

Quantum optical neural networks

Gregory R. Steinbrecher¹, Jonathan P. Olson², Dirk Englund¹, and Jacques Carolan^{1*}

¹Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA, USA 02139 and

²Zapata Computing Inc., 501 Massachusetts Ave., Cambridge, MA, USA 02139

(Dated: September 21, 2018)

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, and black-box quantum simulation. We consistently demonstrate our system can generalize from only a small set of training data onto states for which it has not been trained. Our results indicate 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.

I. INTRODUCTION

Deep learning is revolutionizing computing [1, 2] for an ever-increasing range of applications, from natural language processing [3] to particle physics [4] to cancer diagnosis [5]. These advances have been made possible by a combination of algorithmic design [6] and dedicated hardware development [7]. Quantum computing [8], 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 [9, 10], and the number and connectivity of qubits have improved [11, 12], making so-called Noisy Intermediate Scale Quantum (NISQ) processors [13] capable of tasks too hard for a classical computer a near-term prospect. Experimental progress has been met with algorithmic advances [14] and near-term quantum algorithms have been developed to tackle problems in combinatorics [15], quantum chemistry [16] and solid state physics [17]. However, it is only recently that the potential for quantum processors to accelerate machine learning has been explored [18].

Quantum machine learning algorithms for universal quantum computers have been proposed [19–21] and small-scale demonstrations implemented [22], though the requirements for practical protocols remain an open question [23]. Relaxing the requirement of universality, quantum machine learning for NISQ processors has emerged as a rapidly advancing field [24–27] 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.

In this work, we introduce an architecture for neural networks unique to quantum optical systems: the

Quantum Optical Neural Network (QONN). We argue that many of the features which are natural to quantum optics (mode mixing, optical nonlinearity) can directly be mapped to neural networks. Moreover, technological advances driven by trends in photonic quantum computing [28–30] and the microelectronics industry [31] offer a plausible route towards large-scale, high-bandwidth QONNs, all within a CMOS compatible platform.

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 data compression, with applications in quantum communication and quantum networks, which again relies on the ability to train our systems using a subset of possible inputs. Finally, we apply our system to a classical machine learning controls task, balancing an inverted pendulum by a reinforcement learning approach. Our results may find application both as an important technique for designing next generation quantum optical systems, as well as a versatile experimental platform for near-term optical quantum information processing and machine learning.

In prototypical neural networks [see Fig.1(a)] an input vector $\vec{x} \in \mathbb{R}^n$ is passed through multiple layers of: (1) linear transformation, i.e. 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 b_i (typically referred to as the perceptron or neuron, see Fig.1(a), 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 rela-

