b-jet identification with a Variational Quantum Classifier

Davide Nicotra
Universiteit Maastricht

The Paper!

- Most of the work done during my Master thesis in Padua
- First quantum application to hadronic jet identification

JHEP

Published for SISSA by Springer

Received: March 1, 2022
Accepted: June 20, 2022
Published: August 1, 2022

Quantum Machine Learning for b-jet charge identification

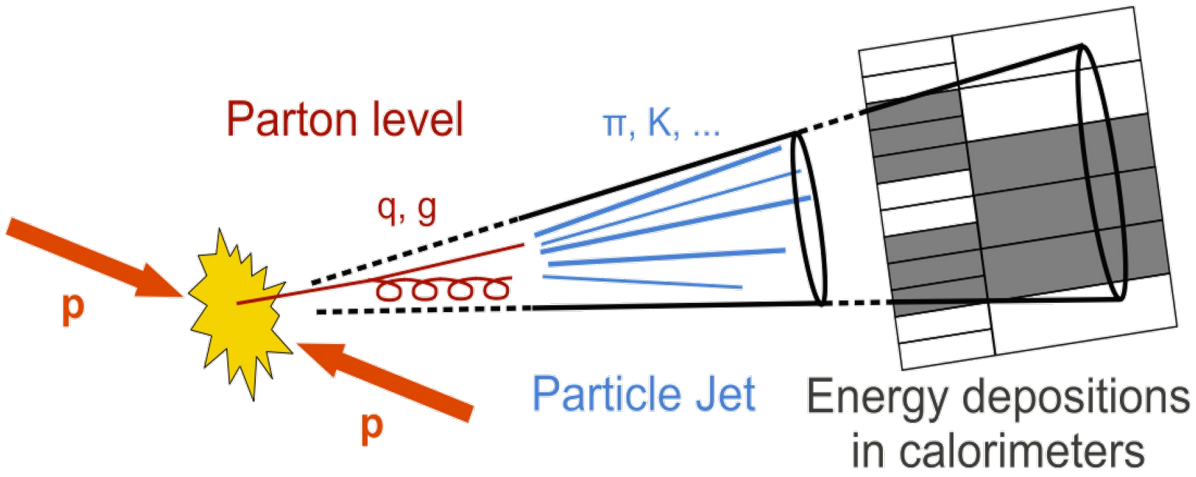
Alessio Gianelle,[a] Patrick Koppenburg,[b] Donatella Lucchesi,[a,c] Davide Nicotra,[c,d] Eduardo Rodrigues,[e] Lorenzo Sestini,[a] Jacco de Vries[d] and Davide Zuliani[a,c,f]

[a] *INFN Sezione di Padova, Padova, Italy*
[b] *Nikhef National Institute for Subatomic Physics, Amsterdam, Netherlands*
[c] *Università degli Studi di Padova, Padova, Italy*
[d] *Universiteit Maastricht, Maastricht, Netherlands*
[e] *Oliver Lodge Laboratory, University of Liverpool, Liverpool, U.K.*
[f] *European Organization for Nuclear Research (CERN), Geneva, Switzerland*

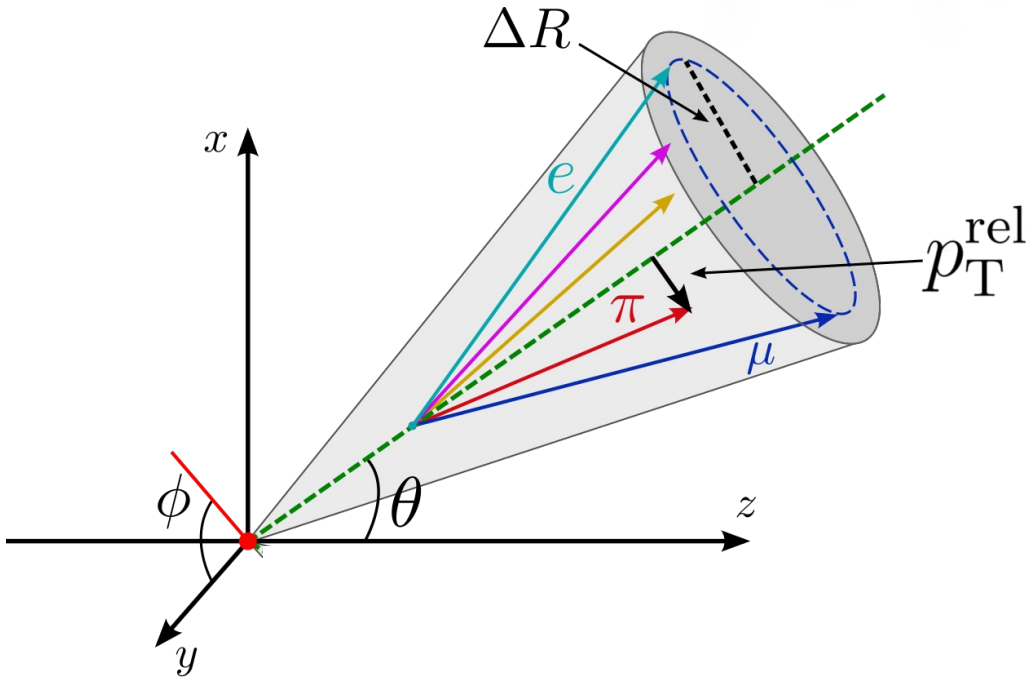
https://link.springer.com/article/10.1007/JHEP08(2022)014

The Problem: b-jet identification

- At LHCb we study *b* quarks produced in proton-proton high energy collision.
- We need to identify jets coming from *b* quarks!

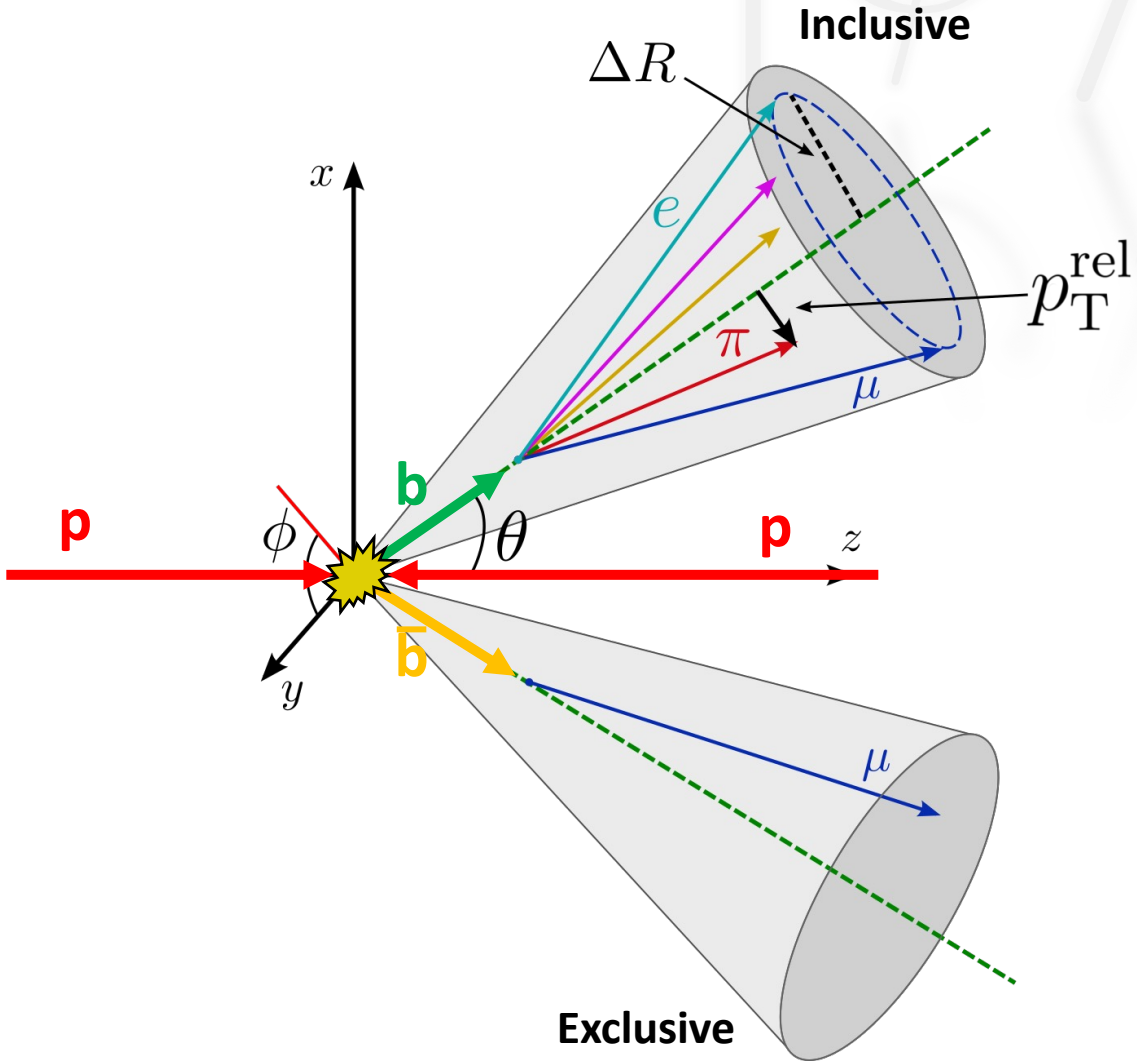


- The hadronization process dilutes the information carried by the *b* quark. We need to reconstruct it!



