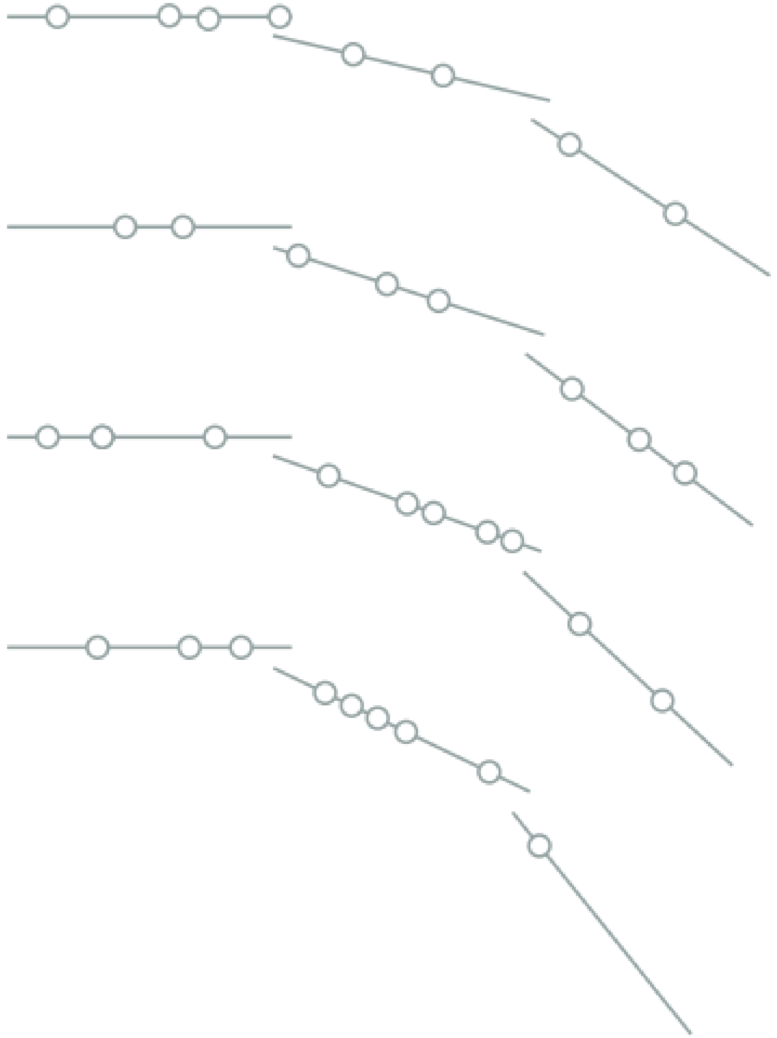
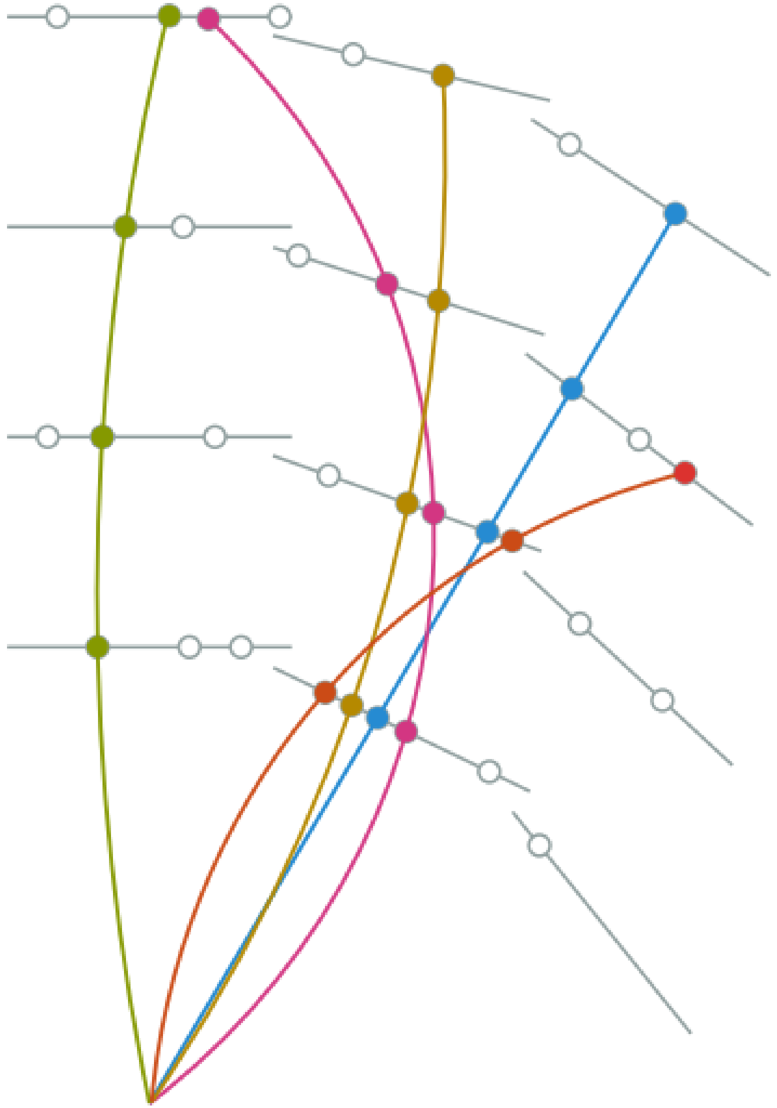
# Particle tracking

Moritz Kiehn
Université de Genève, DPNC
BTTB8, Tbilisi, January 2020

UNIVERSITÉ DE GENÈVE





Obviously!

# Overview



Basic ingredients

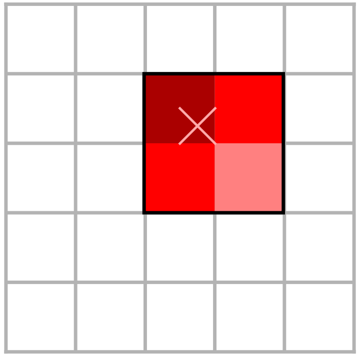
Classical algorithms

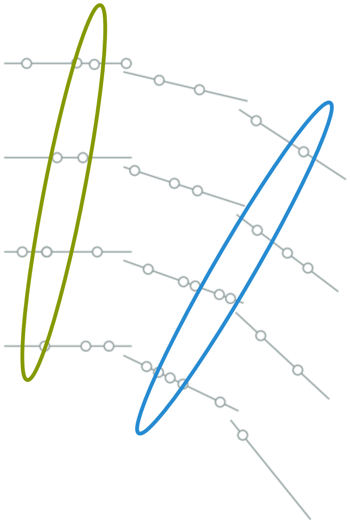
High luminosity and machine learning

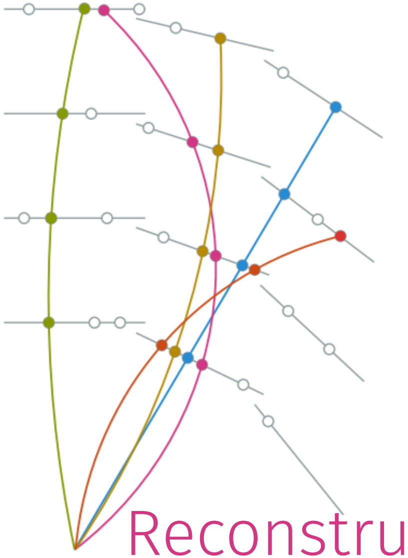
Disclaimer:
This is a personal selection

# Typical reconstruction chain









Reconstruction

## Local Reco

Bits to hits,
clustering

## Track Finding

Group hits to tracks

## Track Fitting

Estimate parameters

## Analysis

# Uncertainty and significance



Significance of marked point?

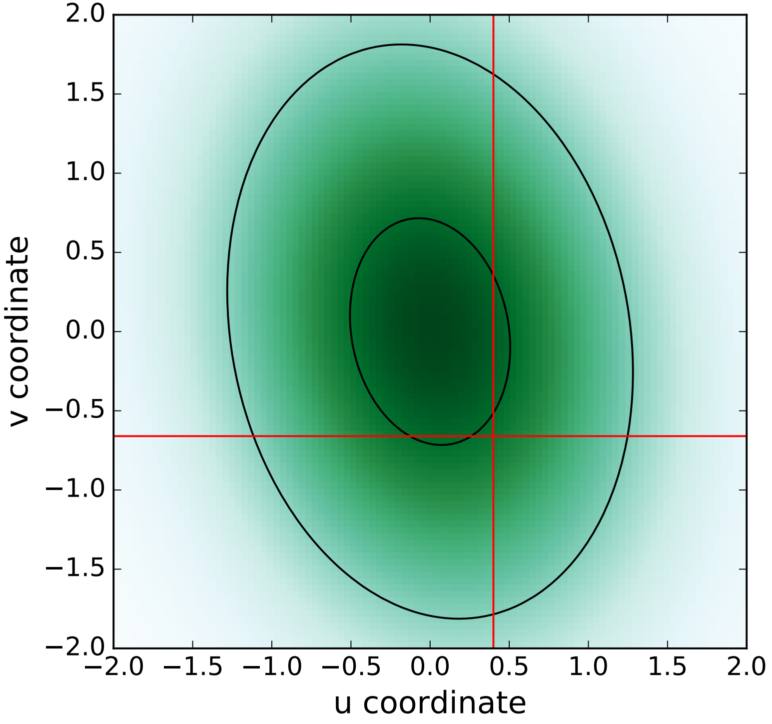
$$s_u = \frac{u}{\sigma_u} < 1 \qquad s_v = \frac{v}{\sigma_v} < 1$$

Ignores correlations

$$\Sigma = \begin{pmatrix} \sigma_u^2 & c_{uv} \\ c_{uv} & \sigma_v^2 \end{pmatrix}$$

Use Nd generalization instead

$$s = \sqrt{\left(\vec{x}^T \Sigma^{-1} \vec{x}\right)}$$



# Track model



Propagated track parameter

Initial track parameter

Propagation length

Stochastic component w/ covariance Q

$$\vec{a} = f(\vec{a}_0, l) + \sigma(Q)$$

Linearize

$$\vec{a} \approx \vec{f}_0 + F\,\overrightarrow{\Delta a_0} + \sigma(Q)$$

Typical parameters w/ reference surface

Forward $(x, y, t, x', y', \beta^{-1})$

Collider $(d_0, z_0, t_0, \phi, \theta, q/p)$

w/o reference surface

$x, y, z, t, p_x, p_y, p_z, q$

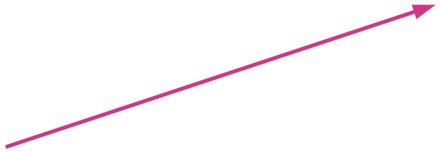
# Deterministic/ stochastic components



## Deterministic trajectory

### No magnetic field

Straight (analytic)

