Computing, Software and Deep Learning in High Energy Physics

NCSR DEMOKRITOS
Feb 2018

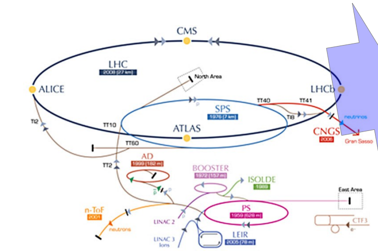
Jean-Roch Vlimant

Outline

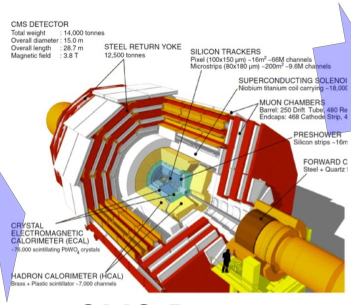
- Overview of the challenge
- Computing **for** and **with** Machine Learning
    - ➔Access to data
    - ➔Access to resource
    - ➔Access to software
- Summary & Conclusions

Overview

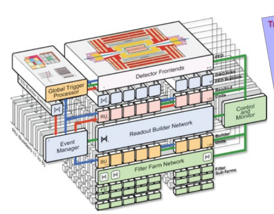
Big Science Pipeline



Large Hadron Collider
40 MHz of collision

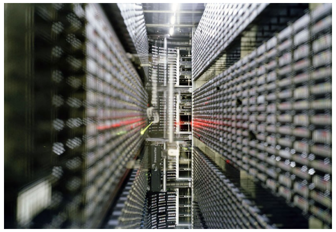


CMS Detector
1PB/s

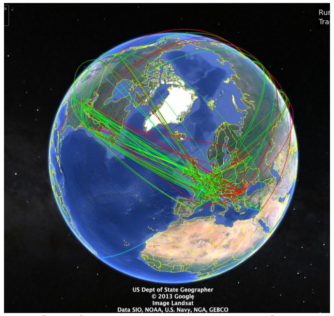


CMS L1 & High-Level Triggers
50k cores, 1kHz

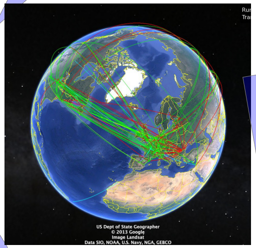
CERN Tier-0
Computing Center
20k cores dedicated



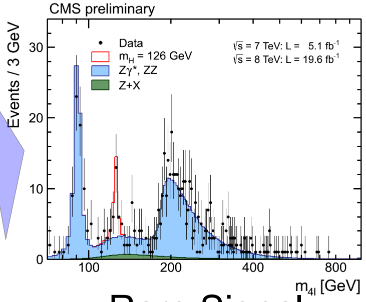
CERN Tier-0/Tier-1
Tape Storage
200PB total



LHC Computing Grid
200k cores pledge to
CMS over ~100 sites

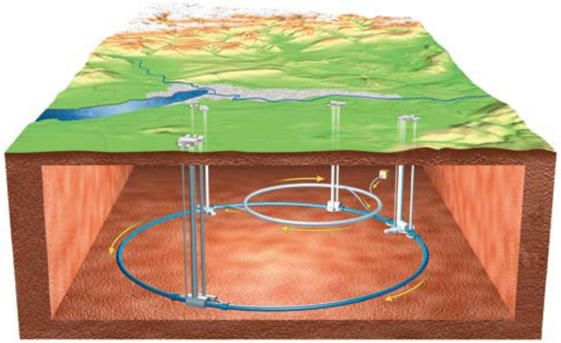


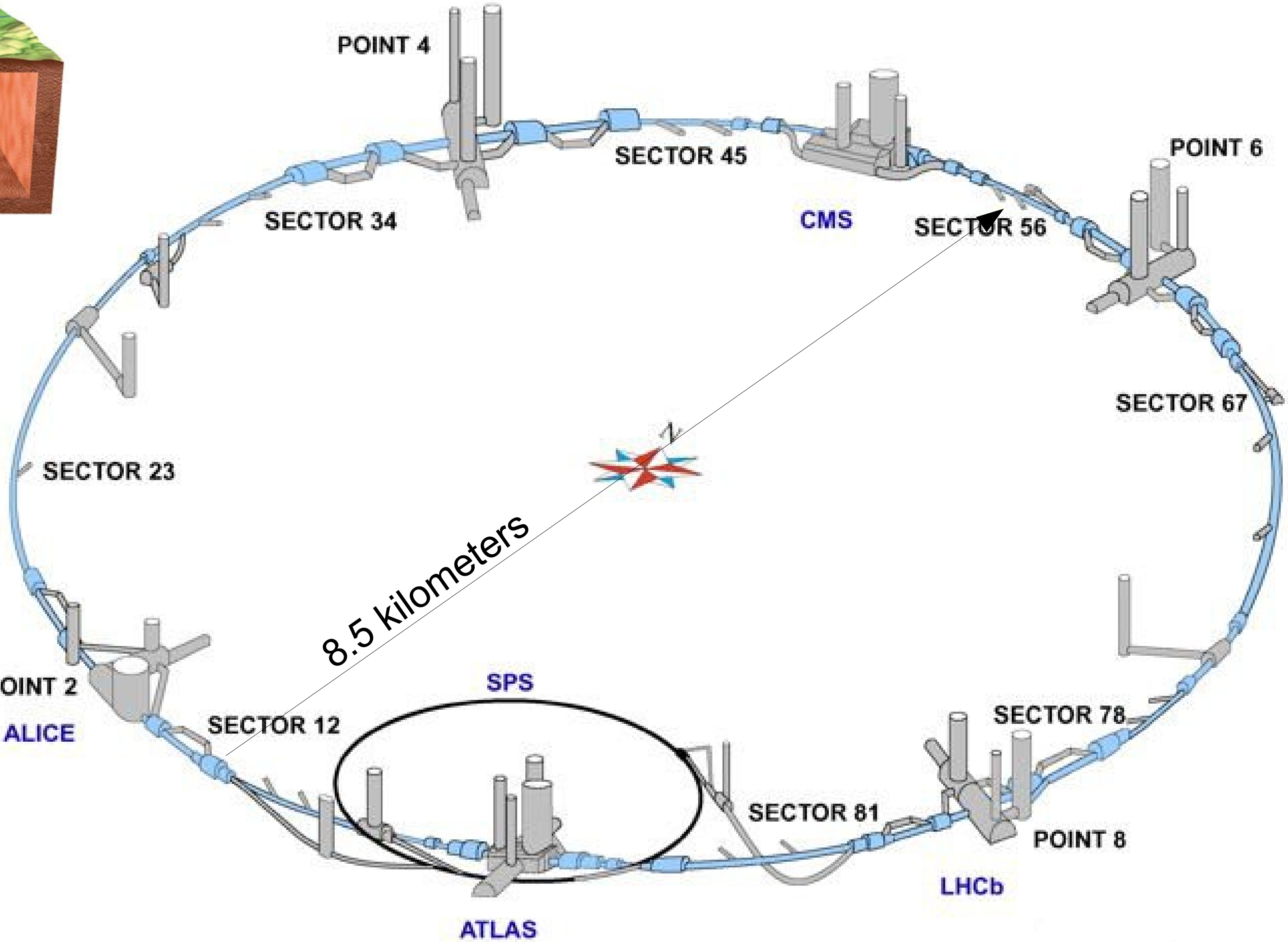
LHC Grid
Remote Access
to 100PB of data



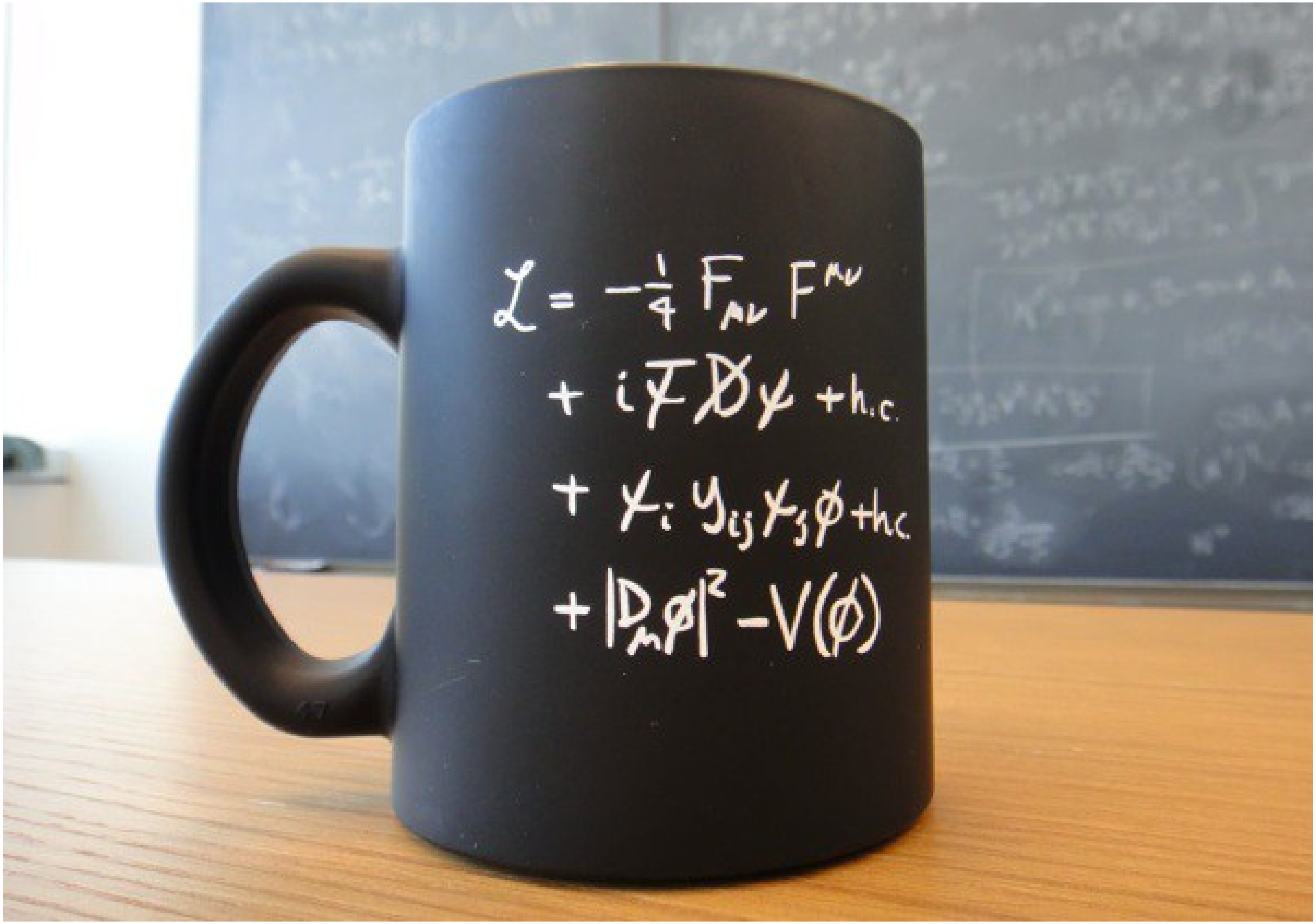
Rare Signal
Measurement
~1 out of 10^6

The Large Hadron Collider



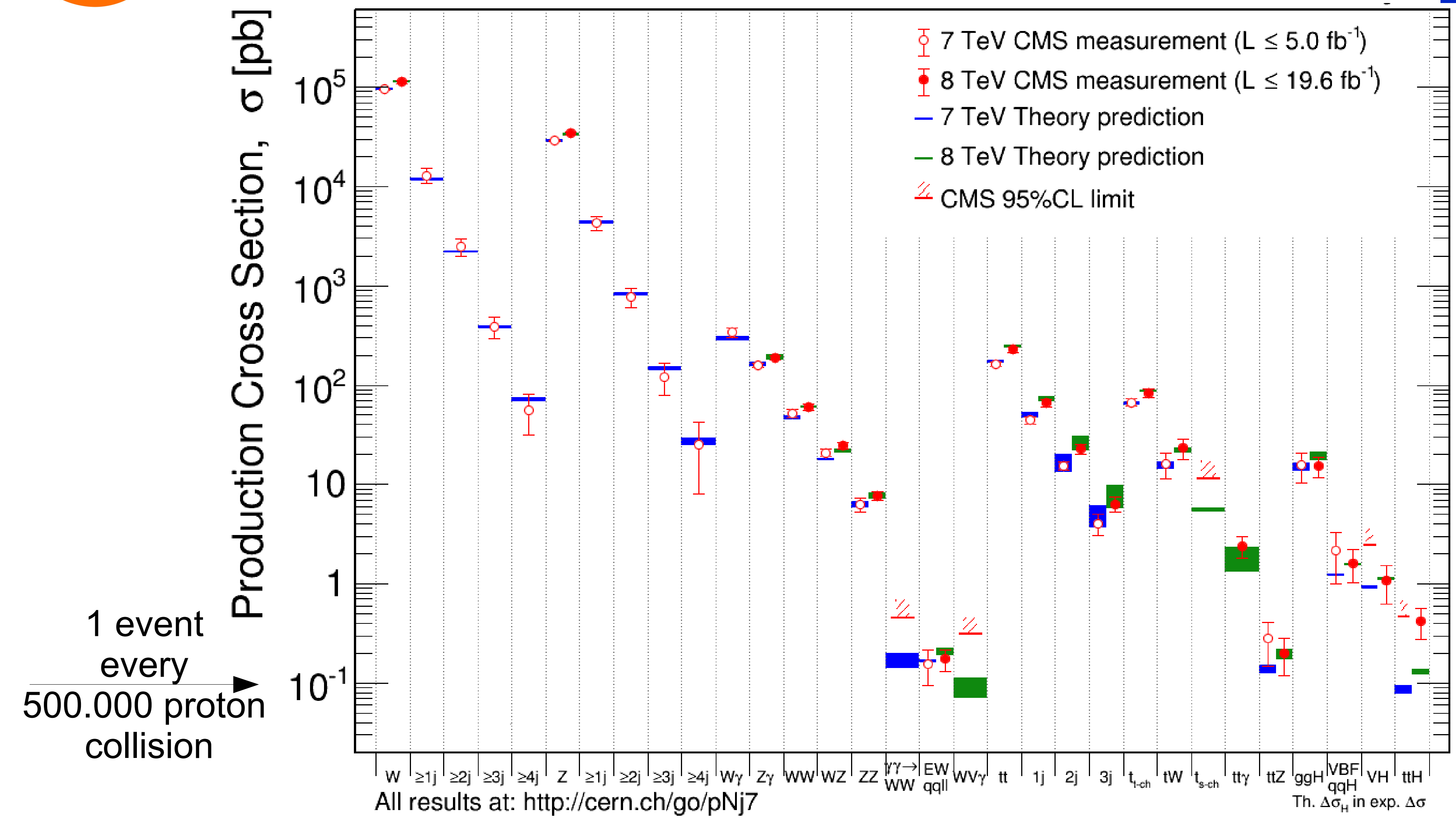


The Standard Model



Well demonstrated effective model
We can predict most of the observations
We can use a large amount of simulation

