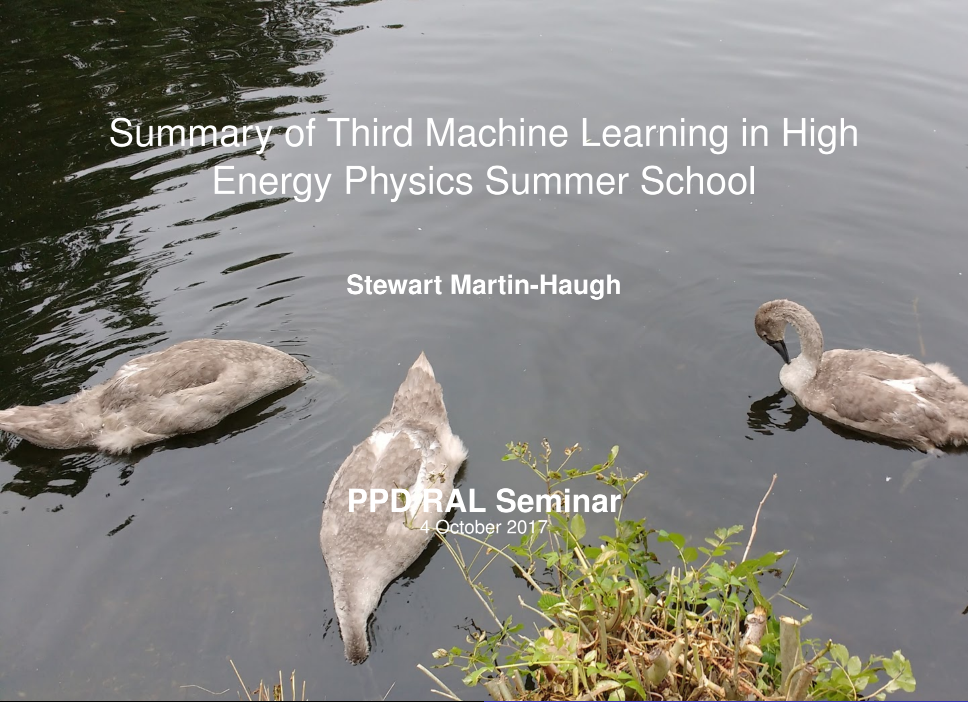


# Summary of Third Machine Learning in High Energy Physics Summer School

**Stewart Martin-Haugh**

**PPD RAL Seminar**

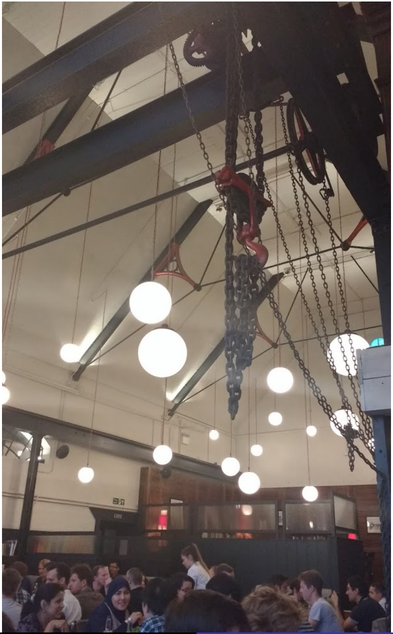
4 October 2017

# Introduction

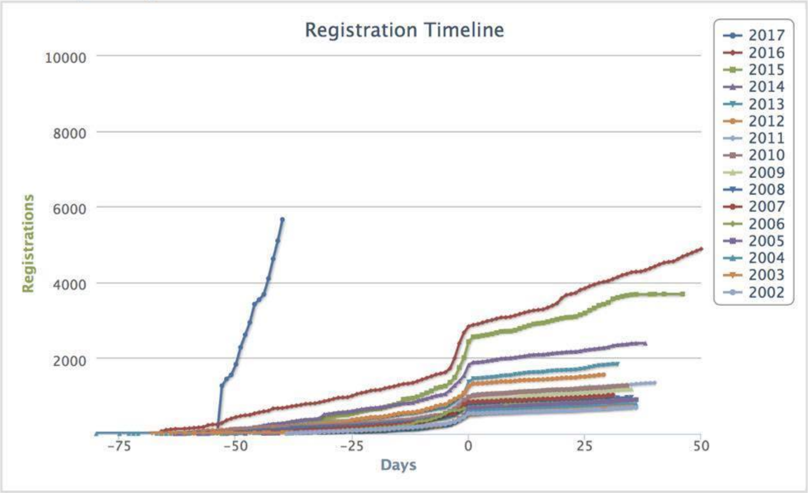
- Third Machine Learning in High Energy Physics Summer School held at University of Reading, 17-23 July 2017
- 60 participants: PhD students and early-career postdocs
- Very broad overview of topics in machine learning
- Indico and Github
- Each day structured into lectures and hands-on sections, with topical seminars

# School

# School Dinner at the Bel and Dragon, Reading



# Machine learning - hype?




Registrations for Neural Information Processing Systems conference (0=early registration deadline) reference

# Topics

- Basics: linear regression and gradient descent
- Decision trees
  - Boosting and bagging
- Neural networks
  - Convolutional NNs
  - Recurrent NNs
  - Unsupervised learning
  - Deep neural networks

