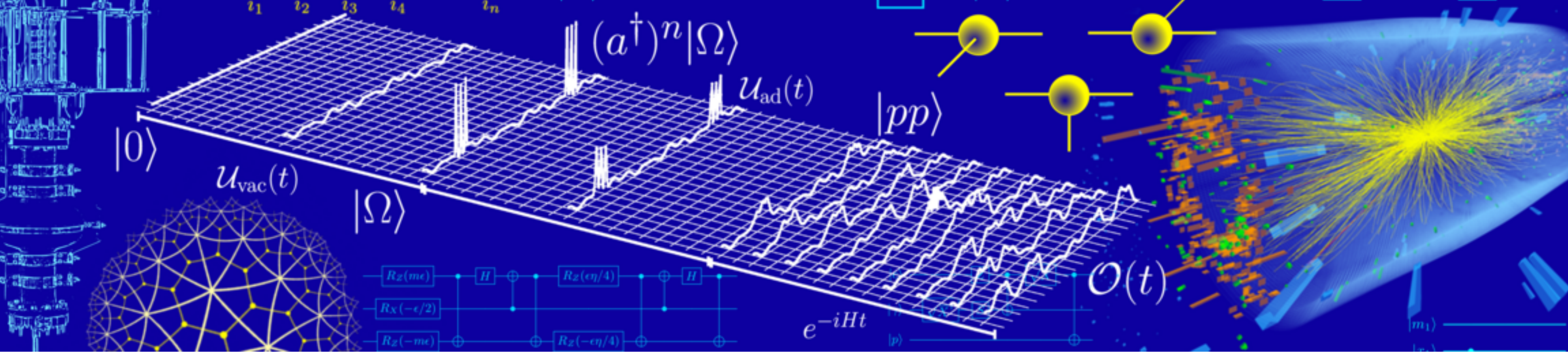


1 Particle - 1 Qubit

- Particle Physics Data Encoding for QML -

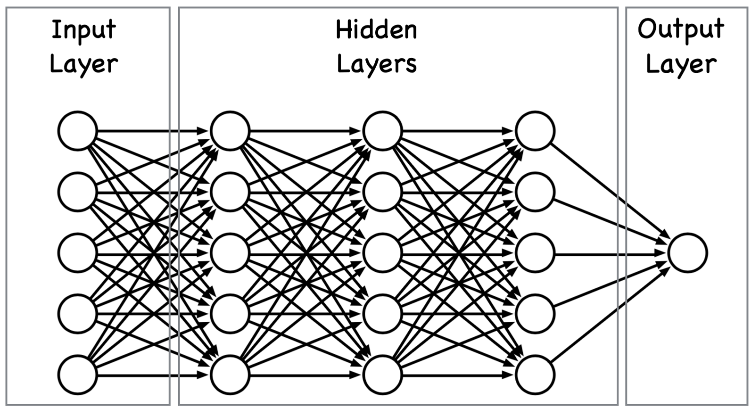
Michael Spannowsky

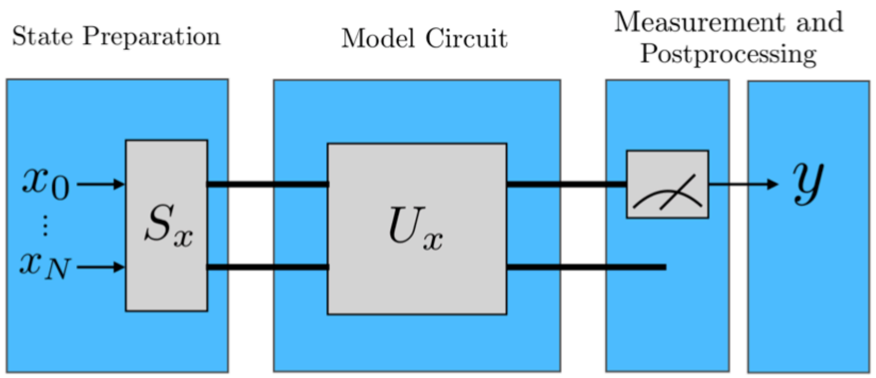
IPPP, Durham University

Classical
ML Algorithms

Quantum
ML Algorithms

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





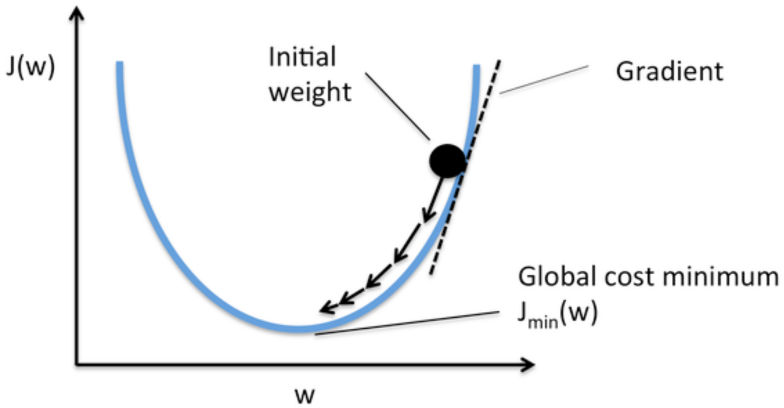
2. the calculation of a loss function used to define the task the method

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

ground state

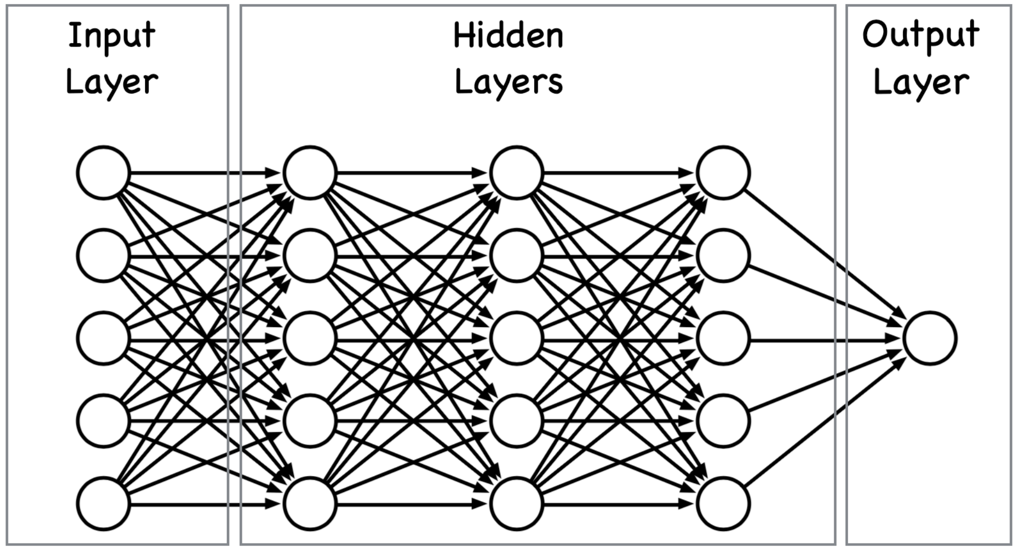
$$|\Gamma\rangle := \arg\min_{|\psi\rangle \in \mathcal{D}} \frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}$$

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



quantum: annealing

hybrid: classical opti.

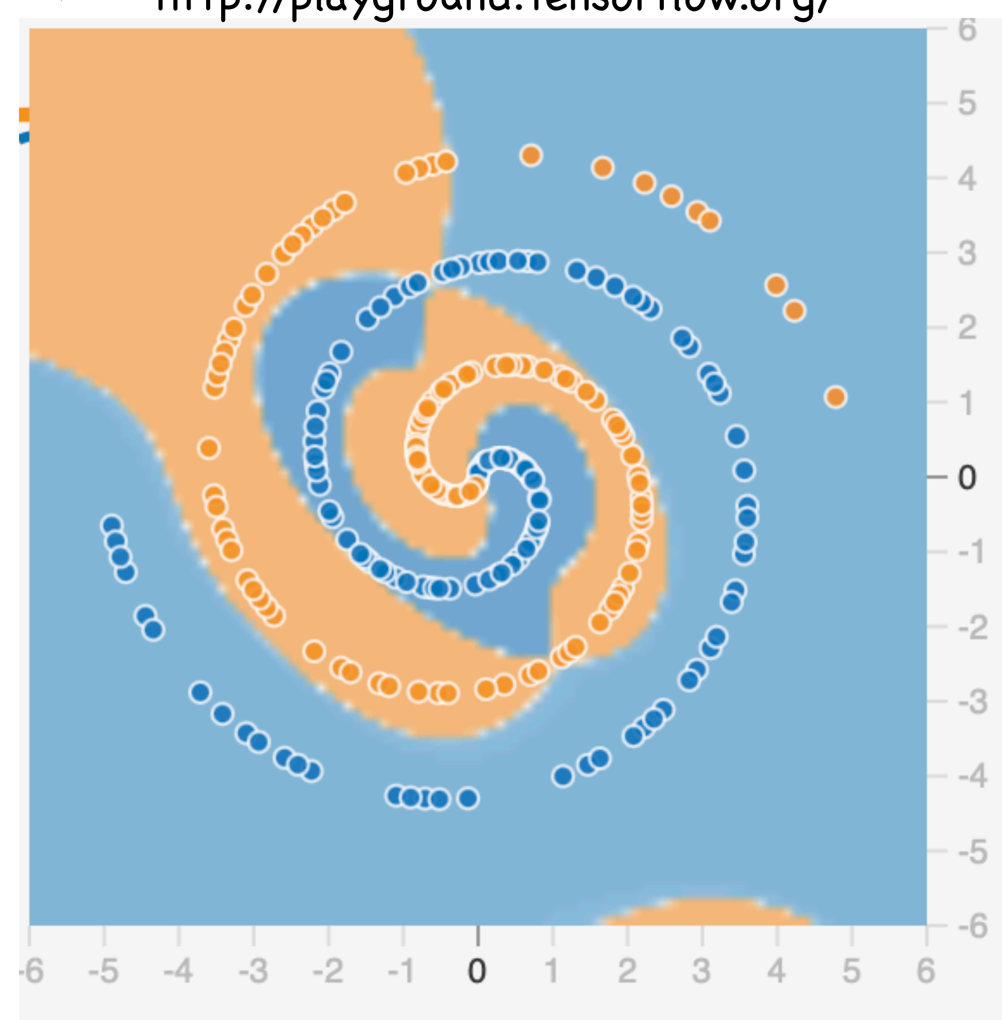


After training of NN

-> inference (apply NN to test samples)

NN output is score for for event to look like signal (1) or background (0)

