Diving into Deep Learning with keras using your ntuples

Ryan Reece (UCSC)

ryan.reece@cern.ch

tutorial envisioned and designed by

Amir Farbin (UTA)

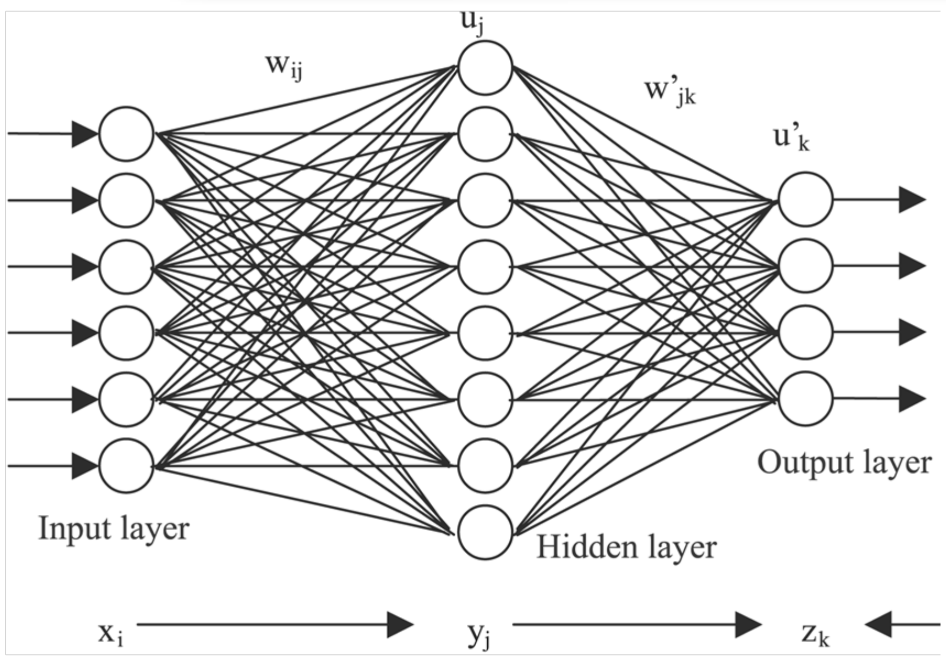
I'm not an expert

- I'm just starting to learn this stuff too. I'm an enabler.
- I guess I've drank the coolaid, or I am interested enough to evangelize some because I think we, physics experimentalists, should think more about what is happening in ML *right now*.
- I am also *skeptical* about how quickly physicsts will adapt to new techniques, as we are careful and good at reconstruction/analysis. But the gains could be important.
- In addition to Amir Farbin, I've learned a lot from David Rousseau and Michael Kagan, who run the new ML group in ATLAS. We had a workshop last March that brought a lot of this to my attention: https://indico.cern.ch/event/483999/

Computing setup

- We are following the tutorial here: `https://twiki.cern.ch/twiki/bin/view/AtlasComputing/SoftwareTutorialDeepLearning`
- Got to the Setup on lxplus section and add to the PATH and source activate to setup my (Ryan's) installation on afs.
- You can follow the Installation instructions to install the full environment on your own machines on your own time.
- Try running the test out of the box:
  `python -m EventClassificationDNN.Experiment --cpu`
- Assuming that is ok for you, let's pause the walkthrough on the twiki to finish the introduction to DL in these slides. Then back to the TWiki.

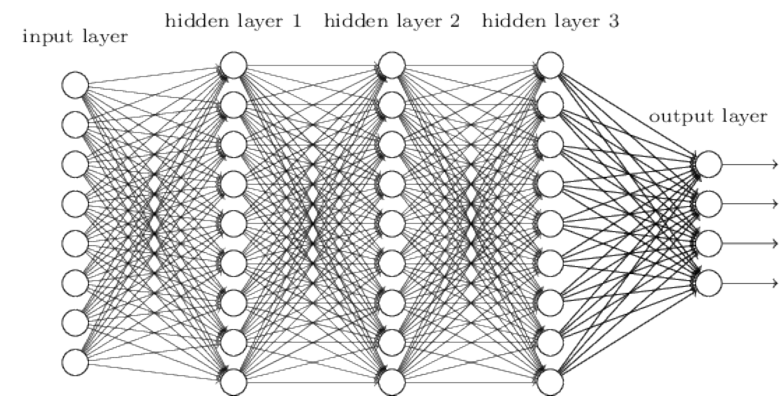
Neural Nets



Neural nets have:

- input varaiables, x_i
- weights, w_{ij}
- activation function (sigmoid, tanh, ...), u_j
- output variables, y_j
- a *learning rule* to update the weights.
- a learning step is called an "*epoch*."
- Optimizing the weights is called "*training*."

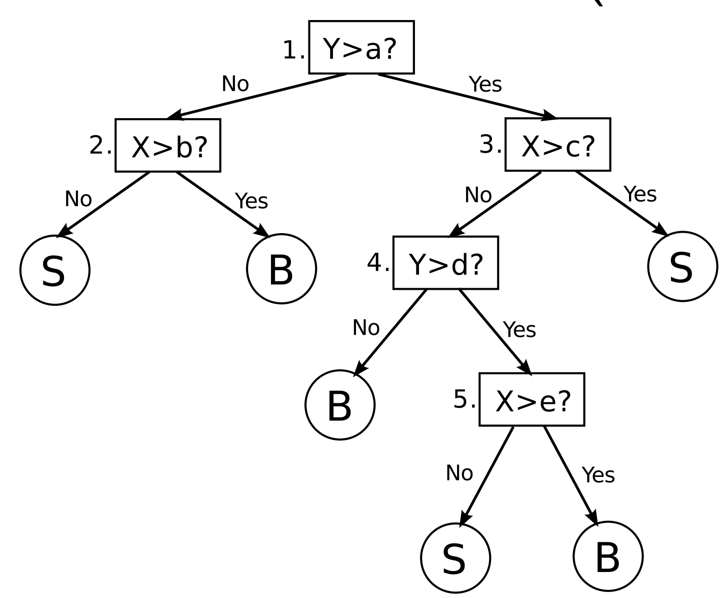
"*Deep*" networks have multiple hidden layers

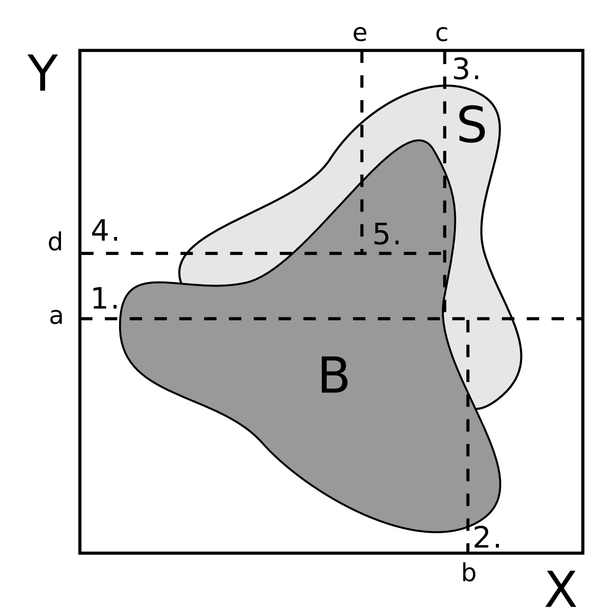


Can be used for *classification* or *regression*.

Similar to other multivariate techniques, cutting on a classifier makes some acceptance blob in parameter space.

