Accelerating Precision Medicine with Quantum Computing and AI

Exploring the future of Personalized Health Care

Shaurya Singh, Abrahim Mahmud

2022
Jan 23

Contents

The convergence of quantum computing, artificial intelligence, and precision medicine promises to revolutionize health care. Precision medicine methods identify phenotypes of patients with less-common responses to treatment or unique healthcare needs. AI leverages sophisticated computation and inference to generate insights, enables the system to reason and learn, and empowers clinician decision-making through augmented intelligence. Quantum-accelerated machine learning could support further breakthroughs in this area and ultimately enable causal inference models for drugs, identifying and explaining relationships among interventions and treatments on the one hand, and outcomes on the other in real-time, to provide the next-best medical action at the individual level.

- **Keywords:** Quantum Computing, Artificial intelligence, Precision Medicine
- **Environment:** Emacs 29.0, Org mode 9.6 & Python 3.9.10
- **License:** MIT

"By augmenting human performance, AI has the potential to markedly improve productivity, efficiency, workflow, accuracy, and speed, both for [physicians] and for patients ... What I'm most excited about is using the future to bring back the past: to restore the care in healthcare." — Eric Topol

1 Introduction

Healthcare data—such as information from clinical trials, disease registries, electronic health records (EHRs), and medical devices—is growing at a compound annual growth rate of 36 percent (NCHIT, 2021). Increasingly, this data helps address 4 challenges associated with healthcare:

- Better health
- Lower cost
- Enhanced patient experiences
- Improved healthcare practitioner work lives.

At the same time, healthcare consumers are making more decisions and have to navigate an increasingly complex system. Traditionally, diagnosing a patient's condition has been based heavily on the umbrella approach, which is based solely on the patient-reported symptoms, resulting in an umbrella diagnosis and treatment that frequently fails to achieve their intended effects due to individual variability. As a

Figure 1: IBM System One

result, many existing therapies fail to achieve their intended effects due to individual variability. Precision medicine aims to allow tailor prevention and treatment with an individual approach. (Ravizza et al., 2019). Due to the complexity of human biology, individualized medicine requires taking into account aspects that go well beyond standard medical care. Medical care only has a relative contribution of 10 to 20 percent of outcomes; health-related behaviors, socioeconomic factors, and environmental aspects account for the other 80 to 90 percent (Magnan & , 2017). Computationally, the interdependencies and correlations among these diverse contributors create formidable challenges concerning optimizing treatment effectiveness. (McDermott et al., 2012)

Significant investments are being made to deliver the right data and powerful insights at the point of care. Industry incumbents and new entrants alike are trying to create digital experiences that reinforce healthy, preventive behaviors. Despite that, accounting for the exponential possibilities from this diversity of new data is stretching the capabilities of classical computing systems. (Grumbling & Horowitz, 2020).

2 AI in Precision Medicine

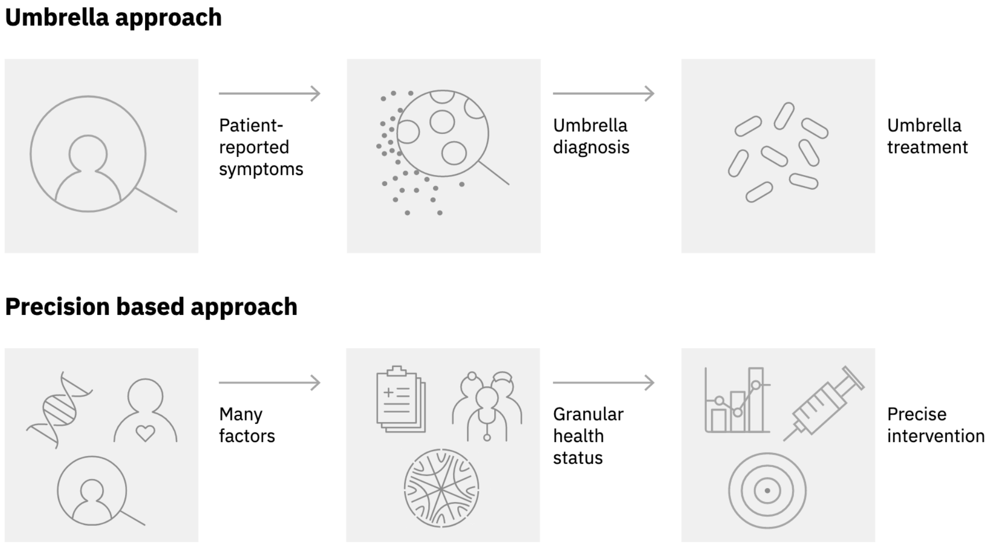


Figure 2: Precision Approach vs. Umbrella Approach

A key aspect of tailoring medical approaches is proactivity. Early treatments and preventive interventions tend to drastically improve outcomes and optimize costs. Classical machine learning has already shown some promise in predicting the risk of future diseases for a range of patient groups based on EHRs (Jauk et al., 2020). Supervised and unsupervised quantum-enhanced machine learning techniques could allow earlier more accurate, and more granular risk predictions. Eventually, medical practitioners might even have the tools to understand how an individual's risk for any given condition changes over time, enabled by continual virtual diagnostics based on ongoing data streams from individuals. (Kumar et al., 2021)

2.1 Accelerating AI with Quantum Computing

Quantum computers process information in a fundamentally different way from traditional computers. Previous computer technology advancements—such as integrated circuits—enabled faster computing, but were still based on classical information processing. These computers manipulate quantum bits (qubits). These are unlike classical bits, which store information as either a 0 or 1, and they can display uniquely quantum properties, such as entanglement. As a result, it becomes possible to construct quantum algorithms that can outperform their classical counterparts which are not able to leverage quantum phenomena. (Montanaro, 2016)

- Chemistry, machine learning/artificial intelligence (AI), optimization, or simulation tasks. Machine learning has shown potential to be enhanced by quantum computing and is symbiotically helping drive quantum advances (Carleo & Troyer, 2017)
- Complex correlations and interdependencies among many highly interconnected elements, such as molecular structures in which many electrons interact
- Inherent scaling limits of relevant classical algorithms. For instance, the resource requirements of classical algorithms may increase exponentially with problem size, as is the case when simulating the time evolution of quantum systems (Whitfield et al., 2011).

