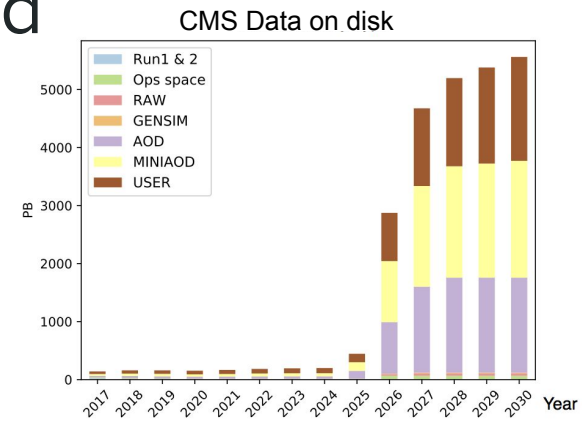
Software Working Group Introduction

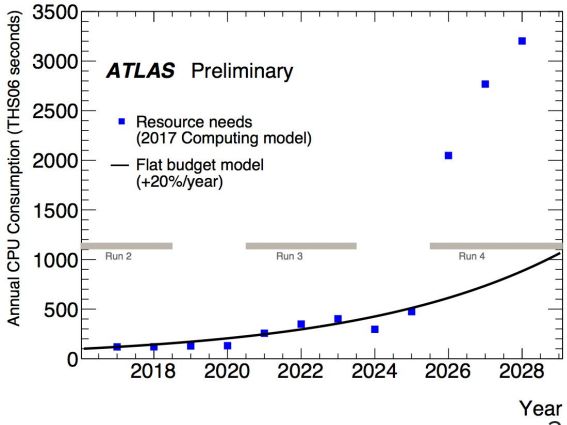
Graeme Stewart (co-convenor Jakob Blomer)

2018-03-16

Challenges to the HL-LHC and beyond

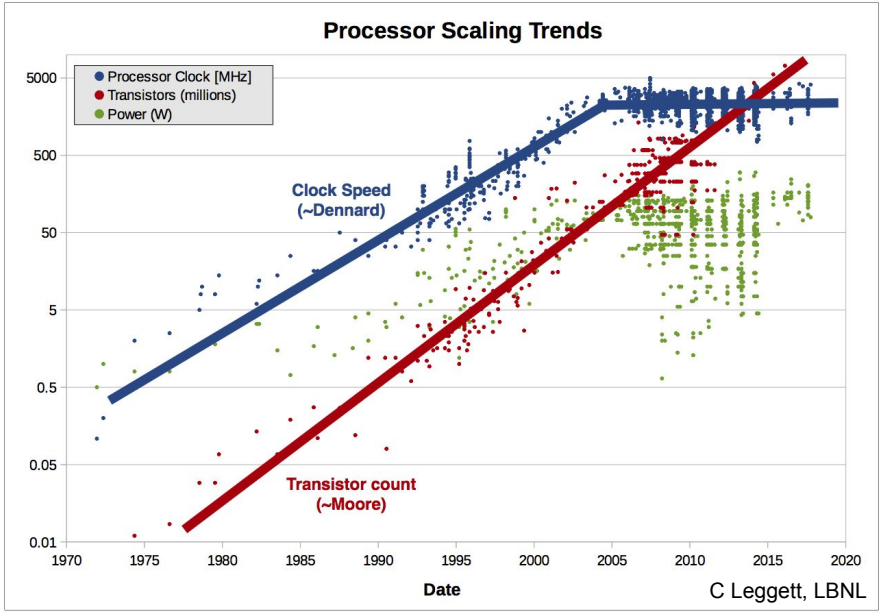
- High-Luminosity LHC is far from being a solved problem for software and computing
  - Naive extrapolation from today is not affordable
- Beyond HL-LHC, there are a number of different options for new machines
  - Lepton colliders (ILC, CLIC, FCC-ee) have overall less serious computing challenges
    - Require performant, robust, easy to use/deploy software
  - Hadron colliders (HE-LHC, FCC-hh) bring a massive data rate and complexity problem
    - Extreme for everything: generators, simulation, reconstruction, analysis
- Whatever the future, we pass through the HL-LHC on the way
  - HEP Software Foundation Community White Paper maps out that path





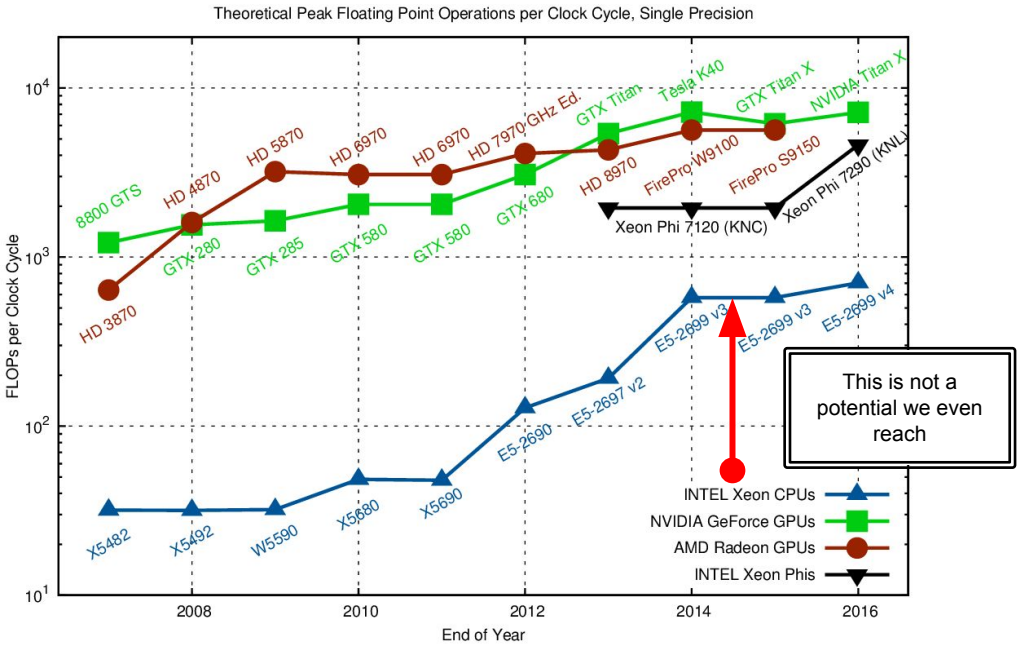
Processor evolution

- Moore's Law continues to deliver increases in transistor density
  - Doubling time is lengthening
- Clock speed increases stopped around 2006
  - No longer possible to ramp the clock speed as process size shrinks (Dennard scaling failed)
- So we are basically stuck at ~3GHz clocks from the underlying Wm^{-2} limit
  - This is the *Power Wall*
  - Limits the capabilities of serial processing
  - CPU based concurrency still in development for Run 3



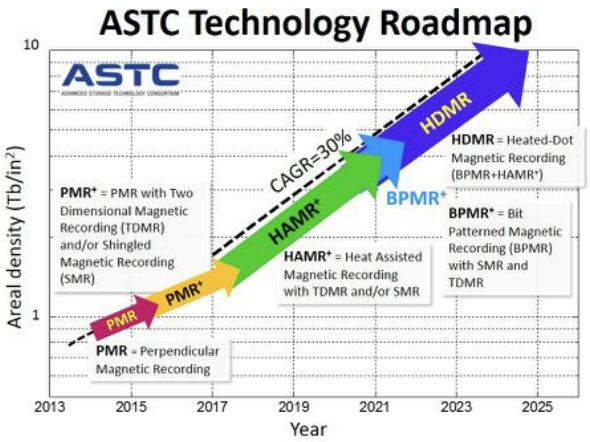
Compute Accelerators

- Most of the CPU die goes to things other than doing maths
  - Even CPU vector registers are hard for us to exploit
- Accelerators have a different model
  - Many cores, high floating point throughput, but lose a lot of ‘ease of use’
- We have to adapt to maintain our ability to use processors effectively

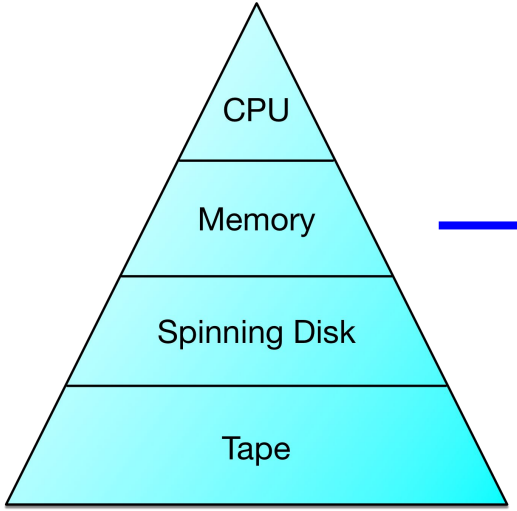


Other Technology Trends

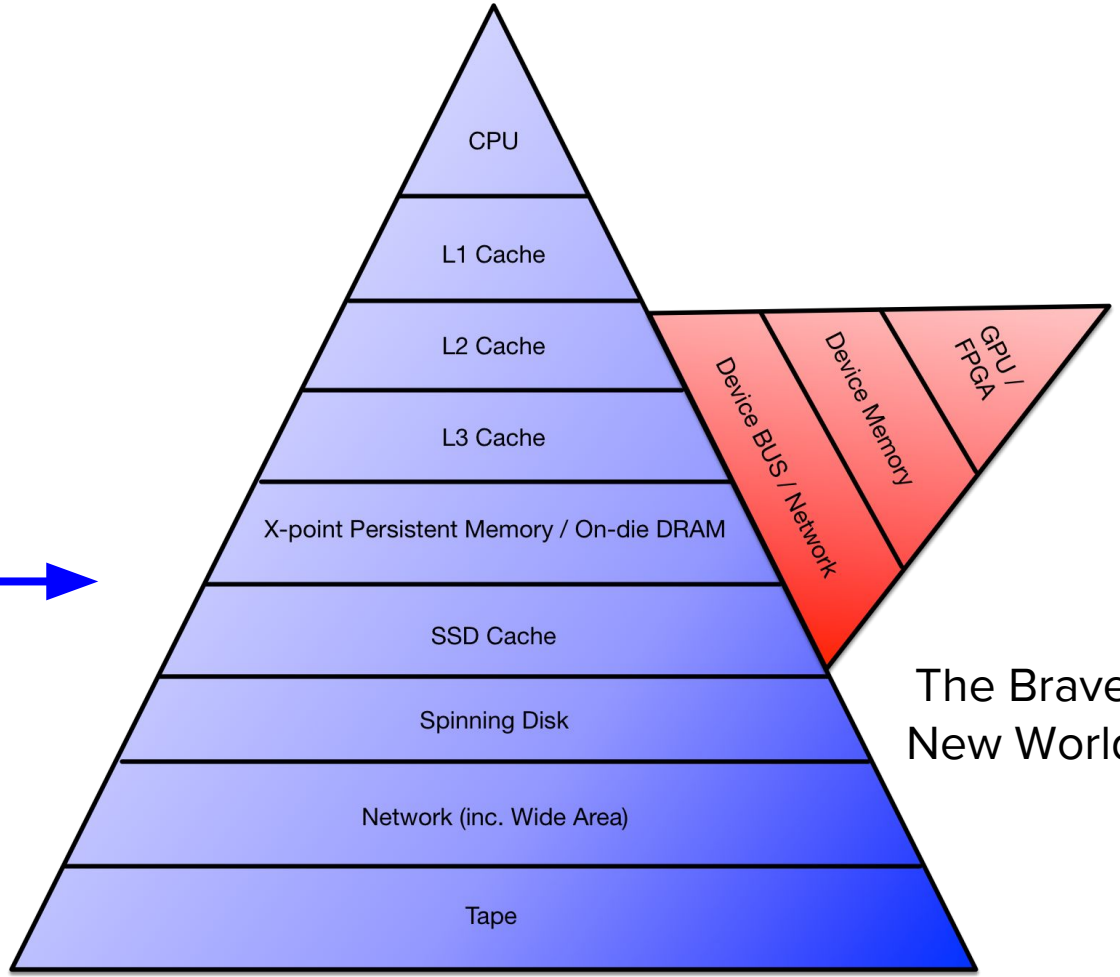
- Memory
  - DRAM improvements now modest
  - Overall, memory ‘landscape’ becomes more complex
    - Memory/storage boundary blurring
- Storage
  - Spinning disk capacity keeps climbing
    - Time to read and cost improves, but slowly
  - SSDs can read much faster, but price remains too high for bulk storage
  - Tape remains cheap to buy, slow to access with few companies left, O(1)
- Networks
  - Capacity increases expected to continue, latency will not change
  - Next generation networks offer capability to open channels between sites on demand
    - Useful, but an additional complexity
- Note: Game changer technologies might appear, but we cannot count on them



