# Machine Learning and Statistics in HEP

## part 3


David Rousseau, IJCLab-Orsay

rousseau@ijclab.in2p3.fr @dhpmrou


ICFA 2023 Instrumentation School

Mumbai, Feb 2023

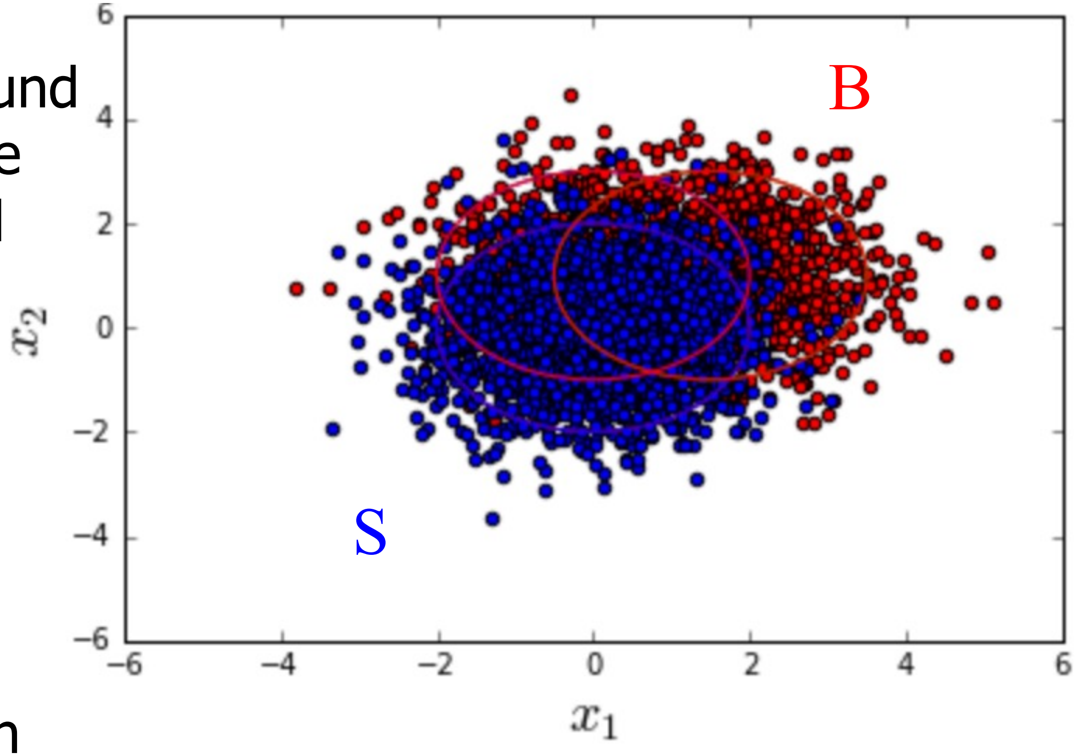
# Outline

- Mostly Machine Learning with Statistics interludes
- Part 1 : Overview
  - What is Machine Learning?
  - Specificities of ML in physics
  - Useful concepts
- Part 2 : wider and deeper
  - NN on HEP data
    - various hammers and nails (including wrong ones)
  - Graph NN
  - Anomaly detection
- Part 3 : even wider and deeper
  - ML training tricks
  - Surrogate models
  - Recommendations for ML software and tools

See CERN Inter-Experiment Machine Learning workshop May 2022

# No miracle

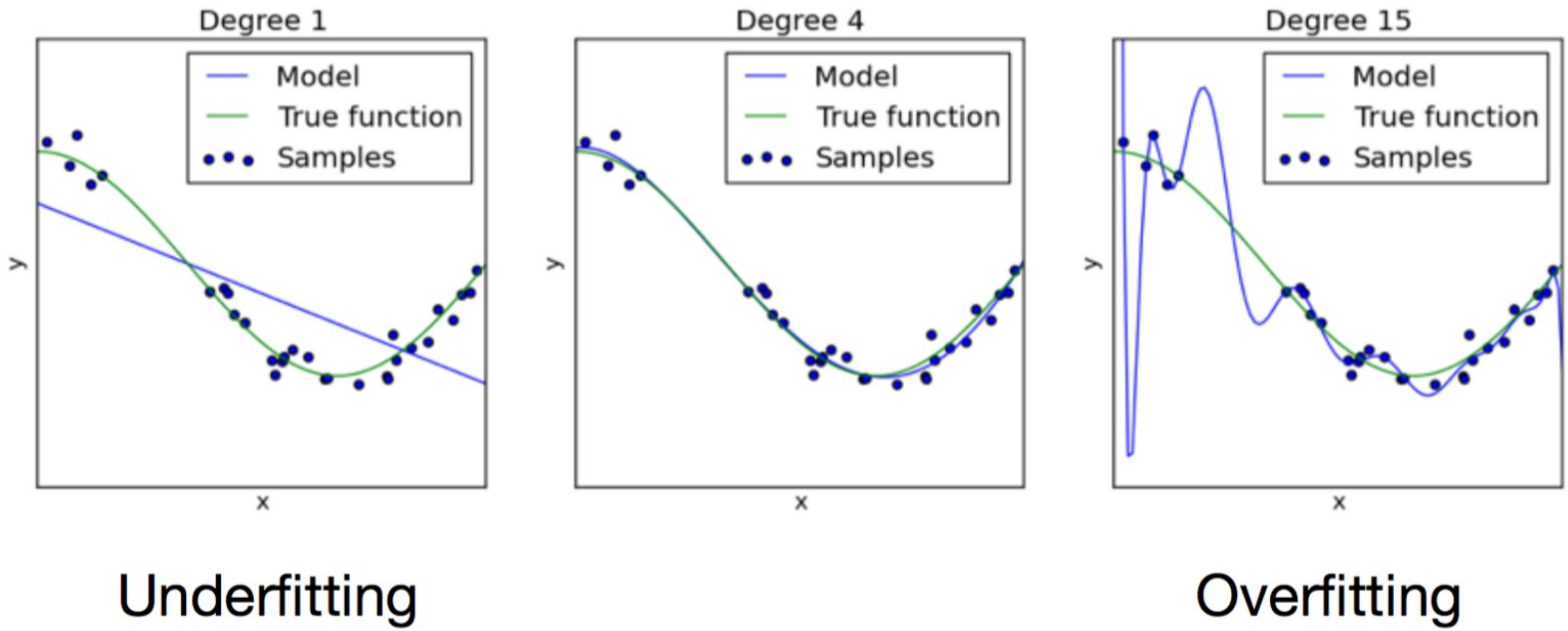
- ML (nor Artificial Intelligence) does not do any miracles
- For selecting Signal vs Background and underlying distributions are known, nothing beats likelihood ratio! (often called “Bayesian limit”):
  - $L_S(x)/L_B(x)$
- OK but quite often $L_S$ $L_B$ are unknown
  - + x is n-dimensional
- ML starts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



# Under/Over-training

# What is Overfitting

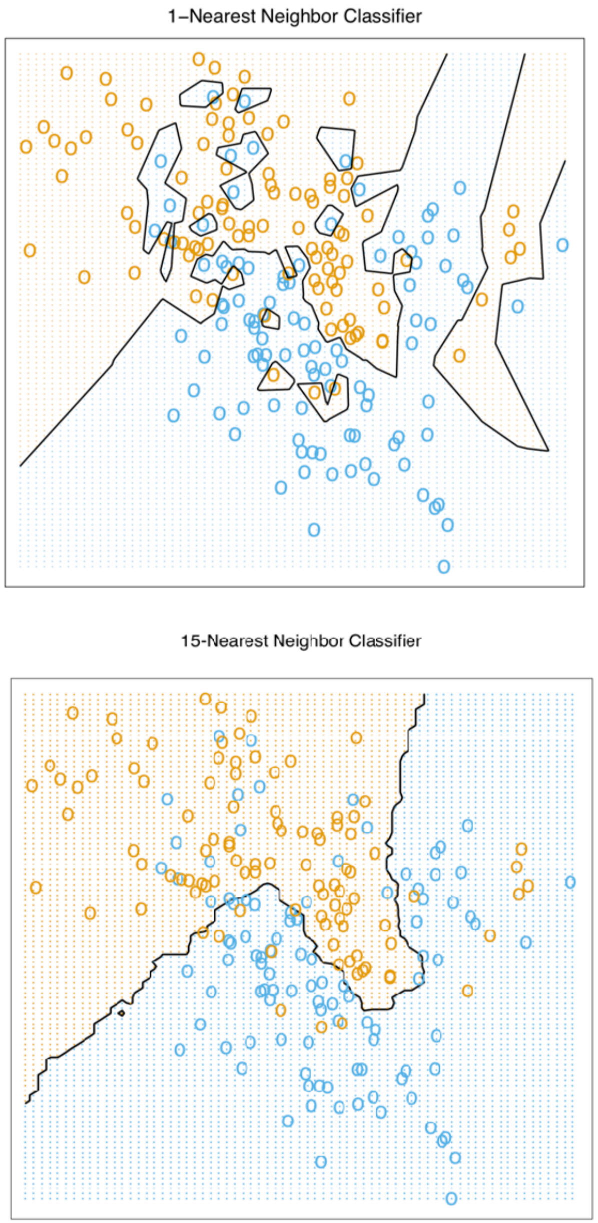


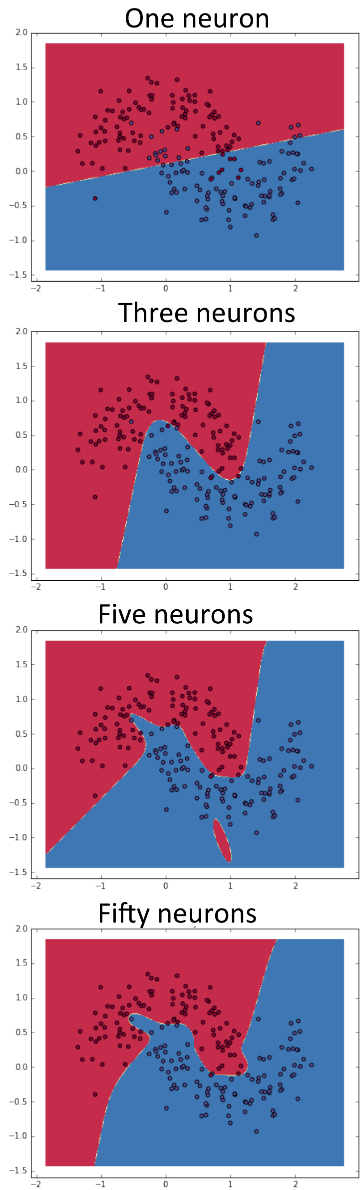



http://scikit-learn.org/

- What models allow us to do is **generalize** from data
- Different models generalize in different ways

# Overtraining examples




- Overtraining affect all algorithms
- ...when model is too complex wrt amount of training data

http://www.wildml.com/2015/09/impl

# under/over training

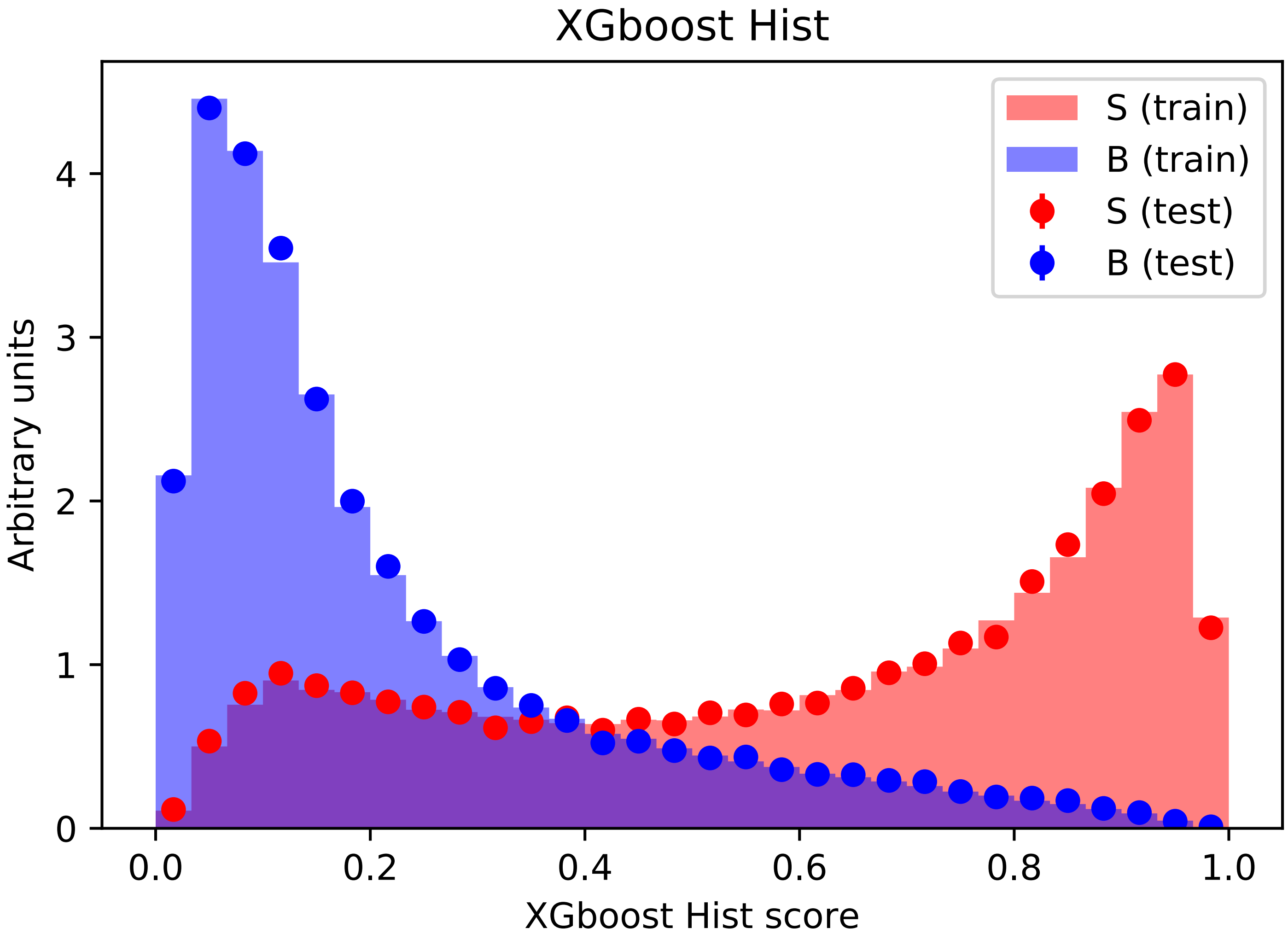
Gilles Louppe, github


Some overtraining is good

…provided CrossValidation done correctly!

