Machine Learning in Particle Physics (I) – Introduction

Wolfgang Waltenberger
ÖAW and Uni Wien

ESI Winter School, Vienna,
Feb 10 - 20, 2020

Disclaimer

- Gregor, Lisa and I are all (mostly) LHC physicists. Within these lectures we will thus keep a strong focus on the context of the LHC.

- You may find all my material for this winterschool at

  https://github.com/WolfgangWaltenberger/winterschool

- We will use google's colab for the tutorial

- On the github page given above of the repo you will find a link to our google colab instance

Syllabus

For today and tomorrow, we aim to put our focus on

- generative models
- decorrelating variables
- outlier detection
- uncertainties

Wednesday	Thursday
9:00 Introductory Lecture, Detailling our Use Cases and Problems	**9:00** Generative Models and Outlier Detection (I)
11:00 Uncertainties, Decorrelating Variables	**11:00** Generative Models and Outlier Detection (II)
14:00 Tutorial, Uncertainties & Decorrelating Variables	**14:00** Tutorial, Generative Models and Outlier Detection

Particle Physics and Machine Learning – Lecture I:

Use Cases, Needs, and Challenges

Theory

Machine Learning in Theoretical High Energy Physics

$$
\begin{aligned}
\mathcal{L} = & -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} \\
& + i \bar{\psi} \not{D} \psi + h.c. \\
& + \bar{\psi}_i y_{ij} \psi_j \phi + h.c. \\
& + |D_\mu \phi|^2 - V(\phi)
\end{aligned}
$$

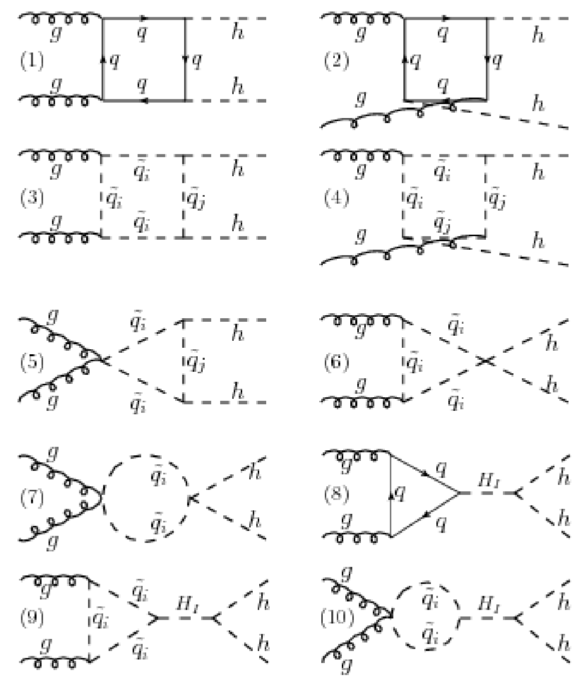
Machine learning in Theoretical Particle Physics

Obviously theoretical particle physics is all about performing many complex (e.g. quantum field theoretical) calculations. It is thus very natural that machine learning applications arose recently. Let me mention a few subjectively selected works:

- a supervised fully-connected network that interpolates between quantum field theoretical calculations to speed up the process
- A generative model to create simulations of quark-gluon plasmas
- Unsupervised learning techniques to identify “classes” of string theories that possibly contain the particle content of the Standard Model

“Deep Cross Sections”

Problem: LHC theorists have to compute a very large number of quantum field theoretical processes for many similar e.g. supersymmetric theories – a complicated, tedious, and repetitive task.



Solution: use a neural network to interpolate between the solutions that you computed “manually”.

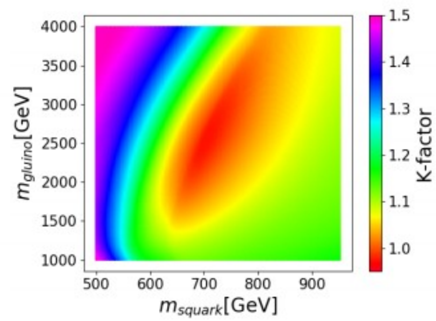


FIG. 1: A temperature plot for the K-factor in the wino scenario, predicted by a neural network, for $\tilde{\chi}_2^0\tilde{\chi}_1^{\pm}$ in the $m_{\tilde{q}}/m_{\tilde{g}}$ plane, already showing a non-trivial K-factor landscape for two free parameters. The electroweakino masses are set to 400 GeV.

DeepXS: Fast approximation of MSSM electroweak cross sections at NLO

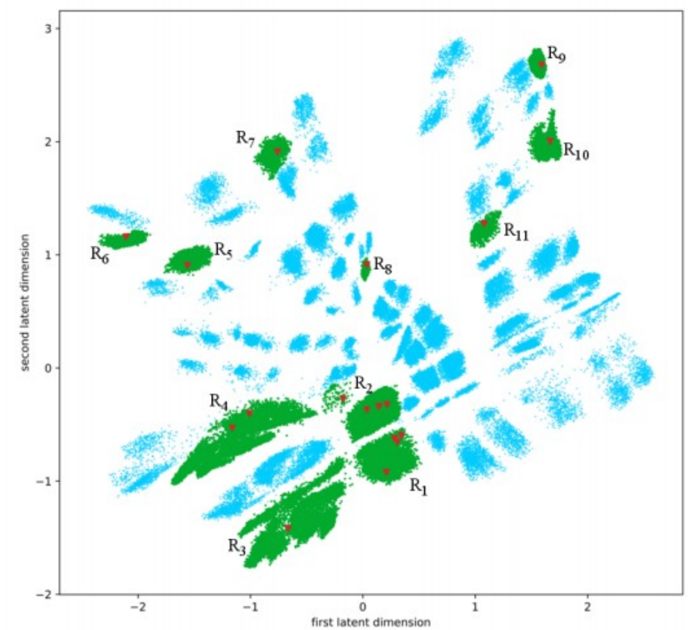
Sydney Otten,[1,2,*] Krzysztof Rolbiecki,[3,†] Sascha Caron,[1,4]
Jong-Soo Kim,[5,6] Roberto Ruiz de Austri,[7] and Jamie Tattersall[8,9]

https://arxiv.org/pdf/1810.08312.pdf

"Clustering string theories"

Problem: the landscape of string theories is vast, and only some pockets of it produce a particle spectrum that contains the known Standard Model particles. How to identify these pockets?

Solution: use e.g. an autoencoder to learn low-dimensional representations of string theories.
Use a clustering algorithm in latent space to find similar types of string theories that contain the Standard model particle content (or something similar)



gure 2: *The landscape of $\mathcal{O}(700{,}000)$ $\mathbb{Z}_6$-II models extracted from the autoencoder: Each point corresponds to a $\mathbb{Z}_6$-II model and MSSM-like models are highlighted as red triangles. It turns out that MSSM-like models populate eleven separated islands. We color these islands in green and label them by $R_1, \ldots, R_{11}$. In addition, all $\mathbb{Z}_6$-II models outside these islands are colored in blue and defined to live in the region R_0.*

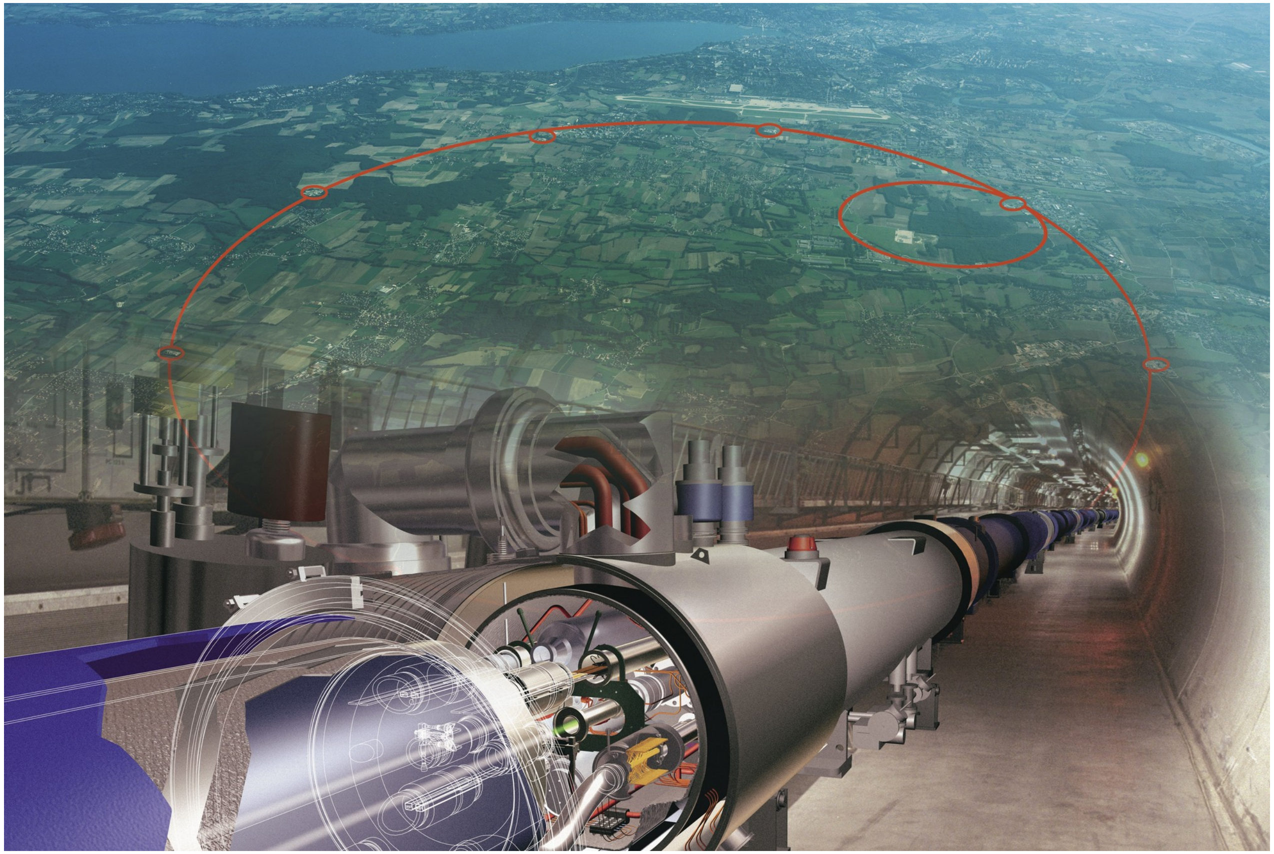
Deep learning in the heterotic orbifold landscape

Andreas Mütter,[1] Erik Parr,[2] Patrick K.S. Vaudrevange[3]

