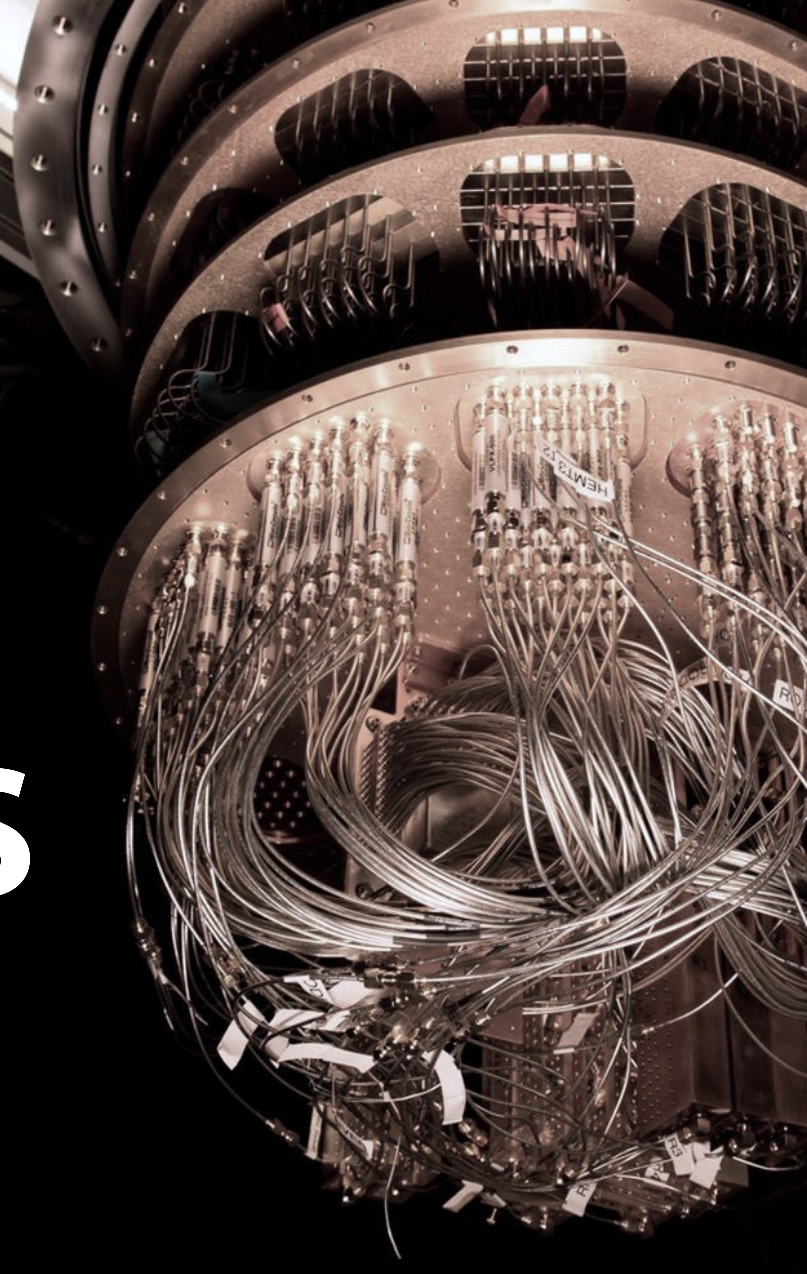
Quantum Computing Applications for ATLAS



Koji Terashi
ICEPP, University of Tokyo

3 March, 2020
ATLAS Weekly

Is it relevant for us??

What can it do?

In distant future, right?

Outline of the Talk

Quantum Computing in a Nutshell

- Fundamentals for quantum computing
- Gate-based quantum computing
- Quantum annealing

Quantum Machine Learning for HEP

- Classification with quantum annealing
- Classification with gate-based quantum algorithm

Other HEP Applications

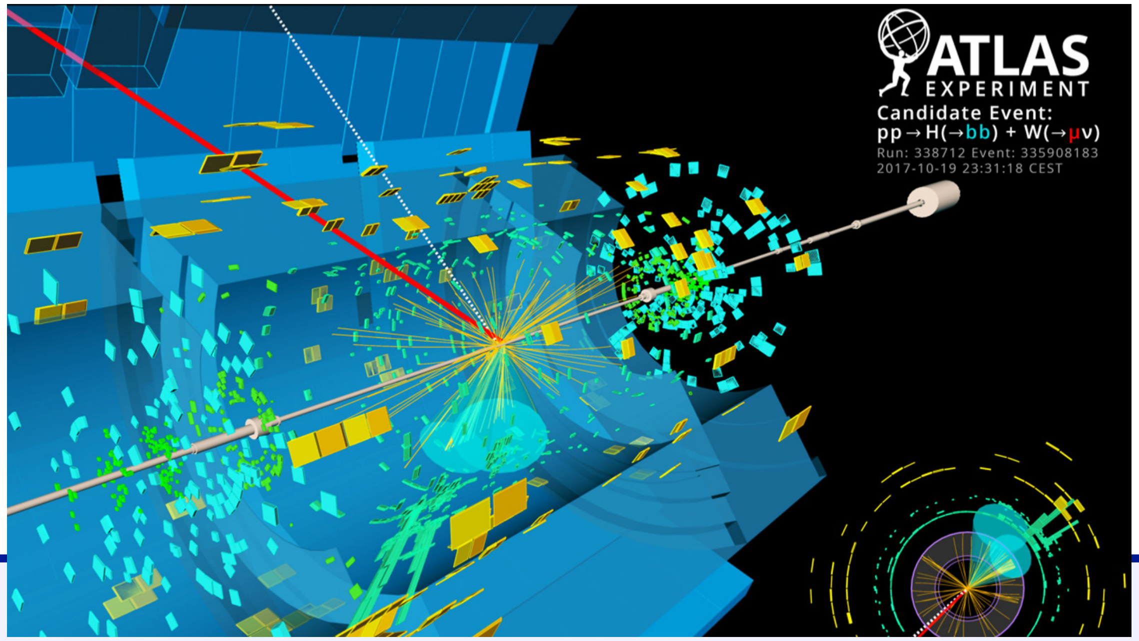
Quantum and High-Energy Physics

Fundamental constituents in nature governed by quantum mechanics

Particle physics directly accessing quantum properties in nature

Typically large computing resources required in HEP experiment

➡ HEP is an interesting place to explore the power of quantum computing

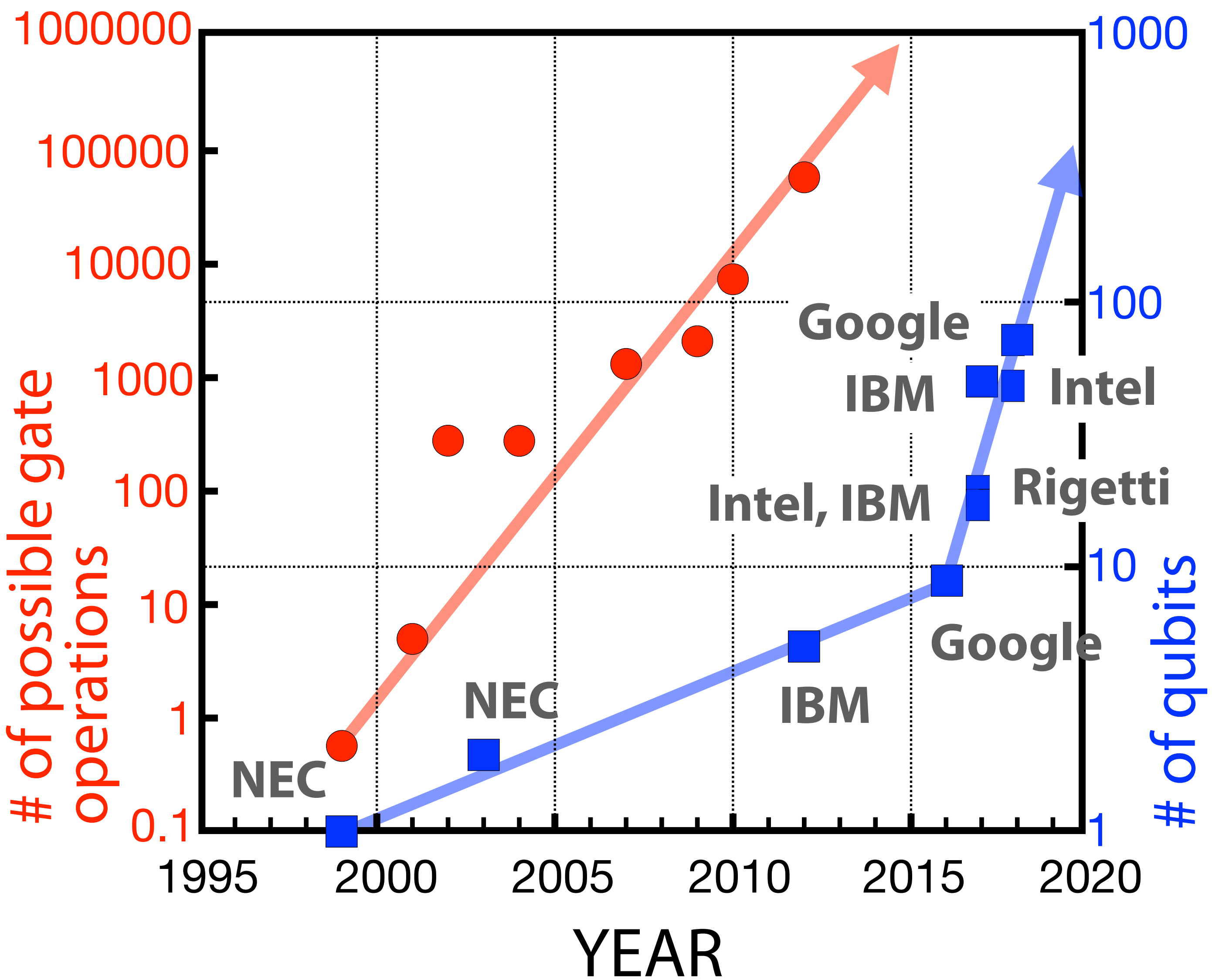


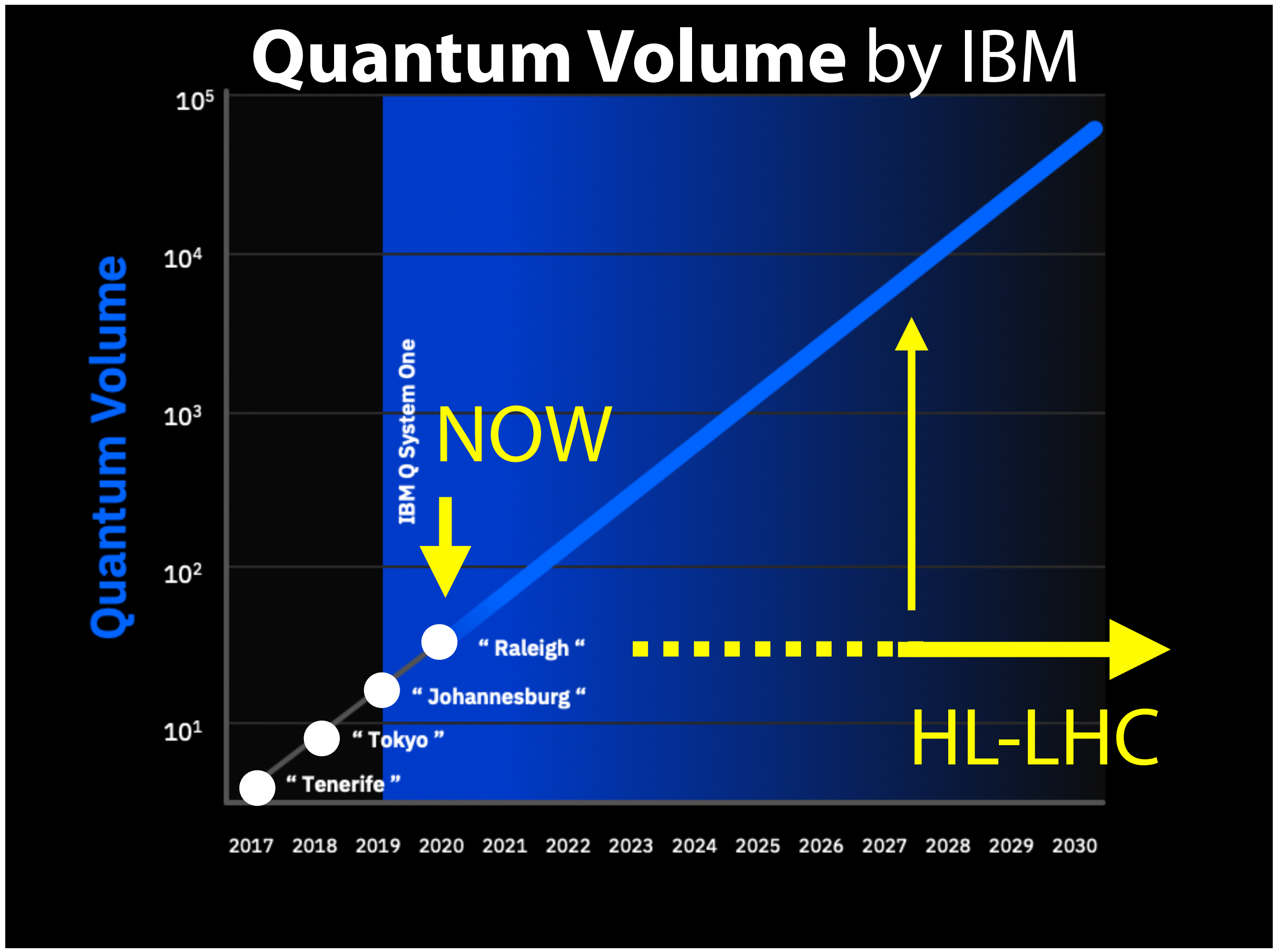
Quantum computing :

- Potential exponential speed-up in certain computational tasks
- ***Huge*** representational power due to complex quantum space

10 qubits	➡	Represent ~10^3 states
50	➡	~10^{15} states
300	➡	~10^{90} states ≈ # of atoms in the Universe!!

Growth of Quantum Computing

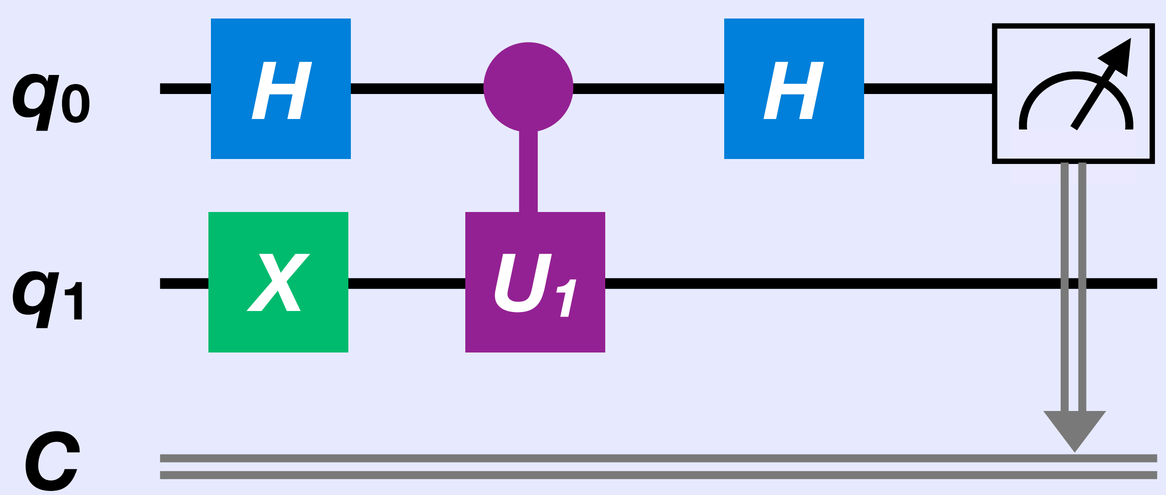




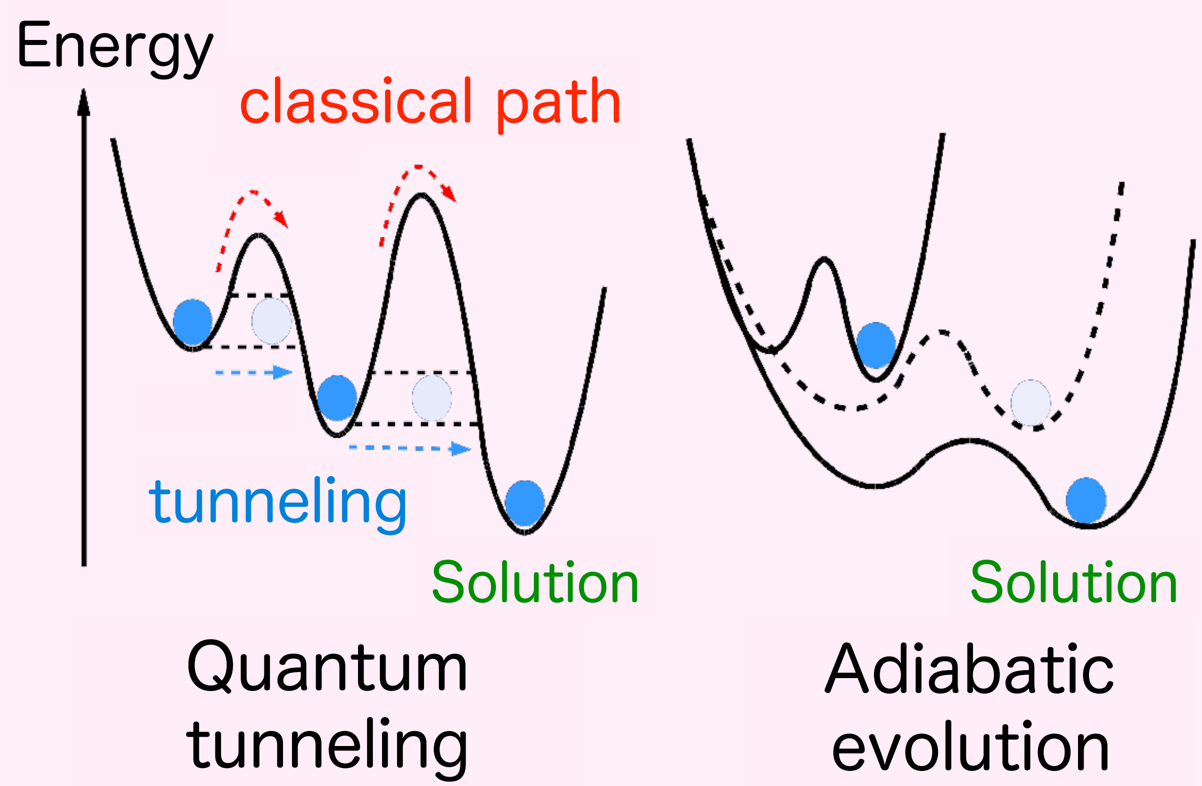
- Rapid progress over last years, largely driven by superconducting qubit system
- "Quantum Volume" (IBM's performance metric) doubles every year so far