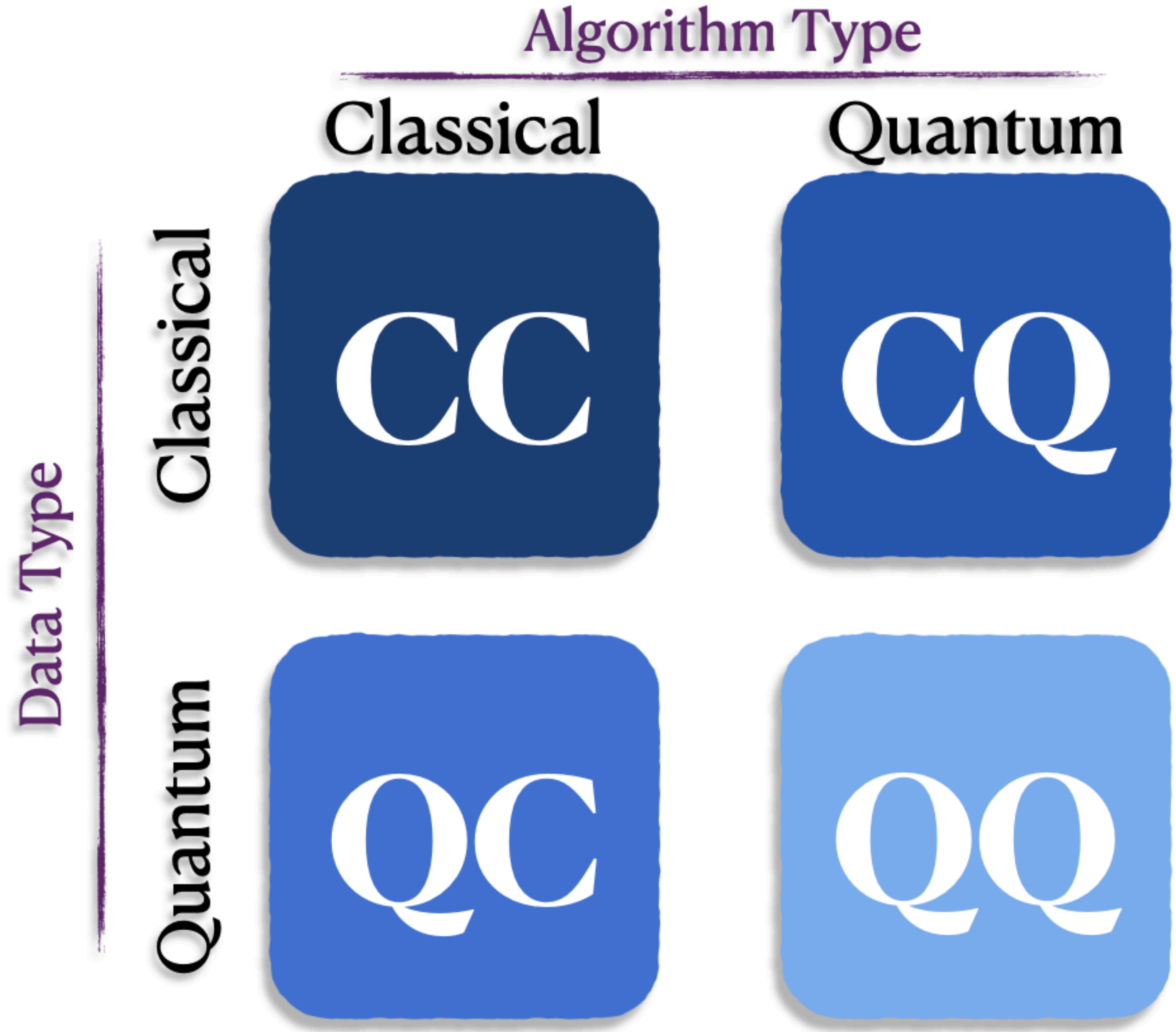
http://playground.tensorflow.org/



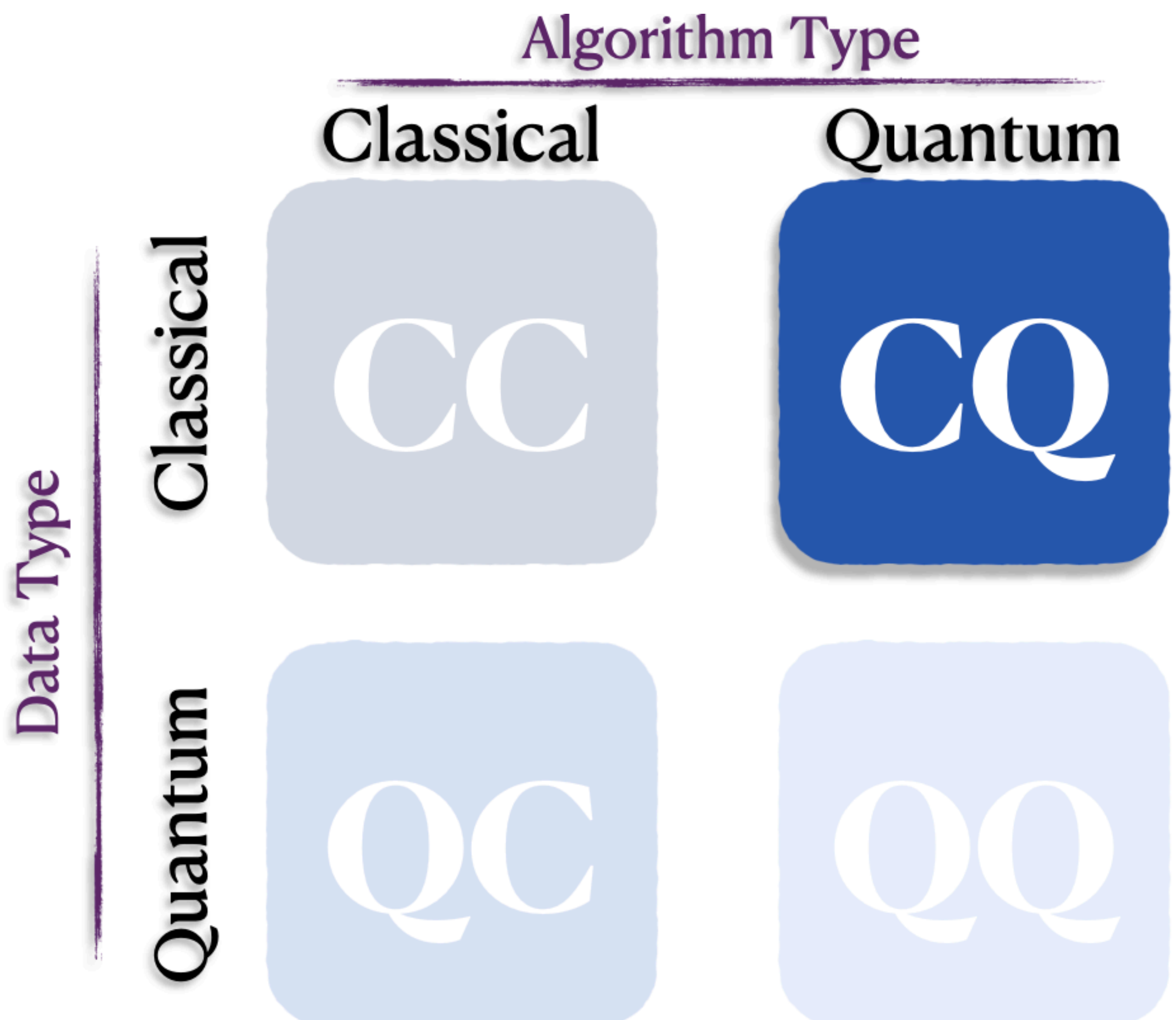
Expressivity and **generalisability** crucial for optimal decision boundaries

defining factors are:

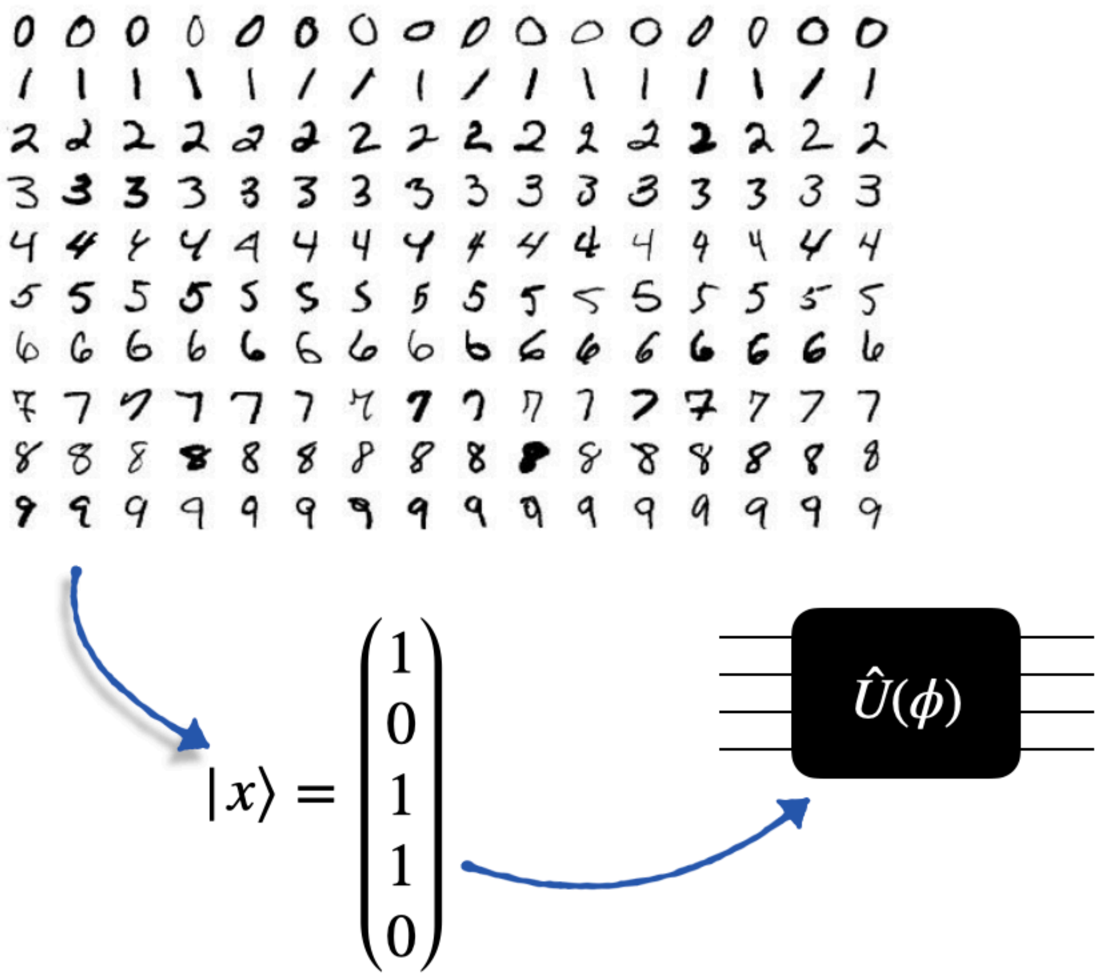
- Width
- Depth
- Activation functions
- Loss function


Algorithm Type
Classical
Quantum
Data Type
Classical
Quantum
CC
CQ
QC
QQ

Most interesting case for QML applications in collider experiments



Classical data processed via quantum algorithms on quantum devices



Encoding

Data → State of a quantum system

basis encoding of binary string $(1,0)$, i.e. representing integer 2

Hamiltonian encoding of a matrix A

$$|\psi\rangle = \underbrace{\alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle}$$

$$U = e^{-iH_A t}$$

amplitude encoding of unit-length complex vector $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$

time-evolution encoding of a scalar t

data encoding in different parts of the state and operator description

angle encoding for $H_A \in \{X, Y, Z\}$ and $t = x$ (feature)

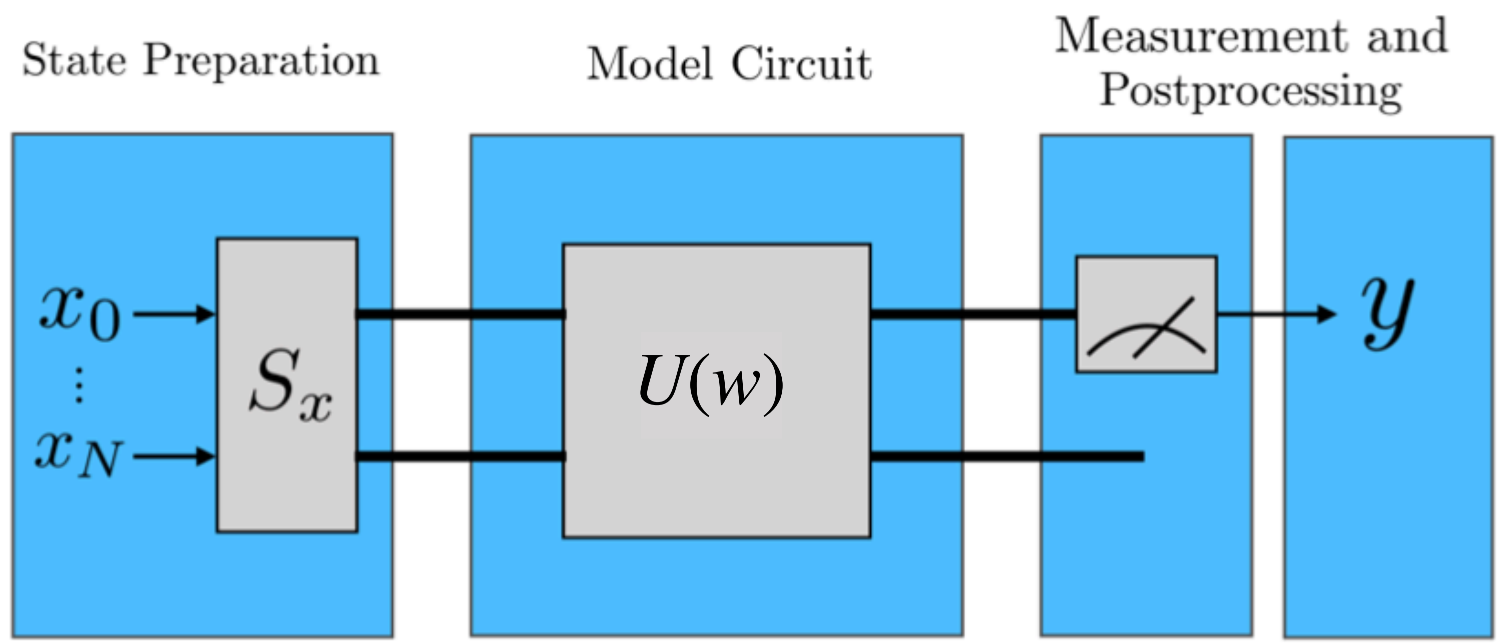
Quantum Machine Learning with a Variational Quantum Circuit

[McClean et al '16]

[Farhi, Neven '18]

[Schuld et al '20]

[Blance, MS '20]



