PyHEP 2021 (virtual) Workshop

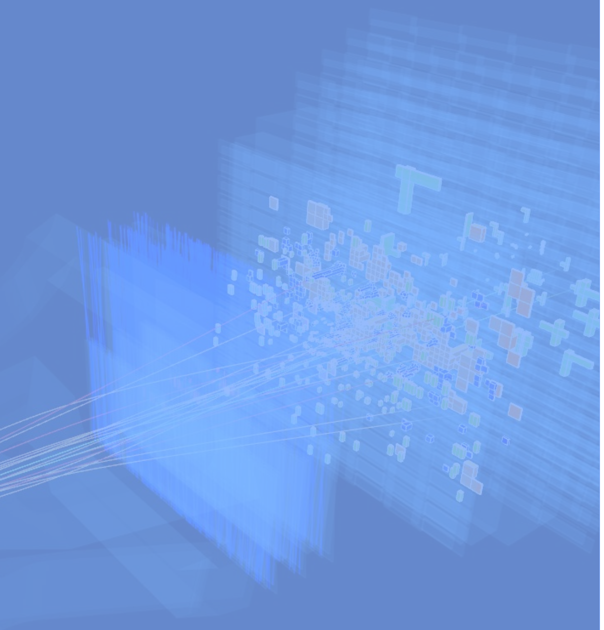
GitHub Repository

# Quantum Machine Learning for jet tagging @ LHCb

Speakers: Davide Nicotra & Davide Zuliani
Co-Authors: Alessio Gianelle, Donatella Lucchesi, Lorenzo Sestini

Università degli Studi di Padova

Istituto Nazionale di Fisica Nucleare



# Outline

**Physics introduction**

Physics cases @ LHCb

**A quantum approach**

Quantum Machine Learning for jet tagging

**A look at the code!**

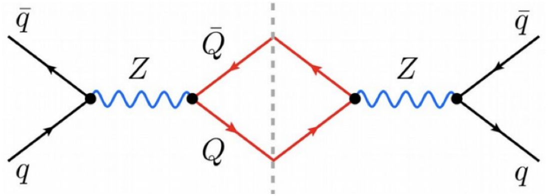
# Physics introduction

# Physics cases

Jet flavor identification is mandatory for several physics cases

- **b/b-bar charge asymmetry**, interesting for **New Physics** searches (our physics case)

$$A_C^{b\bar{b}} = \frac{N(\Delta y > 0) - N(\Delta y < 0)}{N(\Delta y > 0) + N(\Delta y < 0)} \qquad \Delta y = y_b - y_{\bar{b}}$$

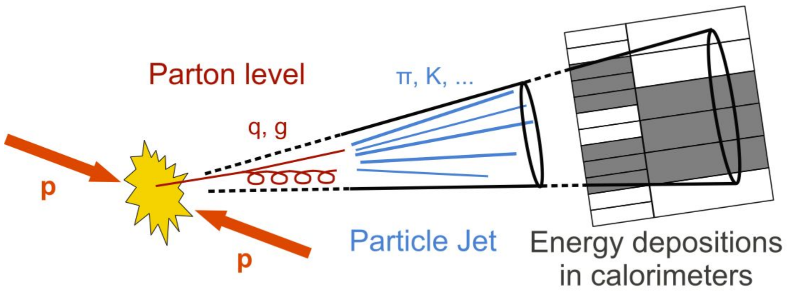


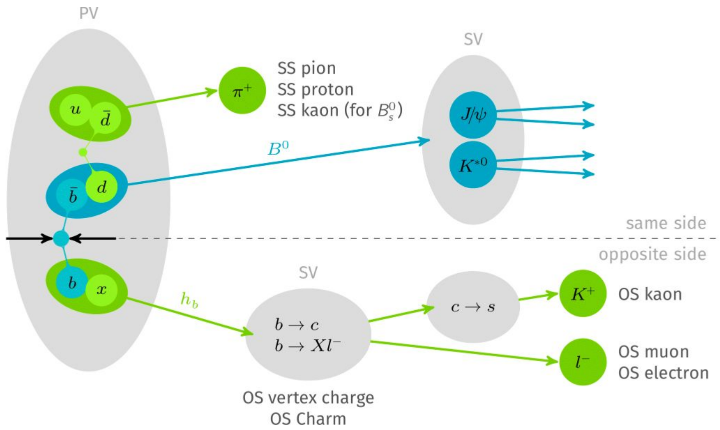

- **identification** of **Higgs boson** decaying to:
  - b b-bar jets (recently observed @ ATLAS & CMS)
  - c c-bar jets (not yet observed)

