Quantum Graph Neural Networks

Guillaume Verdon
X, The Moonshot Factory &
Google Research
Mountain View, CA
gverdon@x.team

Trevor McCourt
Google Research
Venice, CA
trevormccrt@google.com

Stefan Leichenauer, Enxhell Luzhinca,
Vikash Singh, Jack Hidary
X, The Moonshot Factory
Mountain View, CA
{sleichenauer,enxhell,
singvikash,hidary}@x.team

Abstract

We introduce Quantum Graph Neural Networks (QGNN), a new class of quantum neural network ansatze which are tailored to represent quantum processes which have a graph structure, and are particularly suitable to be executed on distributed quantum systems over a quantum network. Along with this general class of ansatze, we introduce further specialized architectures, namely, Quantum Graph Recurrent Neural Networks (QGRNN) and Quantum Graph Convolutional Neural Networks (QGCNN). We provide three example applications of QGNNs: learning Hamiltonian dynamics of quantum systems, learning how to create multipartite entanglement in a quantum network, and unsupervised learning for spectral clustering.

1 Introduction

Variational Quantum Algorithms are a promising class of algorithms are rapidly emerging as a central subfield of Quantum Computing [1, 2, 3]. Similar to parameterized transformations encountered in deep learning, these parameterized quantum circuits are often referred to as Quantum Neural Networks (QNNs). Recently, it was shown that QNNs that have no prior on their structure suffer from a quantum version of the no-free lunch theorem [4] and are exponentially difficult to train via gradient descent. Thus, there is a need for better QNN ansatze. One popular class of QNNs has been Trotter-based [2, 5]. The optimization of these ansatze has been extensively studied in recent works, and efficient optimization methods have been found [6] . On the classical side, graph-based neural networks leveraging data geometry for have seen some recent successes in deep learning, finding applications in biophysics and chemistry [7]. Inspired from this success, we propose a new class of Quantum Neural Network ansatz which allows for both quantum inference and classical probabilistic inference for data with a graph-geometric structure. In the sections below, we introduce the general framework of the QGNN ansatz as well as several more specialized variants and showcase three potential applications via numerical implementation.

2 Quantum Graph Neural Networks

Networked Quantum Systems Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the set of vertices (or nodes) and $\mathcal{E}$ the set of edges. We can assign a quantum subsystem with Hilbert space $\mathcal{H}_v$ for each

Second Workshop on Machine Learning and the Physical Sciences (NeurIPS 2019), Vancouver, Canada.

vertex in the graph, forming a global Hilbert space $\mathcal{H}_\mathcal{V} \equiv \bigotimes_{v\in\mathcal{V}} \mathcal{H}_v$. Each of the vertex subsystems could be one or several qubits, a qudit, a qumode [8], or even an entire quantum computer.[1] The edges of the graph dictate the communication between the vertex subspaces: couplings between degrees of freedom on two different vertices are allowed if there is an edge connecting them. This setup is called a quantum network [9, 10] with topology given by the graph $\mathcal{G}$.

General Quantum Graph Neural Network Ansatz The most general Quantum Graph Neural Network ansatz is a parameterized quantum circuit on a network which consists of a sequence of Q different Hamiltonian evolutions, with the whole sequence repeated P times:

$$U_{\text{QGNN}}(\boldsymbol{\eta}, \boldsymbol{\theta}) = \prod_{p=1}^{P} \left[\prod_{q=1}^{Q} e^{-i\eta_{pq}\hat{H}_q(\boldsymbol{\theta})} \right], \tag{1}$$

where the product is time-ordered [11], the $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$ are variational (trainable) parameters, and the Hamiltonians $\hat{H}_q(\boldsymbol{\theta})$ can generally be any parameterized Hamiltonians whose topology of interactions is that of the problem graph:

$$\hat{H}_q(\boldsymbol{\theta}) \equiv \sum_{\{j,k\}\in\mathcal{E}} \sum_{r\in\mathcal{I}_{jk}} W_{qrjk} \hat{O}_j^{(qr)} \otimes \hat{P}_k^{(qr)} + \sum_{v\in\mathcal{V}} \sum_{r\in\mathcal{J}_v} B_{qrv} \hat{R}_j^{(qv)}. \tag{2}$$