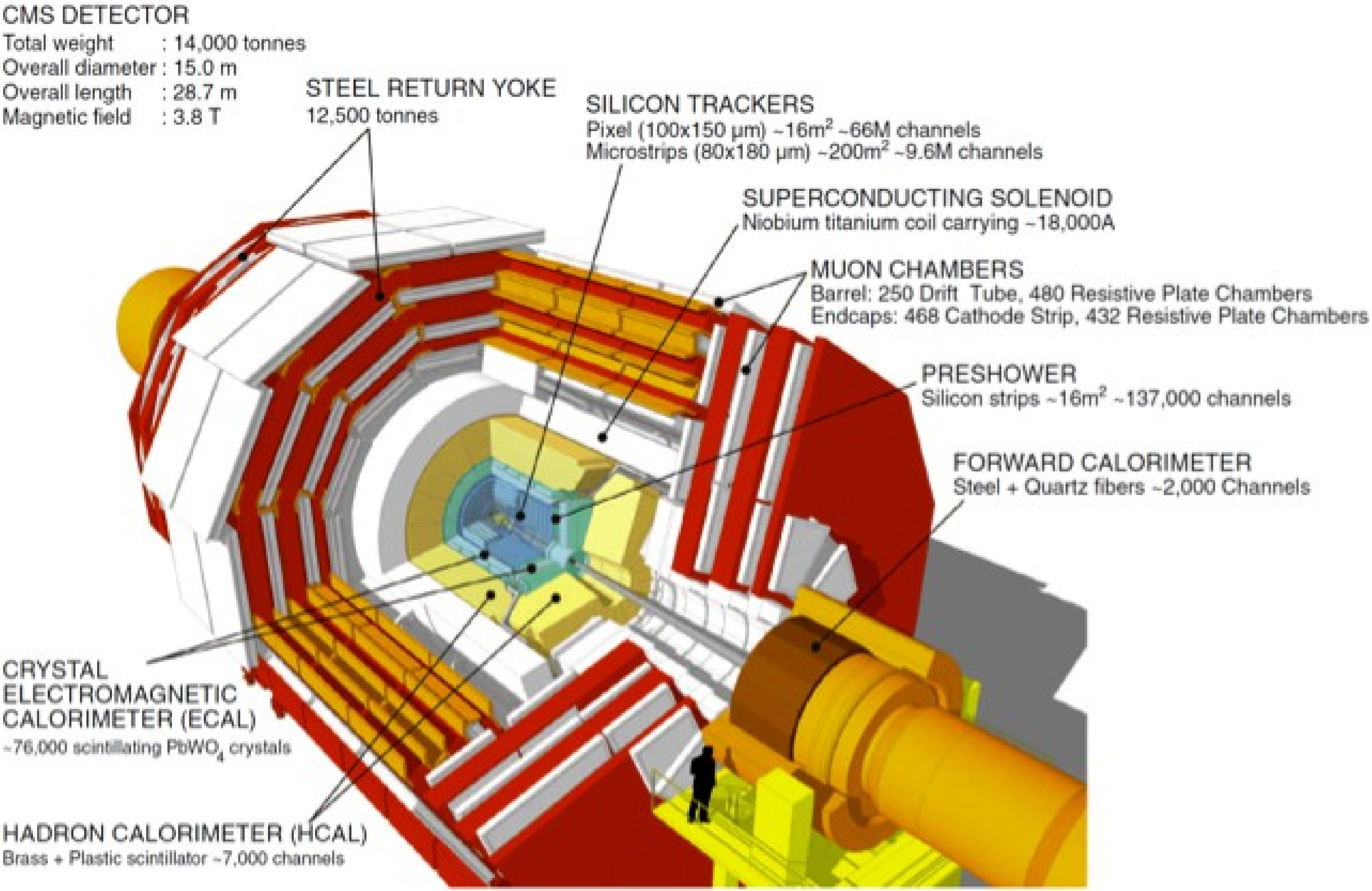
Size Of The Challenge



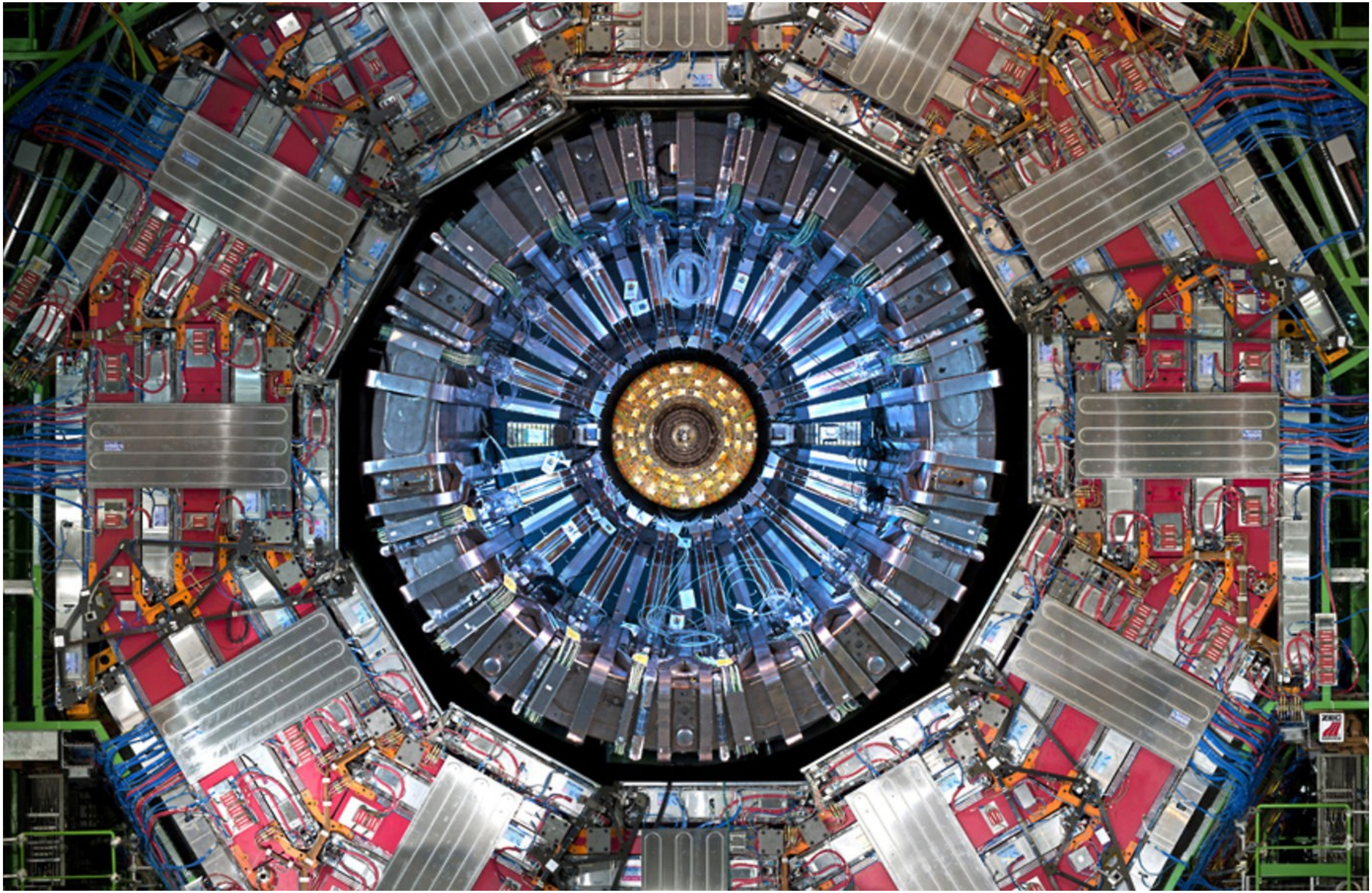
Predictions agree with observation

Need to collect rare events from a large amount of data

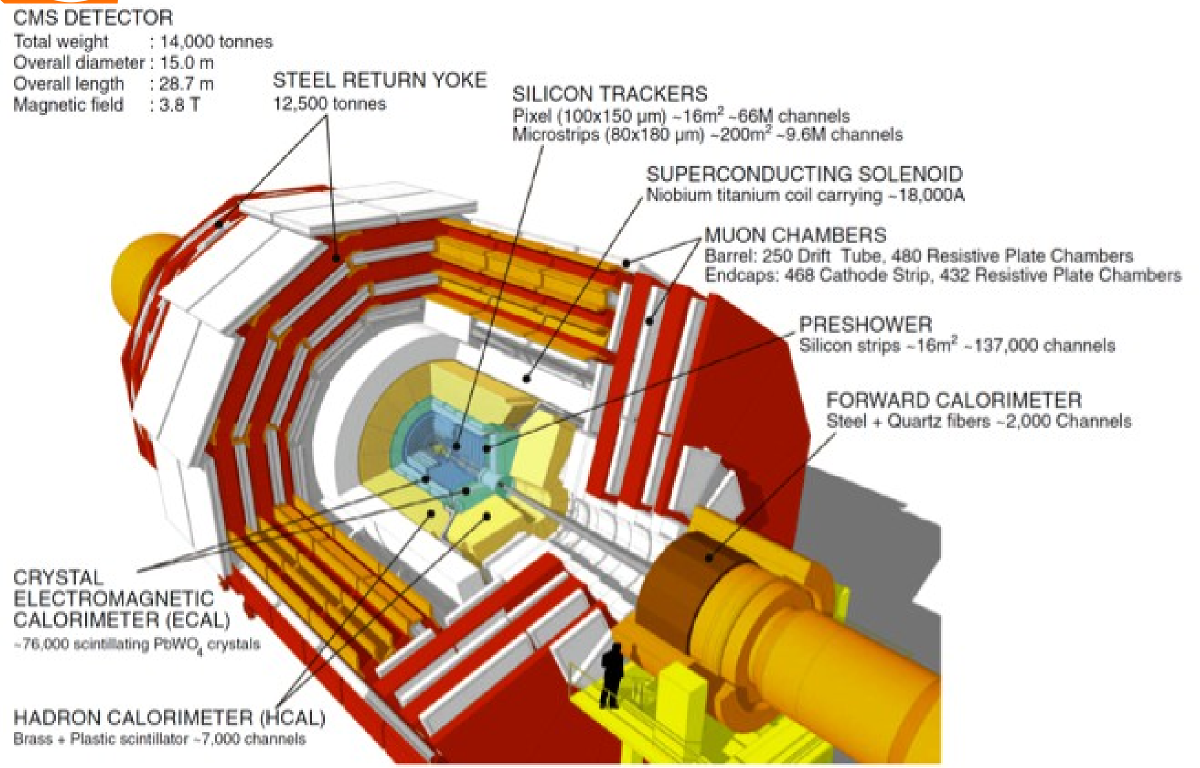
CMS Detector

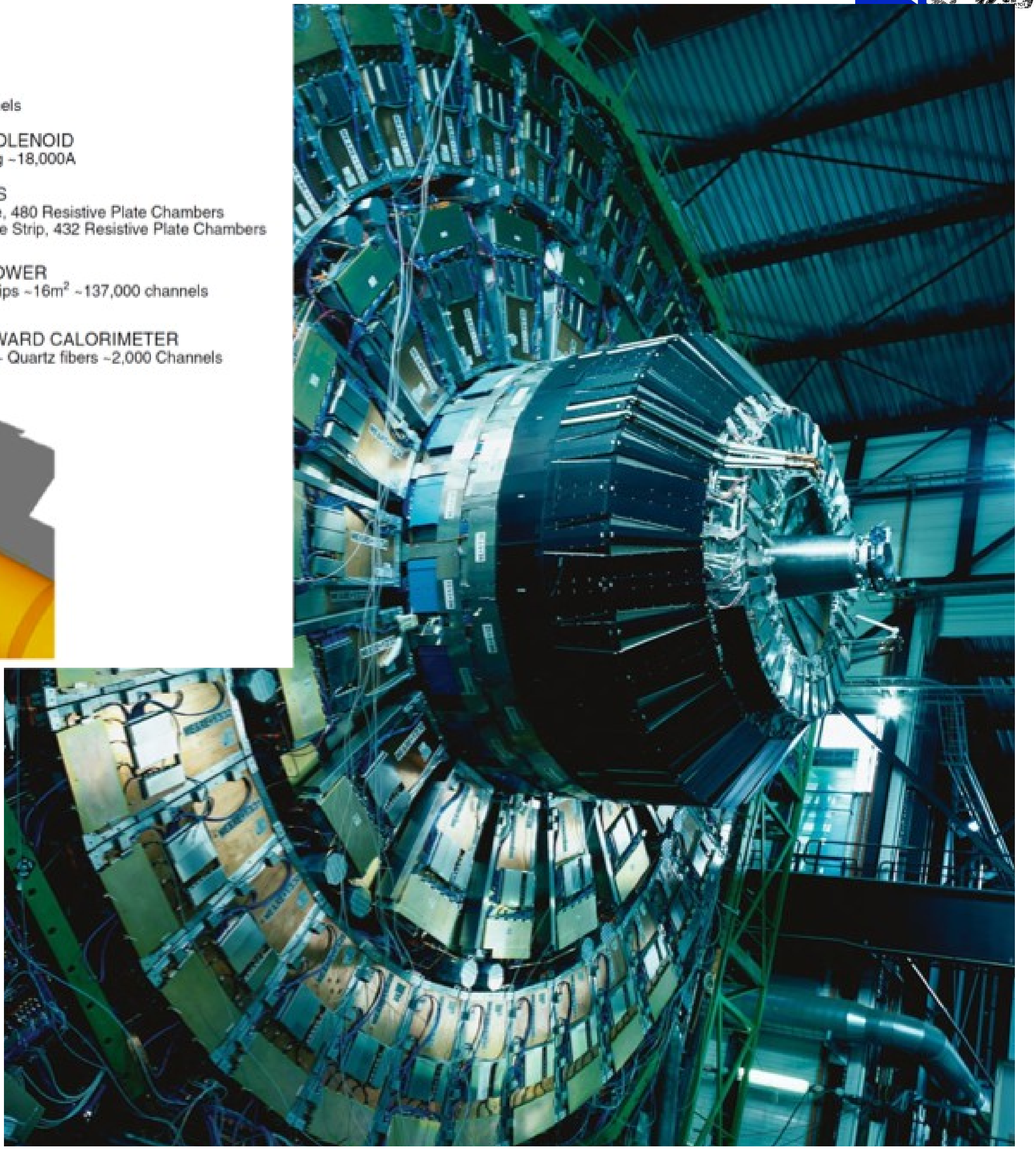


CMS 100 Megapixel Camera



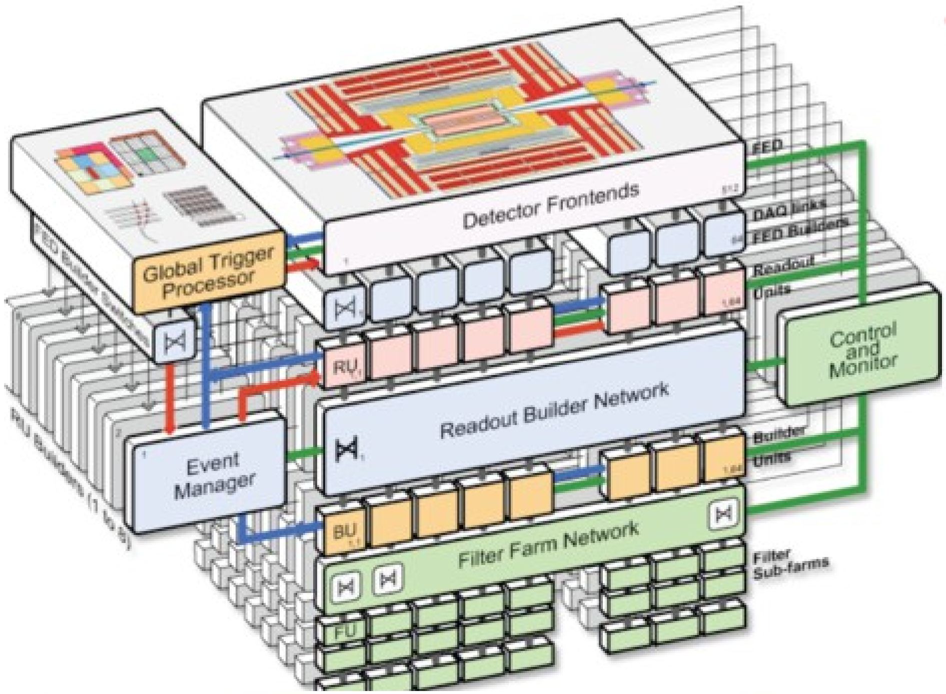
CMS Readout

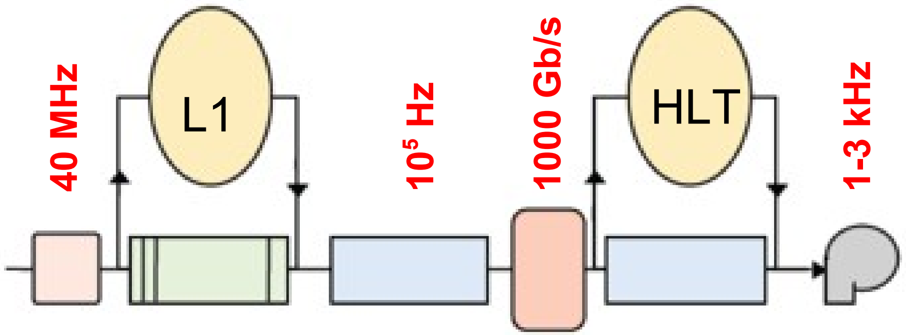


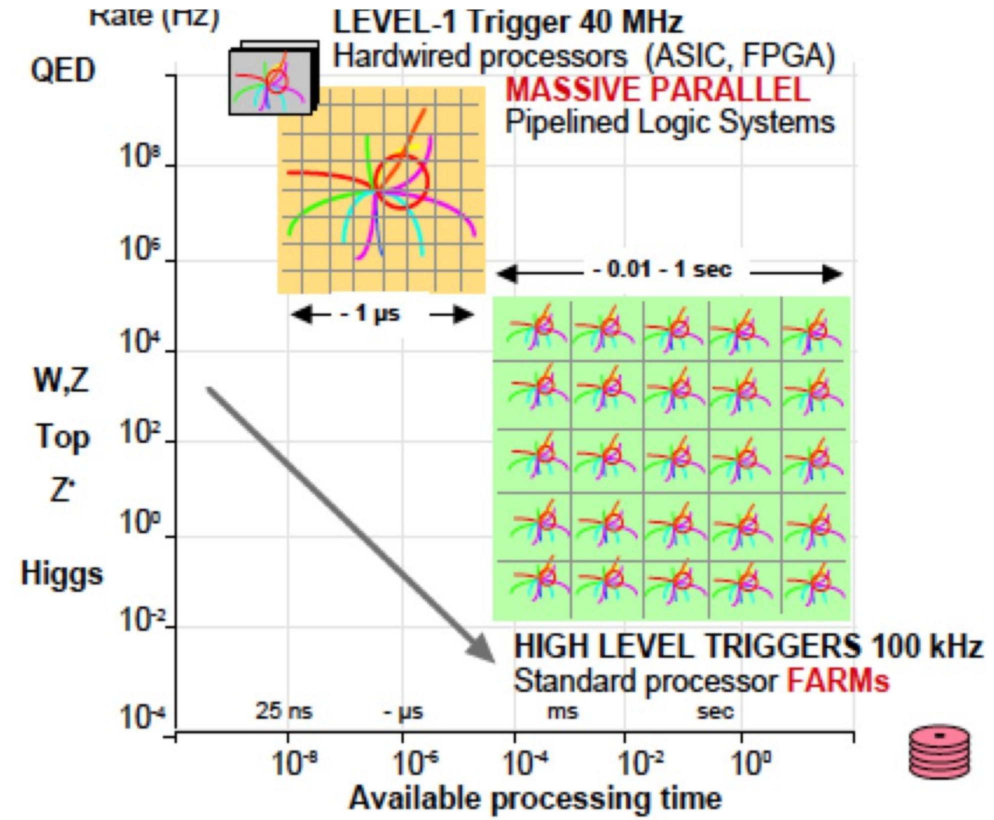


Highly heterogeneous system
Raw data is 100M channels sampled every 25 ns : 1Pb/s
50EB per day in readout and online processing.

Event Filtering

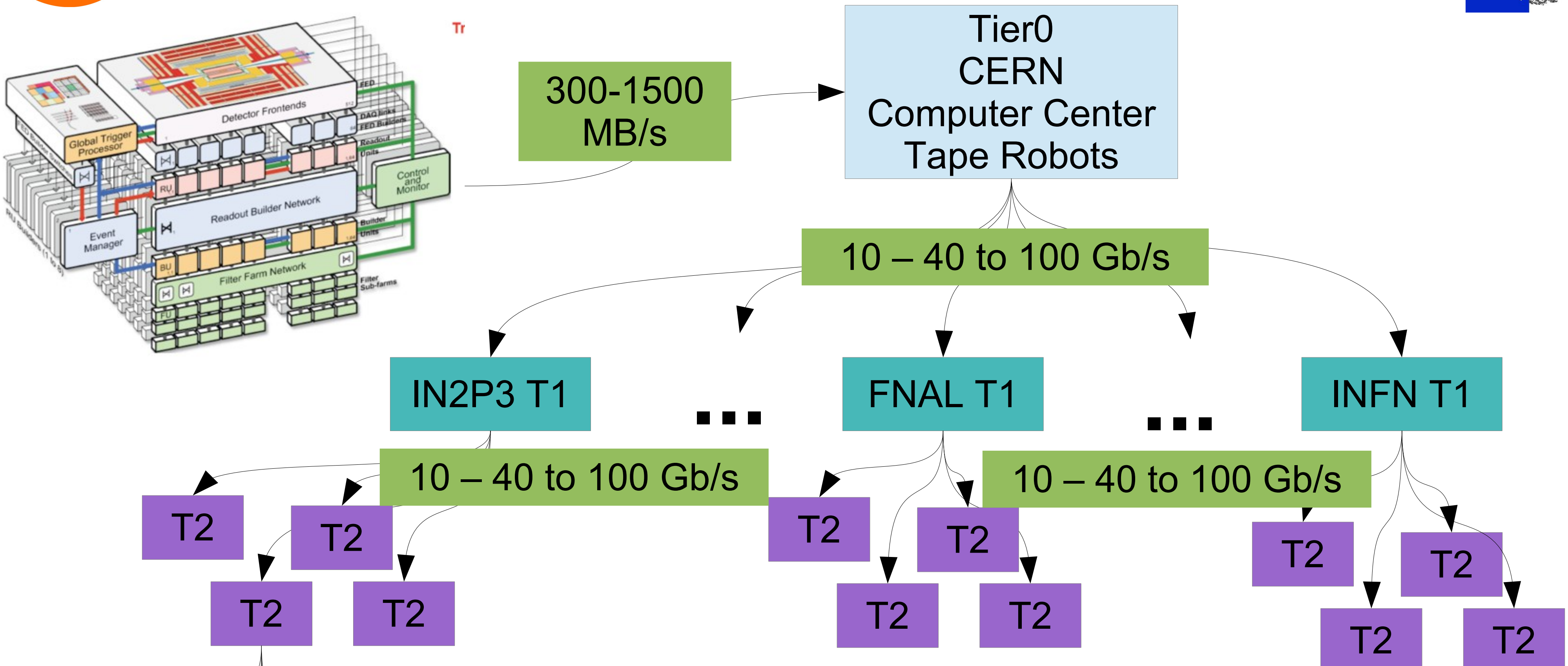


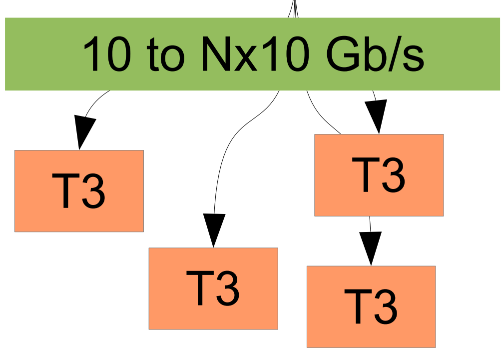




From Big Data to Smart Data
with ultra fast decision

Computing Grid





- **Hundreds of computer centers** (100-10k cores per site)
- Increased use as a cloud resources (any job anywhere)
- Increasing use of additional cloud and HPC resource
- Real time data processing at Tier0
- Data and Simulation production everywhere
- High bandwidth networks between disk storage
- >200k cores pledged world-wide for CMS computing

Why Machine Learning

Why Deep Learning

- LHC Data Processing may **use deep learning methods in many aspects**
- Several class of **LHC Data analysis make use of classifier** for signal versus background discrimination.
  - ✓ Use of BDT on high level features
  - ✓ Increasing use of MLP-like deep neural net

- Deep learning has delivered **super-human performance** at certain class of tasks (computer vision, speech recognition, ...)
  - ✓ Use of convolutional neural net, recurrent topologies, long-short-term-memory cells, ...
- Deep learning has advantage at **training on “raw” data**
  - ➢ Several levels of data distillation in LHC data processing
- Neural net computation is highly parallelizable
  - ➔ Better use of GP-GPU than regular HEP algorithms

- Complex systems to operate, complex signal to analyze
  - ➢ Over-come challenges of data density and volume
  - ➢ Over-come challenges of data and detector complexity
  - ➢ Over-come challenges of ultra-fast decision

Where Deep Learning

- Detector and apparatus control
- Computing GRID, Center & network control
- Operation anomaly detection
- Fast triggering on object or full events
- Fast approximate algorithms
- Rare event detection
- Automated data certification
- Faster simulation software
- Finer even selection
- Better object identification
- More precise event reconstruction
- More robust measurements
- ...

Computing with and for ML

For

Computing resources need to be set to enable machine learning and in particular deep learning

- Heavy training
- Inference in commodity hardware
- Fast inference on dedicated hardware

With

The LHC computing system is a very complex one, with an even bigger data challenge in horizon 2025

- ML to optimize data placement
- ML to model the system
- RL to take control of the system

These slides are mixture of both, organized in the three topics : **data**, **resource** and **software**.

Access to Data

Data Access Patterns

- Analysis access pattern
  - Many users accessing the same set of datasets
  - Balance of cpu-bound and I/O-bound
  - Can mostly afford a one-off read from remote

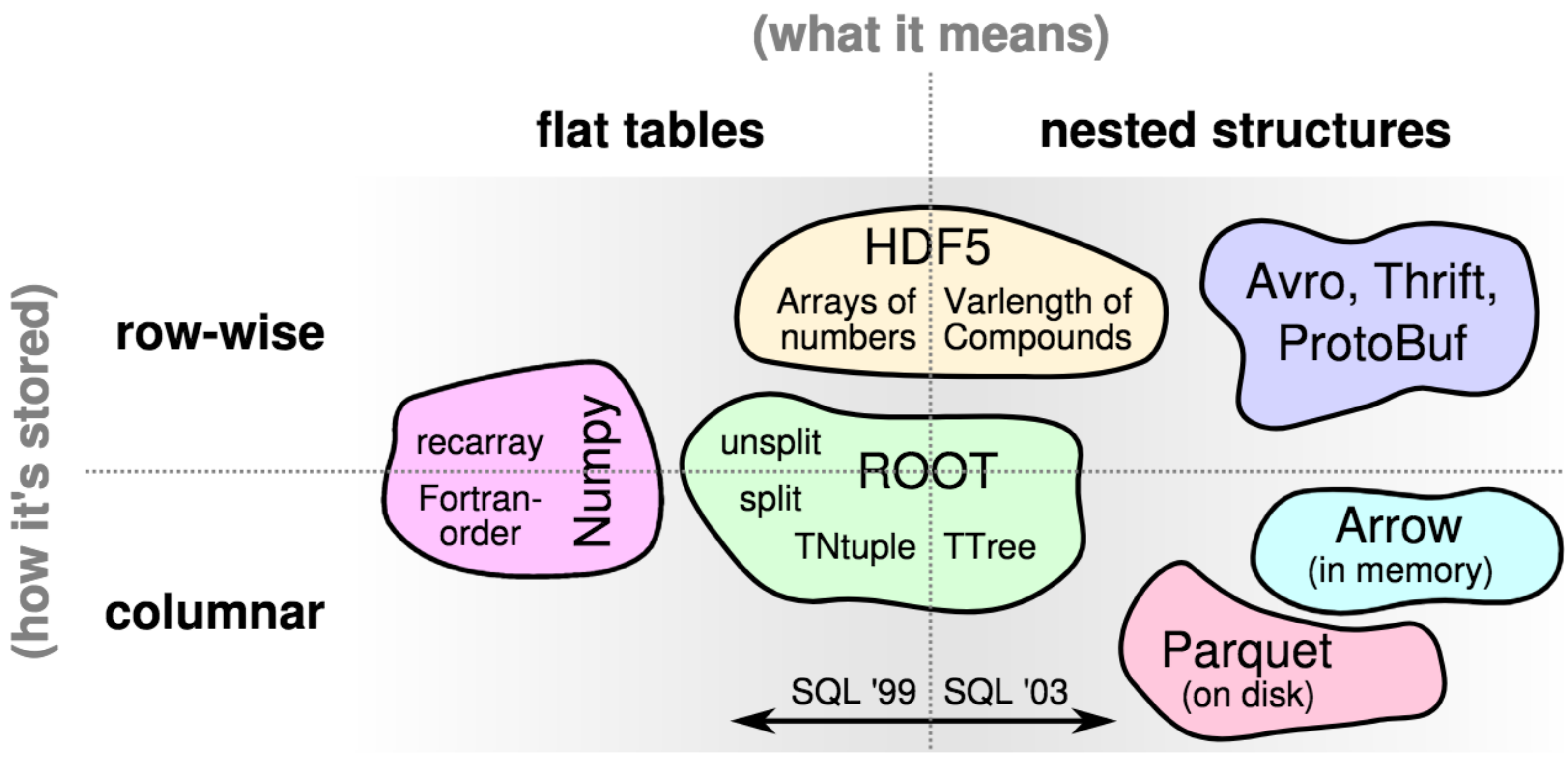
- Model Training access pattern
  - A dataset used for training over many epochs
  - Full dataset read multiple times over in one pass
  - Needs a local storage copy, even a node copy

➔ Some of the grid computing paradigms do not support this natively

Data Formats

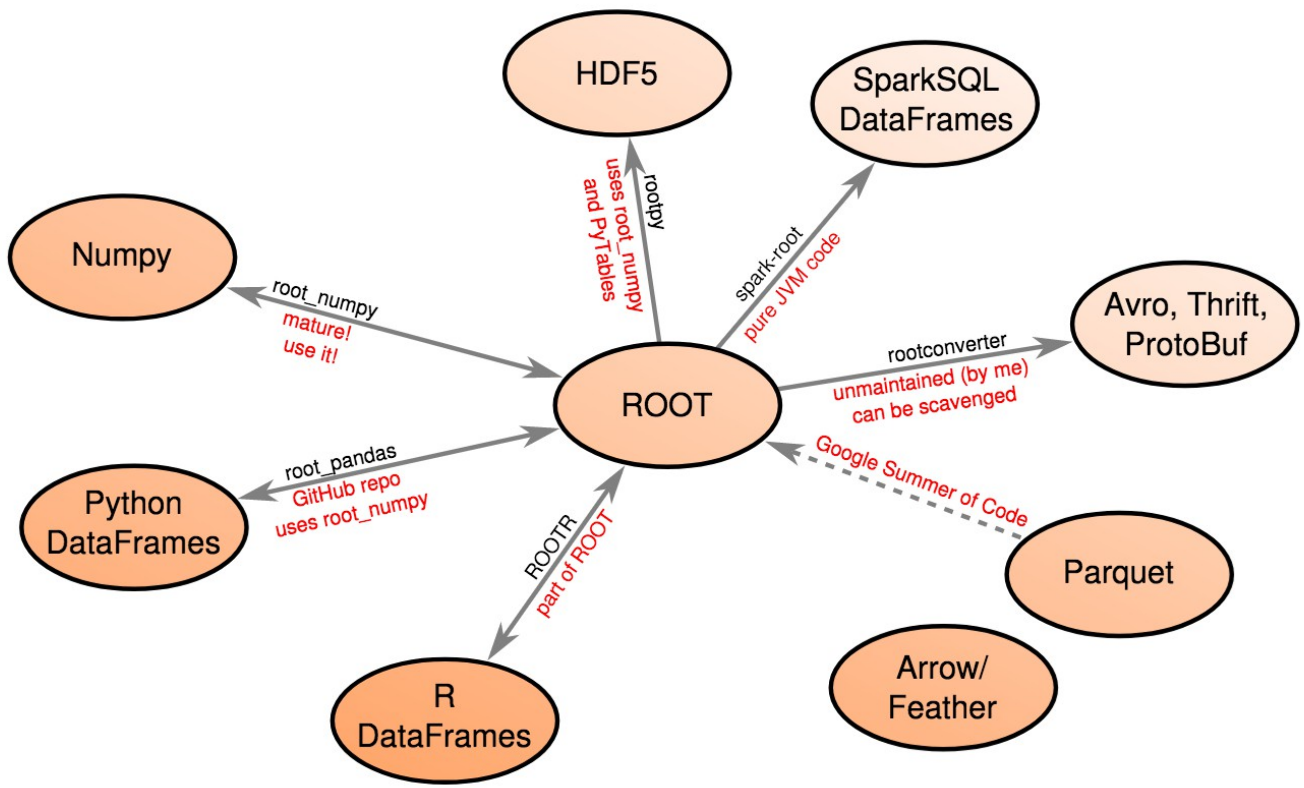
The landscape of generic containers

By "generic," I mean file formats that define general structures that we can specialize for particular kinds of data, like XML and JSON, but we're interested in binary formats with schemas for efficient numerical storage.



Pivarski at https://indico.cern.ch/event/613842/

Data Bridges



Other big-data approaches possible ?
GPU-accelerated sqlite-database, fast indexing, hyper-compression

Data Placement

Problem statement

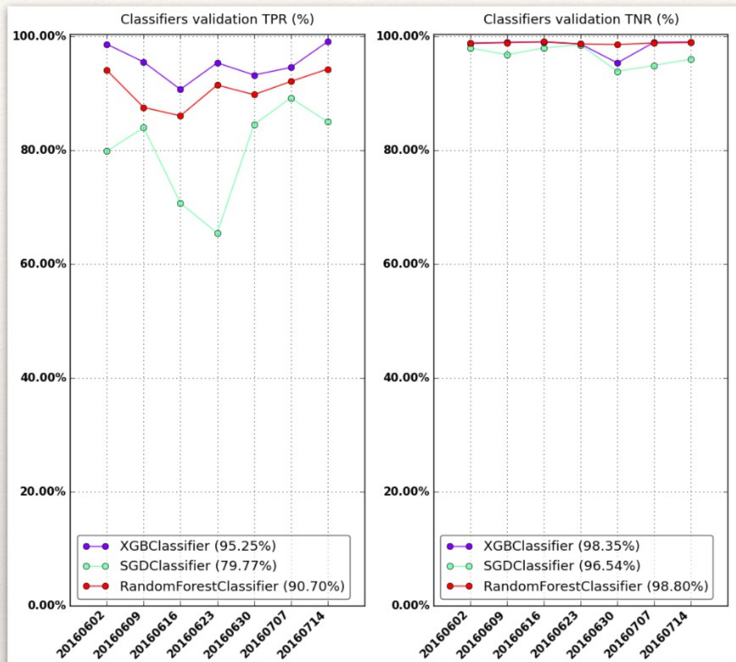
- 100PB of storage non uniformly distributed over 100 sites
- 10s of thousand of analysis datasets or variable importance
- Analysis software not only I/O bound

➔ Locate the dataset on disks over sites so that analyzers can ran on them in short turnaround time.

- The current solution is to **measure** popularity of samples, **replicate** accordingly, and **load balance** disk occupation.
- Can we actually **predict the relevance** of samples based on current utilization's trend ?

Data Popularity

Dataset popularity predictions



Performed studies with various classifiers:

RandomForest, SGD, XGboost

Found similar results with

SparkML

TPR=TP/(TP+FN)
TNR=TN/(TN+FP)

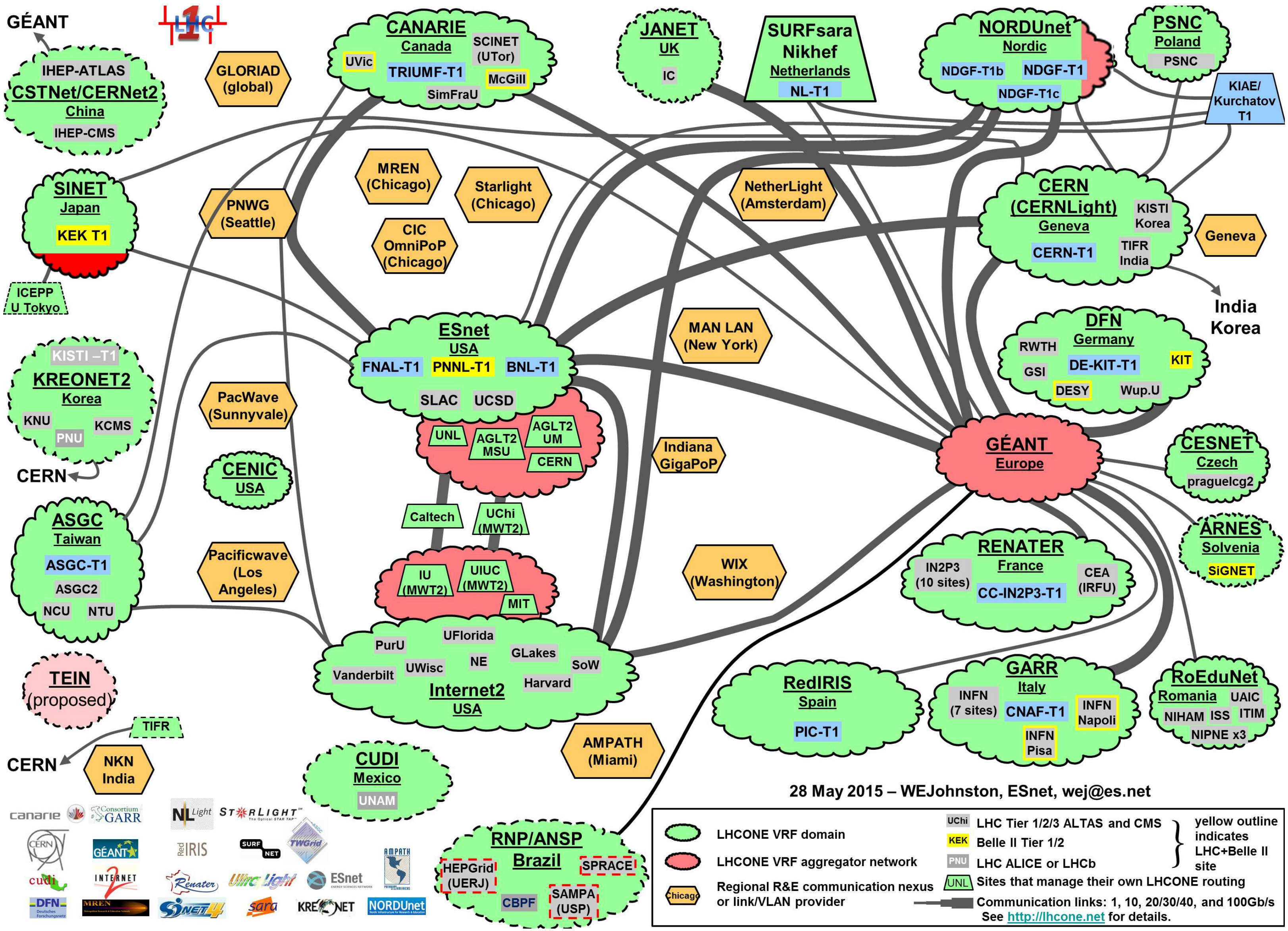
MINIAOD were introduced

in mid 2014

Data tier	TPR	TNR	FP	FN
AOD	0.97±0.05	0.99±0.02	0.005±0.011	0.015±0.029
AODSIM	0.93±0.13	0.99±0.02	0.008±0.016	0.021±0.045
MINIAOD	0.11±0.32	0.99±0.02	0.014±0.026	0.001±0.007
MINIAODSIM	0.49±0.48	0.99±0.02	0.009±0.016	0.007±0.031
USER	0.93±0.15	0.98±0.02	0.014±0.021	0.011±0.023

Kuznetsov, Bonacorsi
https://arxiv.org/abs/1602.07226

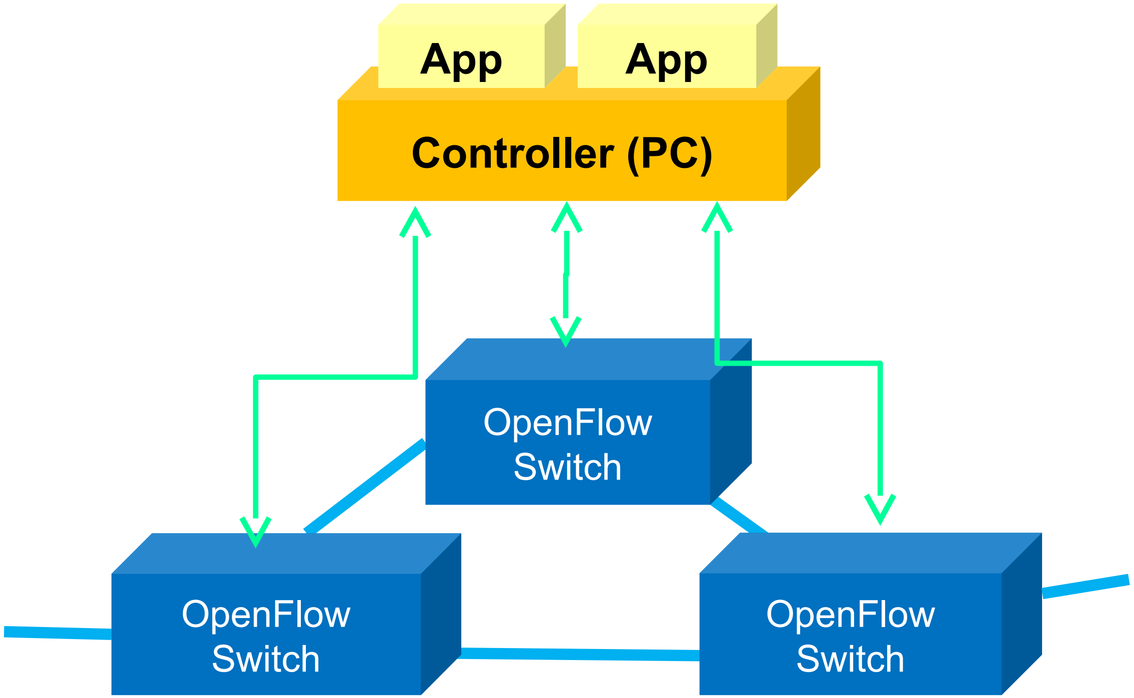
LHC Networking



HEP is a privilege customer of networks

Need to use it efficiently to remain this way

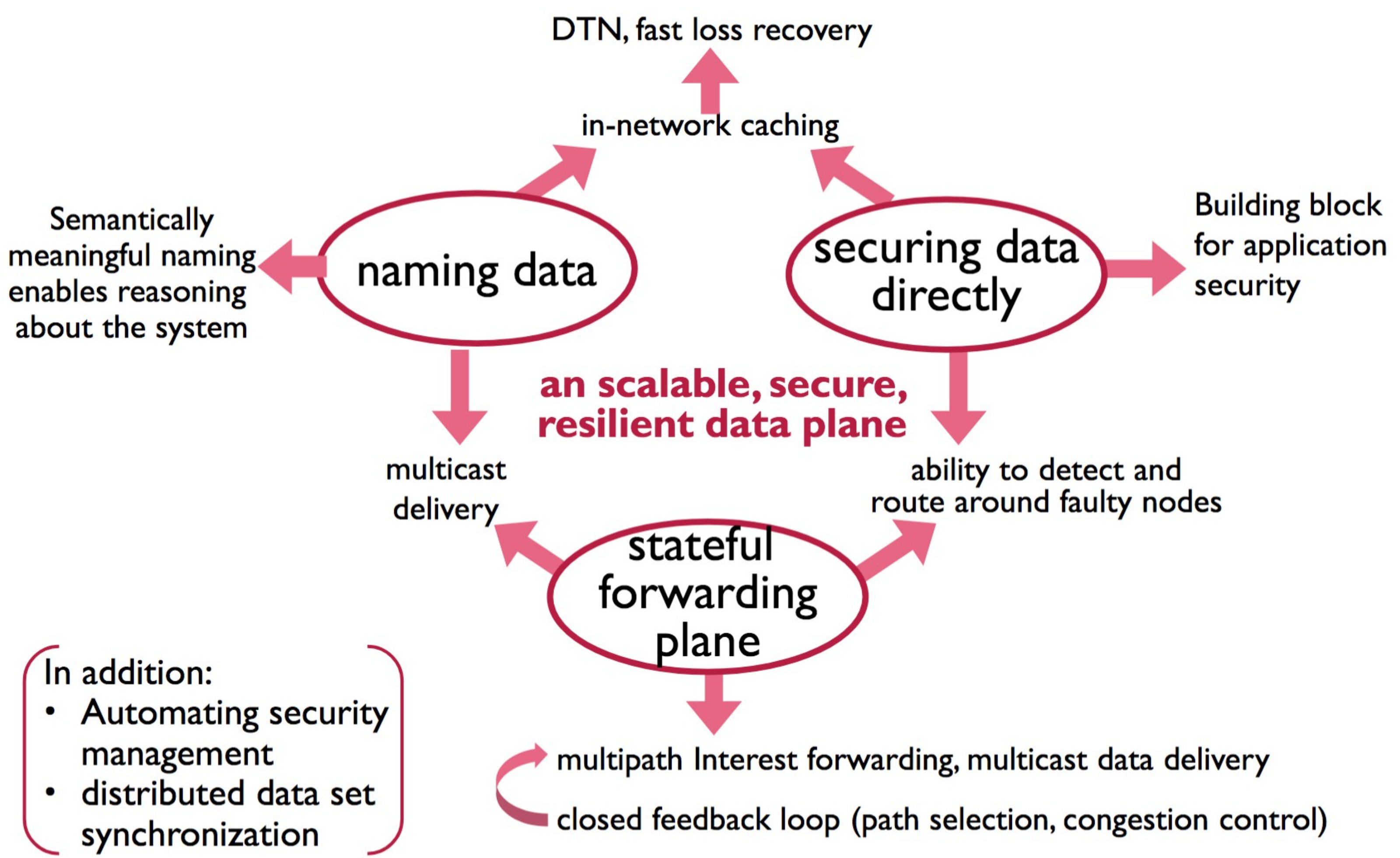
Software Defined Network



New paradigm adopted by research and education network as well as industry
Enables network control by applications
Programmatically define the network functions
Increase use of machine learning techniques
OpenFlow is a standard protocol between controller and network devices