Boosted Decision Trees (BDT)

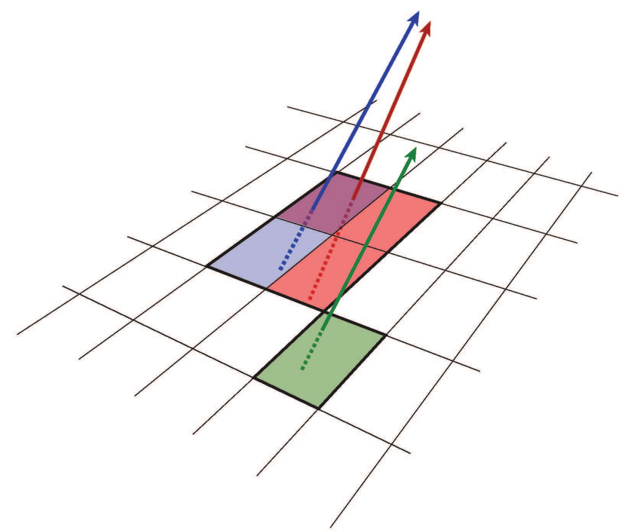


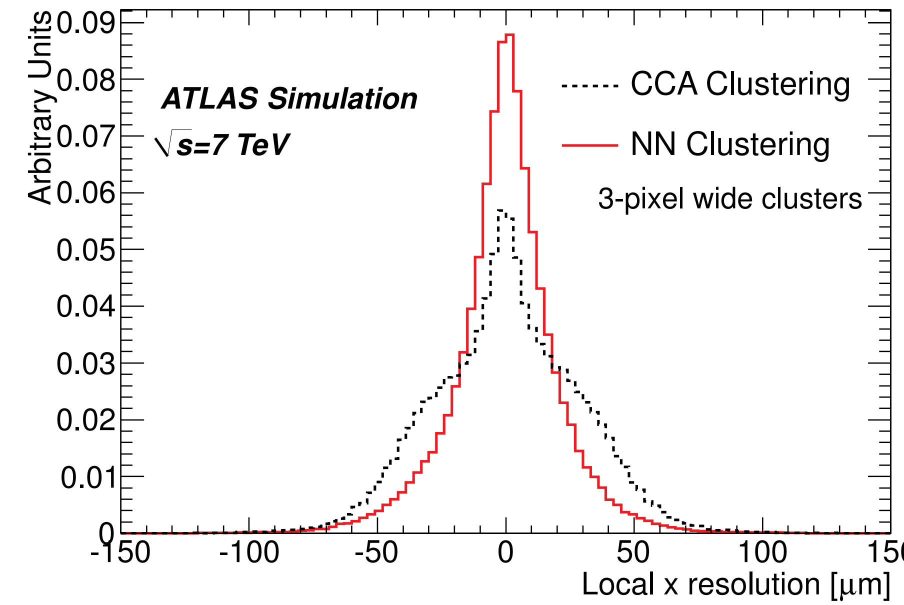


NNs and BDTs in ATLAS

- Using NNs and other MVAs has been common in HEP for years, for pattern recognition, particle ID, event selection...
- In the past, always used *shallow* NNs.
- ATLAS uses NNs in many places, e.g. pixel clustering.
- Jet tagging for taus and b-quarks has used NNs in many iterations (also c, q/g).

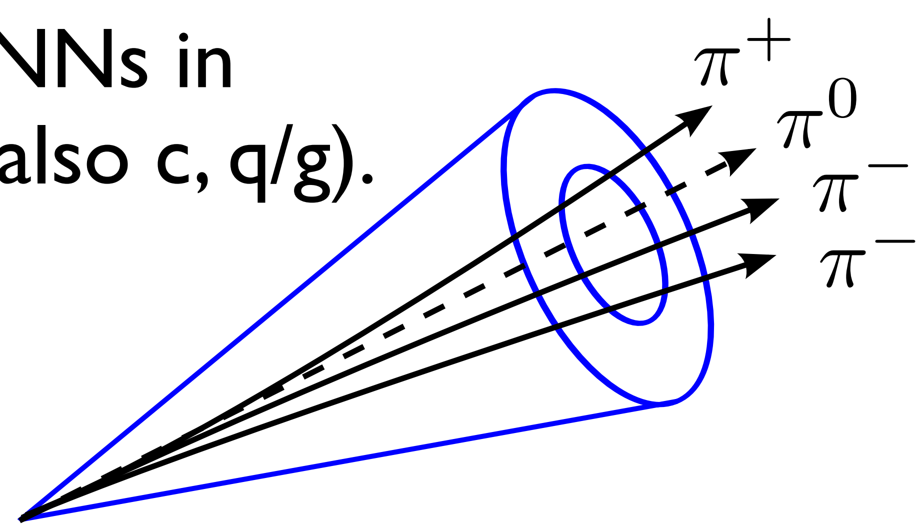
ATLAS pixel clustering with NNs

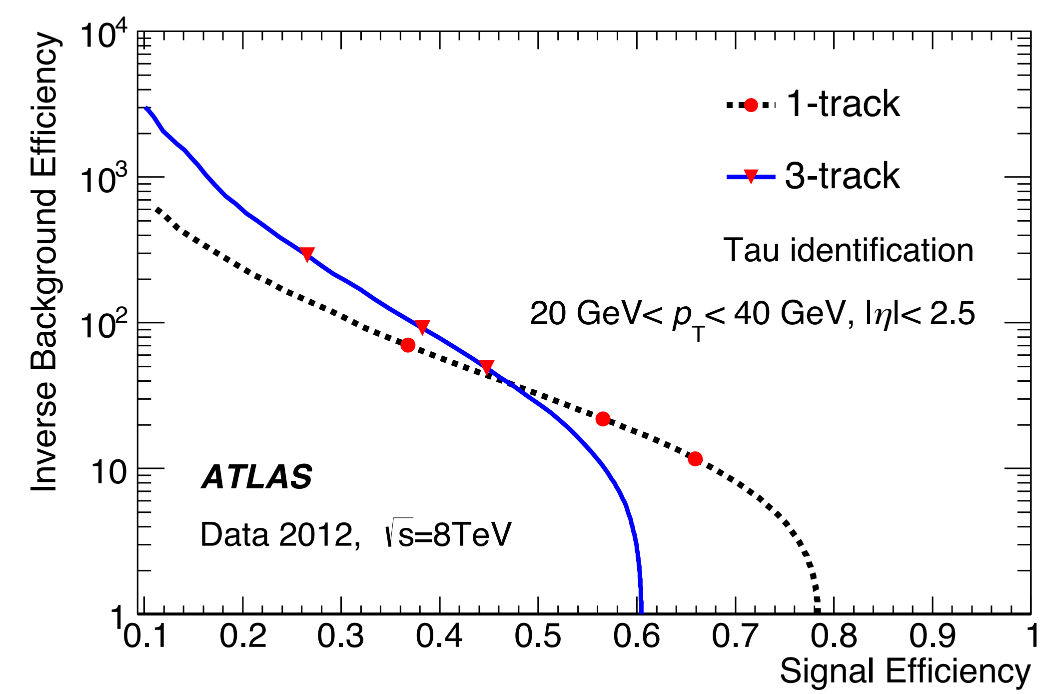




[1406.7690]

