TrackML, the Tracking Machine Learning challenge

David Rousseau

LAL-Orsay

rousseau@lal.in2p3.fr @dhpmrou

Dortmund RAPID workshop, 19th Nov 2018

TrackML sponsors

CERN

kaggle

NVIDIA

UNIVERSITÉ DE GENÈVE

Paris-Saclay Center for Data Science

INSTITUTE DATAIA
Data Science, Intelligence & Society

CHA LEARN

UNIVERSITÉ PARIS SUD

université PARIS-SACLAY

erc

CERN openlab

Inria
INVENTORS FOR THE DIGITAL WORLD

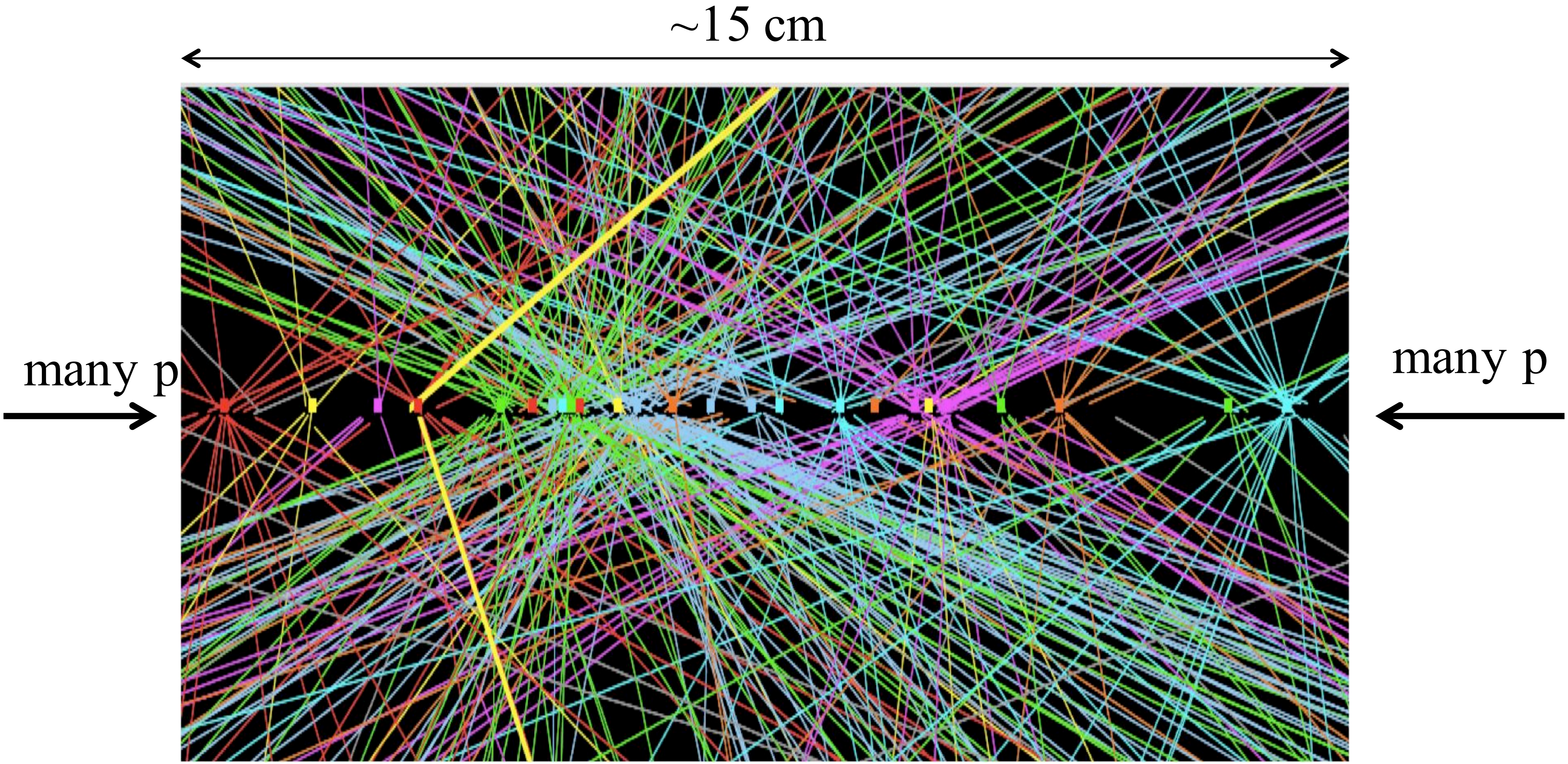
PyTorch

Fermilab

COMMON GROUND

inqnet
Alliance for Quantum Technologies

LHC proton Bunch collision

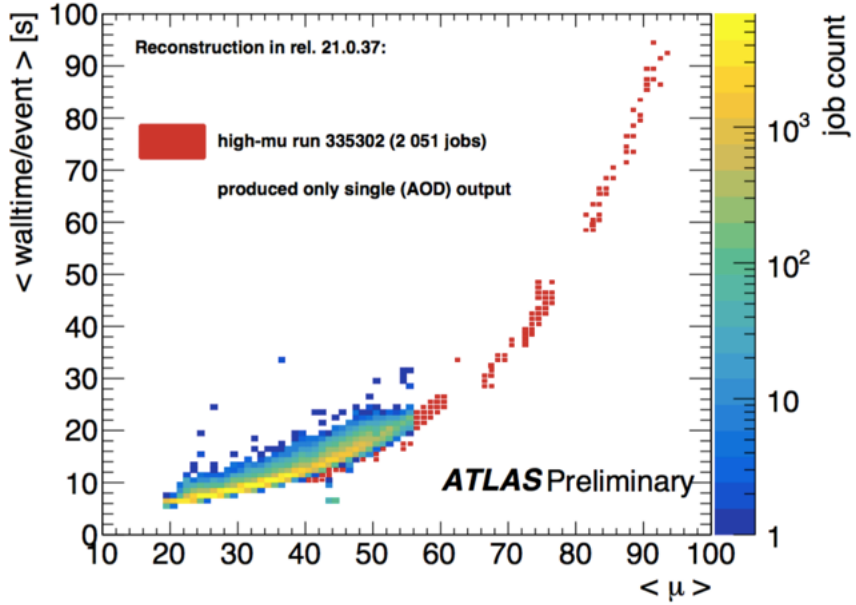


Run 1 situation: 20 parasitic collisions

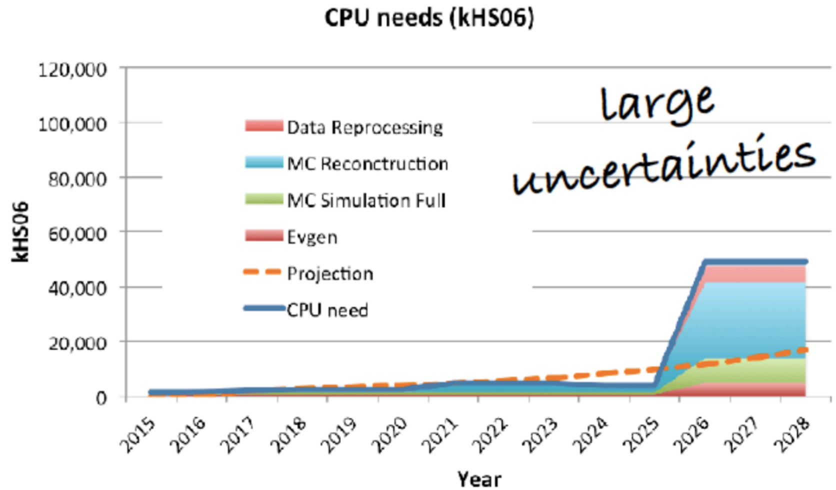
High Lumi-LHC : 200 parasitic collisions

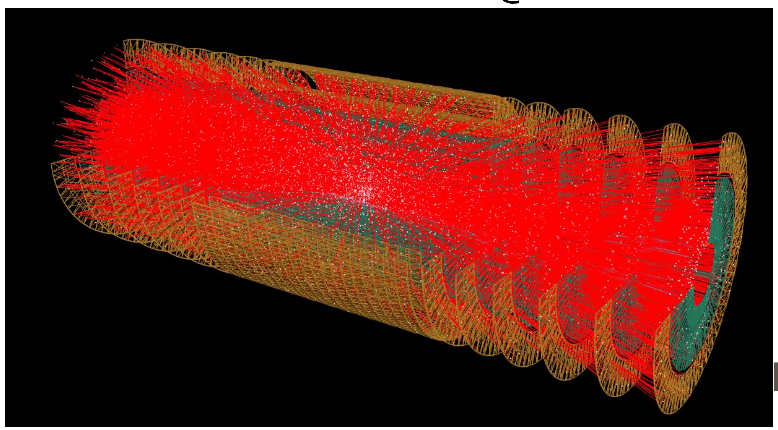
Tracking crisis

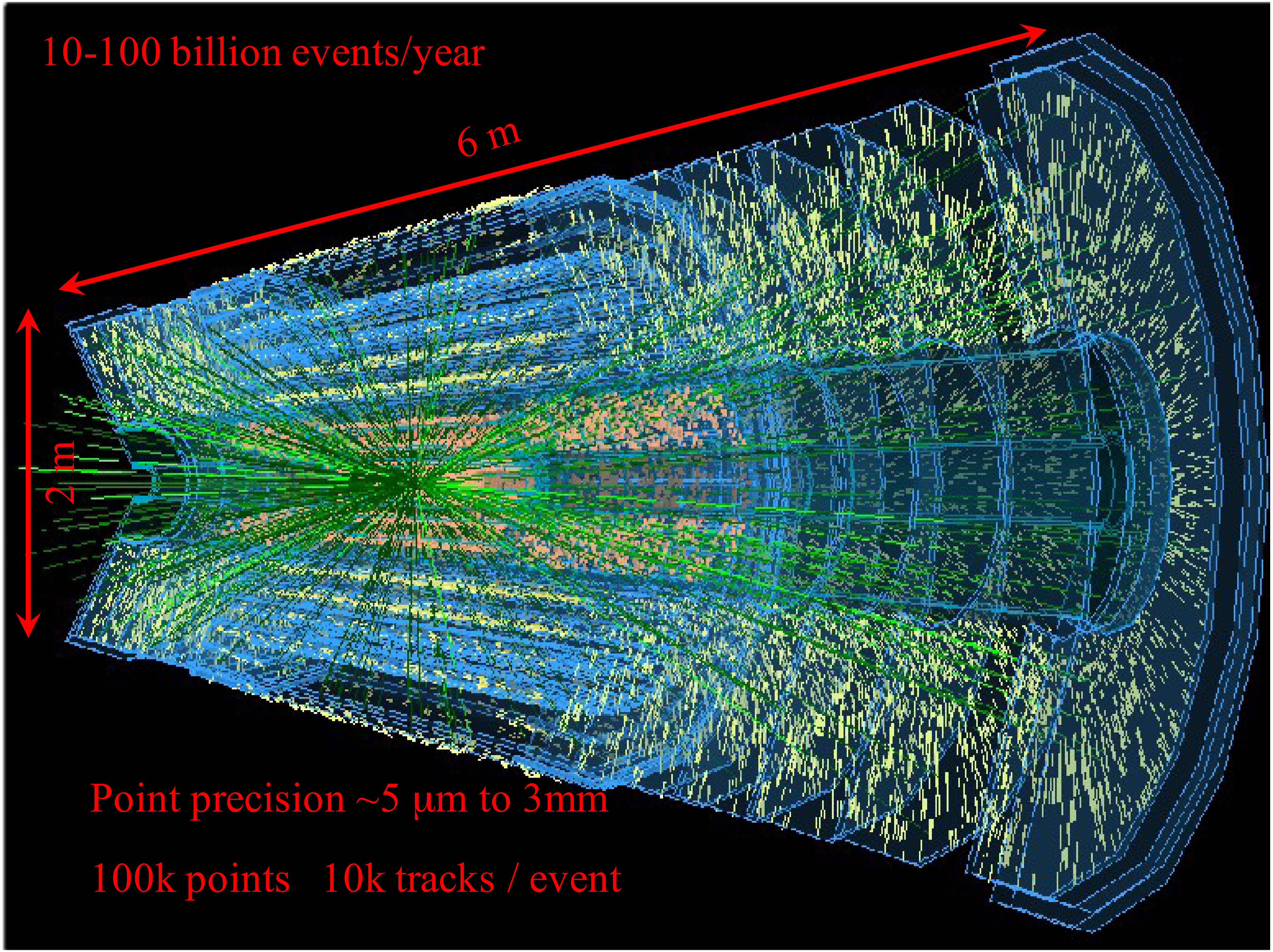
- Tracking (in particular pattern recognition) dominates reconstruction CPU time at LHC
- HL-LHC (phase 2) perspective : increased pileup :Run 1 (2012): <>~20, Run 2 (2015): <>~30,Phase 2 (2025): <>~150
- CPU time quadratic/exponential extrapolation (difficult to quote any number)
- Large effort within HEP to optimise software and tackle micro and macro parallelism. Sufficient gains for Run 2 but still a long way for HL-LHC.
- >20 years of LHC tracking development. Everything has been tried?
  - Maybe yes, but maybe algorithm slower at low lumi but with a better scaling have been dismissed ?
  - Maybe no, brand new ideas from ML (i.e. Convolutional NN)
- ➔Tracking challenge May-November 2018
- Follow us on twitter @trackmllhc !