Caching Network

A quick summary of NDN: 3 simple ideas



Lixia Zhang (UCLA)

Access to Software

Software in HEP

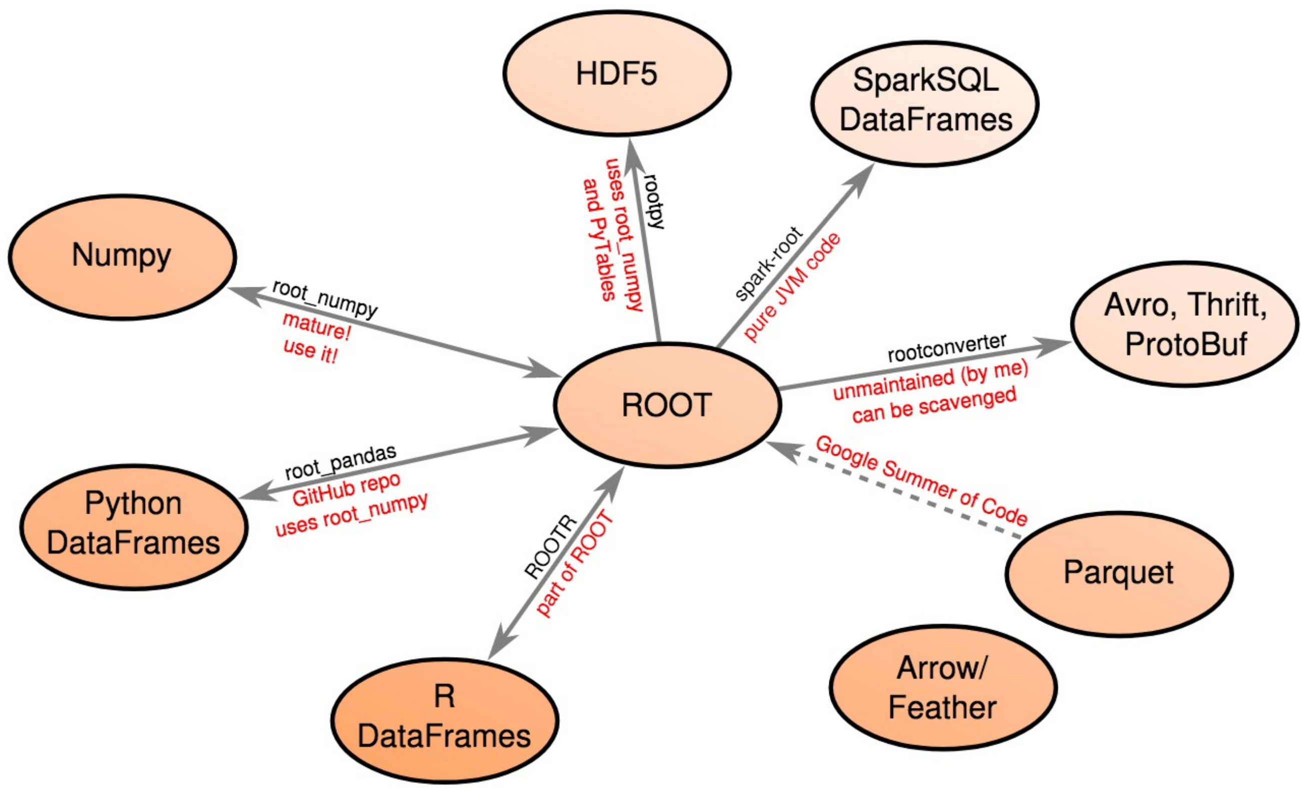
- ROOT-TMVA has made a lot of changes in recent years
  - ➔ Bridge to python, scikit-learn, R, ...
  - ➔ Implementation of deep learning
  - ➔ GPU support, mpi, ...

- Software is commonly distributed over cvmfs (CERN virtual machine file system)
  - ➔ Need only to install cvmfs
  - ✗ Not always the bleeding edge versions

- Many solutions for bridges
  - ➔ Docker, shifter, singularity, ...

- Bleeding edge methods are in pytorch, tensorflow, keras, ... how can both ends meet ?

Software Bridges



The data conversion throw bridges between root and industry software

On-Time Inference

Problem Statement

- Experiments are running within their C++ framework, running over the WLCG on comodity hardware
- Most training libraries are based on python, using GP-GPU

➔ How to run inference efficiently of the trained models

Current Solutions

- C++ implementation of most operators
  - ➔ https://github.com/riga/tfdeploy https://gitlab.cern.ch/mrieger/CMSSW-DNN
  - ➔ https://github.com/lwtnn/lwtnn
  - ✔ Work for most tensorflow and keras models
- Tensorflow C++ backend in CMSSW

Can this be integrated better with the experiment framework ?

Access to Resources

Working with Notebook

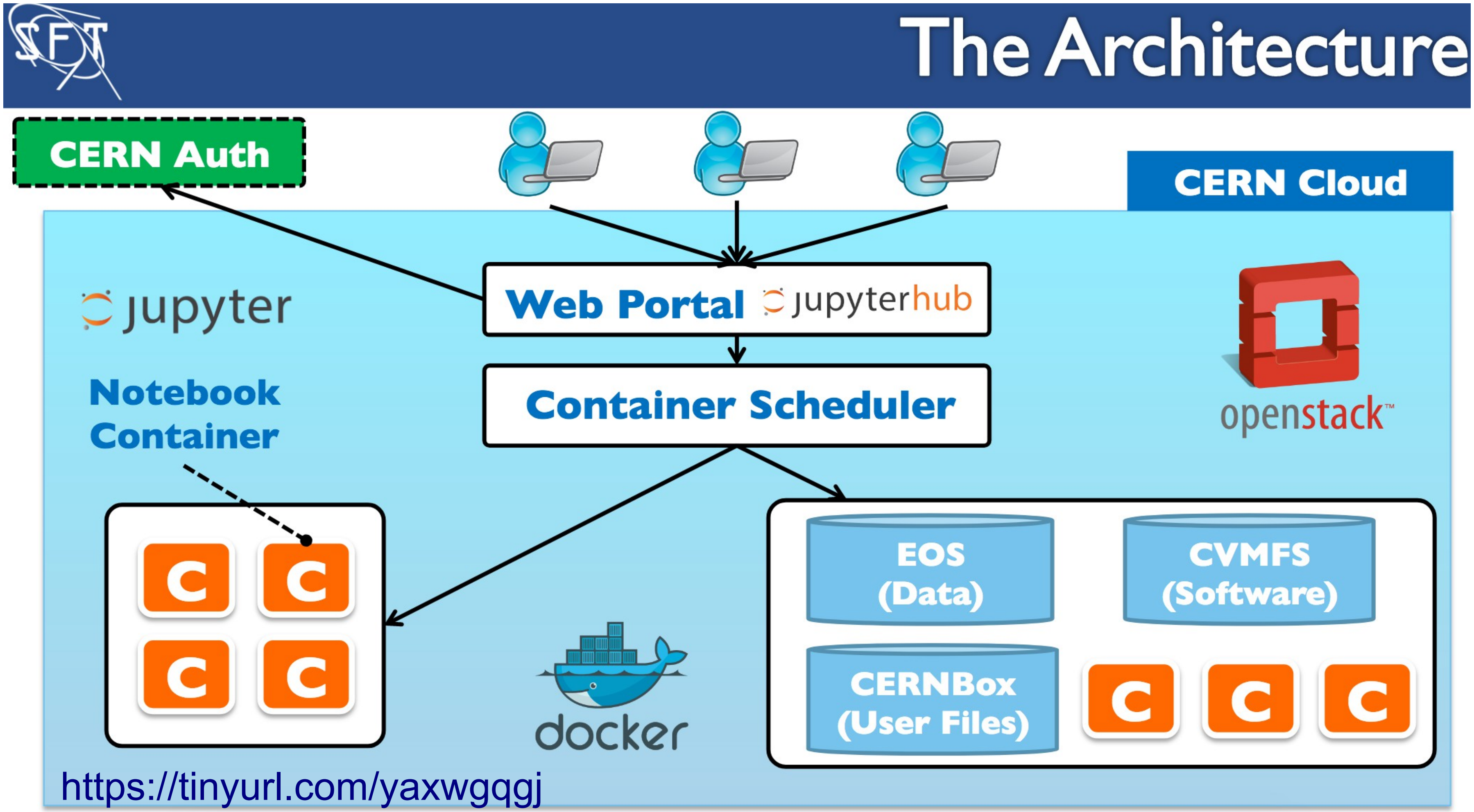
Data Analysis as a Service

- Platform independent: only with a web browser
  - Analyse data via the Notebook web interface
  - No need to install and configure software
- Calculations, input and results "in the cloud"
- Allow easy sharing of scientific results: plots, data, code
  - Storage is crucial, both mass (EOS) and synchronised (CERNBox)
- Simplify teaching of data processing and programming
- Eases analysis reproducibility and documentation
- C++, Python and other languages or analysis "ecosystems"
  - Interfaced to widely adopted scientific libraries
  - e.g. Pandas, Numpy, ROOT, matplotlib, …

https://tinyurl.com/yd5k3cp9

https://swan.web.cern.ch/

Possible Interface To HPC

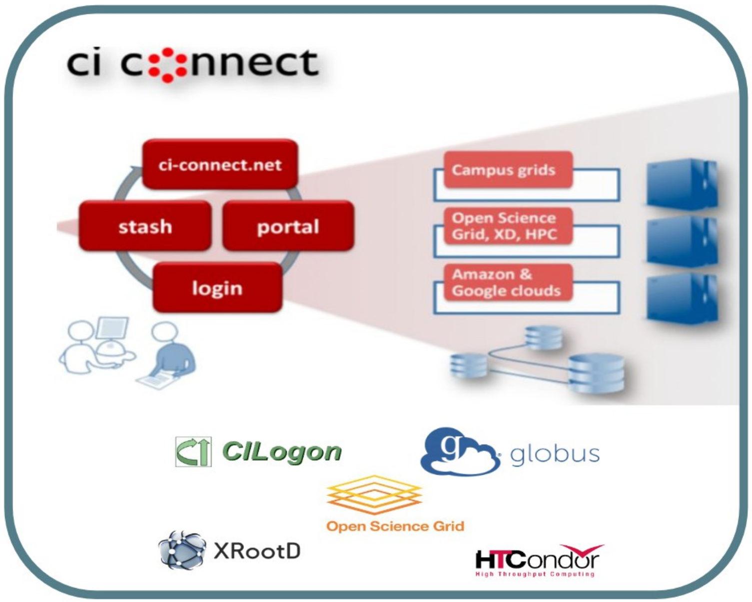


GPU on the GRID

UNIVERSITY OF NOTRE DAME

Technology behind CMS-Connect

Based on CI Connect Platform

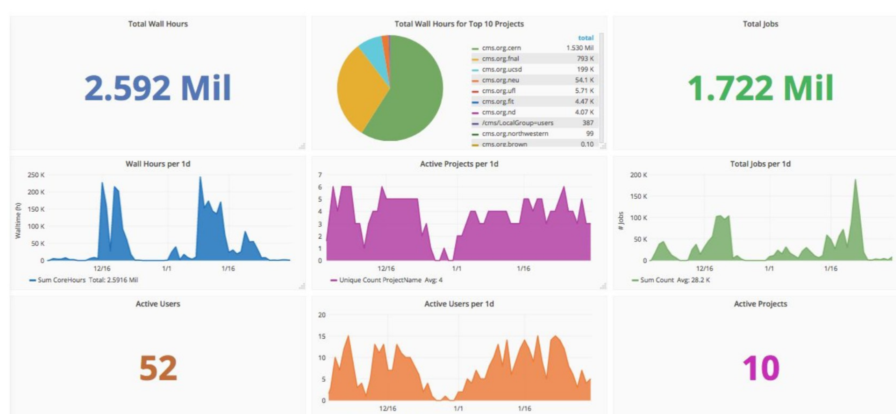


http://connect.uscms.org/

- Globus Platform
  [CILogon + InCommon + X509]
  - Identity Management.
  - Groups, Projects.
- Login Host
  - Auto provisioning of user accounts.
- Connecting CPU/GPU resources
  - HTCondor.
- Distributed Data Access
  - XRootD, Globus access, http.
- Distributed Software
  - cvmfs

https://goo.gl/J7VQtJ

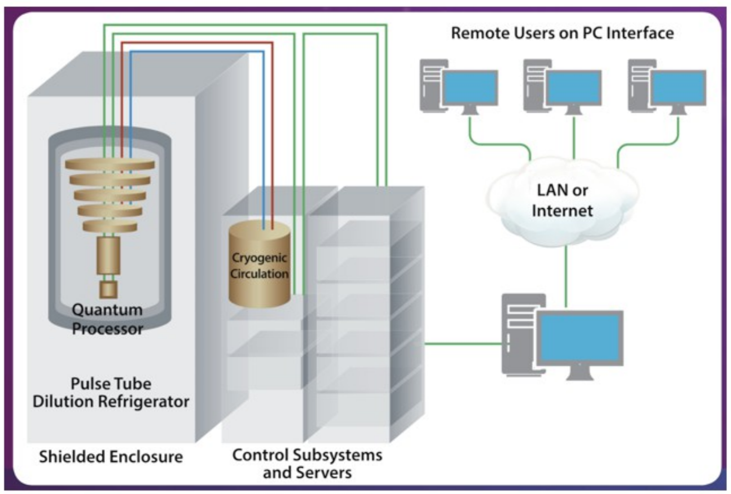
CMS Connect activity in the last 60 days



source

Training As a Service

Working on a D-Wave



- Web Interface to post the problem settings (Hp).
- Asynchronous processing.
- Solution is made available for download.

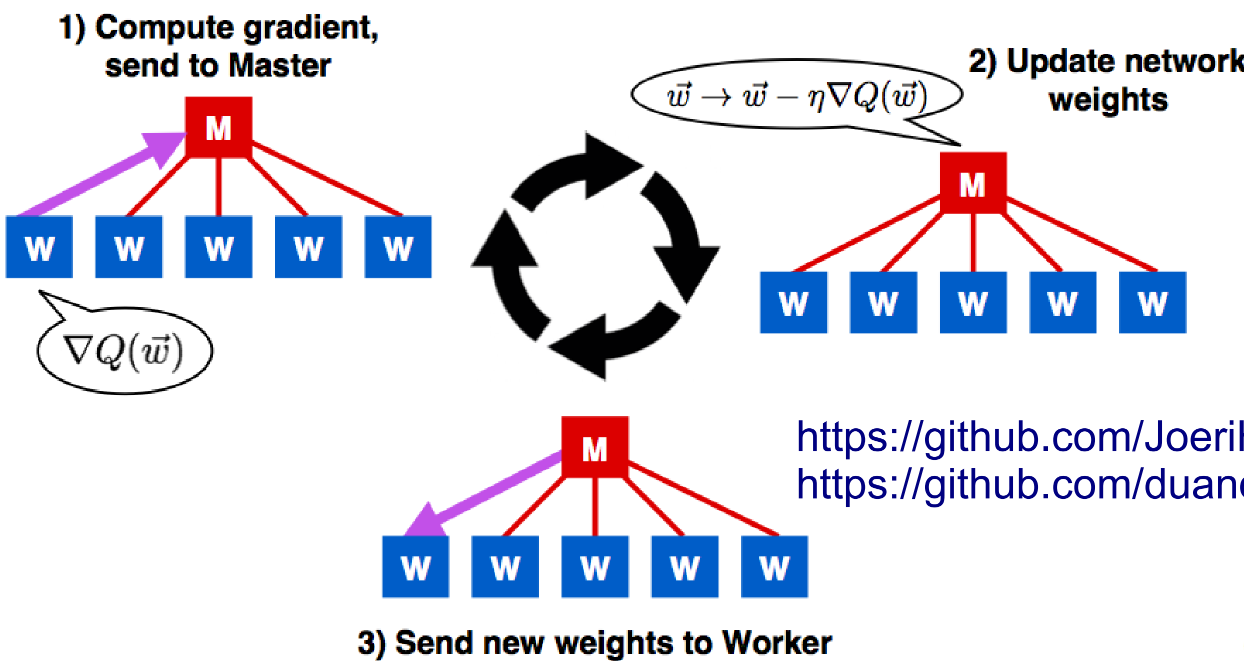
- Distributed library for performing embedding
  - Retain full intellectual property.

- Equivalent restapi to submit and retrieve solutions
  - D-Wave processor as a service

D-Wave Classifier, LAL Seminar, J.-R. Vlimant
02/22/18 22

- Deep Learning training with SGD has a very clear I/O for the user
- Can be abstracted away from the user
- Allows for a secure entry point to large resource, like HPC

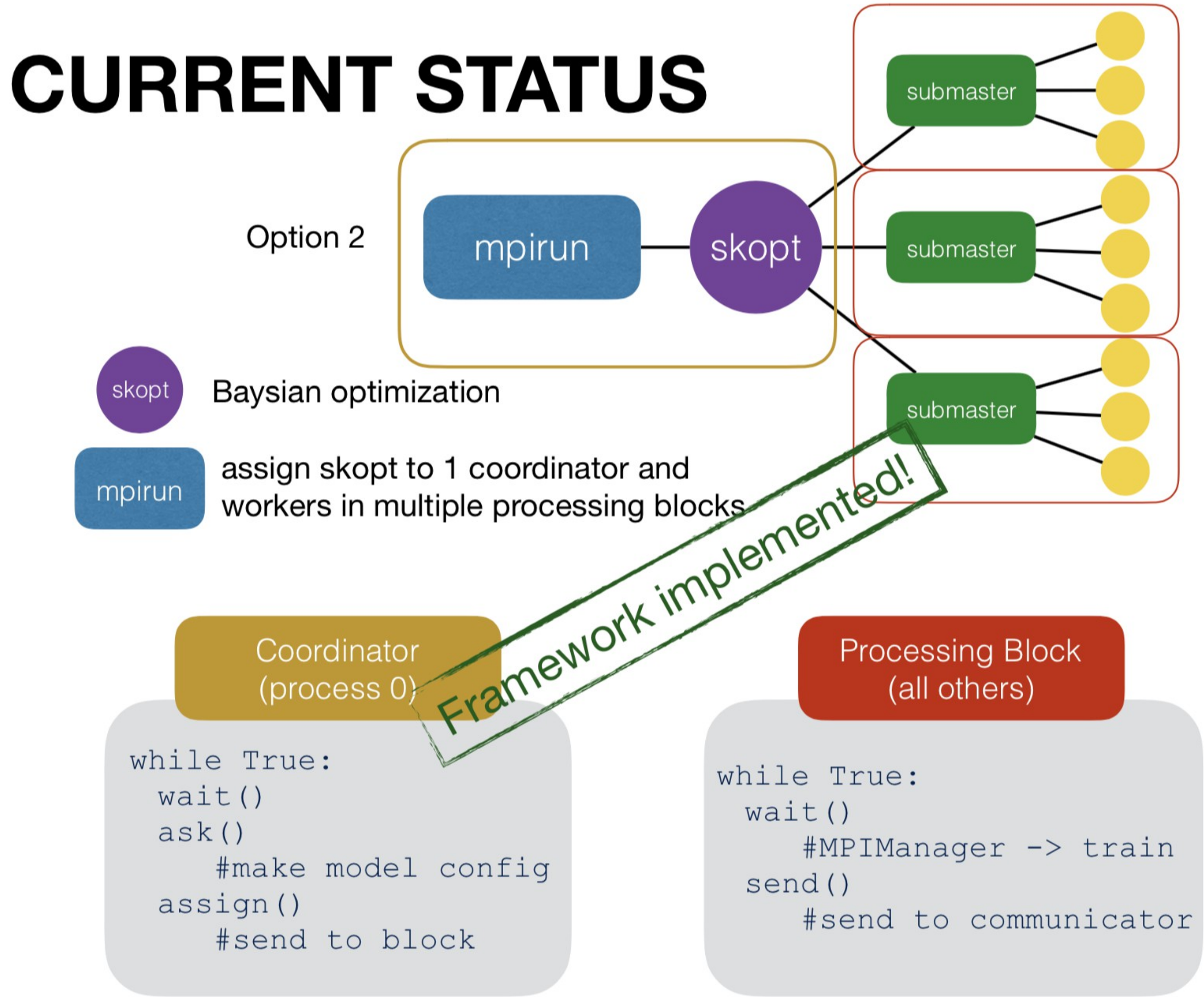
Distributed Training



https://github.com/JoeriHermans/dist-keras
https://github.com/duanders/mpi_learn

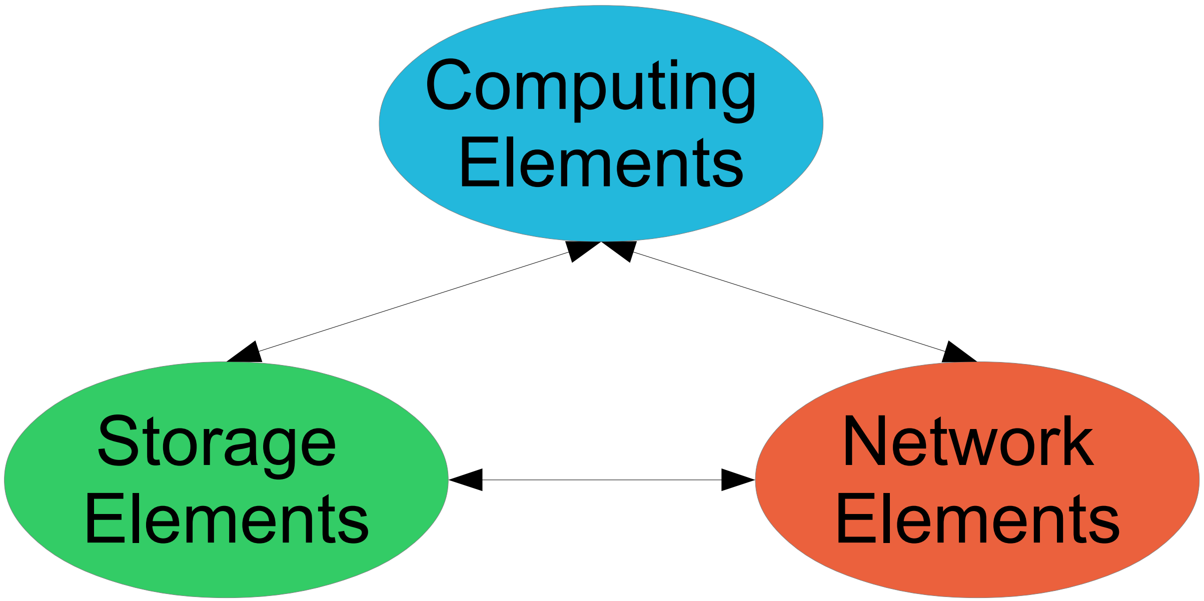
- All deep learning frameworks have developed their own way to do this (elastic or reduce)
- Recent results in scaling up at NERSC
- Can boost science with utilization of HPC
- Still very much in development

Distributed Optimization



Nguyen, Pierini, Anderson, Carta, Vlimant
https://indico.cern.ch/event/683349

Controlled LHC Cloud



- Optimization of each component independently might not lead to the global optimum
- Need to consider the system as a whole

→ Need for a simulator or an environment for exploration
  - Model single element metrics
  - Reinforcement learning to control the system's components

Resource Utilization

CMS metadata on HDFS

- CMS data availability on HDFS: **total size 32+ TB**
  - **AAA** (JSON) user logs accessing XrootD servers, **10TB**
  - **EOS** (JSON) user logs accesses CERN EOS, **4.5TB**
  - **HTCondor** (JSON) CMS Jobs logs, **7.6TB**
  - **FTS** (JSON) CMS FTS logs, **3.5 TB**
  - **CMSSW** (Avro) CMSSW jobs, **0.5TB**
  - **JobMonitoring** (Avro) CMS Dashboard DB snapshot, **0.1TB**
  - **WMArchive** (Avro) CMS Workflows archive, 3TB
  - **ASO** (CSV) CMS ASO accesses, **0.05TB**
  - **DBS** (CSV) CMS Data Bookkeeping snapshot, **1.1TB**
  - **PhEDEx** (CSV) CMS data location DB snapshot, **2.5TB**

- Metadata from all workflow management, job scheduler, data management services on HDFS at CERN
- Lots of possible insight to be gain from these large datasets

➢ Contribution to improve monitoring

➢ Contribution on understanding how to operate the systems

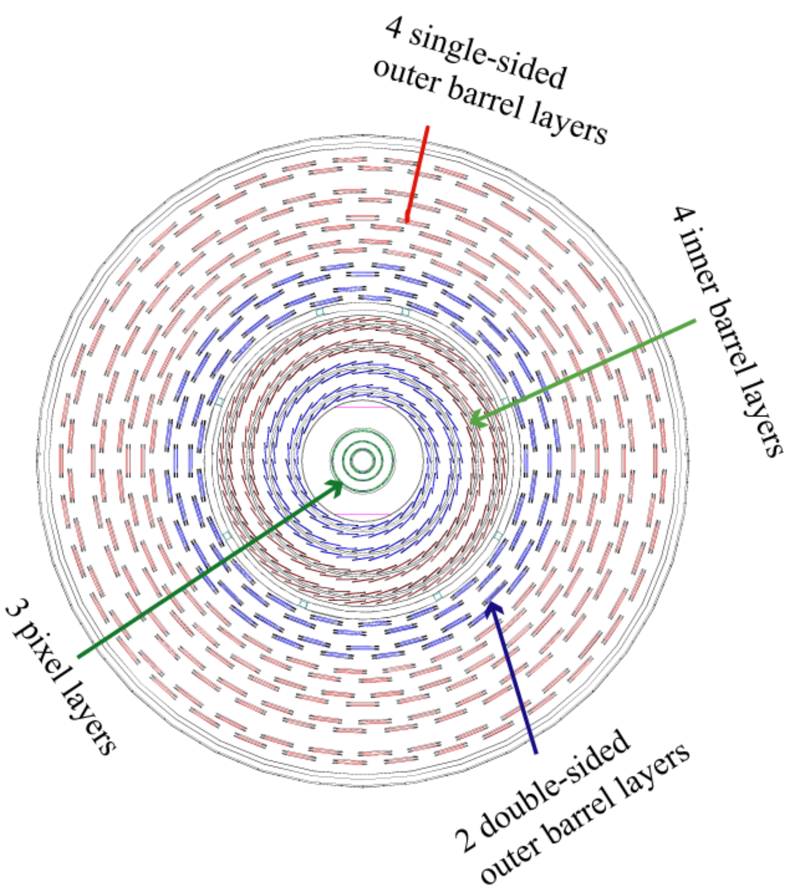
Summary

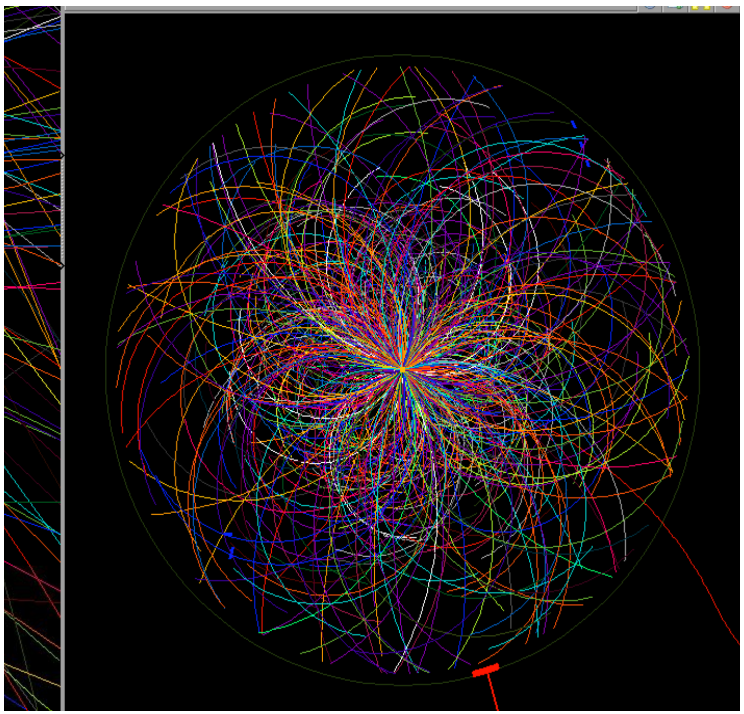
- Lots of potential applications of deep learning within the big data pipeline of LHC data.
- Several potential projects to contribute to providing resources/data/software to physicists at the LHC.

