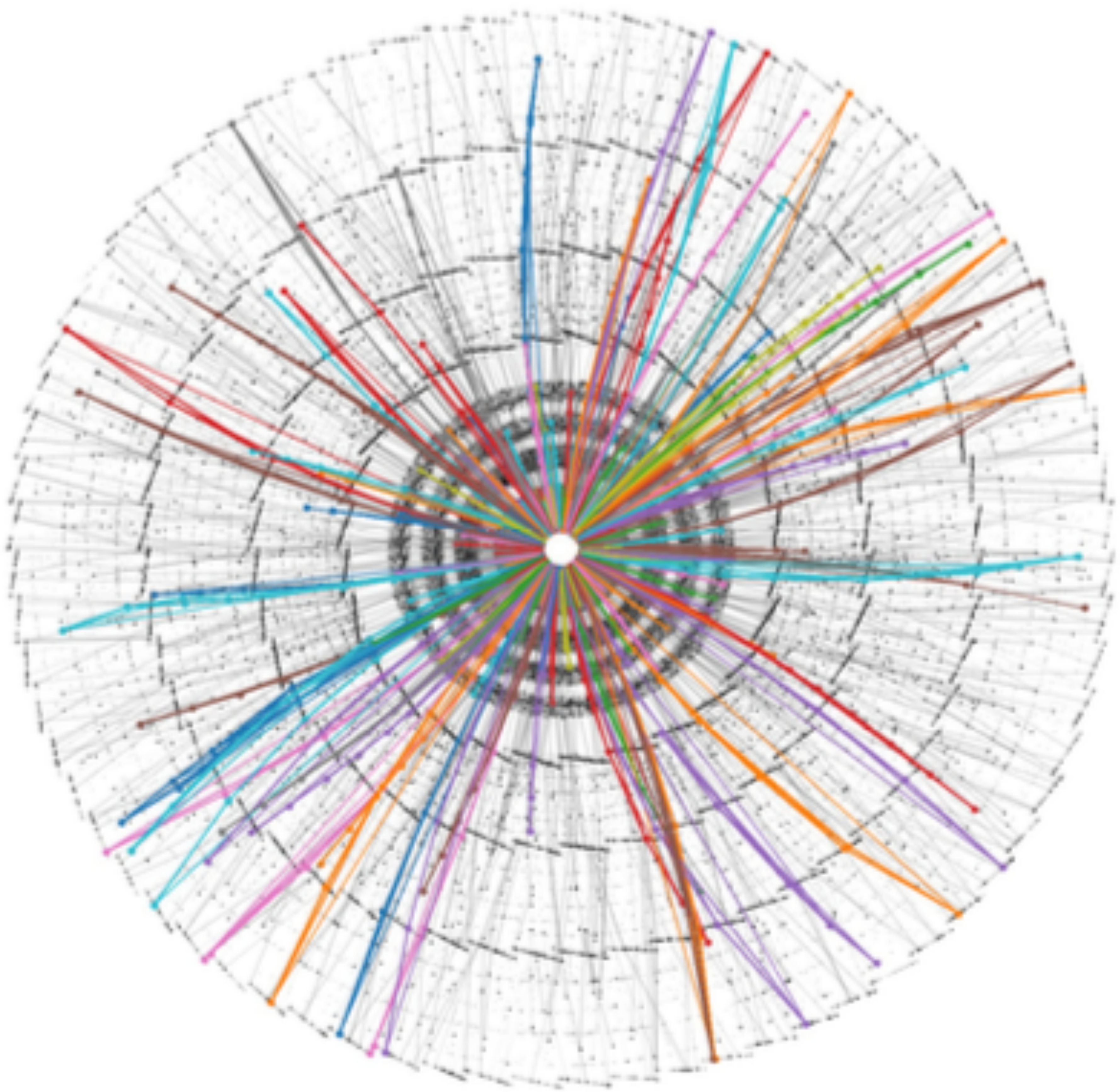


Track Reconstruction & AI

A. Salzburger (CERN)

A. Salzburger, CODEX-b week, CERN, 12/06/2023

What do you mean by AI ?

What was formally called **Machine Learning ?**

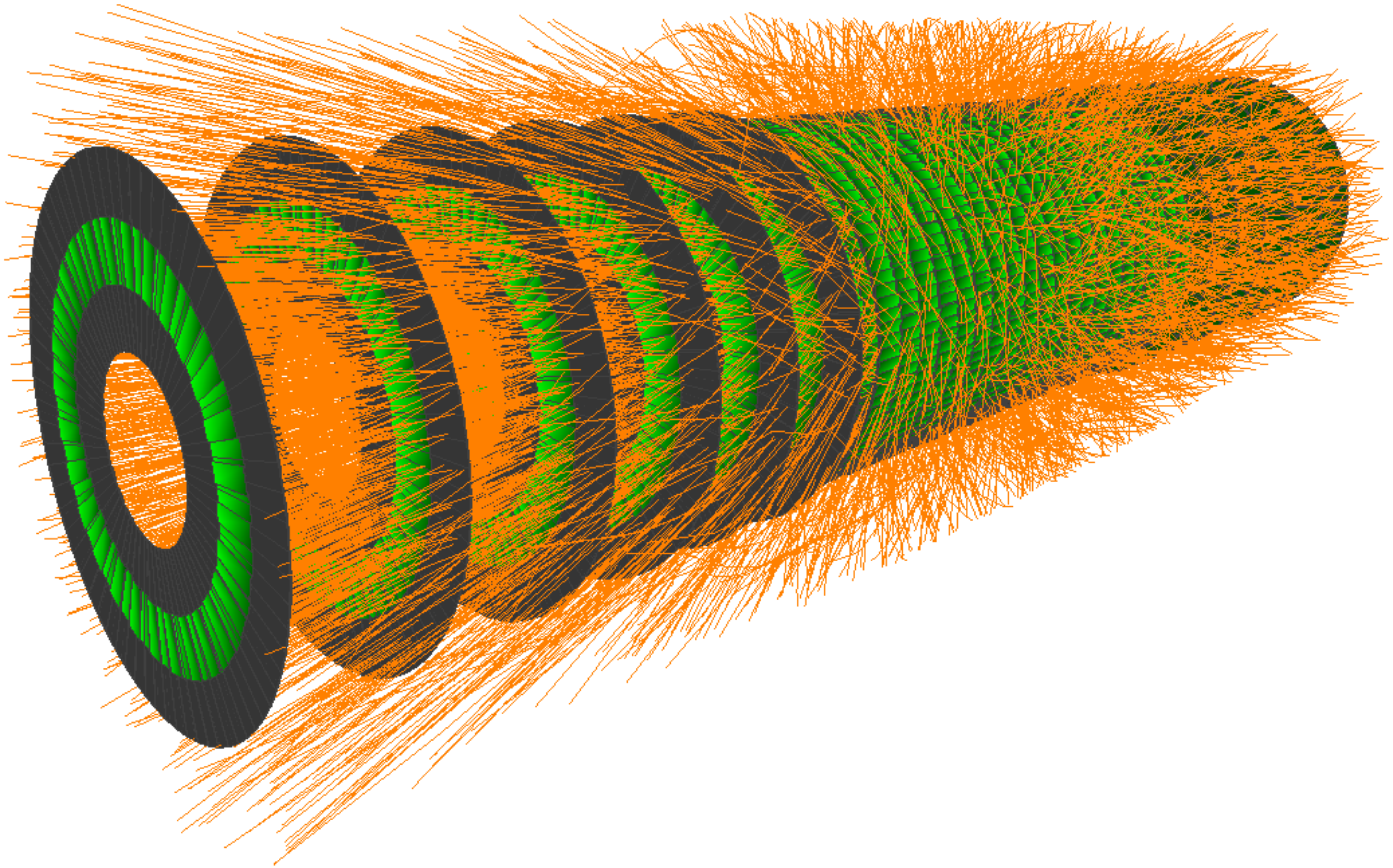
Unsupervised learning / supervised learning

Or more what most people associate nowadays, **Deep Learning ?**

NNs, CNNs, GNNs

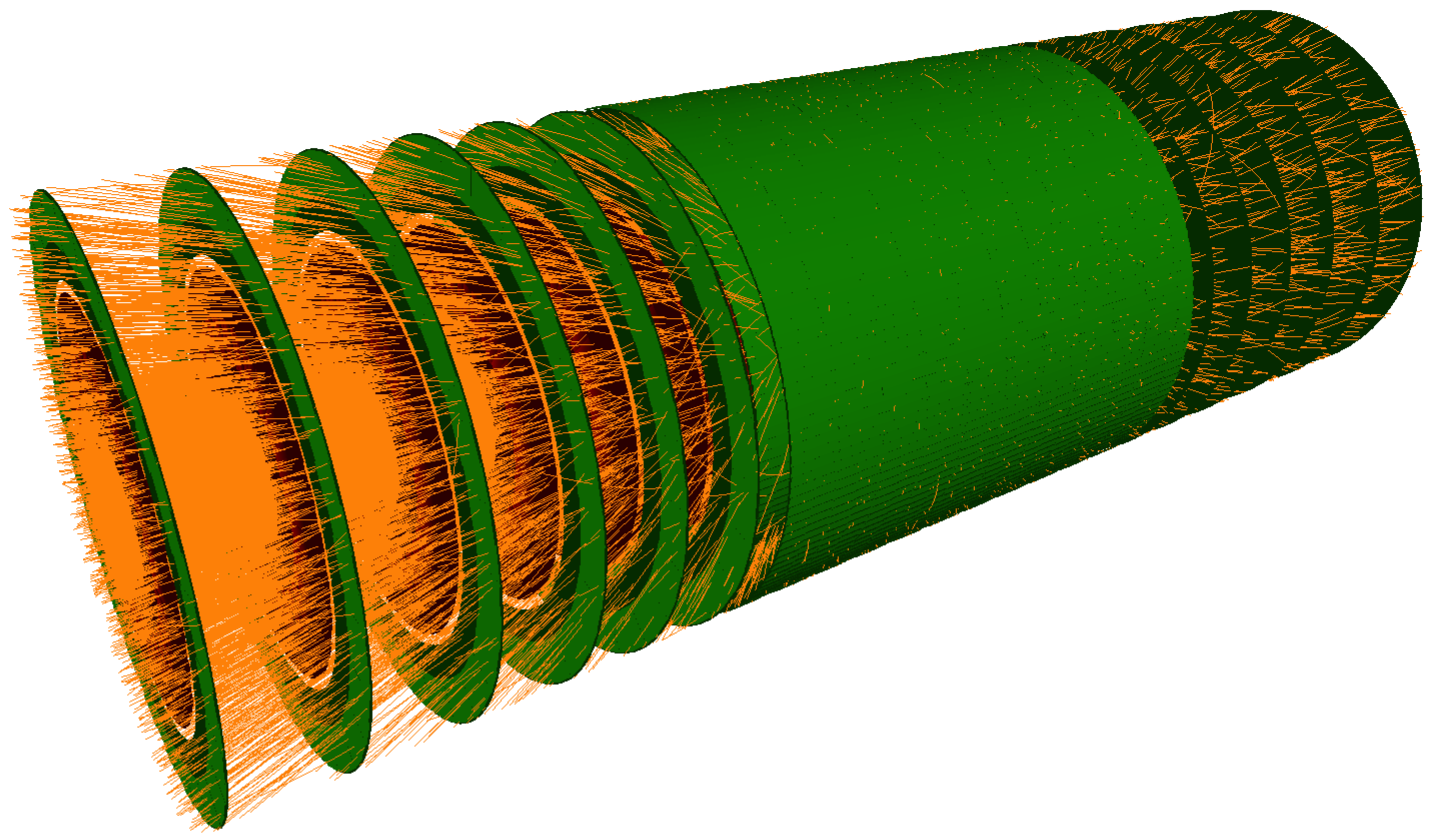
Or **Large Language Models ?**

The problem

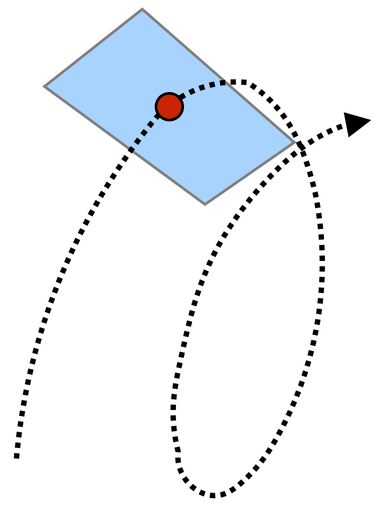


Simulated event with overlaid 200 proton-proton collisions for the TrackML challenge

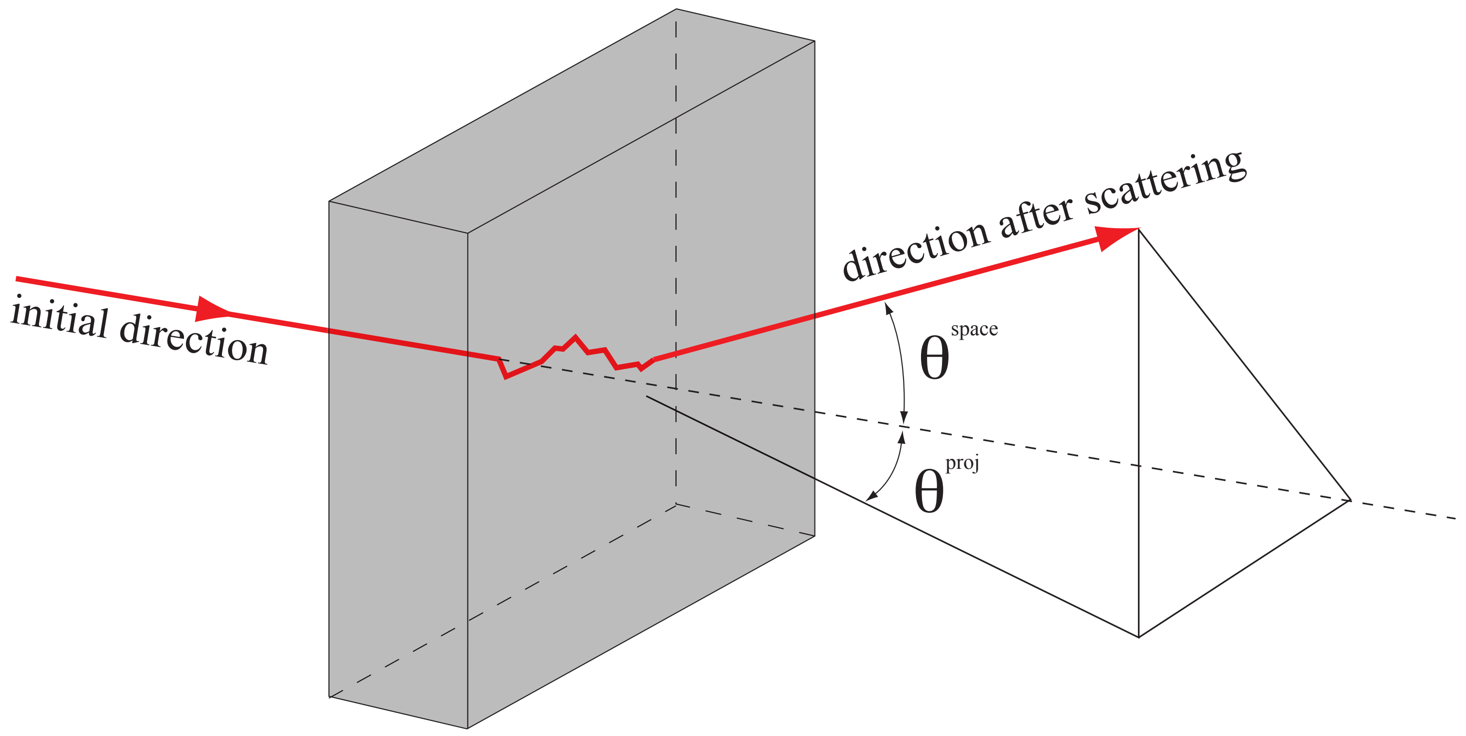
Charged particles in a detector



Charged **particles** traverse the detector, following **physical laws**

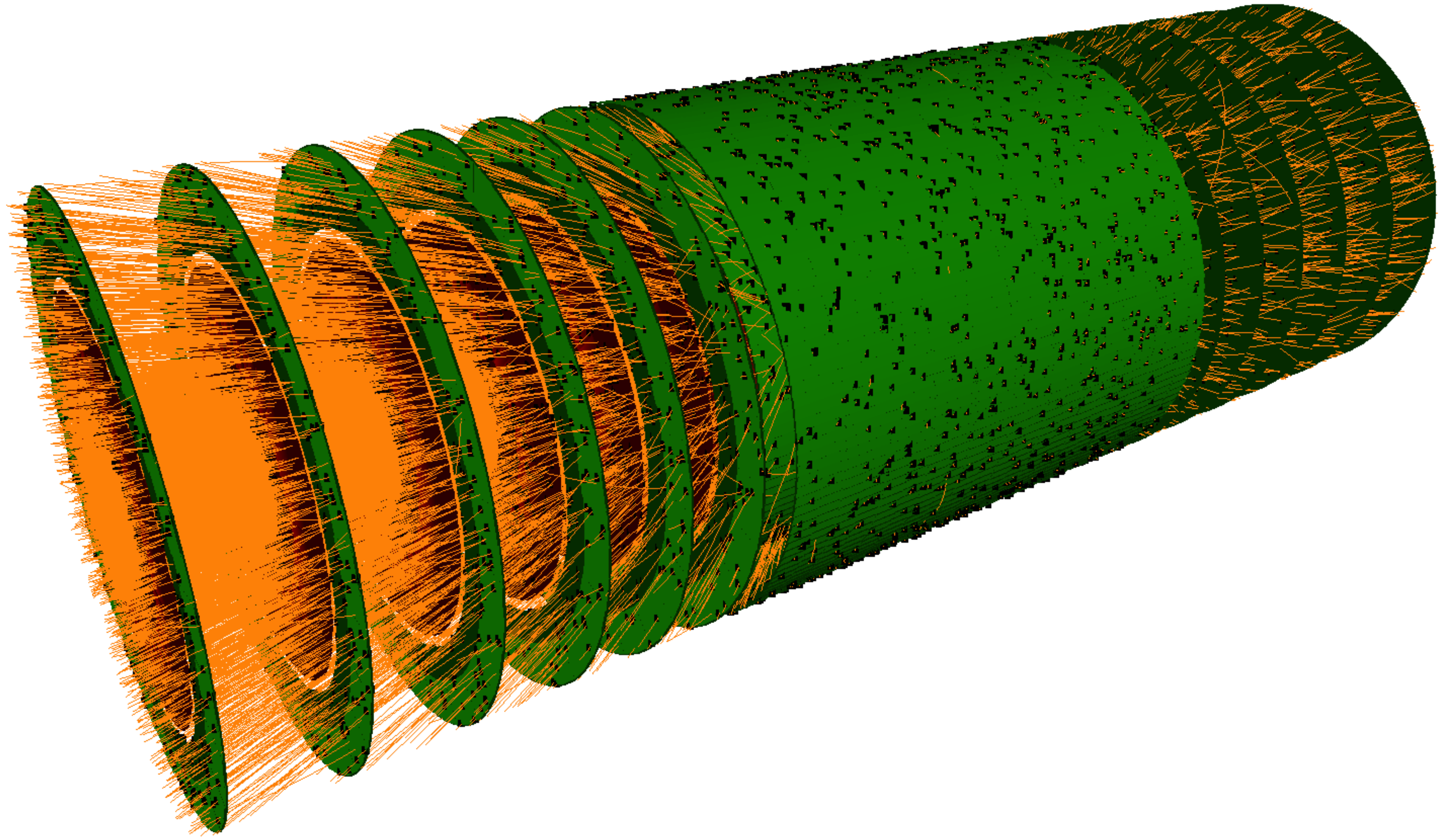


& are disturbed
by **interactions with detector** material.

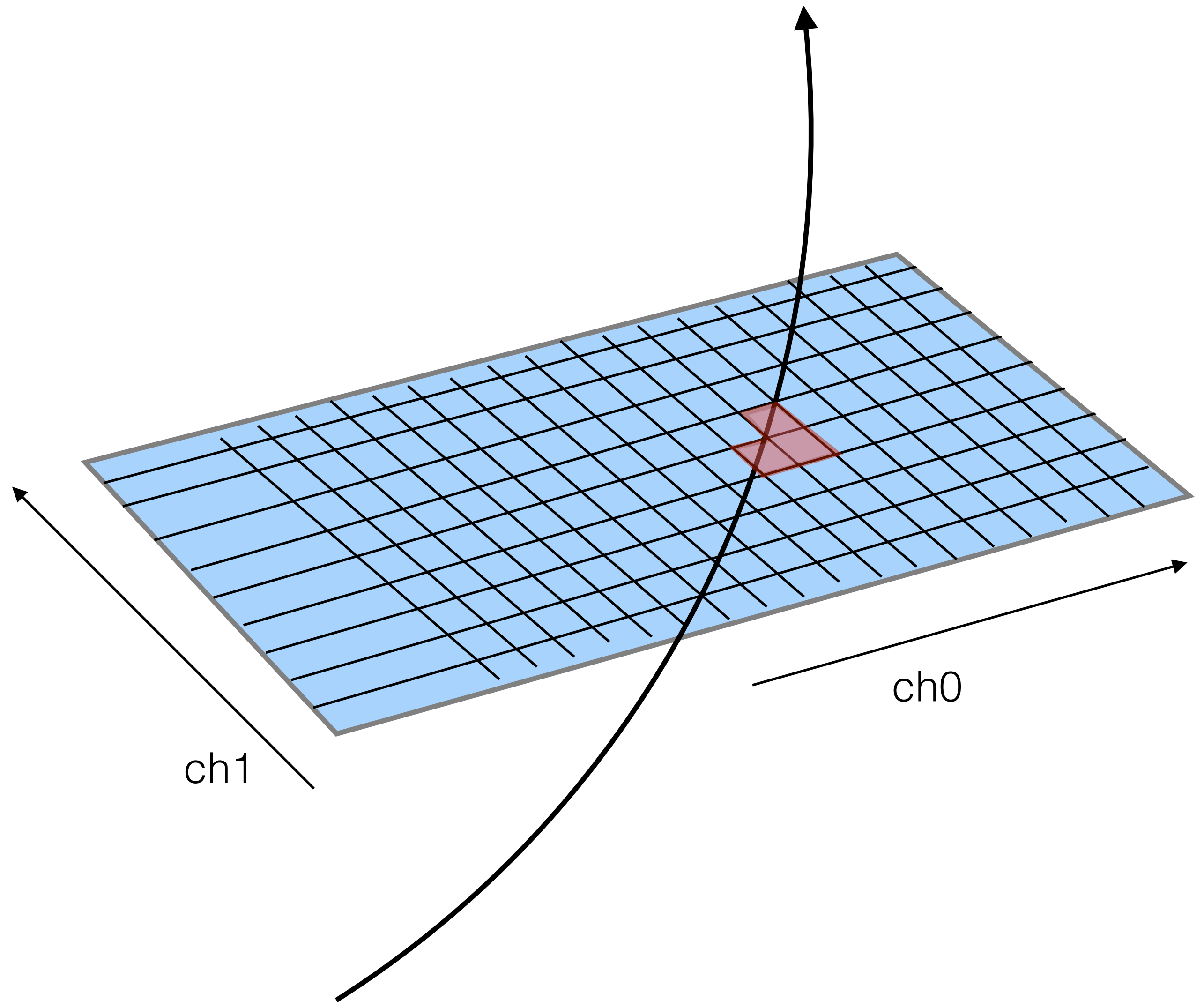


The data

spatial* localisation of charged particles on concrete detector layers

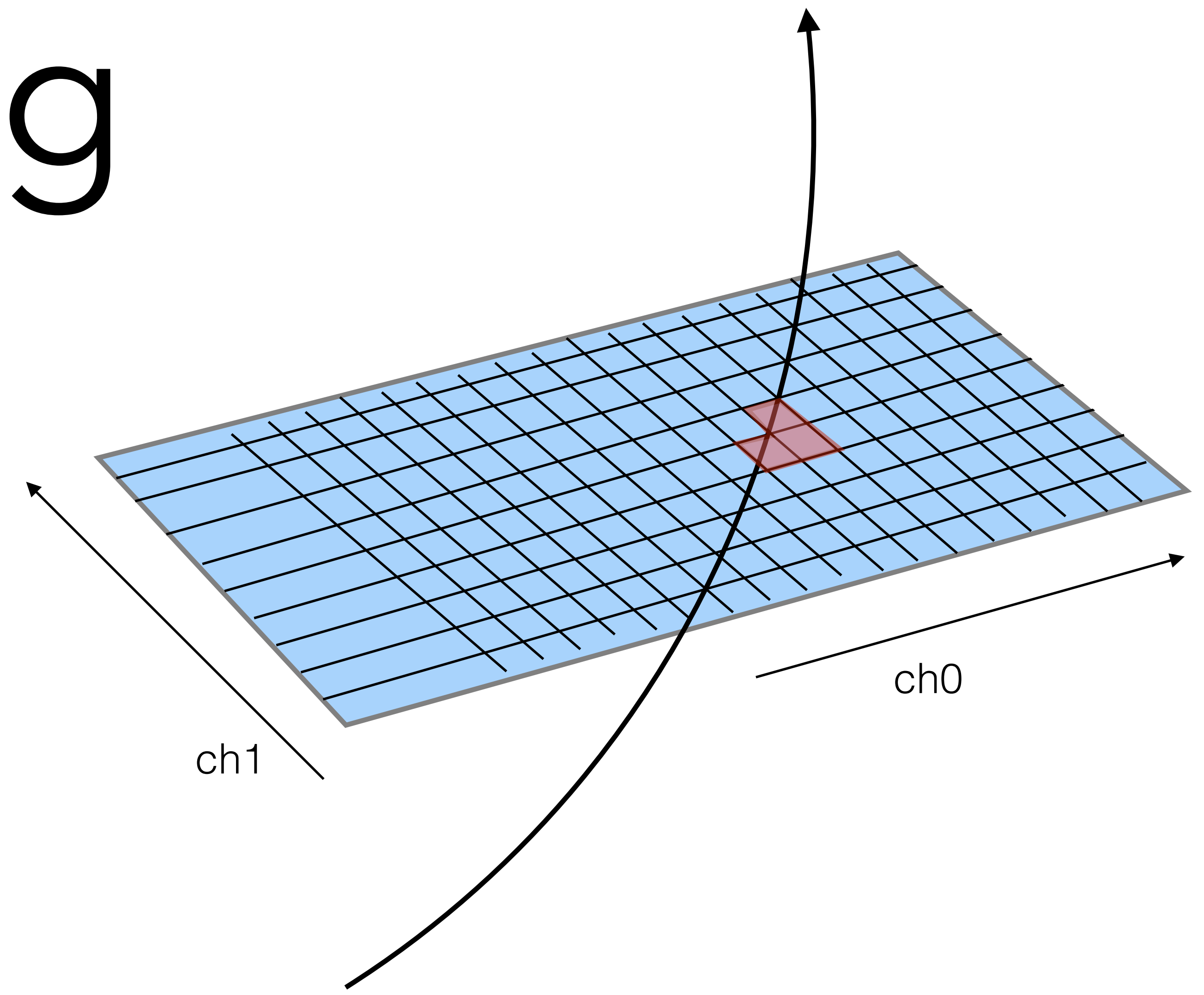


Detection **devices** measure the particle with a given resolution.