→ 150






10-100 billion events/year
6 m
2 m
Point precision ~5 μm to 3mm
100k points   10k tracks / event

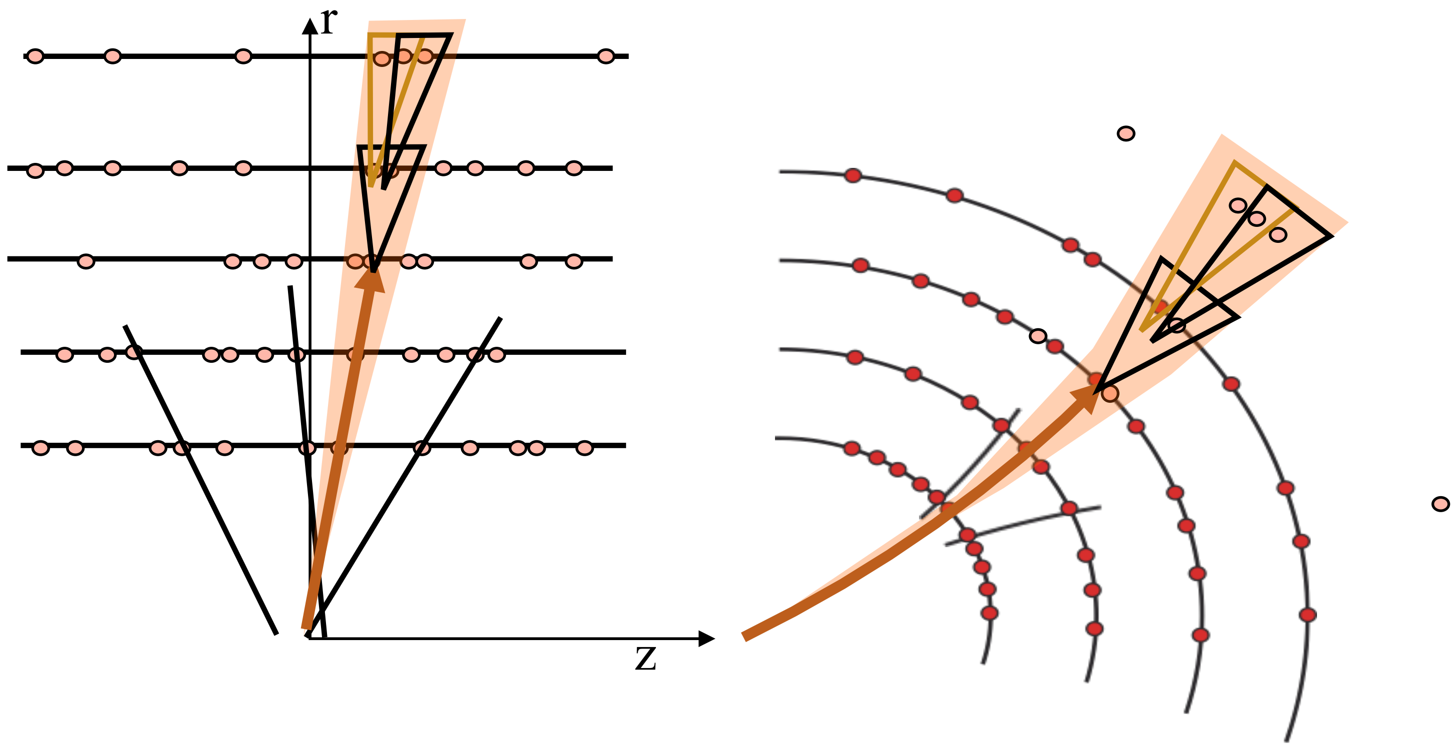
Why is it difficult?

- 100'000 to group into 10'000 tracks of 10 points
  - o ➔~$10^{500'000}$ combinations
  - o ⇒brute force has (really) no chance
- Precision of the points : ~50μm on a volume ~40 m^3
  - o ➔3 10^{14} voxels!
  - o 2D projection ➔2 10^9 pixels !
  - o ⇒ image recognition algorithm have (really) no chance
- Not a classical problem

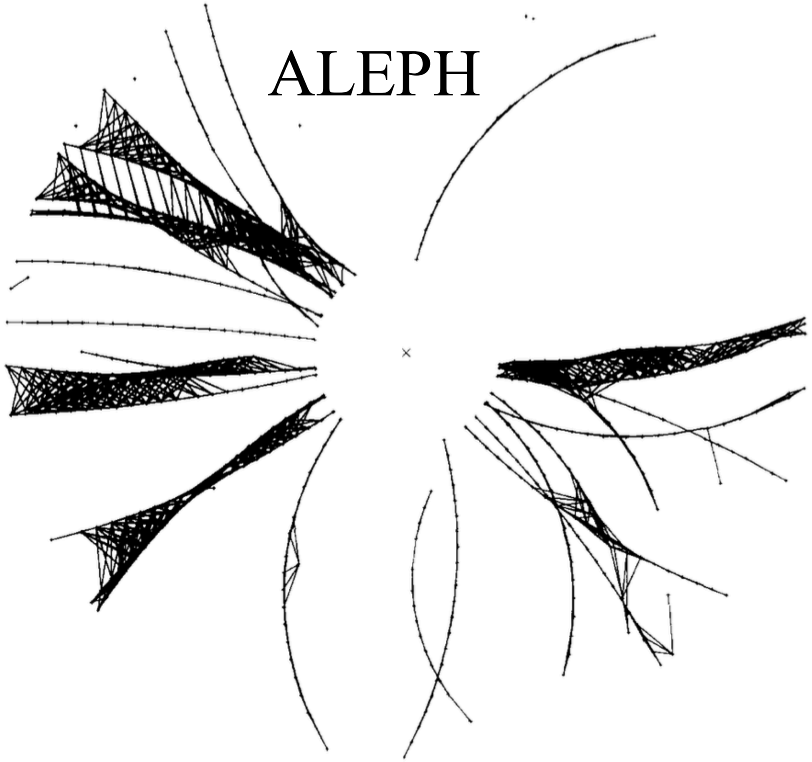
6

Classic HEP Algorithms

- Pattern : connect 3D points into tracks
- Essentially combinatorial approach
- Tracks are (not perfect) helices pointing (approximately) to the origin
- Challenge : explore completely new approaches
- (not part of the challenge : given the points, estimate the track parameters)



An early attempt

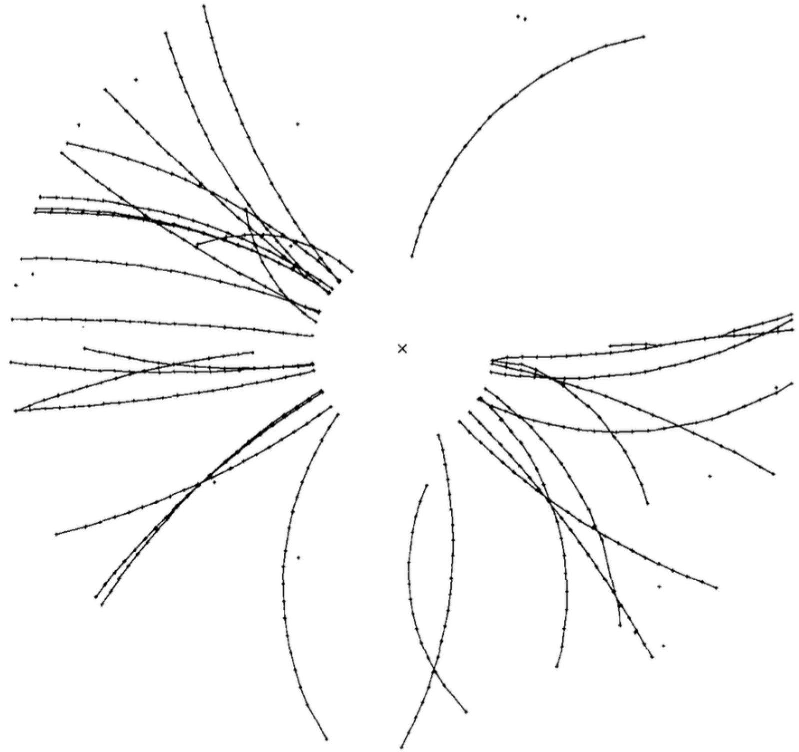


UT
PHYSIC

ıce

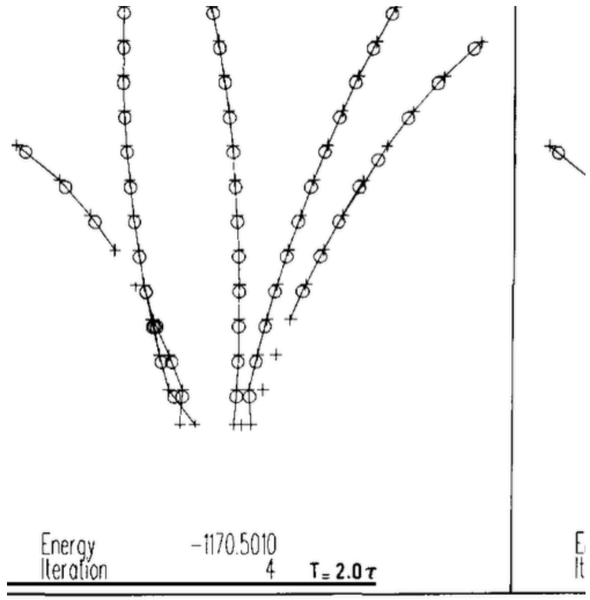
December 1

ıe Learnir



known

- Losely inspired from Traveling Salesman Problem with NN by Hopfield & Tank Biological Cybernetics 52 (1985) 141. or with Minimal Tree Span Cassel & Kowalski Nucl Inst; and Meth 185 (1981) 235
- (large litterature since, e.g. Neural Combinatorial Optimization with reinforcement learning, Bello et al Google Brain 1611.0994)
- Full implementation in ALEPH Stimpfl & Garrido (1990) Computer Physics Comm. 64 (1991) 46.
- However never deployed



Aparté on ML in HEP history

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

NEURAL NETWORKS AND CELLULAR AUTOMATA IN EXPERIMENTAL HIGH ENERGY PHYSICS

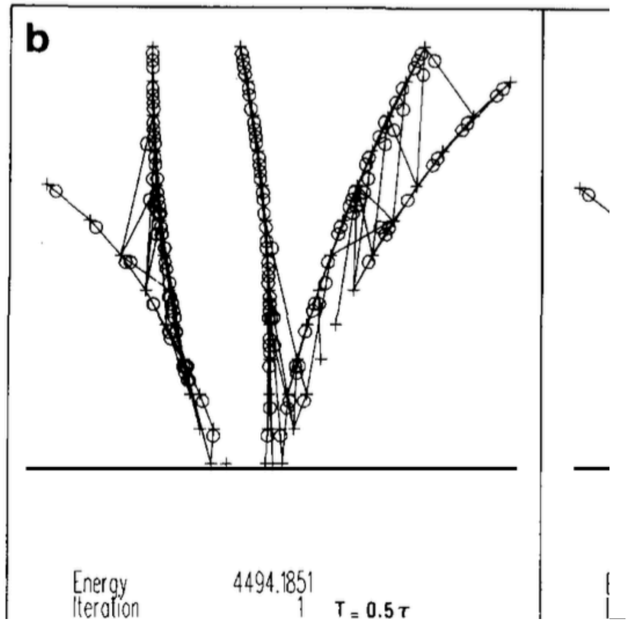
B. DENBY

Laboratoire de l'Accélérateur Linéaire, Orsay, France

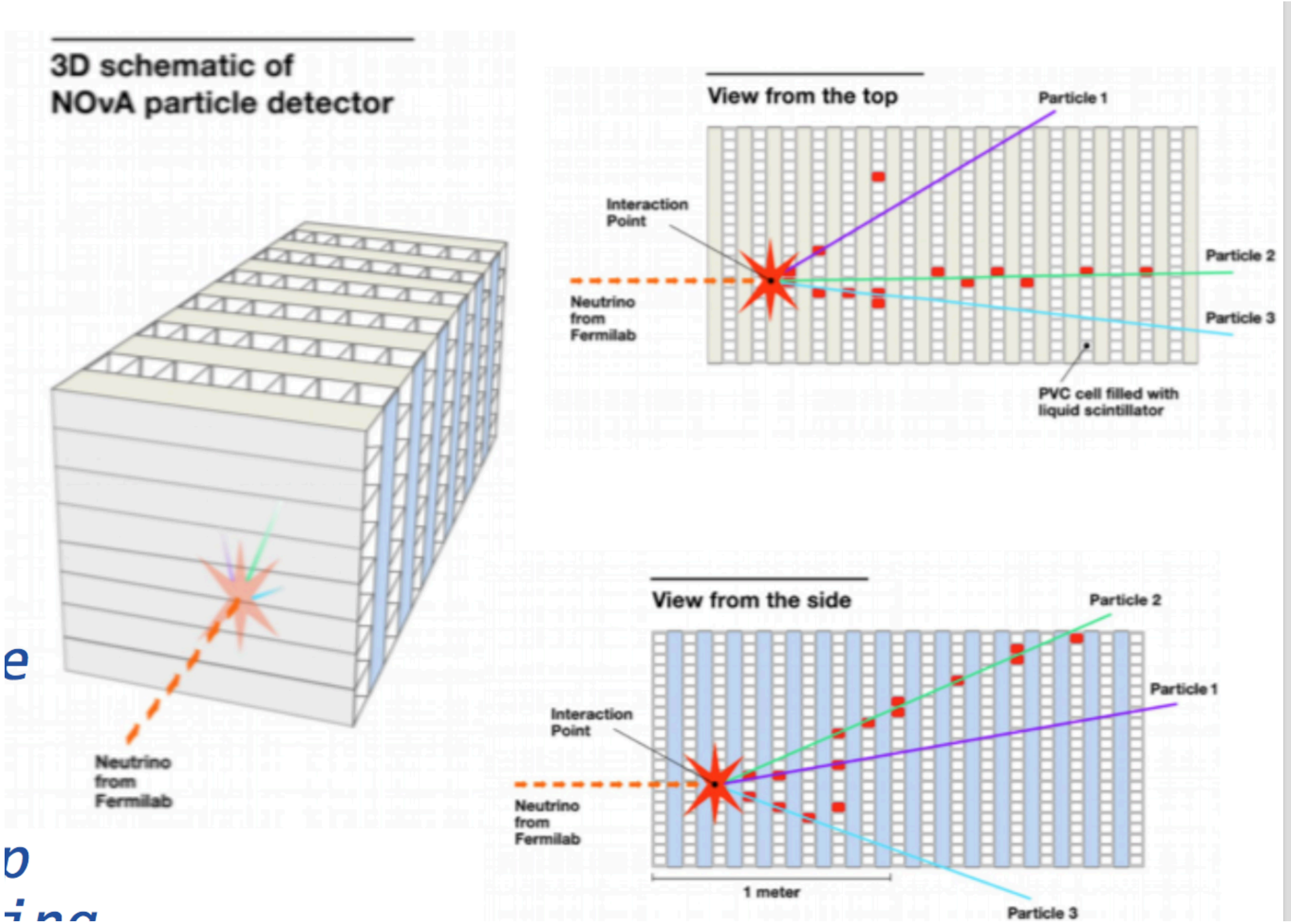
Received 20 September 1987; in revised form 28 December 1987

Bruce Denby

- 1987 Very first Neural Net in HEP paper known
- NN for tracking and calo clustering
- B. Denby then moved from Delphi at LEP to CDF at Tevatron. He still active outside HEP: 2017 analysis of ultrasonic image of the tongue
- 1992 JetNet Carsten Peterson, Thorsteinn Rognvaldsson (Lund U.) , Leif Lonnblad (CERN) (~500 citations) really started NN use in HEP



