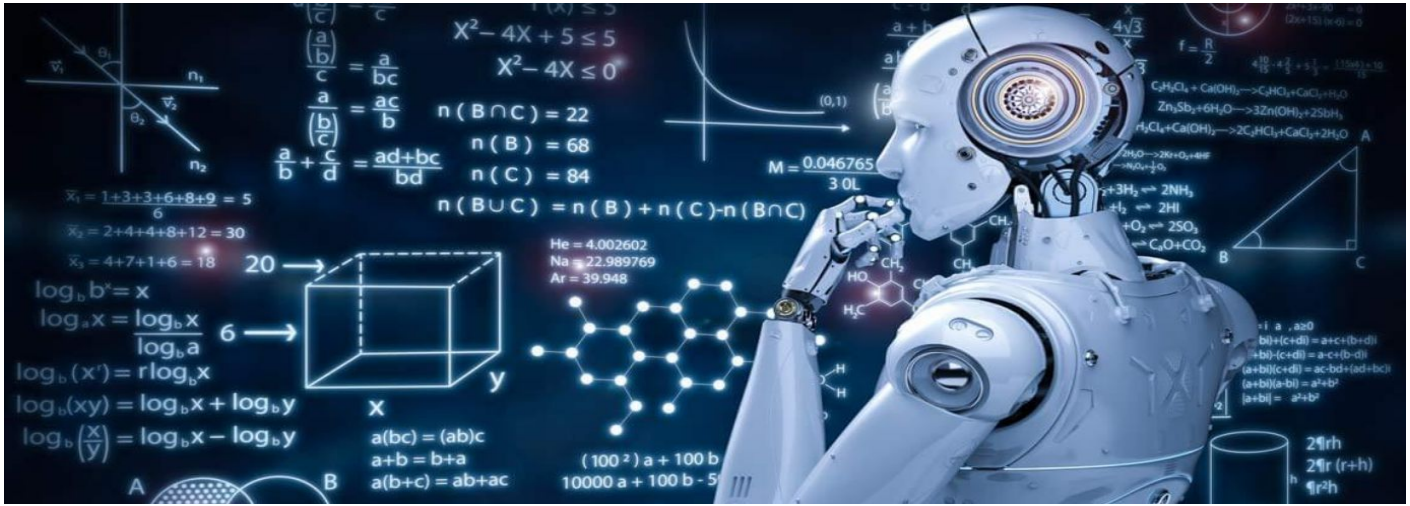
Machine Learning for HEP



Qiang Li Peking University

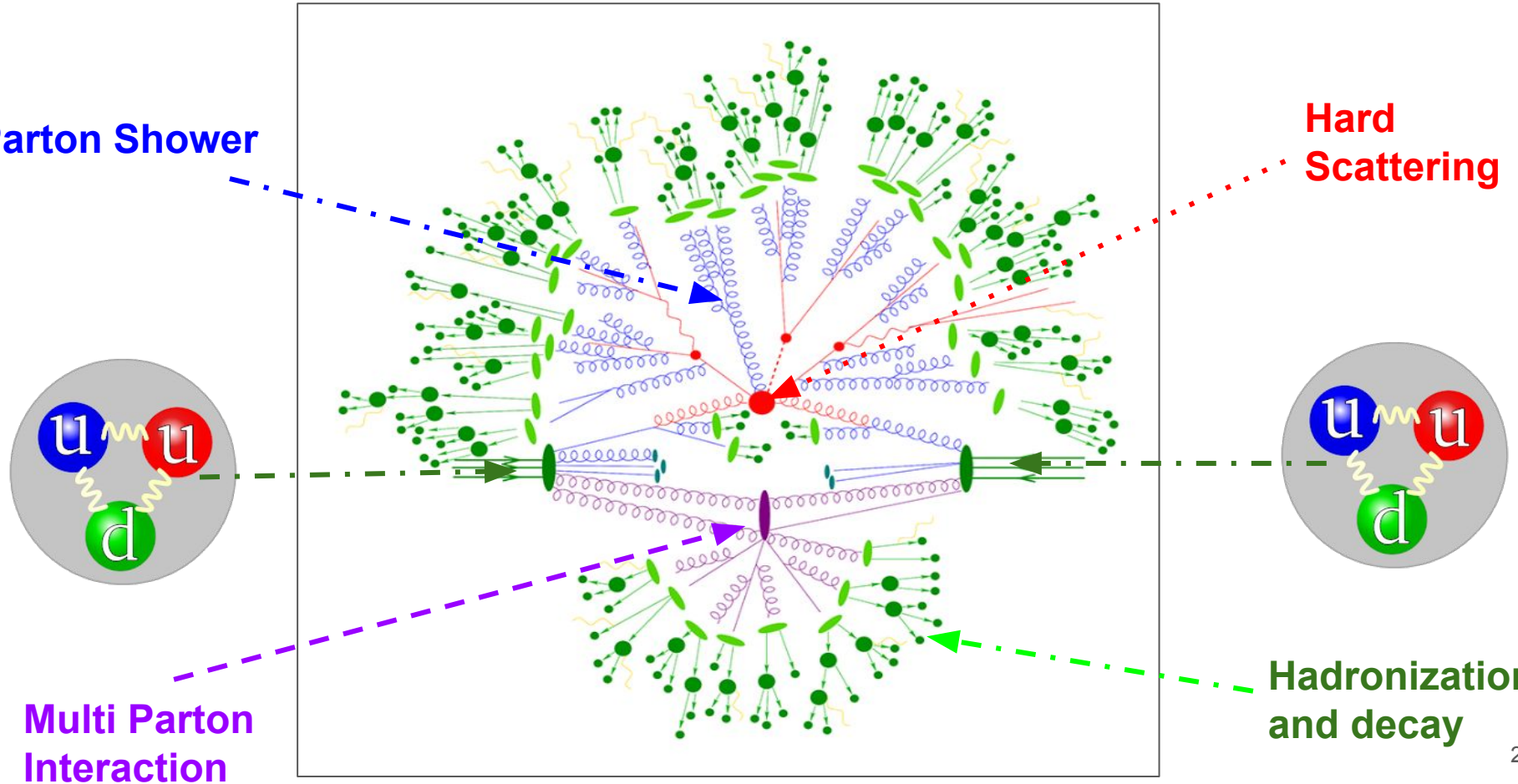
qliphy0@pku.edu.cn 2021/07/02 @IHEP

