Digital Twins for Nuclear and Particle Physics
NP-Twins 2025

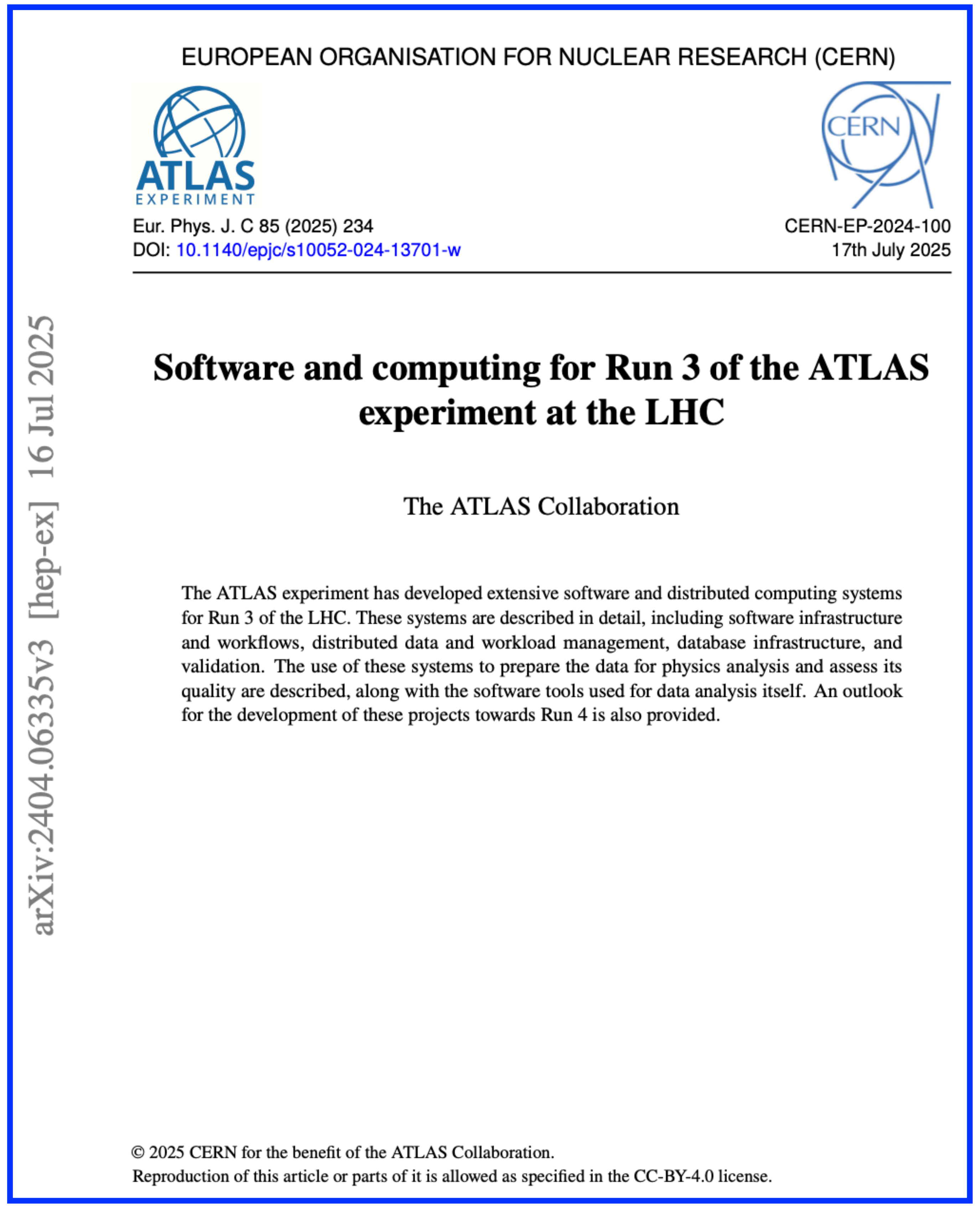
On (track) reconstruction algorithms for the HL-LHC

Jan Stark
Laboratoire des 2 Infinis - Toulouse (L2IT)

L2IT IN2P3 Institut national de physique nucléaire et de physique des particules cnrs Université de Toulouse

Reconstruction in ATLAS and the Athena software

(clickable link)



arXiv:2404.06335v3 [hep-ex] 16 Jul 2025

EUROPEAN ORGANISATION FOR NUCLEAR RESEARCH (CERN)

Eur. Phys. J. C 85 (2025) 234
DOI: 10.1140/epjc/s10052-024-13701-w

CERN-EP-2024-100
17th July 2025

Software and computing for Run 3 of the ATLAS experiment at the LHC

The ATLAS Collaboration

The ATLAS experiment has developed extensive software and distributed computing systems for Run 3 of the LHC. These systems are described in detail, including software infrastructure and workflows, distributed data and workload management, database infrastructure, and validation. The use of these systems to prepare the data for physics analysis and assess its quality are described, along with the software tools used for data analysis itself. An outlook for the development of these projects towards Run 4 is also provided.

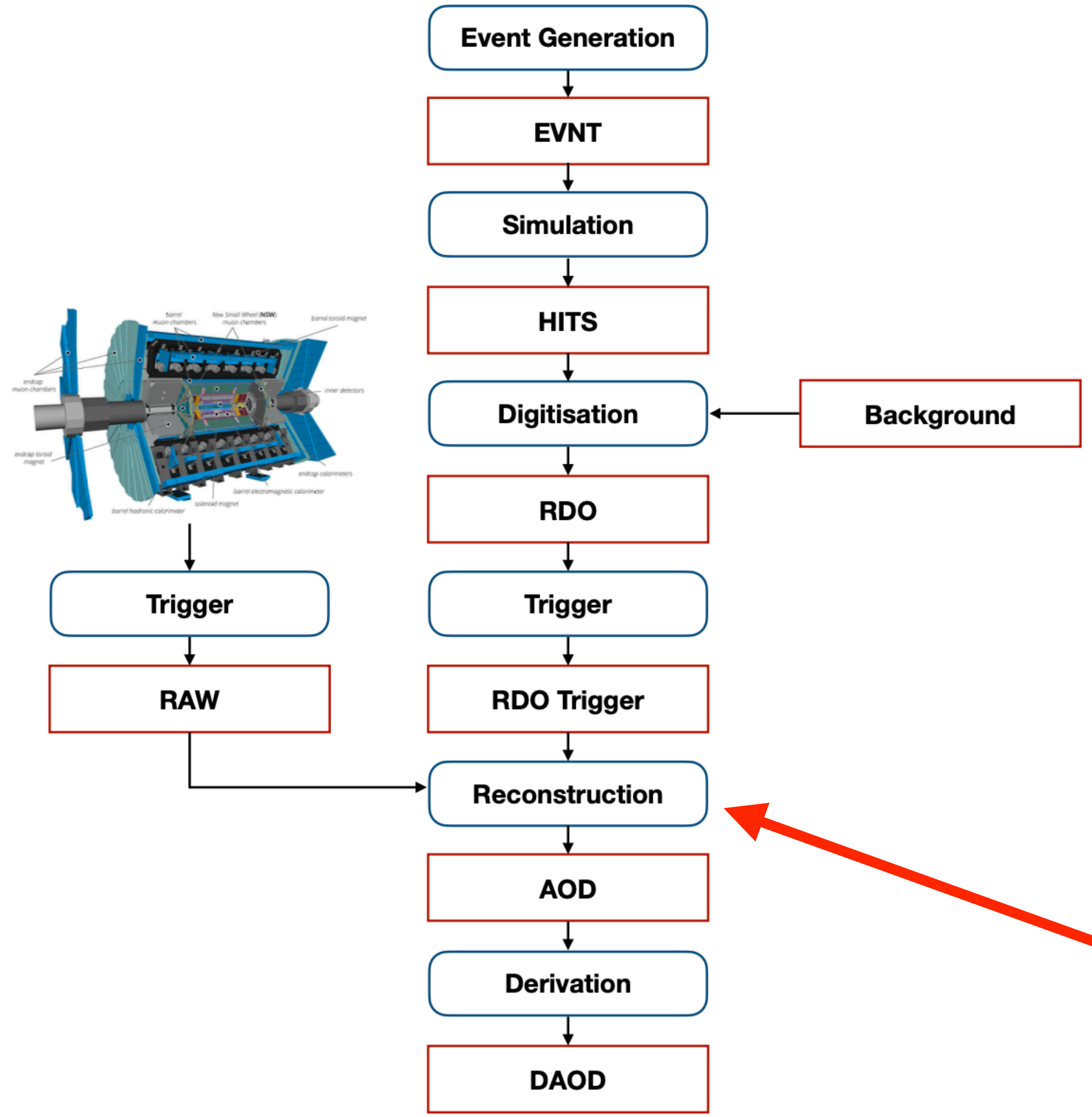


Figure 1: The standard software workflow of the ATLAS experiment. Processing steps are represented by blue ovals, with output formats represented as red boxes. The various steps and data formats are described in the text. The background entering digitisation may be additional simulated HITS files, pre-digitised RDO files, or specially processed RAW detector data.

Language	Files	Comment	Code
C++	17,273	457,373	2,608,231
Python	9,478	211,655	1,009,088
C/C++ Header	20,475	469,490	843,679
Custom Configuration	307	0	368,828
XML	954	12,800	204,169
Shell	1,243	12,283	48,782
CMake / make	2,070	11,021	35,751
Fortran	166	7,674	24,024
Web (HTML, CSS, PHP)	44	289	7,085
CUDA	28	648	5,445
Other	171	3,235	24,027
Total	52,191	1,186,288	5,178,472

5.2 million lines of code,
+ 1.2 million lines of comments

Reconstruction

Reconstruction in ATLAS — the Athena software

many of these aspects are discussed in the talk tomorrow by Francesco Di Bello (clickable link)

3.6 Machine learning and software infrastructure

Machine learning (ML) is used in a wide range of applications in the ATLAS Collaboration, including physics analysis, simulation and physics object identification in the reconstruction and trigger system. Models trained for classification and regression, as well as generative models, are core components of the software and analysis chain. This section describes the infrastructure and tools for ML used within the collaboration.

Classification models are the most commonly employed. For physics analyses, these models are used to categorise events into signal-like or background-like samples, efficiently distinguishing events of interest. Often multivariate event classifiers based on boosted decision trees (BDTs) are used [72–85]. BDTs have also been used for regression [86]. Other analyses have explored the use of neural networks (NNs) for classification, for example in top-quark physics analyses [87–94] to separate signal and background processes. Classification models are also used extensively to accurately identify physics objects, with applications in the identification of hadronically-decaying τ-leptons [95], boosted jets [96] and heavy-flavour jets [32]. More recently, recurrent neural networks (RNNs) [97] are exploited for their ability to harness sequential characteristics to take into account correlations between track impact parameters, thereby improving physics performance for jet flavour tagging [98] and τ-lepton reconstruction [99] (see Section 4.4).

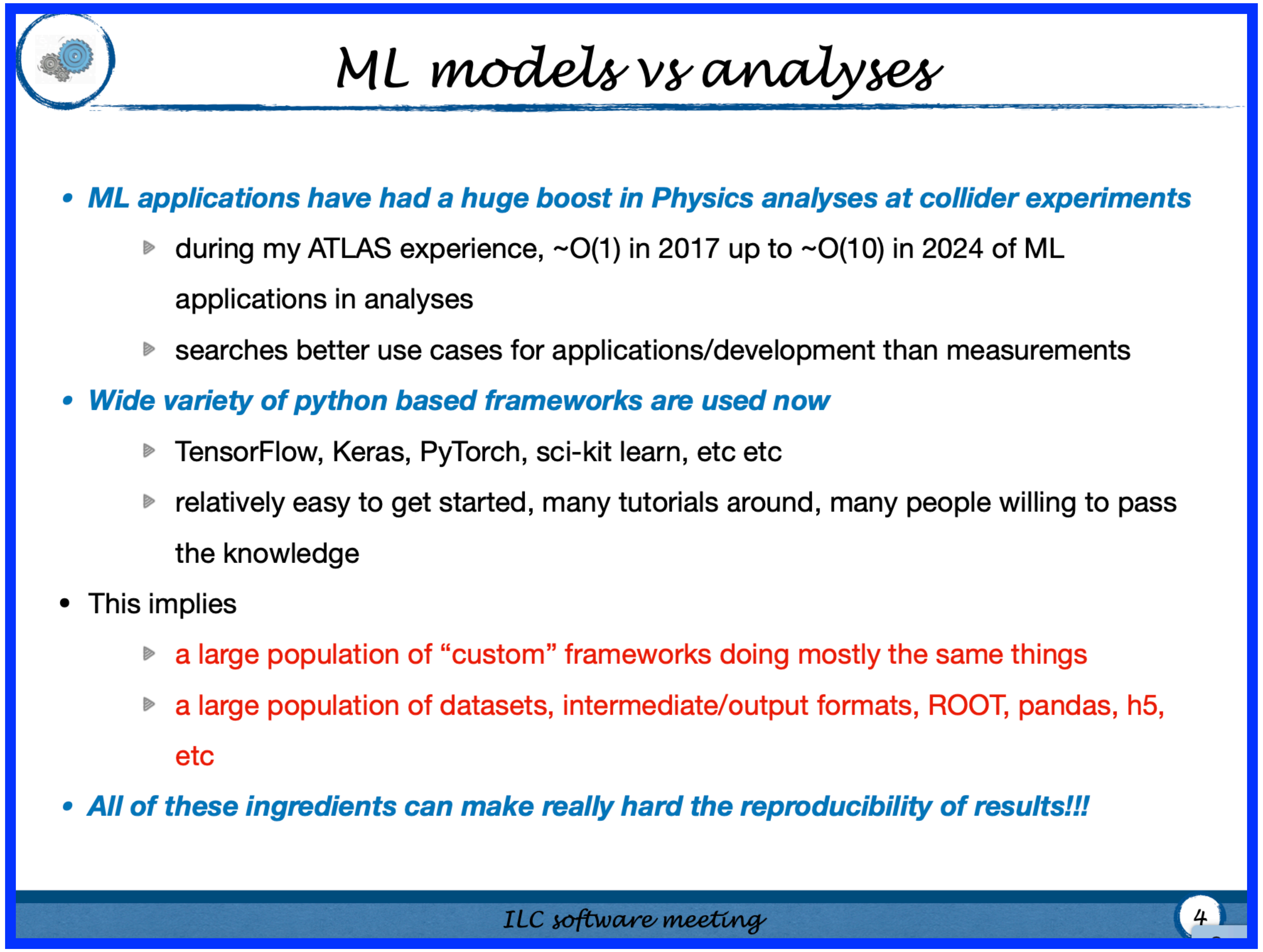
Artificial neural networks have also been used for lower-level reconstruction tasks. For example, to identify merged clusters in the pixel detector, which are created by multiple charged particles traversing the silicon, a clustering algorithm is implemented to identify and split the clusters and provide improved position estimates for the individual cluster pieces [100].