A rather large set of
backup slides for reference

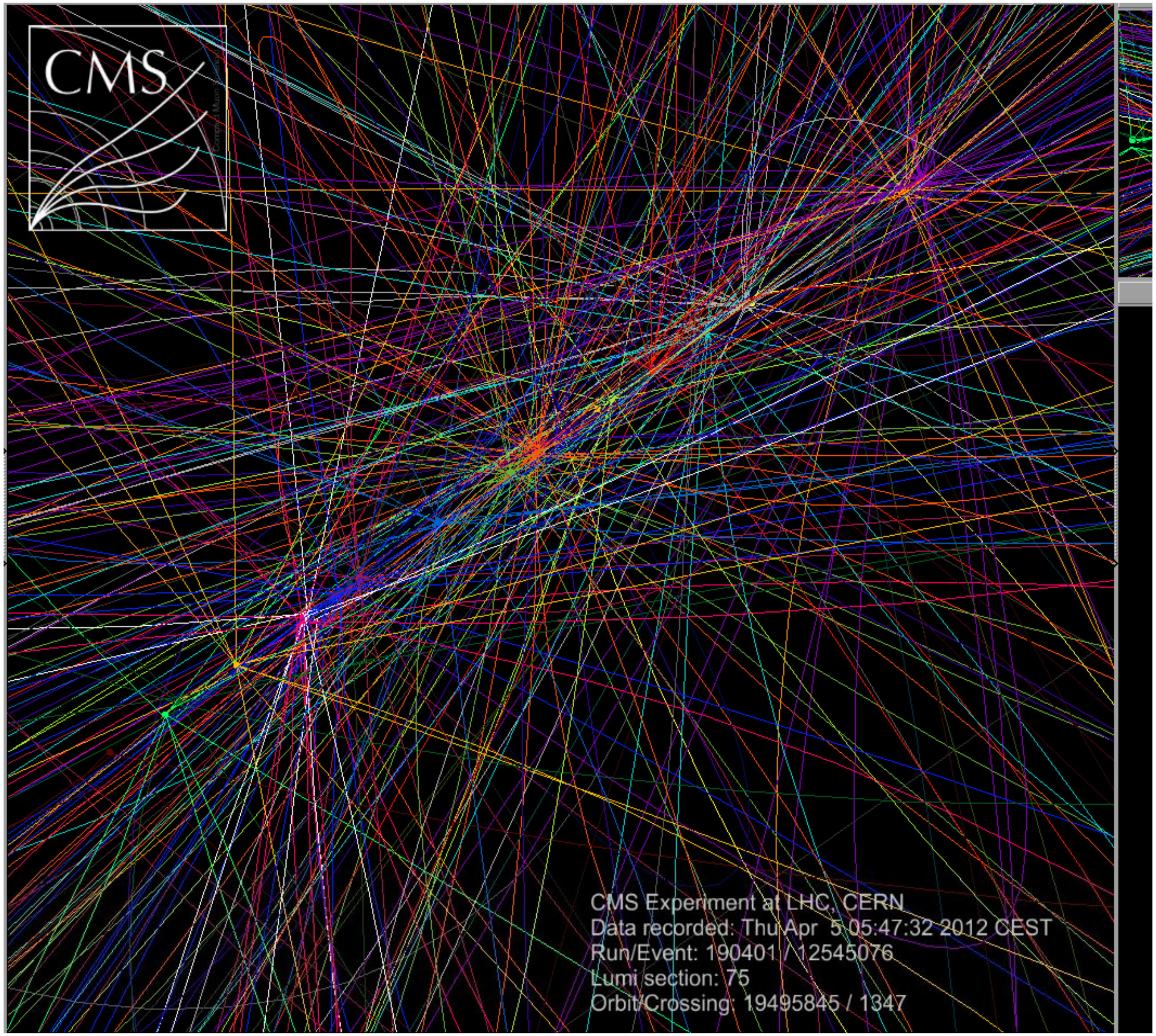
Tracks Pattern Recognition

- From sparse 2D/3D points reconstruct the path of a charged particle
- Iterative process using combinatorics, Kalman Fitting and Filtering
- Most CPU intensive part of the event reconstruction (~10s /event)
- Computation time scales ~quadratically with number of interactions
- Any fraction of patterns that can identified faster will make a difference



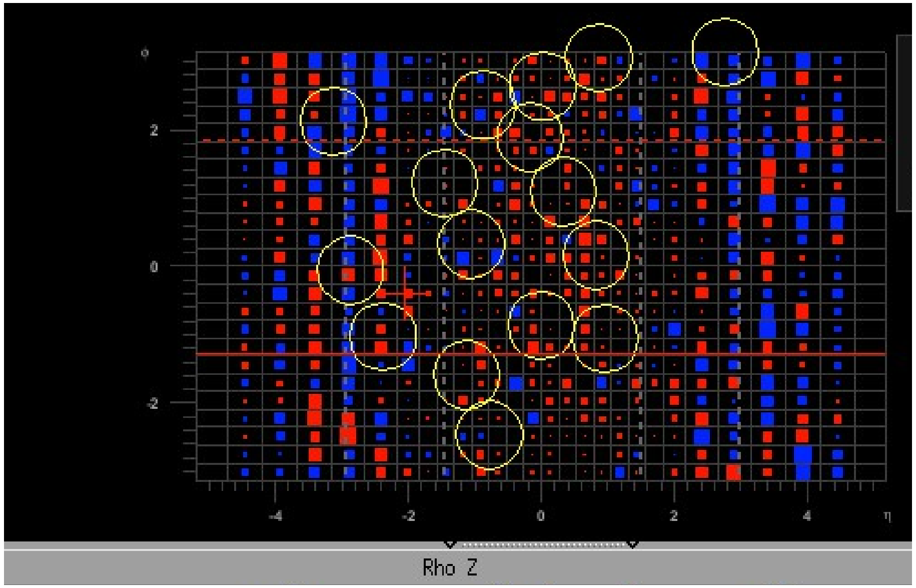


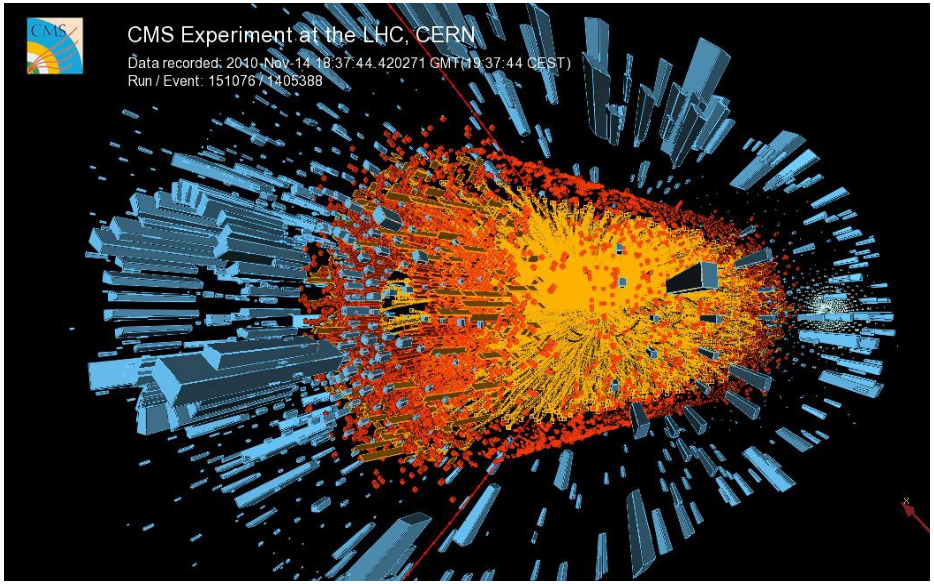
Vertex Identification



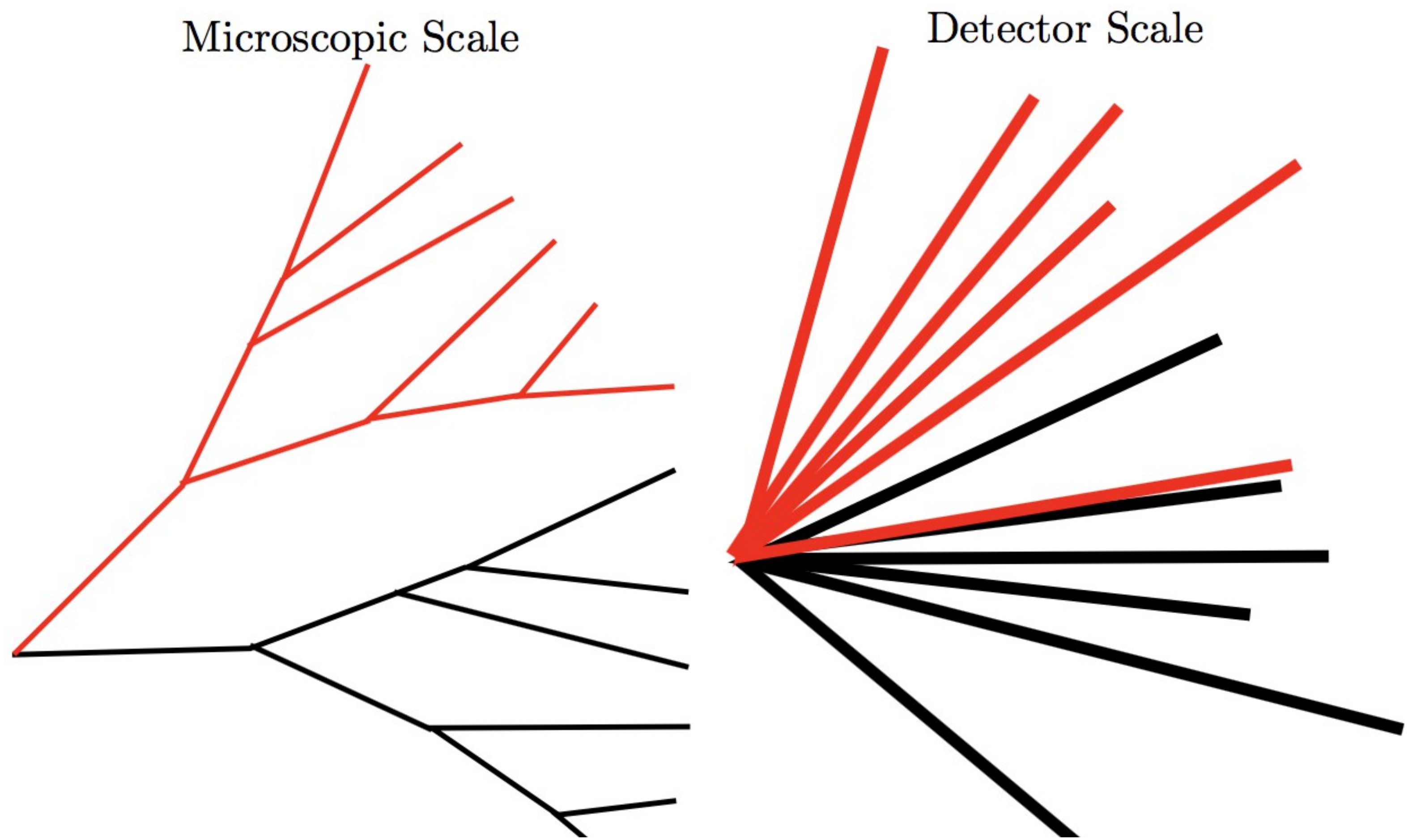
Energy Pattern Recognition

- Particles emitted from the interaction point are stopped in calorimeters (except for muons, neutrinos, …)
- Pattern of energy deposition is somehow characteristic
- Classical, physics driven methods have been used to recollect the total energy and identify the particle
- Efficient classifiers are being used on derived features
- Room for improvement in deriving the low level features
- How to deal with so many overlapping collisions



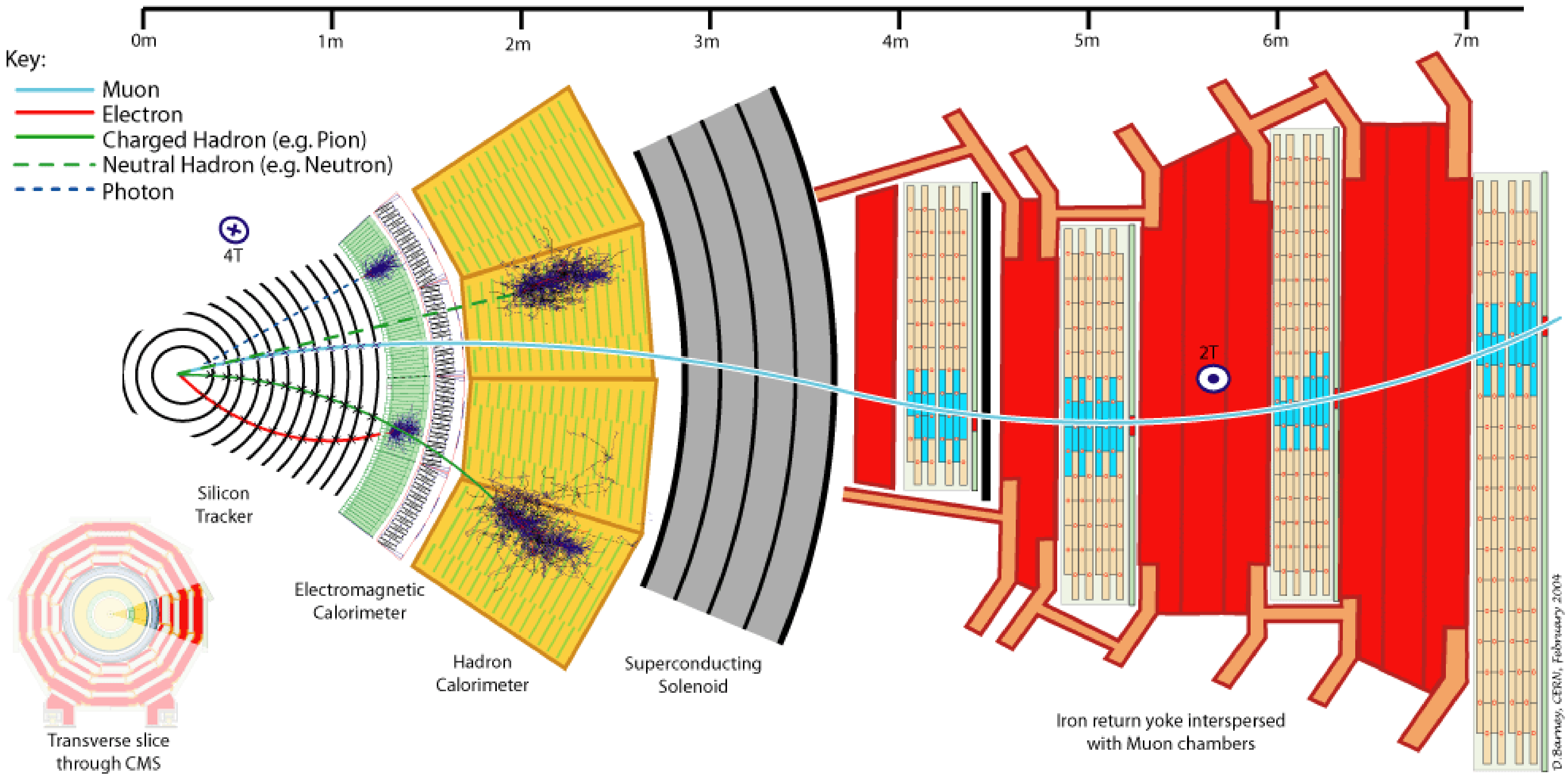


What is Jet

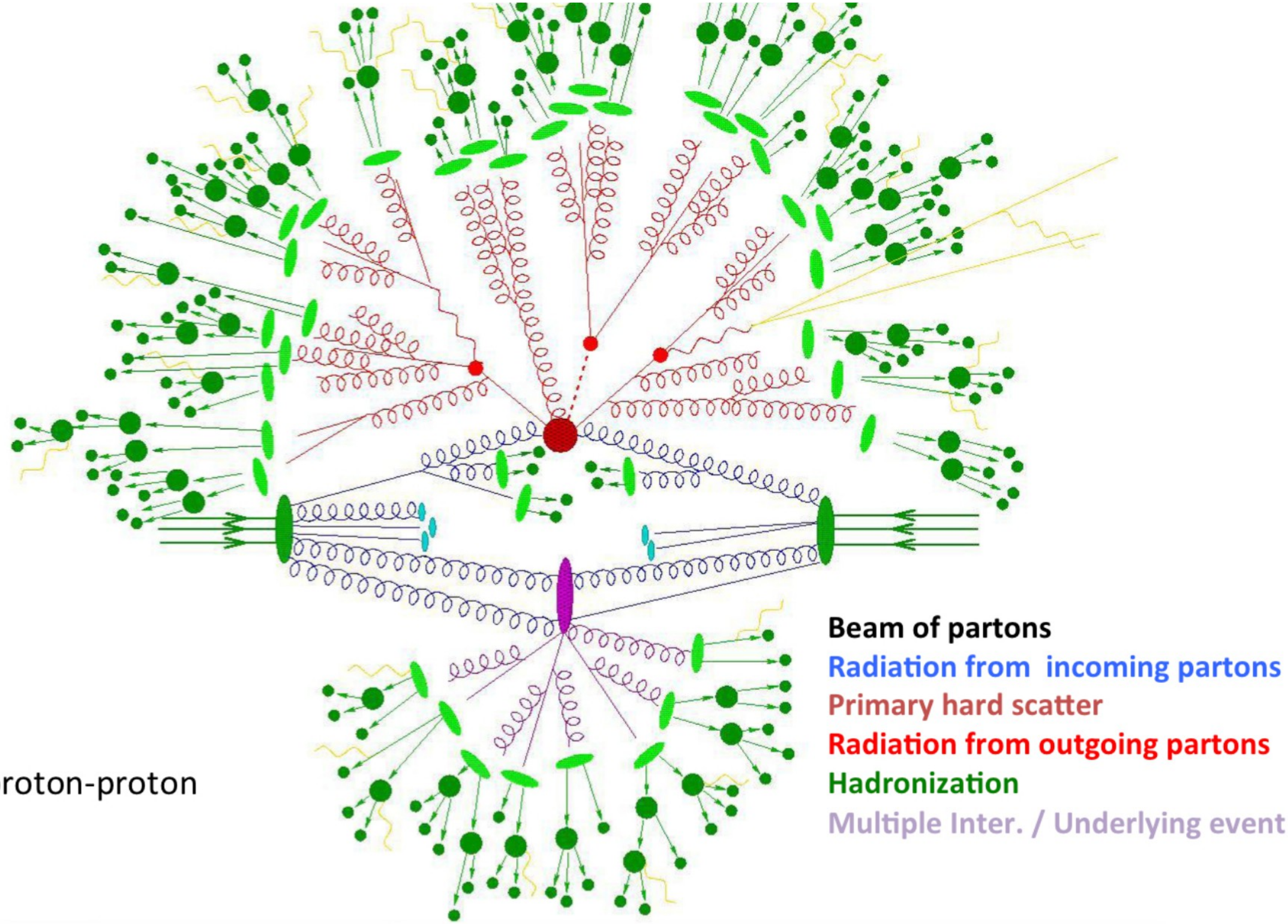


- Partons (quark ,gluons) have to be in pairs or triplets in nature
- Parton gets "suited" with partners as propagating away from creation
- The result is a "jet" of particles in the direction of the original parton
- The jet collimation depends on the energy of the initial parton

A Journey Through Matter



What is an Event



Typical proton-proton collision

Add 40 such on top of each other.
Up to 200 such overlay in the horizon 2025
One event every 25 ns / 40MHz

Why Deep Learning

- LHC Data Processing may **use deep learning methods in many aspects**
- Several class of **LHC Data analysis make use of classifier** for signal versus background discrimination.
  - Use of BDT on high level features
  - Increasing use of MLP-like deep neural net
- Deep learning has delivered **super-human performance** at certain class of tasks (computer vision, speech recognition, ...)
  - Use of convolutional neural net, recurrent topologies, long-short-term-memory cells, ...
- Deep learning has advantage at **training on "raw" data**
  - Several levels of data distillation in LHC data processing
- Neural net computation is highly parallelizable
  - → Better use of GP-GPU than regular HEP algorithms
- Complex systems to operate, complex signal to analyze
  - Over-come challenges of data density and volume
  - Over-come challenges of data and detector complexity
  - Over-come challenges of ultra-fast decision

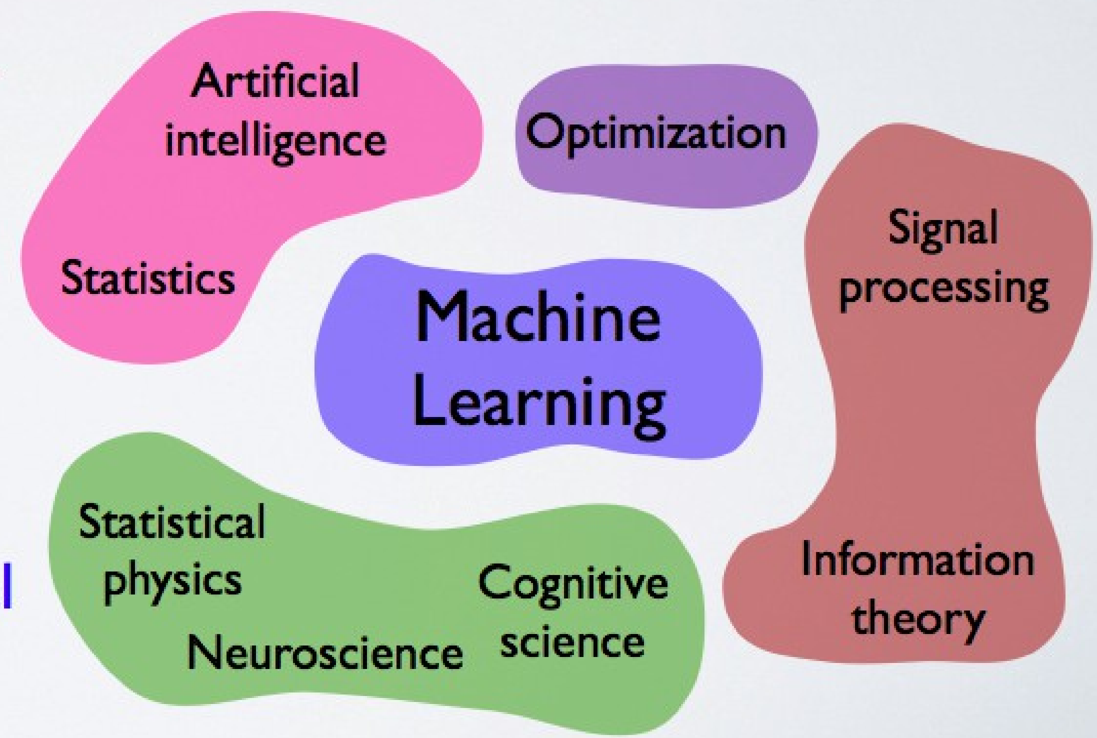
Where Deep Learning

- Detector and apparatus control
- Computing GRID, Center & network control
- Operation anomaly detection
- Fast triggering on object or full events
- Fast approximate algorithms
- Rare event detection
- Automated data certification
- Faster simulation software
- Finer even selection
- Better object identification
- More precise event reconstruction
- More robust measurements
- ...

Machine Learning in a Nutshell

"The science of getting computers to act without being explicitly programmed" - Andrew Ng (Stanford/Coursera)

- part of standard computer science curriculum since the 90s
- inferring knowledge from data
- generalizing to unseen data
- usually no parametric model assumptions
- emphasizing the computational challenges



Balazs Kegl, CERN 2014

What Machine Learning

- Classification and regression
- Deep neural nets (CNN, RNN, ...)
- Unsupervised clustering
- Control theory
- Re-inforcement learning
- Generative models
- Density estimators
- Interaction networks
- Graph networks
- ...

Application to Intensity and Energy Frontiers

(a selected few)

Data Science in HEP Series

http://cern.ch/DataScienceLHC2015

https://indico.hep.caltech.edu/indico/event/102

http://dshep.fnal.gov/

Connecting the Dots Series

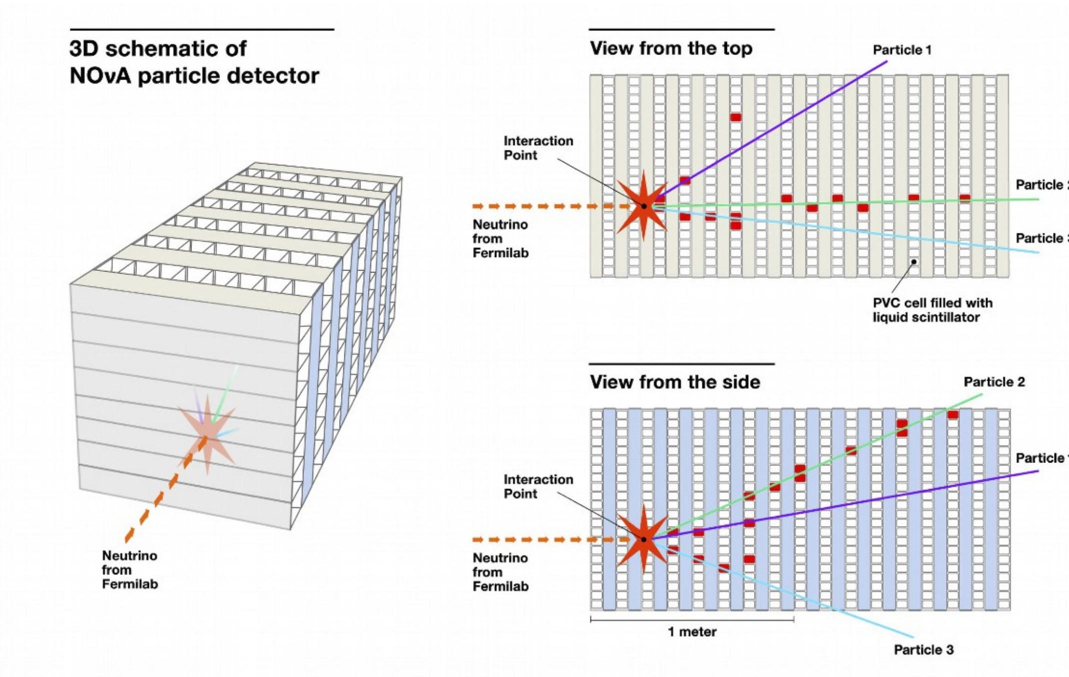
https://indico.hephy.oeaw.ac.at/event/86/

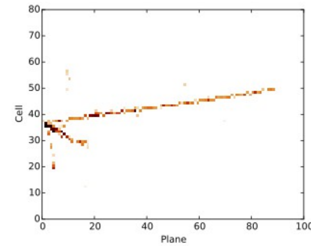
https://ctdwit2017.lal.in2p3.fr/

Hammers and Nails

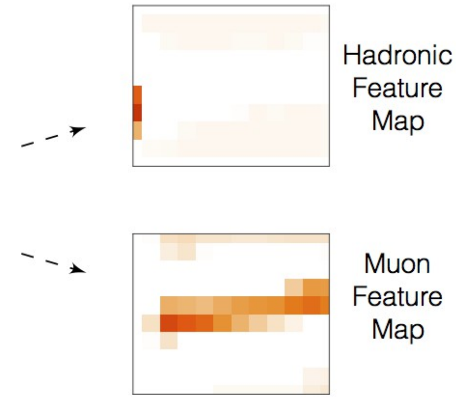
https://www.weizmann.ac.il/conferences/SRitp/Summer2017/

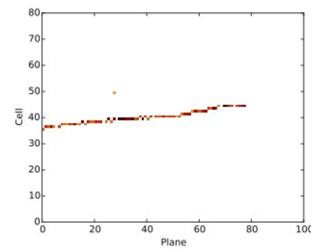
NOVA Event Classification



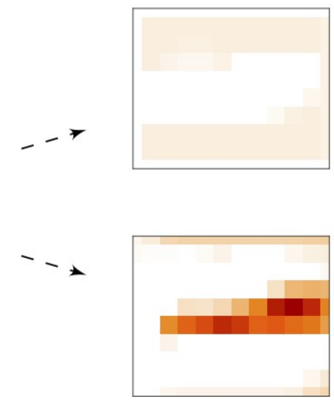


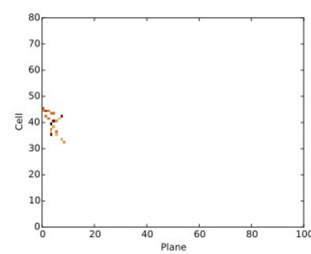
Muon Neutrino
DIS CC



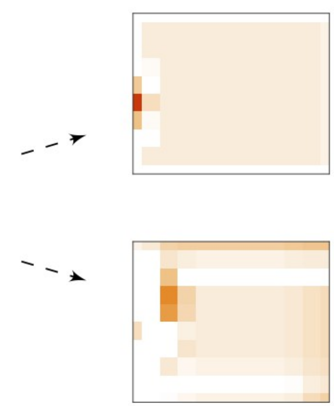


Muon Neutrino
QE CC





Muon Neutrino
NC



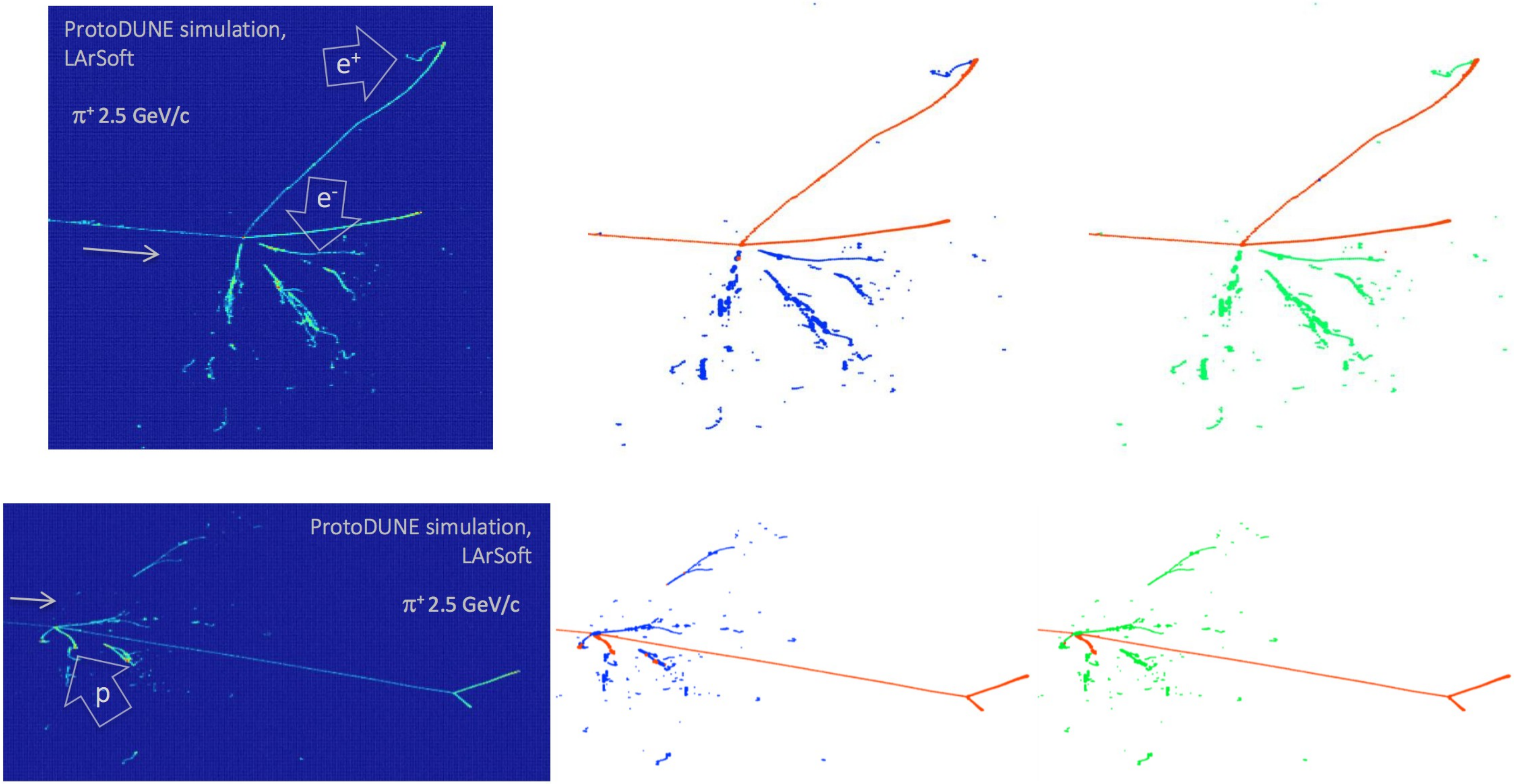
	CVN Selection Value	ν_e sig	Tot bkg	NC	ν_μ CC	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	88.4	509.0	344.8	132.1	32.1	–	14.8%
$s/\sqrt{b}$ opt	0.94	43.4	6.7	2.1	0.4	4.3	49.1%	86.6%
$s/\sqrt{s+b}$ opt	0.72	58.8	18.6	10.3	2.1	6.1	66.4%	76.0%