2.2 Hybrid Quantum-Classical Neural Networks

The most computationally demanding part of gradient-based algorithms, namely, computing the fitness function and its gradient for control input, can be accomplished by the process of evolution and measurement on quantum hardware. By posing queries to and receiving answers from these devices, classical computing devices update the control parameters until an optimal control solution is found (Li et al., 2017). Using this hybrid approach gives rise to interesting areas of research that seek to leverage the principles of quantum mechanics to augment machine learning or vice-versa. Enabling us to enhance classical ML algorithms by outsourcing difficult calculations to a quantum computer.

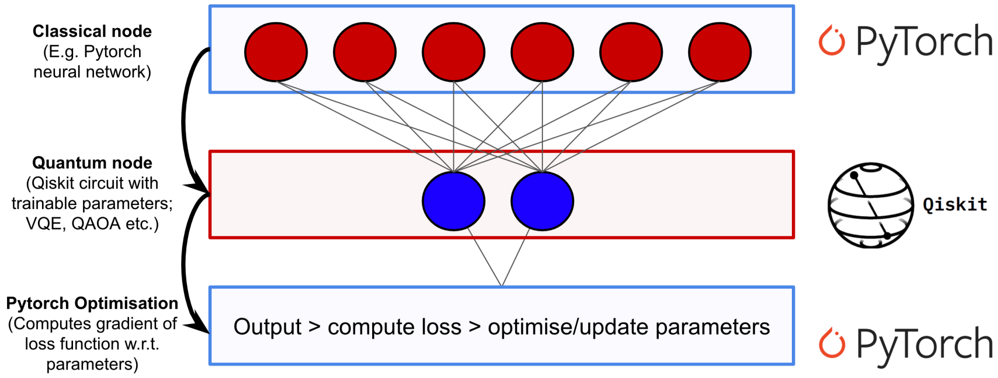

Classical node
(E.g. Pytorch neural network)
Quantum node
(Qiskit circuit with trainable parameters; VQE, QAOA etc.)
Pytorch Optimisation
(Computes gradient of loss function w.r.t. parameters)
Output > compute loss > optimise/update parameters
PyTorch
Qiskit
PyTorch

Figure 3: Example of a Hybrid Quantum-Classical Neural Network

To create a quantum-classical neural network, one can implement a hidden layer for a neural network using a parameterized quantum circuit, a quantum circuit where the rotation angles for each gate are specified by the components of a classical input vector. The outputs from the neural network's previous layer will be collected and used as the inputs for a parameterized circuit. The measurement statistics of the circuit can then be collected and used as inputs for the following layer.

Each classical equation is modeled through the nonlinear function

$$h_x = \sigma(w_a x_a + w_b x_b + w_c x_c) \tag{1}$$

where h_i is the value of neuron i at each hidden layer. $R(h_i)$ represents any rotation gate about an angle equal to h_i and y is the final prediction value generated from the hybrid network.

2.3 Generative Adversarial Networks

Generative Adversarial Networks, or GANs for short, are an approach to generative modeling using deep learning methods, such as convolutional neural networks. Generative modeling is an unsupervised learning task in machine learning that involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset (Choi et al., 2020)

GANs are a clever way of training a generative model by framing the problem as a supervised learning problem with two sub-models: the generator model that we train

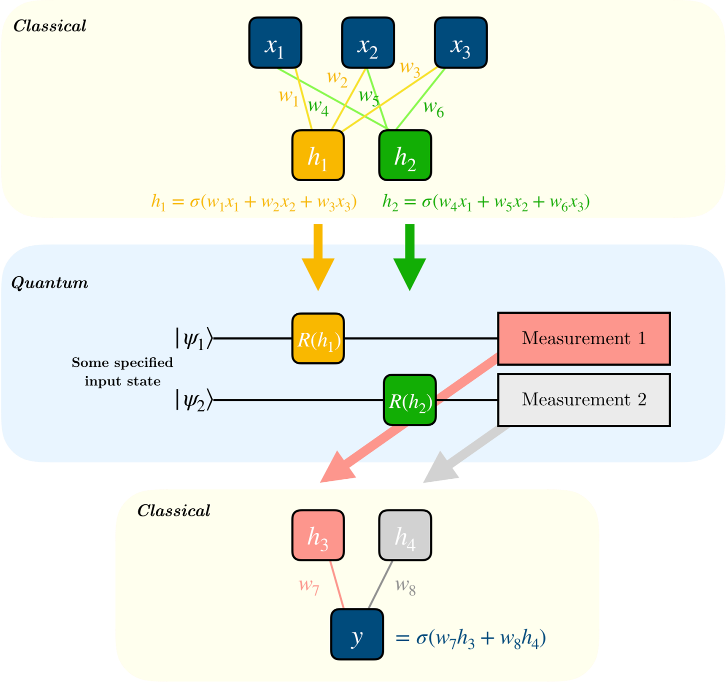


Figure 4: Quantum-classical Neural Network using a Parameterized Quantum Circuit

to generate new examples, and the discriminator model that tries to classify examples as either real (from the domain) or fake (generated). The two models are trained together in a zero-sum game, adversarial, until the discriminator model is fooled about half the time, meaning the generator model is generating plausible examples. This can be modeled through the equation (Wang et al., 2020)

