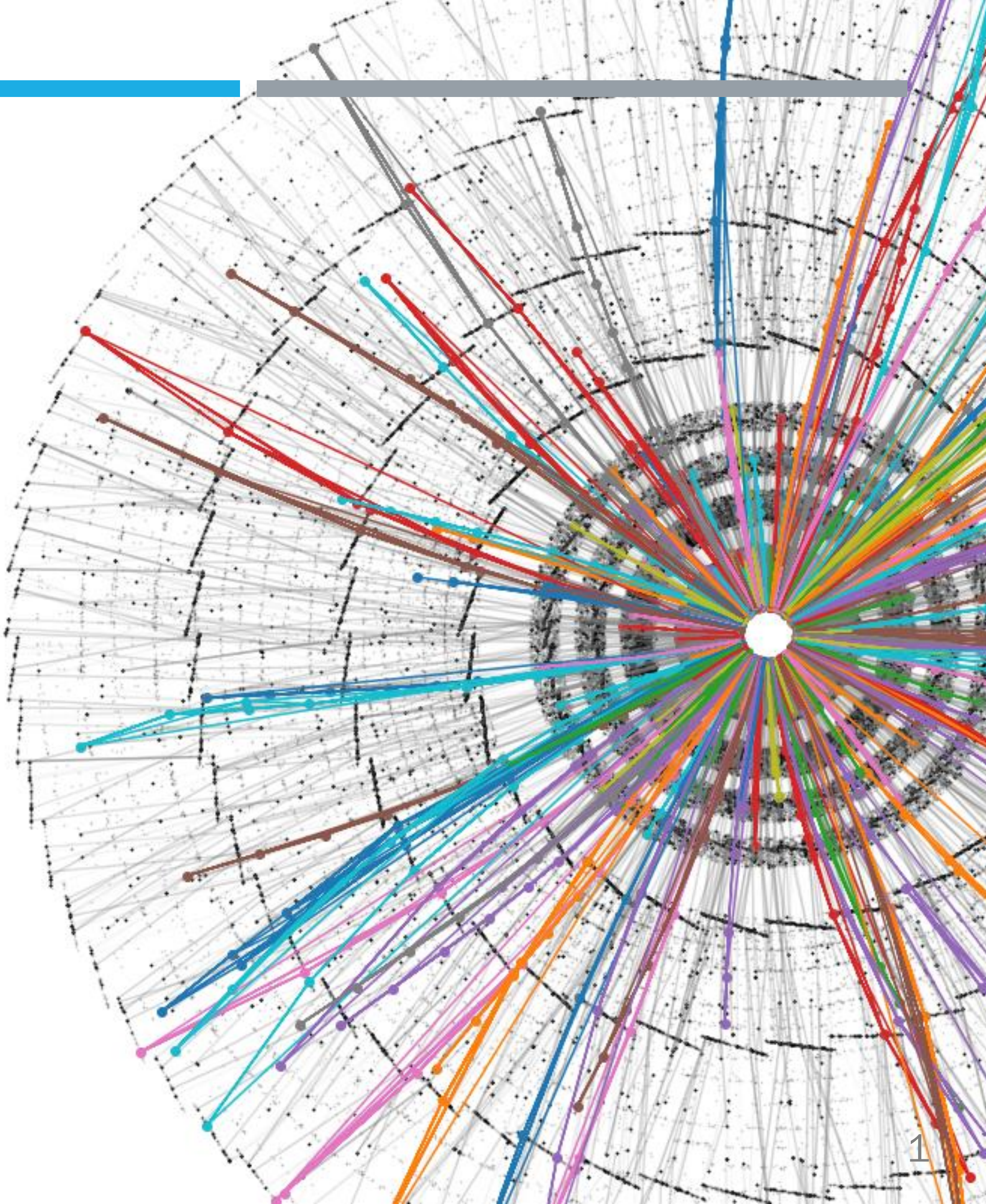
Tracking with Graph Neural Networks

Part 1: Fundamentals

DANIEL MURNANE
BERKELEY LAB, CERN

HIGHRR LECTURE WEEK, HEIDELBERG UNIVERSITY
SEPTEMBER 13, 2023



OVERVIEW

- The importance of tracking in the LHC discovery pipeline
- How have we done tracking in the past
- Tracking with graphs
- Overview of graphs and GNNs
- Construction of graphs
- GNN4ITk project and pipeline
- ATLAS ITk data
- Graph construction in ITk
- The Interaction Network in GNN4ITk
- Graph segmentation techniques
- Track building in GNN4ITk
- Measuring tracking performance
- Track fitting

WARNING: BIAS AHEAD

- I chair the GNN4ITk project in the ATLAS experiment
- I will have a bias towards tracking in ATLAS
- I will also have a bias towards the GNN4ITk "solution" to the ATLAS tracking problem
- However: This approach is the de facto standard way to use GNNs for tracking, since it was first proposed by the HepTrkx project in arxiv:1810.06111

HIGH LUMINOSITY TRACK RECONSTRUCTION

DATA SCIENCE IN THE DISCOVERY PIPELINE

Simulation					Reconstruction				Analysis	
Matrix-element Calculation	Parton-shower / Hadronization	Detector Simulation	Digitization	Topoclusters & Spacepoints	Track Finding & Fitting	Jet Tagging & Vertexing	Particle ID & Particle Flow	Calibration	Likelihood Fitting	Unfolding

Numerical Integration

Markov Chain Monte Carlo

Topological clustering

Kalman Filtering & Fitting

Conformal Fits & Hough Transform

Statistical Techniques, Bayesian Inference

ML TODAY & TOMORROW IN THE DISCOVERY PIPELINE

Simulation				Reconstruction					Analysis	
Matrix-element Calculation	Parton-shower / Hadronization	Detector Simulation	Digitization	Topoclusters & Spacepoints	Track Finding & Fitting	Jet Tagging & Vertexing	Particle ID & Particle Flow	Calibration	Likelihood Fitting	Unfolding

Generative Models: GANs, VAEs, Normalizing Flows and Diffusion

Metric Learning, Object Condensation

Deep Full Event Reconstruction

CNNs, Graph Neural Networks & Transformers

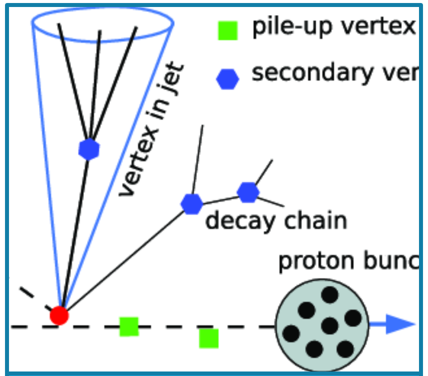
Symmetric ML & Equivariance

Autoencoders & Anomaly Detection

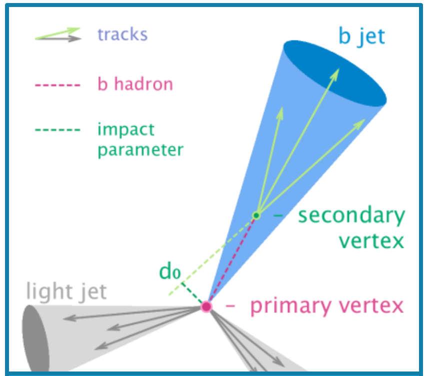
Omnifold and Likelihood-free Inference

TASKS IN AN HL-LHC DETECTOR

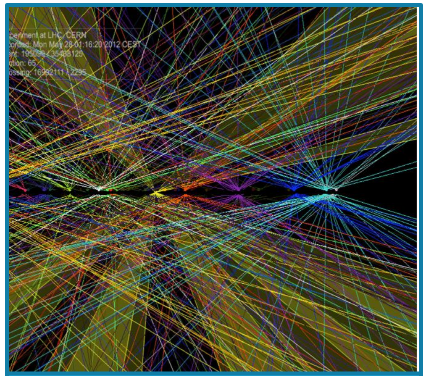
- In order to perform the analysis that leads to discovery (e.g. of dark matter, extra dimensions, SUSY, …), need to make sense of the detector read-out
- There are many tasks required to reconstruct the physics event behind the read-out



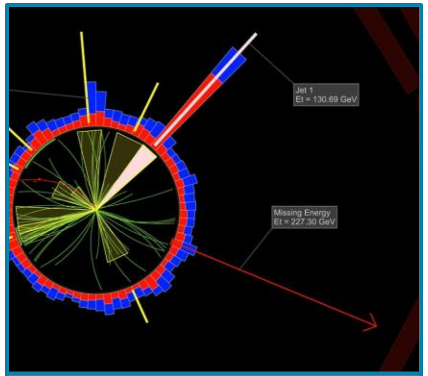
Vertex Reconstruction



Jet Tagging



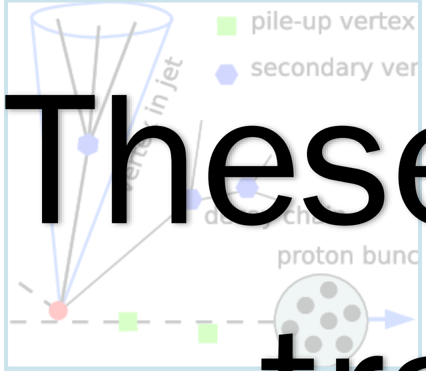
Pile-up Removal



Missing Energy Reconstruction

TASKS IN AN HL-LHC DETECTOR

- In order to perform the analysis that leads to discovery (e.g. of dark matter, extra dimensions, SUSY, …), need to make sense of the detector read-out
- There are many tasks required to reconstruct the physics event behind the read-out



Vertex Reconstruction

Jet Tagging

Pile-up Removal

Missing Energy Reconstruction

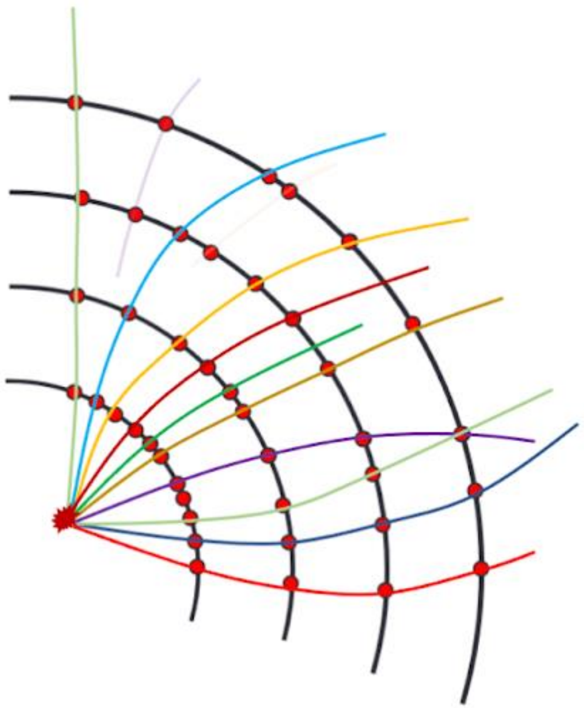
These all require accurate track reconstruction

WHAT IS TRACK RECONSTRUCTION

- Protons collide in center of detector, “shattering” into thousands of particles
- The *charged* particles travel in curved **tracks** through detector’s magnetic field (Lorentz force)
- A track is defined by the **hits** left as energy deposits in the detector material, when the particle interacts with material
- The goal of track reconstruction:

Given set of hits from particles in a detector, assign label(s) to each hit.

Perfect classification: All hits from a particle (*and only those hits)* share the same label



THE IMPORTANCE OF TRACKING

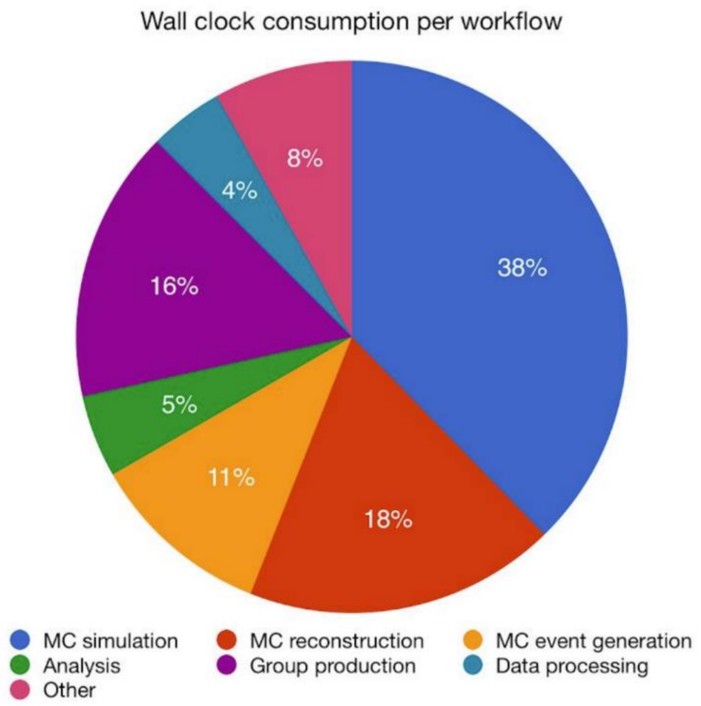
- Finding and fitting tracks accurately is essential for most downstream tasks in ATLAS and many other experiments
- Classic example is b-tagging (which itself is necessary for Higgs searches, top physics, and BSM searches)
- The current ATLAS GNN tagger takes 2 overall jet features, and *21 track features*

Jet Input	Description
p_T	Jet transverse momentum
η	Signed jet pseudorapidity
Track Input	**Description**
q/p	Track charge divided by momentum (measure of curvature)
$\mathrm{d}\eta$	Pseudorapidity of the track, relative to the jet η
$\mathrm{d}\phi$	Azimuthal angle of the track, relative to the jet ϕ
d_0	Closest distance from the track to the PV in the longitudinal plane
$z_0 \sin\theta$	Closest distance from the track to the PV in the transverse plane
$\sigma(q/p)$	Uncertainty on q/p
$\sigma(\theta)$	Uncertainty on track polar angle θ
$\sigma(\phi)$	Uncertainty on track azimuthal angle ϕ
$s(d_0)$	Lifetime signed transverse IP significance
$s(z_0)$	Lifetime signed longitudinal IP significance
nPixHits	Number of pixel hits
nSCTHits	Number of SCT hits
nIBLHits	Number of IBL hits
nBLHits	Number of B-layer hits
nIBLShared	Number of shared IBL hits
nIBLSplit	Number of split IBL hits
nPixShared	Number of shared pixel hits
nPixSplit	Number of split pixel hits
nSCTShared	Number of shared SCT hits
nPixHoles	Number of pixel holes
nSCTHoles	Number of SCT holes

ATL-PHYS-SLIDE-2023-048

THE COST OF TRACKING

- Over half the current ATLAS computing budget is spent on generating and reconstructing simulated data
- In Run 2 in 2018, a typical event (in data) required 1693 HS06-seconds, of which 67% was spent on tracking
- TL;DR: Tracking is an expensive piece of reconstruction, and is therefore an expensive piece of any experiment that has a tracking subdetector

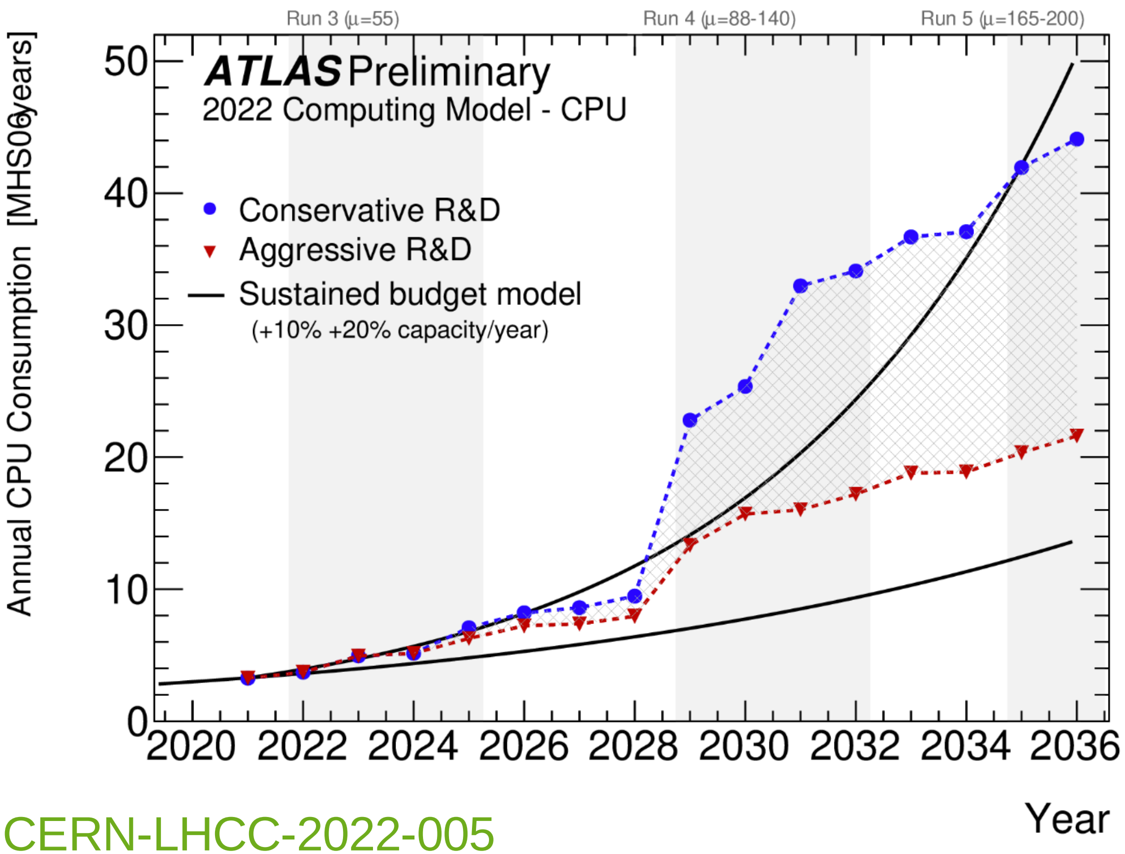


Detector	$\langle\mu\rangle$	inner tracking	muon spectrometer and calorimeter	combined reconstruction	monitoring	total
Run 2	90	1137	149	301	106	1693

https://cds.cern.ch/record/2729668/files/LHCC-G-178.pdf

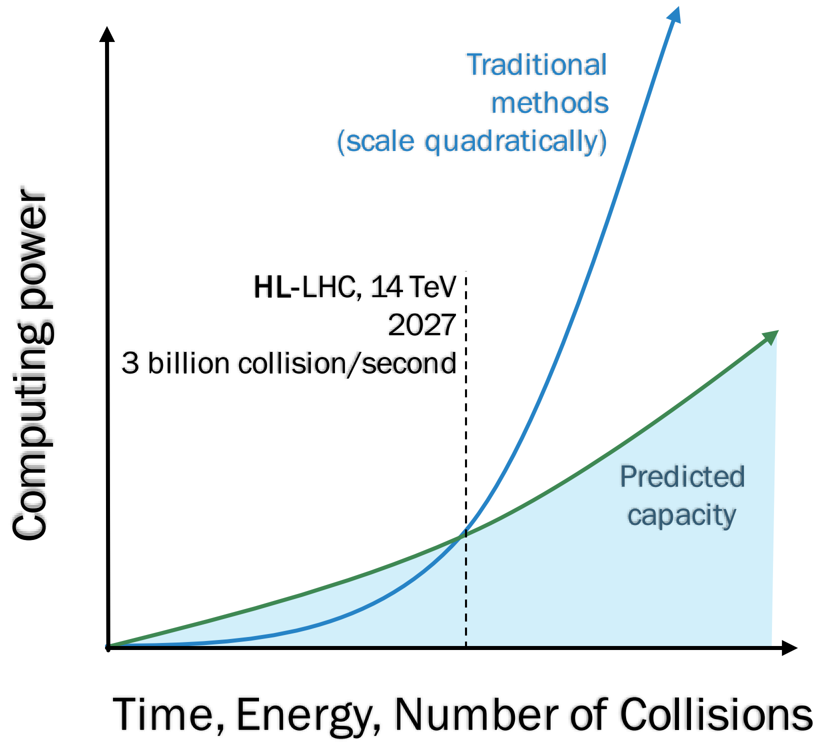
COMPUTE SCALING FOR HIGH LUMINOSITY

ATLAS Computing Requirements Over Time



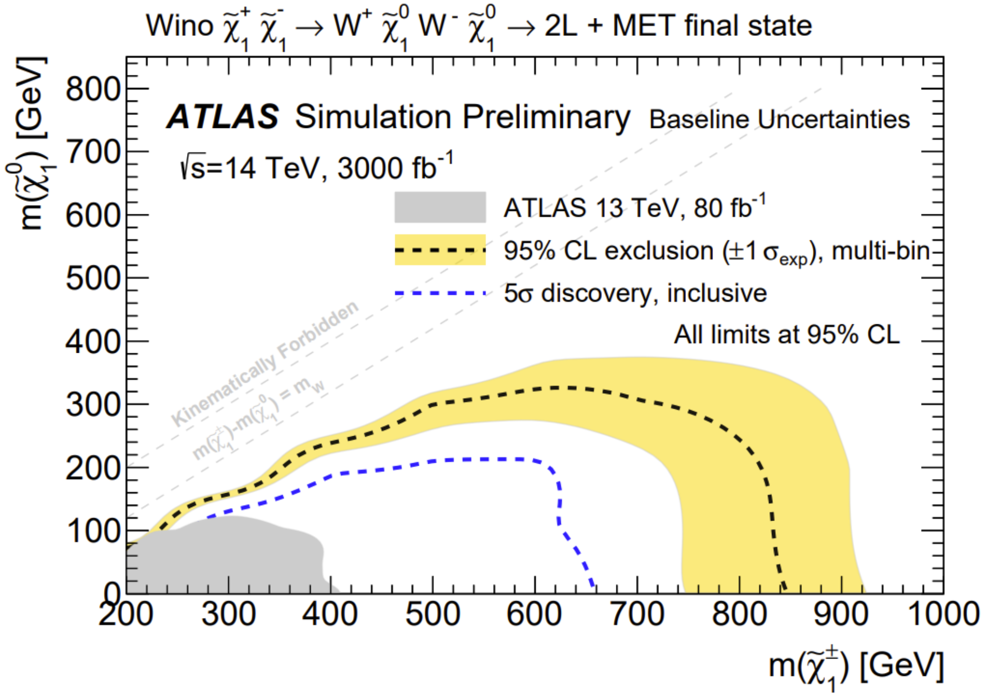
CERN-LHCC-2022-005

In other words…

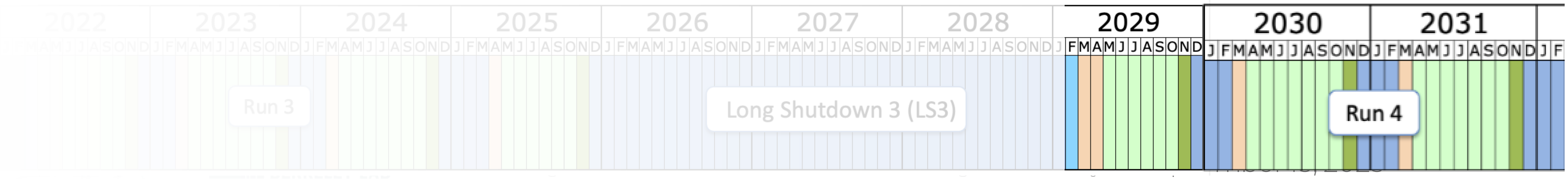


WHY HIGH LUMINOSITY PHYSICS?

1. Better reach for Supersymmetry discovery:
   a) Electroweakino particles produced by much greater range of chargino masses
   b) Gluino exclusion from channels across 0.7-2.0TeV to channels across 2.5-3.2TeV
2. Sensitive to resonances (W', Z') up to 6-8TeV
3. W mass precision improvement from [±9.4MeV](#) to [±6MeV](#)



ATL-PHYS-PUB-2018-048



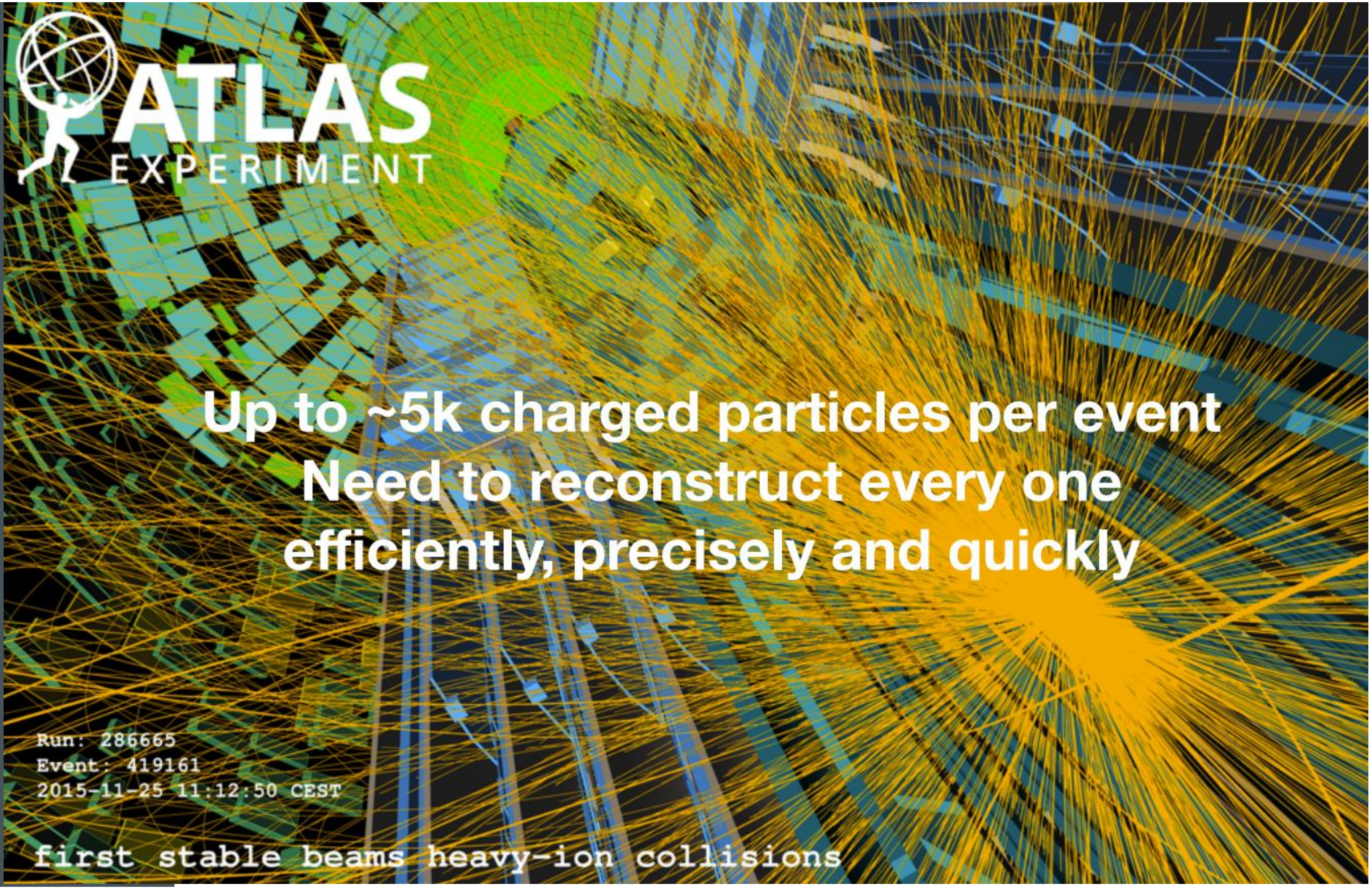
...

LHC Long Term

TRACKING 101

THE GOAL OF TRACKING

The following slides have material borrowed from Heather Gray's excellent talk @ Zurich: https://indico.cern.ch/event/504284/



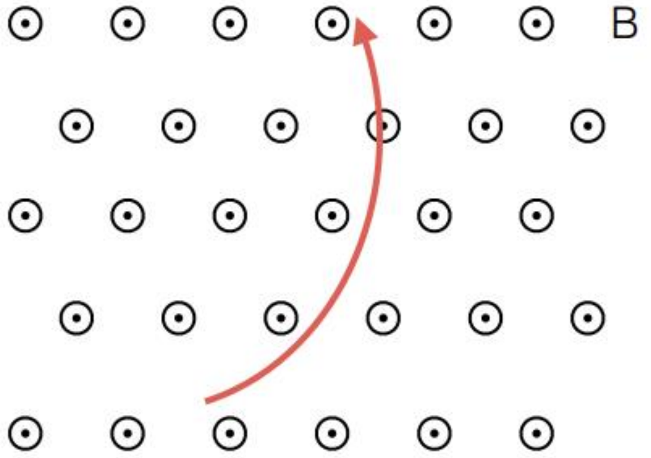
WITHOUT A DETECTOR…

Particles are well-described by helices in a magnetic field

But once they are in a detector…

Particles in a magnetic field

- Magnetic field bends charged particles to measure their momenta
  - in a perfect homogenous field : circle in transverse direction
  - helical track in a solenoidal field
    - transverse & longitudinal components are independent
- ATLAS field is far from homogenous
- Solve equations numerically!

