GNN

GAMMA-RAY TRACKING WITH GRAPH NEURAL NETWORKS

Presented by M. Andersson and T. Bäck

GNN for track reconstruction

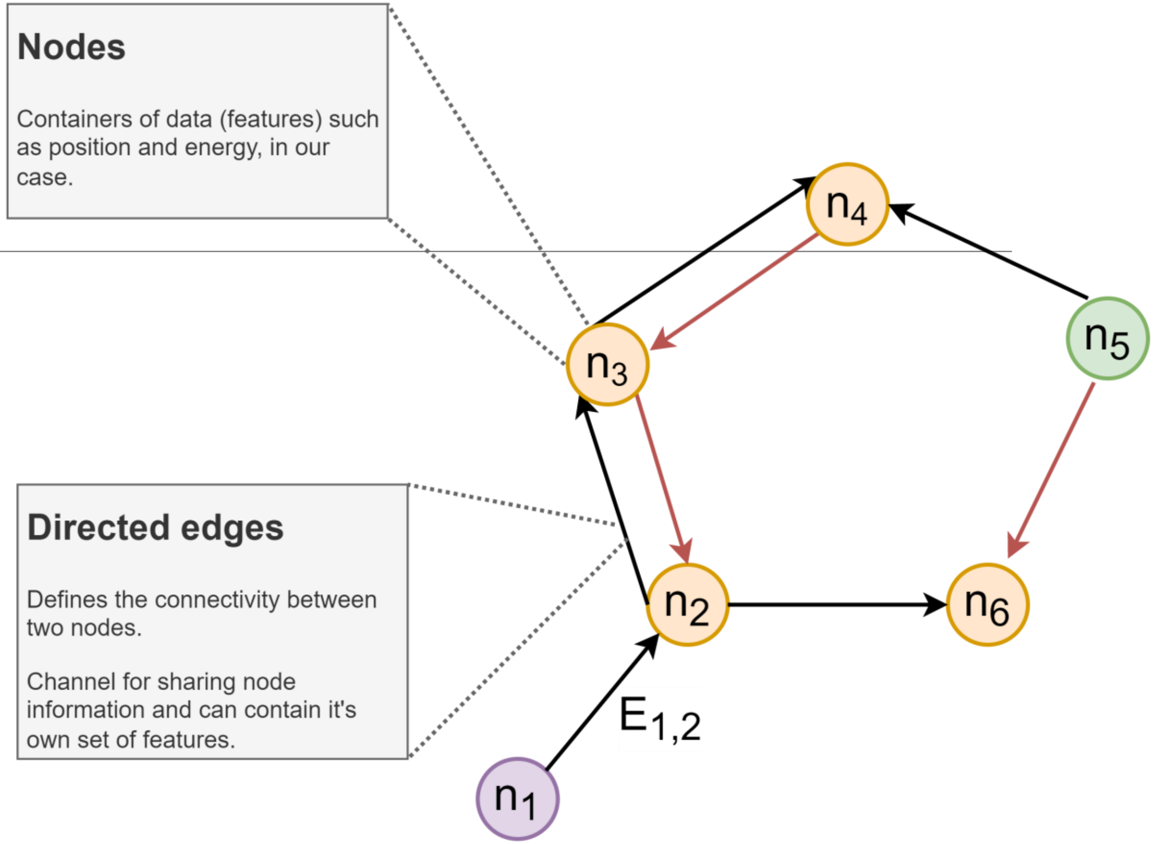
- Data structure.
- Model design.
- Ideal case (unpacked data).
- Noise and packing procedure (packed data).
- Dealing with outscattering.

Graphs:

Instead of expressing our data as points in a 3D grid or other discretized data format, use a graph, defined by nodes and their connective edges.

Both of these properties can serve to store information to be utilized by the network.

In our case: let the nodes store position and energy deposits and let the edges hold training labels.



Graph Neural Networks

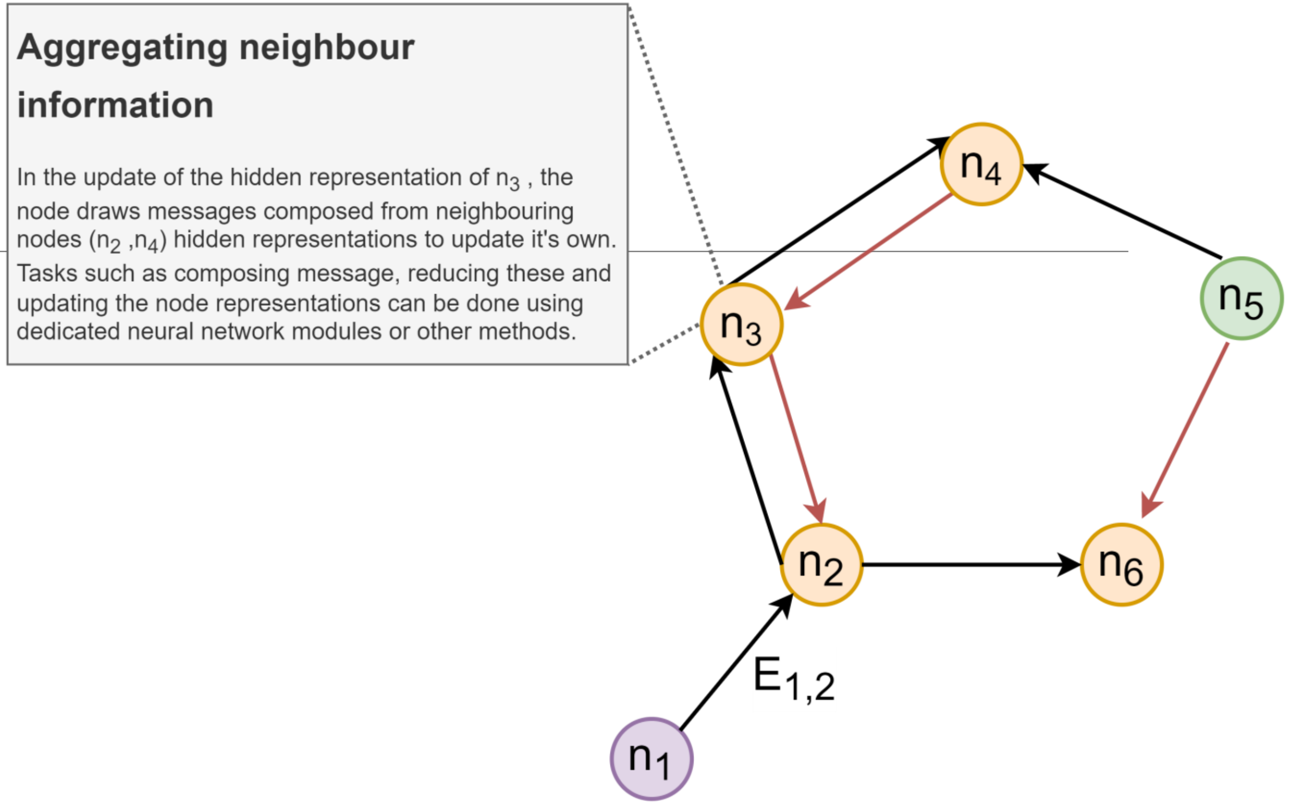
Use a graph data structure in tandem with traditional neural network models to process the input.