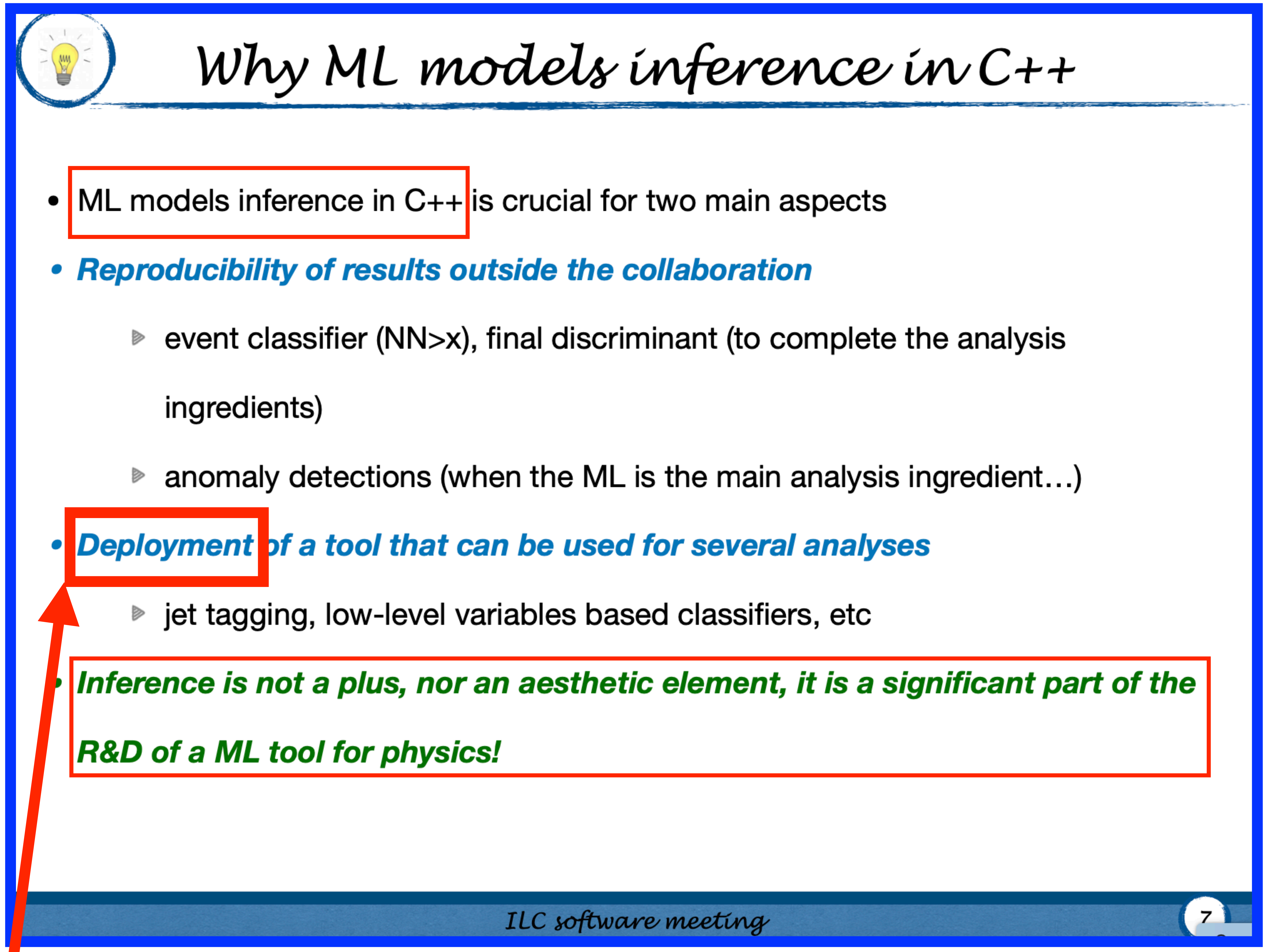
(...)

The first general purpose inference framework integrated into the ATLAS software environment was LWTNN [111], which uses the BOOST library [112] as a JSON parser and EIGEN [113] for tensor operations. Users can convert their trained NN models into JSON format, defining the weights and layer operations. This is then handled by a lightweight class for running inference on single events. For the inference of BDTs, a custom wrapper was written that converts models trained with TMVA, LIGHTGBM or XGBOOST into a common ROOT `TTree` format for inference at runtime. However, as BDTs are mostly used in physics analyses, other libraries are compiled in standalone ntuple processing frameworks to provide inference. Due to the rise of more complex models, which use layers and operations that are not supported by LWTNN, the ONNX RUNTIME (ORT) library [114] was chosen as a general inference library. Most modern ML libraries can save models in an ORT format, or are supported with conversion tools. These include, but are not limited to, PYTORCH, TENSORFLOW, LIGHTGBM and XGBOOST. Using only a single inference library to support many models trained in different frameworks significantly reduces the maintenance load. ORT is observed to provide faster inference than previous implementations and support for batched

ML model inference in C++ — and in Athena

Slides taken from:
Antonio Giannini, « ONNX-based inference in ATLAS », ILC software meeting, 15 May 2024 (clickable link)

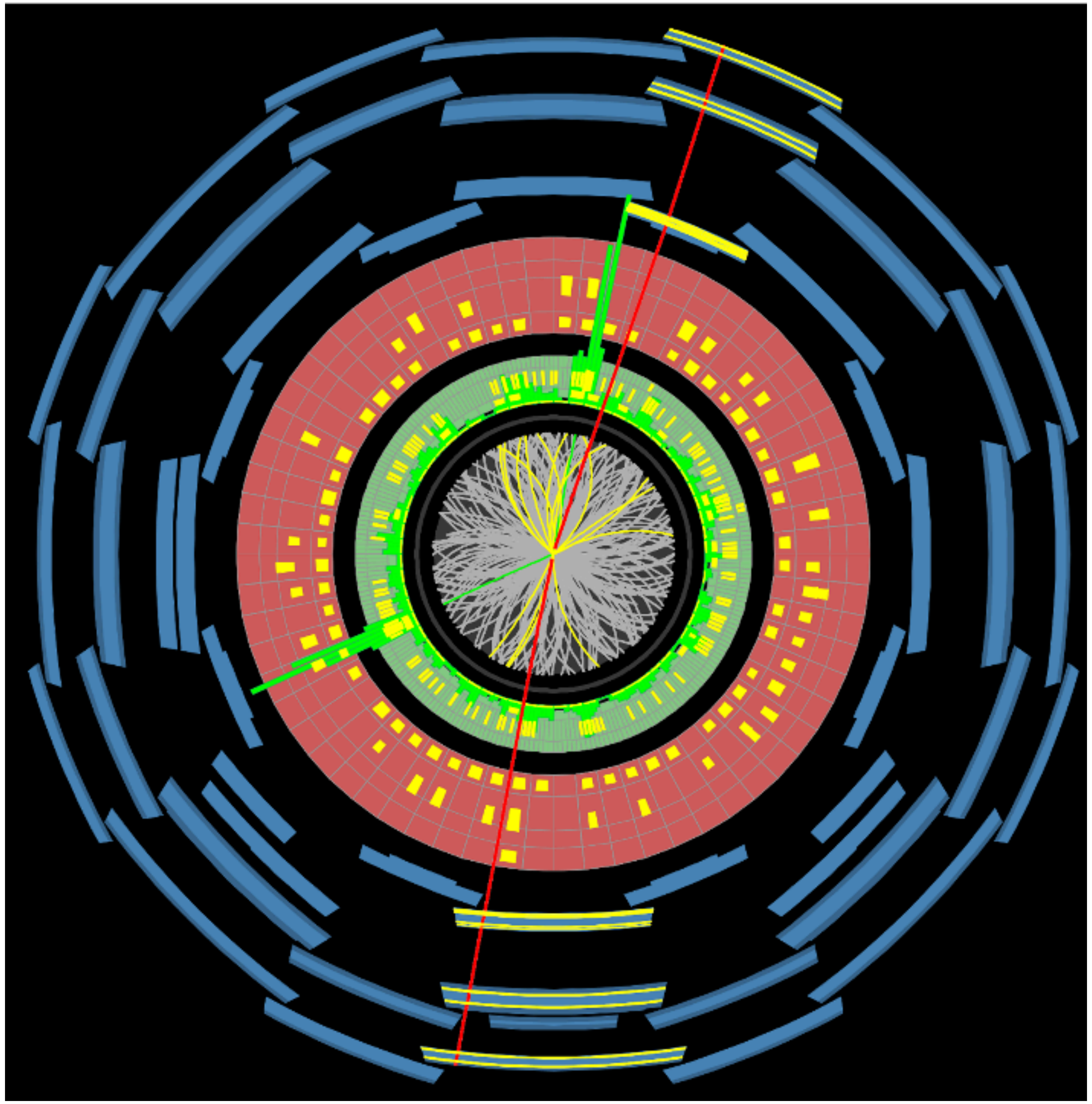




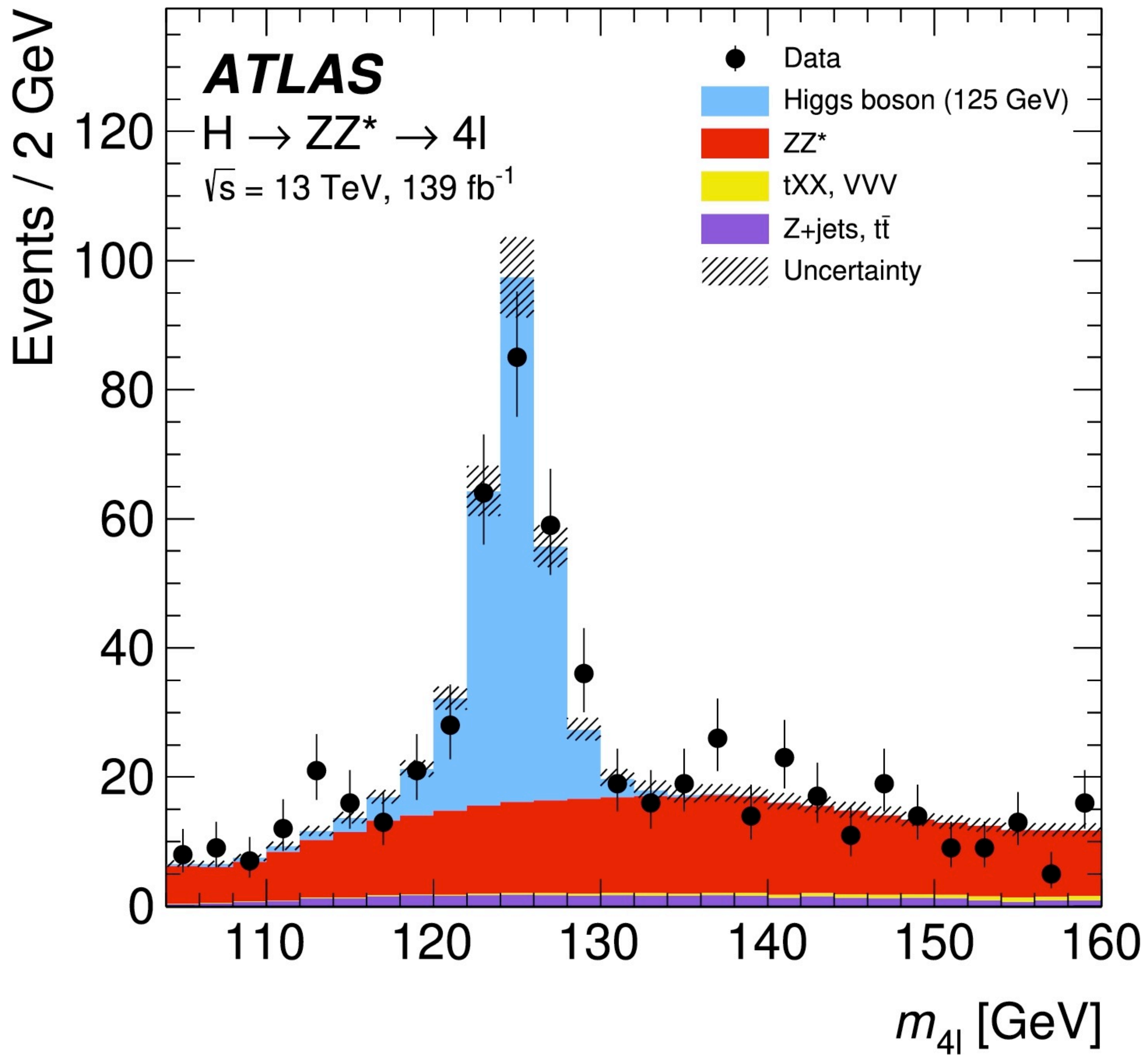
including in Athena, for running on CPU farms