Meaning…



The Good Old Days



The Brave New World

Software needs and challenges

Goal: **ambitious** and **focused** work programme with milestones, deliverables and resource estimates

- Evolution and management of massive code bases created over many years
  - Current software is the base from which we design future detectors
- Meet the software challenges of future experiments
  - Very complex events - hard for reconstruction in particular
  - High rates - efficient, high speed data reduction pipelines
  - Huge volume - massive scale data and processing management
- Landscape for software becomes more varied
  - No more ‘free lunch’ from Moore’s Law
  - Harder to exploit hardware - need to adapt to accelerators and deep technology stack for data flow
- Advances from other fields offer promise, but need adapted
  - Data science and concurrency tools
- These are not problems that can be solved without investment
  - Software R&D program, running alongside detector R&D itself
  - Expect 5 years for advanced prototypes, deployment in 10 years

Software working group

- Open process
  - Gather ideas from the whole of the HEP software and computing community
  - Ensure alignment with developments outside CERN EP
  - 100 people on the mailing list
- Lightning Talks
  - Two sessions of lightning talks [1, 2] - open to anyone to propose a topic
  - Total of 28 short talks presented and discussed
    - Speakers from CERN EP and beyond
- Core group
  - Formed to distill these ideas and guide us towards R&D proposals
  - 15 people (LHC Experiments, CLiC, FCC, SFT, CERN IT)

Lightning talks

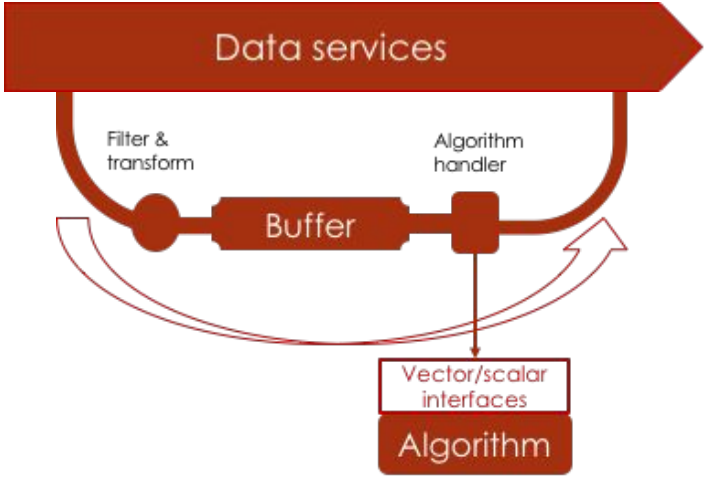
- Simulation for future experiments
- Reconstruction challenges for trackers and calorimeters
- New scalable analysis models
- Applied machine learning
- Tools for concurrency on heterogeneous resources
- Exabyte data flow and data management
- Support for new architectures and SoC systems
- Software integration

Big thanks to all the contributors!

		Challenge Area						Solution Technology						
Title	**Author**	**Simulation**	**TDAQ**	**Reconstruction**	**Analysis**	**Facility**	**Other**	**Concurrency and Accelerators**	**Machine Learing**	**Caching Layers**	**Advanced Networks**	**Storage**	**Turnkey**	**Other**
New Approach for ATLAS Detector Geometry Modelling	Alexander Sharmazanashvili						x							
Software Defined and Named Data Networking, and the Consistent Operations Paradigm	Harvey Newman					x					x			
VectorFlow, a subscription-based vector data-shipping service	Andrei Gheata	x		x				x						
Future technologies for efficient and light-weight handling of non event data	Gianluca Cerminara					x	x			x				
Filesystem-less high performance I/O of HEP Data	Danilo Piparo				x	x						x		
Track Reconstruction in a Concurrent world	Andreas Salzburger			x				x						
Machine Learning technologies applied to online event selection	Maurizio Pierini		x						x					
Machine Learning for Fast Simulation	Sofia Vallecorsa	x							x					
Automation of Data quality and certification with Deep Learning	Giovanni Franzoni		x						x					
Software for Detector Optimisation Studies in EP-LCD	Marko Petric						x						x	
Turnkey software solutions	Benedikt Hegner						x						x	
High throughput data analysis on future heterogeneous platforms	Danilo Piparo				x	x		x						
Heterogeneous computing	Felice Pantaleo					x		x						
Browser as a platform / compute device	Jakob Blomer						x							x
Machine Learning to empower physics modeling	Marilena Bandieramonte	x							x					
Handling inference efficiently in online/offline reconstruction software	Vincenzo Innocente			x		x			x					
Preparing for our Exascale data management challenge	Mario Lassnig					x				x	x	x		
Rethinking Data Center Computing	Giulio Eulisse			x	x	x		x						x
Embedded Linux for Run Control and ARM porting	Ralf Spiwoks		x										x	
Example of an embedded Linux Operating System and System-On-Chip based platform in the DAQ use case	Adrian Fiergolski		x										x	
To boldly go… - Application deployment frameworks for new computing environments	Radu Popescu					x								x
Future Distributed Analysis of HEP Data	Enric Tejedor Saavedra				x							x	x	
Machine learning solutions for simulation and reconstruction of highly granular calorimeters	Jan Kieseler			x				x						
APPAVO - Augmented tools for Particle Physics Analysis, Visualisation and Outreach	Eduardo Rodrigues				x		x							x
Development of a histogram library for concurrent programming	Andrea Bocci		x					x						
Clustering and Tracking with GPUs and machine learning	David Rohr			x				x	x					
Browser Based Detector Displays for Outreach & Education	Alexander Sharmazanashvili						x							x
tkLayout	Stefano Mersi						x						x	

One Lightning Example - VectorFlow

- Vectorisation is great when it happens, but difficult to achieve with our codes today
  - Problem is how to gather appropriate data and fill vector registers with it
    - Does not happen naturally for event by event processing
  - Gather data into a processing buffer from many places
  - Process through an algorithm that has a vectorised interface
- R&D on
  - Concurrency and performance effectiveness
  - Using vectorisation primitives in new areas
  - Integration into existing frameworks
  - Offloading into accelerators
- Adapting to new hardware is difficult work and dedicated expertise is needed to bridge between physics and software



EP department - a centre for software excellence

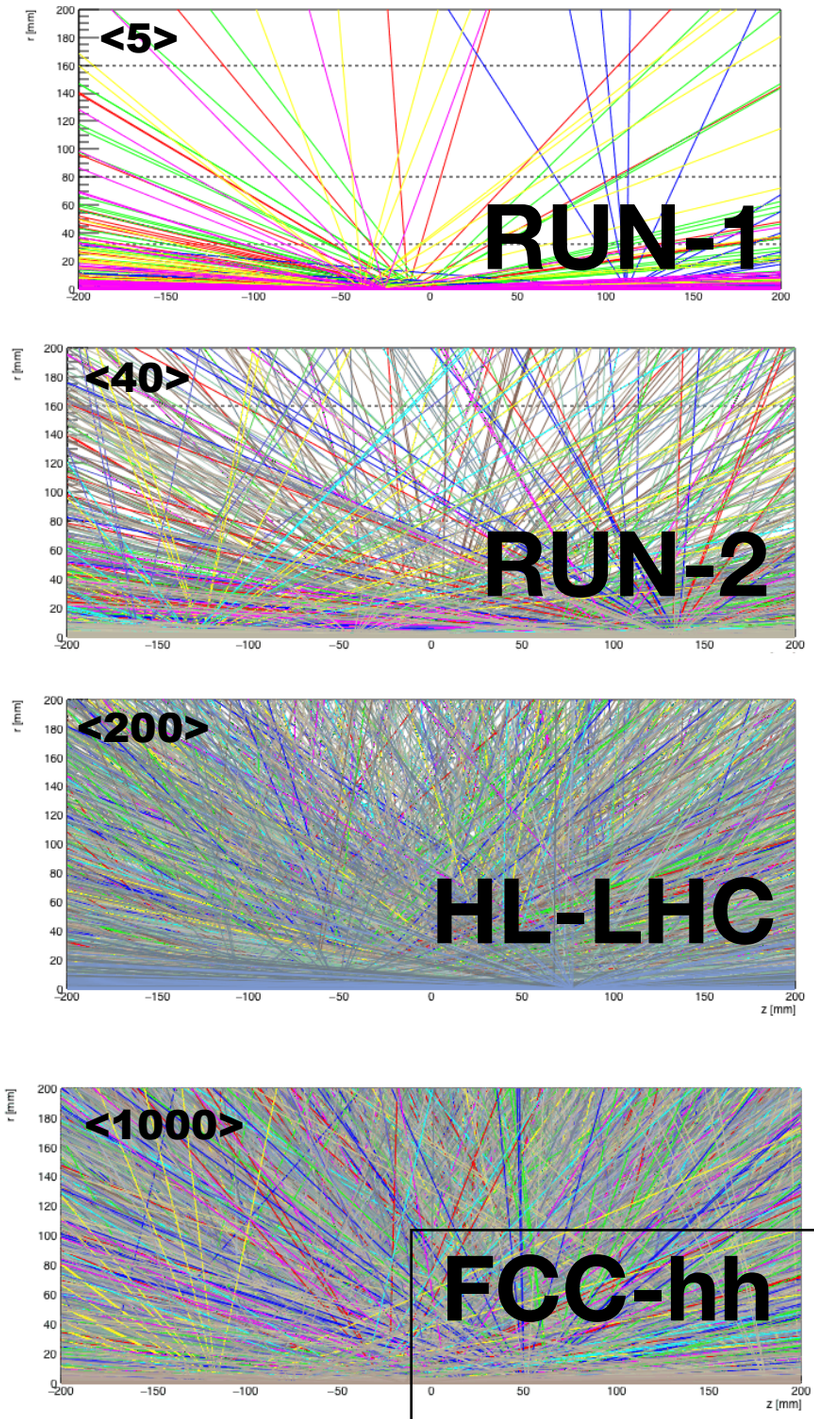
- Very strong software groups in current and future experiments
  - Including Phase II upgrades
  - Frameworks, Tracking, Data Quality, DAQ and Data Flow, Databases, Detector Description
- Key contributor to core HEP libraries
  - ROOT
  - Geant4
- Central role in distributed data management software and operations
  - Data Management and Workload Management
  - Large resource operations management: Trigger Farms, Tier-0s
  - CVMFS
- Close to CERN IT
  - WLCG operations and developments
  - Critical expertise in technology tracking
- Key player in community initiatives through HEP Software Foundation

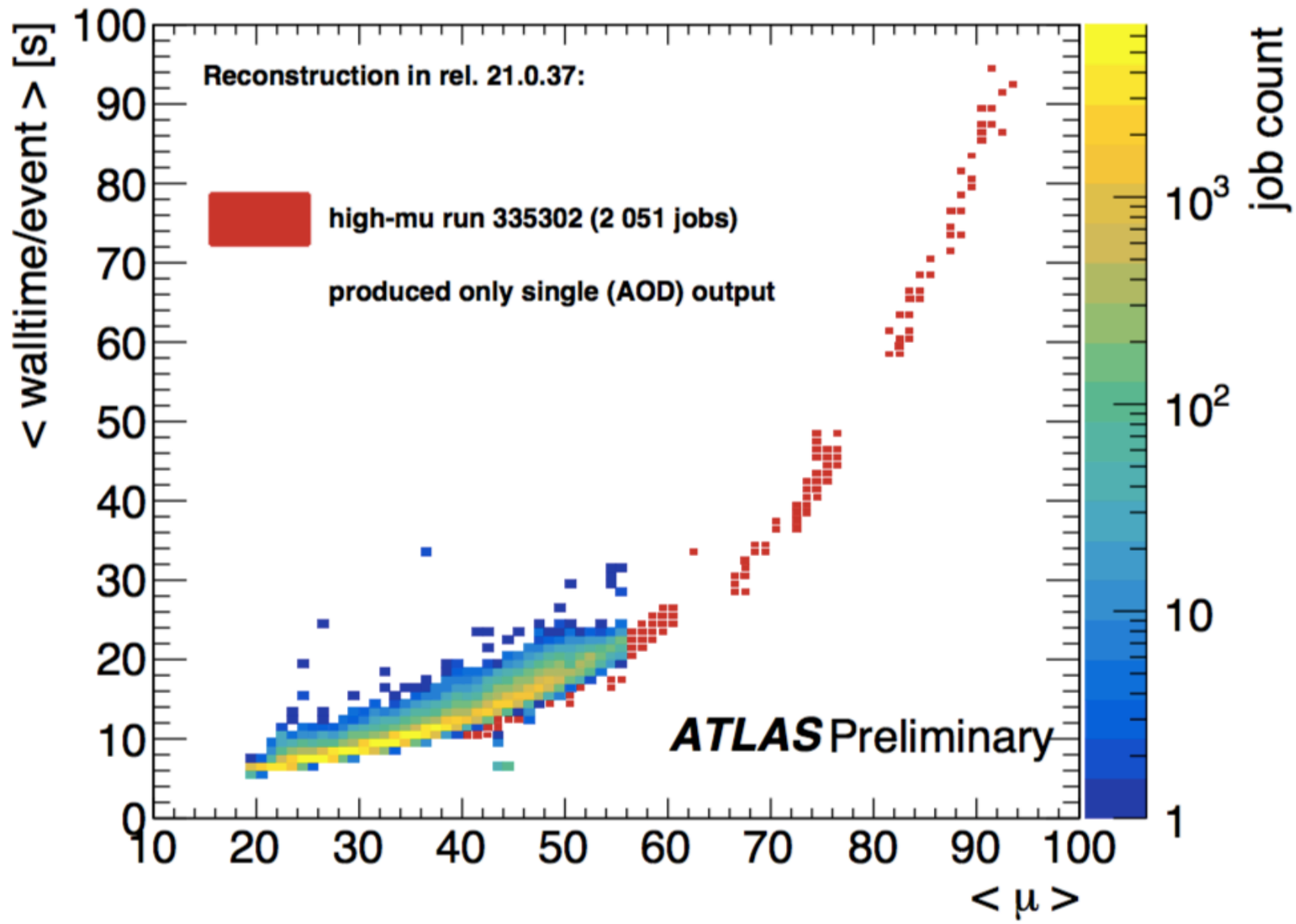
Many projects now picked up more widely than at LHC - a success!