Deep Learning success : NOVA nu

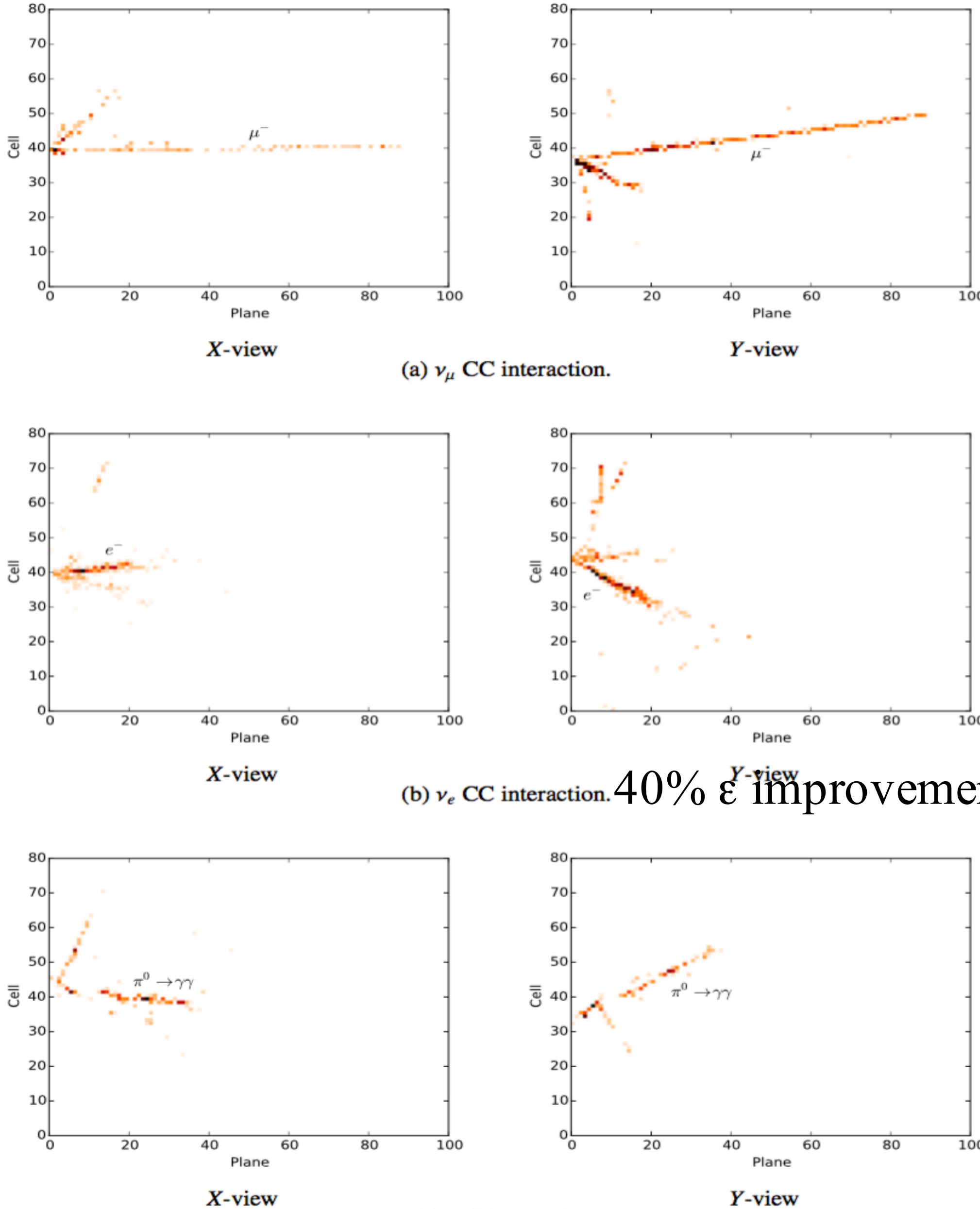


10

11

A recent success with ν : NOVA

arXiv 1604.01444 Aurisano et al



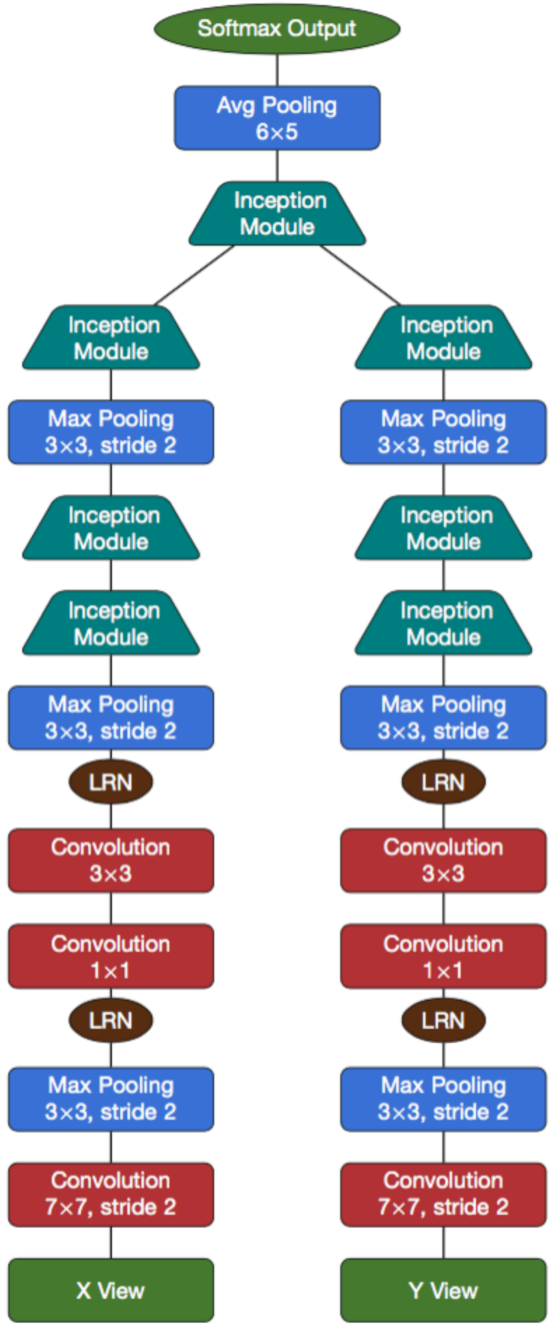
(a) ν_μ CC interaction.

(b) ν_e CC interaction.

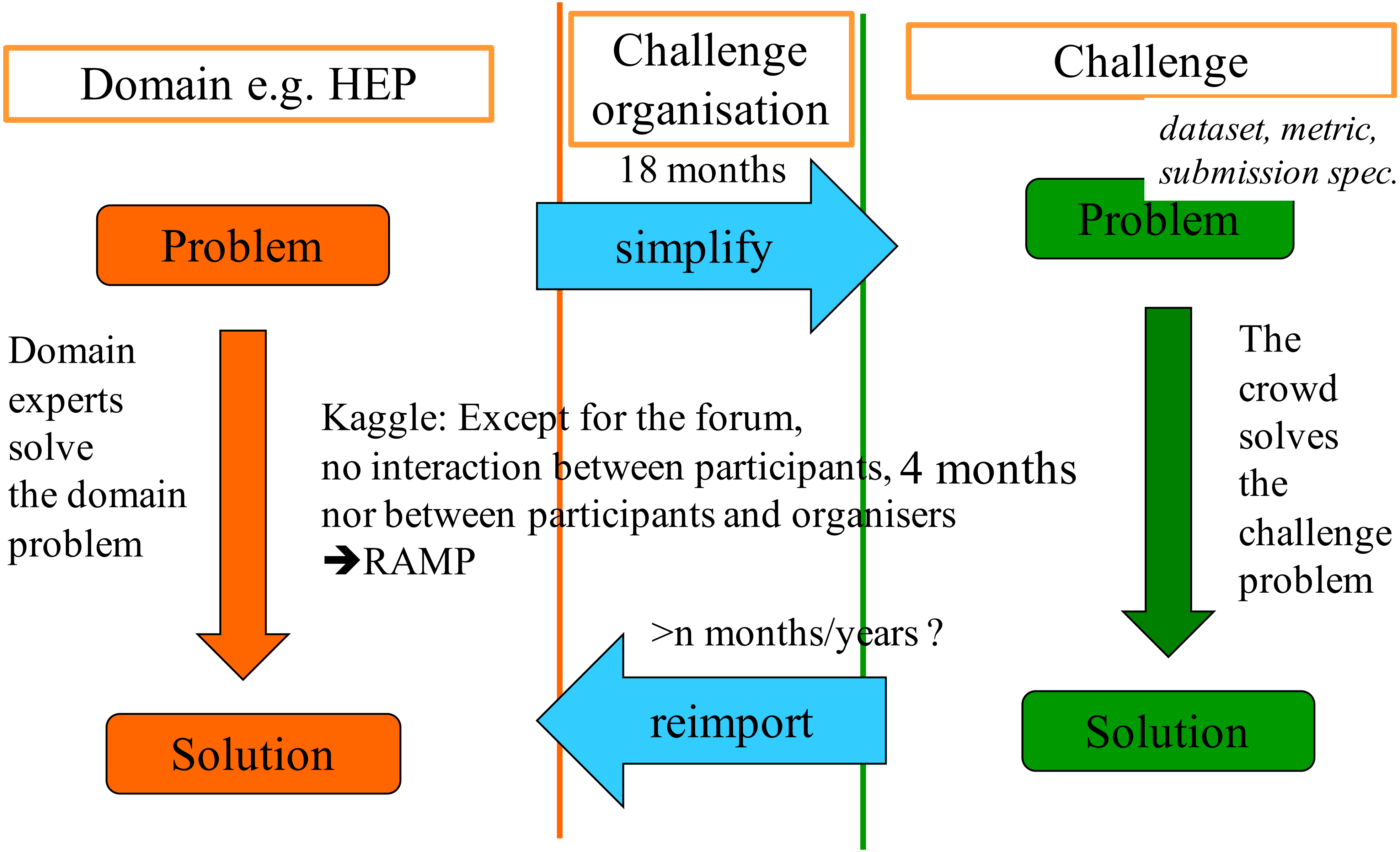
(c) NC interaction.

40% ε improvement

Neutrino interaction classification
Using Convolutional Neural Network (GoogleNet)
Actually used in physics results 1703.03328 and 1706.04592



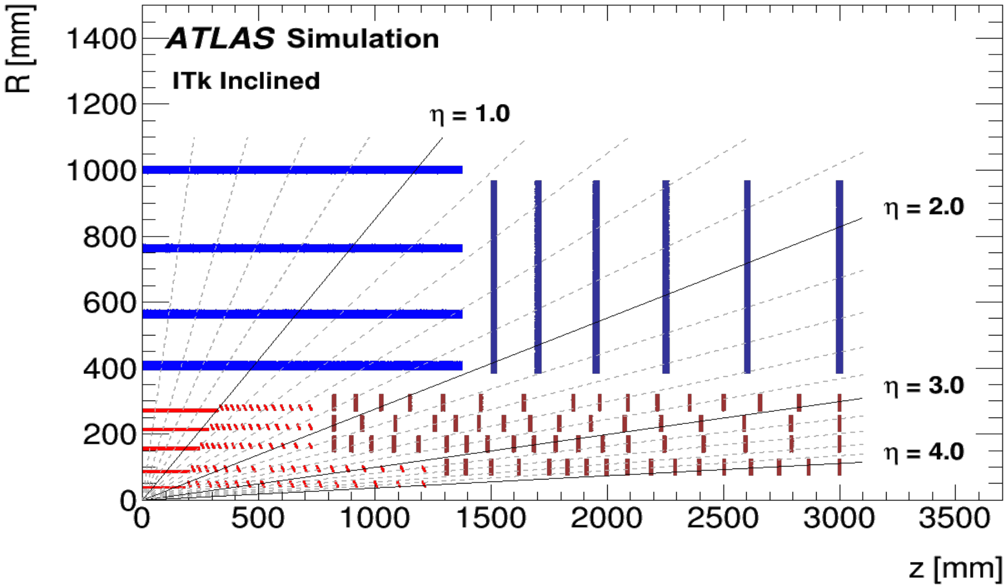
From domain to challenge and back

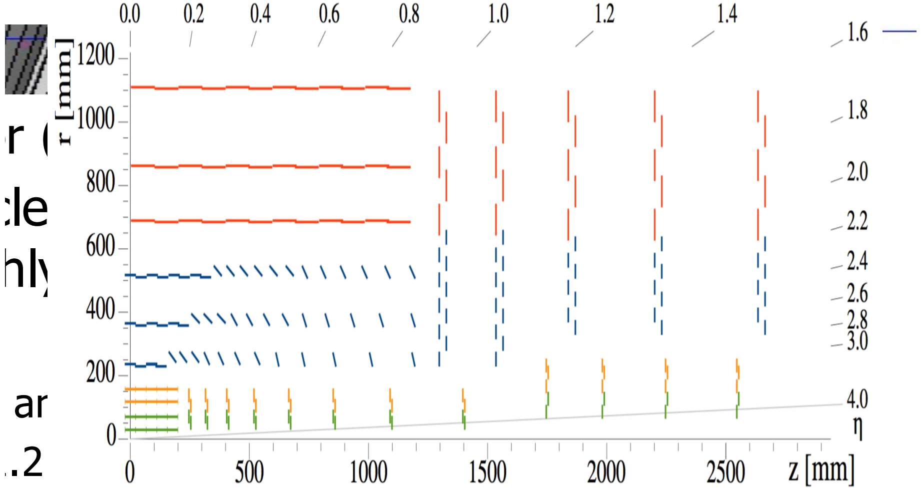


TrackML in a nutshell

- Accurate simulation engine (ACTS https://gitlab.cern.ch/acts/acts-core) to produce realistic events
  - One file with list of 3D points
  - Ground truth : one file with point to particle association
  - Ground truth auxiliary : true particle parameter (origin, direction, curvature)
  - Typical events with ~200 parasitic collisions (~10.000 tracks/event)
- Large training sample 10k events, 0.1 billion tracks, 1 billion points, ~100GByte
- Accuracy phase (May to August 2018) on Kaggle
  - Participants are given the test sample (with usual split for public and private leaderboard) and run the evaluation to find the tracks
  - They should upload the tracks they have found
  - A track is a list of 3D points
  - Score : fraction of points correctly grouped together
  - Evaluation on test sample with per-mille precision on 100 event
- Throughput phase Sep to Nov 2018 on Codalab
  - Strong CPU incentive