Higgs boson

One ***event*** seen in the ATLAS detector

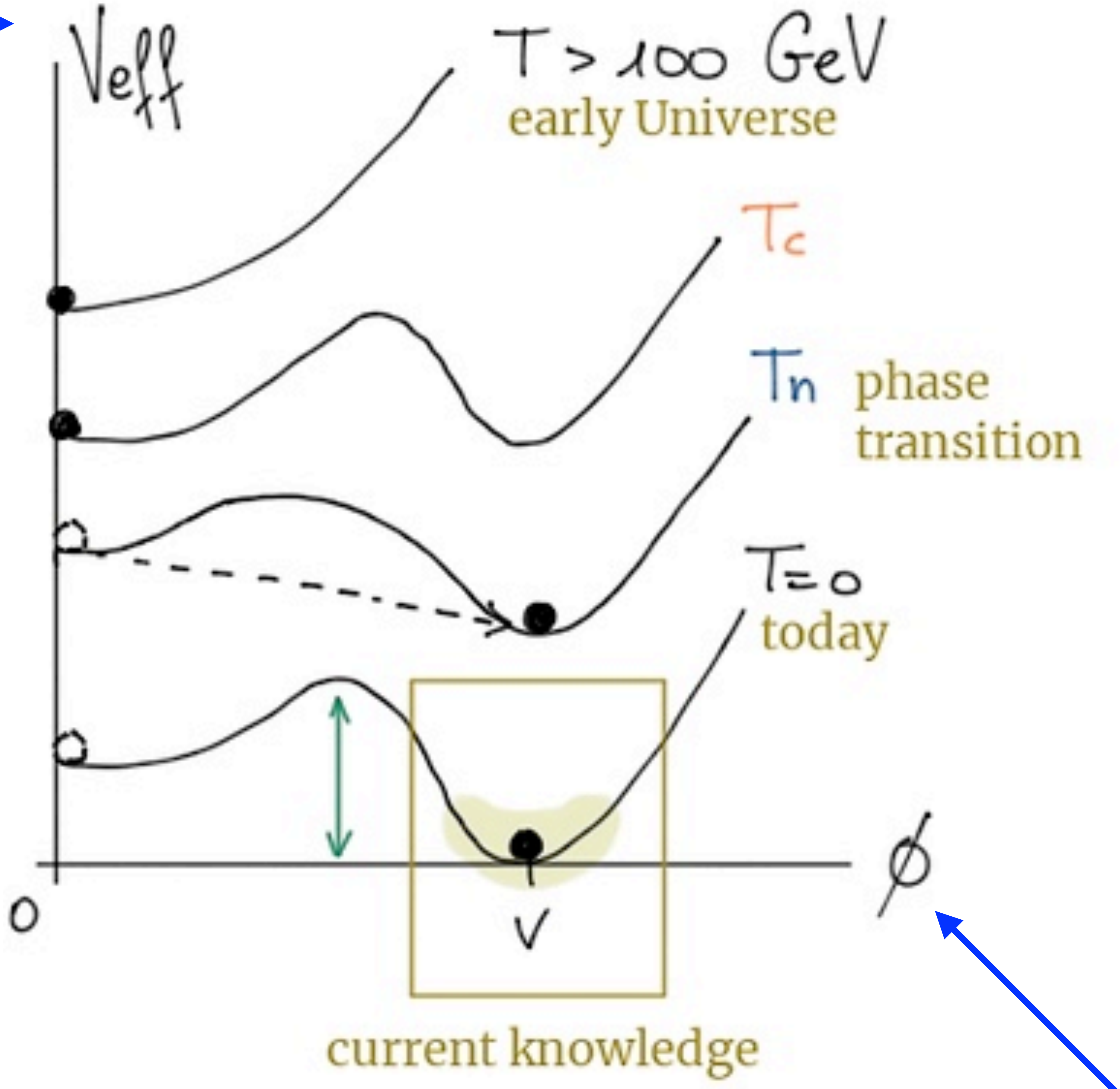


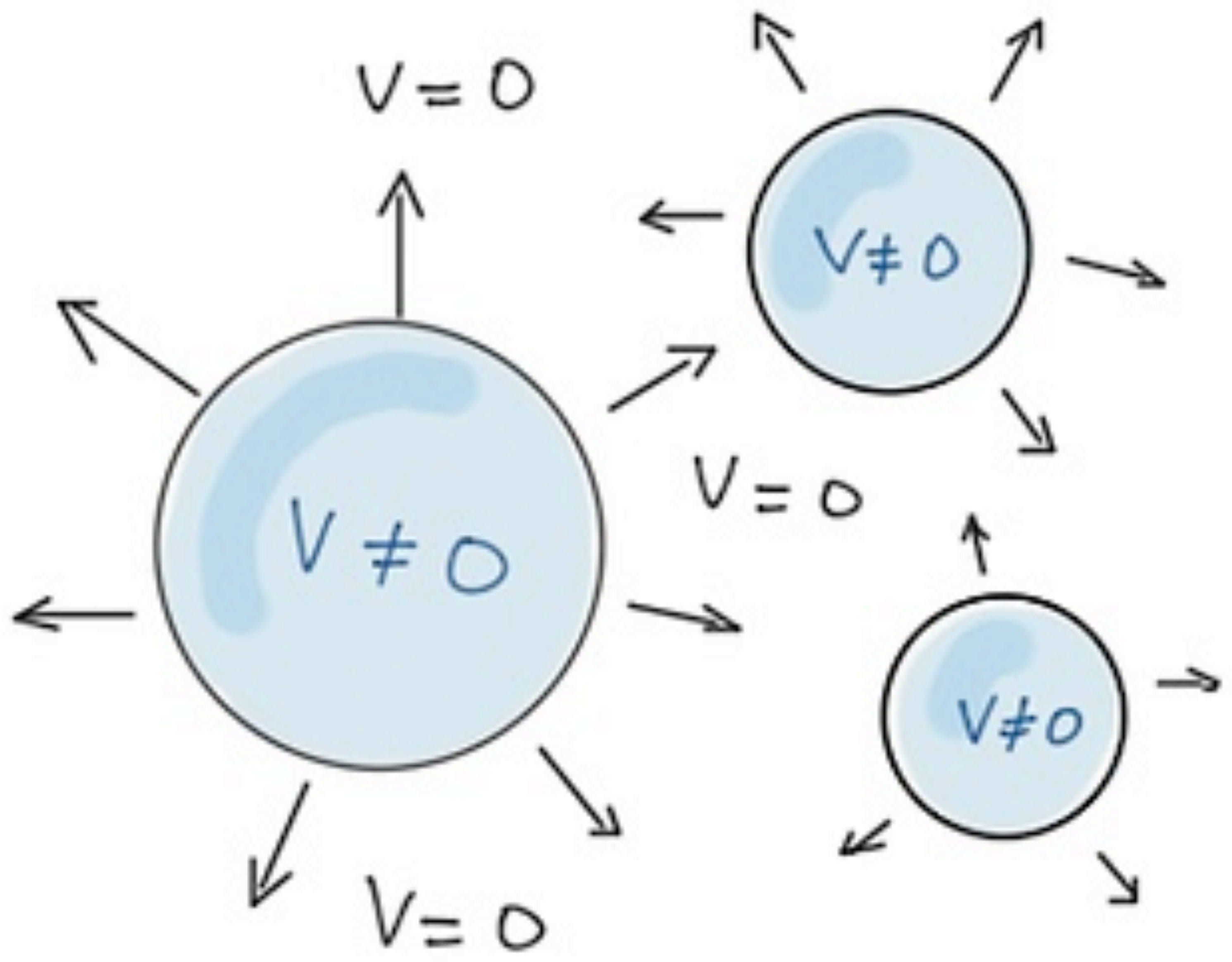
Counting *events in different intervals* of ***reconstructed*** H -> 4l mass (*m_{4l}*)



Higgs boson: its role in the early Universe

quantum field ϕ **interacts** with potential V_{eff}





Need much more data to study the shape of V_{eff}

(because information is given by rare events)

Higgs boson
= excitation in quantum field ϕ

Figure:
Kateryna Radchenko
Cluster of Excellence "Quantum Universe"
Universität Hamburg

Taking data at the LHC is like drinking water out of a firehose

Rheinfall (Rhine Falls):
750 m^3/second

Data from detector

40 million events / second

60 TB / second

(big) **firehose**:
19 liters/second

Data recorded to **disk**

1000 events /second

1.5 GB / second

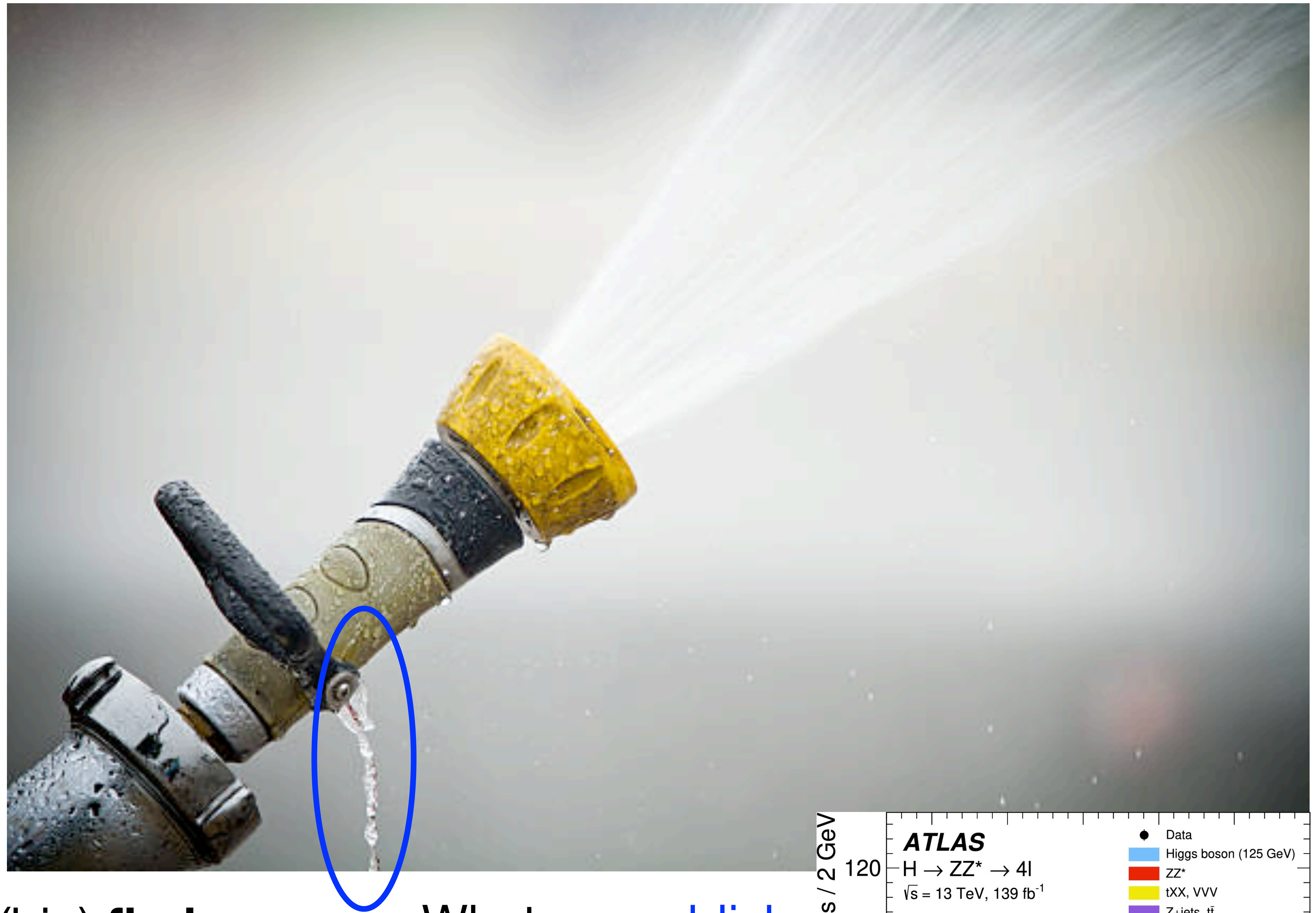
Taking data at the LHC is like drinking water out of a firehose

Rheinfall (Rhine Falls):
750 m³/second

Data from detector

40 million events / second

60 TB / second



(big) **firehose**:
19 liters/second

Data recorded to **disk**

1000 events /second

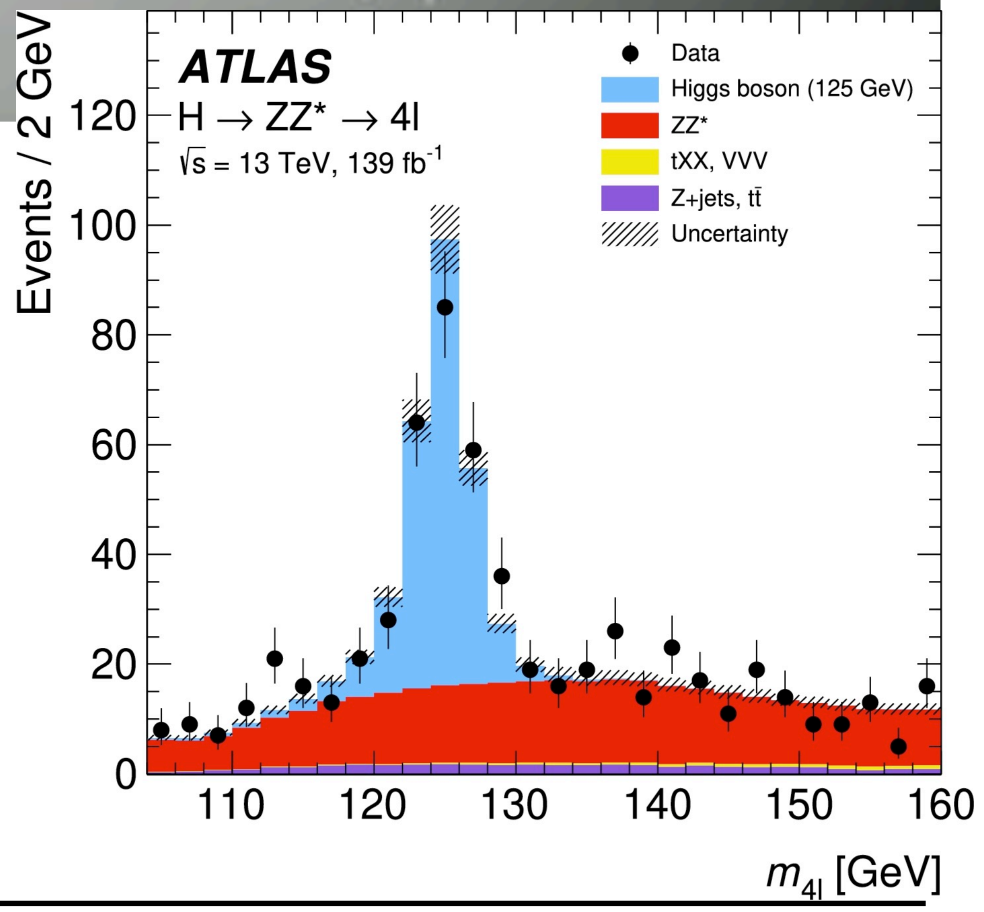
1.5 GB / second

What we publish:
a few **drops**

~1000 events

from

4 year dataset



Taking data at the LHC is like drinking water out of a firehose

AI is frequently used at this level.
We are working on deploying
it at this level and earlier.

Rheinfall (Rhine Falls):
750 m³/second

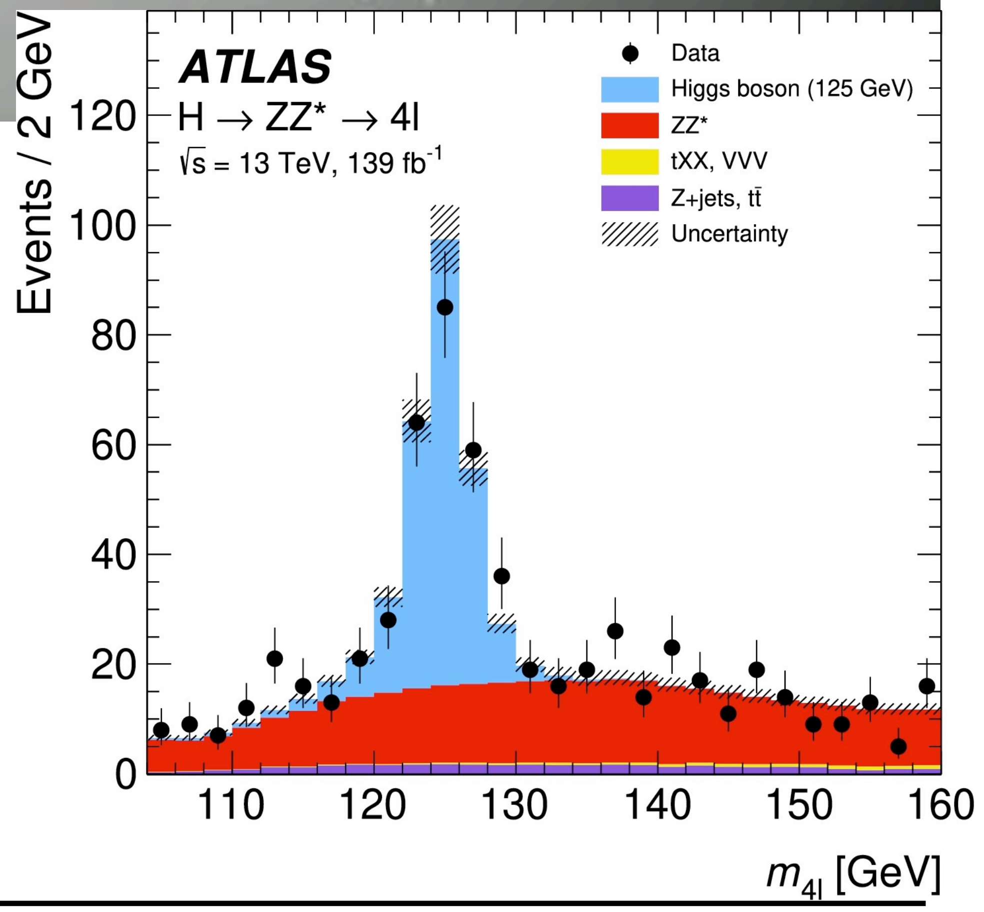
Data from detector
40 million events / second
60 TB / second

