

Advances in Machine Learning tools in High Energy Physics



David Rousseau
LAL-Orsay
rousseau@lal.in2p3.fr

LPSC Seminar, Tuesday 4th October 2016

Outline

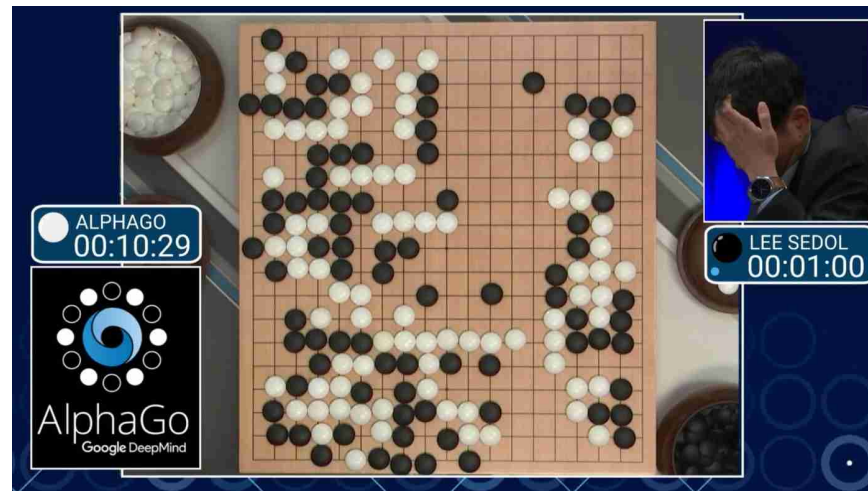


- Basics
- ML software tools
- ML techniques
- ML in analysis
- ML in reconstruction/simulation
- Data challenges
- Wrapping up

ML in HEP



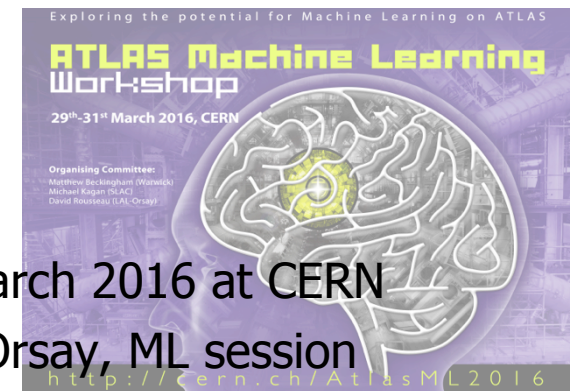
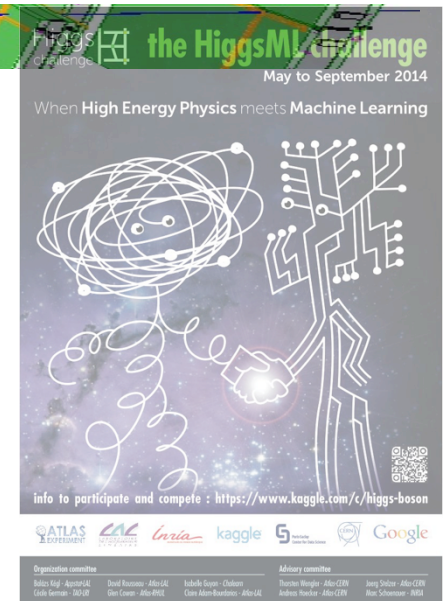
- ❑ Use of Machine Learning (a.k.a Multi Variate Analysis as we used to call it) already at LEP somewhat (Neural Net), more at Tevatron (Trees)
- ❑ At LHC, Machine Learning used almost since first data taking (2010) for reconstruction and analysis
- ❑ In most cases, Boosted Decision Tree with Root-TMVA
- ❑ Meanwhile, in the outside world :



- ❑ “Artificial Intelligence” not a dirty word anymore!
- ❑ We’ve realised we’re been left behind! Trying to catch up now...

Multitude of HEP-ML events

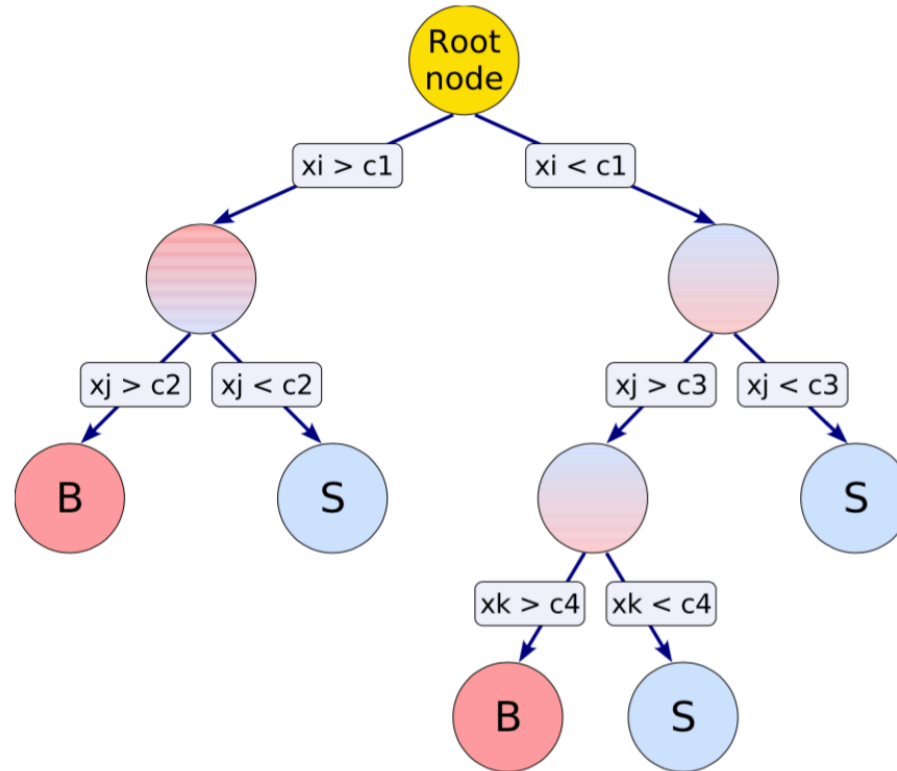
- ❑ HiggsML Challenge, summer 2014
 - → HEP ML NIPS satellite workshop, December 2014
- ❑ Connecting The Dots, Berkeley, January 2015
- ❑ Flavour of Physics Challenge, summer 2015
 - → HEP ML NIPS satellite workshop, December 2015
- ❑ DS@LHC workshop, 9-13 November 2015
 - → future DS@HEP workshop
- ❑ LHC Interexperiment Machine Learning group
 - Started informally September 2015, gaining speed
- ❑ 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
- ❑ (internal) ATLAS Machine Learning workshop 29-31 March 2016 at CERN
- ❑ Hep Software Foundation workshop 2-4 May 2016 at Orsay, ML session
- ❑ TrackML Challenge, summer 2017?



ML Basics

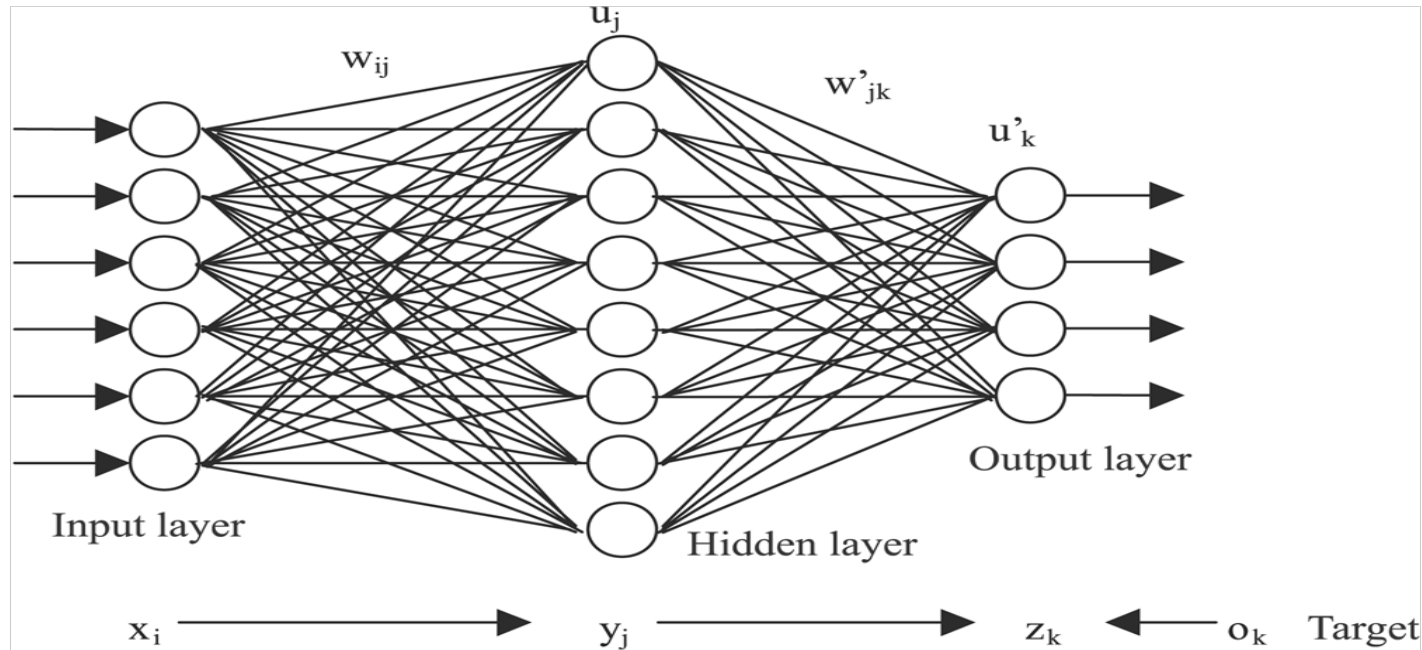


BDT in a nutshell



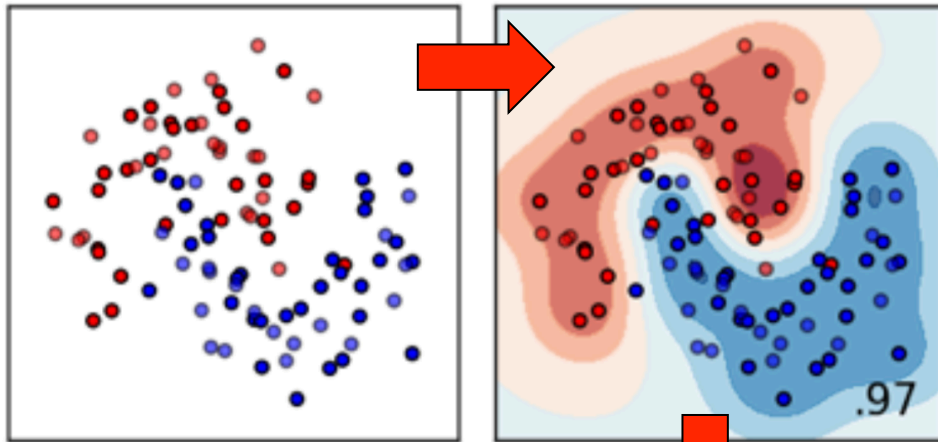
- ❑ Single tree (CART) <1980
- ❑ AdaBoost 1997 : rerun increasing the weight of misclassified entries → boosted trees

Neural Net in a nutshell



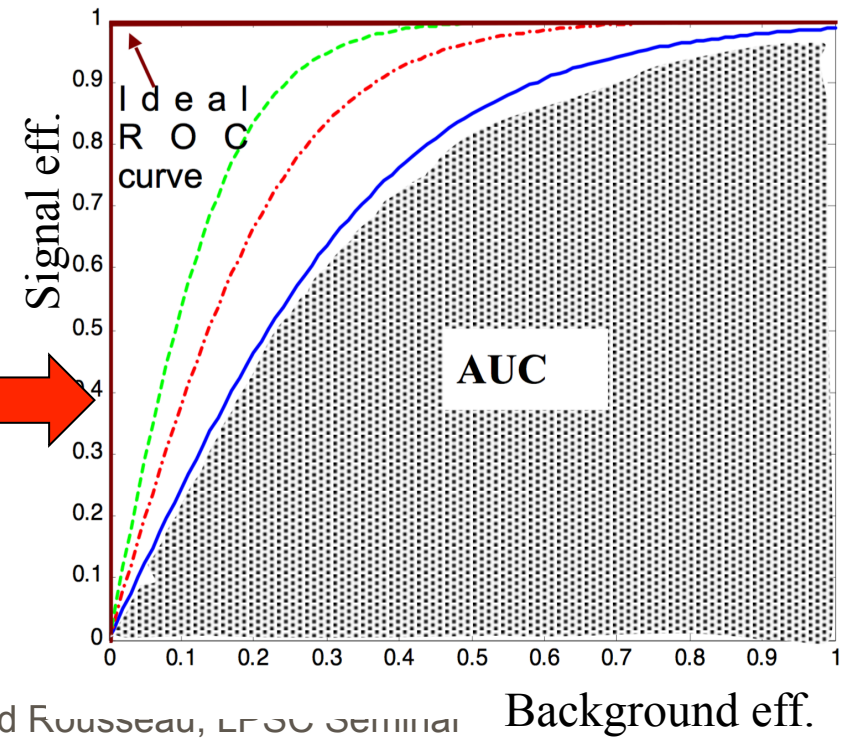
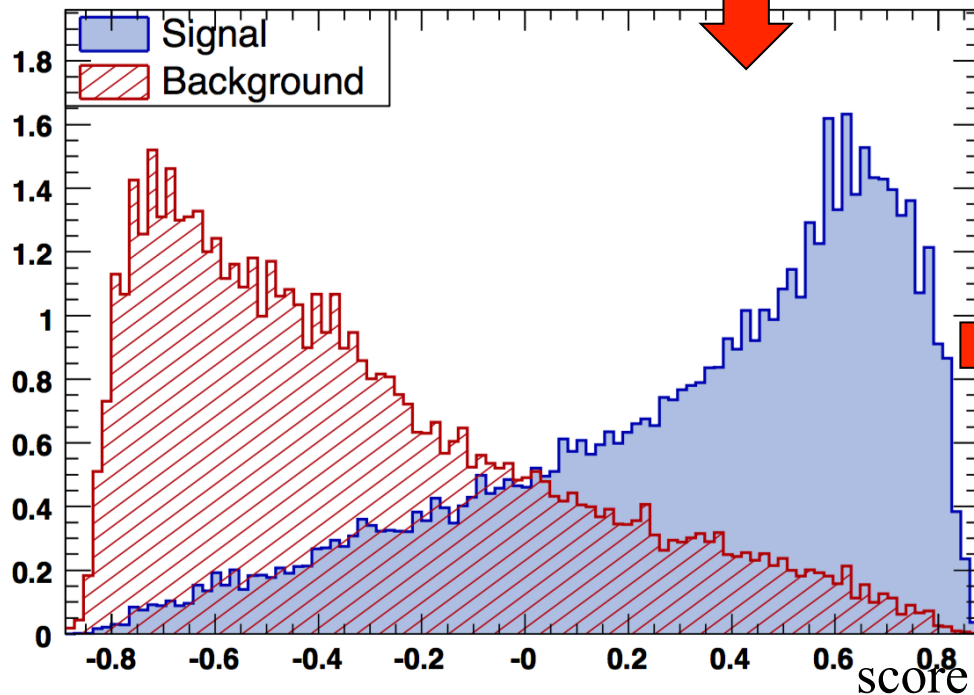
- ❑ Neural Net ~1950!
- ❑ But many many new tricks for learning, in particular if many layers (also ReLU instead of sigmoid activation)
- ❑ "Deep Neural Net" up to 50 layers
- ❑ Computing power (DNN training can take days even on GPU)

Any classifier

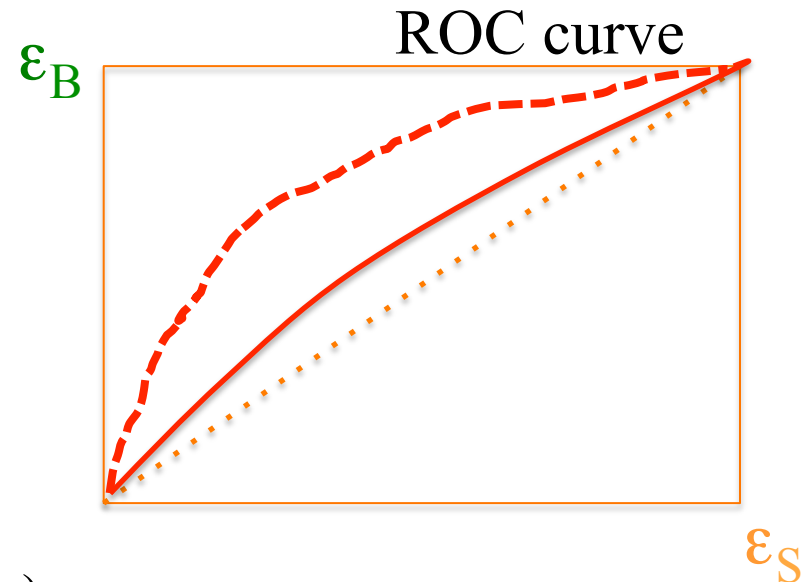
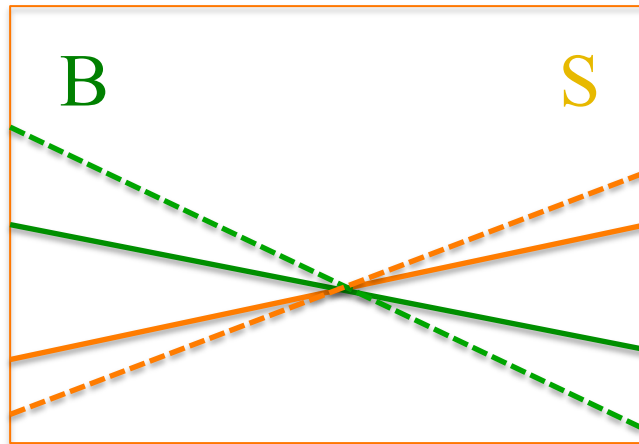


Classification : learn label 0 or 1
Regression : learn continuous variable

AUC : Area Under the (ROC) Curve



Overtraining



- score
- Evaluated on training dataset (wrong)
 - Evaluated on independent test dataset (correct)

Score distribution different on test dataset wrt training dataset
→ "Overtraining" == possibly excessive use of statistical fluctuation

More vocabulary



□ “Hyper-parameters”:

- These are all the “knobs” to optimize an algorithm, e.g.
 - number of leaves and depth of a tree
 - number of nodes and layers for NN
 - and much more
- “Hyper-parameter tuning/fitting” == optimising the knobs for the best performance