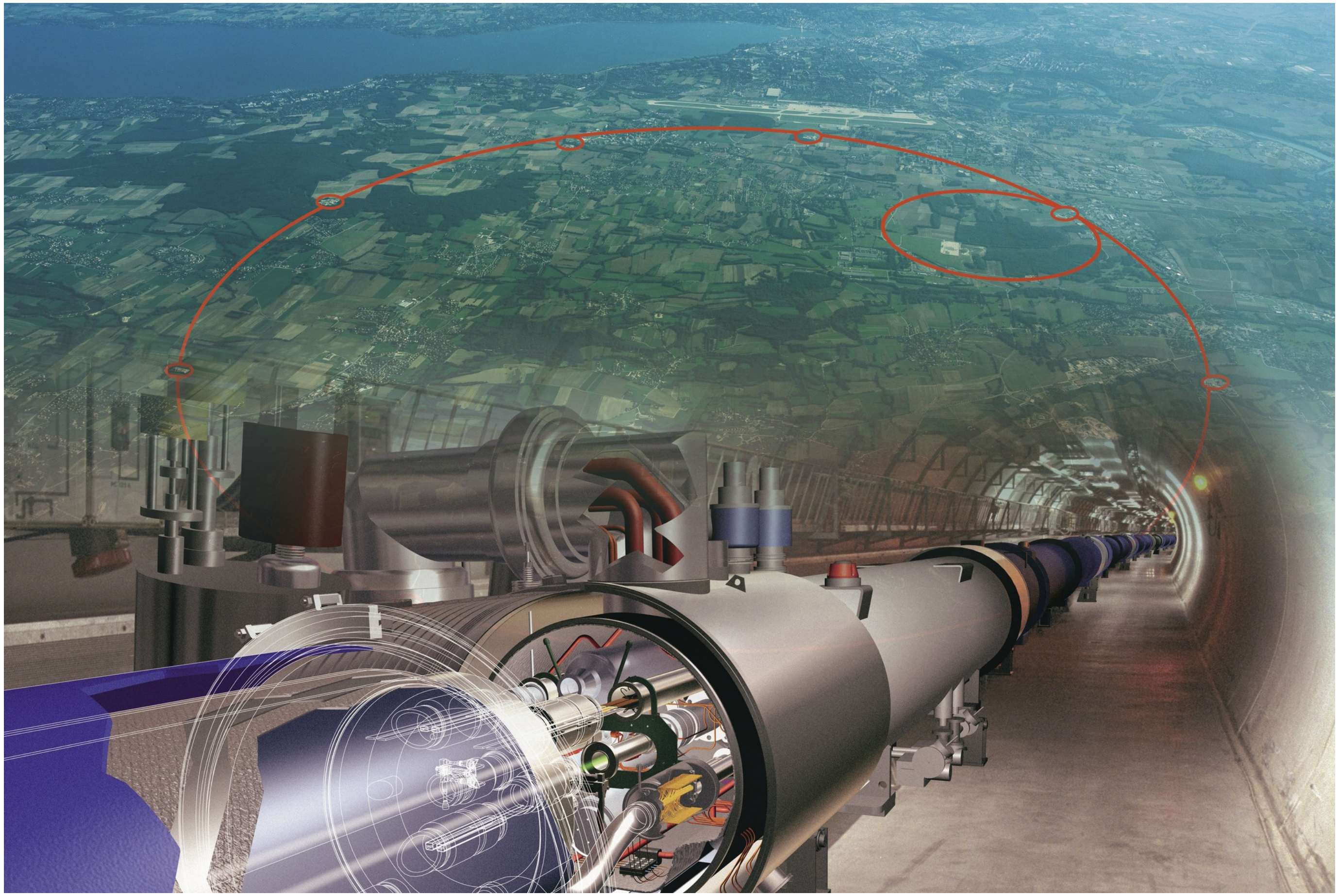
https://arxiv.org/pdf/1811.05993.pdf

LHC

Machine Learning and the Large Hadron Collider

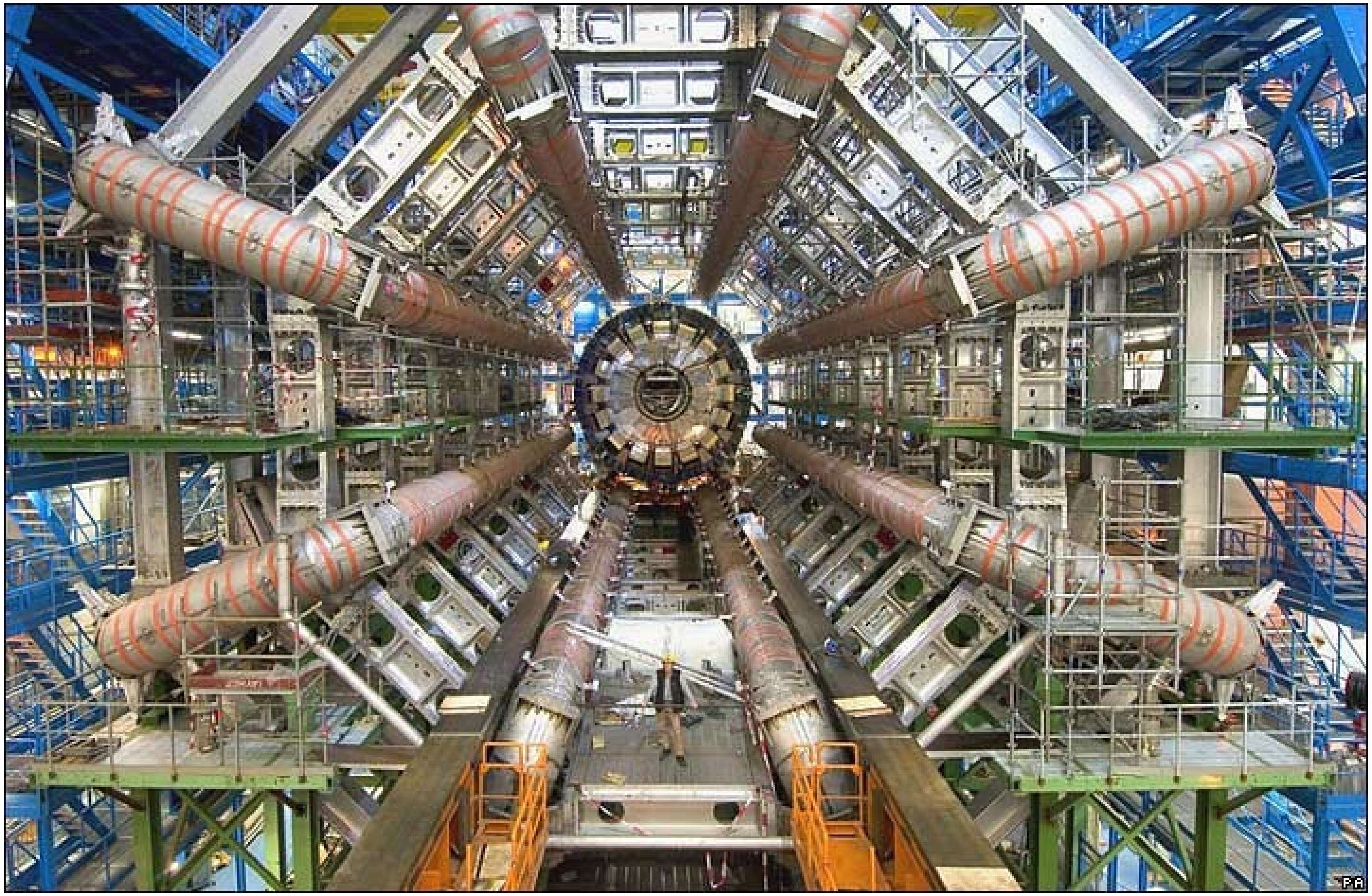


The Large Hadron Collider



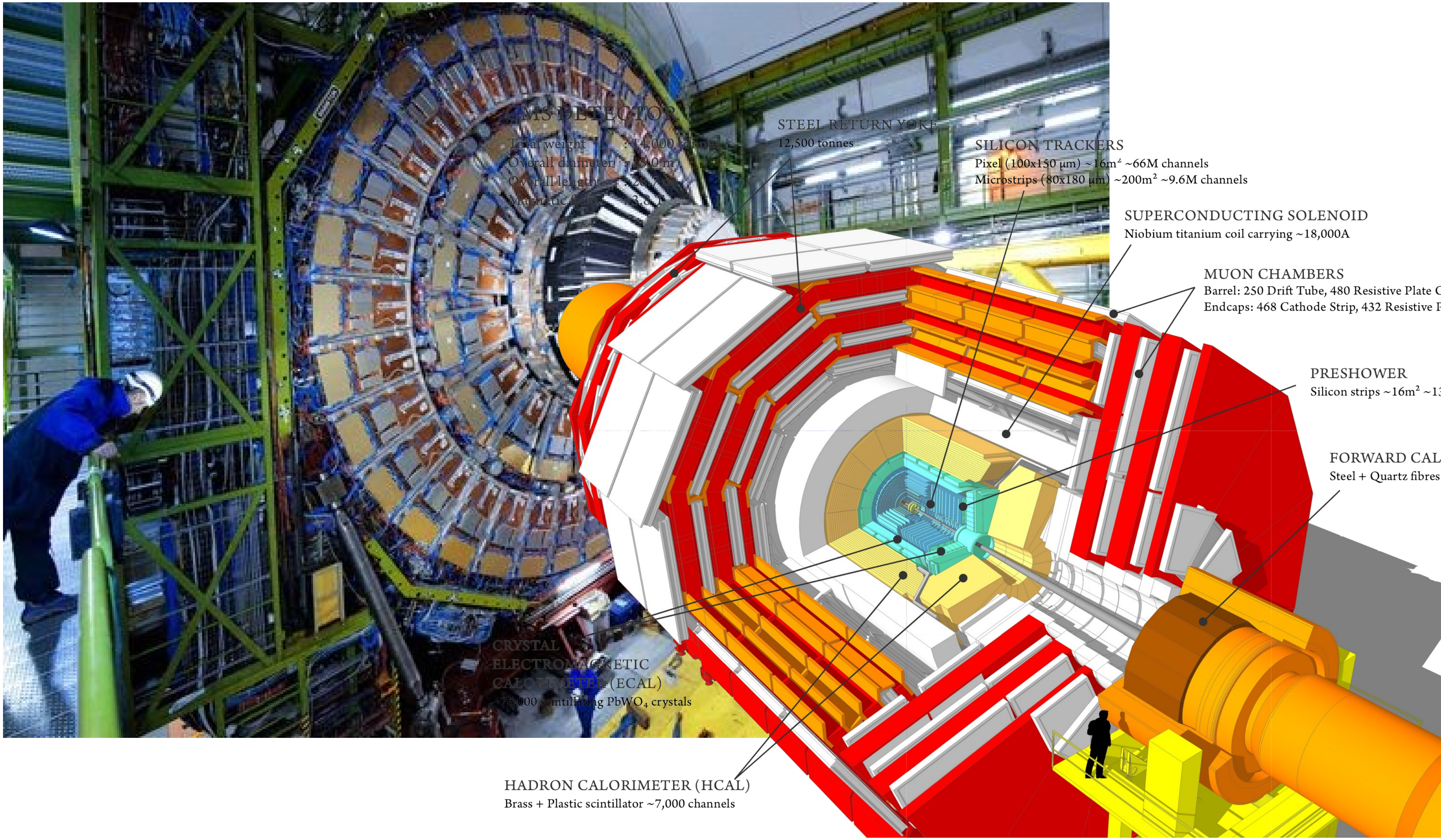
Proton-proton (and lead-lead) collider, 30 km circumference, currently operating at 13 TeV center-of-mass energy. 4 large experiments + a few smaller experiments, 2 general purpose experiments: ATLAS and CMS.

ATLAS – A Toroidal Large ApparatuS



Lorentz factor γ of our protons when they collide is ~ 7000

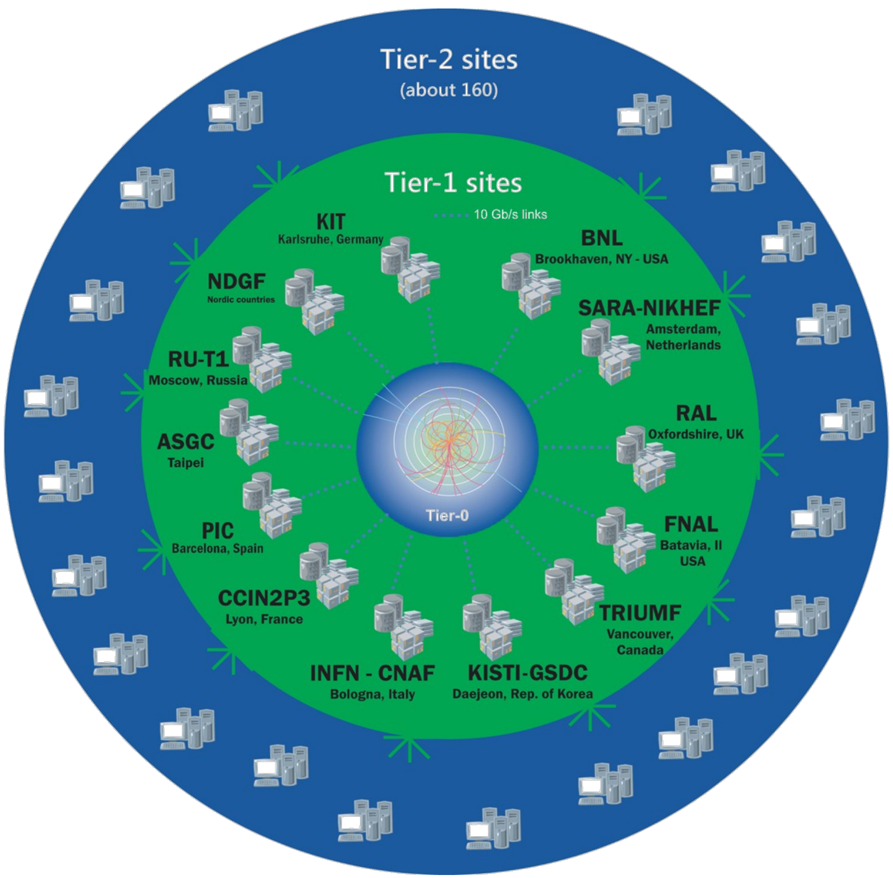
CMS – Compact Muon Solenoid

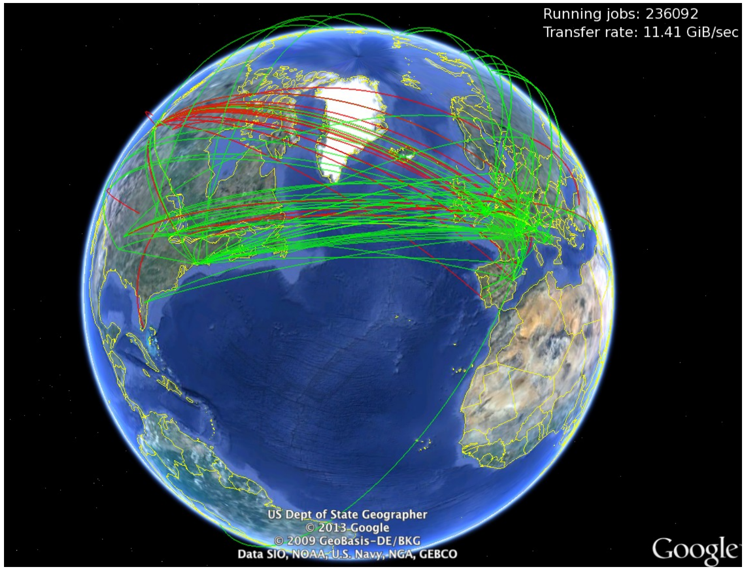


Taking 40 million images of proton-proton collisions every second.

The world-wide LHC Computing Grid

The data that survive our (multi-tiered) trigger systems, get distributed via the World-wide LHC Computing Grid (WLCG)

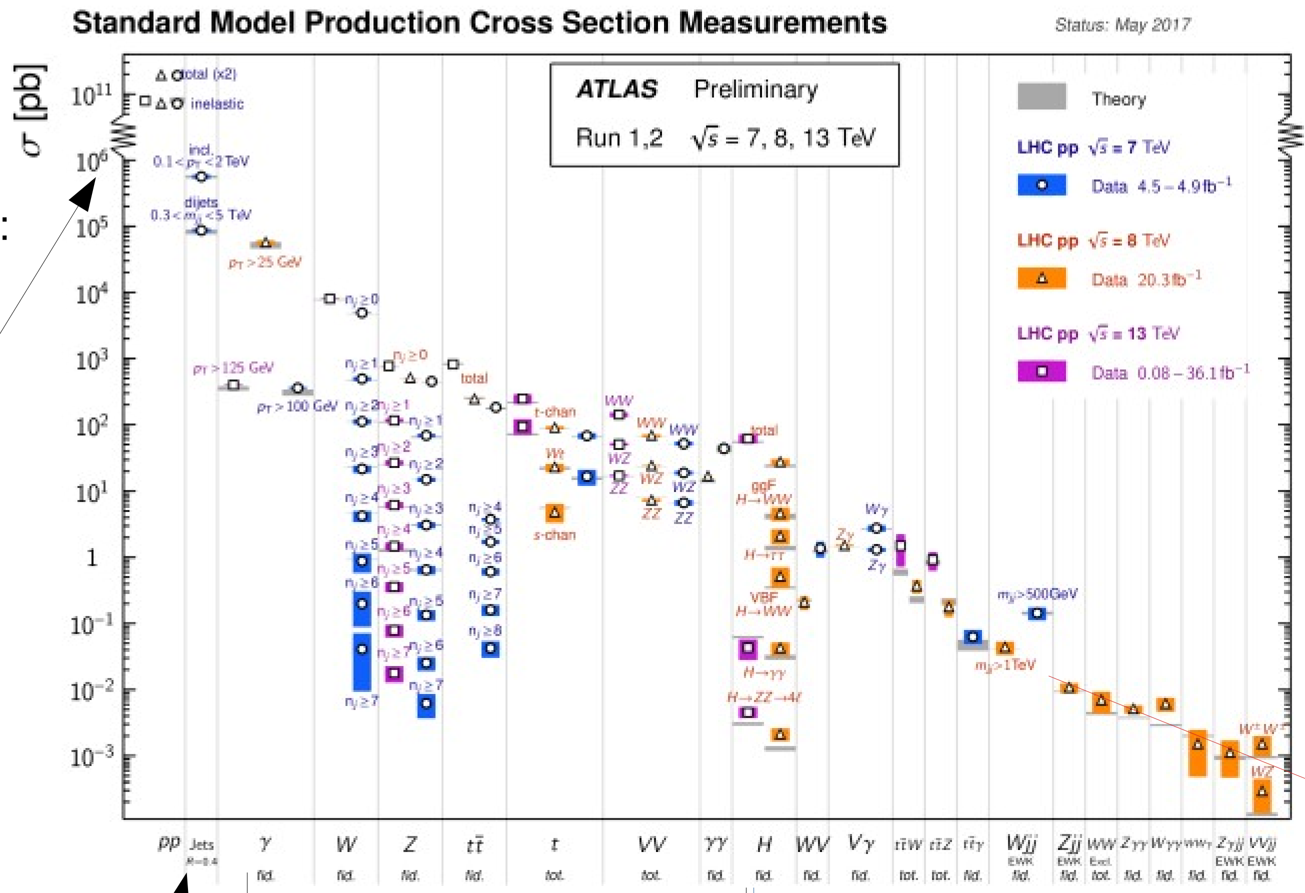




We are currently at ~ 1 Exabyte ($=10^{18}$ bytes) of storage and ~ 1 Million CPU cores.

https://wlcg-public.web.cern.ch/

Searching for the needle in the haystack



“cross section”: measure of frequency of occurence

5 orders of magnitude left out!

Physics processes

Not really interesting

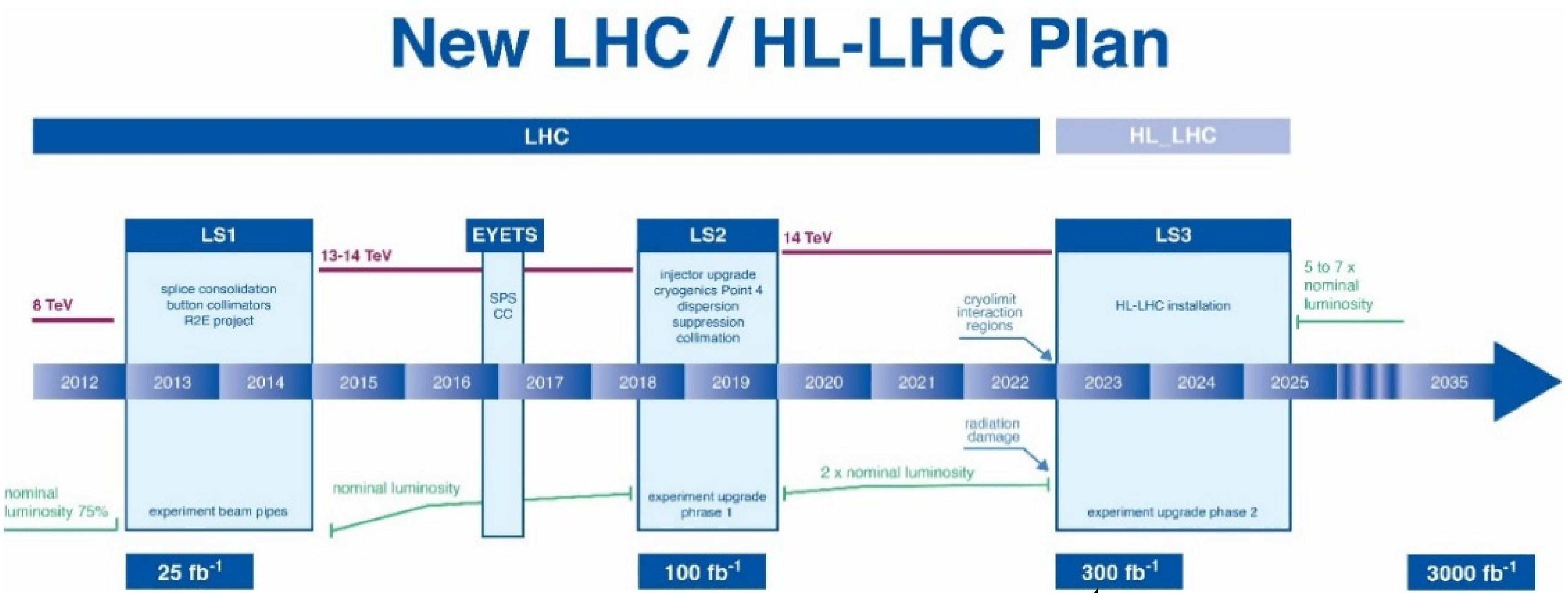
More interesting

Off the chart: expected physics beyond the known Standard Model

Very interesting

Prepare for the Future

And we start preparing for ~ 10 x more data!



