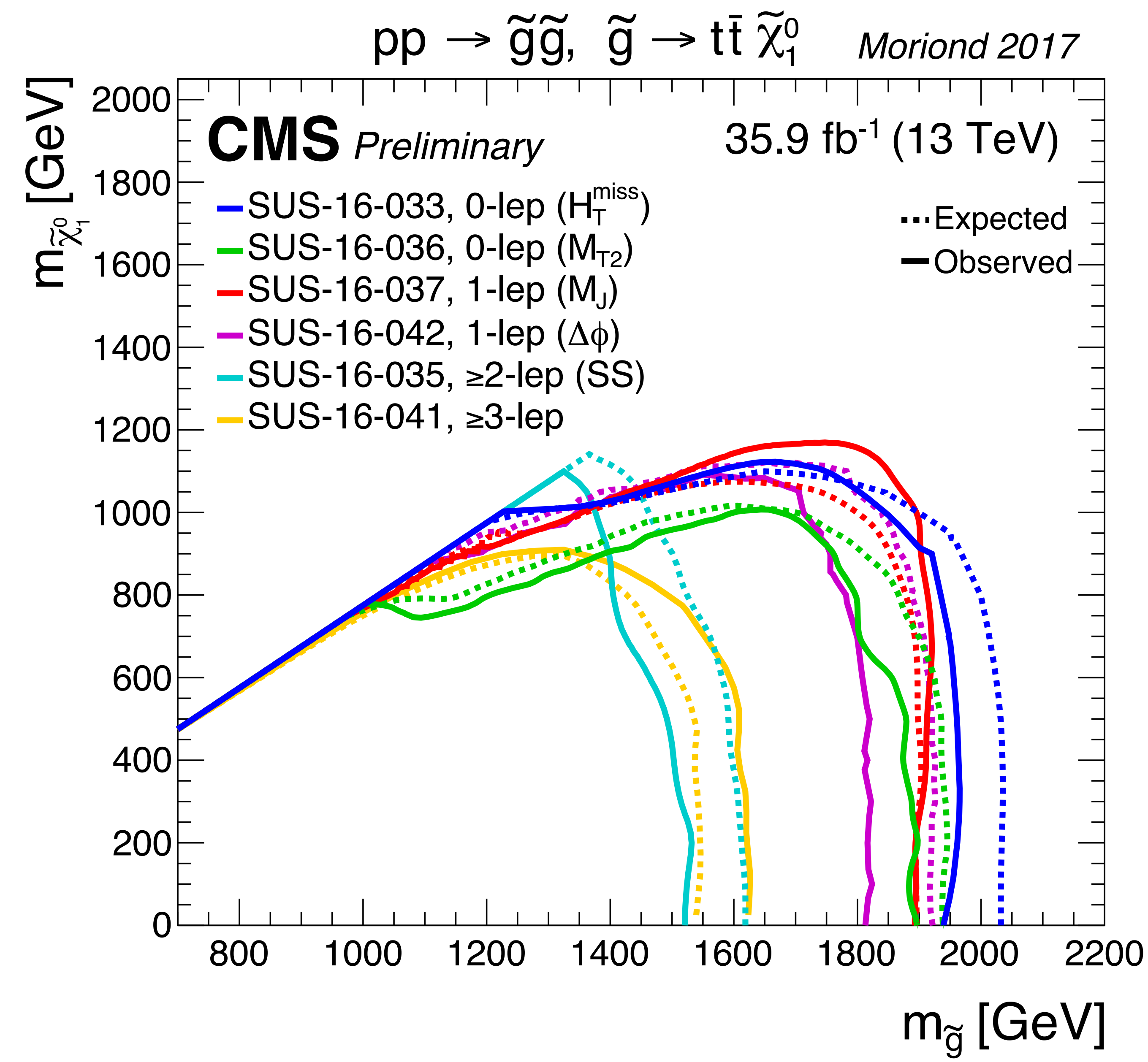
Using Deep Learning to optimise a SUSY search at CMS

Adam Elwood

adam.elwood@desy.de

End of the early data regime at the LHC

- Still no signs of new physics after a significant chunk (36/fb) of 13 TeV data have been analysed
- Early searches aim to be robust and conservative, but better understanding of the data and smaller chance of discovery makes this less necessary (for exploration at least)
- If new physics exists at the LHC it will be in less obvious places
- DL offers a good way to start exploring the phase space more deeply
- Lots of data anticipated in Run 3, which will further increase the utility of ML tools
- Many powerful python-based tools coming out of industry - HEP should consider utilising them where it can



