Accelerating particle tracking for the HL-LHC

02.11.21

Liv Våge

liv.helen.vage@cern.ch

Supervisor: Prof. Alex Tapper

Contents

1. Tracking at CMS
2. Graph neural nets
3. Future work

Particle tracking at CMS

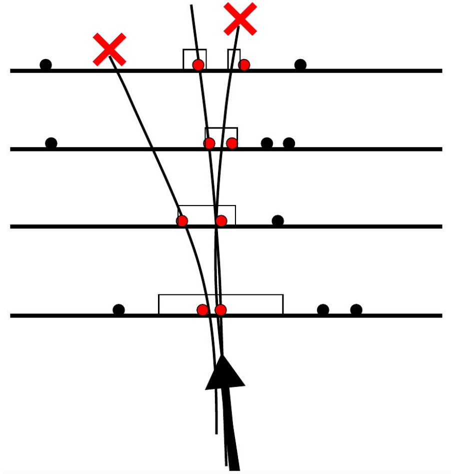
Particle tracking at CMS

Tracks are propagated one at a time

A Kalman filter is used to propagate and fit the tracks

When there are multiple compatible hits, multiple track candidates are made

Animation here



[1]

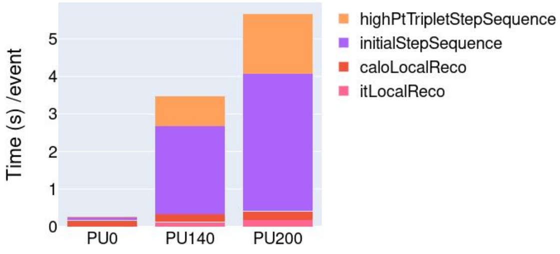
[2]

```
while candidate pool is not empty do:
  initialise empty temporary candidate pool;
  for all candidates in candidate pool do:
    find hits compatible with candidate;
    for all compatible hits do:
      update candidate state with hit;
      if updated state is finished then:
        add candidate to result tracks;
      else:
        add candidate to temporary candidate pool;
  sort temporary candidate pool by quality;
  discard all but best N candidates;
  replace candidate pool with temporary candidate pool;
return result tracks;
```

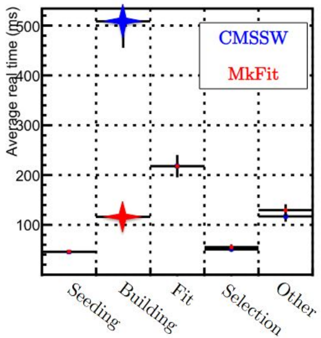
1. CMS Tracker Collaboration et al. "Stand-alone cosmic muon reconstruction before installation ofthe CMS silicon strip tracker". In:Journal of Instrumentation4.05 (2009), P05004.url:https://www.researchgate.net/publication/47684426_Stand-alone_cosmic_muon_reconstruction_before_installation_of_the_CMS_silicon_strip_tracker

2. Sioni Paris Summers. "Application of FPGAs to triggering in high energy physics". PhD thesis.Imperial Coll., London, 2018.url:https://cds.cern.ch/record/2647951

Particle tracking scales badly with pileup

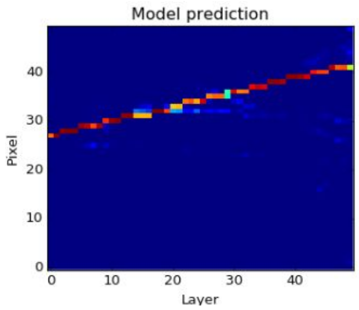


Accelerating the current algorithm might not be enough



[1]

Machine learning is one potential solution



[2]

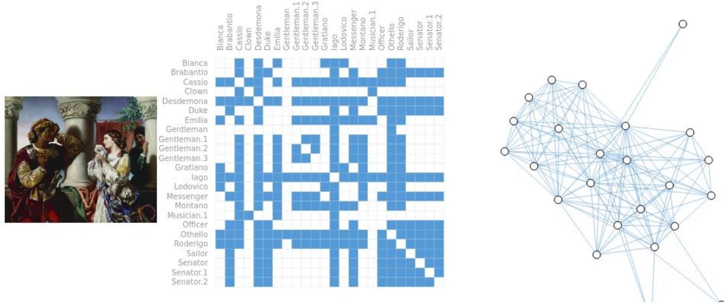
Graph neural nets is one of the most promising ML methods

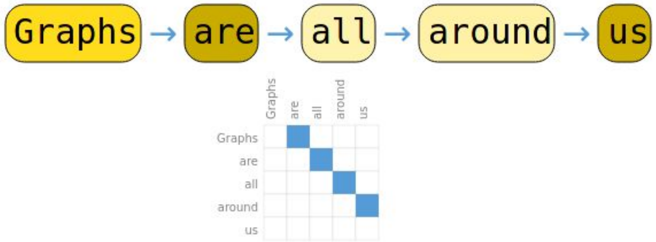
"We believe the GNN approach to be our most promising deep learning solution for addressing the problems in tracking at the HL-LHC" [3]

