Machine Learning in HEP : trends and successes

David Rousseau

LAL-Orsay

rousseau@lal.in2p3.fr @dhpmrou

ILP ML Day, IHP, Paris, 16th Nov 2018

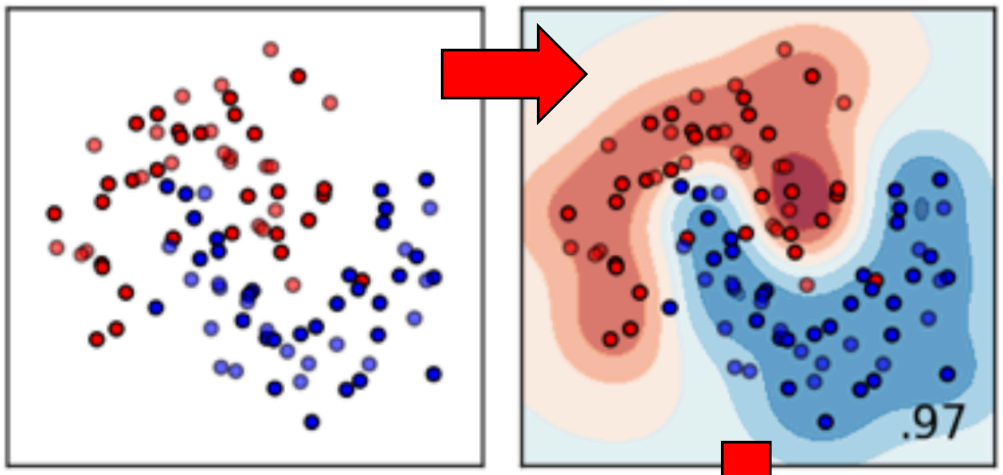
Outline

- ML basics
- ML in analysis
- ML in reconstruction
- ML in simulation
- Wrapping up

- Focus on applications rather than details of the techniques
- Deliberately incomplete (sorry…)
- No likelihood free inference, no classification without labels, no review on ML software, no application to distributed analysis, no GAN to uniformity, no Bayes optimisation, no reinforcement learning, no adversarial example, no probablilistic programming, no learning with quantum computing….

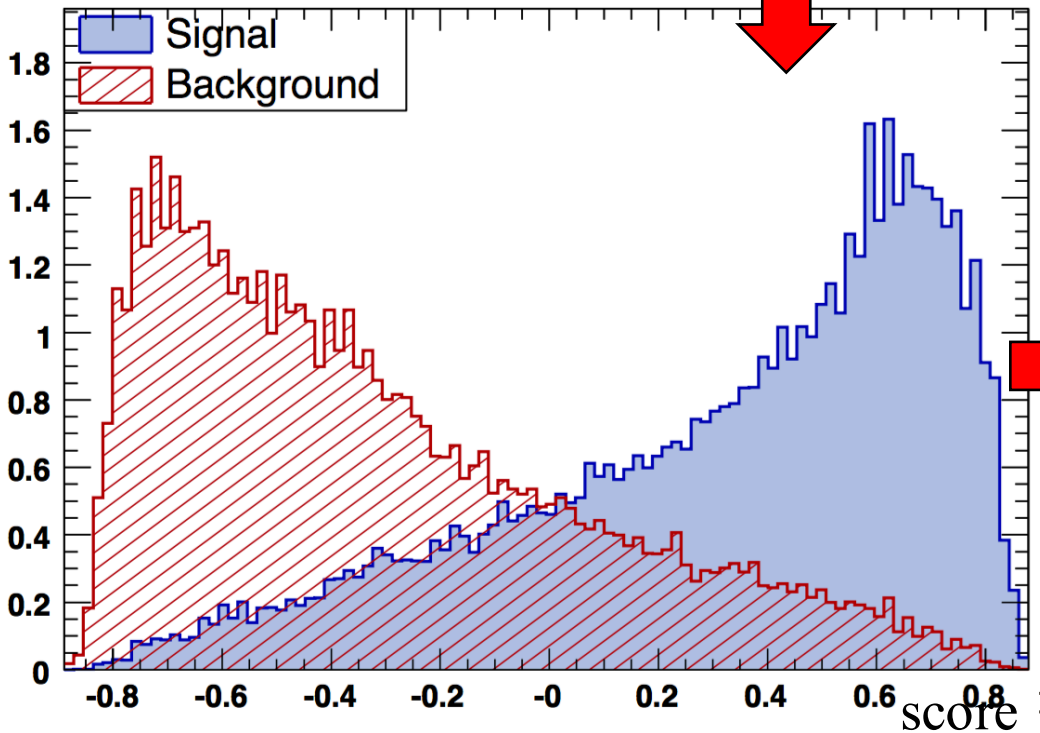
ML Basics

Classifier basics

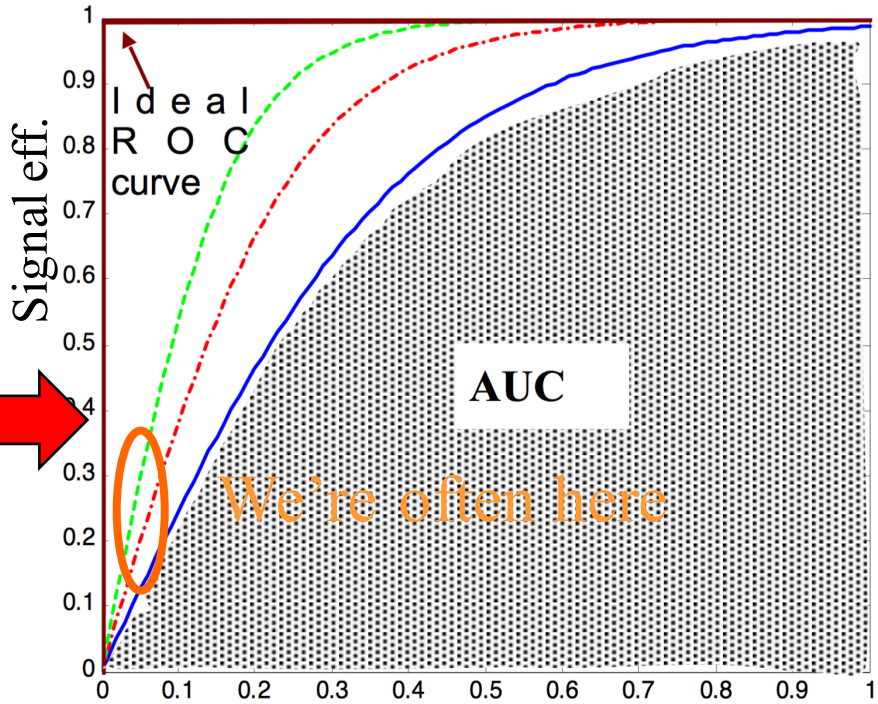


Train on Signal and Background Monte-Carlo
➔learn the separation between S and B distribution
Apply on test sample
Apply on data

Note: instead of *classifying* 0 or 1, can *regress* !



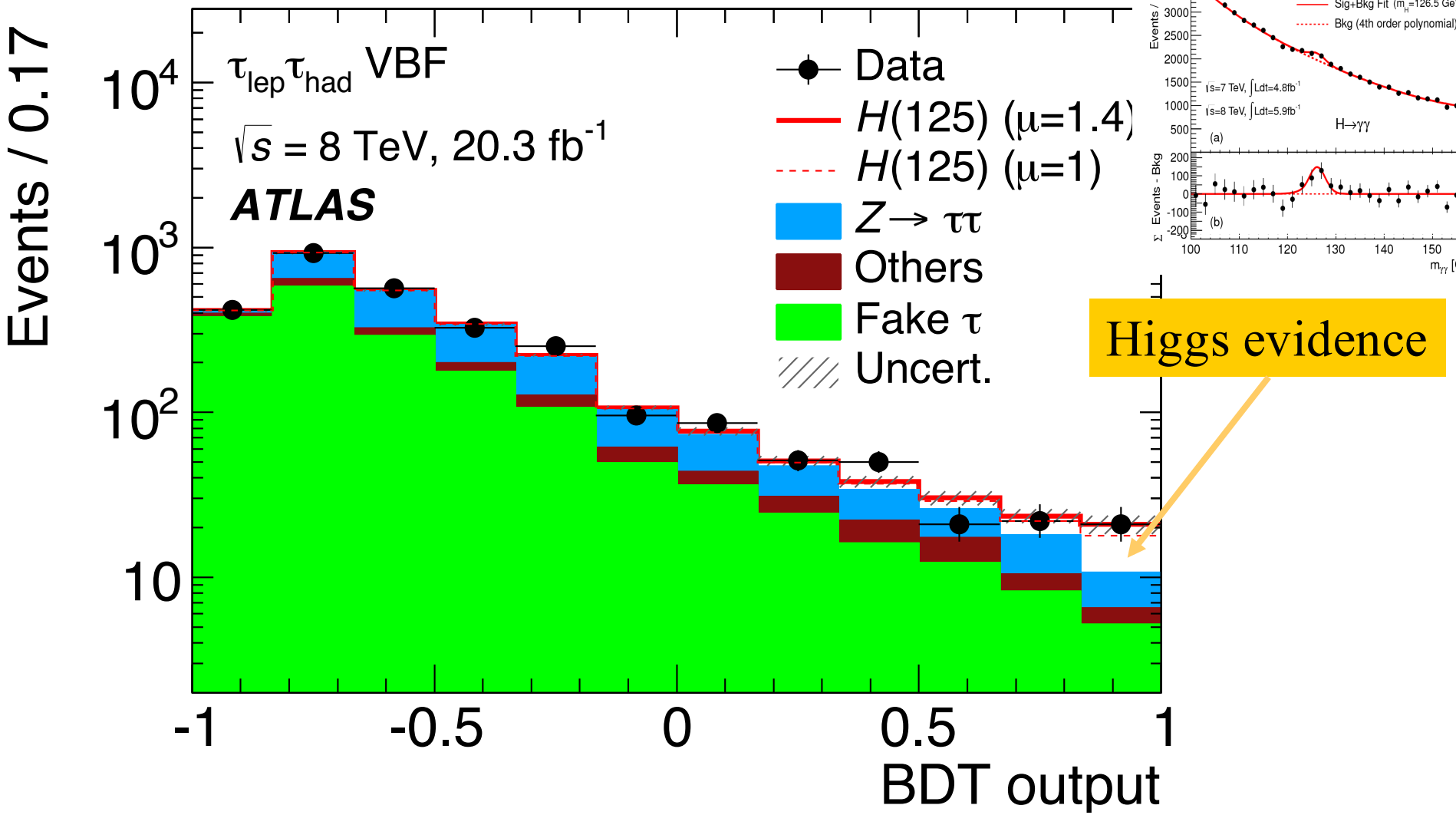
AUC : Area Under the (ROC) Curve



Classifier (2)

JHEP 04, 117 (2015) 1501.04943

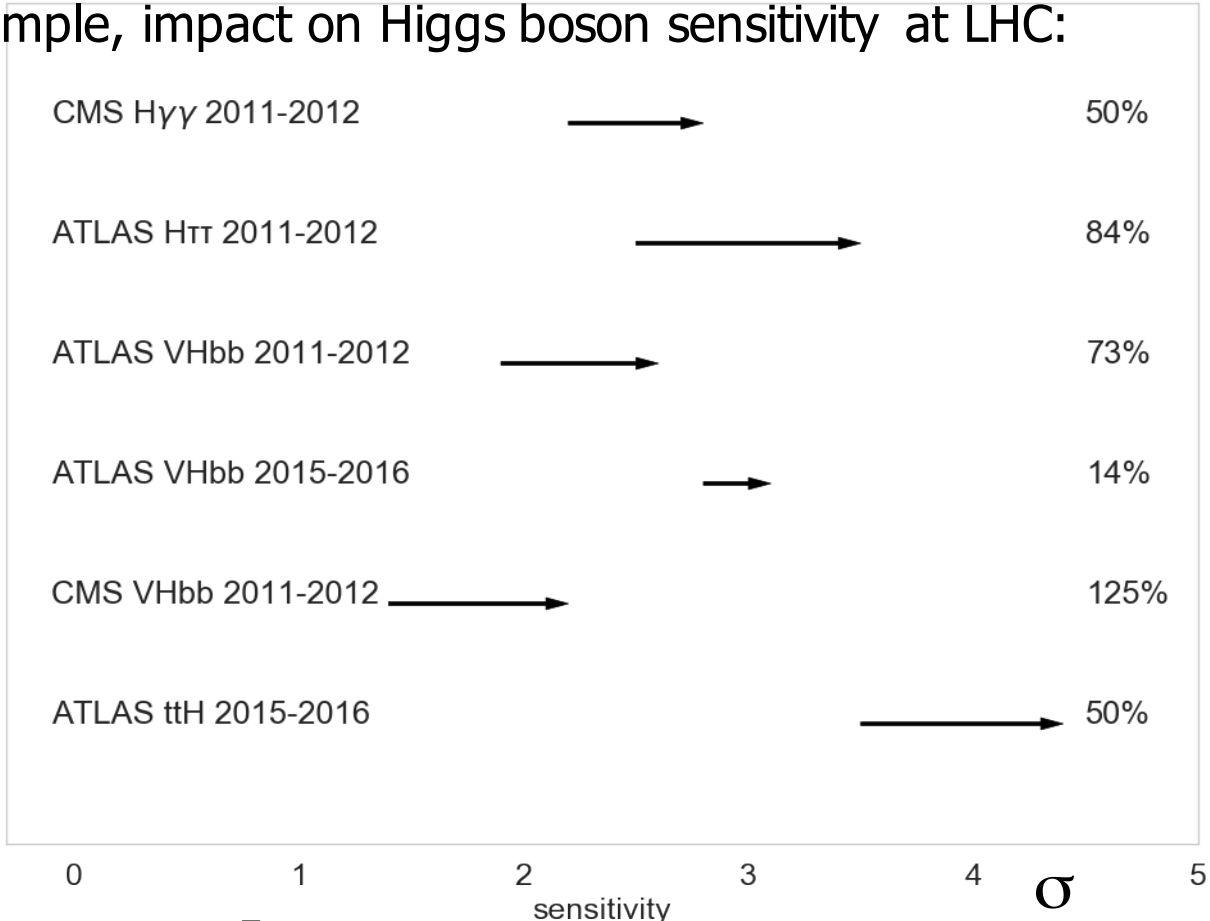
Boosted Decision Tree using ~a dozen of high level variables



Higgs evidence

ML on Higgs Physics

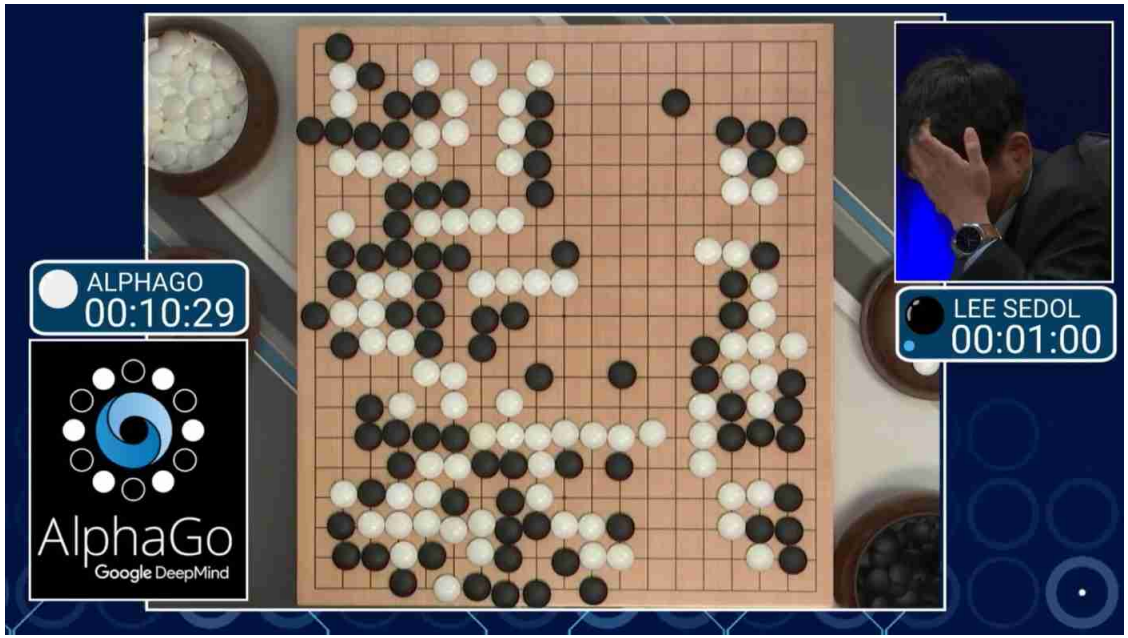
- At LHC, Machine Learning used almost since first data taking (2010) for reconstruction and analysis
- In most cases, Boosted Decision Tree with Root-TMVA, on ~10 variables
- For example, impact on Higgs boson sensitivity at LHC:



➔~50% gain on LHC running

ML in HEP

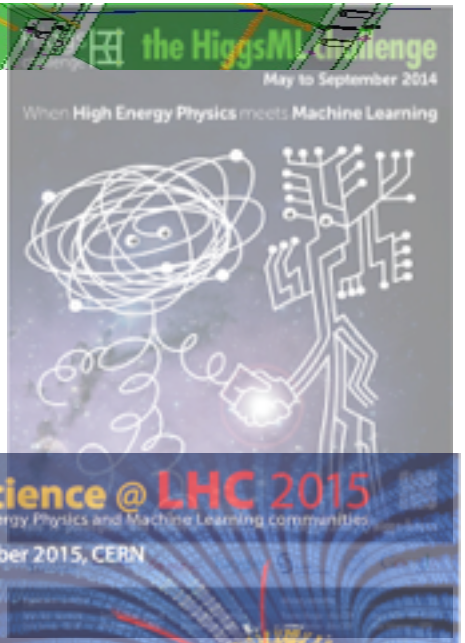
- Meanwhile, in the outside world :

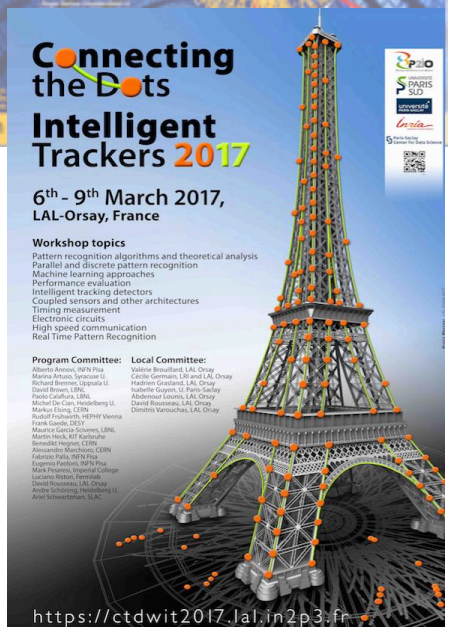


- "Artificial Intelligence" not a dirty word anymore!
- We (in HEP) have realised we're been left behind! Trying to catch up now…

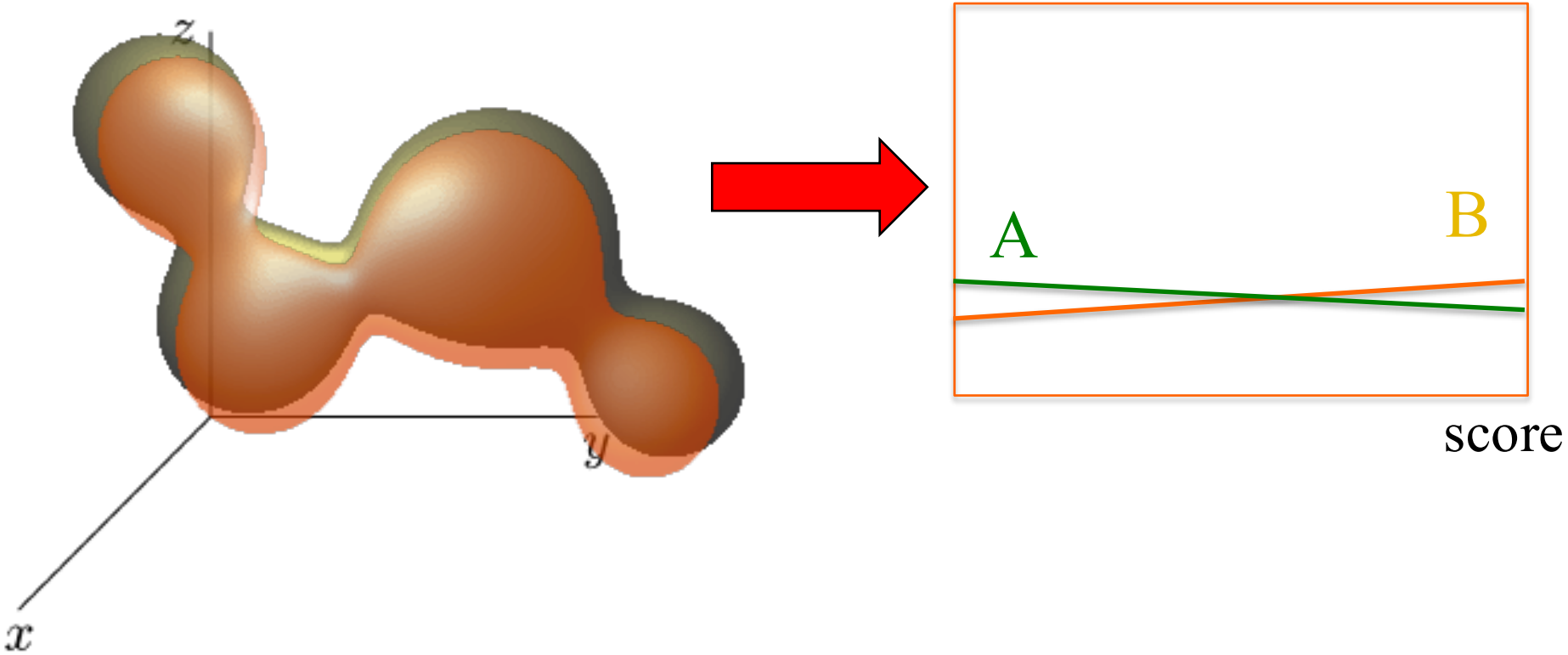
Multitude of HEP-ML events

- HiggsML Challenge, summer 2014
  - ➔HEP ML NIPS satellite workshop, December 2014
- Connecting The Dots, Berkeley, January 2015
- Flavour of Physics Challenge, summer 2015
  - ➔HEP ML NIPS satellite workshop, December 2015
- DS@LHC workshop, 9-13 November 2015
- Moscou/Dubna ML workshop 7-9th Dec 2015
- Heavy Flavour Data Mining workshop, 18-21 Feb 2016
- Connecting The Dots, Vienna, 22-24 February 2016
- Hep Software Foundation workshop 2-4 May 2016 at Orsay, ML session
- Connecting The Dots, LAL-Orsay, 6-9 March 2017
- LHC Interexperiment Machine Learning group
  - IML workshop @CERN 20-22 March 2017, 9-12 April 2018
- DS@HEP workshop @FNAL 8-12 May 2017
- Hammers and Nails, Weizmann, Jul 2017
- ACAT conference Seattle, Sep 2017
- Connecting The Dots, 20-22 March 2018
- CHEP, July 2018 (ML now acknowledged in Track name)
- Tracking ML challenge, summer 2018
- ACAT conference, Saas Fe, March 2019





