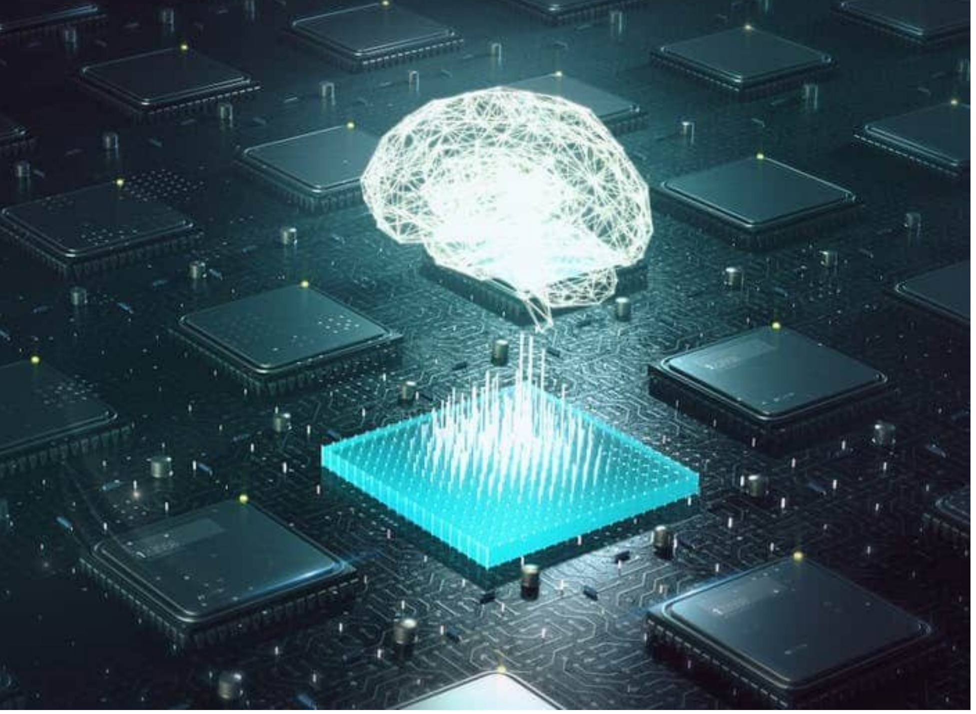
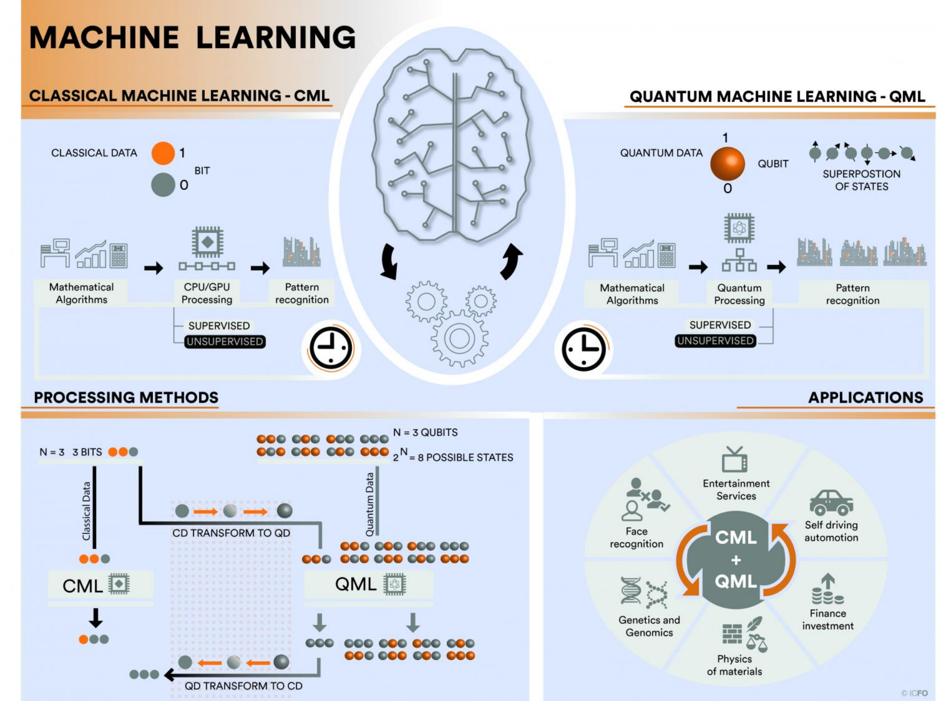


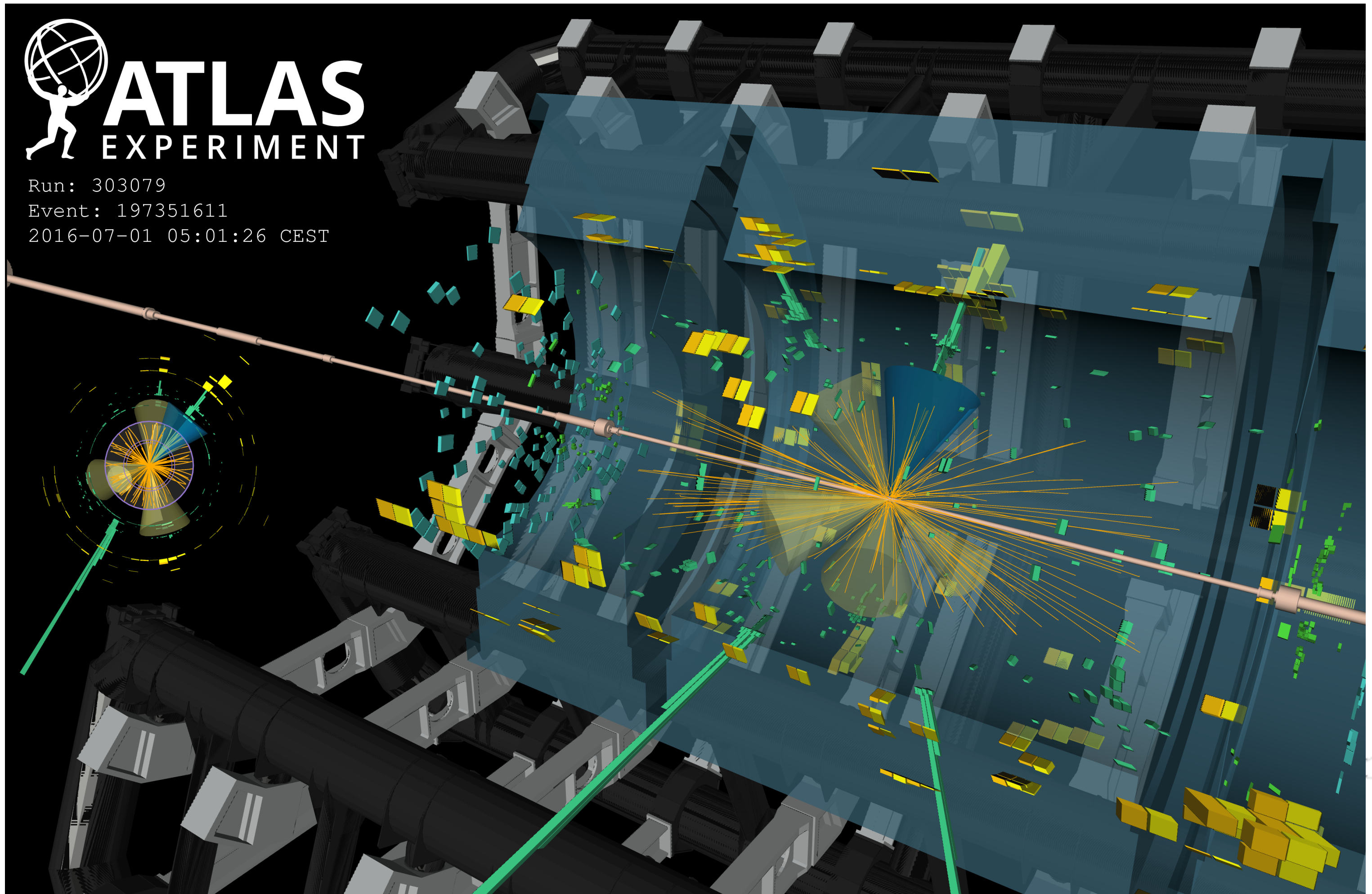


Quantum Machine Learning for High-Energy Physics

Michael Spannowsky

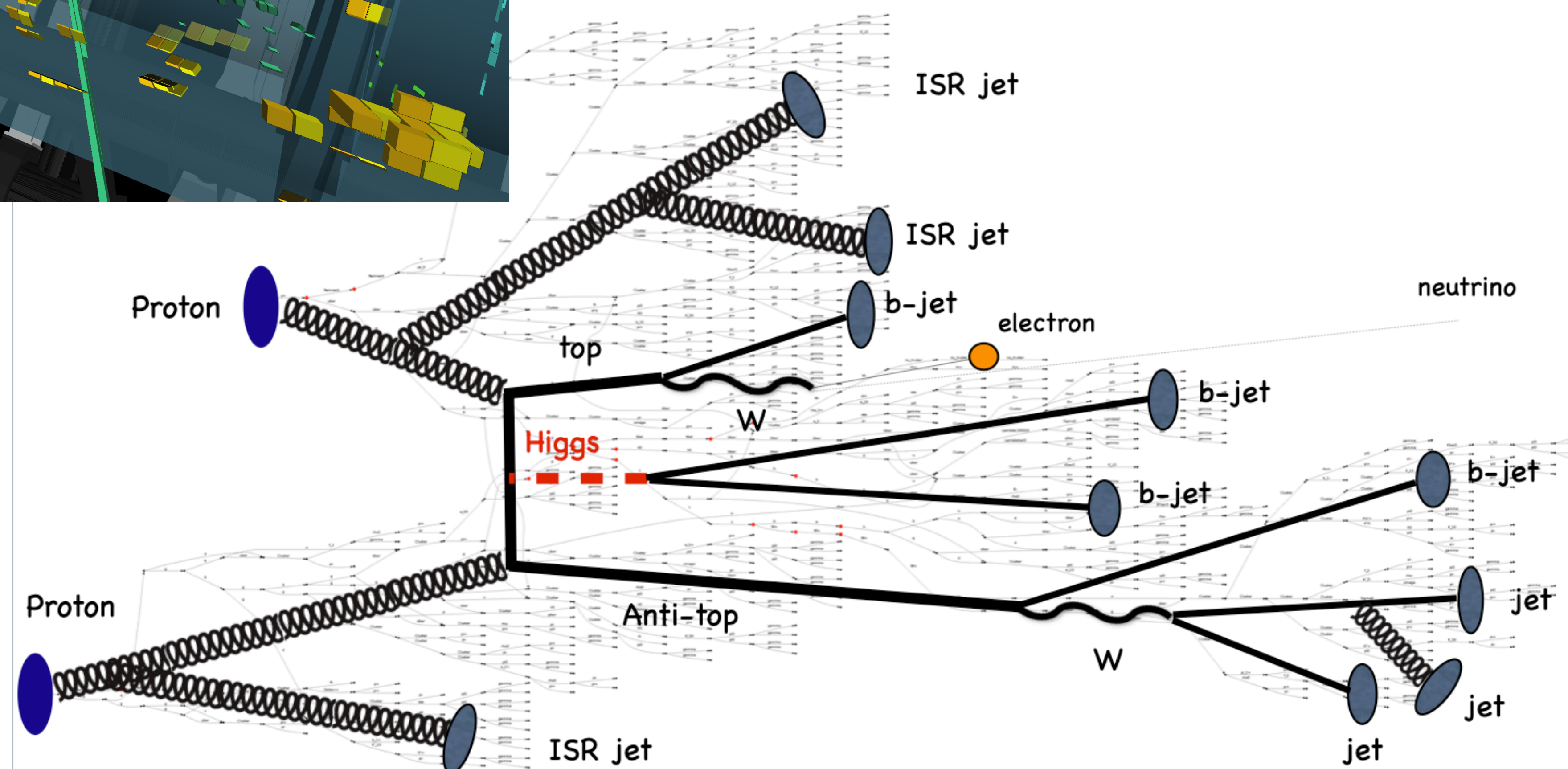
IPPP, Durham University

Complex events LHC



Candidate event tth ATLAS

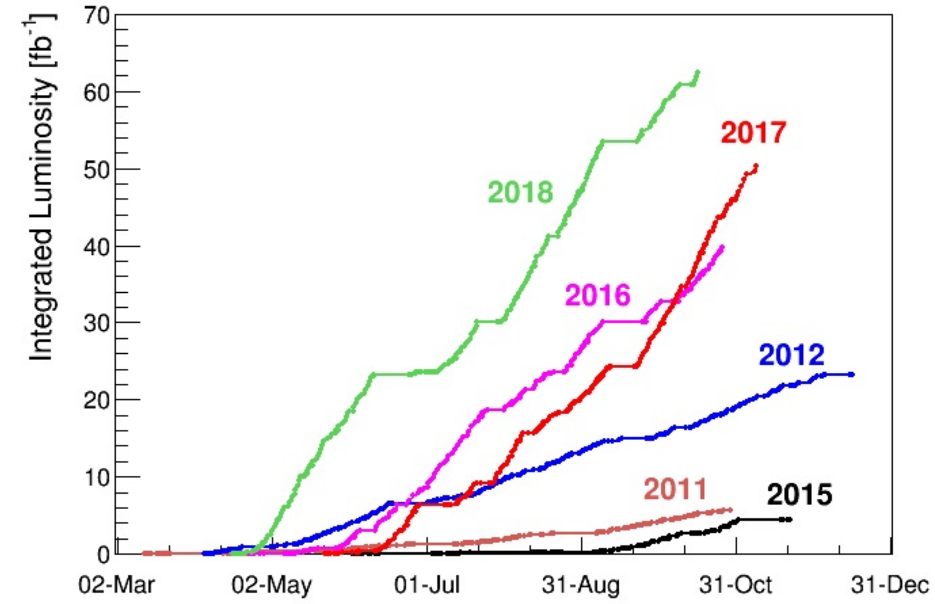
Theoretical picture of underlying dynamics



Big Data at the LHC

→ ATLAS/CMS 200 events/s passing triggers

→ ATLAS/CMS 2 PB/year of data

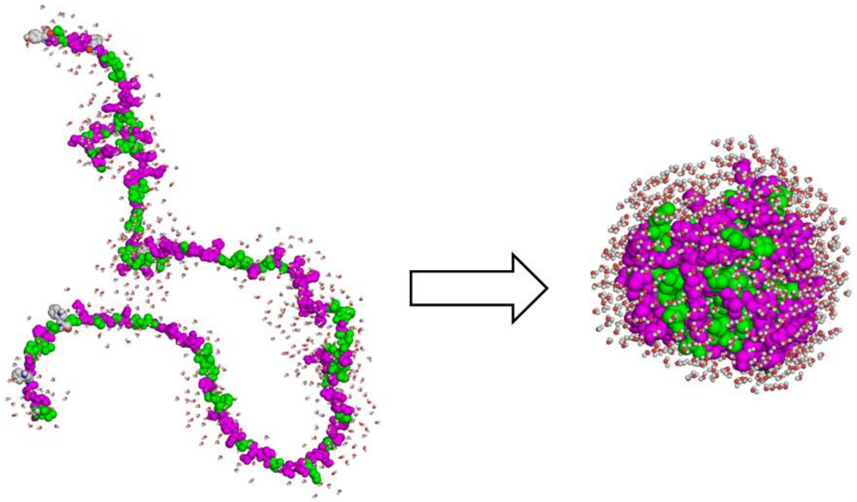


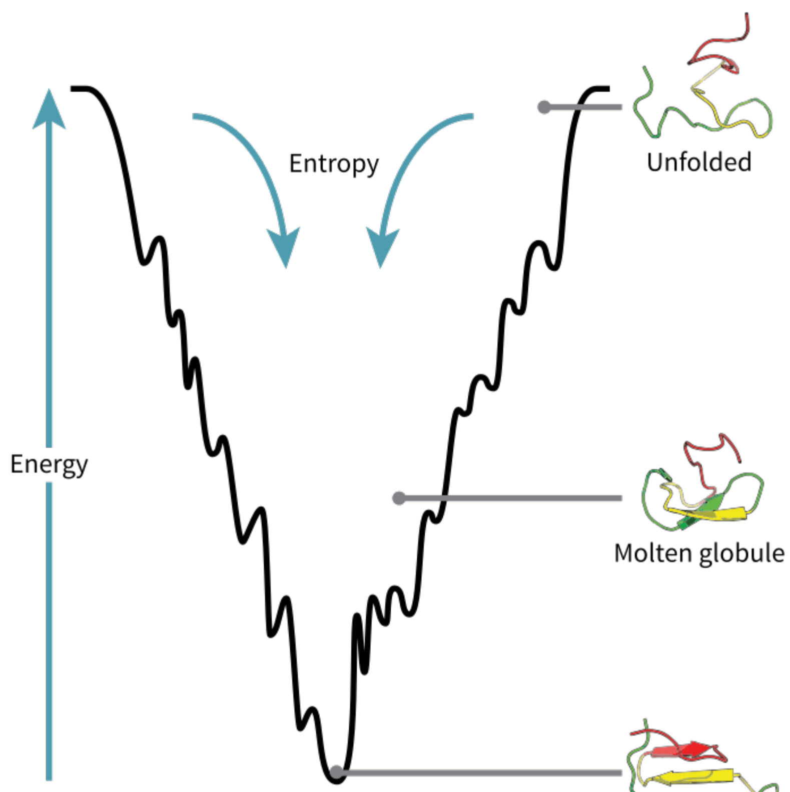
High-Energy Physics
Tremendous amount of highly complex data
However, theoretically very precise description of data

↔ Ideal interplay

Machine Learning
Highly performant data analysis techniques
Large number of techniques well suited to physical properties (RNN=time/ordering, CNN=space/orientation)

Protein-folding and Levinthal's Paradox





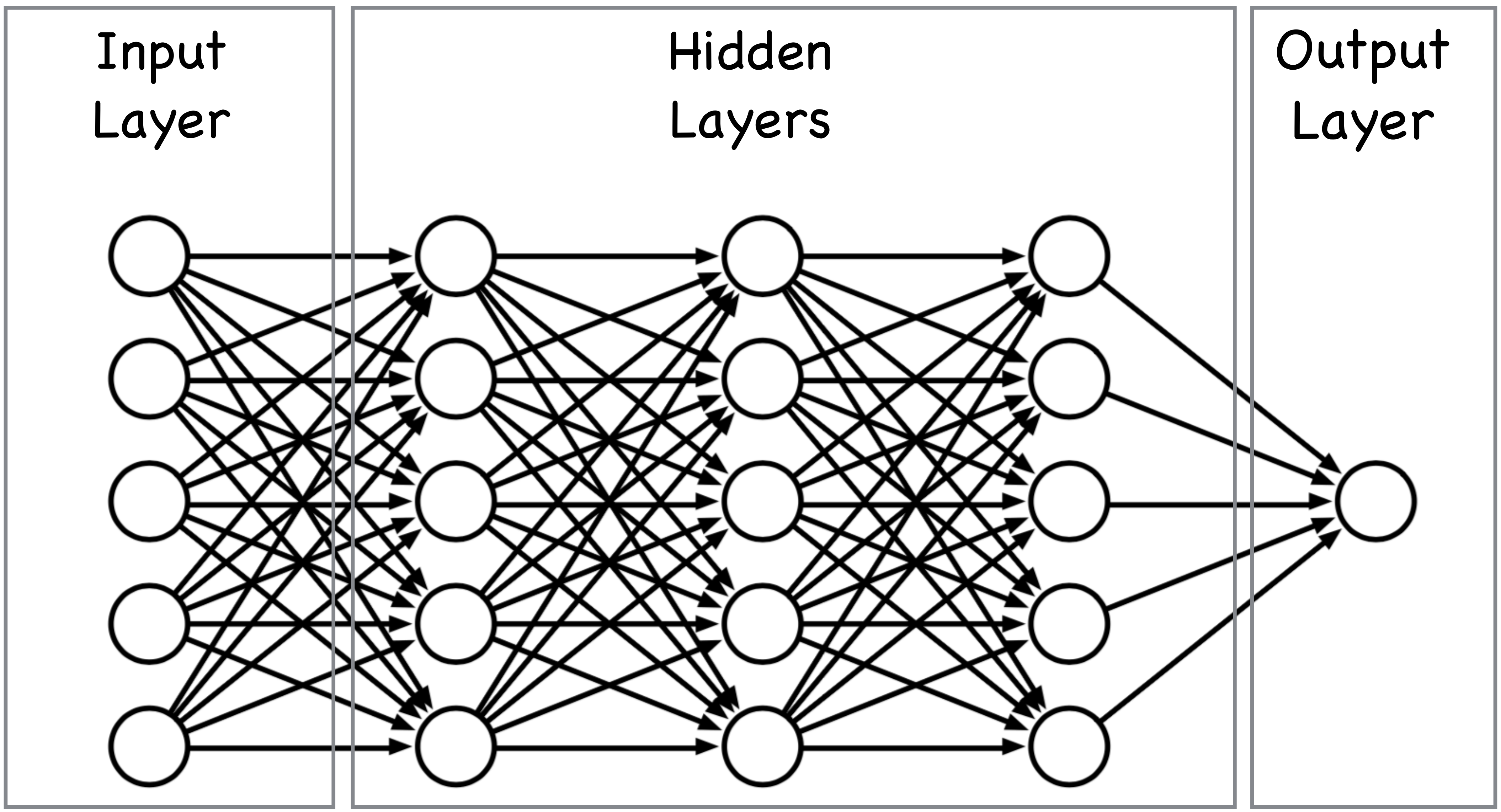
- Elongated proteins fold to same state within microseconds
- Some proteins have 3^{300} conformations
- Levinthal's Paradox (1969):
  Sequential sampling of states would take longer than lifetime of Universe (even if only nanoseconds per state spent)
- Solution: No sequential sampling, but rapid descend into the potential minimum. In proteins due to protein folding intermediates

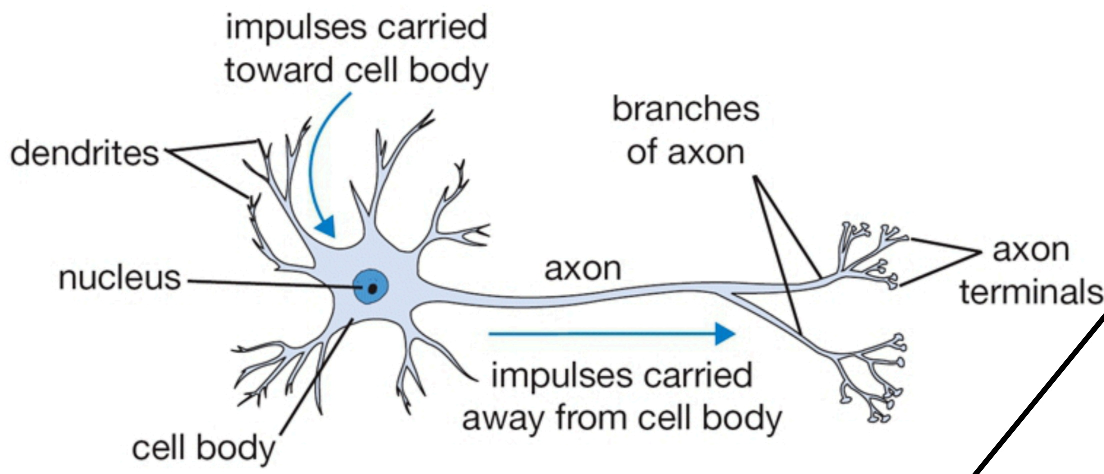
→ **Optimisation = Life**

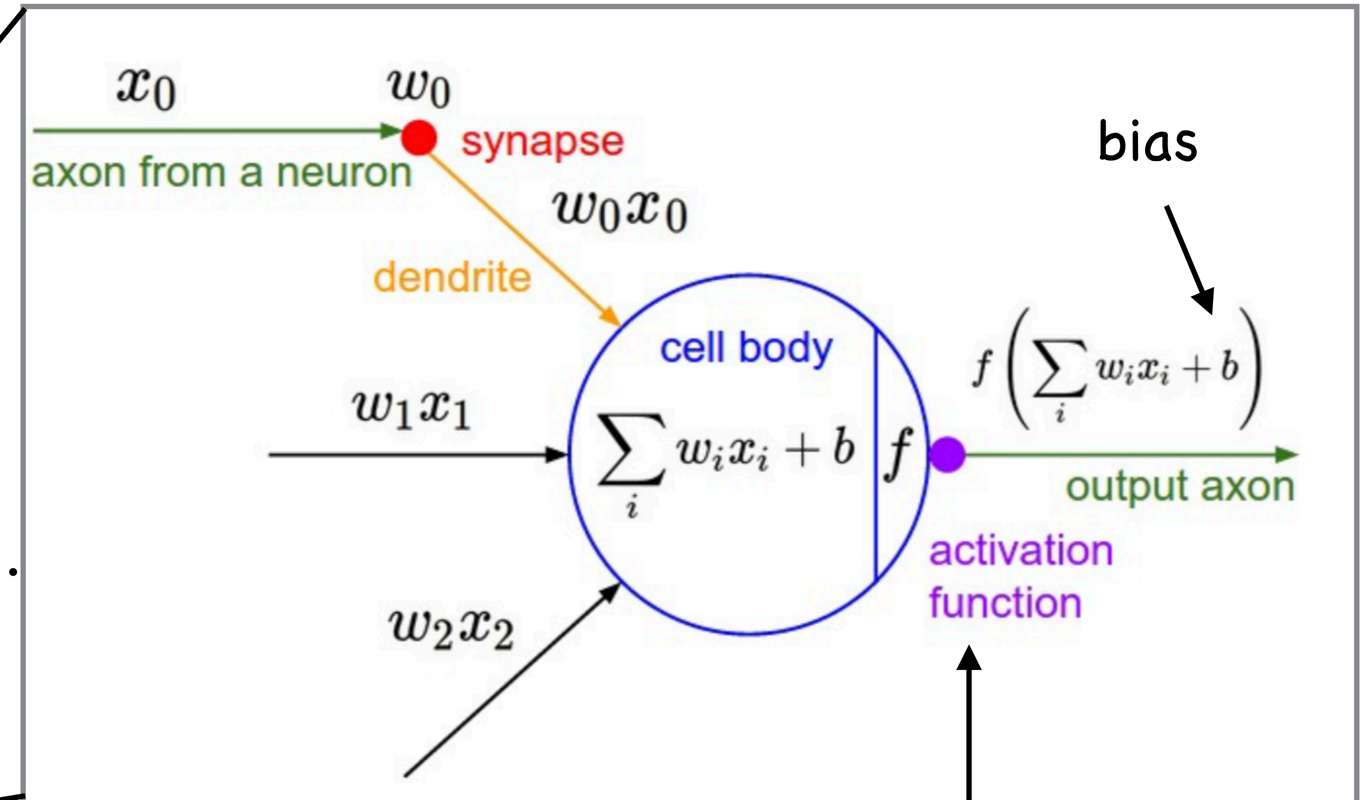
→ Solution of mathematical problem can be found quickly if encoded in ground state of complex system