ATLAS tau identification with BDTs

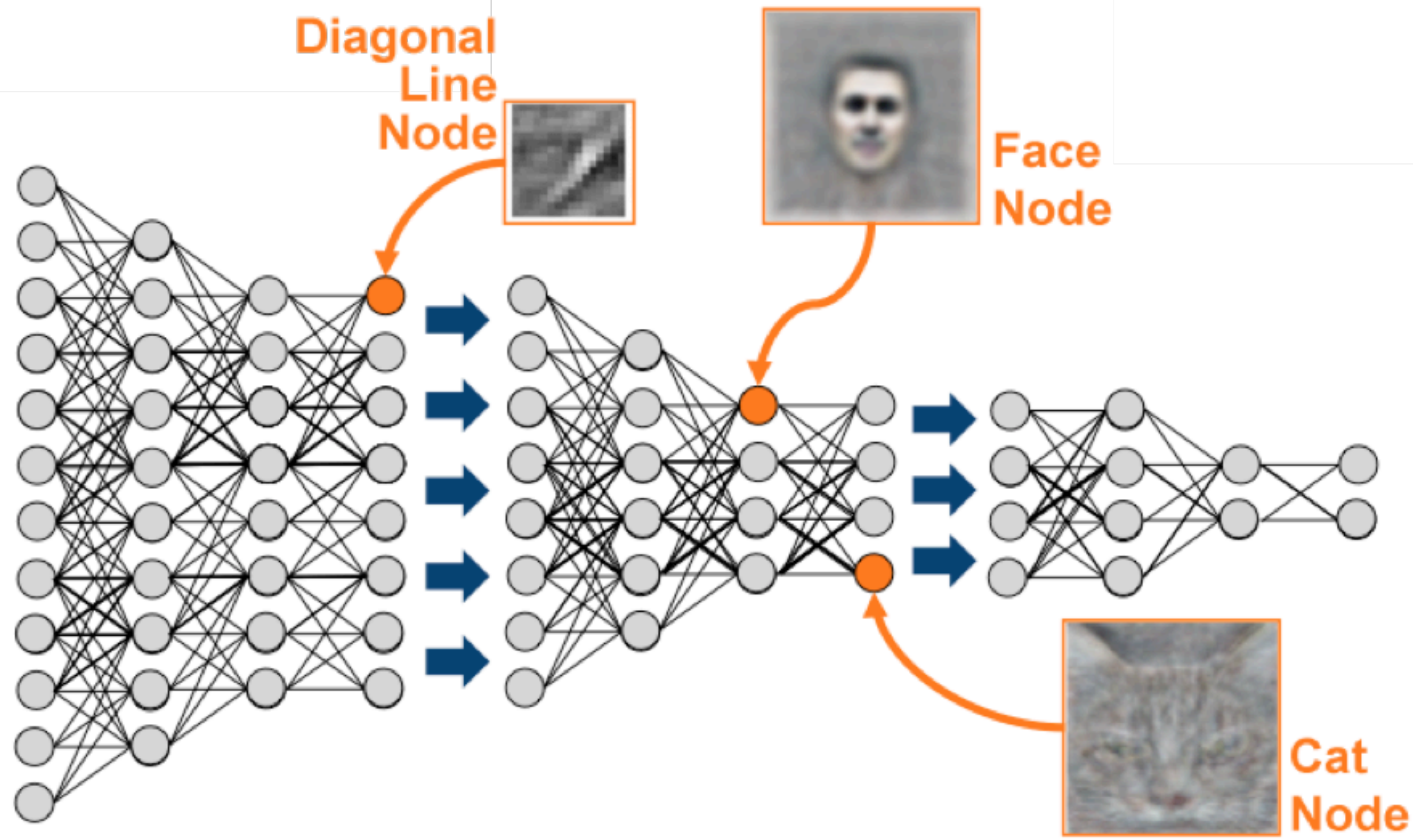




[1412.7086]

Why go *deep*?

- "Vanishing gradient problem" → hard to train many layers.
- Multiple layers allow for *feature extraction*.
- Allow us to better explore and understand our data.
- ***Now in "Deep Learning Renaissance"***



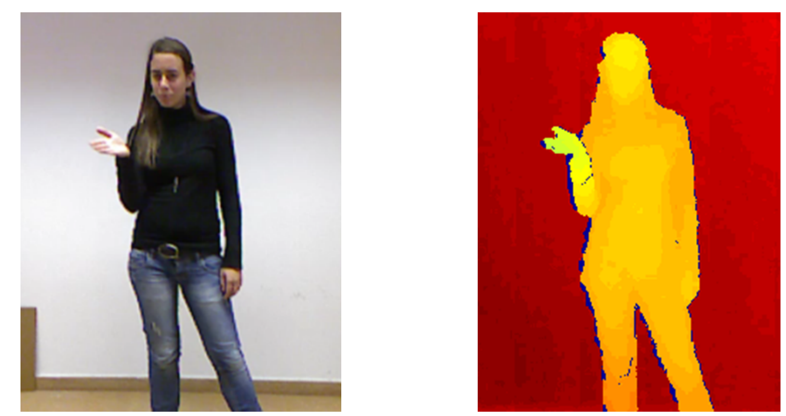
[Forbes/Google]

1. *Better training*: techniques and tools (e.g. smarter NN structures).
2. *Better hardware*: multicore, GPUs, bigger data centers, cloud computing, coming: neuromorphic computing.
3. *More training*: bigger datasets, search, the internet, open science.

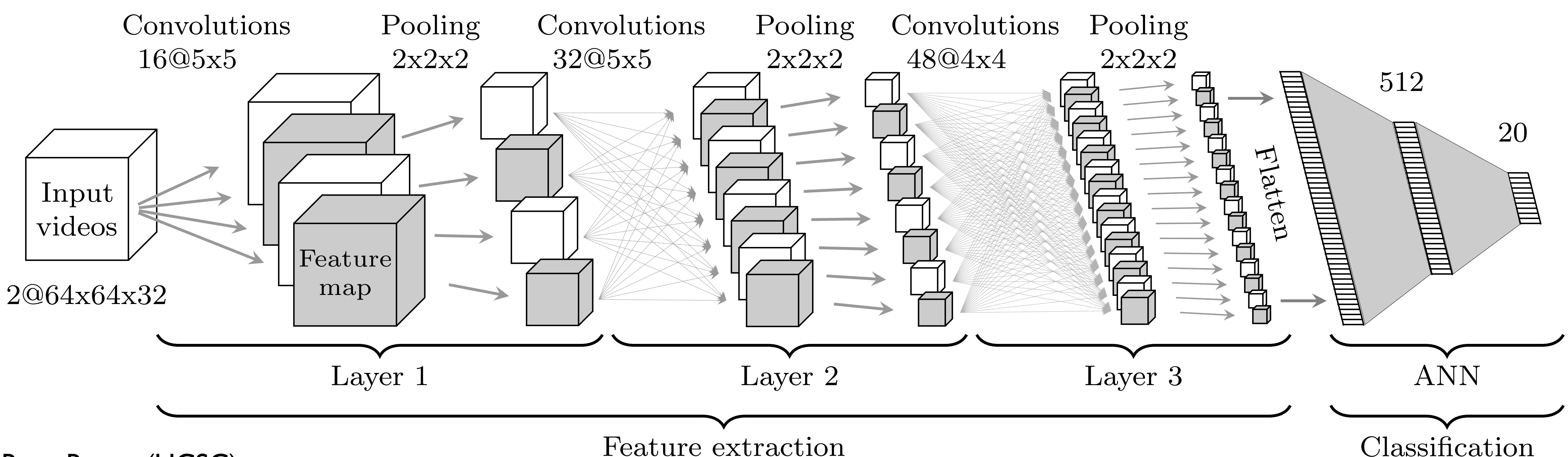
Examples of CNNs

- In 1990s, Yann LeCun pioneered Convolutional Neural Nets (CNN) and used them for Optical Character Recognition.
- Inspired by animal cortex.
- Now it is standard in image recognition and captioning, NLP, computer vision, etc.

Pigou et al. (2014). Sign Language Recognition using Convolutional Neural Networks.



(a) RGB (b) Depth map



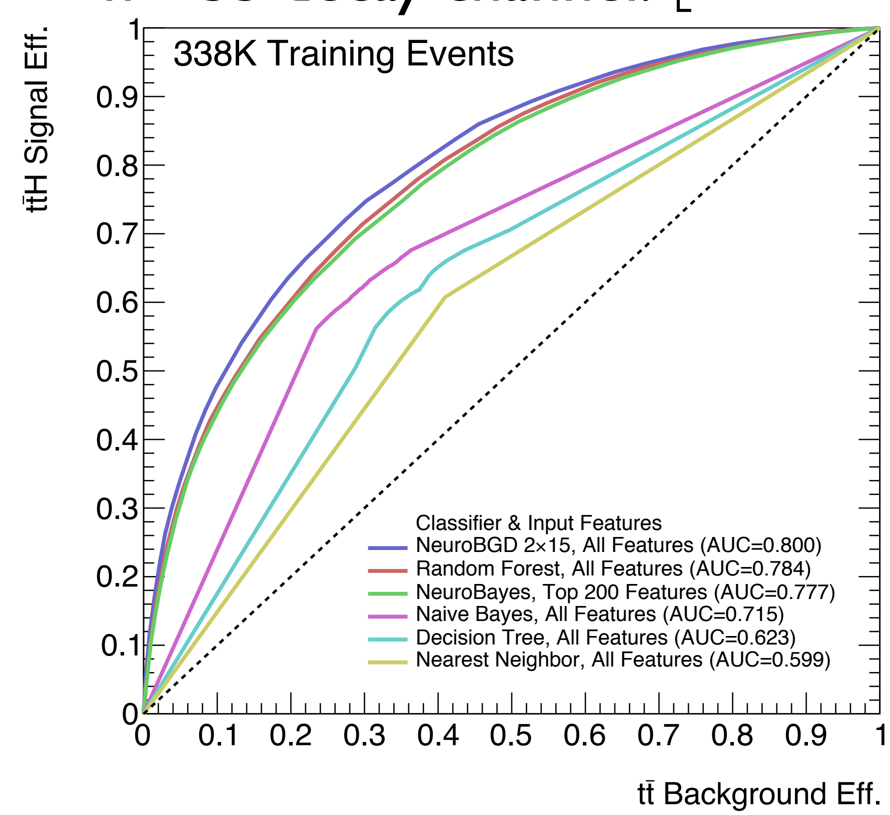
Deep Learning in HEP

- Deep learning does best with raw data and when there are unexploited features.
- raw channels→*tagging*
- basic kinematics→*features*