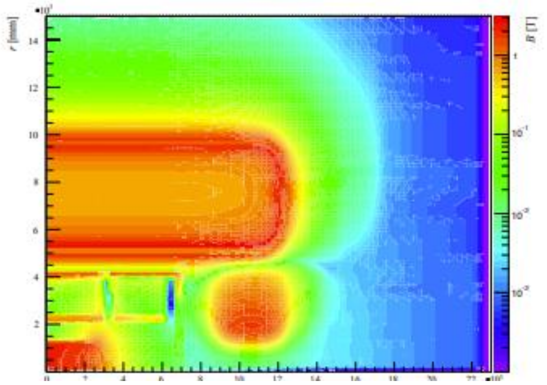


$$\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right]$$

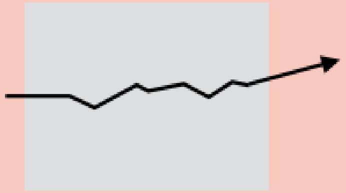
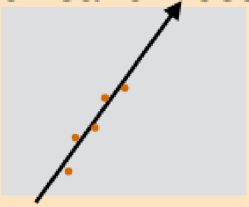
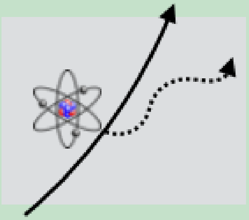
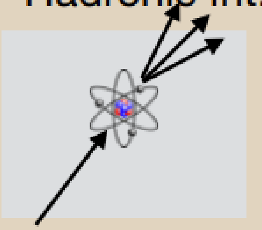
$$\frac{d^2x}{dz^2} = \frac{q}{p}R\left[\frac{dx}{dz}\frac{dy}{dz}B_x - \left(1+\left(\frac{dx}{dz}\right)^2\right)B_y + \frac{dy}{dz}B_z\right]$$

$$\frac{d^2y}{dz^2} = \frac{q}{p}R\left[\left(1+\left(\frac{dy}{dz}\right)^2\right)B_x - \frac{dx}{dz}\frac{dy}{dz}B_y - \frac{dx}{dz}B_z\right]$$

$$R = \frac{ds}{dz} = \sqrt{1+\left(\frac{dx}{dz}\right)^2+\left(\frac{dy}{dz}\right)^2}$$



TYPES OF INTERACTION

Type	Particles	Fund. parameter	Characteristics	Effect
Multiple Scattering 	all charged particles	radiation length X_0	almost gaussian average effect 0 depends ~ $1/p$	deflects particles, increases measurement uncertainty
Ionisation loss 	all charged particles	effective density $A/Z * \rho$	small effect in tracker, small dependence on p	increases momentum uncertainty
Bremsstrahlung 	all charged particles, dominant for e	radiation length X_0	highly non-gaussian, depends ~ $1/m^2$	introduces measurement bias
Hadronic Int. 	all hadronic particles	nuclear interaction length Λ_0	destroys particle, rather constant effect in p	main source of track reconstruction inefficiency

REPRESENTING A HELIX

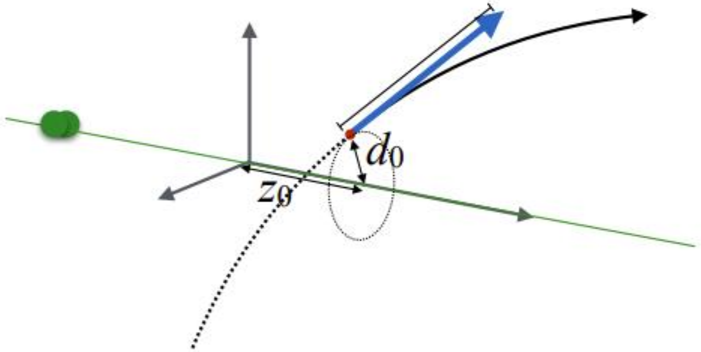
Track Parametrisation

- A trajectory of a charged particle in a magnetic field requires **five track parameters** (q)

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

- Uncertainties encoded in a covariance matrix

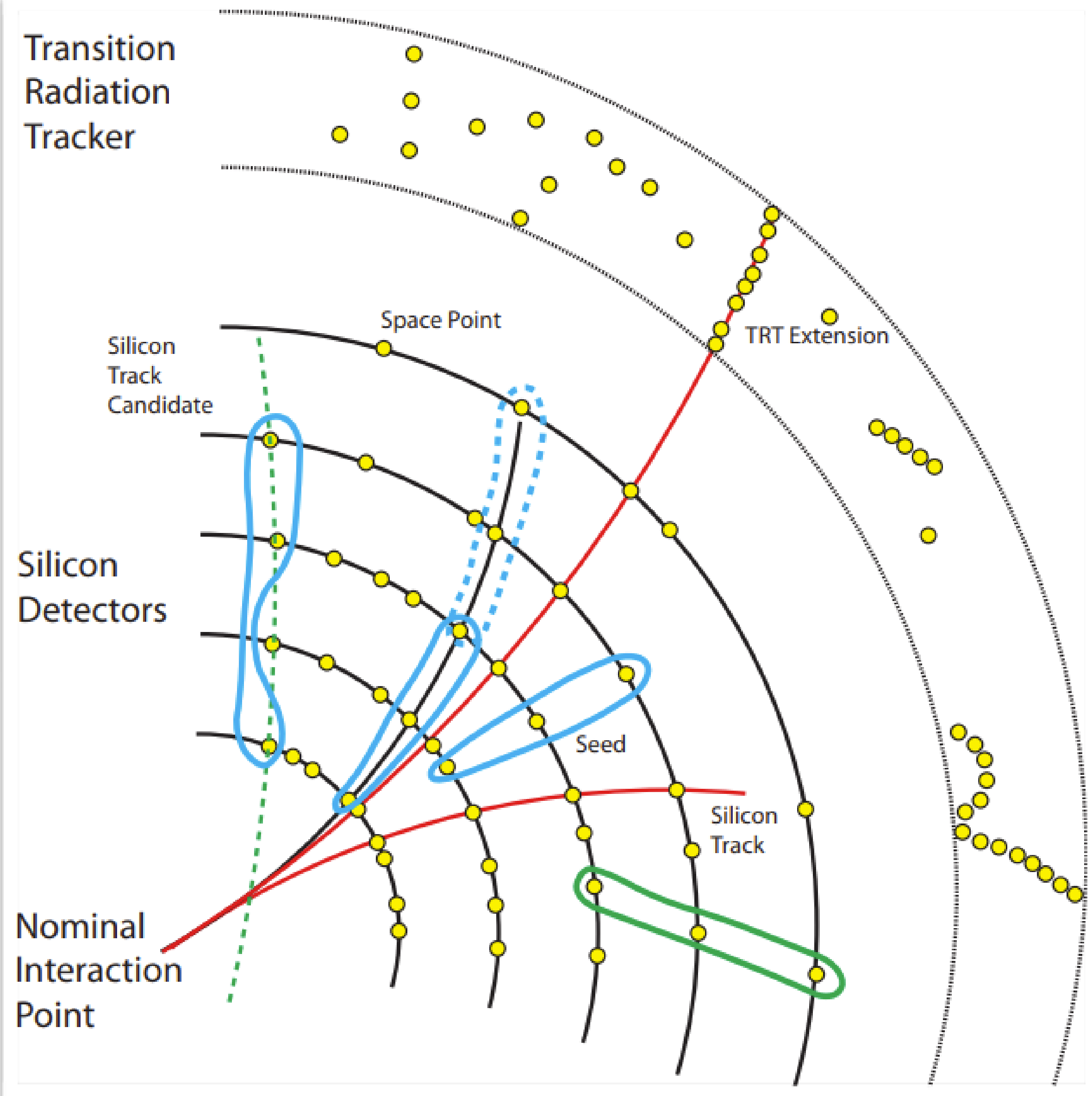
$$\mathbf{C} = \begin{pmatrix} \sigma^2(d_0) & cov(d_0, z_0) & cov(d_0, \phi) & cov(d_0, \theta) & cov(d_0, q/p) \\ . & \sigma^2(z_0) & cov(z_0, \phi) & cov(z_0, \theta) & cov(z_0, q/p) \\ . & . & \sigma^2(\phi) & cov(\phi, \theta) & cov(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$



- Right handed coordinate system
- Azimuthal angle, ϕ, measured in transverse plane in [-π,+π)
- Polar angle, θ measured from z axis in [0,π]
- Pseudorapidity, $\eta = -\ln(\tan\theta/2)$

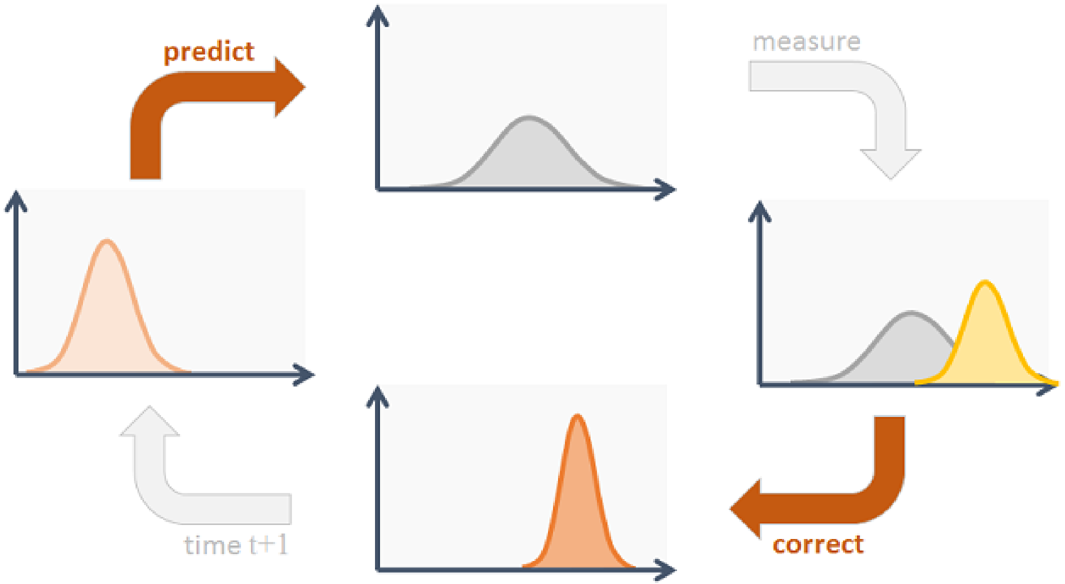
TRACKING TERMINOLOGY

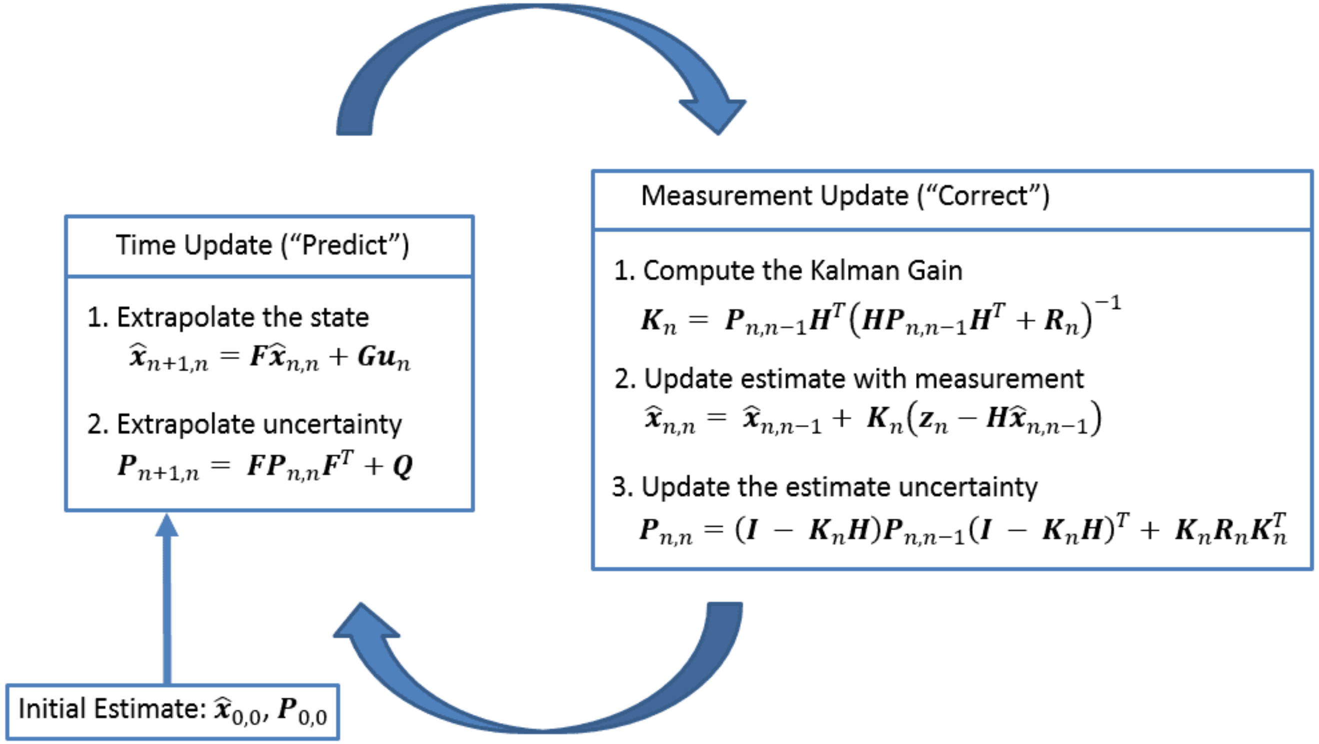
- Tracking typically happens in silicon: channels lie on flat-ish modules
- Like a set of millions of cameras, arranged in layers
- Particles curve out, depositing energy in “clusters” or “spacepoints” or “hits”
- Sequence of hits is a “track”
- A *prediction* of a track is a “track candidate”



TRADITIONAL TECHNIQUES FOR TRACKING

KALMAN FILTER: TRACKING AS NAVIGATION





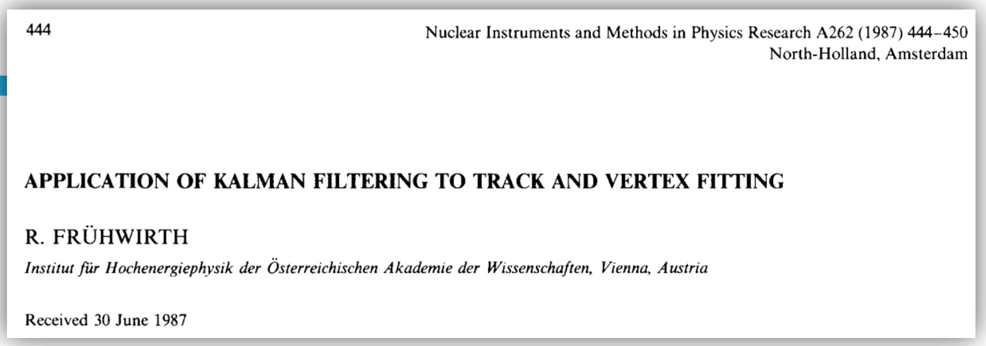
- A recurrent sequence of predictions and corrections according to measurement

https://www.kalmanfilter.net/multiSummary.html

KALMAN FILTER: TRACKING AS NAVIGATION

- The optimal algorithm for any linear system with independent measurements with Gaussian uncertainty
- Self-driving cars are the perfect use-case of this: many independent sensor measurements, with a planned trajectory that is updated in time
- Tracking looks a lot like driving a car…

KALMAN FILTER: TRACKING AS NAVIGATION


444

Nuclear Instruments and Methods in Physics Research A262 (1987) 444–450
North-Holland, Amsterdam

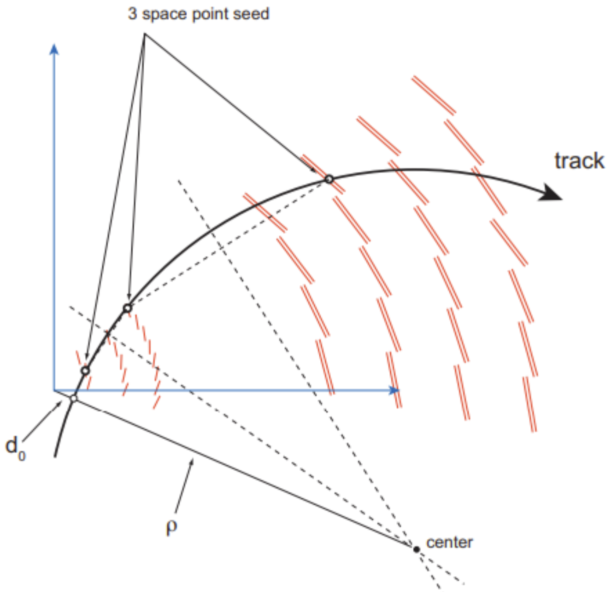
APPLICATION OF KALMAN FILTERING TO TRACK AND VERTEX FITTING

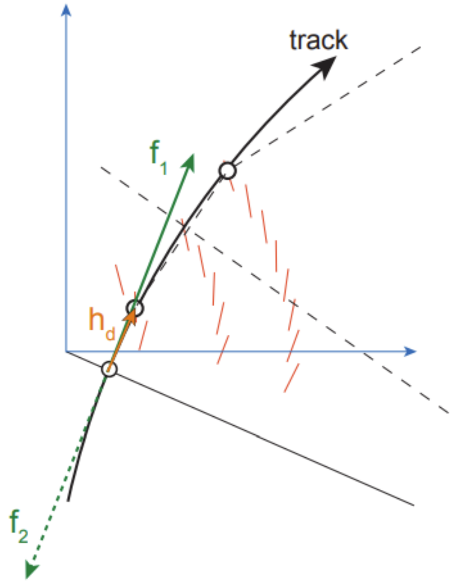
R. FRÜHWIRTH

Institut für Hochenergiephysik der Österreichischen Akademie der Wissenschaften, Vienna, Austria

Received 30 June 1987

- Begin with a seed of 3 or 4 spacepoints
- Produce a prediction of the helix parameters, and the covariance matrix
  - Look at where this helix would intersect with the next layer(s)
  - Look for a nearby hit to this prediction
  - Use the most likely hit to update the model
- Repeat!
- We can also hypothesise a "hole" (missing hit) to handle a skipped layer

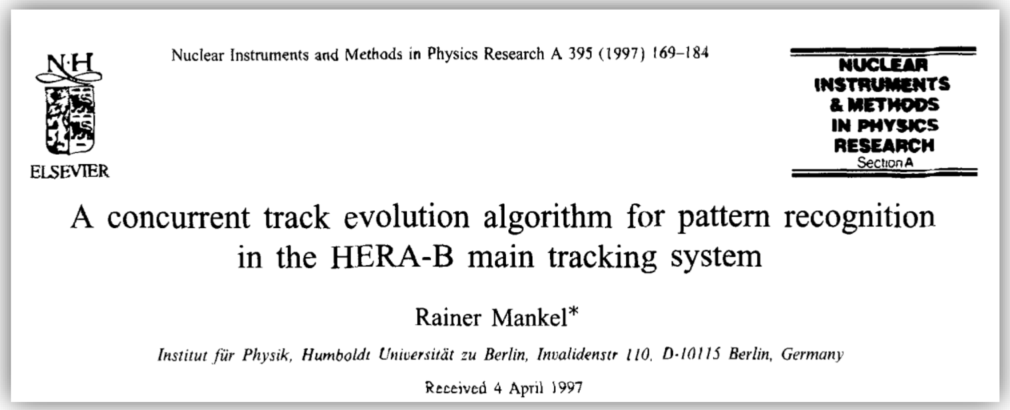




https://cds.cern.ch/record/1281363/files/ATLAS-CONF-2010-072.pdf

COMBINATORIAL KALMAN FILTER

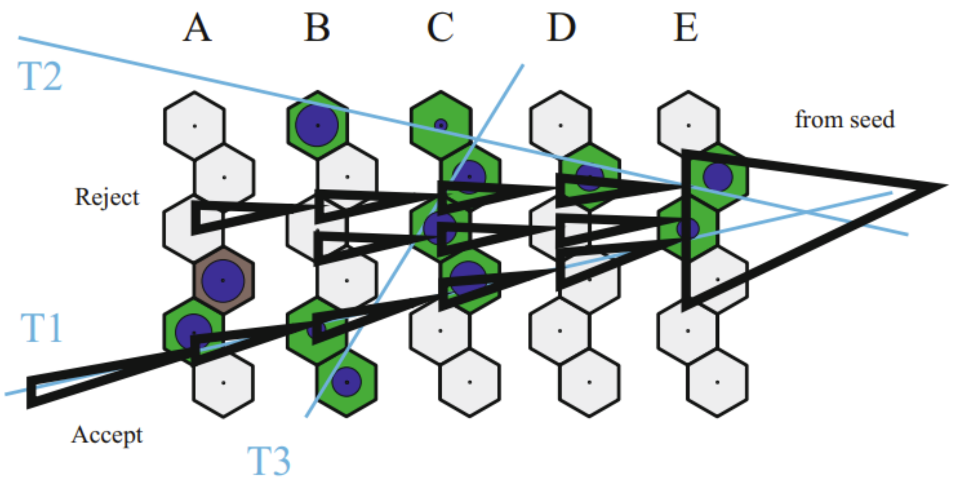
- Running the Kalman Filter naively leads to an exponential explosion *in time* of possible paths
- We can improve this in time (if not necessarily in memory) by recursively looking for combinations of hits that match a prediction
- If several candidate paths match a seed, then the value with the lowest χ^2 value and the most hits is considered the "winner"
- This still scales combinatorially in time-space, and can be very expensive if the number of seeds is high


Nuclear Instruments and Methods in Physics Research A 395 (1997) 169–184
NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A
ELSEVIER

A concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system

Rainer Mankel*
Institut für Physik, Humboldt Universität zu Berlin, Invalidenstr 110, D-10115 Berlin, Germany

Received 4 April 1997

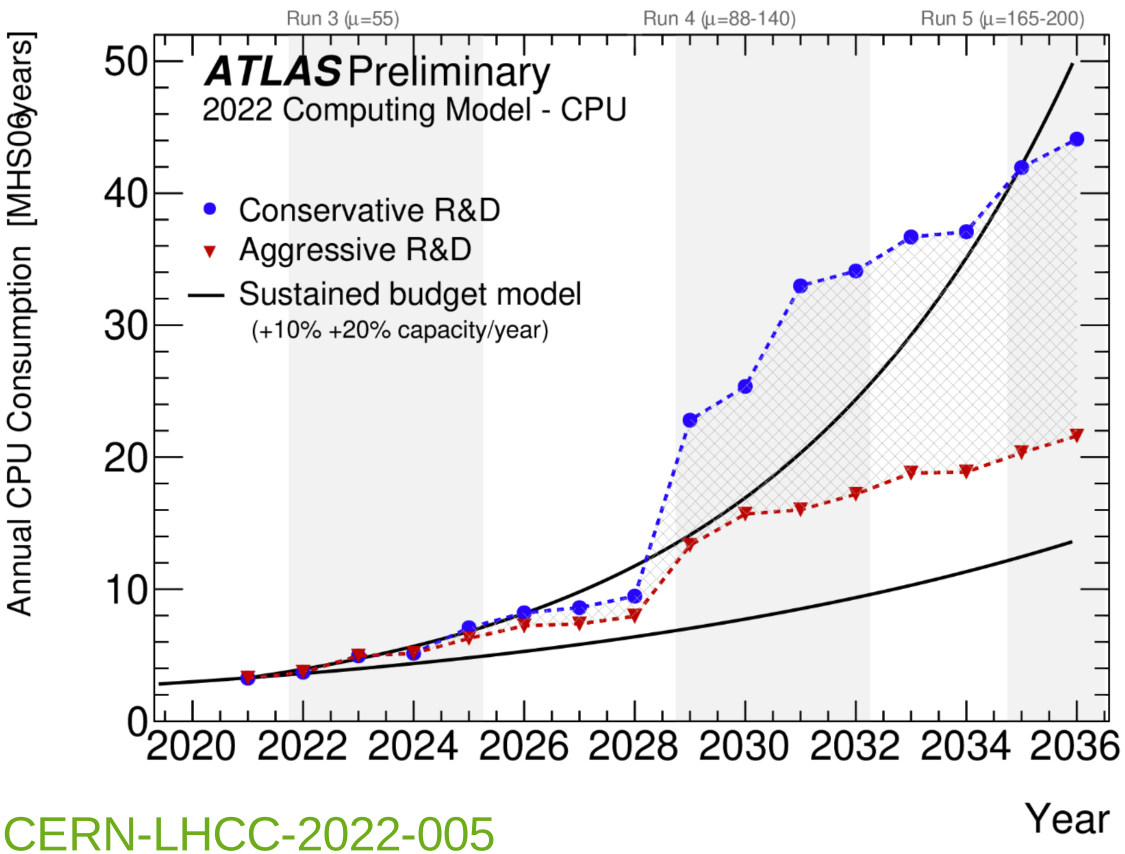


https://www.researchgate.net/publication/344039130_Pattern_Recognition_and_Reconstruction

TRACKING AS GRAPH SEGMENTATION

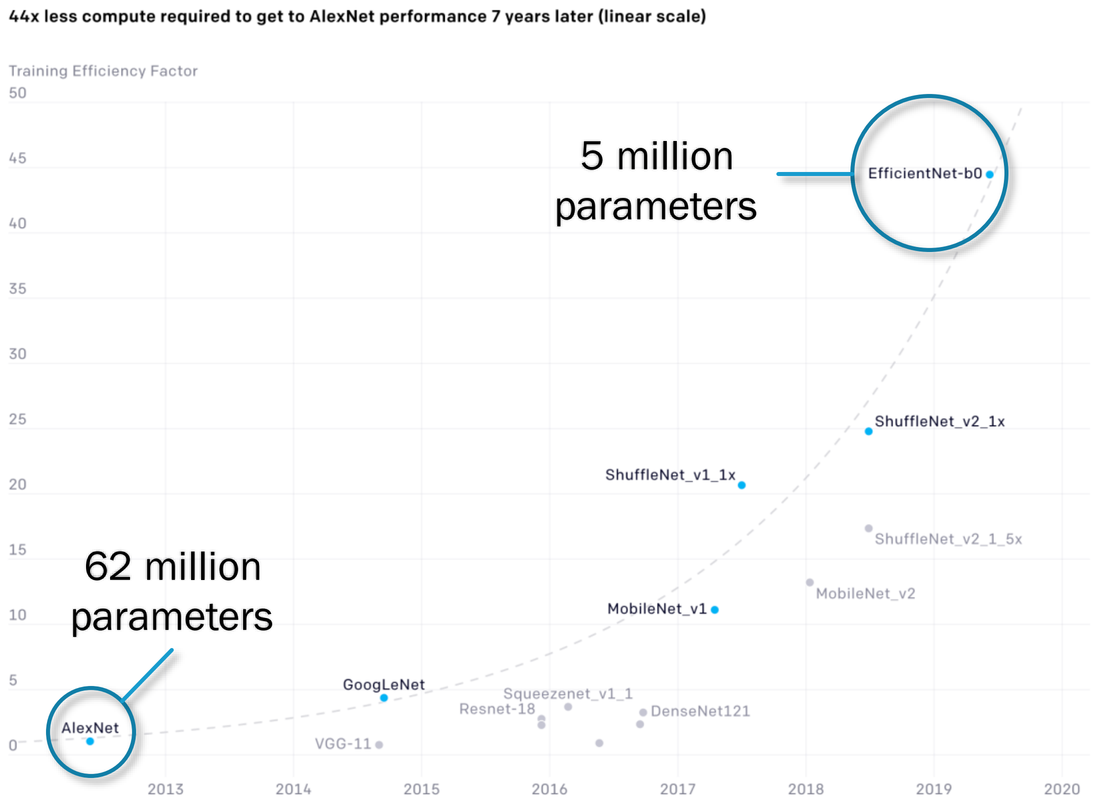
COMPUTE SCALING FOR HIGH LUMINOSITY

<u>ATLAS Computing Requirements Over Time</u>



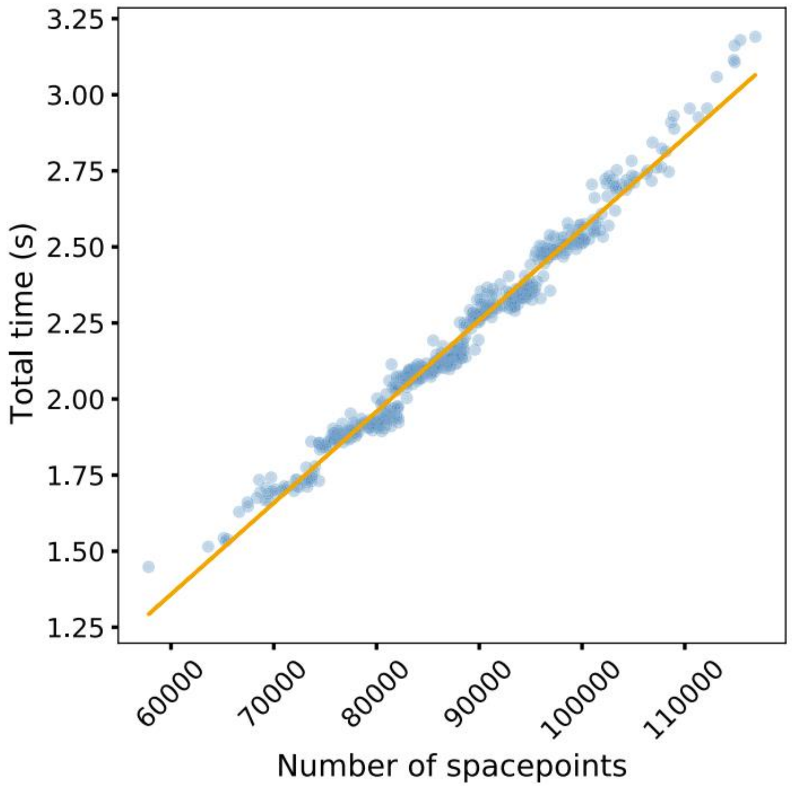
CERN-LHCC-2022-005

<u>ML Image Classification Efficiency Over Time</u>



TEASER: GRAPH-BASED PIPELINE FOR TRACK RECONSTRUCTION

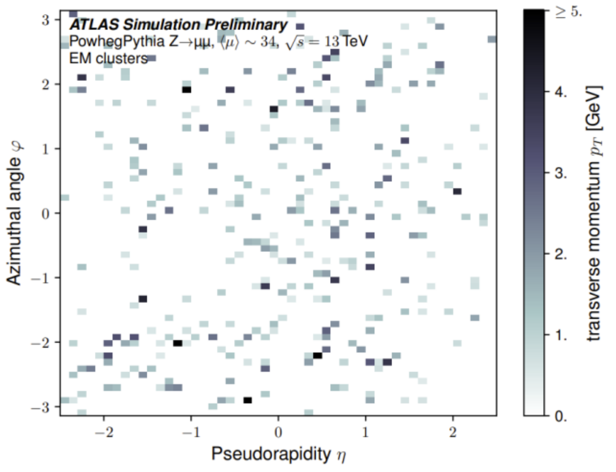
- Using **graph-based ML**, can perform track reconstruction on High Luminosity detector events
- Comparable efficiency and fake rates to traditional algorithms
- Scaling that is approximately linear in event size (on open-source TrackML dataset)



HOW SHOULD WE REPRESENT PARTICLE COLLISIONS?

