A Quantum Graph Neural Network Approach to Particle Track Reconstruction

Connecting The Dots Workshop

Princeton University, Princeton, USA ***(Now Virtual)***

Cenk Tüysüz

Middle East Technical University, Ankara, Turkey

20 / 04 / 2020

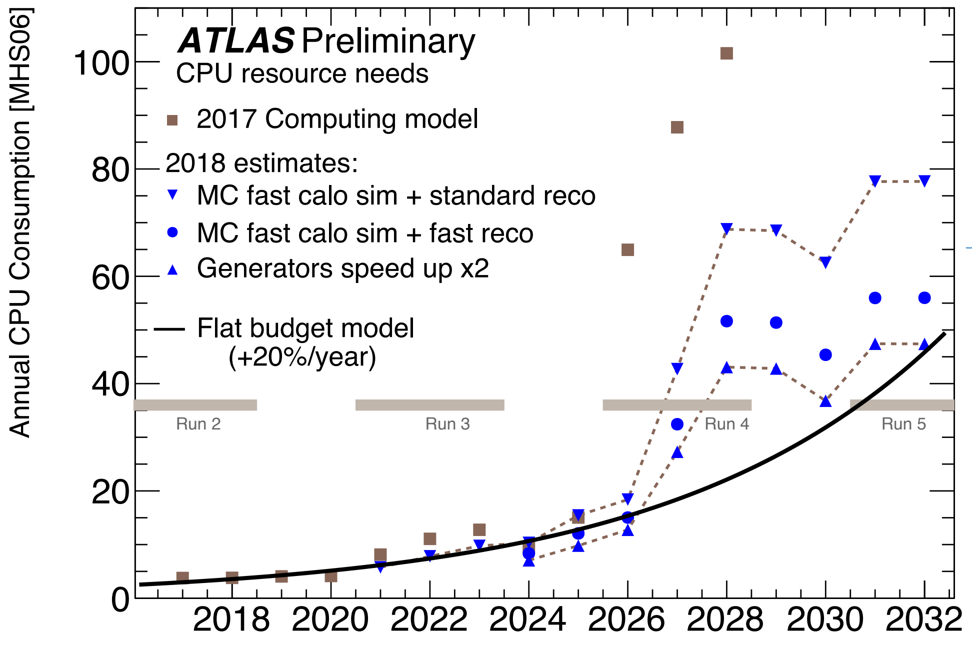
Outline

- Particle Track Reconstruction and TrackML Challenge
- Hep.TrkX Graph Neural Network approach
- Quantum Graph Neural Network approach

High Luminosity LHC

and Particle Track Reconstruction

High Luminosity upgrade of LHC brings many computational challenges.



ATLAS computing model projections for Phase-2

Particle Track Reconstruction to be much harder!

	Run -1	Run -2	Run -3
μ	21	40	150-200?
Tracks	~280	~600	~7-10k

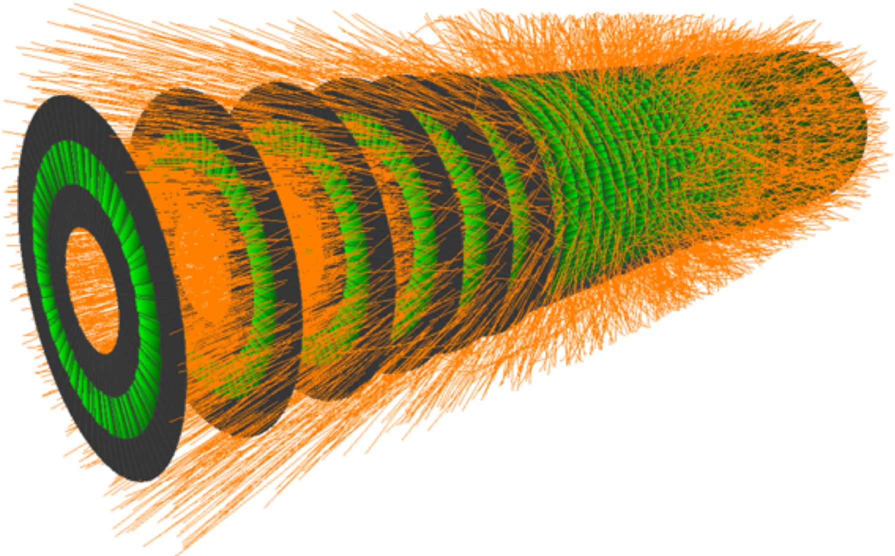
μ: *Average number of interactions per bunch crossing*

H. Gray, Track reconstruction in the ATLAS experiment, 2016.

TrackML Challenge

A Public Machine Learning Challenge for Particle Tracking

https://www.kaggle.com/c/trackml-particle-identification/overview



#	Δpub	Team Name	Notebook	Team Members	Score	Entries	Last
1	—	Top Quarks			0.92182	10	1y
2	—	outrunner			0.90302	9	1y
3	—	Sergey Gorbunov			0.89353	6	1y

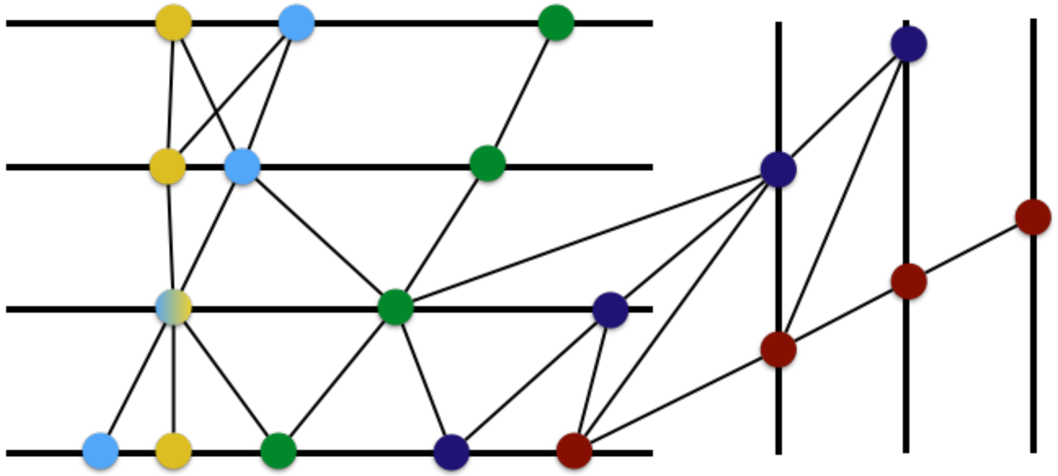
Hep.TrkX GNN

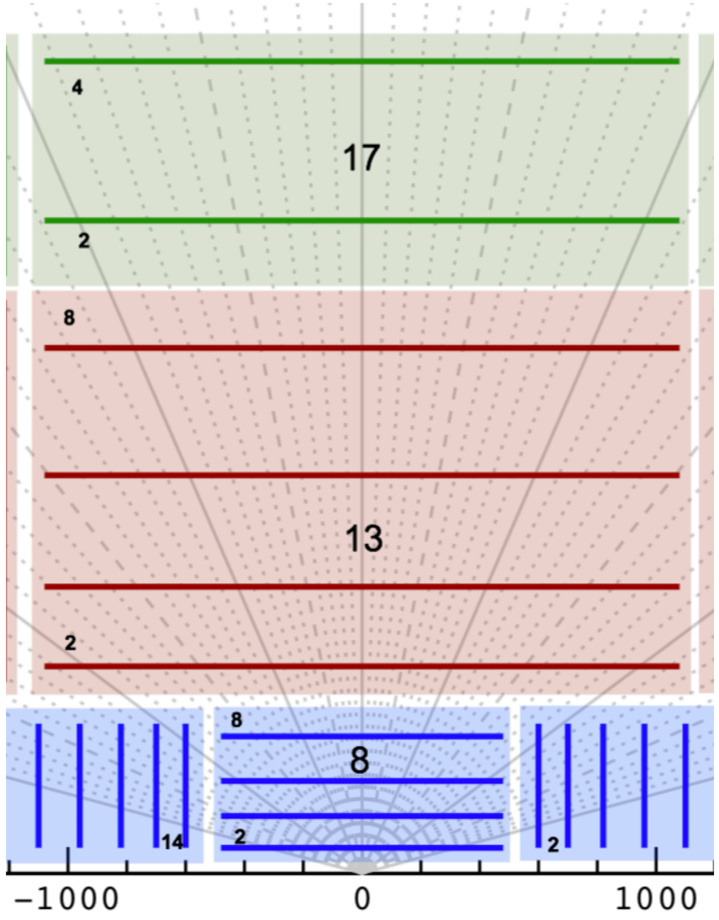
Novel deep learning methods for track reconstruction

