Deep Learning with Python

Dan Guest Lots of other people

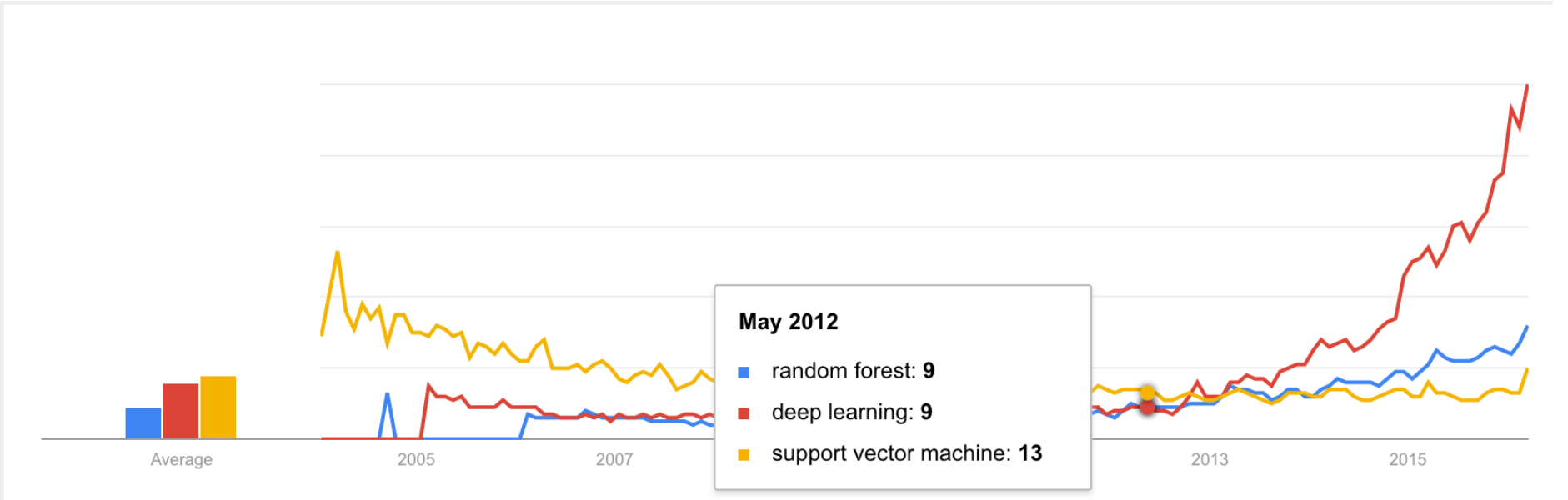
UC Irvine

Lots of other institutes

March 4, 2016

Setting the stage: What's Cool

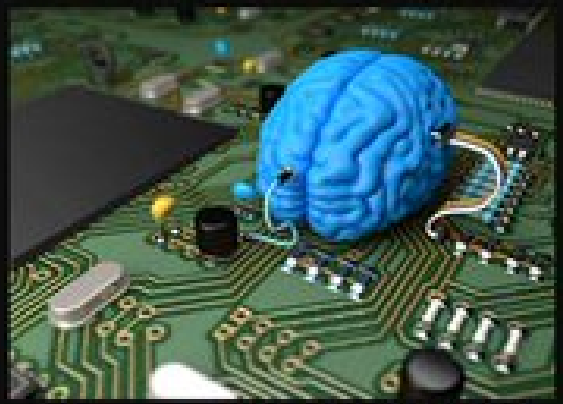
- Results from Google Trends



- Obviously this is mostly buzz
- But deep learning *has* become very popular

Deep Learning

What society thinks I do



What my friends think I do

What other computer scientists think I do

What mathematicians think I do

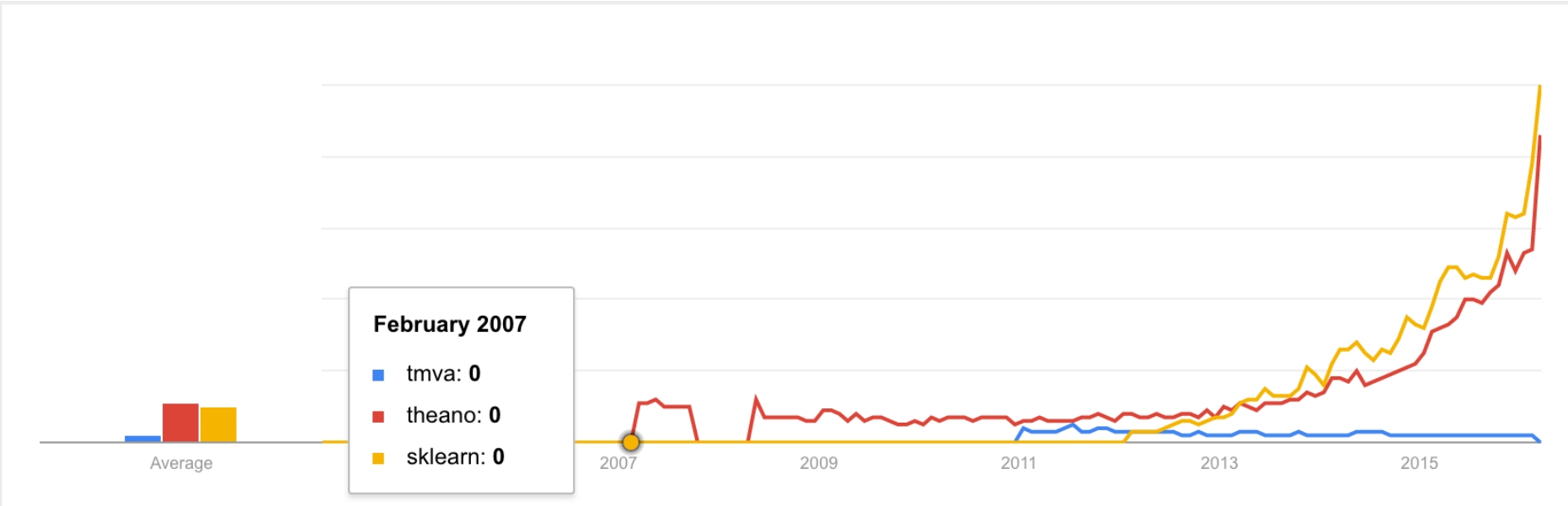
What I think I do

```
from theano import *