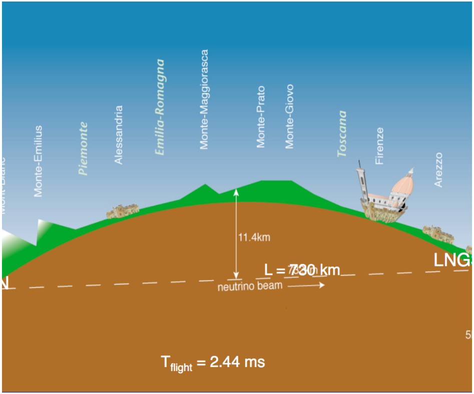
# Tools

- pip, numpy, pandas, matplotlib, seaborn, jupyter, scikit-learn, keras, theano, tensorflow
- Basically, lots of python plugged together
- No ROOT/TMVA etc

# Challenge

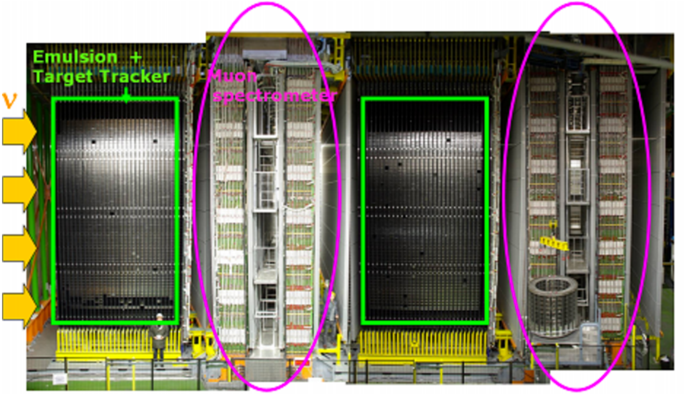
- Kaggle challenge during school
- OPERA emulsion data
- Prize: T-shirt, prestige, give talk at SHiP collaboration meeting

# From CERN to OPERA, overview




- Goal: find neutrino oscillations
- Detector: photo emulsion
- Data taking: 2008-2012
- Results: 5 $\boldsymbol{\nu}_{\mu} \longrightarrow \boldsymbol{\nu}_{\tau}$ observed,
- 2015 – Nobel prize in Physics for discovery of neutrino oscillations
- http://operaweb.lngs.infn.it

# OPERA detector



# OPERA ECC brick

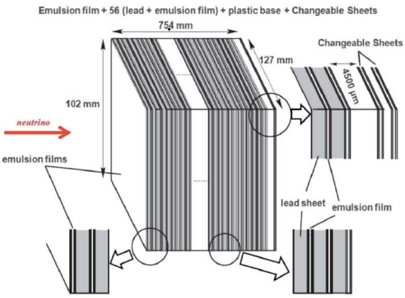
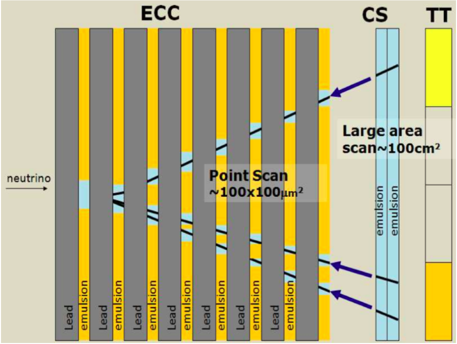
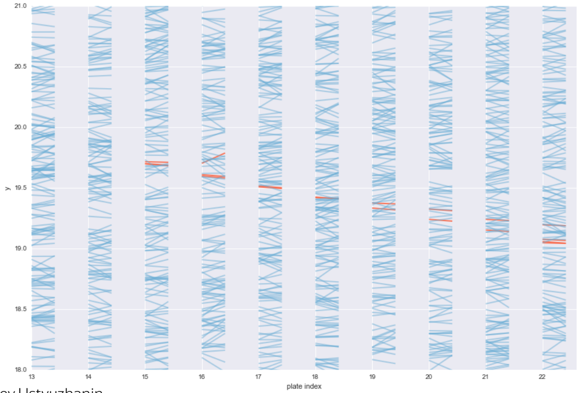



Figure 2.4 – Schematic structure of an ECC brick.



# Brick structure




Atomic track element: **basetrack**

- x,
- y,
- z,
- TX,
- TY,
- $\chi^2$

# Given

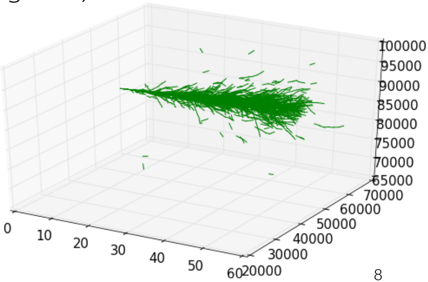
Data Background: 1 brick, ~ $10^6$ base tracks (signal=0)

MC Signal: simulation of pure EM showers
(100 events, $10^2$-$10^3$ basetracks per shower) (signal=1)

DS_1_train.csv, DS_1_test.csv,

Origin of the mother-particle is known (x, y, z, TX, TY, \chi2)

DS_1_electron_train.csv,
DS_1_electron_test.csv



# Challenge

Develop algorithm that can

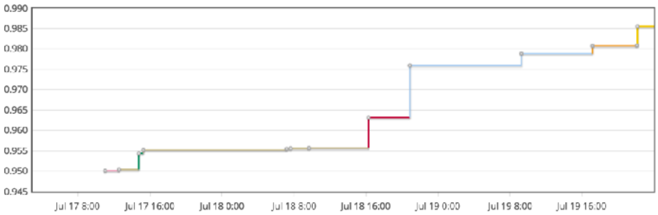
- detect electromagnetic shower basetracks within a brick basetracks (in test sample we have only description of the track (x, y, z, TX, TZ) for every track and set of mother-particles)
- Figure of Merit: ROC AUC

Hosted at: Kaggle, https://inclass.kaggle.com/c/dark-matter-signal-search-episode-1, requires valid account!

