End-to-end deep learning at ~~LHC~~ particle physics experiments

University of Bristol Particle Physics

University of BRISTOL

- Very active department both in terms of analysis and detector development.
  - 8 academics
  - 12 postgraduate researchers
  - ~20-25 PhD students

DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

LHCb

CMS

milliQan
SoLiD
ILC

Overview

- ❑ Online event selection
  - Trigger systems overview
  - Fast deep learning on custom hardware (CMS experiments)

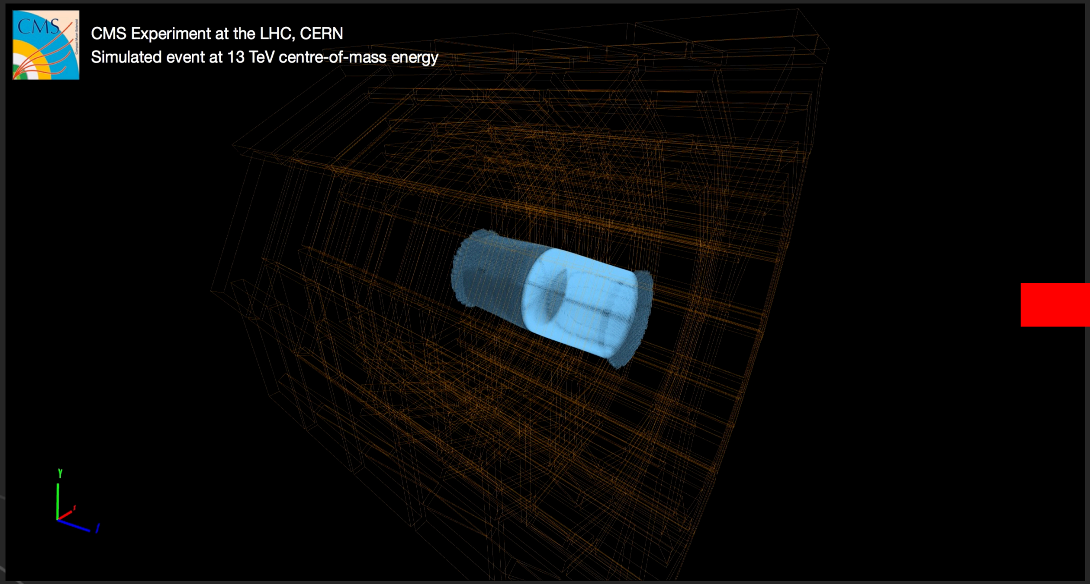
- ❑ Offline event classification
  - Autoencoder waveform classification (LZ experiment)
  - Graph representation of LHC events (CMS experiment)

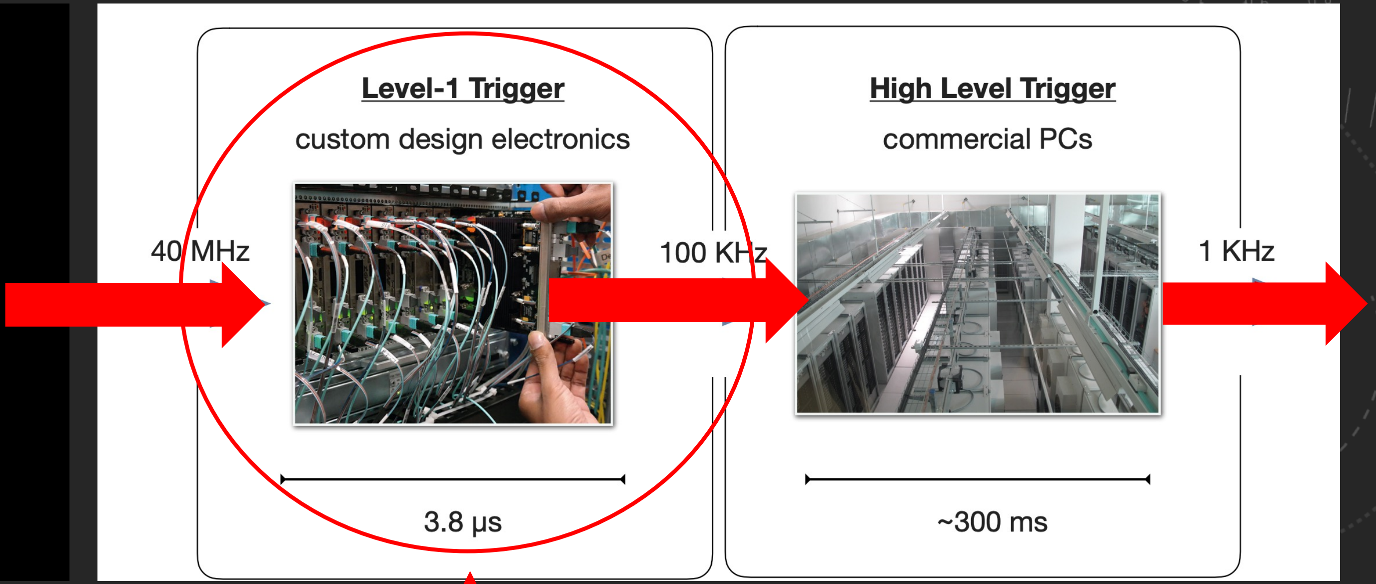
- ❑ Underpinning: simulation
  - Graph adversarial neural networks (GANs) for robust and fast detector simulation synthesis (SHiP & LHCb)