	CVN Selection Value	ν_μ sig	Tot bkg	NC	Appeared ν_e	Beam ν_e	Signal Efficiency	Purity
Contained Events	–	355.5	1269.8	1099.7	135.7	34.4	–	21.9%
$s/\sqrt{b}$ opt	0.99	61.8	0.1	0.1	0.0	0.0	17.4%	99.9%
$s/\sqrt{s+b}$ opt	0.45	206.8	7.6	6.8	0.7	0.1	58.2%	96.4%

- 40% Better Electron Efficiency for same background.

http://arxiv.org/pdf/1604.01444.pdf

Flavor Segmentation

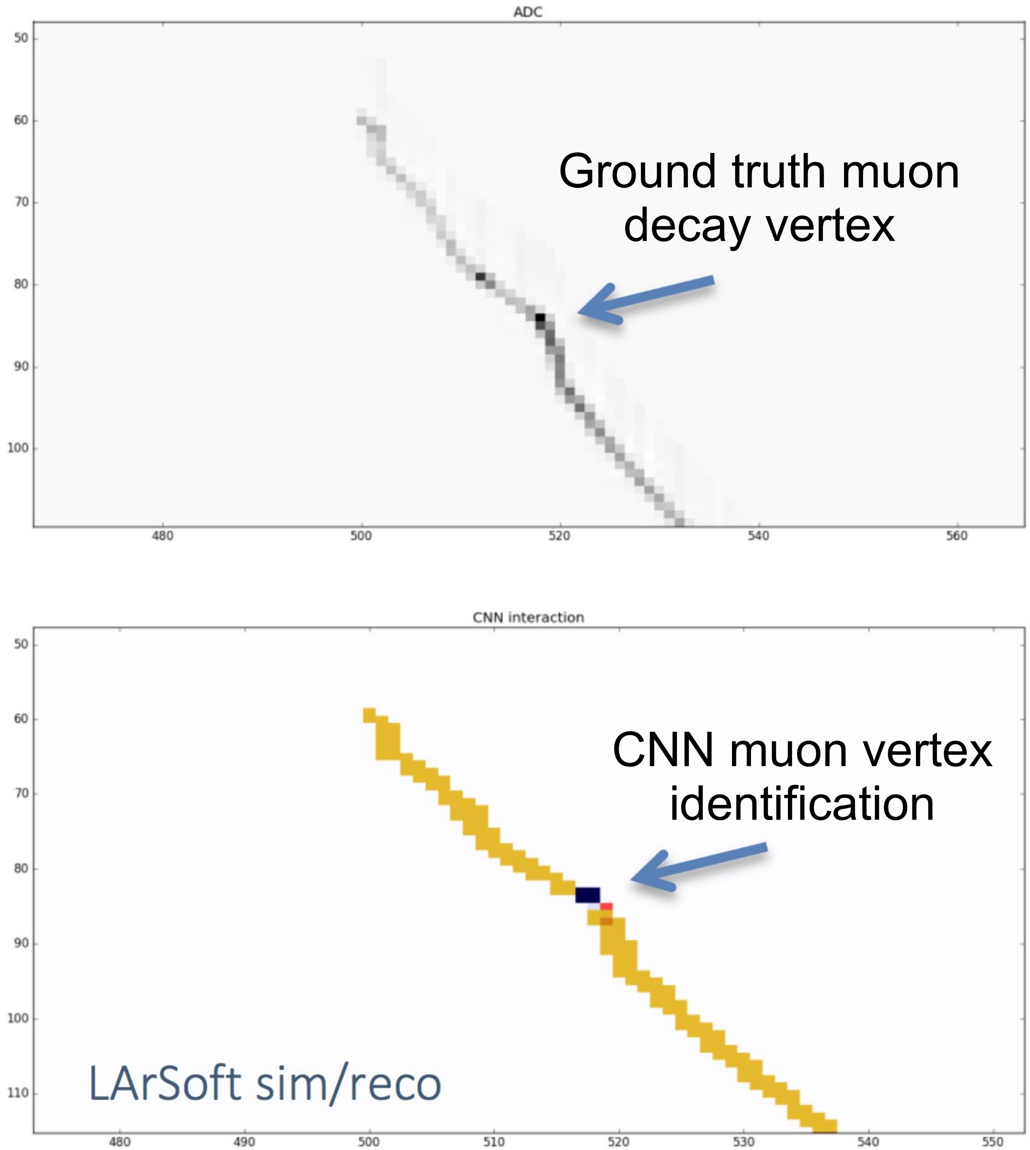


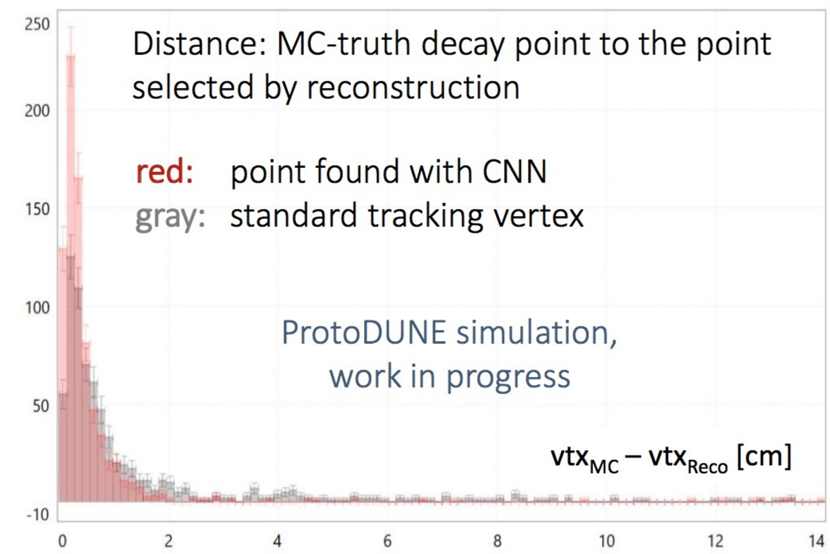
input: 2D ADC

CNN output:
EM-like (blue) / track-like (red)

MC truth:
EM-like (green) / track-like (red)

Decay Point Identifier

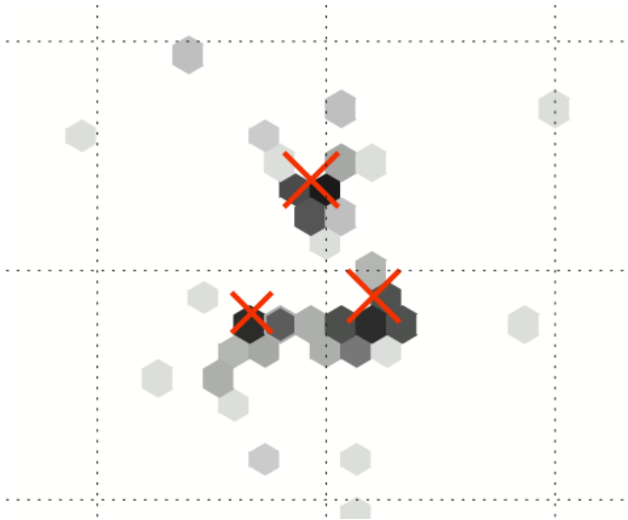




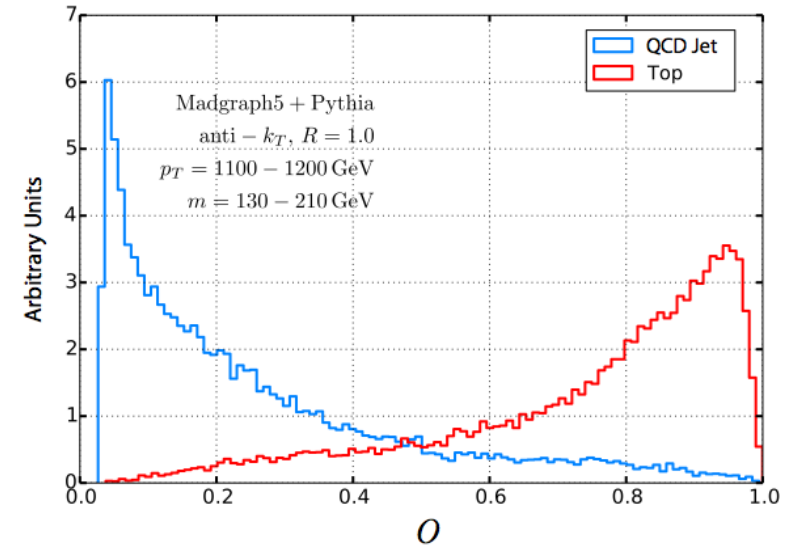
- CNN slightly outperform the classical approach
- Much less complication in programming the vertex finding

Particle Jet Identification

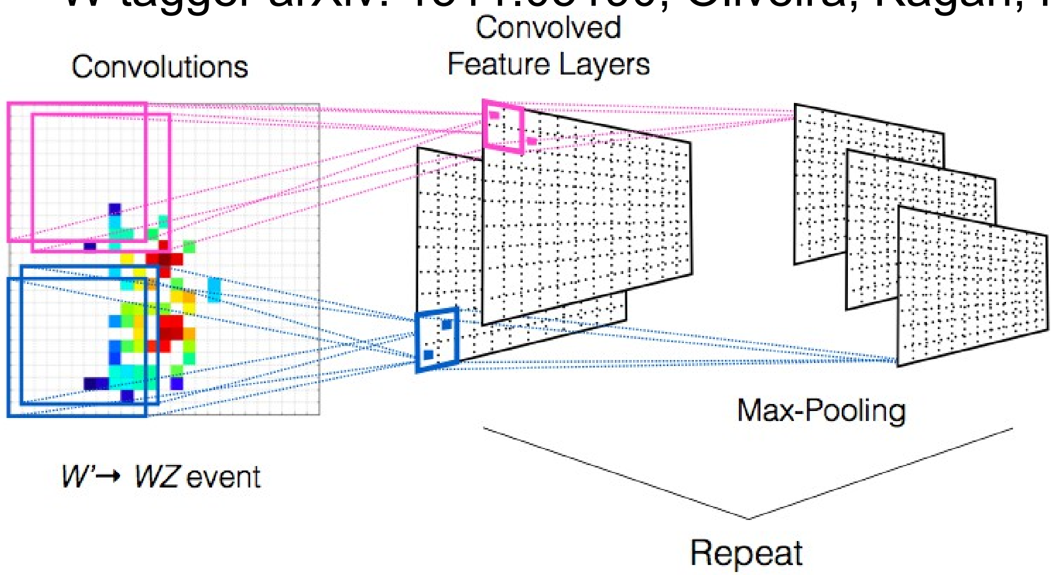
Top Tagger arXiv: 1501.05968 Almeida, Backovic, Cliche, Lee, Perelstein



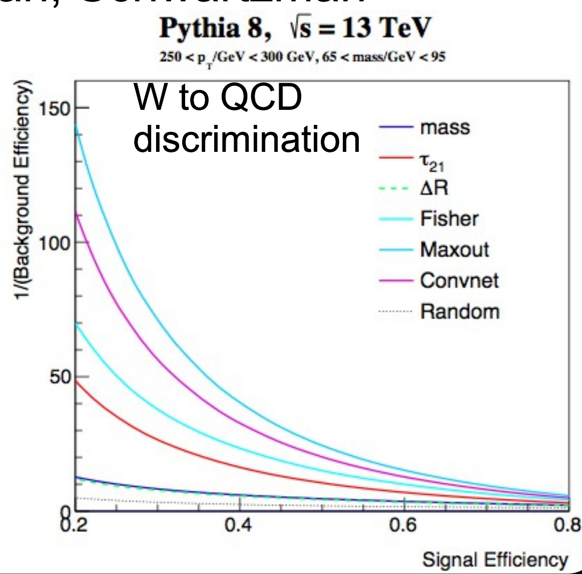
Neural net



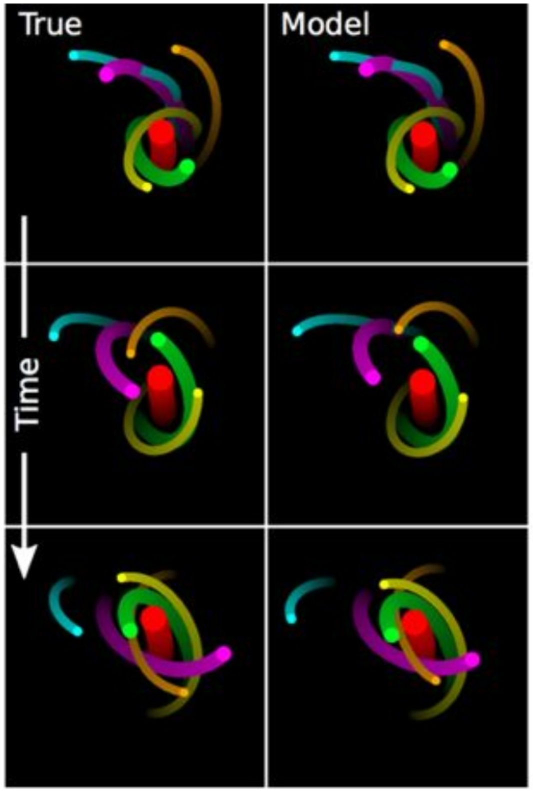
W tagger arXiv: 1511.05190, Oliveira, Kagan, Mackey, Nachman, Schwartzman



Train



Interaction Network For Jet-id



- Graph $G = \langle O, R \rangle$, **o**bjects connected by **r**elations
- Interaction Network

$$\phi_O(a(G,\ X,\ \phi_R(\ m(G)\ )\ ))$$

$$\begin{aligned} m(G) &= B = \{b_k\}_{k=1\ldots N_R} & a(G,X,E) &= C = \{c_j\}_{j=1\ldots N_O} \\ f_R(b_k) &= e_k & f_O(c_j) &= p_j \\ \phi_R(B) &= E = \{e_k\}_{k=1\ldots N_R} & \phi_O(C) &= P = \{p_j\}_{j=1\ldots N_O} \end{aligned}$$

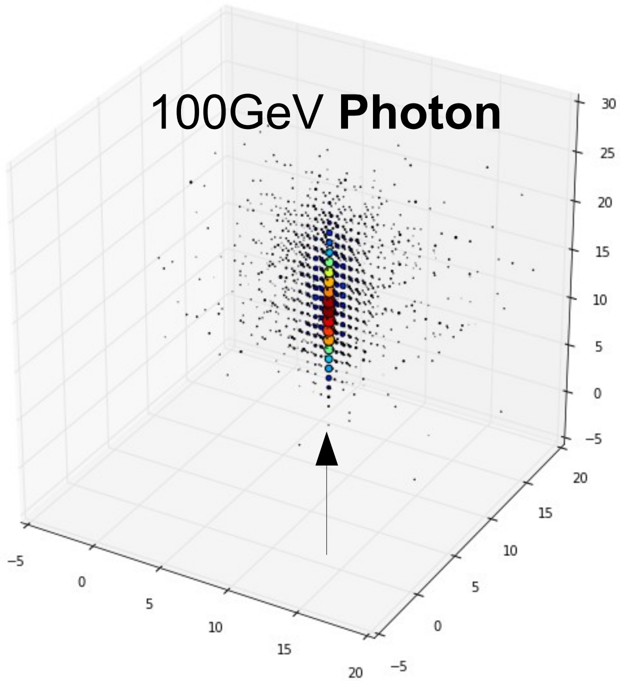
- ϕ_R predicts relational effects
- ϕ_O predicts effect on objects
- Allows for longer-range interactions than a standard CNN

Interaction Networks for Learning about Objects, Relations and Physics
P. W. Battaglia, R. Pascanu, M. Lai, D. Rezende, K. Kavukcuoglu
https://arxiv.org/abs/1612.00222

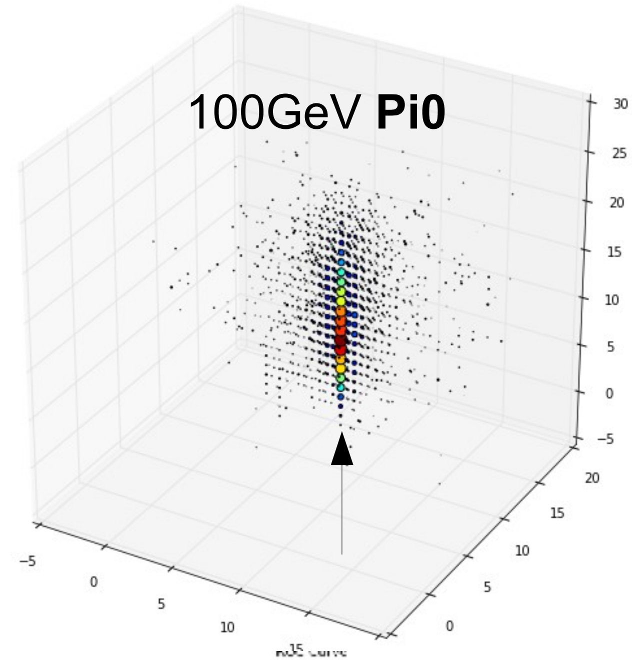
- Learning the relation between particles (gravity, spring, wall, ...)

➔ (on-going work) Applied to jet identification using all particles it is made of

3D Calorimetry Imaging

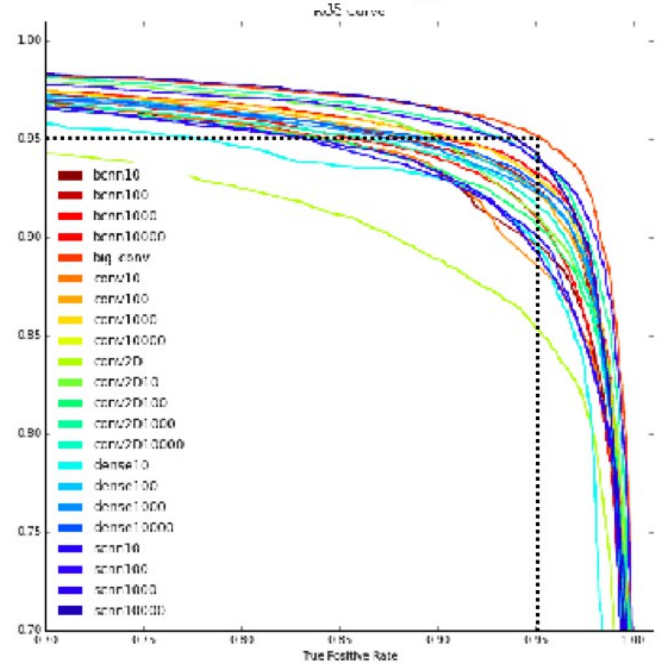


≠



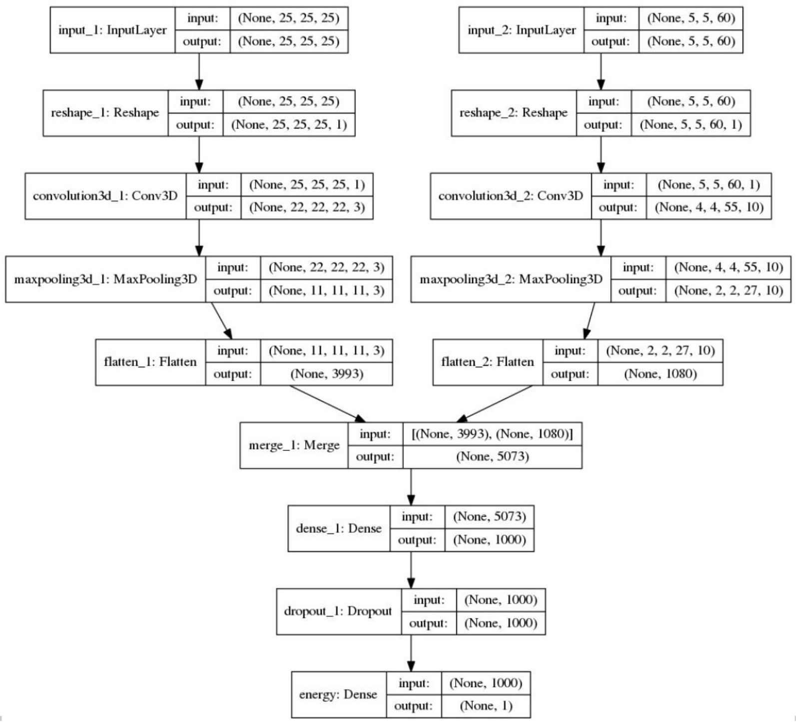
LCD Calorimeter configuration
http://lcd.web.cern.ch
5x5 mm Pixel calorimeter
28 layer deep for Ecal
70 layer deep for Hcal

Photon and pion particle gun
Classification models

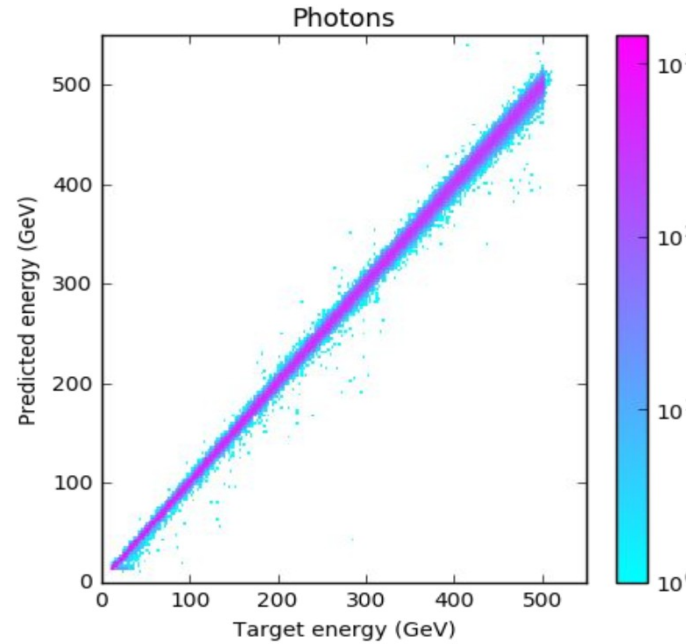


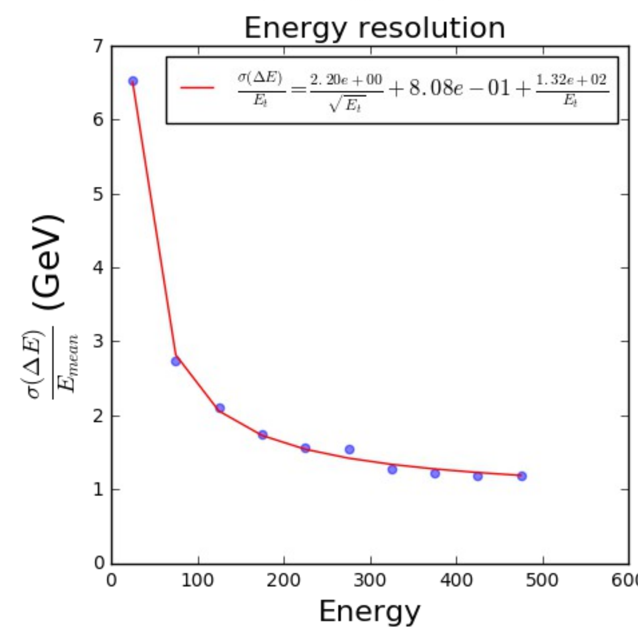
3D Calorimetry Regression

Model topology



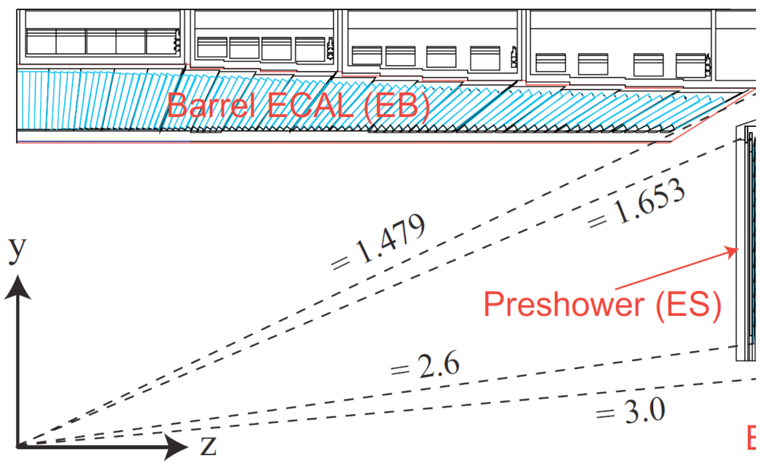
- **MSE**
- **Adam** optimizer
- Conv. layer activation function: **ReLU**
- Dense layer activation function: **linear**



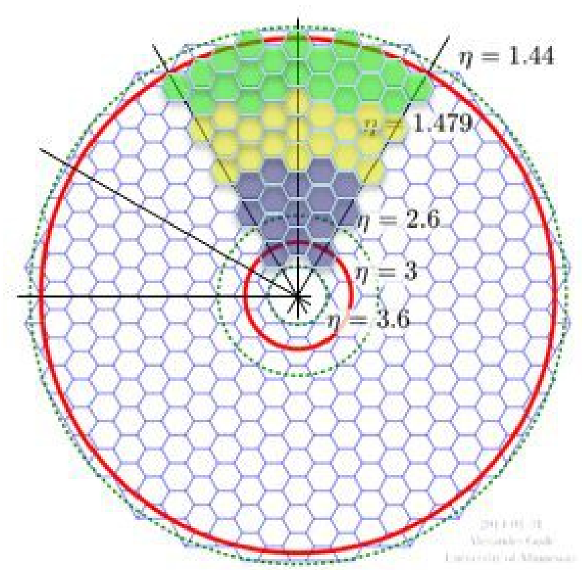


Calibrate the energy deposition using convolution neural nets

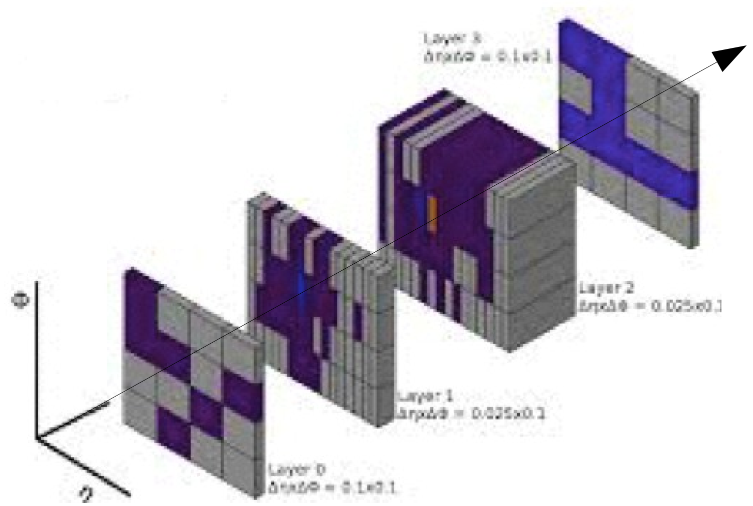
Challenge of Geometry



Projective Geometry



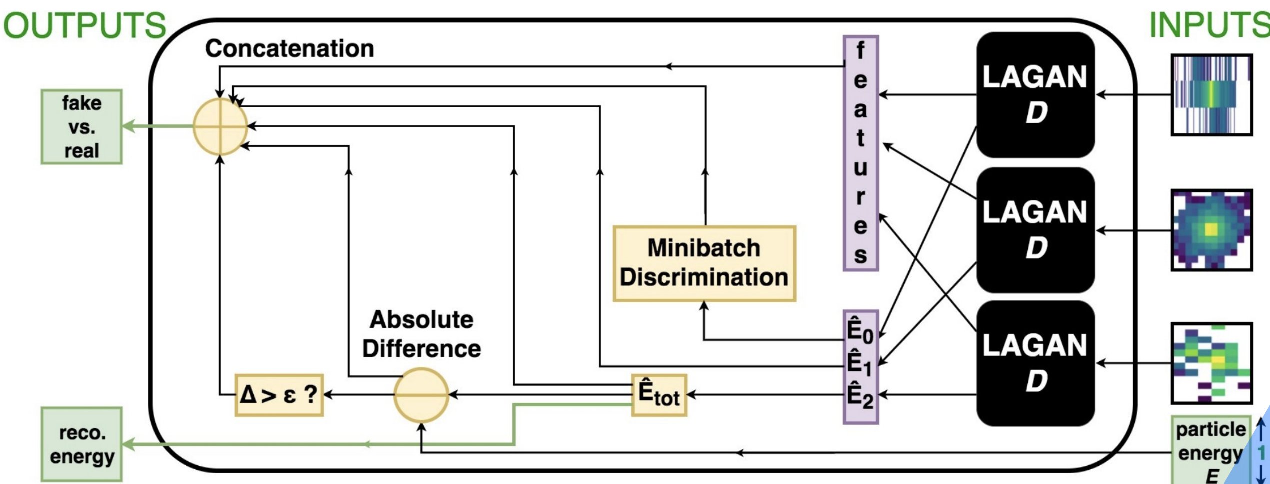
Hexagonal cells



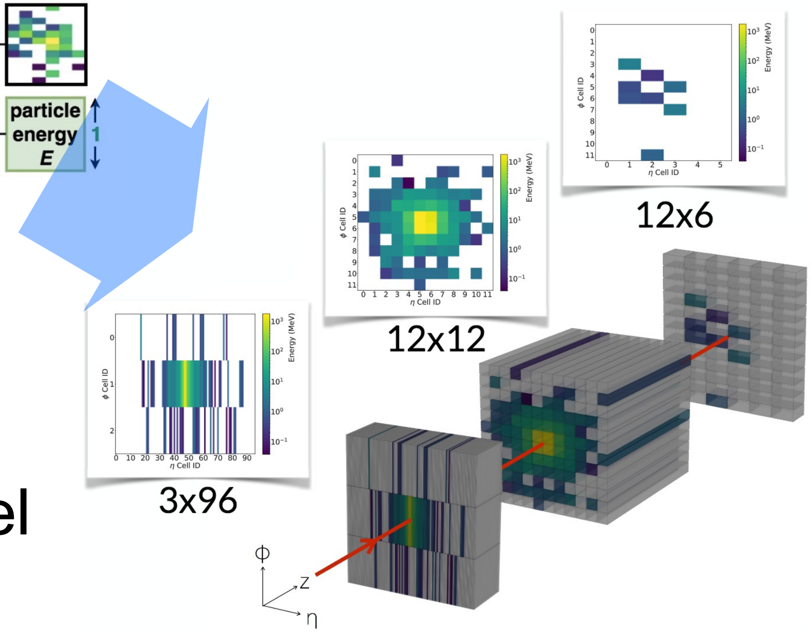
Variable Depth Segmentation

The images we are dealing with are not as regular as standard images. Need for specific new treatment and methods to feed neural nets

