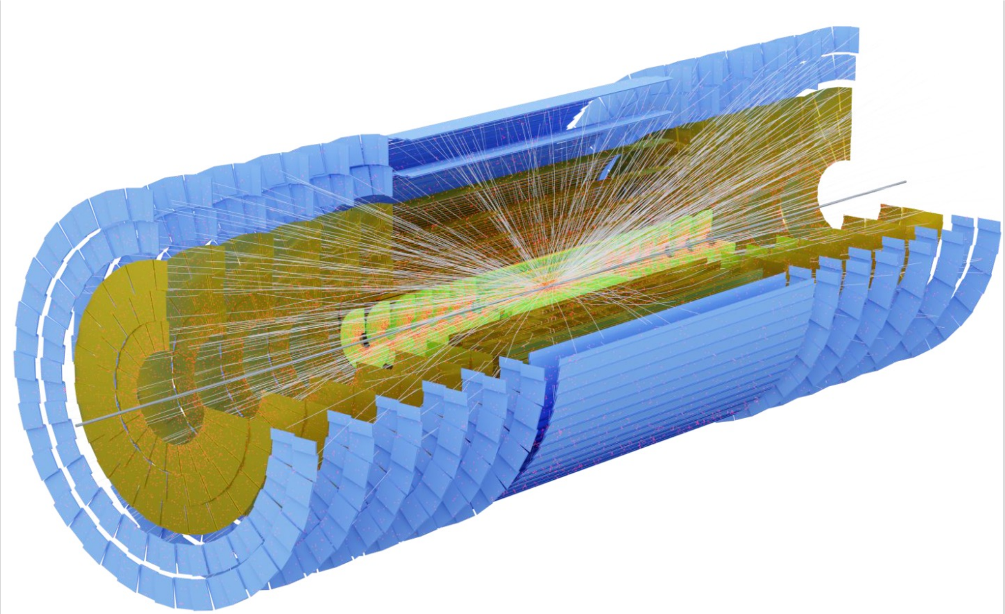
TrackOpt project status update

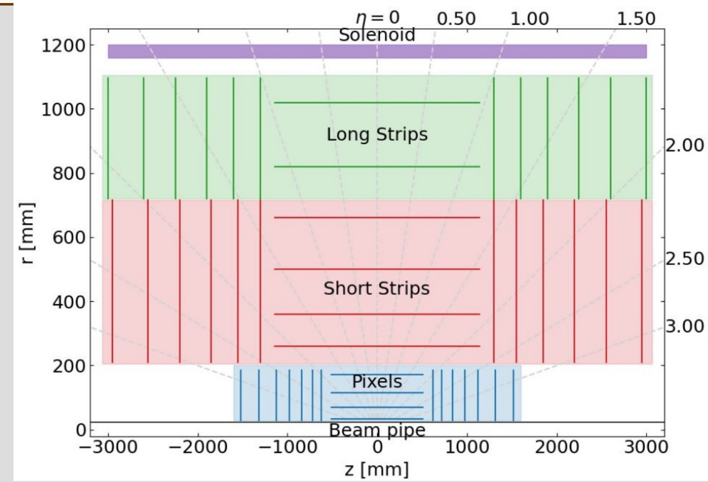
V. Kostyukhin
Siegen university

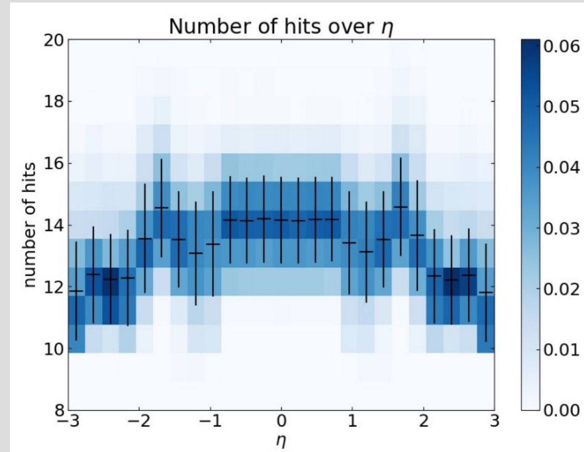
Open Data Detector + ACTS

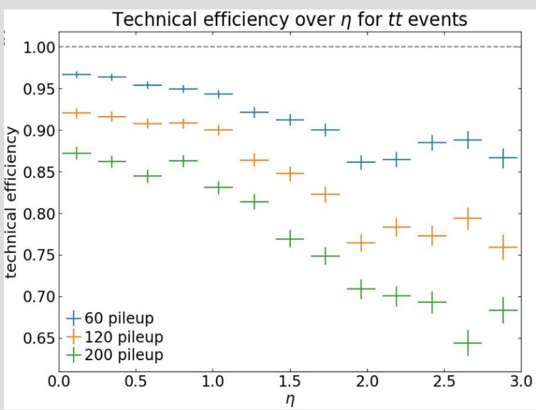
Universität Siegen

ODD – example detector for collider experiments. 2T magnetic field

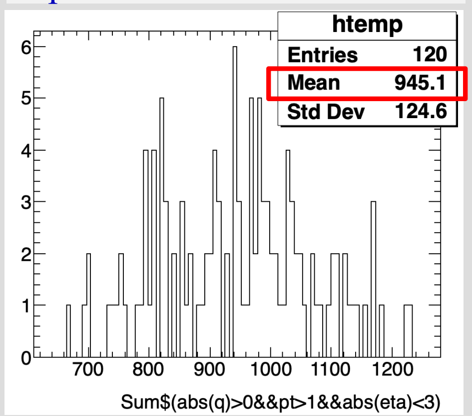








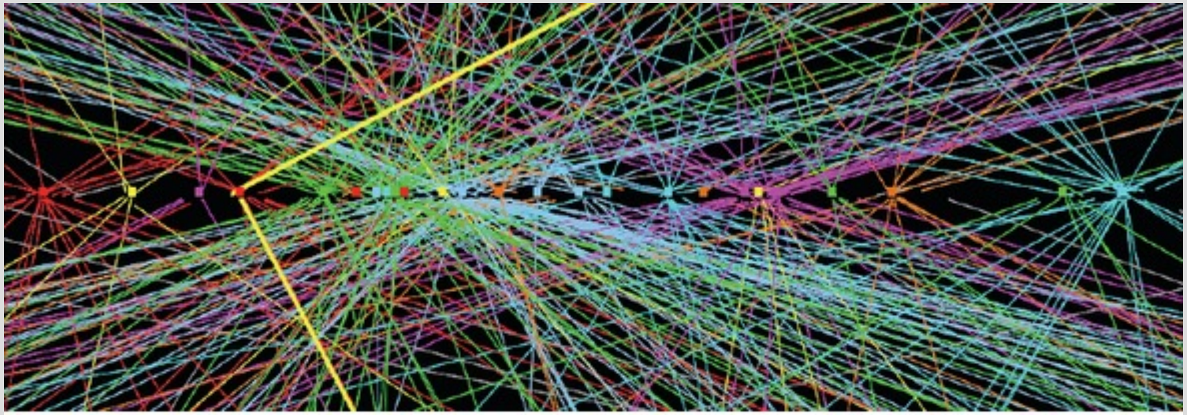
Nparticles for Pt>1GeV

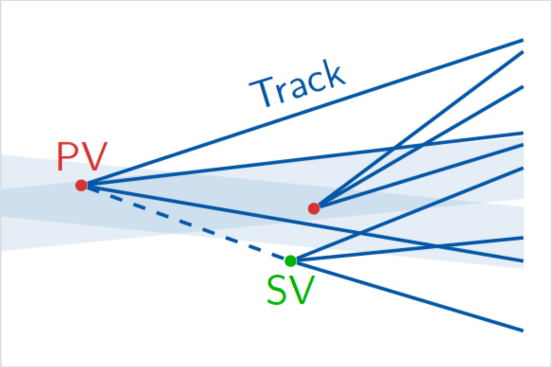


- ➢ Realistic simulation of the detector response to particles (Geant4)
- ➢ real trajectories reconstruction from the detector hit collection(ACTS)

PV vs SV

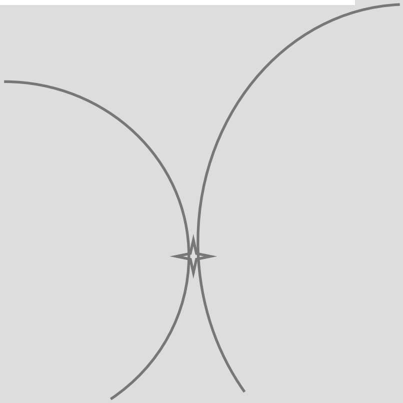