Today’s talks

- Introduction
  - The landscape of challenges and opportunities
- Future Tracking
  - A key unsolved problem for future experiments
- Machine Learning
  - Applied data science and how it can be used in HEP
- End to End Physics
  - Software and data flow solutions for the exabyte era

These are not presented here as concrete R&D proposals, but as samples of some of the most interesting challenges and ideas from the problem and solution space



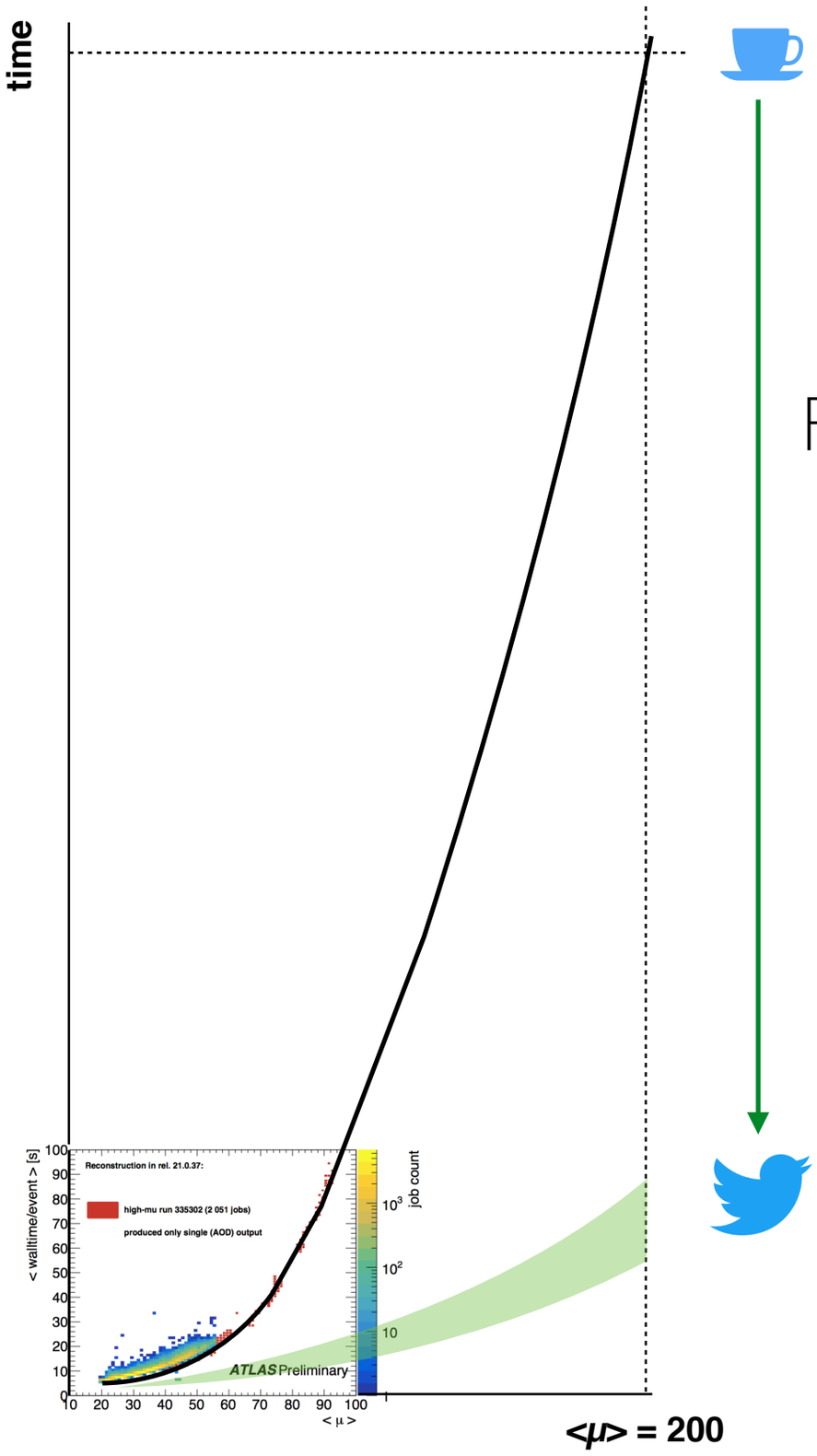


R&D for Future Tracking

A. Salzburger (CERN)

Images:
(left) longitudinal views of vertex region for various scenarios
(right) ALTAS Run-2 CPU scaling with <μ>

Track reconstruction Extrapolation HL-LHC



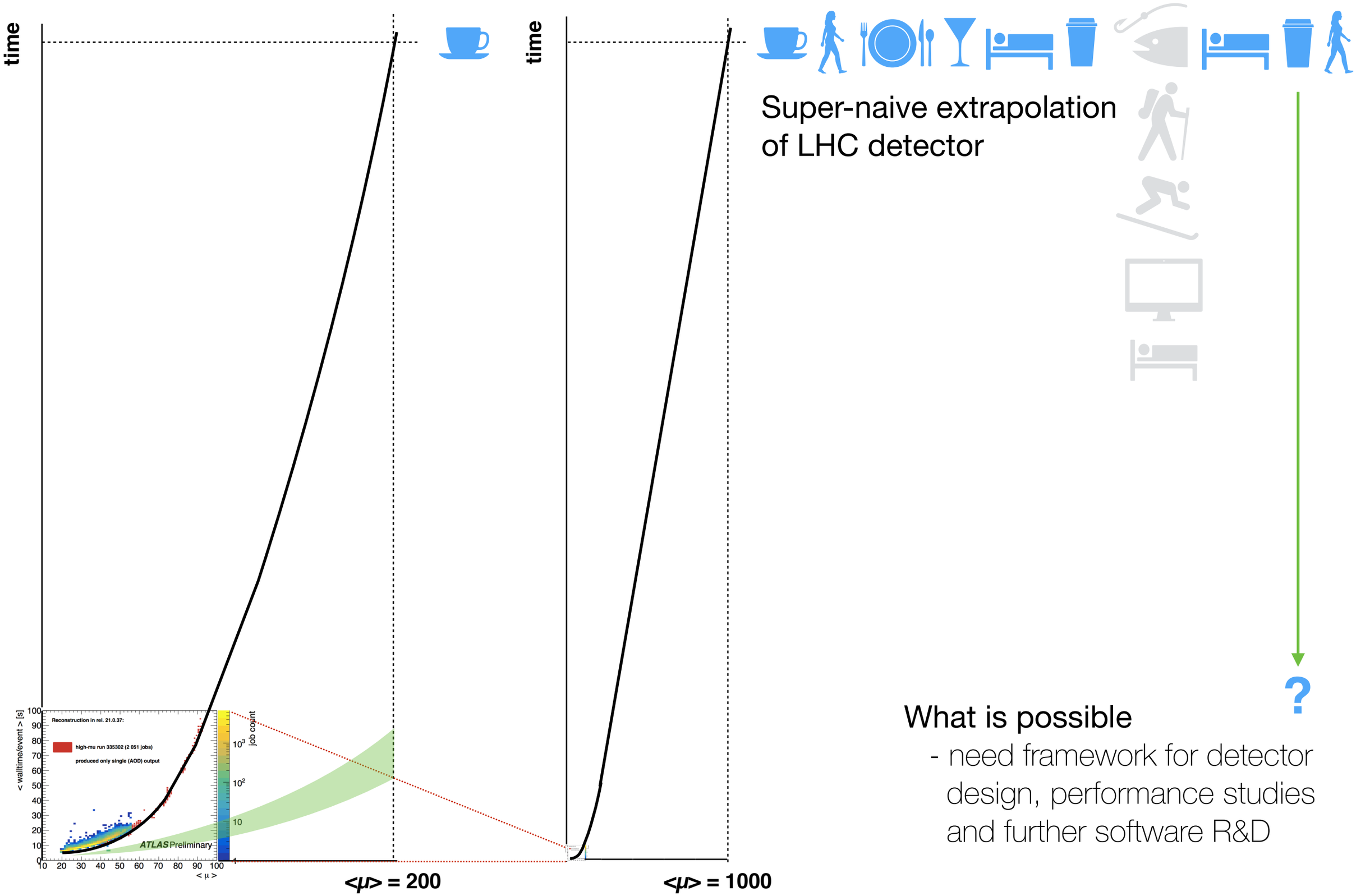
Naive extrapolation of LHC detector
- full reconstruction
- ATLAS/CMS & software designed for <μ>~23

Finding of particle trajectories
- combinatorial problem to solve
- **highly non-linear scaling** of CPU time with increasing event complexity
- dominant CPU consumer

Current numbers for HL-LHC detector
- track reconstruction only
- detector designed for <μ>=200
- current software adapted/optimized for this environment

Factor 10 increase in readout rate from 1kHz to 10 kHz
only partly compensated by hardware speedup

Track reconstruction Extrapolation HL-LHC / FCC-hh



What is possible

- need framework for detector design, performance studies and further software R&D

Risks & Gains

Risk is high

- we *risk* physics potential
  if we do not solve this solve this
- e.g. current LHC analyses saw
  already the advent of MC
  statistic limits for certain analyses

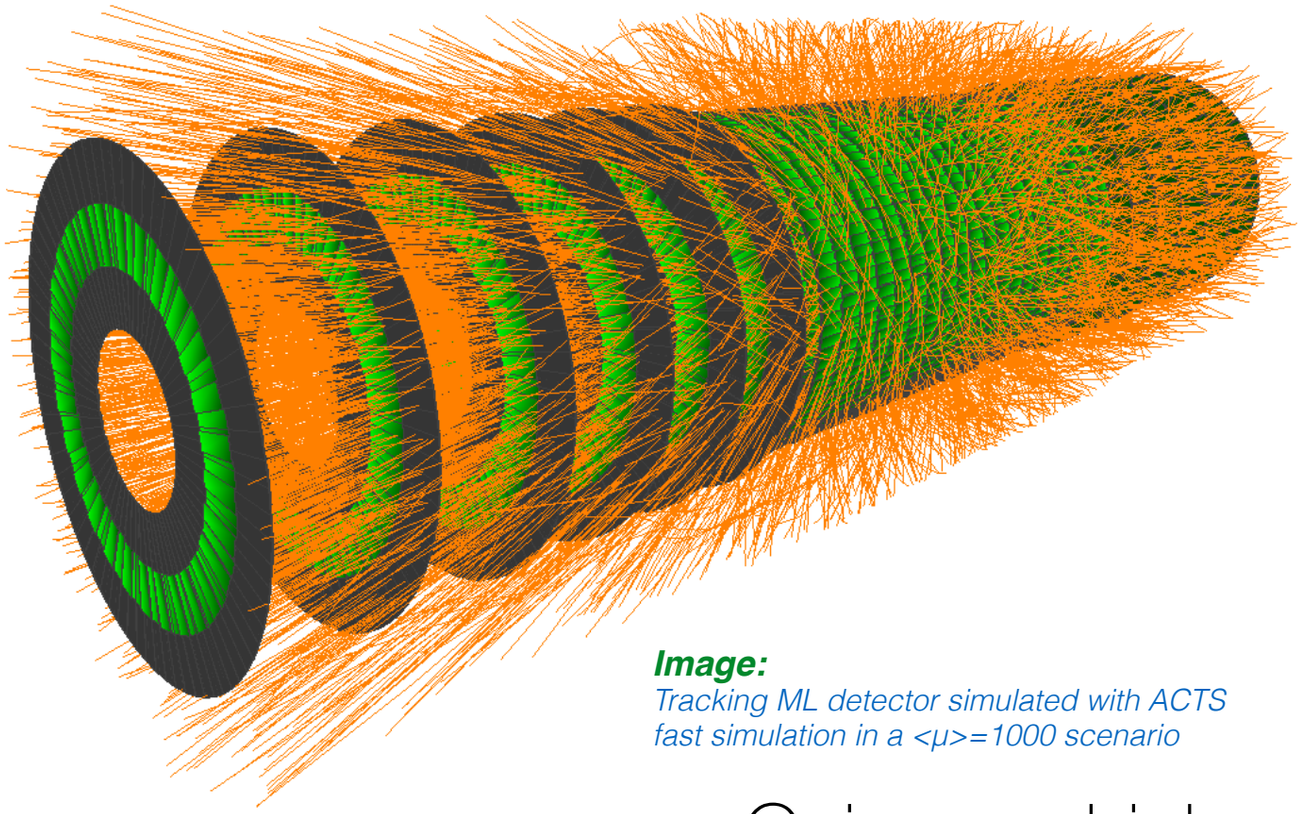


Image:
Tracking ML detector simulated with ACTS fast simulation in a <μ>=1000 scenario

