Exploring Quantum Machine Learning for High-Energy Physics

Jinghong Yang

DPF-PHENO 2024
University of Pittsburgh/and Carnegie Mellon University

UNIVERSITY OF MARYLAND

QLab NATIONAL QUANTUM LABORATORY | IONQ UNIVERSITY OF MARYLAND

Quantum Computing - Why are we interested?

Quantum Computing for High-Energy Physics
State of the Art and Challenges
Summary of the QC4HEP Working Group

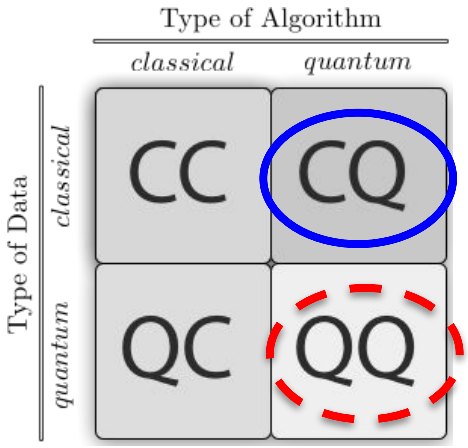
Alberto Di Meglio,[1,*] Karl Jansen,[2,3,†] Ivano Tavernelli,[4,‡] Constantia Alexandrou,[5,3] Srinivasan Arunachalam,[6] Christian W. Bauer,[7] Kerstin Borras,[8,9] Stefano Carrazza,[10,1] Arianna Crippa,[2,11] Vincent Croft,[12] Roland de Putter,[6] Andrea Delgado,[13] Vedran Dunjko,[12] Daniel J. Egger,[4] Elias Fernández-Combarro,[14] Elina Fuchs,[1,15,16] Lena Funcke,[17] Daniel González-Cuadra,[18,19] Michele Grossi,[1] Jad C. Halimeh,[20,21] Zoë Holmes,[22] Stefan Kühn,[2] Denis Lacroix,[23] Randy Lewis,[24] Donatella Lucchesi,[25,26,1] Miriam Lucio Martinez,[27,28] Federico Meloni,[8] Antonio Mezzacapo,[6] Simone Montangero,[25,26] Lento Nagano,[29] Voica Radescu,[30] Enrique Rico Ortega,[31,32,33,34] Alessandro Roggero,[35,36] Julian Schuhmacher,[4] Joao Seixas,[37,38,39] Pietro Silvi,[25,26] Panagiotis Spentzouris,[40] Francesco Tacchino,[4] Kristan Temme,[6] Koji Terashi,[29] Jordi Tura,[12,41] Cenk Tüysüz,[2,11] Sofia Vallecorsa,[1] Uwe-Jens Wiese,[42] Shinjae Yoo,[43] and Jinglei Zhang[44,45]

Quantum computers offer an intriguing path for a paradigmatic change of computing in the natural sciences and beyond, with the potential for achieving a so-called quantum advantage, namely a significant (in some cases exponential) speed-up of numerical simulations. The rapid development of hardware devices with various realizations of qubits enables the execution of small scale but representative applications on quantum computers. In particular, the high-energy physics community plays a pivotal role in accessing the power of quantum computing, since the field is a driving source for challenging computational problems. This concerns, on the theoretical side, the exploration of models which are very hard or even impossible to address with classical techniques and, on the experimental side, the enormous data challenge of newly emerging experiments, such as the upgrade of the Large Hadron Collider. In this roadmap paper, led by CERN, DESY and IBM, we provide the status of high-energy physics quantum computations and give examples for theoretical and experimental target benchmark applications, which can be addressed in the near future. Having the IBM 100 ⊗ 100 challenge in mind, where possible, we also provide resource estimates for the examples given using error mitigated quantum computing.

Quantum Computing for High-Energy Physics **(arXiv:2307.03236v1 [quant-ph] 6 Jul 2023)**

Outline

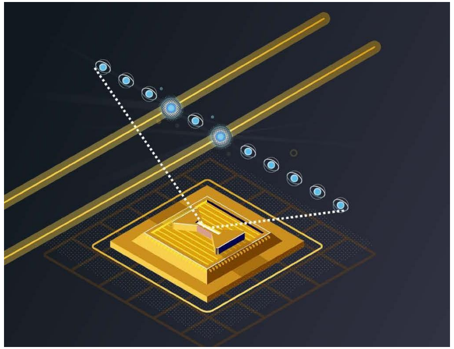
- Introduction to quantum computing
- Machine learning (ML) and quantum machine learning (QML)
- **Analysis 1 :** VQC for LHC data - example of QC/QML to analyze classical data
- **Analysis 2:** Quantum Generative Adversarial Network - example of QC/QML to analyze “quantum” data
- Future outlook



https://en.wikipedia.org/wiki/Quantum_machine_learning

Qubits, Gates, and Circuits

Physical Qubits



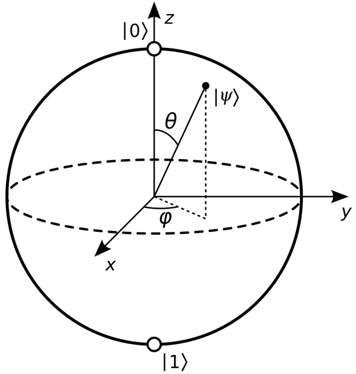
As an example, let's look at an ion trap quantum computer.

Two of the energy levels of the ion was used as the $|0\rangle$ state and $|1\rangle$ state.

Laser are used to control the qubits.

Qubits Representation in Bloch Sphere

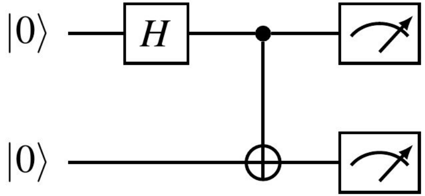
$$|\psi\rangle = \begin{pmatrix} \cos(\theta/2) \\ e^{i\phi}\sin(\theta/2) \end{pmatrix}$$



Quantum Computing Operations

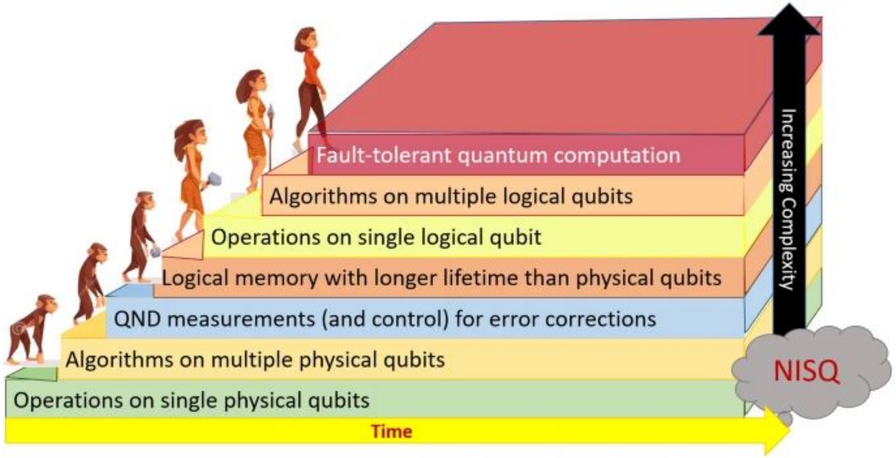
- Initialize
- Go through operations called “quantum gates”
- Measure

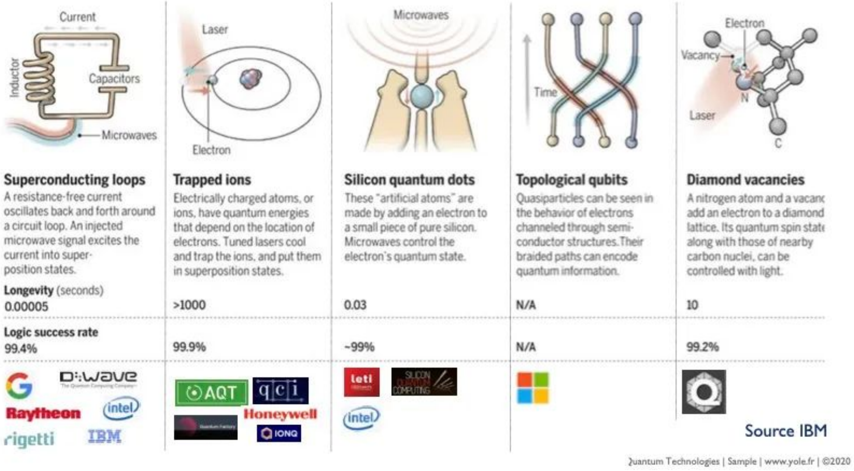
Hadamard, a single qubit gate



CNOT gate, a type of two-qubit gate

Noisy Intermediate-scale quantum era





[NISQ computing: where are we and where do we go? | AAPPS Bulletin]