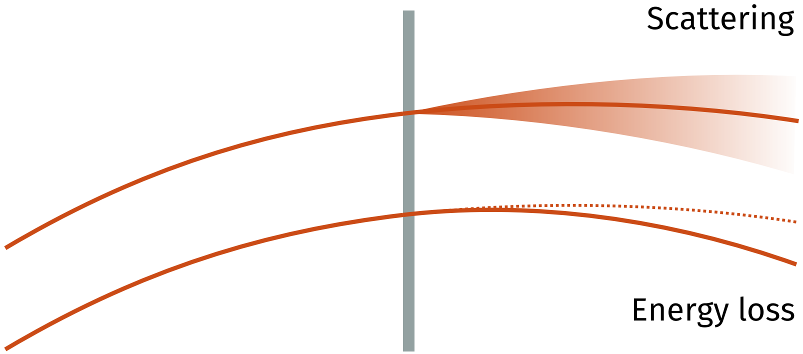


### With magnetic field

Helix for const (analytic)

Helical for non-const (numeric)

## Stochastic material interactions

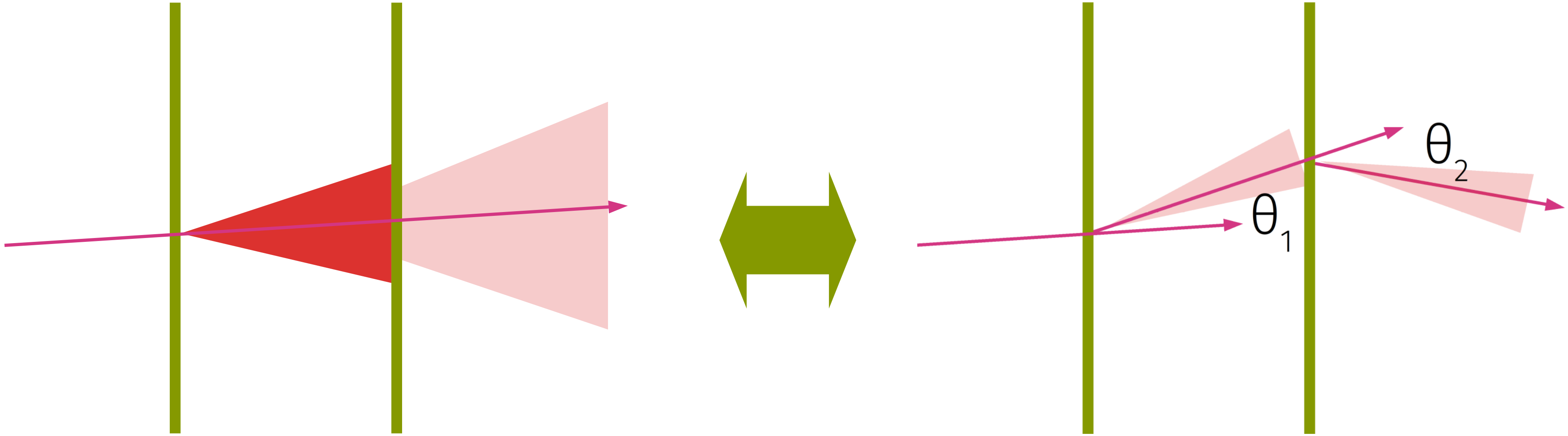



Particle and material dependent

# Possible trade-offs



## Example: multiple scattering




Simple trajectory,
Correlated covariance

Explicit parameters
Simplified covariance

# Measurement model



Position in measurement frame

Track parameters at intersection

Stochastic measurement uncertainty w/ covariance V

$$\vec{m} = h(\vec{a}) + \sigma(V)$$

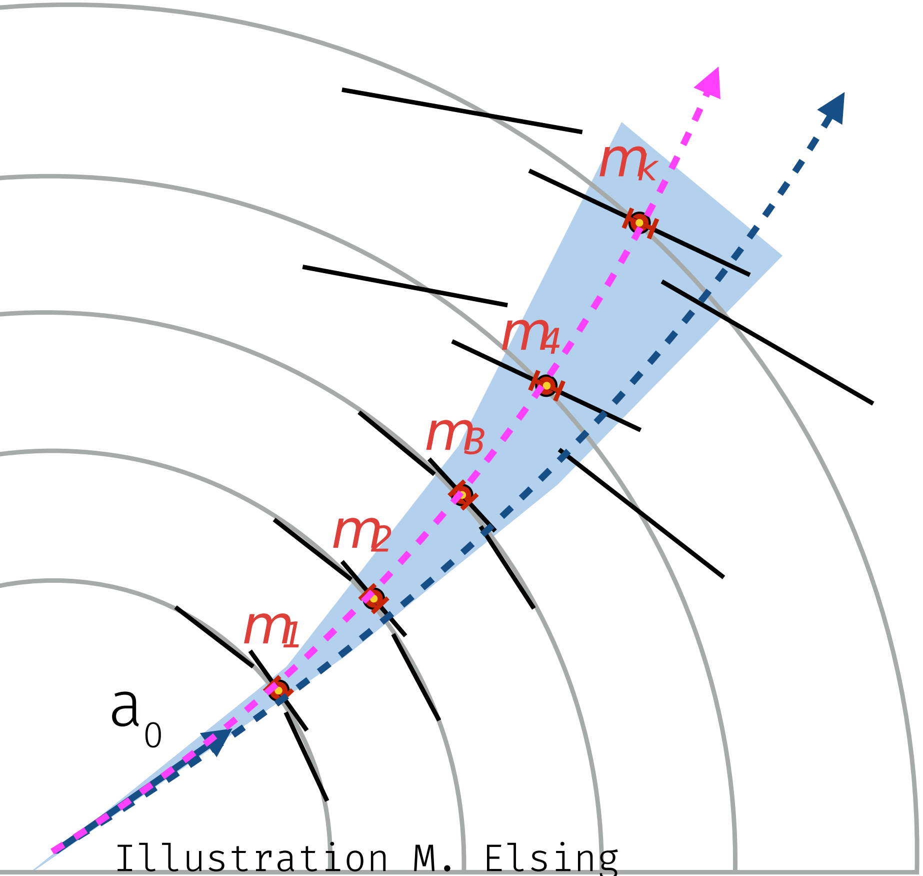
Linearize

$$\vec{m} \approx \vec{h_0} + H\,\overrightarrow{\Delta a} + \sigma(V)$$

Typically:
$h_0$ vanishes, H rotation/projection

# Least squares fitting






Define residuals

$$\vec{r}_i = \vec{y}_i - h_i(f_i(\vec{a}_0))$$

Combine components

$$\vec{r} = \begin{pmatrix} \vec{r}_0 \\ \vec{r}_1 \\ \vdots \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \Sigma_{00} & \Sigma_{01} & \cdots \\ \Sigma_{01} & \Sigma_{11} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Minimize

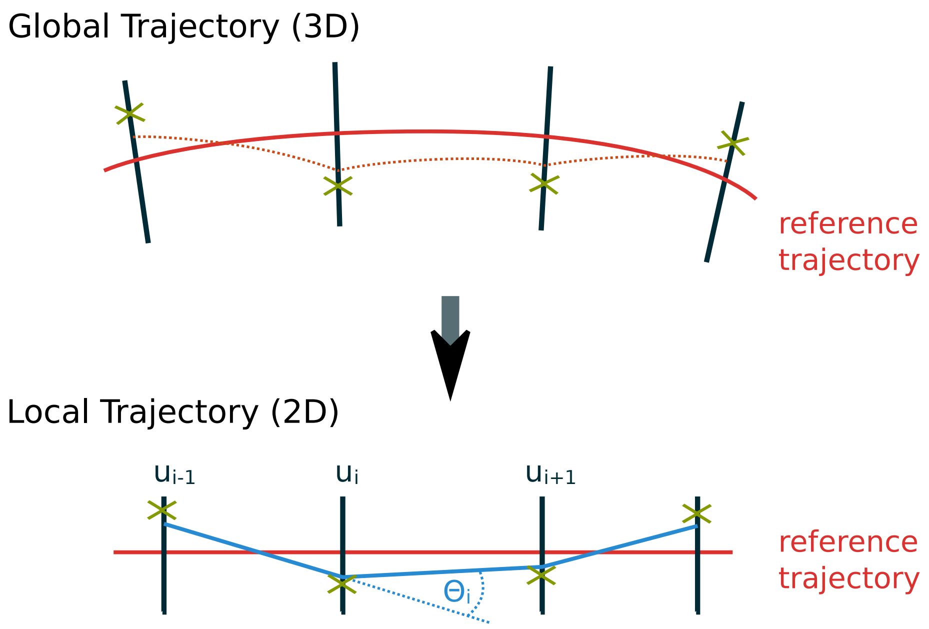
$$S = \vec{r}^T \Sigma^{-1} \vec{r}$$

Solve linearized

$$\overrightarrow{\Delta a_0} = (F^T H^T \Sigma^{-1} H F)^{-1} F^T H^T \vec{r}$$

# General Broken Lines






Specific track model with explicit scattering angles

$$S = \sum \vec{u}_i^T \Sigma_u^{-1} \vec{u}_i + \sum \vec{\beta}(\vec{u})_i^T \Sigma_\beta^{-1} \vec{\beta}(\vec{u})_i$$

No global measurement covariance

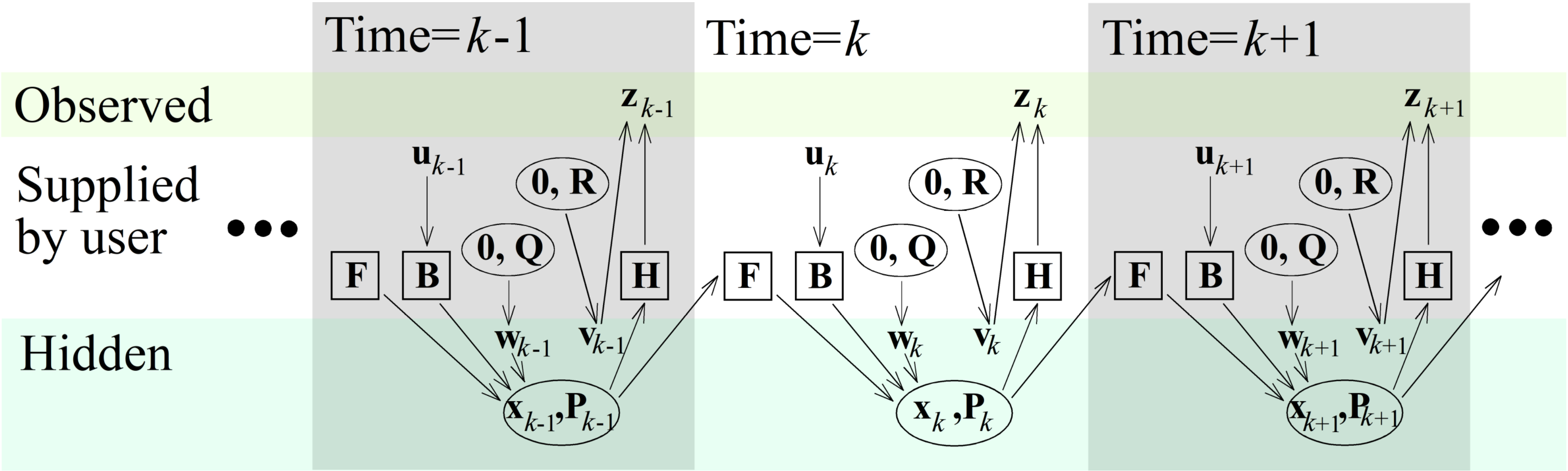
Global parameter covariance

O(N) solution

GBL Website

# Kalman filter a.k.a rocket science






First used in the Apollo space program:
Continously estimate/control hidden state from noisy measurements