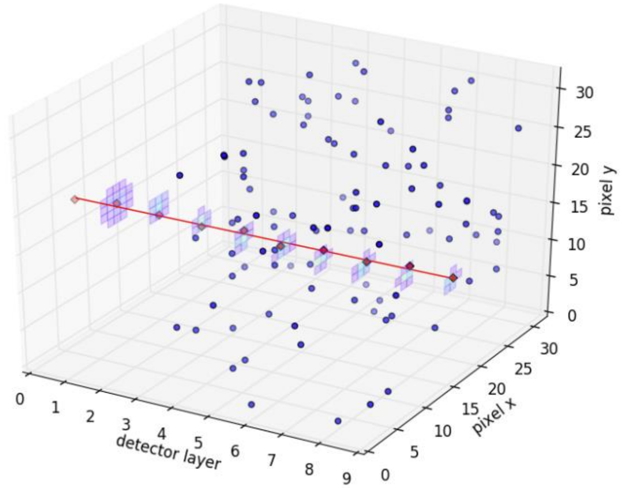
Assuming we want to use deep learning, how can we represent a particle collision?

Image?



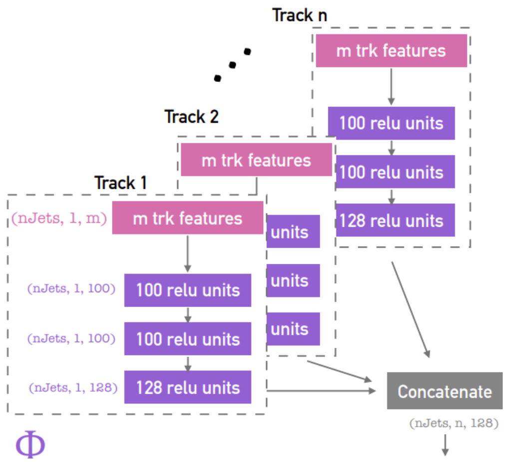
Convolutional Neural Networks with Event Images..., ATLAS Collab.

Sequence?



Particle Track Reconstruction with Deep Learning, Farrell et al

Set/Point Cloud?

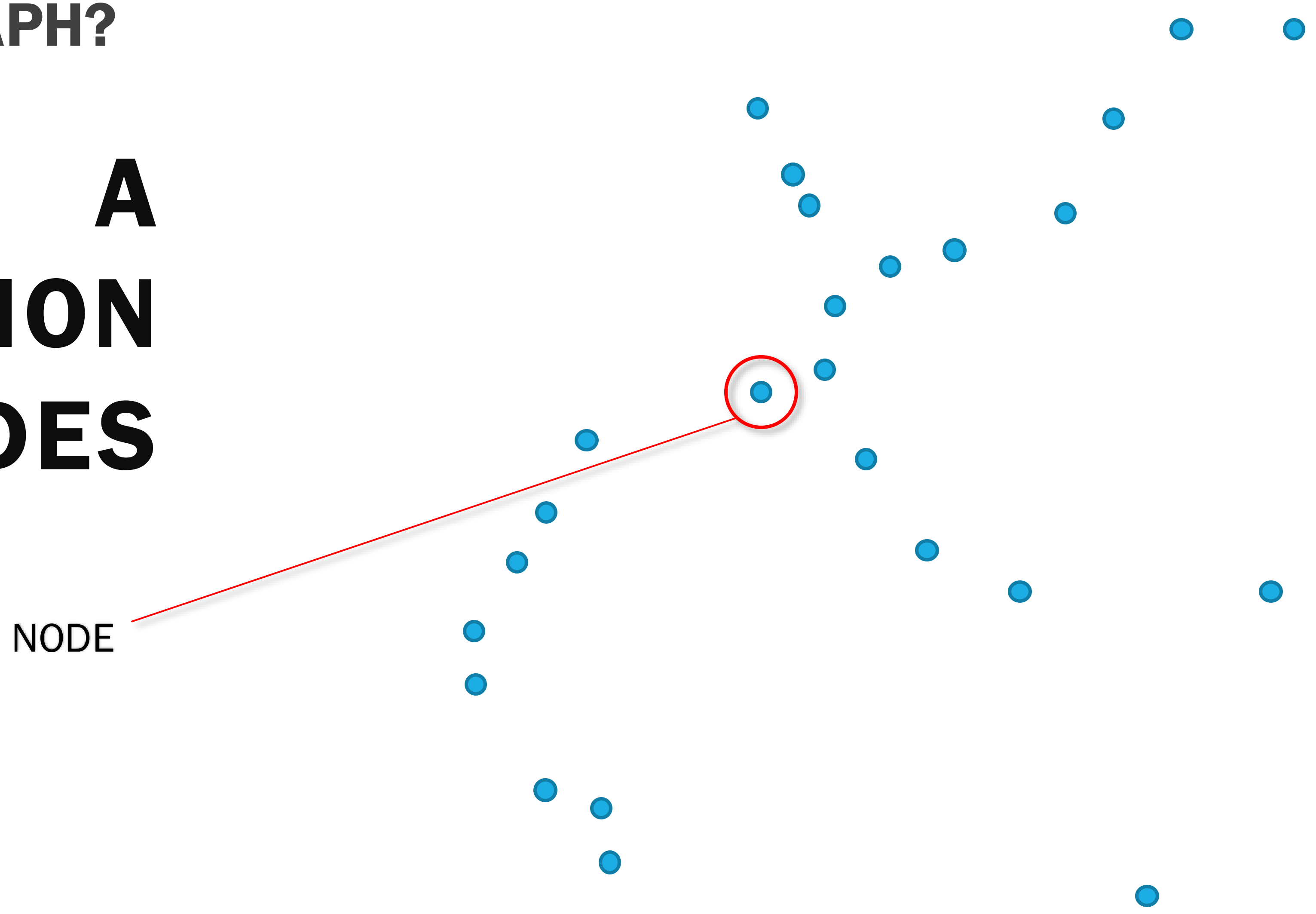


Deep Sets based Neural Networks for Impact Parameter..., ATLAS Collab

For event collision as point cloud, with relationships between points, this is a graph.

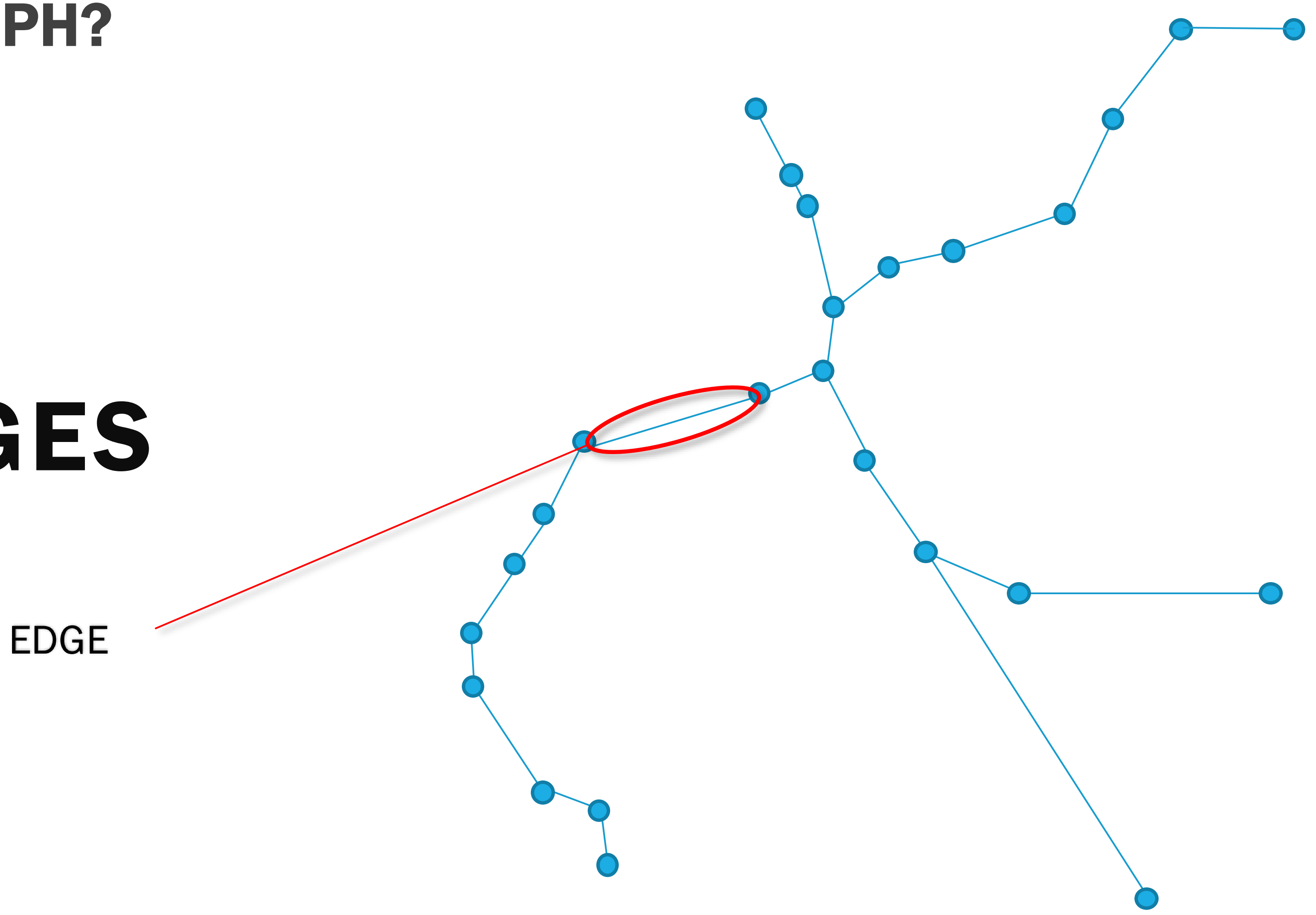
WHAT IS A GRAPH?

A COLLECTION OF NODES



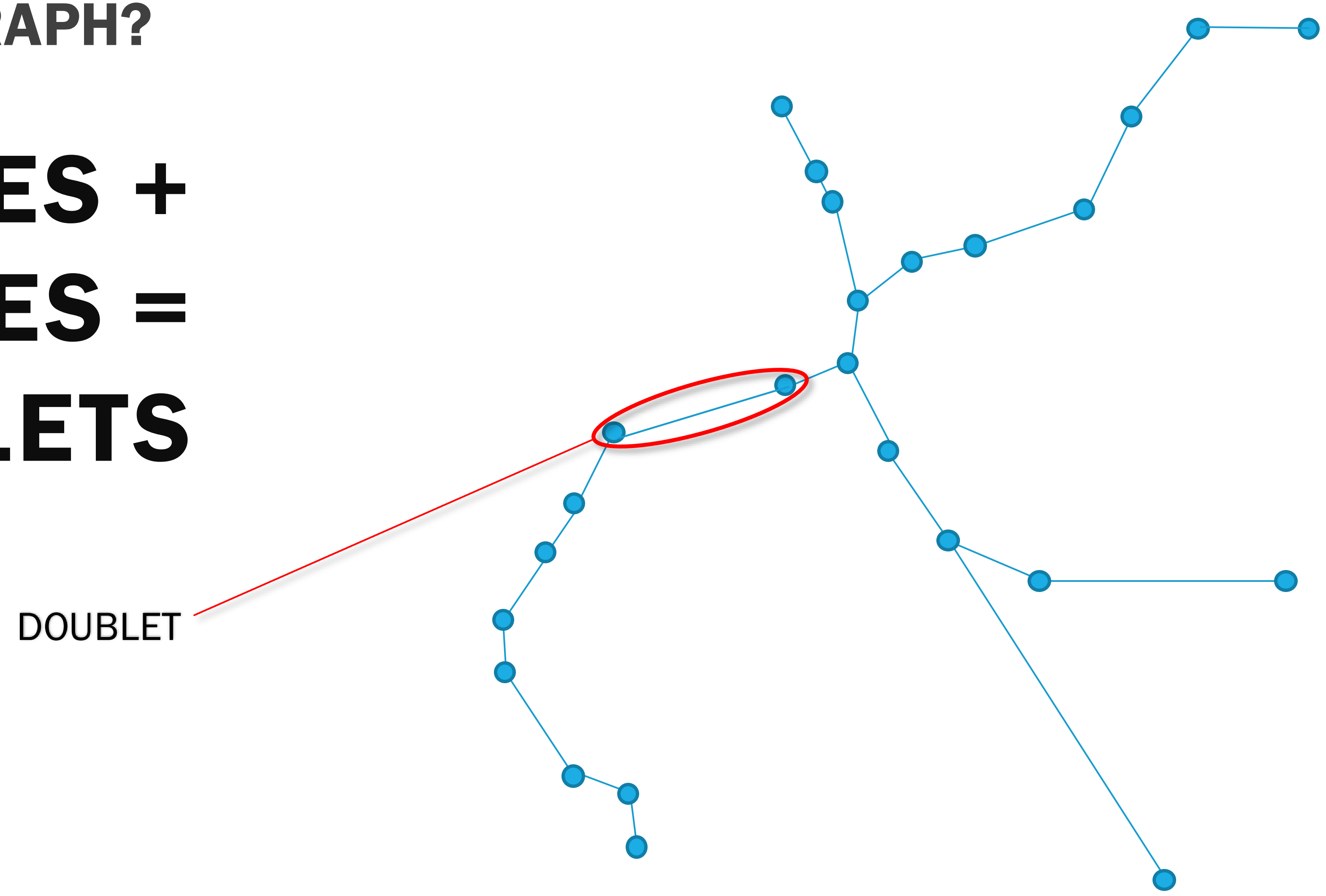
WHAT IS A GRAPH?

AND EDGES



WHAT IS A GRAPH?

NODES + EDGES = DOUBLETS



WHAT IS A GRAPH?

NODES CAN HAVE FEATURES

NODE FEATURE
e.g. "West Oakland"

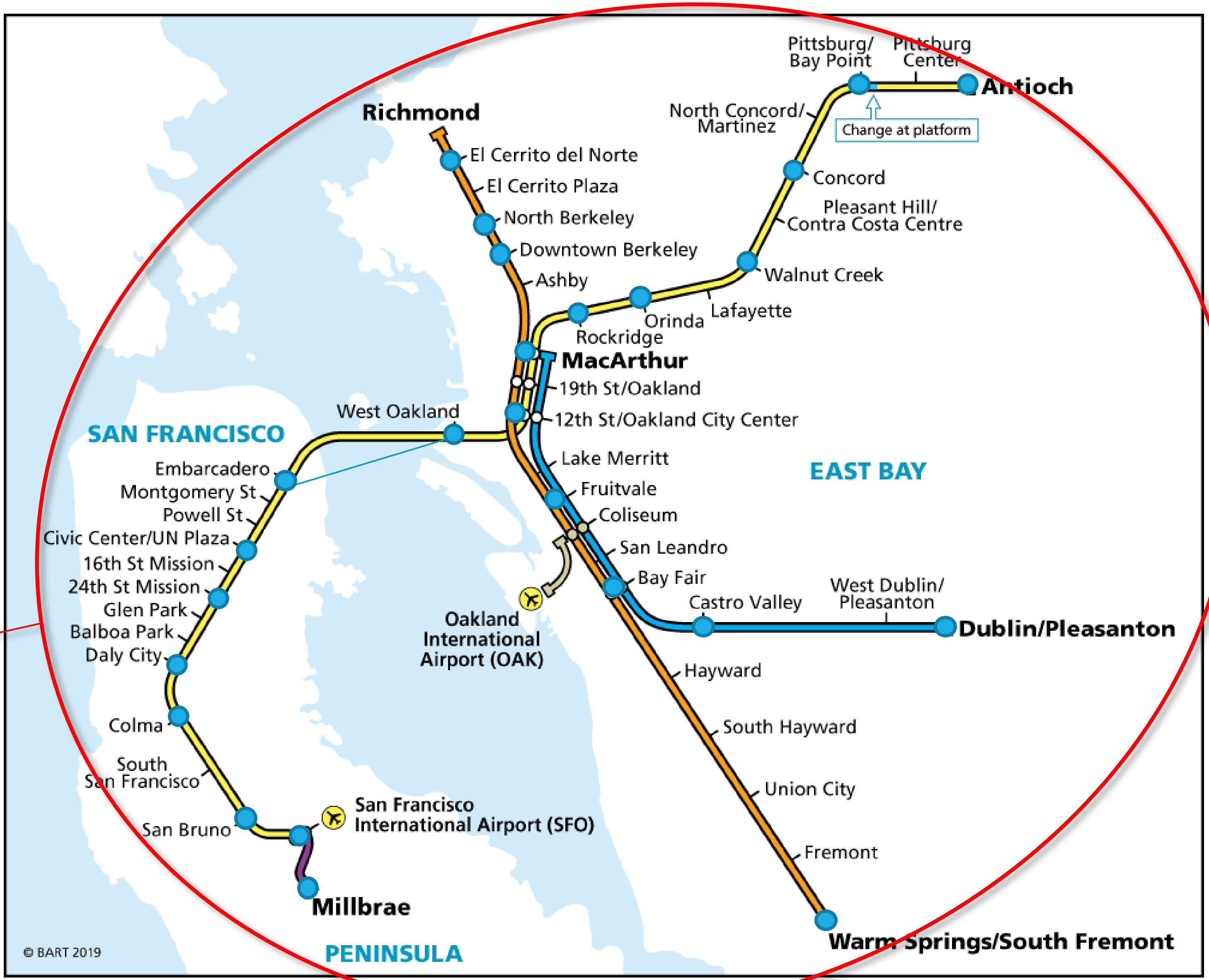
WHAT IS A GRAPH?

EDGES CAN HAVE FEATURES

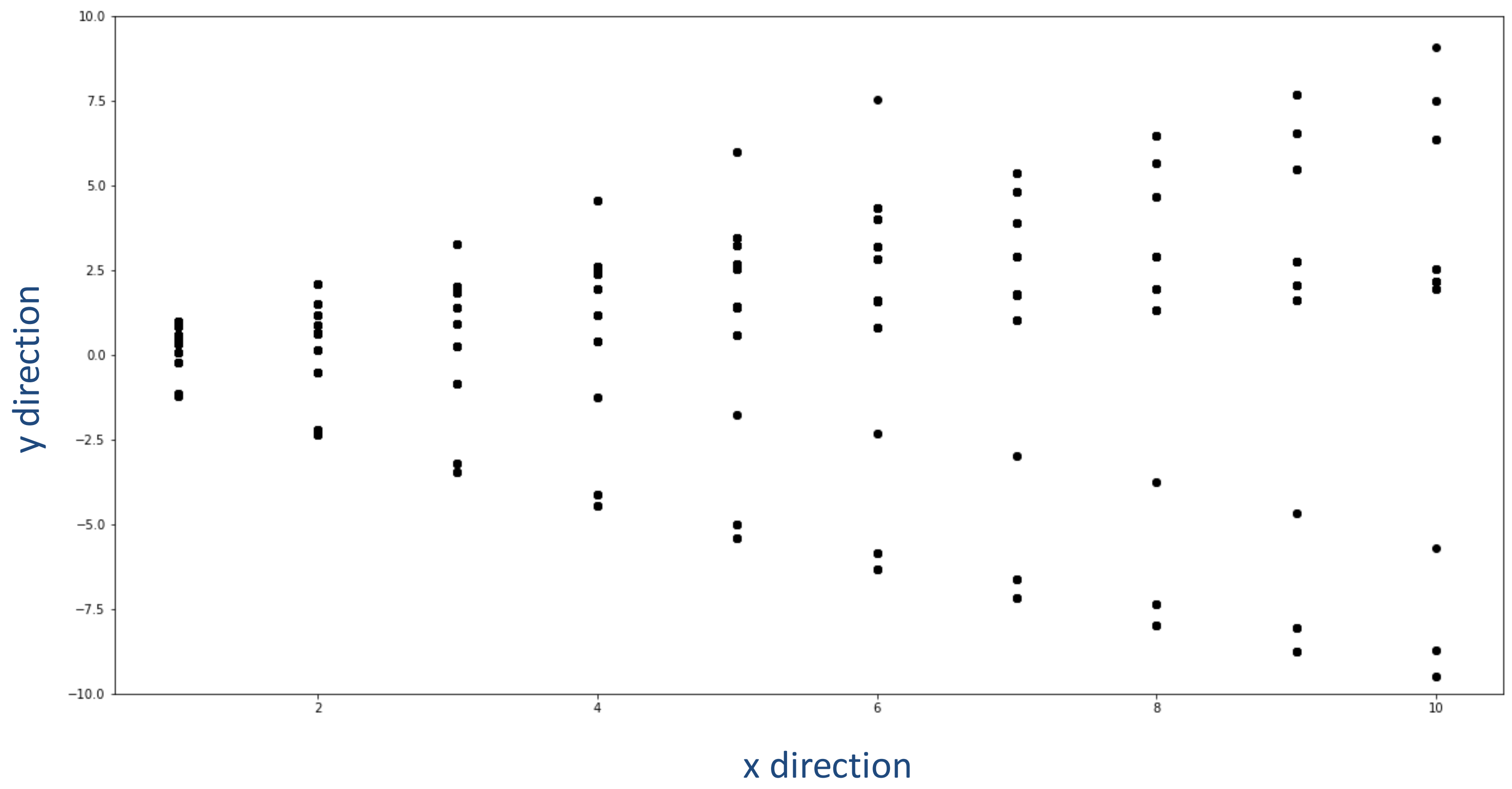
EDGE FEATURE
e.g. “Under Maintenance – Single Track”

THE WHOLE GRAPH CAN HAVE FEATURES

GRAPH FEATURE
e.g. “Sunday Timetable”

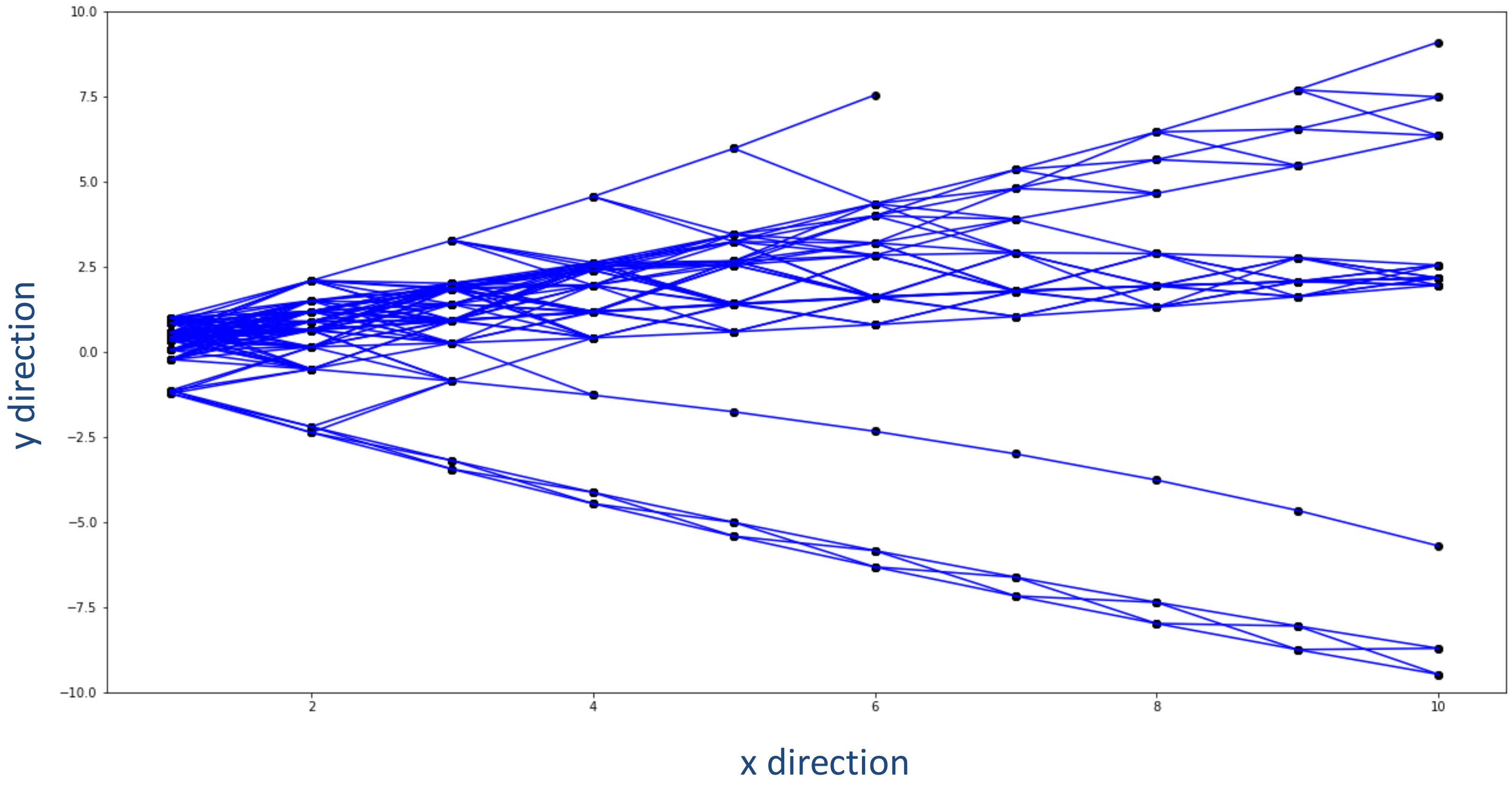


GRAPHS ARE A NATURAL WAY TO REPRESENT TRACKS



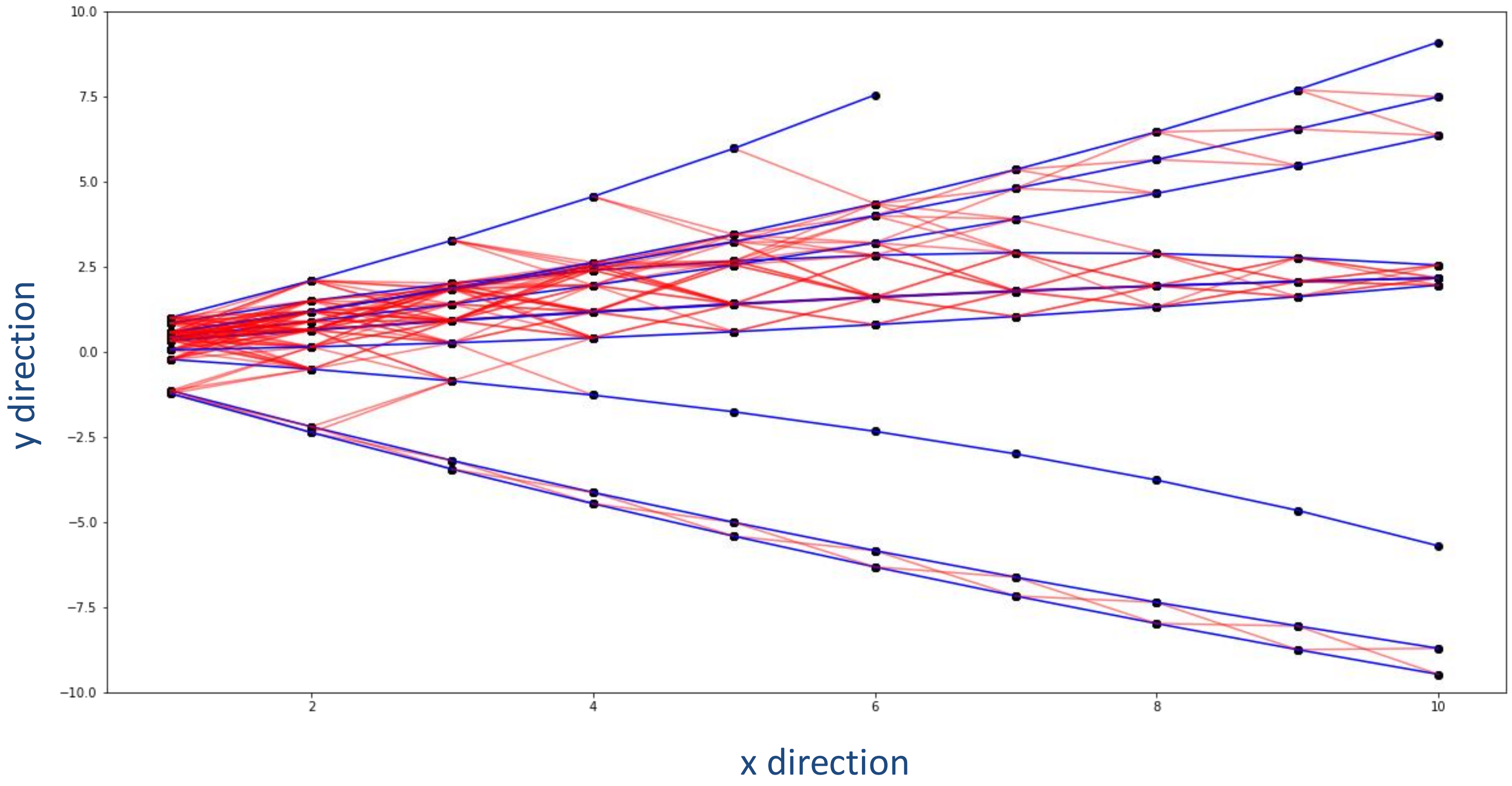
Given hits on layers of a detector

GRAPHS ARE A NATURAL WAY TO REPRESENT TRACKS



Connect the
hits in some way

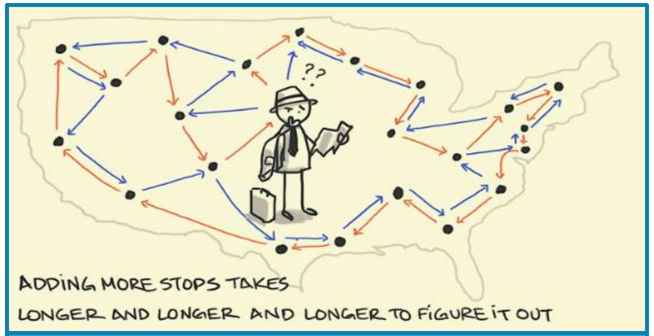
GRAPHS ARE A NATURAL WAY TO REPRESENT TRACKS