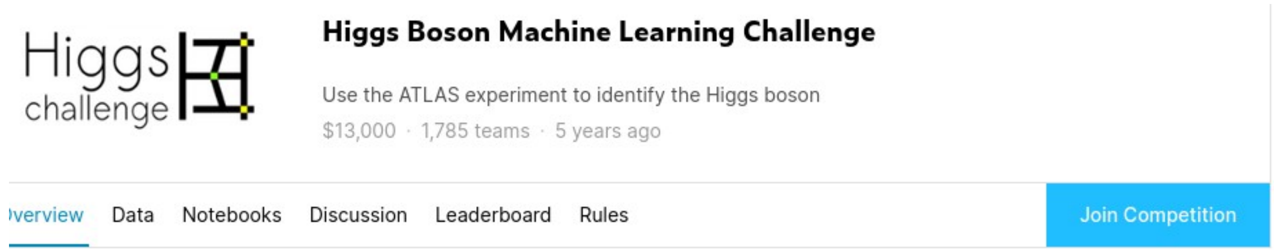
data now
(in funny units)

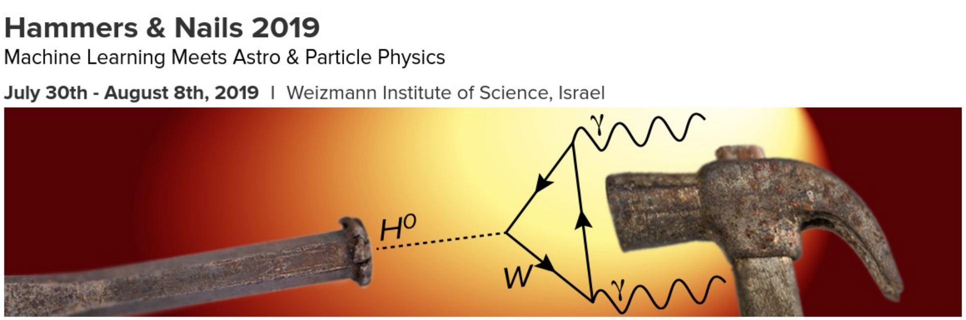
soon: ~ 3
times more
data

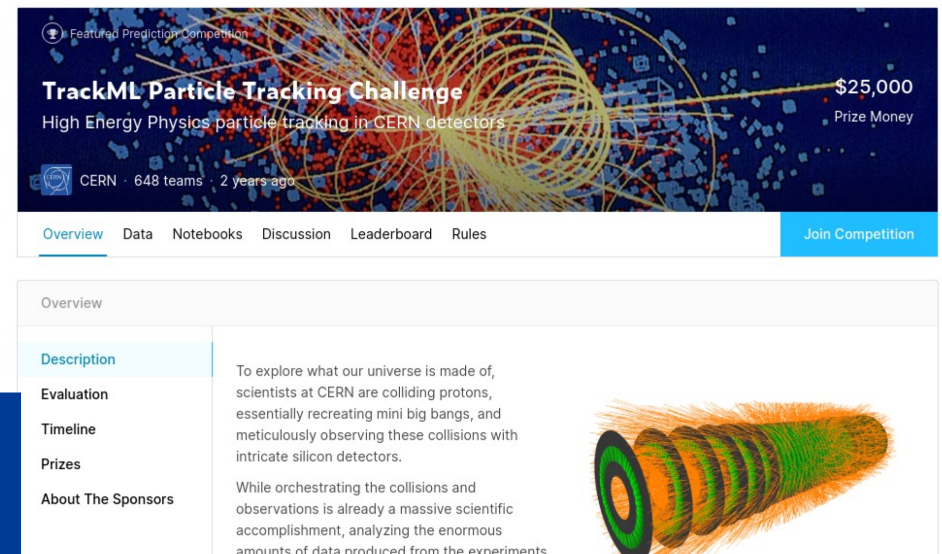
long term
future: ~ 30
more data

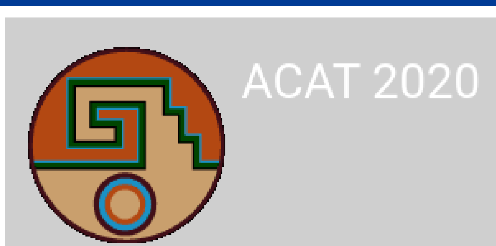
That is to say:

We need all the machine intelligence we can get!









20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research

Hardware

Non-standard Hardware platforms

Our community is also constantly exploring hardware platforms other than CPUs and GPUs.

E.g. our “hardware triggers” use lots of **FPGAs** – Field Programmable Gate Arrays

(FPGAs are something like ASICs, only reprogrammable)

Neural Networks on FPGAs

Our “hardware triggers” use lots of FPGAs – Field Programmable Gate Arrays

There was no solution on the market to quickly upload neuronal networks (for prediction only, without gradients) to the FPGAs, so members of our community developed software for this task: HLS4ML, High-Level Synthesis for Machine Learning

https://github.com/hls-fpga-machine-learning/hls4ml

"Fast inference of deep neural networks in FPGAs for particle physics"

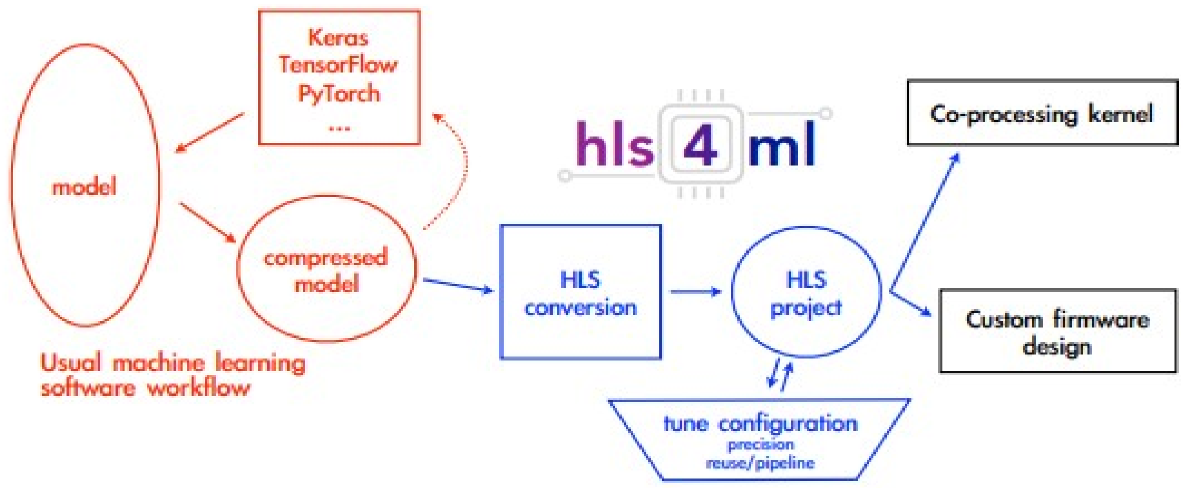


Figure 1: A typical workflow to translate a model into a FPGA implementation using `hls4ml`.

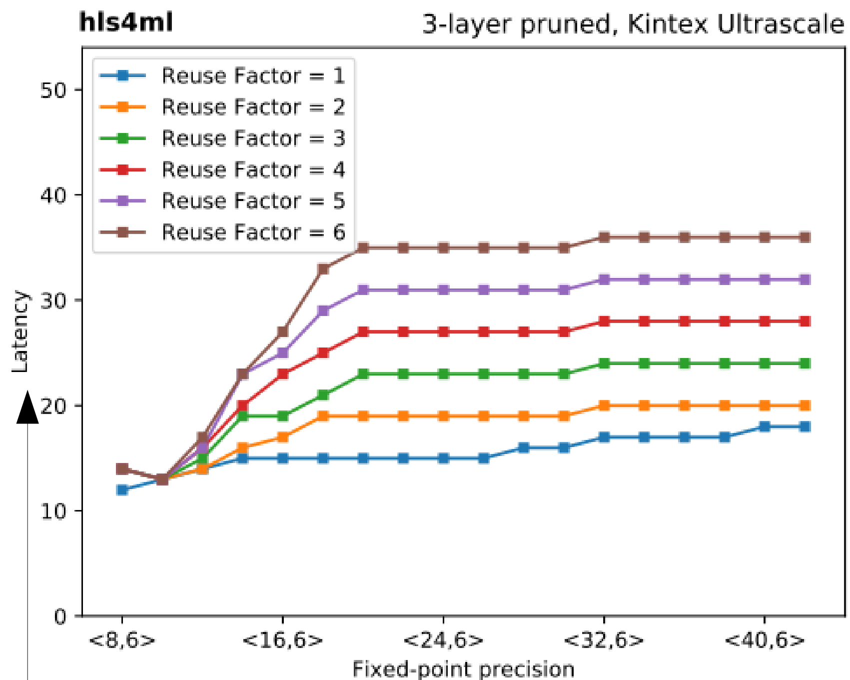
Architectures/Toolkits	Keras/TensorFlow	PyTorch	scikit-learn
MLP	`supported`	`supported`	-
Conv1D/Conv2D	`supported`	`in development`	-
BDT	-	-	`in development`
RNN/LSTM	`in development`	-	-

Currently only the simpler network architectures are supported, but this is changing

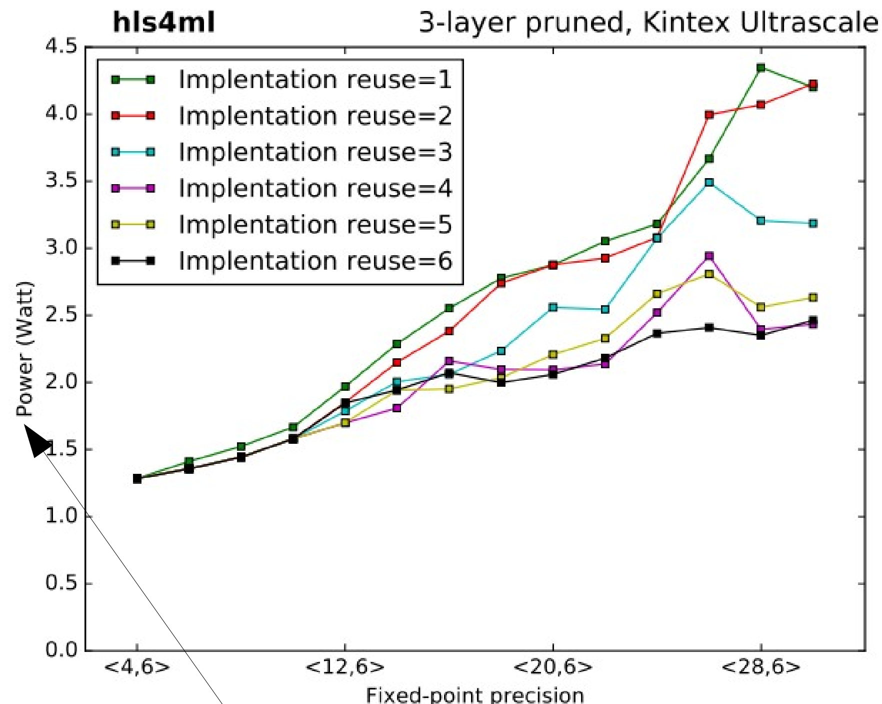
https://arxiv.org/abs/1804.06913

https://github.com/hls-fpga-machine-learning/hls4ml

“Fast inference of deep neural networks in FPGAs for particle physics”



Number of clock cycles: O(100 ns) for a prediction of 3-layer network! Ultra-low latency neural network predictions!



O(1) Watt, not O(100) Watts like GPUs!

https://arxiv.org/abs/1804.06913

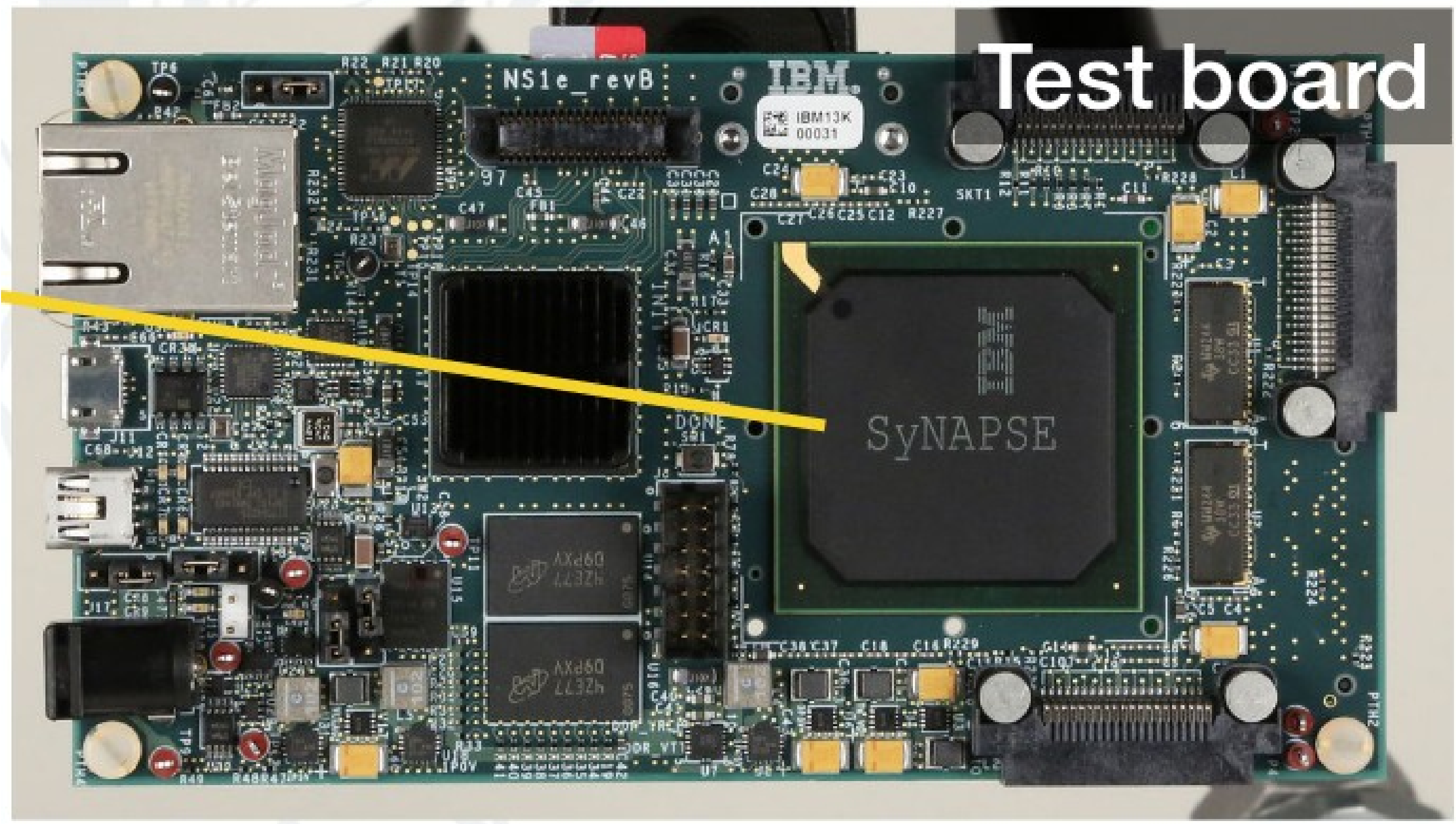
https://github.com/hls-fpga-machine-learning/hls4ml