*carolanj@mit.edu

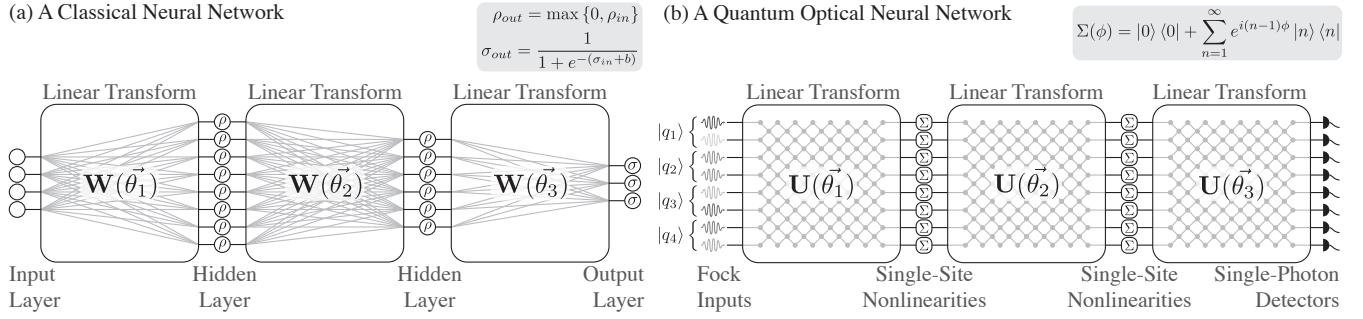


FIG. 1. Quantum Optical Neural Network. (a) An example of a classical neural network architecture. Hidden layers are rectified linear units (ReLU) and the output neuron uses a sigmoid activation function to map the output into the range $(0, 1)$. (b) An example of our quantum optical neural network (QONN) architecture. Inputs are dual-rail Fock states which encode qubits, with a photon in the top mode representing $|0\rangle$ and a photon in the bottom mode representing $|1\rangle$. The single-site nonlinearities are given by $\chi^{(3)}$ functions: a Kerr-type interaction applying a constant phase for each additional photon present. Readout is given by single photon detectors which measure the photon number at each output mode.

tionship 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 [32]. Once the network is trained, it may categorize with high probability new, unlabelled, images of cells as either ‘benign’ or ‘malignant’.

A number of the key components of classical neural networks 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 beam splitters and programmable phase shifts [33, 34]. In the lossless case, an n -mode optical circuit comprising $n(n-1)$ components implements an arbitrary $n \times n$ single particle unitary operation (which can also be used for classical neural networks [35, 36]), and a n -dimensional non-unitary operation can always be embedded across a $2n$ -mode optical circuit [37]. Advances in integrated optics have enabled the implementation of such circuits for applications in quantum computation [38], quantum simulation [39, 40], and classical optical neural networks [41]. Second, optical nonlinearities are a core component of many classical [42, 43] and quantum [44, 45] optical computing architectures. Single photon coherent nonlinearities can be implemented via measurement [44], interaction with three-level atoms [46] or superconducting materials [47], and through all-optical phenomena such as the Kerr effect [48]. While integration of each of these technologies into a single scalable system is an outstanding challenge for the field, for generality, the architecture we present considers idealized components. 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 [49].

II. ARCHITECTURE

As shown in Fig. 1(b), input data to our QONN is encoded as photonic quantum states $|\psi_i\rangle$, either as dual rail qubits requiring two optical modes per photon ($|0\rangle \equiv |10\rangle_{12}, |1\rangle \equiv |01\rangle_{12}$), or more generally as a Fock states $|i\rangle_j$ (corresponding to i photons in the j^{th} optical mode), which for n photons in m modes is described by a $\binom{n+m-1}{m}$ -dimensional complex vector of unit magnitude. 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., [33]. The nonlinear layer Σ comprises single mode $\chi^{(3)}$ interactions in the monochromatic approximation, applying a constant phase for each additional photon present via self-phase modulation [48]. For a given interaction strength ϕ , this unitary can be expressed as $\Sigma(\phi) = \sum_{n=1}^{\infty} |0\rangle \langle 0| + e^{i(n-1)\phi} |n\rangle \langle n|$. The full system comprising N layers is therefore

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

where $\vec{\Theta}$ is a $Nm(m-1)$ -dimensional vector and the strength of the nonlinearity is typically fixed as $\phi = \pi$. Finally, single photon detectors will be used to measure the photon number at each output. We use the results of this measurement, along with a training set of K desired input/output pairs $\{|\psi_{\text{in}}^i\rangle \rightarrow |\psi_{\text{out}}^i\rangle\}_{i=1}^K$, to construct a cost function

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

that is variationally minimized over $\vec{\Theta}$.