Some materials from M. Pierini

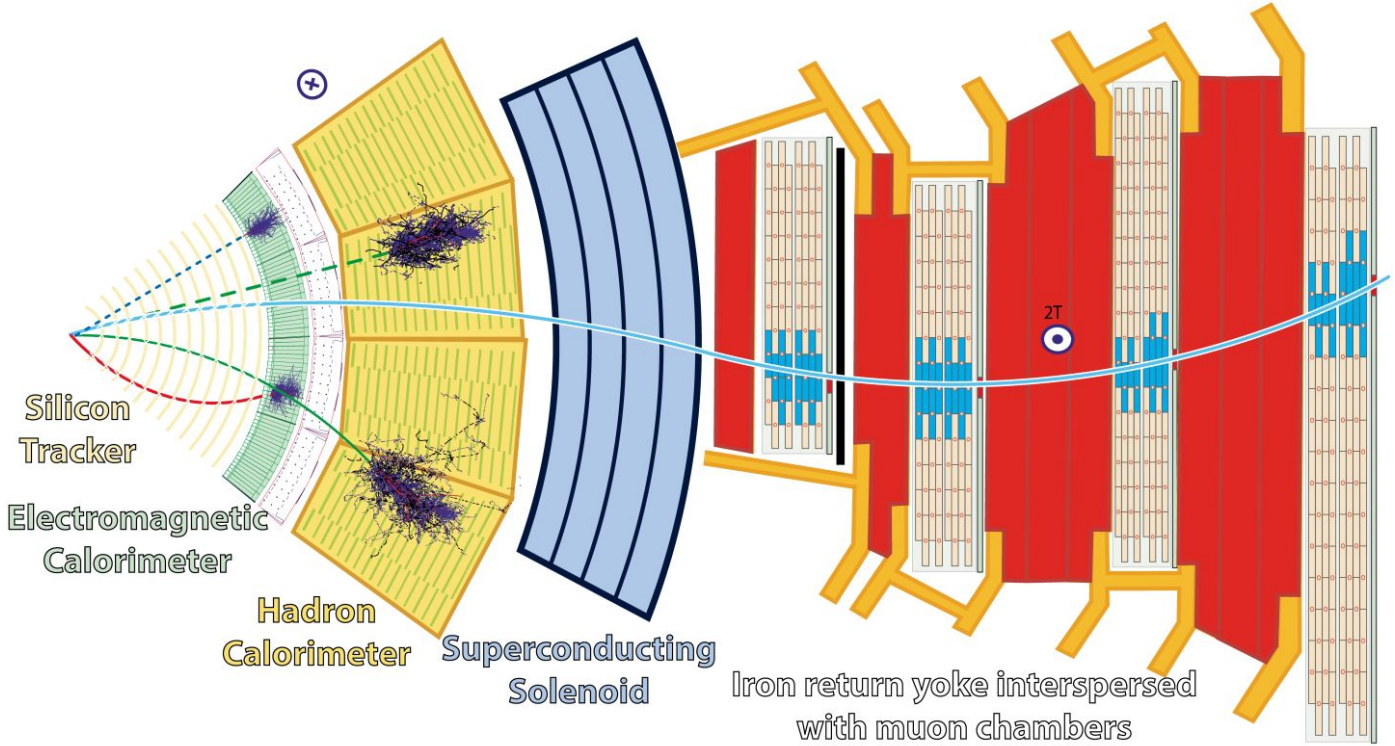
2021 MCnet-Beijing Summer School

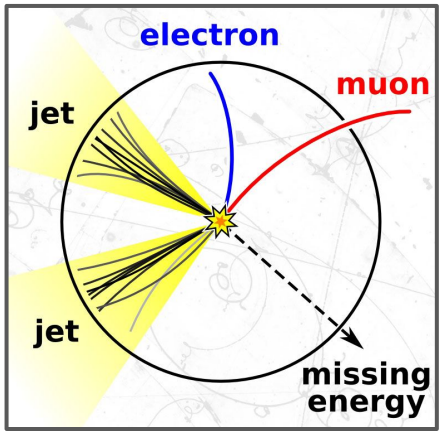
on Monte Carlo Event Generators for High Energy Physics