’ layout





r (

cle

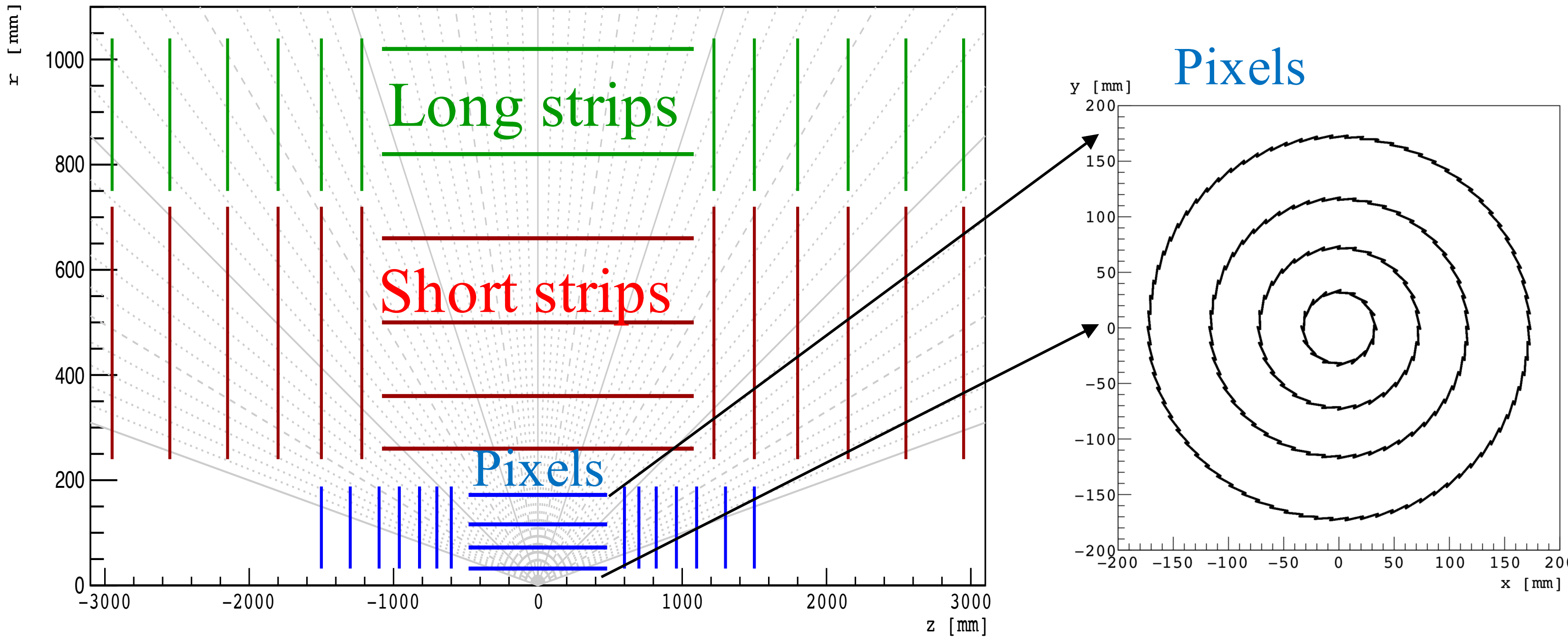
hly

ar

.2

0.8 mm, digital clusters: $\sigma \sim 25\ \mu m \times 3.1mm$

- (note the measurement anisotropy)



Datasets

TrackML

❑ Hit file (measured position mm)

	hit_id	x	y	z	volume_id	layer_id	module_id
0	1	-62.663200	-3.05090	-1502.5	7	2	1
1	2	-66.124702	-1.36730	-1502.5	7	2	1
2	3	-63.697701	1.73267	-1502.5	7	2	1
3	4	-82.501801	-14.09150	-1502.5	7	2	1
4	5	-74.343399	0.84469	-1502.5	7	2	1

❑ Truth file (true position mm particle momentum GeV)

	hit_id	particle_id	tx	ty	tz	tpx	tpy	tpz	weight
0	1	328762978956476416	-62.661499	-3.048720	-1502.5	-1.025760	-0.032316	-24.53690	0.000014
[illegible]	[illegible]	[illegible]	-66.123901	-1.376350	-1502.5	-0.634752	0.007755	-14.21880	0.000008
2	3	72094565116411904	-63.690601	1.726280	-1502.5	-0.826153	0.040302	-19.25260	0.000013
3	4	238697583478833152	-82.507202	-14.093000	-1502.5	-0.244242	-0.062864	-4.57011	0.000006
4	5	0	-74.342796	0.844152	-1502.5	-166440.000000	2483.800049	-986048.00000	0.000000

k to scroll output; double click to hide

Datasets

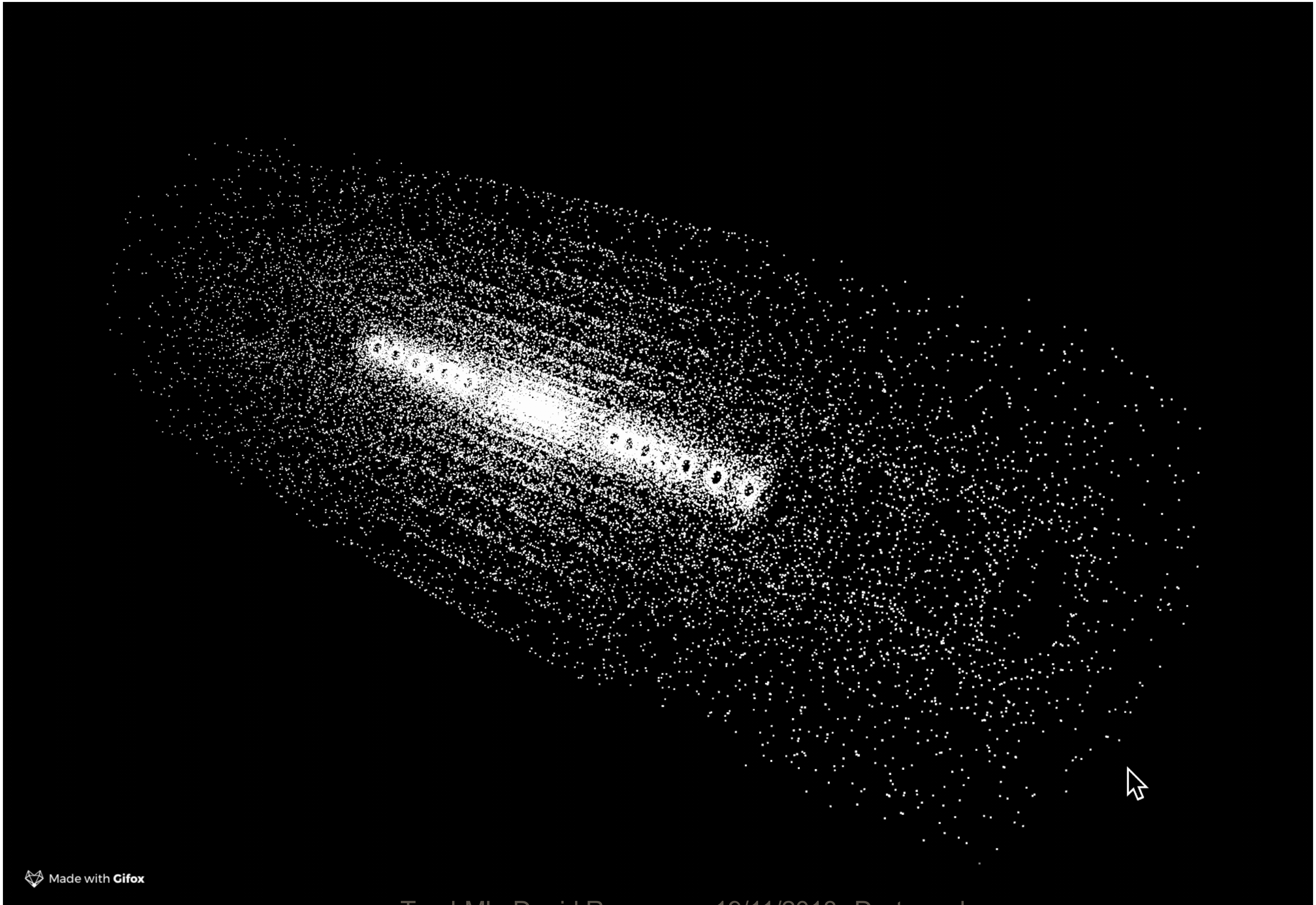
- Particle file   origin vertex (mm) momentum (GeV) charge

	particle_id	vx	vy	vz	px	py	pz	q	nhits
0	4503668346847232	-0.024934	-0.014566	-11.263	-0.055269	0.323272	-0.203492	-1	3
1	4503737066323968	-0.024934	-0.014566	-11.263	-0.948125	0.470892	2.010060	1	10
2	4503805785800704	-0.024934	-0.014566	-11.263	-0.886484	0.105749	0.683881	-1	10
3	4503874505277440	-0.024934	-0.014566	-11.263	0.257539	-0.676718	0.991616	1	11
4	4503943224754176	-0.024934	-0.014566	-11.263	16.439400	-15.548900	-39.824902	1	11

(note : we do not ask participant to reconstruct these track parameters but these could be useful latent variables)

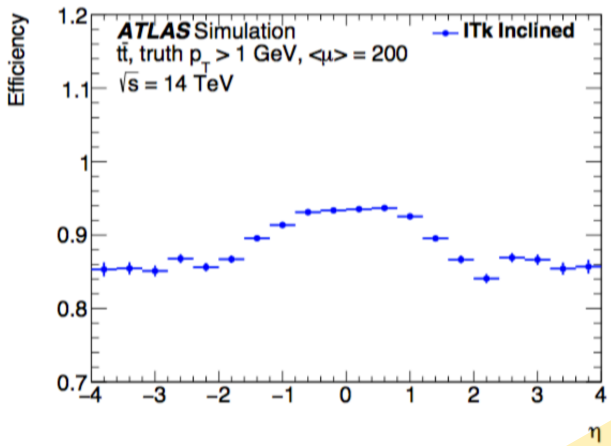
- (static)Detector file   center position (mm)   3x3 rotation matrix

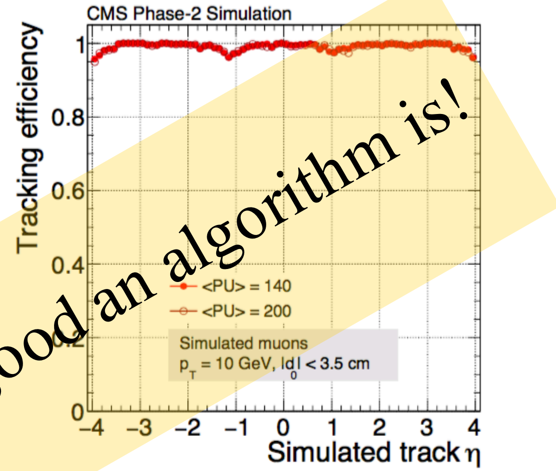
	volume_id	layer_id	module_id	cx	cy	cz	rot_xu	rot_xv	rot_xw	ro
0	6	2	1	-65.7965	-5.17830	-1502.5	0.078459	-0.996917	0.0	-0.99
1	6	2	2	-139.8510	-6.46568	-1502.0	0.046183	-0.998933	0.0	-0.99
2	6	2	3	-138.6570	-19.34190	-1498.0	0.138156	-0.990410	0.0	-0.99
3	6	2	4	-64.1764	-15.40740	-1498.0	0.233445	-0.972370	0.0	-0.97

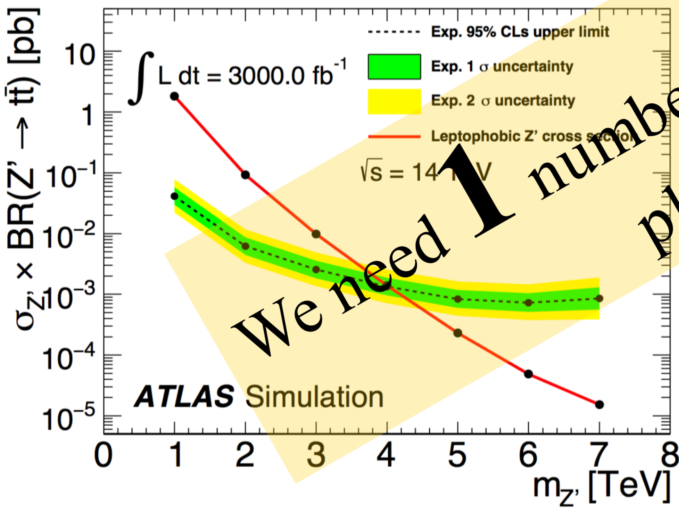

Made with Gifox

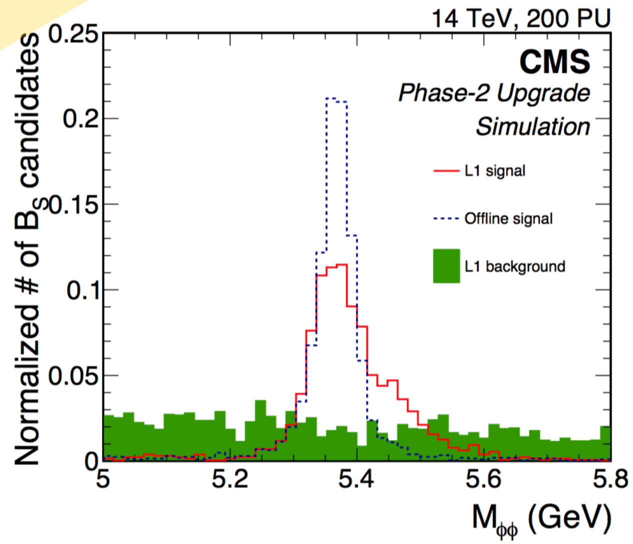
Score

- 2017 CMS tracker Technical Design Report : Chapter 6 expected performance 31 pages 58 figures
- ATLAS Si strip Technical Design Report Chapter 4 ITk Performance and Physics Benchmark Studies 54 page