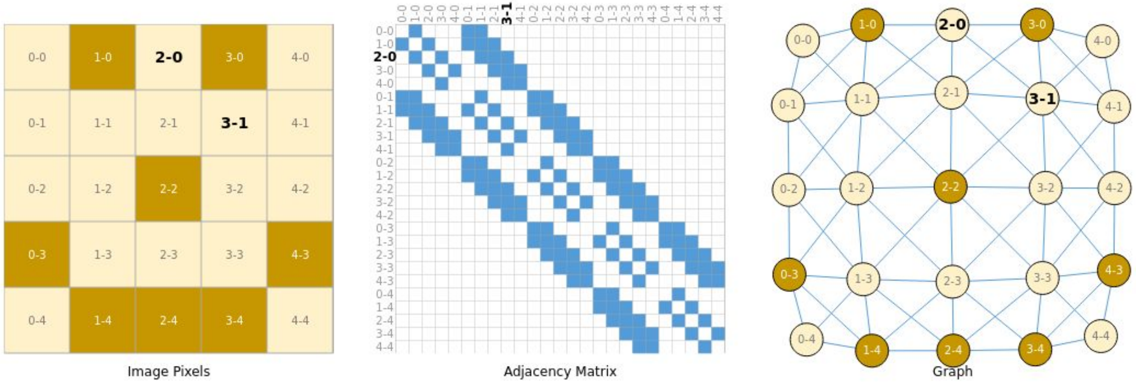
1. Cerati, G., Elmer, P. et al., 2019. Speeding up particle track reconstruction in the CMS detector using a vectorized and parallelized Kalman filter algorithm. *arXiv preprint arXiv:1906.11744*.
2. Farrell, S., Connecting the Dots 2017, https://indico.cern.ch/event/577003/contributions/2476580/attachments/1424886/2185488/Farrell_HEPTrkX_CTD2017.pdf
3. .Farrell, S., Calafiura, P., Mudigonda, et al., 2018. Novel deep learning methods for track reconstruction. *arXiv preprint arXiv:1810.06111*.