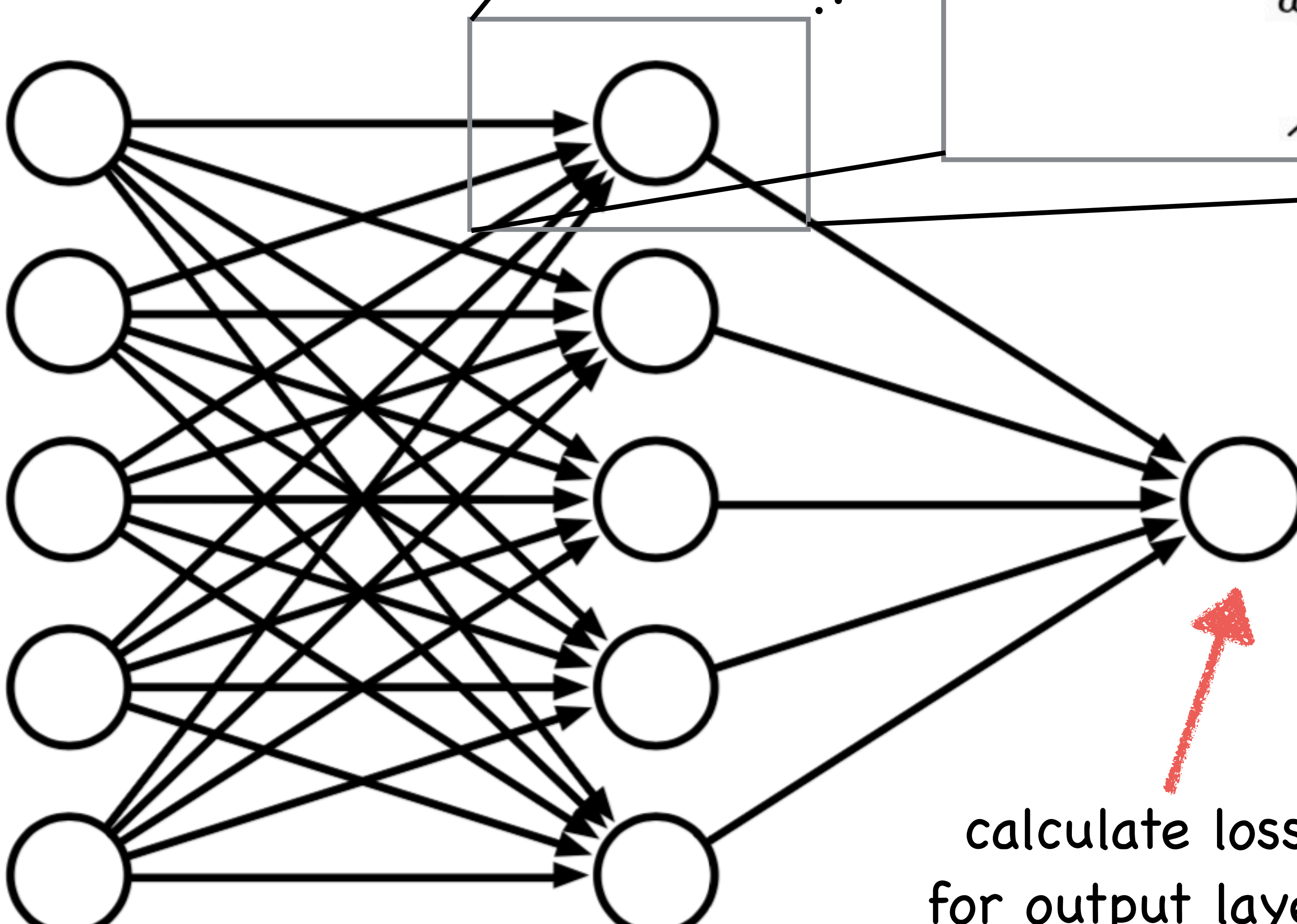
The classical (NN) approach

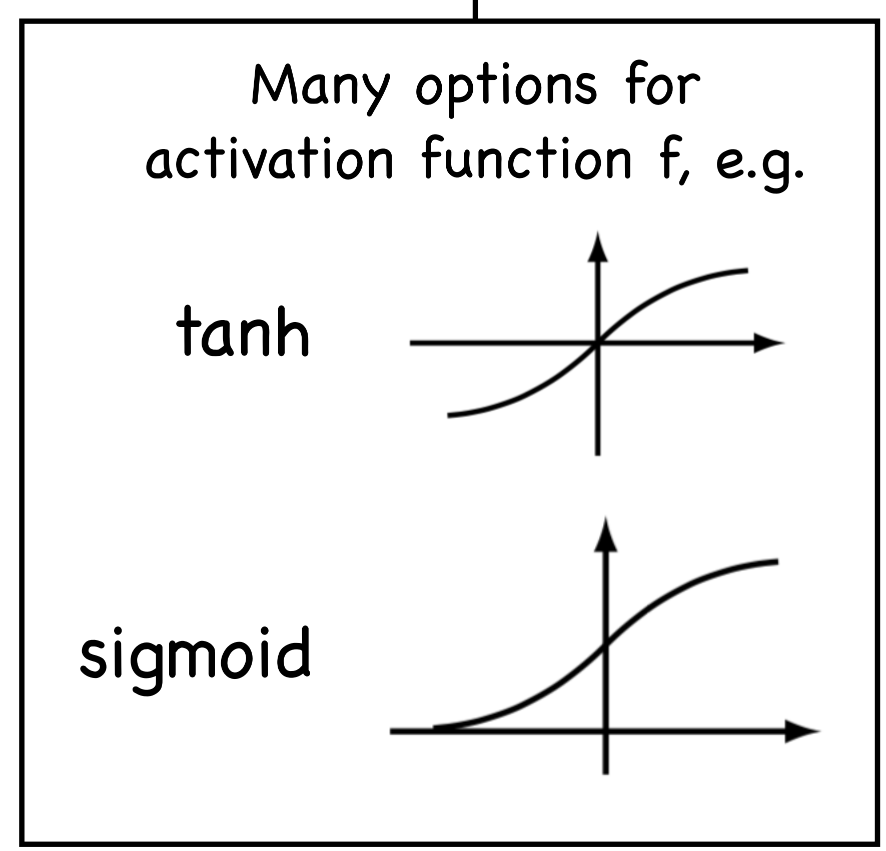
- Let's construct a complex system to optimise
- NN used in various ways. It is foremost a self-adaptive optimisation algorithm




impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body


x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$\sum_i w_i x_i + b$
f
$f\left(\sum_i w_i x_i + b\right)$
bias
$w_1 x_1$
output axon
activation function
$w_2 x_2$


calculate loss for output layer


Many options for activation function f, e.g.
tanh
sigmoid

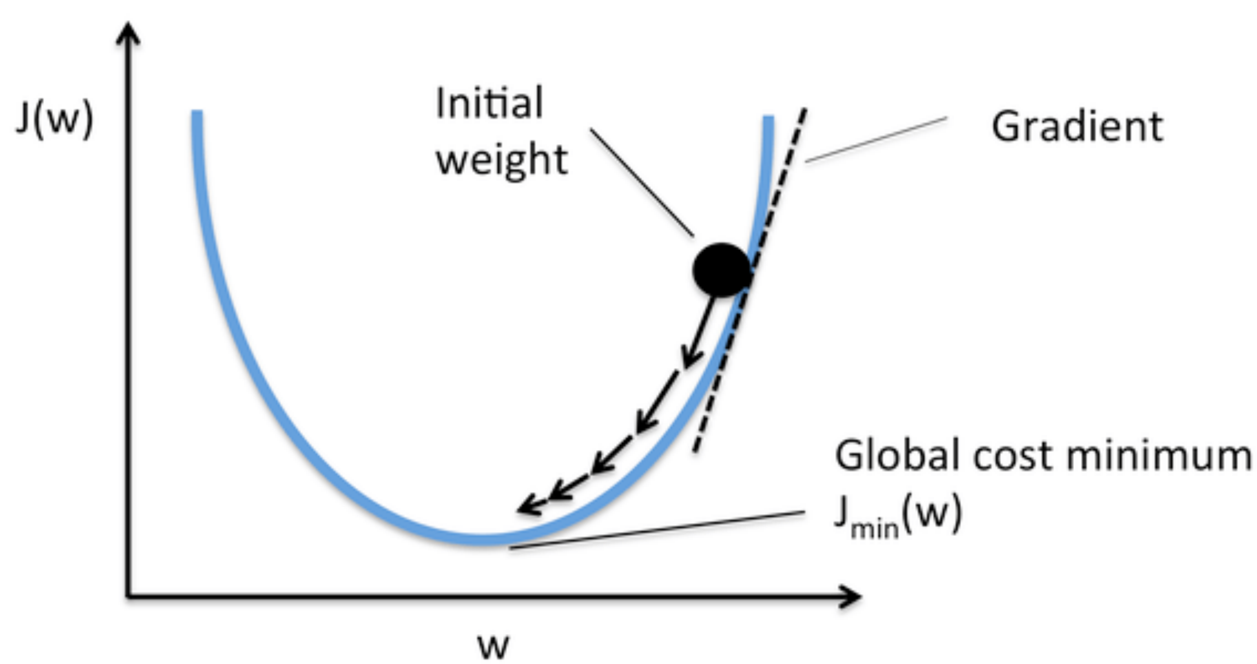
Gradient descent

- After forward propagation, ie. establishing the weights for all nodes (including the output node), we evaluate the **loss function**, to establish the error we are making.

Loss function = difference between predicted and true function,

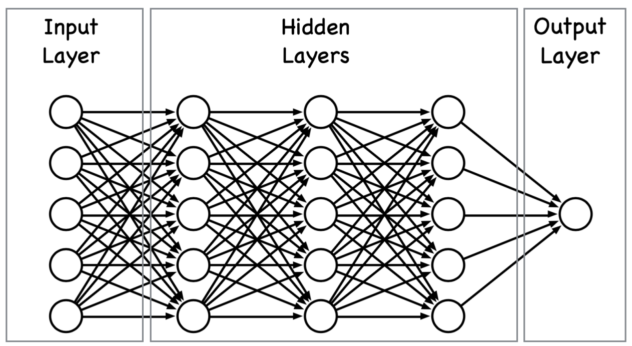
e.g. $E(y, y') = \frac{1}{2}|y - y'|^2$

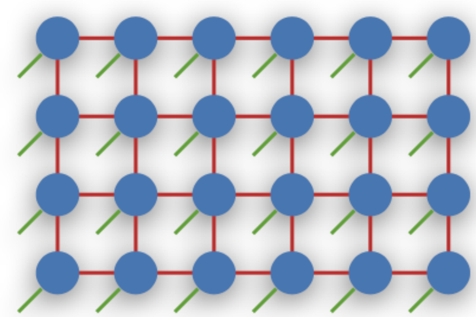
- Gradient descent: change network such, that you move towards the error minimum.
- Compute gradient -> get direction towards error minimum.

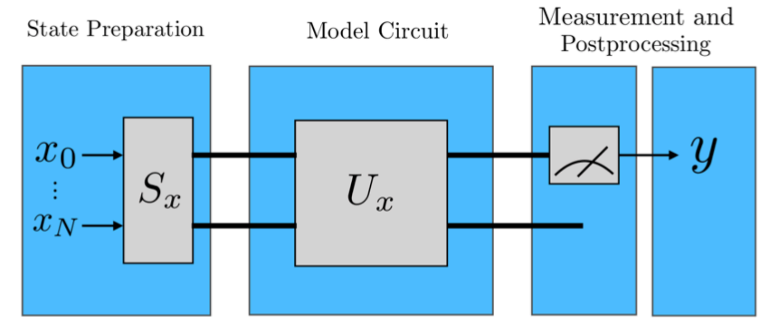


Classical ML Algorithms | Tensor Networks | Quantum Computing

1. an adaptable complex system that allows approximating a complicated function







2. the calculation of a loss function used to define the task the method

$$E(y,y') = \frac{1}{2}|y-y'|^2$$

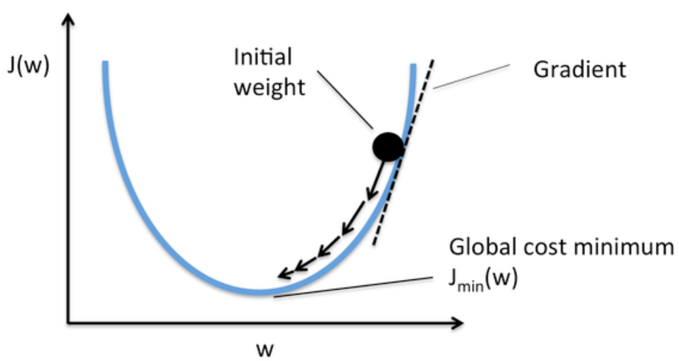
$$B^{s_2}{}_{p_1 p_2}\Gamma^{l p_1 p_2}{}_{s_2} = f^l(\mathbf{x}^{(\mathbf{n})})$$

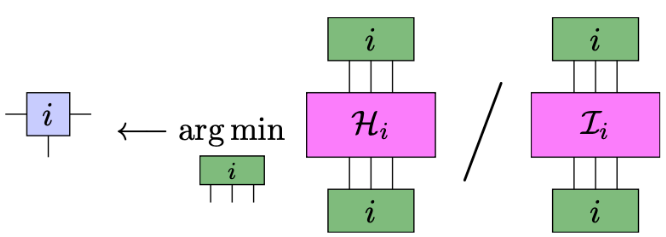
$$\mathcal{L} = L\left(p(l,\mathbf{x}),\ l^{truth}\right)$$

ground state

$$|\Gamma\rangle := \underset{|\psi\rangle\in\mathcal{D}}{\arg\min} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

3. a way to update 1. while minimising the loss function





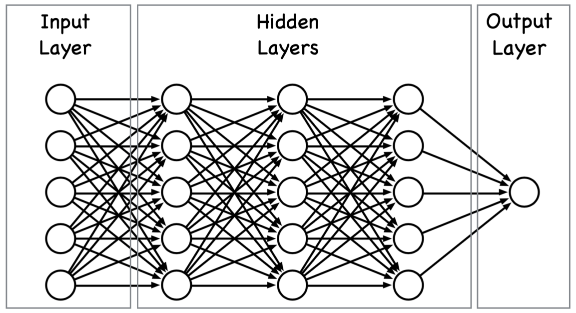
quantum: annealing

hybrid: classical opti.

optimisation → Data Analysis (Classification, anomaly, regression, fitting, ...)
→ Simulation of field theories (Groundstate, tunnelling, Real-time...)
→ Calculation of differential equations, etc etc

How can QNN be superior to NN

1. an adaptable complex system that allows approximating a complicated function



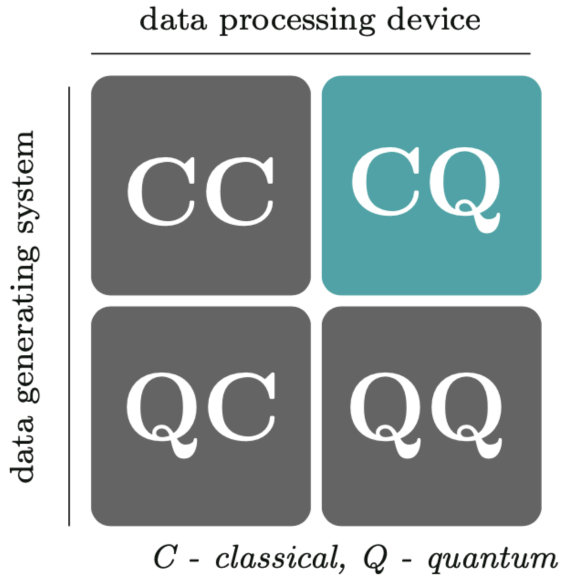
2. loss function

$$E(y, y') = \frac{1}{2}|y - y'|^2$$

- Input to QML can be quantum state

[Huang et al '21]

proven exponential advantage on noisy device over classical algorithm of any size



- QML more expressive

[Eisert, Cramer, Plenio '08] [Alcazar, Leyton-Ortega, Perdomo-Ortiz '20] [Araz, MS '22]

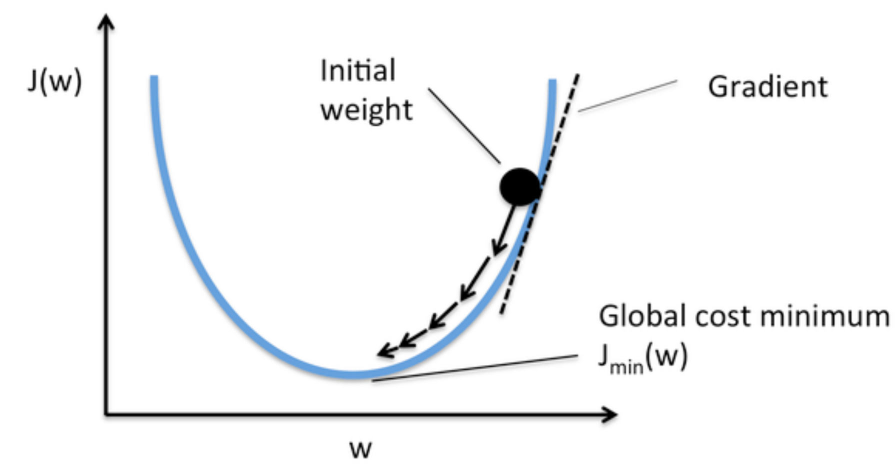
- Hybrid model possible - combination of classical and quantum nodes

- Exploit geometry of quantum loss function

⟶ Faster learning [Stokes, et al '20] [Blance, MS '20]

How can QNN be superior to NN

3. a way to update the network continuously while minimising the loss function, e.g. backpropagation



- Quantum sampling of loss function/energy function
- Faster learning, i.e. faster groundstate finding of loss function
- More reliable in finding the global minimum of the loss function

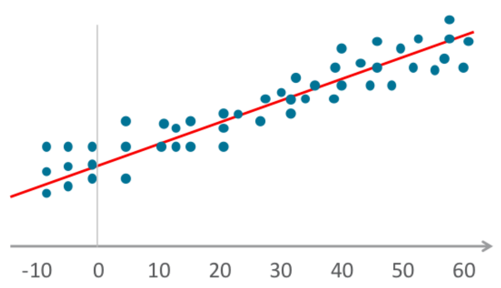
→ Potentially: Less sensitive to Barren Plateaus

Learns faster and from less data

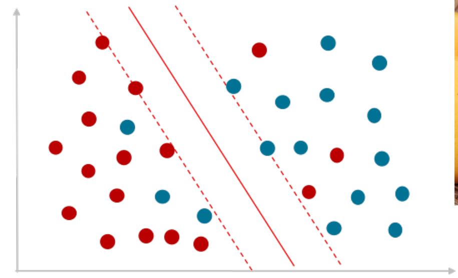
Doesn't get stuck in local minima
(less random in outcomes -> more interpretable)

Supervised

Regression



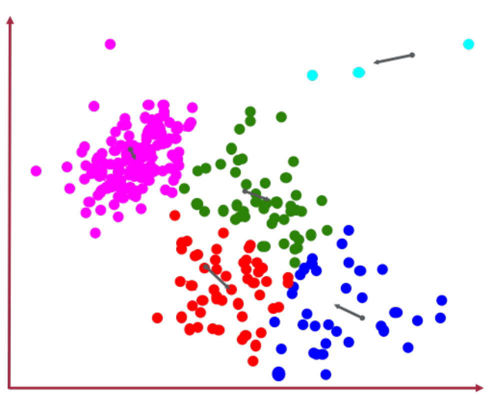
Classification



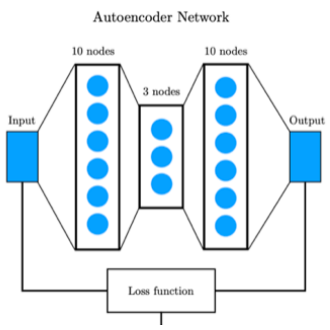
Fine-grained small net

Unsupervised

Clustering

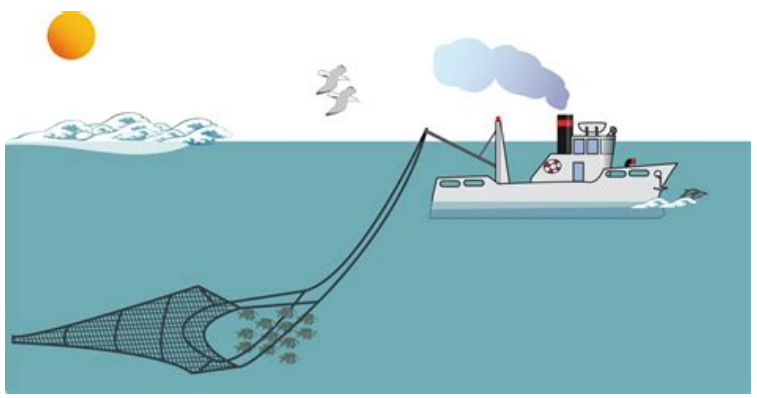


Autoencoder

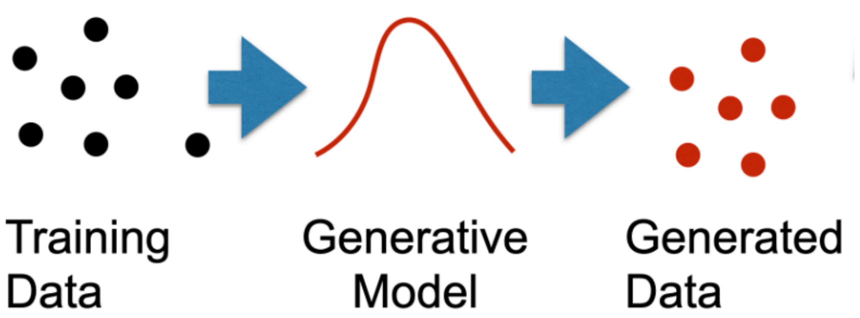


for quantum continuous variable algorithm see [Blance, MS '21]

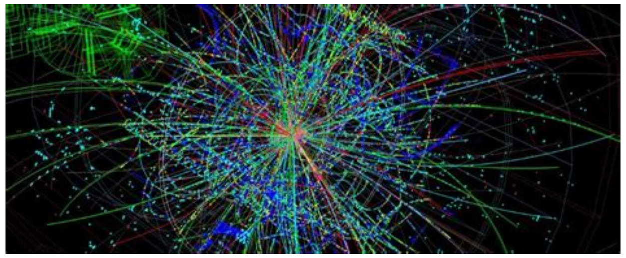
Large net



Generative



Pseudo-Data generation



Analog vs Digital Quantum Computing

Analog and digital quantum computing are two different approaches to quantum computing, each with its own advantages and disadvantages.

Analog Quantum Computing (AQC):

- Based on the principle of quantum evolution of a quantum system, e.g. quantum annealing
- The system uses its intrinsic quantum dynamics, following the Schroedinger Equation
- Ground state represents the solution to the problem at hand
- Not always universal, but often well-suited for optimisation problems

Example: D-Wave Systems. The D-Wave quantum annealer uses a network of qubits that can collectively tunnel through the solution space to find the global minimum of a given function.

Digital Quantum Computing (DQC):

- Digital quantum computing, also known as gate-based quantum computing
- Uses quantum logic gates to perform operations on qubits
- Considered to be more versatile than analog computing.
- However, might require higher level of control over the quantum system, which can be challenging

Example: IBM's and Google's quantum computers use the gate-based model of quantum computing.

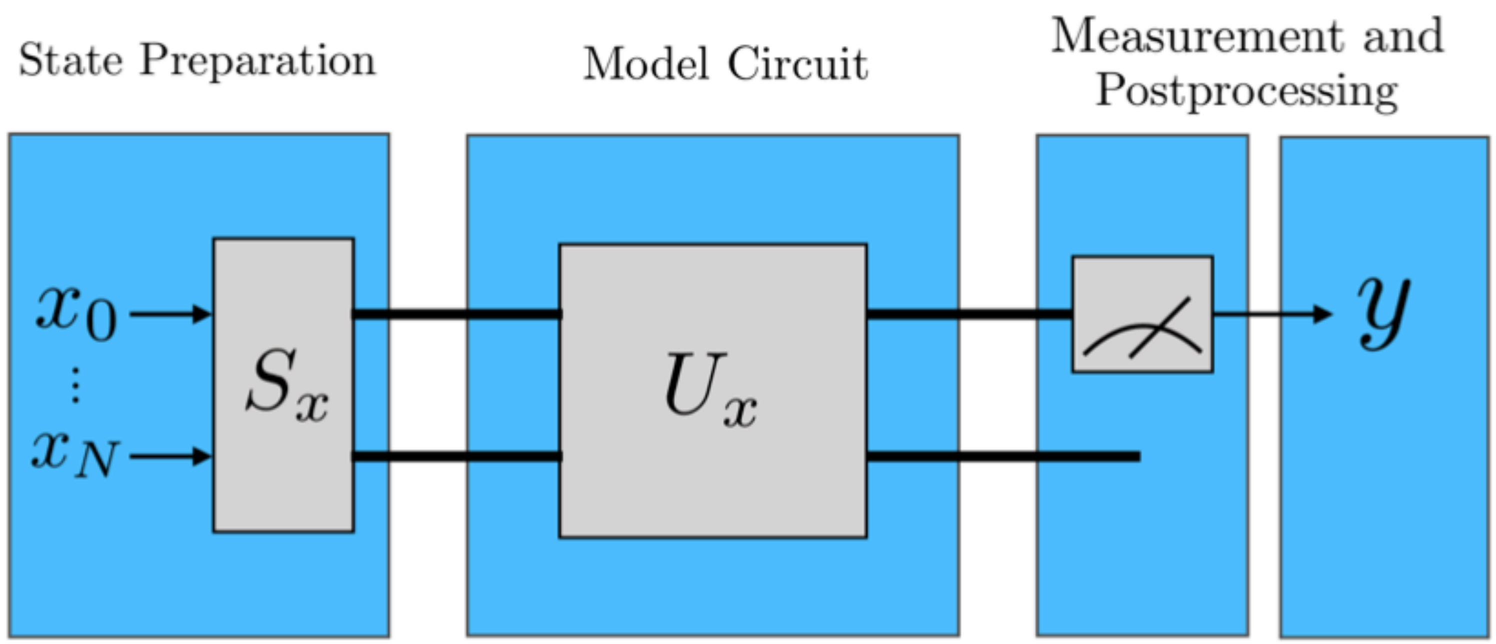
Quantum Machine Learning
with a Variational Quantum Circuit

[McClean et al '16]

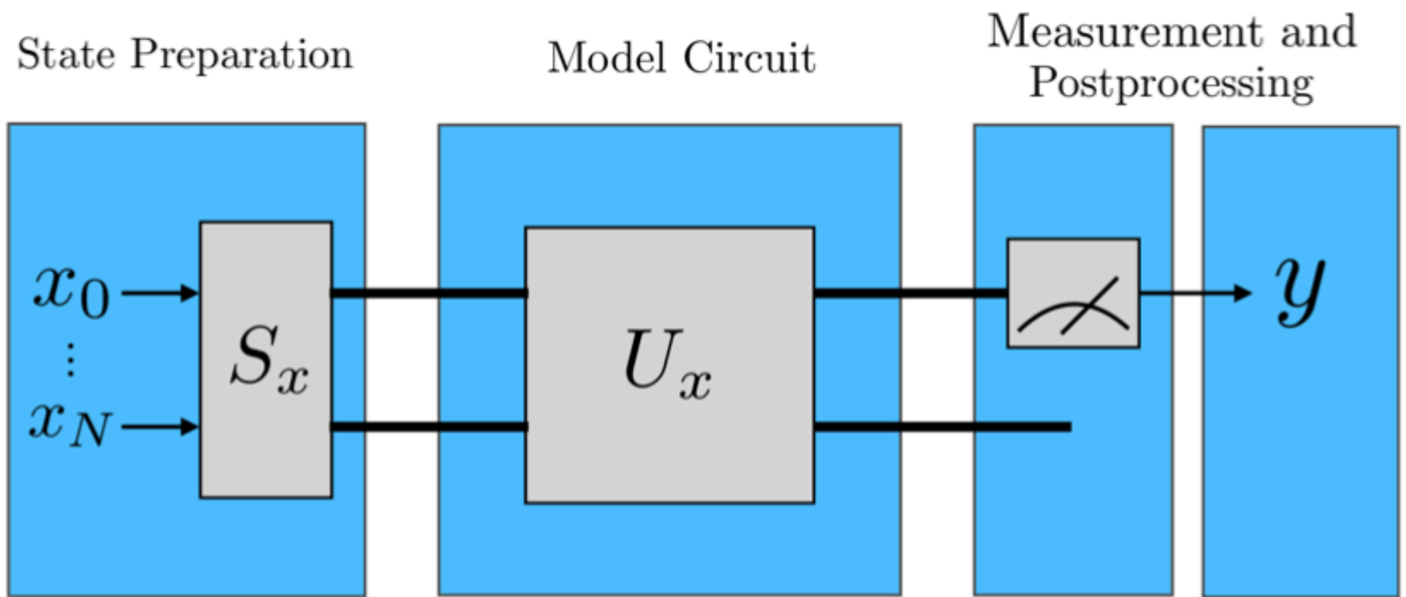
[Farhi, Neven '18]

[Schuld et al '20]

[Blance, MS '20]



Quantum Machine Learning with a Variational Quantum Circuit



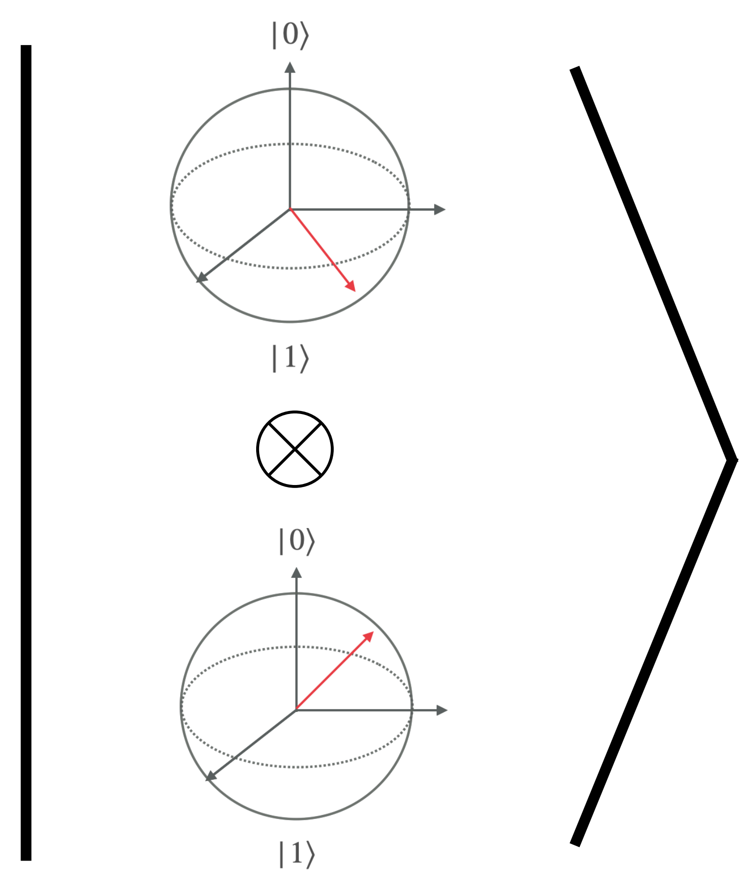
state preparation

n corresponds to # features

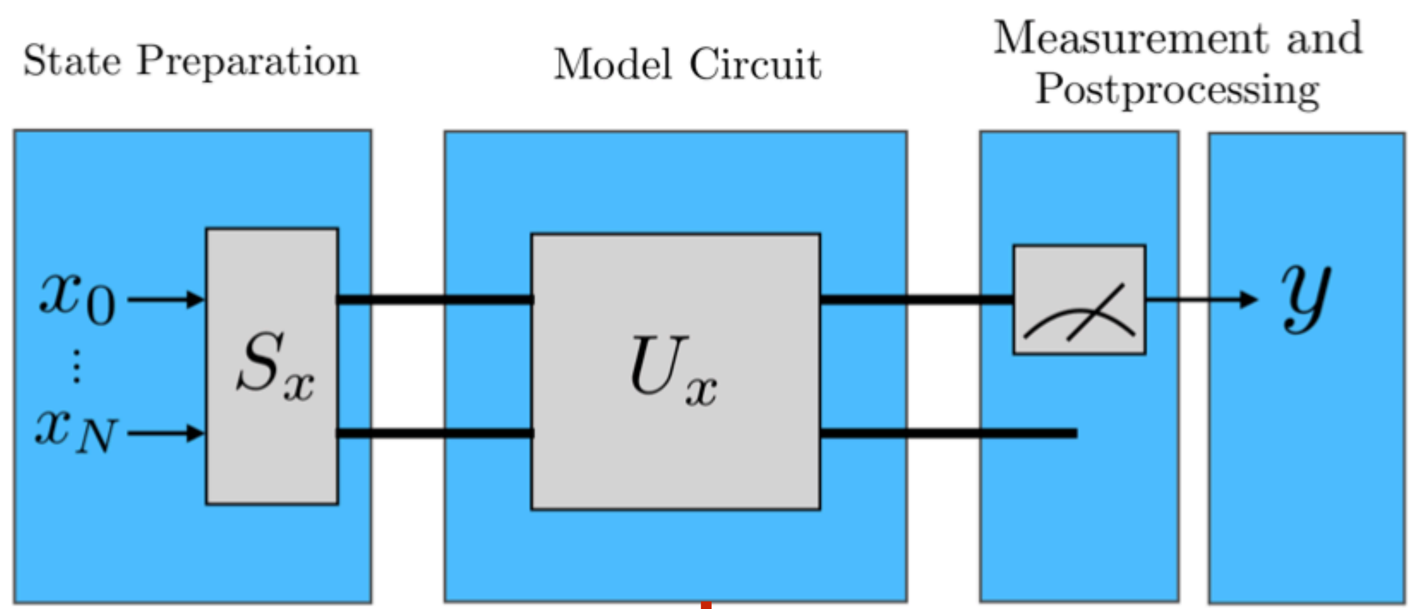
$$x \mapsto S_x|\phi\rangle = S_x|0\rangle^{\otimes n} = |x\rangle$$

e.g. angle encoding

$$|x\rangle = \bigotimes_{i=1}^{n} \cos(x_i)|0\rangle + \sin(x_i)|1\rangle$$



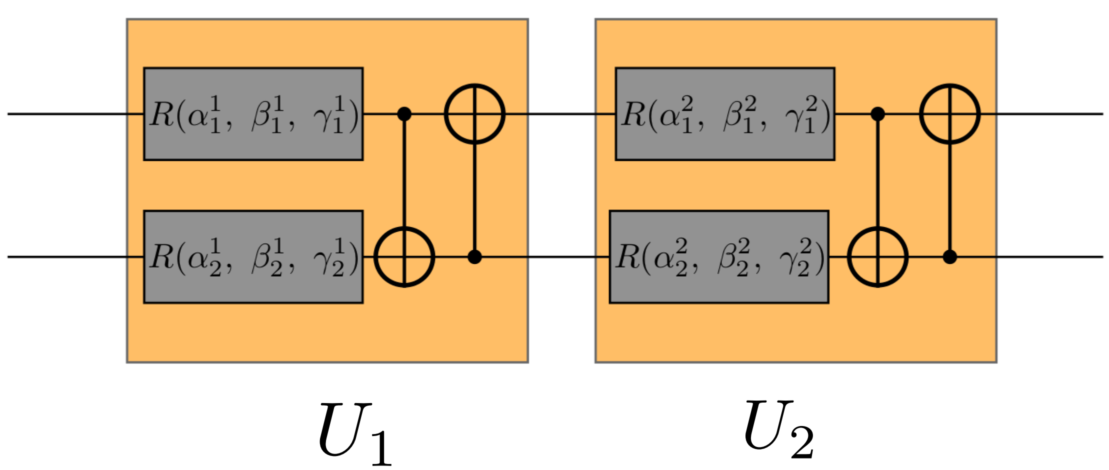
Quantum Machine Learning with a Variational Quantum Circuit