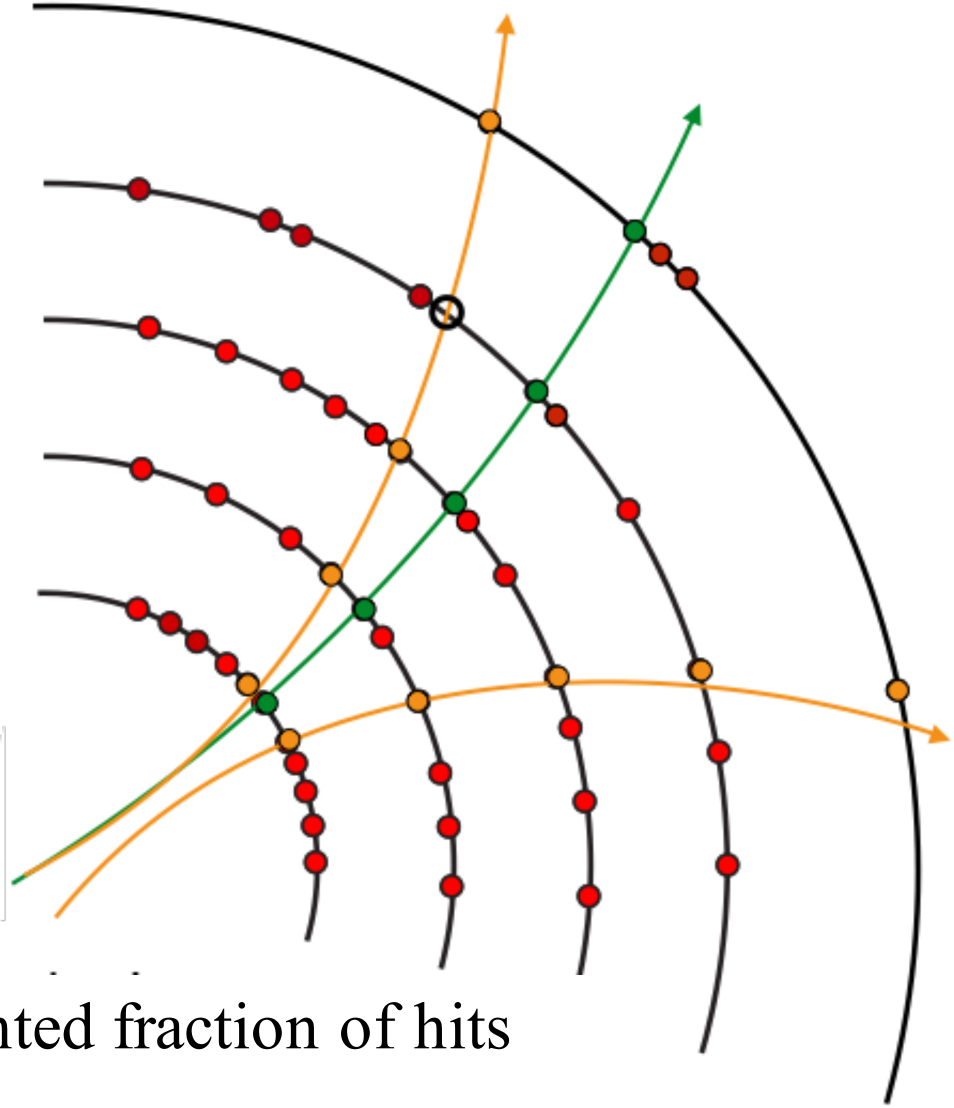




We need 1 number to specify how good an algorithm is!
plus CPU time

Track evaluation

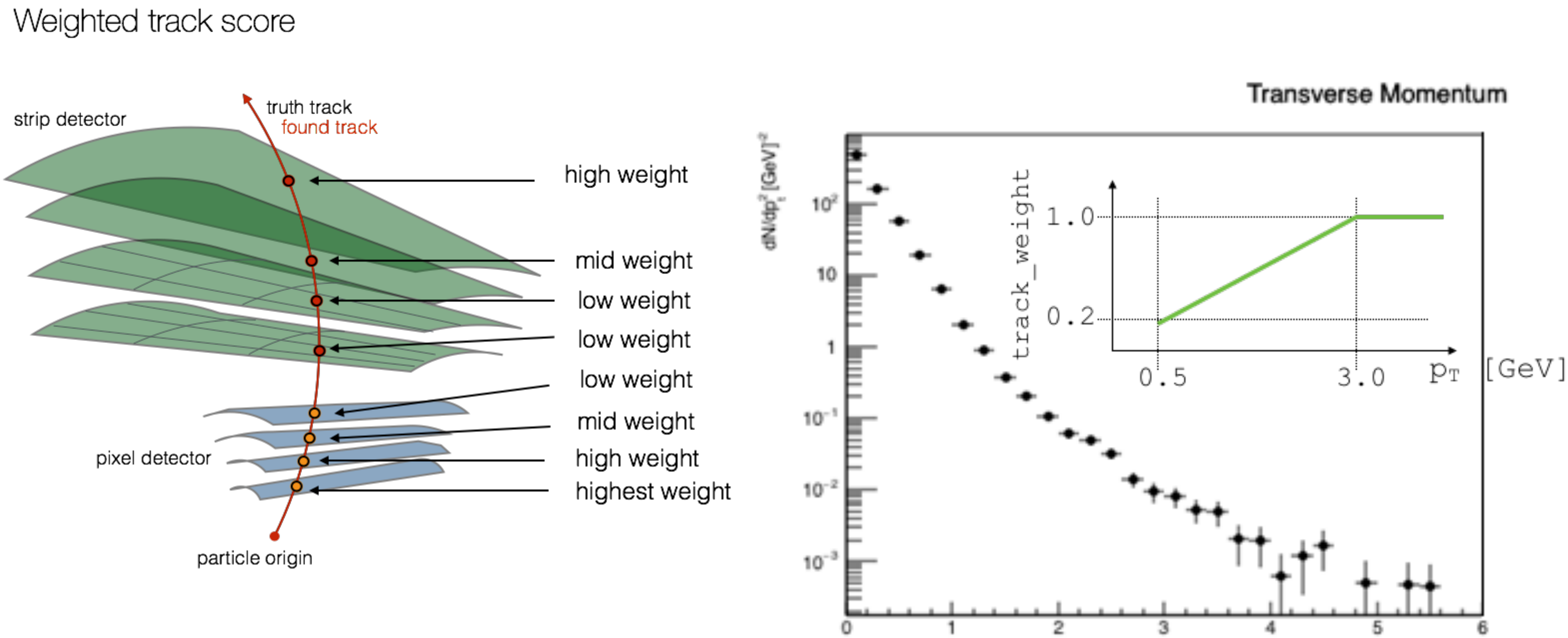
good track	not so good track
many compatible hits	short tracks
completeness	holes
uniqueness	shared hits
low χ^2/ndf	bad fit quality, outliers
small impact parameter (for primaries)	
clusters are compatible	



Big decision : score is ~ « the weighted fraction of hits correctly associated »

Hit weighting

- Define : weight=weight$_{order}$ x weight$_{pt}$



- Weight$_{order}$: more emphasis on first and last hits
- Weight$_{pt}$: more emphasis on high pT tracks
- Weight=0 for noise hits or hits from particle with <=3 hits

Track scoring

- Overall scoring defined at hit level
- Loop on reco tracks
  - Require >50% of hits from same true particle
  - Require >50% of hits from this true particle in this reco track
  - At this point 1⇔1 relationship between true and reco tracks
  - Sum the weights of the intersection (hits belonging both to true and reco track)
- Event score normalised to the sum of weights of all the hits
  - ➔ ideal algorithm has score==1.
- Final score averaged of 100 events➔statistical precision ~0.1%

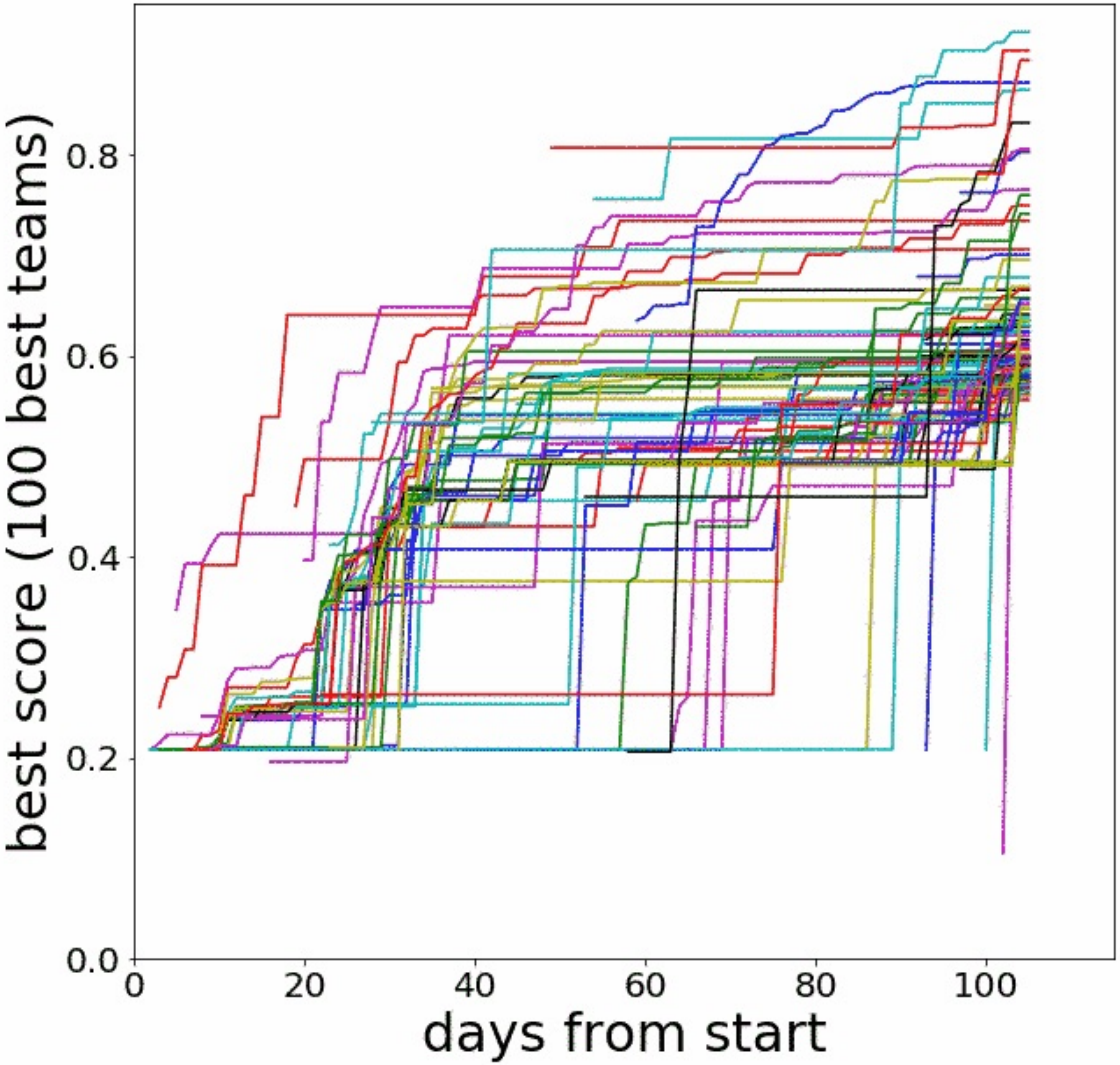
Real life vs challenge

1. Wide type of physics events
2. Full detailed Geant 4 / data
3. Detailed dead matter description
4. Complex geometry (tilted modules, double layers, misalignments…)
5. Hit merging
6. Allow shared hits
7. Output is hit clustering, track parameter and covariance matrix
8. Multiple metrics (see TDR's)

1. One event type (ttbar)
2. ACTS (MS, energy loss, hadronic interaction, solenoidal magnetic field, inefficiency)
3. Cylinders and slabs
4. Simple, ideal, geometry (cylinders and disks)
5. No hit merging
6. Disallow shared hits
7. Output is hit clustering
8. Single number metrics

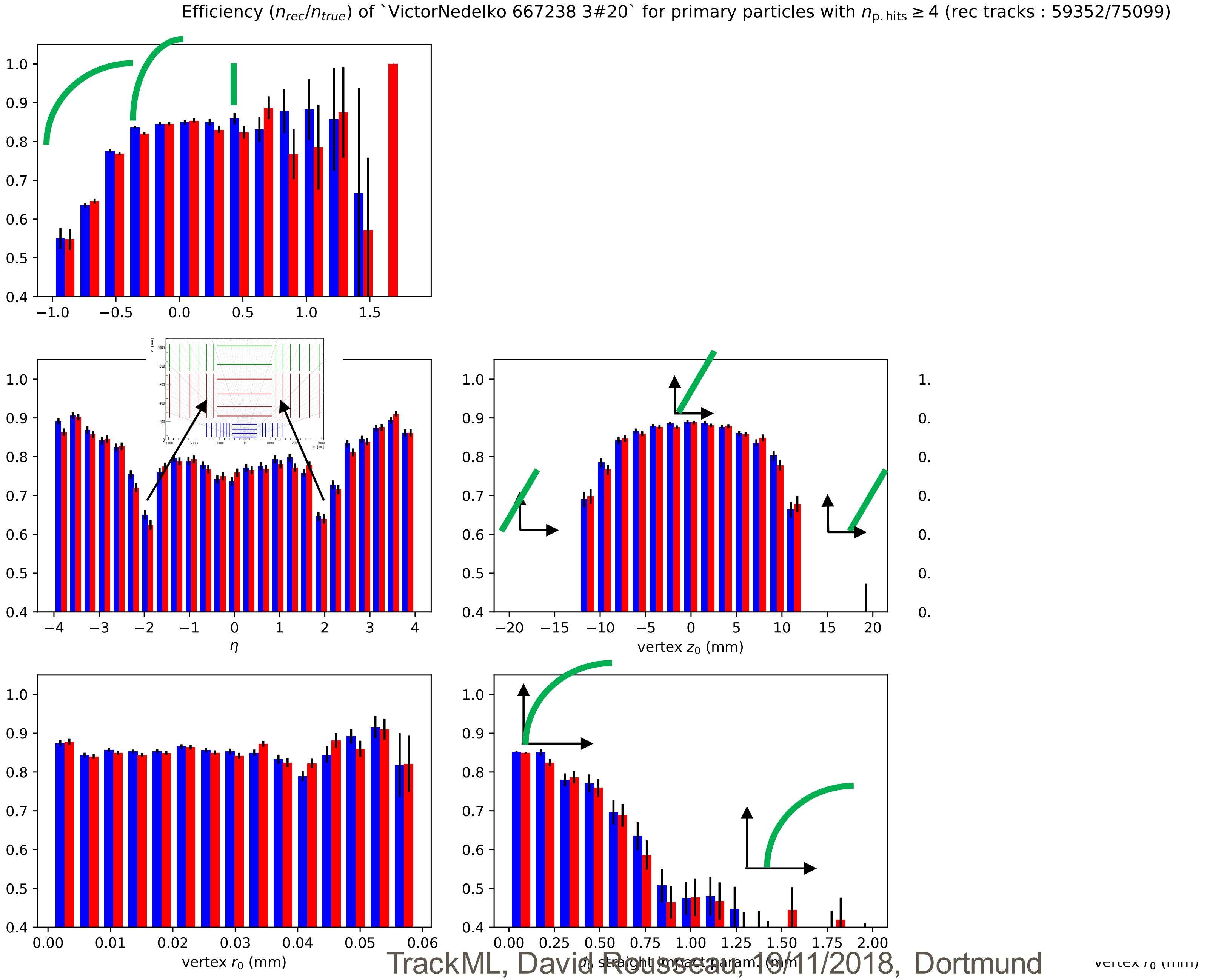
Simpler, but not too simple!