Propagate node information between neighbors.

Each round of message passing and aggregation facilitates greater range of information share.

Use hidden representations of the nodes to store its state.

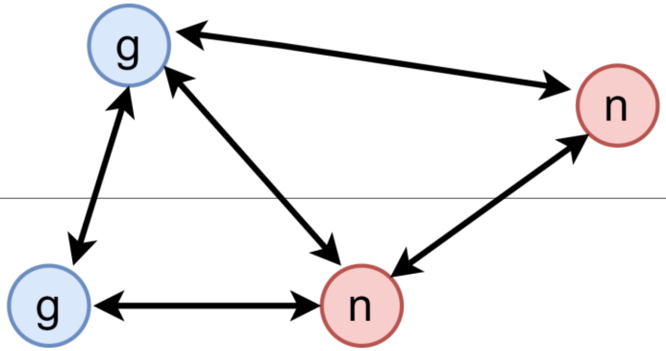
If desired different types of nodes or edges can be handled separately.



GNN classification tasks

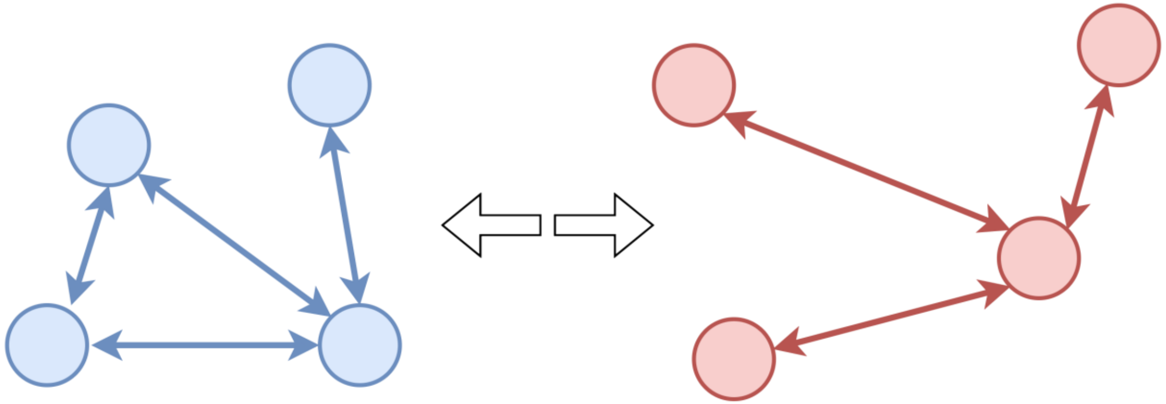
Node classification

For example, is this node the result of a photon or neutron?.



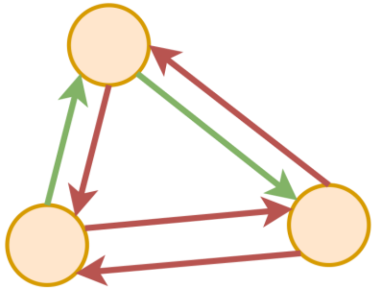
Graph classification

For example, does the event contain outscattering?.



Edge classification

For example, does the edge connect two nodes in a particle track?.



GNN in physics

Models capable of predicting the evolution of interacting systems over time.

High energy physics already investigating the applications of GNN models for tracking, such as the HEP.TrkX or the follow-up Exa.TrkX projects at high data scales.

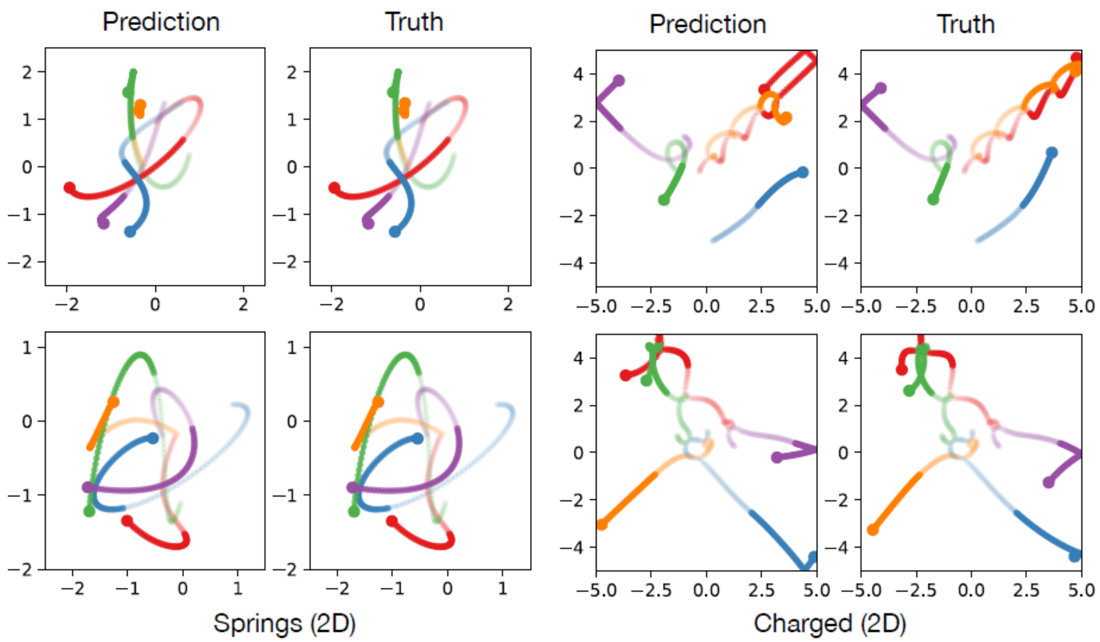
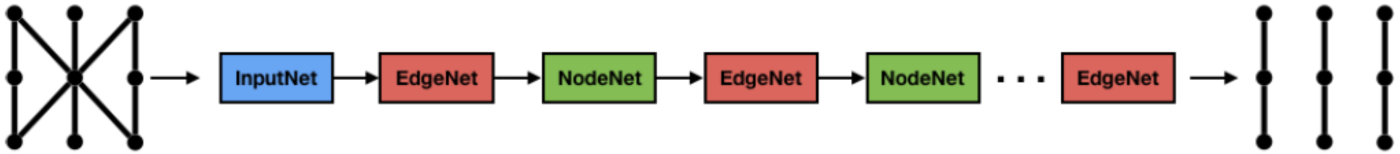


Figure 1: The predicted trajectory for objects under different forces. T. Kipf et al. *Neural relational interface for interacting systems.*

GNN in physics



Models capable of predicting the evolution of interacting systems over time.

High energy physics already investigating the applications of GNN models for tracking, such as the HEP.TrkX or the follow-up Exa.TrkX projects at high data scales.