Gains are high

Substantial R&D is needed several areas:

Code optimization	algorithmic R&D	(low n) Concurrency	Hardware acceleration	Machine Learning

- great **in-house expertise** at CERN that can be fostered
- **symbiotic projects** with **detector R&D, computing** & **machine learning**
- exciting times for **software and algorithm design**
- strengthen **CERN as excellence lab** for software

Track reconstruction R&D

Code optimization	algorithmic R&D	Concurrency	Hardware acceleration	Machine Learning

Community driven common software for track reconstruction

- cutting edge algorithmic solutions for "classical pattern recognition"
  preserving the excellent performance *(physics & failure rate) of LHC experiments*
- expert driven code optimization (strong link to SW R&D)

Common effort of online and offline reconstruction software

- incentives towards tracking at L1 trigger level / trigger-less readout

Inclusion of timing information in track finding & fitting

- synergies with detector R&D for timing detectors

On demand track reconstruction

- region or physics driven reconstruction setups

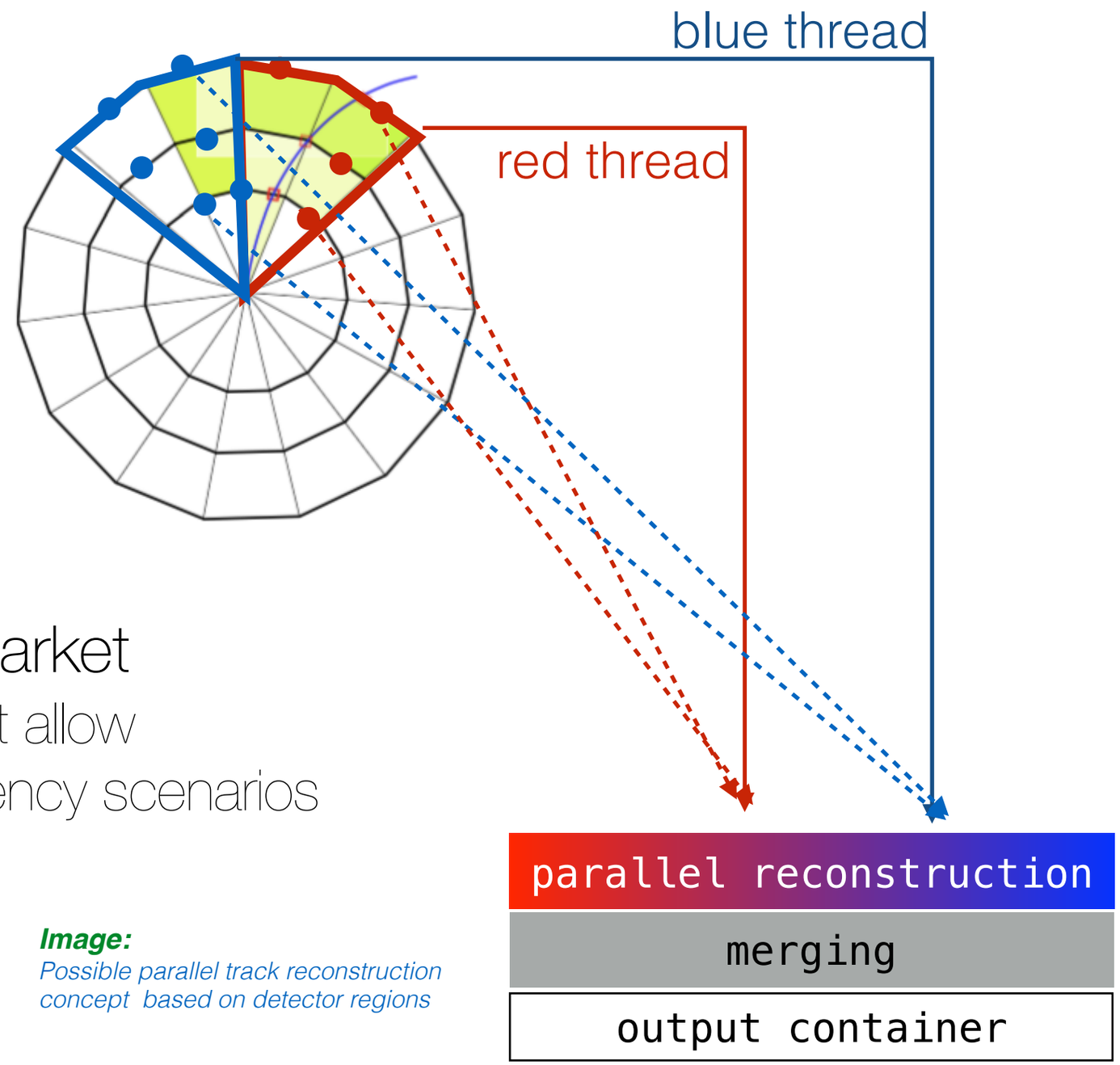
Overall need for an R&D platform for track reconstruction

Track reconstruction R&D

Code optimization	algorithmic R&D	Concurrency	Hardware acceleration	Machine Learning

Adapt track reconstruction for **concurrent execution**

- needs **substantial** work on current algorithms, data structures and data flows (vectorization)



Uncertainty in hardware market

- prepare **flexible toolkits** that allow adaption to several concurrency scenarios

Image:
Possible parallel track reconstruction concept based on detector regions

Example:
ACTS
https://indico.cern.ch/event/699252/contributions/2881457/attachments/1594869/2525543/2018-02-05-ACTS-Lightening.pdf

Track reconstruction R&D

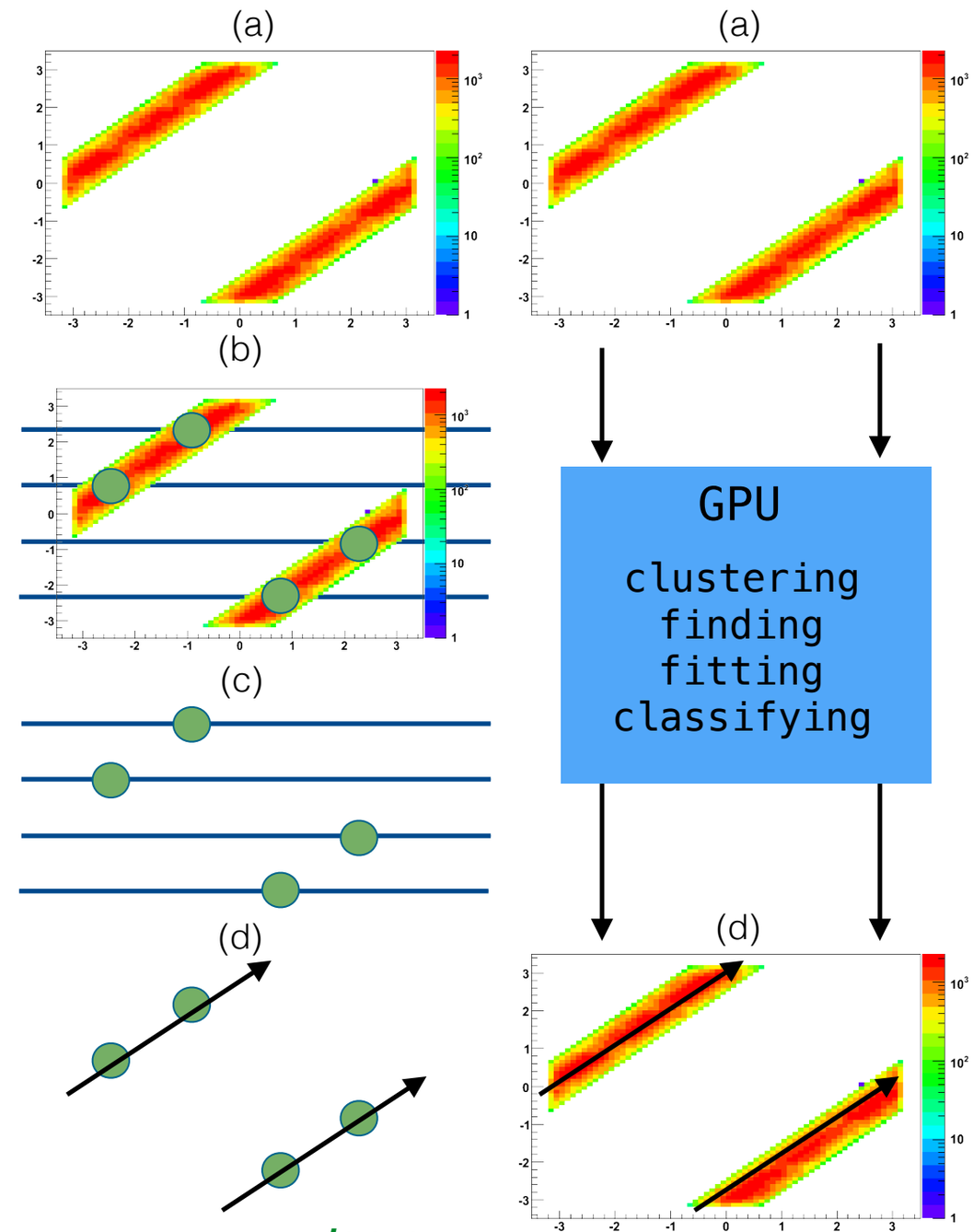
Code optimization	algorithmic R&D	Concurrency	**Hardware acceleration**	**Machine Learning**

GPUs in track reconstruction

- several areas where GPUs could be effectively used
- GPUs work extremely well for certain algorithms/data flow
  e.g. clustering, cellular automaton, hough transform, Kalman filter
- Machine learning applications are "designed" for GPUs

Use of FPGAs and associative memory in track reconstruction

- particular in trigger



Examples:
CMS cellular automaton on GPUs
Clustering & Tracking GPUs/ML
https://indico.cern.ch/event/702570/contributions/2905391/attachments/1606480/2549157/2018-02-26_CERN_RnD_Proposal.pdf

Image:
Traditional path to track finding (left), GPU enabled path directly on charge input (right)

Track reconstruction R&D

Code optimization	algorithmic R&D	Concurrency	**Hardware acceleration**	**Machine Learning**

Clustering hits together is typical ‘unsupervised learning’

- take advantage of the recent advances in the field

  integrate into current track reconstruction software stacks

Convolutional/Recurrent Neural Networks (CNNs/RNNs)