Primary Vertices(PV) – all beam-beam interaction points. 1D problem, doesn't require a curved particle trajectory calculation.





Secondary Vertices(SV) – interaction/decay points in 3D detector volume away from the beam line.
Can be detected by finding a crossing point (in fact a point of the closest approach due to resolution errors) of the curved particle trajectories in 3D space.

In general case the curved trajectory may be very complex due to non-uniform magnetic field.



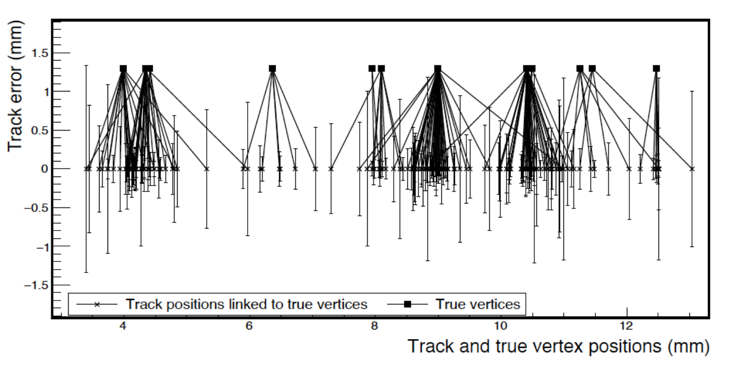
Important particle physics detector parameter – how far from the beamline secondary vertices should be to be detected? Crucial for short-living particle investigation.