Calorimeter GAN

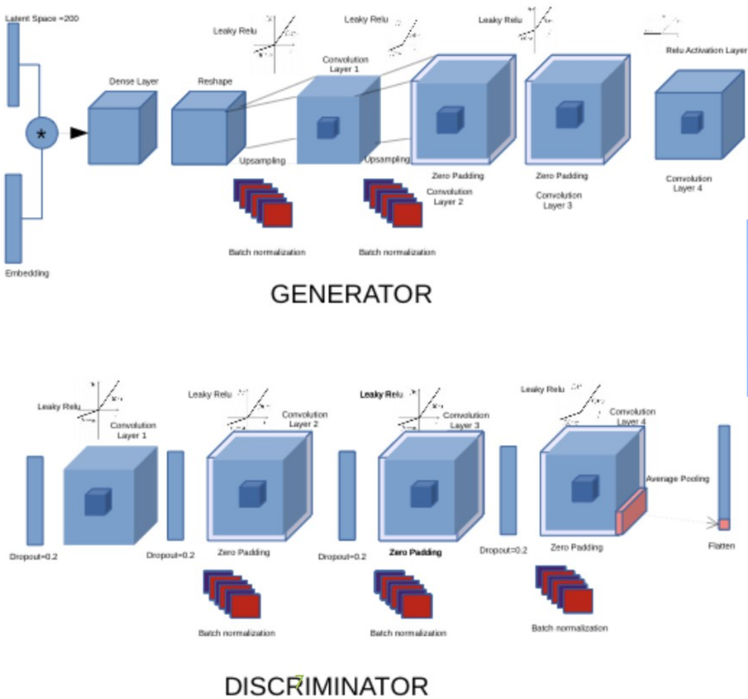


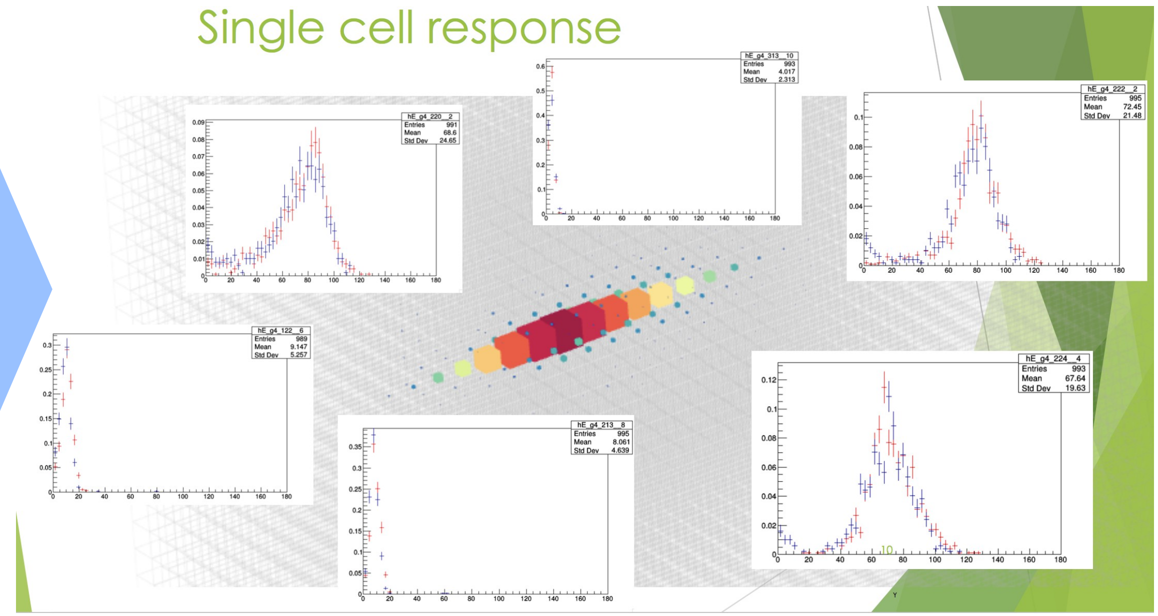
CaloGAN
M. Paganini, L. de Oliveira, B. Nachman
https://arxiv.org/abs/1705.02355



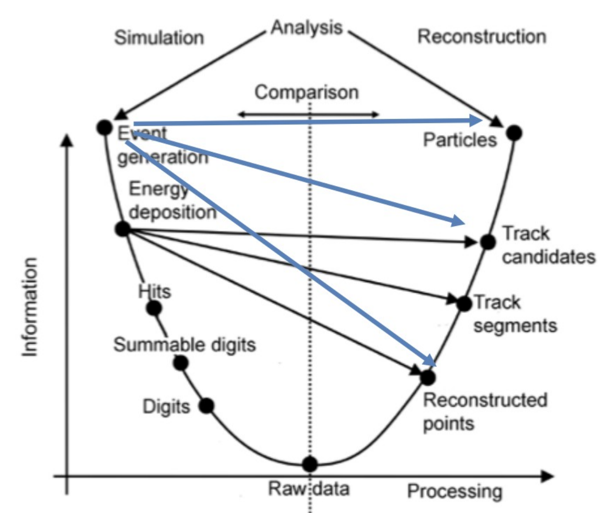
Complex generative adversarial model based on previous 2D GAN
High reward application to fast simulation for the ATLAS experiment

3D GAN





Work in progress base on previous work on 2D GAN
Aim at accelerating part of the GeantV simulation



Collision Event Classification

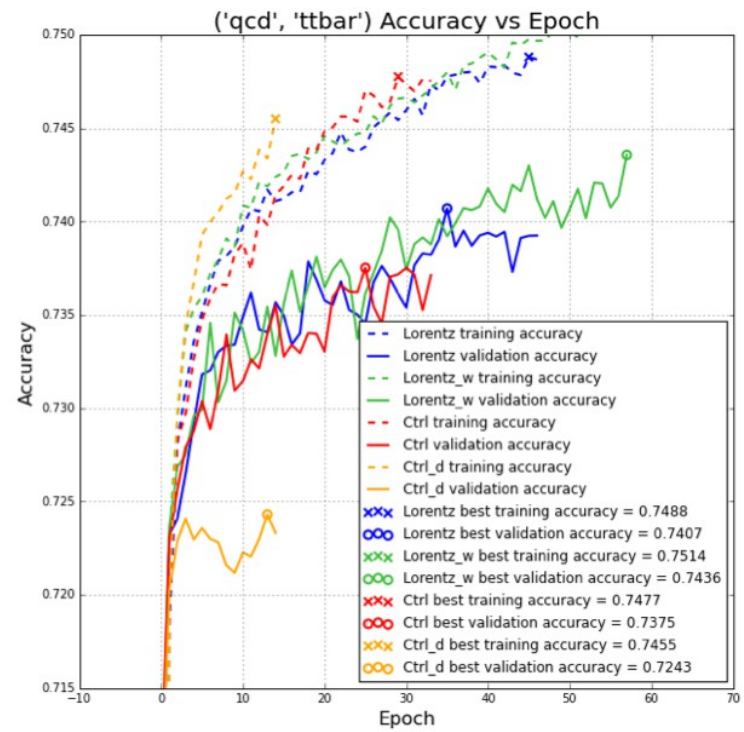
$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} ; v_x\,, v_y\,, v_z \in (-1,1)\,, \qquad (1)$$

$$\mathbf{n} = \frac{\mathbf{v}}{\|\mathbf{v}\|} \qquad (2)$$

$$\gamma = \frac{1}{\sqrt{1-\mathbf{v}\cdot\mathbf{v}}} \qquad (3)$$

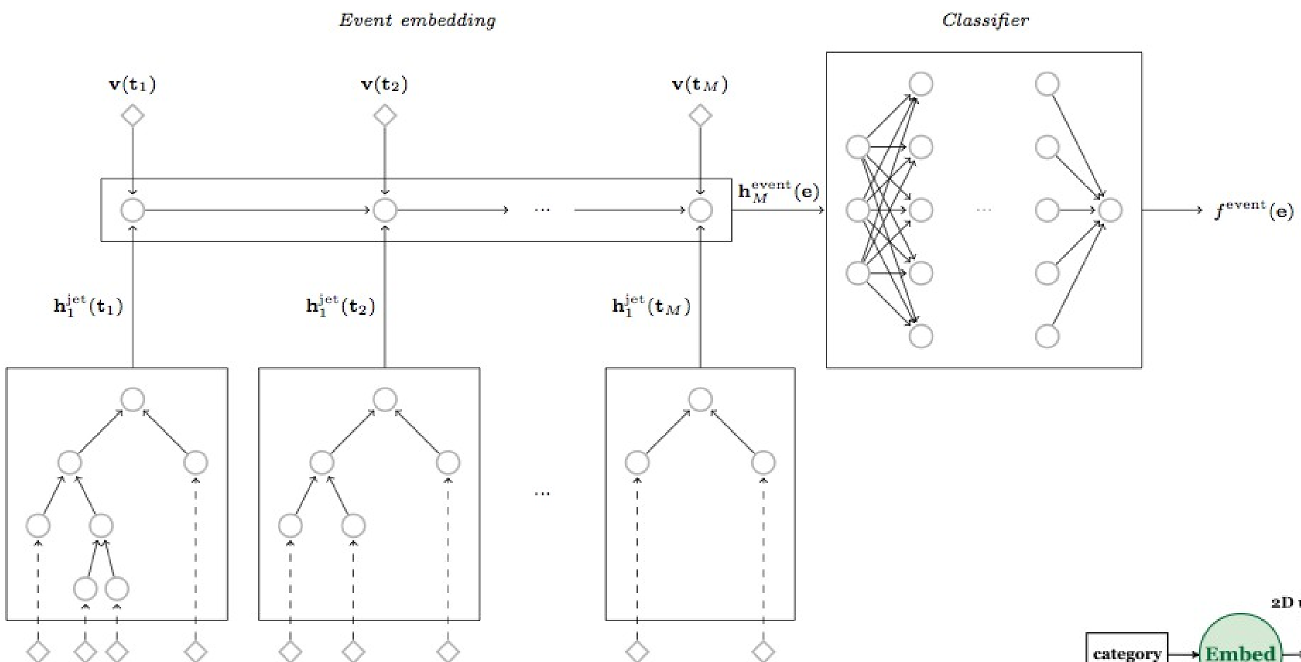
$$B(\mathbf{v}) = \begin{bmatrix} \gamma & -\gamma\beta n_x & -\gamma\beta n_y & -\gamma\beta n_z \\ -\gamma\beta n_x & 1+(\gamma-1)n_x^2 & (\gamma-1)n_x n_y & (\gamma-1)n_x n_z \\ -\gamma\beta n_y & (\gamma-1)n_y n_x & 1+(\gamma-1)n_y^2 & (\gamma-1)n_y n_z \\ -\gamma\beta n_z & (\gamma-1)n_z n_x & (\gamma-1)n_z n_y & 1+(\gamma-1)n_z^2 \end{bmatrix} \qquad (4)$$

$$X = \begin{bmatrix} c\,t \\ x \\ y \\ z \end{bmatrix} , X' = wB(\mathbf{v})X \qquad (5)$$

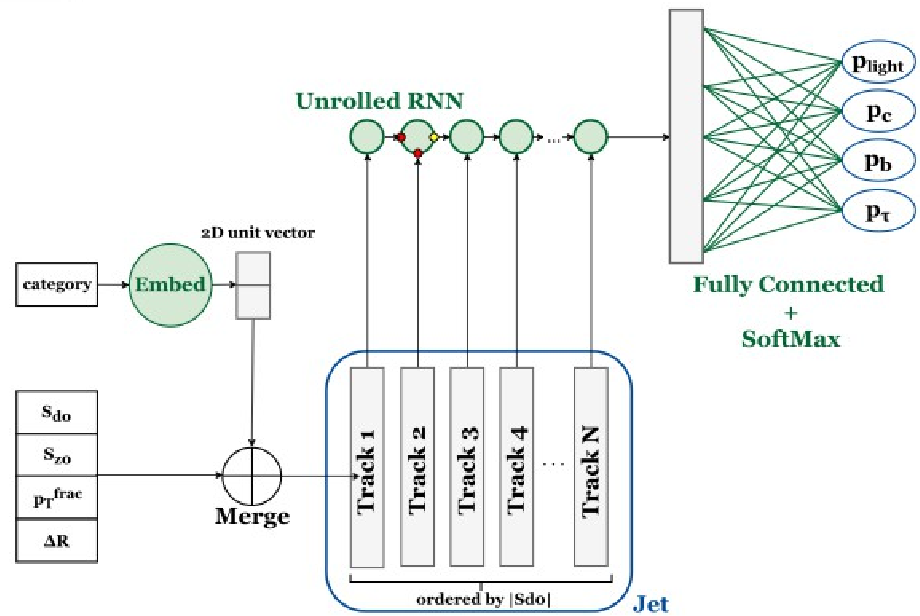


- Full event classification using reconstructed particle 4-vectors
- Recurrent neural nets, Long short term memory cells
- Dedicated layer with Lorentz boosting
- Step toward event classification with lower level data : low level feature as opposed to analysis level variables

Recurrent/Recursive Networks

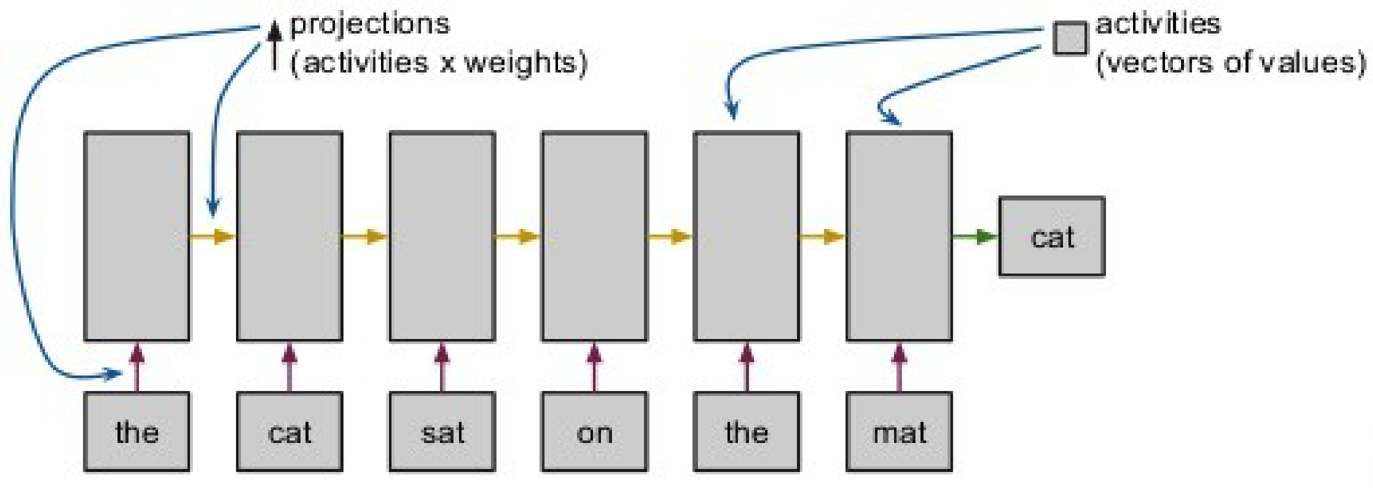


QCD-Aware Recursive Neural Networks for Jet Physics.
Louppe, Cho, Becot, Cranmer
https://arxiv.org/abs/1702.00748

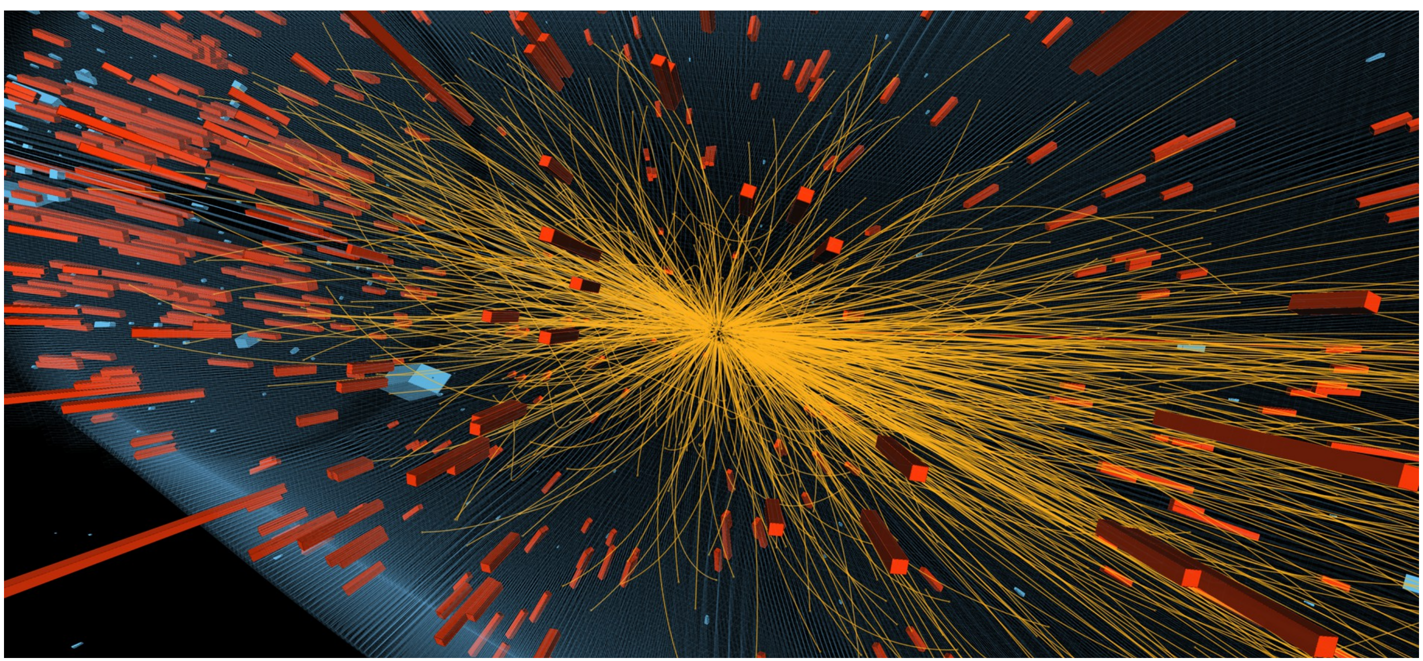


Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment
http://cds.cern.ch/record/2255226

Challenge in Natural Ordering



Text have natural order. RNN/LSTM can correlate the information to internal representation

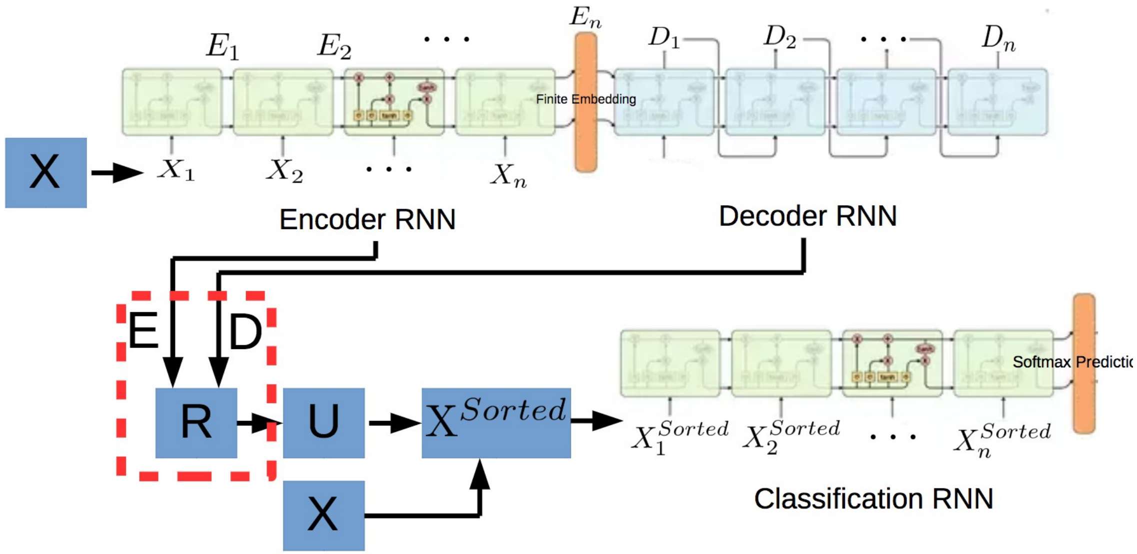


There is underlying order in collision events. Smeared through timing resolution. No natural order in observable

- **Learn how to sort**

Learn How To Sort

Recurrent Neural Networks (RNNs):
-Long Short-Term Memory (LSTM)
-Gated Recurrent Unit (GRU)



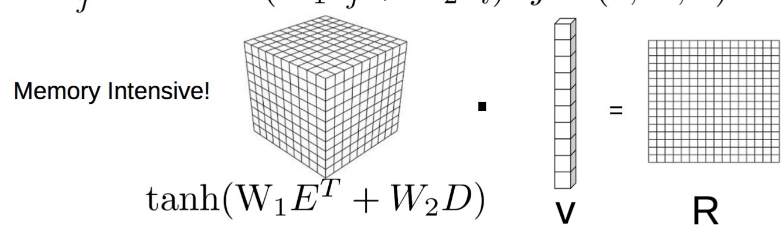
- Custom:

$$R = W1 \begin{bmatrix} — & E_1 & — \\ \vdots & \vdots & \vdots \\ — & E_n & — \end{bmatrix} + W2 \begin{bmatrix} | & \cdots & | \\ D_1 & \cdots & D_n \\ | & \cdots & | \end{bmatrix}$$

Where W_1, W_2 are trainable (nxn) matricies

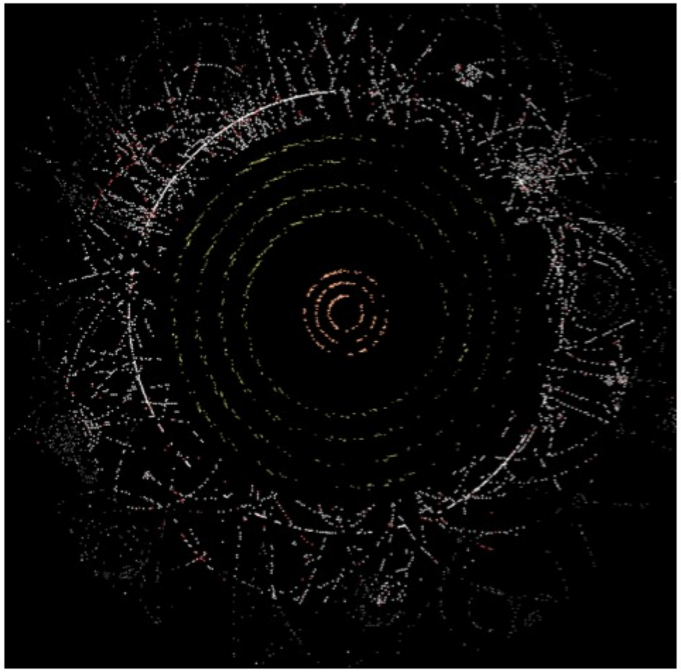
- Ptr_Net(https://arxiv.org/pdf/1506.03134.pdf)

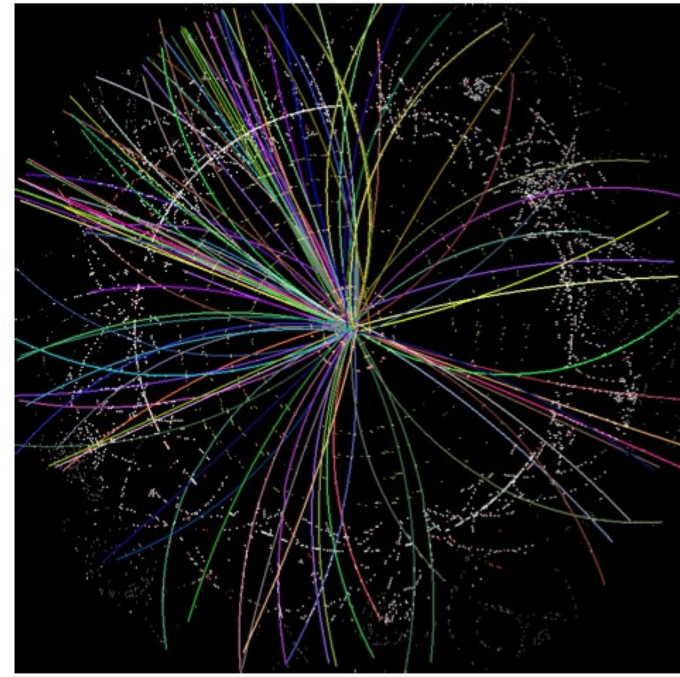
$$\mathrm{R}_j^i = v^T tanh(W_1 e_j + W_2 d_i) \quad j \in (1, ..., n)$$



Sorting and “soft” sorting models can be concurrently trained with recurrent networks
Expensive and tricky to train

Charged Particle Tracking

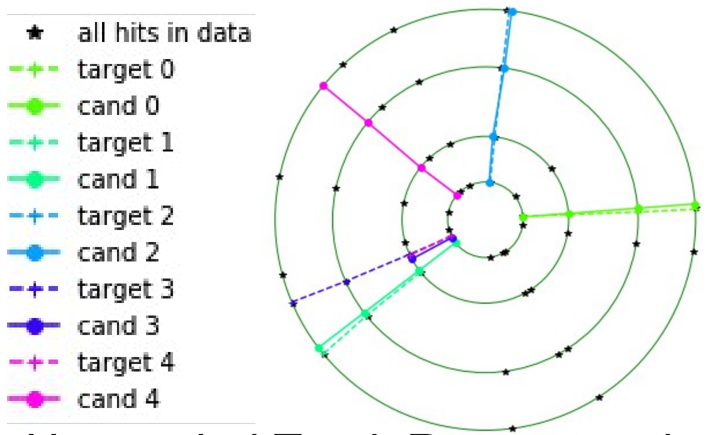




- Perfect example of pattern recognition
- Data sparsity is not common in image processing
- Several angles to tackle the problem. Deep Kalman filter, RNN to learn dynamics, sparse image processing, ...
- Kaggle challenge in preparation

HEP.TrkX Project

https://heptrkx.github.io/



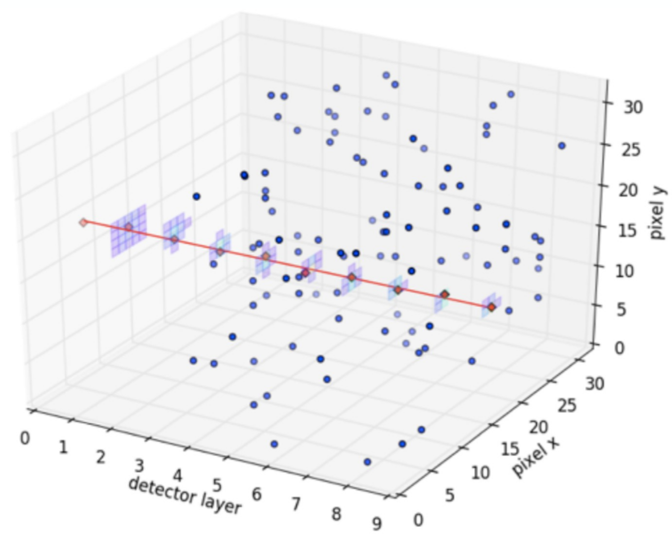
Un-seeded Track Reconstruction

Pilot project funded by **DOE ASCR** and **COMP HEP**. Part of **HEP CCE**.

