Quantum Graph Neural Networks for HEP

Master Thesis presentation

Thibault Desrousseaux

Supervisor: Prof. Dr. G. Dissertori

Co-supervisor: Vasileios Belis

Introduction

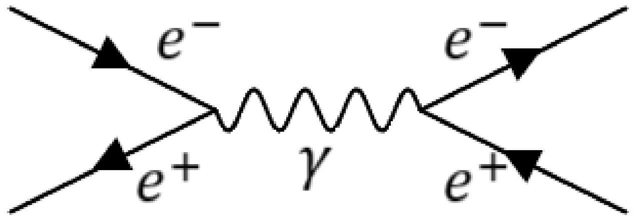
- Two projects:
  - Computation of Feynman diagrams
  - Jet classification
- Goals:
  - Explore potential applications of Quantum Graph Neural Networks (QGNN) for HEP
  - Study QGNN properties, when combined with other methods (data reuploading, equivariance…)

Computation of Feynman diagrams

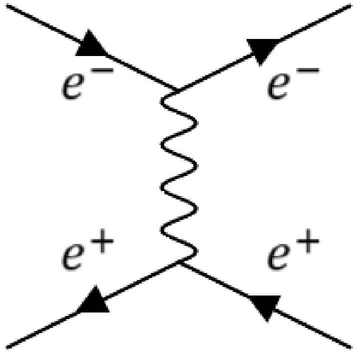
Aim of the project

- Compute scattering amplitude $|\mathcal{M}|^2(G, p, \theta)$
  - Using a classical/quantum neural network

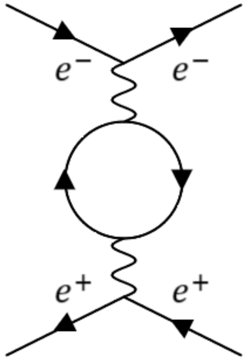
- Two goals:
  - Compute several diagrams with the same model
  - Combine diagrams to compute the total amplitude



S-channel



T-channel



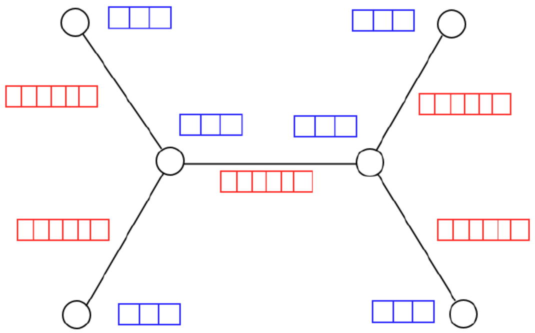
Loop channel

Input: a diagram, p, θ

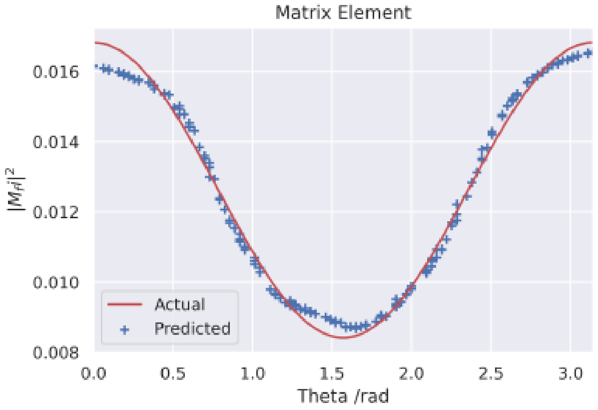
Previous Work

- Classical approach: Mitchell & al.
  - Consider Feynman diagrams as graphs:
    - Edges: particles
    - Nodes: flow of time
  - Use classical graph ML methods (GAT+FNN) to compute the scattering amplitude
  - Limit:
    - They don’t try to compute different amplitudes with the same model
    - They don’t combine amplitudes when they want add diagrams

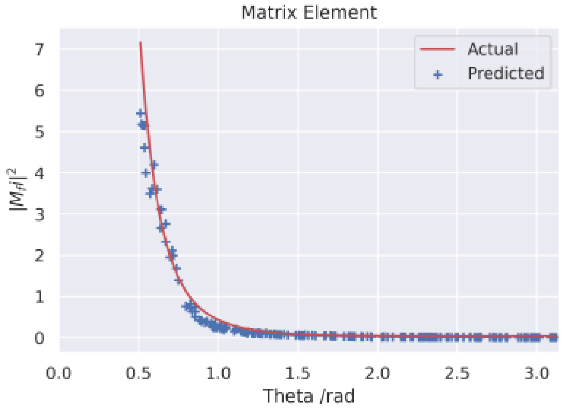
Graph encoding



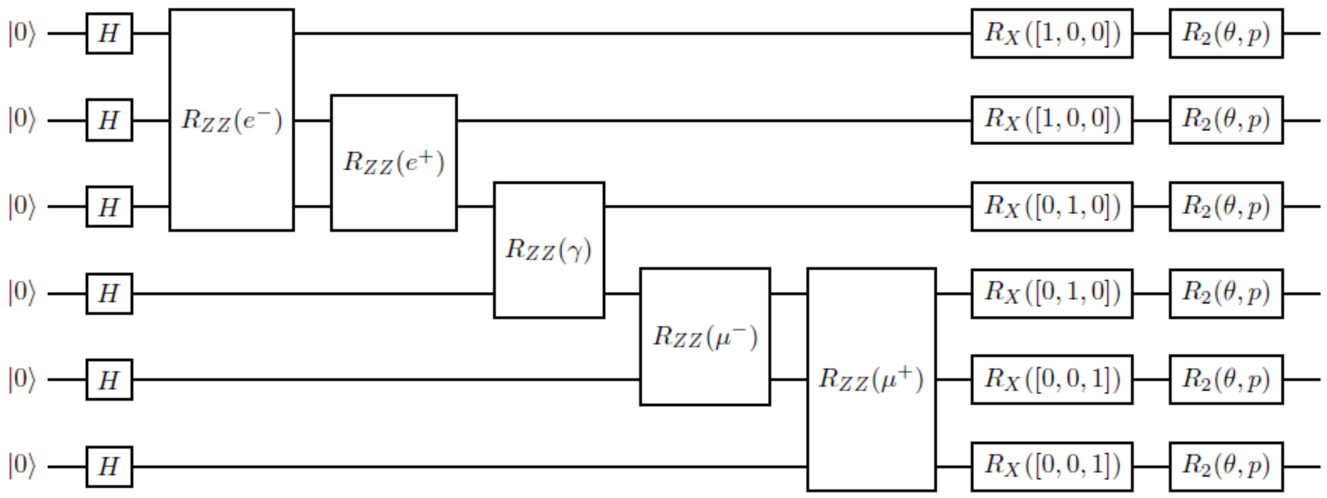
S-channel



T-channel



Previous Work



Feature map

- Quantum approach: L. Ballerio
    - Encode the graph structure in a quantum circuit (feature map)
    - Trainable ansatz and observable
    - Idea: amplitudes add with interferences, like quantum systems