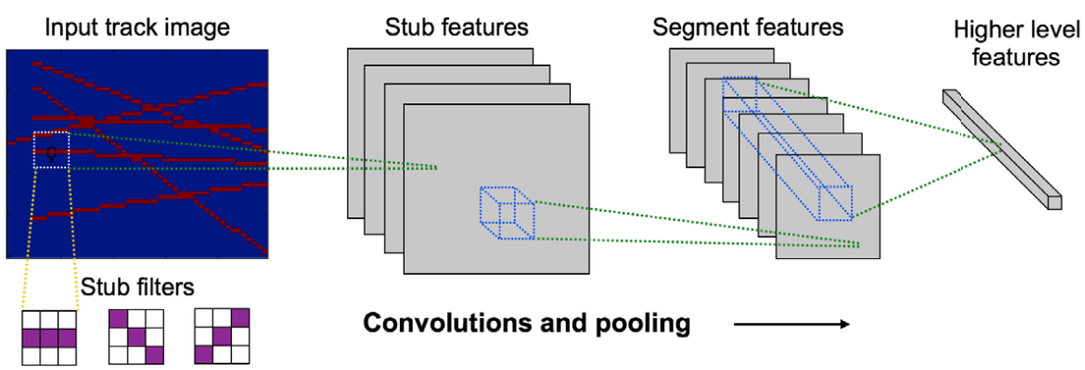
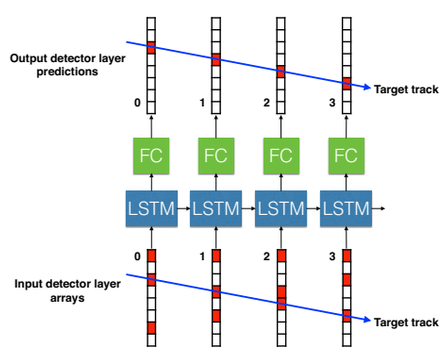


Image:
Convolutional Neural network for track finding via (sub-)feature pooling



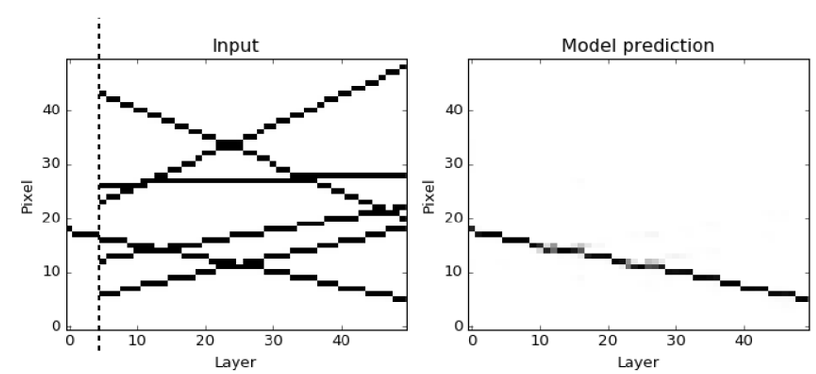


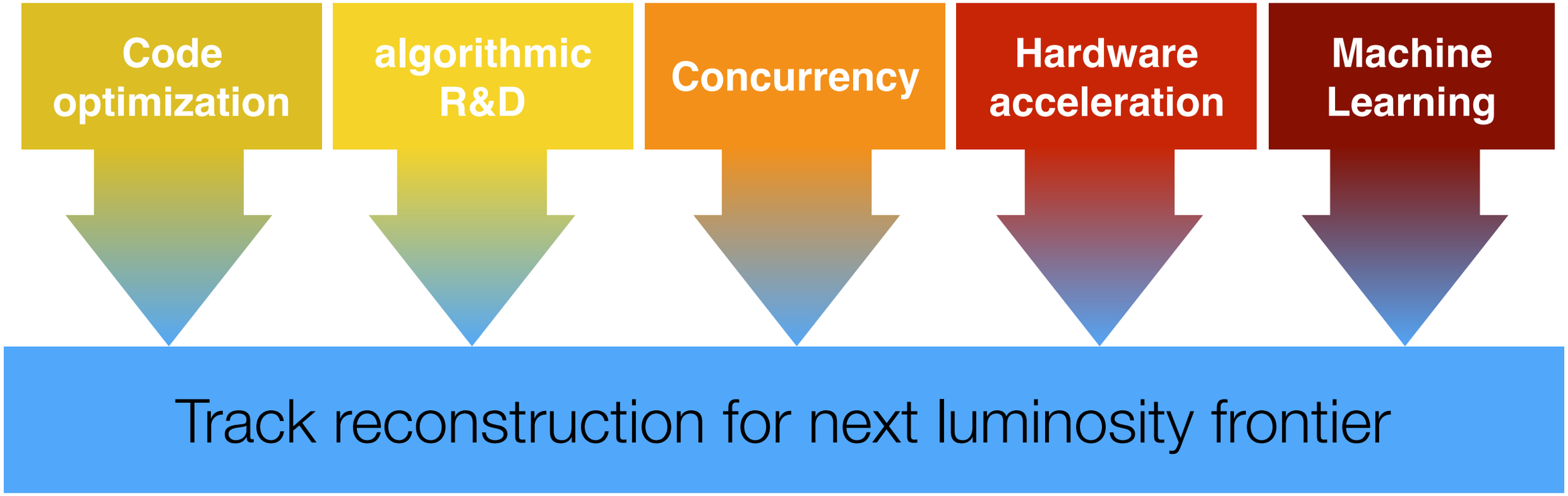
Image:
Recurrent Neural network for prediction via Long Short Term Memory (LSTM)

Examples & more information:

See talk by Maurizio

CERN co-organized, Tracking Machine Learning Challenge (Apr 2018)

Summary



Achieving this will **gain great** physics potential, though requires

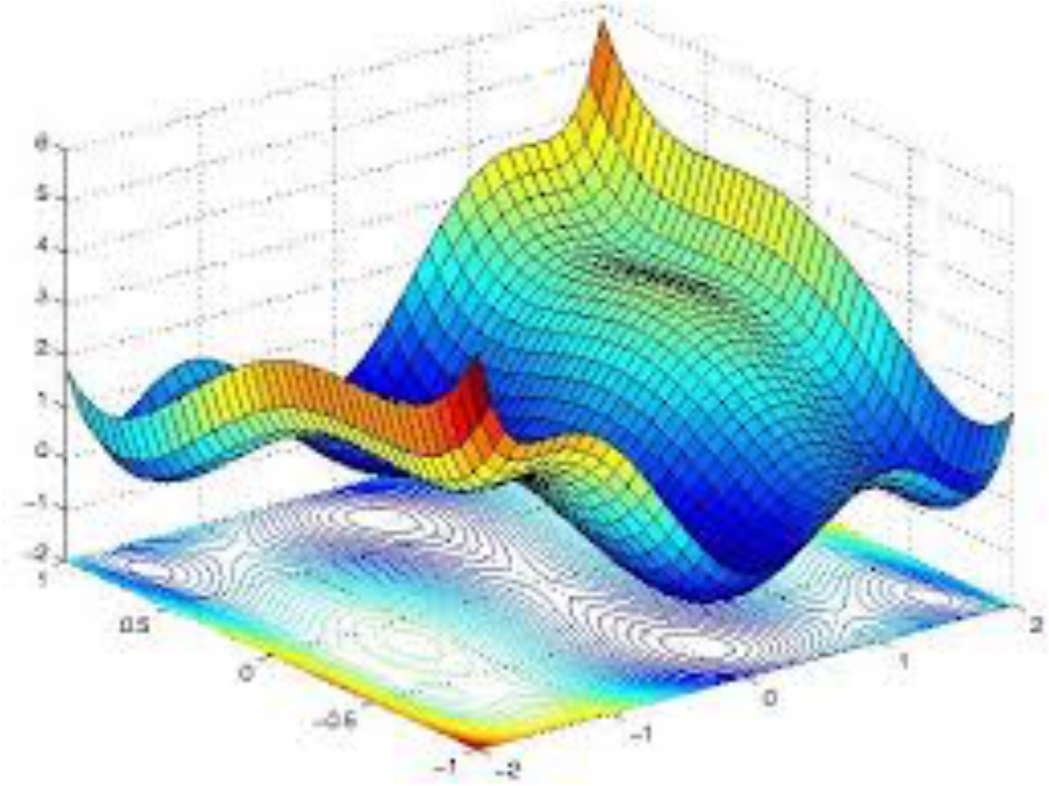
- R&D in all of these areas
- *preserve* the **excellent LHC physics performance** of track reconstruction
- *foster* and *strengthen* **in-house expertise**
- *work closely* with and alongside **detector R&D** lines
- *profit from*, *coordinate with* and *participate in* **ML R&D**
- *strengthen* **common, community-driven and open** software

Machine learning opportunities

Maurizio Pierini

What is Machine Learning

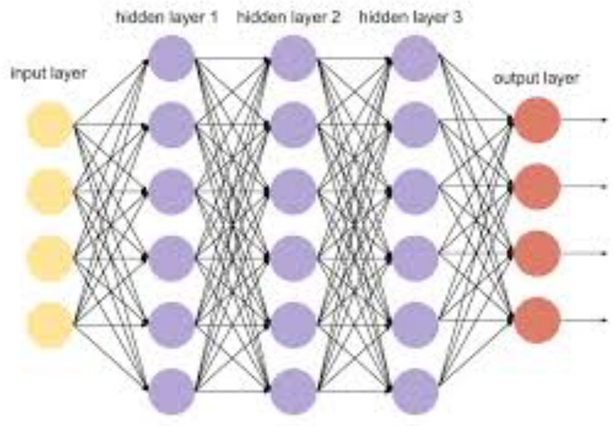
- *Machine learning is a technique by which an algorithm learns by example to accomplish a task*
  - *as opposed of being programmed to do so*
- *Deep Learning is the cutting-edge ML technology*
  - *based on "old-school" neural networks + augmented computational capabilities (e.g. GPUs)*
  - *the breakthrough is fast differentiability (back-propagation) allowing fast optimization*
  - *Learning non-linear functions, can be a fast shortcut to replace heavy processing tasks*

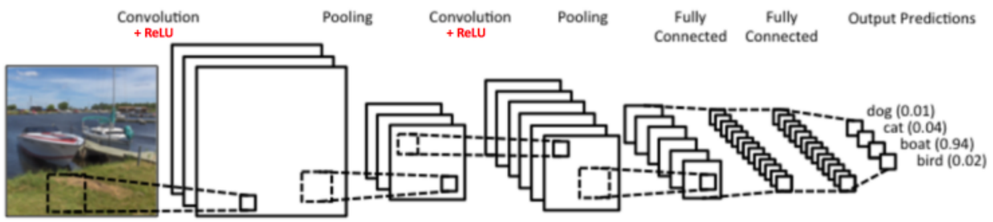


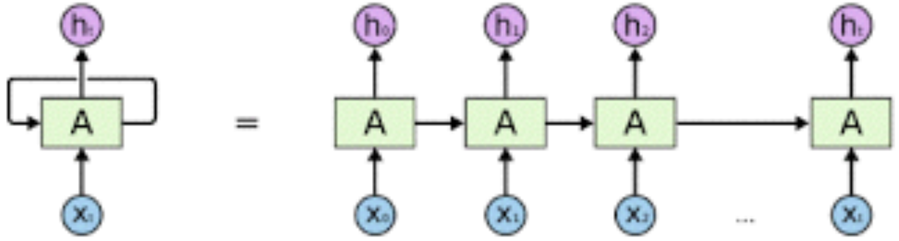
Training a Machine Learning algorithm consists in minimizing a complicated multi-dimensional function

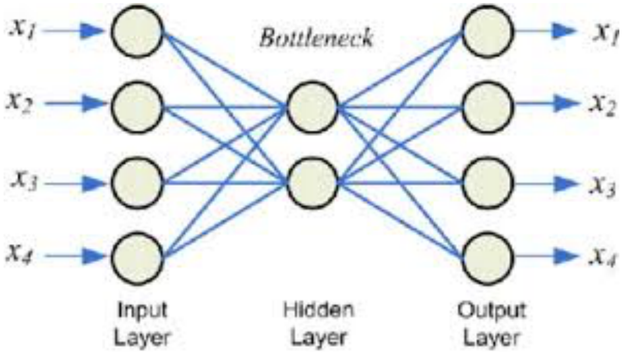
Deep Learning

- *New architectures proposed beyond classic fully-connected layers*
  - *Convolutional neural networks for image processing*
  - *Recursive neural networks for text processing*
  - *Generative adversarial networks for data generation*
  - *Autoencoders for anomaly detection (and data generation etc)*



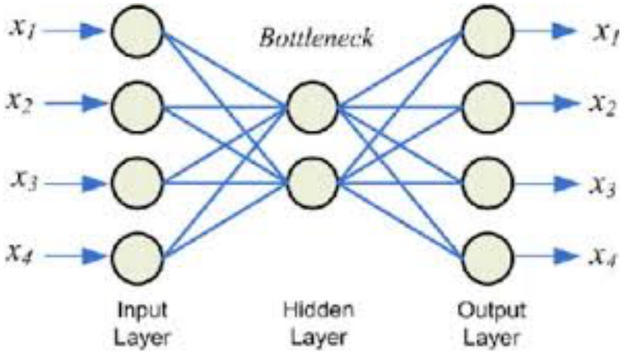






Directions for EP R&D

- *New architectures proposed beyond classic fully-connected layers*
  - *Convolutional neural networks for image processing*
  - *Recursive neural networks for text processing*
  - *Generative adversarial networks for data generation*
  - *Autoencoders for anomaly detection (and data generation etc)*

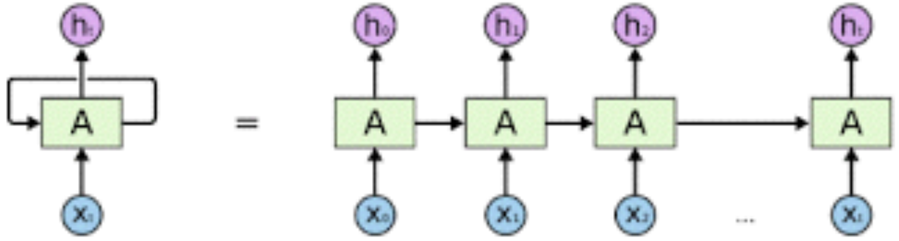


Detector reconstruction as image identification

Tracking, PF-compliant identification, etc.