[The Quantum Technology Ecosystem – Explained]

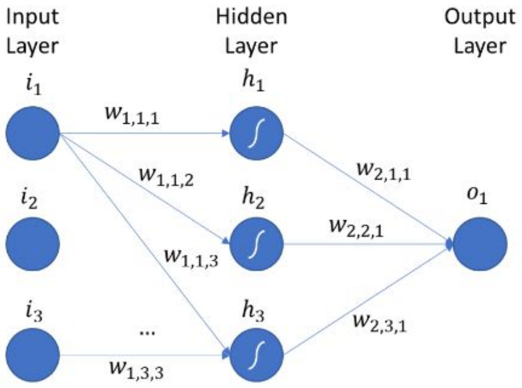
[Quantum Computing Modalities – A Qubit Primer Revisited]

[Accelerating Quantum Computing Readiness: Risk Management and Strategies for Sectors]

Machine Learning

Classical

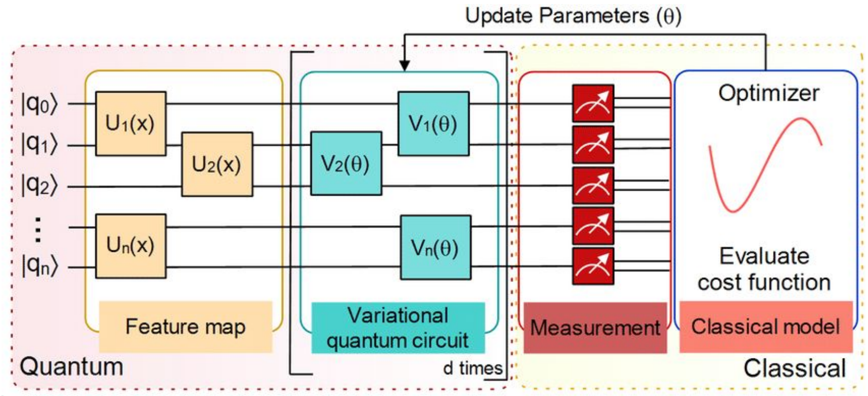
- Classical neural networks
- Parameters of neural network are varied to minimize loss function



https://editor.analyticsvidhya.com/uploads/9709833.png

Quantum

- Variational quantum circuit
- Gate parameters are varied to minimize loss function



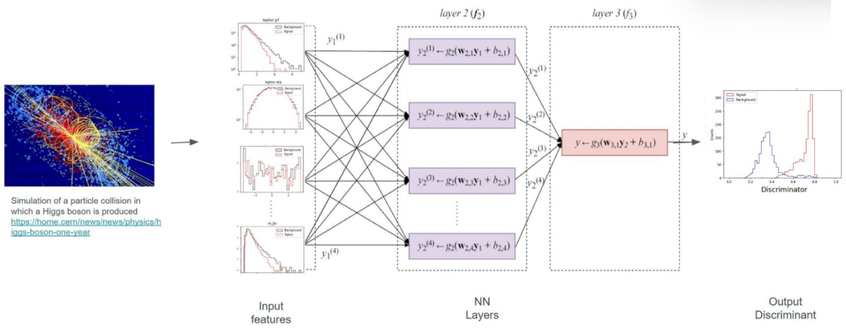
[2105.10162] Variational Quantum Classifiers Through the Lens of the Hessian

A Quantum Classifier

How can variational quantum circuit be used for supervised machine learning?

We need to have

- input $\vec{x}$
- function $f_{\vec{\theta}}$ with trainable parameters $\vec{\theta}$
- output y= $f_{\vec{\theta}}(\vec{x})$



https://themlbook.com/

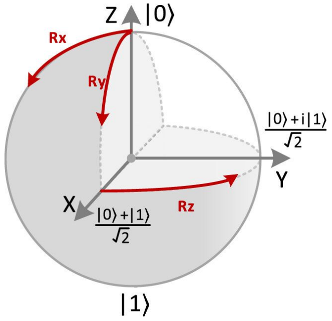
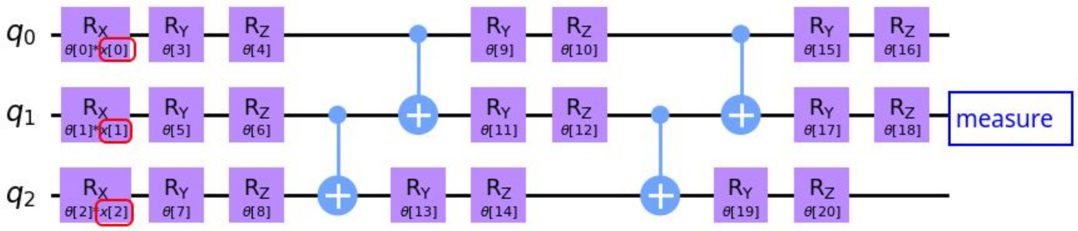
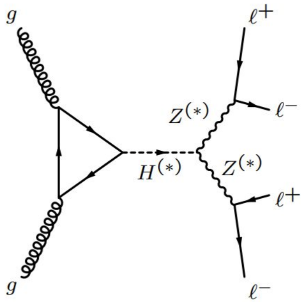


Image from http://dx.doi.org/10.1088/2058-9565/ace378

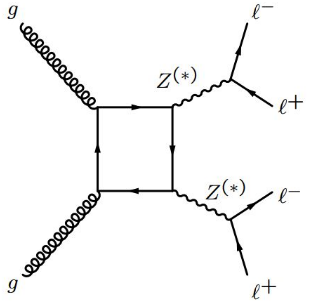


Analysis 1: VQC with LHC data

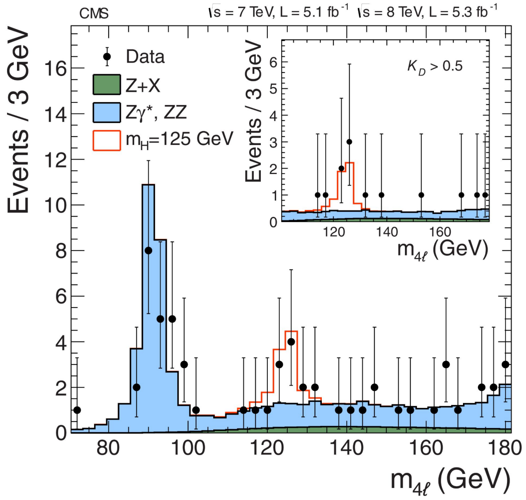
Signal



Background



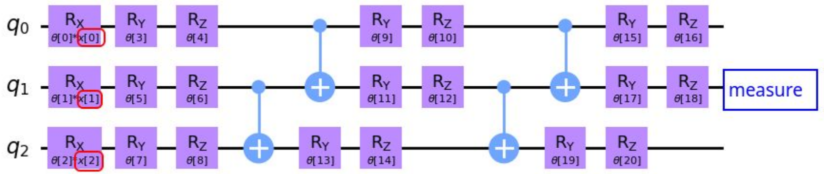
Phys. Lett. B 716 (2012) 30



- MC signal and background events similar to Higgs discovery analysis .
- H to ZZ to 4 leptons (muons in our case)

Features used

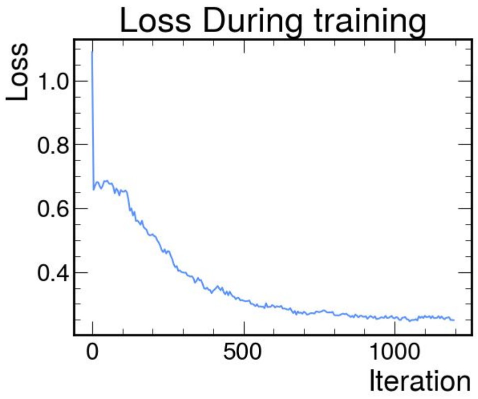
- Total invariant mass of 4 leptons
- Invariant mass of first Z boson
- Invariant mass of second Z boson