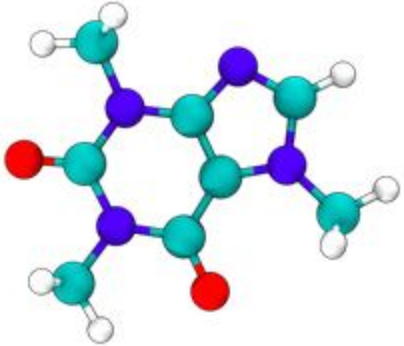
GNN for particle tracking

Why do we want graphs - Graph structures



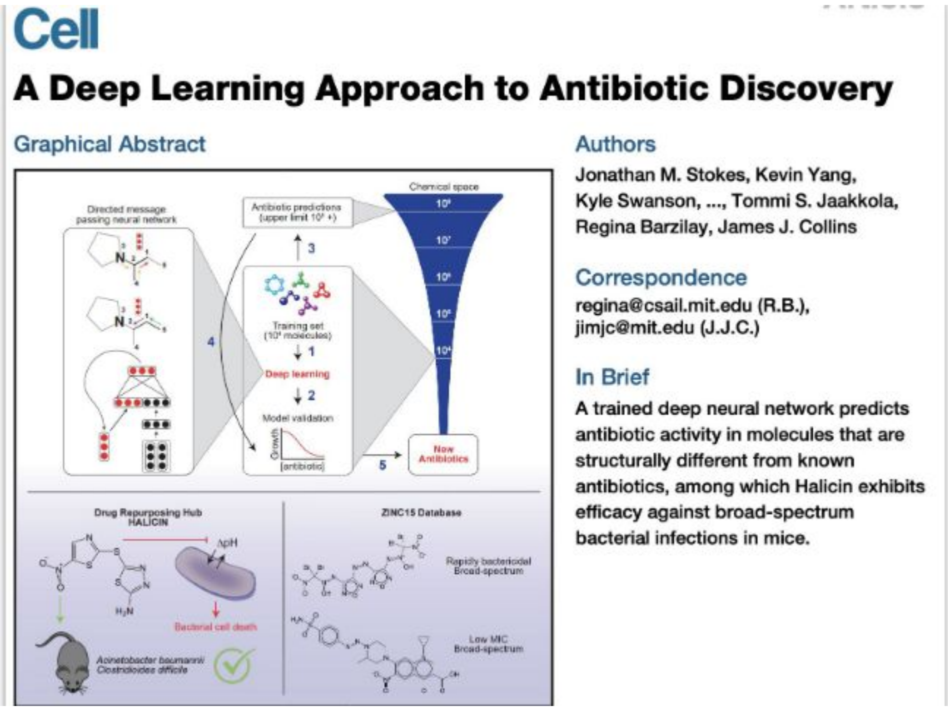






[1] Figures from: https://distill.pub/2021/gnn-intro/

Why do we want graphs - Successful GNNs



Cell

A Deep Learning Approach to Antibiotic Discovery

Graphical Abstract

Authors

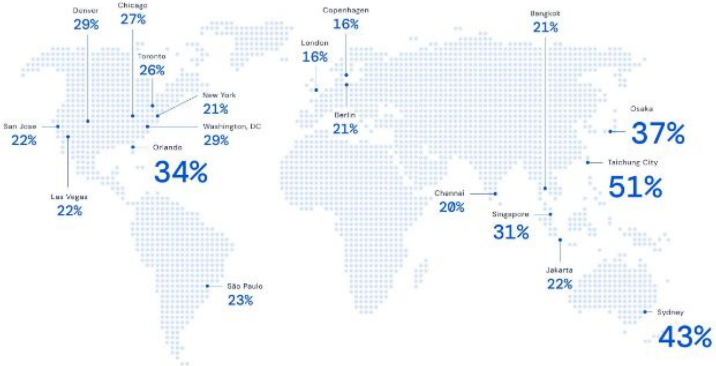
Jonathan M. Stokes, Kevin Yang, Kyle Swanson, ..., Tommi S. Jaakkola, Regina Barzilay, James J. Collins

Correspondence

regina@csail.mit.edu (R.B.), jimjc@mit.edu (J.J.C.)

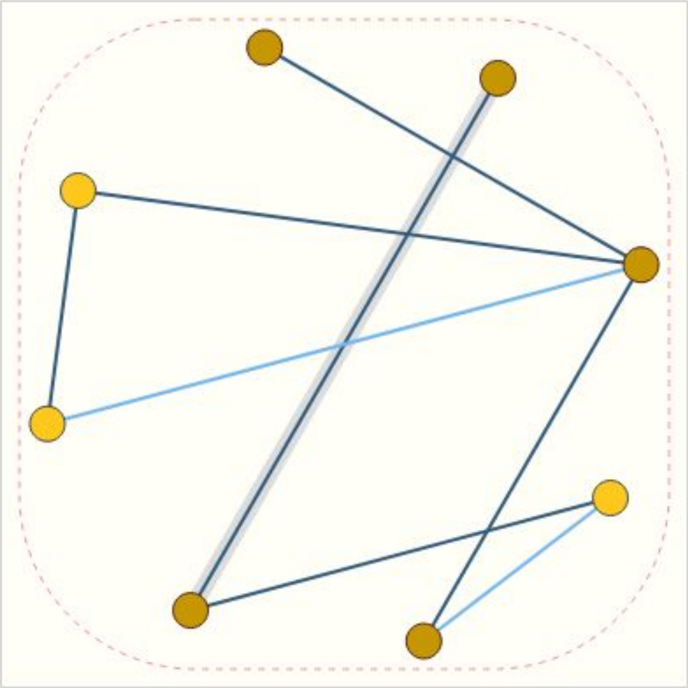
In Brief

A trained deep neural network predicts antibiotic activity in molecules that are structurally different from known antibiotics, among which Halicin exhibits efficacy against broad-spectrum bacterial infections in mice.



[1] Figures from: https://petar-v.com/talks/GNN-Wednesday.pdf

How to describe a graph - Adjacency lists



Nodes

[0, 1, 1, 0, 0, 1, 1, 1] $O(n_{nodes})$

Edges

[2, 1, 1, 1, 1, 2, 1, 1] $O(n_{nodes})$

Adjacency List

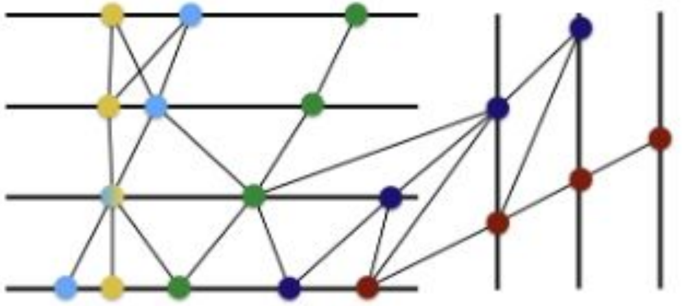
[[1, 0], [2, 0], [4, 3], [6, 2], [7, 1], [7, 3], [7, 4], [7, 5]] $O(n_{edges})$

Global

0

[1] Figure from: https://distill.pub/2021/gnn-intro/

GNN for tracking



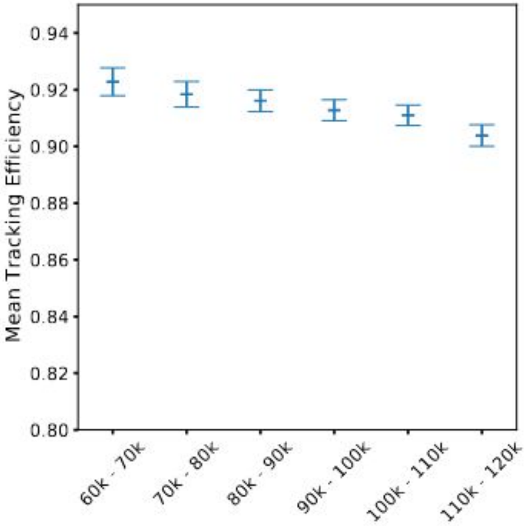
1. Create edges between all hits in adjacent layers (subject to physics constraints)

