mPP ML JC:

arXiv:2005.08582

QUANTUM MACHINE LEARNING IN HIGH ENERGY PHYSICS

Thea Aarrestad (CERN)

18 Feb 2021
Virtual

QML in HEP

Today we'll discuss

1) A brief review of important concepts in QC/QML

2) How QML can be used in HEP through examples

FERMILAB-PUB-20-184-QIS

Quantum Machine Learning in High Energy Physics

Wen Guan[1], **Gabriel Perdue**[2], **Arthur Pesah**[3], **Maria Schuld**[4], **Koji Terashi**[5], **Sofia Vallecorsa**[6], **Jean-Roch Vlimant**[7]

[1] University of Wisconsin-Madison, Madison, WI, USA 53706
[2] Fermi National Accelerator Laboratory, Fermilab Quantum Institute, PO Box 500, Batavia, IL, USA 60510-0500
[3] Technical University of Denmark, DTU Compute, Lyngby, DK
[4] University of KwaZulu-Natal School of Chemistry and Physics, Durban, ZA 4000
[5] ICEPP, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, JP 300-1153
[6] CERN, IT, 1, Esplanade des Particules, Geneva, CH 1211
[7] California Institute of Technology, PMA, Pasadena, CA, USA 91125-0002

E-mail: `jvlimant@caltech.edu`

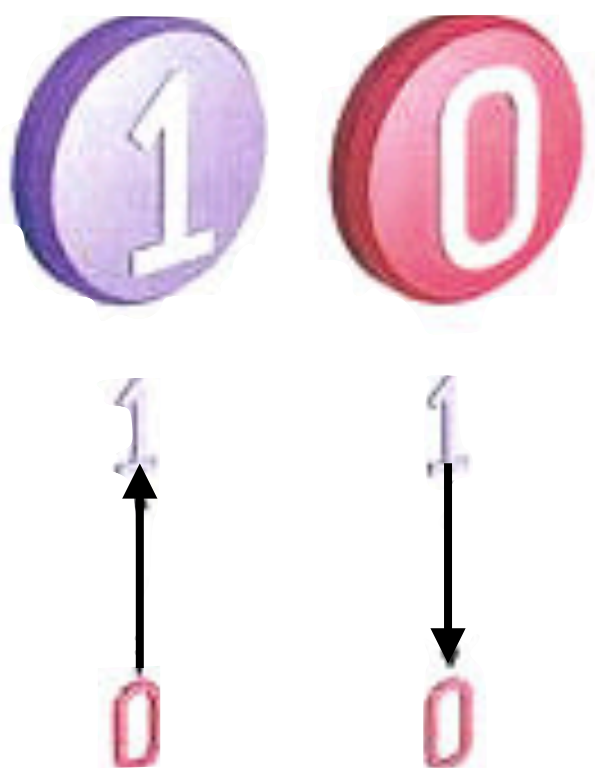
May 2020

Abstract. Machine learning has been used in high energy physics for a long time, primarily at the analysis level with supervised classification. Quantum computing was postulated in the early 1980s as way to perform computations that would not be tractable with a classical computer. With the advent of noisy intermediate-scale quantum computing devices, more quantum algorithms are being developed with the aim at exploiting the capacity of the hardware for machine learning applications. An interesting question is whether there are ways to apply quantum machine learning to High Energy Physics. This paper reviews the first generation of ideas that use quantum machine learning on problems in high energy physics and provide an outlook on future applications.

BASIC CONCEPTS

Overview:
Quantum Computing

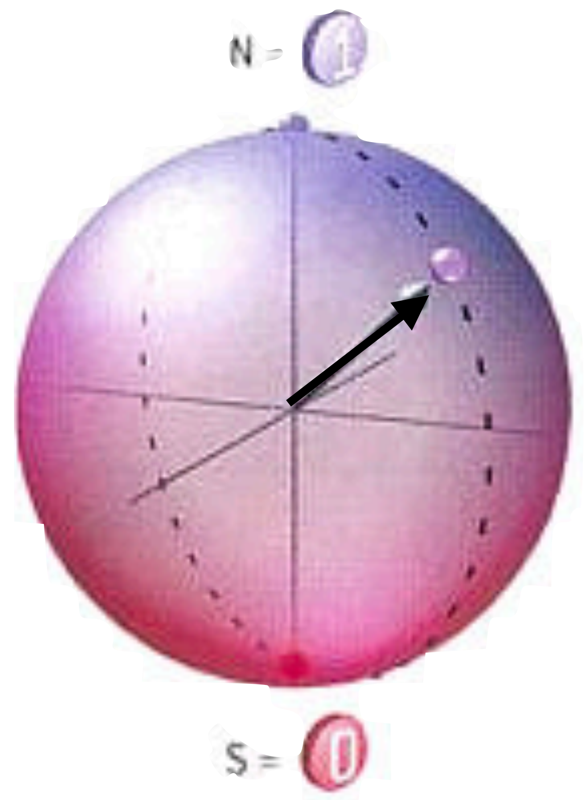
Classical bit:
2 states
(transistor on/off)



Ten bits = 2^{10} (1,024) combinations of 0s and 1s

Can represent 1 number between 0 and 1,023.

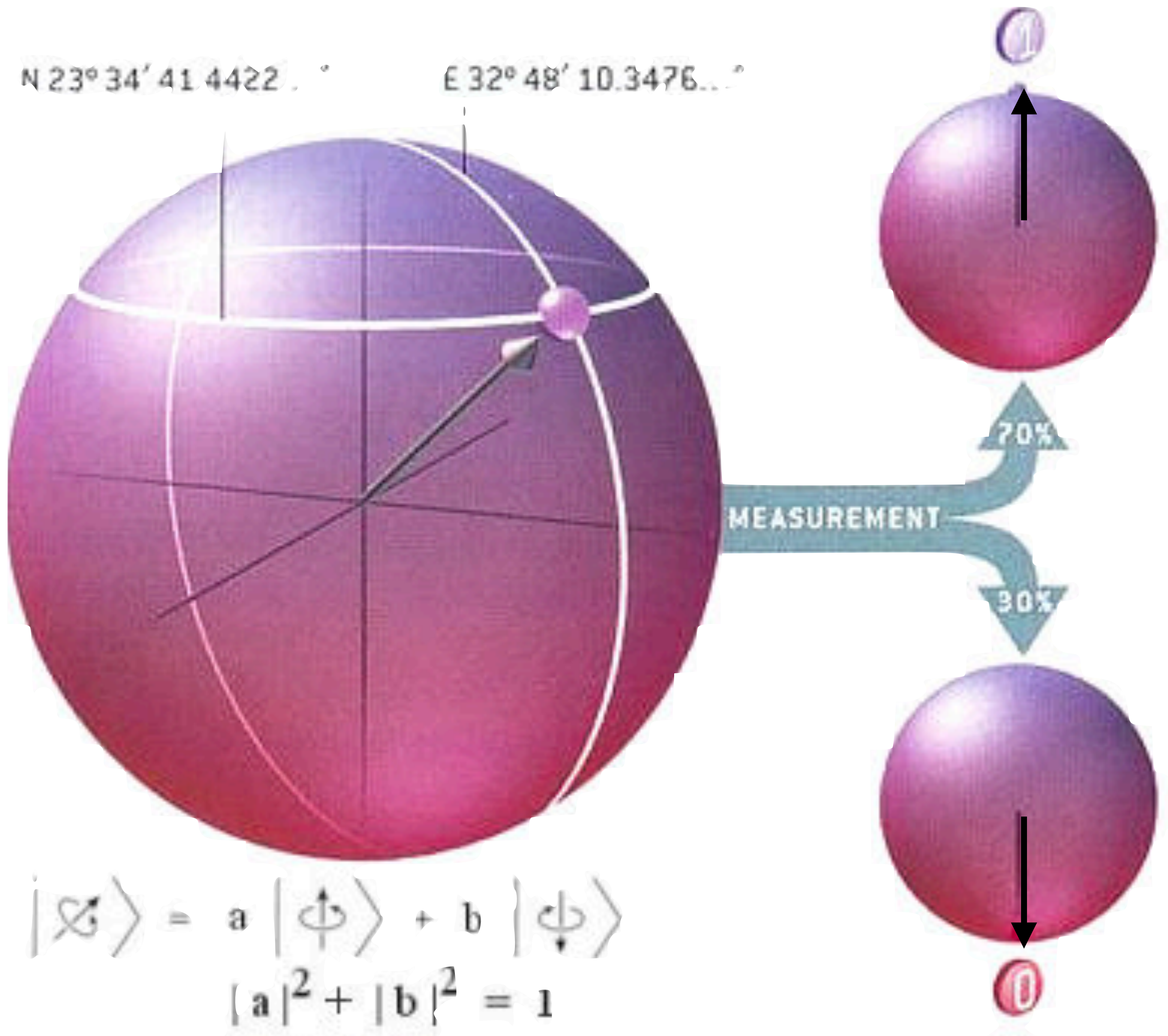
QUBIT:
Many possible states due to superpositions



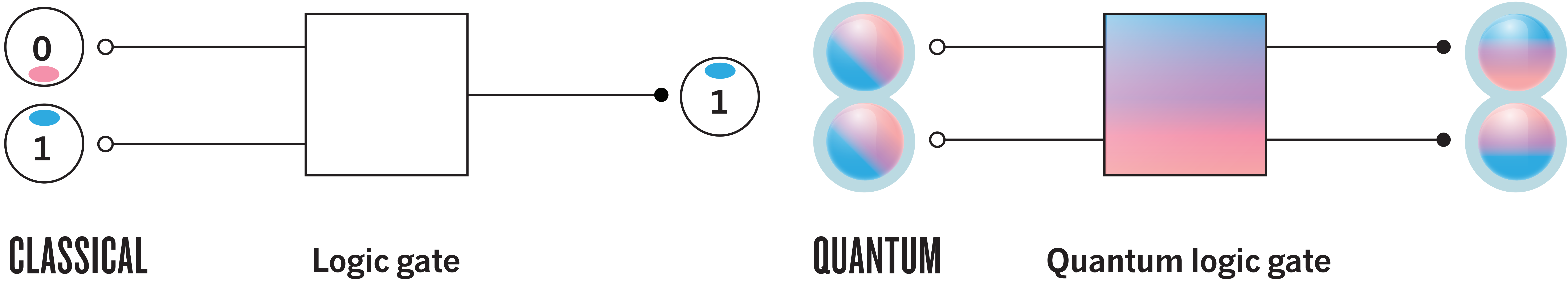
Represent 0 and 1 at the same time

10 qubits encode all 1,024 numbers simultaneously.

