Machine Learning for Charged Particle Tracking and beyond

Learning to Discover : Advanced Pattern Recognition

Institut Pascal, Orsay

14-25 October 2019

Outline

- The need for charged particle reconstruction
  - Combinatorial Track Finder algorithm
    - Computation challenge at the HL-LHC
      - Alternative approaches
        - Resorting to Machine Learning
          - Challenges for ML
            - Applications and R&D

Extra digressions on HEP.TrkX methods included

HEP/Exa.TrkX Project

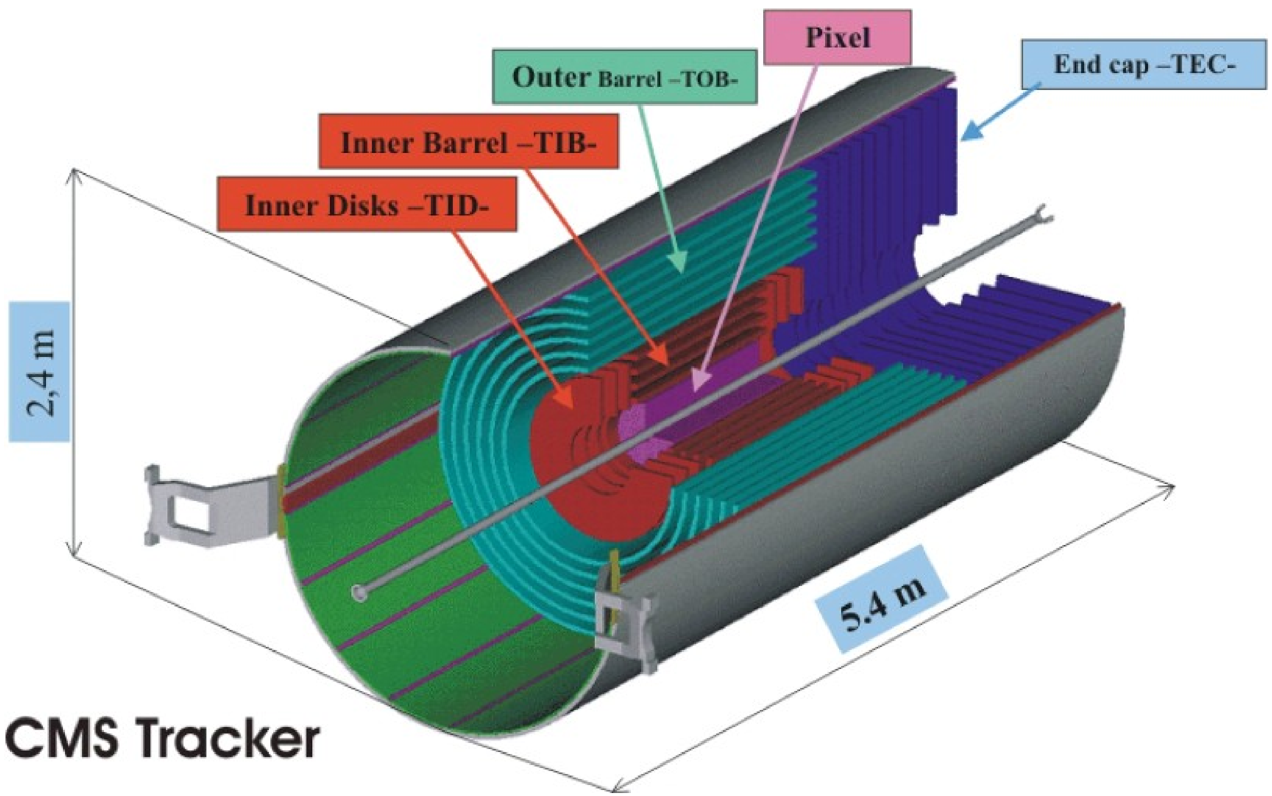
- DOE ASCR, HEP CCE, DOE CompHEP project
- Mission
  - Explore deep learning techniques for track formation
  - Scale up optimization of ML for tracking

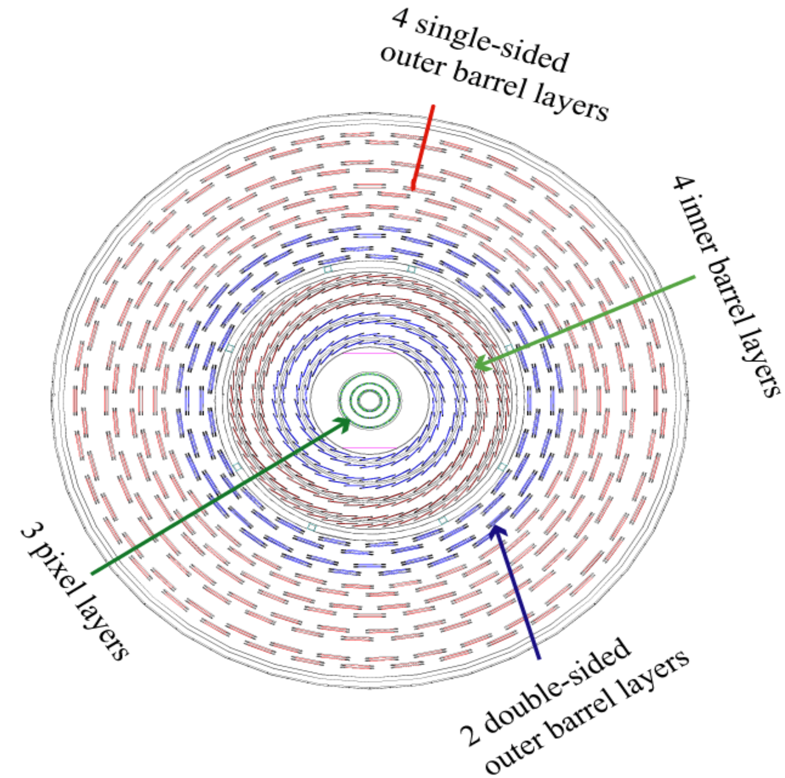
- People
  - **Caltech :** Maria Spiropulu, Jean-Roch Vlimant, Alexander Zlokapa, Joosep Pata
  - **Cincinnati:** Adam Aurisano, Jeremy Hewes
  - **FNAL :** Giuseppe Cerati, Lindsey Gray, Thomas Klijnsma, Jim Kowalkowski, Gabriel Perdue, Panagiotis Spentzouris
  - **LBNL :** Paolo Calafiura, Steven Farrell, Prabhat, Daniel Murnane
  - **ORNL:** Aristeidis Tsaris
  - **SLAC**: Kasuhiro Terao, Tracy Usher

- All material available under
  https://heptrkx.github.io/
  https://exatrkx.github.io/

Tracking Algorithm

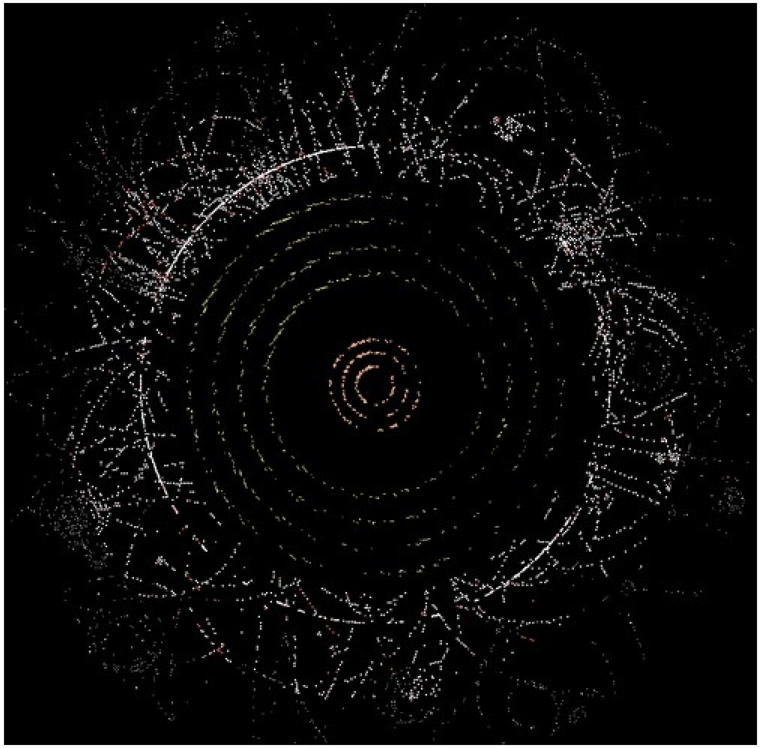
Tracker Detector



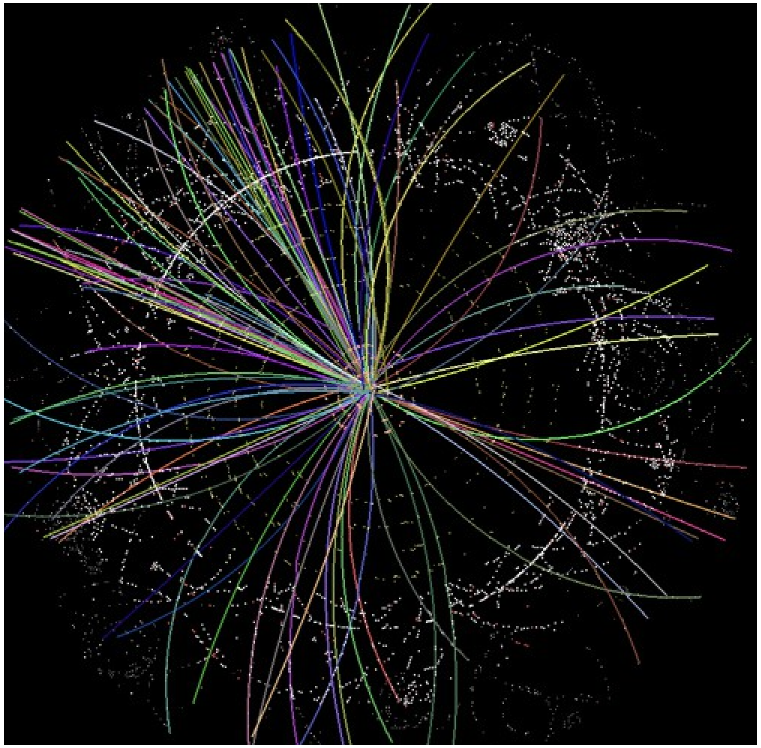


- Particle trajectory bended in magnetic field
- Particle ionize silicon pixel and strip throughout several concentric layers
- Thousands of hits sparsely distributed in space
- Low noise detector, but lots of secondary track hits

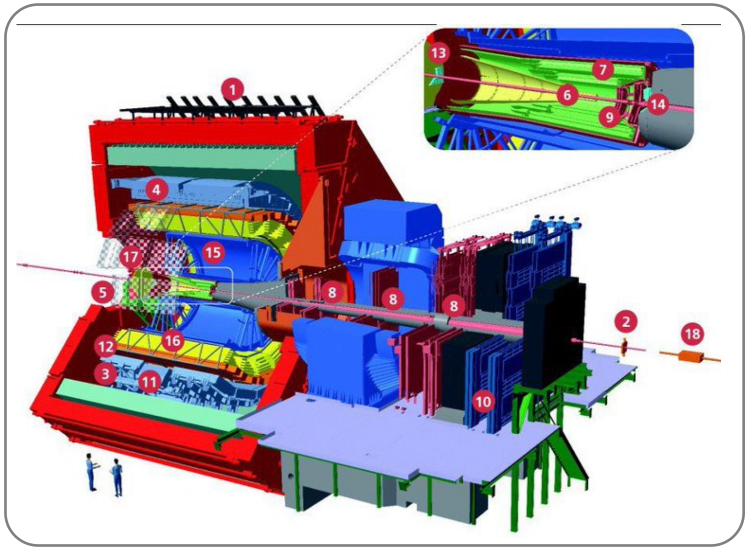
Name of the Game

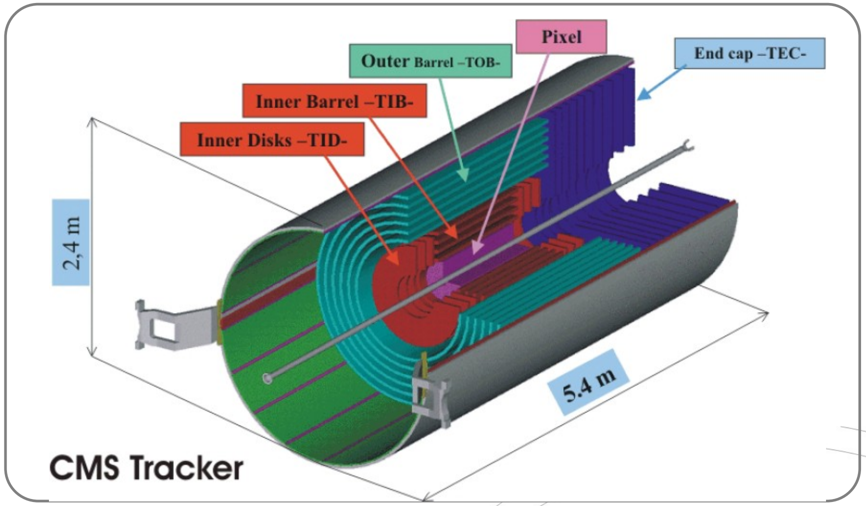


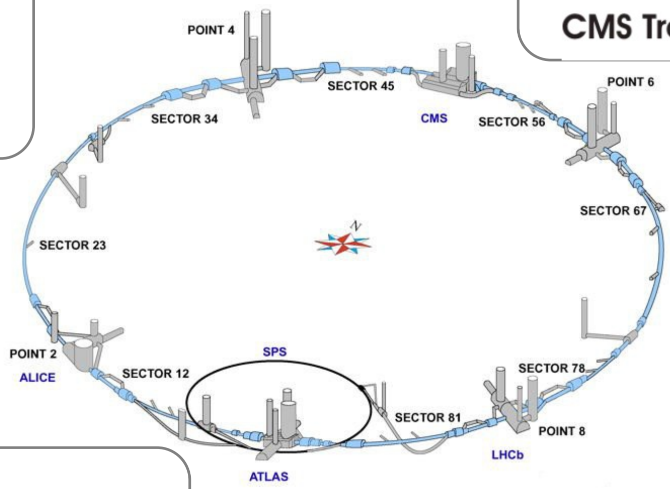
From hits ...

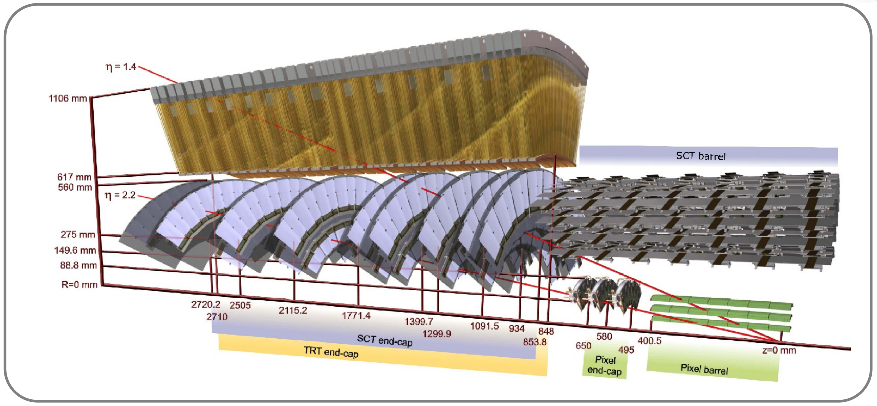


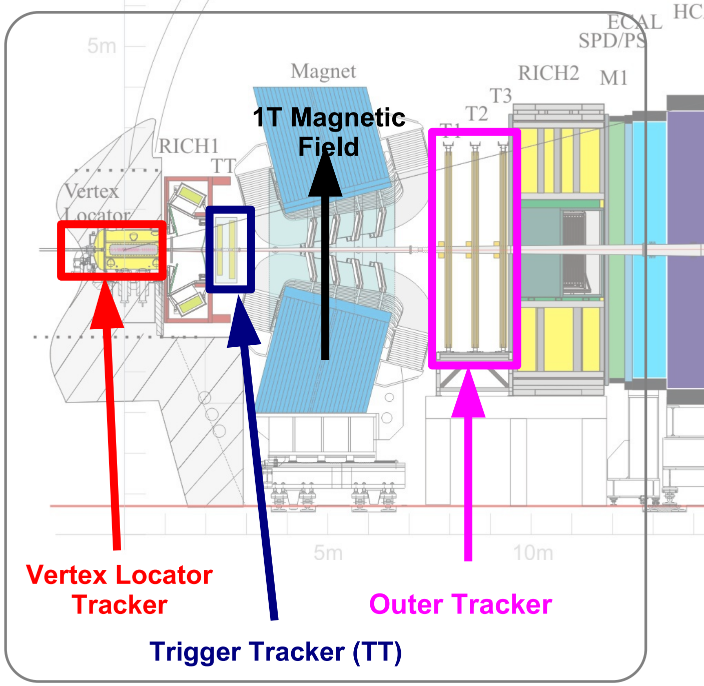
... to trajectory parameters









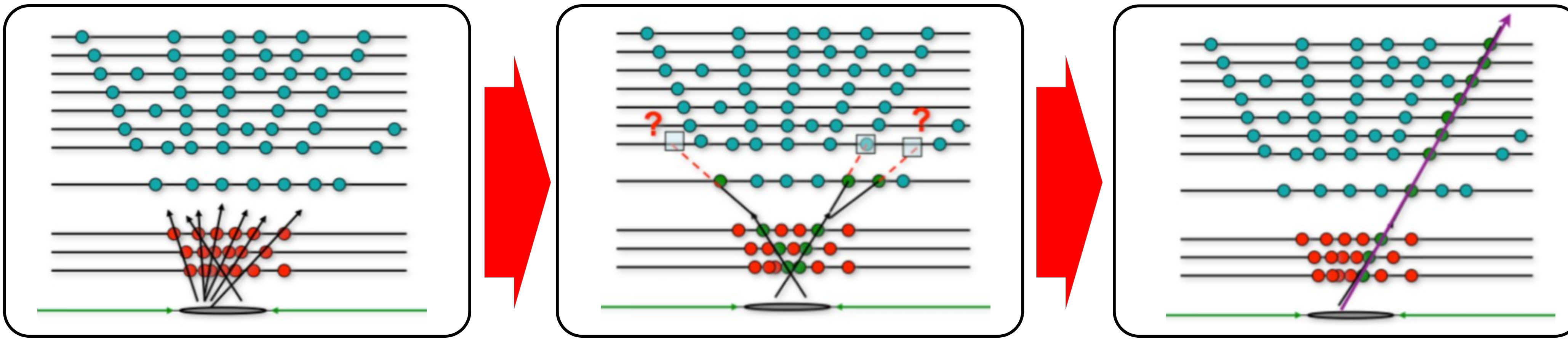


N.B. Not a complete coverage of TPC tracking in this talk.

Tracking in a Nutshell

- Particle trajectory bended in a solenoidal magnetic field
- Curvature is a proxy to momentum
- **Thousands of sparse hits**
- Hits pollution from low momentum, secondary particles

Seeding — Combinatorial Kalman Filter — Fitting with Kalman Filter

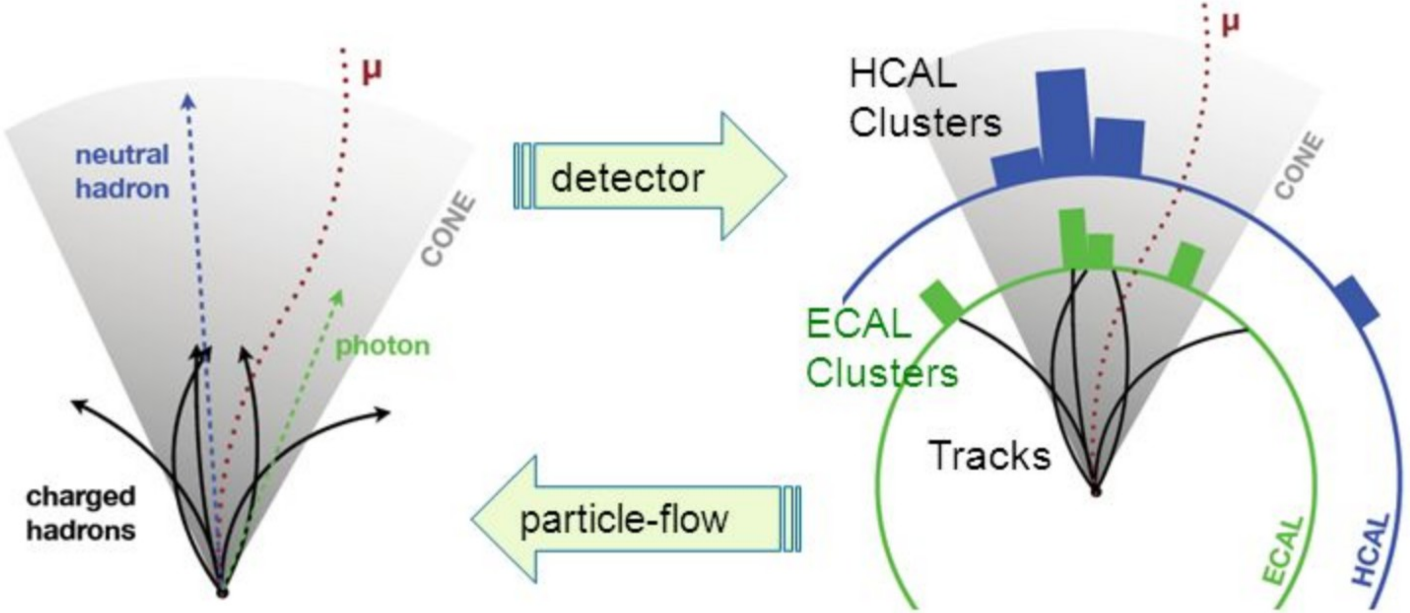


- **Explosion in hit combinatorics** in both seeding and stepping pattern recognition
- **Highly computing consuming task** in extracting physics content from LHC data

Well studied formalism for charged particle reconstruction achieves high performance.

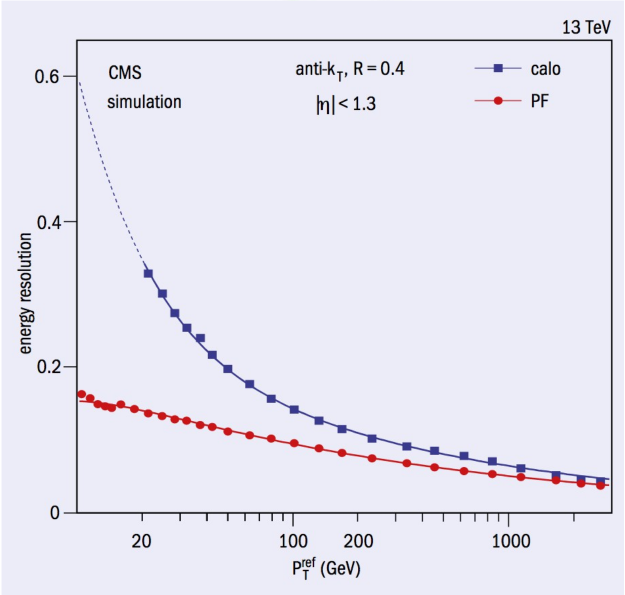
Impact of Tracking

Jet Reconstruction

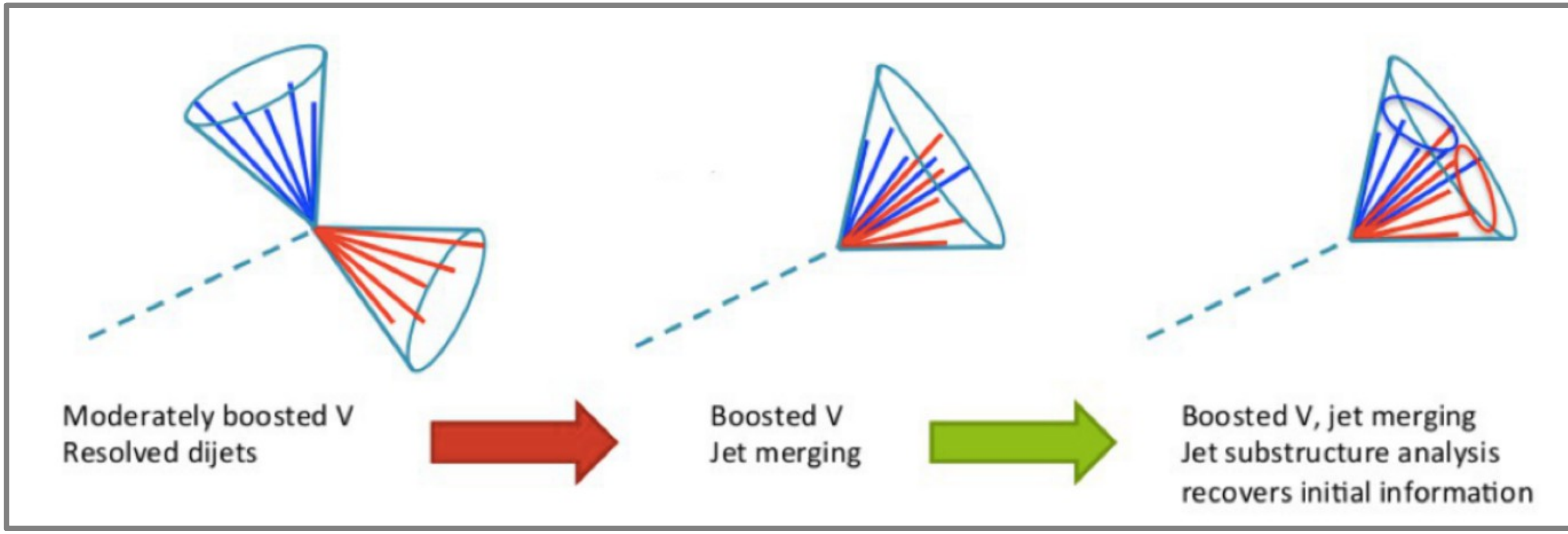


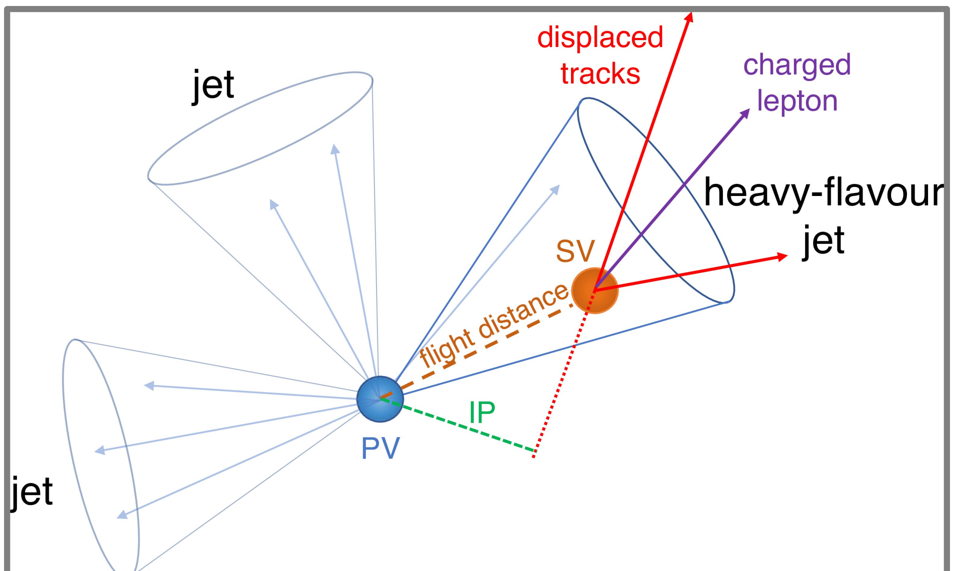
- Tracks are crucial ingredient, together with the calorimeter information in the particle flow algorithm

- Jet reconstructed from particle flow candidates have significantly better energy resolution



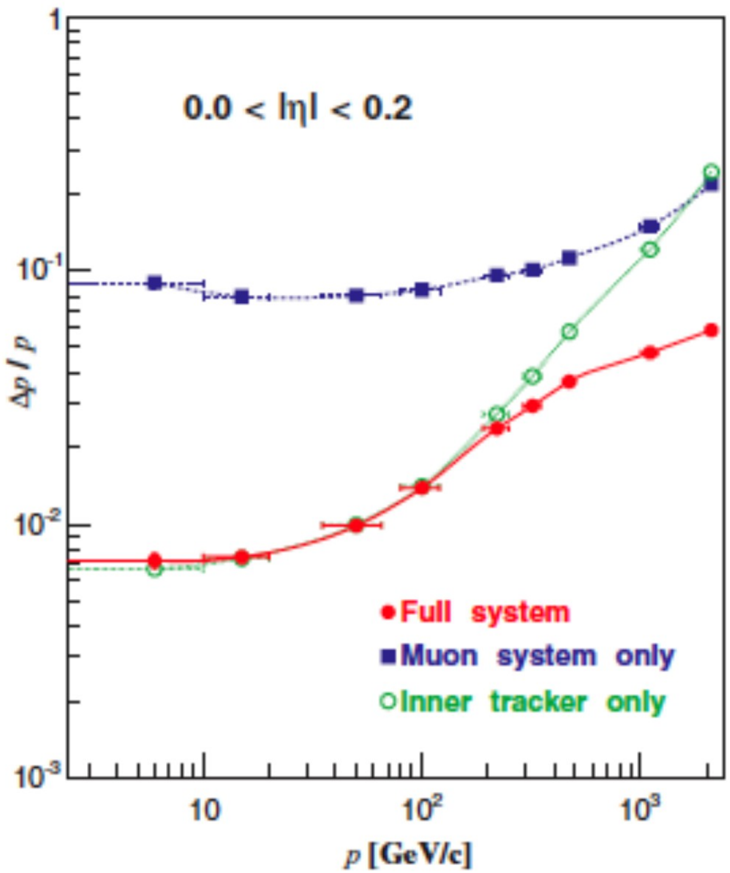
Jet Identification

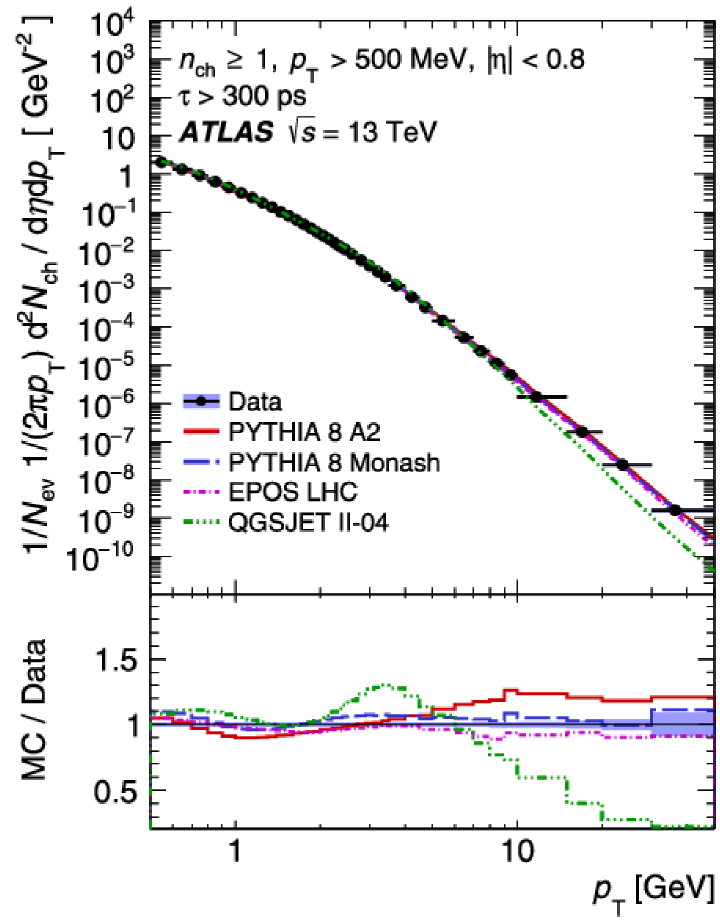




- Jet substructure derived from particle constituents
- Secondary vertex are derived from tracks.
- Tracking is crucial for jet identification

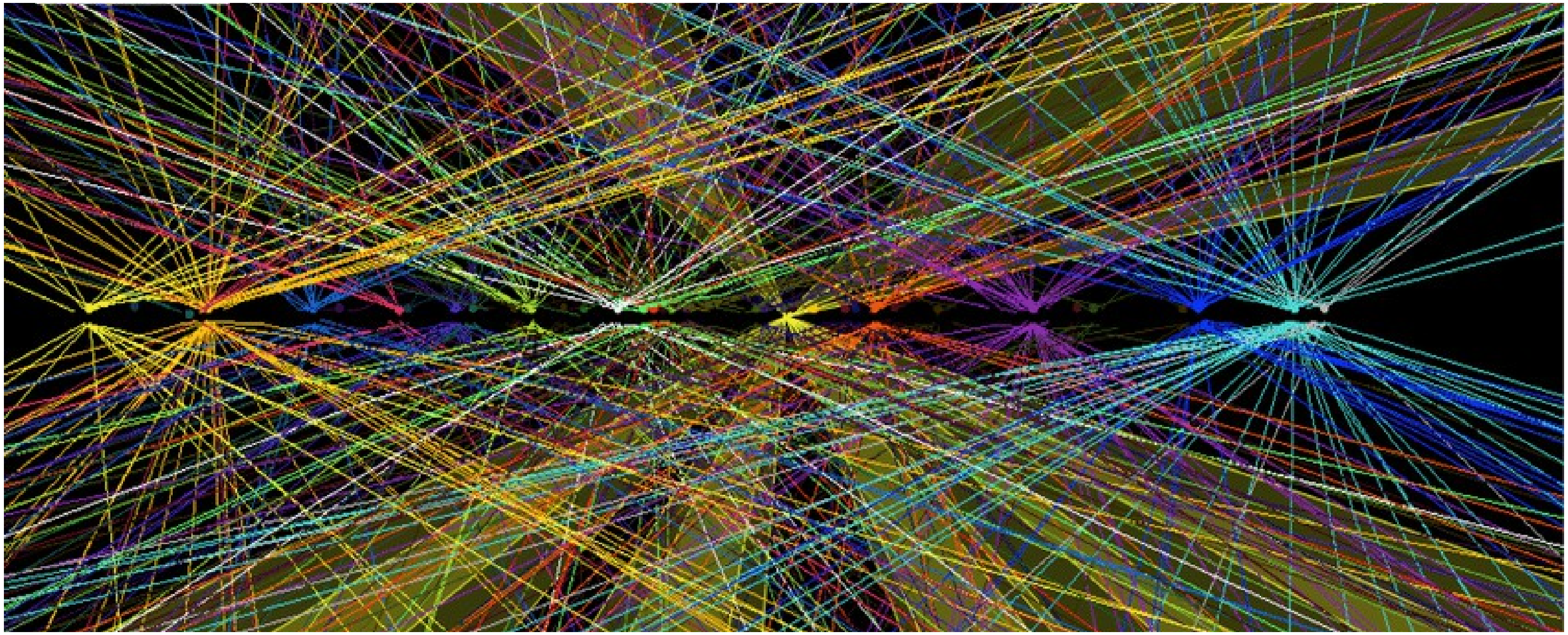
Momentum Resolution





- Experimental resolution is typically much improved at low momentum with tracking device
- Most particles in the collisions are at low momentum

Pile-up Mitigation



- Concurrent proton-proton interaction per bunch crossing are mostly overlapping

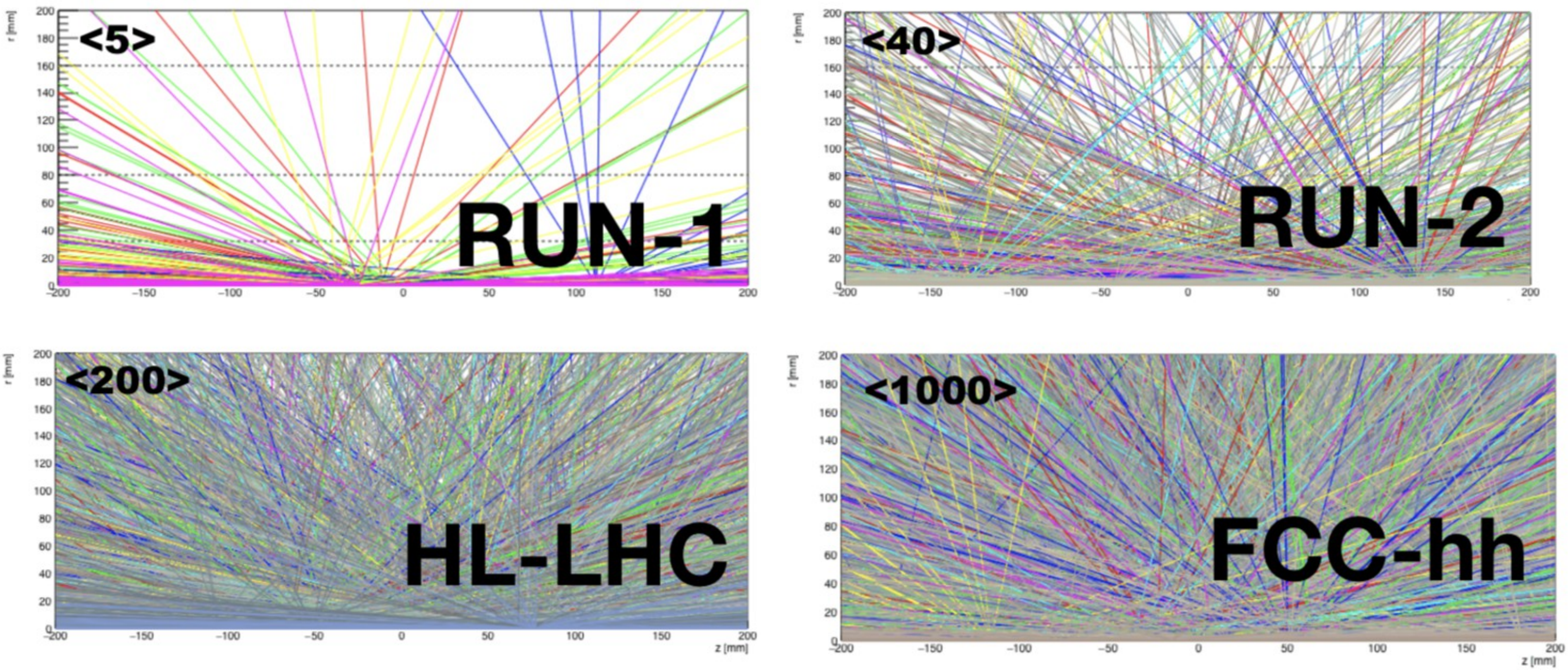
- Determination of separate interactions are made possible with tracks and vertex

- Jet are further improved with Charged Hadron Subtraction

Charged particle reconstruction is a key ingredient the realization of the Physics program at the LHC.

HL-LHC Challenge

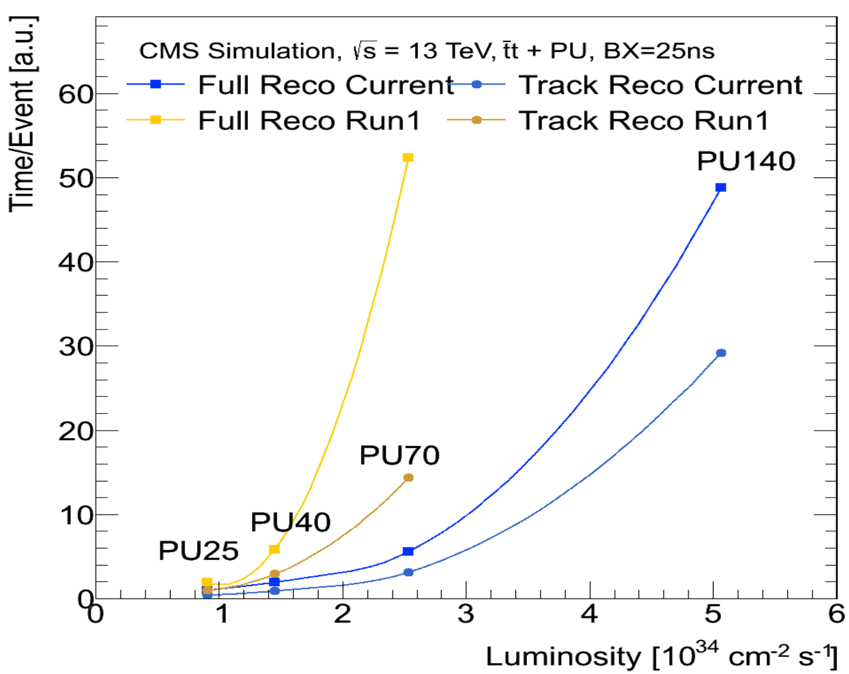
High Luminosity Challenge

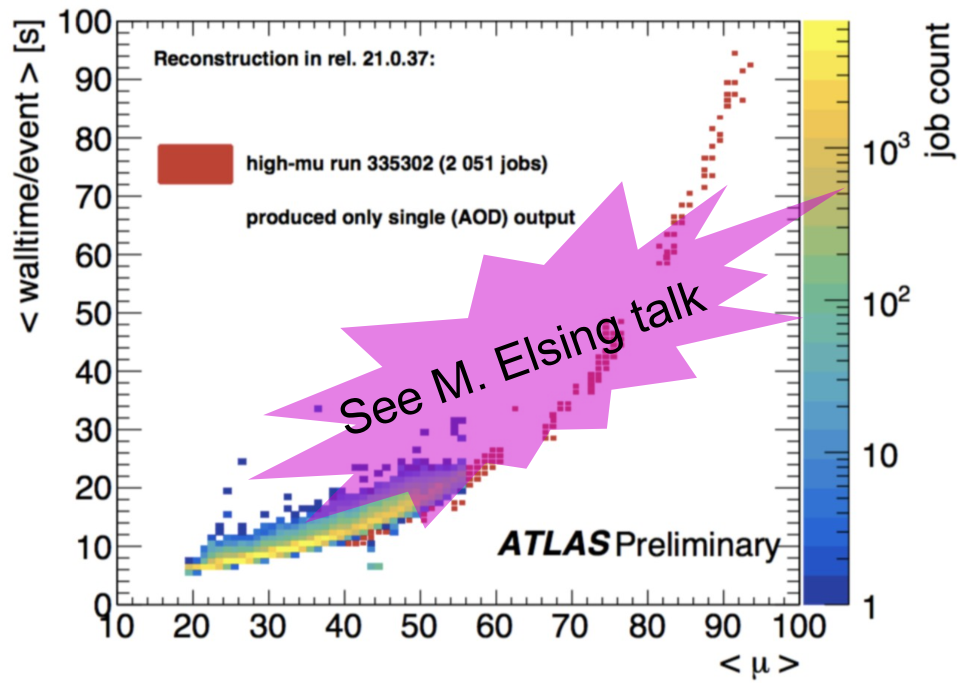


- With higher luminosity comes more reach of rare physics processes
- It also comes with many more pile-up and particles per event

Scaling of Tracking

- Charged particle track reconstruction is one of the **most CPU consuming task** in event reconstruction
- Programatic **optimizations mostly saturated**
- Large fraction of CPU required in the HLT. **Cannot perform tracking** inclusively

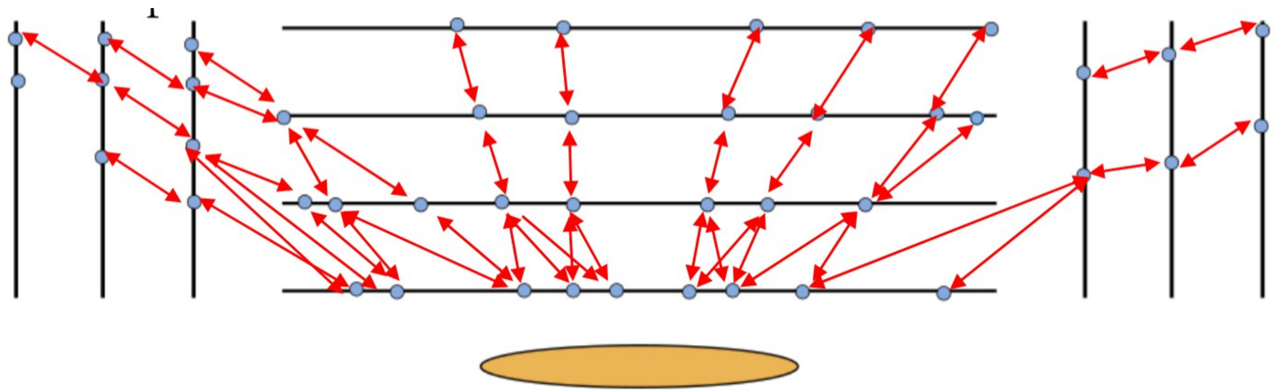




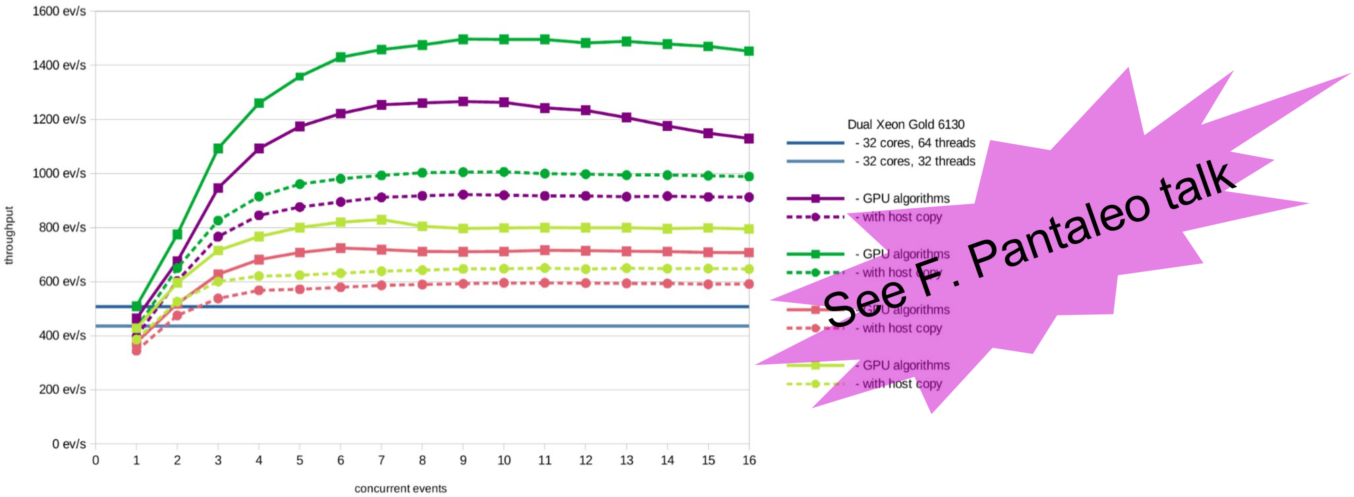
Scaling performance and limits in computation budget call for faster algorithms.

Alternative Approaches

Cellular Automaton Pixel Tracking

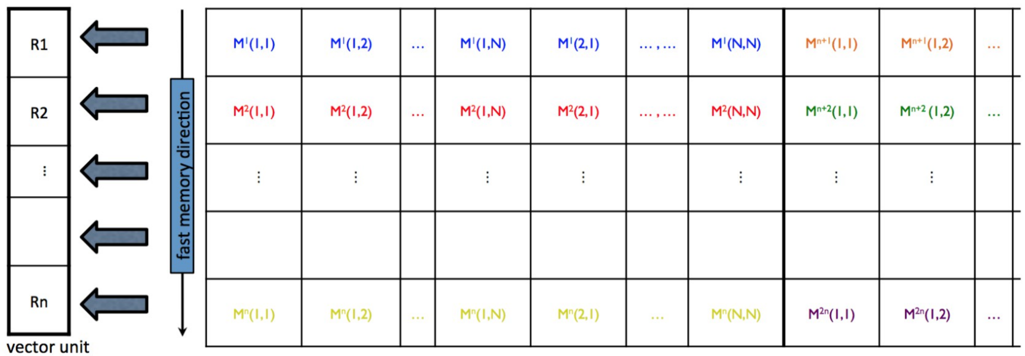


- Outsource track reconstruction in pixel detector to GPU cellular automaton
- Faster, cheaper, more efficient, more precise, ...



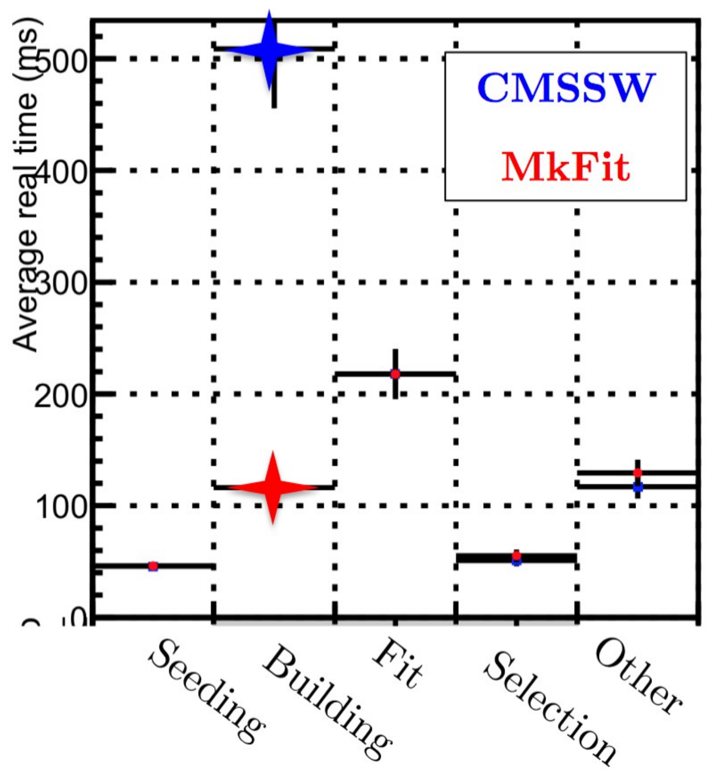
https://iopscience.iop.org/article/10.1088/1742-6596/513/5/052010
https://indico.cern.ch/event/742793/contributions/3274390

Parallelism



Matrix size **NxN**, vector unit size **n**

http://trackreco.github.io/



- Matriplex library for vectorization
- Parallelization of track following
- Pattern recognition can be made faster than traditional track fitting

https://arxiv.org/abs/1906.11744

Hough Transform

Hough algorithm

Discretised maximum likelihood optimisation over

$$L(n\,|\,\{x_i\}) = \sum_i \int \mathrm{d}n\, \delta(d(n, x_i))$$

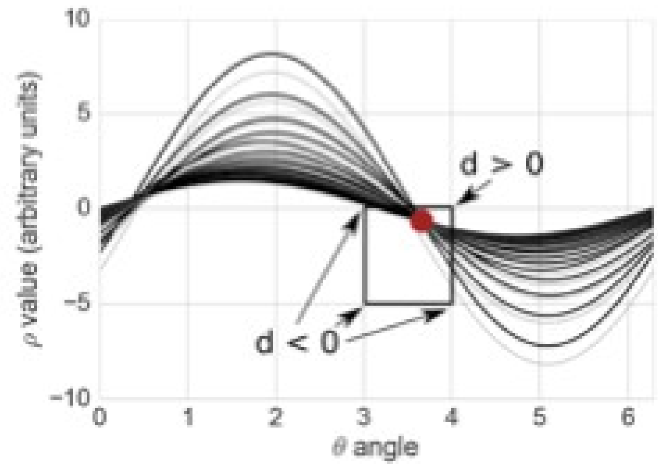
where d is the distance measure of track to hit.
Typically carried out as

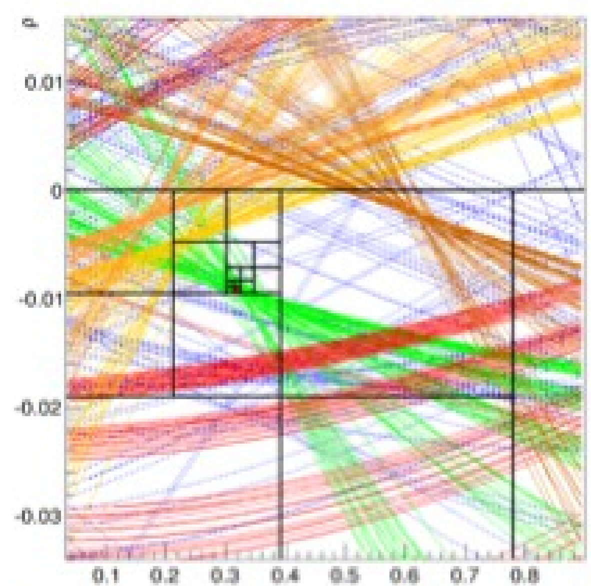
- grid search
- *Fast Hough* bisecting each dimension

over small volumes dn of the parameter space evaluating only the signs of d on the edges.

Refinements

- Weighting of hits versus tracks e.g. on distance d or prior distributions
- Priorisation of search areas
- Overlapping volumes

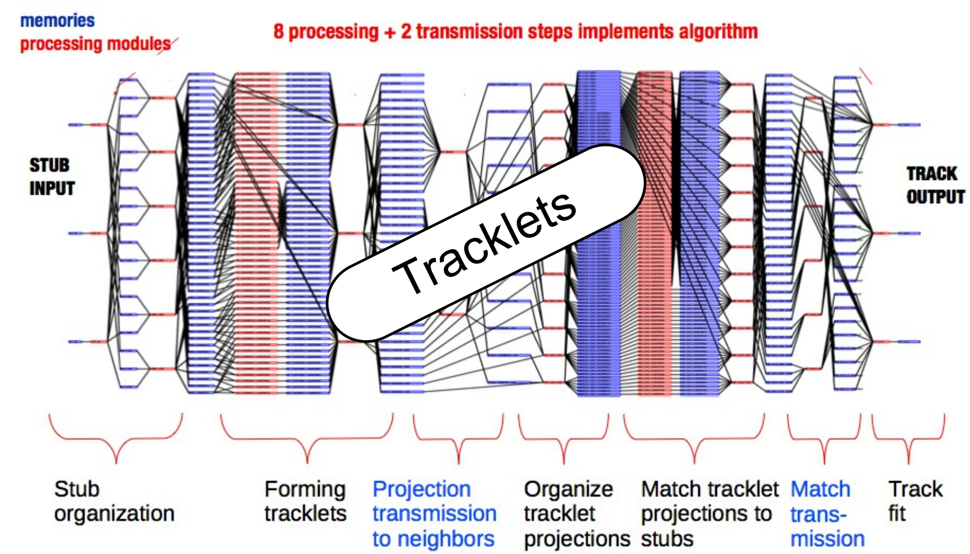


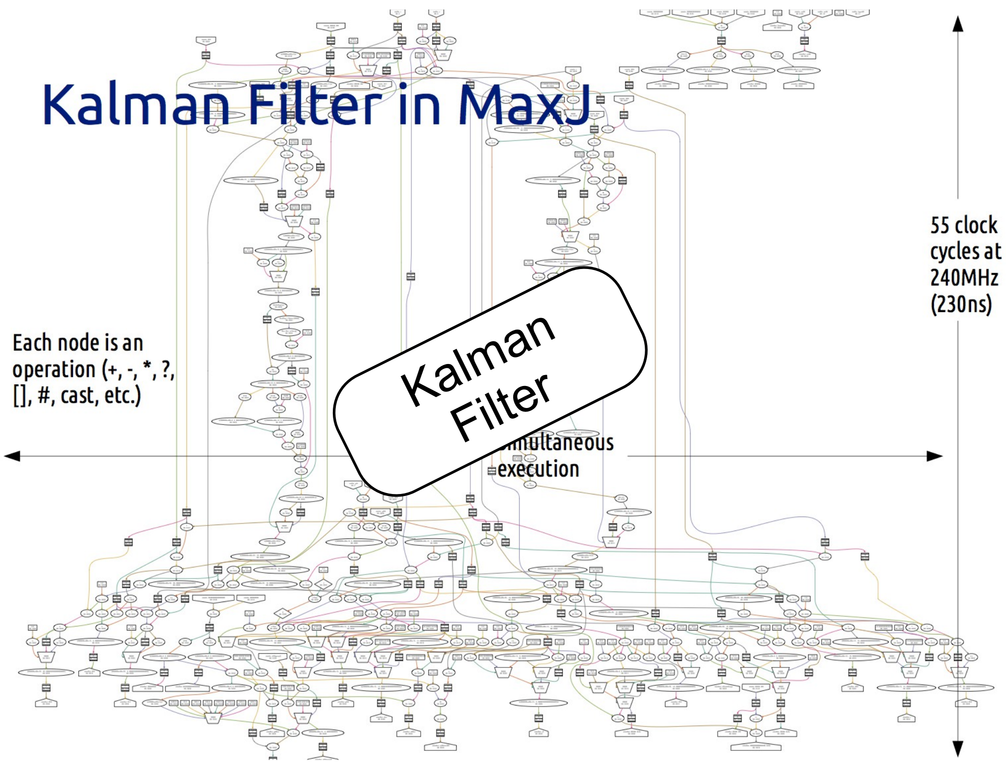


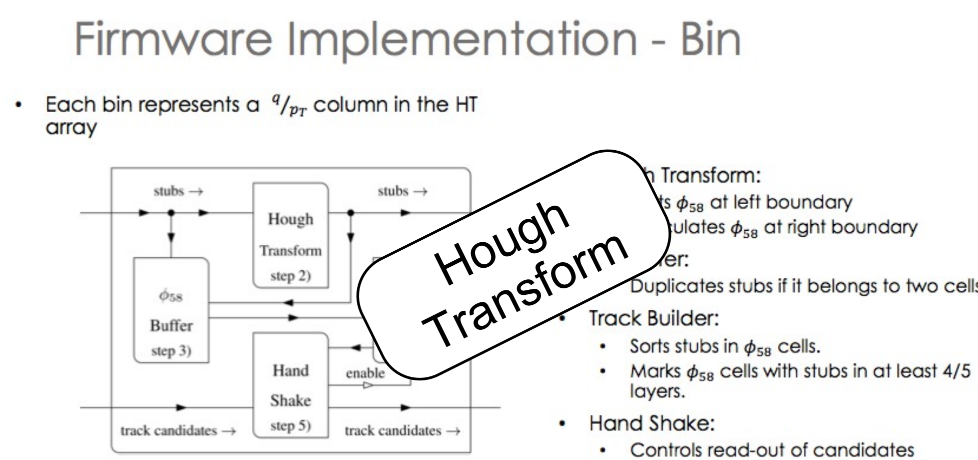
Oliver Frost on behalf of the Belle II collaboration | DESY | 2016-02-22 | Page 4 / 23

Fast Hardware Tracking

- Track trigger implementation for Trigger upgrades development on-going
- Several approaches investigated
- **Dedicated hardware is the key** to fast computation.
- **Not applicable for offline** processing unless by adopting heterogeneous hardware.

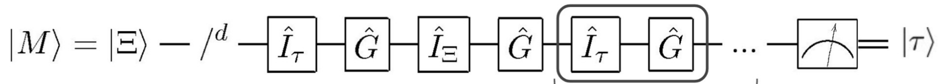






See https://ctdwit2017.lal.in2p3.fr/ and https://indico.cern.ch/event/742793/

Quantum Associative Memory



Grover's cycle. Repeated $T_j = NI\left(\frac{(j+1/2)\pi - \arctan\left(\frac{\bar{k}(0)}{\bar{l}(0)}\sqrt{\frac{m}{N-m}}\right)}{\arccos(1-\frac{2m}{N})}\right)$, $j = 0,1,2,\ldots$ times.

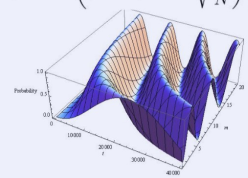
$\sum_{i=1}^{m} k_i(t)|x_i\rangle + \sum_{i=m+1}^{N} l_i(t)|x_i\rangle$

States that match the target pattern.

States that don't match the target pattern.

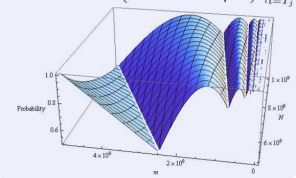
Probability "ramp-up" vs. pattern matches

$P(t,m) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Big|_{N=10^9}$



Peak probability vs. pattern matches and memory capacity

$P(m,N) = \sin^2\left((2t+1)\arcsin\sqrt{\frac{m}{N}}\right)\Big|_{t=T_j}$



$m = 1, N = 10^9 : T_0 = 24836, P_{max} = 0.9999999999965568$

$m = 20, N = 10^9 : T_0 = 5553, P_{max} = 0.9999999991404647$

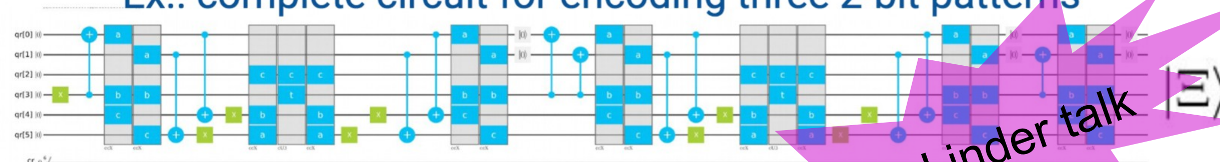
Note: neither quantum noise, nor probabilistic memory cloning operations, are taken into account here.

* $\hat{I}_\tau$ - "quantum oracle" operator. Inverts the phase of state representing the target pattern τ.
$\hat{G}$ - Grover's diffusion operator. Inverts all amplitudes about the amplitudes average.
$\hat{I}_\Xi$ - Inverts phases of all terms originally present in memory.

- Similar to associative memory method with exponentially more storage capacity
- Limitation on hardware size in demonstrator

▸ **QuAM storage circuit generator [implemented]**

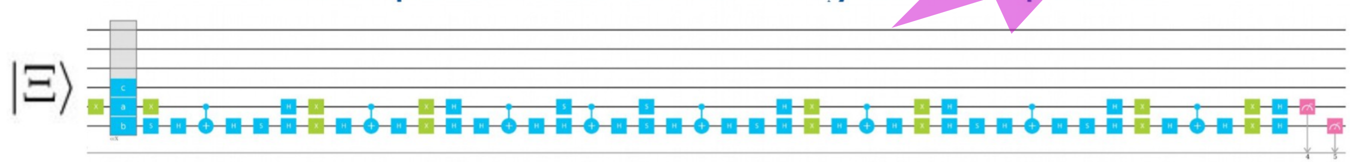
Ex.: complete circuit for encoding three 2-bit patterns



See L. Linder talk

▸ **QuAM retrieval circuit generator [being tested]**

Ex.: complete circuit for retrieving one 2-bit pattern



https://arxiv.org/abs/1902.00498

Tracking with Quantum Annealing

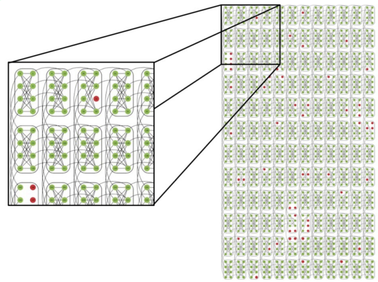
Runs over **all** quBit pairs

$$H_{\text{Ising}} = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$$

External magnetic field

Interactions

Chimera graph embedding



Runs over **adjacent** quBits

$$H_{\text{Ising}} = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$$

External magnetic field

Interactions

- ~1000 qubits on chimera graph can only encode ~40 qubits full Ising Hamiltonian
- Quadratic Unconstrained Binary Optimization (QUBO) can be mapped to an Ising Hamiltonian with change of variable {0,1}↔{-1,1}

Helix Term Segments along an helix

High pT Term Aligned pair of segments

Beam spot Term Segment pointing at the origin

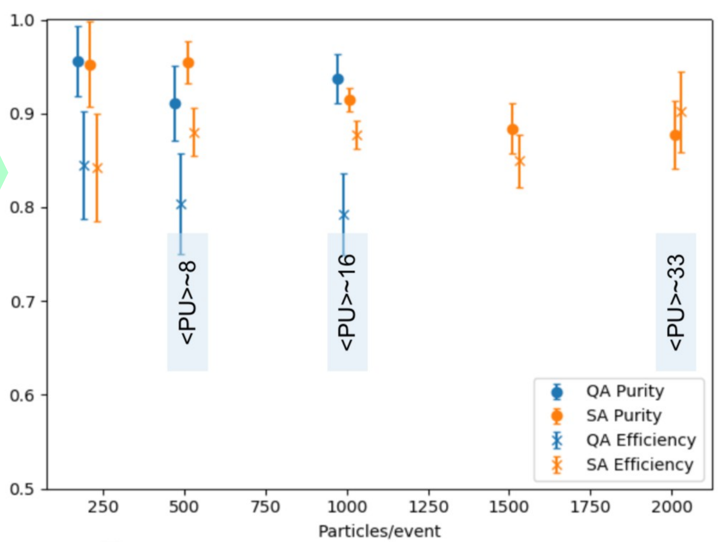
$$\sum_{a,b,c}\left(\frac{\cos^{\lambda}\theta_{abc}+\rho\cos^{\lambda}\phi_{abc}}{r_{ab}+r_{bc}}+\eta\left(z_c-r_c\frac{z_c-z_a}{r_c-r_a}\right)^{\zeta}\right)s_{ab}s_{bc}$$

$$+\alpha\left(\sum_{a,b\neq c}s_{ab}s_{ac}+\sum_{a\neq b,c}s_{ac}s_{bc}\right)+\sum_{a,b}\left(\beta+\gamma P(s_{ab})\right)s_{ab}$$

Bifurcation Term No shared hits on valid segment

Inhibition Term Reduced number of segments

GP Term Use the quality of segment



- QUBO formalism inspired by Hopfield network
- Pattern recognition on Dwave system limited by hardware size

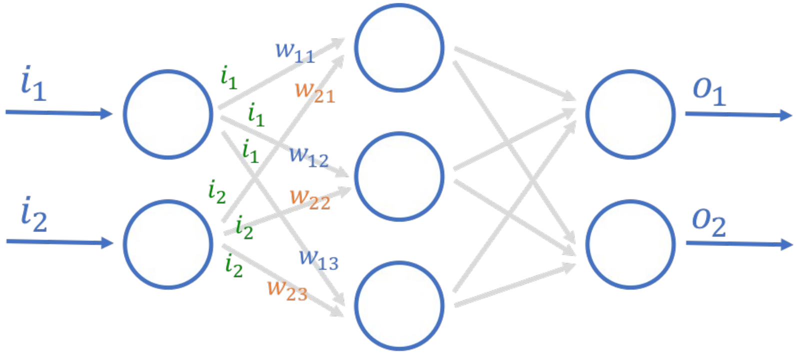
See L. Linder talk

https://arxiv.org/abs/1908.04475

There are other possible ways than machine learning to do tracking.

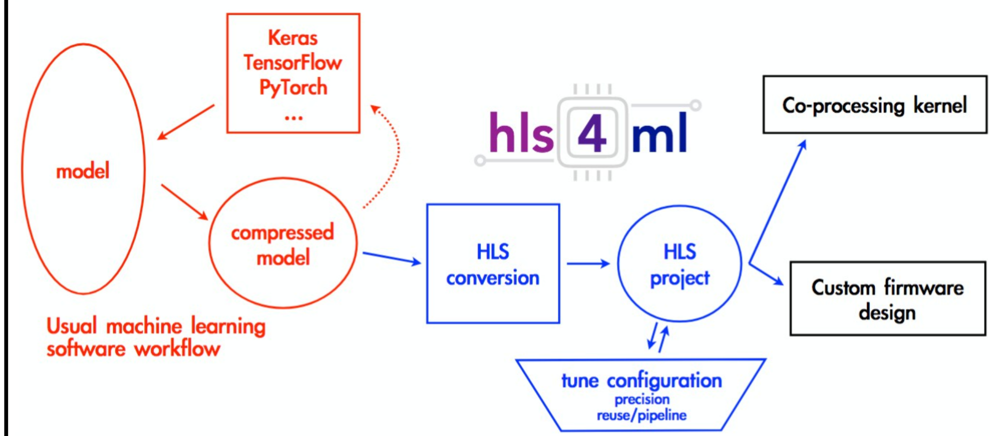
The Case for Machine Learning

Computational Aspect



ANN ≡ matrix operations ≡ parallelizable

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11}\times i_1) + (w_{21}\times i_2) \\ (w_{12}\times i_1) + (w_{22}\times i_2) \\ (w_{13}\times i_1) + (w_{23}\times i_2) \end{bmatrix}$$



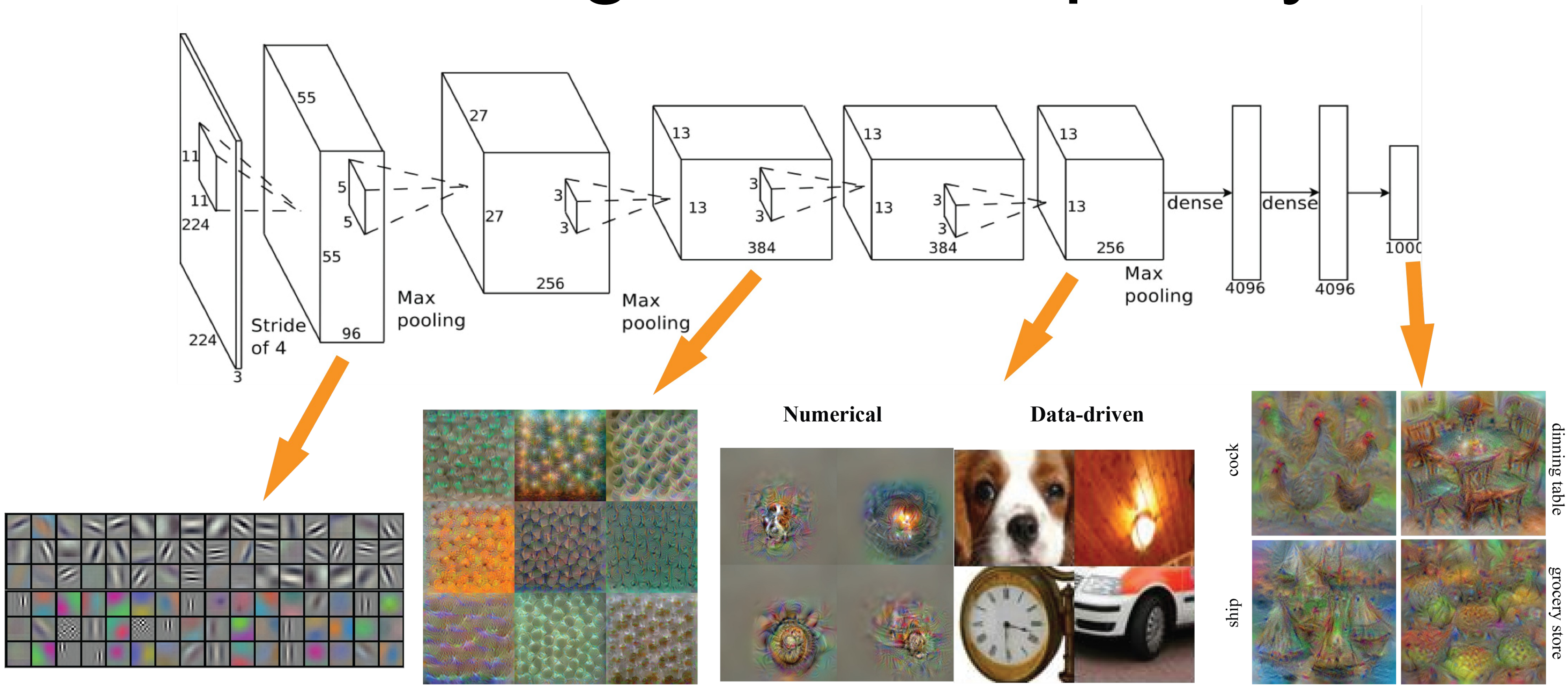
Synthesis FPGA firmware
from trained ANN

https://hls-fpga-machine-learning.github.io/hls4ml/

Bonsai BDT, contained tree growth and feature discretization ; fast classification
https://arxiv.org/abs/1210.6861

- Computation for machine learning prediction from a trained model is parallel and can be fast

Learning from Complexity



Conv 1: Edge+Blob Conv 3: Texture Conv 5: Object Parts Fc8: Object Classes

- Machine learning can extract useful information from complex underlying data structure
- Classical algorithm counter part may take years of development

Scene Labeling

Zagoruyko et al, https://arxiv.org/pdf/1604.02135.pdf

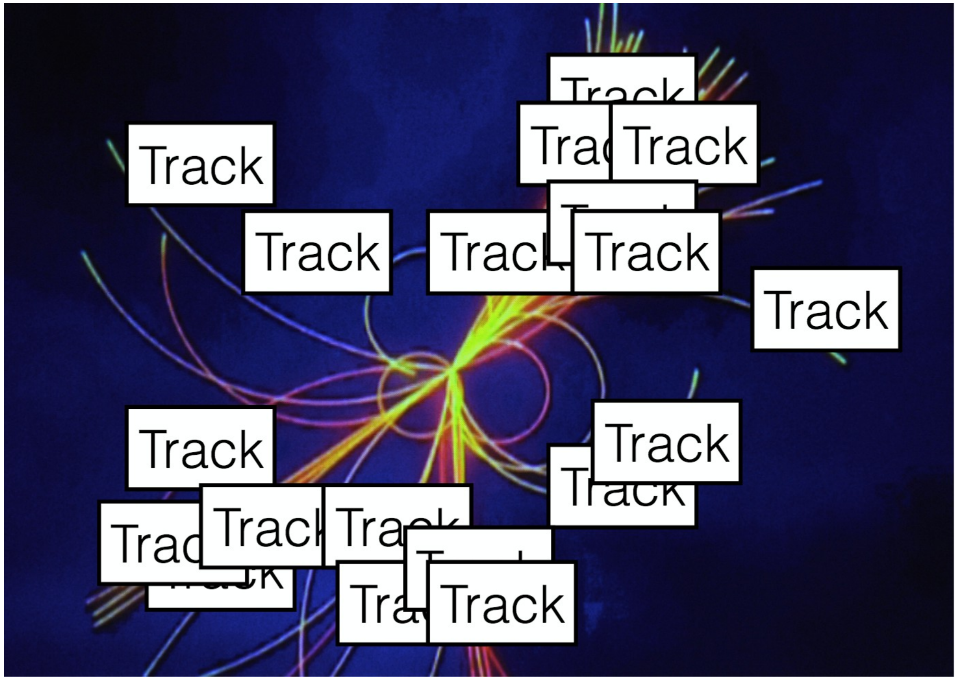


Photo by Pier Marco Tacca/Getty Images

- Recent image processing deep learning applications perform non-trivial image segmentation
- Potential application to hit/track association problem

The Learnable Things

What machine learning model **could** learn

- Amount of material
- Magnetic field
- Alignment
- non-gaussian modeling
- Stochastic process

What machine learning model **should not have** to learn

- Helix dynamic
- ... all physics we know of
- Bethe-Bloch

- In practice, it is not always easy to inject domain knowledge

Machine learning may provide ways to improve tracking or solve the computation issue.

Challenges for Pattern Recognition

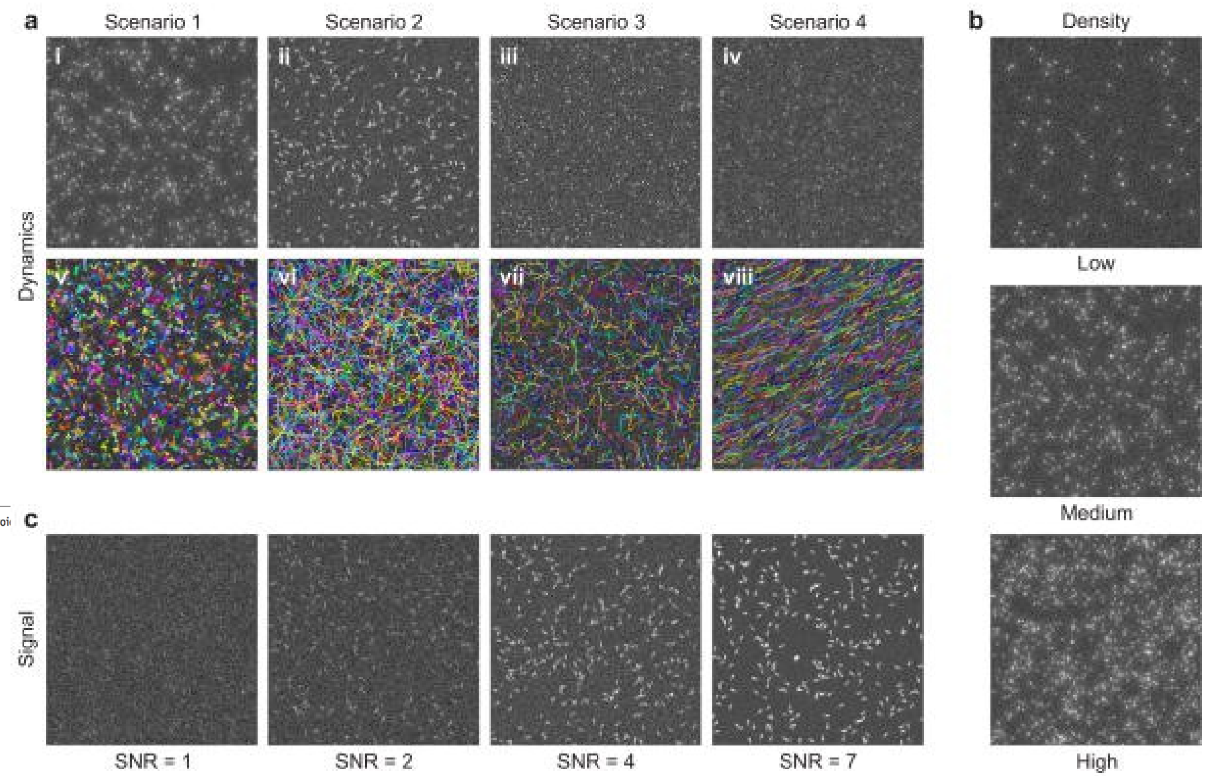
Particle Tracking in Biology

https://www.ncbi.nlm.nih.gov/pubmed/24441936

Table 1 | Participating teams and tracking methods

		Detection			Linking				
Method	**Authors**	**Prefilter**	**Approaches**	**Remarks**	**Principle**	**Approaches**	**Remarks**	**Dim.**	**Refs.**
1	I.F. Sbalzarini Y. Gong J. Cardinale	–	M, C	Iterative intensity-weighted centroid calculation	Combinatorial optimization	MF, MT, GC	Greedy hill-climbing optimization with topological constraints	2D & 3D	32
2	C. Carthel S. Coraluppi	Disk	M, T	Adaptive local-maxima selection	Multiple hypothesis tracking	MF, MT, MM	Motion models are user specified (near-constant position and/or velocity)	2D & 3D	33,34
3	N. Chenouard F. de Chaumont J.-C. Olivo-Marin	Wavelets	M, T	Maxima after thresholding two-scale wavelet products	Multiple hypothesis tracking	MF, MT, MM, GC	Motion models are user specified (near-constant position and/or velocity)	2D & 3D	35–37
4	M. Winter A.R. Cohen	Gaussian, median and morphology	M, T, C	Adaptive Otsu thresholding	Multitemporal association tracking	MF, MT, GC	Post-tracking refinement of detections	2D & 3D	38,39
5	W.J. Godinez K. Rohr	Laplacian of Gaussian or Gaussian fitting	M, T, F, C	Either thresholding + centroid or maxima + Gaussian fitting	Kalman filtering + probabilistic data association	MF, MM	Interacting multiple models using	2D & 3D	29,40
6	Y. Kalaidzidis	Windowed floating mean background subtraction	T, F	Lorentzian function fitting to structures above noise level	Dynamic programming				
7	L. Liang J. Duncan H. Shen Y. Xu	Laplacian of Gaussian	M, T, F	Gaussian mixture model fitting	Multiple hypothesis tracking				
8	K.E.G. Magnusson J. Jaldén H.M. Blau	Deconvolution	M, T, F	Watershed-based clump splitting and parabola fitting	Viterbi algorithm on state-space representation				
9	P. Paul-Gilloteaux	Laplacian of Gaussian or Gaussian filtering	M, T, F	Either maxima with pixel precision (2D) or thresholding + Gaussian fitting (3D)	Nearest neighbor + global optimization				
10	P. Roudot C. Kervrann F. Waharte	Structure tensor	T, F	Histogram-based thresholding and Gaussian fitting	Gaussian template matching				
11	I. Smal E. Meijering	Wavelets	M, F, C	Gaussian fitting (round particles) or centroid calculation (elongated particles)	Sequential multiframe assignment				
12	J.-Y. Tinevez S.L. Shorte	Difference of Gaussian	M, T, F	Parabolic fitting to localized maxima	Linear assignment problem				
13	J. Willemse K. Celler G.P. van Wezel	Gaussian and top hat	T, C	Watershed-based clump splitting	Nearest neighbor				
14	H.-W. Dan Y.-S. Tsai	Gaussian, Wiener and top hat	T, C	Morphological opening–based clump splitting	Nearest neighbor + Kalman filtering				

See **Supplementary Note 1** for further details on methods 1–14. Dim, dimensionality. Detection approaches: M, maxima detection; T, thresholding; F, fitting; C, centroi GC, gap closing.



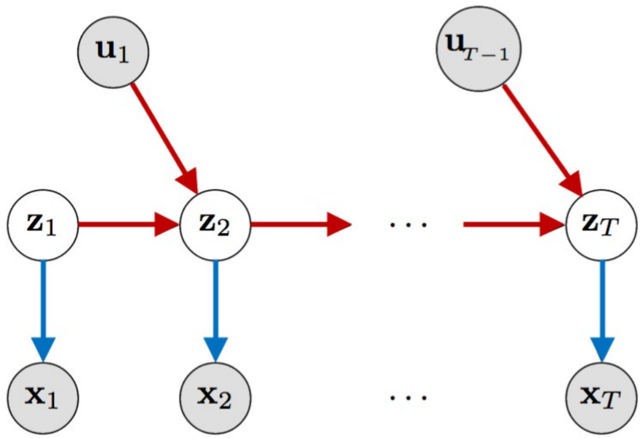
Deep Kalman Filter

Deep Kalman filters

Actions u_t
(e.g., prescribing a medication, performing a surgery)

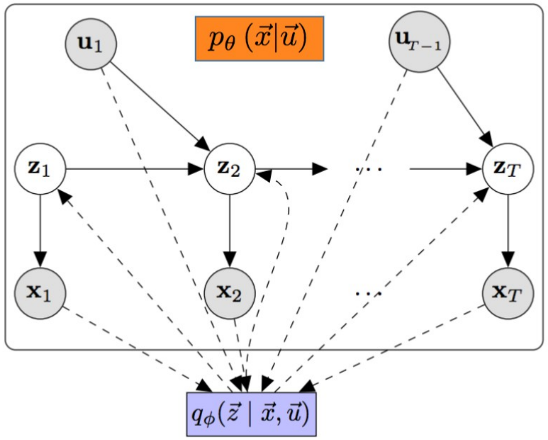
Patient latent state $\mathbf{z_t} \in \mathbb{R}^d$

Observations x_t: Lab test results, diagnosis codes, etc.



Optimize *jointly* over generative model $p_\theta(\vec{x}|\vec{u})$ and variational approximation $q_\phi(\vec{z}|\vec{x},\vec{u})$

Stochastic backpropagation
(Rezende et al. 2014, Kingma & Welling, 2014)



Initial state: $z_1 \sim \mathcal{N}(\mu_0, \Sigma_0)$

Action-transition: $z_t \sim \mathcal{N}\left(G_\alpha(z_{t-1}, u_{t-1}), S_\beta(z_{t-1}, u_{t-1})\right)$

Emission: $x_t \sim \Pi(F_\kappa(z_t))$

Uri Shalit at DSHEP2016
https://indico.hep.caltech.edu/indico/conferenceDisplay.py?confId=102

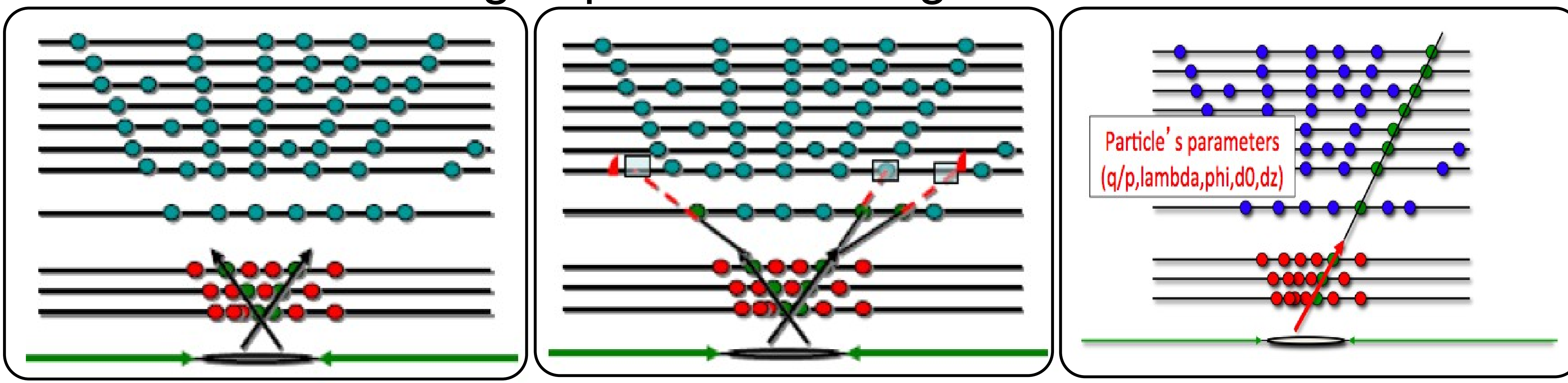
Kalman Filter in Ballistic

- Available methods to track multiple objects using kalman filters
- Deal with “splitting objects”
- Deal with crossing trajetories
- More complexe KF, more computationally intensive ...

Undisclosed contribution during DS@HEP 2016

Pattern Recognition or not

HEP charged particle tracking in a nutshell

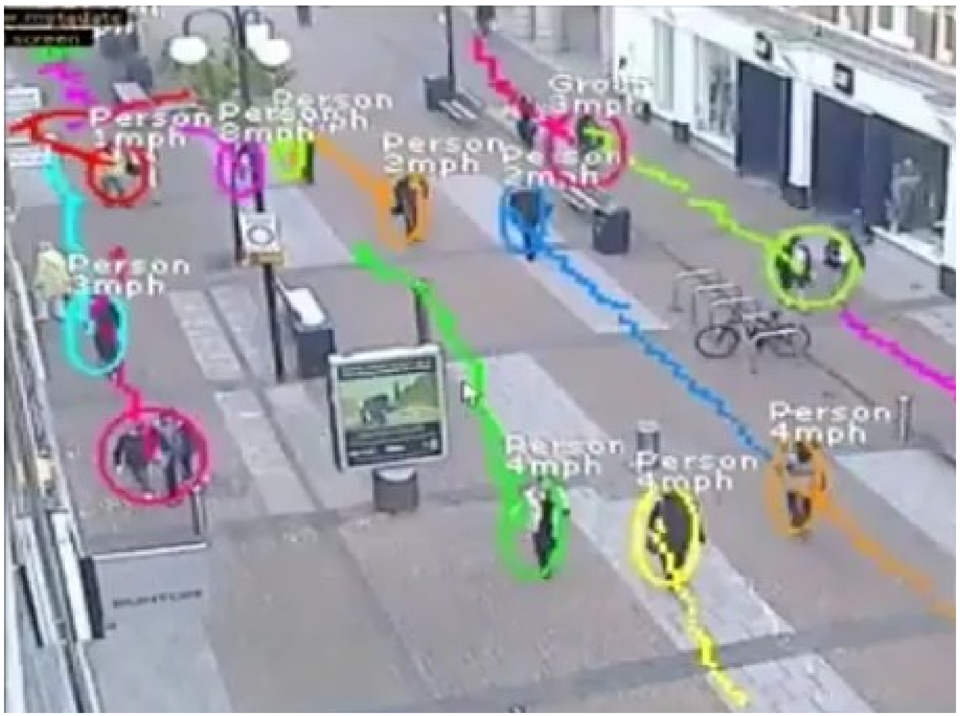


Seeding — Track Building — Track Fitting

- Track building ≡ pattern recognition HEP jargon
- Finding the list of hits belonging to a track ...
- Finding the pattern of hits left by a charged particle in the detector ...

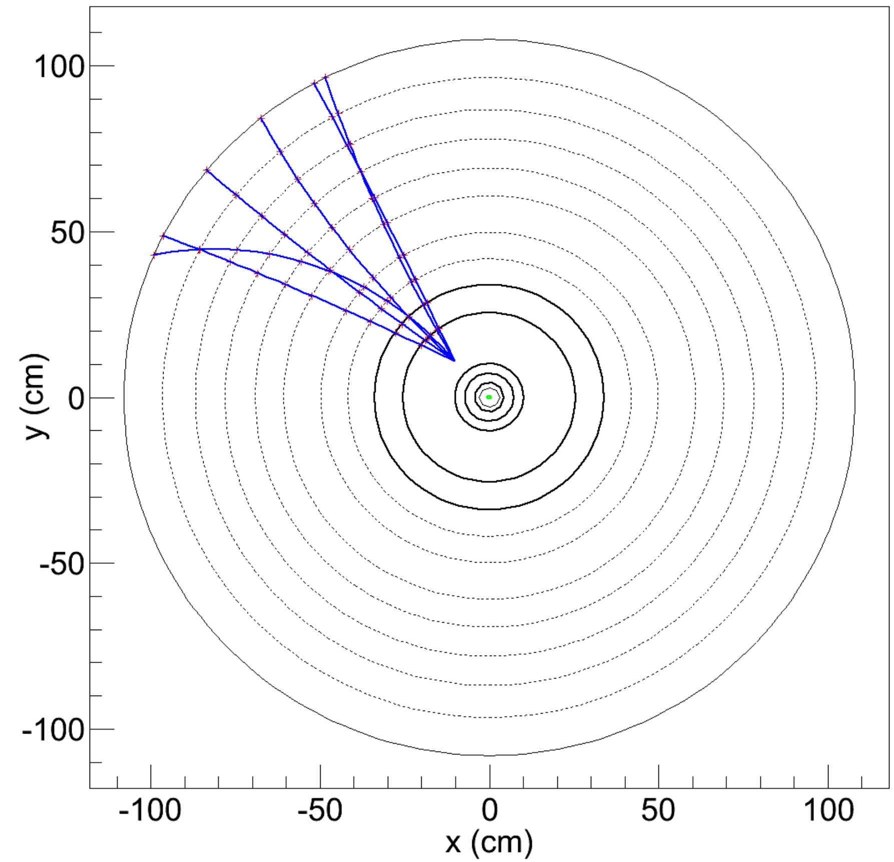
- Not the "usual" data science pattern recognition

Data sparsity

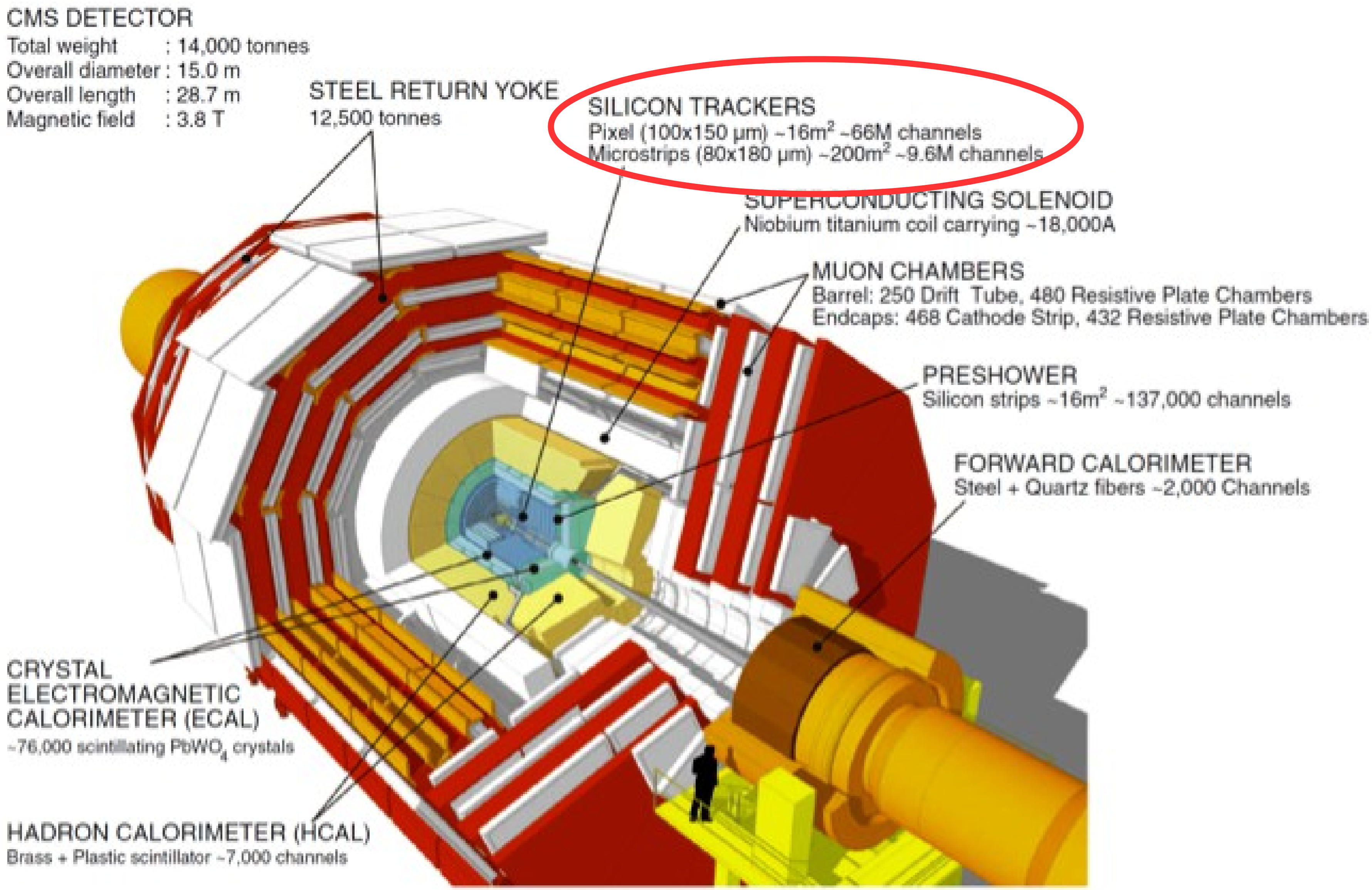


https://privacysos.org/

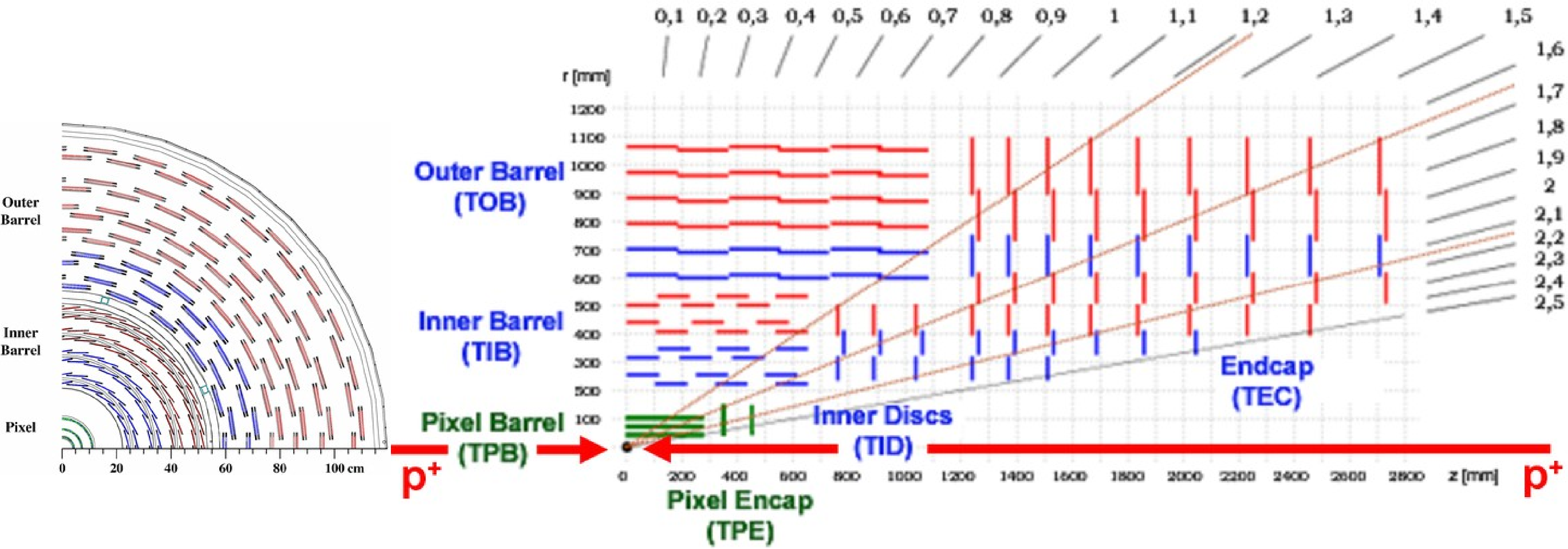
≠



High Dimensionality

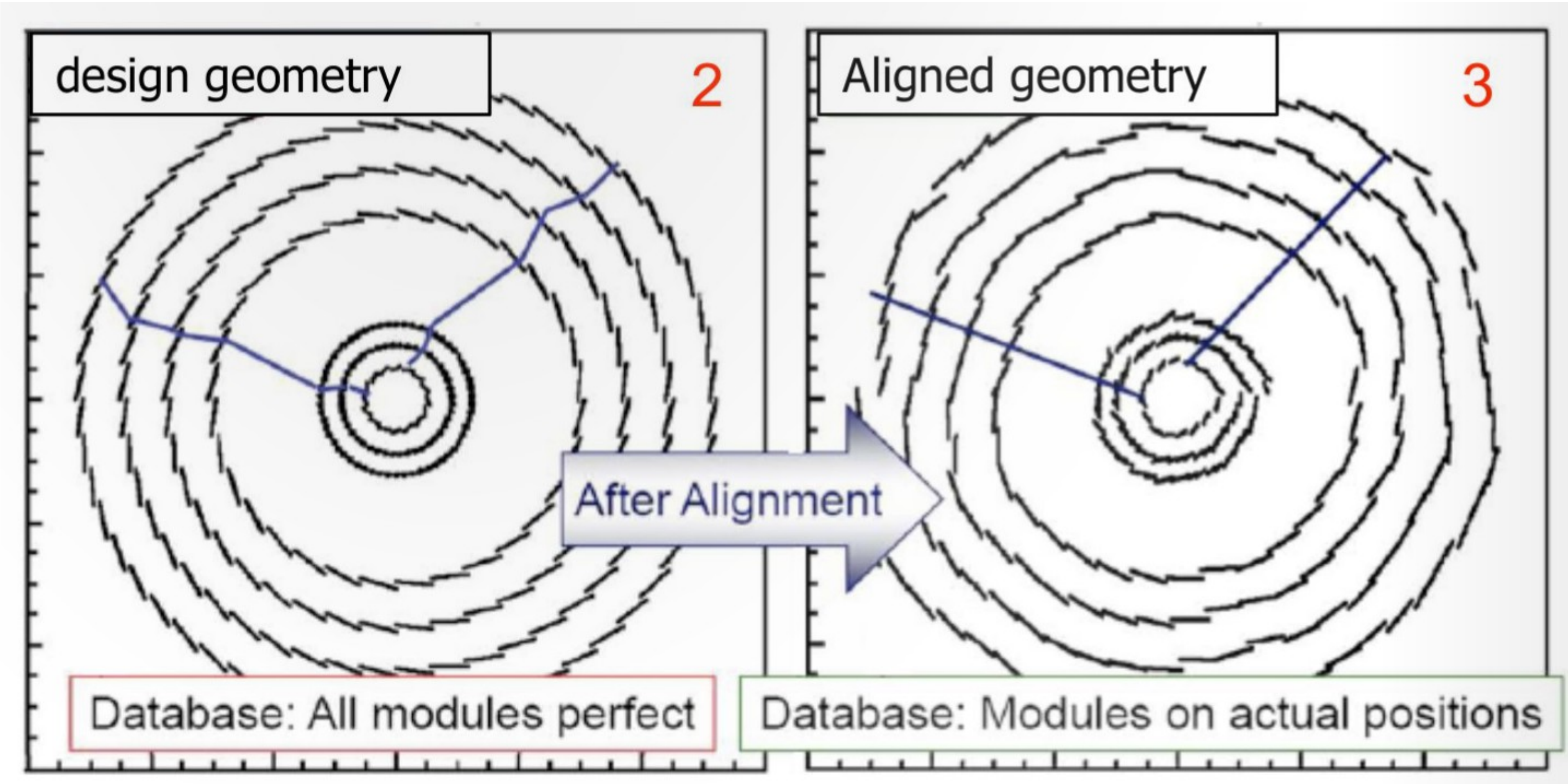


Complex Geometry



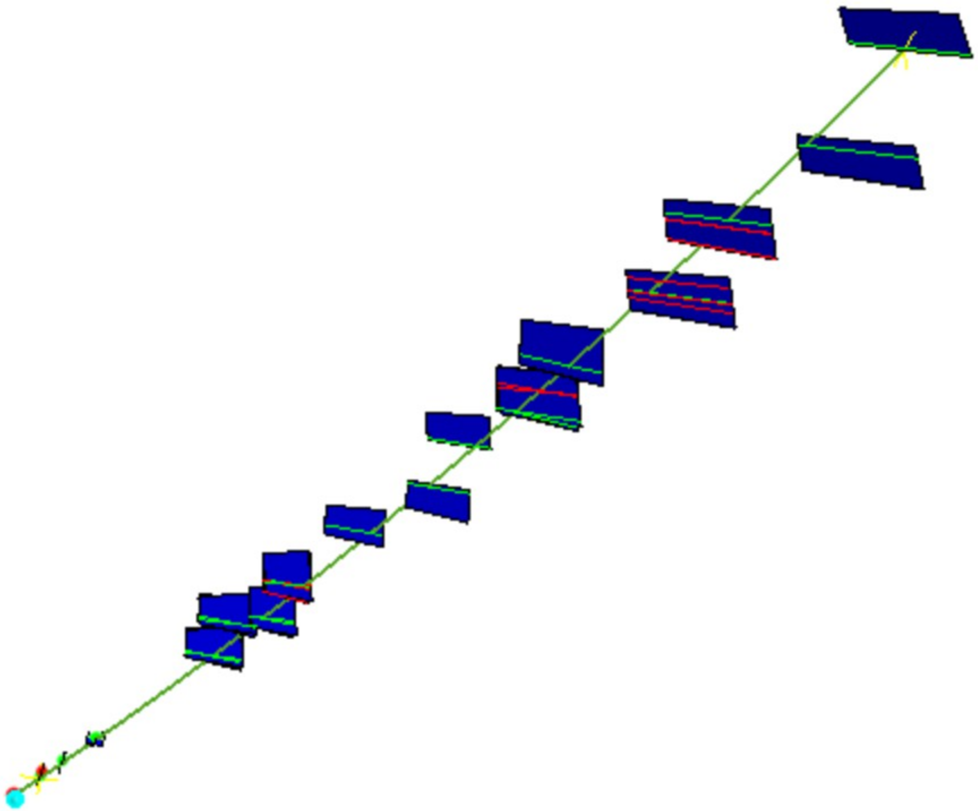
Not the typical data geometry for data science

Mis-aligned Geometry



Mechanical stress (magnetic field, cooling, ...)
does modify the geometry in time

Hit Sequencing



- Hits leave on modules, modules leave on layer, layers are traverse along time.
- “Natural” ordering when trying a hit fitting
- Not so “natural” when doing track building, and hit combinatorics

Figure(s) of Merit(s)

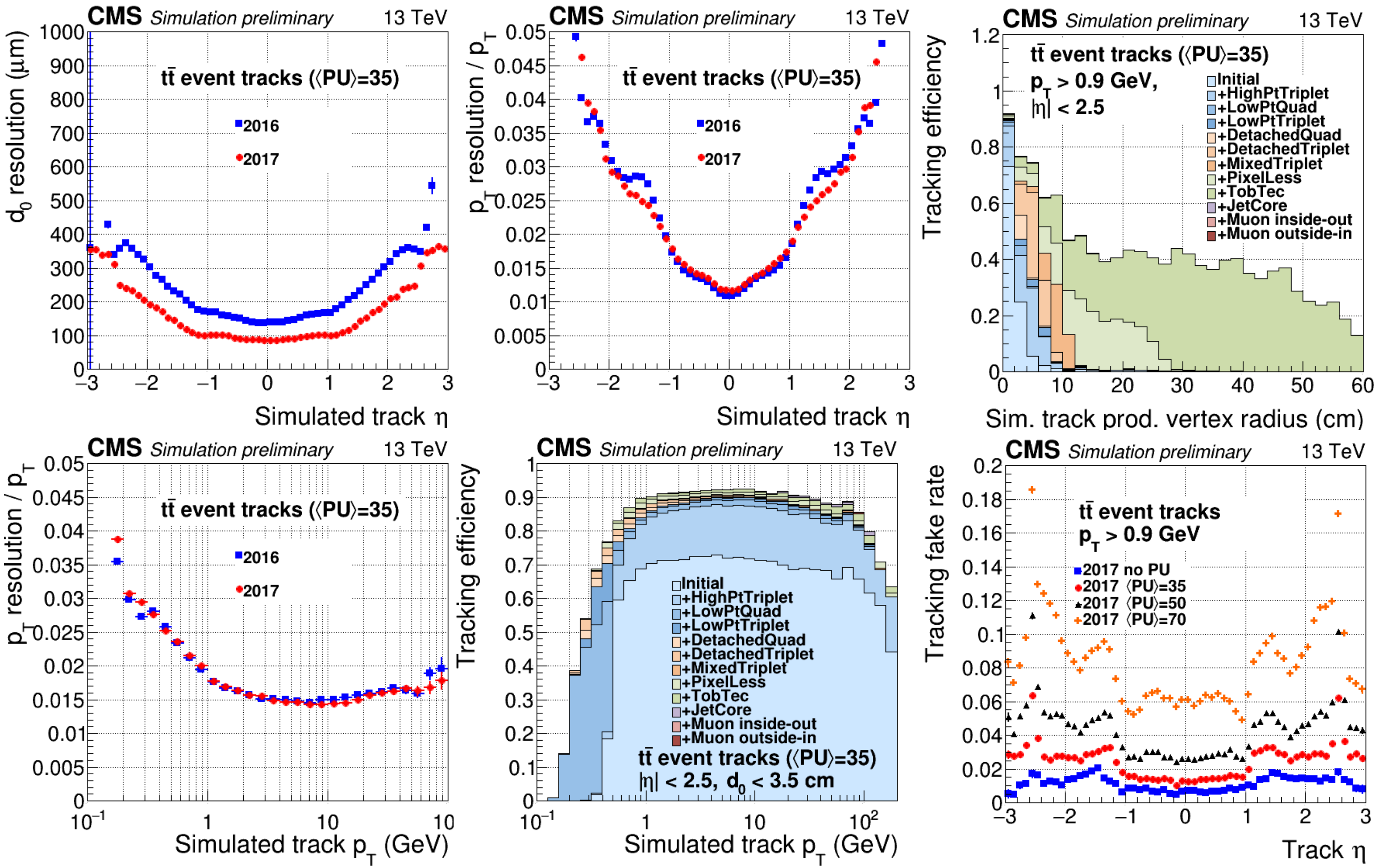


Figure of Merit

- A combination of resolution, fake rate, efficiency, ...

➔ Tracking has been improved within a given a method (CKF+CTF) and within processing time constraints

- Not all tracks are equal. Not all features matter

➔ High dimensional cost function

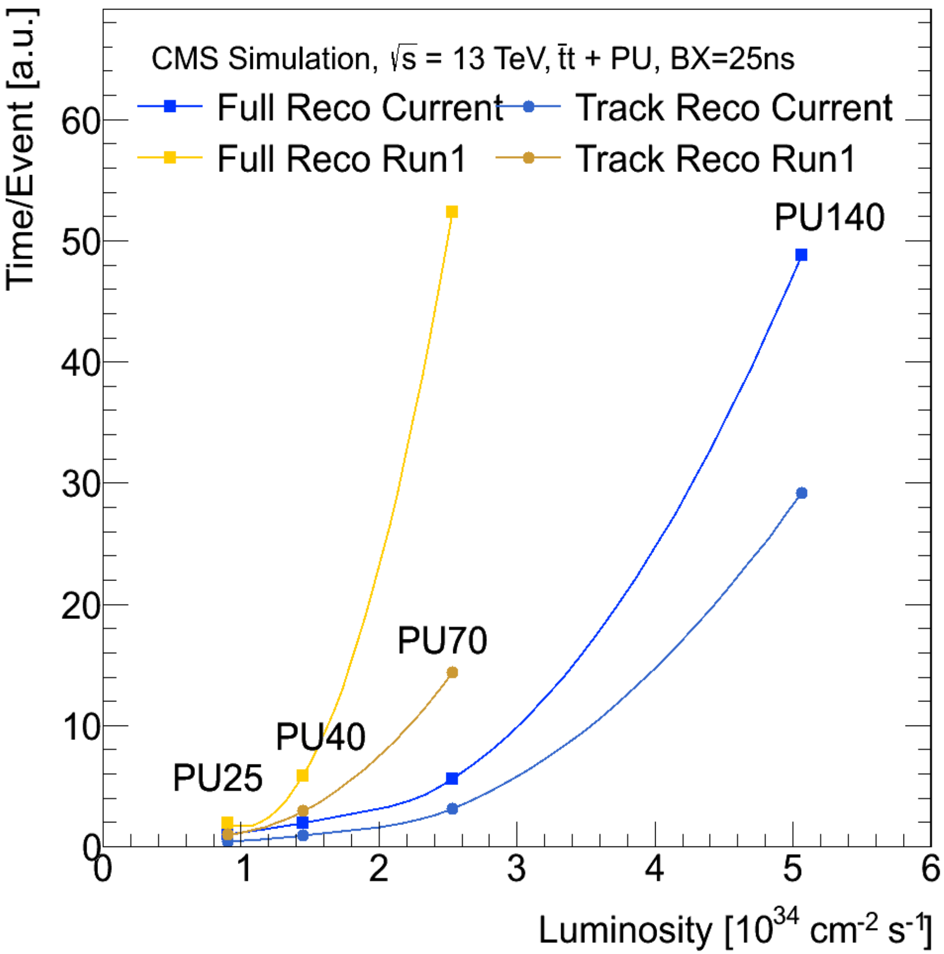
- No golden metric for “tracking” in a general purpose detector

➔ Things would be done differently, if the purpose was different

- Remember the breaking point is computation requirement

➔ Not something that folds in a cost function ...

Computation Performance



- Worse than quadratic
- PU200 is far off the chart.
- Memory consumption not necessarily an issue

There exists specific issues to keep in mind when applying machine learning for tracking.

Particle tracking is an active field in data science
Making a track is called pattern recognition
Tracking data is much sparser than regular images
Tracking device may have up to 10M of channels
Underlying complex geometry of sensors
Unstable detector geometry ; alignment
Not the regular type of sequences
Defining an adequate cost function
A solution must be performant during inference

Applications of ML in Tracking

Where ML Can Fit

Several Times

- **Hits preparation**
- **Seeding**
- **Pattern recognition**
- **Track fitting**
- **Track cleaning**

Where ML Can Fit

Several Times

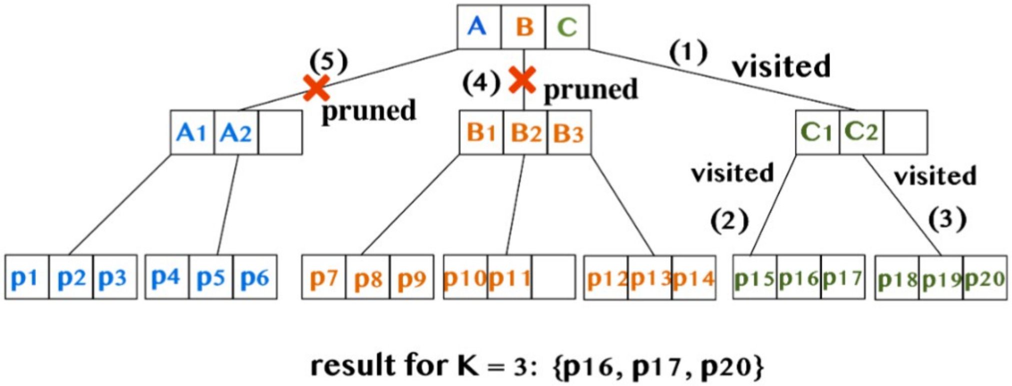
- Hits preparation here
- there Seeding there
- there Pattern recognition here
- there Track fitting here
- here Track cleaning

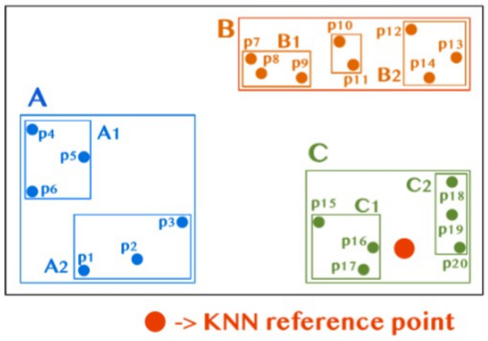
Machine Learning in Tracking

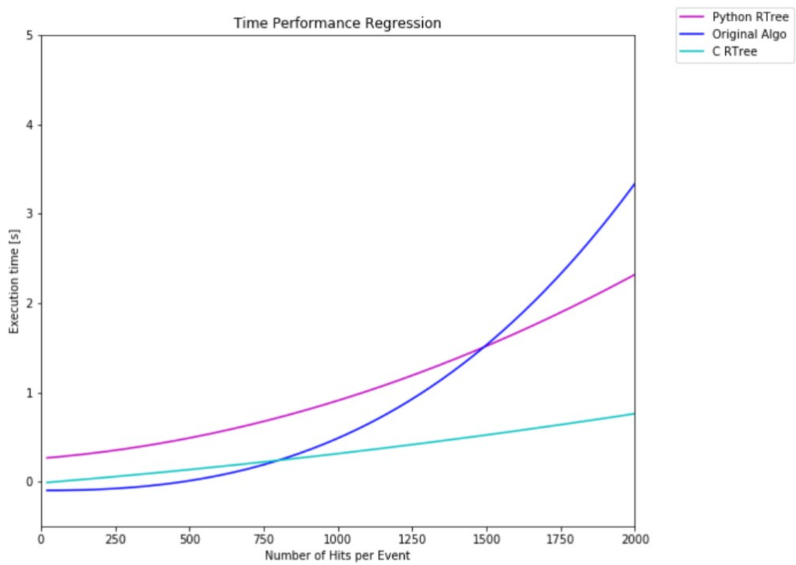
- Seeding and Clustering
- Pattern recognition
- Track Selection
- Track Parameters
- Vertexing

Seeds and Clusters

Hit Searching



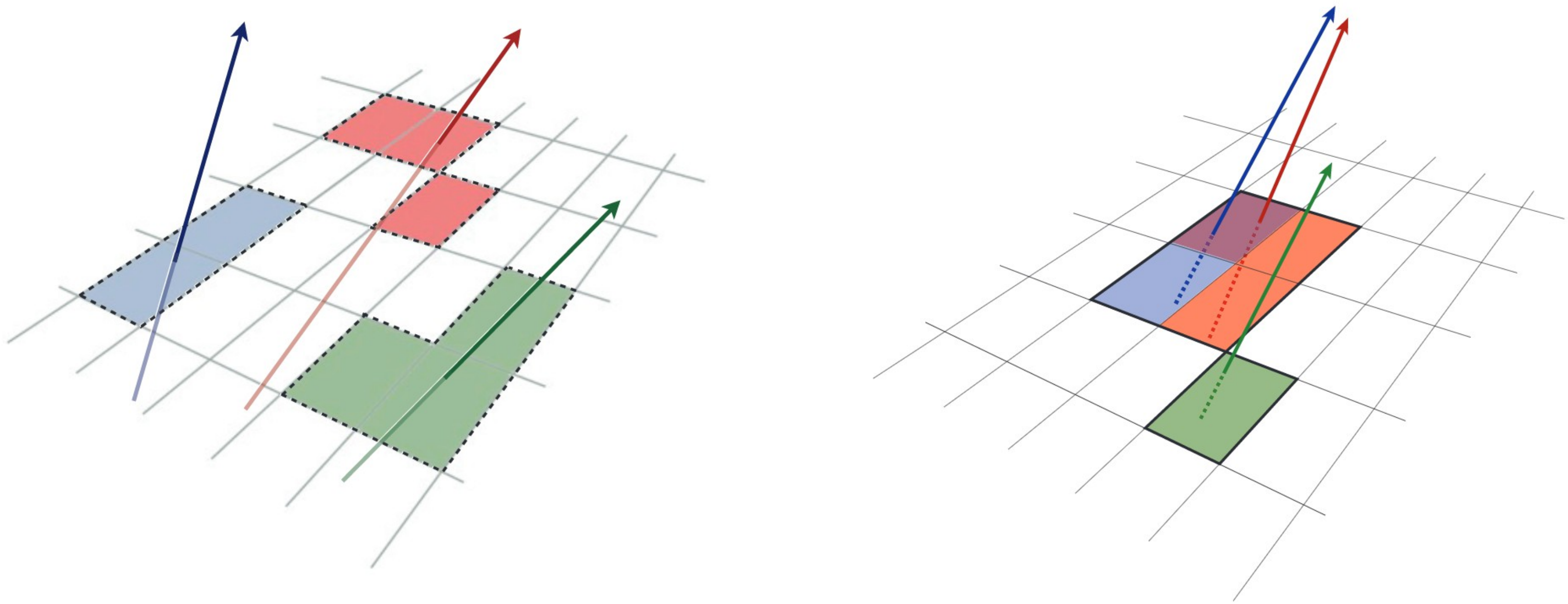




- Search for neighboring hits using R-tree
- Speedup over large event

https://indico.cern.ch/event/742793/contributions/3274363/

Tracking In Dense Environment



(a) Single-particle pixel clusters (b) Merged pixel cluster

Converging tracks are likely in boosted jets and jets dense of charged particles.
Degraded performance

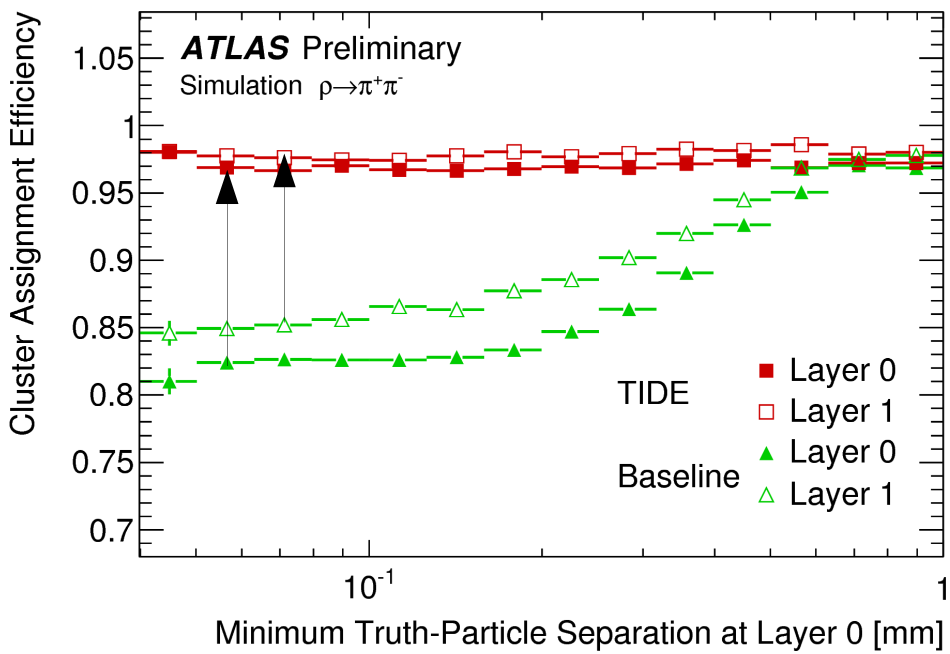
https://arxiv.org/abs/1704.07983
https://link.springer.com/article/10.1140/epjc/s10052-017-5225-7

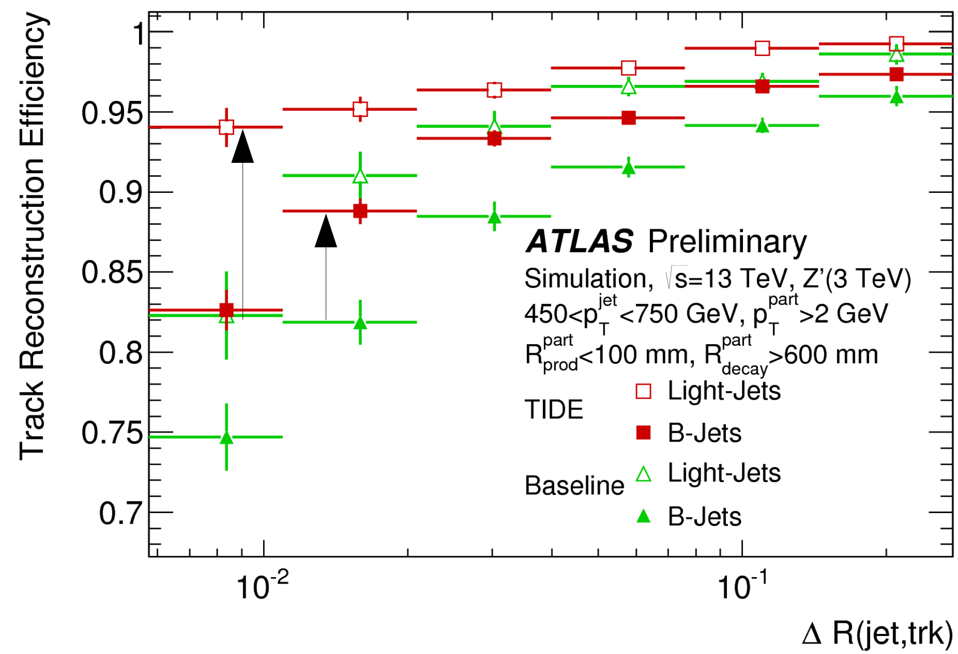
Cluster Splitting

Feed forward NN in three stages

- Determines the category 1-track, 2-tracks, 3-tracks
- Determines the n-crossing positions regression
- Determines the uncertainties as a multi-bin categorization

2 hidden layers fully connected NN with batch norm

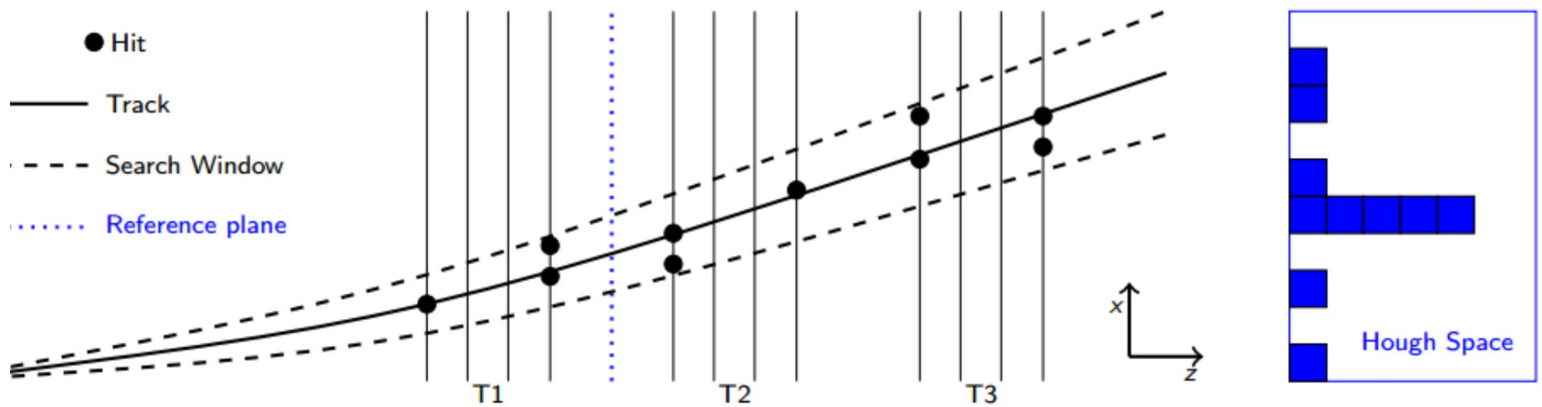




ATL-PHYS-PUB-2015-006

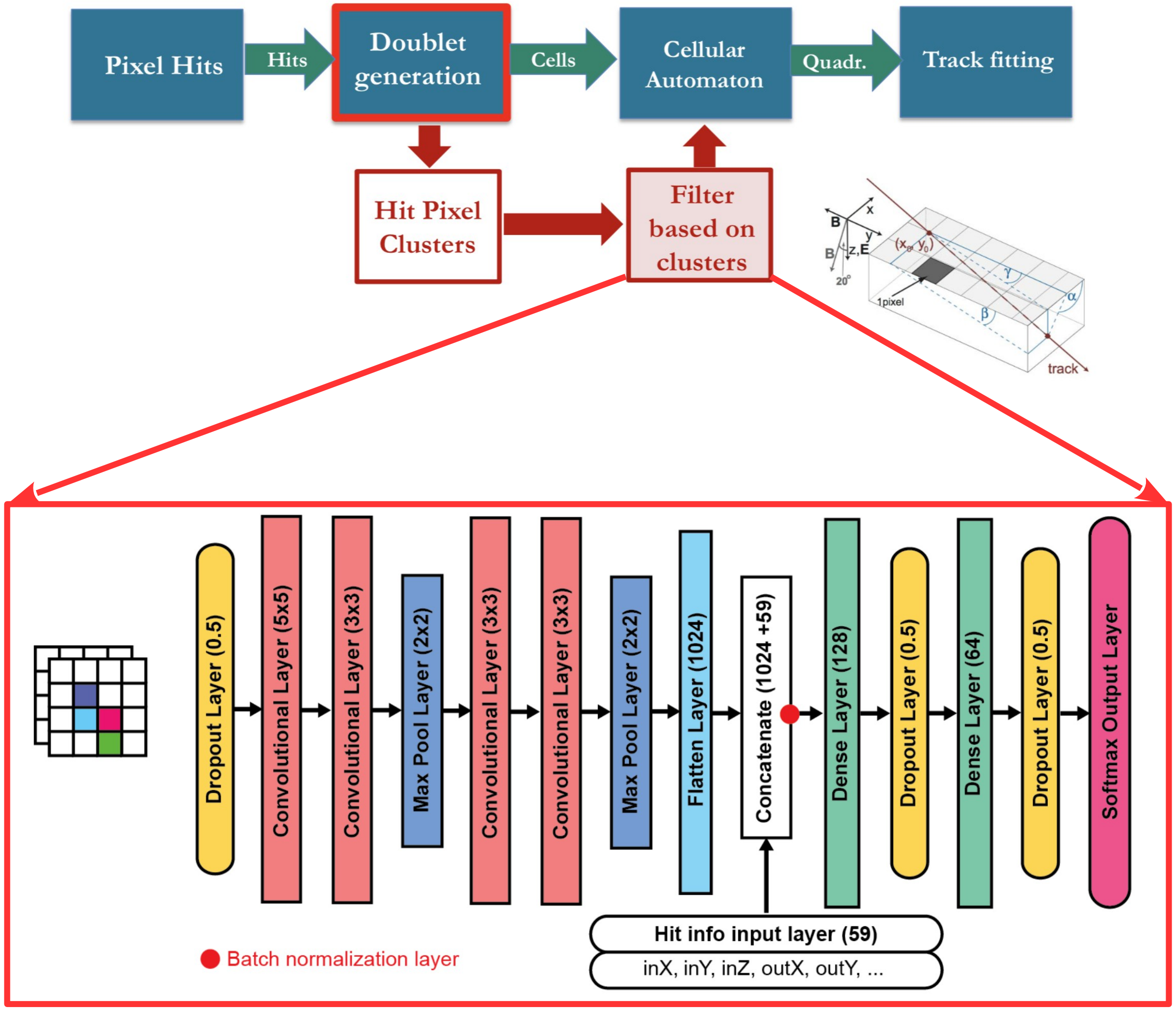
https://link.springer.com/article/10.1140/epjc/s10052-017-5225-7

Seed and Cluster Filtering

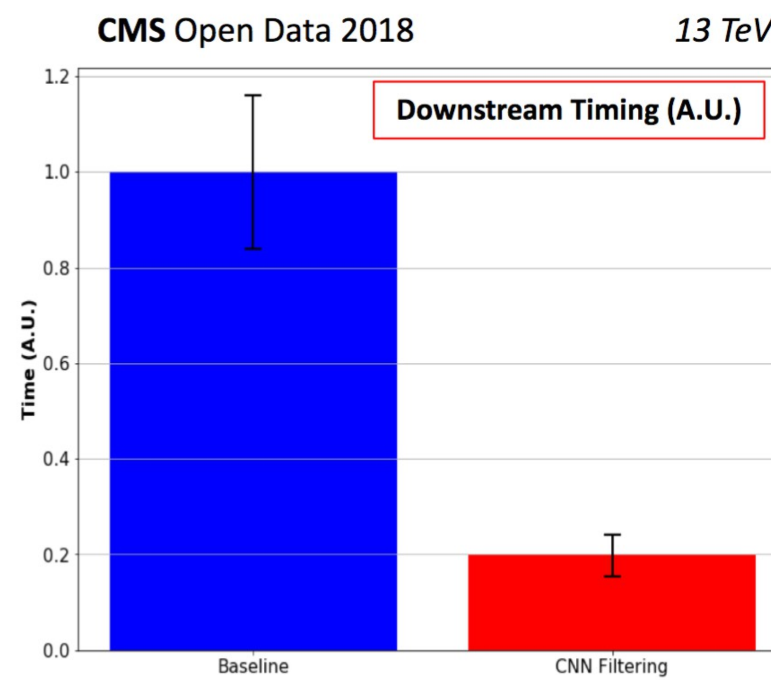


- NN classifier to distinguish good and bad clusters in the hough space during forward tracking
- classifier to distinguish good and bad T-seed (Use of the bonsai BDT https://arxiv.org/abs/1210.6861) during downstream tracking

Seed Cleaning

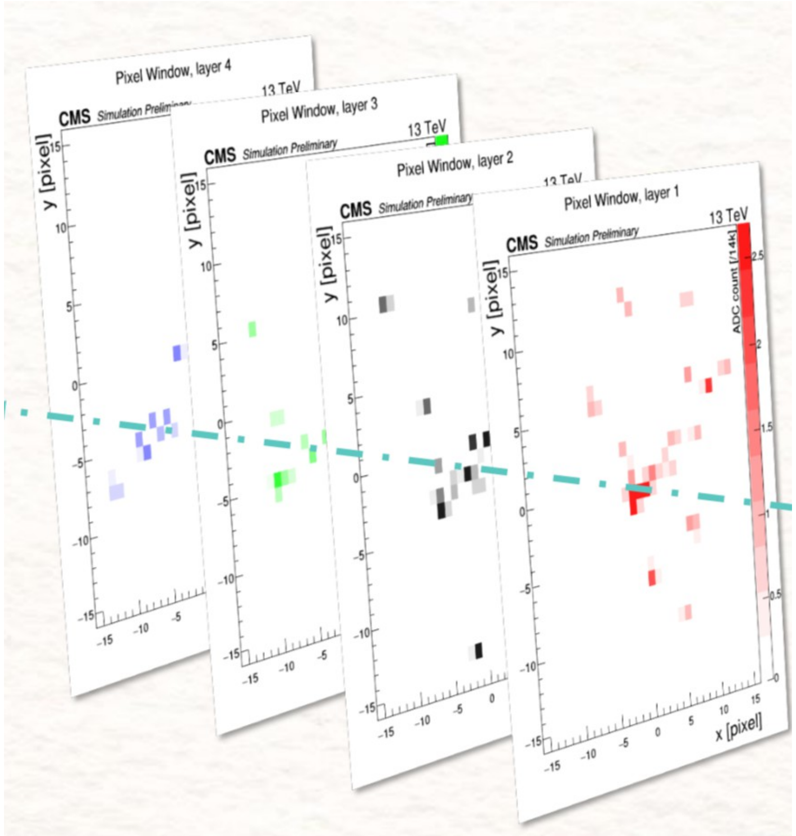


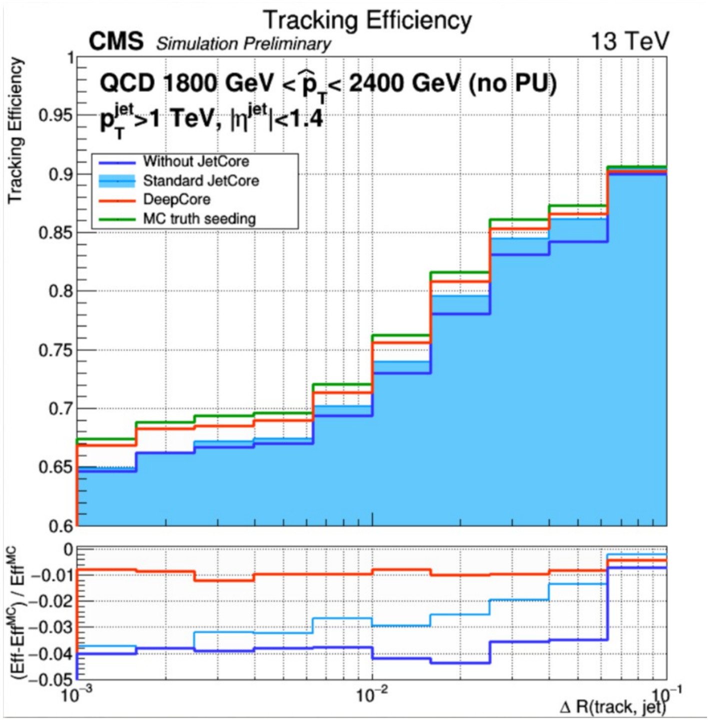
- Categorization of hits doublet using the pixel cluster shapes as input
- Significantly reduce timing in pattern recognition



https://indico.cern.ch/event/742793/contributions/3298727

Seed Finding in Jets



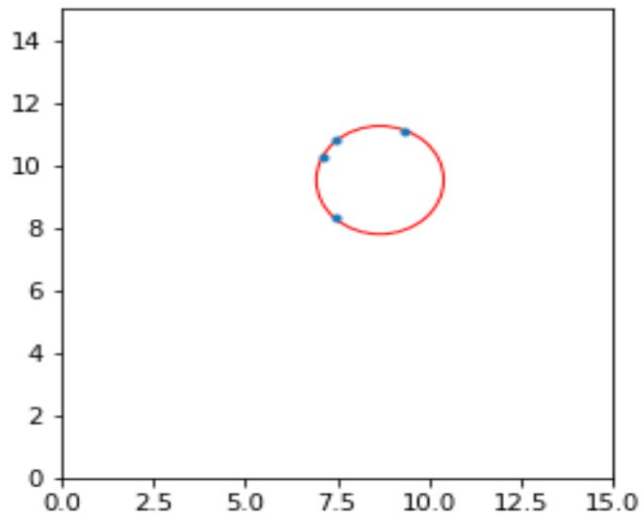


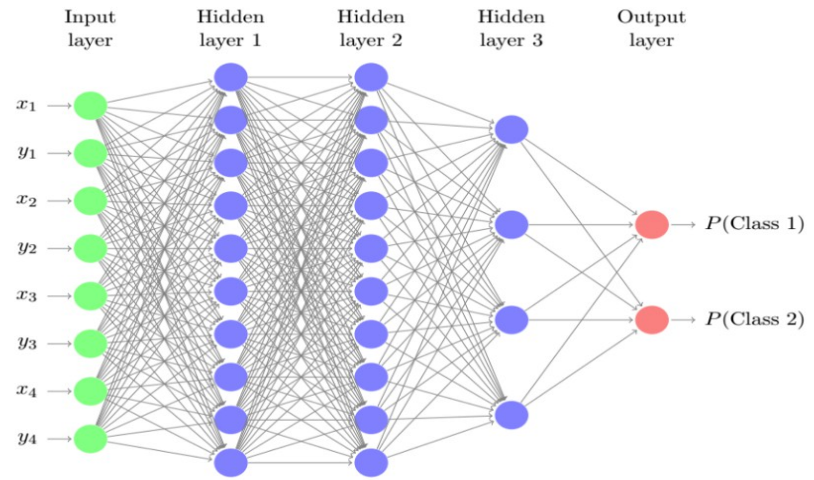
- Predict tracklets parameters from raw pixels using CNN
- Approaching the maximum performance

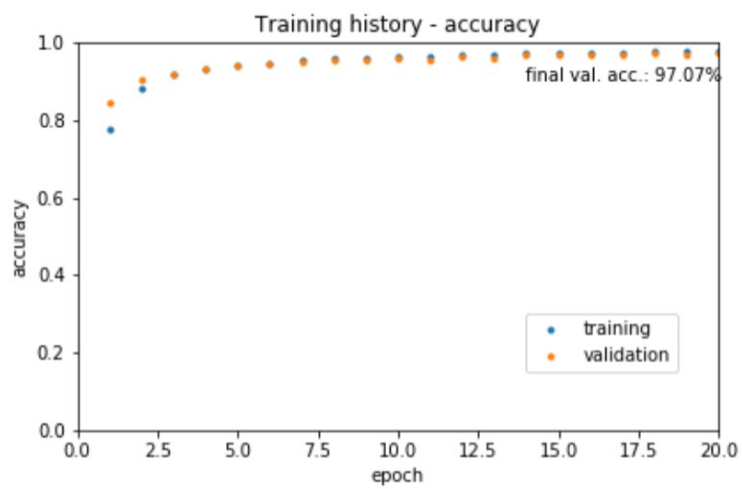
https://indico.cern.ch/event/742793/contributions/3274301/

Helix Testing

R&D







- Classify hit 4-tuple to be on a seed
- Promising at classifying goodness of seeds

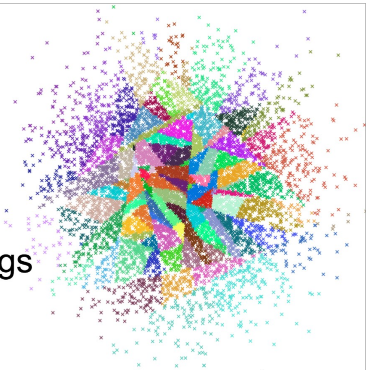
https://indico.cern.ch/event/742793/contributions/3274402

Bucketing

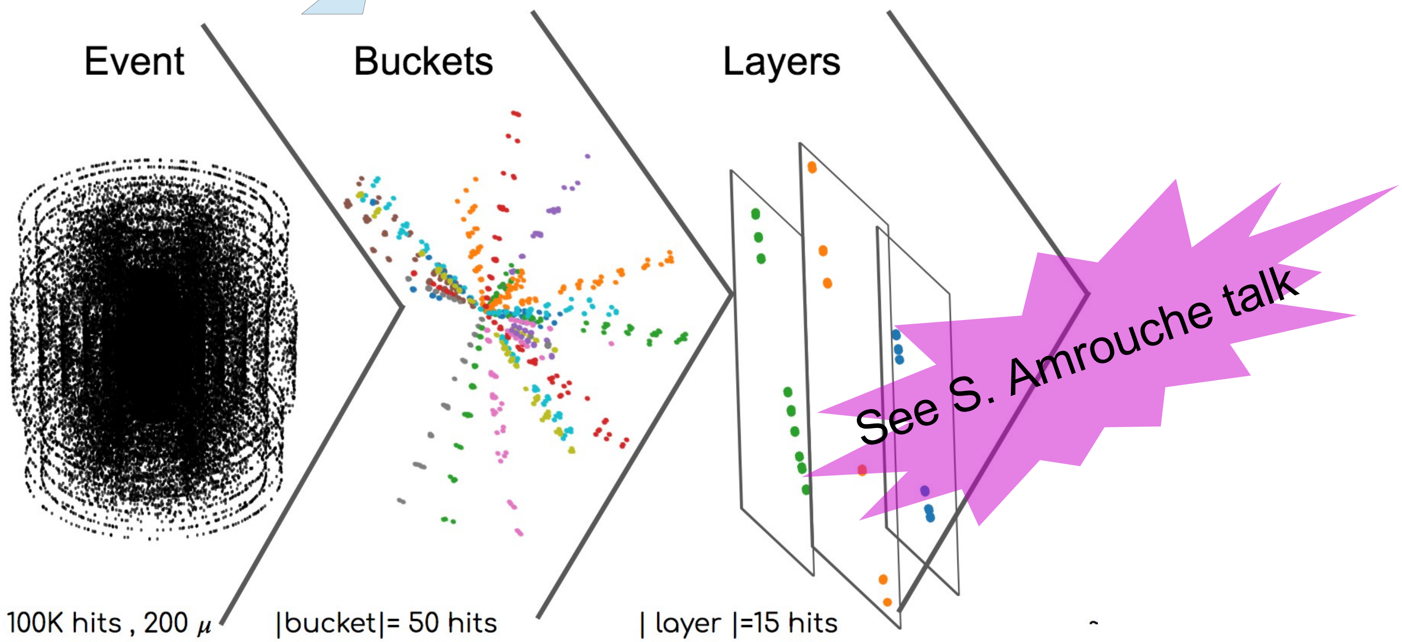
R&D

Spotify / **annoy**

- "Many millions of songs"
- **< 0.1ms** to get **n** similar songs
  [high-dimensional space]
- Unsupervised



- Method inspired from industry tools
- Reduced problem density in buckets of hits



https://indico.cern.ch/event/742793/contributions/3274332/

Track Finding

Non Parametric Functional Kernels

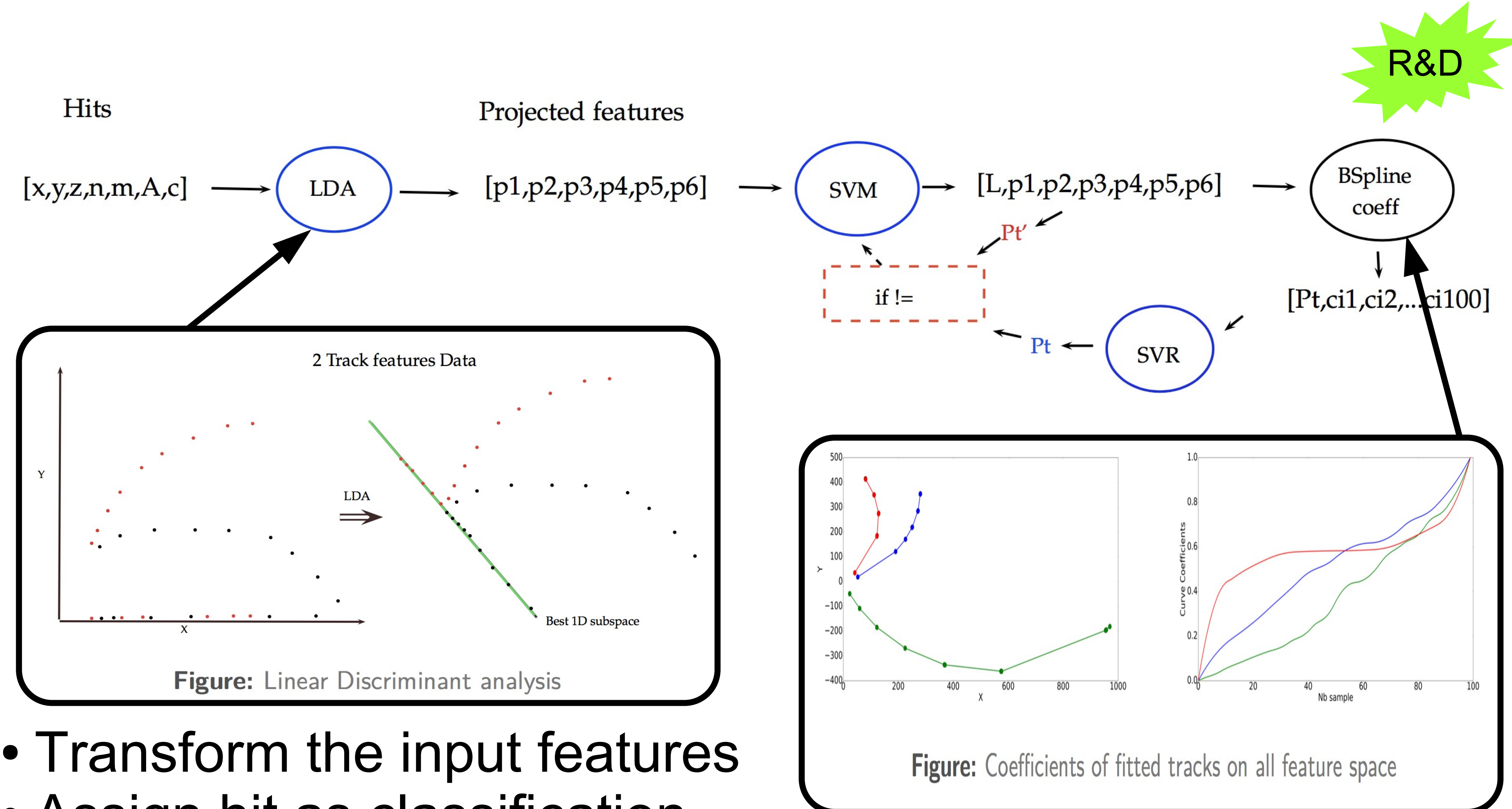


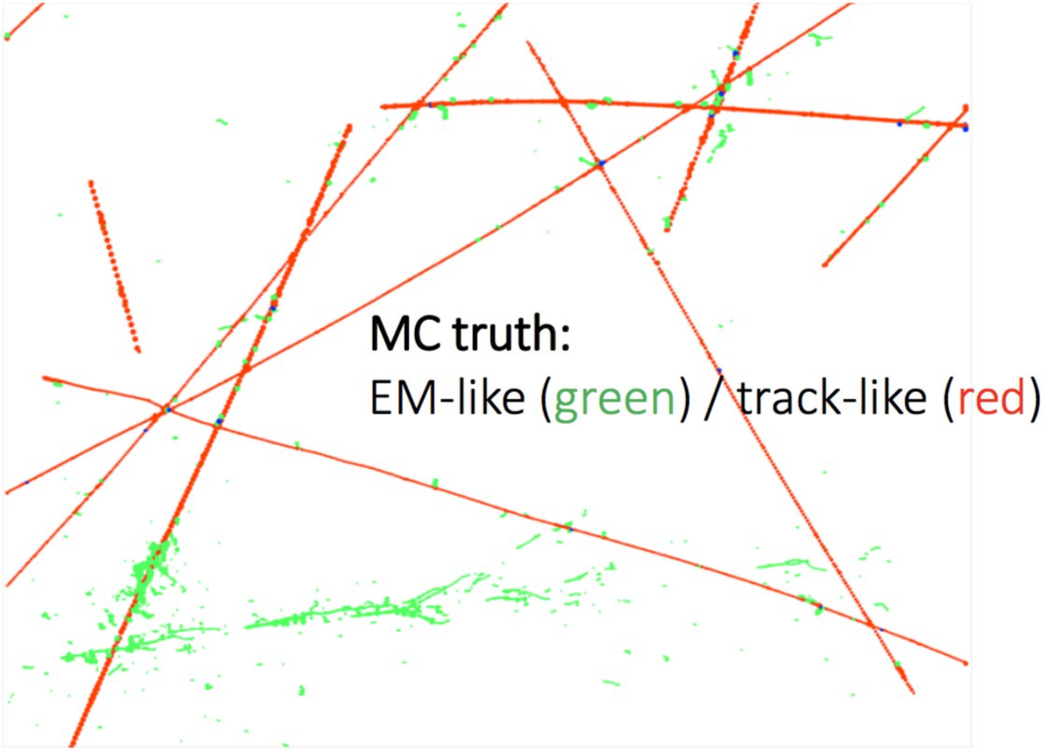
Figure: Linear Discriminant analysis

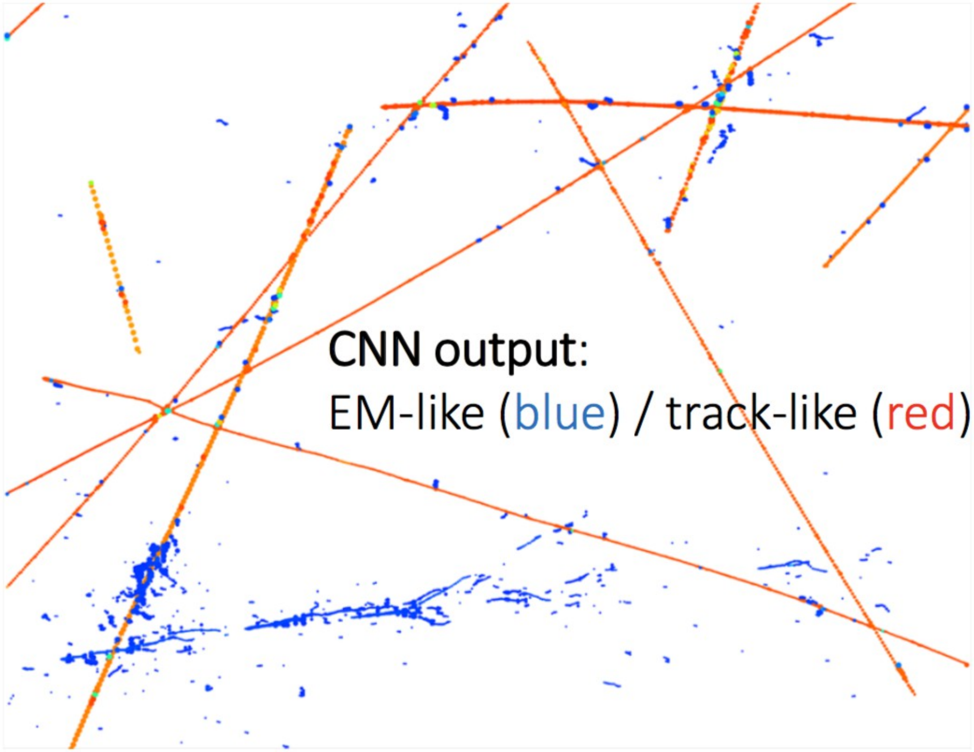
Figure: Coefficients of fitted tracks on all feature space

- Transform the input features
- Assign hit as classification task with SVM

https://indico.cern.ch/event/577003/contributions/2444883/

TPC Activity Segmentation





- Challenge to code explicitly
- Almost text-book example of de-noising AE
- Achieved with CNN

TrackML Challenge

Accuracy Phase

- **First :** ***Top Quarks***
  - Johan Sokrates is an industrial Mathematics master student
  - Pair seeding**,** triplet extension**,** trajectory following**,** track cleaning**,** all **with machine learning for quality selection**
- **Second :**
  - Pei-Lien Chou is a software engineer in image-based deep learning in Taïwan
  - Machine learning to **predict the adjacency matrix**
- **Third :**
  - Sergey Gorbunov is a p**hysicist, expert in tracking**
  - Triplet seeding, trajectory following

- **Jury "Innovative prize"**
  - Yuval Reina is an electronic engineer and Trian Xylouris is an entrepreneur
  - Marginalized Hough transform with **machine learning classifier**
- **Jury "Clustering prize"**
  - Jean-François Puget CPMP is a software engineer at IBM. He is both competition and discussion Kaggle grandmaster
  - **DBSCAN clustering** with iterative Hough transform
- **Jury "Deep Learning prize"**
  - Nicole and Liam Finnie are software engineers
  - DBSCAN seeding, **trajectory following with LSTM**

- **Organization pick**
  - Diogo R. Ferreira is a professor/researcher, focusing on data science and nuclear fusion
  - **Pattern matching**

https://arxiv.org/abs/1904.06778

Throughput Phase

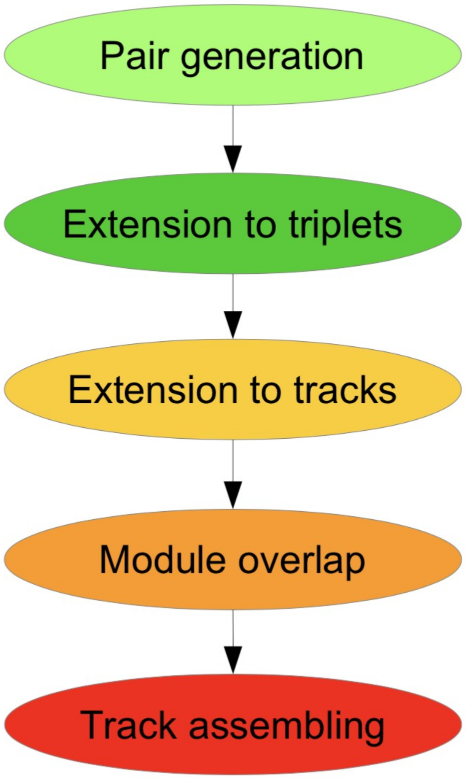
- **First** :
  - Sergey Gorbunov is a physicist, expert in tracking
  - Triplet seeding, multiple passes trajectory following
- **Second** :
  - Dmitry Emeliyanov is a physicist
  - Connection graph, Cellular automaton, graph traversal with Kalman Filter
- **Third** :
  - Marcel Kunze is a computer scientist
  - Solution based on top quark, trained navigation on DAG of voxels to find doublets and triplets

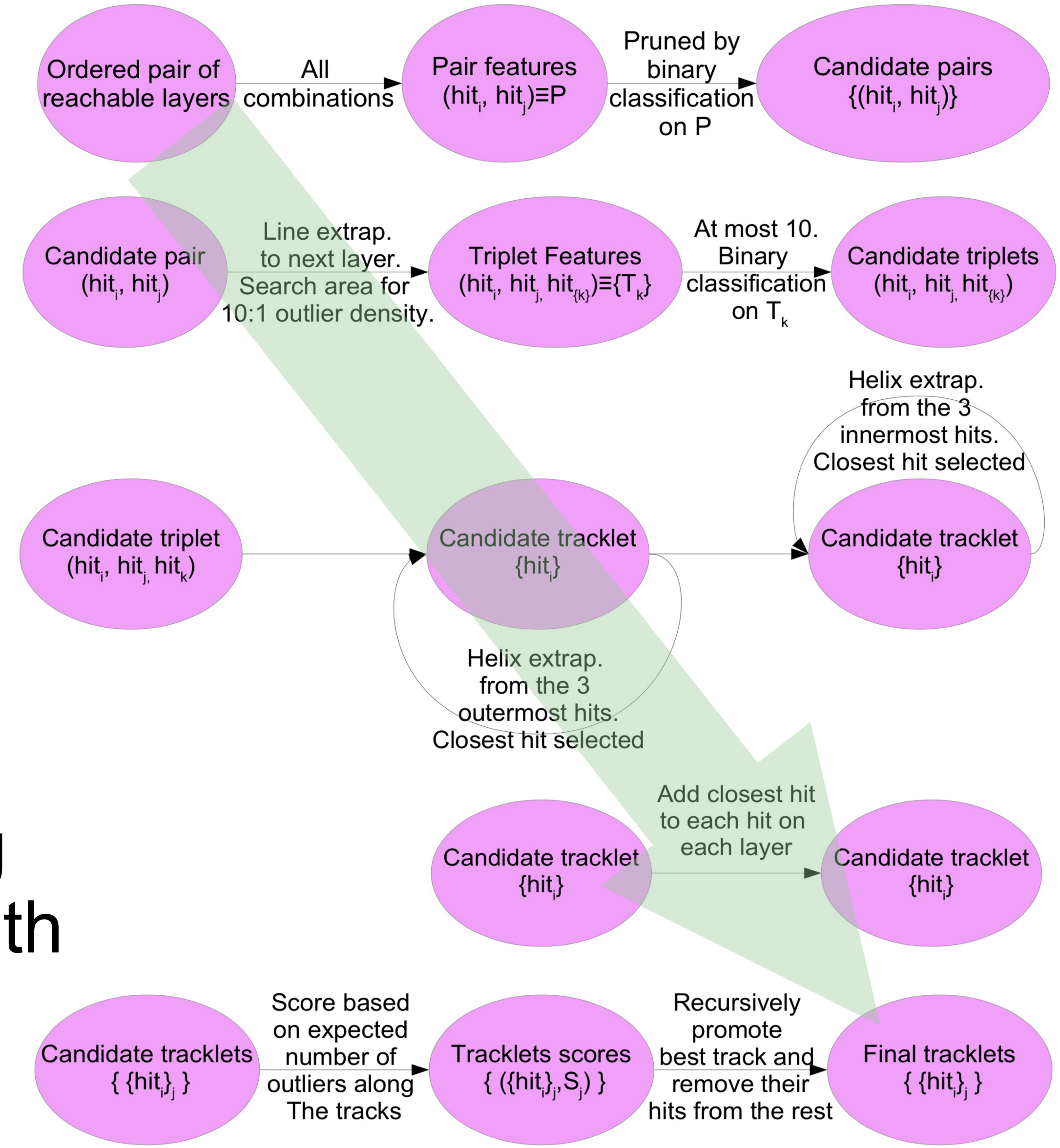
https://indico.cern.ch/event/813759/

Sponsors

See M. Kiehn talk

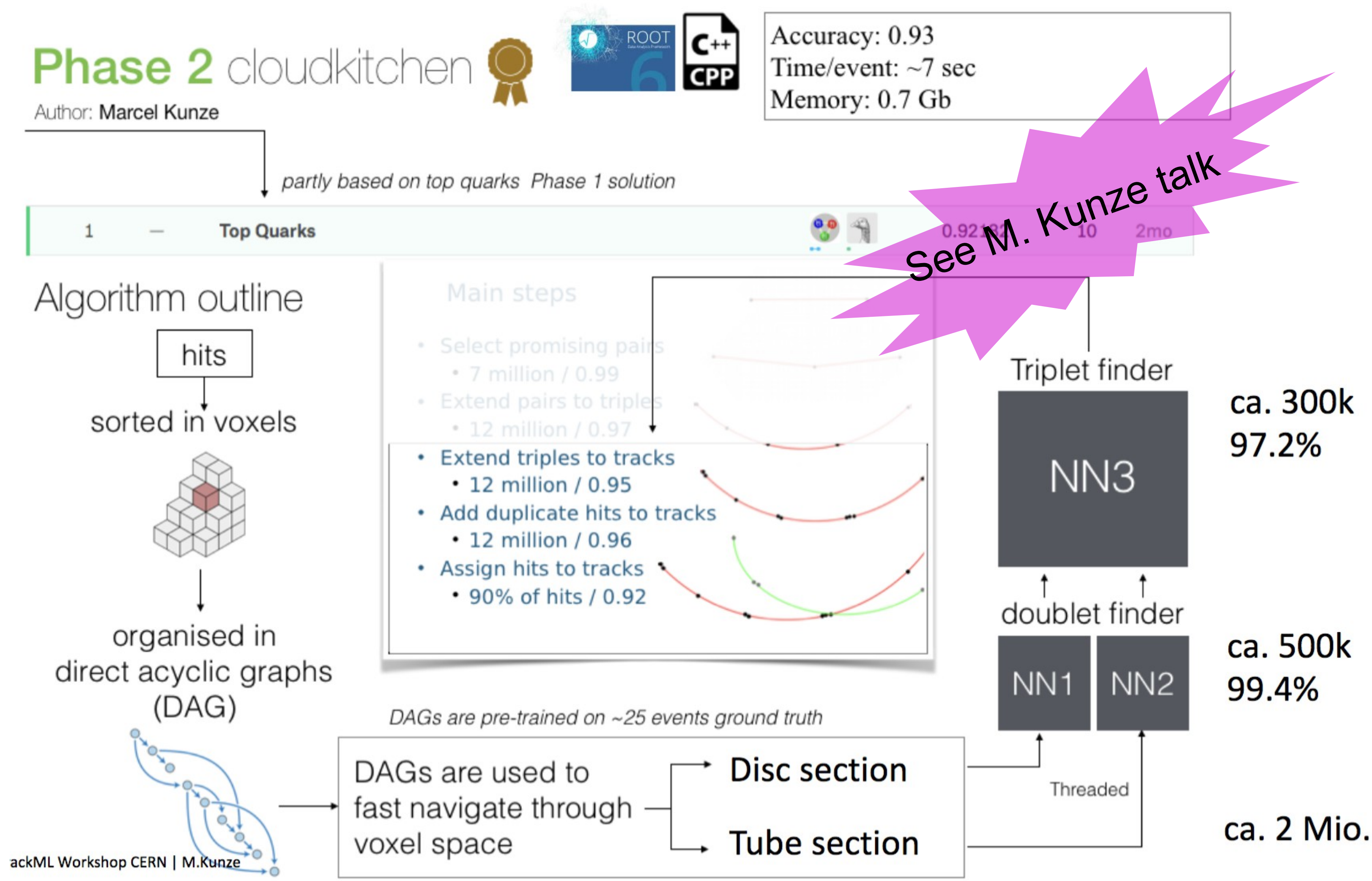
Controlled Track Following





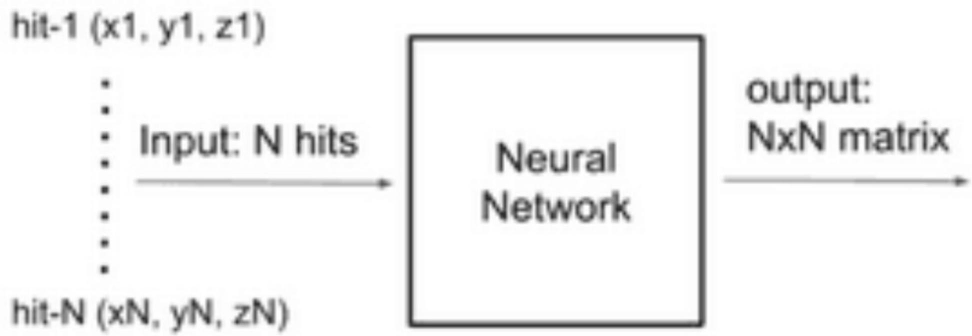
- Standard track following algorithm augmented with classifications

DAG Track Following

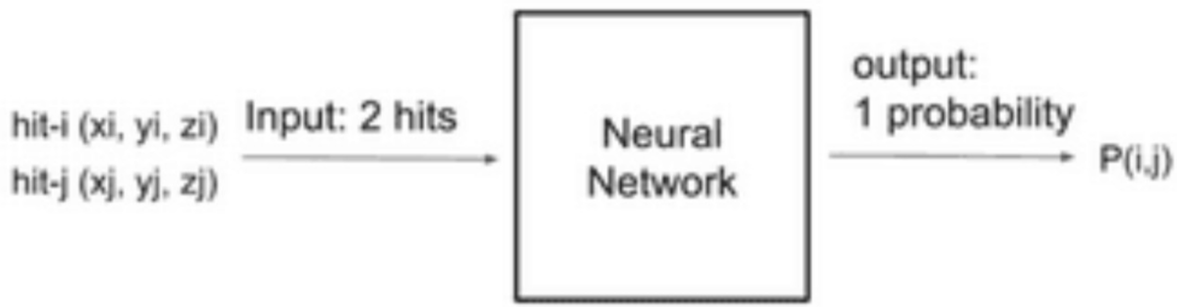


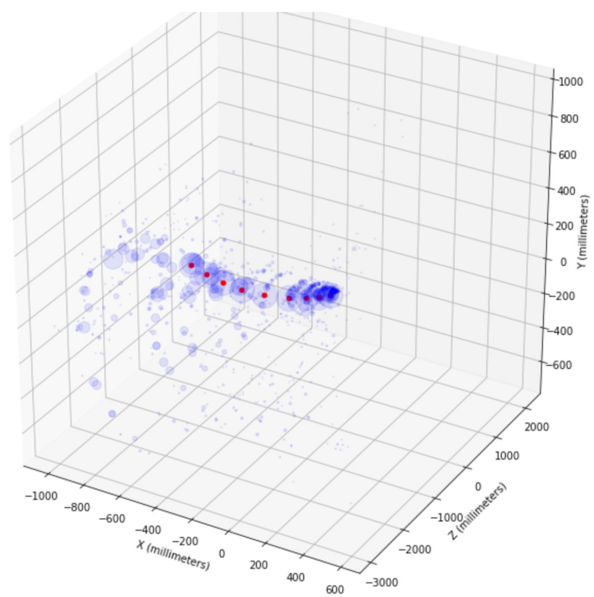
- Improvement over accuracy phase winner
- Hit navigation using direct acyclic graph (DAG)

Deep Hit Adjacency



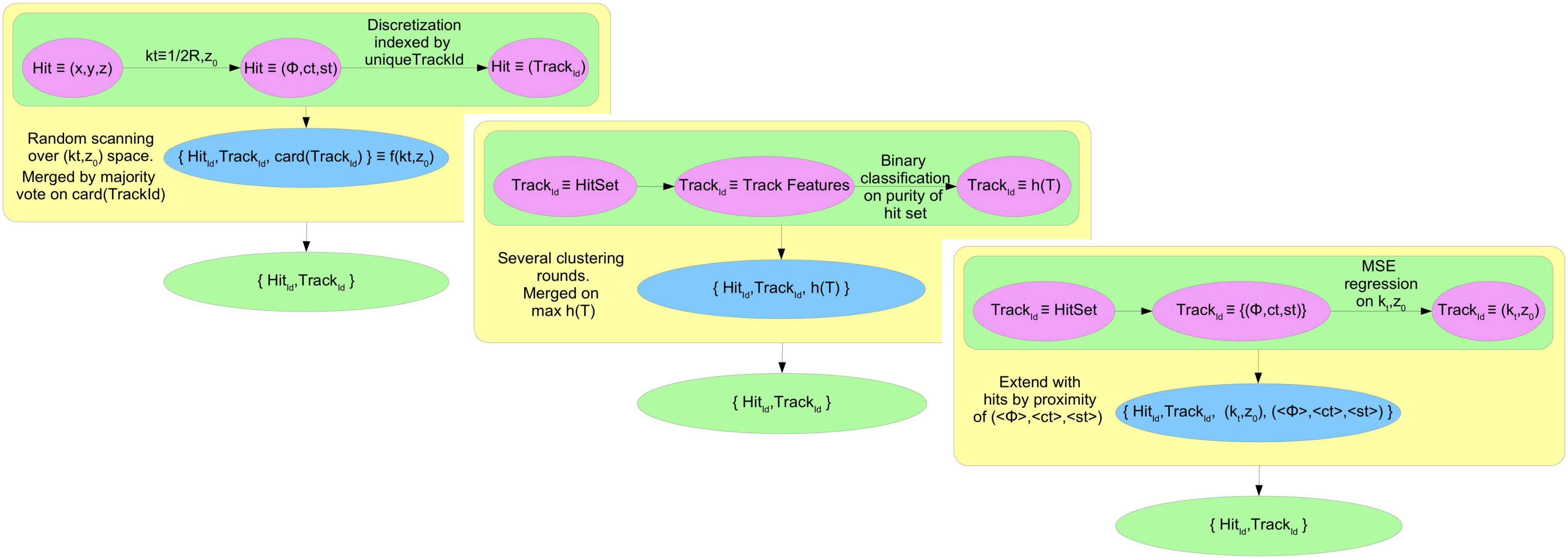
	1	*2*	*3*	*4*	*5*	~	*N*
1	0	1	0	0	1	~	1
2	1	0	0	0	1	~	1
3	0	0	0	1	0	~	0
4	0	0	1	0	0	~	0
5	1	1	0	0	0	~	1
~	~	~	~	~	~	~	~
N	1	1	0	0	1	~	0





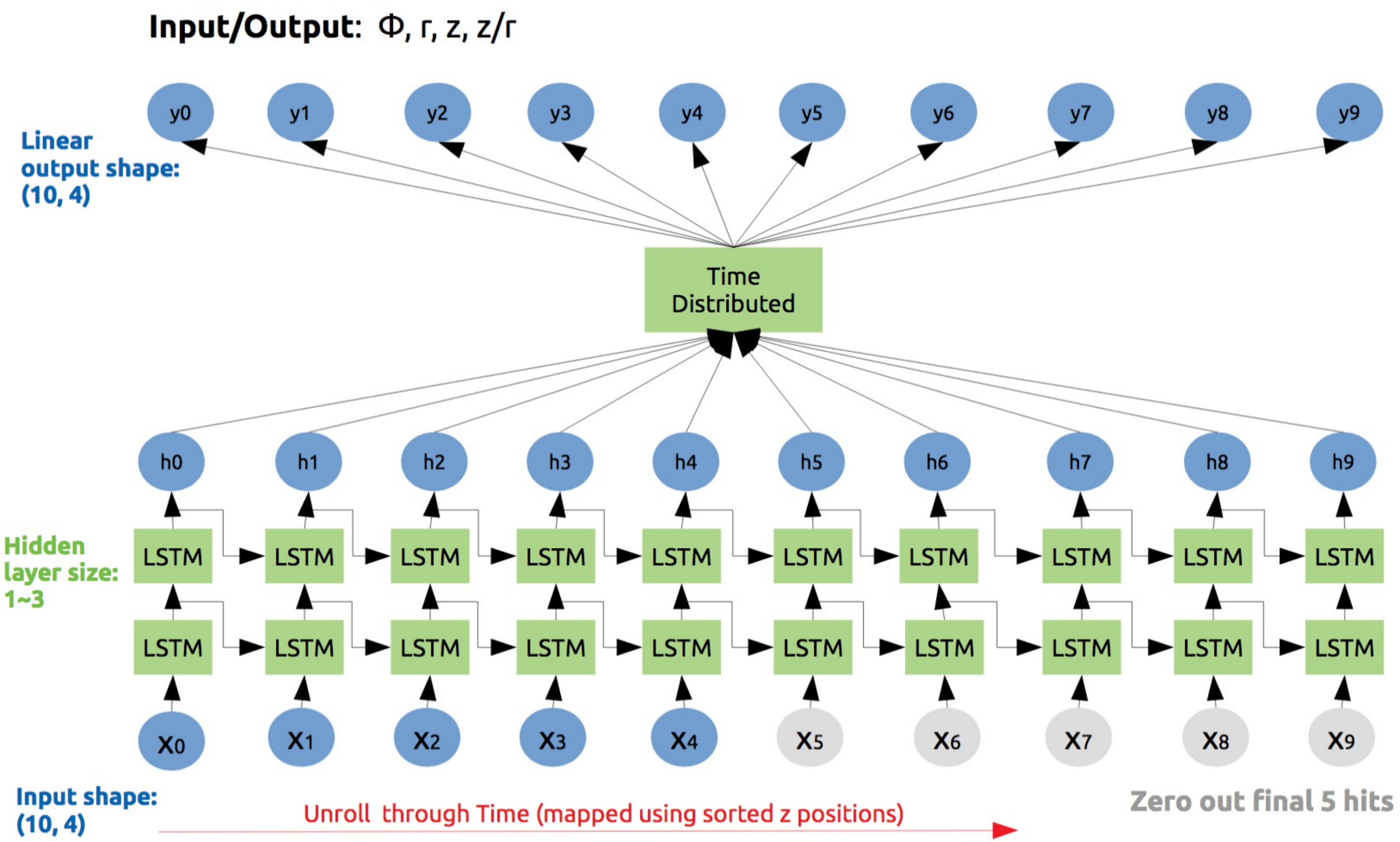
- Deep-learn the full NxN adjacency matrix
- Track following combinatorics
- Impractical computation-wise

Improved Hough Transform



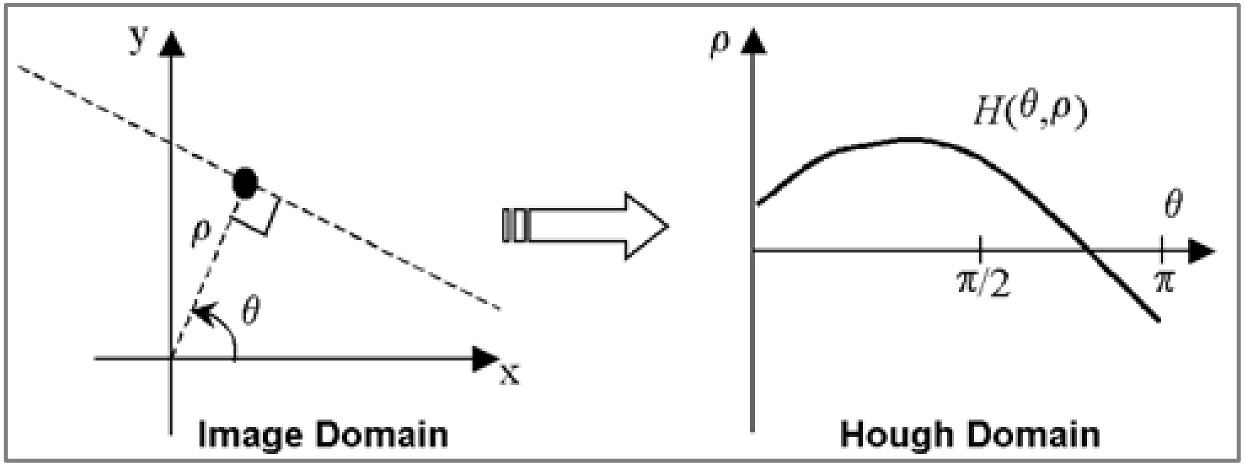
- 5D hough transform made computationally tractable by marginalizing p_T and z_0
- Track extension in track feature's space

LSTM Track Following



- Rely on existing seeding
- Follow tracklets with LSTM for predicting the hit positions

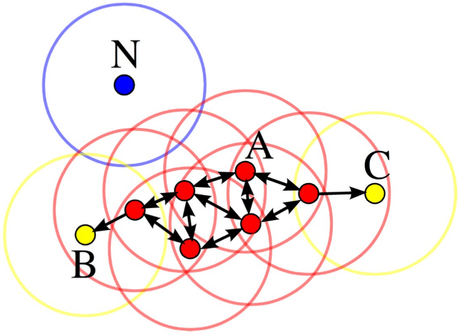
DBSCAN – Hough Transform



DBSCAN?

Density-based clustering

- Few parameters: distance, min #, (metric)
- Simple and available
- Used in starting kit score ≈ 0.2



wikipedia.org/wiki/DBSCAN

- Iterative hough transform using DBSCAN for unbinned clustering in track feature space

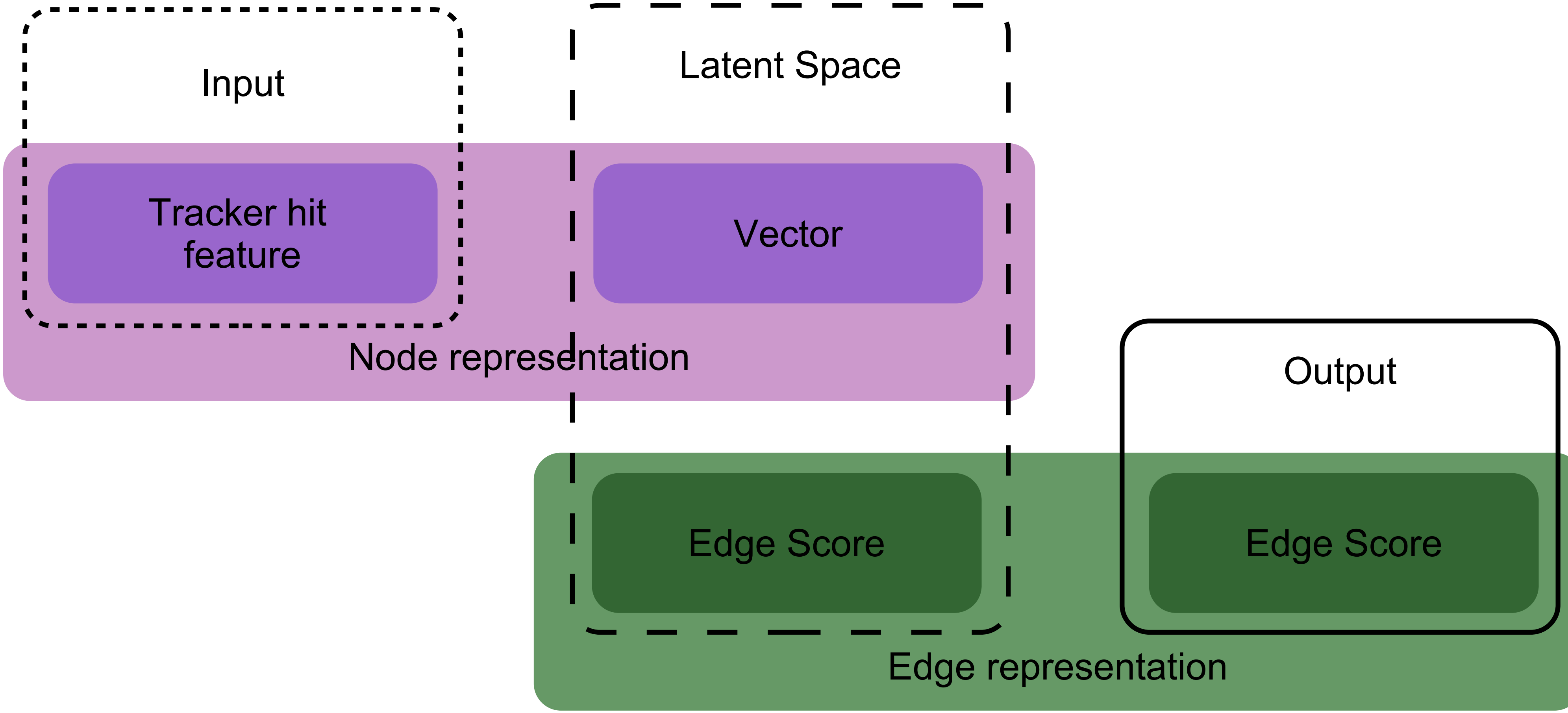
R&D

Edge Classification with Graph Neural Network

https://heptrkx.github.io/

digression

Node & Edge Representations



Latent edge representation taken to be the classification score instead of some latent vector representation

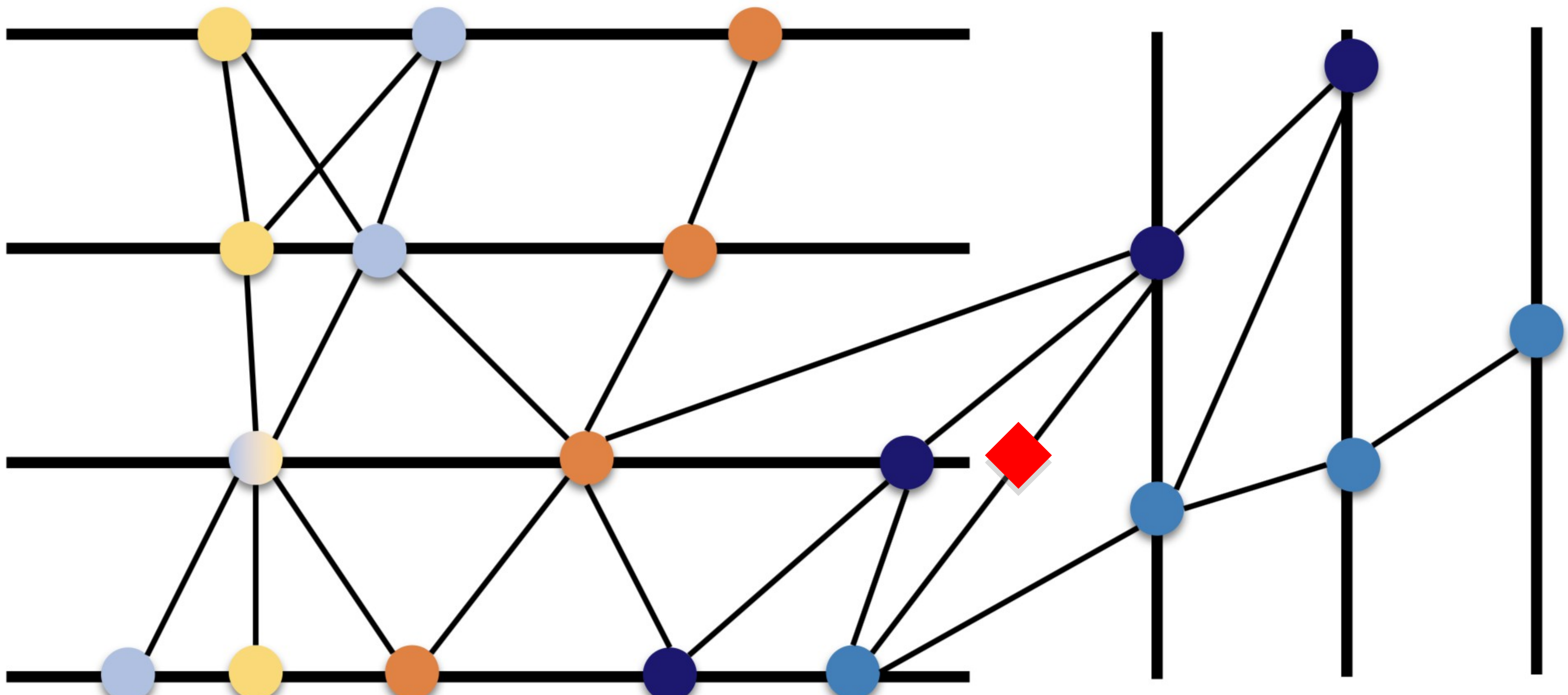
Neural Networks

- **Input Network**
  - ➢ Transforms from hit features (r,φ , z) to the node latent representation (N for 8 to 128)
  - ⬩ Dense : 3→...→N

- **Edge Network**
  - ➢ Predicts an edge weight from the node latent representation at both ends
  - ⬩ Dense : N+N→...→1

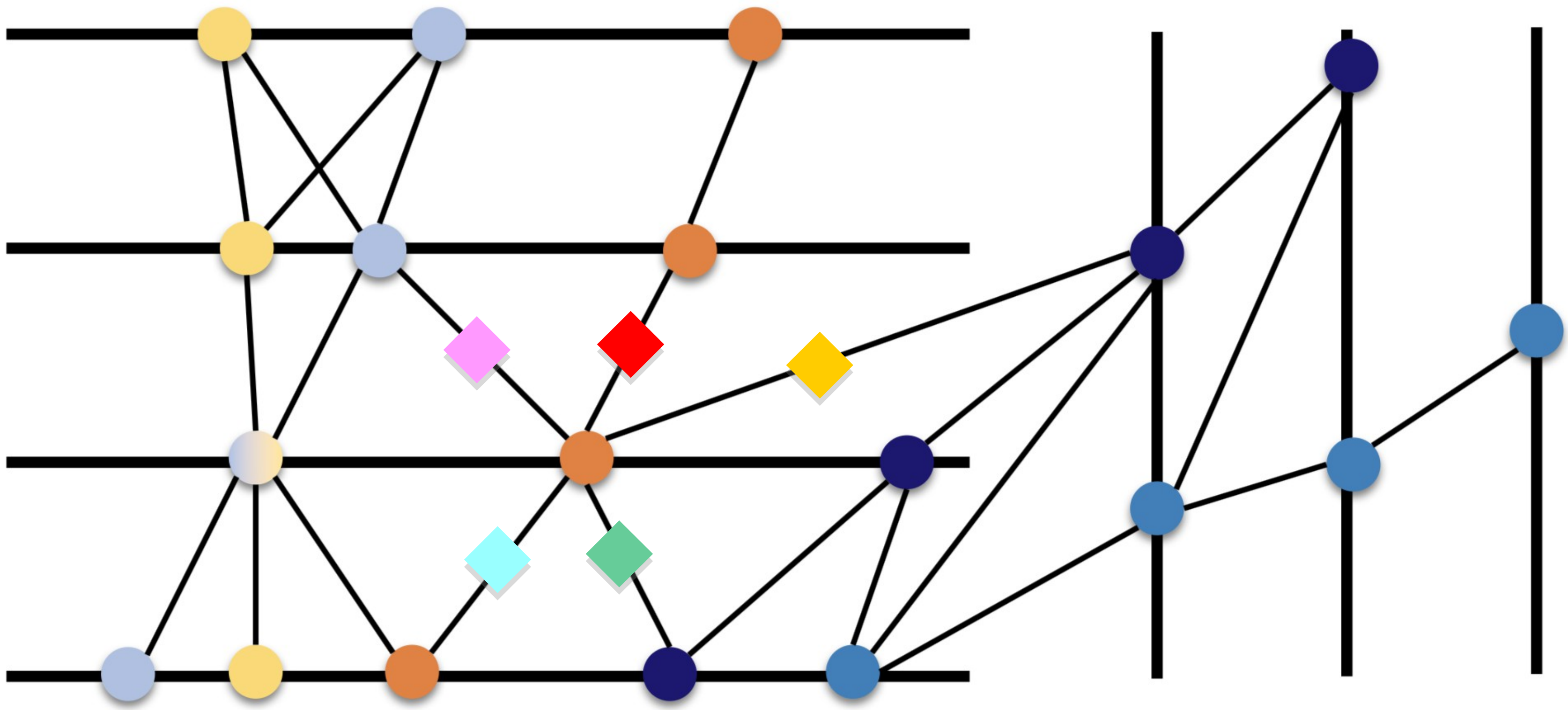
- **Node Network**
  - ➢ Predicts a node latent representation from the current node representation, weighted sum of node latent representation from incoming edge, and weighted sum
  - ⬩ Dense : N+N+N→...→N

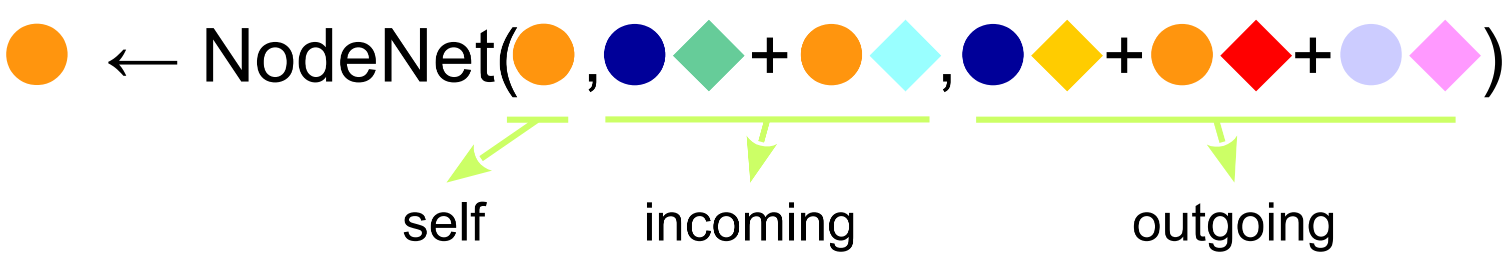
Edge Network



◆ ← EdgeNet(●,●)

Node Network



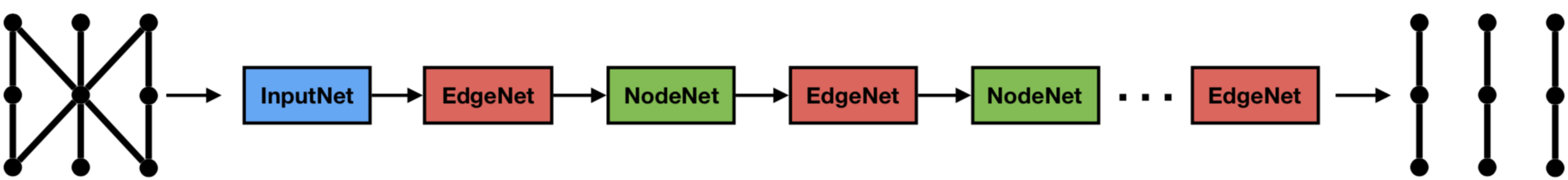


Information Flow

- Graph is sparsely connected from layer to layer
- InputNet + EdgeNet + NodeNet only correlates hits information on triplet of layers
  - The information from the outer hits and inner hits are not combined

- Several possible ways to operate the connection

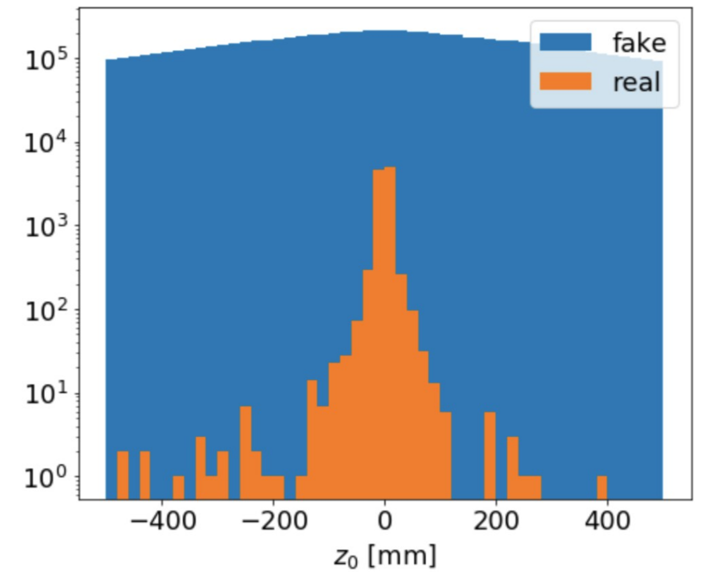
→ Correlates hits information through multiple iterations of (EdgeNet+NodeNet)

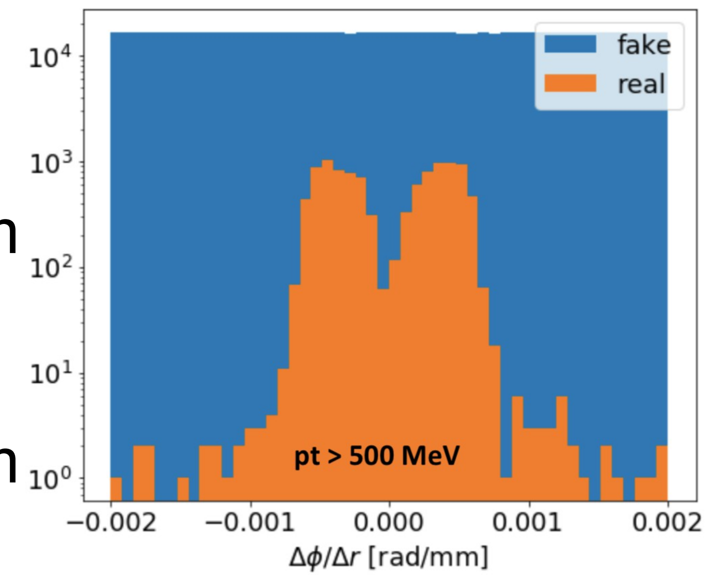


Downgraded Complexity

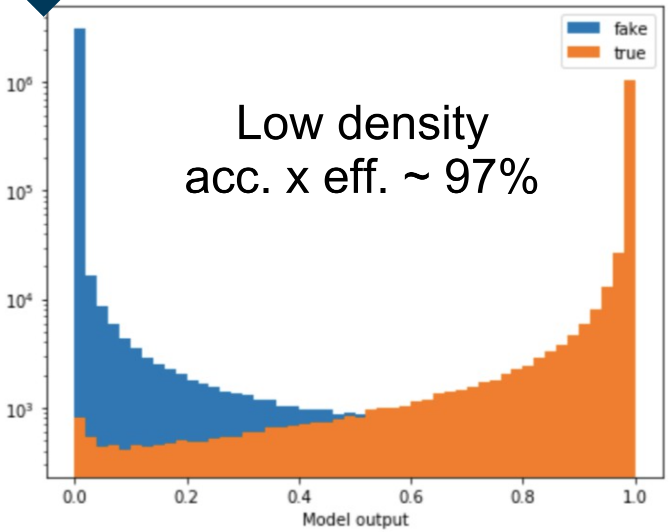
- TrackML dataset generated from ... with an average of 200 pileup events.
- Not computational possible at this time to embed the smallest relevant sector of full event on a graph

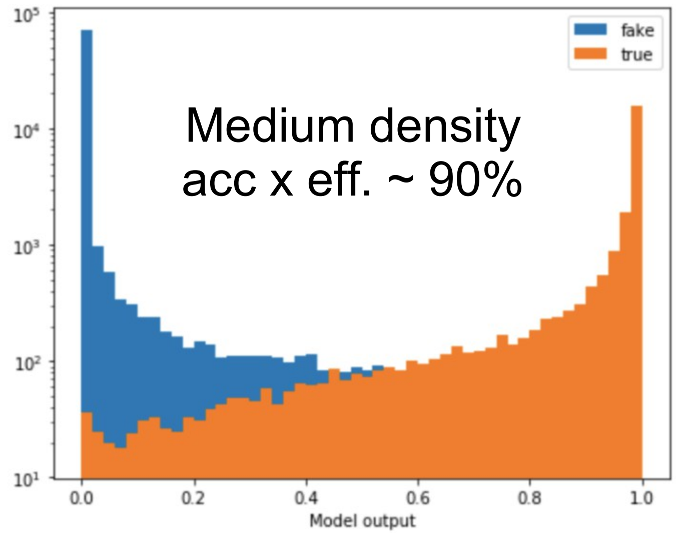
- Sub-dataset are constructed by
  - Low density
    - p_T>1 GeV, $\Delta\varphi$<0.001, Δz_0<200mm
    - acceptance: 99%, purity: 33%
  - Medium density
    - p_T>500 MeV, $\Delta\varphi$<0.0006, Δz_0<150mm
    - acceptance: 95%, purity: 25%
  - High density
    - p_T>100 MeV, $\Delta\varphi$<0.0006, Δz_0<100mm
    - acceptance: 43%, purity: 9%

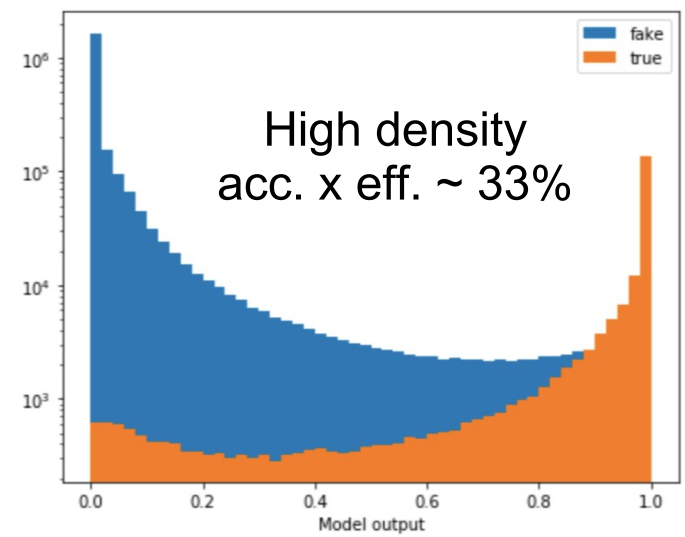




Performance





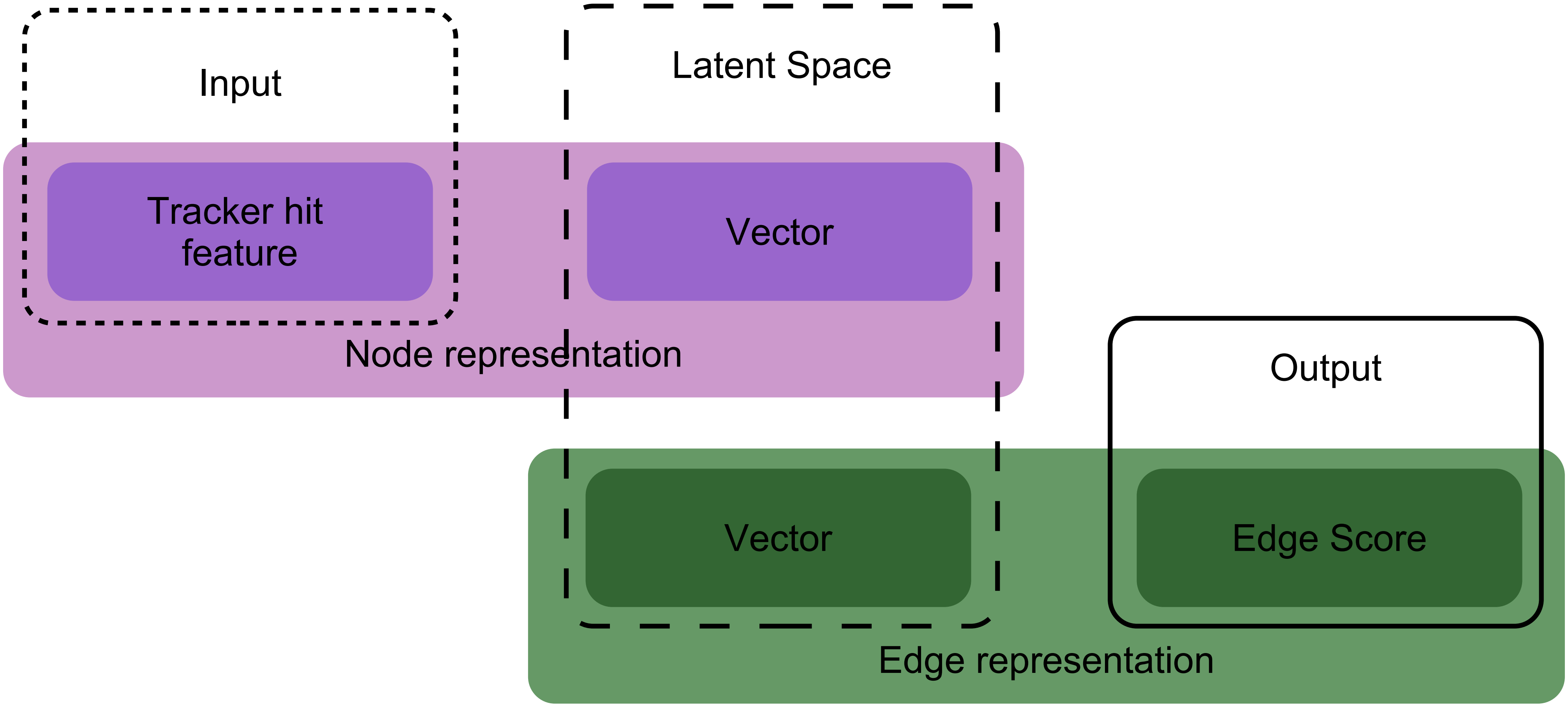


Dealing with Large Graphs

- Full event embedding
  - A graph with ~120k nodes (14.4B edges) and ~1M potential edges is a big graph

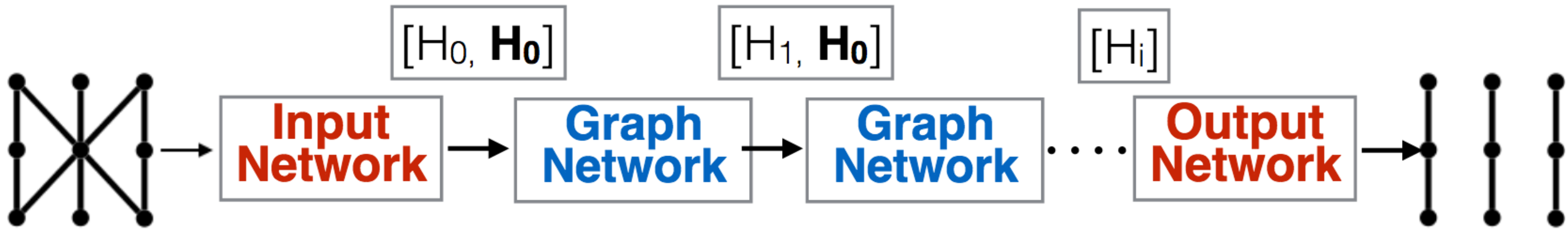
- Split the problem
  - currently using 16 sectors in φ
- Use sparse matrix implementation
  - https://github.com/deepmind/graph_nets for example
- Identify disjoint sub-graphs
  - Geometrical cuts, segment pre-classifier, ...
- Implement distributed learning of large graphs
  - Scope of the Exa.TrkX Project

Node & Edge Representations



Edge representation is not the edge score.
Final edge score extracted from the latent edge representation.

Message Passing Model

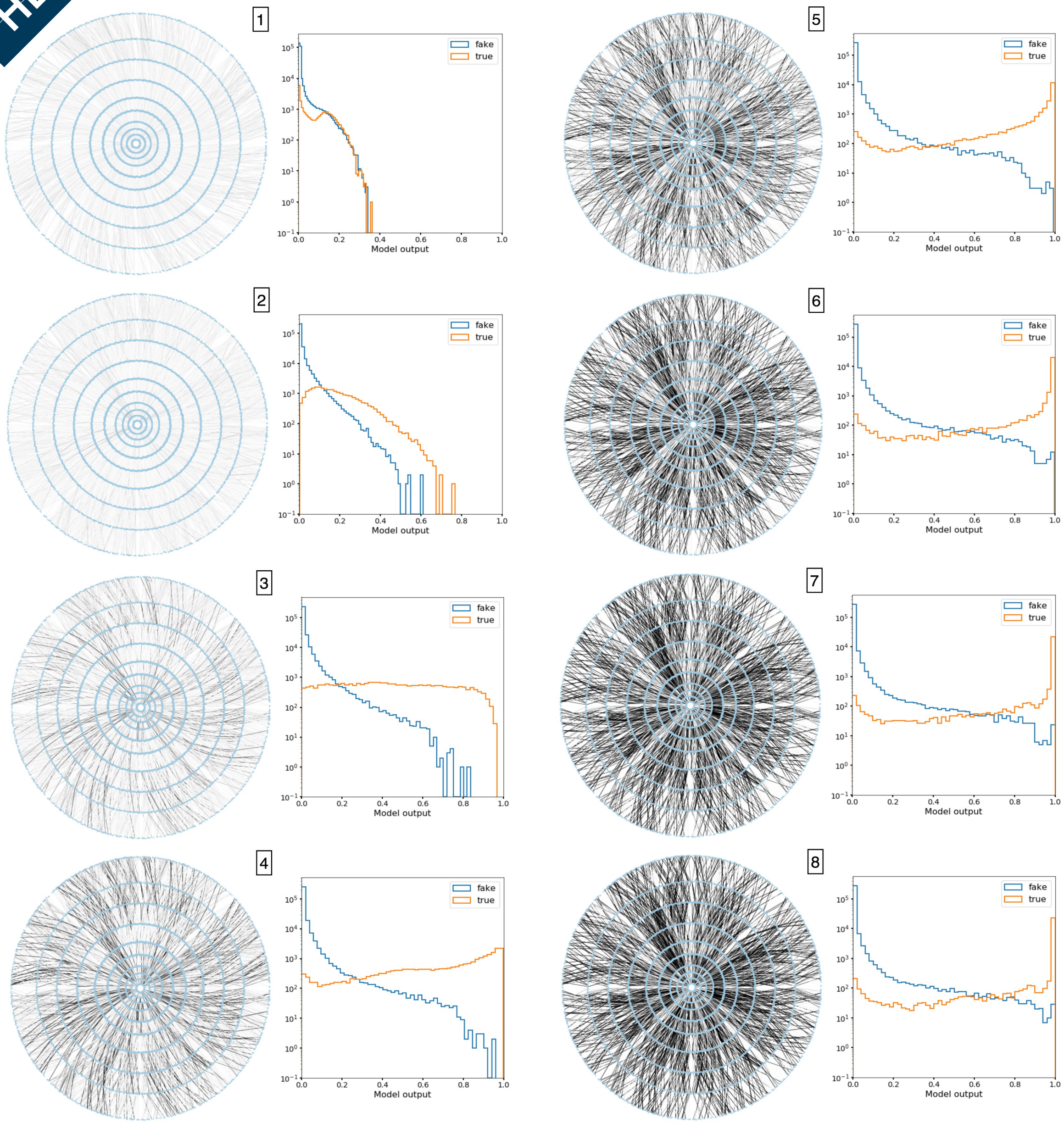


- Same graph connectivity
- No explicit attention mechanism
- Edge representation computed from end-nodes features
- Node representation computed from the sum over all connected edges

→ Correlates hits information through multiple (8) iterations of (Graph Network)

→ Uses https://github.com/deepmind/graph_nets TF library

Information Flow



- Checking edge score after each step of graph network.
- Effective output of the model is in step 8.
- Full track hit assignment learned in last stages of the model.
- Tracklets learned in intermediate stages.

Algorithm Efficiency

one-event	N-particles	ratio w.r.t Total	ratio w.r.t Reconstructable	relative ratio
Total	11170	100%		100%
Reconstructable	9635	86%	100%	86%
Barrel	7492	67%	78%	78%
No-missing hits	6600	59%	69%	88%
Edge selection	3114	28%	32%	47%
Split graph	2668	24%	28%	**86%**
GNN	2590	23%	27%	97%

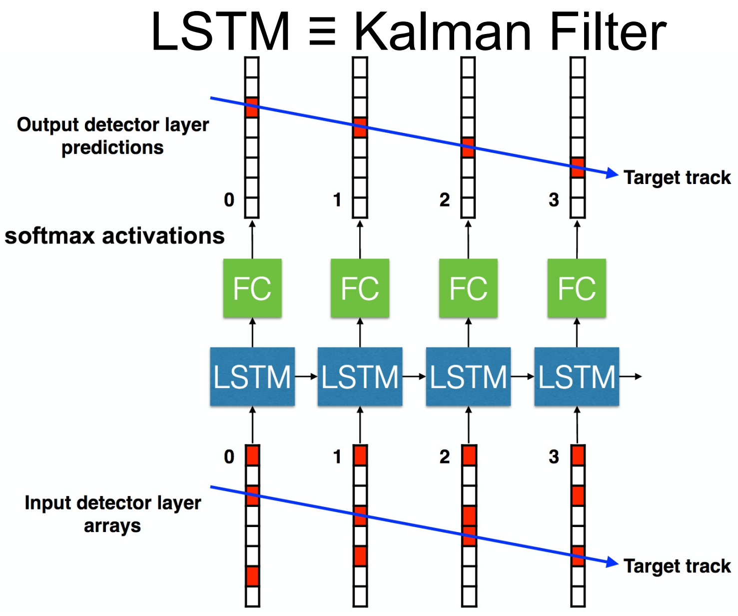
R&D

Surrogate Kalman Filter Approaches

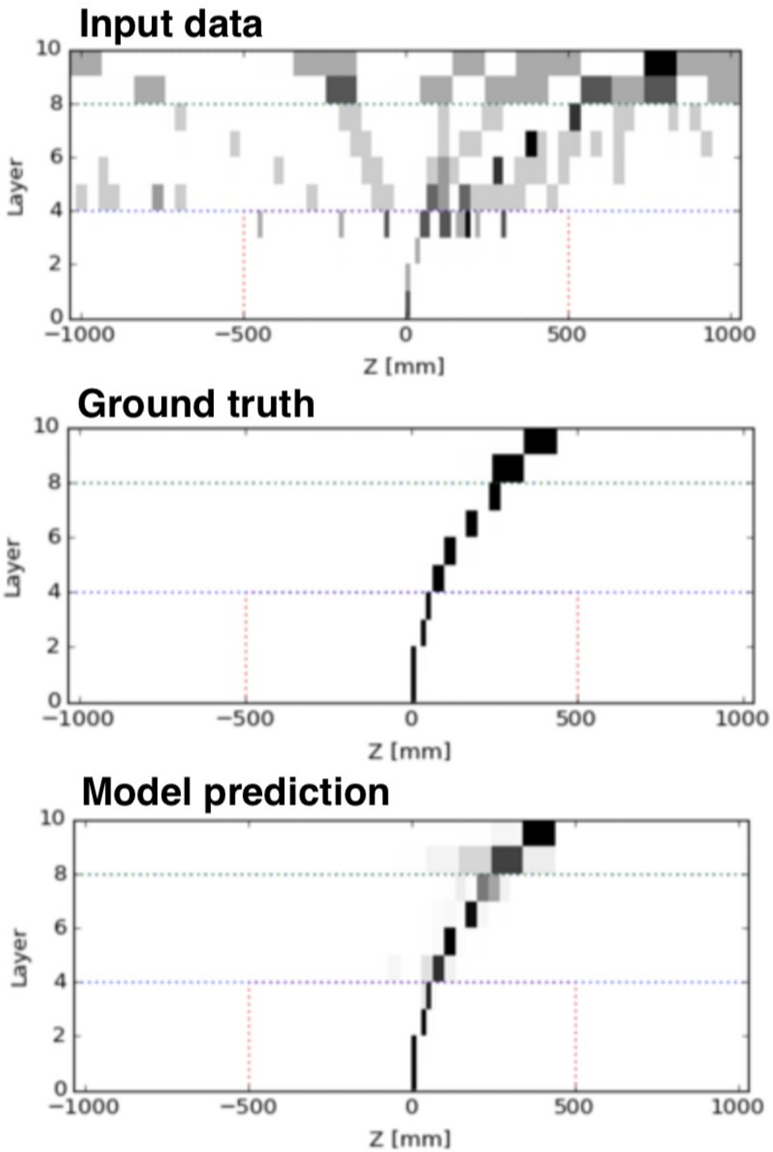
https://heptrkx.github.io/

digression

Finding Tracks with LSTM



- Search seeded from a known tracklet
- Hit location is discretized to fixed length
- Model predicts the binned position of the hit on the next layer



Hit Following with Uncertainty

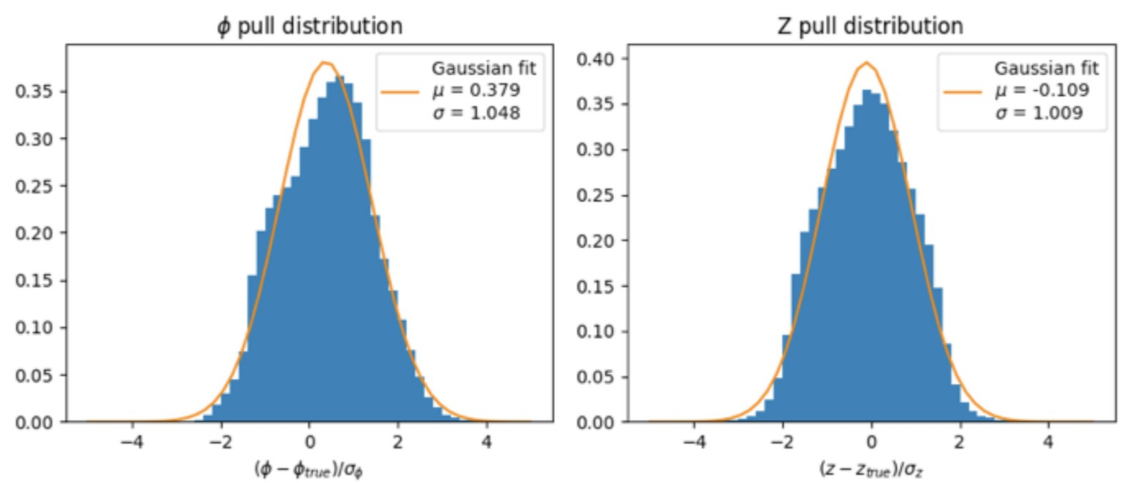
$$\boxed{\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{N-1}} \rightarrow \text{LSTM} \rightarrow \text{FC} \rightarrow \boxed{(\hat{\vec{r}}_1, \Sigma_1), (\hat{\vec{r}}_2, \Sigma_2), \ldots, (\hat{\vec{r}}_N, \Sigma_N),}$$

$$\vec{r} = (r, \phi, z) \qquad \hat{\vec{r}} = (\hat{\phi}, \hat{z}) \quad \Sigma = \begin{pmatrix} \sigma_\phi^2 & \sigma_{\phi z}^2 \\ \sigma_{\phi z}^2 & \sigma_z^2 \end{pmatrix}$$

Loss function incorporates the position and the predicted uncertainty

$$L(x, y) = \log|\Sigma| + (y - f(x))^{\mathrm{T}} \Sigma^{-1} (y - f(x))$$

- Search seeded from a known tracklet
- Hit positions taken in sequential input
- Model predicts the position/error of the hit on the next layer



R&D

Pattern Recognition / Seeding

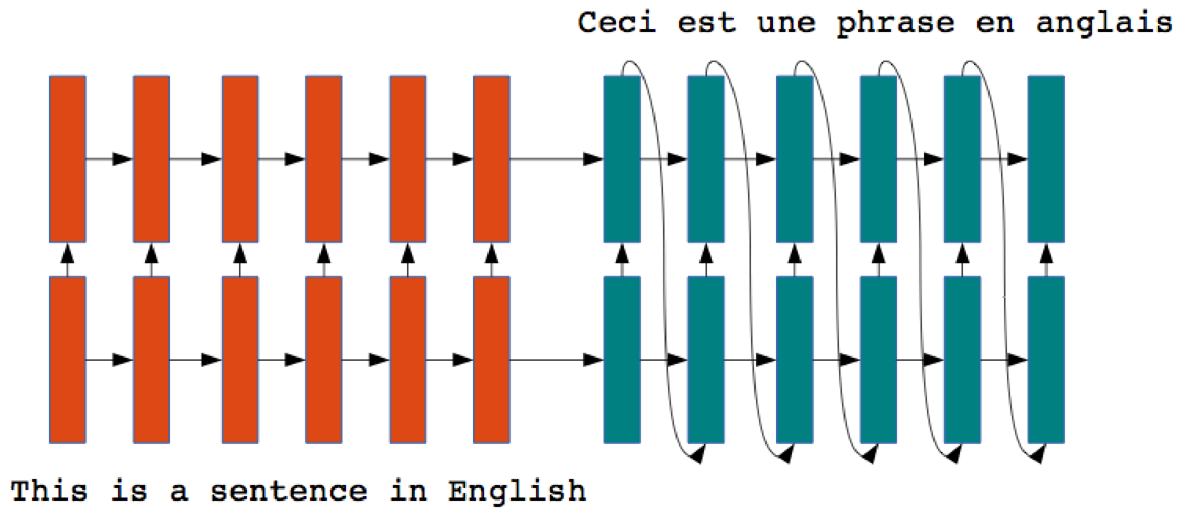
https://heptrkx.github.io/

digression

Text Translation

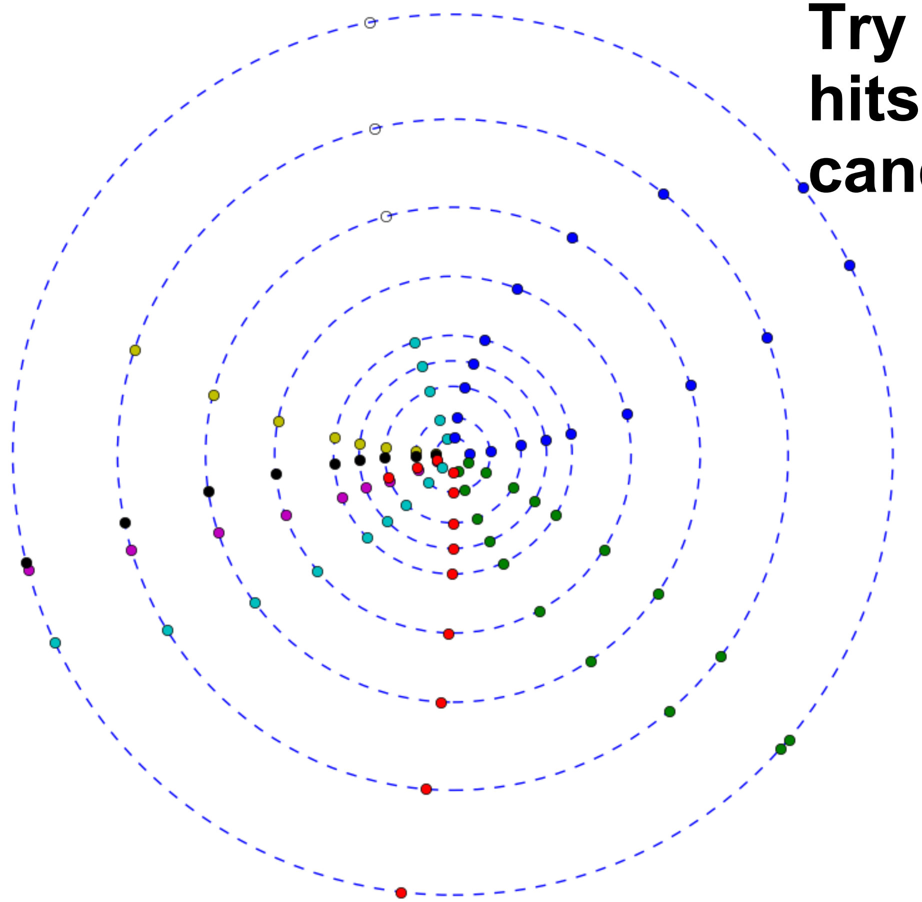
[Sutskever et al. NIPS 2014]

- Multiple layers of very large LSTM recurrent modules
- English sentence is read in and encoded
- French sentence is produced after the end of the English sentence
- Accuracy is very close to state of the art.



→ From sequence of hits on layer to sequence of hits on track

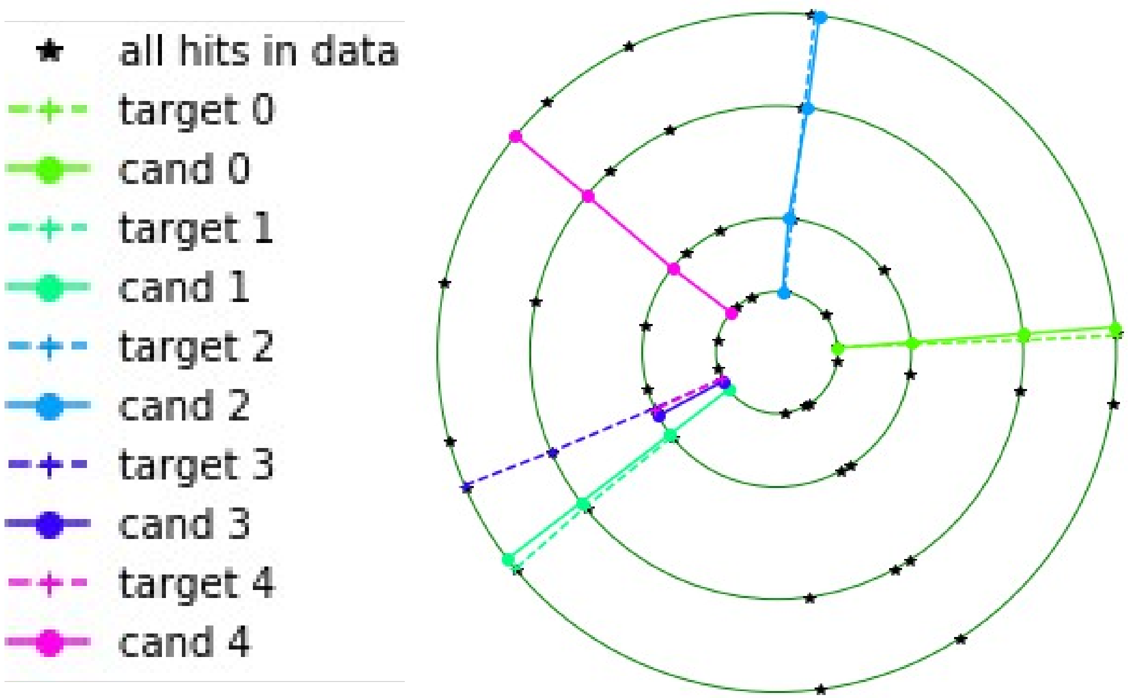
Pattern Recognition



Try to assemble hits into track candidates.

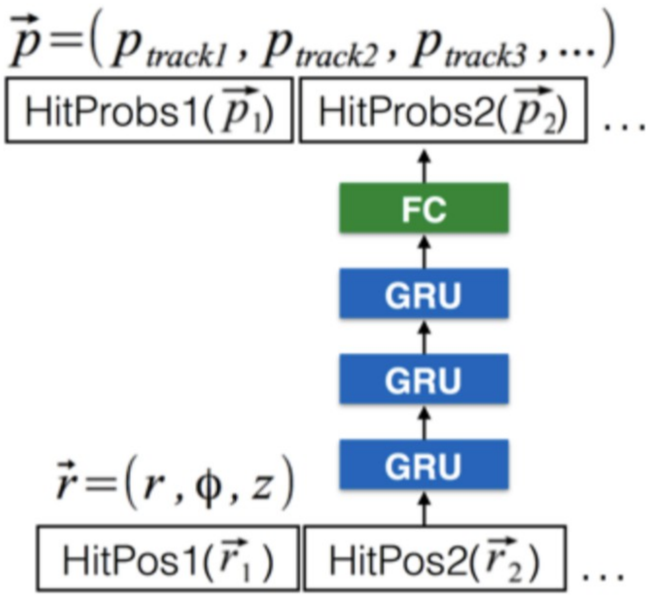
Pattern Recognition with LSTM

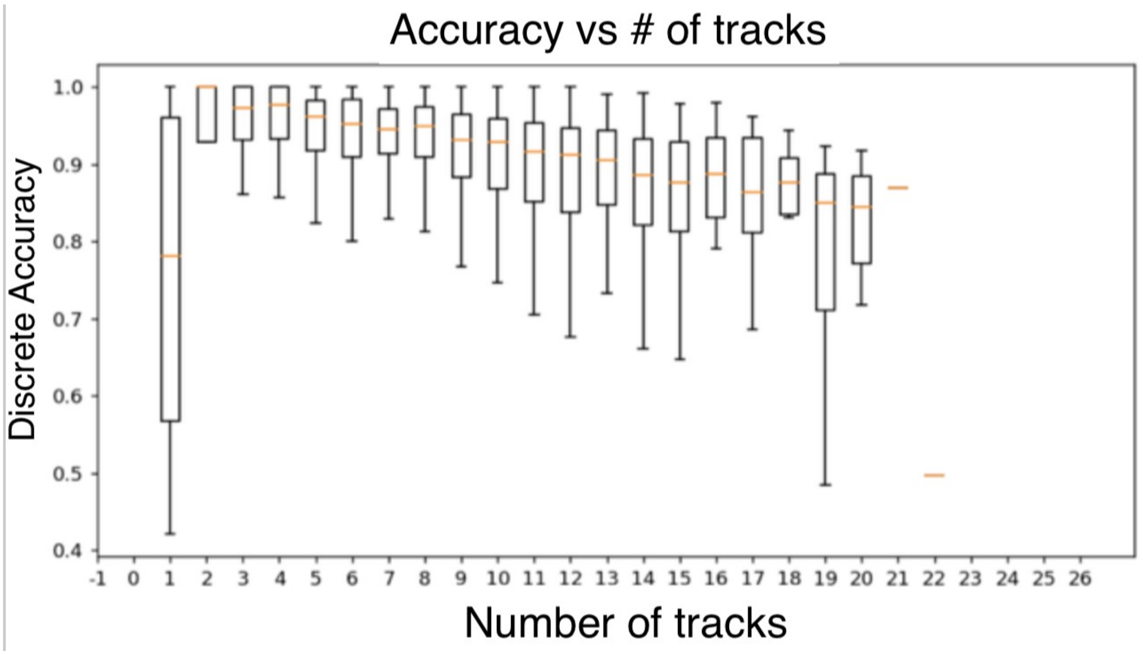
- Input sequence of hits per layers (one sequence per layer)
  - One LSTM cell per layer
- Output sequence of hits per candidates
  - Final LSTM runs for as many candidates the model can predict



- Still work in progress
- Restricted to 4 layers (with seeding in mind)
- Work to some extend

Hit/Track Assignment

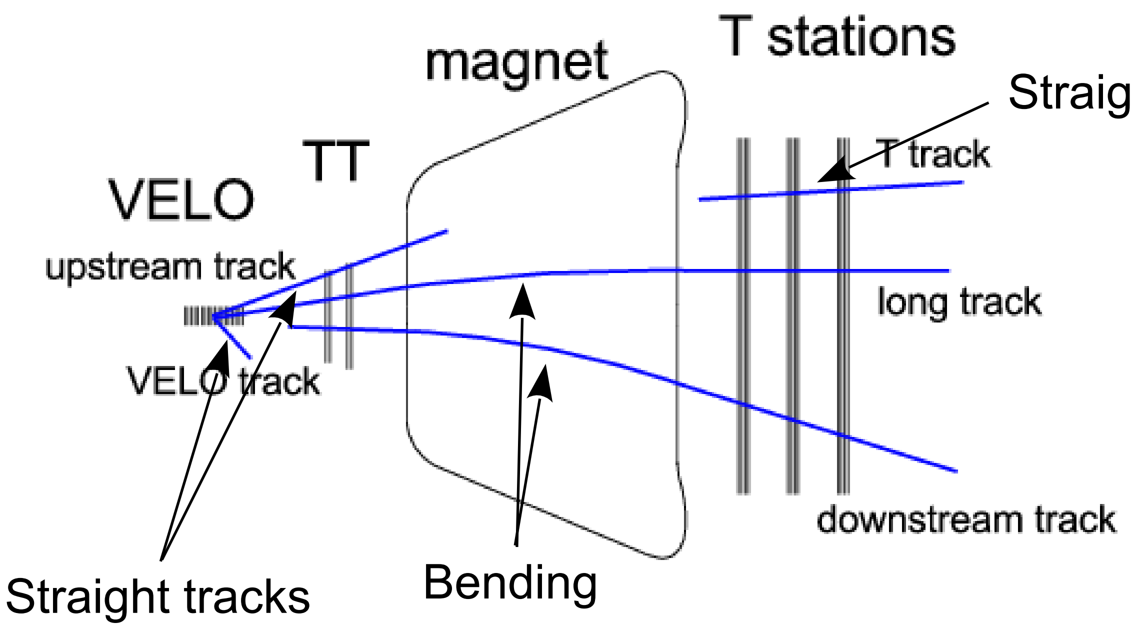


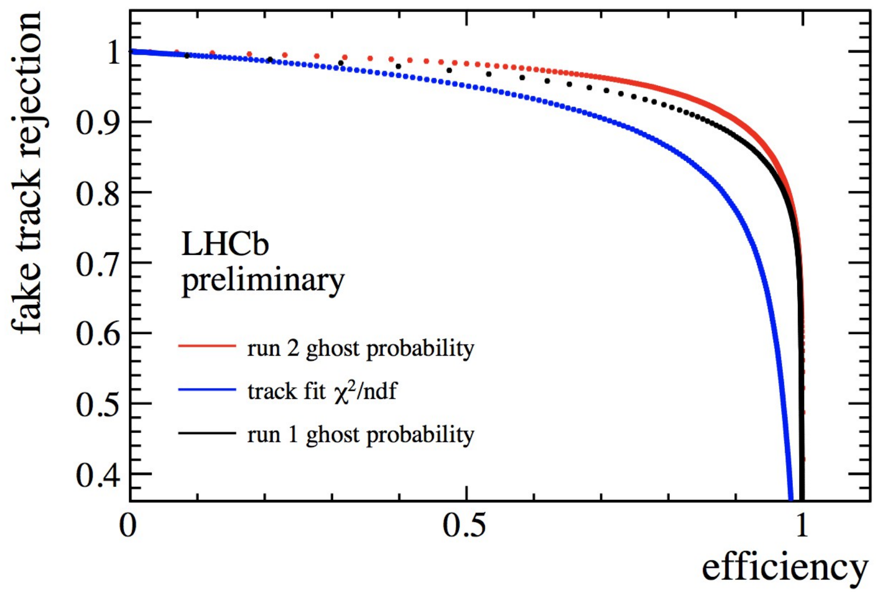


- Unseeded hit-to-track assignment (clustering)
- Hit positions taken in sequential input
- Model predicts the probability that a hit belongs to a track candidate

Track Selection

Track Selection

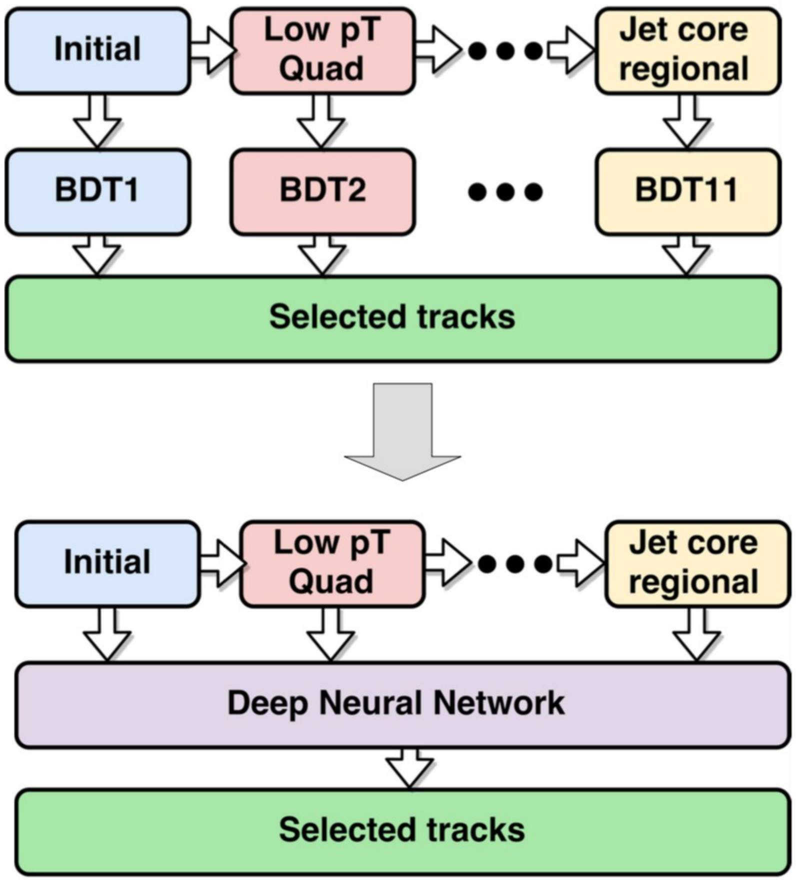


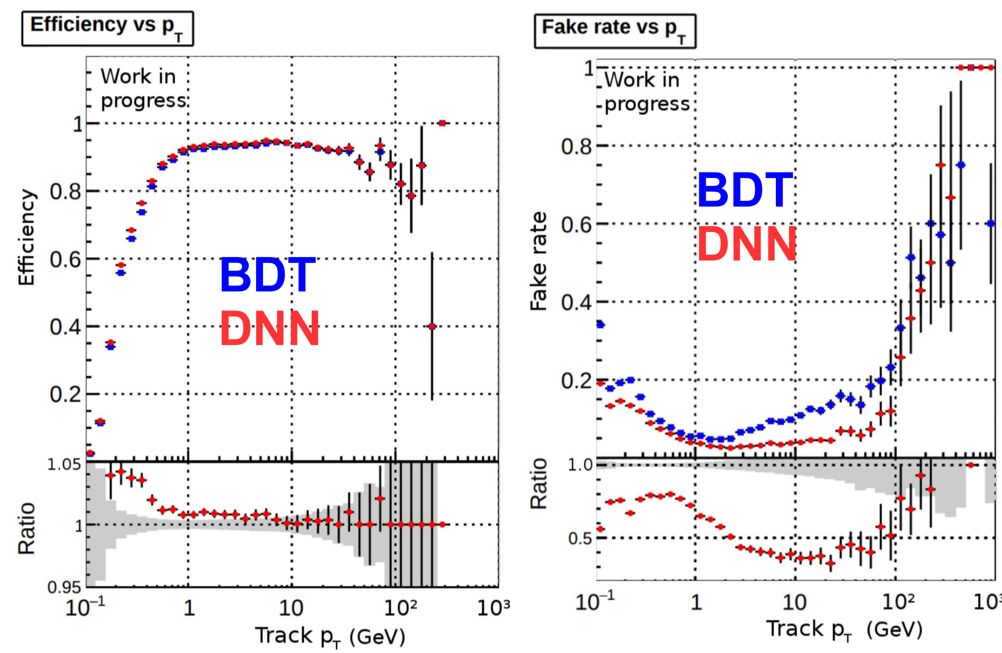


NN classifier implemented to select good from bad tracks in forward tracking and downstream tracking

http://cds.cern.ch/record/2255039

Track Quality with DNN





Simplifies and improves track selection within the scope of CMS iterative tracking

https://indico.cern.ch/event/658267/contributions/2813693/

Track Parameters

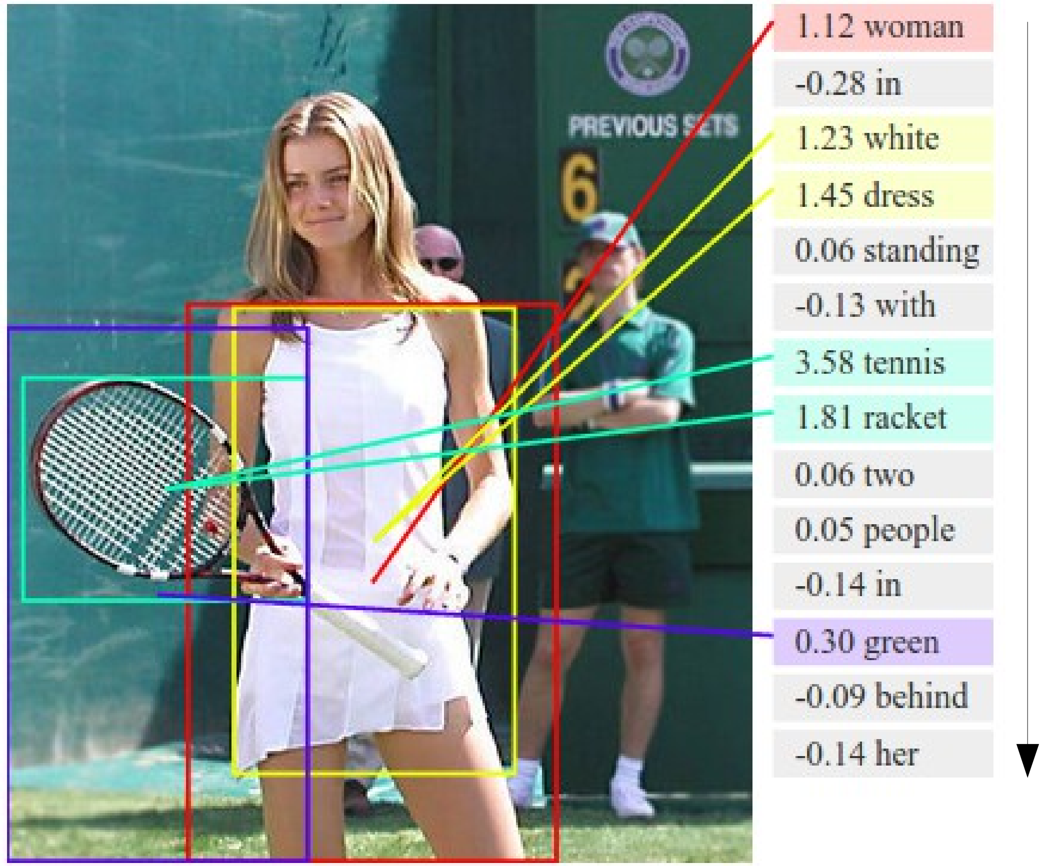
R&D

Track Parameters Measurement

https://heptrkx.github.io/

digression

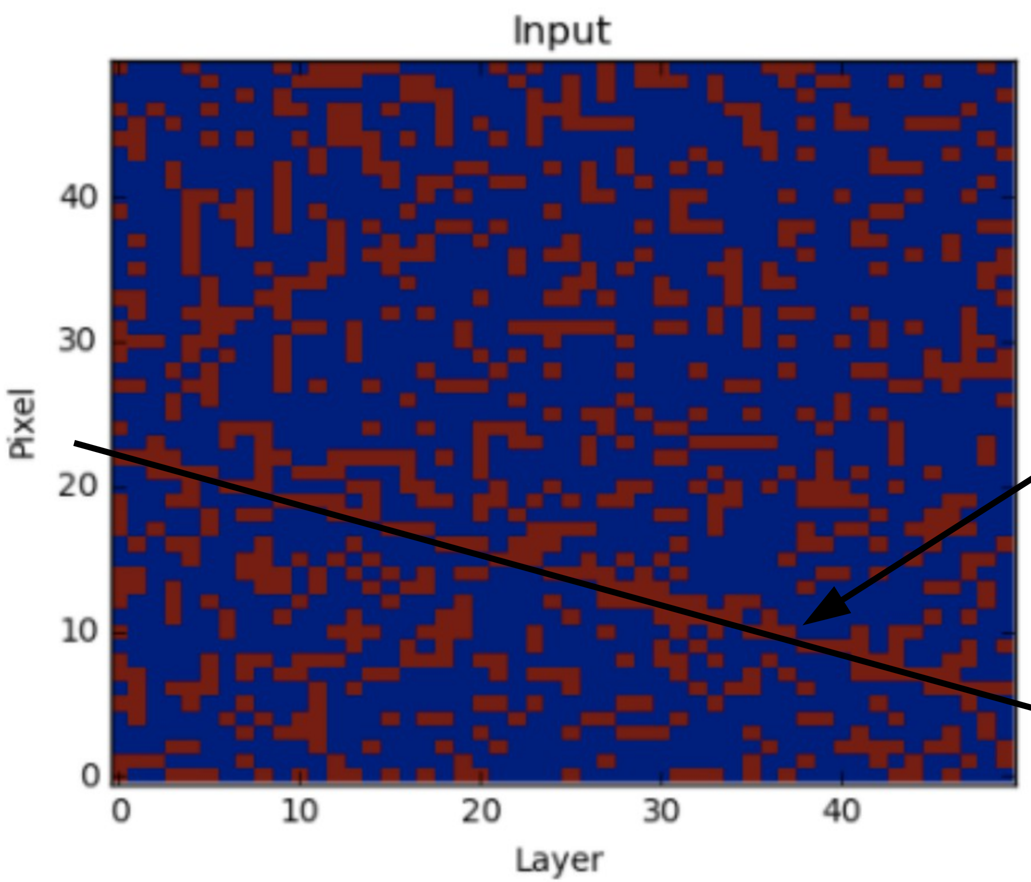
Scene Captioning



Karpathy, Fei-Fei, CVPR 2015

→ Compose tracks explanation from image

Track Parameter Estimation

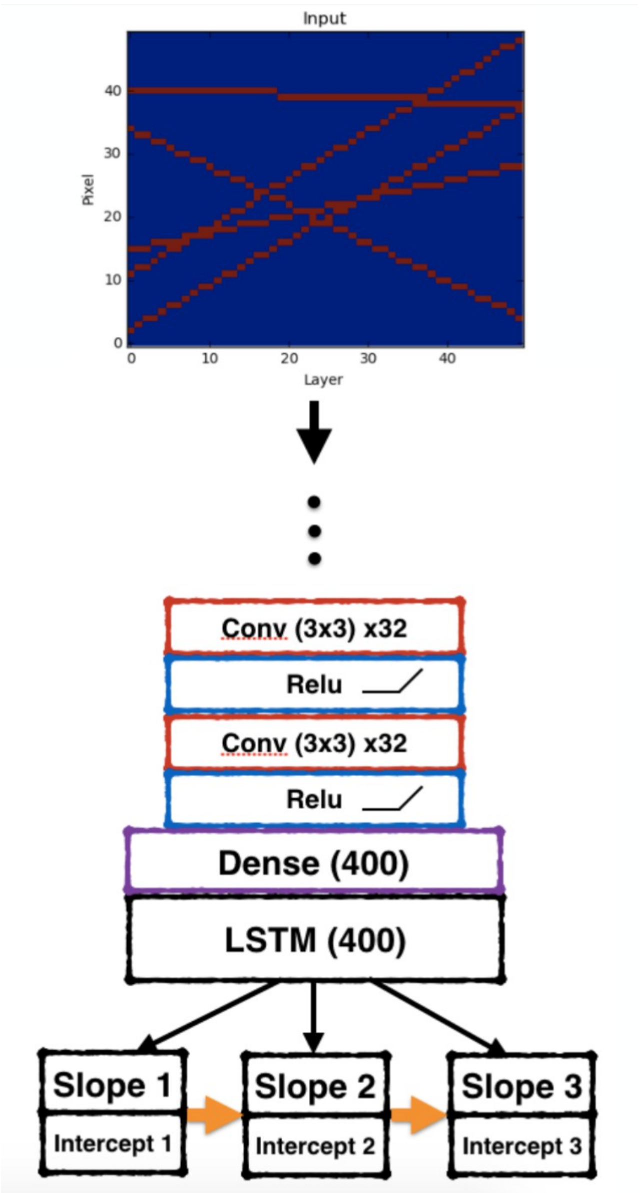


Try to predict the slope and intersect of this track

Multi-Track Prediction with LSTM

- Hit pattern from multiple track processed through convolutional layers

- LSTM Cell runs for as many tracks the model can predict.

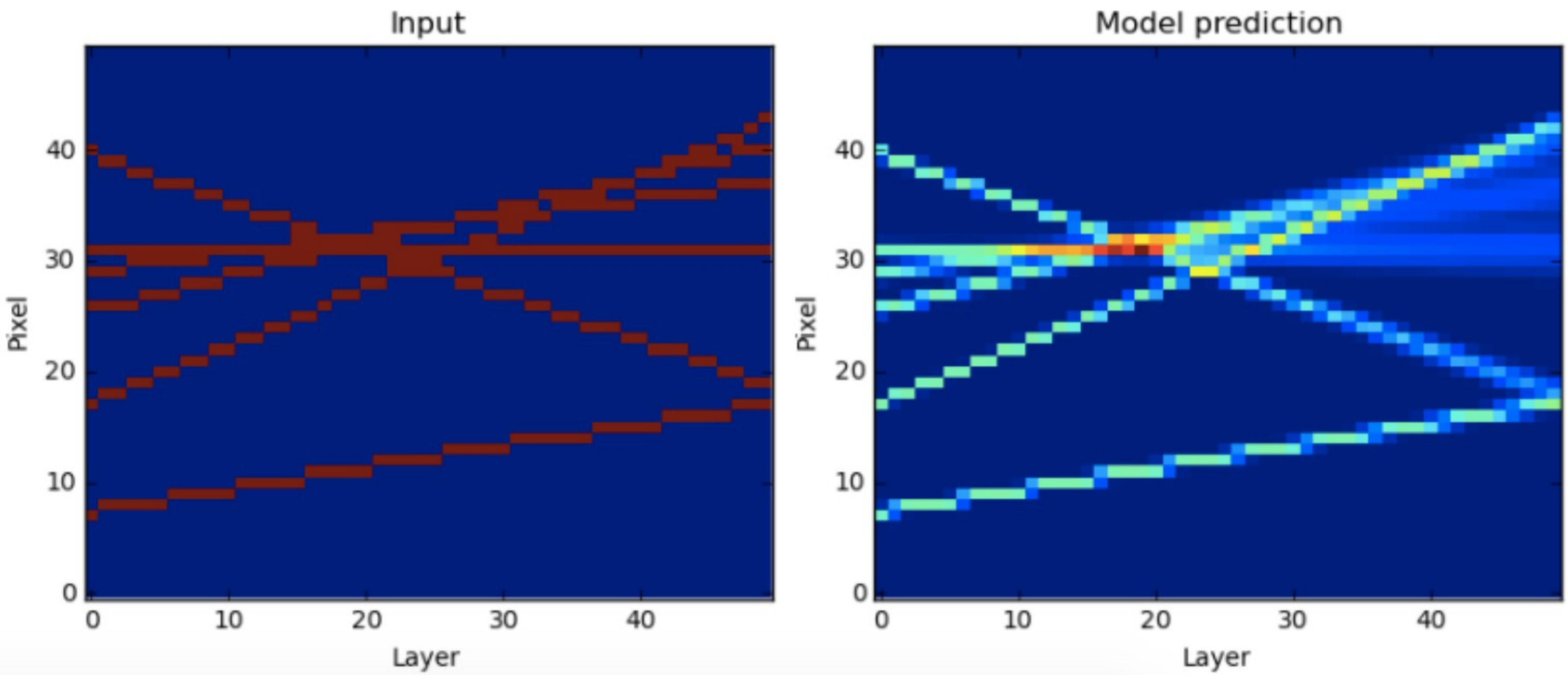


Prediction Track Covariance

Model is modified to predict a covariance matrix for which there is no ground truth, but is used with the modified loss function

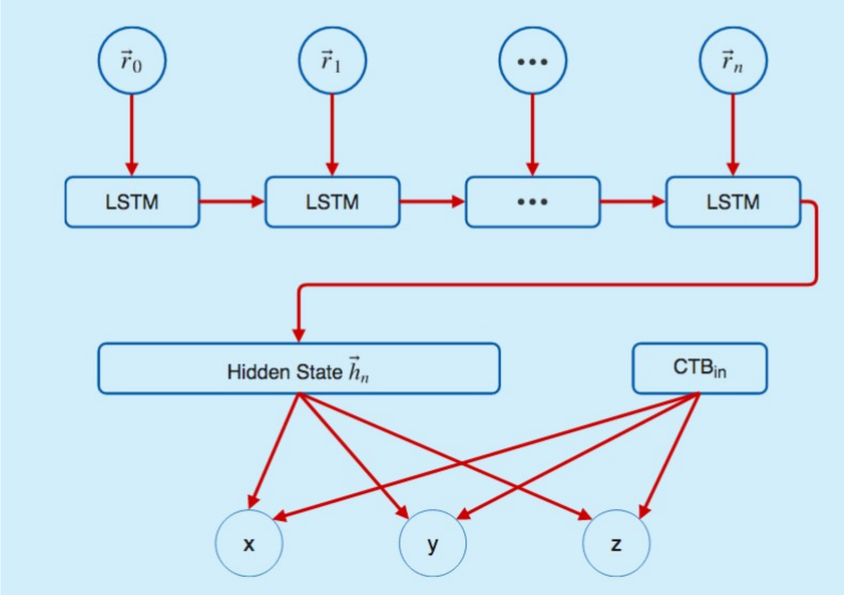
$$L(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{\Sigma}| + (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))$$

Track Parameters Uncertainty

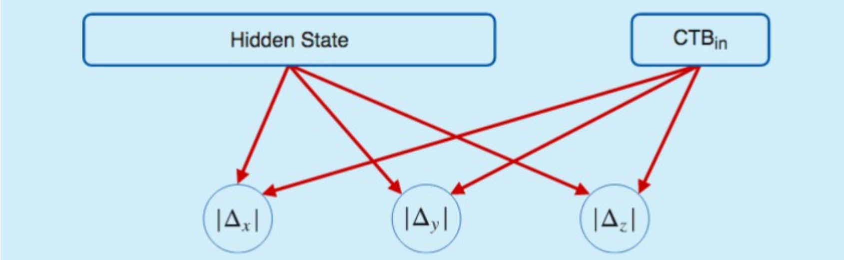


Representation of track slope, intersect and respective uncertainties

Impact Parameters

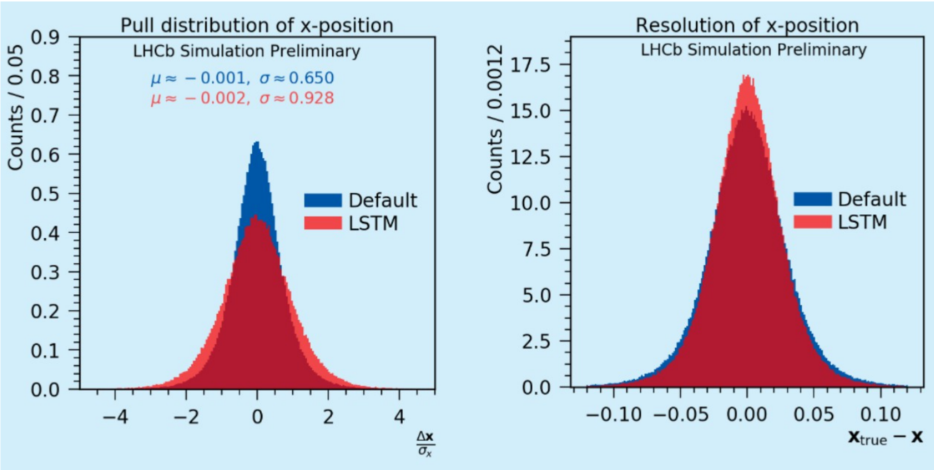


Model architecture to predict CTB position



Model extension to predict the uncertainty on the CTB position

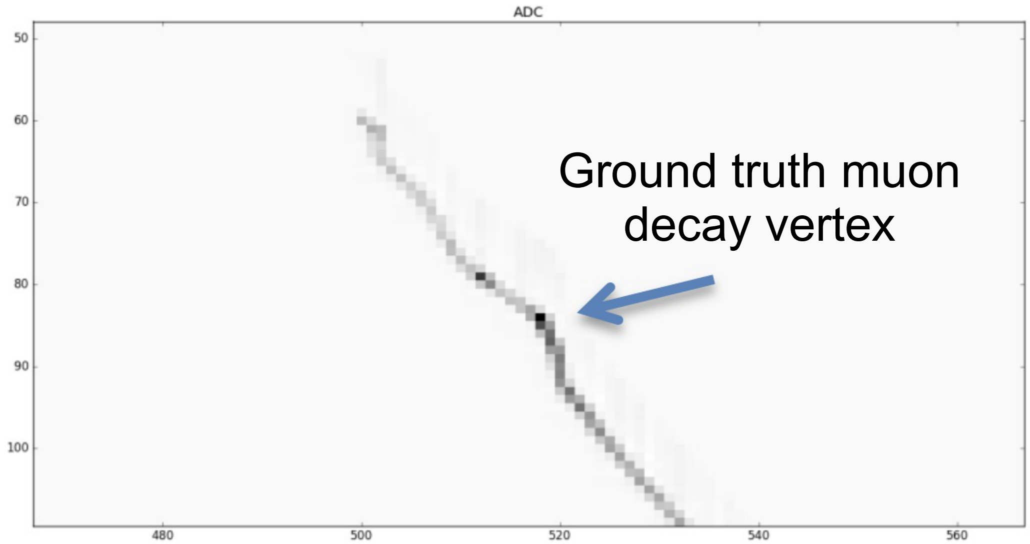
- LSTM model supplements a Kalman Filter approach
- Improve resolution and estimation of track impact parameters in LHCb

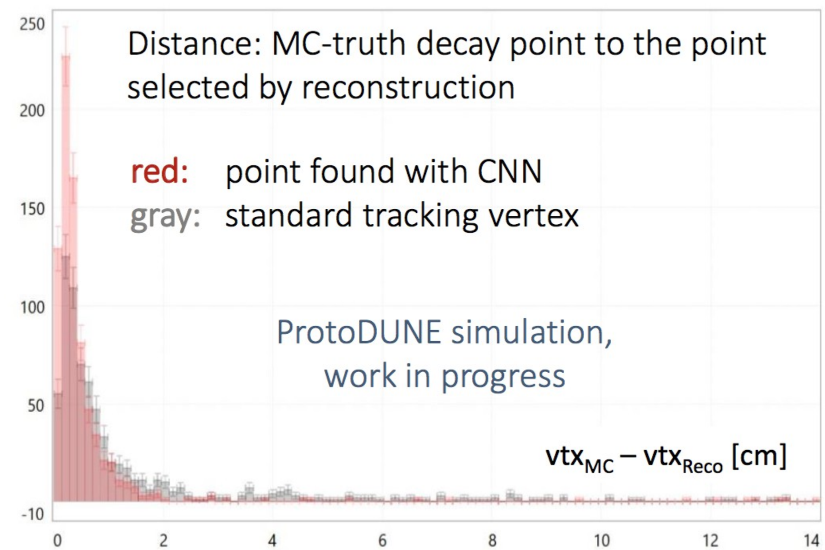


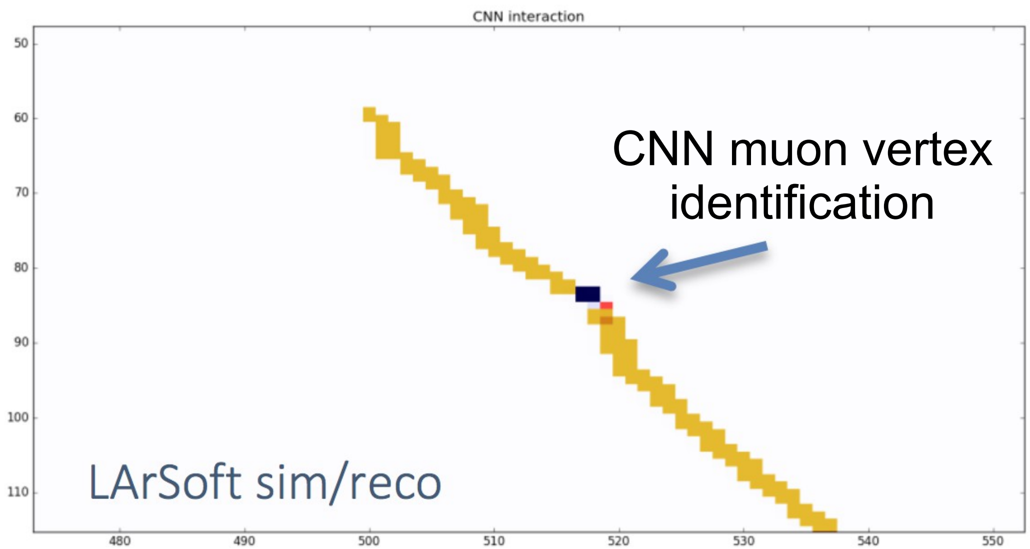
https://indico.cern.ch/event/587955/contributions/2935754/

Vertexing

Decay Point Identifier



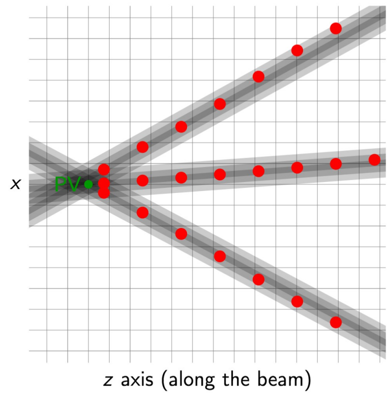


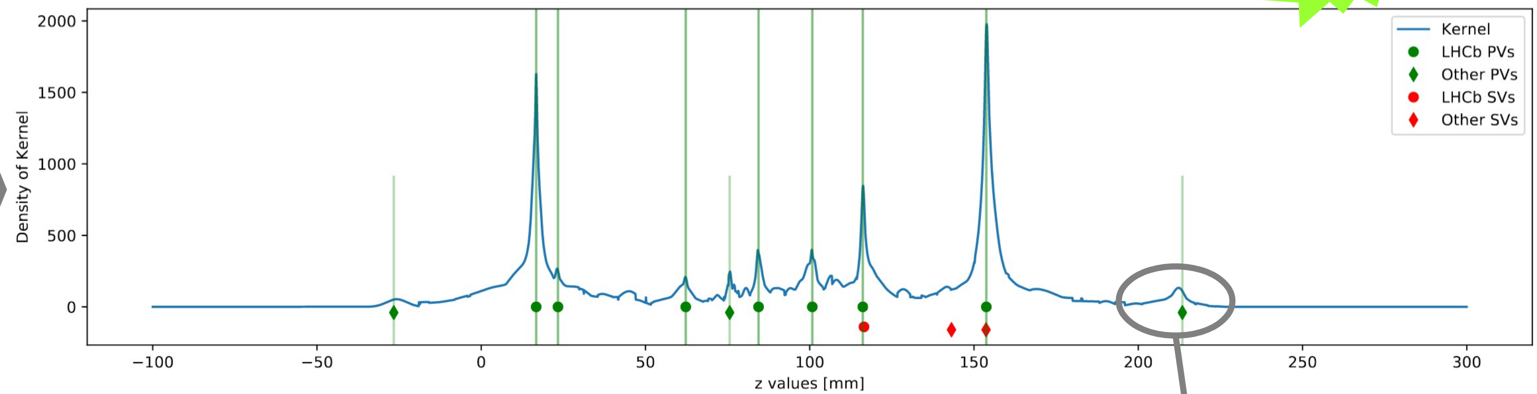


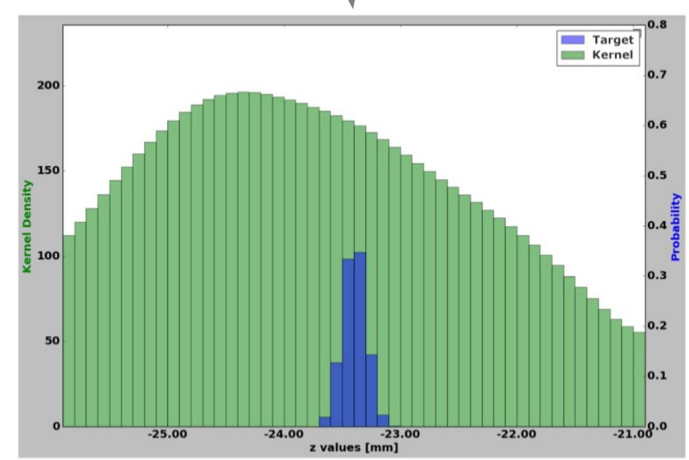
- CNN slightly outperform the classical approach
- Much less complication in programming the vertex finding

Hybrid Vertexing

R&D





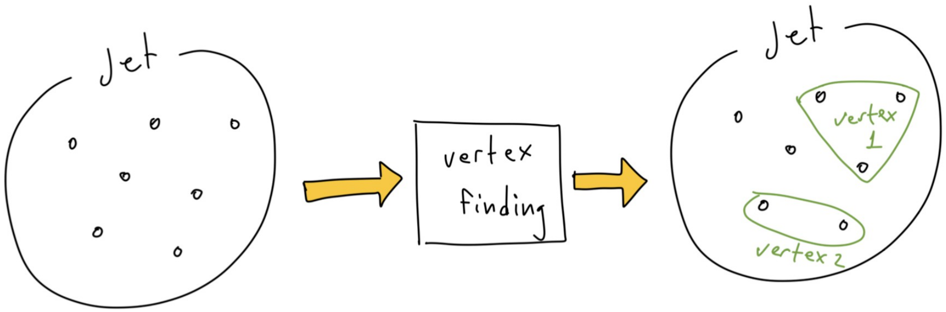


- Form a track density over longitudinal axis using Gaussian kernels
- Learn vertex position from local longitudinal density
- Similar performance with traditional approach.
- Advantage of ML in deployment

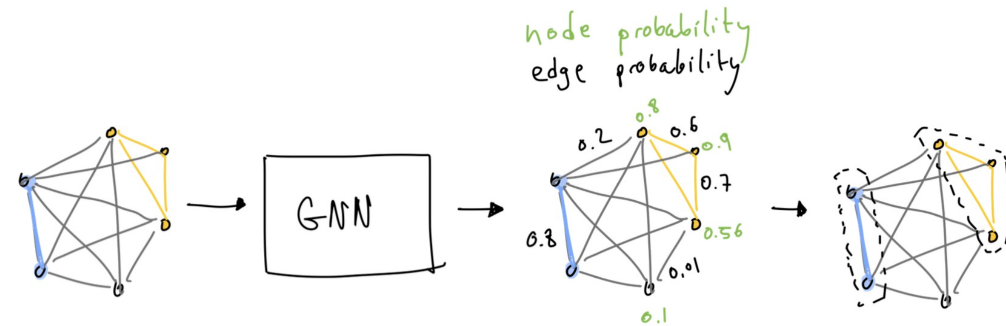
https://indico.cern.ch/event/708041/contributions/3269692/

Graph Network Vertexing

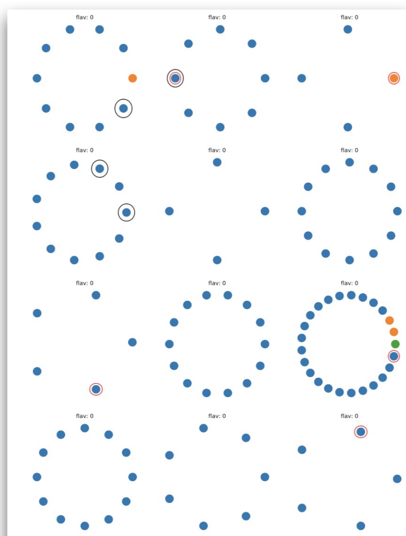
In terms of an algorithms input/output:



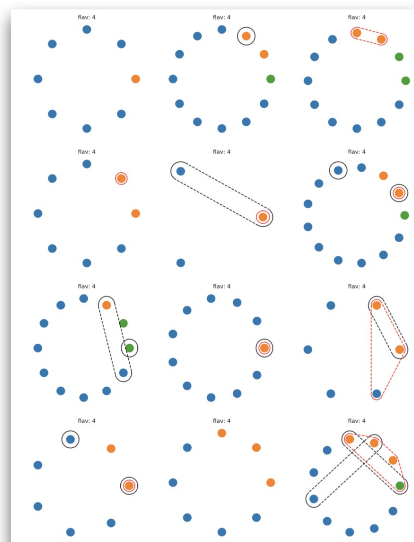
How a GNN is used to assign nodes to vertices:



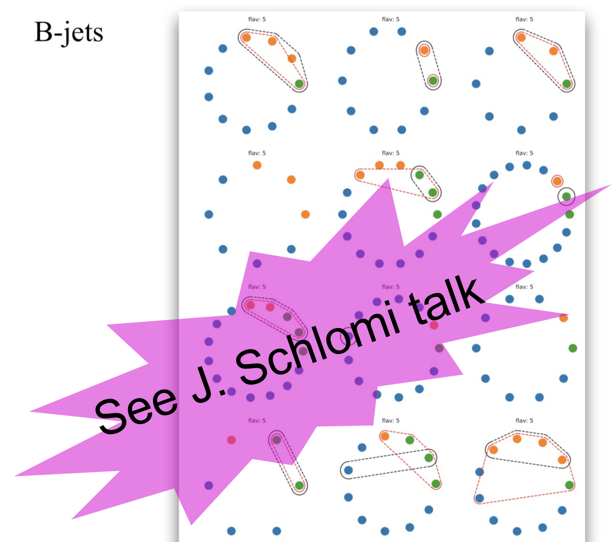
Light-jets



C-jets



B-jets

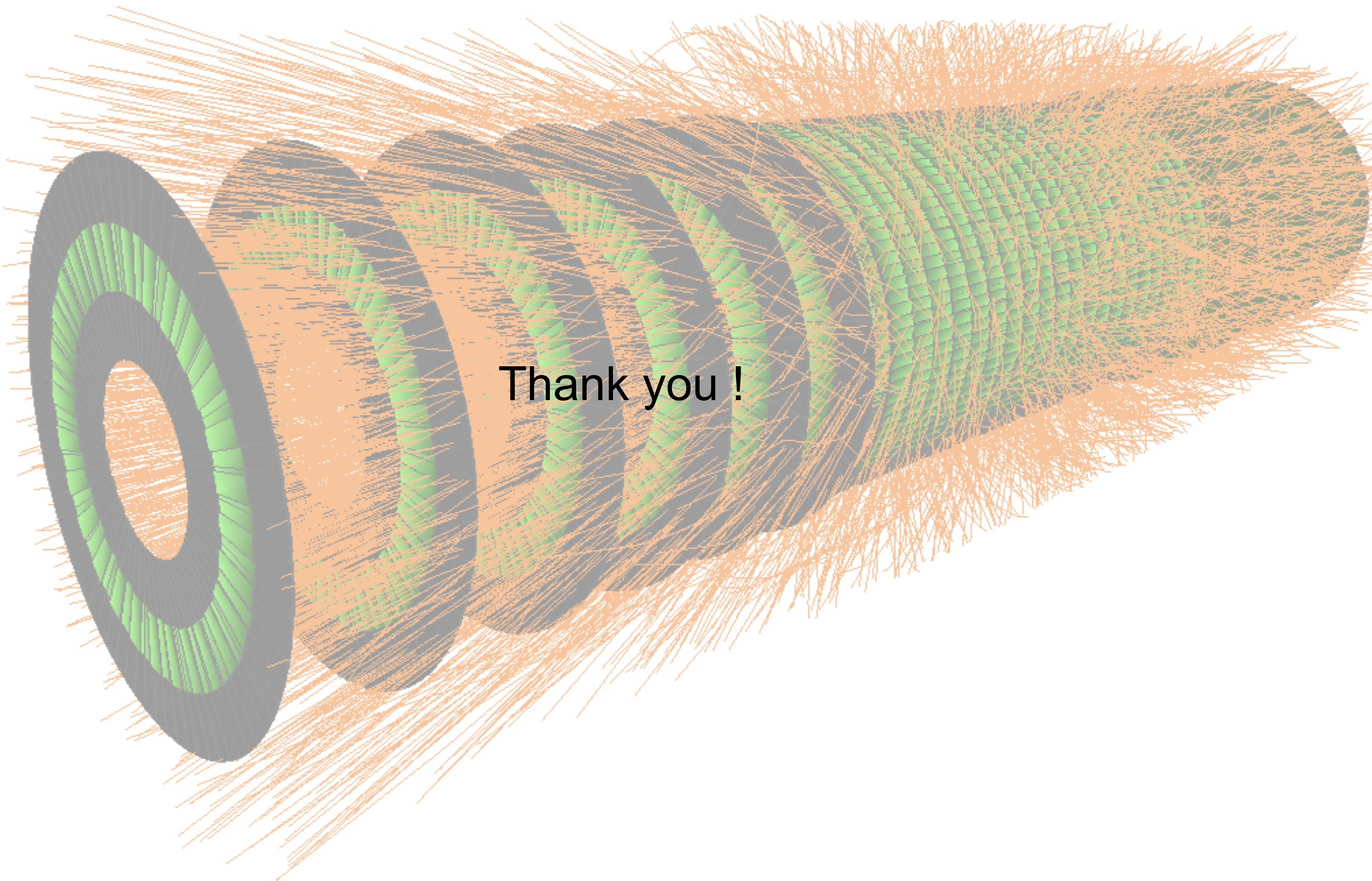


- Track clustering within jets for secondary vertex reconstruction and jet-tagging in-fine

Variety of application of machine learning for tracking-related tasks.

and beyond

- ML-4-Tracking already in production
- End-to-end tracking with ML is unlikely
- Active field of R&D for novel methods
- Interesting directions for ML
  - ✓ Reducing the running complexity
  - ✓ Graph network approach
  - ✓ ML-guided combinatorial track finder
  - ✓ Incorporating domain knowledge


Thank you !