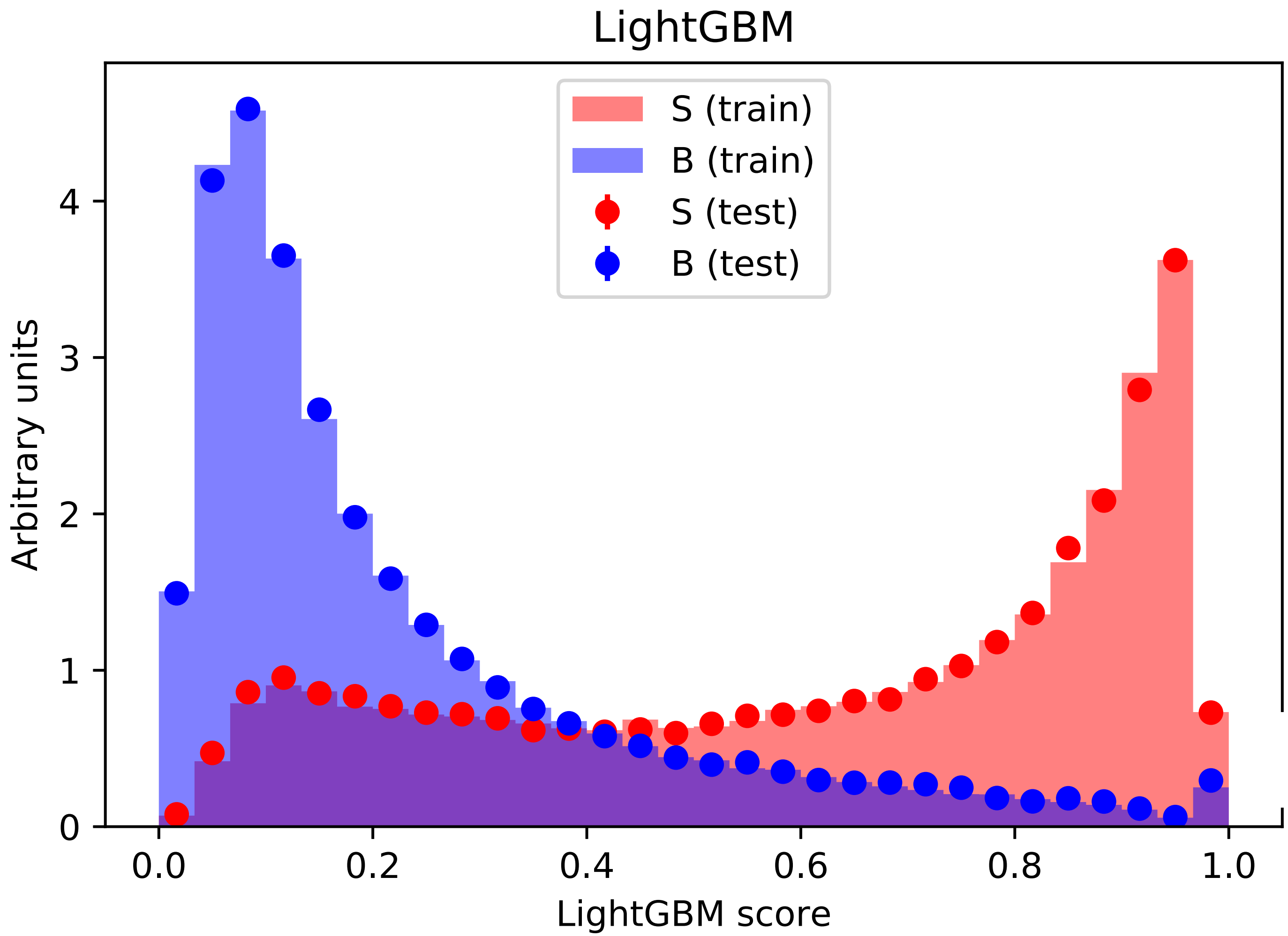
# Training vs test check

- The score distribution should match…
- …some discrepancy is OK



# Training vs test check (2)

- Important to check



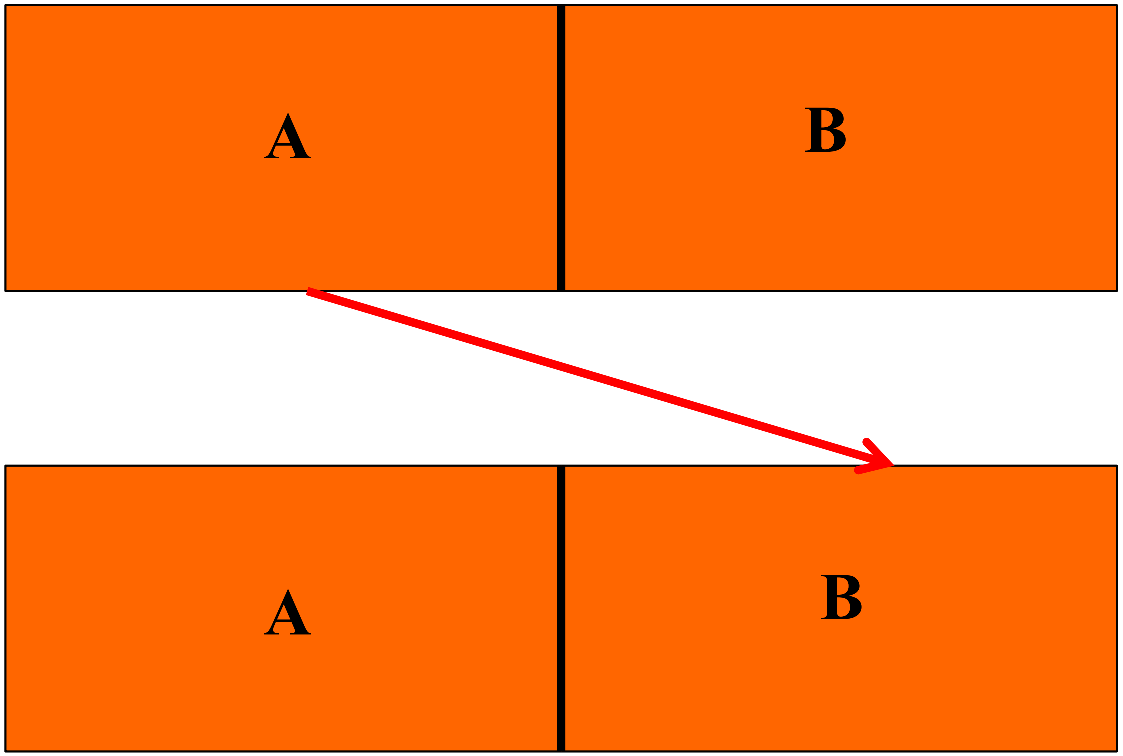

←Bug!

# Cross-validation

# Cross-Validation

One-fold Cross Validation

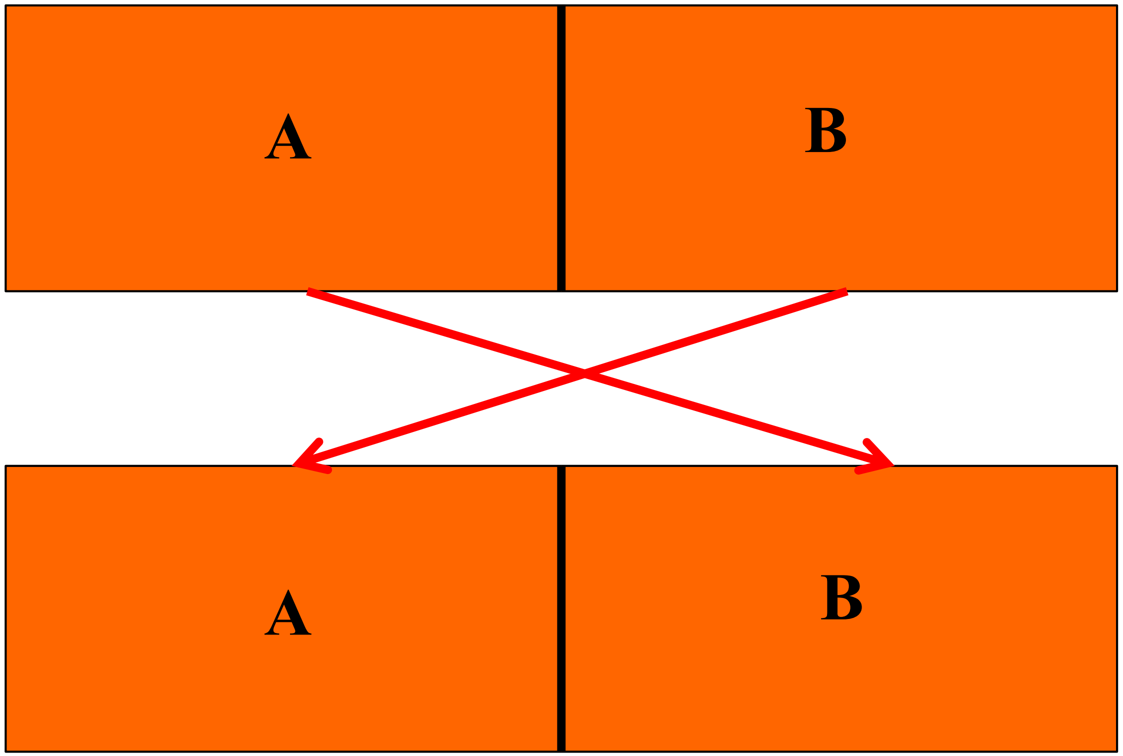
Goal of CV is to measure performance and optimise hyper-parameters




Standard basic way (default TMVA until recently)

# Cross-Validation

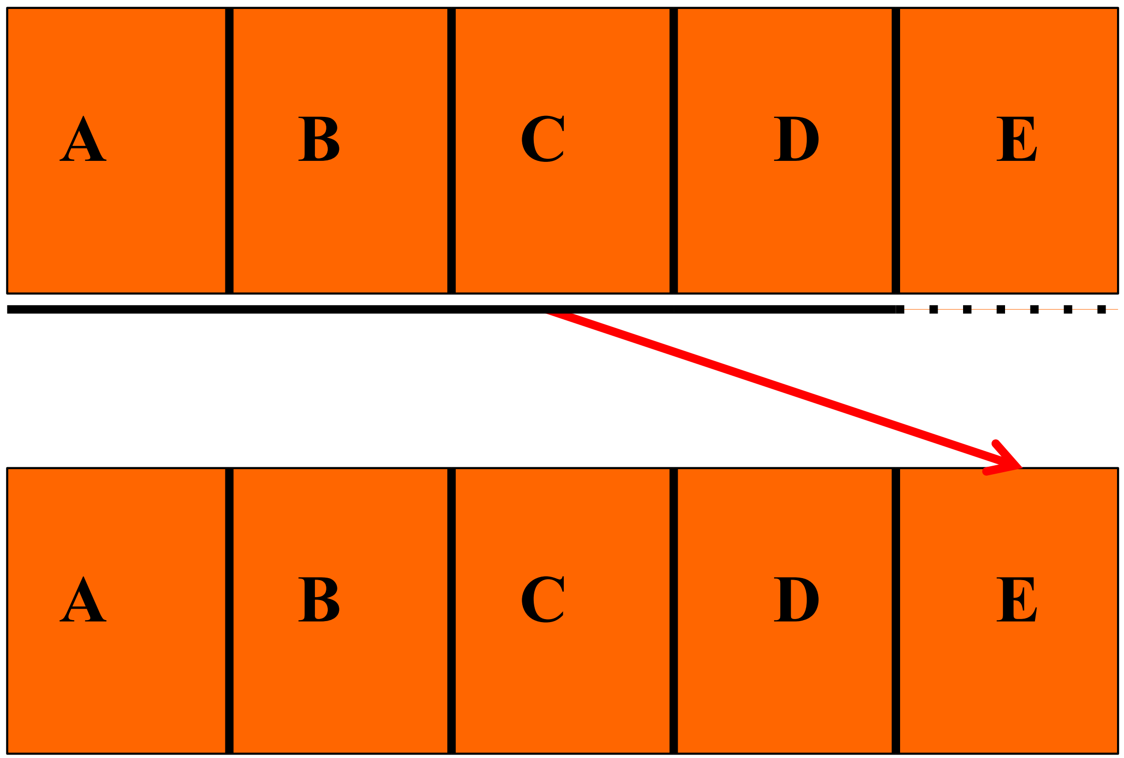
Two-fold Cross Validation




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

# Cross-Validation

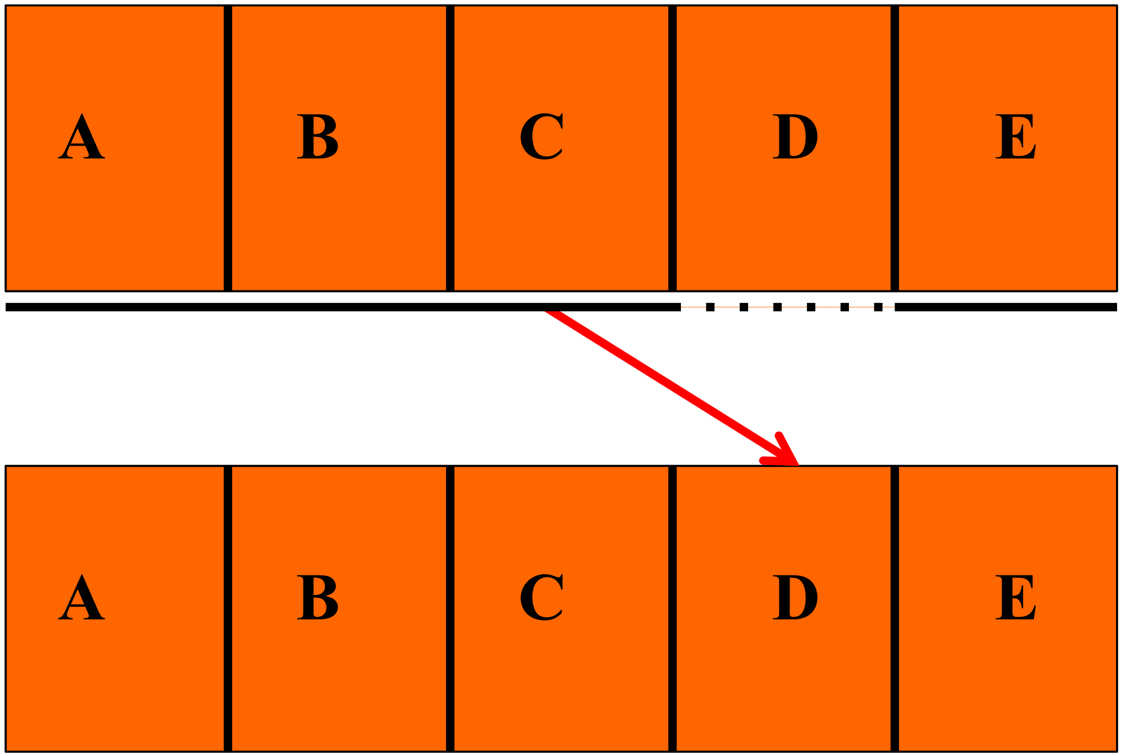
5-fold Cross Validation




same test statistics wrt two-fold CV,
larger training statistics 4/5 over ½ (larger training time as well)