Competition deadline: 19-July-2017 23:59 UTC+0

Prize: memorable prizes + talk on Thursday

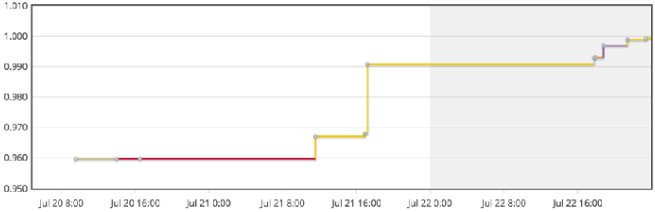
# Dark Matter Search-1




69 players
13 external
~600 submissions

| # | Δrank | Team Name | Score | Entries | Last Submission UTC (Best – Last Submission) |
|---|---|---|---|---|---|
| 1 | — | Benda Xu | 0.98556 | 3 | Wed, 19 Jul 2017 22:09:06 |
| 2 | — | Miha Zgubic | 0.98122 | 23 | Wed, 19 Jul 2017 22:29:42 |
| 3 | — | Konstantin | 0.98099 | 4 | Wed, 19 Jul 2017 23:33:12 |
| 4 | ↑1 | Georgy Chebanov | 0.97063 | 5 | Wed, 19 Jul 2017 02:52:52 |
| 5 | ↑1 | Dzianis Sivets | 0.96933 | 7 | Wed, 19 Jul 2017 16:54:05 (-3.9h) |

# Dark Matter Search-2

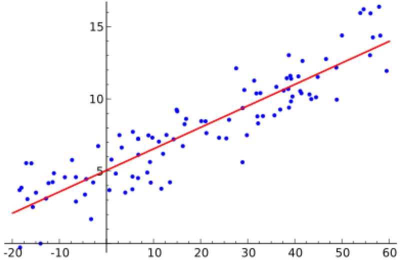


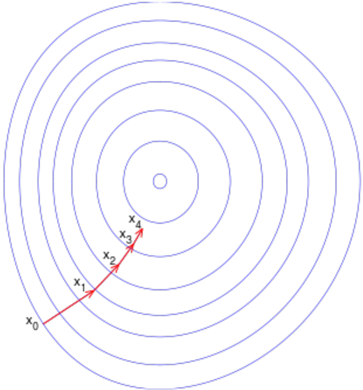

32 teams
34 players
9 external
193 submissions

| # | Δrank | Team Name | Score | Entries | Last Submission UTC (Best − Last Submission) |
|---|---|---|---|---|---|
| 1 | — | denaas | 0.99927 | 9 | Sat, 22 Jul 2017 23:26:47 |
| 2 | — | Kangaroo | 0.99720 | 6 | Sat, 22 Jul 2017 23:51:31 (-0.7h) |
| 3 | — | Konstantin | 0.99695 | 3 | Sat, 22 Jul 2017 22:52:23 |
| 4 | — | MZ<br>• Miha Zgubic<br>• Alex Mason | 0.99356 | 8 | Sat, 22 Jul 2017 23:03:56 (-0.1h) |
| 5 | — | JonahPhilion | 0.98836 | 14 | Sat, 22 Jul 2017 21:05:33 (-1.4h) |
| 6 | — | FraP | 0.98282 | 7 | Sat, 22 Jul 2017 23:38:37 |
| 7 | — | Taka | 0.97923 | 15 | Sat, 22 Jul 2017 21:06:33 (-1.5h) |
| 8 | — | byzhang | 0.97847 | 2 | Sat, 22 Jul 2017 23:07:23 |

# Basics: linear regression and gradient descent

- Building blocks for neural networks
- Surprisingly powerful
- Extensive literature about different gradient descent methods (stochastic, [momentum](momentum))



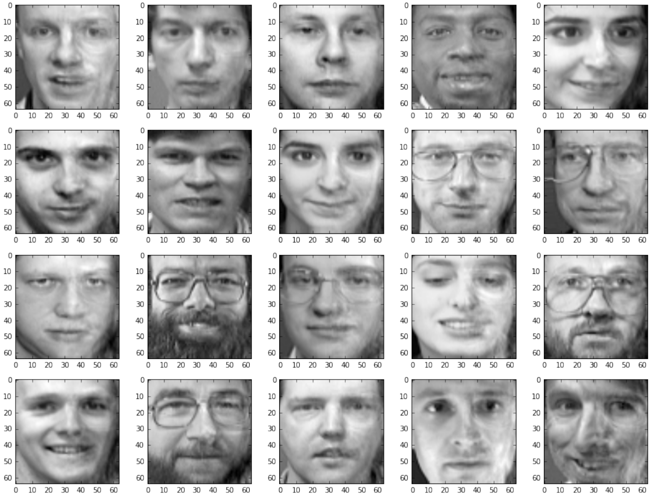



# Linear regression and gradient descent for faces: demo

```
git clone https://github.com/yandexdataschool/mlhep2017.git
cd mlhep2017
jupyter-notebook day1/seminar2/1.2.1\ Linear\ models\ \(Faces\).ipynb
```