Encoding step

Ports data into quantum state

Linear Unitary Operation

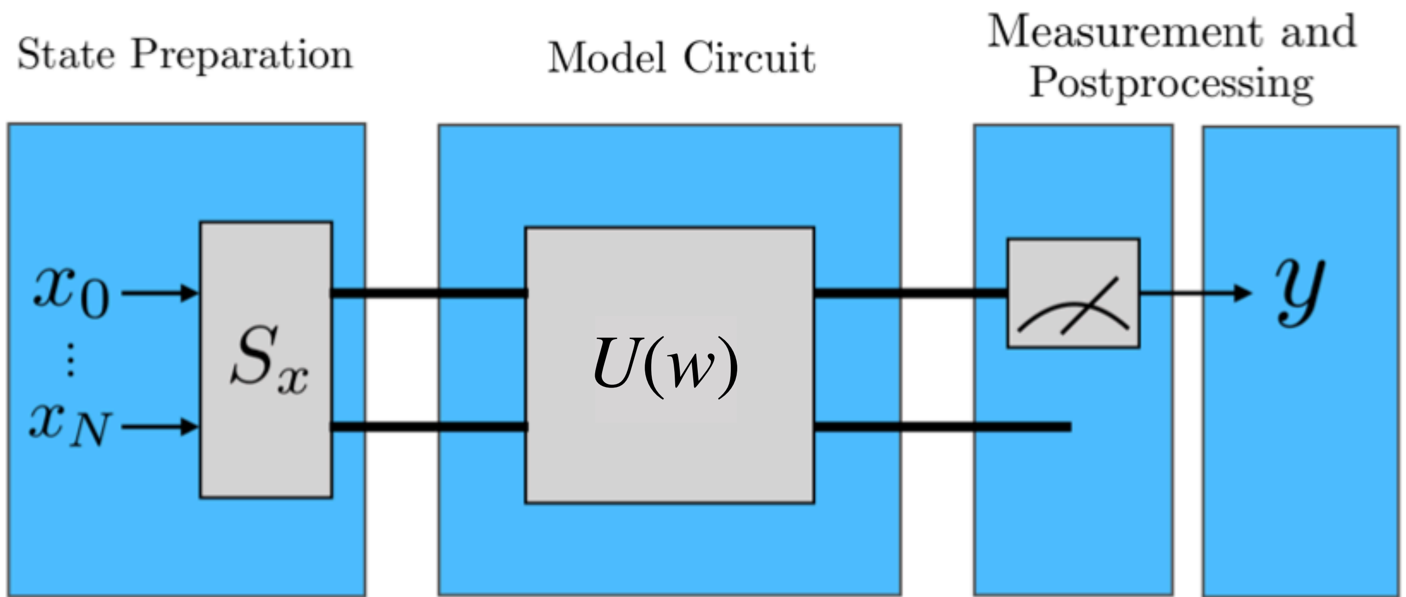
realises entanglement etc

Measurement with respect to operator, e.g. σ_z

expectation value of operator <-> many shots

difference to classical NN - expressibility relies on encoding step

Quantum Machine Learning with a Variational Quantum Circuit



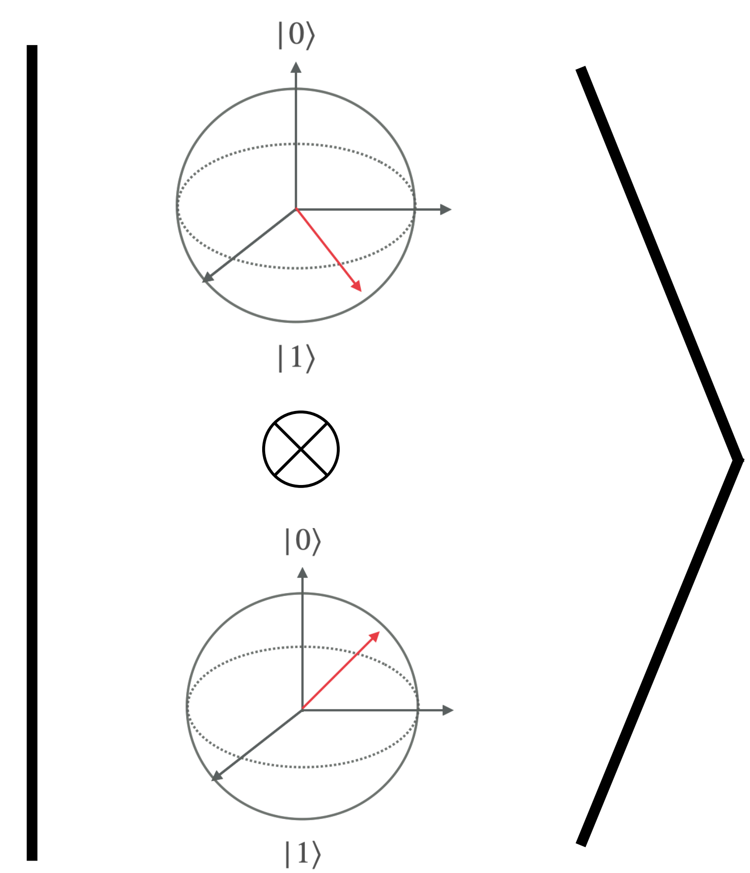
state preparation

n corresponds to # features

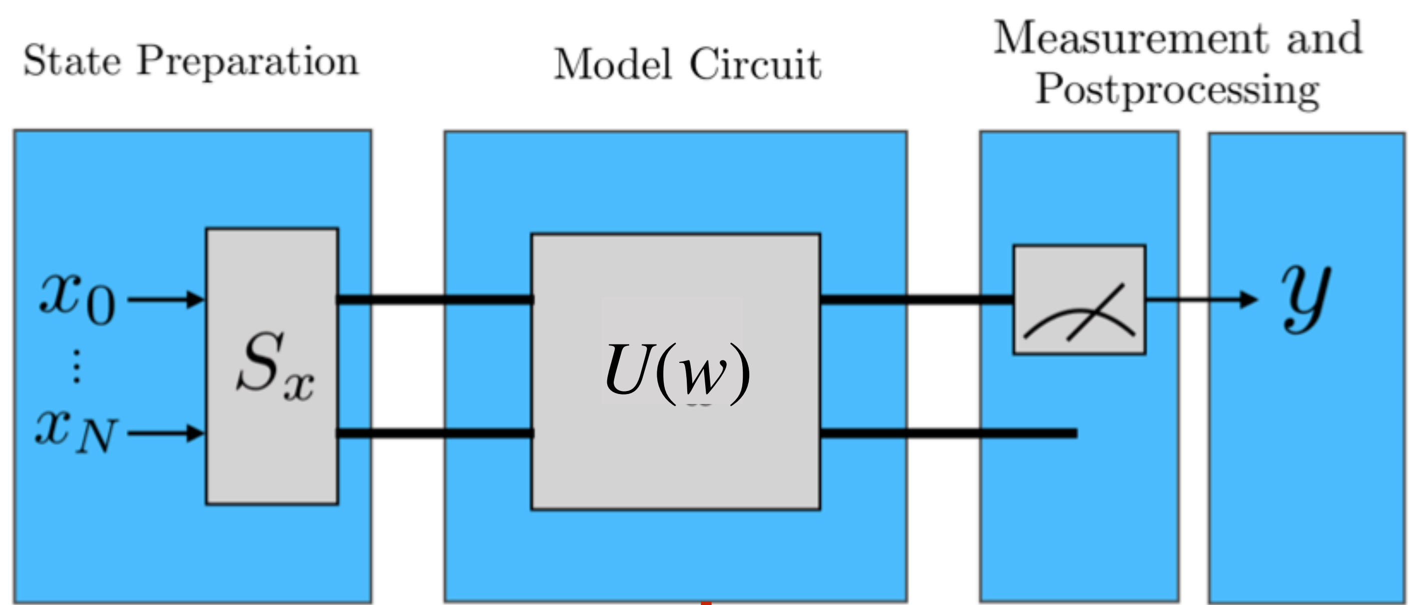
$$x \mapsto S_x|\phi\rangle = S_x|0\rangle^{\otimes n} = |x\rangle$$

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



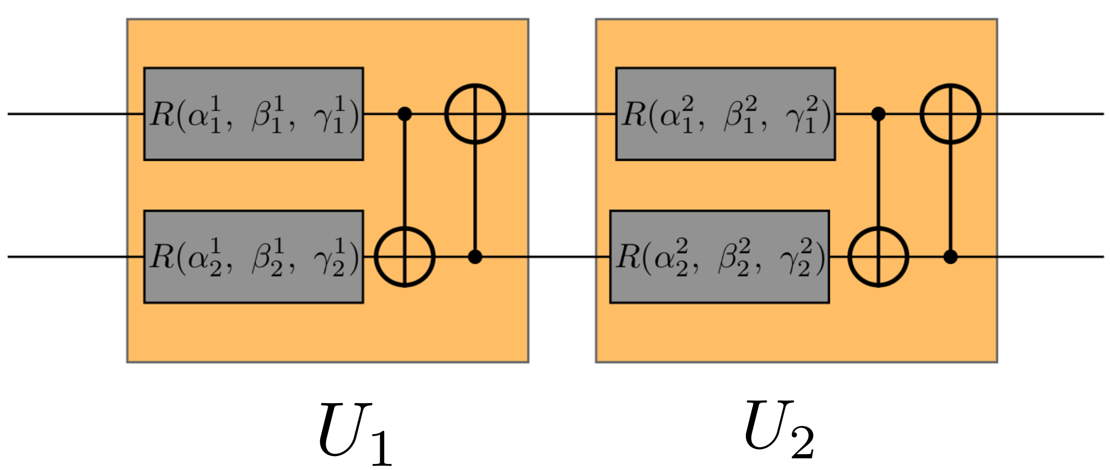
$$|\psi\rangle = U(w)|x\rangle \quad \text{with} \quad U(w) = U_{l_{\max}}(w_{l_{\max}}) \ldots U_l(w_l) \ldots U_1(w_1)$$

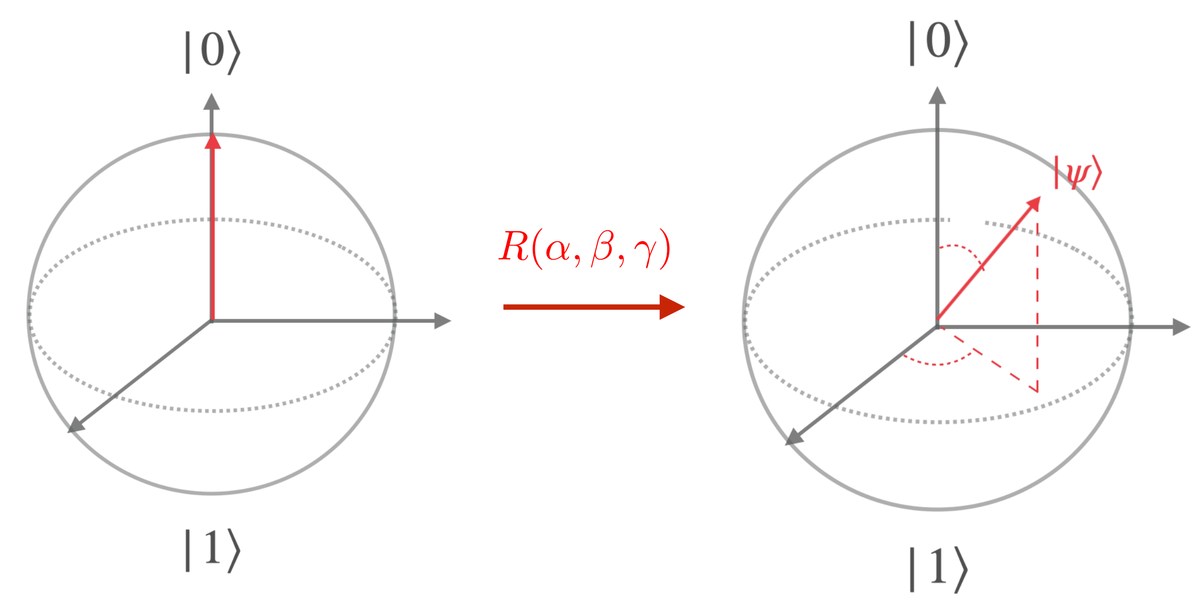
model circuit

trainable parameters

prepared state

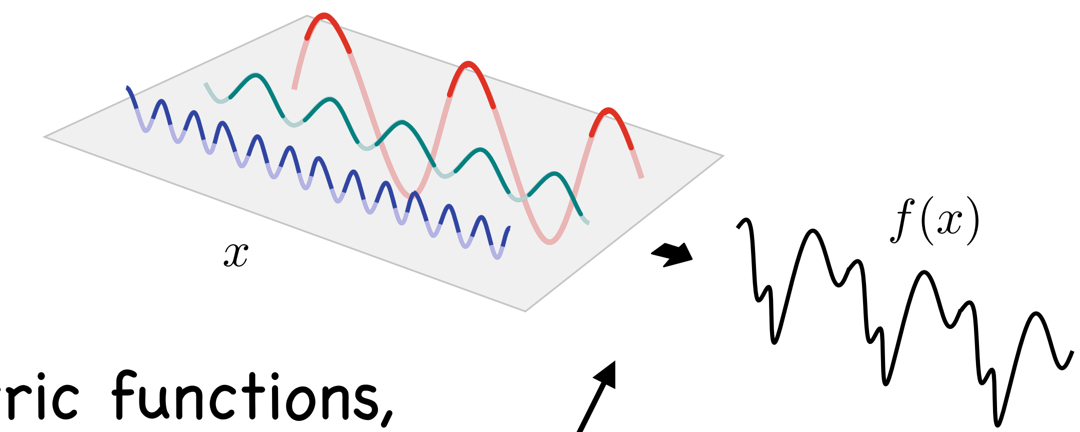
2-layer Variational Quantum Circuit





Rotation + CNOT -> Entanglement

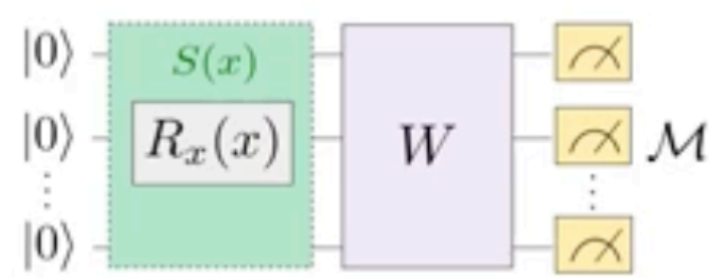
Expressibility of model and encoding



- Most encodings result in sum of trigonometric functions, e.g. angle encoding, time evolution encoding

- Fourier series is universal approximator, but for many encoding strategies quantum models are linear combinations of functions composed of few frequencies

- Pendant to activation functions in the encoding step.