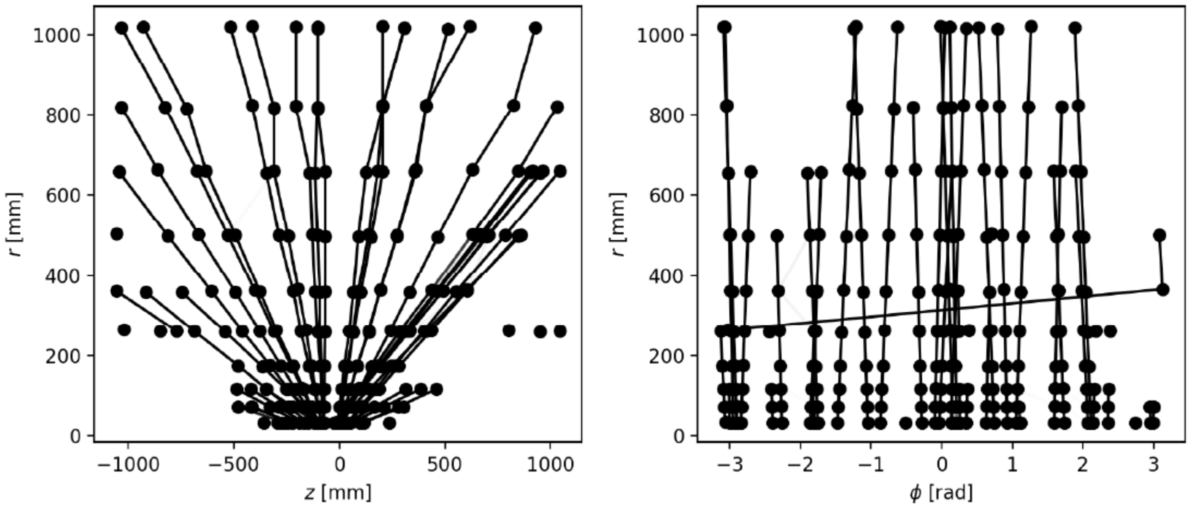
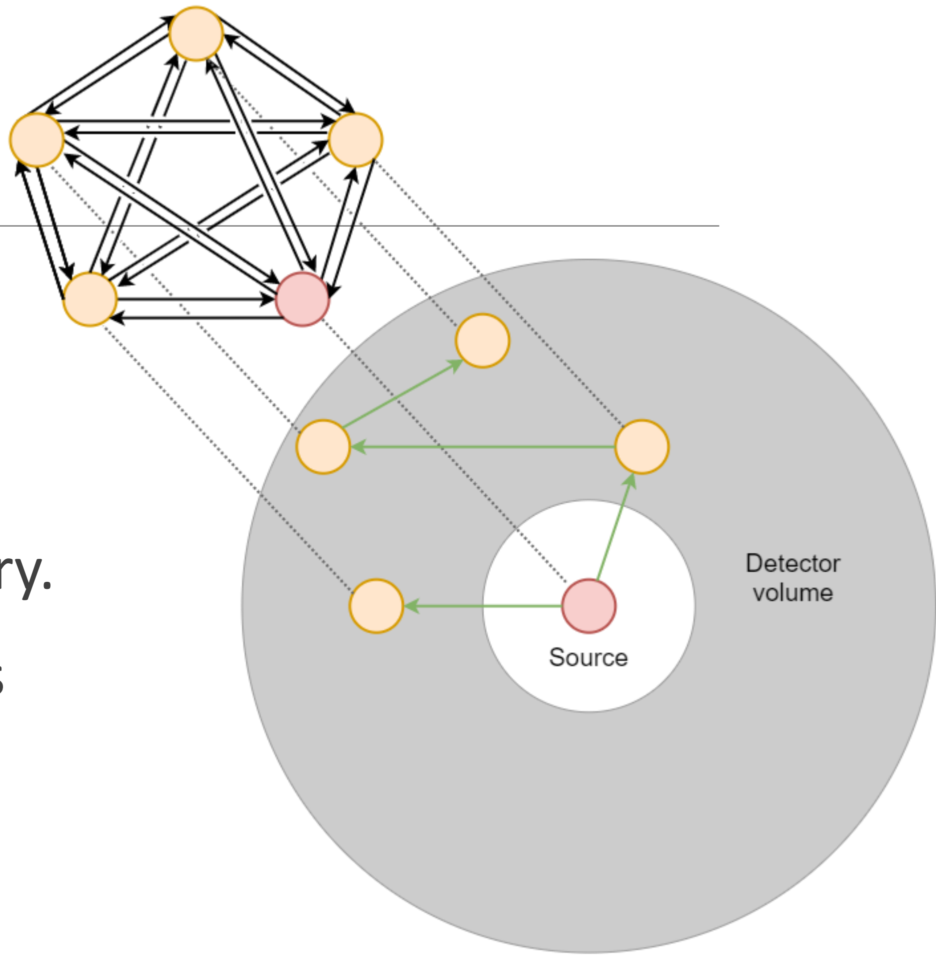


Figure 2: Model for track reconstruction and it's results on a segmented detector for particle tracking. S. Farrell et al. *Novel deep learning methods for track reconstruction.*

S. Farrell et al. Novel deep learning methods for track reconstruction. arXiv:1810.06111, 2018.

Initial considerations



We decided to use complete graph structure to start with.

Simple, thick geometry.

No knowledge of the physical processes or our geometry.

Very simple reconstruction algorithm once network has finished computing edge weights.

Implementation

- Simulation of data in Geant4.
- PyTorch.
- Deep Graph Library.
- CUDA, Training on Nvidia GTX 1060 6GB.

Network architecture

1. Initialize hidden representation vectors (*h*) for nodes from node energy and position.
2. Update edge weight (*w*) from hidden representations.
3. Propagate network information using said edge weights to generate new hidden representations.
4. Repeat step 2-3 (with individual instances of modules).
5. After final edge weights have been generated, use these to classify edges.

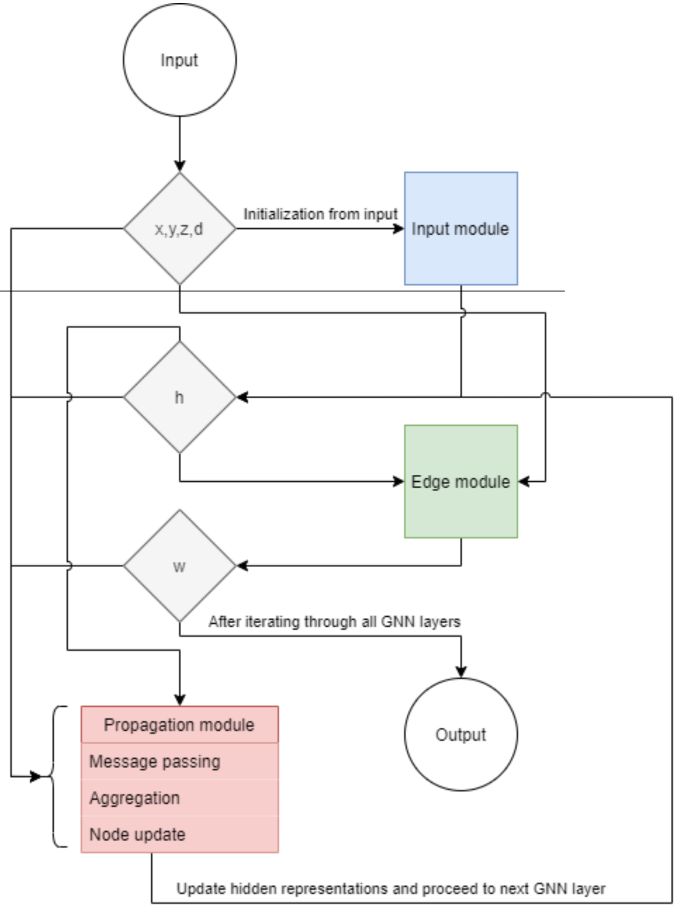


Figure 1: Schematic for information flow of the network.

