Survey on Hybrid Classical-Quantum Machine Learning Models

Maha A. Metawei
Computers and Systems Eng. Department,
Faculty of Engineering, Ain Shams University
Cairo, Egypt
maha_metawei@eri.sci.eg

Hazem Said
Computers and Systems Eng. Department,
Faculty of Engineering, Ain Shams University
Cairo, Egypt
hazem.said@eng.asu.edu.eg

Mohamed Taher
Computers and Systems Eng. Department,
Faculty of Engineering, Ain Shams University
Cairo, Egypt
mohamed.taher@eng.asu.edu.eg

Hesham Eldeib
Computers and Systems Department,
Electronics Research Institute,
Giza, Egypt
eldeeb@eri.sci.eg

Salwa M. Nassar
Computers and Systems Department,
Electronics Research Institute,
Giza, Egypt
salwa@eri.sci.eg

***Abstract*—Optimizing for a classification problem with a 0-1 loss function is an NP-hard problem [1] even for a simple binary classification task. As Preskill stated "We don't expect a quantum computer to solve worst case instances of NP-hard problems, but it might find better approximate solution or find it faster " [2]. Hybrid Quantum Classical Platforms harness the full capacity of Near-Term Quantum Computers. In this study, a focus is given on the variational circuits based approach accomplishing various machine learning tasks. We will survey a variety of experimental demonstrations conducted on actual quantum hardware and actively developing simulation software, anticipated to have a broad range of real-world applications in this increasingly growing field.**

***Index Terms*—Quantum Computing, Machine Learning, Hybrid quantum-classical platforms, Variational Circuits.**

I. Introduction

With the great advance in computer architecture in recent years, a new generation of computing systems is producing promising results in the domain of artificial intelligence. This will open doors to countless, creative applications in diverse fields such as health care, smart transport and e-commerce. In this study, we will review the recent research work trying to find the best possible combination of quantum-classical computing systems to achieve improved machine learning outcomes and speedup. Quantum computation has two main methods. The first generates the desired result by initializing the state of a quantum system and then using the Hamiltonian direct control to evolve the quantum state in a way that has a high probability of answering the question of interest. The Hamiltonian is often changed smoothly in these systems, so the quantum operations are truly analog in nature and can not be corrected completely by error, and will be referred to as "analogous quantum computing." This approach includes adiabatic quantum computing (AQC), quantum annealing (QA), and direct quantum simulation. The second approach, called "gate-based quantum computing," is similar to today's classical approaches, in that the problem is broken down into a sequence of a few very basic "primitive operations," or gates, which have well-defined "digital" measurement outcomes for certain input states. This digital property means that these type of designs can in principle use system-level error correction to achieve fault tolerance. In this study, we will focus on the second method, namely, the gate-based quantum computing. Regardless of the shortcomings of the Noisy Intermediate Scale Quantum (NISQ) hardware in terms of fault tolerance, qubit number, and circuit size, dependency on classical computing systems is unavoidable [2]. The main approach taken by recent researches consists in formalizing problems of interest as variational optimization problems and use hybrid systems of quantum and classical hardware to find approximate solutions. The intuition is that by implementing some subroutines on classical hardware, the requirement of quantum resources is significantly reduced, particularly the number of qubits, circuit depth, and coherence time. Therefore, in the hybrid algorithmic approach NISQ hardware focuses entirely on the classically intractable part of the problem. This article is structured as follows: section II provides an overview of Quantum Computing Systems, Section III provides an insight into how Quantum Computers can benefit different Machine Learning domains. Section IV compares some recent work in the Quantum Machine Learning domain, an overview of Hybrid Quantum-Classical Frameworks is given in Section V, and Section VI concludes the paper giving future work recommendations.