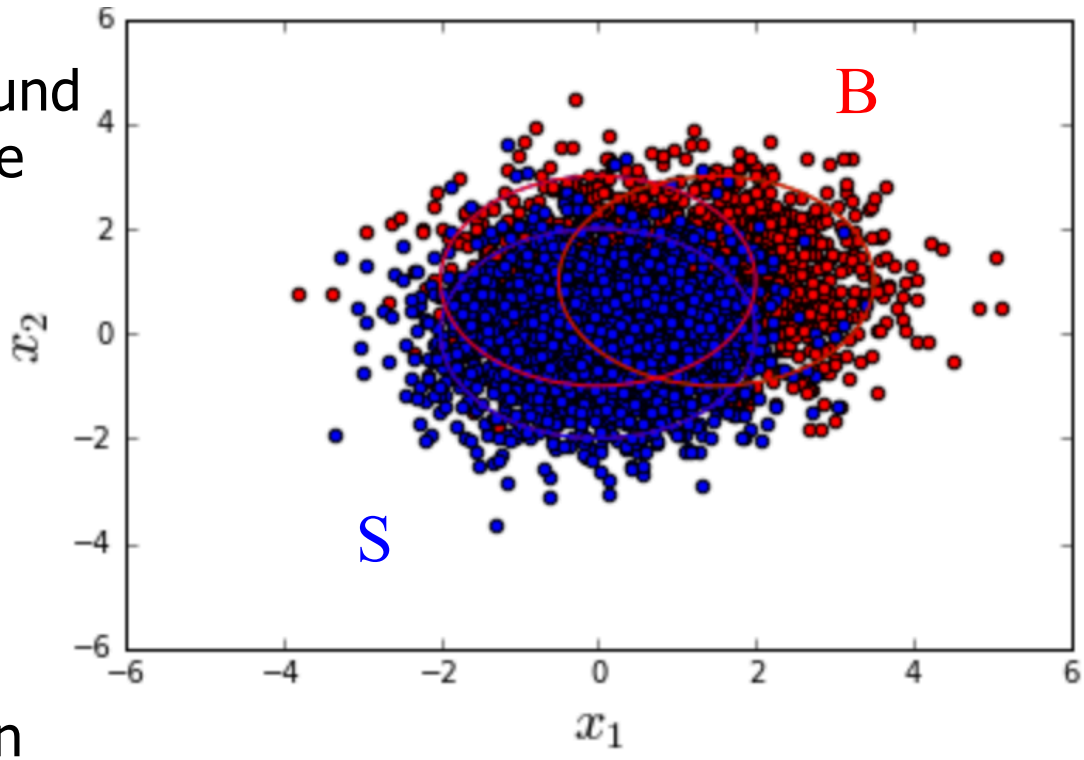
What does a classifier do?



- The classifier "compresses" the two multidimensional "blobs" maximising the difference, without (ideally) any loss of information

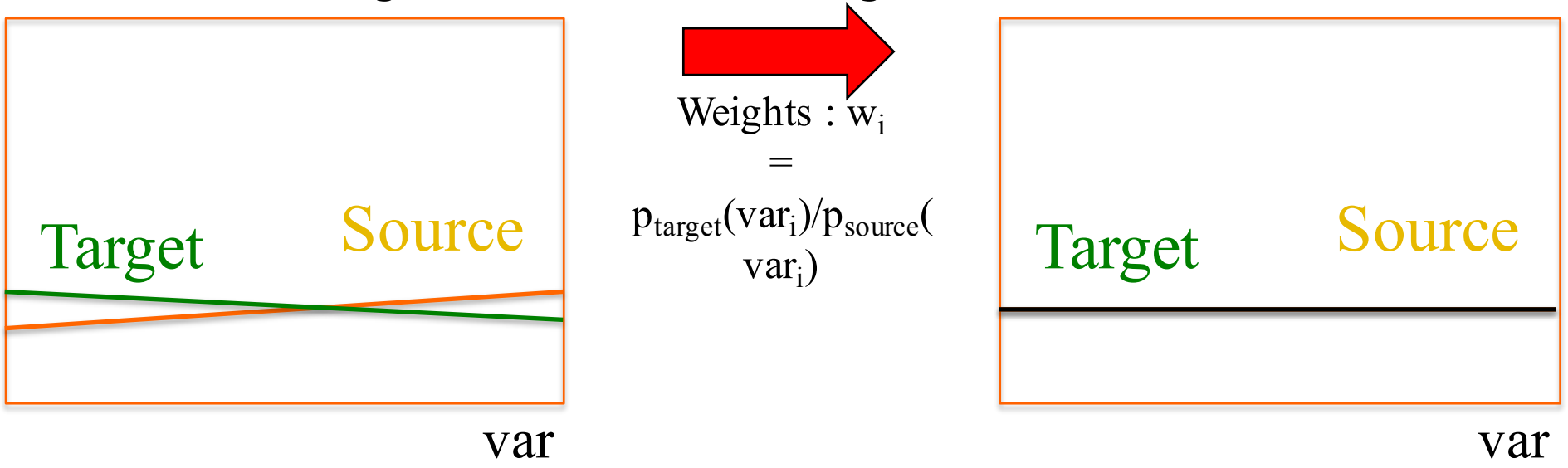
No miracle

- ML (nor Artificial Intelligence) does not do any miracles
- For selecting Signal vs Background and underlying distributions are known, nothing beats ihihood ratio! (often called "Bayesian limit"):
  - $L_S(x)/L_B(x)$
- OK but quite often L_S L_B are unknown
  - + x is n-dimensional
- ML starts to be interesting when there is no proper formalism of the pdf
- ➔mixed approach, if you know something, tell your classifier instead of letting it guess



Re-weighting

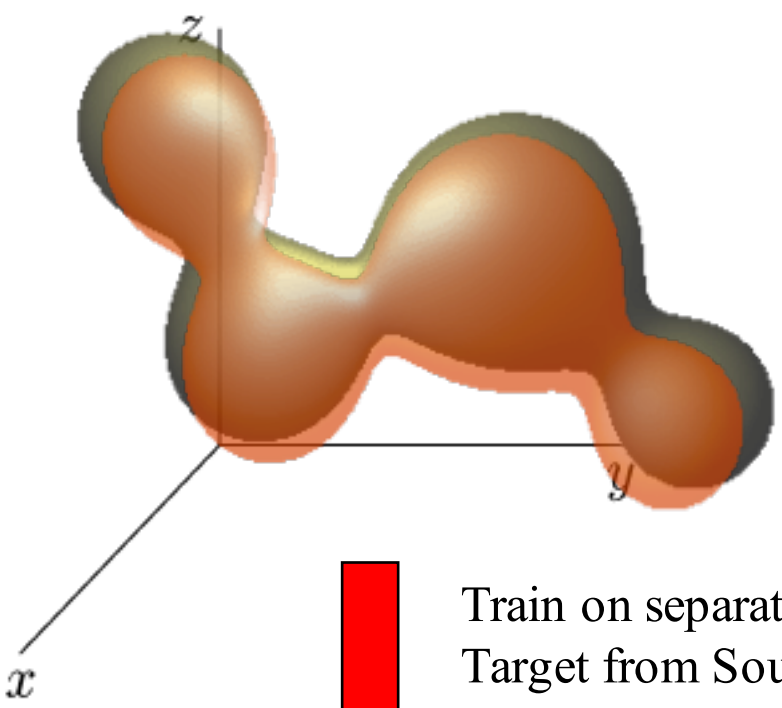
- Suppose a variable distribution is slightly different between a Source (e.g. Monte Carlo) and a Target (e.g. real data)
  - ➔reweight! …then use reweighted events



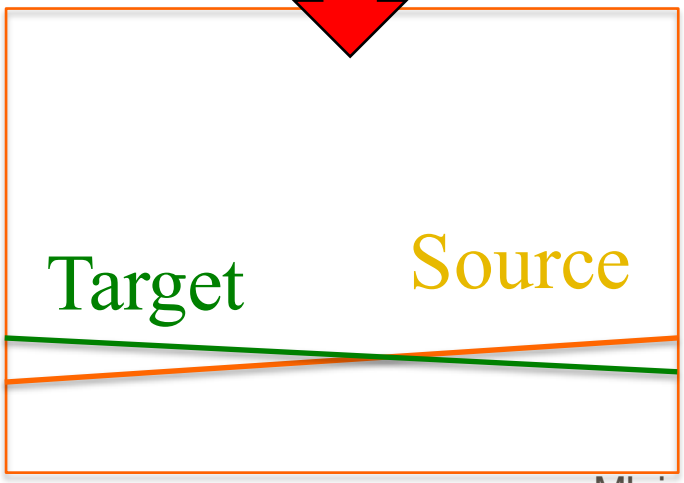
- What if multi-dimension ?
- Usually : reweight separately on 1D projections, at best 2D, because of quick lack of statistics
- Can we do better ?

Multidimension reweighting

See demo on Andrei Rogozhnikov github and also Kyle Cranmer's github Related : uBoost

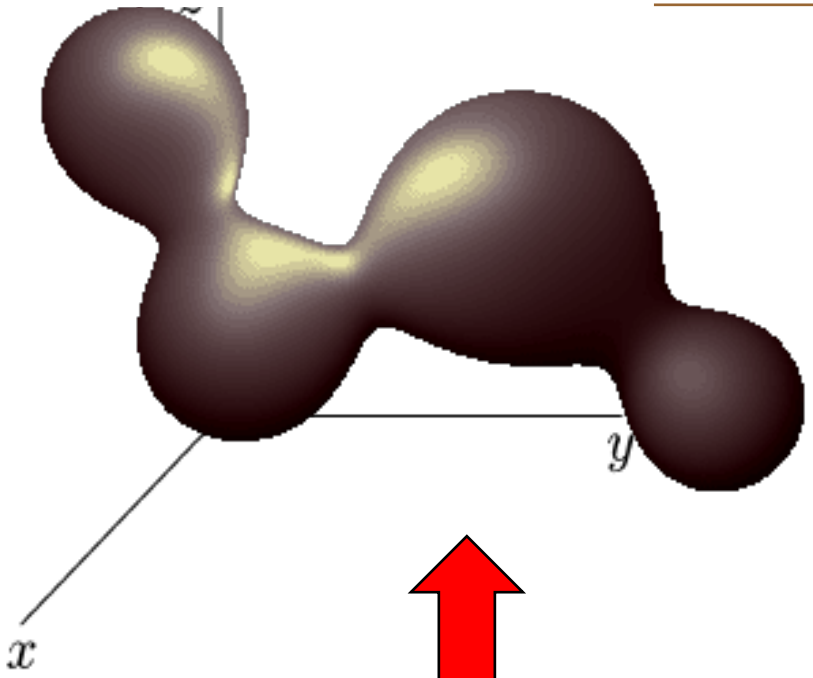


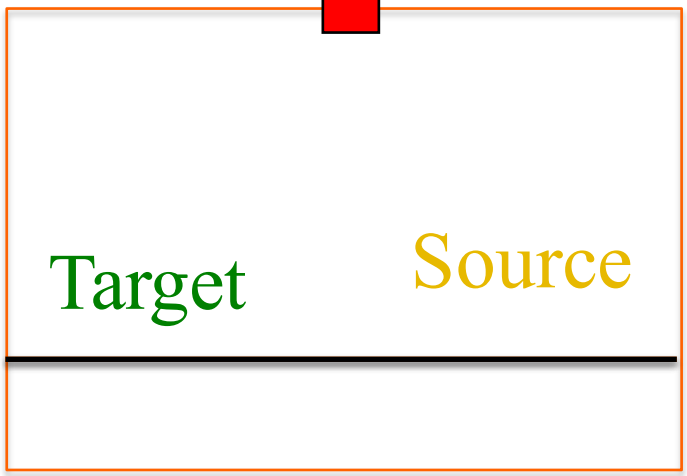
Train on separating
Target from Source



score

Weights : w_i
=
$P_{target}(score_i)/p_{source}(score_i)$





score

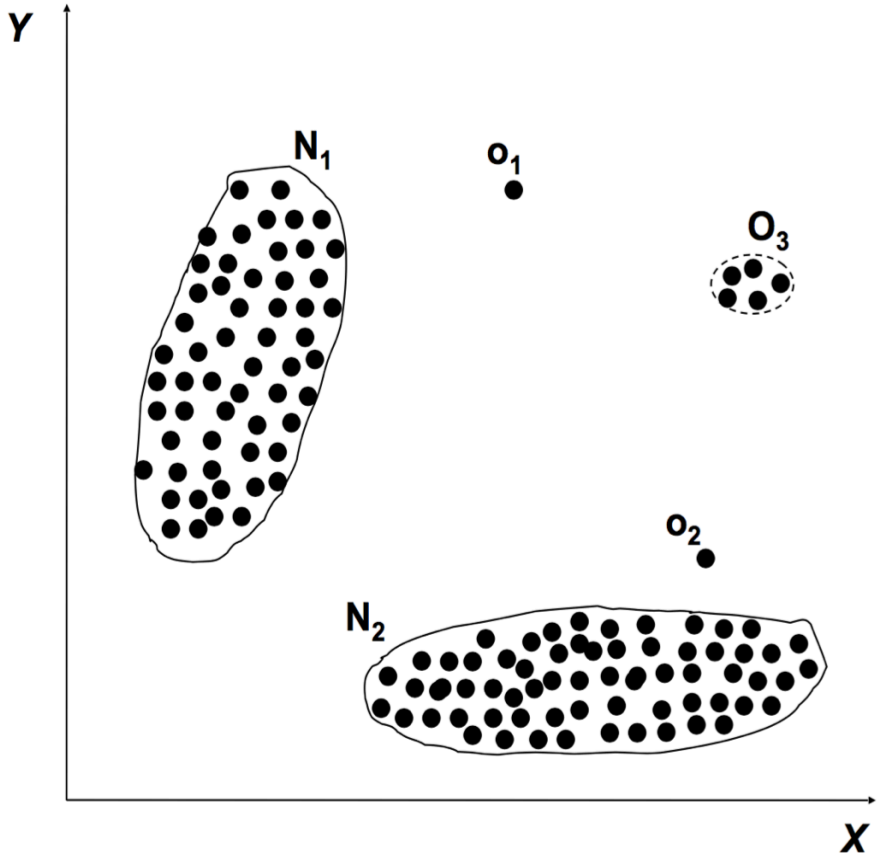
Multi dimensional reweighting (2)

- Reweighting the Source distribution on the score allows multidimensional reweighting without statistics problem
- Usual caveat still hold : Target support should be included in Source support, distributions should not be too different otherwise unmanageable very large or very small weights
- (Note : "reweighting" in HEP language <==> "importance sampling" in ML language)
- Only use (that I know off) in published analyses in LHCb
- Why ?

Anomaly detection

Anomaly detection

- Three approaches:
  - Supervised : model for O and N
  - Semi-supervised : model for N, O is non-N
  - Unsupervised : give the full data, ask the algorithm to cluster N and find the lone entries : o1, o2, O3

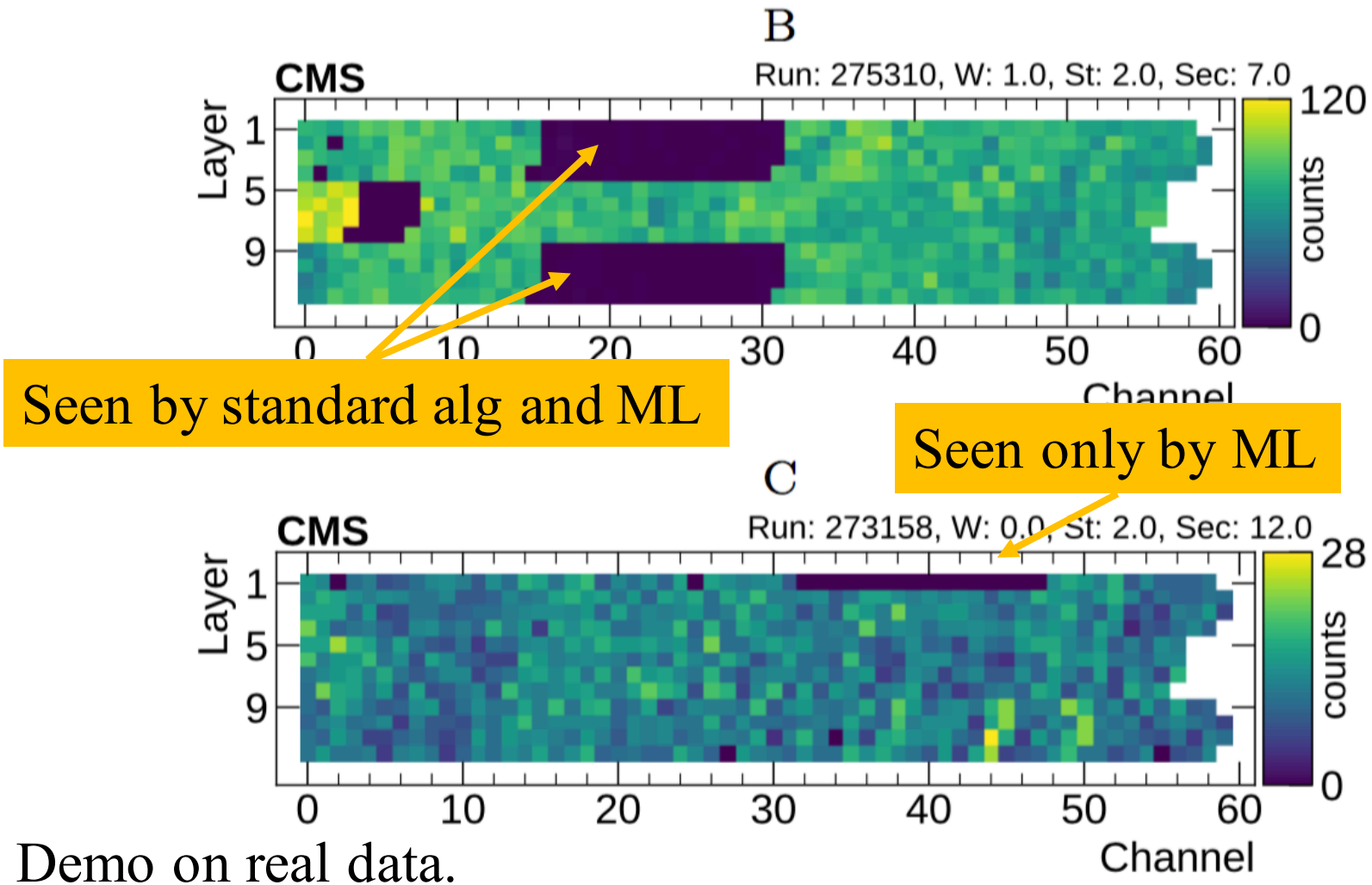


Anomaly detection: supervised

- Suppose you have two independent samples A and B, *supposedly* statistically identical. E.g. A and B could be:
  - MC prod 1, MC prod 2
  - MC generator 1, MC generator 2
  - Geant4 Release 20.X.Y, release 20.X.Z
  - Production at BNL, production at Lyon
  - Data of yesterday, Data of today
- How to verify that A and B are indeed identical ?
- Standard approach : overlay histograms of many carefully chosen variables, check for differences (e.g. KS test)
- One supervised ML approach (not the only one): ~~ask an artificial scientist~~, train your favorite classifier to distinguish A from B, histogram the score, check the difference (e.g. AUC or KS test)
  - ➔only one distribution to check
- Being developped for accelerator monitoring, experiment Data Quality monitoring

Semi-supervised: DQM application

- Example application CMS muon chamber monitoring (with Convolutional NN)



Demo on real data.