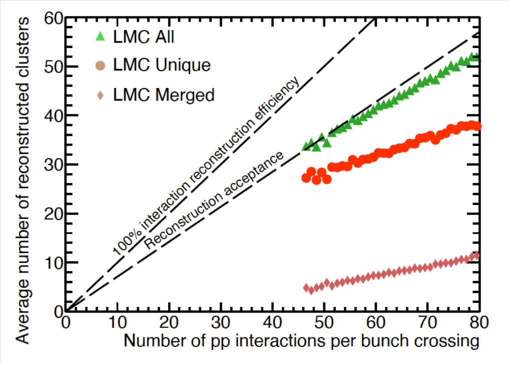
PV reconstruction (1D case)

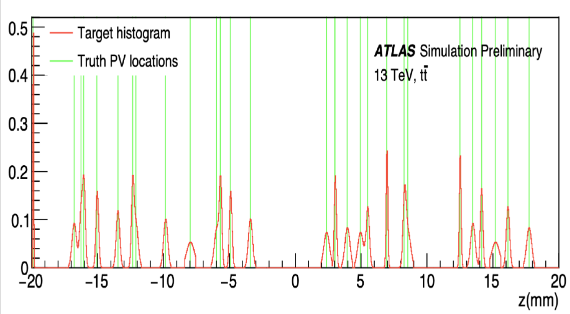
Many "classical" algorithms in the market (ATLAS,CMS,LHCb,etc. have their own)

ML – based approaches:

V. Kostyukhin, M.Keuper, M. Cristinziani *et al* 2023 *JINST* 18 P07013:
Track compatibility graph partitioning using Lifted Multi-Cut algorithm







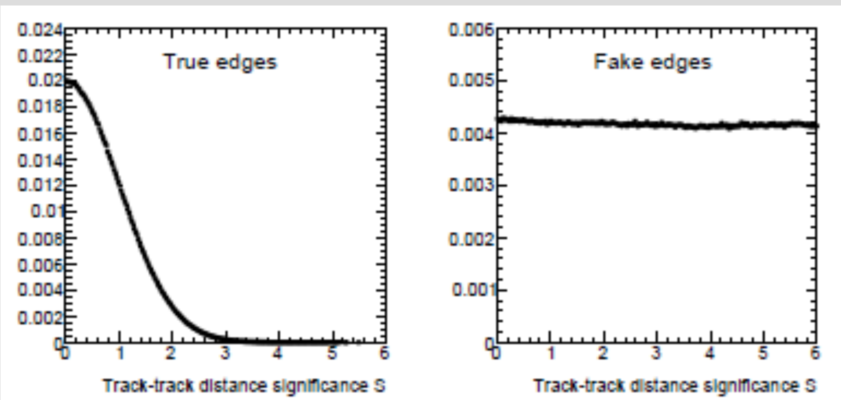
Software used in 2023 *JINST* 18 P07013 is revived and extended by an exact 3D vertexing for the SV reconstruction. Compatibility graph is written (ROOT format currently, needs conversion to smth. suitable for ML software. Clustering of this graph via (Lifted) Multi-Cut is implemented. With a simplest weight currently (next slides), but extension is trivial.

Edge score options (1D space)

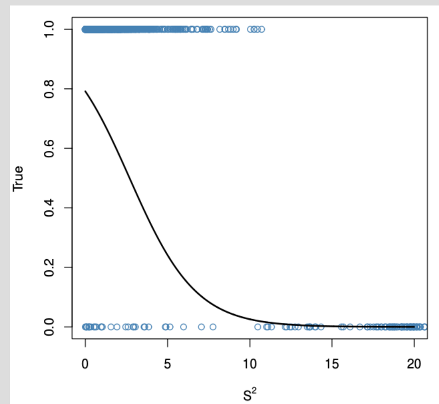
V. Kostyukhin, M.Keuper, M. Cristinziani *et al* 2023 *JINST* 18 P07013:
Track compatibility graph partitioning using Lifted Multi-Cut algorithm

1. Probability distribution ratio $w = \log(\frac{p_{true}}{p_{false}})$ of the minimal track-track distance significance $S = \sqrt{\frac{(x_i - x_j)^2}{\sigma_i^2 + \sigma_j^2}}$;
2. Logistic regression $p = \frac{1}{1+e^{-z}}$ where $z = \beta_0 + \beta_1 \cdot S$. Then $w = \log\left(\frac{1}{1-p}\right)$ weight has necessary features.
3. BDT edge classification (7 variables) score in [-1,1] range

