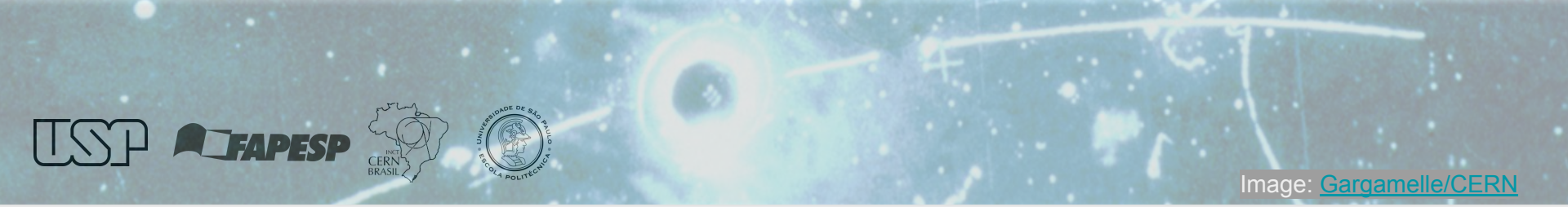


Hypergraph Neural Network 4D track reconstruction pipeline for HL-LHC experiments

Rodrigo Estevam de Paula
Marco Aurelio Lisboa Leite
Vitor Heloiz Nascimento

Universidade de São Paulo (BR)

April 2, 2026

High-Luminosity LHC (HL-LHC)

- LHC will undergo a series of upgrades to extend its discovery potential
  - Raise the rate of collisions per bunch crossing (pileup) by a factor of five
  - Reach higher statistics on known phenomena (e.g. Higgs boson) and observe rare new phenomena.

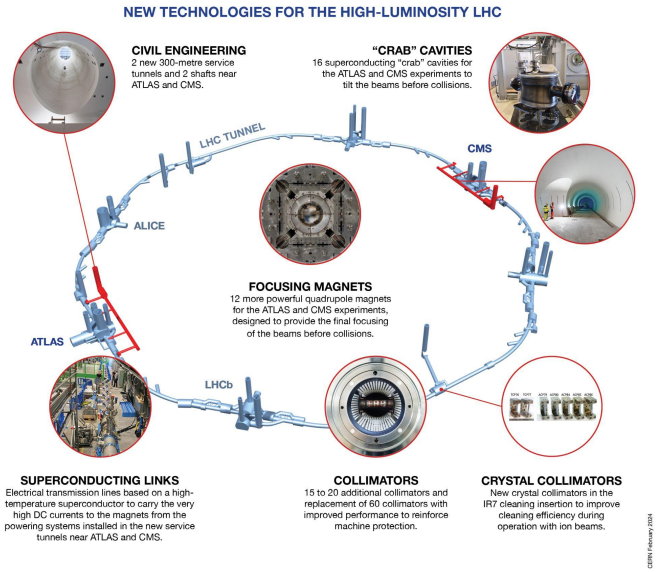
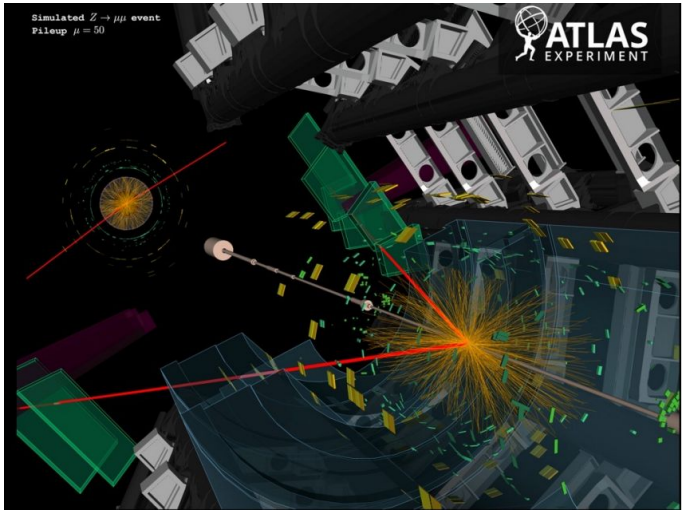


Image: HL-LHC/CERN



Simulation of Z →μμ at the ATLAS experiment with increasing pileup (ATLAS event display UNSG-2021-29)

High-Luminosity LHC (HL-LHC)

- LHC will undergo a series of upgrades to extend its discovery potential
  - Raise the rate of collisions per bunch crossing (pileup) by a factor of five
  - Reach higher statistics on known phenomena (e.g. Higgs boson) and observe rare new phenomena.
- To benefit from HL-LHC, experiments will need to upgrade their detectors
  - Remodel event reconstruction algorithms to handle the increase in data volume

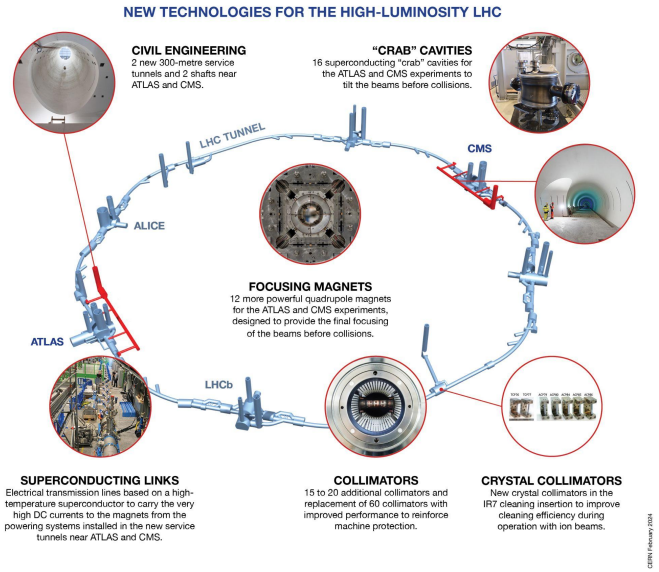
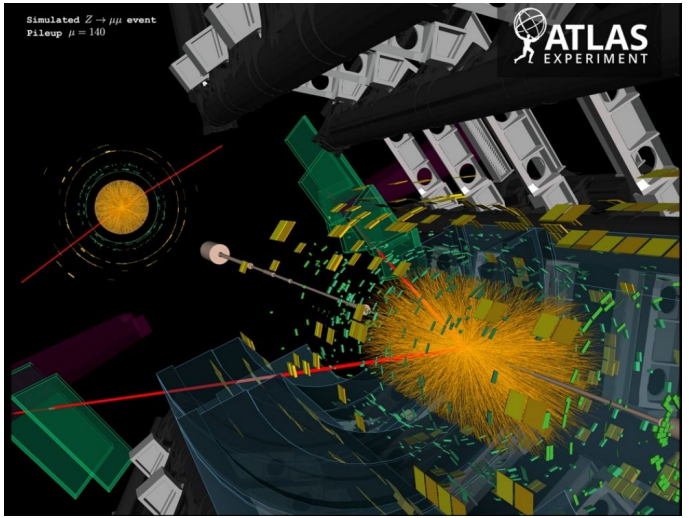