The Problem: b-jet charge tagging

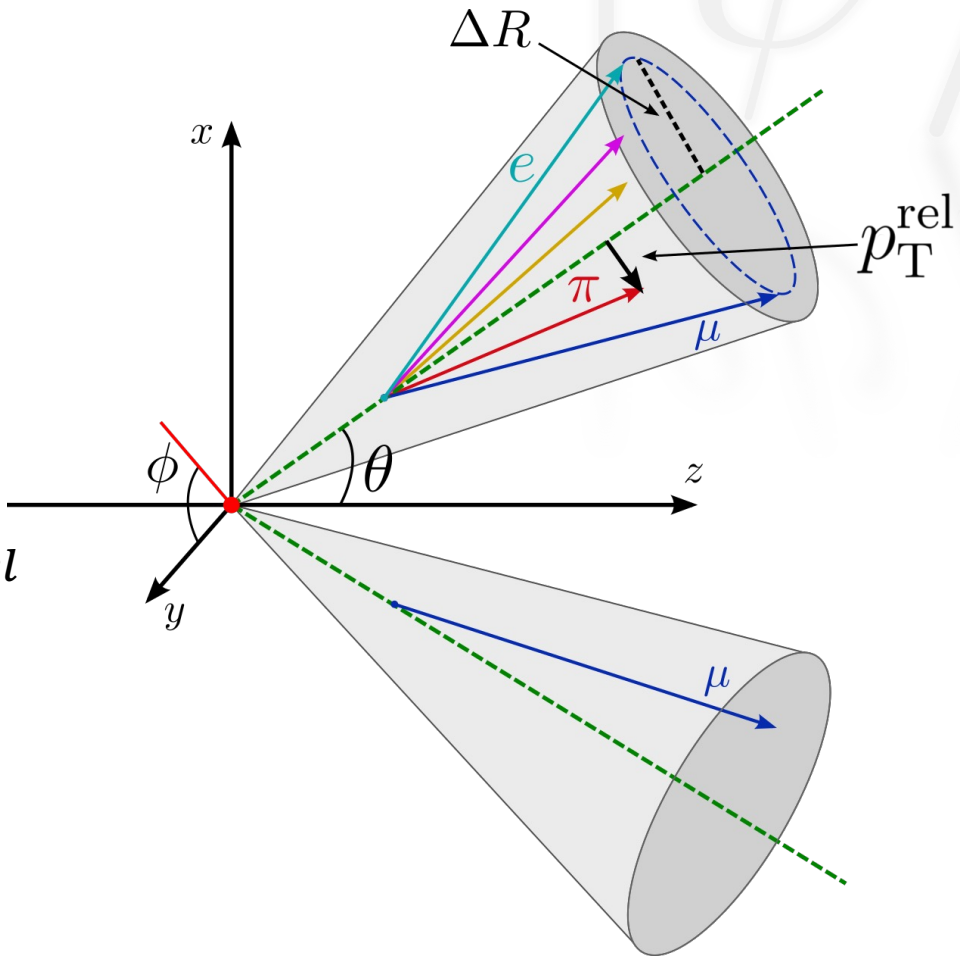
- b quarks are produced in pairs quark-antiquark with opposite charge
- Our task: identifying the **charge** of the b quark from the particle content of the jet
- *Exclusive* methods have been used… but they are not enough!



The Data: variables selection

- Official LHCb simulations opendata CERN*
- $pp \to b\bar{b}$ inclusive di-jet samples at $E = 13\ TeV$
- 5 types of charged (anti-)particles: muons, pions, kaons, electrons and protons
- For each particle we select 3 variables
  - Transverse momentum (relative to the jet axis) p_T^{rel}
  - Distance from the jet axis ΔR
  - Charge Q
- The global jet weighted charge is added

$$Q_{tot} = \frac{\sum_i p_T^i q^i}{\sum_i p_T^i}$$



* https://opendata.cern.ch/record/4910

The Data: two datasets

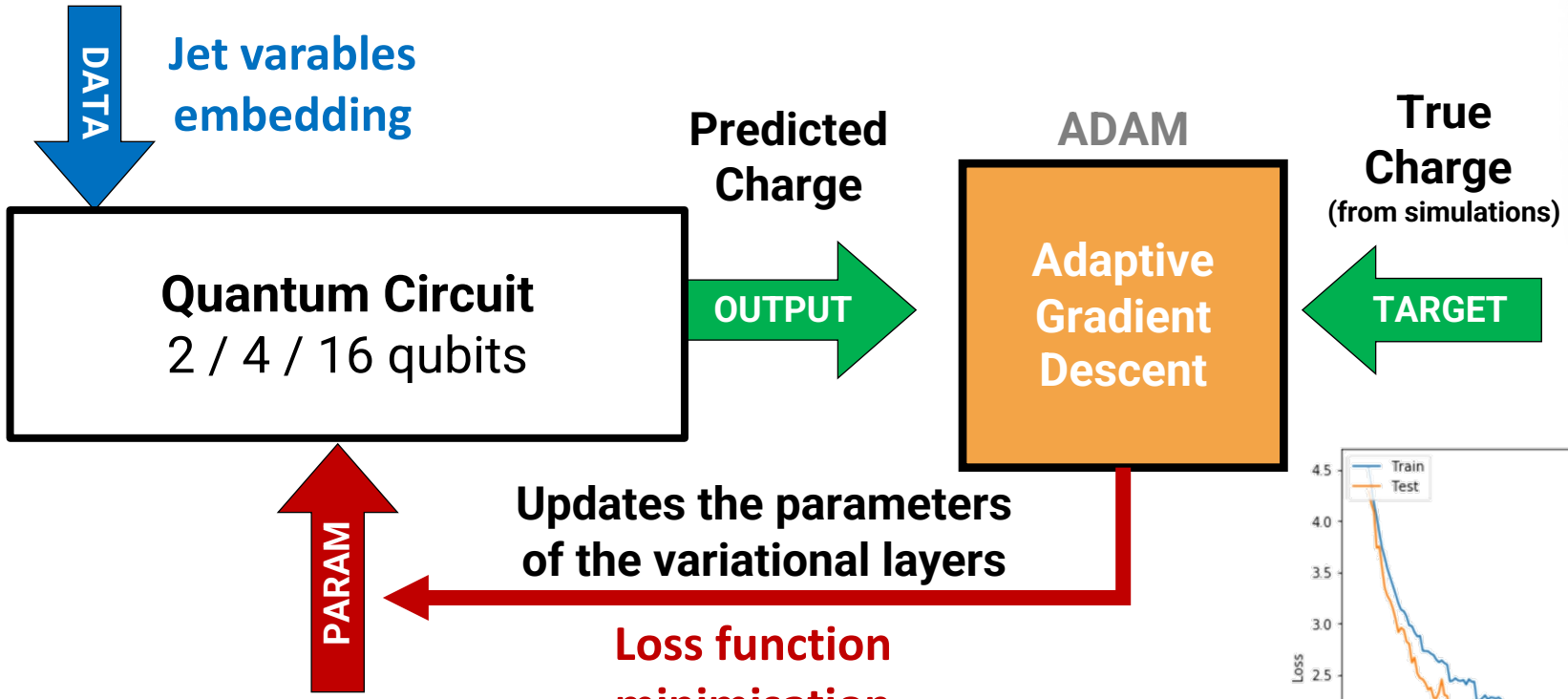
Muon Dataset

- Contains only the muon
- 4 variables only (3 from the muon + the global charge)
- Directly compares to exclusive methods like the *muon tagging*
- Easier task
- ~ 100.000 jets

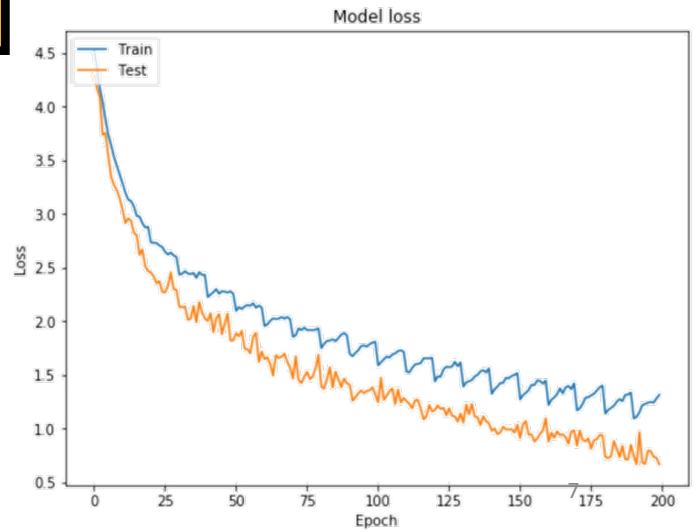
Complete Dataset

- Contains all the particles
- 16 variables (15 from the particles + the global charge)
- Compares to state-of-the-art taggers (DNN or xgboost)
- Harder task
- ~ 700.000 jets

