LABORATÓRIO DE INSTRUMENTAÇÃO
E FÍSICA EXPERIMENTAL DE PARTÍCULAS
partículas e tecnologia

Searching for Rare Events at Colliders Using Deep Learning

LIP Thursday Seminar
2nd July 2020

Miguel Crispim Romão
mcromao@lip.pt

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824093."

POCI/01-0145-FEDER-029147
PTDC/FIS-PAR/29147/2017

Fundação para a Ciência e a Tecnologia

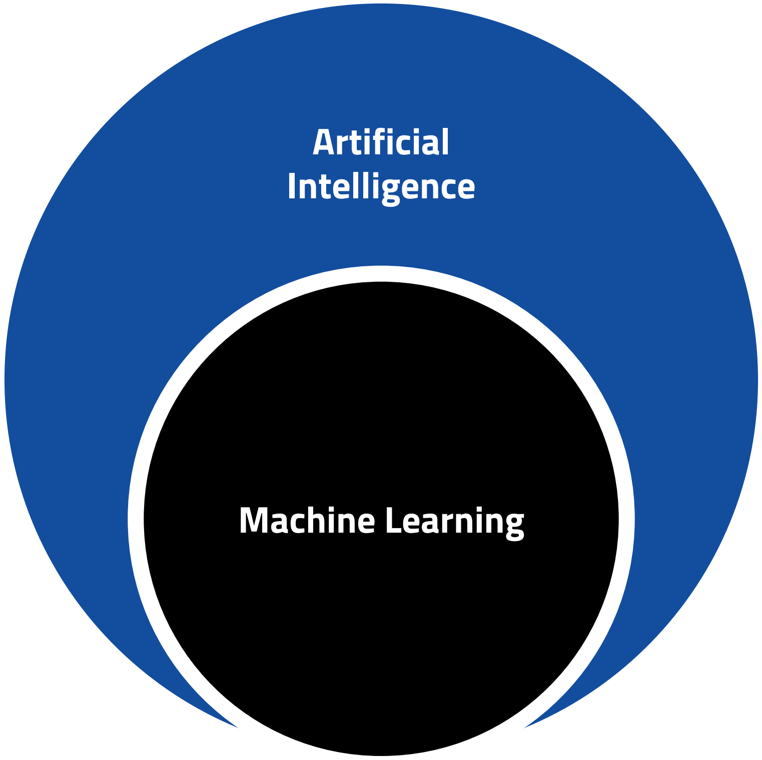
UNIÃO EUROPEIA
Fundo Europeu de Desenvolvimento Regional

Outline

1. Introduction to Machine Learning
2. Deep Learning in Searches for Generic New Physics
3. [WIP] Low Level Data in Rare Phenomena Studies
4. Conclusions

1.
Introduction to Machine Learning

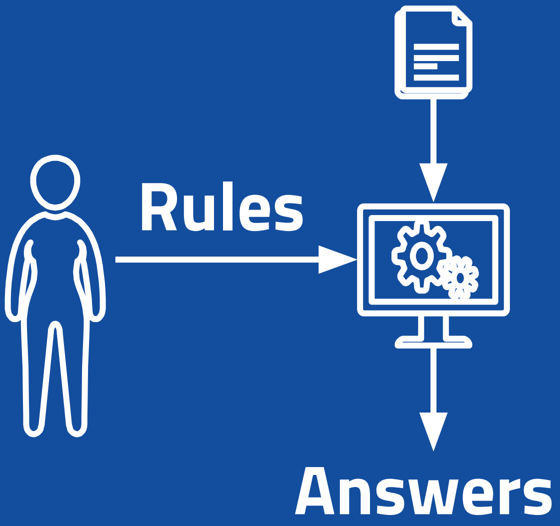
Machine Learning is the subfield of AI that concerns how a machine can learn to perform tasks



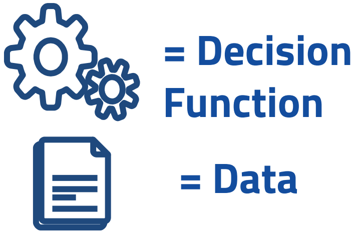
Self-Taught Code

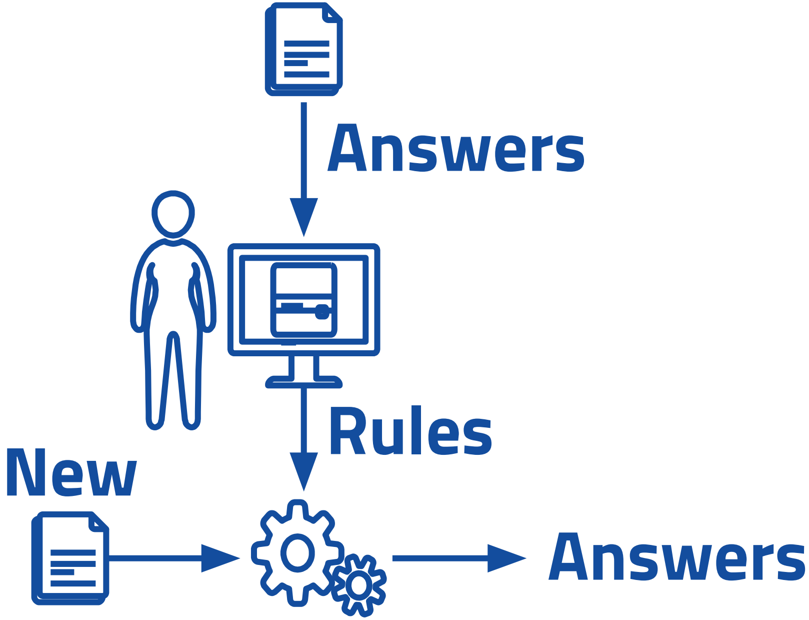
Machine Learning is a different paradigm of computing: a program that learns what it has to do

Classical Programming

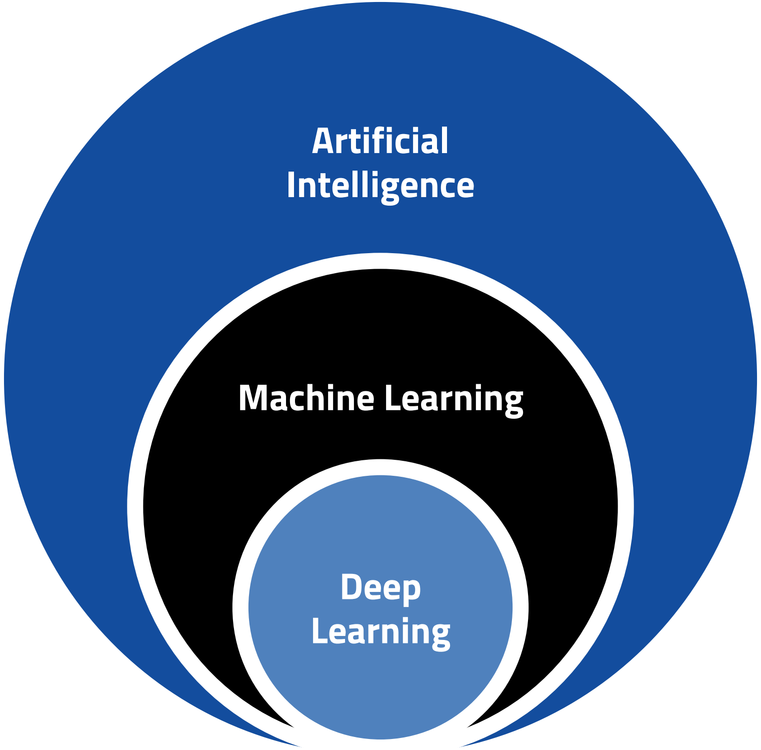


Machine Learning





Deep Learning is a subclass of Machine Learning algorithms that train Neural Networks to perform tasks



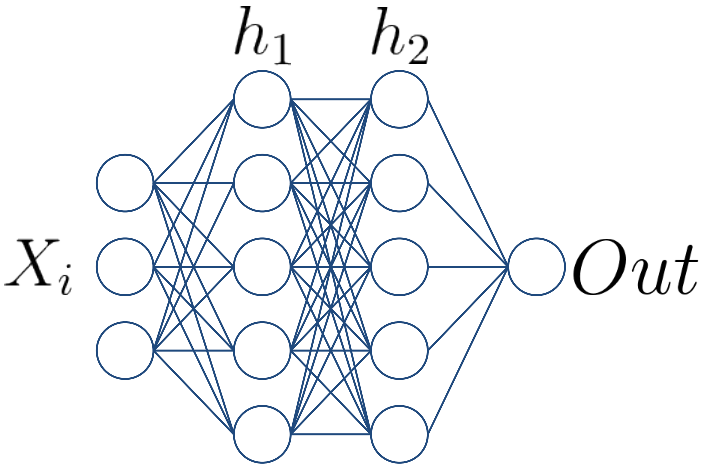
Deep Learning
Neural Networks

Differentiable models that can be trained with **Stochastic Gradient Descent**

Unmatched **representational power** and are capable of **feature abstraction**: deeper layers abstract more complex relations

Extremely versatile and can take in **data of many different shapes and formats**

All state-of-the-art Machine Learning applications are based on Deep Learning and implement Neural Networks



$$\vec{h}_1 = a_1(\mathbf{w_1} \cdot \vec{x} + \vec{b}_1)$$

$$\vec{h}_2 = a_2(\mathbf{w_2} \cdot \vec{h}_1 + \vec{b}_2)$$

$$Out = a_{Out}(\vec{w}_{Out} \cdot \vec{h}_2 + b_{Out})$$

$$a_i = \{\tanh, \sigma, \mathrm{ReLu}, \dots\}$$

$$NN = Out \circ \vec{h}_2 \circ \vec{h}_1$$

Deep Learning
Loss Function

Loss function is a **differentiable measure on how the model is performing** the task and it works as the **minimisation objective in terms of the NN parameters**

Common Loss functions:

- Regression: Mean Squared Error

$$L = \frac{1}{N}\sum_{i}^{N}(y_i - \mathrm{NN}(x_i))^2$$

- Classification: Cross-Entropy

$$L = -\frac{1}{N}\sum_{i}^{N}\sum_{k}^{K} y_{i,k} \log \mathrm{NN}_k(X_i)$$

Data in HEP

In HEP we deal with **huge amounts of data**, comparable only to the big 5 (Microsoft, Google, Amazon, Apple, Facebook), and often **come in many shapes**

In this talk we will focus on two main types of data being collected or simulated at collider experiments:

- ➢ High-level physical observables from reconstructed objects
  - ○ Used in New Physics searches

- ➢ Low-level detector information from calorimeter deposits and tracks
  - ○ Carry no human bias but difficult to handle and use

2.
Deep Learning in Searches for Generic New Physics

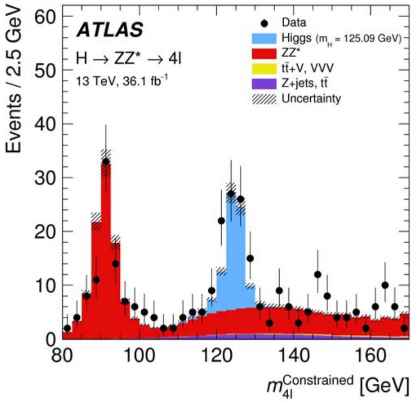
One of the main goals of the LHC is to look for New Physics

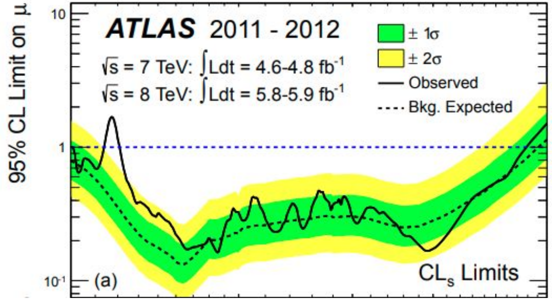
1. Choose BSM signal you are looking for
2. Study favourable kinematic region and final state topology
3. Collect the data in such regime
4. Perform statistical tests on the data on the hypothesis of BSM being present
5. Profit (eventually)

Searches for New Physics
Operationally

- Data composed of high-level variables of reconstructed objects
- Isolate as much signal from background as possible
  - Rule of Thumb: Look for a "signal region" that maximises significance
- Perform the so-called limits, e.g. by employing the CLs method

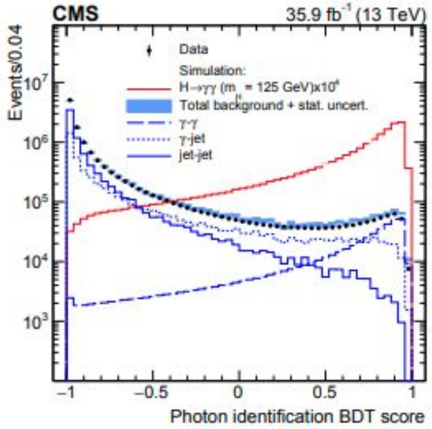
$$Sig \sim \frac{N_S}{\sqrt{N_B}}$$





Searches for New Physics
Machine Learning

- Using Monte-Carlo pseudo-data of Background (SM) and Signal (BSM) we can train a Classifier to isolate signal to increase sensitivity
- Either use a cut-off on the ML classifier outputs or perform CLs directly on its outputs



Transferability of Deep Learning Models in Searches for New Physics at Colliders

MCR, N. F. Castro, R. Pedro, T. Vale

Phys.Rev.D 101 (2020) 3, 035042 [1912.04220]

- How does an NN classifier, trained to separate a specific signal from background, behave when shown a new signal?
- How does this impact upper limits on New Physics?
- Focused on three classes of signals:
  - FCNC
  - VLQ from SM production
  - VLQ from Heavy Gluon production

Transferability of Deep Learning Models
Analogy

Jungle is the Background (SM events) and we want to find monkeys (a BSM candidate)

What happens if instead of monkeys there is another animal in the data?

Would an NN still find the signal?

Transferability of Deep Learning Models
The Background

- A SM cocktail sample was produced in MadGraph5+Pythia8+Delphes
  - 8M Z+J, 3M ttbar, 1.5M per diboson sample
- Targeted processes with dilepton final state, at least one b, and HT > 500
- To guarantee statistics at the tails of the distributions we applied event filter at parton level in pT slices
- The events are represented by variables from the reconstructed objects:
  - (η, φ, pT , m) for 5 leading jets and large-radius jets
  - (η, φ, pT) of the 2 leading electrons and muons
  - Multiplicites, (ET , φ) of the missing transverse energy (MET)

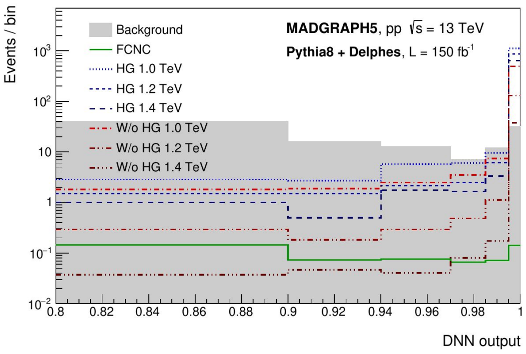
Transferability of Deep Learning Models
The Signals

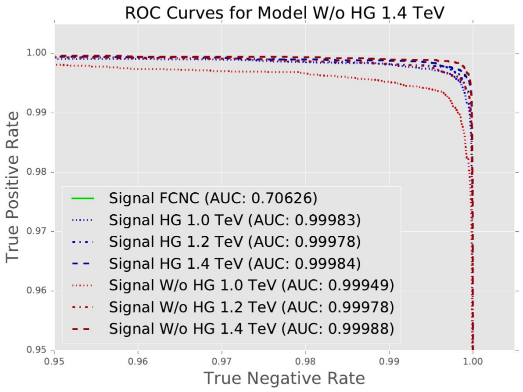
- 7 samples of BSM signals over three classes
- FCNC interaction in single top-quark production
- Vector-Like T quarks produced via SM gluon with three different masses
  - 1.0 TeV
  - 1.2 TeV
  - 1.4 TeV
- Vector-Like T quarks produced via BSM heavy (3TeV) bluon with three different masses
  - 1.0 TeV
  - 1.2 TeV
  - 1.4 TeV

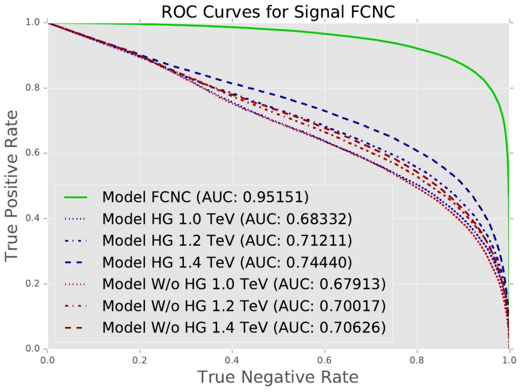
Transferability of Deep Learning Models
Methodology

- For each signal train a supervised DNN classifier
- Use each trained DNN to predict on every combination signal-background
- Assess how discrimination deteriorates as we present a different signal to each DNN through upper limits on expected cross-section

Transferability of Deep Learning Models

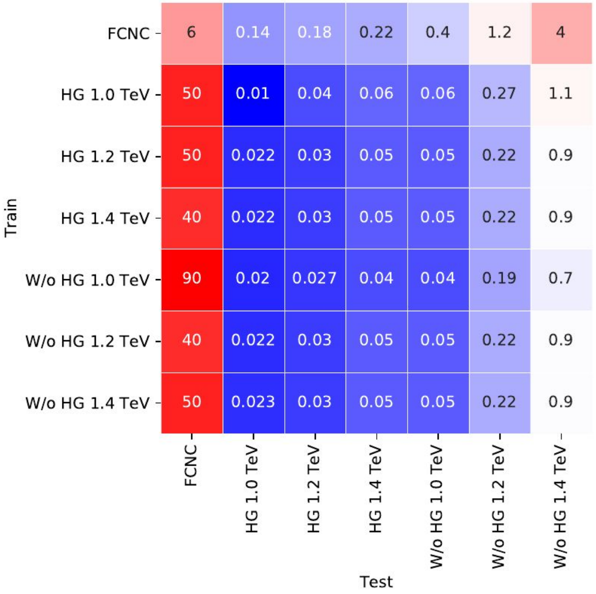


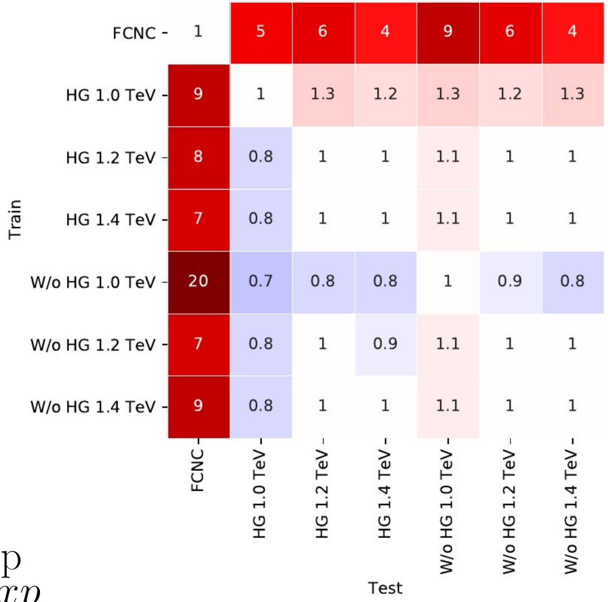




Transferability of Deep Learning Models

Upper Limits





$$\mu = \frac{\sigma_{exp}^{\mathrm{up}}}{\sigma_{th}}$$

Could we not just focus on the jungle?

Since we don't know what BSM candidate is realised in nature, it seems it would be better if we could develop a way of identifying **any type of non SM phenomena**

Unsupervised Methods for New Physics Searches

- Growing interest in Unsupervised approaches to isolate New Physics from SM Background
- Anomaly Detection ML algorithms are finding their way into HEP to help this out
  - 1805.02664, 1808.08992, 1811.10276, 1902.02634, 1903.02032, ...
- A comprehensive live review of ML in HEP curated by CERN's IML WorkGroup: https://github.com/iml-wg/HEPML-LivingReview

Finding New Physics without learning about it: Anomaly Detection as a tool for Searches at Colliders

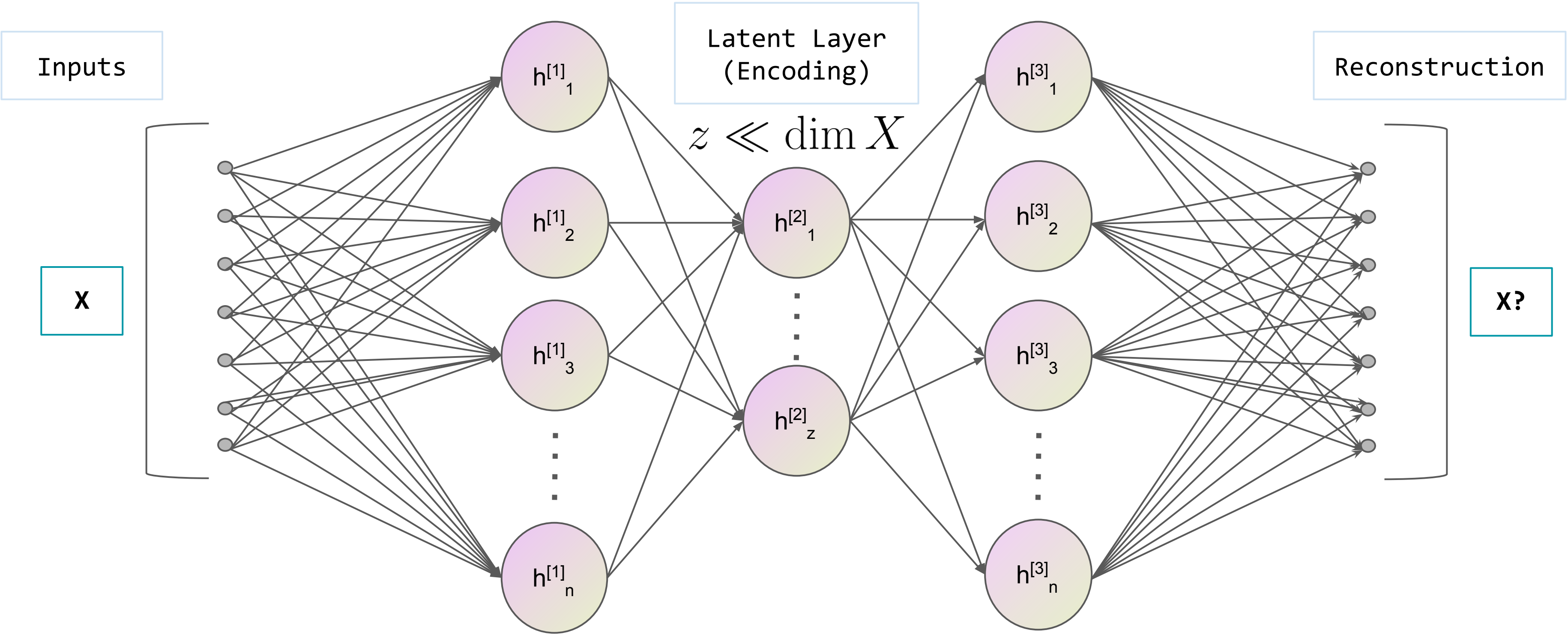
MCR, N. F. Castro, R. Pedro

2006.05432

- We kept the same signals
  - FCNC
  - VLQ from SM production
  - VLQ from Heavy Gluon production
- We compared four AD algorithms
  - Auto-Encoder
  - Deep-SVDD
  - Isolation Forest
  - Histogram Based

Finding New Physics without learning about it

Auto-Encoder



Finding New Physics without learning about it
Auto-Encoder

- The Network is trained by minimising the reconstruction error

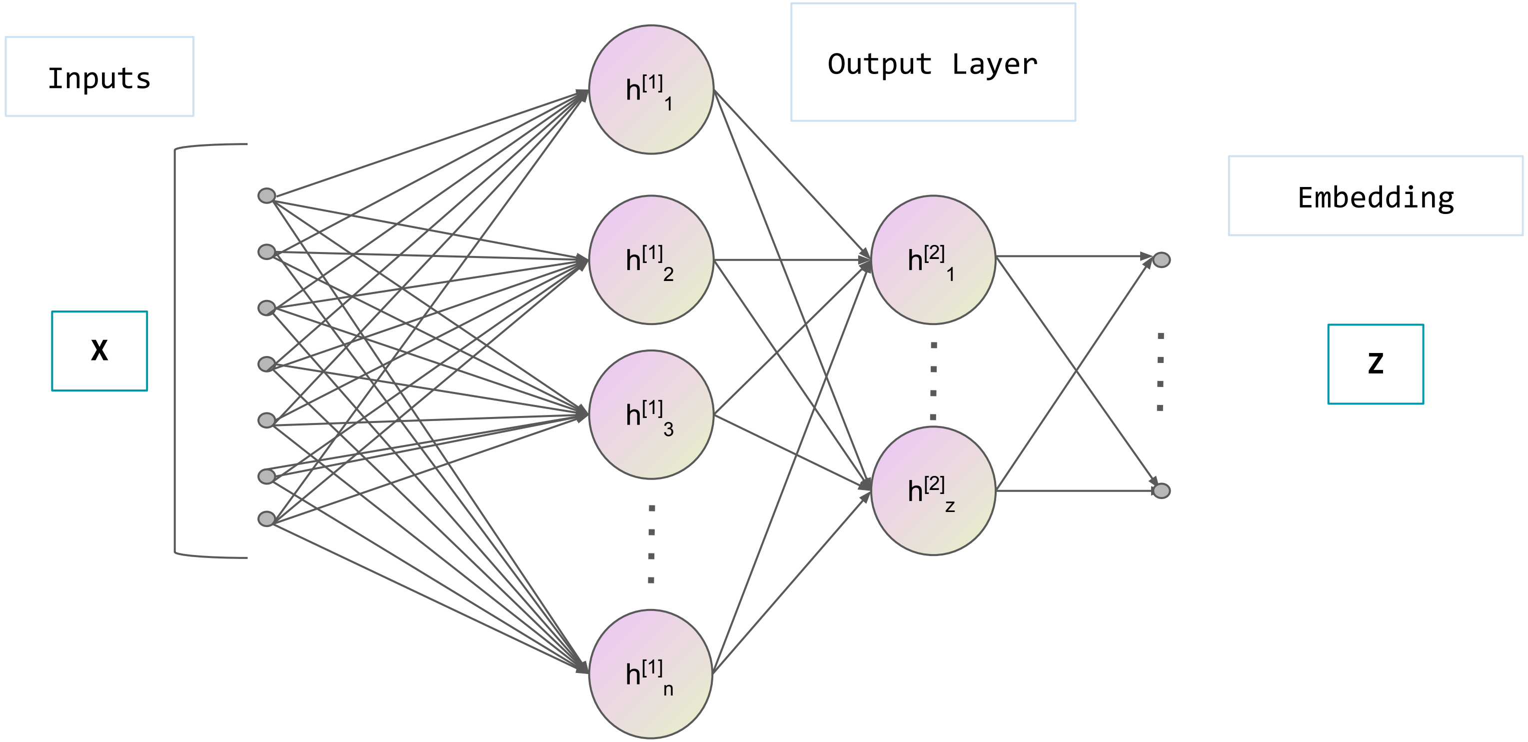
$$L = \frac{1}{N}\sum_{i=1}^{N} |x_i - \mathrm{AE}(x_i)|^2$$

- In principle, events that are easier to reconstruct are the most common

- Reconstruction error of an event can be a measure of how rare it is =>

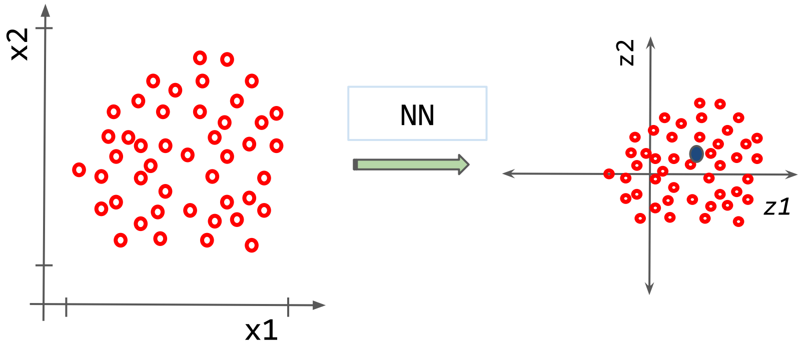
  BSM events should have higher reconstruction error

Finding New Physics without learning about it
Deep-SVDD



Finding New Physics without learning about it
Deep-SVDD

- Before any training, the NN is just a map from the input space to some embedding space



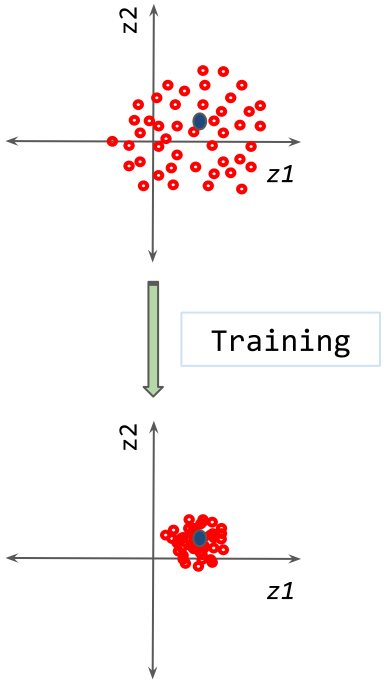
- In this space we can find a “centre of mass”, c, of the points

Finding New Physics without learning about it
Deep-SVDD

- The Network is trained by minimising the distance to the centre of mass

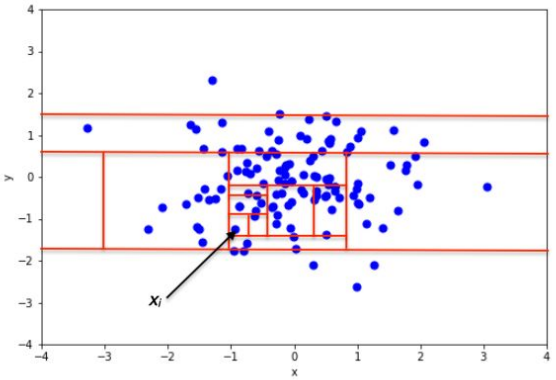
$$L = \frac{1}{N}\sum_{i=1}^{N} |c - \mathrm{NN}(x_i)|^2$$

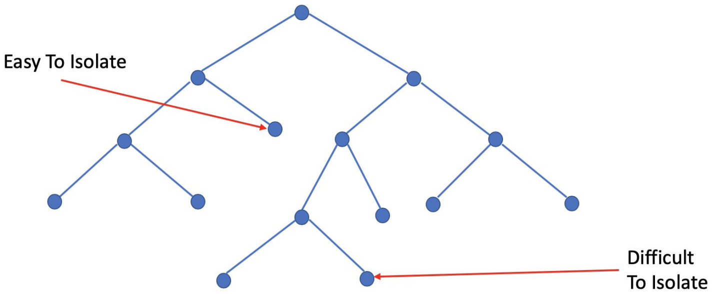
- The bulk of the distribution will be easier to bring to the centre, the rarer events will be further away
- The distance to c becomes then a natural interpretation for *outlyingness* of an event



Finding New Physics without learning about it
Isolation Forest

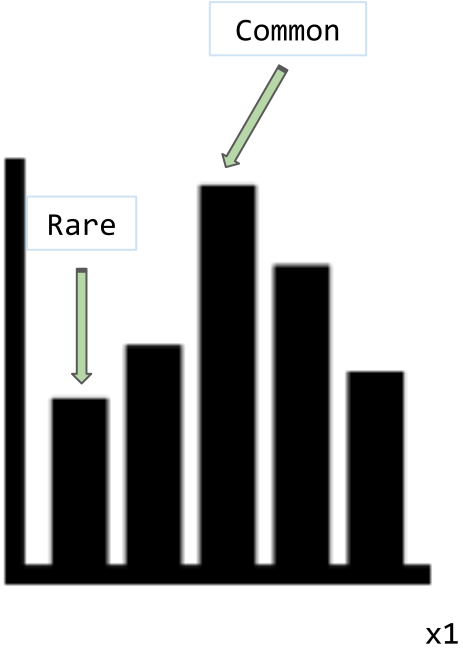
- Recursively partition the data with random cuts
- These cuts can be represented as a tree
- Rare events will be easier to isolate
- Anomaly score given by the inverse of how many nodes it took to isolate





Finding New Physics without learning about it
Histogram Based

- Compute histograms for all variables
- Rare events will more often be in bins of smaller height
- Anomaly score given by the sum of the Log of the heights of each bin an event occupies

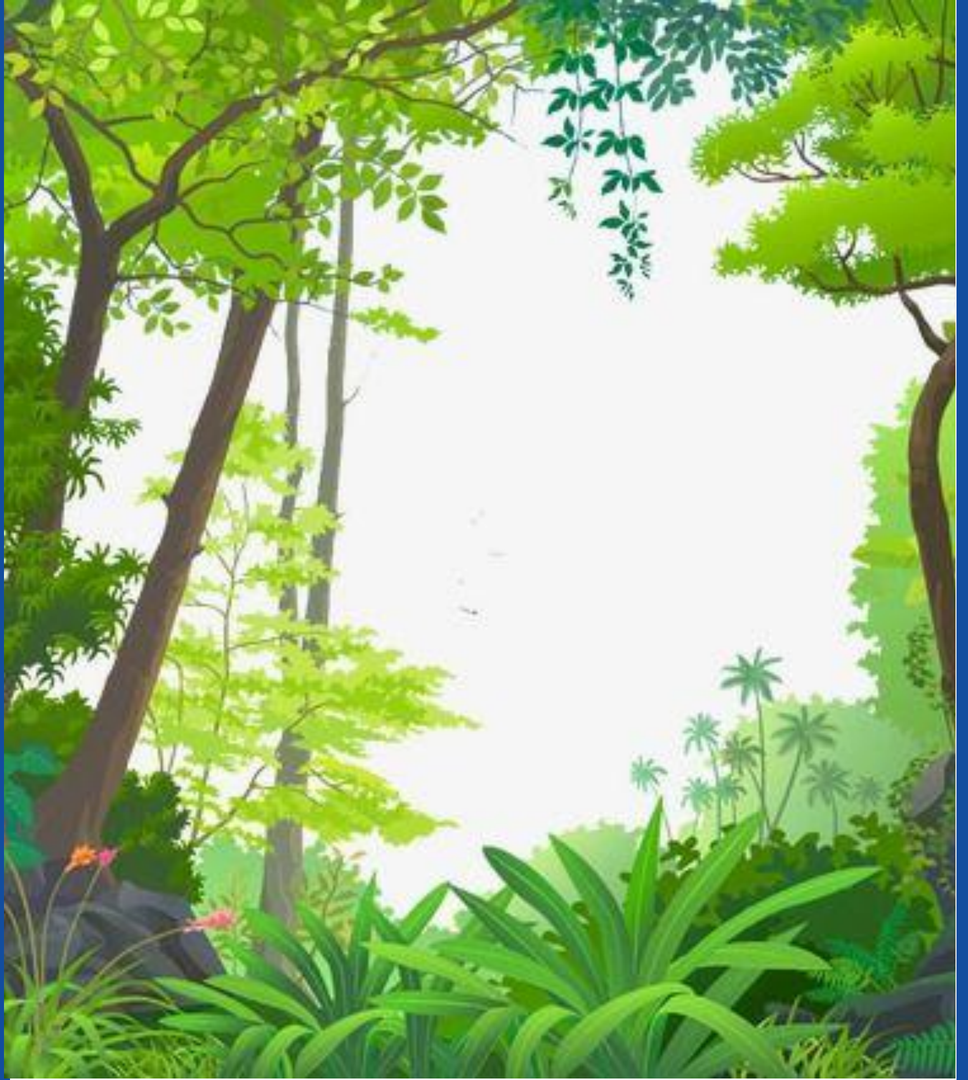


Train only on Standard Model

This way we are learning what a jungle looks like and hopefully we will be able to find any animal!

Are different algorithms correlated?

Are they focusing on the same characteristics?



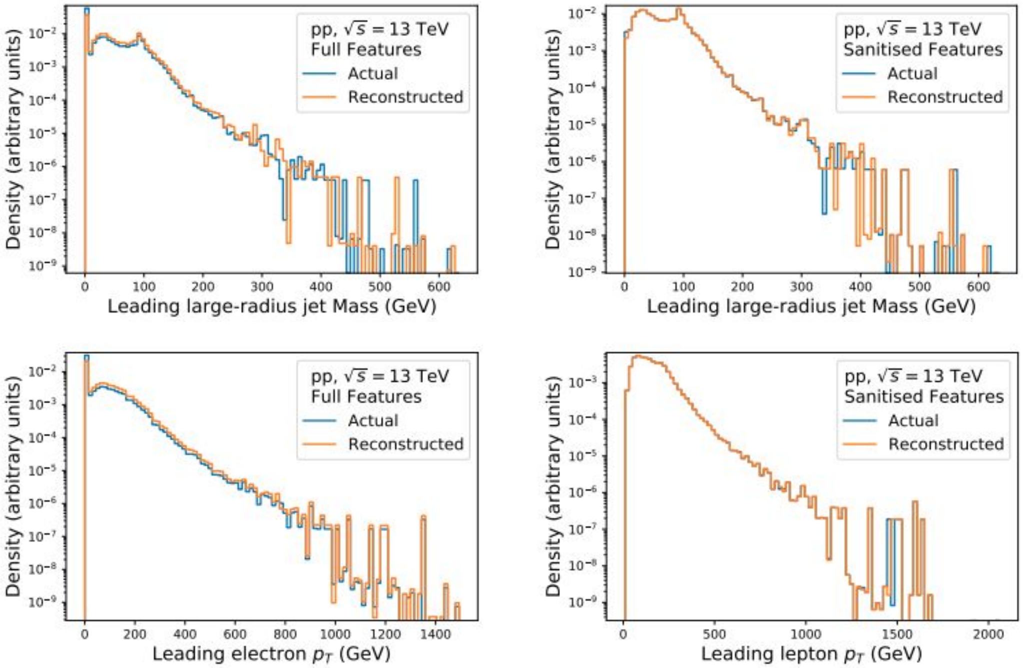
Finding New Physics without learning about it

Sanitisting Features

- Usually, different events have different reconstructed objects => Accumulations of densities in missing values
- NNs do not like discontinuous inputs => This can hinder performance
- We prepared a second set of features which attempts to mitigate this issue
    - Demand a FatJet and drop all the remainder
    - Keep only two leading leptons regardless of flavour

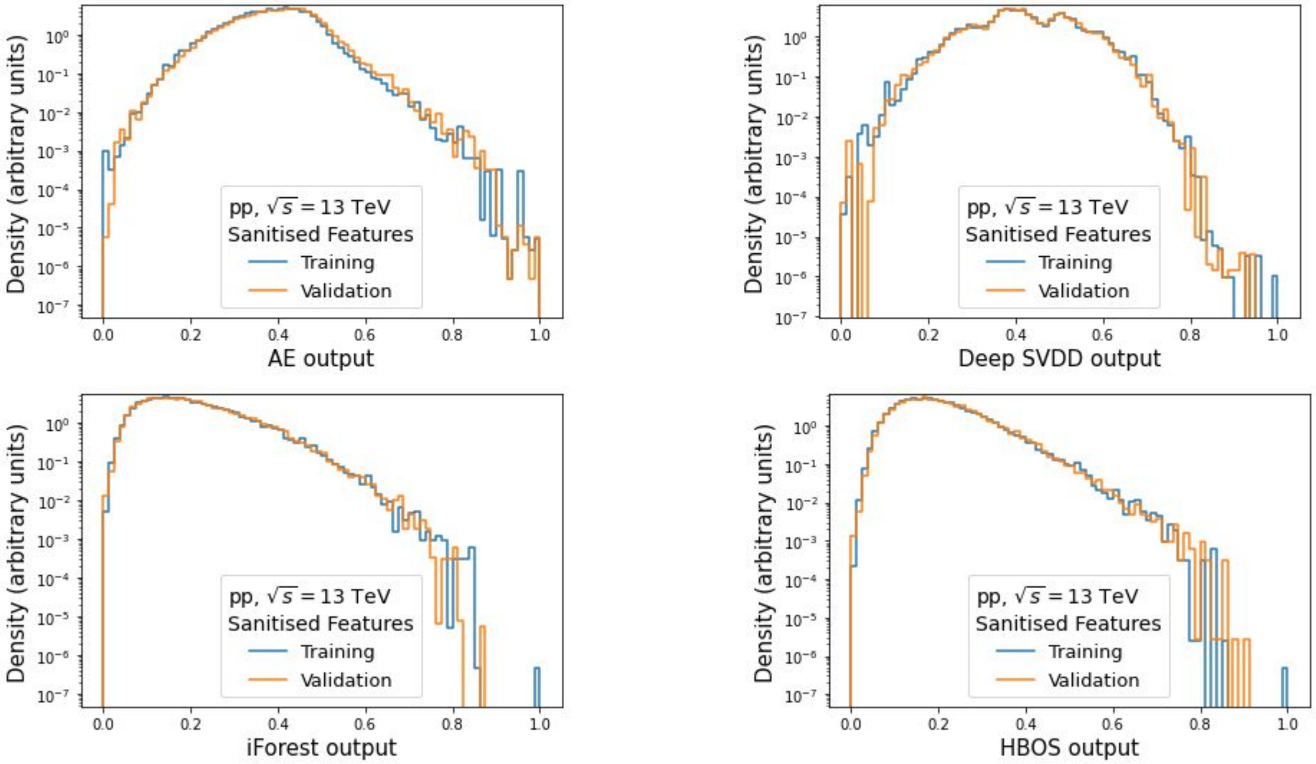
Finding New Physics without learning about it

Sanitisting Features

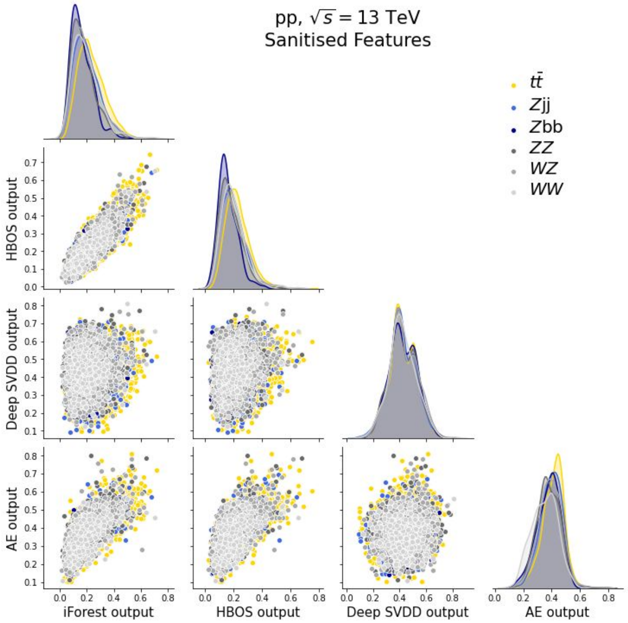


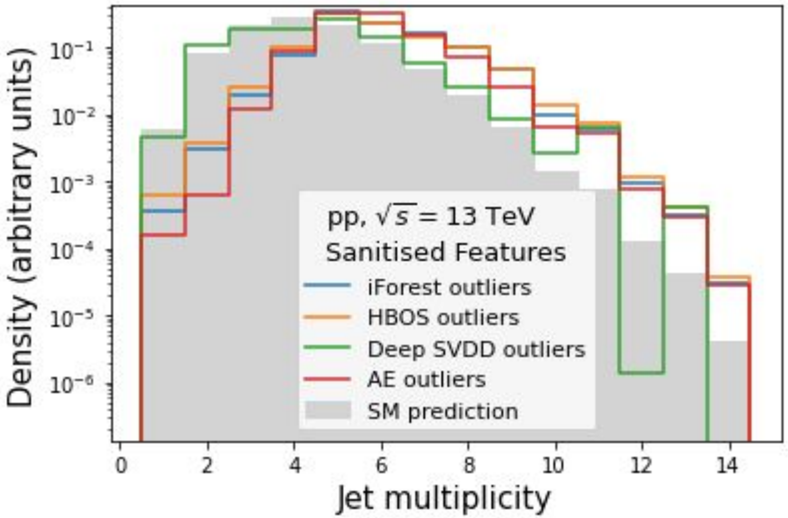
Finding New Physics without learning about it

Results 1: When they see new jungle

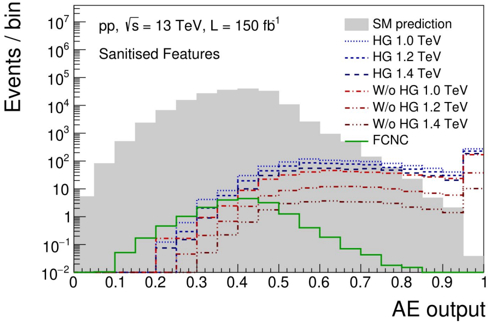


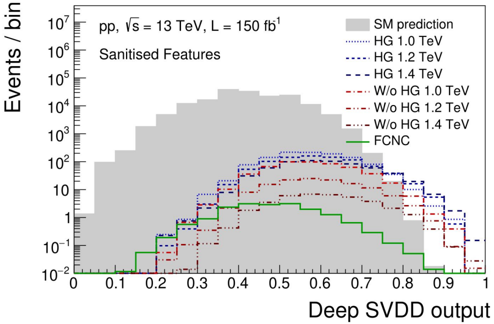
Finding New Physics without learning about it
Results 2: Are all AD algorithms created equally?

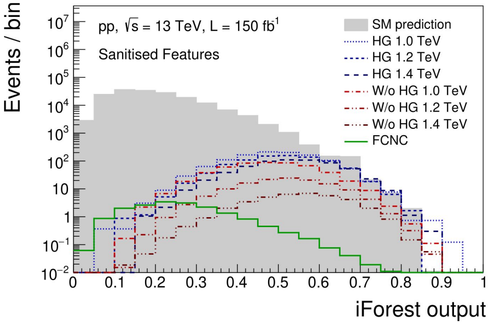


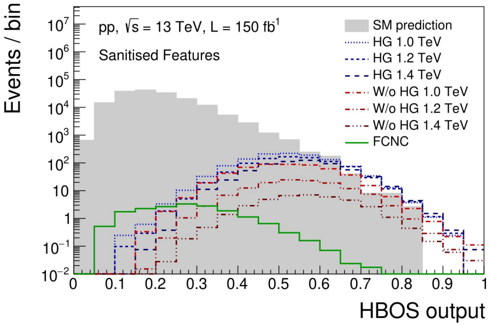


Finding New Physics without learning about it
Results 3: Can they find animals?

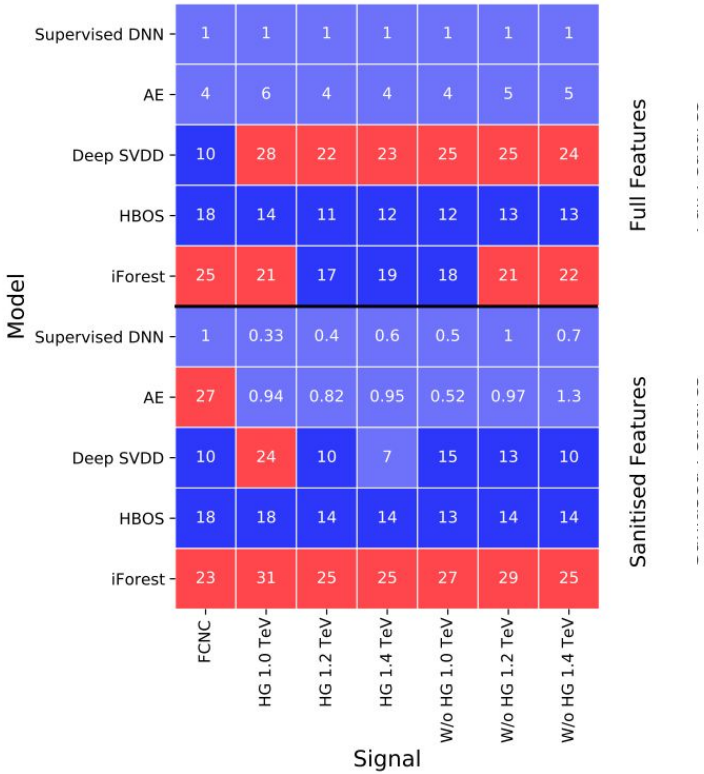








Finding New Physics without learning about it
Results 4: Can we search for New Physics?



Conclusions for this part

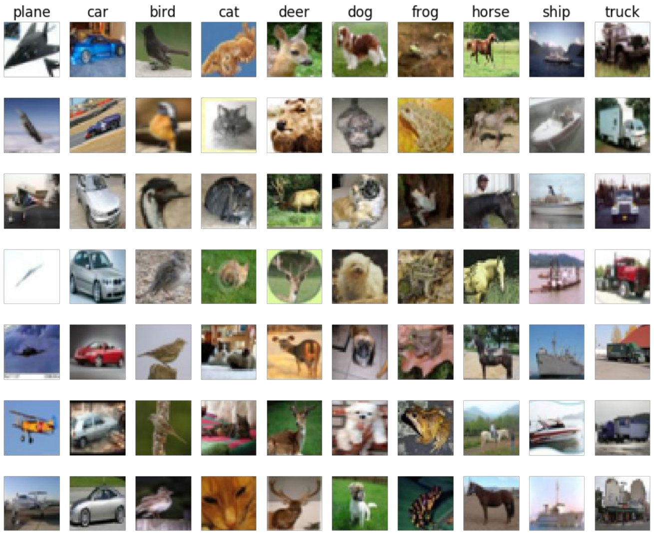
- NN provide very versatile solutions for generic searches
  - Supervised NN classifiers are able to find other signals
  - Unsupervised architectures provide at most an order of magnitude of degradation in sensitivity against supervised
- Unsupervised methods are getting a lot of attention and interest in the community and can provide a BSM independent solution to search for NP
- Future work:
  - Extend to different kinematic and topological regimes
  - Further diversify to other BSM benchmarks
  - Switch to completely unsupervised statistical tests

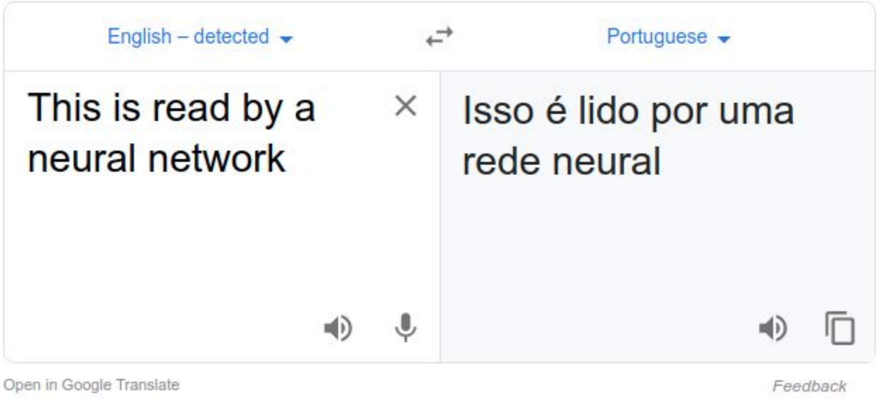
3.

[WIP] Low Level Data in Rare Phenomena Studies

Deep Learning is Versatile

So far we have used tabular data, but Neural Networks can intake many other data formats



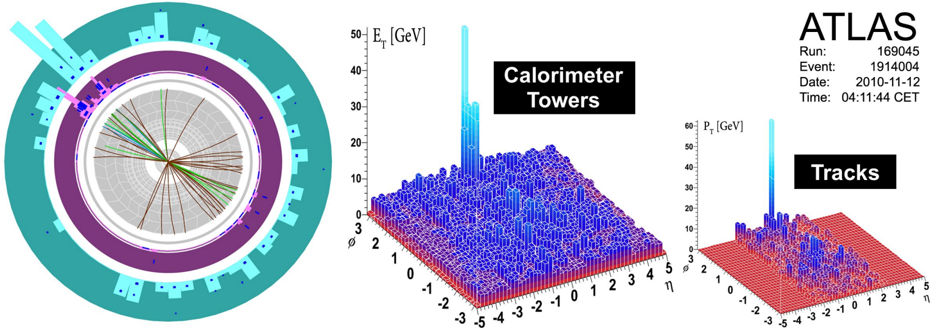


Unstructured Data at the LHC

Before object reconstruction, all we have is low-level data from the detector

This allows us to study Jets in great detail

http://collider.physics.ox.ac.uk/detecting.html
https://cds.cern.ch/record/1309851/plots



Jet Images

- Using the calorimeter deposits we can produce a Jet image
- Deep Learning architectures designed for Computer Vision can perform tasks over images
- 1511.05190, Jet-images — deep learning edition, used such techniques to discriminate between W bosons from QCD multijet processes
- 1612.01551, Deep learning in color, used similar approach to discriminate between Quark and Gluon initiated Jets

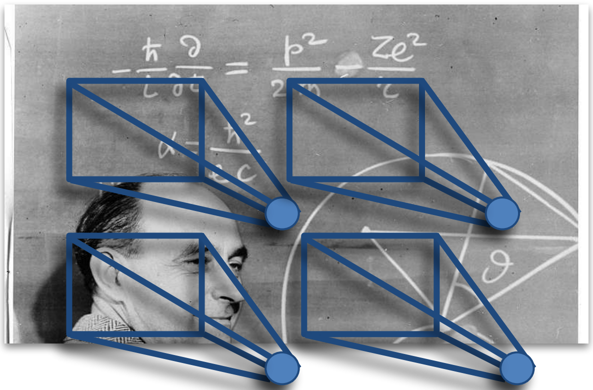
[WIP] Quenched Jet Images

- In the presence of the Quark-Gluon Plasma, the Jet is created and travels through a medium with which it interacts
- The modification of the Jet history due to this is known as **Jet Quenching** and it is still ill-understood
- Can we use similar ideas to further our understanding?
- In collaboration with Guilherme Milhano, Liliana Liliana Apolinário, Filipa Peres, Nuno Castro

[WIP] Quenched Jet Images

Deep Learning Methods for QGP studies

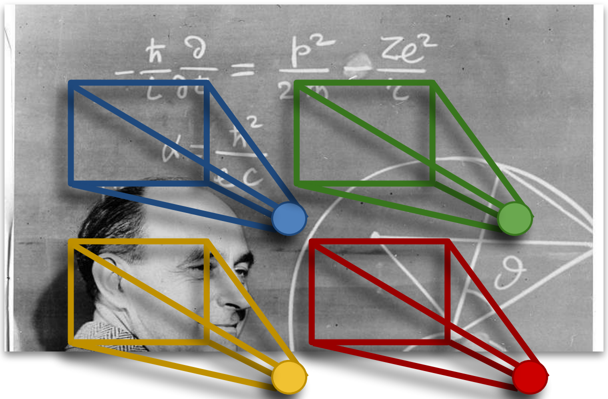
Convolutional Neural Networks



Receptive field moves along the picture =>

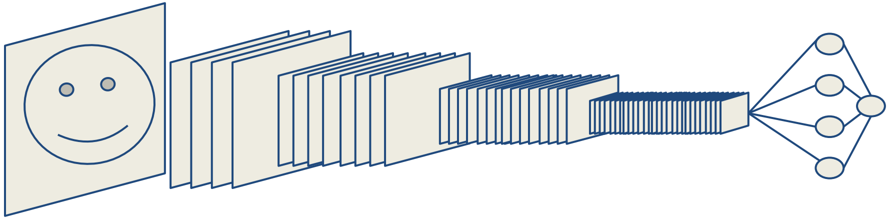
Position Invariant Features

Locally Connected Neural Networks



Receptive field fixed for each region of the

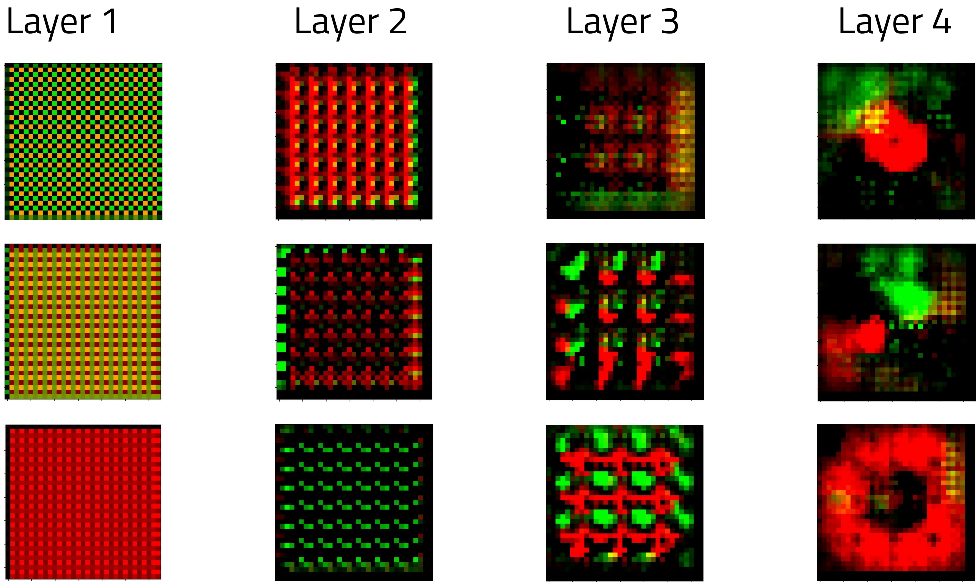
picture => Position Dependent Features



[WIP] Quenched Jet Images

Deep Learning Methods for QGP studies

CNN (receptive field composed of 3x3 filters with stride of 2)



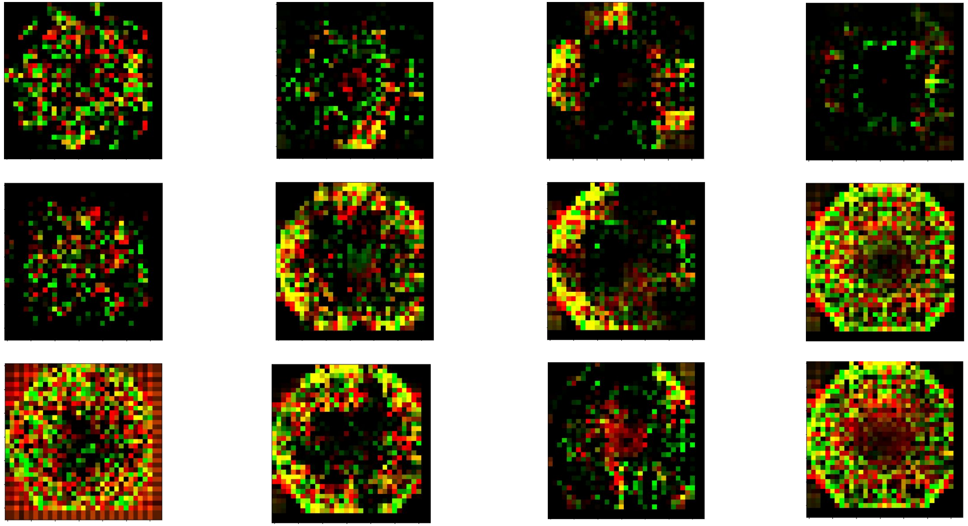

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 33, 33, 2)]       0
_________________________________________________________________
batch_normalization (BatchNo (None, 33, 33, 2)         8
_________________________________________________________________
conv2d (Conv2D)              (None, 16, 16, 128)       2432
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 16, 16, 128)       0
_________________________________________________________________
spatial_dropout2d (SpatialDr (None, 16, 16, 128)       0
_________________________________________________________________
batch_normalization_1 (Batch (None, 16, 16, 128)       512
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 7, 7, 256)         295168
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 7, 7, 256)         0
_________________________________________________________________
spatial_dropout2d_1 (Spatial (None, 7, 7, 256)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 7, 7, 256)         1024
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 3, 3, 384)         885120
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 3, 3, 384)         0
_________________________________________________________________
spatial_dropout2d_2 (Spatial (None, 3, 3, 384)         0
_________________________________________________________________
batch_normalization_3 (Batch (None, 3, 3, 384)         1536
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 1, 1, 512)         1769984
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 1, 1, 512)         0
_________________________________________________________________
spatial_dropout2d_3 (Spatial (None, 1, 1, 512)         0
_________________________________________________________________
batch_normalization_4 (Batch (None, 1, 1, 512)         2048
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 513
=================================================================
Total params: 2,958,345
Trainable params: 2,955,781
Non-trainable params: 2,564
```

[WIP] Quenched Jet Images

Deep Learning Methods for QGP studies

LCNN (receptive field composed of 3x3 filters with stride of 2)

Layer 1 Layer 2 Layer 3 Layer 4



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 33, 33, 2)]	0
batch_normalization (BatchNo	(None, 33, 33, 2)	8
locally_connected2d (Locally	(None, 16, 16, 128)	622592
leaky_re_lu (LeakyReLU)	(None, 16, 16, 128)	0
spatial_dropout2d (SpatialDr	(None, 16, 16, 128)	0
batch_normalization_1 (Batch	(None, 16, 16, 128)	512
locally_connected2d_1 (Local	(None, 7, 7, 256)	14463232
leaky_re_lu_1 (LeakyReLU)	(None, 7, 7, 256)	0
spatial_dropout2d_1 (Spatial	(None, 7, 7, 256)	0
batch_normalization_2 (Batch	(None, 7, 7, 256)	1024
locally_connected2d_2 (Local	(None, 3, 3, 384)	7966080
leaky_re_lu_2 (LeakyReLU)	(None, 3, 3, 384)	0
spatial_dropout2d_2 (Spatial	(None, 3, 3, 384)	0
batch_normalization_3 (Batch	(None, 3, 3, 384)	1536
locally_connected2d_3 (Local	(None, 1, 1, 512)	1769984
leaky_re_lu_3 (LeakyReLU)	(None, 1, 1, 512)	0
spatial_dropout2d_3 (Spatial	(None, 1, 1, 512)	0
batch_normalization_4 (Batch	(None, 1, 1, 512)	2048
flatten (Flatten)	(None, 512)	0
dropout (Dropout)	(None, 512)	0
dense (Dense)	(None, 1)	513

Total params: 24,827,529
Trainable params: 24,824,965
Non-trainable params: 2,564

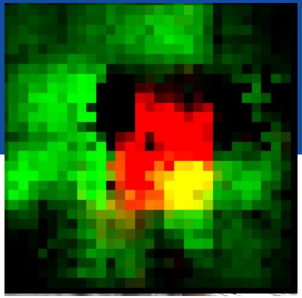
“Conclusions” for this part

- Deep Learning architectures for Computer Vision are able to differentiate quenched from unquenched Jets
- However, the resulting discriminative power is mild and the output of the Neural Networks is not immediately related to mature Jet observables
- We are expanding the study to include other representations of Jets (namely through Lund Plane coordinates) and other architectures

4.
Conclusions

Overarching conclusions

- Deep Learning provides paradigm-shifting possibilities to any data-heavy endeavour and it has finally reached HEP
- It will be at the core of generic searches for New Physics => **Tool for discovery**
- It allows us to use very low-level data, with minimal human bias, which can lead to a deeper understanding of phenomena => **Tool for learning Physics**



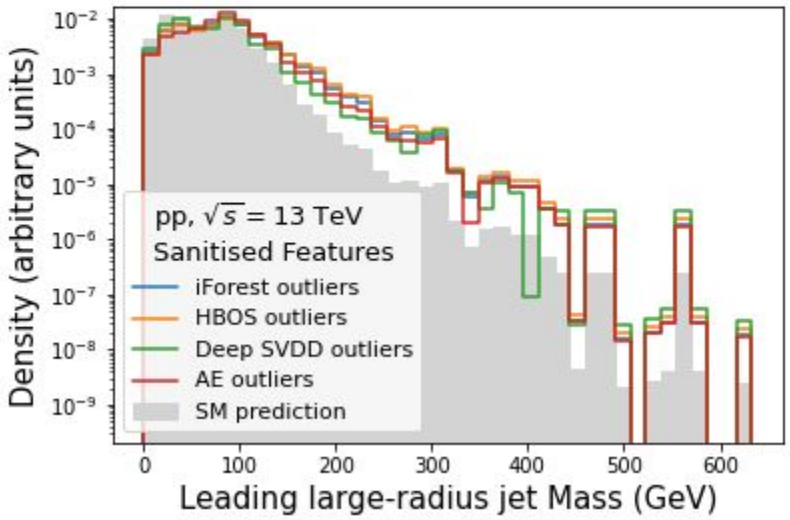
Thanks!

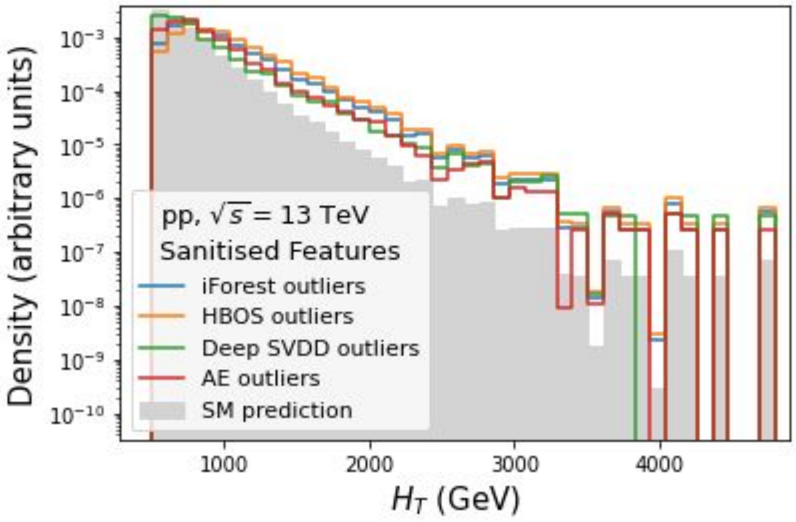
mcromao@lip.pt

n+1.
Backups

Backups

AD outliers are data outliers





Backups
AD mus

Model	Benchmark Signal						
	FCNC	HG			No HG		
		1.0 TeV	1.2 TeV	1.4 TeV	1.0 TeV	1.2 TeV	1.4 TeV
Full features							
Supervised DNN	6^{+3}_{-2}	$0.011^{+0.007}_{-0.004}$	$0.015^{+0.008}_{-0.005}$	$0.016^{+0.009}_{-0.005}$	$0.03^{+0.02}_{-0.01}$	$0.08^{+0.04}_{-0.03}$	$0.20^{+0.12}_{-0.07}$
H_T	110^{+40}_{-30}	$0.14^{+0.07}_{-0.05}$	$0.16^{+0.08}_{-0.06}$	$0.16^{+0.08}_{-0.05}$	$0.4^{+0.2}_{-0.1}$	$1.0^{+0.5}_{-0.3}$	$1.8^{+0.9}_{-0.6}$
Deep SVDD	60^{+30}_{-20}	$0.29^{+0.14}_{-0.09}$	$0.32^{+0.15}_{-0.10}$	$0.4^{+0.2}_{-0.1}$	$0.8^{+0.4}_{-0.2}$	$1.9^{+0.9}_{-0.6}$	5^{+2}_{-1}
AE	30^{+10}_{-10}	$0.06^{+0.04}_{-0.02}$	$0.06^{+0.05}_{-0.02}$	$0.06^{+0.04}_{-0.02}$	$0.12^{+0.08}_{-0.04}$	$0.4^{+0.2}_{-0.1}$	$1.0^{+0.6}_{-0.3}$
HBOS	100^{+40}_{-30}	$0.15^{+0.07}_{-0.05}$	$0.17^{+0.08}_{-0.05}$	$0.19^{+0.09}_{-0.06}$	$0.4^{+0.2}_{-0.1}$	$1.0^{+0.5}_{-0.3}$	$2.7^{+1.2}_{-0.9}$
iForest	200^{+60}_{-40}	$0.22^{+0.11}_{-0.07}$	$0.26^{+0.13}_{-0.09}$	$0.3^{+0.2}_{-0.1}$	$0.6^{+0.3}_{-0.2}$	$1.6^{+0.8}_{-0.6}$	4^{+2}_{-1}
Sanitised features							
Supervised DNN	6^{+3}_{-2}	$0.0035^{+0.0022}_{-0.0009}$	$0.006^{+0.003}_{-0.002}$	$0.009^{+0.004}_{-0.003}$	$0.014^{+0.010}_{-0.005}$	$0.07^{+0.04}_{-0.03}$	$0.15^{+0.09}_{-0.05}$
H_T	100^{+40}_{-30}	$0.14^{+0.07}_{-0.04}$	$0.16^{+0.08}_{-0.05}$	$0.16^{+0.08}_{-0.05}$	$0.4^{+0.2}_{-0.1}$	$1.0^{+0.5}_{-0.3}$	$1.8^{+0.9}_{-0.6}$
Deep SVDD	60^{+30}_{-20}	$0.25^{+0.13}_{-0.08}$	$0.16^{+0.08}_{-0.04}$	$0.12^{+0.05}_{-0.03}$	$0.5^{+0.2}_{-0.1}$	$1.0^{+0.5}_{-0.3}$	$2.0^{+0.8}_{-0.5}$
AE	160^{+60}_{-50}	$0.0099^{+0.0009}_{-0.0007}$	$0.0122^{+0.0006}_{-0.0009}$	$0.0152^{+0.0009}_{-0.0007}$	$0.0165^{+0.0007}_{-0.0011}$	$0.073^{+0.004}_{-0.004}$	$0.27^{+0.02}_{-0.02}$
HBOS	110^{+50}_{-30}	$0.19^{+0.11}_{-0.06}$	$0.21^{+0.12}_{-0.07}$	$0.23^{+0.14}_{-0.08}$	$0.4^{+0.2}_{-0.1}$	$1.1^{+0.7}_{-0.4}$	$2.7^{+1.7}_{-0.9}$
iForest	140^{+60}_{-40}	$0.3^{+0.2}_{-0.1}$	$0.4^{+0.2}_{-0.1}$	$0.4^{+0.2}_{-0.1}$	$0.8^{+0.4}_{-0.3}$	$2.2^{+1.2}_{-0.7}$	5^{+3}_{-2}