- Encoding + W operator give functional form

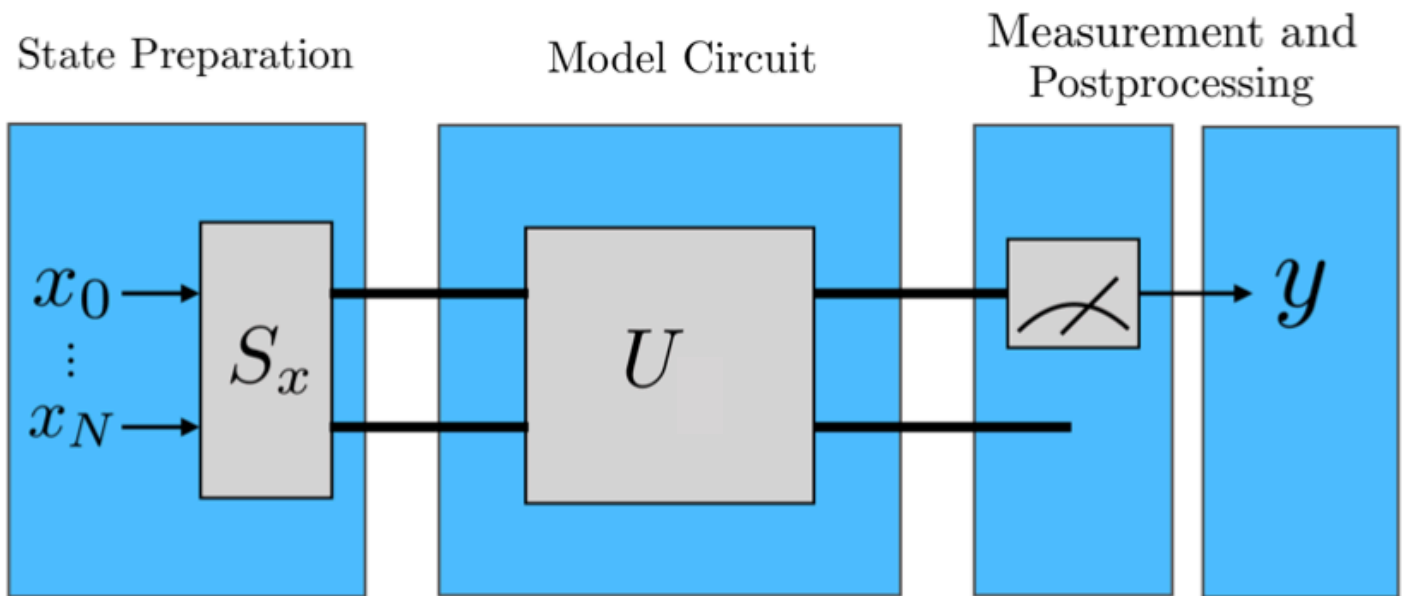
$$f_\theta(x) = \langle \mathscr{M} \rangle_{x,\theta} = A + B\cos(x) - C\sin(x)$$

- Data reuploading can increase expressivity

A, B, C coefficients from parametrised circuit W

trigonometric structure from data encoding

Quantum Machine Learning
with a Variational Quantum Circuit



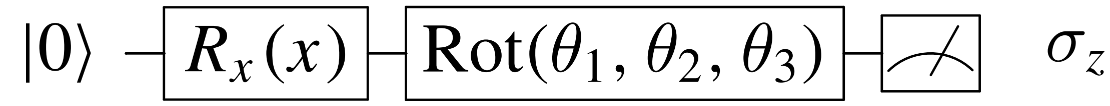
- Entangled state shares information across qubits
- Evaluate expectation value of qubits to construct loss

  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)

Simple example:



gives the model output $f_\theta(x) = \langle 0 | R_x(x)^\dagger \mathrm{Rot}(\theta_1, \theta_2, \theta_3)^\dagger \sigma_z \mathrm{Rot}(\theta_1, \theta_2, \theta_3) R_x(x) | 0 \rangle$

data encoding

$$|\phi(x)\rangle = R_x(x)|0\rangle = \begin{pmatrix} \cos(\frac{x}{2}) & -i\sin(\frac{x}{2}) \\ -i\sin(\frac{x}{2}) & \cos(\frac{x}{2}) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos(\frac{x}{2}) \\ -i\sin(\frac{x}{2}) \end{pmatrix}$$

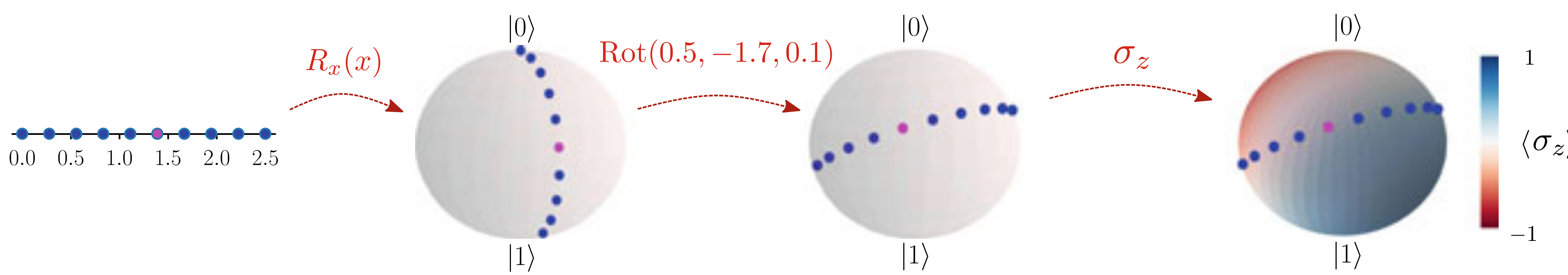
parametrised rotation

$$\begin{aligned} |\psi(x,\theta)\rangle &= \mathrm{Rot}(\theta_1, \theta_2, \theta_3)|\phi(x)\rangle \\ &= \begin{pmatrix} e^{i(-\frac{\theta_1}{2}-\frac{\theta_3}{2})}\cos(\frac{\theta_2}{2})\cos(\frac{x}{2}) + i\,e^{i(-\frac{\theta_1}{2}+\frac{\theta_3}{2})}\sin(\frac{\theta_2}{2})\sin(\frac{x}{2}) \\ e^{i(\frac{\theta_1}{2}-\frac{\theta_3}{2})}\sin(\frac{\theta_2}{2})\cos(\frac{x}{2}) - i\,e^{i(\frac{\theta_1}{2}+\frac{\theta_3}{2})}\cos(\frac{\theta_2}{2})\sin(\frac{x}{2}) \end{pmatrix} \end{aligned}$$

model output

$$f_\theta(x) = \langle \psi(x,\theta)|\sigma_z|\psi(x,\theta)\rangle = \cos(\theta_2)\cos(x) - \sin(\theta_1)\sin(\theta_2)\sin(x)$$

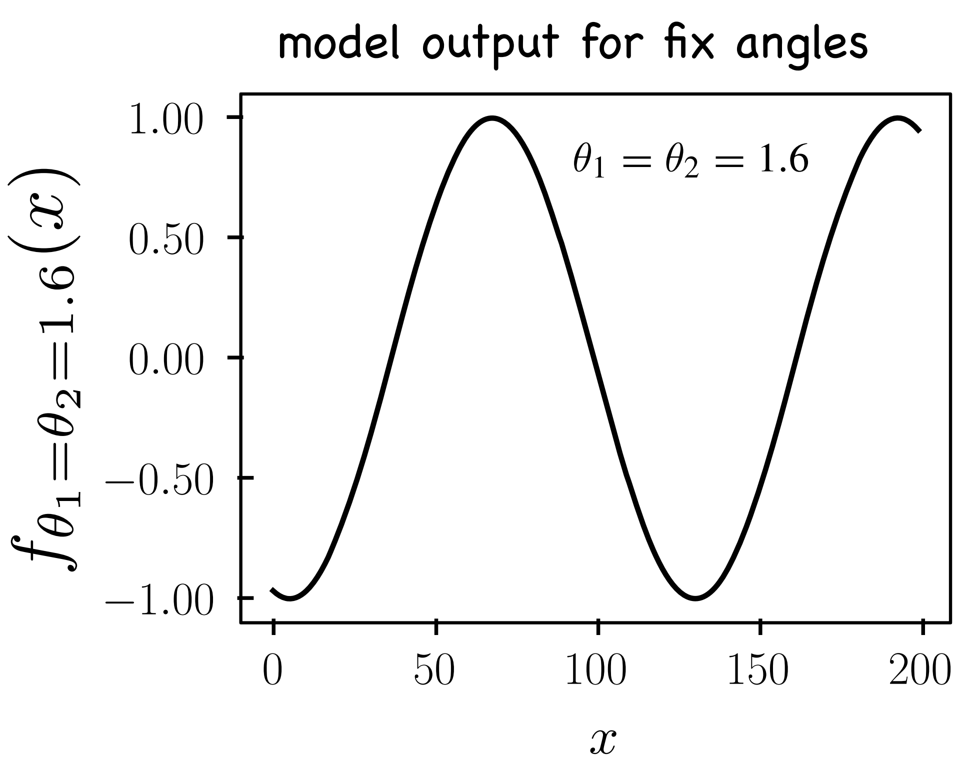
What happens on the Bloch-Sphere

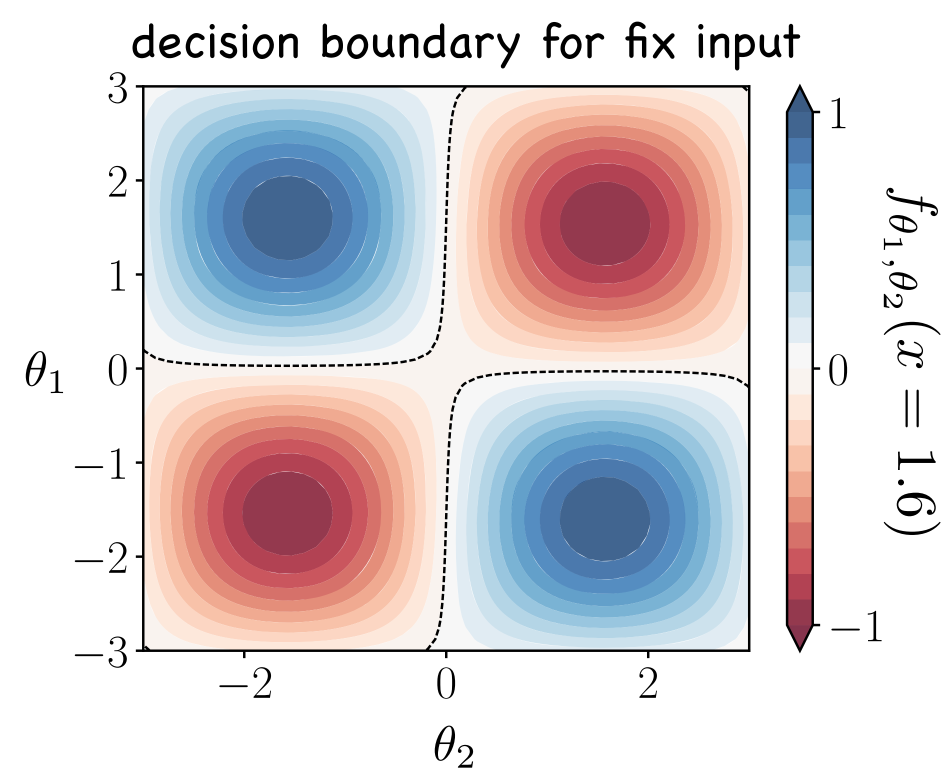


Classifier from simple example:

for binary classifier define e.g. $$y = \begin{cases} 1 \text{ if } f_\theta(x) > 0 \\ -1 \text{ else} \end{cases}$$

for probabilistic classifier (density estimator) $$p(1) = \frac{f_\theta(x)+1}{2}, \qquad p_0 = 1 - p_1$$