The Model: Variational Quantum Classifier



$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

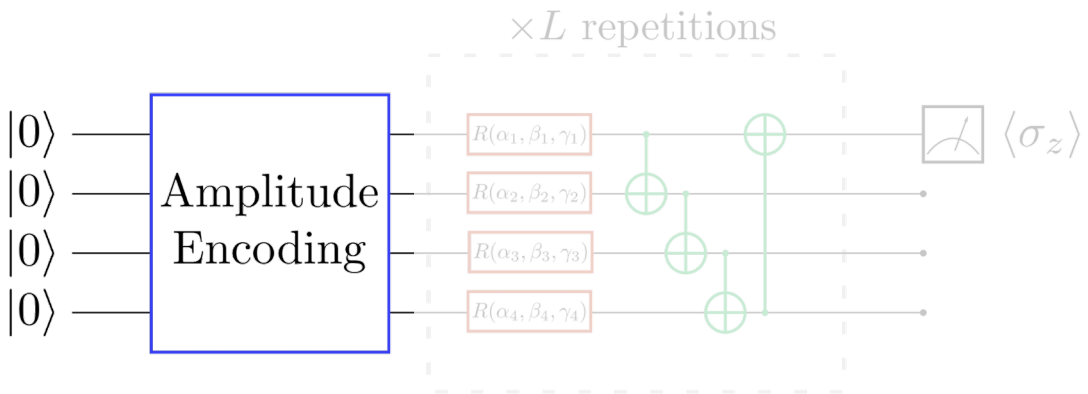


The Model: quantum embeddings

Amplitude encoding

Variables are embedded as amplitudes of a quantum state

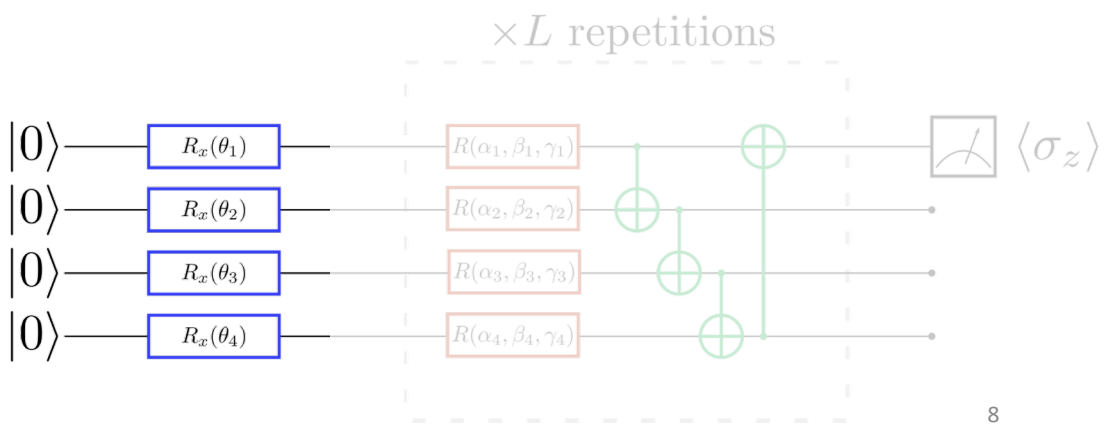
+ Pros:
  - N qubits $\rightarrow 2^N$ variables
  - Very convenient for high-dimensionality datasets
+ Cons:
  - Amplitudes are complex numbers
  - Requires arbitrary state preparation on hardware



Angle encoding

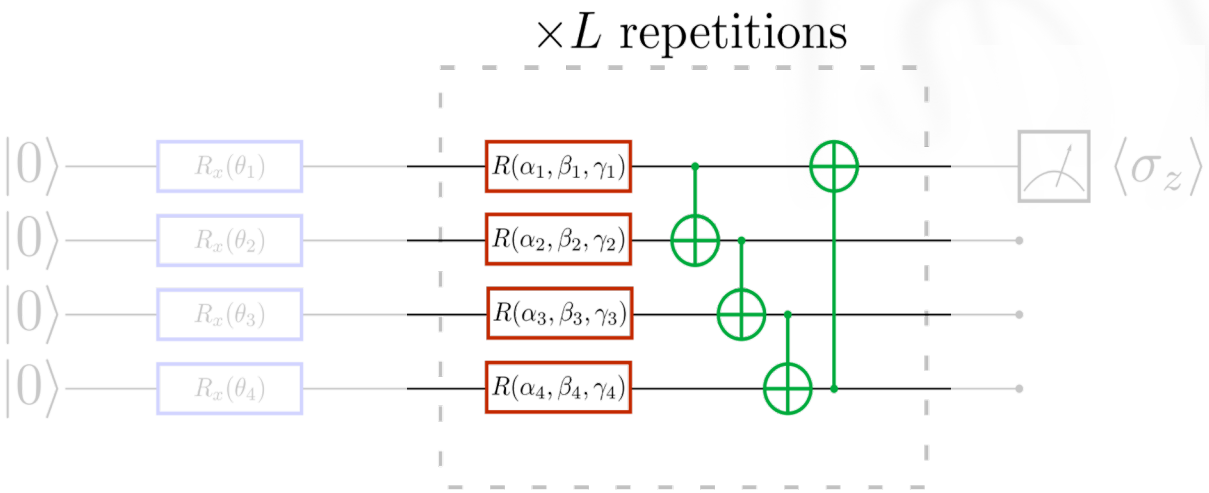
Variables are embedded as angles of rotational gates

+ Pros:
  - Hardware-friendly
  - Logical correspondence between variables and qubits
+ Cons:
  - N qubits $\rightarrow N$ variables
  - Not ideal for high-dimensionality datasets



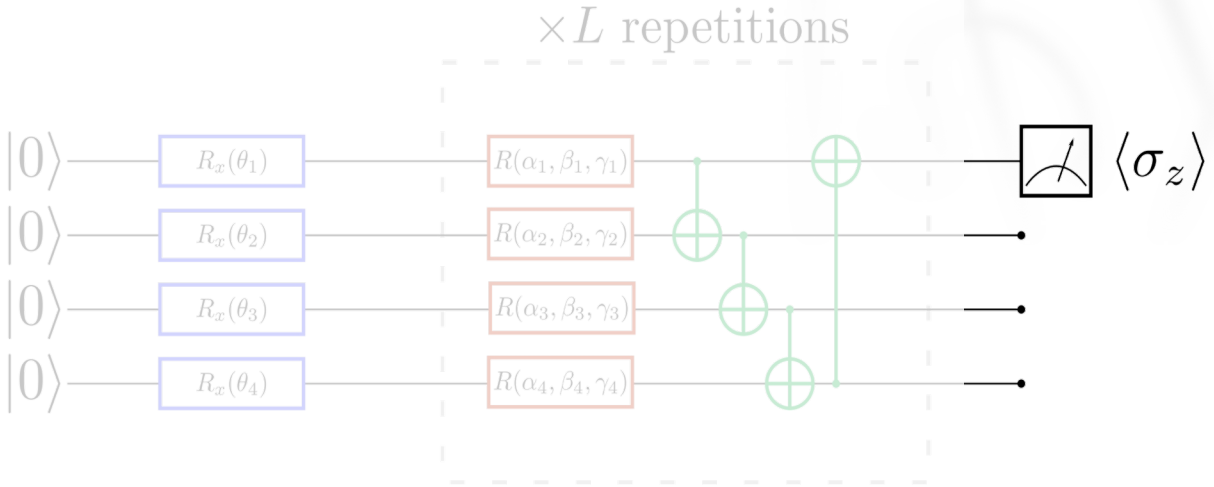
The Model: variational layers

- Strongly Entangling Layers ansaz
  - General rotational gates $R(\alpha, \beta, \gamma)$
  - Entangling CNOT gates in cyclic pattern
- Pros
  - Very expressive ansatz
  - N. of layers is tunable
- Cons
  - Too expressive ansatz (vanishing gradient issue)
  - Not very efficient on NISQ devices



The Model: measurement

- Measurement on the final state are mapped to charge probability predictions
- We measure $\langle \sigma_z^0 \rangle \in [-1, +1]$ and define the probabilities
  - $P_{\mathrm{b}} = \frac{1}{2}(\langle \sigma_z^0 \rangle + 1)$
  - $P_{\bar{\mathrm{b}}} = 1 - P_{\mathrm{b}}$