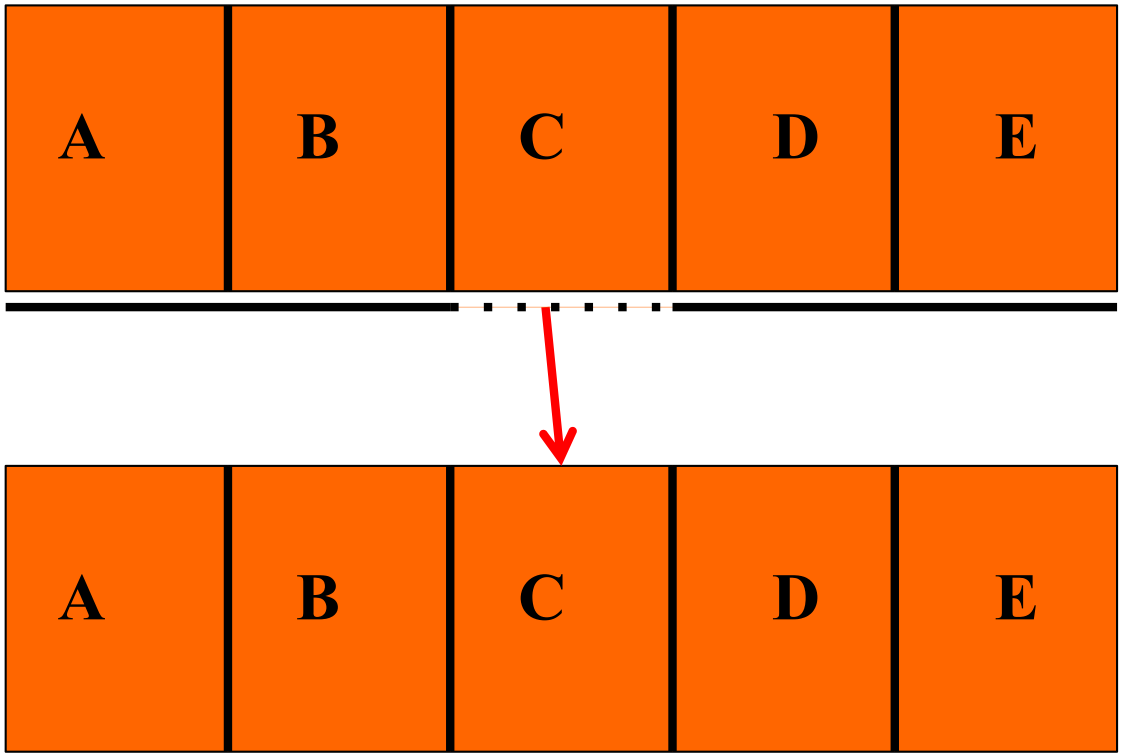
# Cross-Validation

5-fold Cross Validation



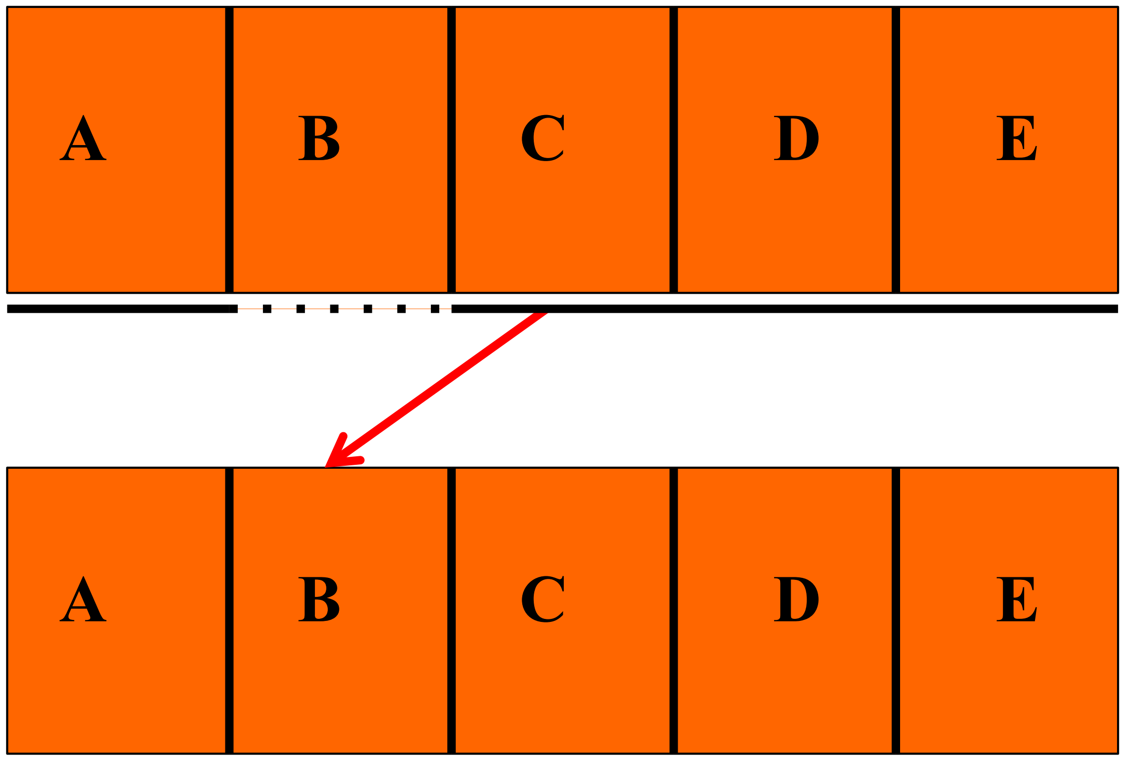
# Cross-Validation

5-fold Cross Validation



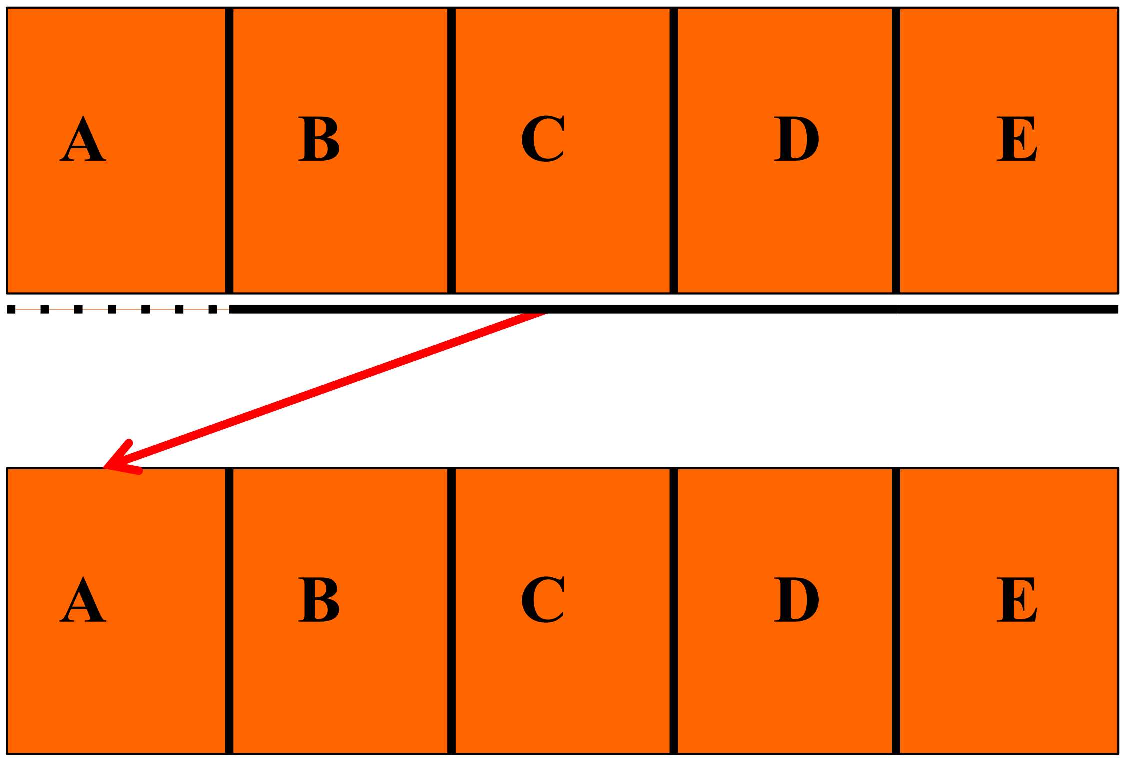
# Cross-Validation

5-fold Cross Validation



# Cross-Validation

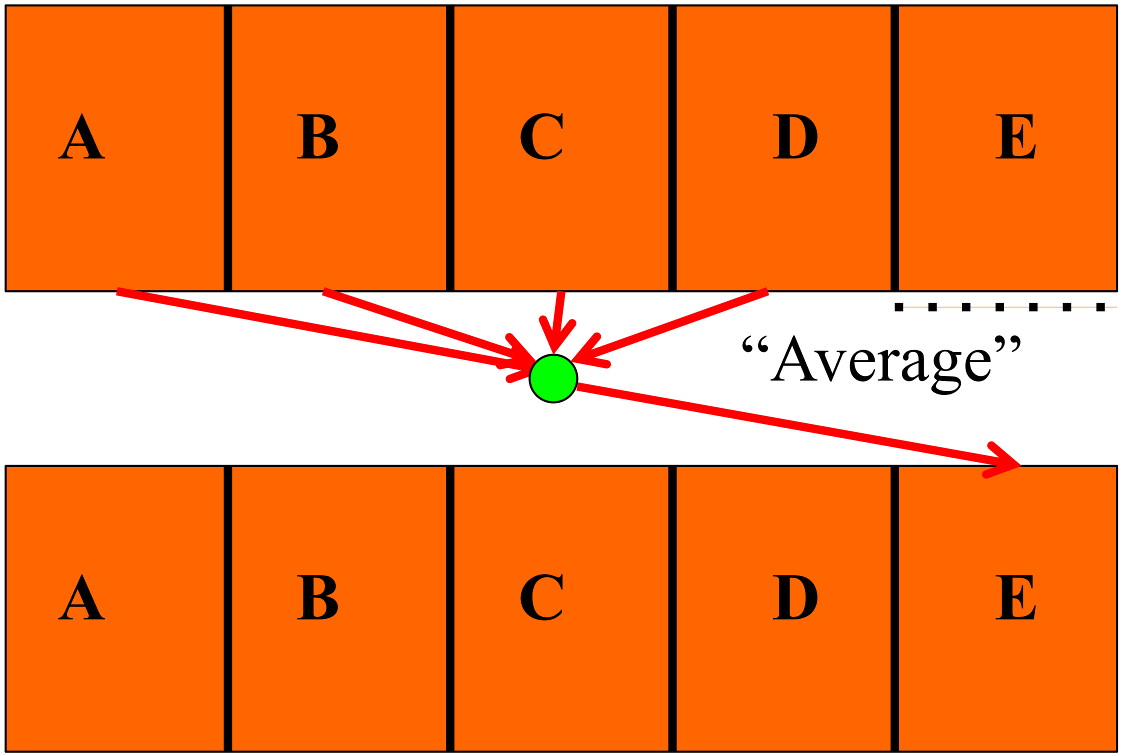
5-fold Cross Validation




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
bonus: variance of the samples an estimate of the statistical uncertainty
(also save on training time)

# Cross-validation

Train on …

Test on …

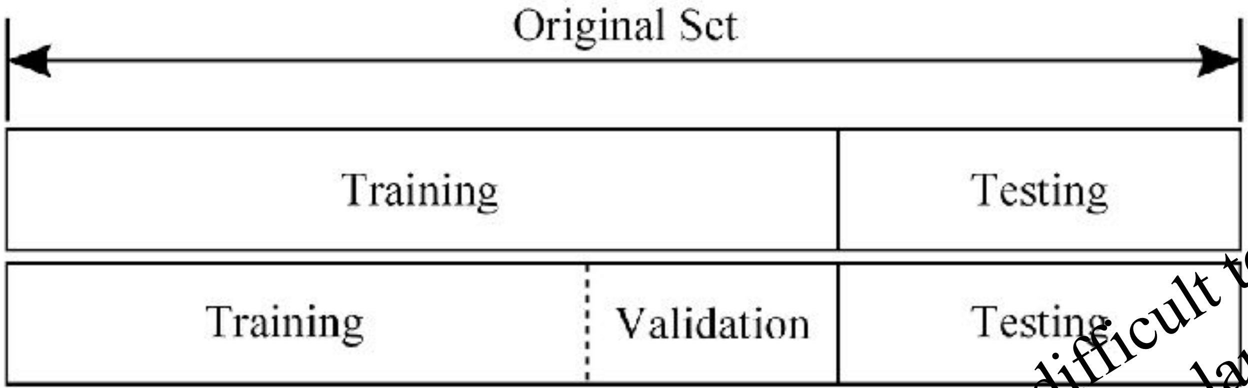
| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| A | X | | | | | → Testing variance |
| B | | X | | | | → Testing variance |
| C | | | X | | | → Testing variance |
| D | | | | X | | → Testing variance |
| E | | | | | X | → Testing variance |

↓ Training variance ↓ Training variance ↓ Training variance ↓ Training variance ↓ Training variance

# Training, Validation, Test

-

**Divide the labelled set into training, validation and testing sets**




More difficult to handle when working in large collaboration

- **<u>Training set:</u> used to train the classifier**
- **<u>Validation set (optional)</u>: choose between different methods, finite-tune parameters,**
- **<u>Testing set:</u> predict the generalization error**

Ideally, look at it only once at the end

**<u>No cheat:</u> do not use the test set to train your algorithm!**

# ML highest crime

# Horror stories

See page

Abuse of validation set (and no test set) lead to exagerated claim

MIT Technology Review

Intelligent Machines

## Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

by **Tom Simonite**

Jun 4, 2015

**The sport of training software to act intelligently just got its first cheating** scandal. Last month Chinese search company Baidu announced that its