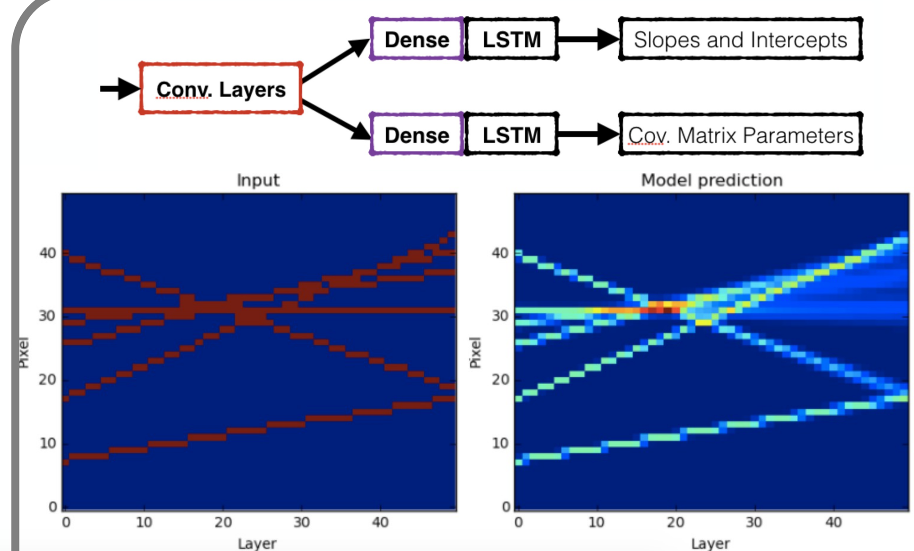
LBNL, Fermilab, Caltech consortium

→ Mission

Explore deep learning techniques for charged particle track reconstruction



Seeded Track Reconstruction



Track Parameter Estimation

Conference Talks

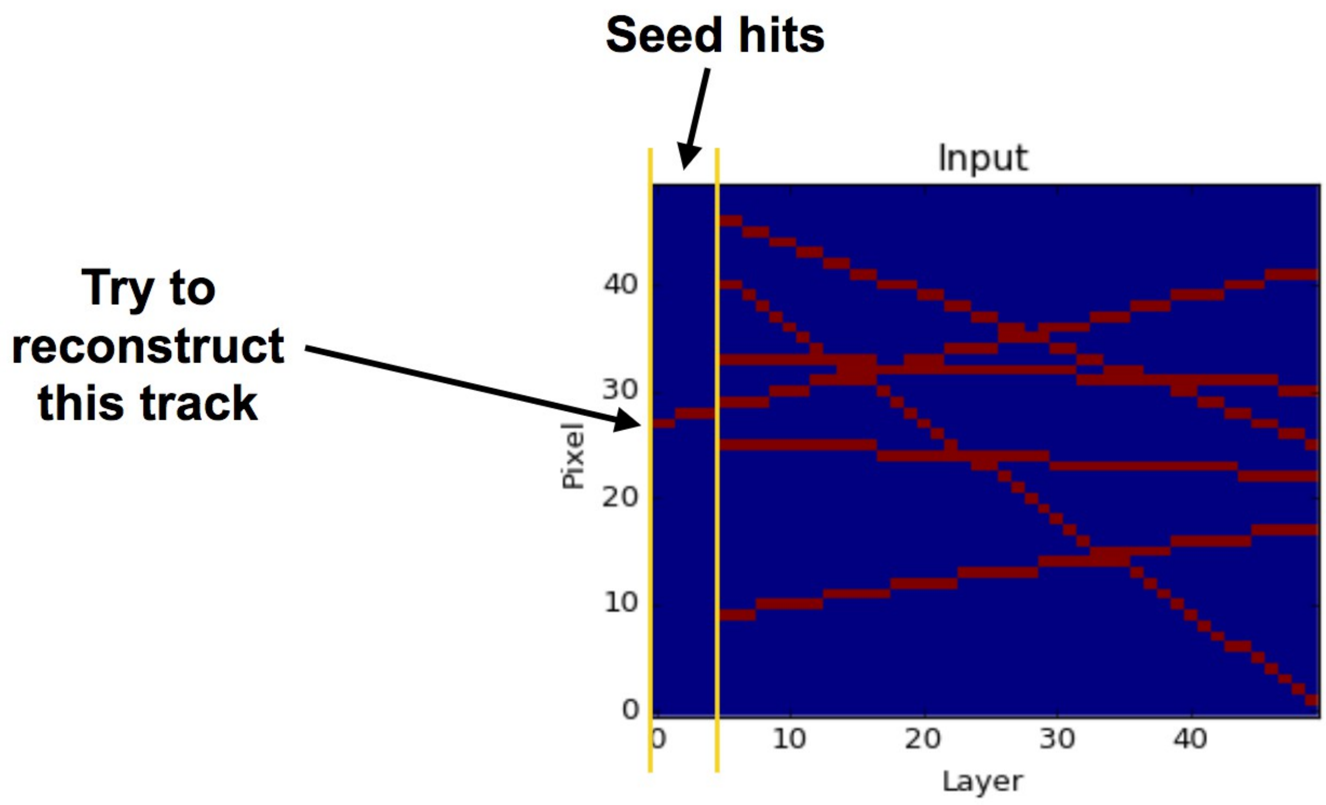
https://indico.cern.ch/event/577003/contributions/2476580/

https://erez.weizmann.ac.il/pls/htmldb/f?p=101:58:::NO:RP:P58_CODE,P58_FILE:5393,Y

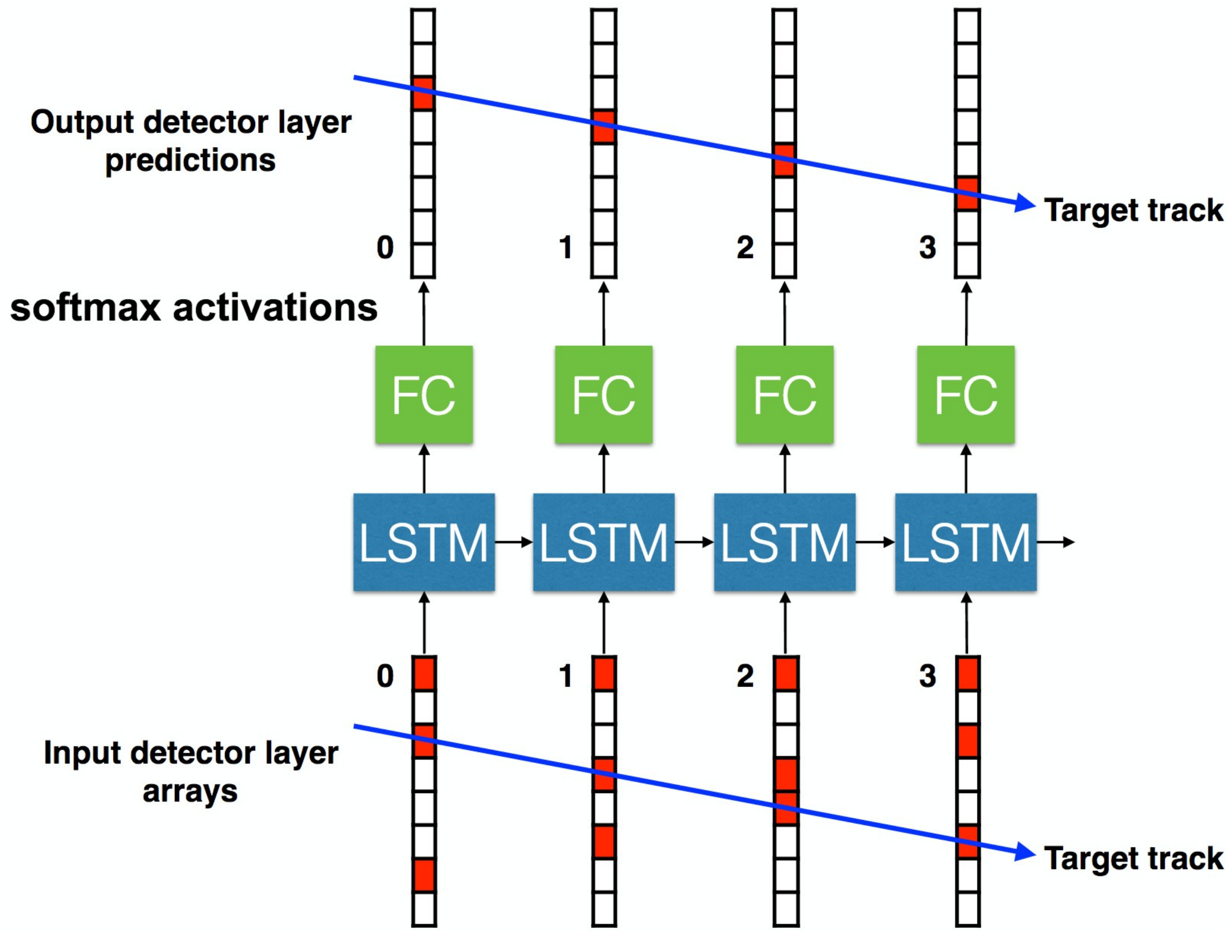
https://indico.cern.ch/event/567550/contributions/2629737/

Seeded Pattern Prediction

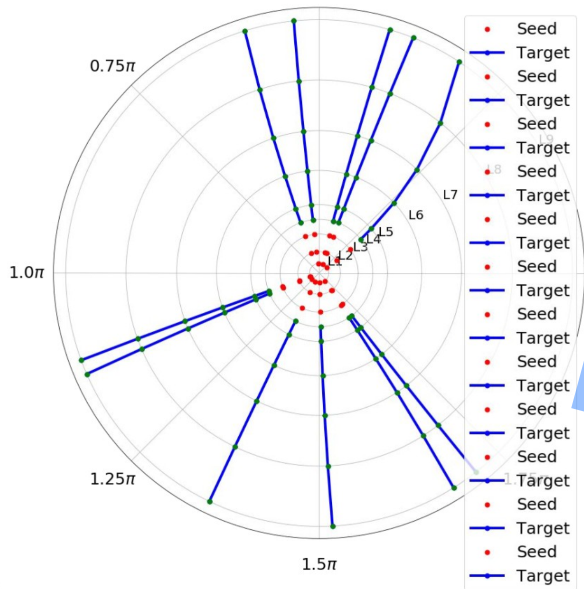
- Hits on first 3 layers are used as seed
- Predict the position of the rest of the hits on all layers

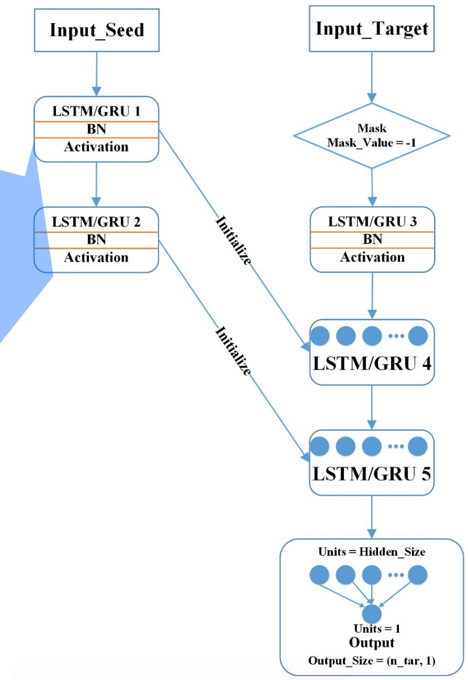


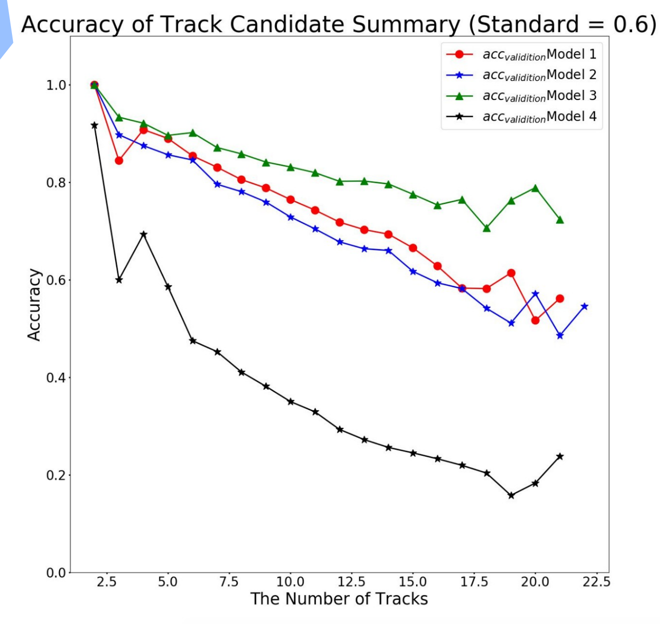
LSTM ≡ Kalman Filter



Un-binned Seeded Tracking







Hit position input not as an image but as a sequence of positions

Overcomes the scalability issue

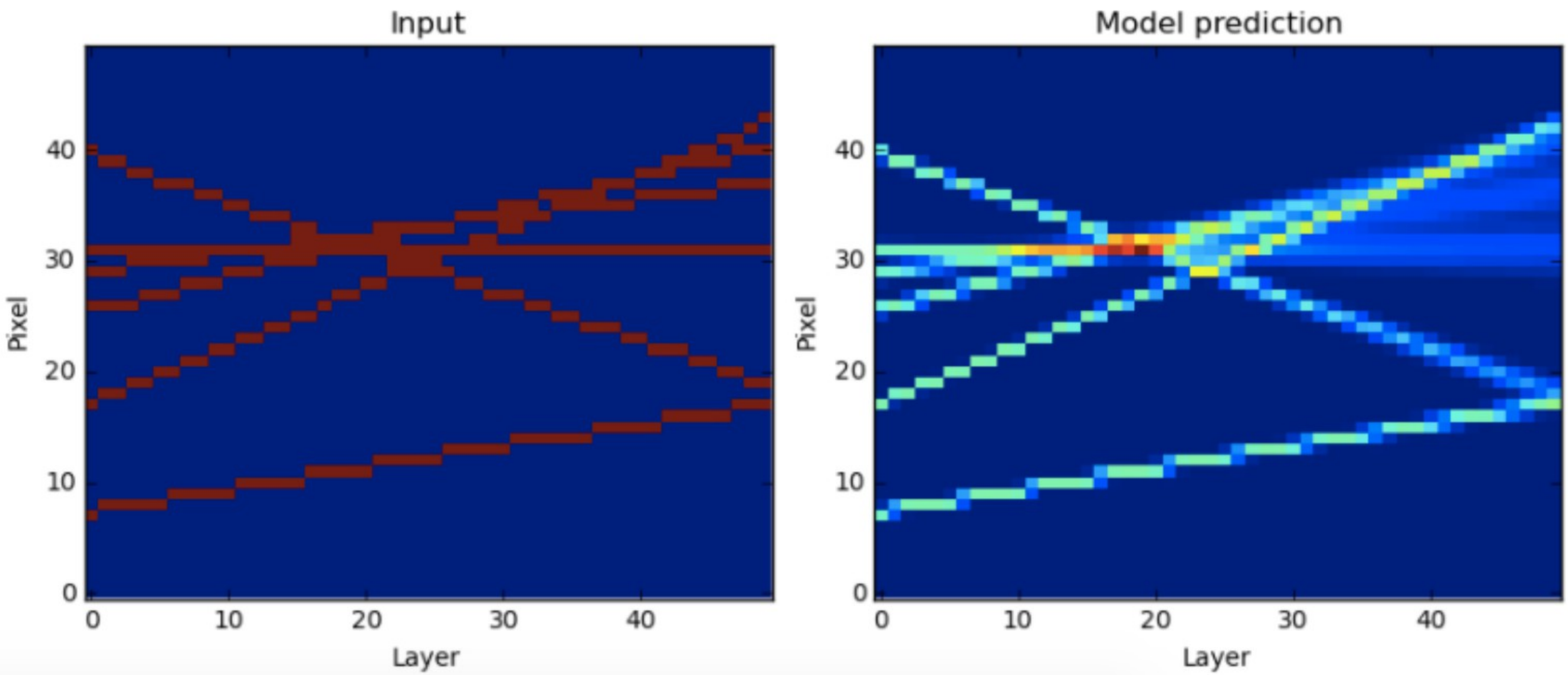
Analog to Kalman Filter approach

Prediction Track Covariance

Model is modified to predict a covariance matrix for which there is no ground truth, but is used with the modified loss function

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

Track Parameters Uncertainty



Representation of track slope, intersect and respective uncertainties

Hopfield Network

$$E = -\frac{1}{2}\left(\sum_{i,j} w_{ij} S_i S_j - 2\sum_i \theta_i S_i\right).$$

- Not a neural network per say
- Fully-connected graph
- Connections pruned based on an enery minimisation model

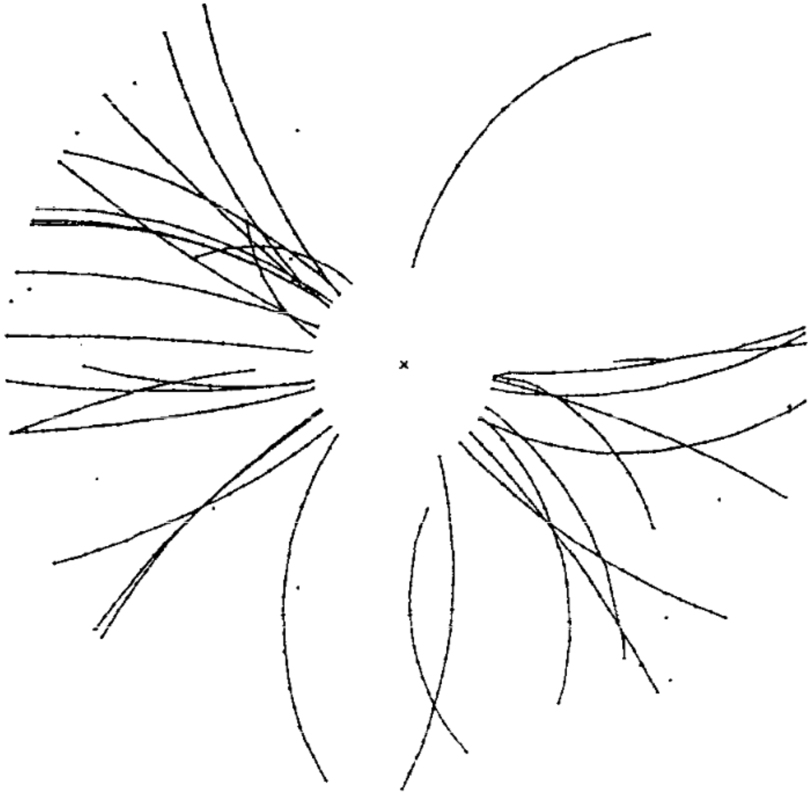
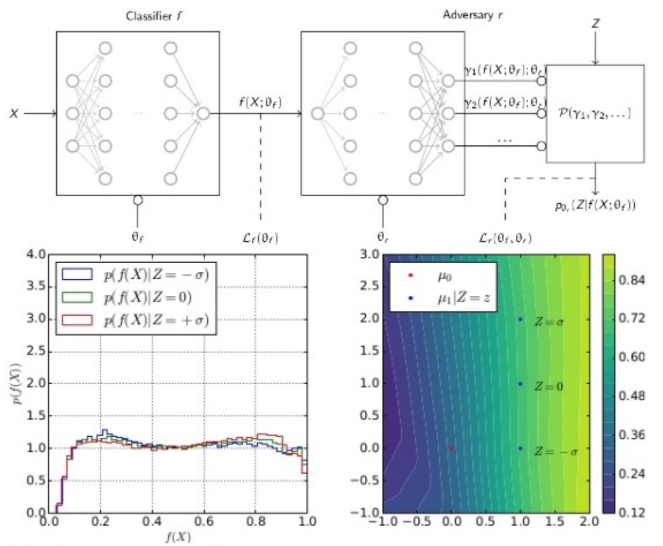


Fig. 4. Tracks in the ALEPH TPC reconstructed with a Hopfield net [13].

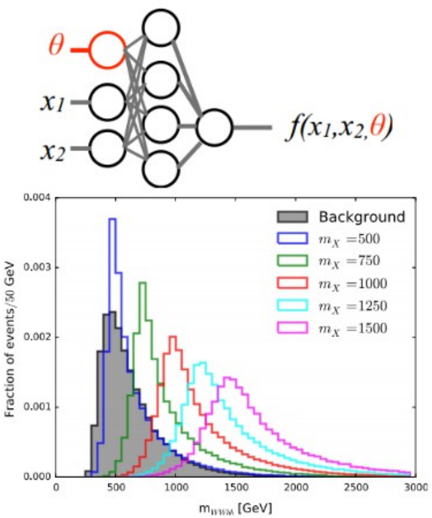
https://link.springer.com/chapter/10.1007/3-540-61510-5_1

Learn With Uncertainty

- Despite the precision of the SM, we still have to deal with:
  - statistical uncertainties (inherent fluctuations)
  - systematic uncertainties (the known unknowns of the model)
- Uncertainty is usually formulated as nuisance parameters $\boldsymbol{\nu}$.



With adversarial training, force the model to be independent of $\boldsymbol{\nu}$.



Add $\boldsymbol{\nu}$ as an input to the model and profile it out later.

Learning to Pivot with Adversarial Networks
G. Louppe, M. Kagan, K. Cranmer
https://arxiv.org/abs/1611.01046

Parameterized Machine Learning for High-Energy Physics
P. Baldi, K. Cranmer, T. Faucett, P. Sadowski, D. Whiteson
https://arxiv.org/abs/1601.07913

Probabilistic Programming

Frank Wood, Atilim Gunes Baydin, TTIC, OXFORD

A Probabilistic *Program*

"Probabilistic programs are usual functional or imperative programs with two added constructs:

(1) the ability to draw values at random from distributions, and

(2) the ability to **condition** values of variables in a program via observations."

Success Stories

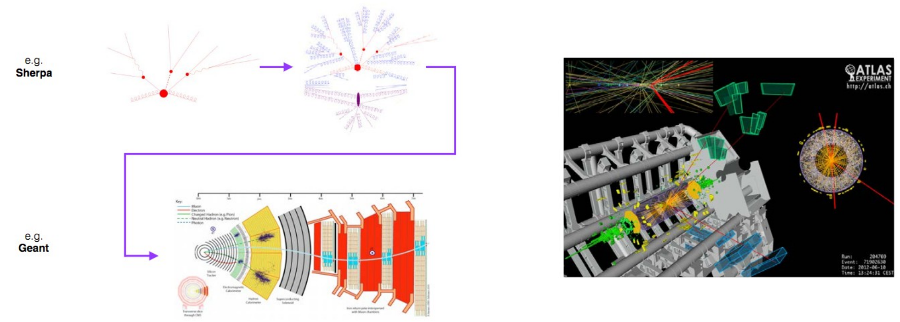
Graphical Models

BUGS STAN

Factor Graphs

Factorie Infer.NET

New Physics via Standard Model + ATLAS Simulator Inversion



x	**y**
event & detector simulators	ATLAS detector output

23

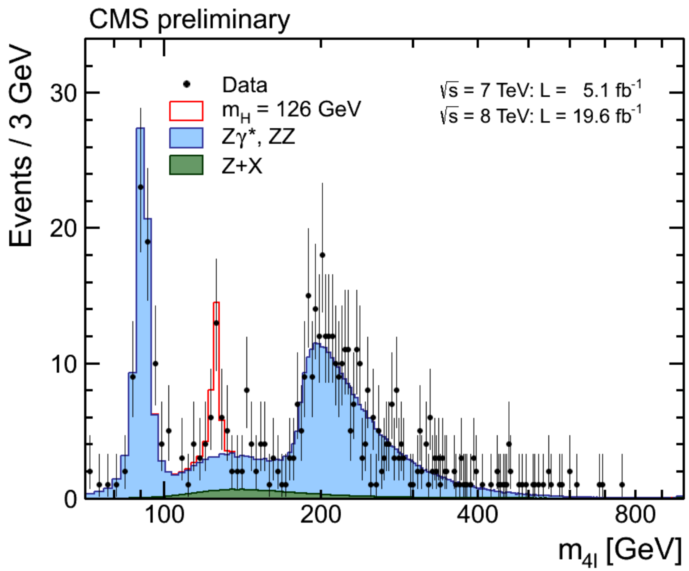
- Dense and exhaustive talk
- Learn how to control a simulator to provide a given output

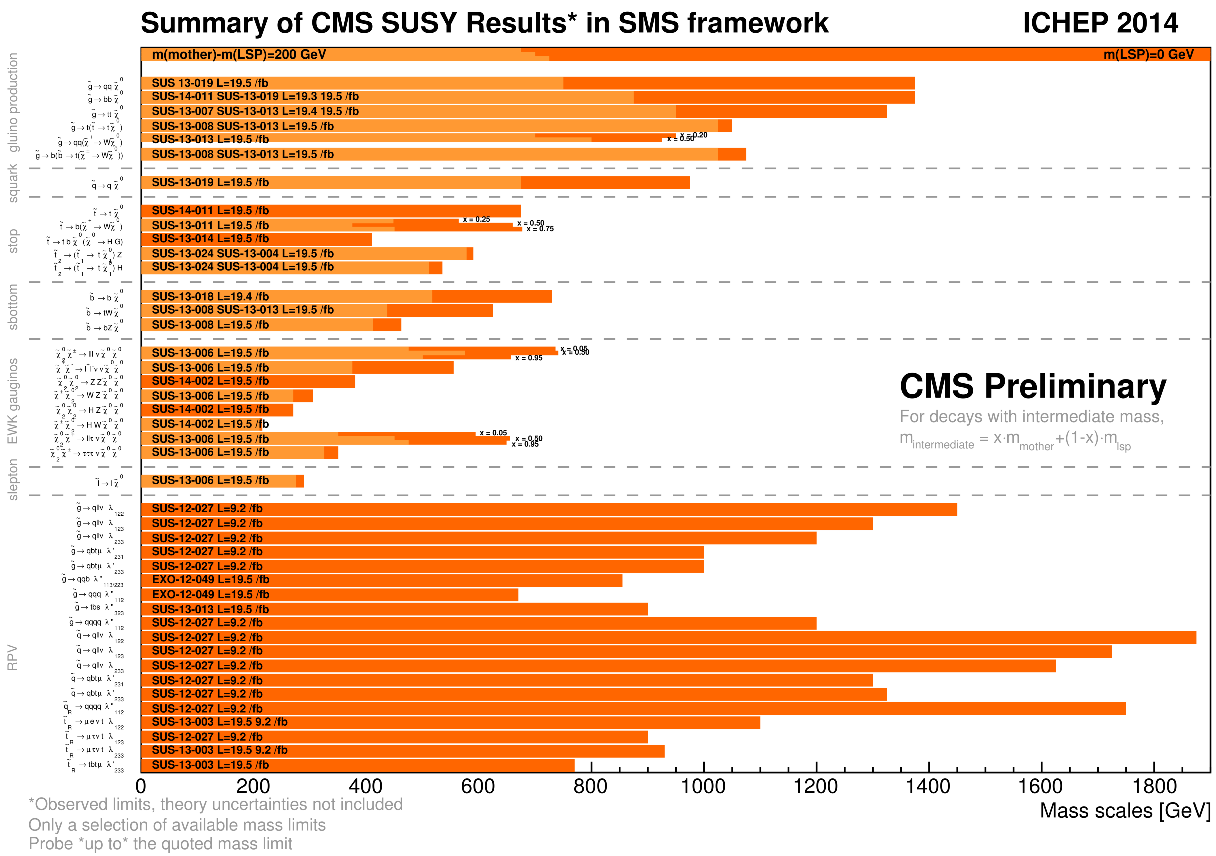
Other Applications

- Outliers selection
- Anomaly detection
- Data quality automation
- Detector control
- Experiment control
- Data popularity prediction
- Computing grid control
- Denoising with auto-encoder
- ...

Search For New Physics

Higgs discovery : we knew what it would look like



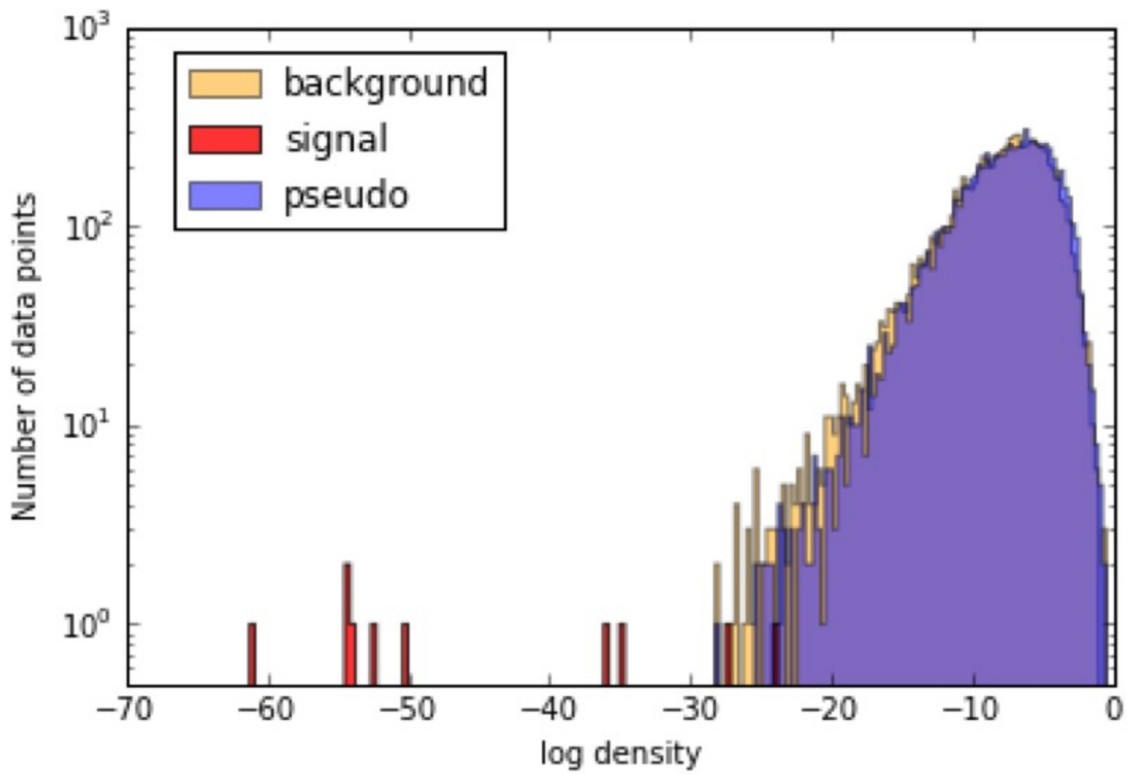


New physics searches (Susy, ...) : we don't know what to expect.