(big) **firehose**:
19 liters/second

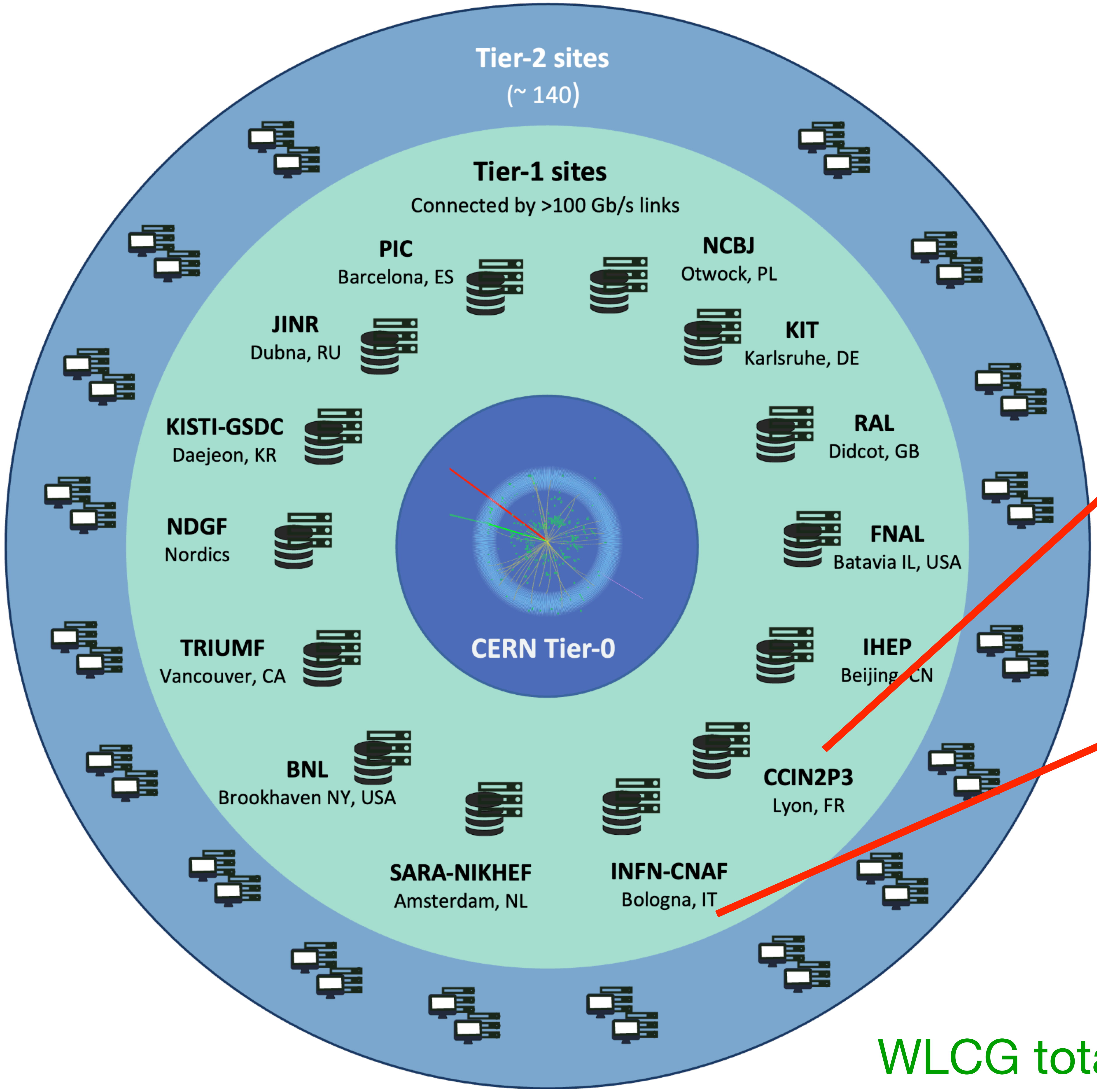
Data recorded to **disk**
1000 events /second
1.5 GB / second

What we publish:
a few **drops**

~1000 events
from
4 year dataset



Worldwide LHC computing grid (WLCG)



Centre de Calcul
de l'IN2P3
(Lyon, FR)

WLCG total: 1.4 million CPU cores and 1.5 exabytes of storage

“10 years to prepare ourselves” for HL-LHC (statement from 2017)

- *Community white paper* (2017)
  - Algorithms, infrastructure, data access…

Great overview of ongoing changes in computing industry, current practices in HEP and required R&D activities in key domains:
- Physics generators
- Detector simulation
- Software trigger & event reconstruction
- Data analysis
- Machine learning
- Data management, …
- Facilities, distributed computing
- ….

arXiv.org > physics > arXiv:1712.06982

Search... All fields Search
Help | Advanced Search

Physics > Computational Physics

A Roadmap for HEP Software and Computing R&D for the 2020s

Johannes Albrecht, Antonio Augusto Alves Jr, Guilherme Amadio, Giuseppe Andronico, Nguyen Anh-Ky, Laurent Aphecetche, John Apostolakis, Makoto Asai, Luca Atzori, Marian Babik, Giuseppe Bagliesi, Marilena Bandieramonte, Sunanda Banerjee, Martin Barisits, Lothar A.T. Bauerdick, Stefano Belforte, Douglas Benjamin, Catrin Bernius, Wahid Bhimji, Riccardo Maria Bianchi, Ian Bird, Catherine Biscarat, Jakob Blomer, Kenneth Bloom, Tommaso Boccali, Brian Bockelman, Tomasz Bold, Daniele Bonacorsi, Antonio Boveia, Concezio Bozzi, Marko Bracko, David Britton, Andy Buckley, Predrag Buncic, Paolo Calafiura, Simone Campana, Philippe Canal, Luca Canali, Gianpaolo Carlino, Nuno Castro, Marco Cattaneo, Gianluca Cerminara, Javier Cervantes Villanueva, Philip Chang, John Chapman, Gang Chen, Taylor Childers, Peter Clarke, Marco Clemencic, Eric Cogneras, Jeremy Coles, Ian Collier, David Colling, Gloria Corti, Gabriele Cosmo, Davide Costanzo, Ben Couturier, Kyle Cranmer, Jack Cranshaw, Leonardo Cristella, David Crooks, Sabine Crépé-Renaudin, Robert Currie, Sünje Dallmeier-Tiessen, Kaushik De, Michel De Cian, Albert De Roeck, Antonio Delgado Peris, Frédéric Derue, Alessandro Di Girolamo, Salvatore Di Guida, Gancho Dimitrov, Caterina Doglioni, Andrea Dotti, Dirk Duellmann, Laurent Duflot, Dave Dykstra, Katarzyna Dziedziniewicz-Wojcik, Agnieszka Dziurda, Ulrik Egede, Peter Elmer, Johannes Elmsheuser, V. Daniel Elvira, Giulio Eulisse, Steven Farrell, Torben Ferber, Andrej Filipcic, Ian Fisk, Conor Fitzpatrick, José Flix, Andrea Formica, Alessandra Forti, Giovanni Franzoni, James Frost, Stu Fuess, Frank Gaede, Gerardo Ganis, Robert Gardner, Vincent Garonne, Andreas Gellrich et al. (210 additional authors not shown)

(Submitted on 18 Dec 2017 (v1), last revised 19 Dec 2018 (this version, v5))

Particle physics has an ambitious and broad experimental programme for the coming decades. This programme requires large investments in detector hardware, either to build new facilities and experiments, or to upgrade existing ones. Similarly, it requires commensurate investment in the R&D of software to acquire, manage, process, and analyse the shear amounts of data to be recorded. In planning for the HL-LHC in particular, it is critical that all of the collaborating stakeholders agree on the software goals and priorities, and that the efforts complement each other. In this spirit, this white paper describes the R&D activities required to prepare for this software upgrade.

Download:
- PDF
- Other formats

Current browse context:
physics.comp-ph
< prev | next >
new | recent | 1712

Change to browse by:
hep-ex
physics

References & Citations
- INSPIRE HEP (refers to | cited by)
- NASA ADS

Export citation
Google Scholar

Bookmark

(link)

2020 update of the
European strategy for particle physics

D. Large-scale data-intensive software and computing infrastructures are an essential ingredient to particle physics research programmes. The community faces major challenges in this area, notably with a view to the HL-LHC. As a result, the software and computing models used in particle physics research must evolve to meet the future needs of the field. ***The community must vigorously pursue common, coordinated R&D efforts in collaboration with other fields of science and industry, to develop software and computing infrastructures that exploit recent advances in information technology and data science. Further development of internal policies on open data and data preservation should be encouraged, and an adequate level of resources invested in their implementation.***

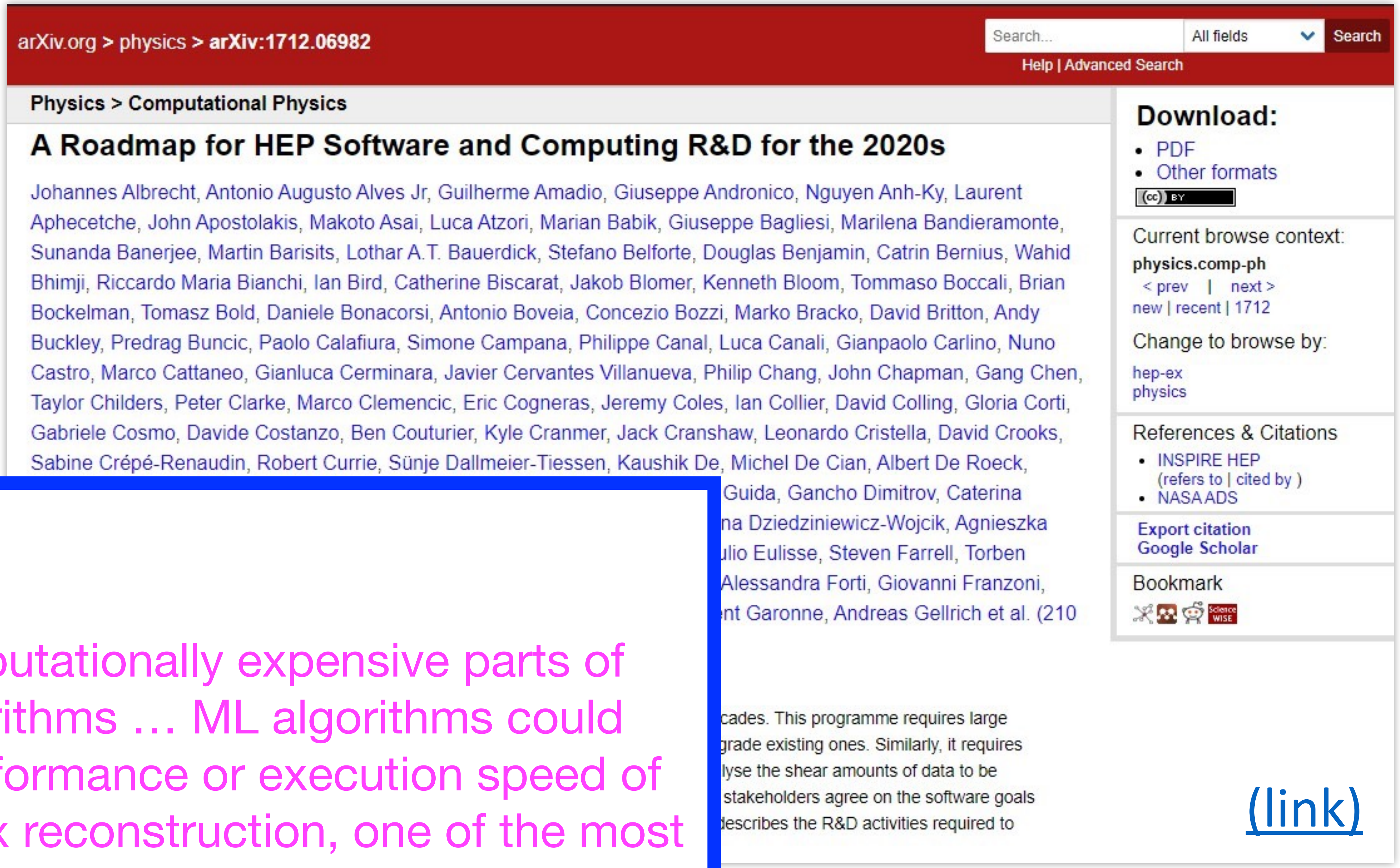
(link)

“10 years to prepare ourselves” for HL-LHC (statement from 2017)

- *Community white paper* (2017)
  - Algorithms, infrastructure, data access…

Great overview of ongoing changes in computing industry, current practices in HEP and required R&D activities in key domains:

- Physics generators
- Detector simulation
- Software trigger & event reconstruction
- Data analysis
- Machine learning
- Data management, …
- Facilities, distributed computing
- ….


arXiv.org > physics > arXiv:1712.06982

Physics > Computational Physics

A Roadmap for HEP Software and Computing R&D for the 2020s

Johannes Albrecht, Antonio Augusto Alves Jr, Guilherme Amadio, Giuseppe Andronico, Nguyen Anh-Ky, Laurent Aphecetche, John Apostolakis, Makoto Asai, Luca Atzori, Marian Babik, Giuseppe Bagliesi, Marilena Bandieramonte, Sunanda Banerjee, Martin Barisits, Lothar A.T. Bauerdick, Stefano Belforte, Douglas Benjamin, Catrin Bernius, Wahid Bhimji, Riccardo Maria Bianchi, Ian Bird, Catherine Biscarat, Jakob Blomer, Kenneth Bloom, Tommaso Boccali, Brian Bockelman, Tomasz Bold, Daniele Bonacorsi, Antonio Boveia, Concezio Bozzi, Marko Bracko, David Britton, Andy Buckley, Predrag Buncic, Paolo Calafiura, Simone Campana, Philippe Canal, Luca Canali, Gianpaolo Carlino, Nuno Castro, Marco Cattaneo, Gianluca Cerminara, Javier Cervantes Villanueva, Philip Chang, John Chapman, Gang Chen, Taylor Childers, Peter Clarke, Marco Clemencic, Eric Cogneras, Jeremy Coles, Ian Collier, David Colling, Gloria Corti, Gabriele Cosmo, Davide Costanzo, Ben Couturier, Kyle Cranmer, Jack Cranshaw, Leonardo Cristella, David Crooks, Sabine Crépé-Renaudin, Robert Currie, Sünje Dallmeier-Tiessen, Kaushik De, Michel De Cian, Albert De Roeck, … Guida, Gancho Dimitrov, Caterina … na Dziedziniewicz-Wojcik, Agnieszka … lio Eulisse, Steven Farrell, Torben … Alessandra Forti, Giovanni Franzoni, … nt Garonne, Andreas Gellrich et al. (210

… cades. This programme requires large … grade existing ones. Similarly, it requires … lyse the shear amounts of data to be … stakeholders agree on the software goals … describes the R&D activities required to

Download:
- PDF
- Other formats

Current browse context: physics.comp-ph
< prev | next >
new | recent | 1712

Change to browse by:
hep-ex
physics

References & Citations
- INSPIRE HEP (refers to | cited by)
- NASA ADS

Export citation
Google Scholar

Bookmark

(link)

This talk

”Replace the most computationally expensive parts of pattern recognition algorithms … ML algorithms could improve the physics performance or execution speed of charged track and vertex reconstruction, one of the most CPU intensive elements of our current software.”

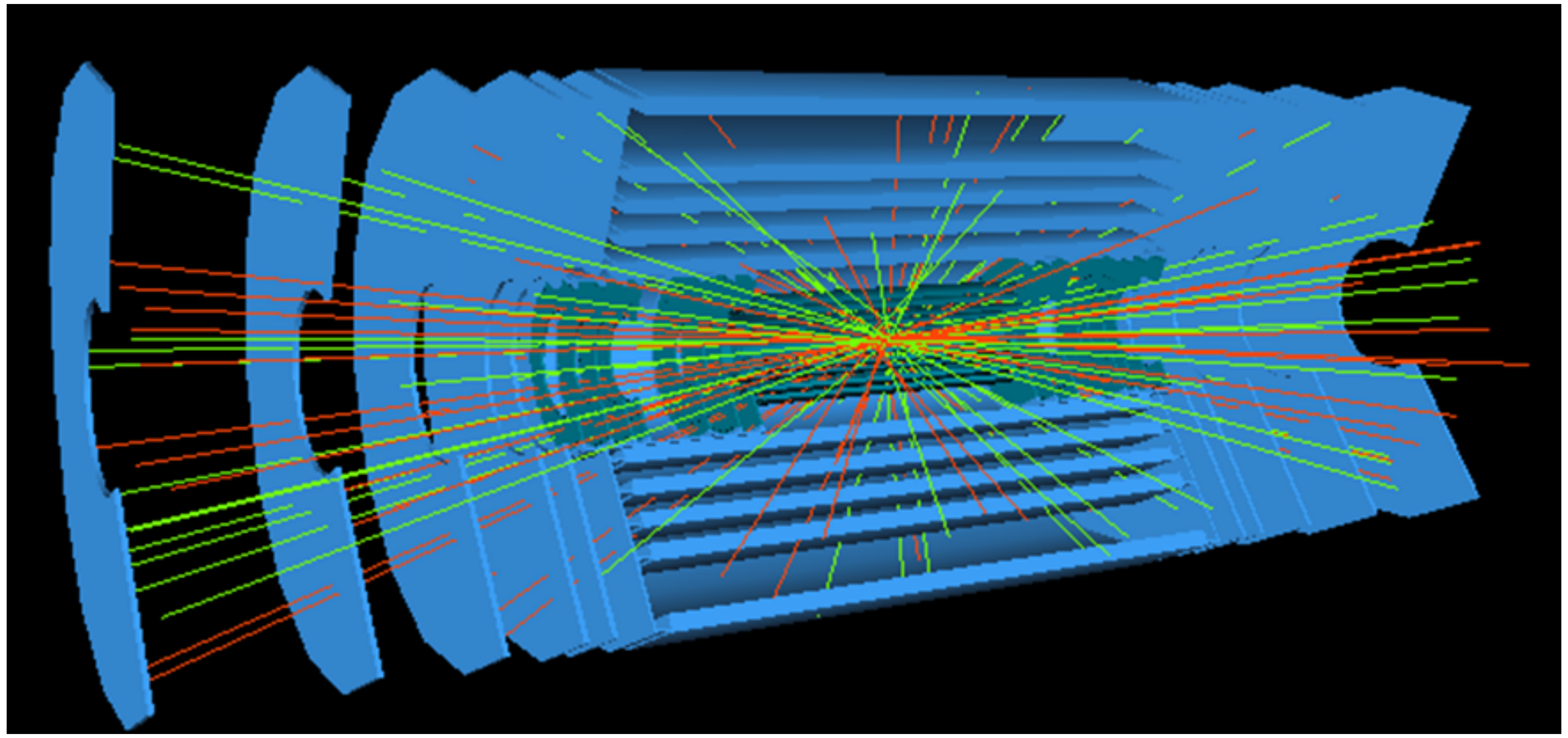
2020 update of the European strategy for particle physics