No signs of ‘natural’ SUSY…

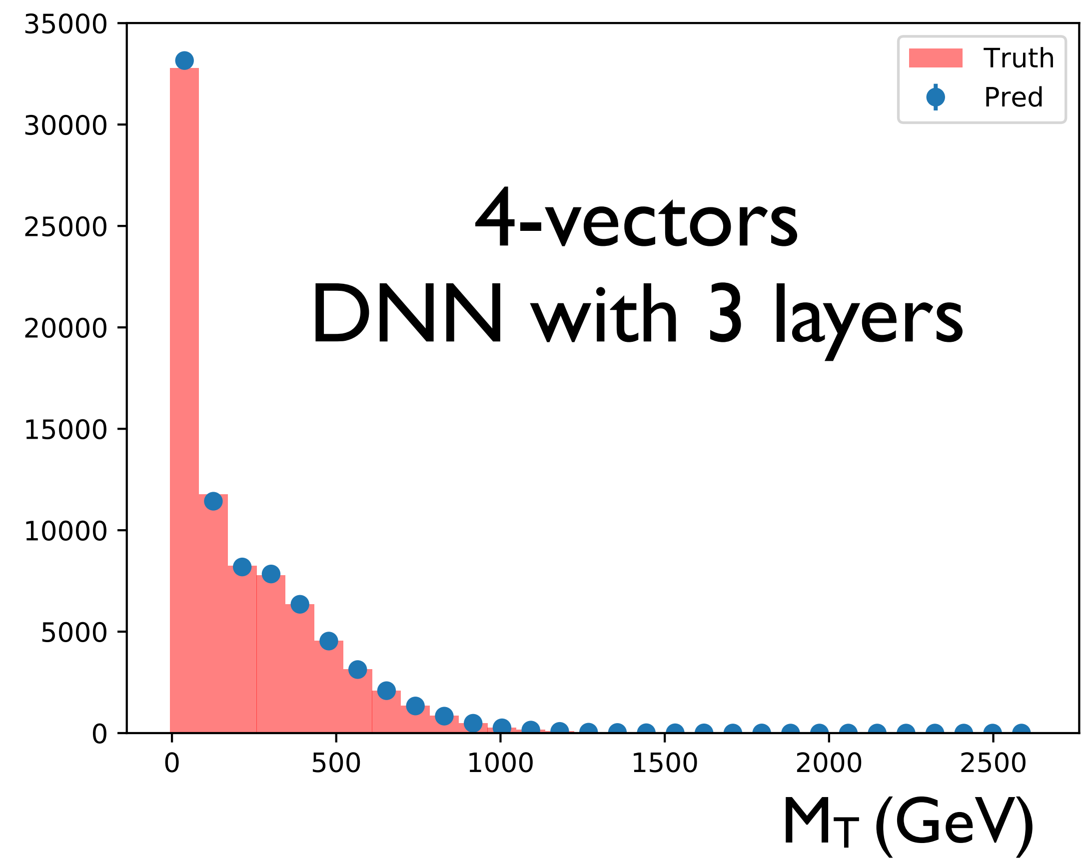
How to apply deep learning to a SUSY search