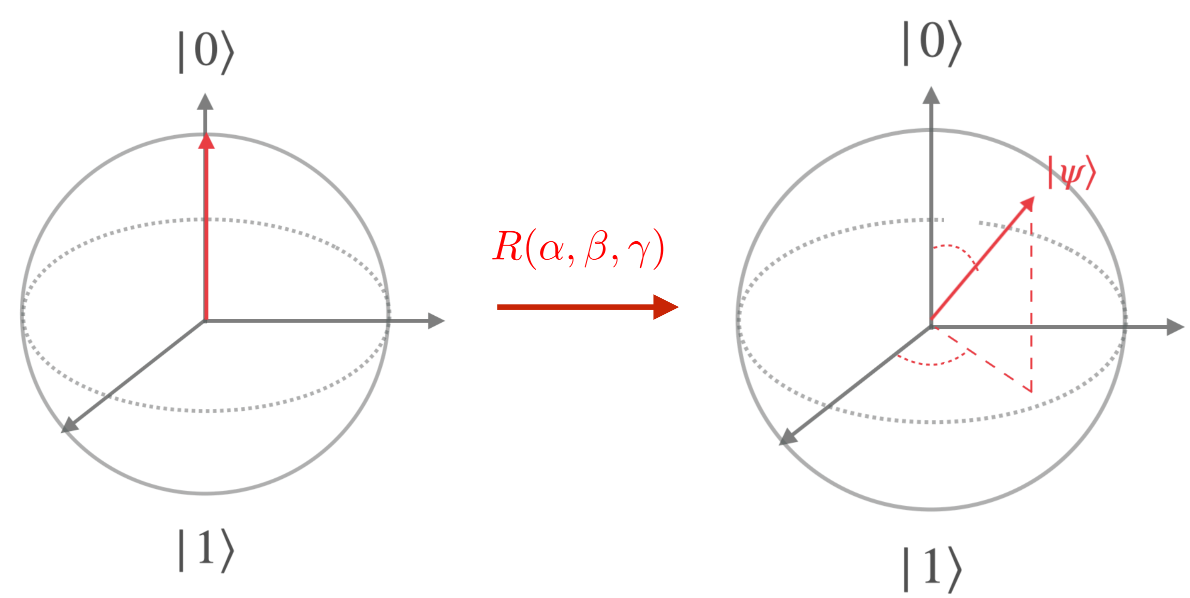


$$|\psi\rangle = U(w)|x\rangle \quad \text{with} \quad U(w) = U_{l_{\max}}(w_{l_{\max}})...U_l(w_l)...U_1(w_1)$$

model circuit — trainable parameters — prepared state

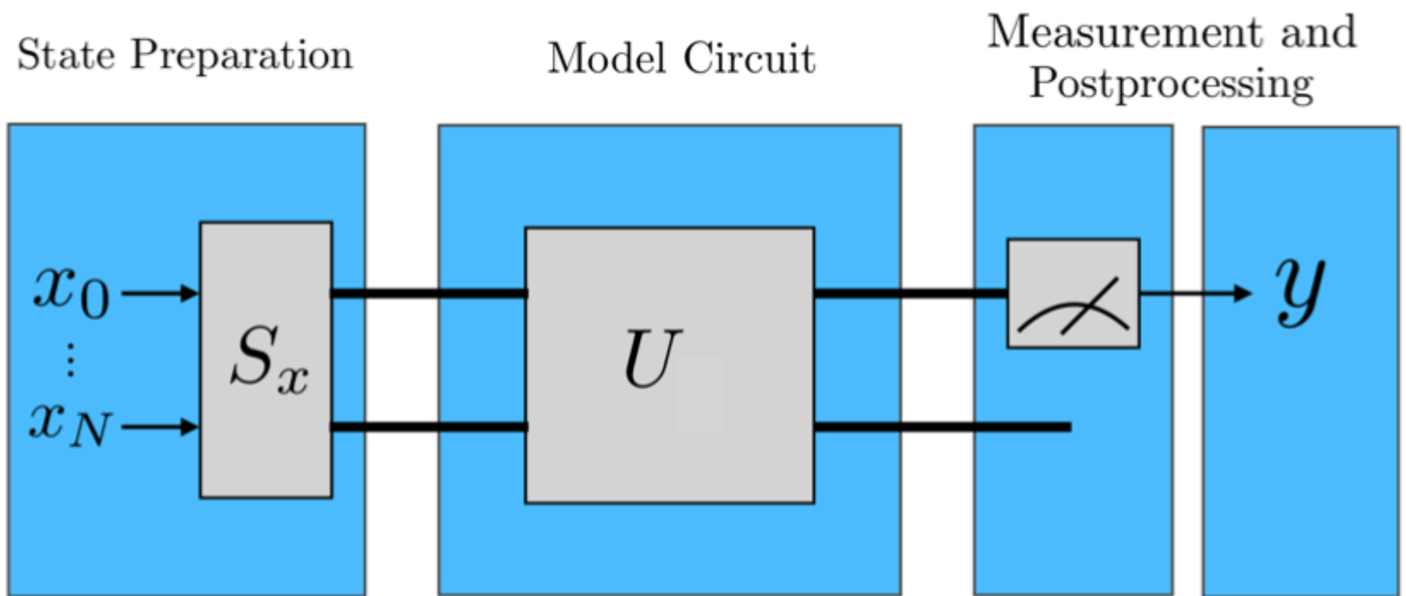
2-layer Variational Quantum Circuit





→ Rotation + CNOT -> Entanglement

Quantum Machine Learning with a Variational Quantum Circuit



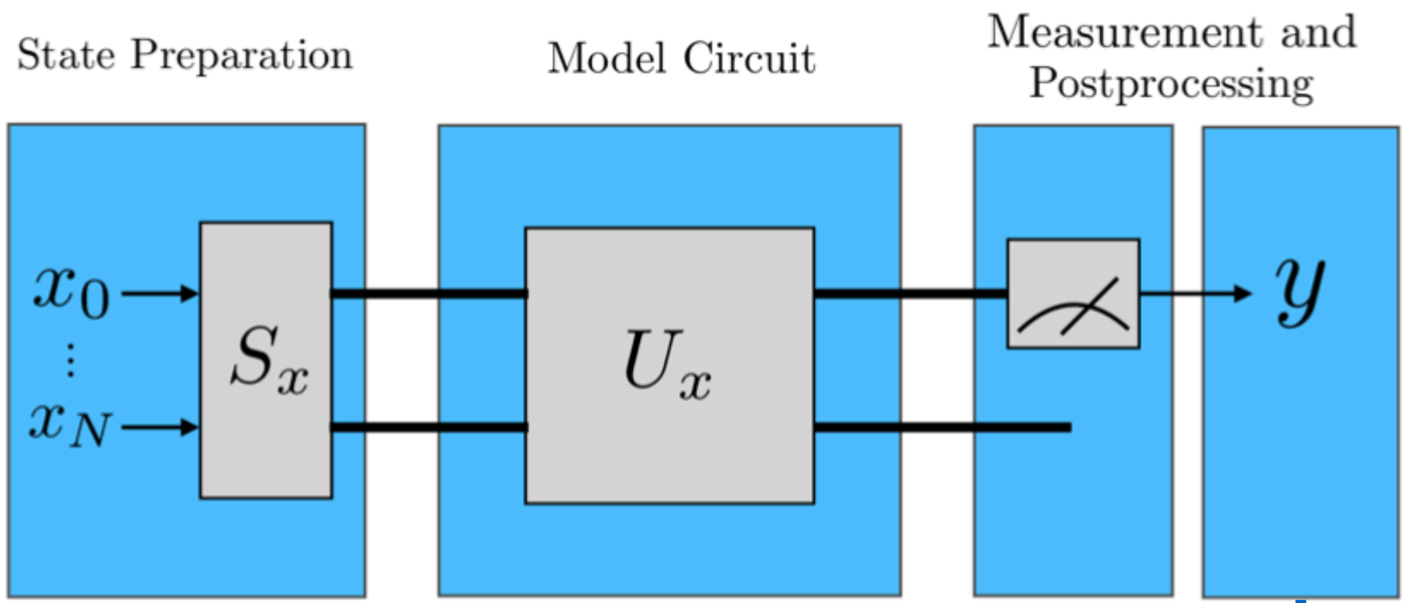
- Entangled state shares information across qubits
- Evaluate expectation value of qubits to construct loss

  for supervised S vs B classification one qubit sufficient

$$\mathbb{E}(\sigma_z) = \langle 0 | S_x(x)^\dagger U(w)^\dagger \hat{O} U(w) S_x(x) | 0 \rangle = \pi(w,x) \quad \text{for} \quad \hat{O} = \sigma_z \otimes \mathbb{I}^{\otimes (n-1)}$$

- Quantum network output: $f(w,b,x) = \pi(w,x) + b$
- Changing operator and loss => VQE, VQT, ... (simulate QFT)

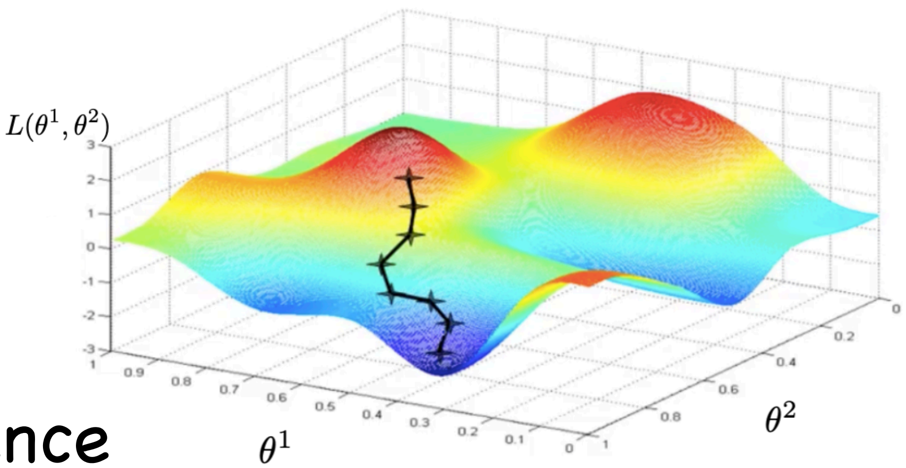
Quantum Machine Learning with a Variational Quantum Circuit



- Hybrid approach (QC to calculate exp. value, CC to optimise U operator)

- Loss function $$L = \frac{1}{n}\sum_{i=1}^{n}\left[y_i^{\text{truth}} - f(w,b,x_i)\right]^2$$

  label (signal, bkg), supervised learning

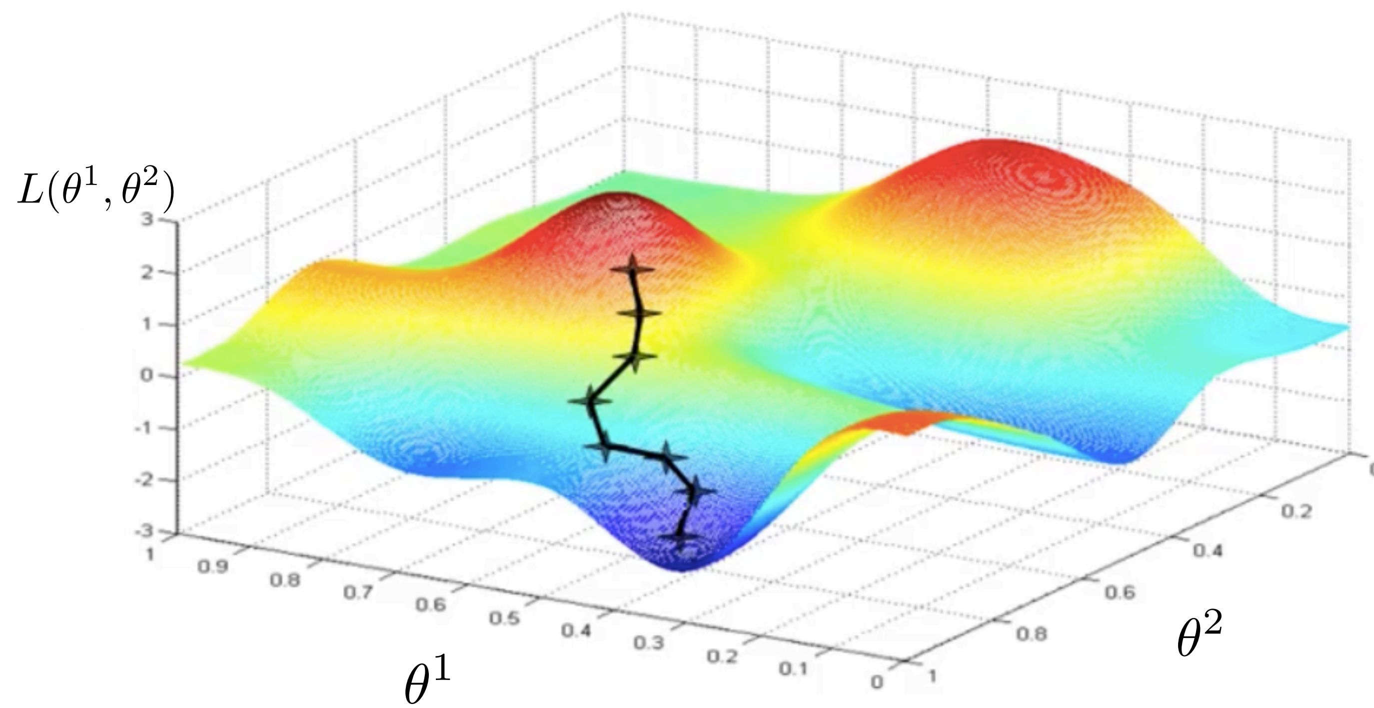


- Quantum gradient descent - for fast convergence

  Fubiny-Study metric underlies geometric structure of VQC parameter space: $\theta_{t+1} = \theta_t - \eta g^{+} \nabla L(\theta)$

[Cheng '10]
[Blance, MS '20]
[Abbas et al '20]

Optimising the loss landscape



classical gradient descent (GD):

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta)$$

quantum gradient descent (QDC):

Fisher Information Matrix F promotes
gradient descent to natural gradient descent (Riemannian geometry):

[Stokes, Izaac, Killoran, Carleo '20] [Blance, MS '20]

$$\theta_{t+1} = \theta_t - \eta F^{-1} \nabla L(\theta)$$

Fubiny-Study metric underlies geometric structure of VQC parameter space (complex projective Hilbert Spaces):

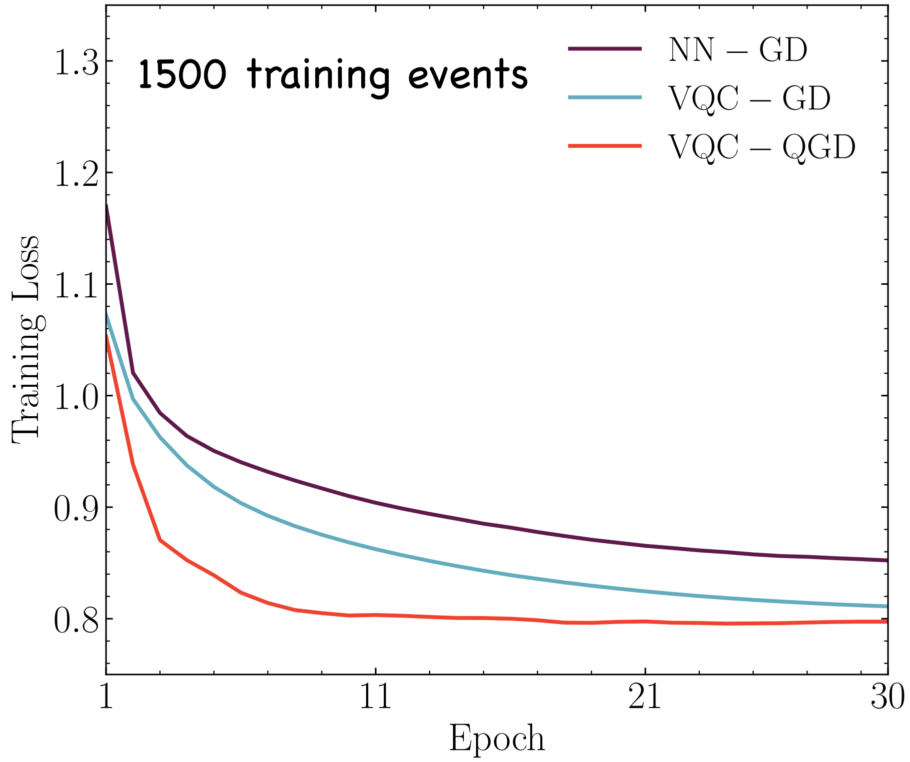
$$\theta_{t+1} = \theta_t - \eta g^{+} \nabla L(\theta)$$

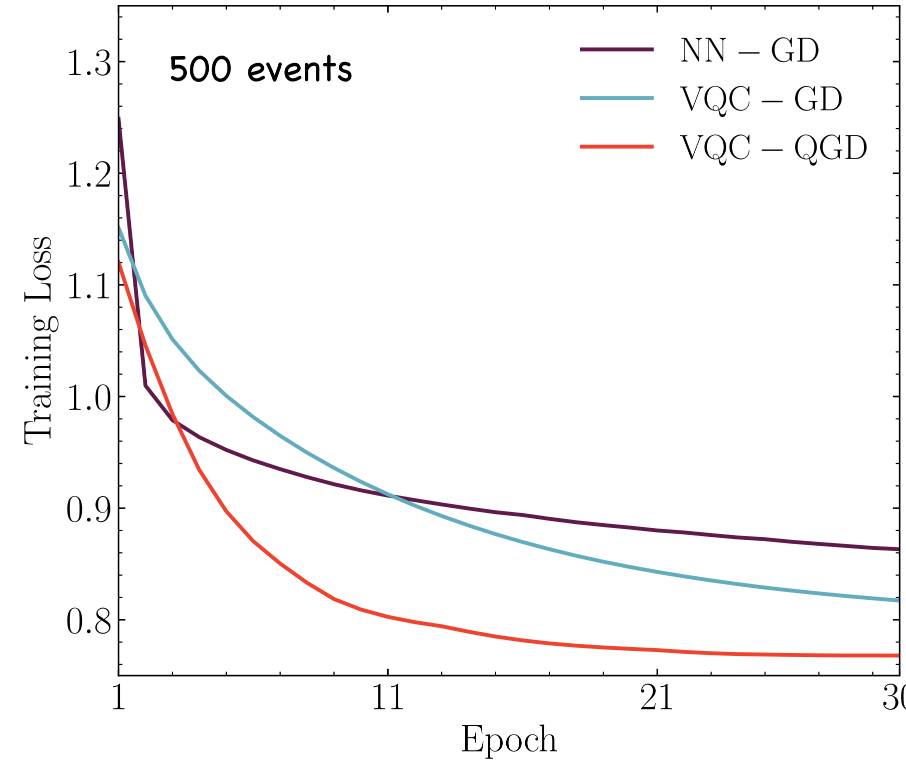
VQC parameters

weights $\theta^w_{t+1} = \theta^w_t - \eta g^{+} \nabla^w L(\theta)$,

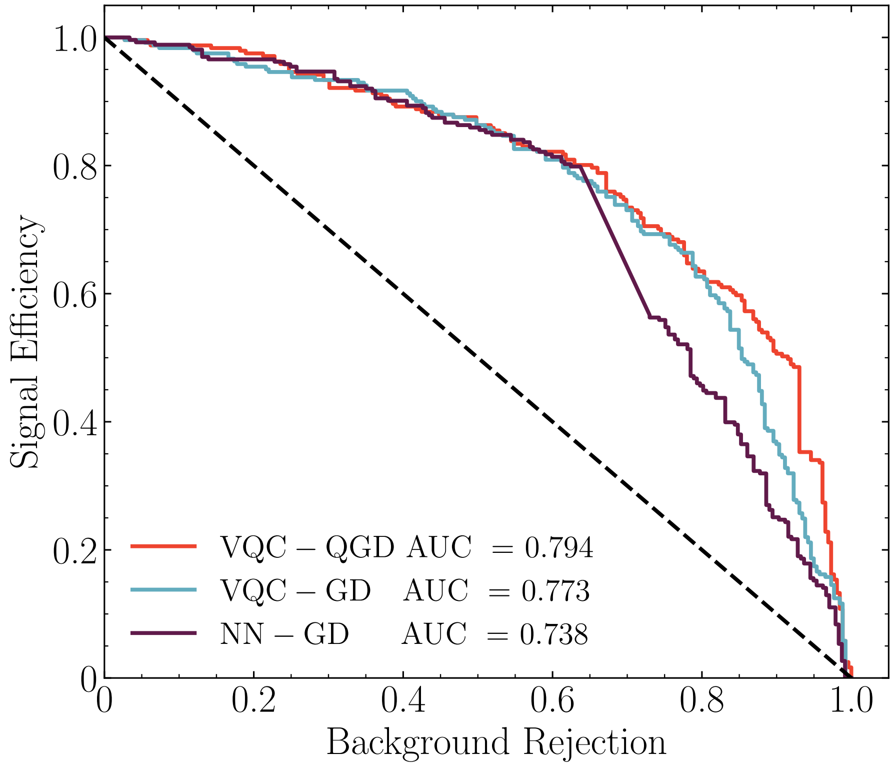
bias $\theta^b_{t+1} = \theta^b_t - \eta \nabla^b L(\theta)$,

Gate quantum machine learning in action





[Blance, MS '20]



QC device vs simulator

Device	Accuracy (%)
PennyLane default.qubit	72.6
ibmq_qasm_simulator	72.6
ibmqx2	71.4

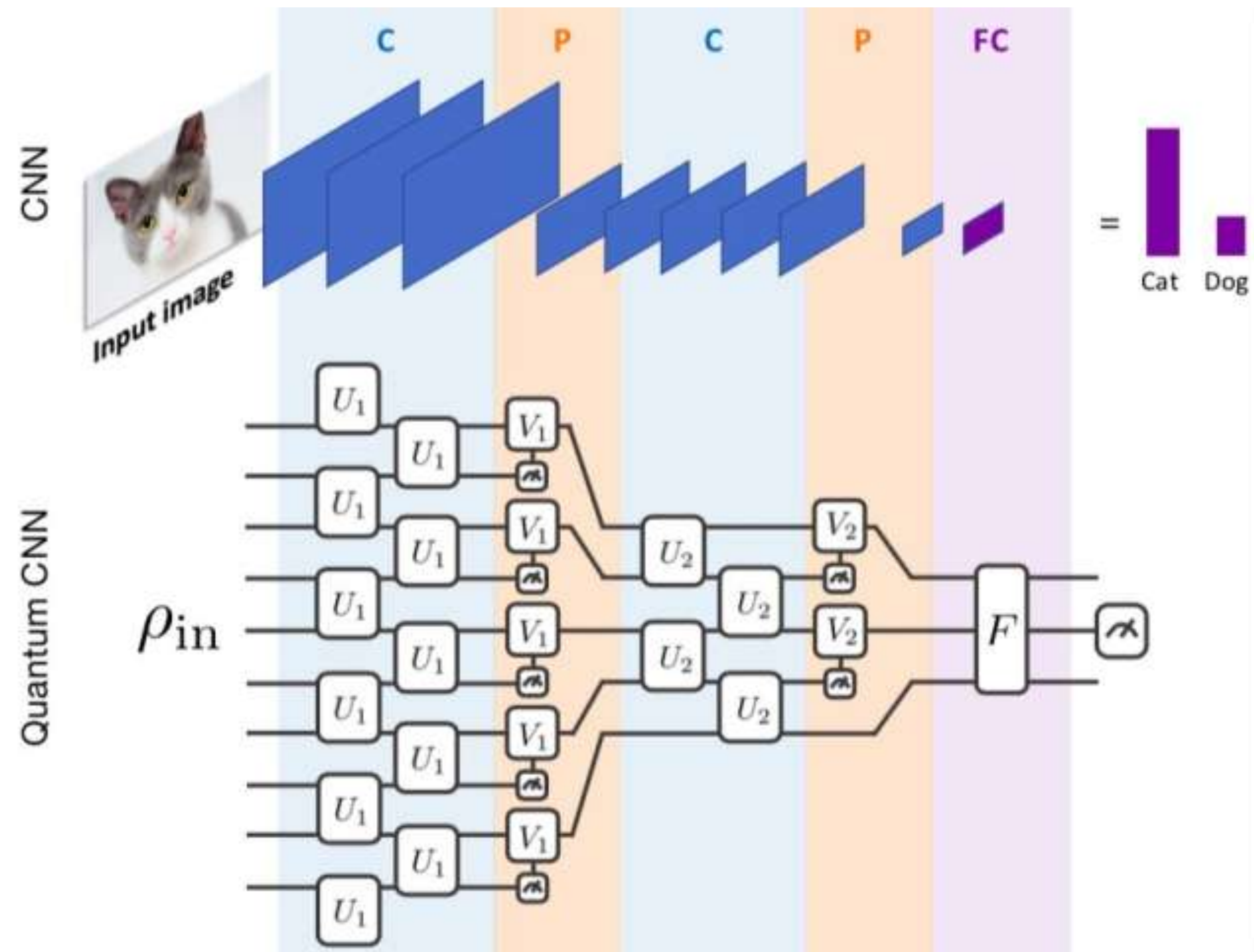
- Applied to $pp \to t\bar{t}$ vs $pp \to Z' \to t\bar{t}$ lept. top dec for 2d feature space only p_{T,b_1} and $\not{E}_T$

Quantum Convolutional Neural Network

- Convolutional layers work by sweeping across the input array and applying different filters (often 2x2 or 3x3 matrices) block by block. Used to detect specific features of the image wherever they might appear.

- Pooling layers used to downsample results of these convolutions to extract most relevant features and reduce the size of the data. Common pooling methods involve replacing blocks of the data with their maximum or average values.