# Linear regression and gradient descent for faces: demo

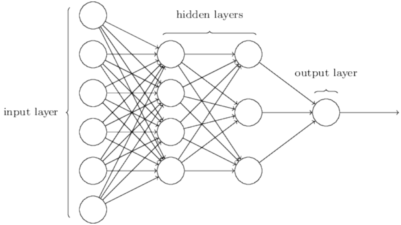
- Using the left half of a face to predict the right half



# Boosted decision trees

- Covered in depth during the school, but already pervasive in HEP
- Will not discuss here

# Neural networks: one-slide introduction

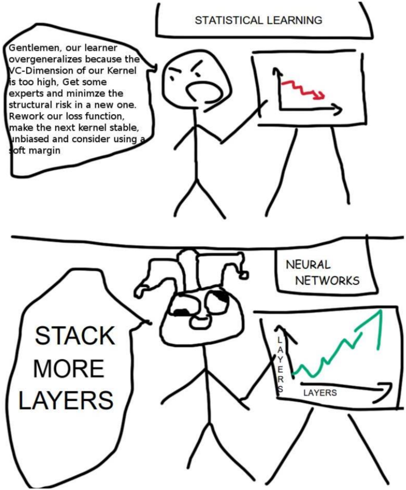



- Neuron is a real-valued function: $f(x_i) = \sigma(w_i x_i + b)$
  - $x$: input, $w$: weight, $\sigma$: activation function (tanh, sigmoid, ReLU)
- Neural network is a network of functions with one or more output discriminants
- Initialise weights randomly
- Train = changing $w$ and $b$ to minimise e.g. mean squared error wrt output discriminant

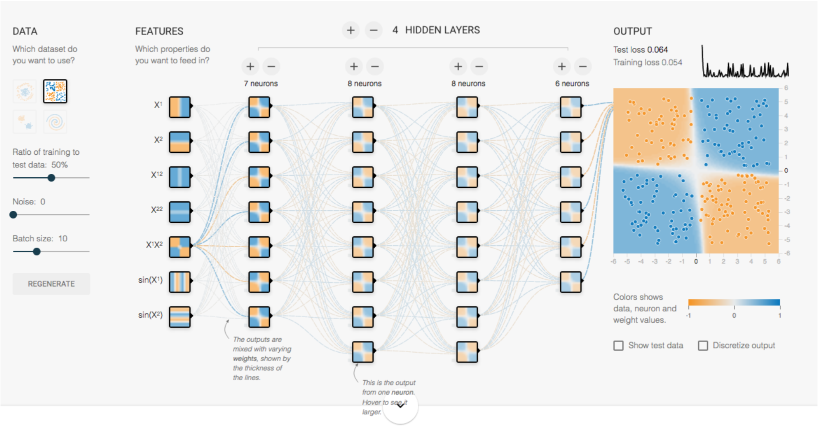
# Neural networks: slightly more detail

Key choices

- Activation function
- Cost function
- Algorithm to minimise weights and biases wrt cost function
- Number of layers



# Neural networks in the browser?



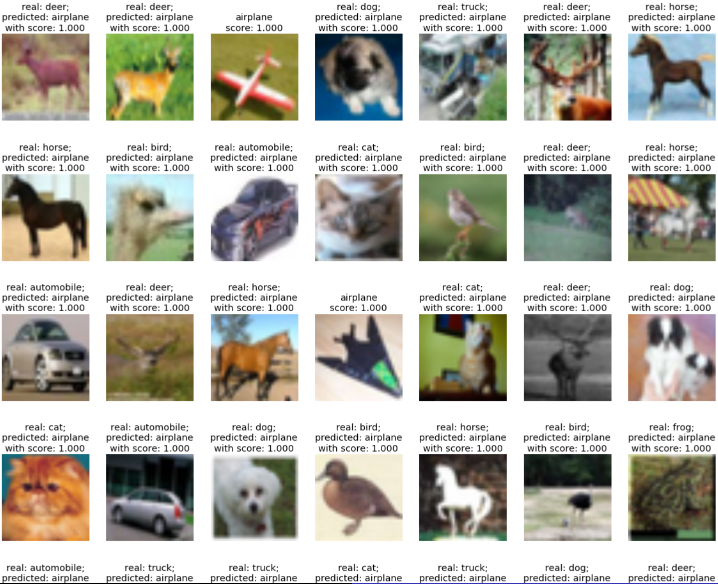

[Tensorflow playground]

# Pattern recognition using pre-trained neural network

```
git clone https://github.com/yandexdataschool/mlhep2017.git
cd mlhep2017
jupyter-notebook day5/seminar0/Using_pre_trained_net.ipynb
```