Also in physics…

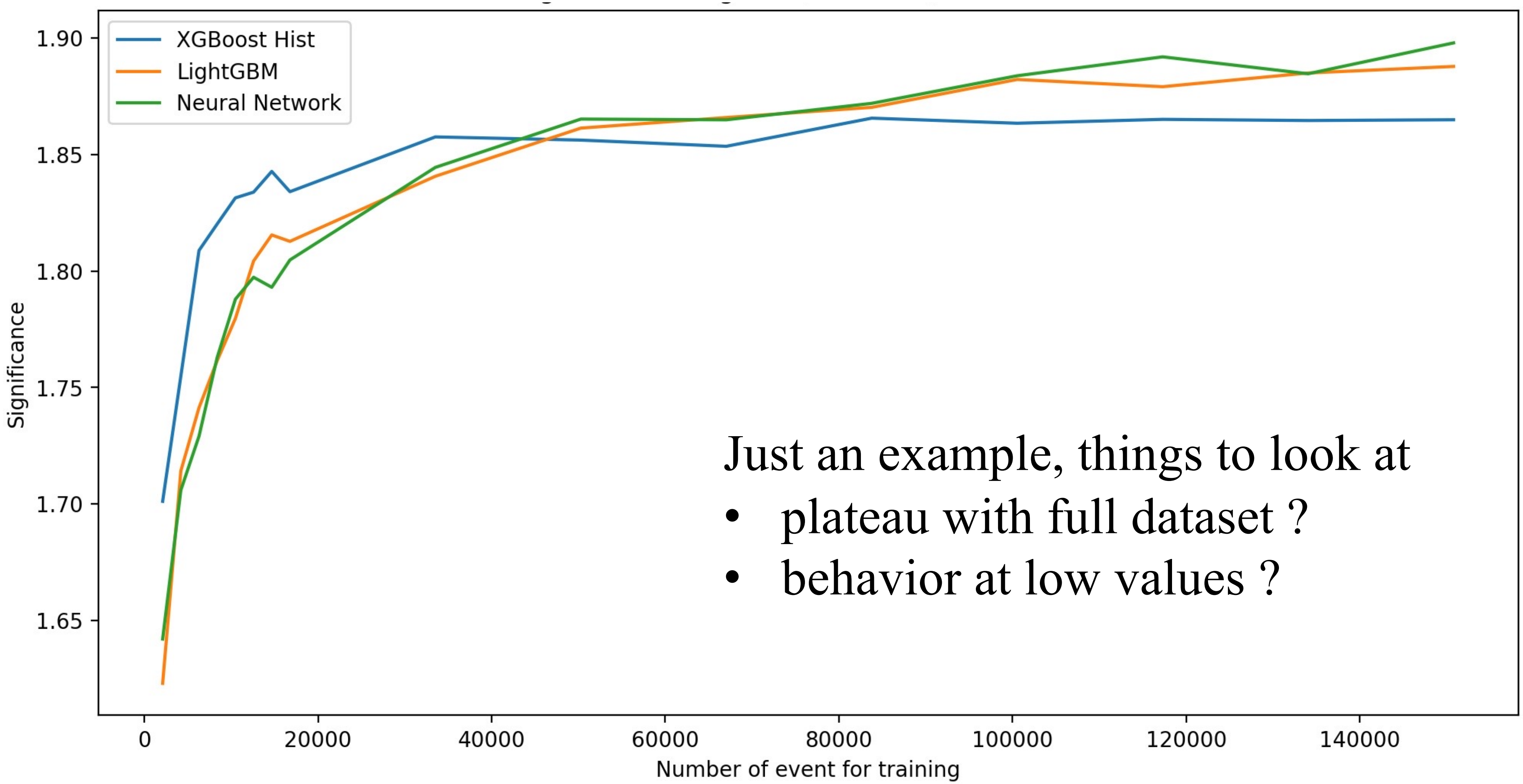
Training on test set particularly bad because undetectable unless:

- training reproducible
- new i.i.d data

# Learning curve

# Learning curve

- Performance as a function of number of training events



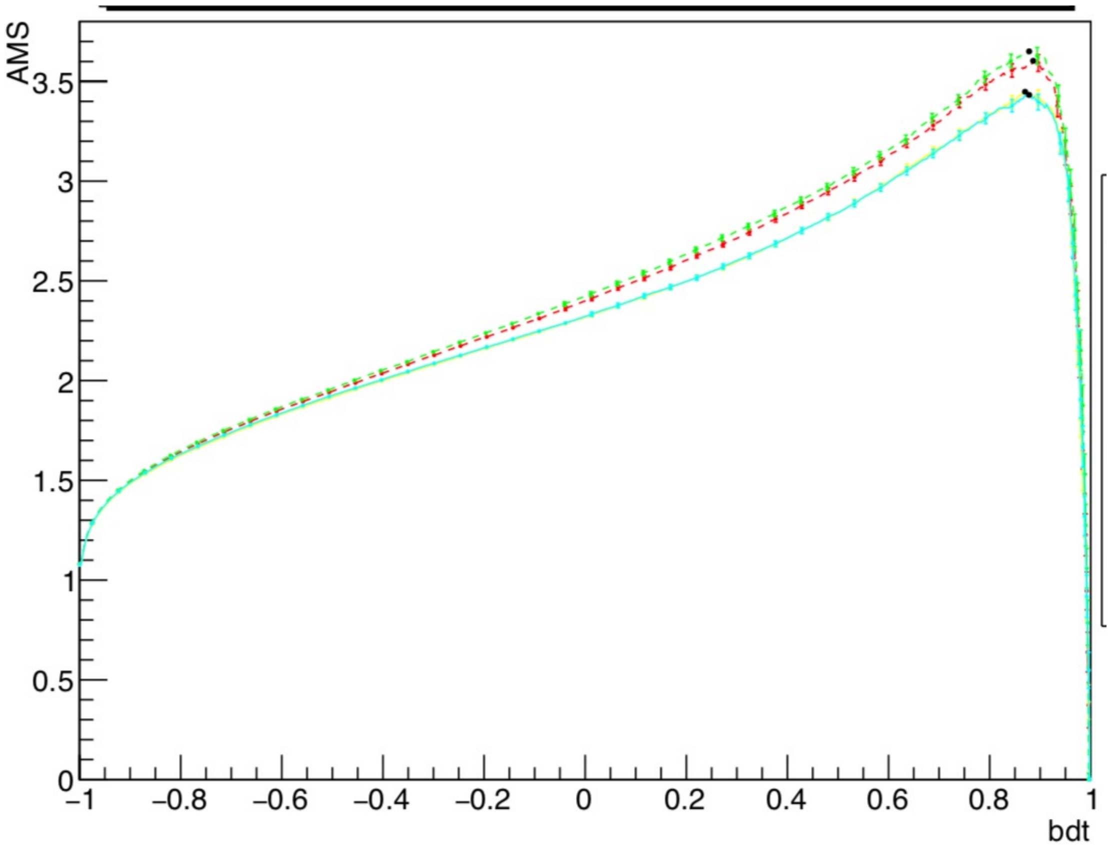

Just an example, things to look at

- plateau with full dataset ?
- behavior at low values ?

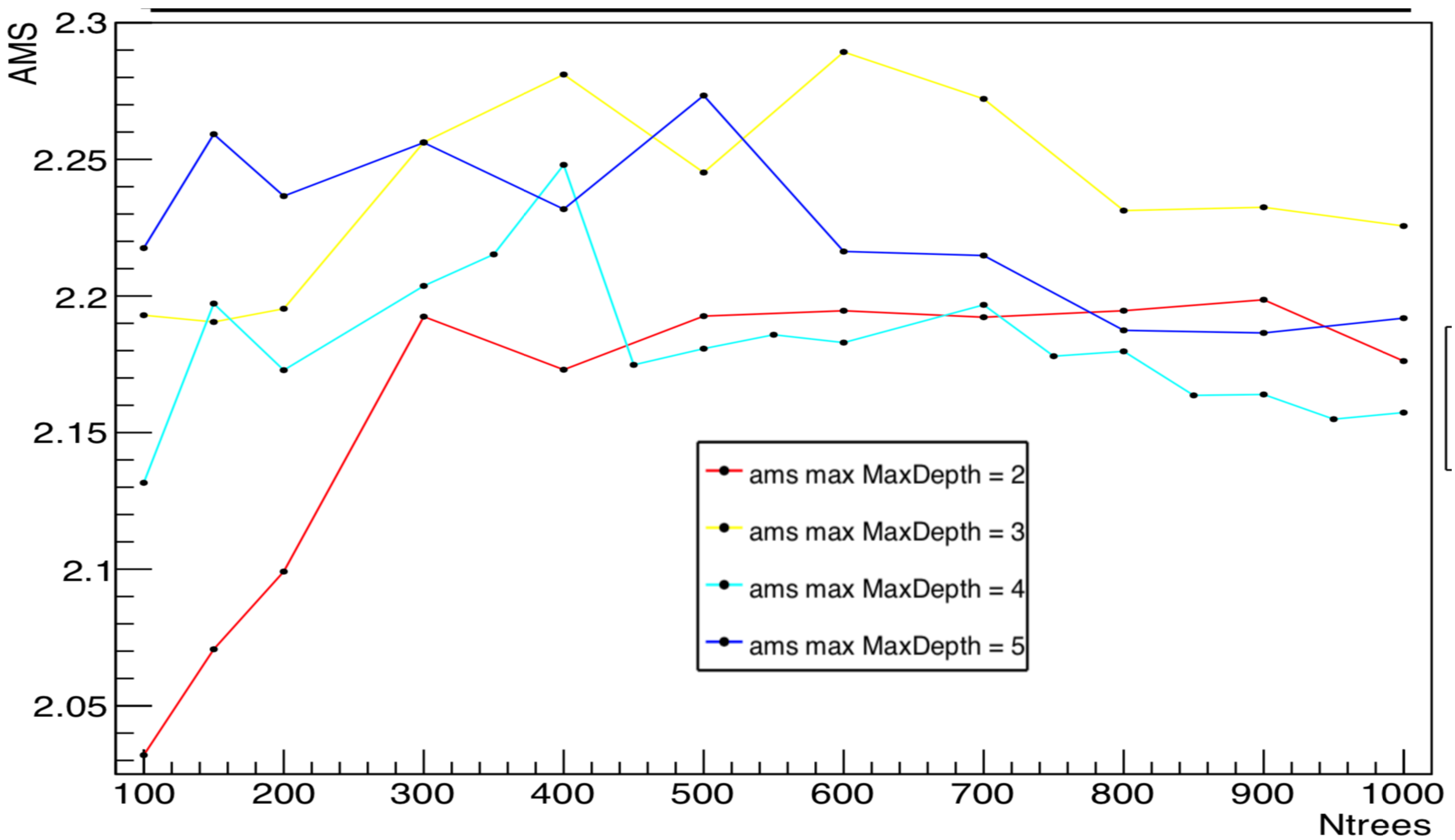
# Hyper-Parameter Optimisation (HPO)

# Hyper Parameters Optimization

- HPO : tune the auxilliary parameters for a specific problem
- =>Redo the study for many combinations
- (also sklearn GridSearchCV)



# HPO (2)


AMS
2.3
2.25
2.2
2.15
2.1
2.05
100
200
300
400
500
600
700
800
900
1000
Ntrees
ams max MaxDepth = 2
ams max MaxDepth = 3
ams max MaxDepth = 4
ams max MaxDepth = 5

# Surrogate models

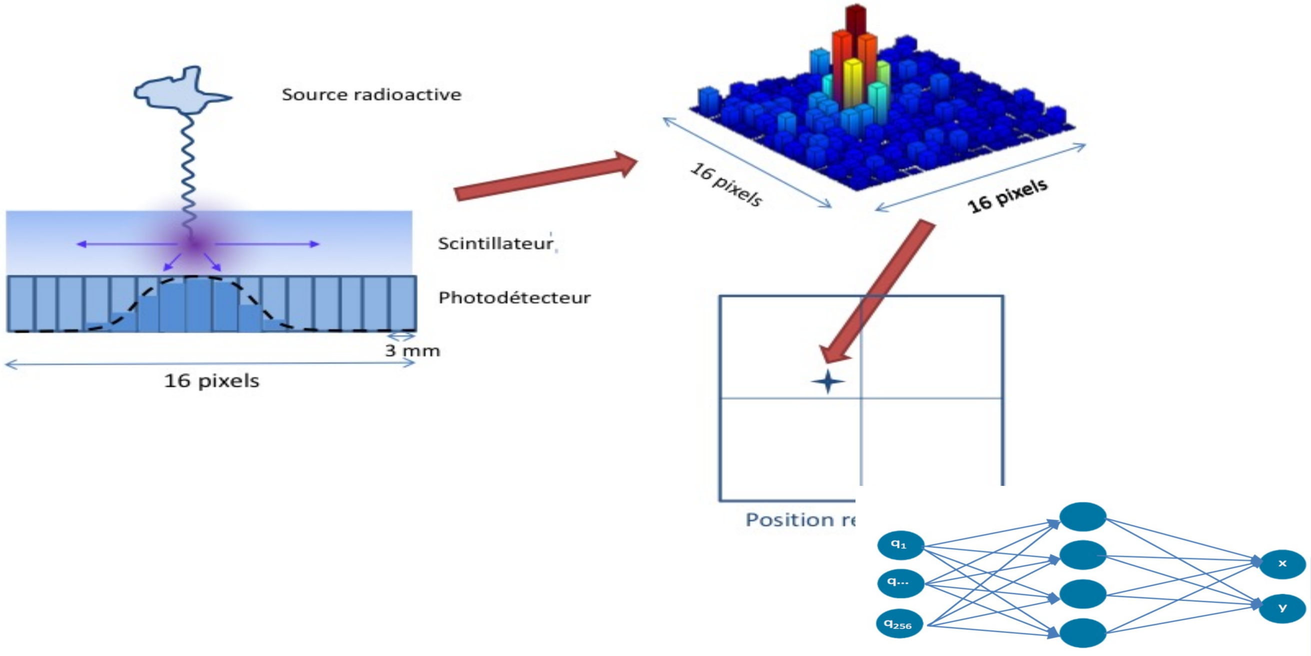
**Universal Approximation Theorem:**

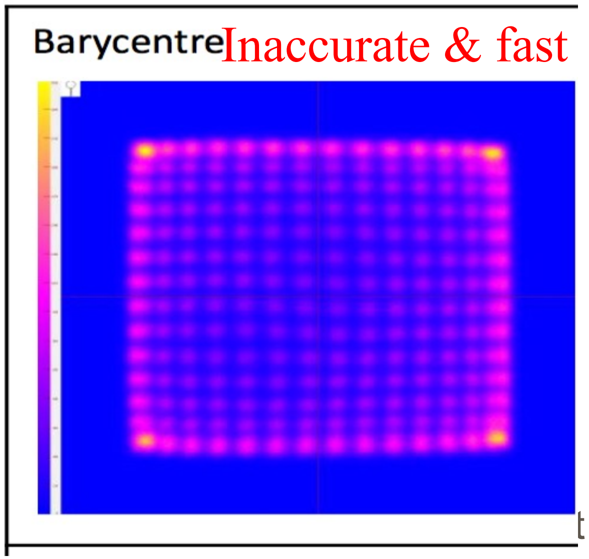
**A NN can emulate with arbitrary precision any model y=f(x) (x and y of any dimension) …**

