Imperial College London

Graph neural nets for CMS particle track reconstruction

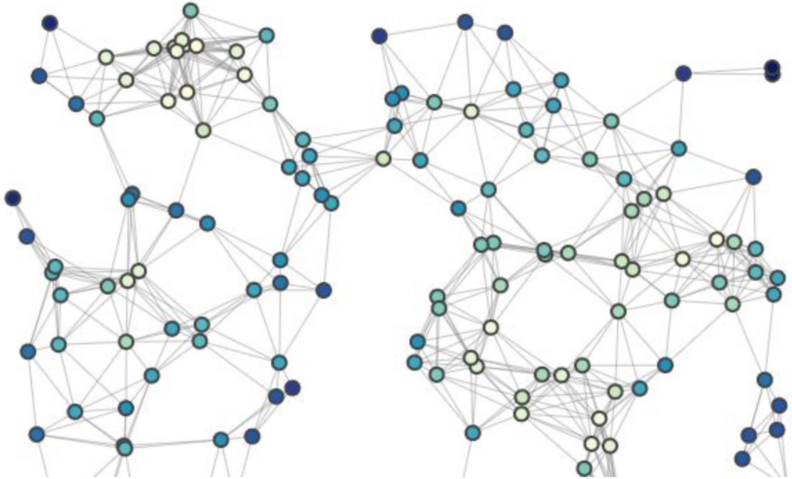
IOP HEPP Conference 03-06.04.22

Liv Helen Våge
Prof. Alex Tapper
liv.helen.vage@cern.ch

Content

Graph neural nets

- Theory
- Particle tracking
- Limitations

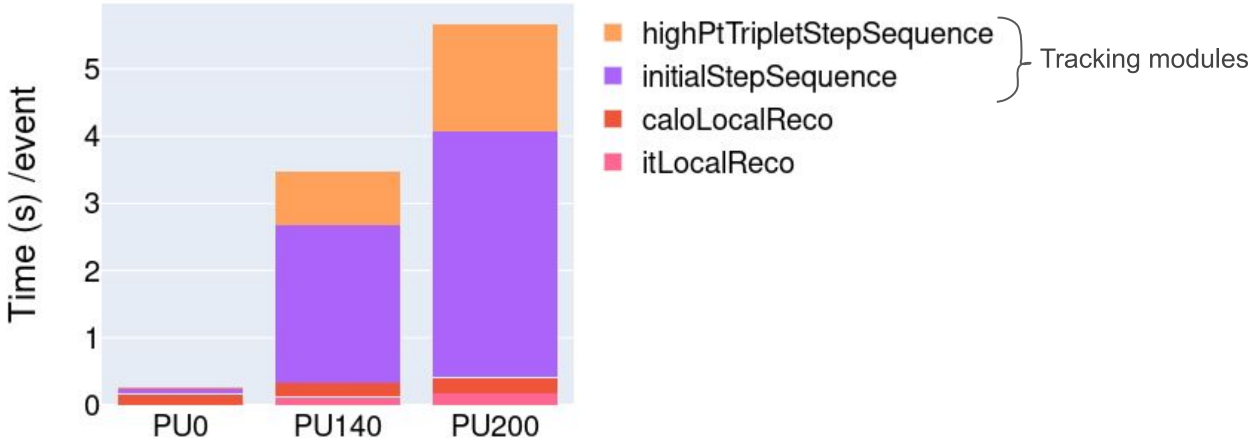


Reinforcement learning

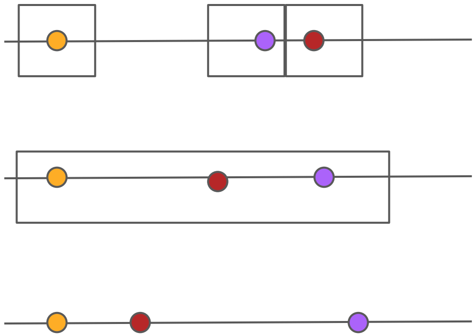
- Theory
- Preliminary results for graph building