Online event selection

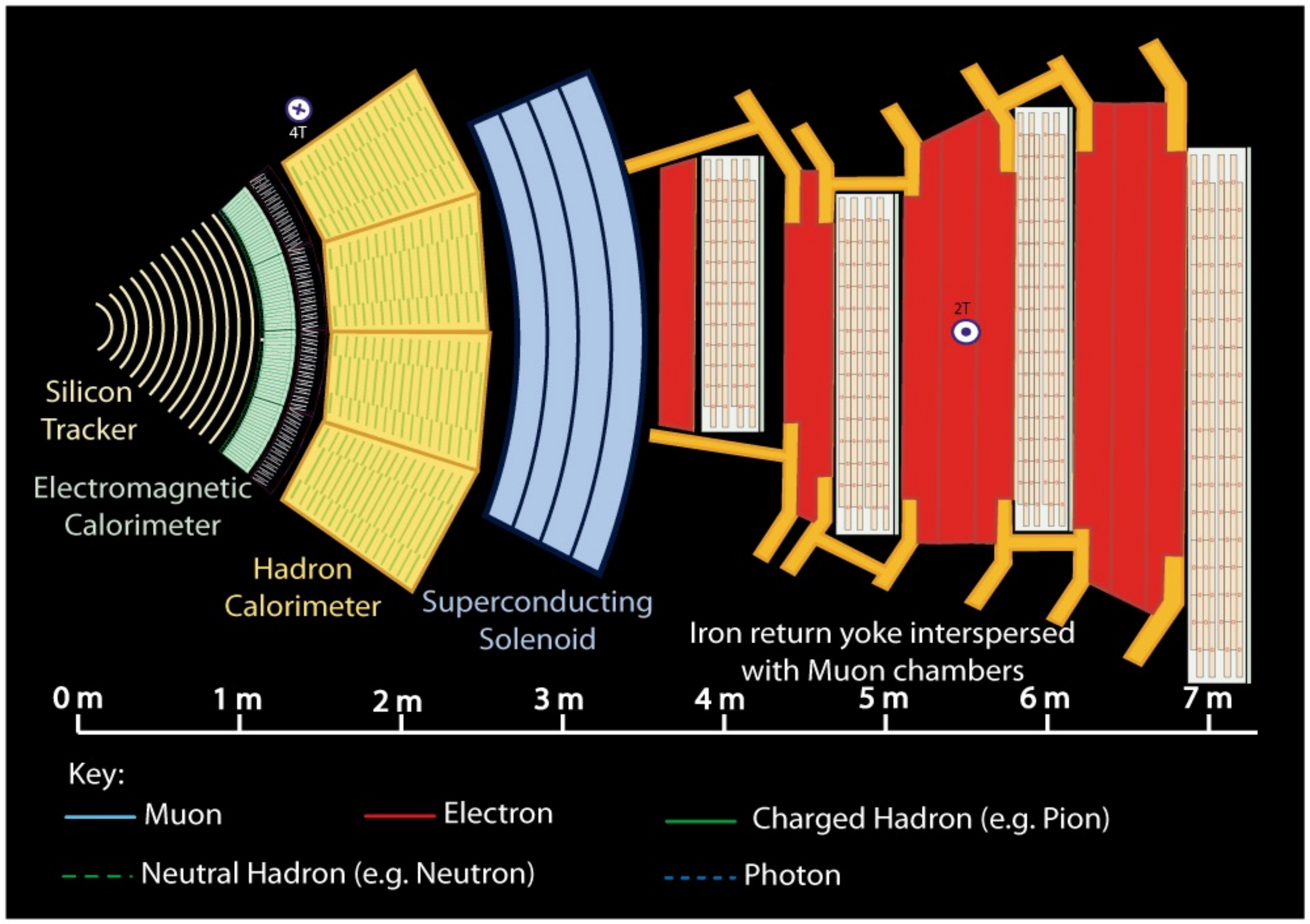
Trigger system

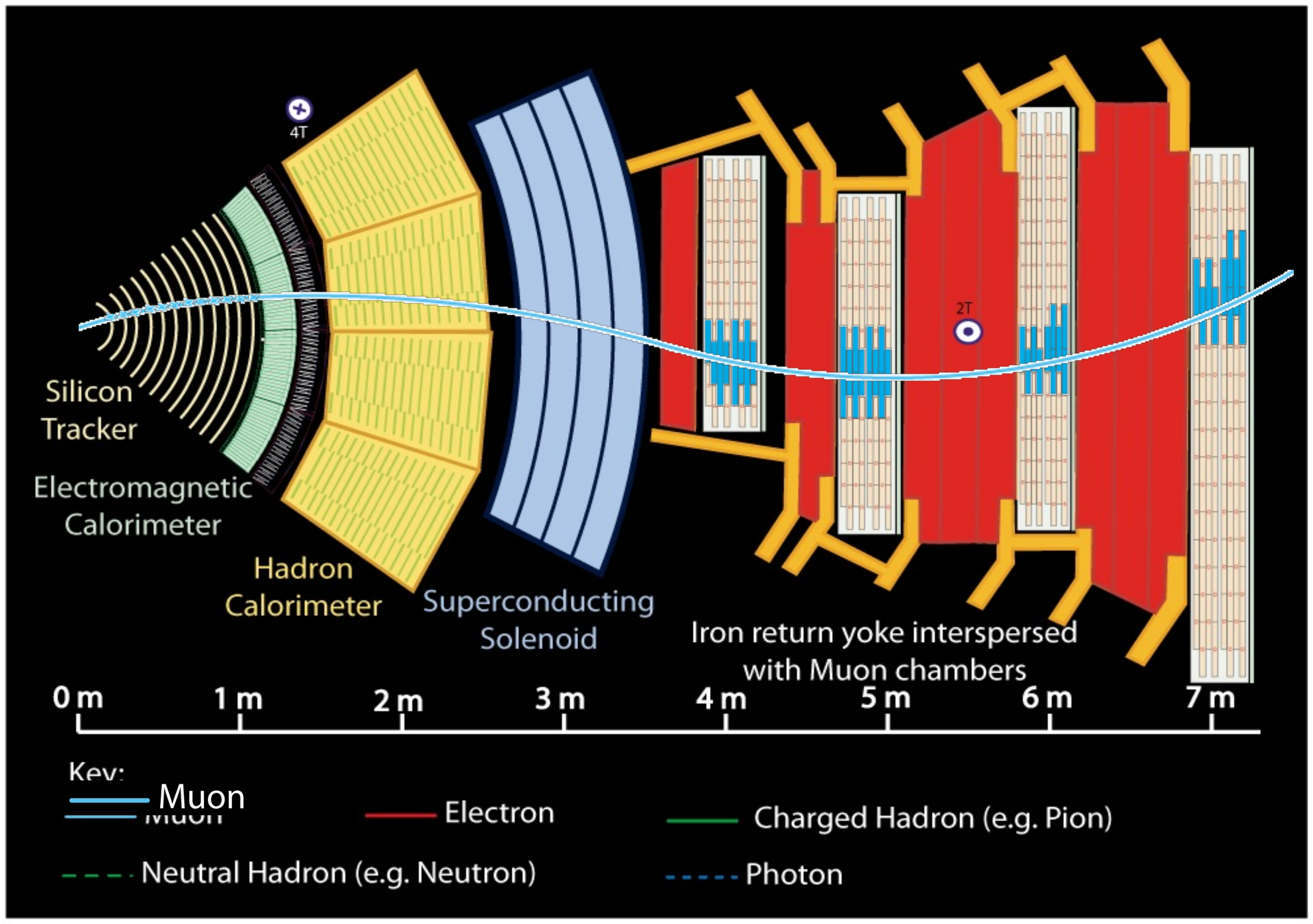
- General purpose detector capturing proton-proton collisions
- CMS ”sees” 40 million collision / second
- Can only record ~1000 collisions / second
- Need to **quickly** and **precisely** select interesting events to store for further analysis
- Interesting events are rare compared to boring background events ~ needle in a haystack!