D. Large-scale data-intensive software and computing infrastructures are an essential ingredient to particle physics research programmes. The community faces major challenges in this area, notably with a view to the HL-LHC. As a result, the software and computing models used in particle physics research must evolve to meet the future needs of the field. ***The community must vigorously pursue common, coordinated R&D efforts in collaboration with other fields of science and industry, to develop software and computing infrastructures that exploit recent advances in information technology and data science. Further development of internal policies on open data and data preservation should be encouraged, and an adequate level of resources invested in their implementation.*** (link)

LHC → HL-LHC

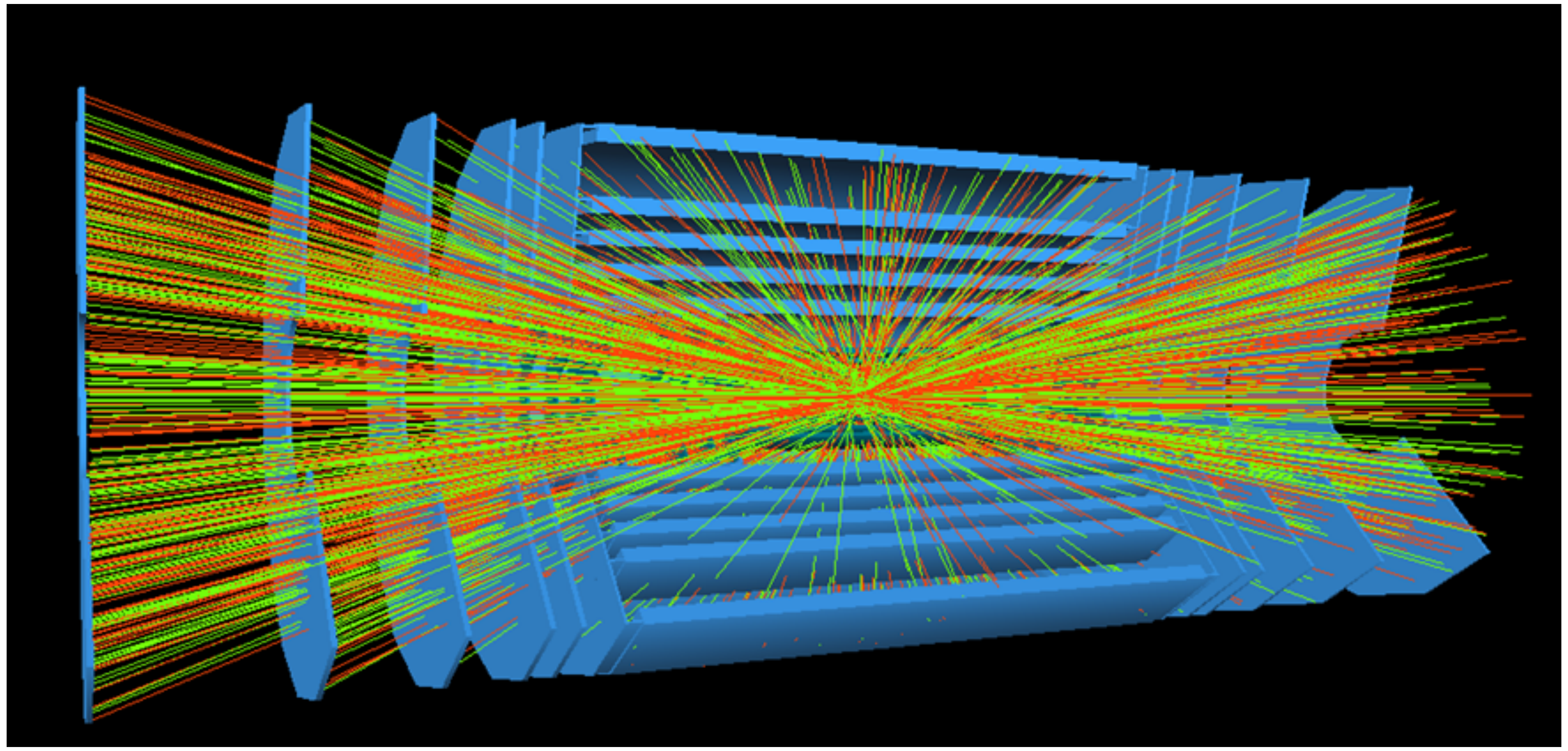
Beams in the LHC : bunches of protons

40 million bunch crossings per second

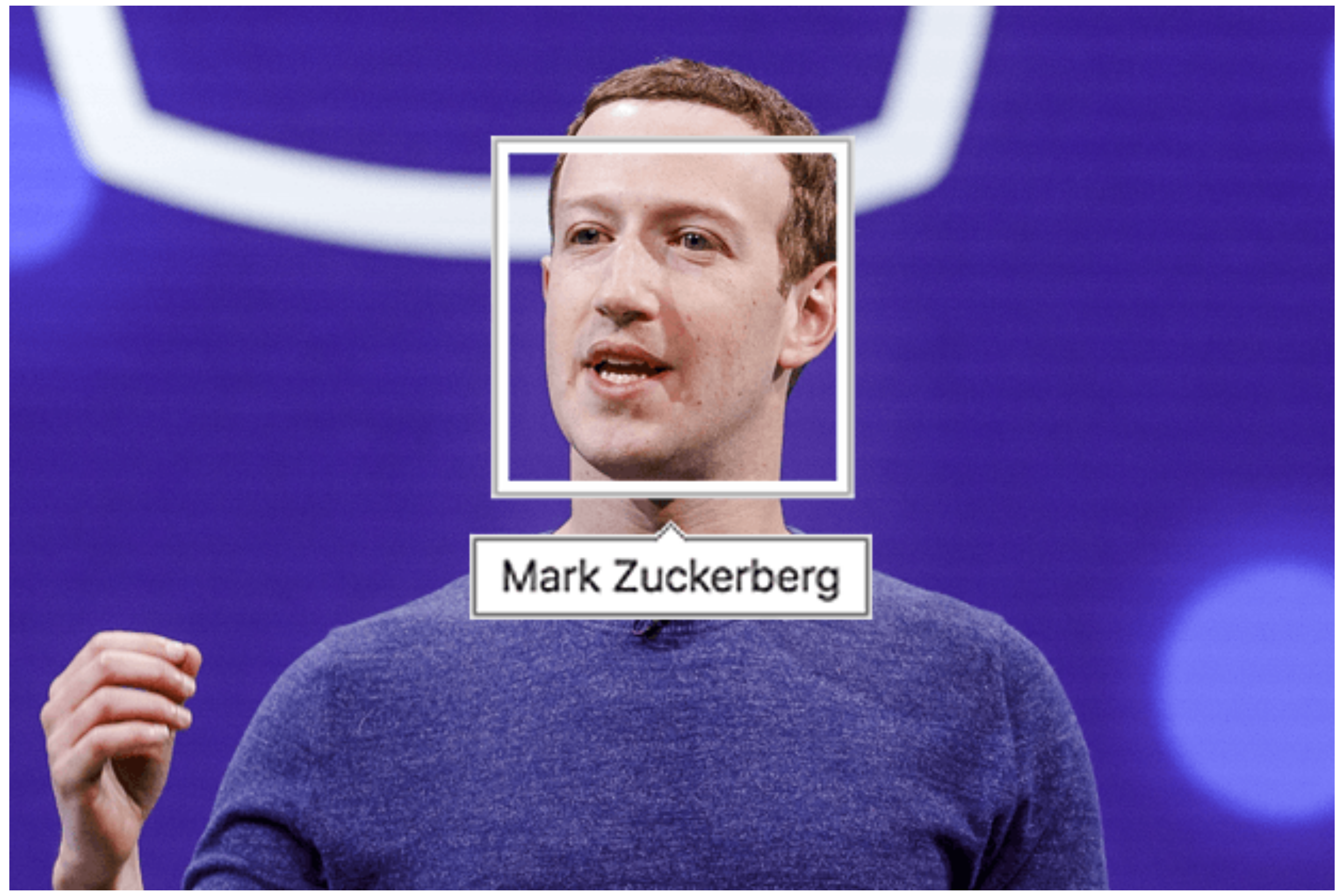


Cannot have more bunches

Put way more protons into each bunch



Machine Learning for track pattern recognition ?

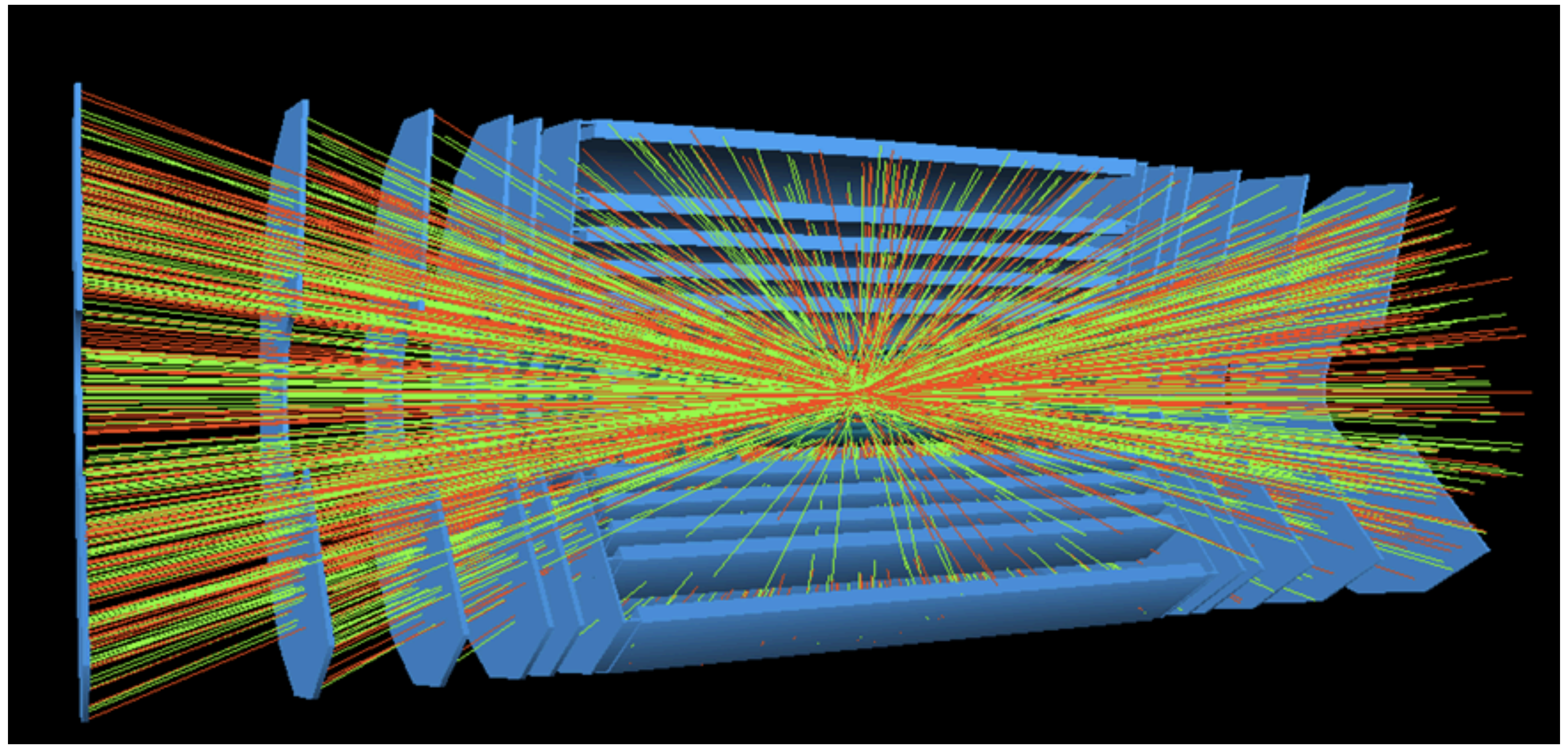


622 * 415 pixels

a large fraction carries information about the person

Can’t use the same tools

How to present tracking data to a neural network ?



ATLAS tracker for HL-LHC:
5 * 10^9 readout channels
~3 * 10^5 3D space-points per event