Motivation

At high pileup track reconstruction takes up almost 50% of the HLT time budget



Kalman filter introduces combinatorial explosion



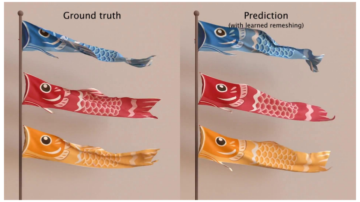
Graph neural nets

The power of graph neural nets come from aggregating and embedding neighbourhood information

Works on non-Euclidian data and result in non-sparse matrices

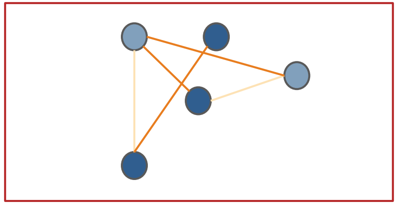
Often a “natural” way of representing geometric data

Successful applications



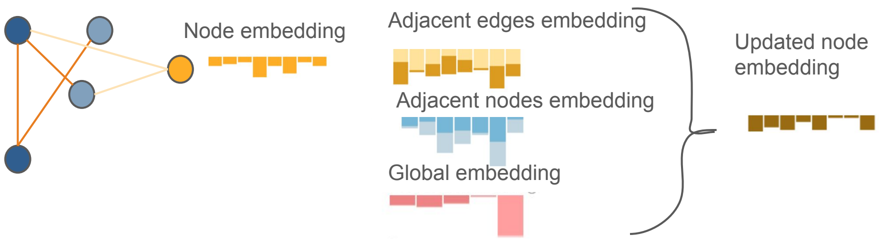
[Physics simulation](#)
[Drug development](#)
[Traffic flow](#)

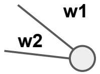
Graph description



Nodes	[0,1,0,0,1]
Edges	[0, 1, 1, 1, 0]
Adjacency list	[[0,1], [0,2], [1,3], [1,4], [3,4]]
Global:	[]

Algorithm





NN $y = \sigma\left(\mathbf{w}^{\mathbf{T}}\mathbf{h} + b\right)$

Update node

Message passing

$$\mathbf{h}_u^{(k)} = \sigma\left(\boxed{\mathbf{W}_{self}^{(k)}\mathbf{h}_u^{(k-1)}} + \boxed{\mathbf{W}_{neigh}^{(k)} \sum_{v\in\mathcal{N}(u)} \mathbf{h}_v^{(k-1)}} + \mathbf{b}(k)\right)$$

Graph neural nets for tracking

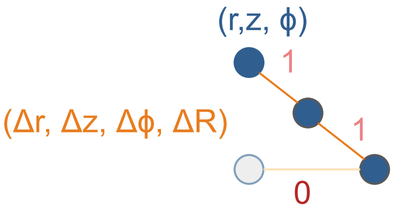
Reconstruct tracks by using GNN for binary edge prediction

Deep Mind's Interaction network used

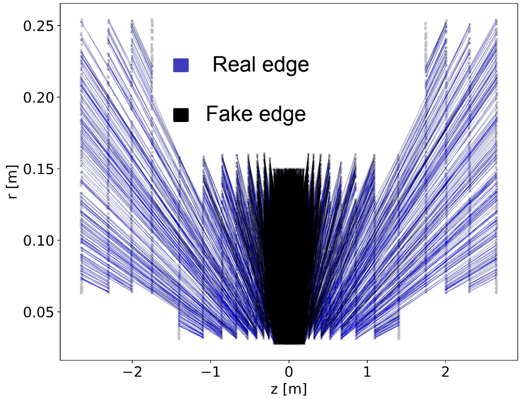
Promising performance with TrackML and CMS Phase 2 Monte Carlo data