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

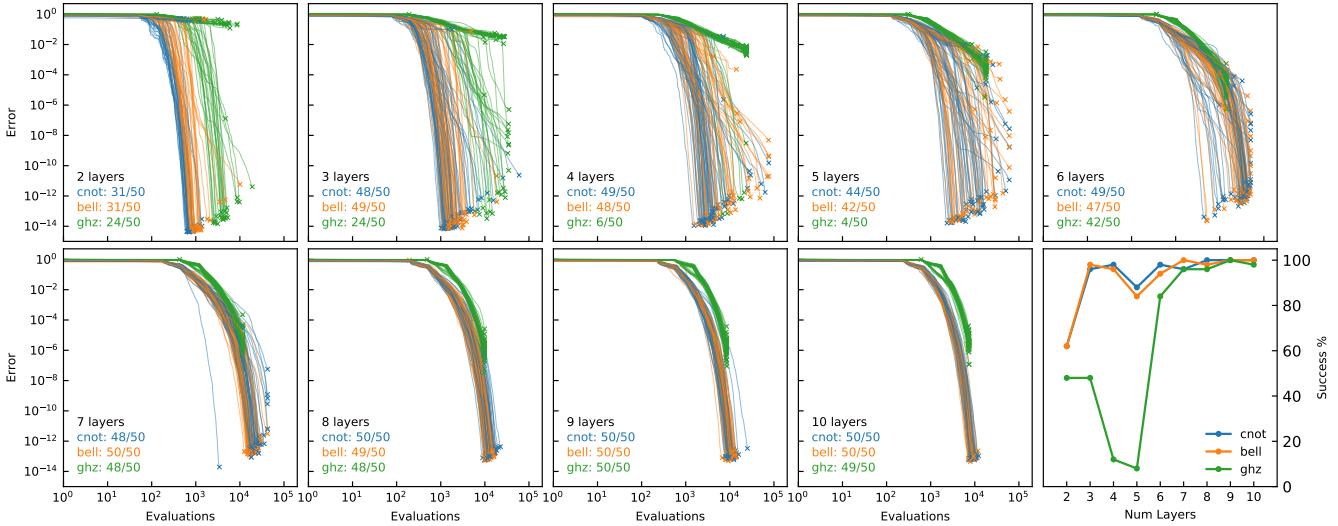


FIG. 2. **Benchmarking Results.** The first nine figures show 50 training runs for each of three representative optical quantum computing tasks: performing a CNOT gate, separating/generating Bell states, and generating GHZ states. At low layer depth, the optimizations frequently fail to converge to an optimal value (we defined an error less than 10^{-4} as “success”), terminating at relatively large errors. This behavior gets worse as we add layers, out to 5 layers, at which point it undergoes a rapid reversal, with the training essentially always succeeding at layer depths of 7 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.

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 [50]). If the target state is 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 [51]. Efficient fidelity proxies provide another route towards estimating salient features of quantum states without reconstruction of the full density matrix [52]. Moreover, the in situ approach may enable a form of error mitigation by routing quantum information around faulty hardware [53]. 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 Appendix A, we describe the computational techniques used in this work.

III. 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, GHZ state generation, and the implementation of the CNOT gate as representative of typical optical quantum information tasks. As described in Appendix B, 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 a CNOT gate, with 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.

IV. 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 $\{|\psi_{\text{in}}^i\rangle\} \{|\psi_{\text{out}}^i\rangle\}$ related by the Hamiltonian of interest $|\psi_{\text{out}}^i\rangle = \exp(-i\hat{H}t)|\psi_{\text{in}}^i\rangle$, and test it on new states

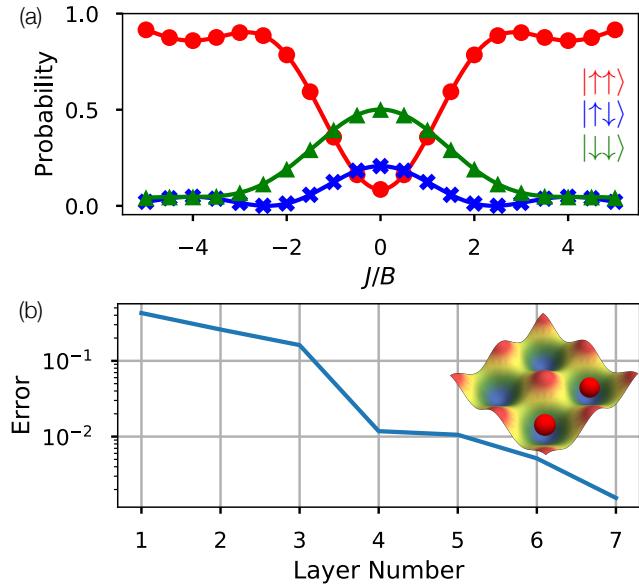


FIG. 3. QONNs for Hamiltonian Simulation. (a) Ising Model. A three layer QONN is trained for a range of interaction strengths J/B and the probability for particular output spin configuration is plotted (points) given the $|\uparrow\uparrow\rangle$ initialization state. The expected evolution is plotted 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 $U/t_{\text{hop}} = 20$ Bose-Hubbard Hamiltonian (schematic shown in inset). Training is performed 20 times for each layer depth, and the lowest test error is recorded.