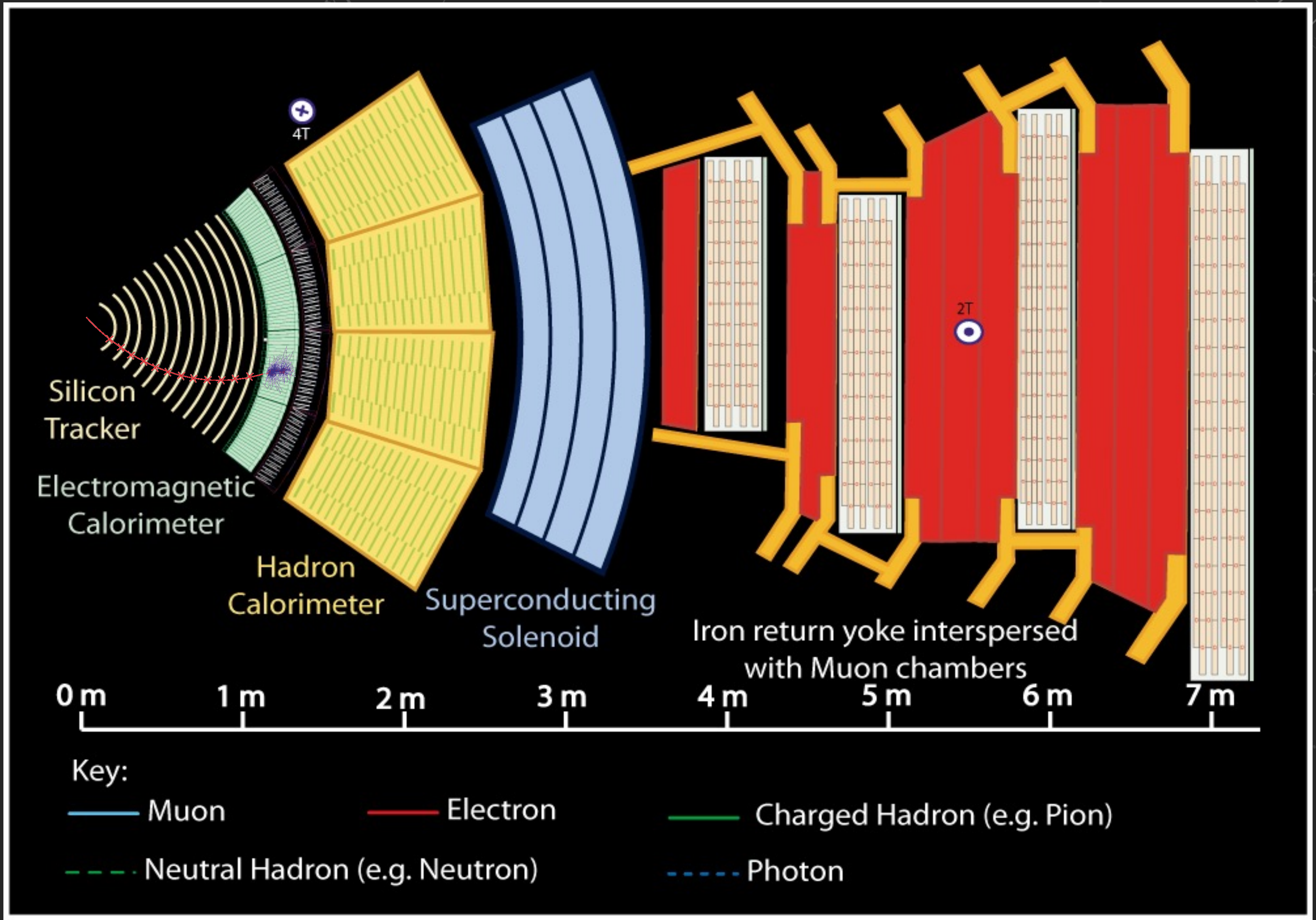


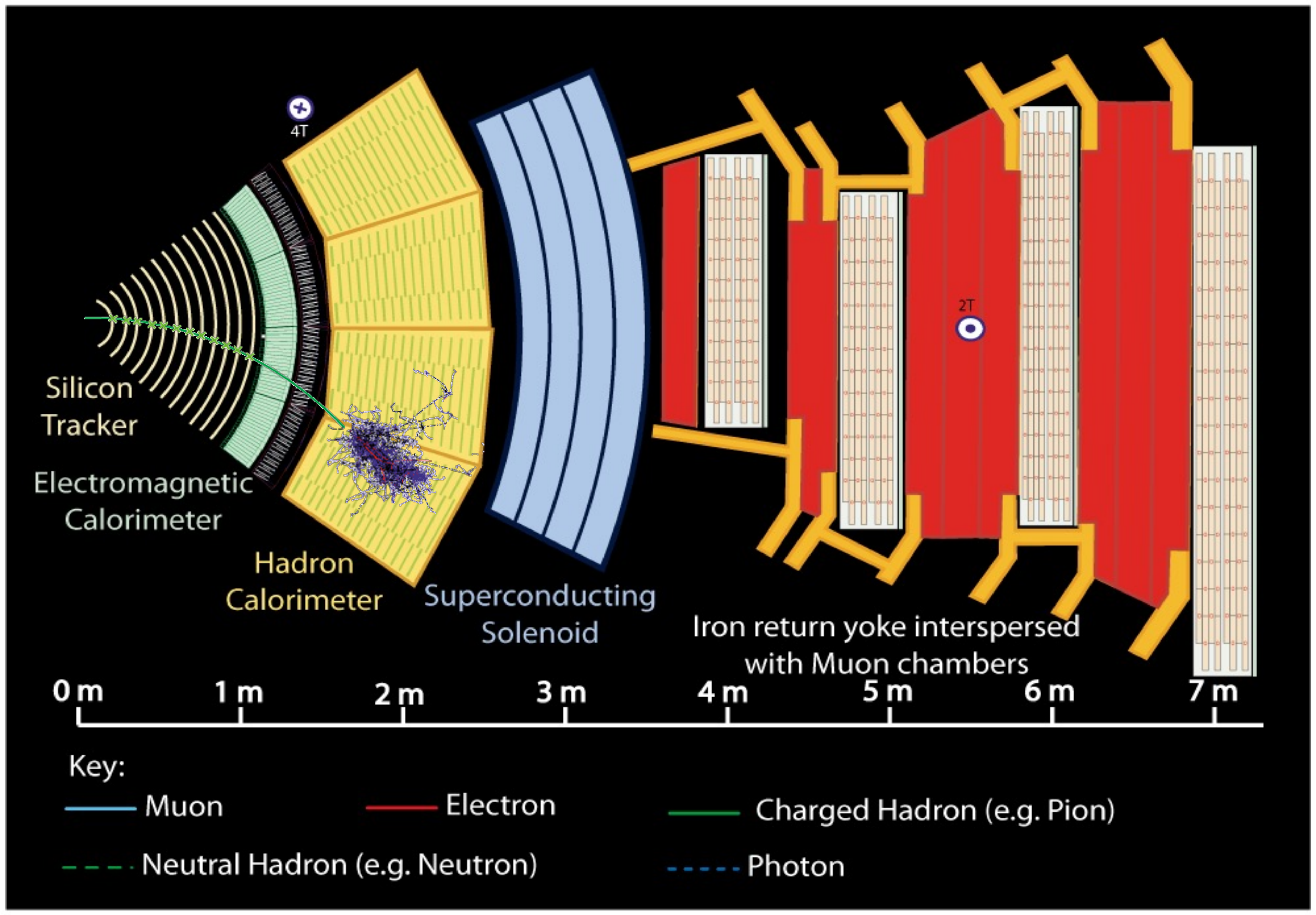


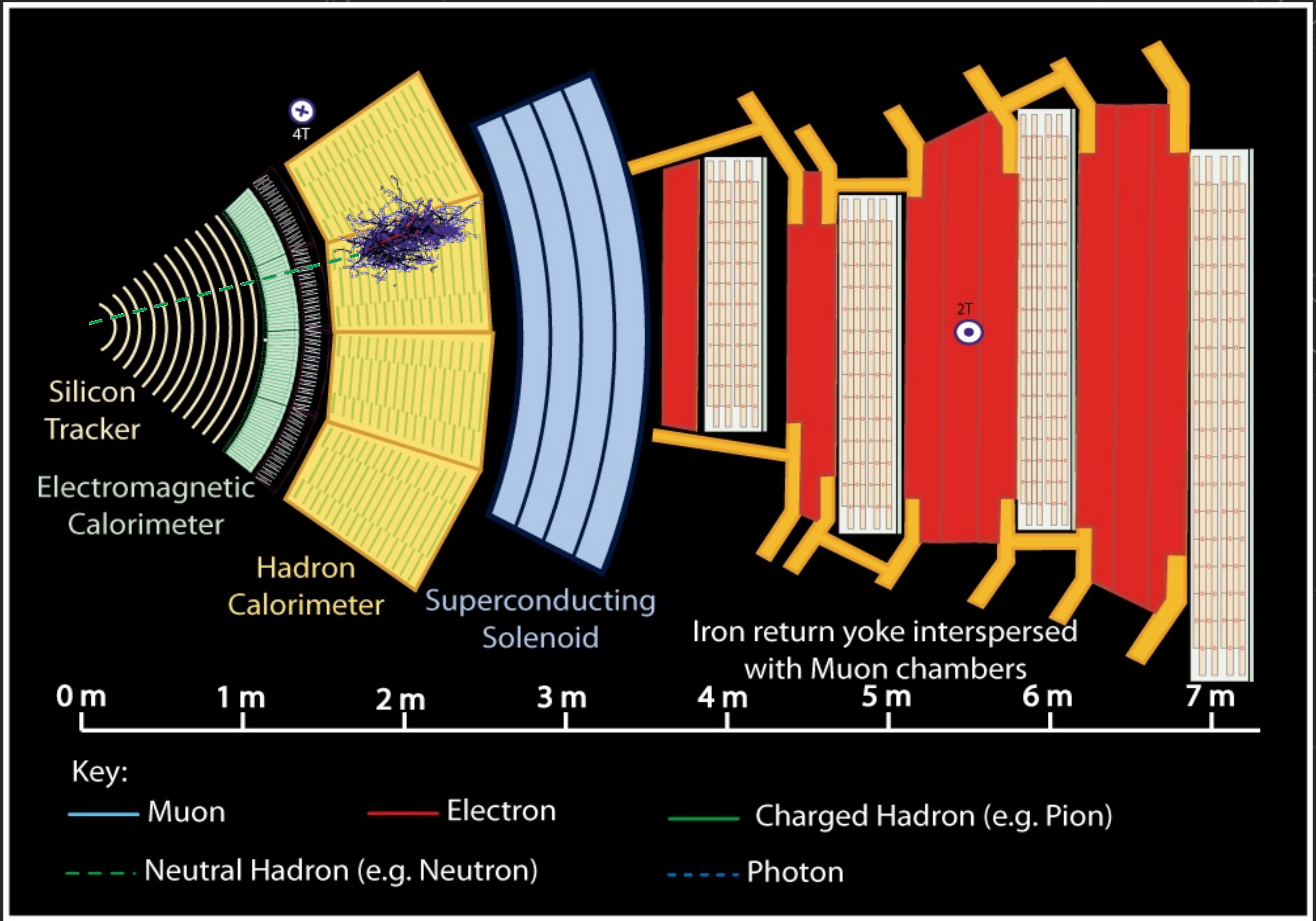
Deep-learning already here!