Work pioneered by ExatrkX

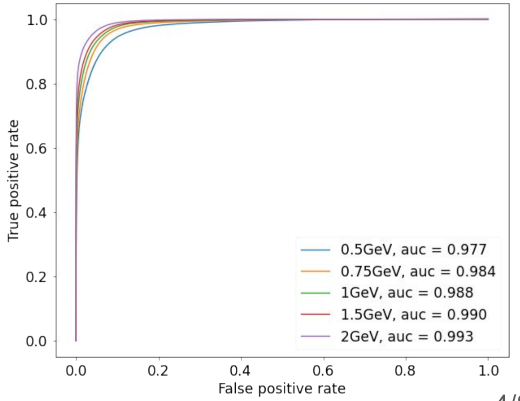
Thanks to a hackathon soon to be part of CMSSW



Built graph with $p_T > 0.5$ GeV

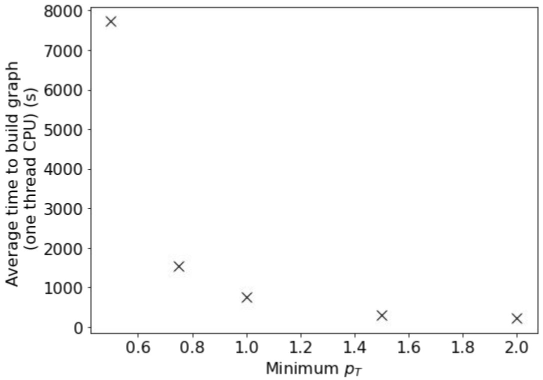


ROC curve for edge classification on CMS MC data



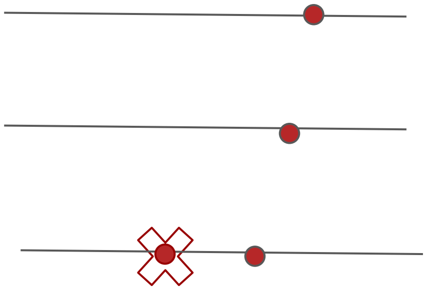
Limitations of graph neural nets for tracking

Graph building is slow



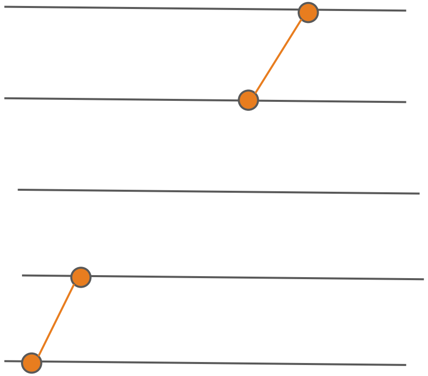
Graph building is orders of magnitude slower than inference

Allows only one hit per layer per track



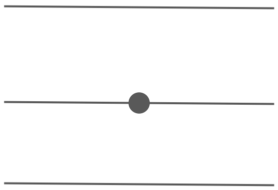
~ 70% of tracks have several hits in a layer (*)

Does not allow missing hits



~ 20 % of tracks have missing hits (*)

Filters out background hits



~ 10 % of simulated tracks have only one hit (*)