978-1-7281-7315-3/20/$31.00 ©2020 IEEE

II. Quantum Computing

Nearly four decades ago, Richard Feynman proposed the concept of performing computations using the Quantum Mechanics laws [3]. This concept was developed in 1985 by David Deutsch producing a Quantum Turing Machine [4]. Many Quantum Algorithms were then created to solve un-

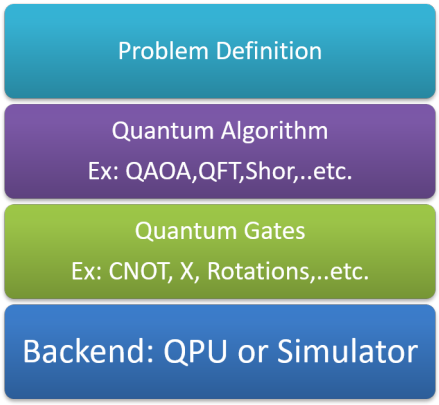

Fig. 1: Software stack

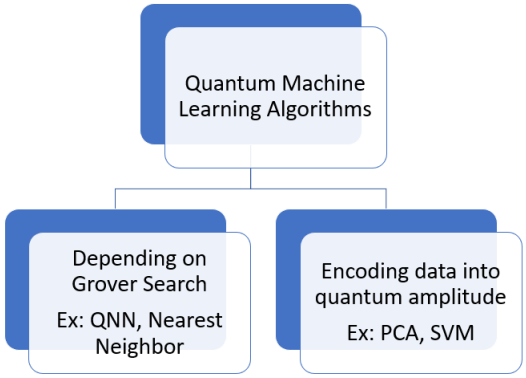

Fig. 2: Quantum Machine Learning Approaches

solvable problems on classical computing systems and offer unprecedented speed-up, the most popular of which is Peter Shor's factorization algorithm [5] that poses a threat to currently used RSA-based encryption methods, another famous algorithm is Grover's algorithm for database search [6]; in $\sqrt{n}$ steps, it can retrieve a predefined entry x in the unsorted list of n objects. A full list of Quantum Algorithms and attempts to implement them can be found in this listing [7]. Quantum Algorithms are implemented using reversible Quantum circuits, which are controlled by a set of unitary operations governed by the Quantum Mechanics laws [8] as seen in section II-A. The biggest challenge facing the Quantum Computing field today is the Algorithms-to-Machines gap, the full potential of Quantum Algorithms has not yet been achieved as the hardware scaling is much slower than the resource requirements of the algorithms [9].

A. *Gate-based Quantum Computing*

Unitary quantum gates are used to modify the qubits' states. The gate series implements a predefined quantum algorithm that tackles the problem. Figure 1 shows the software stack of a typical quantum solution.

The book [10] by Nielsen and Chuang showed different types of reversible Quantum Gates. Quantum Gate's simplest example is the X gate, it functions just like the traditional NOT gate except that the input is a quantum statevector. It's also simple, it's one of the family of 1 qubit gate. For the X gate the corresponding unitary matrix is:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{1}$$

One can deduce that $X|0\rangle = |1\rangle$ and $X|1\rangle = |0\rangle$ The second widely used gate is the CNOT gate with the following matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2}$$

The CNOT is a 2-qubit gate that only flips the target qubit state if the control qubit is $|1\rangle$ One can deduce that $CNOT|00\rangle = |00\rangle$ and $CNOT|01\rangle = |11\rangle$ knowing that the control bit is the rightmost qubit and the left qubit is the target qubit. For each quantum algorithm, there is an endless combinations of quantum gates implementing it. The main objective is to minimize the number of gates used, also known as the circuit depth, and to reduce the interconnections between the qubits. A quantum device with up to 20 qubit can be simulated with a regular laptop. With a supercomputer, one can simulate up to 50 qubit [11].

B. *How a Quantum Computer can learn?*

One may deduce that Quantum Computers will not be used to check e-mails, they are intended to conduct heavy computations with incredible performance, such as the exciting results obtained by the Google Sycamore 53-qubit chip [12] achieving quantum supremacy over the top supercomputer for last year, namely, the IBM Summit.