Source, Public domain

# Kalman forward filter



Predict

$$\vec{a}_{k,k-1} = F_{k-1,k}\vec{a}_{k-1}$$
$$C_{k,k-1} = F_{k-1,k}^{T} C_{k-1,k-1} F_{k-1,k} + Q_k$$

Residuals

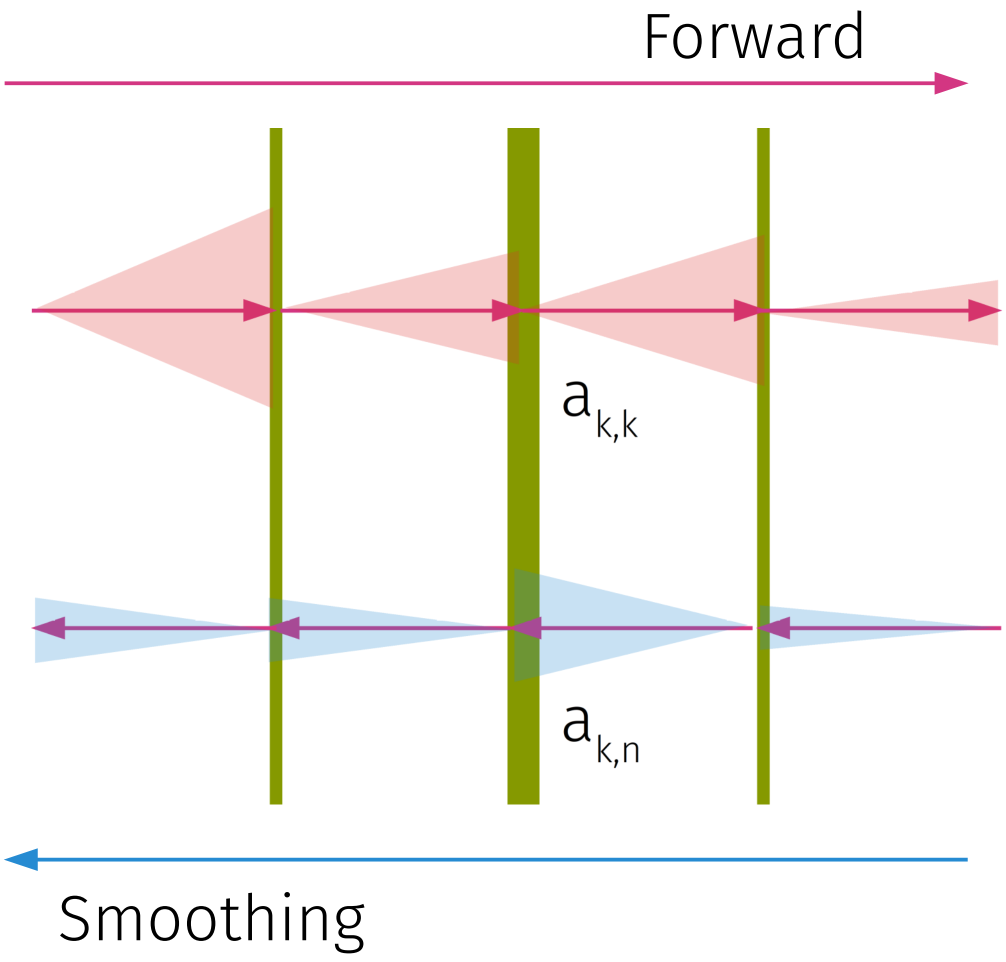
$$\vec{r}_{k,k-1} = \vec{y}_k - H_k \vec{a}_{k,k-1}$$
$$R_{k,k-1} = H_k C_{k-1,k-1} H_k^T + V_k$$

Kalman gain matrix

$$K_k = C_{k,k-1} H_k^T R_{k,k-1}^{-1}$$

Update

$$\vec{a}_{k,k} = \vec{a}_{k,k-1} + K_k \vec{r}_{k,k-1}$$
$$C_{k,k} = \left(I - K_k H_k\right) C_{k,k-1}$$



# Kalman smoothing



Smoother gain matrix

$$K_k = C_{k,k} F^T_{k+1,k} C^{-1}_{k+1,k}$$

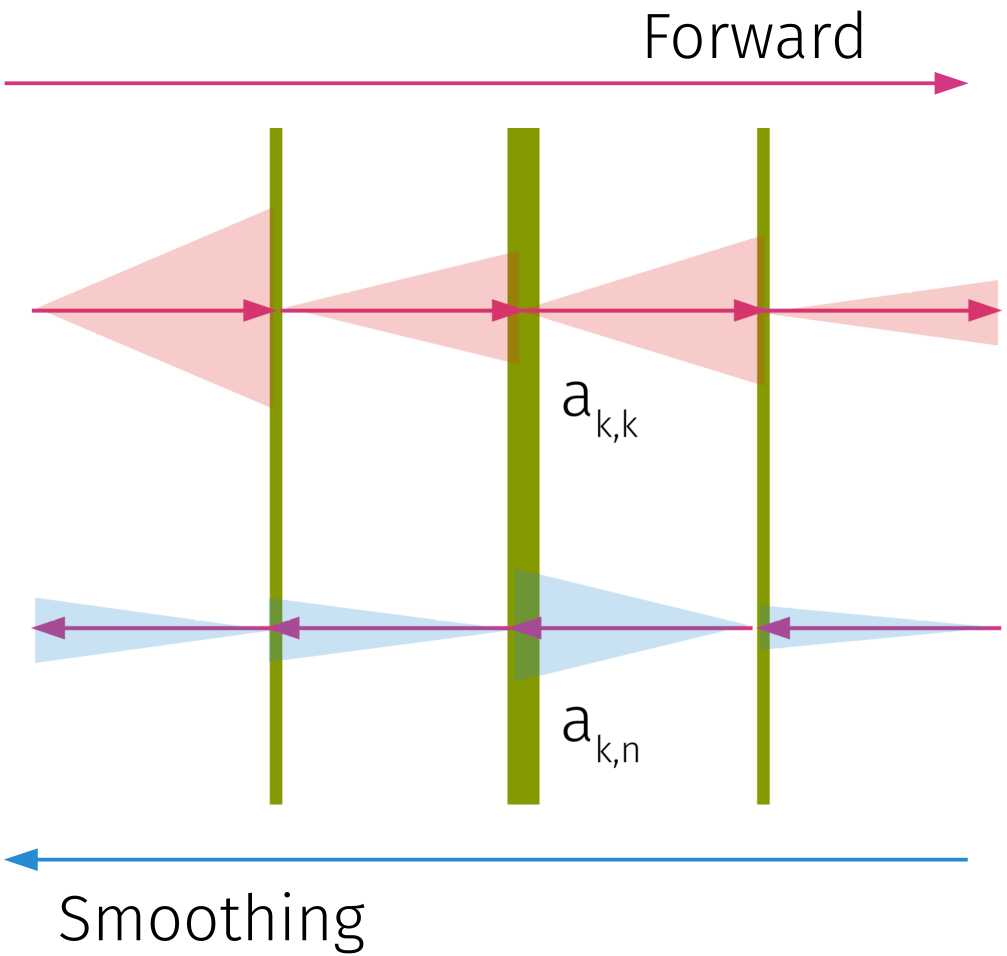
Back smoothing

$$\vec{a}_{k,n} = \vec{a}_{k,k} + A_k \left( \vec{a}_{k+1,n} - \vec{a}_{k+1,k} \right)$$