Neuromorphic Hardware

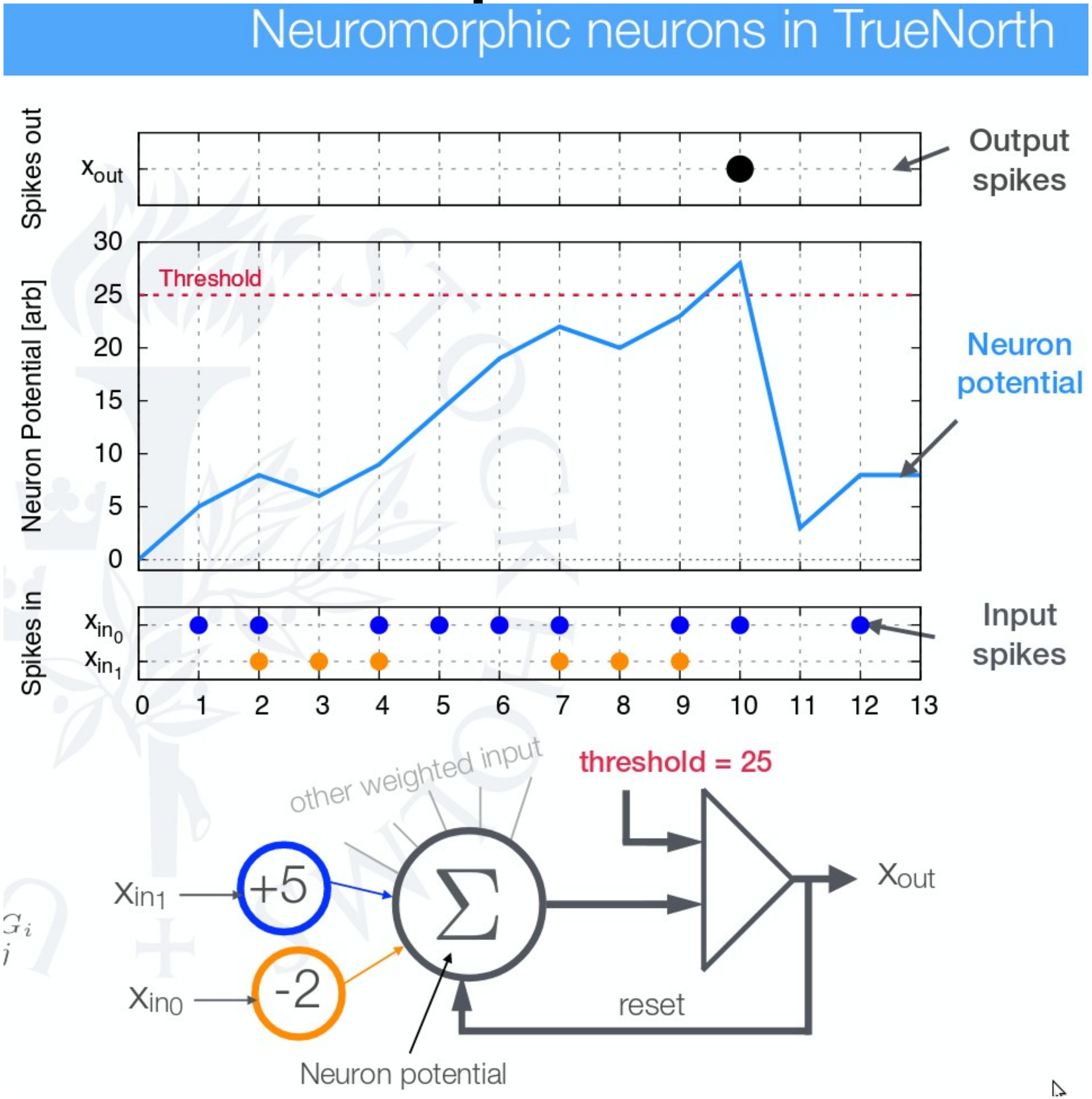
Neuromorphic Chips: experimental, post-von-Neumann architecture, loosely modeled after human brain (analog, spiking signals as opposed to “weights” in ANNs).

“True North” architecture: experimental program conducted by DARPA and IBM

Power consumption: tens of Milliwatts!



Neuromorphic Hardware

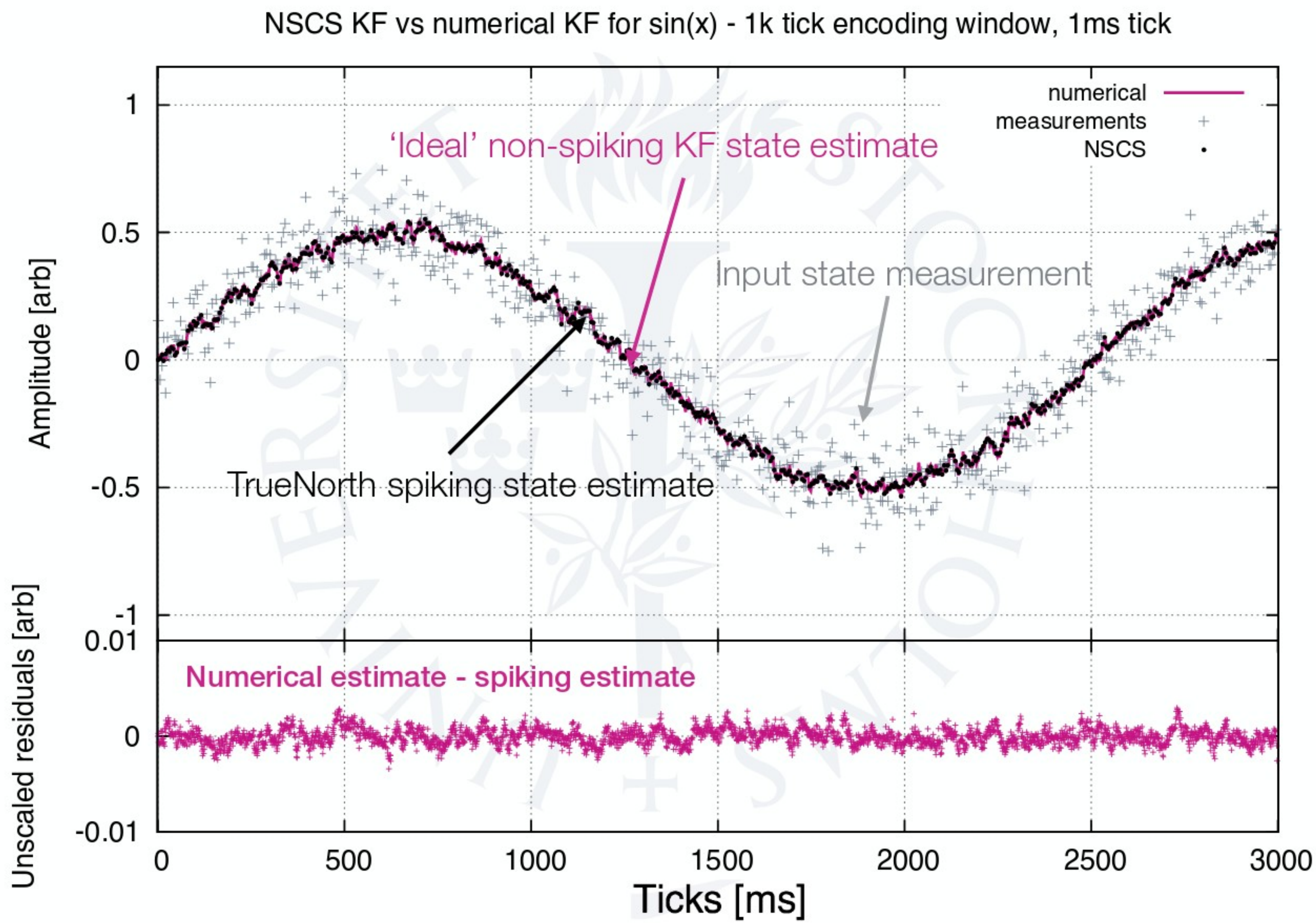


Rebecca Carney, CTD2017 Talk

Neuromorphic Hardware

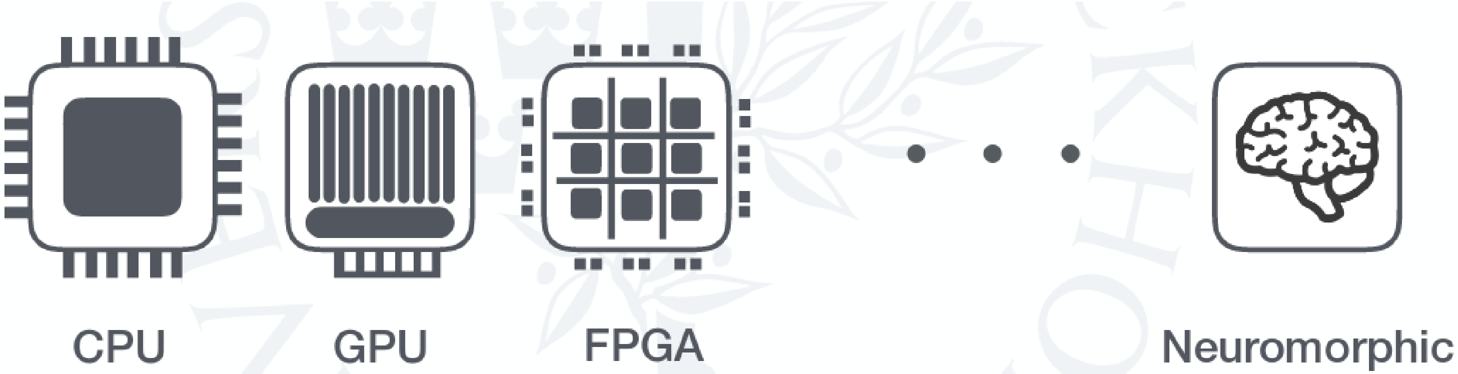
rcarney@lbl.gov

Does it work?



Rebecca Carney, CTD2017 Talk

Neuromorphic Hardware



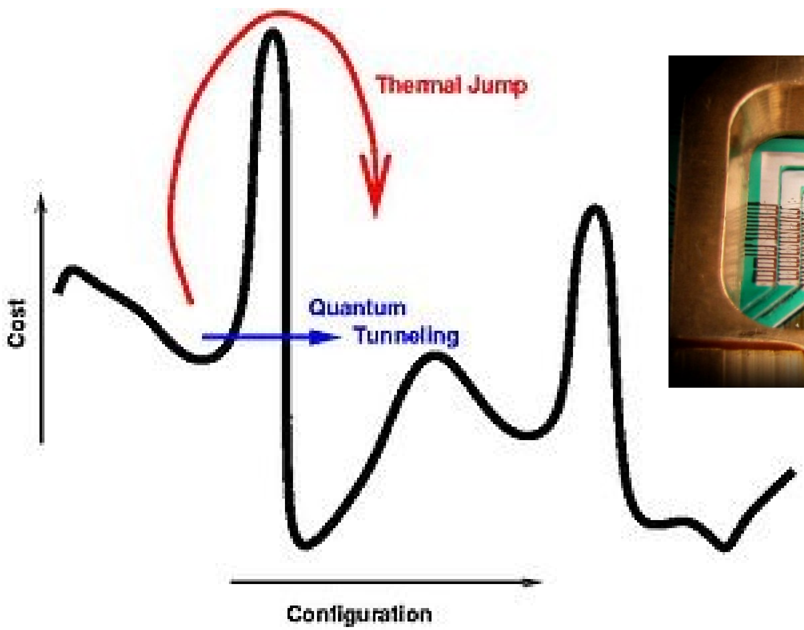
- We have adapted CPU's, GPU's and FPGA's for our needs. Could neuromorphic chips feature in our toolkit in 10 years time? Perhaps but it seems more likely we would use an ASIC ANN.

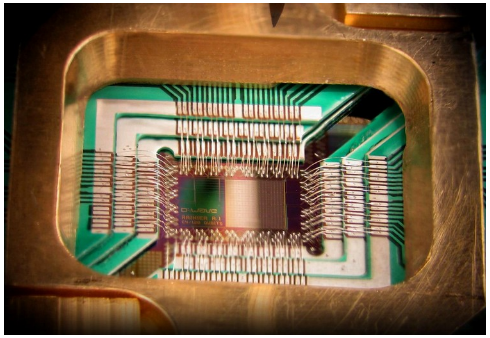
Rebecca Carney, CTD2017 Talk

Adiabatic Quantum Computing (“Quantum Annealers”) for LHC Physics

Idea: map one of our optimization problems (“track finding”, a pattern recognition problem) to a Hamiltonian that can be simulated on a D-Wave quantum annealer.

Let quantum mechanics find the global minimum of your loss function.





https://arxiv.org/abs/1902.08324

Heather Gray, ACAT 2019

Adiabatic Quantum Computing (“Quantum Annealers”) for LHC Physics

2.3.1 Definition of the Quadratic Unconstrained Binary Optimization

The QUBO is configured to identify the best pairs of triplets. It has two components: a linear term that weighs the quality of individual triplets and a quadratic term used to express relationships between pairs of triplets. In our case, the objective function to minimize becomes:

$$O(a, b, T) = \sum_{i=1}^{N} a_i T_i + \sum_{i}^{N} \sum_{j<i}^{N} b_{ij} T_i T_j \tag{3}$$

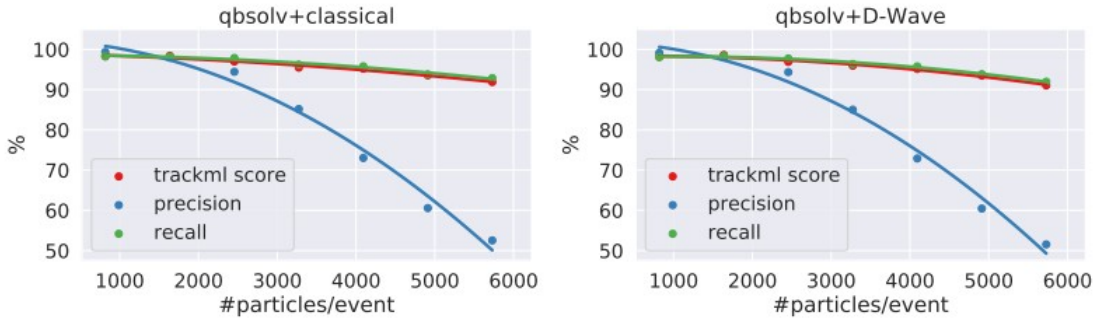


Fig. 3 The performance of classical simulator (left) and D-Wave (right), as measured by TrackML score (red), purity (blue), and efficiency (green), as a function of particle multiplicity.