The idea of exploiting this computation power in machine learning is very promising, which led mathematicians to prove that a quantum computer can *learn* data like the early attempt to design Quantum Neural Networks by Kak in 1994 [13]. Dunjko et al. [14] give an overview of recent attempts to introduce quantum enhancements to classical machine learning models. Those attempts have two approaches for Quantum enhancement of classical machine learning as shown in Figure 2: 1) approaches depending on Grover's search and amplitude amplification to achieve up-to-quadratic speed-ups, and 2) approaches encoding relevant information into quantum amplitudes, with potential for even exponential speedups. Example of machine learning methods with Quantum enhancement are Quantum Support Vector Machine (QSVM), Quantum Principle Component Analysis (QPCA), Quantum-enhanced Kernel Methods, Quantum Phase estimation, and Quantum Matrix inversion [15].

One downside of pure quantum methods is that the requirements for implementation surpass the hardware specifications currently available during the NISQ era, this leads to unavoidable dependence on conventional computing systems and using the quantum device as accelerator. Moreover the use of data

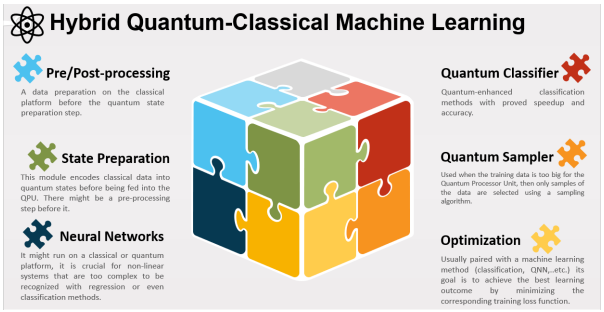

Fig. 3: Hybrid Quantum Classical System Modules

in superposition state requires the use of quantum Random Access Memory (qRAM) to access it efficiently [15], this access might be prohibitive for large scale data.

III. Hybrid Classical-Quantum Machine Learning

Considering a supervised learning task, e.g. classification or regression using classical data. Given a dataset $D = \{(x^{(i)}, y^{(i)})\}^{N}{}_{i=1}$ of N samples, the goal is to learn a model function f : $x \rightarrow y$ that maps each x_i to its corresponding target y_i. What determines a successful learning algorithm is its ability to recognize previously unseen data, or what is known as performance generalization. Success in generalization characterizes the predictive ability of a learner on independent test data. A loss function L is defined as the squared difference between the predicted value $f(x_i)$ and the actual value y_i. A standard approach is to minimize a suitable regularized loss function. Optimising for a classification problem with a 0-1 loss function is an NP-hard problem [1] even for a simple binary classification task. It is often approximated by a convex function that makes optimization easier. The hinge loss, notable for its use by support vector machines, is one such approximation [16]. As Preskill stated "We don't expect a quantum computer to solve worst case instances of NP-hard problems, but it might find better approximate solution or find it faster." [2]. The most recent advances in quantum computing show that machine learning might just be the right field of application. As machine learning usually turns down to a form of multivariate optimization [16]. This form of learning has already demonstrated results on actual quantum hardware or simulation as will be shown in Section IV. As Wittek predicted in his 2014 book [16], hybrid quantum-classical platforms became a promising trend in machine learning systems where the Quantum Processor acts as an accelerator. On the classical processor, arithmetic operations which require a large number of gates would be executed as implementing an algorithm on a near-term quantum computer is strictly tight with a small number of gates due to the coherence time limitation.

By surveying recent research work in the field, one can generalize a Hybrid quantum-classical machine learning models as shown in Figure 3. Each cube face represents a possible combination of Hybrid quantum-classical machine learning modules. For example, the face on the left consists of four jigsaw sub-modules, namely, a pre-processing module, a state

TABLE I: Example of Machine Learning Models

Machine Learning Model	Quantum Platform	HPC Platform	Module Function
Clustering	SVM PCA K-nearest neighbor	SVM PCA	Training Phase
Neural Network	QNN	CNN	Training Phase
Optimization	QAOA	Full Gradient Stochastic Gradient	Feedback Loop