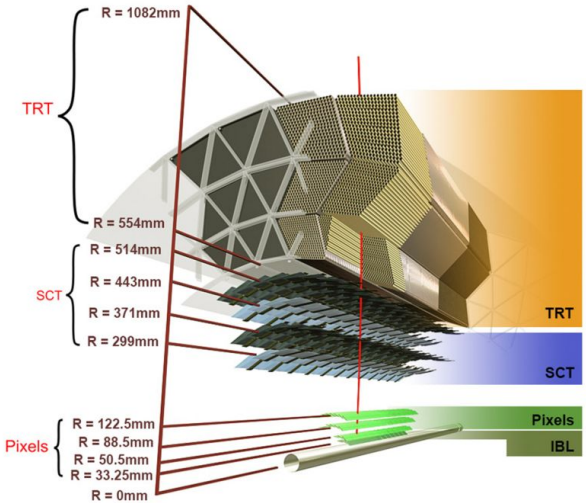
Image: HL-LHC/CERN

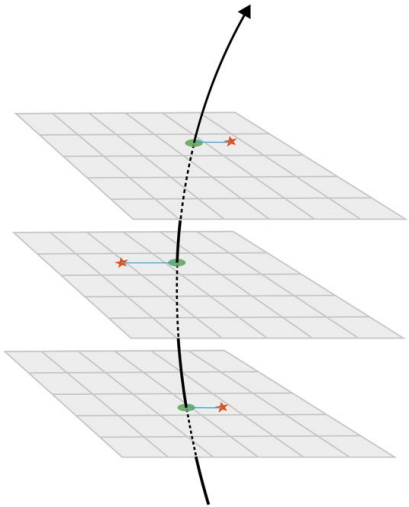


Simulation of Z →μμ at the ATLAS experiment with increasing pileup (ATLAS event display UNSG-2021-29)

Track reconstruction

- Charged particle tracking happens at the innermost region of the detector and is essential to reconstruct the kinematic properties of the events
- The *Inner Detector* (or the future *Inner Tracker*) is constructed of semiconductor sensors that are sensible to passage of charged particles
- The goal of the reconstruction algorithm is to “connect the dots”: find the particle trajectories that generated the event hit distribution





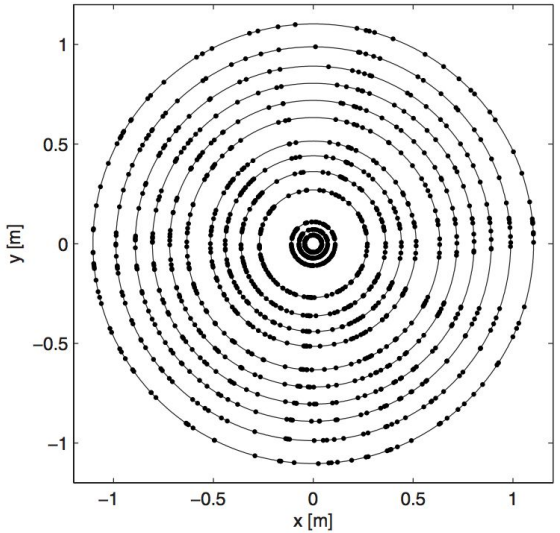
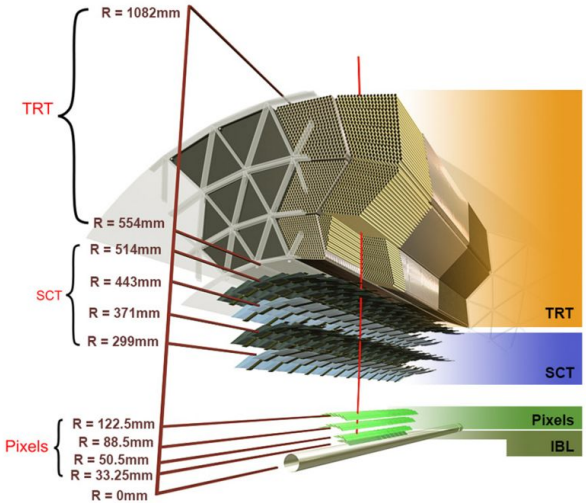


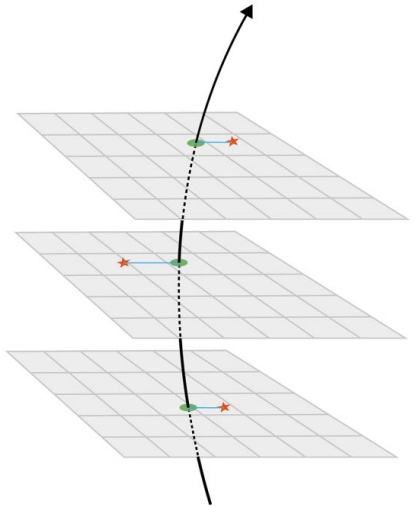
Image: ATLAS Collaboration (CDS link)

A. Strandlie et al. "Track and vertex reconstruction: From classical to adaptive methods"

Track reconstruction

- Charged particle tracking happens at the innermost region of the detector and is essential to reconstruct the kinematic properties of the events
- The *Inner Detector* (or the future *Inner Tracker*) is constructed of semiconductor sensors that are sensible to passage of charged particles
- The goal of the reconstruction algorithm is to “connect the dots”: find the particle trajectories that generated the event hit distribution