which it has not been exposed to.

As a first test we look at the Ising model (see Appendix C) 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. 3(a) we vary the interaction strength $J/B \in [-5, 5]$ 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 [54] or layer-wise training approaches [55, 56] to efficiently train deeper QONN.

Finally we look at a Hamiltonian more natural for photons in optical modes, the Bose-Hubbard model, see Appendix C 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 with $U/t_{\text{hop}} = 20$ and time parameter $t = 1$. We constrain the connectivity to be that of a square lattice, as shown in Fig. 3(b), inset. Figure. 3(b) shows that the linear (i.e. single-layer) system gives a mean error in the test set of 42% and increasing the layer number steadily reduces this error to 0.1% at seven layers. This suggests that deeper networks can express a richer class of quantum functions (i.e. Hamiltonians), a concept familiar in classical deep neural networks [57]. Choosing five layers to give a reasonable trade-off between error ($\sim 1\%$) and computational tractability, we vary the interaction strength in the the range $U/t_{\text{hop}} \in [-20, 20]$. Across all experiments we achieve a mean test error of $2.9 \pm 1.3\%$ (error given by the standard deviation in 22 experiments).

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.

V. 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 [58]. However, such schemes are exponentially sensitive to loss: given a channel transmissivity η and 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 [59].

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 [60]. 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 [61]. 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, H₂, in the STO-3G minimal basis set [62], mapped from their fermionic representation into qubits via the Jordan-Wigner transformation [63]. Ground states in this qubit basis (which we will denote with a subscript as the logical basis L) 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 network consists of $n = 4$ photons in $m = 8$ optical modes.

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

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

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. 4(a,b), we test three training schemes for the QONN autoencoder. (1) local-structured training [Fig. 4(a), blue]: sequentially optimizing 2-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) global-structured training [Fig. 4(a), orange]: where the above layer structure is trained simultaneously rather than sequentially. (3) global-unstructured training [Fig. 4(b), green]: where a 6-layer system acting on all four qubits is trained.

The optimization was performed using an implementation of MLSL [64] (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 H₂ corresponding to bond lengths of 0.5, 1.0, 1.5, and 2.0 angstroms. Both structured optimizations performed comparably, converging to a fidelity of 92.0%. However, we note that the iterative approach could potentially be made more efficient if more stringent convergence criteria were introduced. The unstructured optimization achieved a lower fidelity of 57.9%. It is unclear from our data whether the iterative approach would have better scaling or accuracy than the global optimization in an asymptotic setting.