Quantum Computing

Gate-based Quantum Computing



Quantum Annealing



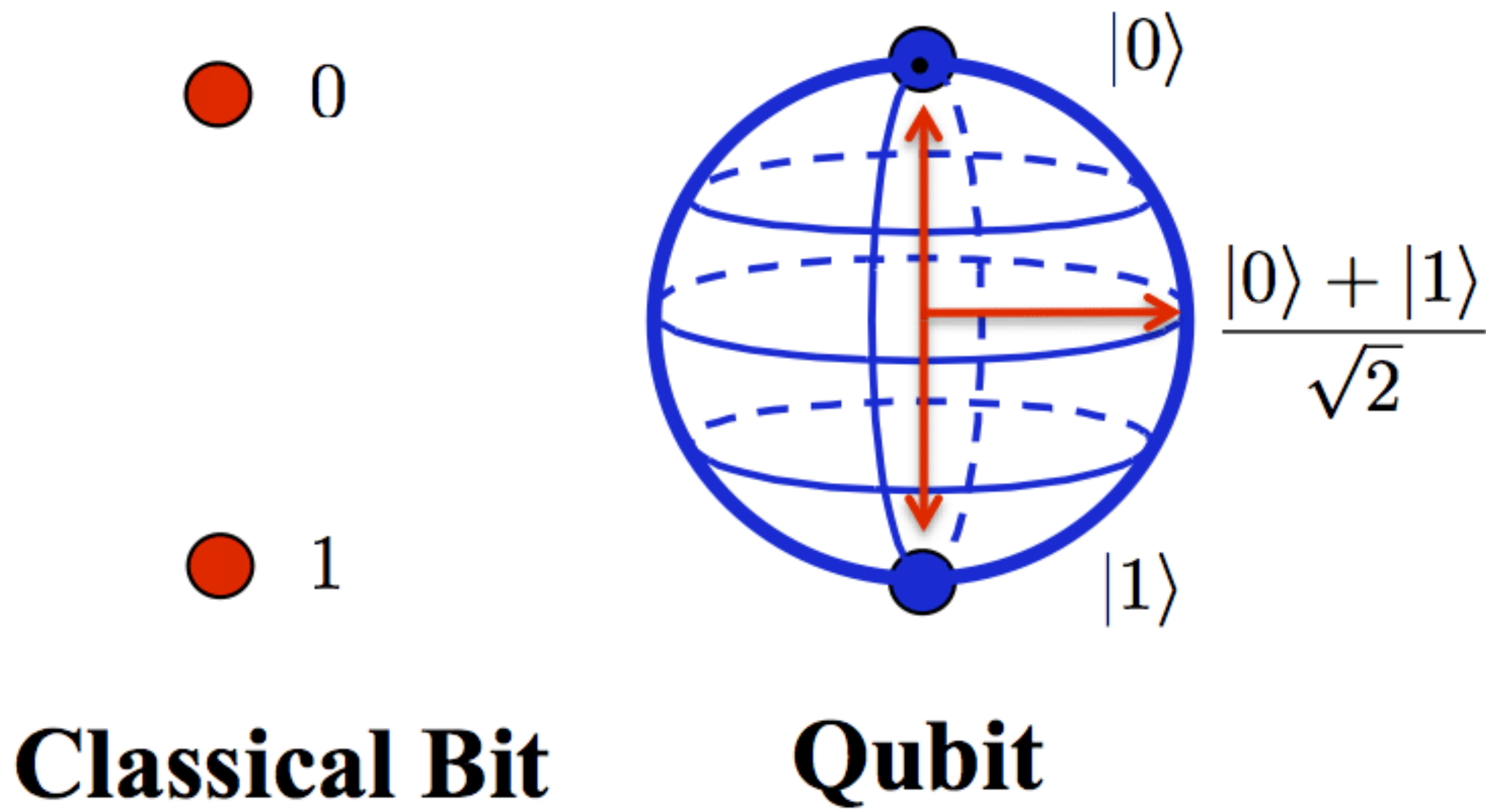
- Develop circuit composed of quantum gates (= algorithm) for each problem
- Applicable to wide variety of problems (➡ Universal)

- Extract solution (= lowest energy state) by slowly changing hamiltonian
- Suitable for optimization problem

Quantum Computing Fundamentals

Qubit = *a basic unit of quantum information* :

- 2-state quantum-mechanical system
- Represent arbitrary state of the superposition of $|0\rangle$ and $|1\rangle$
- State controlled by applying unitary operator (= gate)



Strengths and Weaknesses of Quantum Computing
Zahid Hussain

Quantum Computing Fundamentals

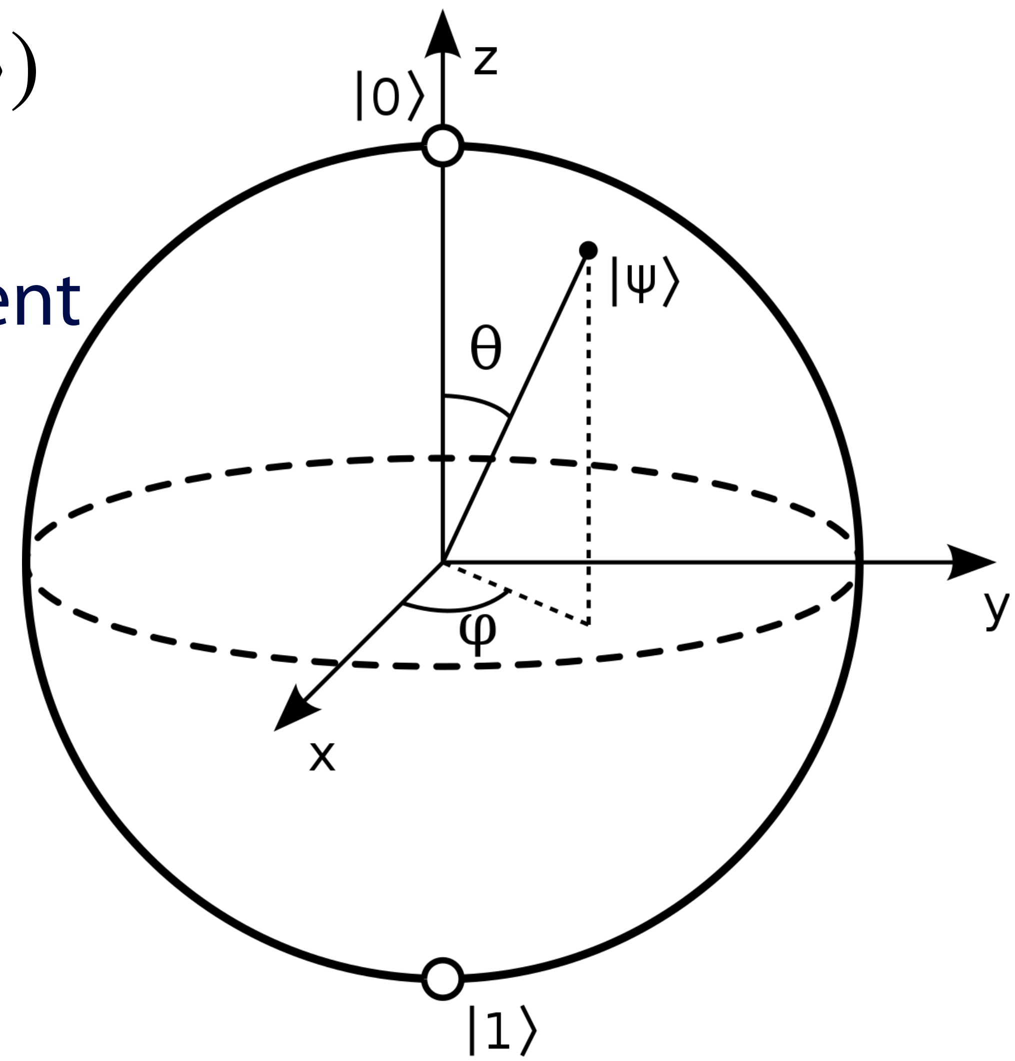
Qubit = *a basic unit of quantum information* :

- 2-state quantum-mechanical system
- Represent arbitrary state of the superposition of $|0\rangle$ and $|1\rangle$
- State controlled by applying unitary operator (= gate)

Bloch sphere representation of a qubit state :

$$|\psi\rangle = e^{i\gamma}(\cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle)$$

- Superposition state given by θ and ϕ angles
- Global phase γ not contribute to measurement



Quantum Computing Fundamentals

Qubit = *a basic unit of quantum information* :

- 2-state quantum-mechanical system
- Represent arbitrary state of the superposition of $|0\rangle$ and $|1\rangle$
- State controlled by applying unitary operator (= gate)

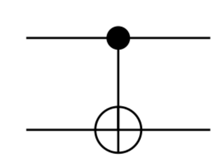
Bloch sphere representation of a qubit state :

$$|\psi\rangle = e^{i\gamma}(\cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle)$$

- Superposition state given by θ and ϕ angles
- Global phase γ not contribute to measurement

Quantum gates to control quantum states :

- Single-qubit gates, e.g, Pauli-X/Y/Z
- 2-qubit gates, e.g, CNOT
- Represented as $2^n \times 2^n$ unitary matrix for n-qubit gate

Operator	Gate(s)	Matrix
Pauli-X (X)	$-\boxed{\mathbf{X}}-$	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Hadamard (H)	$-\boxed{\mathbf{H}}-$	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

Wikipedia

Quantum Computing Fundamentals

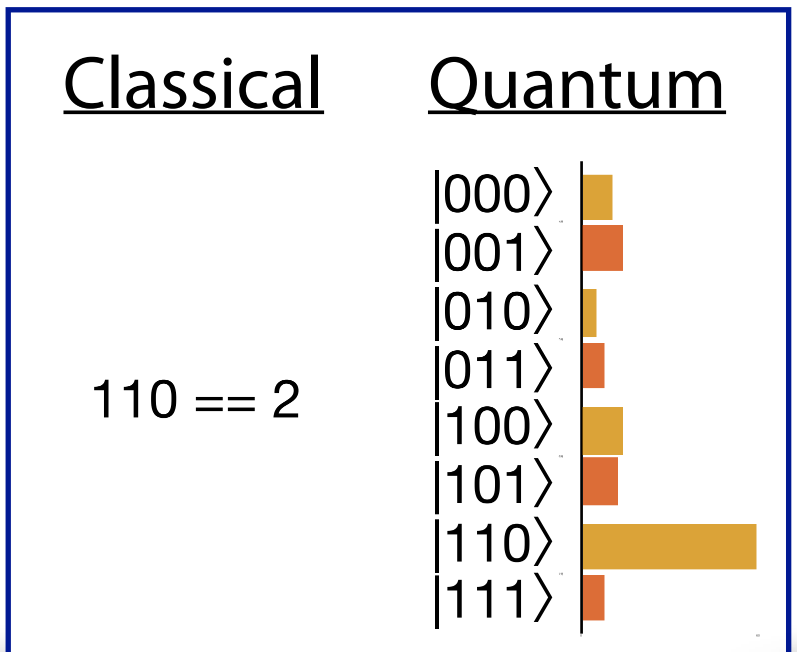
Quantum algorithm (= circuit composed of gates) to solve problem

Quite different from classical computing algorithm:

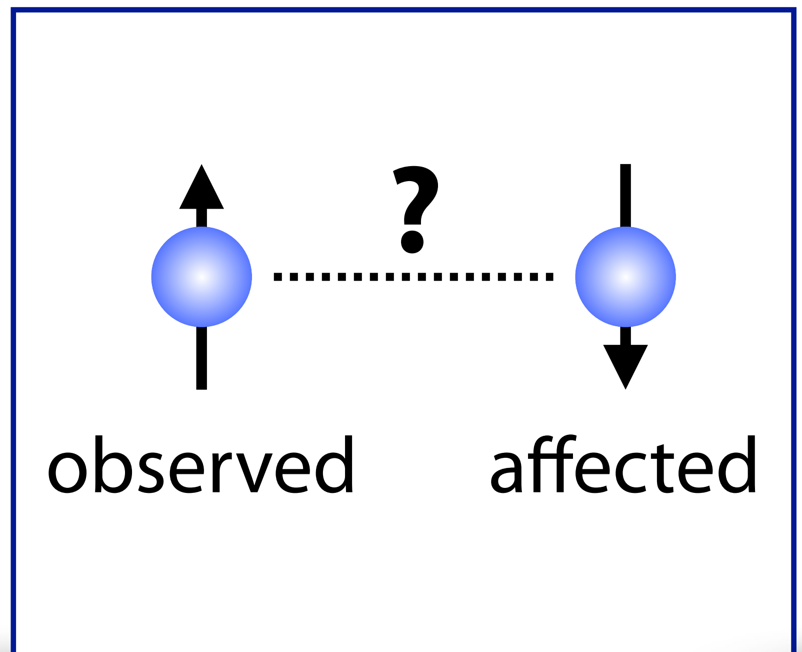
- Measurement of *N* qubits gives just information of *N* classical bits…

Key to achieve quantum advantage :

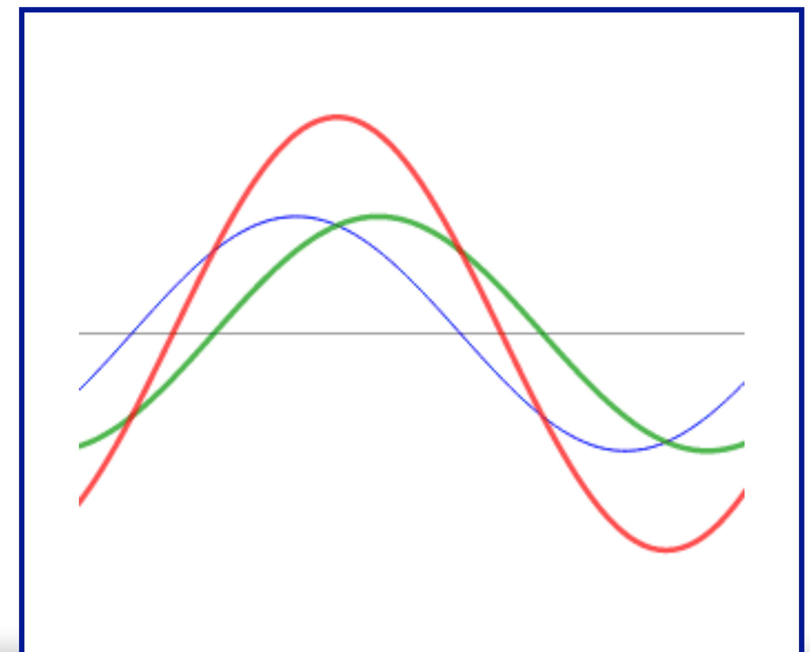
Superposition



Entanglement



Interference



Quantum Computing Fundamentals

Quantum algorithm (= circuit composed of gates) to solve problem

Quite diff

- Measure ... its…

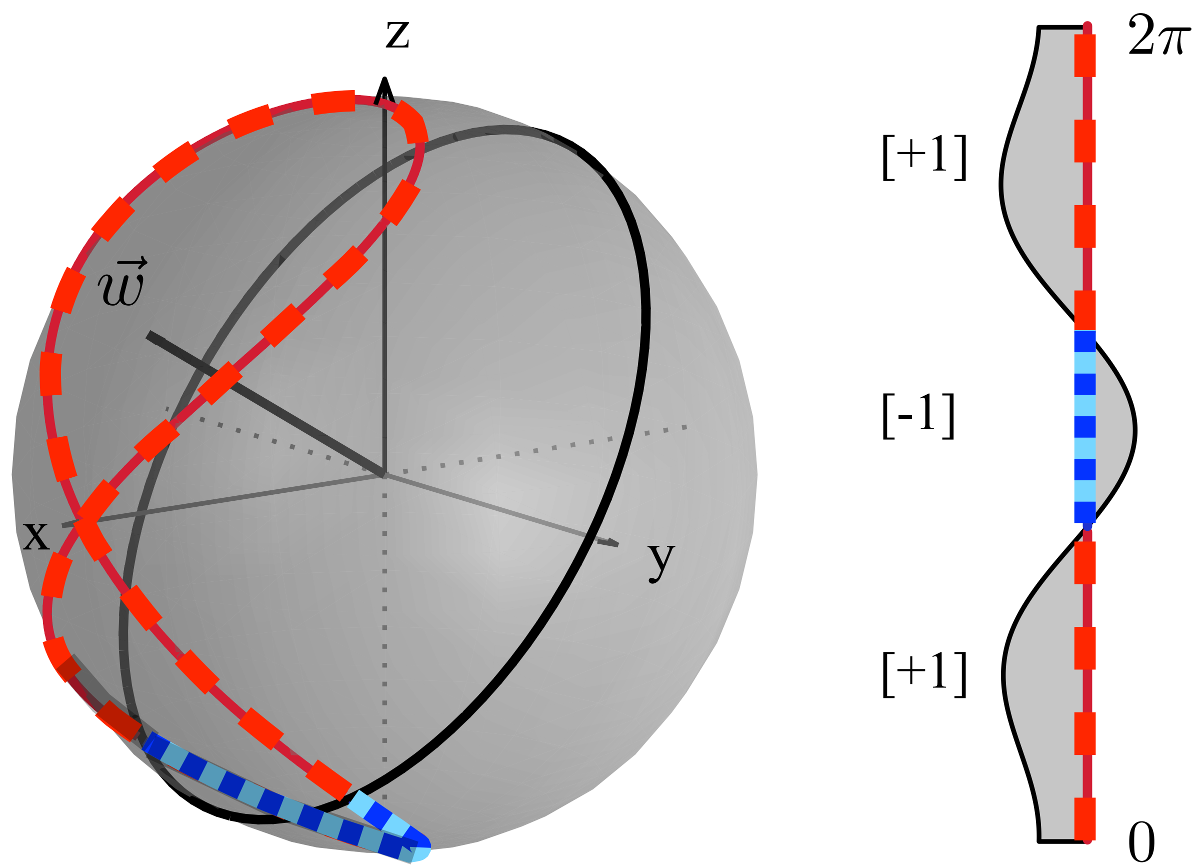
Key to ac

Superpo ... e

Classical

110 == 2

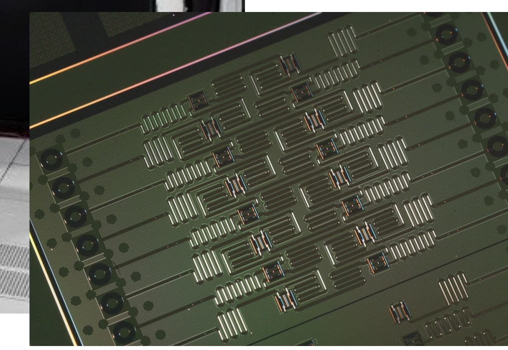
V. Havlicek *et al.*, arXiv:[1804.11326](https://arxiv.org/abs/1804.11326)



Quantum Classical

➡ Exploit these properties in quantum space to enhance the *probability of getting a correct answer* among 2^N possibilities