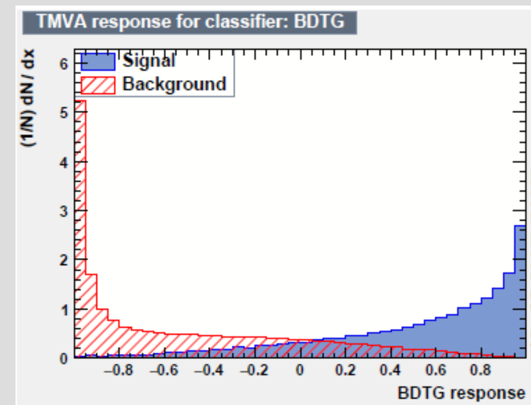
Track-track significance S



Logistic regression



BDT classification



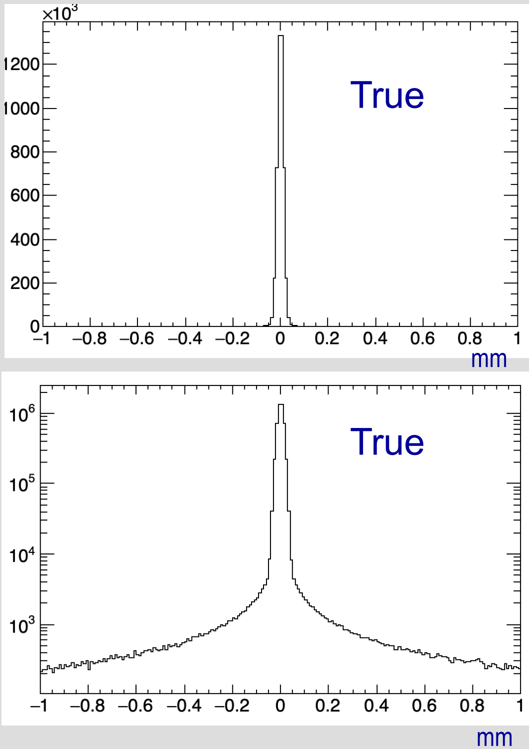
μ=250 pileup reconstruction results

Edge weight		VI	VI weighted	Silhouette	Silhouette weighted	Unique	Merged	Fake	$N_{\text{trk}}^{\text{wrong}}$	CPU
PDF ratio	base	1.782	0.990	0.477	0.526	68.7	53.2	6.4	42%	3.0s
	cnst	1.638	0.887	0.531	0.569	71.0	52.7	5.3	21%	1.7s
Regression	base	1.753	0.961	0.467	0.517	77.1	51.2	11.	38%	3.2s
	cnst	1.672	0.895	0.505	0.547	77.8	51.1	9.9	21%	1.7s
BDT	base	1.691	0.941	0.307	0.040	72.8	52.4	15.	12%	3.0s
	cnst	1.651	0.882	0.330	0.055	74.5	52.0	14.	9%	1.2s

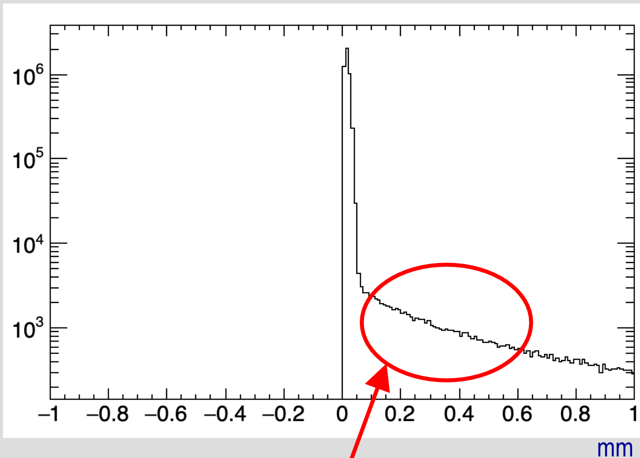
New code uses ODD+ACTS data. Paper results can be reproduced with more realistic data.

SV challenge

Track impact parameters wrt the beamline



True R of vertices



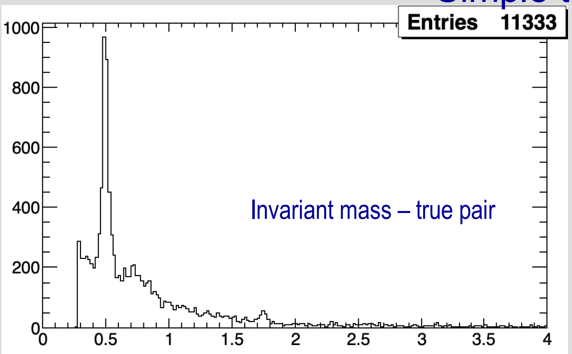
Problematic region for PV/SV reconstruction due to resolution (~equivalent to image blurring?). Hoped to resolve using ML.

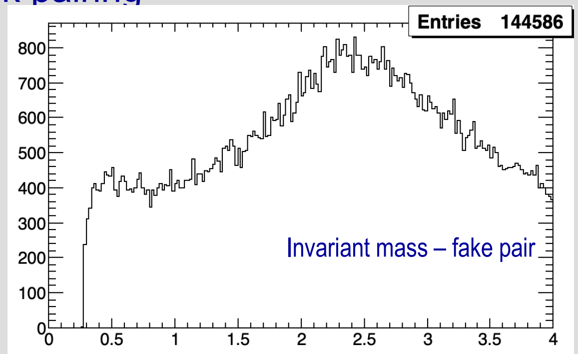
First attempt to run simultaneously PV(1D track-track significance) and SV(3D track-track significance) failed.   Completely crazy SV results due to strong influence of displaced prompt tracks.

SV challenge

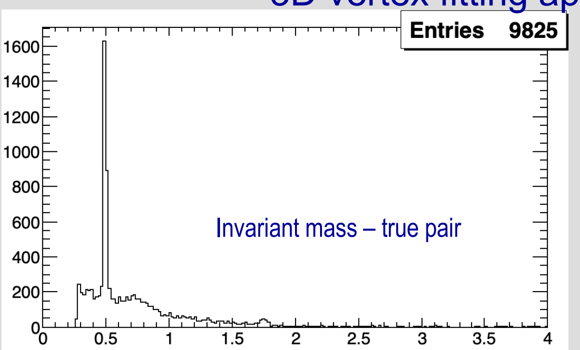
Use a limited SV-only setup – tracks >3σ from beamline