Network architecture

- Simple linear layer for input initialization.
- MLP with single hidden layer for updating edge weights.
- Aggregating messages through single layer and summation, serving as input for GRU cells using the current hidden representation as memory vector. The result is the updated *h*.

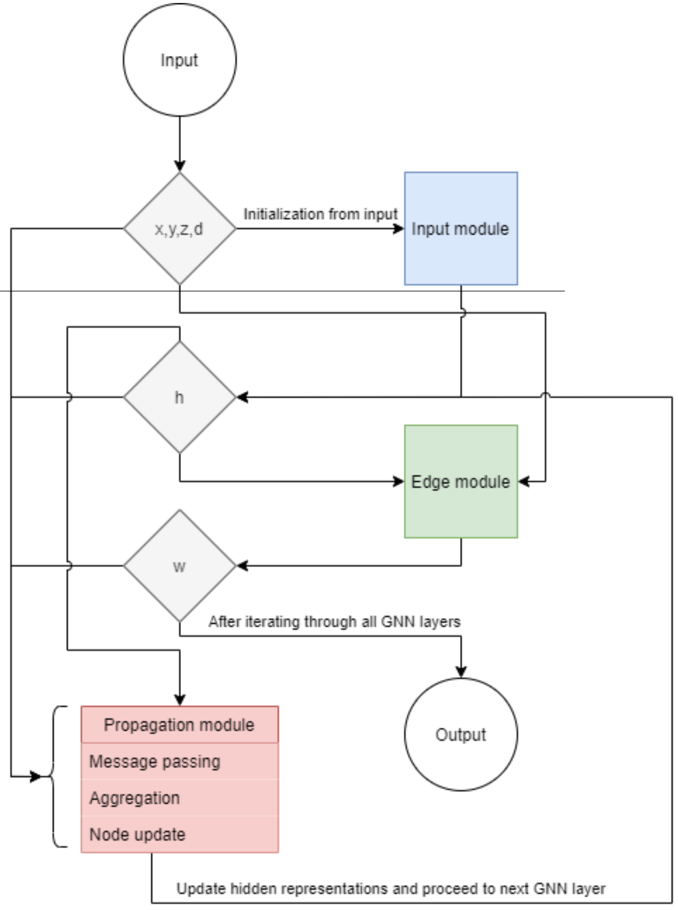
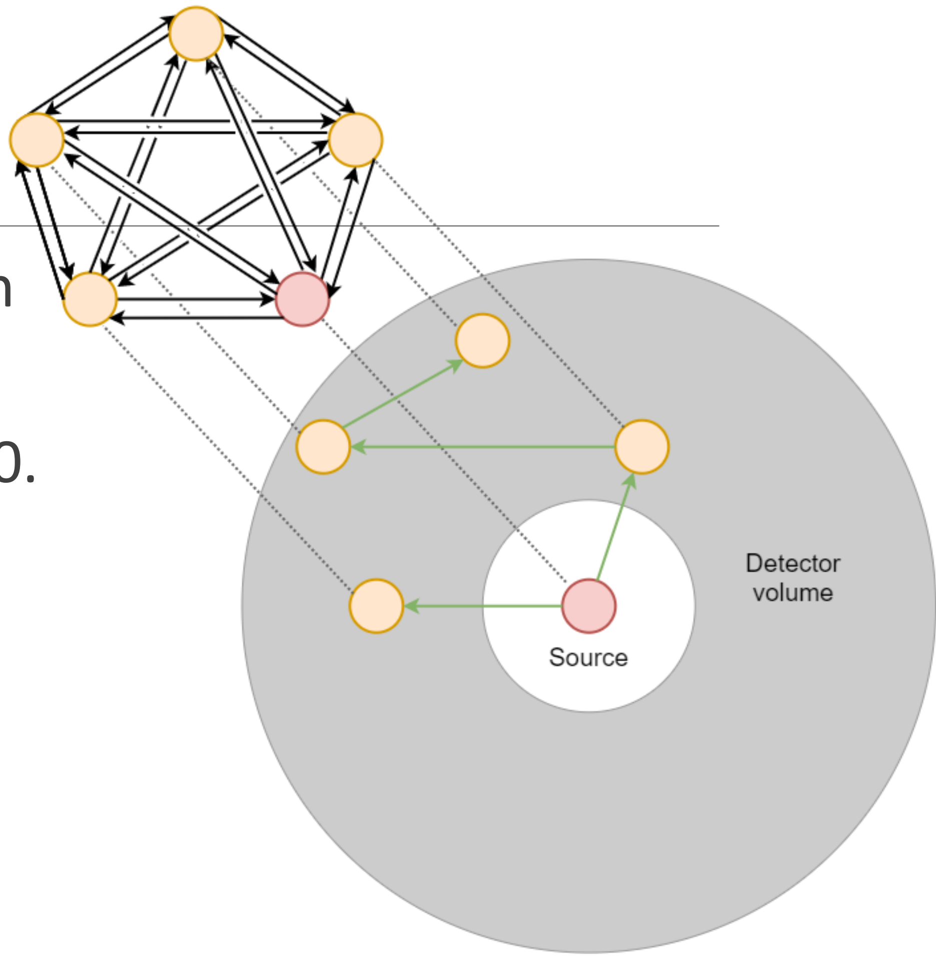


Figure 1: Schematic for information flow of the network.

Training



We used a set of continuous simulated energies to train the network.

Initial data range 10-1000 keV with a multiplicity of 1-10.

Two instances, one trained on data with packing and uncertainties and one on data lacking these features.

Resample uncertainties in energy deposits for packed data, given it’s distribution, for each epoch.

Training (unpacked)

Terminating training after plateau learning rate reduction and no improvements made to loss for a set period of epochs.

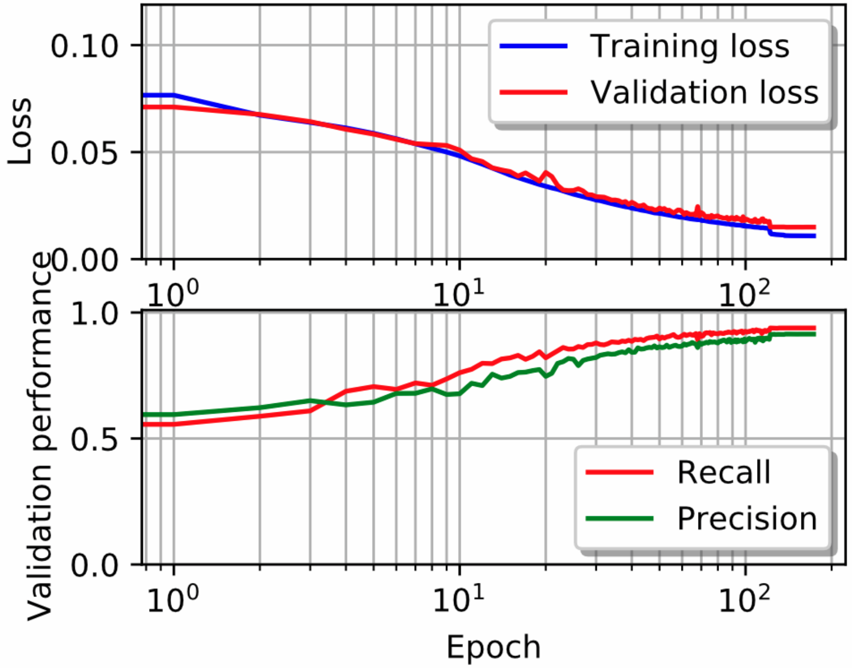


Figure 2: Training terminating after no improvements for 10 successive epochs after plateau learning rate reduction. Recall and precision given for a 0.2 threshold classification on edge weights.