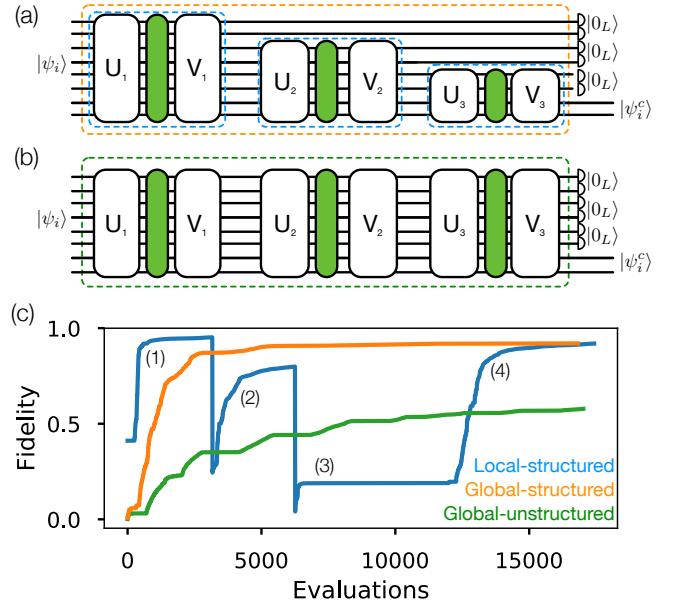


FIG. 4. Quantum Optical Autoencoder. (a, b) Schematics of the QONN architectures corresponding to each of the three training strategies. While the architecture of the global-structured (a, orange) and global-unstructured (b, green) optimizations remained the same throughout the entire optimization, the local-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. (b) 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.

VI. QUANTUM REINFORCEMENT LEARNING

Finally, 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 [65]. Classical deep reinforcement learning uses a policy network, i.e. 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.

We simulate a cart moving on a one dimensional frictionless track, with a pole on a hinge attached to its top (see Fig. 5. inset). At the beginning of the simulation, the cart is initialized to a random position, with the pole at a random angle. At each timestep, the neural net-

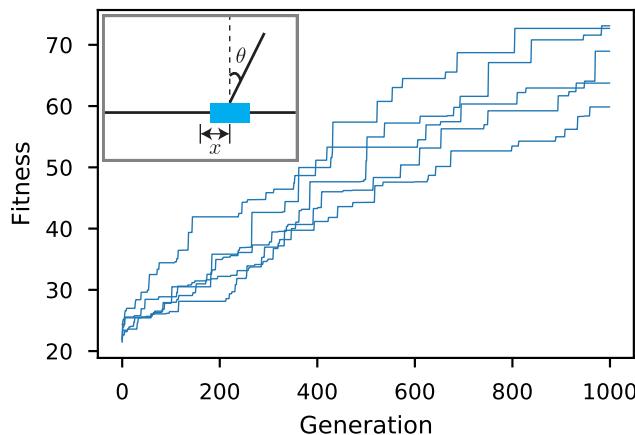


FIG. 5. Quantum Reinforcement Learning. Fitness vs. training generation curves for five different training runs of the reinforcement learning QONN. A higher fitness corresponds to a network that was able to keep the pole upright and the cart within the bounds for more time. Input data to the QONN was encoded onto four qubits. Inset: The problem we are trying to solve, a cart on a bounded one-dimensional track with an inverted pendulum attached to the top.

work 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.

To train a QONN to perform this task, we first encode the four values x , \dot{x} , θ , and $\dot{\theta}$ onto four qubits. We do this by compressing the values for these each into the range $\gamma \in [0, \pi/2]$ and setting the respective input qubit to $\cos(\gamma)|0\rangle + \sin(\gamma)|1\rangle$. We then pass these four photons through a QONN and, at the output, use the first two modes to select an action. 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 [66].

In Fig. 5 we show the results of five training cycles (each with different starting conditions) using a 6-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. Fitness increases with training generation, meaning the QONN consistently learns to balance the pole for longer as time increases: generalizing examples it has previously seen to new instances of the problem.

To cross-check our performance we trained equiva-

lently sized classical networks, i.e. 4-neuron, 6-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 66.4 for the QONN. 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 [67].

VII. 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 nano-fabrication have enabled monolithically integrated circuits with many thousands of optoelectronic components [68], and a feasible route towards large-scale single photon readout [69]. The architecture we present is not limited to the integration of systems with strong single photon nonlinearities, although promising progress has been made towards solid-state waveguide-based nonlinearities [70, 71]. Rather, we anticipate our approach will serve as a natural intermediate step 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 [72]. Future work will likely focus on loss correction techniques, which are also possible in an all optical context [73]. 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.

ACKNOWLEDGMENTS

This work was supported by the AFOSR MURI for Optimal Measurements for Scalable Quantum Technologies (FA9550-14-1-0052) and by the AFOSR program FA9550-16-1-0391, supervised by Gernot Pomerenke. G.R.S. acknowledges support from the Facebook Fellowship Program. J.C. is supported by EU H2020 Marie Skłodowska-Curie grant number 751016.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

Appendix A: 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, which involves the calculation of the permanent of $\binom{n+m-1}{n}^2$ matrices of dimension $n \times n$ [74].