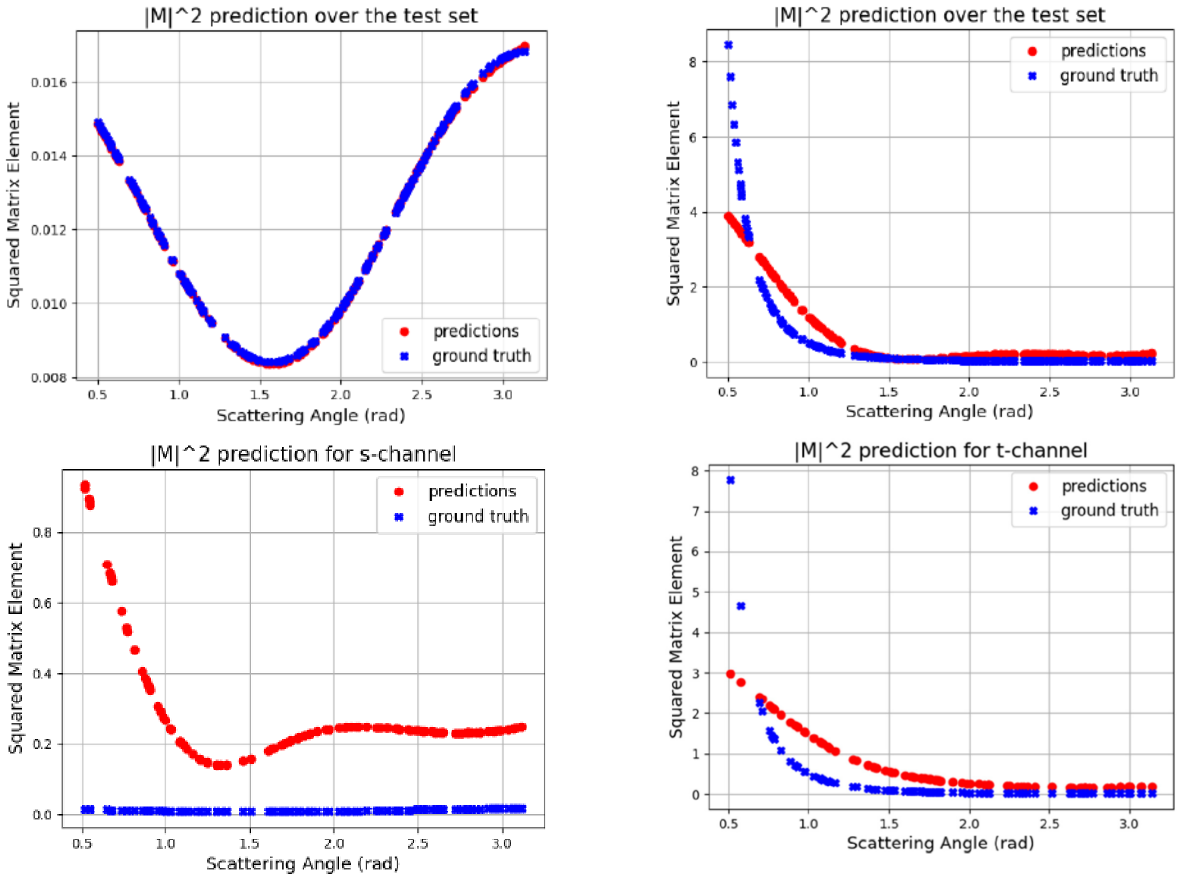
Previous Work

S-channel

T-channel

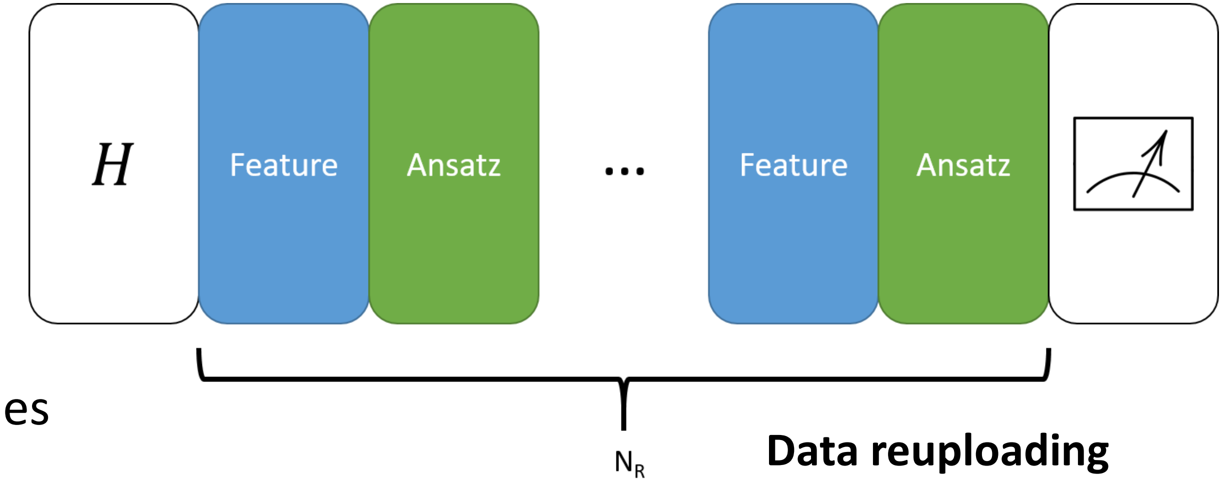
Separate training

Combined training

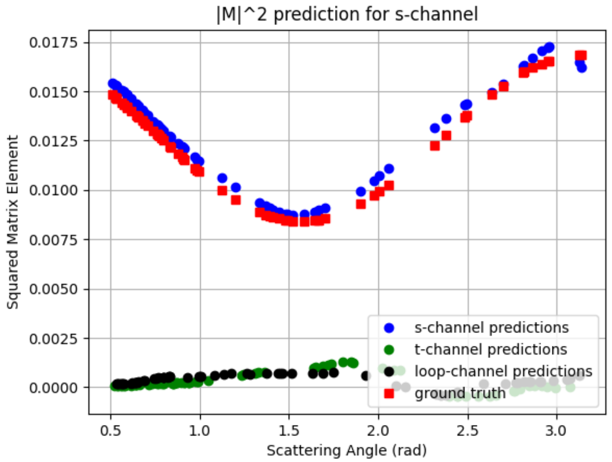


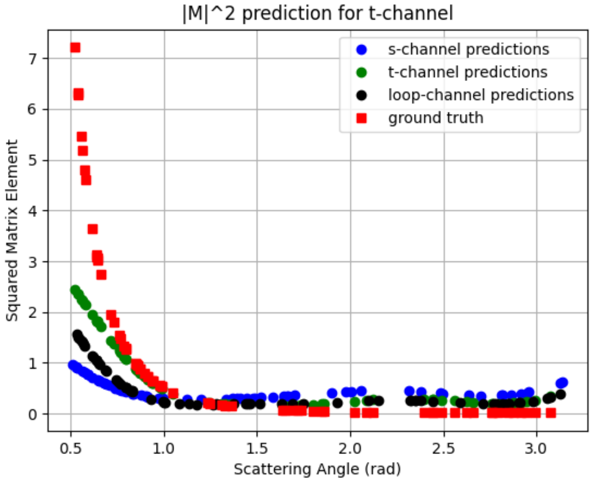
As for the classical model, combined training do not work

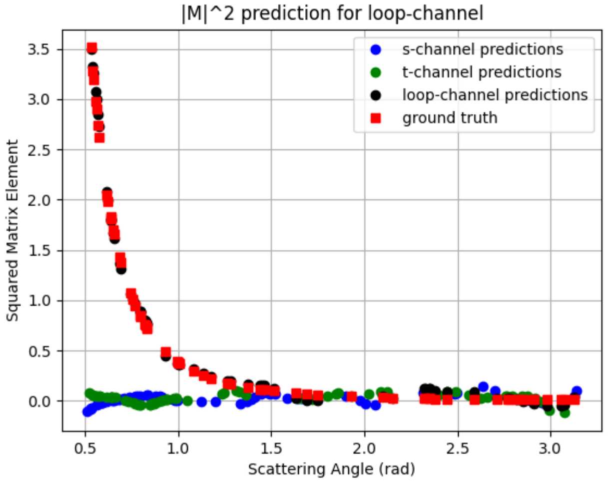
Algorithm and results



- New ideas:
  - Use data reuploading:
    - Equivalent in simple cases to adding frequencies to the Fourier decomposition of the target function
  - Use of one trainable observable per channel