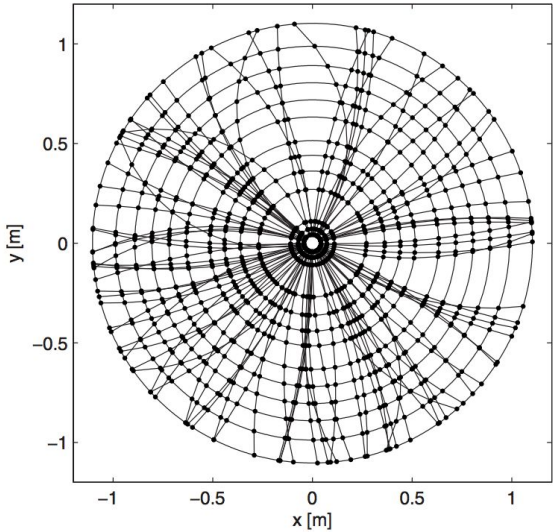
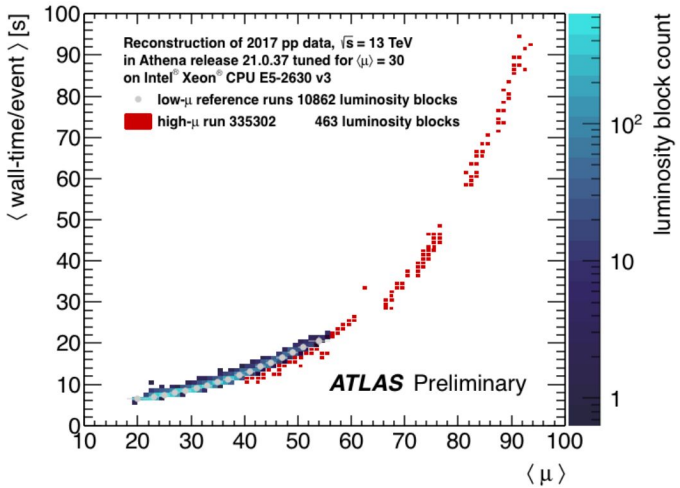


Image: ATLAS Collaboration (CDS link)

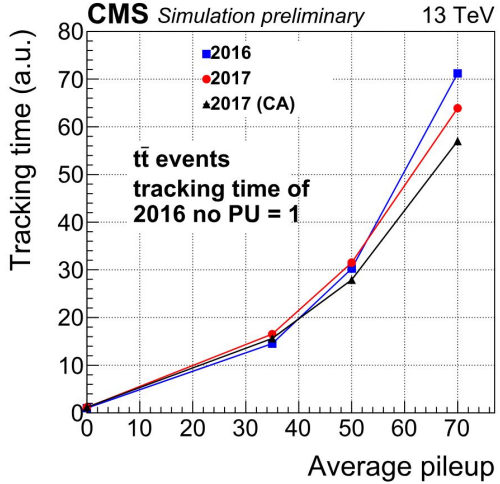
A. Strandlie et al. "Track and vertex reconstruction: From classical to adaptive methods"

Charged particle tracking at the HL-LHC

- Years ago, we realised that this pipeline would not have enough throughput to operate on the HL phase
- Plots show the dependency of the processing time and the pileup (will be $\langle \mu \rangle = 200$ at HL-LHC)
- The processing time of CKF grows quadratically with the number of hits, so it's a major bottleneck
  - Because of its recursive nature, it is also challenging to parallelize
- Need to devise alternatives that perform this task with reasonable run times



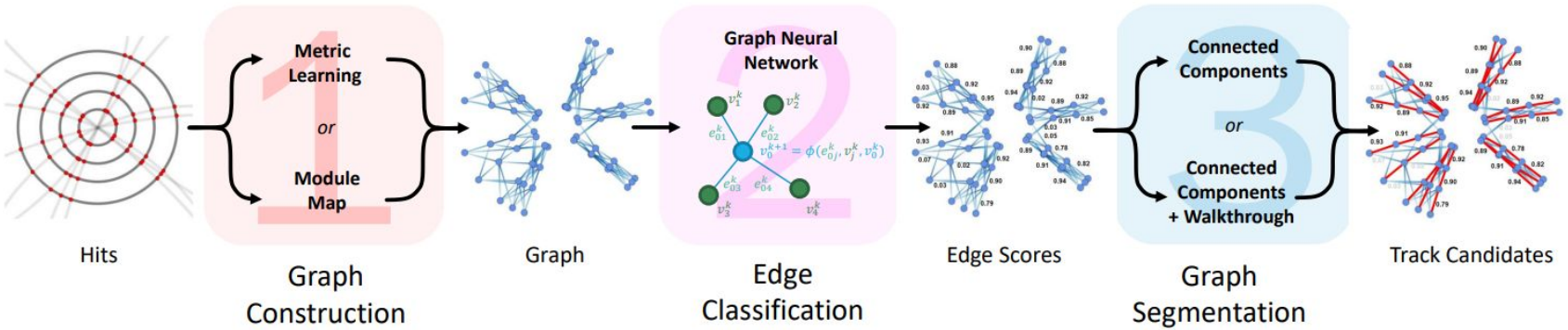
ATLAS Computing and Software - Public Results (Twiki link)



Wall time of the track reconstruction pipeline at the CMS experiment in function of increasing pileup (CMS Collaboration ,2017)

Charged particle tracking at the HL-LHC

- Promising pipelines utilize Graph Neural Networks (GNNs) as a high-throughput alternative
  - Input graph represent hits as nodes and track segments as edges
  - The GNN then estimates the likelihood of the edge being part of a particle track
  - Selected track segments are then grouped into particle tracks
  - Highly parallelizable! Significantly decreasing runtime when processing high number of events

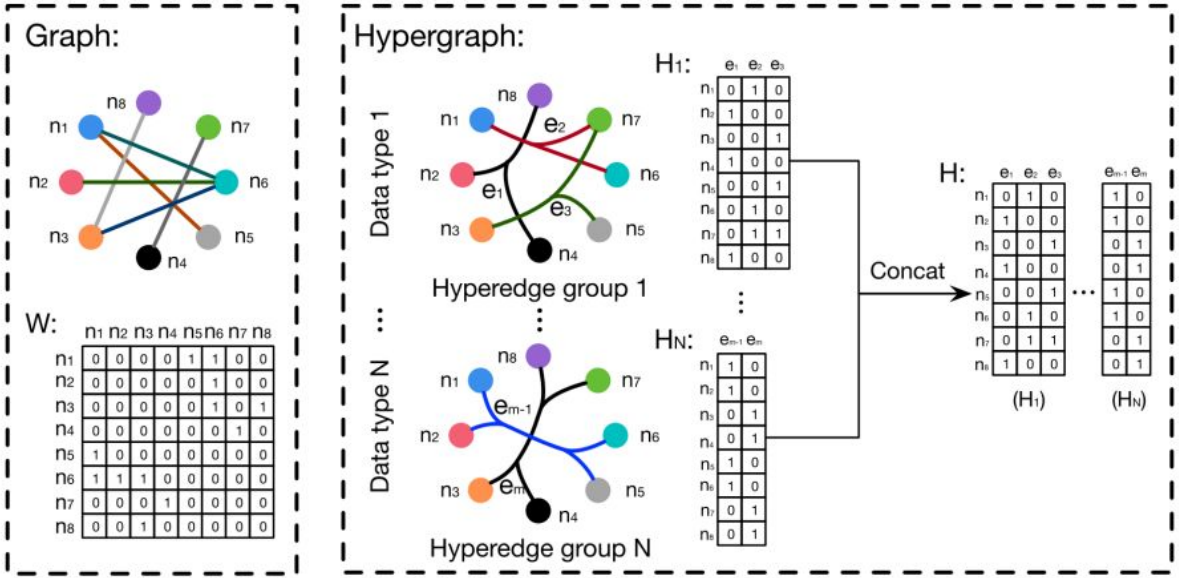


Daniel Murnane (2022) Graph Neural Networs for High Luminosity Track Reconstruction.EP-IT Data Science Seminar, CERN