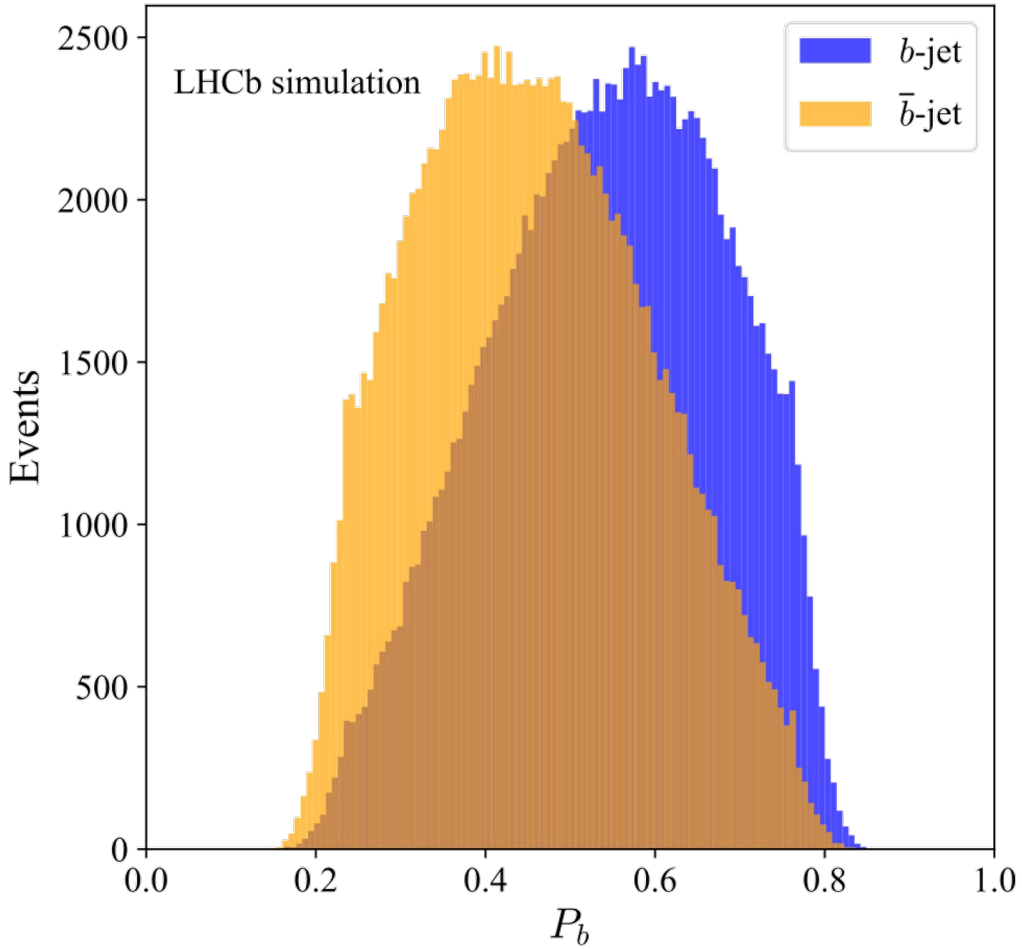


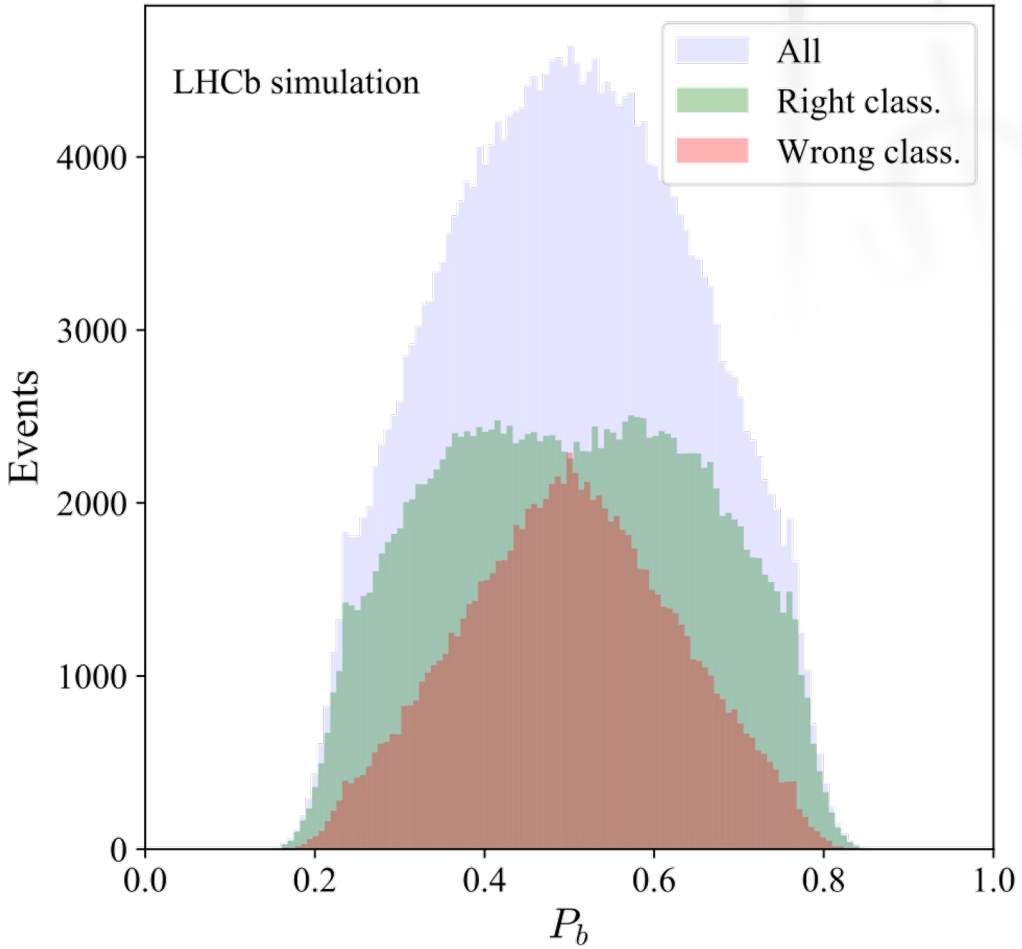
The Training

- Pennylane framework + Jax backend PENNYLANE + JAX
  - Automatic differentiation of quantum circuits
  - JIT execution on GPUs
- State-vector simulation (+ preliminary nosiy simulation)
- Back-propagation
- Distributed Stocastic Gradient Descent
  DataParallel model + Gradient Syncronization
  - Extensive use of Jax's `pmap` and `vmap` functions to parallelize the execution of the model
    Huge Speedup!
- 4 x Nvidia Tesla V100 on Cineca Marconi100