Evolution of leaderboard

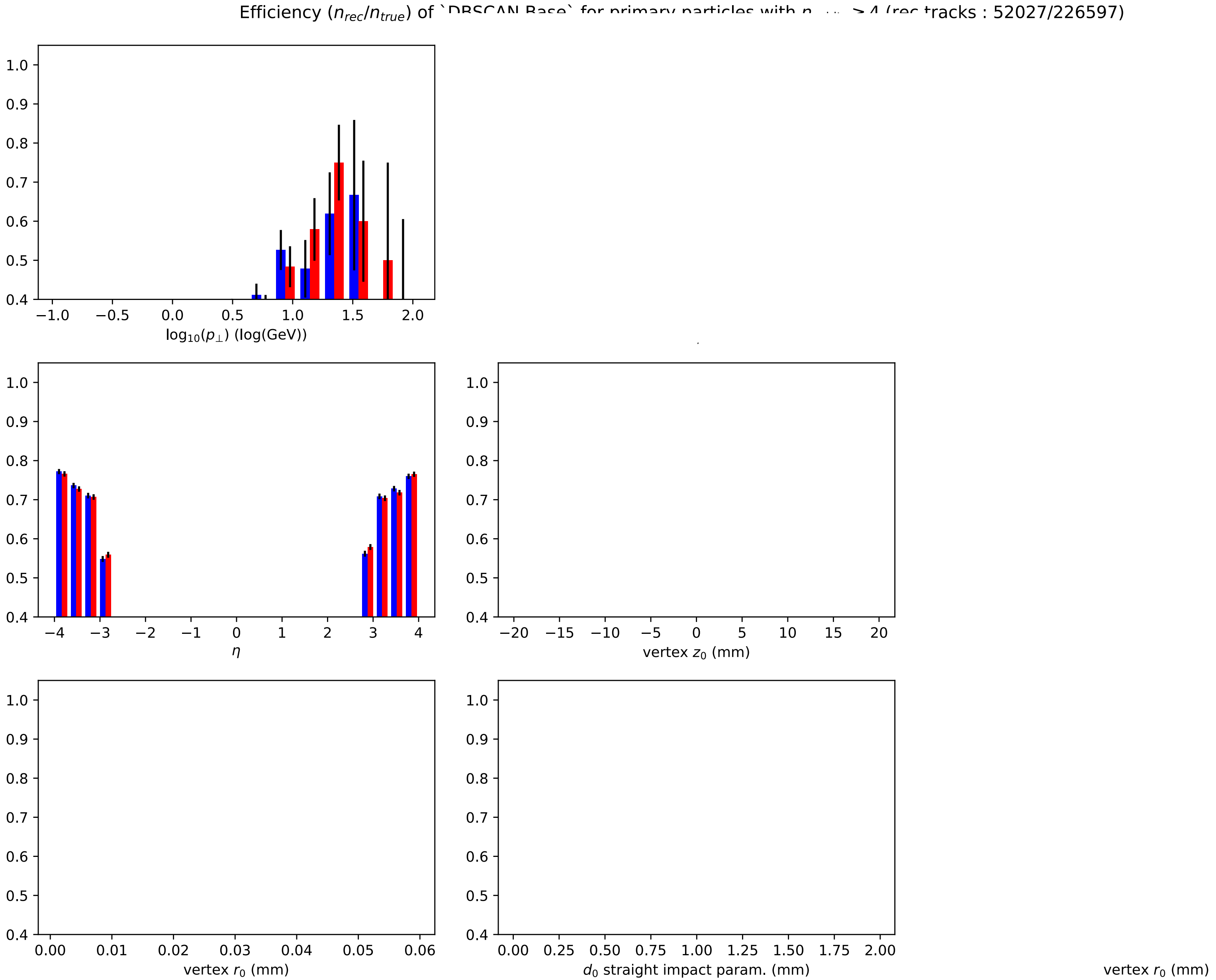


1	—	Top Quarks	0.92182	10	19d
2	—	outrunner	0.90302	9	18d
3	—	Sergey Gorbunov	0.89353	6	18d
4	—	demelian	0.87079	35	1mo
5	—	Edwin Steiner	0.86395	5	18d
6	—	Komaki	0.83127	22	18d
7	—	Yuval & Trian	0.80414	56	18d
8	—	bestfitting	0.80341	6	18d
9	—	DBSCAN forever	0.80114	23	18d
10	—	Zidmie & KhaVo	0.76320	26	18d
11	—	Andrea Lonza	0.75845	15	18d
12	—	Finnies	0.74827	56	18d
13	—	Rei Matsuzaki	0.74035	12	18d
14	—	Mickey	0.73217	10	2mo
15	—	Vicens Gaitan	0.70429	19	1mo
16	—	Robert	0.69955	3	21d
17	—	Yuval-CPMP tribute band	0.69364	20	20d
18	—	N. Hi. Bouzu	0.67573	9	22d
19	—	Steins;Gate	0.66763	12	19d
20	▲1	Victor Nedel'ko	0.66723	4	2mo

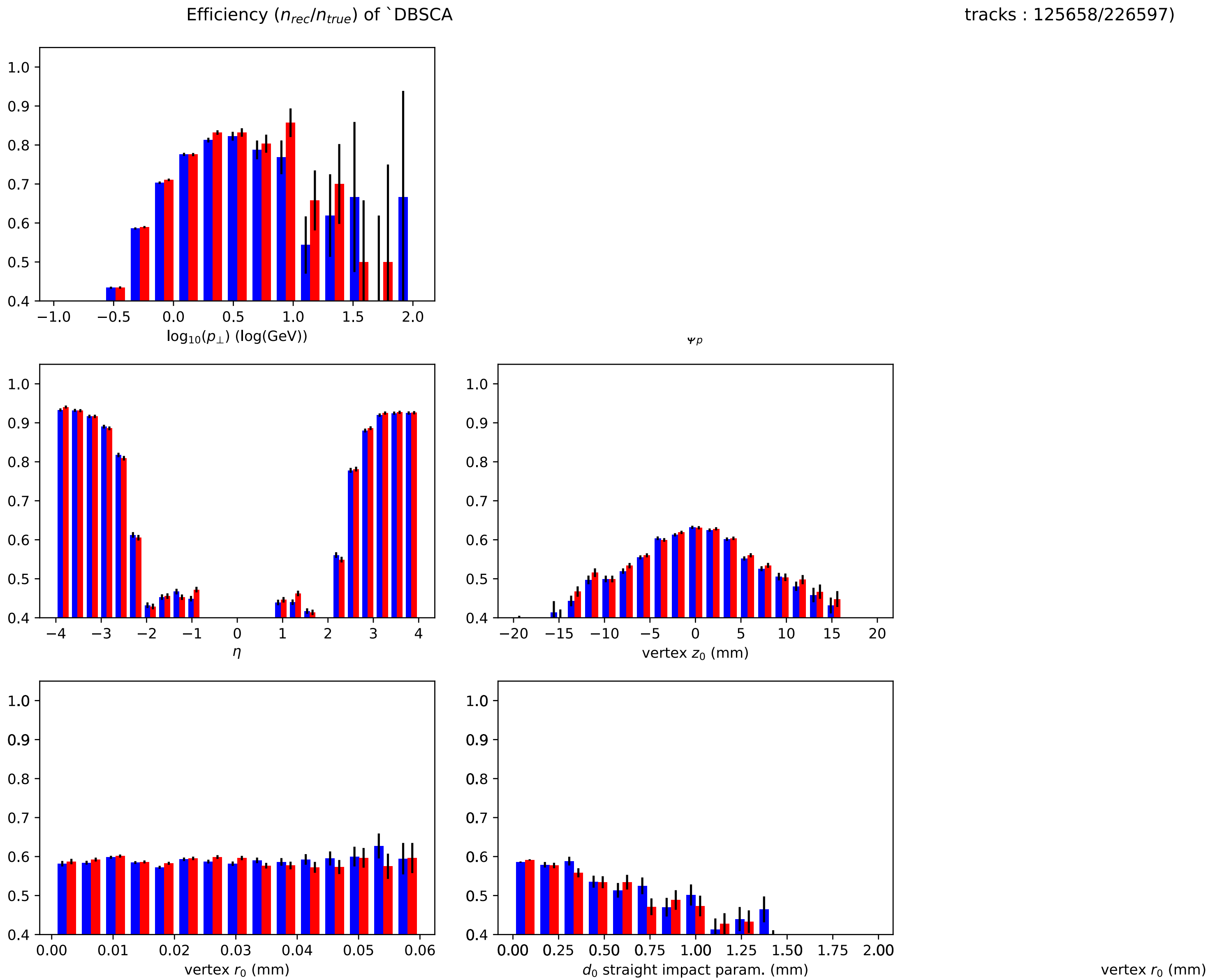
Primary track efficiency Nedelko #20



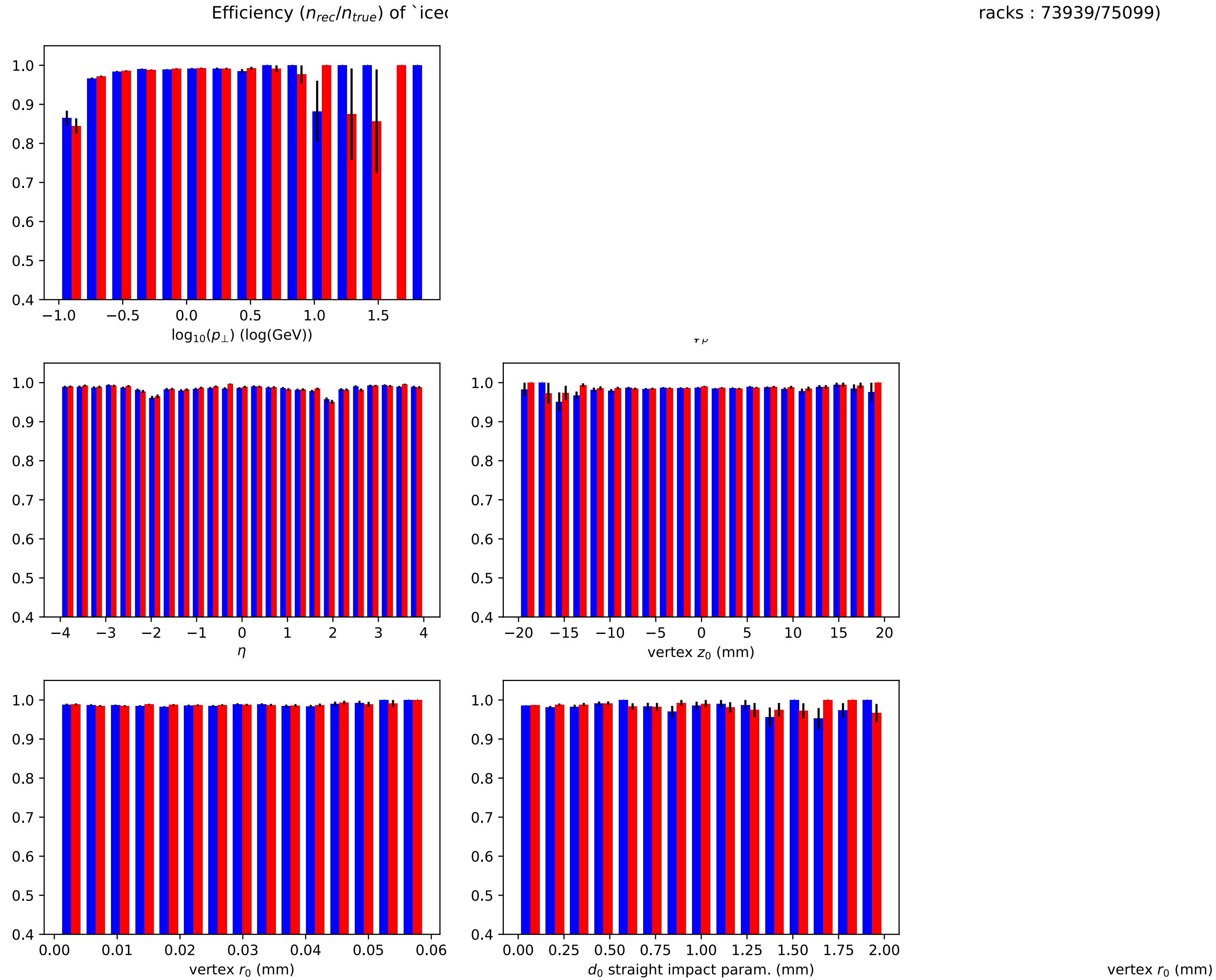
Primary track efficiency : starting



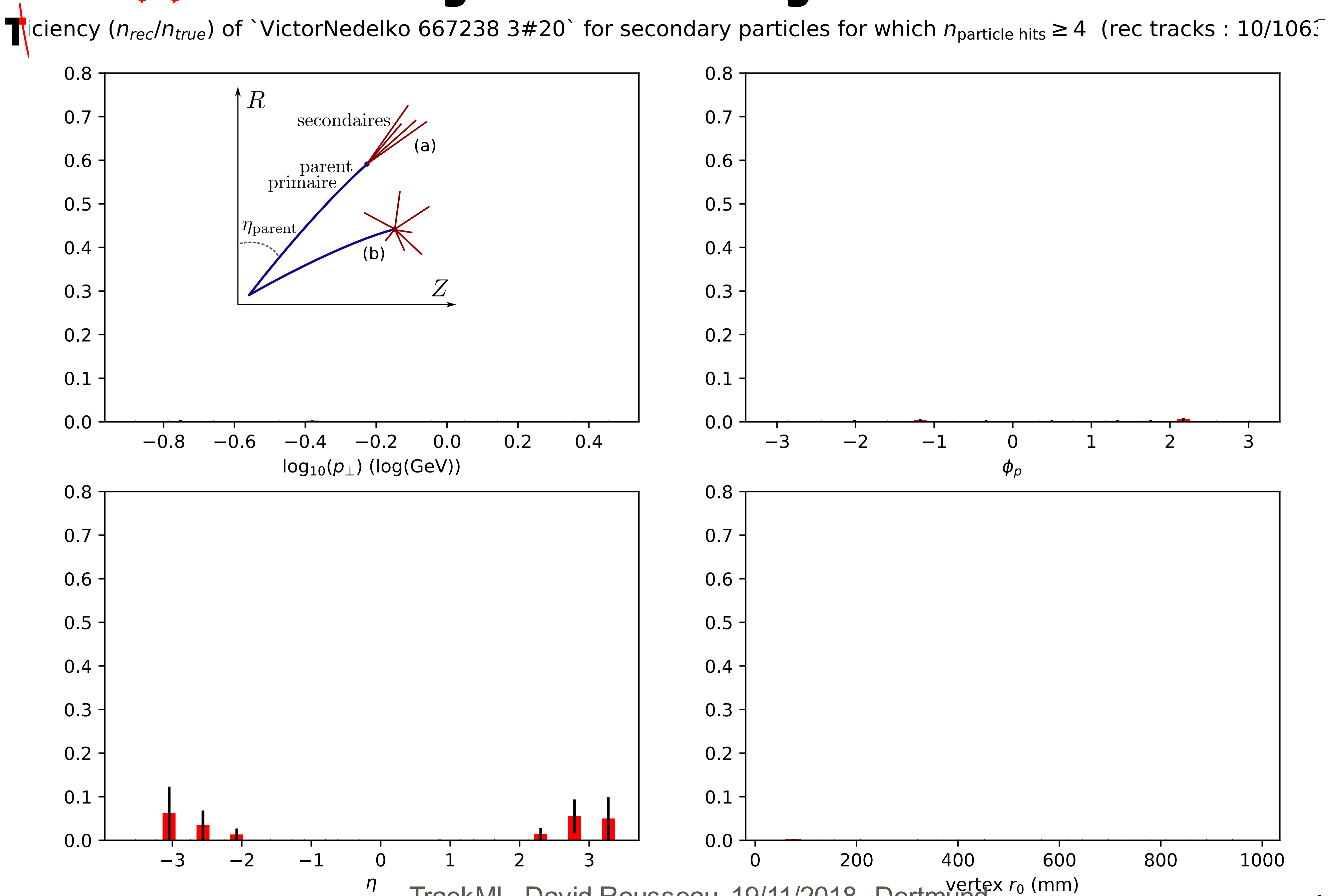
Primary track efficiency : best after 3 weeks