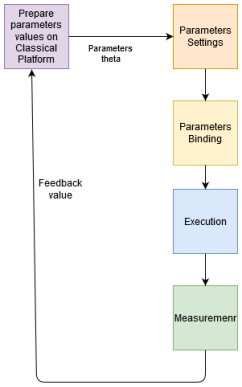

Fig. 4: Typical steps for variational circuit operation

preparation module, a Neural Network module, and an optimization module. This represents a Quantum Neural Network with classical data, that needed to be encoded into quantum states, and a classical optimizer that might be minimizing this Neural Network's loss function. The cube face on the right is another combination of modules that represents another Hybrid quantum-classical model, it consists of state preparation module, a Quantum classifier, a Quantum sampler, and an optimizer.

The Hybrid Quantum-Classical model consists of three phases: 1) State preparation phase: the quantum state is obtained by applying a series of parametrized gates on an initial quantum state, 2) measurement phase, where the Quantum Processor's output is registered and feed into the classical processor, and 3) optimization phase, the classical processor calculates the circuits parameters for the next iteration.

A. *Parametrized Gates*

Parametrization is recurrently used in many quantum algorithms, it became a standard building block for constructing libraries of standard gates and sub-circuits for many simulators [17]. Listing 1 shows a parametrized circuit implementation on IBM Qiskit [18].Figure 4 shows the typical steps of creating and executing a variational circuit. A "Parameter" class can be used to represent the numerical value to be used as the gate parameter. One famous example is the parametrized rotation gate, where the rotation angle can be changed as a parameter as shown below:

Listing 1: Python Code for Parameter Theta setting

```
from qiskit.circuit import Parameter

theta = Parameter('\theta' )

n = 5
qc = QuantumCircuit(5, 1)

qc.h(0)
for i in range(n-1):
   qc.cx(i, i+1)

qc.barrier()
qc.rz(theta, range(5))
qc.barrier()

for i in reversed(range(n-1)):
   qc.cx(i, i+1)
qc.h(0)
qc.measure(0, 0)

qc.draw()
```

Before executing a parametrized circuit, the parameters_method must be called to bind parameters to its corresponding values as follows:

Listing 2: Parameters Binding

```
import numpy as np

theta_range = np.linspace(0, 2 * np.\pi, 128)

circuits = [qc.bind_parameters({theta:
    theta_val})
         for theta_val in theta_range]

print(circuits[-1].draw(fold=120))
print(circuits[-1].parameters)
```

Then the execute method accepts a parameter_binds keyword as an argument as follows:

Listing 3: Binding and Execution

```
from qiskit import BasicAer, execute

job = execute(qc,
     backend=BasicAer.get_backend(
             'qasm_simulator'),
     parameter_binds=[{theta: theta_val}
       for theta_val in theta_range])

# Note: Bind labels are not preserved in
    executed experiments.
counts = [job.result().get_counts(i) for i in
    range(len(job.result().results))]
```

Variational circuits are a subset of parametrized quantum circuits, they implement Quantum Variational Algorithms. They are trending to be used in Machine Learning tasks with a great success as shown in Section IV.

B. *Variational Quantum Algorithms*

Based on the variational Principle [19], variational algorithms use approximations to use solvable problems to solve unsolvable ones. The variational method was originally used to minimize a quantum system's energy states.The basic idea of this approach is that one defines a test wave function (sometimes called an ansatz) as a function of certain parameters, and then one seeks the values of those parameters that minimize the energy expectation value in reference to those parameters. This minimized response is then an approximation to the lowest energy specific state, and the expectation value serves as the upper limit on the ground state's energy.Recent research work [20], [21] proved the quantum advantage over classical variational methods. In particular, if the classical variation method is applied to a n qubits system, an exponential number (in n) of complex numbers is required to generally represent the system's wave function. For a quantum computer, however, one can generate this state directly using a parameterized quantum circuit and then estimate the expected value of the energy by repeated measurements. As examples of Variational Quantum Algorithm, a quick summary and citations are given for three famous algorithms, namely, VQE, VQG, and QAOA. Varational Quantum Eigensolver (VQE) is an application of the variational principle where a quantum computer is used to estimate the expectation value of its electronic Hamiltonian while a classical optimizer is used to adjust the quantum circuit parameters to find the molecule's ground state energy. Variational Quantum Generator (VQG) [22] generates representative data such as text, images, audio, or videos. The system consists of two parts: one variational circuit for encoding classical data into quantum states, the other is the training phase that could be a classical neural network. Quantum Approximate Optimization Algorithm (QAOA) [23] can be used to minimize the training loss function more efficiently than classical optimization techniques thanks to superposition leading to Quantum Parallelism.