Steven Farrell[1,*], *Paolo* Calafiura[1], *Mayur* Mudigonda[1], Prabhat[1], *Dustin* Anderson[2], *Jean-Roch* Vlimant[2], *Stephan* Zheng[2], *Josh* Bendavid[2], *Maria* Spiropulu[2], *Giuseppe* Cerati[3], *Lindsey* Gray[3], *Jim* Kowalkowski[3], *Panagiotis* Spentzouris[3], and *Aristeidis* Tsaris[3]

[1]Lawrence Berkeley National Laboratory
[2]California Institute of Technology
[3]Fermi National Accelerator Laboratory

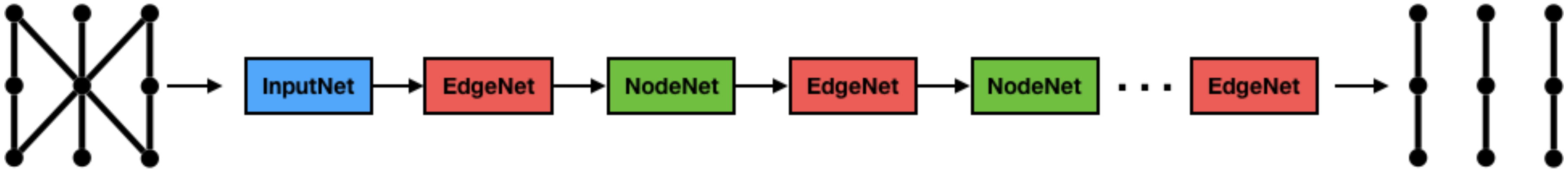
https://arxiv.org/abs/1810.06111

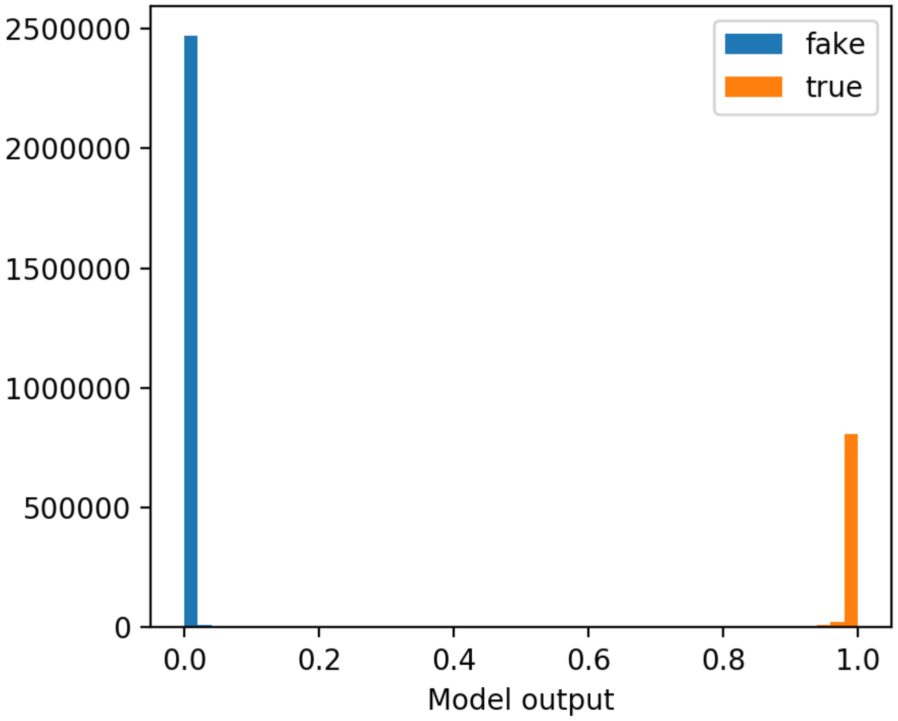


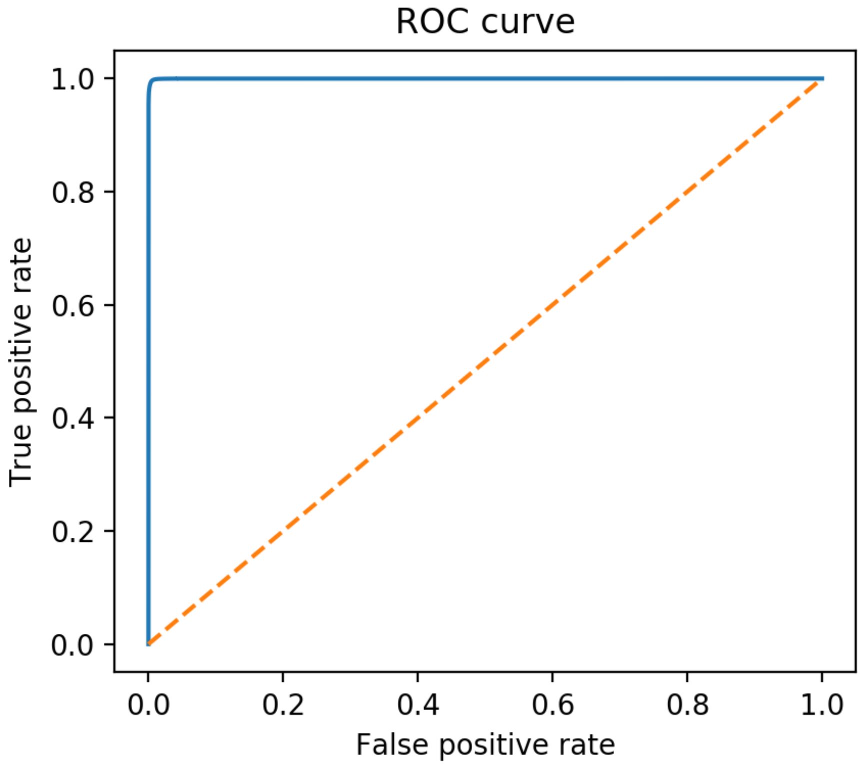


Hep.TrkX GNN

Promising Results







Model Scores:

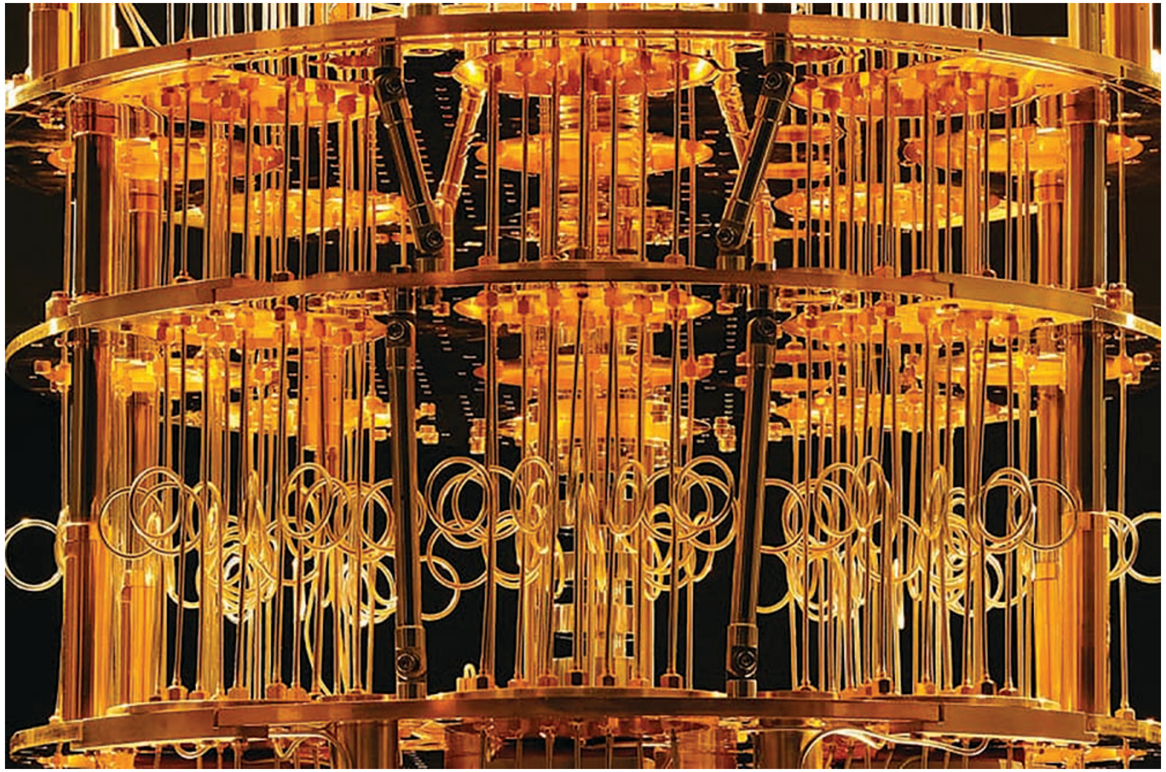
Purity: 99.5%
Efficiency: 98.7%
Overall Accuracy: 99.5%

with 0.5 threshold

Quantum Computing

Quantum computing allows a new way of computation for certain problems including;

- Prime number factorization
- Solving Linear Equations
- Machine Learning!



Credit: IBM Research

Quantum Classifiers

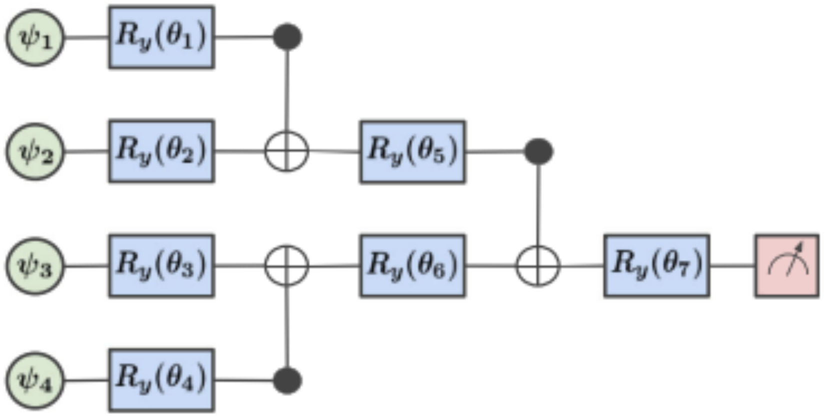
Hierarchical quantum classifiers

Edward Grant[1,2], Marcello Benedetti[1,3], Shuxiang Cao[4,5], Andrew Hallam[6,7], Joshua Lockhart[1], Vid Stojevic[8], Andrew G. Green[6] and Simone Severini[1]

Table 3. Binary classification accuracy on the MNIST dataset

Classifier	Unitaries	Rotations	Is > 4	Is even	0 or 1	2 or 7
TTN	Simple	Real	65.59 ± 0.57	72.17 ± 0.89	92.12 ± 2.17	68.07 ± 2.42
TTN	General	Real	74.89 ± 0.95	83.13 ± 1.08	99.79 ± 0.02	97.64 ± 1.60
MERA	General	Real	75.20 ± 1.51	82.83 ± 1.19	99.84 ± 0.06	98.02 ± 1.40
Hybrid	General	Real	76.30 ± 1.04	83.53 ± 0.21	**99.87** ± 0.02	98.07 ± 1.46
TTN	Simple	Complex	70.90 ± 0.73	80.12 ± 0.64	99.37 ± 0.12	94.09 ± 3.37
TTN	General	Complex	77.56 ± 0.45	83.53 ± 0.69	99.77 ± 0.02	97.63 ± 1.48
MERA	General	Complex	**79.10** ± 0.90	**84.85** ± 0.20	99.74 ± 0.02	**98.86** ± 0.07
Hybrid	General	Complex	78.36 ± 0.45	84.38 ± 0.28	99.78 ± 0.02	98.46 ± 0.19
Logistic	N/A	N/A	70.70 ± 0.01	81.72 ± 0.01	99.53 ± 0.01	96.17 ± 0.01

Mean test accuracy and one standard deviation are reported for TTN, MERA, and hybrid classifiers with five different random initial parameter settings using two different types of unitary parametrization. Hybrid classifiers consist of pre-training a TTN classifier and that transforming it into a MERA classifier by training additional unitaries. Bold values indicate the best result for each classificaiton task



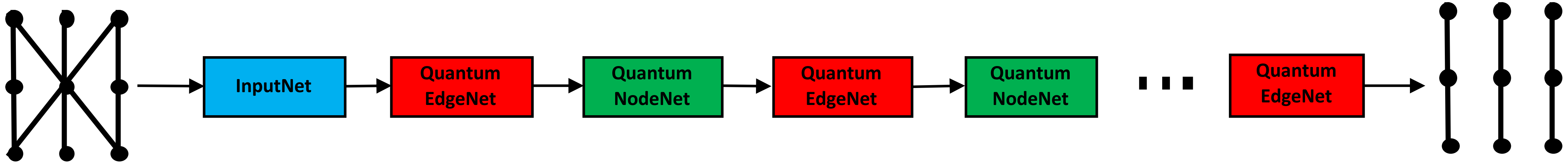
Our Work

Introducing Quantum Graph Neural Networks for Particle Track Reconstruction

- First results were showcased at CHEP 2019, proceeding submitted: https://arxiv.org/abs/2003.08126

Aim:

- Using Gate Level Quantum Computers for Particle Track Reconstruction by altering an already good performing Graph Neural Network Approach (Hep.TrkX)

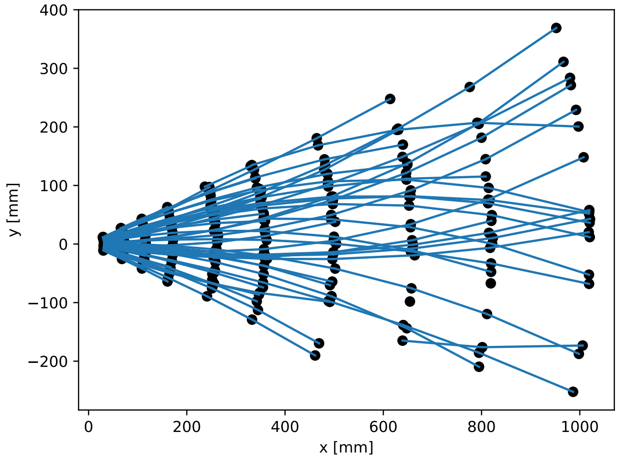


Preprocessing the Data

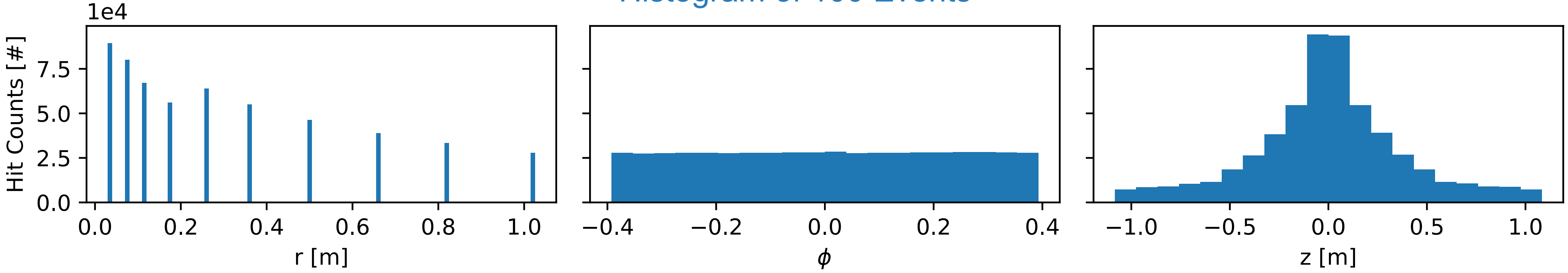
- Each event is divided into 8 segments in η and 2 segments in z directions.
- 1% of TrackML data is used.
- Following cuts are applied to the data to construct graphs:

$\lvert pT \rvert$	> 1 GeV
$\Delta\emptyset/\Delta r$	< 0.0006
z_0	< 100 mm
η	[-5, 5]

Example Subgraph



Histogram of 100 Events



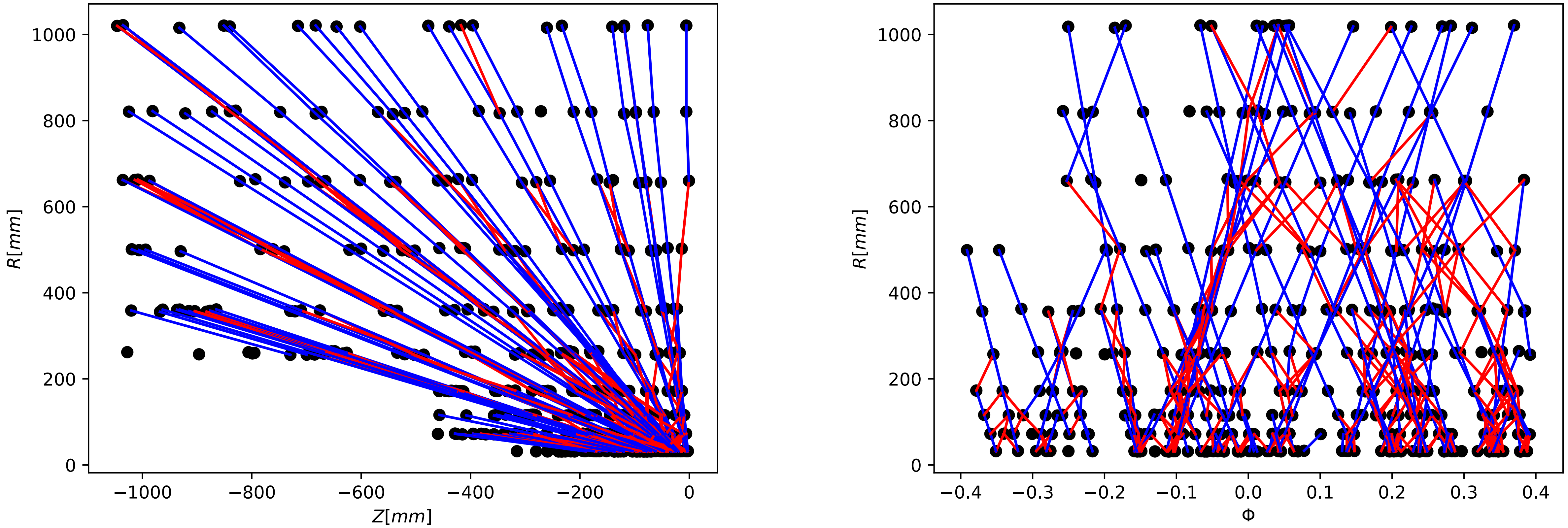
Plotting the Data

In Cylindrical Coordinates

Blue: After preprocessing with Hep.TrkX methods

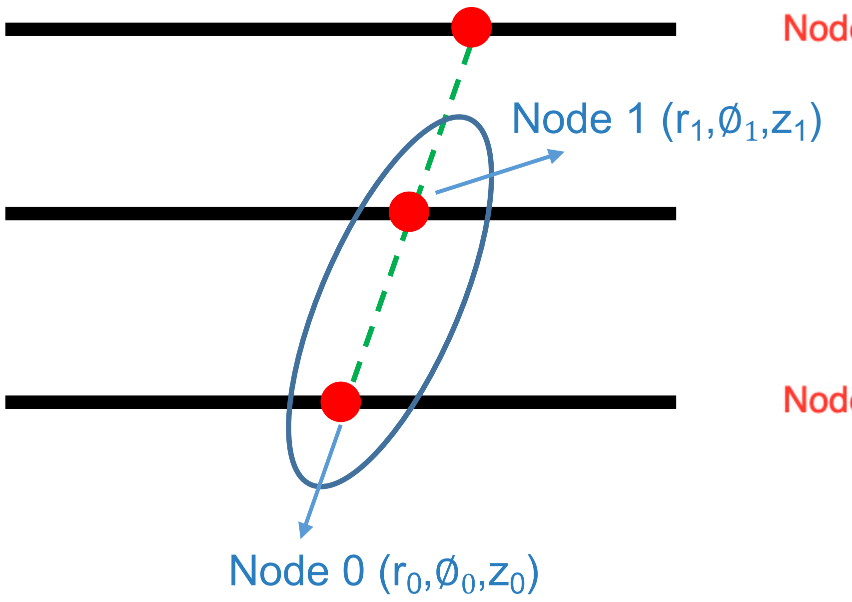
Red: Ground Truth