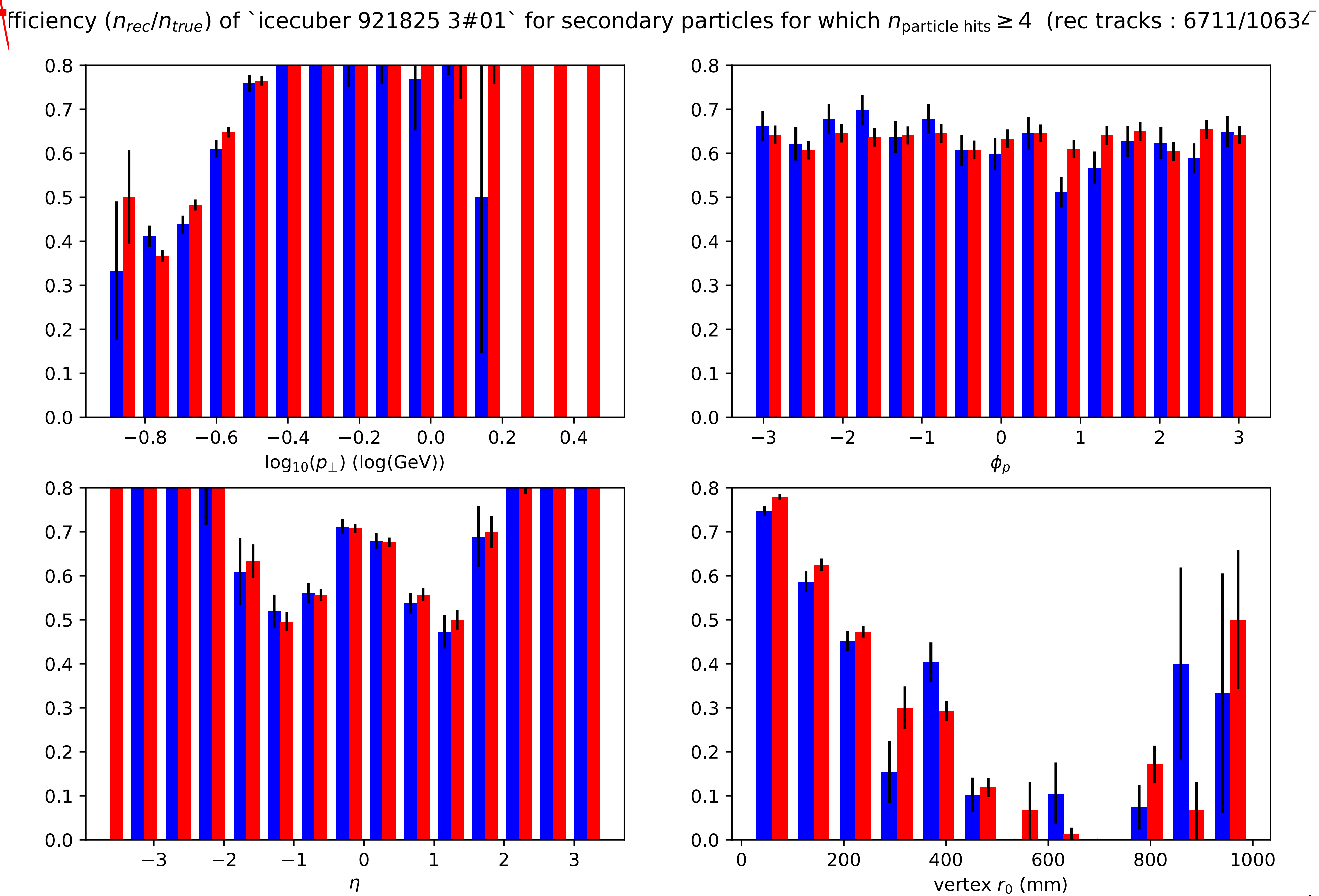
Primary track efficiency TopQuarks #1



Secondary efficiency Nedelko #20



Secondary efficiency TopQuarks #1



A few competitors

Wins 12000$

- icecube #1 92.2 % (master student) : combinatorial approach, with a bit of ML

Wins 8000$

- outrunner #2 90.3% Deep Learning approach
  - Very innovative!
  - But brute force : takes one full day per event !
  - However code is using naïve python nested loops

Wins 5000$

- Sergey Gorbunov #3 89.4% demelian #4 87.1% : HEP tracking trigger experts

Wins Nvidia V100

- Yuval & Trian #7 80.4% : innovative clustering

Wins CERN invite

- CPMP #9 80.1% : DBSCAN unsupervised clustering algorithm
  - we gave DBSCAN in starting kit, with a 20% score, because in only required a few lines

Win NeurIPS invite

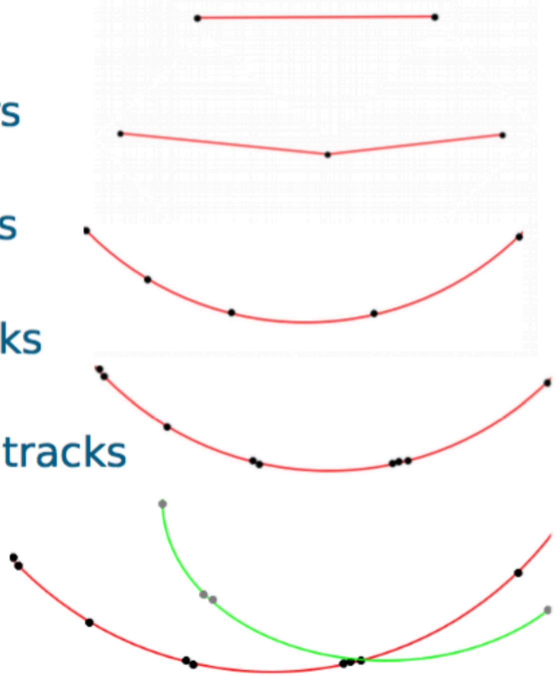
- Finnies #12 74.8% : use LSTM

Phase 1 Top Quarks

	Wall clock time	Peak memory usage
Average	7m17s	2.78GB
Max	11m20s	4.07GB

Main steps

- Select promising pairs
  - 7 million / 0.99
- Extend pairs to triples
  - 12 million / 0.97
- Extend triples to tracks
  - 12 million / 0.95
- Add duplicate hits to tracks
  - 12 million / 0.96
- Merge [illegible] o tracks
  - 90% of hits / 0.92



Findings

- No magic formula
  - We won because we were fast to try out and implement many ideas and got the details right
    - I once earned 0.03 (0.85→0.88) from fixing a tuning parameter
  - In other words: combination of many factors

- Logistic regression for track candidate pruning

- Pure C++, some scikit-learn for training

Phase 1 outrunner

"Wall clock time"
~1 day/event

Pure ML approach using python & Keras

- Event with **N** hits
- predict **N x N** relationships between hits, connect pairs when their probability is 1 (rather than 0)

Training:

- **5** hidden layers with **4k - 2k - 2k - 2k - 1k**
- **27** input variables per pair:
  x, y , z, counts, sum(cells.value) per hit
  two unit vectors per hit for direction from cell information
  4 parameters for linear (z_0) and helical compatibility

Prediction:

- predict relationship probability

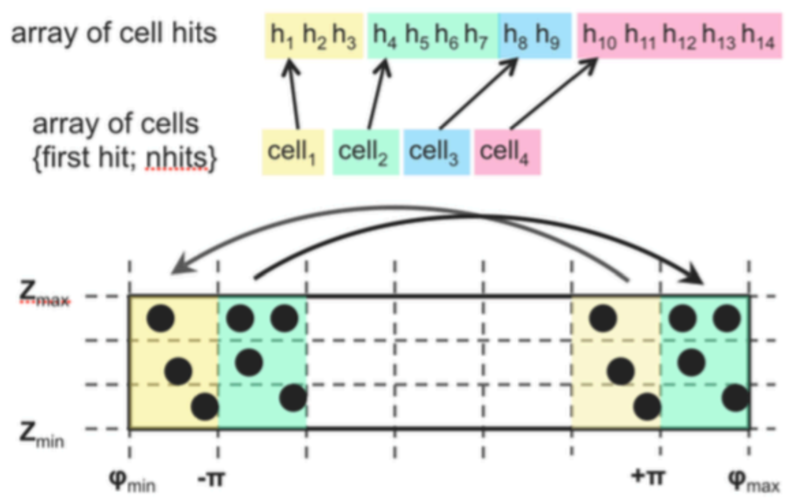
Reconstruct

- starting from one hit, find highest probability pair, then add pairwise hits
- test new hit for compatibility

Phase 1 Sergey Gorbunov

Execution time
1.2 min on single core 2.6 GHz CPU

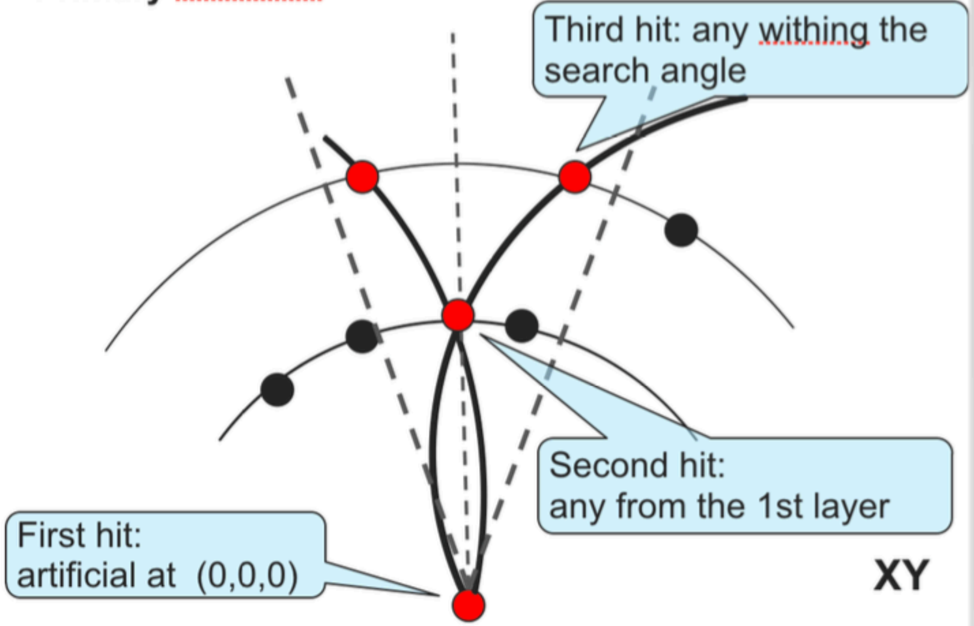
Regular grid with overlaps



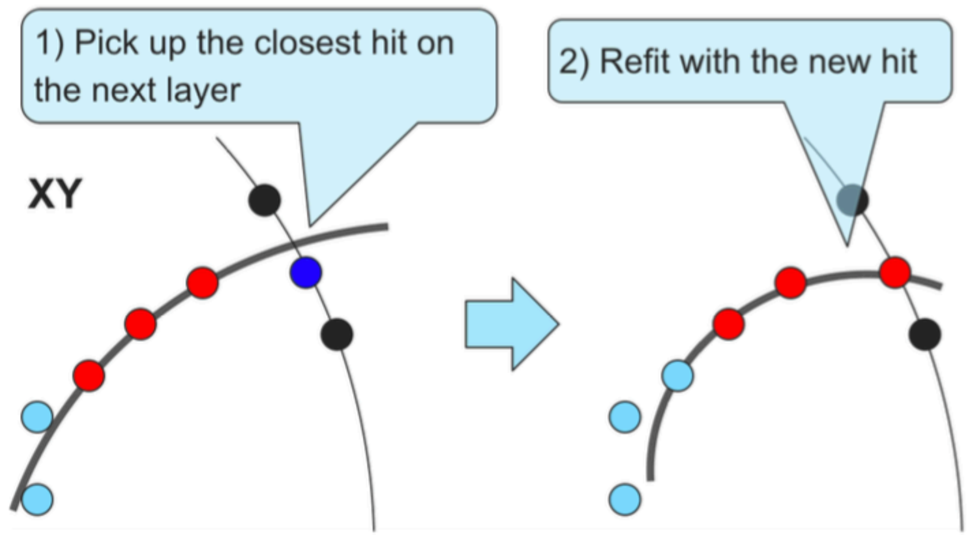
Summary

- A combinatorial algorithm, based on the track following method
- No search branches
- Simple track model: local 3-hit helix
- Fast data access