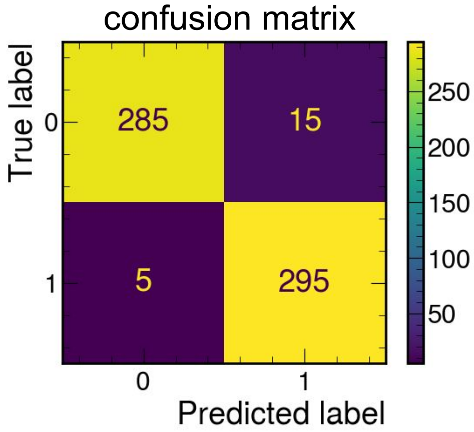


Results - Ideal simulation

Noiseless quantum circuit simulated on a classical computer

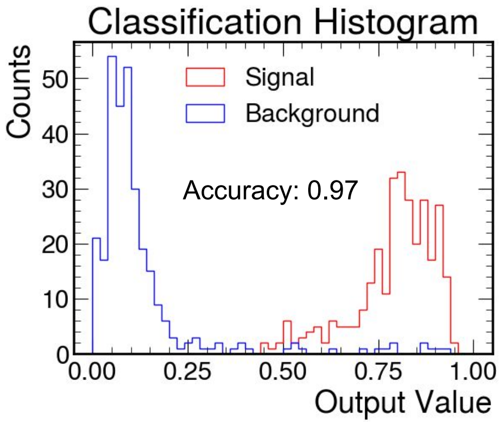
- Training set size: 2400
- Testing set size: 600
- 1 epoch
- Batch size 2
- SPSA optimization
- Loss function: binary cross entropy

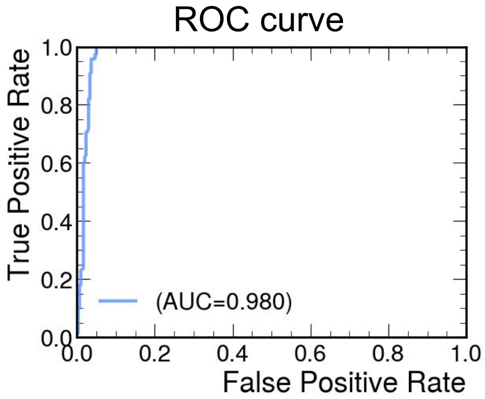




Discrimination plot

Smaller overlap between two peaks
=> better model performance



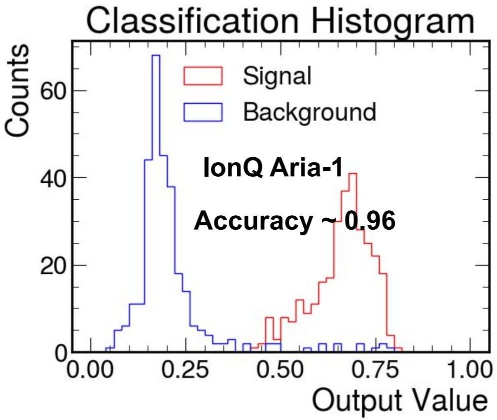


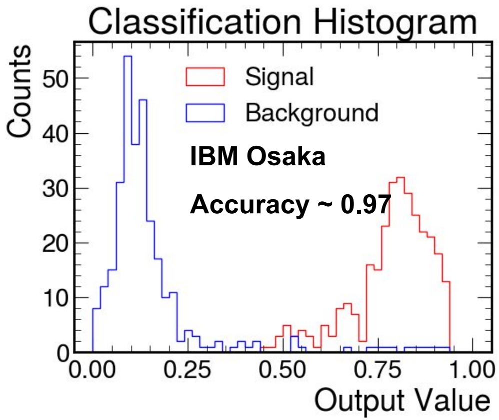
Results - Noisy Simulation

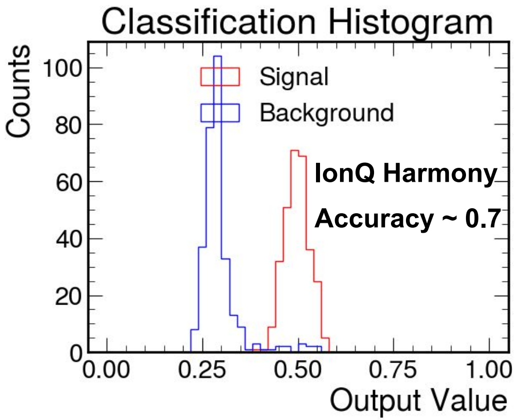
Classical simulation with hardware noise taken into account

- Noise models were used to characterize the hardware noise
- Noise models corresponding to 3 different quantum computers were tested

Other training settings are the same as the ideal simulation.





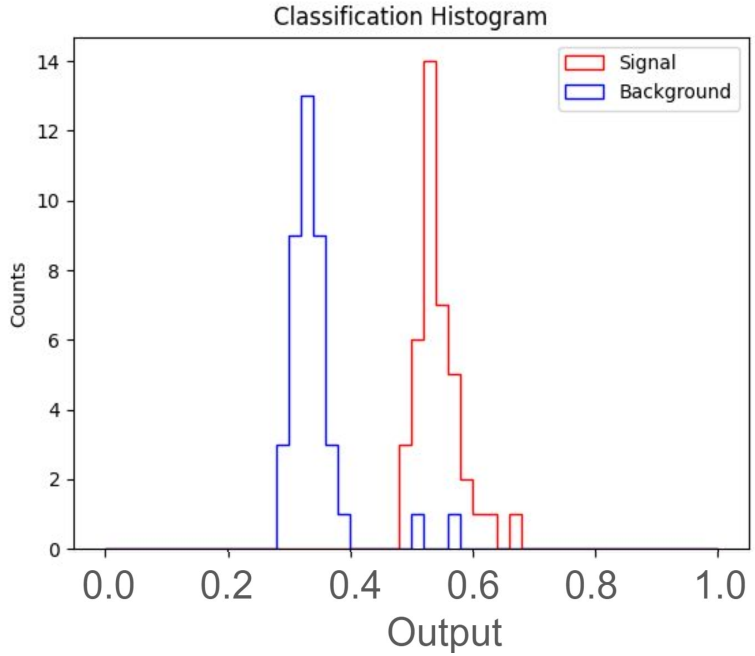


Using IBM Quantum cloud-based simulators
Get Started with Hardware Noise Model Simulation

Would it Work on Real Quantum Hardware?

Result on real quantum hardware

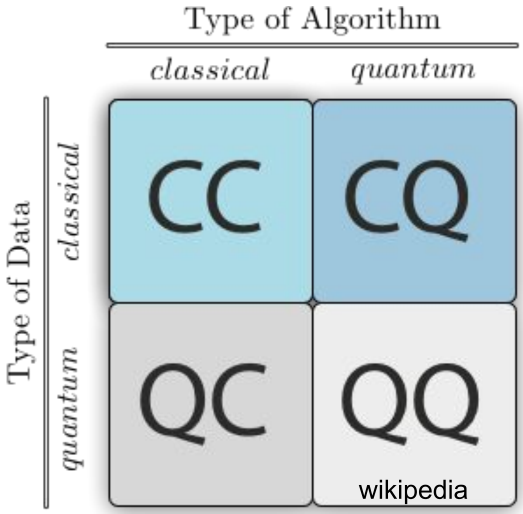
- IBM osaka 127 Qubits
- Training set size: 80
- Testing set size: 80
- 1 epoch
- Batch size 2
- SPSA optimization



A different model was used with same number of qubits and features

QML for “Quantum” Data

- QML may be more suited for quantum data.
- “Quantum memory” or “quantum storage” not yet available
- Maybe classical data generated by a quantum process also has some “quantumness”?
    - Generate toy data with “quantumness”
    - See whether QML can learn the pattern
- Quantum generative adversarial network (QGAN)



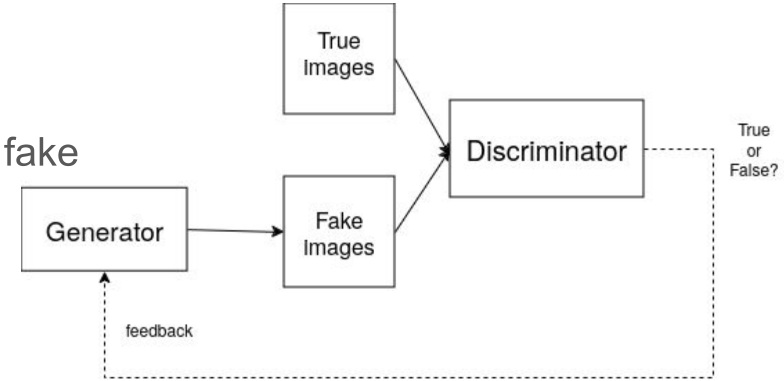
Ultimate goal is to use it for HEP physics processes

[2403.07059] Better than classical? The subtle art of benchmarking quantum machine learning models (https://arxiv.org/abs/2403.07059)

Quantum Generative Adversarial Network (QGAN)

GAN has two parts

- Generator generates fake images
- Discriminator tries to distinguish between real and fake images.

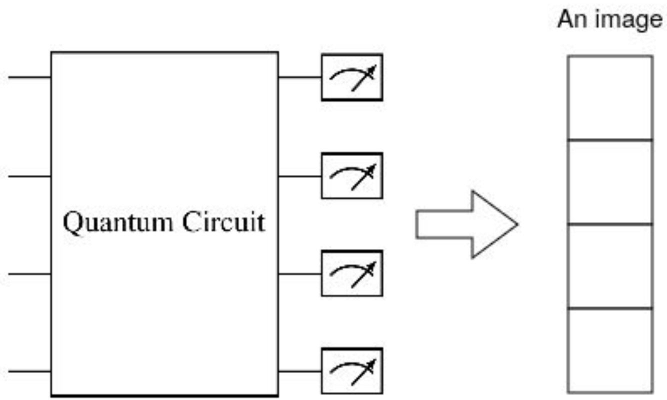


Quantum GAN

- Different approaches exist. Here, I focus on our approach.
- We use classical discriminator and quantum generator

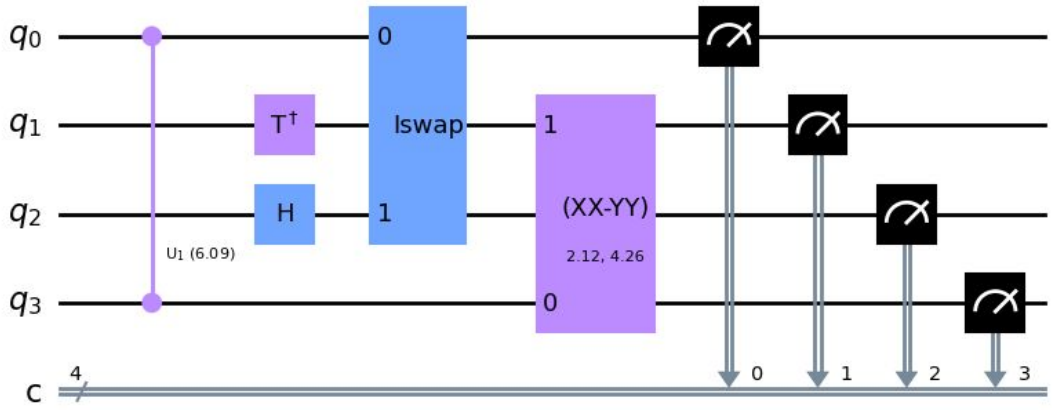
Measurement outcome => images

- Natively suited by binary images (0 or 1)
- For continuous data, use 2 or more qubits for 1 pixel.

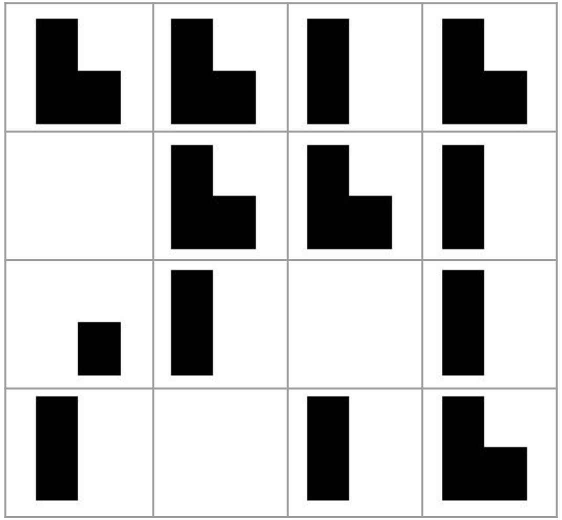


Analysis 2. QGAN for “Quantum” Toy Data

- Starting with toy data produced by a random quantum circuit
  - 4 qubits -> 4 pixel images
  - So that the data will have “quantum correlation”
- Test QGAN performance on the “quantum” toy data
- Compare with Classical GAN

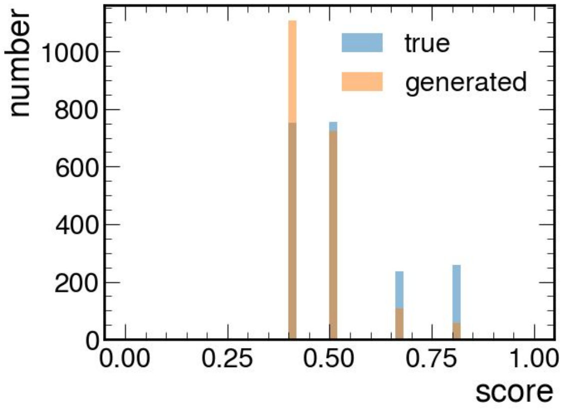


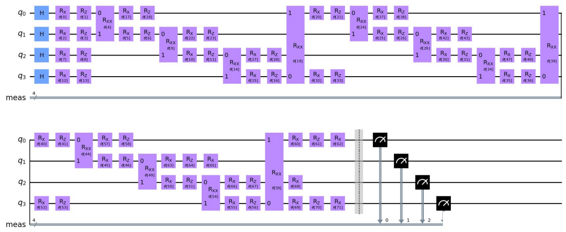
16 images
(each 2 by 2)



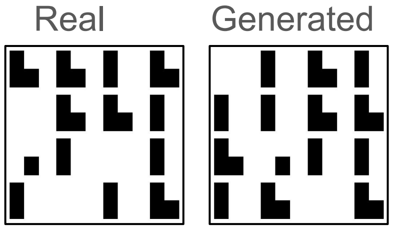
Quantum GAN with discrete toy data

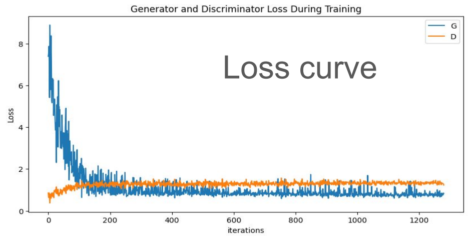
- QGAN trained on discrete toy data
  - Discriminator is MLP with 2 hidden layers
- Non-saturating GAN loss
- Quantum circuit simulated on a classical computer.





16 images
(each 2 by 2)





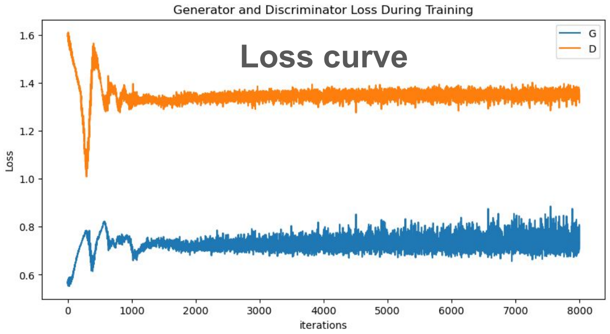
A separate classical neural network was used to visualize QGAN's performance.

(Less separated data => better GAN performance)

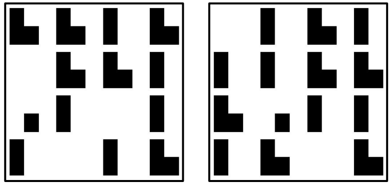
Classical GAN with Discrete Toy Data

- A classical GAN
  - Discriminator: MLP with 2 hidden layers
  - Generator: MLP with 2 hidden layers
- Non-saturating GAN loss

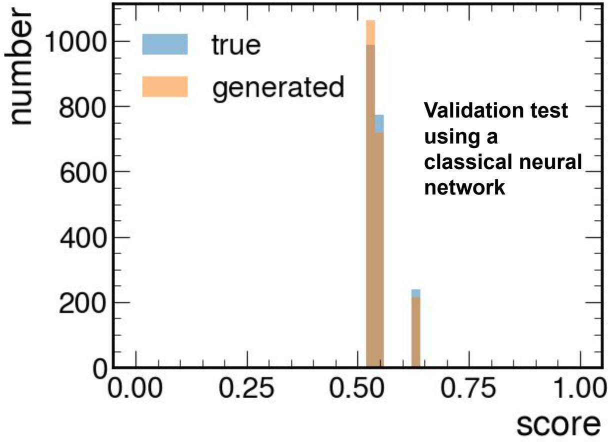
- A separate classical neural network was used to visualize GAN's performance.
  - Less separated data => better GAN performance



Real Generated



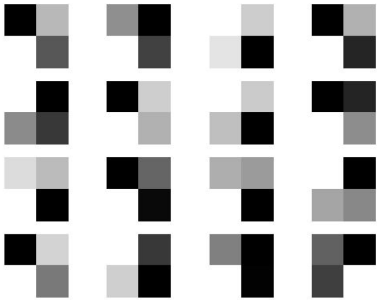
16 images
(each 2 by 2)



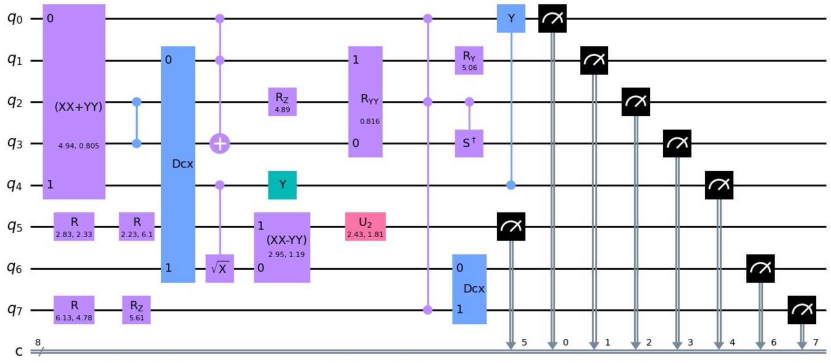
QGAN for "quantum" toy data - continuous

- Can we do the same with continuous data?
  - 1 qubit can only represent binary data
  - 2 qubits can represent 4 numbers
  - n qubits can represent 2^n numbers, approximating continuous data
- Here, we use 2 qubits for each pixel
  - Random variation is added to make the data more "continuous"

16 images
(each 2 by 2)

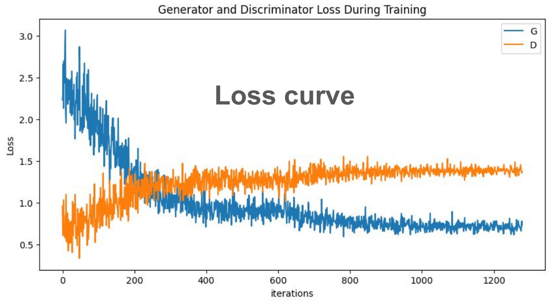


Test the performance of QGAN and classical GAN as before



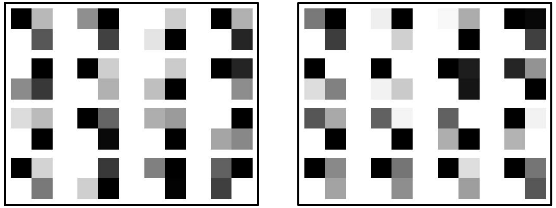
Quantum GAN with Continuous Toy Data

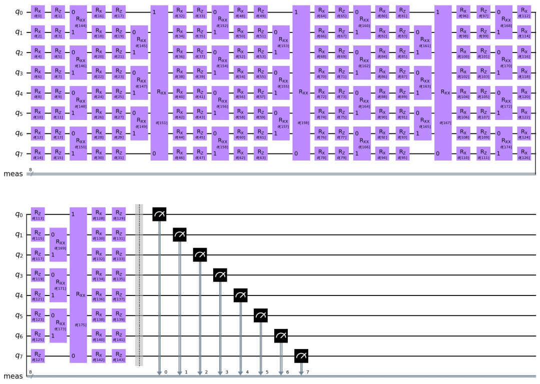
- QGAN trained on toy data generated by a random circuit.
  - Discriminator: MLP with 2 hidden layers
  - Non-saturating GAN loss
- A separate classical neural network was used to visualize GAN's performance.
  - Less separated data => better GAN performance



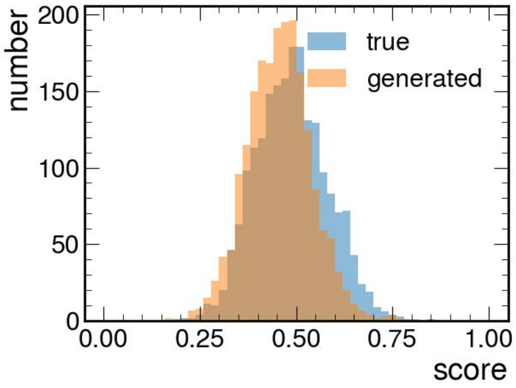
16 images
(each 2 by 2)

Real Generated



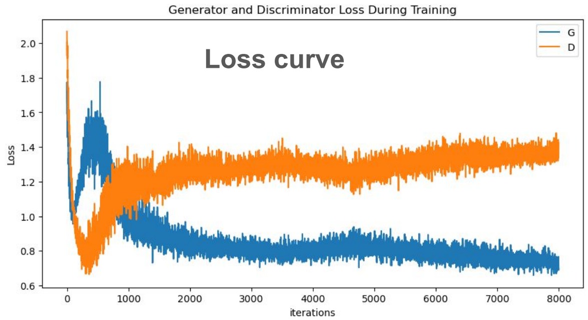


Validation test using a classical neural network

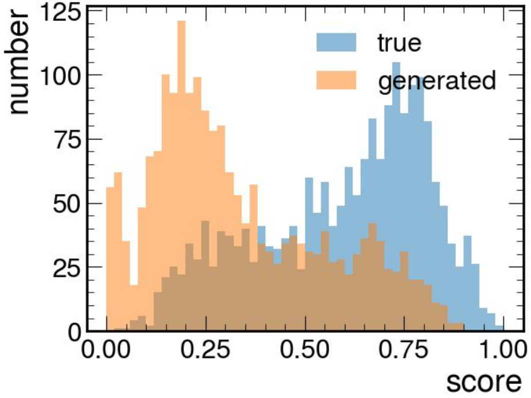


Classical GAN with Continuous Toy Data

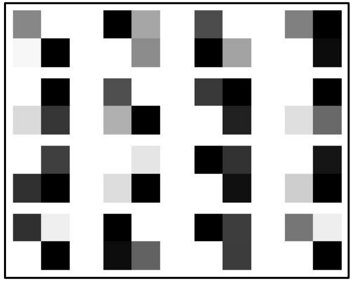
- Classical GAN trained on toy data generated by a random circuit.
  - Discriminator: MLP with 2 hidden layers
  - Generator: MLP with 2 hidden layers
  - Non-saturating GAN loss
- A separate classical neural network was used to visualize GAN's performance.
  - Less separated data => better GAN performance



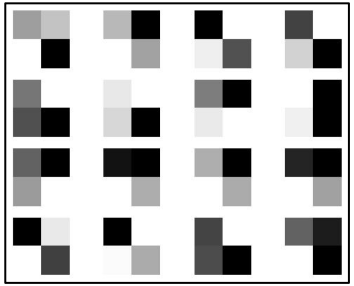
Validation test using a classical neural network



Real



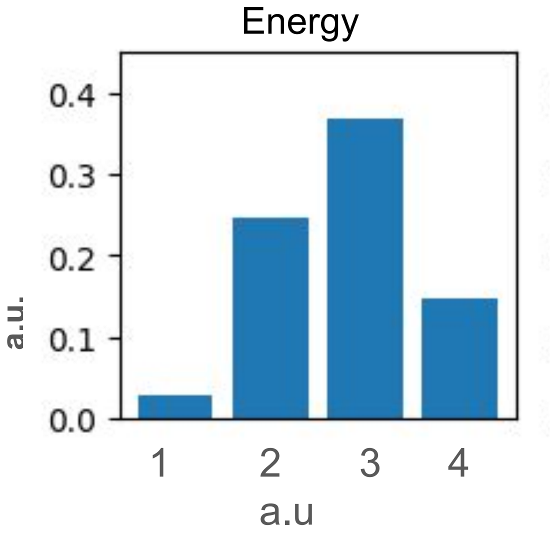
Generated



16 images
(each 2 by 2)

QGAN for Real Classical Continuous Data

- Let’s try QGAN on real data instead of toy data - QGAN for CLIC ECAL Showers
- We used the calorimeter images* and further downscaled it to 4 pixels.
- The value at each pixel represents the energy.
- We use 2 qubits for 1 pixel.



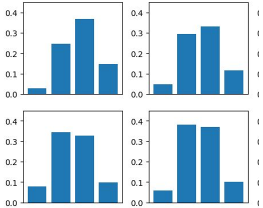
*[2305.07284] A Full Quantum Generative Adversarial Network Model for High Energy Physics Simulations
10.5281/zenodo.7025232

CLIC Calorimeter 3D images: Electron showers at Fixed Angle
[1912.06794] Calorimetry with Deep Learning: Particle Simulation and Reconstruction for Collider Physics

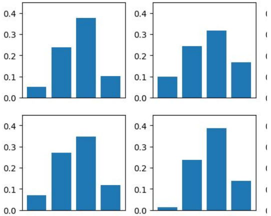
QGAN for CLIC ECAL Showers

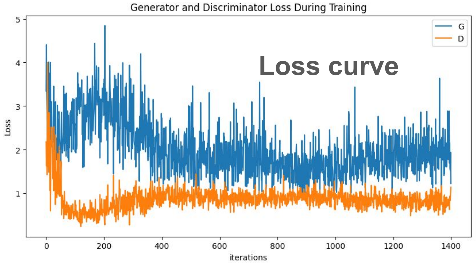
- QGAN Discriminator:
  MLP with 2 hidden layers
- Training set size ~ 1000
- Non-saturating GAN loss

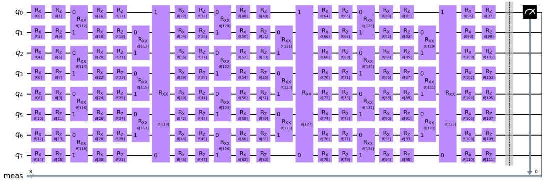
Real images

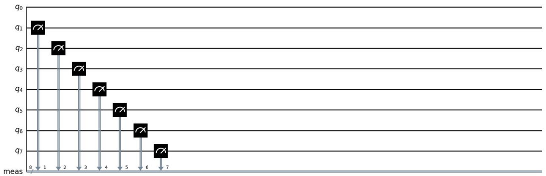


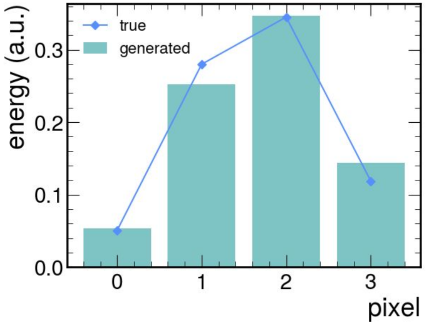
Generated images

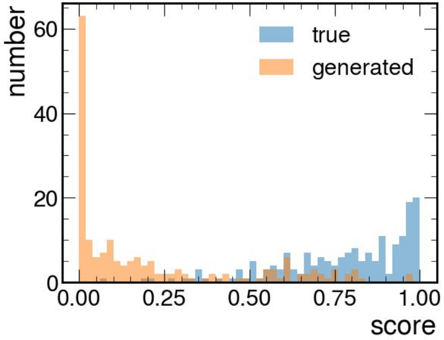












A separate classical neural network to visualize GAN performance

Conclusion and Outlook

- We've showcased preliminary studies employing Quantum Machine Learning (QML) for High-Energy Physics (HEP) data
- We will proceed to more sophisticated cases and search for suitable applications of QML in HEP
- We know Quantum computing has potential but better understanding is needed to unleash its full power.

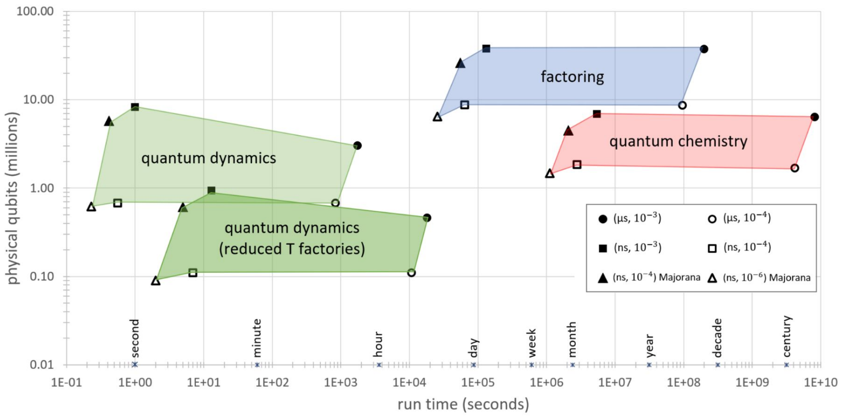


FIG. 3. Estimates of the resources required to implement three applications, assuming the qubit parameter examples specified in Table II. We explore a trade-off in the quantum dynamics application by considering two implementations: one which uses sufficient T factories to supply the needs of the shortest-depth algorithm and another which slows the algorithm down, allowing for a reduced number of T factories.

[Assessing requirements to scale to practical quantum advantage arXiv:2211.07629v1 [quant-ph] 14 Nov 2022]

Thank you!

We are just starting

Any comments, suggestions are most welcome!

Backup Slides

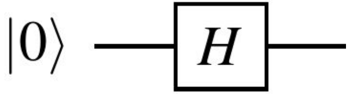
Supervised Machine Learning

- Basically, you have a dataset sample with input $\{\mathbf{x}_i\}$ and output $\{y_i\}$. You want to train $f_{\vec{\theta}}$ to map $\mathbf{x}_i$ to y_i.
- Linear regression: y = k x + b; minimizing chi-square
- Supervised machine learning is similar, but fancier
  - A lot more parameters to ensure flexibility
  - Rather than chi-square, may use other loss functions
  - Certain techniques (e.g. stochastic gradient) are used to find optimal parameters
- In classification tasks, y=1 for signal and y=0 for background.

Examples of quantum gates

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

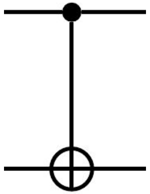
, where $|0\rangle$ and $|1\rangle$ are the basis vectors.



Therefore, the result of this circuit is:

$$H\,|0\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

Examples of quantum gates

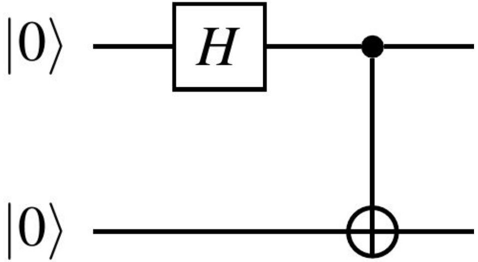


Controlled-NOT (CNOT)

When the control qubit is in $|0\rangle$, do nothing.

When the control qubit is in $|1\rangle$, flip the target qubit.

$$\text{CNOT}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$



Creates an entangled state

Optimization of trainable parameters

- In classical machine learning, one can use gradient descent to optimize parameters.
  - You have loss function to measure how good or bad $f_{\vec{\theta}}$ is
  - Calculate the gradient of loss function with respect to the parameters $\vec{\theta}$.
  - Change $\vec{\theta}$ according to the gradient
- For a quantum circuit, gradient calculation is not easy.
  - gradient calculation scales with O(n). n is the number of parameters.
  - One alternative is simultaneous perturbation stochastic approximation (SPSA)

Simultaneous Perturbation Stochastic Approximation

- An approximation of the gradient.
- In the phase space of parameters $\vec{\theta}$, pick a random direction $\vec{\Delta}$ according to some rules.
- For each iteration step,

$$\vec{g} = \frac{L(\vec{\theta} + c\vec{\Delta}) - L(\vec{\theta} - c\vec{\Delta})}{2c} \begin{pmatrix} \Delta_1^{-1} \\ \Delta_2^{-1} \\ \vdots \\ \Delta_n^{-1} \end{pmatrix}$$

$$\vec{\theta} \to \vec{\theta} - a\vec{g}$$

- Scale with $\mathcal{O}(1)$
- In short, use directional derivative rather than the gradient.

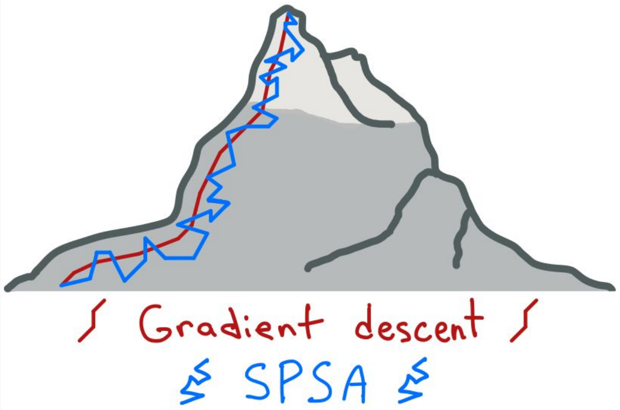


Image from pennylane

Different approaches of QGAN

There are different approaches to generate continuous data

- Expectation value approach
  - Evaluate the circuit for many times and estimate the expectation value of an operator.
  - E.g. 1901.00848, 2305.07284
- Multiple qubits approach
  - Use more than 1 qubit for each pixel
  - E.g. 2109.06315
  - Similar to Quantum Circuit Born Machine
- Others
  - E.g. 2103.15470

Using multiple qubits for 1 pixel

- 1 qubit – suitable for binary data
- 2 qubits
  - 00, 01, 10, 11
  - They can correspond to 1/8, 3/8, 5/8, and 7/8.
  - Then, we apply a uniform random distribution [-1/8, 1/8]
  - Thus, we can approximately represent floats between 0 and 1.
- 3 qubits and beyond
  - Similar strategy

Similar to:
2109.06315

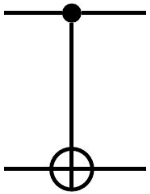
“Discrimination plots”

- After a GAN has been trained, we wish to visualize its performance.
- To achieve this purpose, we train another classical neural network to try to distinguish between the real data and the generated data.
    - If the classical neural network can easily distinguish between them, it means that the generator is not good enough.
    - If the classical neural network fails to effectively distinguish between them, it means the generator is successful.

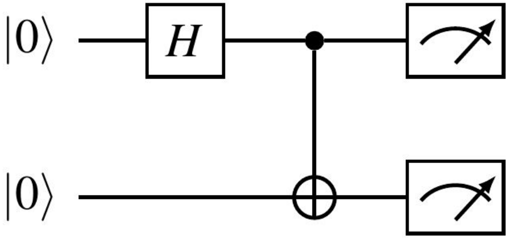
Qubits, Gates, and Circuits

a qubit

a single qubit gate



CNOT gate, a type of two-qubit gate



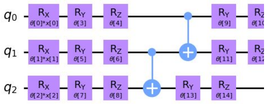
- In classical computers, there are AND, OR, XOR, etc.
- In quantum computers, there are quantum gates.

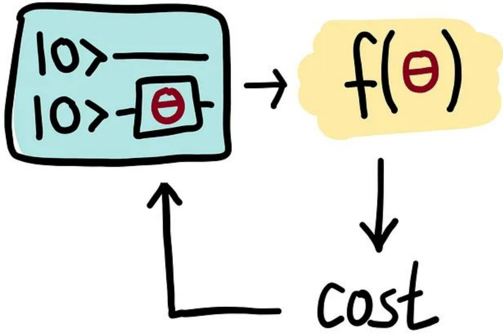
Quantum Machine Learning

https://www.semanticscholar.org/paper/Improving-Convergence-for-Quantum-Variational-using-K%C3%B6lle-Giovagnoli/ecdaf105d218fe8fb04aff83cabfbde2d48d4ecc

Quantum Machine Learning (QML)

- Variational quantum circuit
  - Contain parameterized gates
  - Gate parameters are trainable
- Training process
  - Vary the parameters based on certain algorithms
  - Minimize loss function





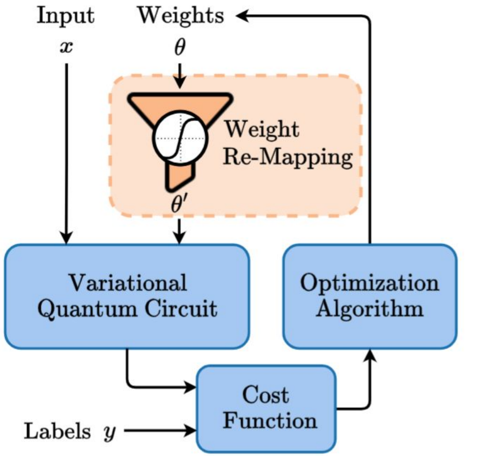
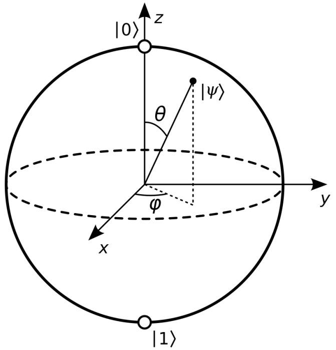


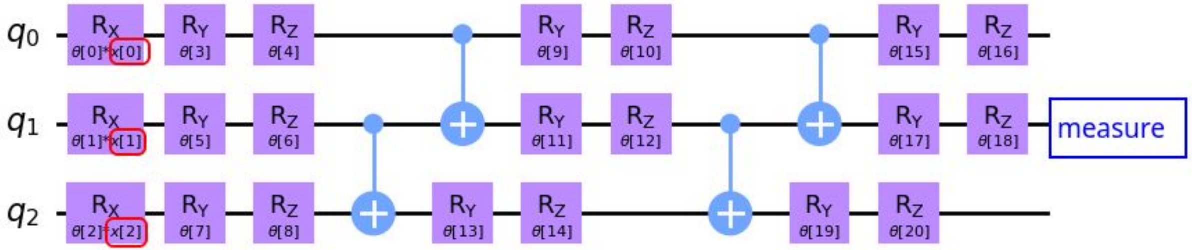
Figure 1: Overview of the Variational Quantum Circuit Training Process with Weight Constraints

https://senpinaki222.medium.com/optimizing-a-variational-quantum-circuit-studying-the-character-of-the-optimized-cost-as-a-a8bac2e9ba46

Quantum Encoding - inputting data into circuit

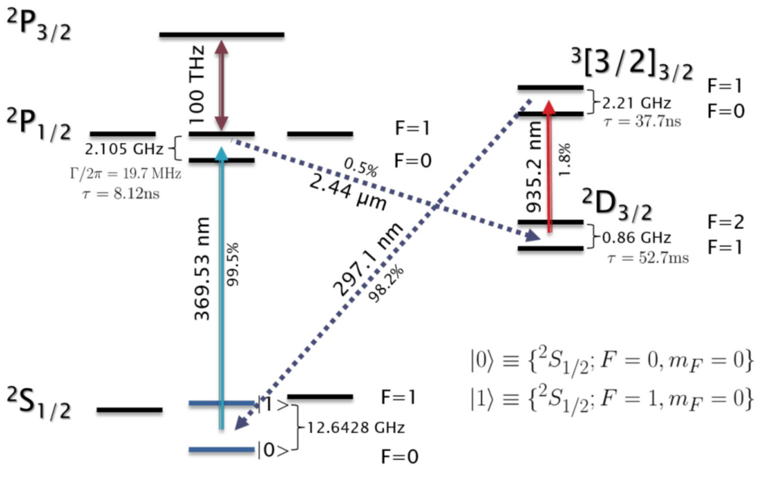
- Quantum circuit: map x to f(x)
- How does the quantum circuit take x as an input?
    - Encode the data using parameterized gates
    - E.g., for input feature x, we apply RX(k*x) gate on a qubit.
    - This put the qubit j in $|\psi\rangle = \begin{pmatrix} \cos(k^j x^j/2) \\ -i\sin(k^j x^j/2) \end{pmatrix}$ state.
    - n features, n qubits





Physics of Quantum Computers

As an example, let's look at an ion trap quantum computer.