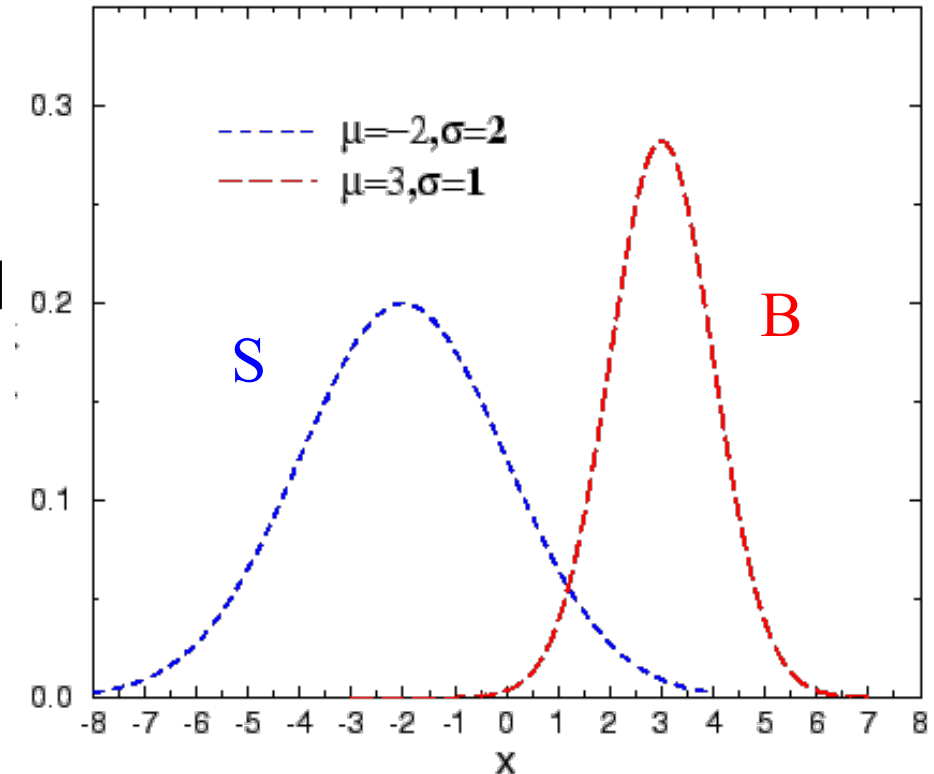
□ “Features”

- variables

No miracle



- ❑ ML does not do miracles
- ❑ If underlying distributions are known, nothing beats Likelihood 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



ML Tools

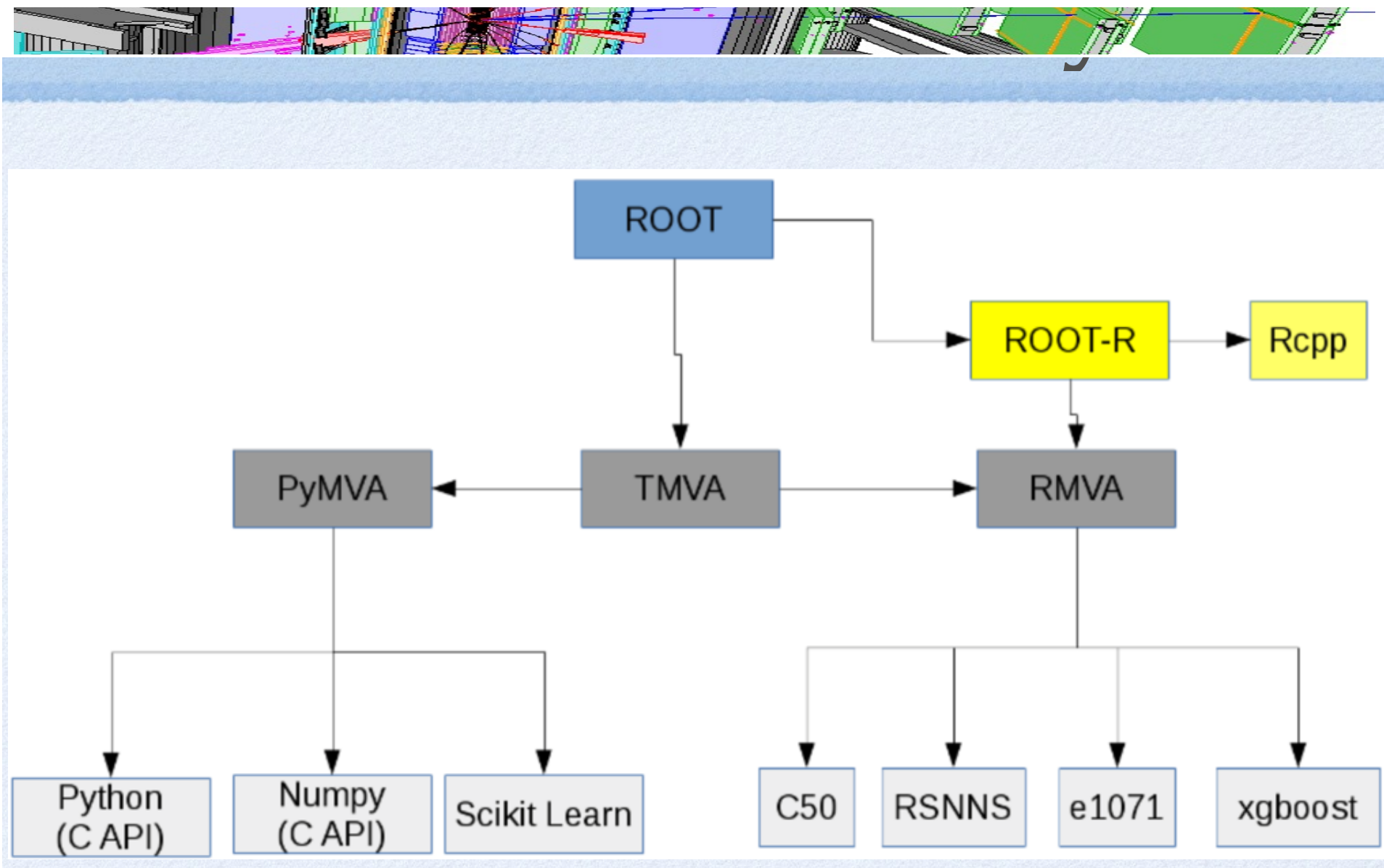


ML Tool : TMVA



- ❑ Root-TMVA de-facto standard for ML in HEP
- ❑ Has been instrumental into “democratising” ML at LHC (at least)
- ❑ Well coupled with Root (which everyone uses)
- ❑ But:
 - Has sterilized somewhat the creativity
 - Mostly frozen the last few years, left behind
- ❑ However:
 - Rejuvenating effort since summer 2015
 - Revise structure for more flexibility
 - Jupyter interface
 - Improve algorithms
 - “Envelope methods” for automatic hyper parameter tuning, cross-validation
 - Interface to the outside world (R, scikit-learn)
- ❑ See [talk Lorenzo Moneta](#) at Hep Software Fondation workshop at LAL in June 2016

TMVA interfaces ROOT v>= 6.05.02



ML Tool : XGBoost



- ❑ XGBoost : Xtreme Gradient Boosting :
<https://github.com/dmlc/xgboost>, [arXiv:1603.02754](https://arxiv.org/abs/1603.02754)
- ❑ Written originally for HiggsML challenge
- ❑ Used by many participants, including number 2
- ❑ Meanwhile, used by many other participants in many other challenges
- ❑ Open source, well documented, and supported
- ❑ Has won many challenges meanwhile
- ❑ Best BDT on the market, performance and speed
- ❑ Classification and regression

ML Tool : SciKit-learn



- ❑ SciKit-Learn : Machine Learning in python
- ❑ Modern Jupyter interface (notebook à la Mathematica)
- ❑ Open source (several core developers in Paris-Saclay)
- ❑ Built on NumPy, SciPy, and matplotlib
- ❑ (very fast, despite being python)
- ❑ Install on any laptop with Anaconda
- ❑ All the major ML algorithms (except deep learning)
- ❑ Superb documentation
- ❑ Quite different look and fill from Root-TMVA
- ❑ Short demo (Navigator should be started)

ML platforms



- ❑ Training time can become prohibitive (days), especially Deep Learning, especially with large datasets
- ❑ With hyper-parameter optimisation, cross-validation, number of trainings for a particular application large ~ 100
- ❑ Emergence of ML platforms :
 - Dedicated cluster (with GPUs)
 - Relevant software preinstalled (VM)
 - Possibility to load large datasets (GB to TB)
- ❑ At CERN SWAN now in production
 - Jupyter interface
 - Access to your CERNbox or to eos

ML Techniques

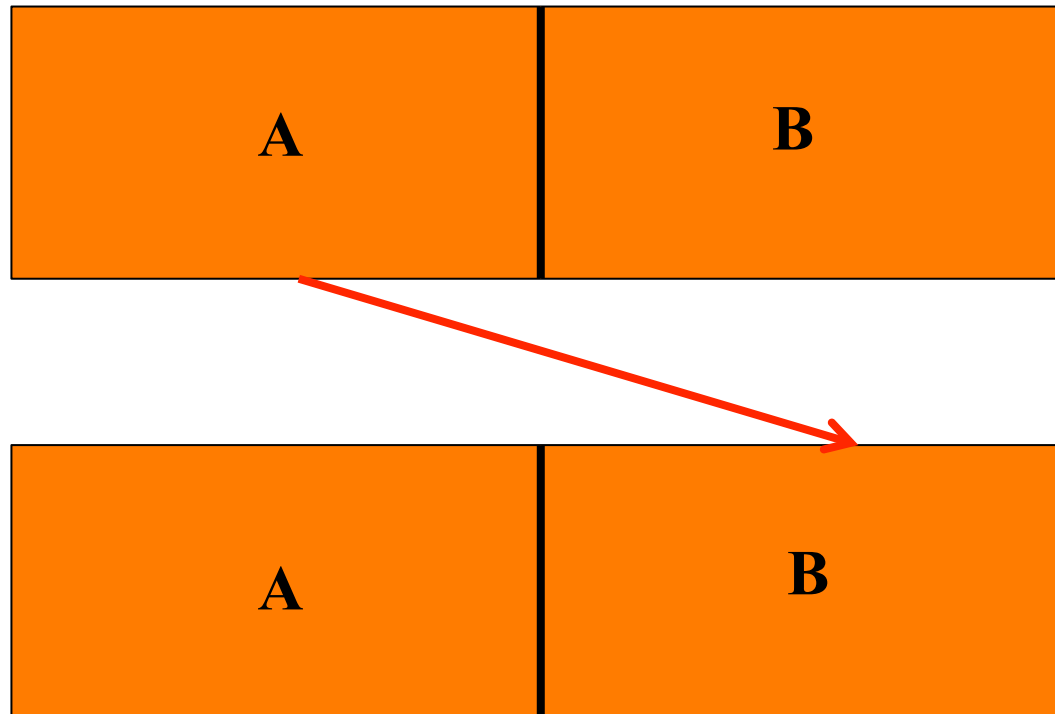


Cross-Validation



One-fold Cross Validation

Goal of CV is to measure performance and optimise hyper-parameters

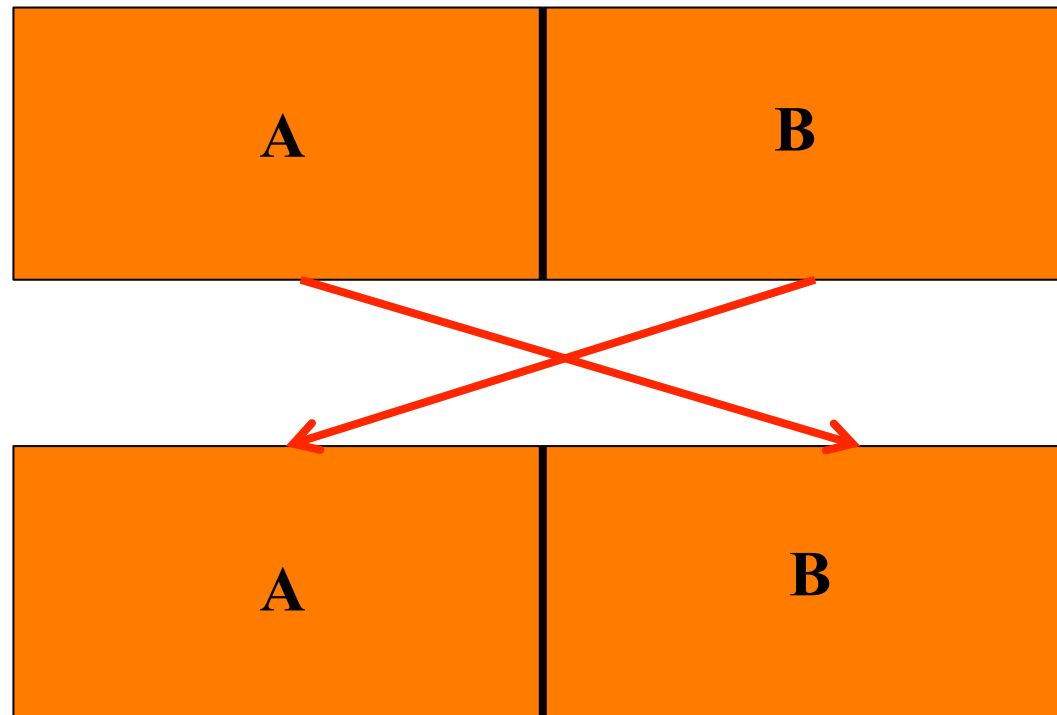


Standard basic way (default TMVA)

Cross-Validation



Two-fold Cross Validation

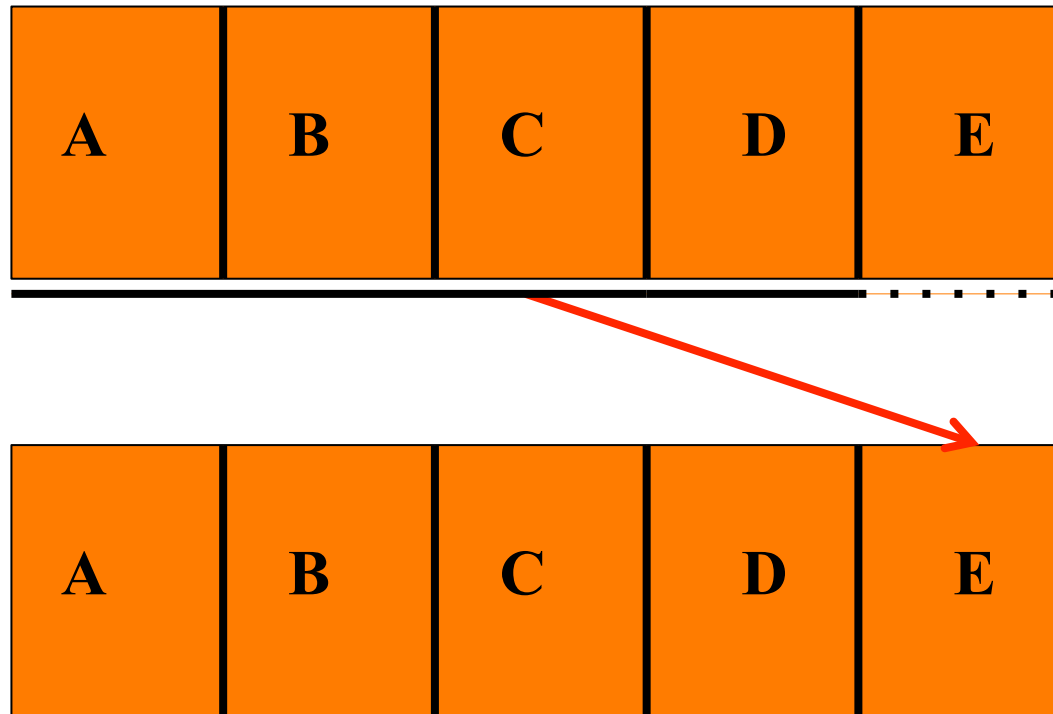


- ➔ test statistics = total statistics
- ➔ double test statistics wrt one fold CV
- ➔ (double training time of course)

Cross-Validation



5-fold Cross Validation

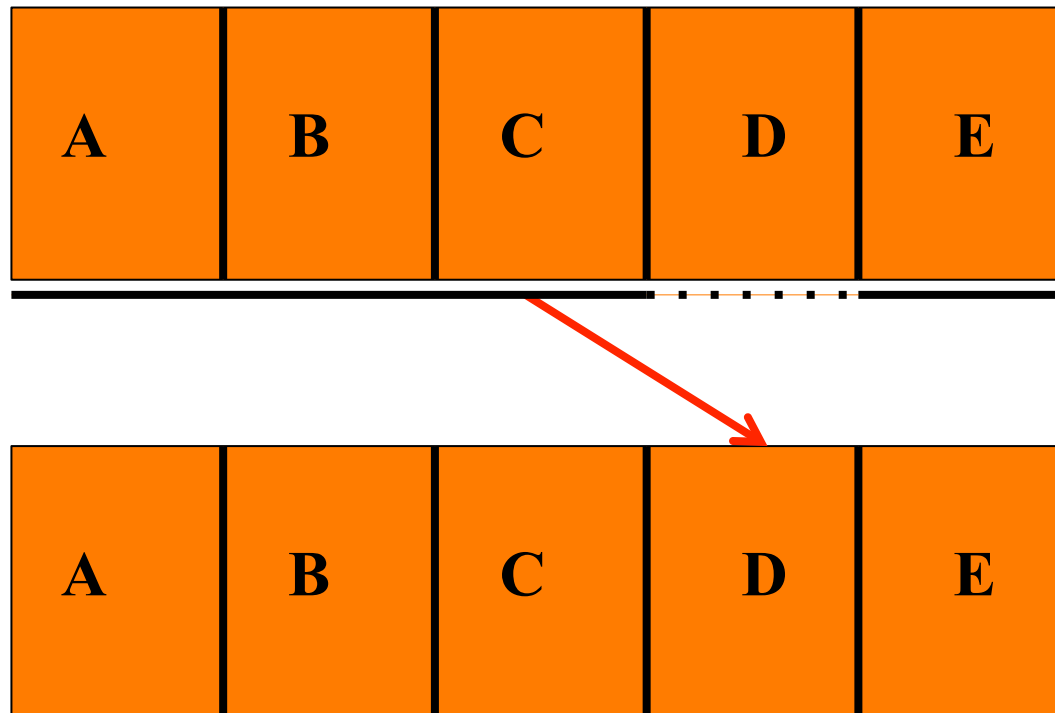


same test statistics wrt two-fold CV,
larger training statistics $4/5$ over $1/2$ (larger training time as well)
bonus: variance of the samples an estimate of the statistical uncertainty