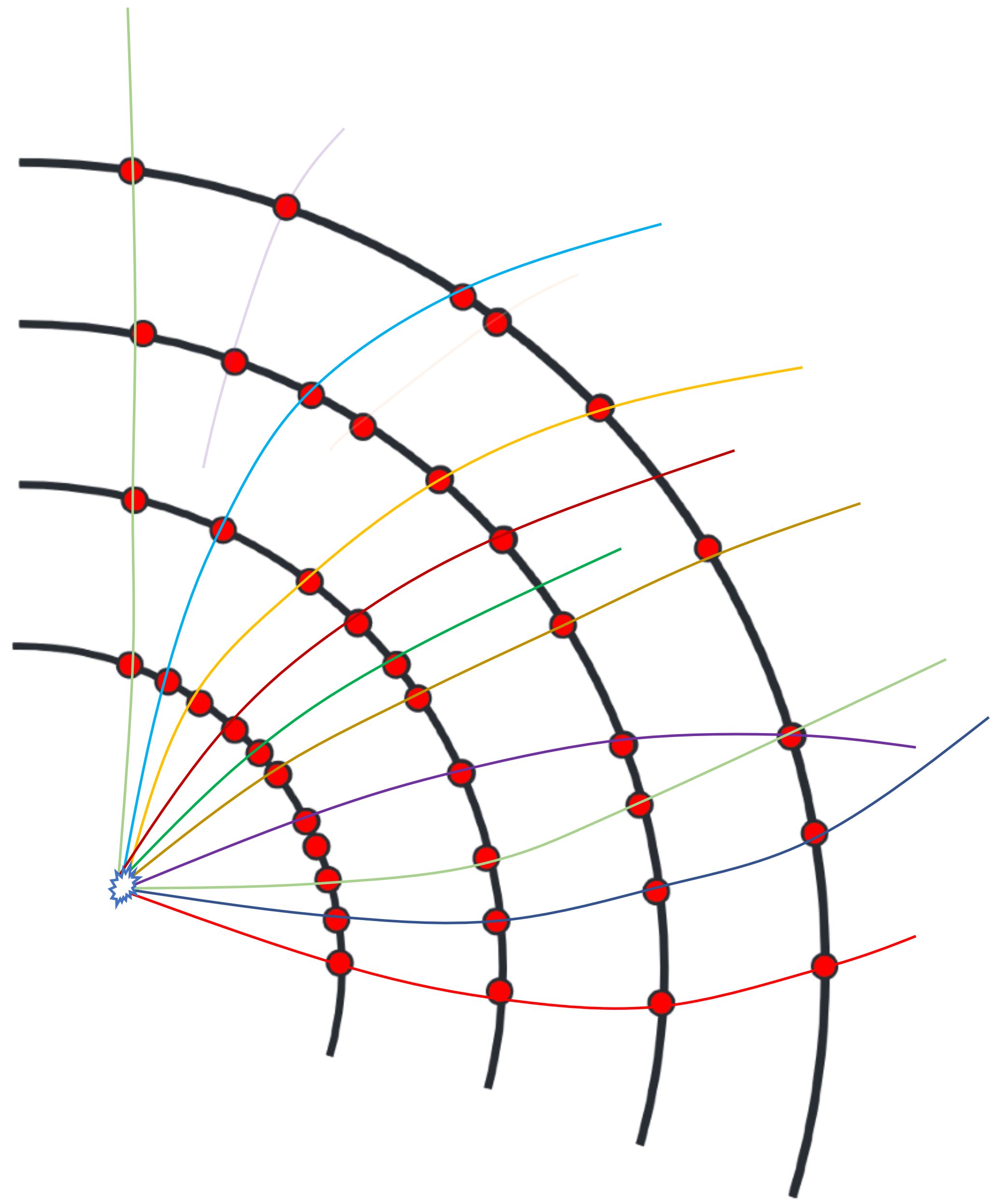
=> data are *sparse*

Representing tracking data using graphs

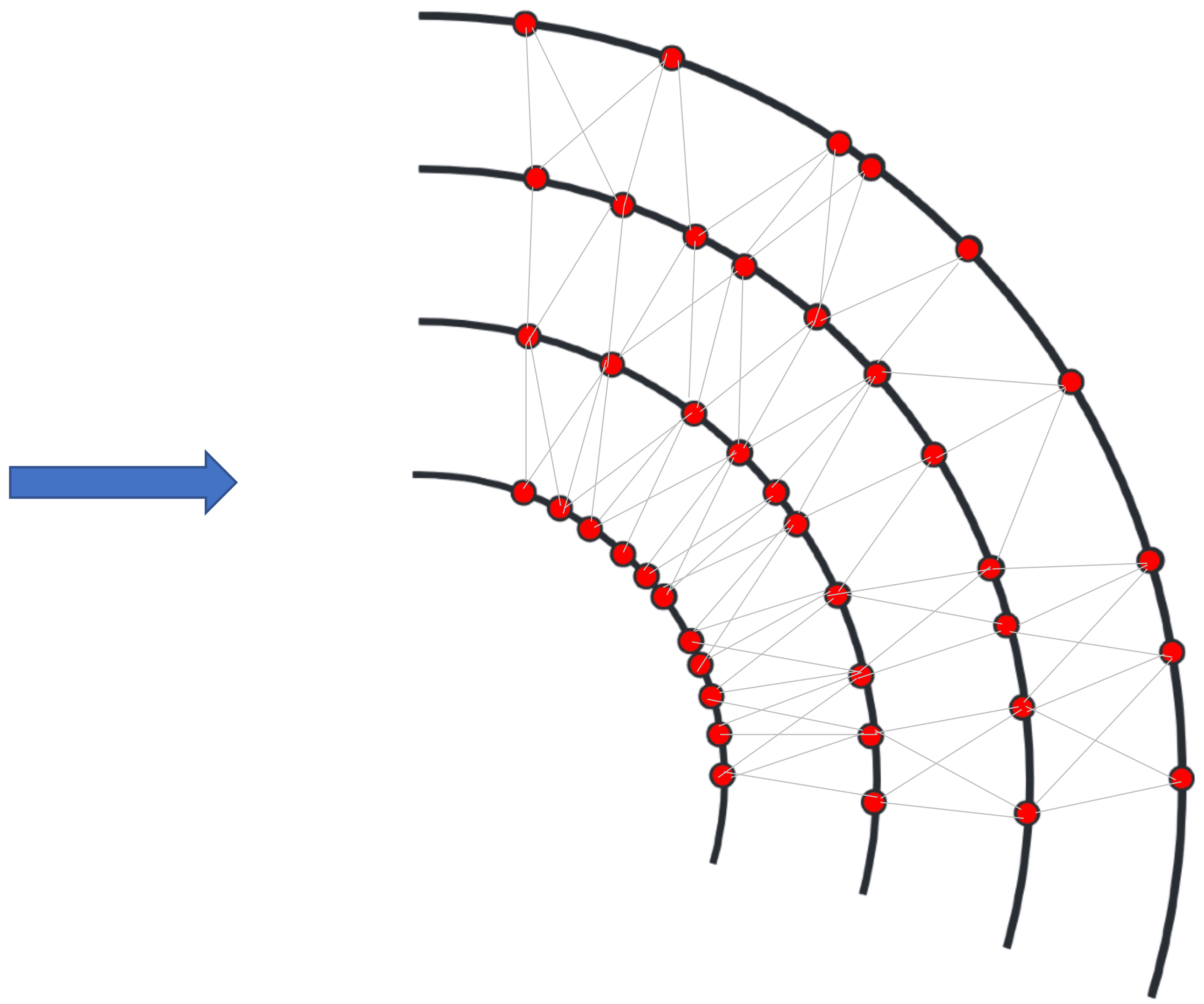
F. Siklér, "Combination of various data analysis techniques for efficient track reconstruction in very high multiplicity events", *Connecting the Dots* conference 2017 (link)

S. Farrell *et al.*, "Novel deep learning methods for track reconstruction", proceedings of *Connecting the Dots* conference 2018 (link)

Charged particles leave hits in the detector



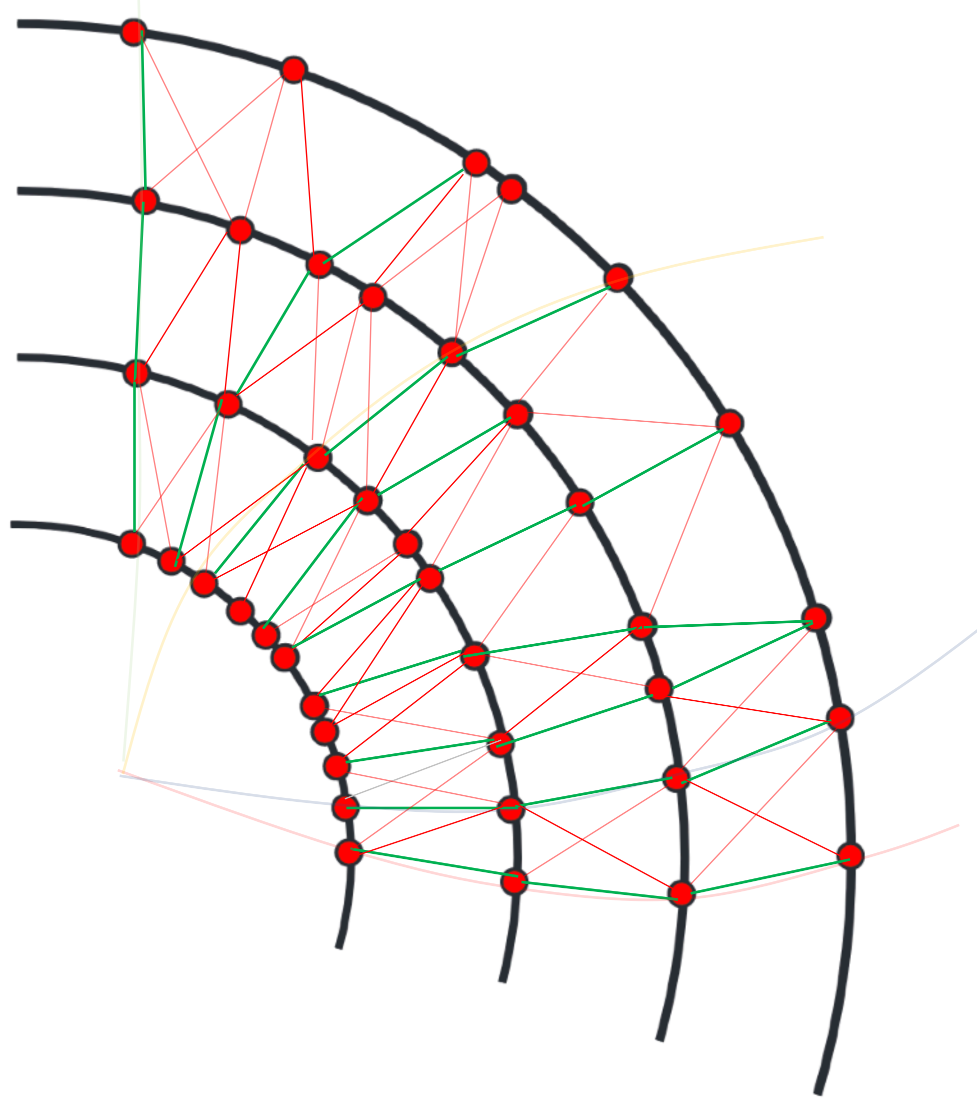
Represent the data using a graph



One node of the graph = one hit in the detector

Connect two nodes using an edge
if "it seems possible" that the two hits
are two (consecutive) hits on a track

Goal:
classify the edges of the graph

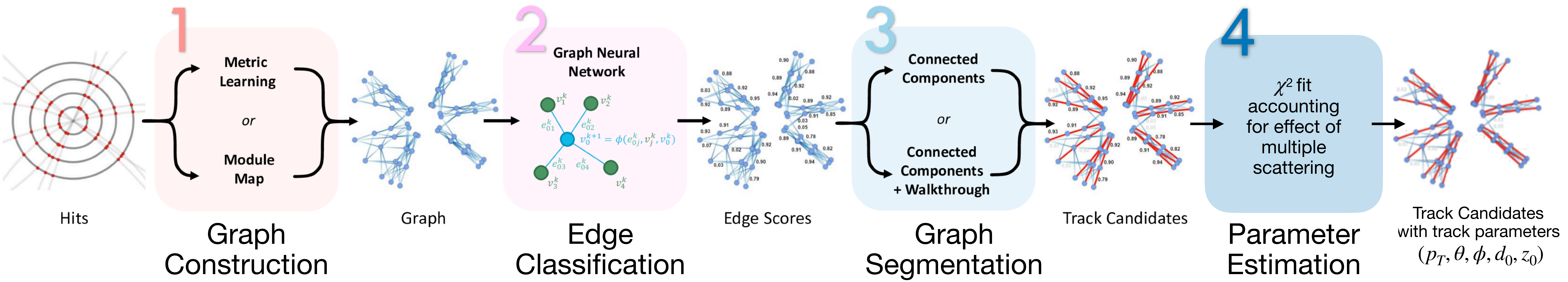


High classification score
=> **high probability** that the edge is part of a track

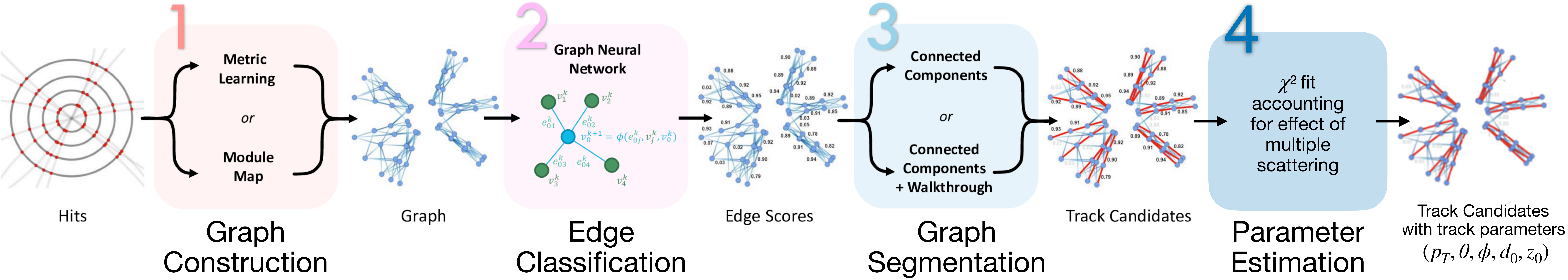
Low classification score
=> **low probability** that the edge is part of a track

More general review article:
"GNNs at the LHC" (link)

Track pattern recognition using GNNs



Track pattern recognition using GNNs



F. Siklér, talk at *Connecting the Dots* 2017

C. Biscarat, S. Caillou, C. Rougier, J. Stark and J. Zahreddine, EPJ Web of Conferences 251, 03047 (2021)

X. Ju *et al.*, Eur. Phys. J. C 81, 876 (2021)

ML versus classical algorithms

Automatic translation of text

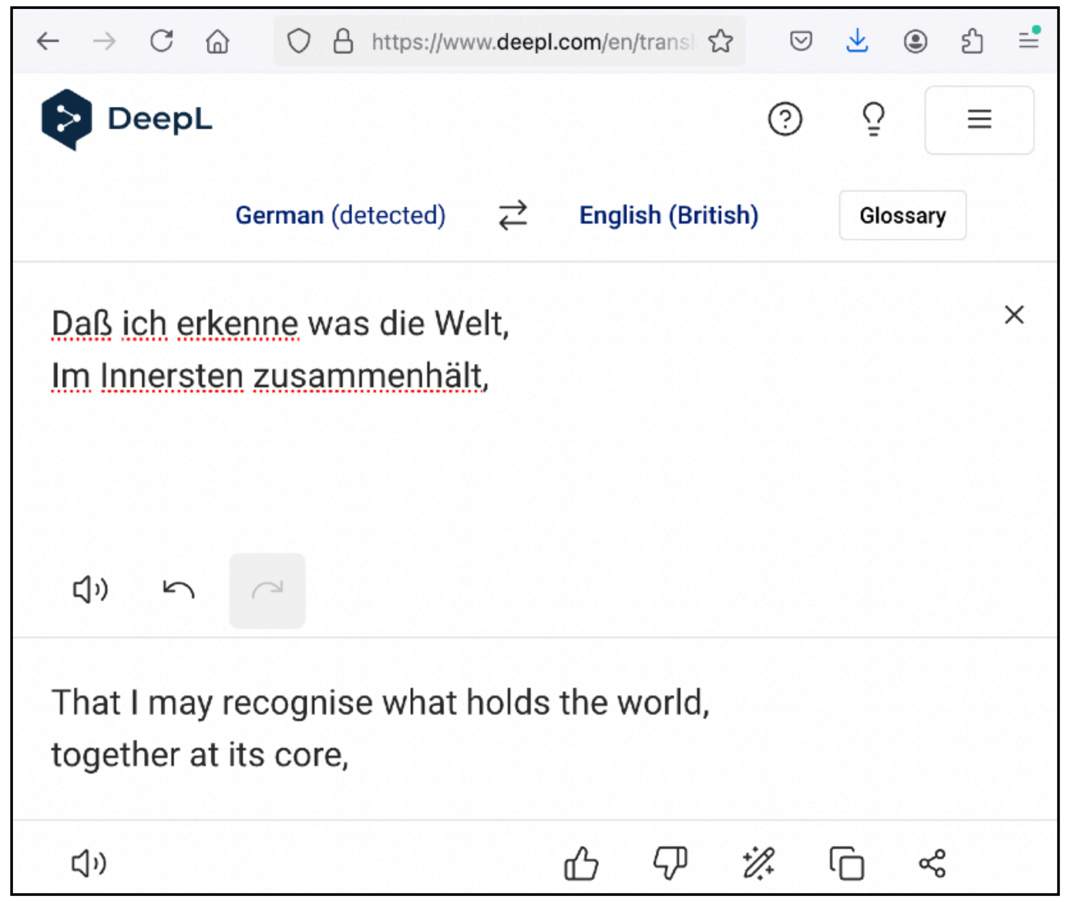
IBM Press release, January 8, 1954

New York, January 7..... Russian was translated into English by an electronic “brain” today for the first time.

[…]

A girl who didn’t understand a word of the language of the Soviets punched out the Russian messages on IBM cards. The “brain” dashed off its English translations on an automatic printer at the breakneck speed of two and a half lines per second.

