernien, H. et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature* **551**, 579–584 (2017).
10. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
11. Montanaro, A. Quantum algorithms: an overview. *npj Quantum Inform.* **15023** (2016).
12. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at arXiv:1411.4028v1 (2014).
13. Aspuru-Guzik, A. & Walther, P. Photonic quantum simulators. *Nat. Phys.* **8**, 285–291 (2012).
14. Wecker, D. et al. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A* **92**, 062318 (2015).
15. Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
16. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014).
17. Cai, X. D. et al. Entanglement-based machine learning on a quantum computer. *Phys. Rev. Lett.* **114**, 110504–110505 (2015).
18. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. Preprint at arXiv:1803.00745v1 (2018).
19. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. Preprint at arXiv:1802.06002v1 (2018).
20. Schuld, M., Bocharov, A., Svore, K. & Wiebe, N. Circuit-centric quantum classifiers. Preprint at arXiv:1804.00633v1 (2018).
21. Havlicek, V. et al. Supervised learning with quantum enhanced feature spaces. Preprint at arXiv:1804.11326v2 (2018).
22. Schuld, M. & Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
23. Aaronson, S. Read the fine print. *Nat. Phys.* **11**, 291–293 (2015).
24. Boixo, S. et al. Characterizing quantum supremacy in near-term devices. Preprint at arXiv:1608.00263v3 (2016).
25. Aaronson, S. & Arkhipov, A. The computational complexity of linear optics. In *Proc. 43rd annual ACM Symposium on Theory of Computing* 333–342 (ACM, New York, NY, USA, 2011). https://doi.org/10.1145/1993636.1993682.
26. Olson, J. The role of complexity theory in quantum optics—a tutorial for Boson ampling. *J. Opt.* **20**, 123501 (2018).
27. Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. 35th Annual Symposium on Foundations of Computer Science* 124–134 (IEEE Computer Society, Washington, DC, USA, 1994). https://doi.org/10.1109/SFCS.1994.365700.
28. Buhrman, H., Cleve, R., Watrous, J. & de Wolf, R. Quantum fingerprinting. *Phys. Rev. Lett.* **87**, 167902 (2001).
29. Lloyd, S. & Abrams, D. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Phys. Rev. Lett.* **83**, 5162 (1999).
30. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
31. Seeley, J. T., Richard, M. J. & Love, P. J. The Bravyi–Kitaev transformation for quantum computation of electronic structure. *J. Chem. Phys.* **137**, 224109 (2012).
32. O'Brien, J. L. Optical quantum computing. *Science* **318**, 1567–1570 (2007).
33. Sun, C. et al. Single-chip microprocessor that communicates directly using light. *Nature* **528**, 534–538 (2015).
34. Sparrow, C. et al. Simulating the vibrational quantum dynamics of molecules using photonics. *Nature* **557**, 660–667 (2018).
35. Kimble, H. J. The quantum internet. *Nature* **453**, 1023–1030 (2008).
36. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. & Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. *CSBJ* **13**, 8–17 (2015).
37. Reck, M., Zeilinger, A., Bernstein, H. J. & Bertani, H. Experimental realization of any discrete unitary operator. *Phys. Rev. Lett.* **73**, 58–61 (1994).
38. Loudon, R. *The Quantum Theory of Light* (Oxford University Press, 2000). https://books.google.com/books?id=guHRngEACAAJ.
39. Carolan, J. et al. On the experimental verification of quantum complexity in linear optics. *Nat. Photon.* **8**, 621–626 (2014).
40. Shen, Y. et al. Deep learning with coherent nanophotonic circuits. *Nat. Photon.* **11**, 441–446 (2017).
41. Bagherian, H. et al. On-chip optical convolutional neural networks. Preprint at arXiv:1808.03303 (2018).
42. O'Brien, J. L. et al. Quantum process tomography of a Controlled-NOT Gate. *Phys. Rev. Lett.* **93**, 080502 (2004).
43. Garcia-Escartin, J. C. & Chamorro-Posada, P. swap test and Hong–Ou–Mandel effect are equivalent. *Phys. Rev. A* **87**, 052330–10 (2013).