The Results

Distributions

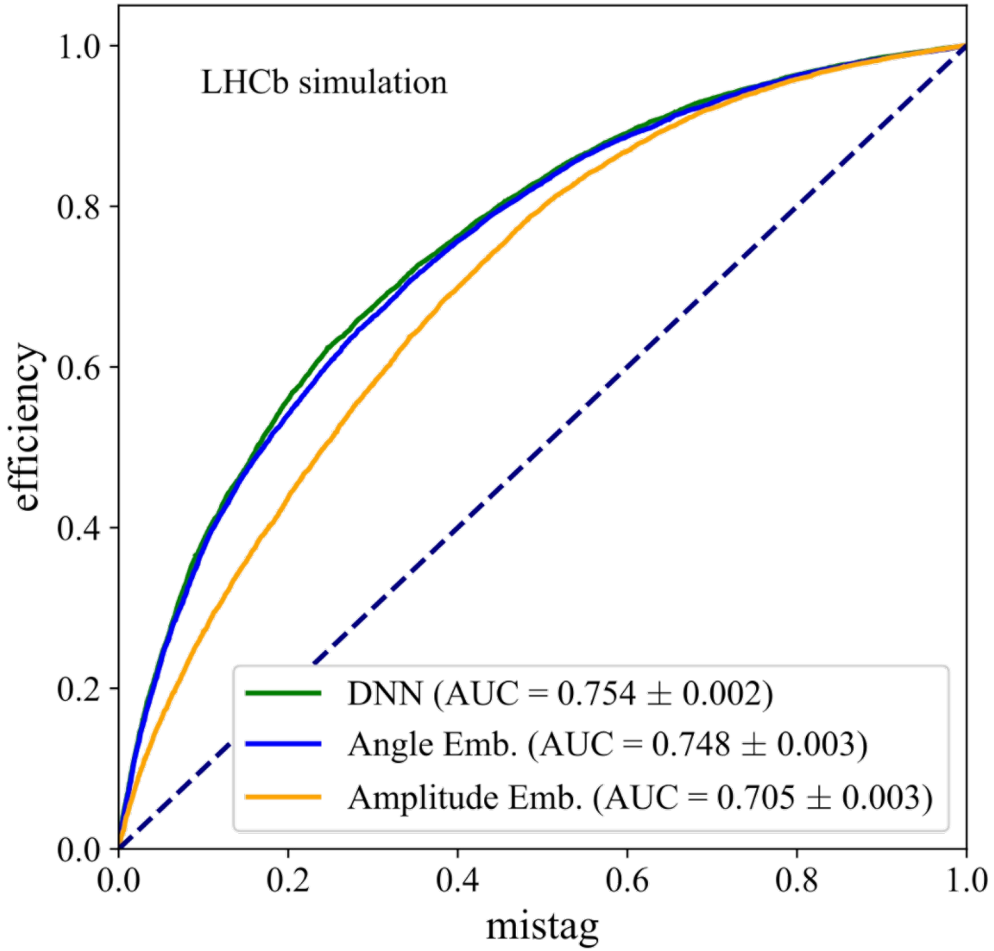




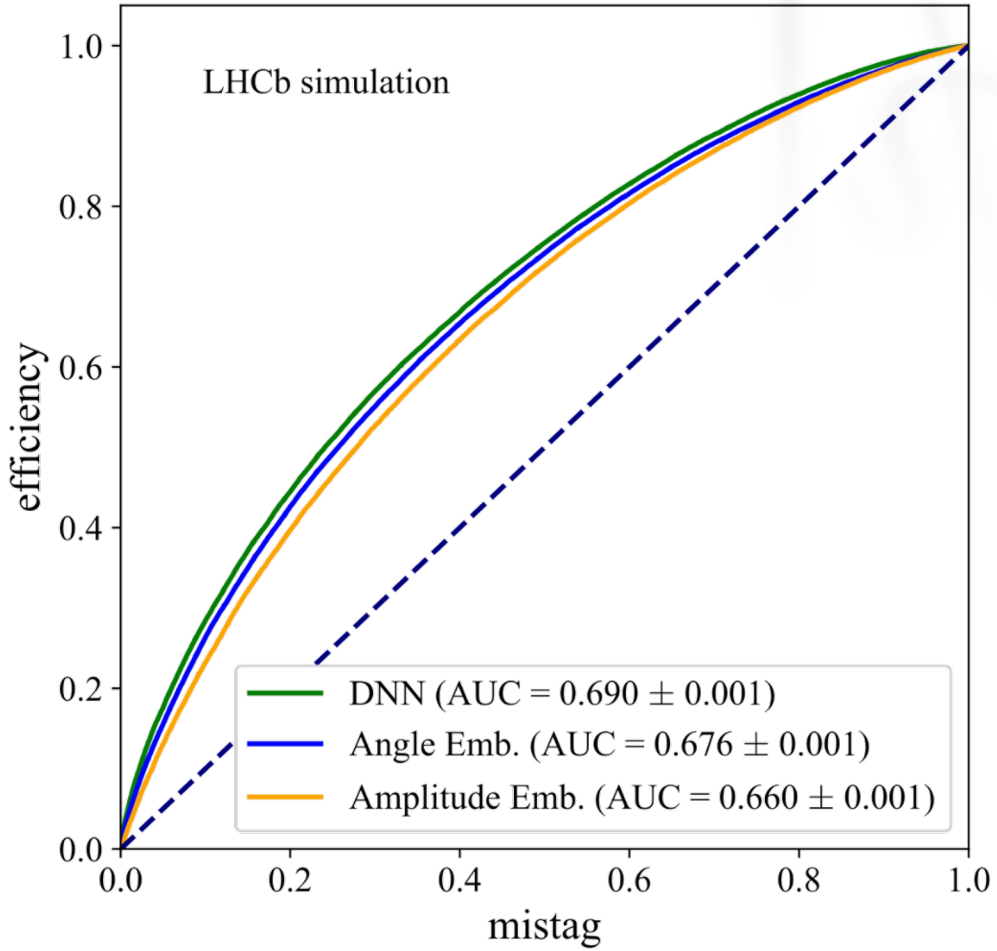
The Results

ROCs

Muon Dataset



Complete Dataset

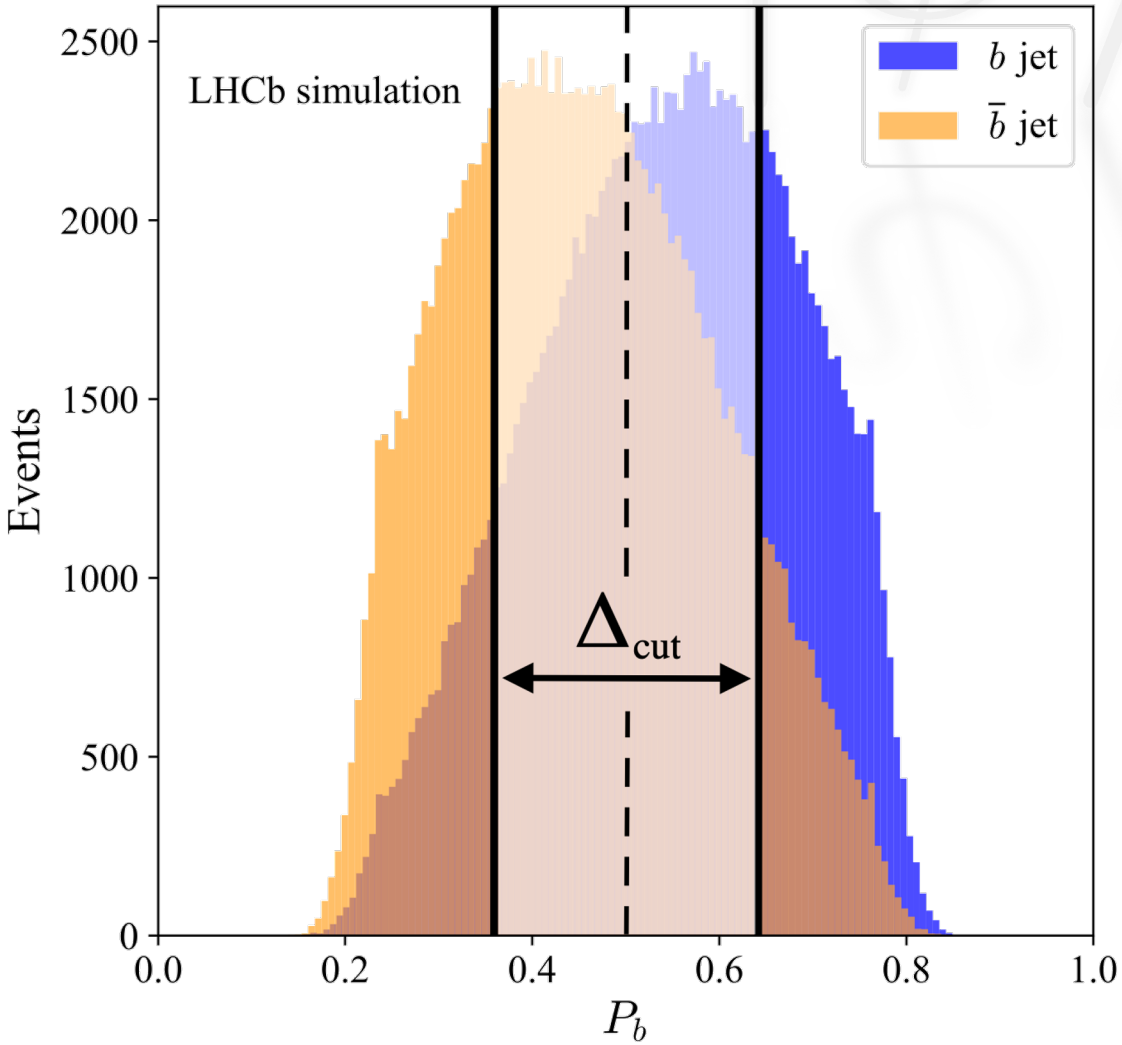


The Results

- The ideal performance metric for this problem is the **tagging power**

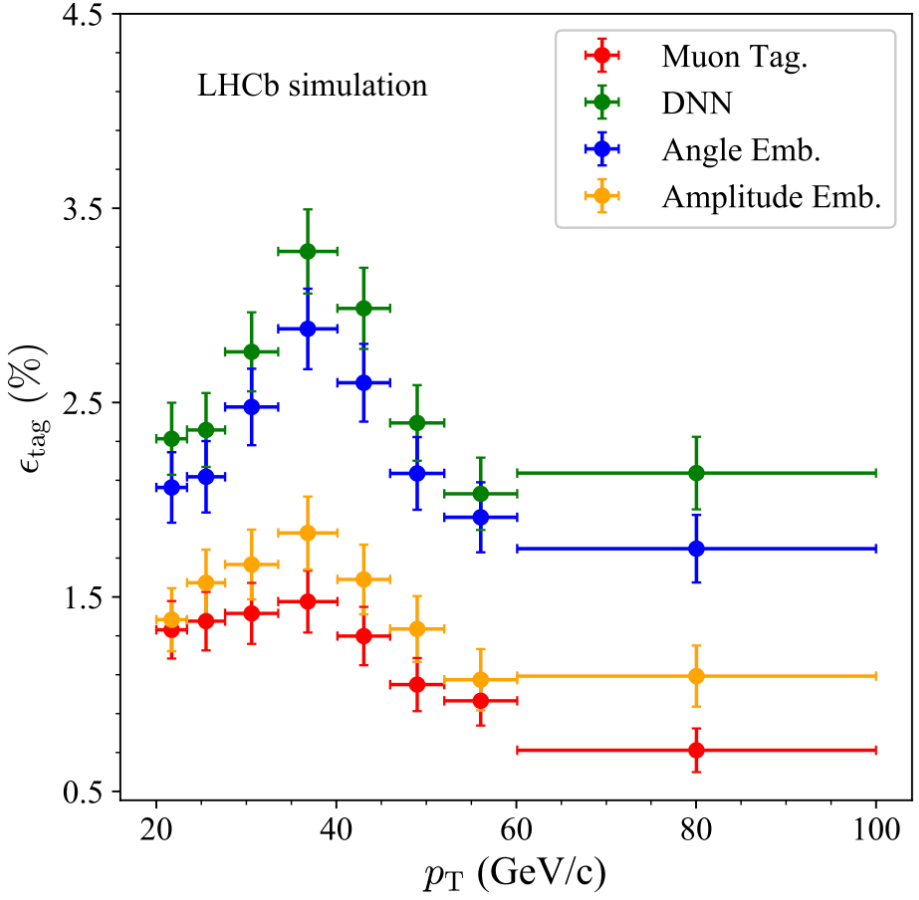