HyperGNN particle tracking pipeline
for HL-LHC experiments

(Why) Hypergraphs?

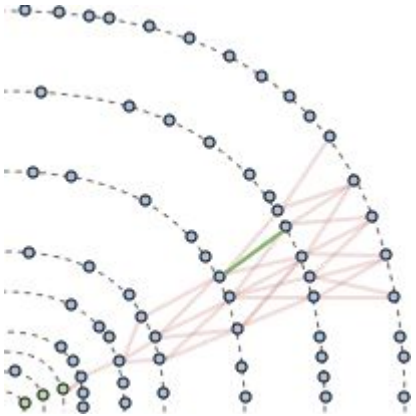
- Hypergraphs are graphs with edges that connect more than two nodes
  - Better represent node relationships that go beyond pairwise connections
- The generalization and stability theorems formulated to GNNs can be extended to HGNNs
  - Possible to do spectral analysis of “filter response”

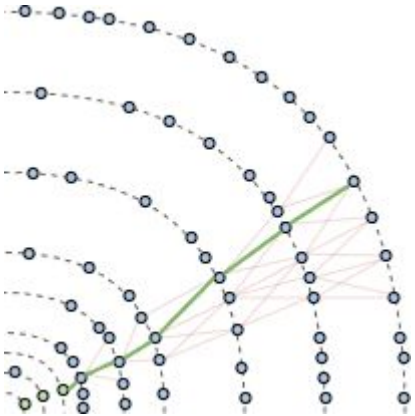


Yifan Feng et. al., Hypergraph Neural Networks

(Why) Hypergraphs in tracking?

- Intuitively, is easier to predict if a whole set of nodes compose a track than if two nodes compose a segment
  - Supposing we model it well with our feature space
- This representation would allow for a two-step pipeline, instead of three, as the track building would not be necessary
  - Only graph building and network inference

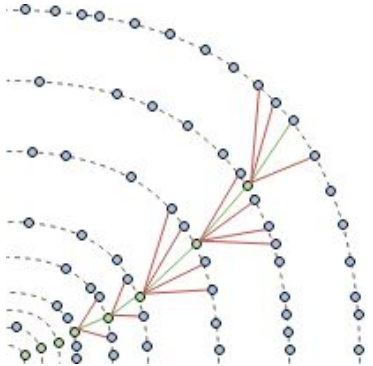




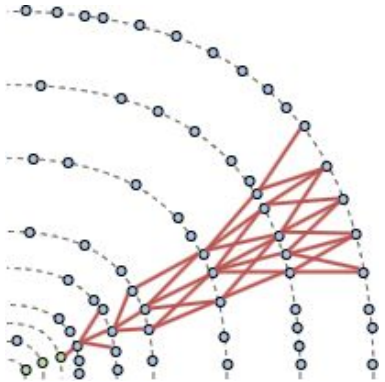
HyperGNN pipeline implementation

Hypergraph building - using simplified CKF

- For now, we use a simplified version of CKF for the hypergraph building
- It performs a multi-path extrapolation, without a measurement selection
  - Each possible path is a different hyperedge
  - We also removed the kalman update and material interaction simulation
  - Kept high efficiency (98.6%) but with low purity
- The network should filter fake tracks, so high efficiency is required but purity is not
- On a second iteration our method was 2x faster than CKF (from 8.0 s to 3.18 s) but with a bigger impact on efficiency
  - Efficiency dropped to 93.2%
  - We still need to find the best working point



Combinatorial Kalman Filter (CKF)



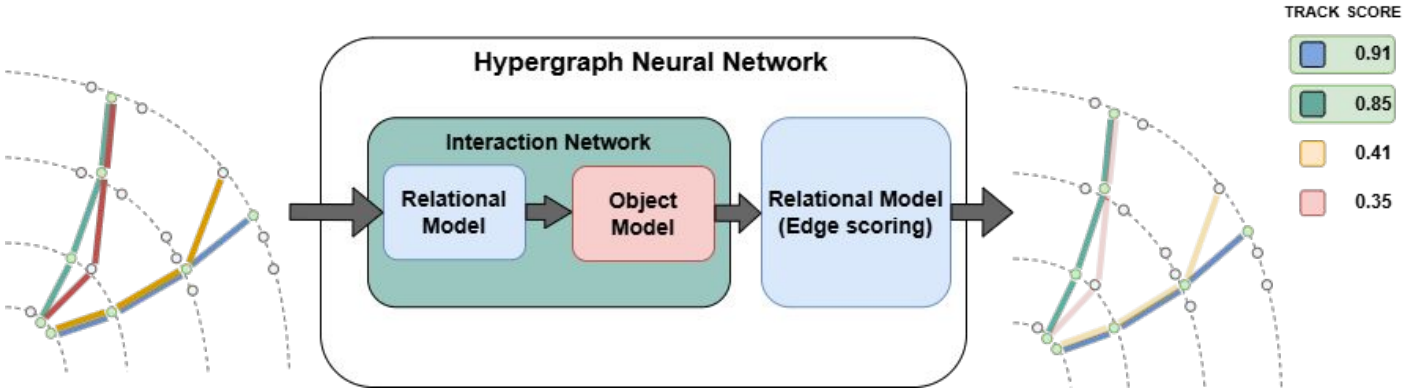
Hypergraph building algorithm

HyperGNN inference

- For the feature space, we chose attributes easily accessible from the track finding algorithm
  - will expand in the future

$$n = \{x, y, z, t, r, \phi, \theta\} \qquad e = \{\mathrm{nMeasurements}, \mathrm{nOutliers}, \mathrm{nHoles}\}$$