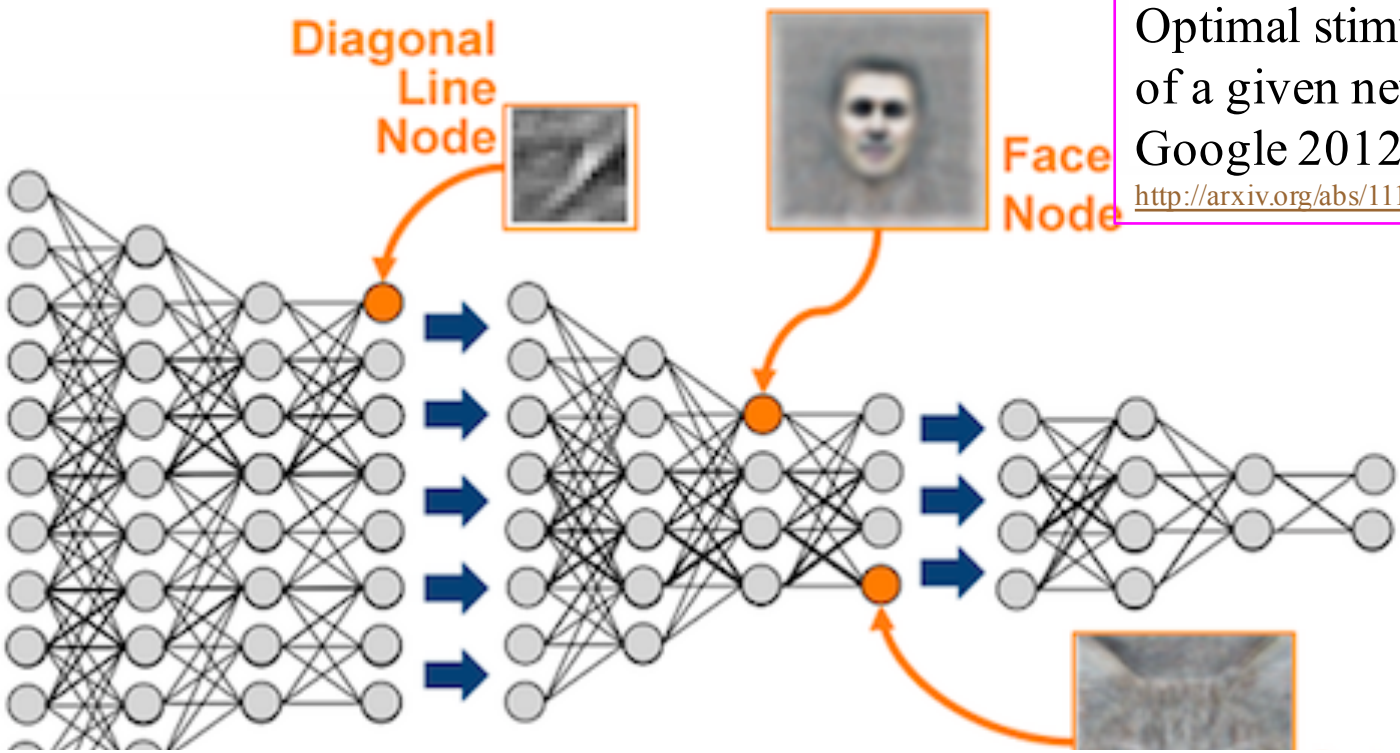
Anomaly detection for physics

Application to new physics

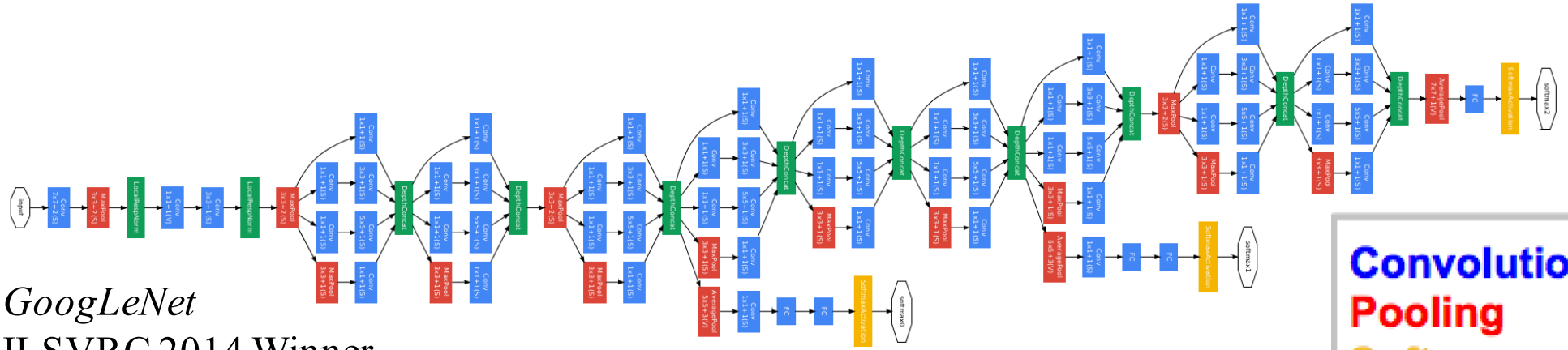
Maurizio Pierini

ML in analysis

Deep learning



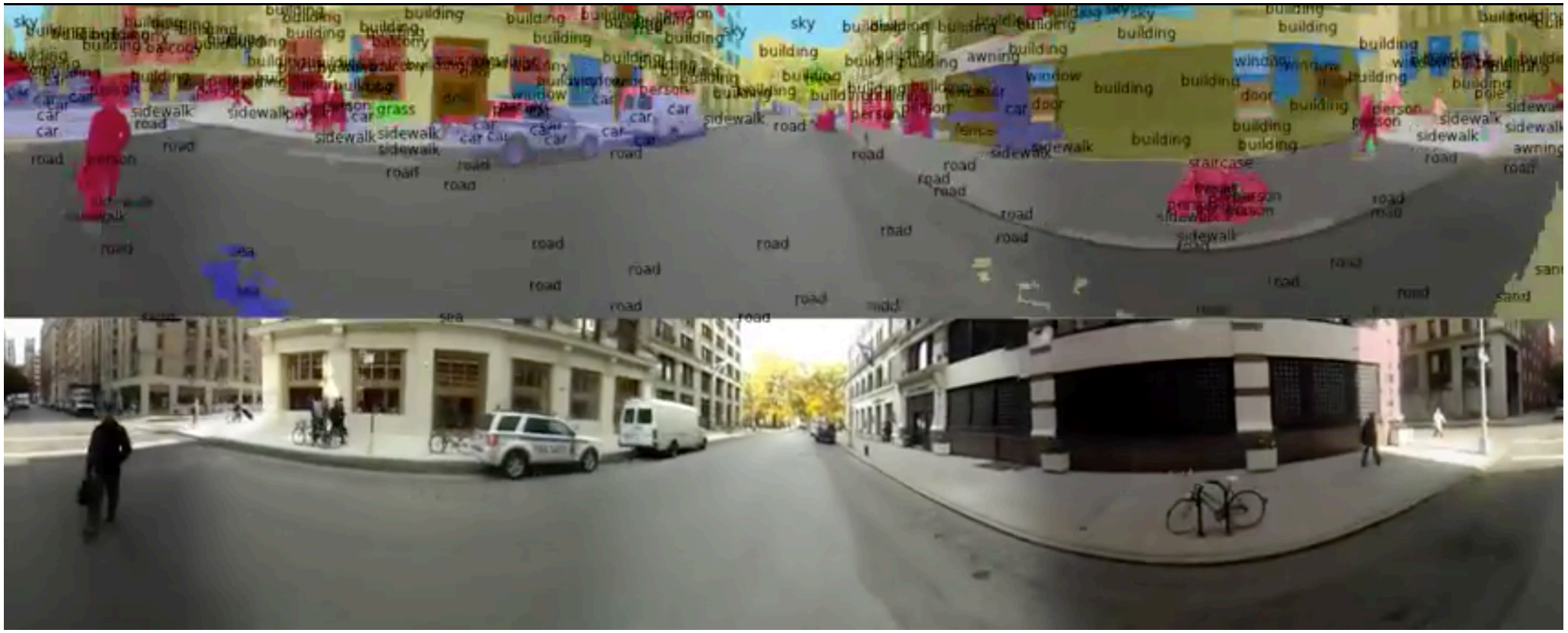
Optimal stimulus
of a given neuron
Google 2012
http://arxiv.org/abs/1112.6209



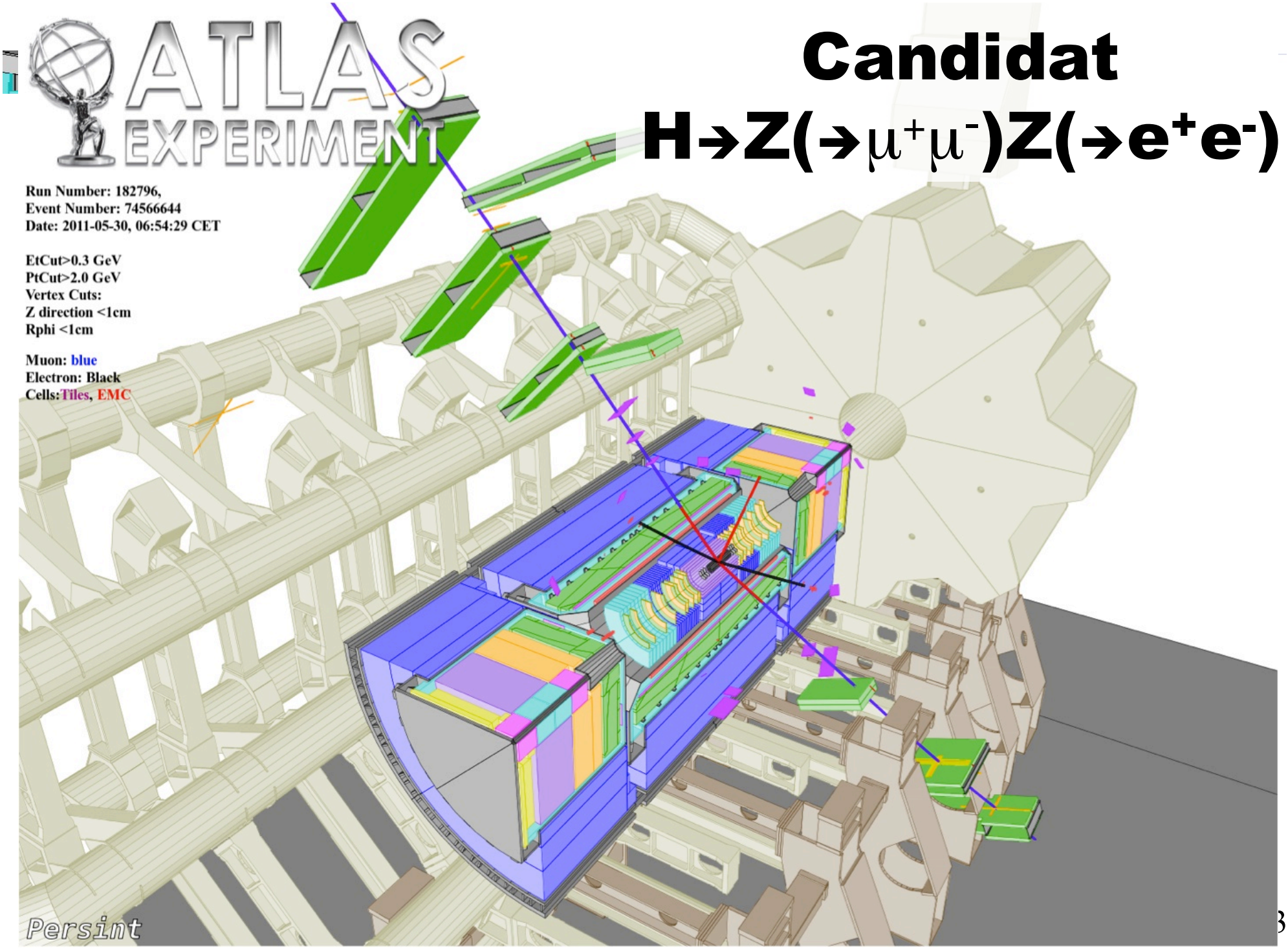
GoogLeNet
ILSVRC 2014 Winner
4M parameters

Convolution
Pooling
Softmax
Other

Typical Deep Learning application

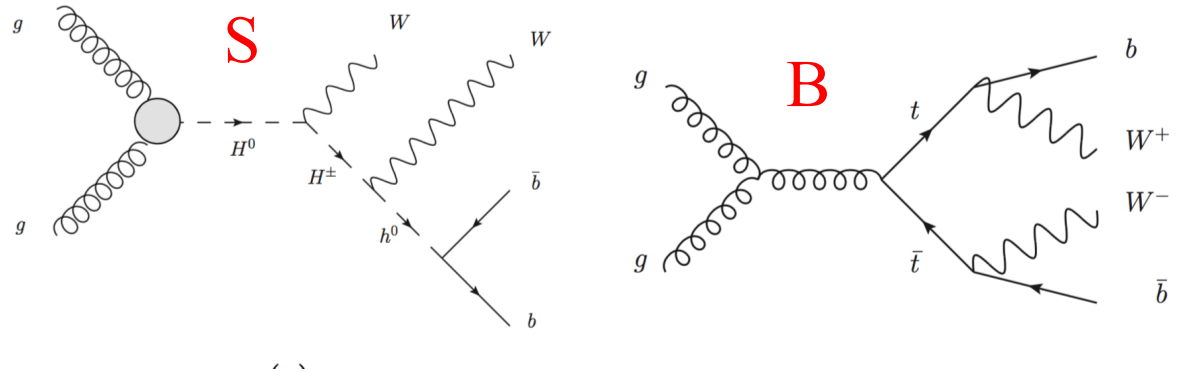


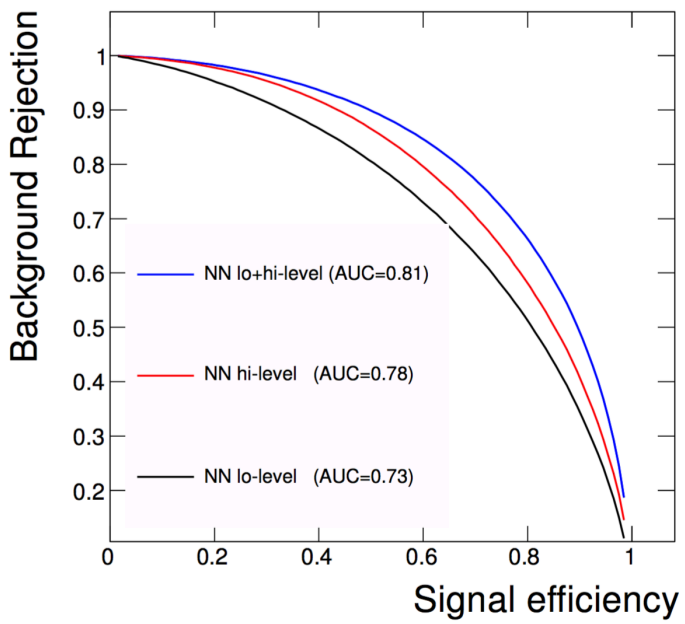
Candidat
$H \rightarrow Z(\rightarrow \mu^+\mu^-)Z(\rightarrow e^+e^-)$



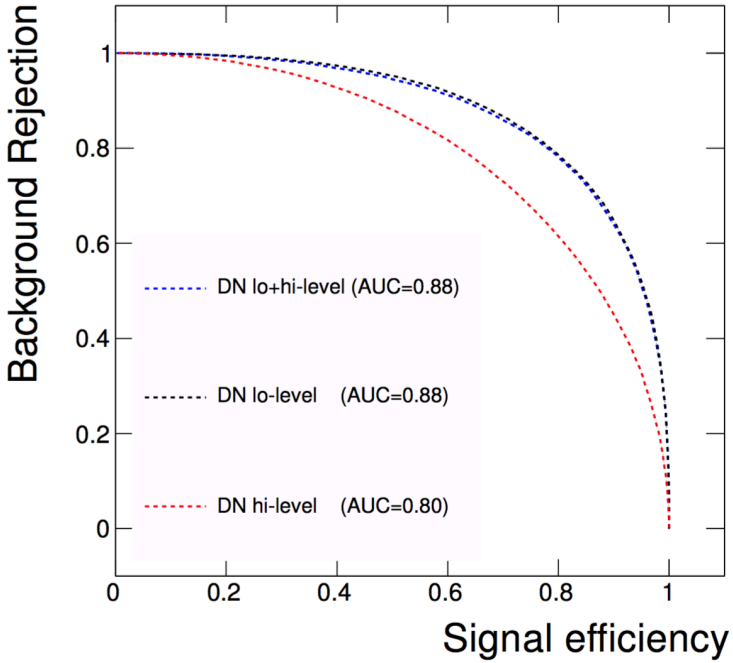
Deep learning for analysis

1402.4735 Baldi, Sadowski, Whiteson





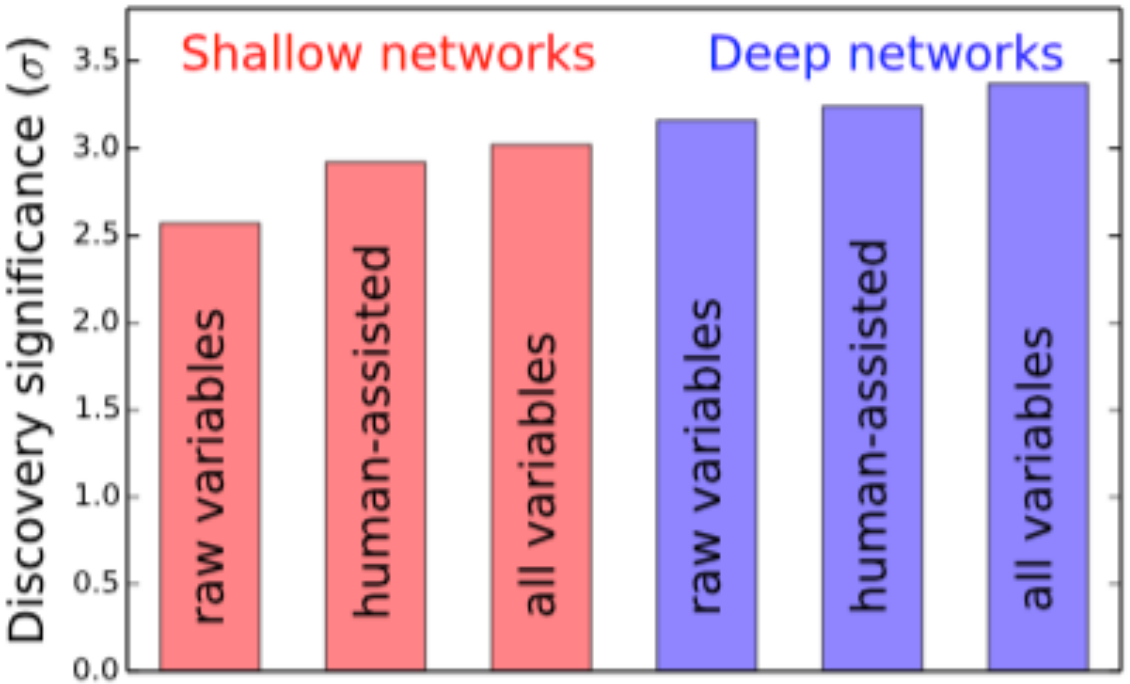
- MSSM at LHC : H^0➔WWbb vs tt➔WWbb
- Low level variables:
  - 3-momentum vectors
- High level variables:
  - Pair-wise invariant masses
- Deep NN outperforms NN, and does not need high level variables
- DNN learns the physics ???