**… if properly trained !**

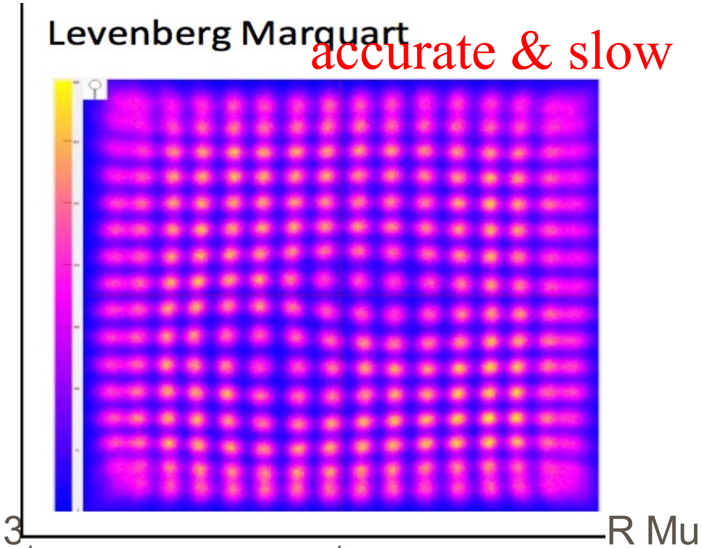
# β,γ camera for medical application

Françoise Bouvet IMNC

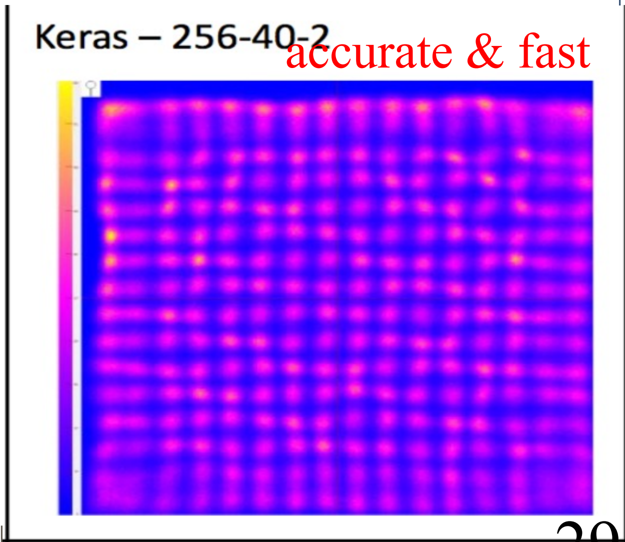



Inaccurate & fast

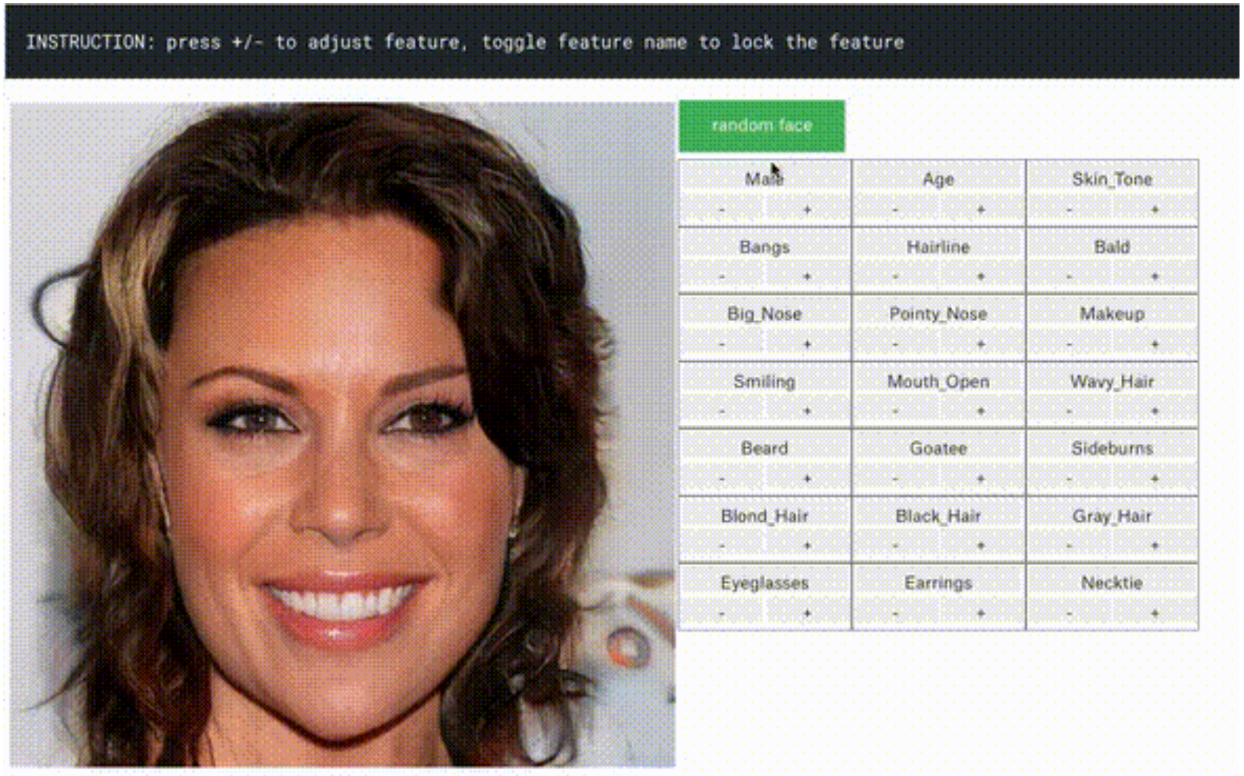



accurate & slow

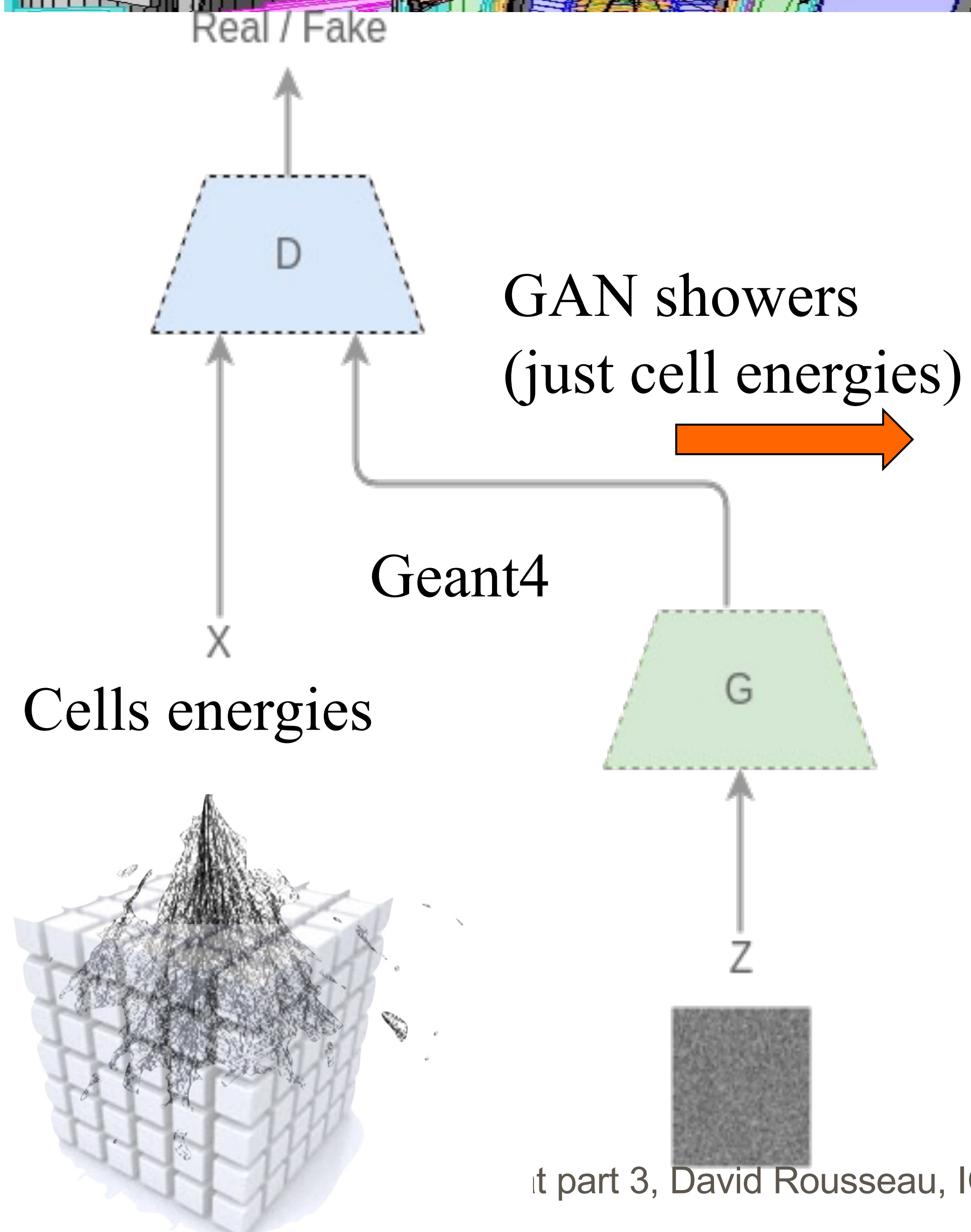



accurate & fast

# Generative model



# GAN for simulation (1)

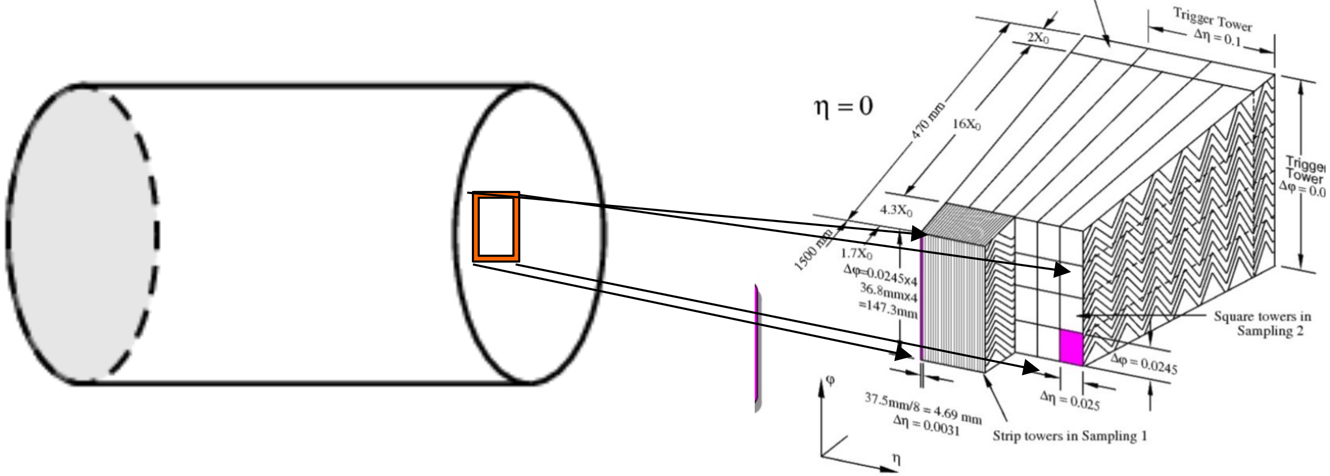



- Half of LHC grid computers (~1.000.000 cores) are crunching Geant4 simulation 24/24 365/365
- …while LHC experiments are collecting more and more events
- ➔reducing CPU consumption of simulation is very important

- 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)