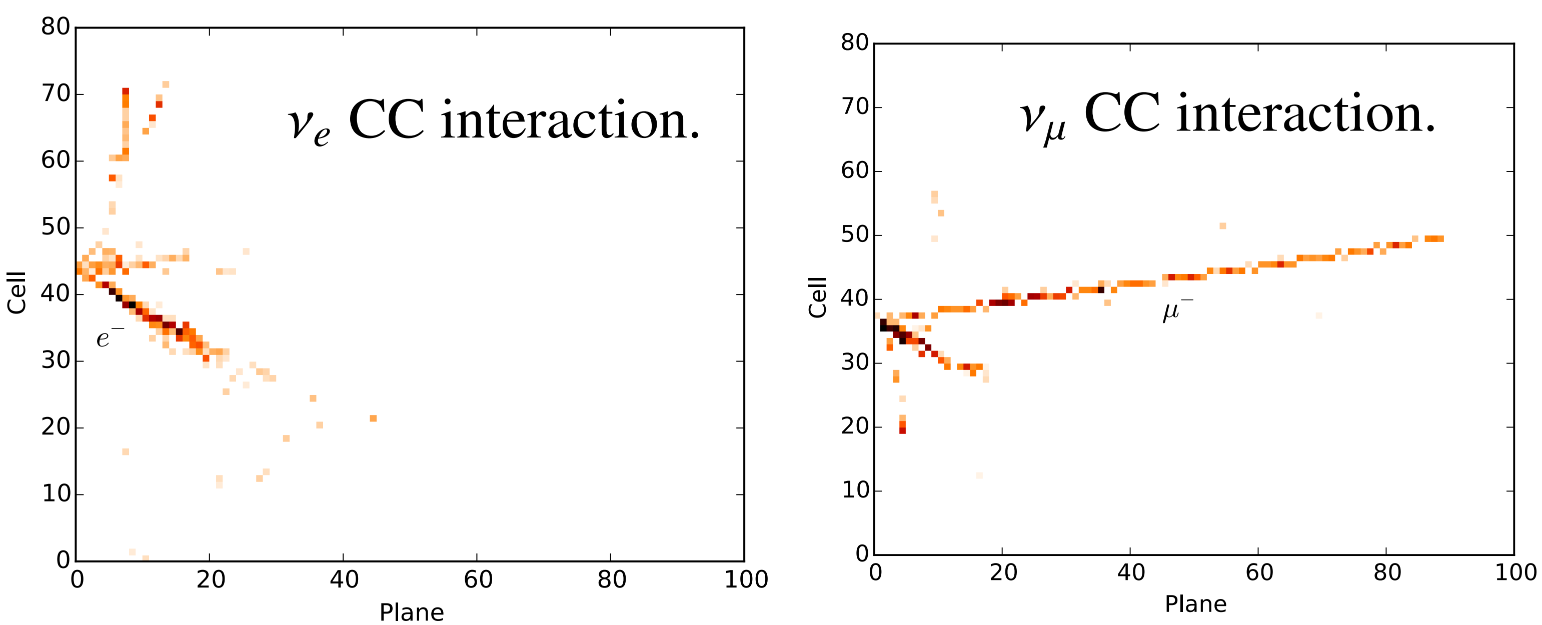
- Baldi *et al.* (2014). Searching for Exotic Particles in High-Energy Physics with Deep Learning. [1402.4735]
- 

Baldi *et al.* (2015). Enhanced Higgs to $\tau^+\tau^-$ Search with Deep Learning. [1410.3469]

- Santos *et al.* (2016). Machine learning techniques in searches for tth in the h→bb decay channel. [1610.03088]



- Aurisano *et al.* (2016). A Convolutional Neural Network Neutrino Event Classifier. [1604.01444]