As with classical neural networks, different optimization algorithms perform better for different tasks. We rely on gradient-free optimization techniques, as computing and backpropagating the gradient through the system likely requires knowledge of the internal quantum state of the system, preventing efficient training. While this might be acceptable for designing small systems in simulation (say, designing quantum gates), it doesn't allow for systems to be variationally trained in situ. We empirically determined that the BOBYQA algorithm [75] performs well for most applications in terms of speed and accuracy for our QONN, and is available in the NLOpt library [76]. For the quantum reinforcement learning experiments, we used our own implementation of evolutionary strategies [66].

The computer used to perform these simulations is a custom-built workstation with a 12-core Intel Core i7-5820K and 64GB 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 0.37.0.

Appendix B: Benchmarking Training

The training set for the Bell-state projector is the full set of Bell states $\{|\psi_{\text{in}}^i\rangle\} = \{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$ 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 $\{|\psi_{\text{out}}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0110\rangle, |0101\rangle\}$. 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 $\{|\psi_{\text{in}}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0110\rangle, |0101\rangle\}$ and $\{|\psi_{\text{out}}^i\rangle\} = \{|1010\rangle, |1001\rangle, |0101\rangle, |0110\rangle\}$. For the GHZ generator we select just a single input-output configuration $\{|\psi_{\text{in}}^i\rangle\} = \{|101010\rangle\}$ and $\{|\psi_{\text{out}}^i\rangle\} = \{(|101010\rangle + |010101\rangle)/\sqrt{2}\}$.

Appendix C: Simulated Hamiltonians

The Ising model we simulate is described by the Hamiltonian

$$H_{\text{Ising}} = B \sum_i \hat{X}_i + J \sum_{\langle i,j \rangle} \hat{Z}_i \otimes \hat{Z}_j, \quad (\text{C1})$$

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

$$\hat{H}_{\text{BH}} = \omega \sum_i \hat{b}_i^\dagger \hat{b}_i - t_{\text{hop}} \sum_{\langle i,j \rangle} \hat{b}_i^\dagger \hat{b}_j + U/2 \sum_i \hat{n}_i (\hat{n}_i - 1), \quad (\text{C2})$$

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.

-
- [1] Y. LeCun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016).
- [3] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, (2016), arXiv:1609.08144v2.
- [4] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, *Nature* **560**, 41 (2018).
- [5] D. Capper, D. T. W. Jones, M. Sill, V. Hovestadt, D. Schrimpf, D. Sturm, C. Koelsche, F. Sahm, L. Chavez, D. E. Reuss, *et al.*, *Nature* **555**, 469 (2018).
- [6] X. Glorot, A. Bordes, and Y. Bengio, in *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (2011) pp. 315–323.
- [7] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, *Proceedings of the IEEE* **105**, 2295 (2017).
- [8] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
- [9] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O’Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov, A. N. Cleland, and J. M. Martinis, *Nature* **508**, 500 (2014).
- [10] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, *Phys. Rev. Lett.* **113**, 220501 (2014).
- [11] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, *Nature* **551**, 579 (2017).