Cross-Validation



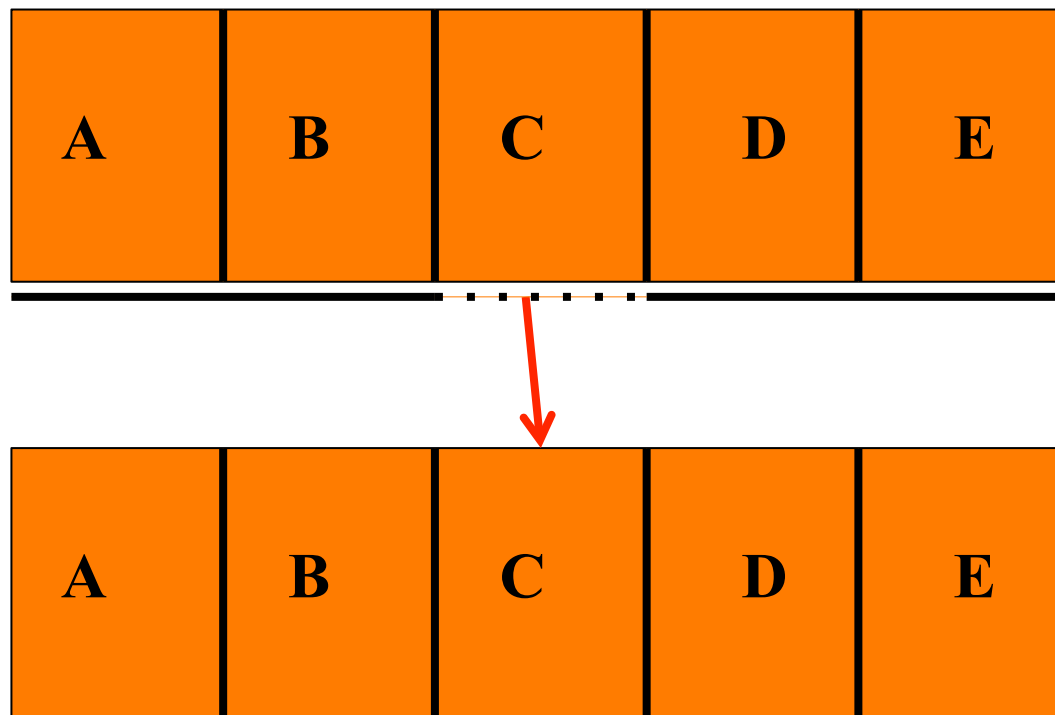
5-fold Cross Validation



Cross-Validation



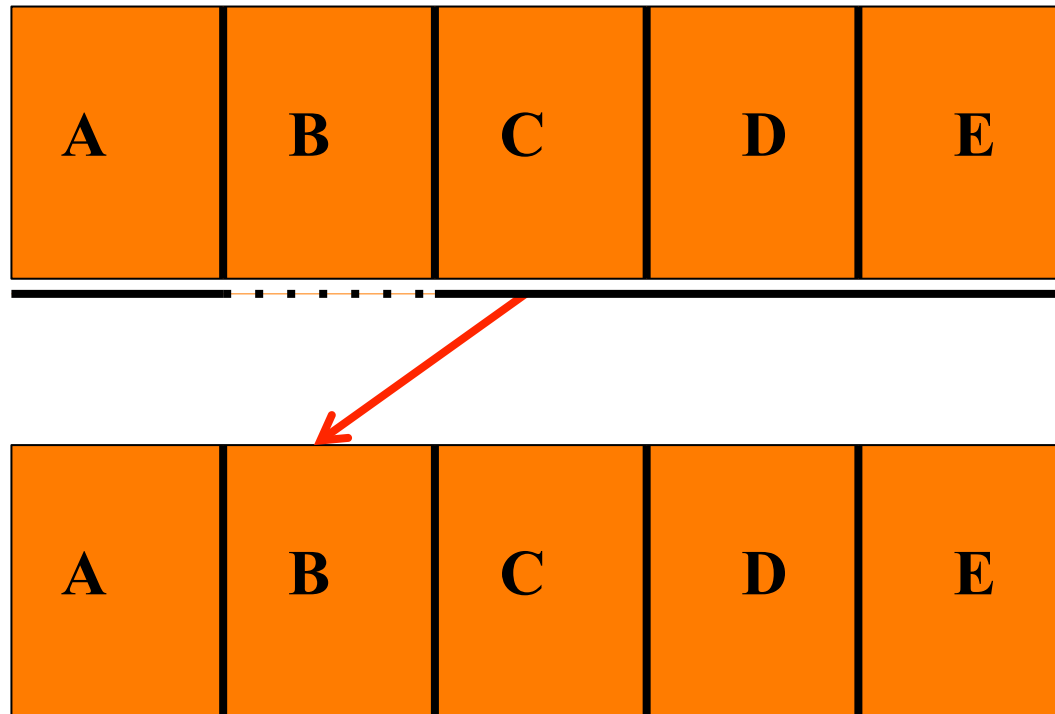
5-fold Cross Validation



Cross-Validation



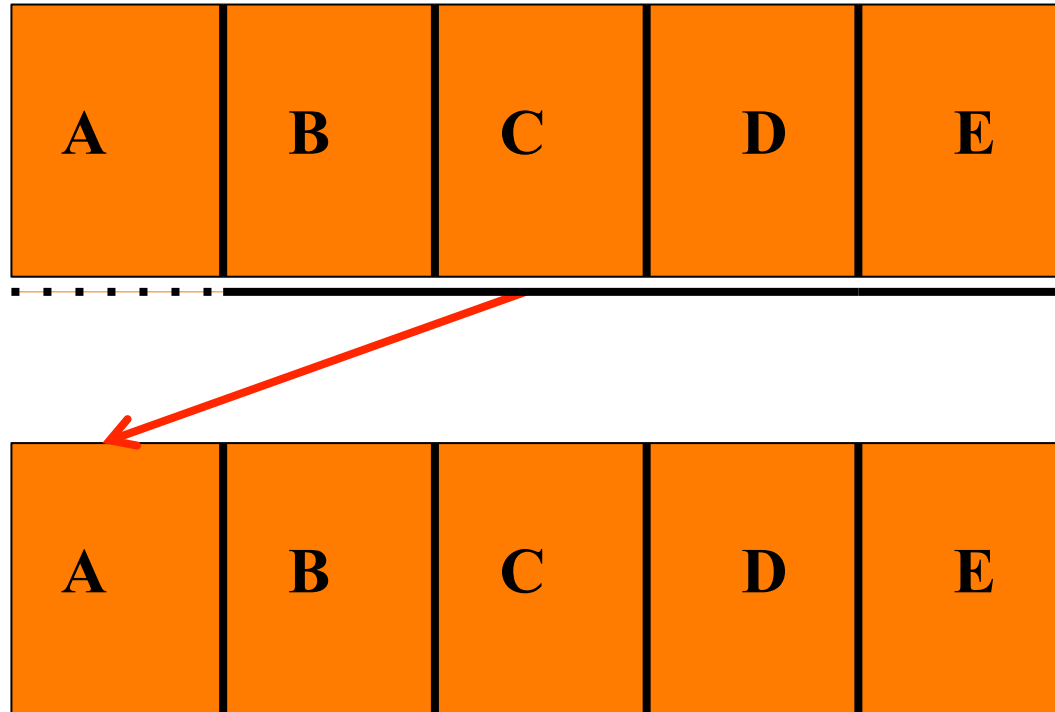
5-fold Cross Validation



Cross-Validation



5-fold Cross Validation

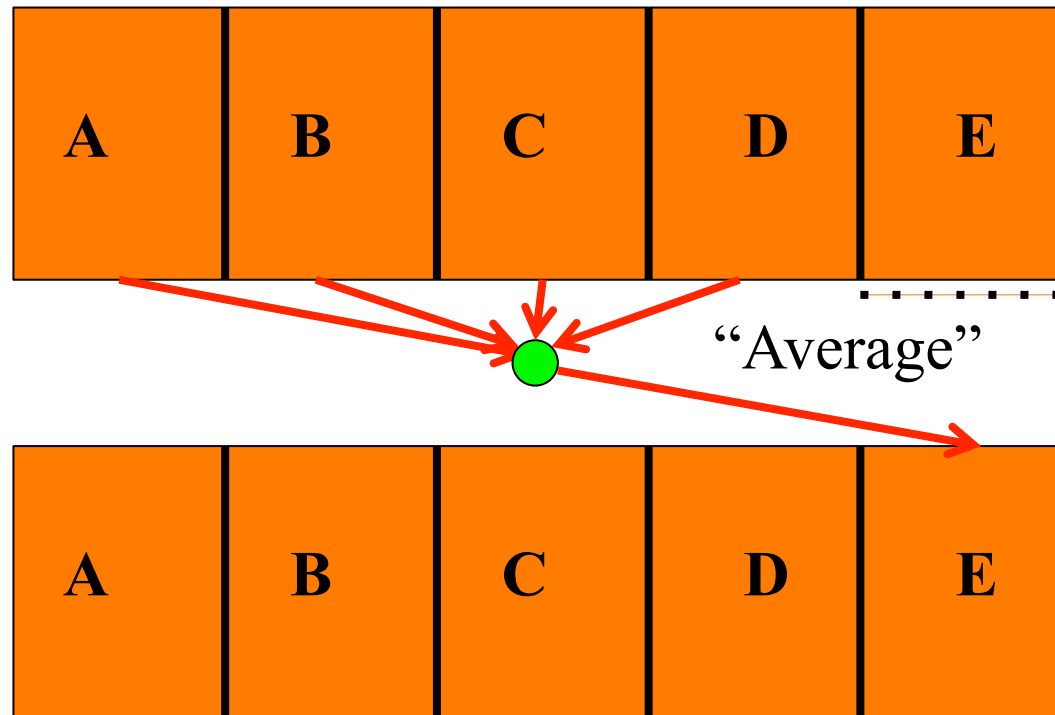


Note : if hyper-parameter tuning, need a third level of independent sample “nested CV”

Cross-Validation



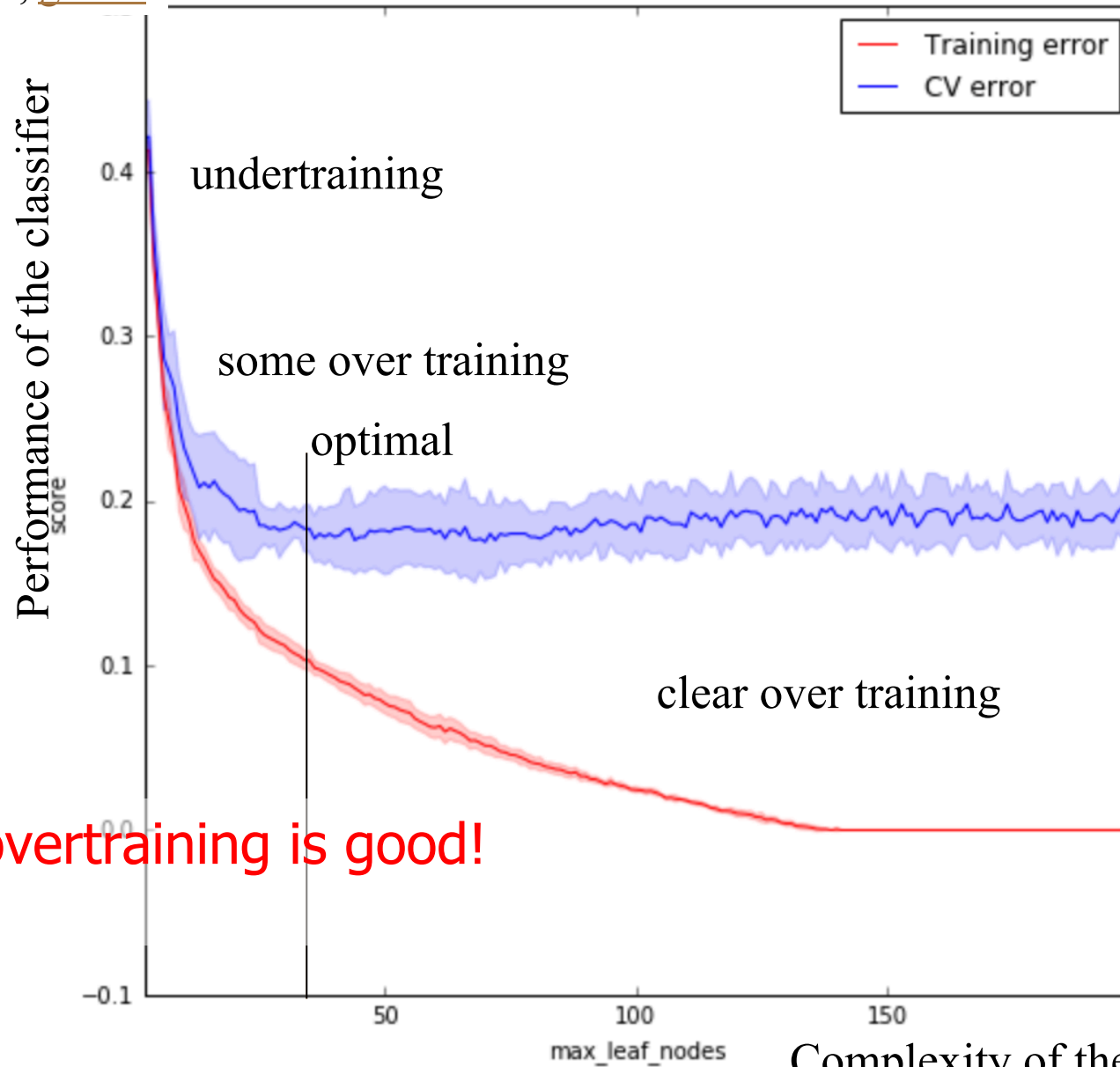
5-fold Cross Validation “à la Gabor”



Average of the scores on A B C D is
often better than the score of one training ABCD
(also save on training time)

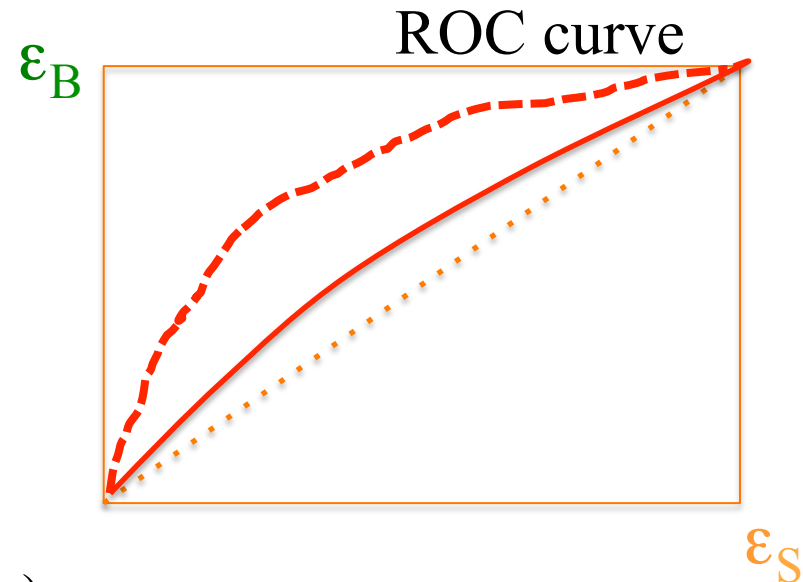
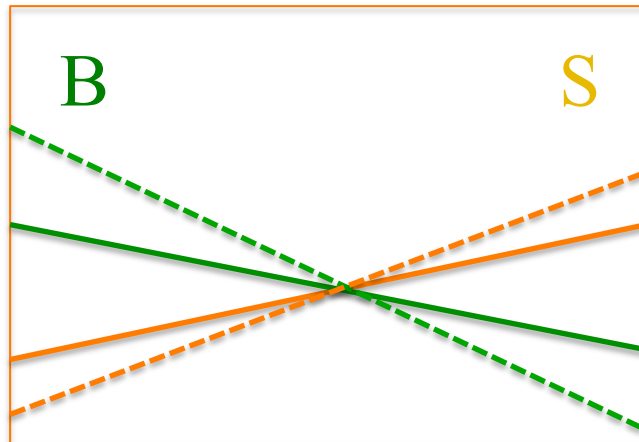
CV, under/over training

Gilles Louppe, [github](#)



Some overtraining is good!

(reminder) Overtraining



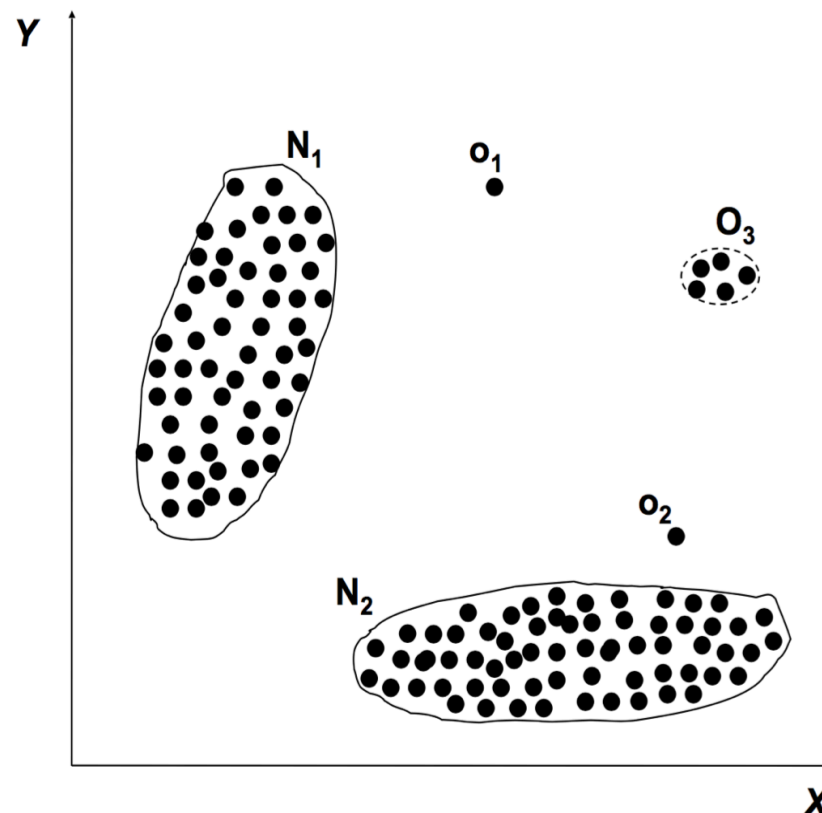
- score
- Evaluated on training dataset (wrong)
 - Evaluated on independent test dataset (correct)

Score distribution different on test dataset wrt training dataset
→ "Overtraining" == **possibly** excessive use of statistical fluctuation

Anomaly : point level



- Also called outlier detection
- “unsupervised learning”
- Two approaches:
 - Give the full data, ask the algorithm to cluster and find the lone entries : o_1 , o_2 , o_3



- We have a training “normal” data set with N_1 and N_2 . Algorithm should then spot o_1, o_2, o_3 as “abnormal” i.e. “unlike N_1 and N_2 ” (no a priori model for outliers)
- Application : detector malfunction, grid site malfunction, or even new physics discovery...