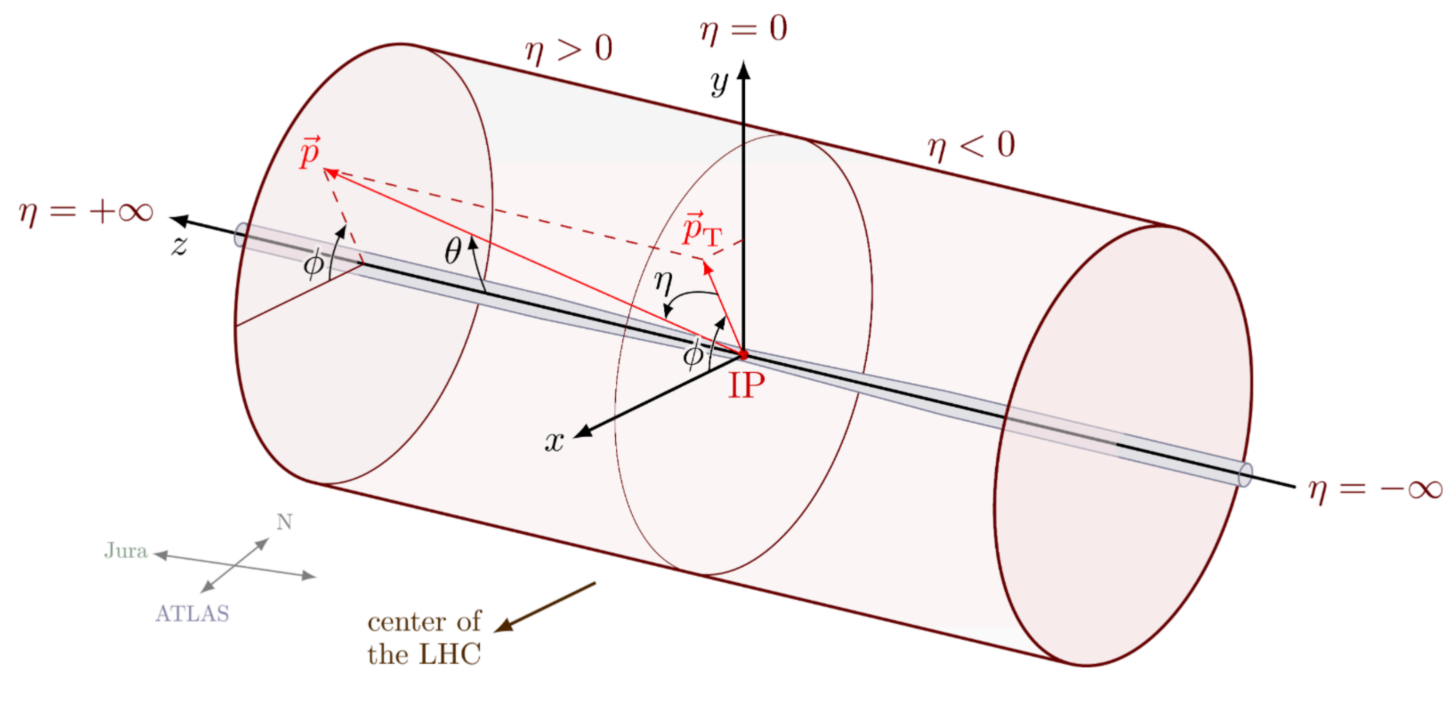
- Each particle defined by 3 features (ϕ, η, p_T)

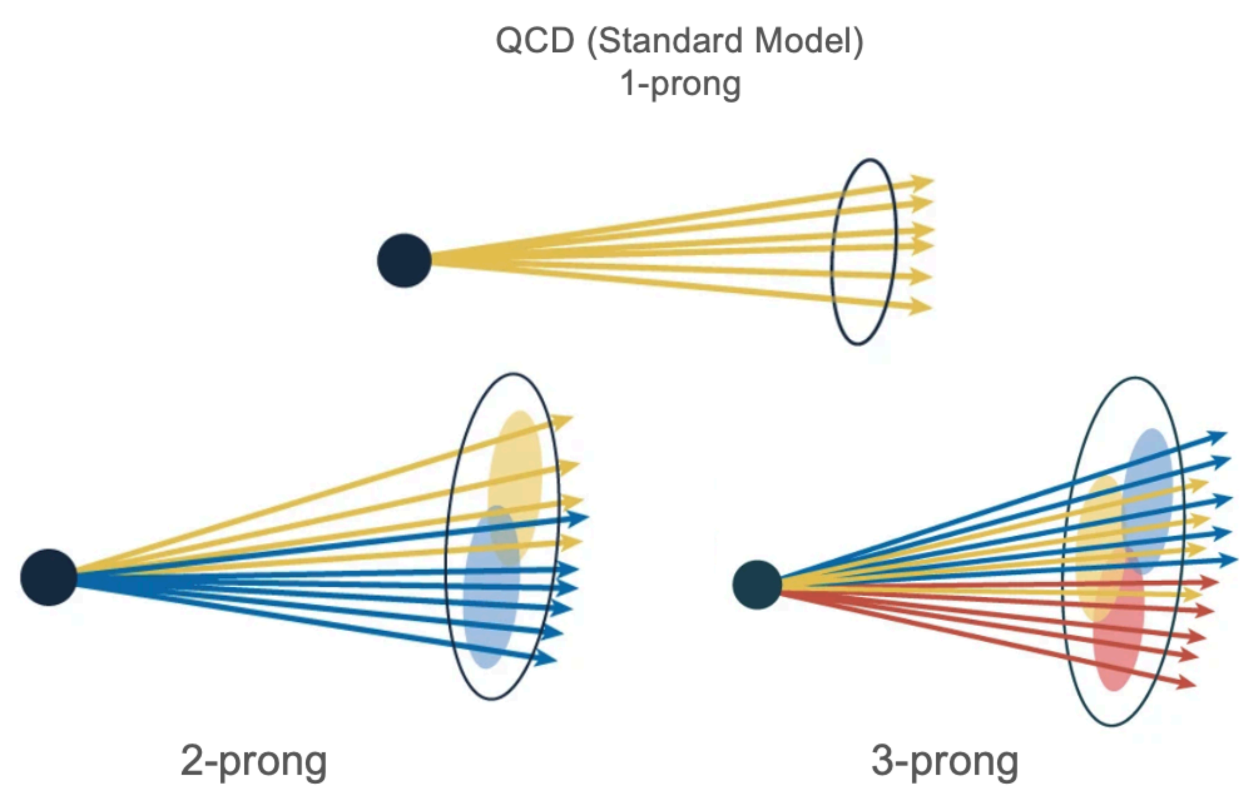
- LHC events consist of $\mathcal{O}(500)$ particles

- Fat jets have $\mathcal{O}(10)$ subjets

↓

Efficient data encoding crucial for realistic data analysis on quantum device





1P1Q Encoding

[Bal, Klute, Maier, Oughton, Pezone, MS '25]

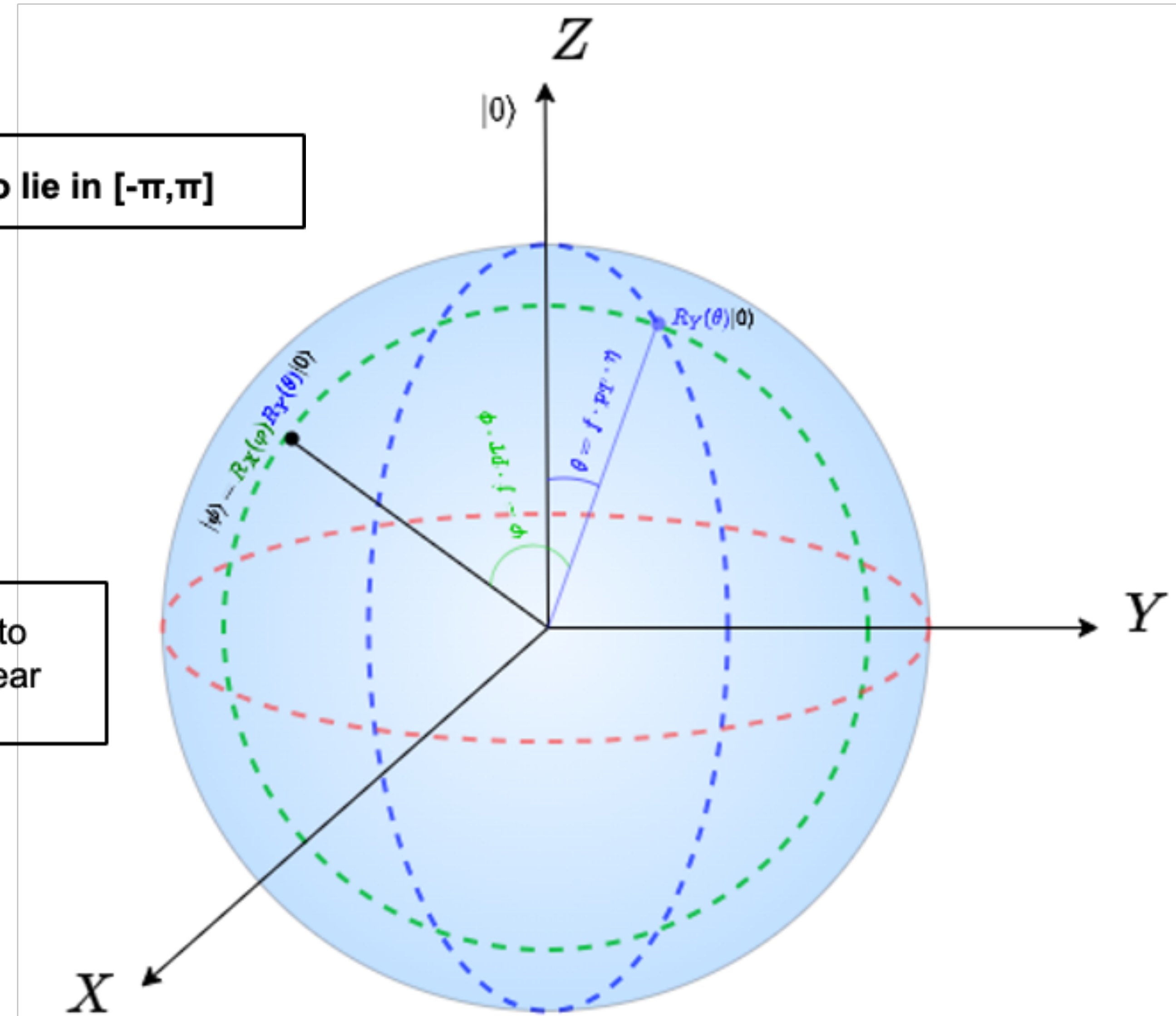
$$(p_T, \eta) \quad \rightarrow \theta = f \cdot \frac{p_T}{p_{T,jet}}(\eta - \eta_{jet})$$

$$(p_T, \phi) \quad \rightarrow \varphi = f \cdot \frac{p_T}{p_{T,jet}}(\phi - \phi_{jet})$$