44. Cramer, M. et al. Efficient quantum state tomography. *Nat. Commun.* **1**, 149–7 (2010).
45. Mower, J., Harris, N. C., Steinbrecher, G. R., Lahini, Y. & Englund, D. High-fidelity quantum state evolution in imperfect photonic integrated circuits. *Phys. Rev. A* **92**, 032322–7 (2015).
46. Clements, W. R., Humphreys, P. C., Metcalf, B. J., Kolthammer, W. S. & Walmsley, I. A. Optimal design for universal multiport interferometers. *Optica* **3**, 1460–1465 (2016).
47. Arjovsky, M., Shah, A. & Bengio, Y. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning* (eds Balcan, M. F. & Weinberger, K. Q.) 1120–1128 (PMLR, New York, NY, USA, 2016).

48. Miller, D. A. B. Self-configuring universal linear optical component [Invited]. *Photon. Res.* **1**, 1–15 (2013).
49. Harris, N. C. et al. Linear programmable nanophotonic processors. *Optica* **5**, 1623–1631 (2018).
50. Carolan, J. et al. Universal linear optics. *Science* **349**, 711–716 (2015).
51. Lanyon, B. P. et al. Towards quantum chemistry on a quantum computer. *Nat. Chem.* **2**, 106–111 (2010).
52. Miller, D. A. B. Device requirements for optical interconnects to silicon chips. *Proc. IEEE* **97**, 1166–1185 (2009).
53. Kok, P. et al. Linear optical quantum computing with photonic qubits. *Rev. Mod. Phys.* **79**, 135–174 (2007).
54. Knill, E., Laflamme, R. & Milburn, G. A scheme for efficient quantum computation with linear optics. *Nature* **409**, 46–52 (2001).
55. Duan, L. M. & Kimble, H. J. Scalable photonic quantum computation through cavity-assisted interactions. *Phys. Rev. Lett.* **92**, 125–4 (2004).
56. Kirchmair, G. et al. Observation of quantum state collapse and revival due to the single-photon Kerr effect. *Nature* **495**, 205–209 (2013).
57. Heuck, M., Jacobs, K. & Englund, D. R. Photon–photon interactions in dynamically coupled cavities. Preprint at arXiv:1905.02134 (2019).
58. Javadi, A. et al. Single-photon non-linear optics with a quantum dot in a wave-guide. *Nat. Commun.* **6**, 8655 (2015).
59. Notaros, J. et al. Ultra-efficient CMOS fiber-to-chip grating couplers. In *Optical Fiber Communication Conference* M2I-5 (Optical Society of America, 2016). http://www.osapublishing.org/abstract.cfm?URI=OFC-2016-M2I.5.
60. Marsili, F. et al. Detecting single infrared photons with 93% system efficiency. *Nat. Photon.* **7**, 210–214 (2013).
61. Najafi, F. et al. On-chip detection of non-classical light by scalable integration of single-photon detectors. *Nat. Commun.* **6**, 1–8 (2015).
62. Zhu, D. et al. A scalable multi-photon coincidence detector based on super-conducting nanowires. *Nat. Nanotechnol.* **13**, 596 (2018).
63. Kim, J.-H. et al. Hybrid integration of solid-state quantum emitters on a silicon photonic chip. *Nano Lett.* **17**, 7394–7400 (2017).
64. Killoran, N. et al. Continuous-variable quantum neural networks. Preprint at arXiv:1806.06871 (2018).
65. Preiss, P. M. et al. Strongly correlated quantum walks in optical lattices. *Science* **347**, 1229–1233 (2015).
66. Krastanov, S. et al. Universal control of an oscillator with dispersive coupling to a qubit. *Phys. Rev. A* **92**, 040303–5 (2015).
67. Lau, H.-K. & James, D. F. V. Proposal for a scalable universal bosonic simulator using individually trapped ions. *Phys. Rev. A* **85**, 289–311 (2012).
68. Peropadre, B., Guerreschi, G. G., Huh, J. & Aspuru-Guzik, A. Proposal for micro-wave Boson sampling. *Phys. Rev. Lett.* **117**, 0489–6 (2016).