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_{\text{generated}}(z)}[1 - \log D(G(z))] \tag{2}$$

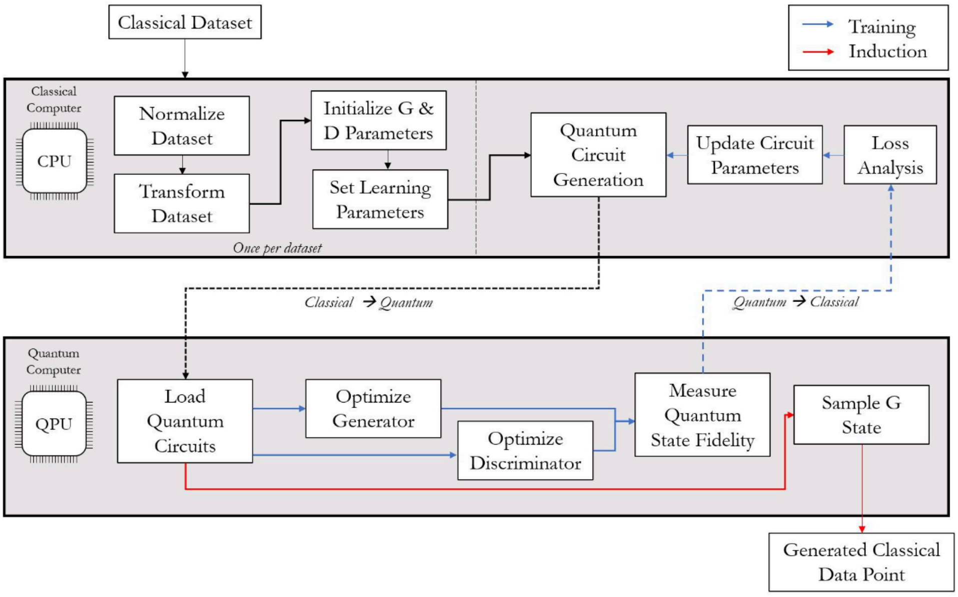
2.3.1 MolGAN

Existing drug discovery pipelines take 5-10 years and cost billions of dollars. Computational approaches aim to sample from regions of the whole molecular and solid-state compounds called chemical space which could be on the order of 1060. Deep generative models can model the underlying probability distribution of both the physical structures and property of drugs and relate them nonlinearly. By exploiting patterns in massive datasets, these models can distill salient features that characterize the molecules. We can utilize Generative Adversarial Networks (GANs) to discover drug candidates by generating molecular structures that obey chemical and physical properties and show affinity towards binding with the receptor for a target disease. (Li et al., 2021)

Currently, this is accomplished through the Tensorflow library MolGAN. However, However, classical GANs cannot explore certain regions of the chemical space and suffer from curse-of-dimensionality. Computing these drug candidates can be computationally expensive, and the resource requirements of these classical algorithms increase exponentially with problem size. On the other hand. A full quantum GAN may require more than 90 qubits even to generate QM9-like small molecules and is impractical in the current day and age

2.3.2 Qubit-efficient Quantum Molecule Generation

Once again, we can apply our hybrid approach. A qubit-efficient quantum GAN with a hybrid generator (QGAN-HG) can be used to learn a richer representation of molecules via searching exponentially large chemical space with fewer qubits and more efficiently than a classical GAN. The QGAN-HG model is composed of a hybrid quantum generator that supports a various number of qubits and quantum circuit layers, and, a classical discriminator. The approach is significantly quicker than our classical GAN model.



3 Prototype

To test the speed, practicality, efficiency, and cost of quantum-accelerated ML as well as its usefulness in precision medicine, we have devised two prototypes written in Python. The concise, expressive, and dynamic nature of the Python language makes it well suited for prototyping tasks. Notebook one will test how viable our QML approach is at accelerating image and text sorting. This script can be adapted to identify mutations, distinguish genomic variants, as well as identify an individual's susceptibility to rare diseases through an analysis of their previous health and history. Notebook 2 will generate viable drugs based on ones that currently exist, and will test how viable our QML approach is to accelerating current conventional drug discovery pipelines. Both models utilize PyTrorch and IBM's Quantum Services for training and testing

The full code for both notebooks are available under the prototype folder.