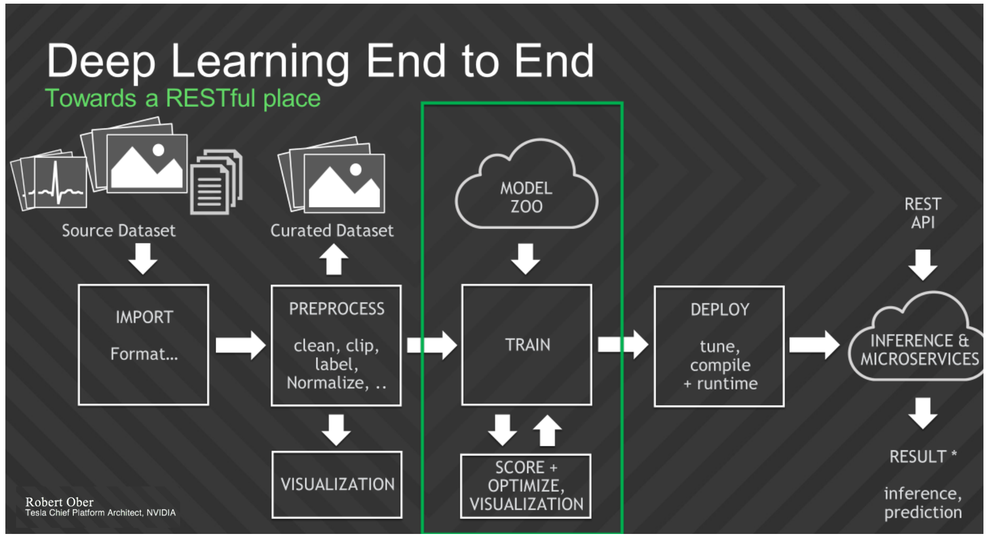
Faster simulation

Detector Monitoring, simulation, etc.



Acquiring critical expertise

- *ML has been used for decades (mainly BDTs @LHC)*
- *Experiments transitioning to Deep Learning since a few years*
  - *Mainly through the push of young PhD students who studied this @school*
  - *Several initiatives @CERN: iML group, Data Science seminars, Conferences and Schools, Data Challenges, Marie Curie networks, one ERC grant*
- *So far, focus is more on the CERN users as a community of data scientists developing models. For a sustainable investment in this direction*



- ***Gain expertise within the Department***
- ***Go beyond the model training: data pipeline, model development, inference (i.e., using the models in what we do)***

Not just Model development

- *Training infrastructure*
  - *Parallel problem, runs efficiently on parallel architectures (e.g., GPUs, FPGAs, and TPUs) with out-of-shelf software solutions (python data science libraries)*
  - *Many groups (ATLAS, CMS, LHCb, ALICE) have in-house resources and recipes where proof-of-principle studies are running.*
  - *A user-oriented structured and general solution will be needed*
- *Porting Inference to real-life activity*
  - *Can run efficiently on modern parallel architectures with less resources request*
  - *But has to be integrated into our frameworks. Customisation needed at this stage*
    - *A model out-of-the-box from TensorFlow takes +100MB of memory. Smarted ways exist, but need to be developed and adapted to our needs*
  - ***Need to develop DL-compliant next-generation software (and possibly DL-compliant next-generation detectors, trigger farms, etc)***

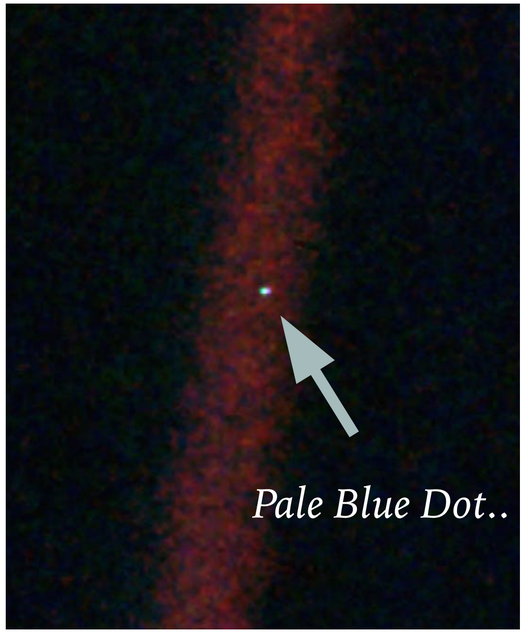
PHYSICS
FROM AN END TO END SYSTEM

Giulio Eulisse (EP-AIP)

CHALLENGE: CONVERT MEGAHERTZ TO PAPERS AND PEOPLE

How do we convert 40+ MHz collision rate...

...for a population of a few 10K physicists across the globe...

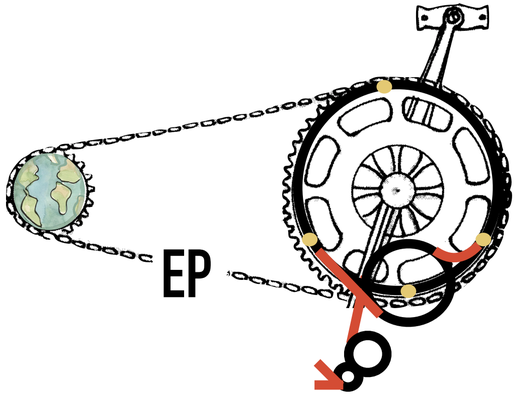


...to expand human knowledge...

WHY R&D FOR THE WHOLE END-TO-END CHAIN?

EP is end-to-end provider

Not only algorithms or plots: data acquisition, hardware / software integration, data & workflow management, software frameworks and toolkits.



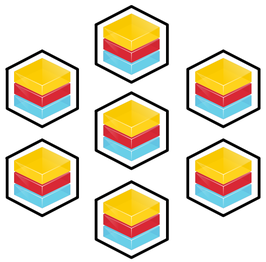
Cooperation

Being part of EP is all about collaboration with others. Worrying about end-to-end means worrying about integration with the rest of the world.

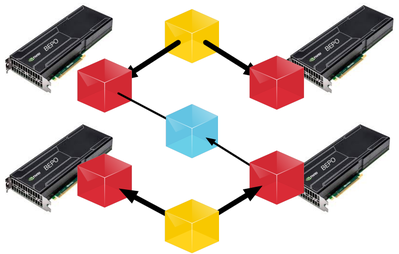
Modular solution(s)

Different design choices imply different trade-offs and might need different solutions. No "silver bullet", but modular ecosystems of interacting products.

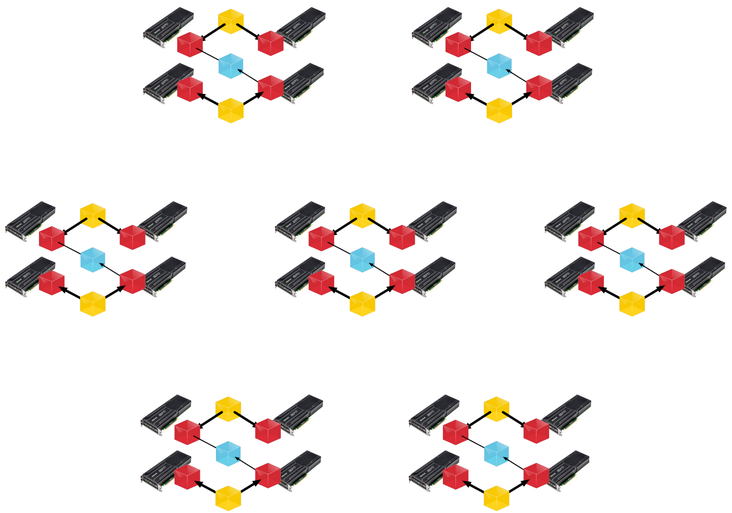
CHALLENGE: TACKLING EVOLVING COMPUTING INFRASTRUCTURE



From homogeneous, standalone resources…



…to heterogeneous datacenters. Blending of traditional Online and Offline roles (e.g. ALICE O2, LHCb)…



…actually a few of them, requiring negotiations with our WLCG partners…

UNKNOWN FUTURE: STAYING AGILE

Common trends among experiments:

- **Heterogeneous systems:** *different hardware depending on performed tasks (e.g.: GPUs, Tensor Units, low-power CPUs, FPGAs).*
- **Analysis facilities:** *few, well connected datacenters with dedicated general purpose clusters with high throughput interconnections between nodes.*
- **HPC-like resources:** *highly interconnected nodes which get most of their FLOPs from GPU-like hardware.*

Something completely different?

- **Opportunistic (commercial) clouds:** *cheaper computational resources, cost shifted to expensive connectivity / storage price. Could provide resources on demand.*
- **Distributed volunteer computing:** *unreliable in the past, can this be fixed by novel algorithms and an adequate business model?*

DATA MANAGEMENT

HEP computing is about data

*HEP is at the **forefront of scalability** needs for data management due to size and world wide collaborations.*
***Future experiments far more challenging** – increase in both data volume and number of objects to be stored.*

Data Lakes & Analysis Facilities

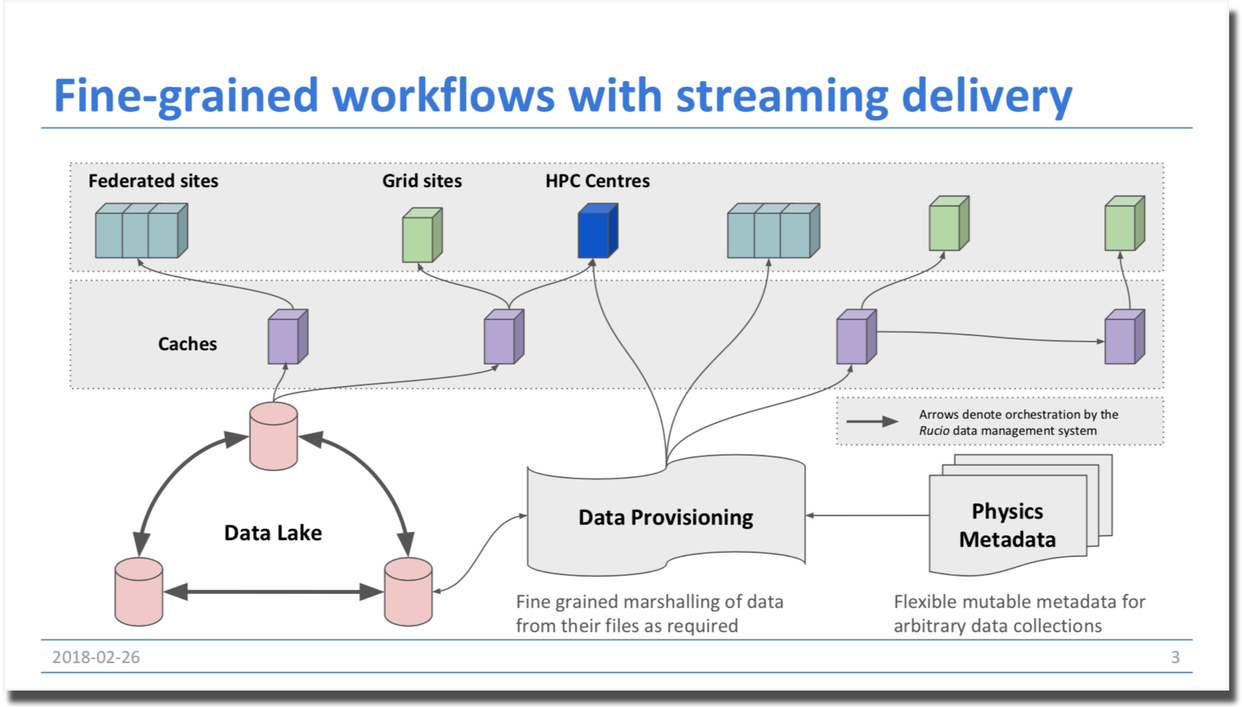
Fewer, well connected sites which act as authoritative source for caching layer seem to be a common trend for future designs.