69. Verdon, G., Pye, J. & Broughton, M. A universal training algorithm for quantum deep learning. Preprint at arXiv:1806.09729v1 (2018).
70. Bengio, Y., Lamblin, P., Popovic, D. & Larochelle, H. Greedy Layer-Wise Training of Deep Networks. In *Advances in Neural Information Processing Systems 19* 153–160 (MIT Press, 2007). http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf.
71. Raghu, M., Poole, B., Kleinberg, J., Ganguli, S. & Sohl-Dickstein, J. On the expressive power of deep neural networks. Preprint at arXiv:1606.05336v6 (2016).
72. Gisin, N. & Thew, R. Quantum communication. *Nat. Photon.* **1**, 165–171 (2007).
73. Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2**, 023023 (2017).
74. Wan, K. H., Dahlsten, O., Kristjánsson, H., Gardner, R. & Kim, M. Quantum gen-eralisation of feedforward neural networks. *npj Quantum Inf.* **3**, 36 (2017).
75. Olson, J., Sim, S. & Cao, Y. Implementation of cusp using cirq. https://github.com/zapatacomputing/cusp_cirq_demo (2018).
76. Helgaker, T., Olsen, J. & Jorgensen, P. *Molecular Electronic Structure Theory* (Wiley, Chichester, UK, 2013).
77. Tranter, A. et al. The Bravyi–Kitaev transformation: properties and applications. *Int. J. Quantum Chem.* **115**, 1431–1441 (2015).
78. Kan, A. H. G. R. & Timmer, G. T. Stochastic global optimization methods part I: clustering methods. *Math. Program.* **39**, 27–56 (1987).
79. Barto, A. G., Sutton, R. S. & Anderson, C. W. Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst. Man Cybernet.* **SMC-13**, 834–846 (1983).
80. Salimans, T., Ho, J., Chen, X., Sidor, S. & Sutskever, I. Evolution strategies as a scalable alternative to reinforcement learning. Preprint at arXiv:1703.03864 (2017).
81. Ristè, D. et al. Demonstration of quantum advantage in machine learning. *npj Quantum Inf.* **3**, 16 (2017).
82. Pant, M., Krovi, H., Englund, D. & Guha, S. Rate-distance tradeoff and resource costs for all-optical quantum repeaters. *Phys. Rev. A* **95**, 012304 (2017).
83. Varnava, M., Browne, D. E. & Rudolph, T. How good must single photon sources and detectors be for efficient linear optical quantum computation? Preprint at arXiv:quant-ph/0702044v2 (2007).
84. Miatto, F. M., Epping, M. & Lütkenhaus, N. Hamiltonians for one-way quantum repeaters. *Quantum* **2**, 75 (2018).
85. Chung, S., Abediasl, H. & Hashemi, H. A monolithically integrated large-scale optical phased array in silicon-on-insulator CMOS. *IEEE J. Solid-State Circuits* **53**, 275–296 (2017).
86. Shapiro, J. H. Single-photon Kerr nonlinearities do not help quantum computa-tion. *Phys. Rev. A* **73**, 1502–1511 (2006).
87. Niu, M. Y., Chuang, I. L. & Shapiro, J. H. Hardware-efficient bosonic quantum error-correcting codes based on symmetry operators. *Phys. Rev. A* **97**, 032323 (2018).
88. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proc. ICML Workshop on Unsupervised and Transfer Learning* (eds Guyon, I., Dror, G., Lemaire, V., Taylor, G. & Silver, D.) 17–36 (PMLR, Bellevue, Washington, USA, 2012).
89. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**, 040502 (2018).
90. Scheel, S. Permanents in linear optical networks. Preprint at arXiv:0406127v1 (2004).
91. Powell, M. J. *The BOBYQA Algorithm for Bound Constrained Optimization without Derivatives*. Cambridge NA Report NA2009/06 26–46 (University of Cambridge, Cambridge, 2009).
92. Johnson, S. G. The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt (2011).
93. Hughes, T. W., Minkov, M., Shi, Y. & Fan, S. Training of photonic neural networks through in situ backpropagation and gradient measurement. *Optica* **5**, 864–871 (2018).