Clustering

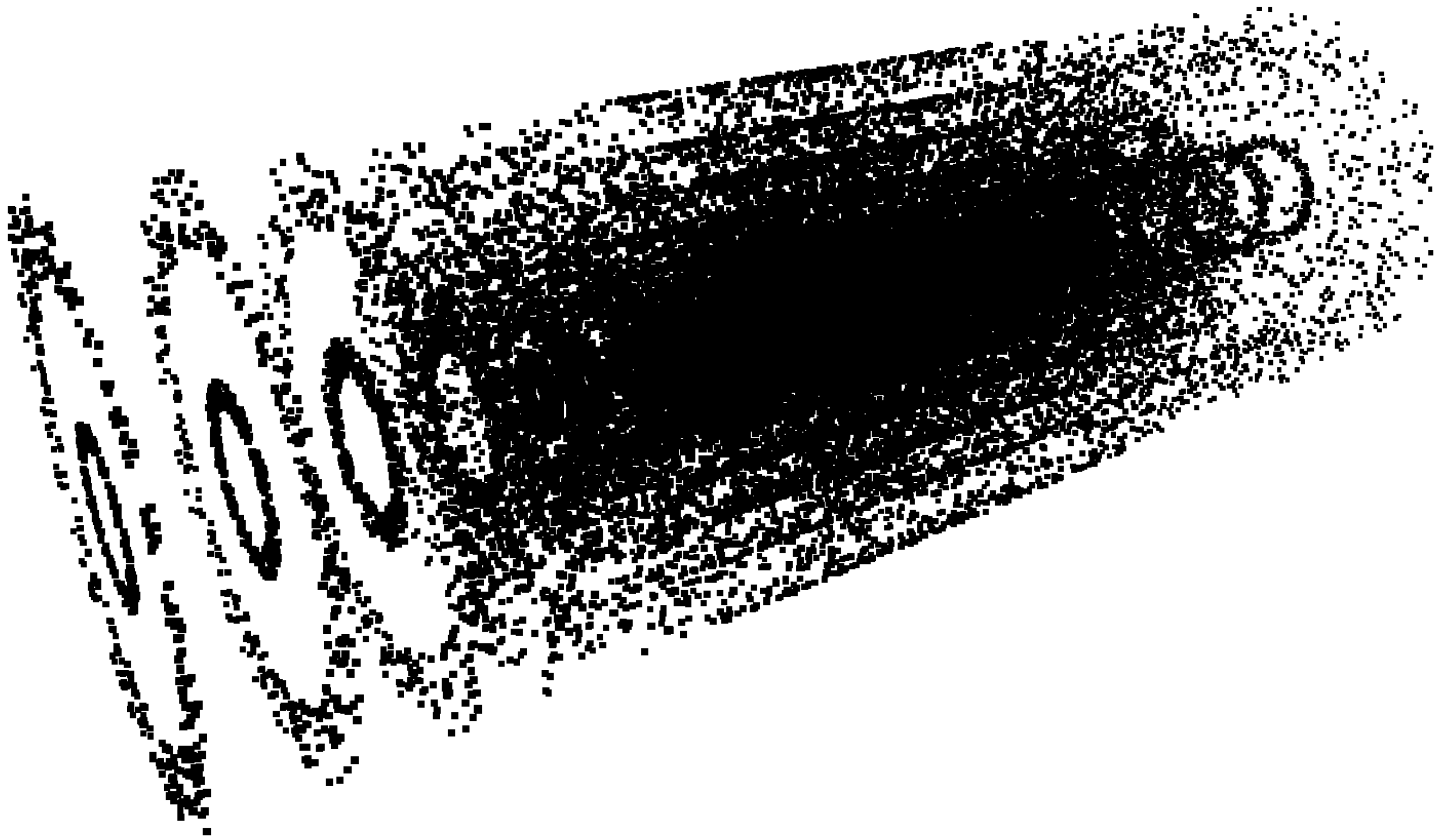
Unsupervised learning such as clustering (individual channels into a single measurement) is also classified as machine learning, e.g. k-means:



$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

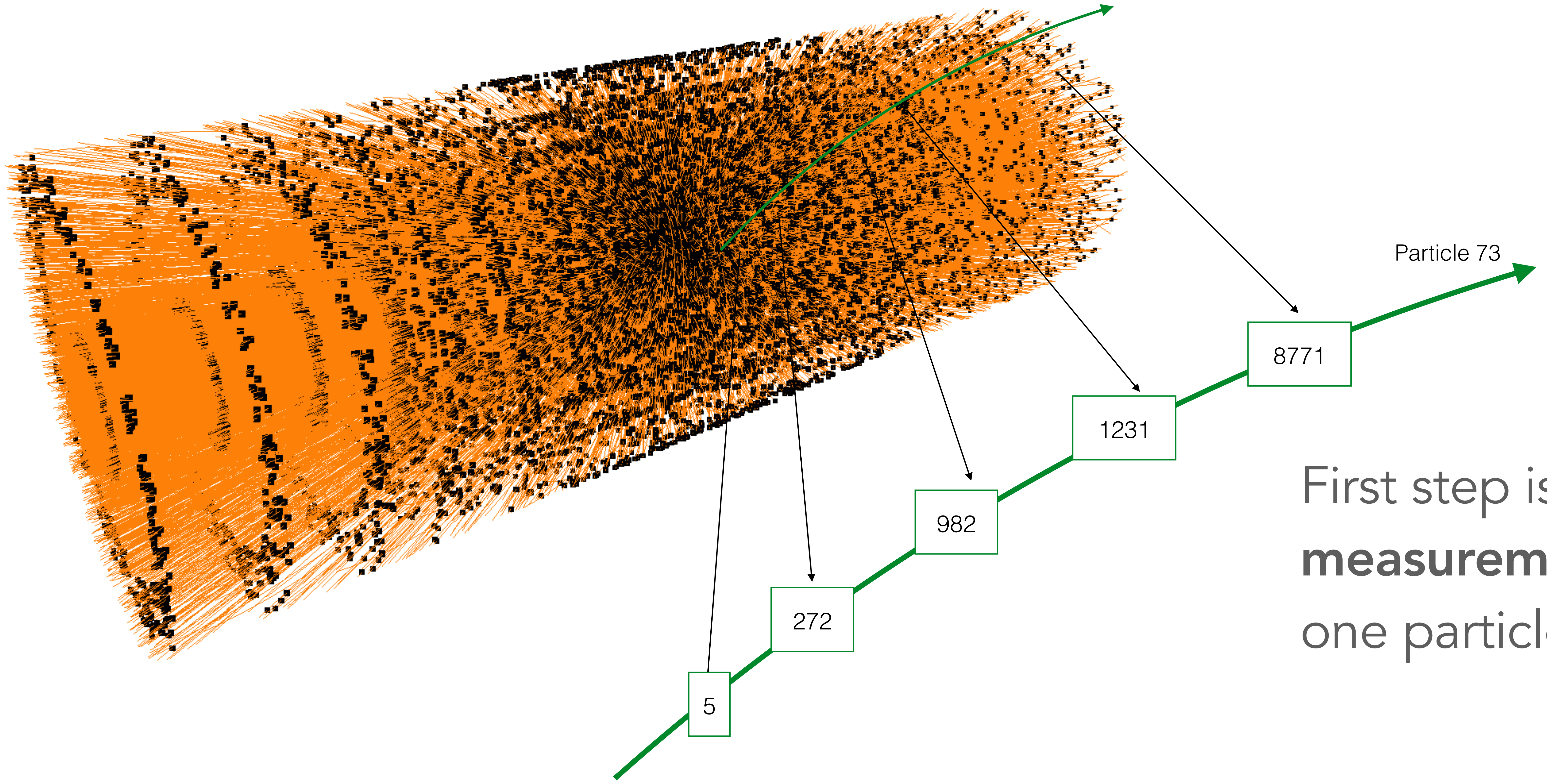
Illustration:
Parts of the map of the 1854 cholera outbreak in London's Soho district by **Dr. John Snow**.

Input: a (bit more than a) point cloud



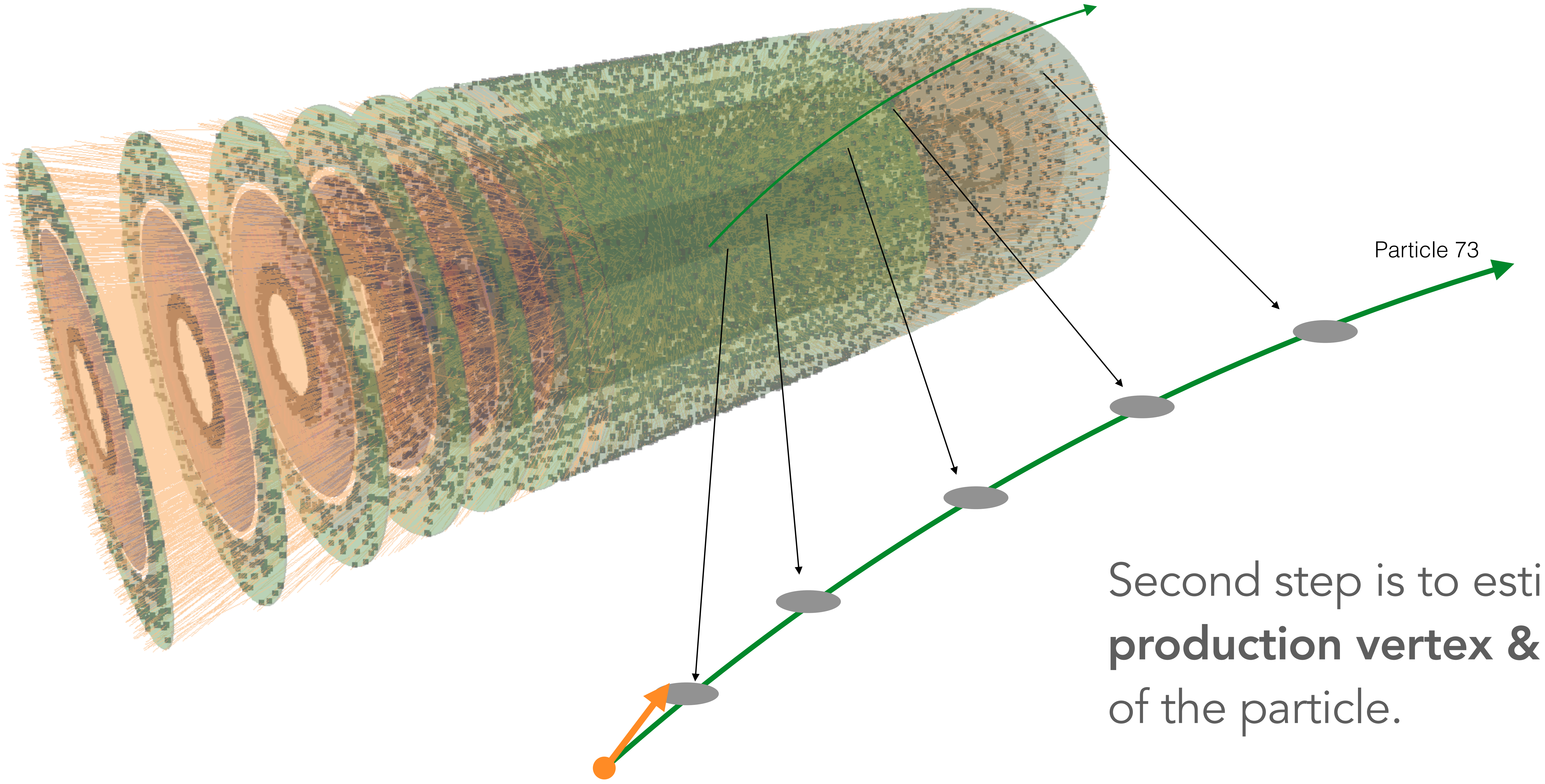
Input data is **point cloud** with certain **local features**:
- cluster shapes
- energy deposits
- local environment
- time (limited availability)

0: a labelling problem



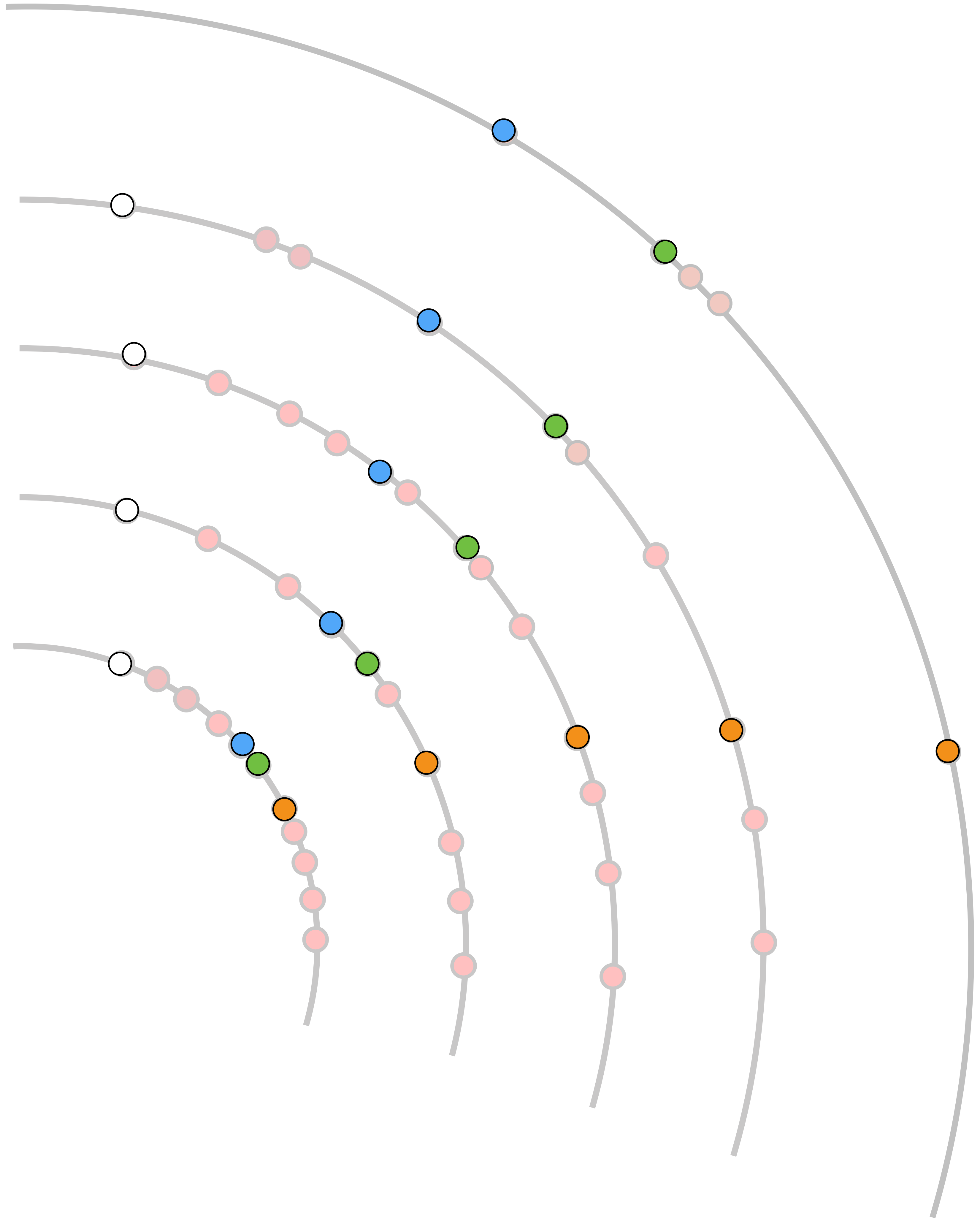
First step is to label **measurements** that stem from one particle.

1: an inference problem



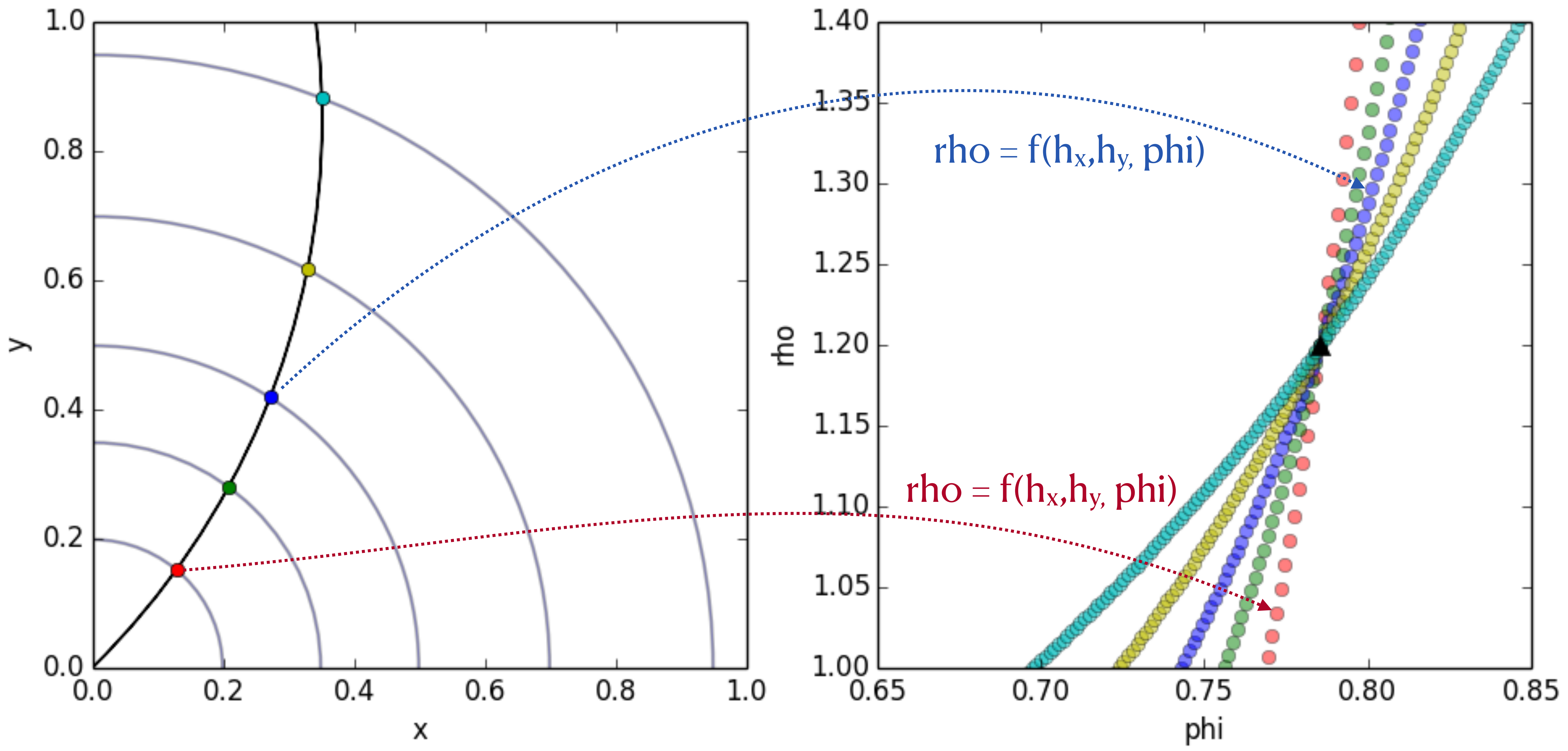
Second step is to estimate the **production vertex & kinematic properties** of the particle.

Clustering problem



https://fusiontables.google.com/DataSource?docid=1HsIb_r4gYYmIz8y_UE1h-X8yUtAYW2lNy99BR_c#map:id=3

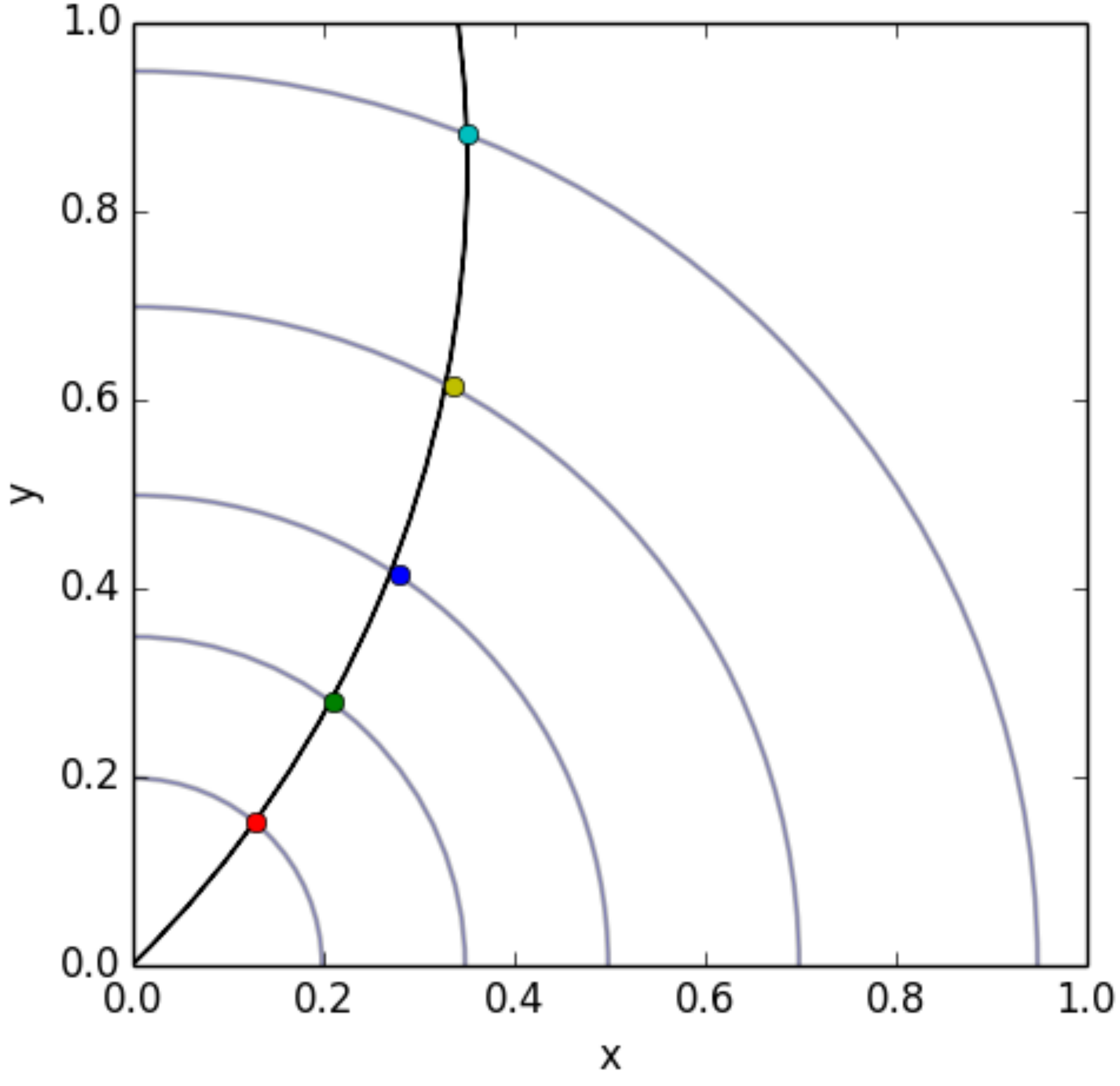
A simple transform problem?

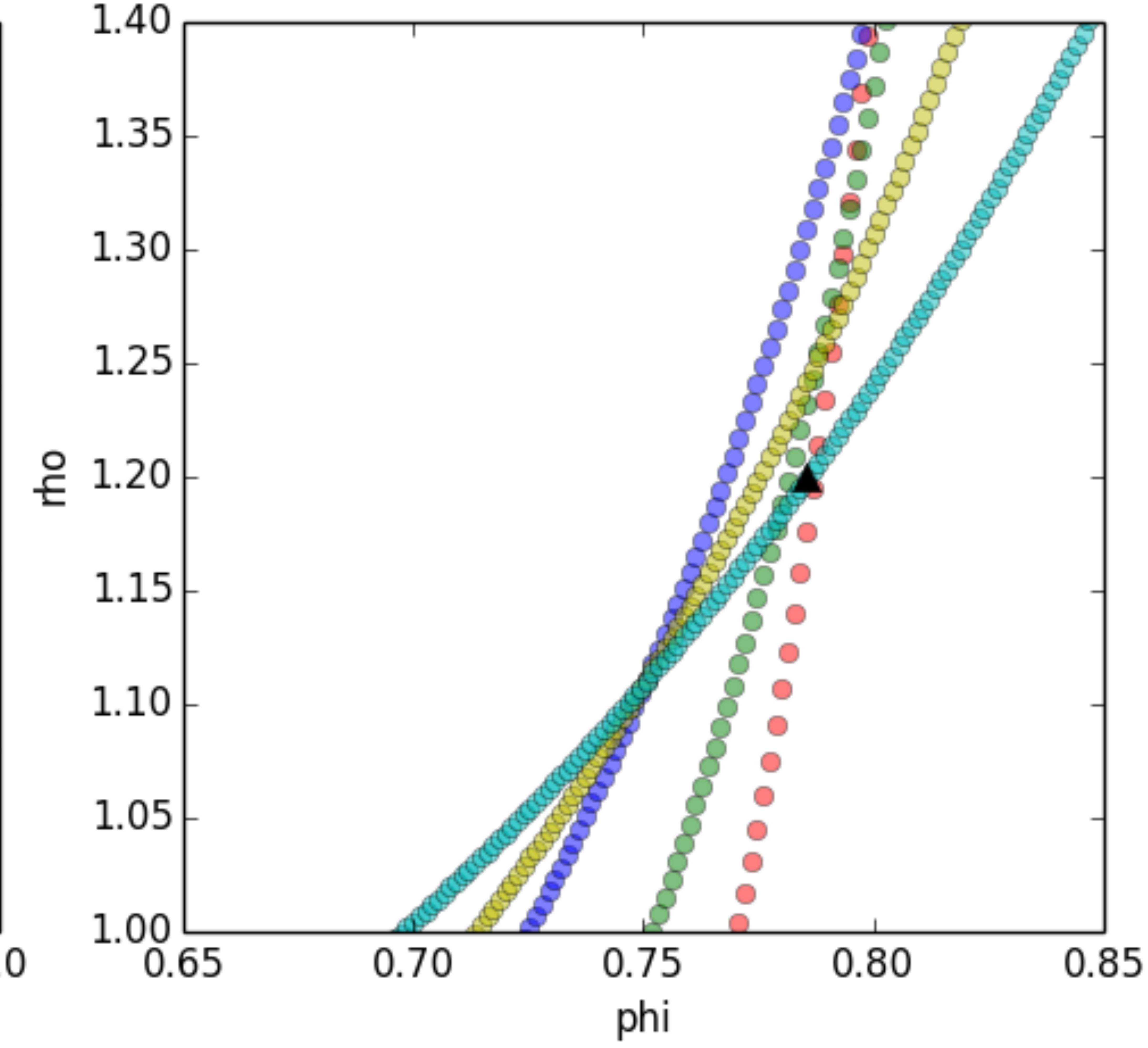


▲ … common (=true) solution compatible for all hits in (x,y) space

A. Salzburger, 2nd workshop on Tracking in Particle Physics Experiments, Huizhou, July 2025

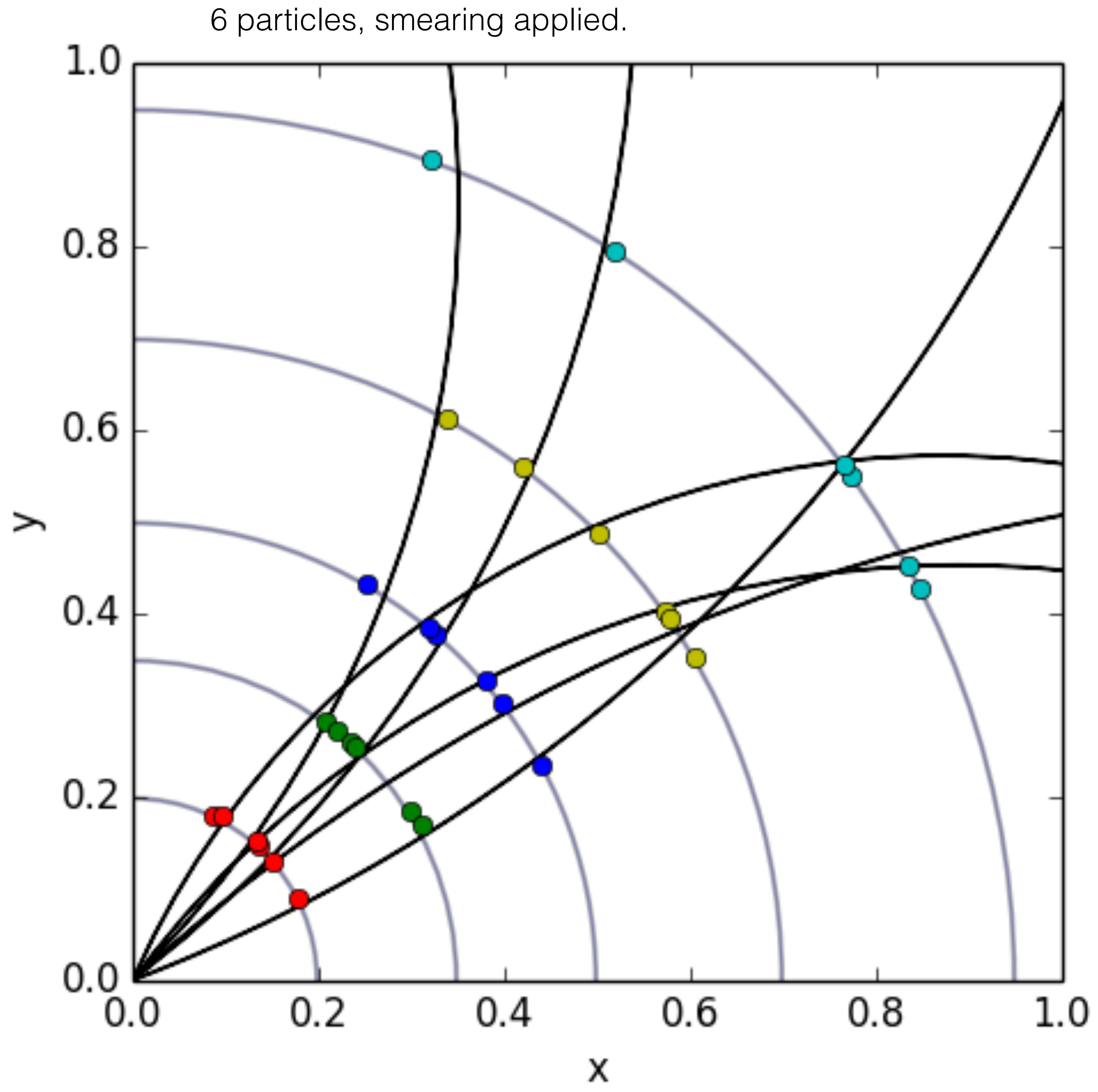
Probably not in the real world …

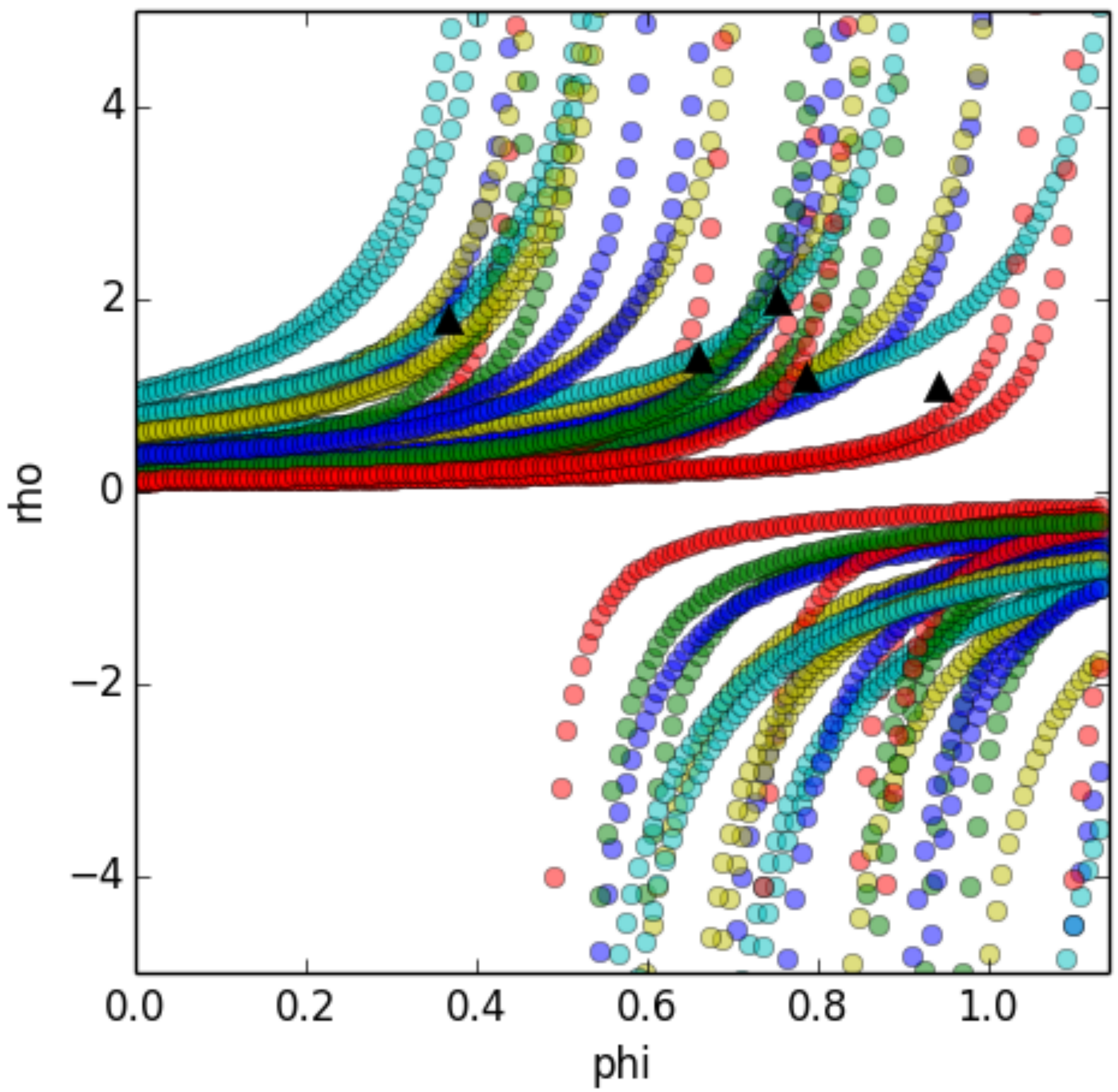




Material interaction (e.g. Scattering) & measurement uncertainty distort the picture …

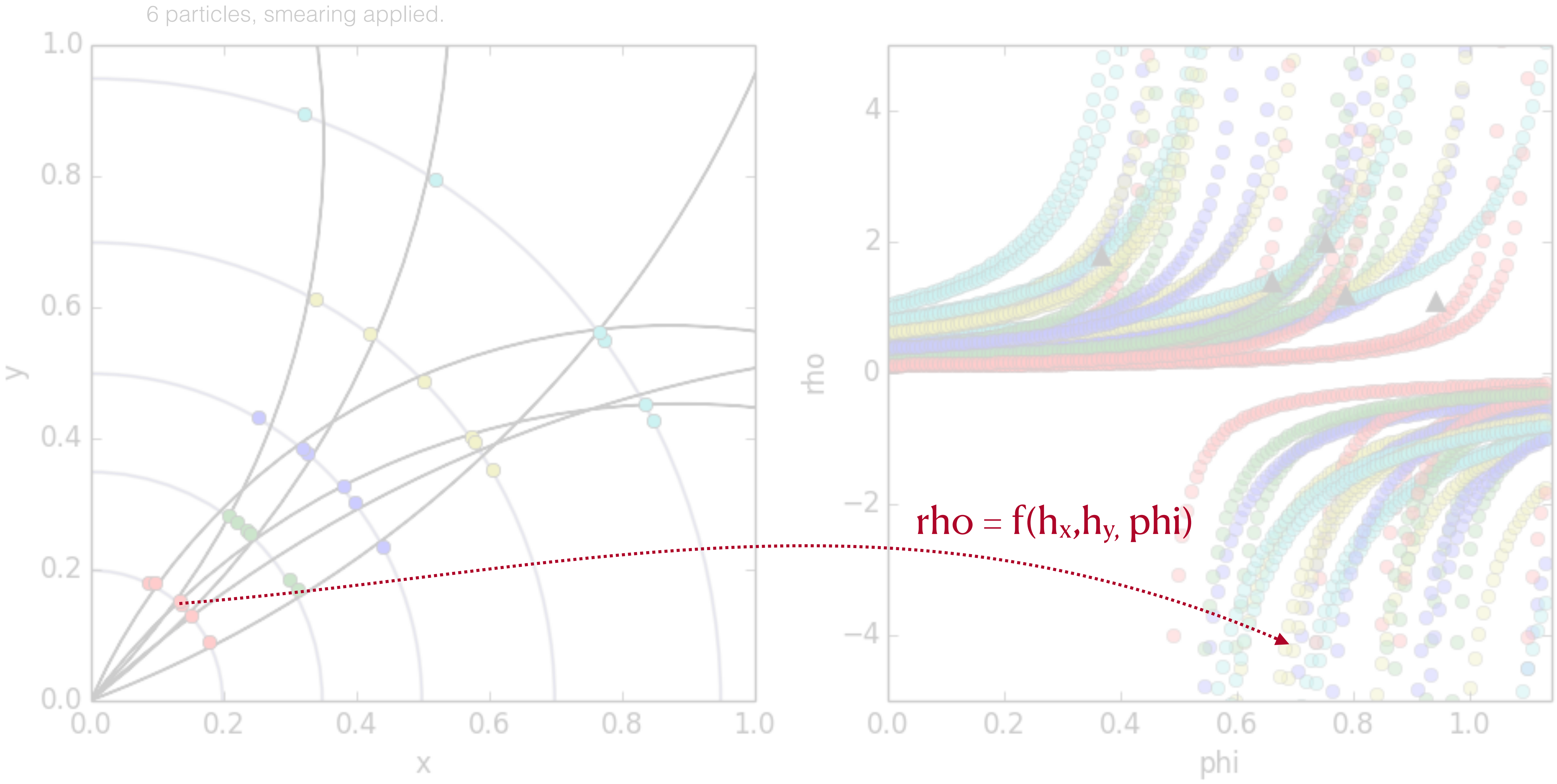
Probably not in the real world …





Material interaction (e.g. Scattering) & measurement uncertainty distort the picture …

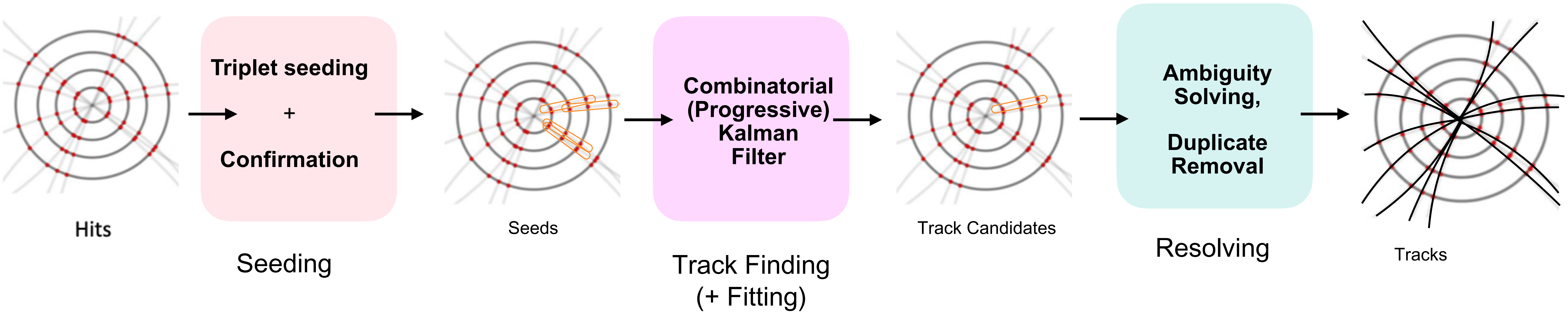
Or we just have to learn f ?



Metric learning has been attempted here and there in the community.

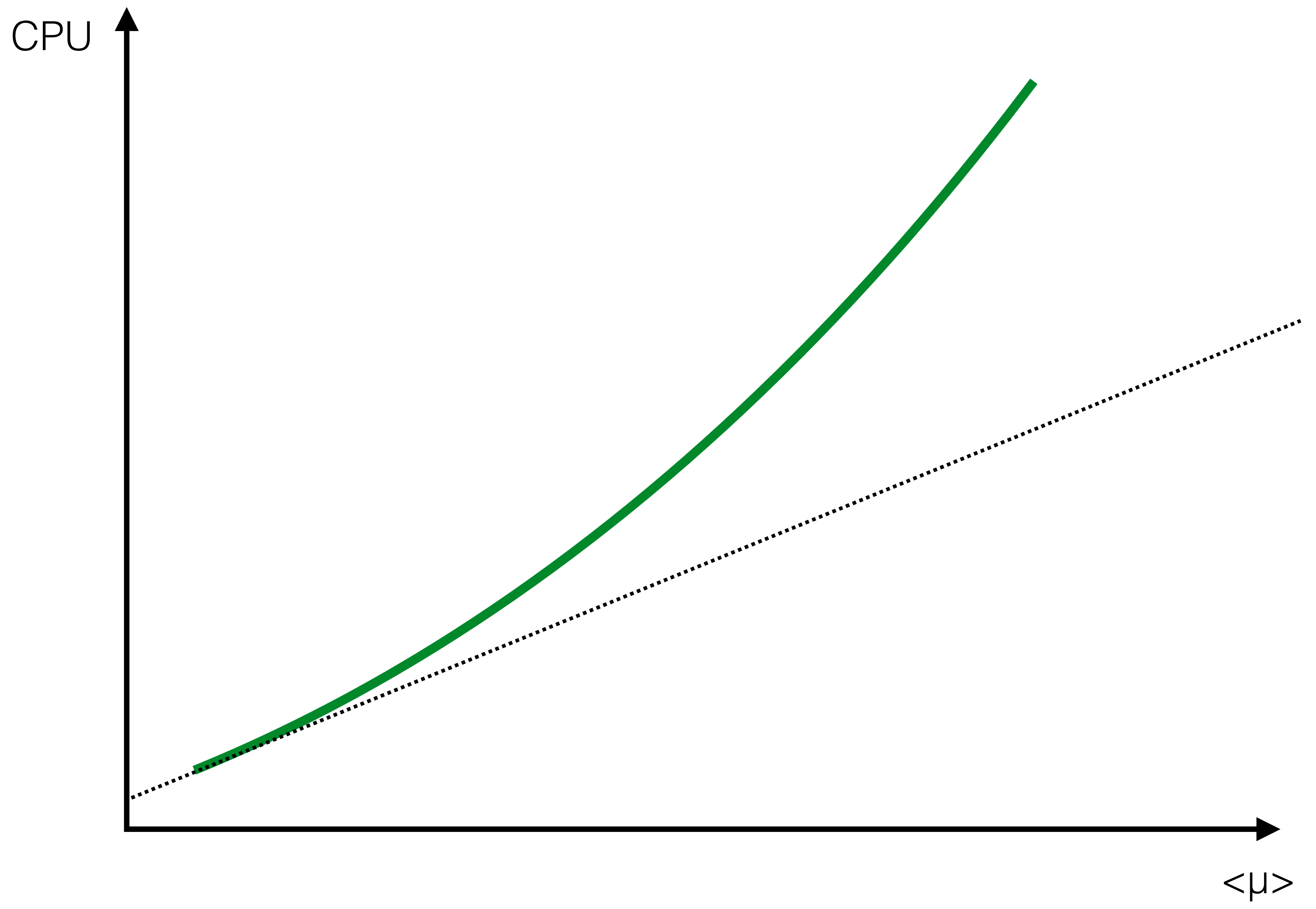
A classical reconstruction chain

Often current state of the art implementation

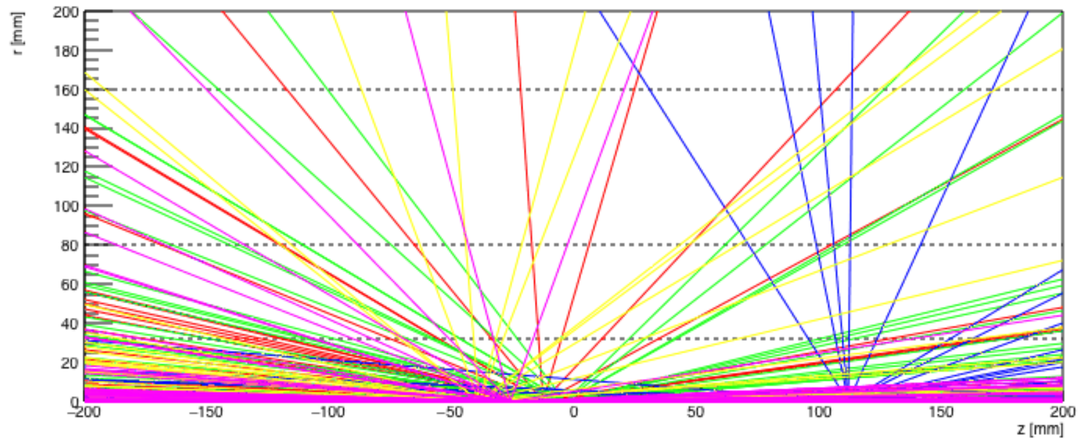


More or less the ATLAS Track reconstruction chain… with a little bit of ML sprinkled in.

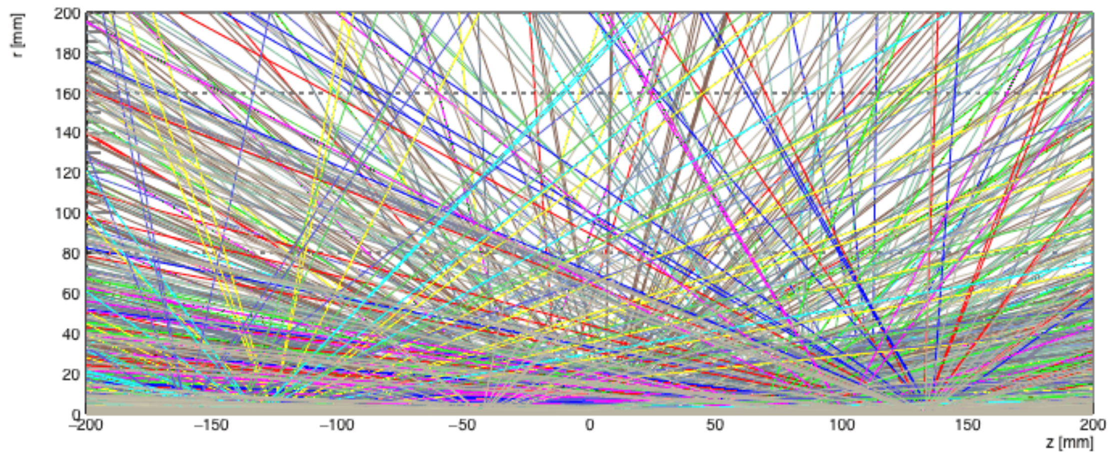
Classical algorithm* scaling



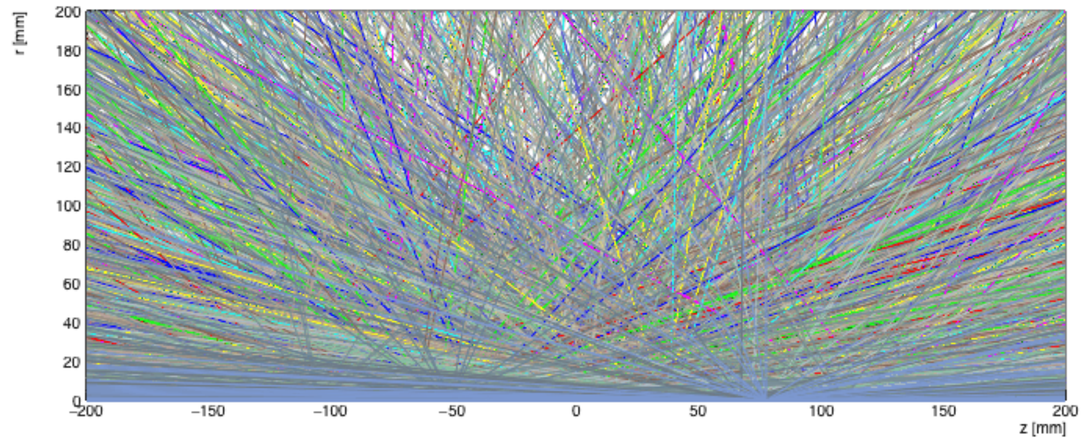
*combinatorial track search



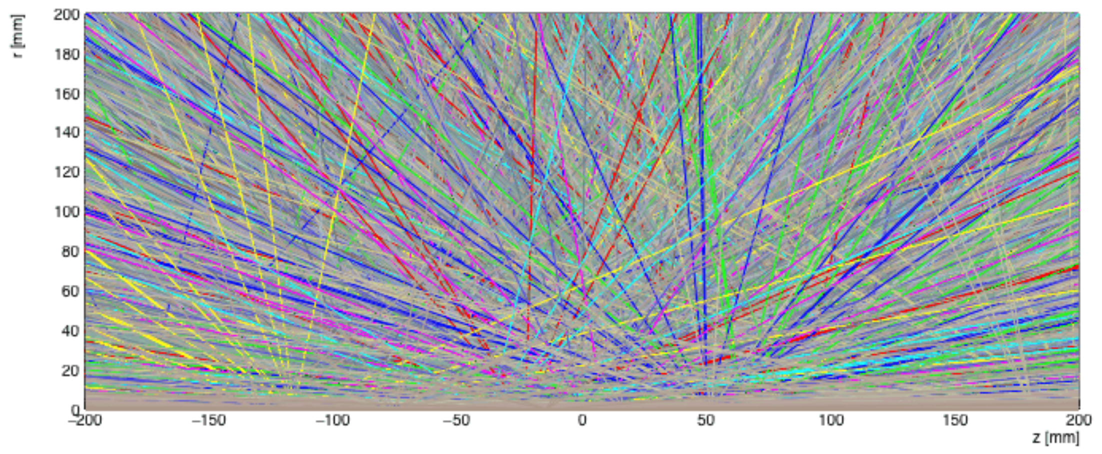
LHC Run-1, <μ> ~ 5



LHC Run-2, <μ> ~ 20

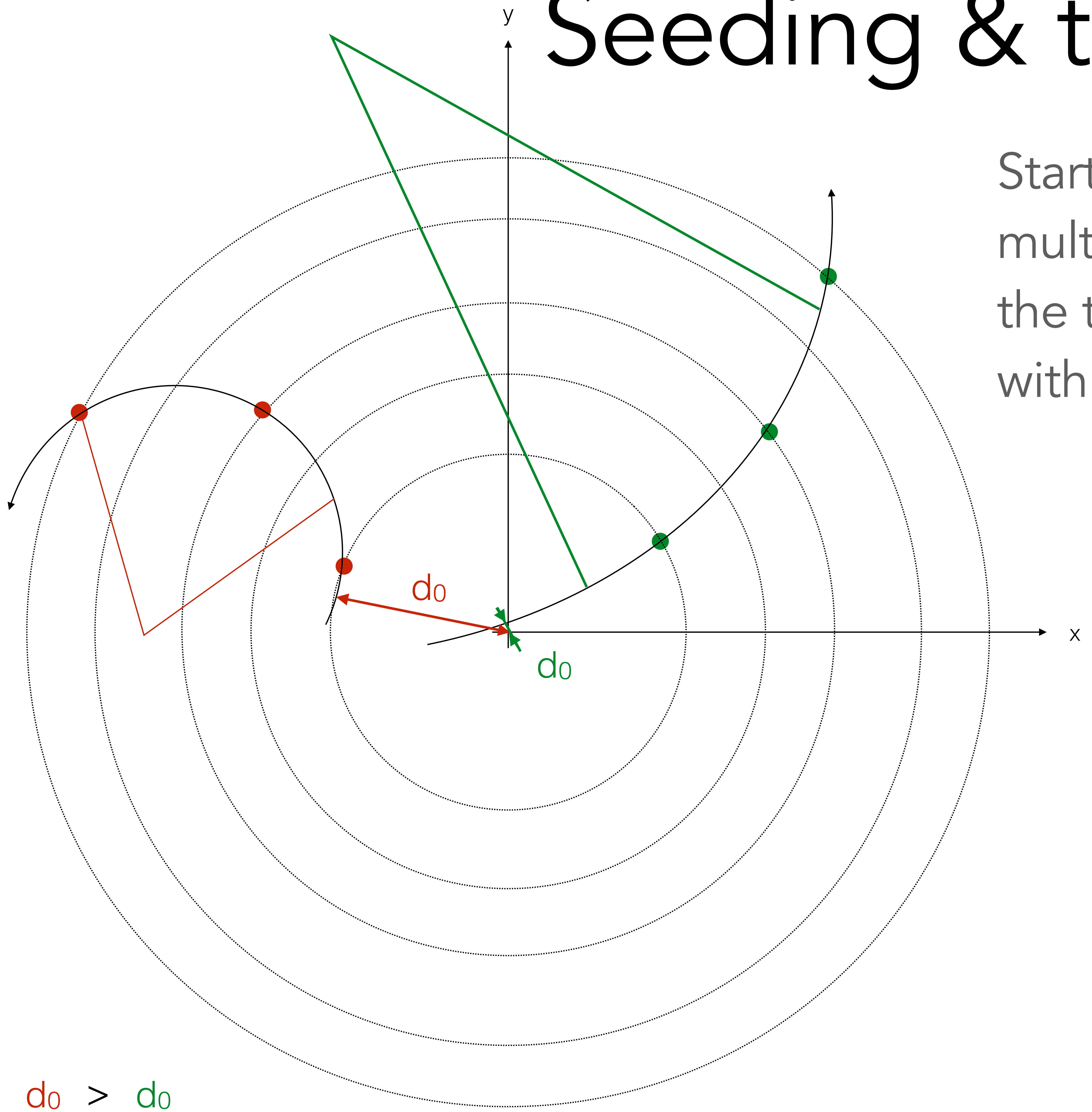


HL-LHC, <μ> ~ 200



FCC-hh (25ns) <μ> ~ 1000

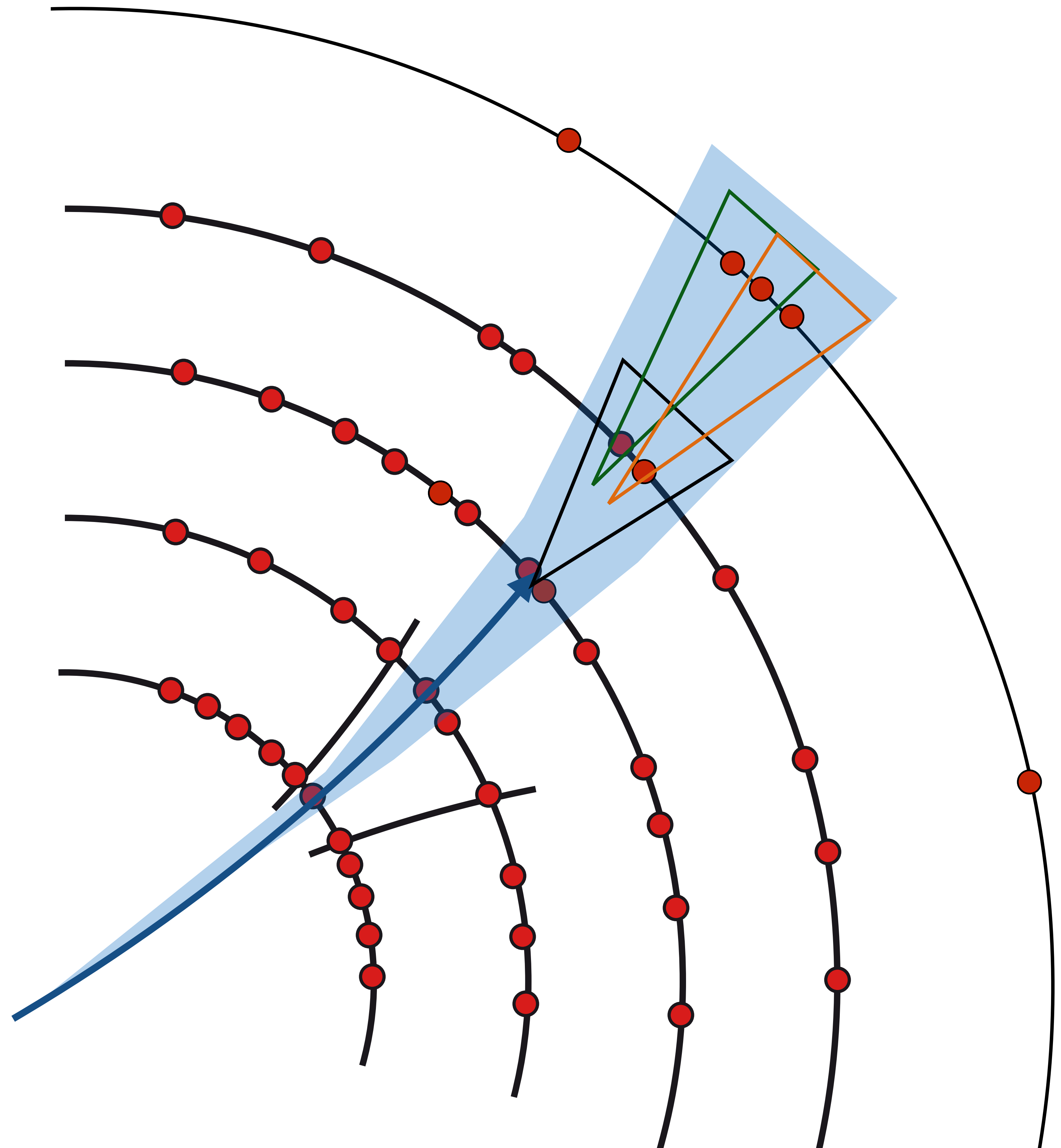
Seeding & track following



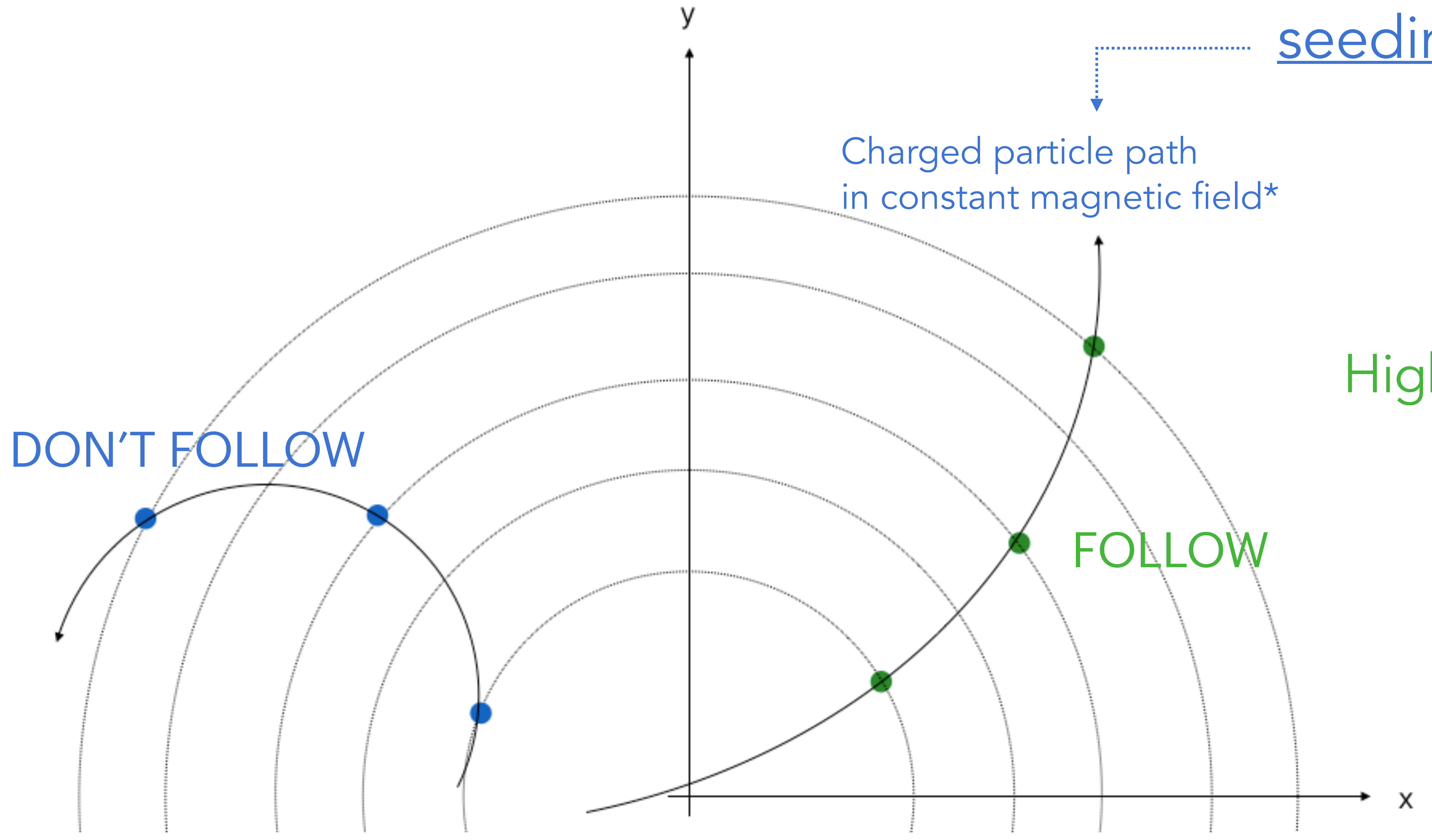
d_0 > d_0

p_T < p_T

Start finding track seeds, e.g. doublets, triplets, multiplets that are compatible with the track hypothesis, follow **promising ones** with a filter



Labelling: classification

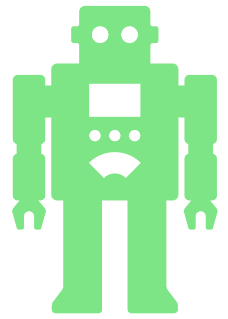


One classical approach to track finding is seeding & track following

CPU intensive

Highest purity of seeds required

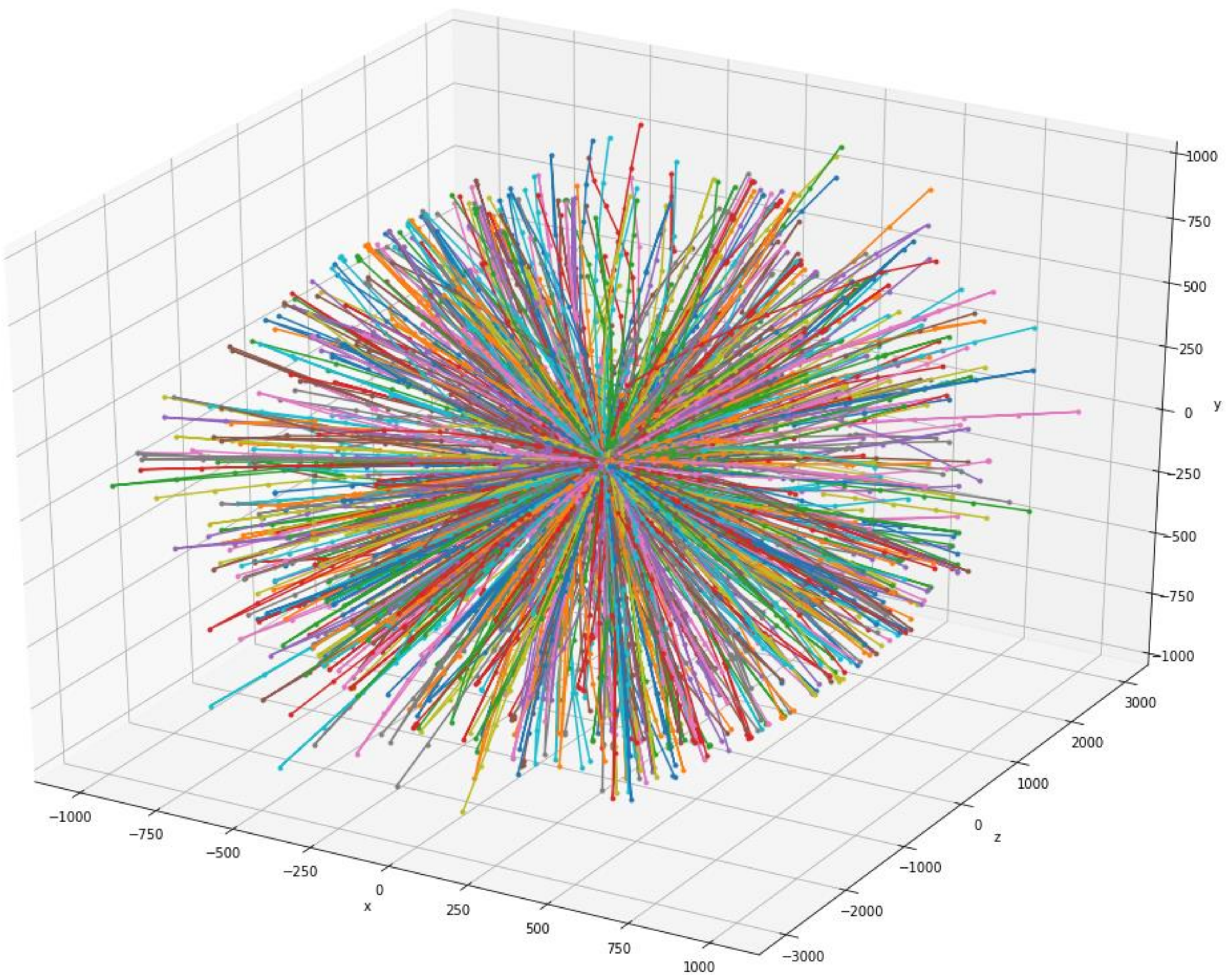
classification problem



*along z axis

Labelling: classification

Classification is perfect ML problem, replace cut & check technique



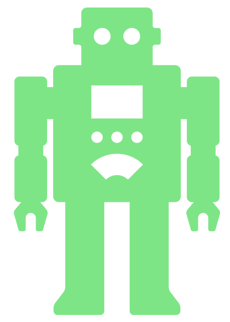
True particle tracks
from simulated event

created random training
dataset of **4-hit** combinations
with categories

- **good: 4/4** correct
- **medium: 3/4** correct
- **bad: <2/4** correct

$[x,y,z]_i$ 12 input features

N x M
hidden layers



good **medium** **bad** 3 output features

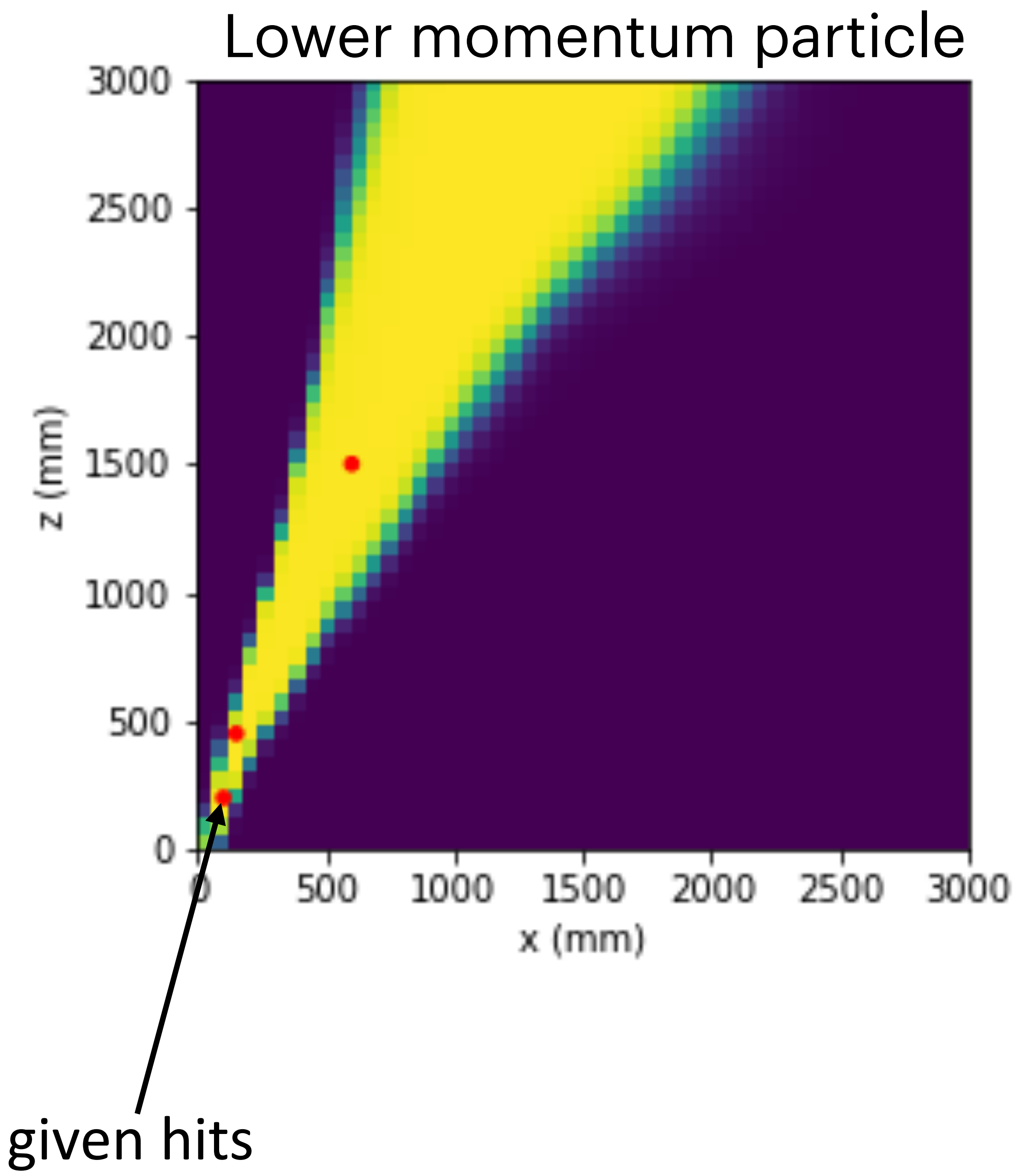
[F. Dietrich, E. Kneringer, AS : Track Seed Classification using NNs]

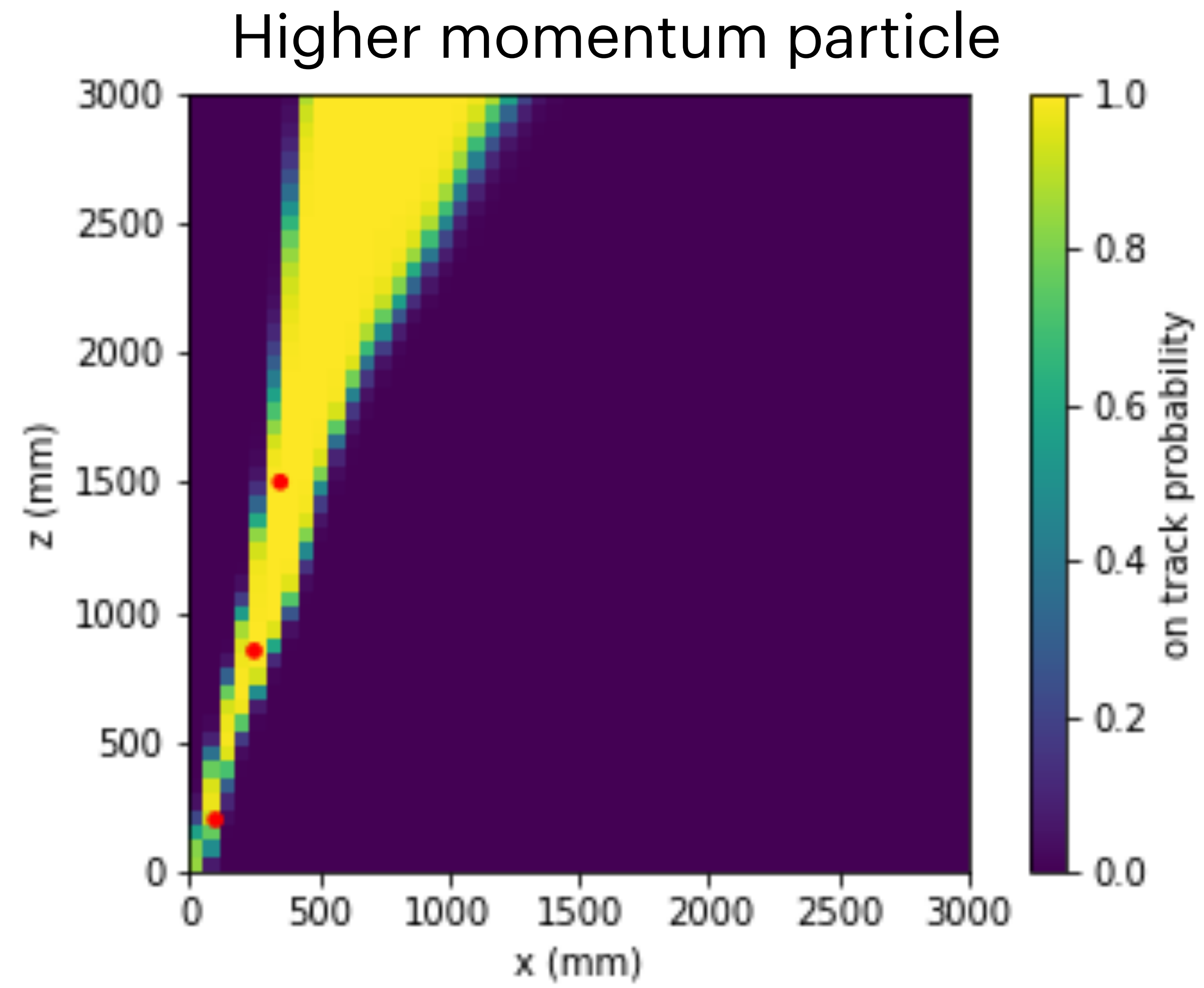
Labelling: classification

Powerful seed classification, here optimistic scenario
bad/medium training seeds created by distorting good seeds

Predicted Class

Actual Class	good	med	bad
good	98.5%	1.5%	0.1%
med	3.5%	95.7%	0.8%
bad	0.2%	3.2%	96.7%

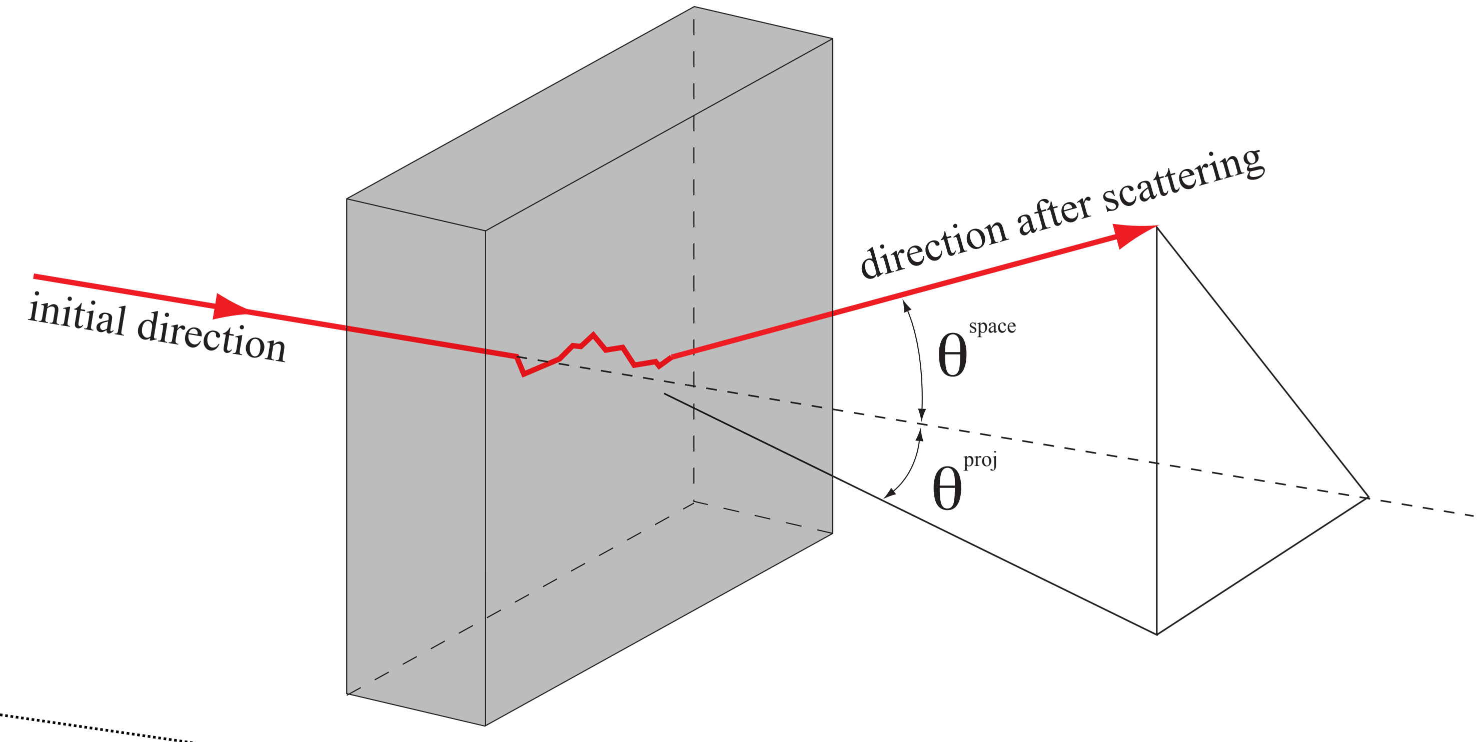




[F. Dietrich, E. Kneringer, AS : Track Seed Classification using NNs]

Labelling: classification & physics insight

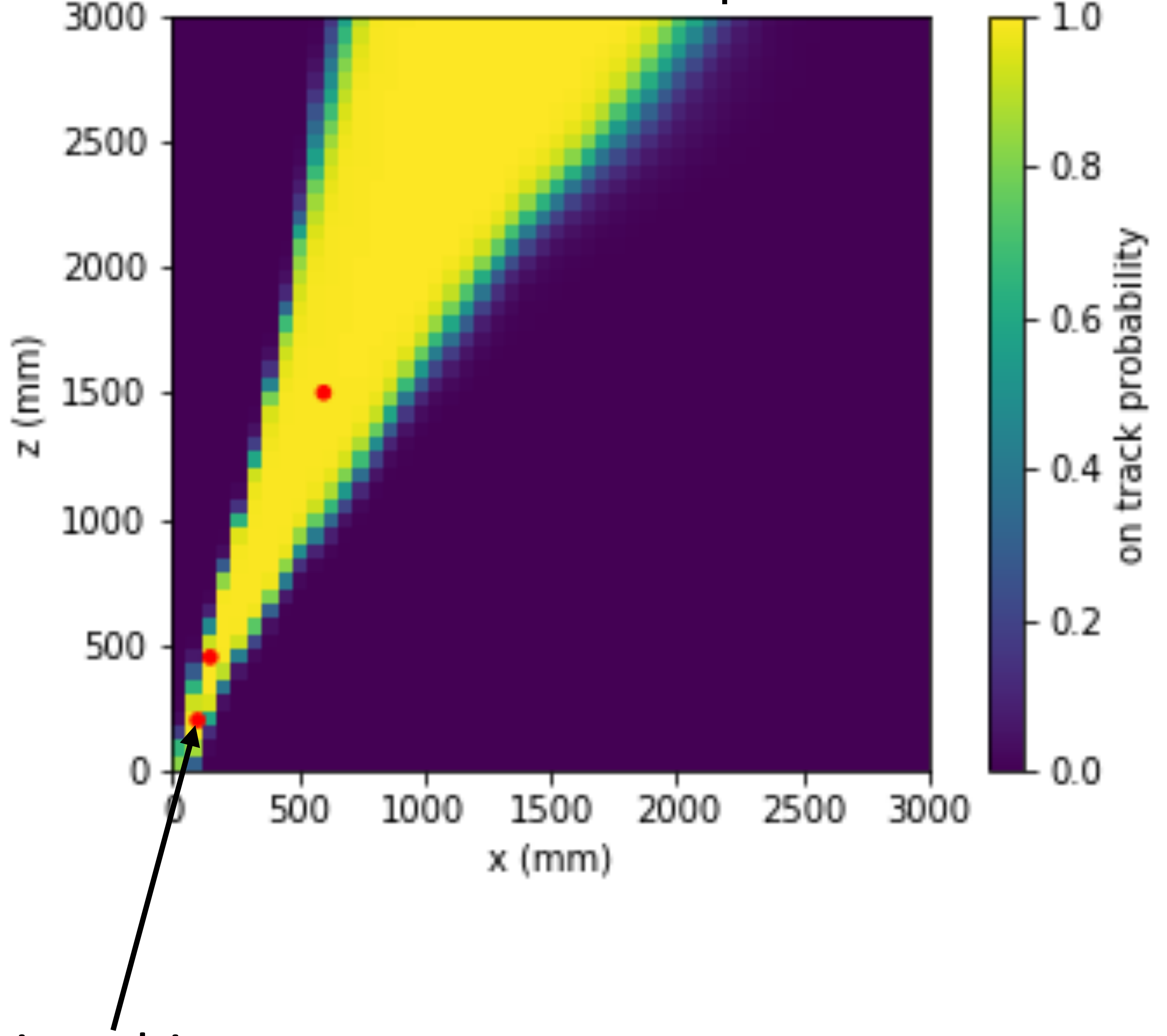
$$\theta_0 = \frac{13.6\ \mathrm{MeV}}{\beta c p}\, z \sqrt{x/X_0}\left[1 + 0.038 \ln(x/X_0)\right]$$



Predicted Class

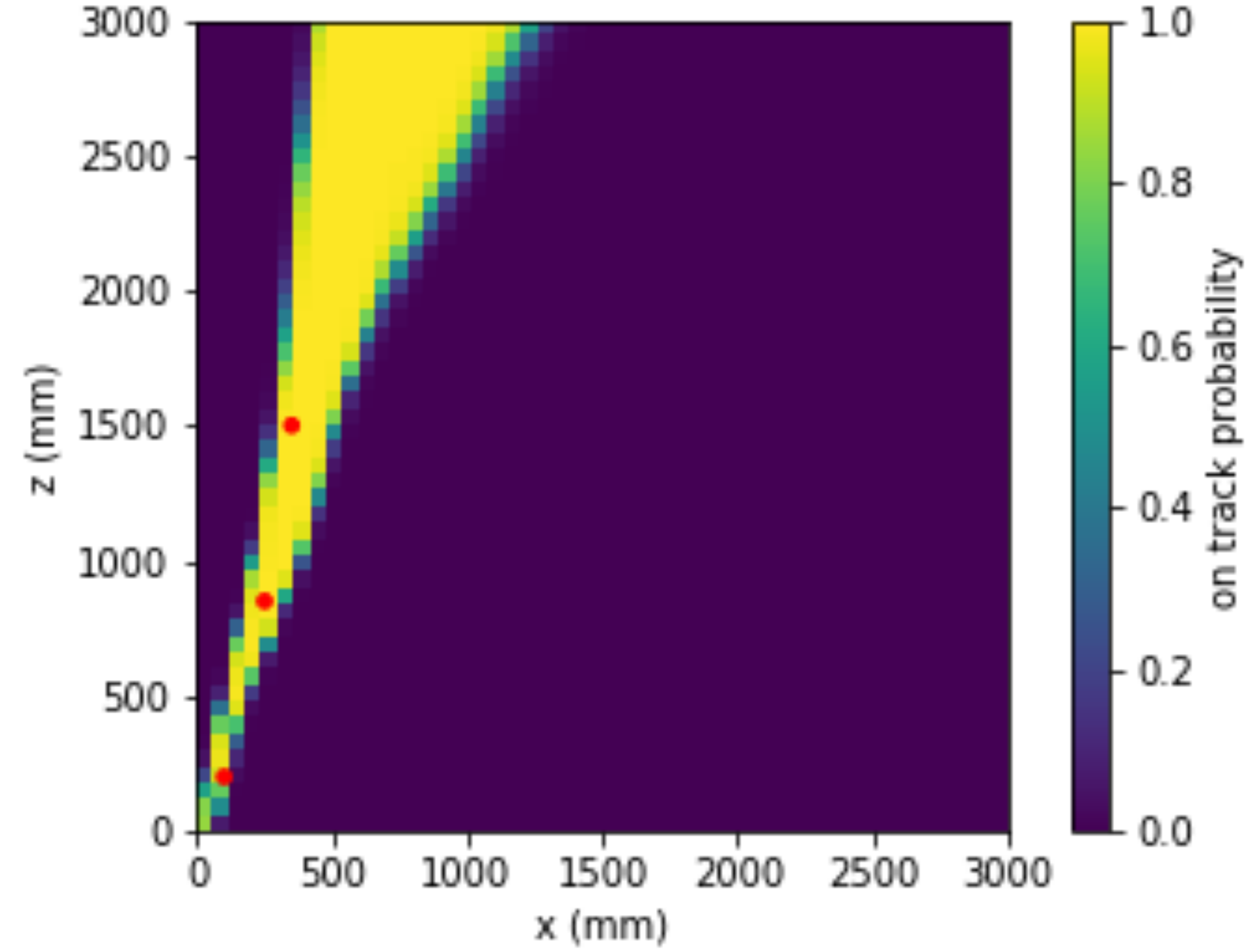
Actual Class	good	med	bad
good	98.5%	1.5%	0.1%
med	3.5%	95.7%	0.8%
bad	0.2%	3.2%	96.7%

Lower momentum particle



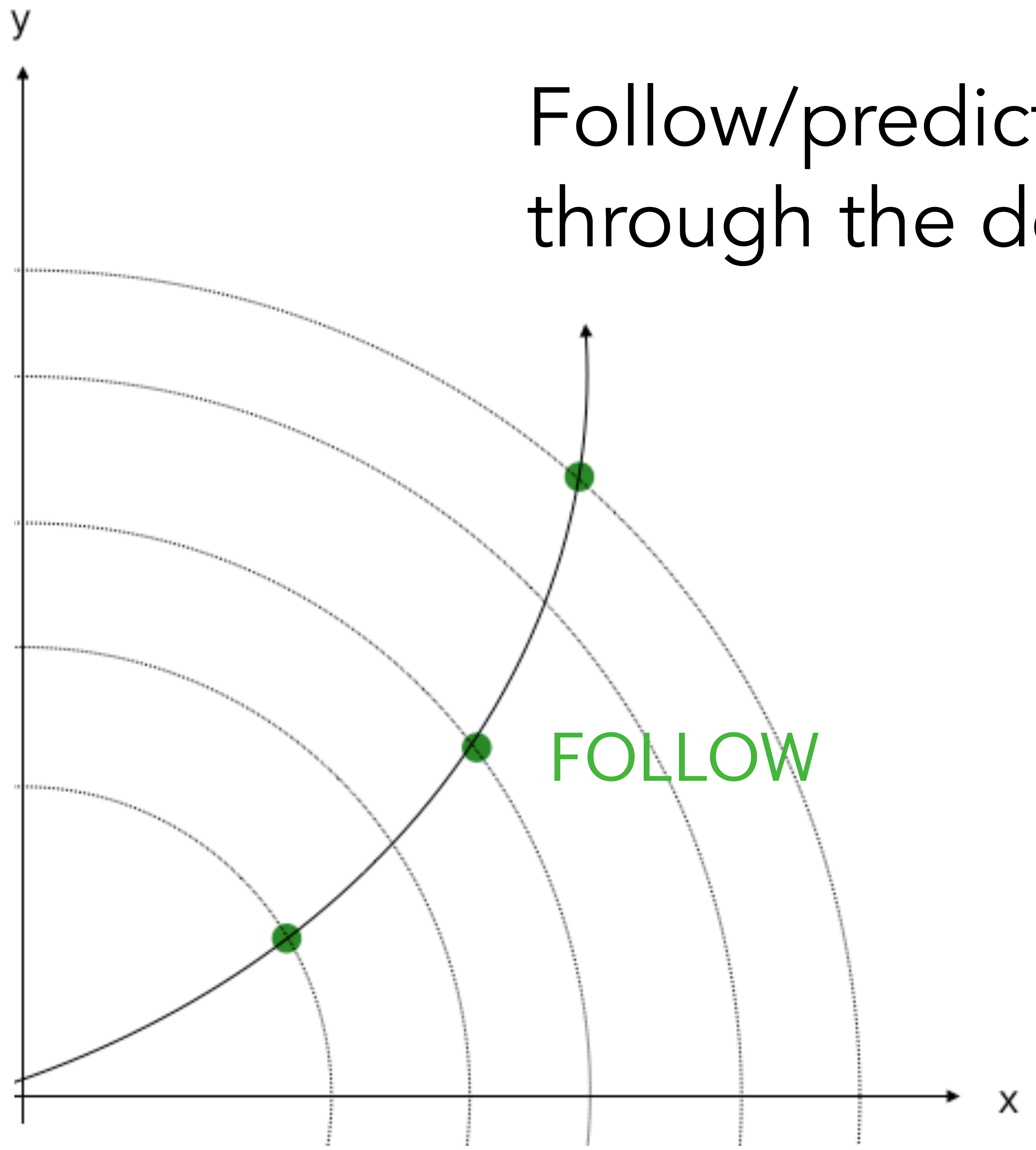
given hits

Higher momentum particle

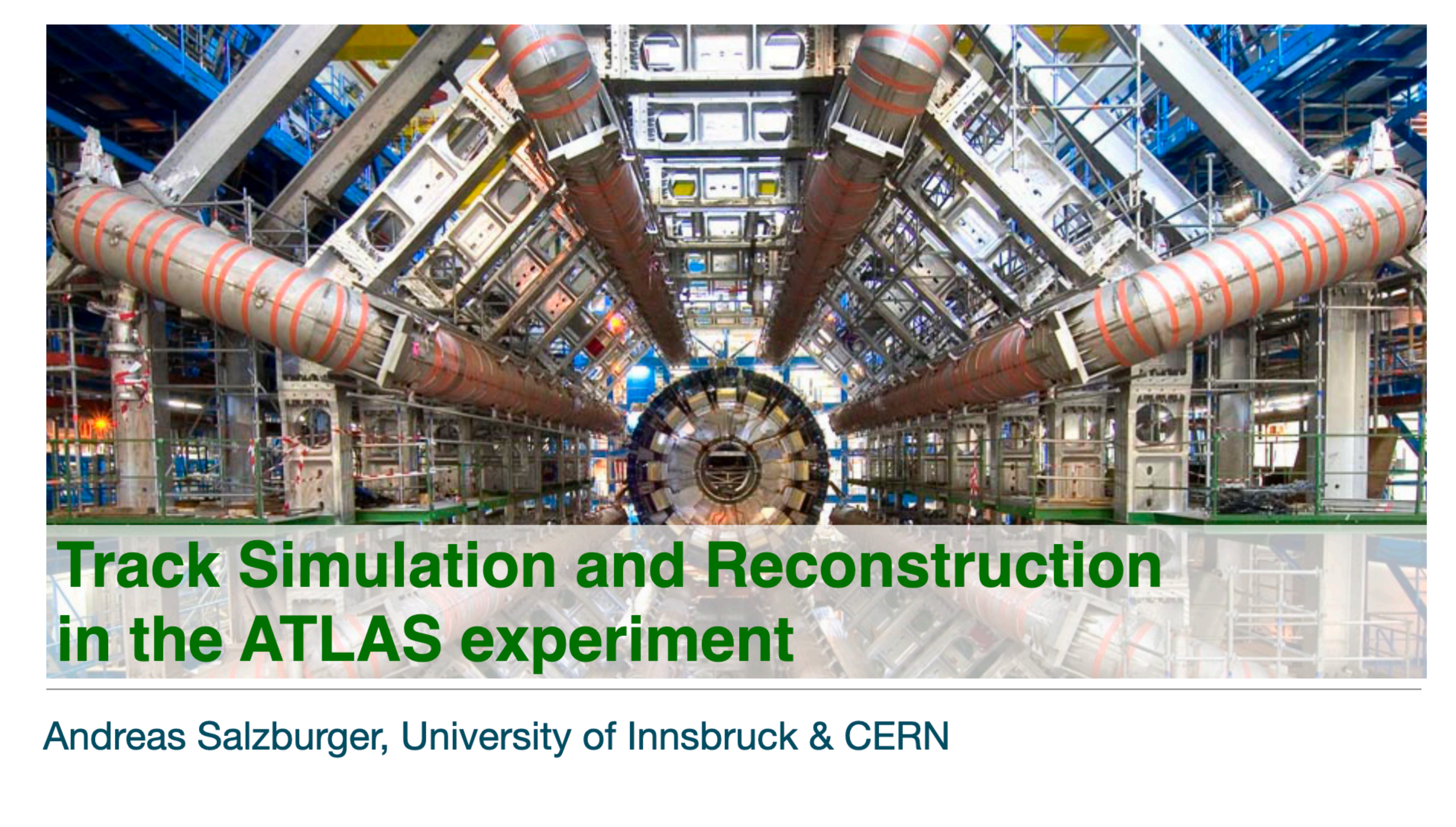


[F. Dietrich, E. Kneringer, AS : Track Seed Classification using NNs]

Predictions: a case study



Follow/predict the trajectory through the detector

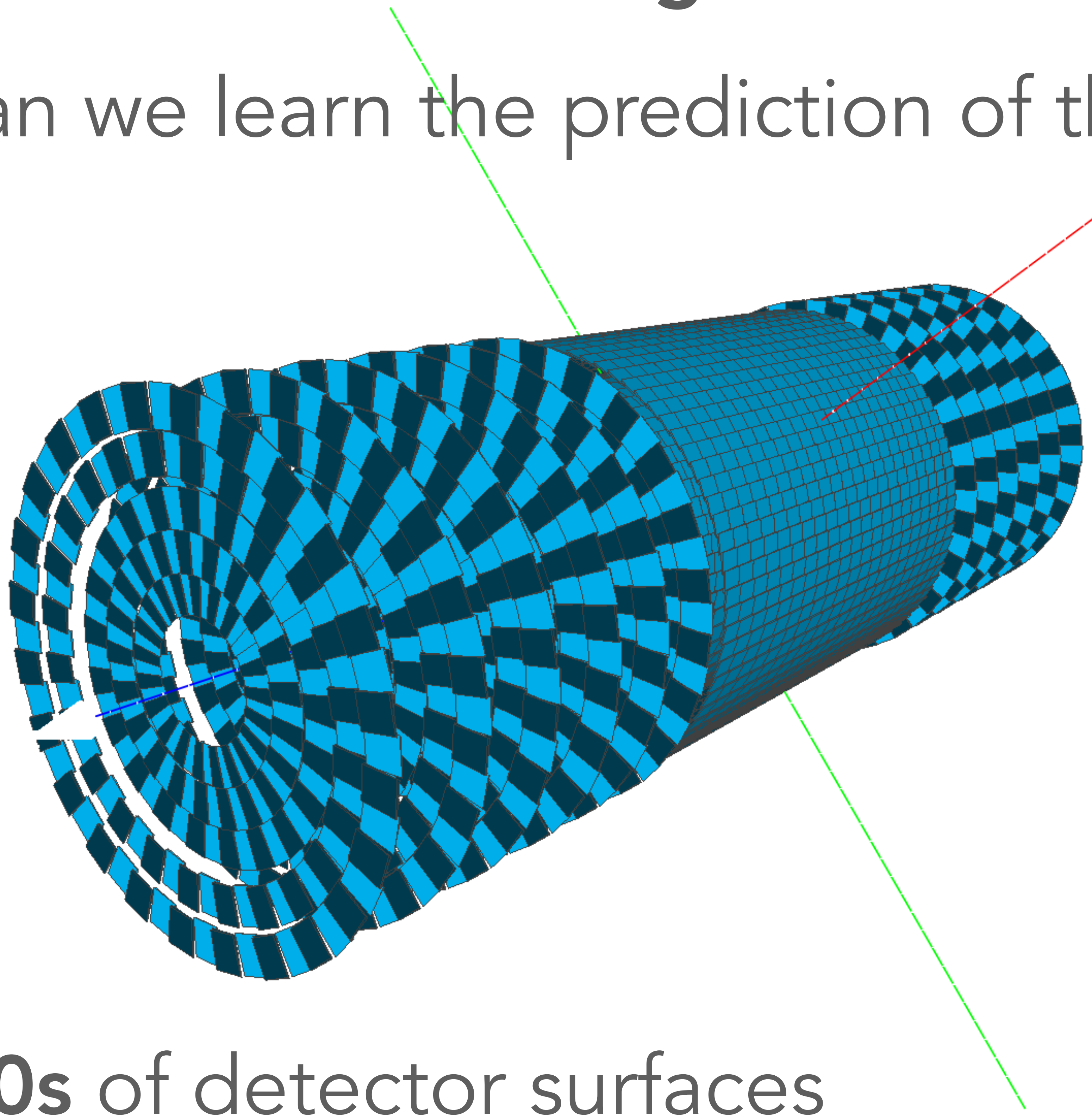


[AS: PhD Thesis]

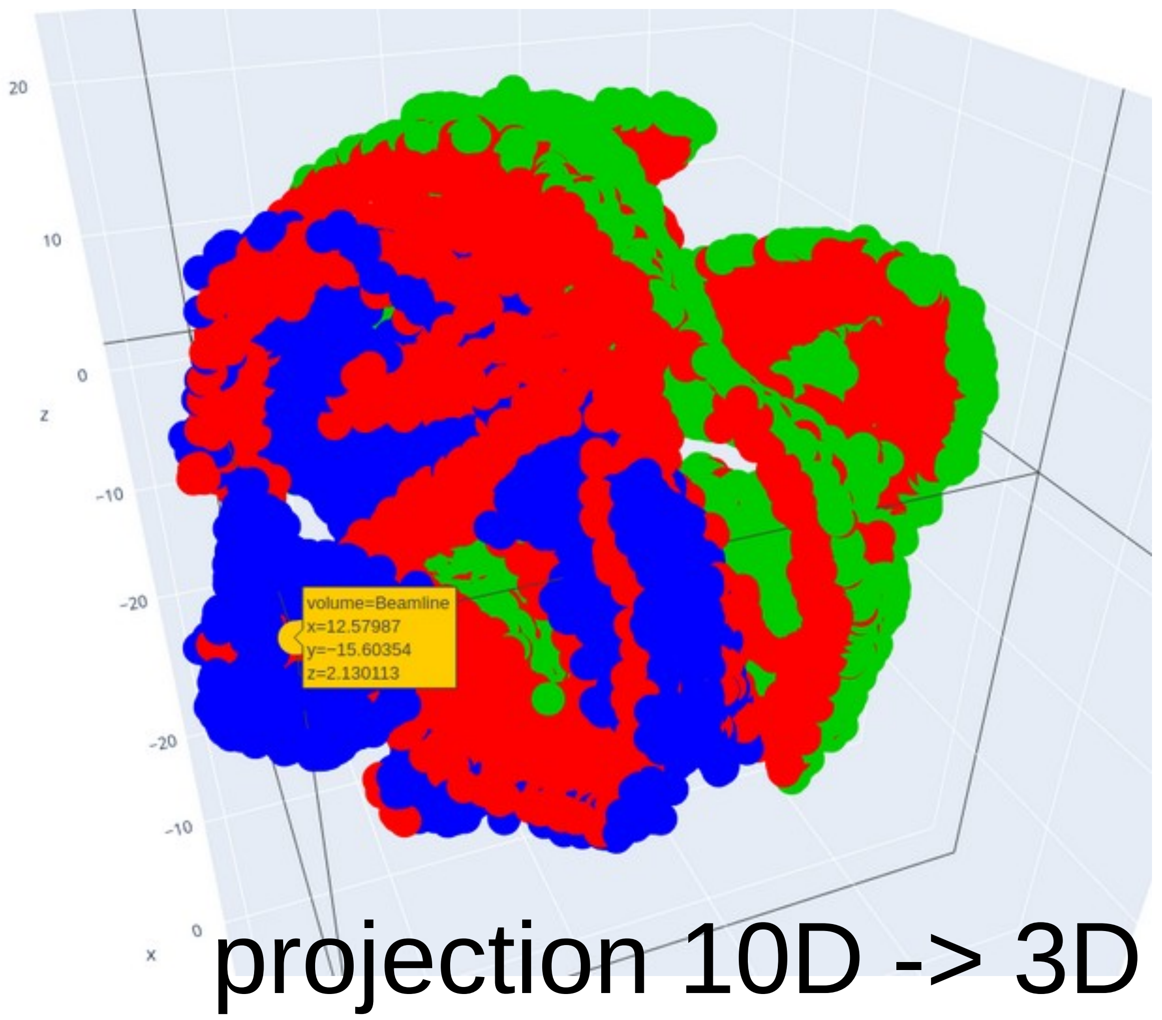
Predictions: a case study

Instead of **hand-crafting** a detector description & navigation

- can we learn the prediction of the **next** detector surface?

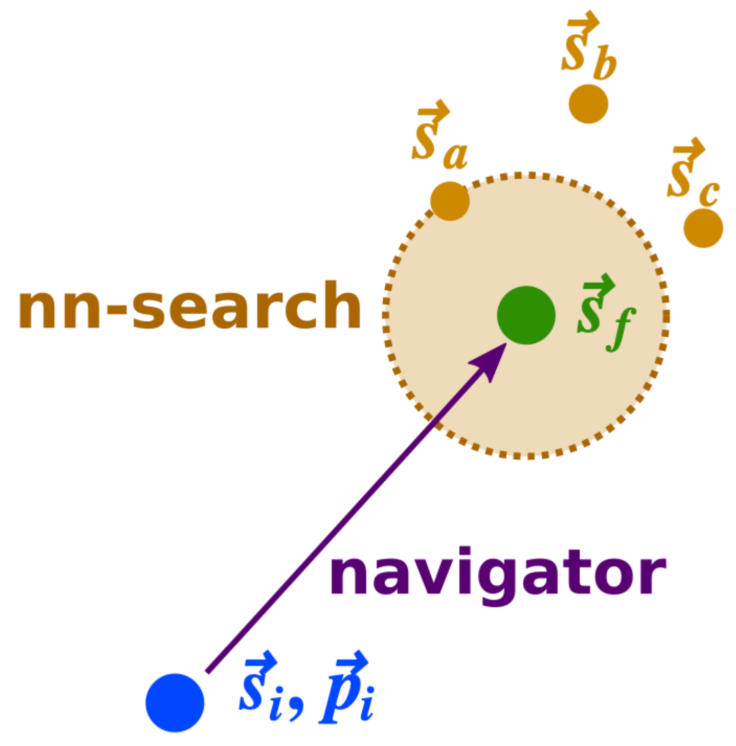


1000s of detector surfaces

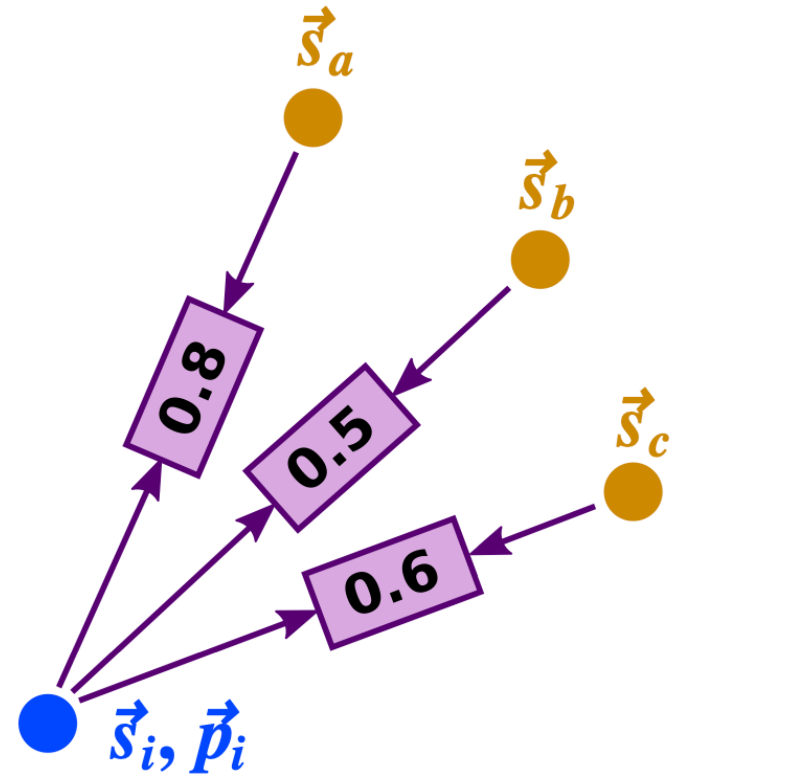


Embedding space

a) Target prediction:



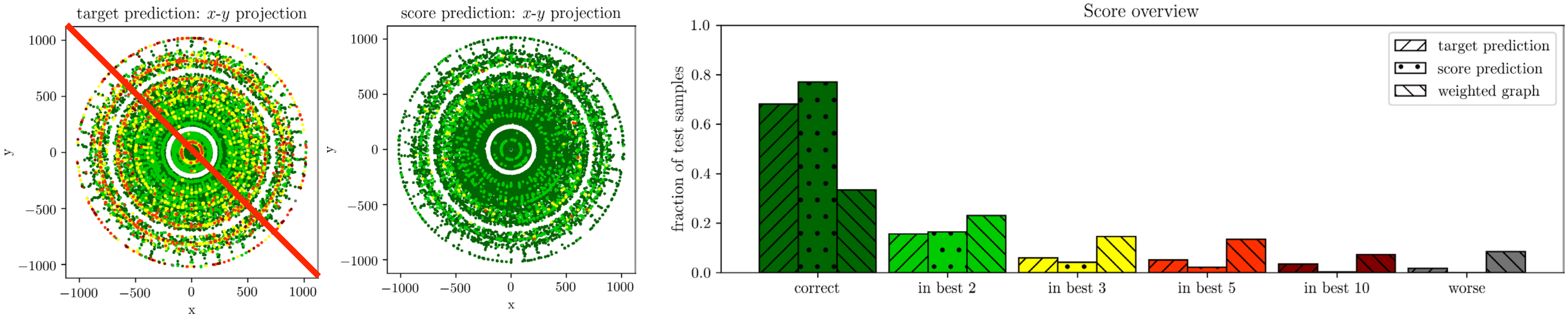
b) Target score prediction:



Prediction

[Huth, Wettig, AS: EPJ Web Conf. 251 (2021) 03053]

Predictions: a case study & a lesson



Score **prediction** model gave some reasonable **performance…**

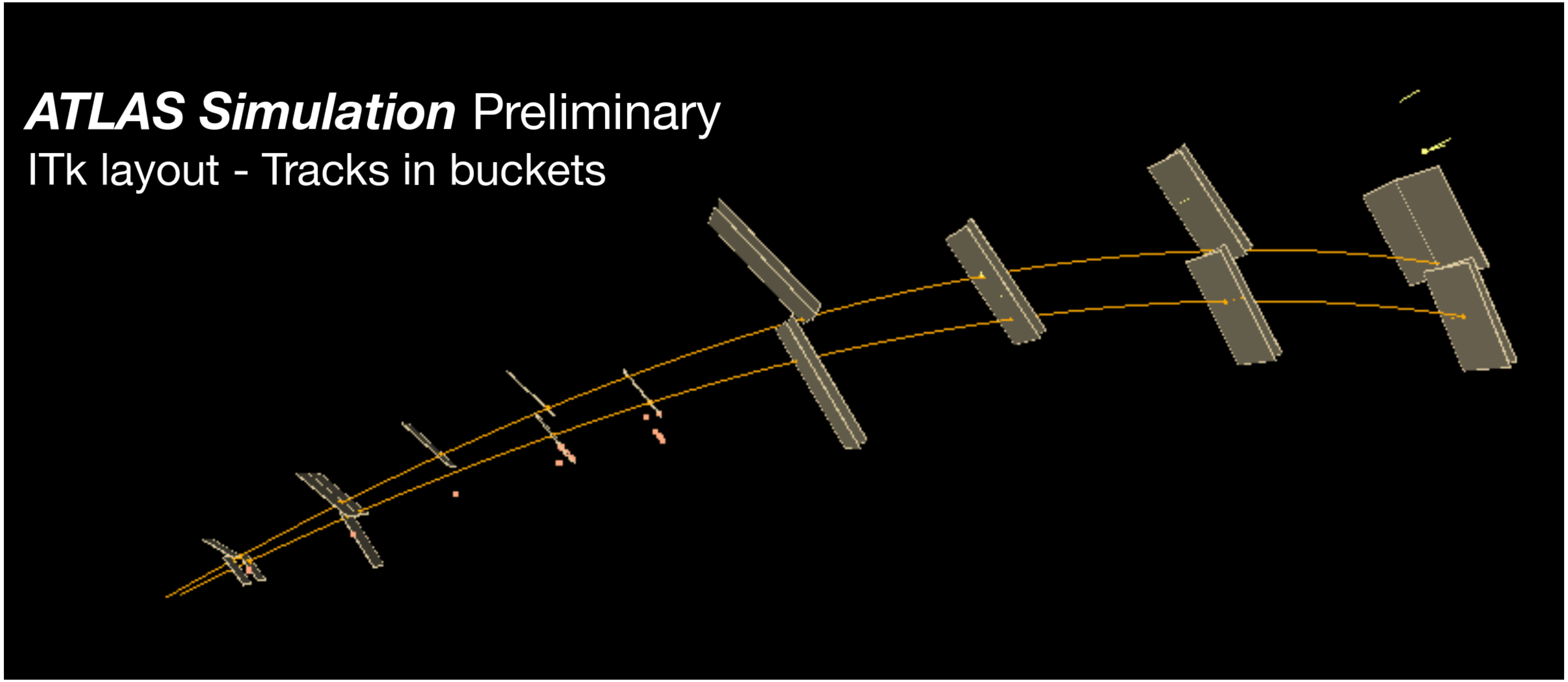
… **if** we searched in the **best 10 predicted**

fed into **original navigator** code (based on successive straight **line intersection tests**)

Just to find out, that runs now slower & is less precise.

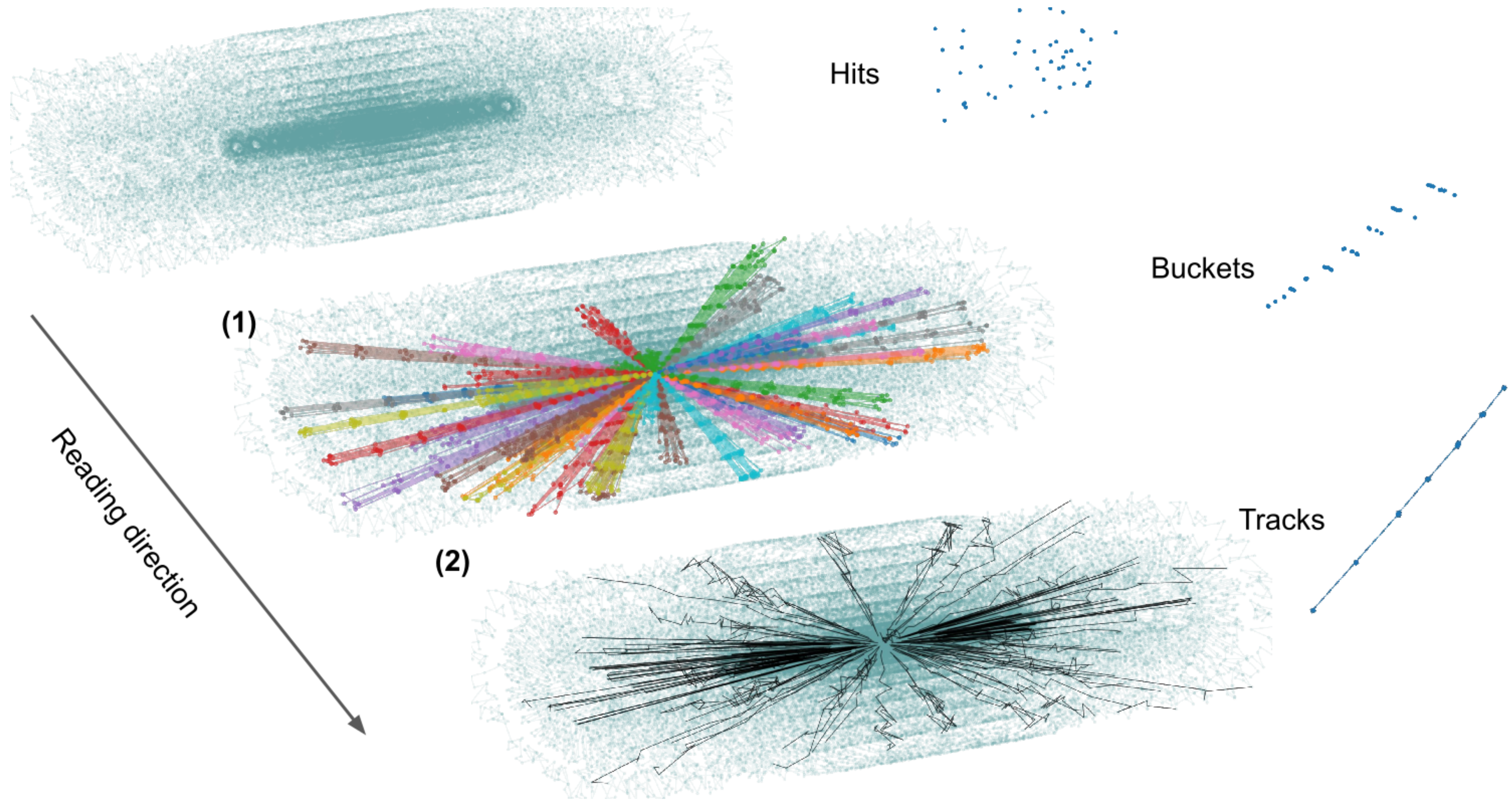
[Huth, Wettig, AS: EPJ Web Conf. 251 (2021) 03053]

Labelling: Music Neighbours



Trajectories from simulated particles in the ATLAS upgrade tracker, **found** with (the help of) **Spotify**

Labelling: Music Neighbours



[S. Amrouche, T. Golling, M. Kiehn, AS: Music, Neighbours & Tracking]
[S. Amrouche, N. Calace, T. Golling, M. Kiehm. AS : Hashing & similarity learning]

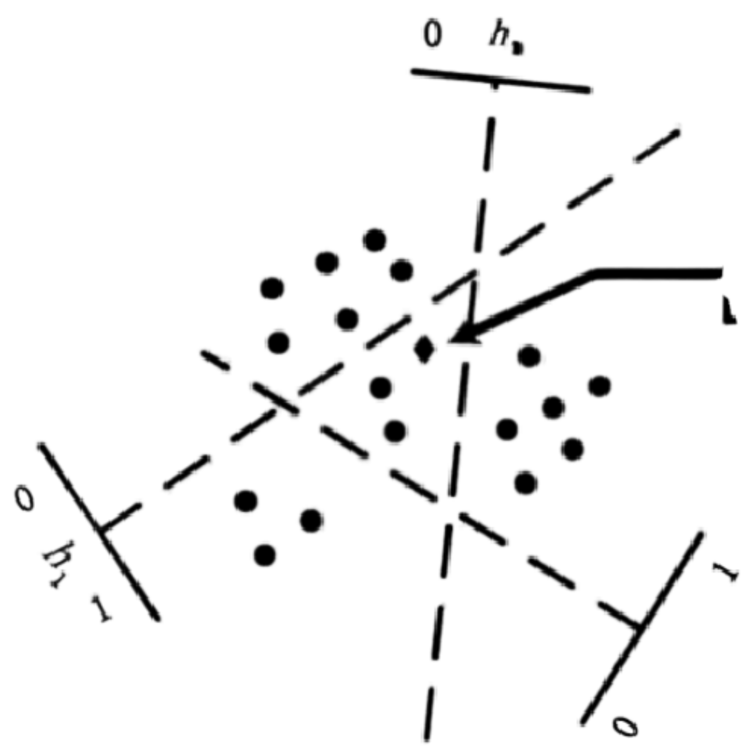
Labelling: Music Neighbours

Perfect hash function would solve the tracking problem

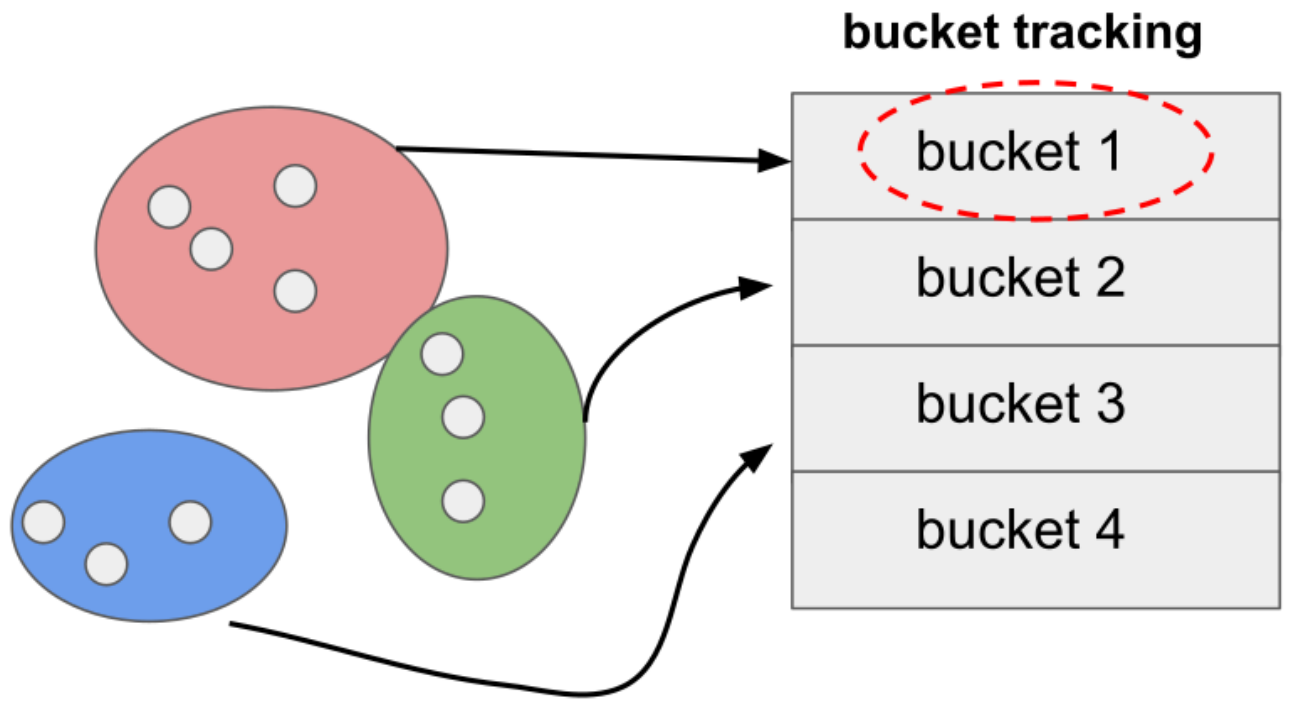
```
h(hit) = track number
```

Approximate hashing, however, can be done

```
h(track 1, hit 0) = group x
h(track 1, hit 1) = group x
h(track 0, hit 1) = group x
```



RADNOM PROJECTIONS

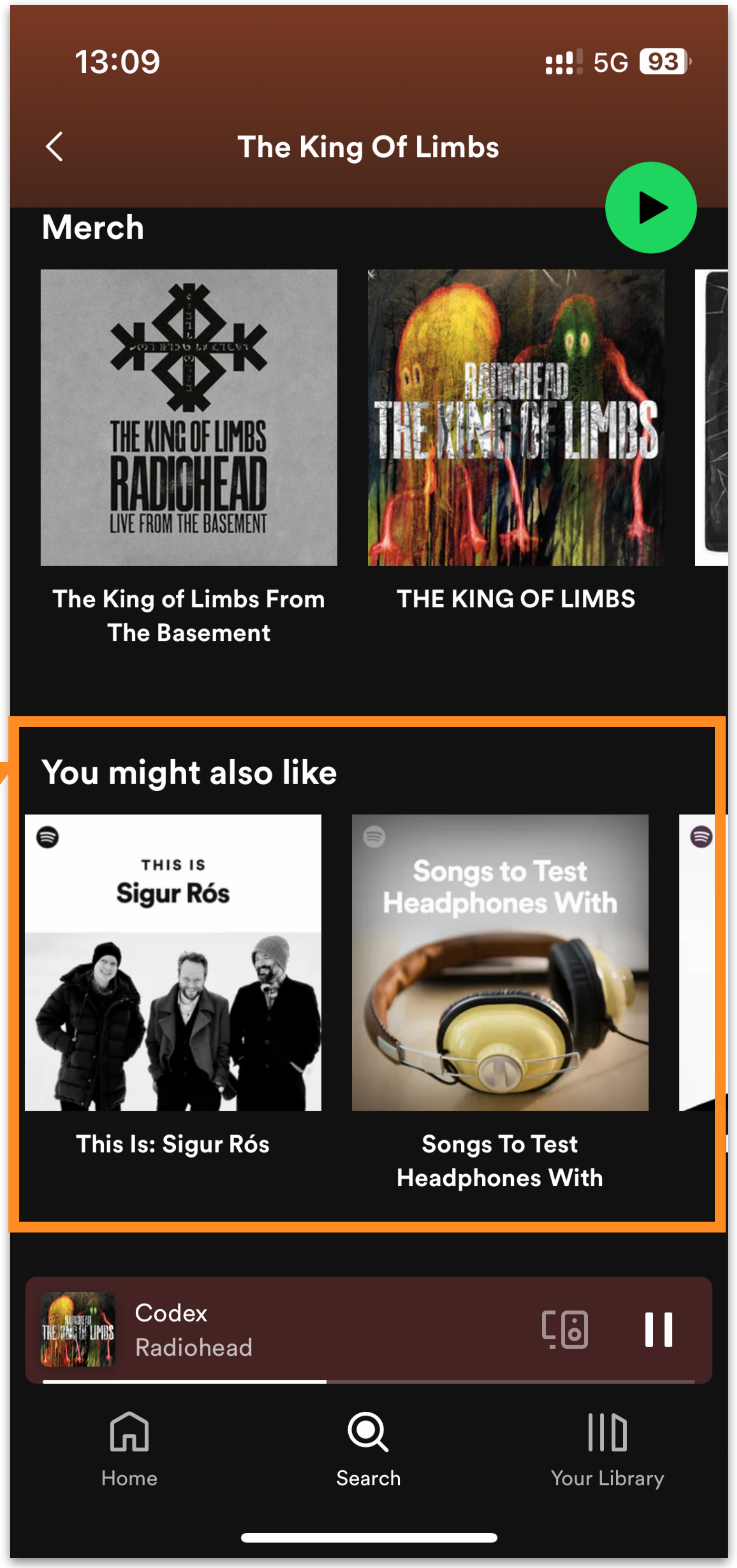


APPROXIMATE NEAREST NEIGHBOURS

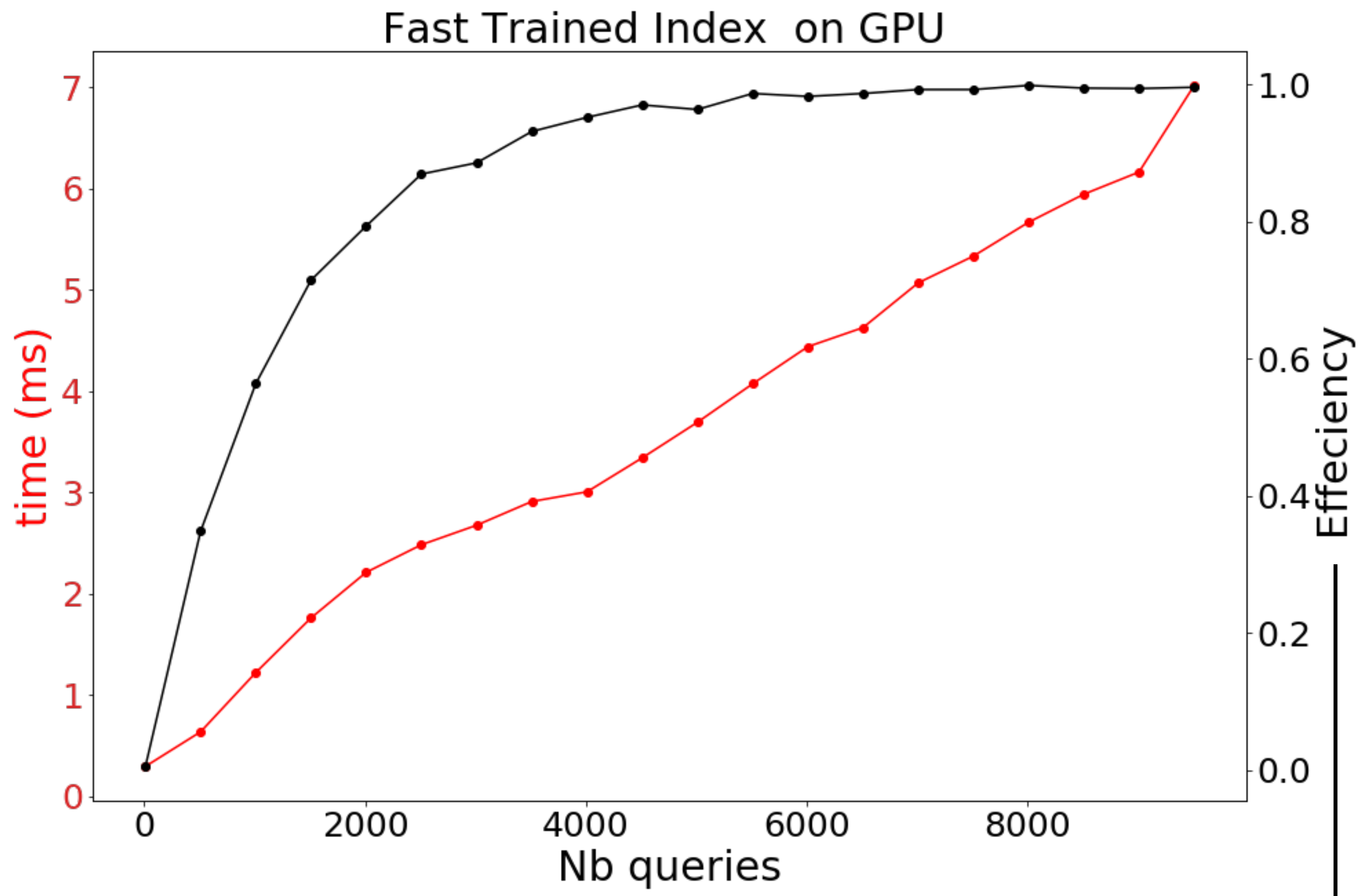
[S. Amrouche, T. Golling, M. Kiehn, AS: Music, Neighbours & Tracking]
[S. Amrouche, N. Calace, T. Golling, M. Kiehm. AS : Hashing & similarity learning]

Labelling: Music Neighbours

Spotify's approximate nearest neighbourhood library: [ANNOY]



Industry/open source libraries offer quite some **potential** also for science applications

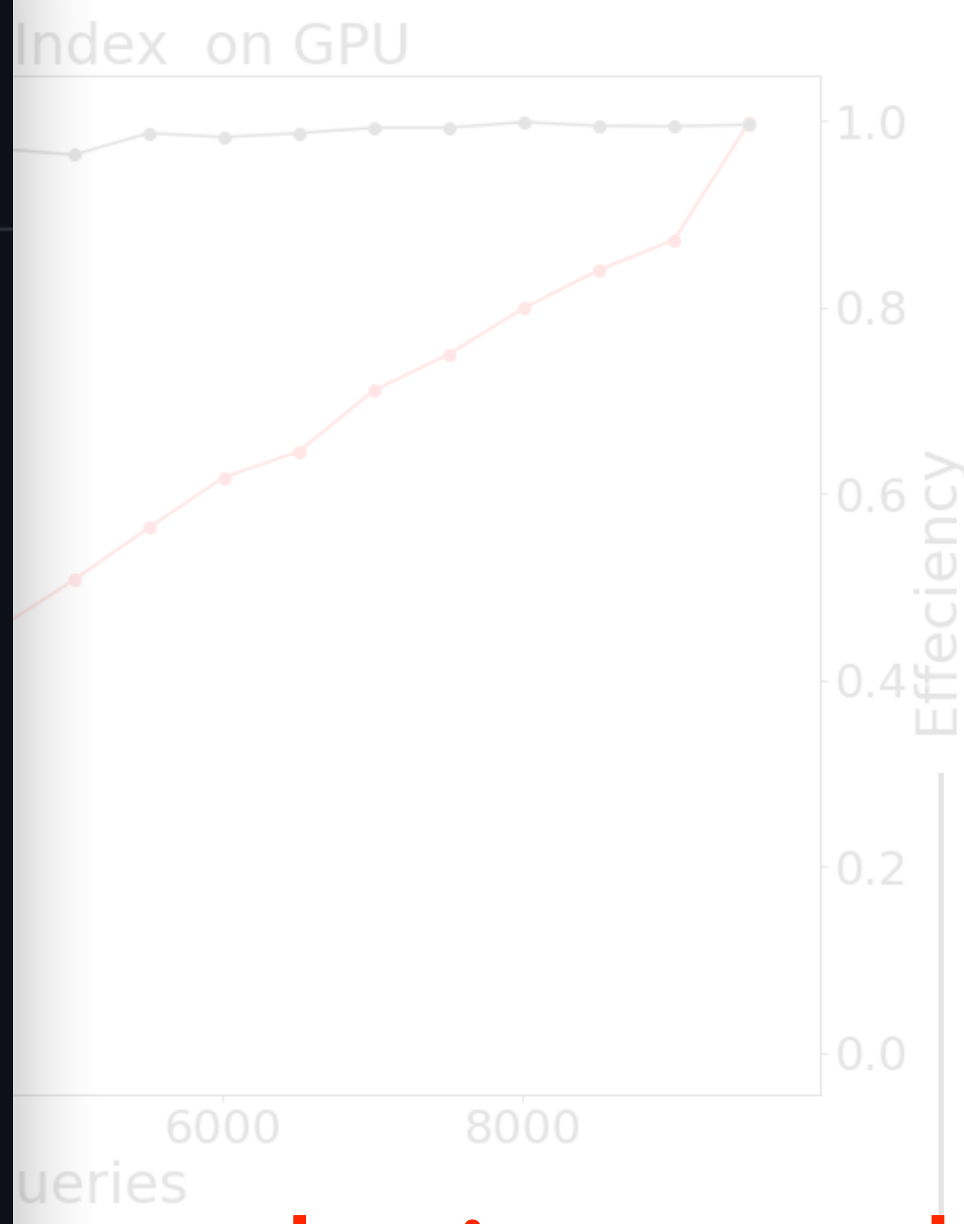


To find a bucket with at least 4/hits of the track contained (good enough for track seeding)

[S. Amrouche, T. Golling, M. Kiehn, AS: Music, Neighbours & Tracking]
[S. Amrouche, N. Calace, T. Golling, M. Kiehm. AS : Hashing & similarity learning]

Labelling: Music Neighbours

Industry/open source libraries offer quite some **potential** also for science applications, **but …**



.. no business model!

(In other words)

To find a bucket with at least 4/hits of the track contained (good enough for track seeding)

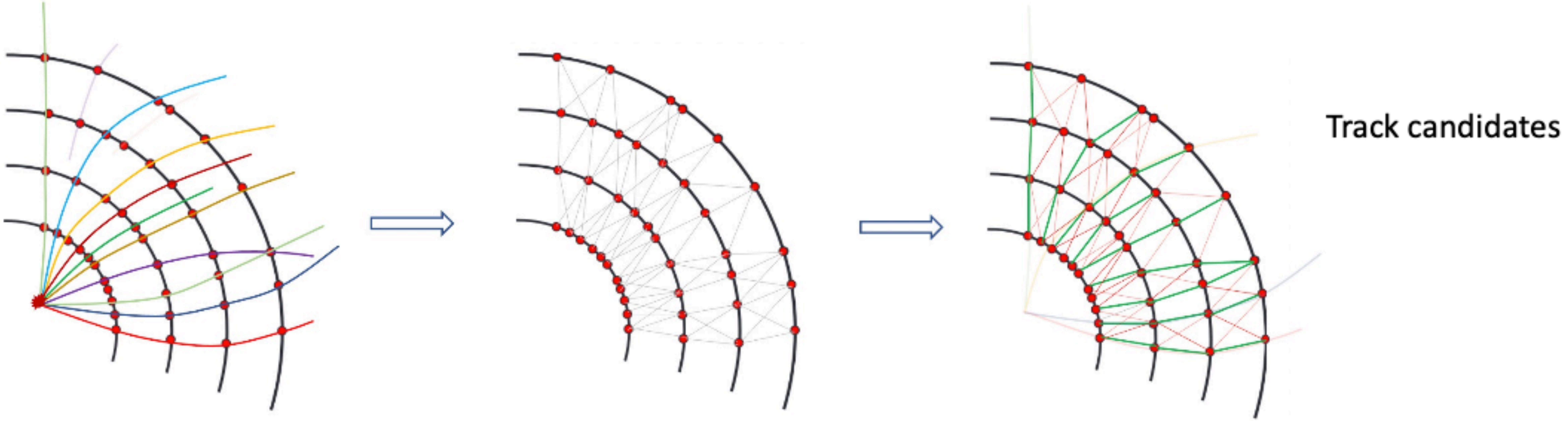
[S. Amrouche, T. Golling, M. Kiehn, AS: Music, Neighbours & Tracking]

[S. Amrouche, N. Calace, T. Golling, M. Kiehm. AS : Hashing & similarity learning]

Labelling: Graph Networks

Connecting a **point to cloud** to **paths:**

- perfect fit for a Graph (Neural) Network architecture



[Exa.TrkX]

Labelling: Graph Networks

NEURAL NETWORKS AND CELLULAR AUTOMATA IN EXPERIMENTAL HIGH ENERGY PHYSICS

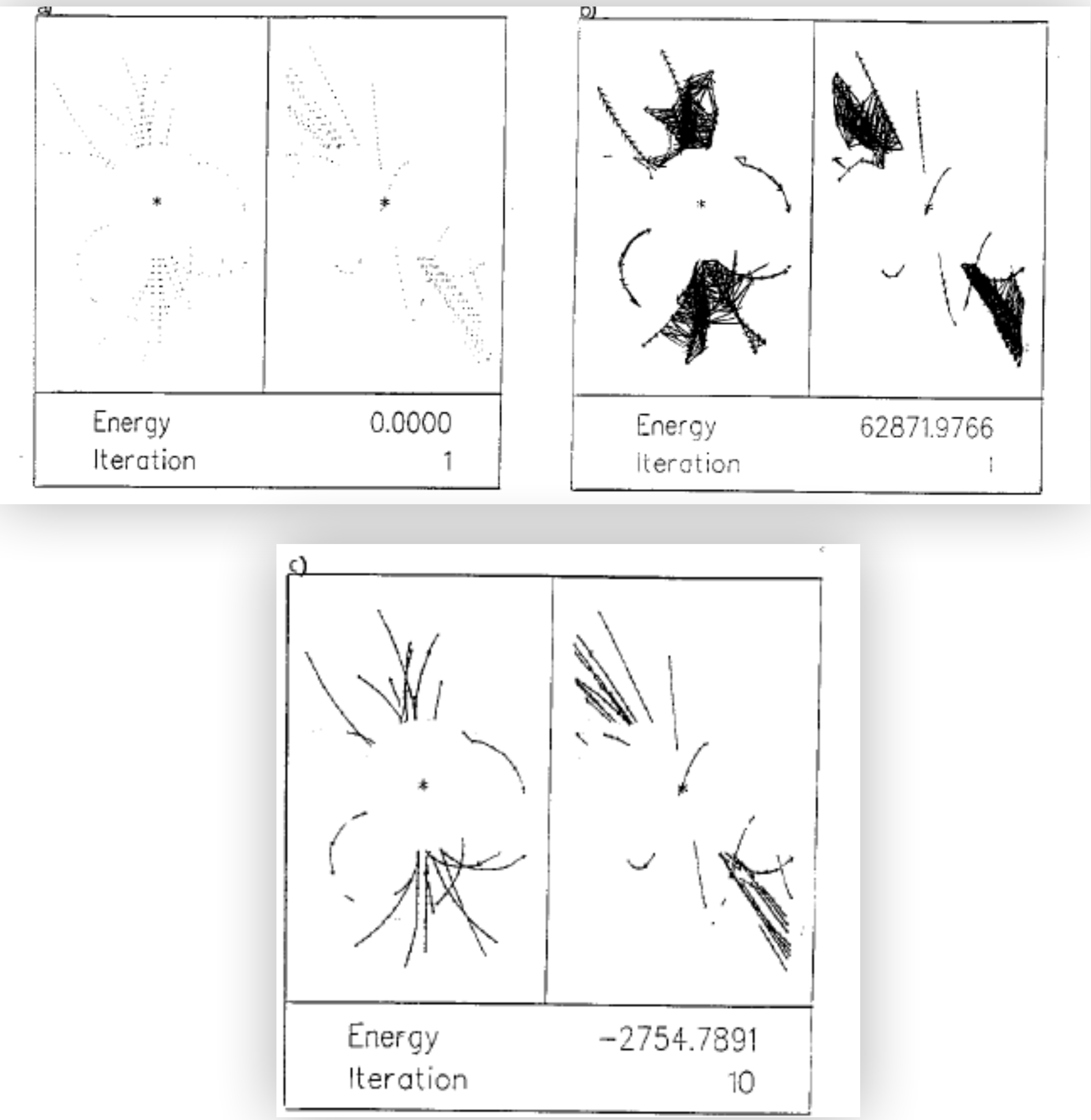
B. DENBY

Laboratoire de l'Accélérateur Linéaire, Orsay, France

Received 20 September 1987; in revised form 28 December 1987

Within the past few years, two novel computing techniques, cellular automata and neural networks, have shown considerable promise in the solution of problems of a very high degree of complexity, such as turbulent fluid flow, image processing, and pattern recognition. Many of the problems faced in experimental high energy physics are also of this nature. Track reconstruction in wire chambers and cluster finding in cellular calorimeters, for instance, involve pattern recognition and high combinatorial complexity since many combinations of hits or cells must be considered in order to arrive at the final tracks or clusters. Here we examine in what way connective network methods can be applied to some of the problems of experimental high energy physics. It is found that such problems as track and cluster finding adapt naturally to these approaches. When large scale hard-wired connective networks become available, it will be possible to realize solutions to such problems in a fraction of the time required by traditional methods. For certain types of problems, faster solutions are already possible using model networks implemented on vector or other massively parallel machines. It should also be possible, using existing technology, to build simplified networks that will allow detailed reconstructed event information to be used in fast trigger decisions.

Computer Physics Communications 49 (1988) 429–448
North-Holland, Amsterdam

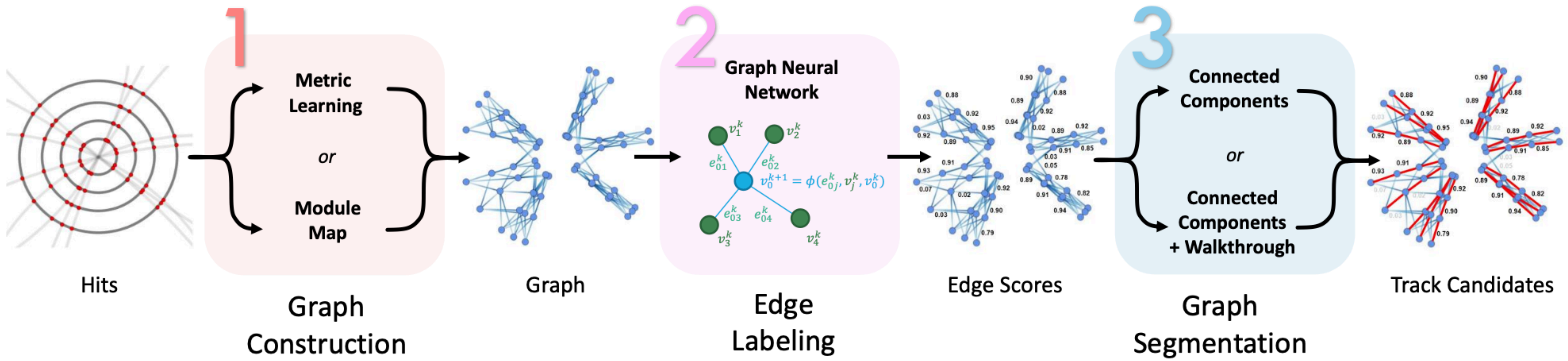


Long before Exa.TrkX AND L2IT

Labelling: Graph Networks

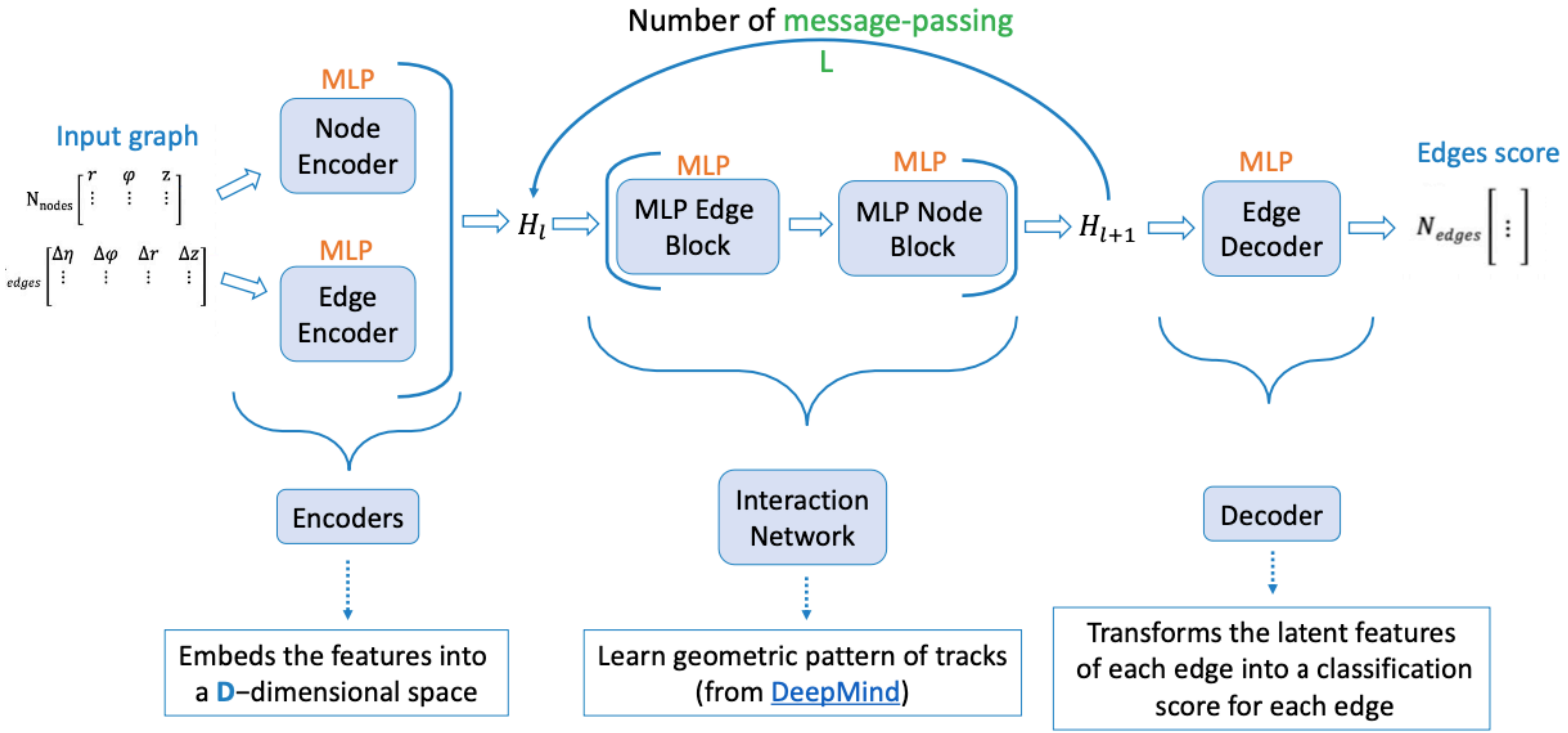
Connecting a **point to cloud** to **paths:**

- perfect fit for a Graph (Neural) Network architecture



Exa.TrkX AND L2IT

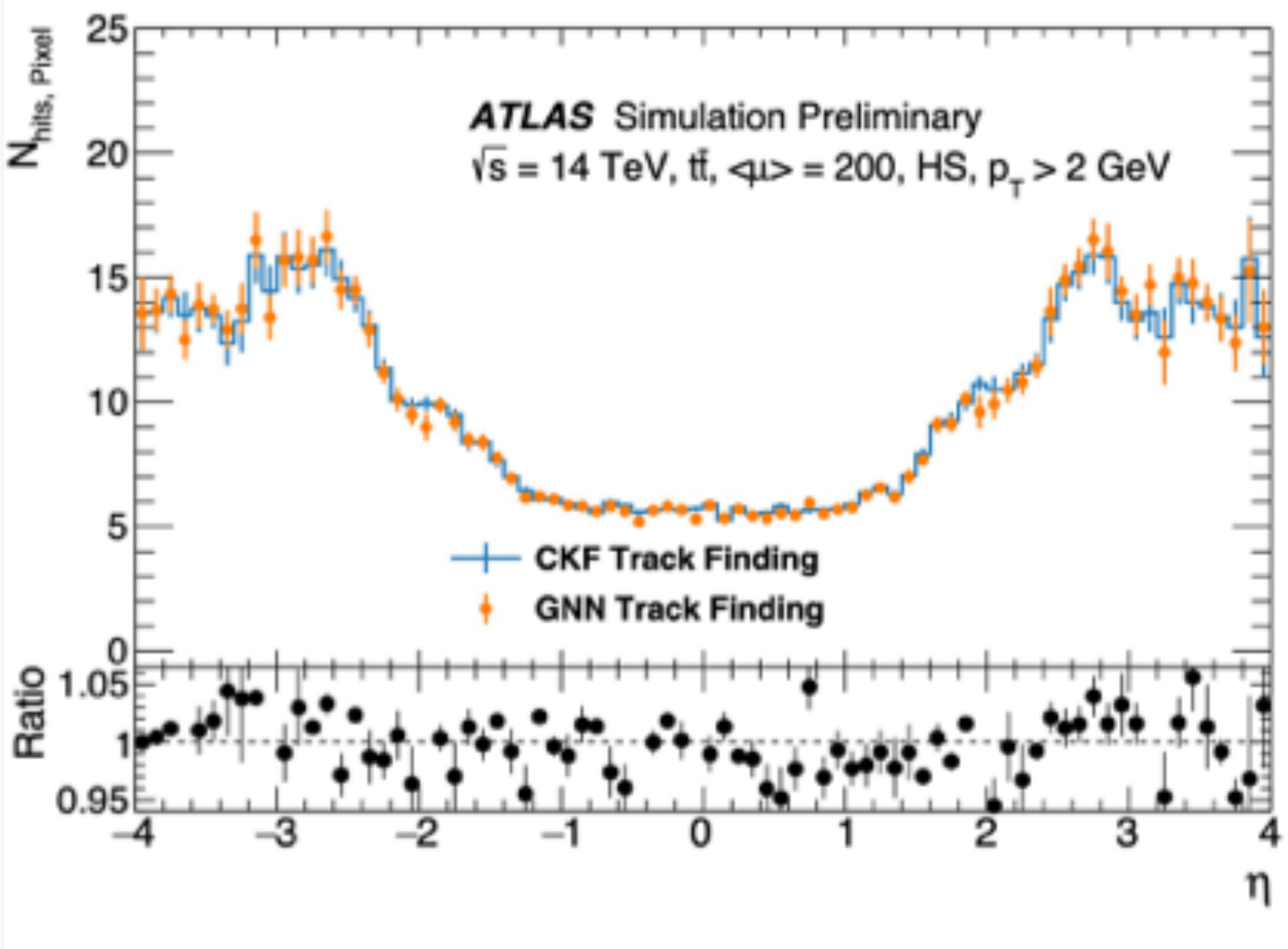
Labelling: Graph Networks

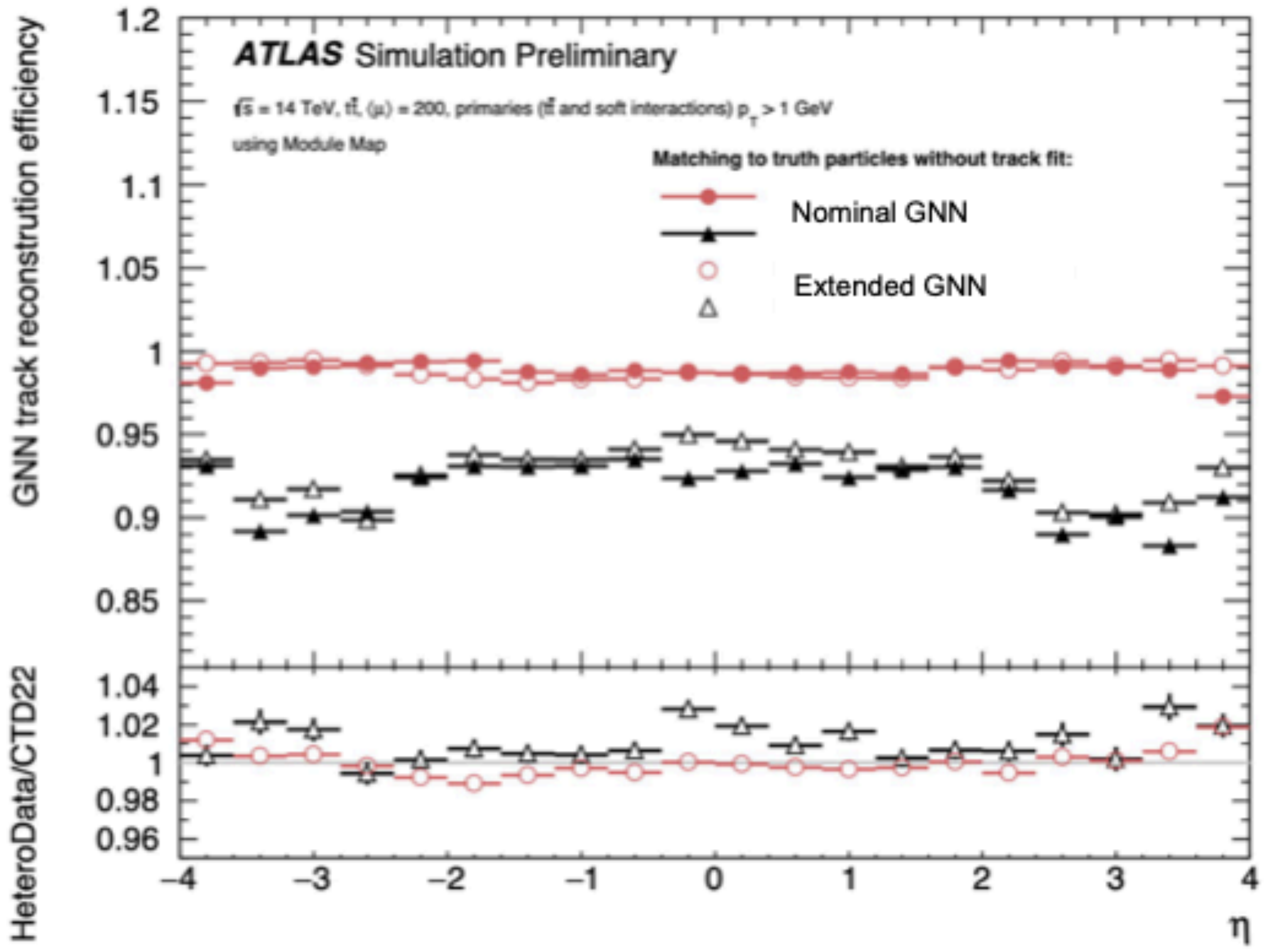


Labelling: Graph Networks

Performance is approaching ATLAS ITk track reconstruction for bulk pile-up tracks

- electrons, dense environments, etc. to be checked next



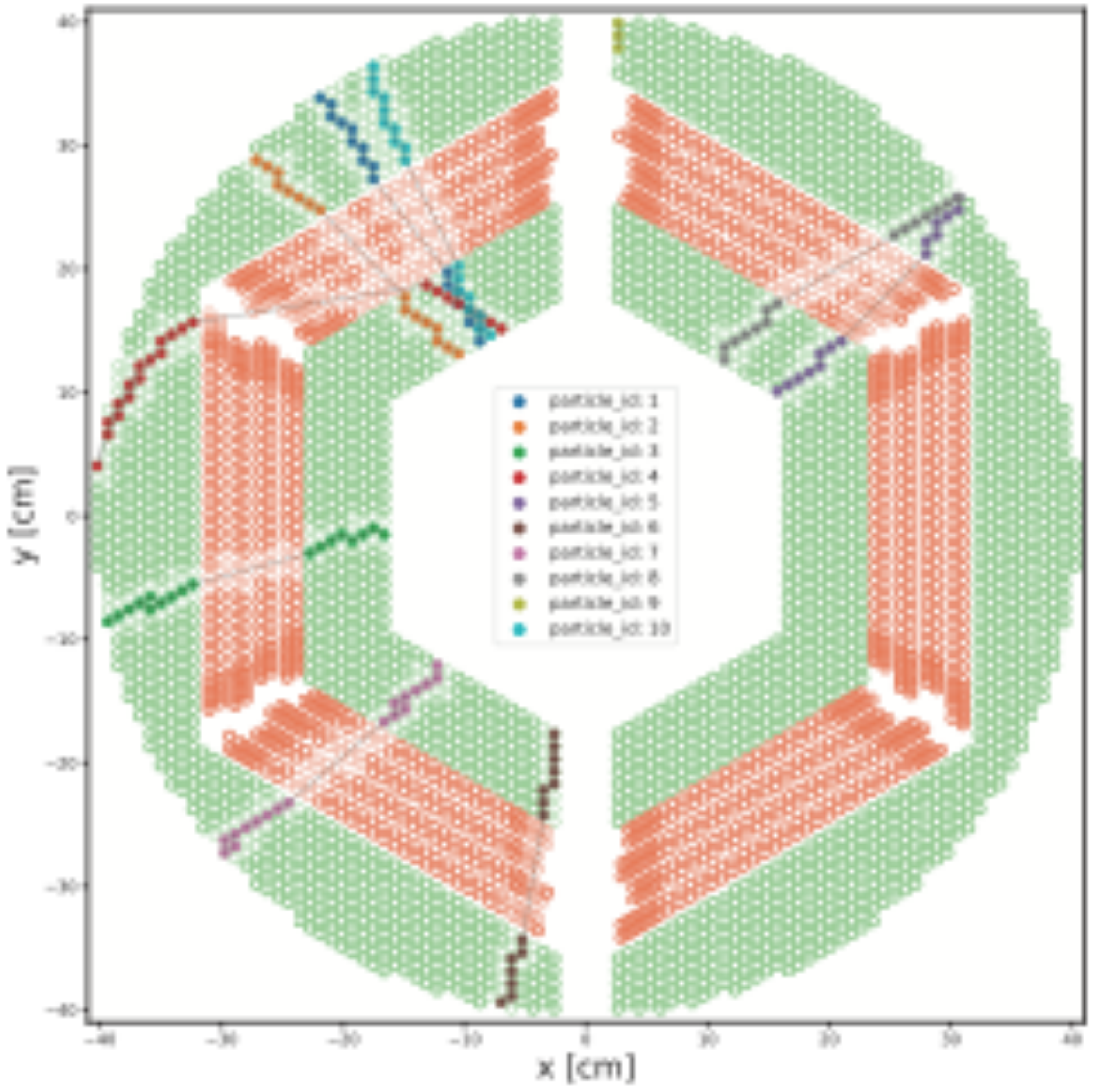


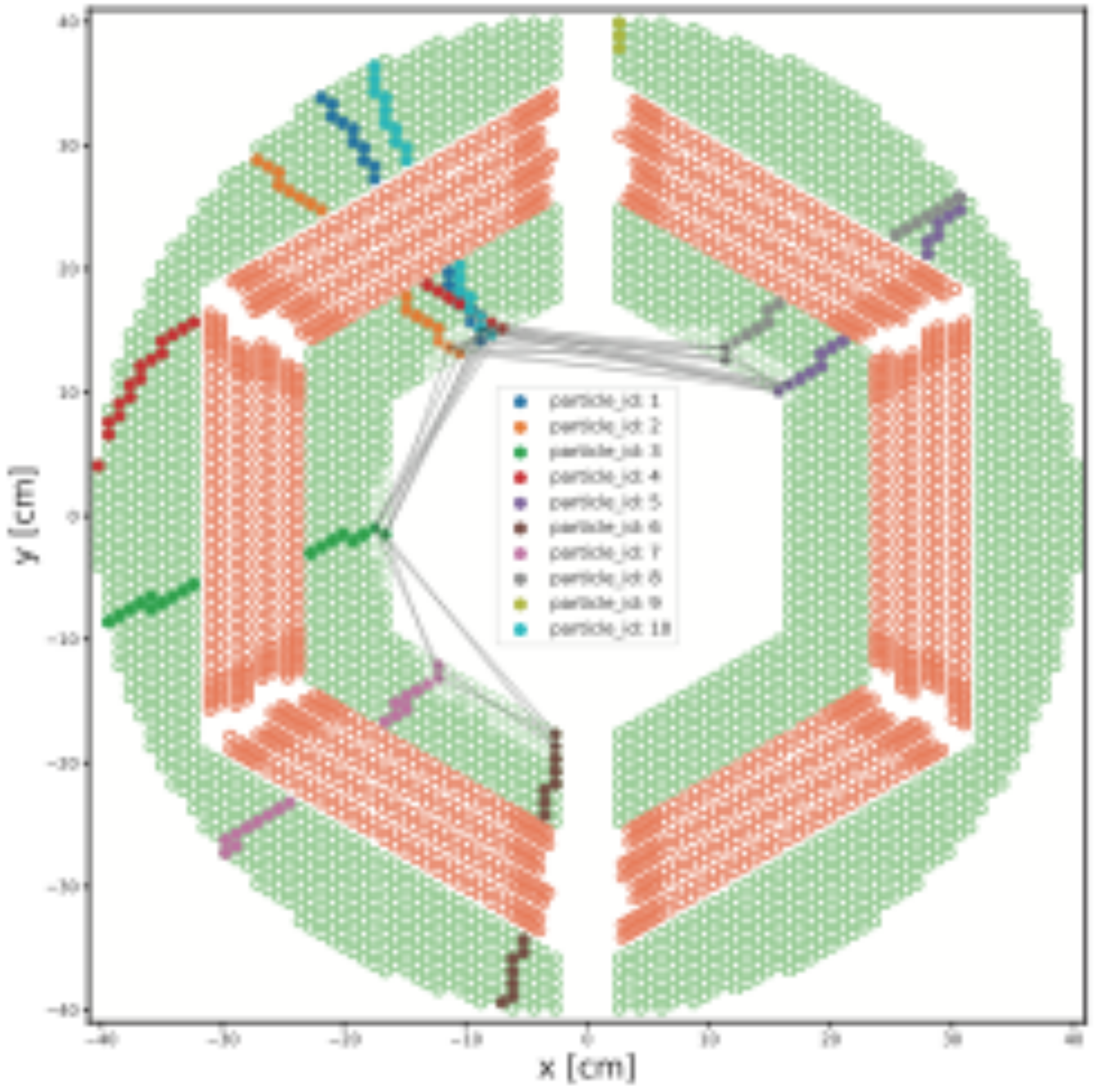
[X. Ju, CHEP2023]

Transferability: Graph Networks

Exa.TrkX pipeline recently also tried for PANDA experiment

- showed acceptable performance on an entirely different detector

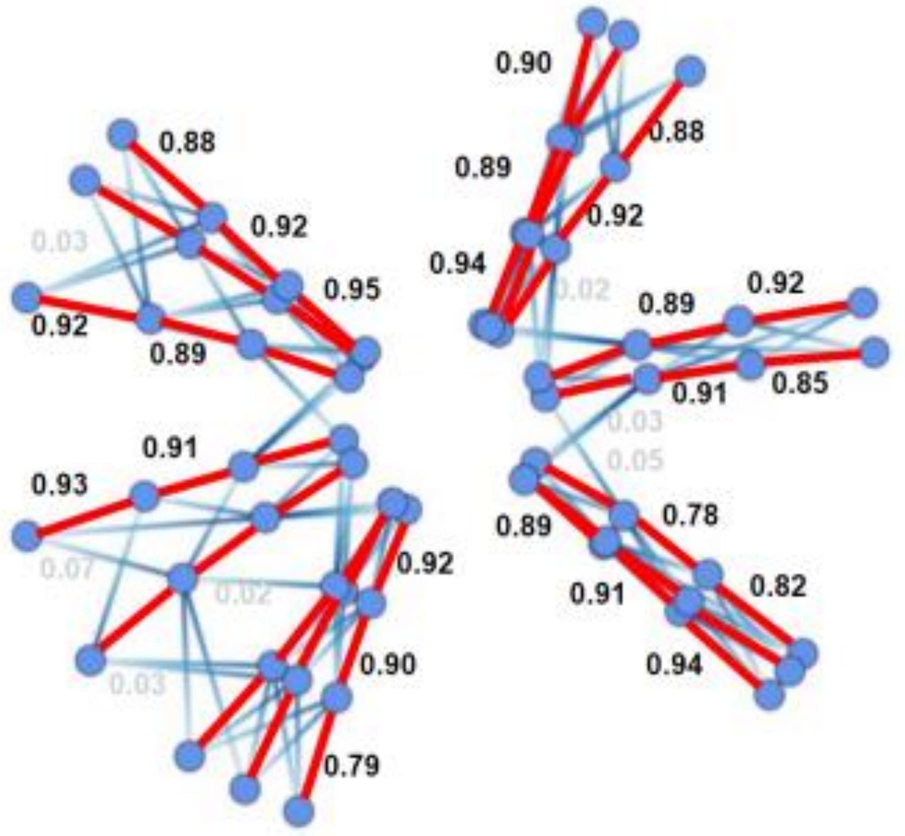




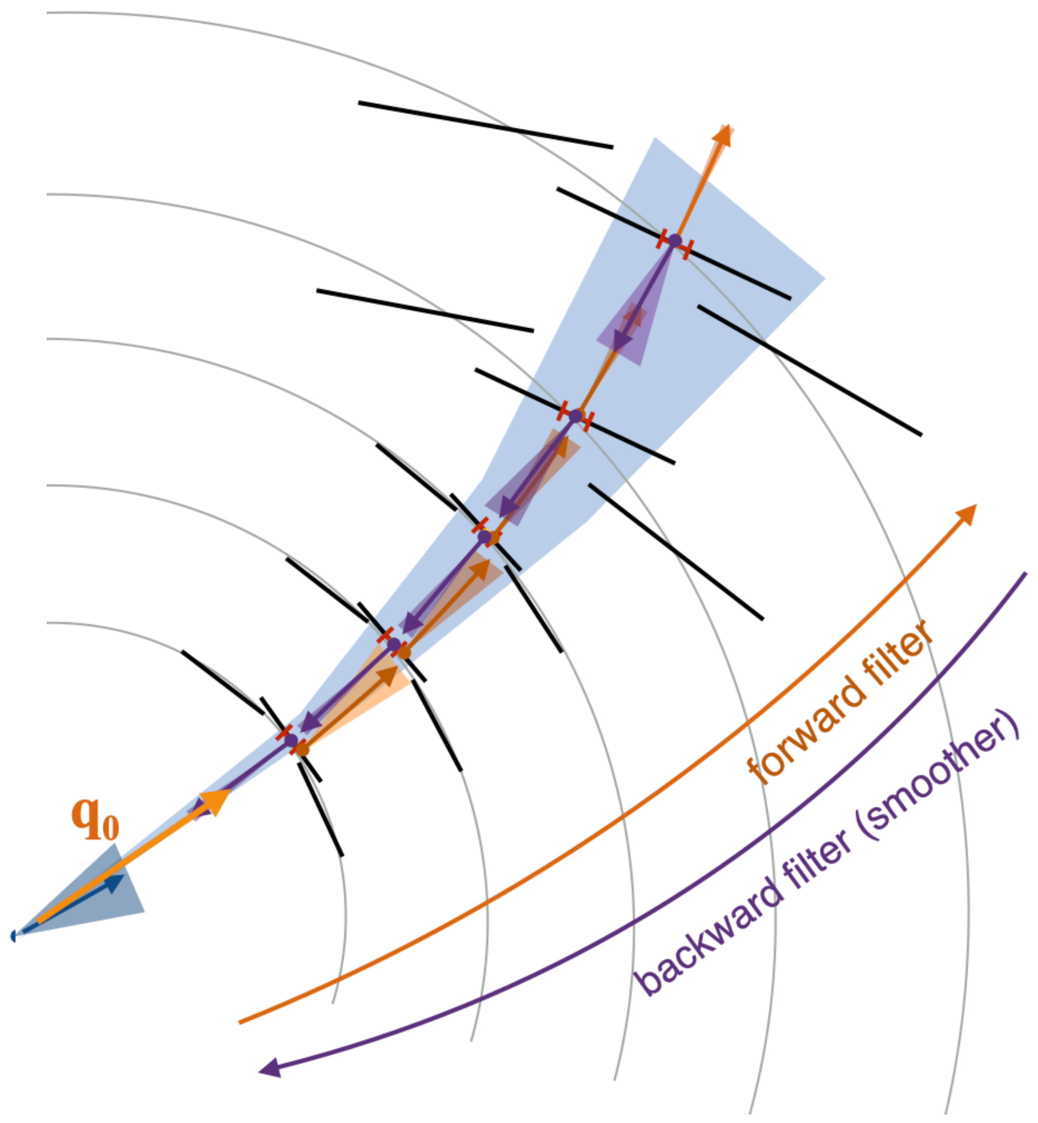
But is this the right tool for such a setup?

Labelling & Inference

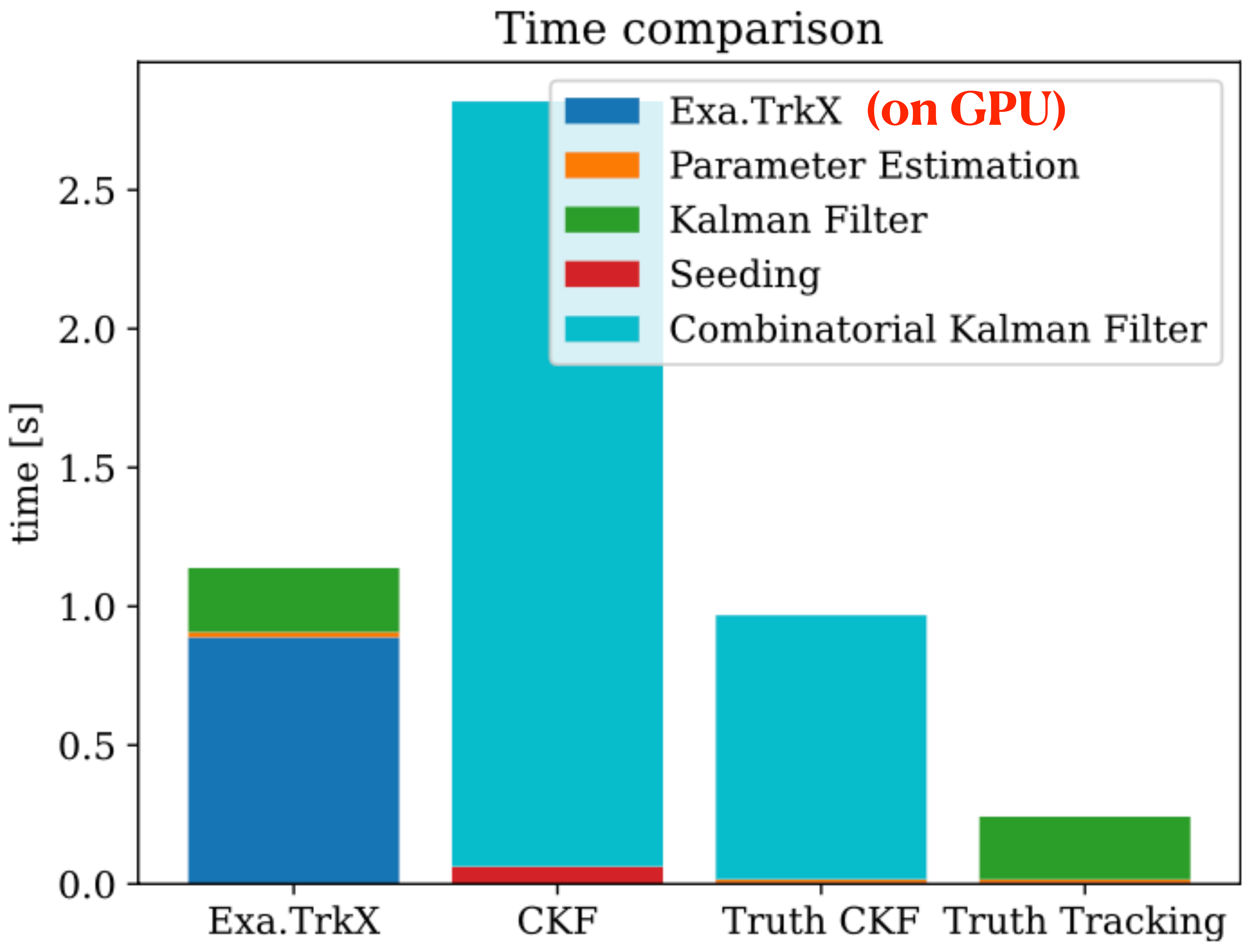
Deployed GNN is a track labelling module, parameters of the particle not yet estimated



GNN track candidate output →



Classical estimator (Kalman Filter) →



achieving **comparable computing performance**
(other algorithms are single-core CPU)

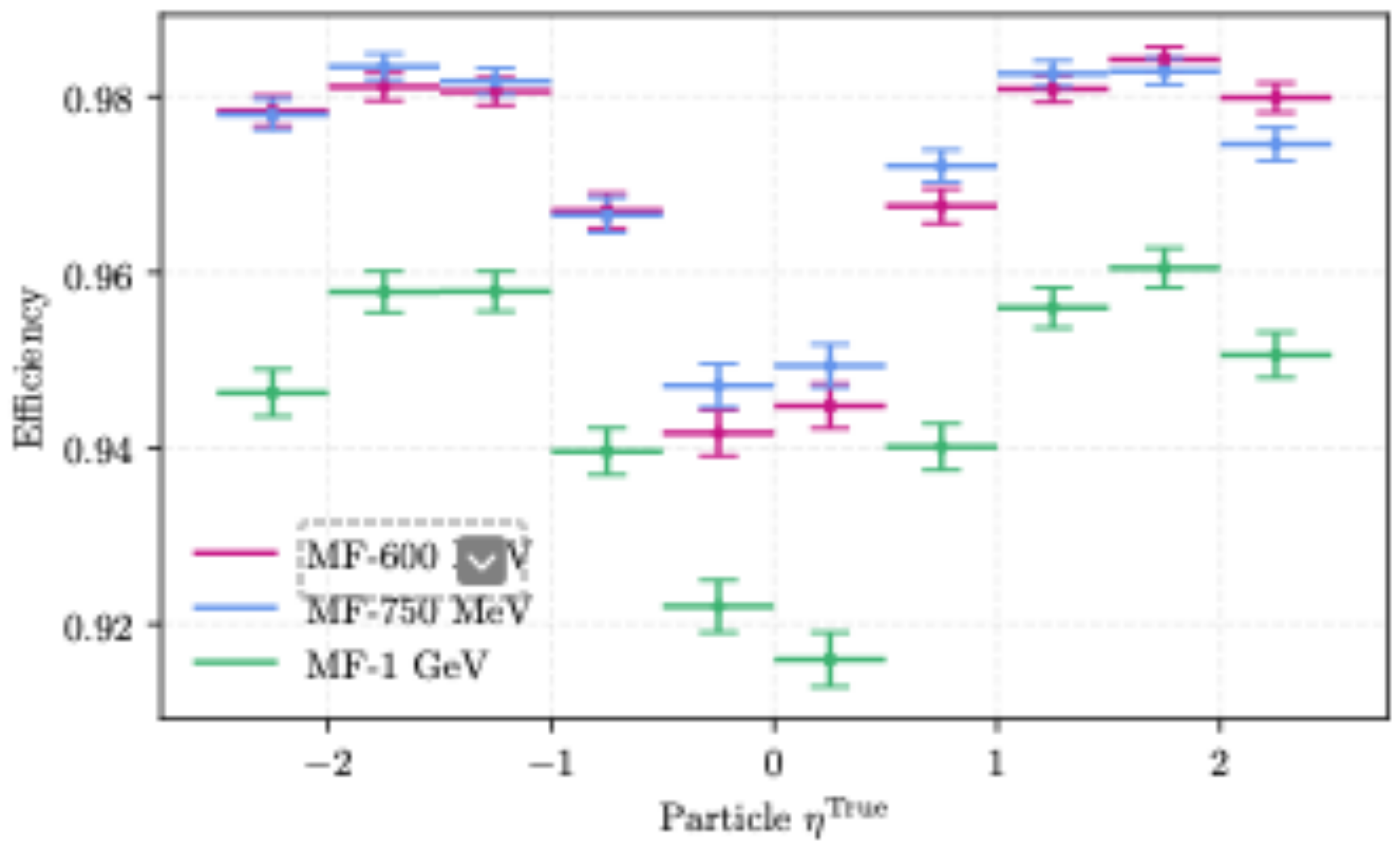
[Huth, ..., Wettig, AS : Applying the Exa.TrkX piple to the Open Data Detector]

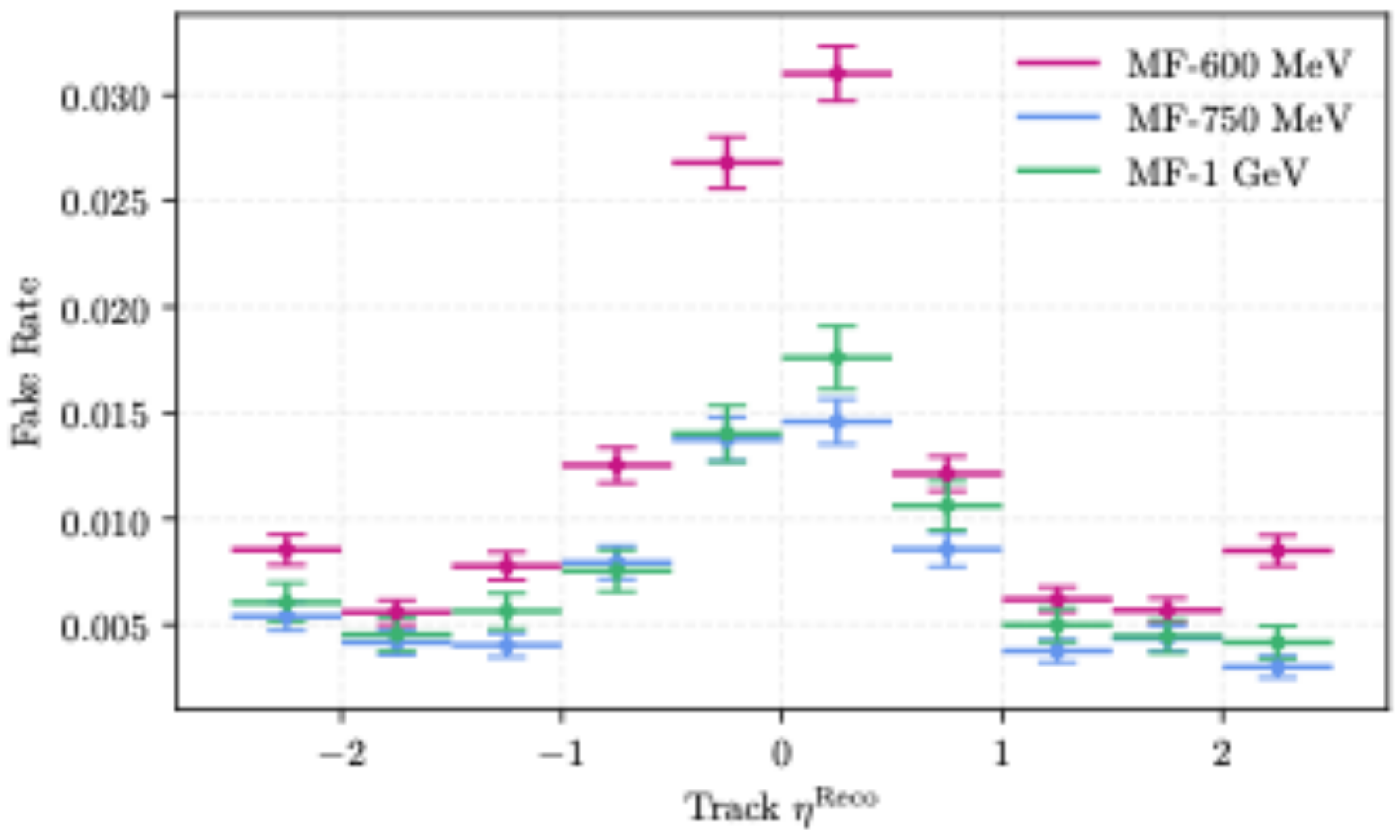
Some modern approaches

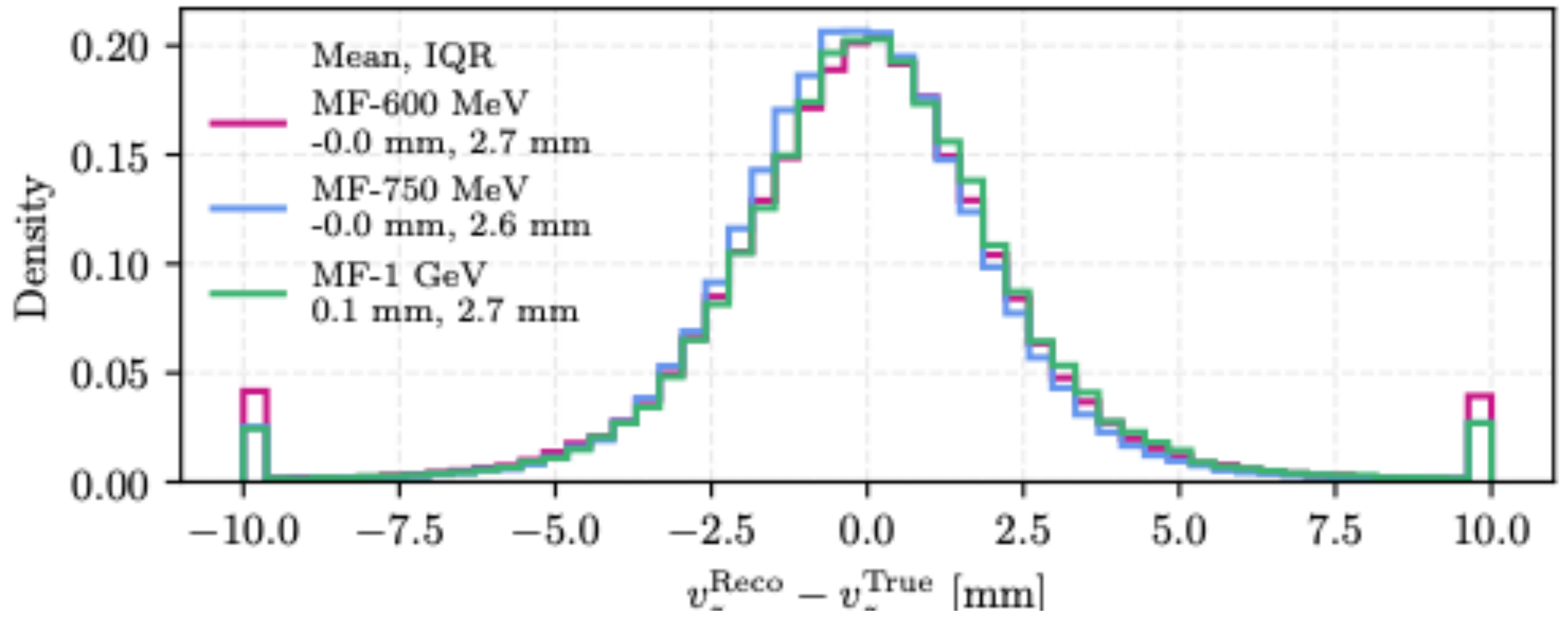
Recent years saw a series of modern architectures being used for track reconstruction:

e.g. using **transformer network** (in this study with attention from nearby hits only)

revolutionised LLMs







+ regression

good tracking efficiency (Tracking ML challenge dataset)

[Stroud, Duckett, Hart, Pond, Rettie, Facini, et Scanlon, arXiv:411.07149]

More on inference/regression

Kalman Filter is a linear dynamical system

- can be modelled as a NN: Deep Kalman Filter [https://arxiv.org/abs/1511.05121]

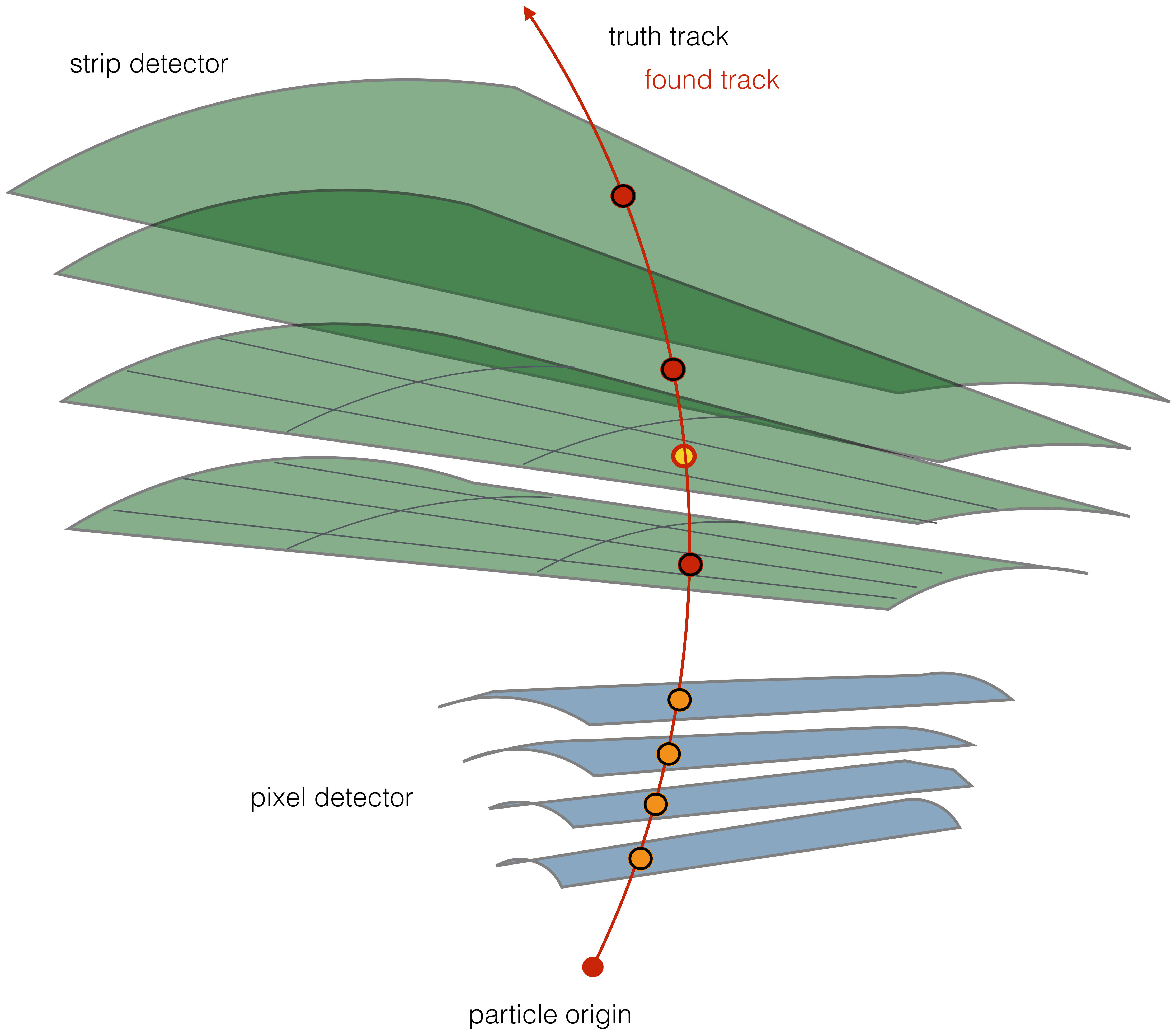
Combinatorial Kalman Filter can be modelled as a RNN

- initial attempts done by Hep.TrkX [talk]

Possibilities for dedicated non-gaussian error fitters, e.g. GSF

- electron inference is a tricky business …

Track classification

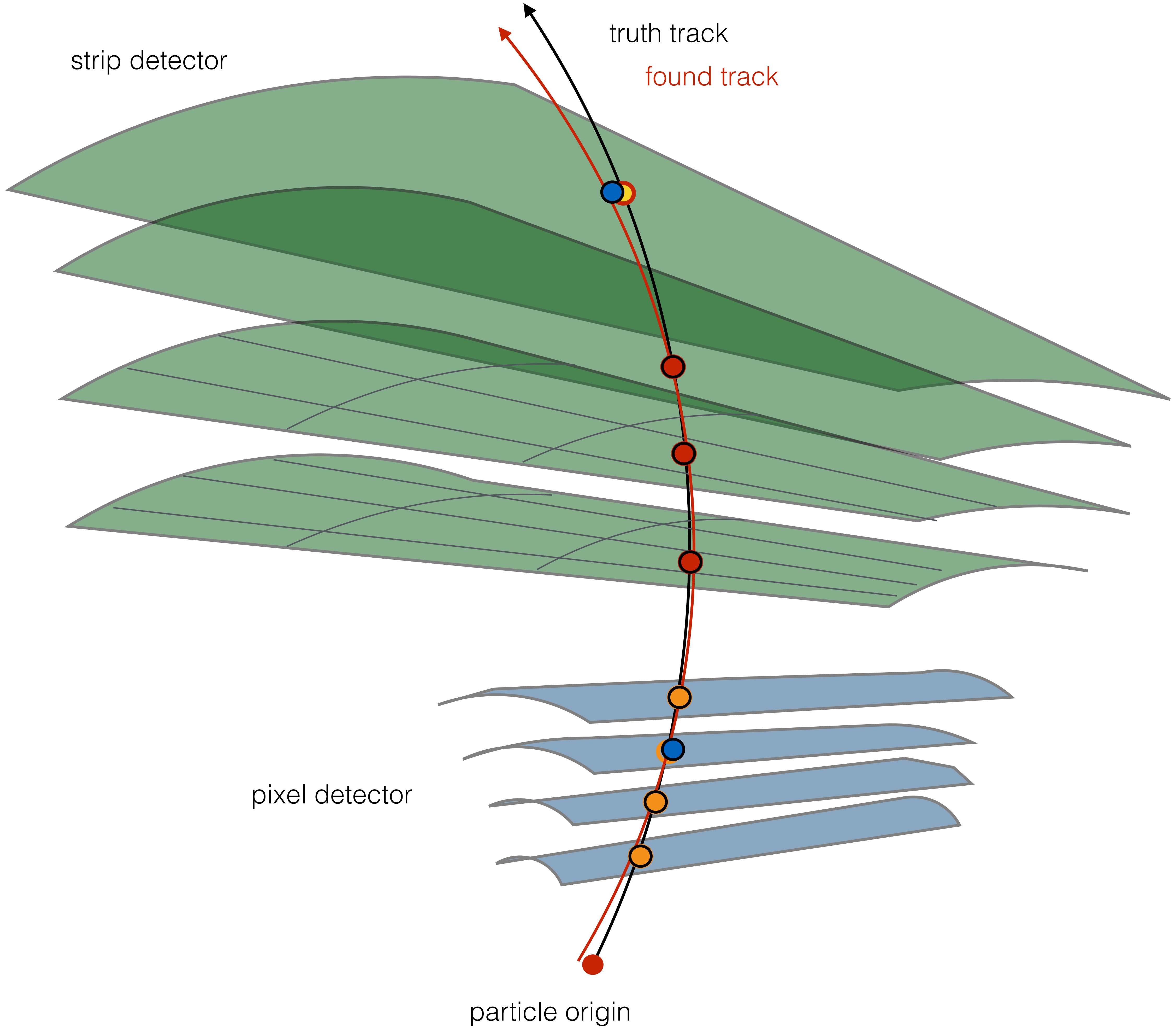


4 pixel hits, 4 strip hits created

4 pixel hits, 3 strip hits found and assigned

that’s an ok track,
you got 7 out of 8,
naive score = 7/8 = 0.875

Track classification



4 pixel hits, 4 strip hits created

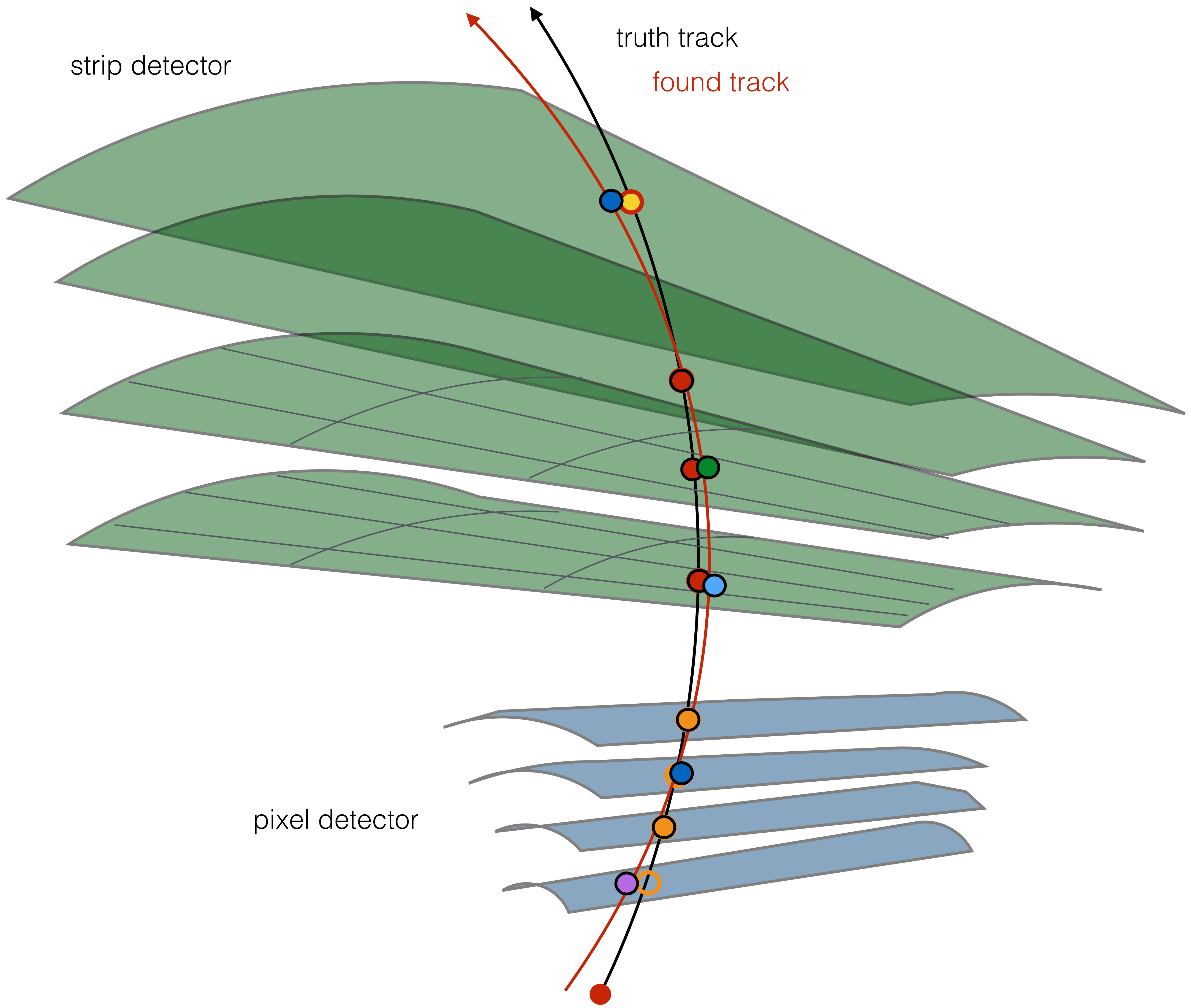
4 pixel hits, **4** strip hits found
2 wrongly associated

that's not very good
you got 6 out of 8,
<u>naive</u> score = 6/8 = 0.75

your track is rather distorted

did you really measure the particle ?

Track classification



4 pixel hits, 4 strip hits created

4 pixel hits, **4** strip hits found
randomly associated (3 associated)

that's garbage
you got 3 out of 8,
naive score = 3/8 = 0.375

your track is **a ghost**

that should not even give you a score !
in fact, it should count as score = -1

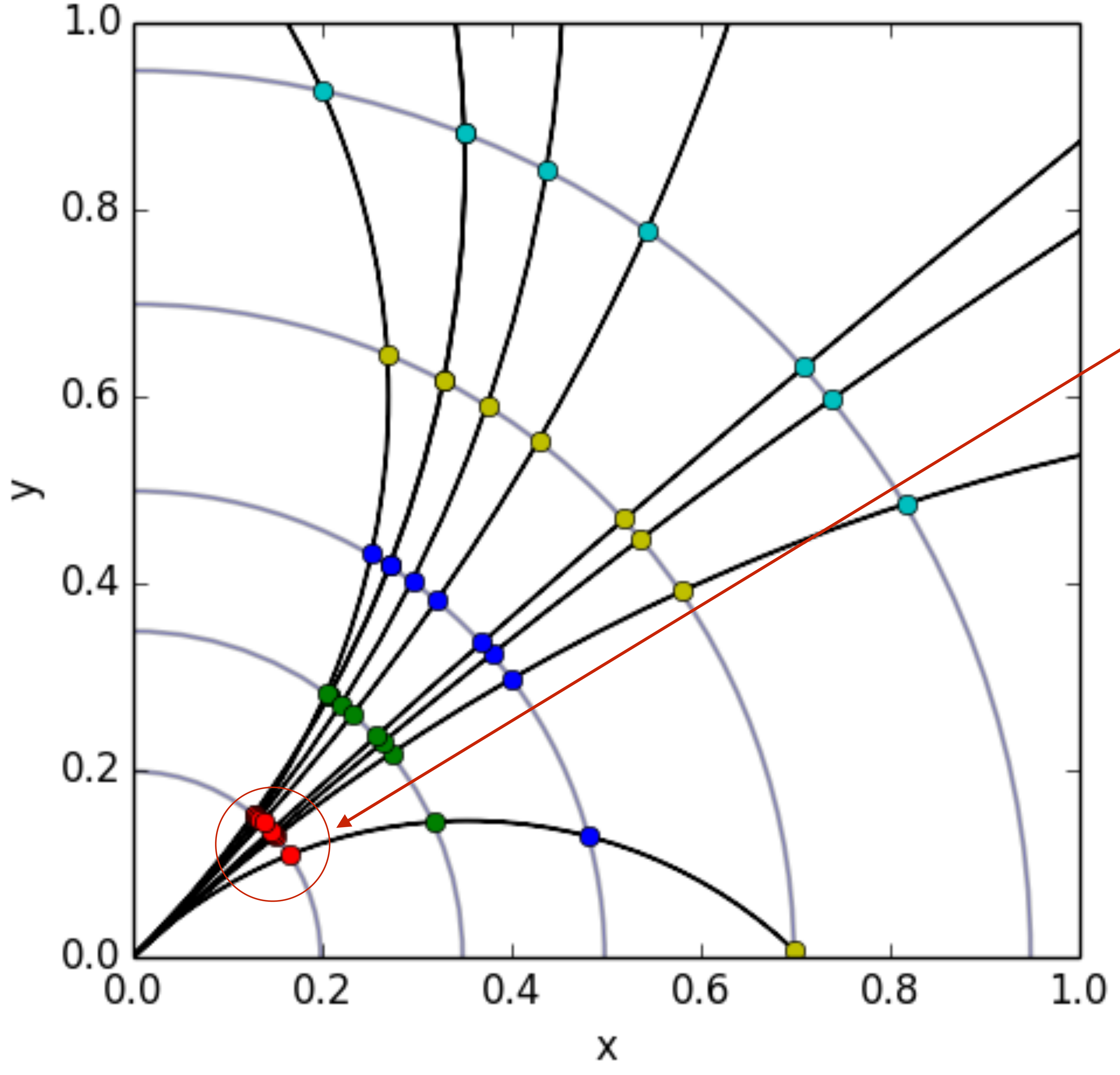
ML Track classification

Ambiguity solving with a trained Neural Network

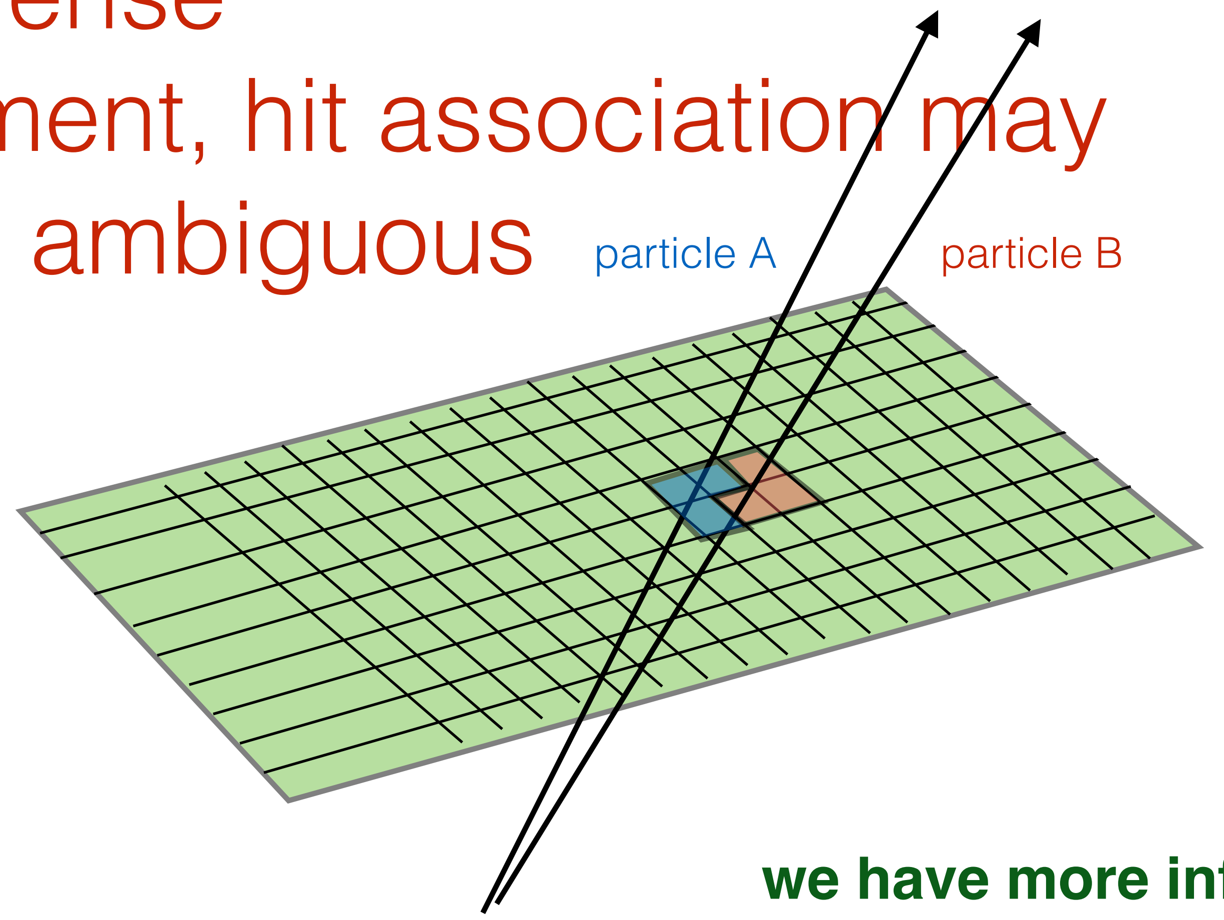
- case study on Open Data Detector [C. Allaire, CHEP2023, Parallel Tallk]
- earlier attempts, e.g. BDT with CMS tracking were similarly successful

	Number of tracks	Number of truth particles	Efficiency (good tracks)	Efficiency (truth tracks)	Duplicate Rate	Fake Rate	Solver speed [ms/event]
CKF	7995	834.7	100 %	100 %	89.5 %	0.06 %	0
CKF + Greedy Solver	823.6	821.4	81.5 %	98.4 %	0.17 %	0.10 %	184
CKF + ML Solver	811.7	810.7	84.2 %	97.1 %	0.05 %	0.06 %	41.2

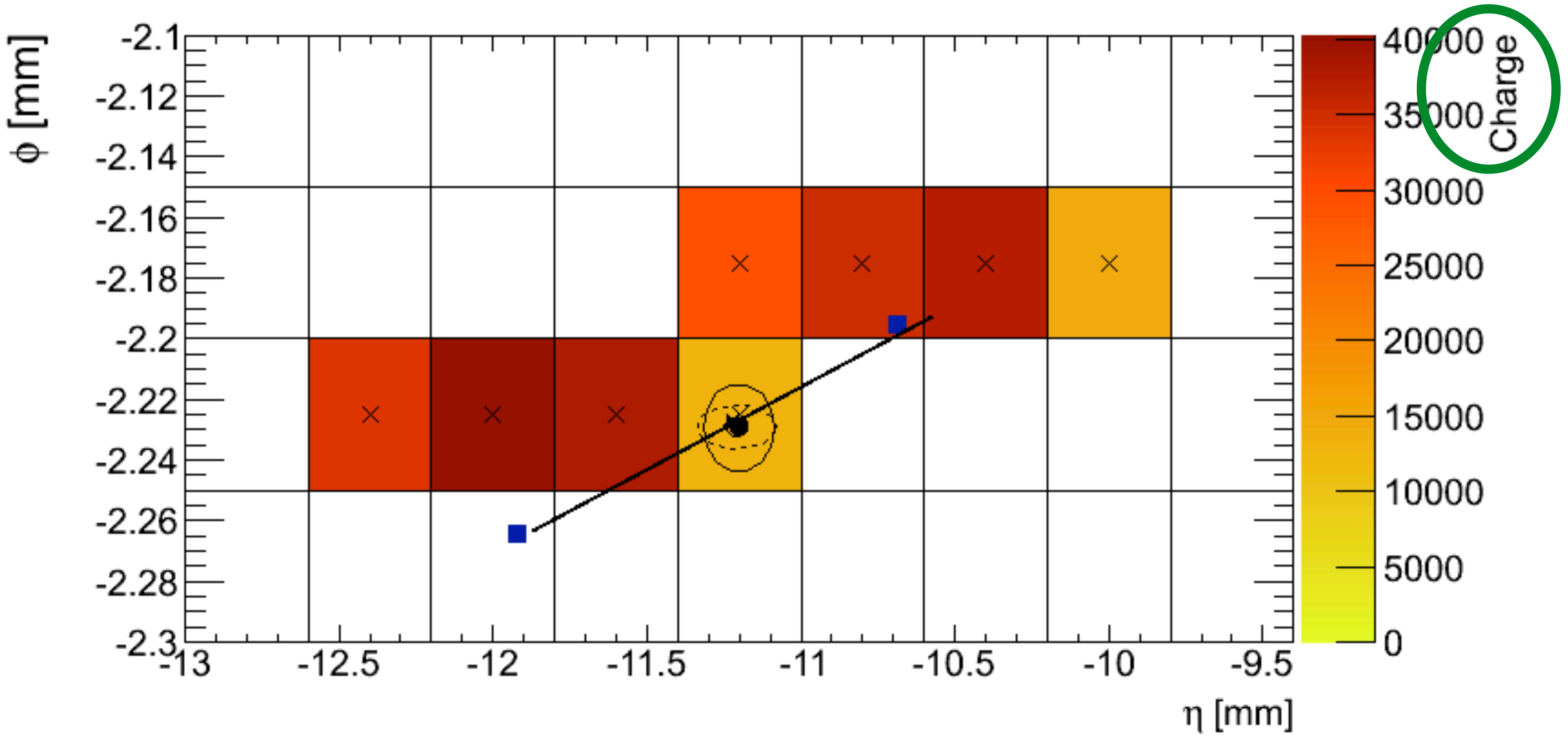
More on ambiguity solving



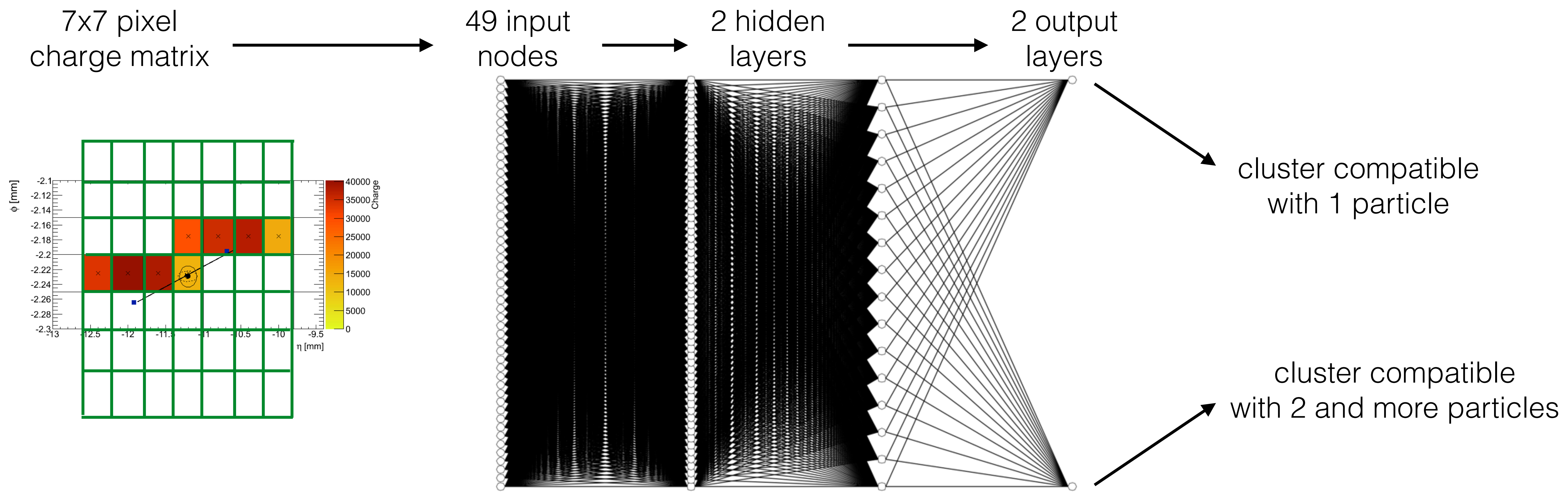
locally dense environment, hit association may become ambiguous

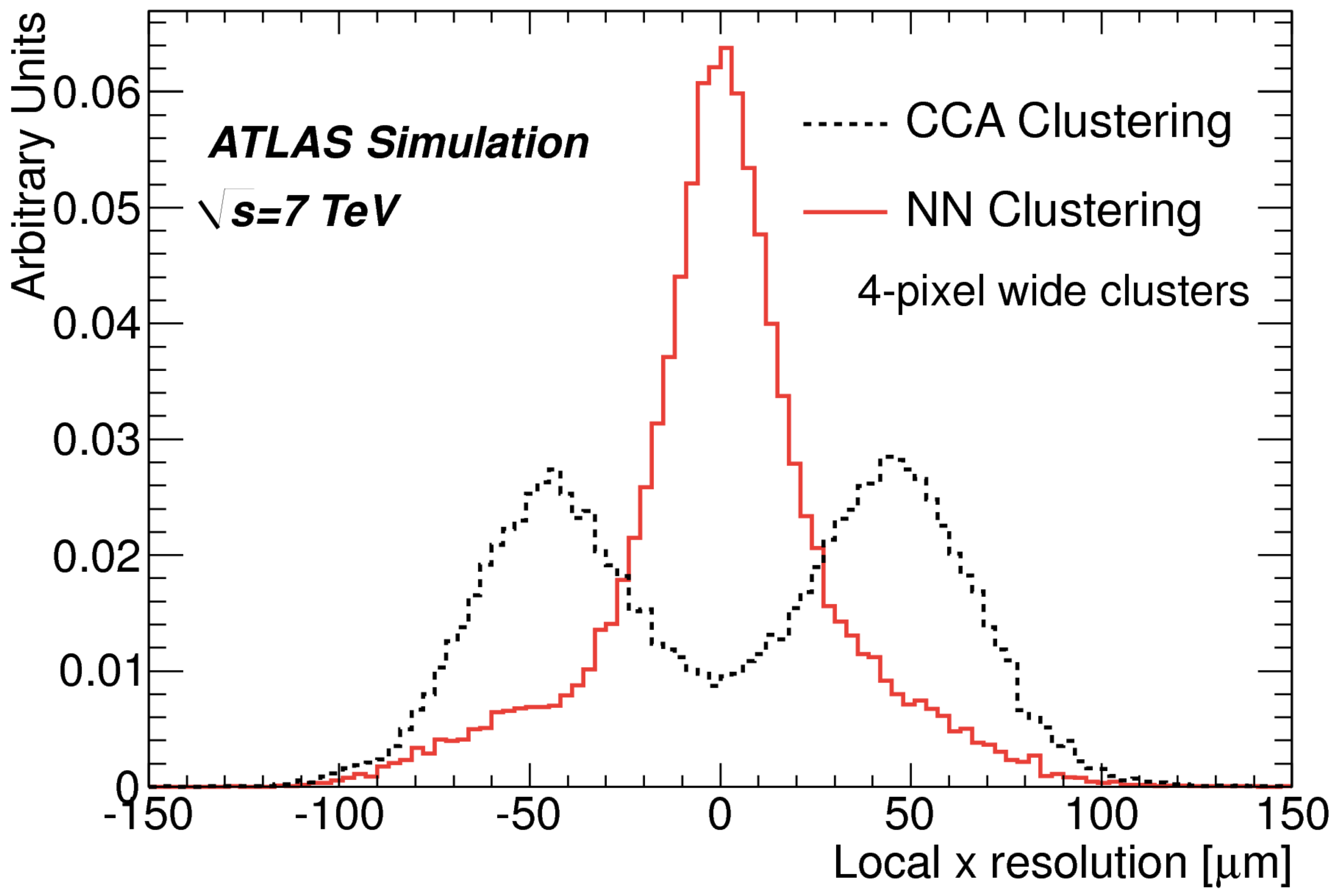


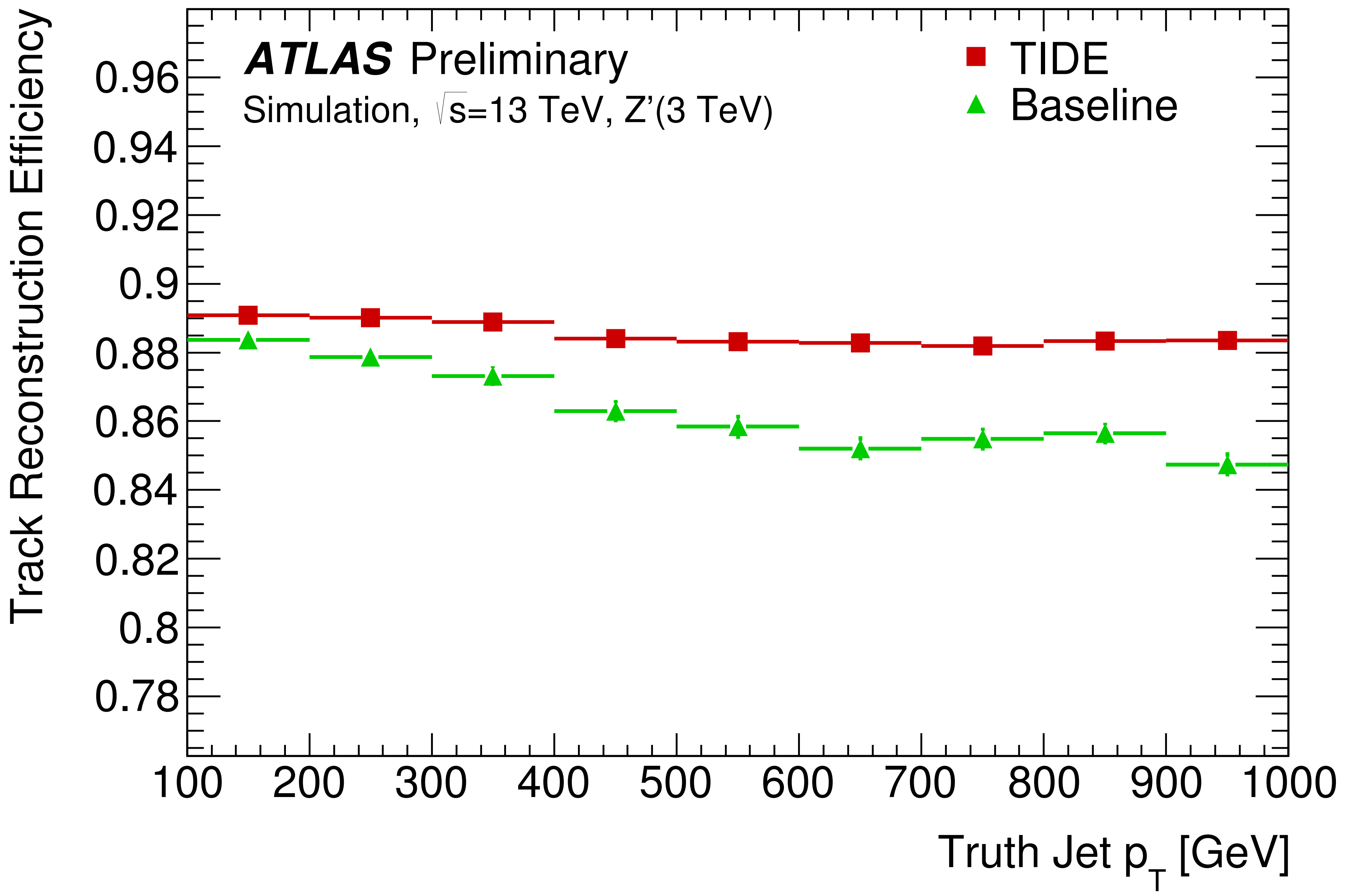
we have more information



Dense environment resolving







[ATL-PHYS-PUB-2015-006]

Tracking ML challenge

TrackML

Simulated **dataset** produced (training, testing, evaluation)

- two challenges organised for DS community

Phase 1

kaggle

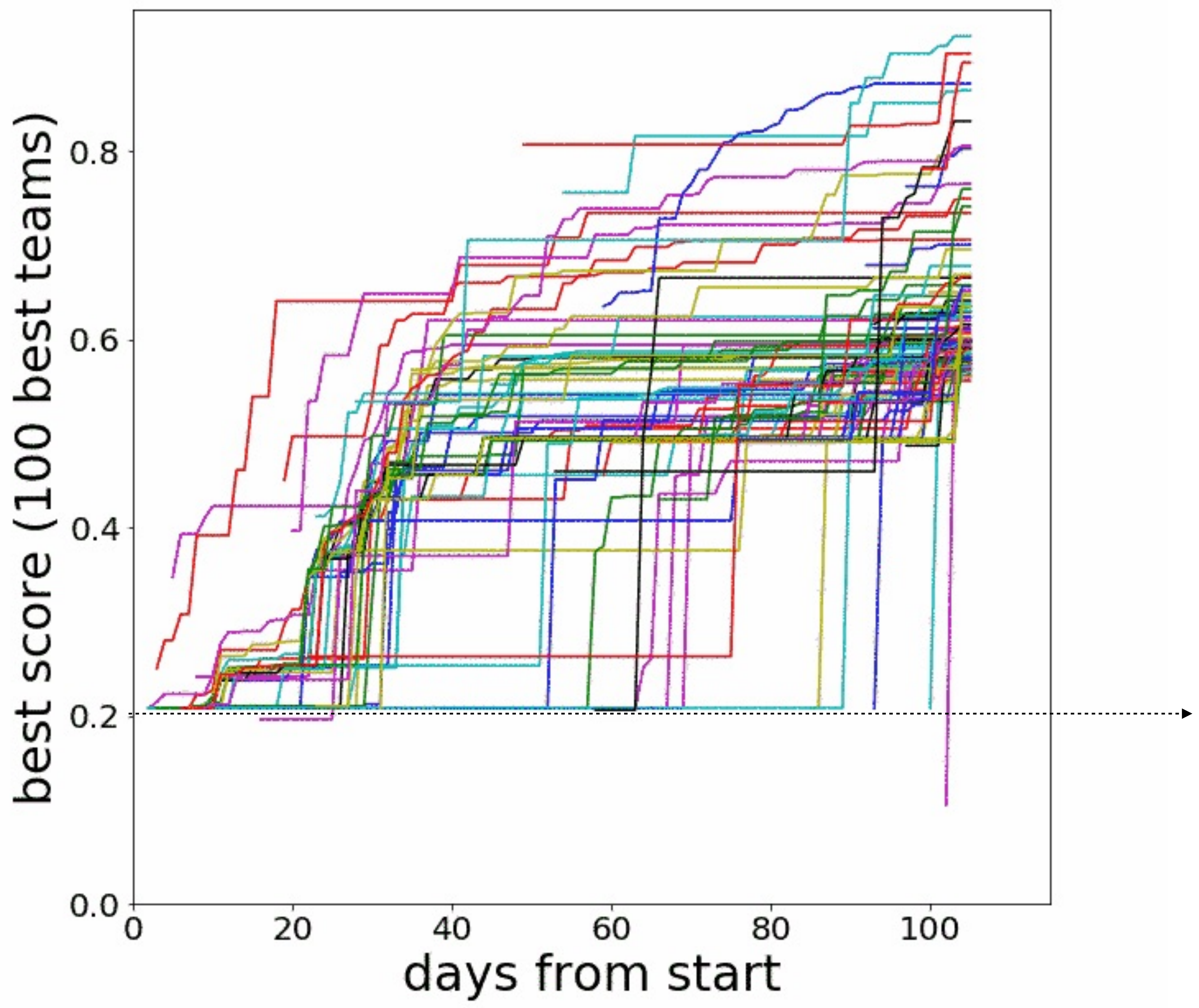
focus on accuracy

656	776	5,837
Teams	Competitors	Entries

Phase 2

CodaLab

accuracy & speed



https://arxiv.org/abs/1904.06778

Some concluding remarks

Machine learning is becoming an **increasingly important component** in track (or event) reconstruction in HEP

- unsupervised learning techniques are bred & butter since ever
- ML assistes modules in classification in production
- first end-to-end solutions for track finding deployed

There can be a **huge benefit in applying ML** to track reconstruction

- yet one shouldn’t just blindly use it
- we have gathered a lot of knowledge which we shouldn’t forget