IV. Related Work

Date et al. [24] combined the HPC platform with an Adiabetic Quantum Processor to perform a classification task of the MNIST dataset. The implementation depended on both Quantum and Classical platforms. Their experimental work investigated two classification models, namely, the Deep Belief Network (DBN) and the Restricted Boltzmann Machines (RBM). The process performance bottlenecks were: 1) Matrix Computation, 2) Sampling Task. The experimental results showed that heavy matrix computations are more efficient on classical platform while sampling task is more convenient to the quantum one, as quantum mechanical processes are used to generate samples, making them truly random. Barkoutsos et al. [25] developed an improved platform to perform combinatorial optimization problems on hybrid quantum-classical variational circuits. The Conditional Value-at-Risk was used as an aggregation function. It was empirically shown – using classical simulation as well as quantum hardware – that this approach

TABLE II: Experimental environment settings in related work.

Reference	Task	Platform	Model	Simulation	Quantum Device
Date *et al.* [24]	Classification	TF-D-Wave Sampling lib	RBM,DBN	✓	✓
Barkoutsos *et al.* [25]	Optimization	Qiskit-COBYLA	QAOA,VQE	✓	✓
Romero *et al.* [22]	Generative	Forest-PyTorch	VQG	✓	✓
Liu *et al.* [26]	Classification	VQC-PyTorch	QCCNN	✓	-

leads to faster convergence to better solutions for all combinatorial optimization problems tested. Romero et al. [22] offer an approach to perform generative modeling of continuous probability distributions under a Hybrid Quantum-Classical platform. The same architecture employed in VQG can be used to build models for classification, where the training of the proposed generator can be performed with both classical and quantum discriminators, taking advantage of the integration of gradient estimation of variational circuits and automatic differentiation strategies. Liu et al. [26] proposed a hybrid quantum-classical convolutional neural network (QCCNN), inspired by convolutional neural networks (CNNs) but using the quantum advantage to enhance the feature mapping process which is the most computational intensive part of CNN. The generalized feature map with a parametrized quantum circuit is able to explore the correlations of neighbouring data points in a exponentially large linear space,. Table II shows the experimental environment for each related work.

V. Underlying Hybrid Frameworks

To guarantee a smooth integration of the different hybrid system platforms, many open source tools developed easy and ready to use tools to develop and integrate such heterogeneous systems. Xanadu PennyLane [27] is used as a hub between different frameworks constructing the hybrid system as shown in Figure 5

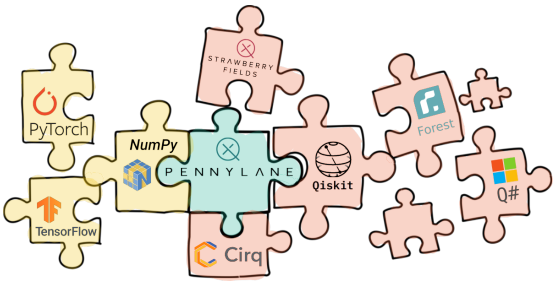

Fig. 5: PennyLane Plugins

Quantum TensorFlow [28] is the integration between Google Cirq Quantum Simulator and the famous Classical Deep Learning platform Tensorflow. By surveying recent research work, one can conclude that both IBM Qiskit and Rigetti Forest integrate smoothly with PyTorch as they are all implemented in Python.