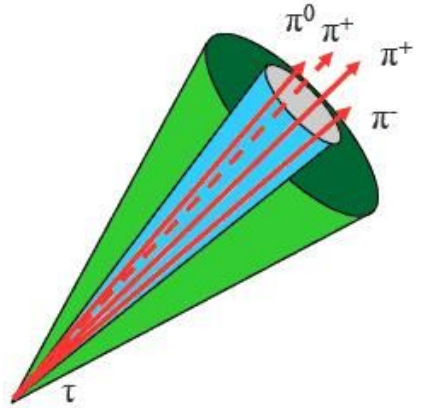
High Energy Collisions



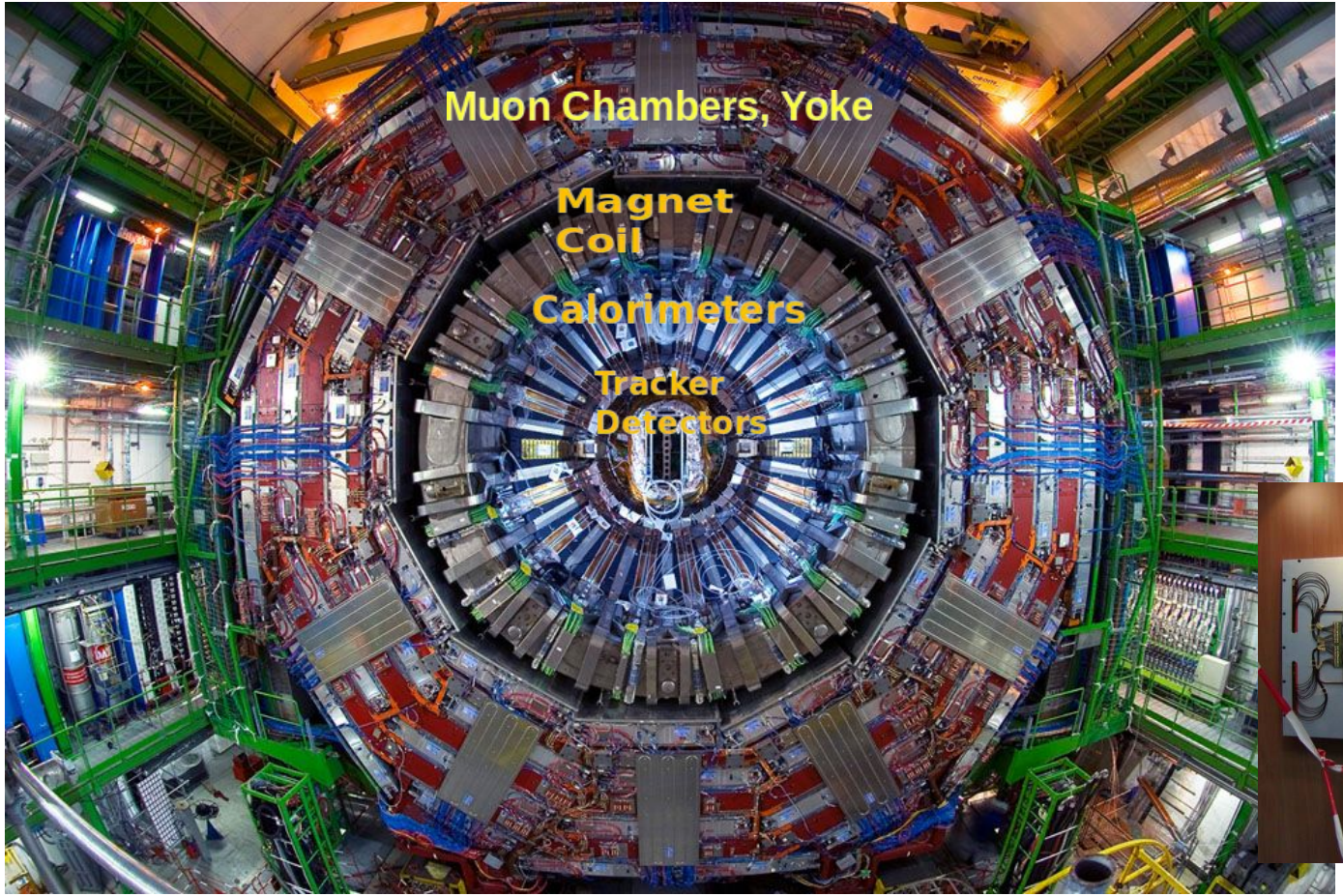
Detectors ⇒ Information





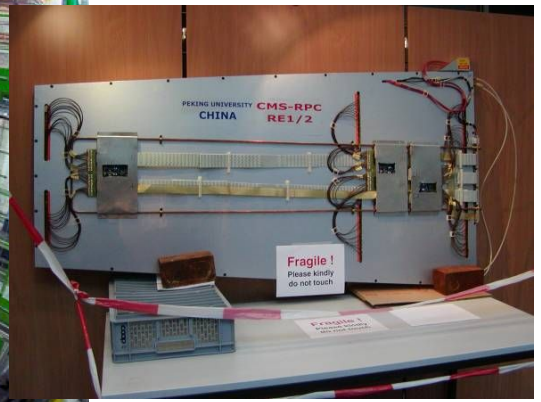


CMS Detectors

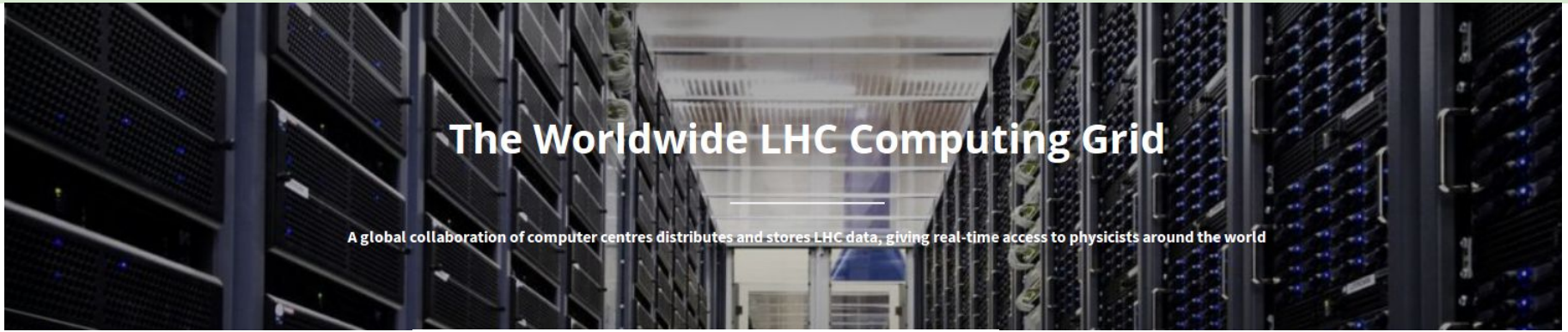


weight: 12500 t
overall diameter: 15 m
overall length: 21.6 m

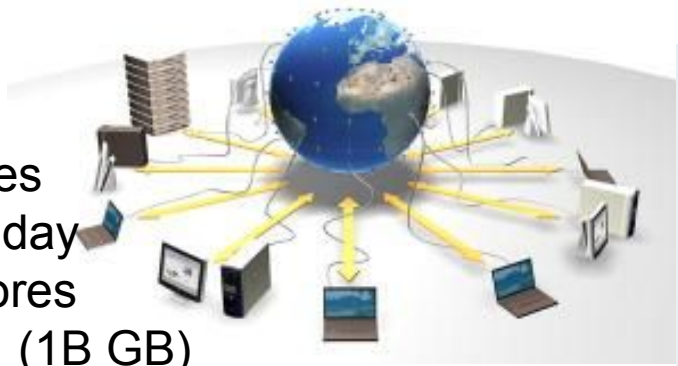
Camera?
Recorder?

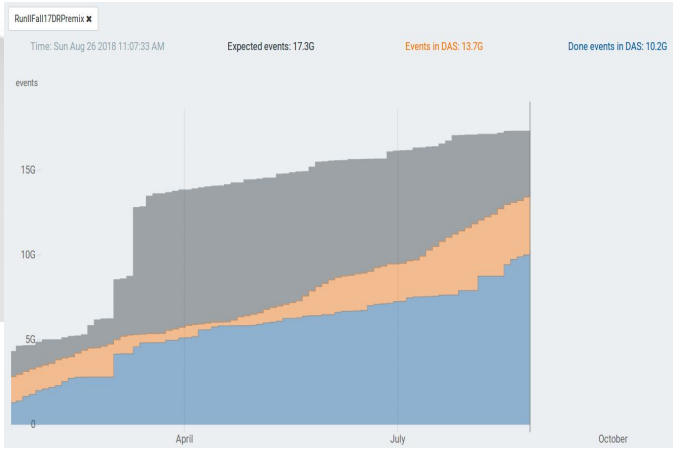