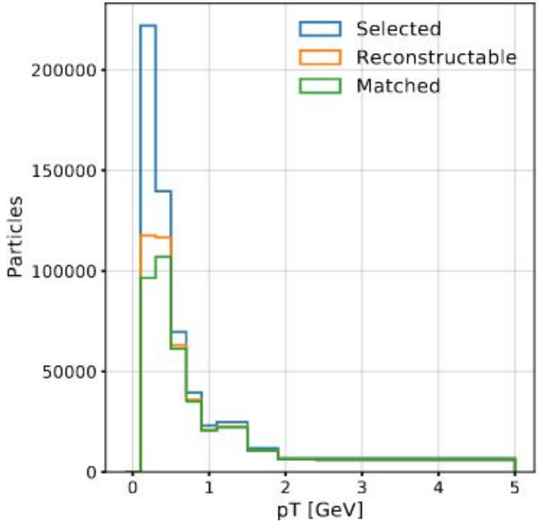
2. Train GNN to do binary classification on the edges - is it a real track or not

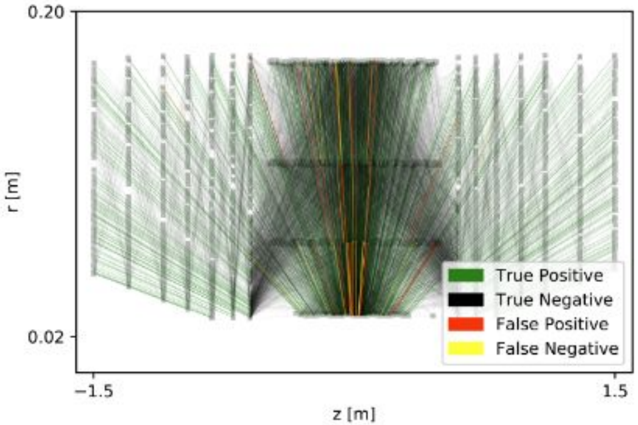
[1] Farrell, S., Calafiura, P., Mudigonda, et al., 2018. Novel deep learning methods for track reconstruction. *arXiv preprint arXiv:1810.06111*.

Results with TrackML data

Promising performance







[1]

Results with CMS MC data

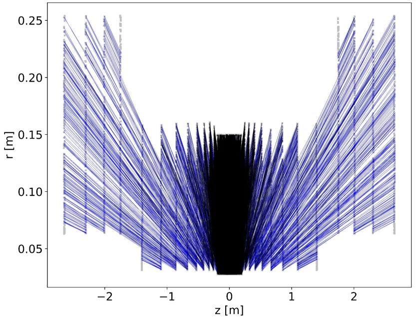
- CMSSW, 1000 ttbar events

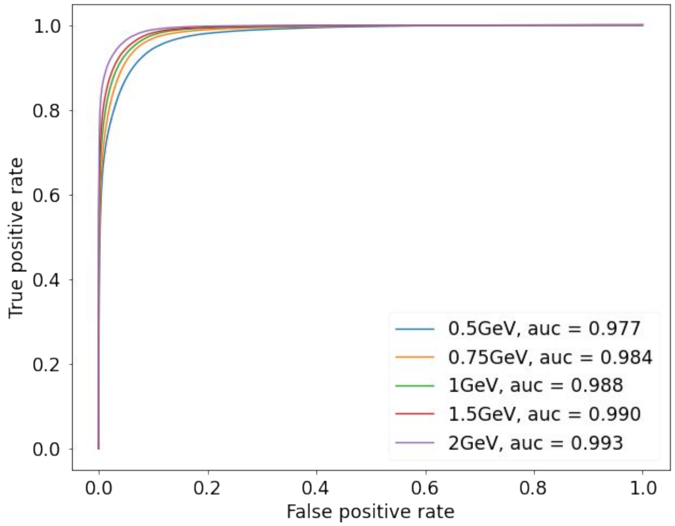
- Good performance

- Graph building takes a long time

  (GNN training < 15 minutes on GPU, graph building takes up to a few hours on CPU)