→ **Unsupervised machine learning**

Outlier Identification

- Train a NADE (https://arxiv.org/abs/1306.0186) model on mixture of the known backgrounds
- Use a synthetic dataset with small injected signal
- Log density singles out the injected signal

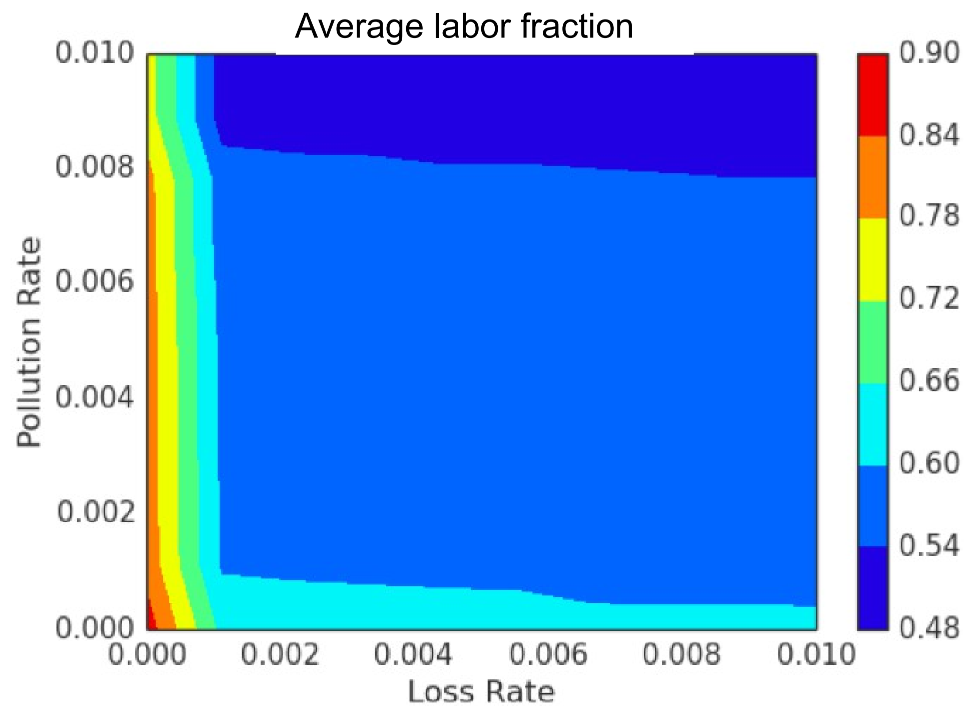


Anomaly Learning

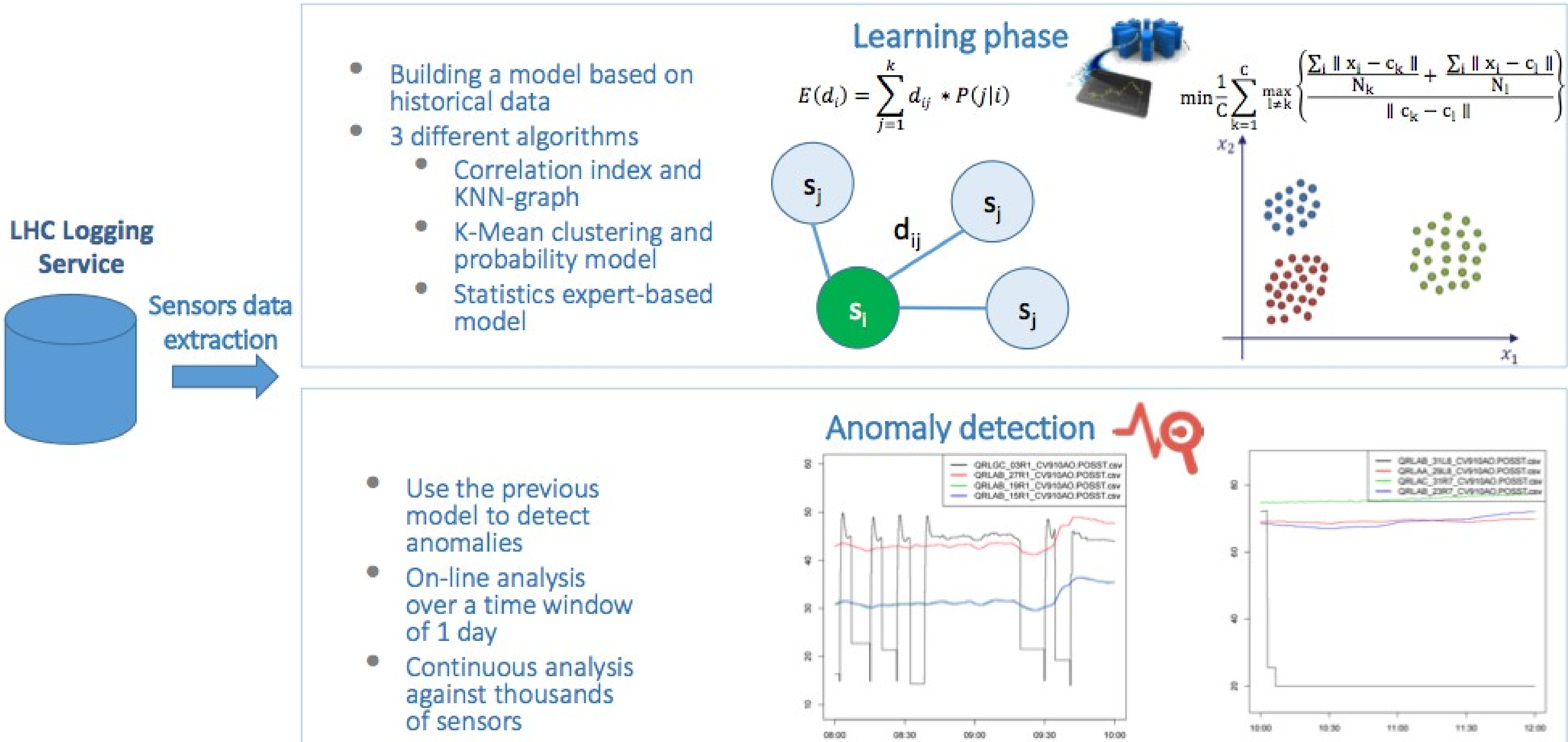
- Not 100% of the data taken at the experiements are good for analysis (detector effect, calibration, software defect, …)
- Luminosity block ≡ 23s of beam
- Histograms made per luminosity block are scrutinized by experts to decide on good/bad data
- Several layers of scrutiny, labor intensive

- The machine learning approach
  - Identifies relevant features
  - Calculates percentile per lumiblock
  - Trains rolling classifiers

- **Accepting 1% data loss could save 40% of the workload on the certification team**



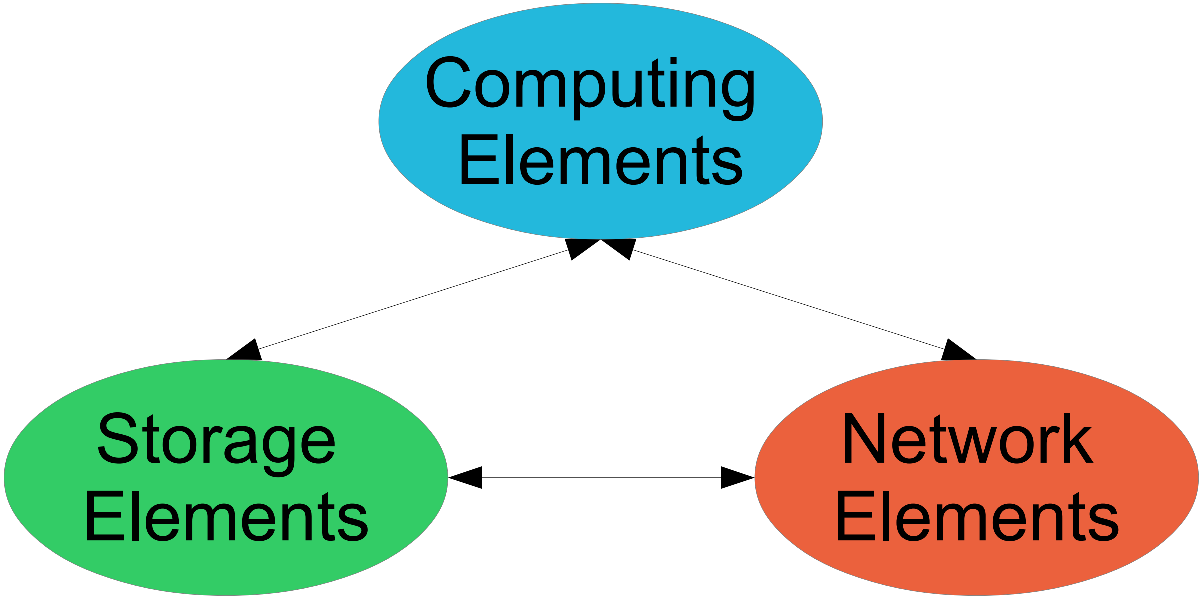
Cryogenic Anomaly Detection



- Project from the LHC cryogenic team

https://indico.cern.ch/event/514434/

GRID Echosystem



- Optimization of each component independently might not lead to the global optimum
- Need to consider the system as a whole

- Model single element metrics with deep learning
- Reinforcement learning to control the system's components

Accelerating and Emerging Technologies

Caltech iBanks Cluster

Caltech GPU Servers

- 4 compute nodes :
  Intel® Xeon® CPU with **NVIDIA® TITAN (1x2)**, **GTX 1080 (2x8)**, **TITAN-X (1x8)**
- 1 head node for login, jupyterhub, home directory, nfs, www.
- 1 shared disk server (**20TB**) over **10GBs** NICS

➔ Partnering vendors/donators supermicro, cocolink, dell, intel, nvidia

➔ **Prototyping** and **small scale** training

ALCF

Cooley

- 126 compute nodes :
  Two 2.4 GHz **Intel® Haswell® E5-2620** v3 processors per node (6 cores per CPU, 12 cores total) and **NVIDIA® Tesla® K80**
- Theoretical Peak Performance :
  293 Tflops

➔ **Development Project** with 8k core hours

CSCS Piz Daint

Piz Daint

- 5272 compute nodes : **Intel® Xeon® E5-2690** @ 2.60GHz (12 cores, 64GB RAM) and **NVIDIA® Tesla® P100**
- Theoretical Peak Performance : 10 **Pfops**

→ **Allocation** 9k node-hours

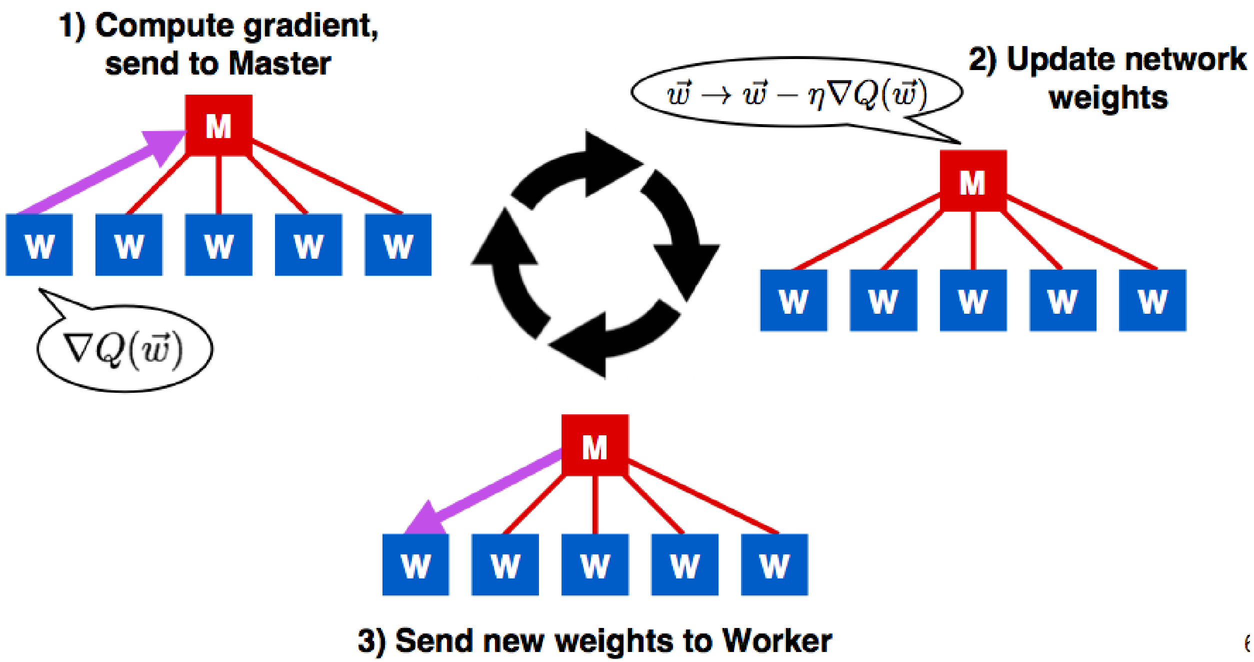
OLCF

Titan

- 18688 compute nodes : **2.2GHz AMD® Opteron® 6274** processors per node (16 cores per CPU) and **NVIDIA® Tesla® K20X**
- Theoretical Peak Performance : **20 Pflops**

→ **Allocation** 2M node-hours

Distributed Learning



- Deep learning with elastic averaging SGD https://arxiv.org/abs/1412.6651
- Revisiting Distributed Synchronous SGD https://arxiv.org/abs/1604.00981
- Implementation with Spark and MPI for the Keras framework https://keras.io/
  - https://github.com/JoeriHermans/dist-keras
  - https://github.com/duanders/mpi_learn

Motivation

- Prototyping and training with keras (http://keras.io/)
- Use of GP-GPU can significantly speed up training of deep or not so deep neural net
  - ➔ A typical 10x
- Training of large model on large dataset can still take several days to convergence on single GPU
- Even more painful in case of scanning or tuning of hyper-parameter

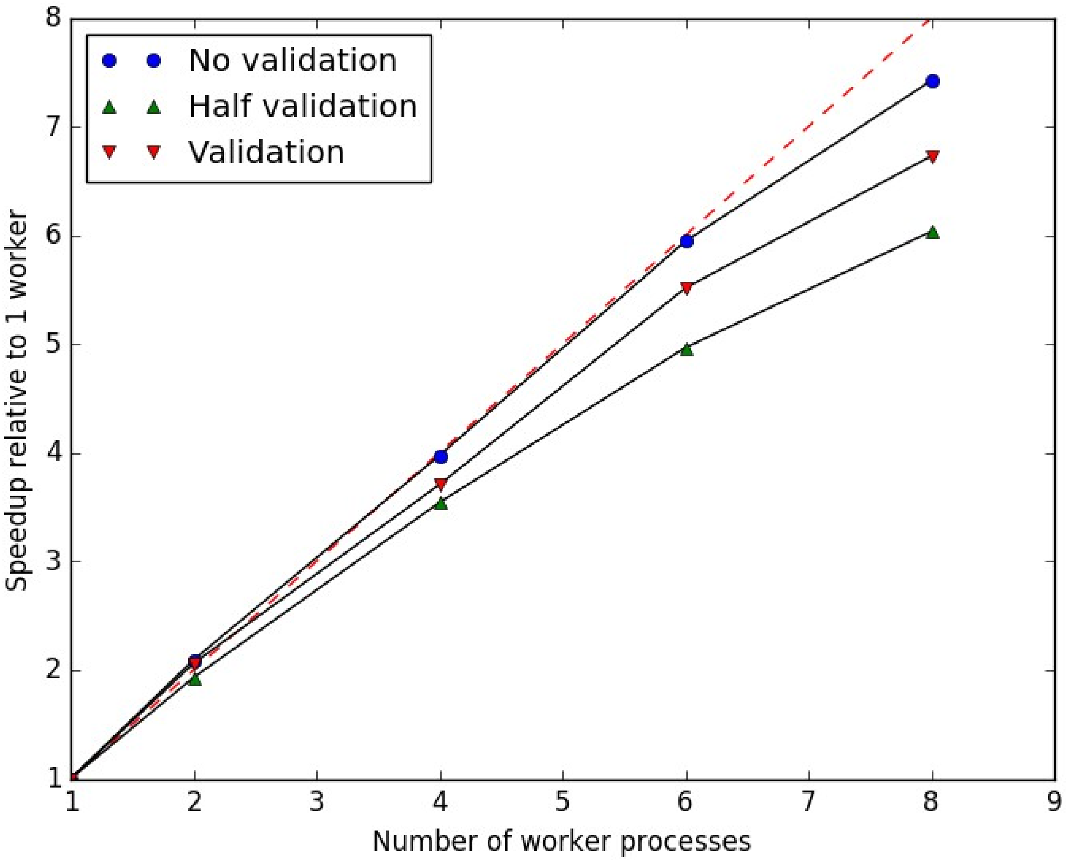
➔ Speed up can be obtained
  - ✓ Data parallelism for large dataset (strong scaling)
  - Model parallelism for large model (weak scaling)

Training Speed-up

- Benchmark on single server with 8 GPUs

- MPI spawns workers on different cores
- Each core is instructed to use a different GPU

→ **Speed-up is quasi-linear with number of GPU**
→ **7x on a single server**



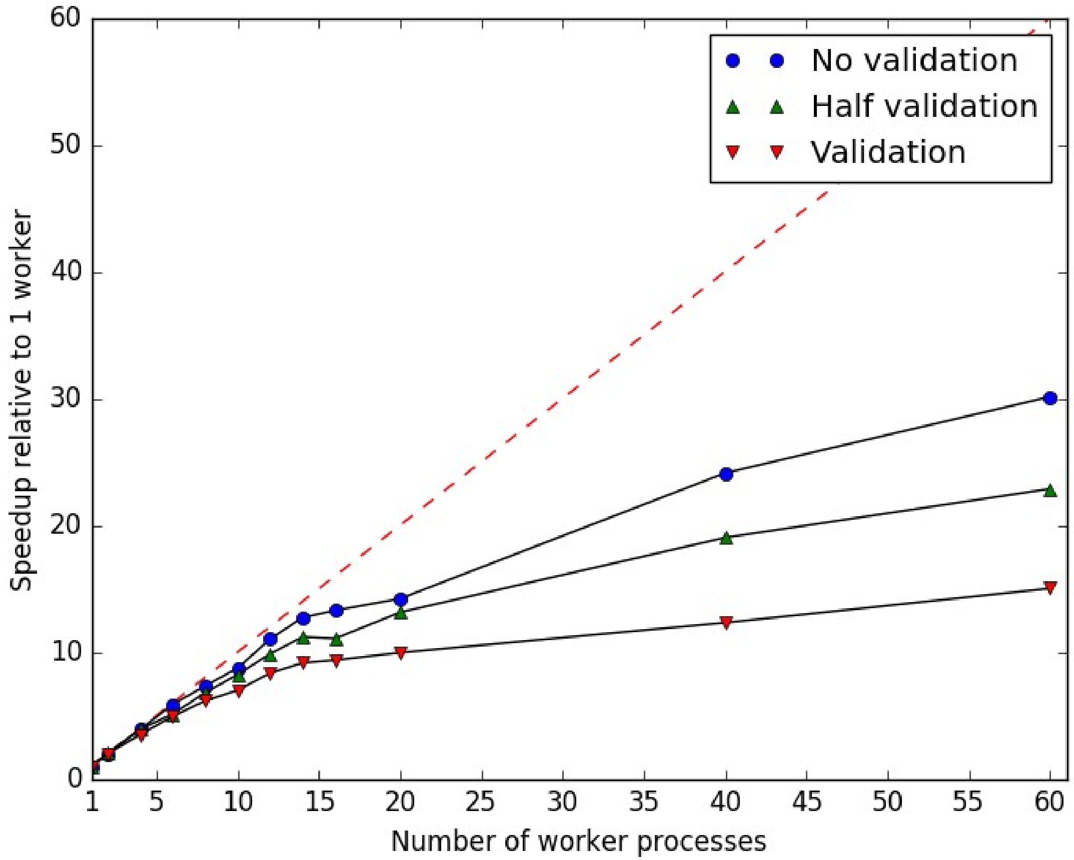
Speed-up Scaling

- Benchmark on Cooley GPU cluster at ALCF

- MPI spawns workers on different nodes
- Each node uses its GPU

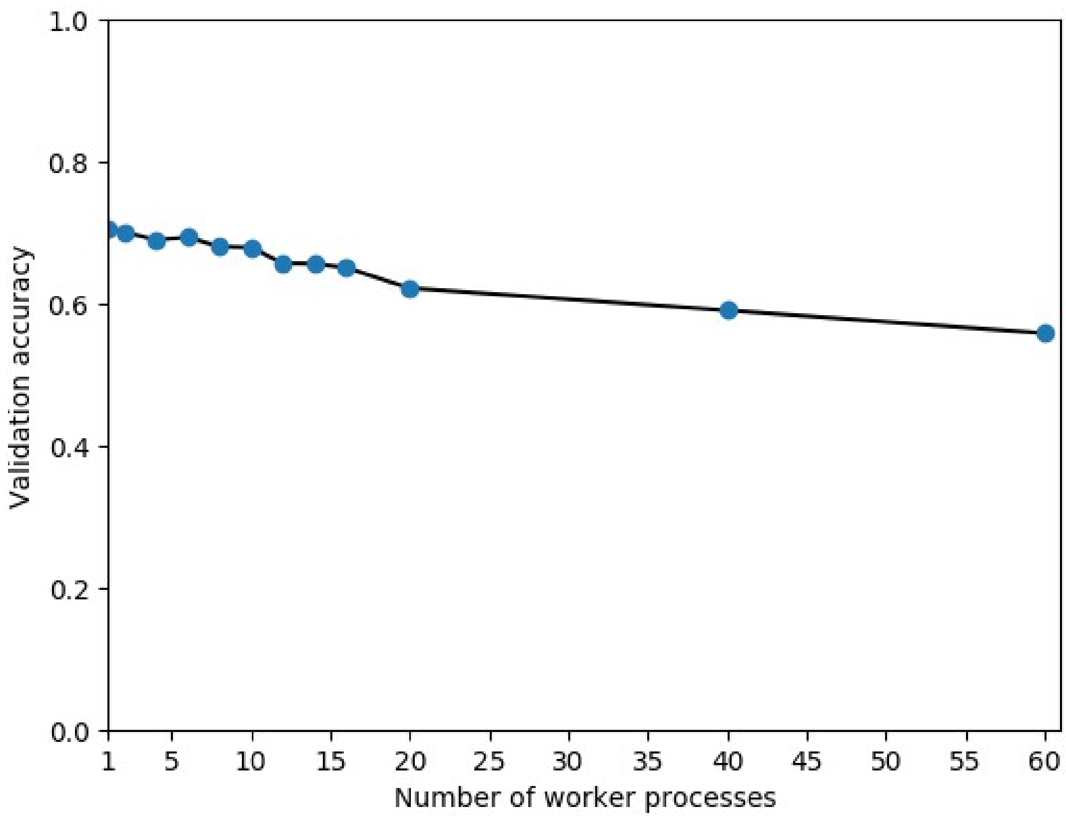
→ **Speed-up is quasi-linear up to ~15 GPU**
→ **Loss in scaling above**
→ **30x using 60 nodes**

✗ Scaling still to be understood



Stale Gradients

Validation accuracy after a fixed number of epochs of training

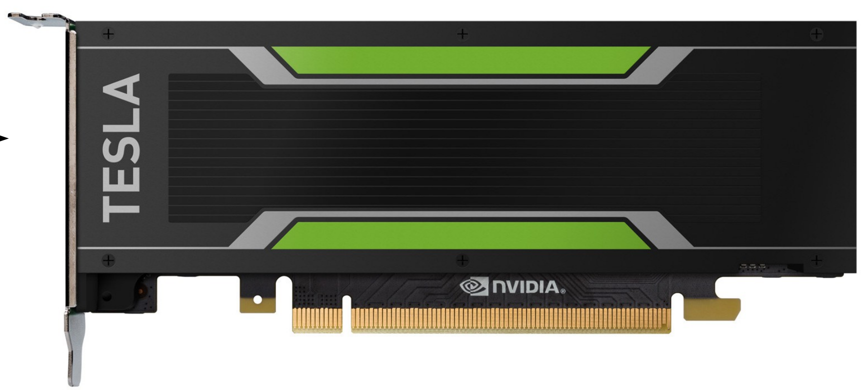


- Workers end up producing gradients from outdates weights
- Slow down of the convergence with larger number of workers
- Effect can be mitigated with tuning of momentum https://arxiv.org/abs/1606.04487

Hardware Consideration

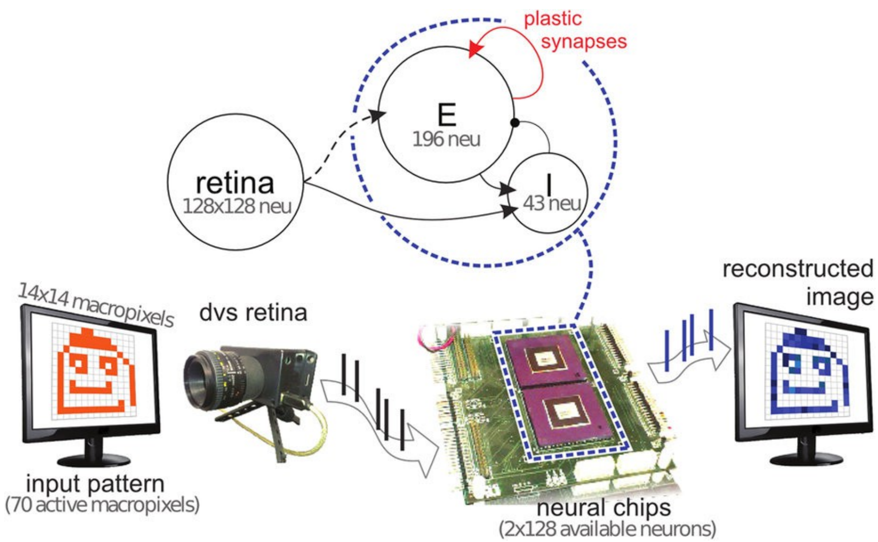
Training vs Inference

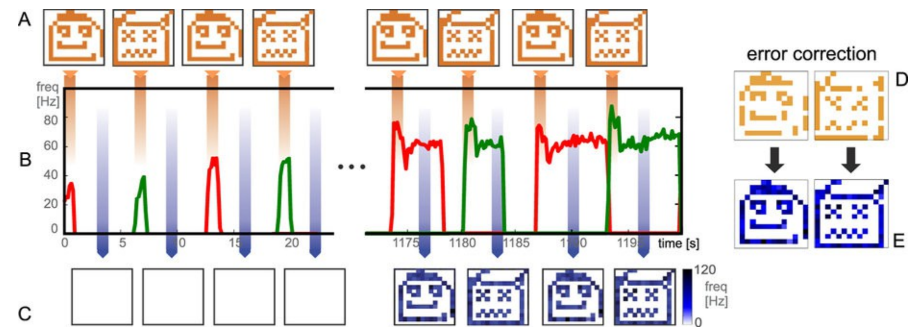
- GPUs are the workhorse for parallel computing
- Enable training large models, with large dataset
- **Deep learning facility clusters**



- Emergence of smaller GPU
- Not dedicated to training
- Strike the balance between Tflops/$ for inference
- **Deployment on the grid**

Neuromorphic Hardware

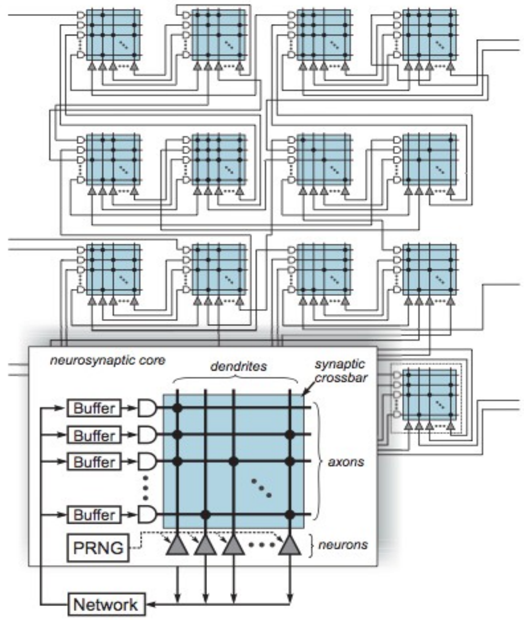


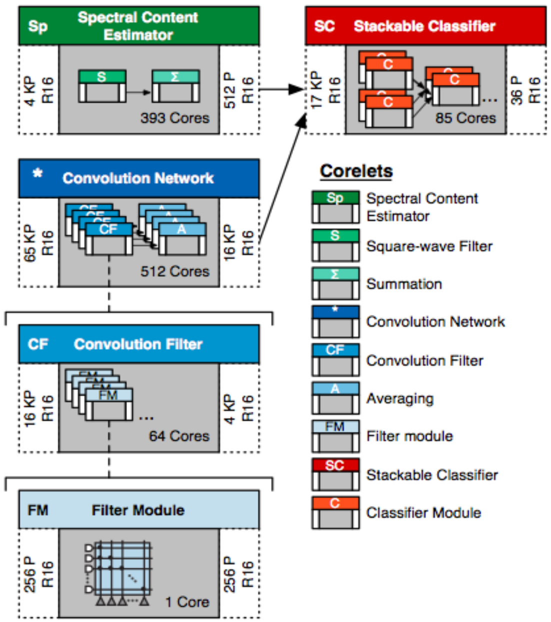


http://www.nature.com/articles/srep14730

- Implementing plasticity in hardware
- Process signal from detector and adapt to categories of pattern (unsupervised)
- Post-classified from data analysis or rate throttling
- ➢ NCCR consortium assembling to develop this technology further, with our use case in mind

Cognitive Computing





- Spiking neural net as processing units :
  ➔Cognitive Computing Processing Unit : CCPU
- Adopt a **new programming scheme**, translate existing software
- See Rebecca Carney's talk for more details