Big data Science



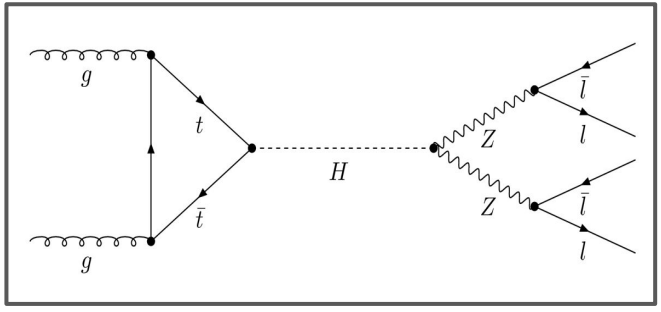
42 countries
170 computing centres
Over 2 million tasks/ day
1 million computer cores
1 exabyte of storage (1B GB)

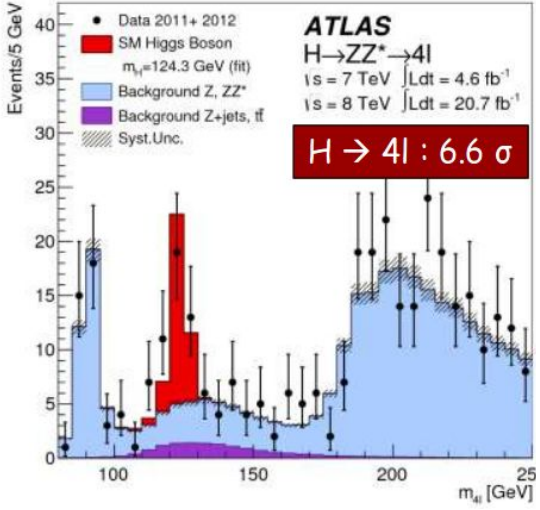


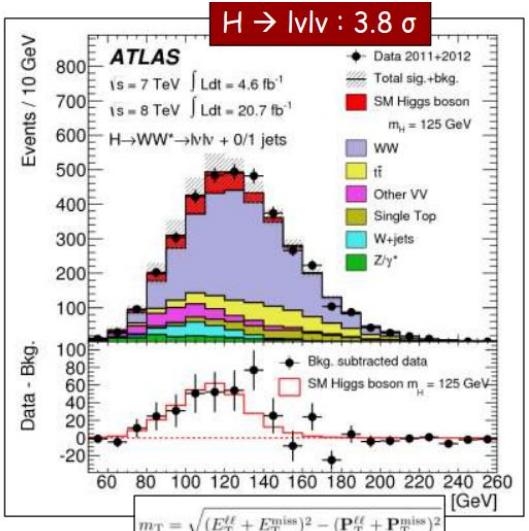


CMS： 15B events in 8 months

Data Mining







Needles in a haystack

In ATLAS, up to July 4, 2012:

A million billion collisions

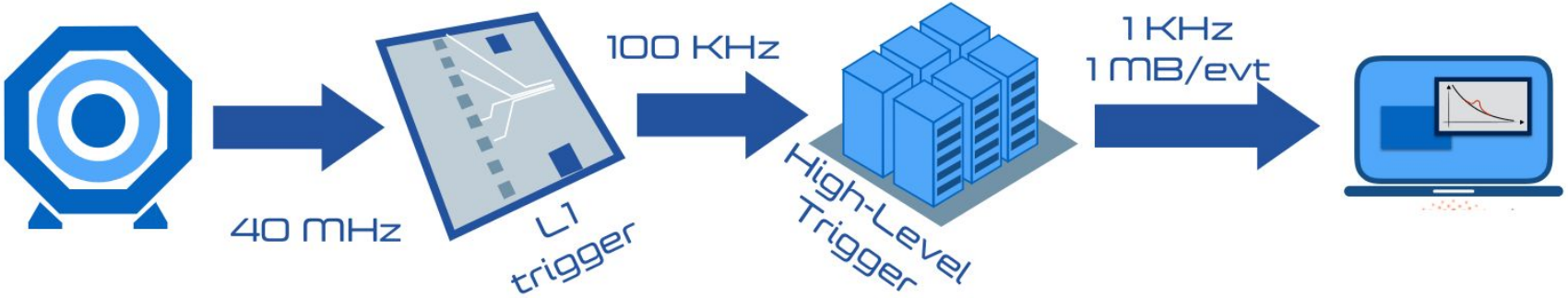
4.2 billion events analyzed

240,000 Higgs particles produced

~350 diphoton Higgs events detected

~8 four-lepton Higgs events detected

LHC Data Flow

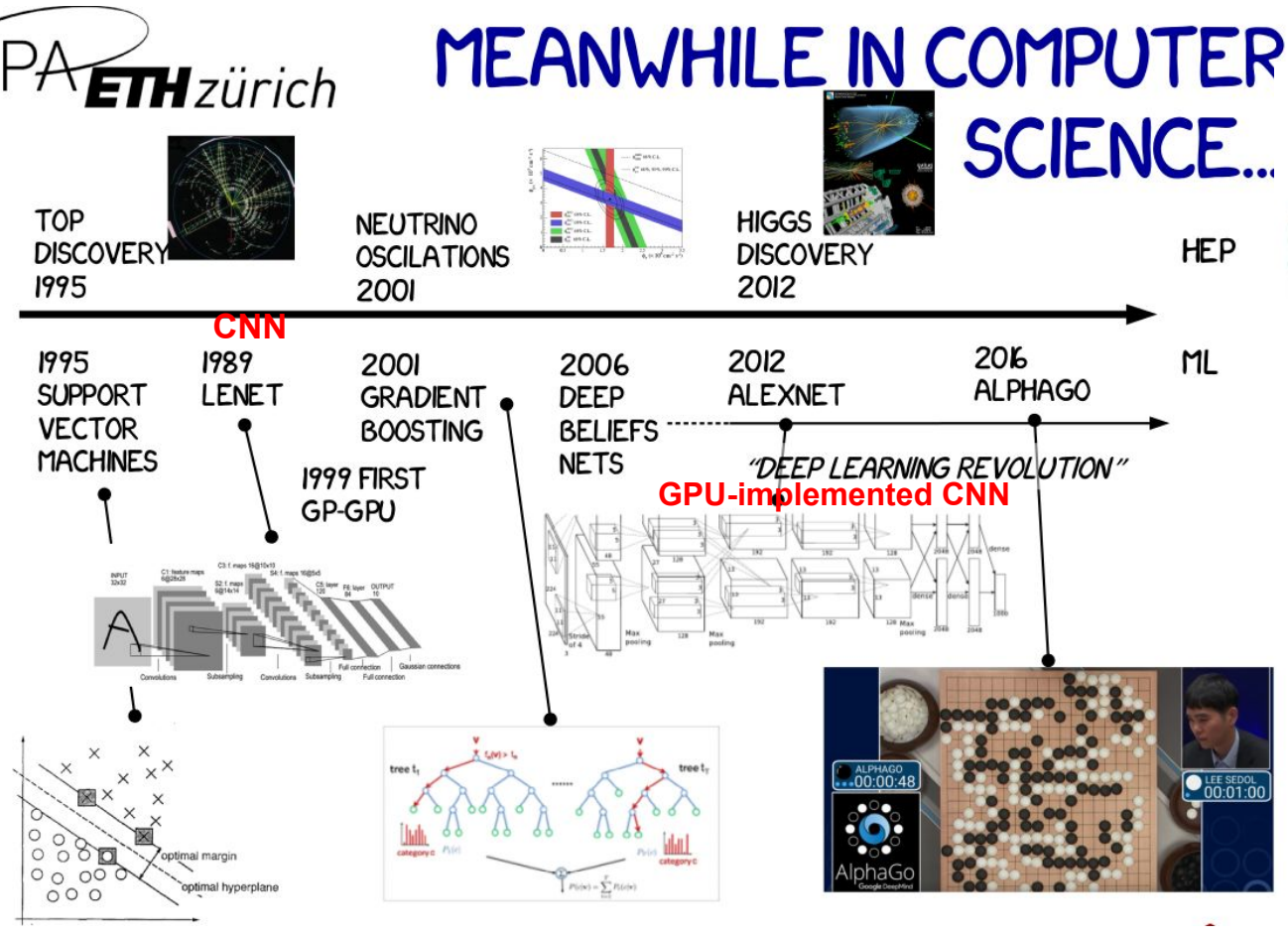


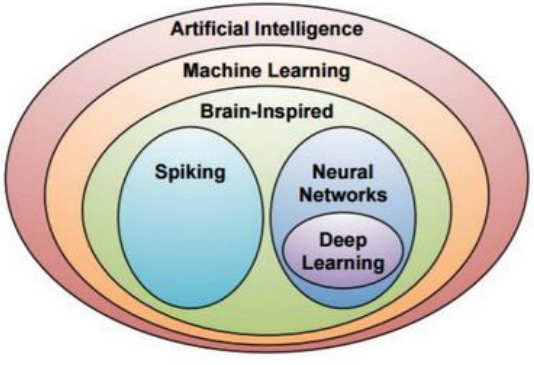
- **L1 trigger:** local, hardware based, on FPGA, @experiment site
- **HLT:** local/global, software based, on C/GPU, @experiment site
- **Offline:** global, software based, on C/GPU, @CERN T0
- **Analysis:** user-specific applications running on the grid

ML for : Particle ID; Signal Mining; Inference accelerator ; Automatic anomaly detection...

Brief history of ML

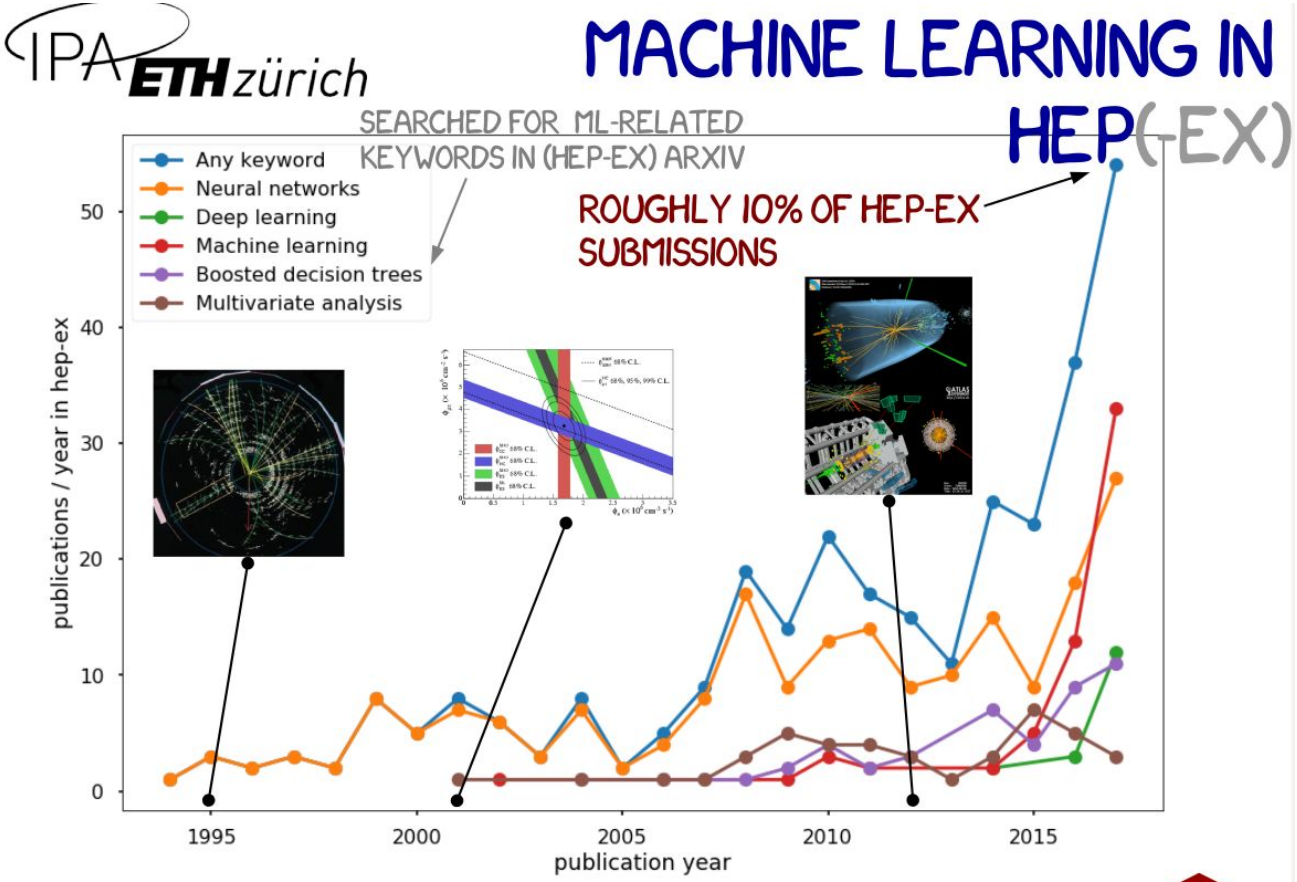
IPA ETH zürich

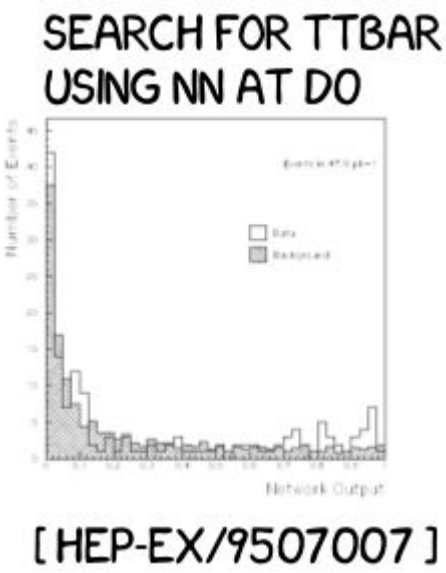




Turing Award 2018

ML in HEP





Tevatron: Top quark
LHC: Higgs discovery
miniBOONE: PID

ML in HEP

Using neural networks to identify jets

#151

Leif Lonnblad (Lund U.), Carsten Peterson (Lund U.), Thorsteinn Rognvaldsson (Lund U.) (May, 1990)

Published in: *Nucl.Phys.B* 349 (1991) 675-702

DOI cite 120 citations

High Energy Physics - Phenomenology

[Submitted on 20 Jan 2011 (v1), last revised 20 Jun 2011 (this version, v2)]

Searches for the t' of a fourth family

Bob Holdom, Qi-Shu Yan

We study the detection of the t' of a fourth family during the early running of LHC with 7 TeV collision energy and 1 fb^{-1} integrated luminosity. By use of a neural network we show that it is feasible to search for the t' even with a mass close to the unitarity upper bound, which is in the 500 to 600 GeV range. We also present results for the Tevatron with $10\ \text{fb}^{-1}$. In both cases the search for a fourth family quark doublet can be significantly enhanced if one incorporates the contribution that the b' can make to a t'-like signal. Thus the bound on the mass of a degenerate quark doublet should be stronger than the bounds obtained by treating t' and b' in isolation.

HEP and ML

Peter Higgs

CH FRS FRSE FInstP

Nobel laureate Peter Higgs at a press conference, Stockholm, December 2013

Born	Peter Ware Higgs 29 May 1929 (age 90) Newcastle upon Tyne, England, UK
Residence	Edinburgh, Scotland, UK
Nationality	British[1]
Alma mater	King's College London (BSc, MSc, PhD)
Known for	Higgs boson Higgs field Higgs mechanism Symmetry breaking
Institutions	University of Edinburgh Imperial College London University College London King's College London
Thesis	*Some problems in the theory of molecular vibrations* (1955)
Doctoral advisor	Charles Coulson[2][3] Christopher Longuet-Higgins[2][4]

Charles Alfred Coulson: 应用数学家，化学家

Christopher Longuet-Higgins: 理论化学家，40岁（1970s），改行做人工智能

Doctoral advisor	Christopher Longuet-Higgins[3][4][5]
Doctoral students	Richard Zemel[6] Brendan Frey[7] Radford M. Neal[8] Ruslan Salakhutdinov[9] Ilya Sutskever[10]
Other notable students	Yann LeCun (postdoc) Peter Dayan (postdoc) Zoubin Ghahramani (postdoc)

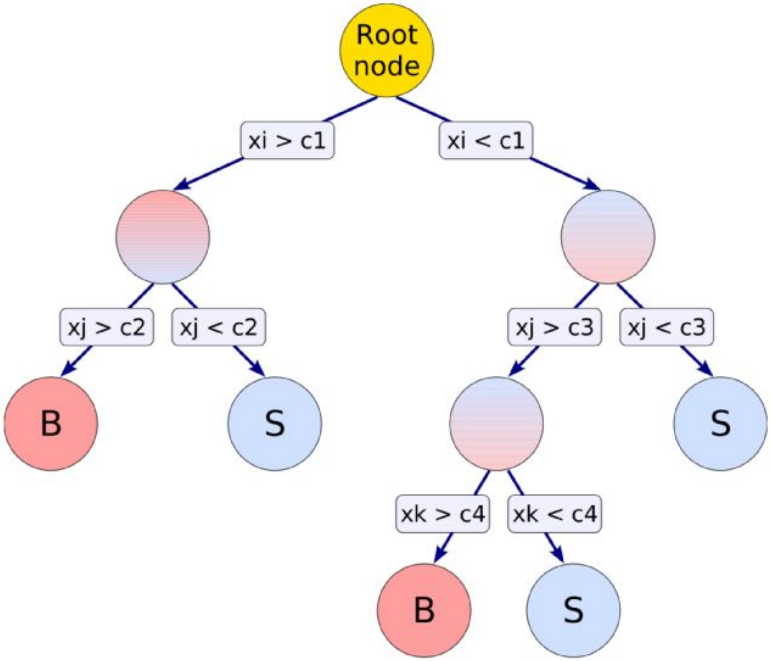
Geoffrey Hinton

FRS FRSC CC

Hinton in 2013

Born	Geoffrey Everest Hinton 6 December 1947 (age 71)[1] Wimbledon, London
Residence	Canada
Alma mater	University of Cambridge (BA) University of Edinburgh (PhD)

BDT introduction



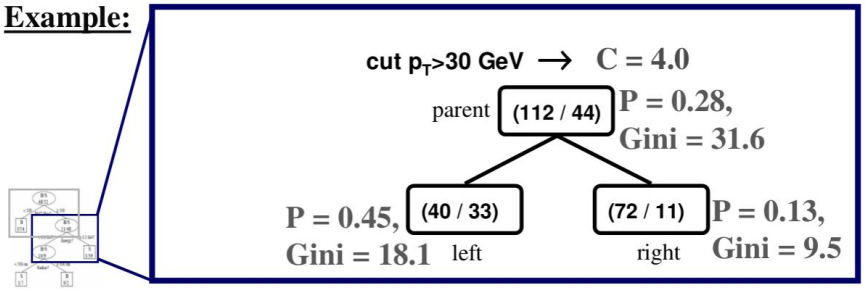
Tree
Node
Leaf

Gini: Note that Gini is 0 for all signal or all background. W_i is the weight of event "i".

$$G_{ini} = \left(\sum_{i=1}^{n} W_i \right) P(1-P)$$

- Pick the branch to maximize the change in gini.
- Criterion C = $\text{Gini}_{\text{parent}} - \text{Gini}_{\text{right-child}} - \text{Gini}_{\text{left-child}}$

Example:



- Optimize each node (e.g. p_T>30 GeV) by maximizing "C".