Final states detected by the experiment → **jets**

# b-jet tagging @ LHCb

@ LHC is **fundamental** to identify the **flavour** of the quark originating the jet → **jet tagging**



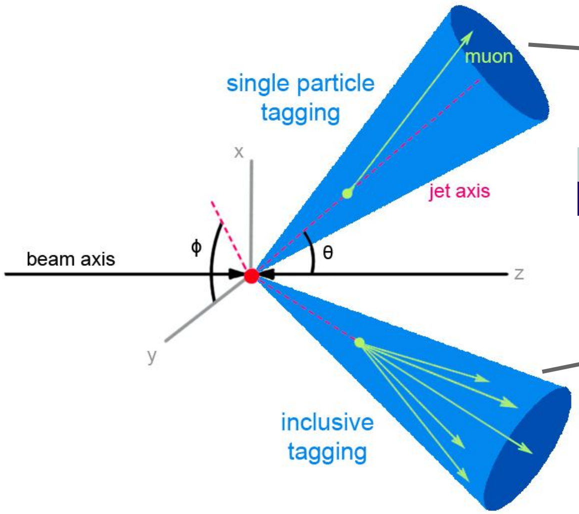

*"Jets are streams of particles produced by QCD processes in proton-proton collisions"*

In particular @ LHCb we are interested in studying jets generated by b and b-bar quarks

**b-jet tagging**

# Classical tagging methods

There are two possible approaches to achieve this task: **exclusive** and **inclusive** algorithms




**exclusive algorithms use information from a specific process to infer the quark flavour**

**muon tagging**: a $\mu$ coming from the b (or b-bar) semileptonic decay (P =10%) is used to tag the jet ($\mu$ and b charges are correlated)

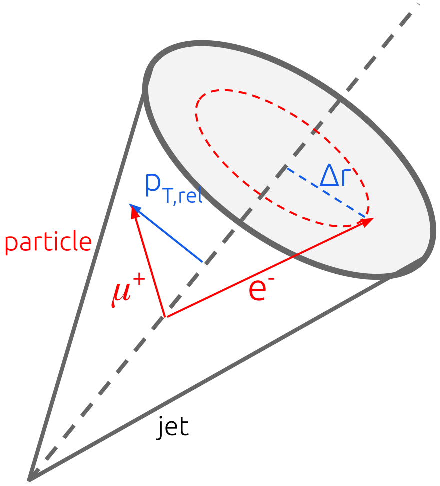
**inclusive algorithms use information from the whole jet substructure**

Machine Learning (ML) algorithms such as **Deep Neural Networks** (DNN)

LHCb Open Data

# Dataset

**Official LHCb simulation** of di-jet generated by b and b-bar quarks @ $E_{cm}$ = 13 TeV (Run 2 condition) → ~ 700.000 jets (60% training, 40% testing & evaluation)




Inside each jet we consider 5 types of particles

**muon electron pion kaon proton**

and for each type we select 3 variables:

- $p_{T,rel}$: transverse momentum relative to the jet axis
- $\Delta r$: distance relative to jet axis
- q: charge of the particle