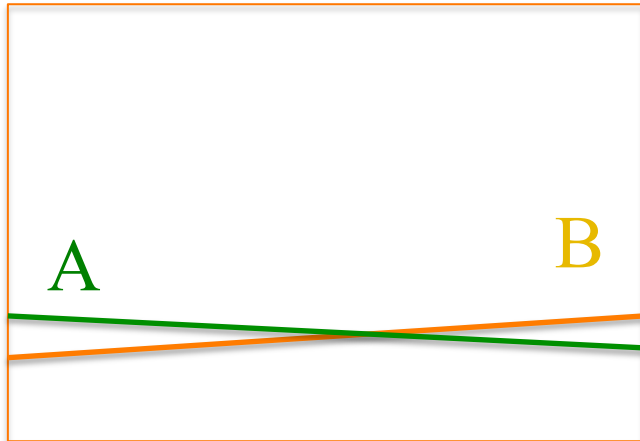
Anomaly : population level



- ❑ Also called collective anomalies
- ❑ 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 CERN, production at BNL
 - 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)
- ❑ ML approach : ~~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

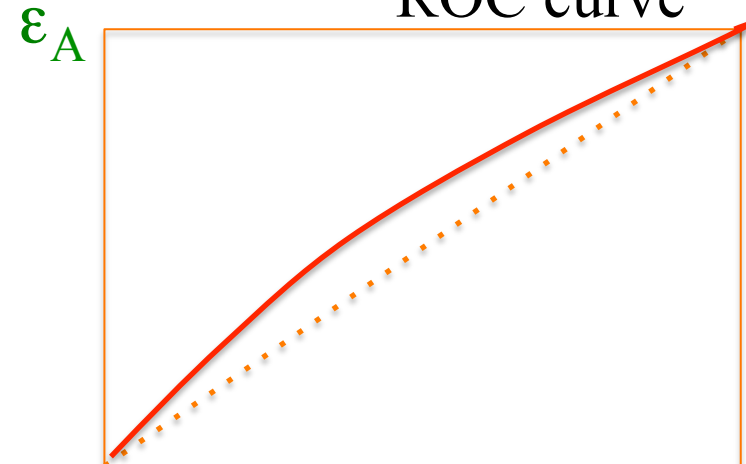


Small non-local difference



score

ROC curve



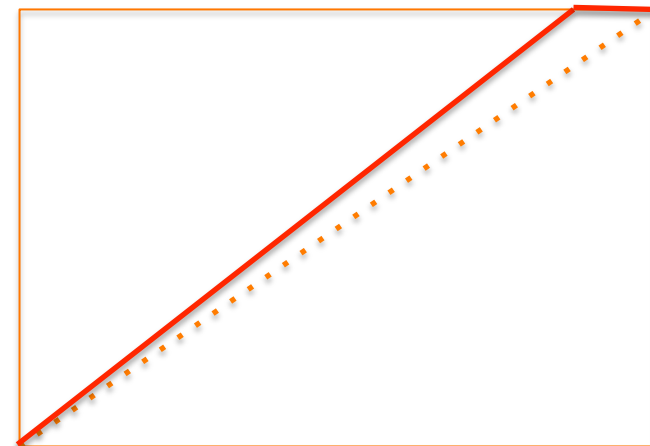
ϵ_B

Local big difference (e.g. non overlapping distribution, hole)



score

ϵ_A



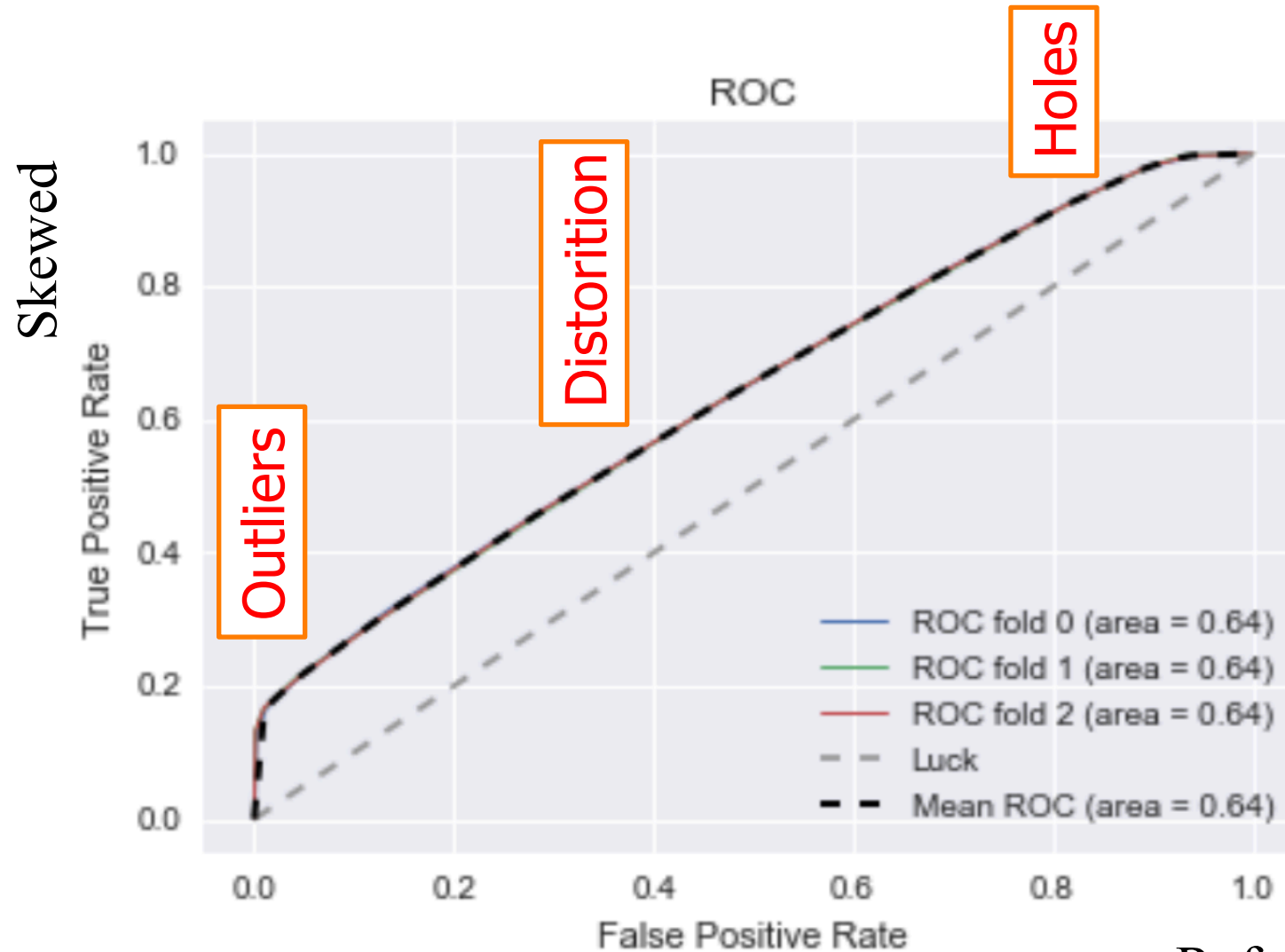
ϵ_B

HSF ML RAMP on anomaly



- ❑ RAMP : collaborative competition around a dataset and a figure of merit. Organised in June 2016 by CDS Paris Saclay with HEP people. See [agenda](#).
- ❑ Dataset built from the Higgs Machine Learning challenge dataset (on CERN Open Data Portal)
 - Lepton, and tau hadron 3 momentum, MET : PRImary variables
 - DERived variables (computed from the above) from Htautau analysis
 - Jet variables dropped
- ❑ →reference dataset
- ❑ “Skewed” dataset built from the above, introducing small and big distortions:
 - Small scaling of Ptau
 - Holes in eta phi efficiency map of lepton and tau hadron
 - Outliers introduced, each with 5% probability
 - Eta tau set to large non possible values
 - P lepton scaled by factor 10
 - Missing ET + 50 GeV
 - Phi tau and phi lepton swapped → DERived variables inconsistent with PRImary one
- ❑ →skewed dataset

HSF ML RAMP on anomaly (2)



Reference

HSF RAMP (2)

team	submission	accuracy
mcherti	adab2_mt1_calibrated	0.611
dhrou	adab2_mt1	0.611
kazeevn	GradientBoosting	0.596
glouppe	bags2	0.594
glouppe	boosting-duo	0.595
mcherti	adaboost2	0.594
glouppe	bags	0.593
mcherti	adaboost1	0.593
djabbz	beta tester	0.591
soobash	ExtraTreesClassifier	0.576
mcherti	extratrees1	0.562
dhrou	DRv0	0.553
calaf	starting_kit_paolo	0.526

Breakthrough : add new variable:

$$\Delta m_T = \sqrt{(2P_{IT} * MET * (1 - \cos(\phi_l - \phi_{MET})))} - m_T$$

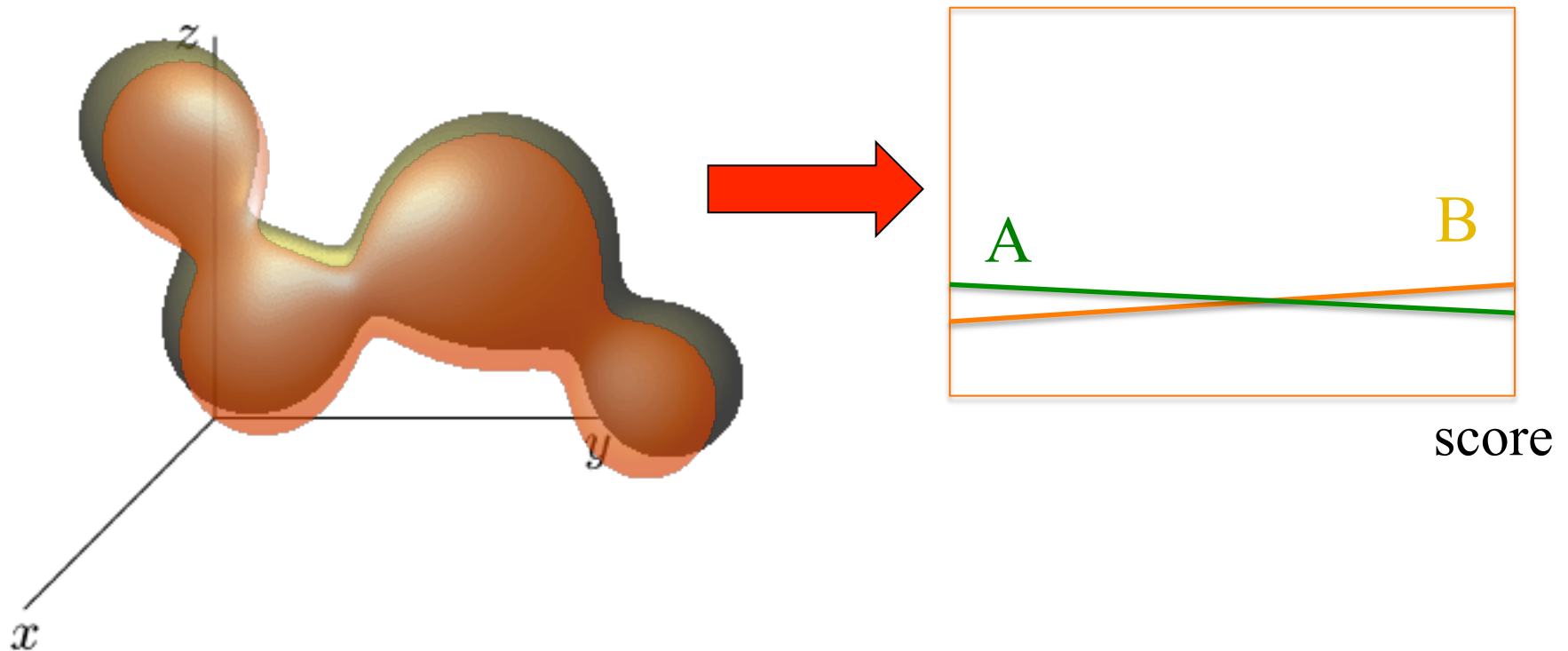
Non zero for some outliers

→ classifiers were unable to guess it

Classifier optimisation

→ what functional form classifiers can learn ?

What does a classifier do?



- The classifier “projects” the two multidimensional “blobs” maximising the difference, without (ideally) any loss of information

Re-weighting



- Suppose a variable distribution is slightly different between a Source (e.g. Monte Carlo) and a Target (e.g. real data)

○ → reweight!



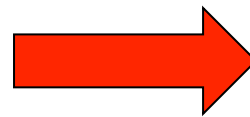
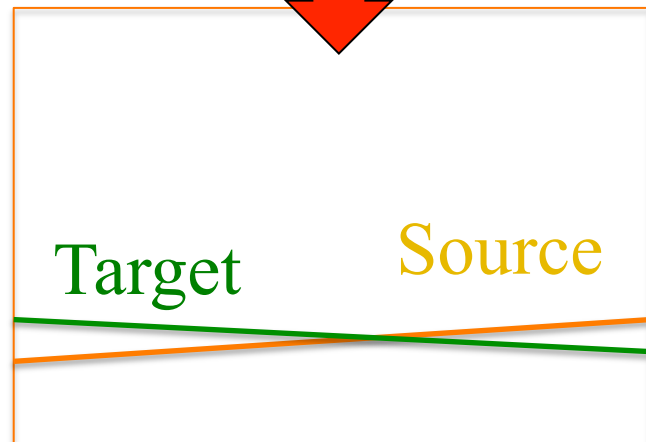
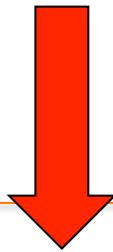
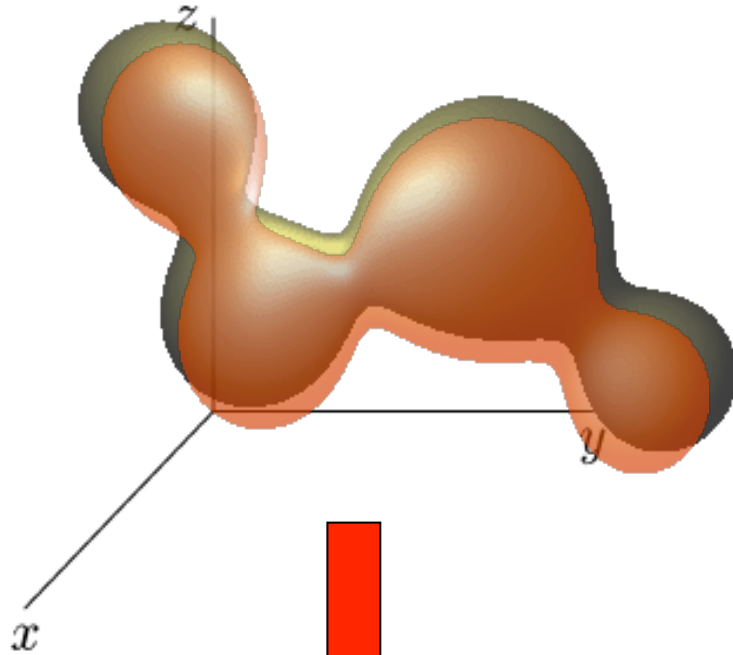
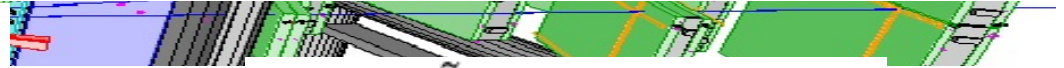
$$\begin{aligned} \text{Weights : } w_i \\ = \\ P_{\text{target}}(\text{var}_i) / \\ p_{\text{source}}(\text{var}_i) \end{aligned}$$



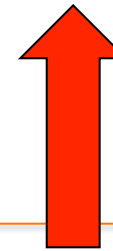
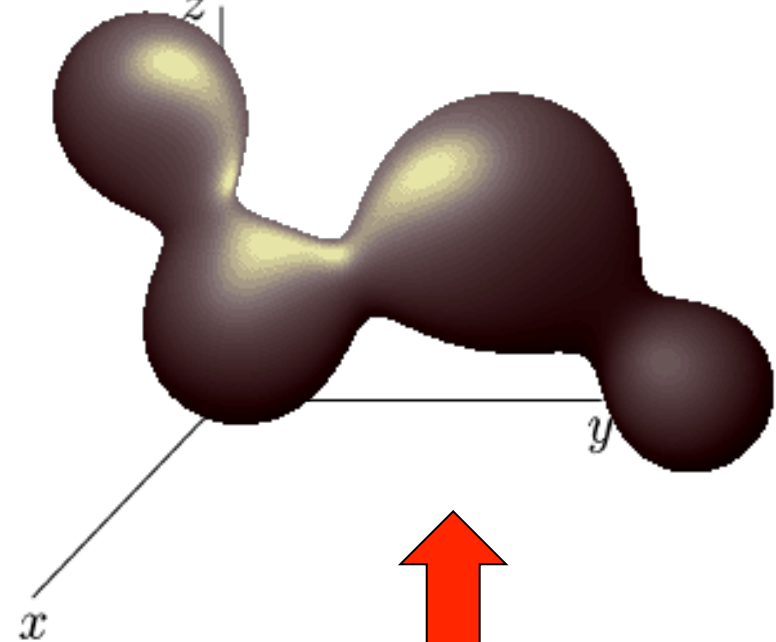
- What if multi-dimension ?
- Usually : reweight separately on 1D projections, at best 2D, because of quick lack of statistics
- Can we do better ?

Multidimension reweighting

See demo on Andrei Rogozhnikov github



$$\text{Weights : } w_i = \frac{P_{\text{target}}(\text{score}_i)}{P_{\text{source}}(\text{score}_i)}$$



Multi dimensional reweighting (2)



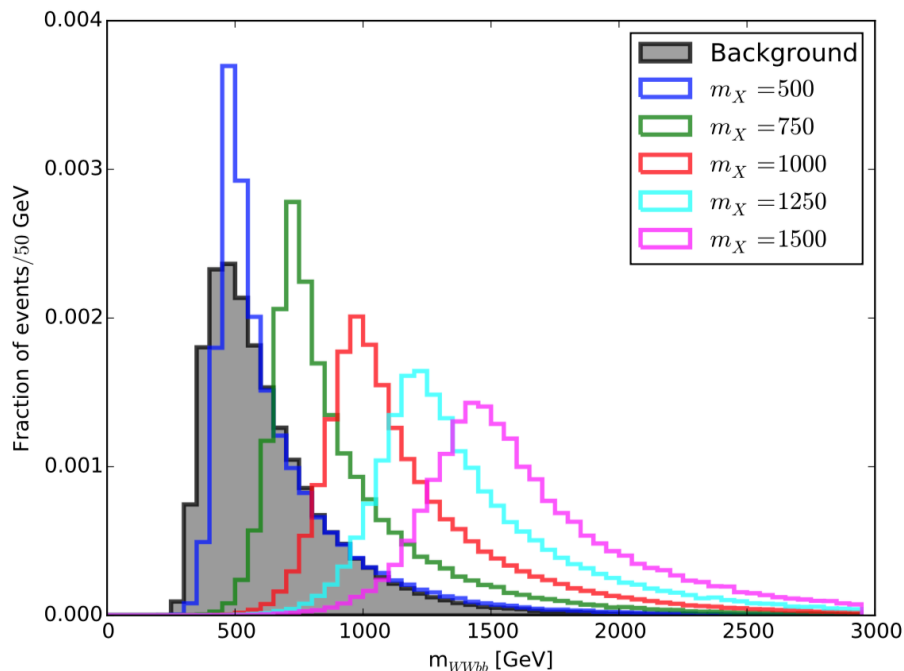
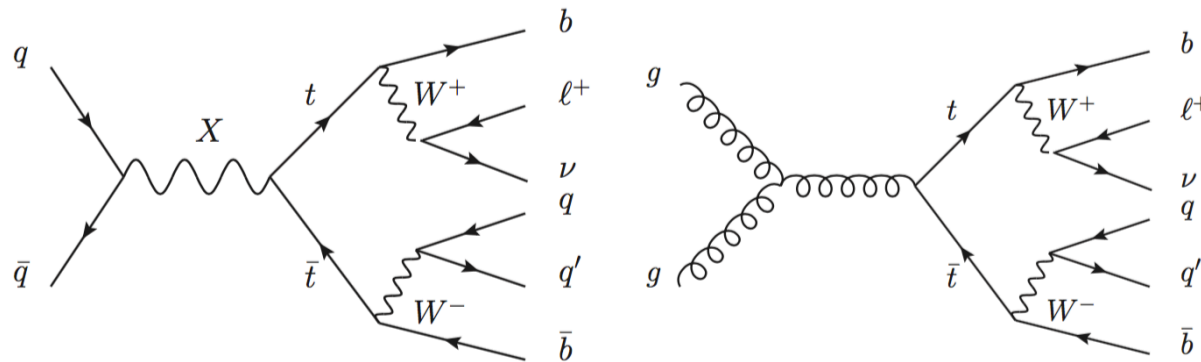
- ❑ Reweighting the Source distribution on the score allows multidimensional reweighting without statistics problem
- ❑ Usual caveat still hold : Target support should be included in Source support, distributions should not be too different otherwise unmanageable very large or very small weights
- ❑ (Note : “reweighting” in HEP language \Leftrightarrow “importance sampling” in ML language)

ML in analysis



Parameterised learning

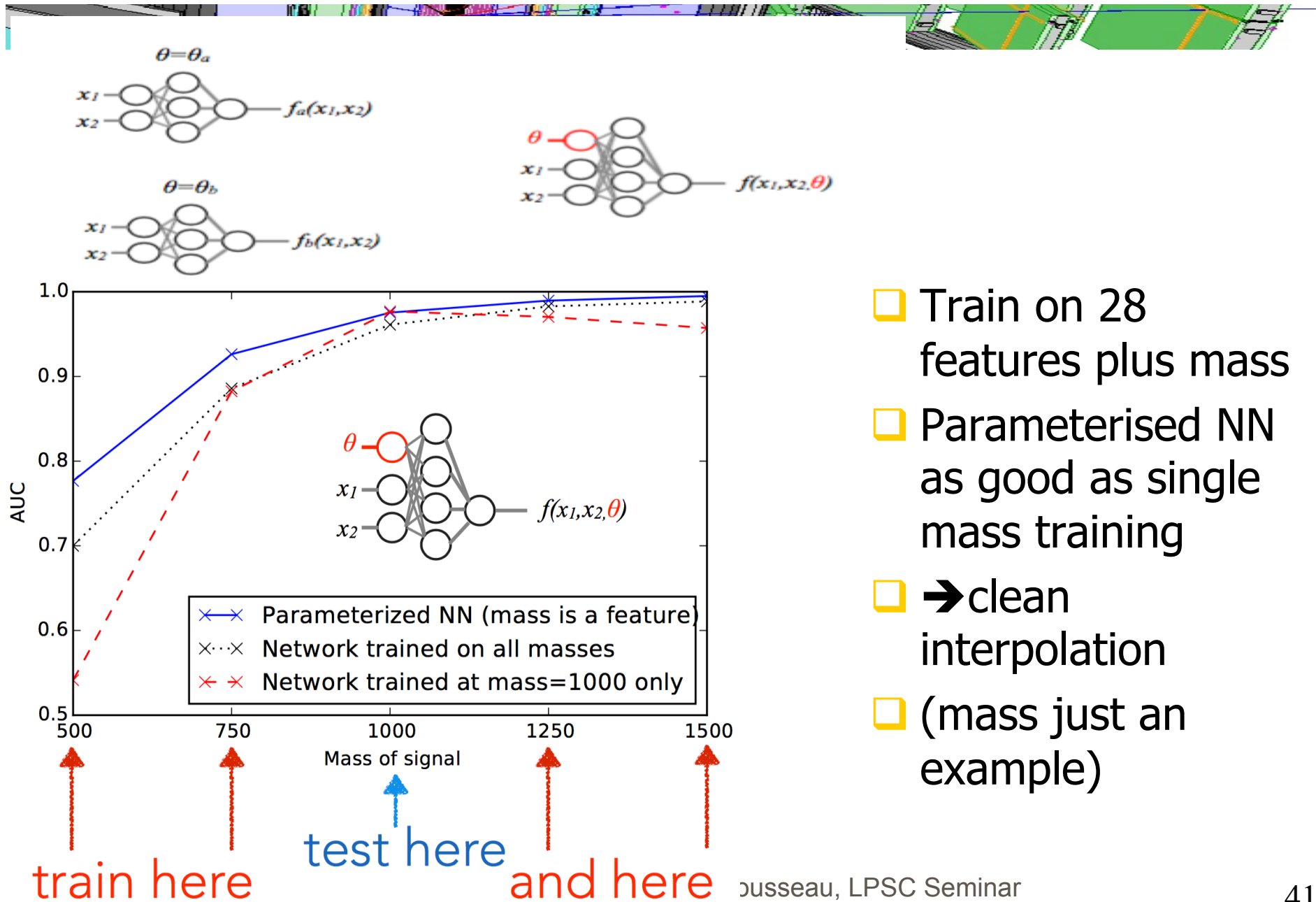
[1601.07913](#) Baldi, Cranmer, Faucett, Sadowksi, Whiteson



□ Typical case: looking for a particle of unknown mass

□ E.g. here $t\bar{t}$ decay

Parameterised learning (2)



- Train on 28 features plus mass
- Parameterised NN as good as single mass training
- → clean interpolation
- (mass just an example)

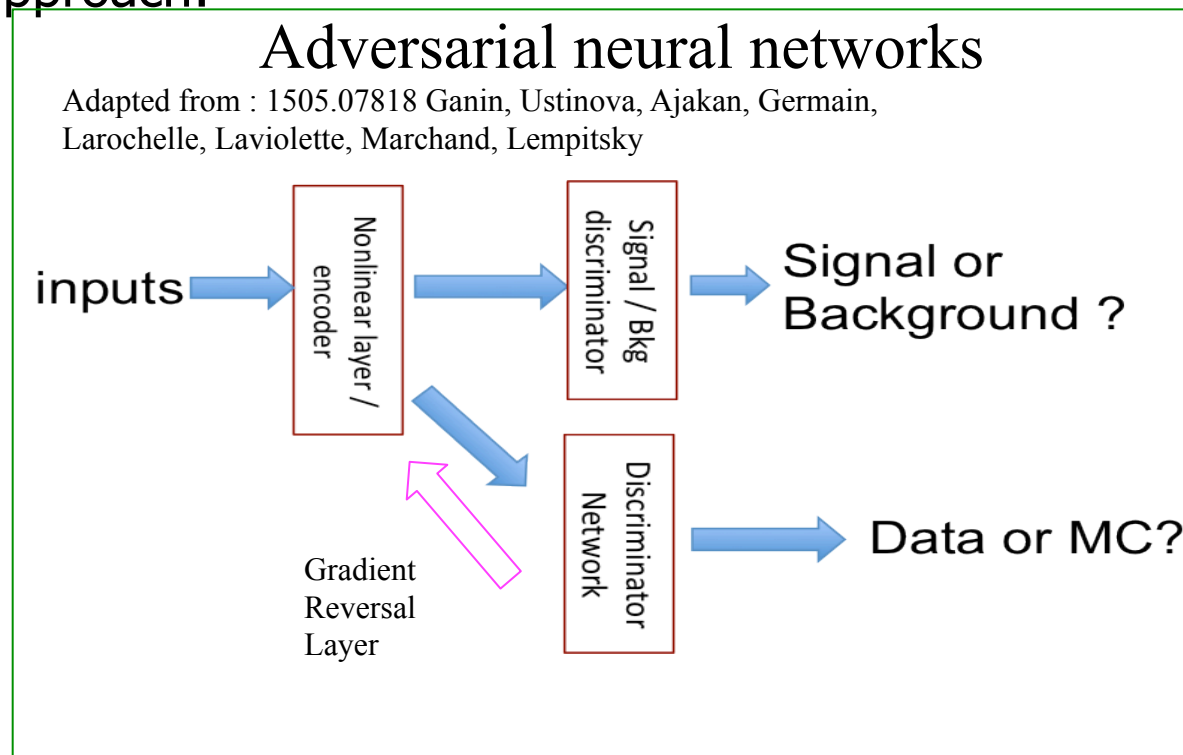
Systematics



- ❑ Our experimental papers typically ends with
 - measurement = $m \pm \sigma(\text{stat}) \pm \sigma(\text{syst})$
 - $\sigma(\text{syst})$ systematic uncertainty : known unknowns, unknown unknowns...
- ❑ Name of the game is to minimize quadratic sum of :
$$\sigma(\text{stat}) \pm \sigma(\text{syst})$$
- ❑ ML techniques used so far to minimise $\sigma(\text{stat})$
- ❑ Impact of ML on $\sigma(\text{syst})$ or even better global optimisation of $\sigma(\text{stat}) \pm \sigma(\text{syst})$ is an open problem
- ❑ Worrying about $\sigma(\text{syst})$ untypical of ML in industry

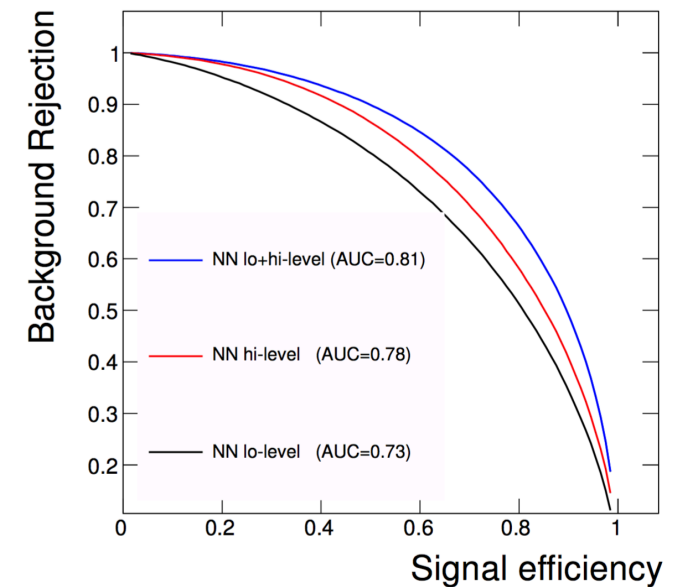
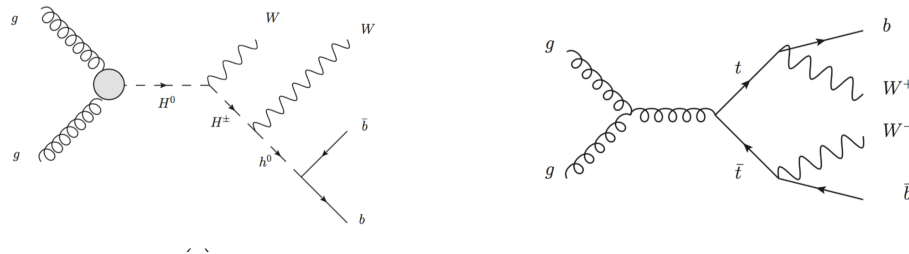
Systematics (2)

- ❑ 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
- ❑ One possible approach:

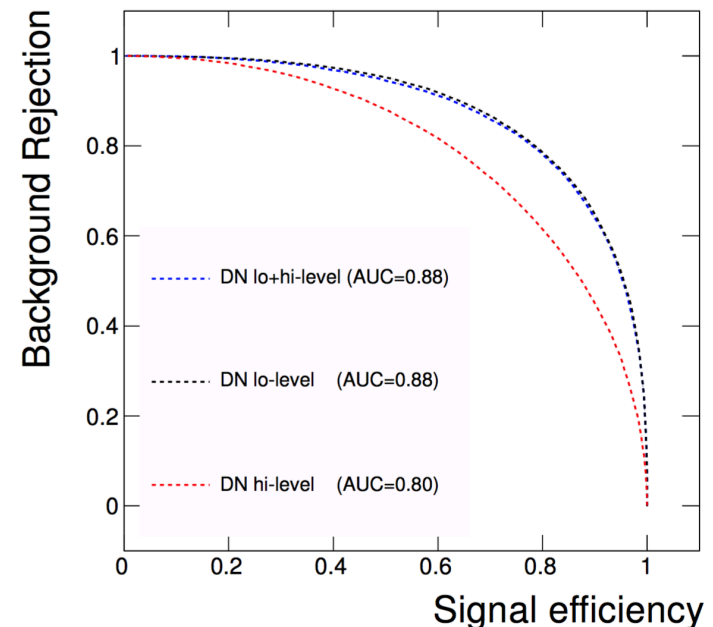


Deep learning for analysis

1402.4735 Baldi, Sadowski, Whiteson



- ❑ MSSM at LHC : $H^0 \rightarrow WWbb$ vs $tt \rightarrow WWbb$
- ❑ Low level variables:
 - 4-momenta
- ❑ High level variables:
 - Pair-wise invariant masses
- ❑ Deep NN outperforms NN, and does not need high level variables
- ❑ DNN learns the physics ?



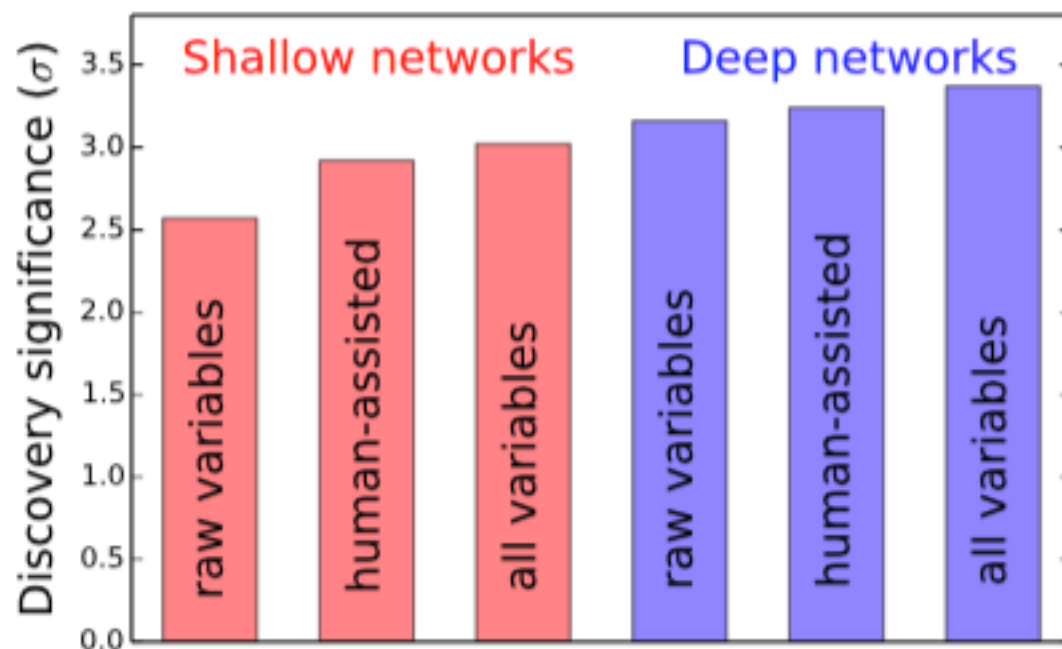
Deep learning for analysis (2)

1410.3469 Baldi Sadowski Whiteson



□ H tautau analysis at LHC: $H \rightarrow \tau\tau$ vs $Z \rightarrow \tau\tau$

- 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

□ ~ 10 M events used for training (> 10 full G4 simulation in ATLAS)

ML in reconstruction

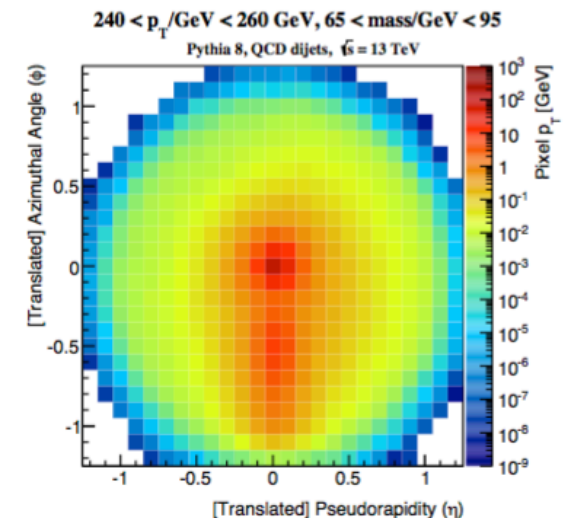
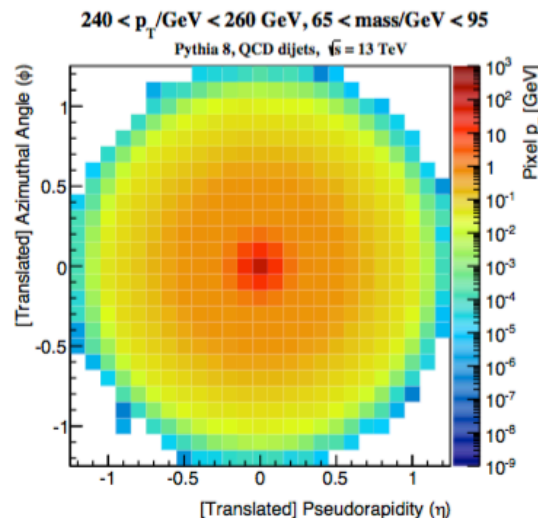
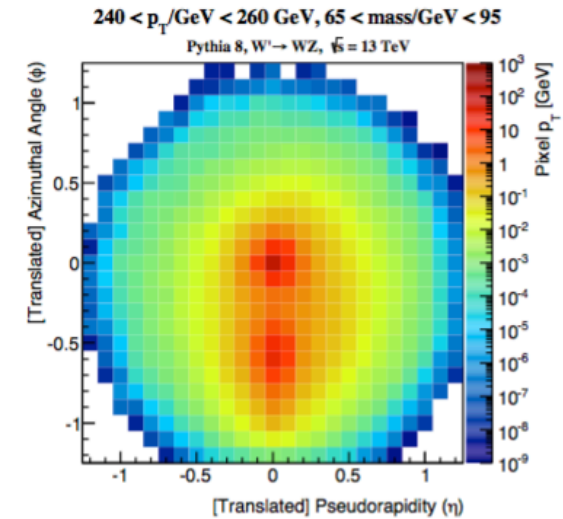
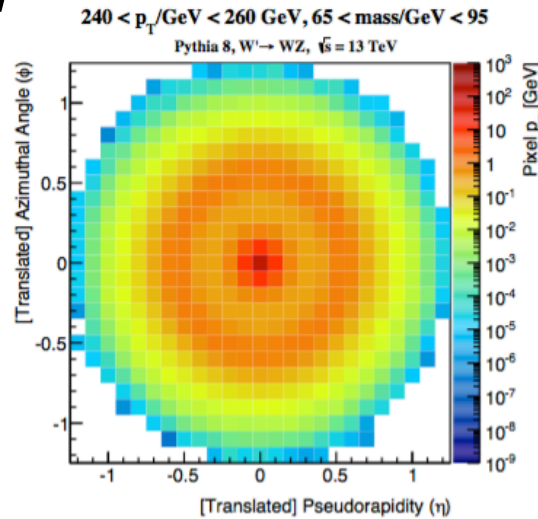
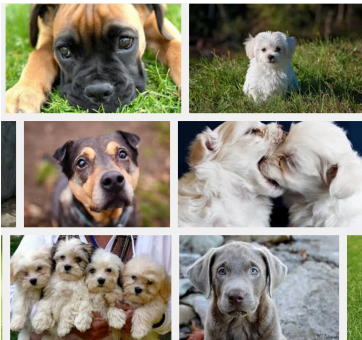


Jet Images

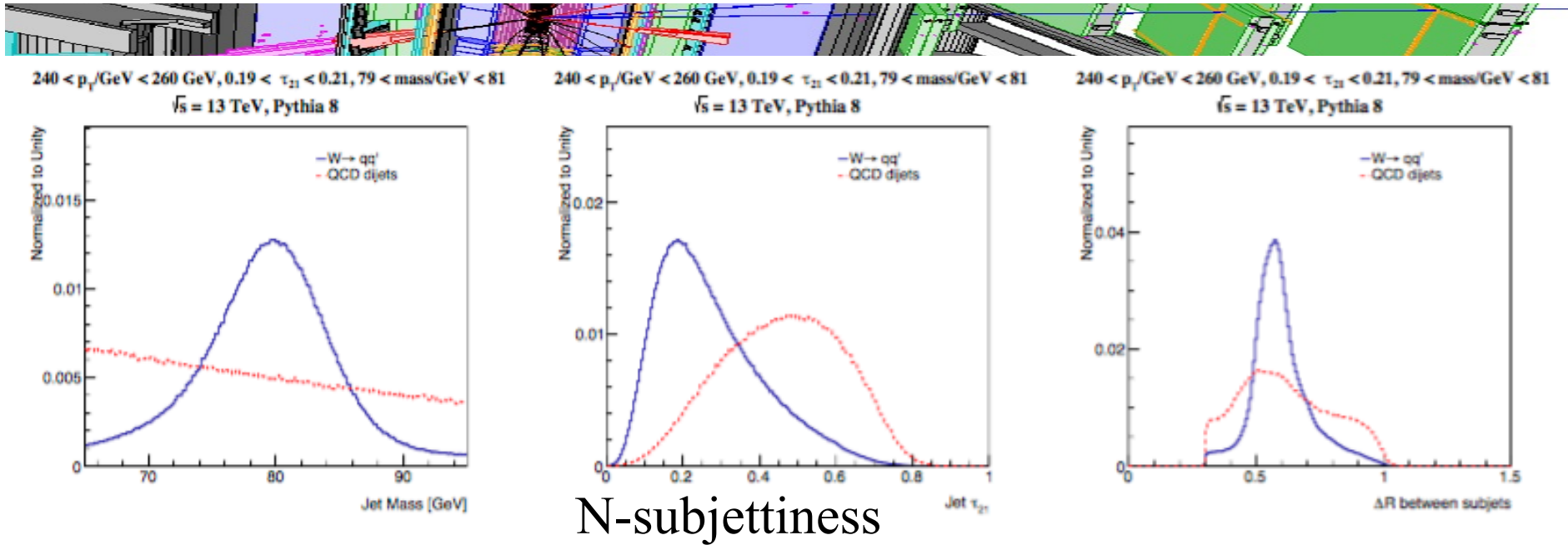
[arXiv 1511.05190](https://arxiv.org/abs/1511.05190) de Oliveira, Kagan, Mackey, Nachman, Schwartzman