Results (unpacked)

Peak to total: 85%.

Deficit at higher energies.

Excess at lower energies.

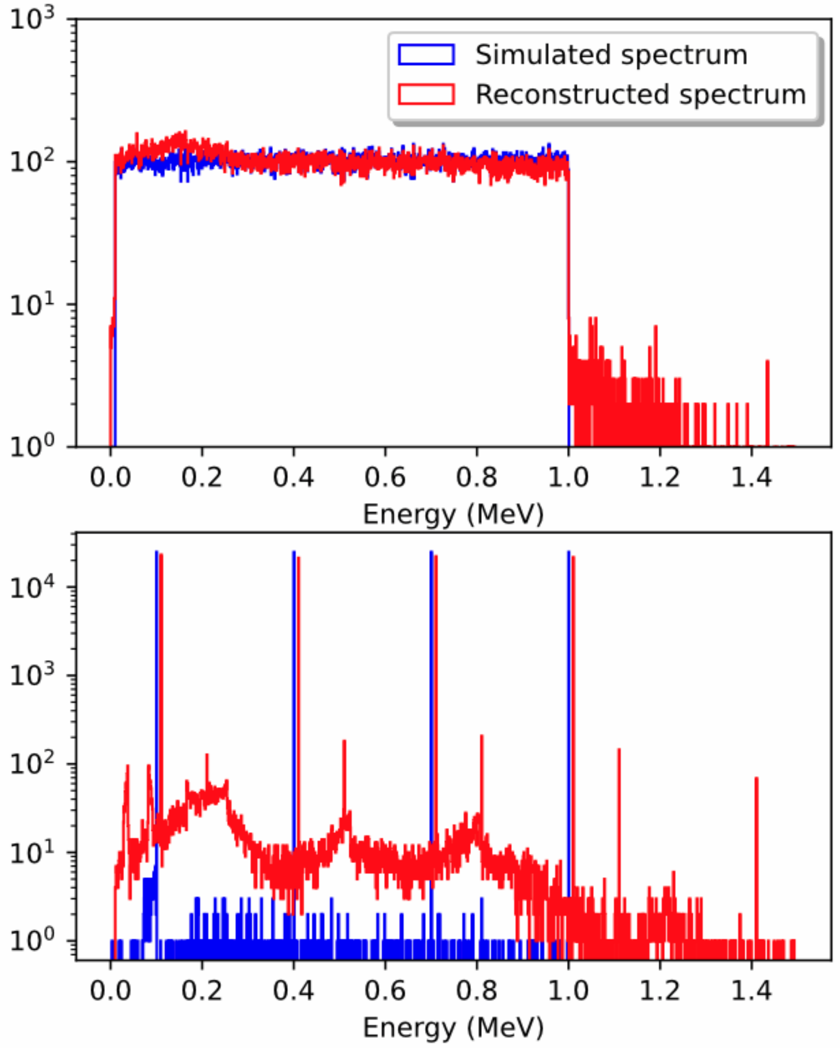


Figure 3: The underlying spectra (blue) and its reconstruction (red) for continuous and discrete data.

Results (unpacked)

Let us take a closer look at where things go wrong.

-Peak at 100-250 keV.

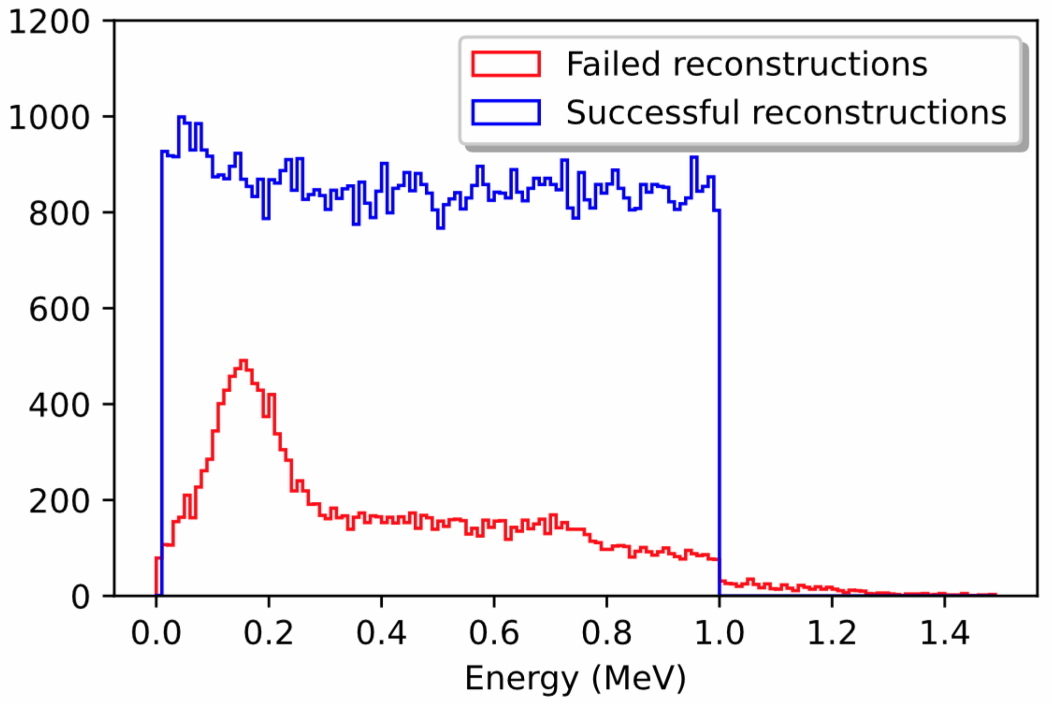


Figure 4: Correctly reconstructed part of the spectrum (blue) flawed reconstructions (red).

Results (unpacked)

Let us take a closer look at where things go wrong.

-Peak at 100-250 keV.

Overabundance of single interaction tracks.

Confusion regarding photo-absorption/Compton cross-section in energy region?