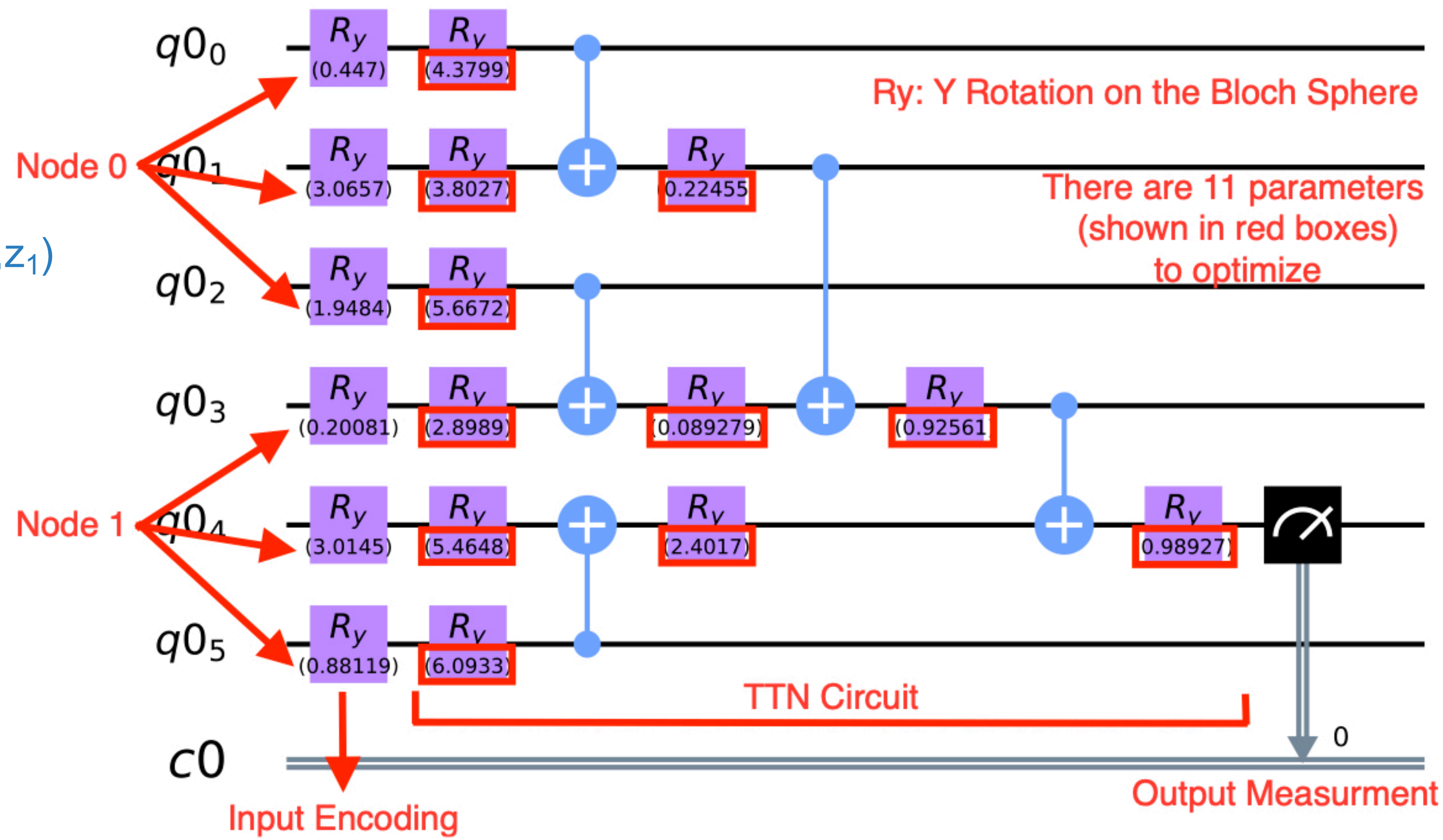
1/16 of an event



A Quantum Classifier

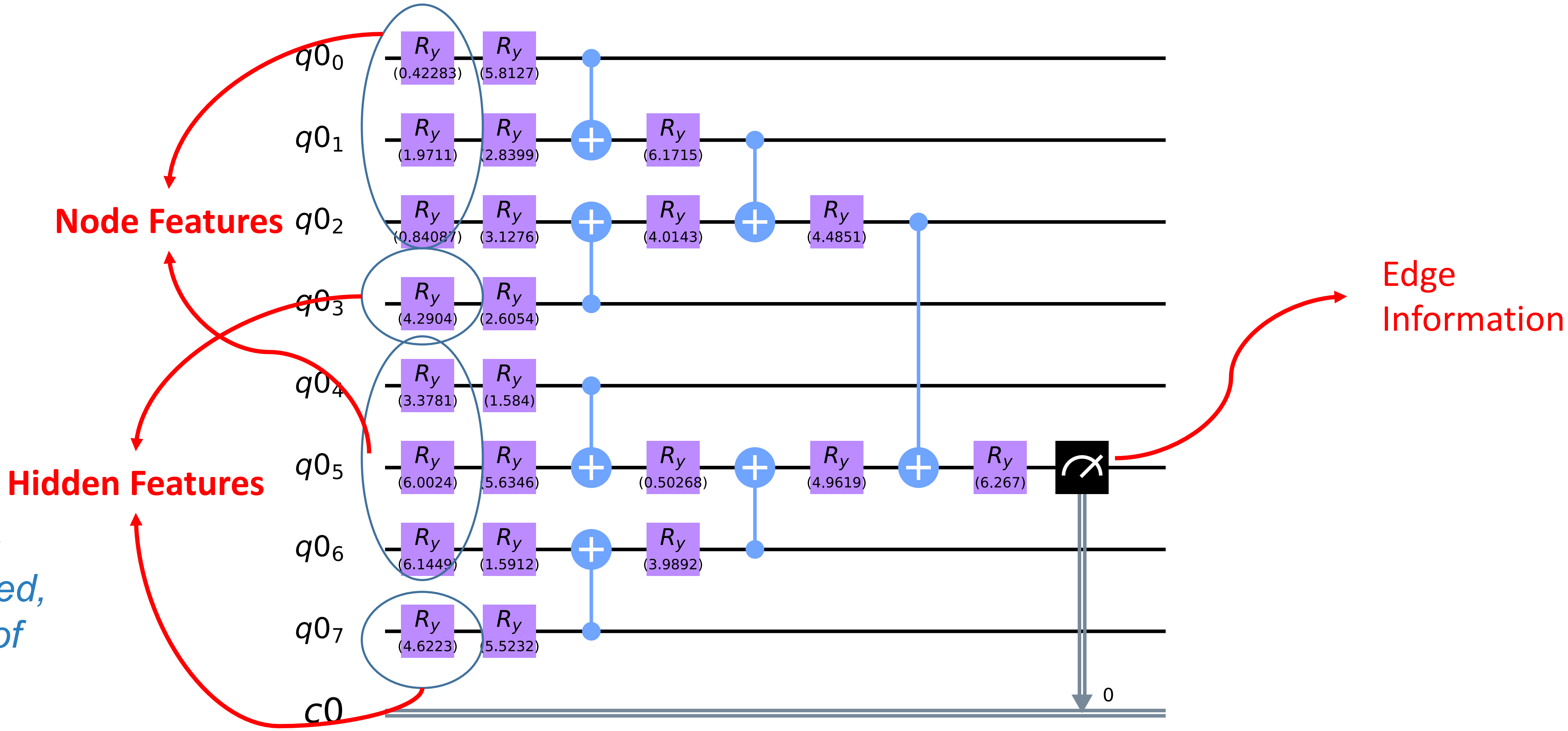
How does it work?





Quantum Networks

Quantum Edge Network

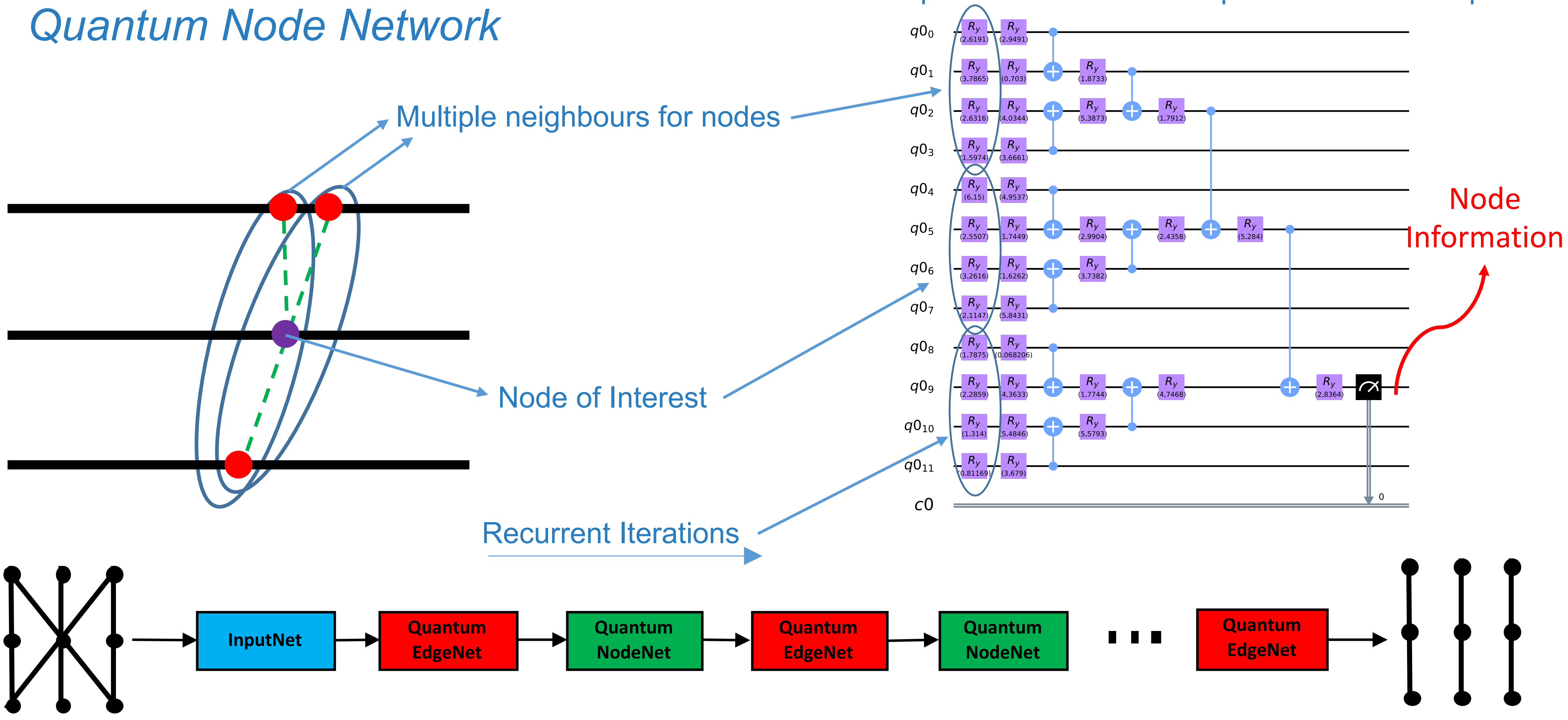


Number of qubits is required to be increased, to increase the size of hidden dimension.