Primary tracklets

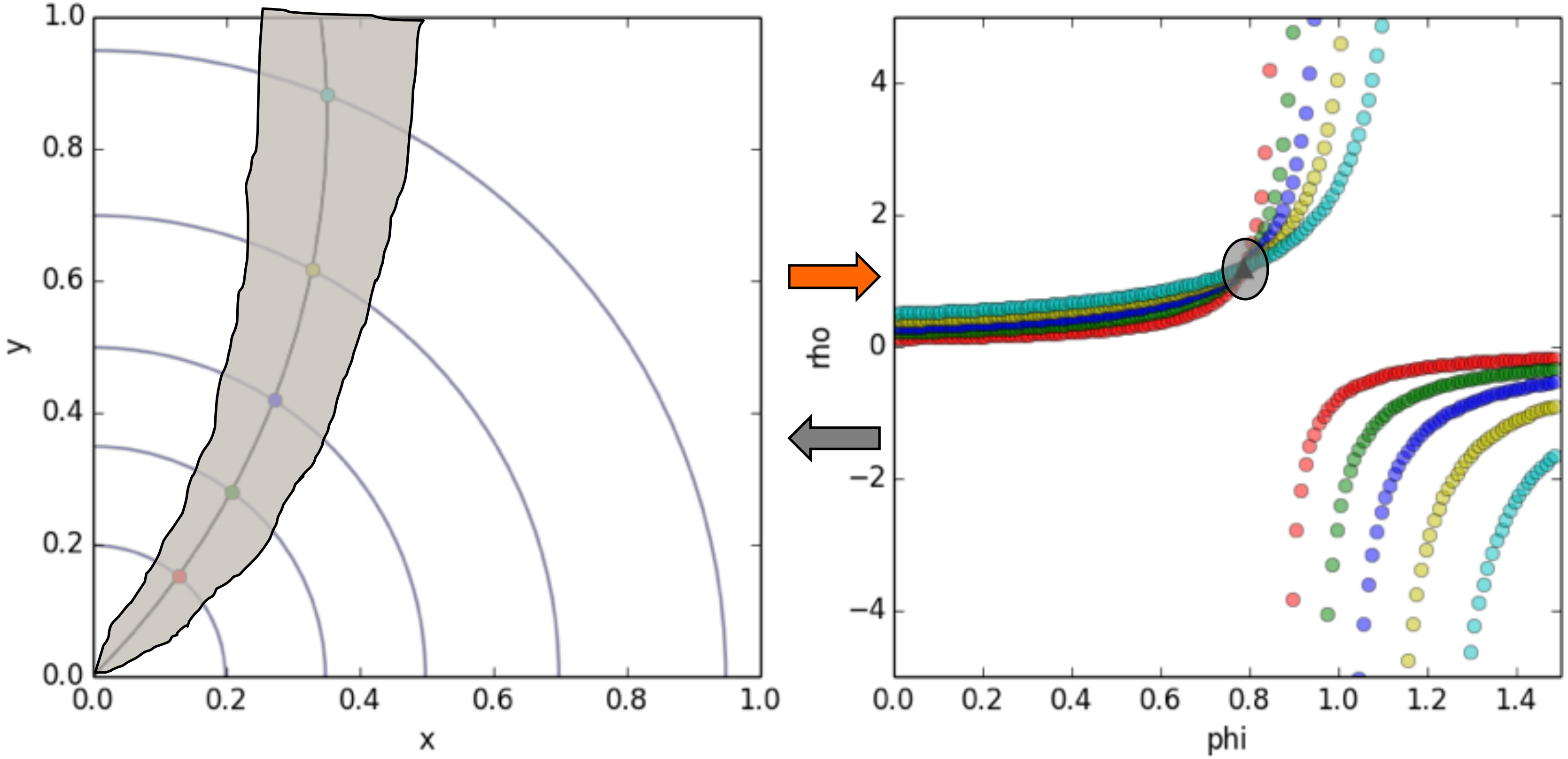


Prolongation of tracklets



#7 Yuval & Trian

Hough transform with 2 parameters:



- Now one could do :
  1. Randomly select a curvature rho
  2. Histogram phi for this rho
  3. =>pick the tracks for this rho
  4. Merge with tracks already found
  5. Iterate to 1
- This is ~ what Yuval & Trian are doing
  - Randomise 2 parameters
  - Histogram in 3 dimensions

Throughput Phase

Launched 6th Sep 2018 until 12th March 2019 on Codalab

Dataset

- Not identical
- Detector is the same
- Simplification:
  - Only primary particles enter the scoring (much less particles not pointing approximately to 0 0 0)
- Features fix
  - Beam spot sigma_z 5.5mm➔5.5 cm
  - Module thickness halved
  - Looping particles removed
  - Electrons multiple scatterign fixed (was causing 0.5% « crazy » tracks

Schematic

CodaLab

	hit_id	x	y	z	volume_id	layer_id	module_id
0	1	-64.409897	-7.163700	-1502.5	7	2	1
1	2	-55.336102	0.635342	-1502.5	7	2	1
2	3	-83.830498	-1.143010	-1502.5	7	2	1
3	4	-96.109100	-8.241030	-1502.5	7	2	1

event(s) are loaded in memory

start

API to call python/C++ API

User executable

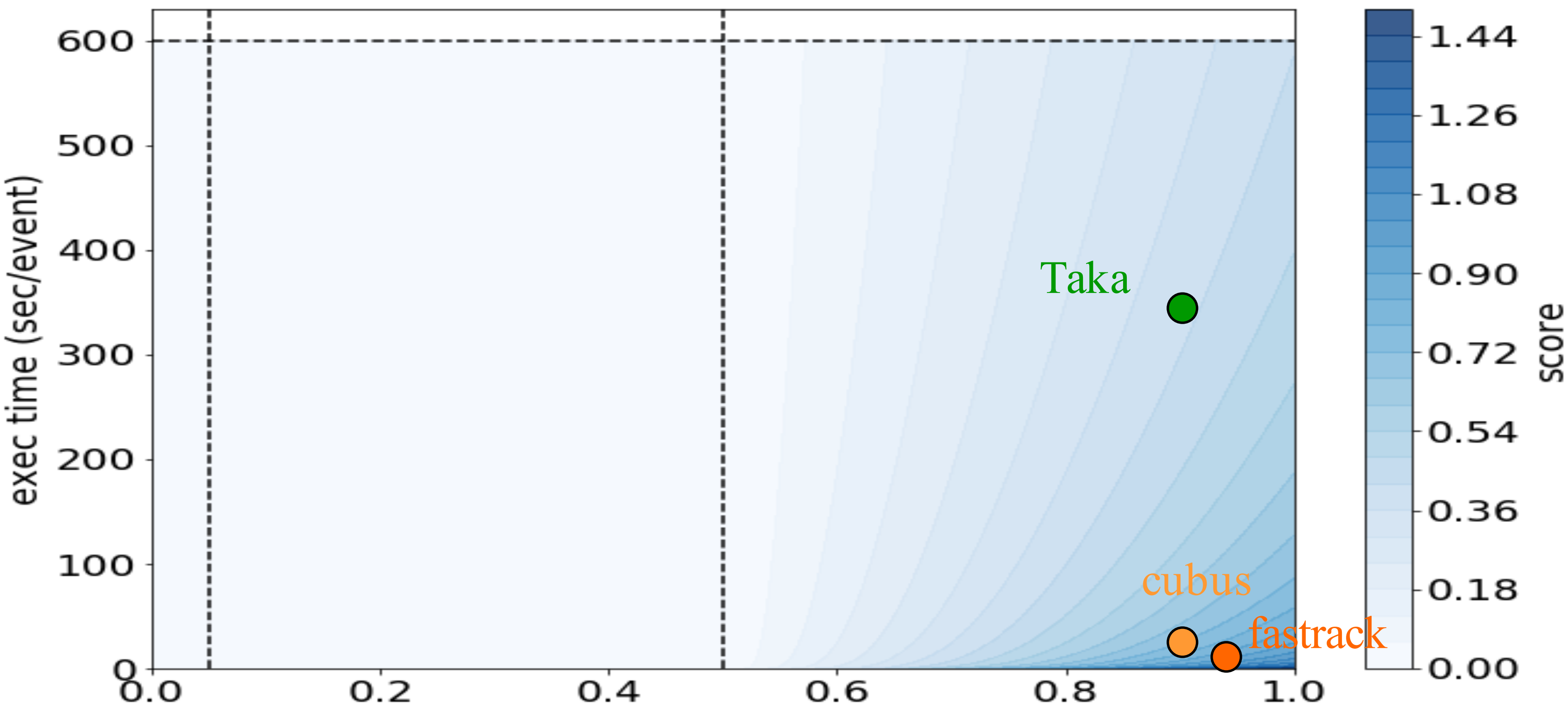
stop

solution

VM 2 cores, 4 Gb memory

Throughput scoring

- *Ranking score* :
  - 0 if time >600 s or accuracy <50%
  - $\sqrt{log}(1 + 600/time) * (accuracy - 0.5)^2$
- Documented software of first phase #1 #2 #3 #7 #9 #11 #12 released
  - Can be used as starting point but need retuning



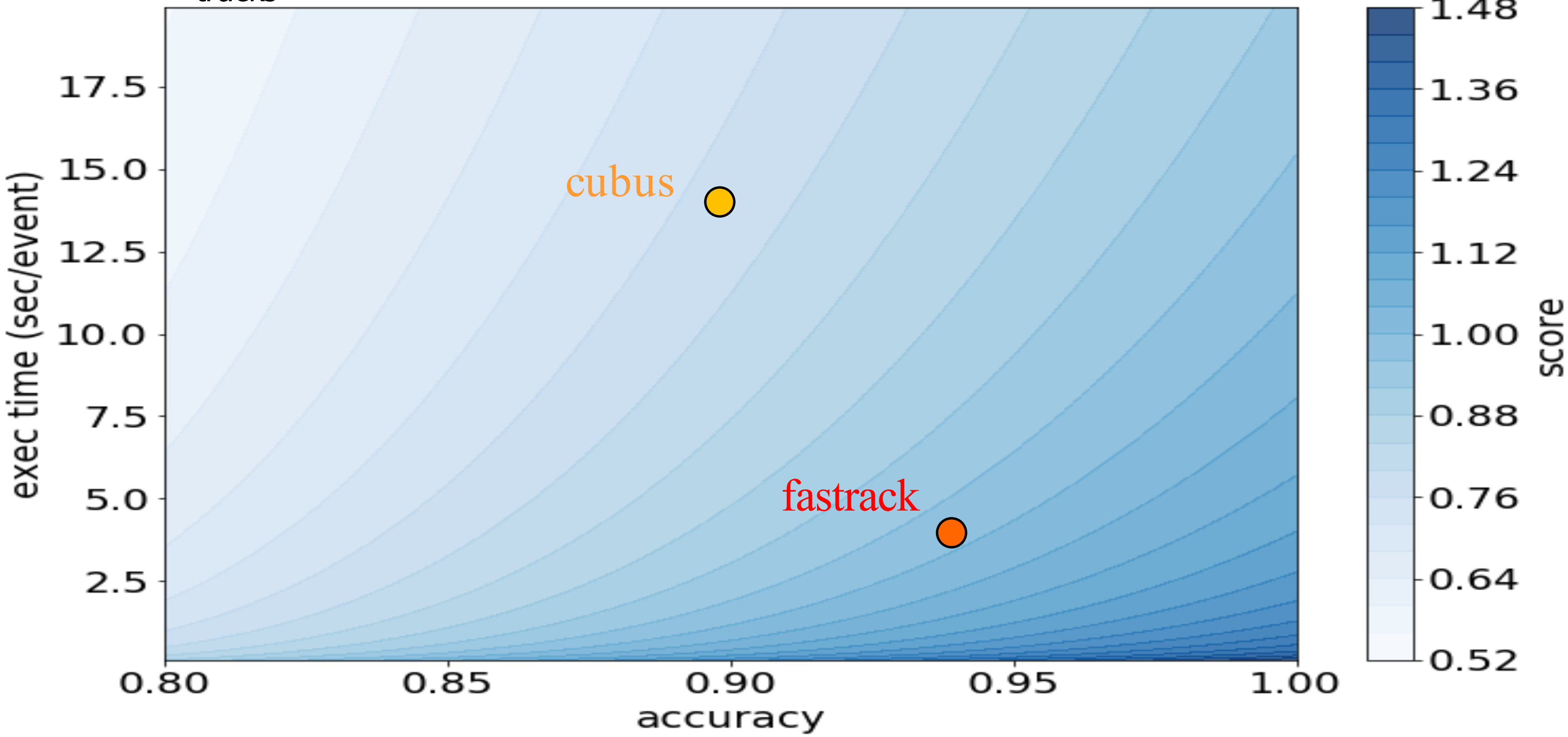
Throughput phase LB

RESULTS

#	User	Entries	Date of Last Entry	score ▲	accuracy_mean ▲	accuracy_std ▲	computation time (sec) ▲	computation speed (sec/event) ▲	Duration ▲
1	fastrack	22	10/19/18	1.0009 (1)	0.938 (1)	0.00 (7)	161.88 (10)	3.24 (10)	201.00 (6)
2	cubus	8	09/13/18	0.7719 (2)	0.895 (3)	0.01 (5)	675.35 (11)	13.51 (11)	724.00 (7)
3	Taka	8	10/20/18	0.3934 (3)	0.906 (2)	0.00 (6)	19321.21 (15)	386.42 (15)	19744.00 (12)
4	khavo	3	10/29/18	0.0000 (4)	0.304 (4)	0.03 (1)	18015.06 (14)	360.30 (14)	18419.00 (11)
5	traffic_congestion	2	10/21/18	0.0000 (4)	0.082 (7)	0.01 (4)	49.67 (9)	0.99 (9)	88.00 (5)
6	nmb	3	10/20/18	0.0000 (4)	0.123 (6)	0.02 (3)	1864.97 (12)	37.30 (12)	1940.00 (8)
7	kara.dhara	1	10/17/18	0.0000 (4)	0.082 (7)	0.01 (4)	49.19 (3)	0.98 (3)	87.00 (4)
8	sanjaykr10	1	10/17/18	0.0000 (4)	0.082 (7)	0.01 (4)	49.35 (5)	0.99 (5)	86.00 (3)
9	EdmonWales	1	10/14/18	0.0000 (4)	0.082 (7)	0.01 (4)	49.23 (4)	0.98 (4)	86.00 (3)
10	dcoldeira	1	10/13/18	0.0000 (4)	0.082 (7)	0.01 (4)	49.66 (8)	0.99 (8)	86.00 (3)
				0.0000					

Throughput phase LB

- 113 registered, but only 3 with non zero scores, 13 with zero score
- =>disappointing participation, many hypotheses why
- ➔originally scheduled to end early November, postpone to the end to 12 March 2019
- Working on a « reboot » profitting from NeurIPS visibility
- On the other hand fastrack results are astonishing
    - ATLAS code recently sped up from 250s to 10s … however this is for track pT>900 MeV ~15% of TrackML tracks



More information

- Contact : trackml.contact@gmail.com (or myself rousseau@lal.in2p3.fr)
- https://sites.google.com/site/trackmlparticle
- Twitter : @trackmllhc
- Accuracy phase @ Kaggle : https://www.kaggle.com/c/trackml-particle-identification
- Throughput phase @ Codalab : https://competitions.codalab.org/competitions/20112
  - Still running till 12th March, you can still participate !!!
  - Leaderboard prices #1 7k$ #2 5k$ #3 3k$
  - Special Jury prize : another Nvidia V100, CERN invite