Embedding of particle tracking data using hybrid quantum classical neural networks

25th International Conference on Computing in High-Energy and Nuclear Physics

20.05.2021

Carla Rieger (ETH Zurich)

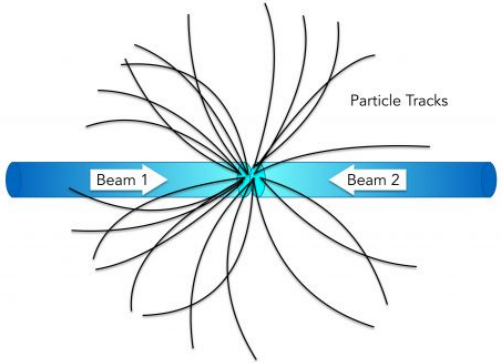
Cenk Tüysüz (METU), Kristiane Novotny (gluoNNet), Sofia Vallecorsa (CERN), Bilge Demirköz (METU), Karolos Potamianos (gluoNNet, Oxford University), Daniel Dobos (gluoNNet, Lancaster University), Jean-Roch Vlimant (Caltech), Richard Forster (gluoNNet)

Outline

- The particle track reconstruction problem and data set
- The embedding within the data processing pipeline
- Quantum circuit model architectures and training results
- Comments on future improvements

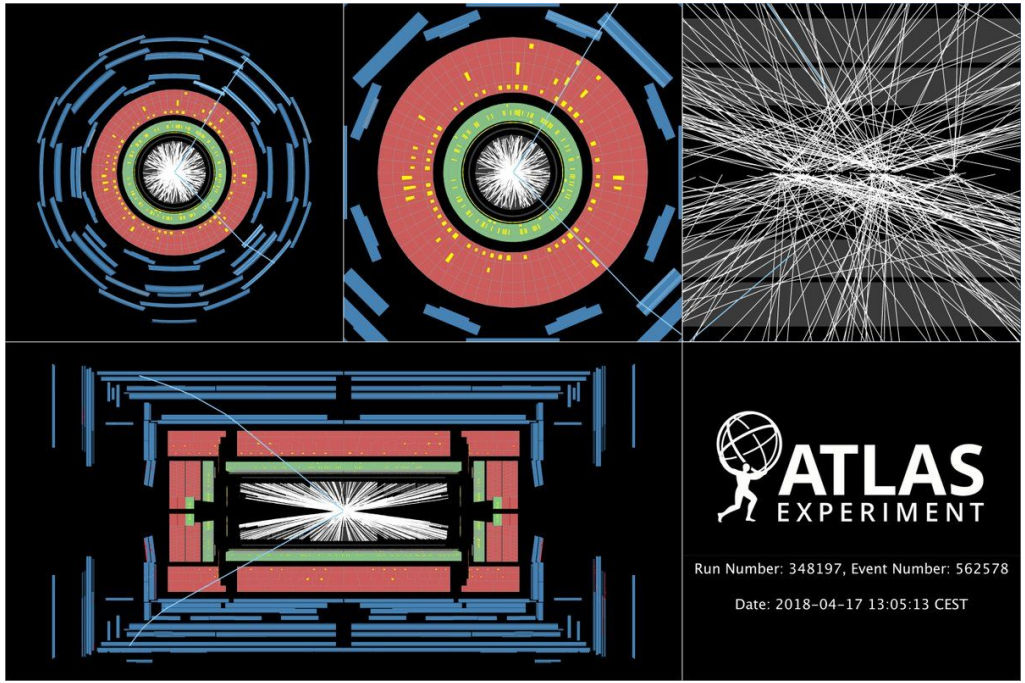
Large Hadron Collider (LHC)