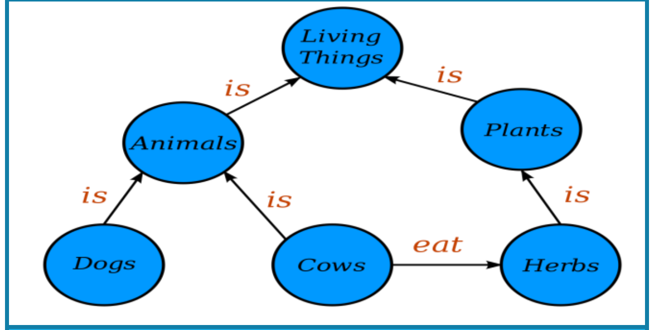
- Tracks should be found amongst the connected nodes.
- **Note the trade-off:** Rather than needing to classify or cluster nodes with many labels, we only need binary classification of edges
- However, introduce the extra step of building tracks from classified edges

INTRO TO GRAPH NEURAL NETWORKS

GRAPH NEURAL NETWORK APPLICATIONS



Travelling Salesman Problem



Knowledge Graph Comprehension

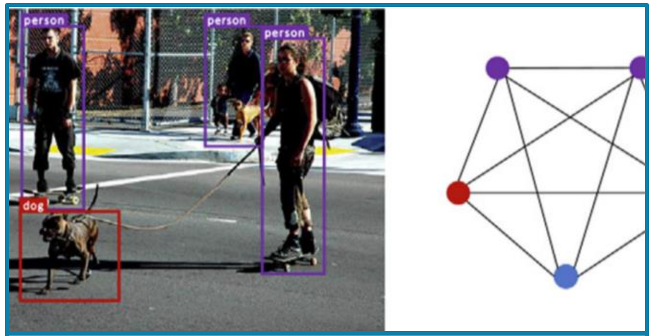
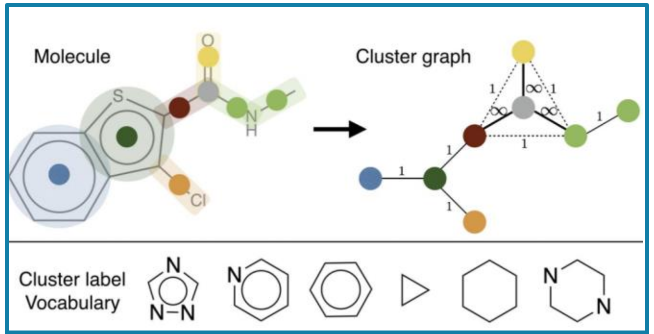
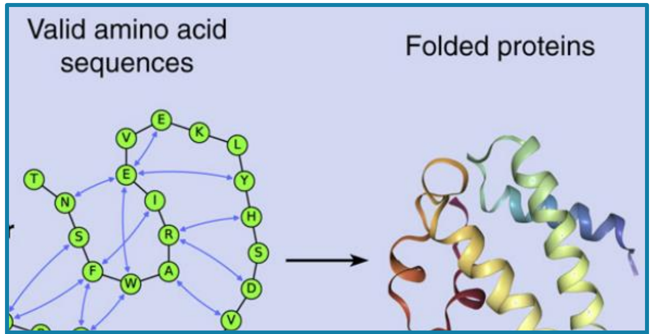


Image Comprehension

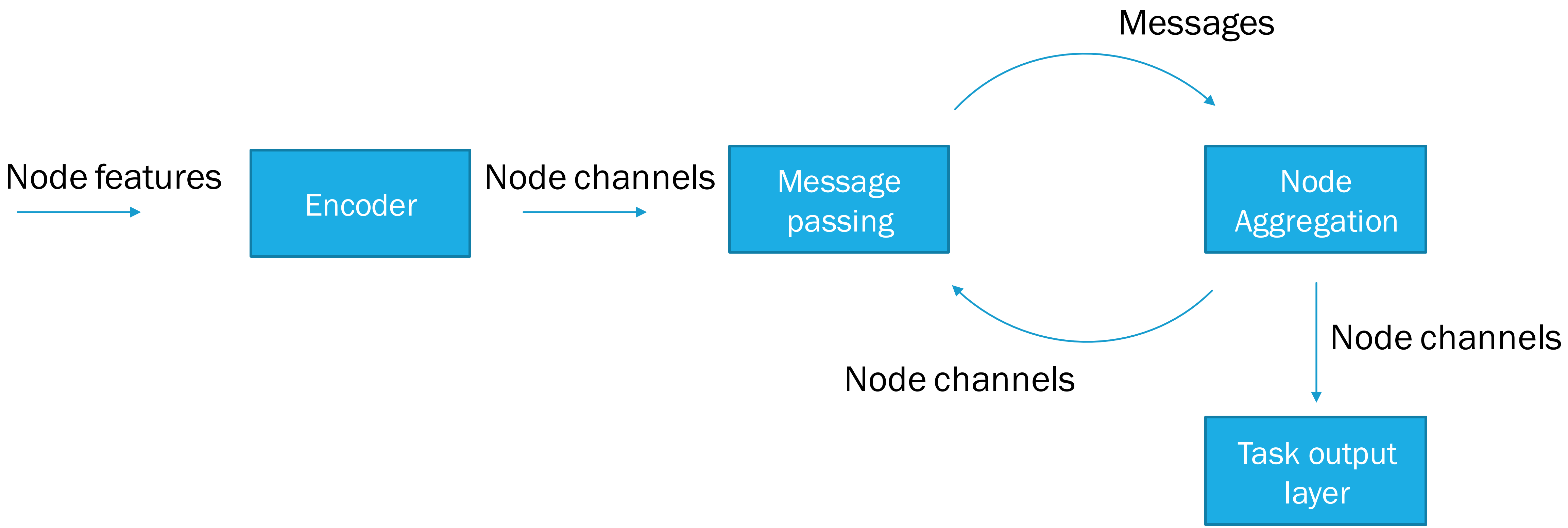


Molecular Chemistry



Protein Comprehension

GRAPH NEURAL NETWORK PROCEDURE



STEP 1: MESSAGE PASSING MECHANISM

For each node neighborhood:

a) Pass node channels through a multi-layer perceptron (MLP) encoder
b) Pass encoded channels along each edge to the central node of the neighborhood

Note: This is quite inexpensive since we store N_{nodes} for backpropagation

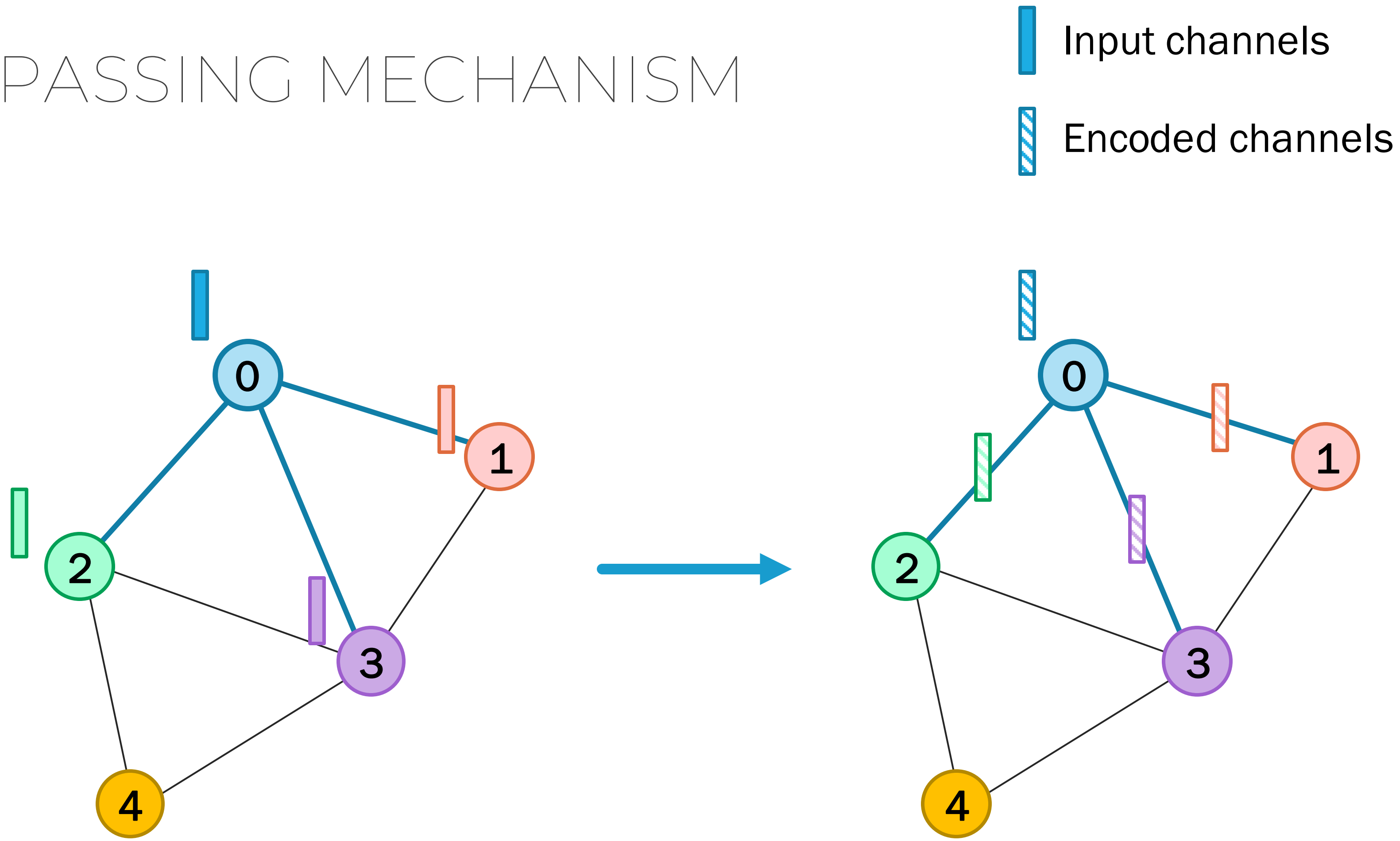


Figure inspired by Koshi et. al.

STEP 2: AGGREGATION

At each node:

Sum all messages

Note: Called *isotropic* message passing. Introduced as “Graph Convolution Network”

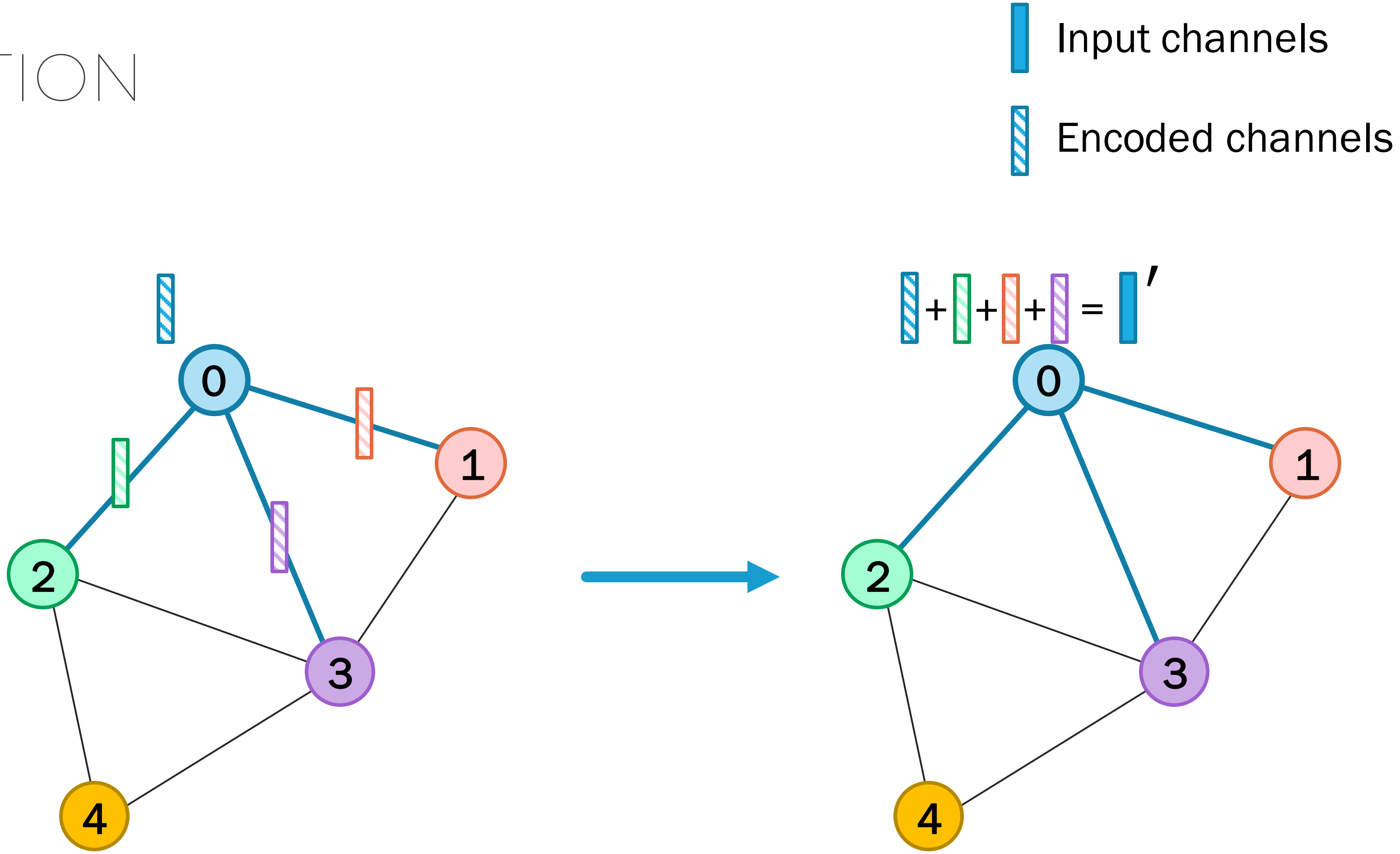
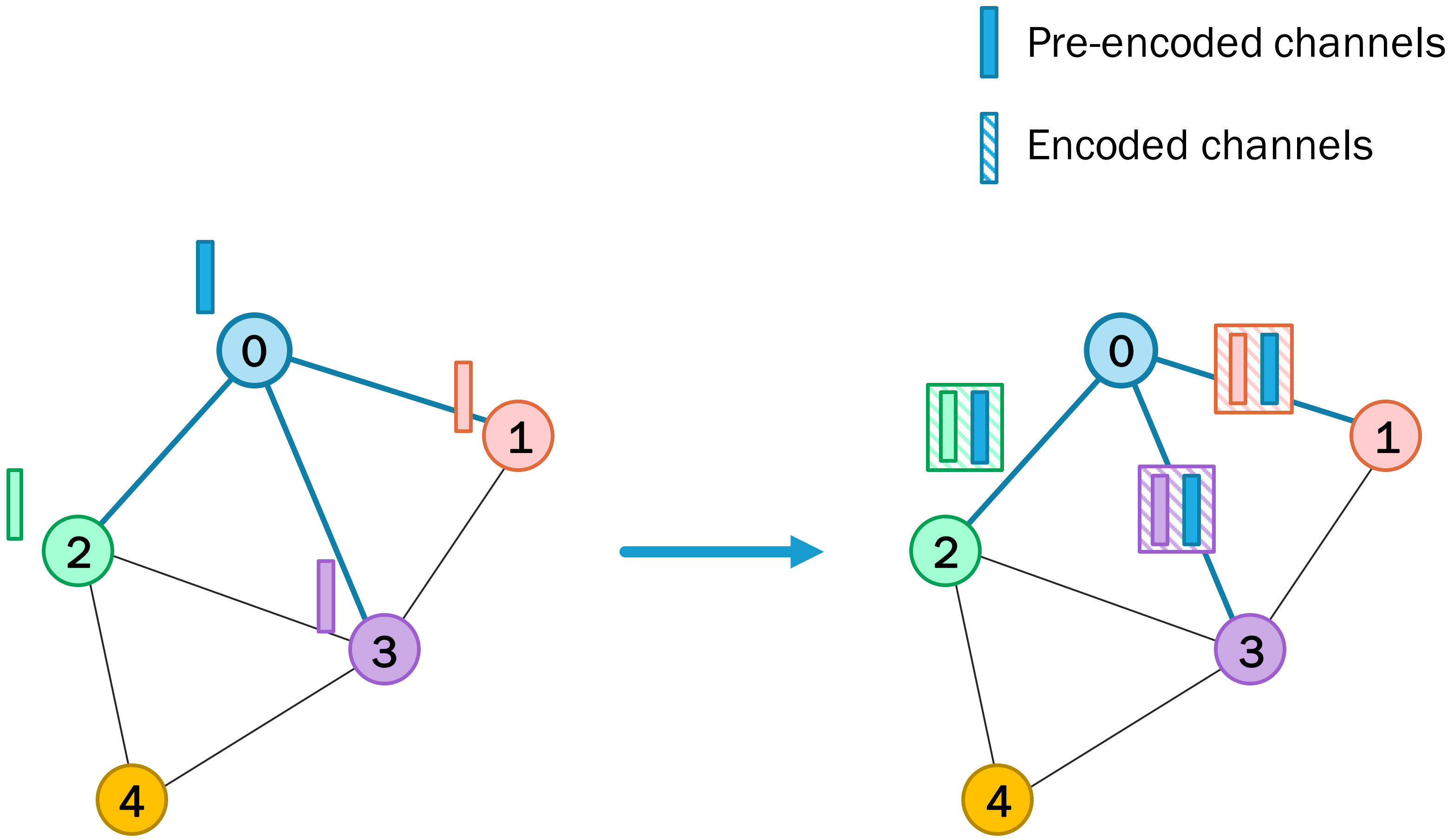


Figure inspired by [Koshi et. al.]

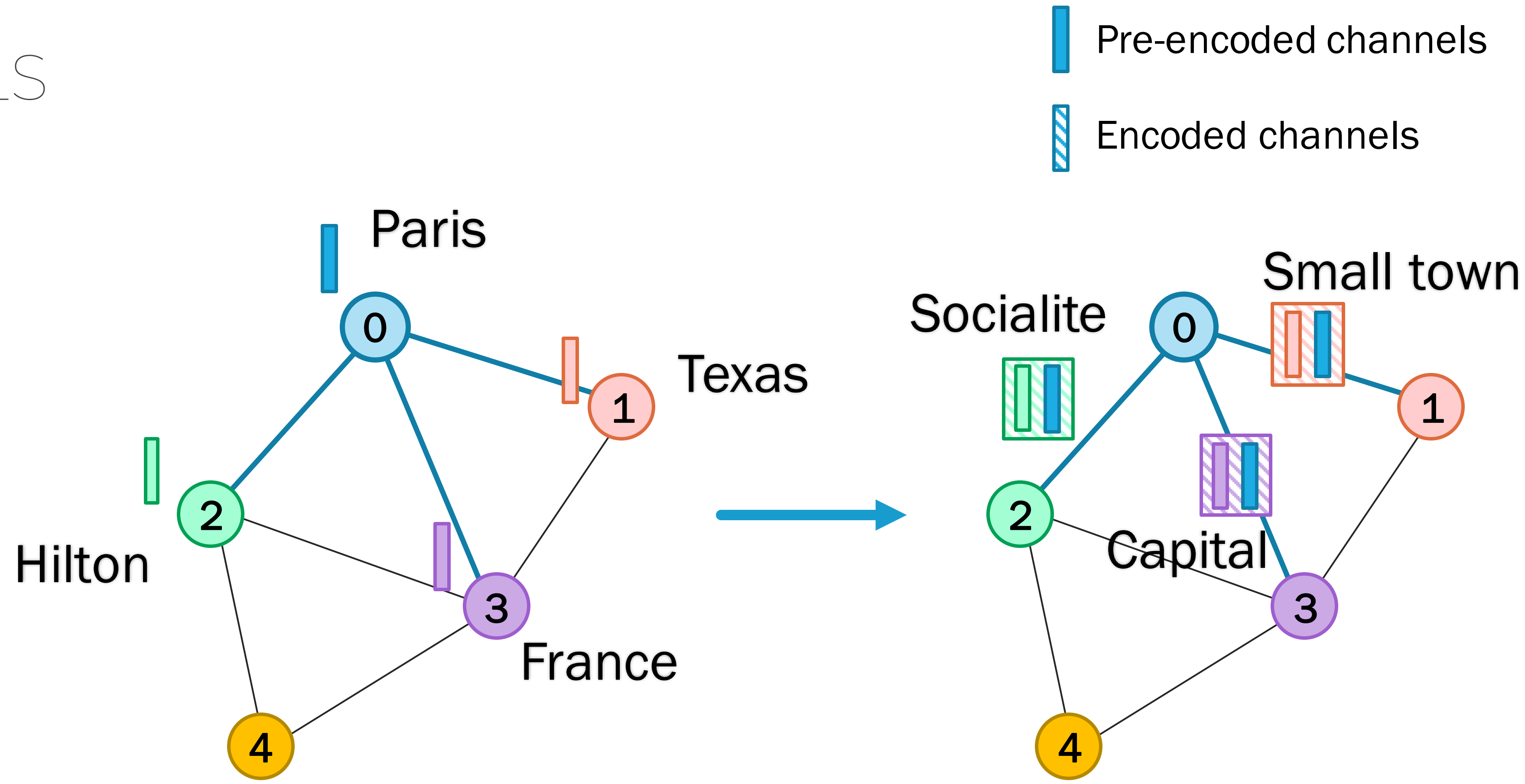
EDGE CHANNELS

- Isotropic message passing can't differentiate importance of neighbors
- *An*isotropic message passing: encode a combination of node and neighbor along each edge
- Much more expensive – now need to store N_{edges} for backpropagation
- But much more powerful

Found in "Graph Attention Network" and "Interaction Network"

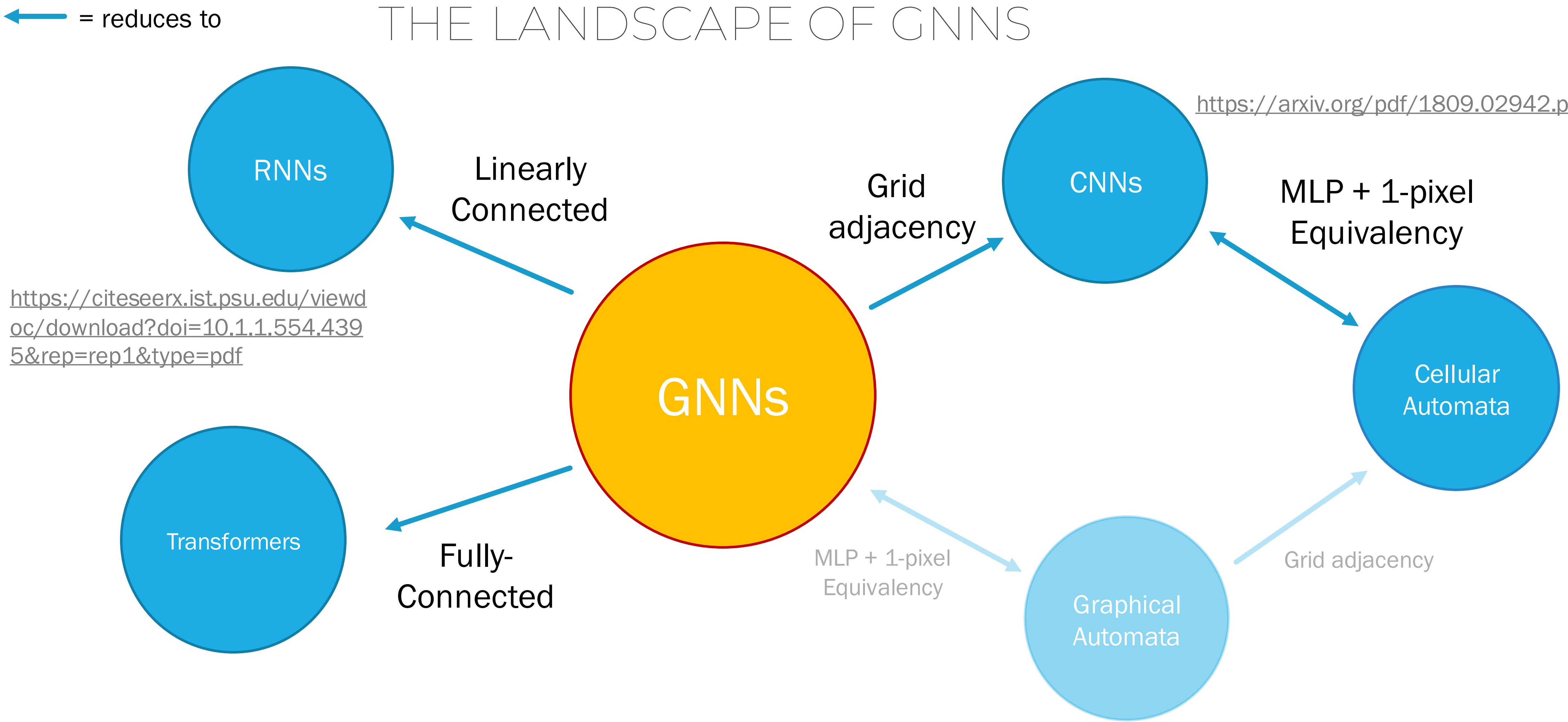


EDGE CHANNELS



- Can access *contextual* relationships

THE LANDSCAPE OF GNNS



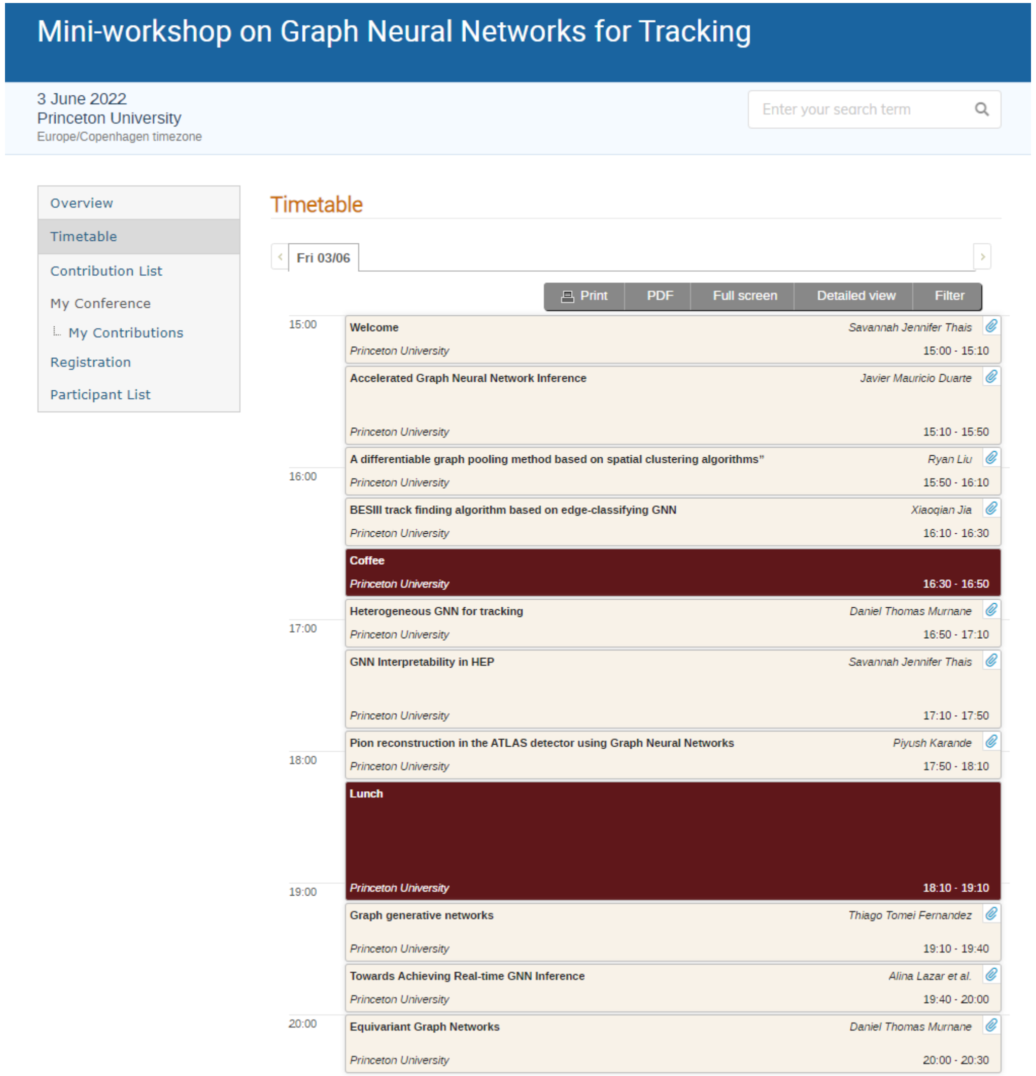
https://arxiv.org/pdf/1809.02942.pdf

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.554.4395&rep=rep1&type=pdf

https://github.com/murnanedaniel/GNN-as-Transformer-as-GNN/blob/main/0-Transformer_vs_GNN_Annotated.ipynb
https://arxiv.org/pdf/2012.09699.pdf