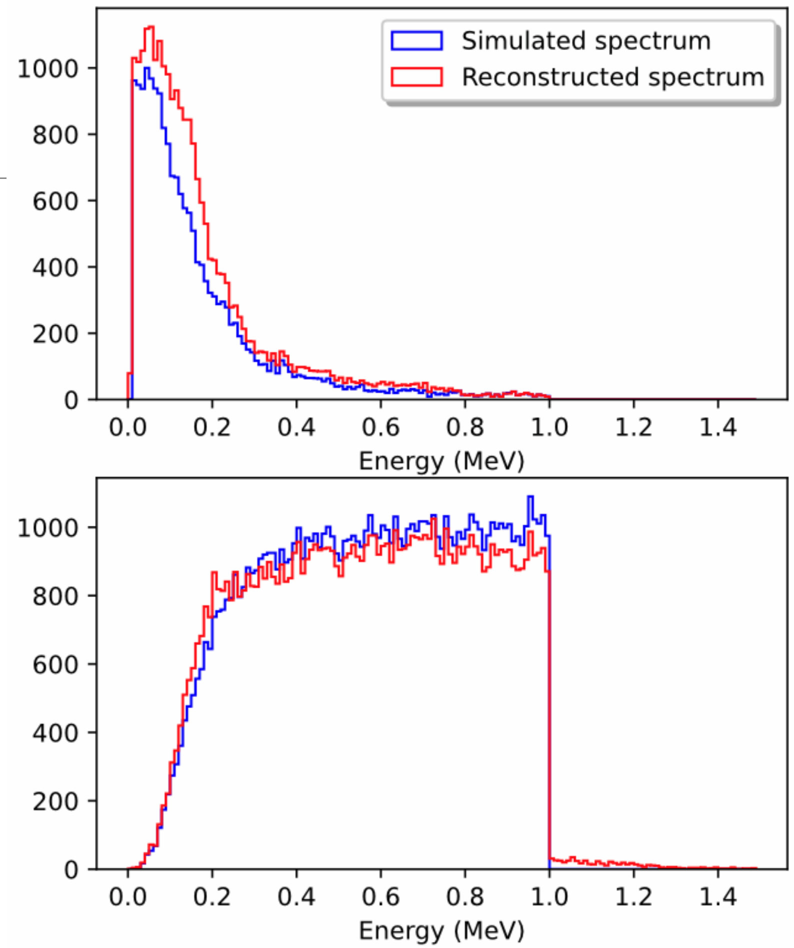


Figure 5: Comparing the reconstructions (red) to the true spectra (blue) where the top figure show tracks containing only a single node and the bottom shows multi node tracks.

Results (unpacked)

Missing nodes are track reconstructions that have been divided into multiple tracks.

Entangled tracks contain nodes from different simulated tracks.

Wrong order tracks contain the right nodes, but has a connection between them that is flawed.

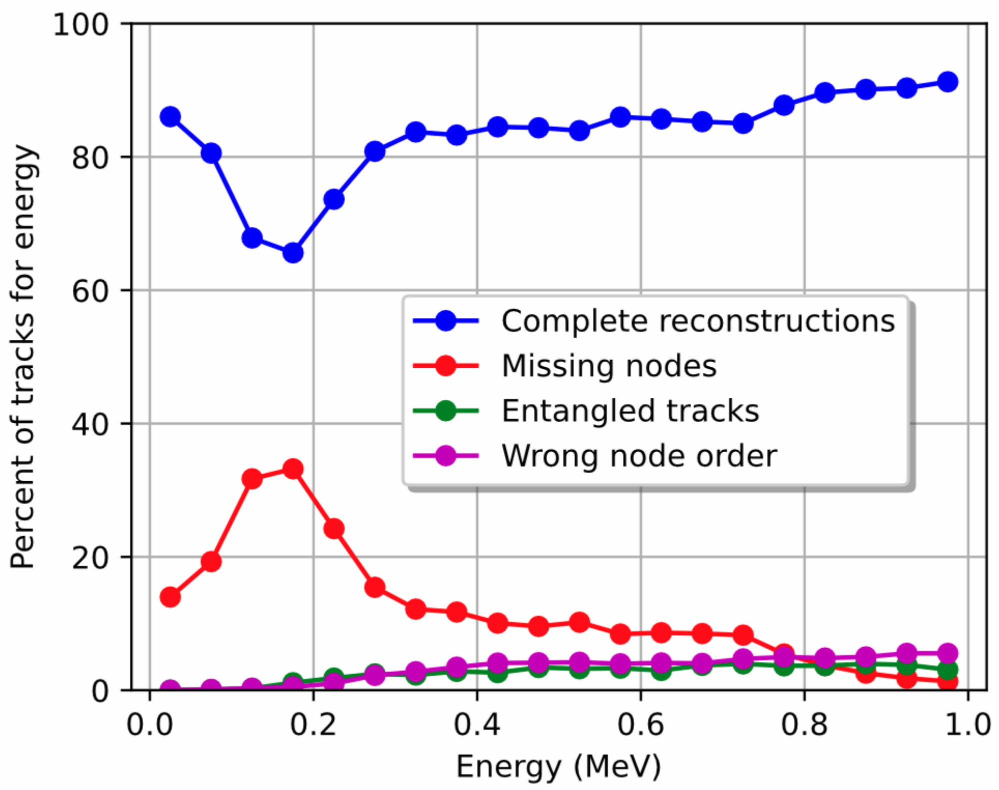


Figure 6: Track reconstruction results as a function of energy.

Results (unpacked)

Missing nodes are track reconstructions that have been divided into multiple tracks.

Entangled tracks contain nodes from different simulated tracks.

Wrong order tracks contain the right nodes, but has a connection between them that is flawed.

Fairly stable across different multiplicities.

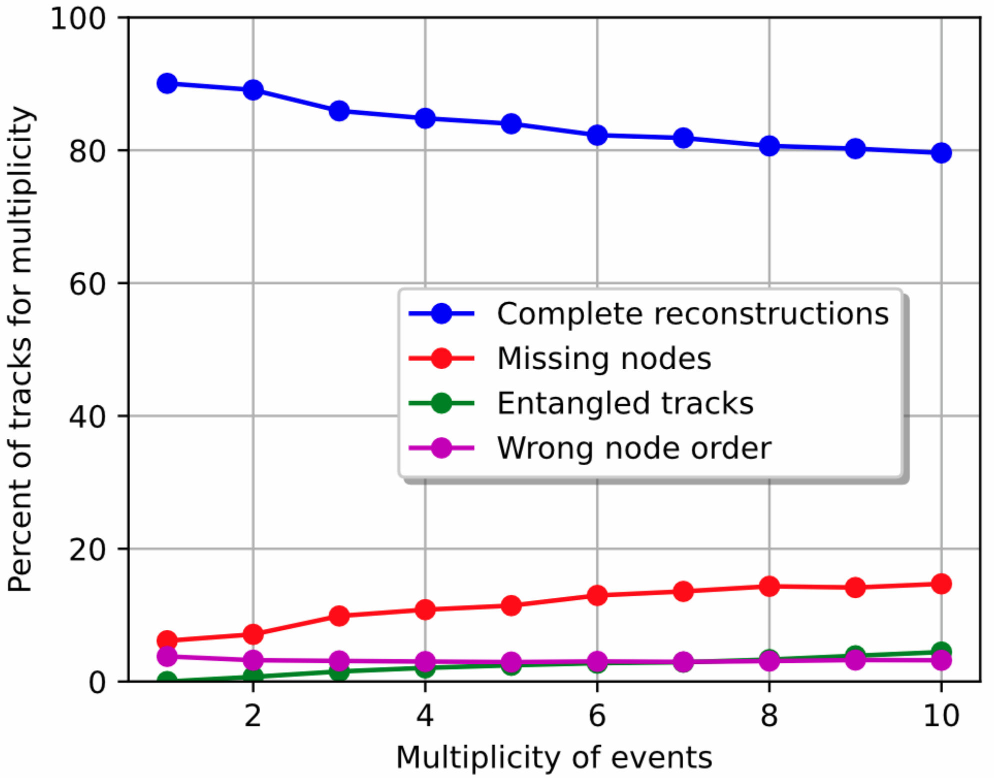


Figure 7: Track reconstruction results as a function of multiplicity.