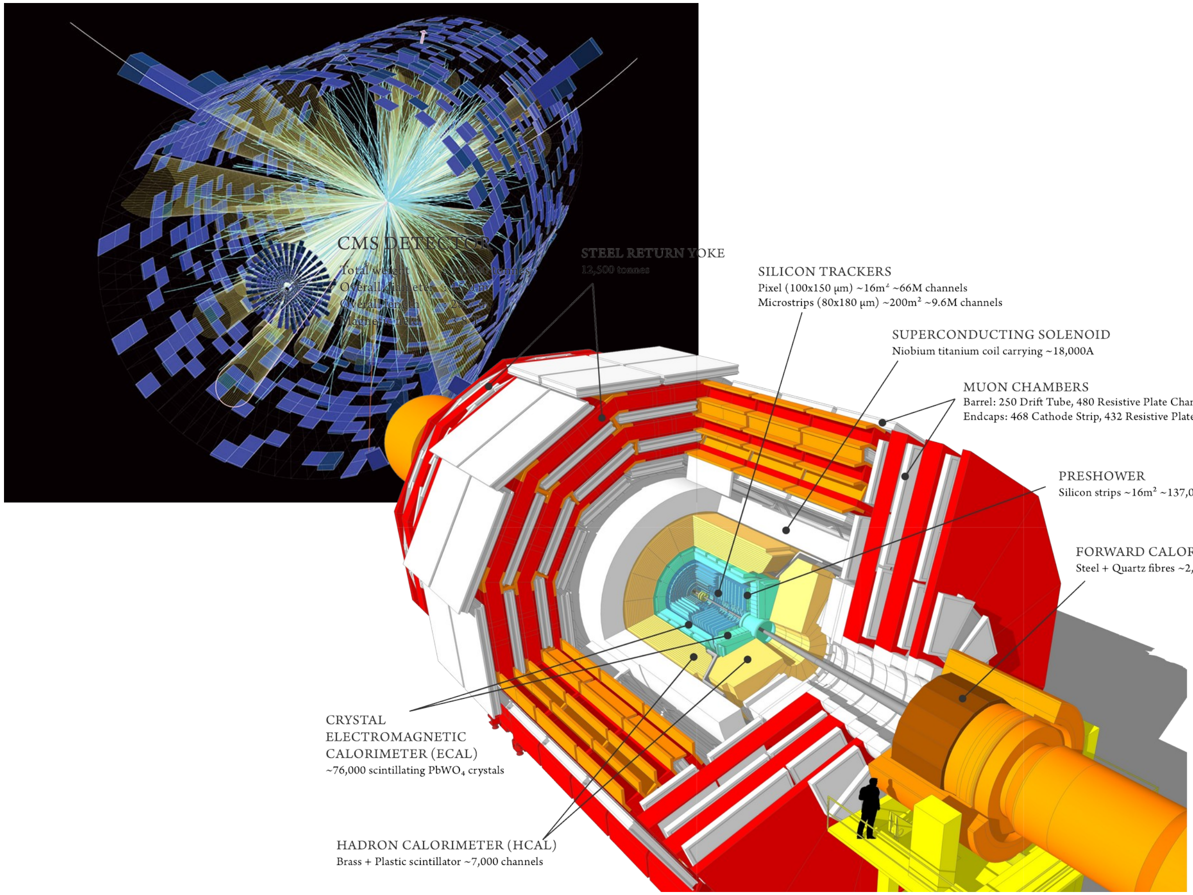
https://arxiv.org/abs/1902.08324

Heather Gray, ACAT 2019

Machine Learning Applications @ LHC

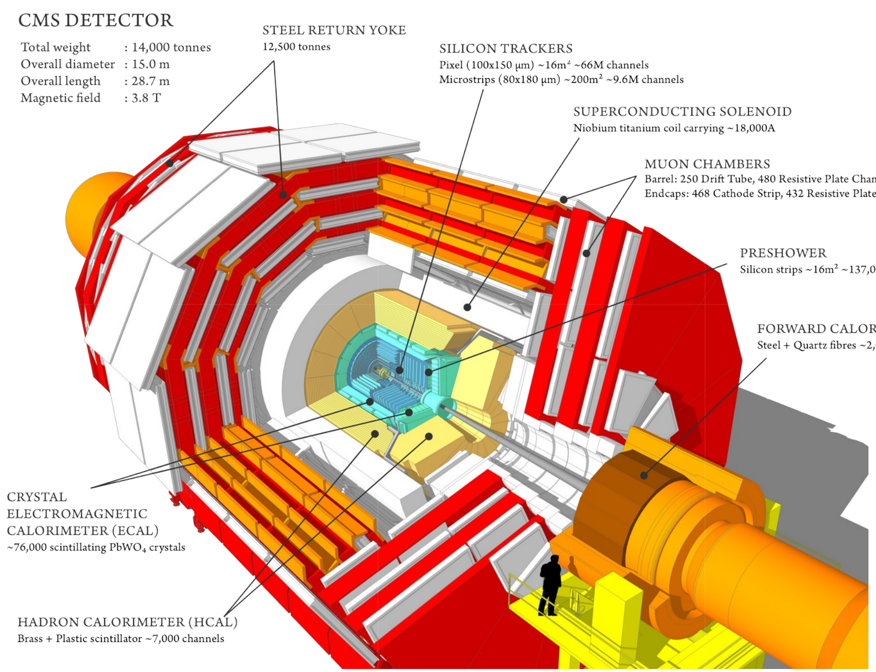
LHC’s Data Structures

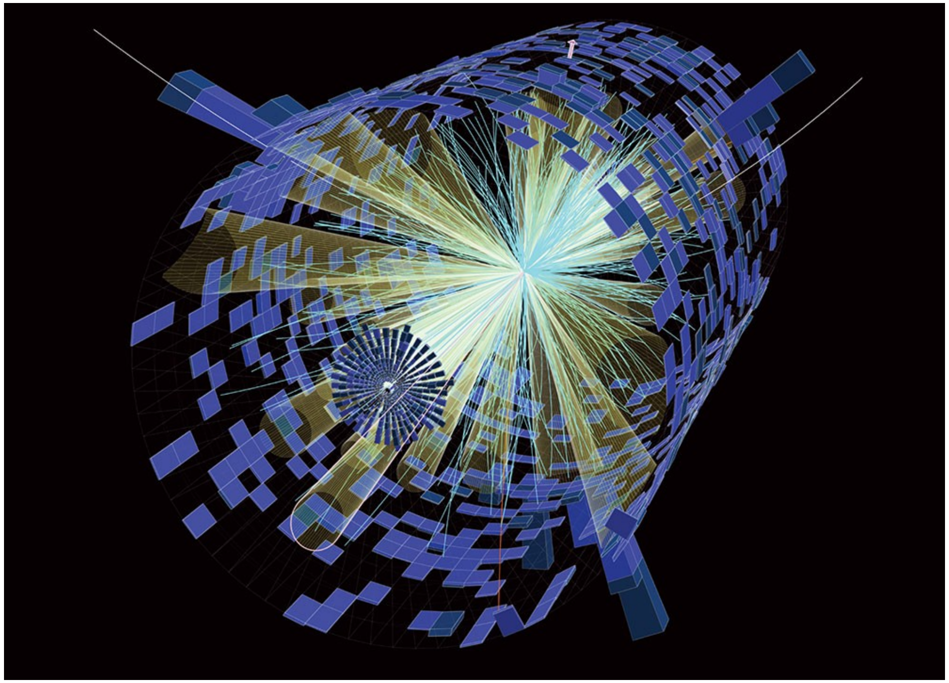
- Most of the times, we can think of an “event” as our fundamental data structure
- An event is one “image” of a proton-proton collision
- Our detectors are “layered” and use heterogeneous technologies. One event is thus the combination of many different types of detectors.



We have a gazillion such events, both “real”, measured events, and simulated events, so we can compare e.g. the measured events with certain hypothetical physics scenarios.

LHC’s Data Structures





We have a gazillion such events, both “real”, measured events, and simulated events, so we can compare e.g. the measured events with events from certain hypothetical physics scenarios.

We usually can treat an event to be statistically independent from any other event, which very often makes parallelization a very easy task for us – there is usually no need for cross-talk between threads, processes, nodes.

LHC's Data Structures

Our final "physics results" are typically then given as a statement on a large statistics of "events".

One of the "money plots" in the 2012 Higgs discovery paper by ATLAS.

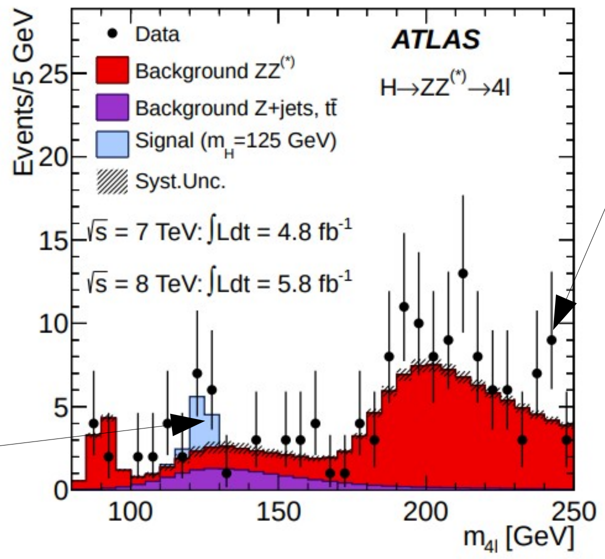


Figure 2: The distribution of the four-lepton invariant mass, $m_{4\ell}$, for the selected candidates, compared to the background expectation in the 80–250 GeV mass range, for the combination of the $\sqrt{s} = 7$ TeV and $\sqrt{s} = 8$ TeV data. The signal expectation for a SM Higgs with $m_H = 125$ GeV is also shown.

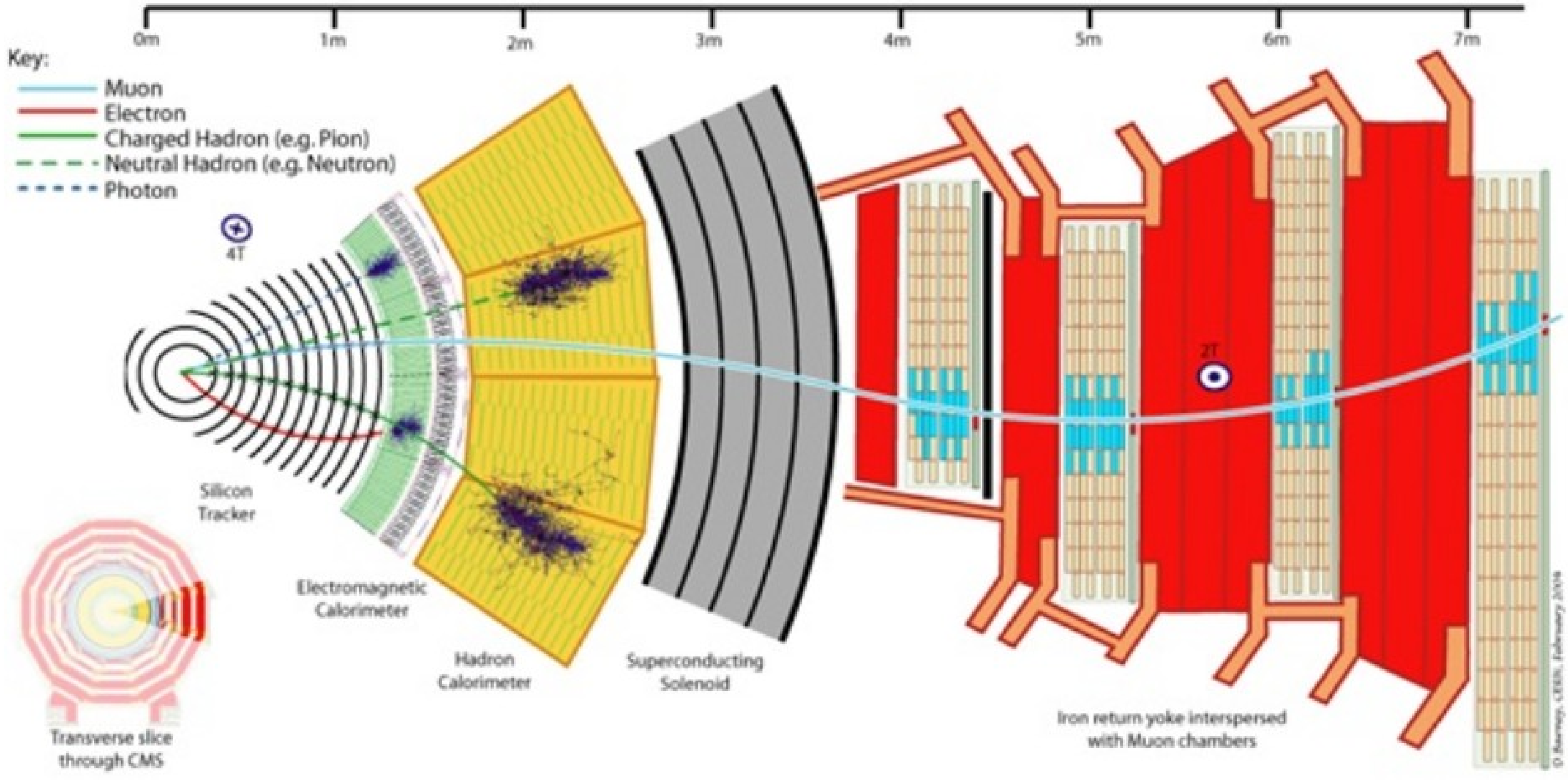
The black dots show the number of events in that bin

The blue area Indicates the excess in number of events due to a Higgs particle, as predicted by theory

https://arxiv.org/pdf/1207.7214.pdf

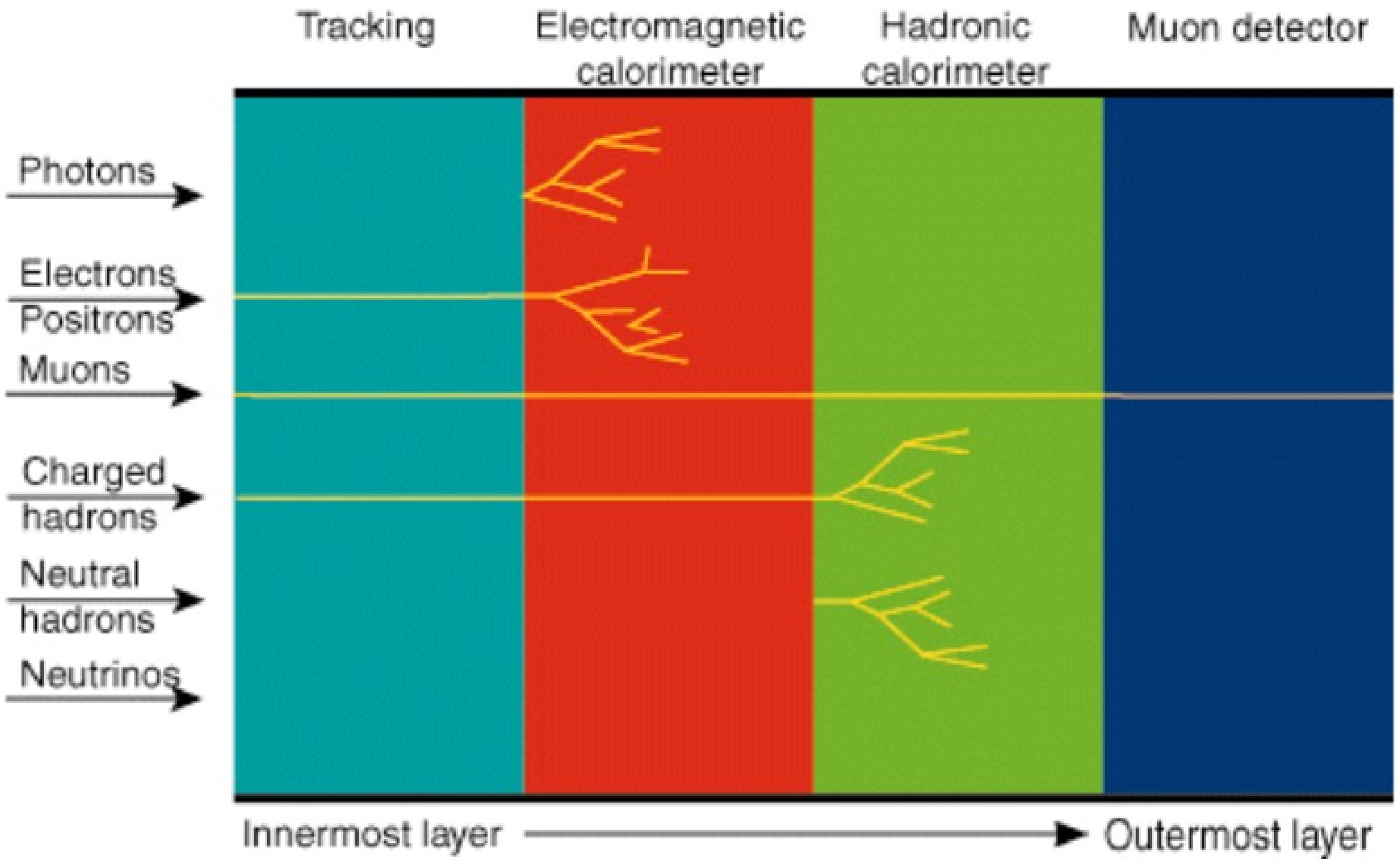
LHC's Data Structures

We think of an event as being composed of "physics objects":