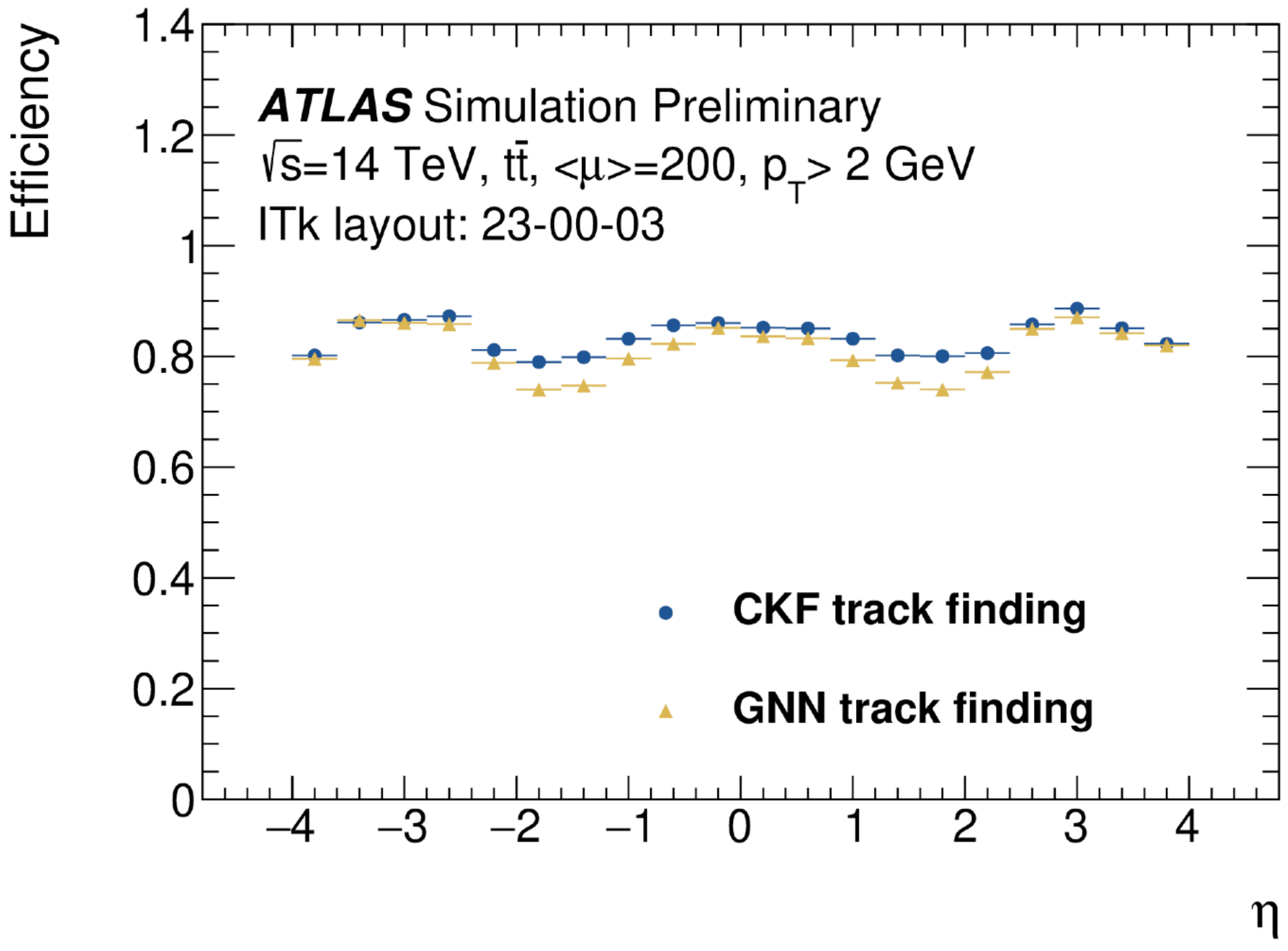
However, the triumphant headlines hid one little detail. No one mentioned … (link)



Today: ML achieves what classical algorithms do not

Reconstruction of charged particle tracks

ATLAS IDTR-2023-06

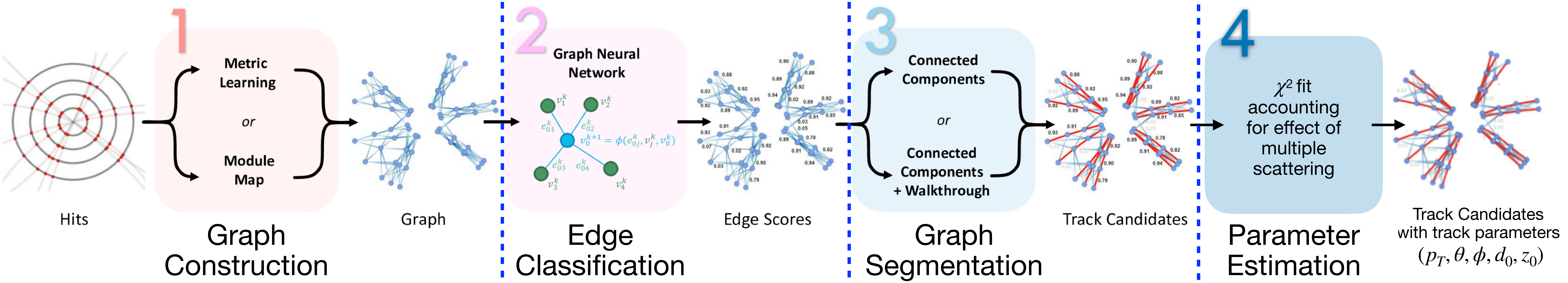


ATLAS input to 2025 update of European strategy for particle physics: ATL-SOFT-PUB-2025-002

“Now, machine learning is being developed for lower-level tasks […]

These developments have a very high threshold to adoption: they must improve upon the well-understood and extremely precise methods that have been used in ATLAS searches and measurements to date, not introduce any features […]”

Towards deployment



Track Reco

ModuleMapGraph ::GraphCreator

NVIDIA CUDA

Industry Inference Engineering

GNN

PyTorch ONNX TensorRT

ModuleMapGraph ::TrackReco

NVIDIA CUDA

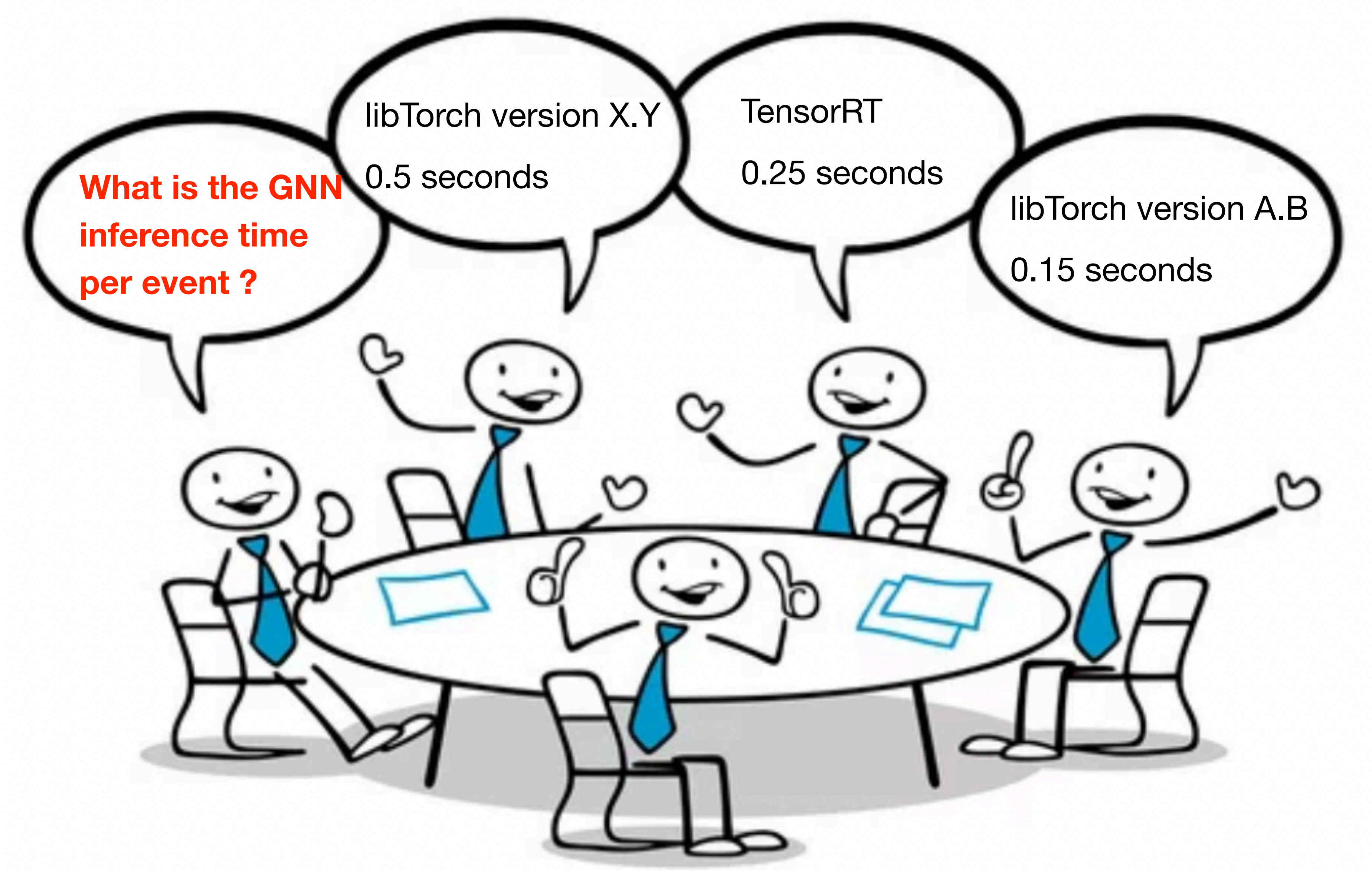
acts

tracc

NVIDIA CUDA SYCL

Towards deployment

In contrast:
have well-understood benchmarks for our CPU-based workloads



Benchmark Results

CPU	HS06	Clock speed (MHz)	L2+L3 cache size (grand total, KB)	Cores (runs)
Intel Xeon E5-2660v3	488	2600	5120+51200	40 HT on
Intel Xeon E5-4669v4	1836	2200	22528+225280	176 (HT on)
Intel Xeon E5-2699v4	987	2200	11264+112640	88 (HT on)
Intel Xeon E5-2620v4	305	2100	4096+40960	32
Intel Xeon Gold 6130	577	2100	32768+45056	32 (HT off)

Track Reco

L2IT
ModuleMapGraph ::GraphCreator
NVIDIA CUDA

Industry Inference Engineering
GNN
PyTorch ONNX
TensorRT

L2IT
ModuleMapGraph ::TrackReco
NVIDIA CUDA

acts
tracc
NVIDIA CUDA SYCL

Towards deployment

https://gitlab.cern.ch/gnn4itkteam/ModuleMapGraph

Standalone C++ and CUDA code to Compute a Module Map and to build Graph from it

Module Map Graph library
Track reconstruction for HEP
copyright © 2022,2023,2024,2025 COLLARD Christophe
copyright © 2022,2023 ROUGIER Charline

Laboratoire des 2 Infinis - Toulouse - France (L2IT)

contributors:
CAILLOU Sylvain
HUTH Benjamin
KAKATI Nilotpal
SHIELDS Edward
STARK Jan
VALLIER Alexis

Use standard tools whenever we can.

Develop our own when we have to.

Track Reco

L2IT	Industry Inference Engineering	L2IT	acts
ModuleMapGraph ::GraphCreator	GNN	ModuleMapGraph ::TrackReco	tracc
NVIDIA CUDA	PyTorch, ONNX, TensorRT	NVIDIA CUDA	NVIDIA CUDA, SYCL

Conclusions/observations

Reviewed progress in one of the ML-based charged particle tracking algorithms from its start in 2017 until now.
Now moving to deployment and production.

Must not forget: we are drinking data out of a firehose ! Implementation must run **fast**.

Fast inference is also important for industry and other other fields.
Benefit a lot from fast inference engines from industry and academia that do the heavy lifting for us.

- would be nice to have standard benchmarks (that test both hardware and inference software), à la HS06 or HS23
- ideally these also need to run affordably on CPUs

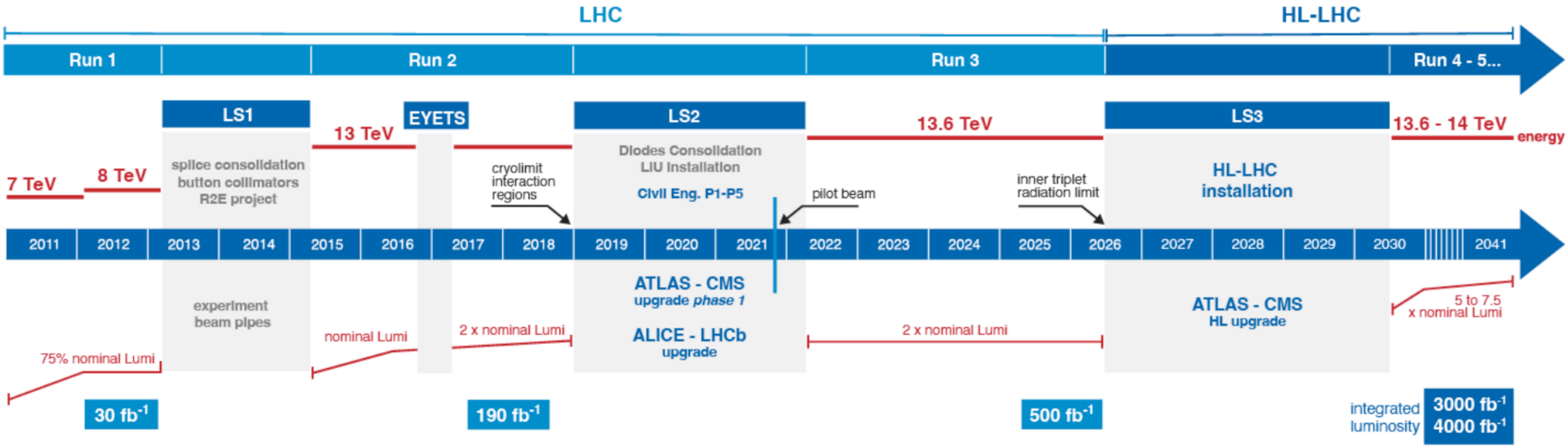
Need very little domain-specific, dedicated GPU code, but it needs to be efficient.

- E.g. for "data preparation": graph creation for use with GNNs. Presented MMG package.
- Train more people that can design these codes ?