Results (packed)

Packing distance 5 mm.

Using uncertainties from AGATA technical design report.

Estimated peak to total: 74%.

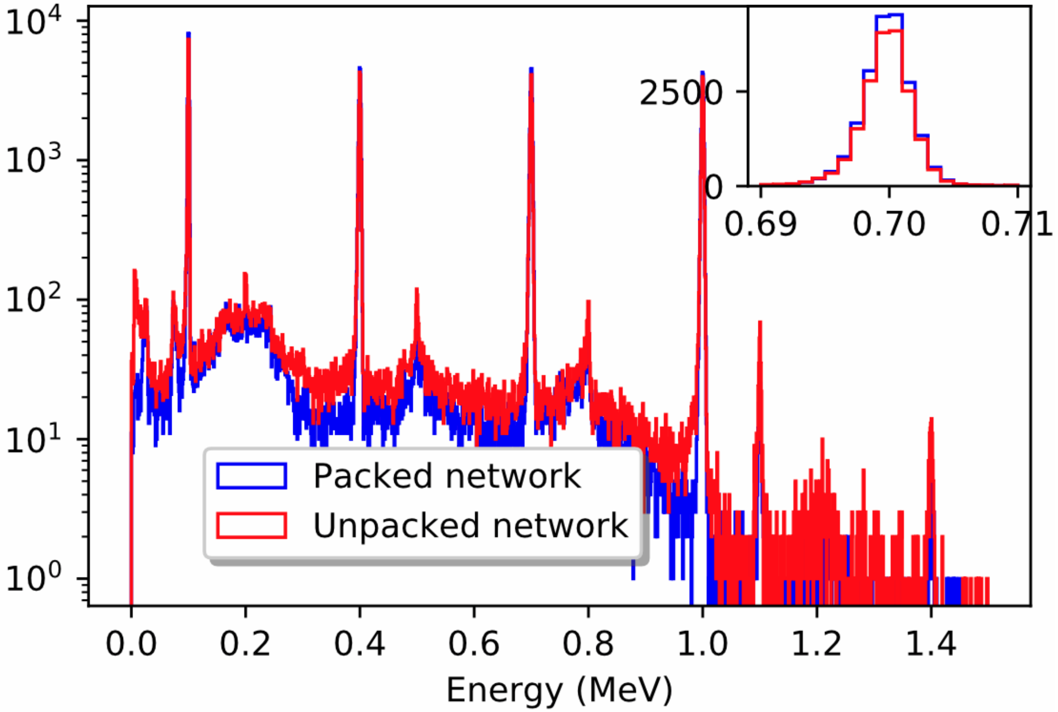
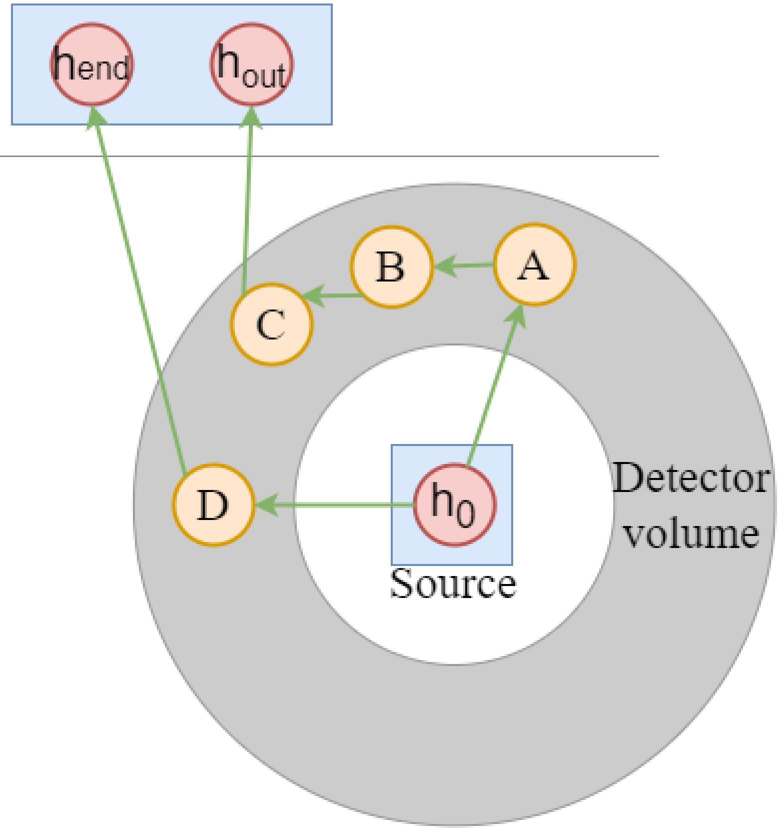


Figure 8: Spectra from packed data processed by networks trained on packed (blue) and unpacked (red) data.

Handling outscattering



In a geometry with a more realistic thickness we expect significant outscattering.

Introduce “helpernodes” to assist in filtering the reconstructions.

These nodes feature binary labels as an extension of the node features.

We start with nodes for source, termination inside the detector and outscattering.

Filtering and performance

In order to evaluate the method to back tracking and forward tracking, some changes where made.

-Higher energy interval (ignoring pair production).

-Higher multiplicity (up to 15).

-Energy independent position uncertainty.

-Thinner detector geometry (9 cm).

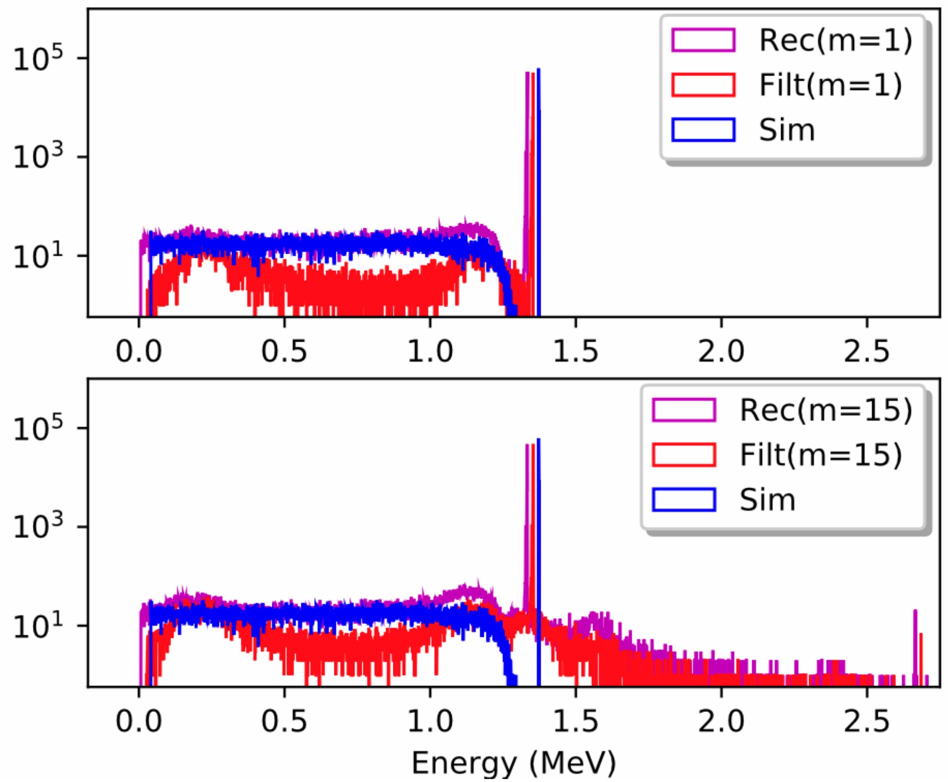


Figure 9: Result of filtering reconstructions based on helper nodes for a 1332 keV peak and thin shell.