Scaled to lie in [-π,π]

$$f = 1 + \frac{2\pi}{1 + e^{-w}}$$

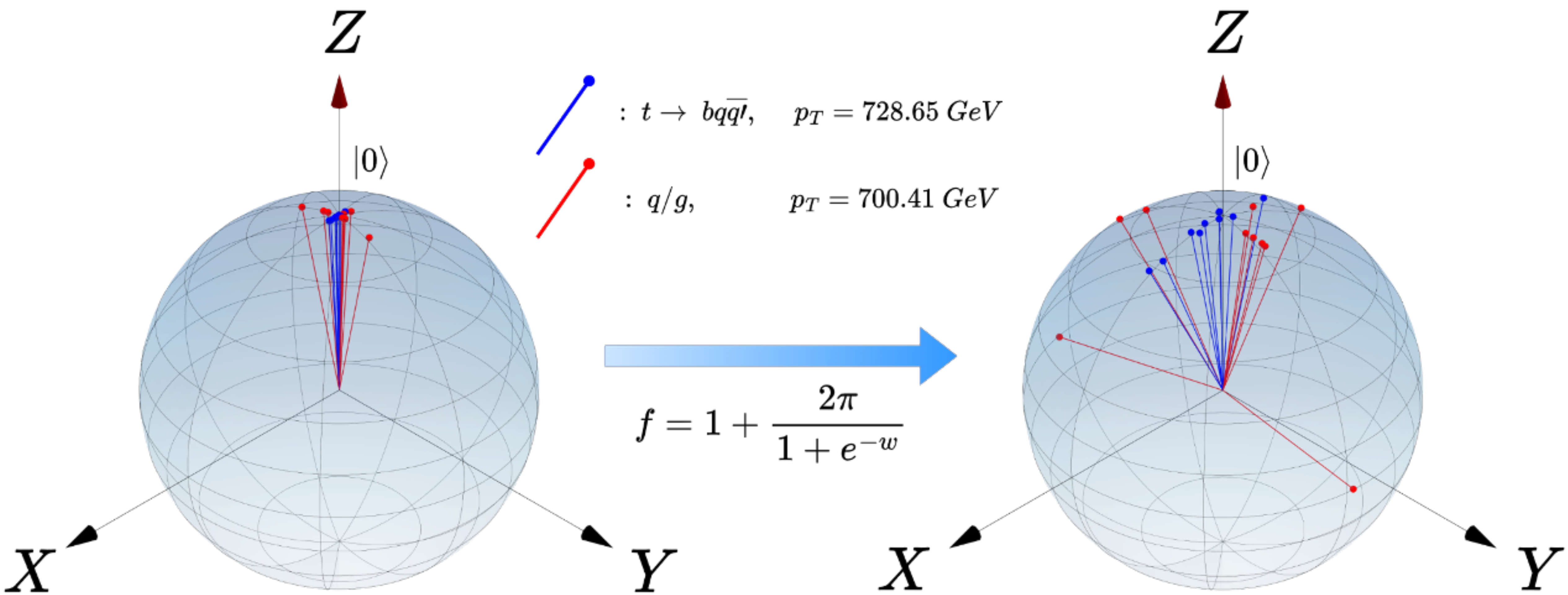
trainable parameter to prevent clustering near North Pole



$$(p_\mathrm{T}, \eta, \phi) \quad \rightarrow \quad \big|\psi\rangle = R_X(\varphi)R_Y(\theta)\big|0\rangle$$

$$(p_\mathrm{T}, \eta, \phi) \quad \rightarrow \quad |\psi\rangle \quad = \left(\cos\frac{\theta}{2}\cos\frac{\varphi}{2} + i(\sin\frac{\theta}{2}\sin\frac{\varphi}{2})\right)|0\rangle$$
$$+ \left(\sin\frac{\theta}{2}\cos\frac{\varphi}{2} - i(\cos\frac{\theta}{2}\sin\frac{\varphi}{2})\right)|1\rangle$$

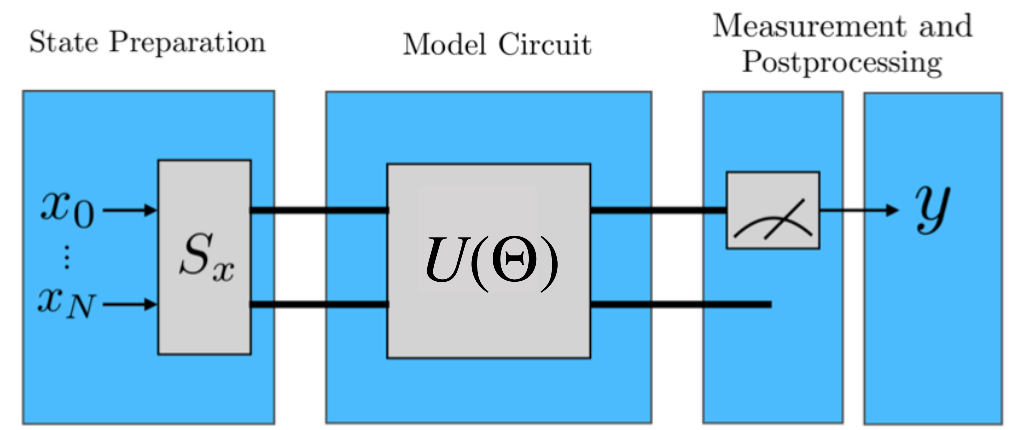
1P1Q Encoding



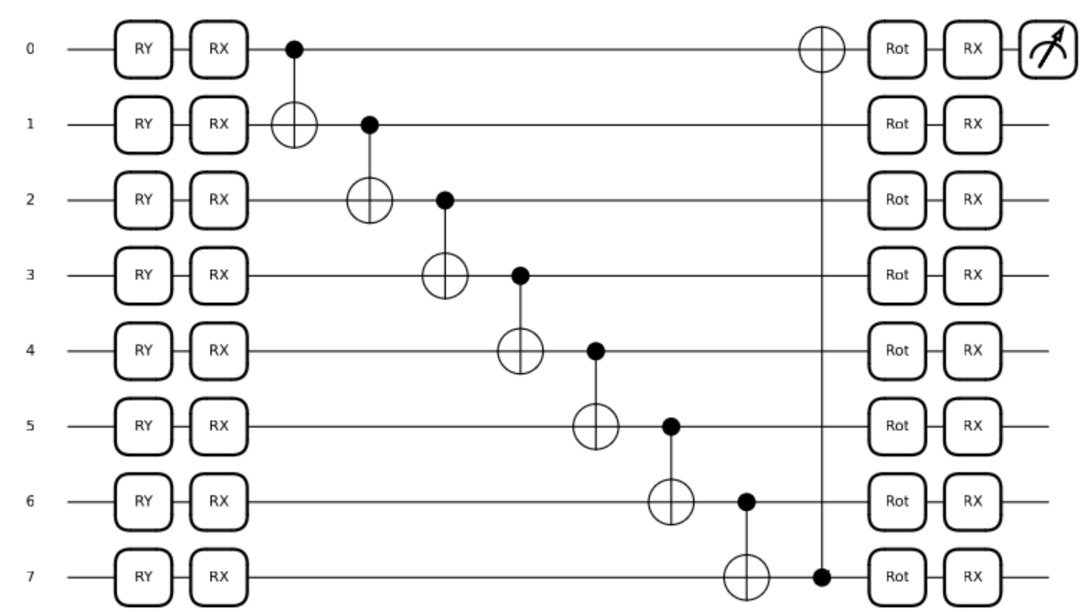
- Representation of specific event as 1P1Q encoded on qubits
- Each finalstate particle in event correponds to one vector on bloch sphere

Supervised-Learning with Variational Quantum Circuit

- VQC supervised learning algorithm
- Use the JetClass dataset first introduced by authors of Particle Transformer (ParT)
- Train on labelled data, signal = boosted top quarks bkg = QCD fat jets
- Avoid jet bias -> flat p_T in [500,1000] GeV
- Train only on 3 basic kinematic featrues (p_T, η, ϕ) with appropraite sclaing and normalization



$$U(\Theta) = \left(\bigotimes_{i=1}^{N-1} R_X(\phi_i) R_Y(\theta_i) R_Z(\omega_i) \right) \otimes \left(\bigotimes_{i=0}^{N} C_{i,(i+1) \bmod N} \right)$$



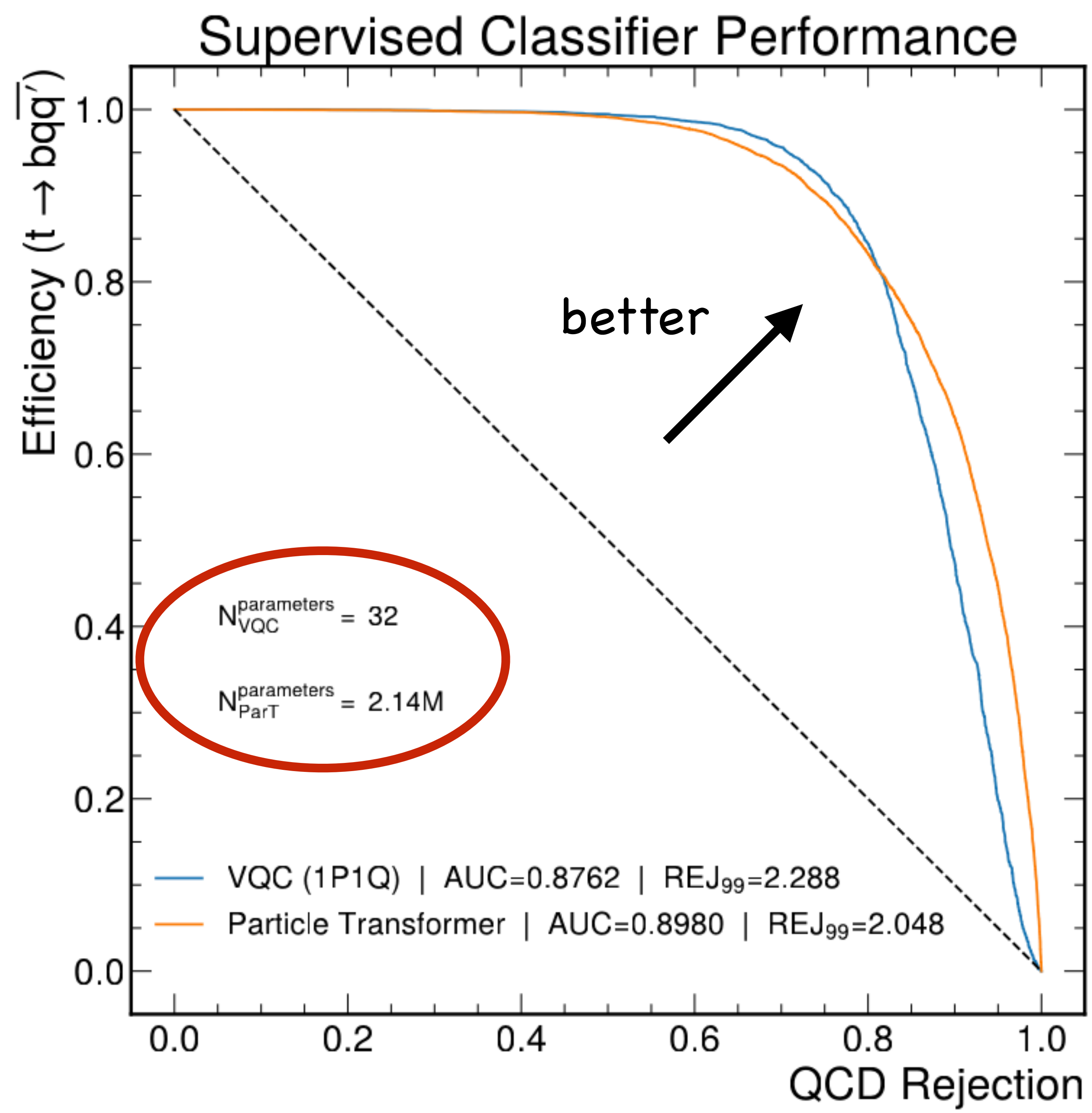
→ Assign one particle to one qubit

→ Jet represented by N qubits (N hardest constits)

Results for VQC with 1P1Q Encoding

- Input are 10 hardest subjects of fat jet

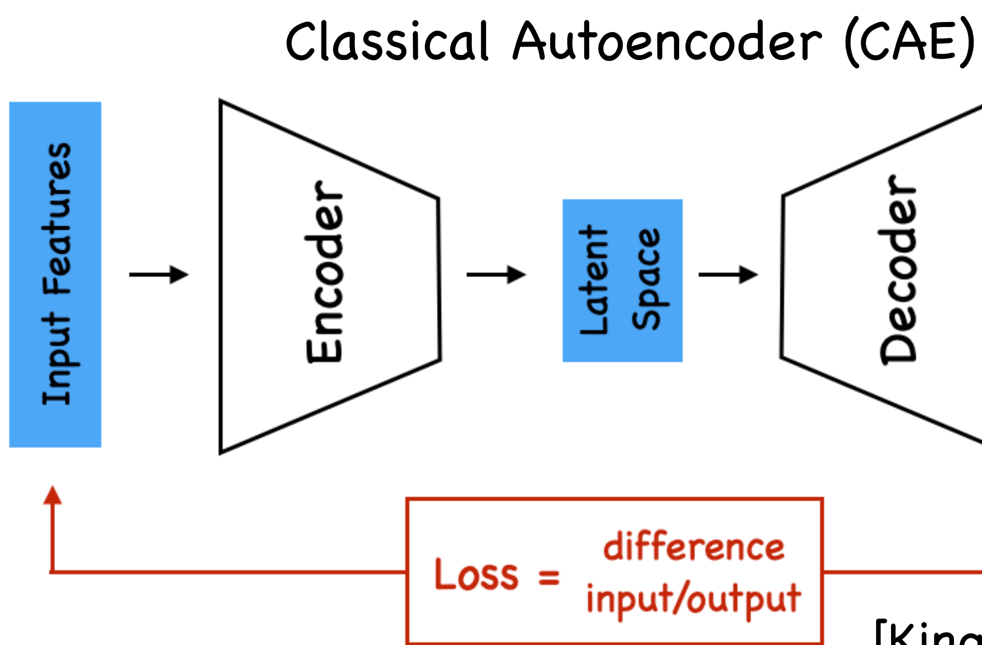
- Tiny VQC performs comparably to state-of-the-art Particle Transformer