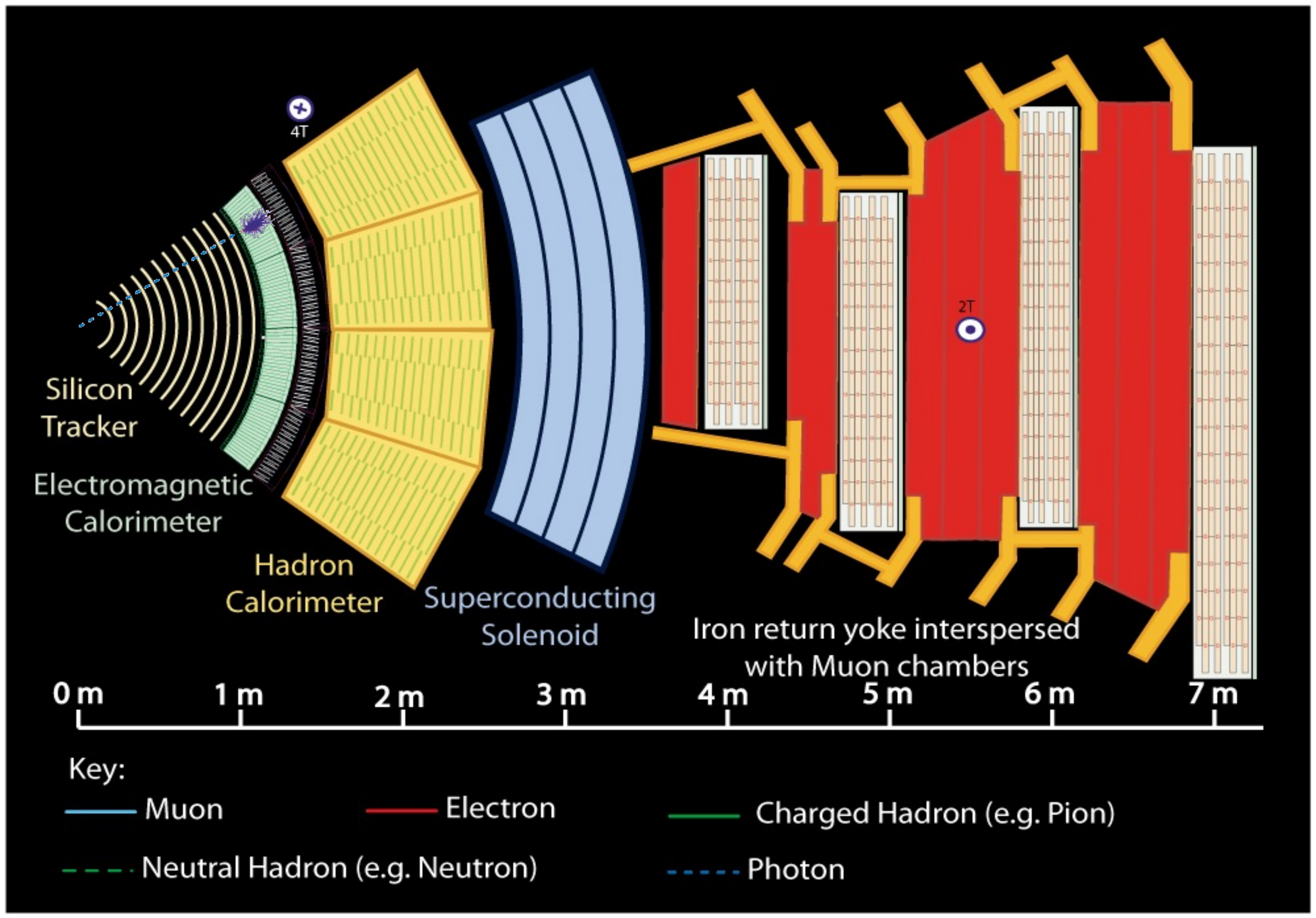

4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon


4T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0 m
1 m
2 m
3 m
4 m
5 m
6 m
7 m
Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon

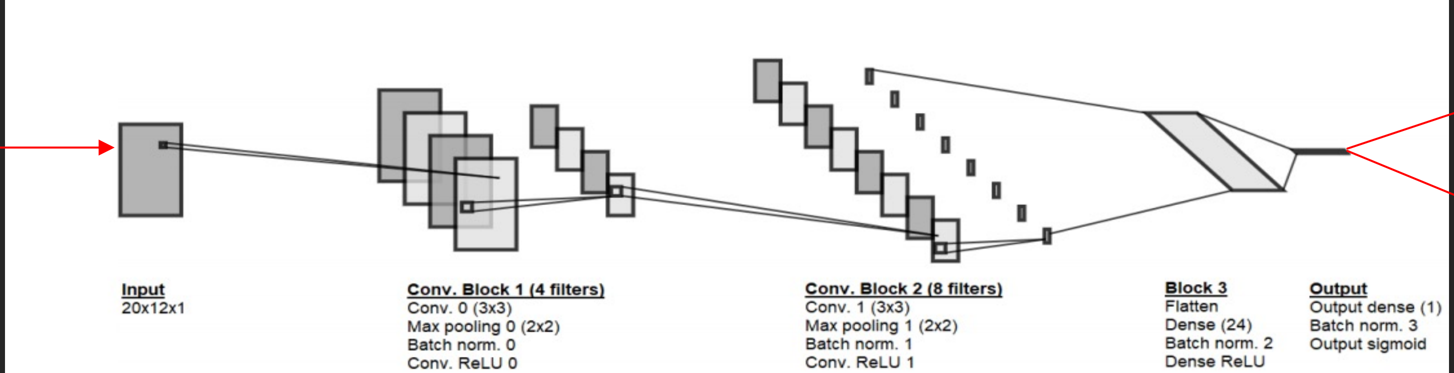
Human vs. Machine

- Typically:

```
def my_trigger_algo(event):
    if has_4_jets = True and total momentum > 400 GeV :
        keep
    elif has has_4_jets = True and missing momentum > 200 GeV:
        keep
    else:
        delete!
```

- Or, let a convolutional neural network decide:
  - Process events as 2-D histograms (images) of reconstructed physics objects binned in detector angular acceptances.