and particle track reconstruction



https://atlas.cern/updates/atlas-news/counting-collisions

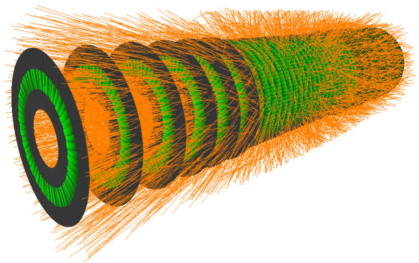
An event view from ATLAS Experiment

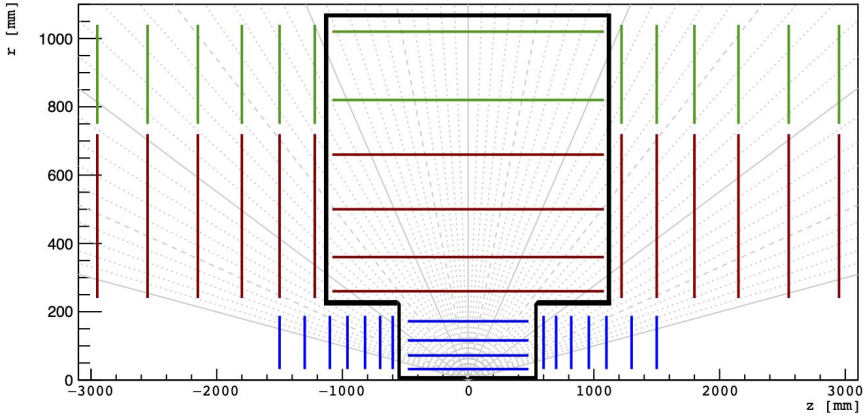


https://cds.cern.ch/record/2315786

TrackML Dataset

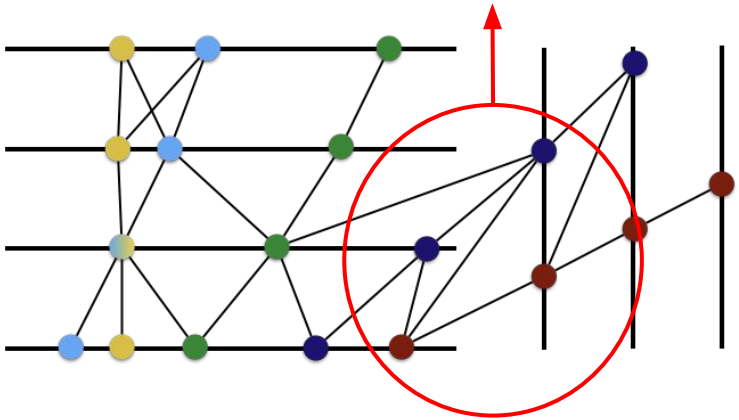
https://www.kaggle.com/c/trackml-particle-identification/overview





Contains: 10k collision events (200 soft QCD interactions)
(arXiv: 1904.06778)

endcaps produce a lot of ambiguity and therefore many track candidates, we omit endcaps as we want to limit our model to simpler cases.

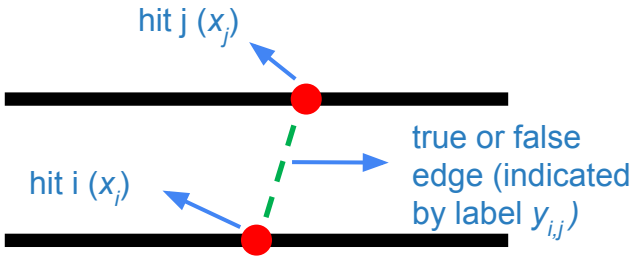


Retrieved from: Farrell et al. 2018 (arXiv: 1810.06111)

Learning the embedding of the hit data set

The data processing pipeline

First preprocessing steps of the data set → Embedding of pairs of hits (doublets): $\Phi(x_i, \theta)$ in 4D embedding space → Graph construction and classification with (quantum) GNN



A doublet in original space.

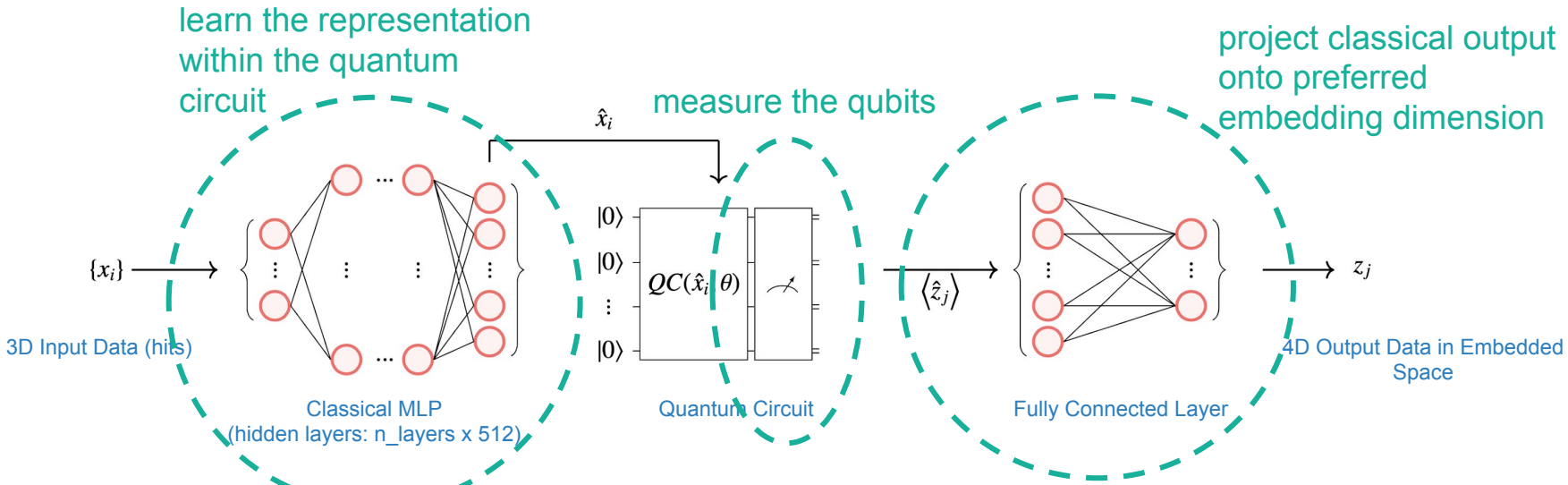
Hinge Embeddig Loss:

$$s_n = (x_i, x_j, y_{i,j})$$

$$loss(s_n) = \begin{cases} \max\{0, \|\Phi(x_i,\theta) - \Phi(x_j,\theta)\|_2\}, & \text{if } (x_i, x_j) \text{ belong to the same trajectory} \\ 0, & \text{if } (x_i, x_j) \text{ belong to different trajectories.} \end{cases}$$

Similar to Choma et al. 2020 (arXiv: 2007.00149)

Hybrid Neural Network architecture



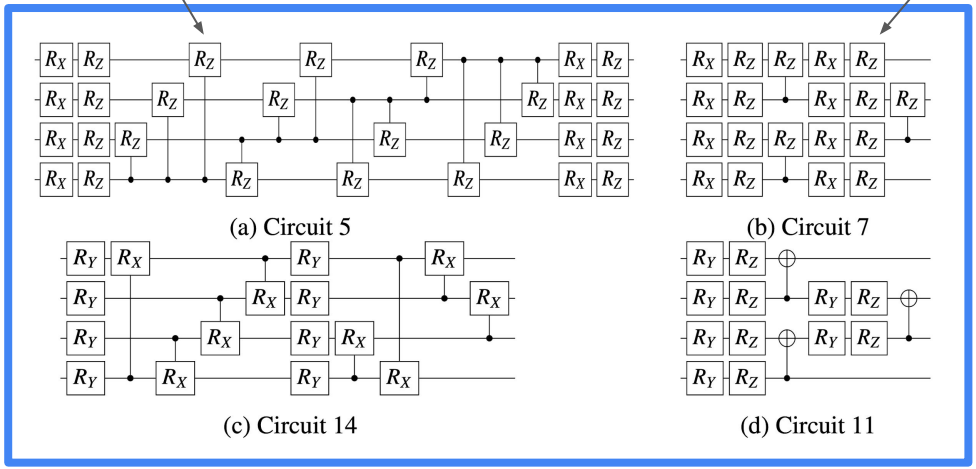
General model of the hybrid architecture

MLP adapted from: Choma et al. 2020 (arXiv: 2007.00149)

Quantum Circuit Approach

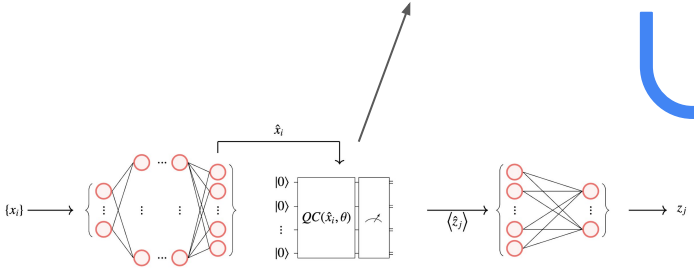
$$CR_Z(\hat{x}_i) = \mathbb{1} \otimes |0\rangle\langle 0| + \exp\left(-i\frac{\hat{x}_i}{2}\sigma_Z\right) \otimes |1\rangle\langle 1|$$

$$R_Z(\hat{x}_i) = \exp\left(-i\frac{\hat{x}_i}{2}\sigma_Z\right)$$



- these quantum circuits were chosen due to their different values regarding entanglement and expressibility
- the 4-qubit quantum circuits encode the output of the classical MLP as angles within the rotational gates displayed