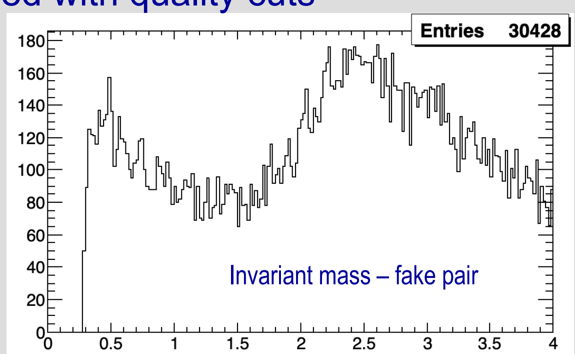
Simple track pairing

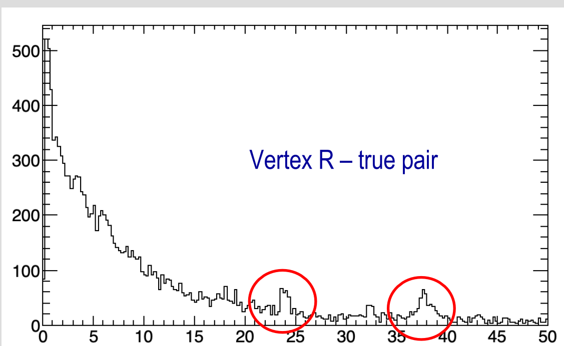


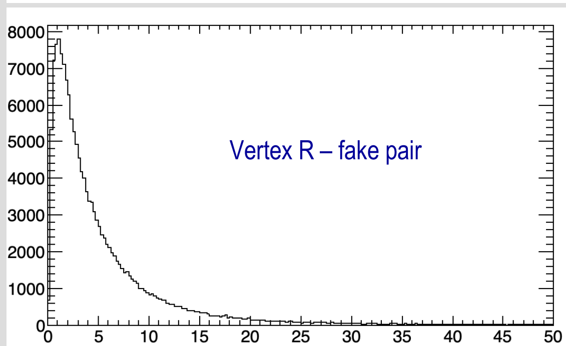


3D vertex fitting applied with quality cuts









3D SV vertex fitting works well:

- ✓ More narrow K0 mass peak
- ✓ Material interactions are detected

Vertex quality cuts reduce true statistics:

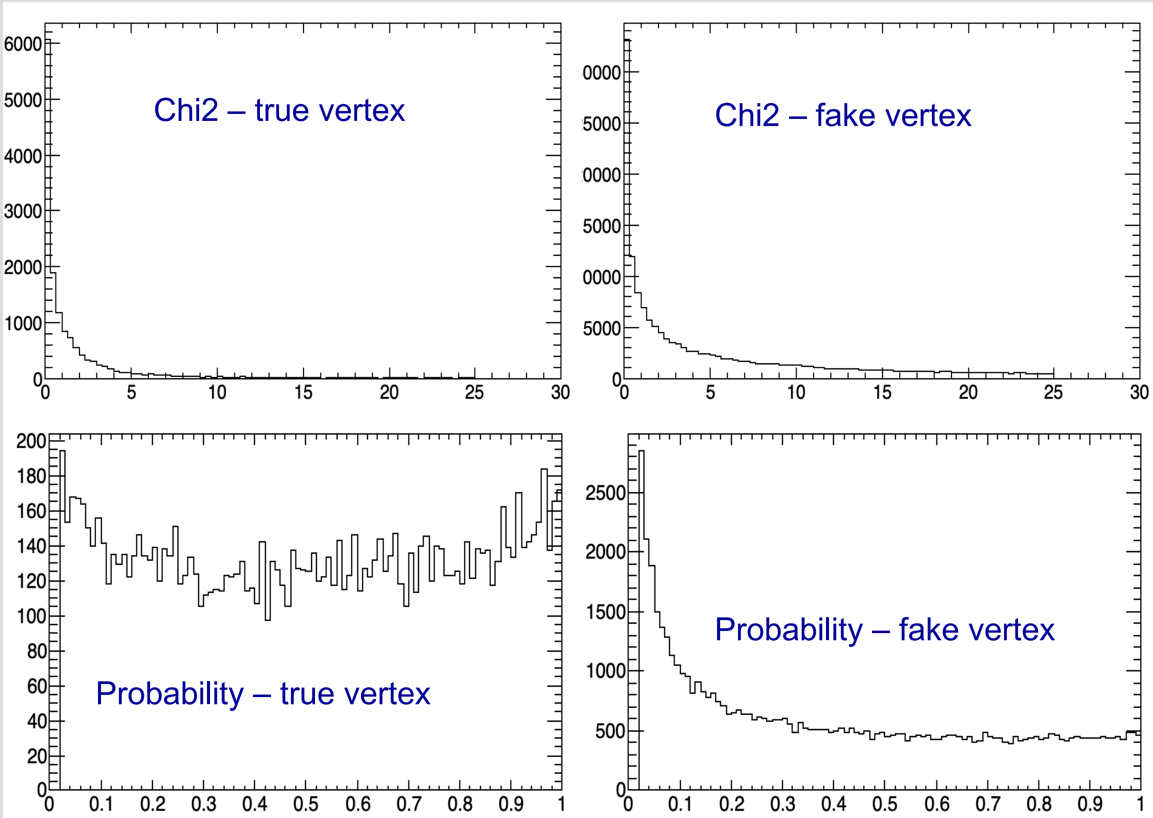
11333->9825

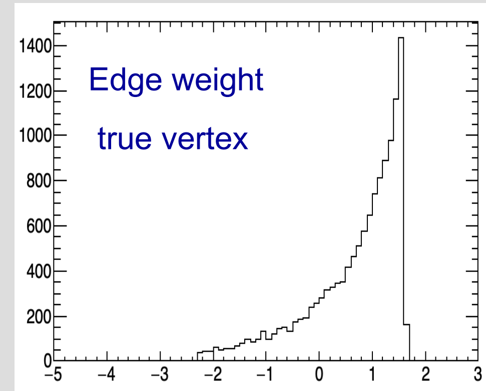
efficient rejection of fake track pairs:

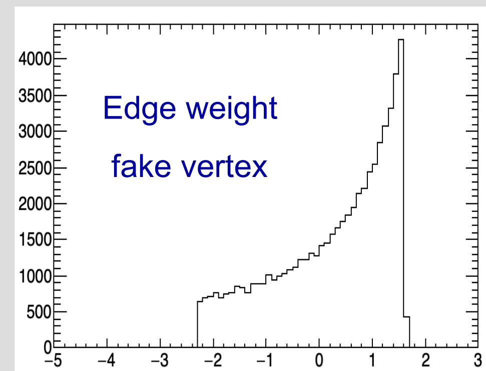
144586->30428

SV challenge

Use a limited SV-only setup – tracks >3σ from beamline







SV challenge

Universität Siegen

Use a limited SV-only setup – tracks >3σ from beamline – Multi-cut clustering

Vertex from cluster Chi2=286.723 R=0.52350 tracks=81,90,575
Vertex from cluster Chi2=0.519171 R=4.95209 tracks=327,593
Vertex from cluster Chi2=0.000203 R=10.1322 tracks=528,687
Vertex from cluster Chi2=0.356875 R=33.6912 tracks=602,609
True vertex R=5.31045 tracks=81,90

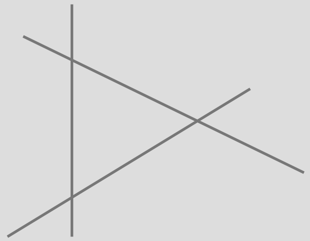
Vertex from cluster Chi2=4260.56 R=1.26509 tracks=76,320,391,435
Vertex from cluster Chi2=11929.1 R=0.65933 tracks=90,470,473,483
True vertex R=3.8789 tracks=84,85,90
True vertex R=15.7865 tracks=470,473,483

Vertex from cluster Chi2=416.186 R=1.61419 tracks=235,286,736
Vertex from cluster Chi2=0.33225 R=4.09057 tracks=254,679
Vertex from cluster Chi2=1.52774 R=18.7752 tracks=260,330
Vertex from cluster Chi2=0.00979 R=1.30765 tracks=262,298
Vertex from cluster Chi2=199.657 R=1.9585 tracks=326,482,488
Vertex from cluster Chi2=0.35785 R=0.71531 tracks=372,653
Vertex from cluster Chi2=0.00689 R=8.61447 tracks=402,404
Vertex from cluster Chi2=2.95114 R=5.0224 tracks=593,601,615,622
True vertex. R=29.7053 tracks=146,152
True vertex R=4.99748 tracks=601,615
True vertex. R=1.21047 tracks=262,298
True vertex. R=3.03938 tracks=235,286

Vertex from cluster Chi2=174.399 R=19.6805 tracks=49,86,171,172,207,497,520,697,
Vertex from cluster Chi2=1.40795 R=7.78051 tracks=59,106,
Vertex from cluster Chi2=7099.77 R=0.75759 tracks=98,110,111,118,120,223,
Vertex from cluster Chi2=1.37971 R=4.47632 tracks=430,436
True vertex=9.06084 tracks=98,118,120
True vertex=28.5759 tracks=171,172
True vertex=4.58186 tracks=430,436

Observations:

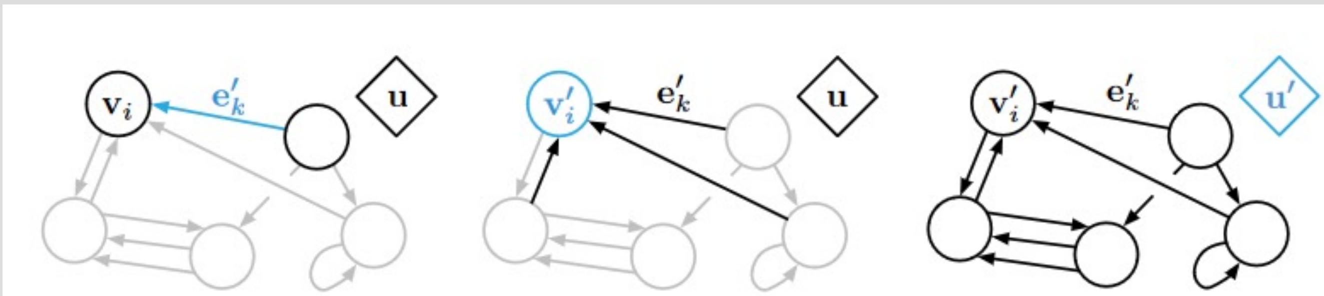
1. Many fake clusters. Expected (density+resolution)
2. Clusters with big χ^2 – pair-wise compatibility doesn't guarantee compactness in 3D space. Problem!
3. Some true vertices are not reconstructed. Puzzling, debugging needed.