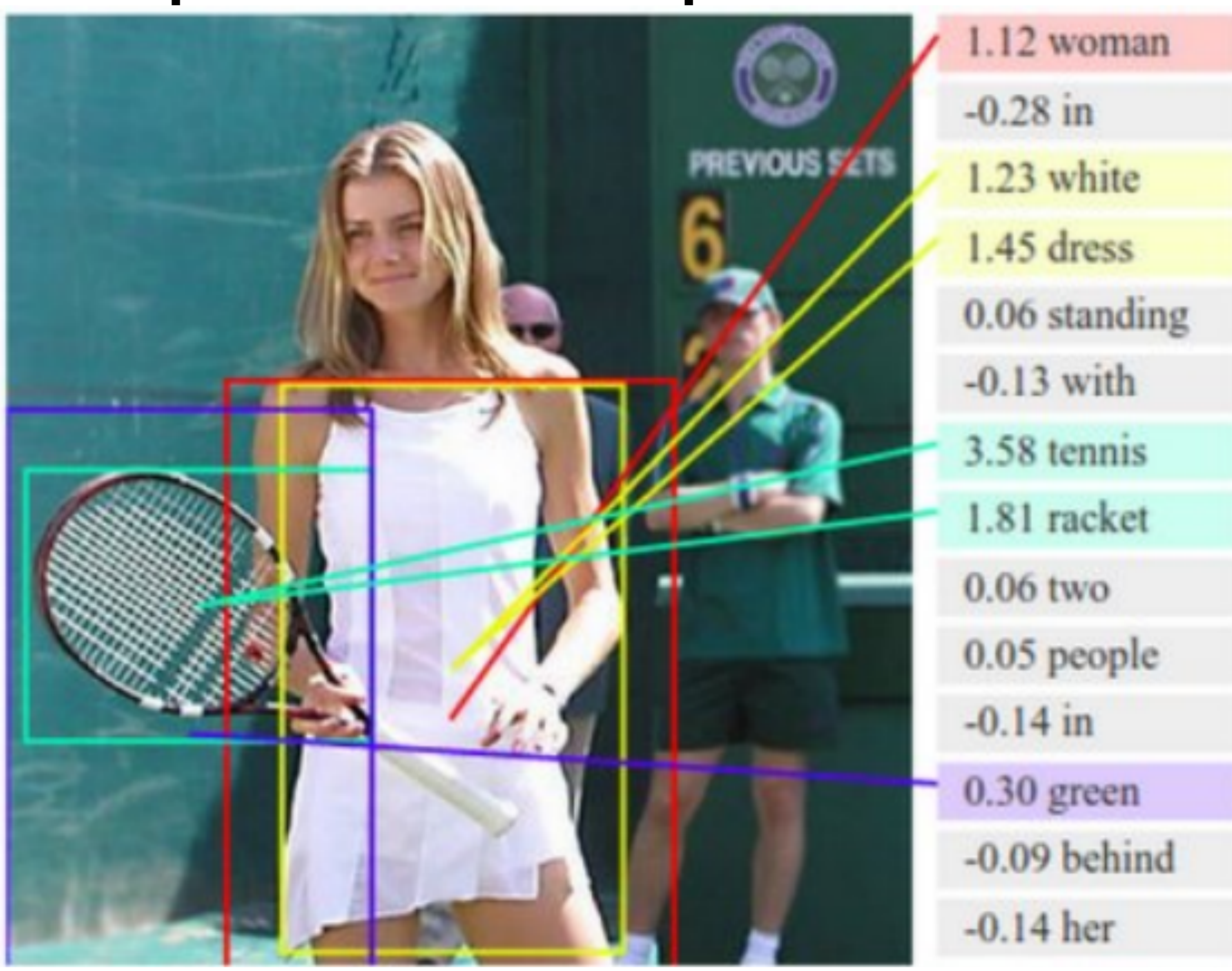


out performs NOvA's conventional reconstruction

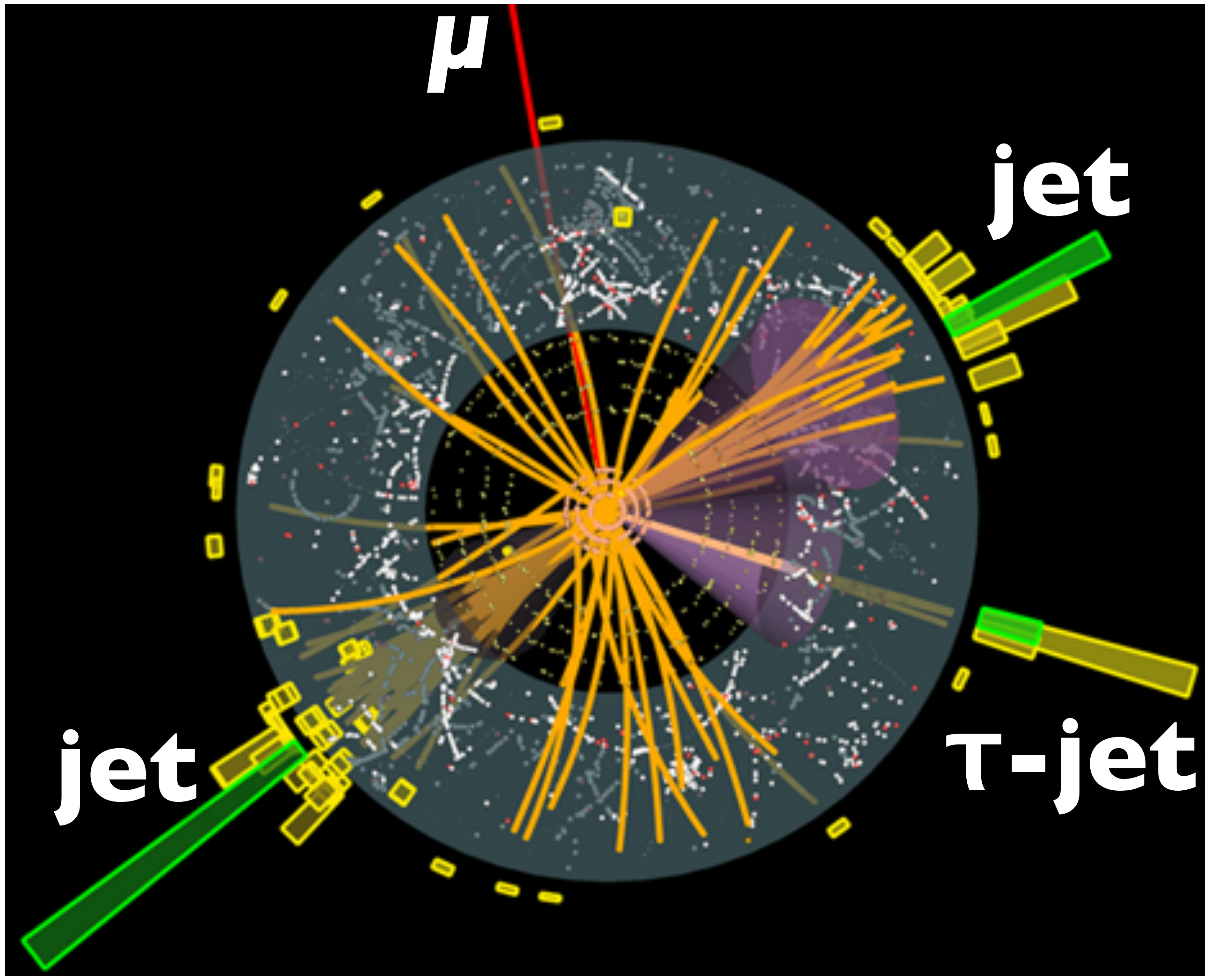
Deep learn from *raw* inputs

The vision as explained to me by Amir:

ImageNet
competition example



Future of ATLAS?



Solutions to big problems in ATLAS?

- Discover new features in the data and analysis techniques?
- Better particle and event classification?
- Faster, better pattern recognition and tracking for HL-LHC?
- Faster, better, data-driven simulations from generative models?

Data Science & Deep Learning Tools

- **scipy**
  - matplotlib - common plotting library
  - numpy - arrays and numerics in python
  - pandas - library for reading/writing/plotting structured data
- **scikit-learn** - various ML and classification packages for python
- **tensorflow/theano** - computer algebra systems designed for machine learning
- **keras** - python ML framework wrapping calls to tensorflow or theano backends.

Amir's DLKit

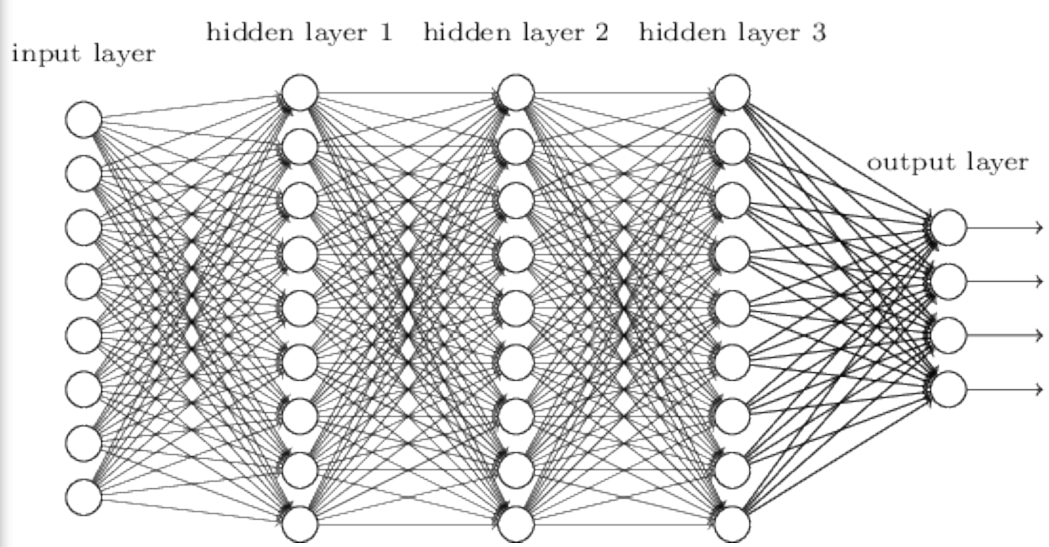
- DLKit is Amir's toolkit built around keras for handling datasets/models/results. As you learn keras, you'd probably build something like it.

- In the top-level file: DLKit/EventClassificationDNN/Experiment.py

```
# Build the Model
from EventClassificationDNN.Classification import FullyConnectedClassification
```

- The Build function actually constructs the NN using keras:

```
def Build(self):
  model = Sequential()
  model.add(Dense(self.width,
            input_dim=self.N_input,
            init=self.init))
  model.add(Activation('tanh'))
  for i in xrange(0,self.depth):
    model.add(BatchNormalization())
    model.add(Dense(self.width,init=self.init))
    model.add(Activation('tanh'))
    model.add(Dropout(0.5))
    model.add(Dense(1,input_dim=self.width))
  self.Model=model
```



Sequential()

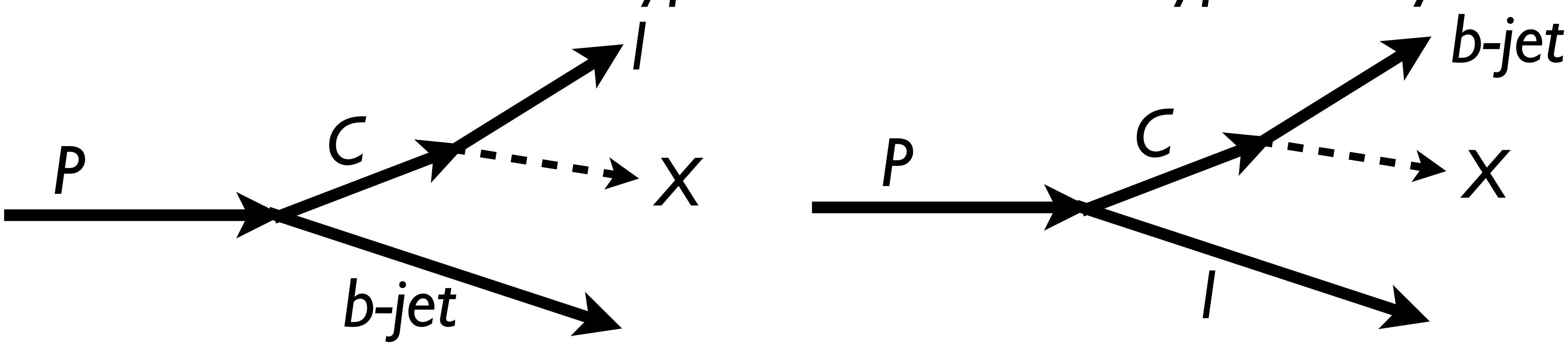
Amir's DLKit

- You need to convert TTrees to hdf5.
- Specify your input files in

  `EventClassificationDNN/InputFiles.py`

- The lines like:
  `[InputData, "AA_Gen"]`,
  are labeling InputData as being of true class "AA_Gen"

Goal: discriminate *A-type* from *B-type* decays.