Quantum Computers



IBM Q

Superconducting qubit machines

Publicly available from

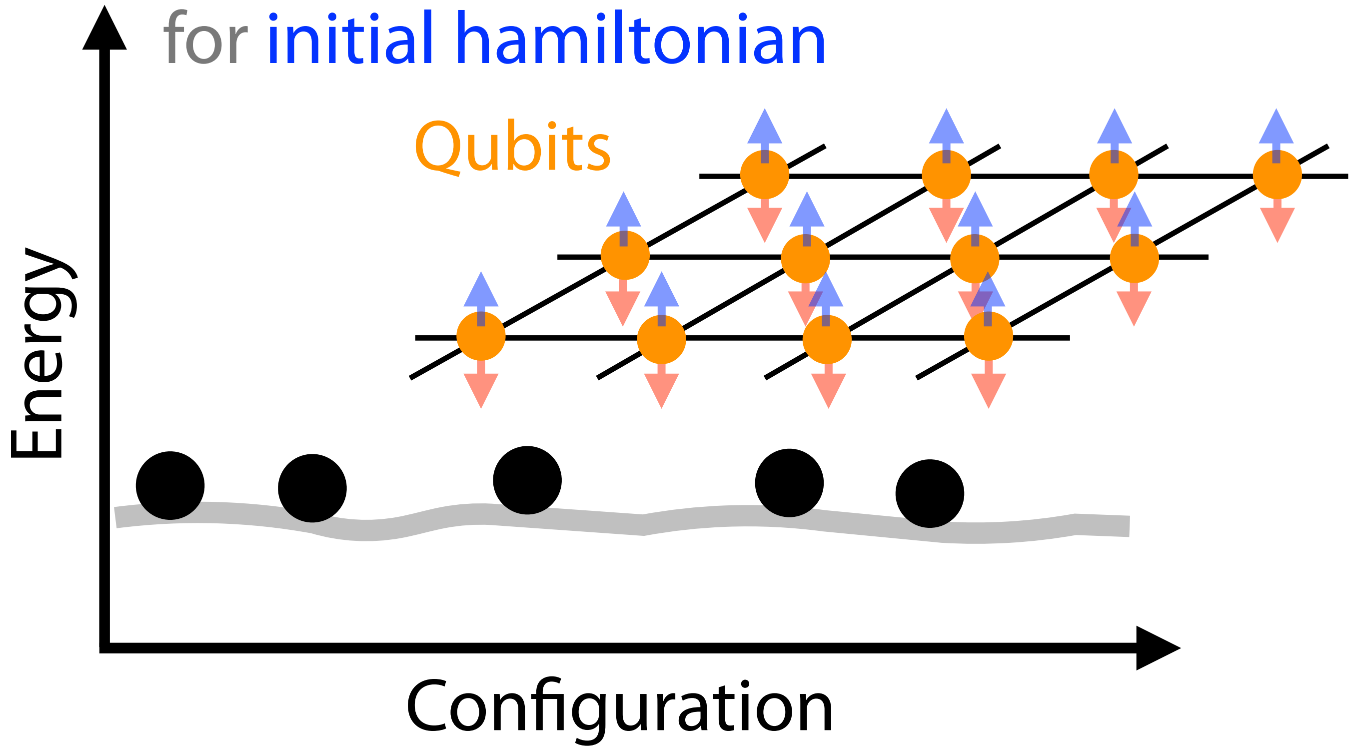
- **IBM, Rigetti** (gate-based)
  - 5/15/20/28/53 qubits from IBM
  - Coherence time ~ $O(10)$ μs
- **D-Wave** (quantum annealing)
  - 2,048 qubits (not full connection)
  - Annealing time = 1~2,000 μs
    Coherence time ~ $O(\mathrm{ns})$

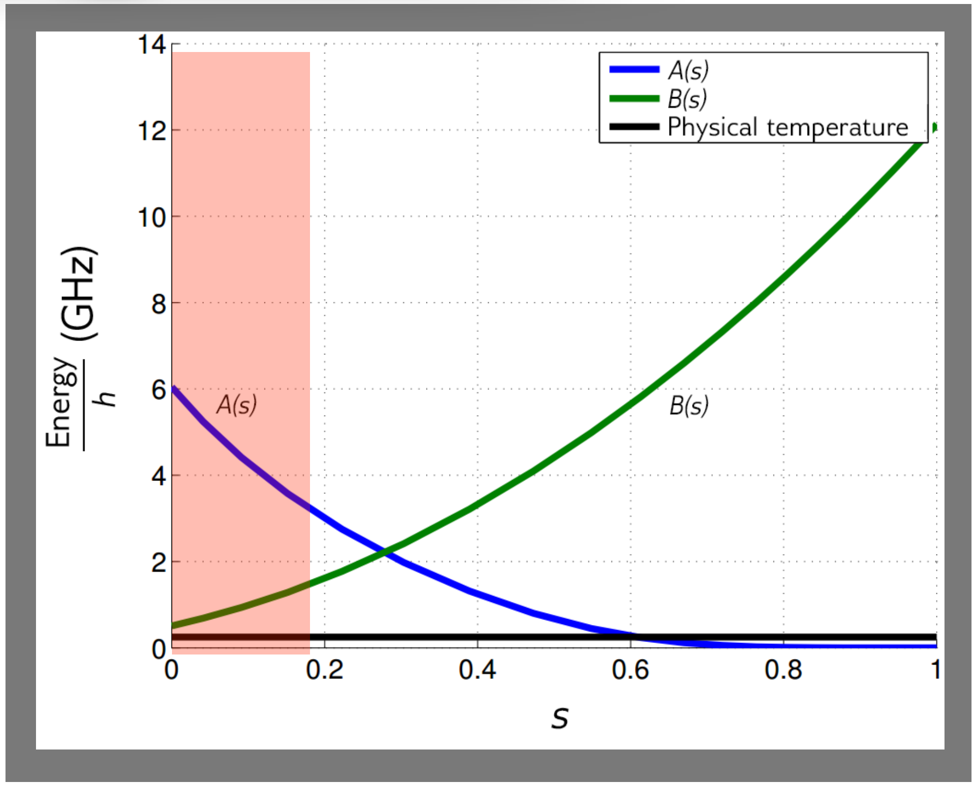
D-Wave

Typically trade-off between the number of qubits and coherence time

How Quantum Annealing Works

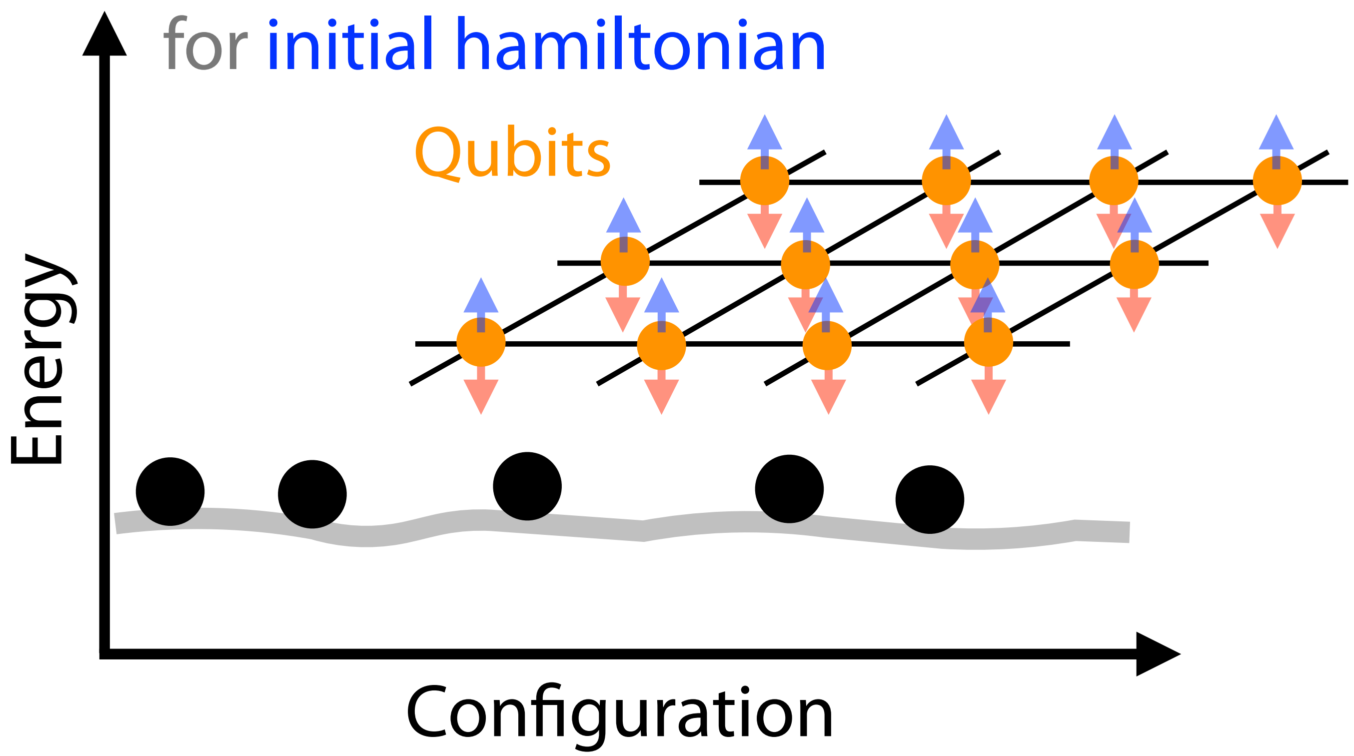
All configurations at ground state for initial hamiltonian

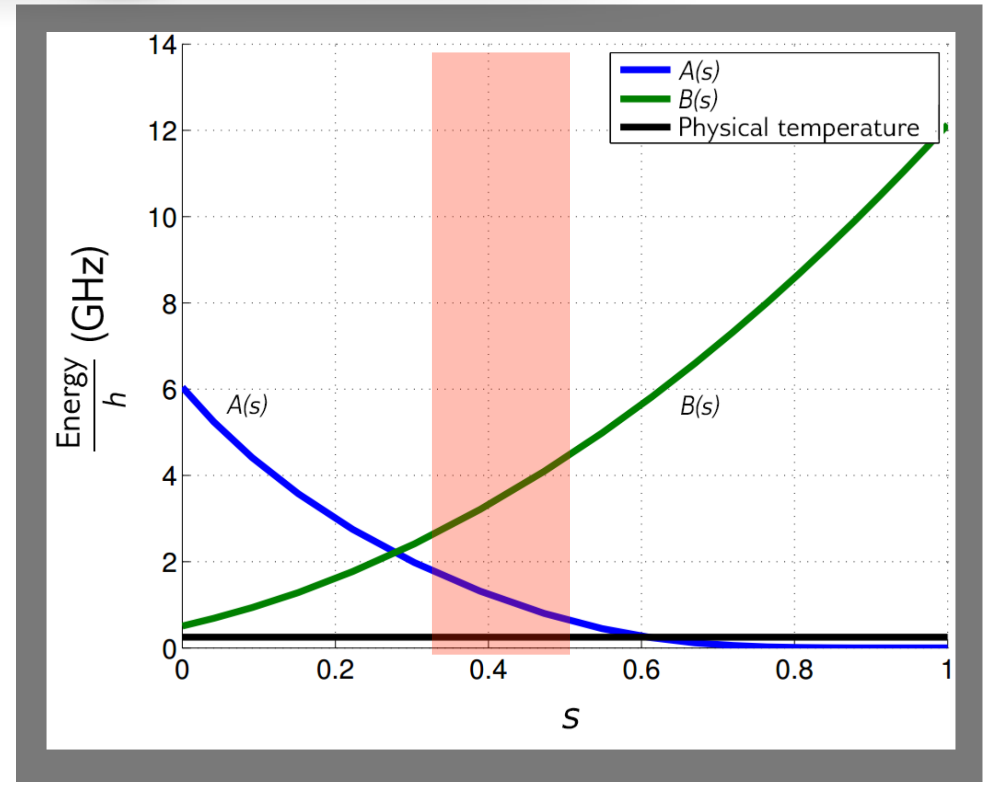




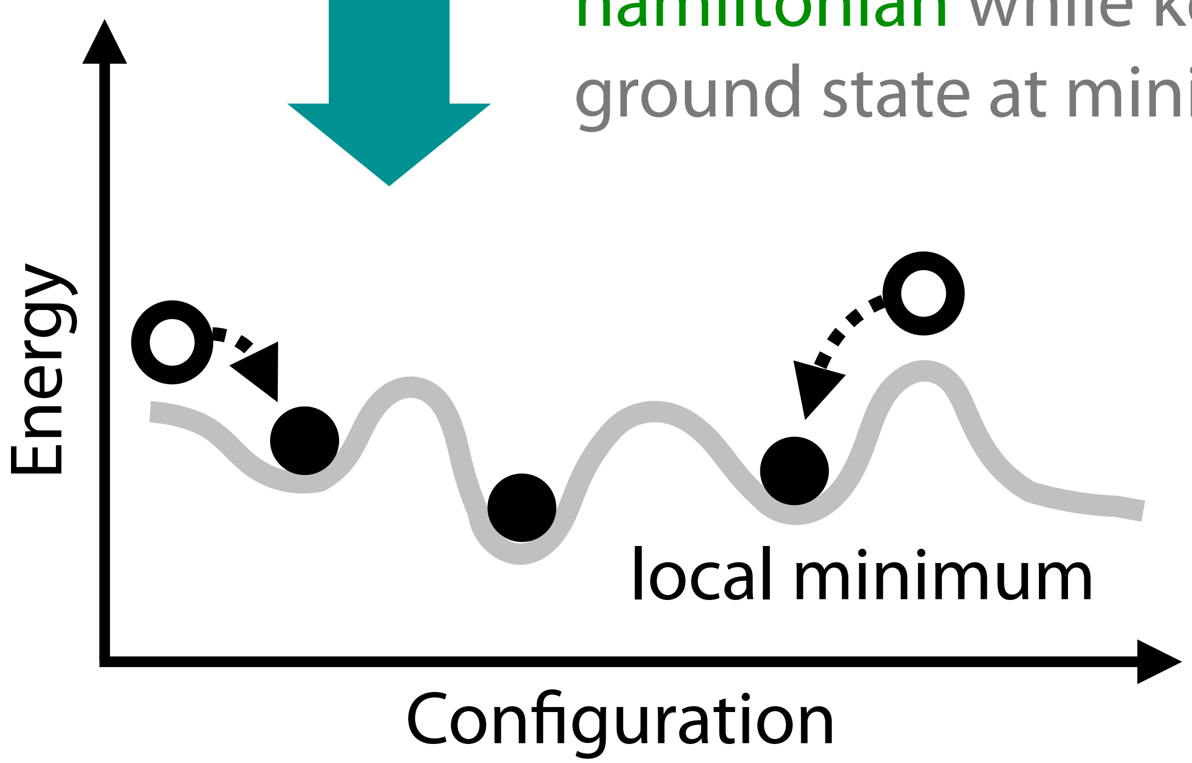
How Quantum Annealing Works

All configurations at ground state
for initial hamiltonian



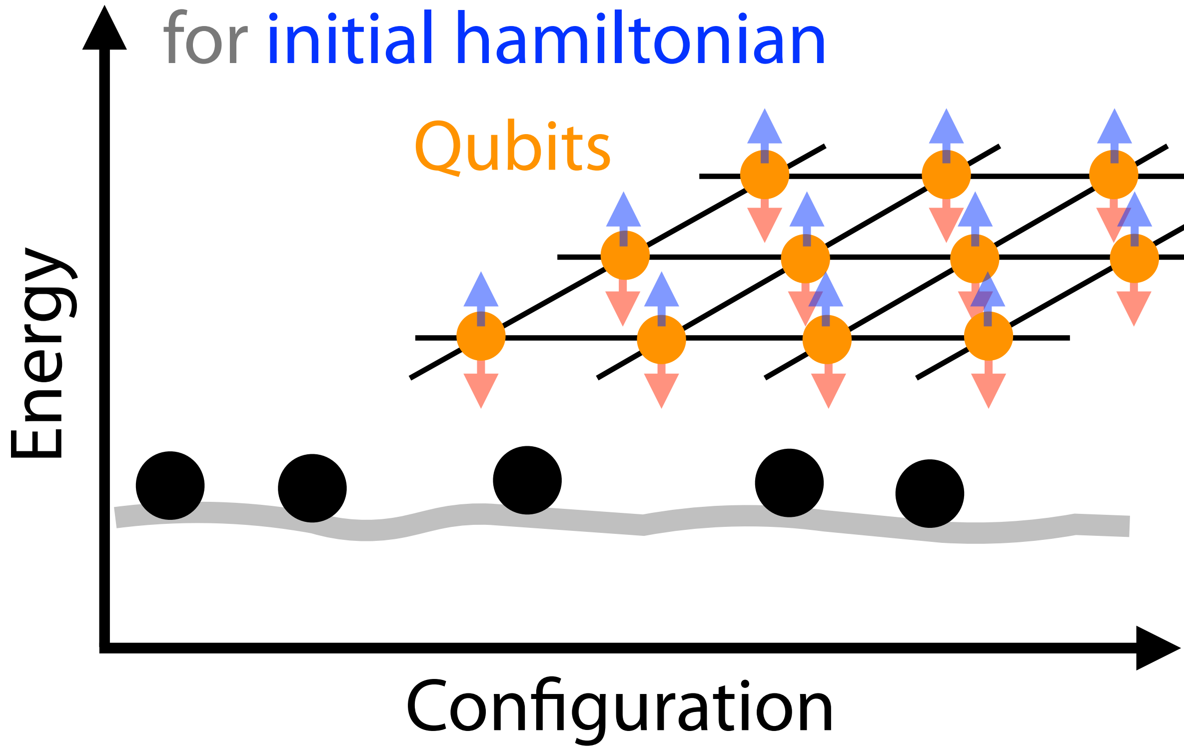


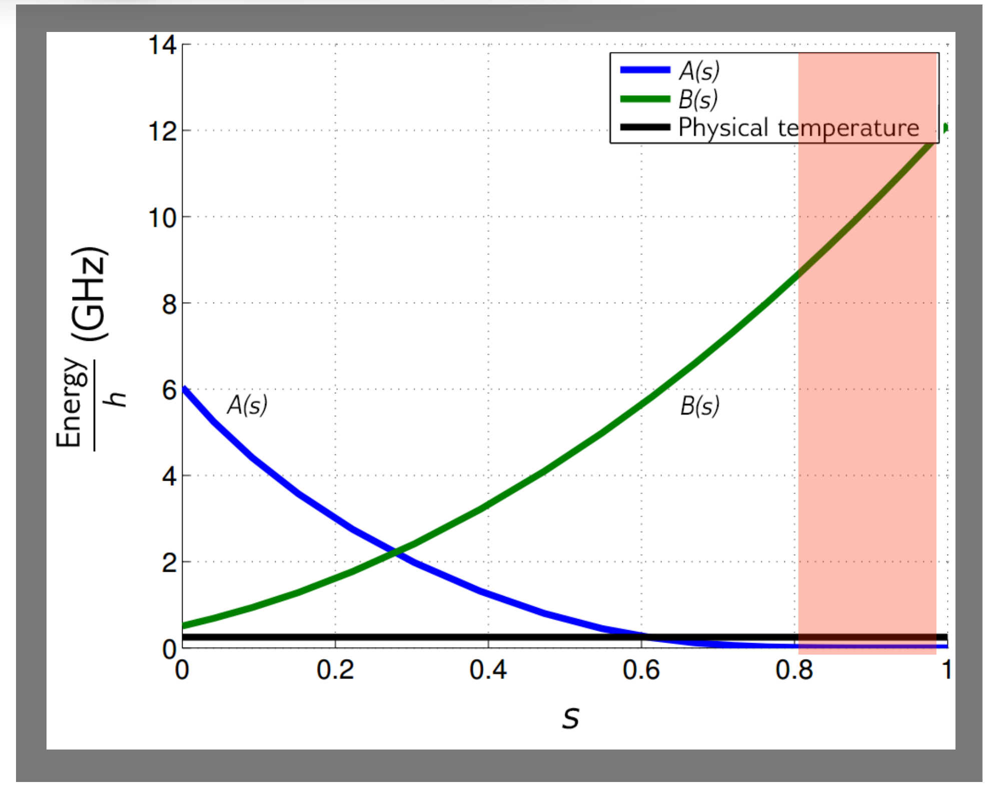
Slowly introduce problem
hamiltonian while keeping
ground state at minimum