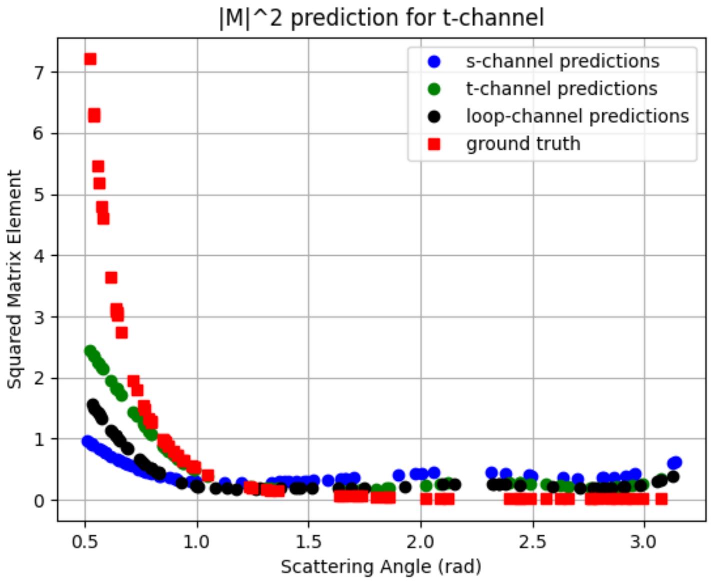




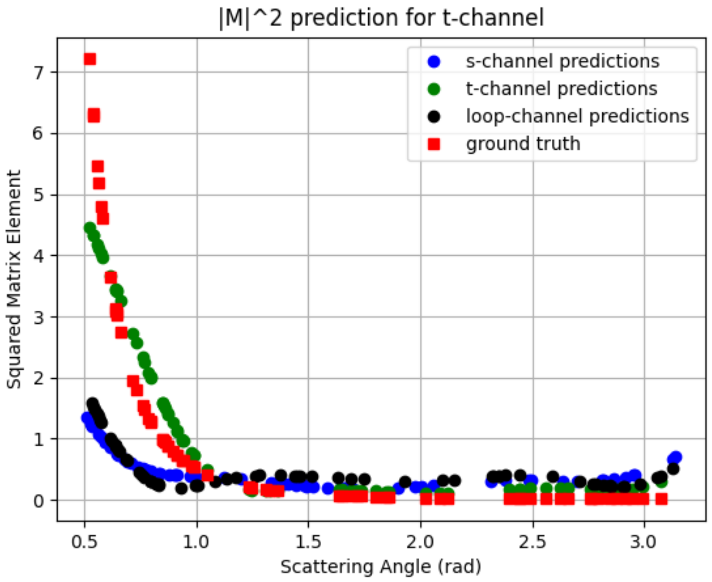


Studies around the algorithm

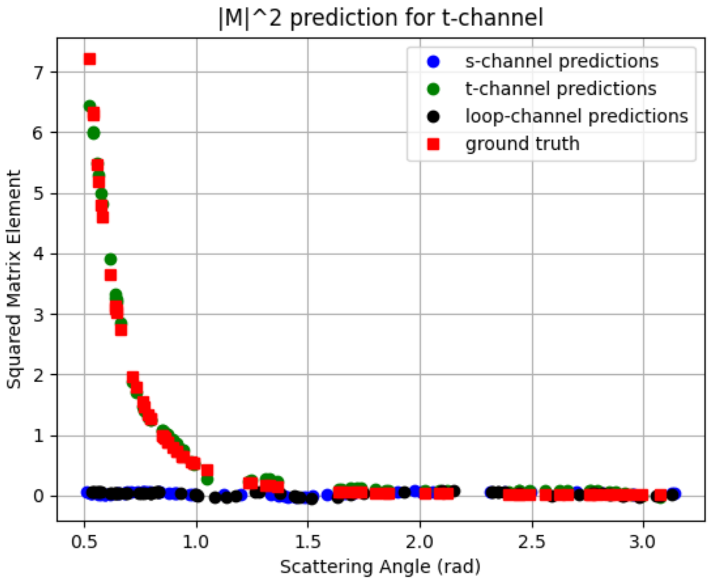
- How does the model improve with the number of reuploads?



3 reuploads



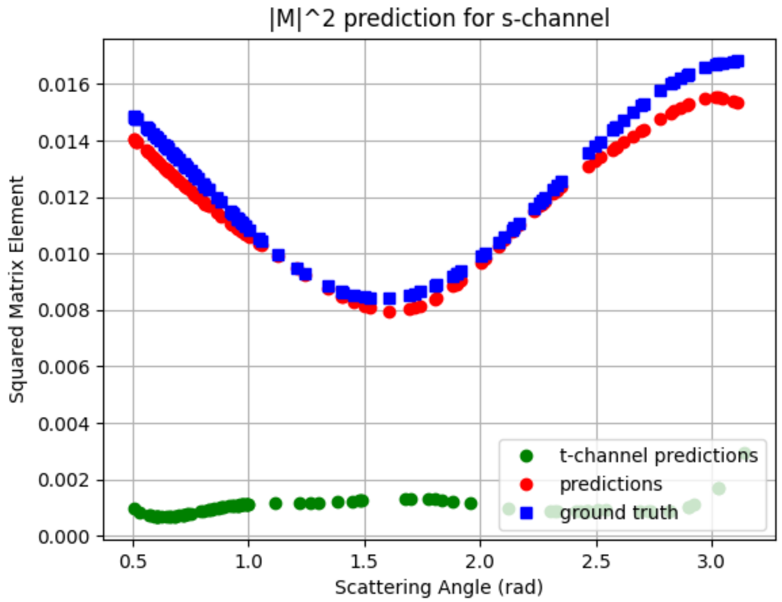