Here the W_{qrjk} and B_{qrv} are real-valued coefficients which can generally be independent trainable parameters, forming a collection $\boldsymbol{\theta} \equiv \cup_{q,j,k,r}\{W_{qrjk}\} \cup_{q,v,r} \{B_{qrjk}\}$. The operators $\hat{R}_j^{(qv)}, \hat{O}_j^{(qr)}, \hat{P}_j^{(qr)}$ are Hermitian operators which act on the Hilbert space of the j^{th} node of the graph. The sets $\mathcal{I}_{jk}$ and $\mathcal{J}_v$ are index sets for the terms corresponding to the edges and nodes, respectively. To make compilation easier, we enforce that the terms of a given Hamiltonian $\hat{H}_q$ commute with one another, but different $\hat{H}_q$s need not commute.

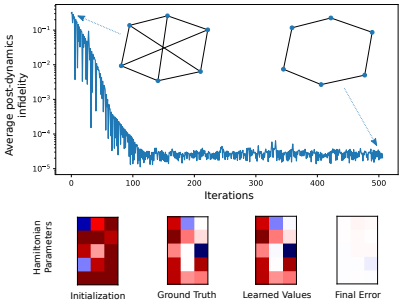
In order to make the ansatz more amenable to training and avoid the barren plateaus/no free lunch problem [4], we need to add some constraints and specificity. To that end, we now propose more specialized architectures where parameters are tied spatially (convolutional) or tied over the sequential iterations of the exponential mapping (recurrent).

Quantum Graph Recurrent Neural Networks (QGRNN) We define quantum graph recurrent neural networks as ansatze of the form of (1) where the temporal parameters are tied between iterations, $\eta_{pq} \mapsto \eta_q$. In other words, we have tied the parameters between iterations of the outer sequence index (over $p = 1, \ldots, P$). This is akin to classical recurrent neural networks where parameters are shared over sequential applications of the recurrent neural network map. As η_q acts as a time parameter for Hamiltonian evolution under $\hat{H}_q$, we can view the QGRNN ansatz as a Trotter-based [12, 11] quantum simulation of an evolution $e^{-i\Delta\hat{H}_{\text{eff}}}$ under the Hamiltonian $H_{\text{eff}} = \Delta^{-1}\sum_q \eta_q \hat{H}_q$ for a time step of size $\Delta = \|\boldsymbol{\eta}\|_1 = \sum_q |\eta_q|$. This ansatz is thus specialized to learn effective dynamics on a graph. In Section 3 we demonstrate this by learning the effective real-time dynamics of an Ising model on a graph using a QGRNN ansatz.

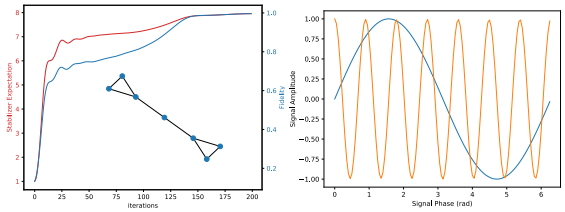
Quantum Graph Convolutional Neural Networks (QGCNN) Classical Graph Convolutional neural networks rely on a key feature: that of permutation invariance. In other words, the ansatz should be invariant under permutation of the nodes. This is analogous to translational invariance for ordinary convolutional transformations. In our case, permutation invariance manifests itself as a constraint on the Hamiltonian, which now should be devoid of *local* trainable parameters, and should only have global trainable parameters. So the $\boldsymbol{\theta}$ parameters become tied over indices of the graph: $W_{qrjk} \mapsto W_{qr}$ and $B_{qrv} \mapsto B_{qr}$. A broad class of graph convolutional neural networks we will focus on is the set of so-called Quantum Alternating Operator Ansatze [5], the generalized form of the Quantum Approximate Optimization Algorithm ansatz [2].