3.1 Image recognition (QuTorch-HG)

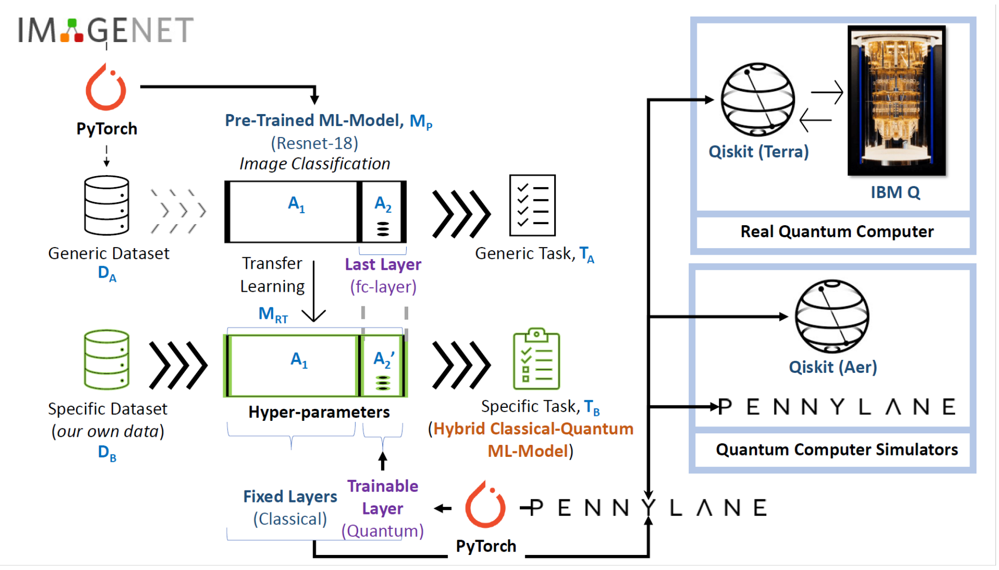


Figure 5: Hybrid Base Nueral Network

We have created a base neural network model, which utilizes hybrid machine learning to create a model trained from any dataset in ImageNet format. The base model is used as the base for Transfer Learning, on an Image Classification task (based on resnet18). The last layer of this pre-trained model (fully connected/FC layer) is then modified through a quantum machine learning framework, generating a new model.

We will be testing its efficiency, practicality, and accuracy. We are training the model with 4 Qubits at 8 epochs. Loss is calculated using the Huber loss method, with the following equation

$$L_\delta = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & if\ |(y - \hat{y})| < \delta \\ \delta((y - \hat{y}) - \frac{1}{2}\delta) & otherwise \end{cases} \tag{3}$$

Accuracy is calculated via a confusion matrix, with the following metrics

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{7}$$

3.2 Quantum Accelerated Drug & Molecule Generation (QGAN-HG)

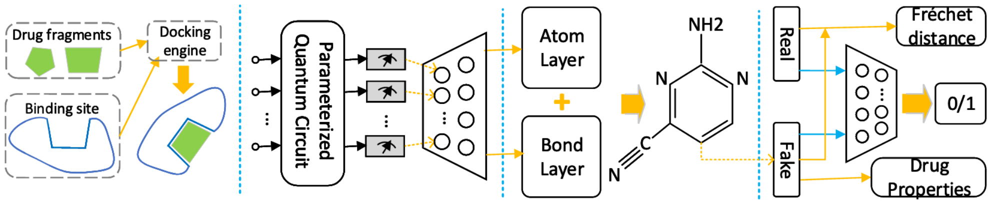


Figure 6: Quantum Accelerated Molecule Generation

Firstly, only generated molecules that have a high affinity towards the receptor binding sites are considered valid. (Andreyev & Zybailov, 2020) Next, a parameterized quantum circuit with last-layer N measures the expectation values and processes them through a classical stage. Then, we apply the atom layer and bond layer to generate synthetic molecular graphs. Afterward, a batch of real molecules from a training dataset (in this case we are using QM9) and a batch of generated synthetic molecules are fed into a classical discriminator for real/synthetic prediction and Frechet distance score calculation.

3.3 Data

Our Image Data is obtained from Standford's ImageNet collection, a large-scale ontology of images built upon the WordNet structure. ImageNet aims to populate the majority of the 80,000 synsets of WordNet with an average of 500–1000 clean and full-resolution images, with currently over 14,197,122 images and 21841 synsets indexed. The specific dataset used in this example can be found at Paul Timothy: Chest X-RAY Pneumonia Dataset and is licensed under CC0 1.0: Public Domain.

The Molecular data used to train our MolGAN and QuGAN models is the QM9 Dataset obtained from Anatole von Lilienfeld. The dataset contains the computed geometric, energetic, electronic, and thermodynamic properties for 134k stable small organic molecules made up of CHONF. These molecules correspond to the subset of all 133,885 species with up to nine heavy atoms (CONF) out of the GDB-17 chemical universe of 166 billion organic molecules. The model is trained on geometries, corresponding harmonic frequencies, dipole moments, polarizabilities, energies, and enthalpies. (Raghunathan Ramakrishnan et al., 2019)

3.4 Tools and Hardware

The open-source Qiskit framework provides convenient access to multiple quantum simulators as well as a real quantum computer backend. The user can choose to utilize either IBM's cloud-based QASM simulator technology, Google's local equivalent Cirq, or Pennylane's quicker but less accurate lightning simulator. All three backends allow for quick training and testing via quantum simulators and real quantum hardware (Cross, 2018).

While simulated hardware produces *slightly* different results to actual hardware, the margin is within 1%. The tests below were conducted using actual IBMQ hardware, on the `ibmq_lima` quantum computer (Quantum, 2021)

Code is developed in emacs' org-mode, combined with Jupyter kernels allowing for quick prototyping, and utilize PyTorch for pre-processing and post-processing of our neural network, taking advantage of GPU Acceleration via Nvidia CUDA if available. This allows us to process images in real time on Google's Compute Engine VM's via Google Colab, allowing for low operating costs, high performance, and good portability. Org-Mode allows us to document our code and easily integrate with other languages, while staying in a simple plain text format.