This example developed by Chris Rogan and Amir Farbin.

Amir’s DLKit

- Also specify the input variables from your data in:

  `EventClassificationDNN/InputVars.py`

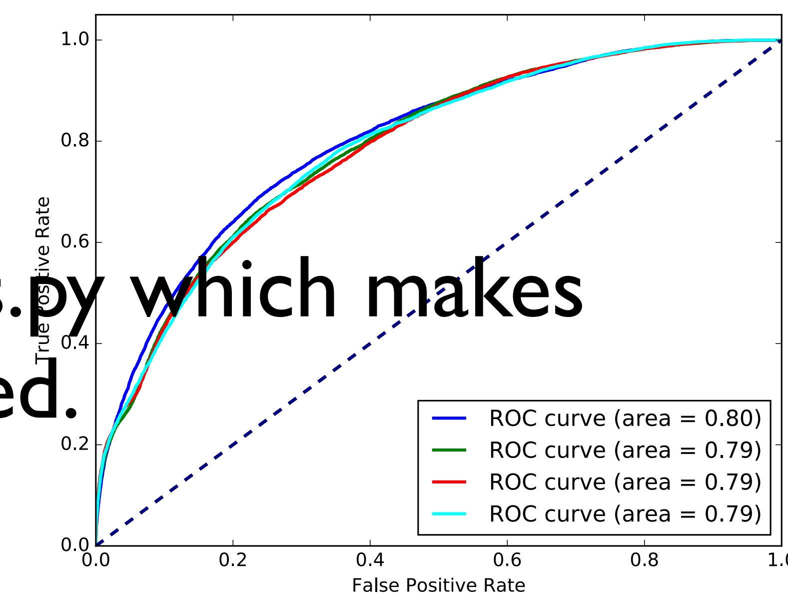
- The list **`FieldGroups`** groups together variables of common normalizations, like 0-1, -π--+π, energies, etc.

- **`SelectedFields`** selects which variables to use as input to the NNs. You can change these with:

  `-v --varset` e.g. `-v 0 (everything)`
  `-v 1 (“jigsaw”)`
  `-v 2 (four-vectors)`

- Also note the file `EventClassificationDNN/ScanConfig.py` which is meant to sample the depth/width structure of the NN for study and optimization.

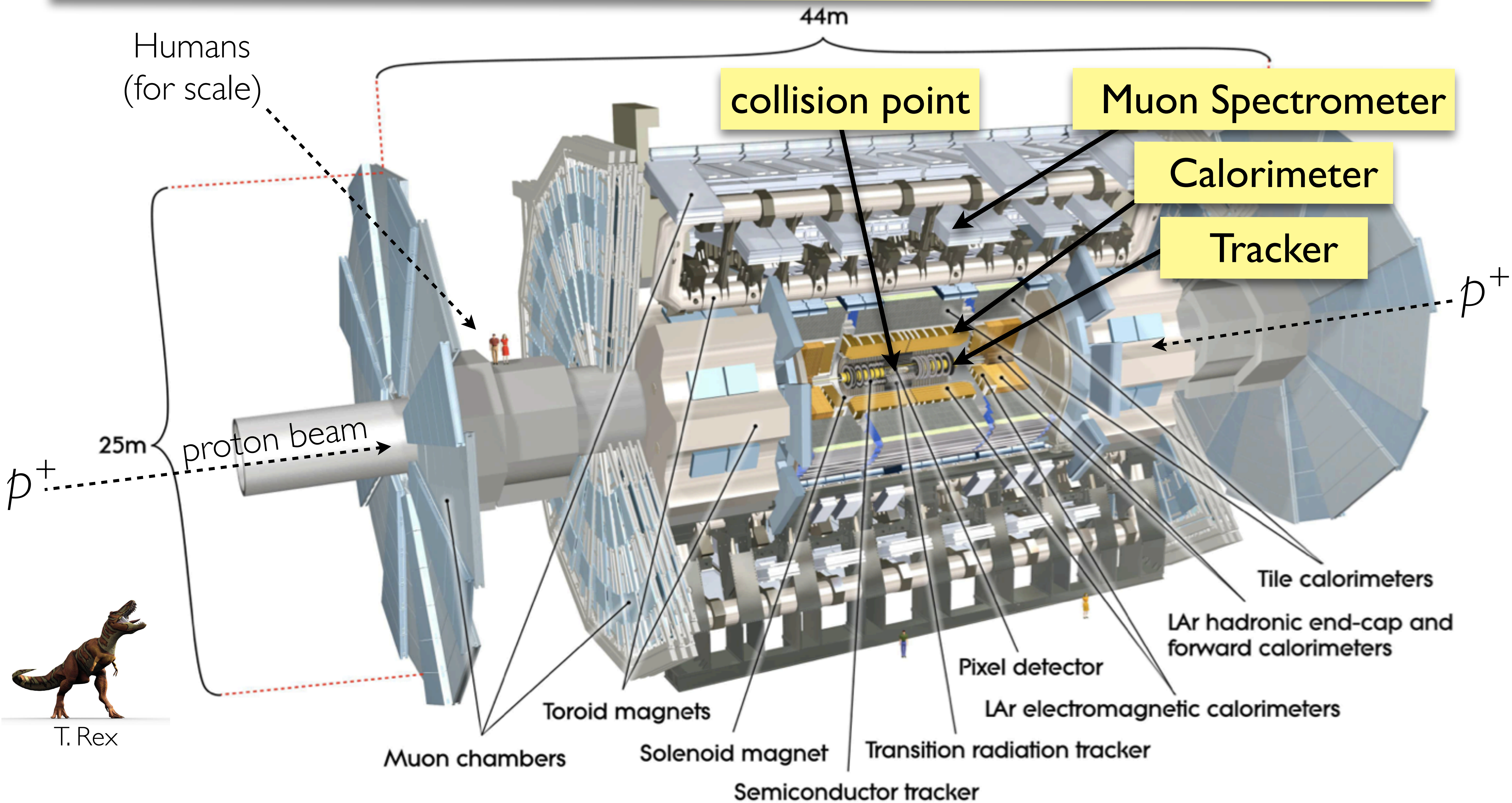
- Running the Experiment should also run Analysis.py which makes some ROC plots and could be further customized.