Deep learning for analysis (2)

1410.3469 Baldi Sadowski Whiteson

- H tautau analysis at LHC: H➔tautau vs Z➔tautau
  - Low level variables (4-momenta)
  - High level variables (transverse mass, delta R, centrality, jet variables, etc…)



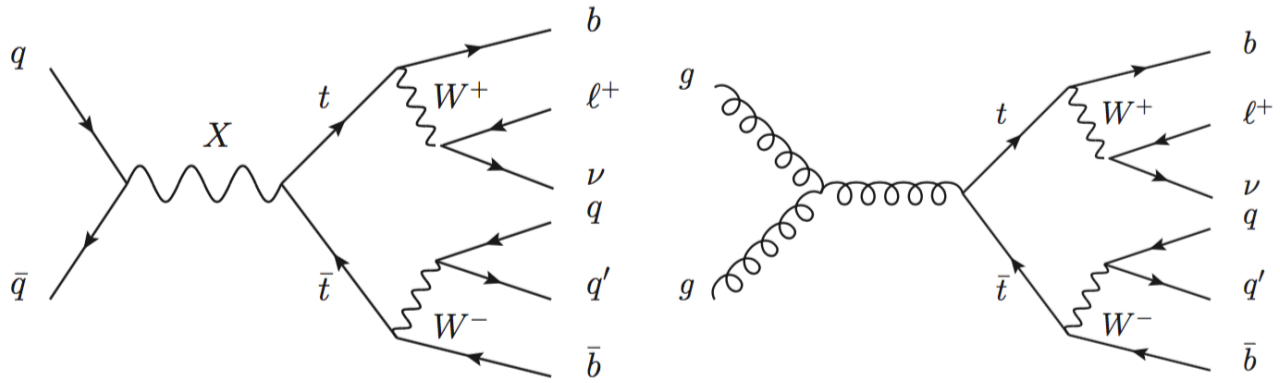
- Here, the DNN improved on NN but still needed high level features
- Both analyses with Delphes fast simulation
- ~100M events used for training (>>100* full G4 simulation in ATLAS)

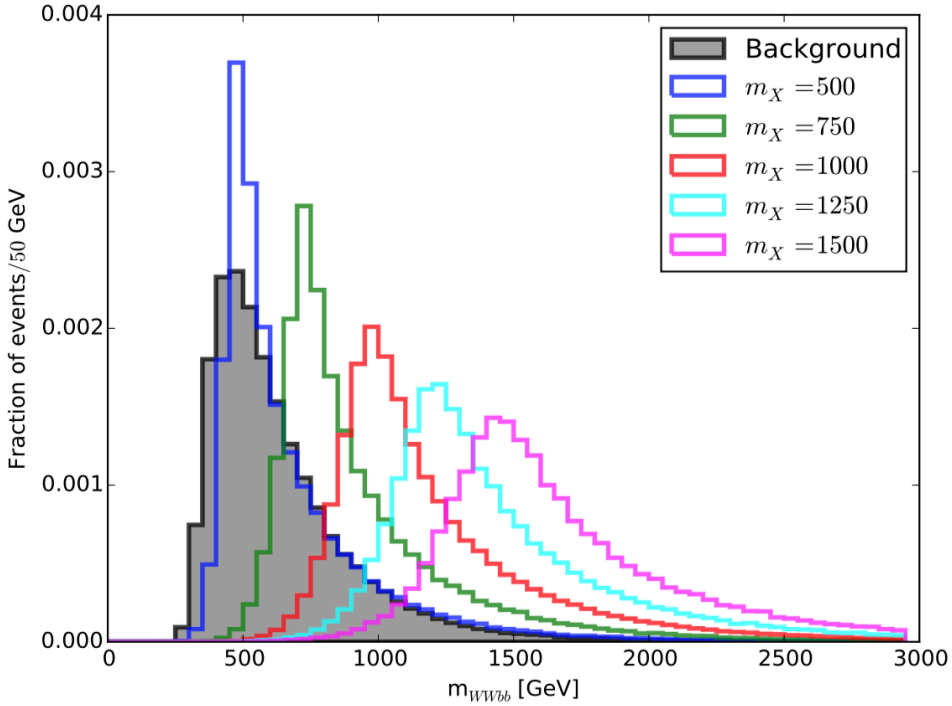
DNN for analysis (3)

- No published LHC analyses using DL (CMS 2018 ttH « DNN » just two layers)
- Recent trend is to feed more (up to 20) variables to classifiers, even low level ones (2/3-vectors of particles) (see recent ATLAS/CMS ttH papers)
- A few NN in top and Higgs physics but no clear advantage wrt BDT
- Not completely clear why: most likely hypothesis : lack of training MC (Baldi et al papers use >10E6 events, shile a typical LHC analysis has at most 100K, even less, after all preselection)
- ➔DNN, not a drop-in replacement/improvement on BDT
- However some promising successes : ➔

Parameterised learning

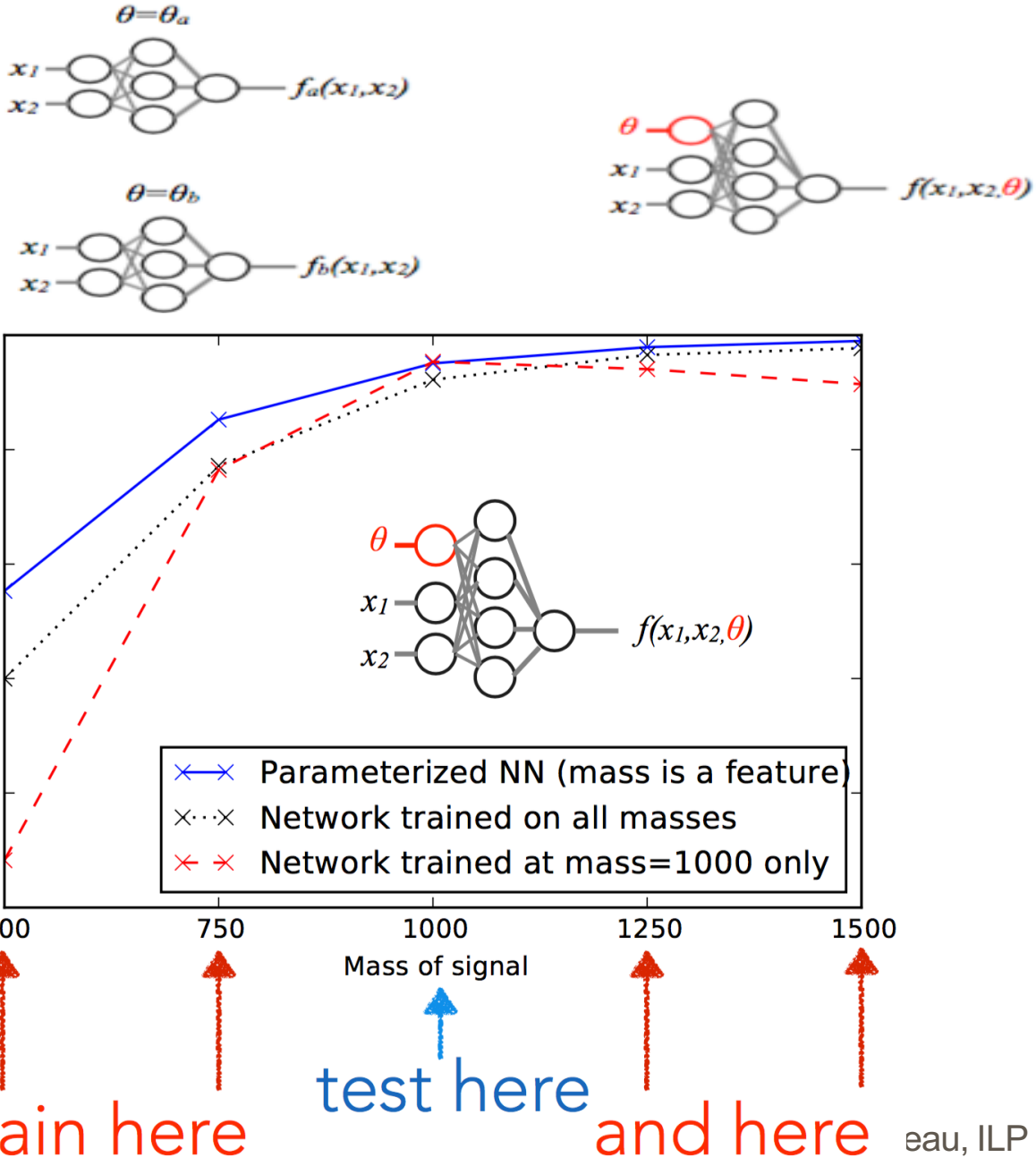
1601.07913 Baldi, Cranmer, Faucett, Sadowksi, Whiteson





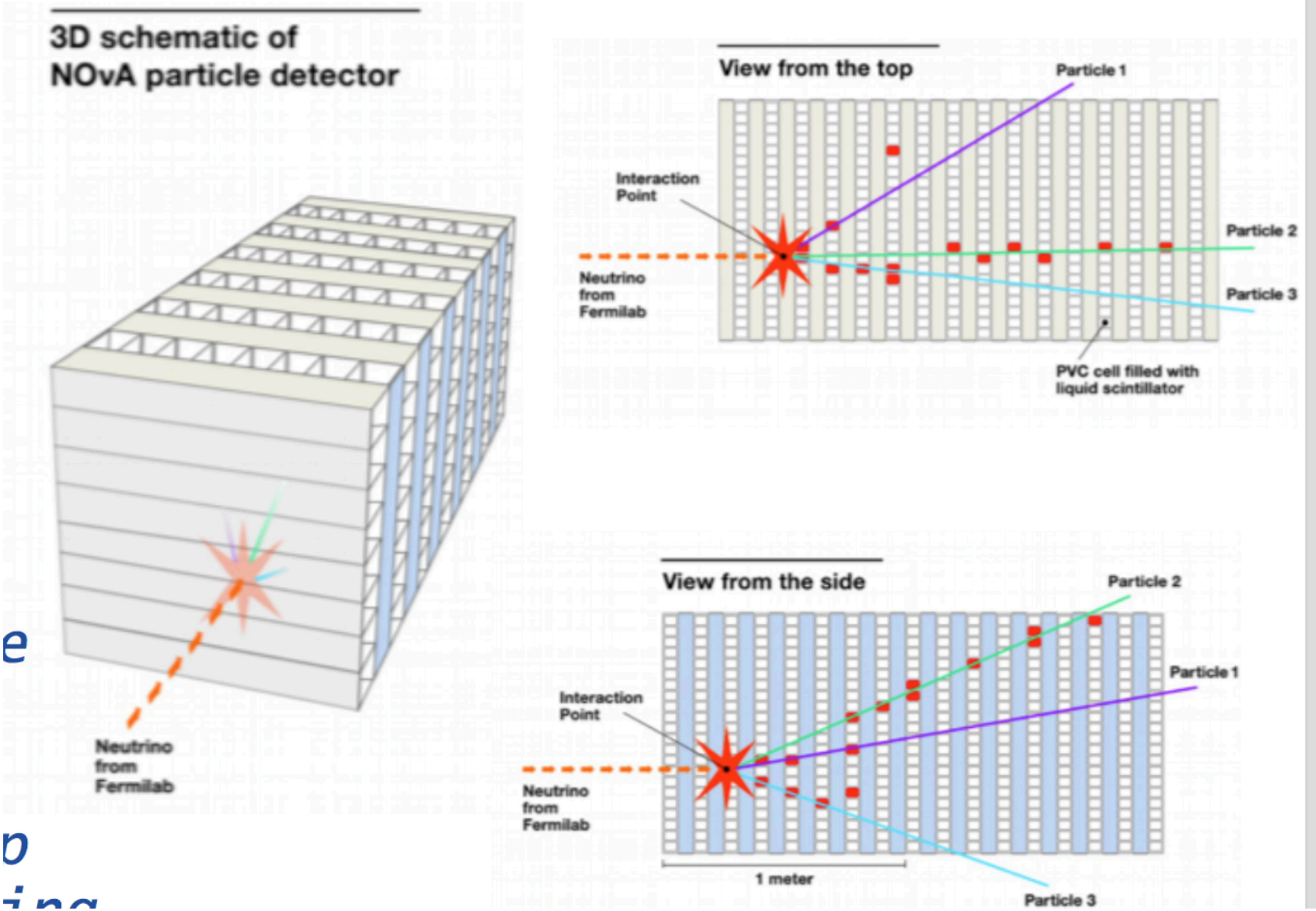
- Typical case: looking for a particle of unknown mass
- E.g. here tt decay

Parameterised learning (2)



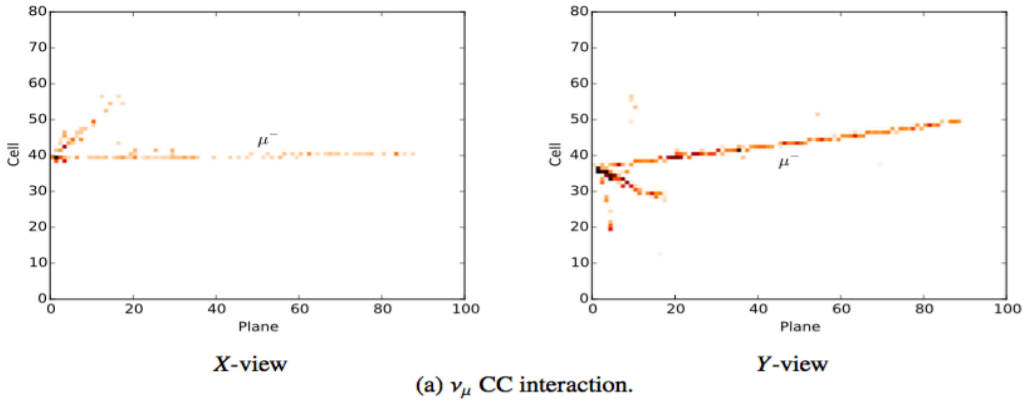
- Train on 28 features plus true mass
- Parameterised NN as good as single mass training
- ➔clean interpolation
- (mass just an example)
- Very recently used by CMS bblνl ν search https://arxiv.org/pdf/1708.04188.pdf

Deep Learning success : NOVA

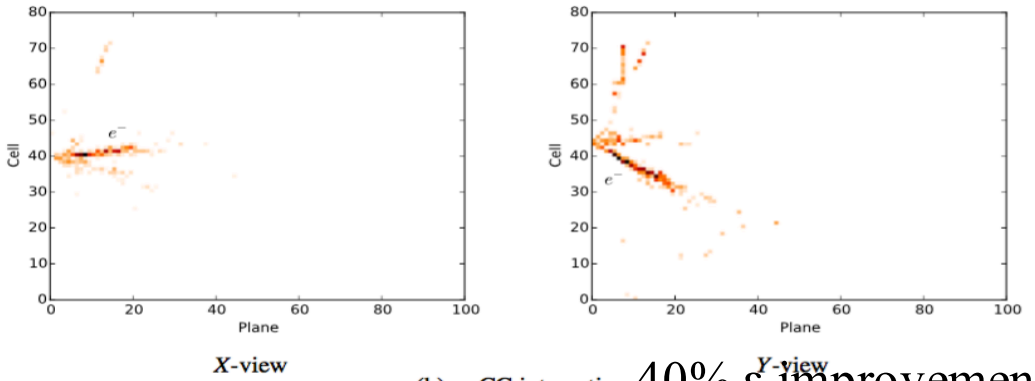


Nova (2)

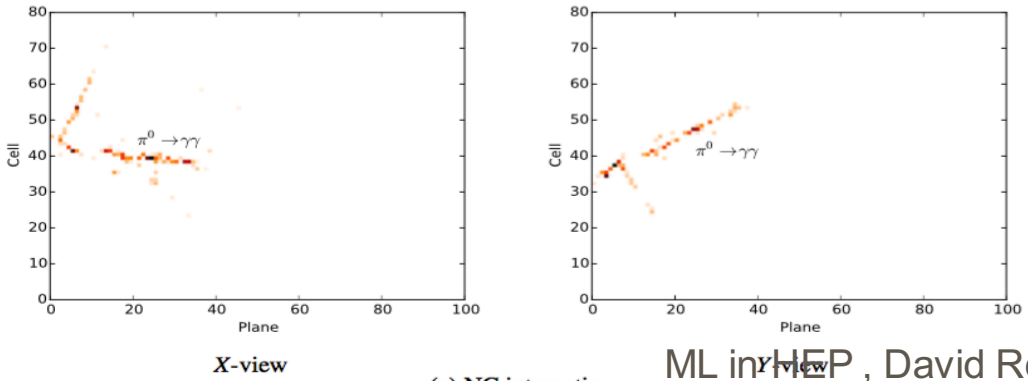
arXiv 1604.01444 Aurisano et al



(a) ν_μ CC interaction.



(b) ν_e CC interaction.

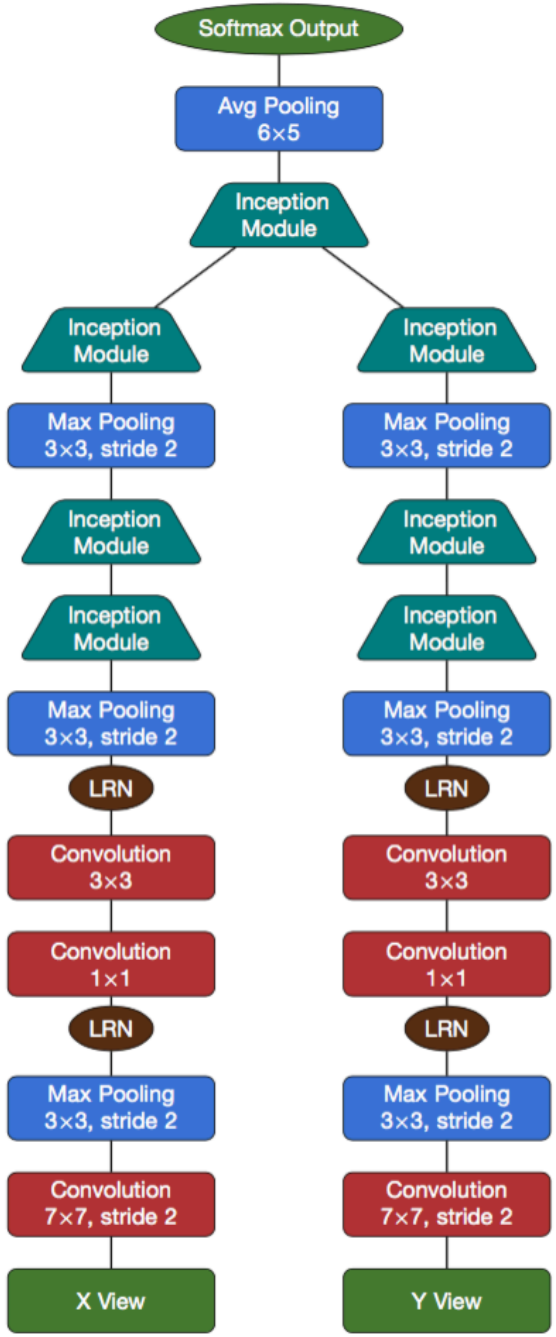


(c) NC interaction.

40% ε improvement

Neutrino interaction classification
Using Convolutionnal Neural Network (GoogleNet)

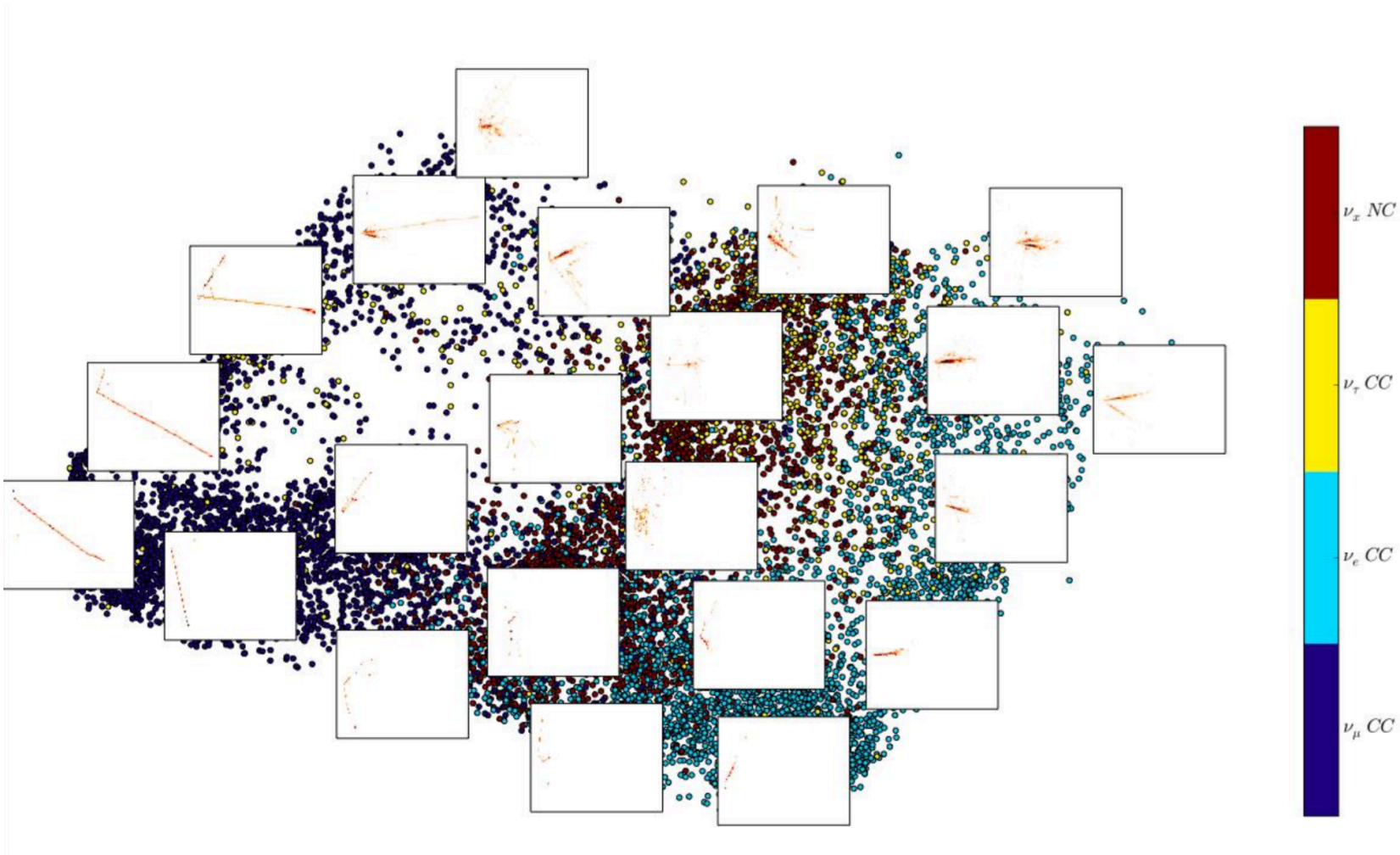
Actually used in physics results 1703.03328 and 1706.04592



Aparté on t-SNE

van der Maaten and Hinton. JMLR 9 2008

- Non-linear dimensionality compression, very popular in ML, unknown (almost) in HEP, except NOVA:



Systematics-aware training

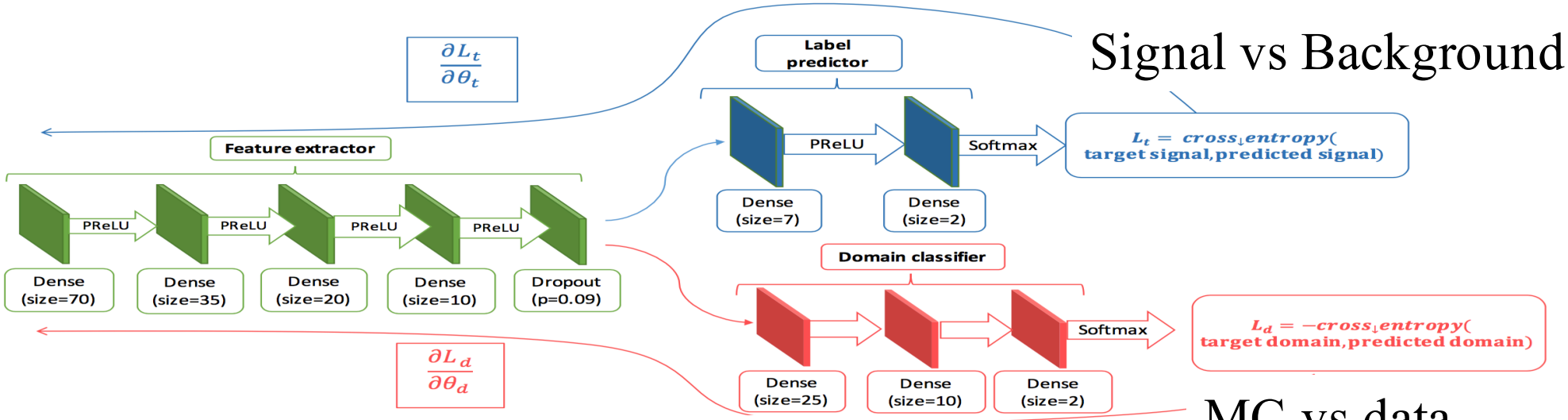
See Victor Estrade CHEP 2018

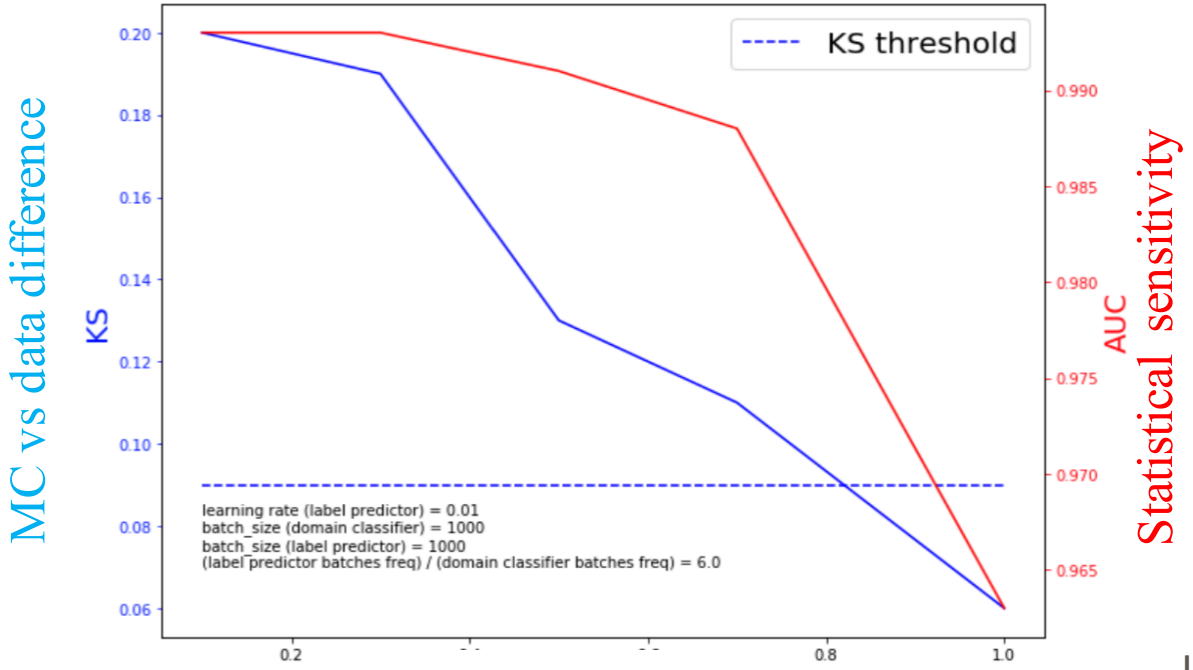
- Our experimental measurement papers typically ends with
  - measurement = m ± σ(stat) ± σ(syst)
  - σ(syst) systematic uncertainty : known unknowns, unknown unknowns…
- Name of the game is to minimize quadratic sum of :

  σ(stat) ±σ(syst)
- ML techniques used so far to minimise σ(stat)
- Impact of ML on σ(syst) or even better global optimisation of σ(stat) ± σ(syst) is an open problem
- Worrying about σ(syst) untypical of ML in industry (… until recently *fake news*)
- However, a hot topic in ML in industry: *transfer learning*
- E.g. : train image labelling on a image dataset, apply on new images (different luminosity, focus, angle etc…)
- For HEP : we train with Signal and Background which are not the real one (MC, control regions, etc...)➔source of systematics

Syst Aware Training: adversarial

Inspired from 1505.07818 Ganin et al : ACAT 2017 Ryzhikov and Ustyuzhanin





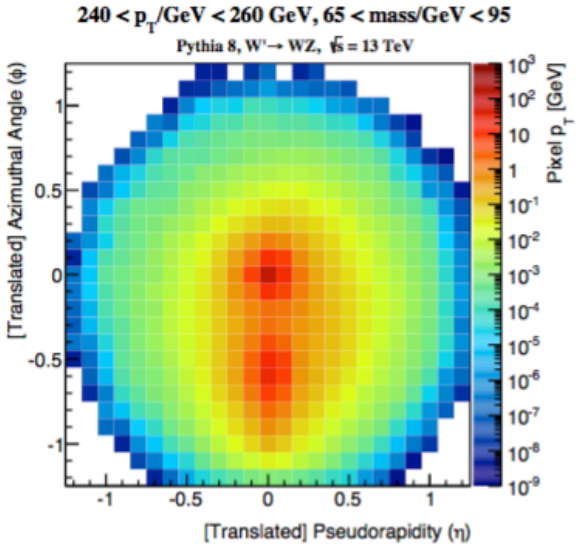
ML in reconstruction

Jet Images

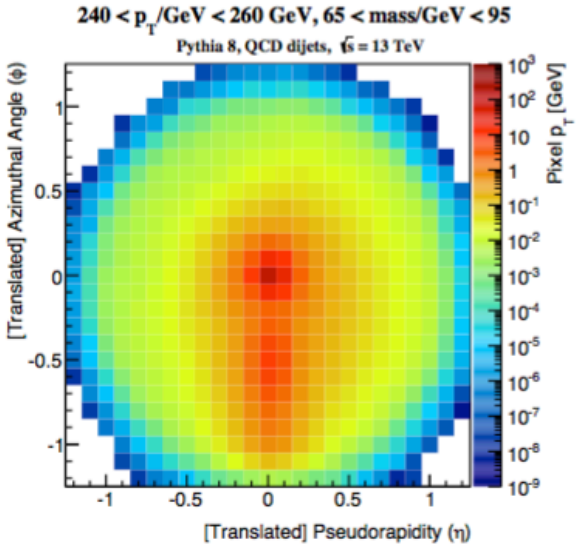
arXiv 1511.05190 de Oliveira, Kagan, Mackey, Nachman, Schwartzman

- Distinguish boosted W jets from QCD
- Particle level simulation
- Average images:

Boosted W➔qq jet

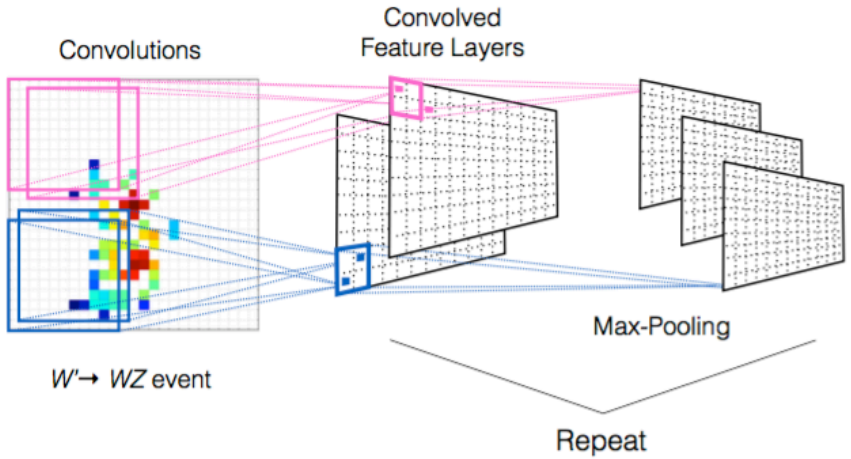


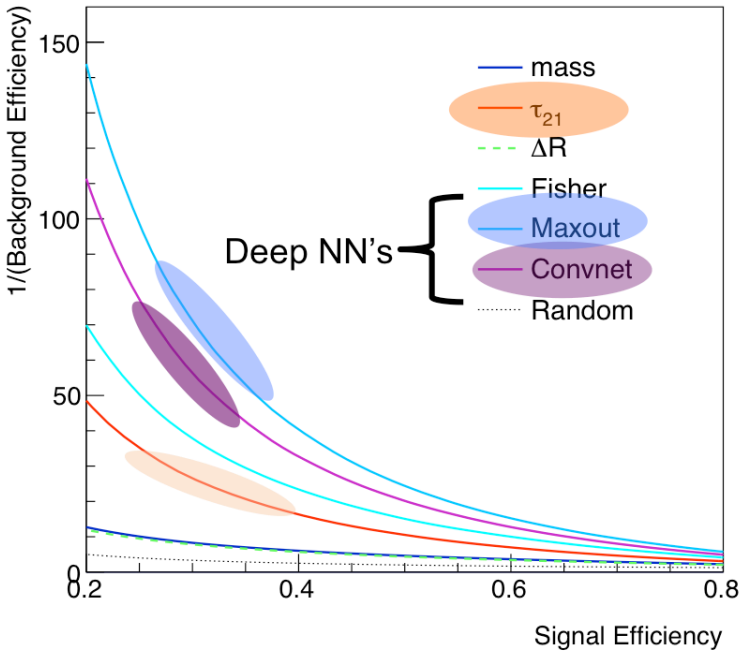
QCD



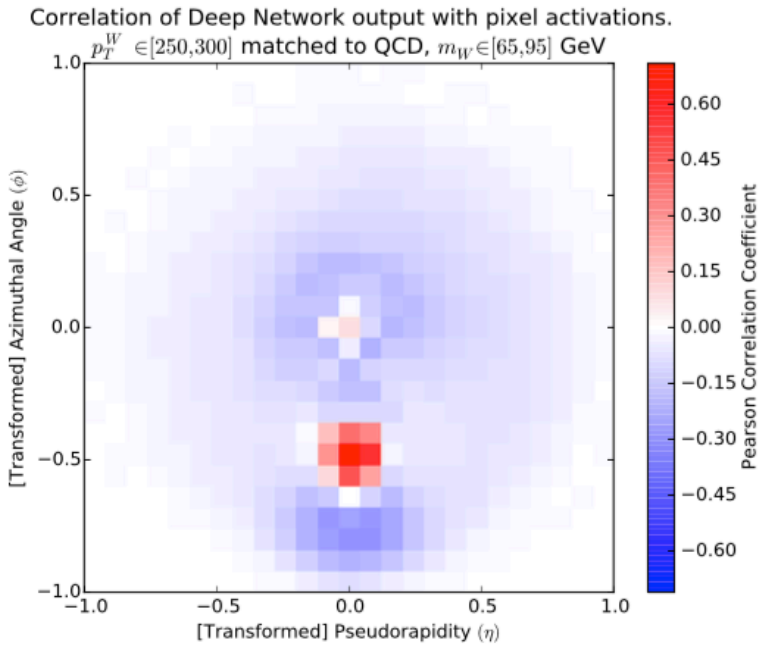
Jet Images : Convolution NN

arXiv:1511.05190

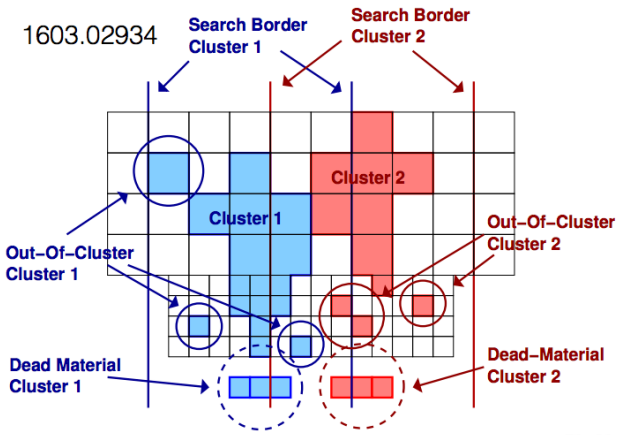




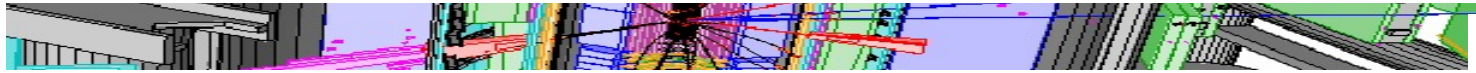
- Variables build from CNN outperform the more usual ones



- What the CNN sees (the "cat" neurone")
- Now need proper detector and pileup simulation ATL-PHYS-PUB-2017-017
- ➔3Dimension ?

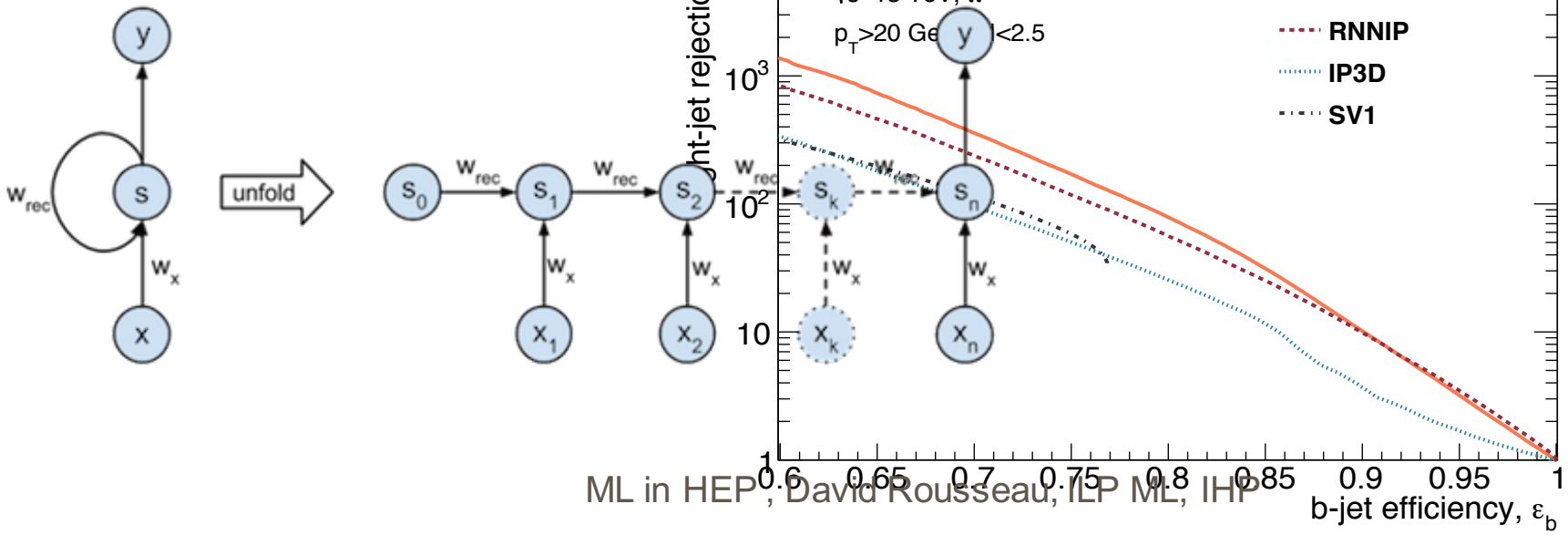


RNN for b tagging



ATL-PHYS-**PUB**-2017-003

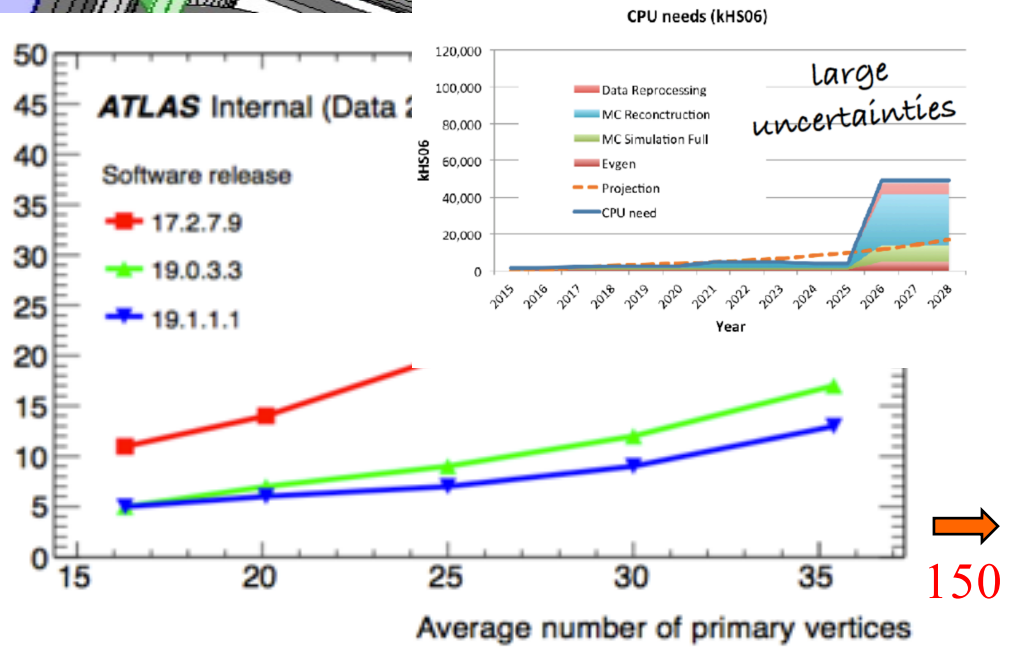
- BDT and usual NN expect a fix number of input. What to do when the number of inputs is not fixed like the tracks for b-quark jet tagging ?
- Recurrent Neural Networks (RNN) have seen outstanding performance for processing sequence data
  - Take data at several "time-steps", and use previous time-step information in processing next time-steps data
- For b-tagging, take list of tracks in jet and feed into RNN
  - Basic track information like d0, z0, pt-Fraction of jet, …
  - Physics inspired ordering by d0-significance
- RNN outperforms other IP algorithms
  - No explicit vertexing, still excellent performance
  - First combinations with other algorithms in progress
- Learning on sequence data may be important in other places!
  - Combining tracks with clusters? Tr

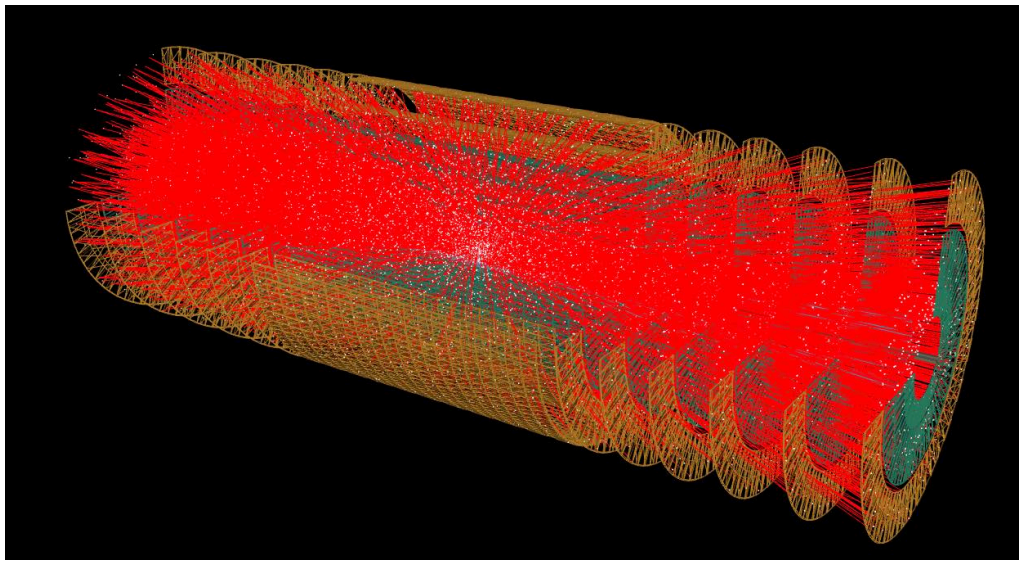


TrackML tracking challenge

Tracking competition

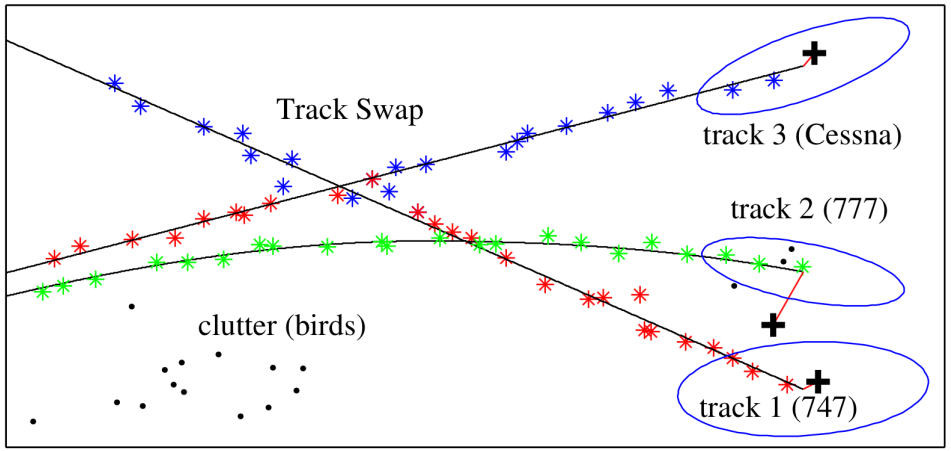
- Tracking (in particular pattern recognition) dominates reconstruction CPU time at LHC
- HL-LHC (phase 2) perspective : increased pileup :Run 1 (2012): <>~20, Run 2 (2015): <>~30,Phase 2 (2025): <>~150
- CPU time quadratic/exponential extrapolation (difficult to quote any number)
- Large effort within HEP to optimise software and tackle micro and macro parallelism. Sufficient gains for Run 2 but still a long way for HL-LHC.
- >20 years of LHC tracking development. Everything has been tried?
  - Maybe yes, but maybe algorithm slower at low lumi but with a better scaling have been dismissed ?
  - Maybe no, brand new ideas from ML (i.e. Convolutional NN)
- ➔Tracking challenge launched May-Aug 2018 on Kaggle : just accuracy
- ➔Throughput phase launched on Codalab : Sep-Mar 2019 : accuracy AND speed
- 125 events x (10'000 tracks / 100'000 points)
- Follow us on twitter @trackmllhc !
- Details on : https://sites.google.com/site/trackmlparticle/