- Still many simplifications

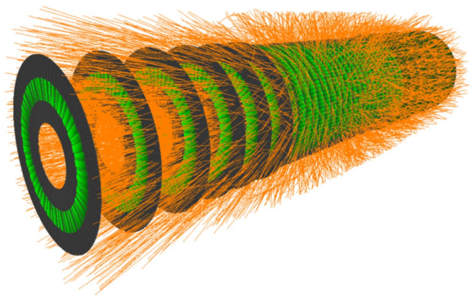




Future work

Extend the GNN

- Extend to outer detector
- Allow multiple hits per layer
- Allow missing hits



Create workflow in CMSSW

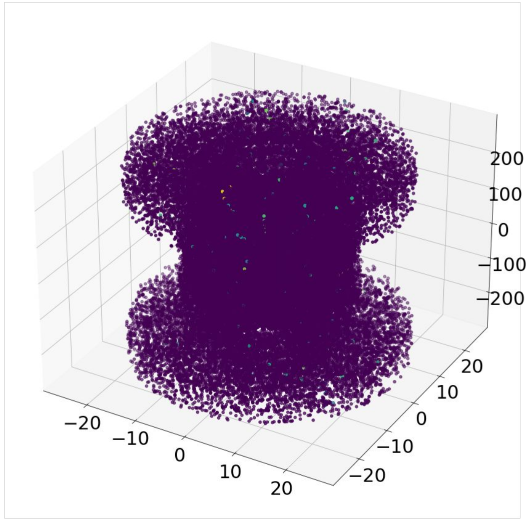
- Complete workflow created during hackathon
- Upcoming talk about the hackathon

Accelerate on FPGA

[1] Heintz, Aneesh, et al. "Accelerated charged particle tracking with graph neural networks on FPGAs." *arXiv preprint arXiv:2012.01563* (2020).

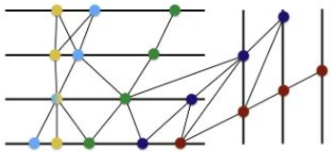
Summary

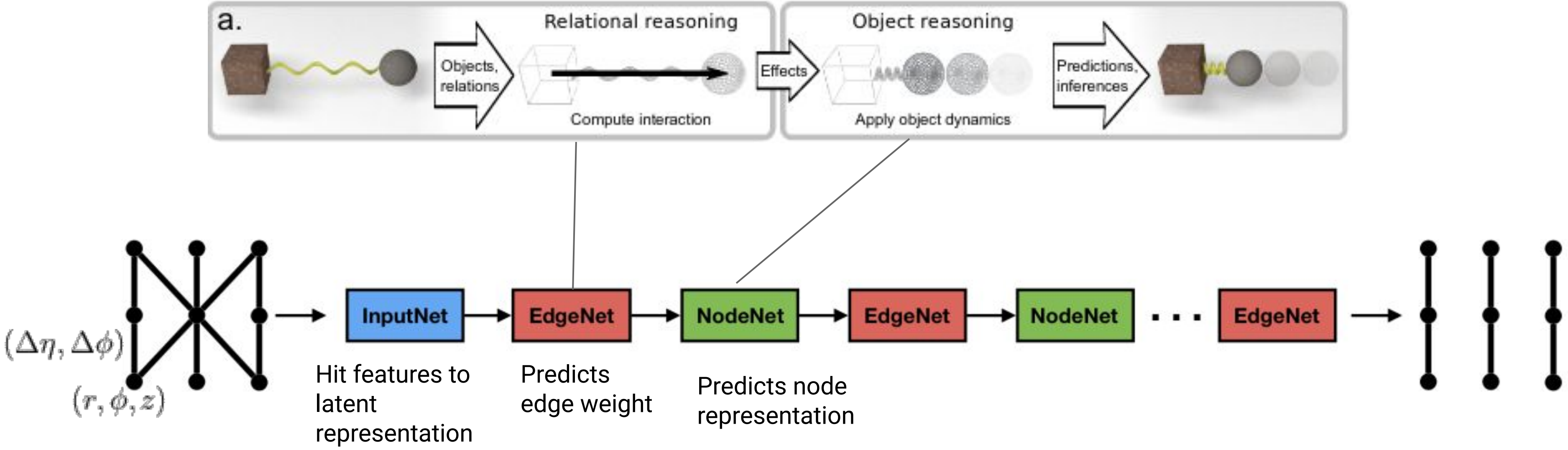
- GNN for particle tracking is showing promise
- Many simplifications need to be addressed
- Could be a candidate for FPGA acceleration