LHC's Data Structures

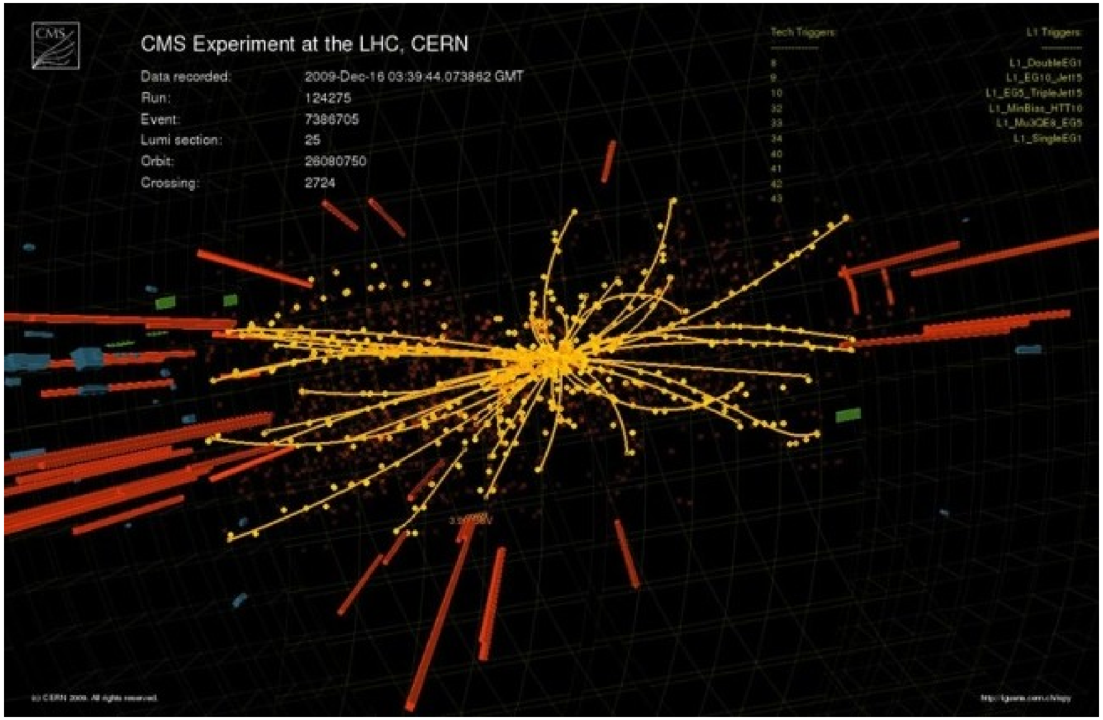
We think of an event as being composed of "physics objects":



LHC’s Data Structures

In every layer, and for every physics object we have a few machine learning problems.

Innermost layer: reconstruction of charged tracks in our silicon “tracker”:

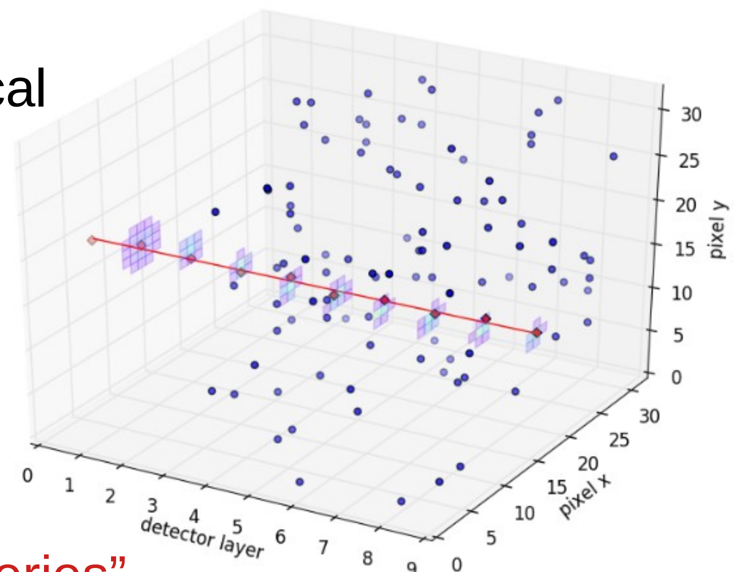


Problem: given the interactions (“hits”) of the charged particles with the silicon detectors, reconstruct (find and fit) the particle tracks. Pattern recognition as well as regression problem!

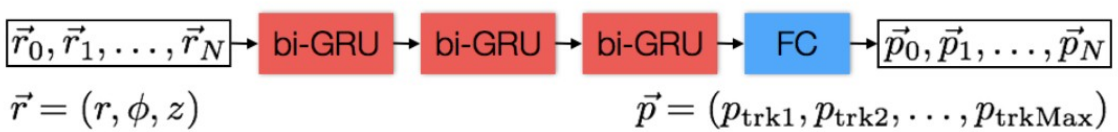
LHC's ML Challenges

In every layer, and for every physics object we have a few machine learning problems.

Innermost layer: reconstruction of charged tracks in our silicon "tracker":



- **In the past**, we have used more "classical" statistical methods, such as (combinatorial) Kalman filter, EM algorithms, Gaussian mixture models to solve the task.
- **Now also** trying e.g. recurrent neural networks, interpret the track as "something similar to a time series"

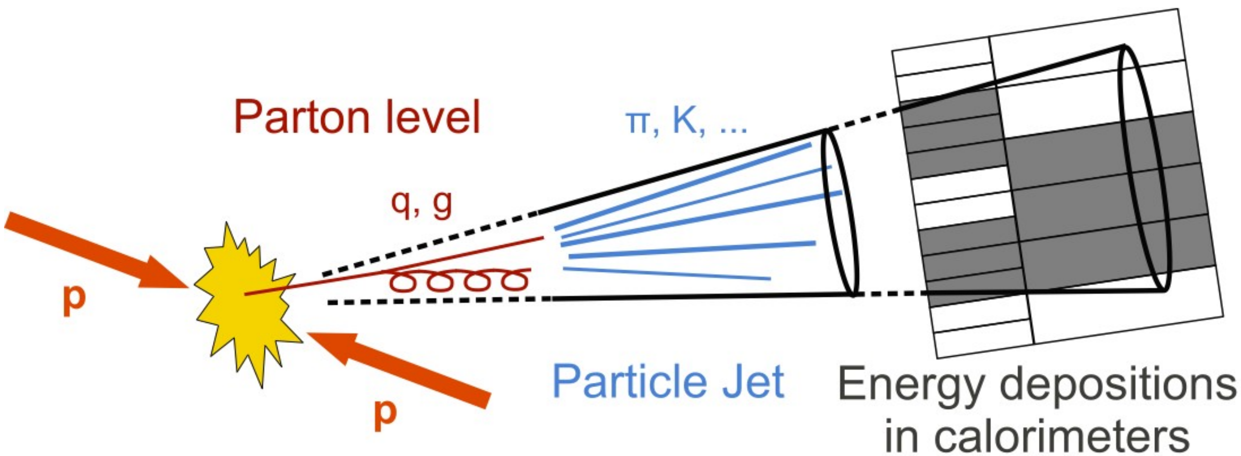


Deep Learning for Physical Sciences, NIPS 2017

LHC’s ML Challenges

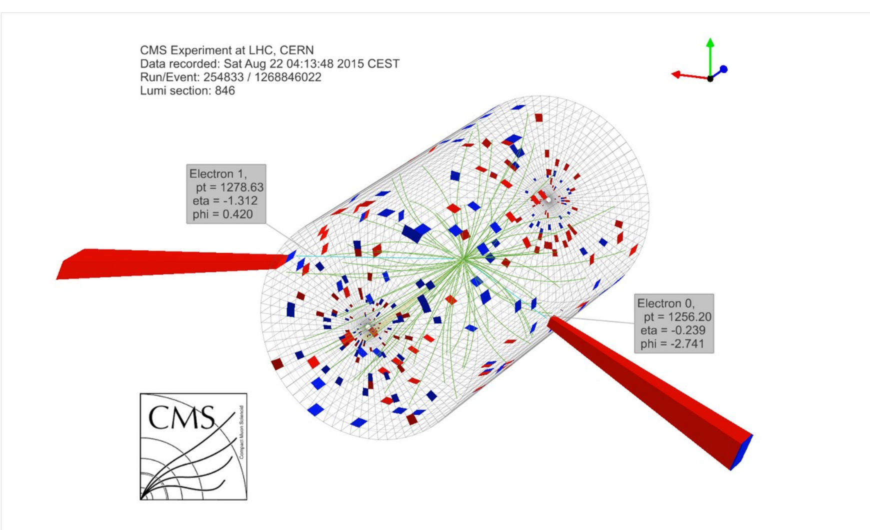
In every layer, and for every physics object we have a few machine learning problems.

Calorimeters: identification of particle “jets” (spray of particle that we see whenever quarks get produced – confinement) – and their origins



Task: learn a jet classifier
that identifies highly boosted “W” jets

In the past: many different types of physics-driven clustering algorithms for finding the jets, simple multi-variate Algorithms for identifying the physics process behind the jet



LHC's ML Challenges

In every layer, and for every physics object we have a few machine learning problems.

Calorimeters: identification of origin of particle "jets" (spray of particle that we see whenever quarks get produced – confinement)

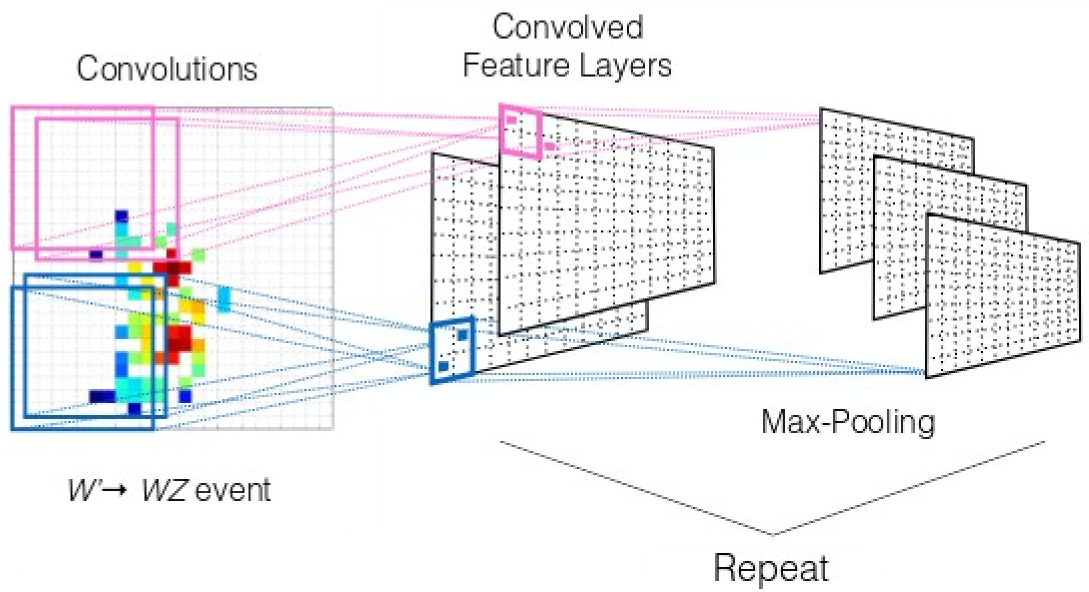


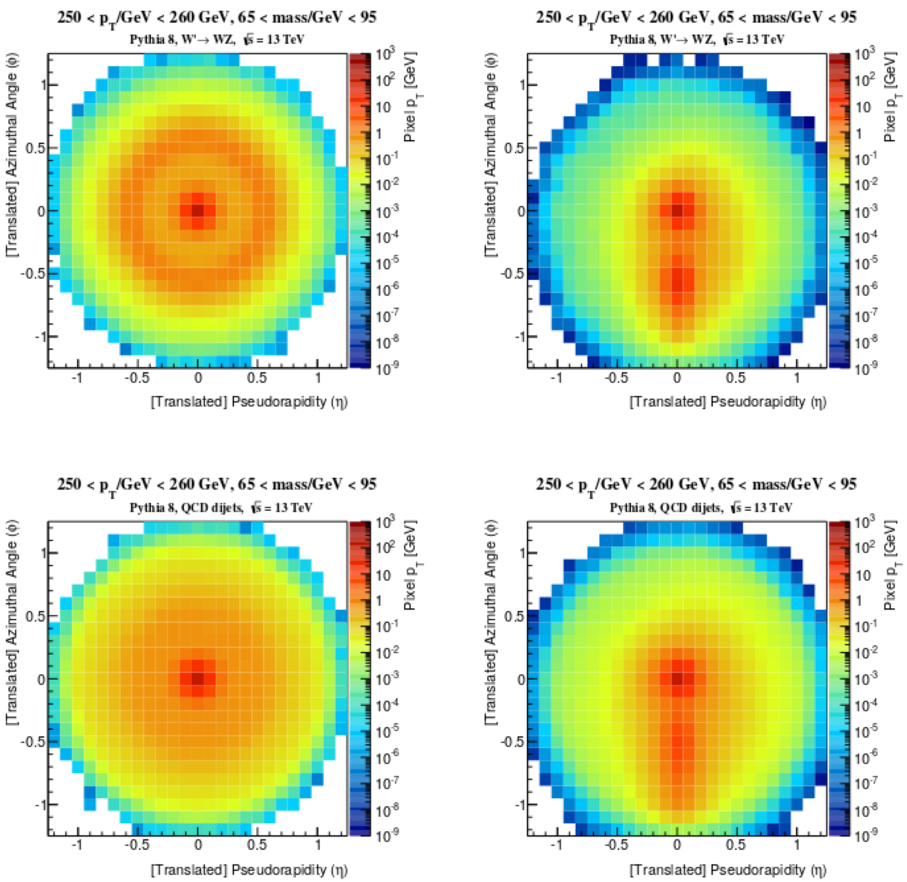
Figure 5: The convolution neural network concept as applied to jet-images.