- [12] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe, *Nature* **551**, 601 (2017).
- [13] J. Preskill, *Quantum* **2**, 79 (2018).
- [14] A. Montanaro, *npj Quantum Information* , 15023 (2016).
- [15] E. Farhi, J. Goldstone, and S. Gutmann, (2014), arXiv:1411.4028v1.
- [16] A. Aspuru-Guzik and P. Walther, *Nat. Phys.* **8**, 285 (2012).
- [17] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, *Phys. Rev. A* **92**, 062318 (2015).
- [18] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature* **549**, 195 (2017).
- [19] A. W. Harrow, A. Hassidim, and S. Lloyd, *Phys. Rev. Lett.* **103**, 150502 (2009).
- [20] S. Lloyd, M. Mohseni, and P. Rebentrost, *Nat. Phys.* **10**, 631 (2014).
- [21] P. Rebentrost, M. Mohseni, and S. Lloyd, *Phys. Rev. Lett.* **113**, 130503 (2014).
- [22] X. D. Cai, D. Wu, Z. E. Su, M. C. Chen, X. L. Wang, L. Li, N. L. Liu, C. Y. Lu, and J. W. Pan, *Phys. Rev. Lett.* **114**, 110504 (2015).
- [23] S. Aaronson, *Nat. Phys.* **11**, 291 (2015).
- [24] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, (2018), arXiv:1803.00745v1.
- [25] E. Farhi and H. Neven, (2018), arXiv:1802.06002v1.
- [26] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, (2018), arXiv:1804.00633v1.
- [27] V. Havlicek, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, (2018), arXiv:1804.11326v2.
- [28] J. L. O'Brien, *Science* **318**, 1567 (2007).
- [29] J. L. O'Brien, A. Furusawa, and J. Vučković, *Nat. Photon.* **3**, 687 (2009).
- [30] T. Rudolph, *APL Photonics* **2**, 030901 (2017).
- [31] C. Sun, M. T. Wade, Y. Lee, J. S. Orcutt, L. Alloatti, M. S. Georgas, A. S. Waterman, J. M. Shainline, R. R. Avizienis, S. Lin, B. R. Moss, R. Kumar, F. Pavanello, A. H. Atabaki, H. M. Cook, A. J. Ou, J. C. Leu, Y.-H. Chen, K. Asanović, R. J. Ram, M. A. Popović, and V. M. Stojanović, *Nature* **528**, 534 (2015).
- [32] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, *CSBJ* **13**, 8 (2015).
- [33] M. Reck, A. Zeilinger, H. J. Bernstein, and H. Bertani, *Phys. Rev. Lett.* **73**, 58 (1994).
- [34] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, *Optica* **3**, 1460 (2016).
- [35] M. Arjovsky, A. Shah, and Y. Bengio, in *International Conference on Machine Learning* (2016) pp. 1120–1128.
- [36] L. Jing, Y. Shen, J. Peurifoy, S. Skirlo, Y. LeCun, and M. Tegmark, (2016), arXiv:1612.05231v3.
- [37] D. A. B. Miller, *Photon. Res., PRJ* **1**, 1 (2013).
- [38] J. Carolan, C. Harrold, C. Sparrow, E. Martín López, N. J. Russell, J. W. Silverstone, P. J. Shadbolt, N. Matsuda, M. Oguma, M. Itoh, G. D. Marshall, M. G. Thompson, J. C. F. Matthews, T. Hashimoto, J. L. O'Brien, and A. Laing, *Science* **349**, 711 (2015).
- [39] B. P. Lanyon, J. D. Whitfield, G. G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, A. Aspuru-Guzik, and A. G. White, *Nat. Chem.* **2**, 106 (2010).
- [40] C. Sparrow, E. Martín López, N. Maraviglia, A. Neville, C. Harrold, J. Carolan, Y. N. Joglekar, T. Hashimoto, N. Matsuda, J. L. O'Brien, D. P. Tew, and A. Laing, *Nature* **557**, 660 (2018).
- [41] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, and M. Soljačić, *Nat. Photon.* **11**, 441 (2017).
- [42] D. A. B. Miller, *Proceedings of the IEEE* **97**, 1166 (2009).
- [43] D. A. B. Miller, *Nat. Photon.* **4**, 3 (2010).
- [44] E. Knill, R. Laflamme, and G. Milburn, *Nature* **409**, 46 (2001).