Nix and NixOS are used for dependency management, allowing us to create reproducible, binary cached, distributed, and non-privileged builds for our software. This also

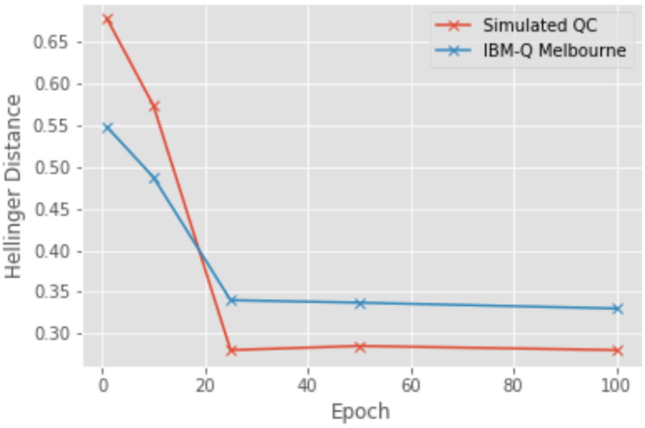
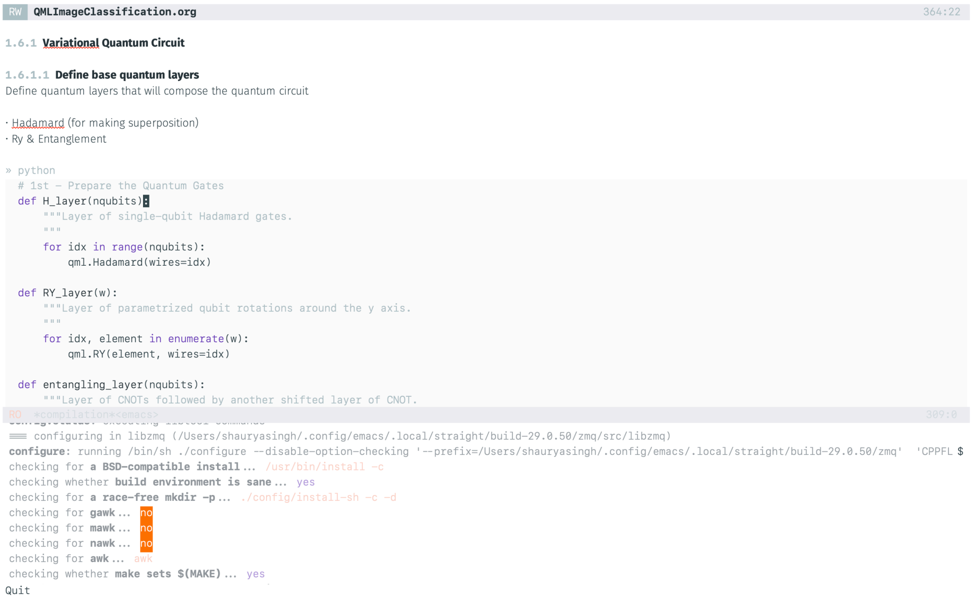


Figure 7: Actual vs. Simulated Hardware



1.6.1 Variational Quantum Circuit

1.6.1.1 Define base quantum layers

Define quantum layers that will compose the quantum circuit

- Hadamard (for making superposition)
- Ry & Entanglement

» python

```
# 1st – Prepare the Quantum Gates
def H_layer(nqubits):
    """Layer of single-qubit Hadamard gates.
    """
    for idx in range(nqubits):
        qml.Hadamard(wires=idx)

def RY_layer(w):
    """Layer of parametrized qubit rotations around the y axis.
    """
    for idx, element in enumerate(w):
        qml.RY(element, wires=idx)

def entangling_layer(nqubits):
    """Layer of CNOTs followed by another shifted layer of CNOT.
```

```
═══ configuring in libzmq (/Users/shauryasingh/.config/emacs/.local/straight/build-29.0.50/zmq/src/libzmq)
configure: running /bin/sh ./configure --disable-option-checking '--prefix=/Users/shauryasingh/.config/emacs/.local/straight/build-29.0.50/zmq'  'CPPFL $
checking for a BSD-compatible install ...  /usr/bin/install -c
checking whether build environment is sane ...  yes
checking for a race-free mkdir -p ...  ./config/install-sh -c -d
checking for gawk ...  no
checking for mawk ...  no
checking for nawk ...  no
checking for awk ...  awk
checking whether make sets $(MAKE) ...  yes
Quit
```

Figure 8: Jupyter in Org-Mode

allows for quicker deployments on hardware and provides stability for the end-user, as well as allows us to quickly spin up development environments on UNIX-based hardware.

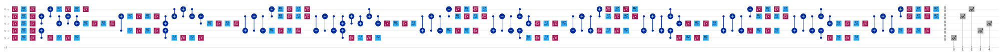


Figure 9: Sample Generated Quantum Circuit on IBMQ

4 Results

4.1 Speed

This is the largest benefit of quantum-accelerated machine learning. We can see that in both algorithms, quantum computing provided an exponential increase in speed over the non-accelerated counterpart. In the case of QGAN, We can see anywhere from an 8-32% decrease in the time needed to generate molecules, with the same input parameters. In the case of our QuTorch-HG algorithm, it can process a batch of images within 1/10th of a second, allowing for models to be trained at 95% accuracy in under 5 minutes. A similar model, written with TensorFlow and trained on the same CPU, took 32 minutes to achieve 94.3% accuracy.