$$\epsilon_{tag} = \epsilon_{eff}(1 - 2\omega)^2$$

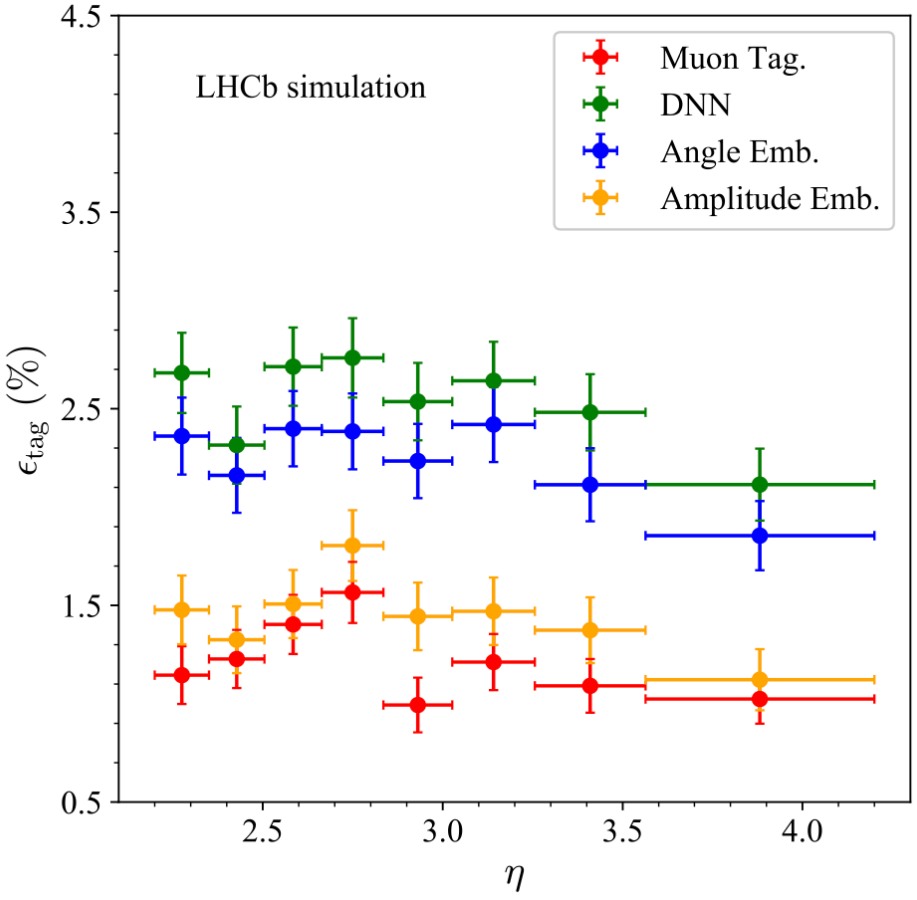
  - ϵ_{eff}: tagging efficiency
  - ω: mis-tag
- It represents the ***effective* fraction of jets** that contributes to a measurement
- We can perform a **cut** on the probability distribution to maximise the tagging power