Quantum Spectral Graph Convolutional Neural Networks (qsgcnn) We can take inspiration from the continuous-variable quantum approximate optimization ansatz introduced in [13] to create

[1] One may also define a Hilbert space for each edge and form $\mathcal{H}_\mathcal{E} \equiv \bigotimes_{e\in\mathcal{E}} \mathcal{H}_e$. The total Hilbert space for the graph would then be $\mathcal{H}_\mathcal{E} \otimes \mathcal{H}_\mathcal{V}$. For the sake of simplicity and feasibility of numerical implementation, we consider this to be beyond the scope of the present work.



(1a) Top: Batch average infidelity $(1 - F)$ with respect to ground truth state sampled at 15 randomly chosen times. We see the initial guess has a densely connected topology and the QGRNN learns the ring structure of the true Hamiltonian. Bottom: Ising Hamiltonian parameters (weights & biases) on a color scale.



(1b) Left: Stabilizer Hamiltonian expectation and fidelity over training iterations. A picture of the quantum network topology is inset. Right: Quantum phase kickback test on the learned GHZ state. We observe a 7x boost in Rabi oscillation frequency for a 7-node network, thus demonstrating we have reached the Heisenberg limit of sensitivity for the quantum sensor network.

a variant of the QGCNN: the Quantum Spectral Graph Convolutional Neural Network (QSGCNN). We show here how it recovers the mapping of Laplacian-based graph convolutional networks [14] in the Heisenberg picture, consisting of alternating layers of message passing, node update, and nonlinearities.

Consider an ansatz of the form from (1) with four different Hamiltonians ($Q = 4$) for a given graph. First, for a weighted graph $\mathcal{G}$ with edge weights Λ_{jk}, we define the *coupling Hamiltonian* as $\hat{H}_C \equiv \frac{1}{2}\sum_{\{j,k\}\in\mathcal{E}} \Lambda_{jk}(\hat{x}_j - \hat{x}_k)^2$. The Λ_{jk} here are the *weights* of the graph $\mathcal{G}$, and are *not* trainable parameters. The operators denoted here by $\hat{x}_j$ are quantum continuous-variable position operators, which can be implemented via continuous-variable (analog) quantum computers [8] or emulated using multiple qubits on digital quantum computers [15, 16]. After evolving by $\hat{H}_C$, which we consider to be the message passing step, one applies an exponential of the *kinetic* Hamiltonian, $\hat{H}_K \equiv \frac{1}{2}\sum_{j\in\mathcal{V}} \hat{p}_j^2$. Here $\hat{p}_j$ denotes the continuous-variable momentum (Fourier conjugate) of the position, obeying the canonical commutation relation $[\hat{x}_j, \hat{p}_j] = i\delta_{jk}$. We consider this step as a node update step. In the Heisenberg picture, the evolution generated by these two steps maps the position operators of each node according to $e^{-i\gamma\hat{H}_K}e^{-i\alpha\hat{H}_C} : \hat{x}_j \mapsto \hat{x}_j + \gamma\hat{p}_j - \alpha\gamma\sum_{k\in\mathcal{V}} L_{jk}\hat{x}_k$, where $L_{jk} = \delta_{jk}\left(\sum_{v\in\mathcal{V}}\Lambda_{jv}\right) - \Lambda_{jk}$ is the *Graph Laplacian* matrix for the weighted graph $\mathcal{G}$. We can recognize this step as analogous to classical spectral-based graph convolutions. One difference to note here is that *momentum* is free to accumulate between layers.