```

What I actually do

Setting the stage: What do people use?

- Google Trends



- Also arguably just buzz
- But could predict where the software will go
- We should be careful: we don't want to miss the boat

What about us (ATLAS in my case)?

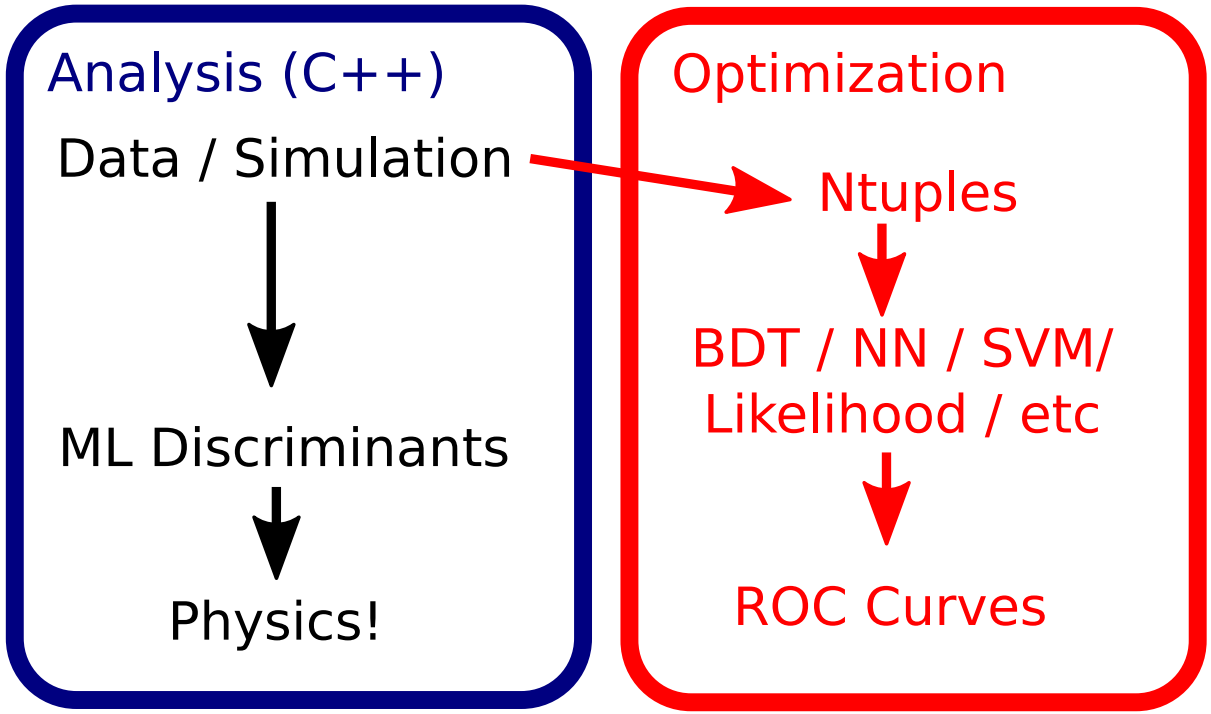
- Everyone uses TMVA
  - Lots of talk about adding features
- But other (python-based) packages have some advantages:
  - Supported by outside community
  - We don't have to write yet another framework
  - Python is easy
  - **The software is already there**