Measurement:
Yield outcome 0 or 1, probability depends on latitude



Overview:
Quantum Computing

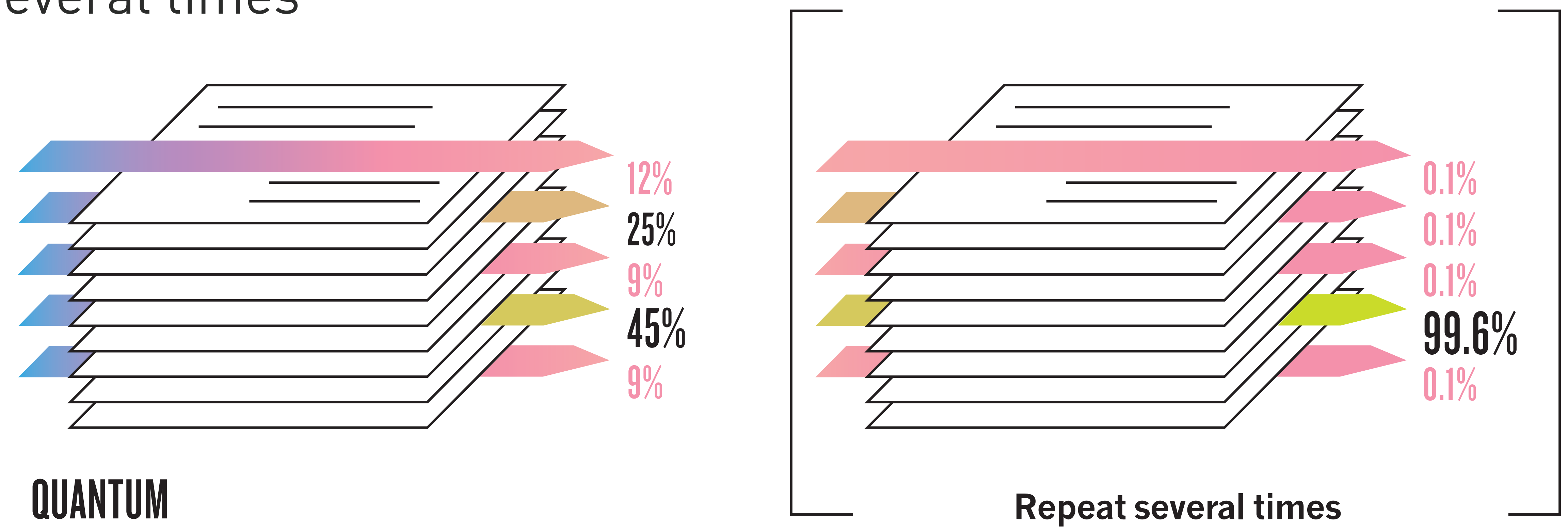


Classical logic gate:
Operates on single bits
Output 0 or 1

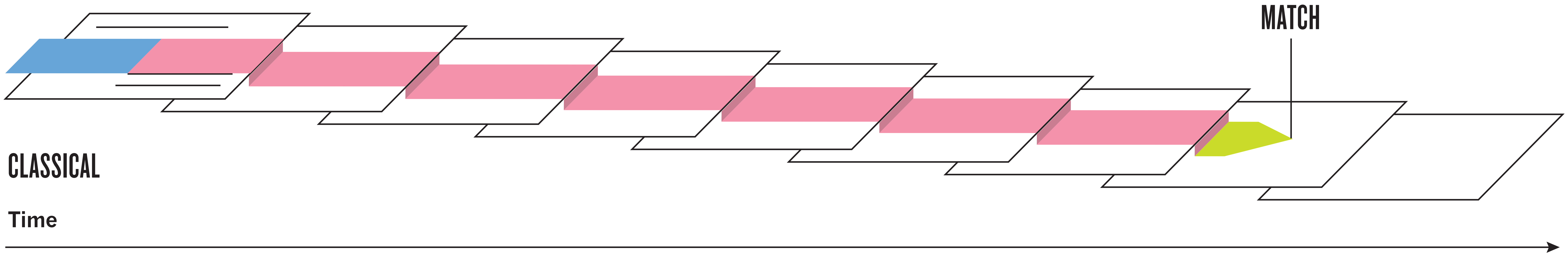
Quantum logic gate:
Operate on entire superposition state of all qubits
Transforms into another superposition state encoding all numbers

Overview:
Quantum Computing

Quantum computing:
All possibilities analysed at the same time, but must be repeated several times



Classical computing: Each possibility individually



Overview: Quantum Computing

This is really great, unfortunately we're no where near large universal gate model quantum computers

- Loss of coherence due to noise, adding qubits difficult, nearest-neighbour interaction only, error correction

Overview:
Quantum Computing

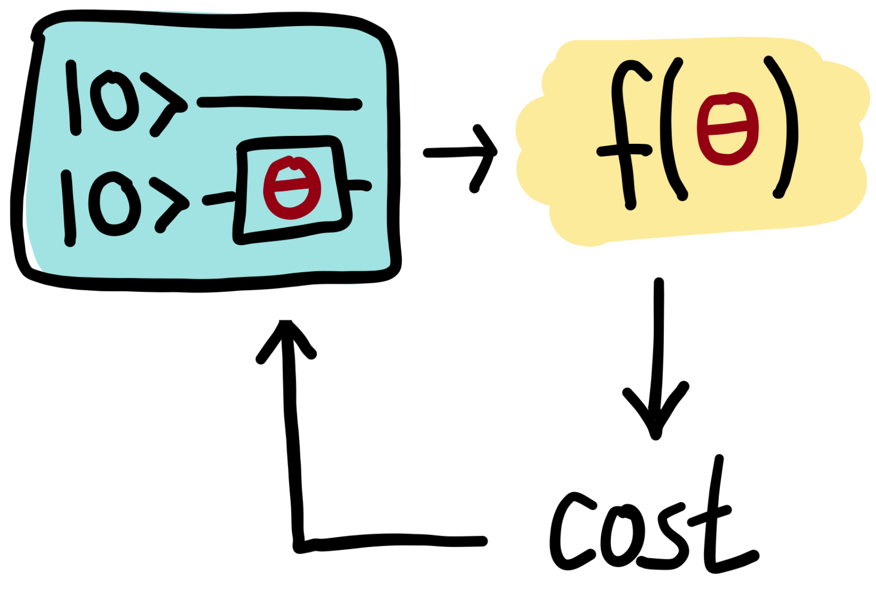
This is really great, unfortunately we're no where near large universal gate model quantum computers

- Loss of coherence due to noise, adding qubits difficult, nearest-neighbour interaction only, error correction

<u>Variational circuits (NISQ algorithms)</u>

Fixed size quantum circuits that depend on free parameters θ

- Ingredients: Preparation of fixed initial state, quantum circuit U(θ), and measurement of observable.
- Parameters can be optimised for specific task.
- Expectation value $f(\theta) = \langle 0 | U^{\dagger}(\theta)\hat{B}U(\theta) | 0 \rangle$ define cost
- Trained by classical optimisation algorithm



Overview:
Quantum Computing

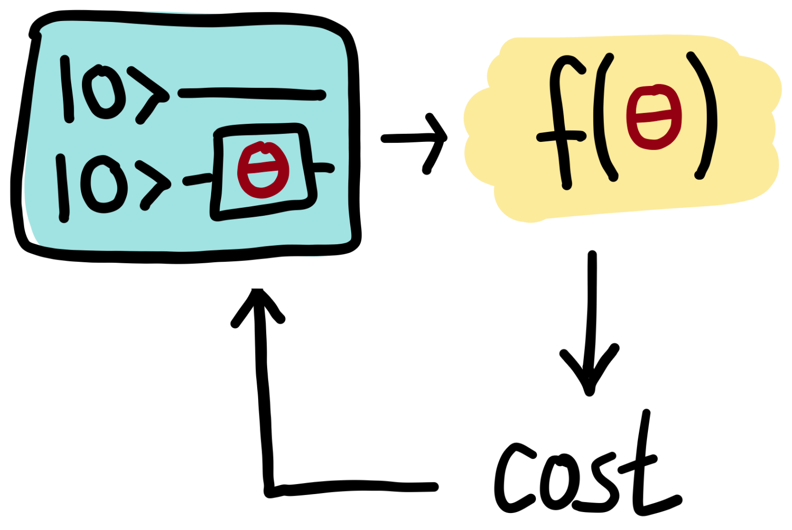
This is really great, unfortunately we're no where near large universal gate model quantum computers

- Loss of coherence due to noise, adding qubits difficult, nearest-neighbour interaction only, error correction

NISQ algorithms - Variational circuits

Fixed size quantum circuits with variable parameters θ

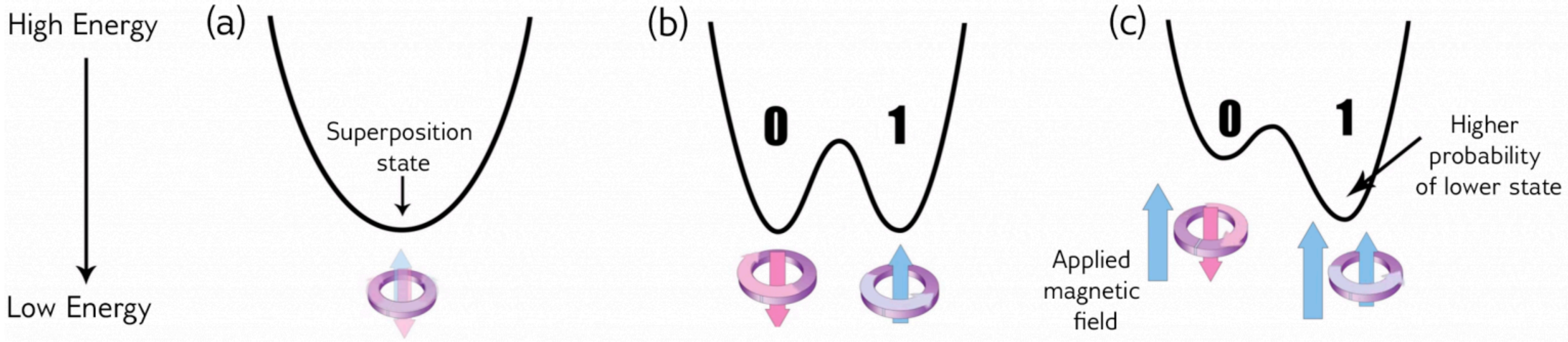
- Consists of fixed initial state, quantum circuit U(θ), and measurement of observable.
- Parameters can be optimised for specific task.
- Typically expectation value $f(\theta) = \langle 0|U^\dagger(\theta)\hat{B}U(\theta)|0\rangle$ define cost
- Trained by classical optimisation algorithm



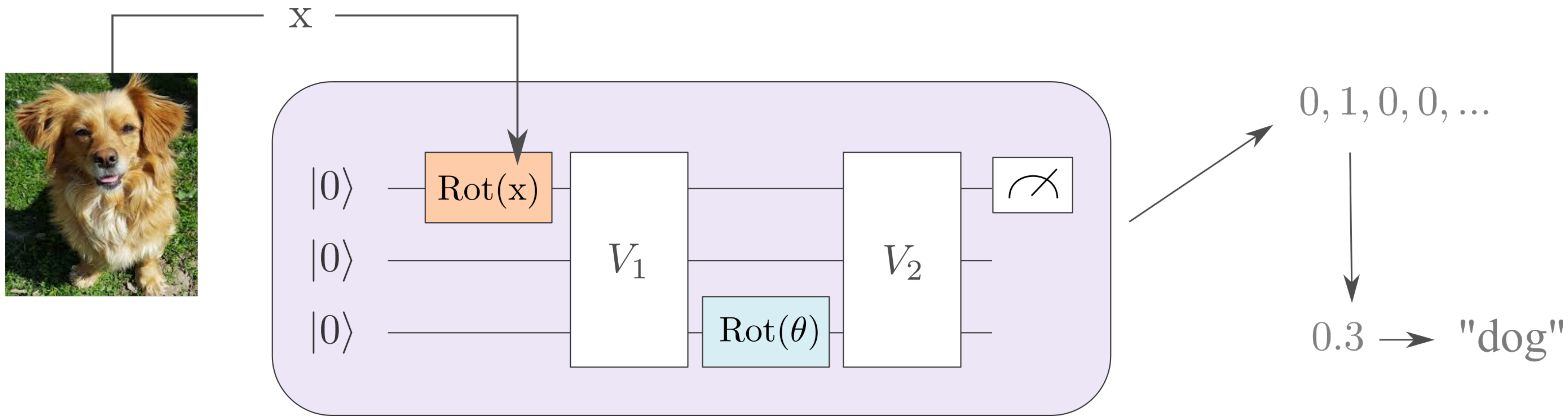
Quantum annealing

Solve optimisation problems by finding lowest energy state(minimum point over large number of variables)

- Formulate problem as objective function (Ising, QUBO)
- Each state represented as energy level, simulated and lowest energy result obtained
- Less affected by noise, but also less flexible
- Trainable bias(external field) and weights (coupler)



Overview:
Trainable variational circuits



$$f(\theta) = \langle 0 | U^{\dagger}(\theta) \hat{B} U(\theta) | 0 \rangle$$

Rot(x): Encode input, e.g rotate qubit of angle x
Rot(θ): Trainable parameters

Overview:
Trainable quantum annealers

Want to find lowest energy state of final Hamiltonian of system (Similar to Ising model):

Lowest-energy state of this is solution!
Includes ‘weights’ and ‘biases'

$$H_P = \sum_{i,j=1}^{n} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i=1}^{n} h_i\sigma_i^z$$

Overview:
Trainable quantum annealers

Want to find lowest energy state of final Hamiltonian of system (Similar to Ising model):

Lowest-energy state of this is solution!
Includes 'weights' and 'biases'

$$H_P = \sum_{i,j=1}^{n} J_{ij}\sigma_i^z \sigma_j^z + \sum_{i=1}^{n} h_i \sigma_i^z$$

Greatly simplified if initialising Hamiltonian in simple ground state, and evolving it slowly

$$H_I = \sum_{i=1}^{n} \sigma_i^x$$

all qubits in superposition state of 0 and 1

- Quantum adiabatic theorem: If the system begins close to an eigenstate, it remains close to an eigenstate
- Measuring final state solves optimisation

Overview: Trainable quantum annealers

Want to find lowest energy state of final Hamiltonian of system (Similar to Ising model):

Lowest-energy state of this is solution!
Includes ‘weights’ and ‘biases'

$$H_P = \sum_{i,j=1}^{n} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i=1}^{n} h_i\sigma_i^z$$