Next, we must add some non-linearity in order to give the ansatz more capacity.The next evolution is thus generated by an *anharmonic* Hamiltonian $\hat{H}_A = \sum_{j\in\mathcal{V}} f(\hat{x}_j)$, where f is a nonlinear function of degree greater than 2, e.g., a quartic potential of the form $f(\hat{x}_j) = ((\hat{x}_j - \mu)^2 - \omega^2)^2$ for some μ, ω hyperparameters. Finally, we apply another evolution according to the kinetic Hamiltonian. These last two steps yield an update $e^{-i\beta\hat{H}_K}e^{-i\delta\hat{H}_A} : \hat{x}_j \mapsto \hat{x}_j + \beta\hat{p}_j - \delta\beta f'(\hat{x}_j)$, which acts as a nonlinear mapping. By repeating the four evolution steps described above in a sequence of P layers, i.e.,

$$\hat{U}_{\text{QSGCNN}}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}) = \prod_{j=1}^{P} e^{-i\beta_j\hat{H}_K}e^{-i\delta_j\hat{H}_A}e^{-i\gamma_j\hat{H}_K}e^{-i\alpha_j\hat{H}_C}$$

with variational parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma},\boldsymbol{\delta}\}$, we then recover a quantum-coherent analogue of the node update prescription of [14] in the original graph convolutional networks paper.

3 Applications & Experiments

Learning Quantum Dynamics with Quantum Graph Recurrent Neural Networks Learning the dynamics of a closed quantum system is a task of interest for many applications [17], including device characterization and validation. In this example, we demonstrate that a Quantum Graph Recurrent Neural Network can learn effective dynamics of an Ising spin system when given access to the output of quantum dynamics at various times.

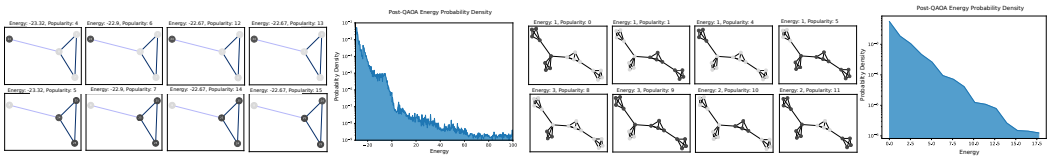


Figure 2: QSGCNN spectral clustering results for 5-qubit precision (left) with quartic double-well potential and 1-qubit precision (right) for different graphs. Weight values represented as opacity, output sampled node values as grayscale. Lower precision on the right allows for more nodes in the simulation. The graphs displayed are the most probable (populated) configurations, and to their right is the output probability distribution over potential energies. We see lower energies are most probable and that these configurations have node values clustered.

Our target is an Ising Hamiltonian on a particular graph, $\hat{H}_{\text{target}} = \sum_{\{j,k\}\in\mathcal{E}} J_{jk}\hat{Z}_j\hat{Z}_k + \sum_{v\in\mathcal{V}} Q_v\hat{Z}_v + \sum_{v\in\mathcal{V}}\hat{X}_j$. We are given copies of a low-energy state $|\psi_0\rangle$ as well as copies of the state $|\psi_T\rangle \equiv \hat{U}(T)\,|\psi_0\rangle = e^{-iT\hat{H}_{\text{target}}}$ for some known but randomly chosen times $T \in [0, T_{\text{max}}]$. Our goal is to learn the target Hamiltonian parameters $\{J_{jk}, Q_v\}_{j,k,v\in\mathcal{V}}$ by comparing the state $|\psi_T\rangle$ with the state obtained by evolving $|\psi_0\rangle$ according to the QGRNN ansatz for a number of iterations $P \approx T/\Delta$ (where Δ is a hyperparameter determining the Trotter step size). We achieve this by training the parameters via Adam [18] gradient descent on the average infidelity $\mathcal{L}(\boldsymbol{\theta}) = 1 - \frac{1}{B}\sum_{j=1}^{B} |\,\langle\psi_{T_j}|U^j_{\text{QGRNN}}(\Delta,\boldsymbol{\theta})\,|\psi_0\rangle\,\rangle\,|^2$ averaged over batch sizes of 15 different times T. The ansatz uses a Trotterization of a random densely-connected Ising Hamiltonian as its initial guess, and successfully learns the Hamiltonian parameters within a high degree of accuracy as shown in Fig. 1a.