So what's holding us back?

- Inertia: TMVA is already used
- Lack of "glue packages" to work with our existing code
  - ROOT doesn't interface well with numpy/hdf5/python

Our software framework

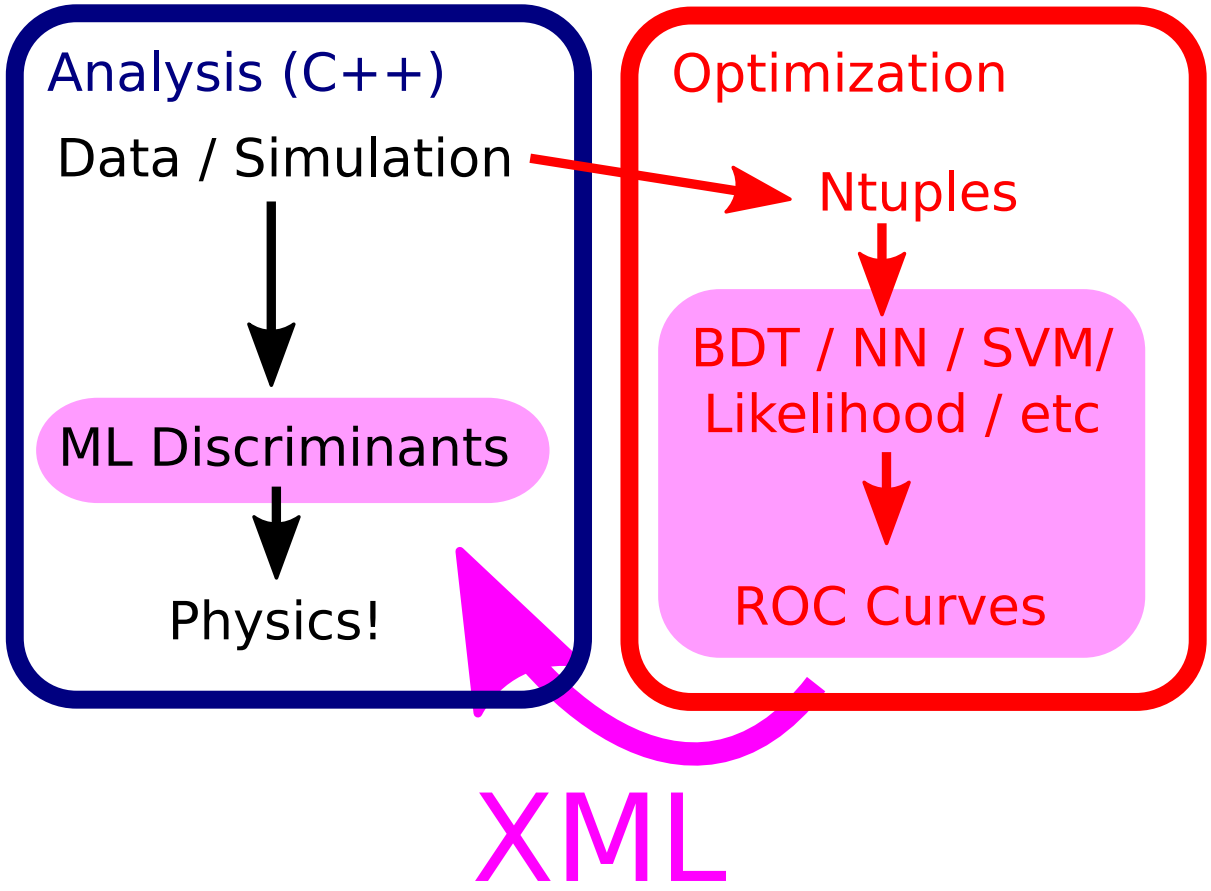
- Our analysis runs (almost entirely) in C++
- Optimization is unconstrained