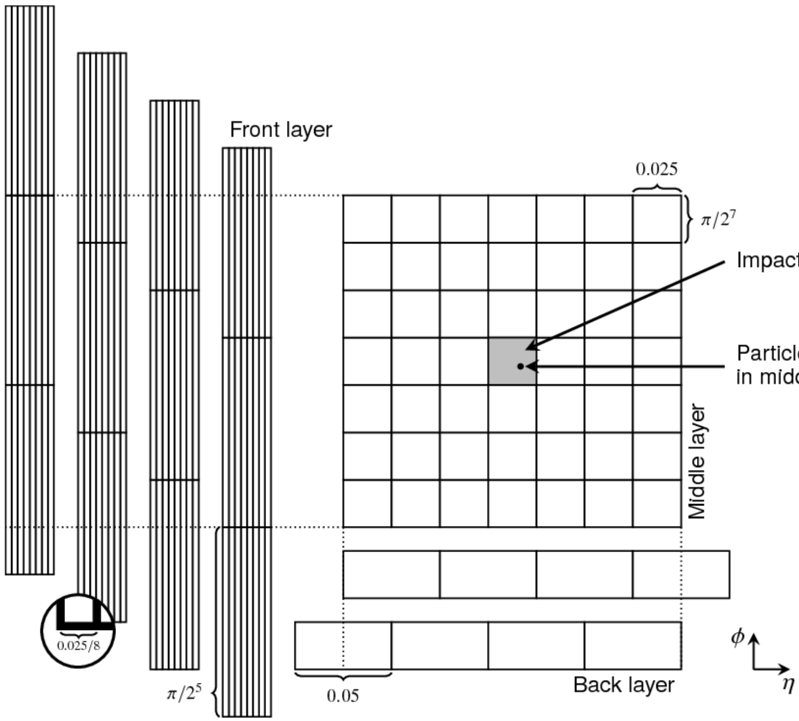
# ATLAS calo simulation

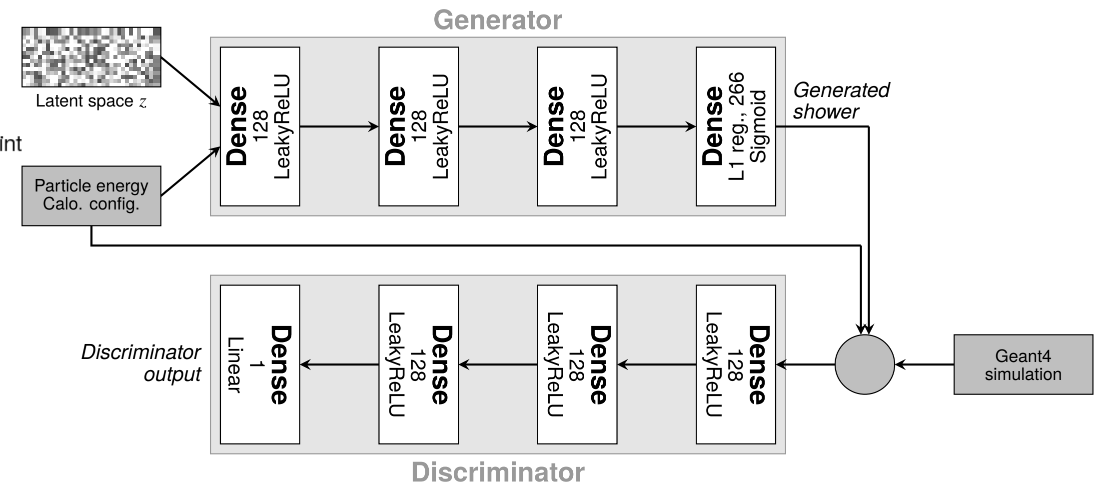
arXiv:2210.06204



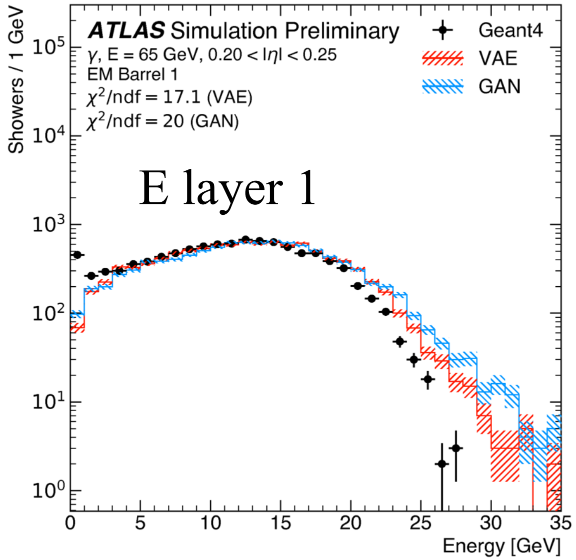

+ η, ϕ translation

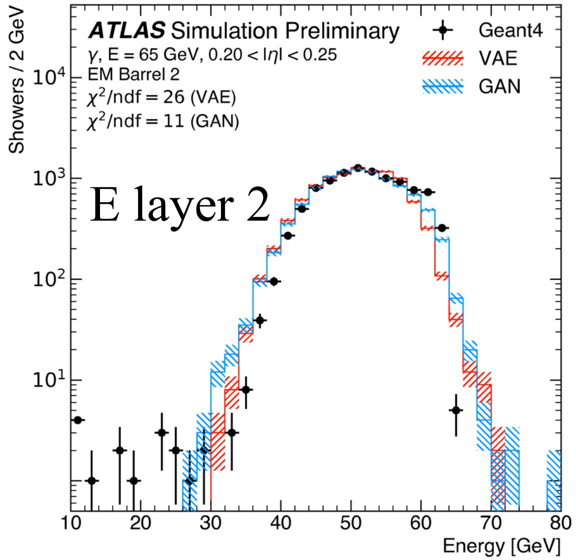
177000 cells ➔266 cells

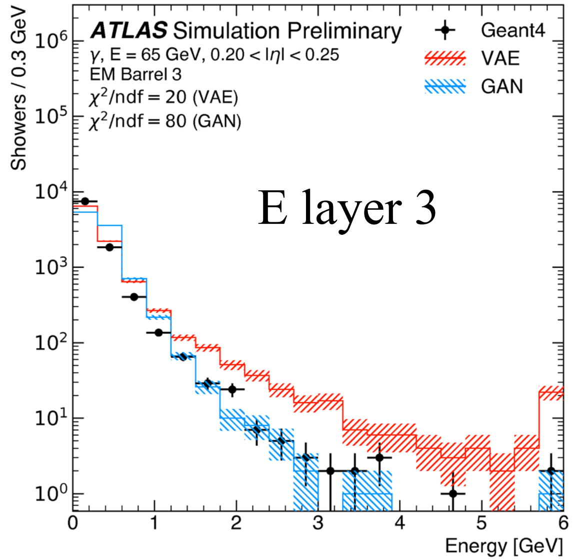


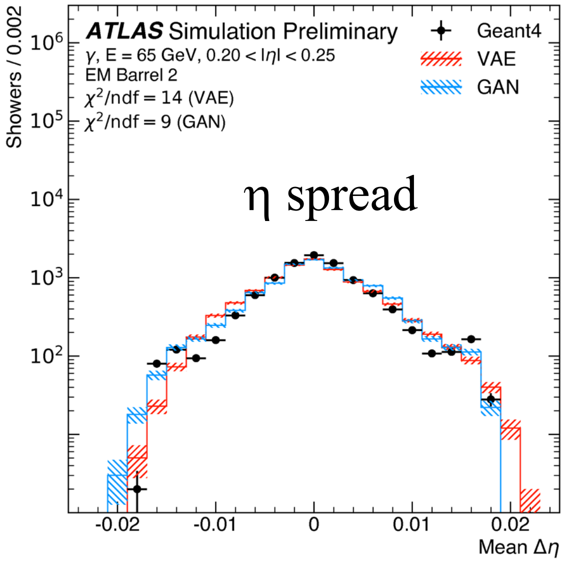



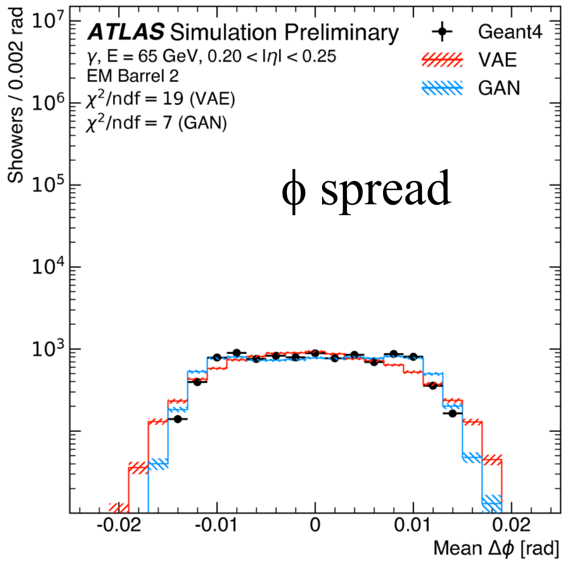
# Results



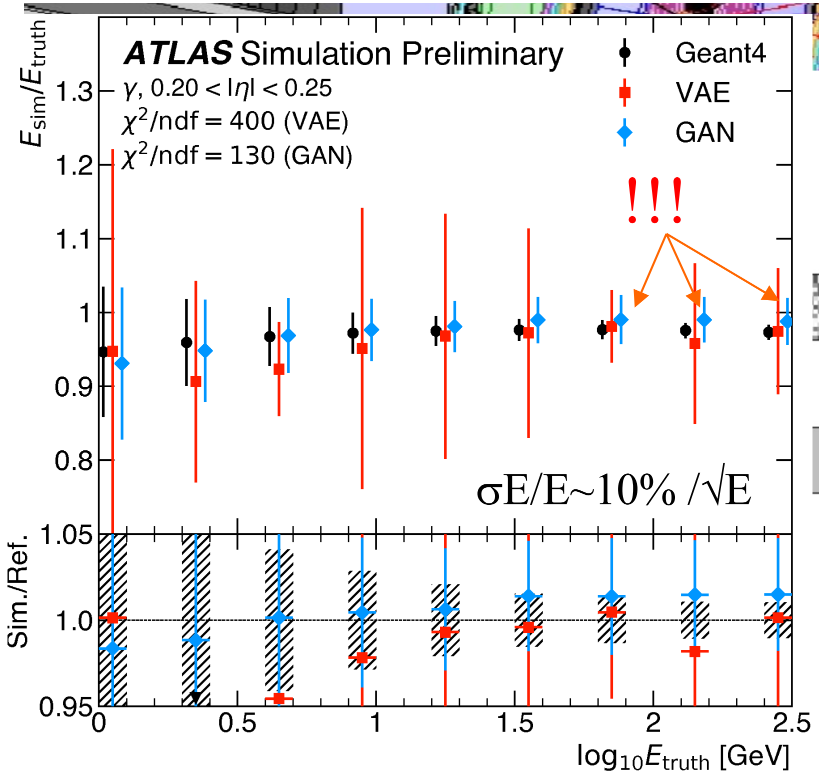



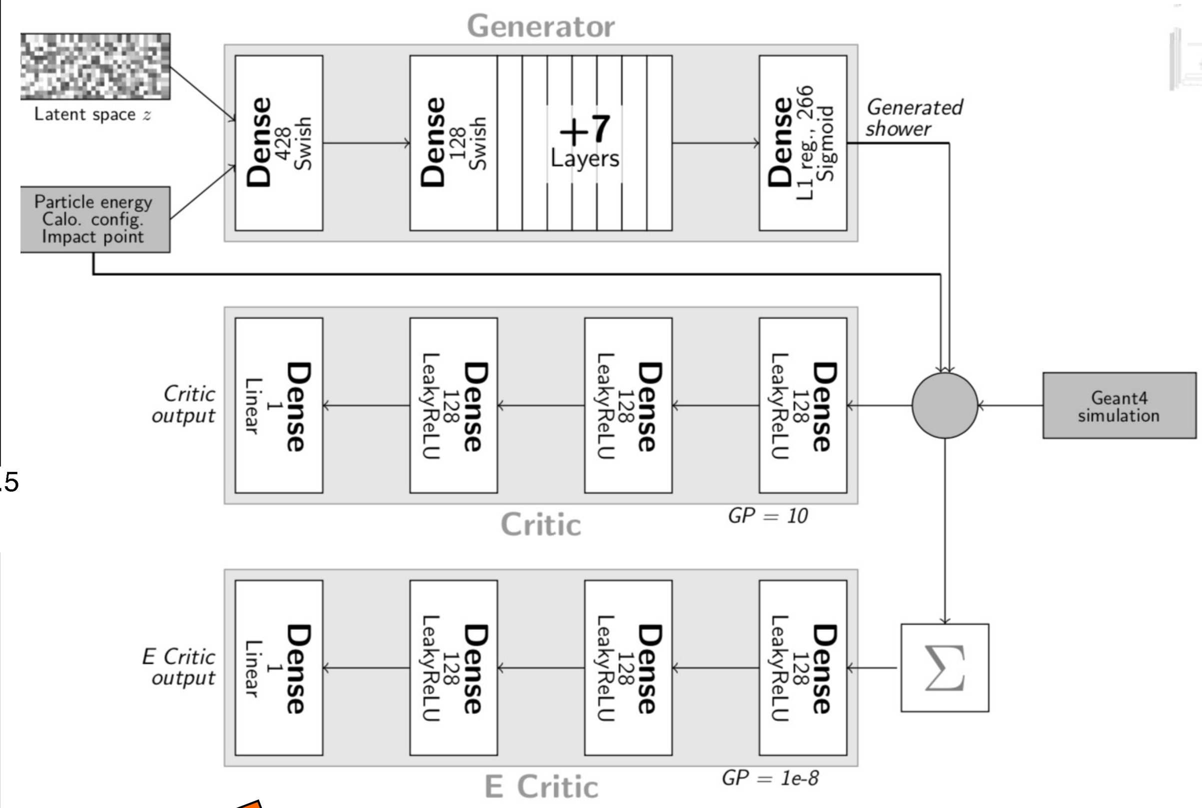

- Speed: <1ms compared to 10s
- Promise : as accurate as more classical parameterisation approach with less hand tuning
- Sufficient accuracy ?
- Handling of edges ?

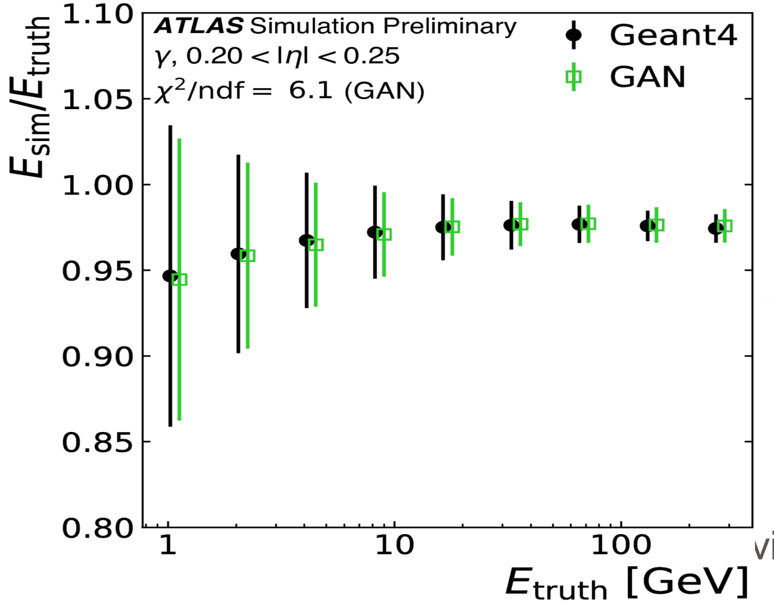
Just an example, many more generative models in arXiv

# Simulation of energy resolution




Updated architecture



# So much progress since 2016

See nice introduction

#Dall-e Diffusion model

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

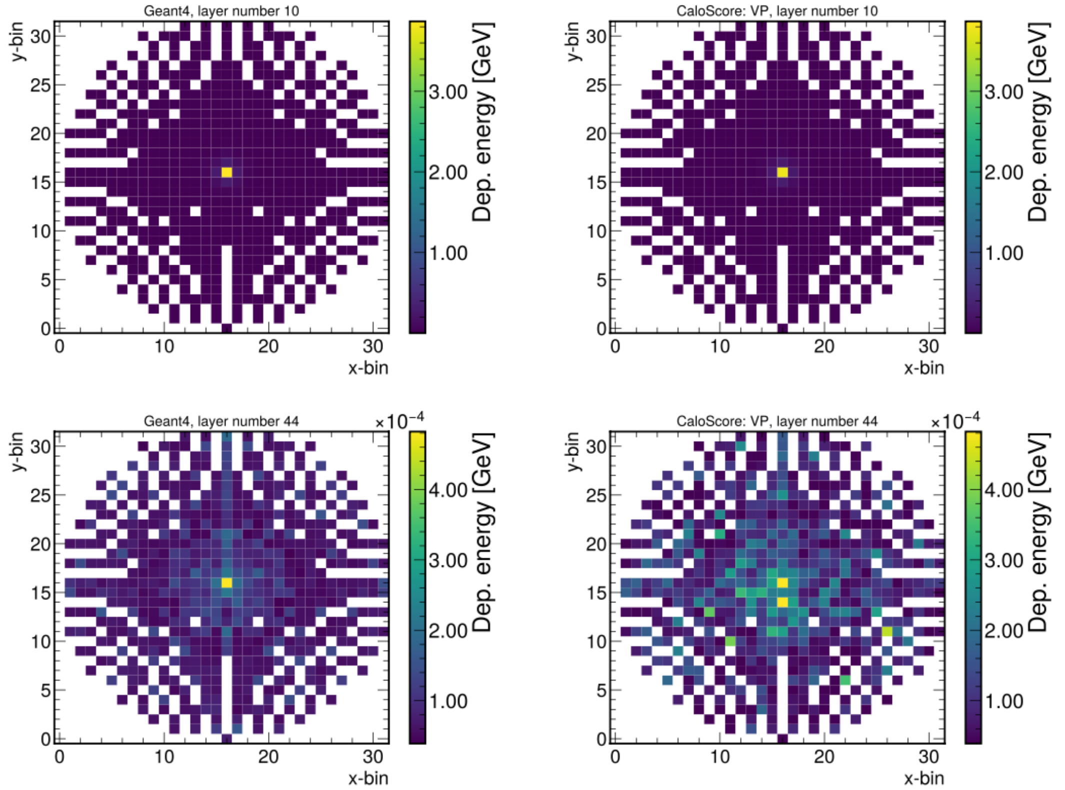
a corgi's head depicted as an explosion of a nebula

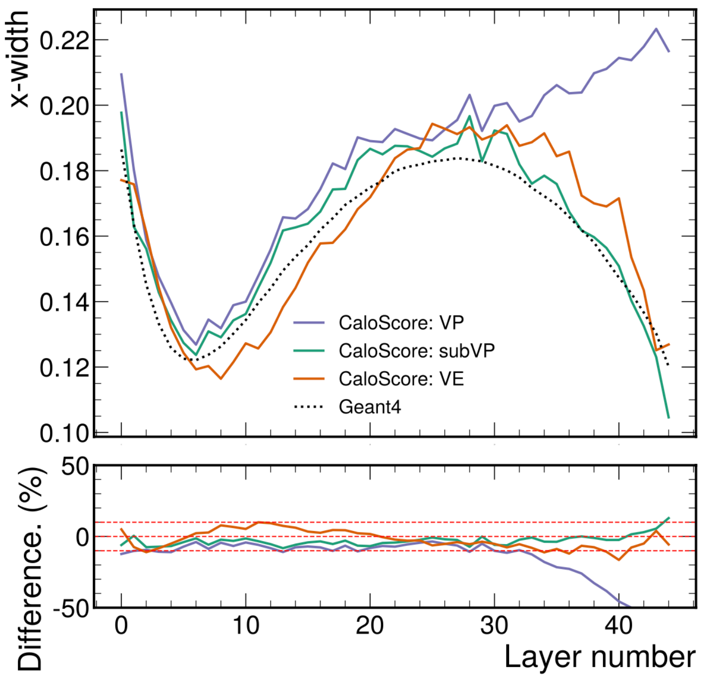
GAN becoming obsolete, can we do something with diffusion model ?