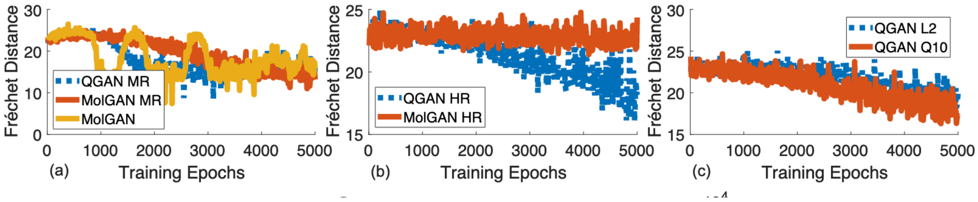


Figure 10: Training comparison among GAN flavors

4.2 Accuracy

In both prototypes, accuracy was as expected. The QuTorch-HG algorithm tested at around 96.23% accuracy on average after 8 epochs. The QGAN prototype created valid molecules 100% of the time during our testing.

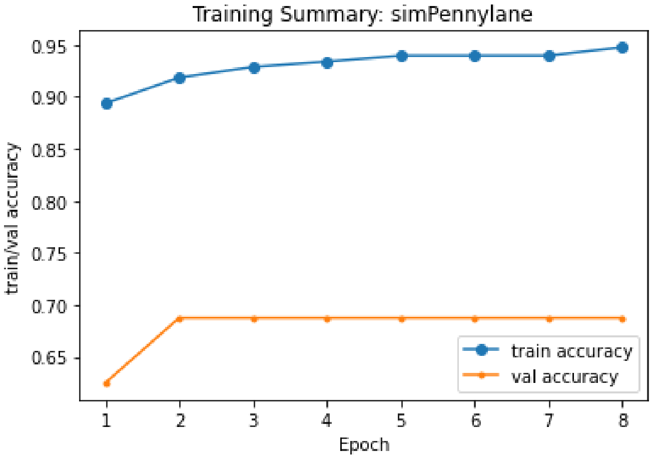


Figure 11: Training Accuracy vs. Epochs

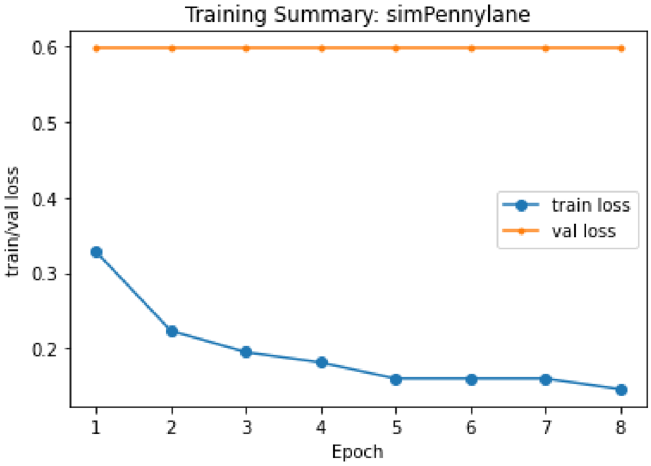


Figure 12: Training Accuracy vs. Epochs

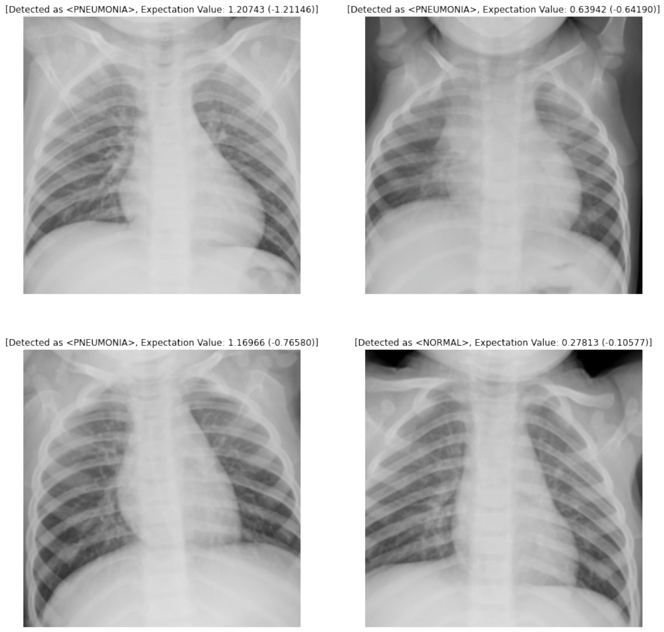

[Detected as <PNEUMONIA>, Expectation Value: 1.20743 (-1.21146)]
[Detected as <PNEUMONIA>, Expectation Value: 0.63942 (-0.64190)]
[Detected as <PNEUMONIA>, Expectation Value: 1.16966 (-0.76580)]
[Detected as <NORMAL>, Expectation Value: 0.27813 (-0.10577)]