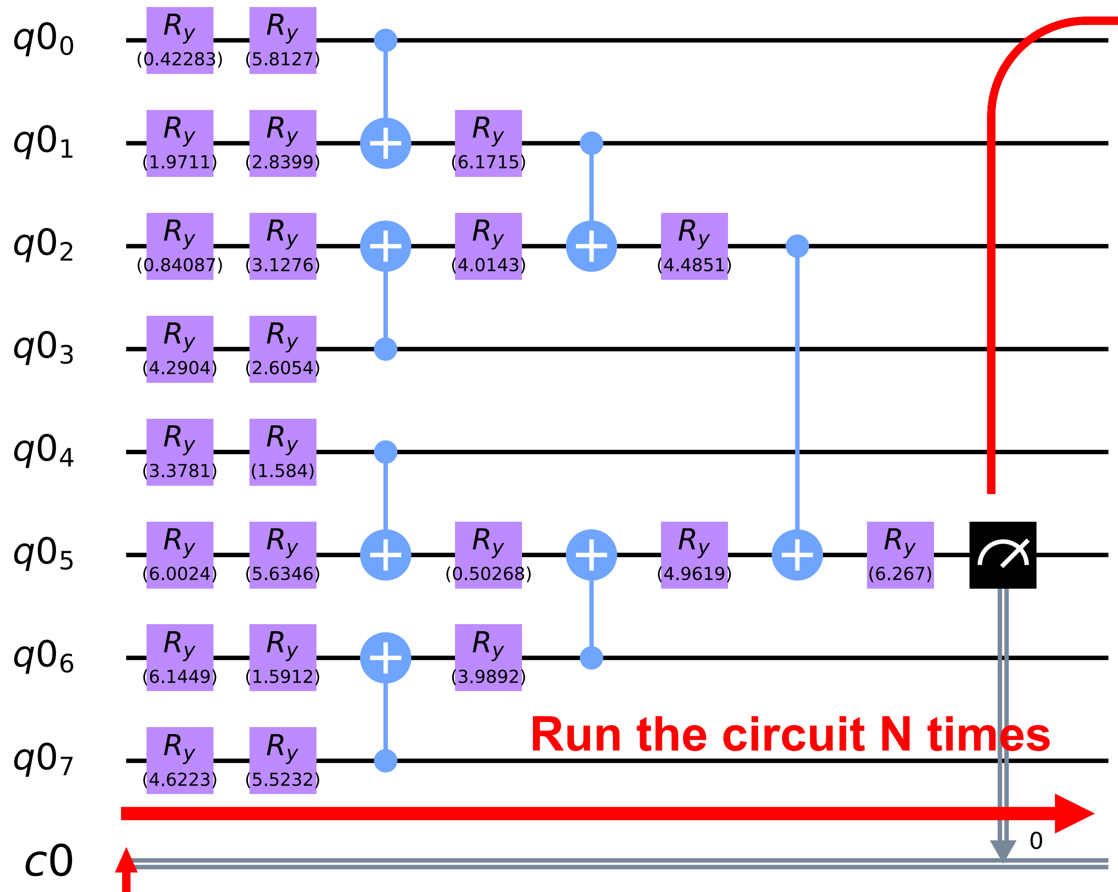
Quantum Networks

Quantum Node Network

A circuit is setup for each possible neighbor.
2 independent circuits are required for this example.



Training the Network



Averaging the measurement outcomes gives a probability of being an edge.

Calculate an error from the ground truth data.

Calculate the gradients of the parameters using parameter shift rule.

Update the parameters:

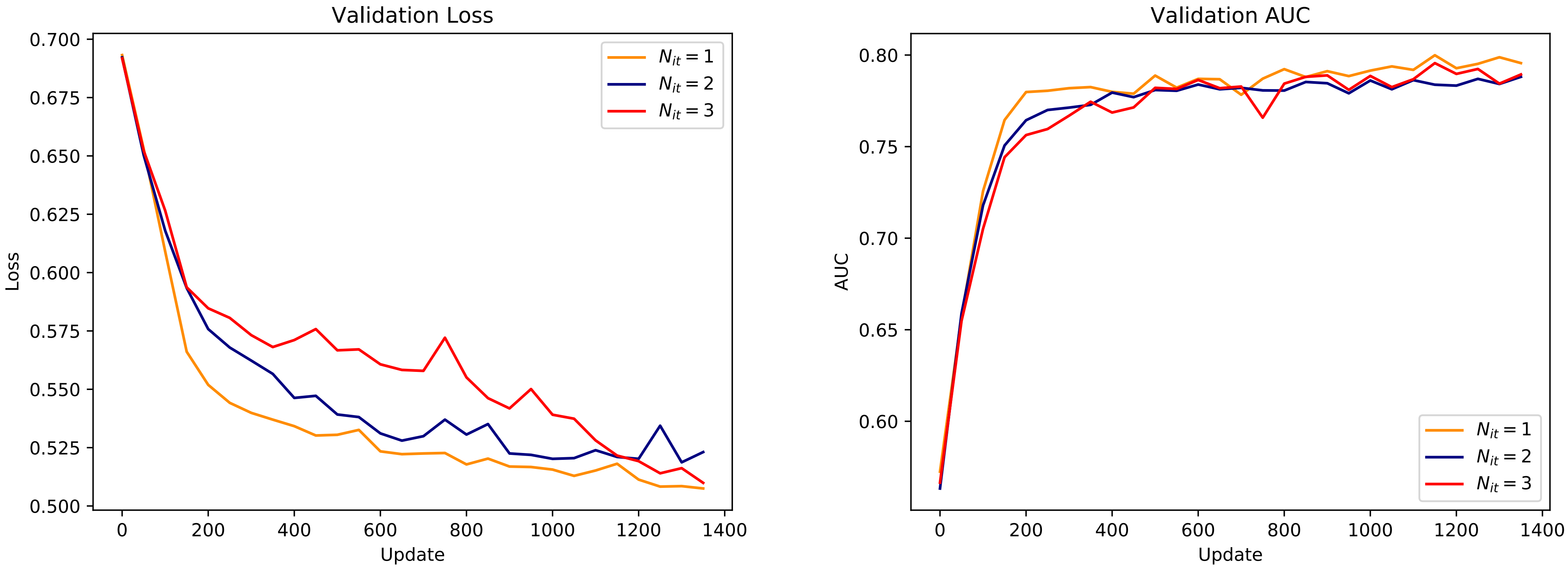
$$\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \, \nabla_{\theta} \mathcal{J}(\theta)$$

Repeat!