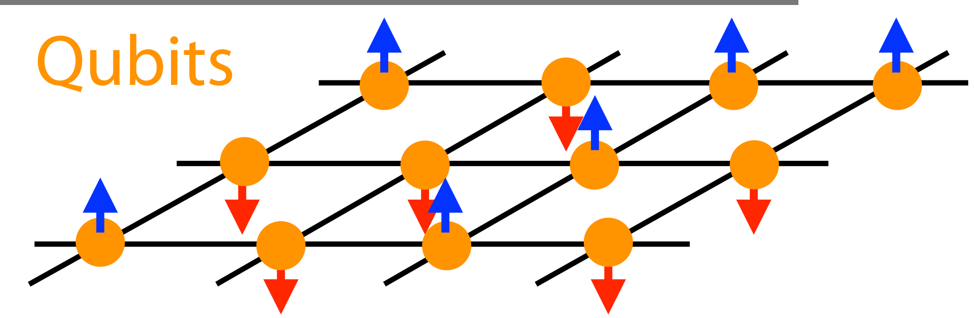


How Quantum Annealing Works

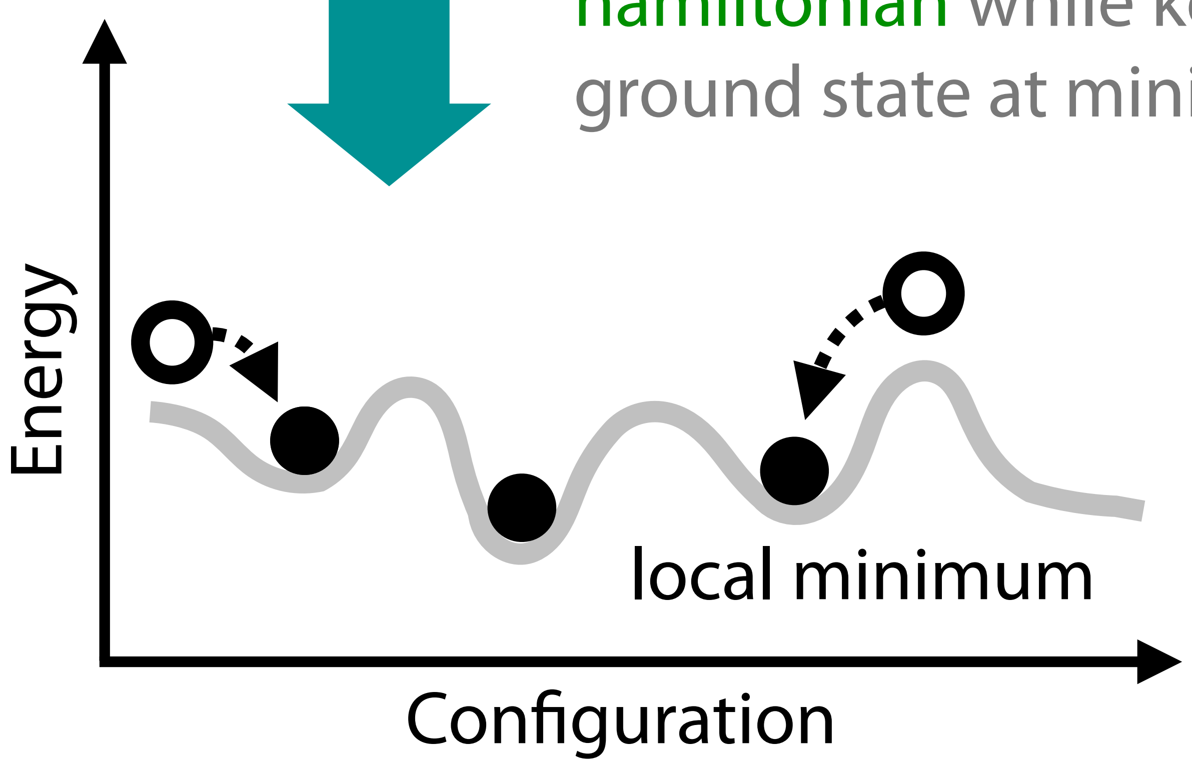
All configurations at ground state for initial hamiltonian

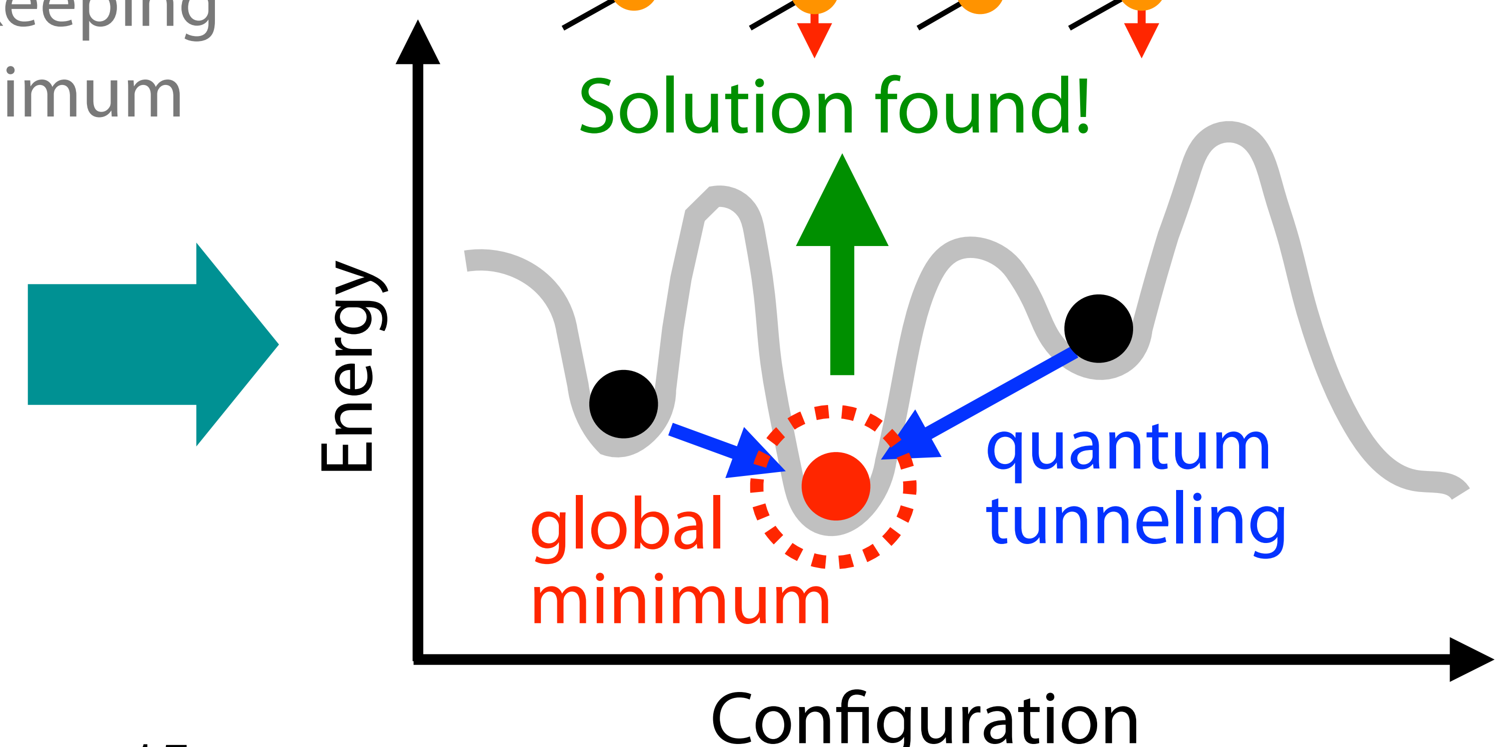






Slowly introduce problem hamiltonian while keeping ground state at minimum





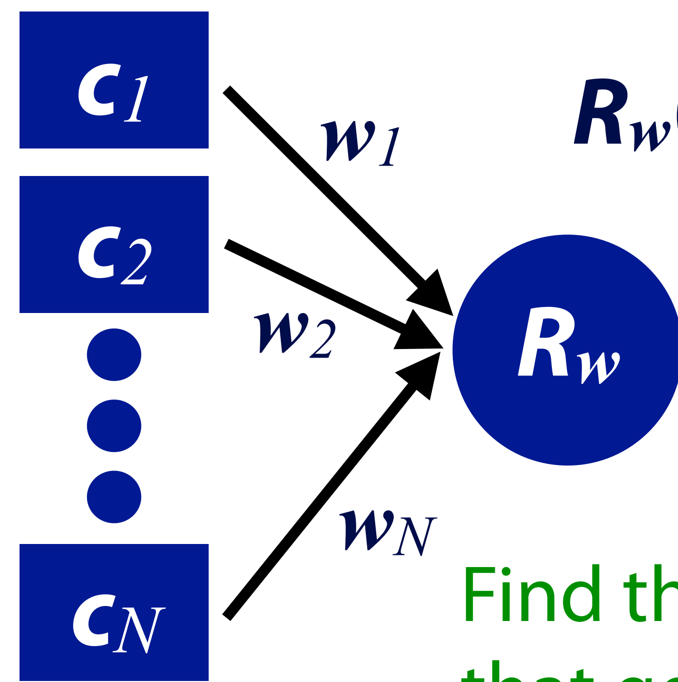
Quantum Machine Learning for HEP

- Classification with quantum annealing
- Classification with gate-based quantum algorithm
  - Wisconsin group
  - Tokyo ICEPP group

ML with Quantum Annealing

A. Mott *et al.*, [Nature 550, 375 (2017)]

- Have **weak classifier** $\boldsymbol{c}_i(x)$ from each input variable
- Define **strong classifier** $\boldsymbol{R}_w(x)$ as a linear combination of $\boldsymbol{c}_i(x)$



$$\boldsymbol{R}_w(x) = \sum w_i \boldsymbol{c}_i(x)$$

$$w_i \in \{0, 1\}^N$$

Find the configuration of w that gets $\boldsymbol{R}w$ to truth label y

Minimize objective function $\boldsymbol{\delta w} = ||y - \sum w_i \boldsymbol{c}_i(x)||^2$

$$\boldsymbol{\delta w} \;\propto\; \sum_{i,j} \boldsymbol{C}_{ij} w_i w_j + \sum_i (\lambda - 2\boldsymbol{C}_i) w_i$$

$$\boldsymbol{C}_{ij} = \sum \boldsymbol{c}_i(x)\boldsymbol{c}_j(x)$$

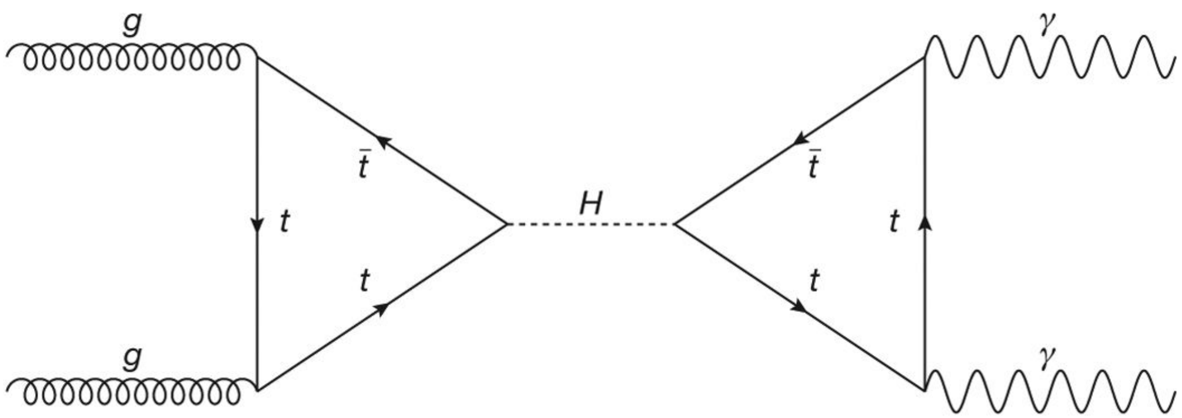
$$\boldsymbol{C}_i = \sum \boldsymbol{c}_i(x) y$$

➡ Solving as **QUBO** (Quadratic Unconstrained Binary Optimization) using quantum annealing

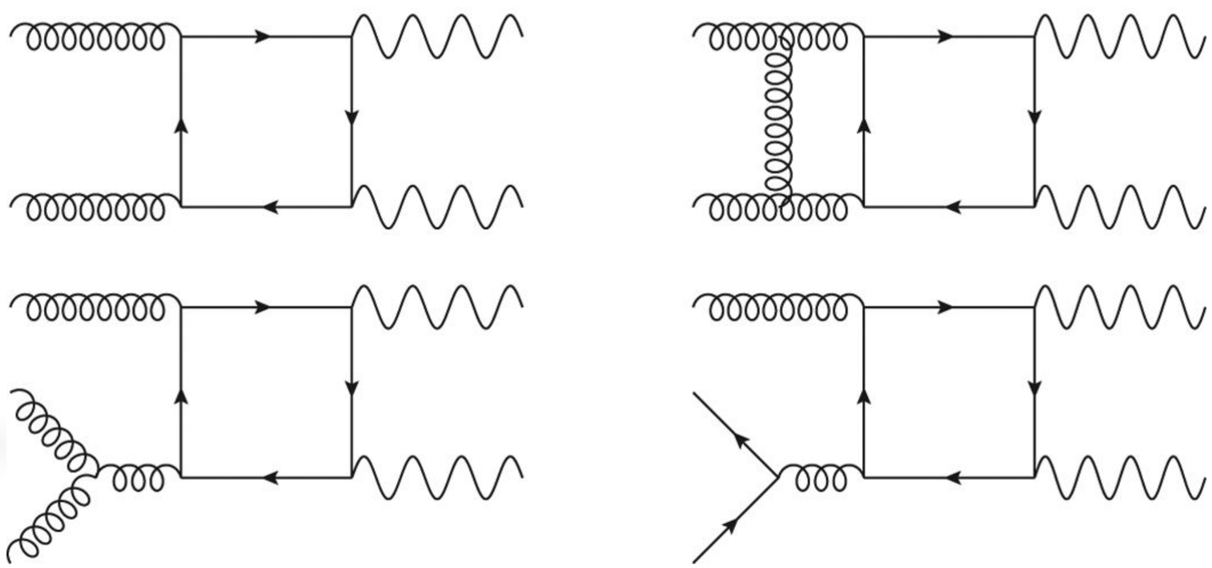
QAML Classification

Attempt to classify $H \rightarrow \gamma\gamma$ signal from background

Signal



Background

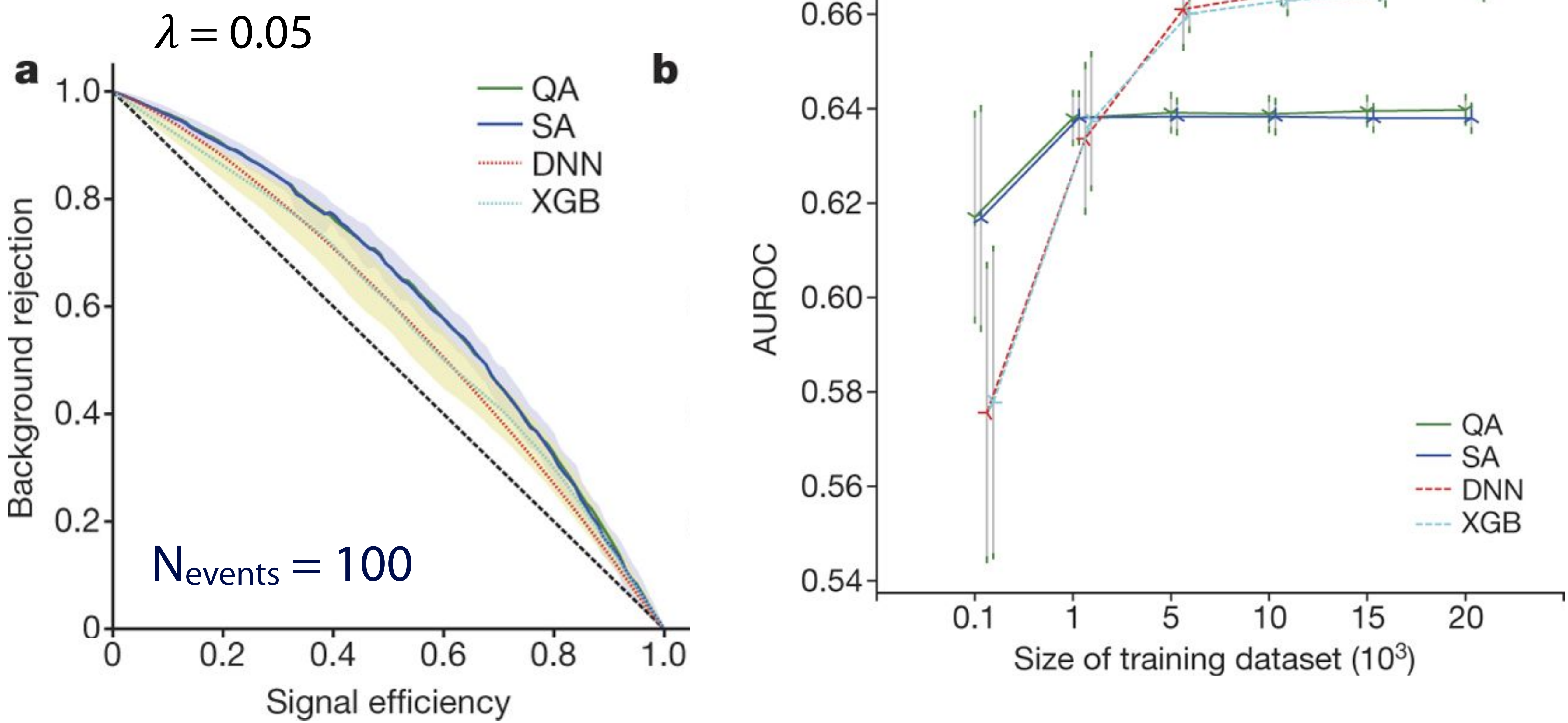


- Photon p_T cuts applied for realistic trigger
- Di-photon mass cut applied to select "Higgs-like" events