Back-up slides

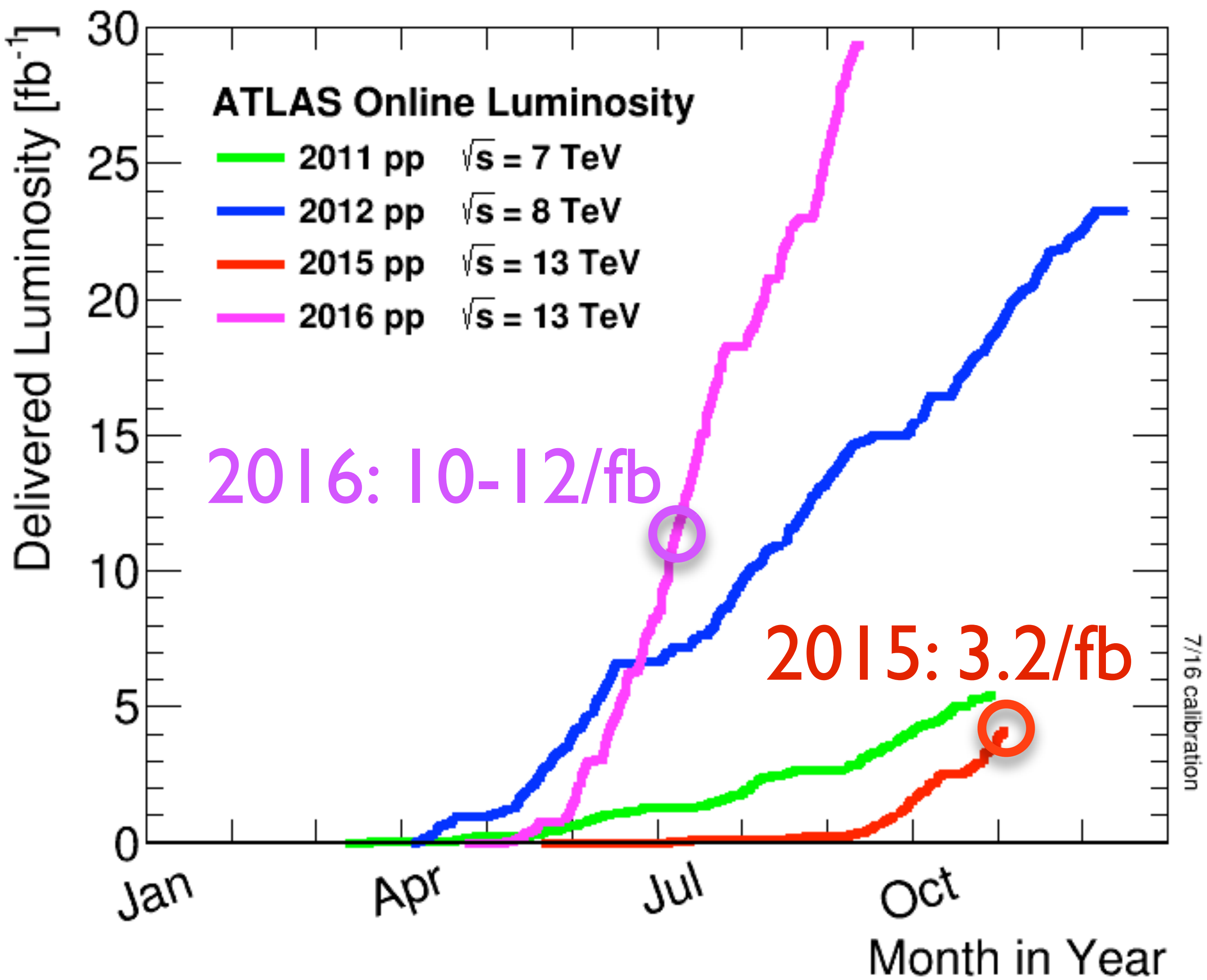
ATLAS Detector

ATLAS is a 7 story tall, 100 megapixel "camera", taking 3-D pictures of proton-proton collisions 40 million times per second, saving 10 million GB of data per year, using a world-wide computing grid with over 100,000 CPUs. The collaboration involves more than 3000 scientists and engineers.

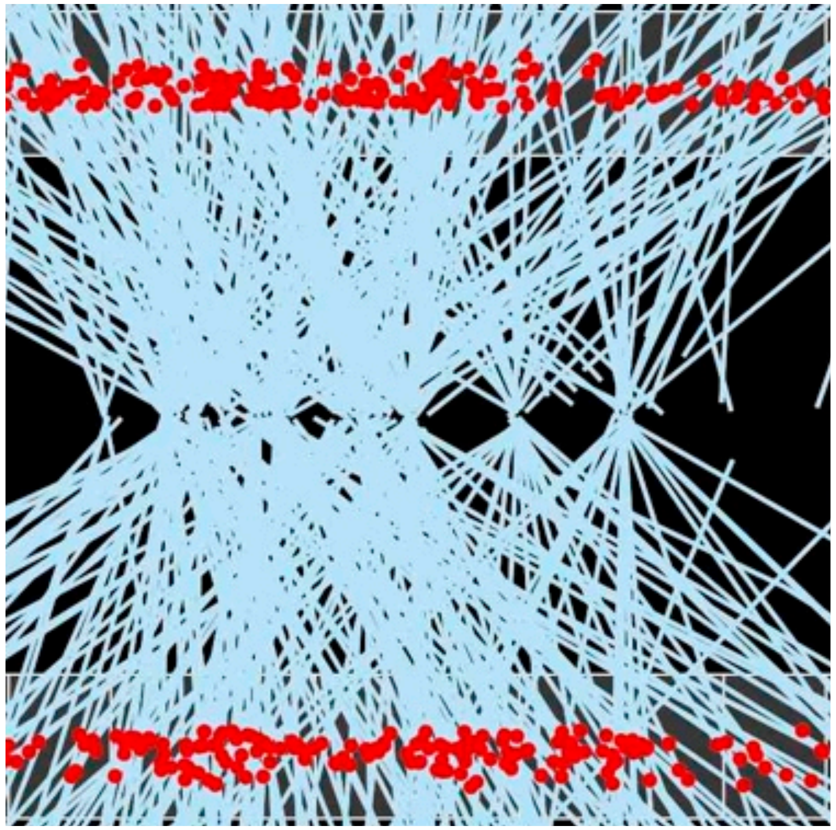


Datasets

The LHC has performed extremely well!!



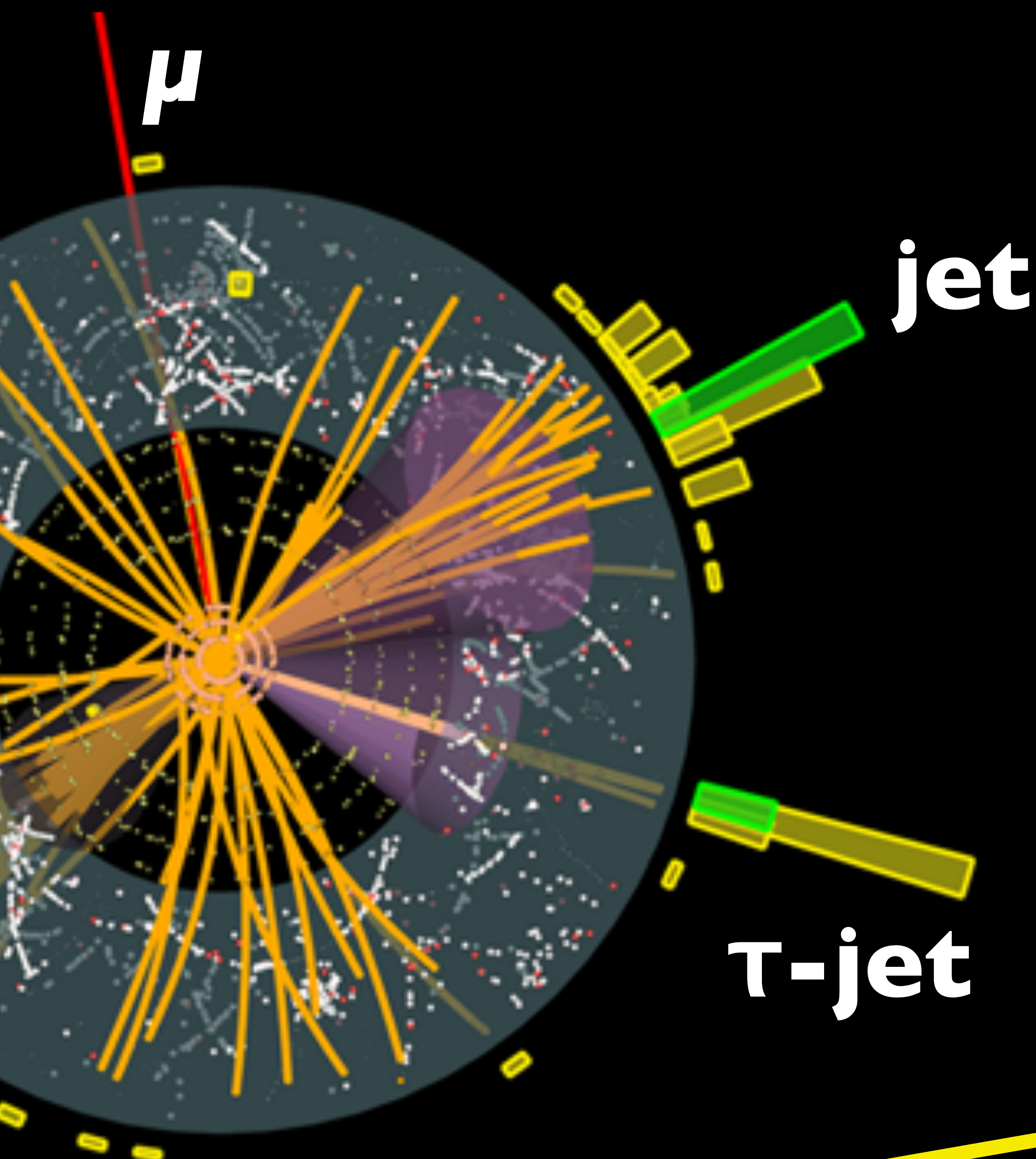
Recently broke inst. lumi. records > 10^{34} cm^{-2}s^{-1}



Typically 20-40 verticies per bunch crossing

Latest analyses combine collision data at $\sqrt{s}$=13TeV collected in the years 2015 and 2016, giving a total integrated lumi ≈ 13-15 fb^{-1}.

What do we reconstruct?