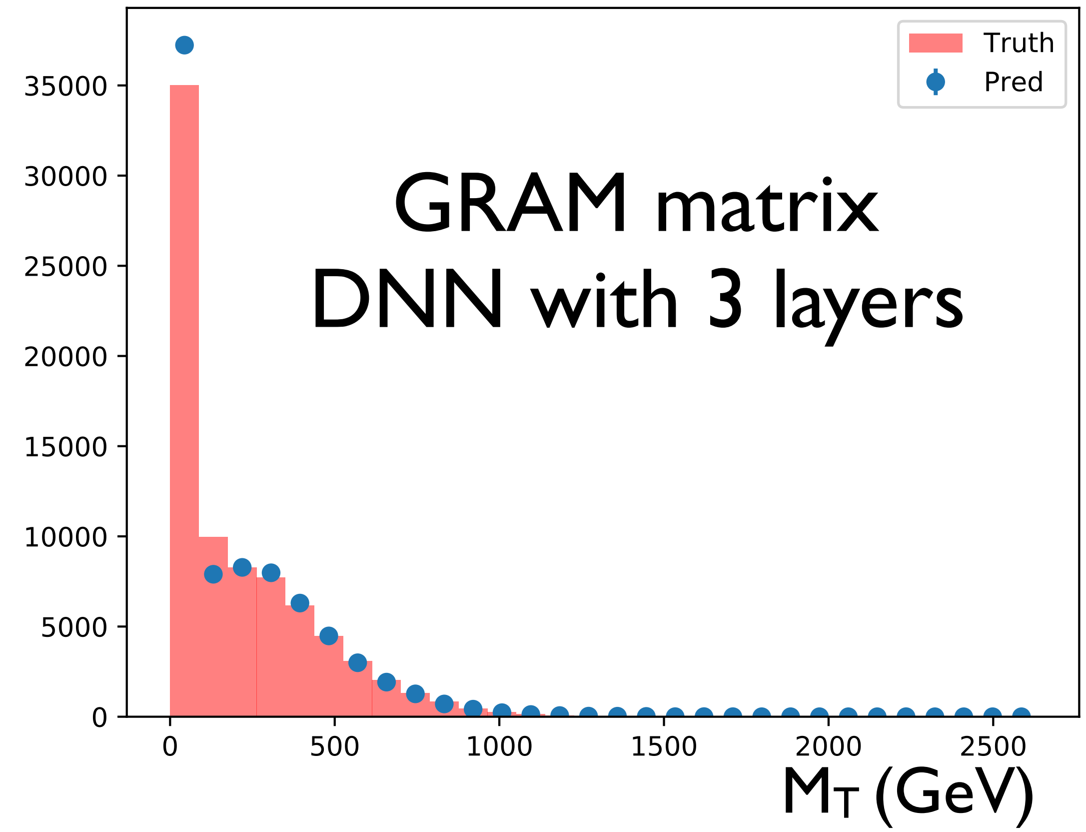
- Physics analysis
  - Develop techniques for intelligent classification of signal vs background, most current work so far has been in object reconstruction
  - DNNs are much better suited to non-binary classification, can gain from classifying multiple types of backgrounds?
  - Can also just use it to optimise binning for a cut based analysis
- Obvious issue is the reliance on simulation for training
  - Only use input variables with good simulation/data agreement * (but take care of correlations)?
  - Train purely on data, one sample signal enriched and one background enriched? **
  - Utilise an adversarial network to ensure the classifier cannot differentiate data from simulation? ***
  - Treat the classifier score as a standard high level variable, compare data and simulation in a control region and use the difference to quantify a systematic?

* https://indico.cern.ch/event/646801/contributions/2651014/attachments/1487701/2311252/6Jul2017_CMSML_systematics.pdf
** http://arxiv.org/abs/1702.00414 *** http://jmlr.org/papers/volume17/15-239/15-239.pdf , https://arxiv.org/abs/1702.05464

Interpretability of results

- Important requirement for new physics searches is the ability of the theory community to reinterpret results
    - This is relatively straightforward with a simple cut and count analysis
    - Can make the ML model available but need to ensure it is not dependent on complicated correlations related to the detector
- Define Lorentz-invariant physical quantities as input instead of 4-vectors?
    - E.g. inner product of all pairs 4-vectors of all objects in an event (GRAM matrix)
    - Need to factor out detector effects?
- Also important to ensure searches are not too model dependent
    - Try some form of anomaly detection?
    - Train over several generic signal models?