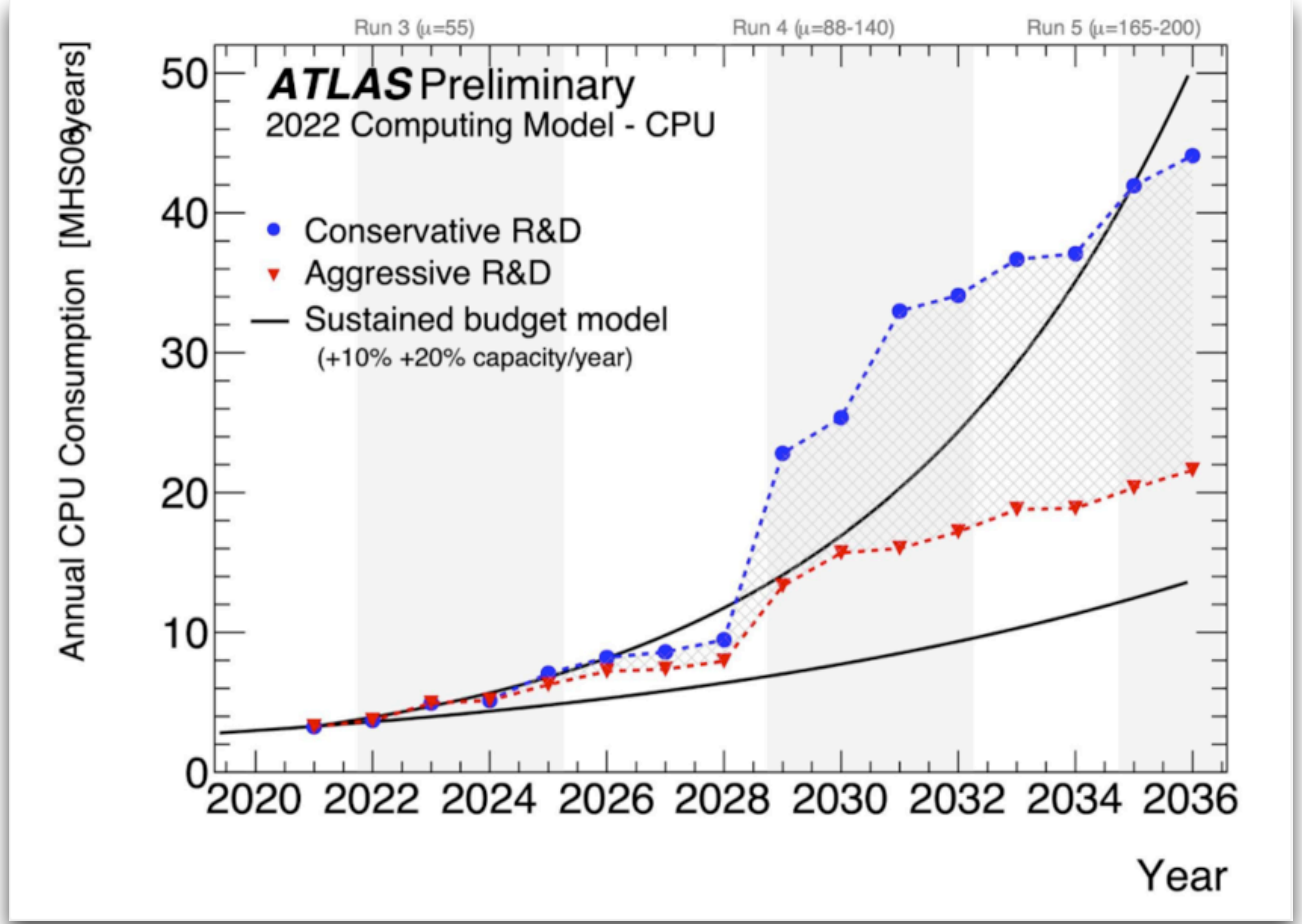
Do not hesitate to get in touch with us; eager to hear experience/needs from others.

Backup material

LHC and HL-LHC schedule



Projected computing needs



“10 years to prepare ourselves” for HL-LHC (statement from 2017)

- *Community white paper* (2017)
  - Algorithms, infrastructure, data access…

- Specific actions:
  - HEP Software Foundation (HSF)
  - Software Institute for Data-Intensive Sciences (SIDIS)
  - Creation of the Journal “Computing and software for big Science” (Springer)
  - IRIS-HEP (NSF project, US)
  - International project “Data Organization, Management and Access” (DOMA)

arXiv.org > physics > arXiv:1712.06982

Search... All fields Search
Help | Advanced Search

Physics > Computational Physics

A Roadmap for HEP Software and Computing R&D for the 2020s

Johannes Albrecht, Antonio Augusto Alves Jr, Guilherme Amadio, Giuseppe Andronico, Nguyen Anh-Ky, Laurent Aphecetche, John Apostolakis, Makoto Asai, Luca Atzori, Marian Babik, Giuseppe Bagliesi, Marilena Bandieramonte, Sunanda Banerjee, Martin Barisits, Lothar A.T. Bauerdick, Stefano Belforte, Douglas Benjamin, Catrin Bernius, Wahid Bhimji, Riccardo Maria Bianchi, Ian Bird, Catherine Biscarat, Jakob Blomer, Kenneth Bloom, Tommaso Boccali, Brian Bockelman, Tomasz Bold, Daniele Bonacorsi, Antonio Boveia, Concezio Bozzi, Marko Bracko, David Britton, Andy Buckley, Predrag Buncic, Paolo Calafiura, Simone Campana, Philippe Canal, Luca Canali, Gianpaolo Carlino, Nuno Castro, Marco Cattaneo, Gianluca Cerminara, Javier Cervantes Villanueva, Philip Chang, John Chapman, Gang Chen, Taylor Childers, Peter Clarke, Marco Clemencic, Eric Cogneras, Jeremy Coles, Ian Collier, David Colling, Gloria Corti, Gabriele Cosmo, Davide Costanzo, Ben Couturier, Kyle Cranmer, Jack Cranshaw, Leonardo Cristella, David Crooks, Sabine Crépé-Renaudin, Robert Currie, Sünje Dallmeier-Tiessen, Kaushik De, Michel De Cian, Albert De Roeck, Antonio Delgado Peris, Frédéric Derue, Alessandro Di Girolamo, Salvatore Di Guida, Gancho Dimitrov, Caterina Doglioni, Andrea Dotti, Dirk Duellmann, Laurent Duflot, Dave Dykstra, Katarzyna Dziedziniewicz-Wojcik, Agnieszka Dziurda, Ulrik Egede, Peter Elmer, Johannes Elmsheuser, V. Daniel Elvira, Giulio Eulisse, Steven Farrell, Torben Ferber, Andrej Filipcic, Ian Fisk, Conor Fitzpatrick, José Flix, Andrea Formica, Alessandra Forti, Giovanni Franzoni, James Frost, Stu Fuess, Frank Gaede, Gerardo Ganis, Robert Gardner, Vincent Garonne, Andreas Gellrich et al. (210 additional authors not shown)

(Submitted on 18 Dec 2017 (v1), last revised 19 Dec 2018 (this version, v5))

Particle physics has an ambitious and broad experimental programme for the coming decades. This programme requires large investments in detector hardware, either to build new facilities and experiments, or to upgrade existing ones. Similarly, it requires commensurate investment in the R&D of software to acquire, manage, process, and analyse the shear amounts of data to be recorded. In planning for the HL-LHC in particular, it is critical that all of the collaborating stakeholders agree on the software goals and priorities, and that the efforts complement each other. In this spirit, this white paper describes the R&D activities required to prepare for this software upgrade.

Download:
- PDF
- Other formats

Current browse context:
physics.comp-ph
< prev | next >
new | recent | 1712

Change to browse by:
hep-ex
physics

References & Citations
- INSPIRE HEP (refers to | cited by)
- NASA ADS

Export citation
Google Scholar

Bookmark

(link)

- The 2020 update of the EU strategy for particle physics

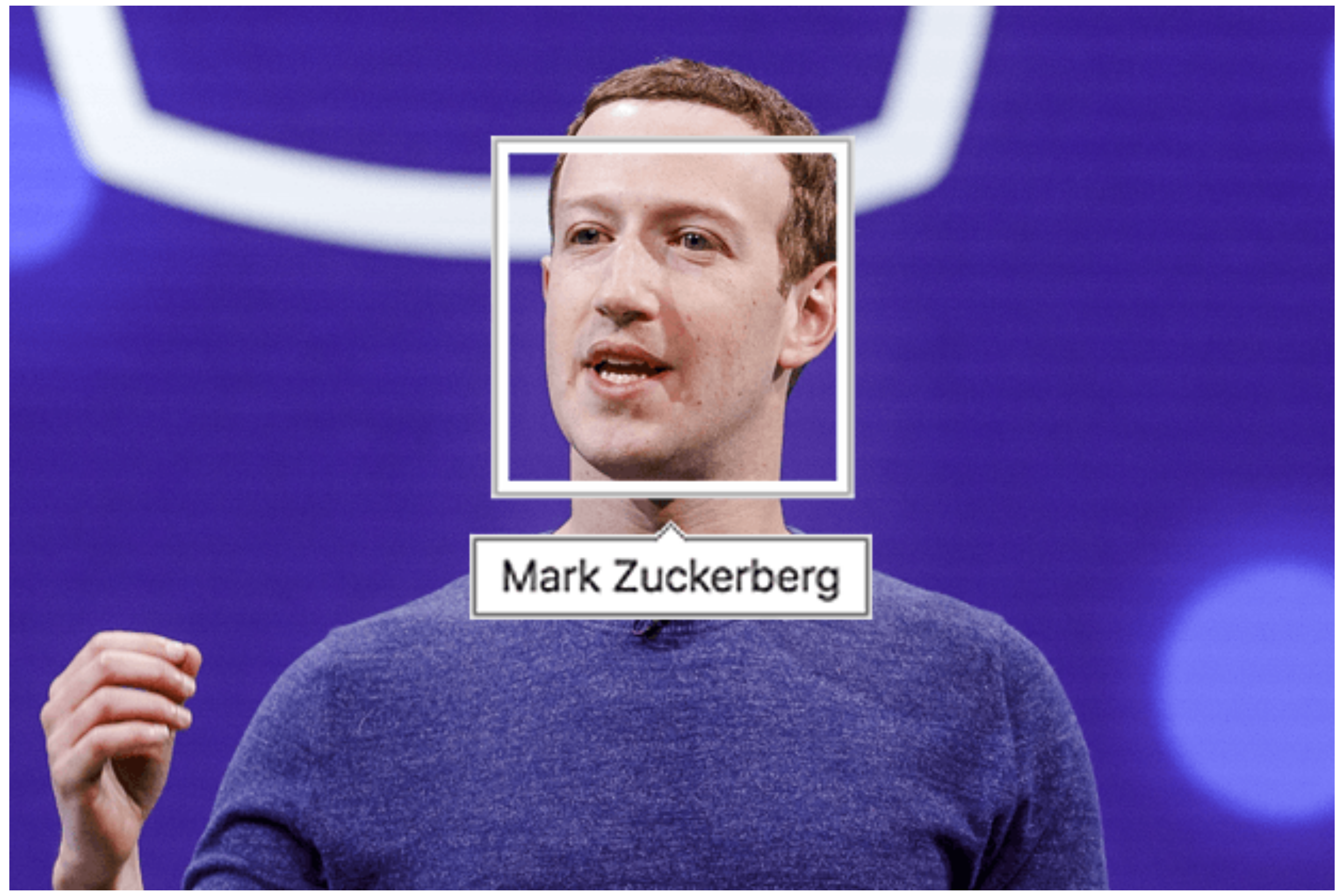
D. Large-scale data-intensive software and computing infrastructures are an essential ingredient to particle physics research programmes. The community faces major challenges in this area, notably with a view to the HL-LHC. As a result, the software and computing models used in particle physics research must evolve to meet the future needs of the field. ***The community must vigorously pursue common, coordinated R&D efforts in collaboration with other fields of science and industry, to develop software and computing infrastructures that exploit recent advances in information technology and data science. Further development of internal policies on open data and data preservation should be encouraged, and an adequate level of resources invested in their implementation.***

(link)

Machine Learning for track pattern recognition ?

Challenge on Kaggle platform (in 2018): [(link)]

Article in proceedings of CHEP 2018: [(link)]

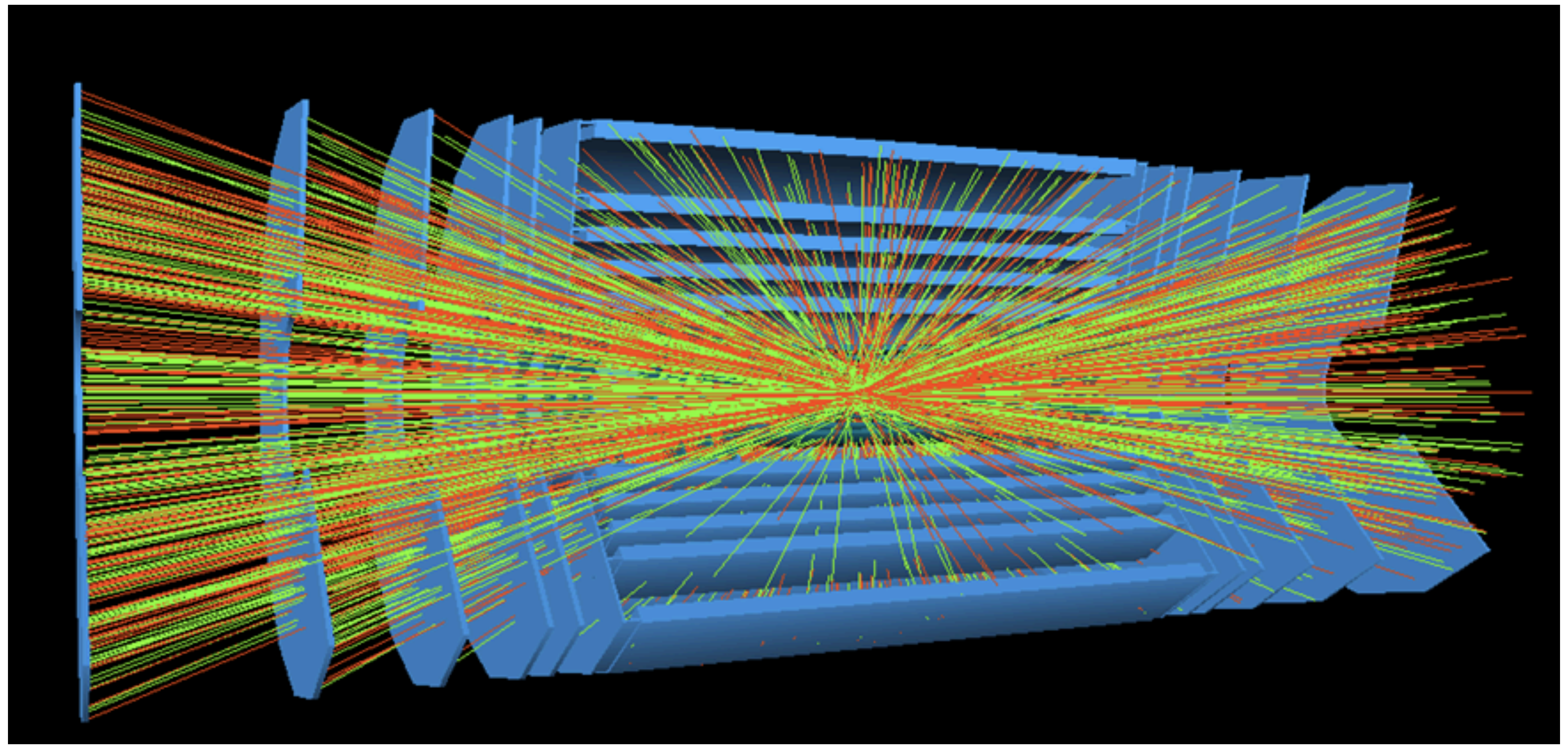


622 * 415 pixels

a large fraction carries information about the person

Can't use the same tools

How to present tracking data to a neural network ?



ATLAS tracker for HL-LHC:
5 * 10^9 readout channels
~3 * 10^5 3D space-points per event

=> data are *sparse*

ACORN tracking software

https://gitlab.cern.ch/gnn4itkteam/acorn

GNN4ITkTeam / acorn

acorn

documentation | pipeline passed | coverage 16.00% | DOI 10.5281/zenodo.14526454

This repository contains the framework used for developing, testing and presenting the GNN-based ITk track reconstruction project GNN4ITk.

Related work can be found here:

1. https://arxiv.org/abs/2103.06995
2. https://www.epj-conferences.org/articles/epjconf/abs/2021/05/epjconf_chep2021_03047/epjconf_chep2021_03047.html
3. https://cds.cern.ch/record/2815578?ln=en.

This repository is still under development and may be subject to breaking changes.

Get Started

To get started, run the setup commands (Install instructions section below), then take a look at the examples in the `examples` directory. Instructions and further details about the framework are available under the subdirectory of interest - `examples`, `acorn/stages` or `acorn/core`.

Install

IMPORTANT! Please use the `dev` branch to run all Examples: it is the latest version and is fully supported!

To install ACORN, assuming GPU capability with **cuda version >= 12.2**, run the following commands.

```
git clone --recurse-submodules ssh://git@gitlab.cern.ch:7999/gnn4itkteam/acorn.git && cd acorn
```