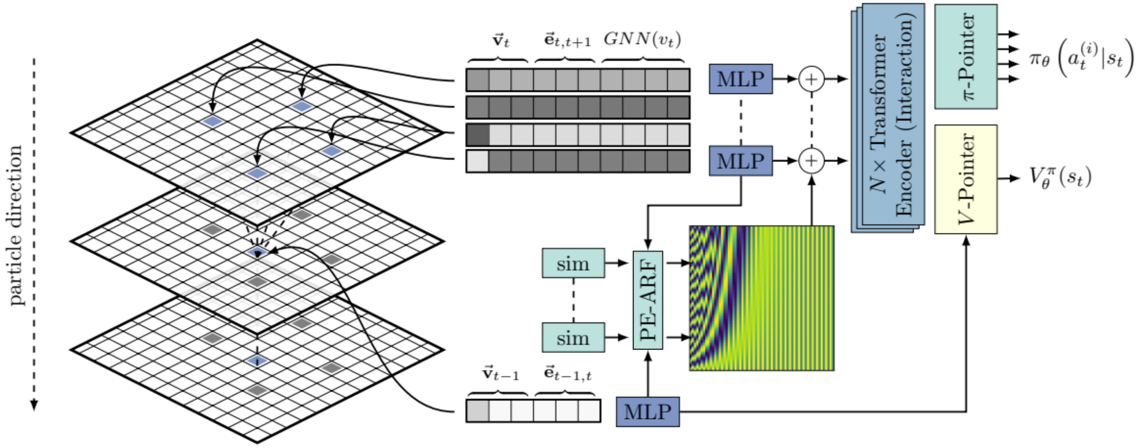
GNNS IN TRACKING

- As mentioned in the introduction, the "HepTrkX" formulation of GNN tracking is the de facto standard
- A workshop last year on GNN Tracking →
- Almost all contributions are affiliated with Exatrkx, or use a codebase forked from or motivated by the Exatrkx approach
- A variety of experiments are applying this fully supervised, edge-classification pipeline

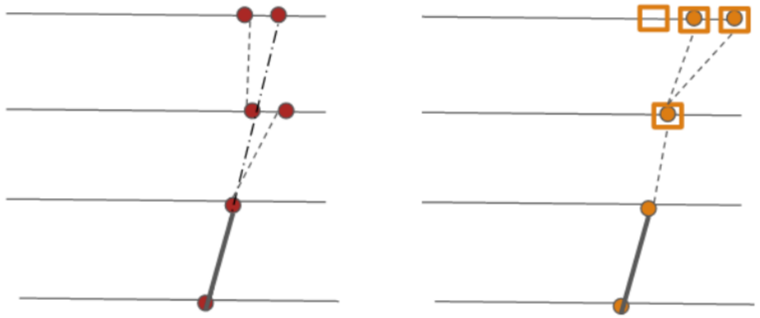


GNNS IN TRACKING

- As mentioned in the introduction, the "HepTrkX" formulation of GNN tracking is the de facto standard
- A workshop last year on GNN Tracking →
- Almost all contributions are affiliated with Exatrkx, or use a codebase forked from or motivated by the Exatrkx approach
- A variety of experiments are applying this fully supervised, edge-classification pipeline
- Another promising approach is reinforcement learning, which may or may not use deep geometric learning (i.e. graph techniques)

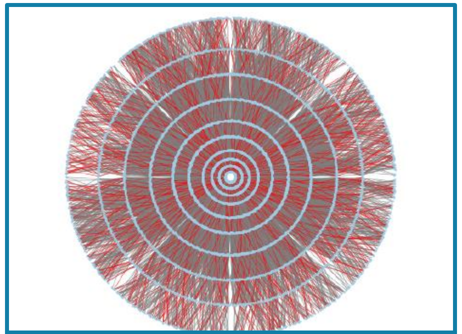


Tobias Kortus , Ralf Keidel and Nicolas R. Gauger, 2022

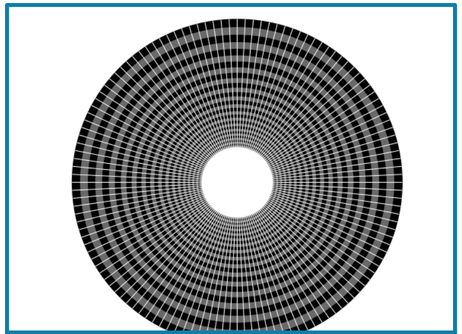


Våge, Liv CTD Proceedings 2022

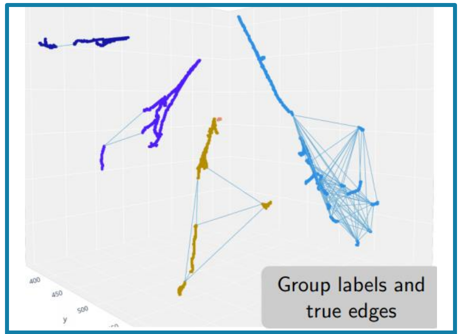
GNNS ELSEWHERE IN PARTICLE PHYSICS



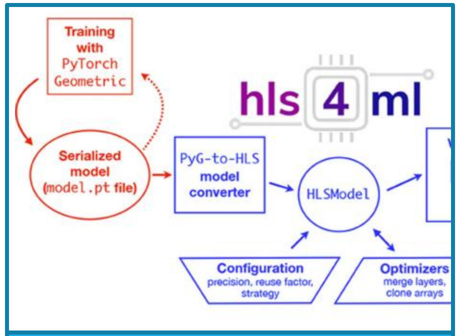
High Lumi Generic Tracking



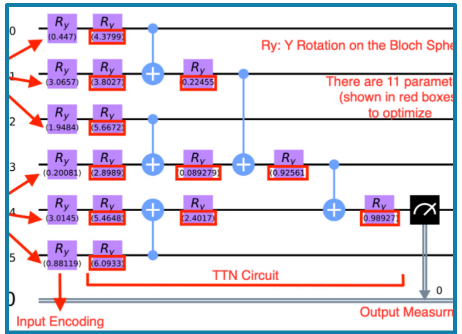
High Lumi CMS Calorimetry



LArTPC Particle Reconstruction



FPGA-based Track Reconstruction



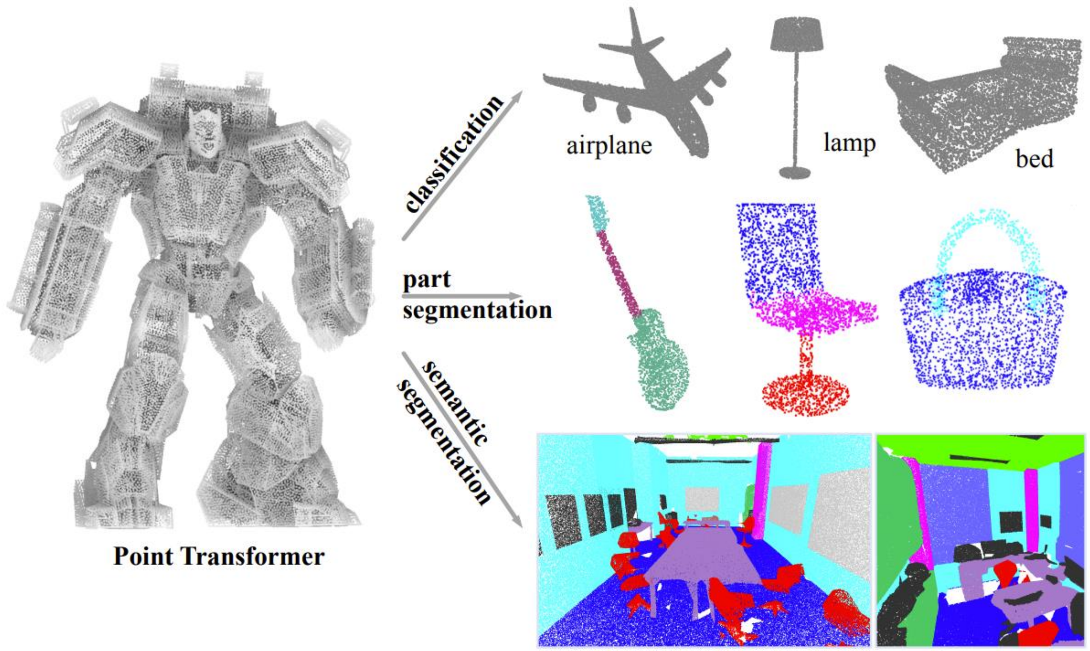
Quantum Track Reconstruction

- Very large and active field of study!
- Comprehensive review of GNNs for Track Reconstruction - arXiv:2012.01249
- White paper on progress and future of the field – arXiv:2203.12852

THE TOPOLOGY PROBLEM

TRACKING WITH GRAPHS VS. POINT CLOUD

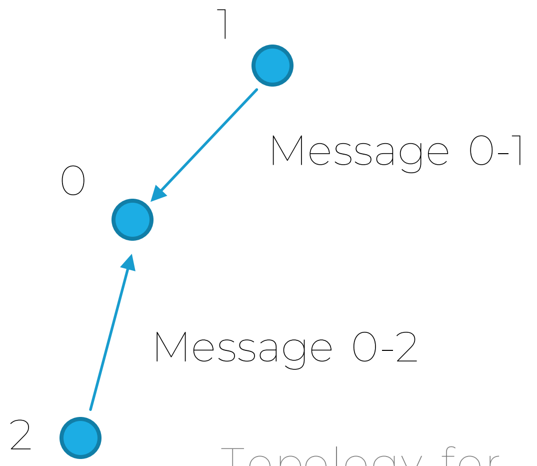
- Could use a transformer – this is now the state-of-the-art in jet tagging with GN2 tagger (ATL-PHYS-PROC-2023-017) and ParticleTransformer
- I.e. Treat as a point cloud with all-to-all connections, and compute attention between each pair
- This is tractable in a jet of $O(1k)$ clusters = $O(1m)$ attention weights
- A HL-LHC ATLAS event has $O(100k)$ clusters = $O(10b)$ attention weights…
  - Can discuss this later if there's interest!
- We thus want to impose some intuitive way of connecting hits much below $O(n^2)$



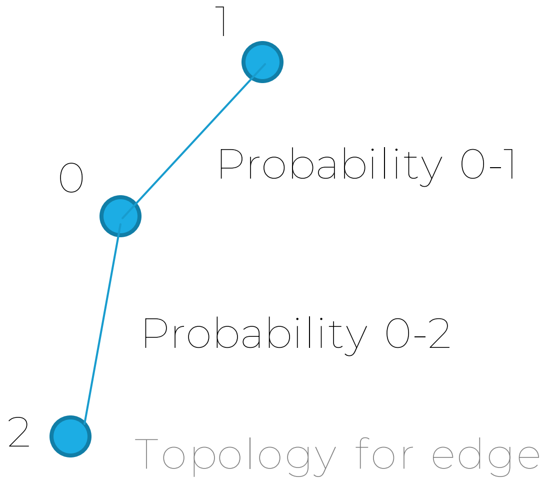
https://arxiv.org/pdf/2012.09164.pdf

TOPOLOGY FOR MESSAGE PASSING AND SEGMENTATION

- In tracking, the graph structure (“topology”) has two purposes:
  1. To pass hidden features (“messages”) from hit to hit, to minimise the loss, presumably solving an N-step combinatorial problem across tracklets
  2. As “possible connections” between hits, therefore the edges need to be classified as true of fake
- No inherent reason the two structures have to be the same. E.g. could pass messages totally randomly, but still try to classify the edges between likely hits
- For simplicity, we create a single graph that serves both purposes: Edges transmit messages, *and* they are the target of the classification model



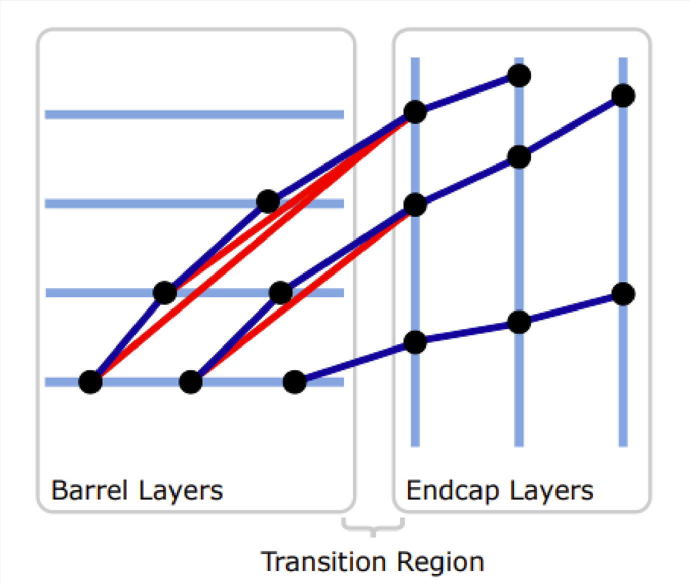
Topology for message passing



Topology for edge classification

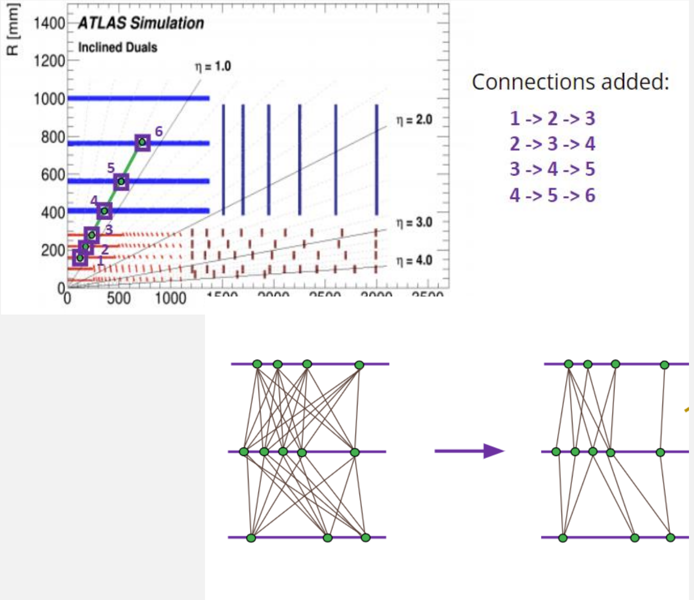
THREE WAYS TO BUILD A GRAPH

GEOMETRIC HEURISTICS



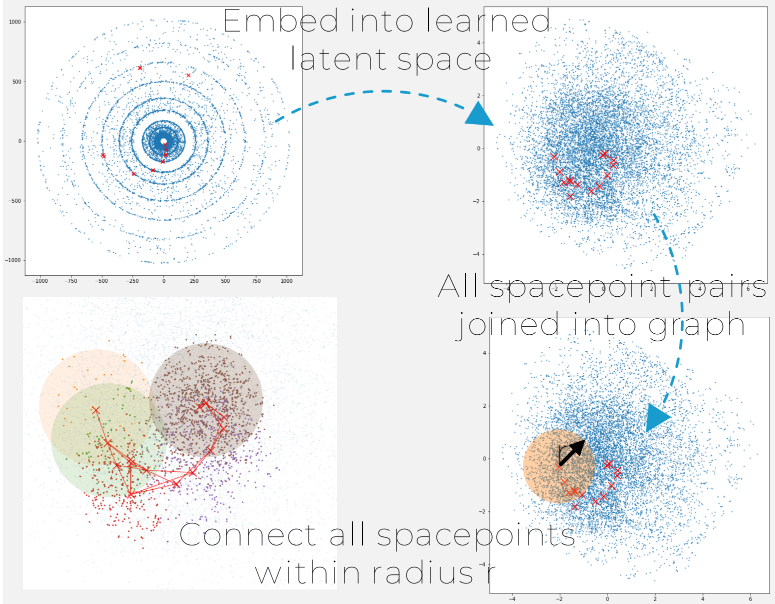
1. Consider all connections on sequential layers
2. Apply some hard geometric cuts according to heuristic knowledge of particles of interest

MODULE MAP



1. Build a module-to-module map from data
2. Apply some hard geometric cuts *for each* module-to-module possible connection

METRIC LEARNING



https://arxiv.org/abs/2103.16701

THE GNN4ITK PIPELINE

WHO IS INVOLVED?

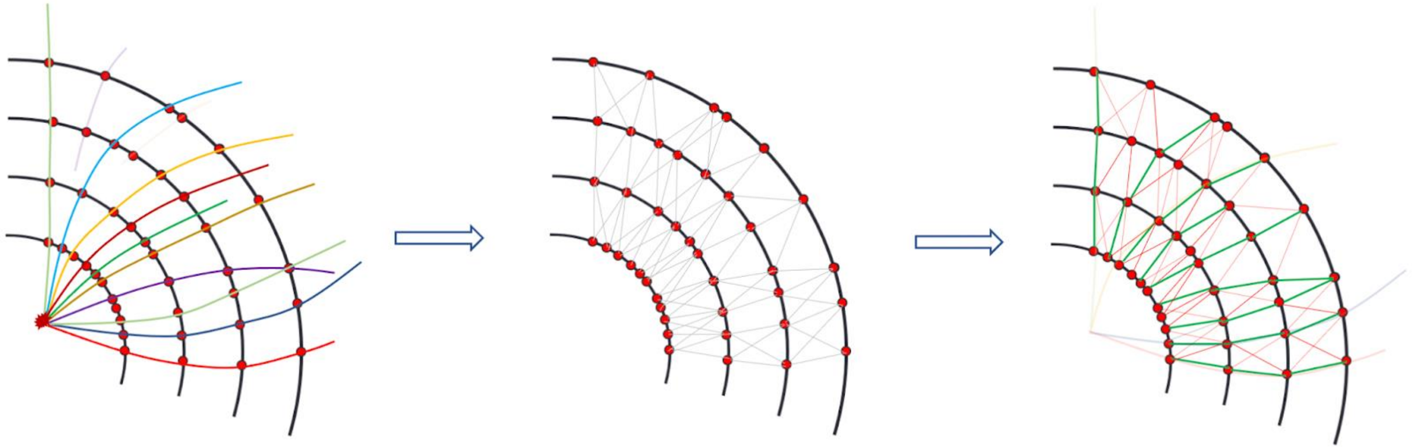
- Two groups worked on the results in this presentation, and both first tested methods on TrackML, based on the GNN-based reconstruction introduced in arxiv:1810.06111 and arxiv:2003.11603
- L2IT: Laboratoire des deux Infinis, institute based at the University of Toulouse, within the Institute of Nuclear Physics and Particle Physics
- Exa.Trkx: A DoE Office of Science-funded collaboration of LBNL, Caltech, FNAL, SLAC and a collaboration of US institutions including Cincinnati, Princeton, Urbana-Champaign, Youngstown State, and others
- Now, other groups have joined the effort, or are applying the R&D to particular applications, such as ATLAS trigger: Heidelberg University, Niels Bohr Institute, UC Irvine

GRAPH REPRESENTATION OF AN EVENT

- The goal of track reconstruction:

Given set of hits in a detector from particles, assign label(s) to each hit.

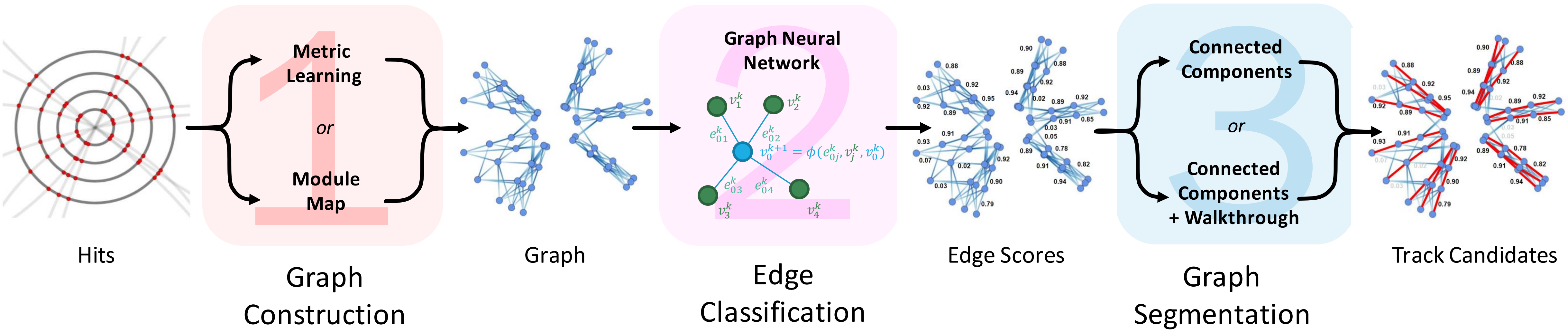
Perfect classification: All hits from a particle (*and only those hits*) share the same label



- What does it mean to represent an event with a graph?
  - Treat each hit as a **node**
  - A node can have features (e.g. position, energy deposit, etc.)
  - Nodes can be connected by **edges**, that represent the possibility of belonging to the same track
- Goal: **Use ML and/or graph techniques to segment or cluster the nodes to match particle tracks**
- **Proof-of-concept:** TrackML community challenge dataset with simplified simulation

PIPELINE OVERVIEW

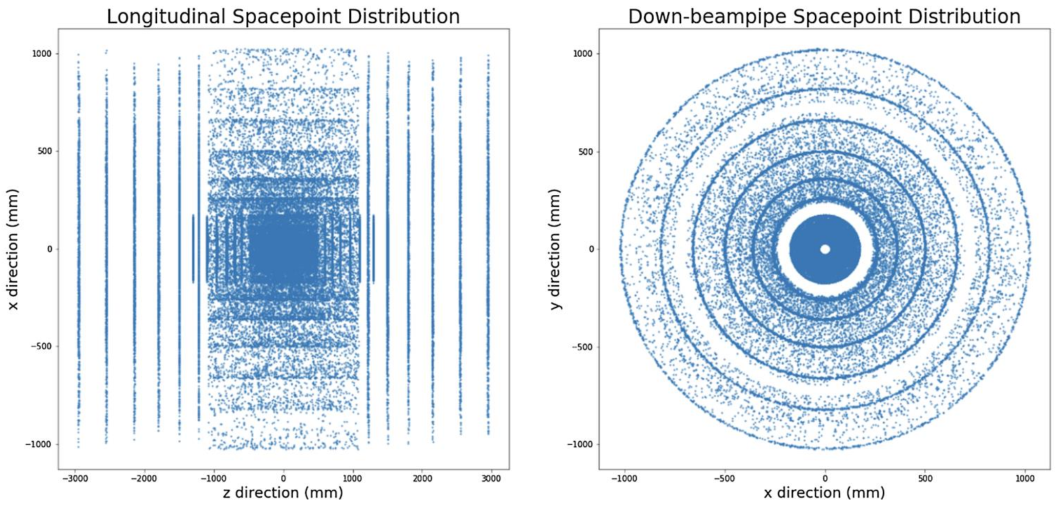
- Current pipeline of the L2IT-Exatrkx collaborative effort
- Each stage offers multiple independent choices, depending on hardware and time constraints



DATASETS

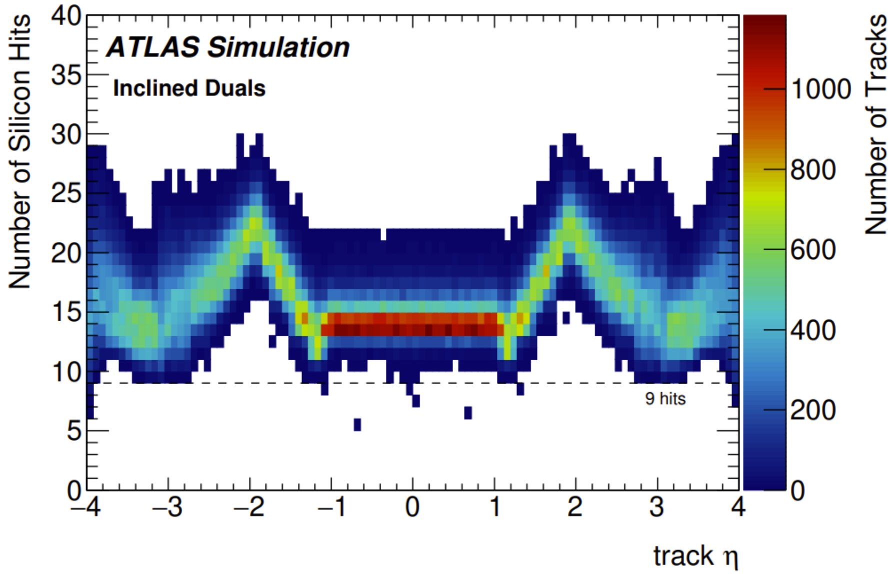
- Two datasets used to study this pipeline. For absolute clarity, when citing a result specific to one dataset, will place the badge of TrackML or ATLAS ITk on slide:

TrackML



- Mean number of spacepoints: 110k
- Simplified simulation: No secondaries and optimistic charge information

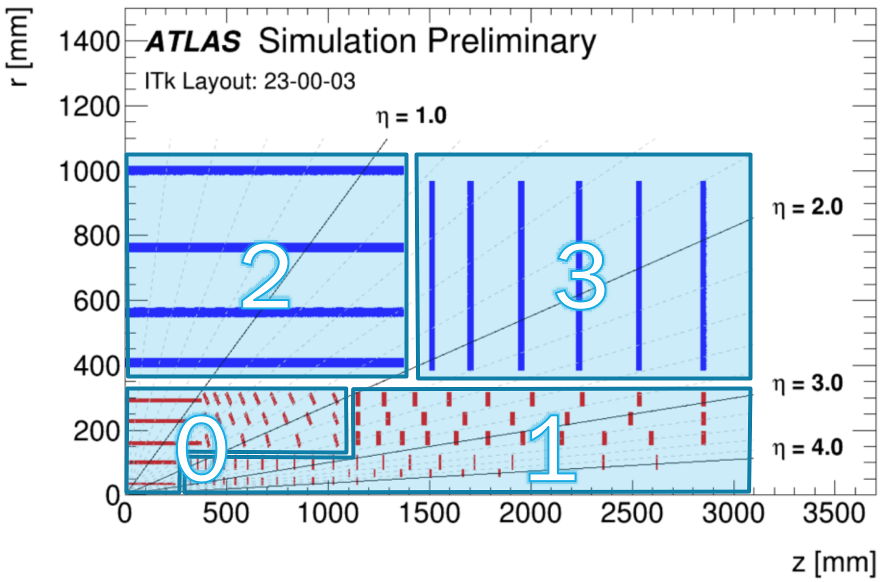
ATLAS ITk



- Mean number of spacepoints: 310k
- Full simulation

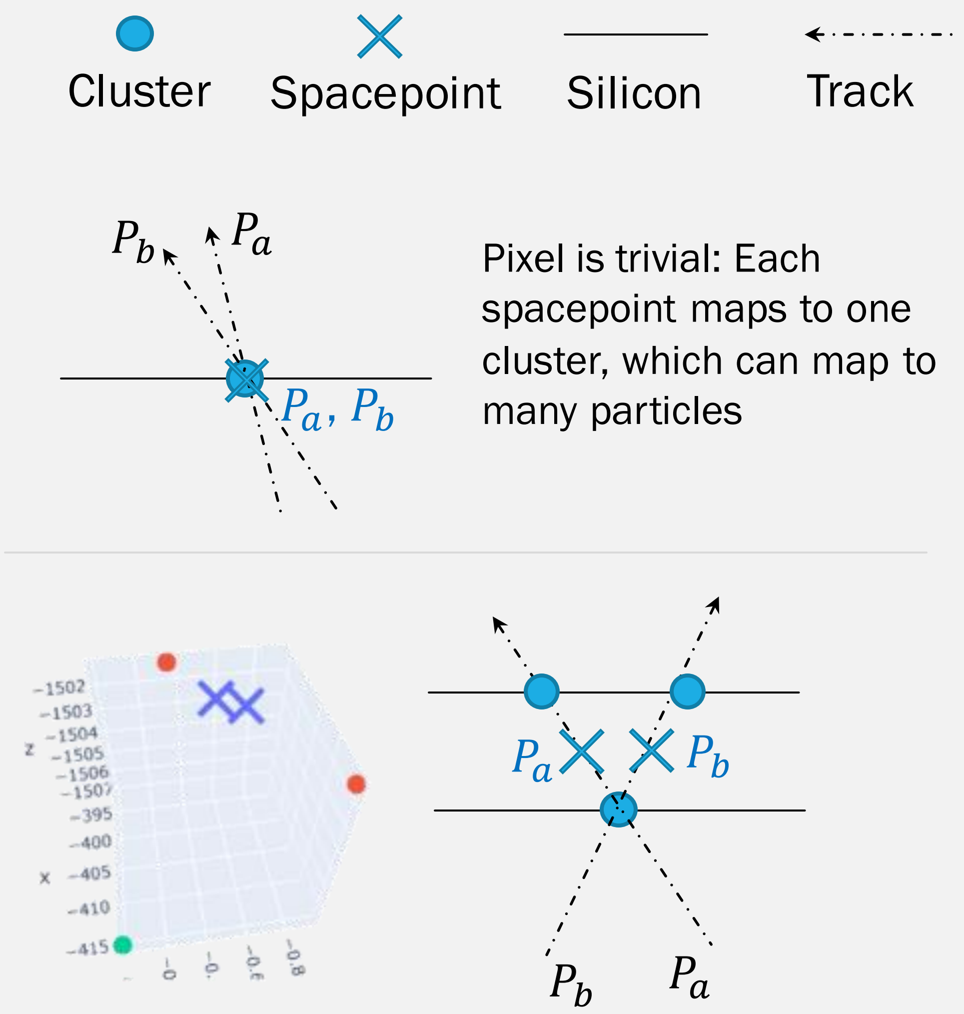
ATLAS ITK GEOMETRY

- Generation script* using Athena, $t\bar{t}$ at $\mu = \langle 200 \rangle$: with statistics dominated by soft interactions
- ITk consists of barrel and endcap, each with pixels and strips:



0: Pixel barrel
1: Pixel endcap
2: Strip barrel
3: Strip endcap

- Spacepoints (3D representations of track hits) are defined depending on strip or pixel:



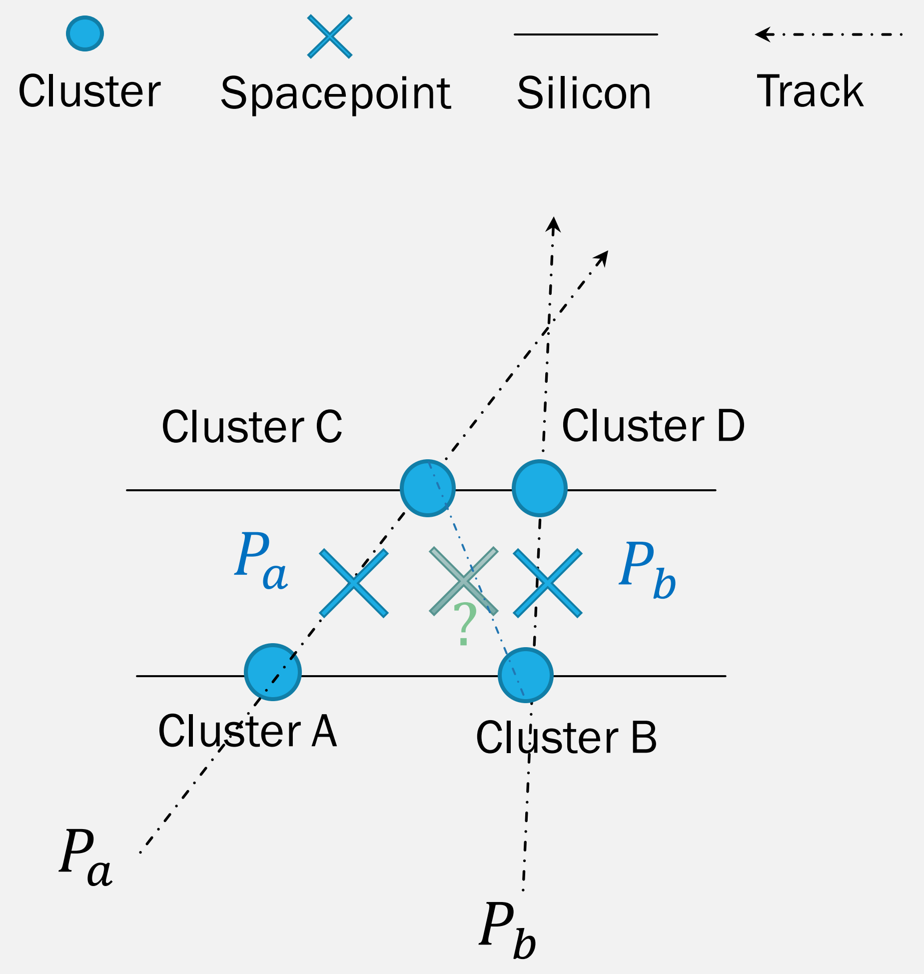
Pixel is trivial: Each spacepoint maps to one cluster, which can map to many particles

Strip: Each spacepoint maps to two clusters – one on either side of strip, which can map to many particles

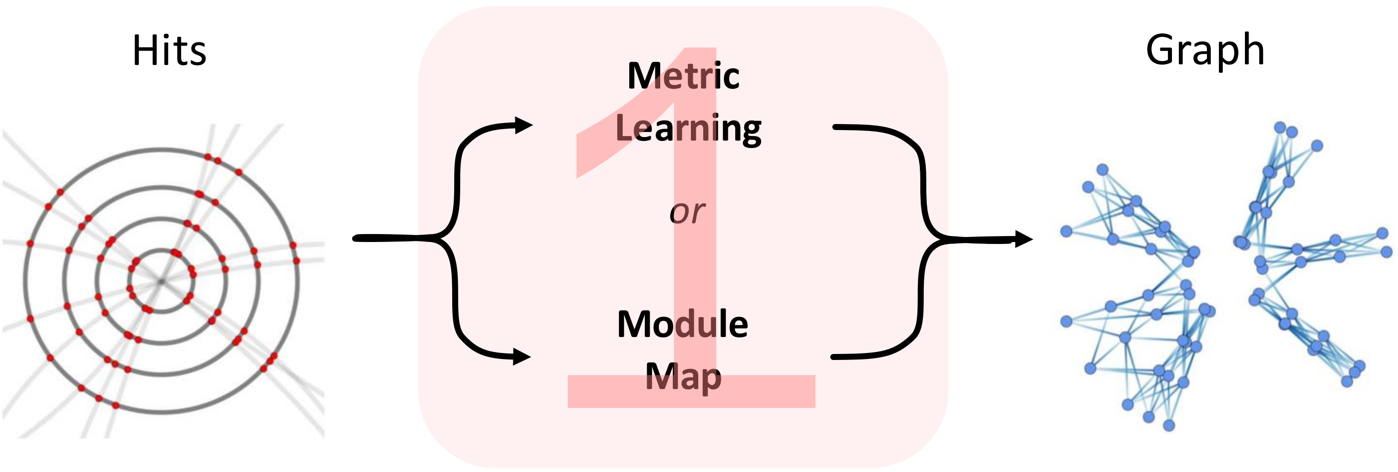
*Thanks Noemi Calace

ATLAS ITK GEOMETRY

- Fiducial particles are charged, with $\boldsymbol{\eta} \in [-4, 4]$, and production radius < 260mm
- Each event has O(15k) fiducial particles, O(300k) spacepoints
- We define background spacepoints as including:
  - Those left by non-fiducial or intermediate particles (i.e. any particle barcodes not retained during simulation), or
  - Those mis-constructed in the strip regions as ghost spacepoints
- An event has O(170k) background spacepoints

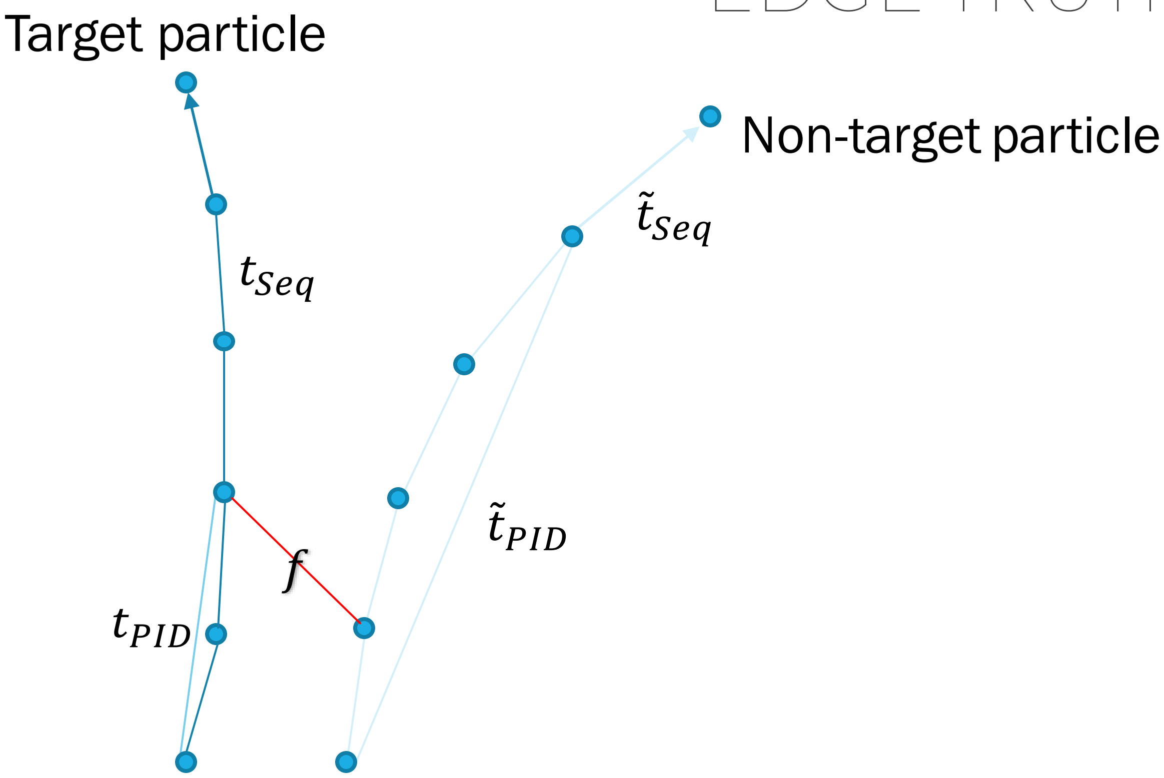


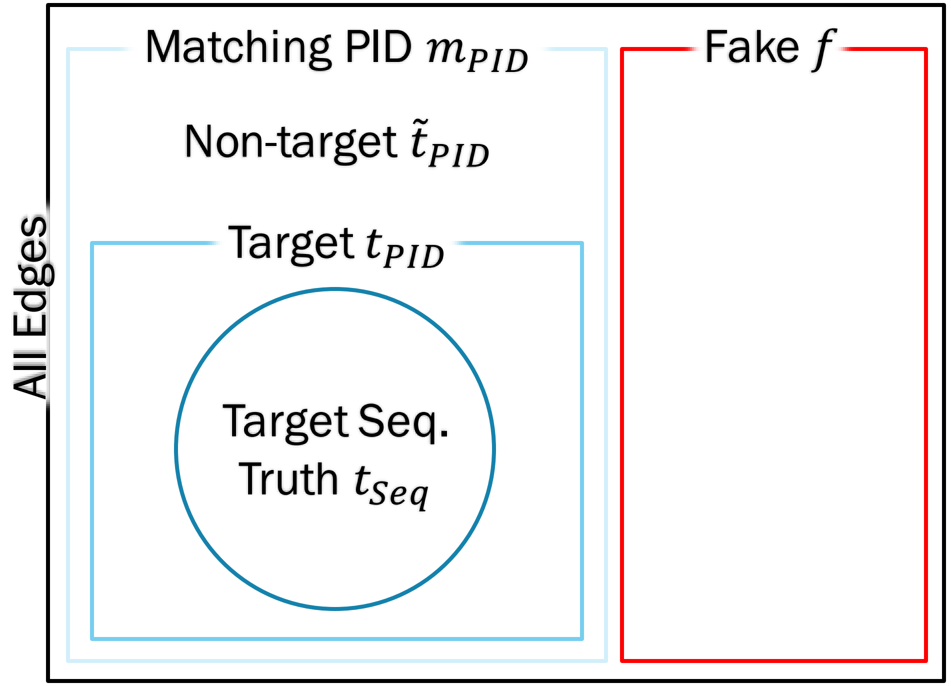
Ghost spacepoint: Incorrectly constructed from clusters left by different particles



Graph Construction

EDGE TRUTH DEFINITIONS





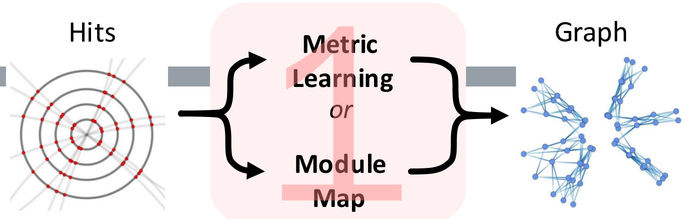
Target particle:

- $p_T > 1$ GeV, and
- At least 3 SP on different modules, and
- Primary

Therefore, define efficiency and purity (note that we mask out sequential non-target) for a graph with edges e

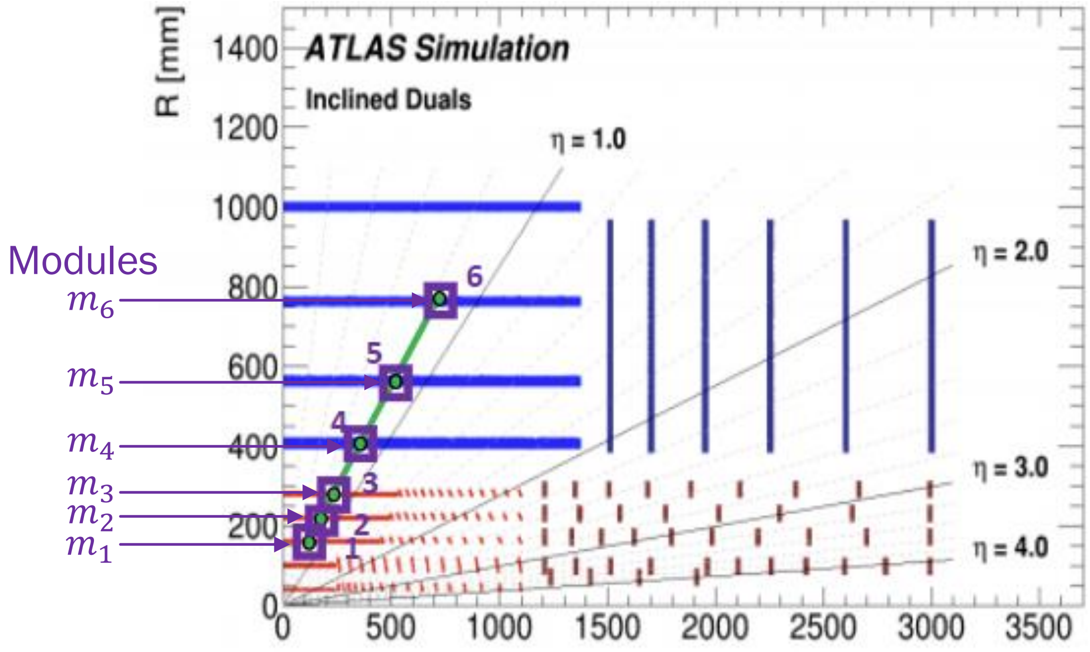
$$\text{Efficiency} = \frac{|e \cap t_{Seq}|}{|t_{Seq}|}, \quad \text{Purity} = \frac{|e \cap t_{Seq} - \tilde{t}_{Seq}|}{|e - \tilde{t}_{Seq}|}$$

MODULE MAP - DOUBLETS

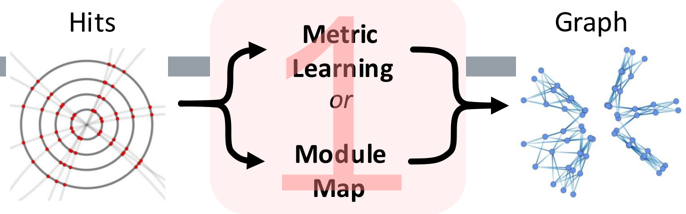


Graph Construction

- The idea: Build a map of detector modules, where a connection *from* module A *to* module B means that at least one true track has passed sequentially through A to B
- Step 1: Build all combinations of sequential doublets for an event, register an A-to-B entry if a doublet passes through. O(90k) events used to build these combinations
- Step 2: For *each* A-to-B entry, also register/update the max and min values of a set of geometric observables. Apply these cuts when building the graph in inference



$$\text{Map} = \{m_1 : m_2, m_2 : m_3, \ldots, m_5 : m_6\}$$

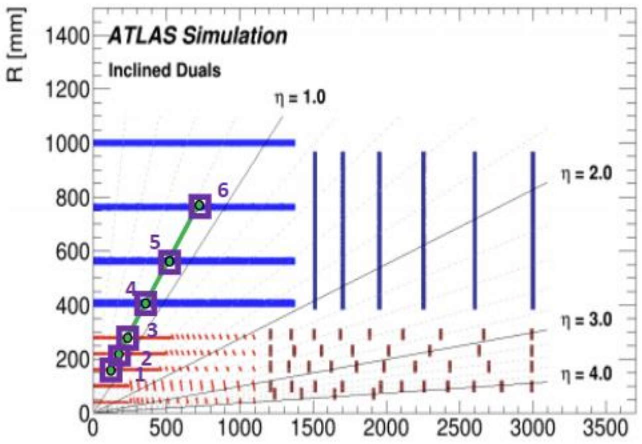


Graph Construction

MODULE MAP – TRIPLETS

- **The idea**: Build a map of detector modules, where a connection *from* module A *to* module B *to* module C means that at least one true track has passed sequentially through A to B to C
- Step 1: Build all combinations of sequential triplets for an event, register an A-to-B-to-C entry if a triplet passes through
- Step 2: For *each* A-to-B-to-C entry, also register/update the max and min values of a set of geometric observables. Apply these cuts when building the graph in inference

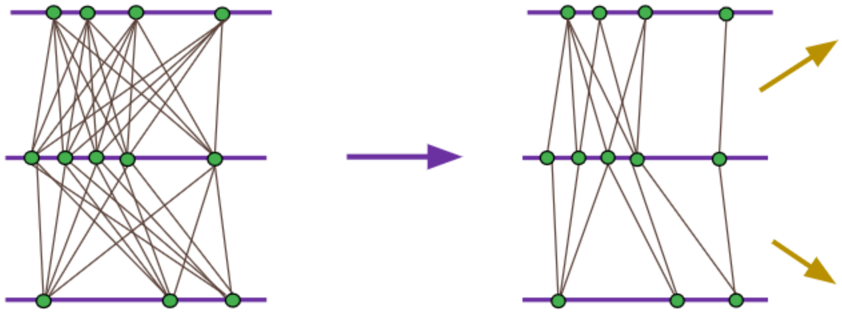
Step 1



Connections added:

1 -> 2 -> 3
2 -> 3 -> 4
3 -> 4 -> 5
4 -> 5 -> 6

Step 2



- $z_0 = z_{h1} - r_{h1} \times \left(\frac{\Delta z}{\Delta r}\right)$
- $\phi_{slope} = \frac{\Delta\phi}{\Delta r}$
- $\Delta\phi = \phi_{h2} - \phi_{h1}$
- $\Delta\eta = \eta_{h2} - \eta_{h1}$

- $z_0 = z_{h1} - r_{h1} \times \left(\frac{\Delta z}{\Delta r}\right)$
- $\phi_{slope} = \frac{\Delta\phi}{\Delta r}$
- $\Delta\phi = \phi_{h2} - \phi_{h1}$
- $\Delta\eta = \eta_{h2} - \eta_{h1}$

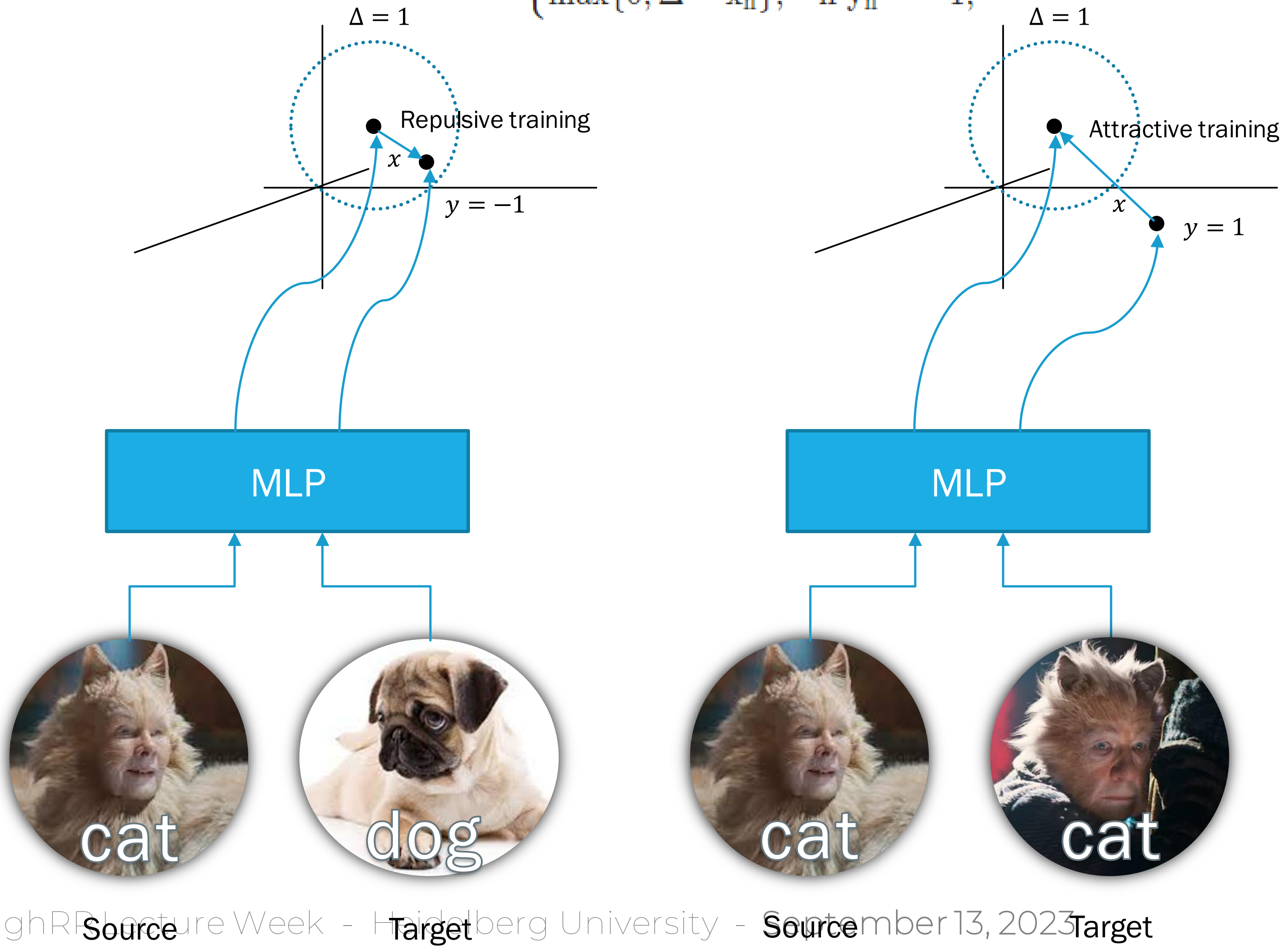
- $\Delta\frac{\Delta y}{\Delta x} = \frac{\Delta y_{12}}{\Delta x_{12}} - \frac{\Delta y_{23}}{\Delta x_{23}}$
- $\Delta\frac{\Delta z}{\Delta r} = \frac{\Delta z_{12}}{\Delta r_{12}} - \frac{\Delta z_{23}}{\Delta r_{23}}$

METRIC LEARNING INTUITION

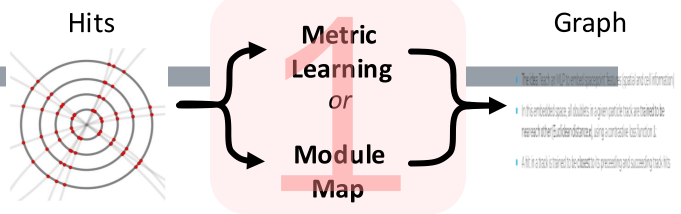
- Encode / embed input into N-dimensional space
- Reward (low loss) matching pairs within unit distance
- Punish (high loss) mismatching pairs within unit distance
- Repeat for many pairs

“Contrastive” hinge loss

$$l_n = \begin{cases} x_n, & \text{if } y_n = 1, \\ \max\{0, \Delta - x_n\}, & \text{if } y_n = -1, \end{cases}$$



METRIC LEARNING



Graph Construction

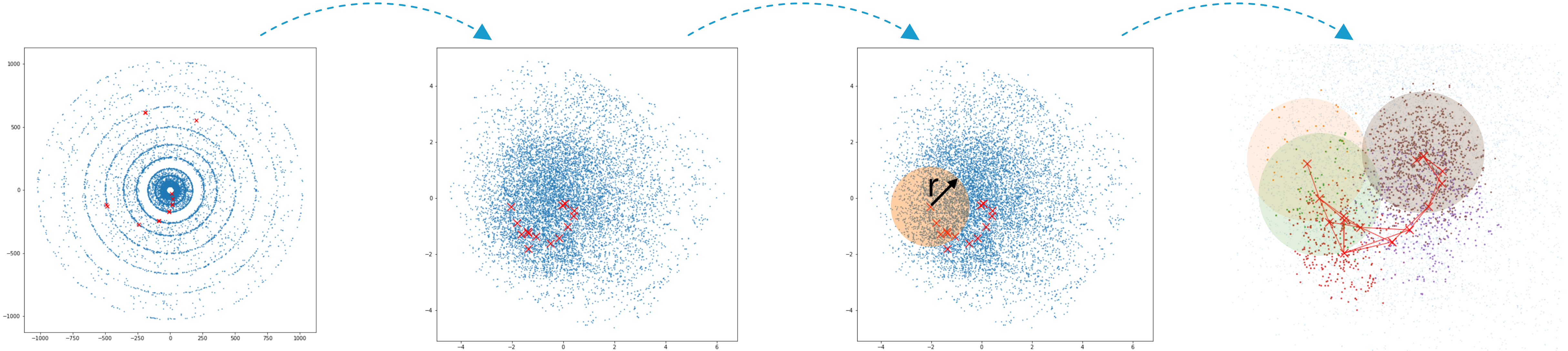
- The idea: Teach an MLP to embed spacepoint features (spatial and cell information)
- In this embedded space, all doublets in a given particle track are trained to be near each other (Euclidean distance $\boldsymbol{x}$), using a contrastive loss function L:
- A hit in a track is trained to be closest to its preceeding and succeeding track hits

$$L = \begin{cases} x, & \text{if true pair} \\ \max(0, r - x), & \text{if false pair} \end{cases}$$

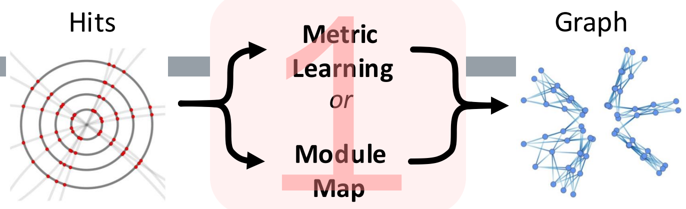
Embed into learned latent space

Connect all spacepoints within radius r

All spacepoint pairs joined into graph



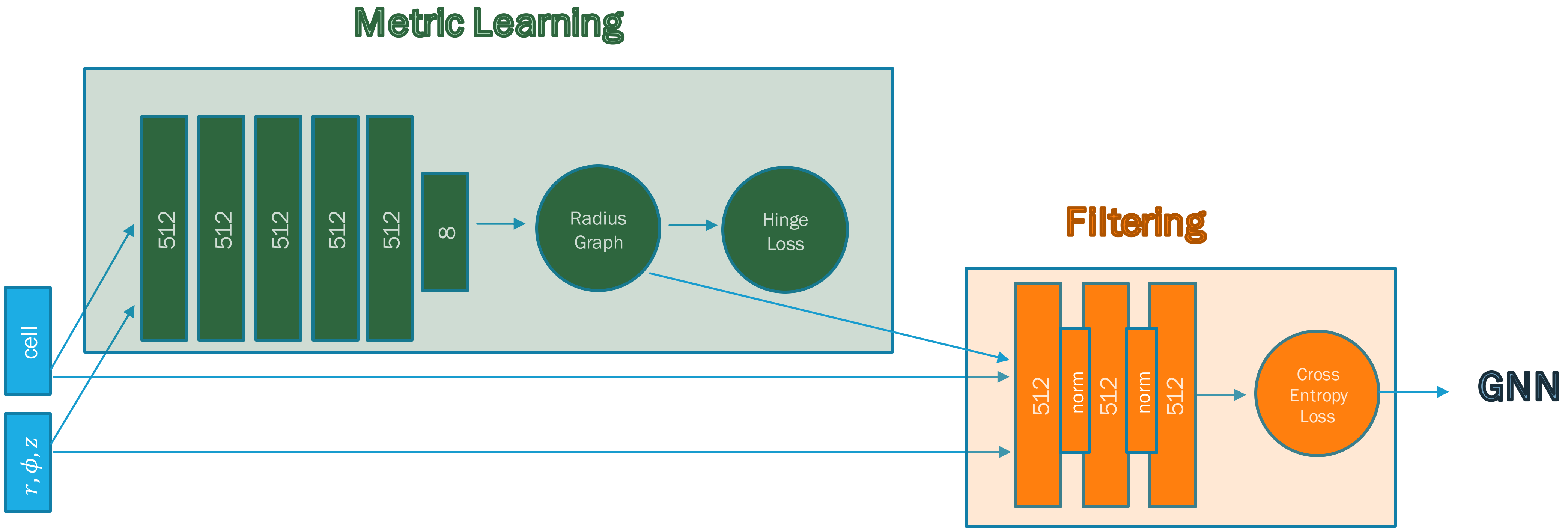
METRIC LEARNING - FILTERING



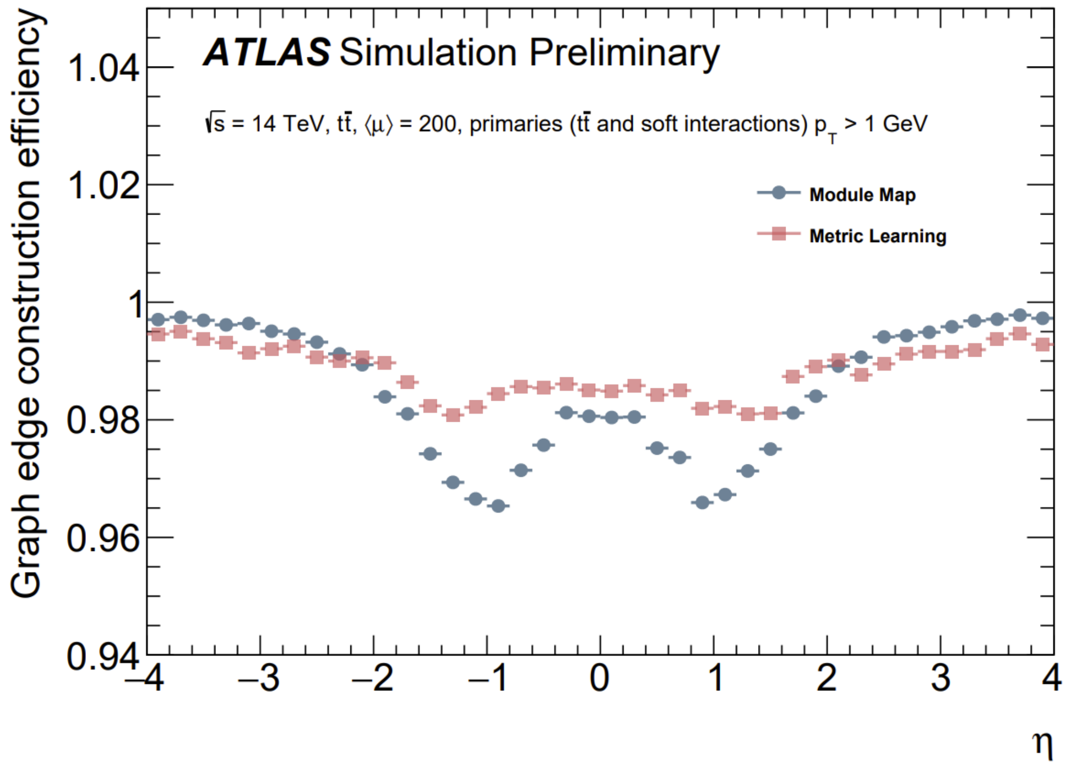
Graph Construction

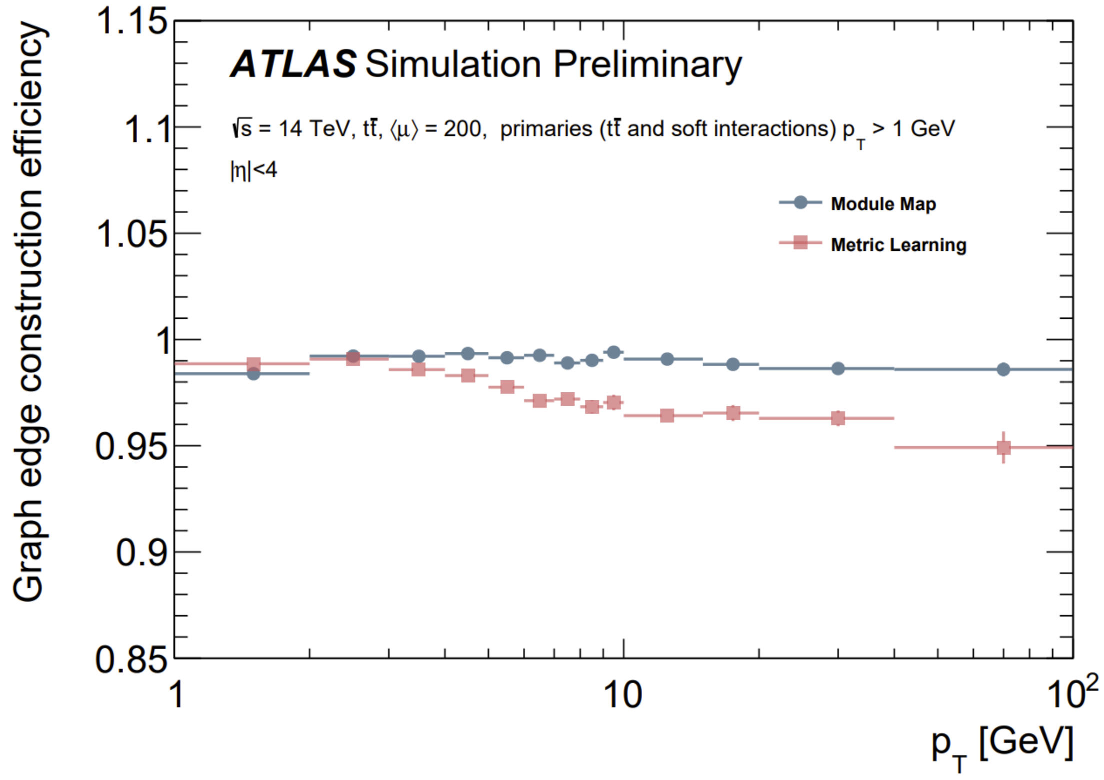
ATLAS ITk

- Output graph of metric learning is impure: 0.2%
- Can pass edges through a simple MLP filter to filter out the easy fakes
- Improves purity to 2%, so graph can be trained entirely on a single GPU