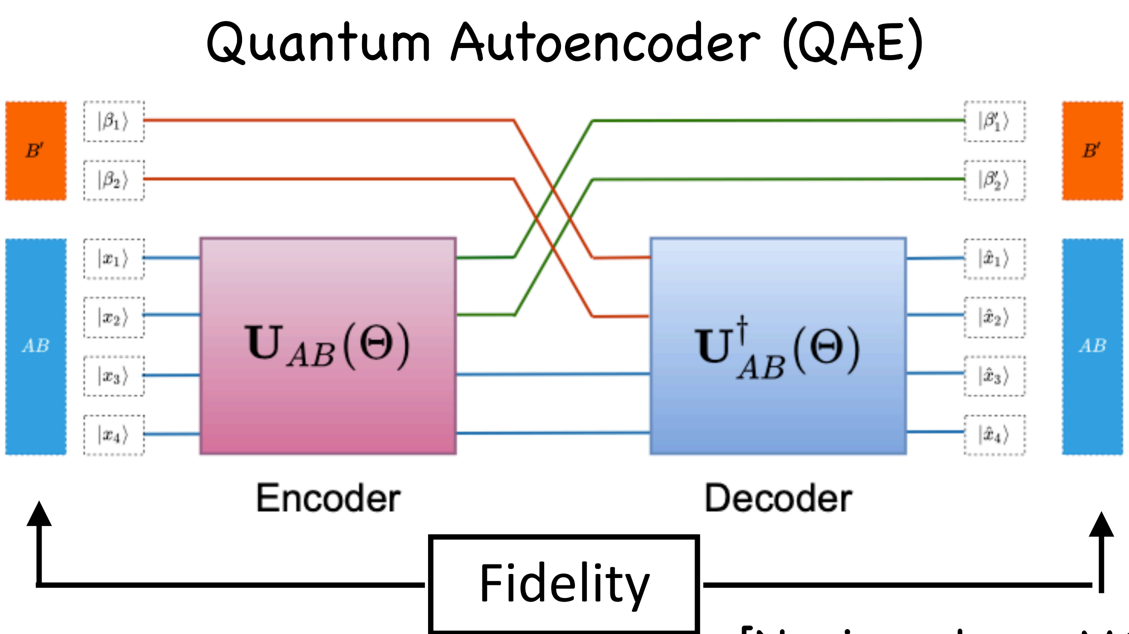
- Drastically reduced model parameters: VQC (**32**) vs Transformer (**2.14 Millions**)

[Bal, Klute, Maier, Oughton, Pezone, MS '25]

Unsupervised learning with Quantum Autoencoders



[Kingma, Welling '13]



[Ngairangbam, MS, Takeuchi '21]

- Freature input is encoded into information bottleneck, i.e. latent space with smaller dimension that feature space

- Latent space decoded into reconstructed output, which is then compared with input via loss-function (often MSE)
  -> Encoder+Decoder trained together to produce output similar to input

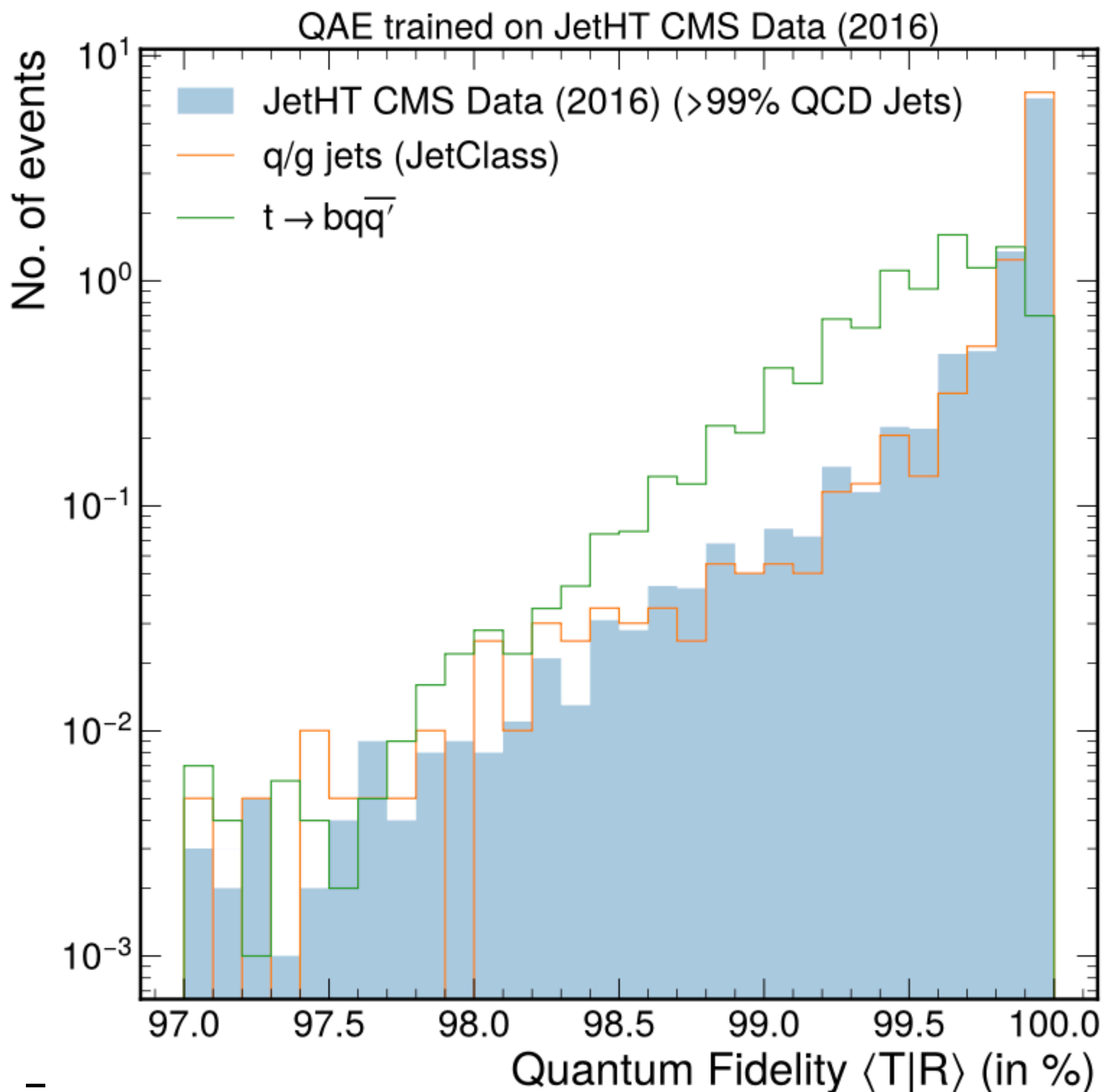
- Quantum AE needs to work with unitary gate operations.
  Thus, need trash states to realise information bottleneck

- Fidelity for QAE obtained from CMS data

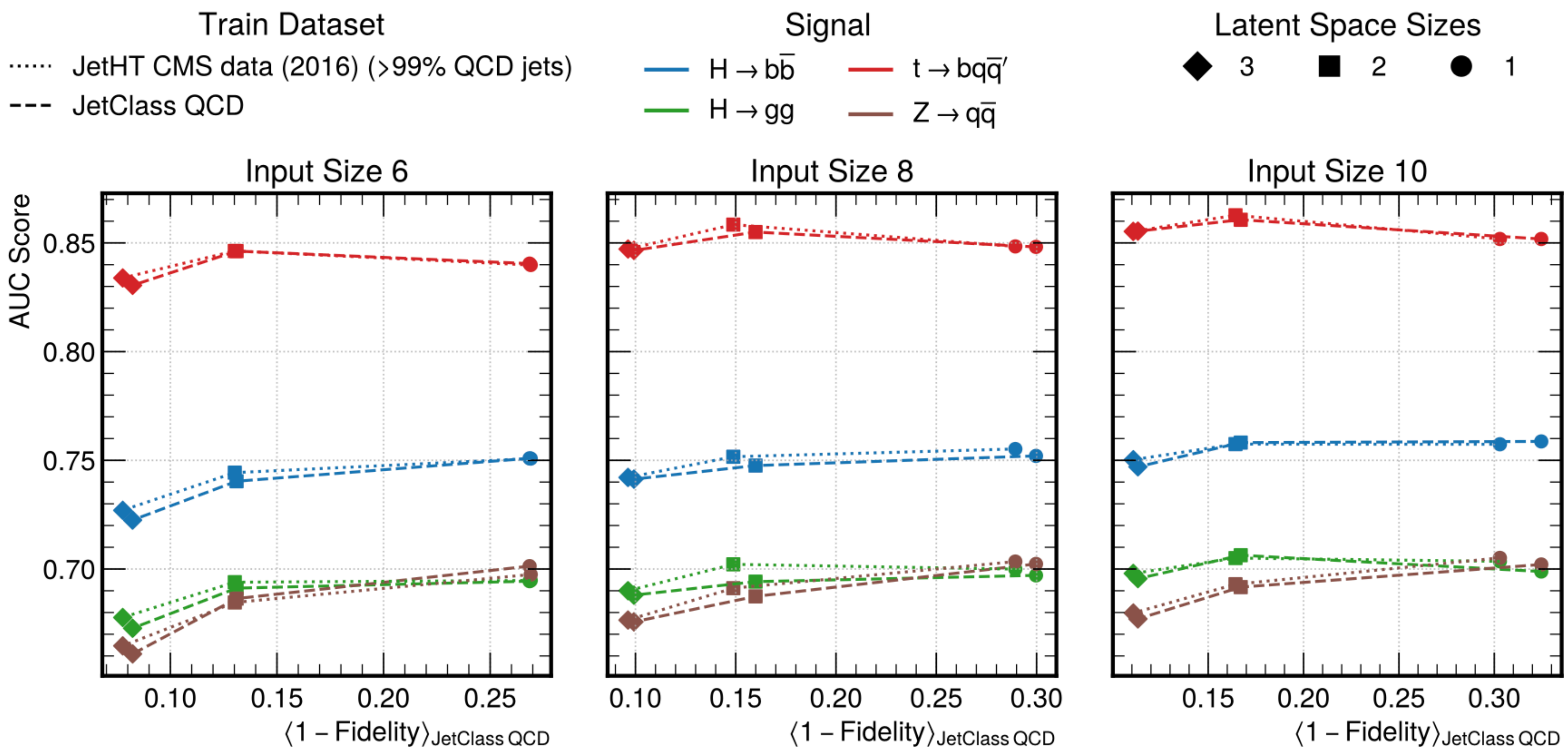
- Very good data - simulation agreement

- Signals and bkgs flat in pT in [500, 1000] GeV

- Signals:
  $H \to c\bar{c}$, $H \to gg$
  $W \to q\bar{q}$, $Z \to q\bar{q}$, $t \to bq\bar{q}$