\+ 1 global variable → total jet charge

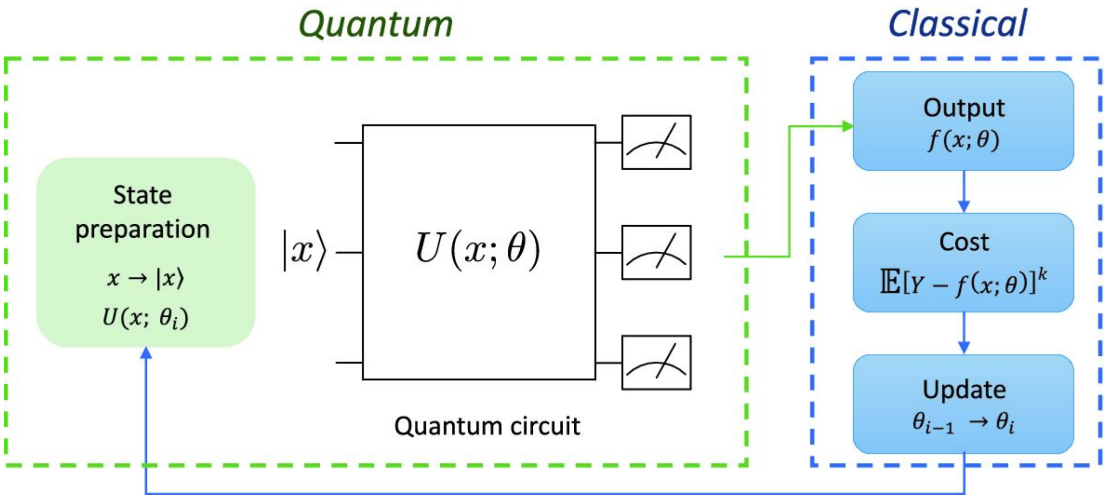
**for a total of 16 input variables**

**the official LHCb simulation resembles real LHCb data**

# A quantum approach

# Going to quantum...

classification problem = **Variational Quantum Classifier**



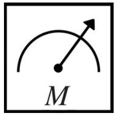

Data are fed into variational quantum circuit.

Measurements of qubits are mapped to probabilities for labels.

Probabilities are used to estimate a cost function which is optimized through a classical optimizer

# Loss function and Optimizer

### Measurement




$$\mathbb{E}(\sigma_z) = M \quad M \in [-1, 1]$$

**Tagging probabilities**

$$P_b = \frac{M+1}{2}$$

$$P_{\bar{b}} = 1 - P_b$$

### Cost function

$$MSE(\theta_i) = \frac{1}{N} \sum_{dataset} (M - M_{true})^2$$

$$CE(\theta_i) = -\frac{1}{N} \sum_{dataset} \sum_{i \in \{b, \bar{b}\}} p_i \log q_i$$

### Classical optimization

Gradient-Free optimizers.
e.g.

- COBYLA
- SPSA

Gradient Descent optimizers:
e.g.

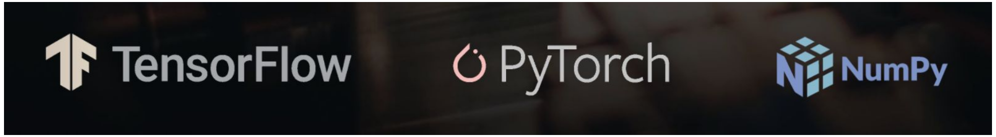
- ADAM
- RMSProp

# Libraries

PENNYLANE

- PROS
  - Interface to multiple quantum simulators and hardware


  - Better for QML and Hybrid QML
  - Native interfaces to ML libraries Autograd, PyTorch and Tensorflow

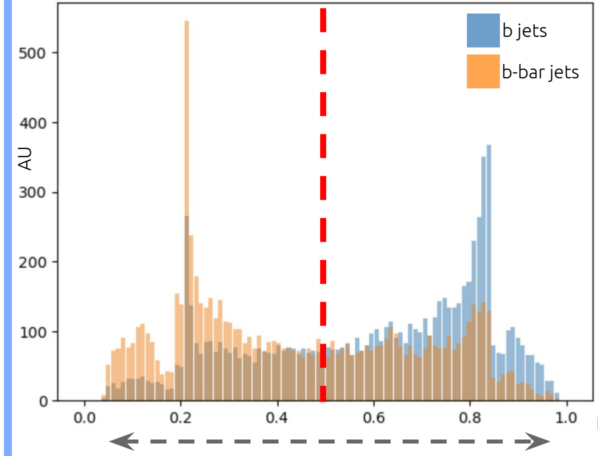


# Training results

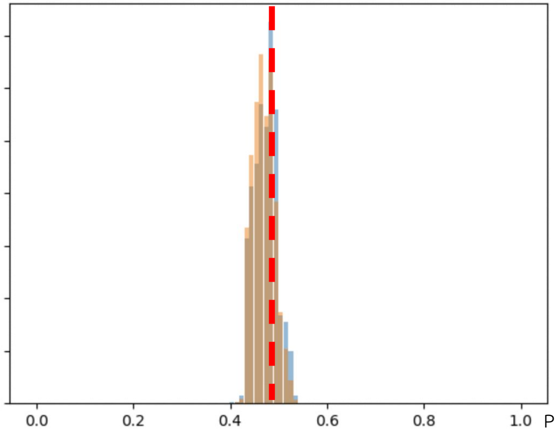
We show some results on training performances for the VQC