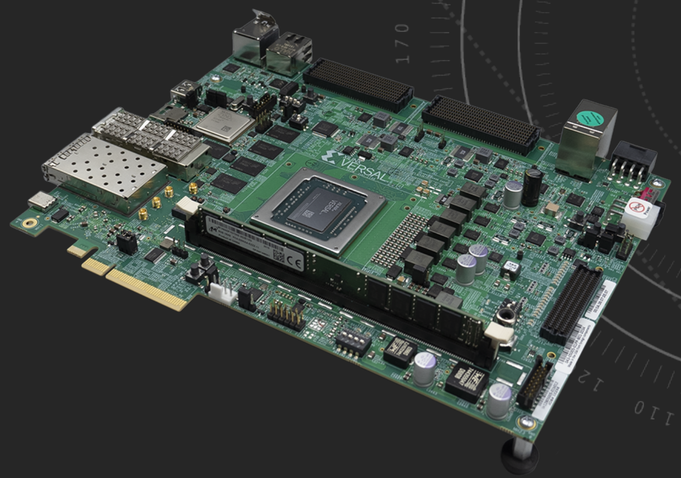
Event represented as image



signal

background

Deep Learning on FPGAs

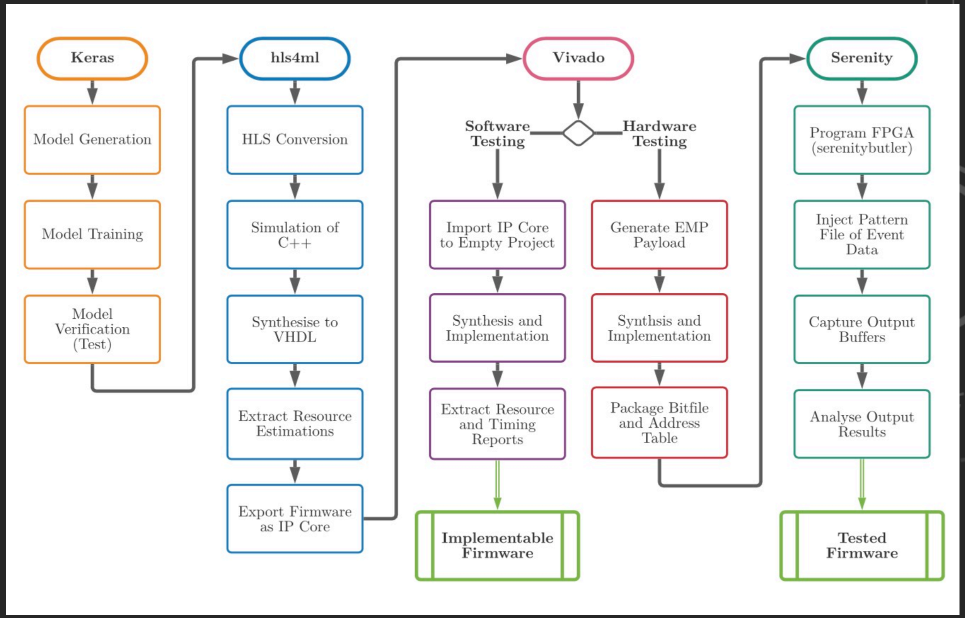


- ❑ Field-programmable gate array
  - ✓ Very quick (process an even in ~micro-second)
  - ✓ Highly parallelizable
  - ✓ Specification to meet user needs

  - ▪ Firmware packages need to be implemented using VHDL/HLS
  Very difficult to implement a neural network at such low-level.

- ❑ Advent of hls4ml arXiv:2103.05579
  - ✓ Synthesizes HLS firmware package from TensorFlow PyTorch implementation!

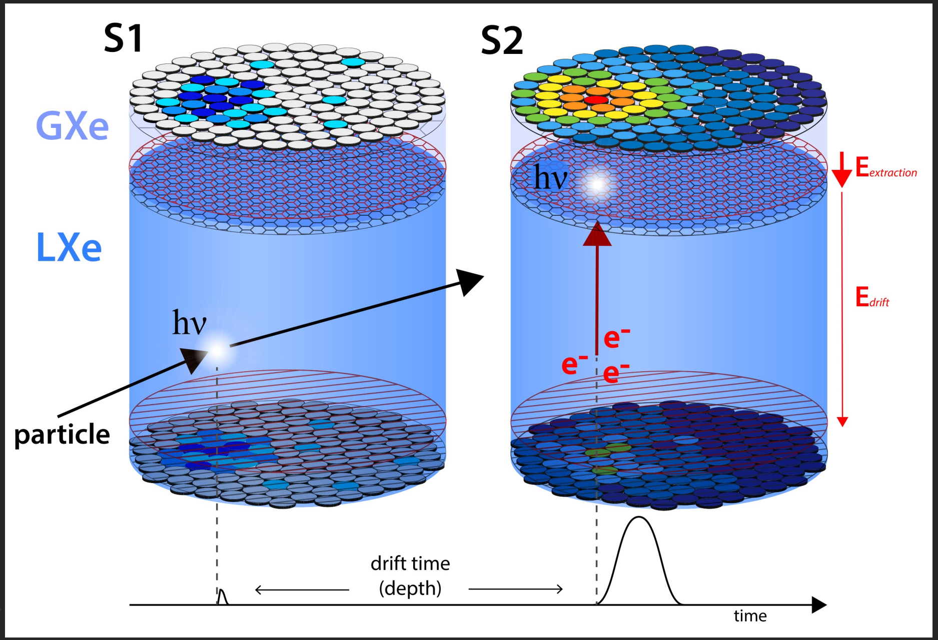
    Current pipeline

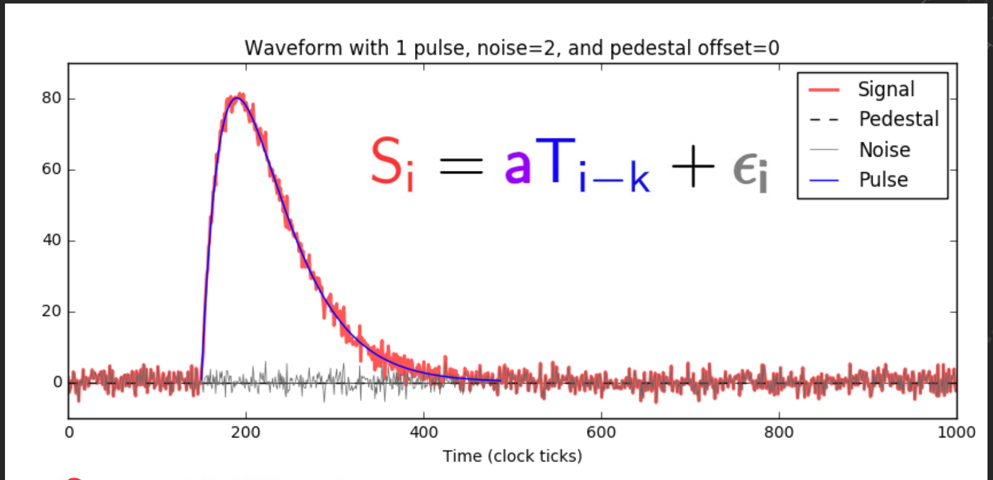


Offline event selection

Waveform analysis

- Recognising signals at the LZ experiment:
  - Scintillation occurs within the detector from incoming particle (possibly Dark Matter)
  - Event produce time-series waveform on which both pulses are visible
  - Waveform parametrised in terms of drift times, amplitudes, etc..