Task: learn a jet classifier that identifies highly boosted "W" jets

In this paper: treat our calorimeters like an image, apply convolutional neural networks, learn difference between jets originating from gluons and quarks ("QCD"), from W bosons, or from hypothetical new particles

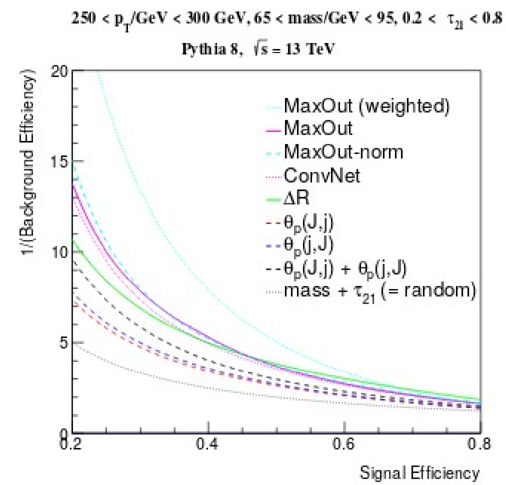
https://arxiv.org/pdf/1511.05190.pdf

jet images



average signal

average background

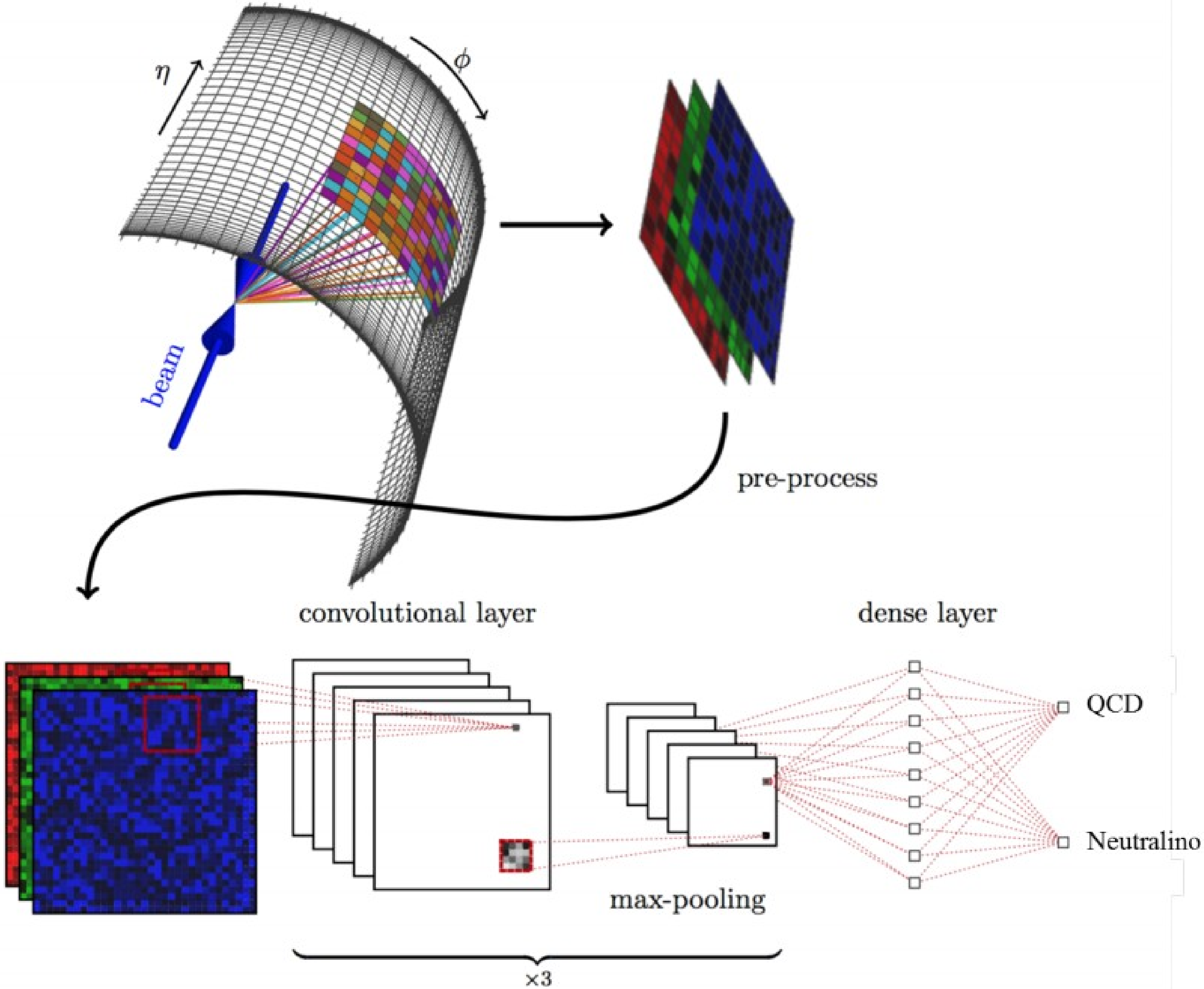


Receiver Operating Characteristic (ROC) curves for various choices of hyperparameters

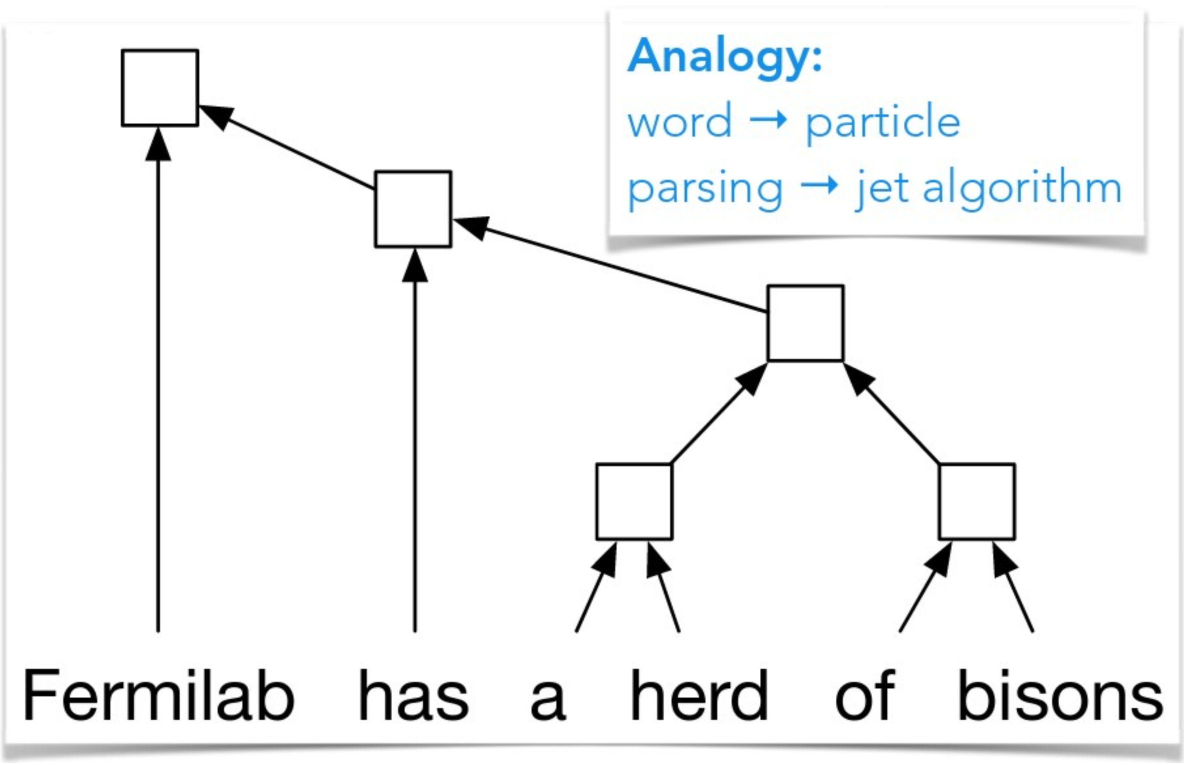
Figure 2: The average jet image for signal W jets (top) and background QCD jets (bottom) before (left) and after (right) applying the rotation, re-pixelation, and inversion steps of the pre-processing. The average is taken over images of jets with 240 GeV $< p_T <$ 260 GeV and 65 GeV < mass < 95 GeV.

https://arxiv.org/pdf/1511.05190.pdf

Jet into images



Recursive neural networks for jets



Task: learn a jet classifier that identifies highly boosted "W" jets

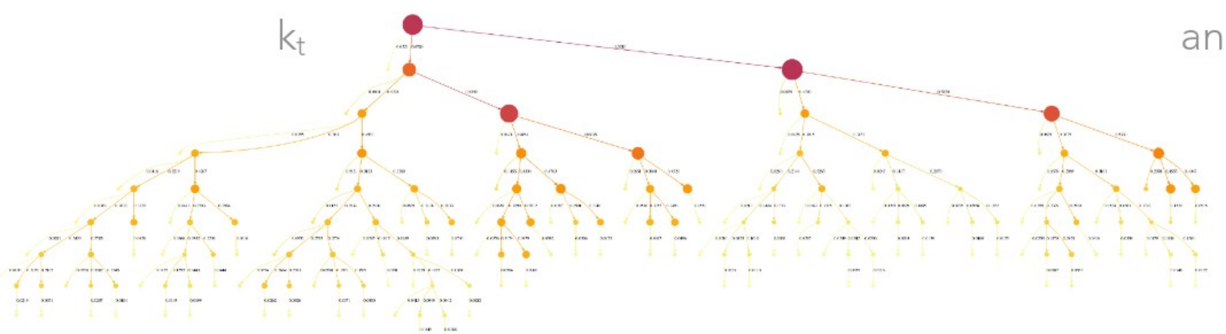
Idea: this "spray of particles" that makes up a jet is somewhat reminescent of the recursive nature of parse trees and recursive neural networks – treat it similar to how you would treat a natural sentence.

See also talk at
DS@HEP 2017 workshop

https://arxiv.org/pdf/1702.00748.pdf

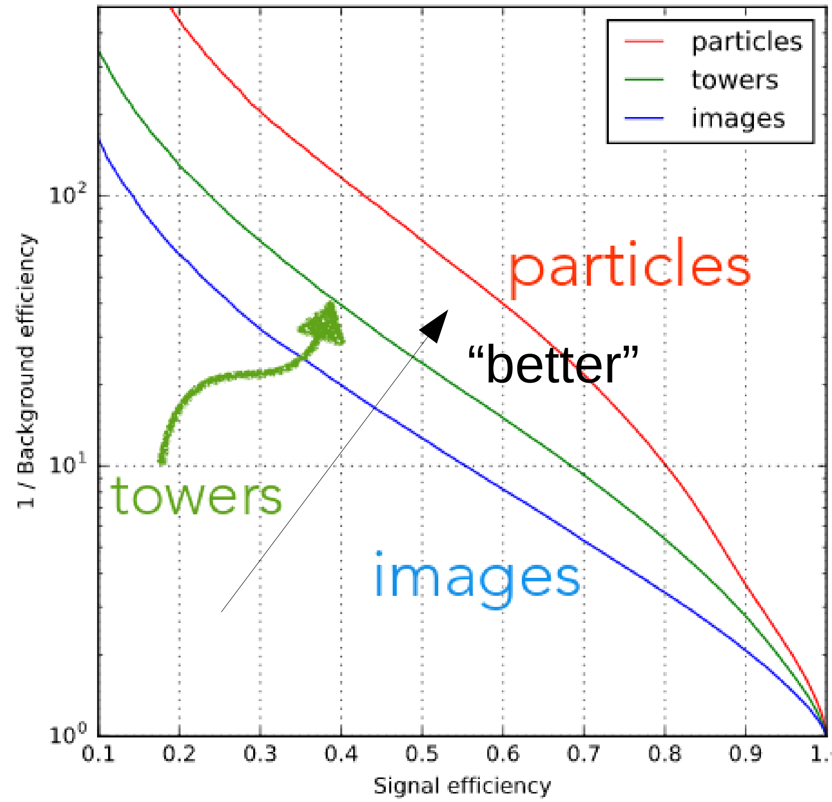
Application (of a similar class of algorithms) in particle physics: jet classification

“QCD-aware recursive neural networks for jet physics”



Visualisation of k_t jet algorithm.

“particles”: recursive network
“images”: convolutional network



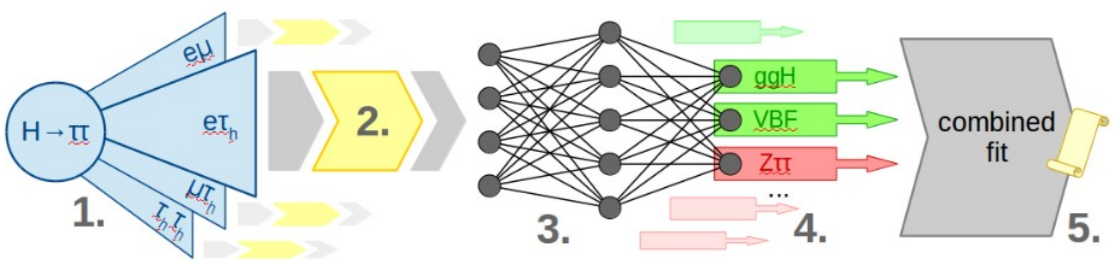
ROC curve of W jet tagging example, comparison with jet images algorithm

See also talk at
DS@HEP 2017 workshop

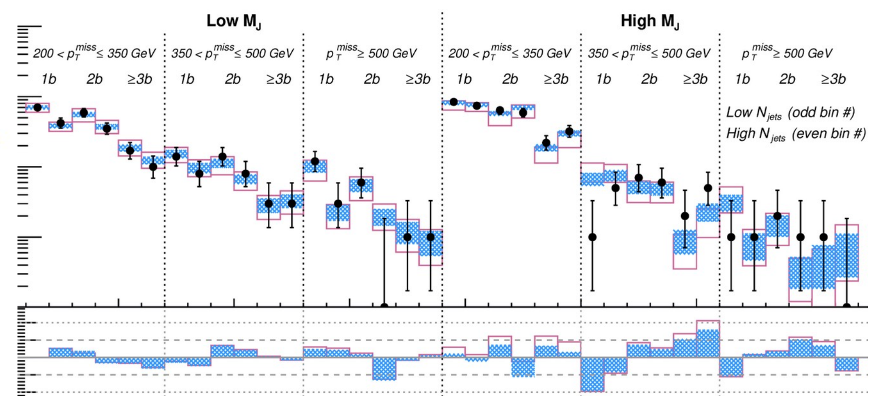
https://arxiv.org/pdf/1702.00748.pdf

From physics objects to physics results

From these deeply-learned physics objects we then need to arrive at quantitative statements about fundamental physics, like the estimated mass of a known particle or the level of certainty with which we exclude a certain model beyond known physics.



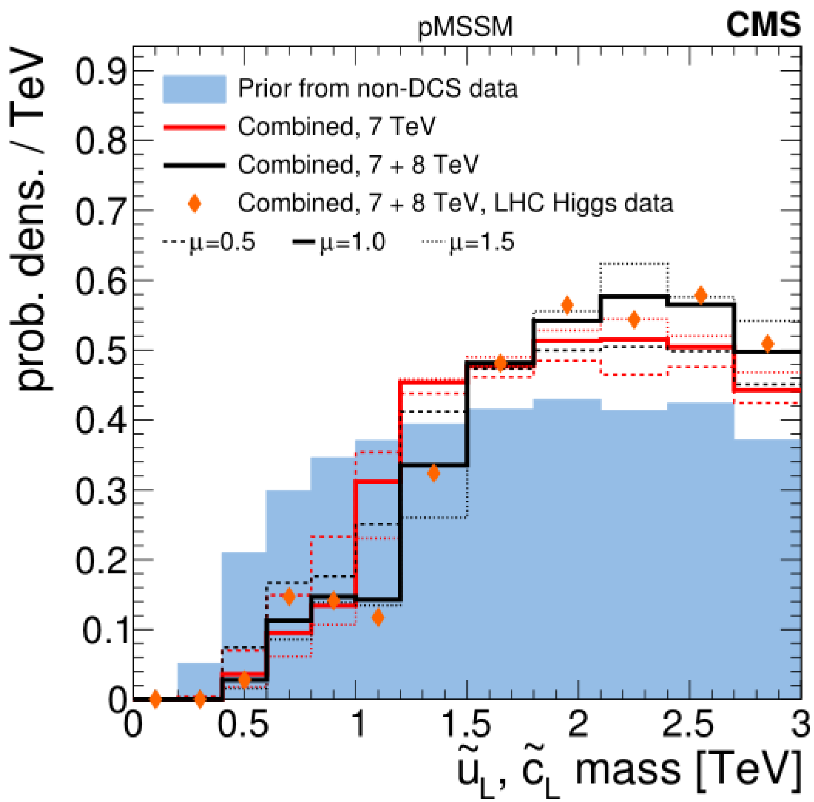
Learning a classifier of the “Higgs production mechanism”, then performing a classical, frequentist maximum likelihood fit



Various different physically meaningful, hand-crafted event selections. The final steps of A search for supersymmetry at the LHC.

From physics objects to physics results

From these deeply-learned physics objects we then need to arrive at quantitative statements about fundamental physics, like the estimated mass of a known particle or the level of certainty with which we exclude a certain model beyond known physics.



arXiv:1606.03577

Our “Bayesian”, “subjective” knowledge about supersymmetry, before (blue area) and after (black and red lines) having directly searched for SUSY.