$$C_{k,n} = C_{k,k} - A_k \left( C_{k+1,k} - C_{k+1,n} \right) A^T_k$$

Final residuals and X²

$$\Delta \chi^2_{k,n} = \vec{r}^{\,T}_{k,n} R^{-1}_{k,n} \vec{r}_{k,n}$$



# Track fitting summary

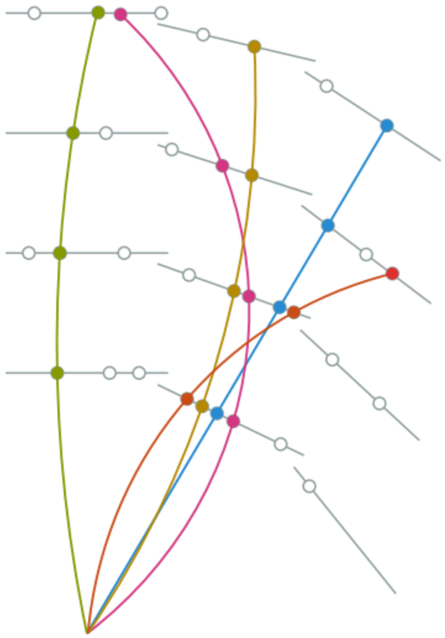


Alll methods require lineariz(ed,able) track models

Model = trajectory + noise

You always need initial parameters

No single algorithm fits for everything



# Track finding



Global methods

Example: Hough transform

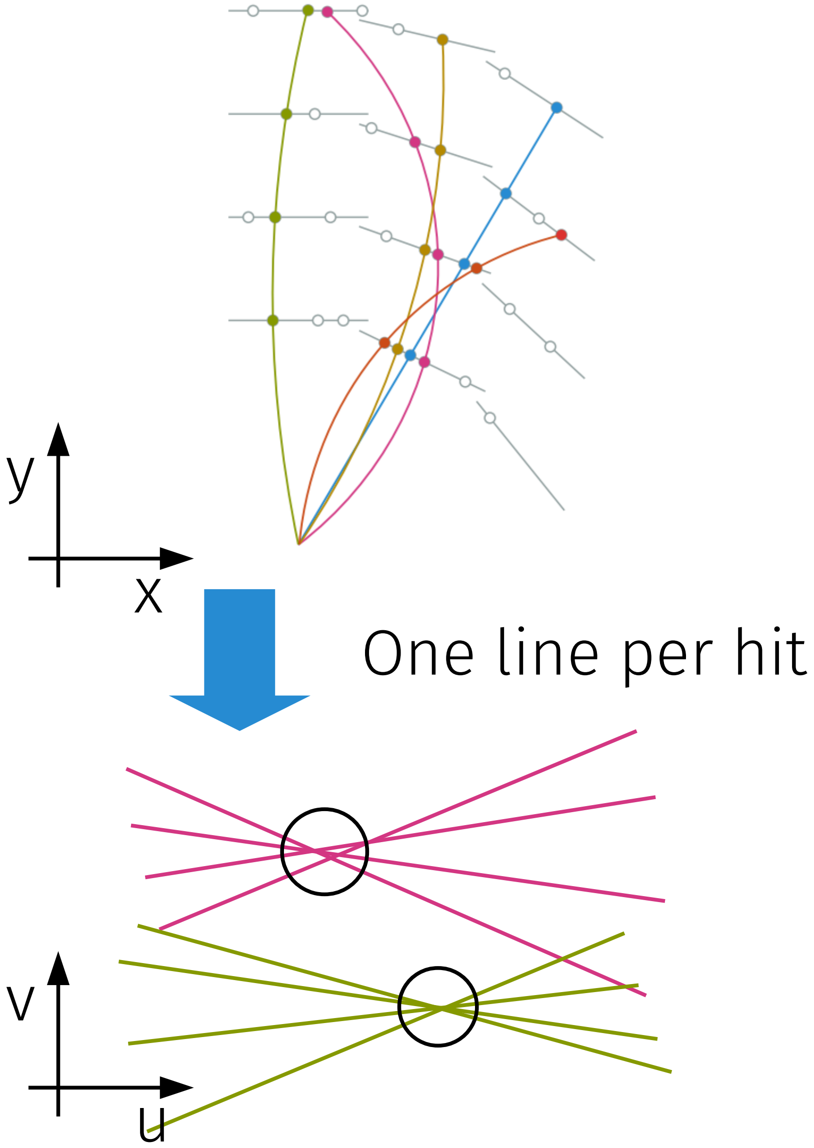
Transform hit

$$u=\frac{x}{x^2+y^2} \qquad v=\frac{y}{x^2+y^2}$$

Draw line for u/v that satisfy

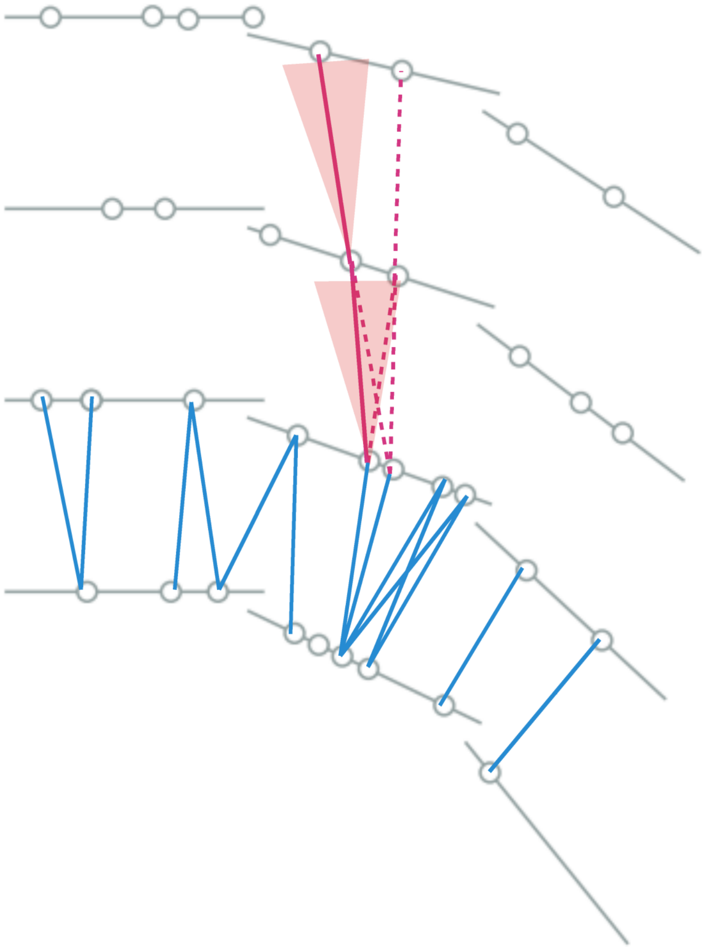
$$v=-\frac{x}{y}u+\frac{x^2+y^2}{2y}$$

Fast, but assumes perfect circles
Sensitive to density



# Combinatorial Kalman Filter





Build track seeds w/ 2-4 hits

Estimate initial track parameters

Propagate through detector and pick up matching hits

- SKF: only the best match
- CKF: bifurcate matches

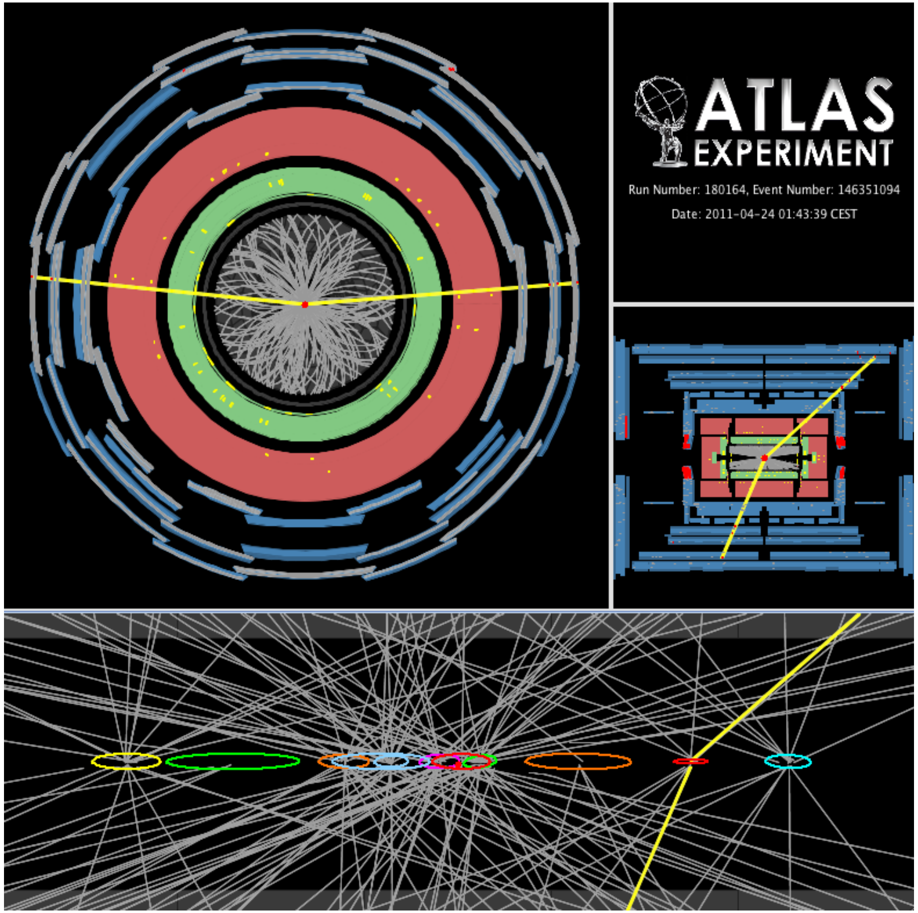
KF already provides matching criterium / significance