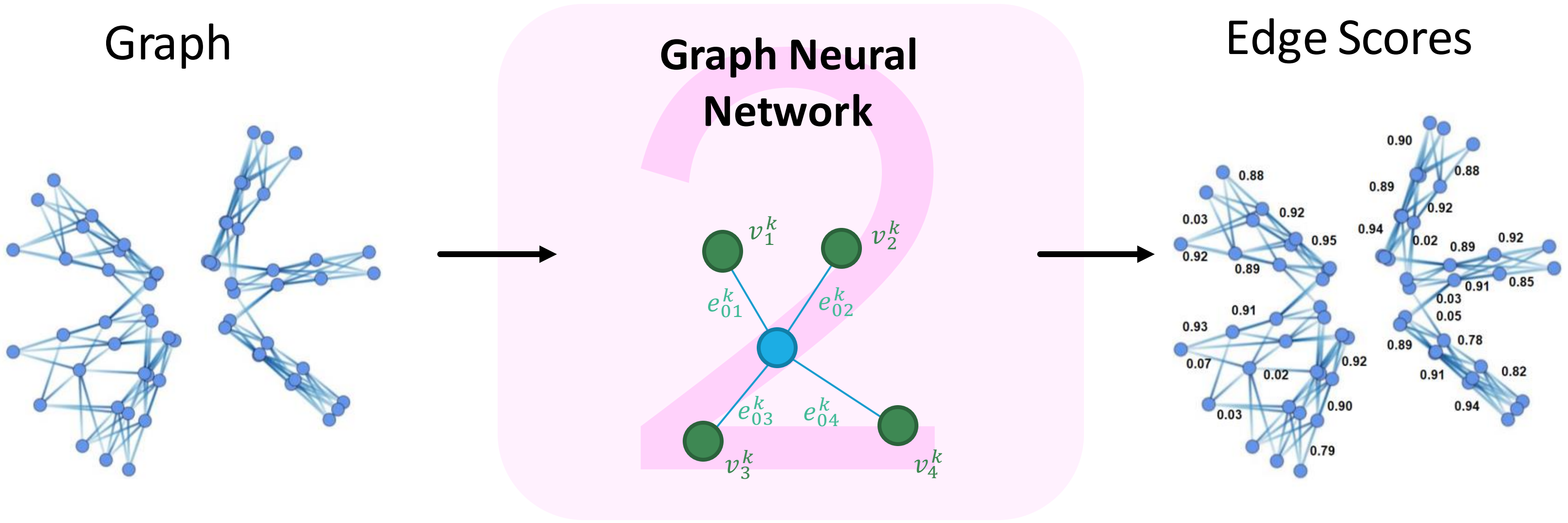
GRAPH CONSTRUCTION RESULTS





- Drop in efficiency at low η due to poor barrel strip resolution (will discuss further!)

- Drop in efficiency at high p_T due to low training statistics



Edge Labeling

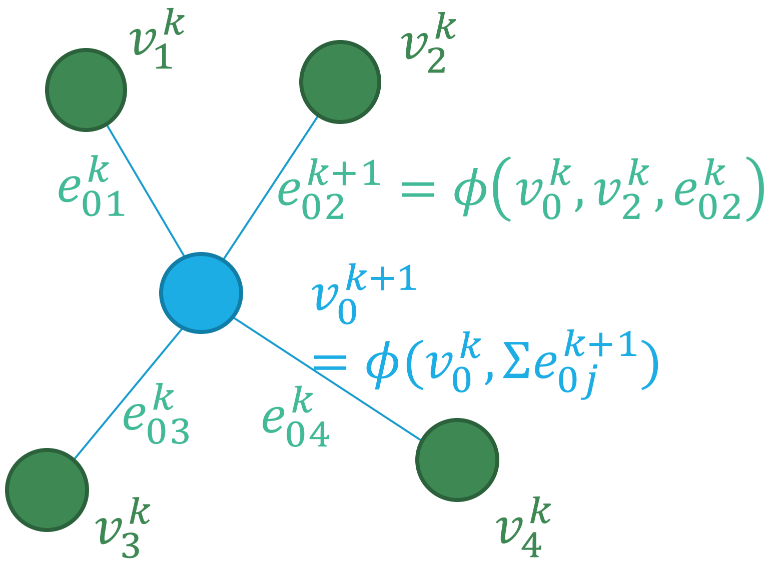
EDGE CLASSIFICATION WITH GRAPH NEURAL NETWORK

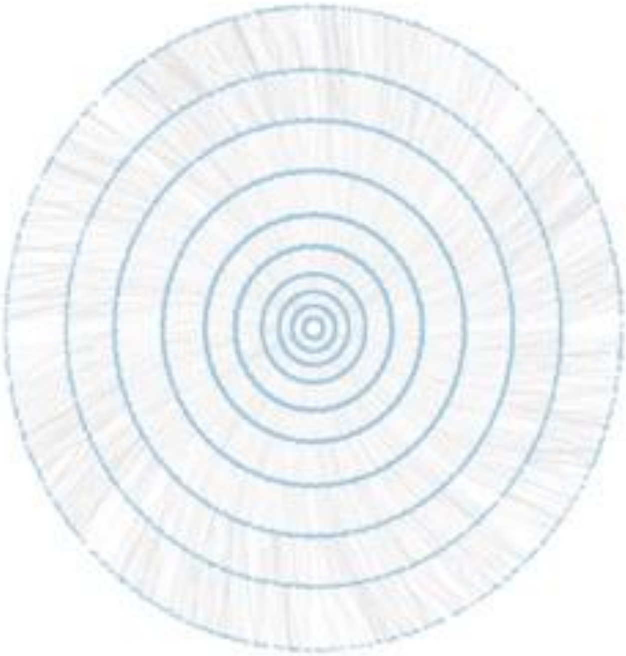
1. Node features (spatial position) are encoded
2. Encoded features are concatenated and encoded to create edge features
3. Edge features are aggregated around nodes to create next round of encoded node features (i.e. message passing)
4. Each iteration of message passing improves discrimination power

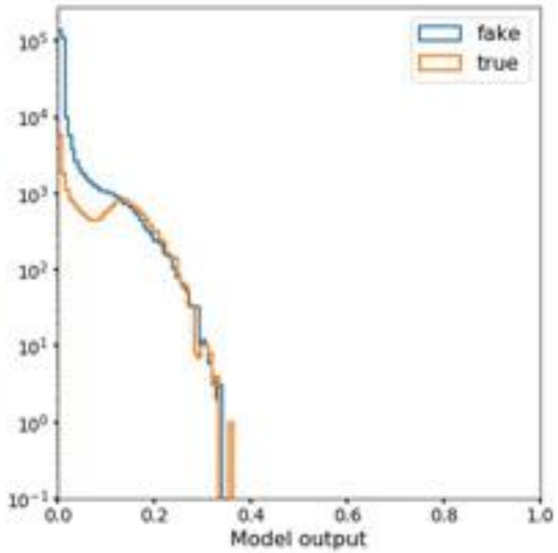
v_i^k node features
e_{ij}^k edge features
at iteration k

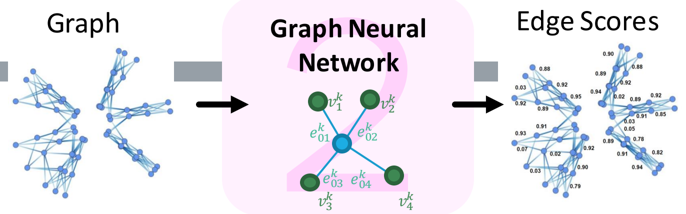
Battaglia, Peter, et al. "Interaction networks for learning about objects, relations and physics." 2016.

INTERACTION NETWORK









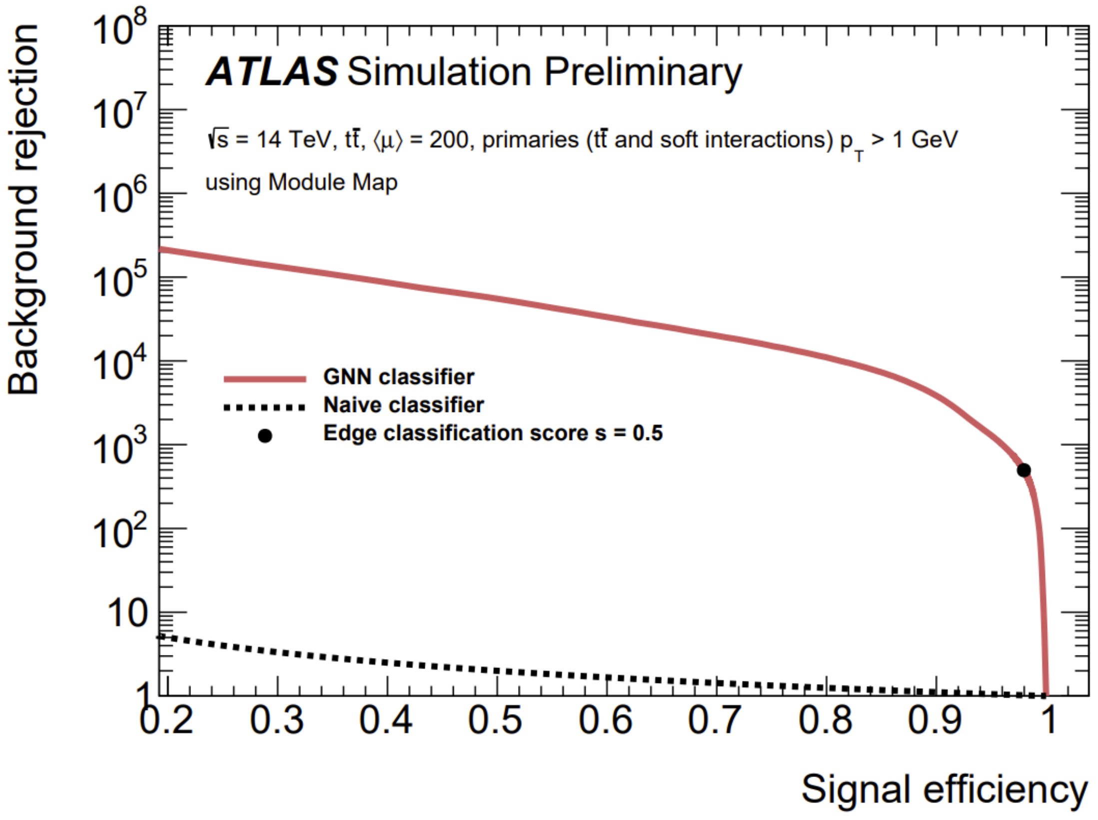
Edge Labeling

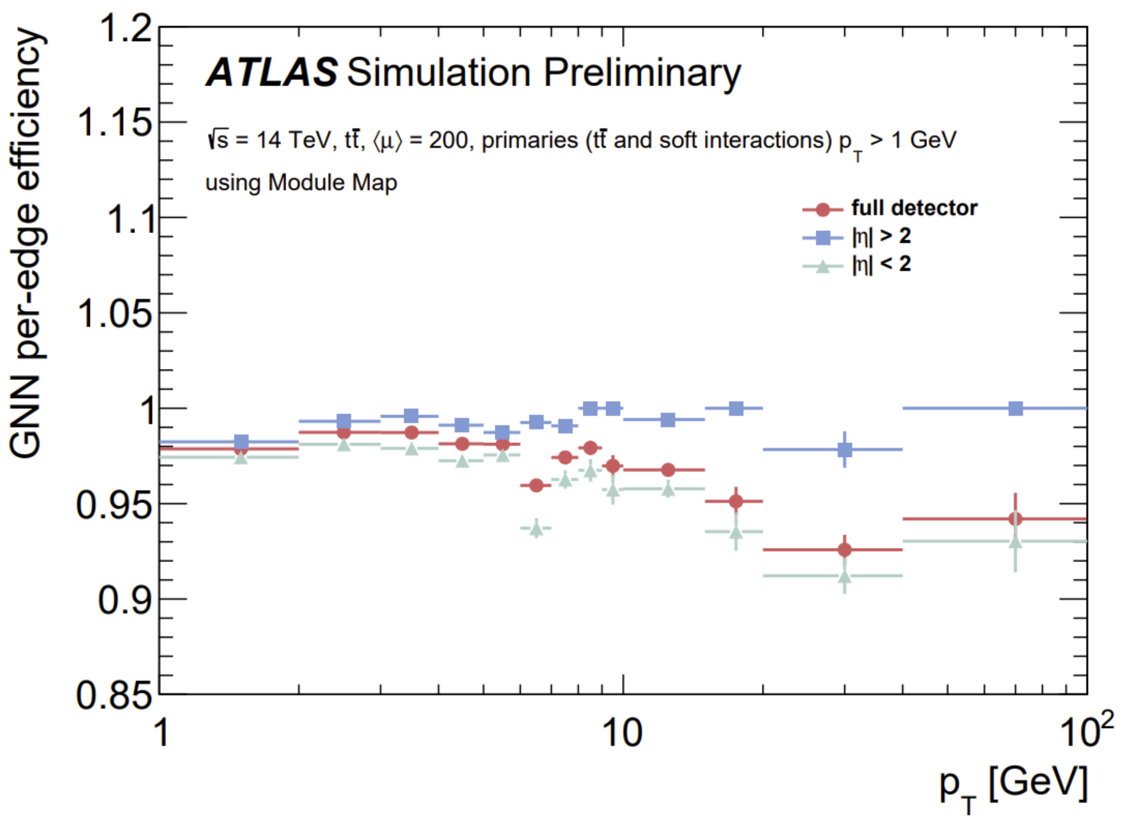
LOSS FUNCTION DESIGN

- The **target** of the GNN and track reconstruction is edges from primary particles with pT>1 GeV that have left at least 3 hits on different modules in the detector (see slide 12)
- Have very small set of target edges (1-2% of edges are true target t_{seq})
- Solution: t_{seq} $y = 1$ weighted up by $\times 10$, sequential background $\tilde{t}_{seq}$ masked, all others $y = 0$
  - Weighting gives much better performance at high-efficiency
  - Masking gives much better performance around the 1 GeV cutoff

GNN EDGE CLASSIFICATION RESULTS

ROC CURVE & EDGEWISE PERFORMANCE VS. p_T

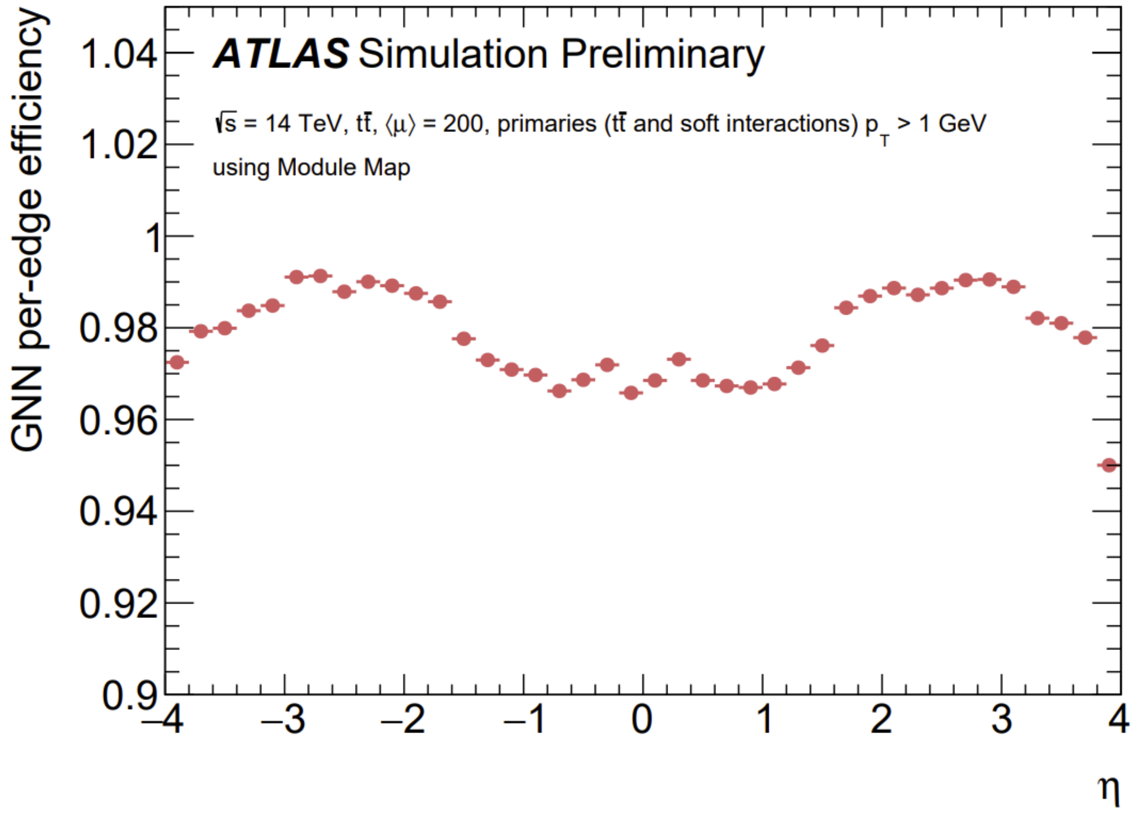


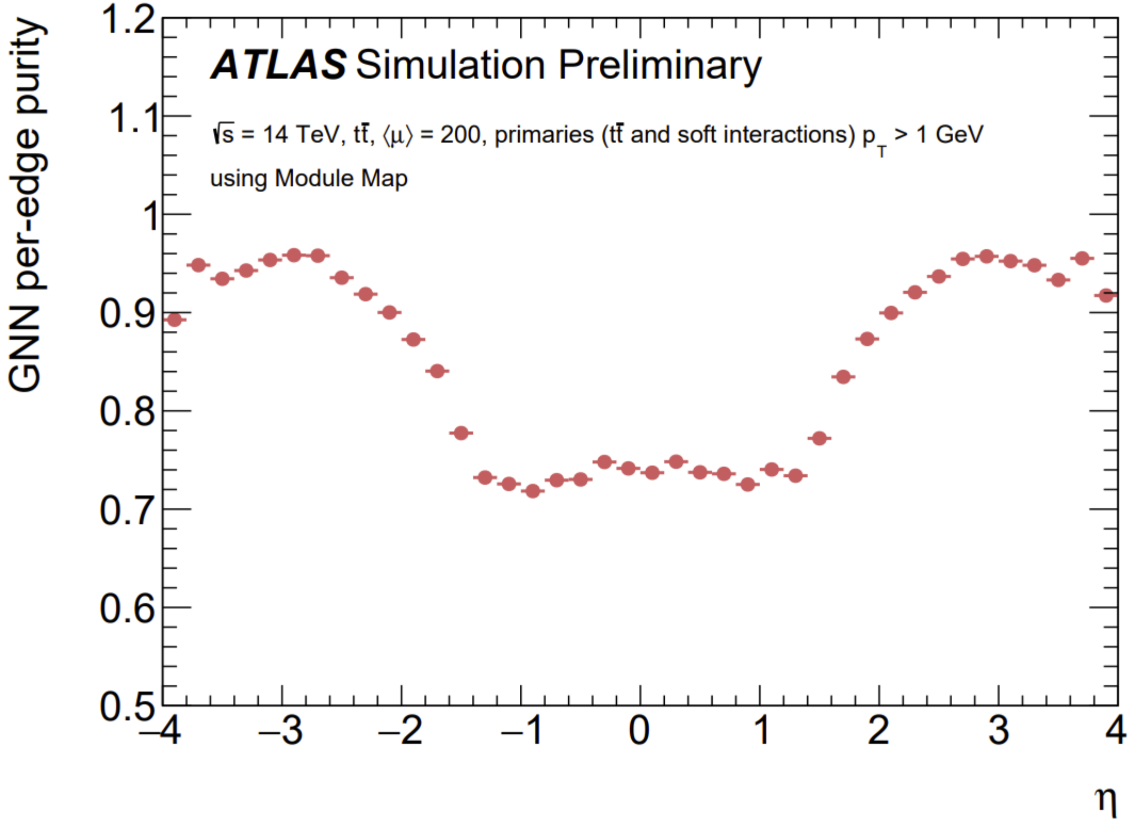


- Edge cut of 0.5 on output of GNN edge classifier

GNN EDGE CLASSIFICATION RESULTS

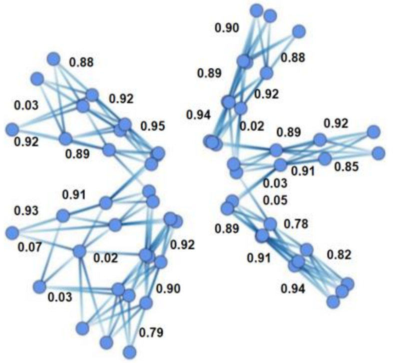
EDGEWISE PERFORMANCE VS. η





- Again, see a drop in performance at low η

Edge Scores

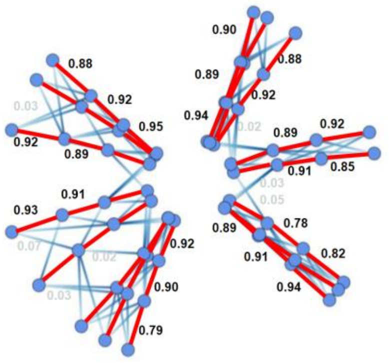


Connected Components

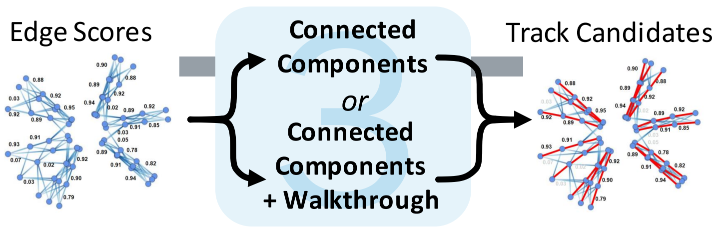
or

Connected Components + Walkthrough

Track Candidates



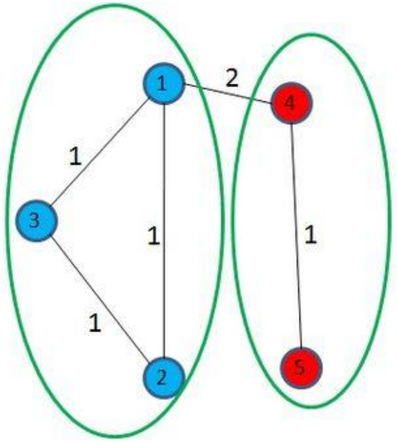
Graph Segmentation



Graph Segmentation

GRAPH PARTITIONING 101

- Graph: given a graph with expected "components" or "communities", how can we partition into those likely components
- Potentially a *very* (i.e. NP-hard) expensive step
- Typical partitioning approaches try to cut the fewest edges, to produce the most densely connected communities
- But this is not really aligned with track finding, since tracks ideally only have one incoming, one outgoing edge per node

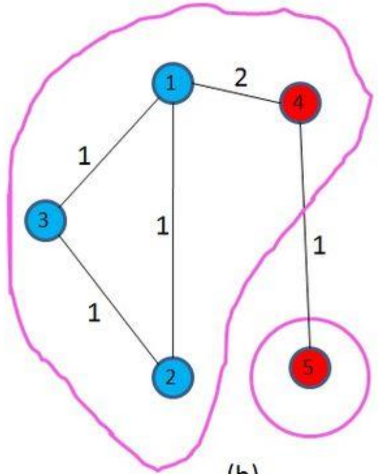


(a)

Community ID	1	2
1	6	2
2	2	2

$$Q = \frac{6}{12} - \left(\frac{8}{12}\right)^2 + \frac{2}{12} - \left(\frac{4}{12}\right)^2 = \frac{16}{144}$$

Ratio cut=2/(3*2)=1/3

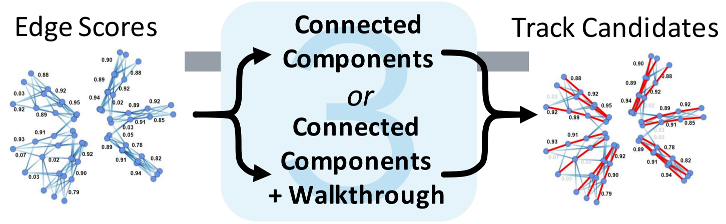


(b)

Community ID	1	2
1	10	1
2	1	0

$$Q = \frac{10}{12} - \left(\frac{11}{12}\right)^2 + \frac{0}{12} - \left(\frac{1}{12}\right)^2 = \frac{-2}{144}$$

Ratio cut=1/(4*1)=1/4



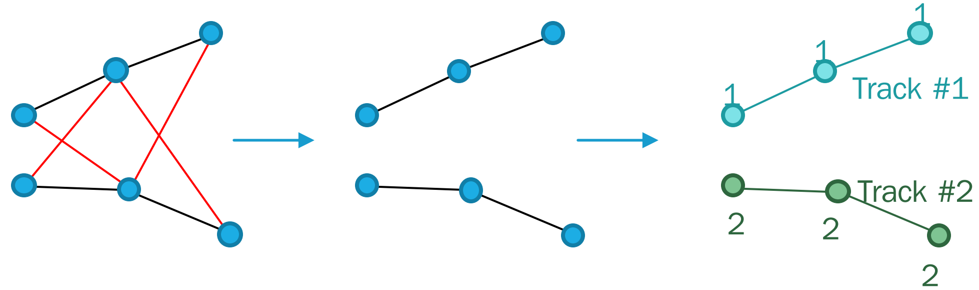
Graph Segmentation

TRACK CANDIDATES CONSTRUCTION

- We now have labelled edges. Want to now label each *node* depending on connectivity.
- Two distinct approaches: component-based segmentation, or path-based segmentation.