5 reuploads

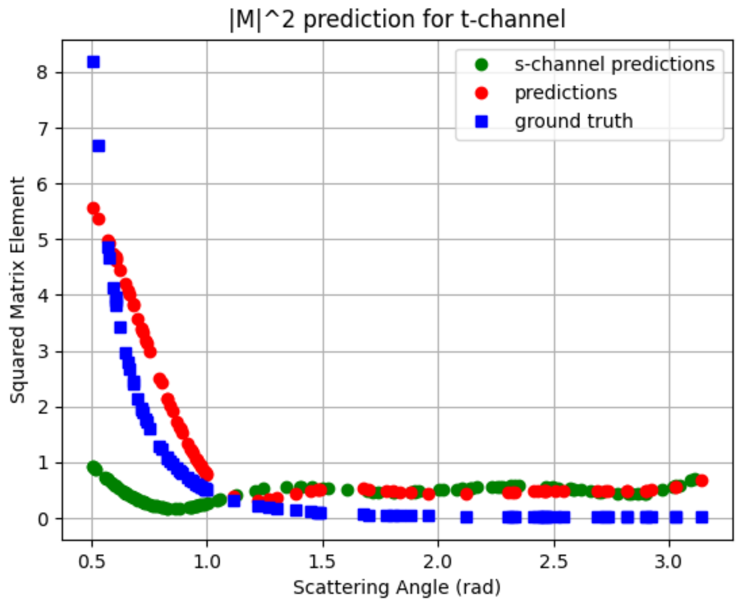


7 reuploads

Studies around the algorithm

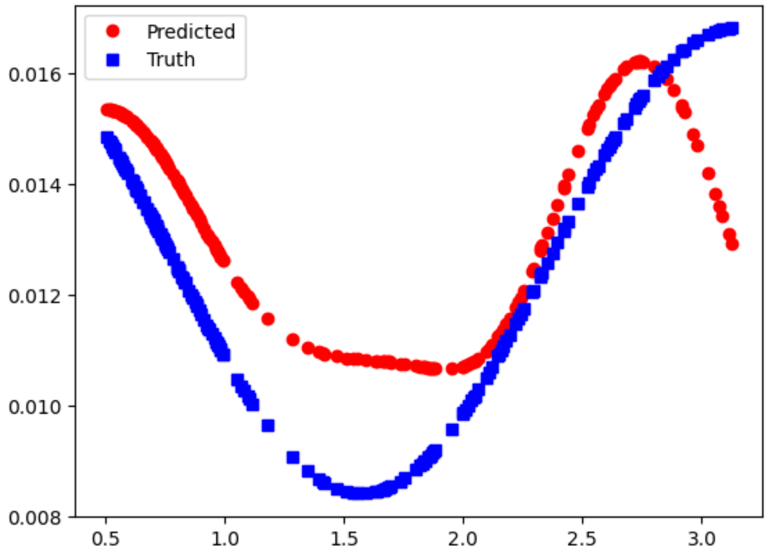
- Which information in the encoding is *really* important?
  - Keep only the topology of the graph
  - Training works



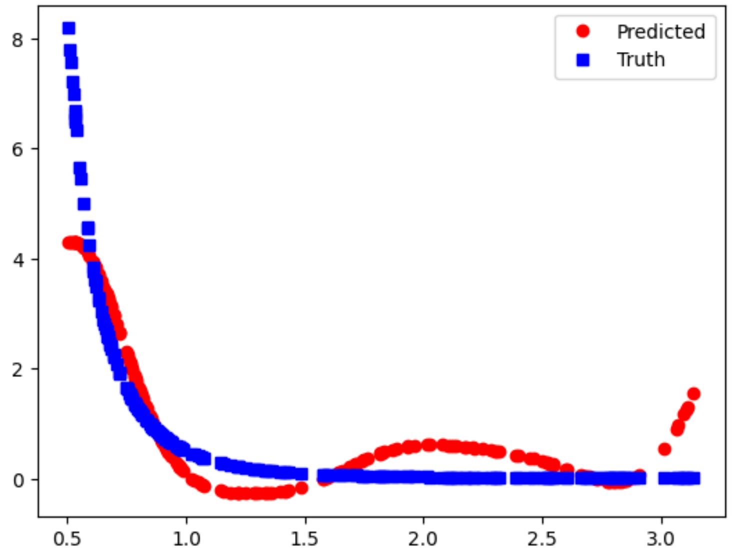


Studies around the algorithm

- Is this encoding just a fancy switch?
  - Change the encoding into an angle encoding
  - The model do not train correctly



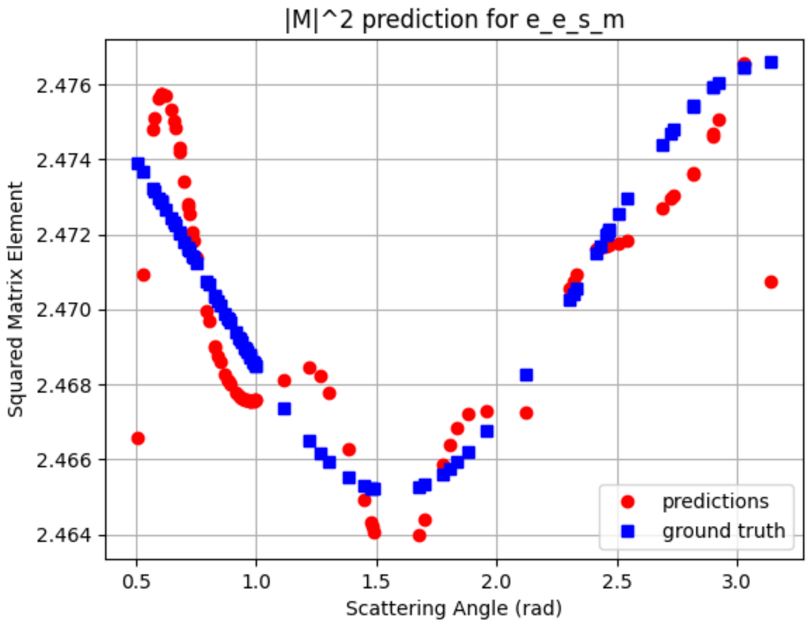
S-channel

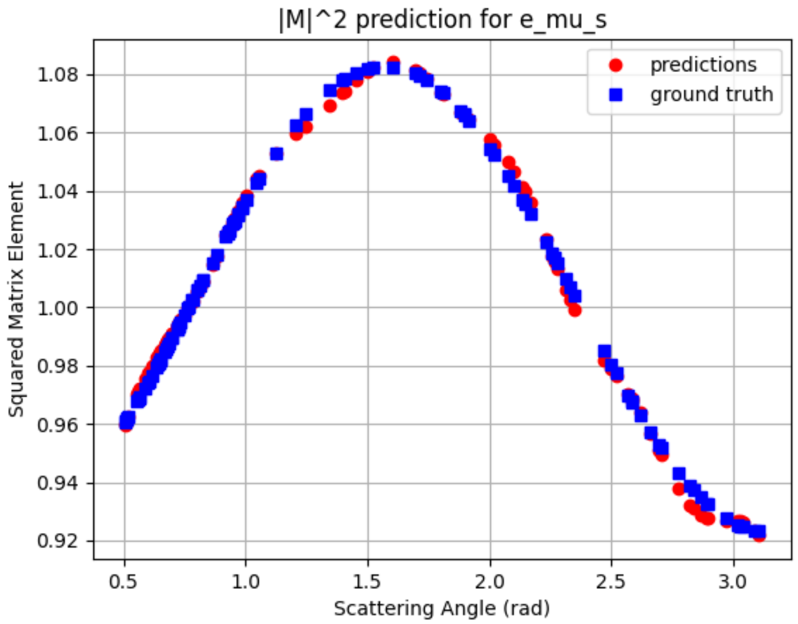


T-channel

Studies around the algorithm

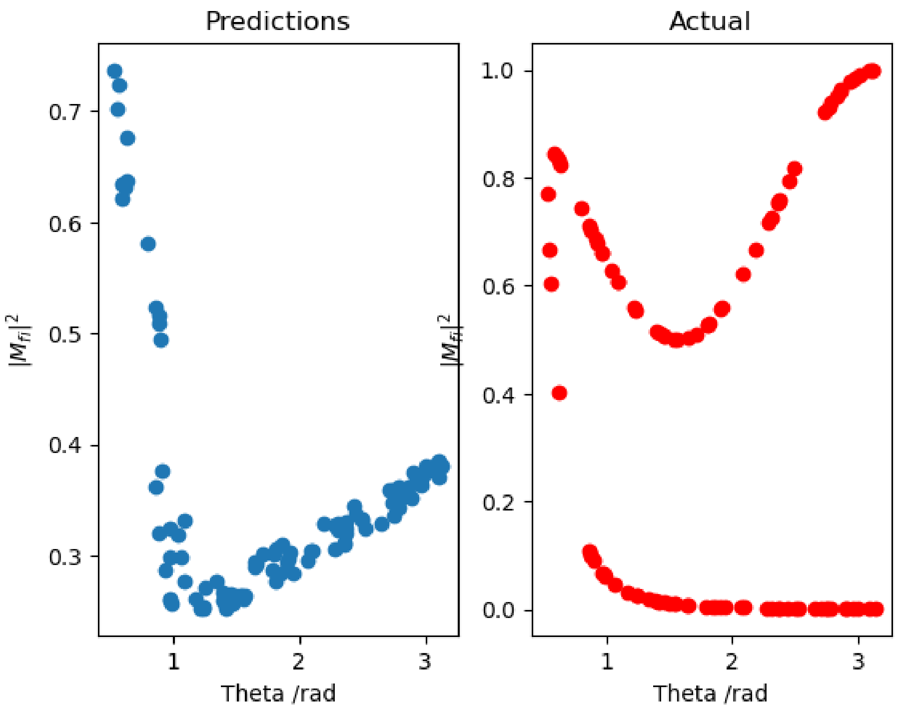
- Can the encoding work if the topology is the same?
  - Use $e^+e^- \to e^+e^-$ and $e^+e^- \to \mu^+\mu^-$:
  - The model trains



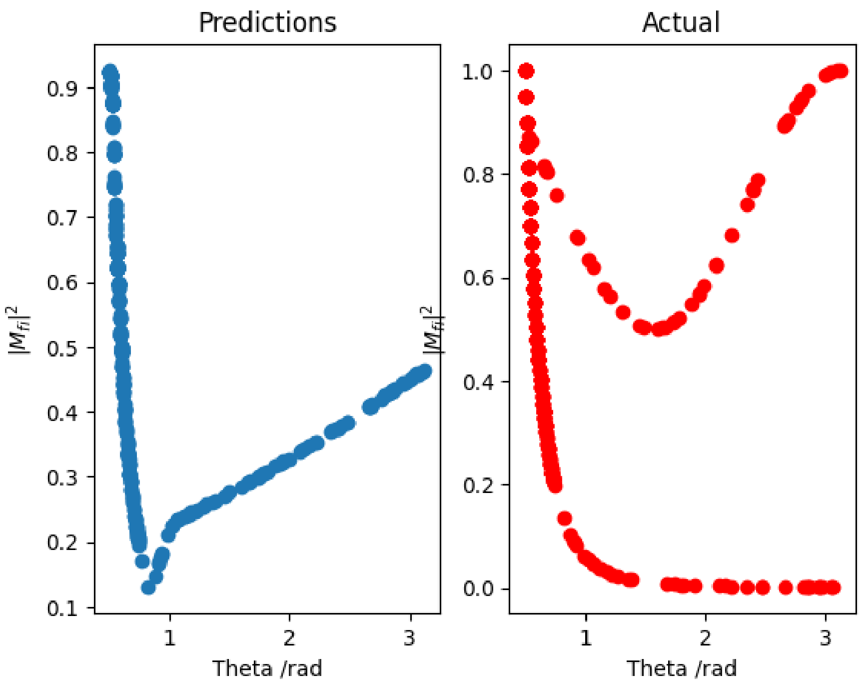


Studies around the algorithm

- Is it possible to reproduce the results with the classical NN?



With one observable



With two observables

Computing the total amplitude?

- Original hope: the algorithm would produce a state corresponding to the complexe amplitude: $|\mathcal{M}_s\rangle$ or $|\mathcal{M}_t\rangle$
- Squared amplitude: $|\mathcal{M}_s|^2 = \langle\mathcal{M}_s|O|\mathcal{M}_s\rangle$
- Total amplitude (using LCU):

$$|\mathcal{M}_s + \mathcal{M}_t|^2 = \frac{1}{2}\left(\langle\mathcal{M}_s| + \langle\mathcal{M}_t|\right) O \left(|\mathcal{M}_s\rangle + |\mathcal{M}_t\rangle\right)$$

- However:
  - Complexe amplitudes actually depend on the spins
  - There is a extra degree of freedom (relative phase)

Conclusion

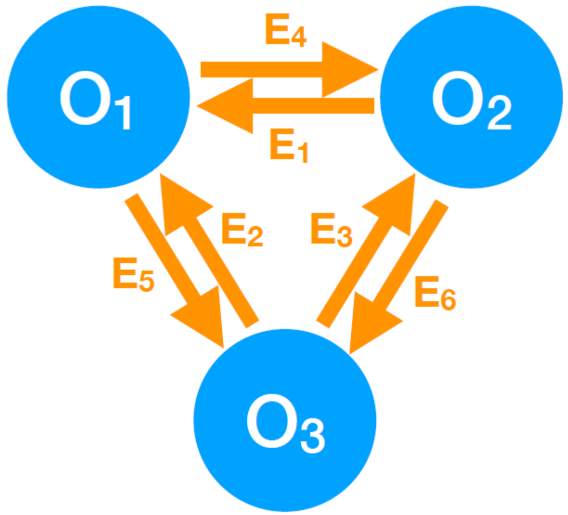
- New QGNN able to perform both classification and regression at the same time
- Graph encoding combined with data reuploading is a promising method
- Showed that computing the total amplitude won’t work

- Could we apply this algorithm to other problems?

Jet Classification with a hybrid equivariant neural network

Overview

- Task: classification of five types of jets: q,t,W,Z,g
- 6 constituents per jet, for each constituent: p_T, η, ϕ
- Inspired of an article by Moreno & al.
  - Graph structure :
    - Fully connected graph
    - Node: constituent
    - Edge: concatenation of the parameters of the two constituents



Equivariance

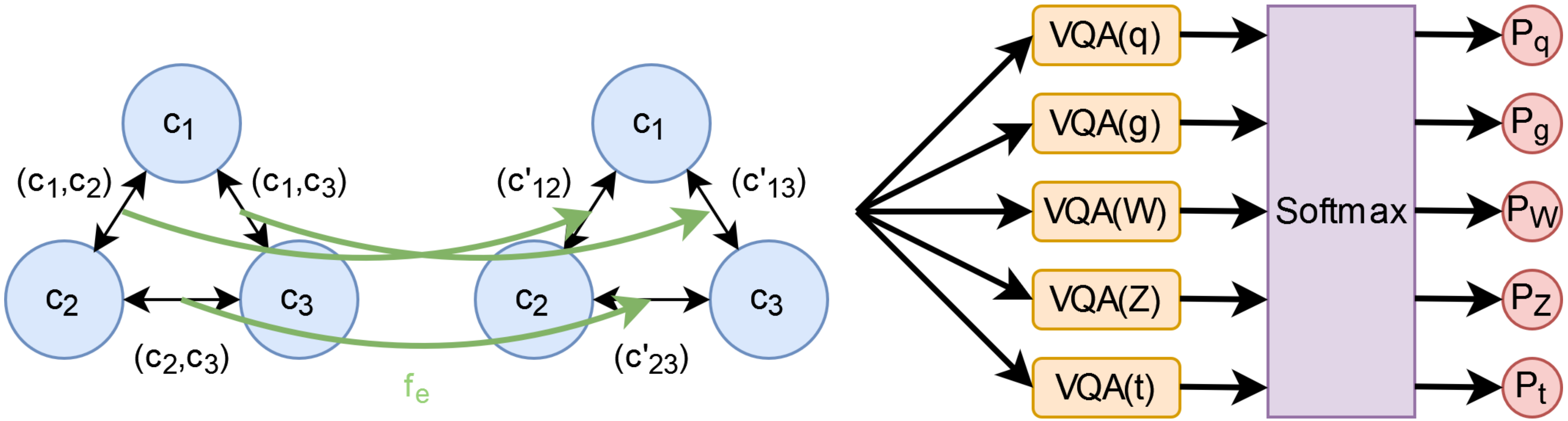
- For a group G, a function f is G-equivariant with respect to its representations ρ and ρ' if:

$$\forall g \in G, f \circ \rho(g) = \rho'(g) \circ f$$

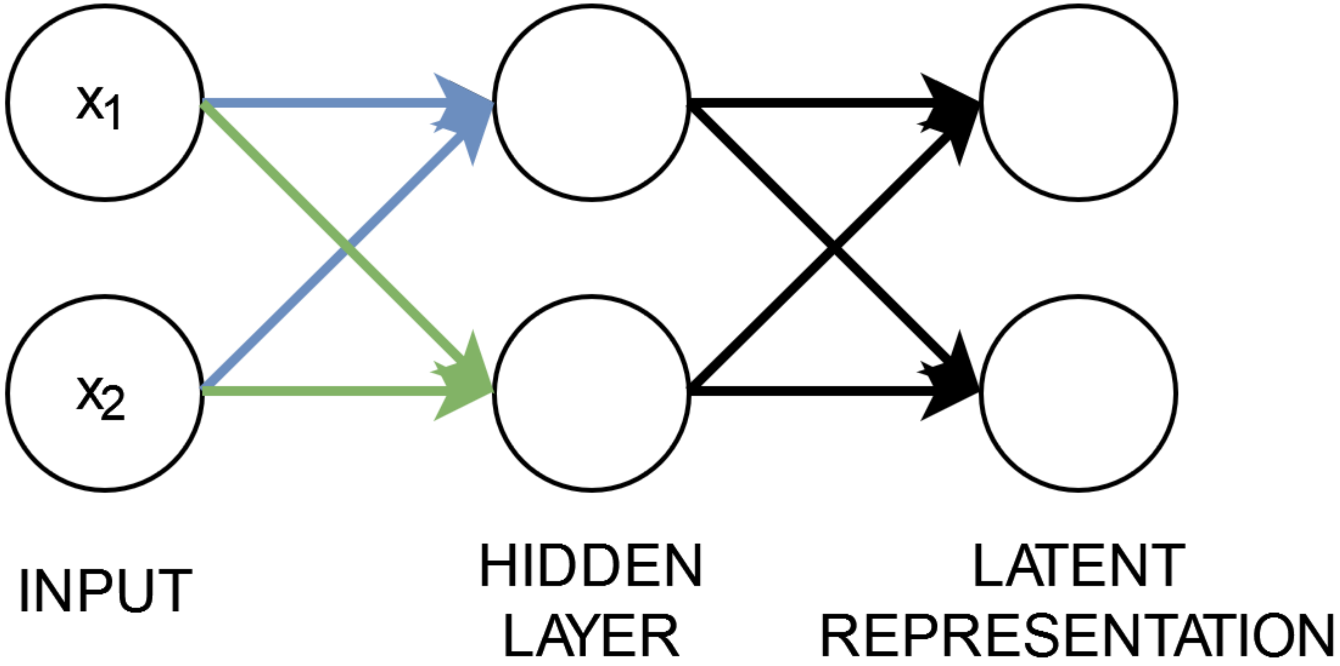
- Less restrictive than invariance
- Recipe for the permutation group for qubits:
  - 1-qubit gate: apply it on all qubits
  - 2-qubit gate: apply it on all pairs of qubits

The Hybrid Algorithm

- FNN applied to the edges
  - Transform the edge information for the quantum circuit
  - One hidden layer
  - The same FNN for all the edges
- The resulting graph is encoded in 5 different quantum classifiers
  - With data reuploading, and invariant observables
- The whole algorithm is permutation-equivariant