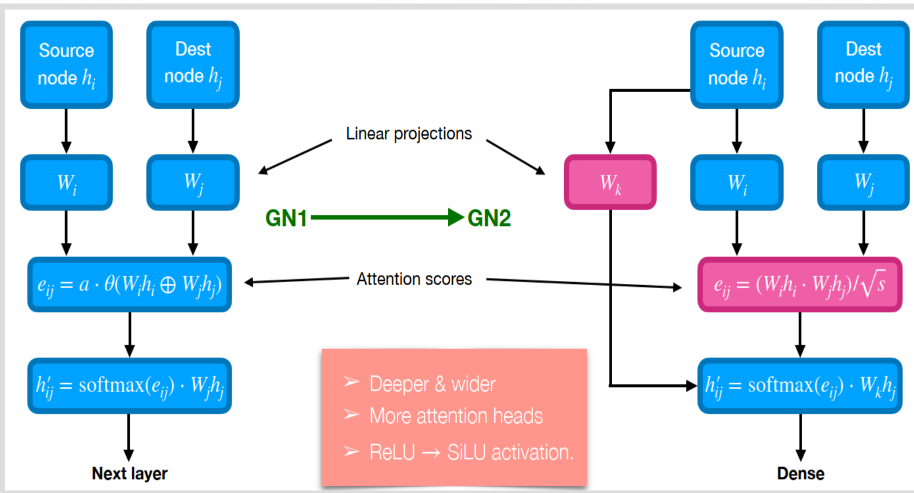
This is not a 3-track SV!

Plans

- Dump compatibility graph with node/edge features vectors (informative track/vertex features to be chosen…)
- Try simple edge classification BDT/NN for better edge weight construction
- Use GNN (with attention?) for the graph processing
- Simulate (ODD+ACTS) different physics sample better suitable for SV development



(a) Edge update (b) Node update (c) Global update



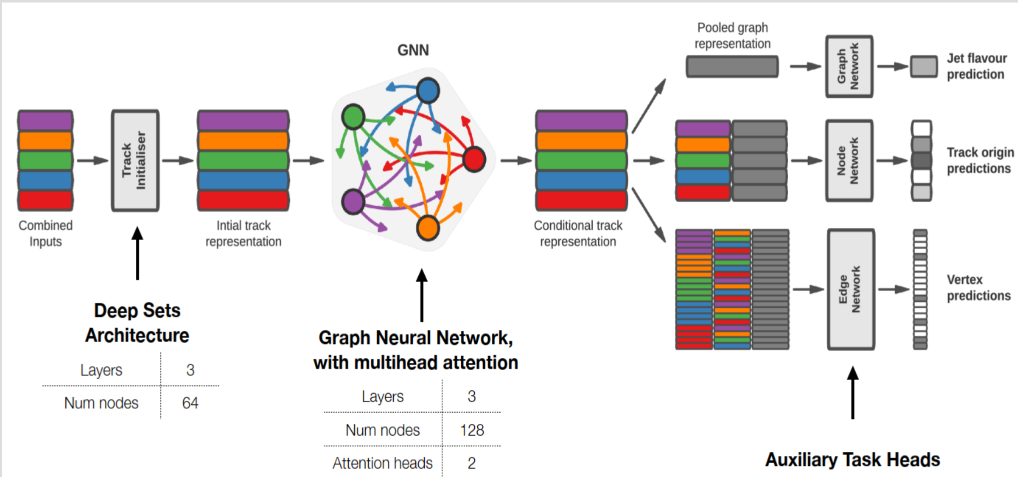
C++ problems

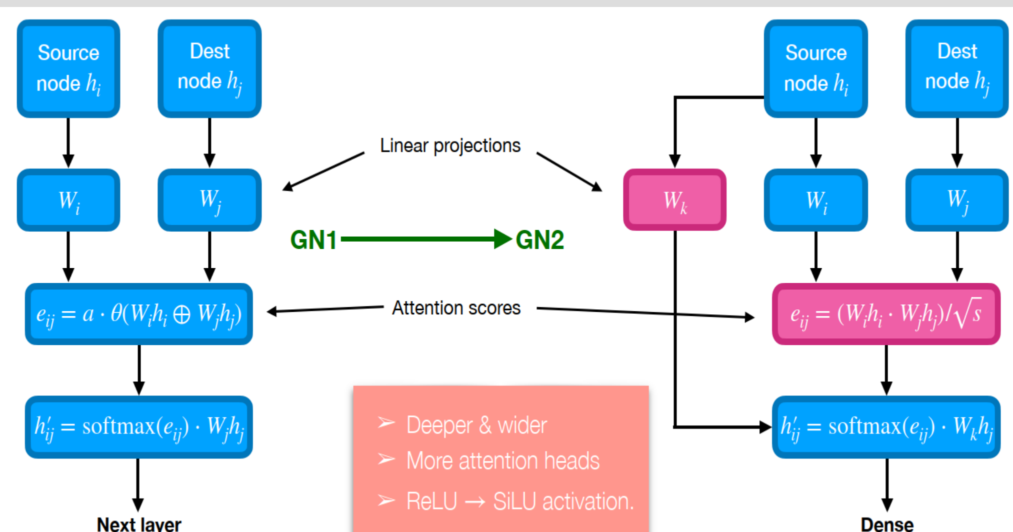
```
../complete-graph.hxx:25:21: warning: 'iterator<std::random_access_iterator_tag, const andres::graph::Adjacency<>>' is
deprecated [-Wdeprecated-declarations]
   25 |      :   public std::iterator <
```

```
/complete-graph.hxx:35:30: warning: 'Base' is deprecated [-Wdeprecated-declarations]
   35 |         typedef typename Base::iterator_category iterator_category;
      |                              ^
/Library/Developer/CommandLineTools/SDKs/MacOSX15.4.sdk/usr/include/c++/v1/__iterator/iterator.h:23:29: note:
'iterator<std::random_access_iterator_tag, const andres::graph::Adjacency<>>' has been explicitly marked deprecated here
   23 | struct _LIBCPP_TEMPLATE_VIS _LIBCPP_DEPRECATED_IN_CXX17 iterator {
```