Training Results of the QGNN

Training with Different Amount of Iterations (Single epoch)

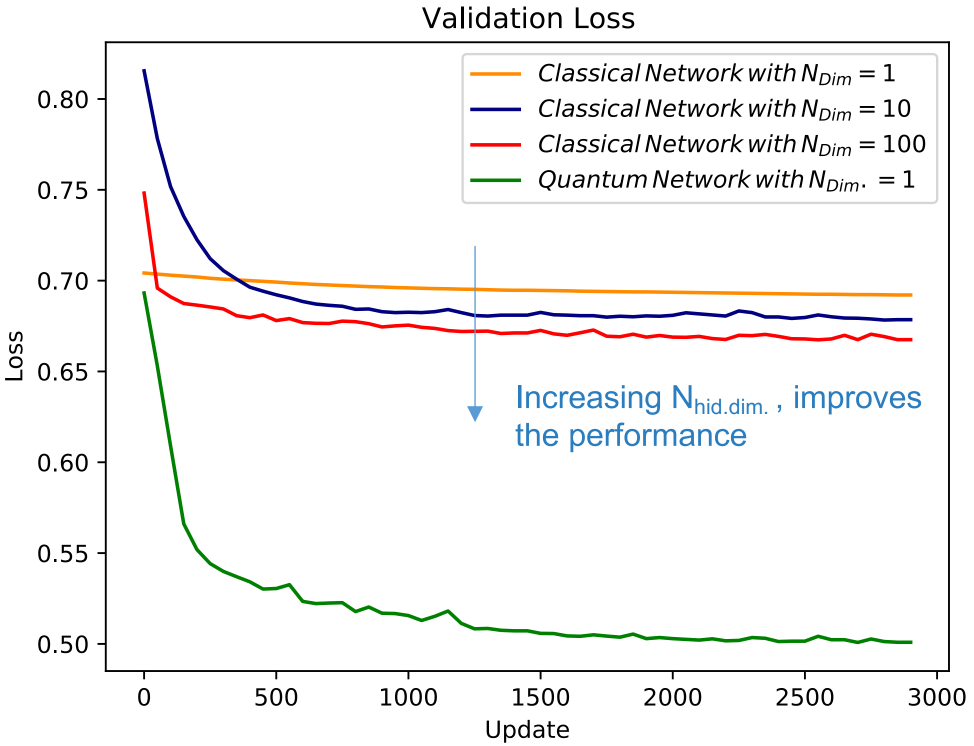
AUC: Area Under ROC, a measure of accuracy for different thresholds. AUC = 1.0 means perfect score
See slide 25 for details.

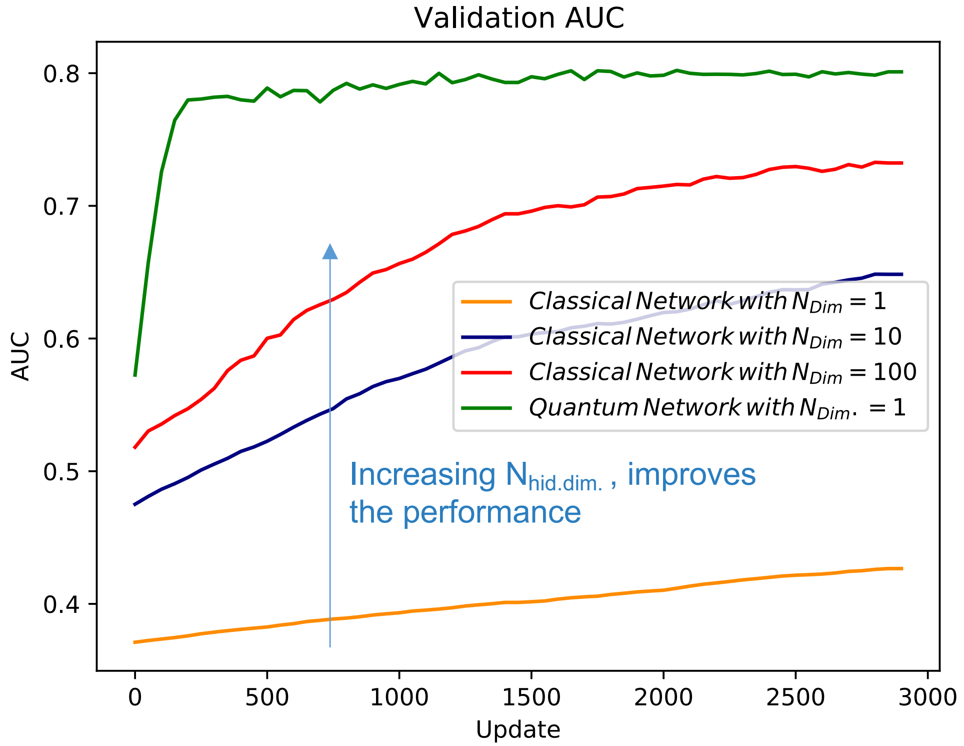


Training set: 1400 subgraphs, Validation set: 200 subgraphs,
using ADAM, binary cross entropy, lr = 0.01, shots =1000. **Hidden Dimension Size = 1**.

Training Results of the QGNN

Comparison to Simple Classical Networks (2 epochs)





Same Dataset. Classical Networks have x100 learning rate.

Simple experiments with Classical Networks show the potential for the Quantum Network.

Conclusion

First results are promising. With this approach we are optimistic to get better results.

There are things to explore;

- More layers (iterations)
- More hidden features (qubits)
- Different Quantum Networks/Architectures
- Testing with more data

Simulation times of Quantum Networks are limiting fast development, due to very long run times $\mathcal{O}$(weeks).

Contributors

C. Tüysüz[1,2], F. Carminati[3], B. Demirköz[1], D. Dobos[4,6], F. Fracas[3], K. Novotny[4], K. Potamianos[4,5], S. Vallecorsa[3], J.R. Vlimant[7]

[1]Middle East Technical University, Ankara, Turkey, [2]STB Research, Ankara, Turkey, [3]CERN, Geneva, Switzerland, [4]gluoNNet, Geneva, Switzerland, [5]DESY, Hamburg, Germany, [6]Lancaster University, Lancaster, UK, [7]California Institute of Technology, Pasadena, California, USA

QUESTIONS?

Email: ctuysuz@cern.ch

Twitter: @cenk_tuysuz

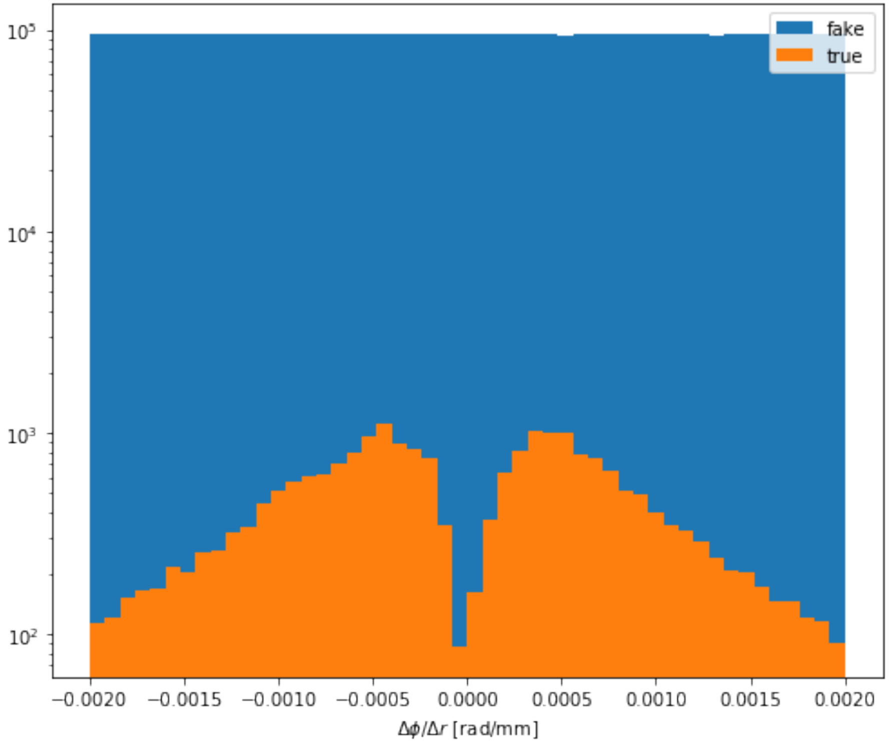
github.com/cnktysz/heptrkx-quantum

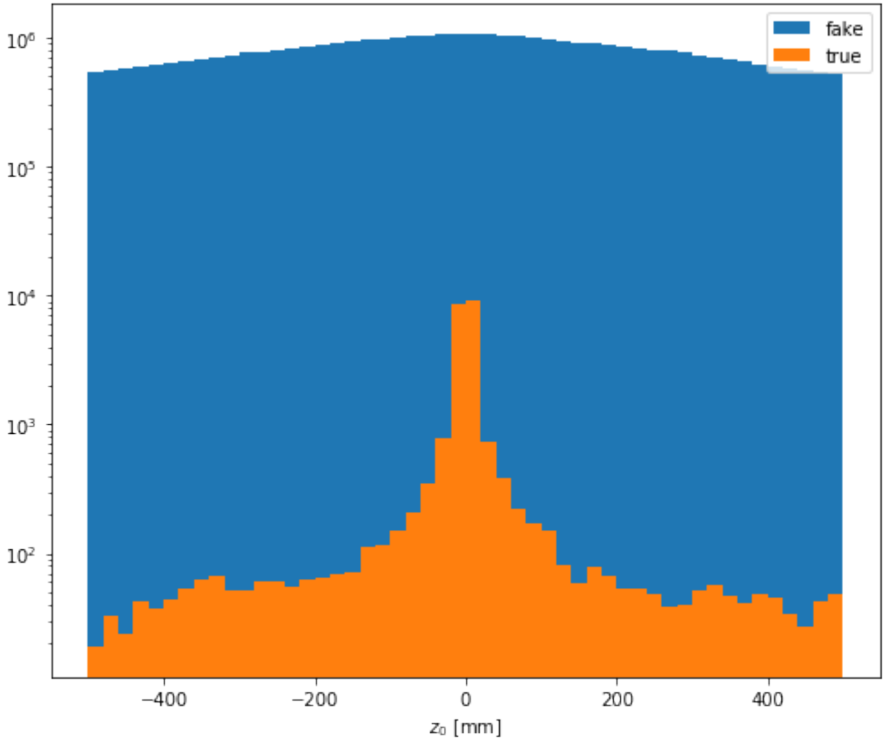
Backup Slides

Preprocessing

Following cuts are applied to the TrackML data:

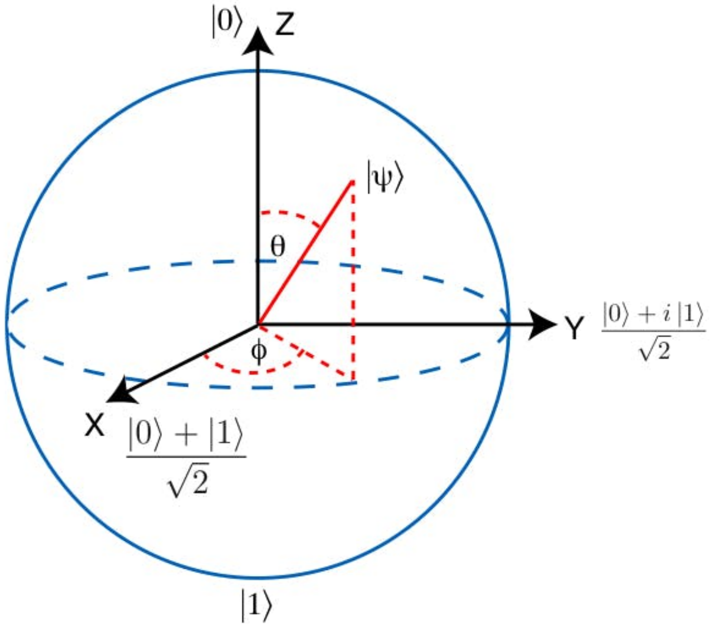
- P_t > 1.0 GeV
- $\Delta\emptyset/\Delta r < 0.006$
- z_o < 100
- -5 > $\eta > 5$





Plot from: https://github.com/HEPTrkX/heptrkx-gnn-tracking

Quantum Gates



Bloch sphere courtesy of
http://www.laborsciencenetwork.com

Simple Gates :

single parameter, rotation on a plane

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \rightarrow \text{Apply } R_y(\theta) \rightarrow \begin{bmatrix}\cos(\frac{\theta}{2})\\ \sin(\frac{\theta}{2})\end{bmatrix}$$

General Gates:

multiple parameters, rotation on the whole bloch sphere

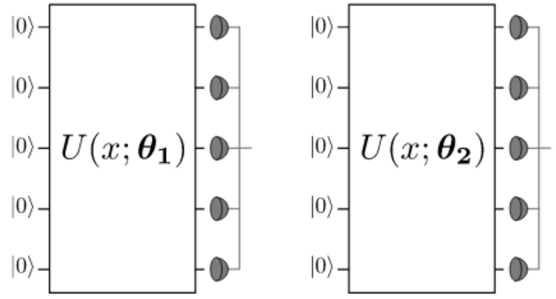
$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \rightarrow \text{Apply } U_3(\theta, \emptyset, \lambda) \rightarrow \begin{bmatrix}\cos(\frac{\theta}{2})\\ e^{i\emptyset}\sin(\frac{\theta}{2})\end{bmatrix}$$

Taking Gradients of a Q. Circuit

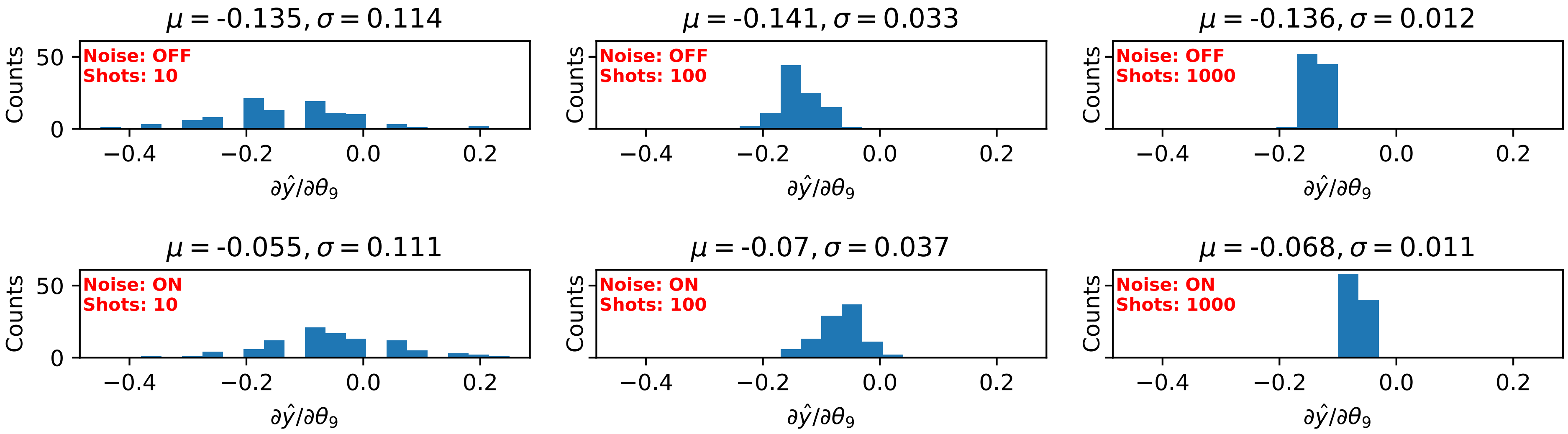
Gradient taking operation can be composed as 2 Quantum Circuits.

Pennylane is a software that supports automatic differentiation of quantum circuits.

$$\nabla_\theta f(x;\boldsymbol{\theta}) = f(x;\boldsymbol{\theta}_1) - f(x;\boldsymbol{\theta}_2)$$



https://pennylane.ai/



Noisy simulations are promising!

AUC

True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

ROC is the curve for TPR vs FPR.
AUC is the integral of ROC.

AUC = 1.0 means perfect score.