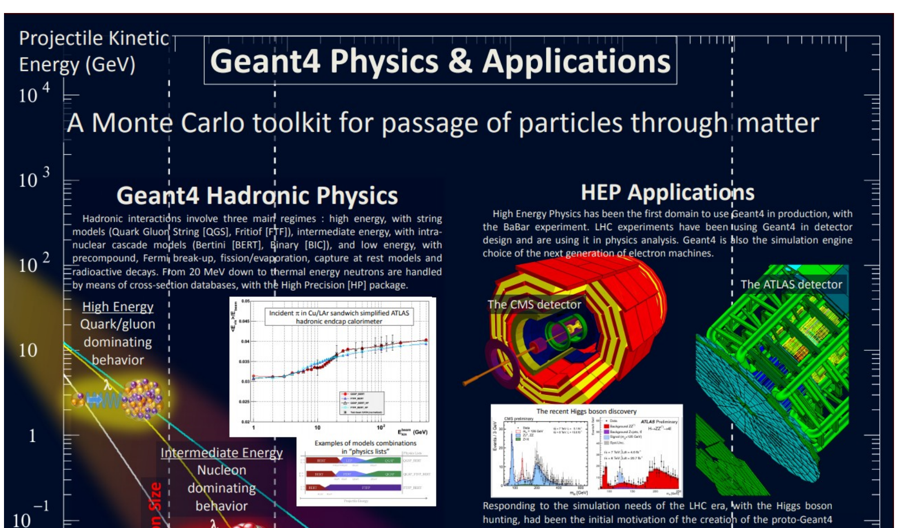
LHC's ML Challenges: Event Simulation

A roughly estimated 75% of the almost 1 Exabyte of LHC data is simulations, not measured data! Simulations for us can be costly. Why? Because we need to

1) Simulate the proton-proton collision. That's applied, detailed quantum field theory, often at higher perturbation orders.

2) Simulate the "hadronization", i.e. the forming of particles and particle jets from quarks and gluons.

3) Simulate the response of the detector, using e.g. Geant4 (see below). This is often enormously ressource-heavy, taking easily to 1 CPU hour per event –



And we need > billions of events!

http://geant4.web.cern.ch/

Event generation – Generative models to the rescue?

3) Simulate the response of the detector, using e.g. Geant4 (see below). This is often enormously ressource-heavy, taking easily to 1 CPU hour per event –

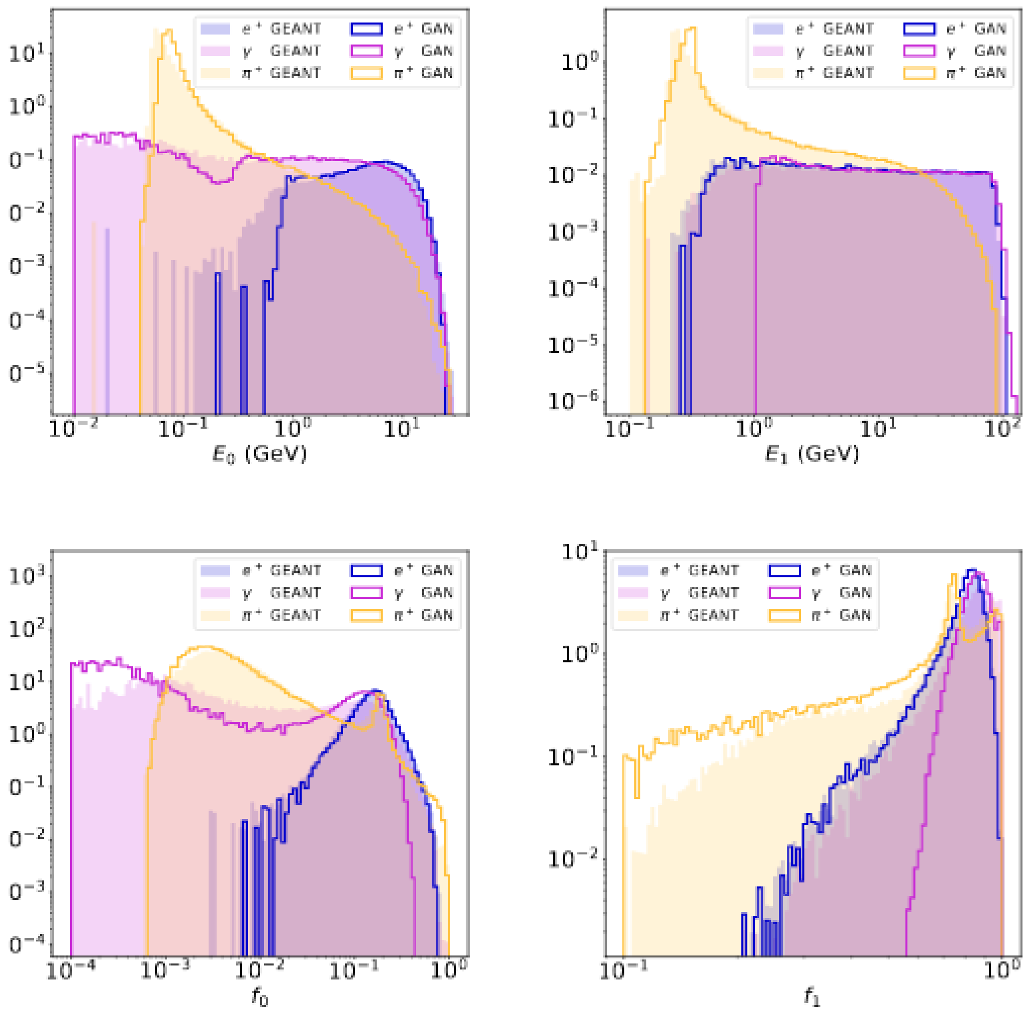
And we need > billions of events!

Possibility that generative models might help.

Problem: simulation of LHC collision events is costly

Solution: simulate smaller number of events, train e.g. generative adversarial networks on simulated events and/or real data, so you **ideally** you would have a hyper-fast generator of an unbounded number of events (obviously it is not that easy, but Gregor will tell you more).

GANs in Particle Physics: Fast simulation of ATLAS calorimetry (Calo GAN)



Comparison of shower shape variables, Calo GAN versus full Geant 4 simulation

See also: talk in ACAT 2017

https://arxiv.org/pdf/1705.02355.pdf

Preparing for the Unexpected

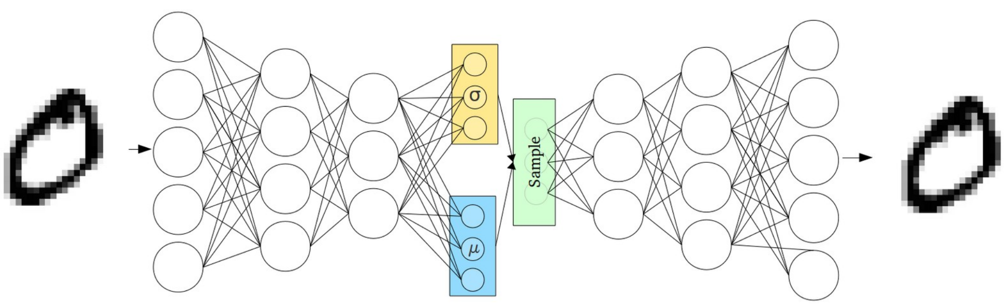
- Remember in the very first steps of our data processing chain we already had to discard most of the events?

  Problem: all our “classical” searches for new physics rely on some kind of idea of what new physics might look like. What if new physics manifests itself in a form that we did not anticipate?

https://arxiv.org/pdf/1811.10276.pdf

Variational autoencoders for anomaly detection

Idea: use an (variational) autoencoder to learn low-dimensional representations of a soup of "Standard Model" events. Apply to real data: flag events with high loss, they seem to be not quite as we expect events to be.



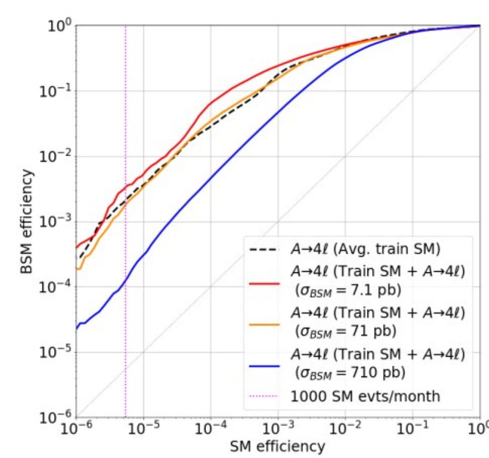


Figure 11. ROC curves for the VAE trained on SM contaminated with and without $A \to 4\mu$ contamination. Different levels of contamination are reported corresponding to 0.02% (σ = 7.15 pb - equal to the estimated one to have 100 events per month), 0.19% (σ = 71.5 pb) and 1.89% (σ = 715 pb) of the training sample.

Variational autoencoders for anomaly detection

Idea: use an (variational) autoencoder to learn low-dimensional representations of a soup of “Standard Model” events. Apply to real data: flag events with high loss, they seem to be not quite as we expect events to be.

frequency with which events from **unknown** physics are “flagged”

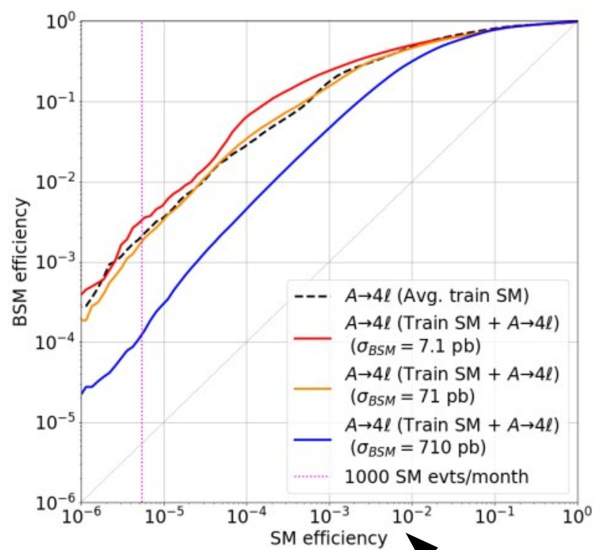


Figure 11. ROC curves for the VAE trained on SM contaminated with and without $A \to 4\mu$ contamination. Different levels of contamination are reported corresponding to 0.02% ($\sigma = 7.15$ pb - equal to the estimated one to have 100 events per month), 0.19% ($\sigma = 71.5$ pb) and 1.89% ($\sigma = 715$ pb) of the training sample.

frequency with which events from **known** physics are “flagged”

Honorable mentions: ML in particle physics

- arxiv:1608.08225

Why does deep and cheap learning work so well?*

Henry W. Lin, Max Tegmark, and David Rolnick
Dept. of Physics, Harvard University, Cambridge, MA 02138
Dept. of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139 and
Dept. of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139

Physics	Machine learning
Hamiltonian	Surprisal $-\ln p$
Simple H	Cheap learning
Quadratic H	Gaussian p
Locality	Sparsity
Translationally symmetric H	Convnet
Computing p from H	Softmaxing
Spin	Bit
Free energy difference	KL-divergence
Effective theory	Nearly lossless data distillation
Irrelevant operator	Noise
Relevant operator	Feature

TABLE I: Physics-ML dictionary.

A dictionary with ML terms and “similar” physics terms.

Question: in e.g. computer vision the space of all possible image is enormous! How can machine learning with training samples < 10^{10} work at all?

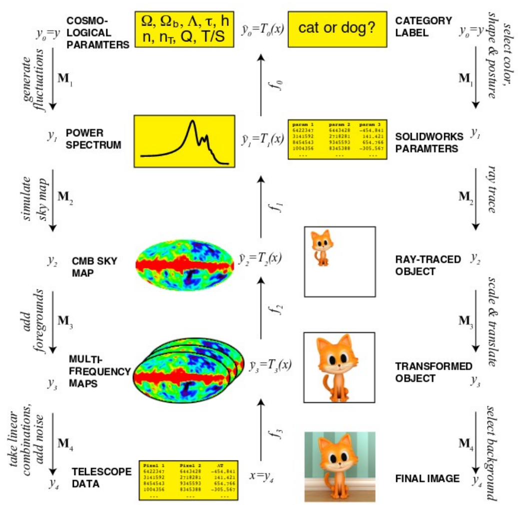
Answer: by exploiting compositeness, locality, symmetries (e.g. ConvNets = translation invariance!), polynomial negative log-likelihoods, etc etc

Honorable mentions: ML in particle physics

- arxiv:1608.08225

Why does deep and cheap learning work so well?*

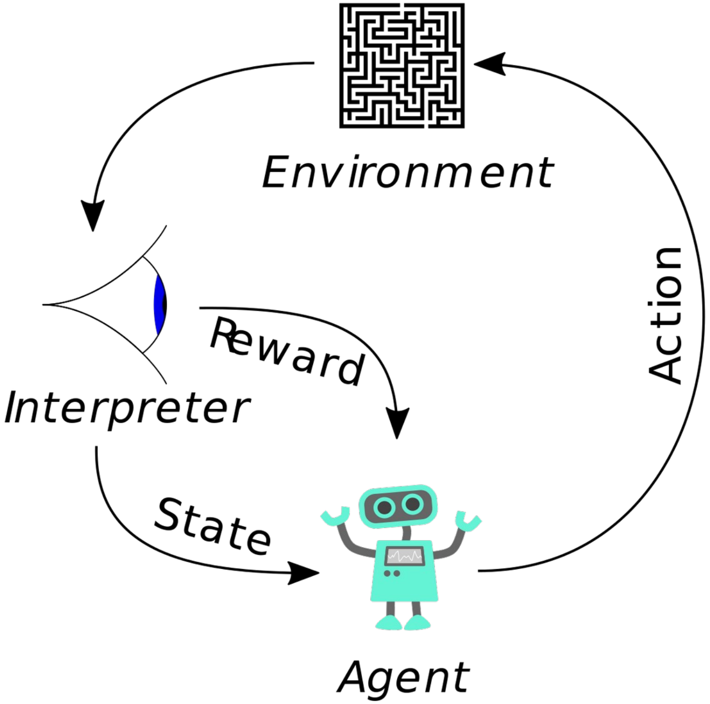
Henry W. Lin, Max Tegmark, and David Rolnick
Dept. of Physics, Harvard University, Cambridge, MA 02138
Dept. of Physics, Massachusetts Institute of Technology, Cambridge, MA 02139 and
Dept. of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139



The question of physical / thermodynamic foundations of (machine) learning arises. Can we hope to improve learning from such analogies?
Though to be fair: attempts at information theoretical groundings are currently more hotly debated, see e.g. Tishby et al., see e.g. https://arxiv.org/pdf/1703.00810.pdf

Recap: reinforcement learning (in a single slide)

Reinforcement learning has **not** yet taken off within particle physics.
Recap reinforcement learning:



Reinforcement learning:
The world is in a state “s” of the set of all states “S”
The agent can perform an action “a” of the set of all actions “A”.
Learn the policy “p: s → a” that maximizes expected future reward “r”.
We should see applications in “control theory” → calibration and maintenance of running detectors. But we do not. Why?

To repeat (I)

- Particle physics in the 21st century is an Exabyte-scale endeavour

- There is a seemingly endless number of applications of supervised learning problems or all stages of our data processing pipelines. Our data structures are often unusually complex, but typically fewer “layers of abstraction” are to be learned.

- We use self-supervised learning techniques for simulation of data and anomaly detection algorithms.

To repeat (II)

- We make use of FPGAs and explore more exotic hardware.
- Our theorists are using machine learning to interpolate between their calculations, to find “classes” of solutions, and for simulation.
- People ponder the relationship of learning with information theory and statistical physics.
- Reinforcement learning for detector control seems an interesting topic for the future.
- We are (more than the experts) concerned with estimating errors on the predictions – see next topic!

Next

Machine Learning in Particle Physics (II) – Uncertainties and Correlations

https://github.com/WolfgangWaltenberger/winterschool →