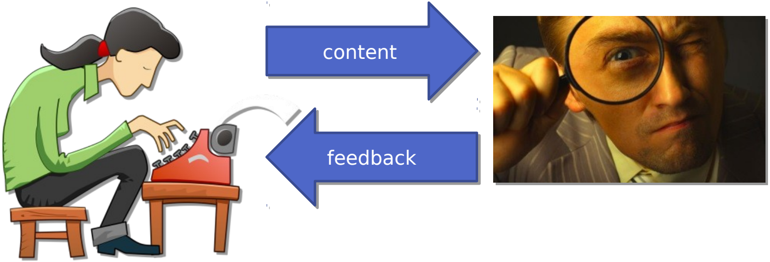
# Convolutional neural networks

- State of the art for image recognition, if you train it well



# Generative Adversatial Networks

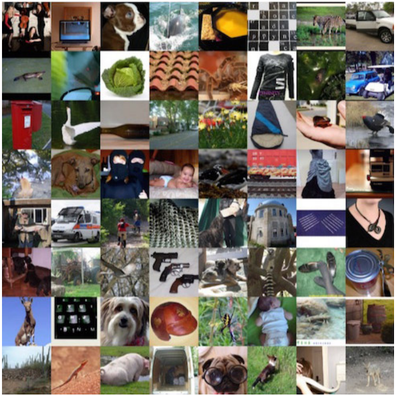
**Generator** **Discriminator**




Generate image
(should be plausible)

Tell if image is plausible
(image) → P(fake)

# Generative adversarial networks in action



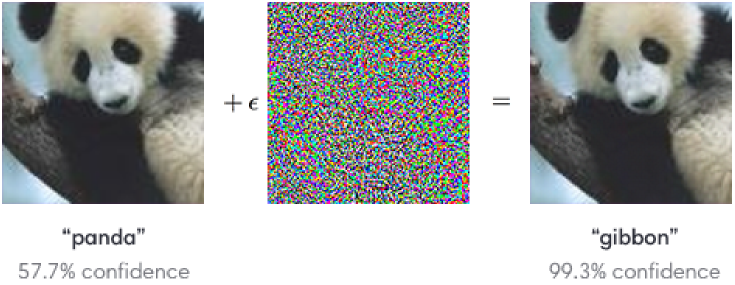
https://blog.openai.com/generative-models/

# Generative adversarial networks in action

- Direct applications to event generation and detector simulation discussed in e.g. ACAT talk

# Attacking with adversarial examples

- Not too relevant for HEP, but interesting for e.g. self-driving cars
- Attacker can craft an image that looks identical to a human but fools an NN



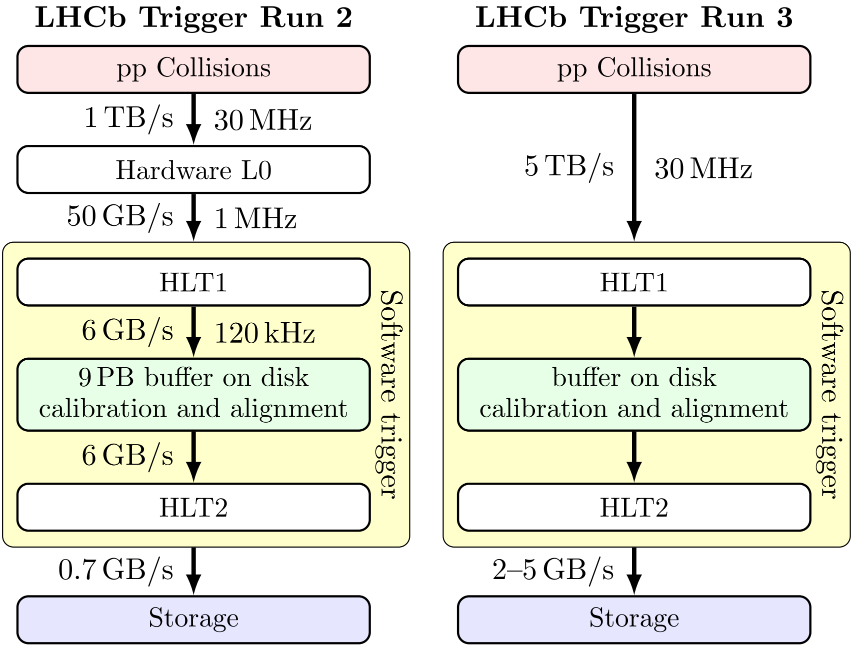

https://blog.openai.com/adversarial-example-research/, arxiv:1412.6572

# Machine Learning in the LHCb Trigger and Beyond

Mike Williams

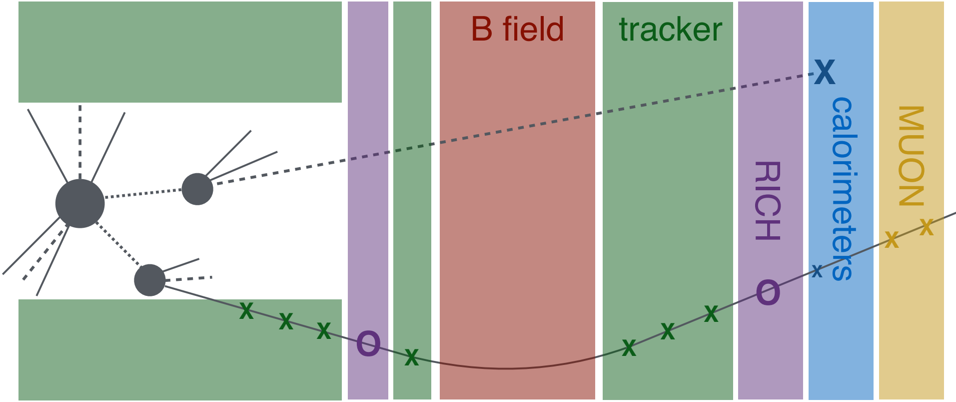
Department of Physics & Laboratory for Nuclear Science
Massachusetts Institute of Technology

July 19, 2017


LHCb Trigger Run 2
pp Collisions
1 TB/s
30 MHz
Hardware L0
50 GB/s
1 MHz
HLT1
6 GB/s
120 kHz
9 PB buffer on disk
calibration and alignment
6 GB/s
HLT2
Software trigger
0.7 GB/s
Storage
LHCb Trigger Run 3
pp Collisions
5 TB/s
30 MHz
HLT1
buffer on disk
calibration and alignment
HLT2
Software trigger
2–5 GB/s
Storage

# HLT1

HLT1 has 25k physical cores (>50k logical cores) and access to all raw data, but cannot afford to do full event reconstruction. Choose to do charged-particle tracking with a threshold of $p_T > 0.5$ GeV (included PV making).

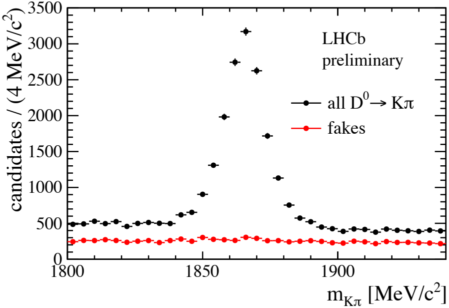


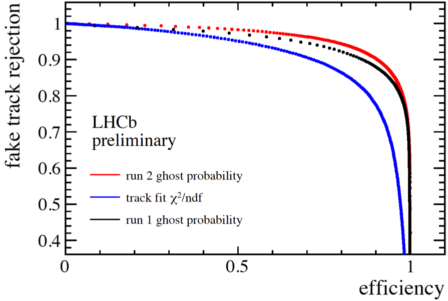

LHCb builds VELO segments first, then extends these to the next station, then beyond the B field to the final station before Kalman filtering all tracks.

# Fake-Track Killer

Fake-track-killing neural network, most important features are hit multiplicities and track-segment chi2 values from tracking subsystems.

LHCb-PUB-2017-011

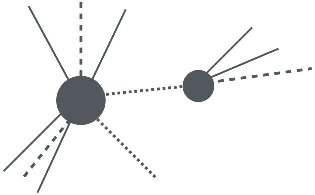



Run in the trigger on all tracks, so must be super fast. Use of custom activation function and highly-optimized C++ implementation (ROOT's TMVA package provides stand-alone C++ code to run the trained algorithm).

# HLT1 ML Selections

About 70% of the output bandwidth from HLT1 is taken up by inclusive selections that seek to efficiently select almost any heavy flavor decay that could be of interest.

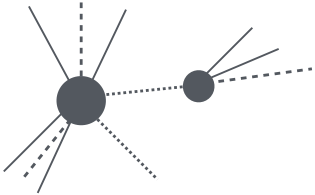


- A one-track algorithm based on the $p_T$ and IP $x^2$ (track-quality criteria applied as pre-selection; there is also a version of this that only considers muons).
- A two-track (SV) algorithm based on vertex $x^2$, flight distance $x^2$, scalar track pT sum, and n(small IP $x^2$) tracks (also has a heavy-flavor-like preselection).

The majority of the LHCb physics program uses data selected in HLT1 by these algorithms, which use MatrixNet (trained by our Yandex friends).

# HLT2 Topological Trigger

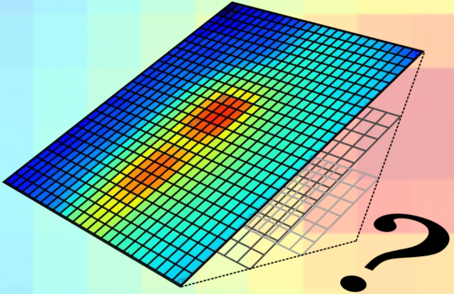
About 40% of the final output bandwidth is given to inclusive selections that seek to efficiently select almost any b-hadron decay that could be of interest.



- An SV algorithm that considers 2, 3, and 4-track vertices (seeded by HLT1 ML selections).
- The ML uses corrected mass, vertex $\chi^2$, scalar track $p_T$ sum, flight distance $\chi^2$, pseudorapidty (PV-SV), min(track $p_T$), n(small IP tracks), IP $\chi^2$, n(very b-like tracks).
- All features are discretized in the ML for stability, robustness, etc.

V.Gligorov, MW, JINST 8 (2012) P02013.

This algorithm has run since the start of 2011, and has collected the data used by ~200 papers! It was re-tuned for Run 2 by Yandex (now based on MatrixNet, was a BDT in Run 1). T.Likhomanenko et al [1510.00572]