The Results

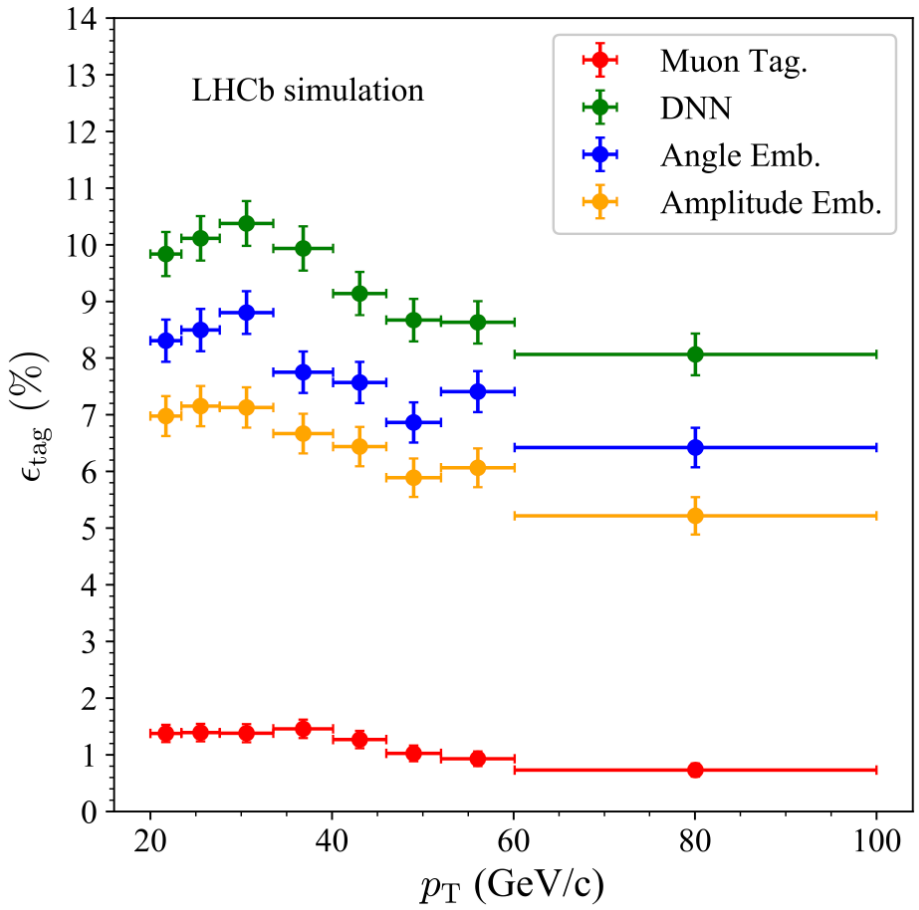
Muon Dataset

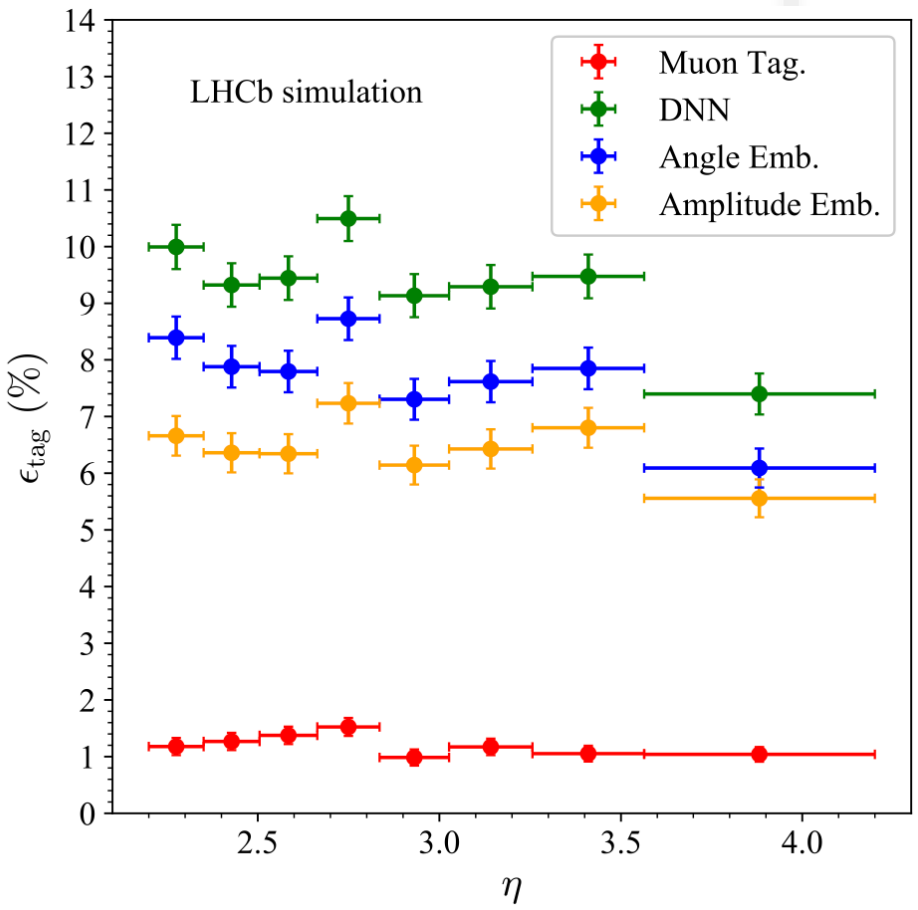




The Results

Complete Dataset





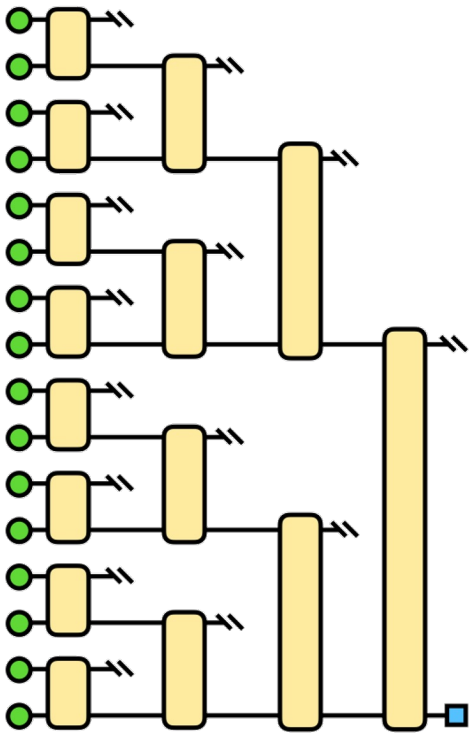
Summary

- First application of QML to b-jet tagging!
- Quantum and Classical ML have shown similar performances (DNN is still leading the comparison)

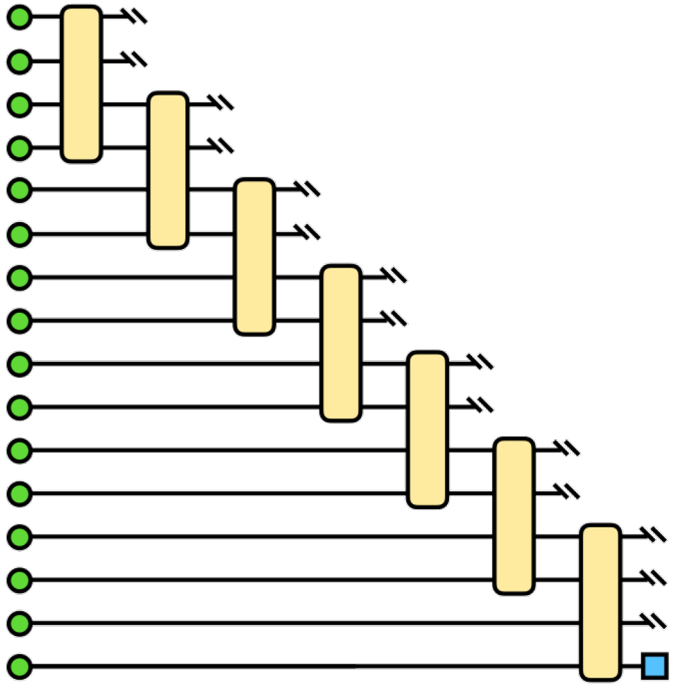
- Results obtained on simulators in ideal conditions… what would happen on hardware?
  - Hardware-efficient embedding (Angle embedding)
  - Hardware-inefficient variational ansatz (SEL) -> Not suitable for NISQ devices