- We devised an HyperGNN architecture inspired by the Interaction Network (IN) design
  - Relational model ⇨ hyperedge-wise (track candidates) convolution
  - Object model ⇨ node-wise (hits) convolution
  - We chose the maximum function as an aggregator and edge-wise attention mechanism
- Implemented with PyTorch and the conv.HypergraphConv layer



HyperGNN tracking pipeline

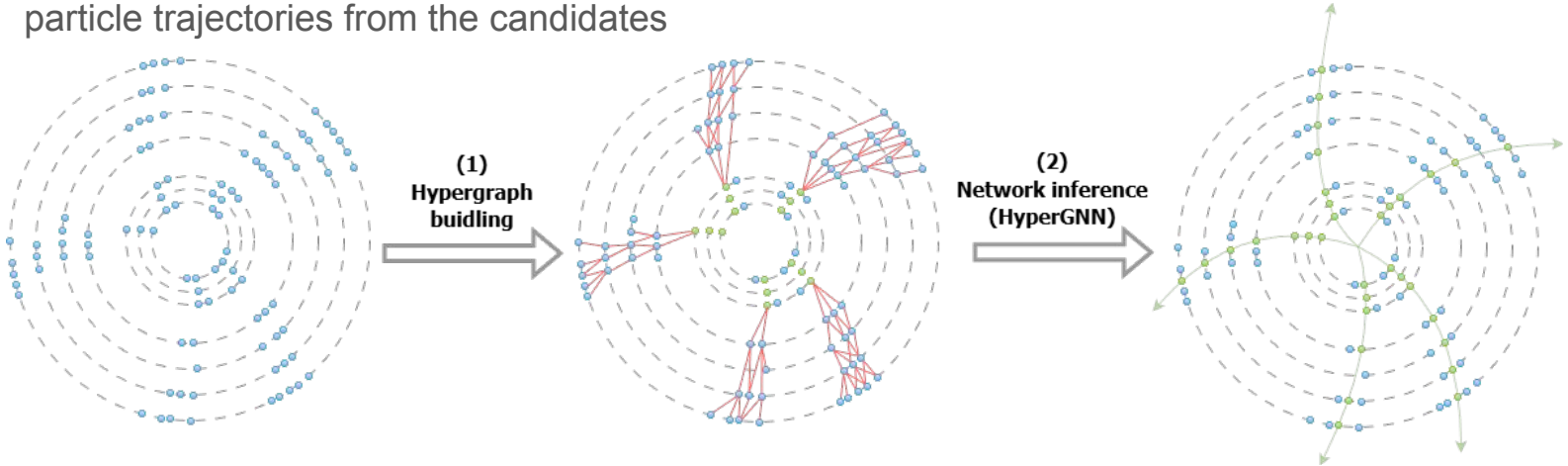
- Modelling the event data as a hypergraph, we would have:
  - nodes: measurements (particle hits)
  - hyperedges: collection of measurements that compose a track candidate

Hypergraph building procedure (1)

- We devised a new track finding algorithm based on the CKF to build the hypergraph
  - No measurement selection, resulting in a multi-path extrapolation
  - High rate of ‘fake tracks’ is affordable as the network will be responsible for filtering it out

Network inference (2)

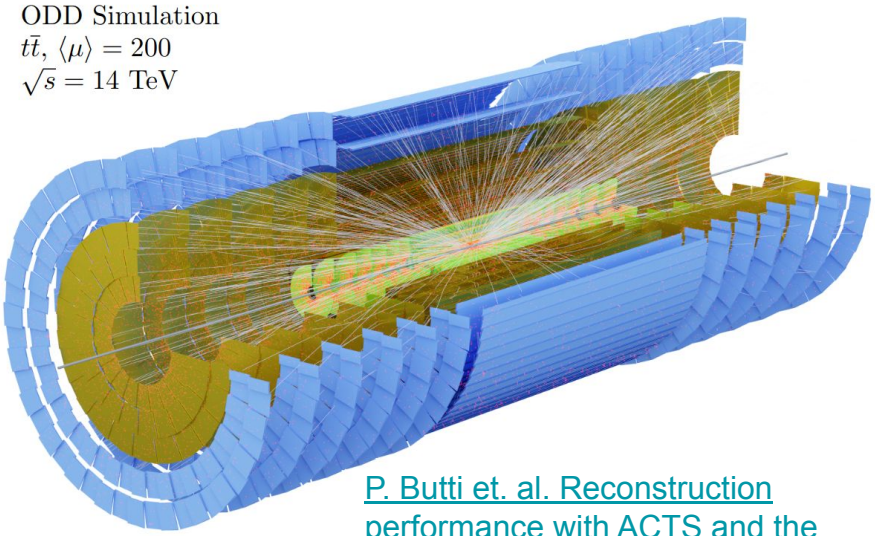
- We implemented an HyperGNN architecture inspired by the Interaction Network (IN) design to select particle trajectories from the candidates



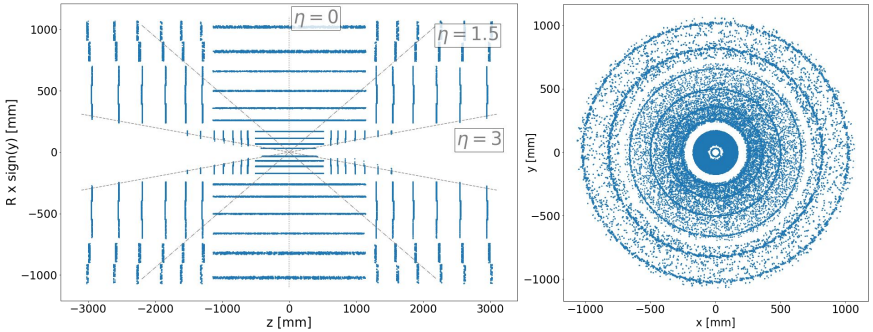
HyperGNN pipeline evaluation

HyperGNN pipeline evaluation setup

- Using the A Common Tracking Software (ACTS) framework for event simulation and evaluation of the track reconstruction
  - Simulated pp → t-tbar and ⟨μ⟩ = 200 (pileup) on the Open Data Detector (ODD)
  - Using FATRAS for detector response
  - pT > 1 GeV and |η| < 3
  - training on 450 events, testing on 50