Figure 13: Analysis of Pneumonia through our QuTorch-HG algorithm

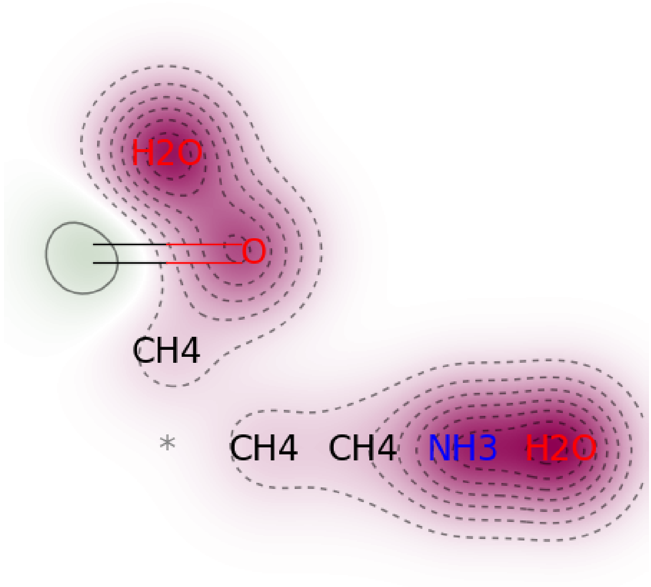


Figure 14: Sample Generated Molecule through our QGAN-HG algorithm

4.3 Pricing

As of early 2022, IBM Quantum Services allows researchers and students to use their 5 qubit quantum computers for development free of charge. The GPU accelerator was provided by Google's Colab program, free of charge as well. Those looking for real-time analysis can utilize Google Compute Engine VM's, such as the A2 Accelerator for just $0.009 an hour. Our hybrid model is efficient and as all computation is handled through the cloud, energy costs are nominal.

On the other hand, our algorithms can also be applied to healthcare cost analysis, such as improving insurance pricing computations, allowing for lower average premiums, as well as better-tailored premium options. We strongly believe investing in quantum computing now will result in increased profits in the future.

4.4 Frontend

In order to provide users an easy interface to collect and store their data, we created a Frontend written in React. Through React, we created a website that can collect crucial information, display user data, as well as ensure each user has their data easily accessible and protected through a database managed on firestore, backed up Google's OAuth authentication. Each user can view their likely hood for rare diseases, can document any symptoms or issues they're experiencing, talk with a live doctor, and analyze XRAY-scans.

The full code for both notebooks are available under the frontend folder.

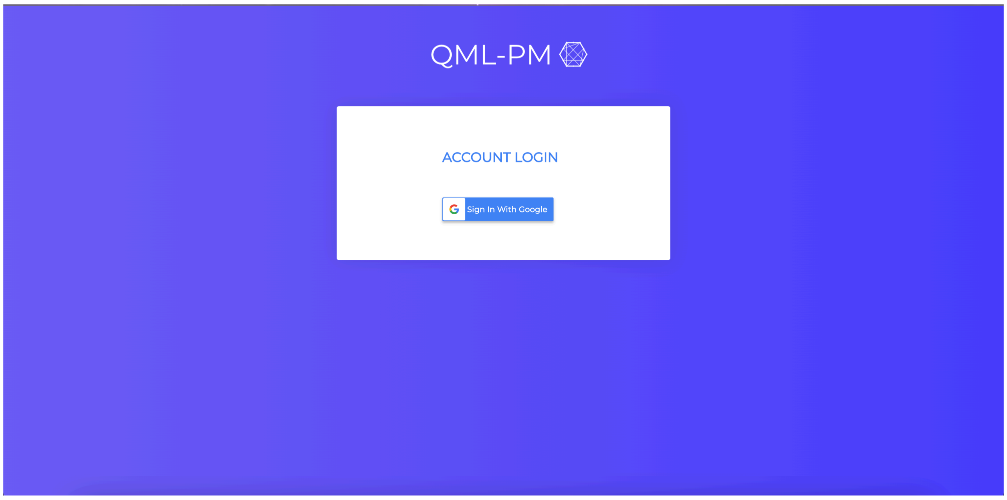
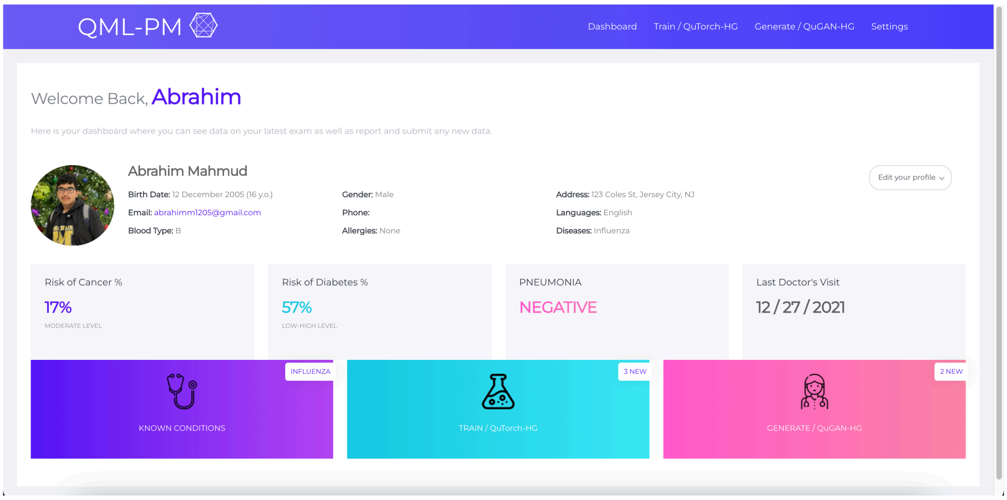


Figure 15: Google Login with Firebase



Welcome Back, Abrahim

Here is your dashboard where you can see data on your latest exam as well as report and submit any new data.