- QCNN uses only O(log(N)) variational parameters for input size of N qubits



conv layer
pooling layer
Fully connected layer

[Cong, Choi, Lukin '19]

Some results

Input to QCNN: ground state wave function for Hamiltonian

$$\hat{H} = -J\sum_{i=1}^{n-2}(\hat{Z}_i\hat{X}_{i+1}\hat{Z}_{i+2}) - h_1\sum_{i=1}^{n}(\hat{X}_i) - h_2\sum_{i=1}^{n-1}(\hat{X}_i\hat{X}_{i+1})$$

with couplings (J, h1, h2)

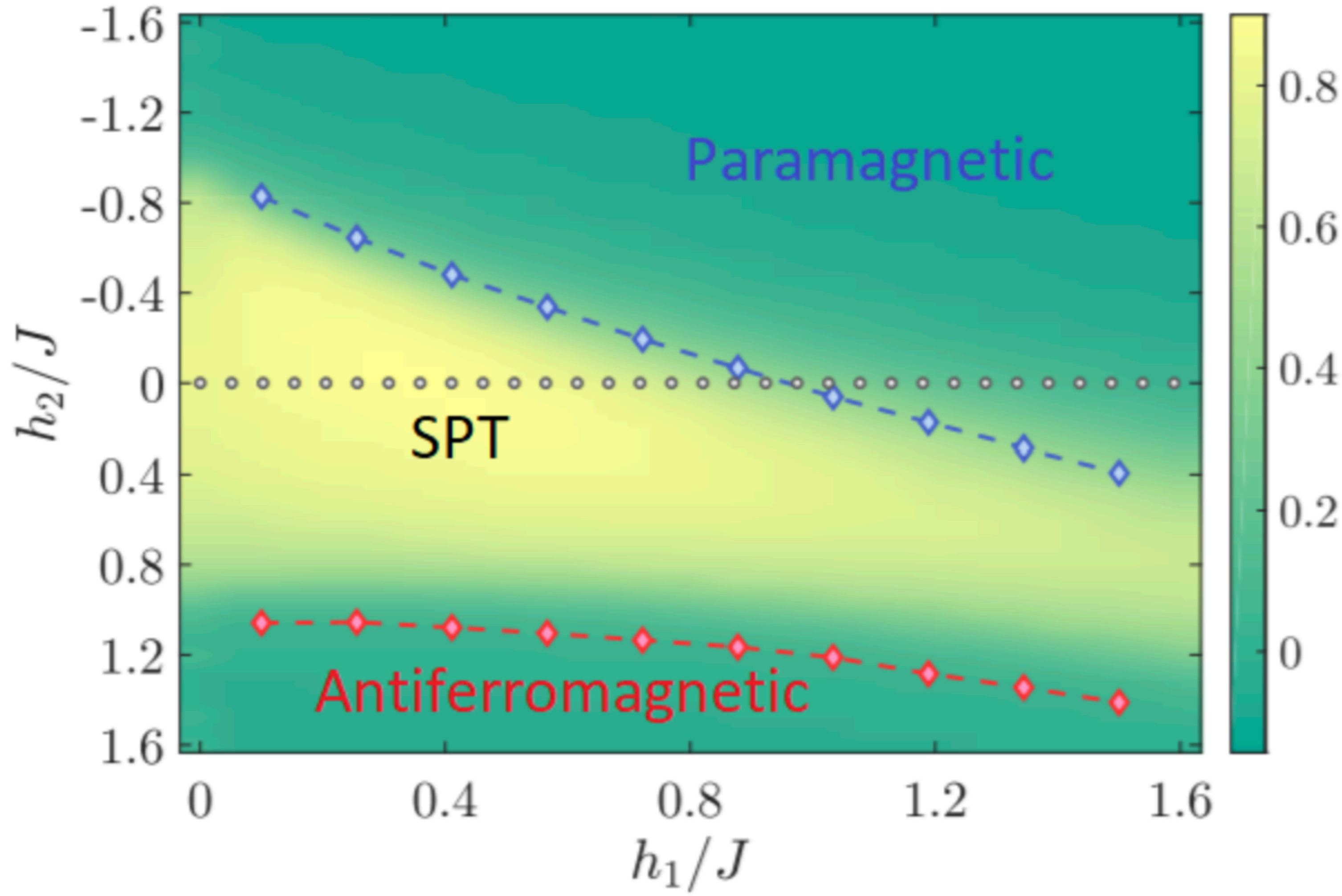
[Cong, Choi, Lukin '19]

[Nagano, et al '23]

Grey dots, training data set

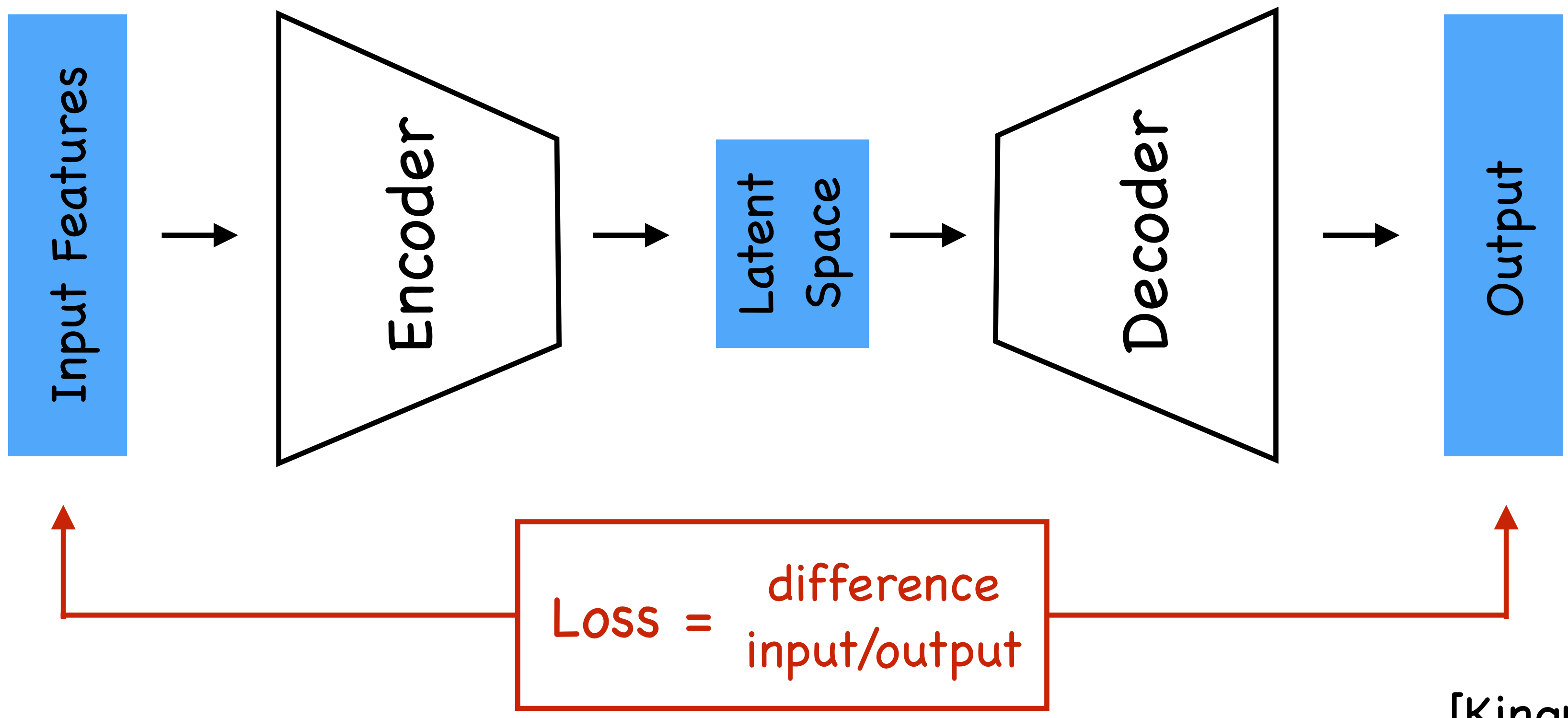
Red and blue dots are true boundaries as calculated by DMRG

Yellow and green shaded area is NN output for the different phases



Autoencoder for unsupervised learning

Most popular NN-based anomaly detection method

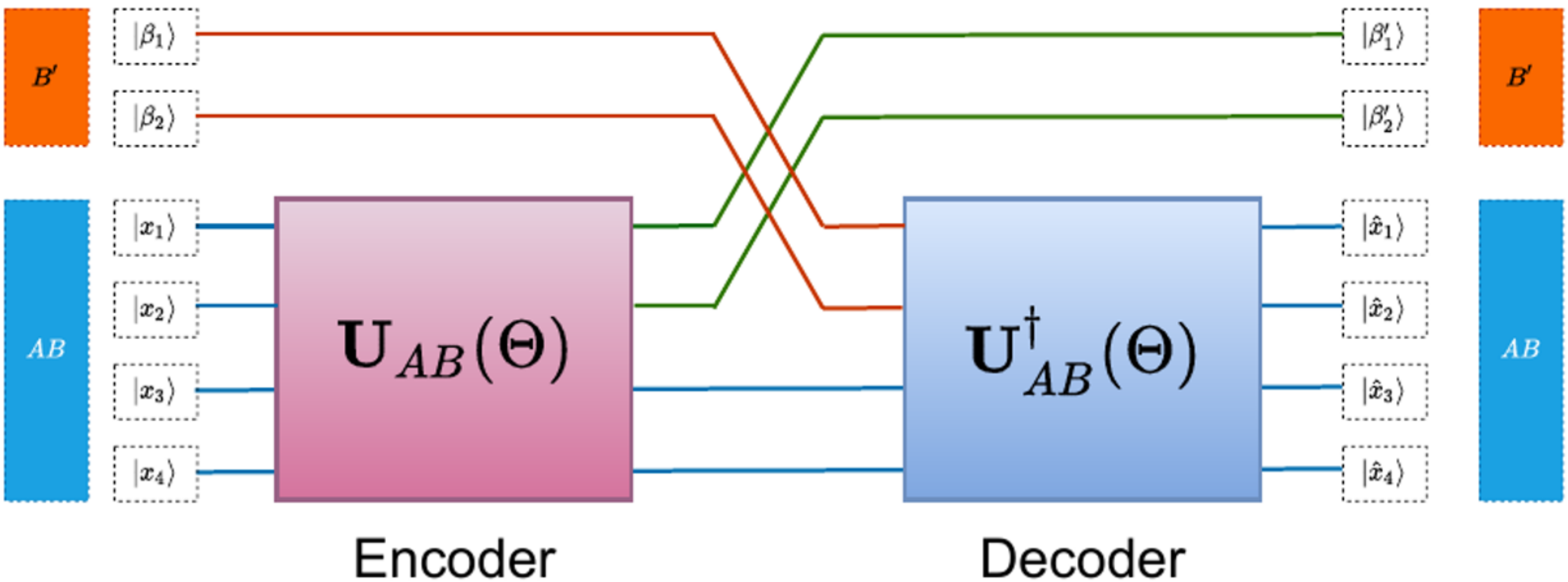


[Kingma, Welling '13]

- in first step input is encoded into information bottleneck
- between input/output layer and bottleneck can be several hidden layers (conv./deep NNs) -> highly non-linear
- after bottleneck decoding step
- Reconstructed output is then compared with input via loss-function (often MSE)
- NN is trained such that input and output high degree of similarity

Unsupervised learning with quantum-gate Autoencoder

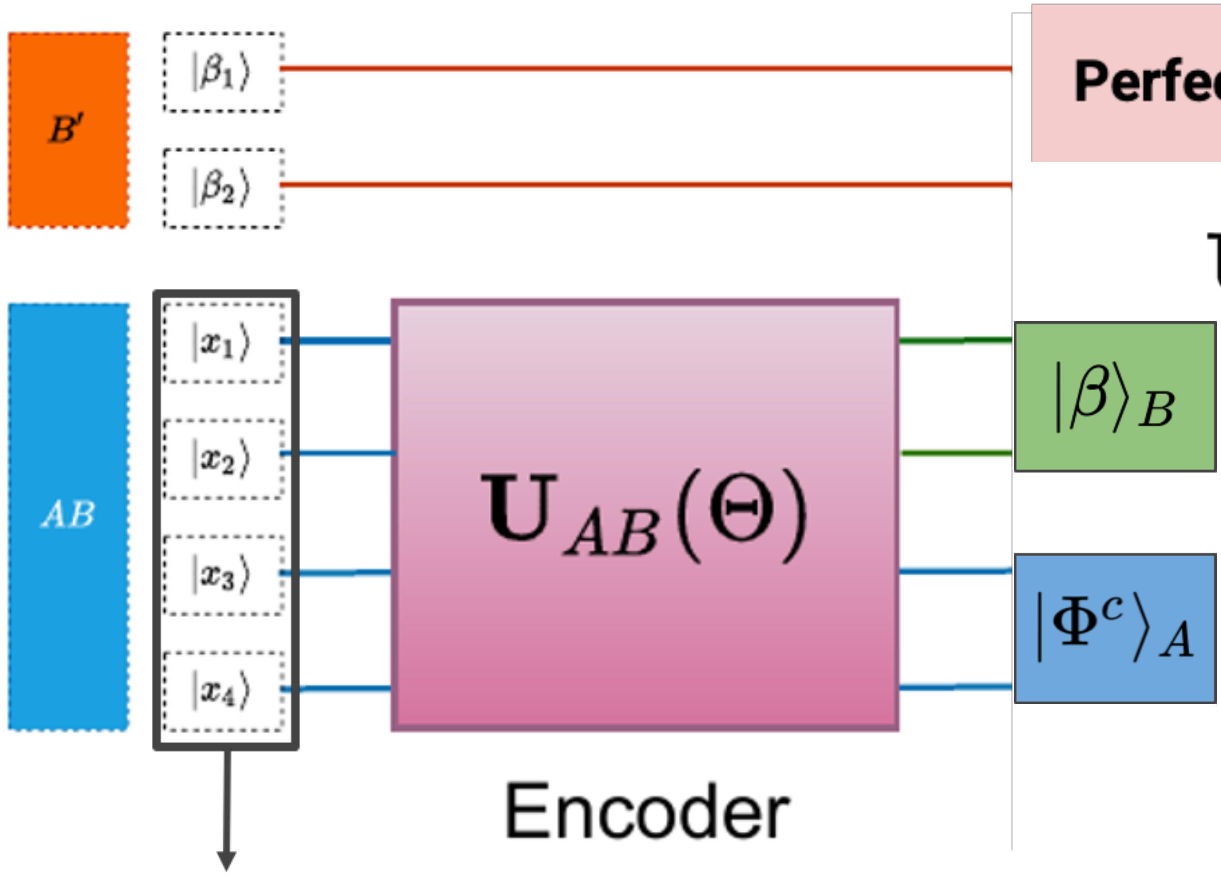
[Ngairangbam, MS, Takeuchi '21]



Quantum unitary transformations = probability conserving!

Induce ***information bottleneck*** by **discarding states of B system** after encoding, and **replacing with reference states B'** with no connection with the encoder.

Unsupervised learning with quantum-gate Autoencoder



Perfect reconstruction possible if

$$\mathbf{U}_{AB}|\Phi\rangle_{AB} = |\Phi^c\rangle_A \otimes |\beta\rangle_B$$

Contains all information of input

Train by tuning the unitary so that the output of the B subsystem is same as the reference state in B'

$$|\Phi\rangle_{AB}$$

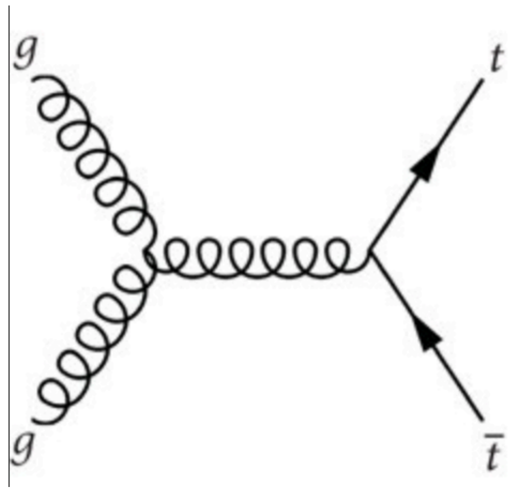
$$|\chi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$$

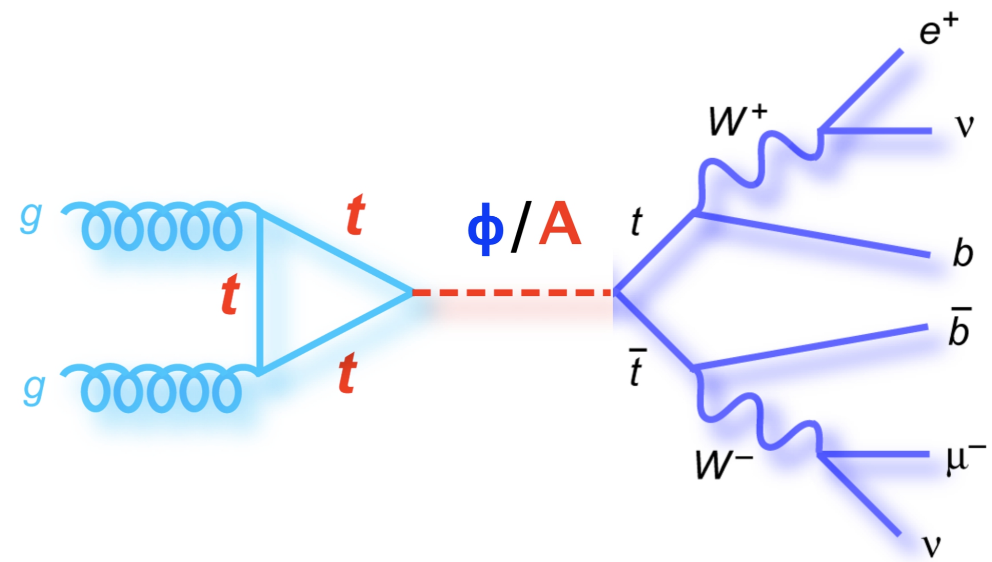
$$\mathbf{U}_{AB}|\Phi\rangle_{AB} = |\chi\rangle_{AB}$$

Quantum similarity measured via fidelity

$$F(|\phi\rangle, |\psi\rangle) = F(|\psi\rangle, |\phi\rangle) = |\langle\phi|\psi\rangle|^2$$

- background signal





$$t\,\bar{t} \to b\,\bar{b}\,W^+\,W^- \to b\,\bar{b}\,l^+\nu\,l'^-\,\bar{\nu}'$$

Use four variables: $(p_T^{b_1}, p_T^{l_1}, p_T^{l_2}, MET)$

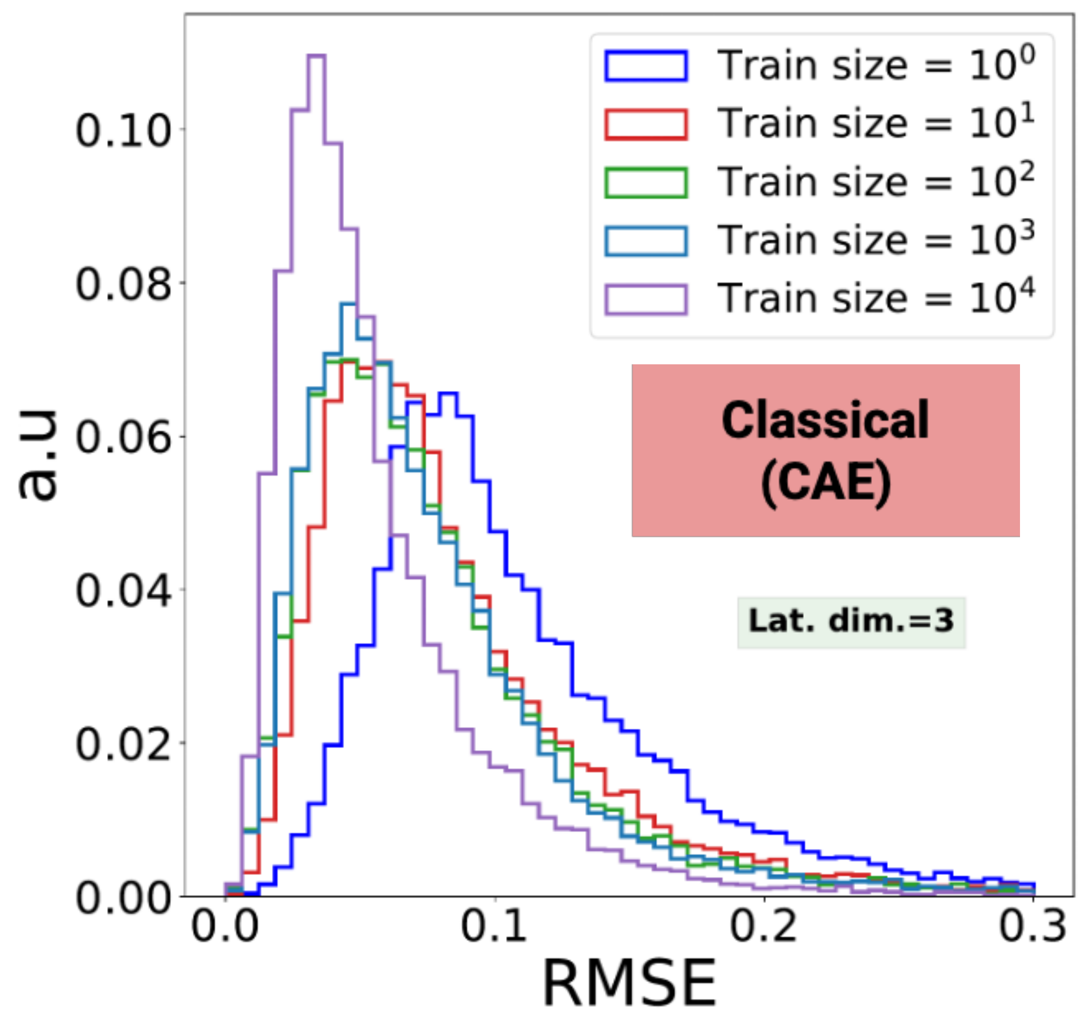
training only on background -> anomaly detection

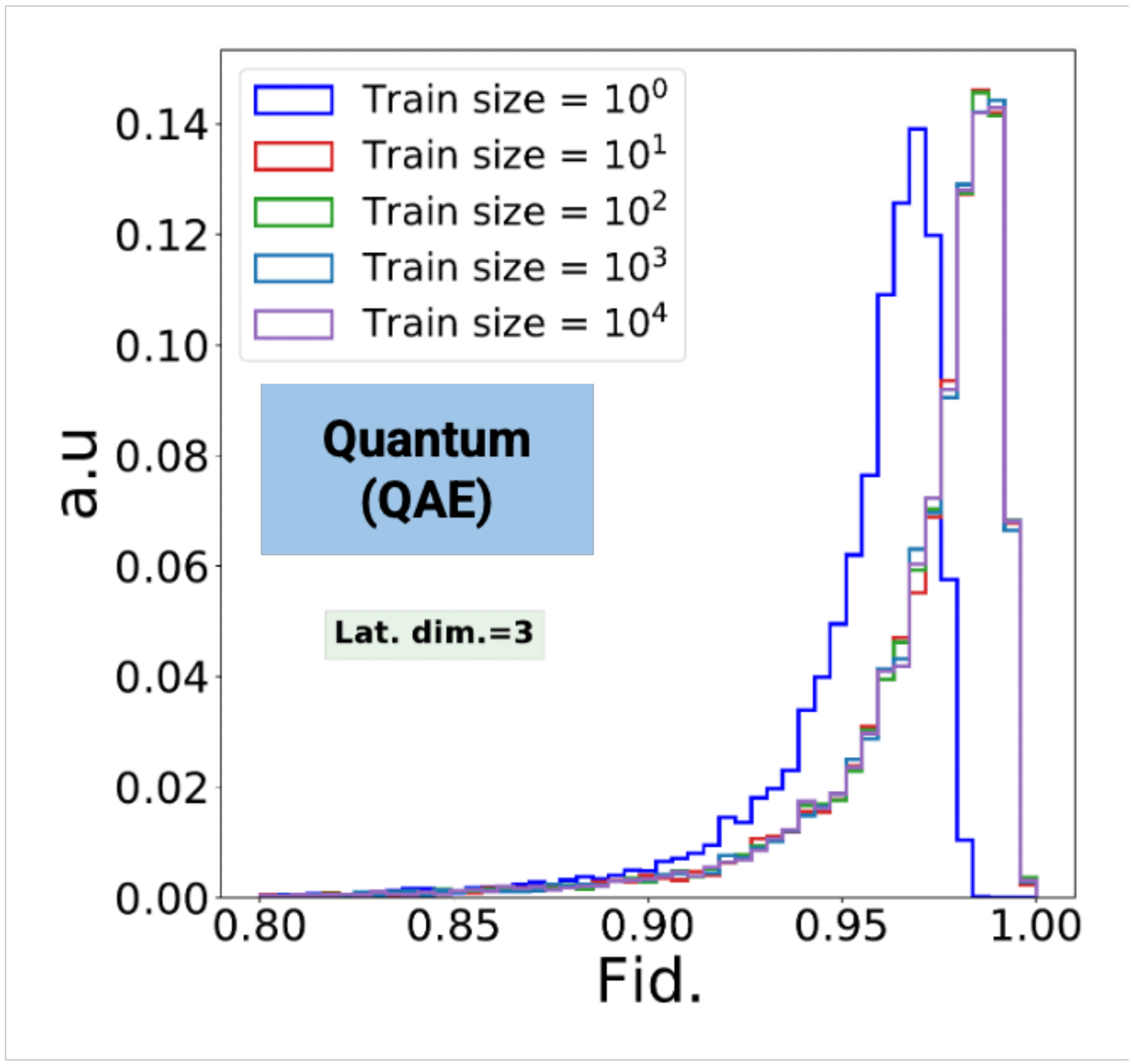
Train a classical autoencoder(CAE) and quantum autoencoder (QAE) with latent spaces: 1,2 and 3 and different sizes of training dataset.