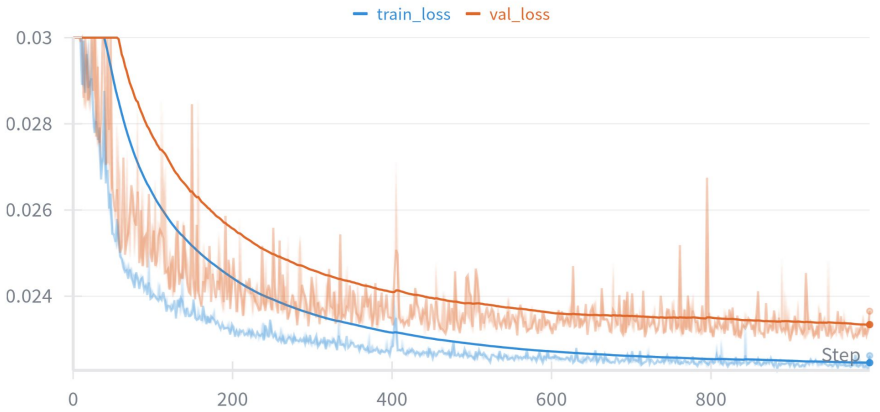


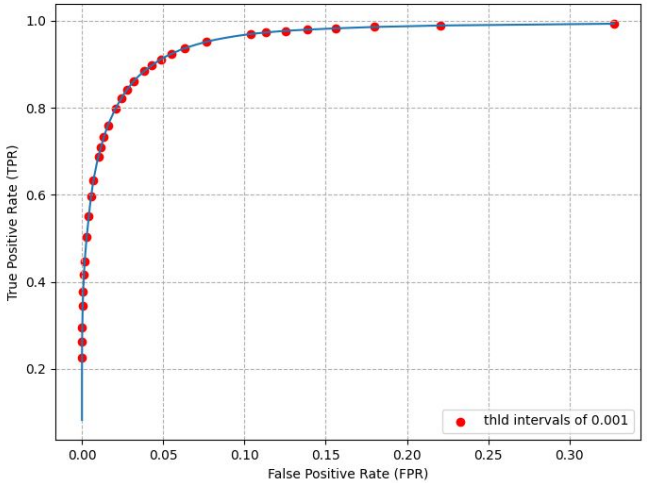
P. Butti et. al. Reconstruction performance with ACTS and the Open Data Detector



Pipeline evaluation - HyperGNN training

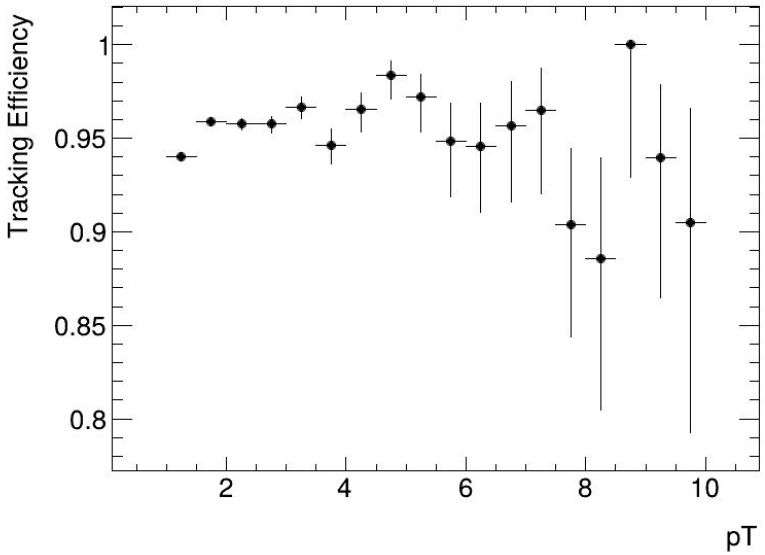
- We use 50 of the training events for validation and threshold optimization
  - Chose a threshold that keeps the FPR bellow 10%
- Different models sizes were tested, varying number of nodes on the hidden layers
  - Best performing models occupy ~4 GB of RAM during training
- Still need to improve the training procedure and tune the hyperparameters
  - Find best learning rate and weight decay

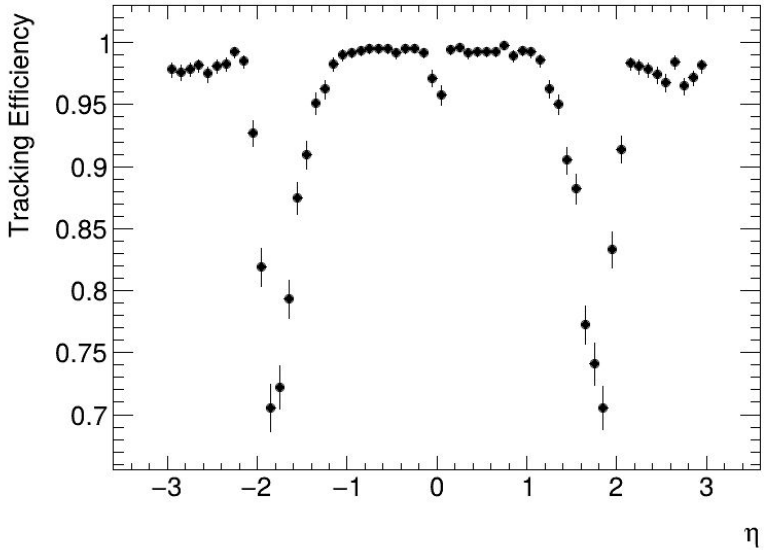




HyperGNN pipeline evaluation

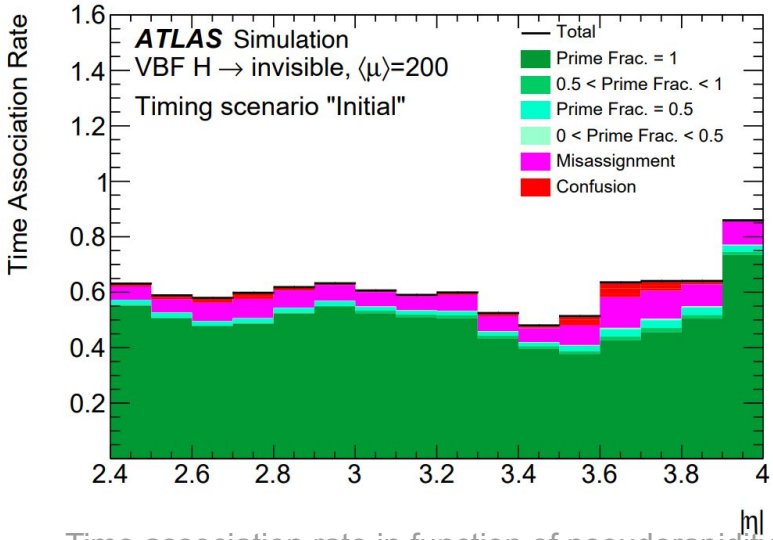
- The pipeline achieved a global efficiency of 95%
  - 98.6% on the graph building stage
  - 95% on the network selection
- Plots below show efficiency of the network selection stage
  - Need to integrate output back to ACTS in order to generate global efficiency plots