Greatly simplified if initialising Hamiltonian in simple ground state, and evolving it slowly

$$H_I = \sum_{i=1}^{n} \sigma_i^x$$

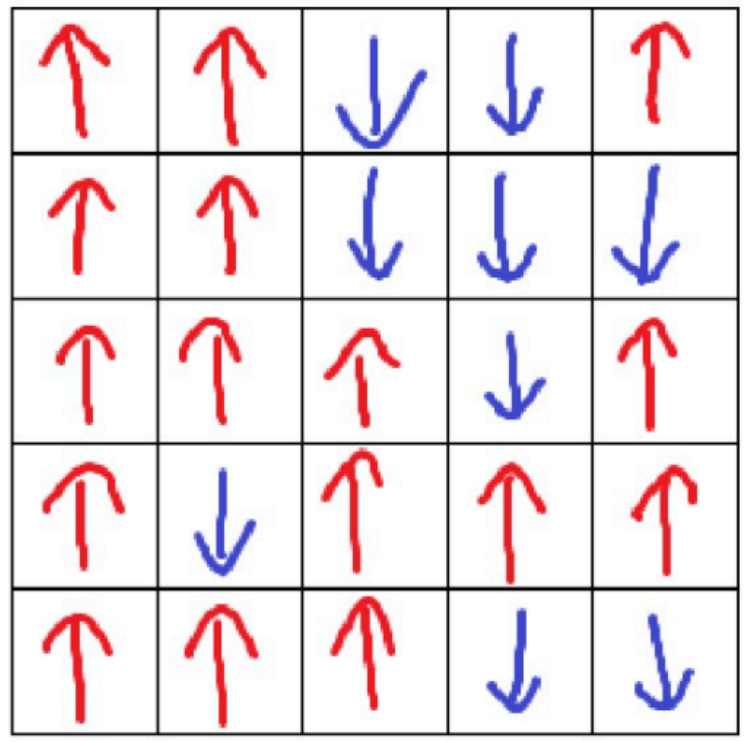
all qubits in superposition state of 0 and 1

- Quantum adiabatic theorem: If the system begins close to an eigenstate, it remains close to an eigenstate
- Measuring final state solves optimisation

To express problem to enable solution by minimization, formulate objective function (mathematical expression of system energy).

- Ising model or QUBO
- Low energy states == good solutions

$$\min_{\boldsymbol{x}} E(\boldsymbol{x}) = \sum_{i,j=1}^{n} \underbrace{J_{ij}}_{\text{‘weights’}} x_i x_j + \sum_{i=1}^{n} \underbrace{h_i}_{\text{‘biases’}} x_i$$

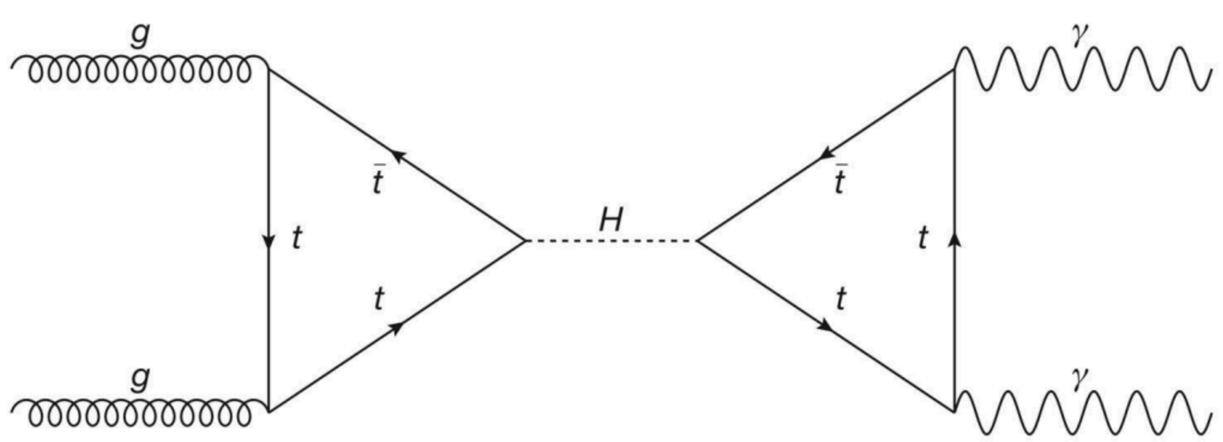


Quantum annealers for <u>training</u>, or as samplers?

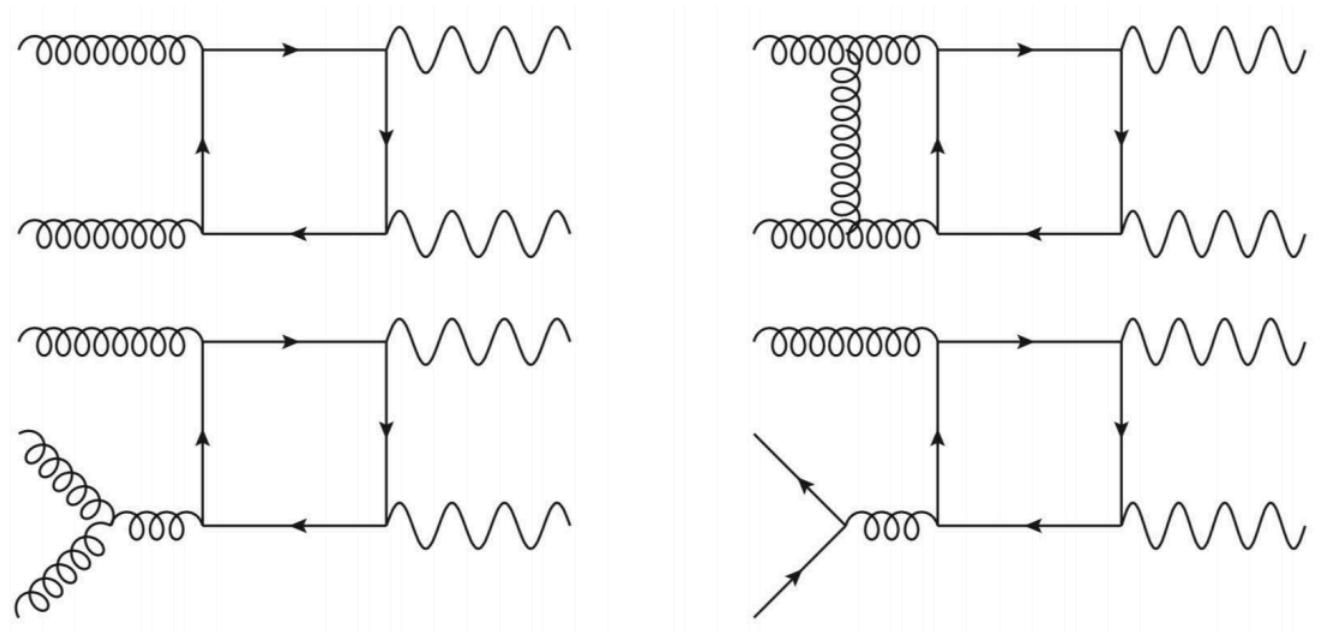
APPLICATIONS

Applications: Quantum Annealing

Higgs boson



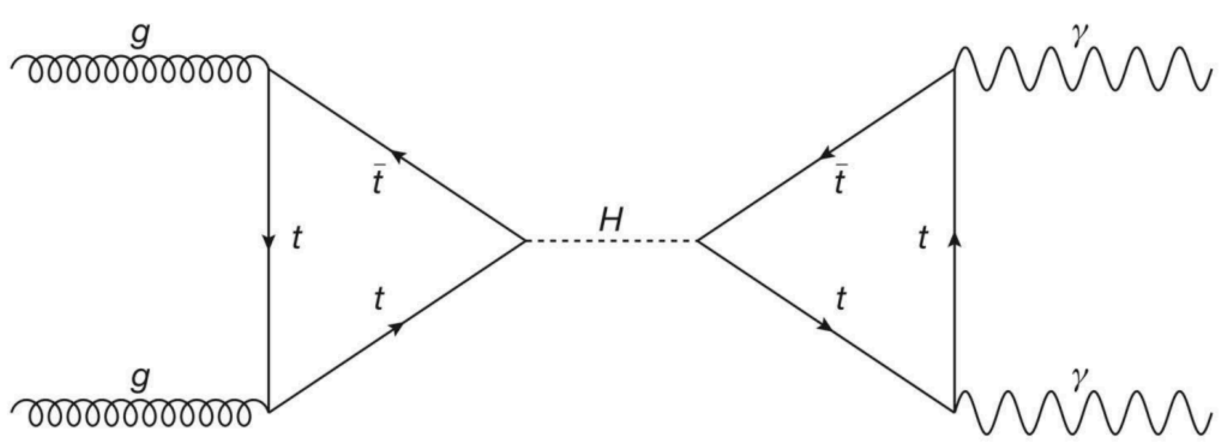
Other Standard Model (SM) processes



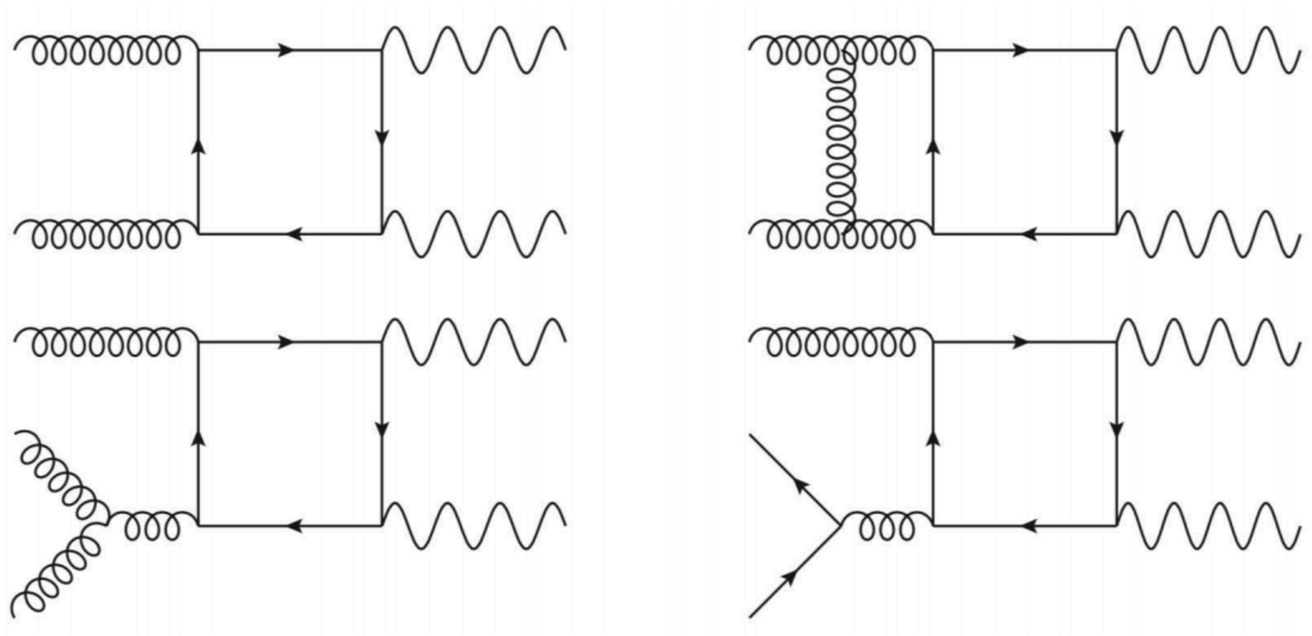
Di-photon event classification with quantum adiabatic machine learning (QAML)