Backup

Interaction network

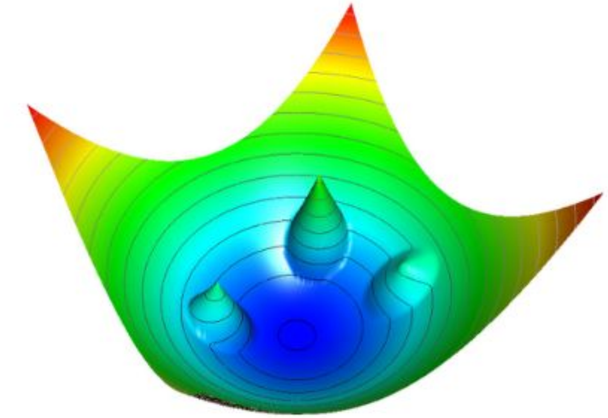


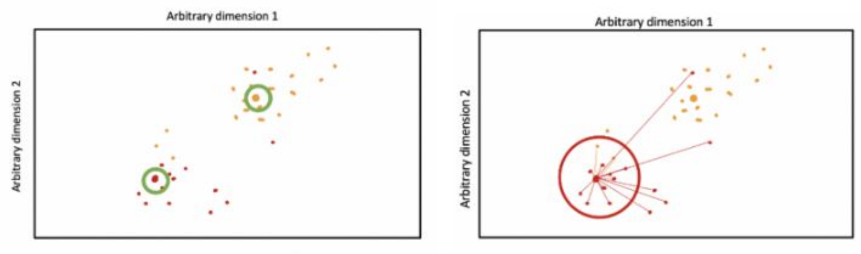


[1] Figures and interaction network from [Battaglia, P., Razvan P. et.al, 2016 Google DeepMind]

CMS GNN Hackathon

- Implementing graph building in CMSSW
- Interface with SONIC
- Track building and fitting
- Performance evaluation
- Object condensation



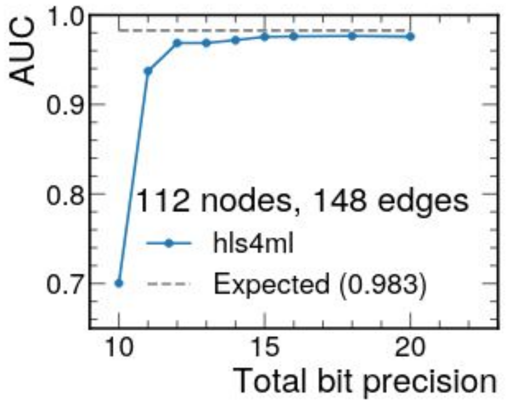


FPGA acceleration

- Already tried in [1]

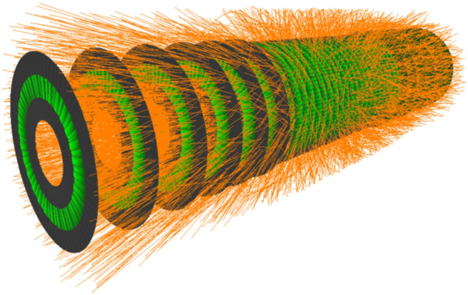
- Main challenge is having small enough GNN while keeping high performance

- Graph building should also be implemented on FPGA



[1] Heintz, Aneesh, et al. "Accelerated charged particle tracking with graph neural networks on FPGAs." *arXiv preprint arXiv:2012.01563* (2020).

Limitations



Limitation

- Lower bound on momentum
- Only inner tracker
- One hit per layer per particle
- No missing hits

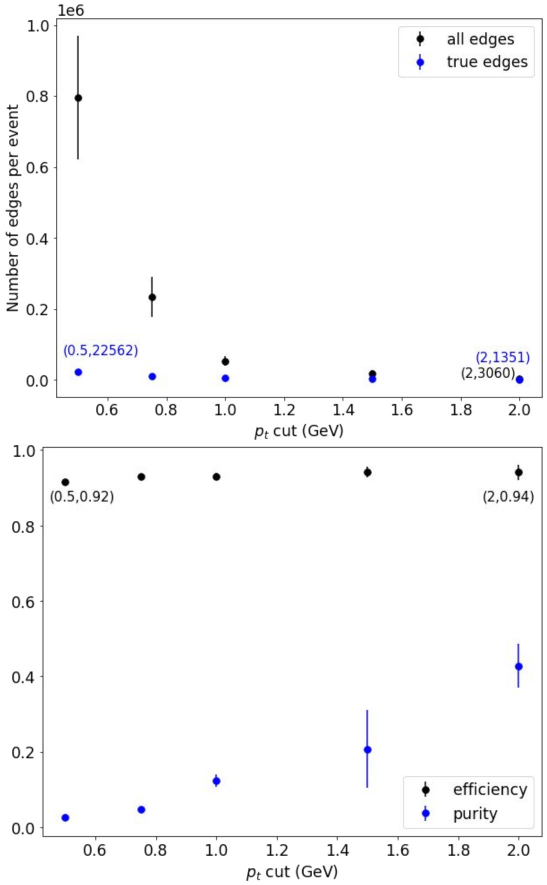
Potential solution

- Simplify NN architecture
- Reduce fake edges built by imposing better physics constraints
- Duplicate removal, allow shared hits, special treatment of low momentum particles
- Allow building segments between non-adjacent layers