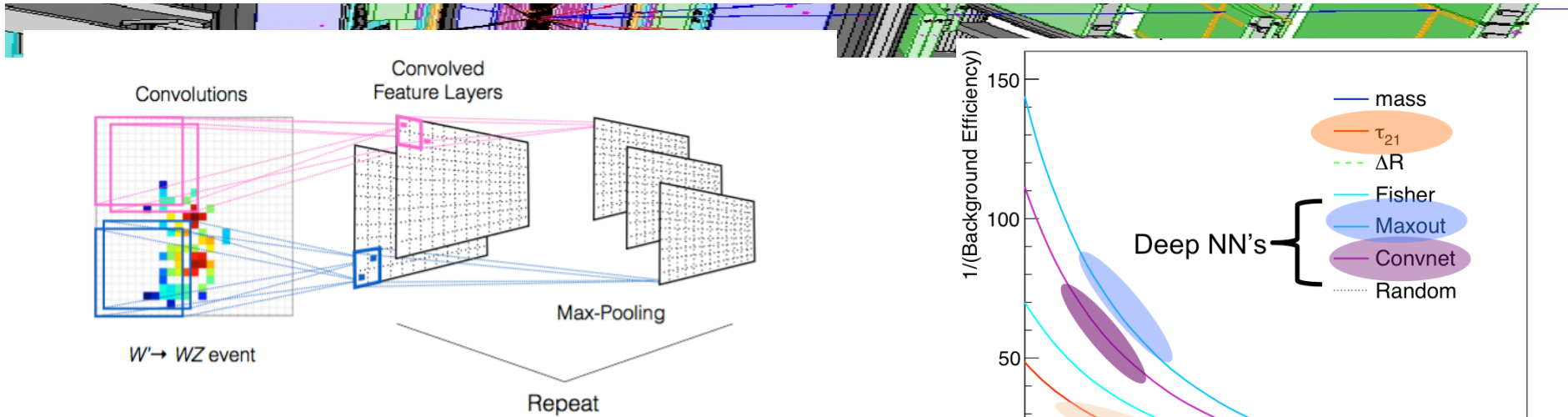
- Distinguish boosted W jets from QCD
- Particle level simulation
- Average images:



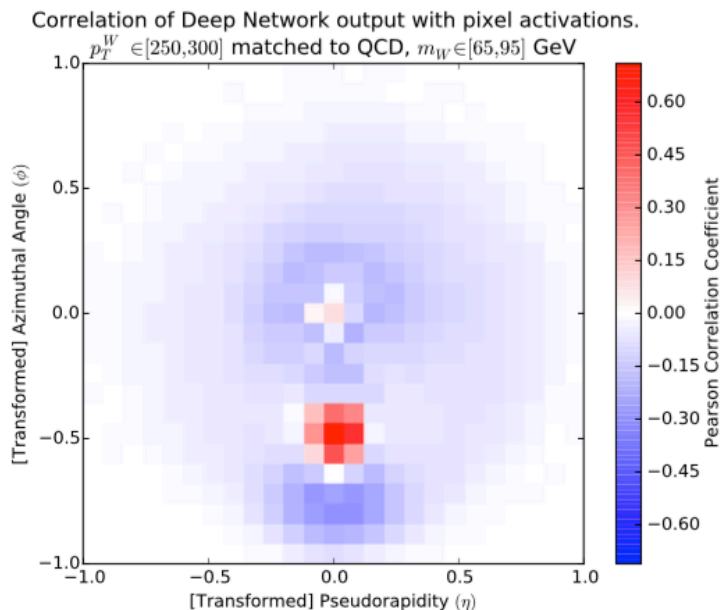
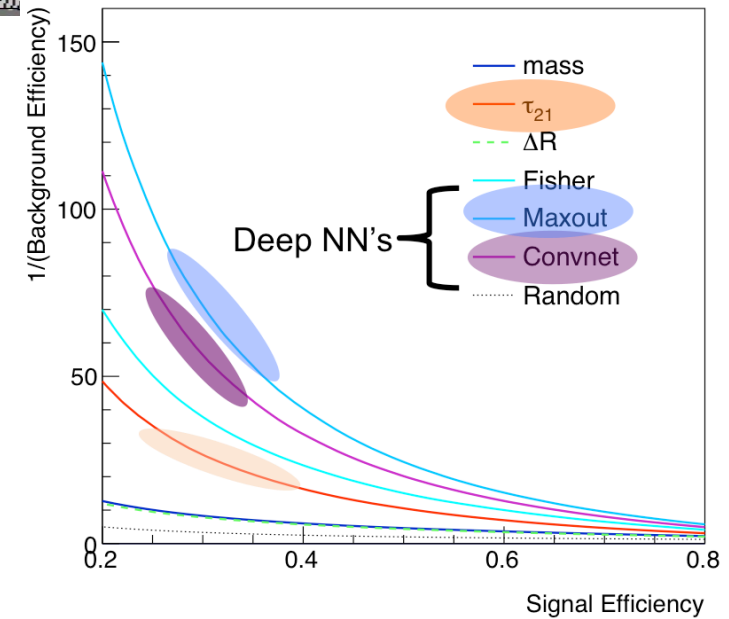
Boosted jets : standard variables



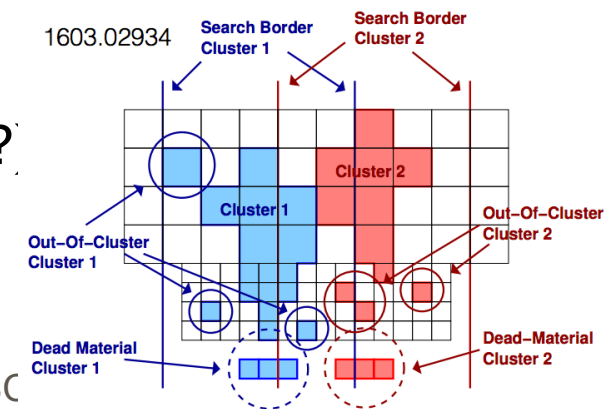
Jet Images : Convolution NN



Variables build from CNN outperform the more usual ones



- What the CNN sees (the "cat" neurone")
- Now need proper detector and pileup simulation
- 3Dimension (calo depth as a color?)

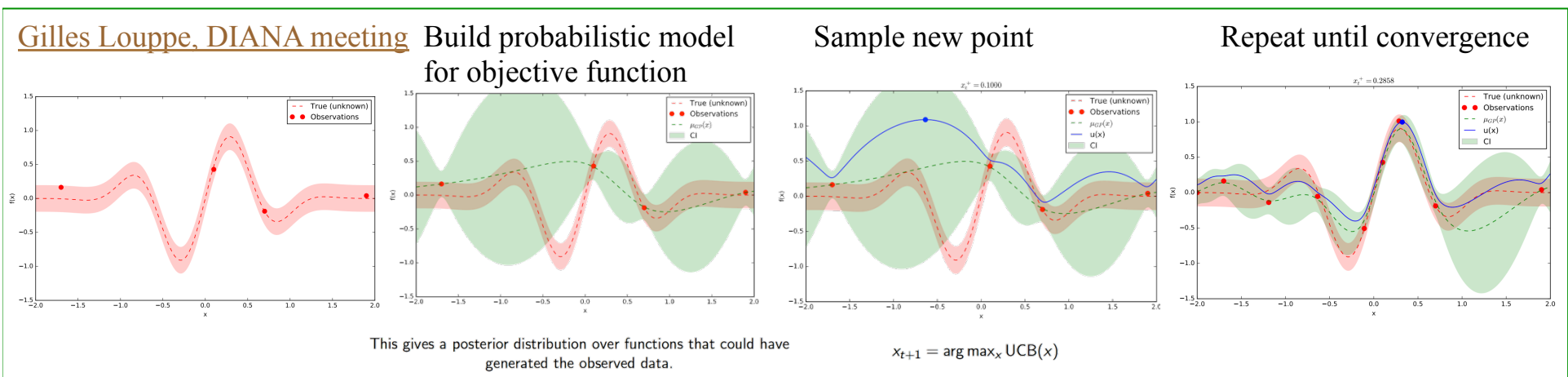


L in HEP, David Rousseau, LPSC

ML in Simulation



- ❑ We invest a lot of resources (CPU: $\sim 100k$ cores/experiment *year, human) on very fine tuned simulations:
 - so far very manual optimisation by super experts
 - optimisation in many dimensions parameter space, with costly evaluation
- ❑ Now turning to more modern techniques e.g.:
 - Bayesian Optimization and Gaussian Processes



- ❑ Another avenue : multivariable regression to parameterise detector response
- ❑ By the way : Bayesian Optimisation can also be used to optimised analysis

Data Challenges



Challenges (competition)



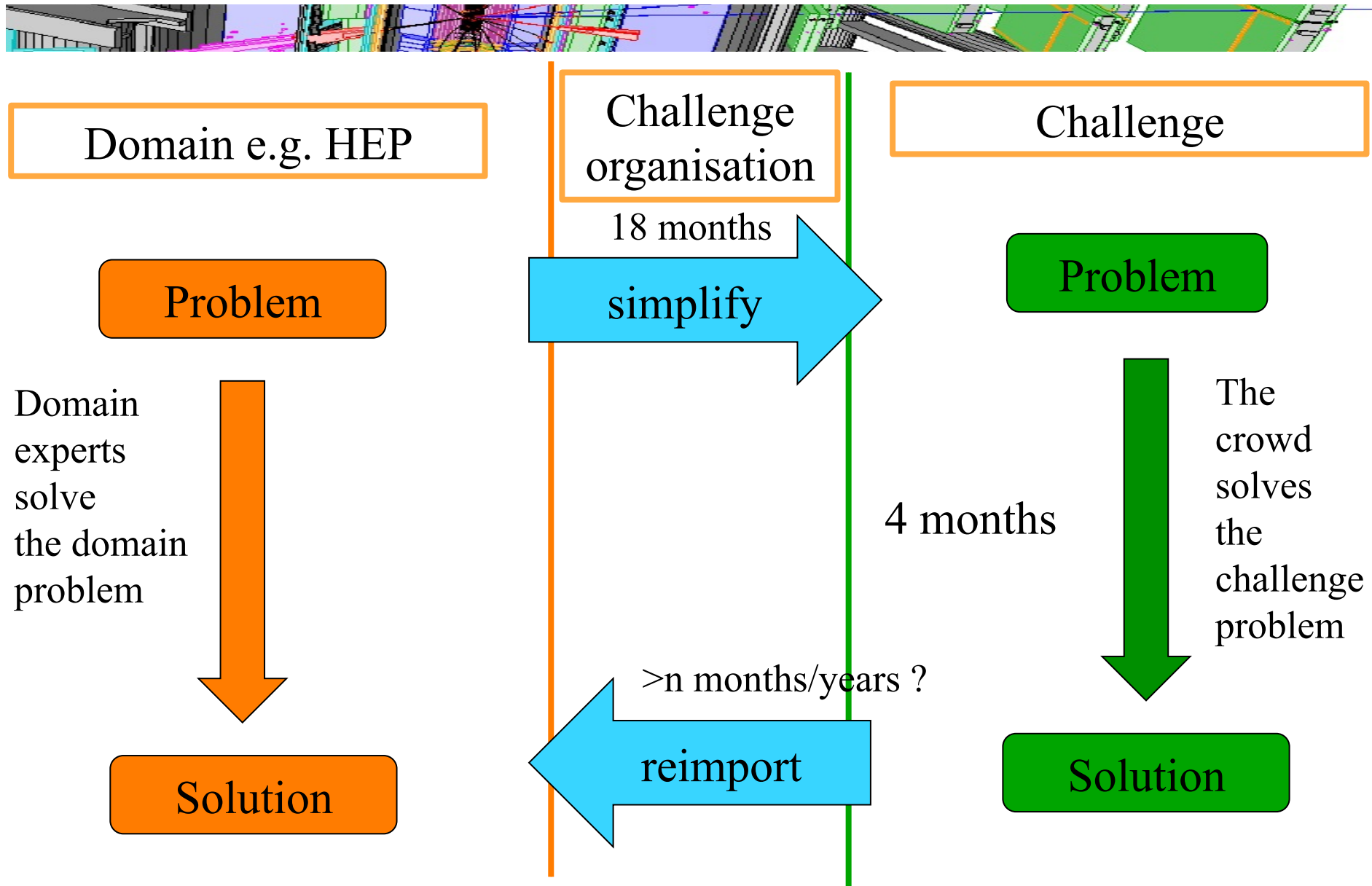
- ❑ Challenges are essentially a way to create a buzz around an open dataset dressed with a benchmark
 - HiggsML (ATLAS) 2014
 - FlavourML (LHCb) 2015
 - future TrackML (ATLAS+CMS) 2017
- ❑ Buzz in non-HEP world to get the attention of ML specialists

HiggsML in a nutshell



- ❑ Why not put some ATLAS simulated data on the web and ask data scientists to find the best machine learning algorithm to find the Higgs ?
 - Instead of HEP people browsing machine learning papers, coding or downloading possibly interesting algorithm, trying and seeing whether it can work for our problems
- ❑ Challenge for us : make a full ATLAS Higgs analysis simple for non physicists, but not too simple so that it remains useful
- ❑ Also try to foster long term collaborations between HEP and ML

From domain to challenge and back



HiggsML : Committees



❑ Organization committee:

- ATLAS
 - David Rousseau : Atlas-LAL
 - Claire Adam-Bourdarios : Atlas-LAL (outreach, legal matter)
 - Glen Cowan : Atlas-RHUL (statistics)
- Machine Learning
 - Balazs Kegl : Appstat-LAL
 - Cécile Germain : TAO-LRI
 - Isabelle Guyon : Chalearn (now chaire Paris Saclay) (challenges organisation)

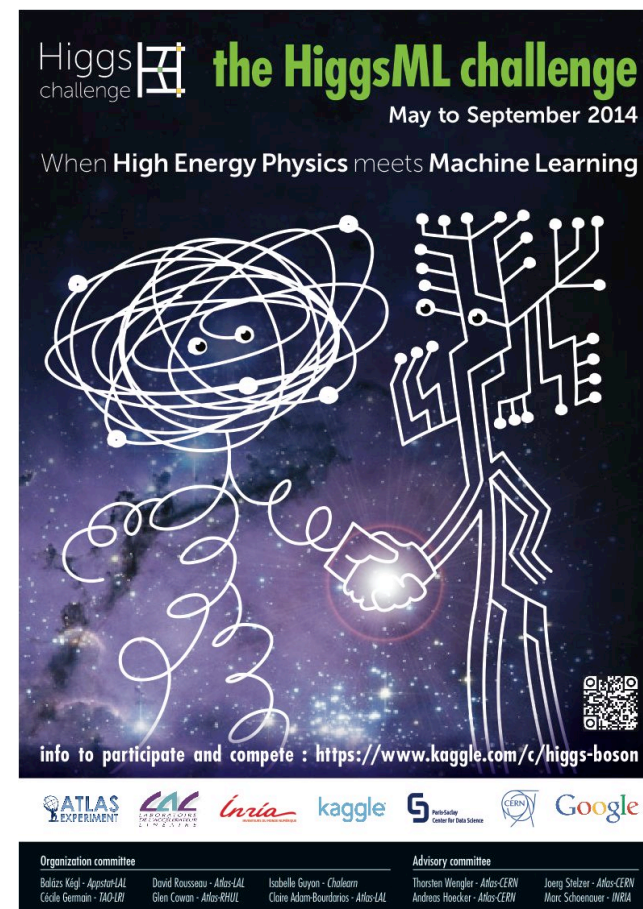
❑ Advisory committee:

- Andreas Hoecker : Atlas-CERN (PC, TMVA)
- Joerg Stelzer : Atlas-CERN (TMVA)
- Thorsten Wengler : Atlas-CERN (ATLAS management)
- Marc Schoenauer : INRIA

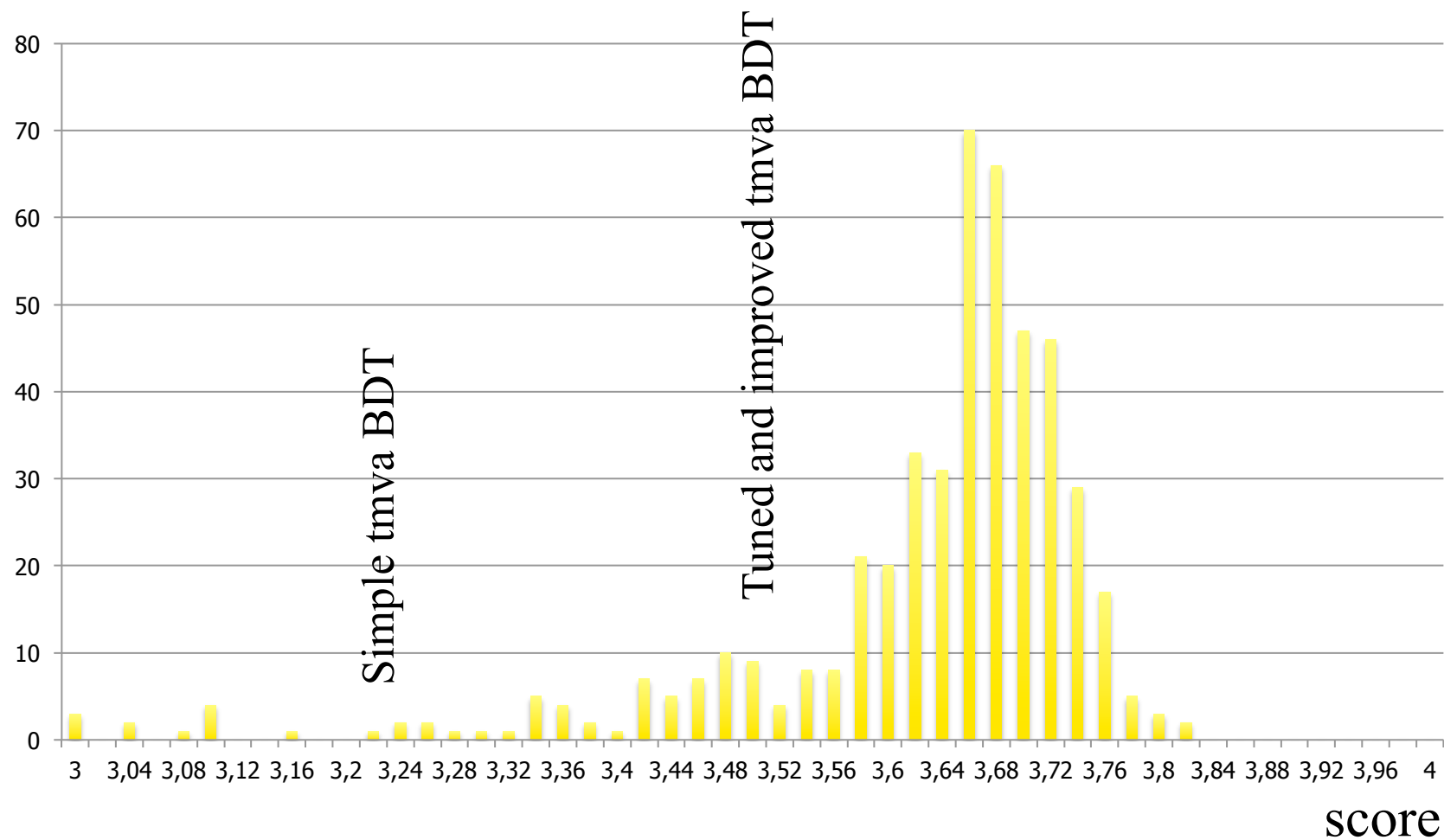
Higgs Machine learning challenge

- ❑ See [talk DR CTD2015 Berkeley](#)
- ❑ An ATLAS Higgs signal vs background classification problem, optimising statistical significance
- ❑ Ran in summer 2014
- ❑ 2000 participants (largest on Kaggle at that time)
- ❑ Outcome
 - Best significance 20% than with Root-TMVA
 - BDT algorithm of choice in this case where number variables and number of training events limited (NN very slightly better but much more difficult to tune)
 - XGBoost best BDT on the market (quite wide spread nowadays)
 - Wealth of ideas, documented in [JMLR proceedings v42](#)
 - Still working on what works in real life what does not
 - Raised awareness about ML in HEP
- ❑ Also:
 - Winner Gabor Melis hired by DeepMind
 - Tong He, co-developper of XGBoost, winner of special “HEP meets ML” price got a PhD grant and US visa