S_i The i-th ADC sample, is given by

a the pulse amplitude, multiplied by

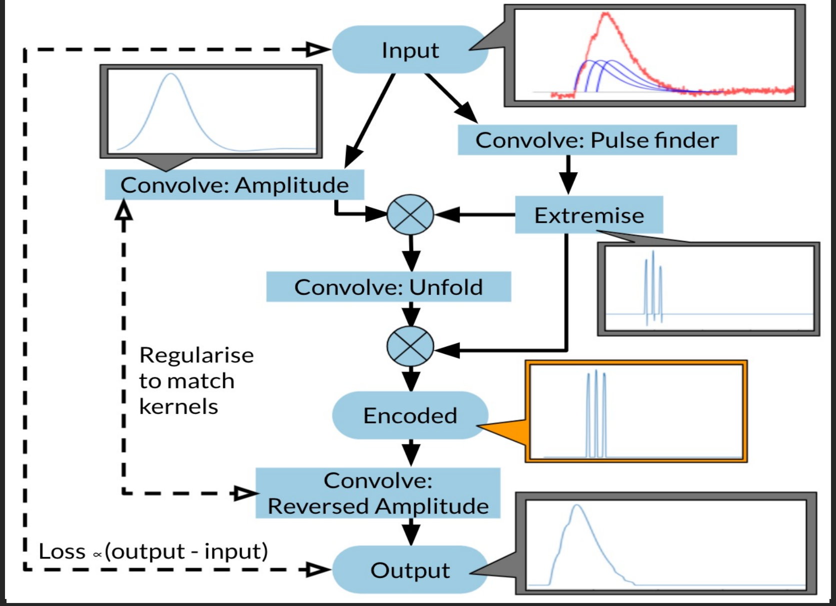
T_{i-k} the pulse shape which happened at *k*, plus some

ϵ_i noise

The matched filter method: fit known template to waveform for physics extraction

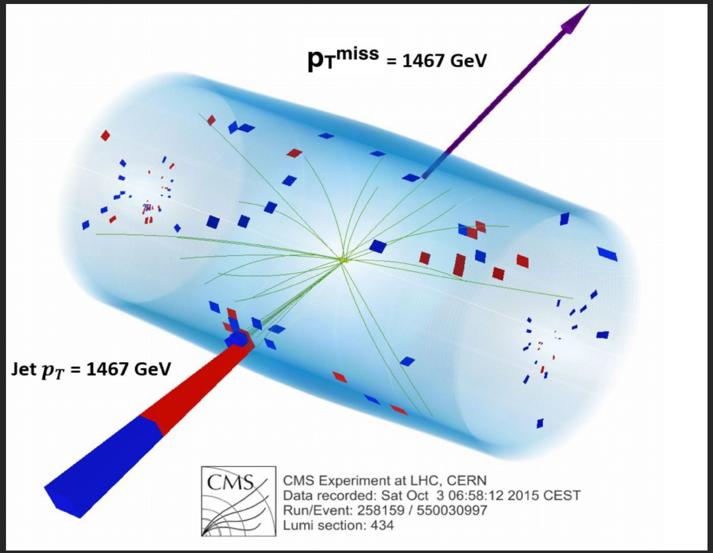
Autoencoder

- If template is known, convolution offers the best possible signal extraction.
- Convolution layers in CNN **are** matched filters
- Strategy:
  - Use autoencoder architecture (loss function defined by distance between input & output)
  - Two convolution layers, parametrised by learnable kernel of weights which replicate amplitude & pulse finding
  - Extremise encoded (latent) space for amplitude analysis (extraction of physics)
  - Convolve sparse encoded space with learn filter to return to original waveform

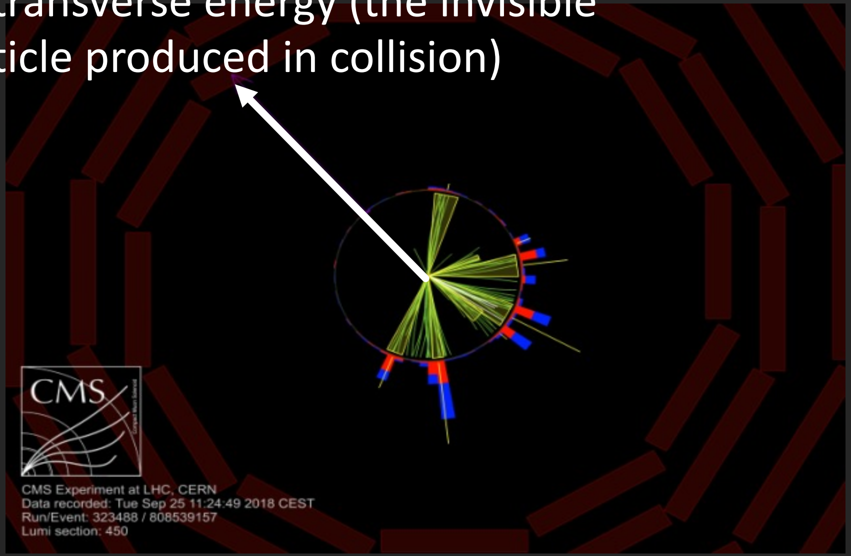


Invisibly decaying Higgs Boson H→ inv

- The Higgs Boson is of interest in search for "Beyond Standard Model Physics" since it couple proportionally to mass.
- BSM particle (e.g., Dark Matter) couple very weakly to SM and hence leave no trace in the detector
  - Can use a trick: Since initial transverse momenta of incoming protons = 0, we can look for momenta imbalance post collision (i.e. , we may infer their existence from missing energy in the collision)
  - P_T^{miss} is the observable we use



Missing transverse energy (the invisible particle produced in collision)



P_T^{miss} + jet final states in the CMS detector.

- Unfortunately SM particles, namely neutrinos or leptons produced outside of detector acceptance will give an identical signature! Need some way of classifying these processes…

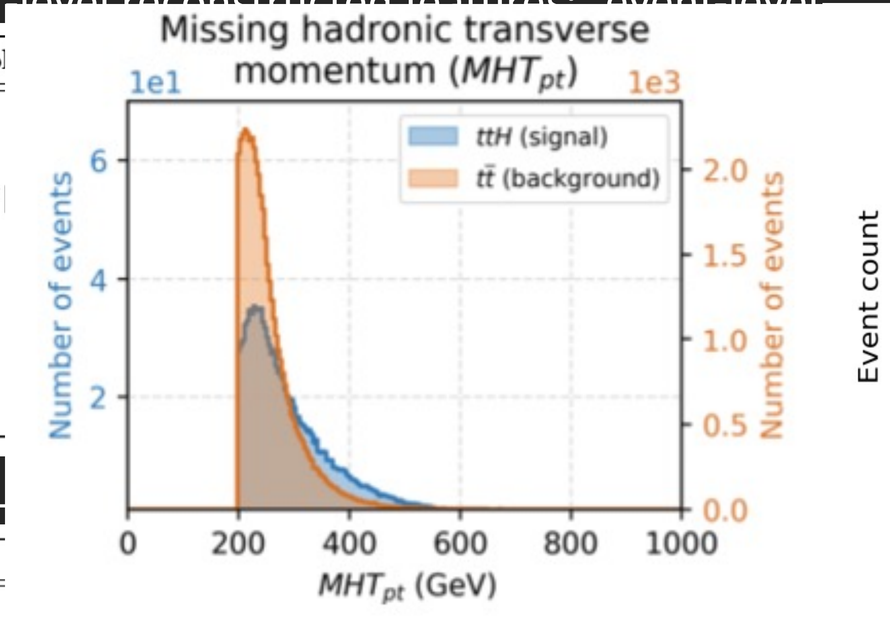
Signal vs Background classification

H→ inv

- Capture event topology by looking at both "object" and "event" level features
  - Process both data types by dedicated deep learning architectures
- Goal: discriminate ttH signal from $t\bar{t}$ decaying semi-leptonically (where lepton evades detection)