Quantum Graph Convolutional Neural Networks for Quantum Sensor Networks Quantum Sensor Networks are a promising area of application for the technologies of Quantum Sensing and Quantum Networking/Communication [9, 10]. A common task considered where a quantum advantage can be demonstrated is the estimation of a parameter hidden in weak qubit phase rotation signals, such as those encountered when artificial atoms interact with a constant electric field of small amplitude [10]. A well-known method to achieve this advantange is via the use of a quantum state exhibiting multipartite entanglement of the Greenberger–Horne–Zeilinger kind, also known as a GHZ state [19]. Here we demonstrate that, without global knowledge of the quantum network structure, a QGCNN ansatz can learn to prepare a GHZ state. We use a QGCNN ansatz with $\hat{H}_1 = \sum_{\{j,k\}\in\mathcal{E}}\hat{Z}_j\hat{Z}_k$ and $\hat{H}_2 = \sum_{j\in\mathcal{V}}\hat{X}_j$. The loss function is the negative expectation of the sum of stabilizer group generators which stabilize the GHZ state [20], i.e., $\mathcal{L}(\boldsymbol{\eta}) = -\langle\bigotimes_{j=0}^{n}\hat{X} + \sum_{j=1}^{n-1}\hat{Z}_j\hat{Z}_{j+1}\rangle$ for a network of n qubits. Results are presented in Fig. 1b. Note that the advantage of using a QGNN ansatz on the network is that the number of quantum communication rounds is simply proportional to P, and that the local dynamics of each node are independent of the global network structure. As a further test of the validity of the GHZ state, we show the results of a quantum phase kickback test which demonstrates the quantum sensing signal sensitivity speedup [21].

Unsupervised Learning with Quantum Anharmonic Graph Convolutional Networks As a final set of applications, we consider applying the QSGCNN from Section 2 to the task of Spectral Clustering. Spectral clustering involves finding low-frequency eigenvalues of the graph Laplacian and clustering the node values in order to identify graph clusters. In Fig. 2 we present the results for a QSGCNN for varying multi-qubit precision for the representation of the continuous values, where the loss function that was minimized was the expected value of the anharmonic potential $\mathcal{L}(\boldsymbol{\eta}) = \langle\hat{H}_C + \hat{H}_A\rangle_{\boldsymbol{\eta}}$. Of particular interest to near-term quantum computing [22] is the single-qubit precision case, where we modify the QSGCNN construction as $\hat{p}_j \mapsto \hat{X}_j$, $\hat{H}_A \mapsto I$ and $\hat{H}_C \mapsto \frac{1}{2}\sum_{\{j,k\}\in\mathcal{E}}\lambda_{jk}(|1\rangle\langle 1|_j - |1\rangle\langle 1|_k)^2$, where $|1\rangle\langle 1|_k = (\hat{I} - \hat{Z}_k)/2$. We see that using a low-qubit precision yields sensible results, thus implying that spectral clustering could be a promising new application for near-term quantum devices.