(main objects)

- muons
- electrons & photons
- jets of hadrons
- τ- and *b*-tagged jets
- missing energy

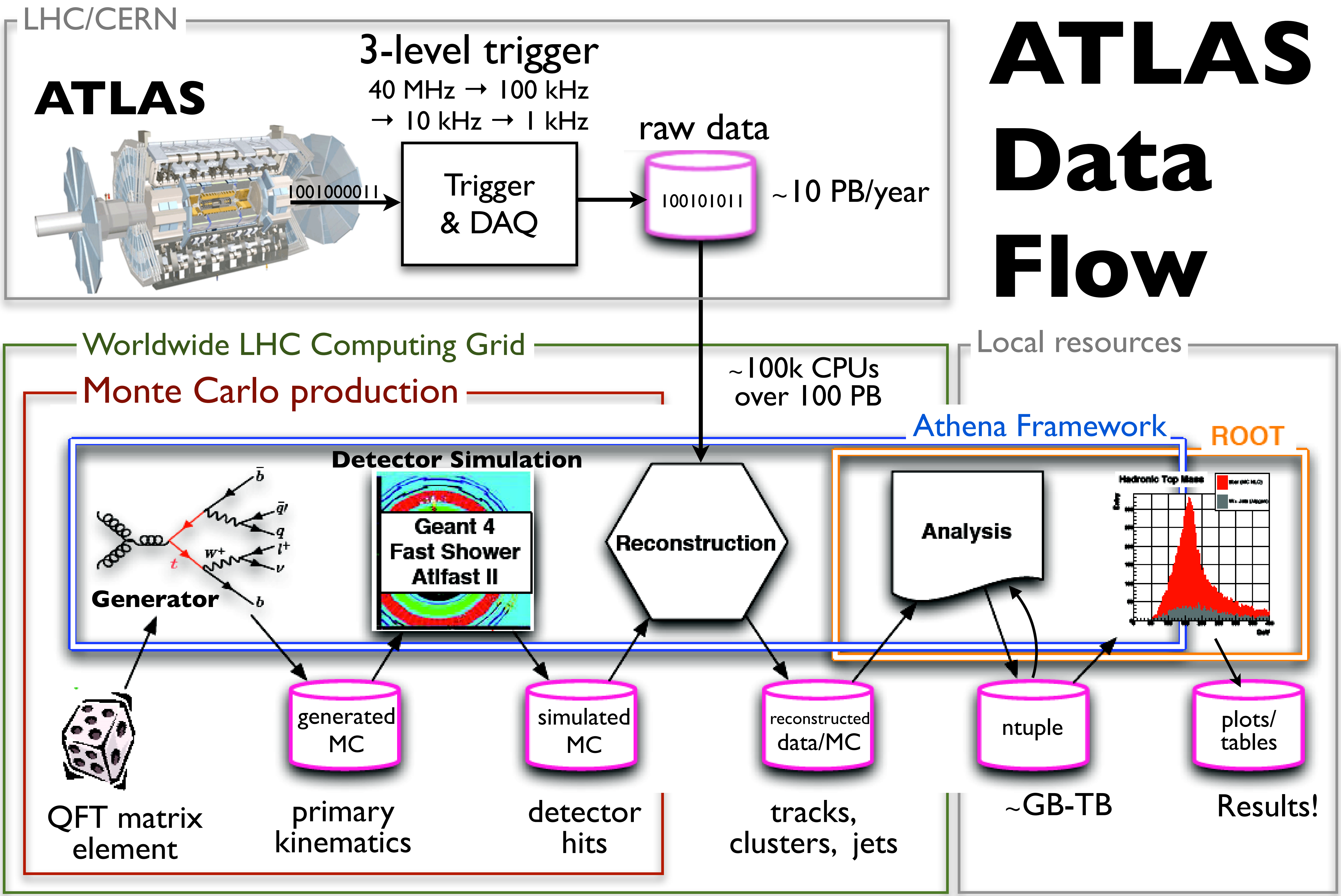
How do we search?

ATLAS Physics Groups

SM	Higgs	SUSY	Exotics
W, *Z*, top,...	*H*→γγ, *ZZ*, *WW*, ...	*l*+jets,γ+jets, ...	*Z*',*W*', ...

Currently ATLAS has published 579+ papers

ATLAS Data Flow

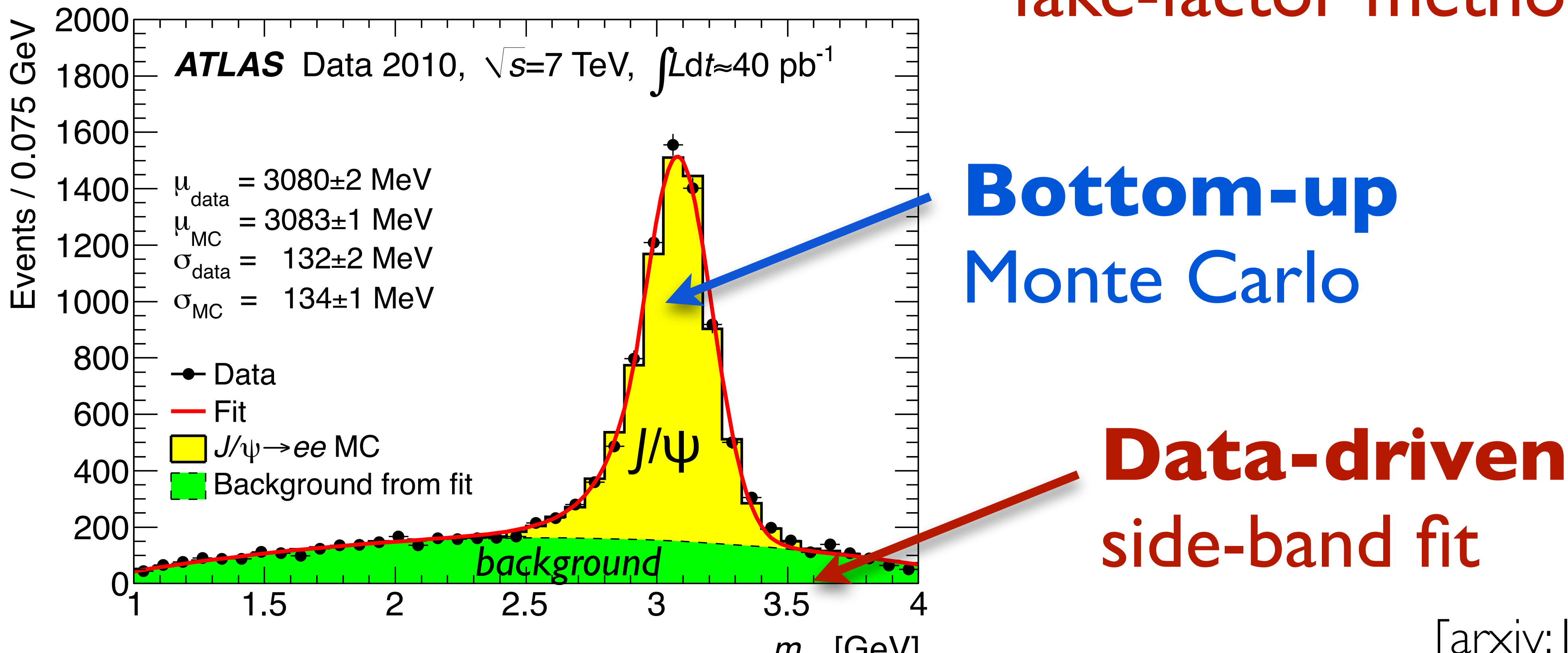


Building a model

$N(expected) = \underbrace{N(correct\text{-}ID)}_{} + \underbrace{N(fake)}_{}$

- ***Bottom-up***
- well-identified objects have scale factors from control regions
- estimated with detailed Monte Carlo simulation

- ***Top-down*, "data-driven"**
- various magic with data depending on the analysis and your creativity
- side-band fit
- fake-factor method



[arxiv:1110.3174]

Tau Reconstruction

- Tau candidates are seeded by anti-k_t calorimeter jets (R=0.4) formed from topological clusters with local hadronic calib.
- Tracks are matched to this calorimeter object and discrimianting variables calculated from the combined tracking+calo information.
- Best vertex chosen from those matching tracks in core cone $\Delta R<0.2$.
- Core track with $\Delta R<0.2$ associated to the tau.
- Annulus $0.2<\Delta R<0.4$ used to calculate tracking and calorimeter isolation variables.
- *New in Run-2*: π^0 counting using strips in EM calorimeter and subtracting charged energy matched to tracks. Improves jet rejection and energy resolution.