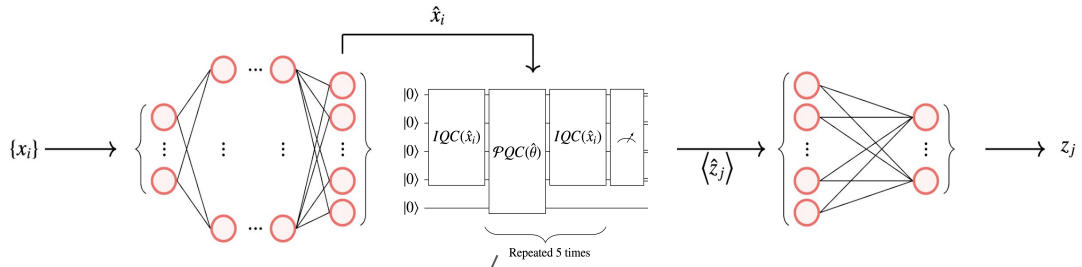
$$QC_{id} : \mathbb{R}^{n_{parameters}} \longrightarrow \mathbb{R}^{n_{measurements}}$$



Circuit	Parameters $n_{parameters}$	Entanglement (the higher the better)	Expressibility (the lower the better)
5	28	0.290	0.051
7	19	0.212	0.104
11	12	0.538	0.139
14	16	0.545	0.011

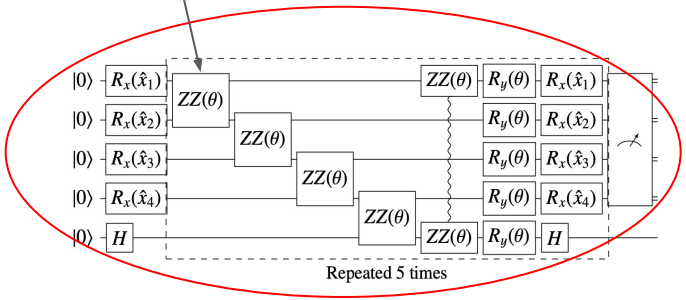
Circuits adapted from and values calculated as in: Sim et al. 2019 (arXiv:1905.10876)

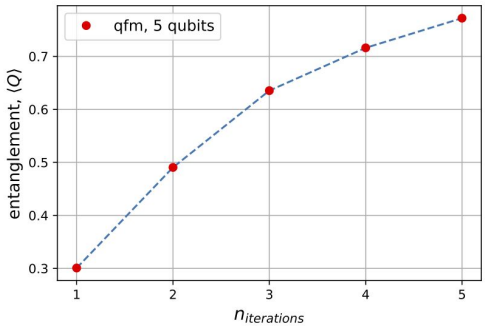
Quantum Feature Map Approach (QFM)

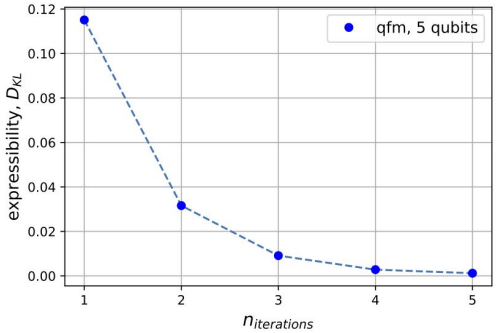


- QFM iteratively encodes the input x_i
- additionally, there are optimizable parameters θ included within the quantum circuit
- repeating blocks of the circuit as indicated on the left increases the entanglement and decreases the expressibility value, which is preferable