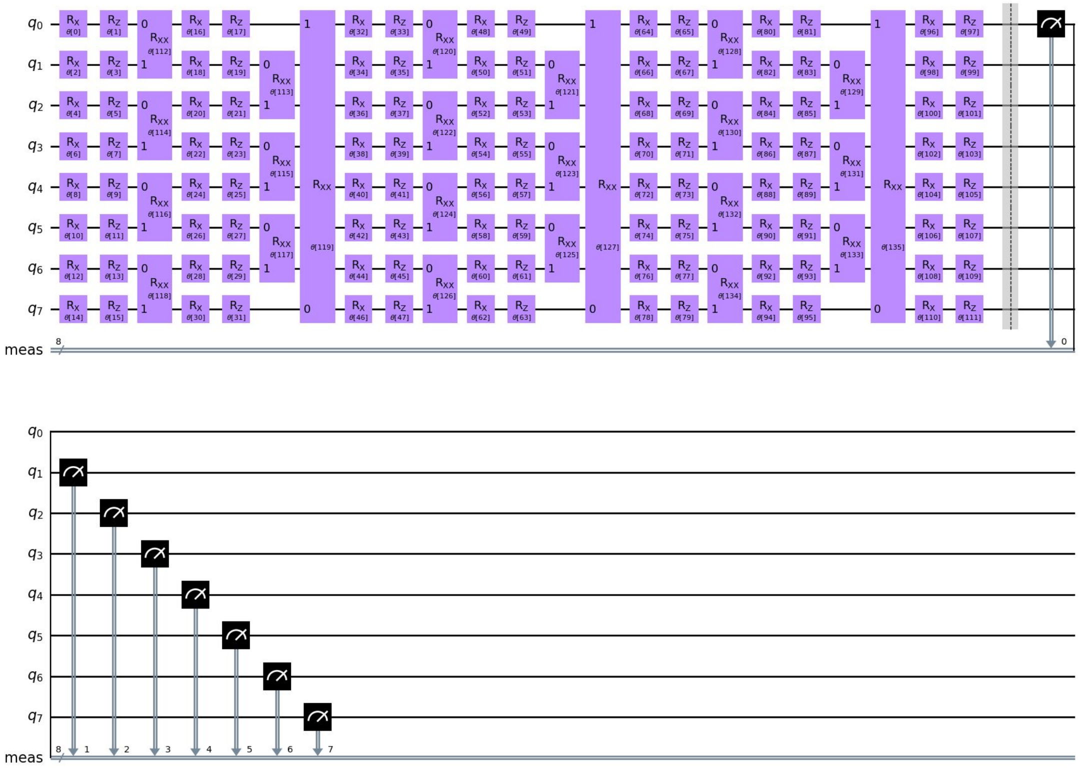
$|0\rangle \equiv \{{}^2S_{1/2}; F=0, m_F=0\}$

$|1\rangle \equiv \{{}^2S_{1/2}; F=1, m_F=0\}$

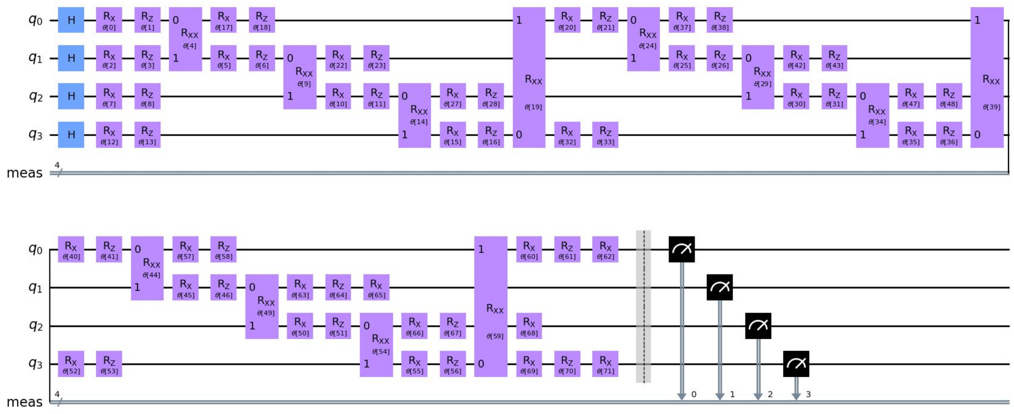
Figure 2.1: Energy levels of interest for the ${}^{171}\mathrm{Yb}^+$ qubit. The qubit is defined as $|0\rangle \equiv |F=0; m_F=0\rangle$ and $|1\rangle \equiv |F=1; m_F=0\rangle$.

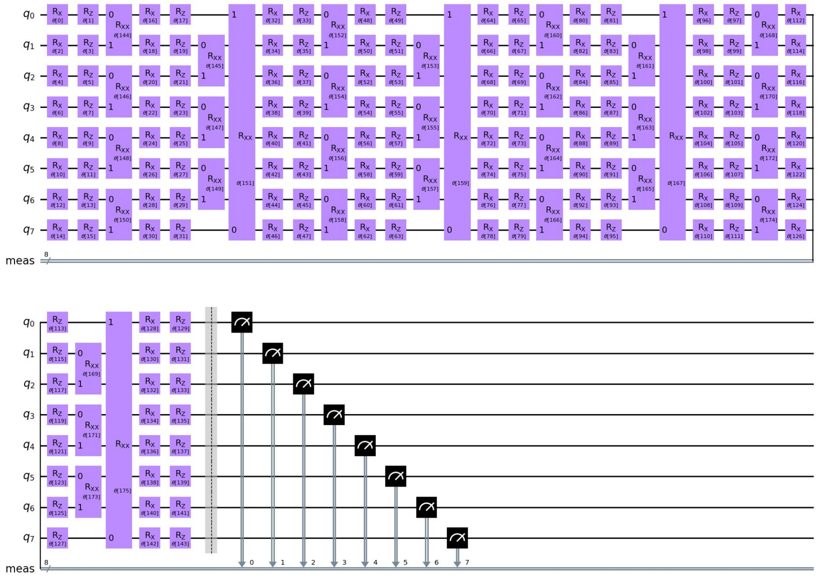
[Building and Programming a Universal Ion Trap Quantum Computer]

QGAN for CLIC ECAL Showers



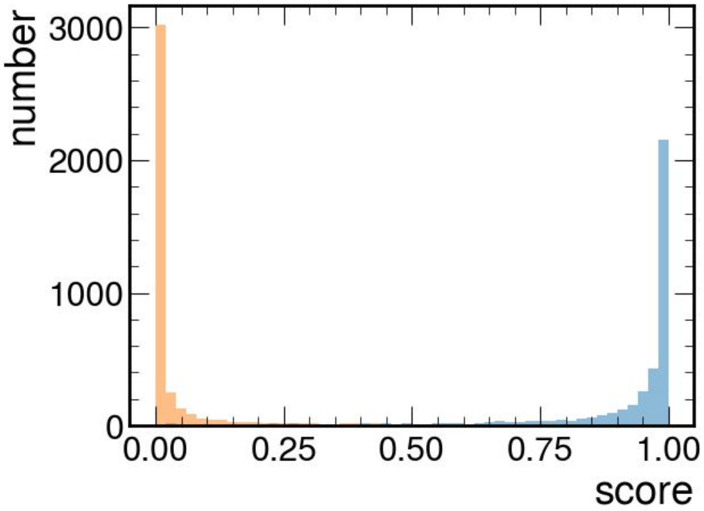
QGAN for “quantum” toy data





Sanity check for QGAN

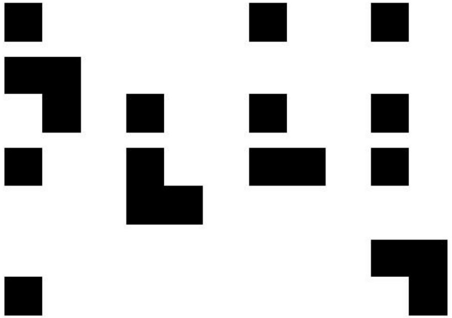
- For the continuous toy data, it is pure noise? Is there anything to learn for QGAN?
- While the output may look random, different random circuits will produce results that are distinguishable from each other.



QGAN with discrete toy data - another dataset

We also tried discrete "quantum" data with a more complex dataset.

16 real images
(each 2 by 2)



16 fake images
(each 2 by 2)