Performance QAE for different signals



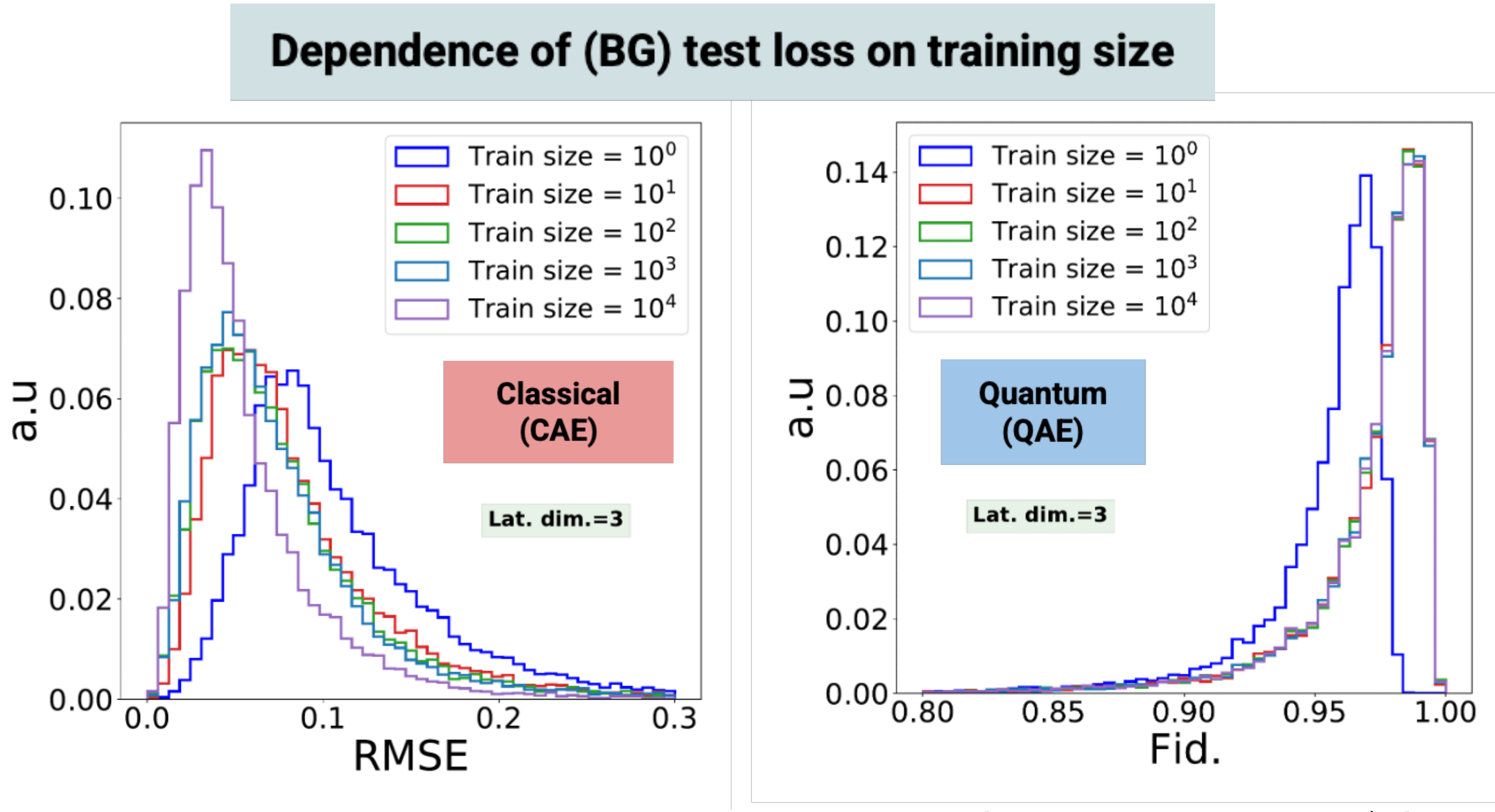
Comparison with CAE

CAE: 30-20-16-12-6

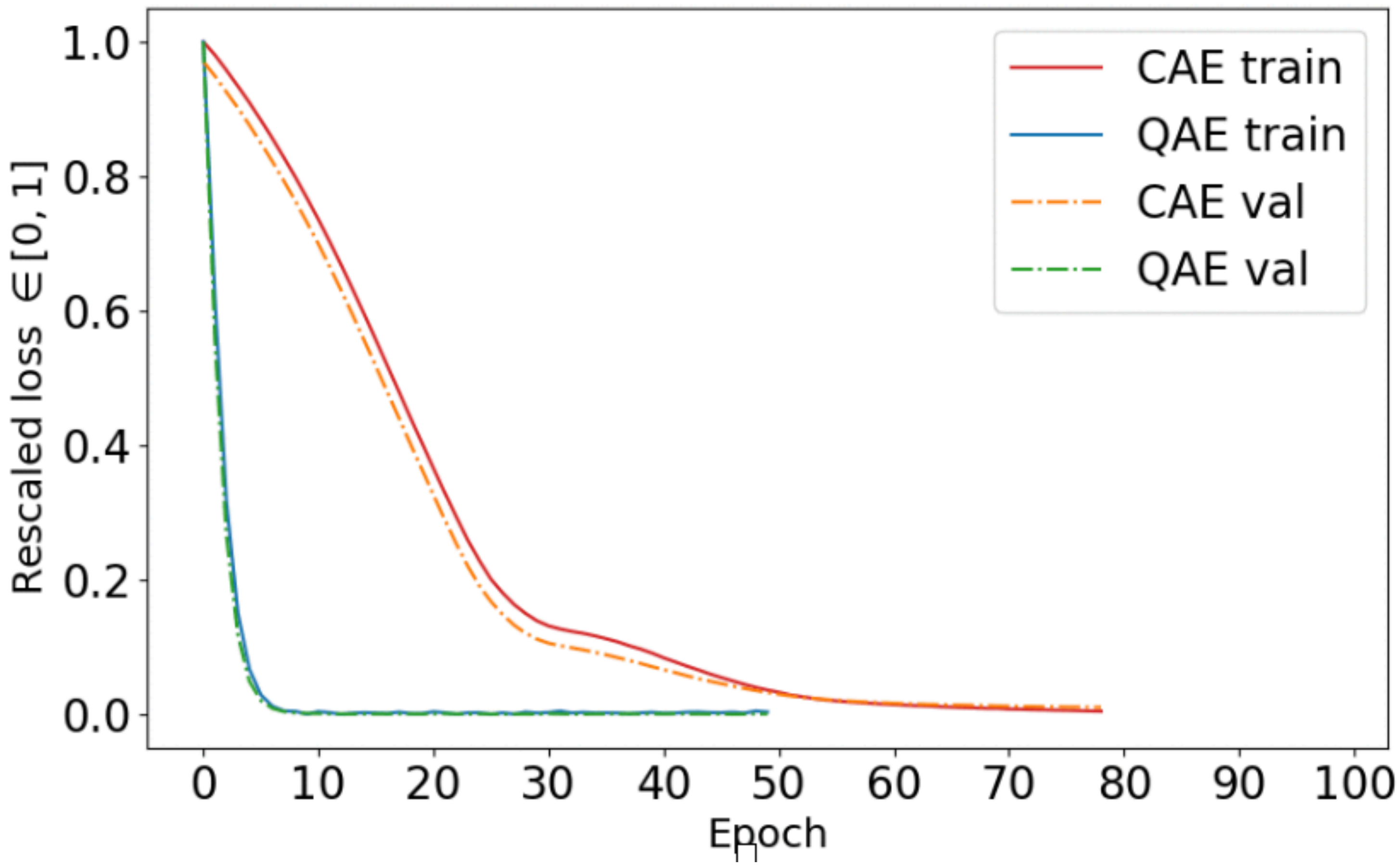
latent space

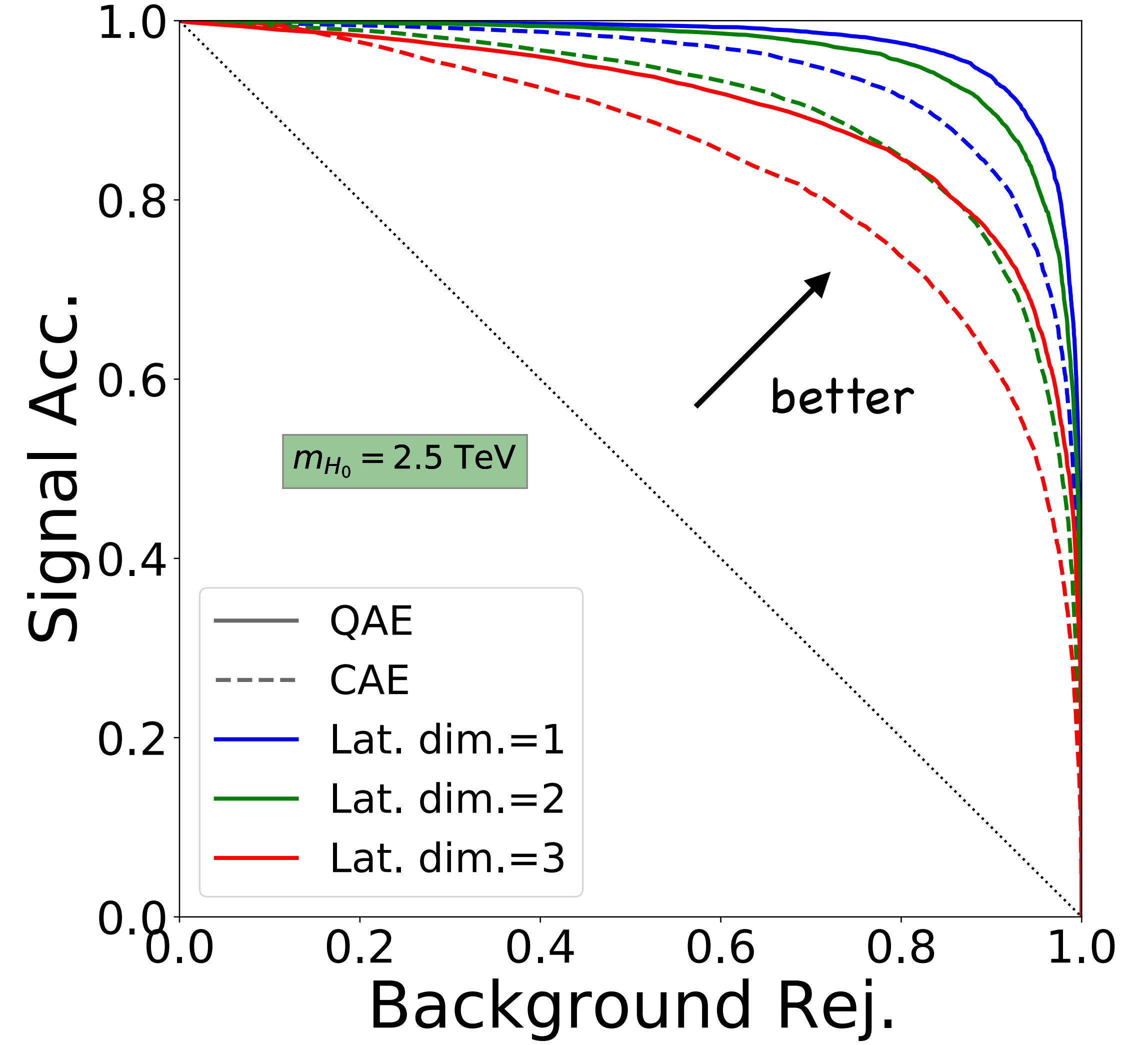
Signals / Algorithm	$W \to q\bar{q}$	$H \to b\bar{b}$	$t \to bq\overline{q'}$
QAE	0.693	0.757	0.861
CAE	0.671	0.739	0.858

Results: Training size dependence



[Ngairangbam, MS, Takeuchi '21]

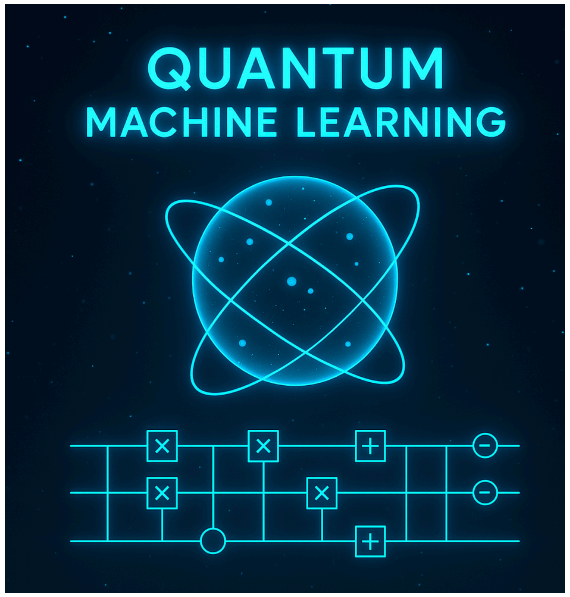




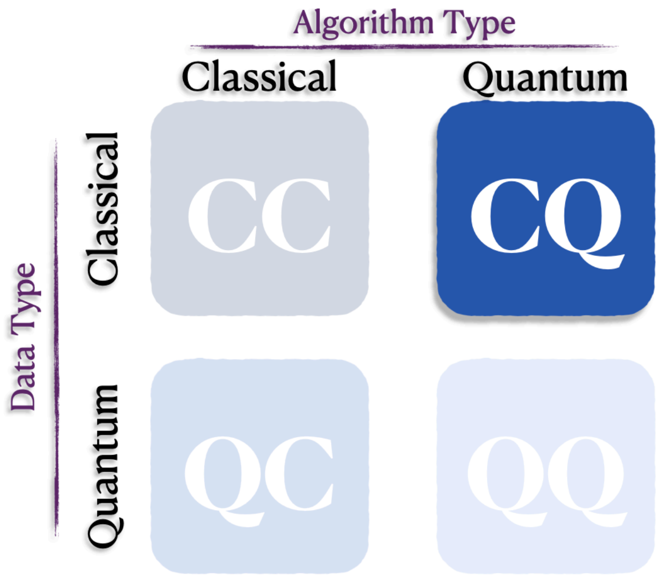
[Ngairangbam, MS, Takeuchi '21]

→ Much faster training and better performance for Quantum autoencoder

→ In our test cases QAE > CAE for much larger classical networks



Summary



- Encoding integral part of Quantum Neural Networks
  -> expressivity of the network
  -> resource saving and fast

- 1P1Q data encoding ideally suited for HEP events and final states on qubit devices

- Astonishing results for QNN, i.e. VQC and QAE, in comparison to classical neural networks

→ requires better understanding