Applications: Quantum Annealing

Higgs boson



Other Standard Model (SM) processes



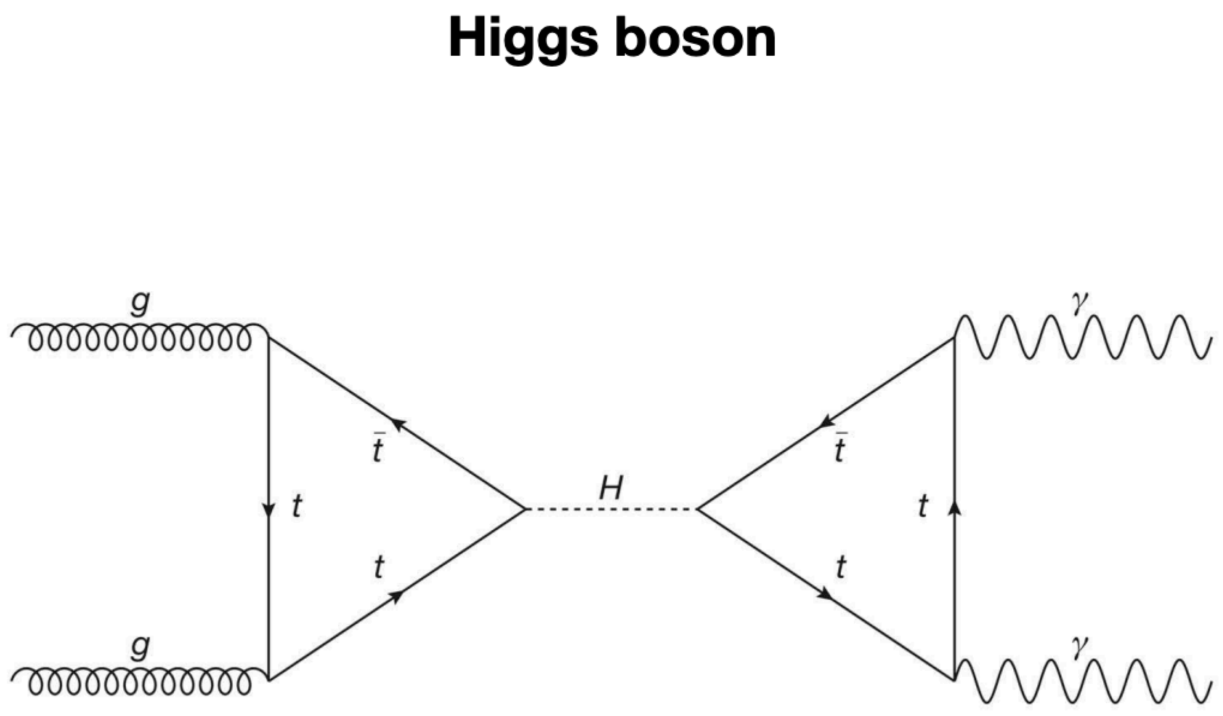
Di-photon event classification with quantum adiabatic machine learning (QAML)

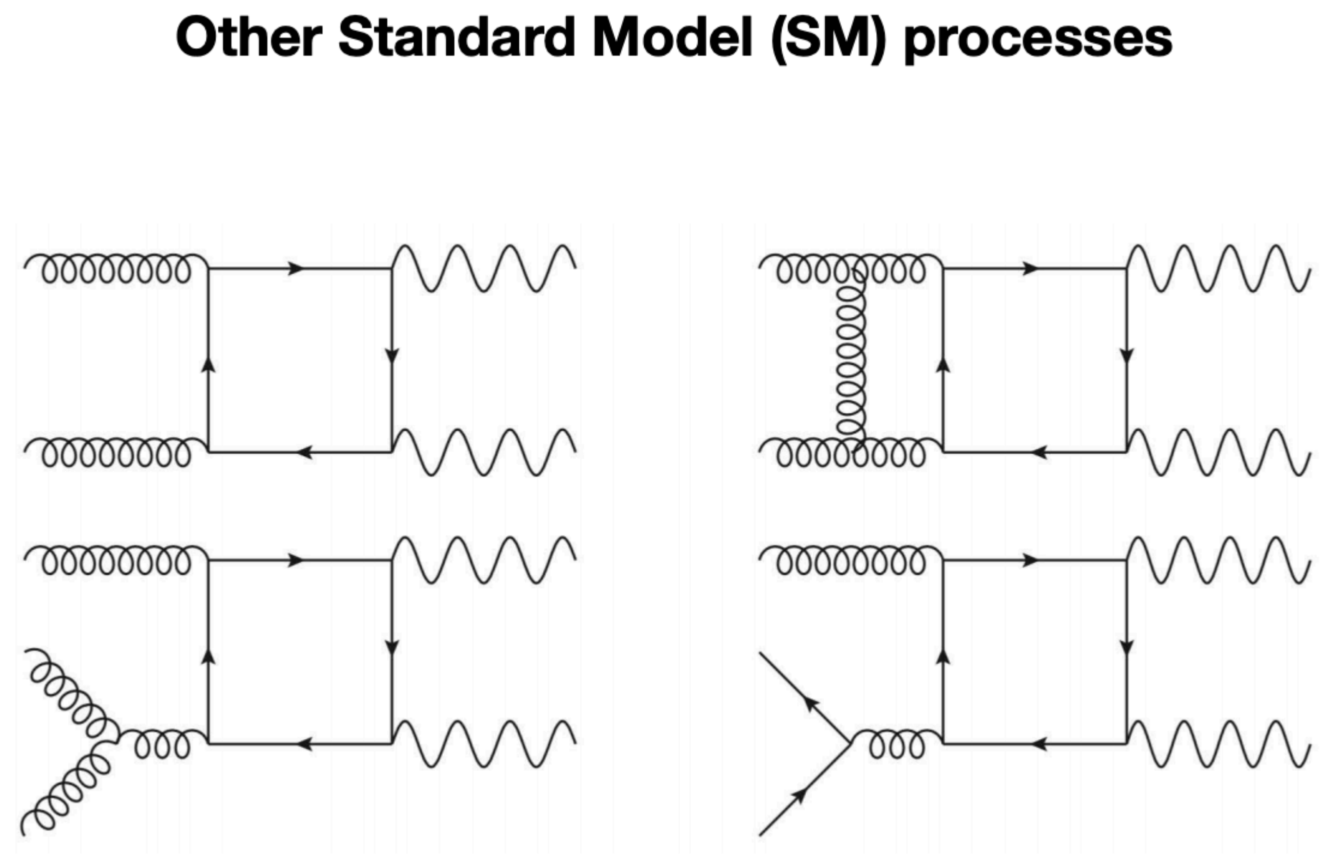
- Quantum annealing optimisation to find best linear combination of ‘weak classifiers’
- From 36 weak classifiers $c_i(x_\tau)$, build one strong classifier from linear combination with trainable ‘classifier weights’, **w**
- **Key:** How to formulate objective function so that best w’s are found

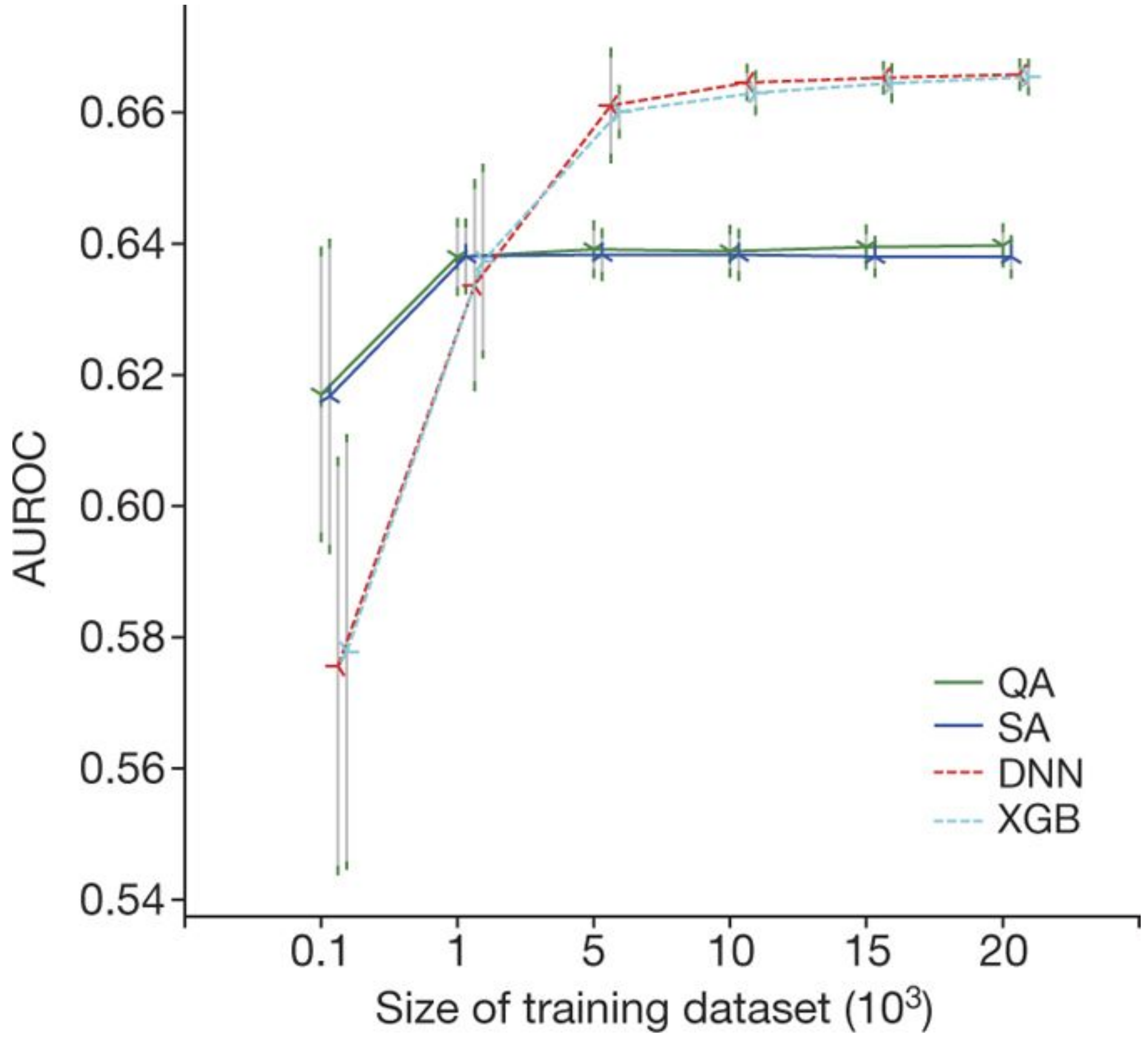
$$E(\boldsymbol{w}) = \sum_{i,j=1}^{n=36} C_{ij} w_i w_j + \sum_{i=1}^{n=36} 2(\lambda - C_i) w_i$$

$$C_{ij} = \sum_\tau c_i(x_\tau) c_j(x_\tau) \quad C_i = \sum_\tau c_i(x_\tau) y_\tau$$

Applications: Quantum Annealing







Also describes QAML Zooming (extend binary linear combination to continuous by running iteratively)

Di-photon event classification with quantum adiabatic machine learning (QAML)

- Quantum annealing optimisation to find best linear combination of 'weak classifiers'
- From 36 weak classifiers $c_i(x_\tau)$, build one strong classifier from linear combination with trainable 'classifier weights', **w**
- **Key:** How to formulate objective function so that best w's are found. QUBO

$$E(\boldsymbol{w}) = \sum_{i,j=1}^{n=36} C_{ij} w_i w_j + \sum_{i=1}^{n=36} 2(\lambda - C_i) w_i$$