Advances of ML in HEP, David Rousseau, LPSC Seminar



Best private scores



LHCb : flavour of physics

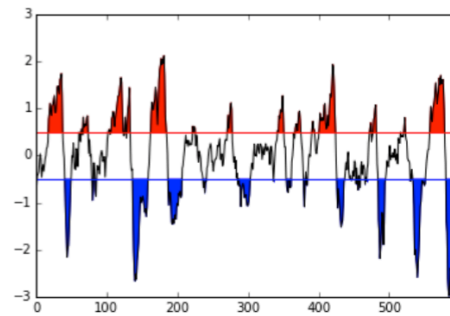
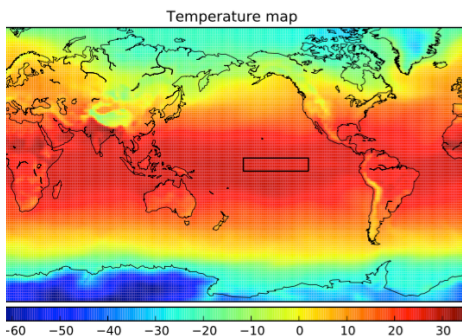
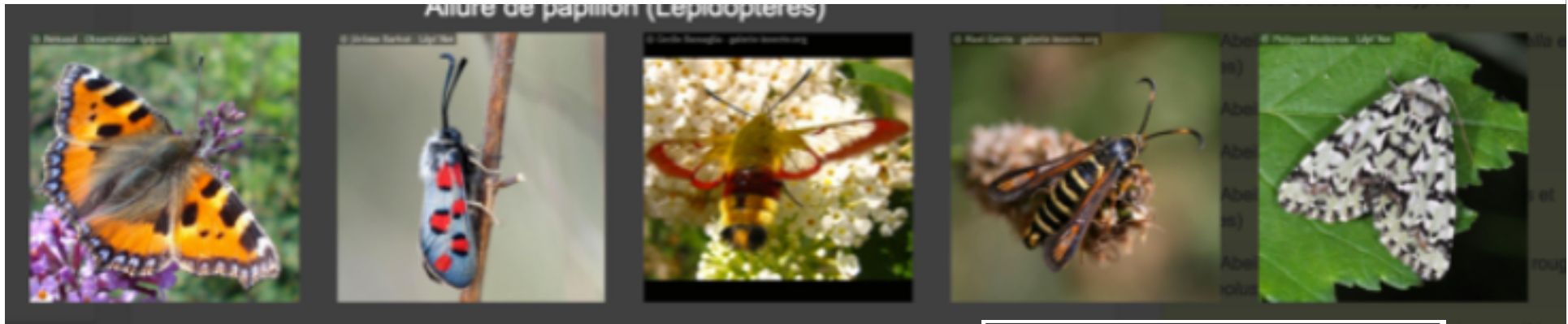
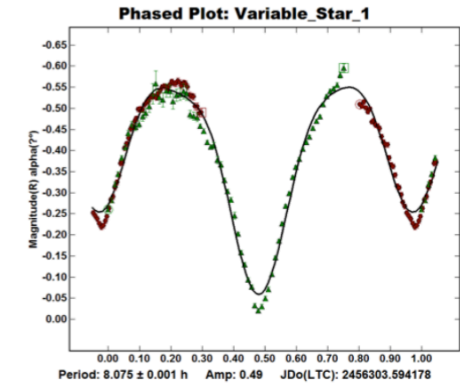


- ❑ LHCb organised in summer 2015 another challenge “flavour of physics”: search for LFV decay $\tau \rightarrow \mu\mu\mu$
- ❑ similar to HiggsML, with a big novelty:
 - some variables known to be poorly described by MC
 - algorithm had to behave similarly on data and MC in a control region $D0 \rightarrow K\pi\pi$
- ❑ ➔ Nice idea, however, never underestimates the machine learners: They devised an algorithm which
 - was able to distinguish control region from signal region
 - was behaving well (data=MC) in the control region
 - but was recklessly abusing the data/MC difference in the signal region
- ❑ ➔ rules had to be changed in the middle of the challenge to disallow this
- ❑ Anyway, this does show that systematics is tricky to handle

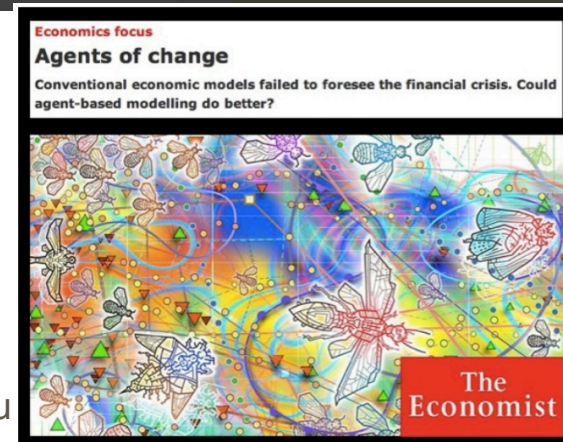
Beyond challenges : RAMP



- ❑ (Already mentioned for Anomaly Detection)
- ❑ Run by CDS Paris Saclay
- ❑ Main difference wrt to HiggsML:
 - participants post their software, which is run by the RAMP platform
 - one day hackathon
 - participants are encouraged to re-use other people's software
- ❑ Can adapt to all domains:



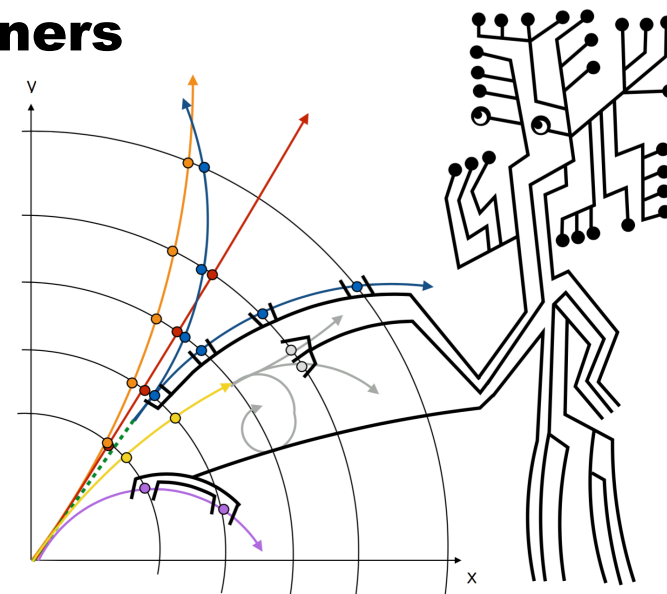
Advances of ML in HEP, David Rousseau



Towards a Future Tracking Machine Learning challenge



**A collaboration between ATLAS and CMS physicists,
and Machine Learners**

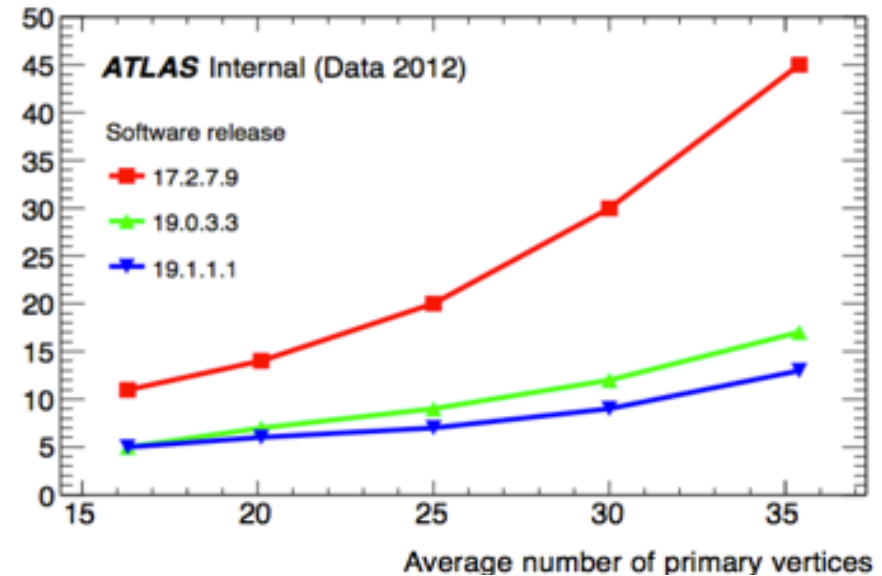


TrackML : Motivation 1

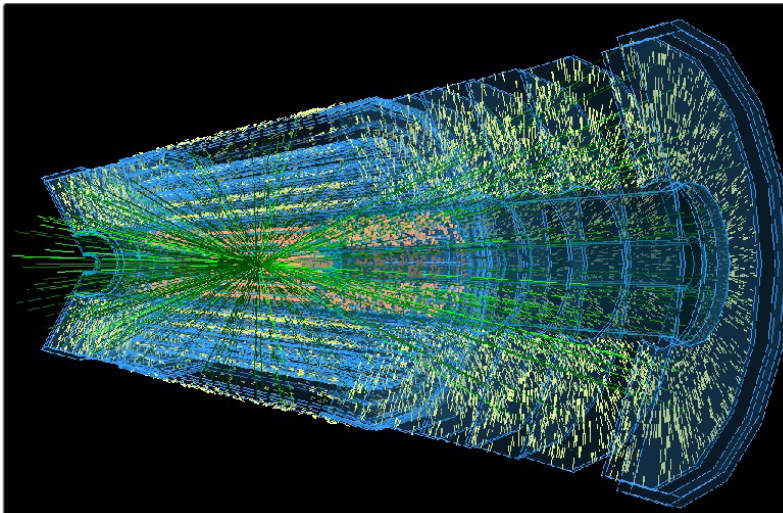


Graeme Stewart ECFA HL-LHC workshop 2014

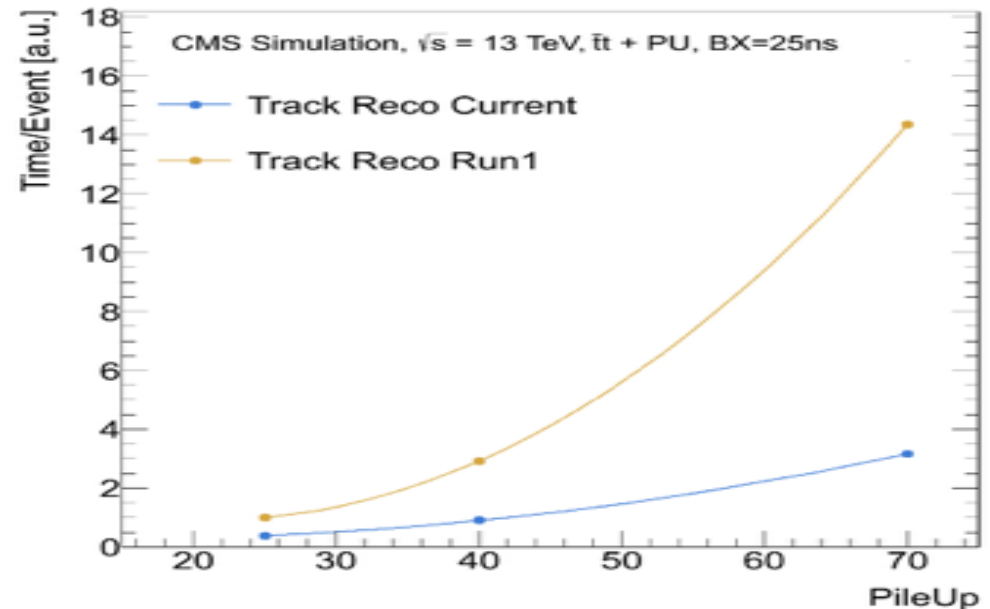
- ❑ See details [DR talk at CTD2016](#)
- ❑ Tracking (in particular pattern recognition) dominates reconstruction CPU time at LHC
- ❑ HL-LHC (phase 2) perspective : increased pileup :
 - Run 1 (2012): $\langle n \rangle \sim 20$
 - Run 2 (2015): $\langle n \rangle \sim 30$
 - Phase 2 (2025): $\langle n \rangle \sim 150$
- ❑ CPU time quadratic/exponential extrapolation (difficult to quote any number)



➔
150



Advances of ML in HE



TrackML : Motivation 2



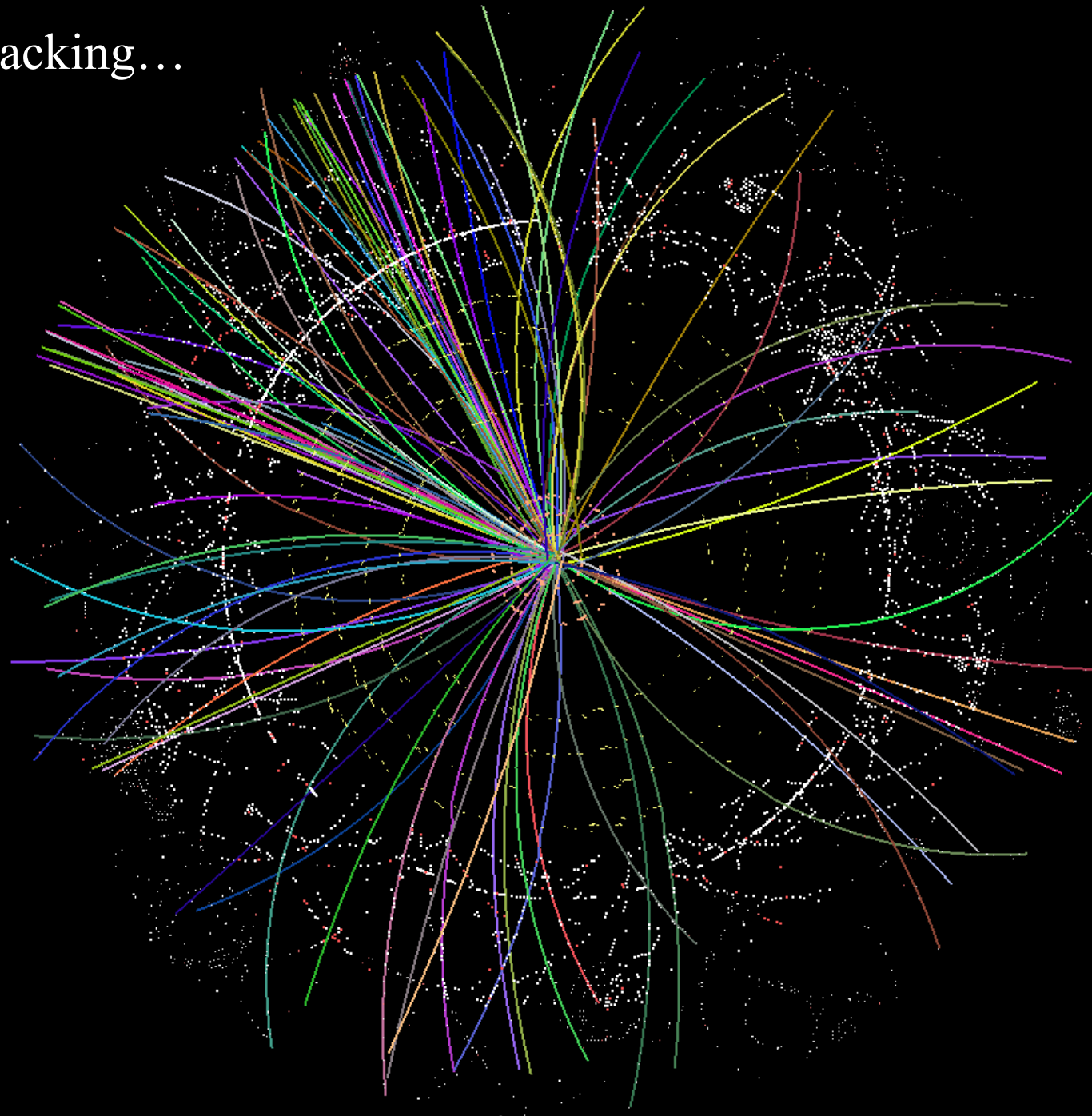
- ❑ LHC experiments future computing budget flat (at best)
- ❑ Installed CPU power per \$=€=CHF expected increase factor ~ 10 in 10 years
- ❑ Experiments plan on increase of data taking rate ~ 10 as well ($\sim 1\text{kHz}$ to 10kHz)
- ❑ \rightarrow HL reconstruction at $\mu=150$ need to be as fast as Run1 reconstruction at $\mu=20$
- ❑ \rightarrow requires very significant software improvement, factor 10-100
- ❑ 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)
- ❑ Need to engage a wide community to tackle this problem