Higher-level reconstructed features: "event-level"

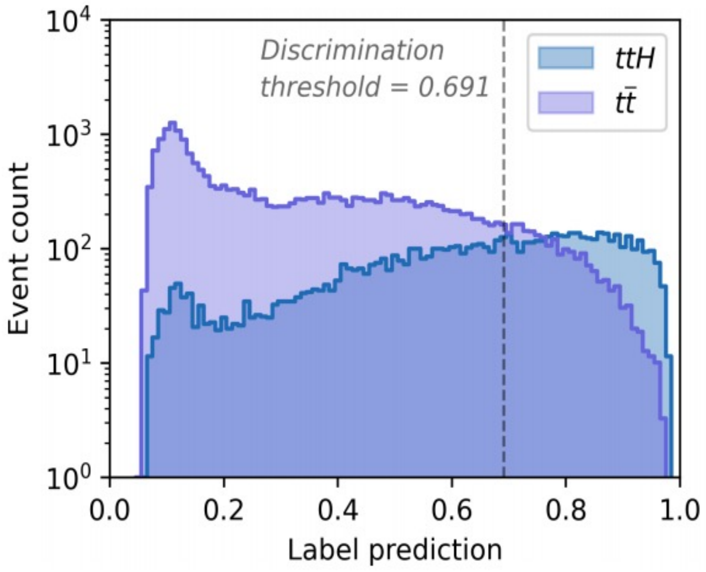
Symbol
H_T
$\lvert H_T^{miss} \rvert$
m_{jj}
n_j
n_{bj}
χ_{min}
$\hat{\omega}_{min}$
$\tilde{\omega}_{min}$

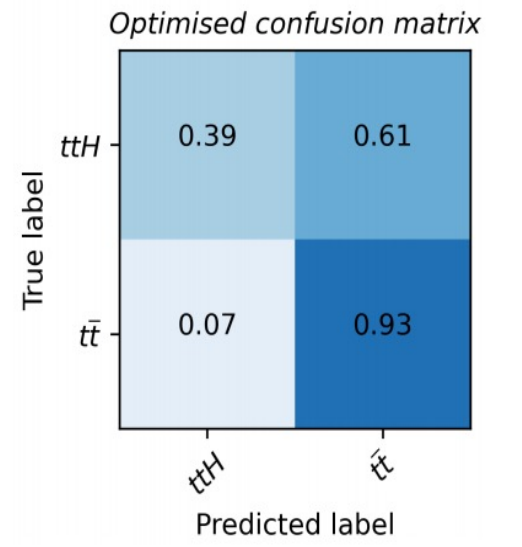


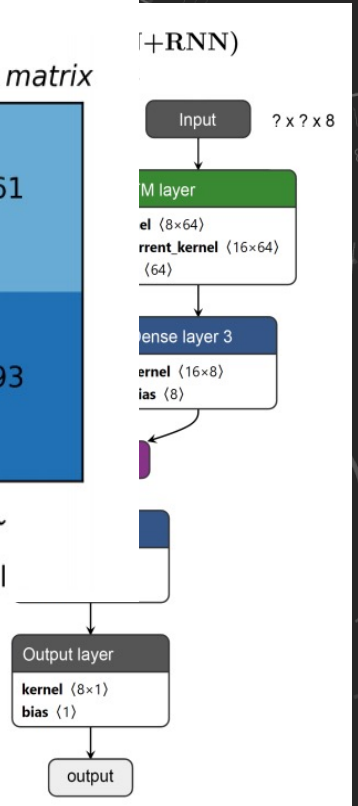
Lower-

Symbol		
$\lvert p_T \rvert$		
m_{jet}	Jet_mass	Total mass of all jets
ϕ	Jet_phi	ϕ of all jets
η	Jet_eta	Pseudorapidity of all jets
Ω_{jet}	Jet_area	Angular area of jet projected on the η-ϕ plane
d	Jet_btagDeepB	Discriminator output of DeepB tagging algorithm[7]
$\Gamma^{\pm}$	Jet_chHEF	Charged hadron energy fraction
Γ^{0}	Jet_neHEF	Neutral hadron energy fraction

Network ensemble:



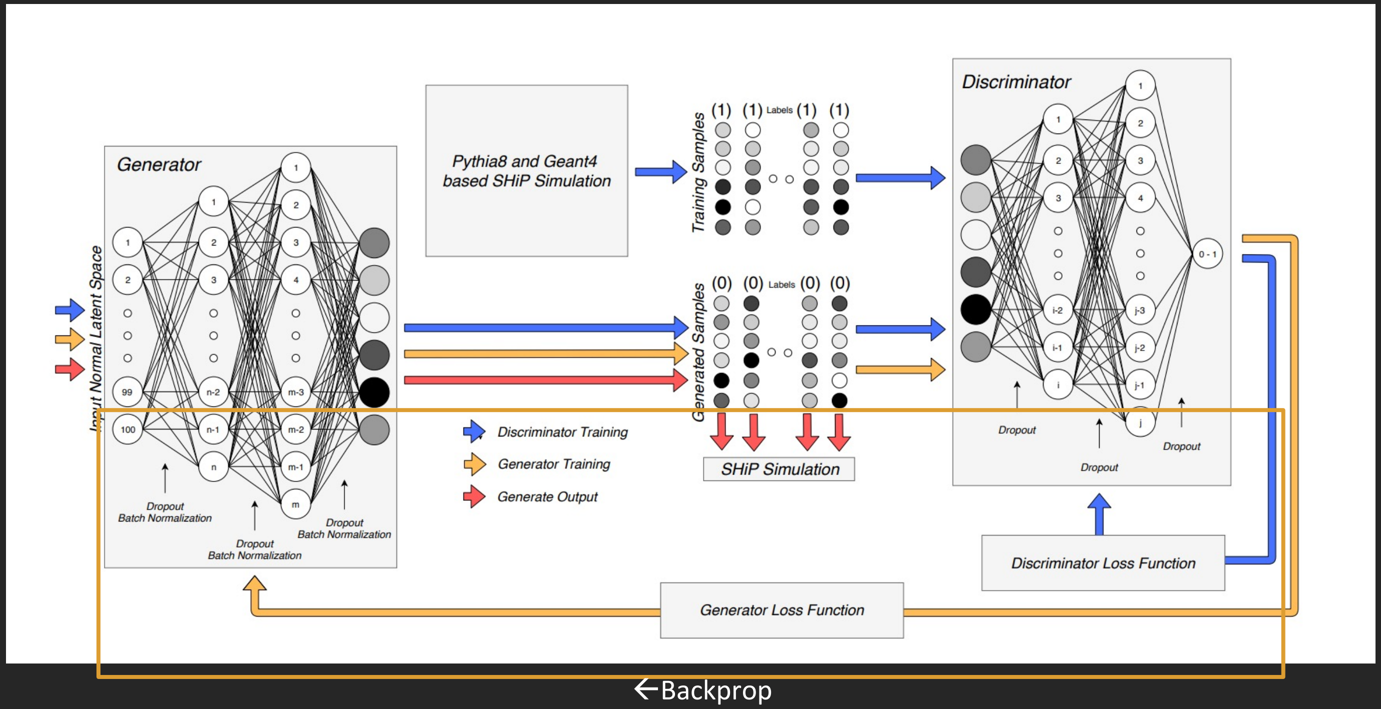




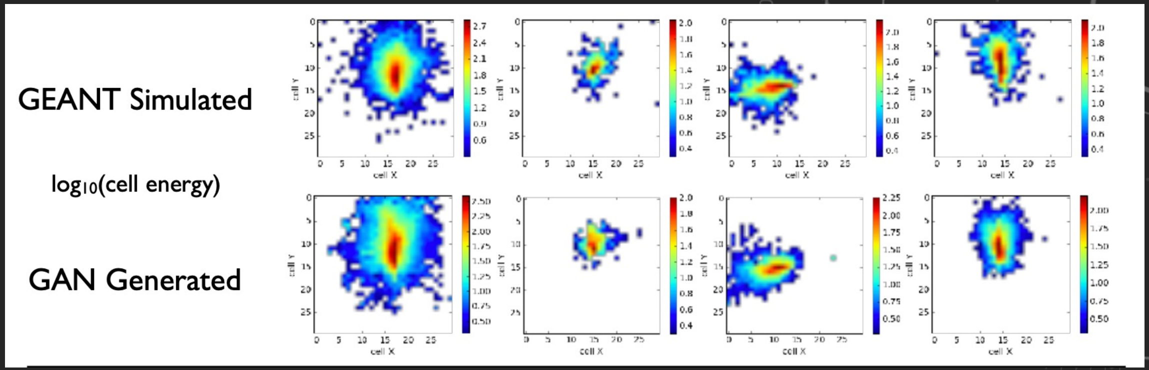
The underpinning: Simulation

GANs for detector simulations

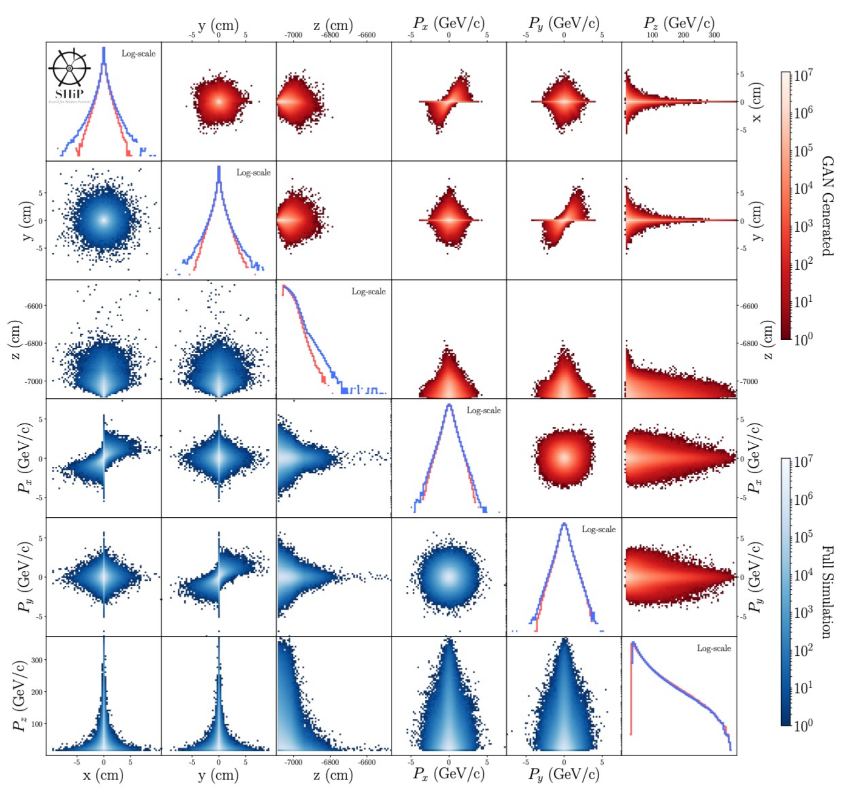
- The vast majority of ML in particle physics relies on supervised learning (labelled data)
- 80% of HEP computing resources are used for MC simulation in HEP experiments
- Cannot relay on constant rise in computer performance
  - Need for fast, reliable and less expensive simulation.
- Lends itself to ”teacher-student” architecture, a.k.a. Adversarial Networks (GANs)
- One of the most intensive tasks is *jet image* production – 2D energy deposits in the hadronic calorimeter



Benchmarking GANs



Jet images at LHCb HCAL



Muon features for GAN based and fully simulated (PYTHIA & GEANT4) muons produced at the SHiP experiment: arXiv:1909.04451v2

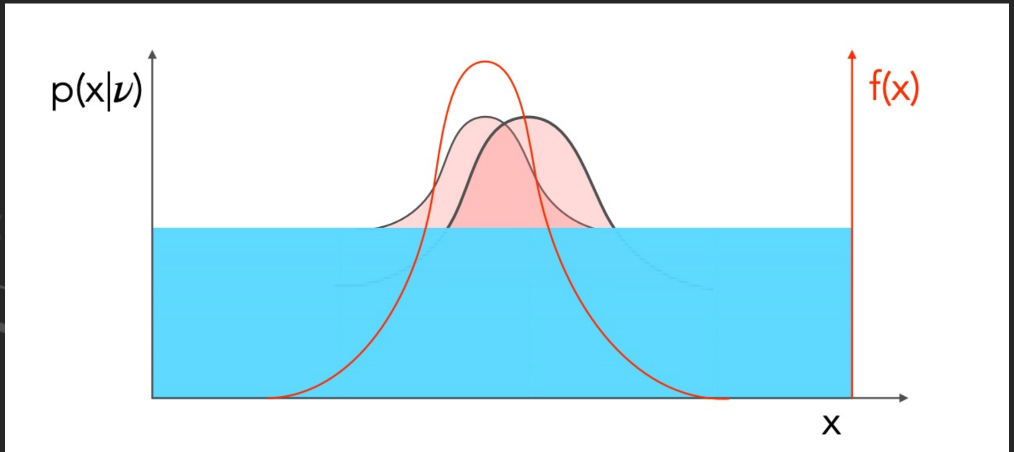
- So, where are we?
  - Fast simulation for LHC detectors in Run III is the natural target
  - Generative space is still evolving
  - Results give early promise but not quite production quality yet

Speed-up on single `Nvidia Pascal` card

Target simulation method	Muons produced in 5 minutes	Time to simulate single muon (s)
`Pythia8` and `GEANT4`	~ 1	1.1×10^{-1}
GAN (CPU)	7.5×10^{5}	4.0×10^{-4}
GAN (GPU)	3.5×10^{6}	8.6×10^{-5}

Very quick word on experimental uncertainties

- Event classification example, i.e. signal / background classification for downstream statistical inference
- Relies on supervised learning (simulated, labelled data)
- X = some discriminating variable and p = probability distribution
- Typically, would choose some threshold in p to ”cut on” and gain most sensitivity to signal
- **BUT** what if have mismodelled regions of X?
- If X shifts then so does our “signal peak”, however, classifier distribution is fixed → loose sensitivity to signal
- We want our classifier to learn correlations between features in some higher-dimensional space and **not** relay on the absol value of X.
- Need to **decelerate** X from classifier output → custom loss function: $\mathcal{L}_\theta = \sum_{i=1}^{N} \ell(f_\theta(x), y)_i$ + $\mathrm{dCov}^2 = \frac{1}{n}\sum_j \sum_k \hat{x}_{jk}\hat{y}_{jk}$



DisCo Fever: Robust Networks Through Distance Correlation
arXiv:2001.05310

In summary…

- University of Bristol active in both software and hardware development spanning a wide range of particle physics experiments;
  - The volume of data produced at these experiments is accelerating deep learning R&D on unpresented scales.
  - Next generation experiments at the LHC & beyond leaning heavily into deep learning solutions (live DL data-streams already becoming a reality at LHCb with Apache Kafka)!

- Deep learning within the particle physics domain brings with it a whole host of challenges:
  - Fast, reliable and implementable DL in online classification,
  - The ever increasing event rates,
  - Explainable AI (physicist not comfortable with black-boxes),
  - Training under experimental uncertainties.

- Contact:
  - CERN: Maciej.mikolaj.glowacki@cern.ch
  - Bristol: ys20884@Bristol.ac.uk