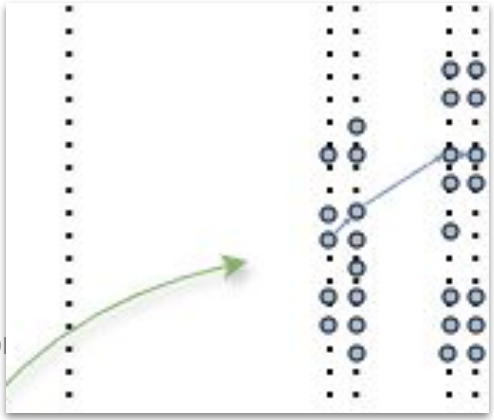
4D tracking pipeline with hypergraphs

- We plan to adapt this pipeline to perform 4D tracking with a timing layer (e.g. ATLAS HGTD)
- The main challenge of 4D tracking is the association of tracks with hits on the timing layer
  - Misassignaments significantly degrade the track time association and primary vertex time (t_0) reconstruction

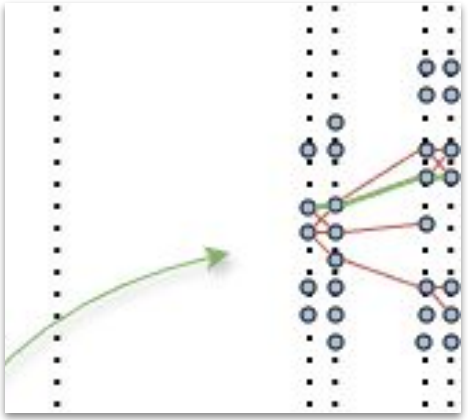


Time association rate in function of pseudorapidity o
the ATLAS experiment with HGTD
ATLAS HGTD TDR (ATLAS-TDR-031)

- An extrapolation with time information and that takes the whole path into account may enhance the association performance



Extrapolation with CKF like algorithm



Extrapolation with hyperGNN

Summary and Next steps

Summary

- We proposed a HyperGNN pipeline that shows high reconstruction efficiency on simulated HL-LHC events
  - IEEE NSS conference (proceedings) and Connecting the Dots workshop (proceedings in review)

Next Steps

- Optimize graph building algorithm for runtime
  - Testing the substitution of the numerical integration extrapolation by a parameterized procedure
  - Adapt the code for parallel processing on GPUs
- Improve network design
  - Find the workpoint that best ballance efficiency, memory usage and throughput
- Integrate endcap timing layers for 4D tracking (e.g. ATLAS HGTD)
  - Adapt our model to make best usage of the region limited time measurements
- Full comparison analysis with current GNN approaches

Contributions to HGTD reconstruction in Athena

Changing HGTD reconstruction to use ACTS

- My current contribution task at the HGTD community is to finish the ACTS version of the track-time association algorithm
  - The table below shows structural differences from legacy to ACTS
- A significant increase in efficiency is not expected, but we look forward to reducing the rates of confusion and misassignments

	Legacy	ACTS CKF
Branching: when reaching each layer, extrapolation can branch out by “picking” different hits around the extrapolation point	just follows one path, picking the closest hit from the extrapolation point	customizable, can be set to follow N nearest hits.
Time propagation: is time also extrapolated to next layers and included in the Kalman filtering?	All propagation is only geometric	propagates time and check for time consistency at extrapolation level
χ^2 evaluation: what is used to calculate the score of the extension (ones with high chi2 are cut)	only geometric information	can use the time coordinate together with geometric

Progress up to this point

Inclusions done on the ACTS CKF association algorithm:

- Kalman update: now estimates for next layers can be based on the filtered state, not predicted
- Measurement selector: without this tool the extrapolation branches out of control, returning hundreds of extension per track
  - The numMeasurementsCutOff defines how many closest hits will be picked
- Track Selector: I made a simple one to select the extension with lowest χ^2 for each track particle.
  - Also with a threshold to filter misassignments
- Time propagation: Time wasn't being propagated in the correct unit

Truth Matching using ACTS tools:

- Asserting correct associations and confusion: done by comparing the truth particles from the ITk track and the ones that originated the HGTD hits from its extension. Confusion occurs when there's a mismatch between the two.
- Asserting misassignments: for each layer, evaluate if hits from the track are expected there. If not and the association algorithm picks one anyway, we have a case of misassignment
- These decorations allow using HGTD Analysis package to generate performance plots
- More details on the implementation is present at the backup

Next steps

- Start working on separating truth matching from extension
  - Output the trackProxy of the extensions (and the trackParticle it's mapped to) and access it at the truth matching algorithm.
- Migrate HGTD performance tools to Athena, making it easier and faster to evaluate performance.
  - Still need to finish truth matching algorithm by including flagging of shadowing when more than one particle contributes on the cluster
  - Could also be used for the one-pass reconstruction (ITk track reconstruction including HGTD layers)
- Finish CKF extension implementation
  - Migrate the class to inherit from *TrackFindingBase*
  - Make the implemented track selector work
  - Include a branch stopper (combination of measurement selector and track selector)

Thank you for your attention!

Questions?

rodrigoestevam99@usp.br

References

[1] ATLAS Collaboration. *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC.* Physics Letters B,2021. https://doi.org/10.1016/j.physletb.2012.08.020.

[2] M. Gullstrand, and S.Maraš. “Using Graph Neural Networks for Track Classification and Time Determination of Primary Vertices in the ATLAS Experiment” (Dissertation). Disponível em http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-288505

[3] JU, X. et al. *Performance of a geometric deep learning pipeline for HL-LHC particle tracking*. The European Physical Journal C, v. 81, n. 10, p. 876, out. 2021. ISSN 1434-6052. DOI: 10.1140/epjc/s10052- 021- 09675- 8. Disponivel em https://doi.org/10.1140/epjc/s10052-021-09675-8 .