ML SV reconstruction (in jets)

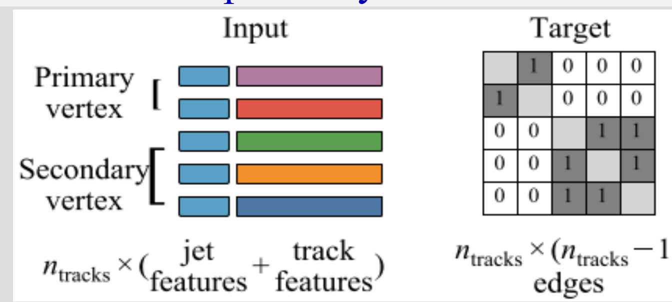
ATL-PHYS-PUB-2022-027 “Graph Neural Network Jet Flavour Tagging with the ATLAS Detector”





Origin papers:
arXiv:2002.08772 “Set2Graph: Learning Graphs From Sets”
arXiv:2008.02831 “Secondary Vertex Finding in Jets with Neural Networks”

In short:
Set of track parameters
→ track compatibility matrix

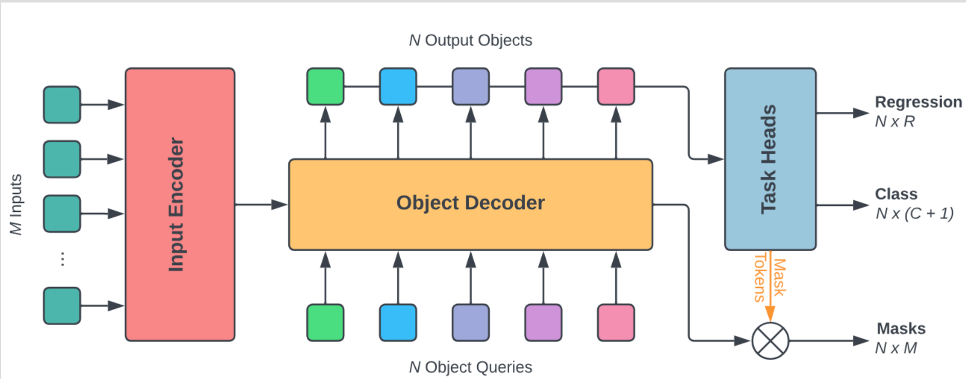


GNx software by itself returns just the track compatibility graph edge weights. The GNx algorithm uses a simple “union find” algorithm for the real graph/matrix partitioning.

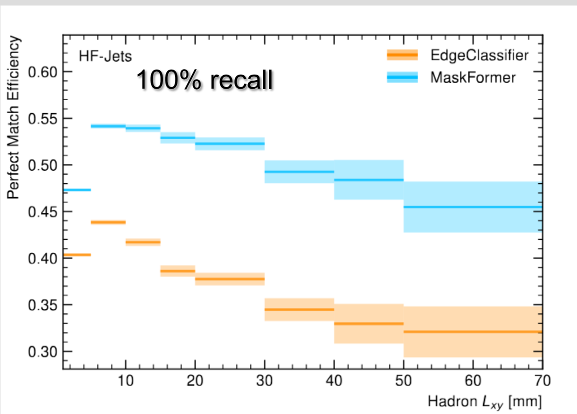
No real vertexing, no vertex quality check. From note:
“A vertex is considered matched if it contains at least 65% of the tracks in the corresponding truth vertex and has a purity of at least 50%. “

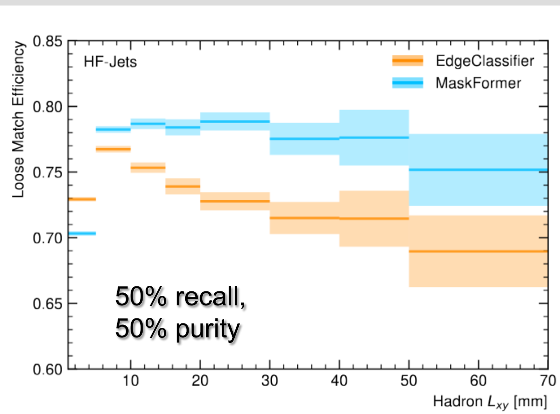
SV reconstruction (in jets)

arXiv:2312.12272 “Secondary Vertex Reconstruction with MaskFormers”



1. Vertex features reconstruction via regression
2. Track to vertex assignment via Masks (explicit partitioning)





“Edge classifier” is the algorithm from the previous page

1. No universality – NN is trained on B/C vertices. No extension to many other SV types, especially for exotic vertices with unknown properties
2. Efficiency/purity is far from ideal.

Backup: Real event in the ATLAS detector