## Probability distribution

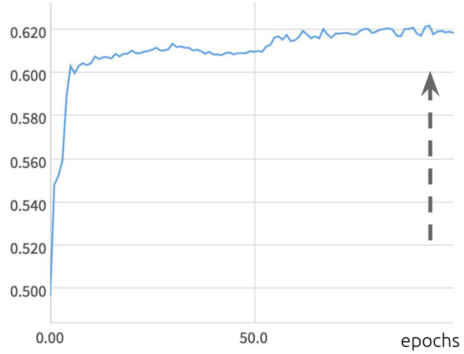
### Result for last epoch



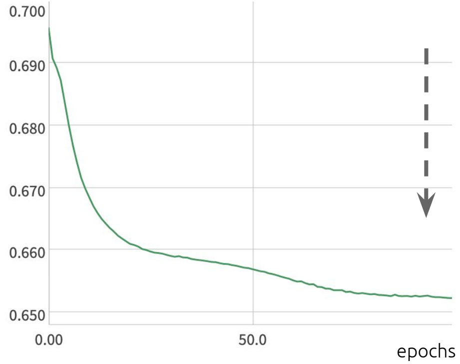

### Variations through epochs




## Accuracy




## Loss

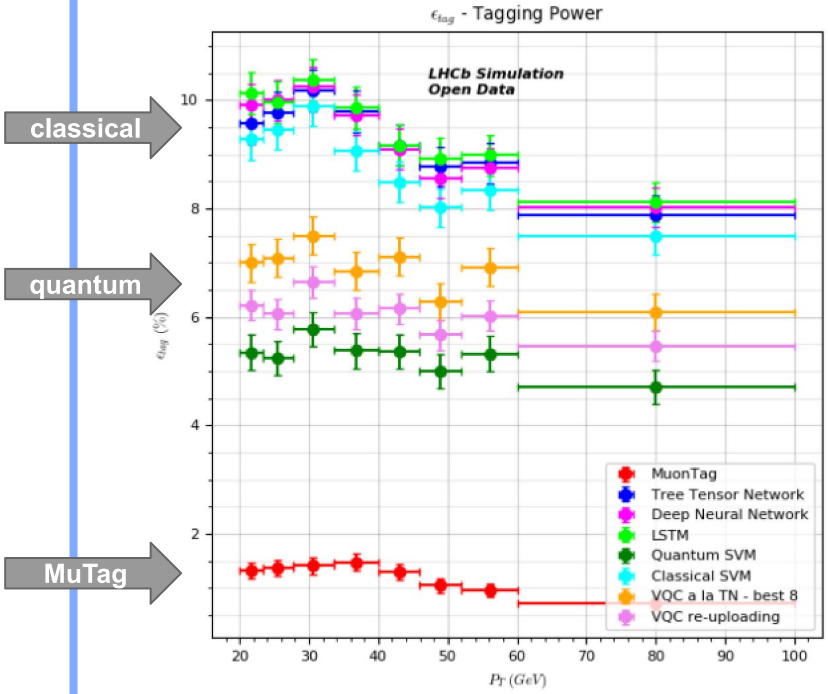


# A look at the code!

[GitHub Repository]

# Some (more) interesting results…

# **Results on** noiseless **simulator**




We show **results** for our quantum algorithm
The tagging power $\varepsilon_{tag}$ is the figure of merit for our physics case

- results for classical algorithms are obtained using all the variables and the whole training dataset
- several quantum circuit geometries have been considered (more details in the backup)
- QML performs slightly worse than classical algorithms, but room for improvements!