- Lots of interesting results (Jet Images, etc) that haven't been ported back to C++

ROOT Solution

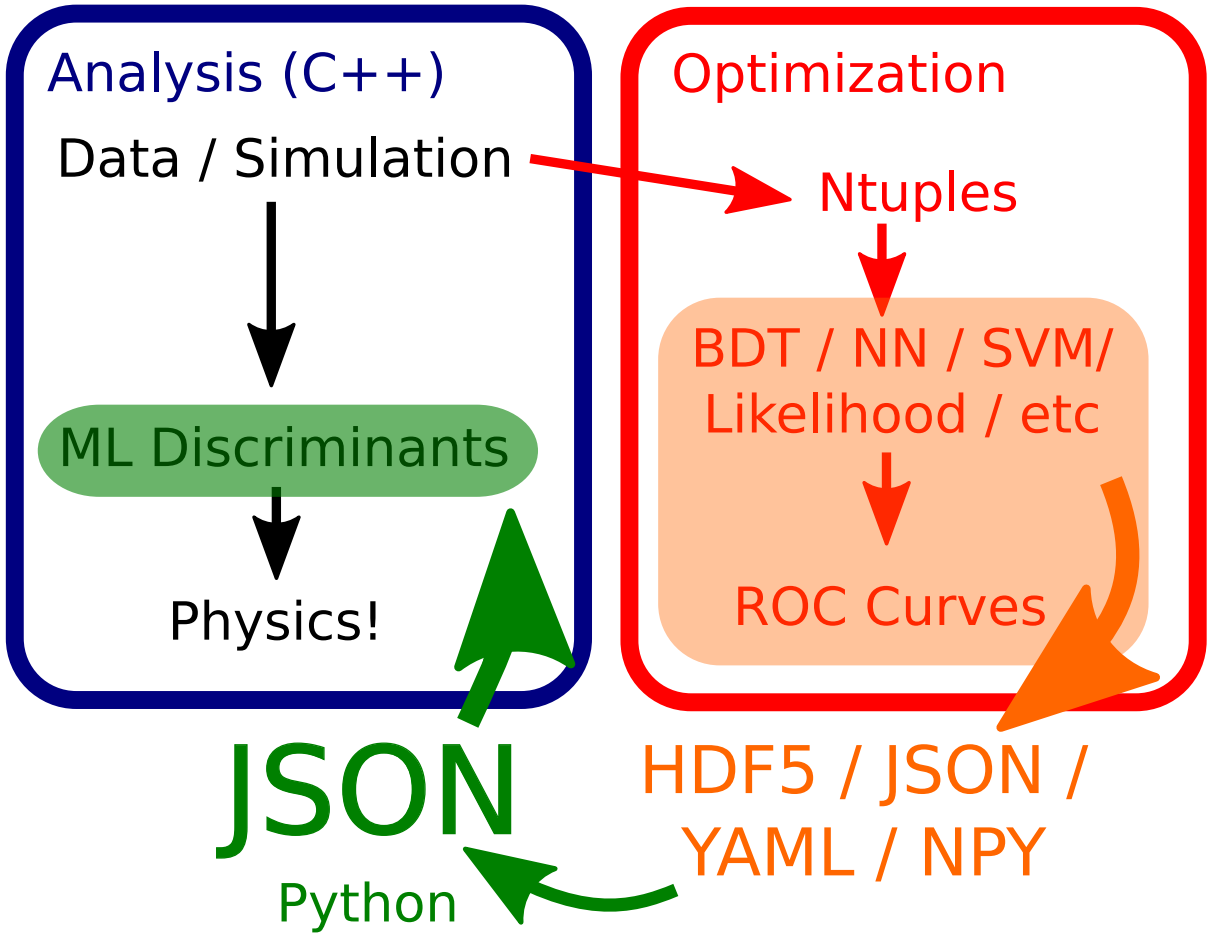
- TMVA works in C++!
- We can convert to TMVA xml
- . . . sometimes



- We can port sklearn BDTs to TMVA

Our Solution

- Use whatever we want (Keras)
- Write the C++ class



- Use root_numpy to convert ROOT to numpy
- Save NN as JSON, write minimal C++ code for classification