$$ZZ(\theta) = \exp\left(-i\frac{\theta}{2}(\sigma_Z)^{\otimes 2}\right)$$





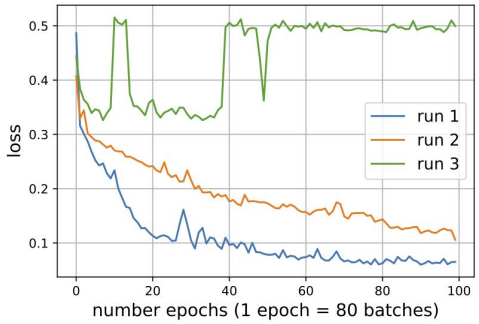


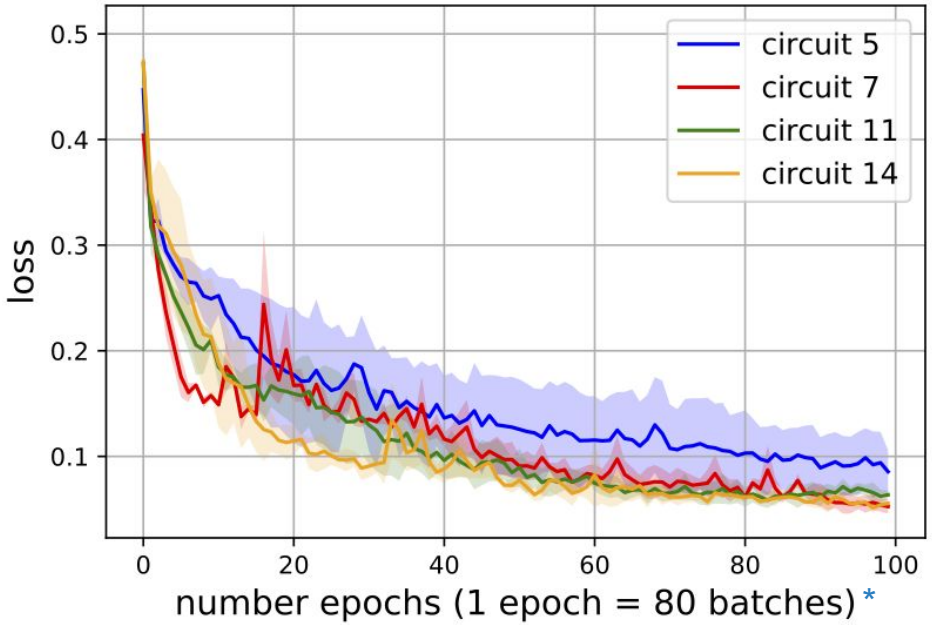
Adapted from: Lloyd et al. 2020 (arXiv:2001.03622)

Ent./Expr. values calculated as in: Sim et al. 2019 (arXiv:1905.10876)

Training results

Quantum Circuit Approach





- training time proportional to number of gates in the quantum circuit
- observation of plateaus in training/validation loss for circuit 5, which includes the highest number of QC parameters in this test

Circuit	Parameters $n_{parameters}$	Entanglement (the higher the better)	Expressibility (the lower the better)	Training time (average per batch)
5	28	0.290	0.051	$37 \pm 8s$
7	19	0.212	0.104	$20 \pm 4s$
11	12	0.538	0.139	$14 \pm 4s$
14	16	0.545	0.011	$16 \pm 4s$

Training data set: 8k hits, validation data set: 2k hits, using ADAMAX optimizer, n_layers = 10, hinge embedding loss, lr = 1e-2.

* mean and indicated std of 3 independent runs, plot without circuit 5 run 3 for better visualization

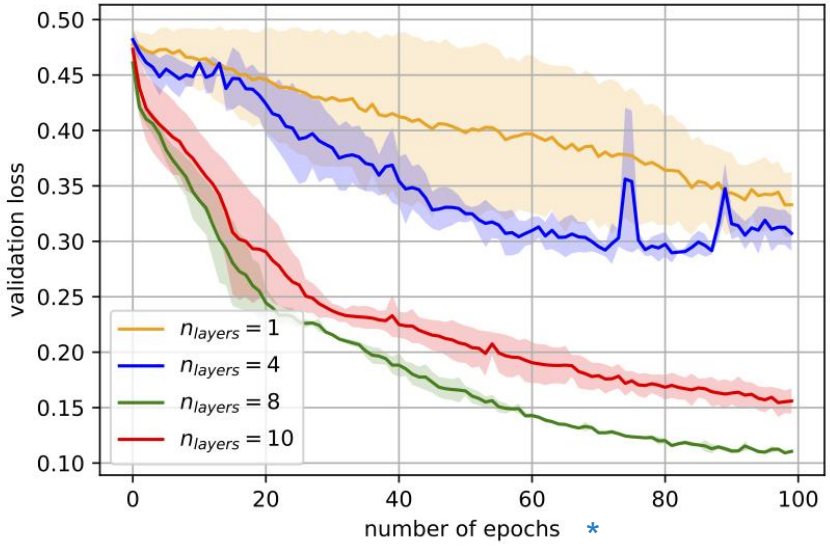
Carla Rieger

Ent./Expr. values calculated as in: Sim et al. 2019 (arXiv:1905.10876)

Training results

QFM Approach

Circuit	Parameters $n_{parameters}$	Entanglement (higher value preferred)	Expressibility (lower value preferred)	Training time (average per batch)
QFM (5 qubits) ($n_{iteration} = 5$)	74	0.772	0.001	$5min38 \pm 8s$
14 (4 qubits) ($n_{iteration} = 1$)	16	0.545	0.011	$16 \pm 4s$



- learning rate has to be lowered using this architecture by factor 0.1
- similar performance of 1 and 4 layer version, as well as for 8 and 10 classical layers
- validation loss converges to high validation loss for low number of layers
- std much higher when training with less classical layers
- possibility for better convergence when training for more than 100 epochs (especially for 8 and 10 layers)

Training data set: 8k hits, validation data set: 2k hits, using ADAMAX optimizer, hinge embedding loss, lr = 1e-3.

* mean and indicated std of 2 independent runs

Ent./Expr. values calculated as in: Sim et al. 2019 (arXiv:1905.10876)

Conclusion

How to improve?

- test different quantum encoding schemas
- explore more quantum circuits
- explore different architectures
- train on more doublet data

Challenges

- quantum models are hard to simulate and the simulation times are long, especially for larger models that include more trainable parameters

Things to explore

- explore effects when training with real hardware and noise models
- test non-hybrid quantum architectures

Contributors

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

[1]ETH Zurich, Zurich, Switzerland,
[2]Middle East Technical University, Ankara, Turkey,
[3]gluoNNet, Geneva, Switzerland,
[4]CERN, Geneva, Switzerland,
[5]Oxford University, Oxford, UK,
[6]Lancaster University, Lancaster, UK,
[7]California Institute of Technology, Pasadena, California, USA

Thank you.

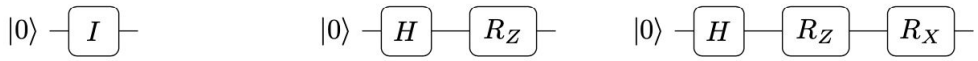
Email: carieger@ethz.ch

Results shown here will be published soon, with a complete overview.
The most recent version of the code will be published under https://qtrkx.github.io.

Backup Slides

Q.C. for Machine Learning

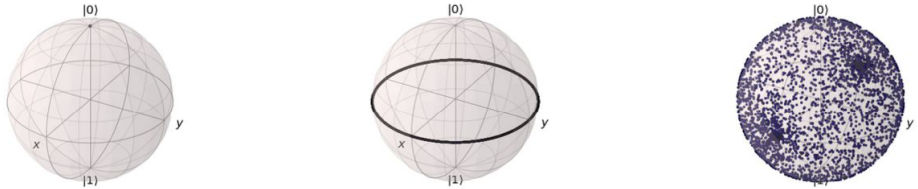
Parameterized Gates



$$RZ(\lambda) = exp(-i\frac{\lambda}{2}Z) = \begin{pmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{pmatrix}$$

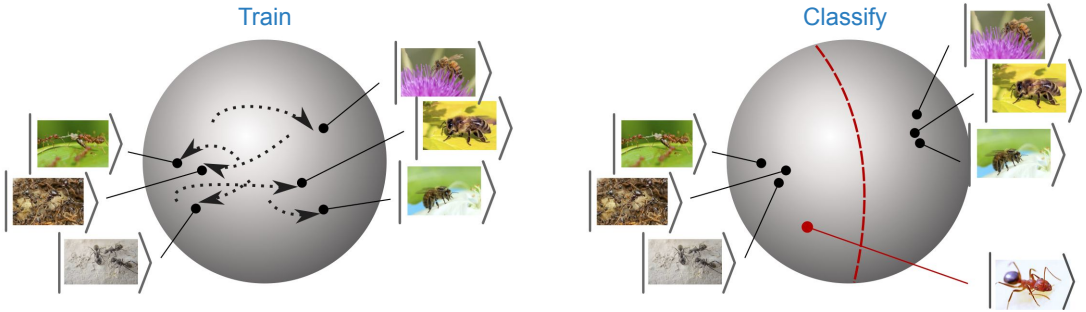
$$RX(\theta) = exp(-i\frac{\theta}{2}X) = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

We can use parameterized gates to embed data in the Hilbert Space.



Adapted from: Sim et al. 2019 (arXiv:1905.10876)

Then, we can use other parametrized gates that we can optimize to do tasks such as classification.



Adapted from: Lloyd et al. 2020 (arXiv:2001.03622)