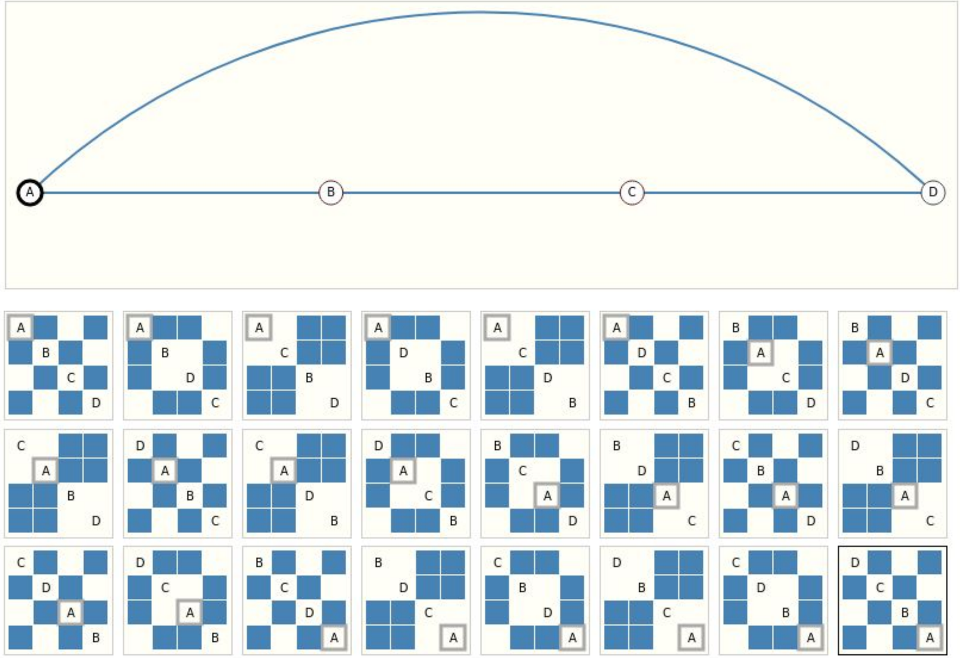
Results with CMS MC data

- Potentially millions of edges

- Somewhat high efficiency, low purity



How to describe a graph - Adjacency matrices



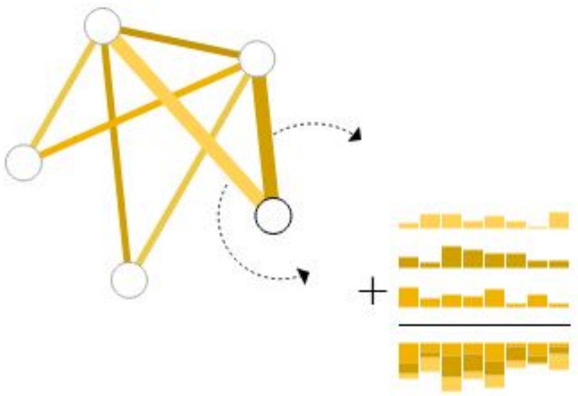
Not permutation invariant

Very sparse

$O(n^2_{nodes})$

[1] Figures from: https://distill.pub/2021/gnn-intro/

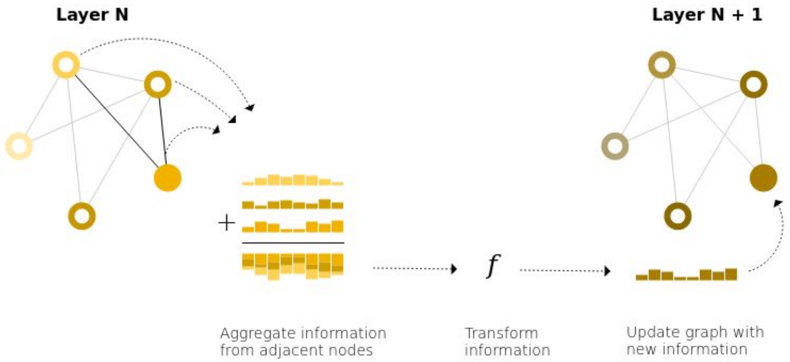
How to describe a graph - Permutation invariance and NNs



1. For each node in the graph, *gather* all the neighboring node embeddings (or messages)

2. Aggregate all messages via an aggregate function (like sum)

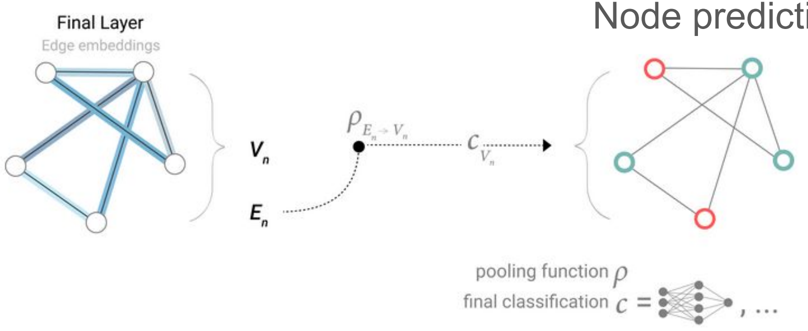
3. All pooled messages are passed through an *update function*, usually a learned neural network