$$\Delta\chi^2_{k,k} = \vec{r}^{\,T}_{k,k} R^{-1}_{k,k} \vec{r}_{k,k}$$

# Tracking at High Luminosity

# Track density at low luminosity



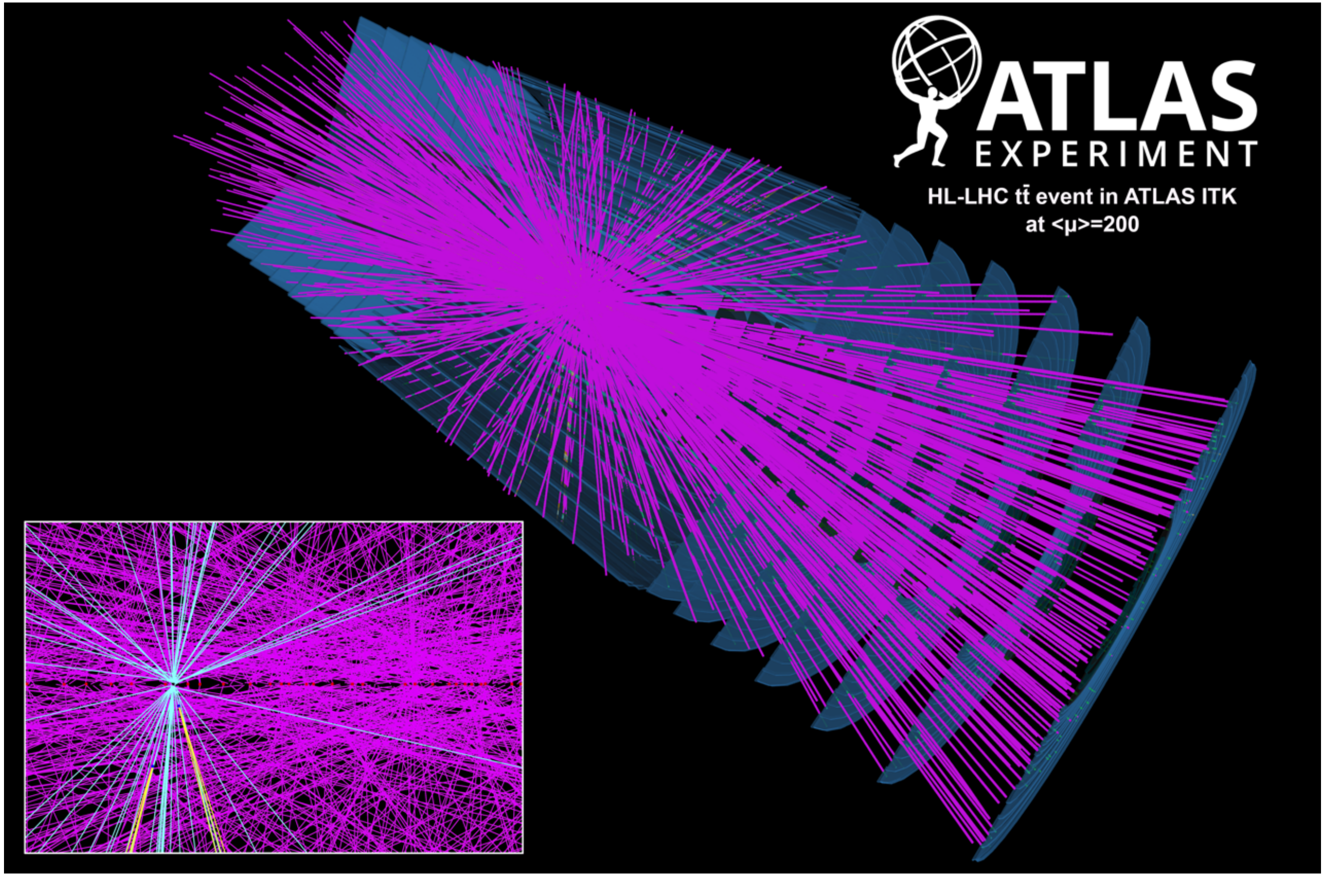



Run 1 conditions

11 reconstructed vertices

# Track density at high luminosity

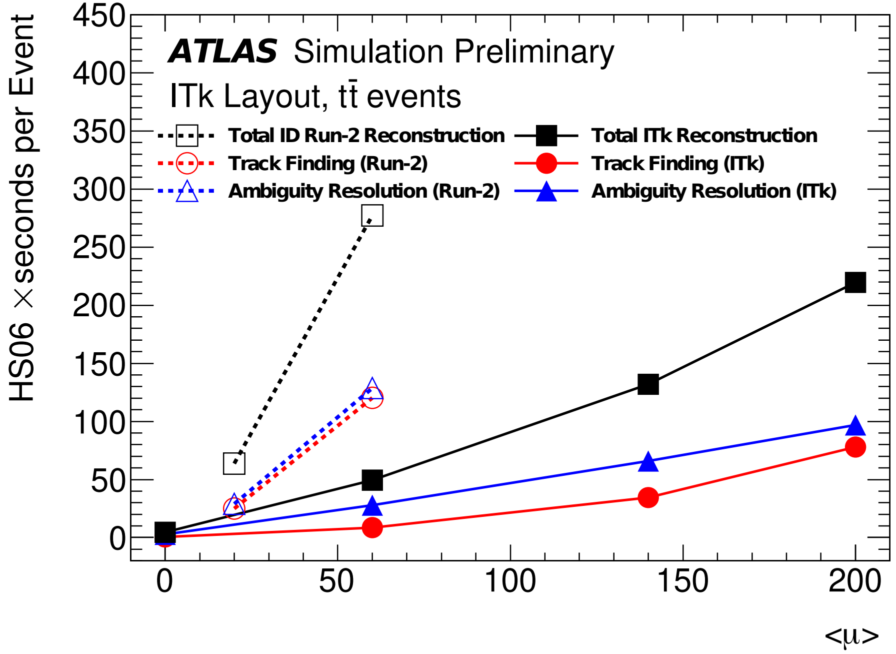


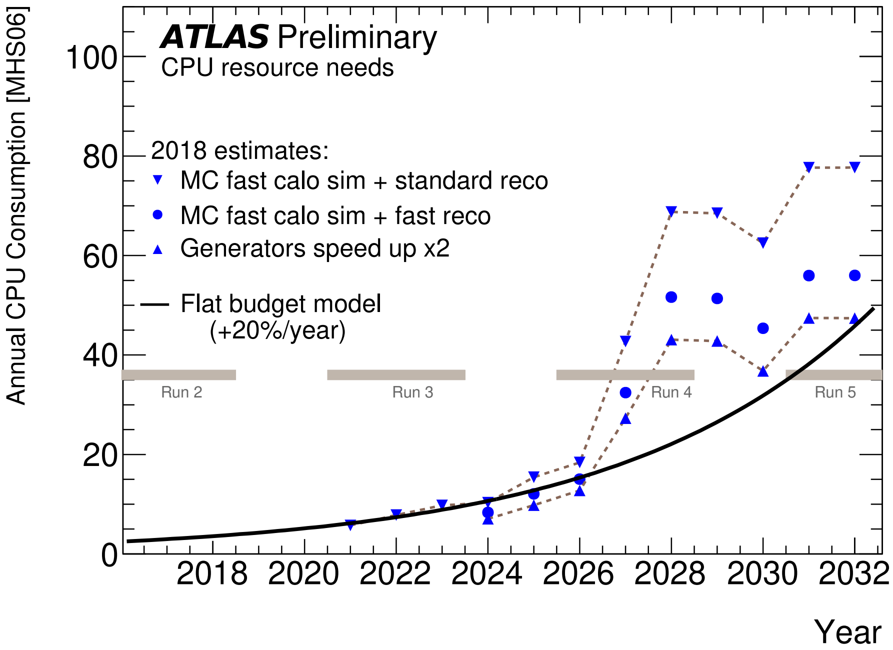



Run 4 conditions (expected)

~200 interactions

# Runtime scaling combinatorial approach 




What can we do?

- Improve implementation
- Improve algorithms

See ATL-PHYS-PUB-2019-041

# ACTS – A Common Tracking Software



- Experiment-independent, platform for tracking tools:
  - Geometry, navigation, track finding and fitting, …
- Open-source (MPLv2)
- Modern code for modern architectures
- Hackable

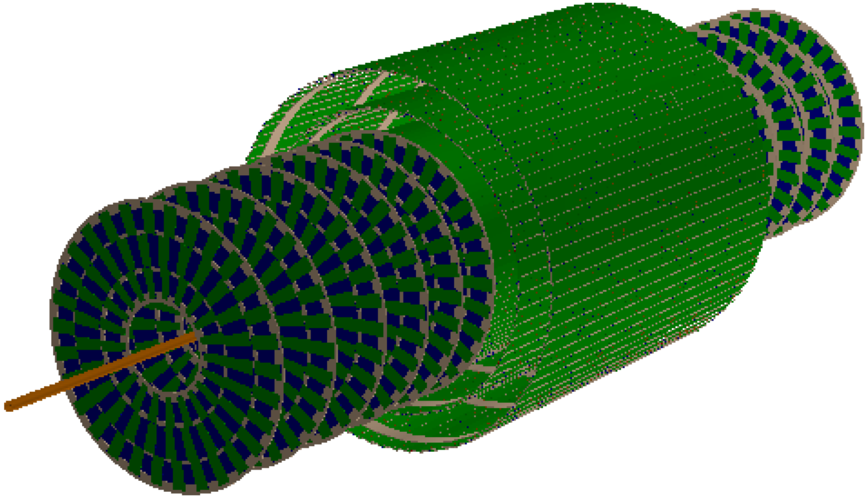
cern.ch/acts

# Example detector geometries

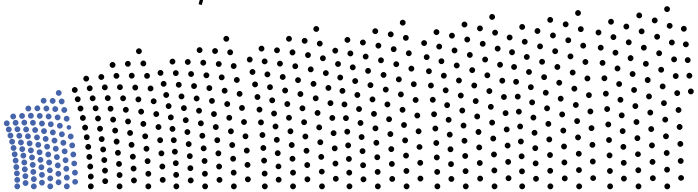


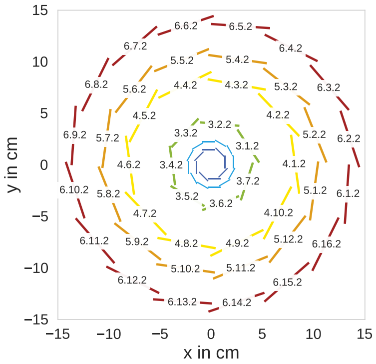
## OpenDataDetector

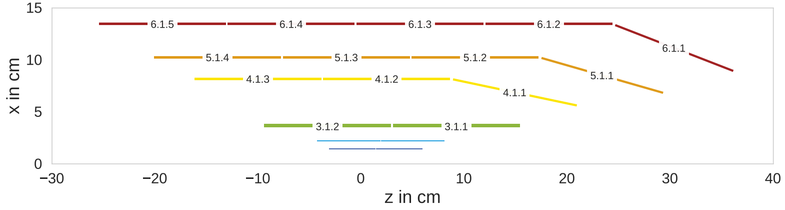
Virtual, test detector



## Belle 2 CDC/ VXD






Both are Work-in-Progress

# The TrackML challenge



Modern algorithms
Non-obvious approaches

For charged particles in (high energy) colliders

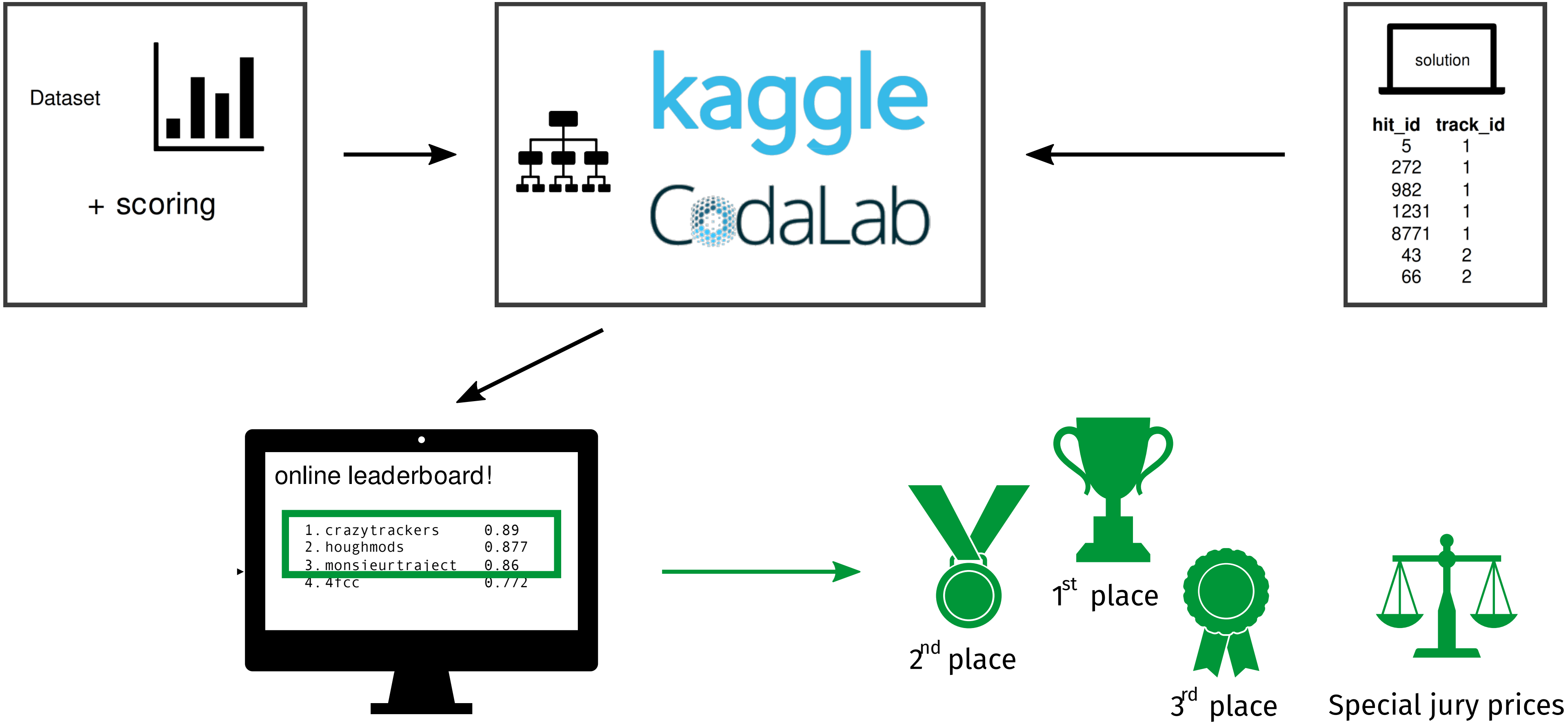
A public machine learning challenge for tracking algorithms

Fresh eyes not just physicists

Cash prizes for motivated participants

On Kaggle and on Codalab and the final workshop

Task/problem Platform Participants 

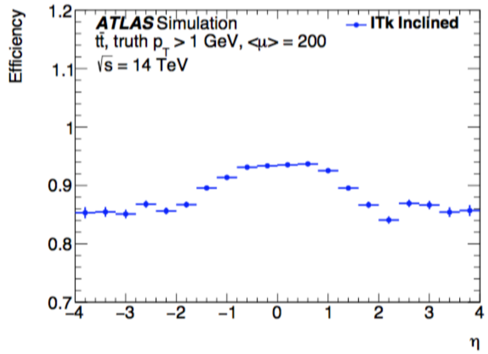


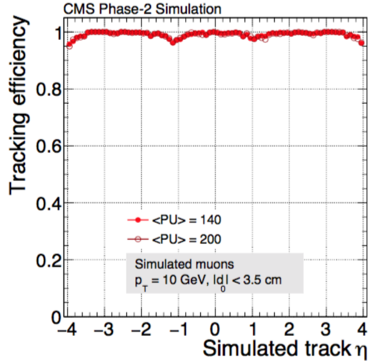
# What is the main problem?