VI. Conclusion and Future Work

During the current Age of the Noisy Intermediate Scale Quantum (NISQ), Quantum Algorithms can be run with strict limitations on Near-Term Quantum Computers. It influences how we approach computational challenges in many areas and one important example is machine learning. Performance improvement could be demonstrated theoretically and experimentally, however, scaling up the quantum hardware is crucial to deploy real-life sized applications. This article shows some recent research work to accomplish different Machine Learning Tasks based on hybrid Quantum-Classical Platforms. This research area is still active for unsupervised learning tasks that could be applied in different domains like finance, logistics, material science, and astronomy. More future research work is needed to investigate what the variational circuits approach can offer compared to purely classical models for unsupervised learning models. For example, some theoretical works [29], [30] indicate that variational circuits might bring an advantage for discrete generative tasks, however the extent to which this impacts applications such as image, sound and language generation will require extensive computational studies on real datasets [22].

Acknowledgment

We thank Dr. Sherif Saif and Dr. Amany Abdelsamea for their guidance and helpful feedback throughout this research work.

References

[1] V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu, "Agnostic learning of monomials by halfspaces is hard," *SIAM Journal on Computing*, vol. 41, no. 6, pp. 1558–1590, 2012.

[2] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[3] R. P. Feynman, "Simulating physics with computers," *Int. J. Theor. Phys*, vol. 21, no. 6/7, 1999.

[4] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[5] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[7] S. Jordan, "Quantum algorithm zoo," *Retrieved June*, vol. 27, p. 2013, 2011.

[8] M. M. Wilde, *Quantum information theory*. Cambridge University Press, 2013.

[9] M. Martonosi and M. Roetteler, "Next steps in quantum computing: Computer science's role," *arXiv preprint arXiv:1903.10541*, 2019.

[10] M. A. Nielsen and I. Chuang, "Quantum computation and quantum information," 2002.
[11] R. Babbush, D. W. Berry, J. R. McClean, and H. Neven, "Quantum simulation of chemistry with sublinear scaling in basis size," *npj Quantum Information*, vol. 5, no. 1, pp. 1–7, 2019.
[12] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
[13] S. C. Kak, "Quantum neural computing," in *Advances in imaging and electron physics*. Elsevier, 1995, vol. 94, pp. 259–313.
[14] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," *Reports on Progress in Physics*, vol. 81, no. 7, p. 074001, 2018.
[15] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
[16] P. Wittek, *Quantum machine learning: what quantum computing means to data mining*. Academic Press, 2014.
[17] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, "Quantum chemistry in the age of quantum computing," *arXiv preprint arXiv:1812.09976*, 2018.
[18] H. Abraham and A. et al., "Qiskit: An open-source framework for quantum computing," 2019.
[19] I. Ekeland, "On the variational principle," *Journal of Mathematical Analysis and Applications*, vol. 47, no. 2, pp. 324–353, 1974.
[20] A. Peruzzo *et al.*, "A variational eigenvalue solver on a quantum processor. eprint," *arXiv preprint arXiv:1304.3061*, 2013.
[21] D. Wecker, M. B. Hastings, and M. Troyer, "Progress towards practical quantum variational algorithms," *Physical Review A*, vol. 92, no. 4, p. 042303, 2015.
[22] J. Romero and A. Aspuru-Guzik, "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions," *arXiv preprint arXiv:1901.00848*, 2019.
[23] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.
[24] P. Date, C. Schuman, R. Patton, and T. Potok, "A classical-quantum hybrid approach for unsupervised probabilistic machine learning," 01 2020, pp. 98–117.
[25] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving variational quantum optimization using cvar," 2019.
[26] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, "Hybrid quantum-classical convolutional neural networks," *arXiv preprint arXiv:1911.02998*, 2019.
[27] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, C. Blank, K. McKiernan, and N. Killoran, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.
[28] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters *et al.*, "Tensorflow quantum: A software framework for quantum machine learning," *arXiv preprint arXiv:2003.02989*, 2020.
[29] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, no. 6, pp. 595–600, 2018.
[30] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, "The expressive power of parameterized quantum circuits," *arXiv preprint arXiv:1810.11922*, 2018.