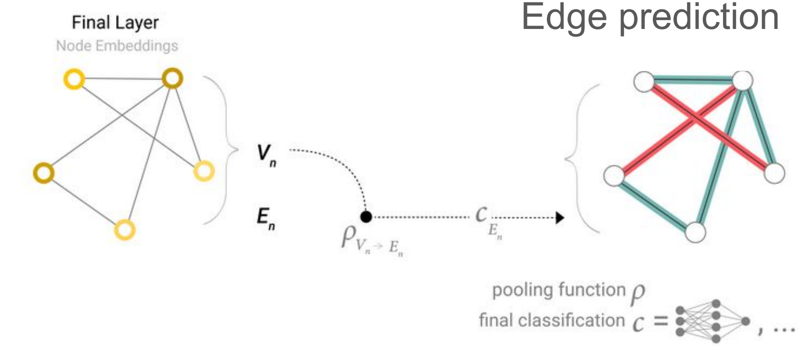
[1] Figure and text from: https://distill.pub/2021/gnn-intro/

What can we do with GNNs

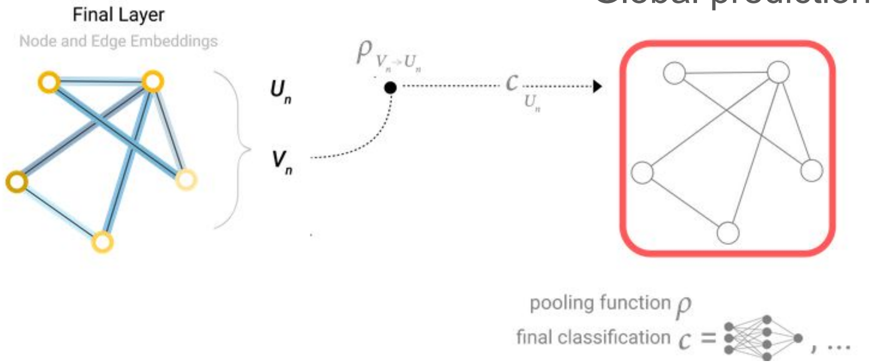
Node prediction



Edge prediction



Global prediction



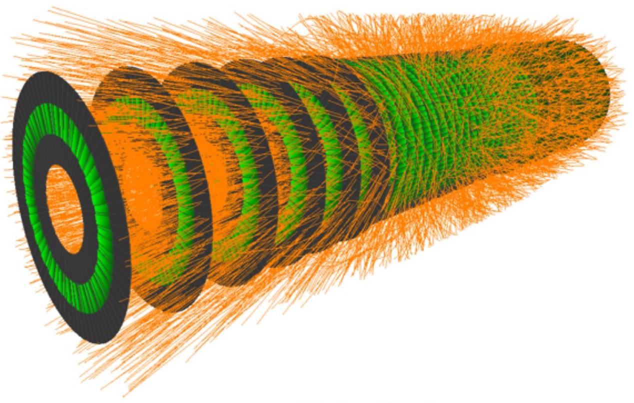
[1] Figures from: https://distill.pub/2021/gnn-intro/

GNN for HEP

Shlomi J., Battaglia P. and Vlimant J., 2020
[2]

Graph classification	Node classification / regression	Edge classification
Jet tagging	Pileup mitigation	Tracking
Event classification	Calorimeter reconstruction	Secondary vertex reconstruction
	Particle flow reconstruction	
	Efficiency parameterization	

Previous ML tracking work



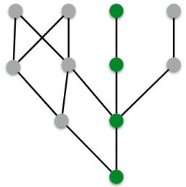
Exa.TrkX (earlier Hep.TrkX) have tried several ML approaches

They believe GNN is the most promising approach

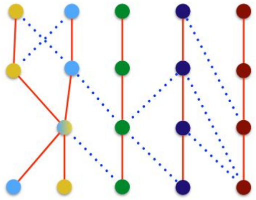
People involved:

- Maria Spiropulu, Jean-Roch Vlimant (Caltech)
- Giuseppe Cerati, Lindsey Gray, Thomas Klijnsma, Jim Kowalkowski (FNAL)
- Paolo Calafiura (PI), Nick Choma, Sean Conlon, Steven Farrell, Xiangyang Ju, Daniel Murnane,
- Ankit Agrawal, Alexandra Day, Claire Lee, Wei-keng Liao, (Northwestern)
- Gage DeZoort, Savannah Thais (Princeton)
- Pierre Cote De Soux, François Drielsma, Kasuhiro Terao, Tracy Usher (SLAC)
- Adam Aurisano, Jeremy Hewes (UCincinnati)
- Markus Atkinson, Mark Neubauer (UIUC)
- Aditi Chauhan, Alex Schuy, Shih-Chieh Hsu (UWashington)
- Alex Ballow, Alina Lazar (Youngstown State)

Hit classification



Segment classification



[1] Figure from: https://www.kaggle.com/c/trackml-particle-identification and Hep.TrkX, 2018, Farrell S., Calafiura P. et.al for Connecting the Dots 2018 [2],[3]