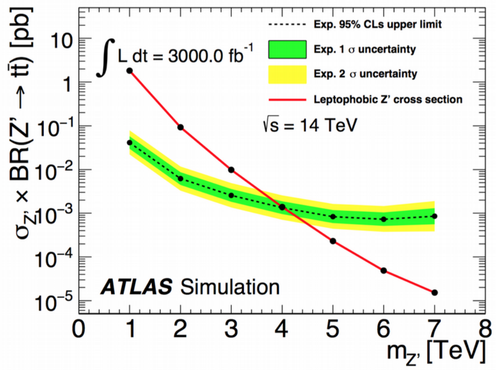


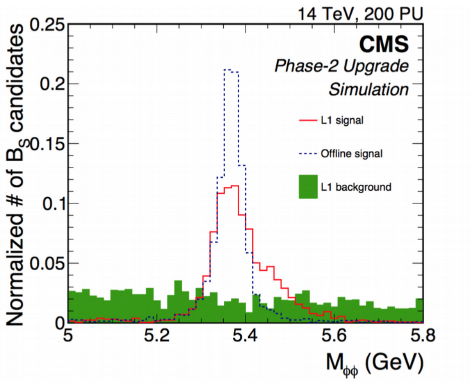
Tracking has many metrics

- Global efficiency
- Efficiency for certain classes
- Fake rate vs. purity
- Momentum resolution
- Impact resolution
- Physics impact
- …



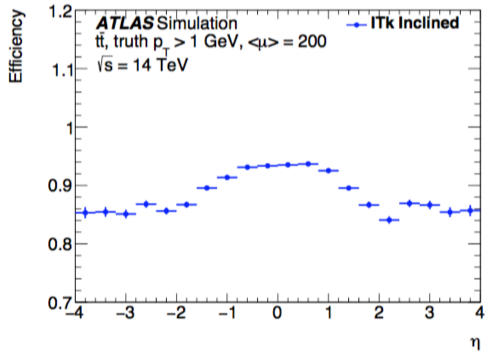



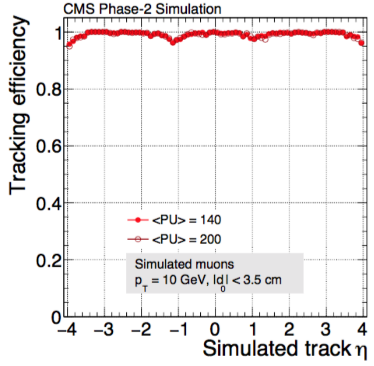
# What is the main problem?

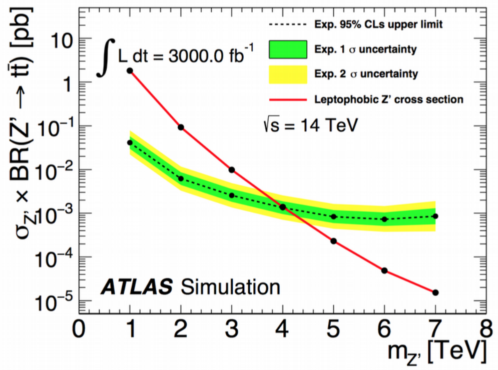


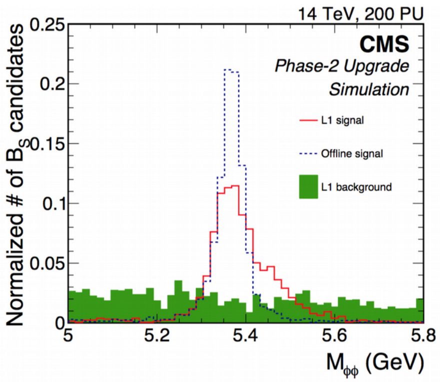
Tracking is multitudes

- Track seeding
- Track finding (extension)
- Track fitting
- Primary/secondary vertex finding
- …



# The problem is connecting the dots

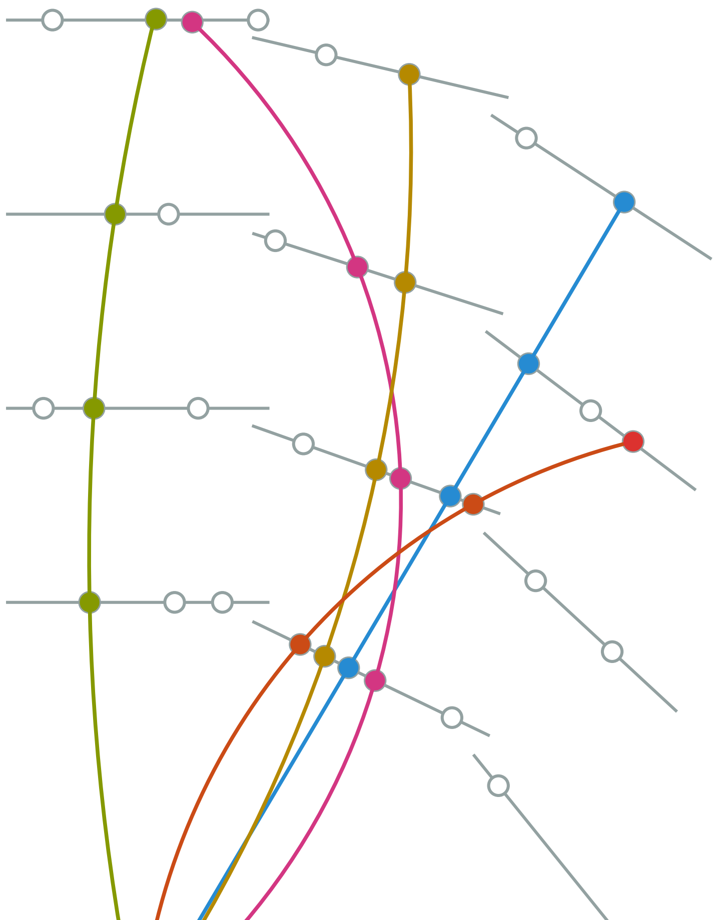


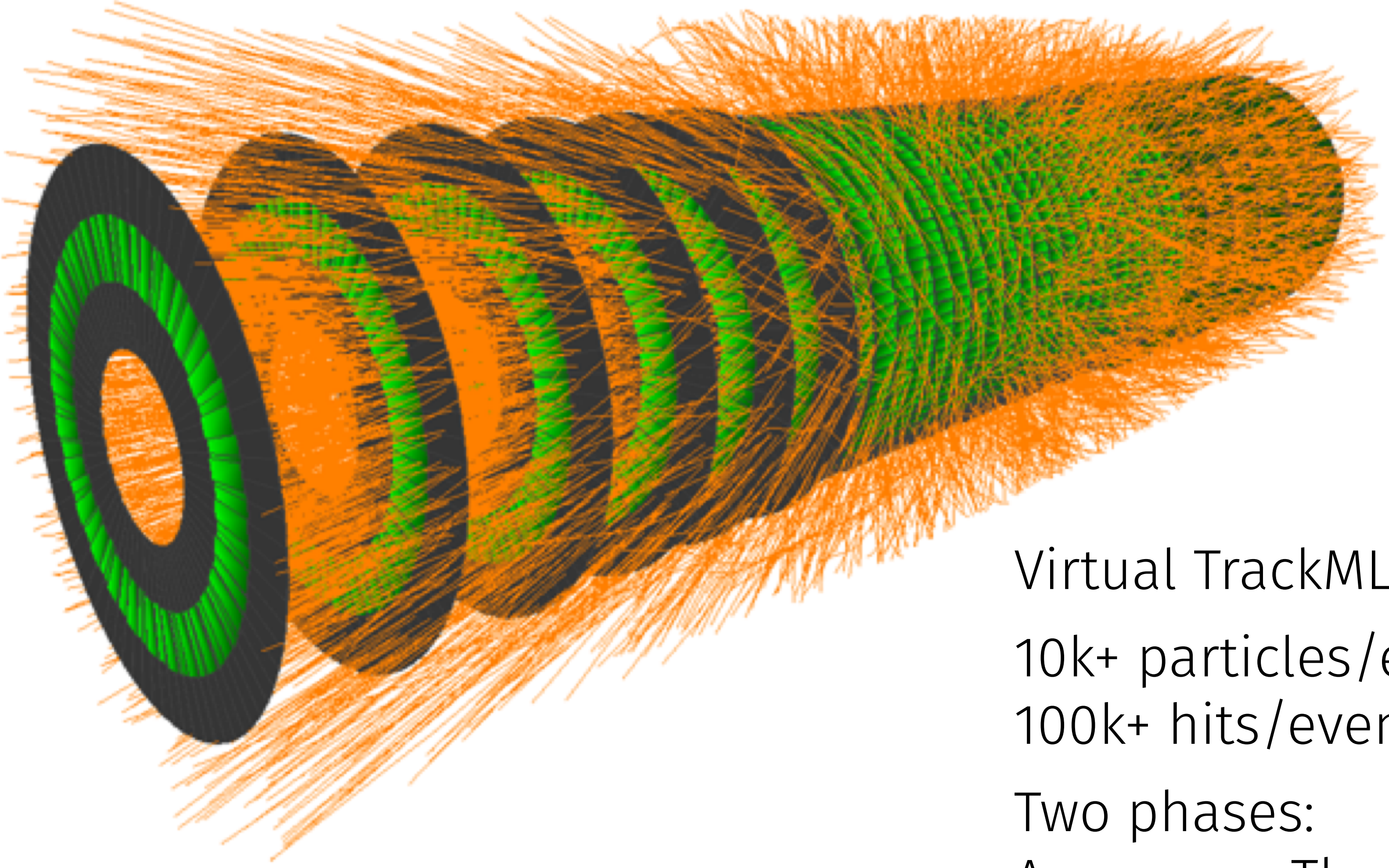
No parameter estimation

(Kalman filter works)

No hit merging/splitting

(NN mostly work)





Virtual TrackML Detector

10k+ particles/event
100k+ hits/event

Two phases:
Accuracy + Throughput

# Accuracy phase on kaggle



Ran until August 2018

600+ participants

Submit results only

Only measure accuracy

12k€, 8k€, 5k€ prizes
+ NVIDIA V100 GPU

| | | | | | |
|---|---|---|---|---|---|
| 1 | — | Top Quarks | | 0.92182 | 10 |
| 2 | — | outrunner | | 0.90302 | 9 |
| 3 | — | Sergey Gorbunov | HEP | 0.89353 | 6 |
| 4 | — | demelian | HEP | 0.87079 | 35 |
| 5 | — | Edwin Steiner | | 0.86395 | 5 |
| 6 | — | Komaki | | 0.83127 | 22 |
| 7 | — | Yuval & Trian | | 0.80414 | 56 |
| 8 | — | bestfitting | | 0.80341 | 6 |
| 9 | — | DBSCAN forever | | 0.80114 | 23 |
| 10 | — | Zidmie & KhaVo | | 0.76320 | 26 |
| 11 | — | Andrea Lonza | | 0.75845 | 15 |
| 12 | — | Finnies | | 0.74827 | 56 |
| 13 | — | Rei Matsuzaki | | 0.74035 | 12 |
| 14 | — | Mickey | | 0.73217 | 10 |
| 15 | — | Vicens Gaitan | | 0.70429 | 19 |
| 16 | — | Robert | | 0.69955 | 3 |
| 17 | — | Yuval-CPMP tribute band | | 0.69364 | 20 |
| 18 | — | N. Hi. Bouzu | | 0.67573 | 9 |
| 19 | — | Steins;Gate | | 0.66763 | 12 |
| 20 | ▲1 | Victor Nedel'ko | | 0.66723 | 4 |