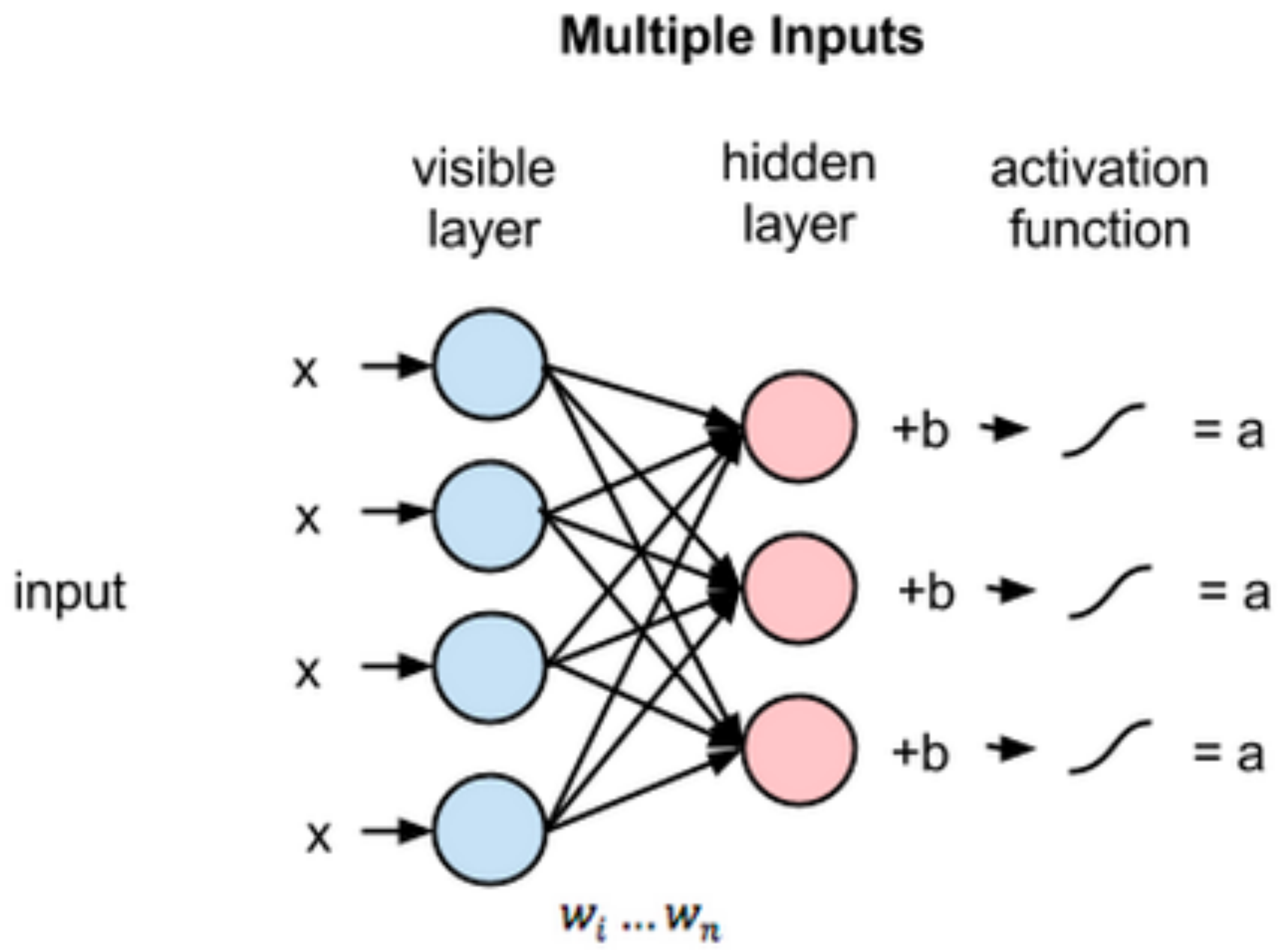
$$C_{ij} = \sum_\tau c_i(x_\tau) c_j(x_\tau) \quad C_i = \sum_\tau c_i(x_\tau) y_\tau$$

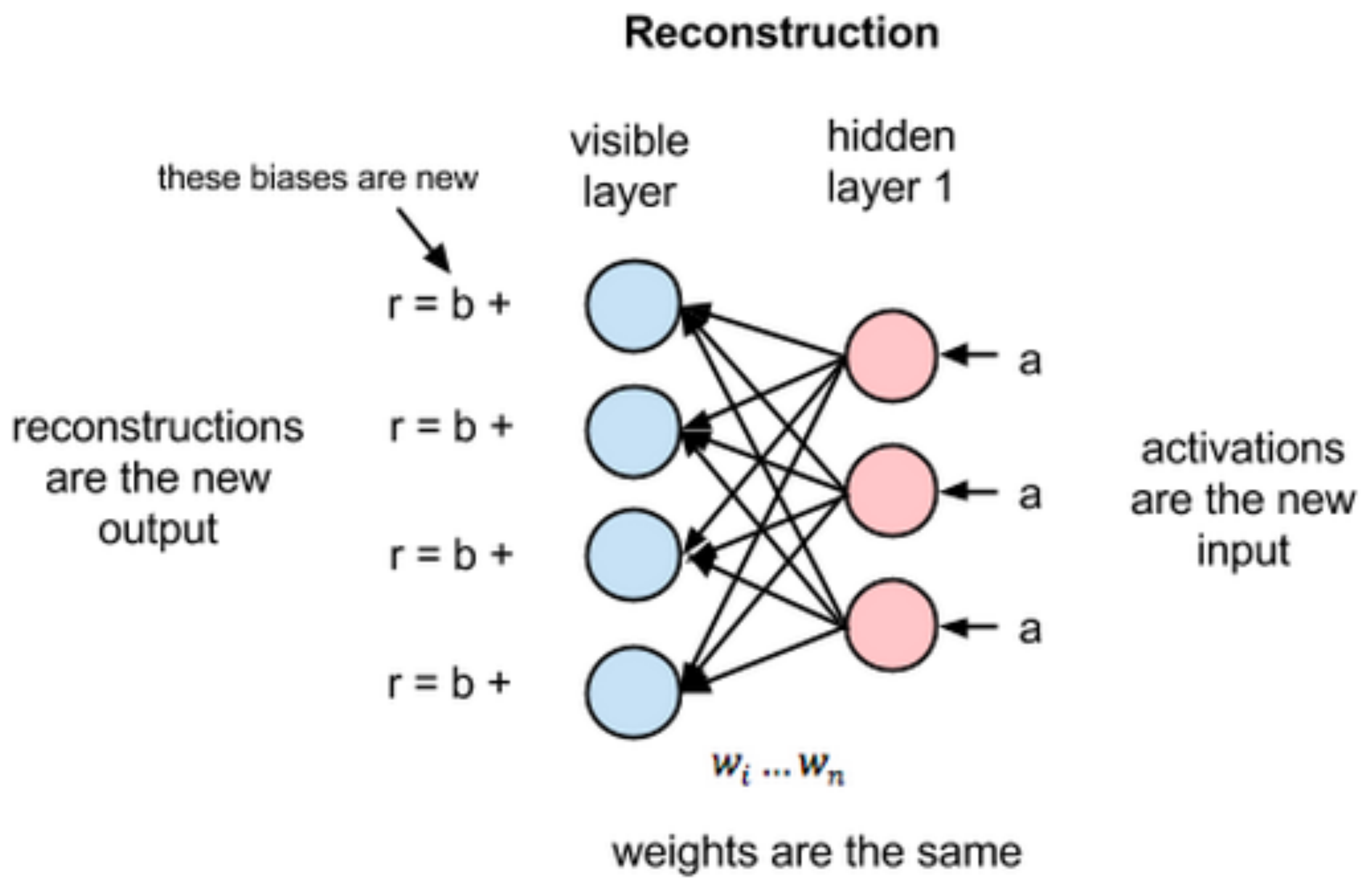
Applications - Quantum Annealing: Classification in cosmology with qRBM

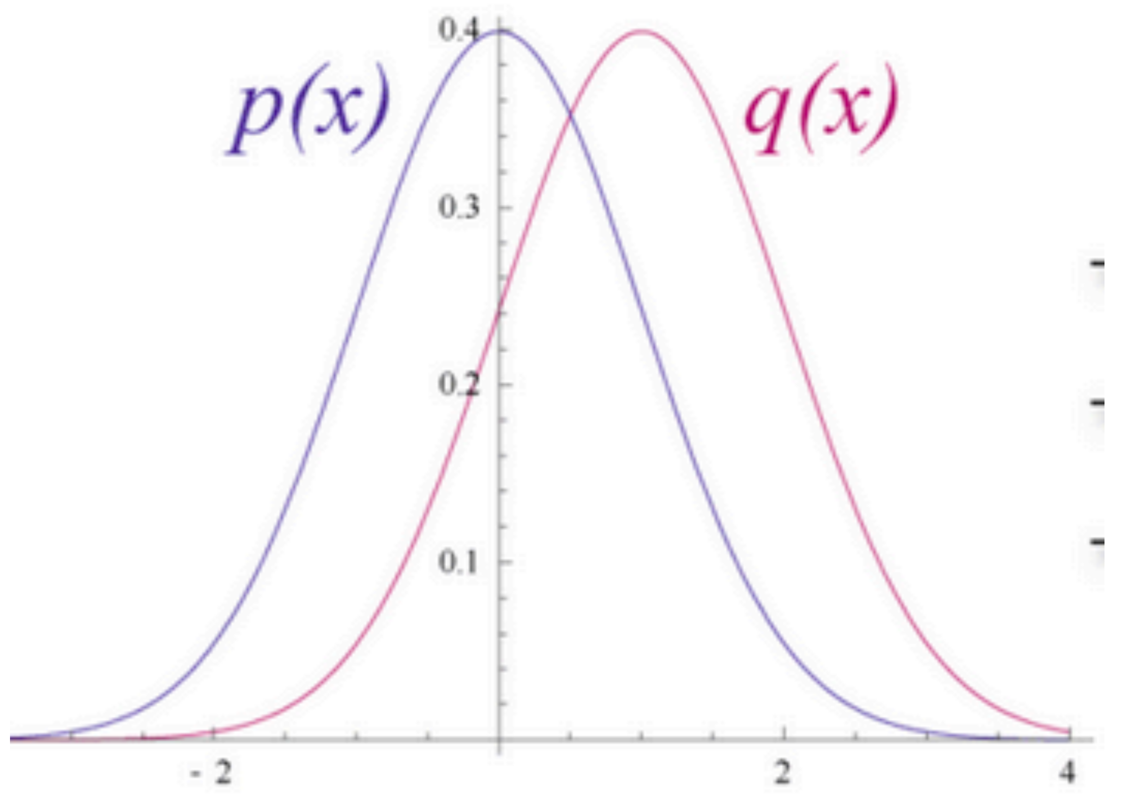
Quantum annealers as sampling engines? Demonstrated in galaxy classification problem (arXiv:1911.06259)

Recall: Restricted Bolzmann Machines

- Two-layer (visible and hidden) stochastic generative models
- Gradient ascent in log(prob) that Boltzmann machine would generate observed data when sampling from equilibrium