Component-based

E.g. connected components algorithm:

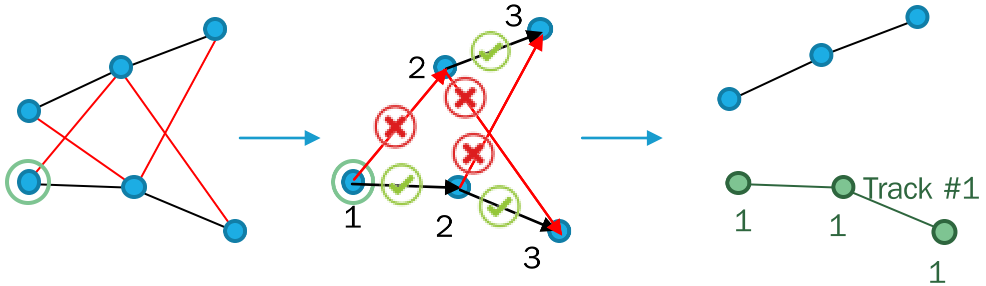


Classified edges | Ignore cut edges | Label connected components

- Pros: Fast - $O(N_{nodes})$
- Cons: Can merge tracks into one candidate

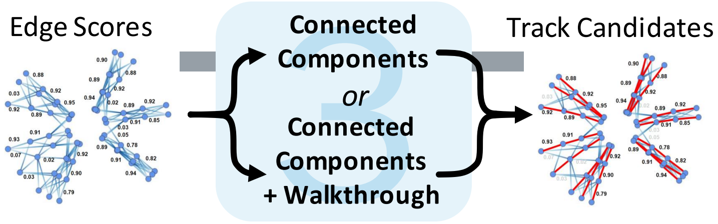
Path-based

E.g. walkthrough algorithm:



Classified edges, Starting node | Choose high score junctions | Remove a high-scoring path

- Pros: Handles hits as a sequence, as a track should be
- Cons: Potentially slow - $O(N_{edges})$, needs a *directed* graph



Graph Segmentation

TRACK CANDIDATES CONSTRUCTION

- We now have labelled edges. Want to now label each *node* depending on connectivity.
- Two distinct approaches: component-based segmentation, or path-based segmentation.

Component-based

E.g. connected components algorithm:

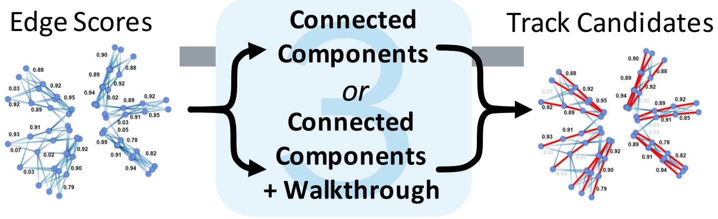
- Pros: Fast - $O(N_{nodes})$
- Cons: Can merge tracks into one candidate

Path-based

E.g. walkthrough algorithm:

- Pros: Handles hits as a sequence, as a track should be
- Cons: Potentially slow - $O(N_{edges})$, needs a *directed* graph

Both methods by construction associate each hit with only one track

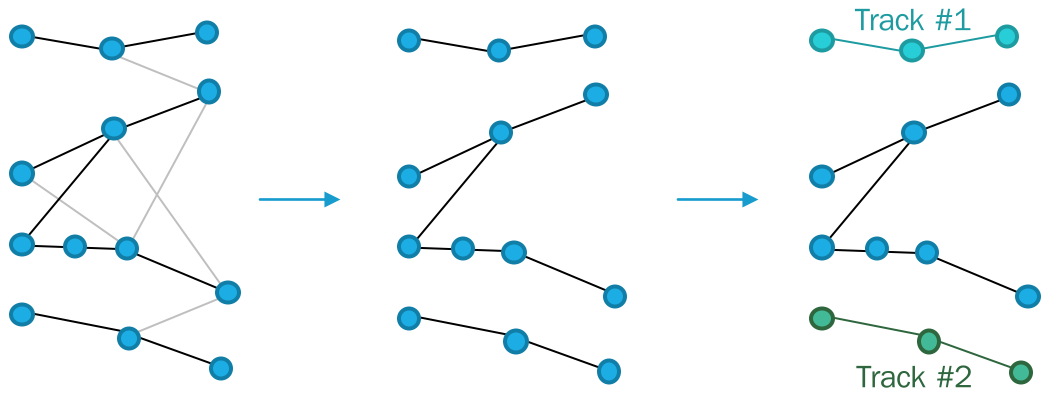


Graph Segmentation

TRACK CANDIDATES CONSTRUCTION

- One can combine the good features of each approach:

1. Connected Components

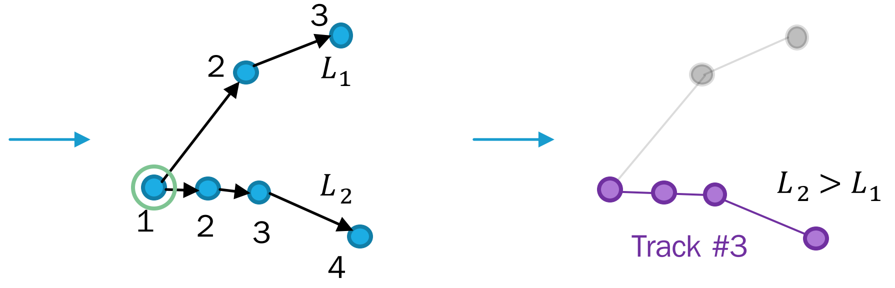


Classified edges

Cut score < 0.2

Label simple candidates

2. Walkthrough, a.k.a "Wrangler"



Walk through paths from starting node, count length L

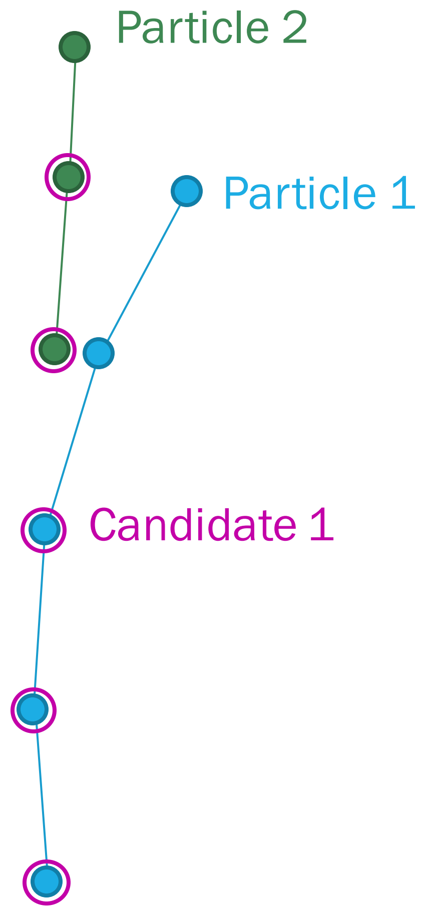
Assign longest path as candidate

FINDING & FITTING PERFORMANCE

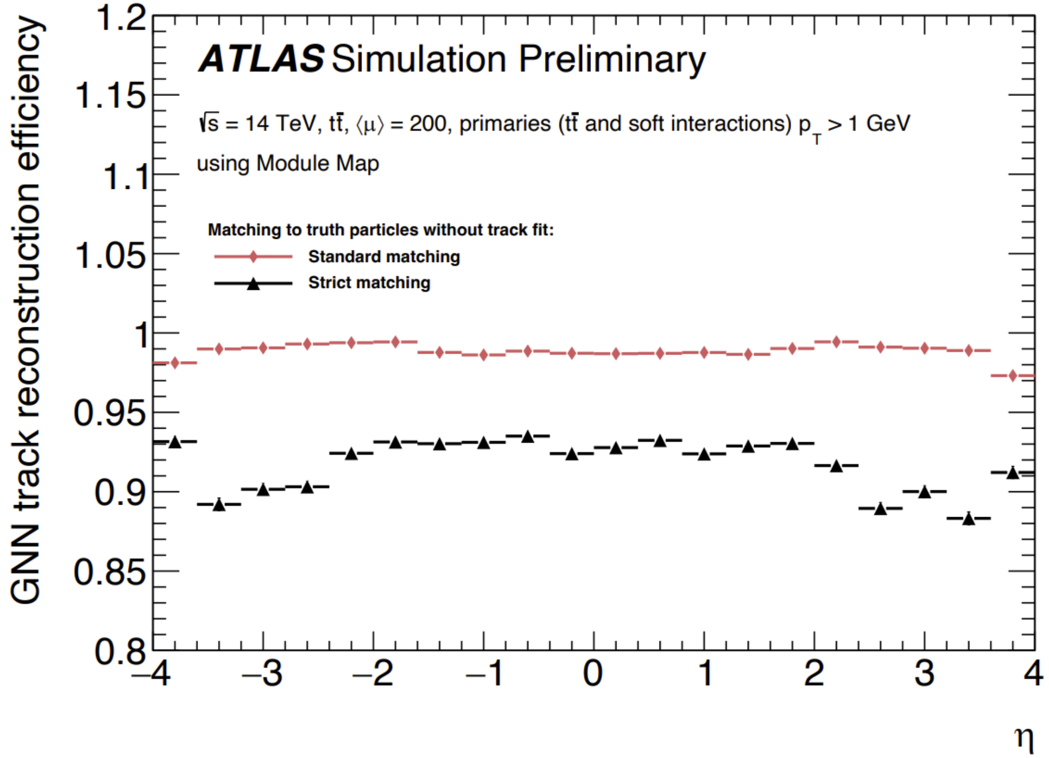
TRACK MATCHING DEFINITIONS

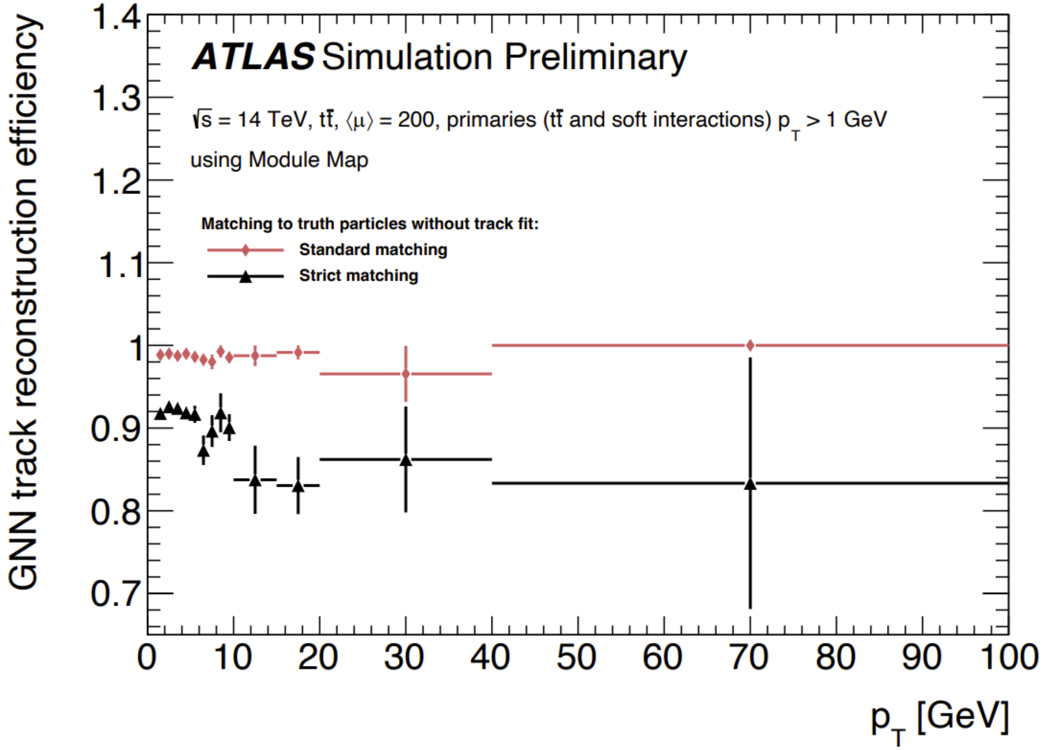
- $N(P_i, C_j)$ is the number of spacepoints shared by particle i and candidate j
- Particle i is called "matched" if, for some j, $\frac{N(P_i,C_j)}{N(P_i)} > f_{truth}$
- Candidate j is called "matched" if, for some i, $\frac{N(P_i,C_j)}{N(C_j)} > f_{reco}$
- Particle i and candidate j are called "double matched" if, for some i and j, $\frac{N(P_i,C_j)}{N(P_i)} > f_{truth}$ and $\frac{N(P_i,C_j)}{N(C_j)} > f_{reco}$
- $eff = \frac{\sum_i P_i(matching\ condition)}{\sum_i P_i}$, $pur = \frac{\sum_j C_j(matching\ condition)}{\sum_j C_j}$

Standard matching: single-matched particles with $f_{truth} = 0.5$
Strict matching: double-matched particles with $f_{reco} = 1.0$



TRACK RECONSTRUCTION RESULTS

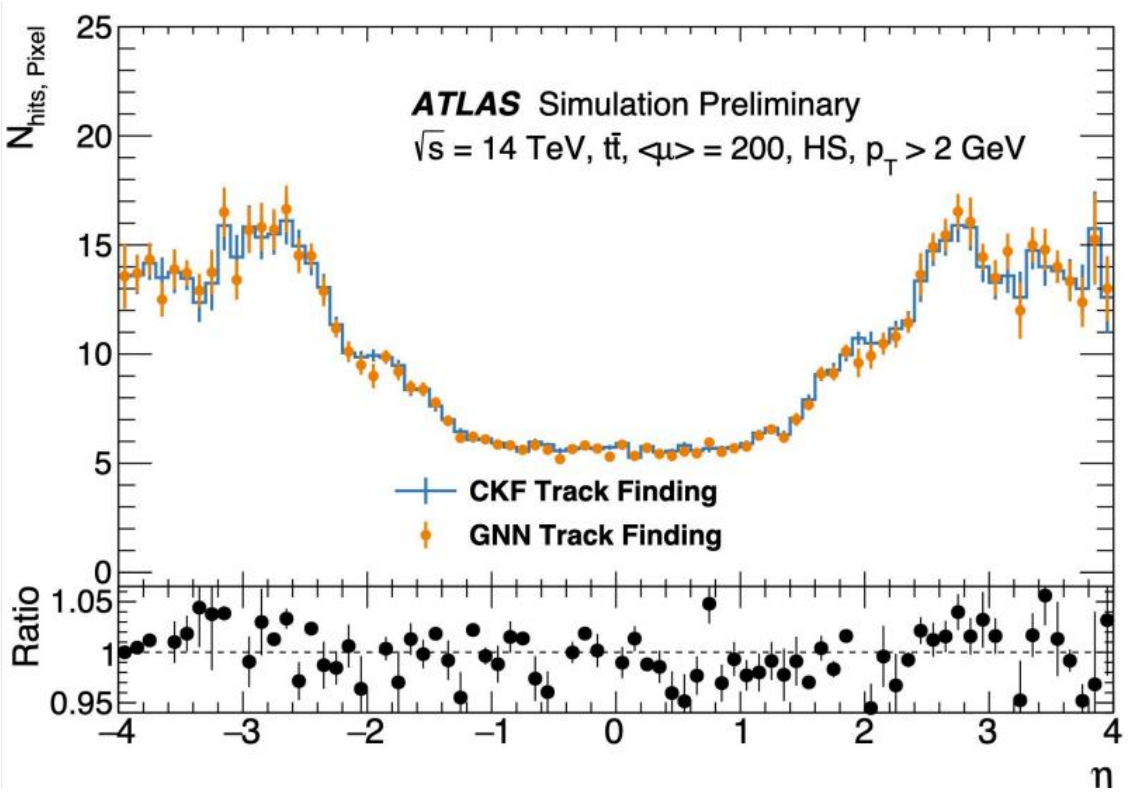


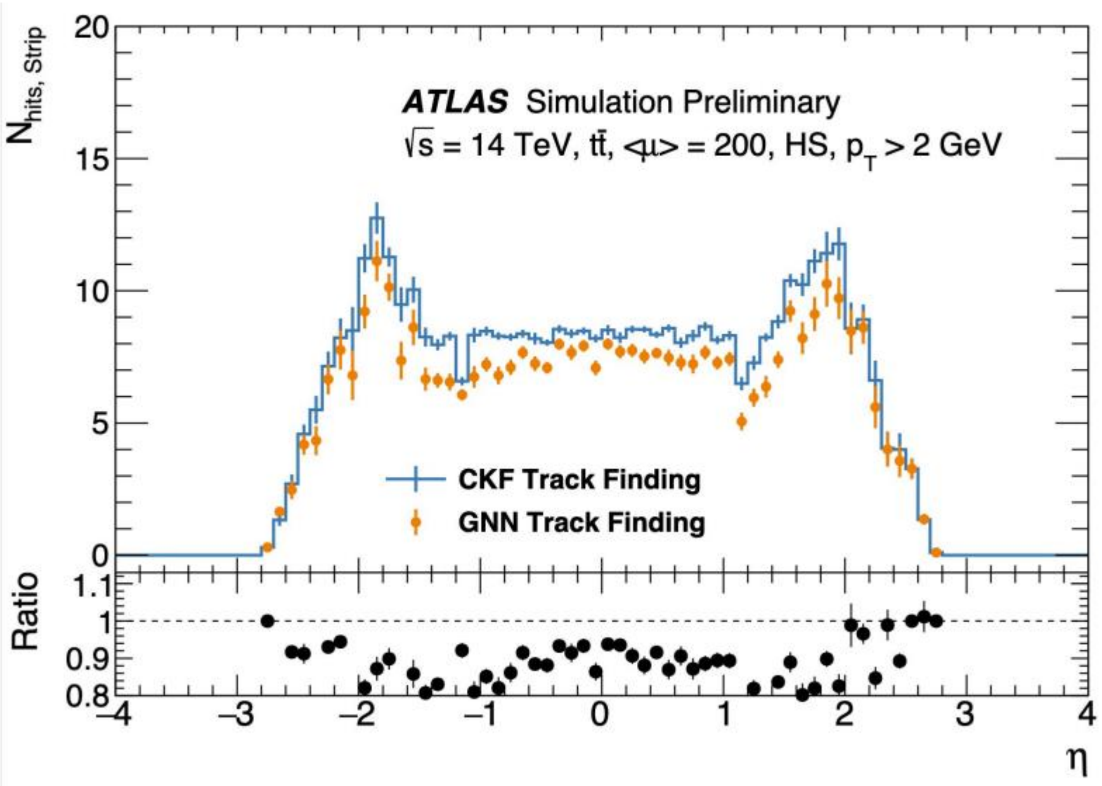


Standard matching: single-matched particles with $f_{truth} = 0.5$
Strict matching: double-matched particles with $f_{reco} = 1.0$

- Fake rate is $O(10^{-3})$ using standard truth matching

TRACK RECONSTRUCTION RESULTS





Observe that the GNN track candidates have fewer hits than CKF. Will return to this!

TRACK FITTING 101

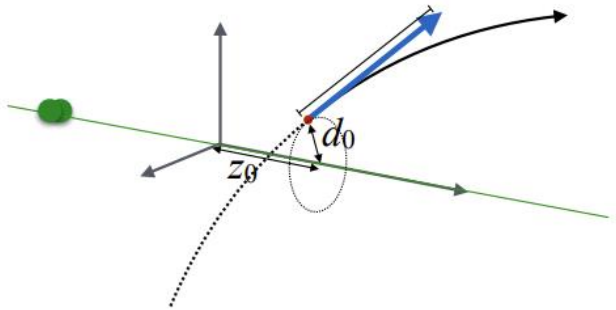
- Why do we fit?
- Many downstream tasks need the momentum and "impact parameters"
- Can use the fitted parameters to "tidy up" the track finding

- A trajectory of a charged particle in a magnetic field requires **five track parameters** (q)

$$\mathbf{q} = (d_0, z_0, \phi, \theta, q/p)$$

- Uncertainties encoded in a covariance matrix

$$\mathbf{C} = \begin{pmatrix} \sigma^2(d_0) & cov(d_0, z_0) & cov(d_0, \phi) & cov(d_0, \theta) & cov(d_0, q/p) \\ . & \sigma^2(z_0) & cov(z_0, \phi) & cov(z_0, \theta) & cov(z_0, q/p) \\ . & . & \sigma^2(\phi) & cov(\phi, \theta) & cov(\phi, q/p) \\ . & . & . & \sigma^2(\theta) & cov(\theta, q/p) \\ . & . & . & . & \sigma^2(q/p) \end{pmatrix}$$



- Right handed coordinate system
- Azimuthal angle, ϕ, measured in transverse plane in [-π,+π)
- Polar angle, θ measured from z axis in [0,π]
- Pseudorapidity, $\eta = -\ln(\tan\theta/2)$

FAST CIRCLE FIT WITH THE CONFORMAL MAPPING METHOD

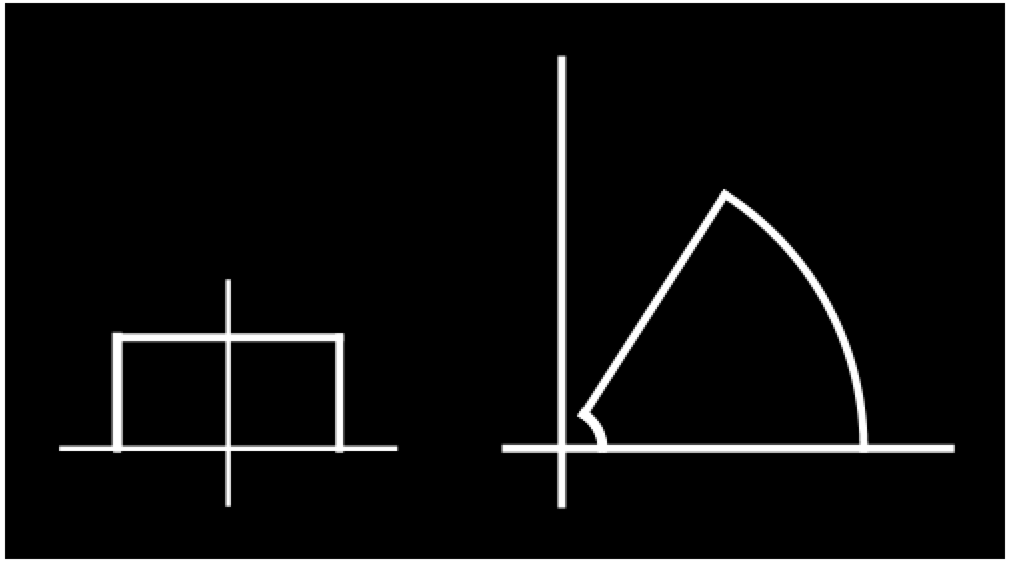
M. Hansroul, CERN/DD
H. Jeremie and D. Savard, Université de Montréal

TRACK FITTING 101: FITTING WITH GLOBAL χ^2

- We can propose a shape of the track – a helix
- We can then simply minimise the sum of the square of the residuals of the measurements to produce the set of five track parameters

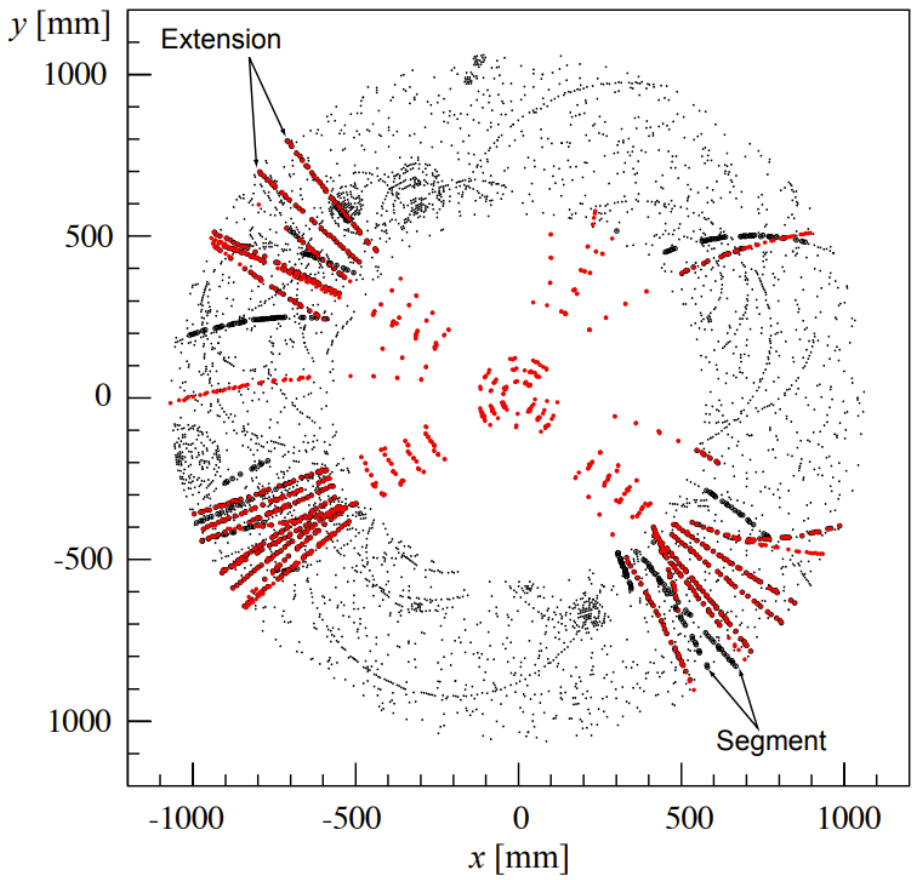
- This can be done *very* efficiently by:

1. Mapping to the conformal plane →
2. Making the assumption that the impact parameters (the point of closest approach of the helix to the origin) is very small



TRACK FITTING 101: FITTING WITH KALMAN FILTER

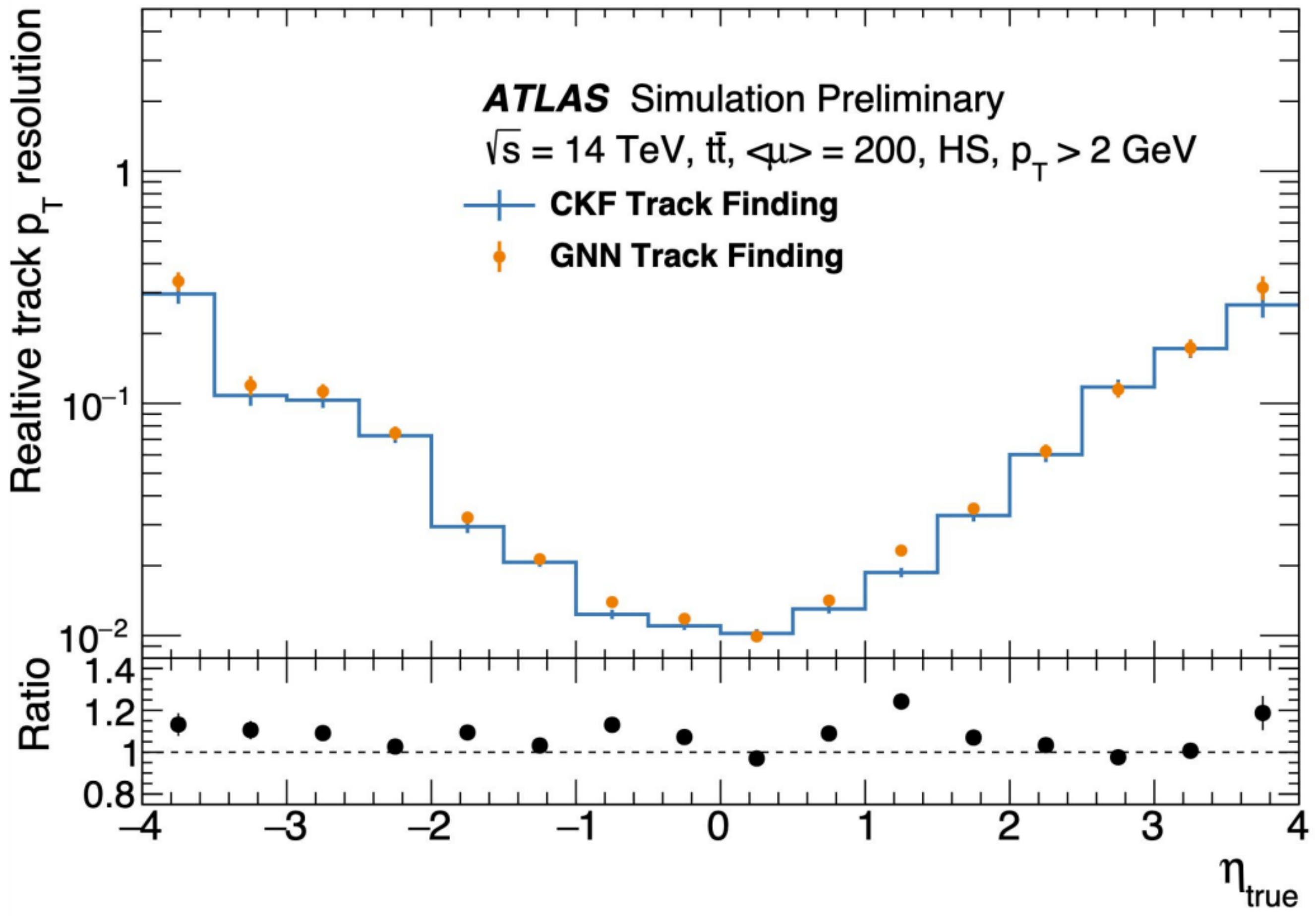
- Recall that the Kalman Filter track finding produces a prediction of the helical parameters in order to find the next hit
- We can thus use the same model to fit to a track
- However, to get good performance: First run KF forwards to build the model, then run it back from out-to-in: called "smoothing"



http://physik.uibk.ac.at/hephy/theses/diss_as.pdf

FITTING PERFORMANCE OF TRACK CANDIDATES

- We see that the tracks found by the GNN are within 30% of the "quality" of the tracks found by the CKF
- Quite promising, given that the CKF *assumes* helicity, while the GNN makes no such assumption



Relative track p_T resolution is measured as the multiplication of p_T^{true} and the RMS of the pull distribution of $(q/p_T^{reco} - q/p_T^{true}) / q/p_T^{true}$.