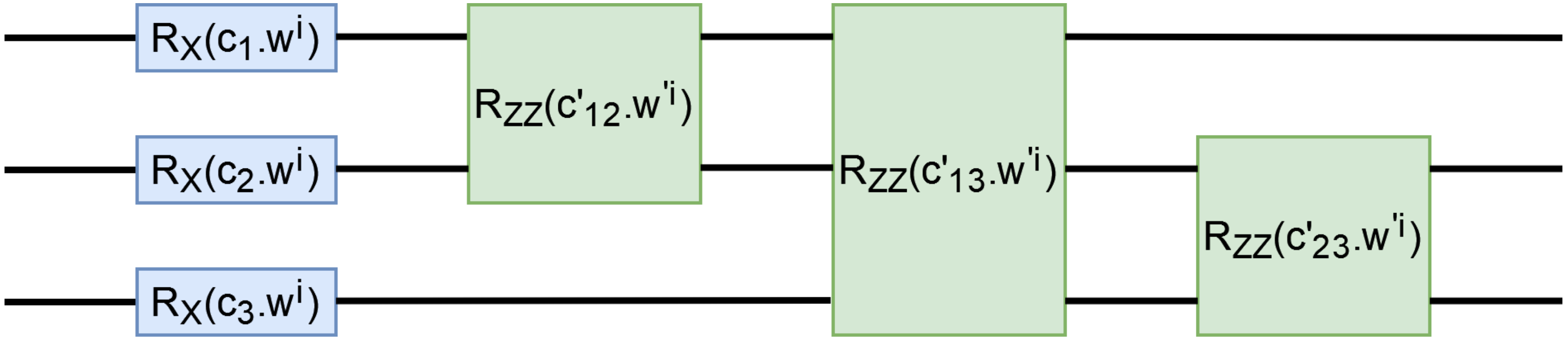


The Classical layers

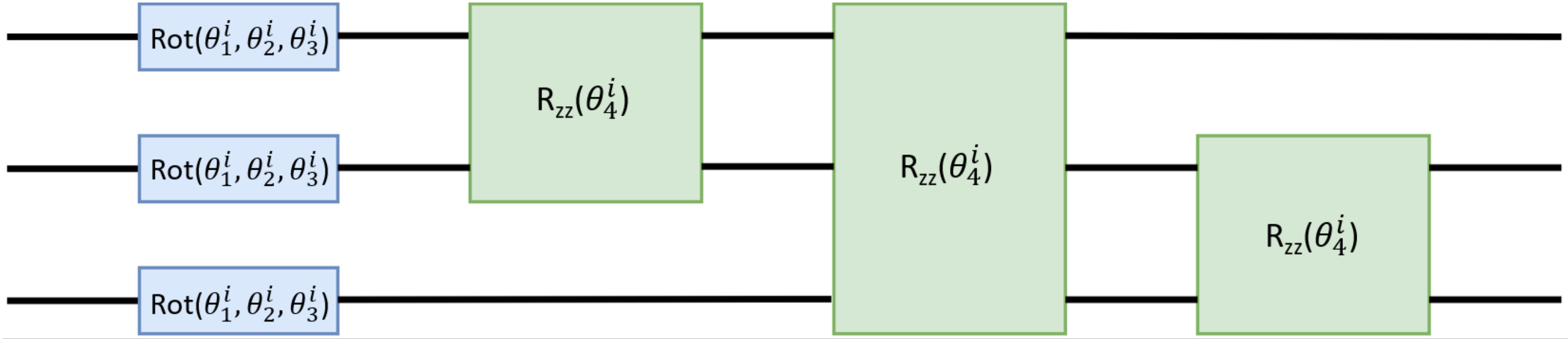


- Dimension:
  - Input: 6
  - Hidden layer: 12
  - Latent representation: 6

The Quantum Classifier



Feature map

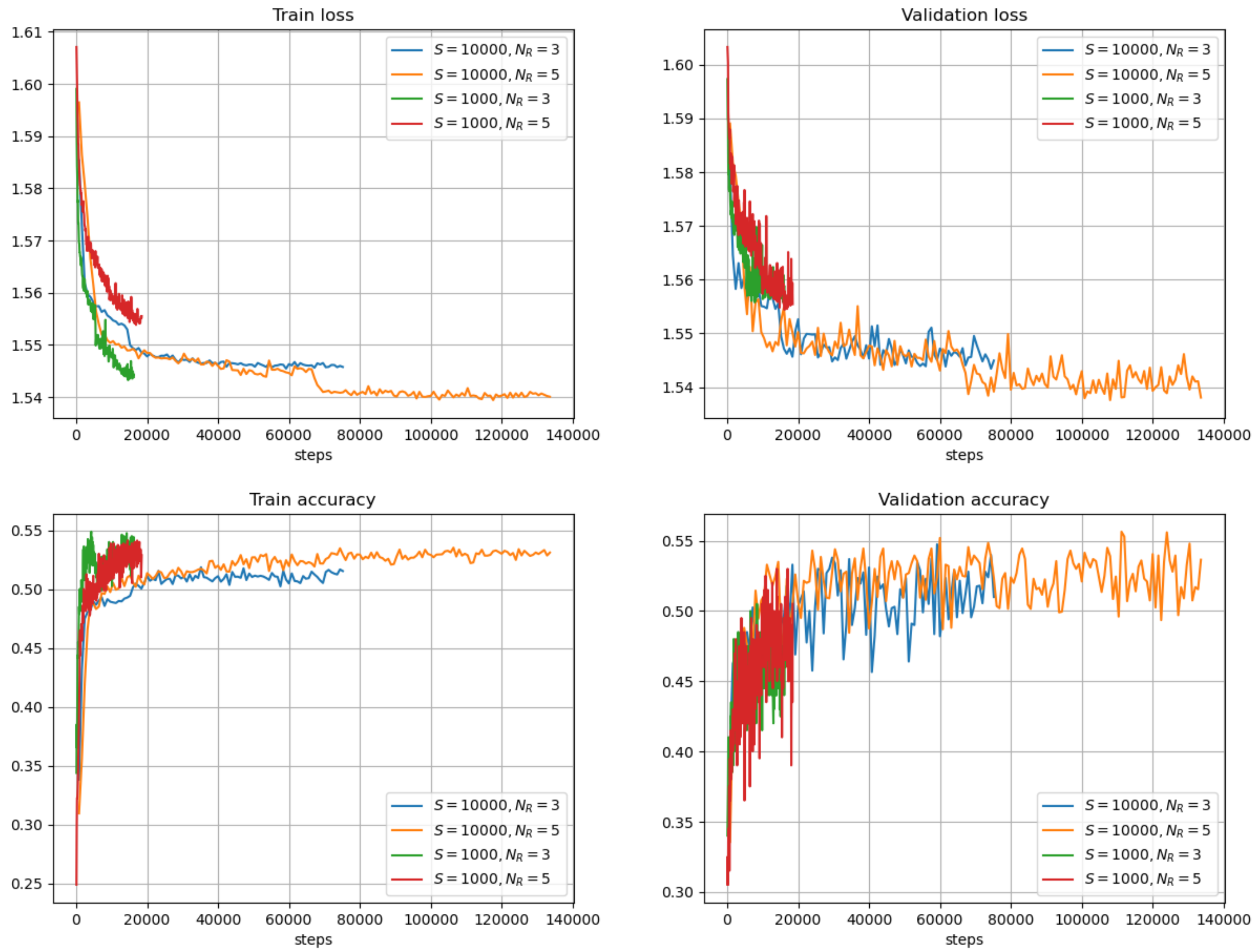


Ansatz

Results

- Size of the dataset:
  - S=1000/10000
- Number of reuploads:
  - N_R=3/5
- Equivalent MLP: 47% accuracy

Model	Loss	Accuracy	Training time
$S = 10000, N_R = 3$	1.5454	0.5037	2.724 days
$S = 10000, N_R = 5$	1.5397	0.5269	8.151 days
$S = 1000, N_R = 3$	1.5547	0.4747	14.84 h
$S = 1000, N_R = 5$	1.5590	0.4929	1.152 day



$$acc(x, y) = \mathbb{1}(\mathrm{argmax}(x) = \mathrm{argmax}(y))$$

Conclusion

- Two projects that illustrate the potential of reuploading for QGNN
  - In the case of Feynman diagrams: much better resultsthan the other existing methods
  - Jet classification: need some more work to compare its performances to other methods

Backup