# Thank you for your attention

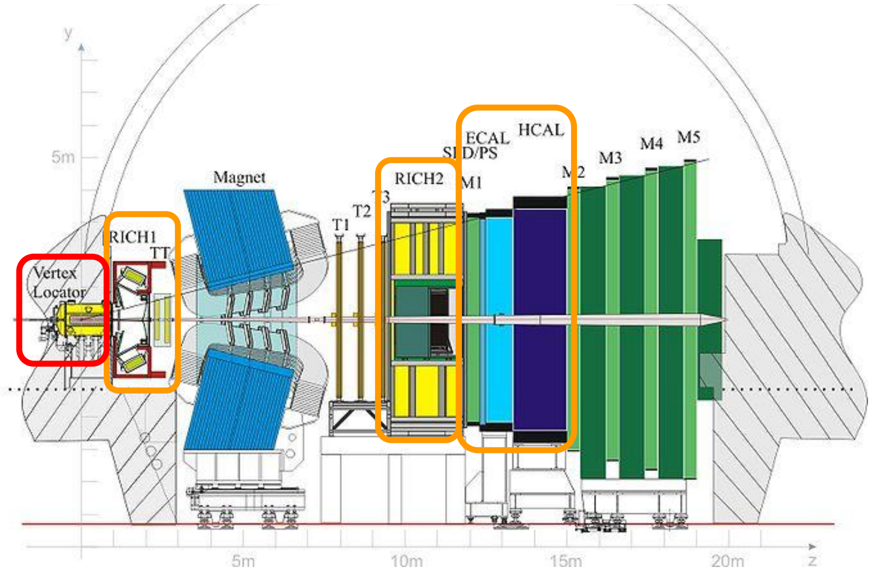
UNIVE
DEGLI
DI PAD

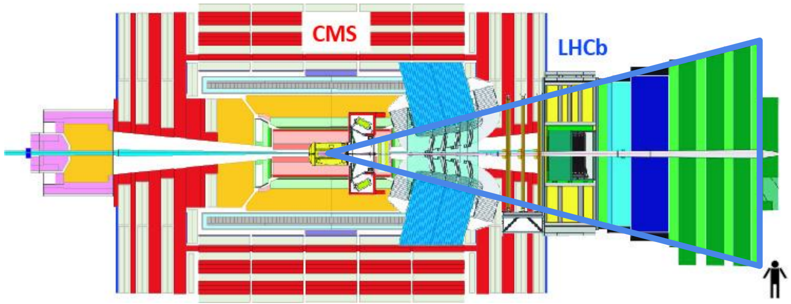
Istituto Nazionale di Fisica Nucleare

# Backup slides

# LHCb

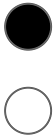
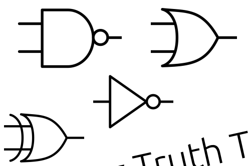
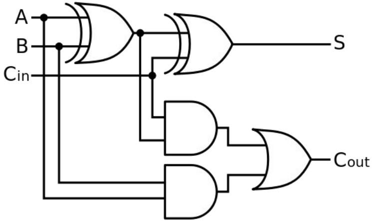
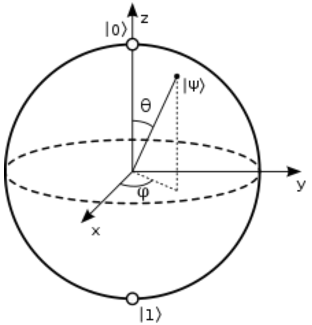
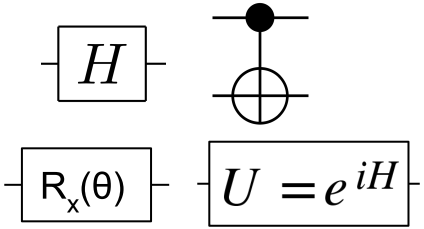
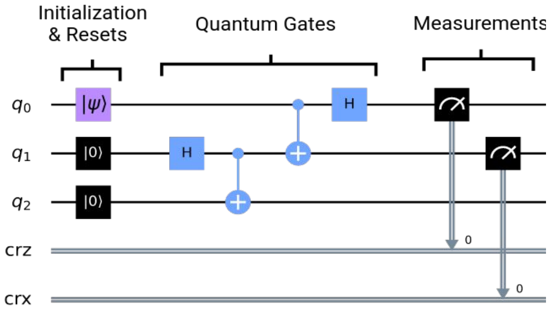
LHCb is a **forward spectrometer** designed to study **flavour physics**




- complementary phase-space region w.r.t. ATLAS & CMS
- excellent vertex reconstruction
- excellent Particle Identification (PID)

# What is a quantum circuit?

| | Element of information | Operations | Circuit |
|---|---|---|---|
| **Classical computer** | 0 1  |  **Any** Truth Table |   |
| **Quantum computer** |   |   **Any** Unitary Matrix |   |

# Quantum gates

### Rotation gates

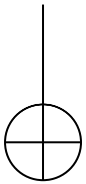
$$\boxed{R_X(\theta)} = e^{i\theta\frac{\sigma_x}{2}}$$

$$\boxed{R_Y(\theta)} = e^{i\theta\frac{\sigma_y}{2}}$$

$$\boxed{R(\theta_1,\theta_2,\theta_3)} = e^{i\frac{\theta_1\sigma_x+\theta_2\sigma_y+\theta_3\sigma_z}{2}}$$