- Also used in (h->inv)jj trained on (Z->inv)jj, with full CAE optimisation

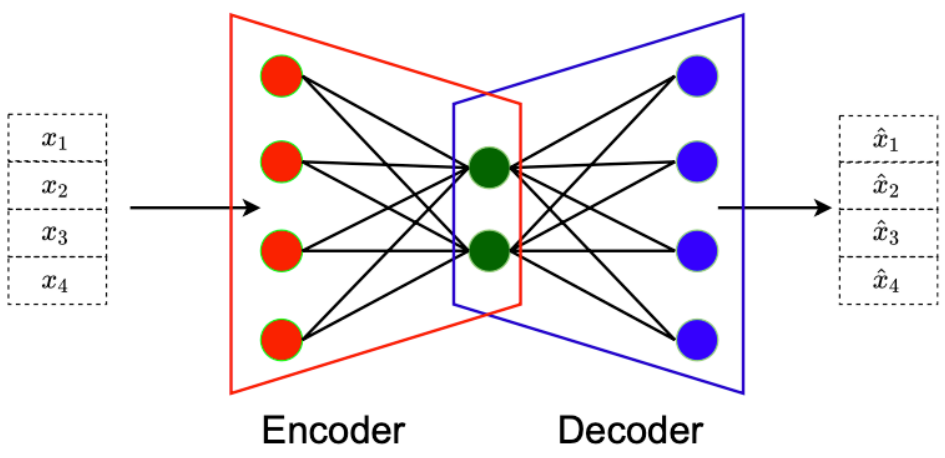
Results: Training size dependence

Dependence of (BG) test loss on training size

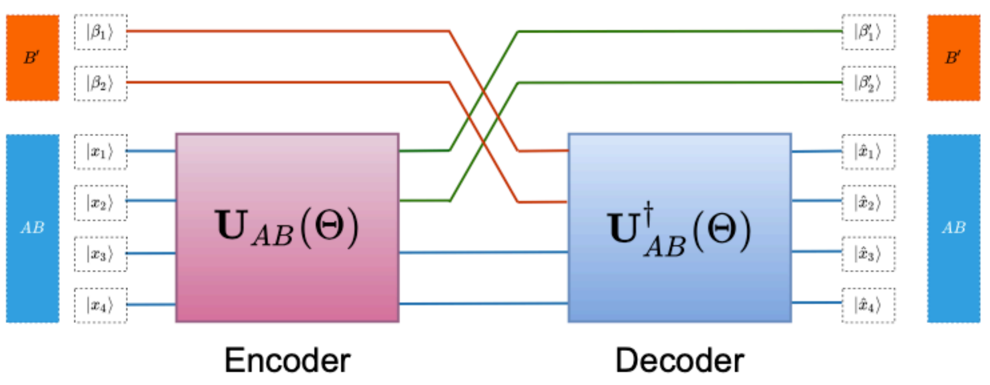


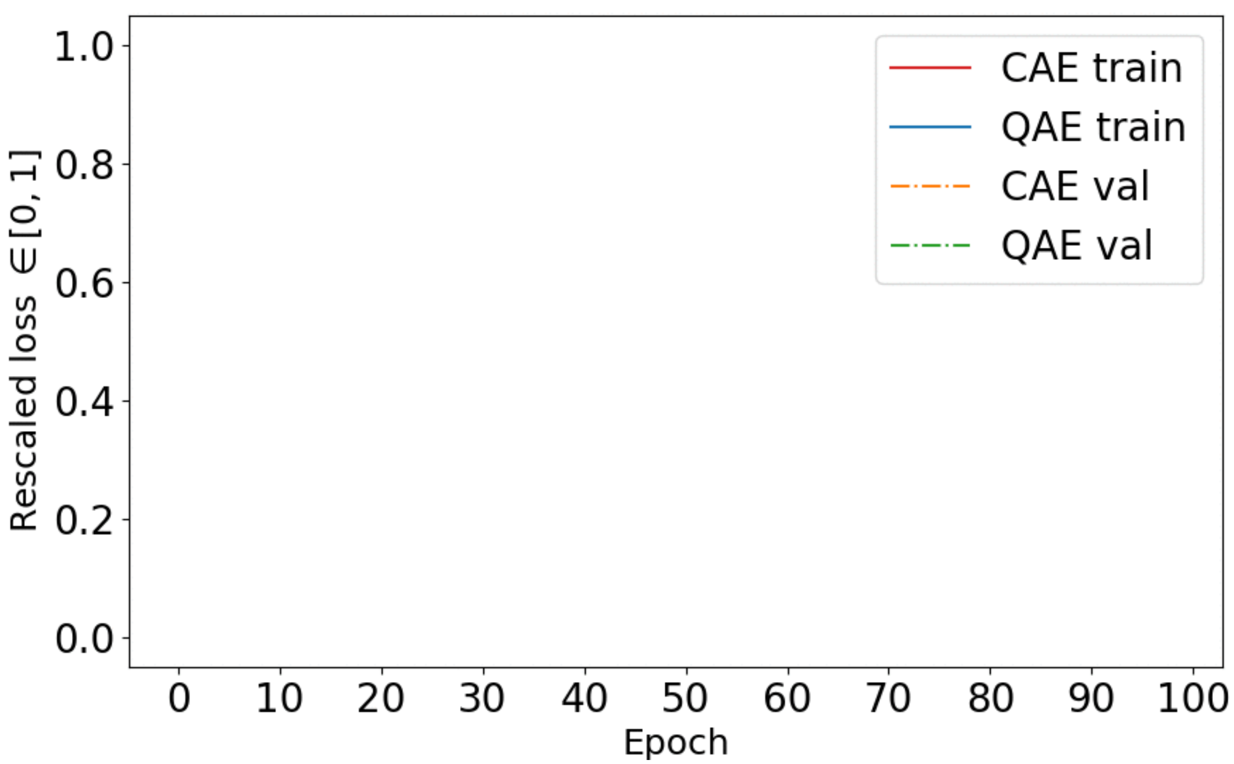


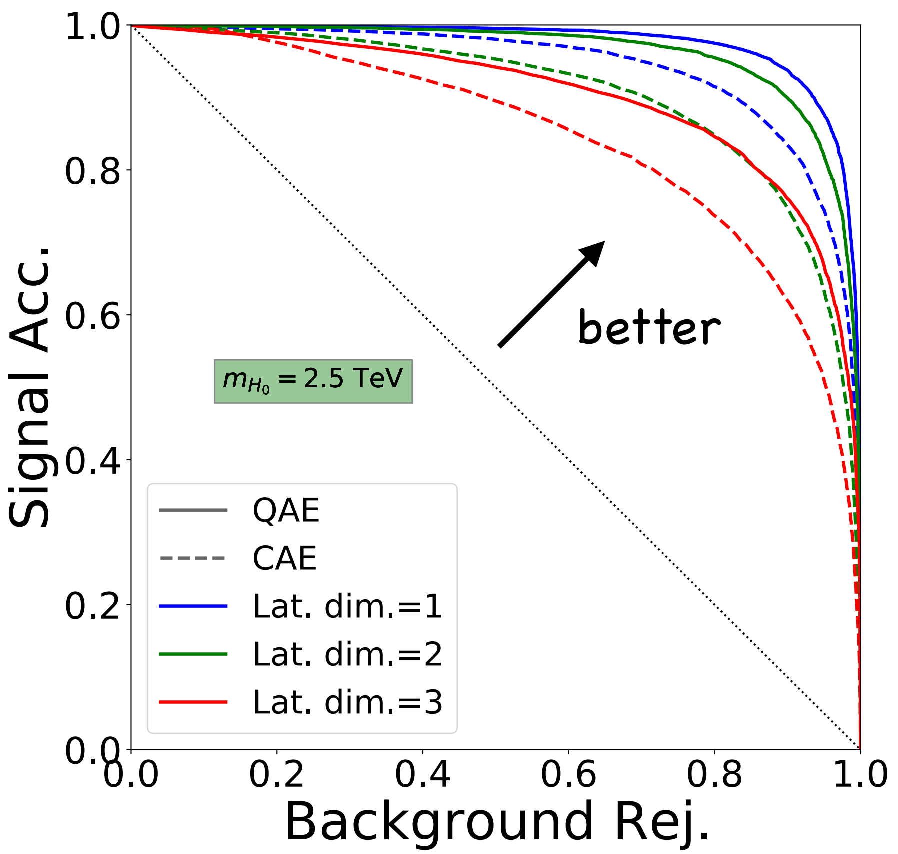
Classical autoencoder



Quantum autoencoder





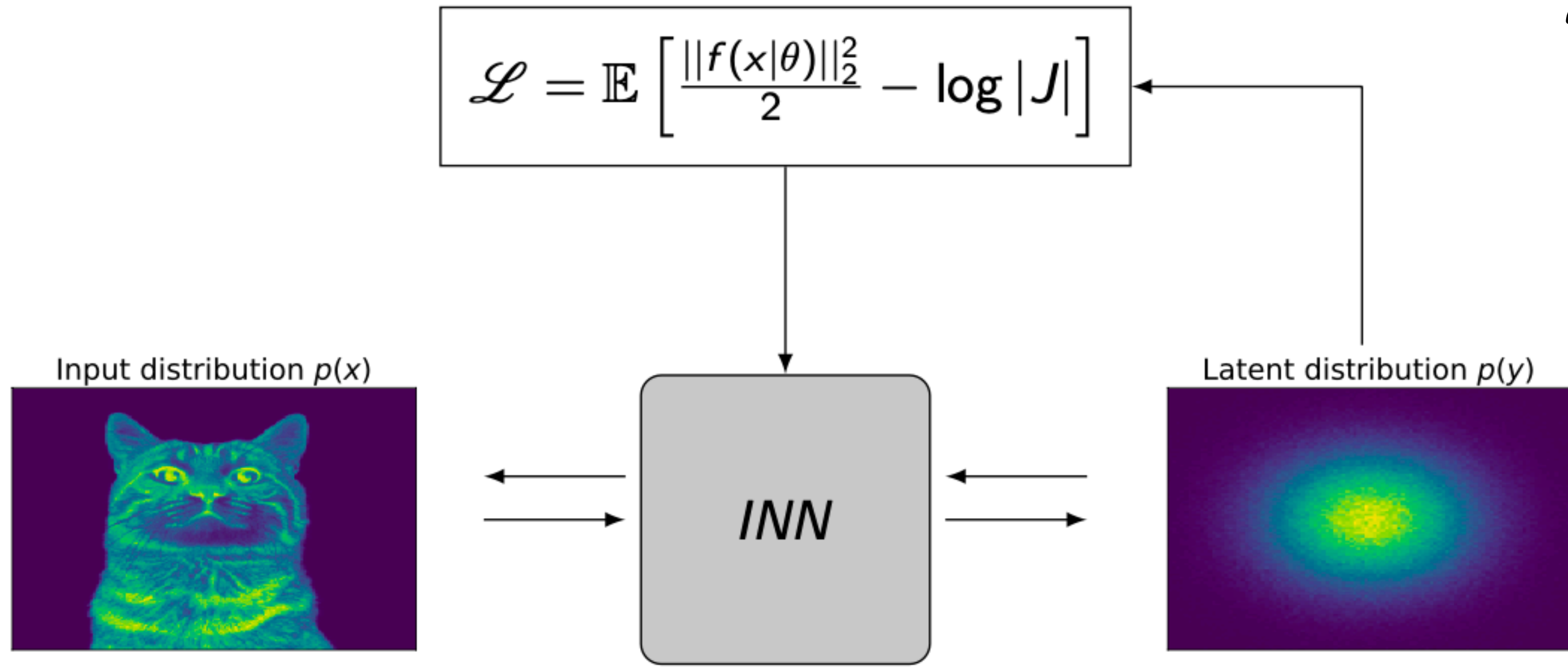


→ Much faster training and better performance for Quantum autoencoder

→ In our test cast, outcome prevails for much larger classical networks

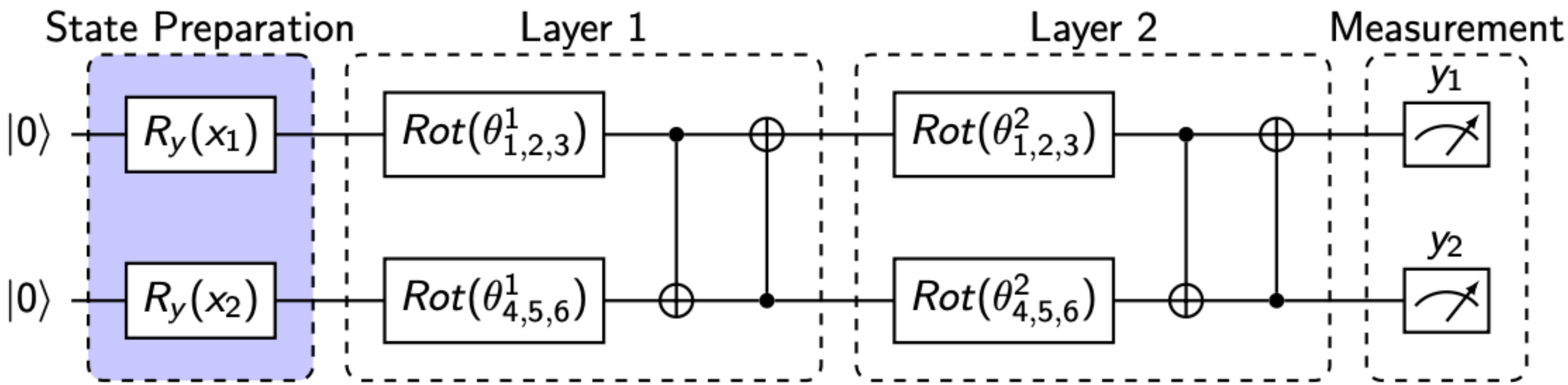
Quantum Invertible Neural Networks
Quantum Normalising Flow

[Rousselot, MS '23
2302.12906]

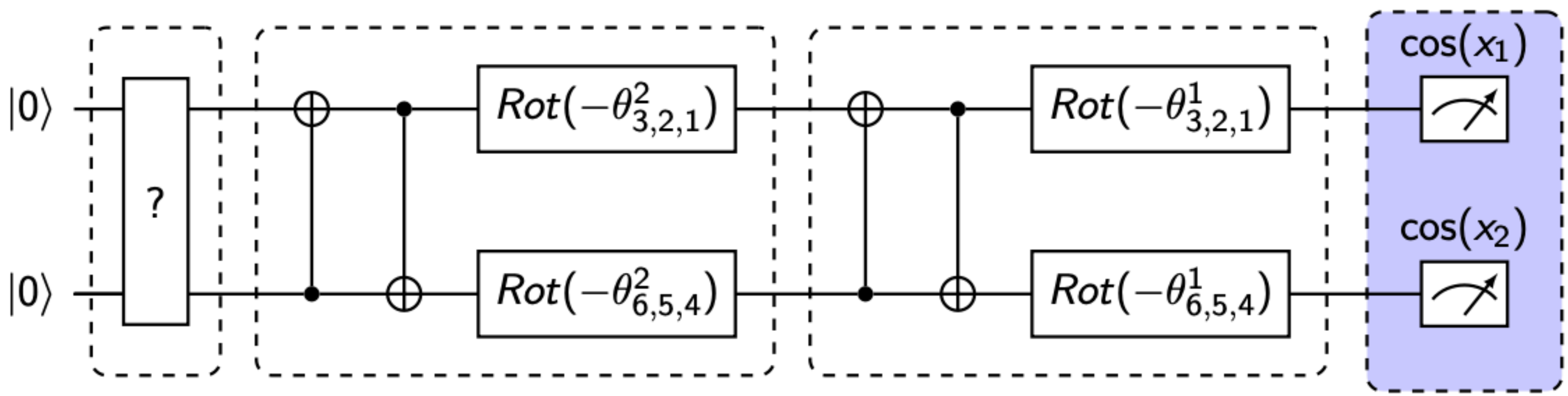


- Train a transformation from input distribution p(x) to gaussian distribution p(y) = N(0,1)
- Can create samples from p(x) by sampling from p(y) and calculating x = Inverse(INN(y))
- Loss function requires jacobian J, which is already available for QNNs via parameter shift rules

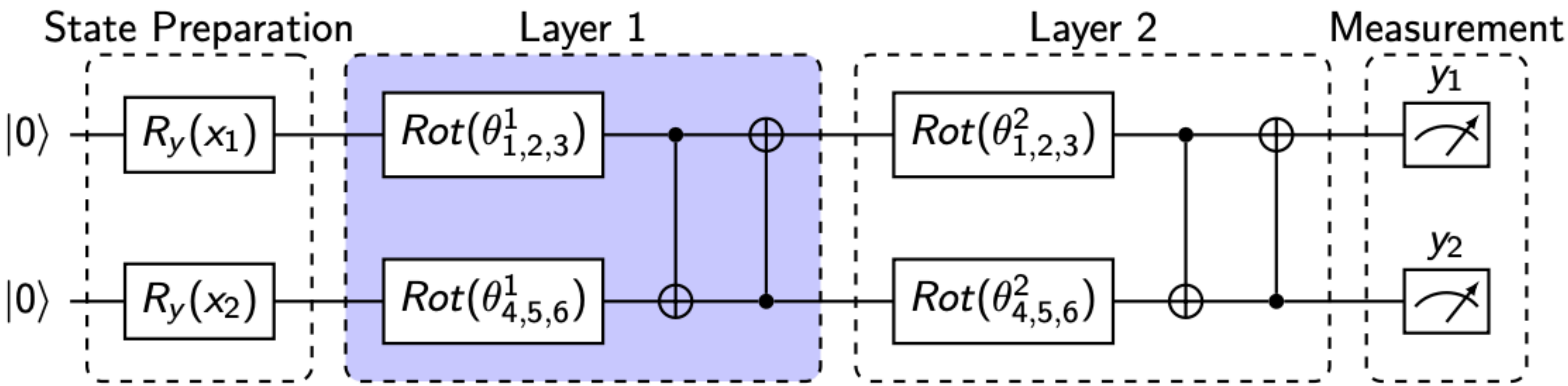
Forward pass:



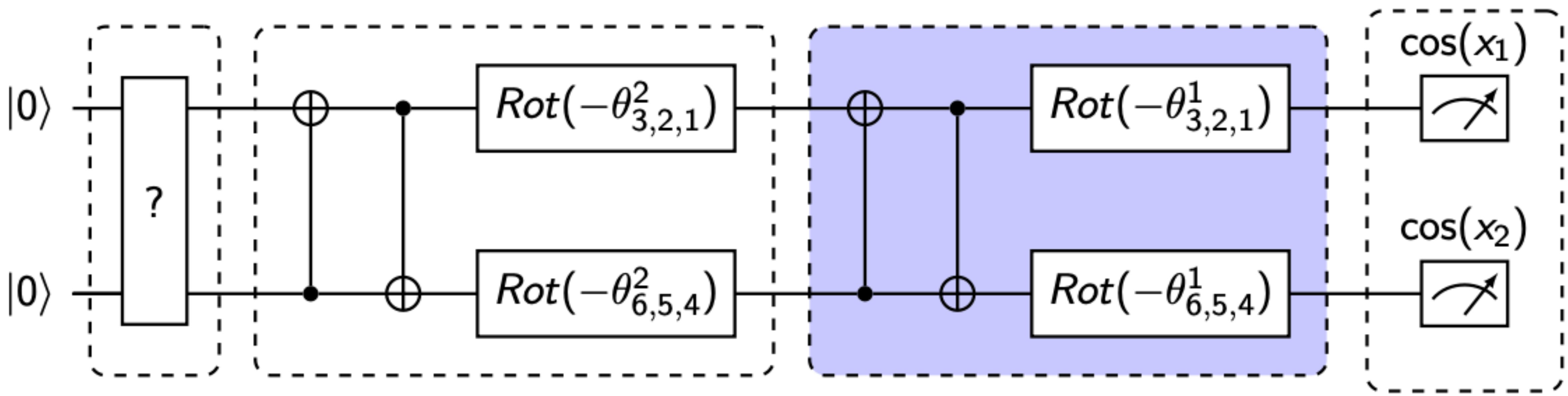
Backward pass:



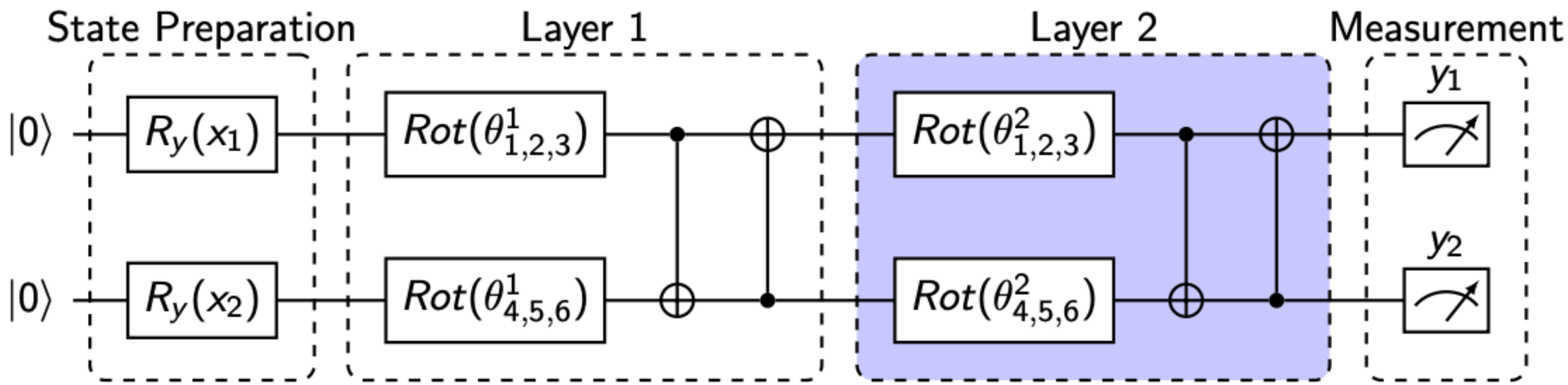
Forward pass:



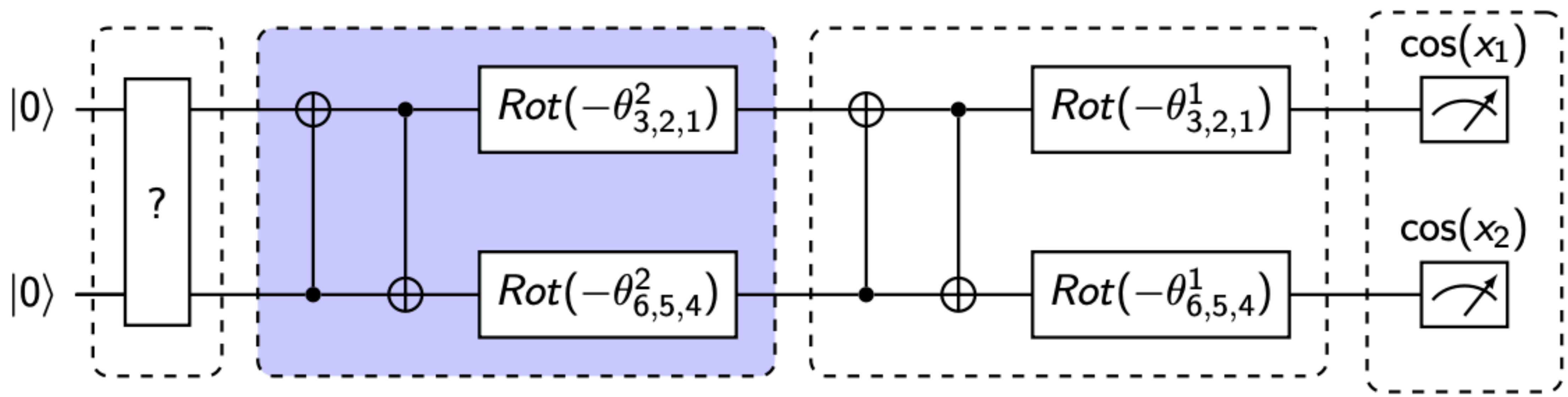
Backward pass:



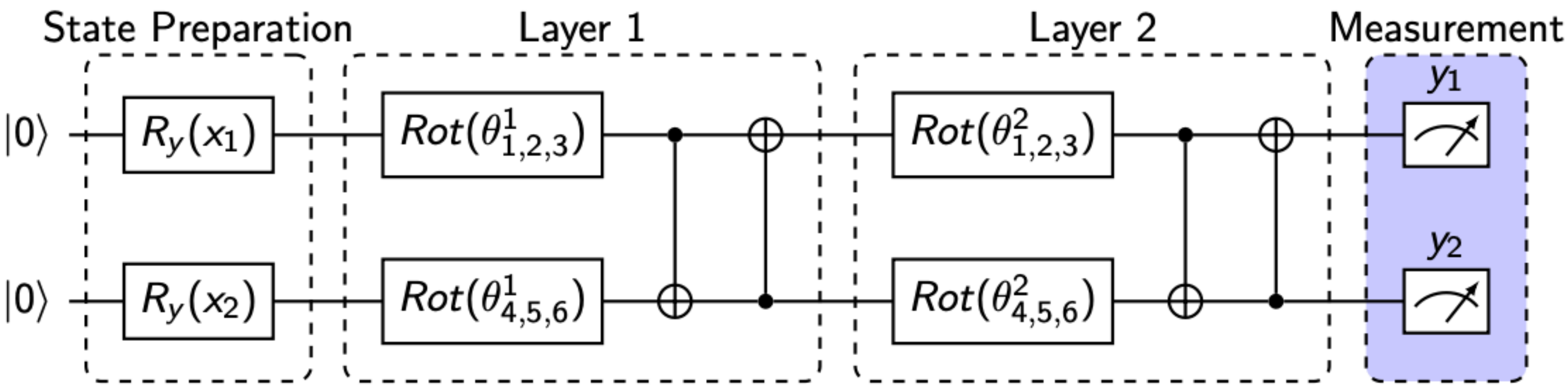
Forward pass:



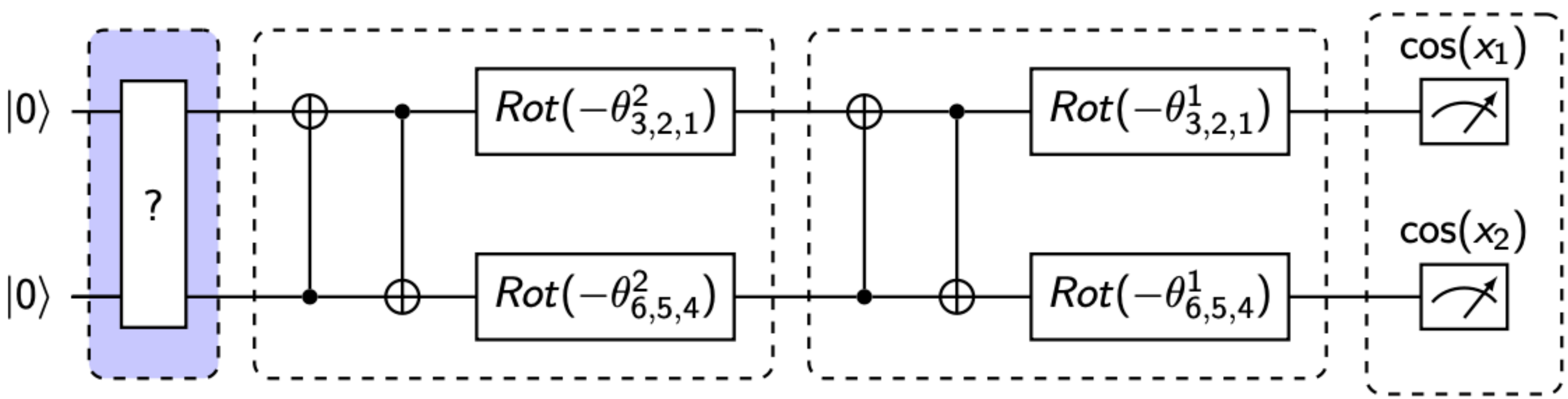
Backward pass:



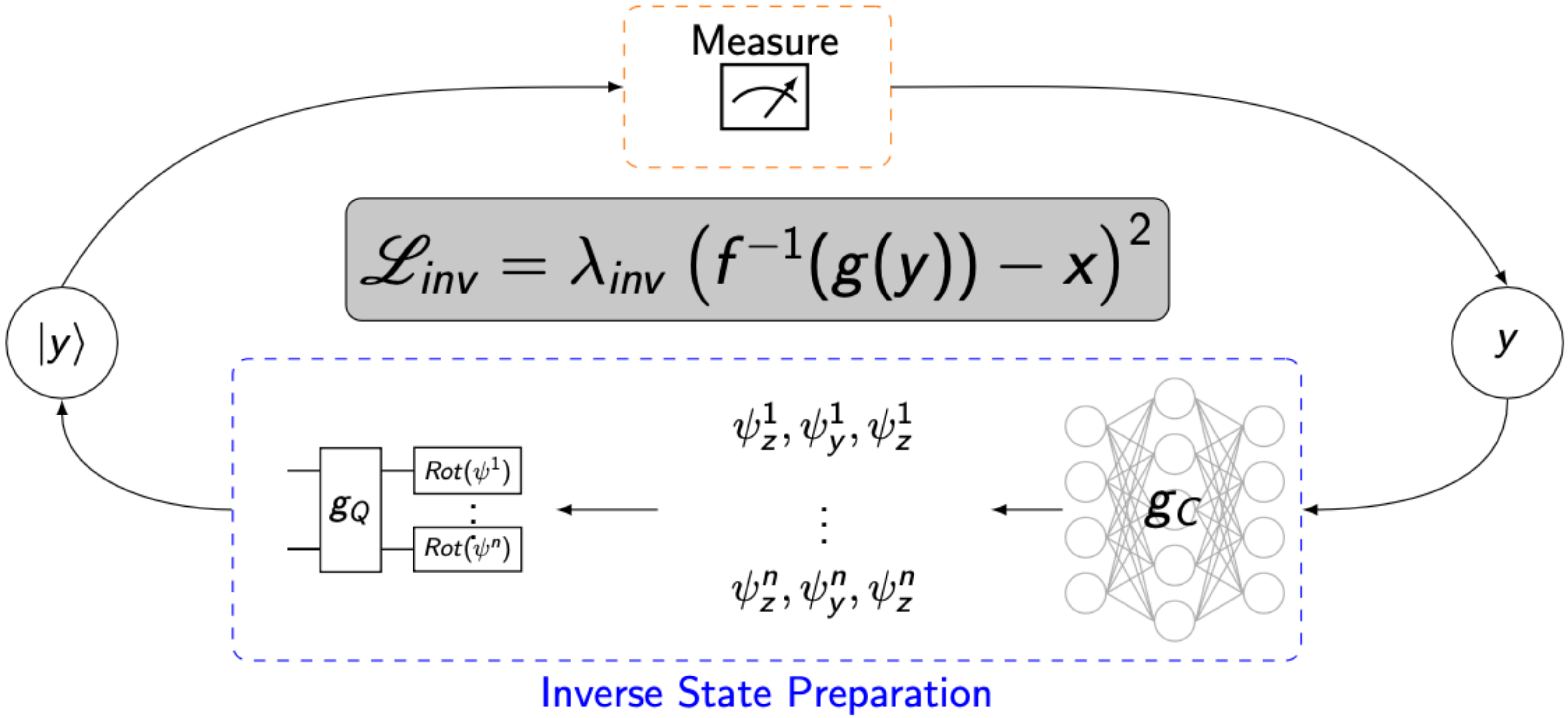
Forward pass:

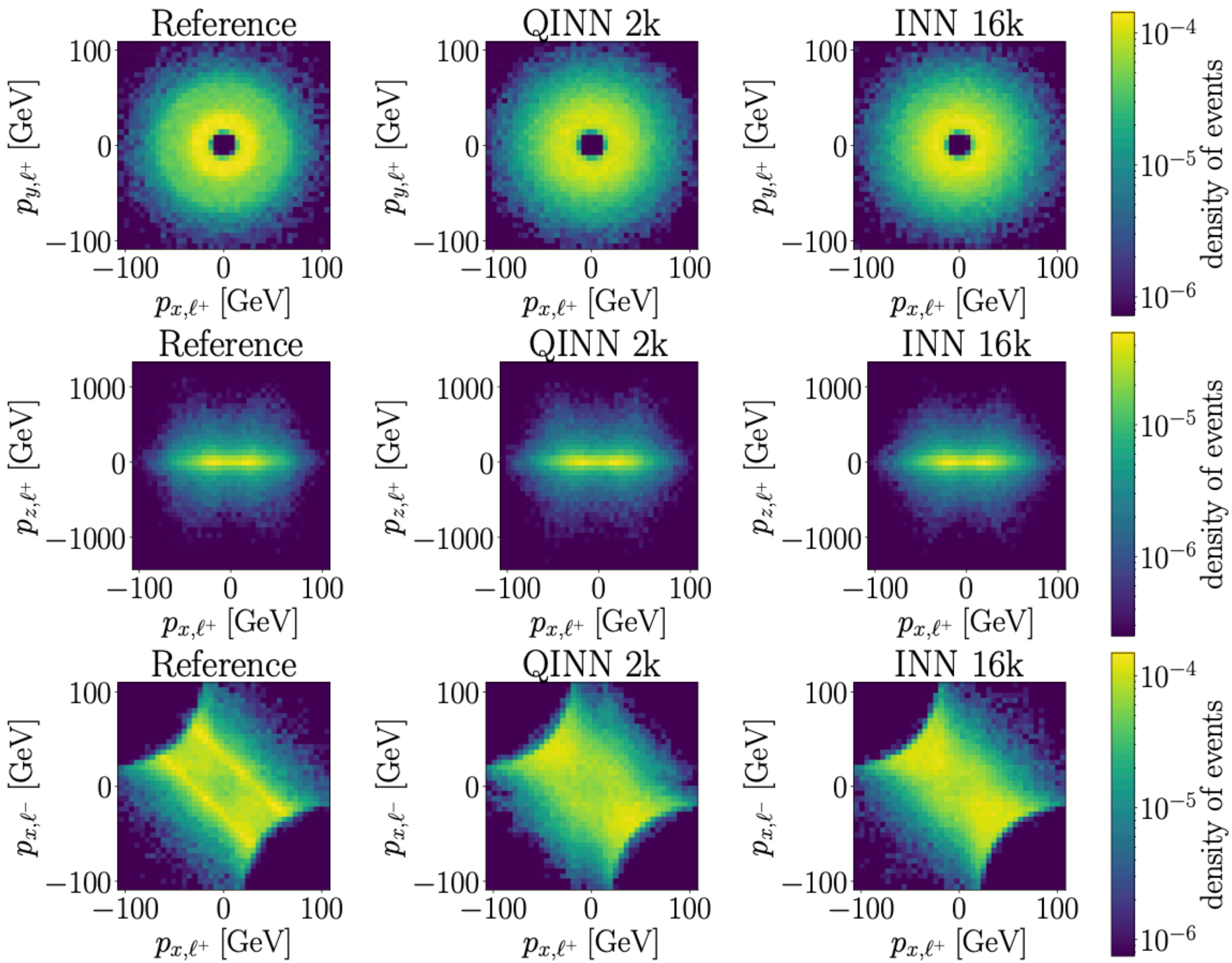


Backward pass:

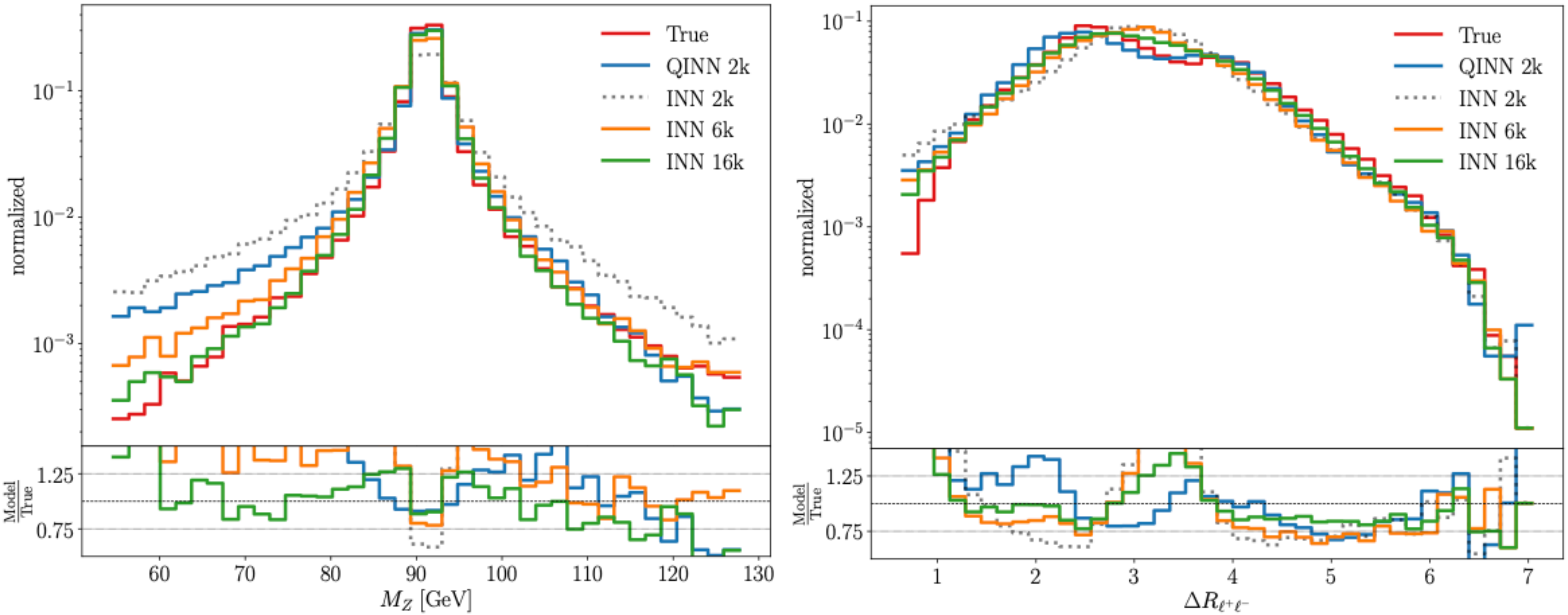


- One solution: Train an **inverse state preparation (ISP)** g : y -> |y> and the model f s.t. $f^{-1}(g(y)) \sim x$




Reference
QINN 2k
INN 16k
p_{y,ℓ^+} [GeV]
p_{x,ℓ^+} [GeV]
p_{z,ℓ^+} [GeV]
p_{x,ℓ^-} [GeV]
density of events
10^{-4}
10^{-5}
10^{-6}
100
0
−100
1000
−1000

process $pp \to Zj \to \ell^+\ell^- j$



Comparison of QINN with INN of varying size

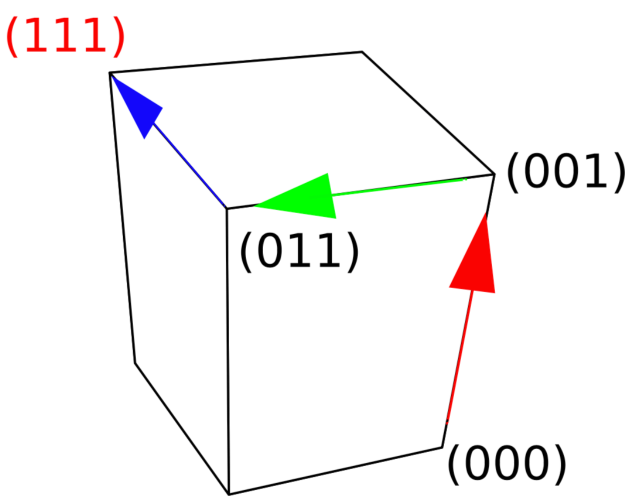
Quantum annealing: Non-universal but powerful?

- Specific Hamiltonian. What does the "anneal" mean?

$$\mathcal{H}_{\rm QA}(t) = \sum_i \sum_j J_{ij}\sigma_i^Z\sigma_j^Z + \sum_i h_i \sigma_i^Z + \Delta(t)\sum_i \sigma_i^X$$

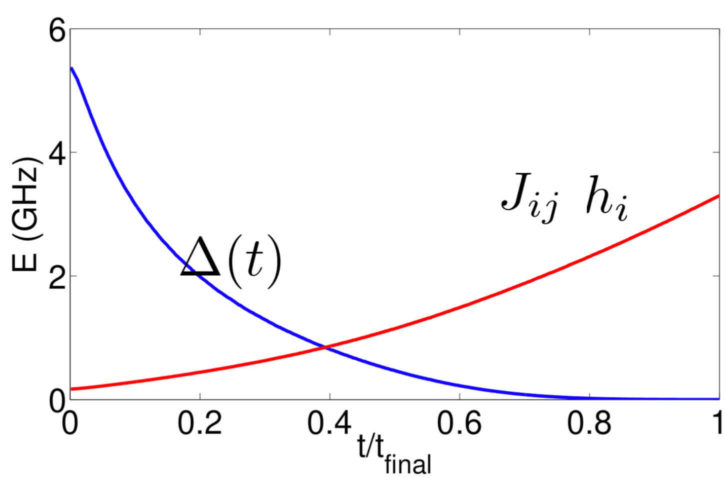
final Hamiltonian (encodes actual problem) — initial Hamiltonian (ground state = superposition of qubits with 0 and 1)

$\Delta(t)$ induces bit-hopping in the Hamming/Hilbert space



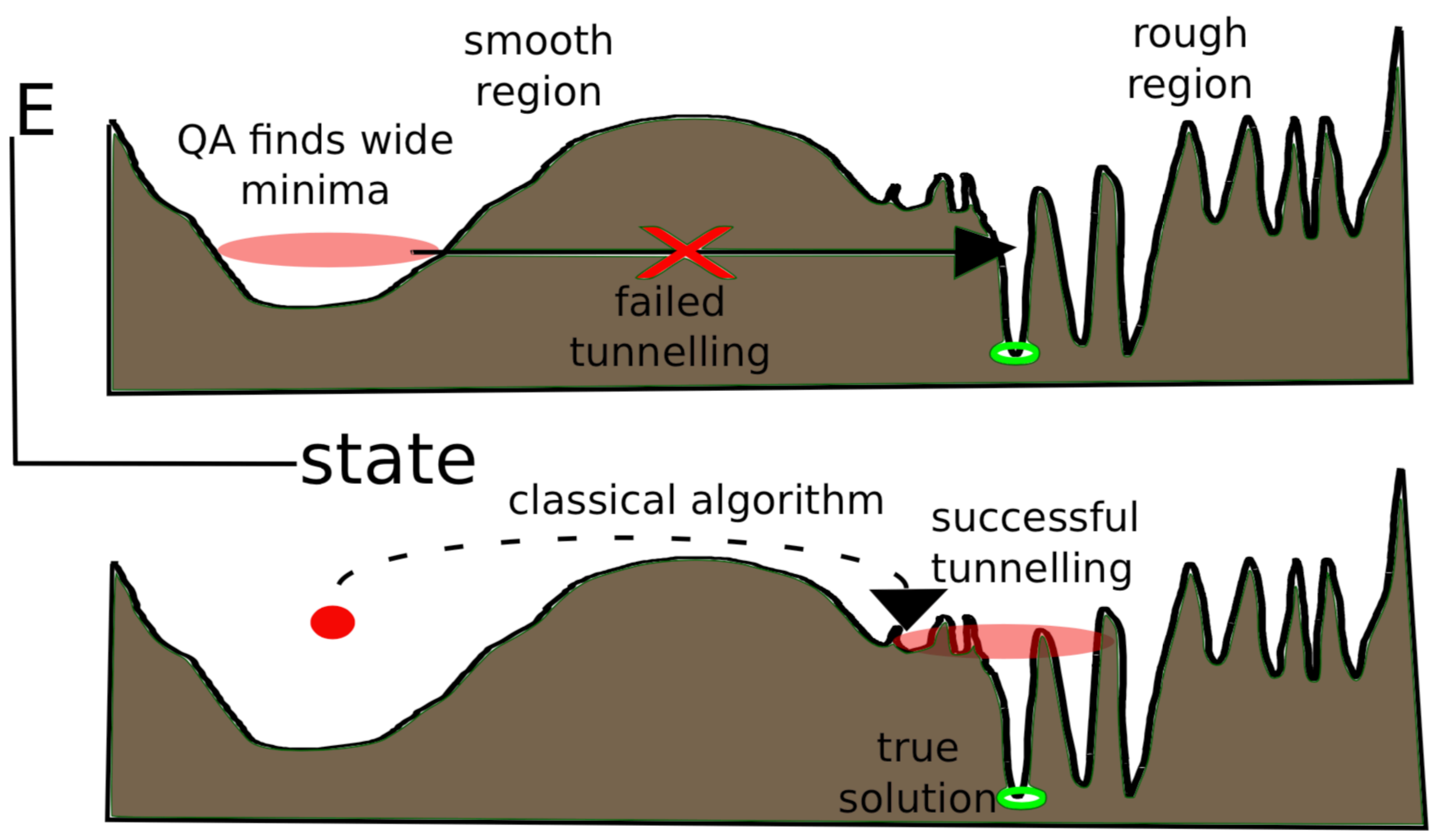
- Anneal idea: transition from ground state of initial Hamiltonian into ground state of problem Hamiltonian

- The idea is to dial this parameter to land in the global minimum (i.e. the solution) of some "problem space" described by J, h:



Thermal (classical) and Quantum Annealing are complementary:

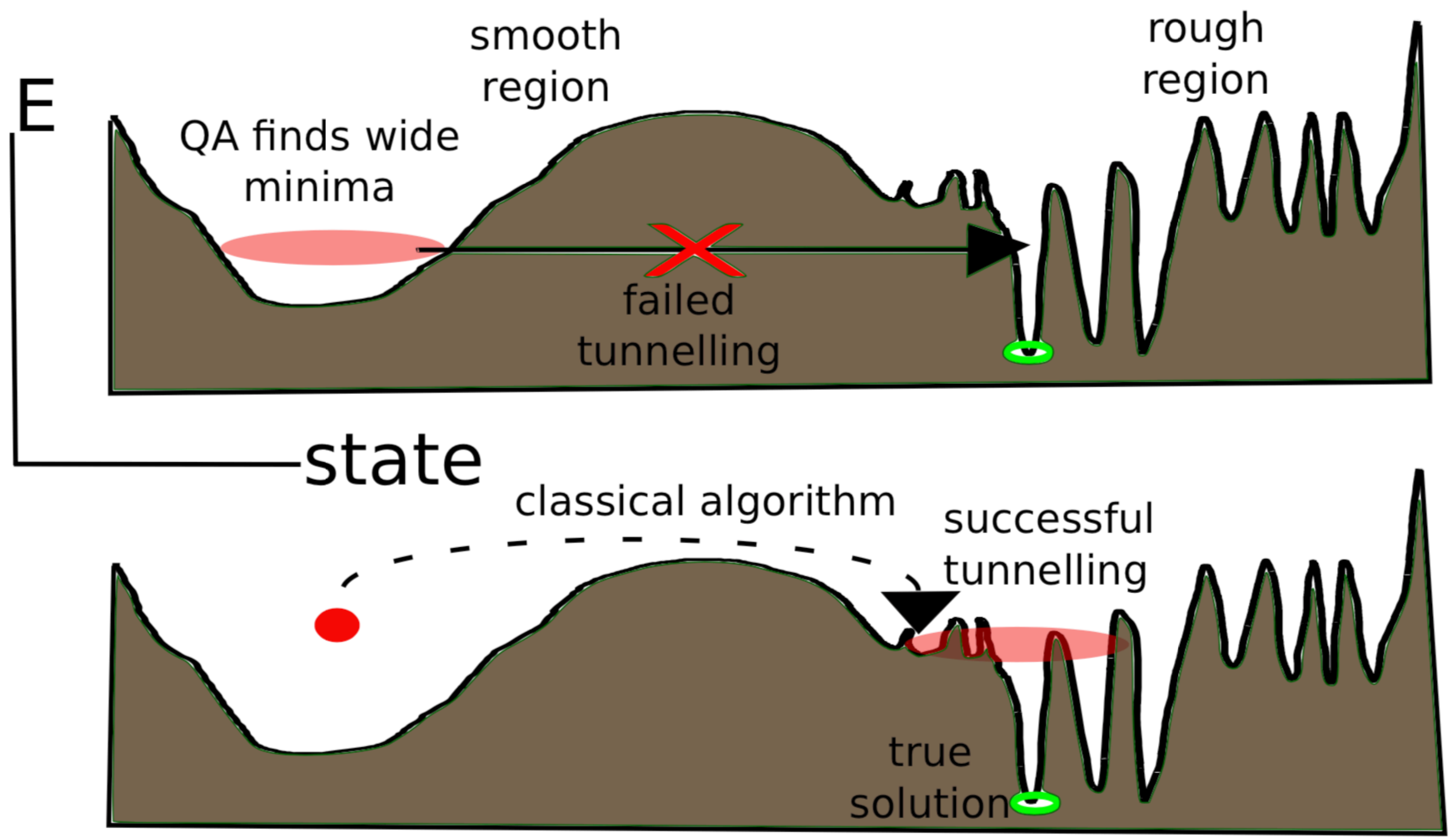
- Thermal tunnelling is fast over broad shallow potentials (Quantum "tunnelling" is exponentially slow)
- Quantum tunnelling is fast through tall thin potentials (Thermal "tunnelling" is exponentially slow - Boltzmann suppression)
- Hybrid approach can be useful depending on solution landscape



- More specifically - thermal annealing uses Metropolis algorithm: accept random σ_i^Z flips with probability

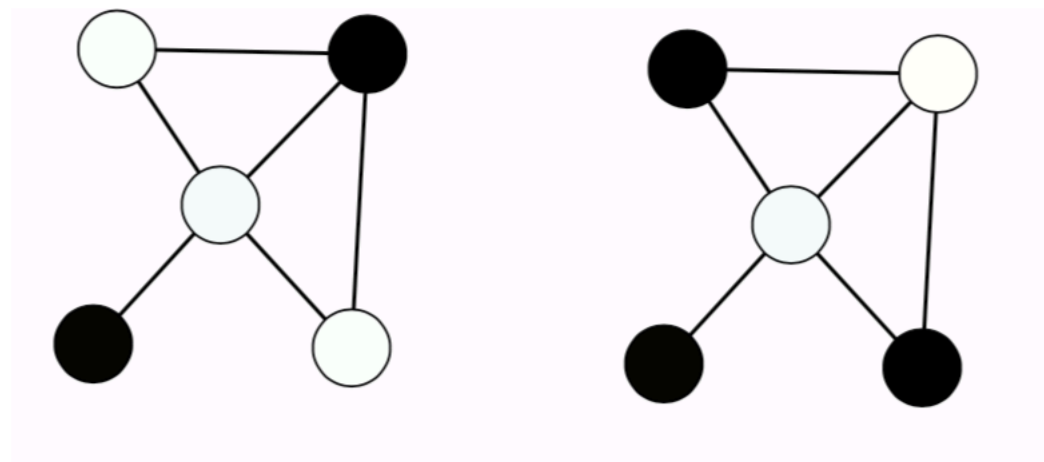
$$P = \begin{cases} 1 & \Delta H \leq 0 \\ e^{-\Delta H/KT} & \Delta H > 0 \end{cases}$$

- Quantum tunnelling in QFT happens with probability $P \sim e^{-w\sqrt{2m\Delta H}/\hbar}$ so by contrast it can be operative for tall barriers if they are thin



How to encode a problem on an Ising model

Example 1: how many vertices on a graph can we colour so that none touch?



NP problem

Let non-coloured vertices have $\sigma_i^Z = -1$ and coloured ones have $\sigma_i^Z = +1$

Add a reward for every coloured vertex, and for each link between vertices i,j we add a penalty if there are two +1 eigenvalues:

$$\mathcal{H} = -\Lambda \sum_i \sigma_i^Z + \sum_{\text{linked pairs } \{i,j\}} \left[\sigma_i^Z + \sigma_j^Z + \sigma_i^Z \sigma_j^Z\right]$$

Example 2: N^2 students are to sit an exam in a square room with NxN desks 1.5m apart. Half the students (A) have a virus while half of them (B) do not.
How can they be arranged to minimise the number of infections due to <2m social distancing?

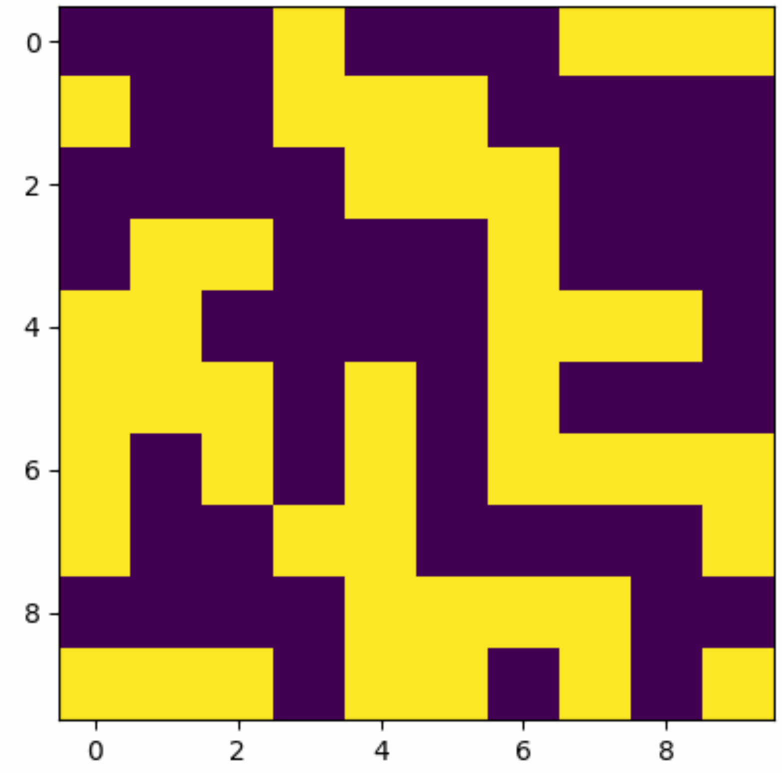
There are N^2 spins σ^Z_{lN+j} arranged in rows and columns. We do not care if A>=<A or B>=<B, but if A>=<B then we put a penalty of 2+ on the Hamiltonian (ferromagnetic coupling)

$$\mathcal{H} = \sum_{\ell m=1}^{N} \sum_{ij=1}^{N} \left(\delta_{\ell m}(\delta_{(i+1)j} + \delta_{(i-1)j}) + \delta_{ij}(\delta_{(\ell+1)m} + \delta_{(\ell-1)m})\right) \left[1 - \sigma^Z_{\ell N+i}\sigma^Z_{mN+j}\right]$$

Finally we need to apply constraint that #A=#B:

$$\mathcal{H}^{(\mathrm{constr})} = \Lambda\,(\#A - \#B)^2 \quad = \Lambda \left(\sum_{\ell,i}^{N} \sigma^Z_{\ell N+i}\right)^2 \quad = \Lambda \sum_{\ell m=1}^{N} \sum_{ij=1}^{N} \sigma^Z_{\ell N+i}\sigma^Z_{mN+j}$$

- Example 2 done with classical thermal annealing using the Metropolis algorithm. Note this represents a search over solution space of 2^100 configurations



- Importantly the constraint hamiltonian cannot be too big otherwise the hills are too high and it freezes too early. This makes the process require a (polynomial sized) bit of "thermal tuning".

- Could be done more easily on quantum annealers as constraints could be high and it would still work, e.g. D-Wave quantum annealer. However, architecture (connectivity of J, h) is limited.

A quantum laboratory for QFT and QML

- going beyond the reach of classical computers -

- Using the spin-chain approach for field theories discussed before, we can encode a QFT on a quantum annealer and study its dynamics directly.

[Abel, MS '20]

- To show that the system is a true and genuine quantum system we investigate if the state can tunnel from a meta-stable vacuum into a the true vacuum.

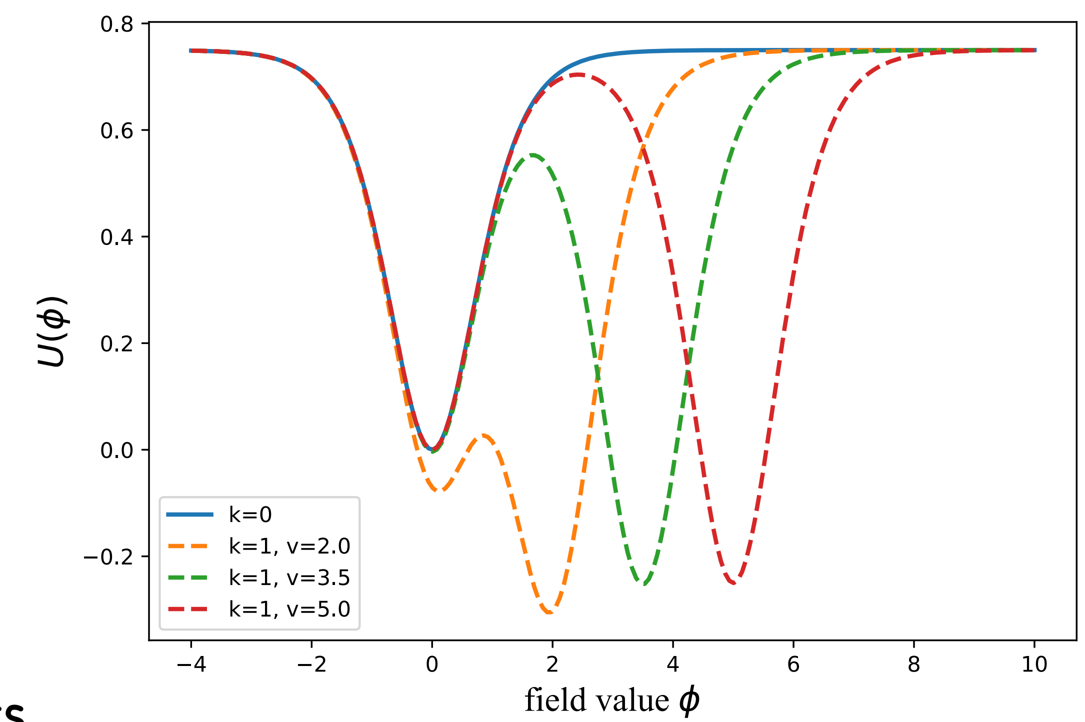
- Choose a potential of interest:

$$U(\phi) = \frac{3}{4}\tanh^2\phi - k(t)\,\mathrm{sech}^2\left(c(\phi - v)\right)$$

time dependent

where $\phi = \eta/\eta_0$

$\phi(t)$ is the field and c, v are dimless constants



The tunnelling probability in a QFT is calculated by evaluating the path-integral in Euclidean space around the action's critical points using the steepest gradient-descent method

$$\langle \eta_i | \eta_f \rangle_E = \int \mathcal{D}\delta\eta \, e^{-\hbar^{-1} \int dt \left(\frac{m(\dot{\eta}_{cl} + \delta\dot{\eta})^2}{2} + U(\eta_{cl} + \delta\eta) - E_0 \right)} = A e^{-\hbar^{-1} S_{E,cl}}$$

quantum annealer

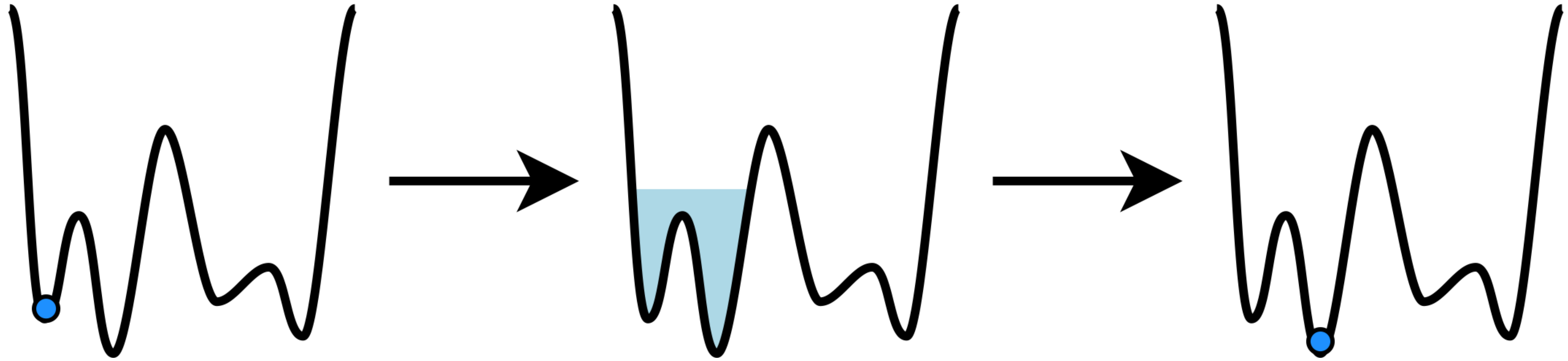
For the tunnelling rate $\Gamma = |\langle \eta_i | \eta_f \rangle_E|^2 \approx e^{-2\hbar^{-1} S_{E,cl}}$ with $S_{E,cl} = \int_{\eta_+}^{\eta_e} d\eta \sqrt{2m(U - E_0)}$

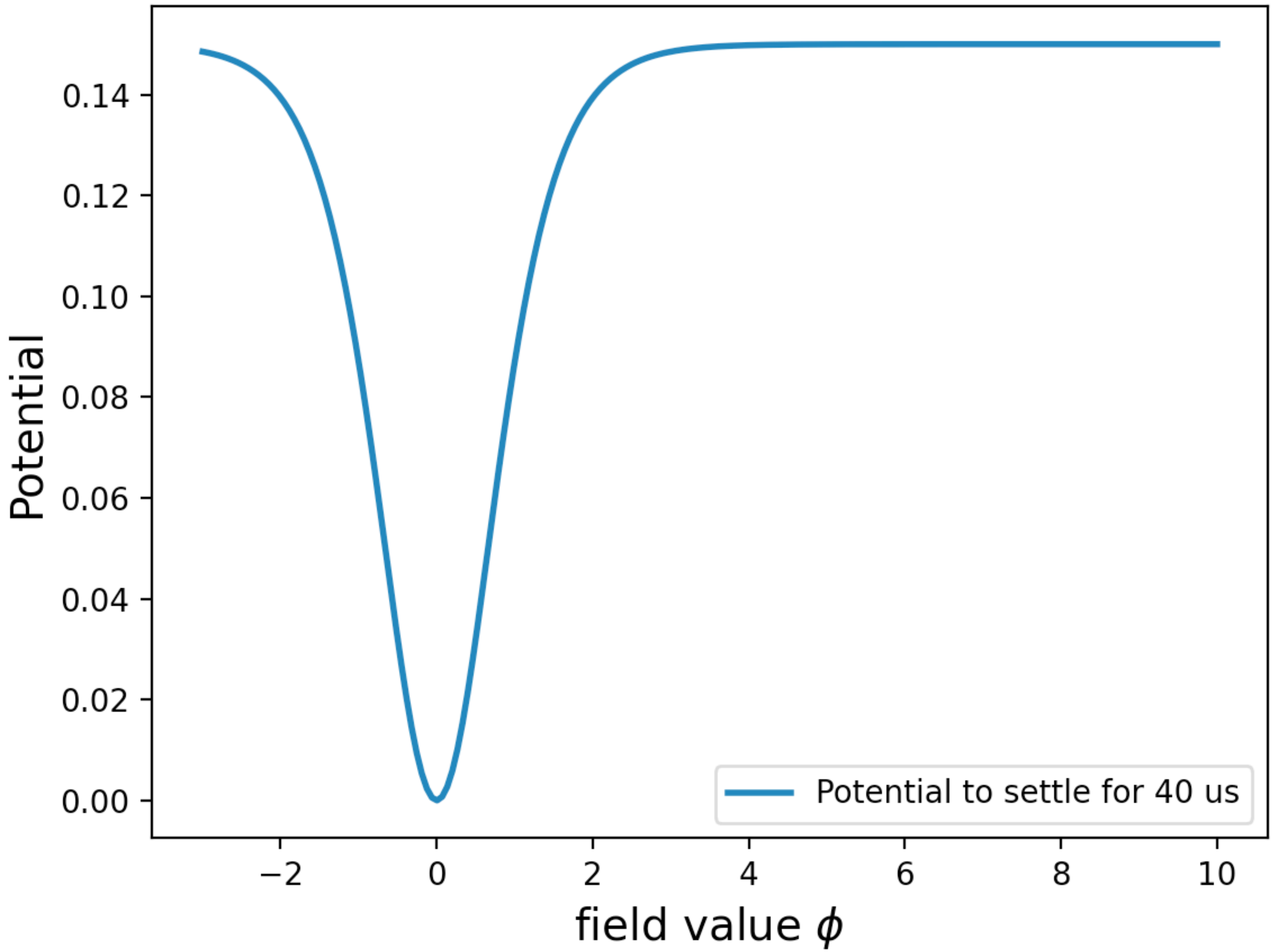
Exponent is object of interest: $\hbar^{-1} S_E \approx \gamma^{-\frac{1}{2}} \int_{\phi_+}^{\phi_e} \sqrt{\frac{3}{4} \tanh^2 \phi - \mathrm{sech}^2(\phi - v)} \, d\phi$ with $\gamma \stackrel{\text{def}}{=} \hbar^2 / 2m\eta_0^2$

$$\log \Gamma \approx -2\hbar^{-1} S_E \approx \sqrt{\frac{3}{\gamma}} \left(\frac{5}{3} - v \right)$$