36 variables constructed in total ➞ Used as weak classifiers

QAML Results

QA = D-Wave



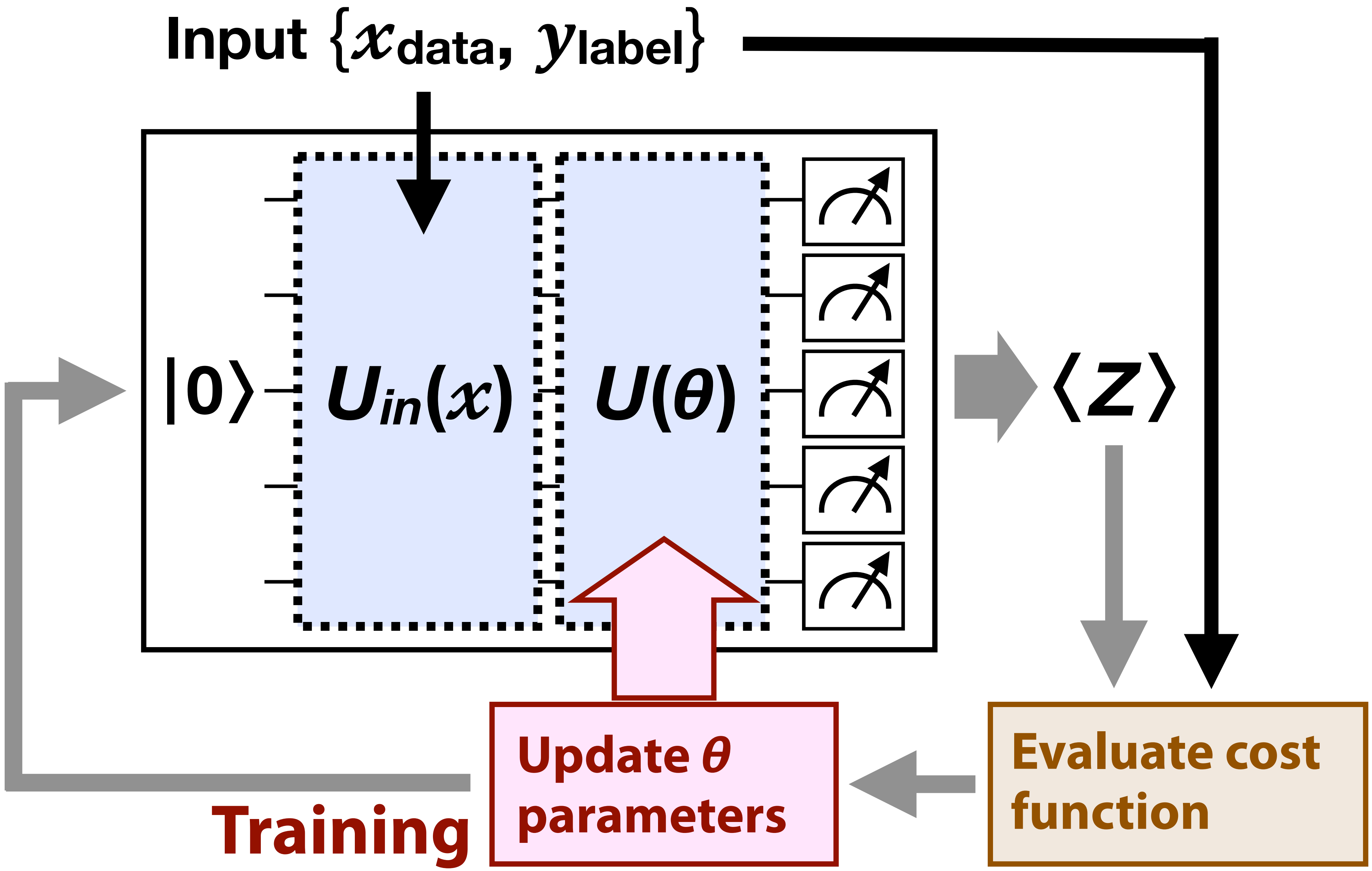
- Annealing (Quantum, QA or Simulated, SA) has comparable performance to the classical ML method like DNN or XGB
- Indication that annealing has better performance at small sample
- DNN or XGB becomes more performant when adding more data

ML with Gate-based Algorithm

Use method based on variational quantum circuit

V. Havlicek *et al.*, arXiv:1804.11326

K. Mitarai *et al.*, arXiv:1803.00745

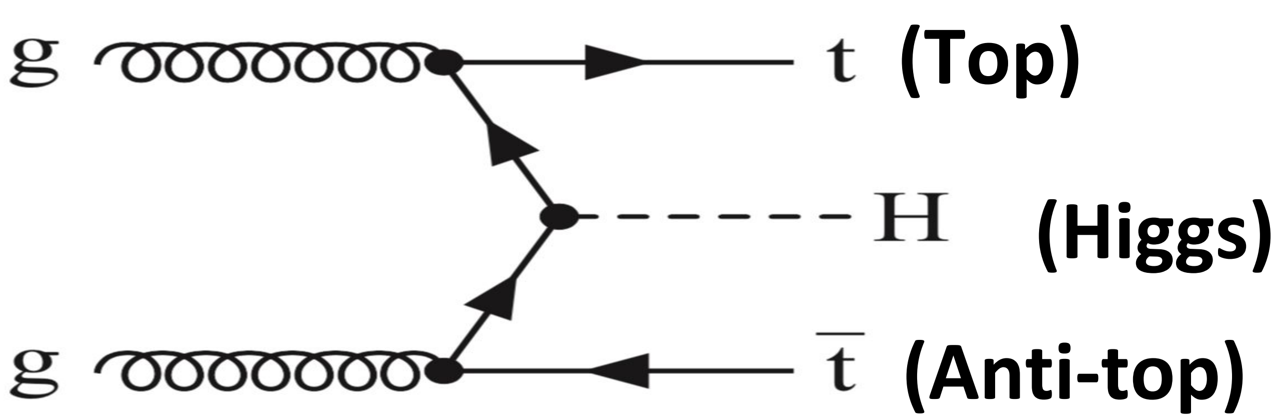


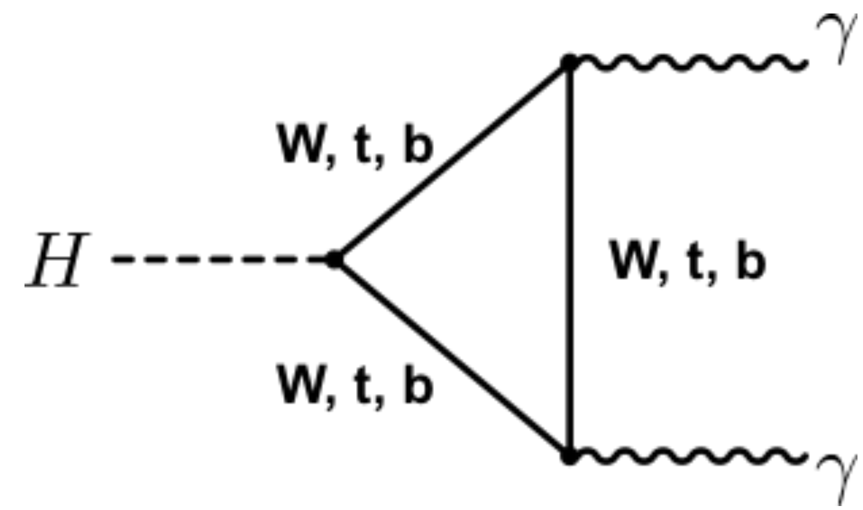
ttH($\to\gamma\gamma$) Classification ~ Wisconsin ~

Talk by S. Sun (Wisconsin) at CERN Openlab Technical Workshop

Application of variational quantum algorithm to HEP data analysis

Classification of ttH($\to\gamma\gamma$) signal



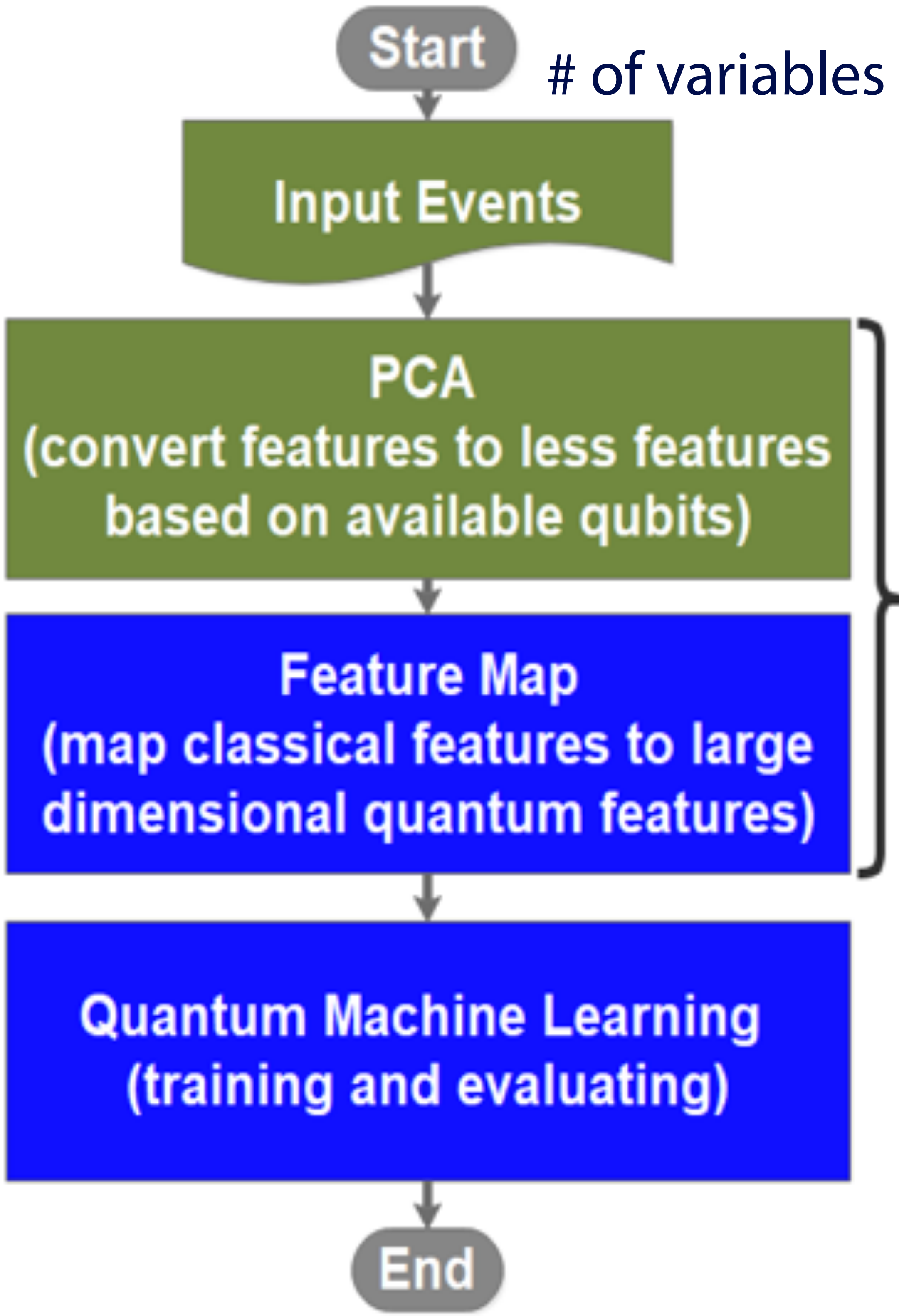


Performance studied using simulator and IBM quantum machine

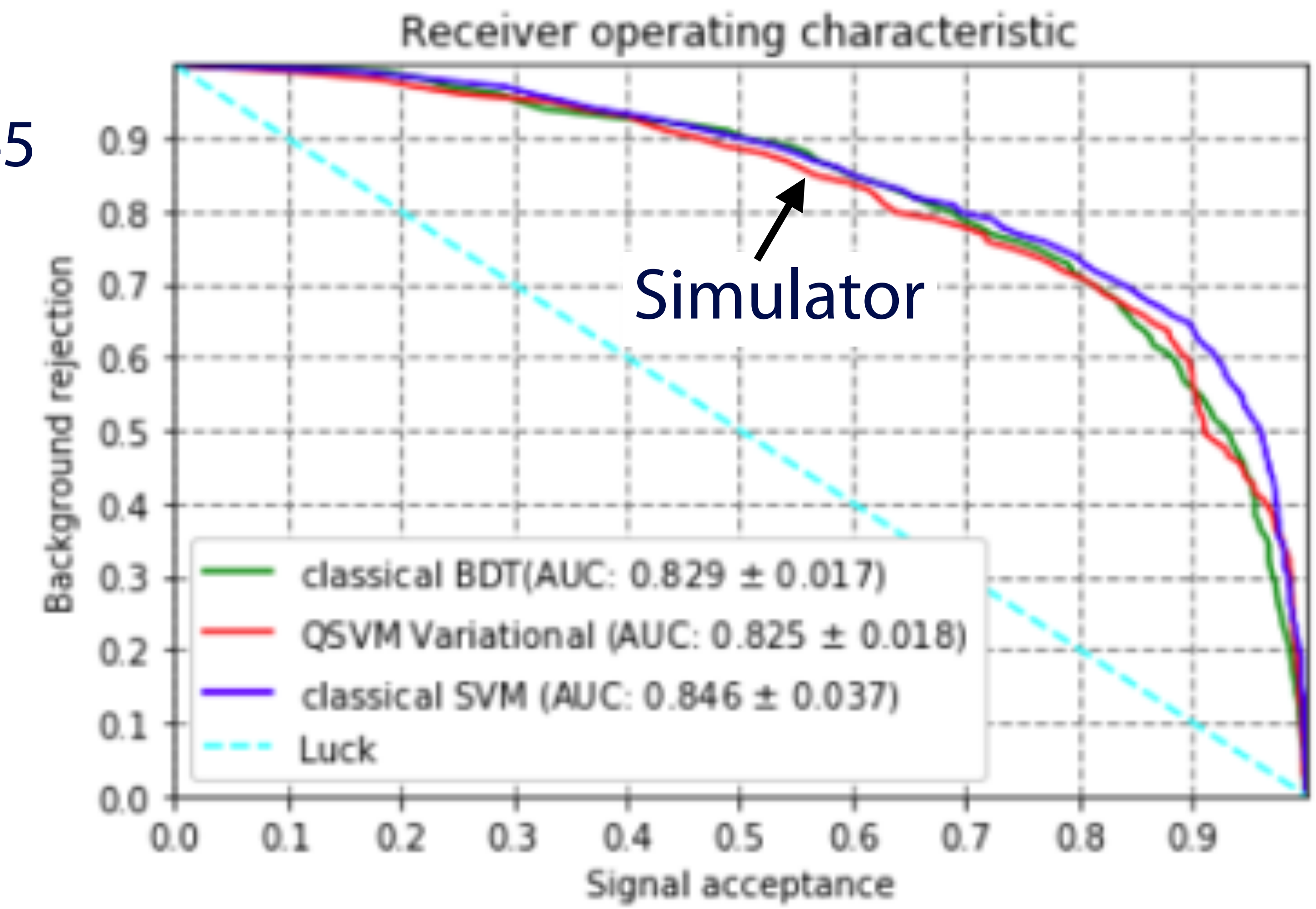
ttH(→$\gamma\gamma$) Classification

Workflow of gate-based QML

# of variables = 45



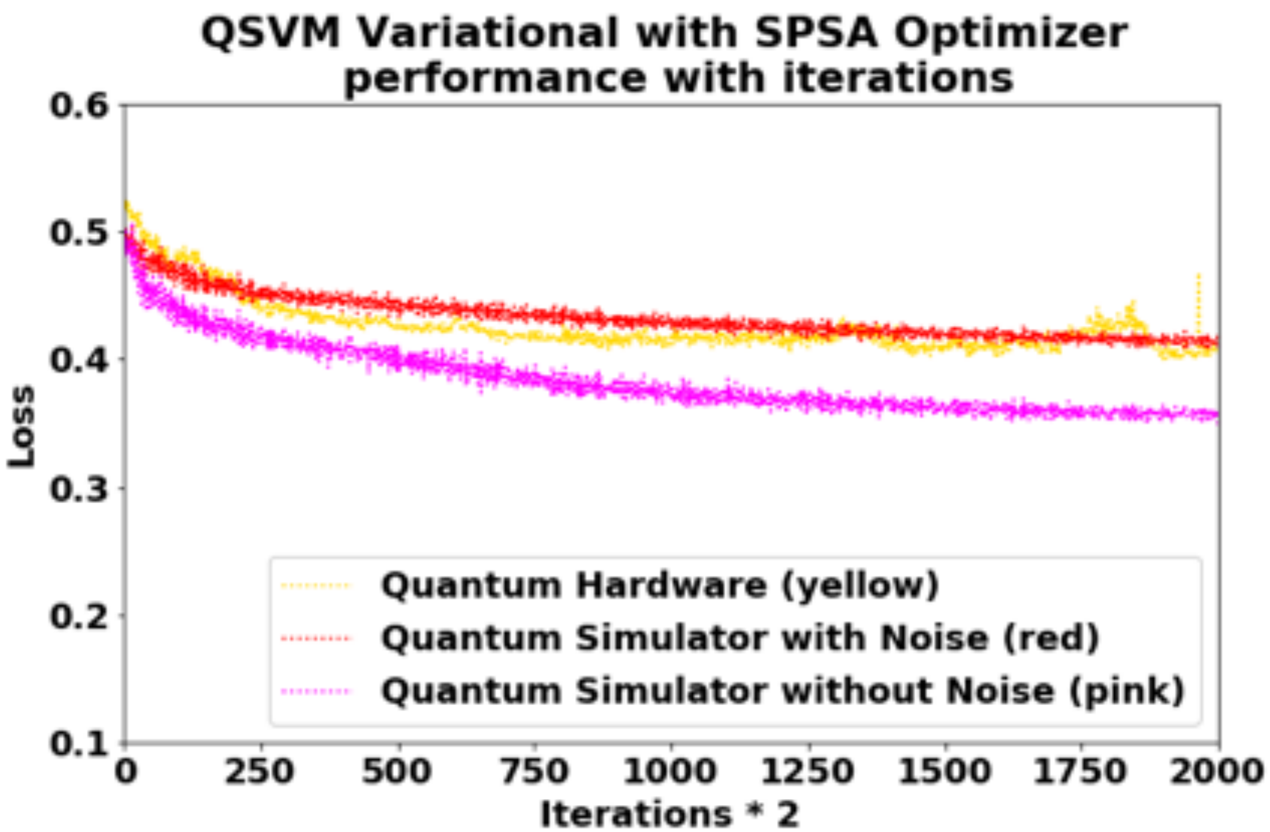
5 variables, 200 events

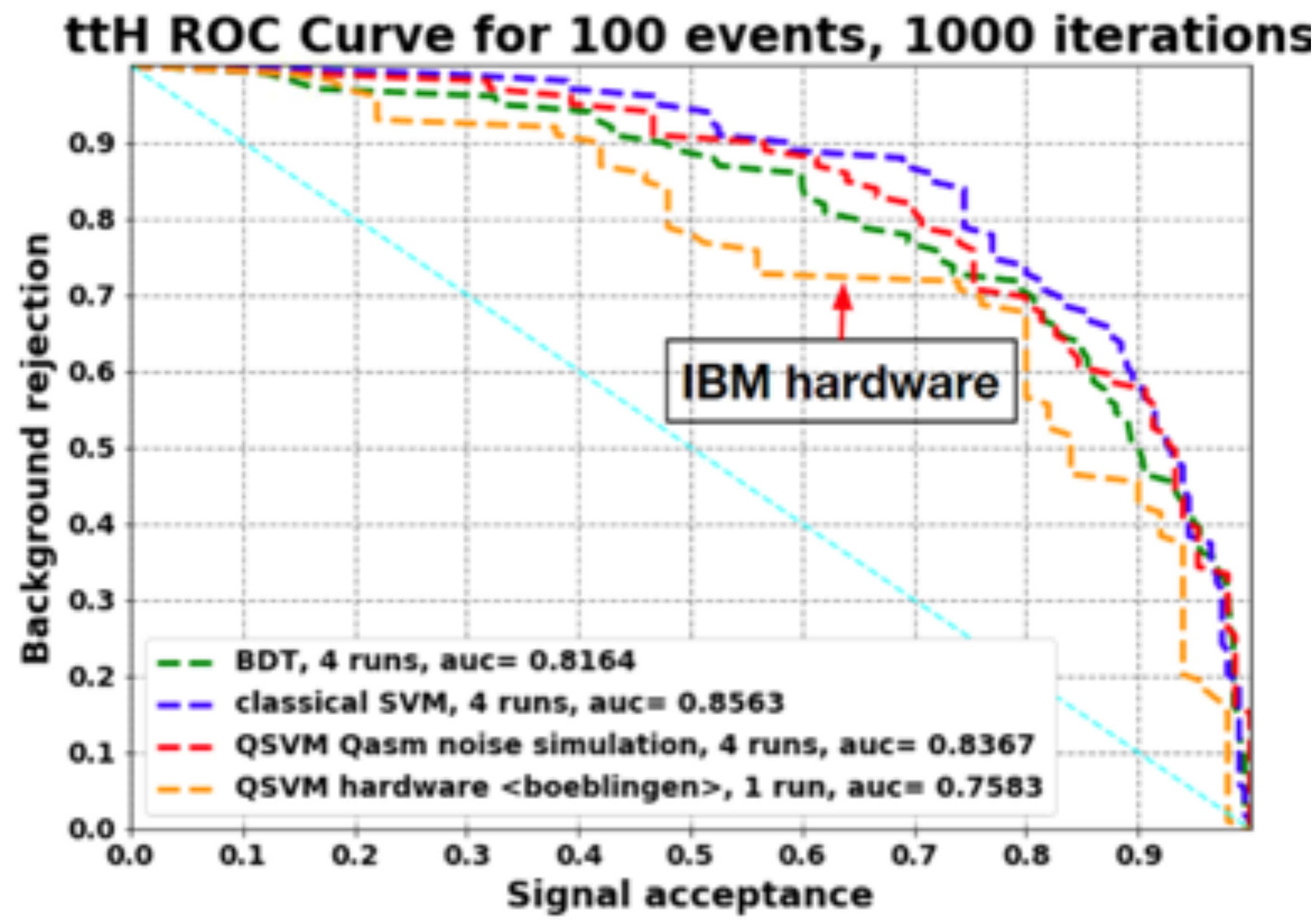


ttH(H→$\gamma\gamma$) AUC	AUC
Classical SVM	0.846 ± 0.037
Classical BDT	0.829 ± 0.017
Quantum Simulation	0.825 ± 0.018

ttH(→$\gamma\gamma$) Classification

5 variables, 100 events





Tested on 20-qubit IBM machine:

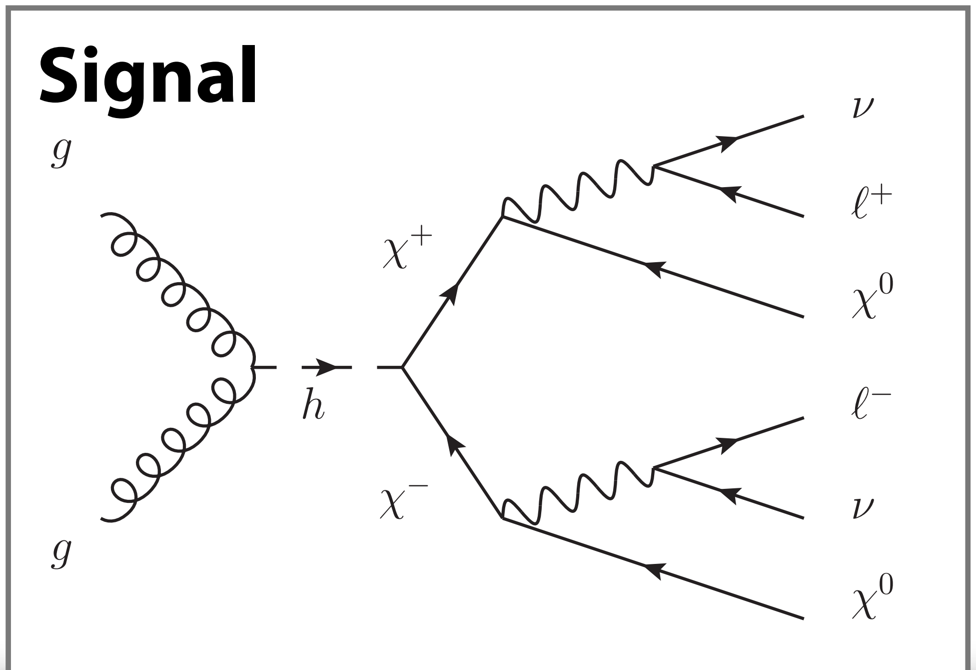
- Quantum hardware learns how to differentiate between signal and background
- Performance (AUC) ~10% worse on hardware than quantum simulator/ classical method

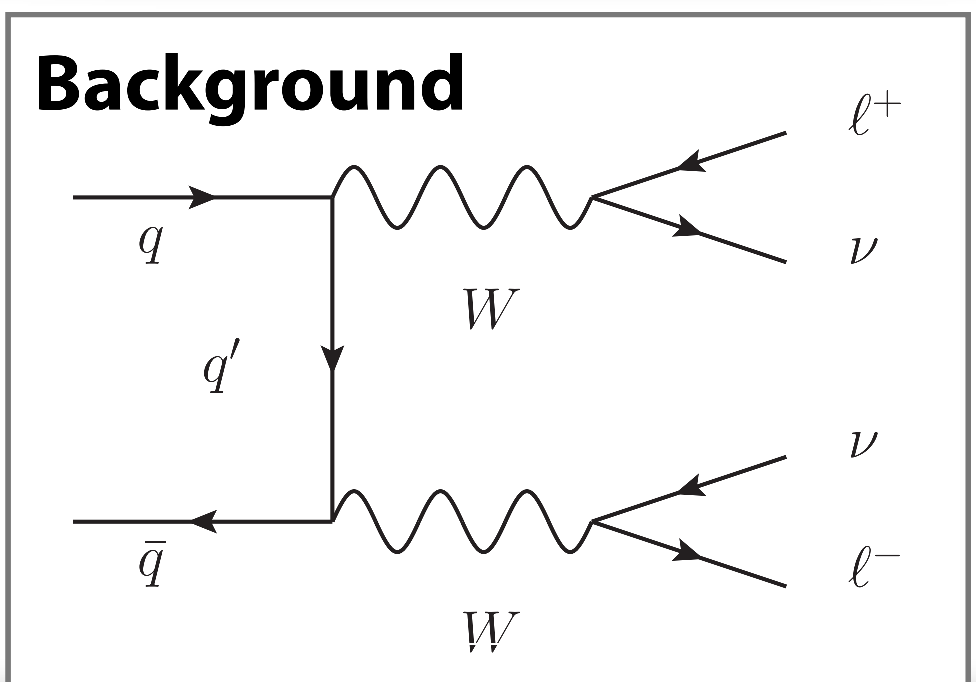
ttH(H→$\gamma\gamma$) AUC	AUC
Classical SVM	0.856
XGBoost BDT	0.816
Quantum Simulation with Noise	0.837
Quantum Hardware	0.758

SUSY Classification ~ Tokyo ICEPP ~

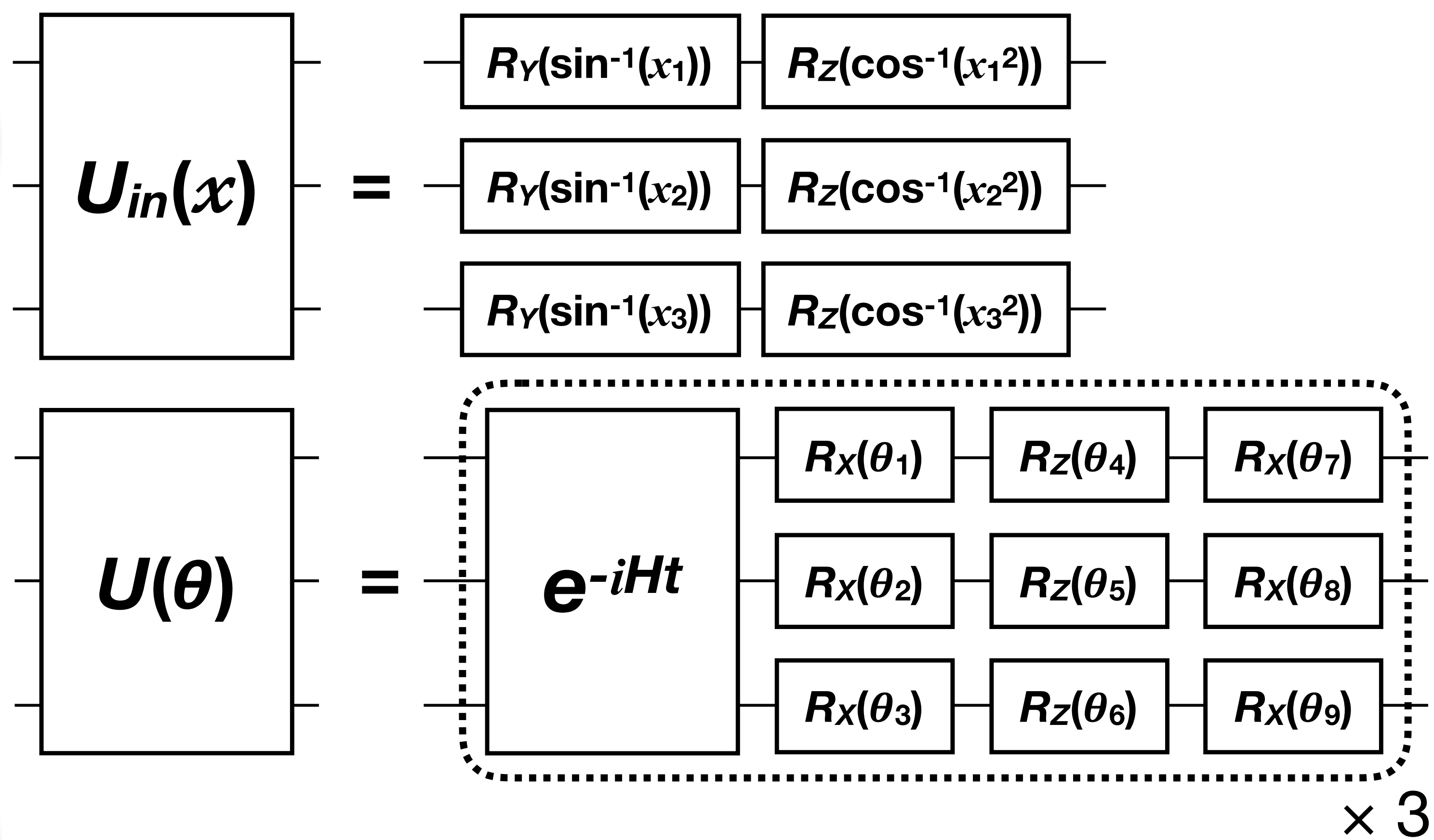
SUSY signal classification using simulator and IBM quantum machine

arXiv:[2002.09935]
Submitted last week!



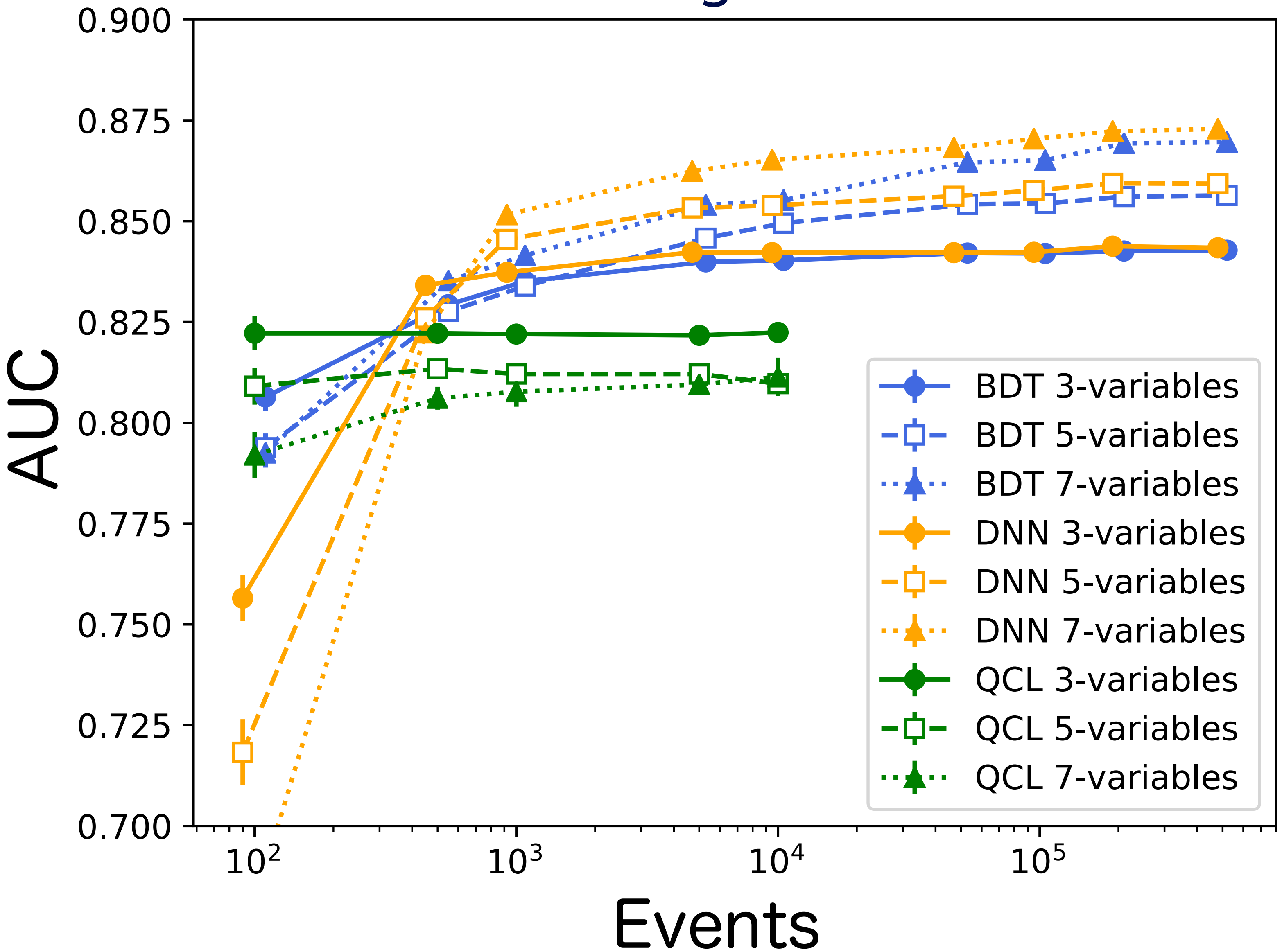


- Chargino-pair production with leptonic *W* decays as signal
- [SUSY dataset] in UC Irvine machine learning repository used



SUSY Classification

QCL = Quantum algorithm with simulator



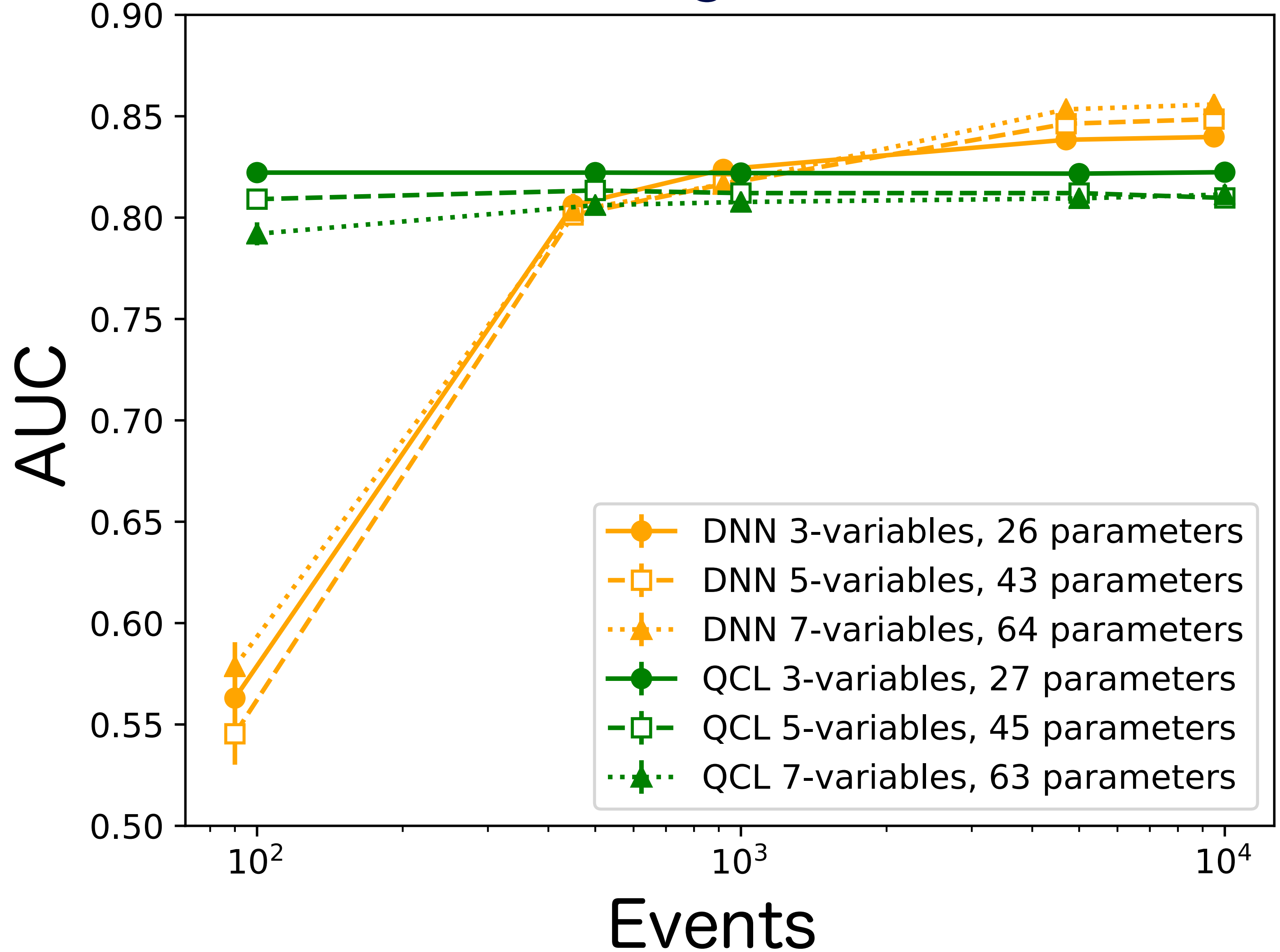
Compared with BDT and DNN :

- BDT and DNN models optimized at each training set to avoid over-training
- Classical algorithms out-perform at large training set

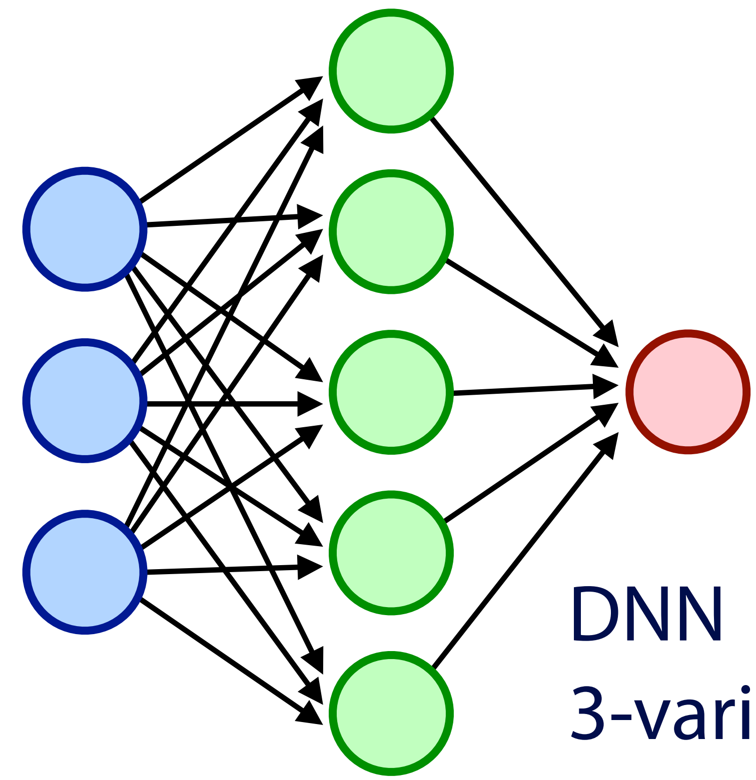
➡ Performance of quantum algorithm comparable to BDT/DNN at small training set with small # of variables