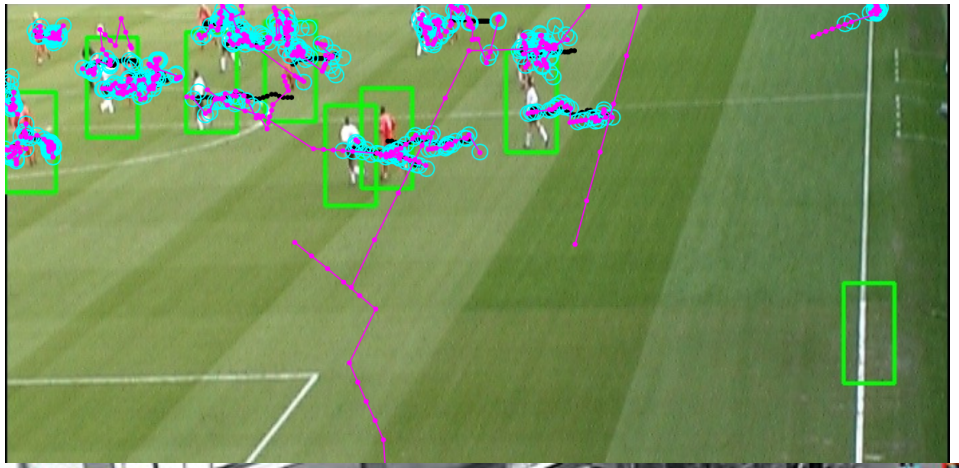


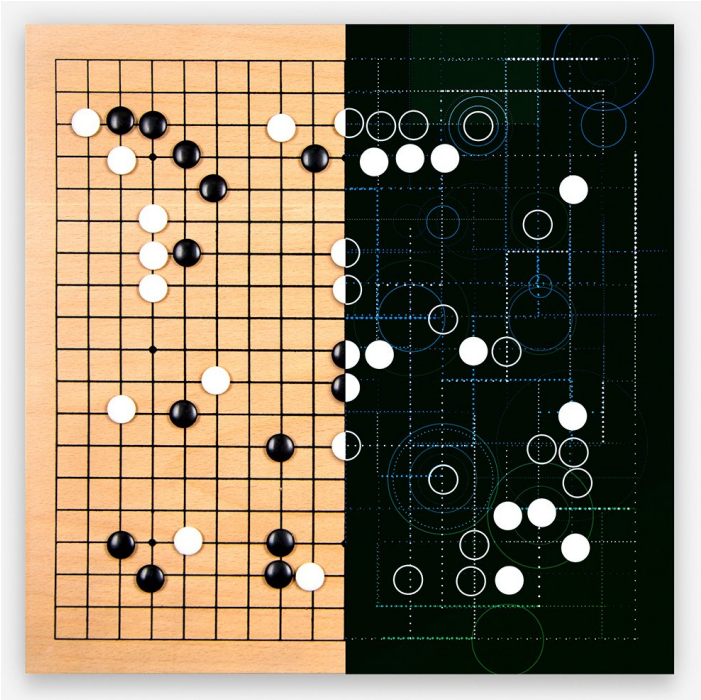


Pattern Recognition/Tracking

- Pattern recognition/tracking is a very old, very hot topic in Artificial Intelligence, but very varied
- Note that these are real-time applications, with CPU constraints







Aparté on ML in HEP history

Computer Physics Communications 49 (1988) 429–448
North-Holland, Amsterdam

NEURAL NETWORKS AND CELLULAR AUTOMATA IN EXPERIMENTAL HIGH ENERGY PHYSICS

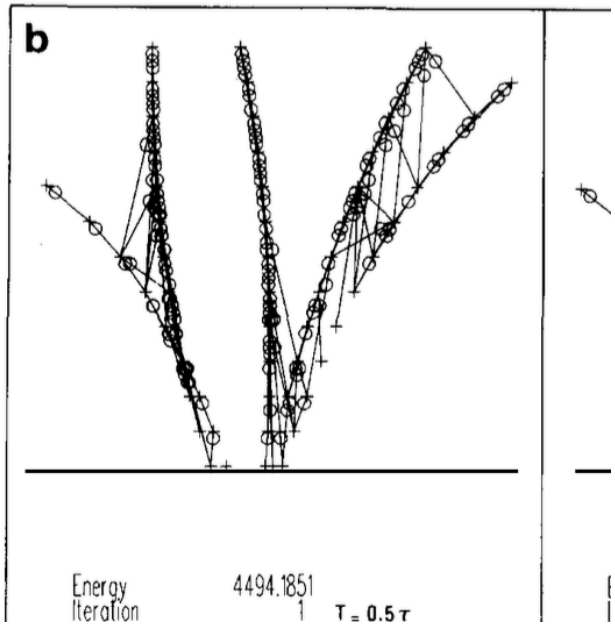
B. DENBY

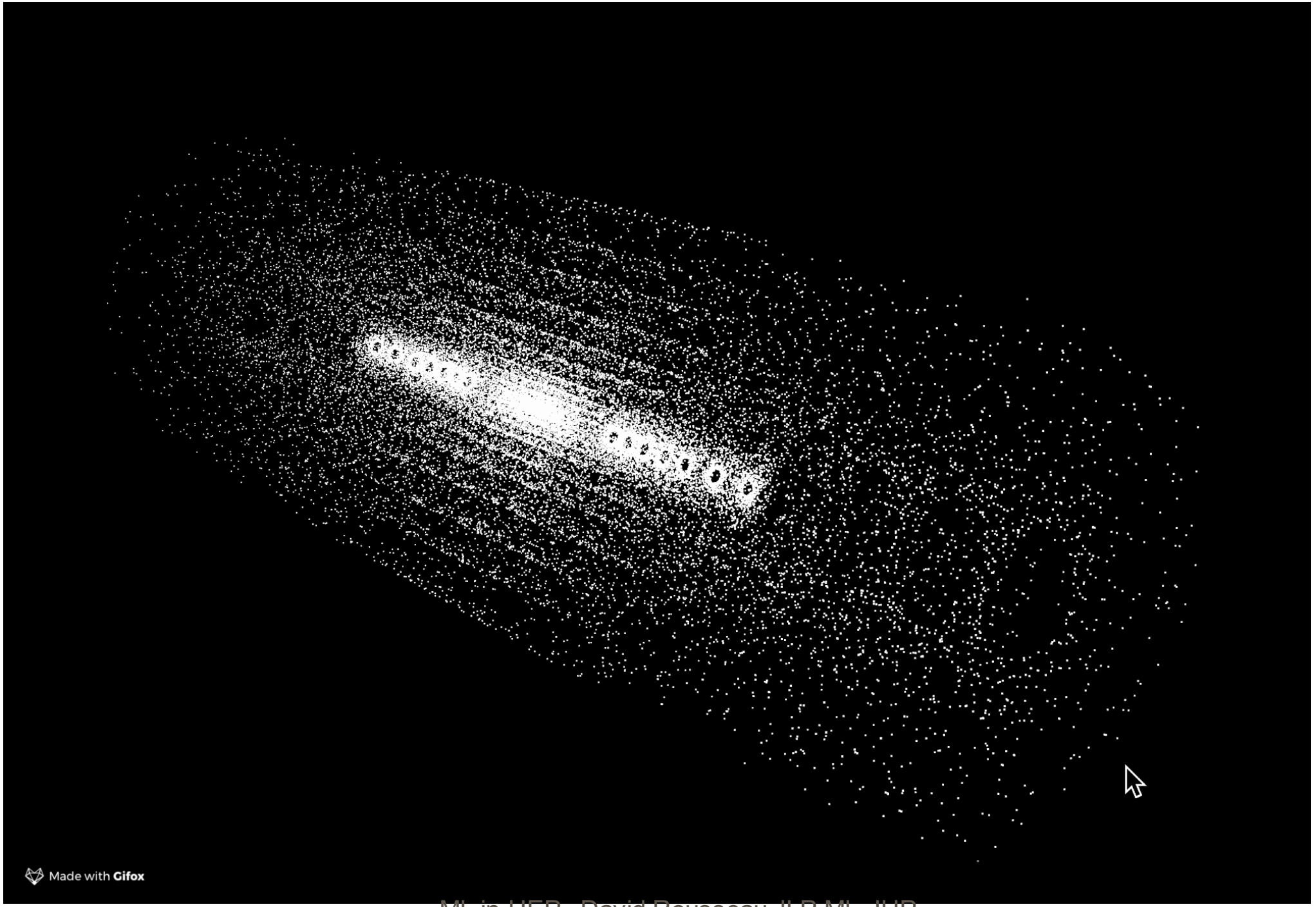
Laboratoire de l'Accélérateur Linéaire, Orsay, France

Received 20 September 1987; in revised form 28 December 1987

Bruce Denby

- 1987 Very first Neural Net in HEP paper known
- NN for tracking and calo clustering
- B. Denby then moved from Delphi at LEP to CDF at Tevatron. He still active outside HEP: 2017 analysis of ultrasonic image of the tongue
- 1992 JetNet Carsten Peterson, Thorsteinn Rognvaldsson (Lund U.) , Leif Lonnblad (CERN) (~500 citations) really started NN use in HEP




Made with Gifox

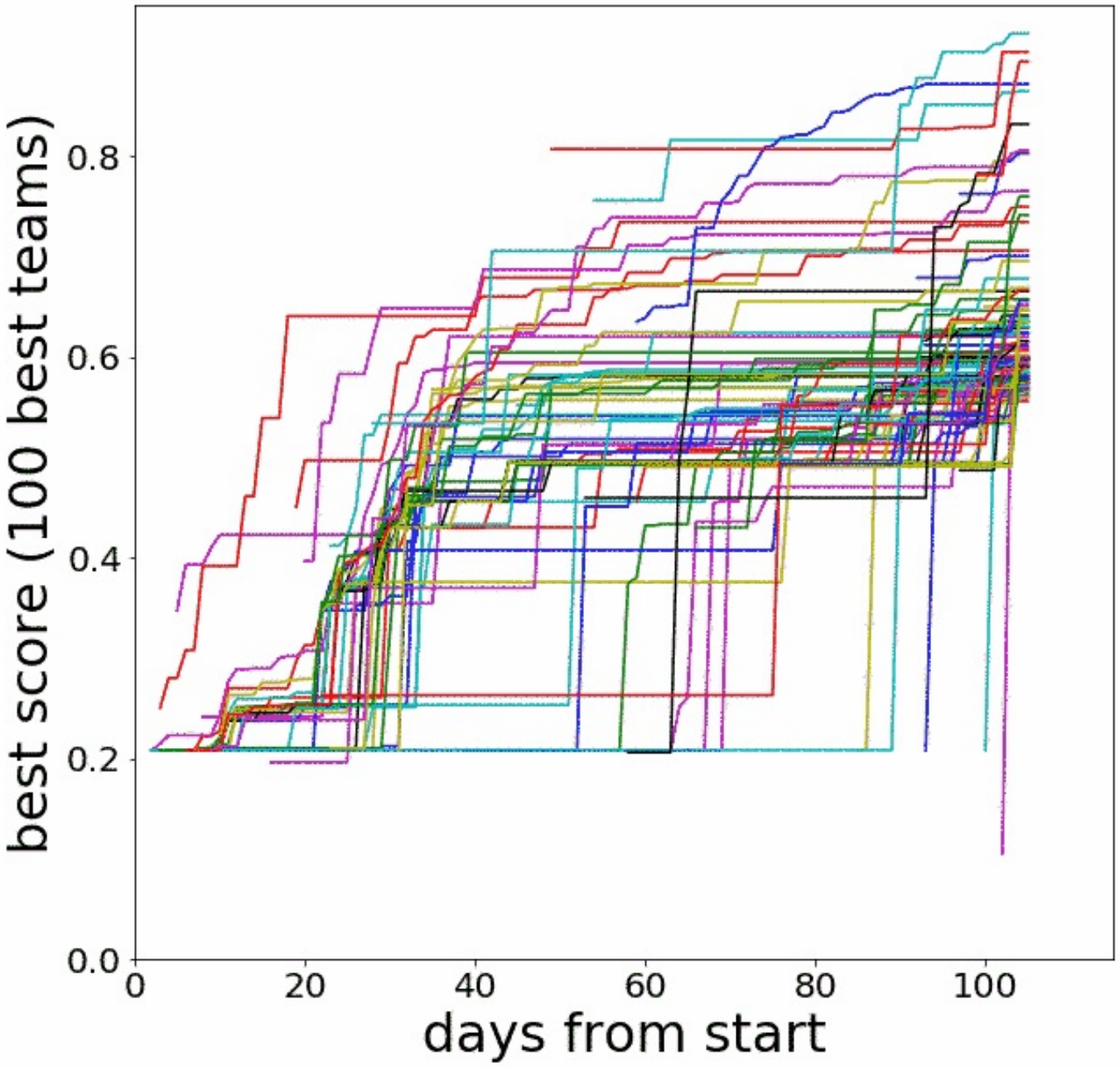
Real life vs challenge

1. Wide type of physics events
2. Full detailed Geant 4 / data
3. Detailed dead matter description
4. Complex geometry (tilted modules, double layers, misalignments…)
5. Hit merging
6. Allow shared hits
7. Output is hit clustering, track parameter and covariance matrix
8. Multiple metrics (see TDR's)

1. One event type (ttbar)
2. ACTS (MS, energy loss, hadronic interaction, solenoidal magnetic field, inefficiency)
3. Cylinders and slabs
4. Simple, ideal, geometry (cylinders and disks)
5. No hit merging
6. Disallow shared hits
7. Output is hit clustering
8. Single number metrics

Simpler, but not too simple!

Evolution of leaderboard

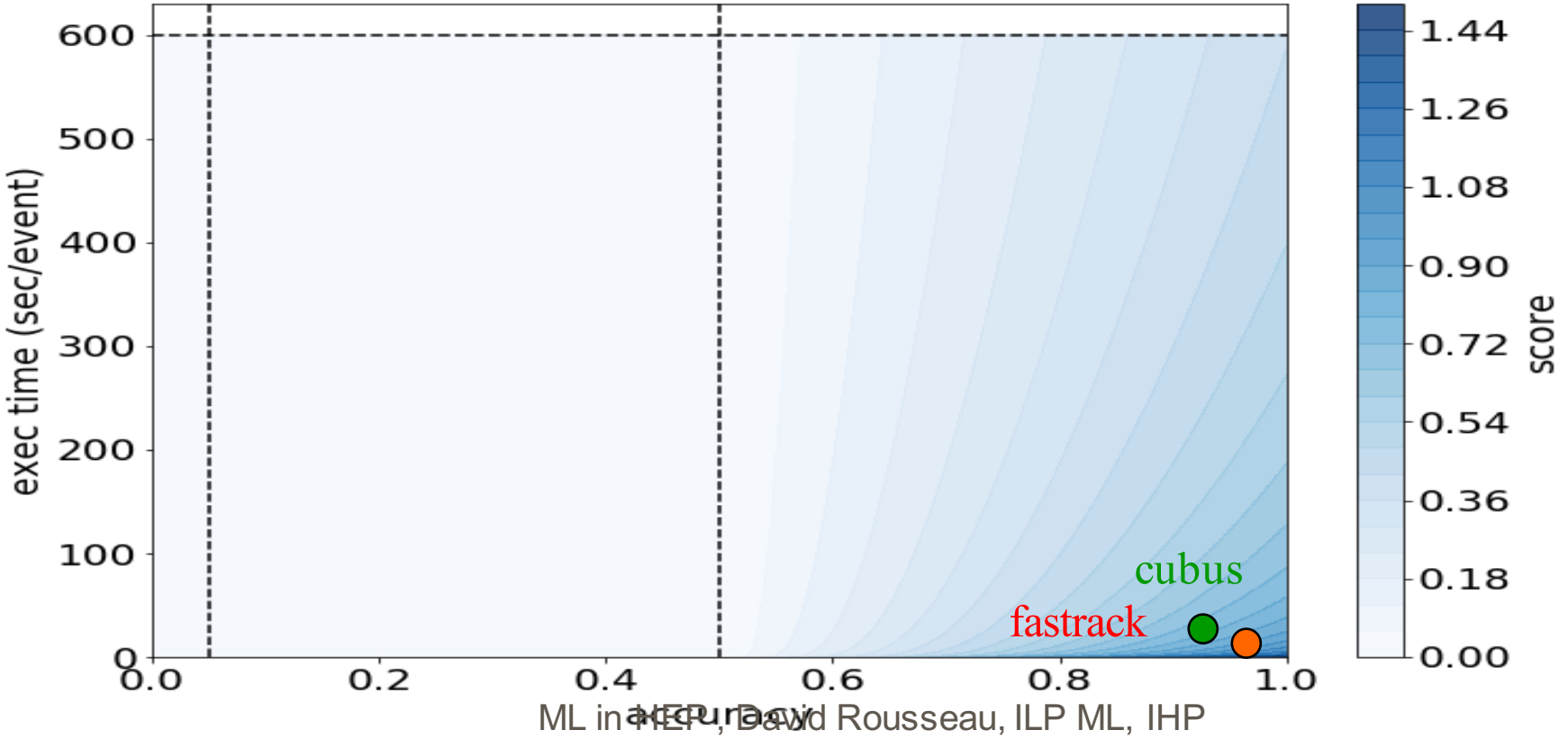


A few competitors

- icecube #1 92.2 % (norvegian CS master student) : combinatorial approach, with a bit of ML
- outrunner #2 90.3% (taïwanese software engineer) Deep Learning approach
  - Very innovative!
  - But brute force : takes one full day per event !
  - However code is using naïve python nested loops
- Sergey Gorbunov #3 89.4% demelian #4 87.1% : (HEP tracking trigger experts in HEP labs) parameterised local helix fitting
- Yuval & Trian #7 80.4% : (greek and israeli computing engineer) innovative clustering
- CPMP #9 80.1% : (french computing engineer) DBSCAN unsupervised clustering algorithm
  - we gave DBSCAN in starting kit, with a 20% score, because in only required a few lines
- Nicole and Liam Finnies #12 74.8% : (german data scientists) use LSTM

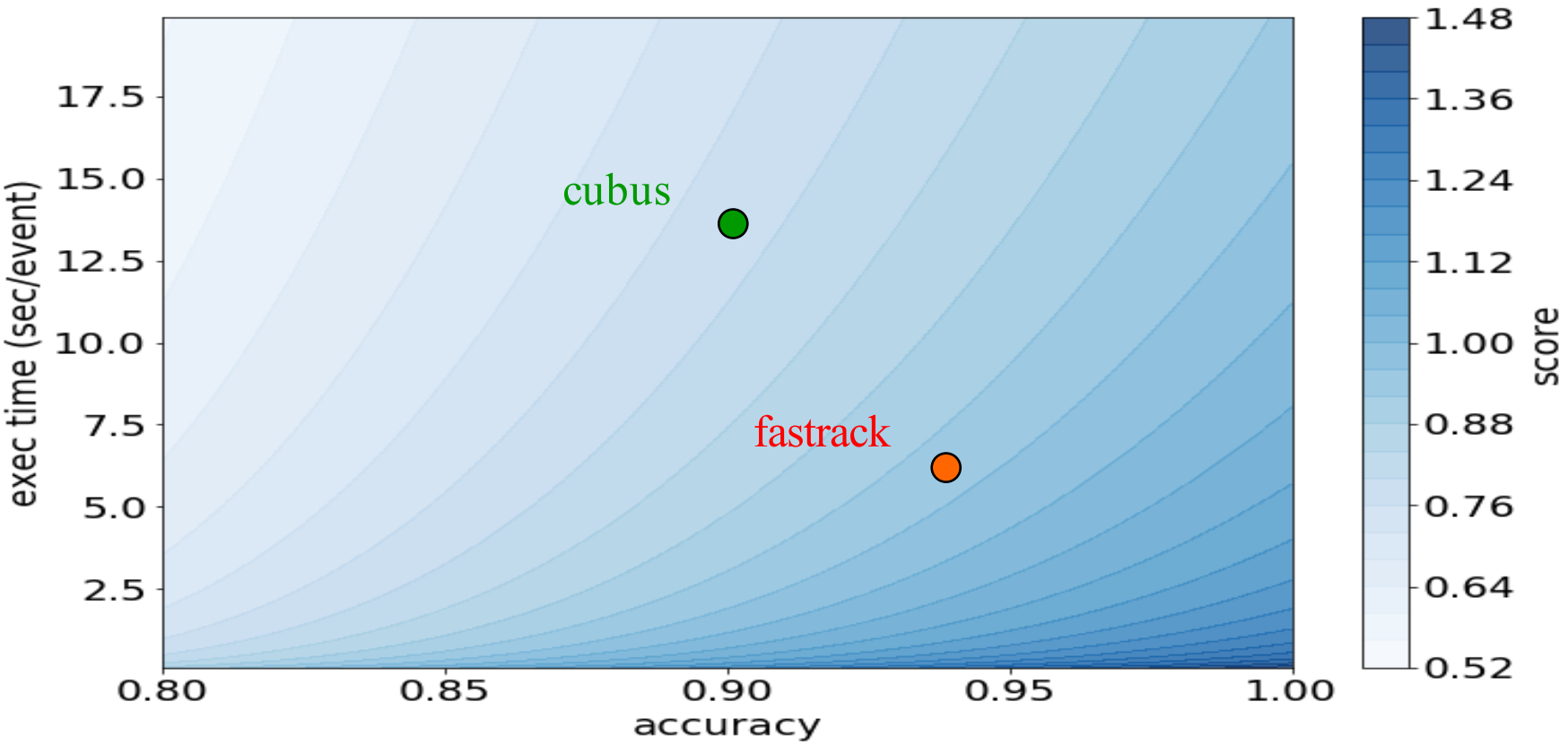
Throughput scoring

- *Ranking score* = $\sqrt{log}(1 + 600/time) * (accuracy - 0.5)^2$
- Documented software of first phase #1 #2 #3 #7 #9 #11 #12 released
  - Can be used as starting point but need retuning



Throughput phase LB

- By beg Nov, 100 registered, but only 2 with non zero scores
- =>disappointing participation, many hypotheses why
- ➔reschedule to end 12th March 2019
- On the other hand fastrack results are astonishing
  - ATLAS code recently sped up from 250s to 10s … however this is for track pT>900 MeV ~15% of TrackML tracks



End to end Learning

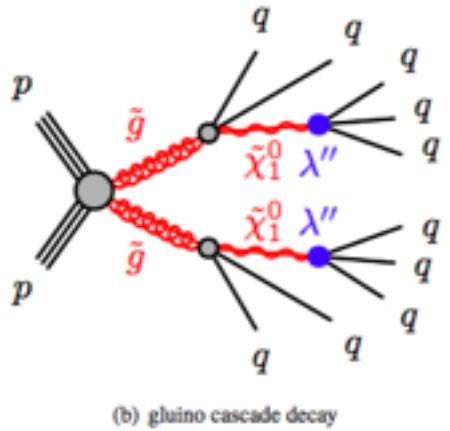
End to end learning

Bhimji et al, 1711.03573

- Train directly for signal on « raw » event ?
- Start from RPV Susy search

ATLAS-CONF-2016-057

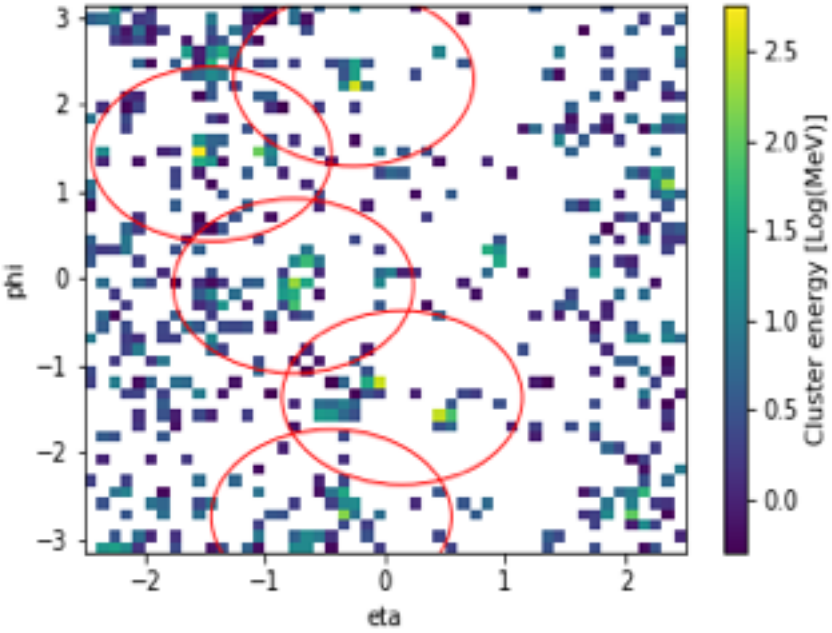
- Fast Simulated events with Delphes



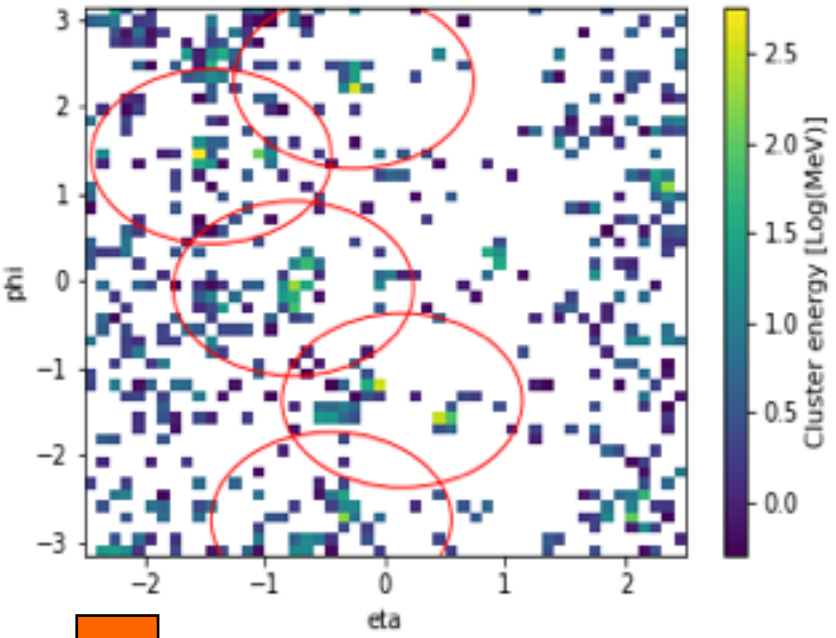
(b) gluino cascade decay

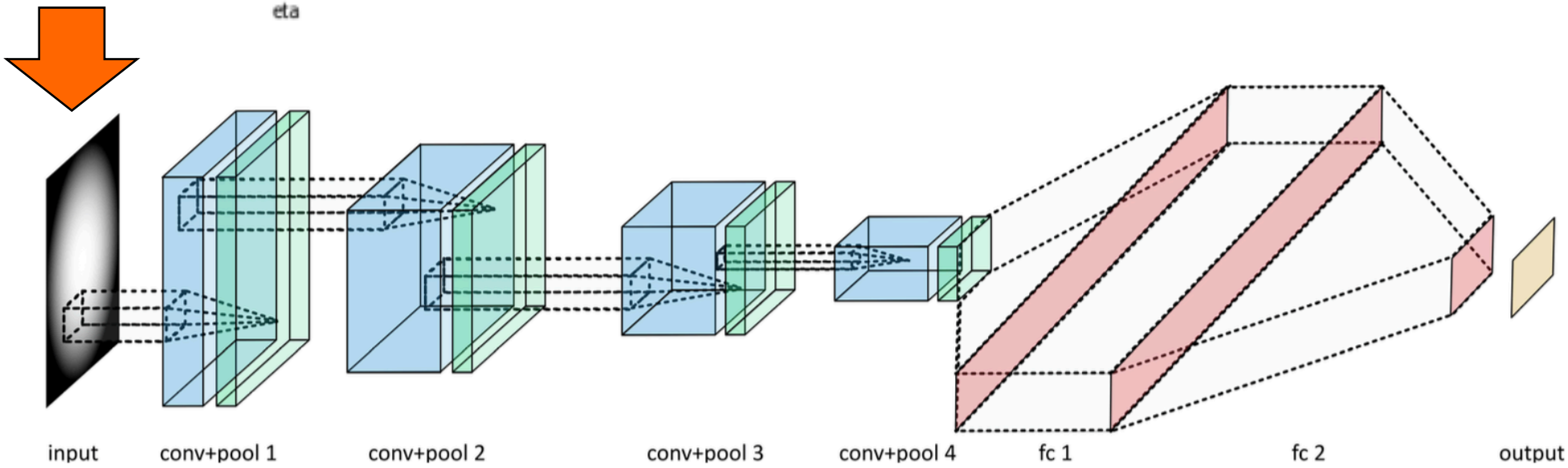
- Project energies on 64x64 ηxϕ grid
- Compare with usual jet Reconstruction and physics Analysis variables such as:

$$M_{\mathrm{J}}^{\Sigma} = \sum_{\substack{p_{\mathrm{T}}>200\,\mathrm{GeV} \\ |\eta|\leq 2.0}}^{4} m^{\mathrm{jet}}$$

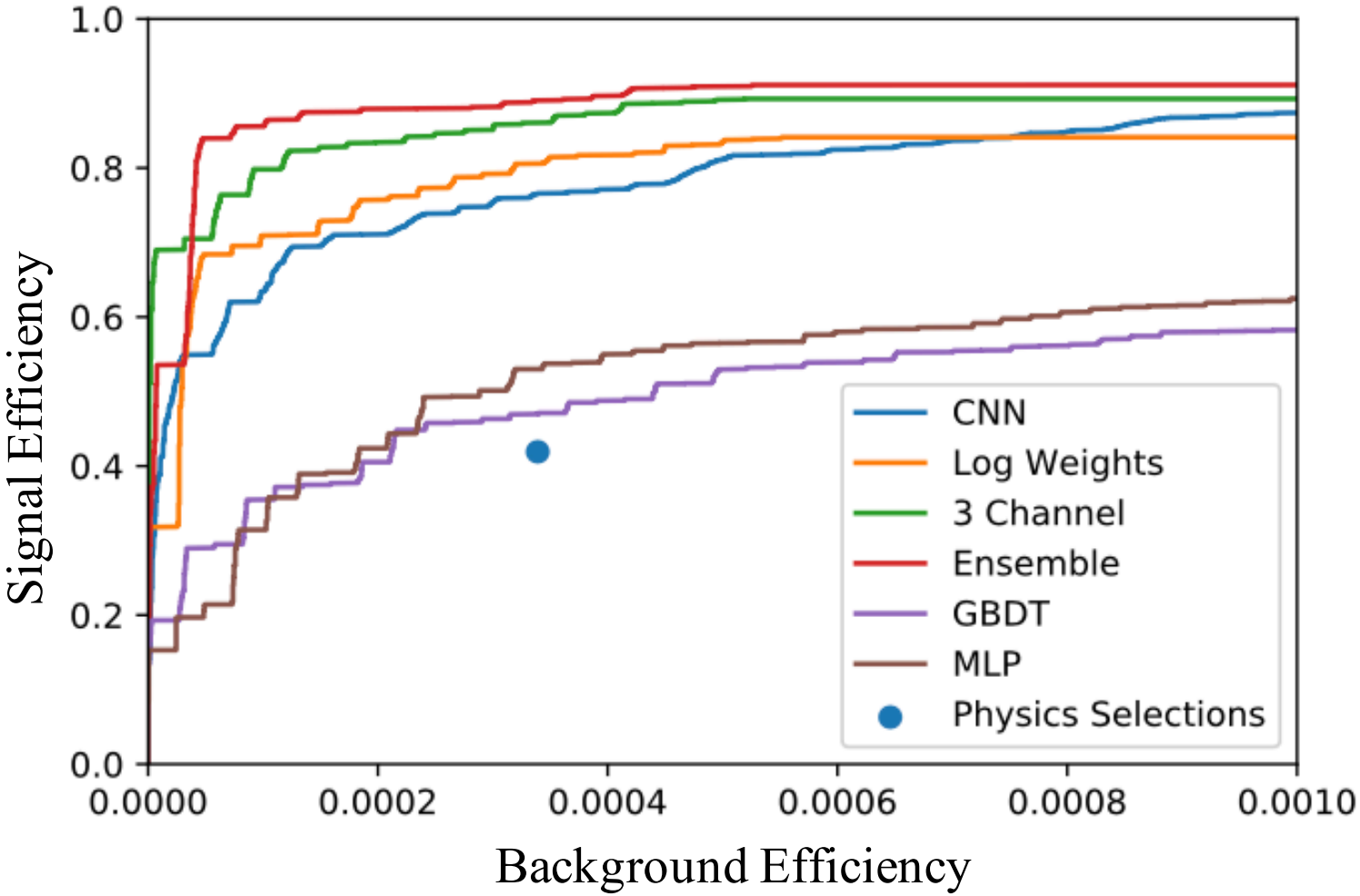


End to end learning (2)





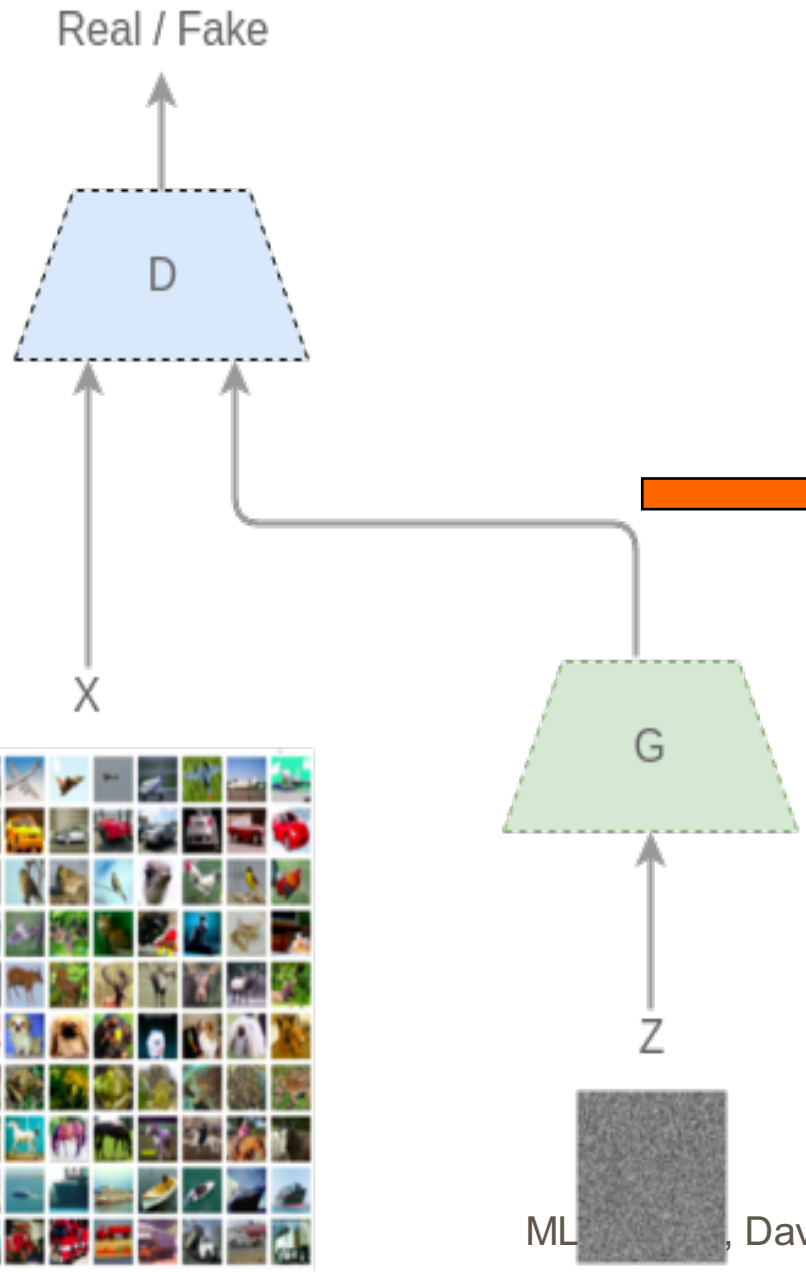
End to end learning (3)

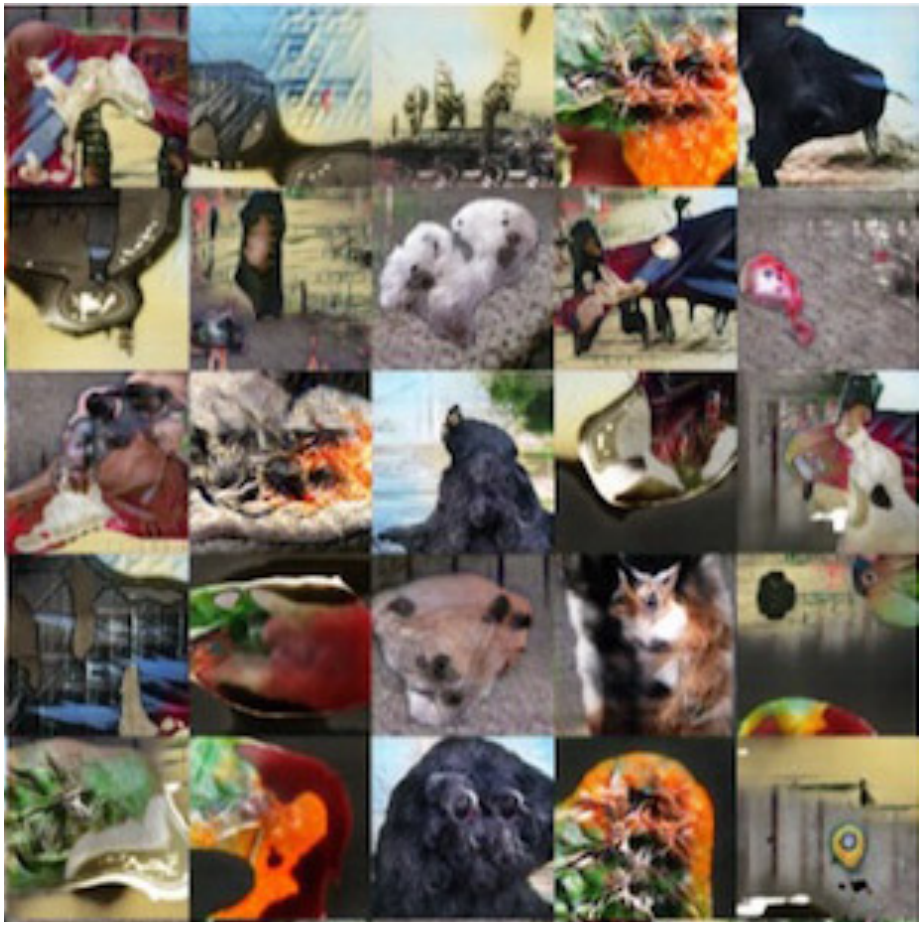


- >x2 gain over BDT/shallow network using physics variable and 5 leading jet 4-momenta
- ➔CNN extract information from energy grid which is lost in the jets ?
- Not sure, they should compare to applying DL on the jets

ML in simulation

Generative Adversarial Network





Condition GAN

Text to image

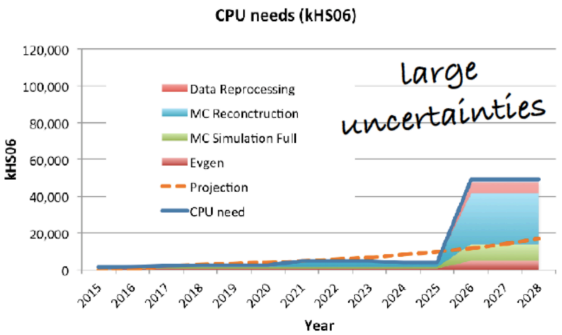
this small bird has a pink breast and crown, and black primaries and secondaries.

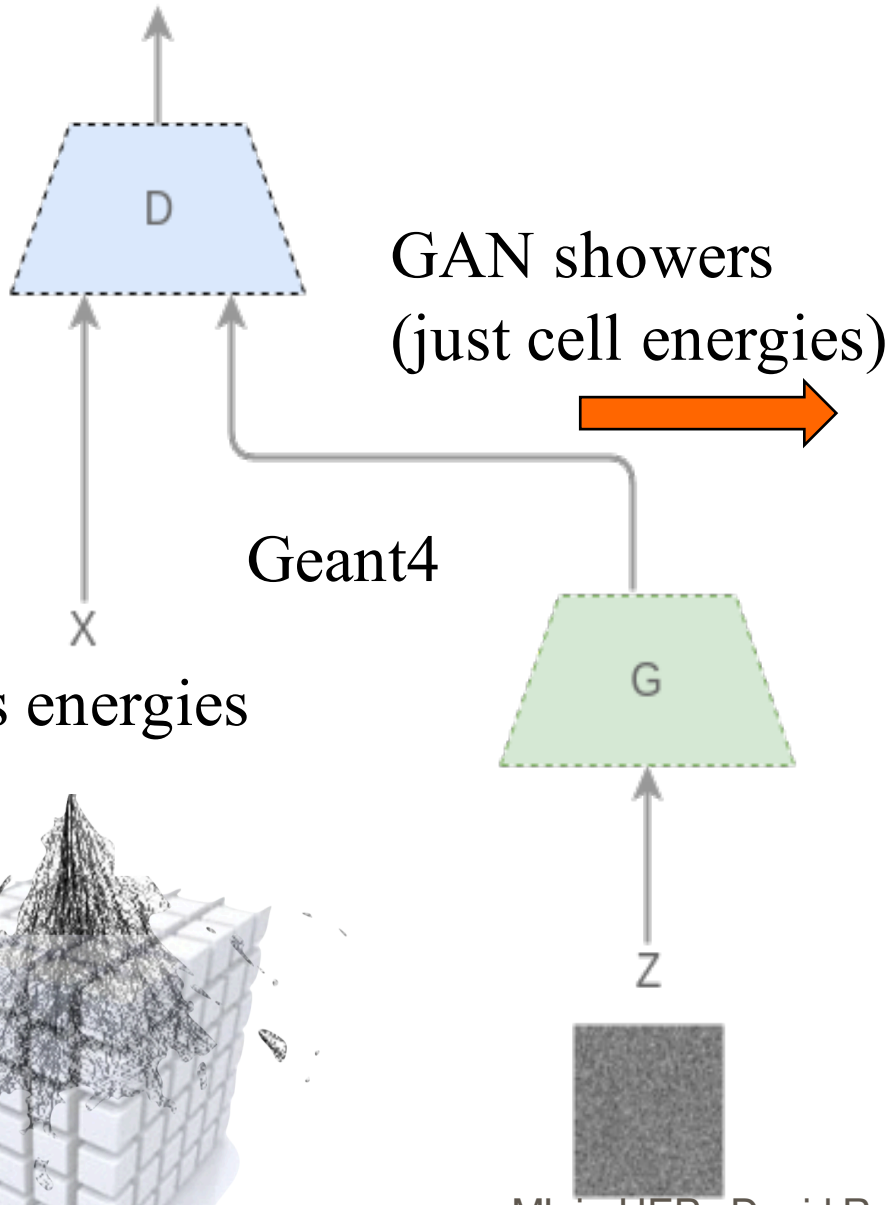
this magnificent fellow is almost all black with a red crest, and white cheek patch.

the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

GAN for simulation (1)