[4] DeZoort, G., et al. "Charged Particle Tracking via Edge-Classifying Interaction Networks," in Computing and Software for Big Science, vol. 5, no. 1, 2021. Available at: https://arxiv.org/abs/2103.16701

[5] ATLAS Collaboration, “Technical Design Report: A High-Granularity Timing Detector for the ATLAS Phase-II Upgrade”. Technical report, CERN, Geneva, 2020. Disponível em: https://cds.cern.ch/record/2719855/files/ATLAS-TDR-031.pdf

[6] Santi, L., “4D Tracking at ATLAS” in HSF Seminar - 4D reconstruction. Available at: https://indi.to/YsmvK

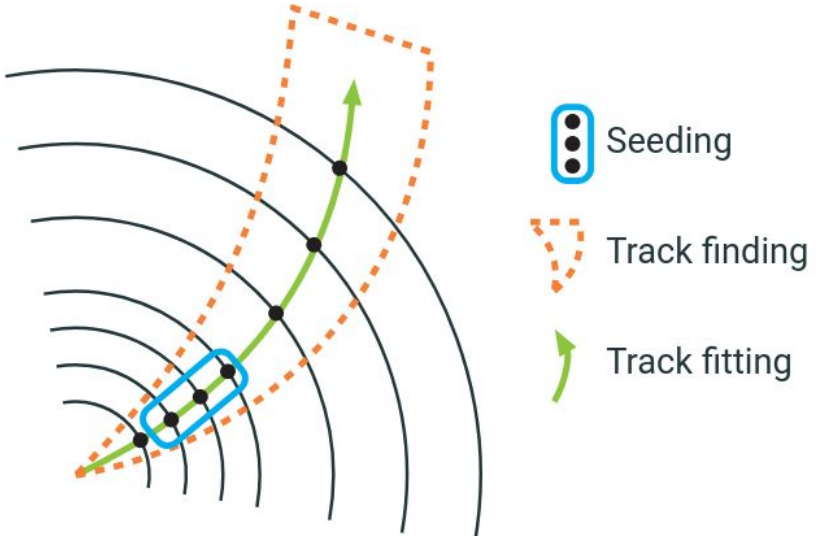
[7] ATLAS Collaboration. *ACTS documentation.*
Disponivel em: https://acts.readthedocs.io/en/latest/index.html

Backup

Track reconstruction chain

The typical reconstruction pipeline has three stages:

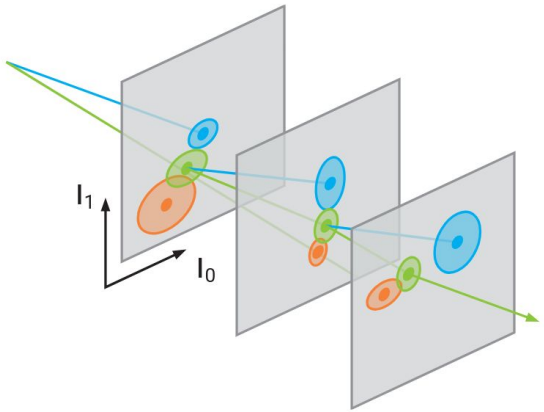
- **Seeding:** Find triplets in the innermost layers that most likely correspond to the beginning of a track particle
- **Track finding:** Extrapolate the seed forwards and finds the collection of hits that compose a track
- **Track fitting:** Fit an helicoidal trajectory into the hit collection to estimate the track kinematic properties

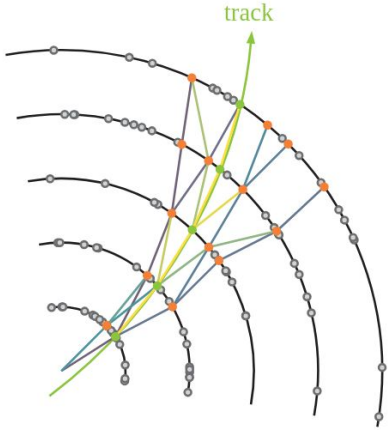


ACTS documentation: Tracking in a nutshell

Track reconstruction chain: Track finding

- If there wasn't any source of errors, the particle would travel in a perfect helicoidal trajectory
- But in practice is necessary to account for deviations in the trajectory
- We employ a Kalman Filter on top of the extrapolator to adjust for these deviation effects
  1) Extrapolate the last hit of the track to the next layer (blue)
  2) Adjust the kalman filter based on the distance between your estimate and the nearest hit (orange)
  3) Apply the filter to correct the trajectory (green)
- The algorithm stops when there are no measurements nearby to perform Step 2)
- In reality Step 2) is performed for then N nearest hits, and the algorithm can branch out to multiple paths -> Combinatorial Kalman Filter (CKF)
  - In this case is also necessary to keep a quality factor to select the best tracks within the branches

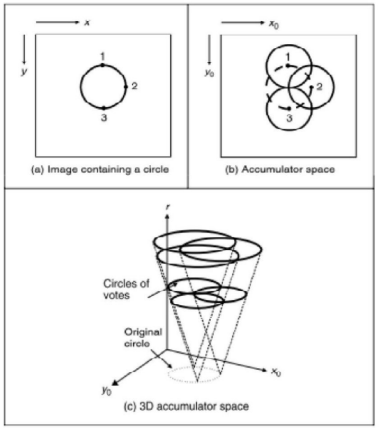


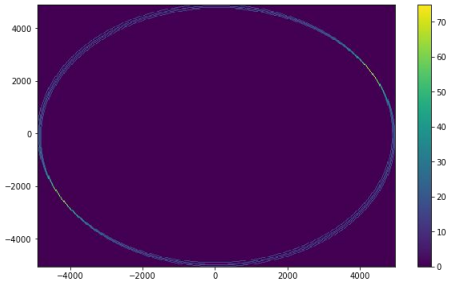


ACTS documentation: Tracking in a nutshell

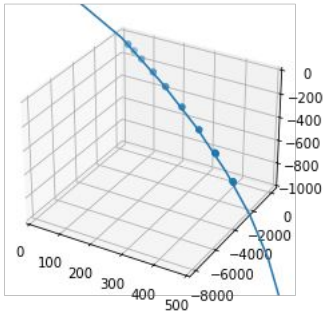
Hypergraph building - parameterized trajectory

- As less accuracy on the extrapolation is required, we are studying the substitution of the numerical integration (4th order Range Kutta) by a parametrized helicoid extrapolation
  - An uniform magnetic field is required (at least per region)
  - The centre of the helicoid, as well as its radius, can be estimated with seed information
- Could be used as a direct substitute for the numerical integration on the layer-by-layer extrapolation
- Or used to find the surfaces that the trajectory interacts with
  - Would be followed by a tree search to find the possible paths





The centre of the helicoid can be found with a Circular Hough Transform (CHT)



Example of extrapolated trajectory with parameterized helicoid compared to particle hits