Exploring new structures

Tree Tensor Network

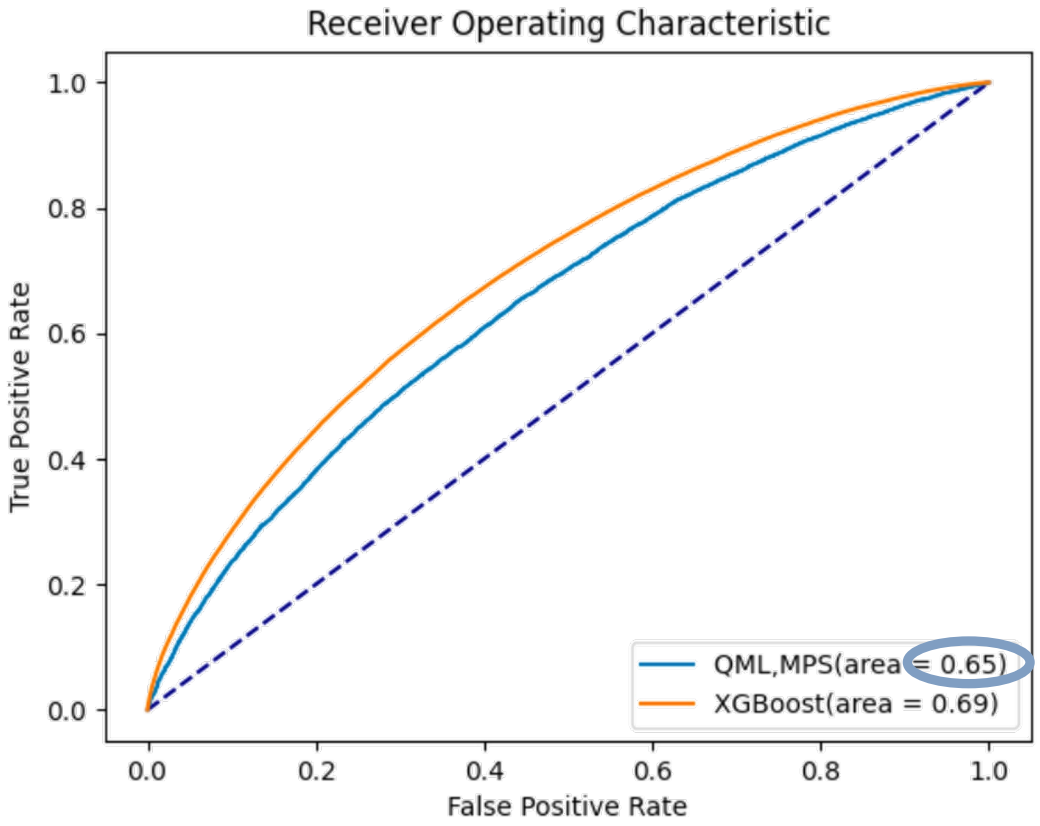


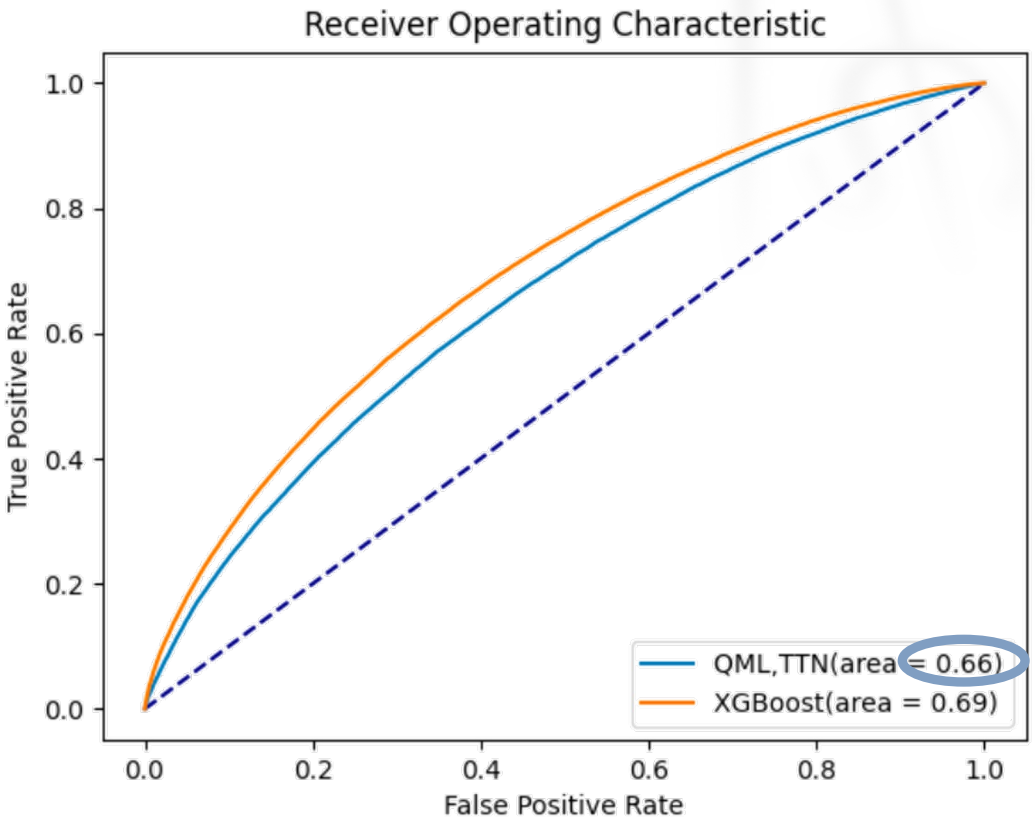
Matrix Product State



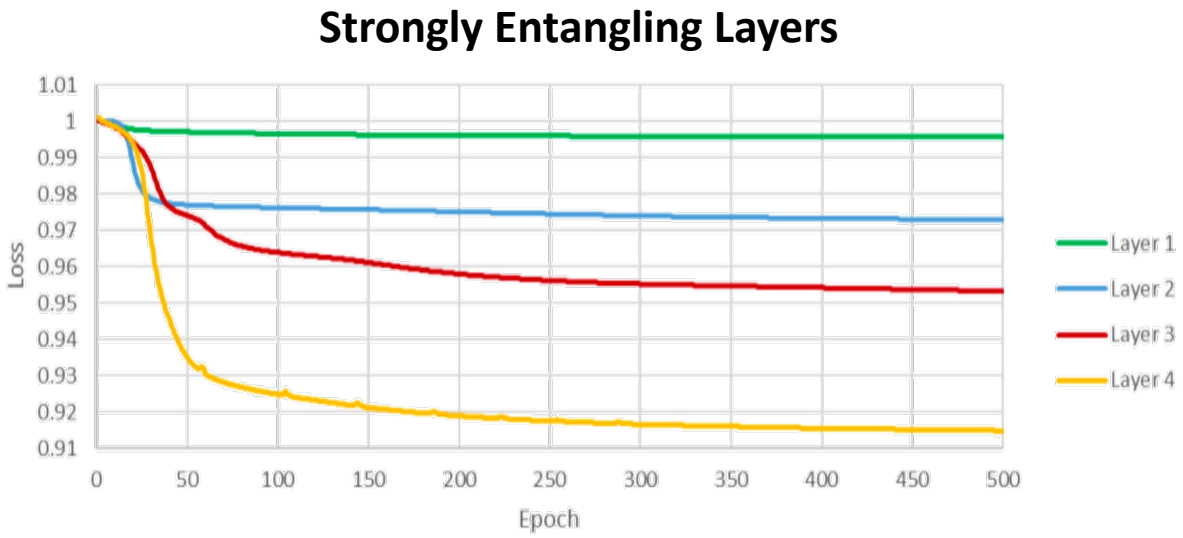
- Ansatz inspired by tensor networks
  - No barren plateau issue
  - Can be made hardware-efficient
- **Can they compete with SEL?**
  - Bachelor's thesis of Alexander Leonidas (UM)

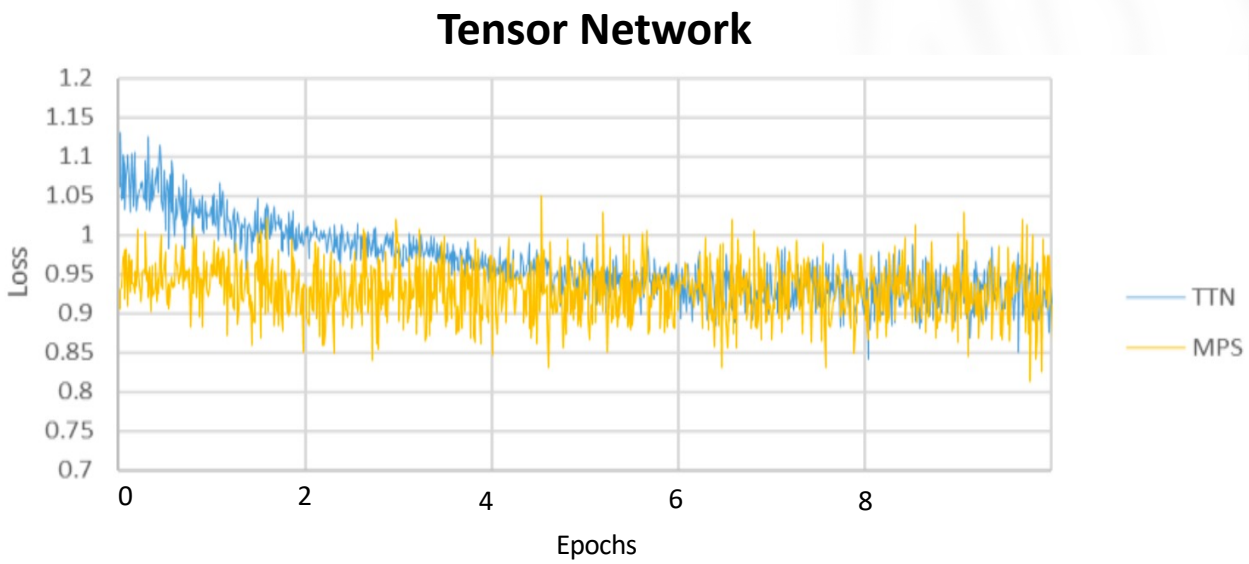
…yes


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
QML,MPS(area = 0.65)
XGBoost(area = 0.69)


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
QML,TTN(area = 0.66)
XGBoost(area = 0.69)

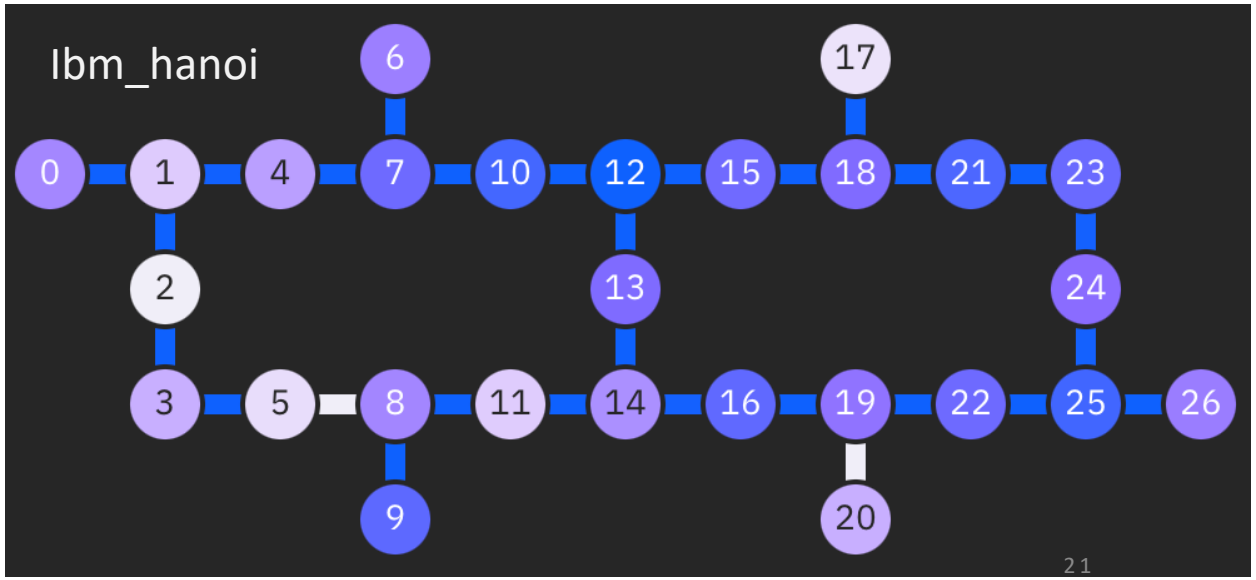
…and they're easier to train





Next step: going on hardware

- Andrew Spiro is starting his thesis project @ UM:
  - Optimise the TTN/MPS ansatz execution on IBM Q hardware
  - Train the optimised model on simulators (training on large datasets is still too expensive on hardware)
  - Run the inference on IBM Q hardware
    - Pennylane can interface with Qiskit
    - Access to IBM Q hardware via SURF



Thanks for the attention