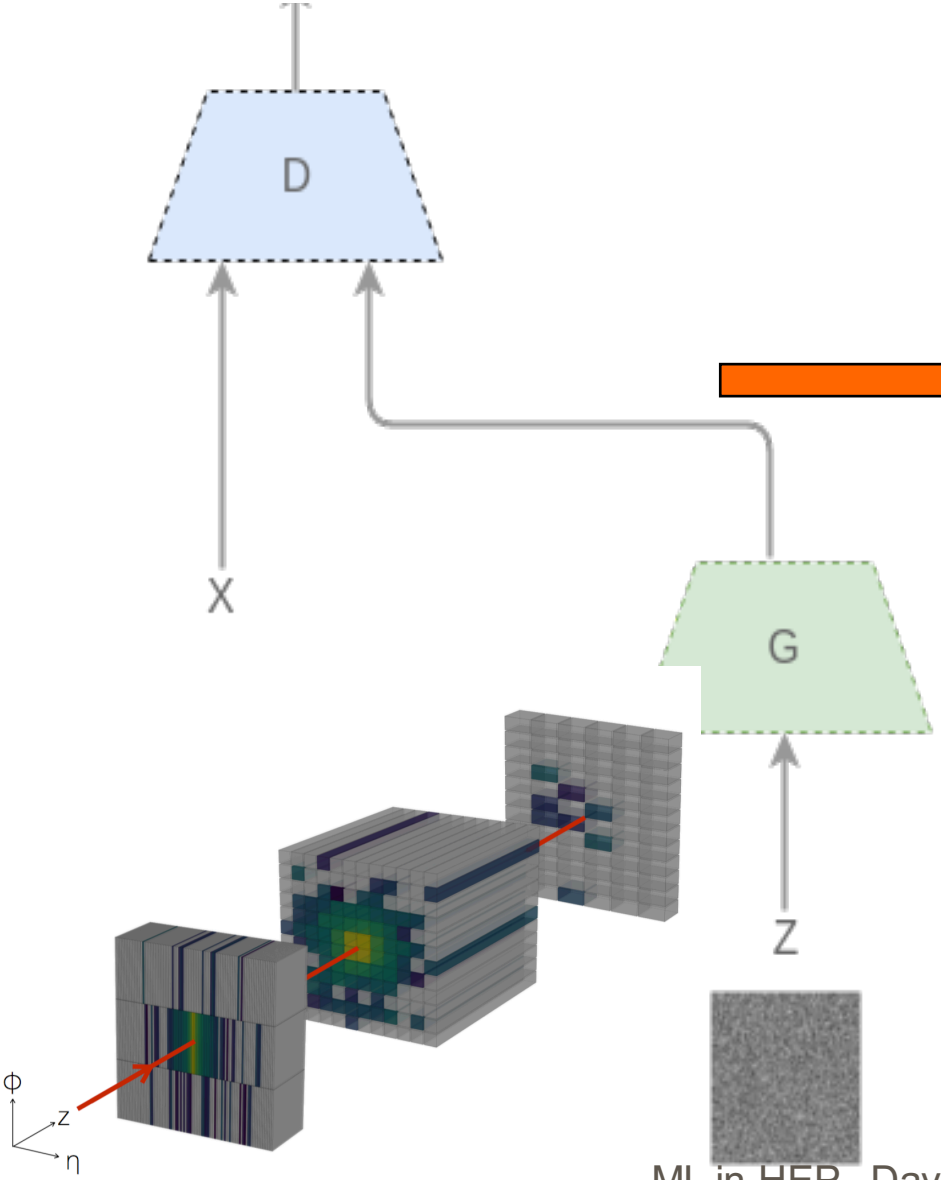




- Half of LHC grid computers (~300.000 cores) are crunching Geant4 simulation 24/24 365/365
- …while LHC experiments are collecting more and more events
- ➔reducing CPU consumption of simulation is very important
- Imagine training a GAN on single particle showers of all types and energies
- Then when an event is simulated it would ask for GAN showers on request (superfast by 3-4 order of magnitude)
- Would replace current fast simulation, frozen shower libraries….
- If/when it works, would require large GPU clusters

GAN for simulation (2)

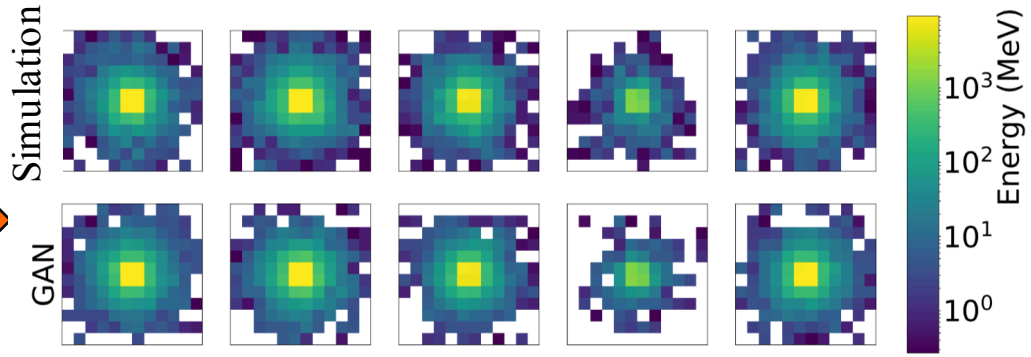
True/False



Paganini et al 1705.02355.

Simplified ATLAS calorimeter

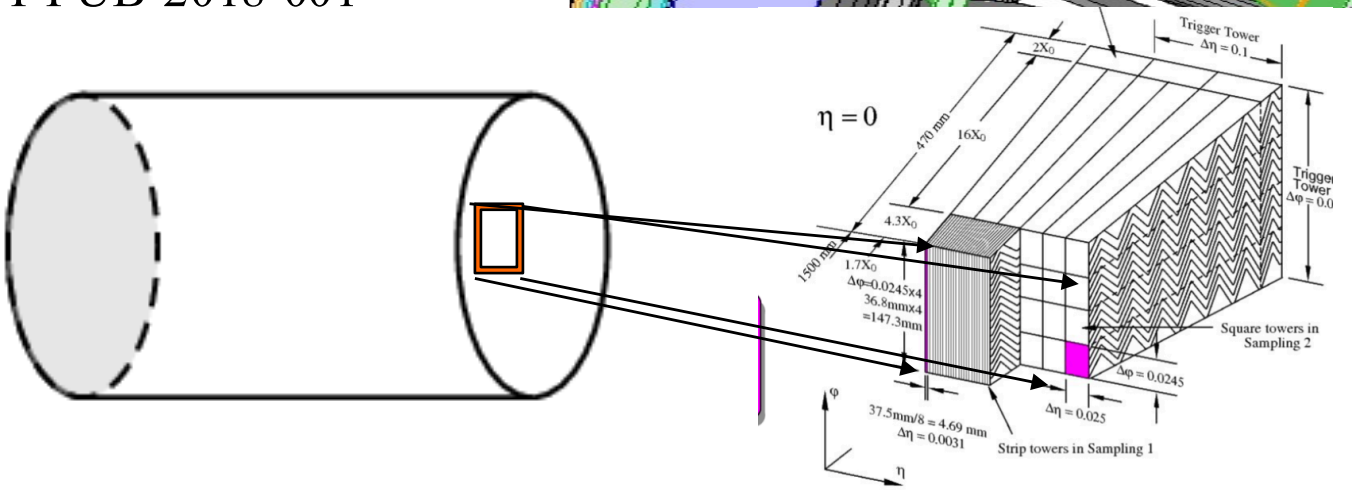
Computing speed-up single shower x1000



Still lot of work to have this work in production

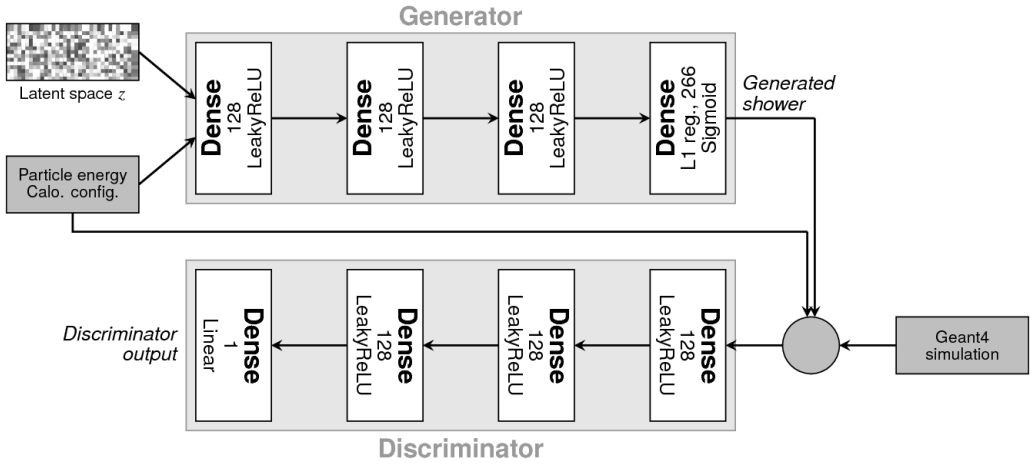
ATLAS calo simulation

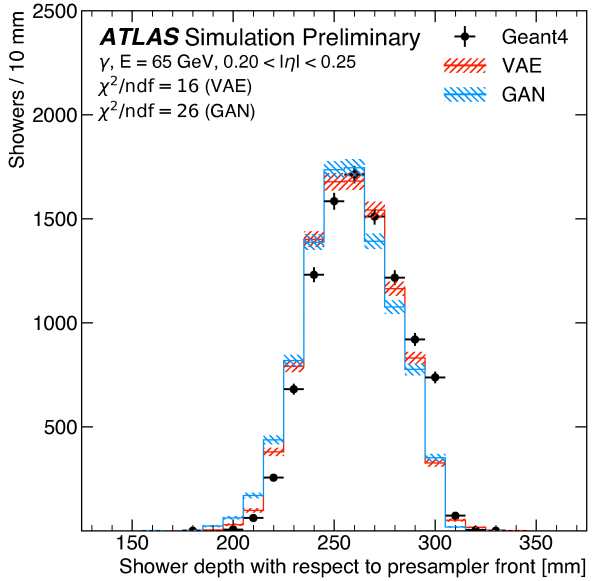
ATL-SOFT-PUB-2018-001



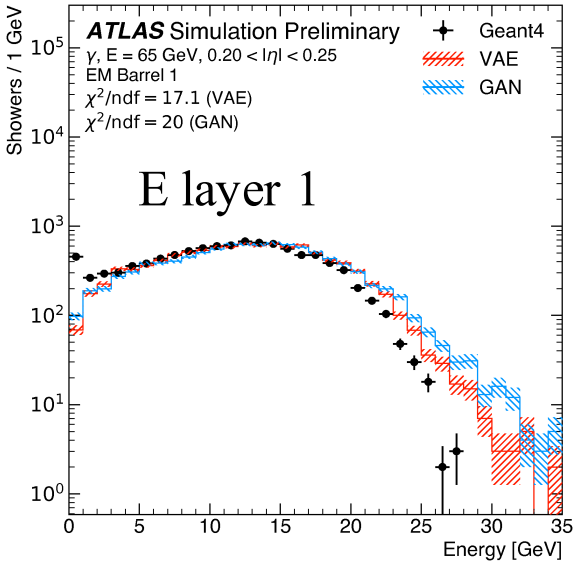
+ η, ϕ translation

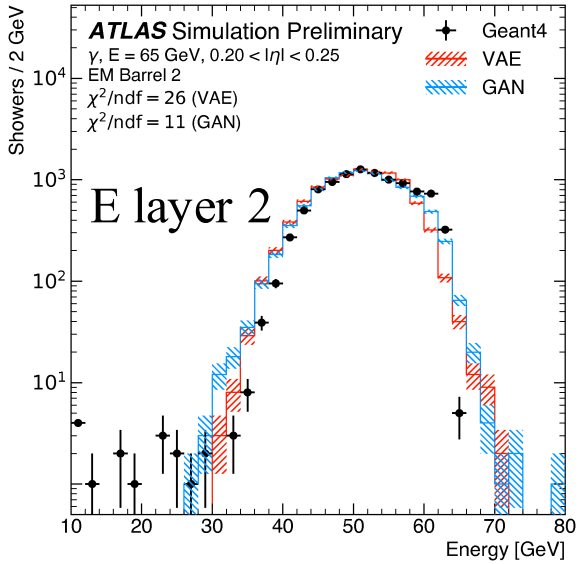
177000 cells ➔266 cells

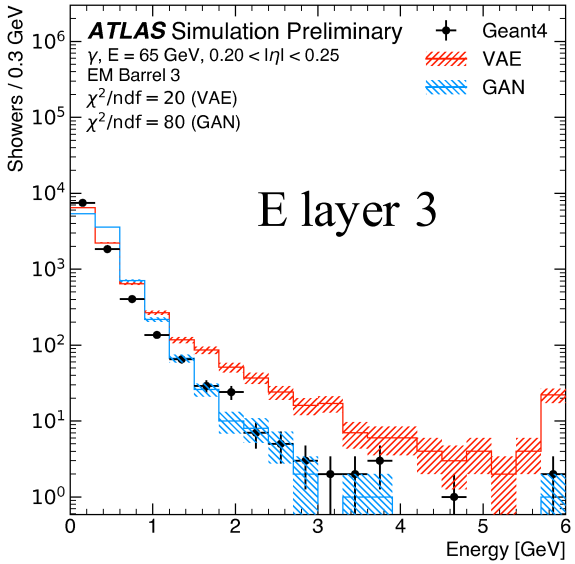


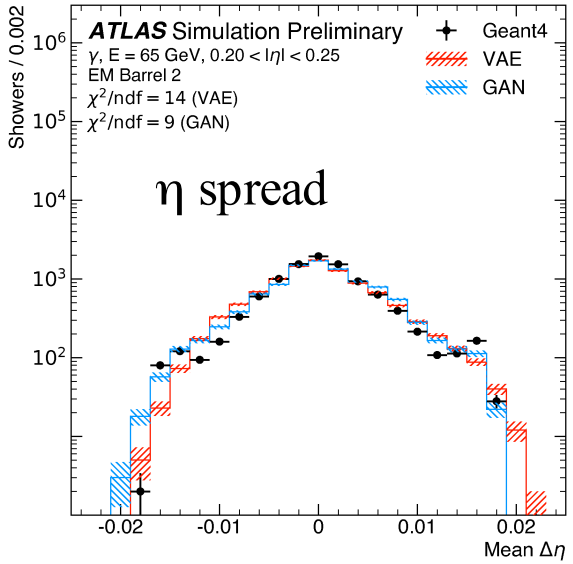


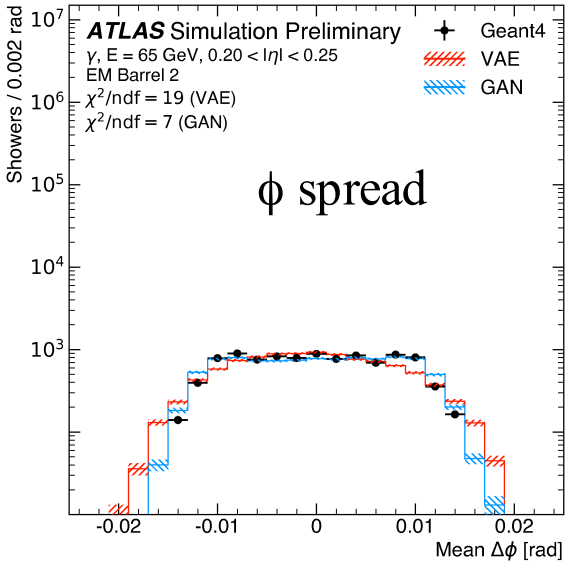
ATLAS Calorimeter GAN











DeepMasterPrints

(b) Real (left) and generated (right) samples for the FingerPass capacitive dataset.

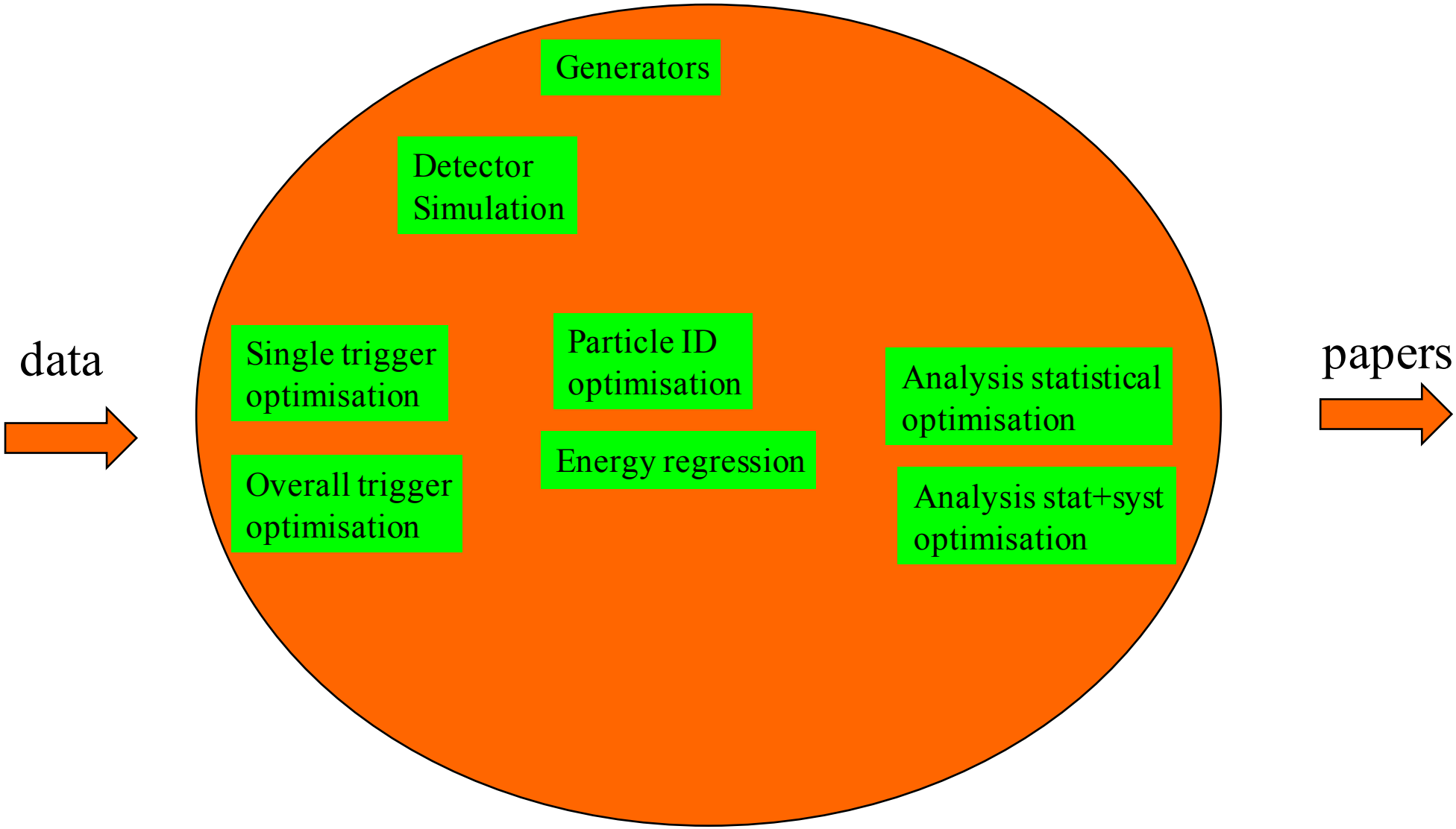
GAN generated fingerprint to fool TouchID like systems

VeriFinger Training	
VeriFinger Test	

Capacitive DeepMasterPrint Matches		
0.01% FMR	0.1% FMR	1% FMR
6.94%	29.44%	89.44%
1.11%	22.50%	76.67%

Wrapping-up

ML playground



ML Collaborations

- Many of the new ML techniques are complex➔difficult for HEP physicists alone
- ML scientists (often) eager to collaborate with HEP physicists
  - prestige
  - new and interesting problems (which they can publish in ML proceedings)
- Takes time to learn common language
- Access to experiment internal data an issue, but there are ways out➔more and more Open Dataset
- Very useful/essential to build HEP - ML collaborations : study on shared dataset, thesis (Computer Science or HEP)
- There is always a friendly Machine Learner on a campus!

Open Data

- Public dataset are essential to collaborate (beyond talking over beer/coffee) on new ML techniques with ML experts (or even physicists in other experiments)
  - can share without experiments Non Disclosure policies
- Some collaborations built on just generator data (e.g. Pythia) or with simple detector simulation e.g. Delphes
  - good for a start, but inaccurate
- Effort to have better open simulation engine (e.g. Delphes 4-vector detector simulation, ACTS for tracking)
- UCI dataset repository has some HEP datasets
- Role of CERN Open Data portal, need be more and more pupulated

Conclusion (1)

- We (in HEP) are analysing data from multi-billion € projects➔should make the most out of it!
- Recent explosion of novel (for HEP) ML techniques, novel applications for Analysis, Reconstruction, Simulation, Trigger, and Computing
- Some of these are ~easy, most are complex: open source software tools are ~easy to get, but still need (people) training, know-how
- Sometimes contradictory results
- Never underestimate the time for :
  - (1) Great ML idea➔
  - (2) …demonstrated on toy dataset➔
  - (3) …demonstrated on semi-realistic simulation ➔
  - (4) …demonstrated on real experiment analysis/dataset ➔
  - (5) …experiment publication using the great idea

(2) Faster ML to production

- Training of HEP students post-docs
  - … and senior scientists
- Campus-level sustained HEP ML collaborations
  - … not just workshops or challenges
- Public datasets
  - …not just toys but also real experimental ones
- Release software with papers
  - …matching "reproducibility" movement in ML
- Computing resources
  - …although (not yet) the limiting factor