# Simulation with diffusion model

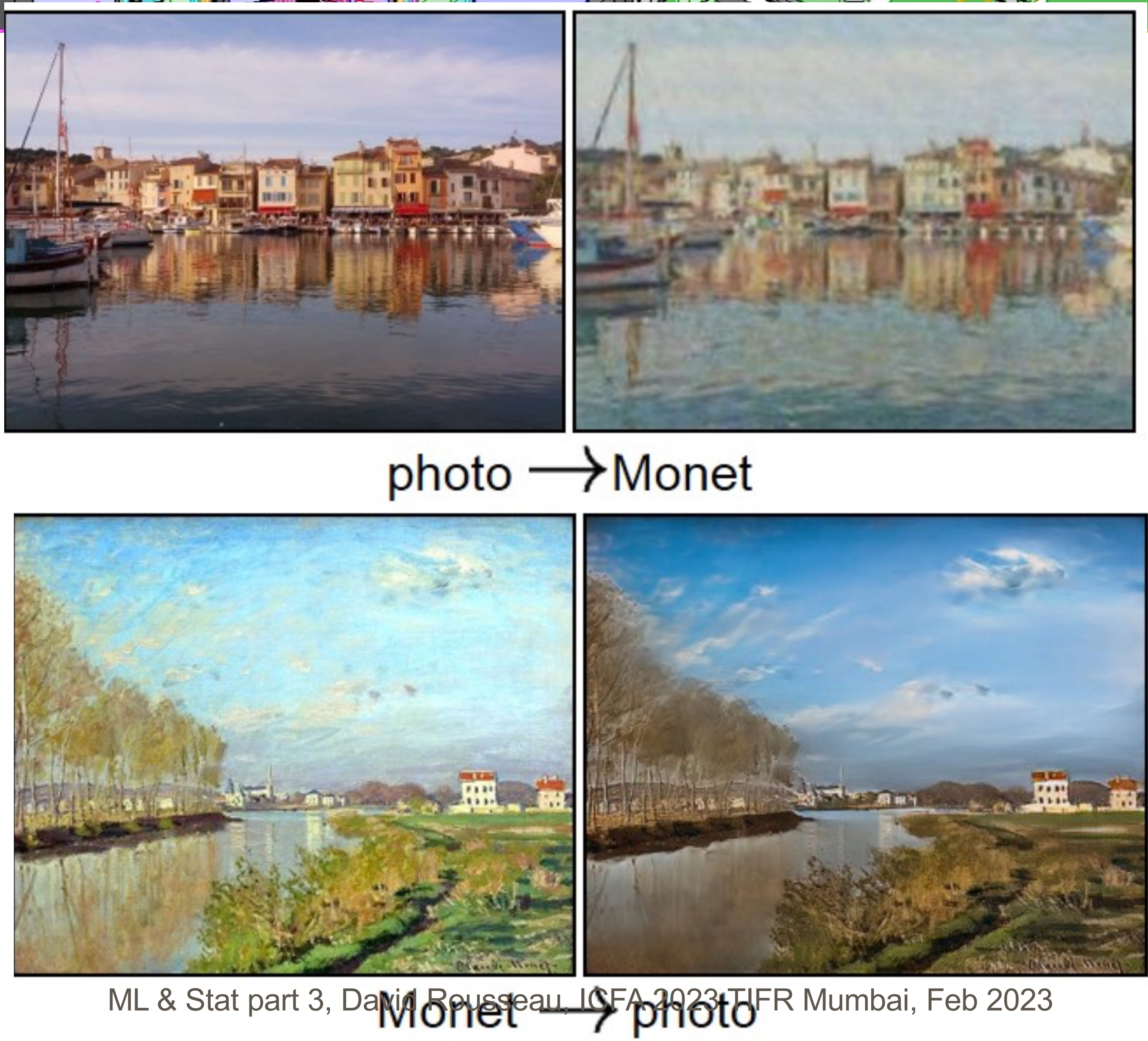
Mikuli, Nachman

- Fast Calorimeter Simulation Challenge Datasets 2022
- Open Geant4 simulation, 3 datasets of increasing detector complexity
- ➔seems to work, claim it is easier to train
- However inference (=generation) slower



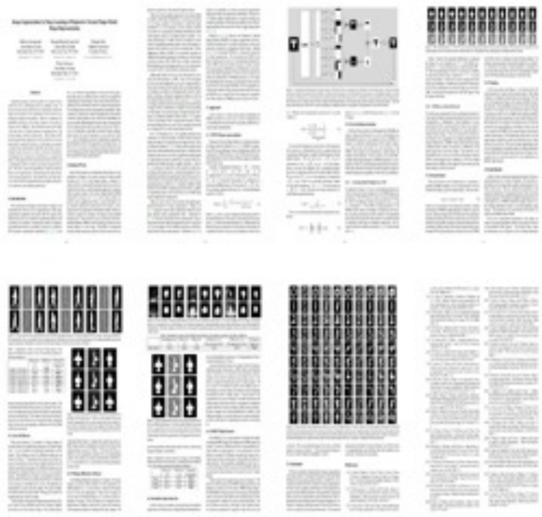
# Another GAN application

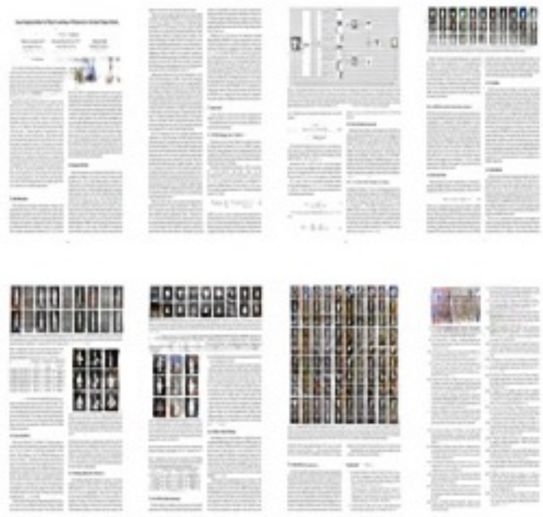


# Paper Visual Layout

Bad paper

Good paper

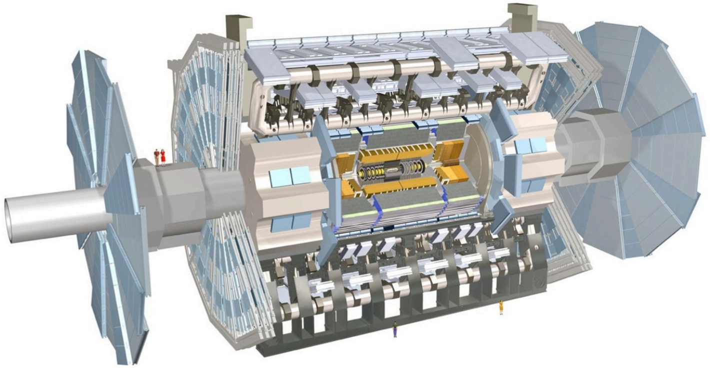




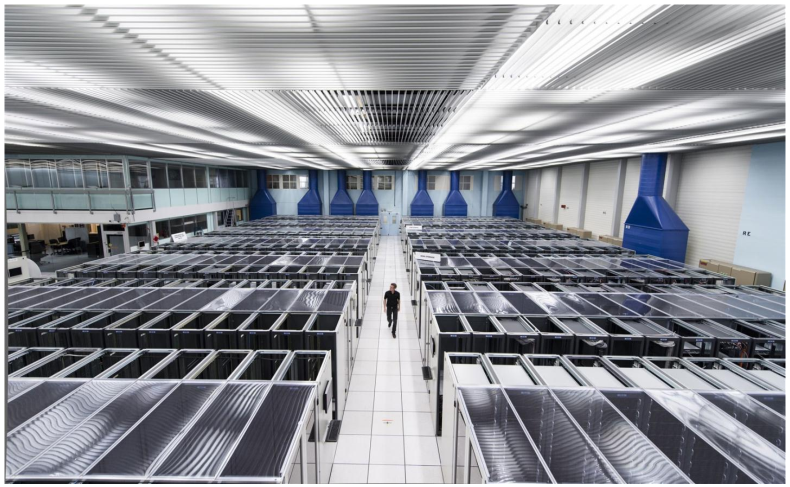
# CycleGAN

# CycleGAN

# Application



Données

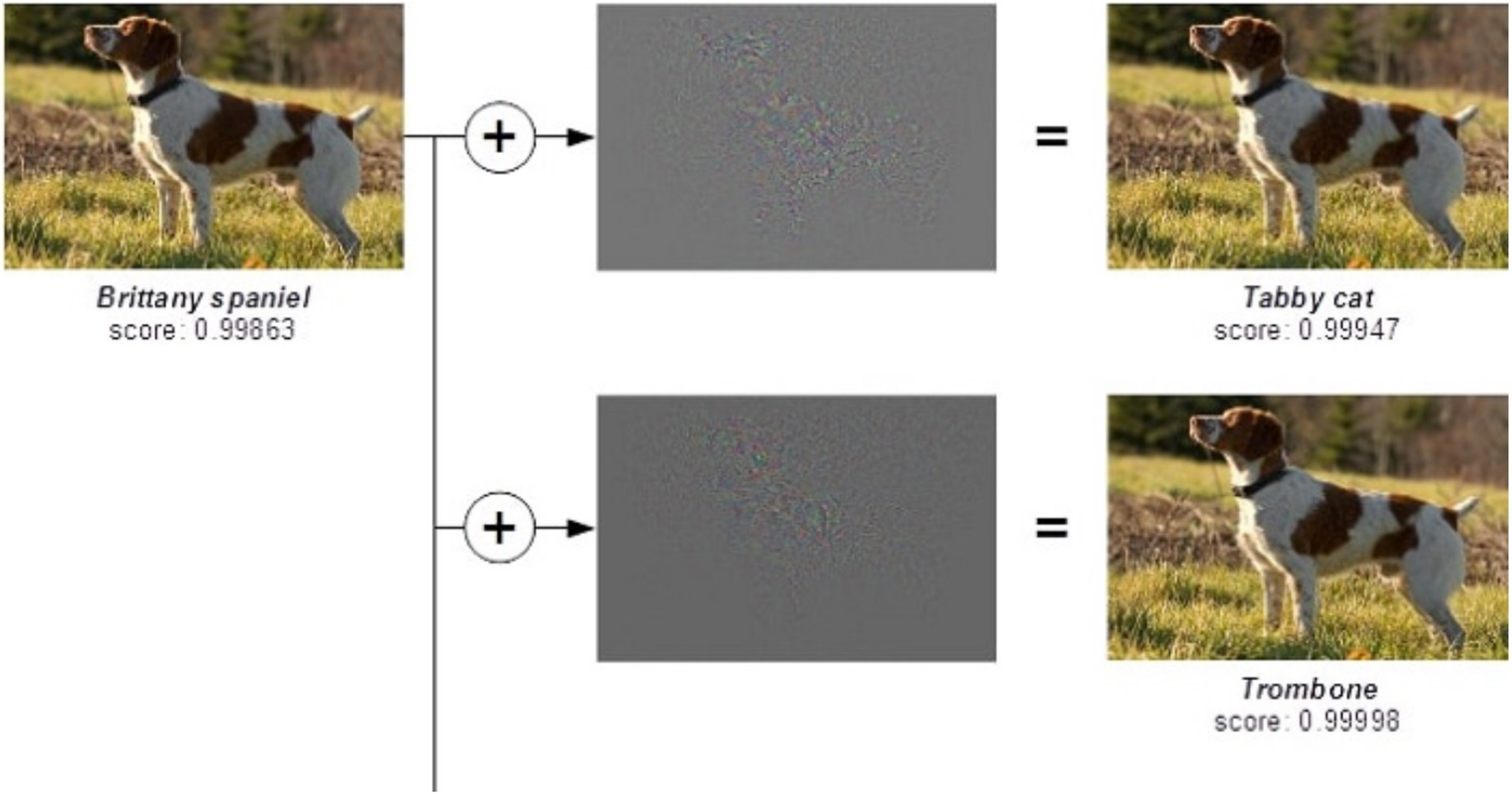
Simulation

Style transfer, to fix simulations ?