# Deep Learning Jet Images

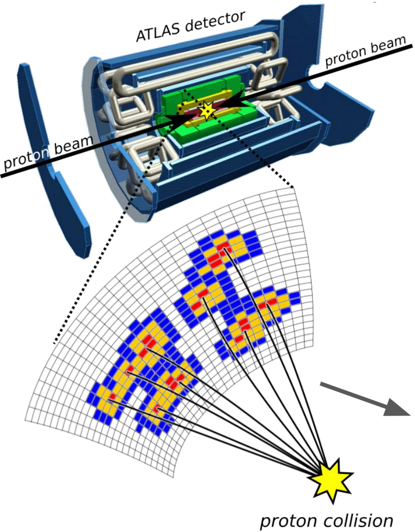

Noel Dawe


MLHEP 2017
Reading, UK

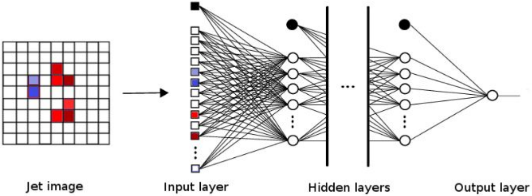

THE UNIVERSITY OF MELBOURNE

CoEPP
ARC Centre of Excellence for Particle Physics at the Terascale

# Machine Learning Jet Substructure

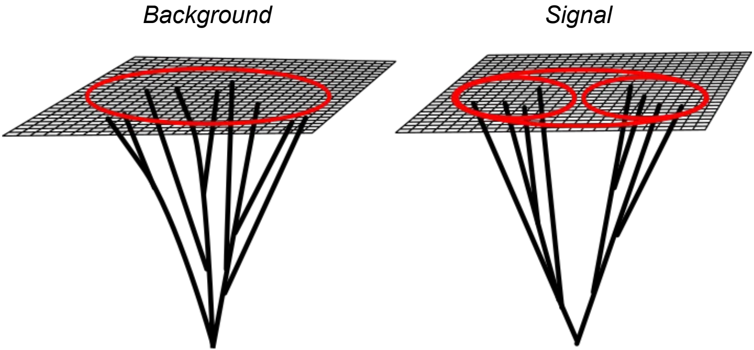



Apply **deep neural networks** common in computer vision applications to distinguish different sources of jets using ***"jet images"***




*Flattening the calorimeter into a 2D image...*

# Challenge: Boosted hadronic W decays vs QCD jets

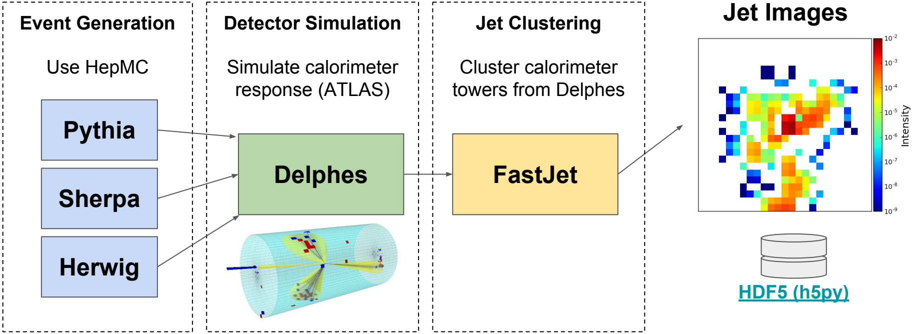



**QCD jet**
**quark/gluon**

**W jet**

**Two subjets with separation scaling as**
***$2\,m_W / p_T$***

# Creating Jet Image Data




Each stage is a Python generator function that yields a numpy array

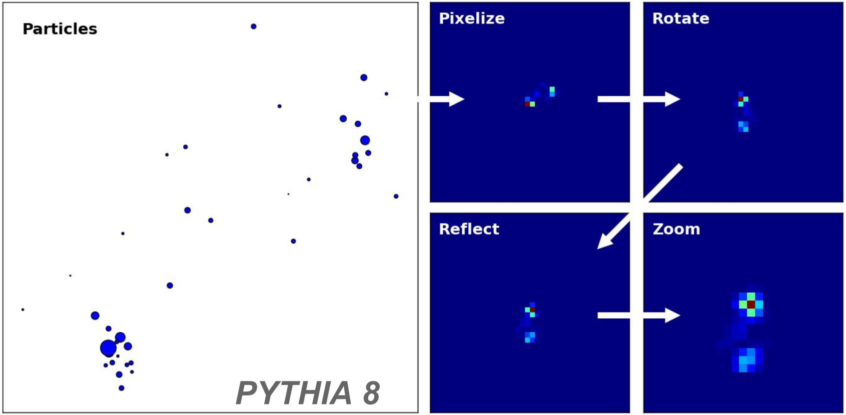
Jet images can be produced and used “on-the-fly” or saved to disk for later use

Heavy use of [Cython](Cython) for interfacing NumPy and the above software

See the code: https://github.com/deepjets/deepjets

# Constructing Jet Images

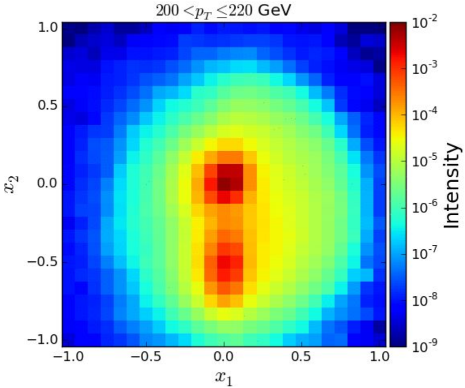
- Sum transverse energy of calorimeter towers in grid of 0.1 x 0.1 in η-φ space
- Perform translations, rotations and reflections in η-φ space
- Zoom the image to minimise $p_T$ dependence
- Crop at 25 x 25 pixels and normalise