SUSY Classification

QCL = Quantum algorithm with simulator



Compared with DNN with similar # of parameters



DNN
3-variables

➡ Possible advantage of quantum algorithm over DNN at small training set, when # of parameters are chosen to be same

Other HEP Applications

Tracking

H. Gray, P. Calafiura, et. al (LBNL)

arXiv:1902.08324

R. Sawada, KT (Tokyo)

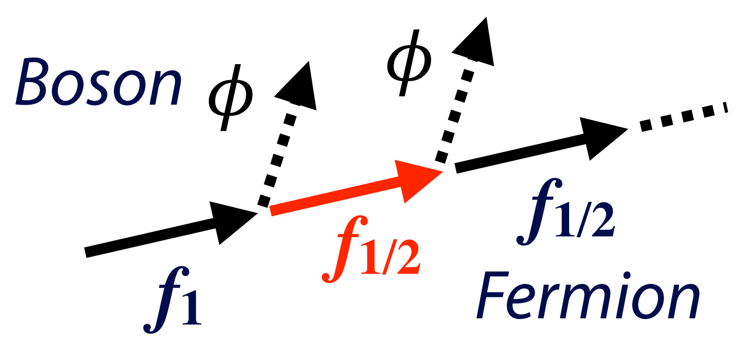
- Quantum Annealing
  - QallSe (HEP.QPR)
- Digital (non-quantum) Annealing
- Gate-based algorithm

Parton Shower Simulation

C. Bauer, B. Nachman et. al (LBNL)

arXiv:1904.03196, 1901.08148

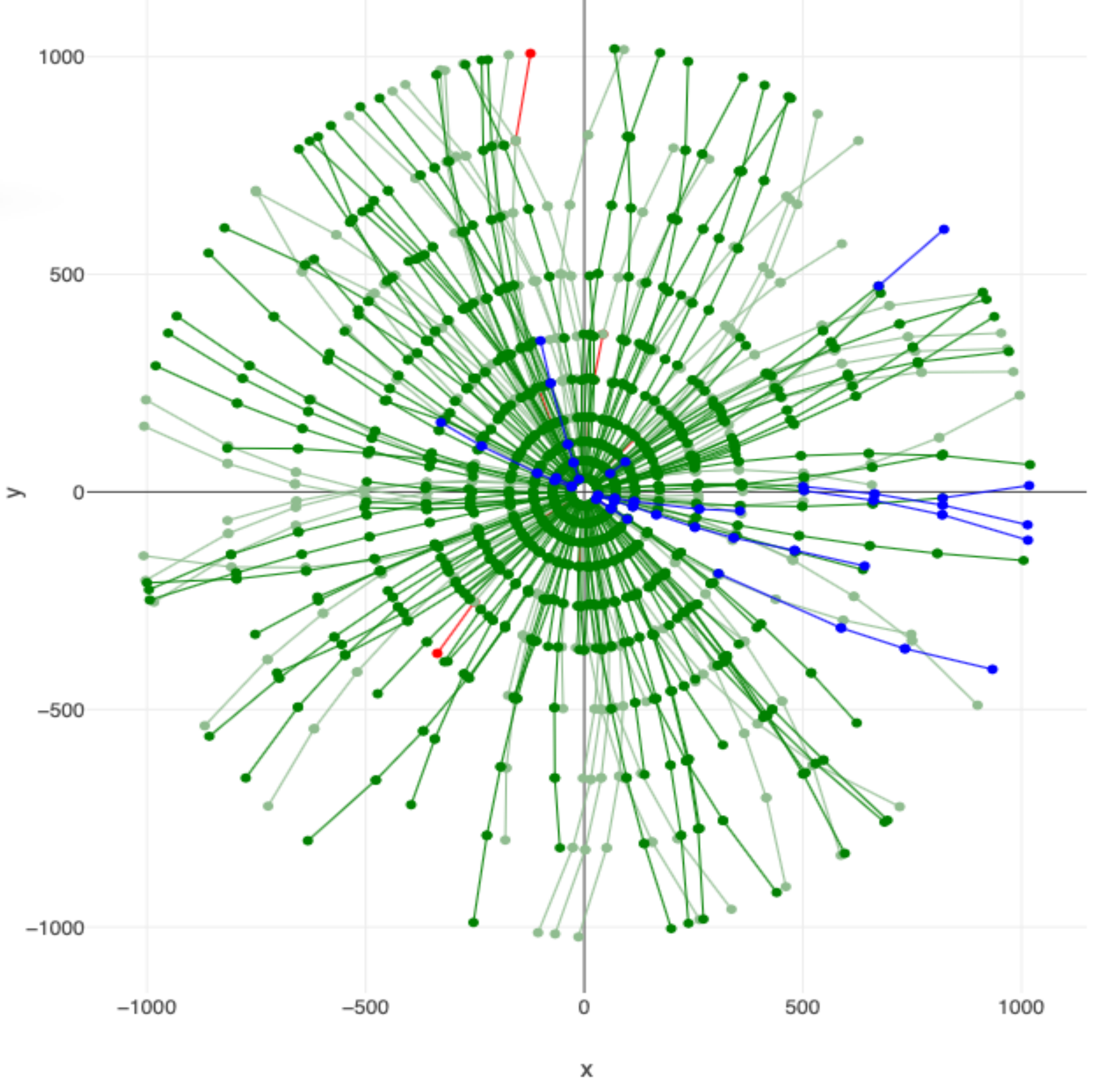
- Interference from different intermediate particles

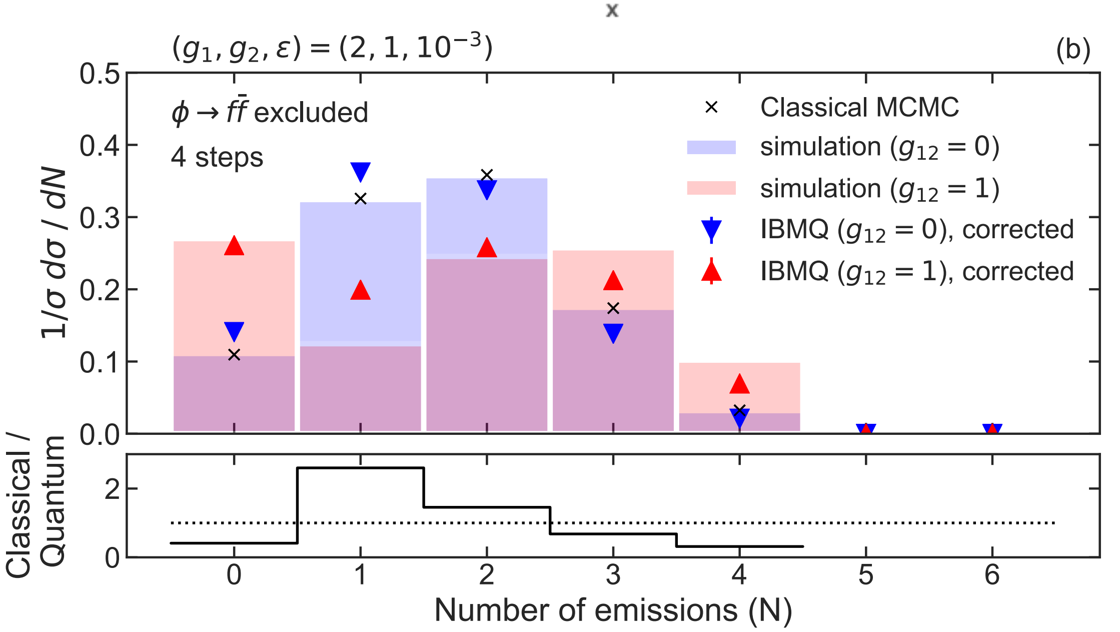


Vertexing arXiv:1903.08879

Unfolding arXiv:1908.08519

Jet Clustering arXiv:1908.08949





Growing interests over the last years!!

Summary

- Presented examples of quantum computing application to machine learning
- Only at the (very) beginning of the exploration of quantum computing for HEP
- Need *useful* application to take advantage of quantum computing for future QC development (ML as an example)

Many technological challenges ahead (e.g, scalability) before making quantum computers competitive to classical ones, but…

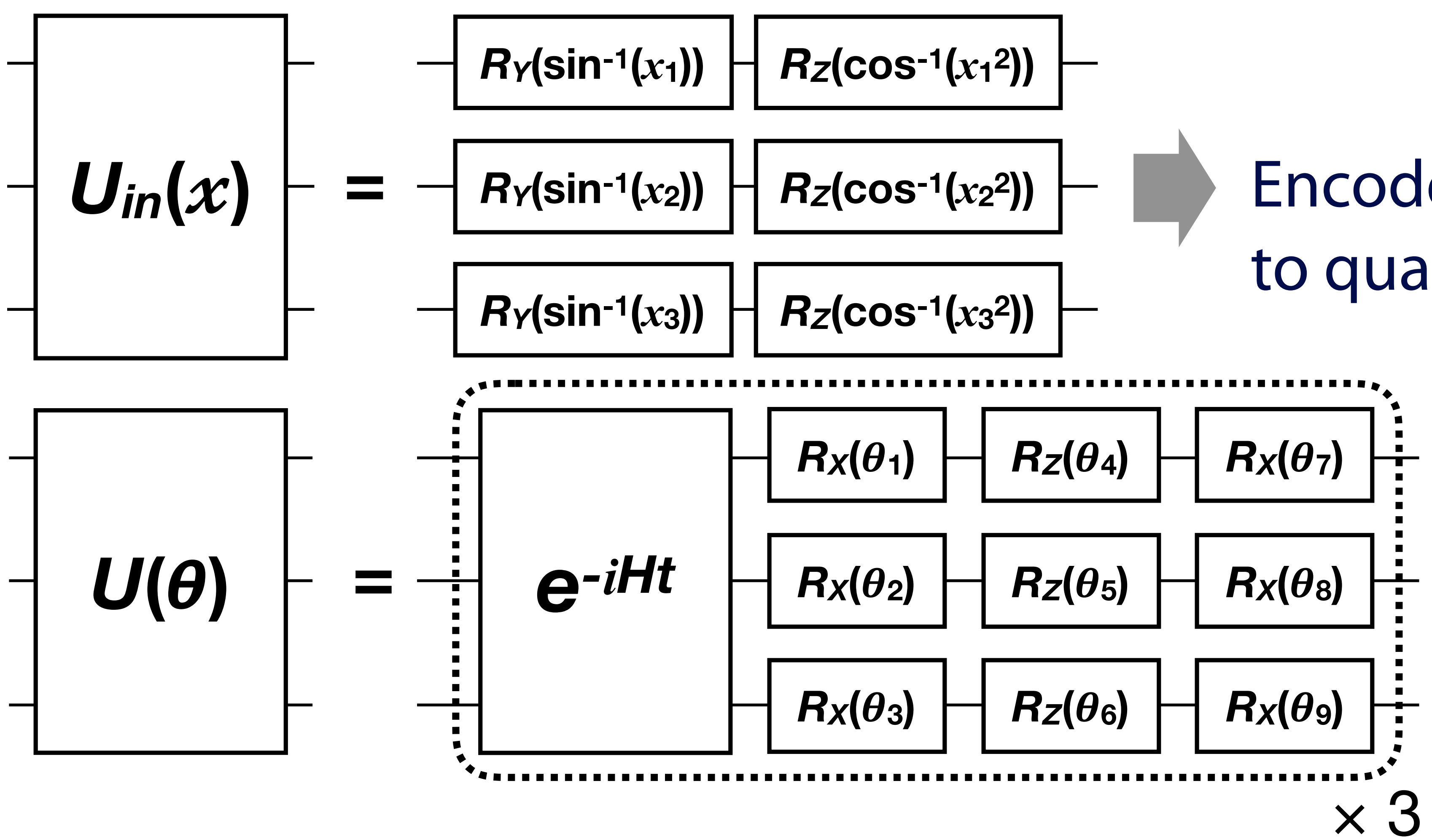
What to expect if x100 more powerful QC in ~7 years (when the HL-LHC starts)?

Backup

Quantum Circuit Learning

Implemented using [Qulacs] simulator
(implemented in C/C++ with Python interface)

K. Mitarai *et al.*,
arXiv:1803.00745



Encode input data to quantum state

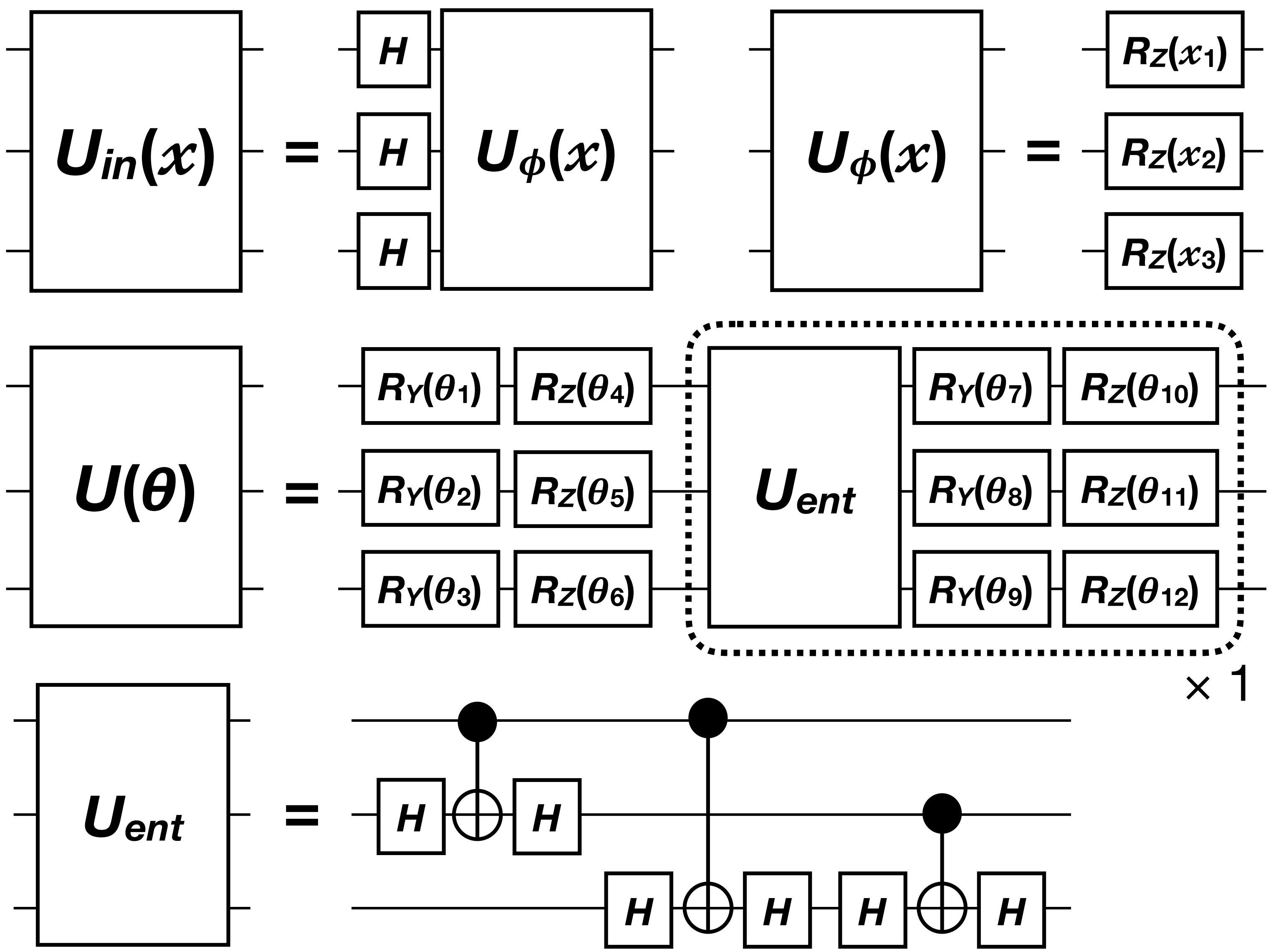
- Use time-evolution gate e^{-iHt} to create entanglement
- Rotation gates with parameters

- Output states from $U(\boldsymbol{\theta})$ measured using Pauli-Z operators
- Cross-entropy used as cost function
- Parameters optimized using COBYLA by minimizing the cost function