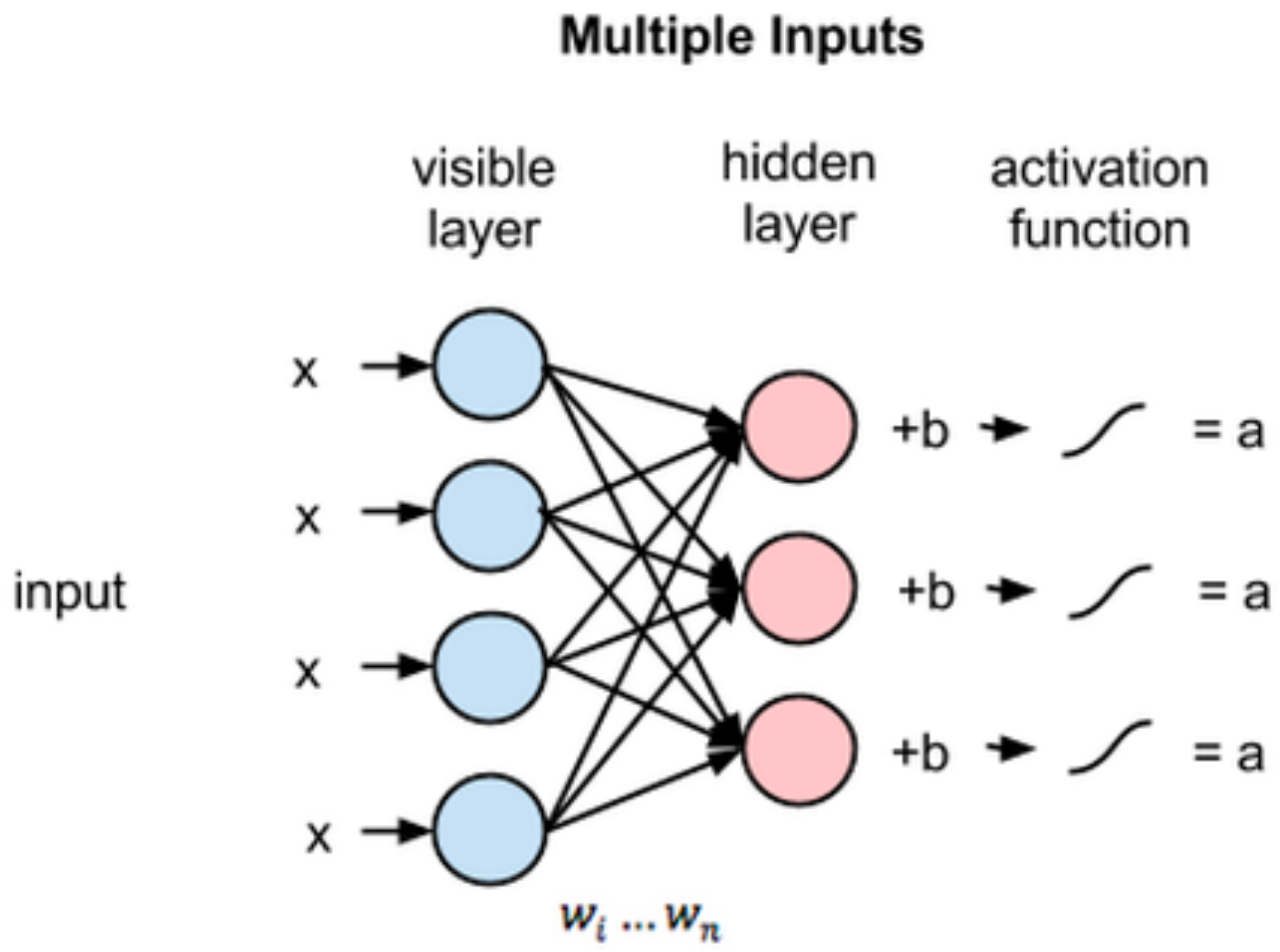


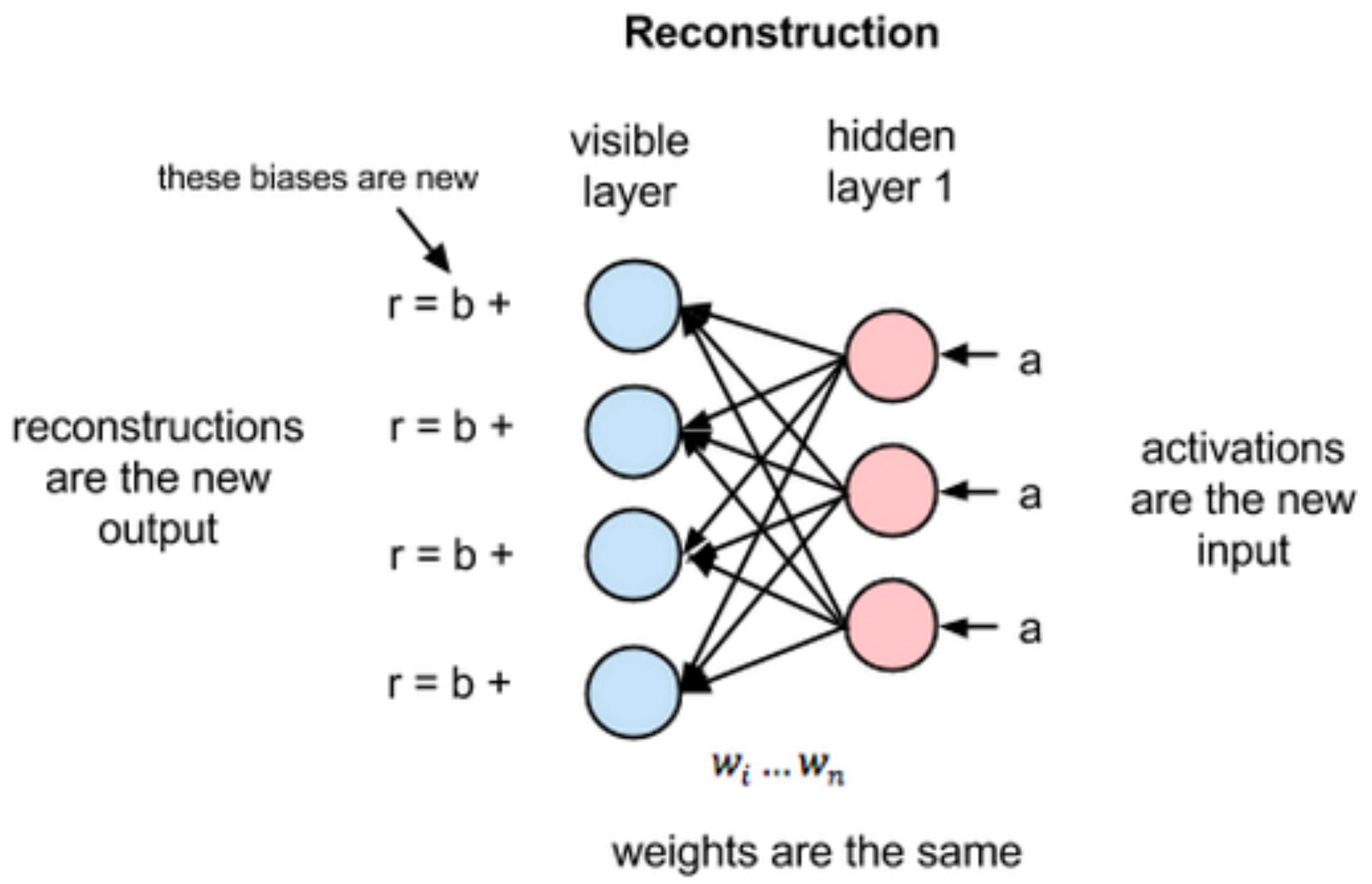
Applications - Quantum Annealing: Classification in cosmology with qRBM

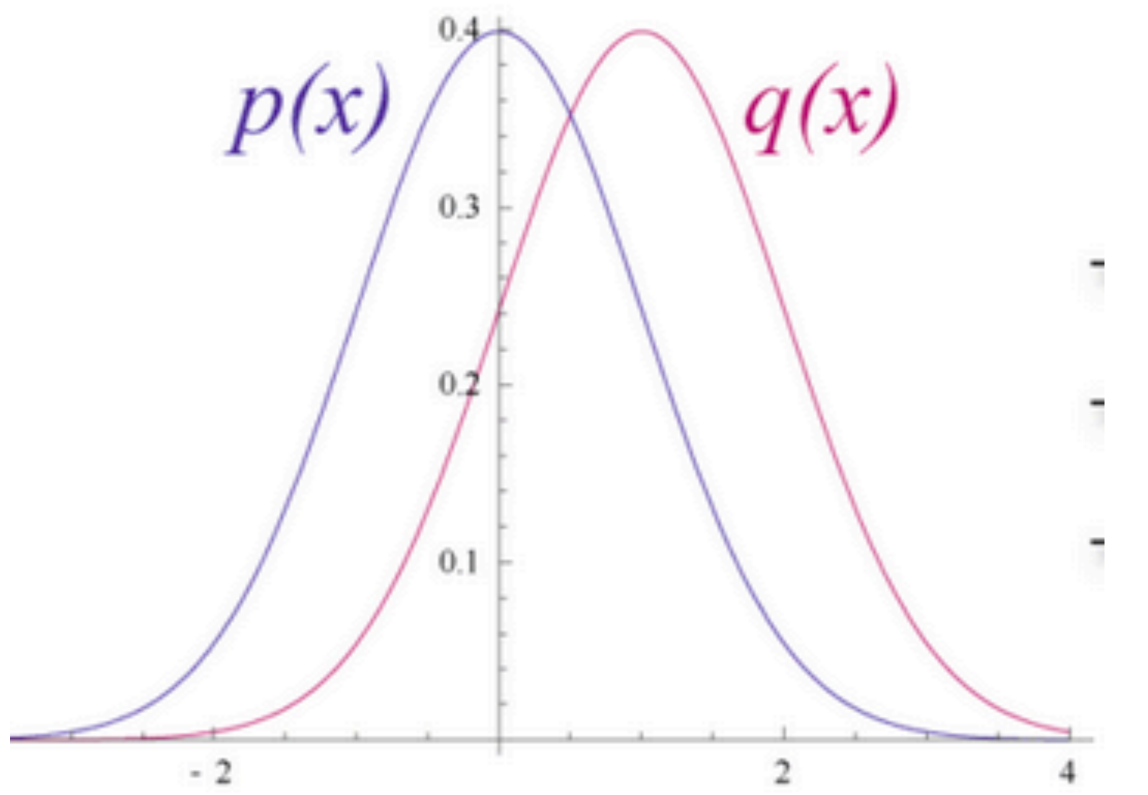
Quantum annealers as sampling engines? Demonstrated in galaxy classification problem (arXiv:1911.06259)

Recall: Restricted Bolzmann Machines

- Two-layer (visible and hidden) stochastic generative models
- Gradient ascent in log(prob) that Boltzmann machine would generate observed data when sampling from equilibrium
- Boltmann machines **are** Ising models
- Tunable couplings between qubits → graph connection weights
  Sampling from graph configuration → natural part of annealing





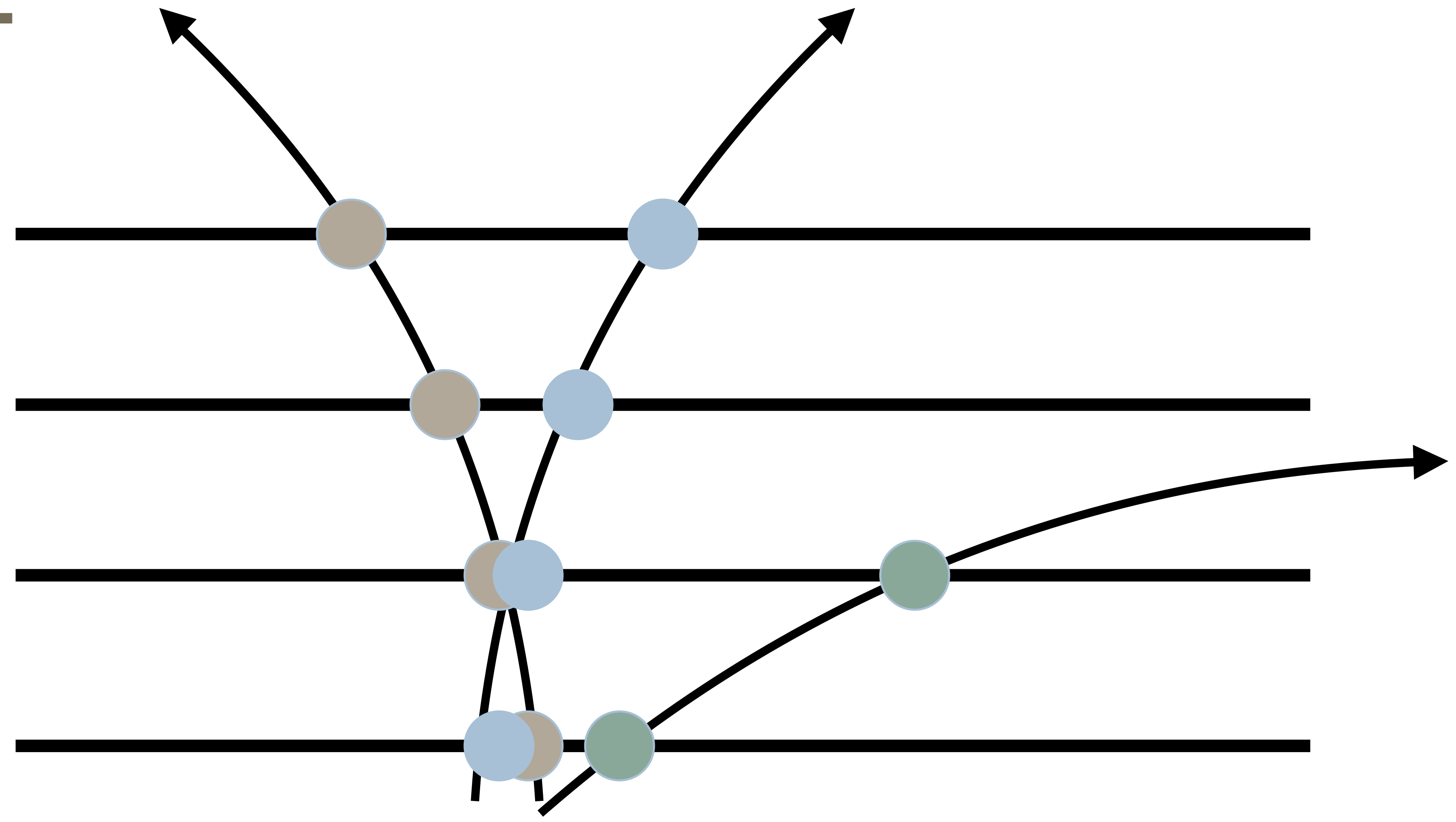


Applications - Quantum Circuit: Quantum GNNs for particle track repo

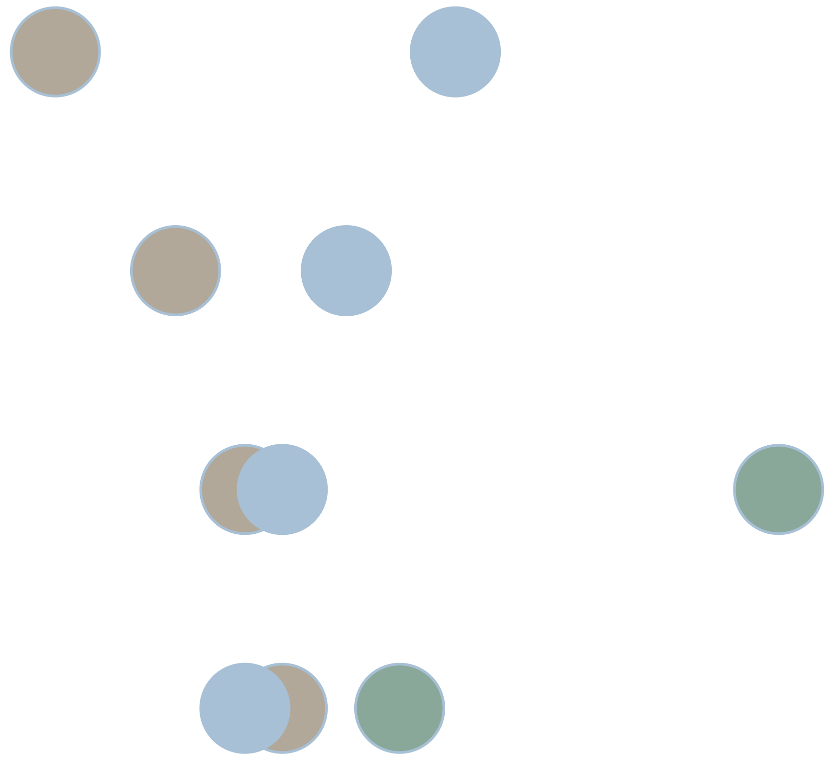
Speed-up tracking with Graph Neural Networks