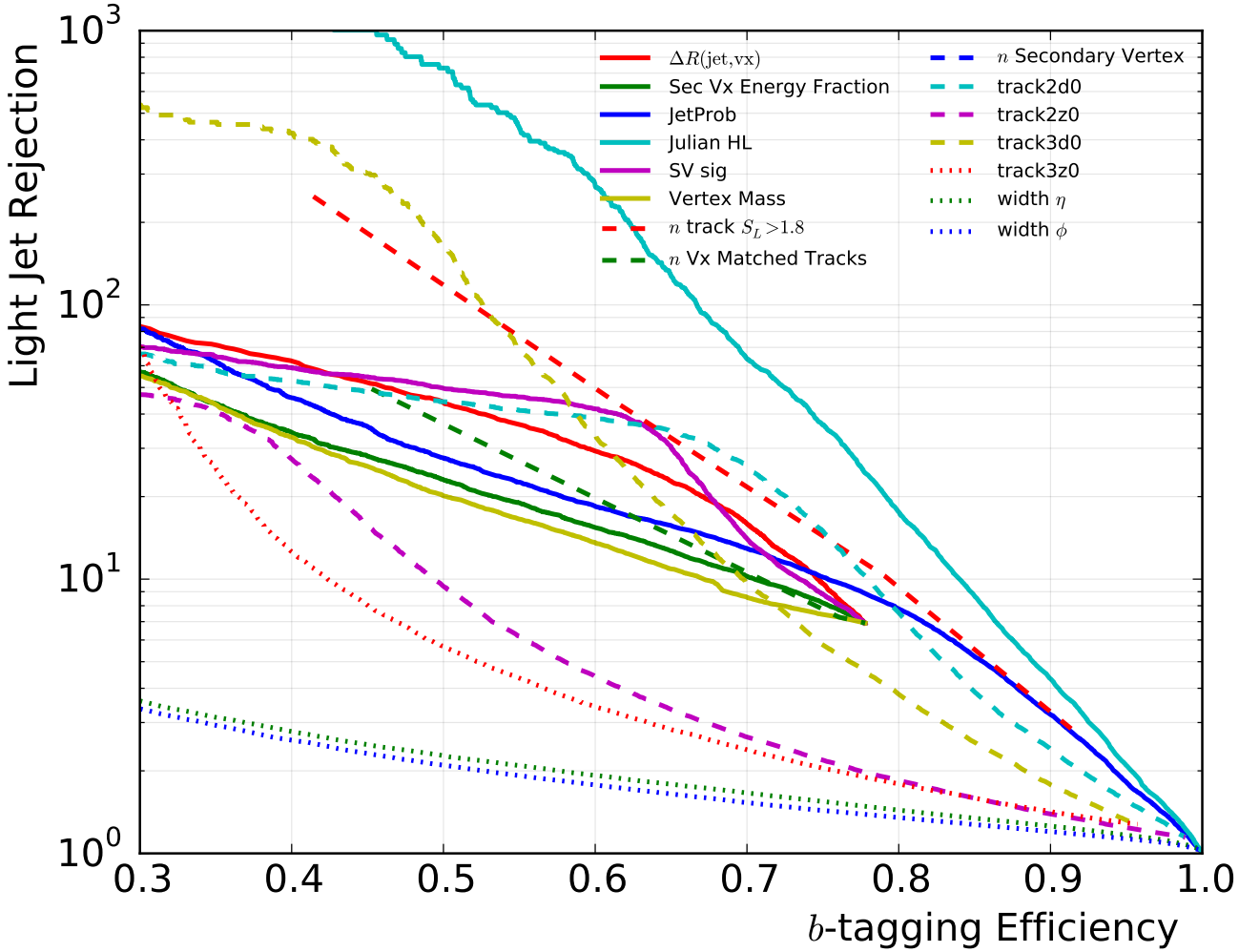
Lightweight Neural Networks

- Very powerful (I'm not going to show you a cat)
- Not very complicated!

$$\boldsymbol{y} = A_{N-1} f(\cdots A_2 f(A_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_2 \cdots) + \boldsymbol{b}_{N-1}$$

- With a few activation functions we cover 90% of use cases
- Code is on github: https://github.com/dguest/lw-client
  - Thoroughly unimpressive!
  - Use Eigen, Boost, but **no other dependencies**
  - Currently supports dense layers, but CovNNs should also be easy
  - RNNs. . . maybe a bit more work
    - First we have to prove they are useful!
  - Supports AGILEPack output, plan to add Keras, pylearn2, anything else?

Does it work?



It works ok!

What Now?

- Don't write more frameworks, we already have them!
  - Python-based (numpy, pandas, sklearn, theano) analysis is extremely popular outside HEP
  - but more "glue package" support would help
- If you want to use deep learning *right now* we should talk