(*) Based on a small sample from CMS Phase 2 MC

Reinforcement learning

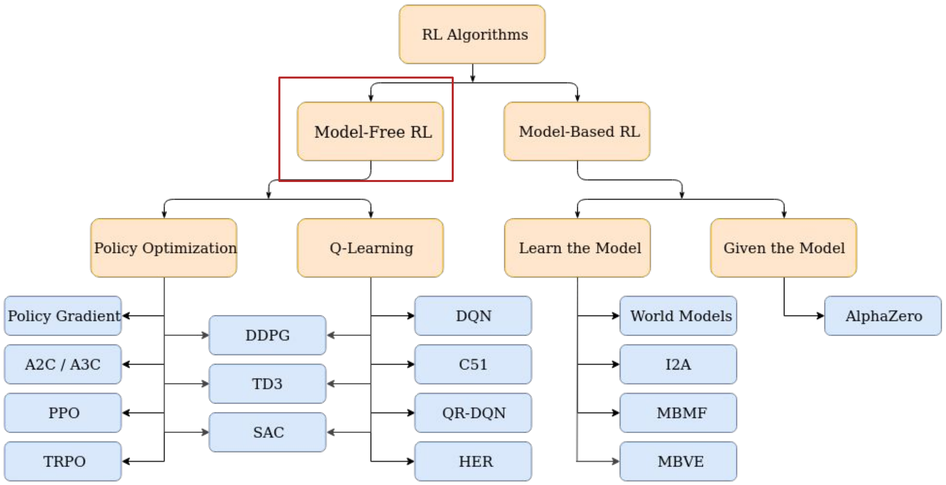
An agent learns actions by rewards

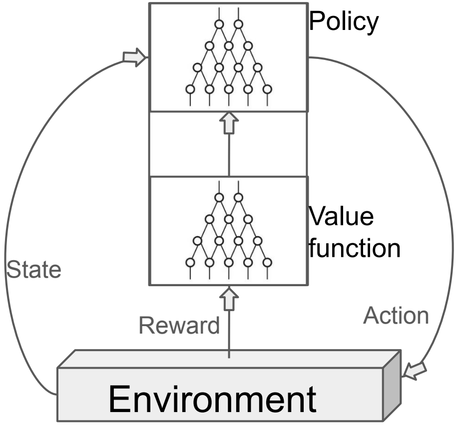
Requires that:

- Environment is intractable
- Agent receives data in response to action
- States are Markovian

Successful applications

[Playing games]
[Walking robots]
[Controlling plasma]



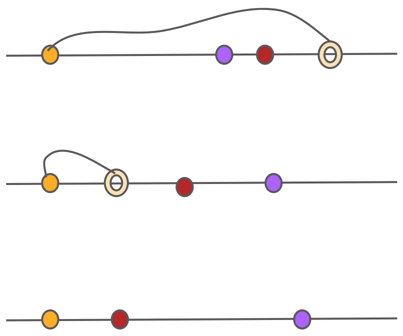


$$\mu_\theta(s_t)$$

$$\pi_\theta(s_t, \omega)$$

$$V^\pi(s) = E_{\tau \sim \pi}[R(\tau) | s_0 = s]$$

Reinforcement learning for graph building



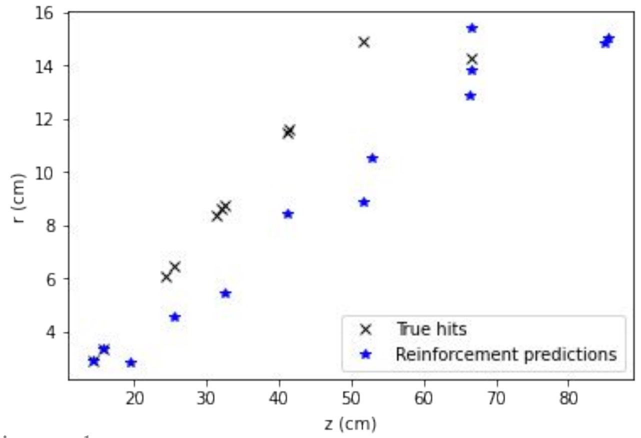
Reward is distance between predicted hit position and real hit position

Does not perform well enough to be a substitute for graph neural nets

Can contribute to smaller, less restricted graphs

Can in theory also learn how long track should be

Can be parallelised easily



Preliminary plot

Conclusion

- Graph neural nets are showing promise for particle tracking
- Reinforcement learning might help solve some of the problems in graph building

Further work:

- Refine reinforcement learning technique
- Integrate reinforcement learning with graph building
- Explore seeded graph building
- Accelerate on FPGA

Albert Einstein: Insanity Is Doing the Same Thing Over and Over Again and Expecting Different Results

Machine learning: