HEP.TrkX project approaching the charged particle reconstruction with deep learning, for online and offline data processing

Jean-Roch Vlimant for the HEP.TrkX project

CHEP 2018
Sofia, 8-13 July 2018

HEP.TrkX Project

- Pilot project funded by DOE ASCR and COMP HEP
- Part of HEP CCE
- Mission
  - Explore deep learning techniques for track formation

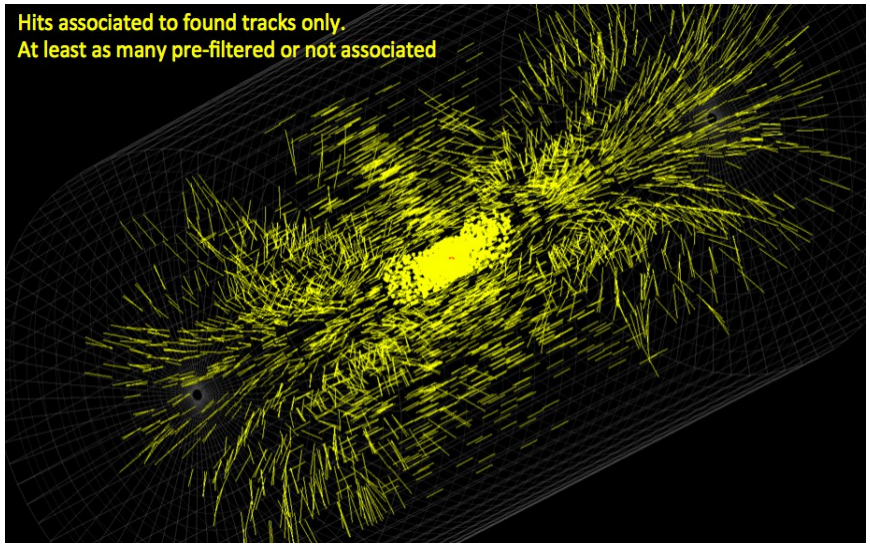
- People
  - **LBL :** Paolo Calafiura, Steve Farrell, Mayur Mudigonda, Prabhat
  - **FNAL :** Giuseppe Cerati, Lindsey Gray, Jim Kowalkowski, Panagiotis Spentzouris, Aristeidis Tsaris
  - **Caltech :** Dustin Anderson, Josh Bendavid, Pietro Perona, Maria Spiropulu, Jean-Roch Vlimant, Stephan Zheng

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

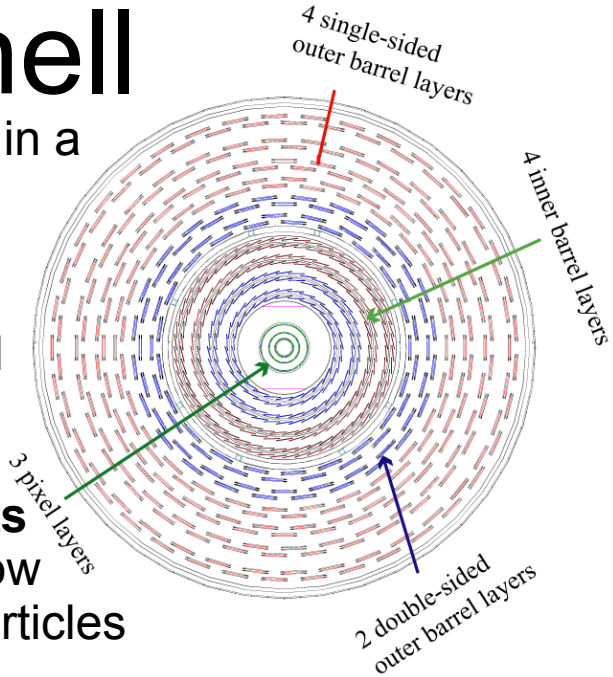
Outline

- The challenge of Charged Particle Tracking
- Pattern Recognition in Data Science
- Several deep learning approaches

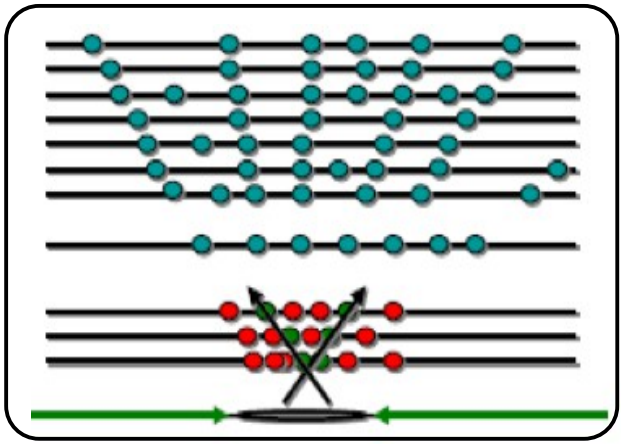
Tracking in a Nutshell



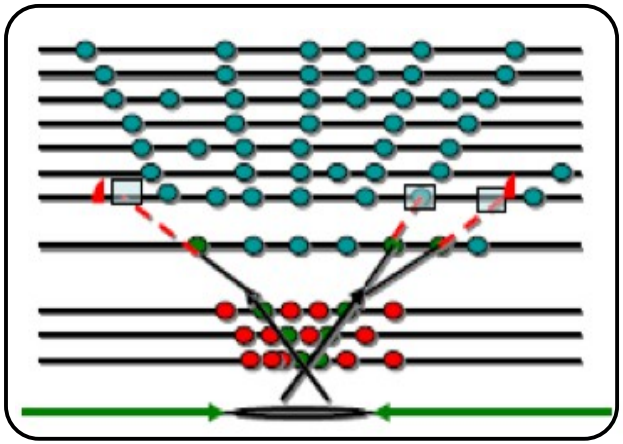
- Particle trajectory bended in a solenoid magnetic field
- Curvature is a proxy to momentum
- Particle ionize silicon pixel and strip throughout several concentric layers
- **Thousands of sparse hits**
- Lots of hit pollution from low momentum, secondary particles

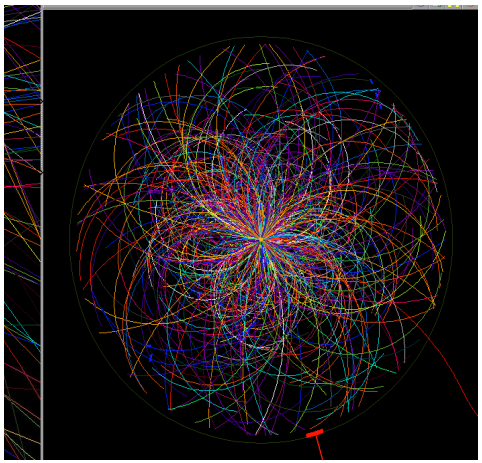


Seeding



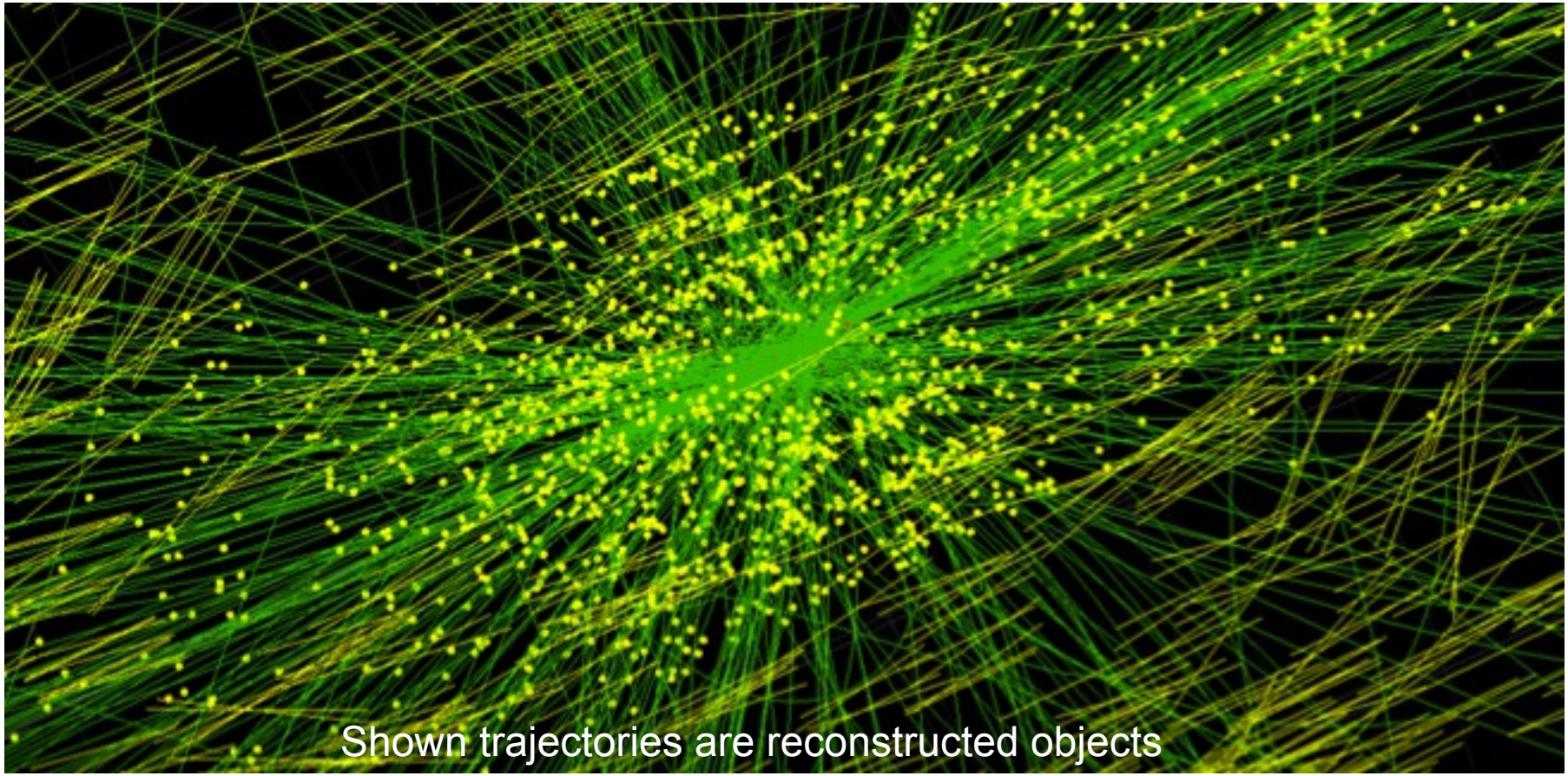
Kalman Filter





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

Complexity and Ambiguity

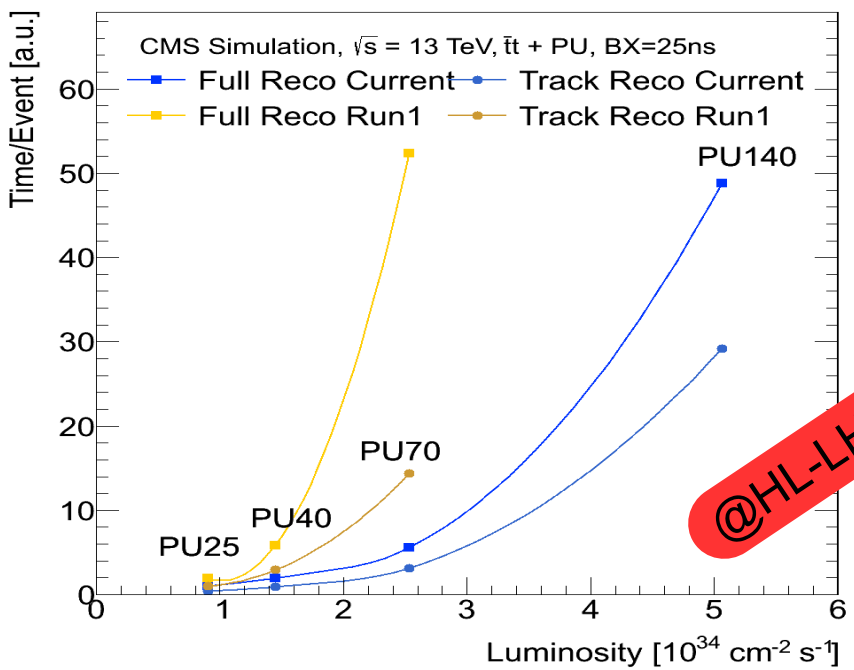


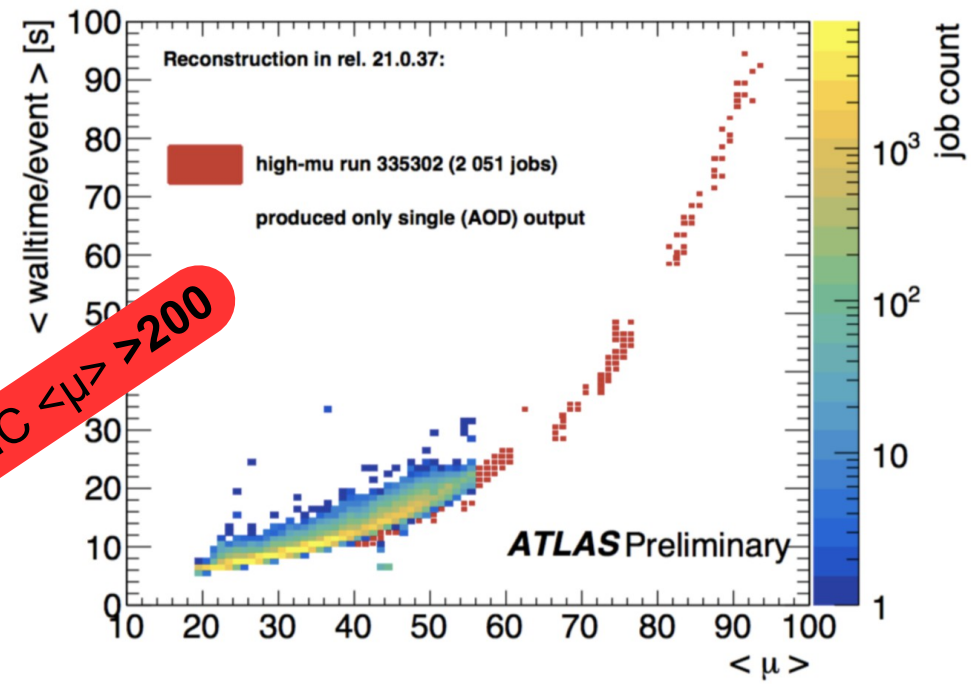
The future holds **much more hits**

High Luminosity LHC
The Challenge

Cost of Tracking

- CPU time consumption in HL-LHC era **surpasses computing budget**
  - ➔ Need for **faster algorithms**
- Charged particle track reconstruction is one of the most **CPU consuming task** in event reconstruction
  - ➔ Optimizations **mostly saturated**
- Large fraction of CPU required in the HLT. **Cannot perform tracking inclusively**
  - ➔ **Approximation** allowed in the trigger

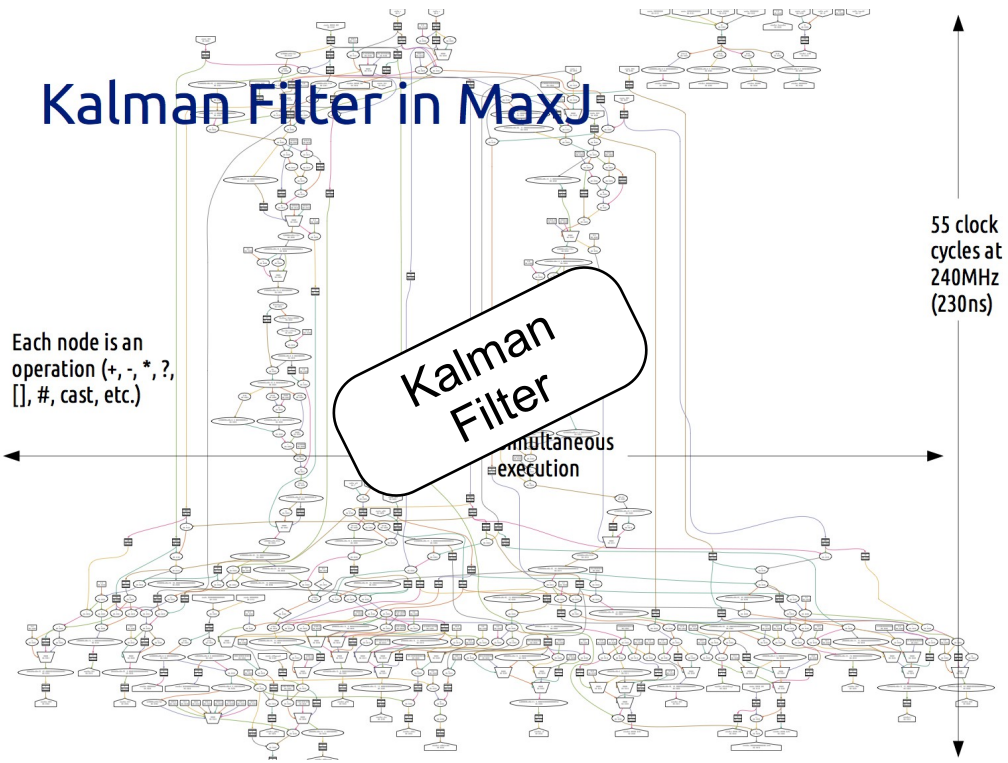


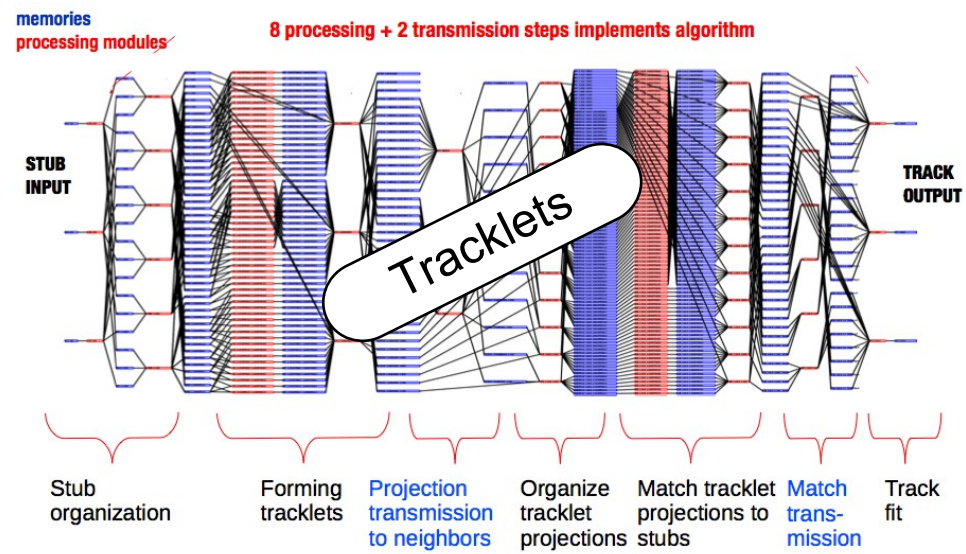


@HL-LHC <μ> >200

Fast Hardware Tracking

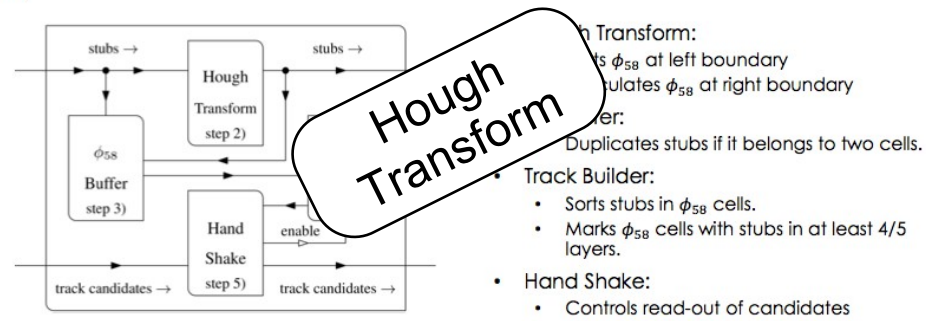
- Track trigger implementation for Trigger upgrades development on-going
- Several approaches investigated
- **Dedicated hardware is the key** to fast computation.
- **Not applicable for offline** processing unless through adopting heterogeneous computing.





Firmware Implementation - Bin

- Each bin represents a $^{q}/_{p_T}$ column in the HT array



See https://ctdwit2017.lal.in2p3.fr/

Motivations

Current algorithms for tracking are highly **performant physics-wise** and scale **badly computation-wise**

Faster implementations are possible with **dedicated hardware**

Go **back to the blackboard** for new approaches

Pattern Recognition With Deep Learning

Machine Learning for Tracking

Zagoruyko et al, 1604.02135

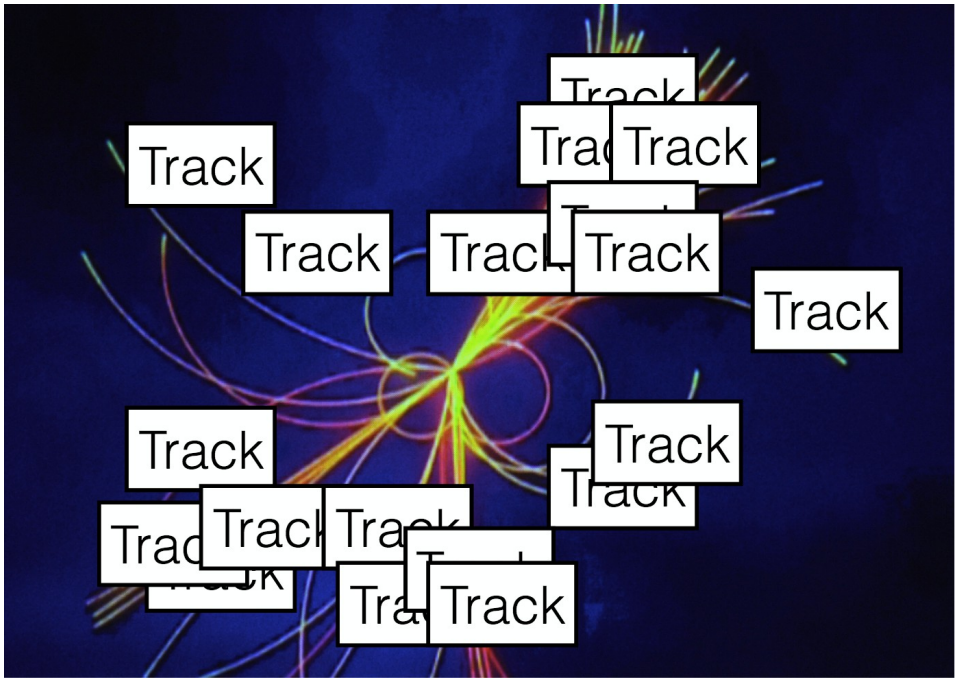


Photo by Pier Marco Tacca/Getty Images

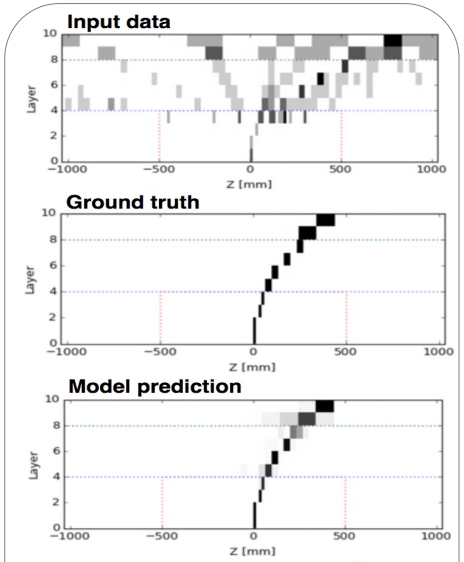
Many possible ways to cast the algorithm of tracking, or part of the current algorithms in a machine learning problem

Similarities and Challenges

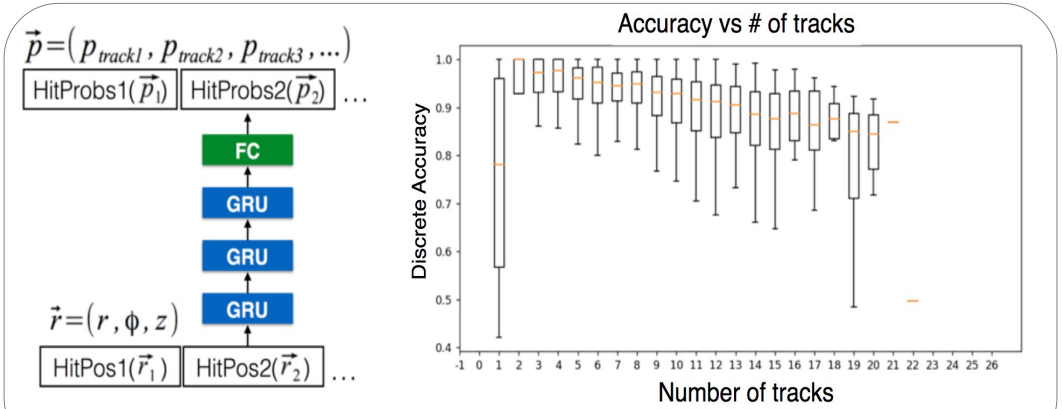
- Particle tracking is an active field in data science
  - Different type of particles
  - Not oriented to code performance
- Making a track is a pattern recognition problem
  - Not the usual one in data science
- Tracking data is much sparser than regular images
  - Test and adapt methods
- Tracking device may have up to 10M of channels
  - Scale up deep learning models
  - Perform tracking by sector
- Underlying geometry of sensor more complex
  - More than a simple picture
  - Barrels and end-caps are not the usual pictures
- Not the regular type of sequences
  - Cover new ground of sequence processing
- Defining an adequate cost function
  - Tracking algorithms are optimized by proxy
- A solution must be performant during inference ...

Approaching tracking with deep learning

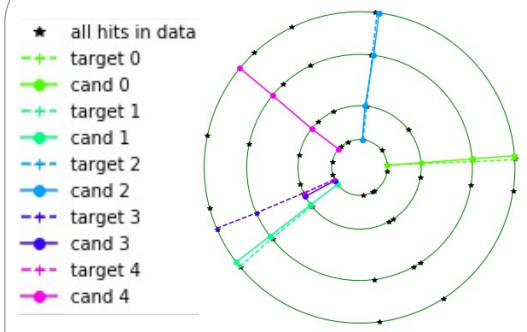
HEP.TrkX Approaches



Track following with RNN

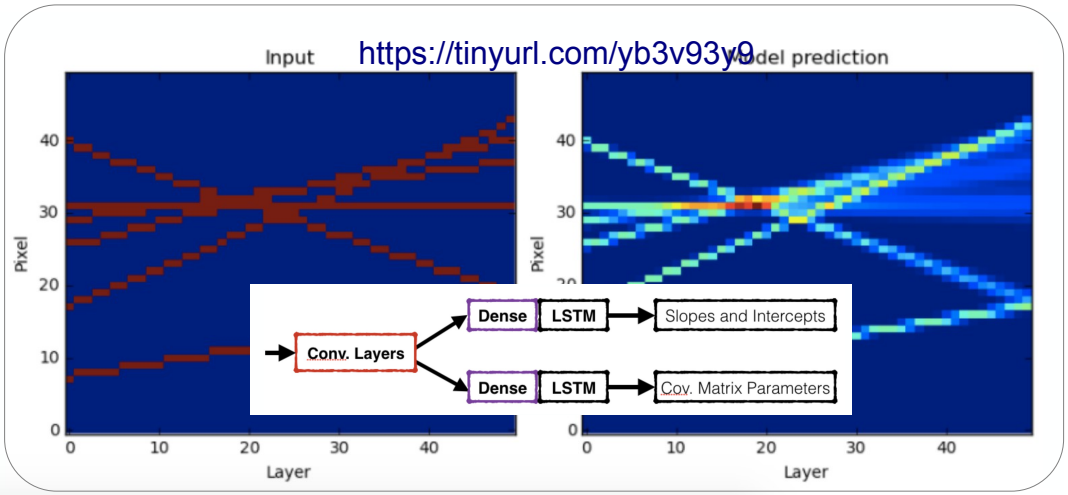


End-to-end hit assignment

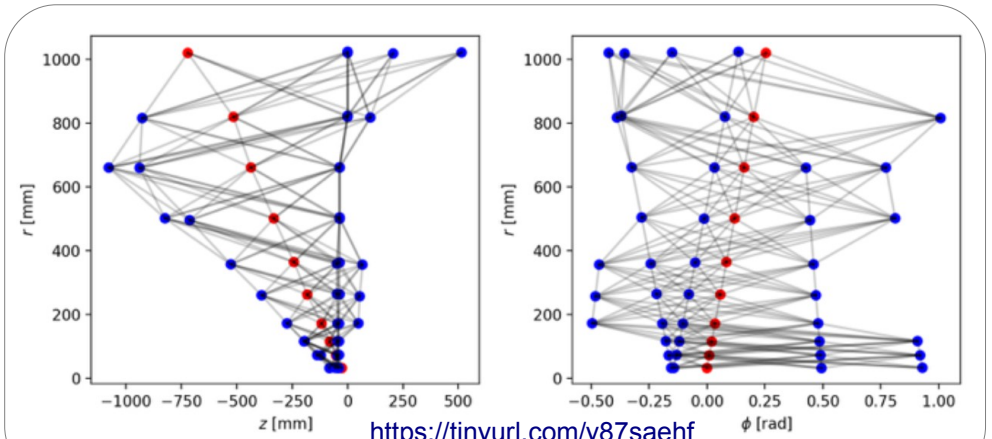


Seq-to-seq track finding

https://heptrkx.github.io/



https://tinyurl.com/yb3v93y9

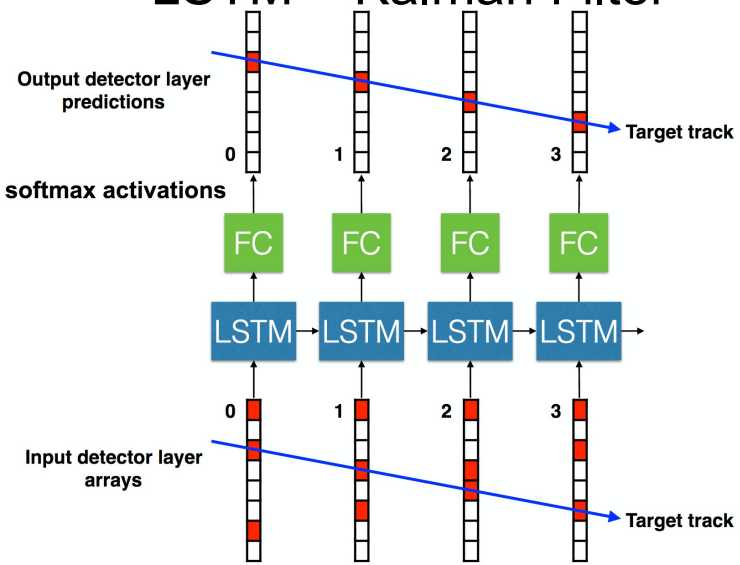


https://tinyurl.com/y87saehf

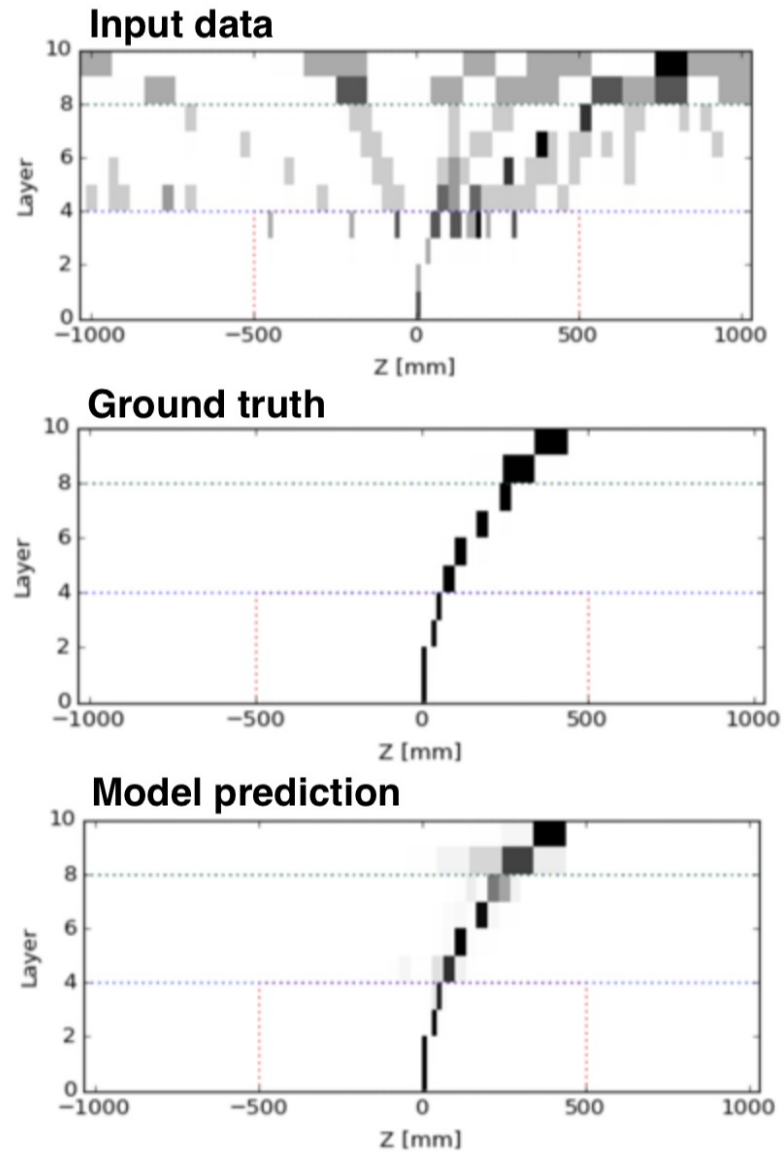
Seeded Track Candidate Making

Finding Tracks with LSTM

LSTM ≡ Kalman Filter



- 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 Prediction with Gaussian Model

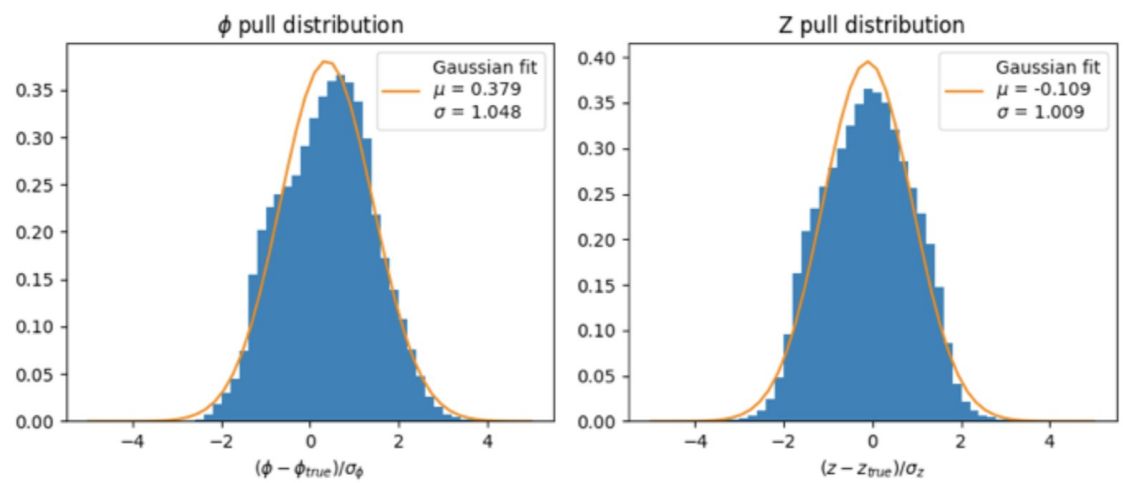
$\vec{r}_0, \vec{r}_1, \ldots, \vec{r}_{N-1}$ → LSTM → FC → $(\hat{\vec{r}}_1, \Sigma_1), (\hat{\vec{r}}_2, \Sigma_2), \ldots, (\hat{\vec{r}}_N, \Sigma_N),$

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

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 of the hit on the next layer



Track Parameters Measurement

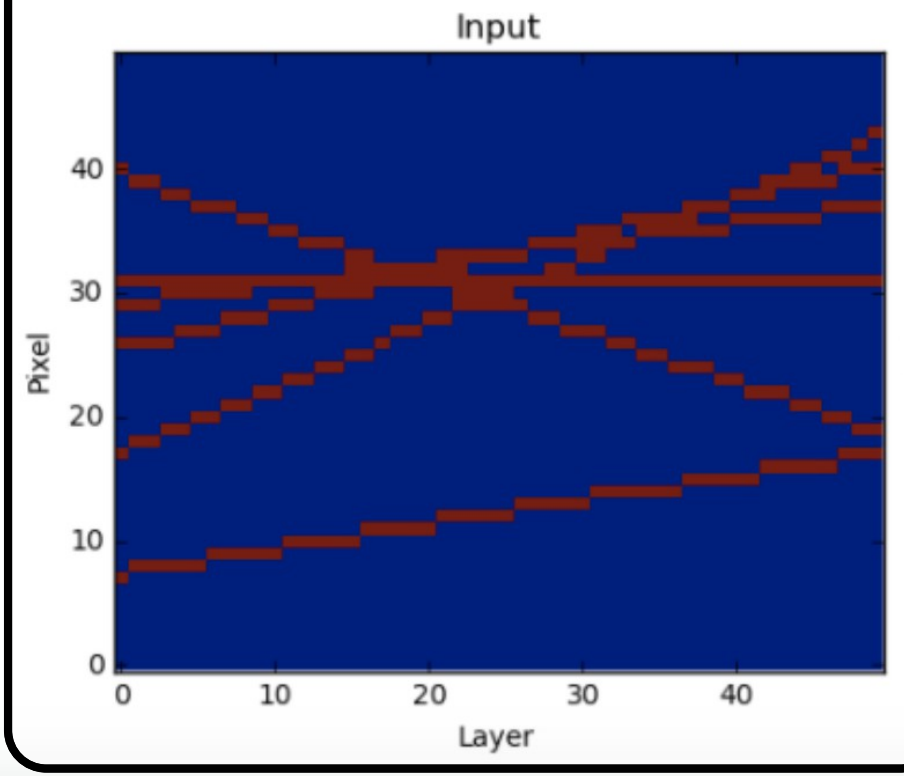
Predicting Covariance Matrix

- The observed hit pattern from multiple track processed through convolutional layers
- LSTM cells are ran multiple time in order to predict a list of particles
- Model is able to predict the covariance matrix of track parameters, incorporated in the loss function

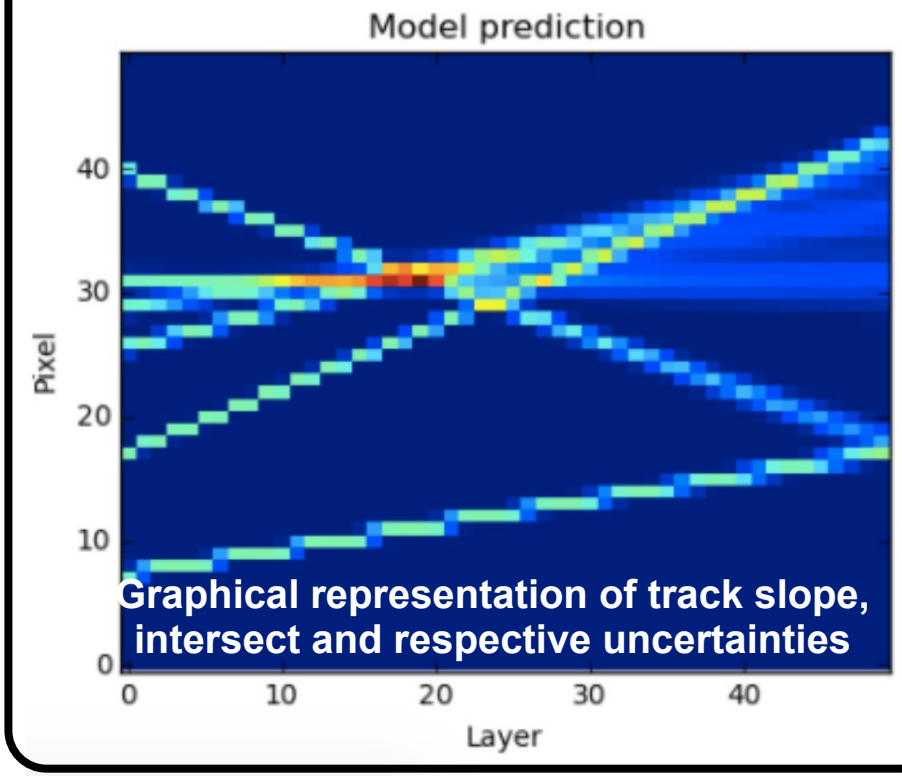
$$L(\boldsymbol{x}, \boldsymbol{y}) = \log |\boldsymbol{\Sigma}| + (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))$$

Track Parameter Prediction

Hit pattern in the detector



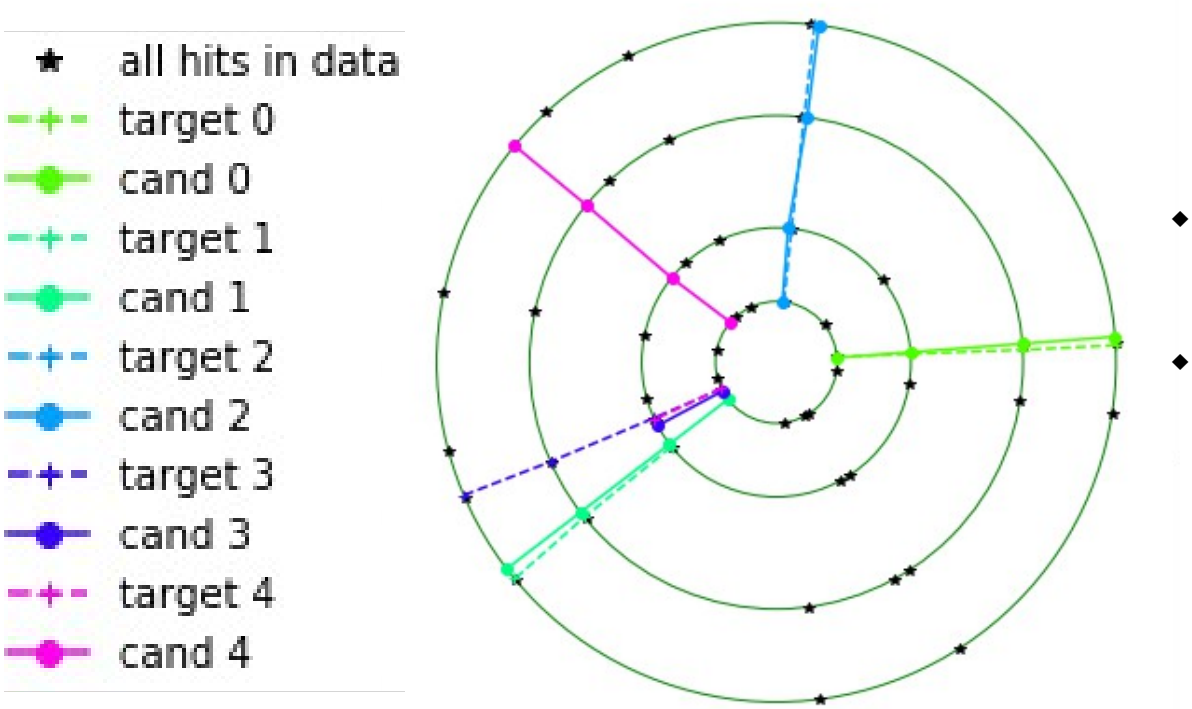
Track parameters and corresponding uncertainties



Hit Assignment Approaches

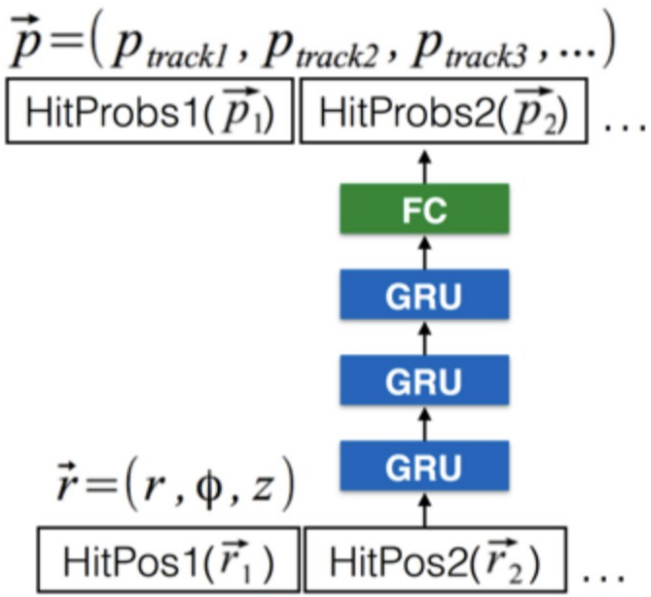
seq-2-seq tracking

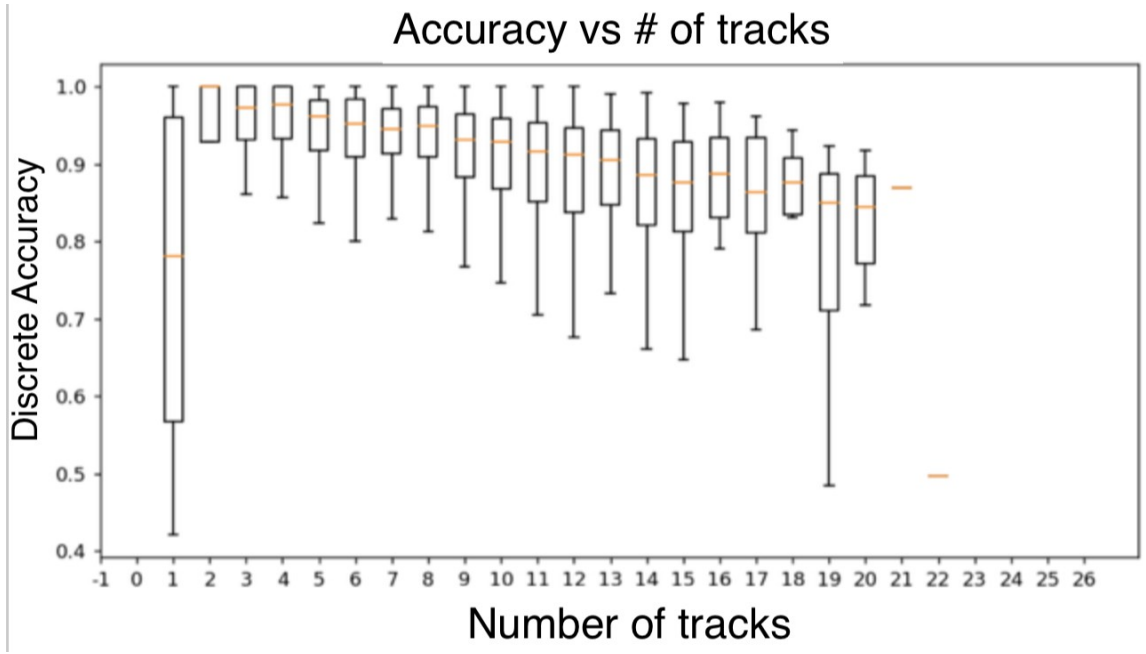
- Input sequence of hits per layers (one sequence per layer)
  - One LSTM cell per layer
- Output sequence of hits per candidates
  - Final LSTM runs for as many candidates the model can predict



- Restricted to 4 layers (with seeding in mind)
- Full performance evaluation still to be done

Hit Assignment Algorithm

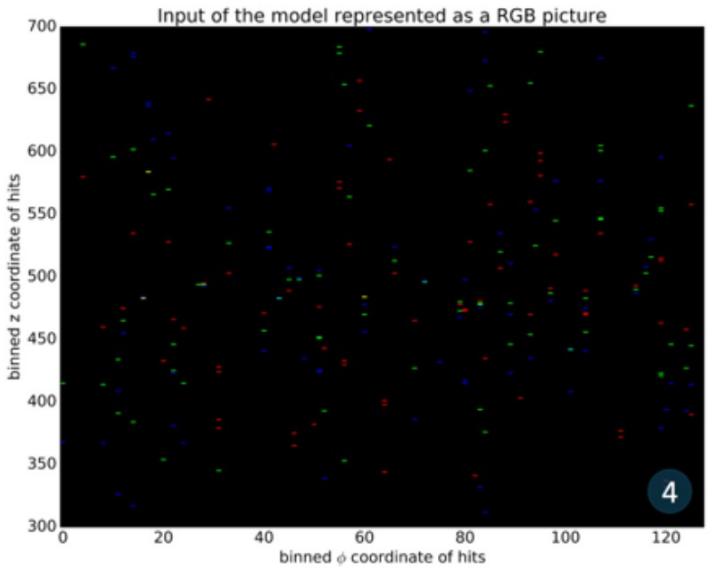


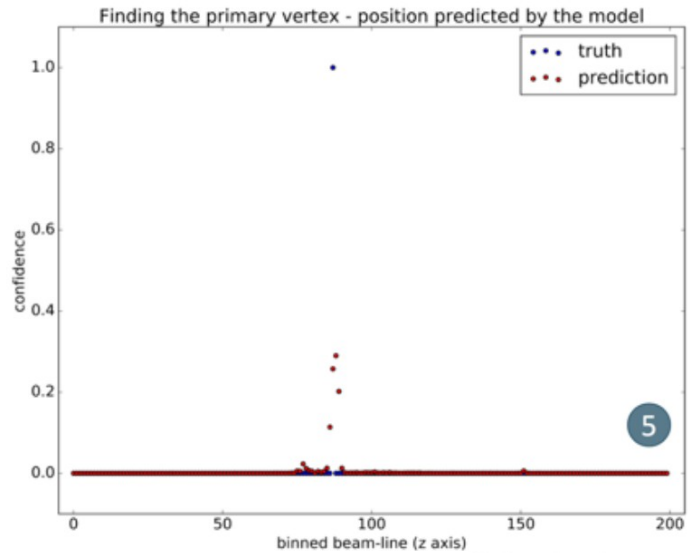


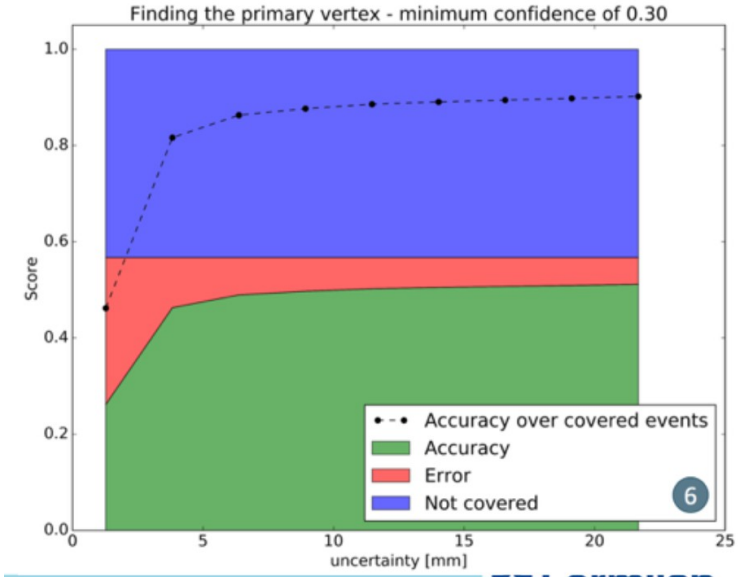
- Unseeded hit-to-track assignment (clustering)
- Hit positions taken in sequential input
- Model predicts the probability that a hit belongs to a track candidate

Vertexing

Vertexing with CNN





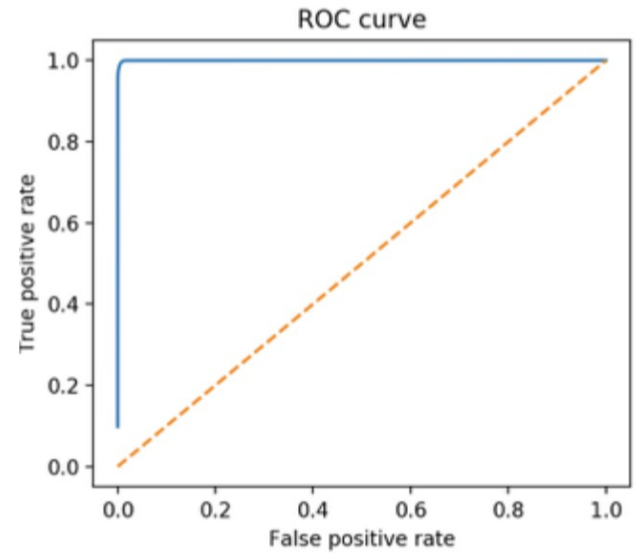


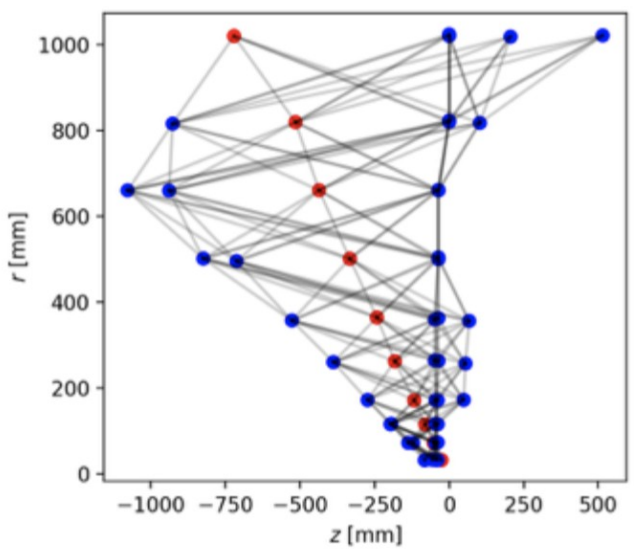
- Using hits binned (η, φ) map in input for a regression of the primary vertex position
- Modest success

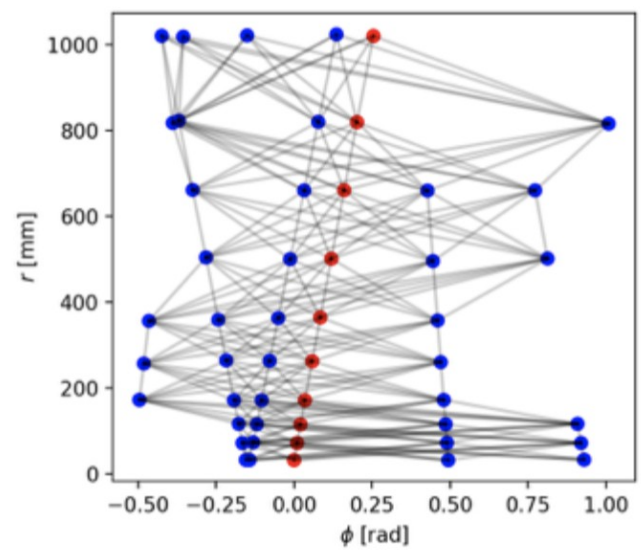
Graph Networks Approach

Seeded Hit Classification with GNN

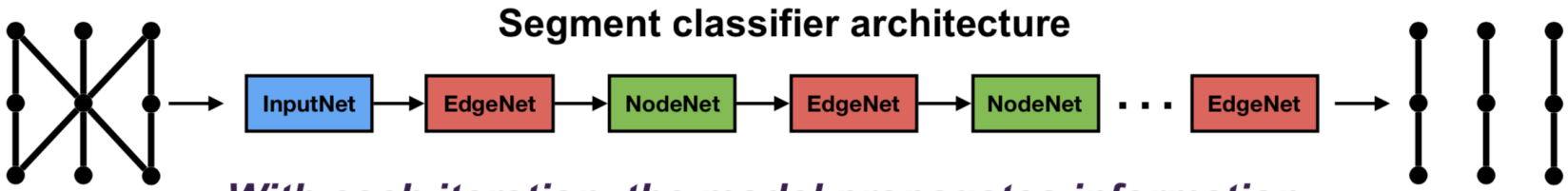
- Seeded hit classification
- Model predicts whether hits belong to the given seed





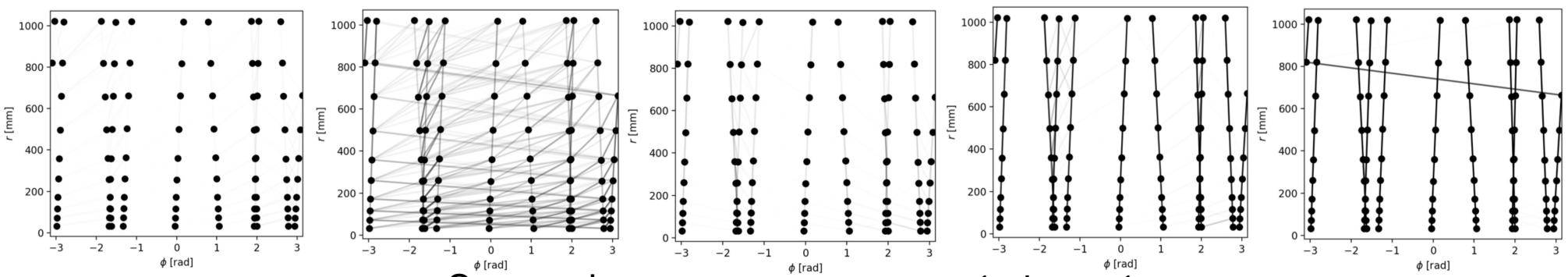


Track Building With GNN



With each iteration, the model propagates information through the graph, strengthens important connections, and weakens useless ones.

- Unseeded hit-pair classification
- Model predicts the probability that a hit-pair is valid



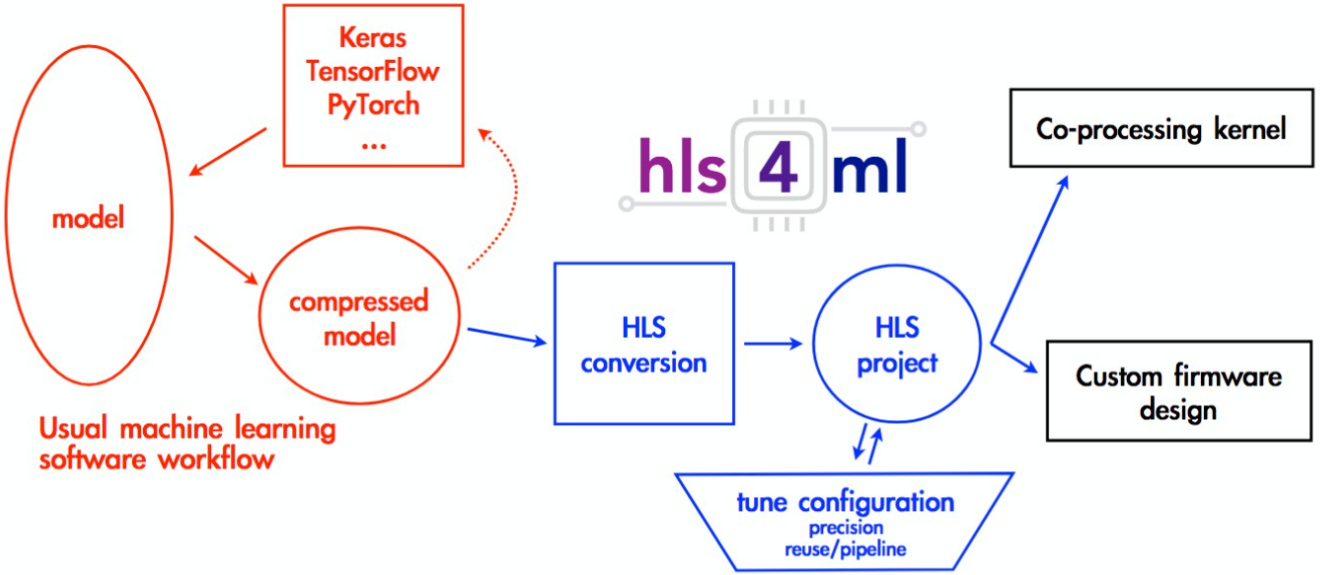
Successive iterations on a selected event

See our poster on Track 6 for more details
https://indico.cern.ch/event/587955/contributions/2937570/

Hardware Consideration

Inference on FPGA

- Demo at NIPS 2017 of implementing neural networks on FPGA
- Collaborating with hls4ml team to push the graph neural networks models to the nexts level



See Jennifer's talk during this event
https://indico.cern.ch/event/587955/contributions/2937529/

Conclusions

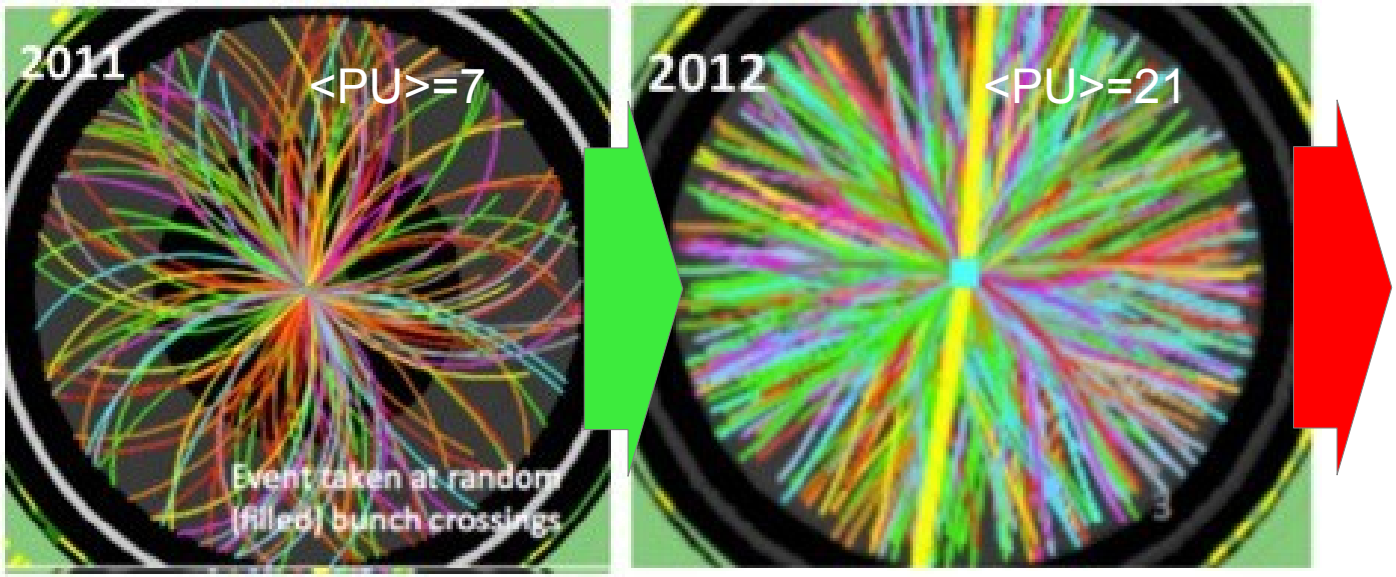
- Pilot project to explore new ideas for charged particle track reconstruction
- Cover lots of ground on new approaches, accumulated insights on applying deep learning to charged particle tracking
- Concentrating on proof of concept with graph neural networks

Acknowledgement

- Part of this work was conducted at "iBanks", the AI GPU cluster at Caltech. We acknowledge NVIDIA, SuperMicro and the Kavli Foundation for their support of "iBanks".

Backup

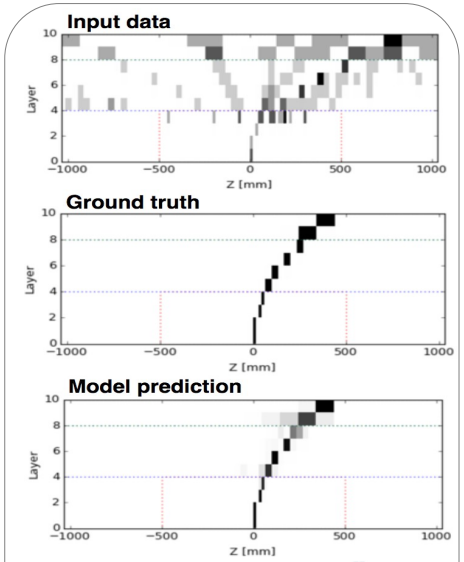
HL-LHC Challenge



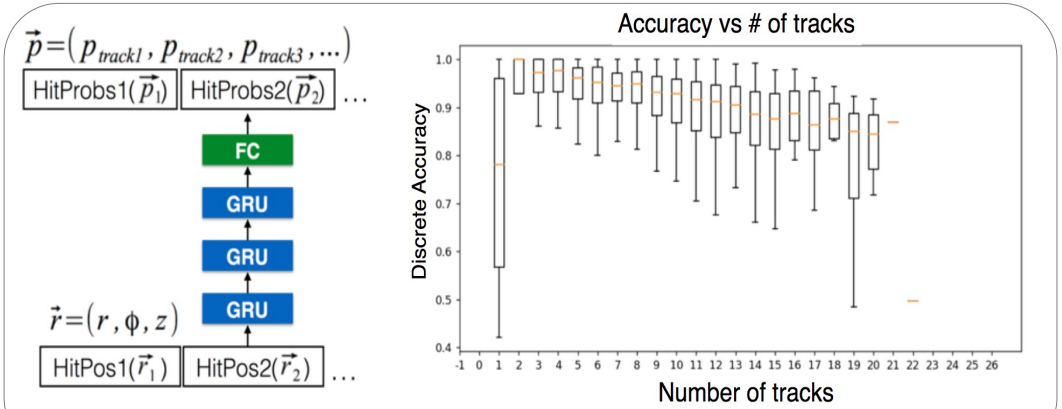
<PU>=140-200
10x more hits
Circa 2025

- CPU time extrapolation into HL-LHC era far **surpasses growth in computing budget**
- **Need for faster algorithms**
- Approximation allowed in the trigger

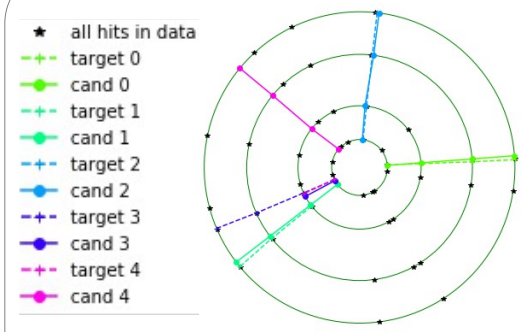
HEP.TrkX Approaches



Track following with RNN

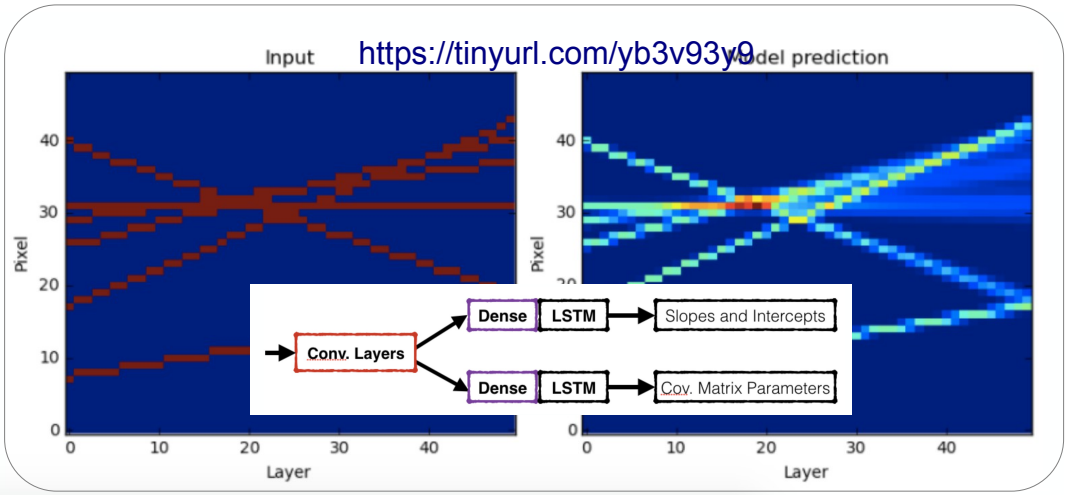


End-to-end hit assignment

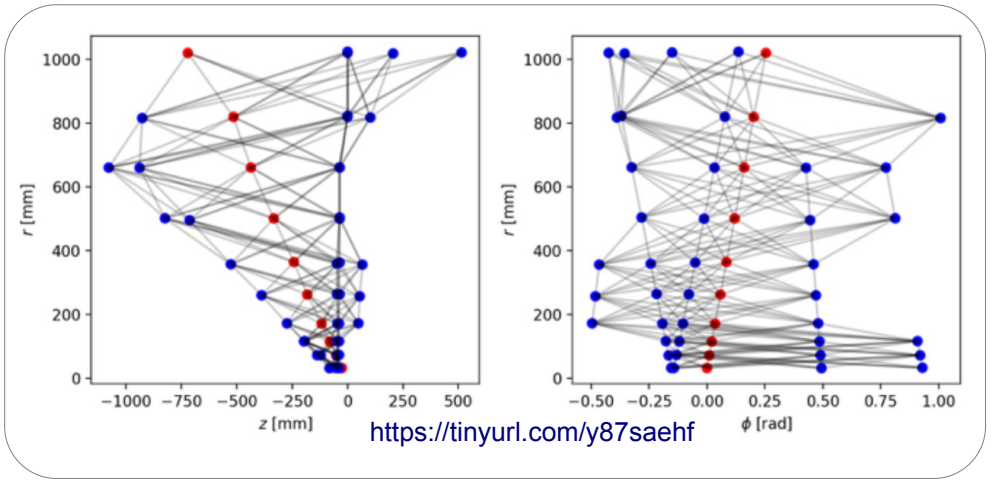


Seq-to-seq track finding

https://heptrkx.github.io/



https://tinyurl.com/yb3v93y9



https://tinyurl.com/y87saehf

Possible Application to Tracking

- **Track candidate**
  - Finding the hits that belong to a track
  - Seed + hits → tracks
- **Track parameters**
  - Measuring the physic quantity of tracks
  - Hits → track kinematics
- **Seeding**
  - Putting together hits into tracks
  - Hits → track

Datasets

A) Highly simplified model* in 2D or 3D

B) More realistic sample from
2D CtD track RAMP* (https://ctdwit2017.lal.in2p3.fr/)
3D ACTS (https://gitlab.cern.ch/acts)

C) Full simulation or real data

*** : this talk**

Scene Labeling

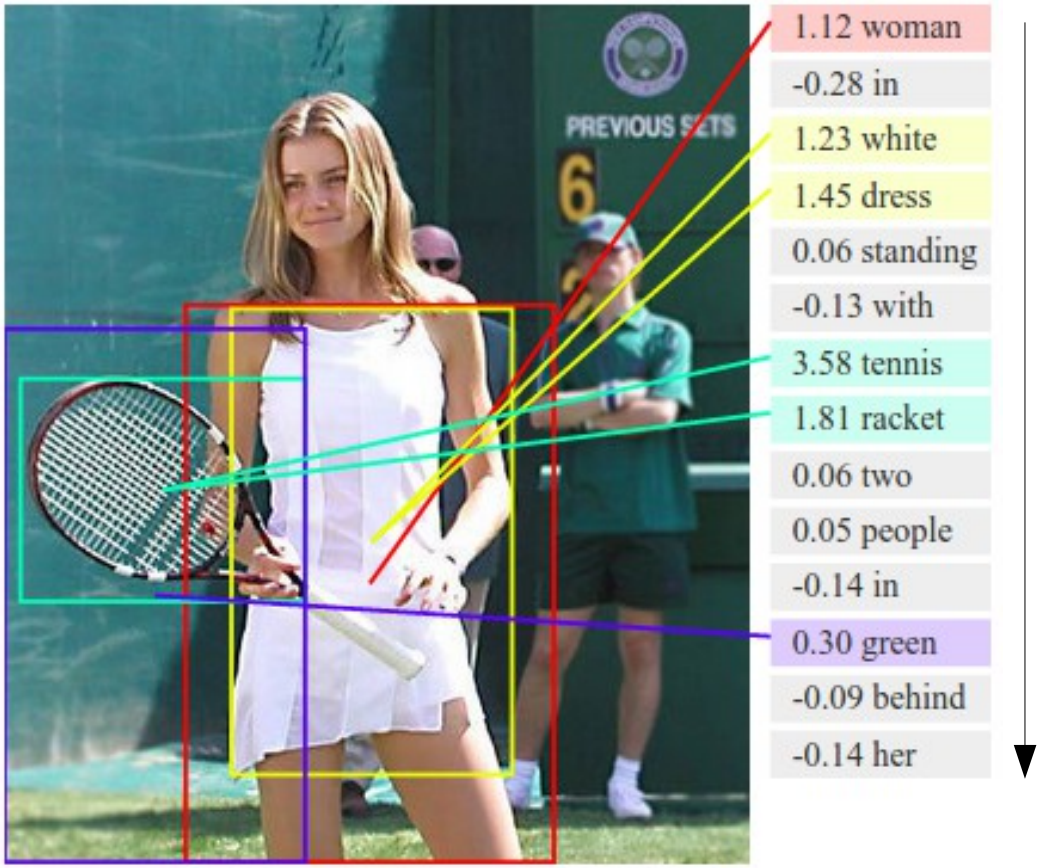
From talk of LeCunn at CERN

Scene Labeling

Farabet et al. ICML 2012, PAMI 2013

➔ Assign hits to track candidates

Scene Captioning

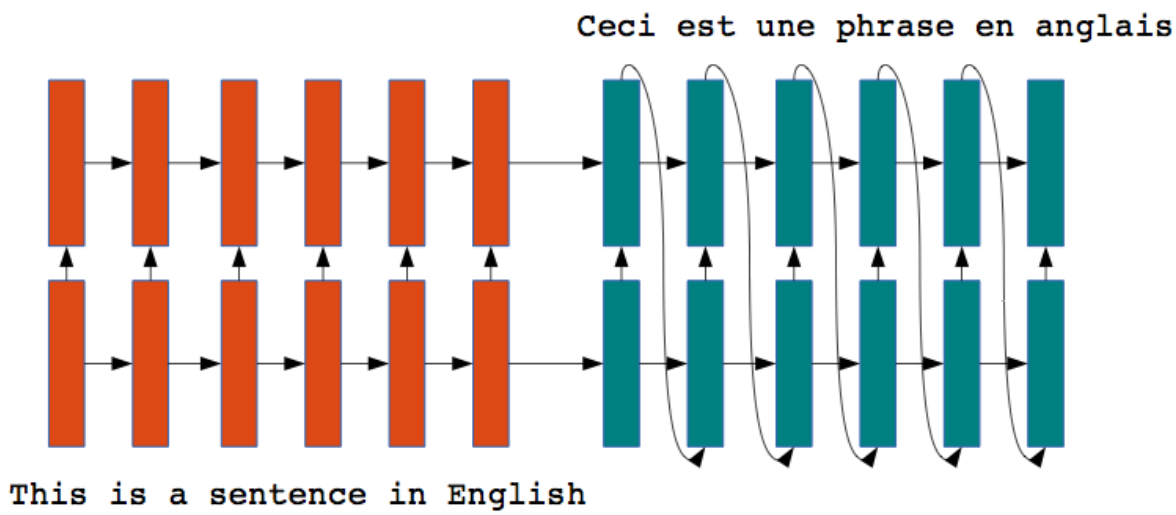


Karpathy, Fei-Fei, CVPR 2015

➔ Compose tracks explanation from image

Text Translation

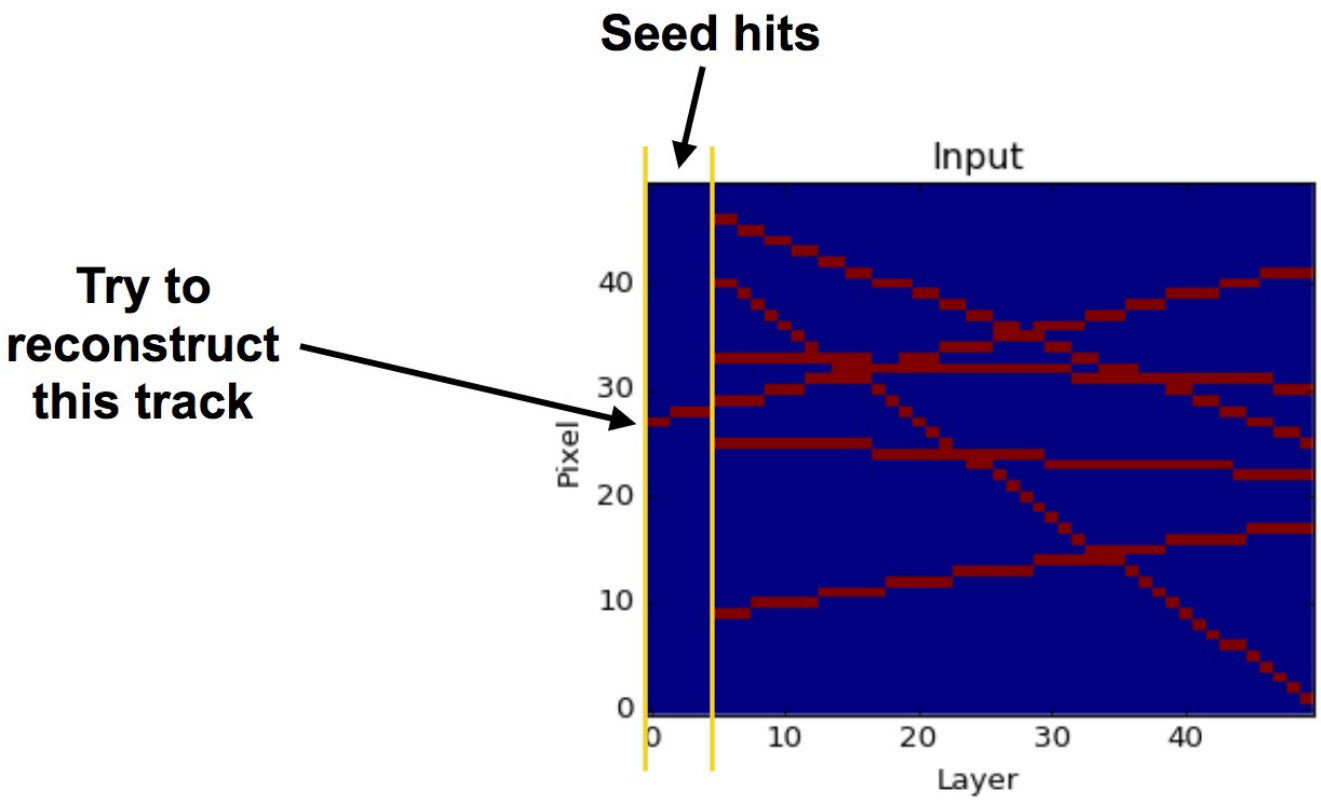
- [Sutskever et al. NIPS 2014]
  - Multiple layers of very large LSTM recurrent modules
  - English sentence is read in and encoded
  - French sentence is produced after the end of the English sentence
  - Accuracy is very close to state of the art.



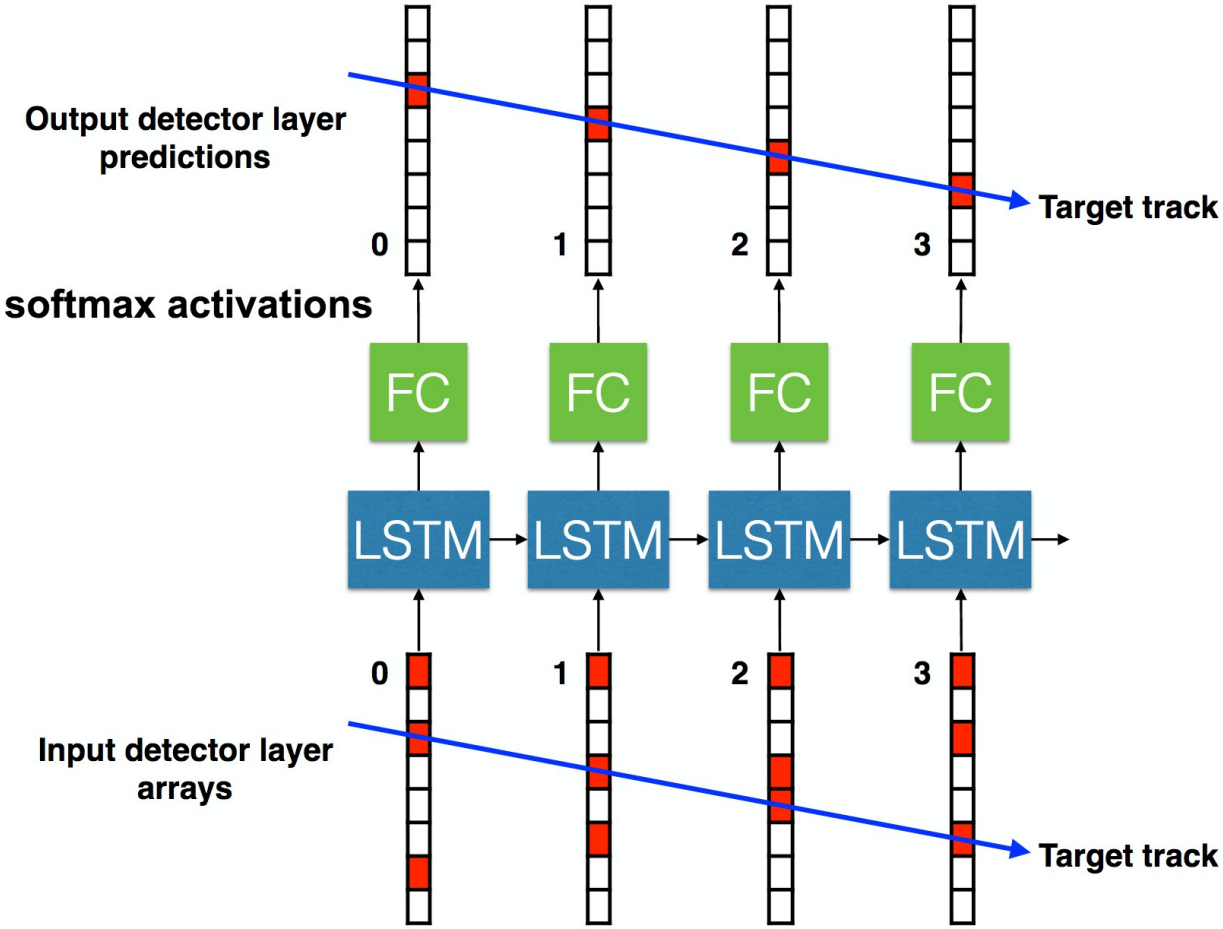
→ From sequence of hits on layer to sequence of hits on track

Seeded Pattern Prediction

- Hits on first 3 layers are used as seed
- Predict the position of the rest of the hits on all layers

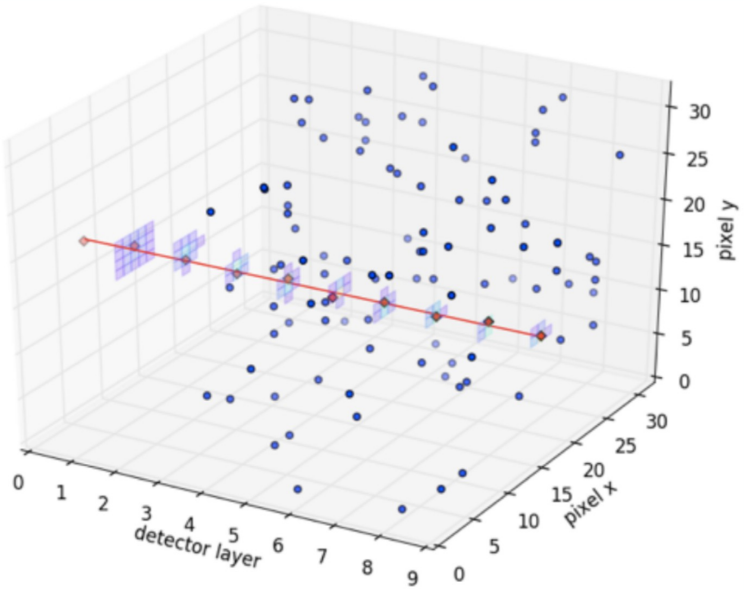


LSTM ≡ Kalman Filter

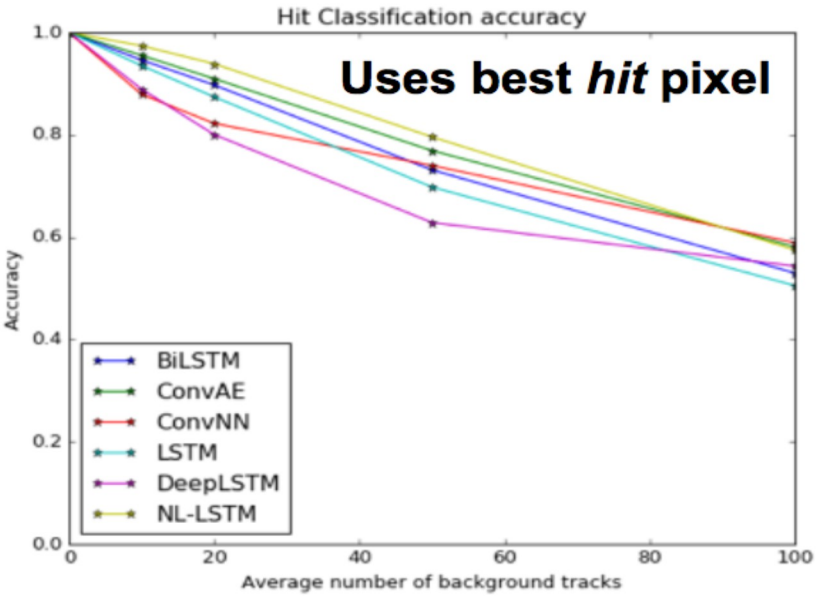


Seeded Pattern Recognition Insights

- For a simplified track models, predicting the track pattern from the seed works
  - In 2D and 3D
  - With some level of noise
  - With other tracks present
  - On layers with increasing number of pixels

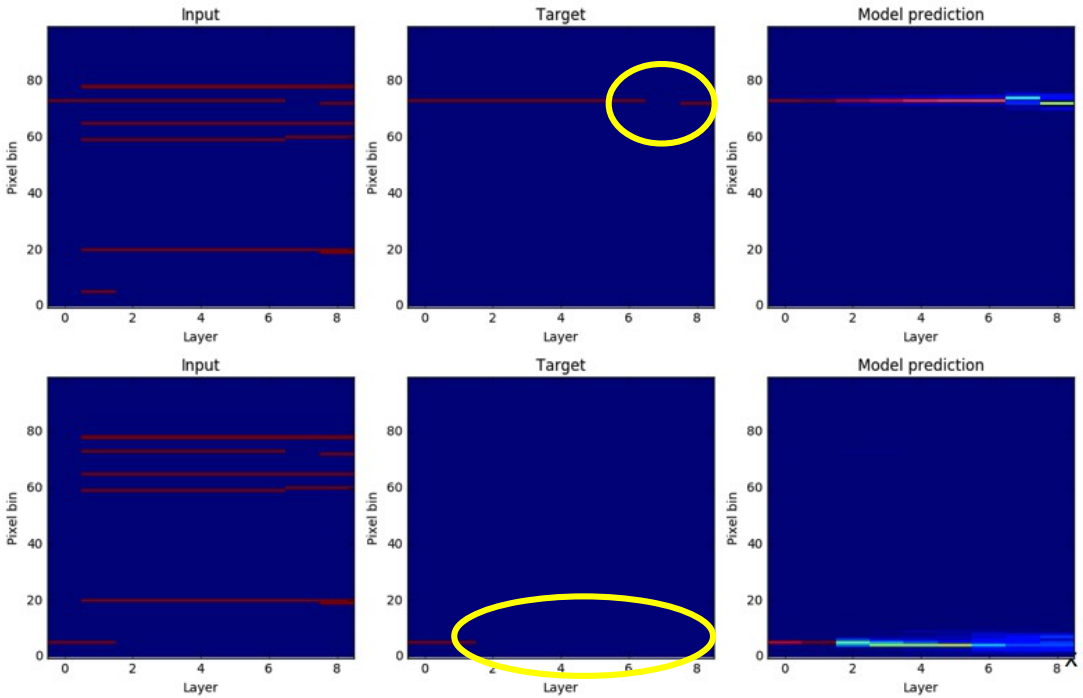


- Several other architectures tried
  - Convolutional neural nets (no LSTM)
  - Convolutional auto-encoder
  - Bi-directional LSTM
  - Prediction on next layer with LSTM

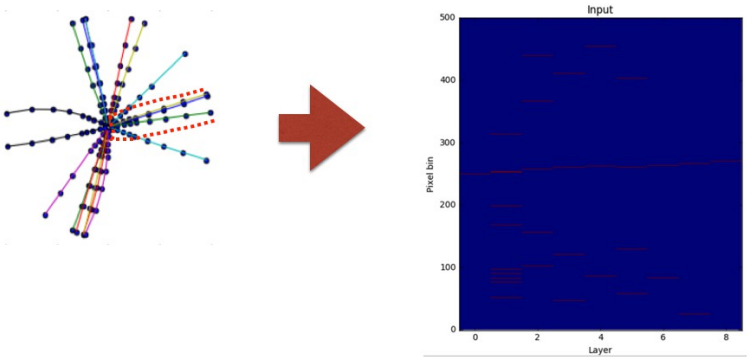


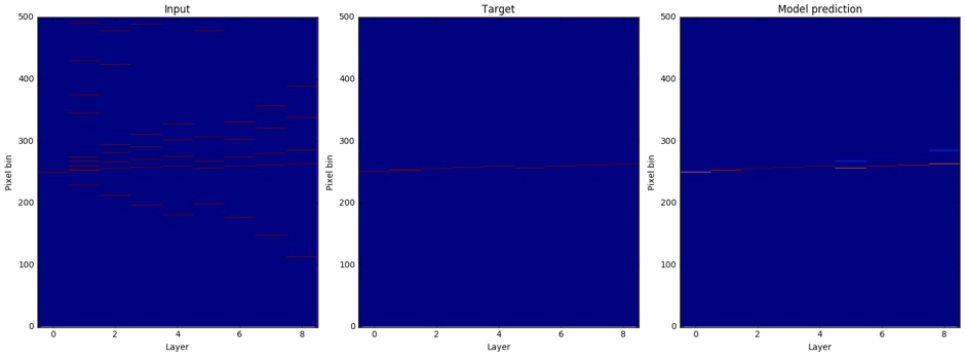
Tracking RAMP at CtD

S. Farrell : Best solution in the Machine Learning category
https://indico.cern.ch/event/577003/contributions/2509988/



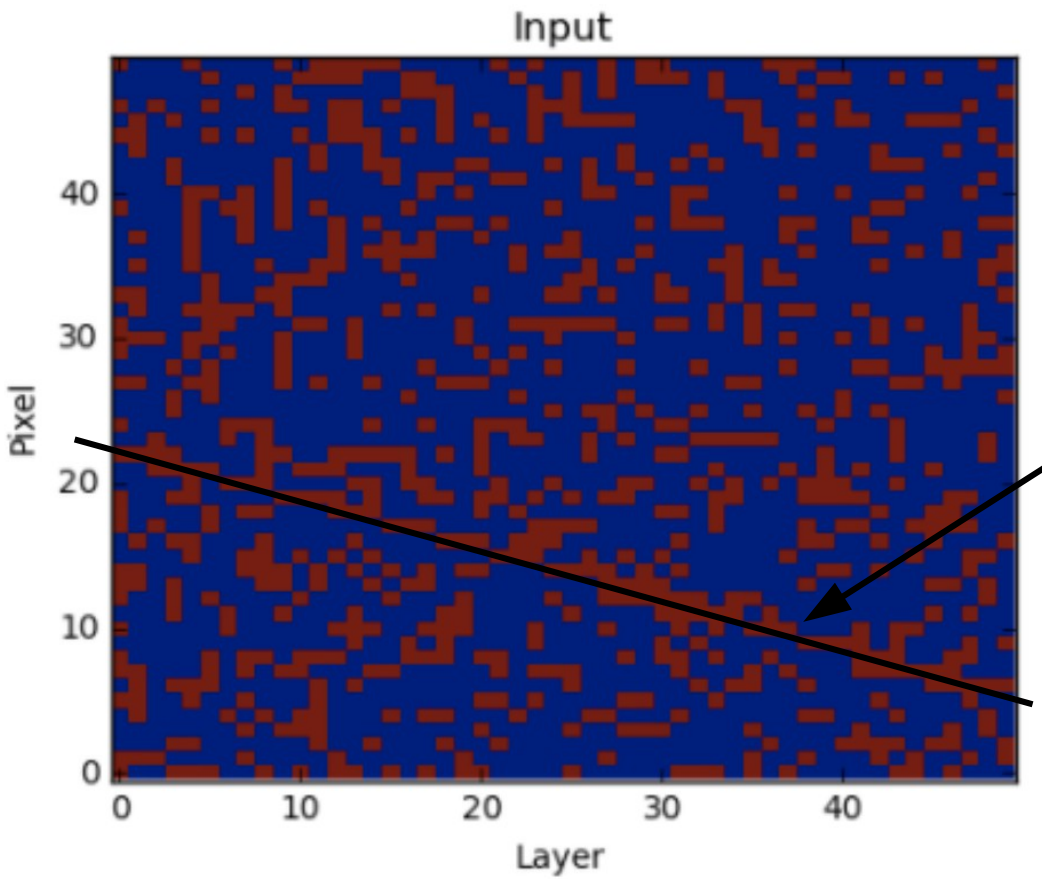
- Increased granularity in “road”
- LSTM for hit assignment
- 95% efficiency





- Down-sampling layer to 100 bins
- LSTM for hit assignment
- 92% efficiency
- Robust to holes and missing hits

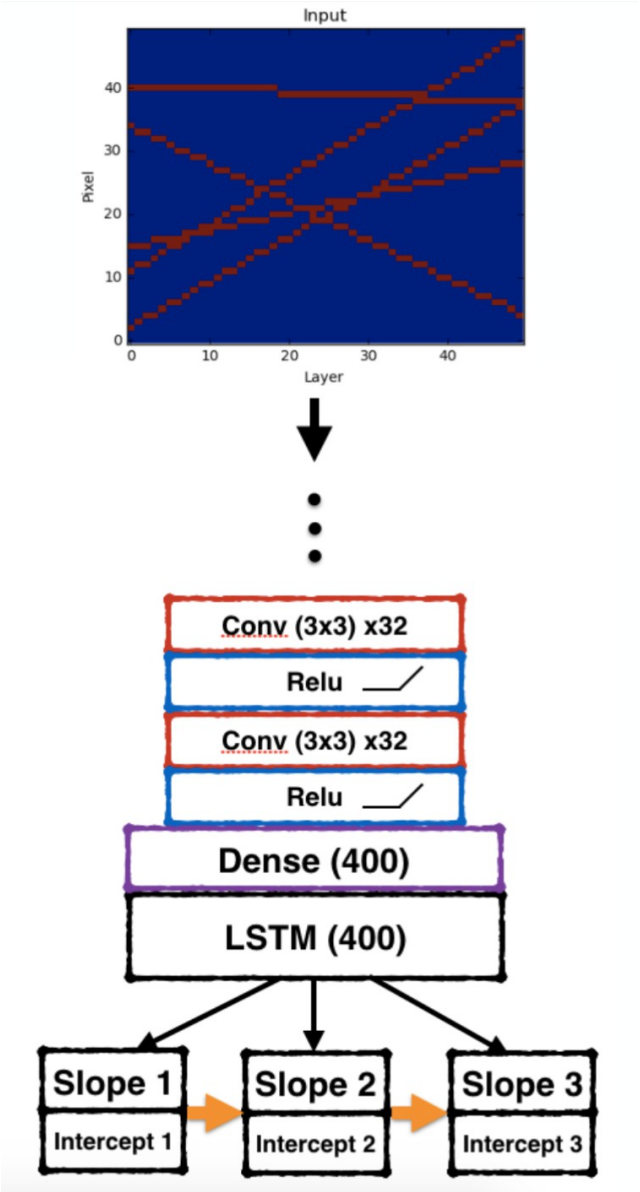
Track Parameter Estimation



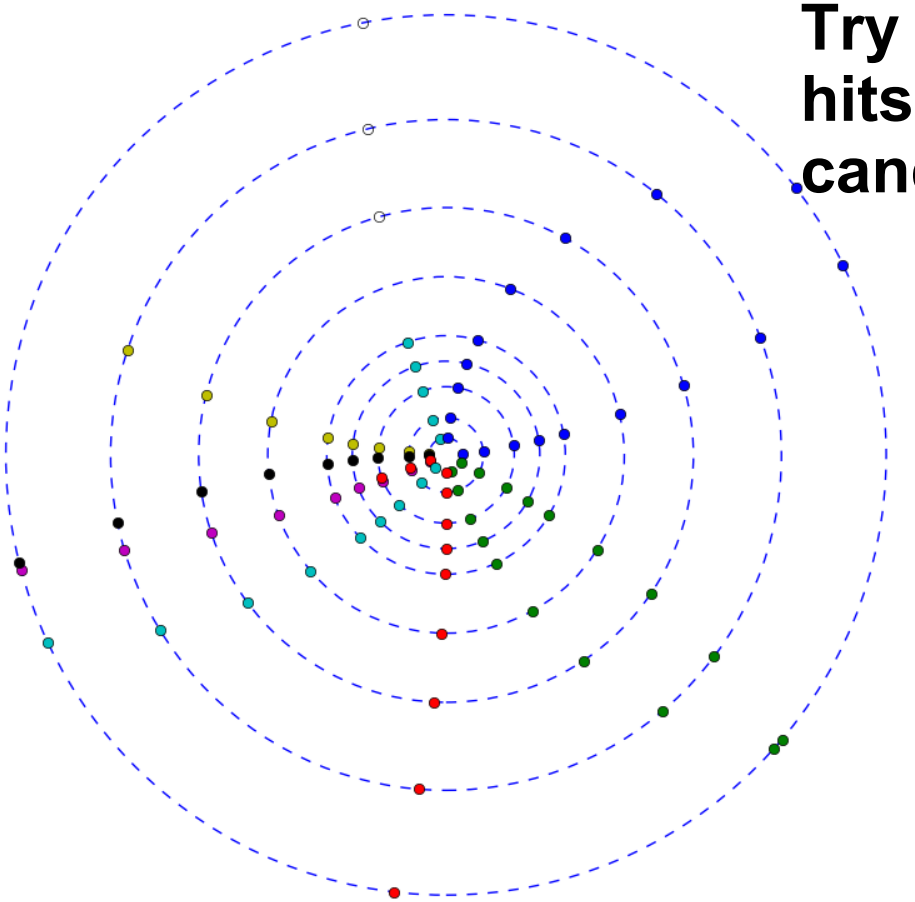
Try to predict the slope and intersect of this track

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.



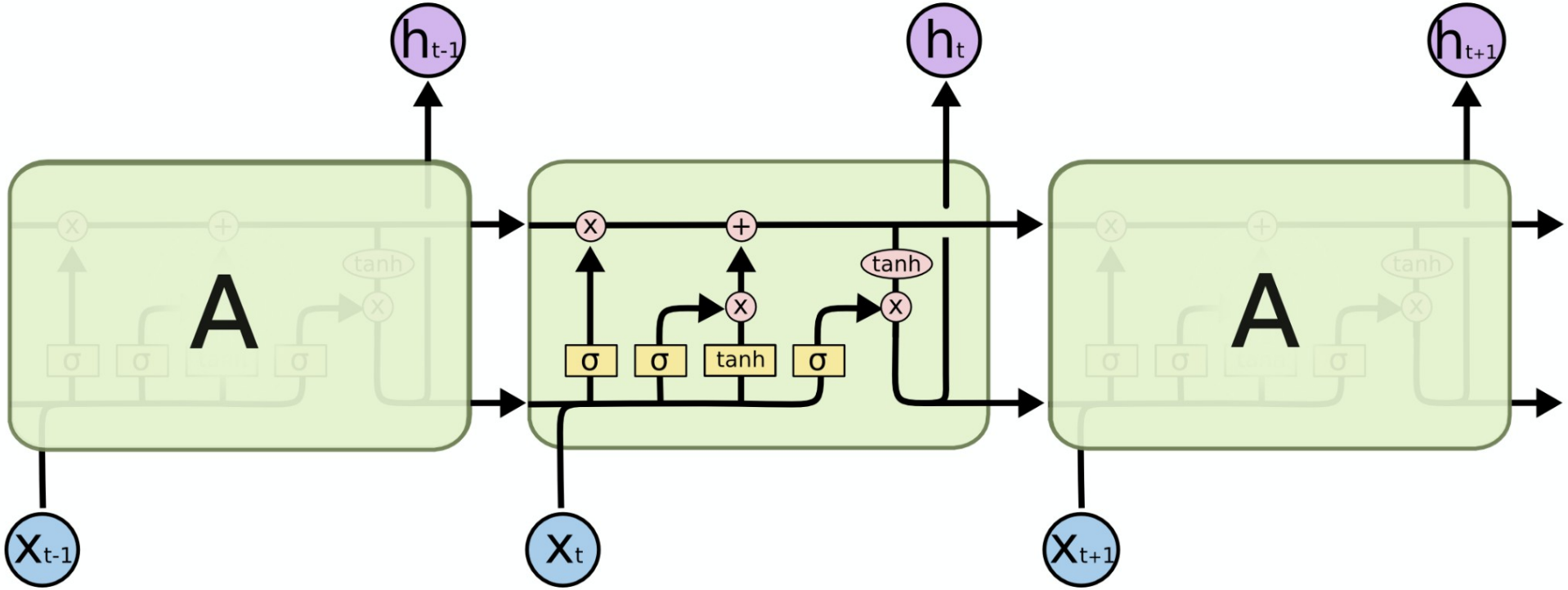
Pattern Recognition



Try to assemble hits into track candidates.

Long Short Term Memory - LSTM

Breakthrough in sequence processing by carrying over an internal state, “memory” of the previous items in the sequence, allowing for long range correlation



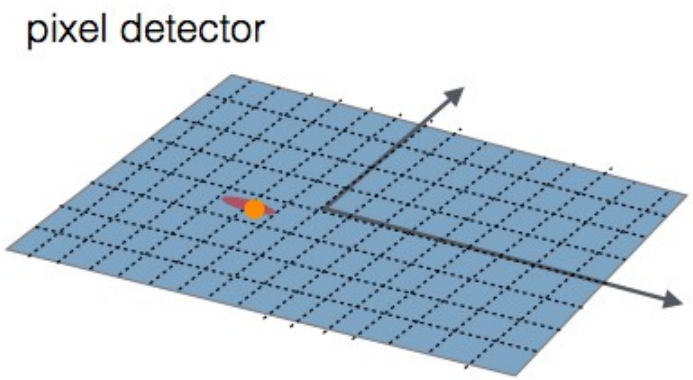
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

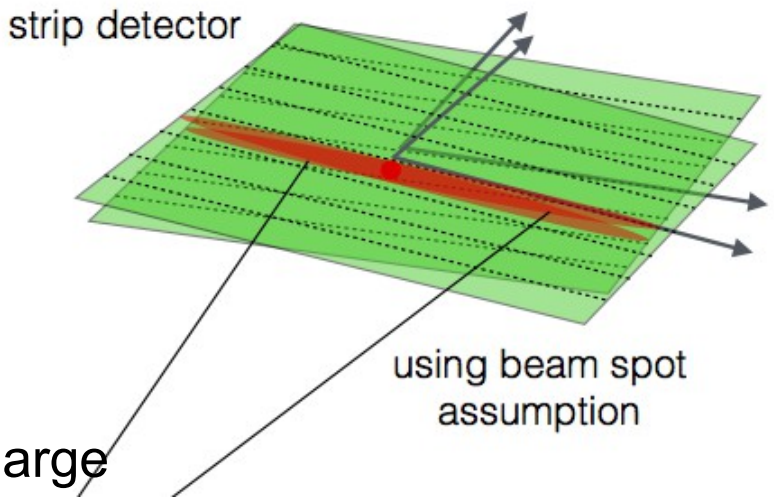
Tracking **Not** In a Nutshell

Several Times

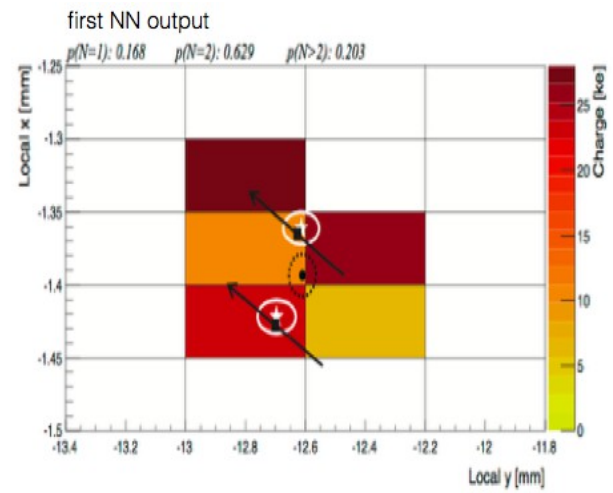
- **Hits preparation**
- **Seeding**
- **Pattern recognition**
- **Track fitting**
- **Track cleaning**

Hit Preparation



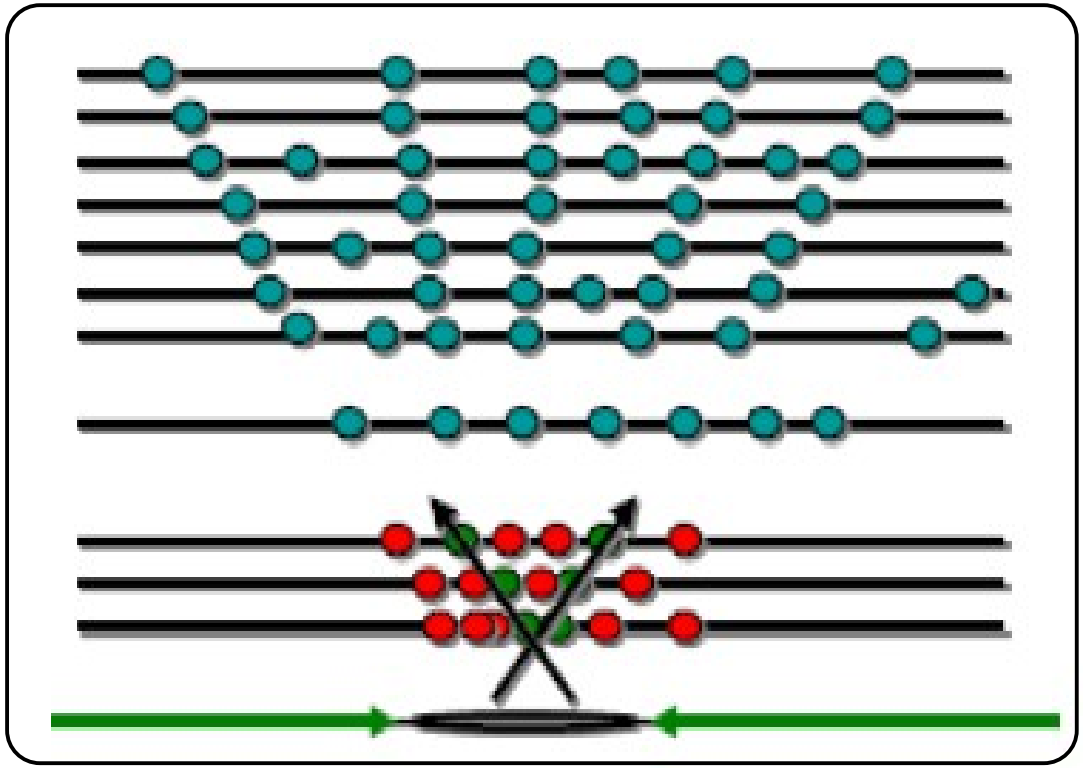


- Calculate the hit position from barycenter of charge deposits
- Use of neural net classifier to split cluster in ATLAS
- Access to trajectory local parameter from cluster shape
- Remove hits from previous tracking iterations
- HL-LHC design include double layers giving more constraints on the local trajectory parameters



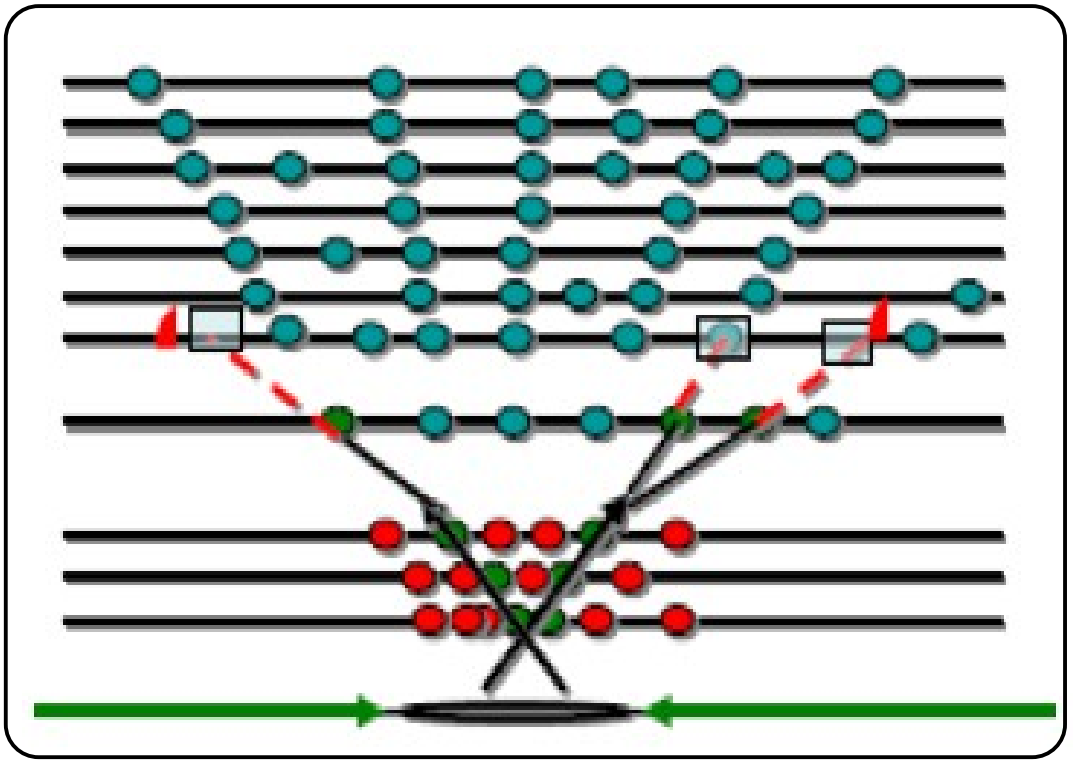
Example of cluster split

Seeding



- Combinatorics of 2 or 3 hits with tight/loose constraints to the beam spot or vertex
- Seed cleaning/purity plays in an important in reducing the CPU requirements of sub-sequent steps
  - ➔ Consider pixel cluster shape and charge to remove incompatible seeds
- Initial track parameters from helix fit

Pattern Recognition



- Use of the Kalman filter formalism with weight matrix
- Identify possible next layers from geometrical considerations
- Combinatorics with compatibles hits, retain N best candidates
- No smoothing procedure
- Resilient to missing modules
- Hits are mostly belonging to one track and one track only
- Hit sharing can happen in dense events, in the innermost part
- Lots of hits from low momentum particles

Kalman Filter

$$K_k = C_{k|k-1} H_k^\top (V_k + H_k C_{k|k-1} H_k^\top)^{-1}$$

$$p_{k|k} = p_{k|k-1} + K_k (m_k - H_k p_{k|k-1})$$

$$C_{k|k-1} = (I - K_k H_k) C_{k|k-1}$$

H_k is the projection matrix

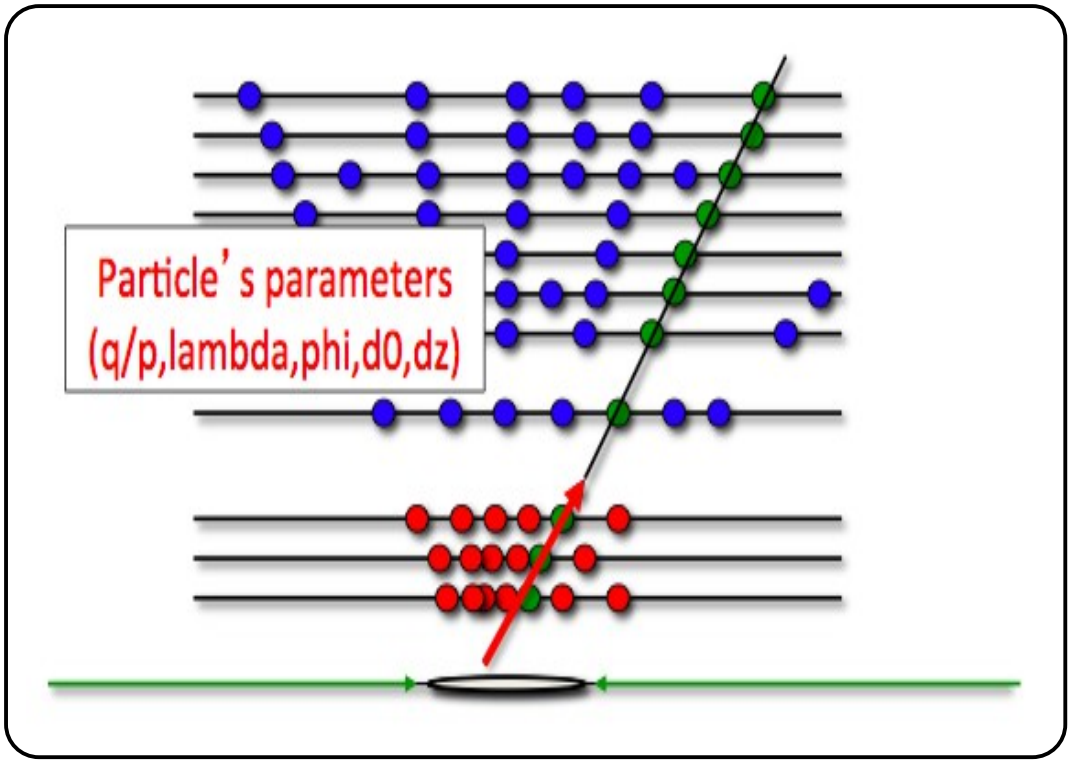
V_k is the hit covariance matrix

$p_{i|j}$ is the trajectory state at i given j

$C_{i|j}$ is the trajectory state covariance matrix at i given j

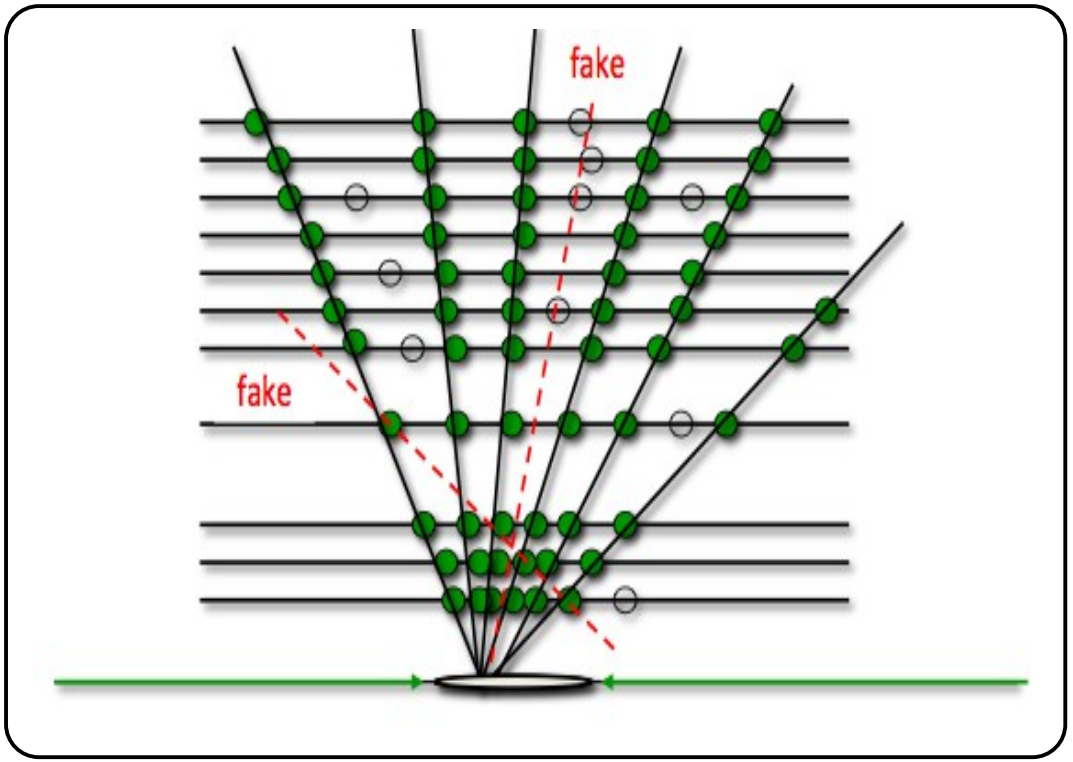
- Trajectory state propagation done either
  - ✓ Analytical (helix, fastest)
  - ✓ Stepping helix (fast)
  - ✓ Runge-Kutta (slow)
- Material effect added to trajectory state covariance
- Projection matrix of local helix parameters onto module surface
  - ➔ Trivial expression due to local helix parametrisation
- Hits covariance matrix for pixel and stereo hits properly formed
  - ✗ Issue with strip hits and longitudinal error being non gaussian (square)

Track Fitting



- Use of the Kalman filter formalism with weight matrix
- Use of smoothing procedure to identify outliers
- Field non uniformity are taken into account
- Detector alignment taken into account

Cleaning, Selection



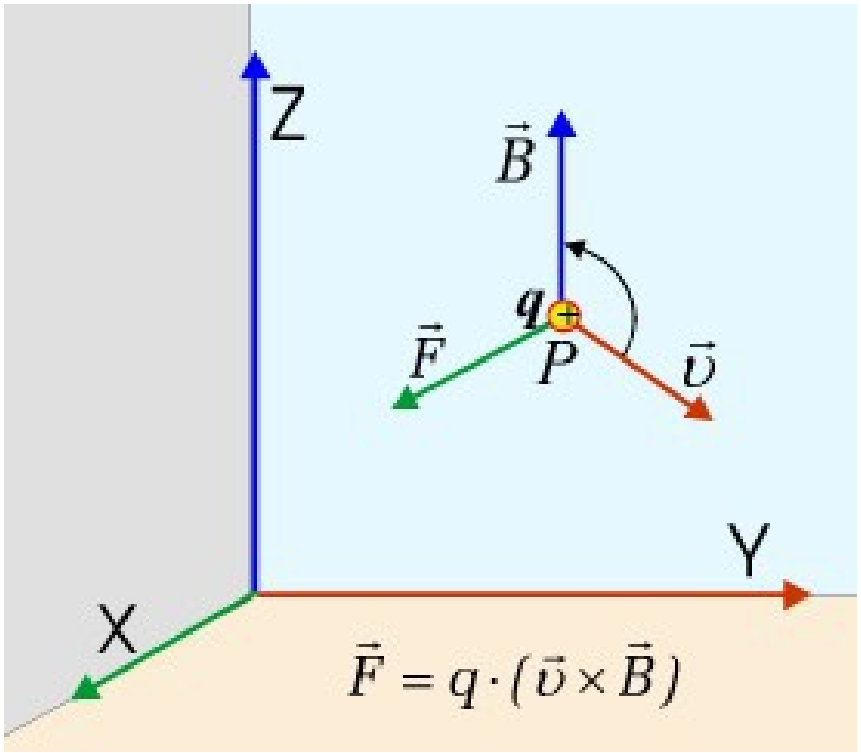
- Track quality estimated using ranking or classification method
  → Use of MVA
- Hits from high quality tracks are remove for the next iterations where applicable

A Charged Particle Journey

First order effect : electromagnetic elastic interaction of the charge particle with nuclei (heavy and multiply charged) and electrons (light and single charged)

Second order effect : inelastic interaction with nuclei.

Magnetic Field

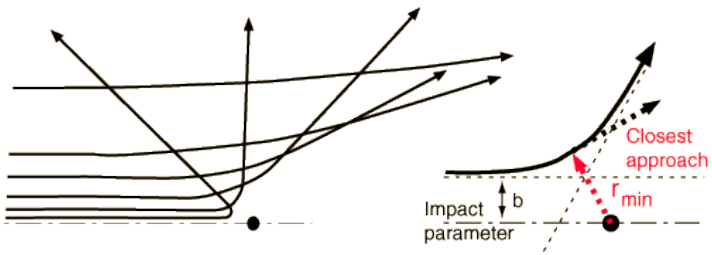


- Magnetic fieldB acts on charged particles in motion : Lorentz Force
- The solution in uniform magnetic field is an helix along the field : 5 parameters
- Helix radius proportional to the component of momentum perpendicular to B
- Separate particles in dense environment

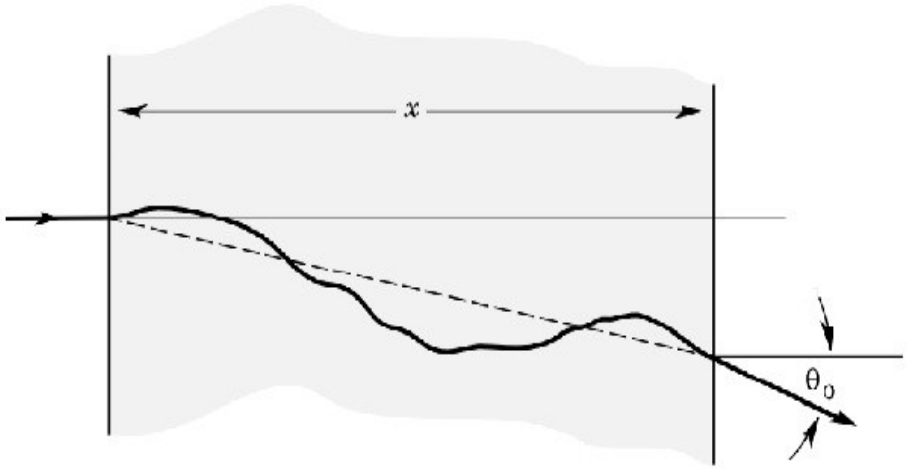
➔ Bending induces radiation : bremsstrahlung

➔ The magnetic field has to be known to a good precision for accurate tracking of particle

Multiple Scattering



- **Deflection on nuclei** (effect from electron are negligible)
- Addition of scattering processes
- Gaussian approximation valid for substantial material traversed



Gaussian Approximation

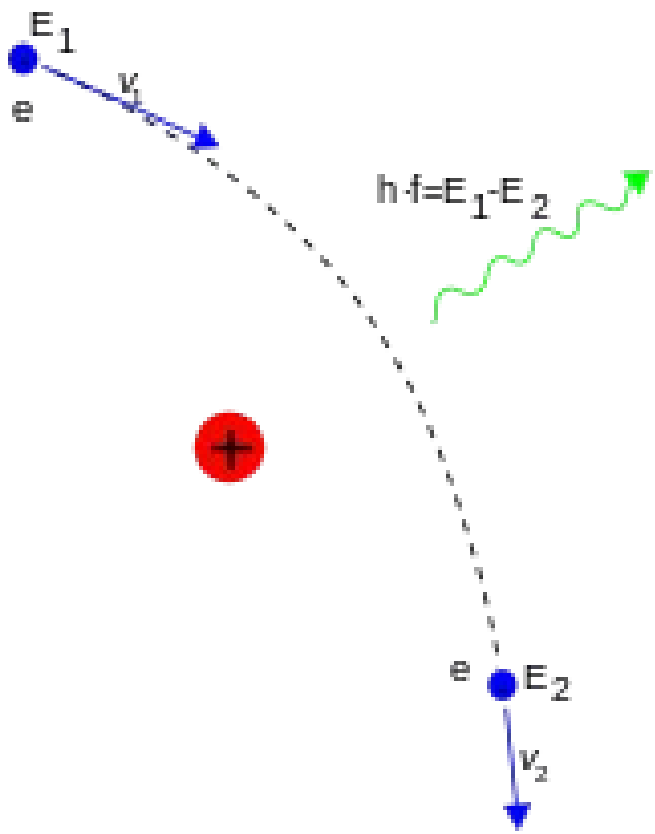
$$\theta^2 = \left(\frac{13.6MeV}{\beta cp}\right)^2 * \frac{x}{X_0}$$

β -particle velocity

ρ – material density

P - particle momenta

Bremsstrahlung



- Electromagnetic radiation of charged particles under acceleration due to nuclei charge
- Significant at low mass or high energy
- Discontinuity in energy loss spectrum due to photon emission and track curvature

➔ Can be observed as kink in the trajectory or presence of collinear energetic photons

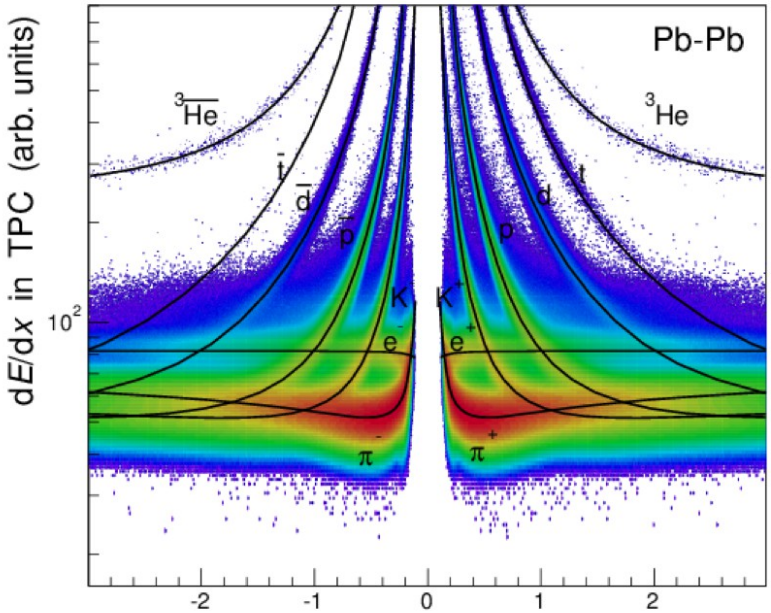
Energy Loss

- Momentum transfer to electrons when traversing material (effect of nuclei is negligible

$$dE/dx = k_1 \frac{Z}{A}\frac{1}{\beta^2}\rho\left(\ln\left(\frac{2m_e c^2\beta^2}{I(1-\beta^2)}\right)-\beta^2-\frac{\delta}{2}\right)$$

- Energy loss at low momentum depends on mass : can be used as mass spectrometer

β -particle velocity
ρ – material density
Z - atomic number of absorber
A – mass number of absorber
I – mean excitation energy
δ – density effect correction factor – material dependent and β dependent



ALICE Experiment

Summary on Material Effects

- Collective effects can be estimated statistically and taken into account in how they modify the trajectory
- Bremstrahlung and nuclear interactions significantly distort trajectories