# Throughput phase on CodaLab



Ran until March 2019

Only 10+ active participants

Submit results only

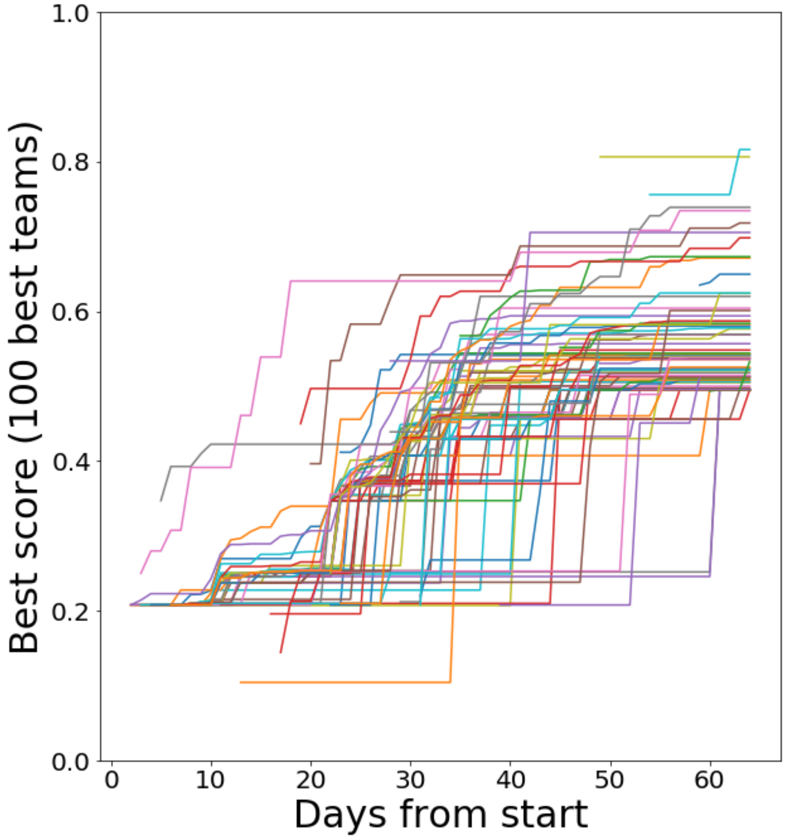
Measure accuracy and speed

7k€, 5k€, 3k€ prizes
+ NVIDIA V100 GPU

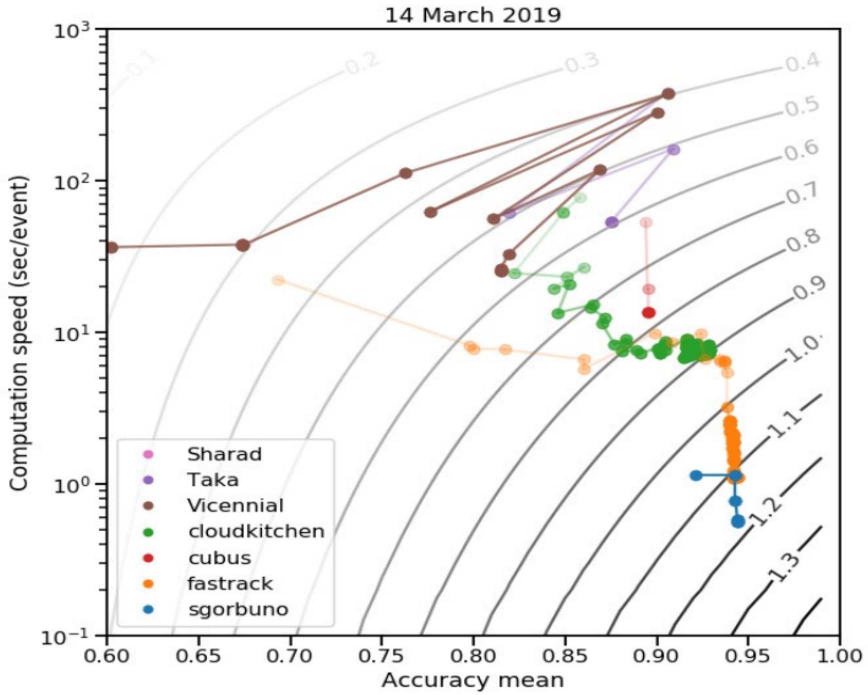
| RESULTS | | | | | | | |
|---|---|---|---|---|---|---|---|
| # | User | Entries | Date of Last Entry | score ▲ | accuracy_mean ▲ | accuracy_std ▲ | com (sec |
| 1 | sgorbuno HEP | 9 | 03/12/19 | 1.1727 (1) | 0.944 (2) | 0.00 (14) | 28. |
| 2 | fastrack HEP | 53 | 03/12/19 | 1.1145 (2) | 0.944 (1) | 0.00 (15) | 55. |
| 3 | cloudkitchen (HEP) | 73 | 03/12/19 | 0.9007 (3) | 0.928 (3) | 0.00 (13) | 364 |
| 4 | cubus | 8 | 09/13/18 | 0.7719 (4) | 0.895 (4) | 0.01 (9) | 675 |
| 5 | Taka | 11 | 01/13/19 | 0.5930 (5) | 0.875 (5) | 0.01 (12) | 266 |
| 6 | Vicennial | 27 | 02/24/19 | 0.5634 (6) | 0.815 (6) | 0.01 (10) | 127 |
| 7 | Sharad | 57 | 03/10/19 | 0.2918 (7) | 0.674 (7) | 0.02 (4) | 190 |
| 8 | WeizmannAI | 5 | 03/12/19 | 0.0000 (8) | 0.133 (11) | 0.01 (11) | 88. |
| 9 | harshakoundinya | 2 | 03/12/19 | 0.0000 (8) | 0.085 (13) | 0.01 (6) | 49. |
| 10 | iWit | 6 | 03/10/19 | 0.0000 (8) | 0.082 (15) | 0.01 (8) | 48. |

# Score progression

## Accuracy




## Throughput



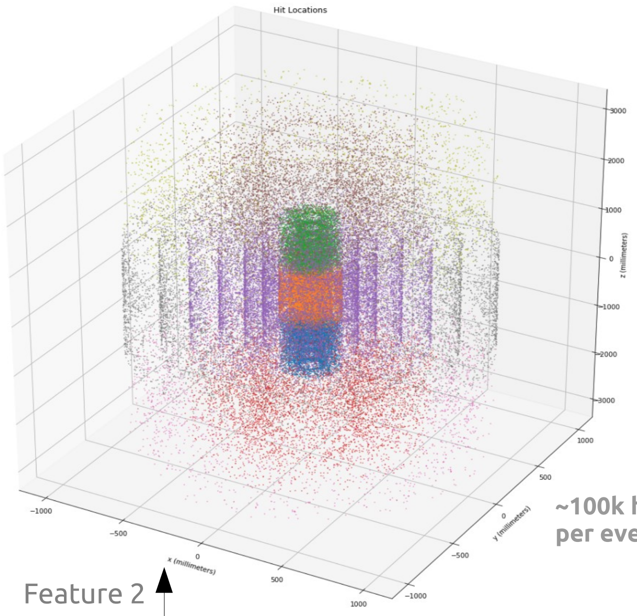

Plots from Laurent Basara

# Accuracy #12: Finnies (Jury Deep Learning Prize)

Liam Finnie & Nicole Finnie
IBM Germany R&D
Bosch Centre for AI

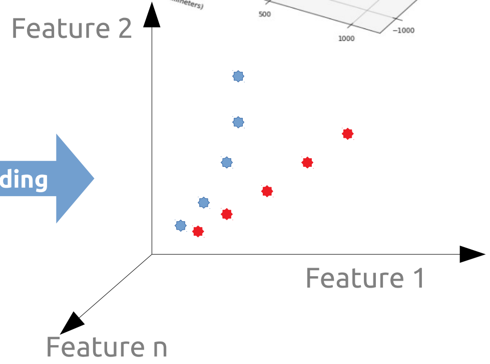
https://github.com/jliamfinnie/kaggle-trackml

# Solution Pipeline




~100k hits (~10k tracks) per event

seeding




inference

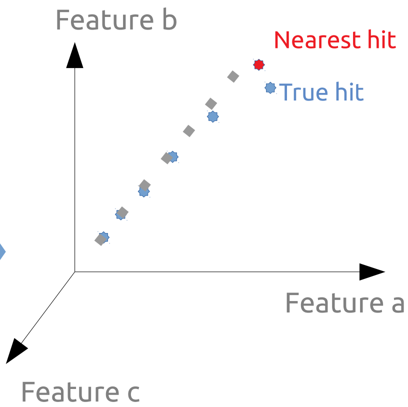
Bi-LSTM

LSTM

LSTM

LSTM

fitting

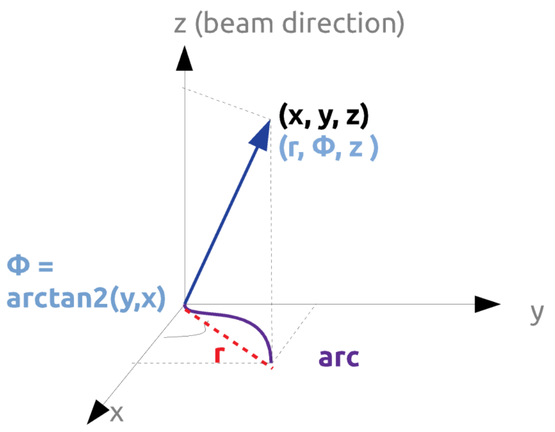



**Track seeding**
(clustering)

**Inference & Ensembling**

**Track fitting**
(k-nearest neighbour)

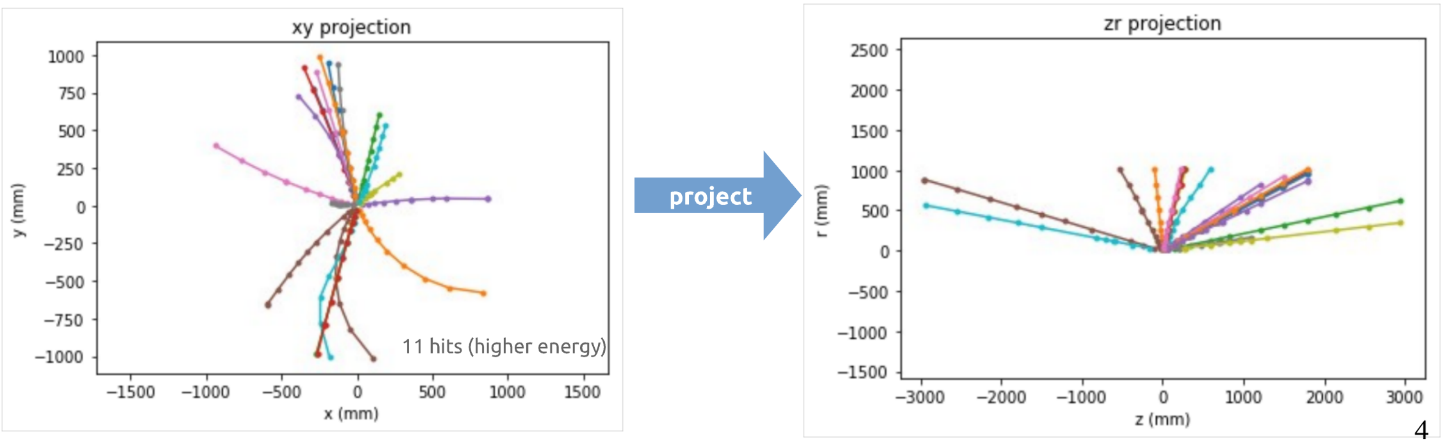
# Feature Engineering… for people who don't know physics :D