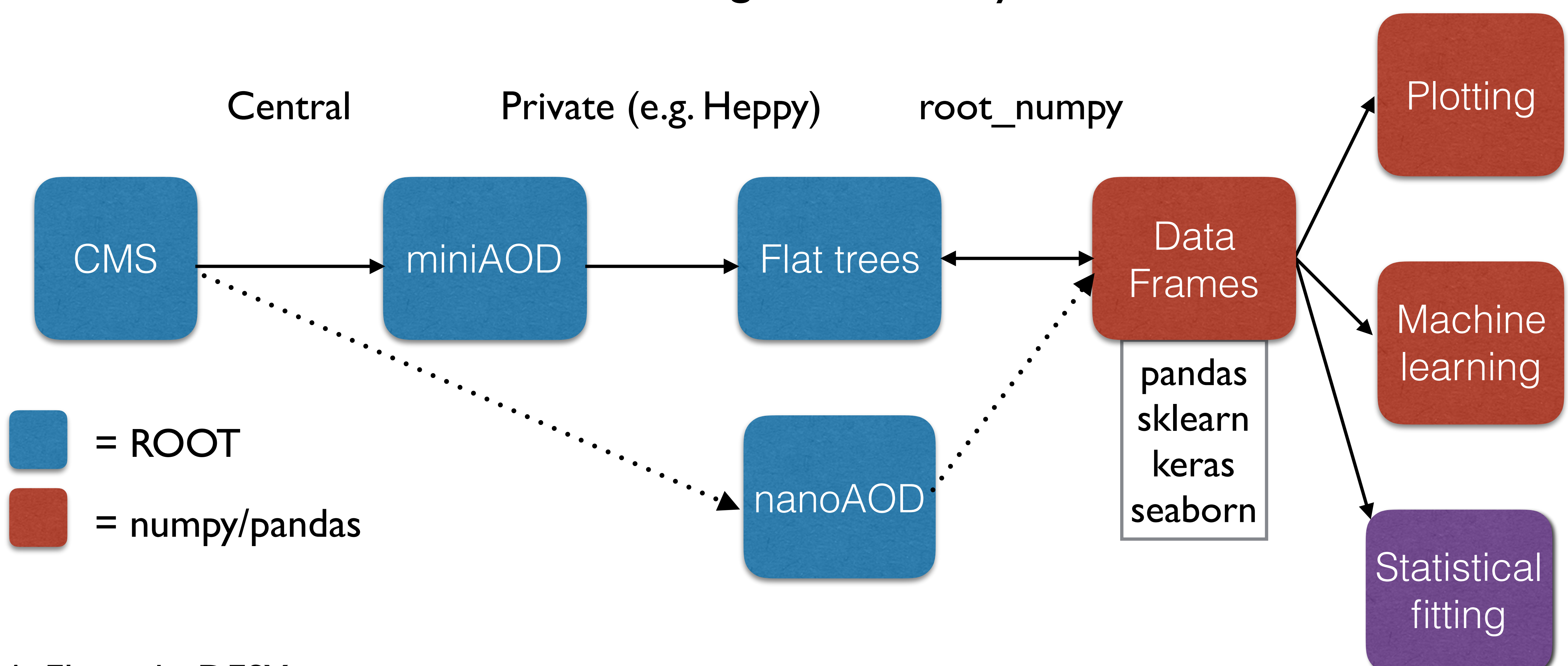
Backup

Data science overview

- In recent years industry has driven a big development in data analysis software
- In particular very powerful and flexible tools for data visualisation and machine learning based around python
- How can we use these tools to our advantage in a physics analysis?
    - Easy and fast data visualisation
    - Cutting edge machine learning techniques
    - Efficient code conveniently accessible through python
    - A software base with industry support
- However, we have to partly move away from the ROOT data format

How we can use the software in a generic analysis

- ROOT is still very good at some things
  - Complex event level calculations
  - Fast and sophisticated event loops
- But not ideal for the final stage of the analysis



GRAM matrix study

- A concrete study gives a reason to develop the tools without the complication of a full analysis framework
- Use simulation made for SVM paper (https://arxiv.org/abs/1601.02809)
  - ~1000 000 events of single stop (signal) and ttbar (background) generated with pythia and Delphes
  - For these results start with 10 000 each of signal and background (for now unweighted) split into a 2/3 training, 1/3 testing set
- Construct a (Lorentz invariant) GRAM matrix from all objects (leptons, jets, MET)
- Use this as input to ML algorithms to see if it can be used to learn information contained in traditional high level variables (M_T, M_{T2W}…)

$$G(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \begin{pmatrix} \langle \mathbf{v}_1, \mathbf{v}_1 \rangle & \cdots & \langle \mathbf{v}_1, \mathbf{v}_k \rangle \\ \vdots & & \vdots \\ \langle \mathbf{v}_k, \mathbf{v}_1 \rangle & \cdots & \langle \mathbf{v}_k, \mathbf{v}_k \rangle \end{pmatrix}$$

Higher level variable learning