Rucio

ATLAS solution for data management system should scale to Run3 needs. Looking ahead to Run4.

A collaborative effort

Championed, but not unique, to ATLAS. Other experiments expressing interest in it.



See M.Lassnig talk

ALTERNATIVES TO FILE-BASED ANALYSIS

HEP computing is about going through data, fast

File-based analysis has served us well. However, many hints we will be moving away from the operational sweet spot soon.

Exploring alternatives to scale further

File-less alternatives, like key-value ***object stores****, are a common solution to scale out data processing while* ***keeping system complexity under control****.*

Many applications

Not only event data, but also applicable to calibrations, quality control plots, monitoring.

Bridge technology to cloud ecosystems?

See D.Piparo & ROOT team lighting talk

INTEGRATION AND DEPLOYMENT OF MACHINE LEARNING EFFORTS

Technique of the future?

Machine Learning is a key problem-solving skill for the years to come. Optimised hardware could provide a factor 100x in performance.

Heterogeneous by design

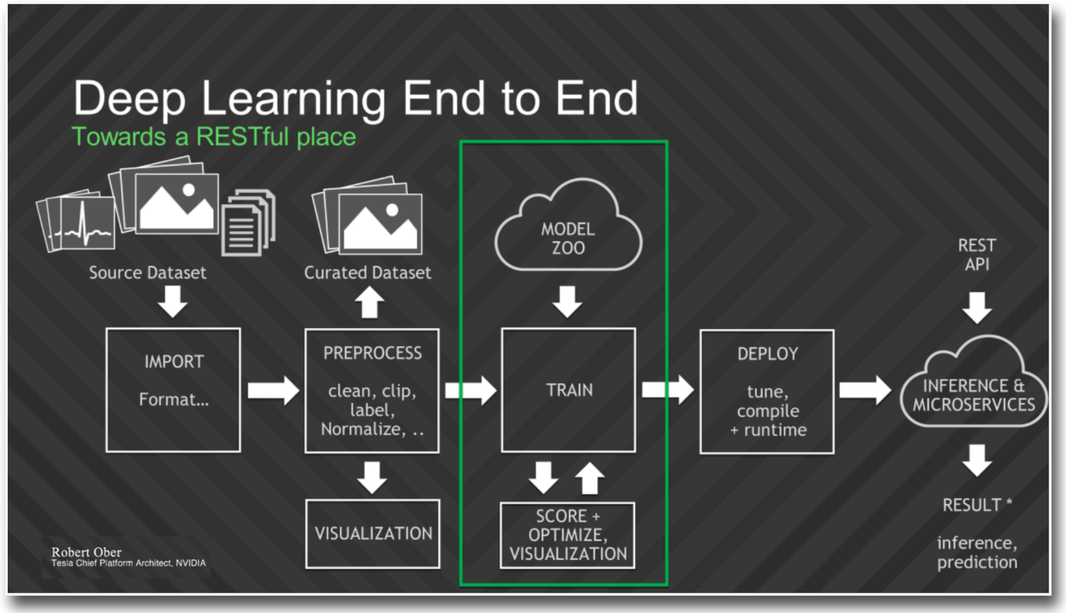
Once again, current ML / DL toolkit play extremely well with GPUs and custom accelerators.

Impedance mismatch

Address integration of our software frameworks with DL models in production. Not only data scientists but also data engineers!

Rapidly moving field

One of the challenges highlighted by previous discussions is that the field is suffering a "precambrian explosion" of tools and techniques.



See V. Innocente talk.

THANK YOU!

Performance optimisation

Data Management

Data Lakes

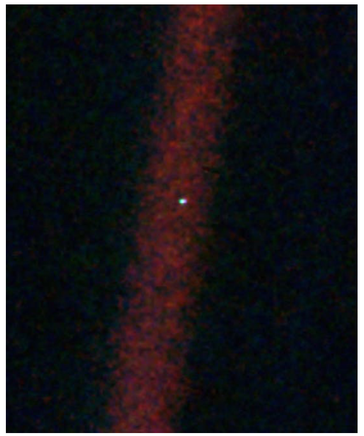
Novel Outreach Solutions

GPUs integration

TPUs

Key-value stores

Novel network fabrics



Heterogeneous infrastructure

Analysis Facilities

ML / DL Integration

Novel collaboration tools

Workload Management

Cloud computing

Improved analysis & simulation toolkits

Backup(s)

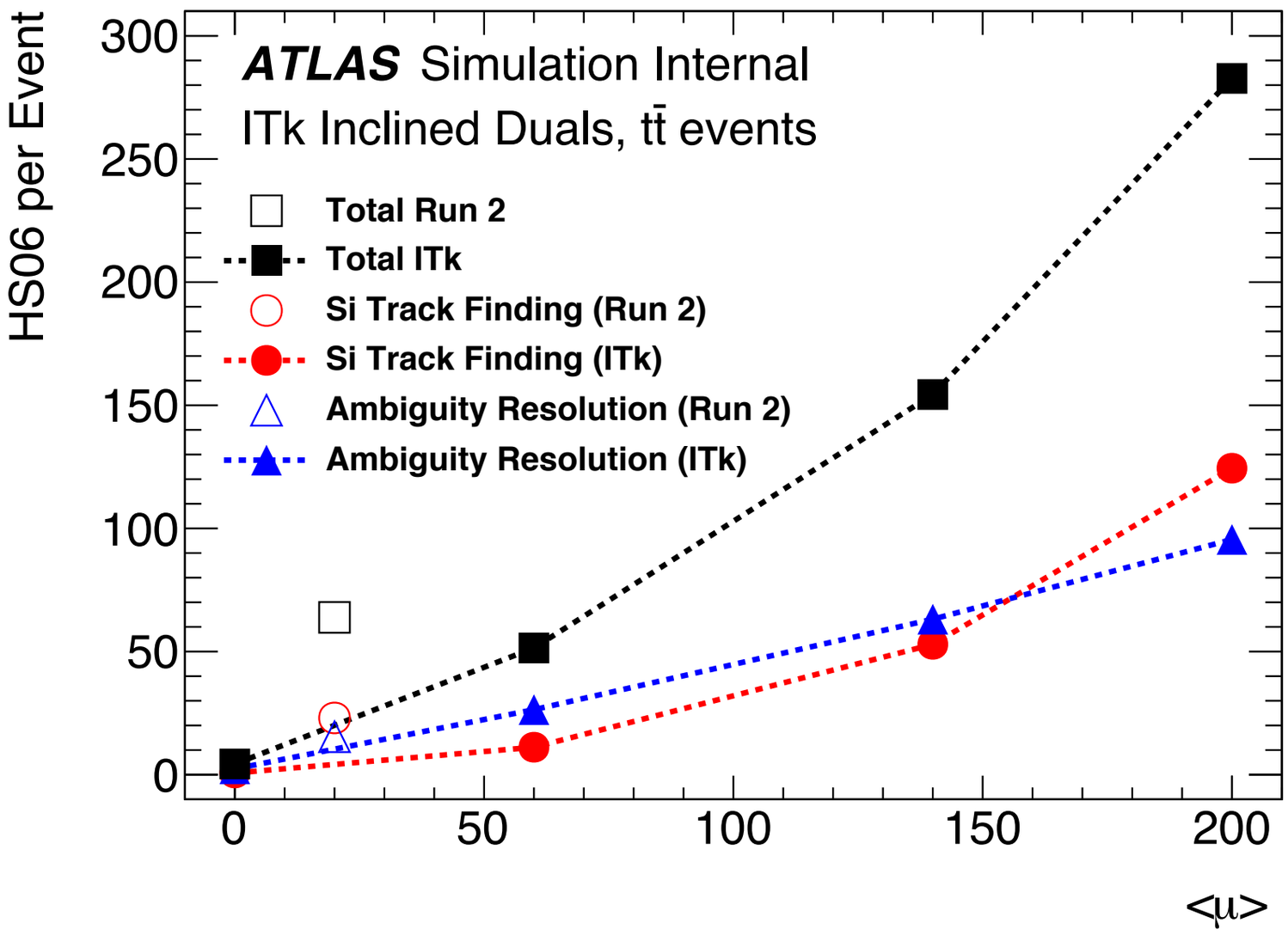
Software Working Group Core Team

Jakob Blomer (Convener)	Danilo Piparo (ROOT, Concurrency)
Graeme Stewart (Convener)	Witek Pokorski (Geant, Generators)
Marco Cattaneo (LHCb)	Radu Popescu (Other languages)
Dirk Duellmann (IT expertise and link)	André Sailer (CLiC, LCD)
Benedikt Hegner (FCC)	Andreas Salzburger (ATLAS, FCC, Tracking)
Mario Lassnig (ATLAS, Data Management)	Niko Neufeld (LHCb, DAQ, FPGAs)
Maurizio Pierini (CMS, Machine Learning)	David Rohr (ALICE, GPUs)
Helge Meinhard (IT R&D)	

HL-LHC CPU extrapolation

Based on ATLAS ITk estimates

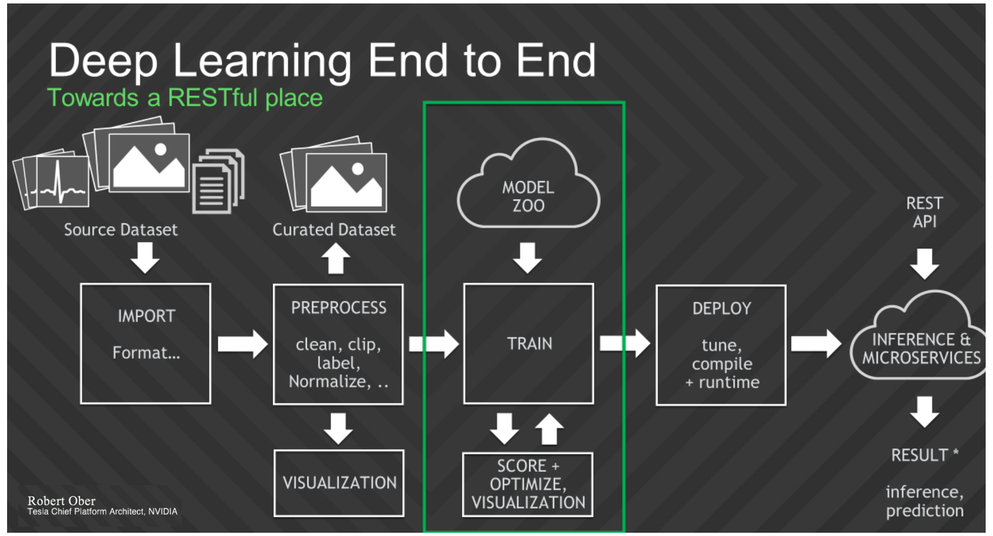
- similar picture for CMS



Inference

Motivation

- Current issues:
  - » Models delivered by scientists way to large (easily >100MB)
  - » Training engines (often used in physics analysis) do not fit production environment (yes, people even invoke Keras from C++)
  - » Default inference engines (Tensorflow C++) do not integrate well with our processing frameworks
    - Manage Threads by themselves for instance
  - » Many new "COTS" optimized inference engines are built to run on cheap or even custom HW
- R&D:
  - » Tuning & compilation of Models
  - » Integration inference engines with processing frameworks
  - » Deployment of heterogeneous solutions in our production environment



Fast Inference

- *Neural network can model non linear functions*
  - *the more complex is the network, the more functions it can approximate*
- *Neural network are faster to evaluate (inference) than typical reco algorithm.*
  - *This is the speed up we need*
- *Neural Networks (unlike other kind of ML algorithms) are very good with raw (non-preprocessed) data (the recorded hits in the event)*

$$(pT, \eta, \phi, E)_{OFFLINE} = f\left((pT, \eta, \phi, E)_{ONLINE} \right)$$

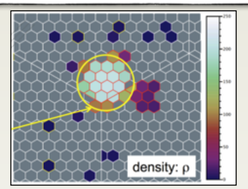
  - *could use them directly on the detector inputs*

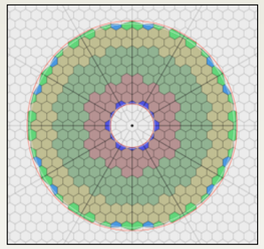
$$(pT, \eta, \phi, E)_{OFFLINE} = g\left(\text{Event hits} \right)$$