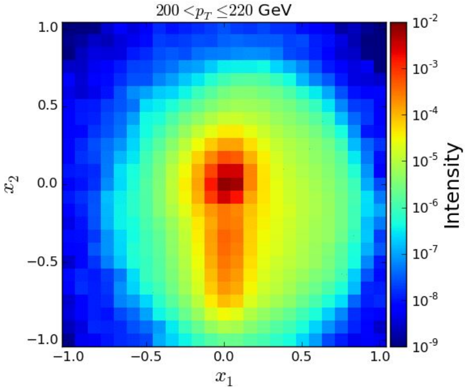
```
da.tensordot(images, w, axes=(0, 0)).compute() / w.sum())
```

(Images are weighted such that the $p_T$ distribution is flat)

## W Jets

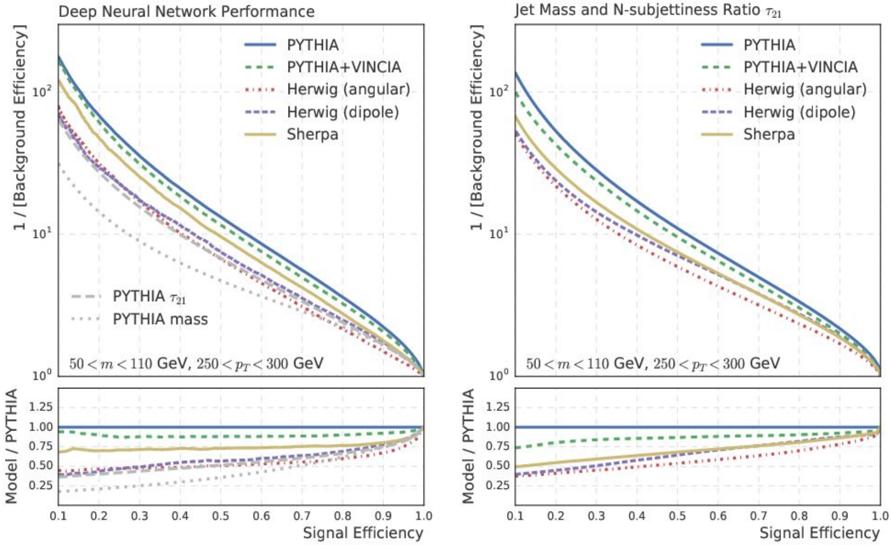



## QCD Jets




***Images zoomed by: $p_T / 2\, m_W$***

# Comparing Generators: Network Performance



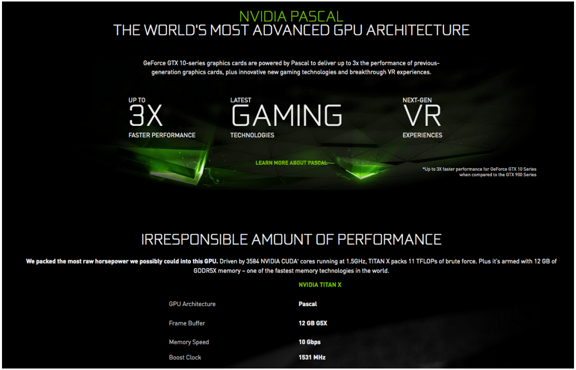

*DNN slightly outperforms traditional techniques and appears to have* ***uncertainties similar in size!***

# **Do** try this at home

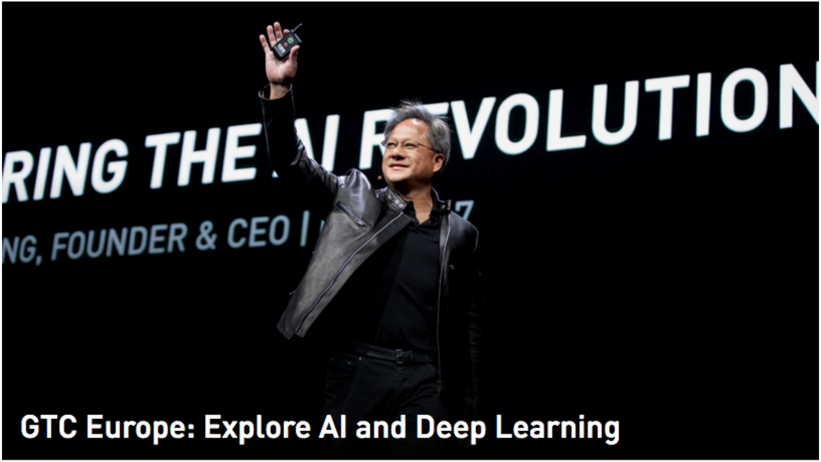
```
ssh -X username@hepacc02.pp.rl.ac.uk
singularity shell --nv /usr/local/scontainers/3.2.1/tf\_gpu-1.2.0-cp35-cuda8-cudnn51.img
python3.5

>>> import tensorflow as tf
```

Run e.g. Tensorflow MNIST tutorial




NVidia Titan X Pascal




**GPU-accelerated deep learning and artificial intelligence showcased at GTC Europe, Oct. 10-12 in Munich.**

I am a visionary. I am a healer. I am a protector. I am a helper. I am a navigator. I am a creator.

I am AI.

I am a visionary. I am a healer. I am a protector. I am a helper. I am a navigator. I am a creator.

I am AI.

# Conclusions

- Excellent open source toolkits and learning resources available from industry/academia
- Historically, best applications in areas where our problem domains map well onto image processing
- Promising new applications of e.g. deep neural networks, generative adversarial networks
- Need to provide convenient standard data sets if we want machine learning experts from outside HEP to work with us
- State of the art in machine learning is a moving target!