D-Wave reverse annealing

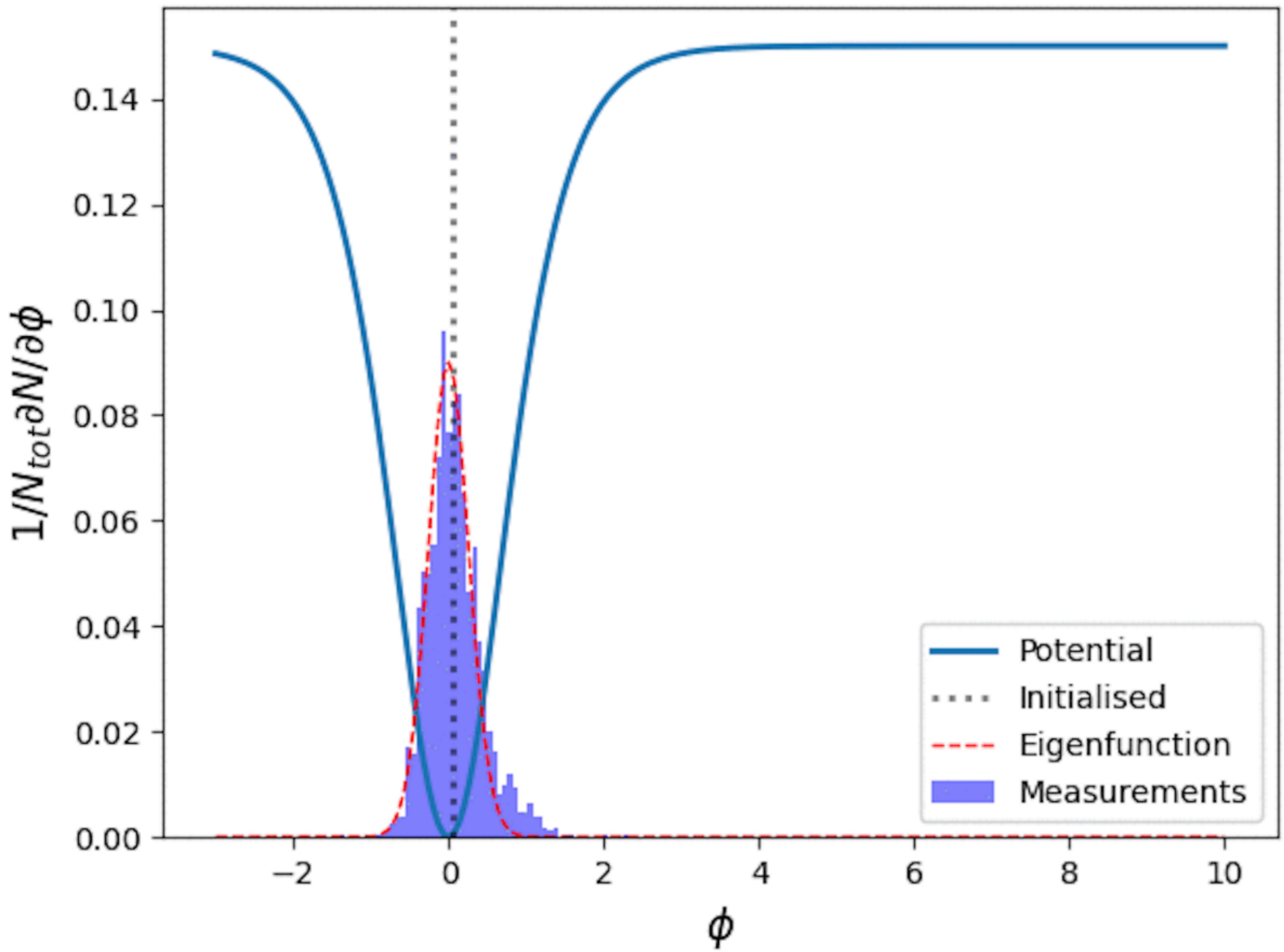
starts at sq=1 (classical) -> sq < 1 (quantum) -> measurement in sq=1 (classical)

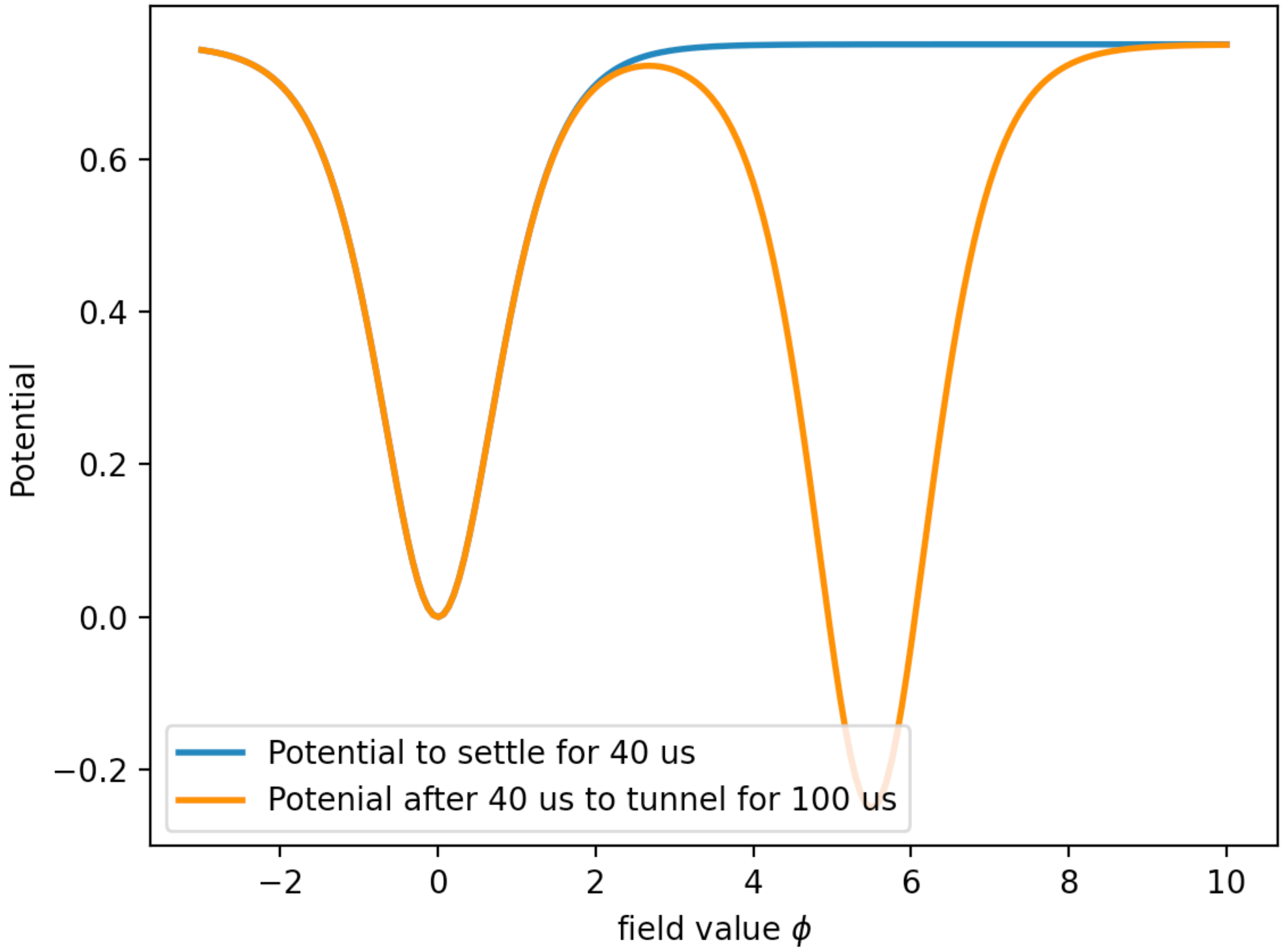




0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Potential
−2
0
2
4
6
8
10
field value ϕ
Potential to settle for 40 us

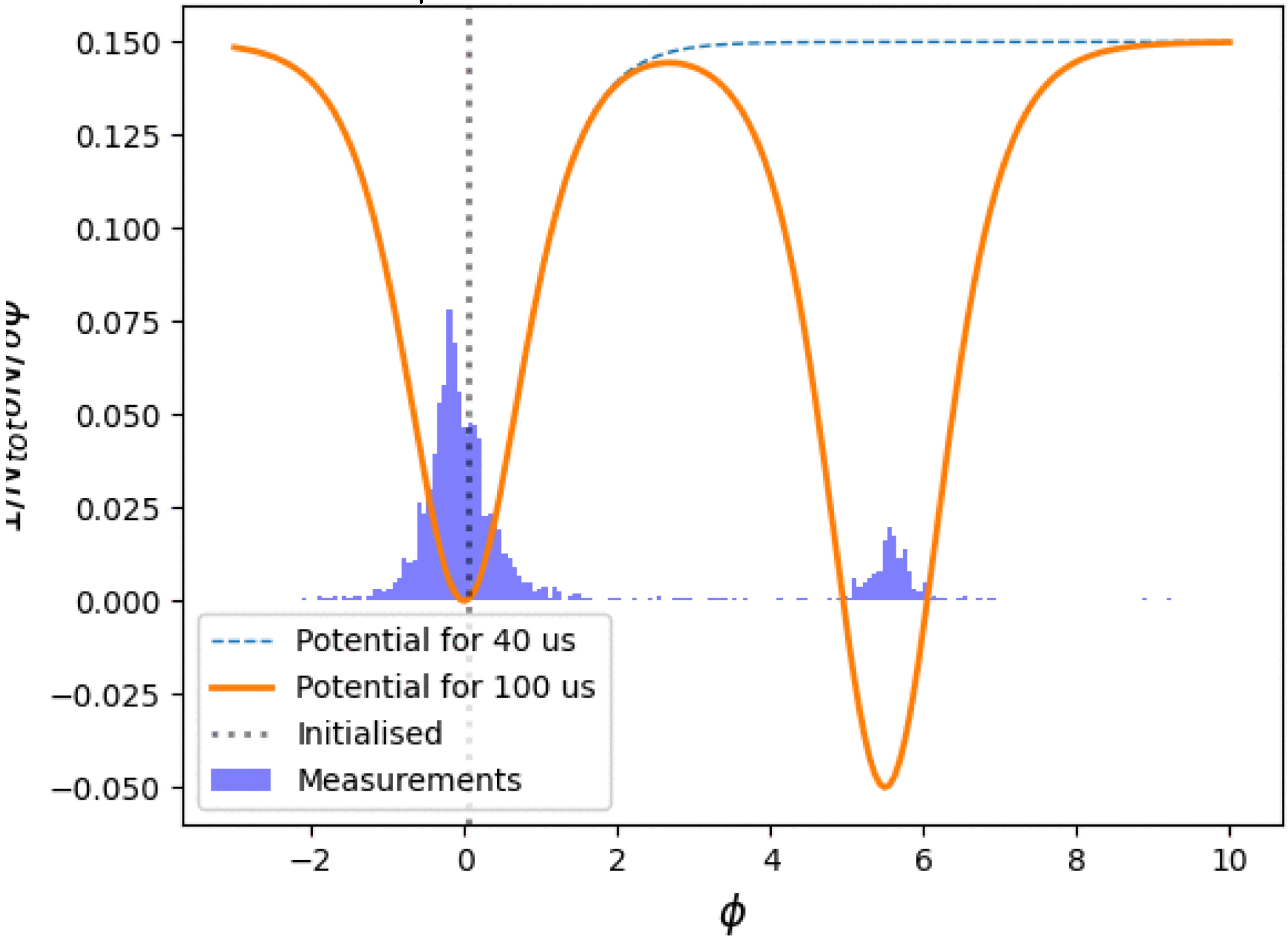
Implemented and executed on D-Wave Q2000 machine





0.6
0.4
0.2
0.0
−0.2
Potential
Potential to settle for 40 us
Potenial after 40 us to tunnel for 100 us
−2
0
2
4
6
8
10
field value ϕ

Implemented and executed on D-Wave Q2000 machine



Calibrating the system with a simple harmonic oscillator

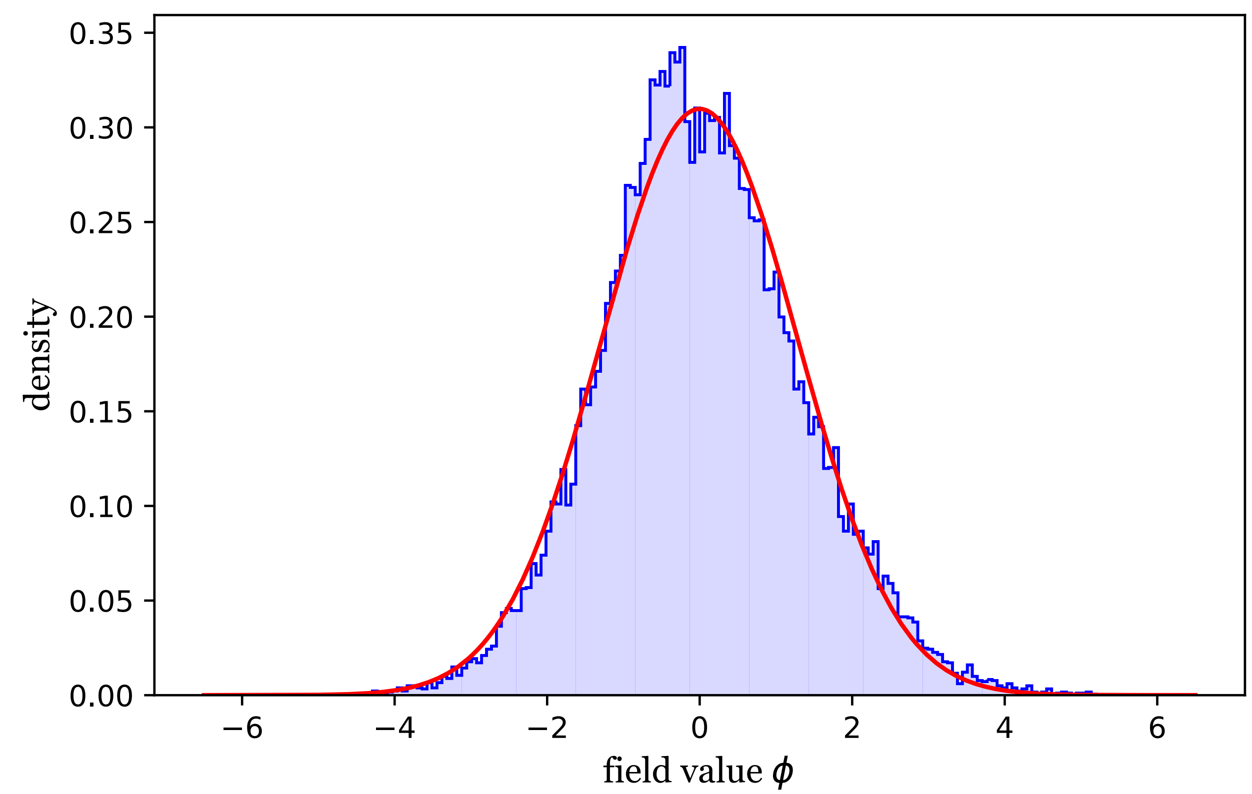
we assume for the potential $U_0(\phi) = \frac{\kappa}{2}\phi^2$ with $\kappa = 0.06$

we initialise classically at $\phi = 0$ and let it settle for $75\ \mu\text{s}$ with $s_q = 0.7$

Fitted with Gaussian

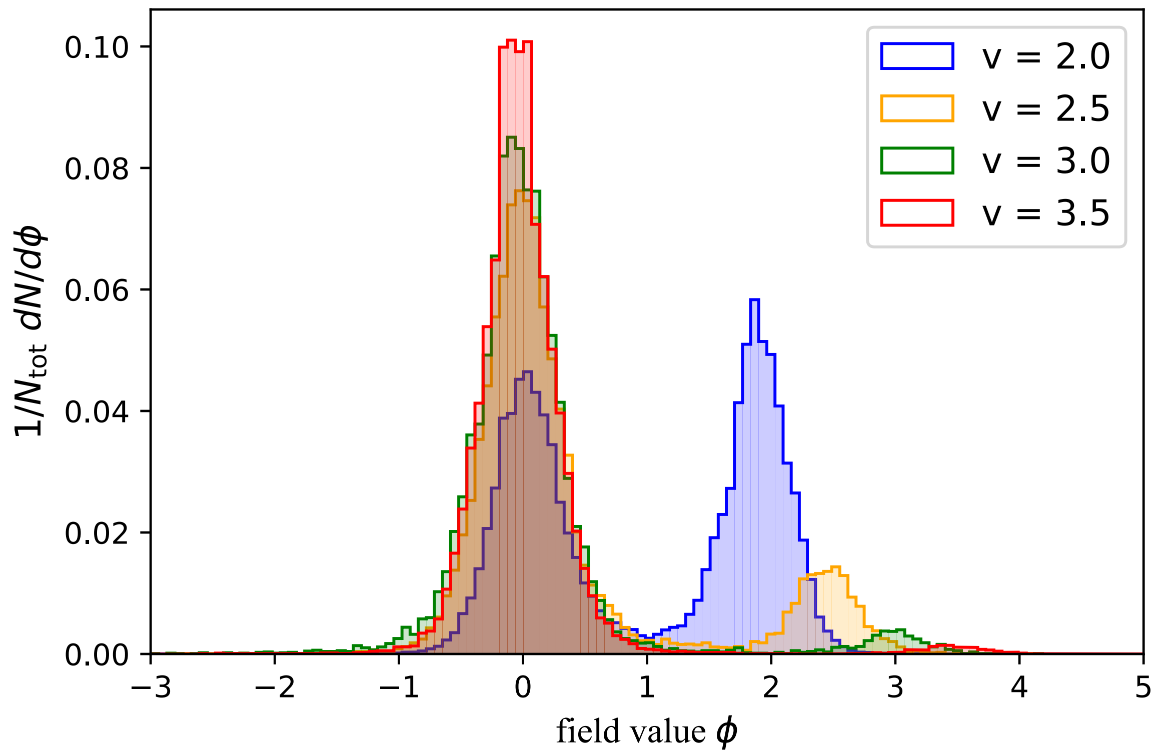
$$|\psi|^2 = \frac{(\kappa/2\gamma)^{\frac{1}{4}}}{\pi^{\frac{1}{2}}} e^{-\sqrt{\kappa/2\gamma}\,\phi^2} \longrightarrow \gamma = 0.33$$

No dynamics was induced by hand on annealer. Thus, constitutes a genuine measurement of the ground state wave function of a quantum mechanical system



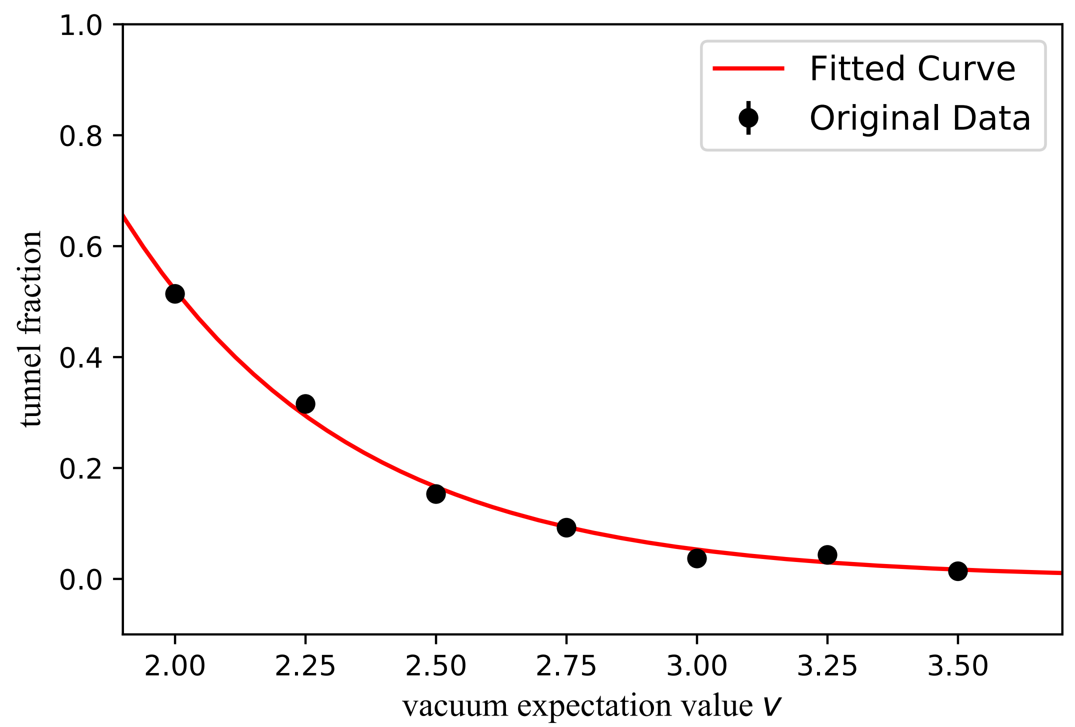
30k shots on D-Wave machine

Results: it decays with v as expected



Perform tunnelling for

$t_{\text{tunnel}} = 100\mu s$ at $s_q = 0.7$



Theory: $\log \Gamma = 3.0 \times (1.66 - v)$

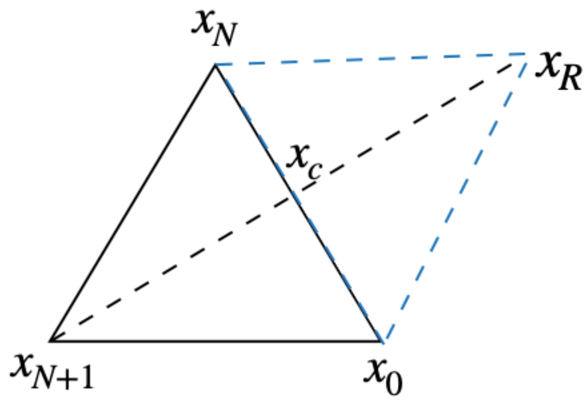
Exp: $\log \Gamma = 2.29 \times (1.71 - v)$

Example 2: Optimisation comparison quantum vs classical

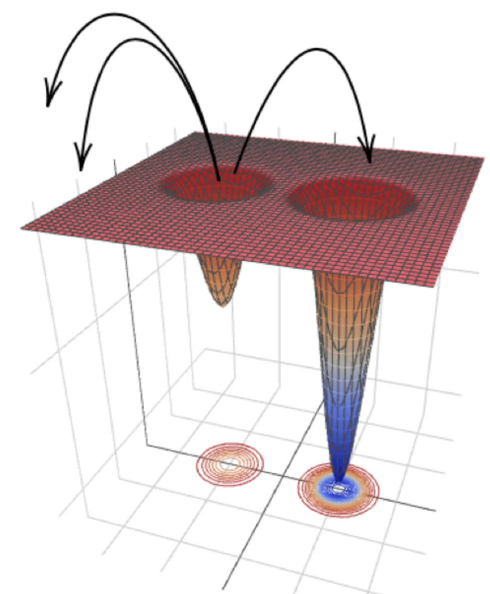
gradient descent

$$x_{i+1} = x_i - \nabla f(x_i)$$

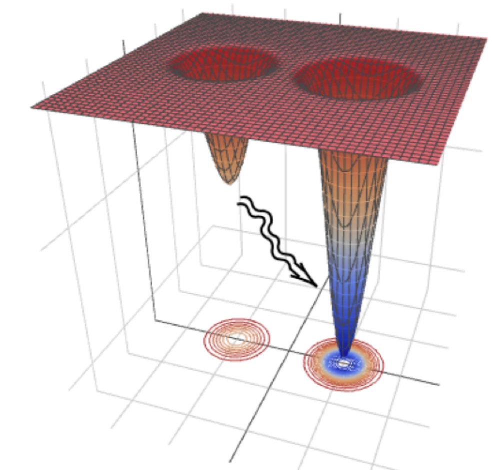
Nelder-Mead



Thermal Annealing

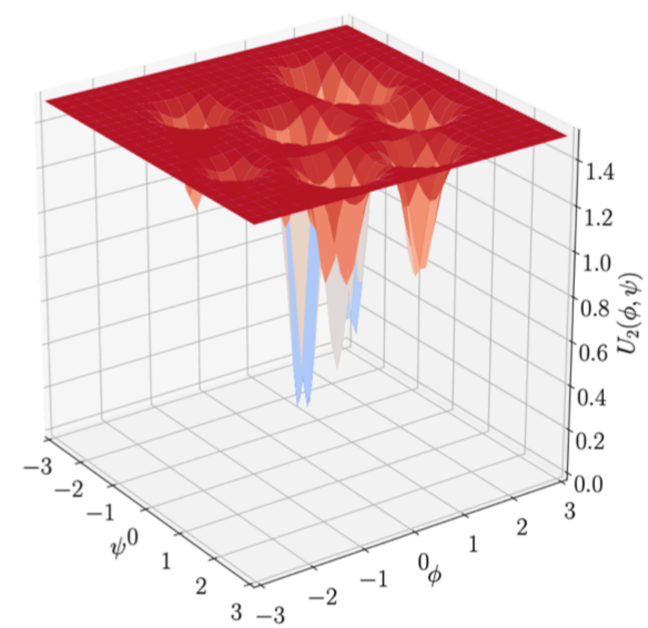


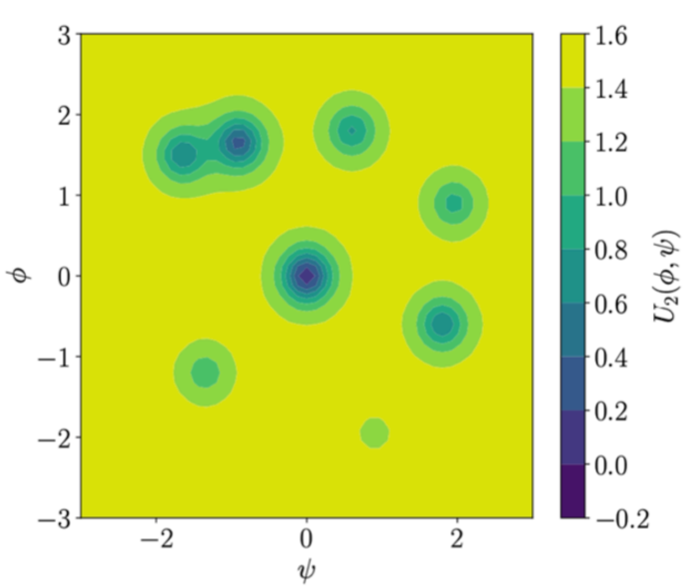
Quantum Annealing



Applied to several examples in [Abel, Blance, MS '21], let's show one here:

Multi-well potential



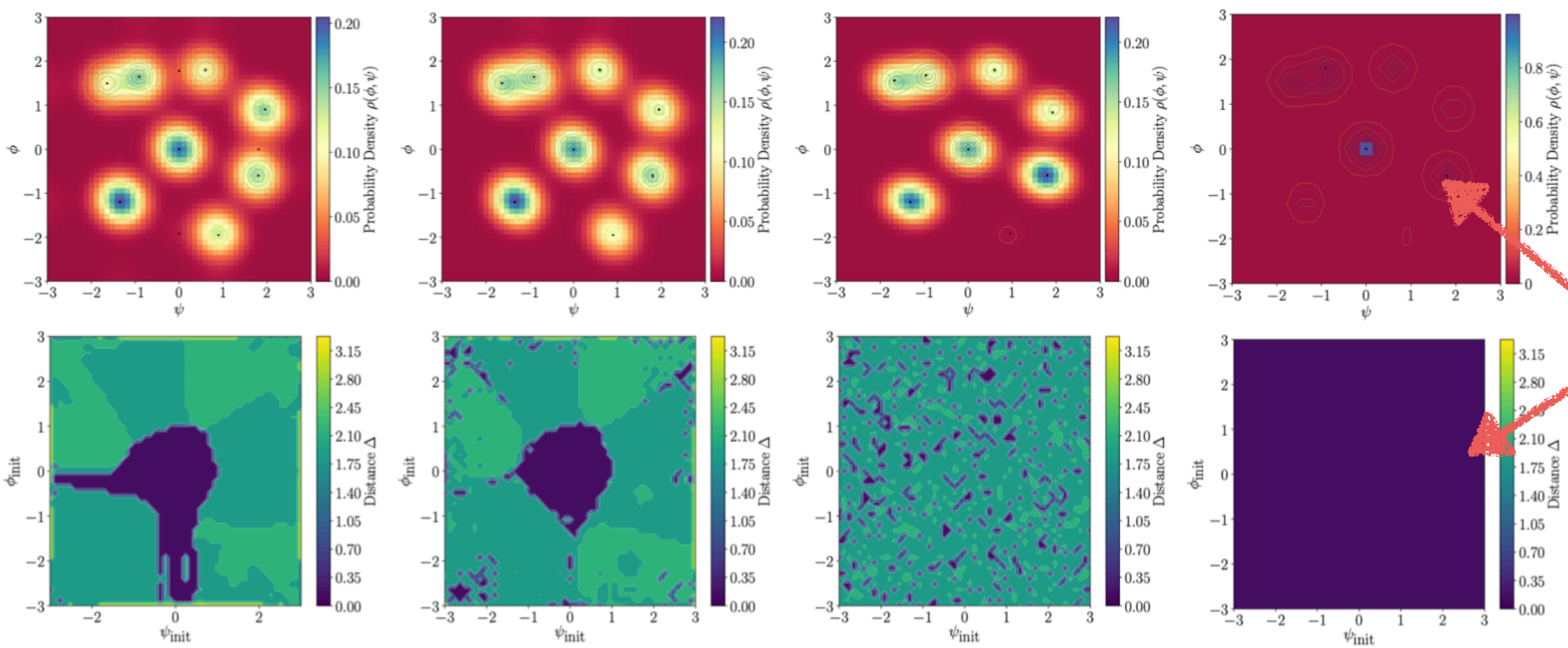


Results for Multi-well potential

[Abel, Blance, MS ’21]

- Quantum algorithms finds global minimum of potential reliably and fast!