**Data we use:** (x, y, z) coordinates of hits

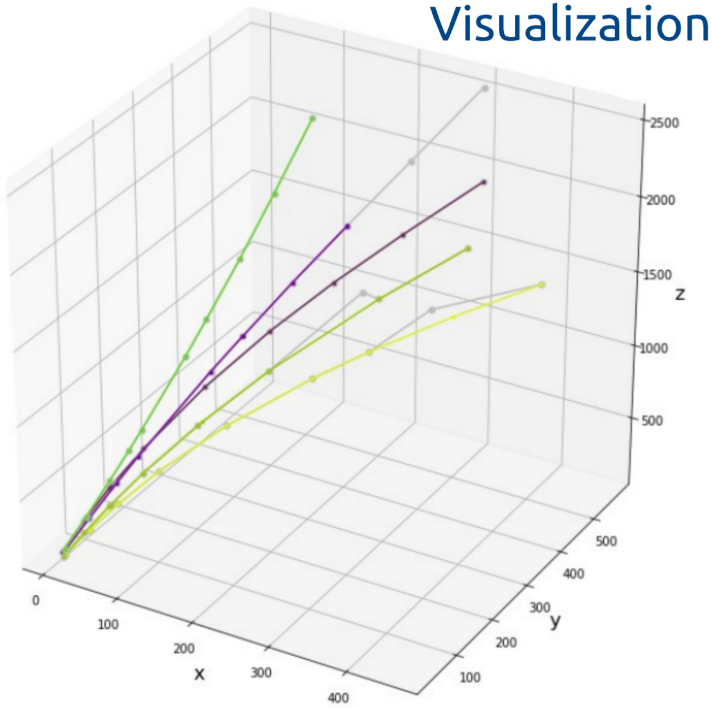
**For clustering**: sin(Φ), cos(Φ), z/arc
(new feature: generate possible arcs using train data)

**For LSTM**: Φ, r, z, z/r



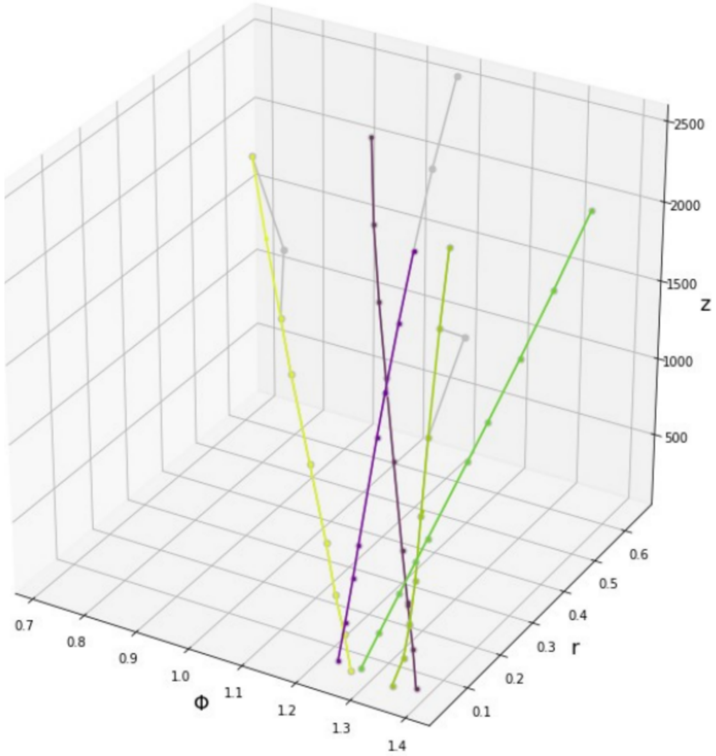

Cartesian -> Polar coordinates: **easier for LSTM to learn**

# Visualization after fitting




Cartesian coordinates

--- predicted (grey)
--- ground truth (colour)




Polar coordinates

# Accuracy #2: outrunner

Pei-Lien Chou

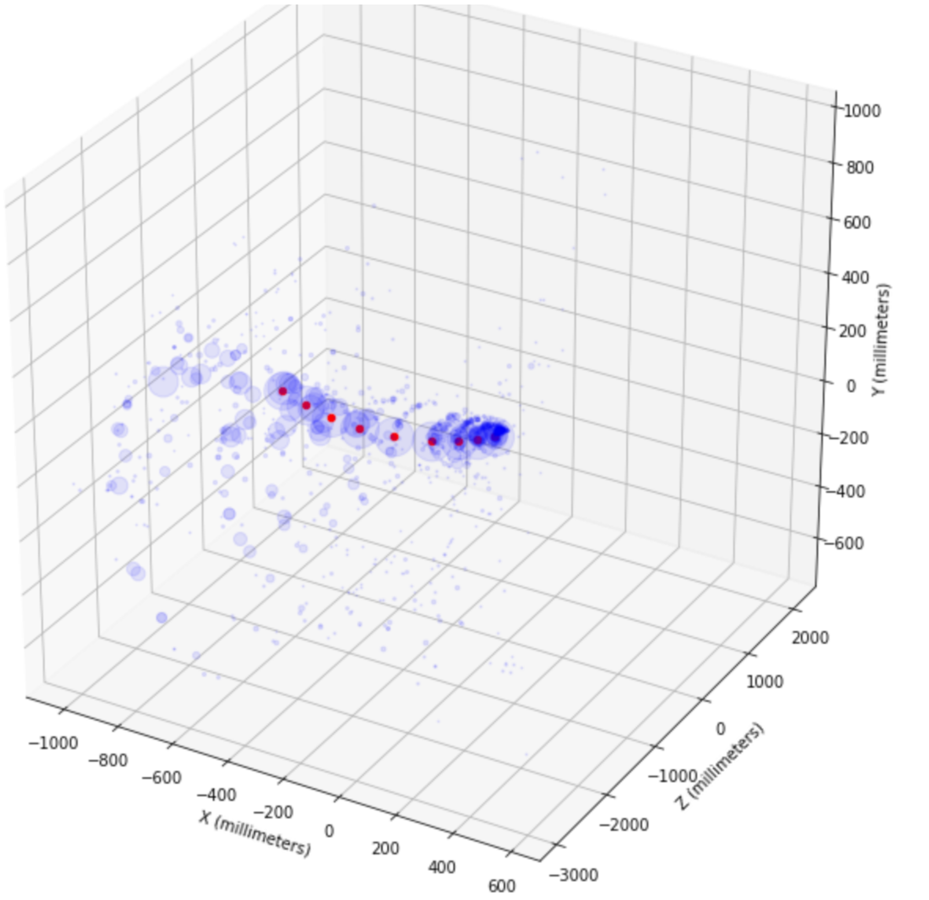
Software engineer image-based deep learning in Taiwan.

Kaggle Notebook

# outrunner – Setup



- Train DNN on hit pairs
  - 27 inputs (x,y,z,cells,…)
  - 4k-2k-2k-2k-1k hidden layers
- Compute full hit adjacency matrix: probability P(i,j) that 2 hits match
- Pick high probability comb.
- Helix-like fit for cleaning




Graphics from outrunner

# ML vs classical reconstruction



Classical reconstruction

Local Reco

Bits to hits, clustering

Track Finding

Track Fitting

Parameter estimation

Analysis

Simulation (Reality)

Digitized hits

Material interactions

Propagation

Particles

ML (supervised): try to learn inverse solution directly

# Summary



Tracking is core reconstruction for many particle physics experiment

Rich set of classical algorithms

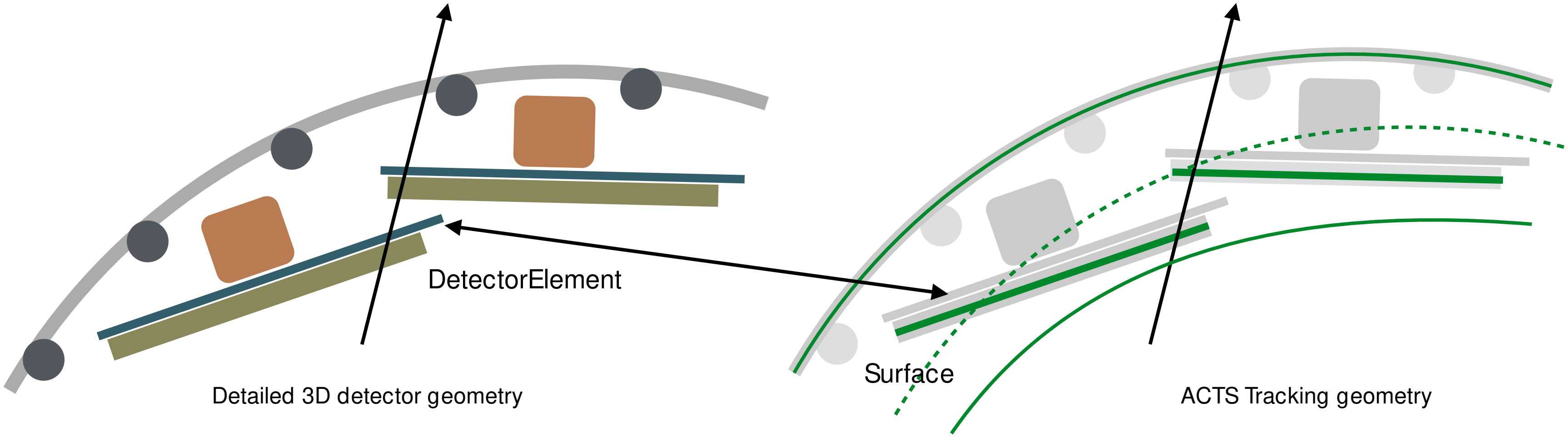
Interesting machine-learning based solutions

Things I did not talk about:

- Tracking on accelerators
- Triggering with tracks
- Local reconstruction
- Outliers and robustness
- Alignment
- …

# Appendix

# (Tracking) Geometry




Simplified geometry based on tracking surfaces

# TrackML Organizers



Paolo Calafiura, Steven Farrell, Heather Gray (LBNL Berkeley), Jean-Roch Vlimant (Caltech), Isabelle Guyon (ChaLearn, U Paris Saclay), Laurent Basara, Cécile Germain (LAL/LRI, U Paris Saclay), David Rousseau, Yetkin Yilnaz (LAL Orsay, U Paris Saclay), Vincenzo Innocente, Andreas Salzburger (CERN), Ilija Vukotic (U of Chicago), Tobias Golling, Moritz Kiehn, Sabrina Amrouche (U Genève), Edward Moyse (U of Massachusetts), Vava Gligorov (LPNHE Paris), Mikhail Hushchyn, Andrey Ustyuzhanin (Yandex)

# TrackML sponsors

Sponsored by

CERN

kaggle

NVIDIA

UNIVERSITÉ DE GENÈVE

Paris-Saclay Center for Data Science

INSTITUTE DATAIA
Data Science, Intelligence & Society

CHA LEARN

COMMON GROUND

CERN openlab

Inria
INVENTORS FOR THE DIGITAL WORLD

Caltech

Fermilab

erc

UNIVERSITÉ PARIS SUD

université PARIS-SACLAY