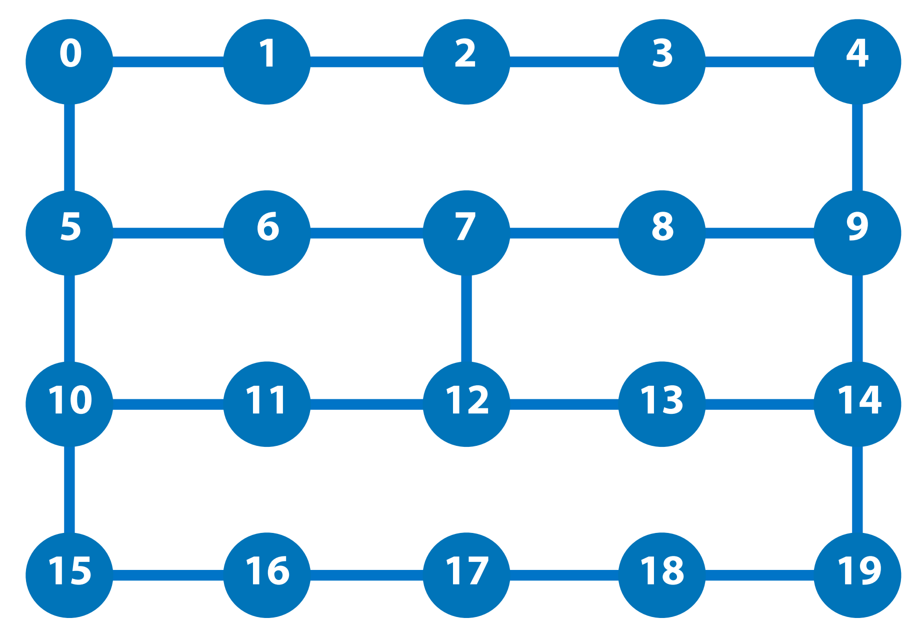
Variational Quantum Classification

Implemented using Qiskit Aqua framework

V. Havlicek *et al.*, arXiv:1804.11326



Johannesburg

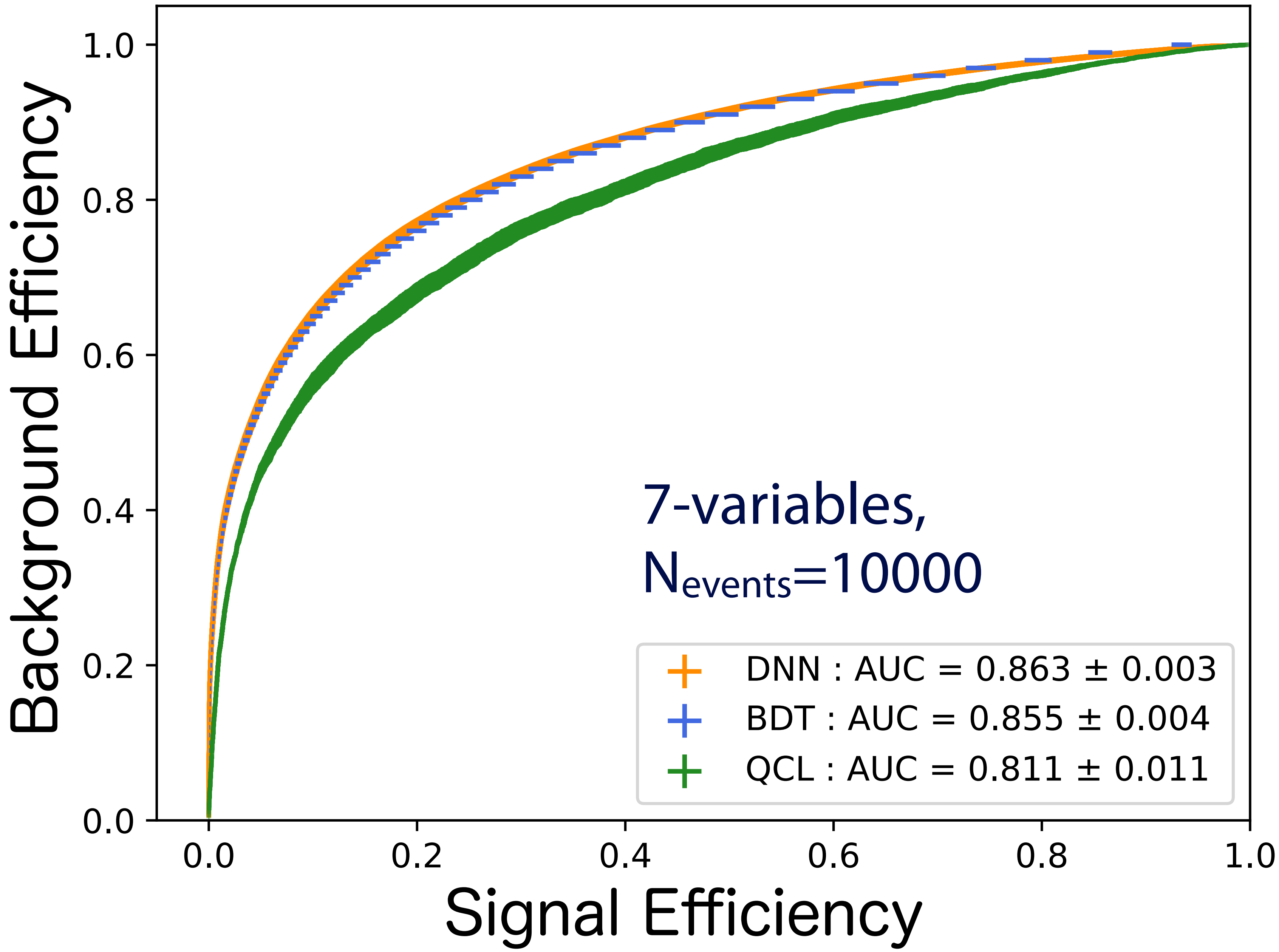


- Single-qubit rotation gates in $U_{in}(x)$
- Entangling gate (U_{ent}) + rotation gates in $U(\theta)$
- Cross-entropy loss with COBYLA minimization

Tested using 20-qubit IBM Q Network device and QASM simulator

QCL Results (Simulator)

arXiv:2002.09935

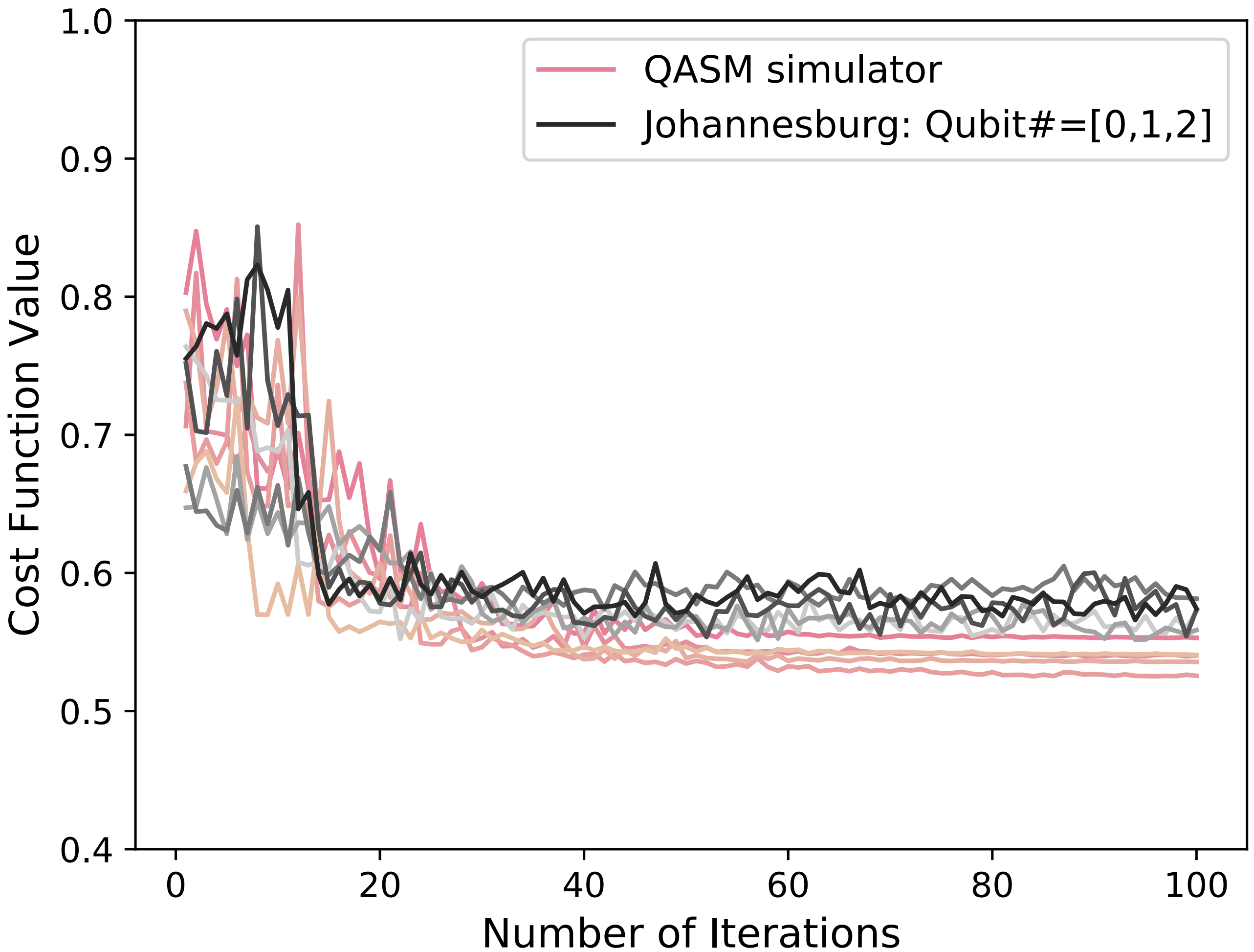


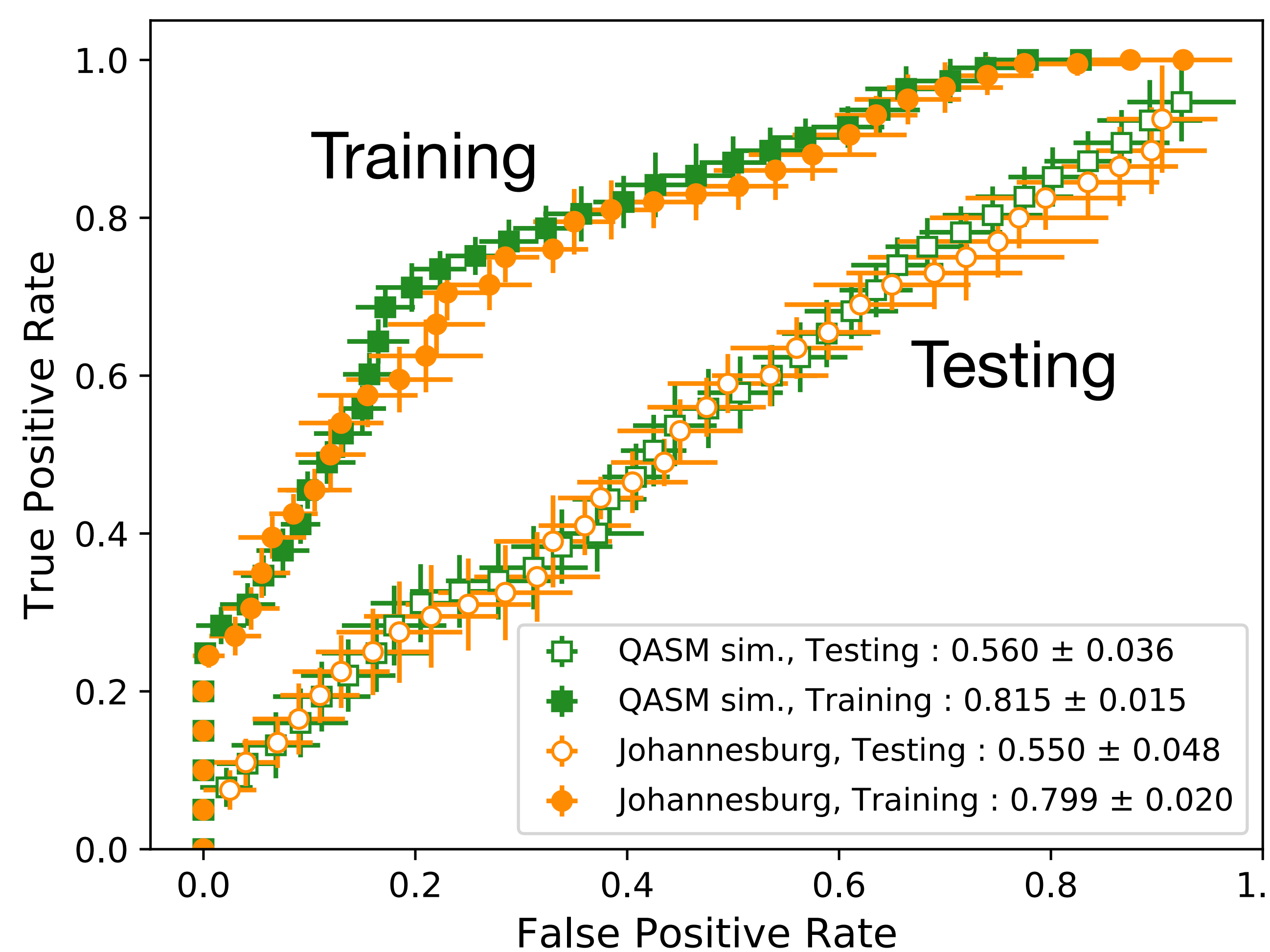
Compared with ML methods: Boosted-Decision Tree and Deep NN

- BDT : Gradient boost, 1-3 max depth, 10-1000 #trees
- DNN : Dense, 2-6 hidden layers, 16-256 nodes, RELU, Adam, $\varepsilon_{learning}$=0.001

VQC Results (Simulator & Hardware)

arXiv:2002.09935



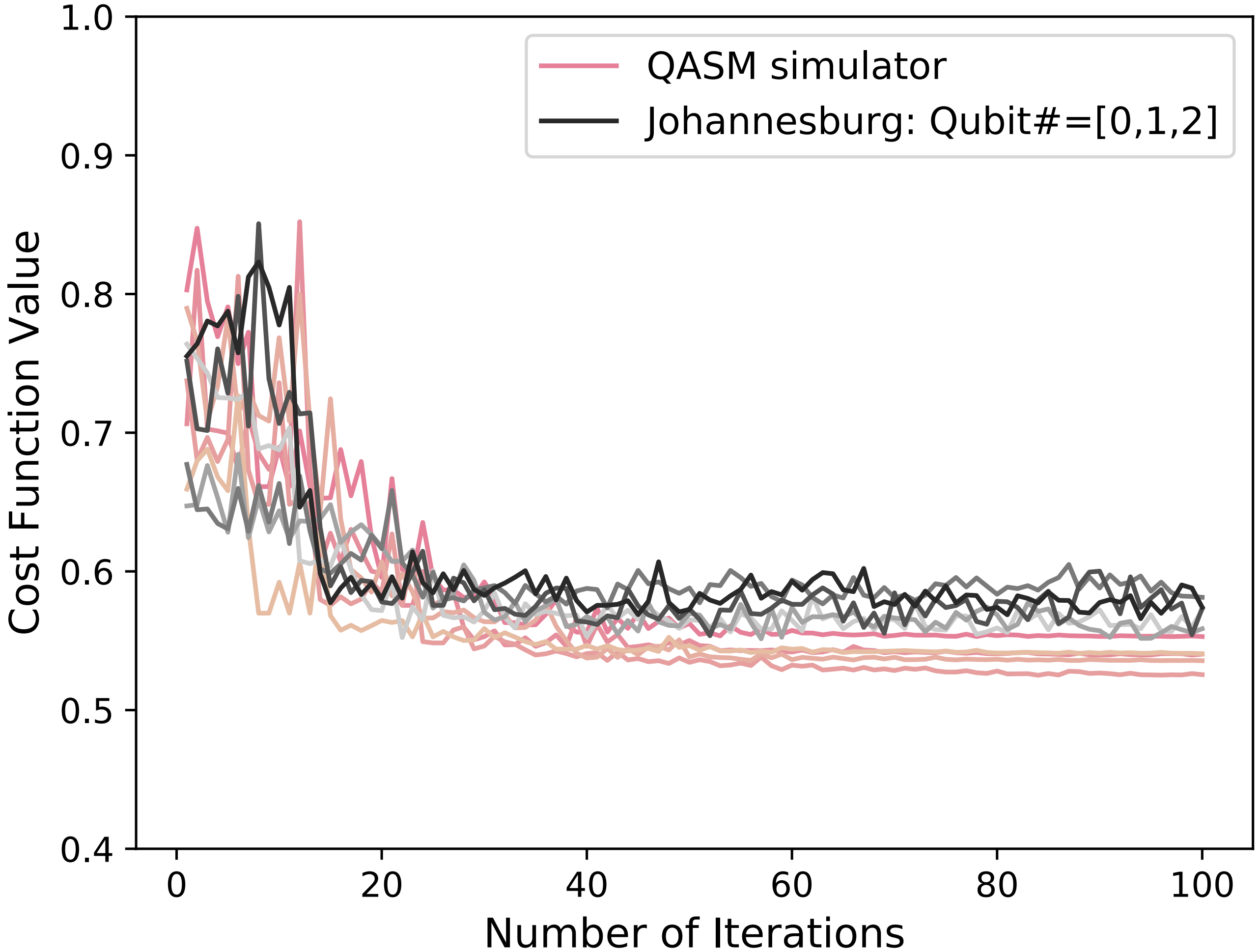


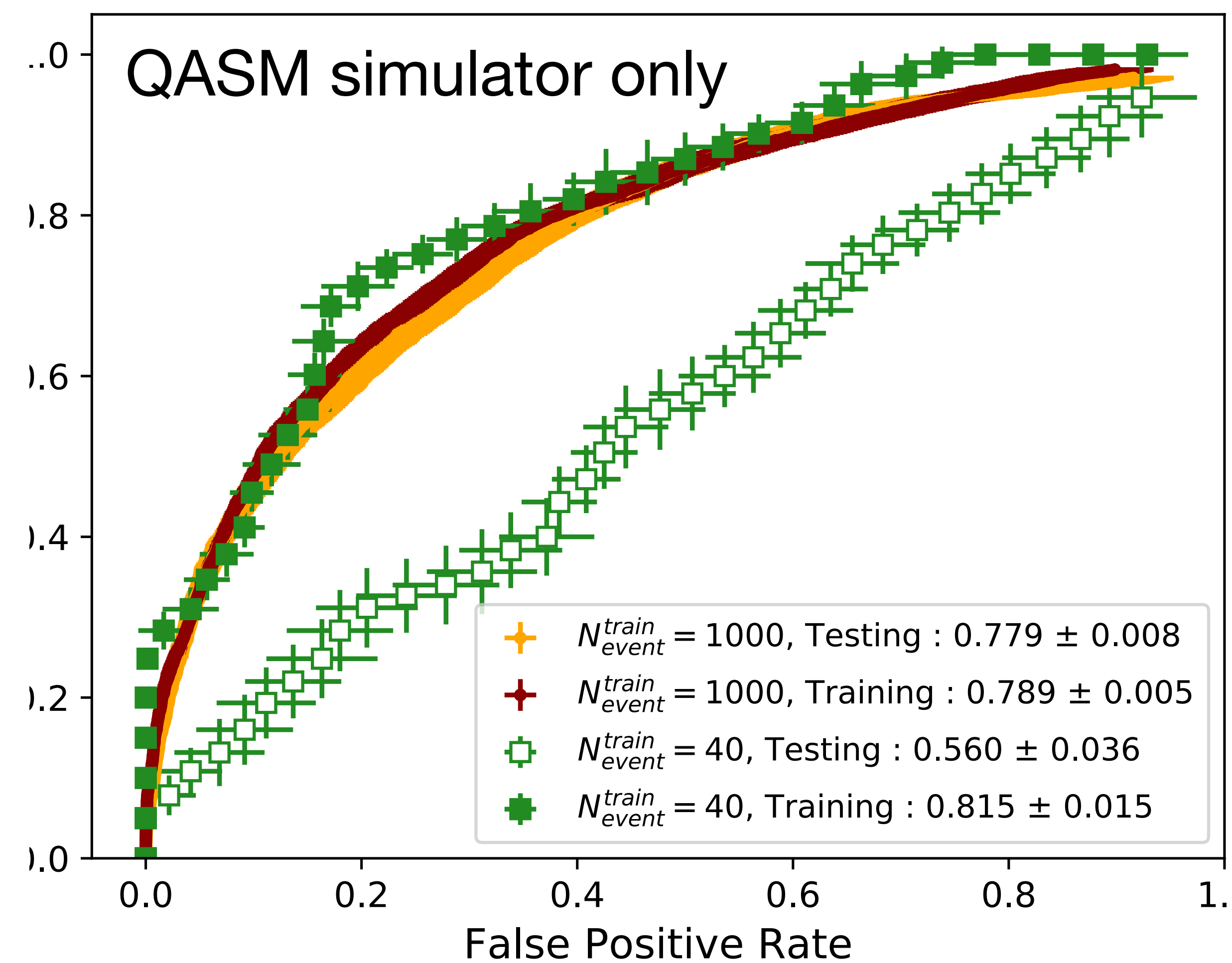
Tested 3-variable classification with 40 training events

- Cost function reaches minimum after ~50 iterations
  - Slight offset for real device (likely caused by error due to noise)
- ROC curves indicate VQC acquires discrimination power with real device
  - Significant over-training observed due to small sample size

VQC Results (Simulator & Hardware)

arXiv:2002.09935





Tested 3-variable classification with 40 training events

- Cost function reaches minimum after ~50 iterations
  - Slight offset for real device (likely caused by error due to noise)
- ROC curves indicate VQC acquires discrimination power with real device
  - No over-training seen once dataset size is increased

Input Features