TrackML : engaging Machine Learners

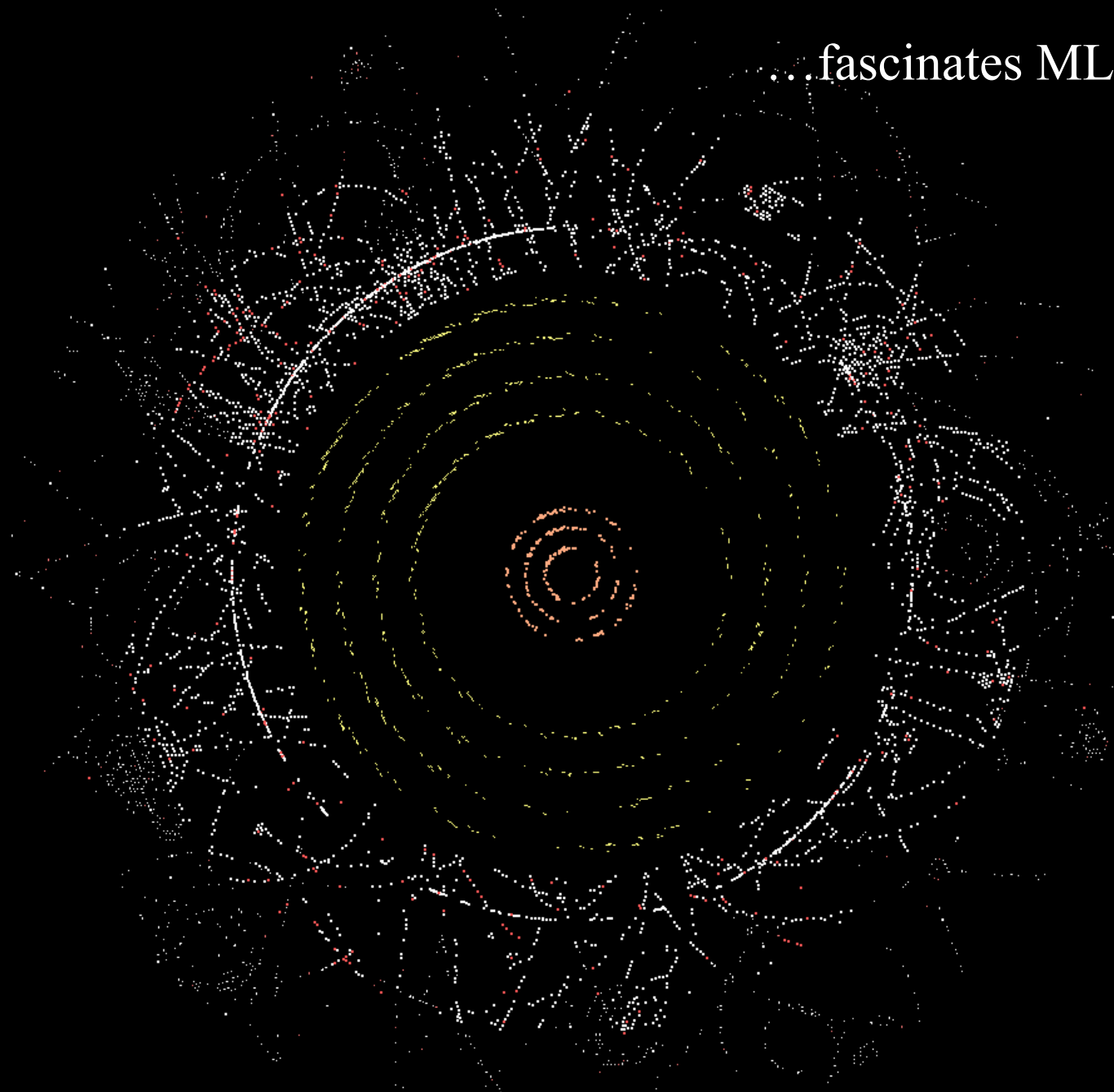


- ❑ Suppose we want to improve the tracking of our experiment
- ❑ We read the literature, go to workshops, hear/read about an interesting technique (e.g. ConvNets, MCTS...). Then:
 - Try to figure by ourself what can work, and start coding → traditional way
 - Find an expert of the new technique, have regular coffee/beer, get confirmation that the new technique might work, and get implementation tips → better
- ❑ ...repeat with each technique...
- ❑ Much much better:
 - Release a data set, with a benchmark, and have the expert do the coding him/herself
 - → he has the software and the know-how so he'll be (much) faster even if he does not know anything about our domain at the beginning
 - → engage multiple techniques and experts simultaneously (e.g. 2000 people participated to the Higgs Machine Learning challenge) in a comparable way
 - → even better if people can collaborate
 - → a challenge is a dataset with a benchmark and a buzz
 - Looking for long lasting collaborations beyond the challenge
- ❑ Focus on the pattern recognition : release list of 3D points, challenge is to associate them into tracks fast. Use public release of ATLAS tracking (ACTS) as a simulation engine and starting kit

HEP tracking...



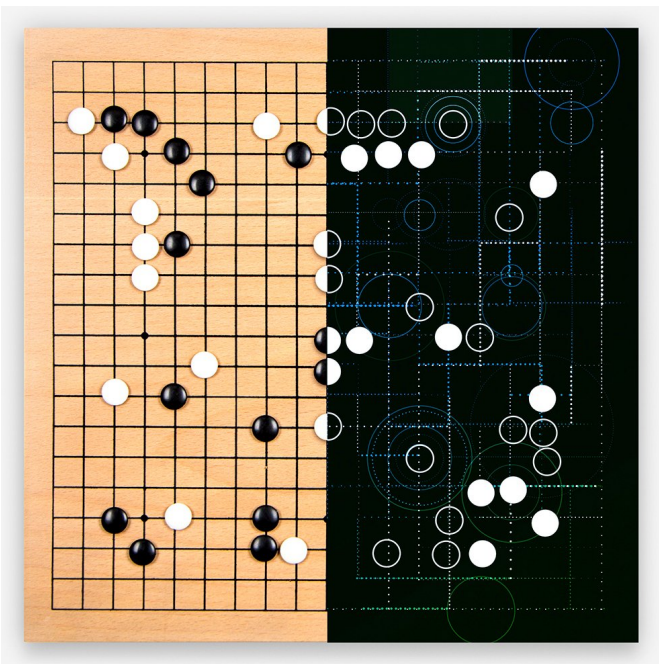
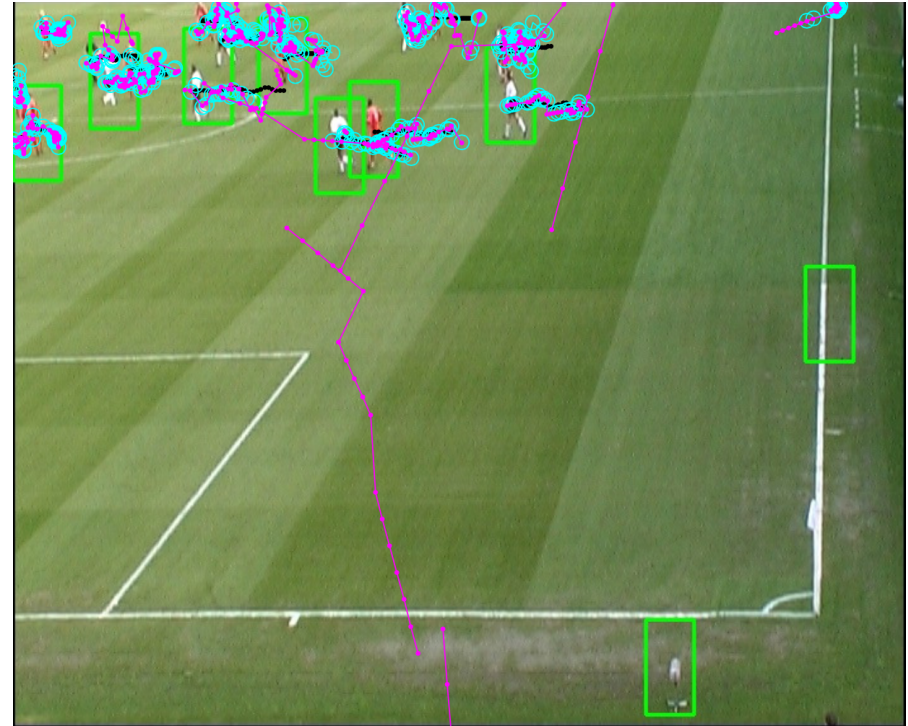
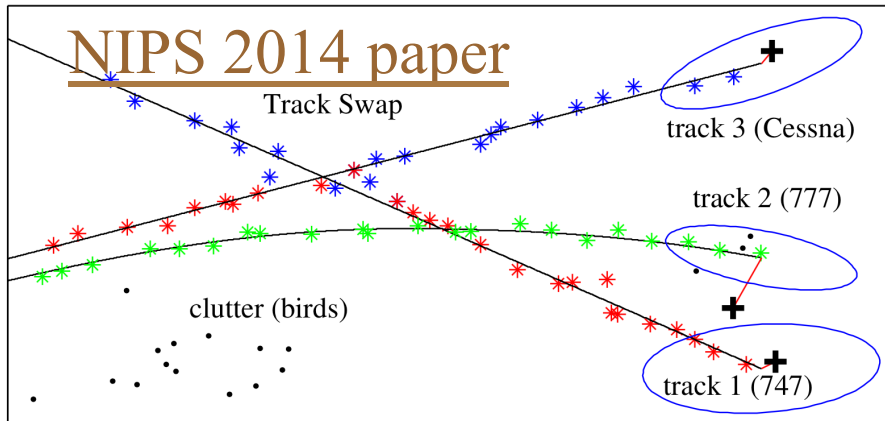
...fascinates ML experts



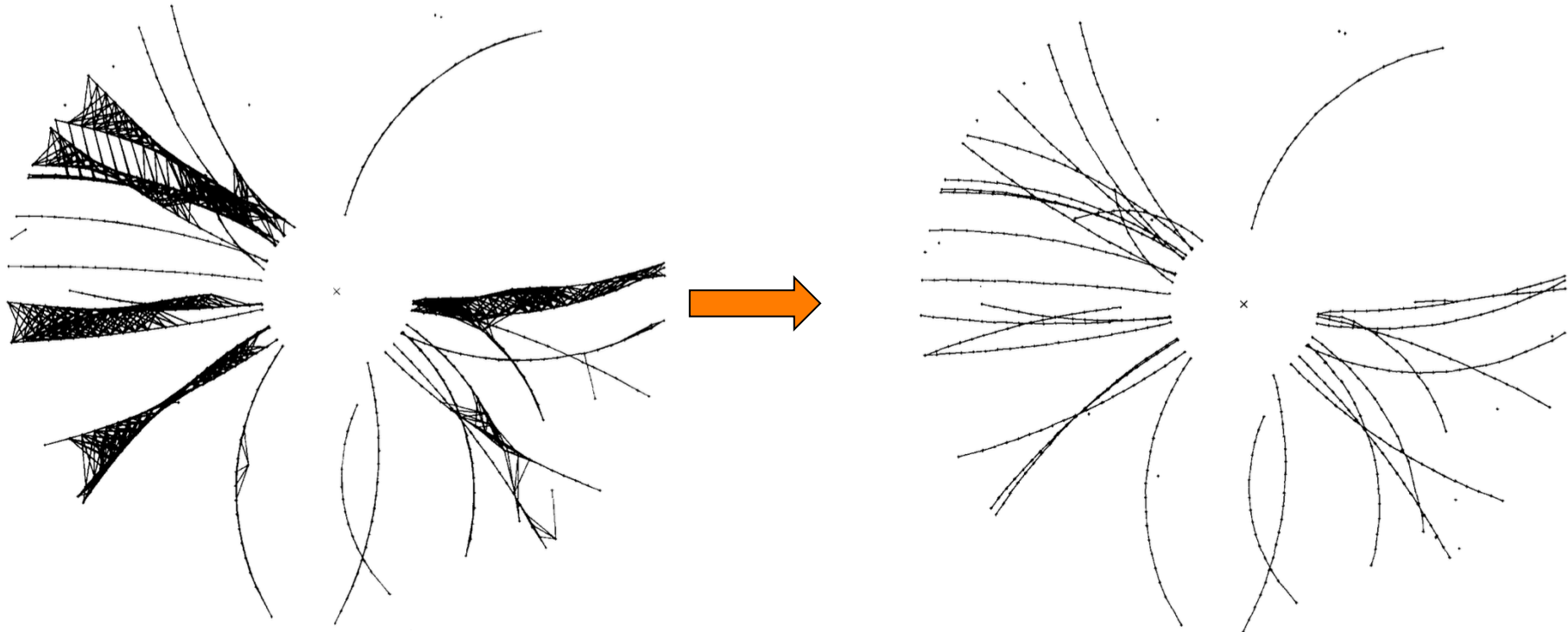
Pattern recognition



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



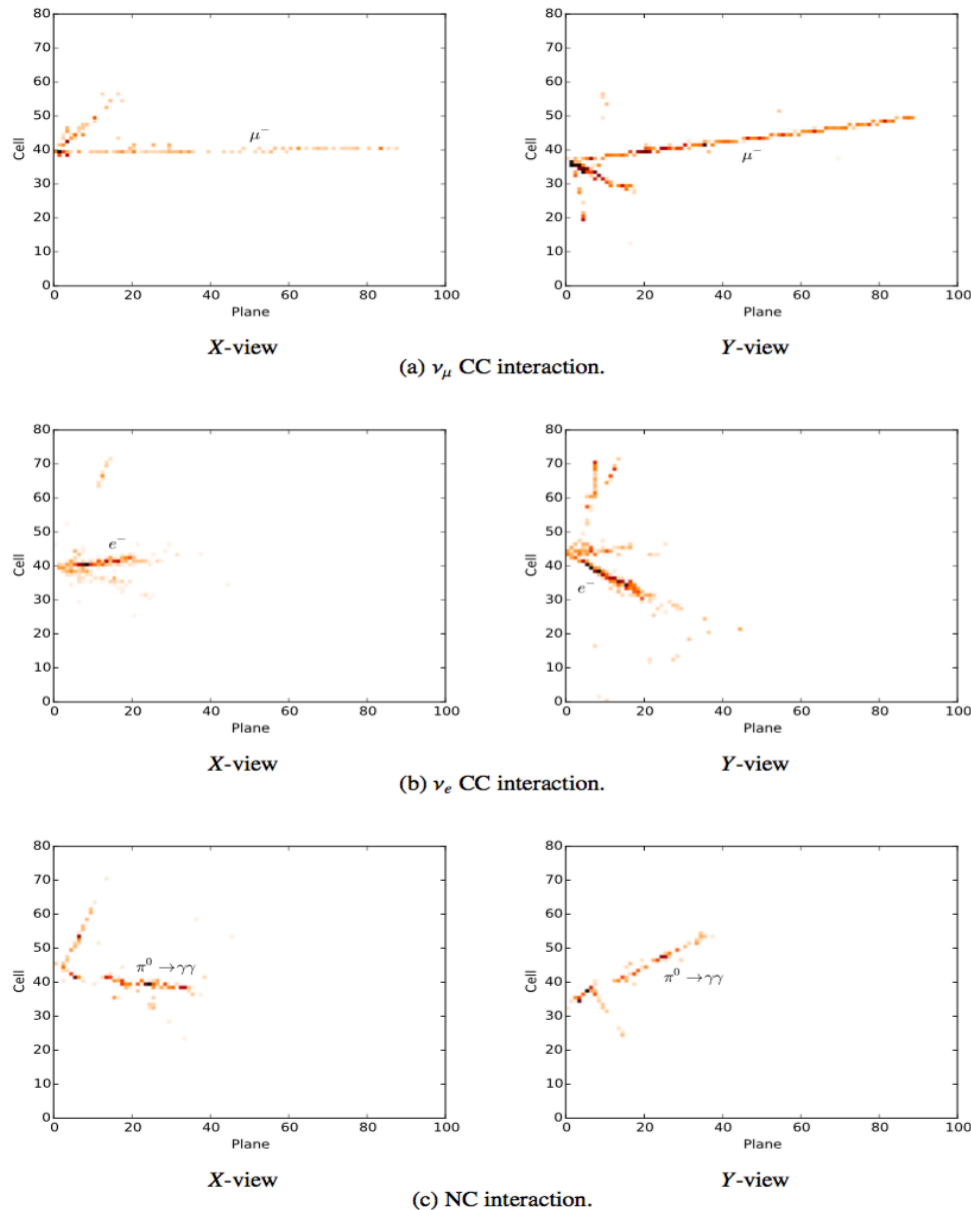
TrackML : An early attempt



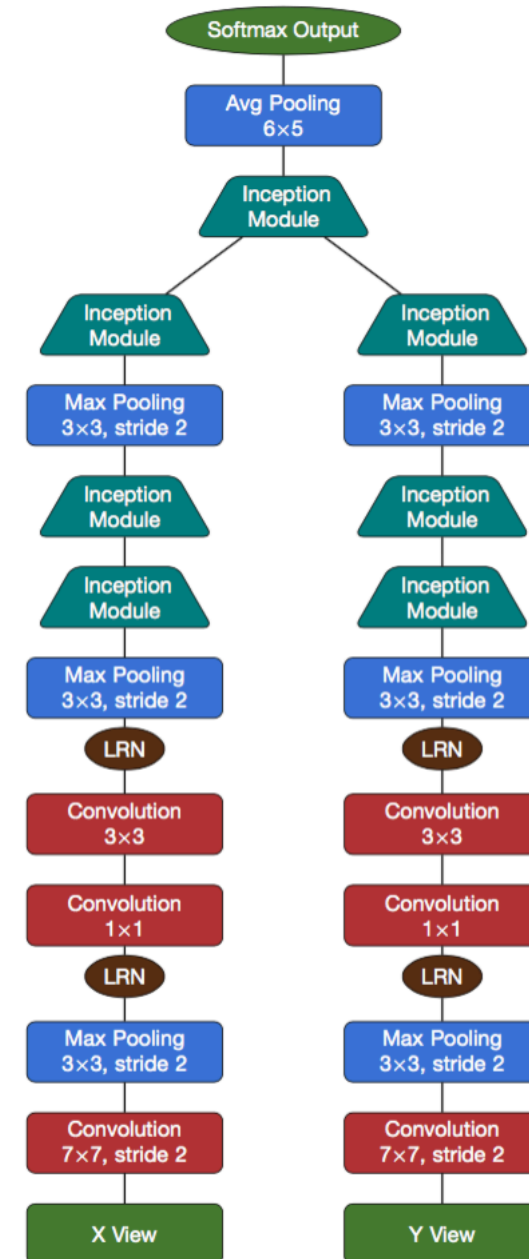
- ❑ Stimpfl-Abele and Garrido (1990) (ALEPH)
- ❑ All possible neighbor connections are built, the correct ones selected by the NN (not used in production)
- ❑ Also PhD Vicens Gaitan 1993, winner of Flavour of Physics challenge

A recent attempt : NOVA

[arXiv 1604.01444](https://arxiv.org/abs/1604.01444) Aurisano et al



Neutrino interaction classification
Using Convolutional Neural Network



Rousseau, L. CC-BY-NC-ND 4.0 International license

Wrapping-up



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 (see later)
- ❑ Note : Yandex Data School of Analysis (with ~10 ML scientists) now a bona fide institute of LHCb
- ❑ Very useful/essential to build HEP - ML collaborations : study on shared dataset, thesis (Computer Science or HEP)
- ❑ Successful collaborations often within one campus
- ❑ Most likely there are friendly ML scientists in Grenoble

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:
 - We (ATLAS) initially saw its use for outreach purposes (CMS has been more open on releasing data)
 - But after all, ML collaboration is a kind of scientific outreach
 - ➔ATLAS uploaded there in 2015 the data from Higgs Machine Learning challenge (essentially 4-vectors from full G4 ATLAS simulation Higgs- \rightarrow tautau analysis)
 - ATLAS consider releasing more datasets dedicated to ML studies

Collection of links



- ❑ In addition to workshops mentioned in the first transparencies, and references mentioned in the talks
- ❑ Interexperiment Machine Learning group (IML) is gathering speed (documentation, tutorials, etc...). Topical monthly meeting.
- ❑ An internal ATLAS ML group has started in June 2016. Probably also in CMS ?
- ❑ IN2P3 School Of Statistics <http://sos.in2p3.fr> very good introduction
- ❑ <https://www.kaggle.com/c/higgs-boson>
- ❑ <https://higgsml.lal.in2p3.fr>
- ❑ [http://opendata.cern.ch/collection/ATLAS-Higgs-Challenge-2014:](http://opendata.cern.ch/collection/ATLAS-Higgs-Challenge-2014)
permanent home of the challenge dataset
- ❑ NIPS 2014 workshop agenda and **proceedings**
<http://jmlr.org/proceedings/papers/v42/>
- ❑ Mailing list opened to any one with an interest in both Data Science and High Energy Physics : HEP-data-science@googlegroups.com

Conclusion



- ❑ Machine Learning techniques widely used in HEP
- ❑ 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:
 - software tools are ~easy to get, but still need know-how
 - forum, workshops etc....
 - collaboration between HEP and ML scientists are often needed
- ❑ More and more open datasets/simulators to favor the collaborations
- ❑ More and more HEP and ML workshops, forums, group, challenges etc...
- ❑ Never underestimate the time for :
 - (1) Great (or small) ML idea →
 - (2) ...demonstrated on toy dataset →
 - (3) ...demonstrated on real experiment analysis/dataset →
 - (4) ...experiment publication using the great (or small) idea