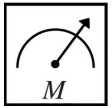
### Entanglement gates

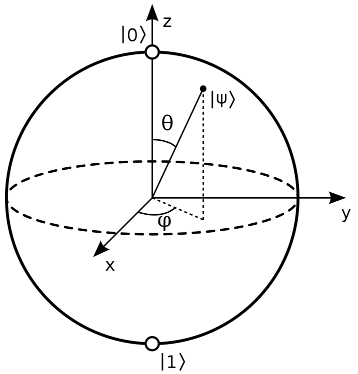
$$\boxed{ZZ\ \theta} = e^{i\theta\frac{\sigma_z\otimes\sigma_z}{2}}$$



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

### Measurement

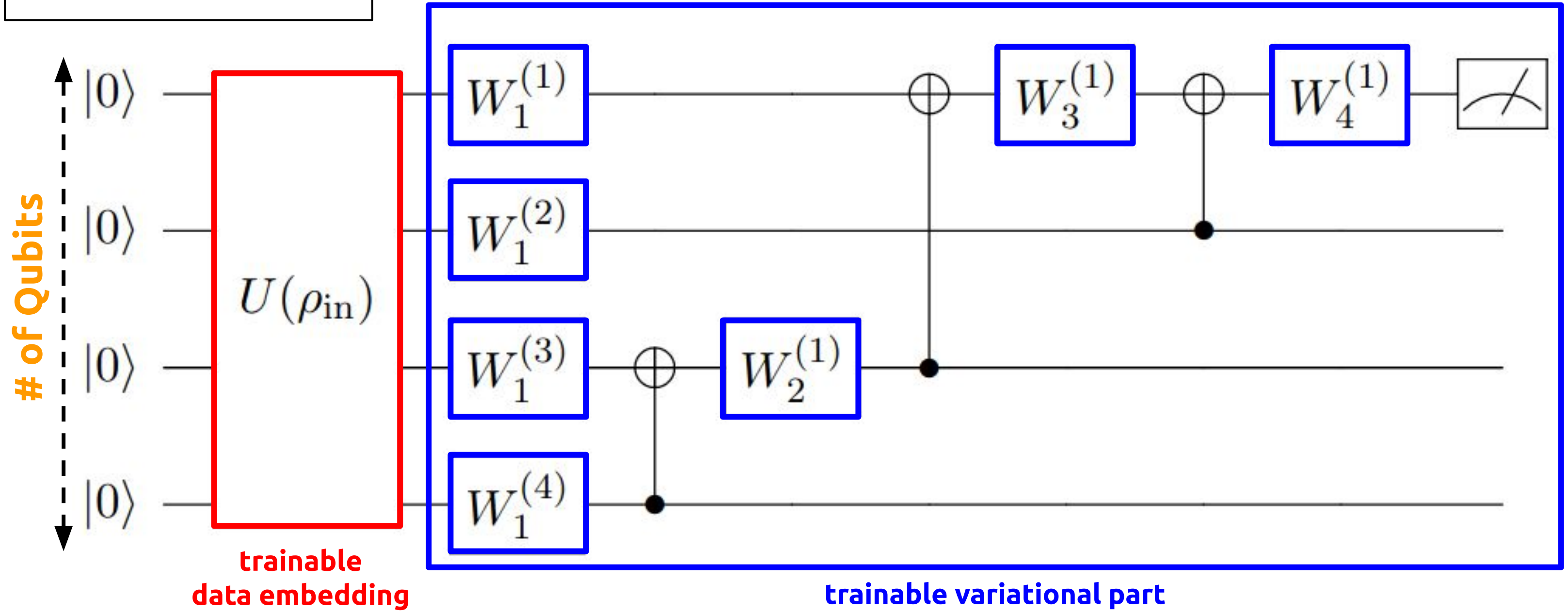


[2011.06258] Toward Trainability of Quantum Neural Networks https://arxiv.org/abs/2011.06258

# VQC "a la TN" classifier

Input
Parameters
Hyper-Parameters



[1907.02085] Data re-uploading for a universal quantum classifier https://arxiv.org/abs/1907.02085

# Data Re-uploading classifier

Input
Parameters
Hyper-Parameters