- Input network: encodes hit information as node features
- Edge network: outputs edge features
- Node network: calculates hidden node features

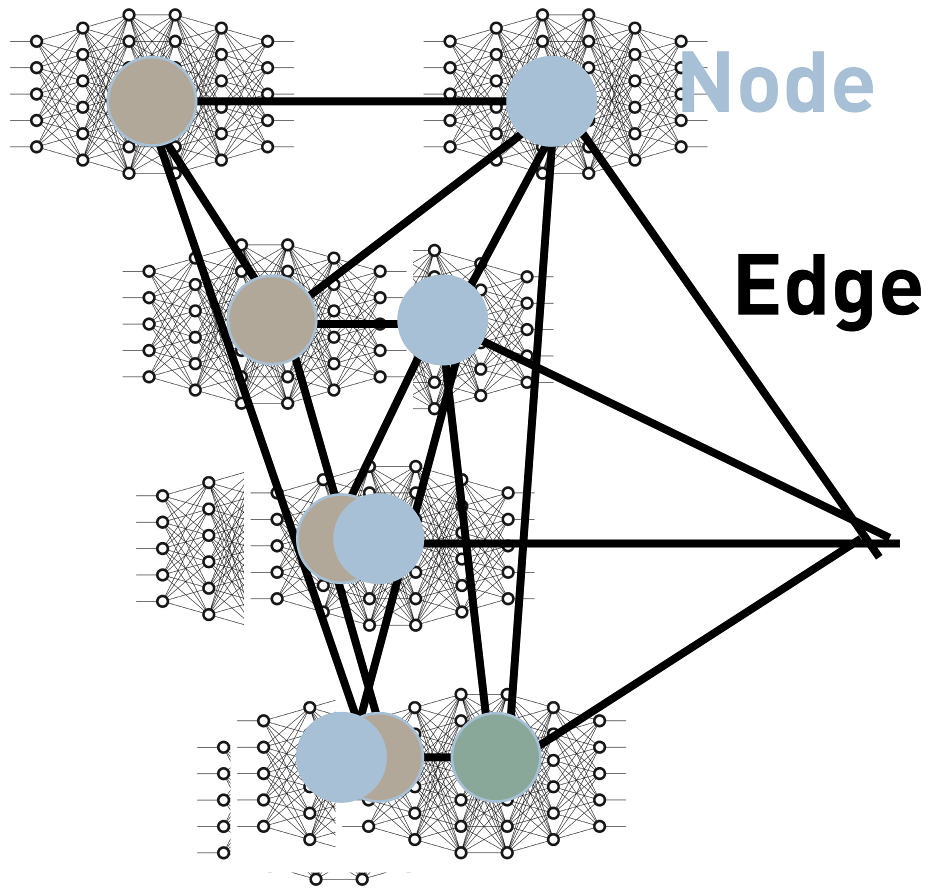
In principle



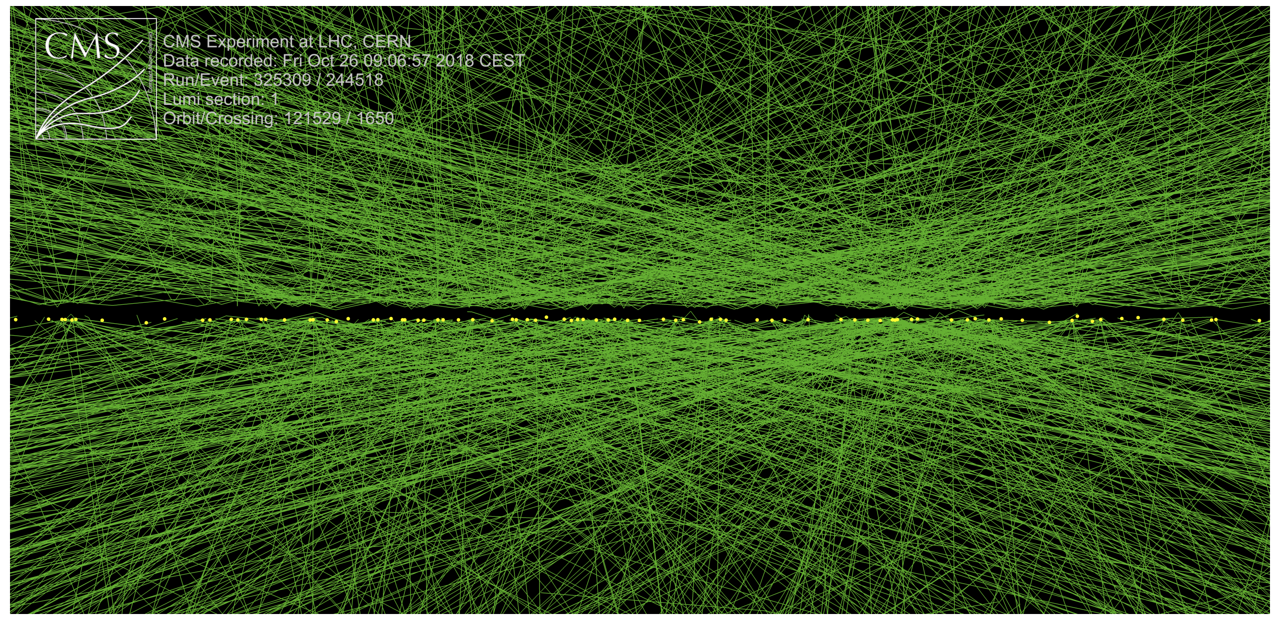
Hits in detector



A graph

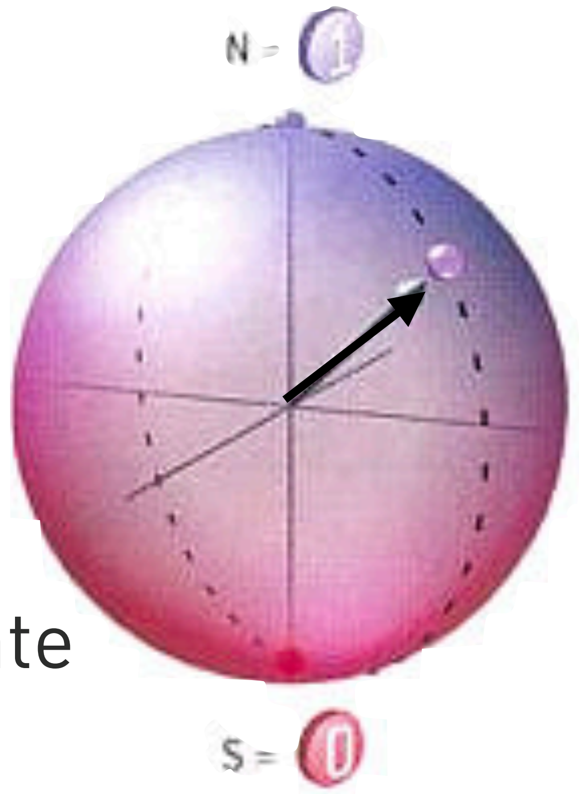


Reality



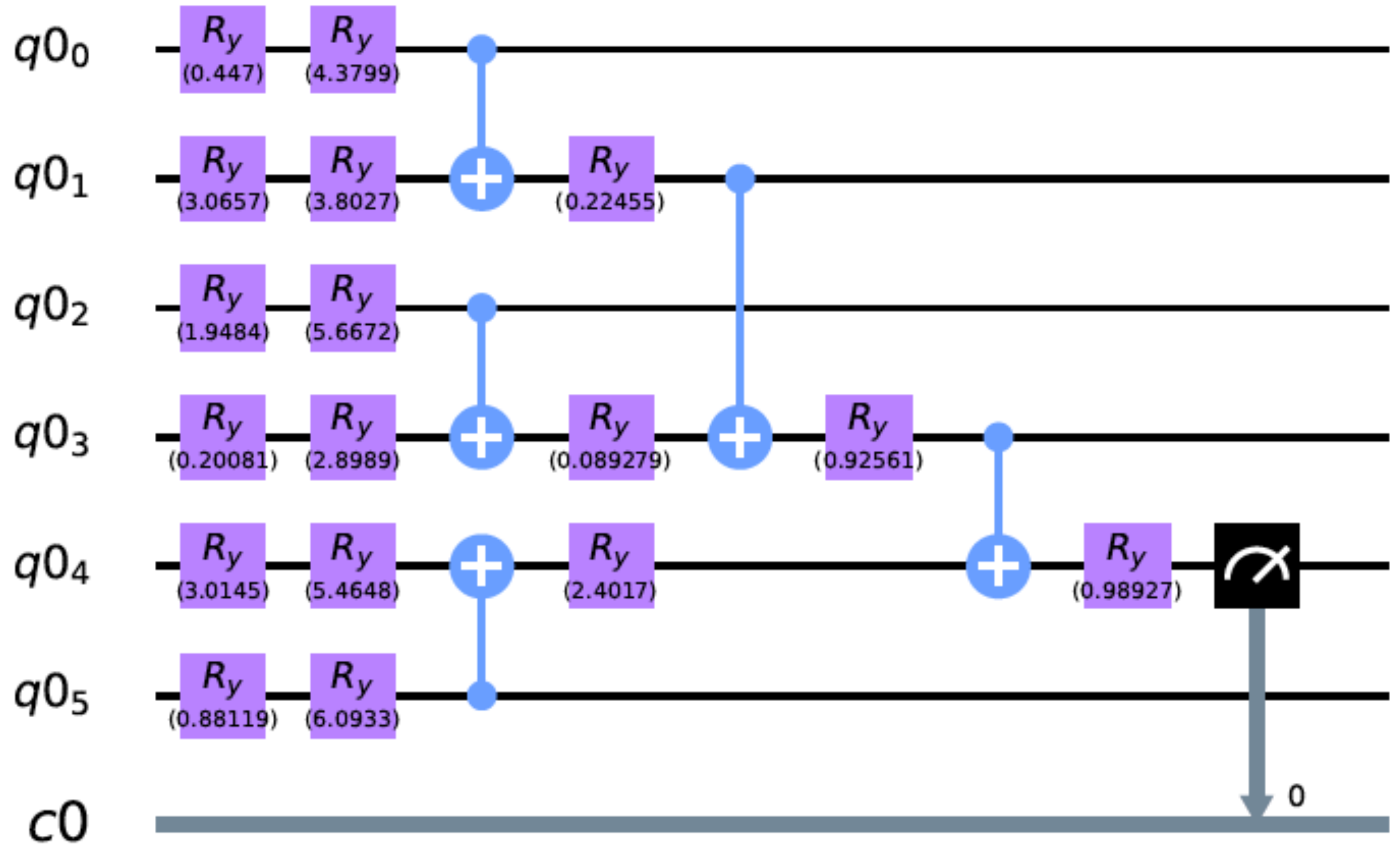
Applications - Quantum Circuit: Quantum GNNs for particle track repo

Replace edge network by quantum Edge network through CNOT gates and rotation (R_Y)