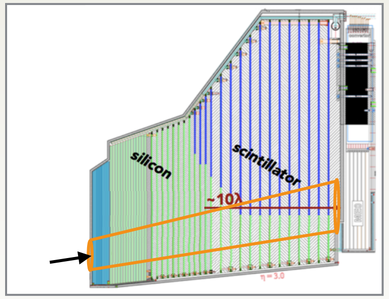
One would have to learn f and g to evaluate them at trigger. Online processing is replaced by offline training

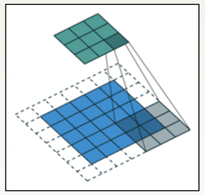
ML & (new) detectors

Map Detector Structure to Neural Network



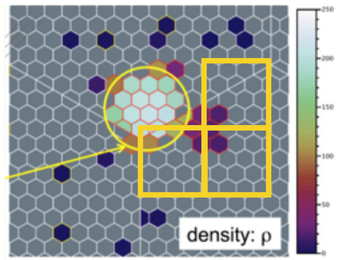


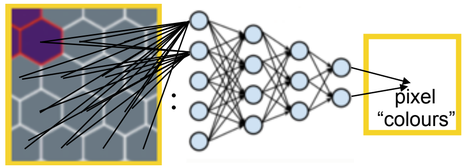




- Sensors hexagonal
- Sensor size/area changes with z,x,y
  - Physics based
- Uniform pixel size in all dimensions

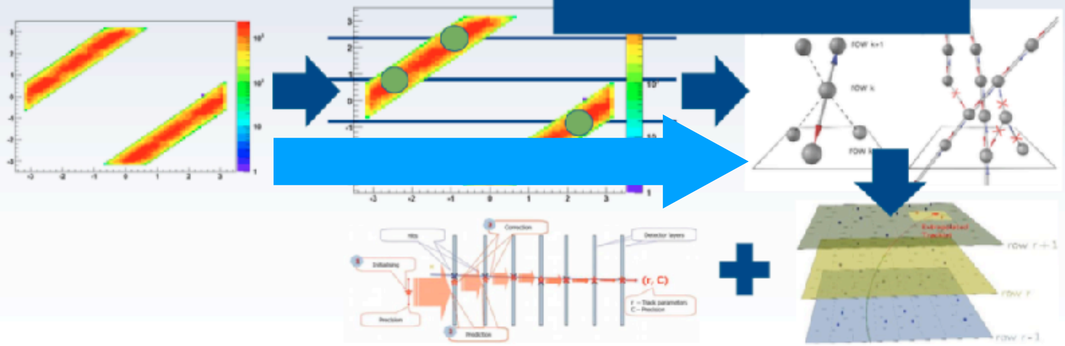
➡Correct representation of the geometry is an issue for **any** non-uniform non-squared sensor design





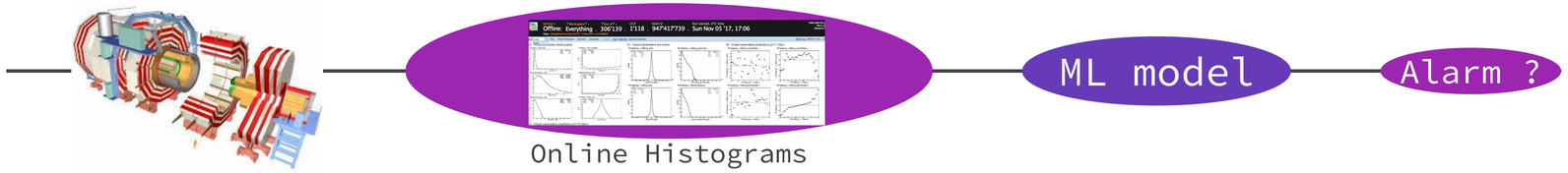
- The normal reconstruction chain is:
  - Clustering, tracking, vertexing, some higher level reco
  - I.e., in the TPC we start with a three-dimensional cha … heatmap is 2D).
  - The detector has some layers (2d planes in the 3d sp
  - We forget about the original charge, and try to conne … GPUs are good at this).
  - Finally, we fit track parameters and extend (track follo … filter).

Why don't we use the charge directly?



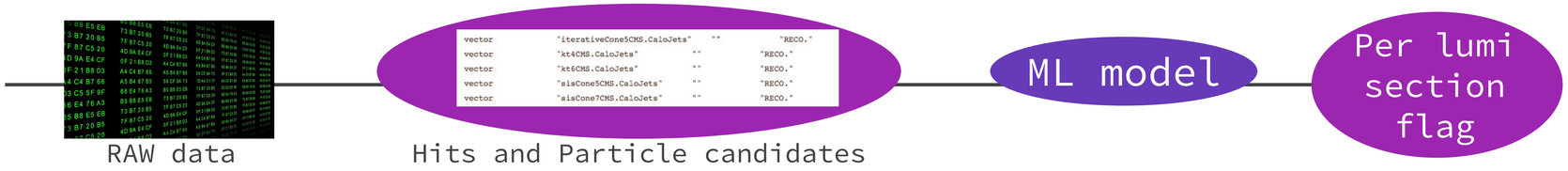
Data Quality

- **Real time feedback** on data quality to the shift crew



- Semi-supervised deep learning models are trained on past histograms to encapsulate the nature of “good data”
- Histograms from live monitoring are evaluated to identify problems with the hardware or the data taking conditions

- **Offline certification** of reconstructed data



- Semi-supervised deep learning models trained on initial runs to encapsulate the nature of “good data”
- Exploit the entire statistical power and full event reconstruction to single out time intervals affected by failures or bad performanve
- Replacing labor intensive run level selection

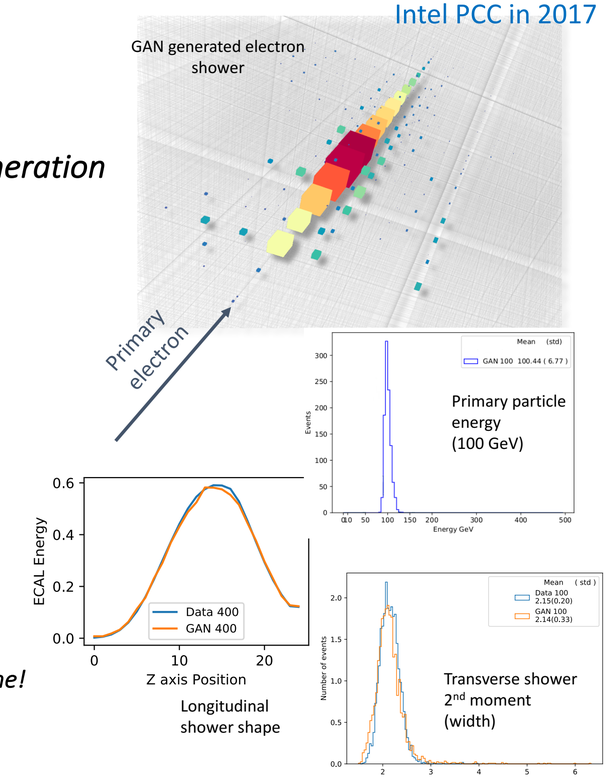
Simulation

Current status

Intel PCC in 2017

First prototype: calorimeter simulation as 3D image generation

- CLIC electromagnetic calorimeter (high granularity)
- 3D Convolutional Generative Adversarial Networks
- Realistic generation of samples
- Detailed physics validation
  - Comparison to full sim
- Relatively simple prototype but very promising approach



Many activities started in experiments along the same research line!

CERN openlab intel

YES ML, BUT HOW?

- Machine learning applied to FASTSIM looks very promising

- What if we go one level beyond and we replace computationally expensive physics models with ML blocks
  - Able to learn complex cross-sections shapes (total, differential)?
  - Able to directly generate the final-state?

→ From "physics-agnostic" to **"physics-aware"** neural networks

Training Physics-aware supervised neural networks[1][2]
- Embed physical-laws underlying the process
- To be used to infer physical quantities (momenta, directions, energies..)
- Both for continuous and discrete processes

[1] "QCD-Aware Recursive Neural Networks for Jet Physics", Kyle Cranmer et Al, https://arxiv.org/abs/1702.00748 - Feb 2017

[2] Physics Informed Deep Learning: Data-driven Solutions of Nonlinear Partial Differential Equations, Maziar Raissi et Al, https://arxiv.org/abs/1711.10561 - Nov 2017

Learning with uncertainties

- *New techniques are developed to deal with the uncertainties of the network training*
  - *Notable examples by people from HEP (https://arxiv.org/pdf/1611.01046.pdf)*

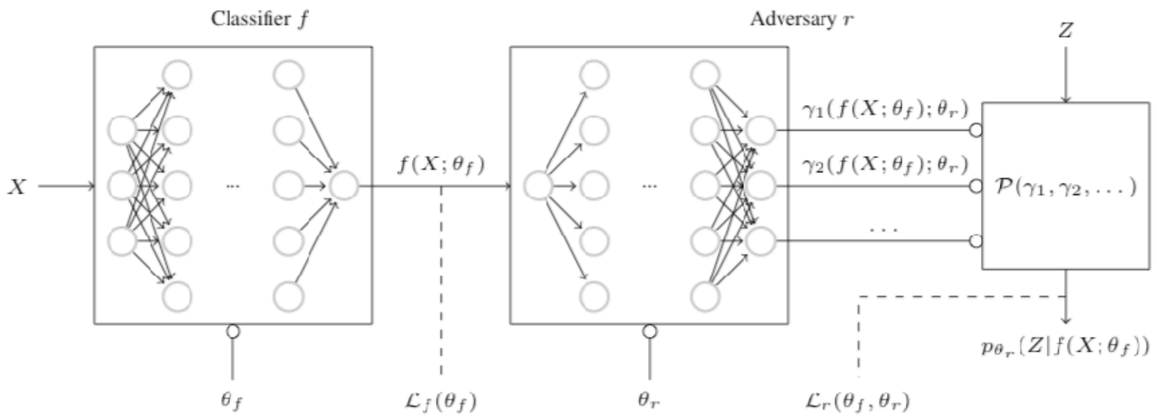


Figure 1: Architecture for the adversarial training of a binary classifier f against a nuisance parameters Z. The adversary r models the distribution $p(z|f(X;\theta_f)=s)$ of the nuisance parameters as observed only through the output $f(X;\theta_f)$ of the classifier. By maximizing the antagonistic objective $\mathcal{L}_r(\theta_f,\theta_r)$, the classifier f forces $p(z|f(X;\theta_f)=s)$ towards the prior $p(z)$, which happens when $f(X;\theta_f)$ is independent of the nuisance parameter Z and therefore pivotal.

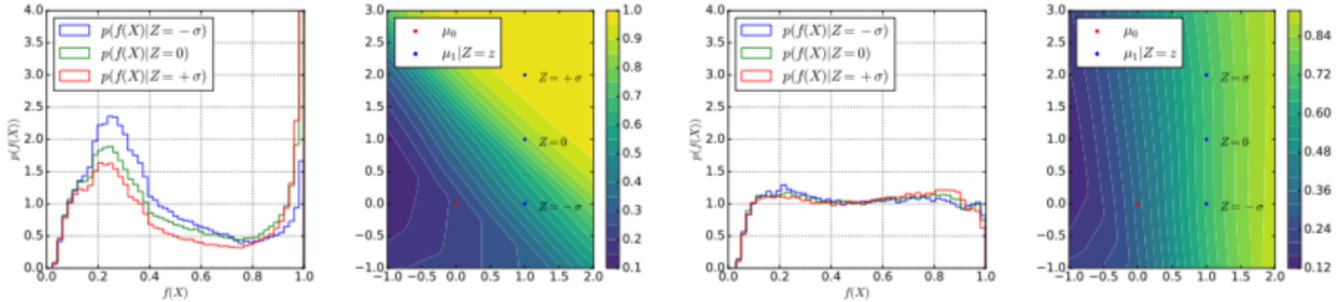


Figure 2: Toy example. (Left) Conditional probability densities of the decision scores at $Z=-\sigma,0,\sigma$ without adversarial training. The resulting densities are dependent on the continuous parameter Z, indicating that f is not pivotal. (Middle left) The associated decision surface, highlighting the fact that samples are easier to classify for values of Z above σ, hence explaining the dependency. (Middle right) Conditional probability densities of the decision scores at $Z=-\sigma,0,\sigma$ when f is built with adversarial training. The resulting densities are now almost identical to each other, indicating only a small dependency on Z. (Right) The associated decision surface, illustrating how adversarial training bends the decision function vertically to erase the dependency on Z.