Abrahim Mahmud

Birth Date: 12 December 2005 (16 y.o.)	**Gender:** Male	**Address:** 123 Coles St, Jersey City, NJ
Email: abrahimm1205@gmail.com	**Phone:**	**Languages:** English
Blood Type: B	**Allergies:** None	**Diseases:** Influenza

Risk of Cancer %	Risk of Diabetes %	PNEUMONIA	Last Doctor's Visit
17%	57%	NEGATIVE	12 / 27 / 2021
MODERATE LEVEL	LOW-HIGH LEVEL		

Figure 16: React-based User Dashboard and Frontend

5 Conclusion and Further Research

Healthcare data, collected from the many different modalities and locations that surround patients—is underused today. Clinical factors only account for a small percentage of health-relevant data. Therefore, obtaining actionable insights from other areas, including genomics, behavior, and social and environmental influences, is critical.

Quantum-enhanced machine learning algorithms stand out for their breadth of application, their efficiency, and relatively low running costs. We are entering an era where the characteristics of health datasets, such as being frequently heterogeneous and unevenly distributed, are producing complex computational challenges for modern AI. For instance, researchers have been exploring how to speed up the computationally expensive algorithms that are at the core of machine learning and AI modeling with quantum approaches, including operations with large matrices.

The maturity horizons of quantum computing use cases in healthcare depend not only on cross-industry quantum algorithm and hardware developments but also on the increasing availability of relevant data. While certain modalities are already being leveraged for advanced computational modeling, health-relevant datasets are rapidly growing from increases in both the number of people being "instrumented," as well as the data types being collected. As the number of accessible health-relevant data sources continues to grow, the potential for quantum computing to add value likewise increases. AI in medicine will continue to proactively predict, analyze, and prevent disease, and quantum acceleration will play a key role in creating target treatments and identifying complex mutations.

6 References

Andreyev, D. S., & Zybailov, B. L. (2020). Integration of flow cytometry and single cell sequencing. *Trends in Biotechnology*, *38*(2), 133–136. https://doi.org/10.1016/j.tibtech.2019.09.002

Carleo, G., & Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, *355*(6325), 602–606. https://doi.org/10.1126/science.aag2302

Choi, K., Grover, A., Singh, T., Shu, R., & Ermon, S. (2020). Fair generative modeling via weak supervision. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 1887–1898). PMLR. https://proceedings.mlr.press/v119/choi20a.html

Cross, A. (2018). The ibm q experience and qiskit open-source quantum computing software. *Aps March Meeting Abstracts*, *2018*, L58–003.

Grumbling, E., & Horowitz, M. (2020). *Quantum computing*. https://doi.org/10.17226/25196

Jauk, S., Kramer, D., Großauer, B., Rienmüller, S., Avian, A., Berghold, A., Leodolter, W., & Schulz, S. (2020). Risk prediction of delirium in hospitalized patients using machine learning: An implementation and prospective evaluation study. *Journal of the American Medical Informatics Association*, *27*(9), 1383–1392. https://doi.org/10.1093/jamia/ocaa113

Kumar, Y., Koul, A., Sisodia, P. S., Shafi, J., Verma, K., Gheisari, M., & Davoodi, M. B. (2021). Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. *Wireless Communications and Mobile Computing*, *2021*, 1–16. https://doi.org/10.1155/2021/1616725

Li, J., Topaloglu, R., & Ghosh, S. (2021). Quantum generative models for small molecule drug discovery. *Arxiv:2101.03438 [Quant-Ph]*. http://arxiv.org/abs/2101.03438

Li, J., Yang, X., Peng, X., & Sun, C.-P. (2017). Hybrid quantum-classical approach to quantum optimal control. *Physical Review Letters*, *118*(15). https://doi.org/10.1103/physrevlett.118.150503

Magnan, S., & . (2017). Social determinants of health 101 for health care: Five plus five. *NAM Perspectives*, *7*(10). https://doi.org/10.31478/201710c

McDermott, J. E., Wang, J., Mitchell, H., Webb-Robertson, B.-J., Hafen, R., Ramey, J., & Rodland, K. D. (2012). Challenges in biomarker discovery: combining expert insights with statistical analysis of complex omics data. *Expert Opinion on Medical Diagnostics*, *7*(1), 37–51. https://doi.org/10.1517/17530059.2012.718329

Montanaro, A. (2016). Quantum algorithms: an overview. *Npj Quantum Information*, *2*(1). https://doi.org/10.1038/npjqi.2015.23

NCHIT. (2021). *Non-federal acute care hospital electronic health record adoption*. https://www.healthit.gov/data/

Quantum, I. (2021). *Ibm quantum services*. https://quantum-computing.ibm.com/

Raghunathan Ramakrishnan, Dral, P., Rupp, M., & O. Anatole Von Lilienfeld. (2019). *Quantum chemistry structures and properties of 134 kilo molecules*. Figshare. https://doi.org/10.6084/M9.FIGSHARE.C.978904.V5

Ravizza, S., Huschto, T., Adamov, A., Böhm, L., Büsser, A., Flöther, F. F., Hinzmann, R., König, H., McAhren, S. M., Robertson, D. H., Schleyer, T., Schneidinger, B., & Petrich, W. (2019). Predicting the early risk of chronic kidney disease in patients with diabetes using real-world data. *Nature Medicine*, *25*(1), 57–59. https://doi.org/10.1038/s41591-018-0239-8

Wang, X., Han, Y., Leung, V. C. M., Niyato, D., Yan, X., & Chen, X. (2020). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *22*(2), 869–904. https://doi.org/10.1109/comst.2020.2970550

Whitfield, J. D., Biamonte, J., & Aspuru-Guzik, A. (2011). Simulation of electronic structure hamiltonians using quantum computers. *Molecular Physics*, *109*(5), 735–750. https://doi.org/10.1080/00268976.2011.552441

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.