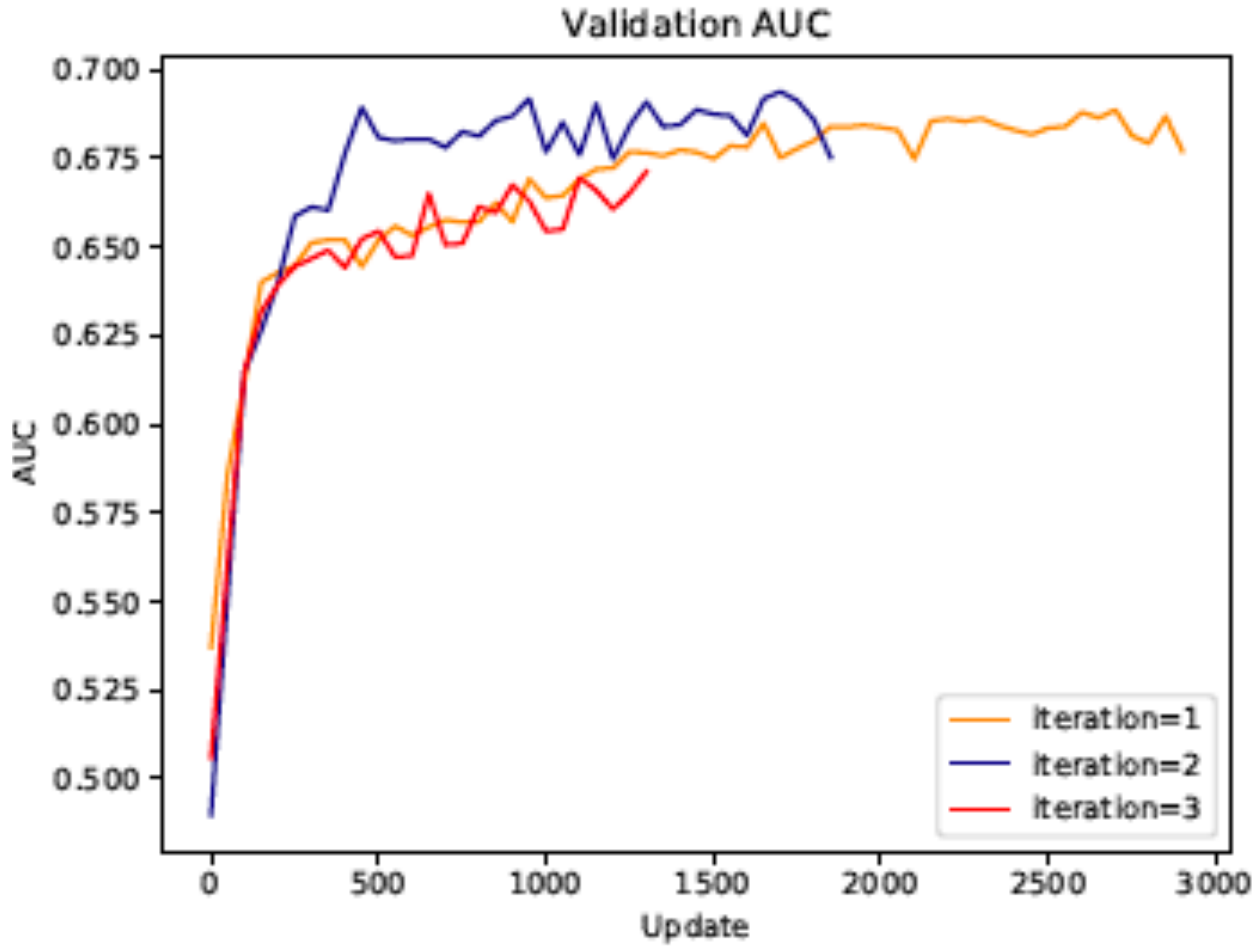
Reimplement classical GNN as qGNN

- Networks are tree tensor networks
- Datapoints: encoded as parameters of R_y gates
- Trainable parameters: Angles of rotations on Bloch sphere. Starts with randomly initialised parameters, which R_y gates rotate according to parameter's value
- CNOT gate is used to introduce correlation between qubits so that values are not independent.
- At the end of the circuit, there is a measurement of a single quit



Proof-of-priniciple. Achieve slightly lower accuracy than classical, power would be in potential speedup!

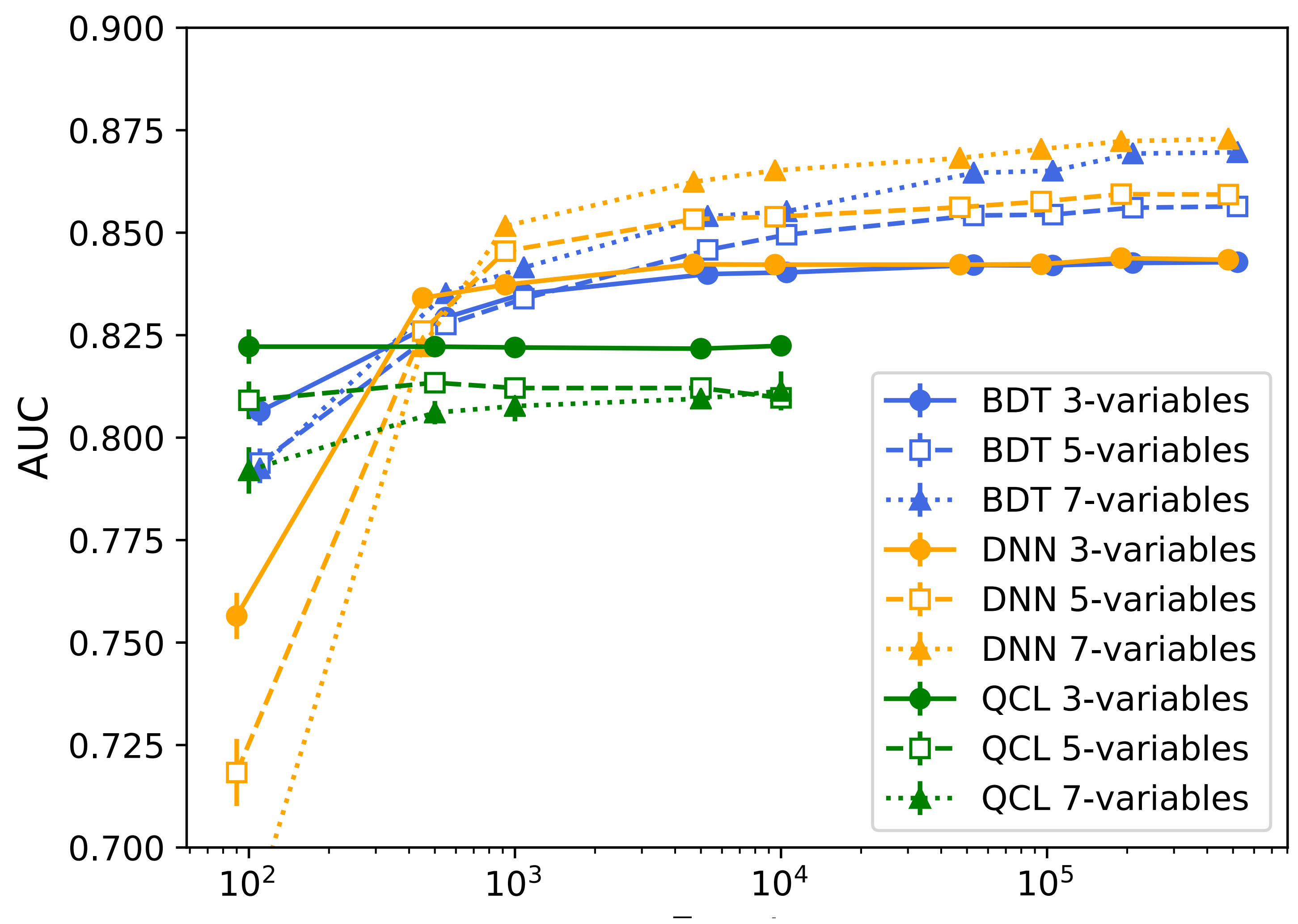
Applications - Quantum Circuit: Quantum GNNs for particle track repo

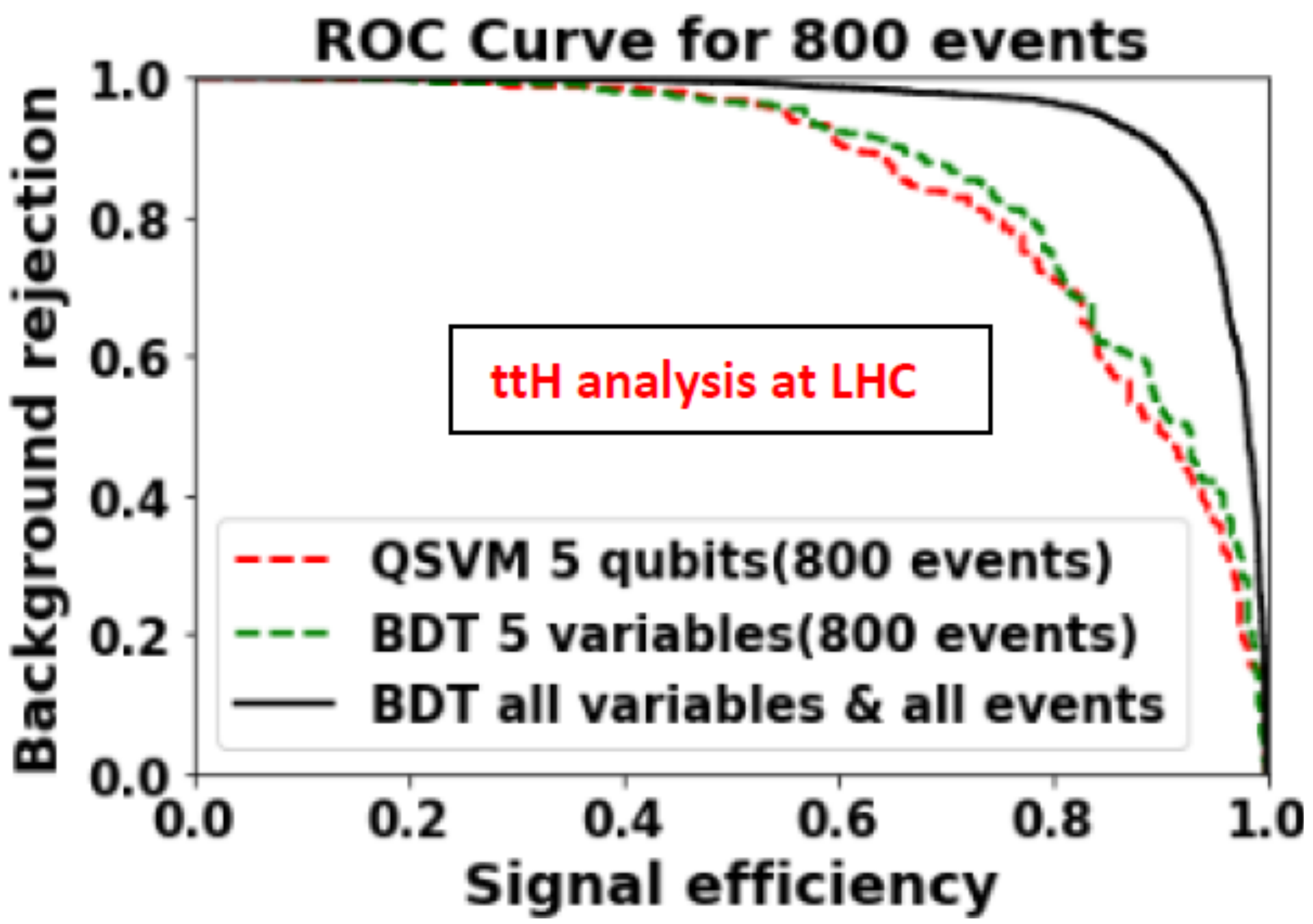


Applications - Quantum Circuit: Classification with Variational Quantum Circuits

Variational Quantum Algorithms for Machine Learning

- Map input data to exponentially large quantum state space to enhance ability to find optimal solution, then the usual $p_y(\vec{x}) \leftarrow \langle \Phi(\vec{x})|W^\dagger(\vec{\theta})M_y W(\vec{\theta})|\Phi(\vec{x})\rangle$





FINAL REMARKS

Final remark on Quantum Data

QML on datasets of quantum states?

- Rather than processing classical signal in photonic sensors, direct them into QC and apply variational circuit trained to extract state information

- Simulate HEP experiment on Quantum Computer followed by QML algorithm to analyse resulting quantum states

Summary

Can achieve good accuracy on certain problems, but cannot evaluate whether quantum speedups will be achieved

- My take-home: We know QML algorithms ‘work’ for HEP problems, but the desired gain in using them (to solve HEP ‘resource-problem’) is not yet clear and will probably not be clear until we have large general QC’s