Filtering and performance

Good increase in peak to total at little cost in efficiency.

Keep in mind that pair production reactions were not included in the simulation data.

Mult.	Algorithm	Pp eff. (%)	P/T (%)
1	BT	36.7	67.4
1	FT	53.6	75.2
1	GNN	60.7	66.0
1	GNNf	58.3	84.9
15	BT	27.3	53.7
15	FT	42.2	63.1
15	GNN	53.8	57.3
15	GNNf	51.0	73.7

Table 1: Efficiency and peak to total for our network with (GNN) and without filtering (GNNf) for single tracks as well as multiplicity 15. Values for back tracking (BT) and forward tracking (FT) methods taken from Lopez*.

**Lopez et al. Gamma-ray tracking algorithms: a comparison. NIM A, 533(3):456–466,2004*

Filtering and performance

Good increase in peak to total at little cost in efficiency.

Keep in mind that pair production reactions were not included in the simulation data.

Mult.	Algorithm	Pp eff. (%)	P/T (%)
1	BT	36.7	67.4
1	FT	53.6	75.2
1	GNN	60.7	66.0
1	GNNf	58.3	84.9
15	BT	27.3	53.7
15	FT	42.2	63.1
15	GNN	53.8	57.3
15	GNNf	51.0	73.7

Table 1: Efficiency and peak to total for our network with (GNN) and without filtering (GNNf) for single tracks as well as multiplicity 15. Values for back tracking (BT) and forward tracking (FT) methods taken from Lopez*.

**Lopez et al. Gamma-ray tracking algorithms: a comparison. NIM A, 533(3):456–466,2004*

Filtering and performance

Good increase in peak to total at little cost in efficiency.

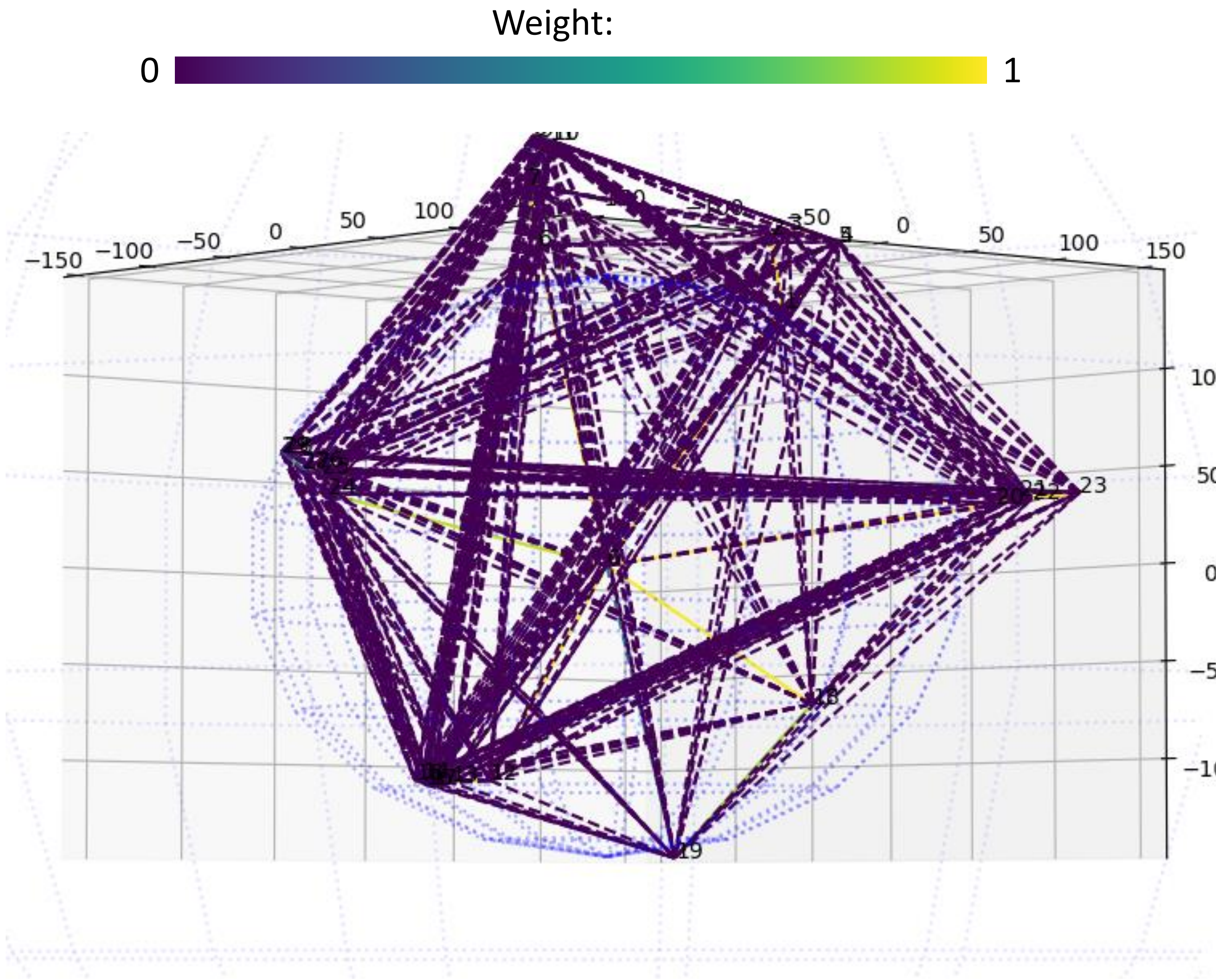
Keep in mind that pair production reactions were not included in the simulation data.

Mult.	Algorithm	Pp eff. (%)	P/T (%)
1	BT	36.7	67.4
1	FT	53.6	75.2
1	GNN	60.7	66.0
1	GNNf	58.3	84.9
15	BT	27.3	53.7
15	FT	42.2	63.1
15	GNN	53.8	57.3
15	GNNf	51.0	73.7

Table 1: Efficiency and peak to total for our network with (GNN) and without filtering (GNNf) for single tracks as well as multiplicity 15. Values for back tracking (BT) and forward tracking (FT) methods taken from Lopez*.

Future work

- Reduce complexity.
- Pair production.
- More complex detector geometry.
- Further use of helper/supernodes?


Weight:
0
1
−150
−100
−50
0
50
100
150
100
50
0
−50
−100
23
19

Weight:

Thank you!