4 Discussion & Conclusion

Results featured in this paper should be viewed as a promising set of first explorations of the potential applications of QGNNs. Through our numerical experiments, we have shown the use of these QGNN ansatze in the context of quantum dynamics learning, quantum sensor network optimization, and unsupervised graph clustering. Given that there is a vast set of literature on the use of Graph Neural Networks and their variants to quantum chemistry, future works should explore hybrid methods where one can learn a graph-based hidden quantum representation (via a QGNN) of a quantum chemical process. As the true underlying process is quantum in nature and has a natural molecular graph geometry, the QGNN could serve as a more accurate model for the hidden processes which lead to perceived emergent chemical properties. We seek to explore this in future work. Other future work could include a benchmark of the QSGCNN for the graph isomorphism problem [23], generalizing the QGNN to include quantum degrees of freedom on the edges, include quantum-optimization-based training of the graph parameters via quantum phase backpropagation [16], and extending the QSGCNN to multiple features per node.

Acknowledgements

Numerics in this paper were executed using a custom interface between Google's Cirq [24] and TensorFlow [25]. The authors would like to thank Edward Farhi, Jae Yoo, and Li Li for useful discussions. GV, EL, and VS would like to thank the team at X for the hospitality and support during their respective Quantum@X and AI@X residencies where this work was completed. X, formerly known as Google[x], is part of the Alphabet family of companies, which includes Google, Verily, Waymo, and others (www.x.company). GV acknowledges funding from NSERC.

References

[1] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.

[2] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[3] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[4] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.

[5] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.

[6] Guillaume Verdon, Michael Broughton, Jarrod R McClean, Kevin J Sung, Ryan Babbush, Zhang Jiang, Hartmut Neven, and Masoud Mohseni. Learning to learn with quantum neural networks via classical neural networks. *arXiv preprint arXiv:1907.05415*, 2019.

[7] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[8] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J Cerf, Timothy C Ralph, Jeffrey H Shapiro, and Seth Lloyd. Gaussian quantum information. *Reviews of Modern Physics*, 84(2):621, 2012.

[9] H Jeff Kimble. The quantum internet. *Nature*, 453(7198):1023, 2008.

[10] Kevin Qian, Zachary Eldredge, Wenchao Ge, Guido Pagano, Christopher Monroe, James V Porto, and Alexey V Gorshkov. Heisenberg-scaling measurement protocol for analytic functions with quantum sensor networks. *arXiv preprint arXiv:1901.09042*, 2019.

[11] David Poulin, Angie Qarry, Rolando Somma, and Frank Verstraete. Quantum simulation of time-dependent hamiltonians and the convenient illusion of hilbert space. *Physical review letters*, 106(17):170501, 2011.

[12] Seth Lloyd. Universal quantum simulators. *Science*, pages 1073–1078, 1996.

[13] Guillaume Verdon, Juan Miguel Arrazola, Kamil Brádler, and Nathan Killoran. A quantum approximate optimization algorithm for continuous problems. *arXiv preprint arXiv:1902.00409*, 2019.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[15] Rolando D Somma. Quantum simulations of one dimensional quantum systems. *arXiv preprint arXiv:1503.06319*, 2015.

[16] Guillaume Verdon, Jason Pye, and Michael Broughton. A universal training algorithm for quantum deep learning. *arXiv preprint arXiv:1806.09729*, 2018.

[17] Nathan Wiebe, Christopher Granade, Christopher Ferrie, and David G Cory. Hamiltonian learning and certification using quantum resources. *Physical review letters*, 112(19):190501, 2014.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Daniel M Greenberger, Michael A Horne, and Anton Zeilinger. Going beyond bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[20] Géza Tóth and Otfried Gühne. Entanglement detection in the stabilizer formalism. *Physical Review A*, 72(2):022340, 2005.

[21] Ken X Wei, Isaac Lauer, Srikanth Srinivasan, Neereja Sundaresan, Douglas T McClure, David Toyli, David C McKay, Jay M Gambetta, and Sarah Sheldon. Verifying multipartite entangled ghz states via multiple quantum coherences. *arXiv preprint arXiv:1905.05720*, 2019.

[22] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[23] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[24] Cirq: A Python framework for creating, editing, and invoking noisy intermediate scale quantum (NISQ) circuits. `https://github.com/quantumlib/Cirq`.

[25] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.