- [45] P. Kok, W. J. Munro, K. Nemoto, T. C. Ralph, J. P. Dowling, and G. J. Milburn, *Rev. Mod. Phys.* **79**, 135 (2007).
- [46] L. M. Duan and H. J. Kimble, *Phys. Rev. Lett.* **92**, 125 (2004).
- [47] G. Kirchmair, B. Vlastakis, Z. Leghtas, S. E. Nigg, H. Paik, E. Ginossar, M. Mirrahimi, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, *Nature* **495**, 205 (2013).
- [48] D. J. Brod and J. Combes, *Phys. Rev. Lett.* **117**, 080502 (2016).
- [49] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, arXiv:1806.06871 (2018).
- [50] J. L. O'Brien, G. Pryde, A. Gilchrist, D. F. V. James, N. K. Langford, T. C. Ralph, and A. G. White, *Phys. Rev. Lett.* **93**, 080502 (2004).
- [51] J. C. Garcia-Escartin and P. Chamorro-Posada, *Phys. Rev. A* **87**, 052330 (2013).
- [52] M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu, *Nat. Comms.* **1**, 149 (2010).
- [53] J. Mower, N. C. Harris, G. R. Steinbrecher, Y. Lahini, and D. Englund, *Phys. Rev. A* **92**, 032322 (2015).
- [54] G. Verdon, J. Pye, and M. Broughton, (2018), arXiv:1806.09729v1.
- [55] Y. Bengio, P. Lamblin, D. Popovic, and H. Larochelle, *Greedy layer-wise training of deep networks* (*Advances in Neural Information Processing*, 2007).
- [56] C. Hettinger, T. Christensen, B. Ehlert, J. Humpherys, T. Jarvis, and S. Wade, (2017), arXiv:1706.02480v1.
- [57] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, (2016), arXiv:1606.05336v6.
- [58] N. Gisin and R. Thew, *Nat. Photon.* **1**, 165 (2007).
- [59] H. J. Kimble, *Nature* **453**, 1023 (2008).
- [60] J. Romero, J. P. Olson, and A. Aspuru-Guzik, *Quantum Science and Technology* **2**, 023023 (2017).
- [61] J. Olson, S. Sim, and Y. Cao, https://github.com/zapatacomputing/cusp_cirq_demo (2018).
- [62] T. Helgaker, J. Olsen, and P. Jorgensen, *Phys. Today* (Wiley, 2013) p. 944.
- [63] A. Tranter, S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, *International Journal of Quantum Chemistry* **115**, 1431 (2015).
- [64] A. H. G. R. Kan and G. T. Timmer, *Mathematical Programming* **39**, 27 (1987).
- [65] A. G. Barto, R. S. Sutton, and C. W. Anderson, *IEEE transactions on systems, man, and cybernetics* , 834 (1983).
- [66] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, arXiv:1703.03864 (2017).
- [67] D. Ristè, M. P. da Silva, C. A. Ryan, A. W. Cross, A. D. Córcoles, J. A. Smolin, J. M. Gambetta, J. M. Chow, and B. R. Johnson, *npj Quantum Information* **3**, 16 (2017).
- [68] S. Chung, H. Abediasl, and H. Hashemi, *IEEE J. Solid-State Circuits* **53**, 275 (2017).

- [69] D. Zhu, Q.-Y. Zhao, H. Choi, T.-J. Lu, A. E. Dane, D. Englund, and K. K. Berggren, *Nature Nanotechnology* **13**, 596 (2018).
- [70] A. Javadi, I. Söllner, M. Arcari, S. L. Hansen, L. Midolo, S. Mahmoodian, G. Kiršanské, T. Pregnolato, E. H. Lee, J. D. Song, S. Stobbe, and P. Lodahl, *Nat. Comms.* **6**, 8655 (2015).
- [71] S. Sun, H. Kim, Z. Luo, G. S. Solomon, and E. Waks, *Science* **361**, 57 (2018).
- [72] J. H. Shapiro, *Phys. Rev. A* **73**, 1502 (2006).
- [73] M. Y. Niu, I. L. Chuang, and J. H. Shapiro, *Phys. Rev. A* **97**, 032323 (2018).
- [74] S. Scheel, (2004), arXiv:0406127v1.
- [75] M. J. Powell, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge , 26 (2009).
- [76] S. G. Johnson, <http://ab-initio.mit.edu/nlopt> (2011).