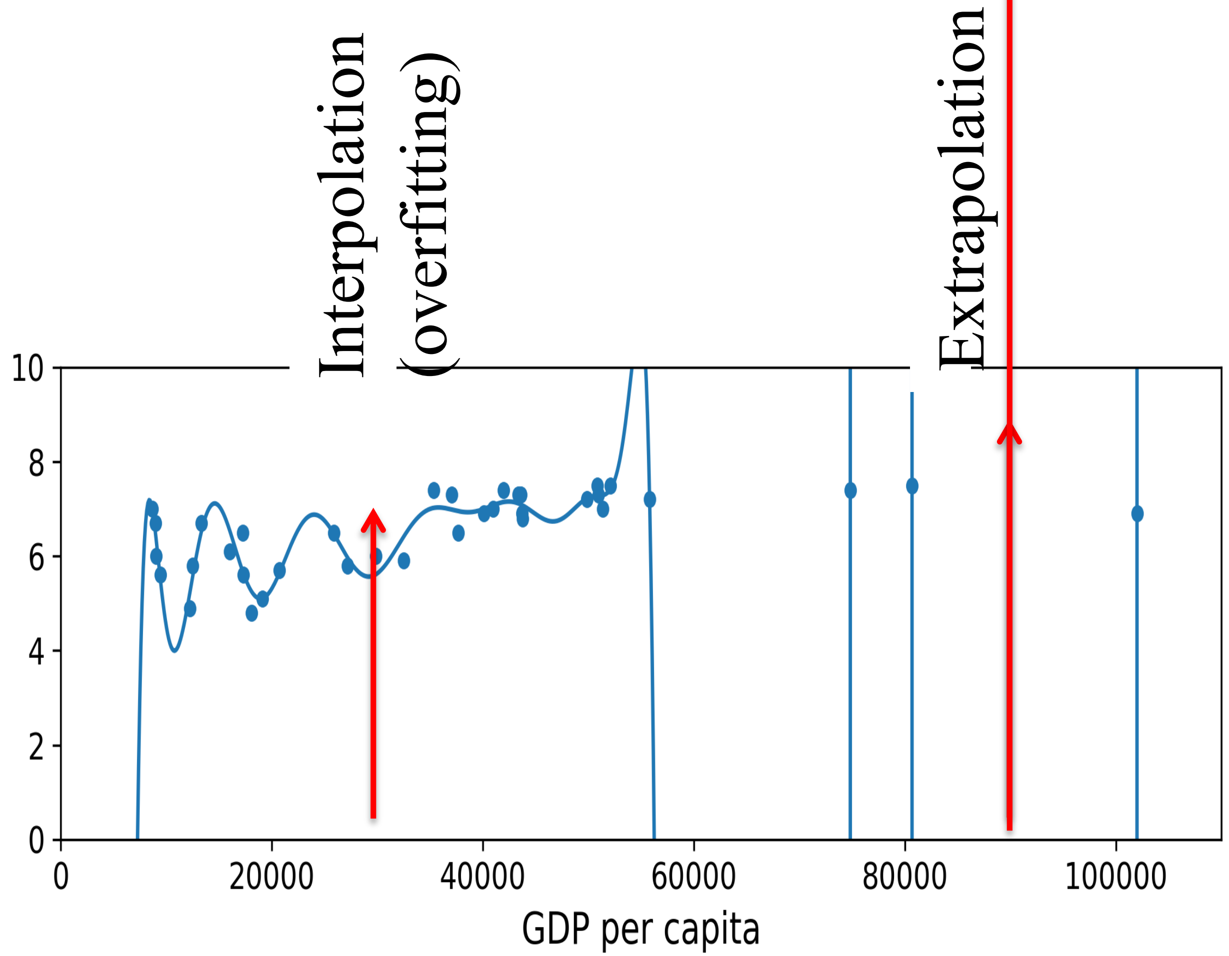
# Adversarial examples

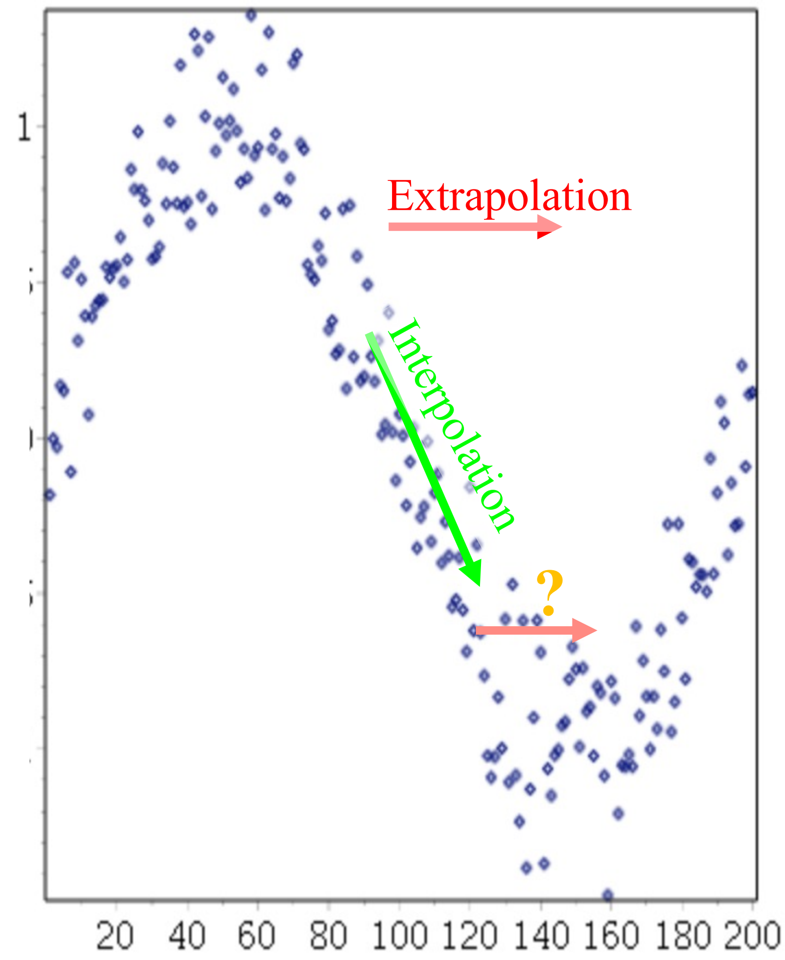
# Adversarial examples

- Subtle alteration of an image fooling a classifier



# Interpolation vs Extrapolation



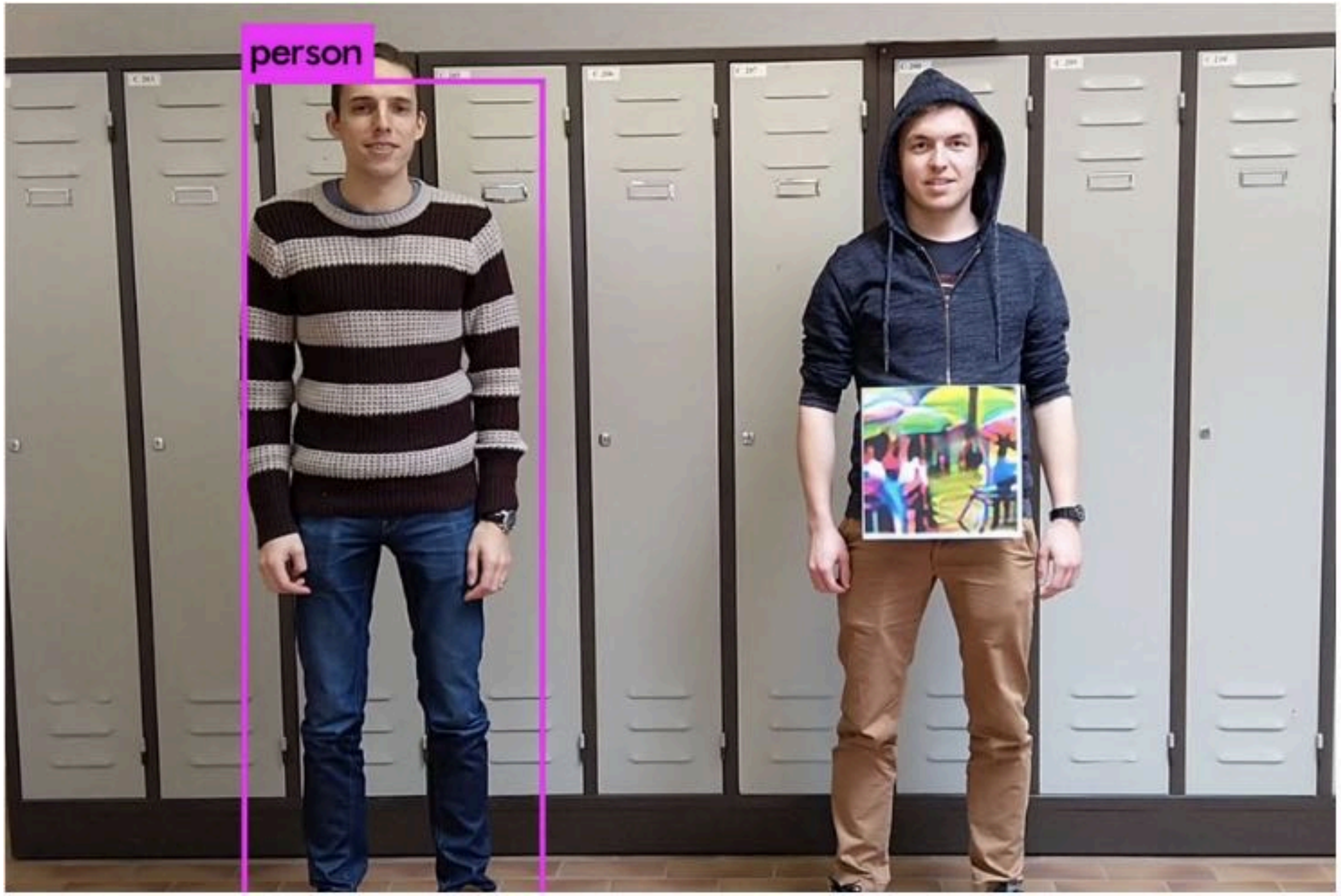




Interpolation/Extrapolation already ill-defined in 2D, what about large dimensions ?

# Adversarial examples (2)

the verge 2019/04/23

- Extraneous object
- ➔more worrying, for HEP it would be e.g. a glitch in the data which is not simulated

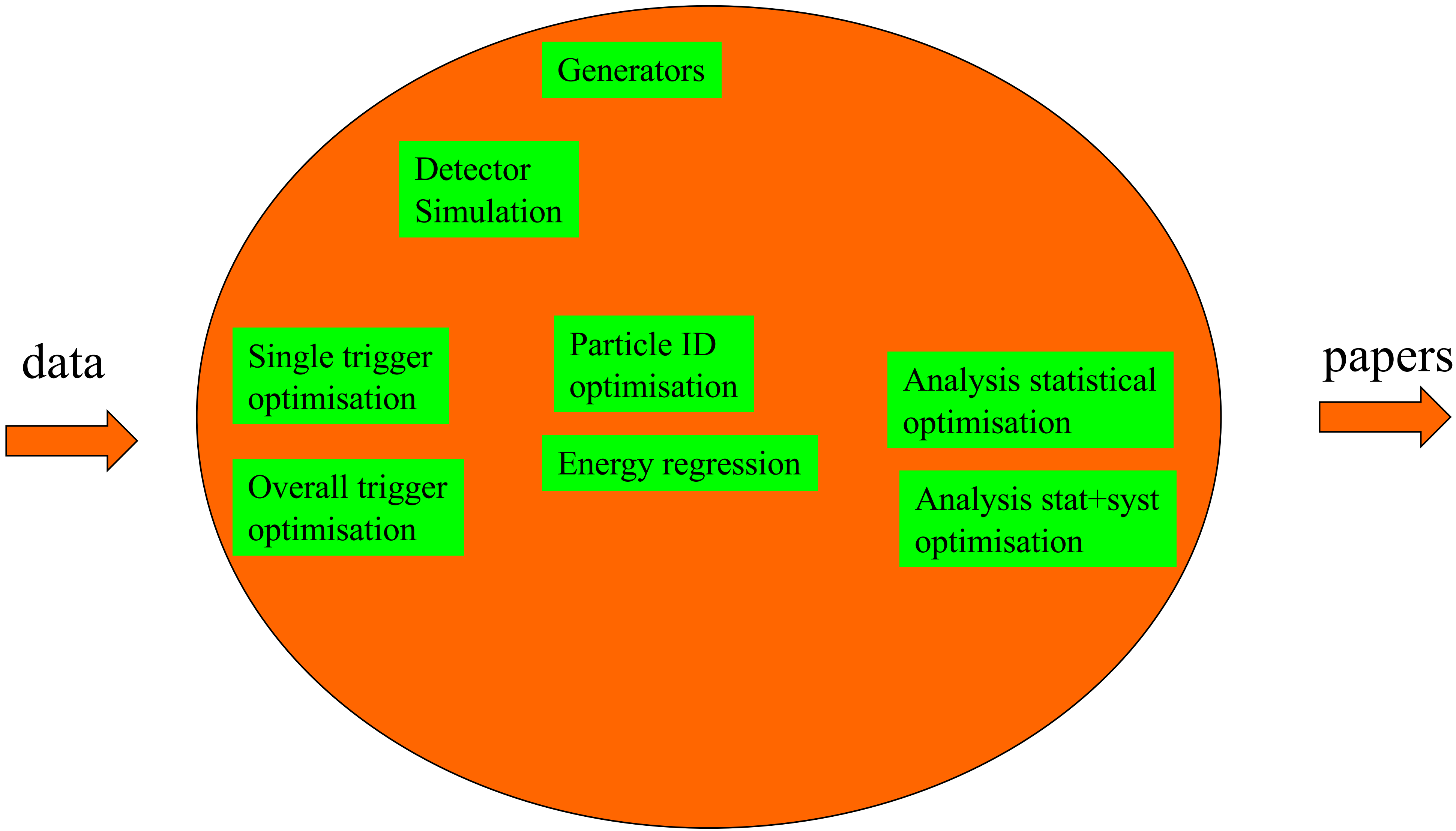


# Adversarial examples

- Like an optical illusion for a NN
- Tune for a specific known NN ➔ fools many NN (share a common behavior)
- Dangerous for physics if we rely more and more on NN ?
- Not really because one has to have a deliberate intent to fool a DNN

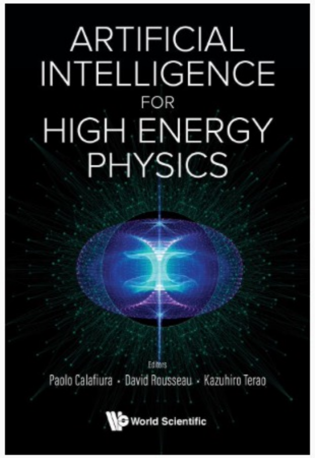
# Wrapping-up

# ML playground



# I want to start with ML where do I start?

- Start with HEP and ML school week online (slides, recording, hands-on) :
  - SOS 2022 (France, in english)
  - HEPML Schoool 2020 next one Apr 2023 in Erice, Italy, no remote participation but updated material will be available
- Available computing resources, GPU ? (laptop is already a very good start)
- Many papers are now releasing code in addition to paper
- The book ($$$ but many chapters on arXiv)




**Artificial Intelligence for High Energy Physics**

https://doi.org/10.1142/12200 | March 2022

Pages: 828

Edited By: Paolo Calafiura (*Lawrence Berkeley National Laboratory, USA*), David Rousseau (*Labor*
*Infinis Irène Joliot-Curie, France*), and Kazuhiro Terao (*SLAC National Accelerator Laboratory, USA*

**View Full Book** **Tools** **Share** **Recommend to Library**

# ML Tool : Root or not Root

- 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: not quite up to date for BDT and Neural Network
- Advice :
  - use uproot (pip install uproot) to read a ntuple into a python notebook (and then write .csv or .h5 file)
  - Then carry on with Xgboost or lightgbm, and python ecosystem scikit-learn, matplotlib etc…
- Note in passing : for (weighted) histogram fitting with proper uncertainties, root is still better than scipy etc… (I would like the pyHep group to take this on board)

# ML Tool : XGBoost

- XGBoost : Xtreme Gradient Boosting : https://github.com/dmlc/xgboost, arXiv: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
- In general, much easier to start with BDT (very fast training and simple to tune), and often sufficient for tabular data

# ML Tool : SciKit-learn

- SciKit-Learn : toolbox for Machine Learning in python
- Open source (several core developers in Paris-Saclay)
- Modern Jupyter interface (notebook à la Mathematica)
- 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

# ML Tool : Neural Networks

- Two major libraries for Neural Networks:
  - TensorFlow (Keras interface) developed by Google
  - pyTorch developed by Meta
  - Both are free open source
  - Both can "talk" to GPU with (in principle) minimal effort (cuda…)

# Why ML for physics is special?

1. Our data are rarely images
2. Often very good (but never perfect) simulation/model
3. Large data and very detailed models: ➔need for speed
4. (almost) all physics papers conclude with a measurement with uncertainty (or Confidence Interval, or p-value…)

➔we are not just taking off-the-shelves tools developed by the GAFAM

➔HEP ML developments relevant to other sciences

# ML in production

- 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 (sklearn, xgboost, Keras, Tensorflow…) are easy to get, but still need (people) training, know-how

- 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

End of Part 3