Method	Time/run (μs)
Nelder-Mead	4900
Gradient Descent	2900
Thermal Annealing	5×10^5
Quantum Annealing	115



(a) Nelder-Mead (b) Gradient descent (c) Thermal annealing (d) Quantum annealing

Quantum annealer almost never gets stuck in wrong minimum

QA is depth savvy, i.e. works qualitatively different

→ Clear significant quantum advantage

Completely Quantum Neural Networks

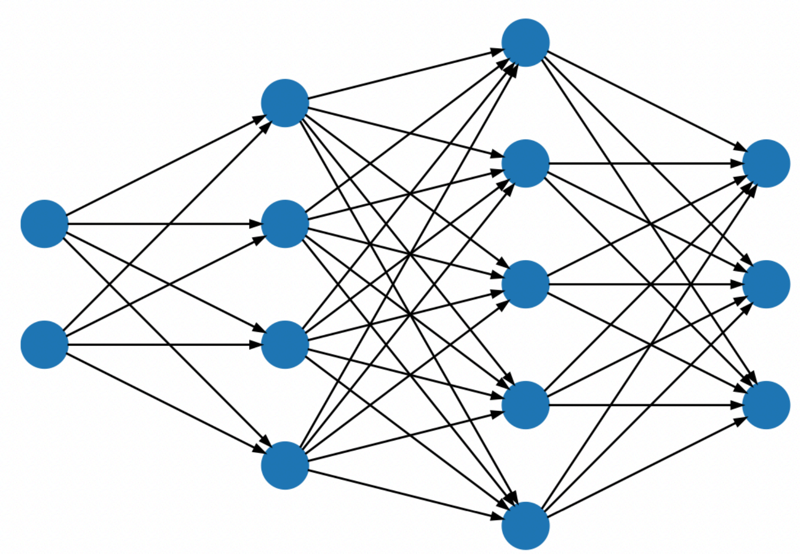
Structure of node i, in layer L $$L_i(x) = g\left(\sum_i w_{ij}x_i + b_i\right)$$

Network output in final layer $$Y = L^{(n)} \circ \ldots \circ L^{(0)}$$

Loss function $$\mathcal{L}(Y) = \frac{1}{N_d}\sum_a |y_a - Y(x_a)|^2$$

[Abel, Criado, MS '22]

- Developed binary encoding of weights (discretised)
- Polynomial approximation of activation function
- Reduction of binary higher-order polynomials into quadratic ones (Ising model)

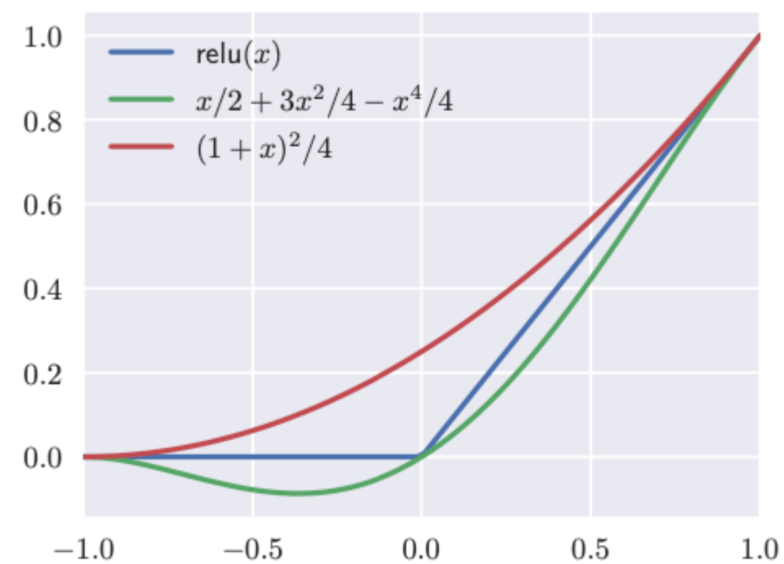


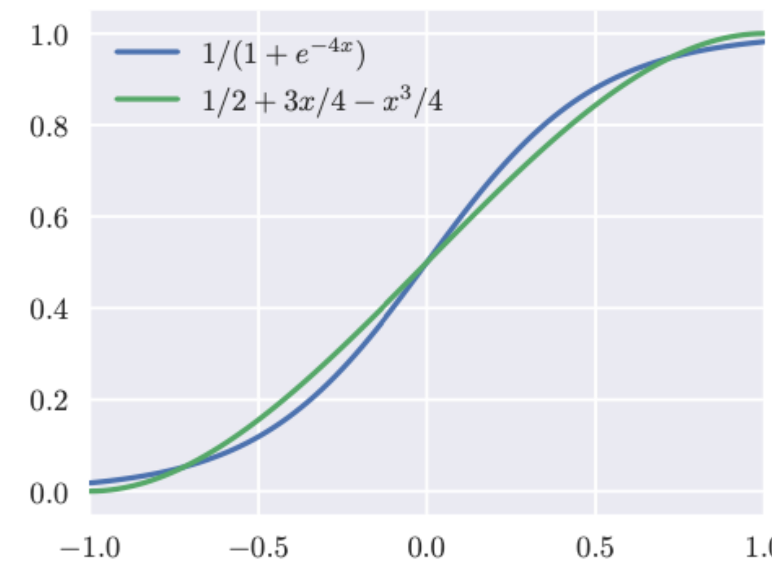
Details about encoding - our approach

Use QUBO encoding to write $\tau_\ell = \frac{1}{2}(\sigma_\ell + 1) \longrightarrow \tau_\ell = 0, 1$

encode weights of NN $p \sim w_{ij}^{(k)},\ b_i^{(k)}$ as binary $p = -1 + \frac{1}{1 - 2^{-N_b}} \sum_{\alpha=0}^{N_b - 1} 2^{-\alpha} \tau_\alpha^p$

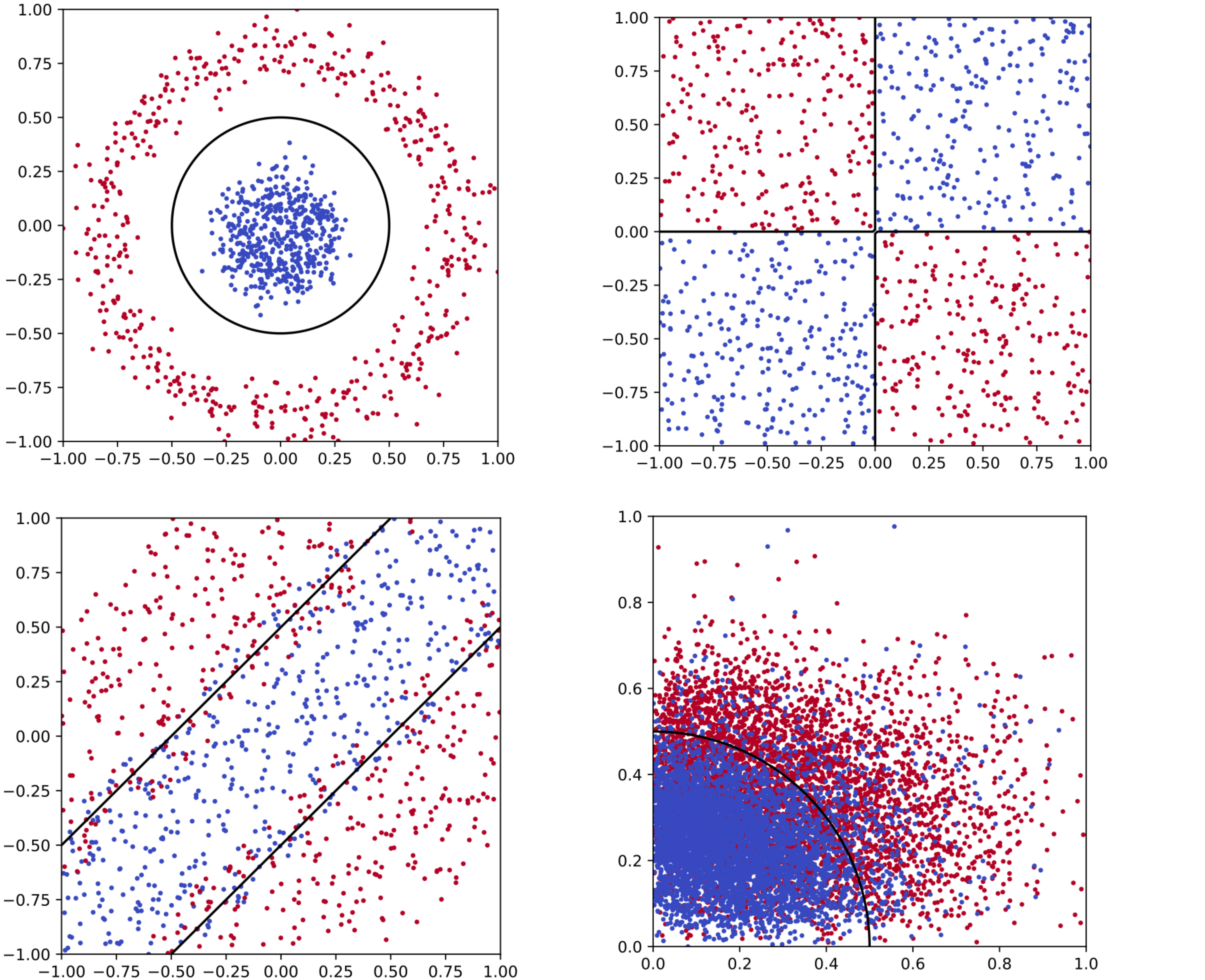
Express activation function as a polynomial:



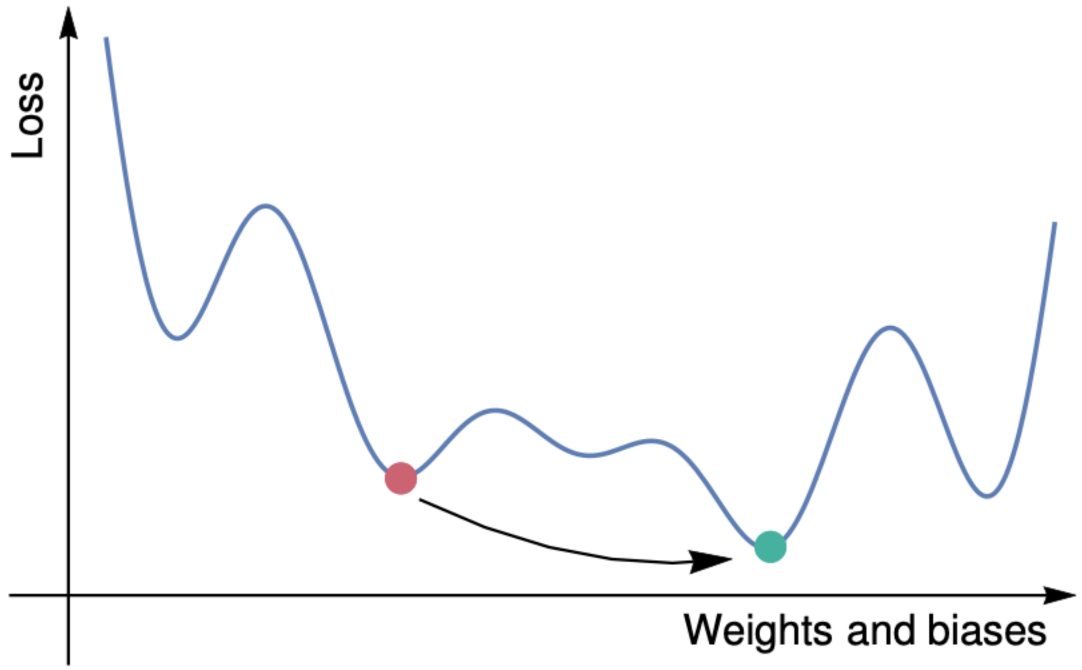


Express loss function using binary-form weights.

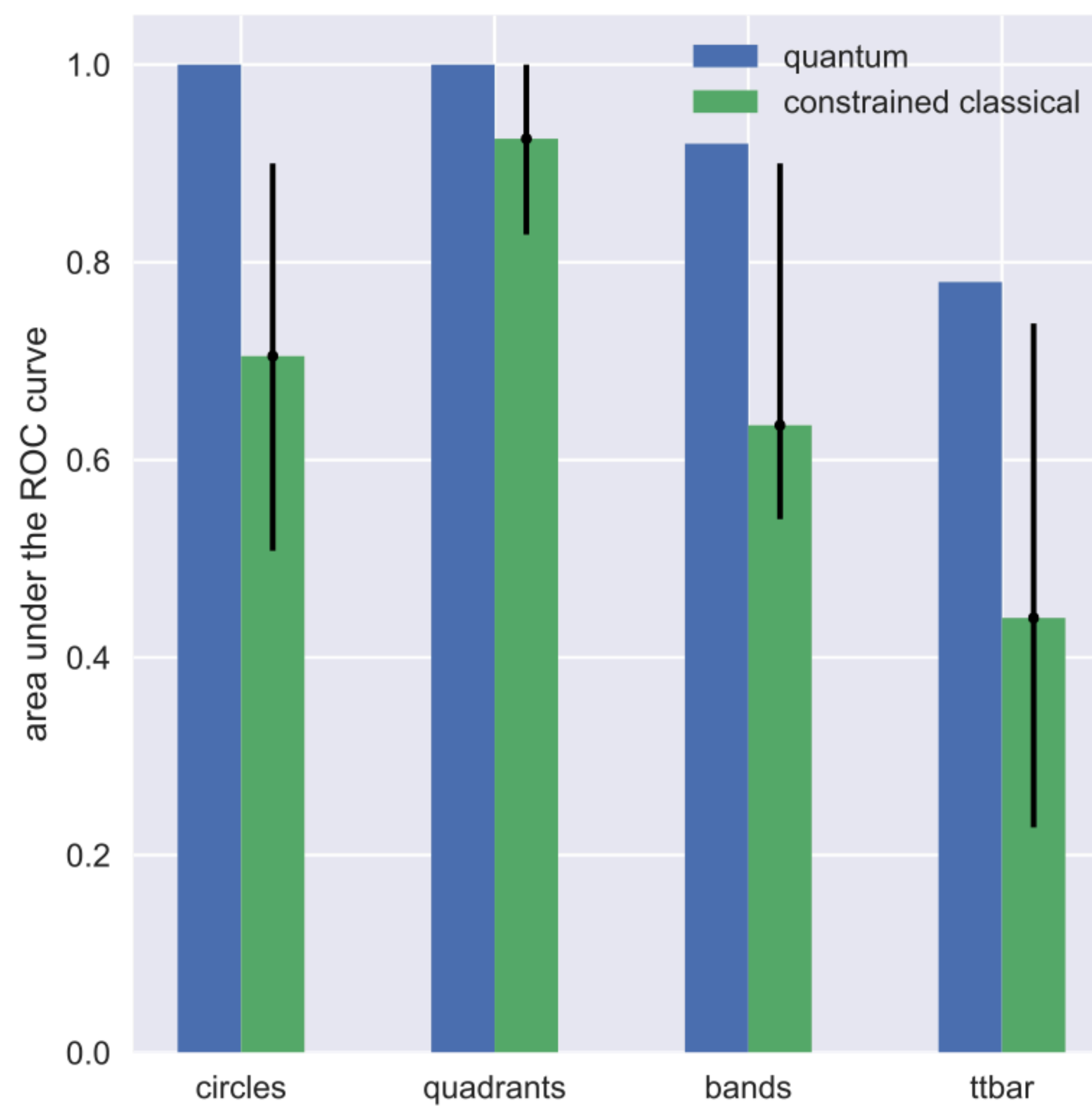
Problem: need to convert to Ising model → quadrature procedure


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50 0.75 1.00
1.0
0.8
0.6
0.4
0.2
0.0
0.0 0.2 0.4 0.6 0.8 1.0

Completely Quantum Neural Networks



Reliable and very fast ground-state finder of loss function



Optimal network training

Application to differential equations and variational methods

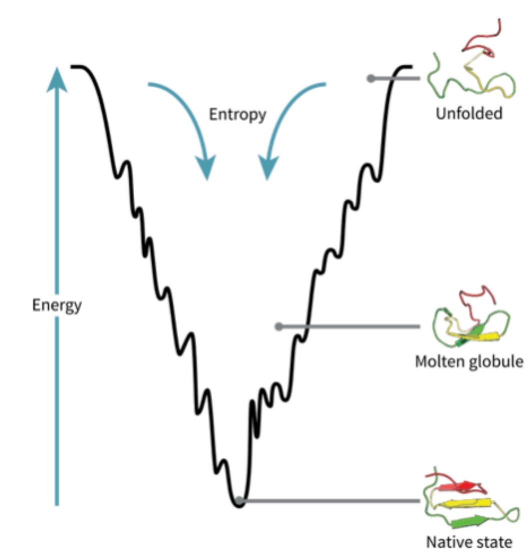
Define your mathematical task as an **optimisation problem**

$$\mathcal{F}_m(\vec{x}, \phi_m(\vec{x}), \nabla\phi_m(\vec{x}), \cdots, \nabla^j\phi_m(\vec{x})) = 0$$

Build the full function, here a DE into the loss function, incl boundary conditions

$$\mathcal{L}(\{w,\vec{b}\}) = \frac{1}{i_{\max}} \sum_{i,m} \hat{\mathcal{F}}_m(\vec{x}^i, \hat{\phi}_m(\vec{x}^i), \cdots, \nabla^j\hat{\phi}_m(\vec{x}^i))^2 + \sum_{\text{B.C.}} (\nabla^p\hat{\phi}_m(\vec{x}_b) - K(\vec{x}_b))^2 \ ,$$

[Piscopo, MS, Waite ’19]



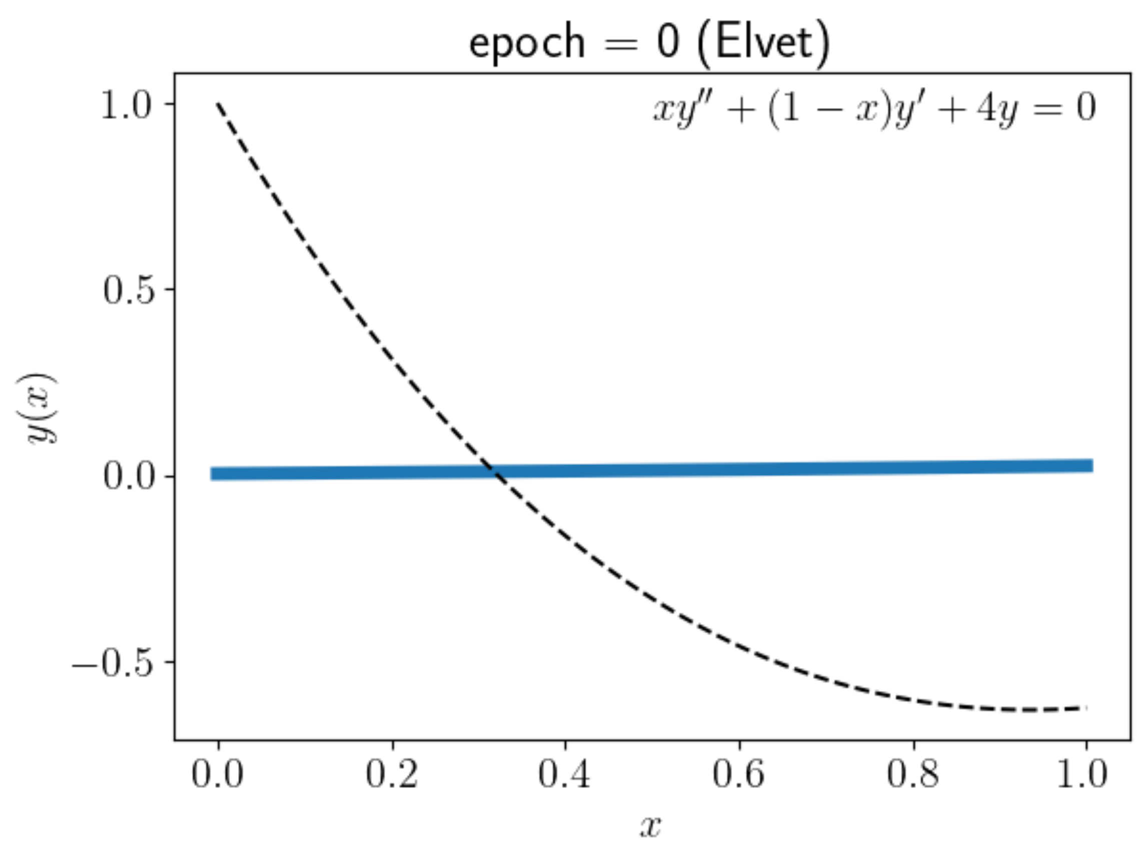
identify trial solution with network output $\hat{\phi}_m(\vec{x}) \equiv \check{N}_m(\vec{x}, \{w, \vec{b}\})$

QADE: Solving differential equations with a quantum annealer

[Criado, MS '22]

Example Laguerre differential equation:

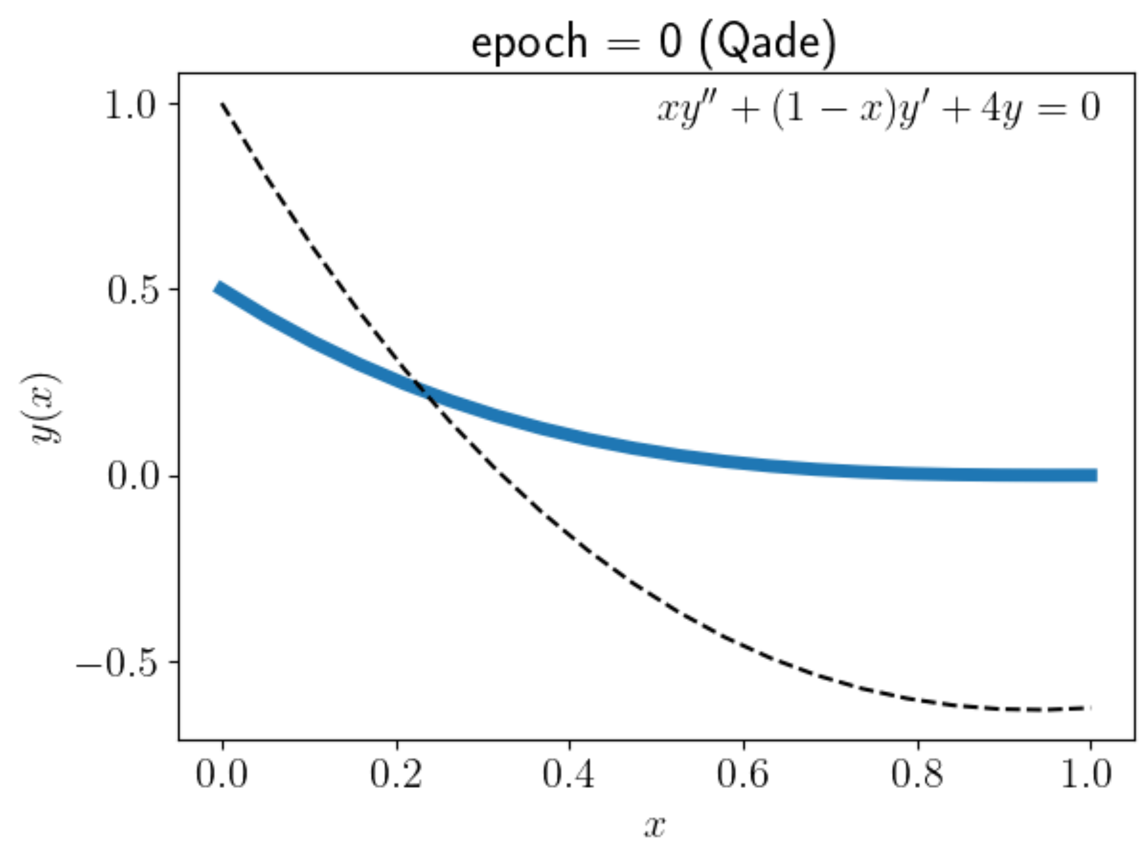
$$xy'' + (1-x)y' + 4y = 0 \quad \text{with} \quad y(0) = 1 \quad \text{and} \quad y(1) = L_4(1)$$



Classical Neural Network

[Piscopo, MS, Waite '19] [Araz, Criado, MS '21]

https://gitlab.com/elvet/elvet



Quantum algorithm

http://gitlab.com/jccriado/qade

QFitter

Example Higgs EFT fit:

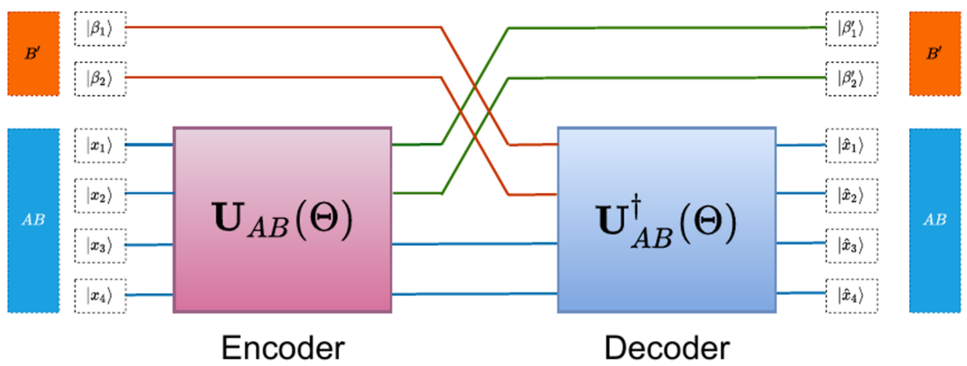
[Criado, Kogler, MS '22]

$$\mathcal{L} = \frac{c_{u3} y_t}{v^2}(\phi^\dagger\phi)(\bar{q}_L\tilde{\phi}u_R) + \frac{c_{d3} y_b}{v^2}(\phi^\dagger\phi)(\bar{q}_L\phi d_R)$$
$$+ \frac{i c_W g}{2m_W^2}(\phi^\dagger\sigma^a D^\mu\phi)D^\nu W^a_{\mu\nu} + \frac{c_H}{4v^2}\left(\partial_\mu(\phi^\dagger\phi)\right)^2$$
$$+ \frac{c_\gamma (g')^2}{2m_W^2}(\phi^\dagger\phi)B_{\mu\nu}B^{\mu\nu} + \frac{c_g g_S^2}{2m_W^2}(\phi^\dagger\phi)G^a_{\mu\nu}G^{a\mu\nu}$$
$$+ \frac{i c_{HW} g}{4m_W^2}(\phi^\dagger\sigma^a D^\mu\phi)D^\nu W^a_{\mu\nu}$$
$$+ \frac{i c_{HB} g'}{4m_W^2}(\phi^\dagger D^\mu\phi)D^\nu B_{\mu\nu} + \text{h.c.}$$

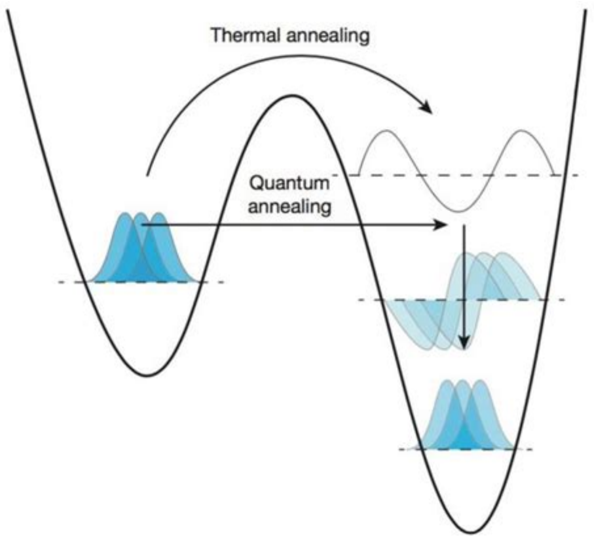
$$\chi^2 = \sum_{ij} V_a C^{-1}_{ab} V_b \qquad V_a = O_a^{(\text{exp})} - O_a^{(\text{th})}(c)$$

- Fast and reliable state-of-the-art Higgs, ELW, ... fits
- Convergence no problem for non-convex $\Delta\chi^2 = \chi^2 - \chi^2_{\text{min}}$ functions

Formulation	Method	Fit time	c_{HW}	c_H	c_g	c_γ	χ^2
Standard	Minuit (initial $c_{HW} = 0$)	2.0 s	-0.009	0.100	1.4×10^{-5}	3.2×10^{-6}	4110
	Minuit (initial $c_{HW} = -0.05$)	2.4 s	-0.050	0.039	-9.7×10^{-6}	-1.0×10^{-4}	135
	Simulated annealing (initial $c_{HW} = 0$)	642 s	-0.009	0.100	1.4×10^{-5}	3.7×10^{-6}	4110
	Simulated annealing (initial $c_{HW} = -0.05$)	644 s	-0.009	0.100	1.4×10^{-5}	3.7×10^{-6}	4110
QUBO	Simulated annealing (Class A)	6.4 s	-0.012	-0.054	-3.0×10^{-5}	3.9×10^{-5}	3910
	Simulated annealing (Class B)	6.4 s	-0.045	-0.175	-3.7×10^{-5}	1.8×10^{-4}	228
	Quantum annealing	0.2 s	-0.047	-0.050	1.9×10^{-5}	7.5×10^{-7}	68



Summary



- Quantum Machine Learning is exciting research area that rapidly expands, supported through private and public sector. Many algorithms to be invented.

- Quantum Machine Learning often shows an improved performance over classical Machine Learning, when limiting to a similar complexity of the model.
  Can exploit QM prop: entanglement, superposition principle and tunnelling

- For more exciting applications (quantum advantage), need development of technical realisation of quantum computers
  (size, fault tolerance, coherence, operations,...)