BDT introduction

- Easy to understand/interpret; Training Fast
- Single tree are not stable
  - a small change/fluctuation in the data can make a large difference!
- Solution: e.g. Boosting! → Boosted Decision Trees
  - Each tree is created iteratively
  - The tree's output ($h(x)$) is given a weight (w) relative to its accuracy
  - The ensemble output is the weighted sum:

$$\hat{y}(x) = \sum_t w_t h_t(x)$$

  - After each iteration each data sample is given a weight based on its misclassification
    - The more often a data sample is misclassified, the more important it becomes
  - The goal is to minimize an objective function

$$O(x) = \sum_i l(\hat{y}_i, y_i) + \sum_t \Omega(f_t)$$

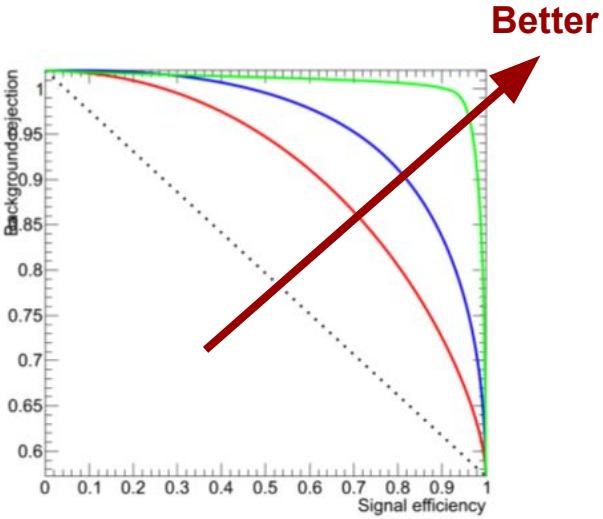
    - $l(\hat{y}_i, y_i)$ is the loss function --- the distance between the truth and the prediction of the ith sample
    - $\Omega(f_t)$ is the regularization function --- it penalizes the complexity of the tth tree

https://en.wikipedia.org/wiki/Boosting_(machine_learning)

BDT Introduction

- **AdaBoost "Adaptive Boosting"**
  - One of the originals
  - Freund and Schapire
- **Gradient Boosting**
  - Uses gradient descent to create new learners
  - The loss function is differentiable
  - Friedman: https://statweb.stanford.edu/~jhf/ftp/trebst.pdf
- **XGBoost "eXtreme Gradient Boosting"**
  - Type of gradient boosting
  - Has become very popular in data science competitions
  - Chen and Guestrin: https://arxiv.org/abs/1603.02754

ROC (Receiver Operating Characteristic) Curves

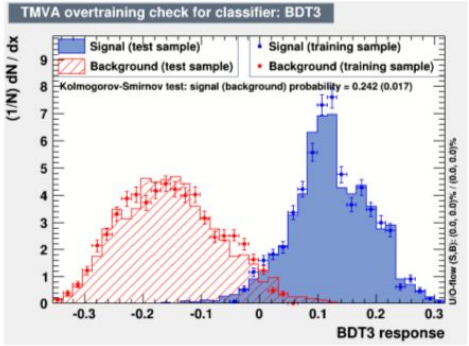


Best classifier can be identified by the largest **AUC** (Area under curve)

Overtraining check:

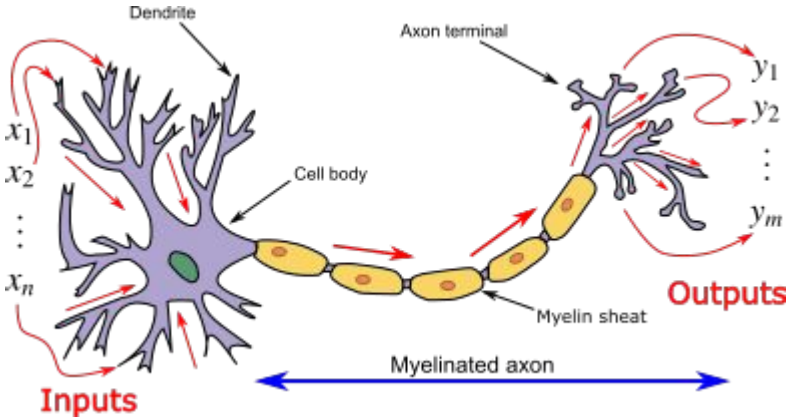
- Split data in **training / test**
- Performance on the training samples should not be better than on the test sample

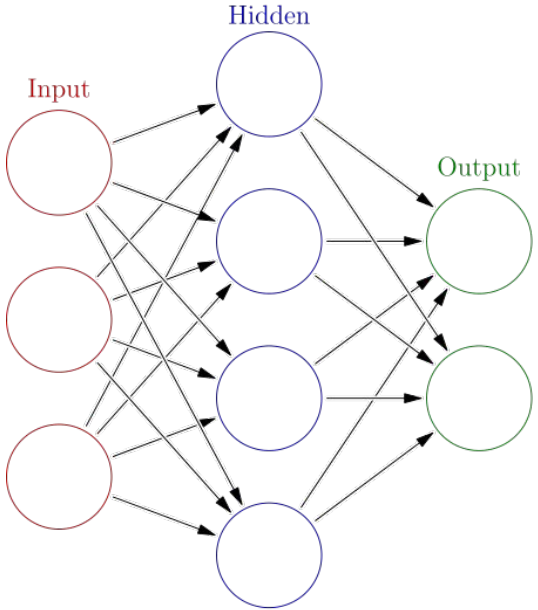
2500 trees



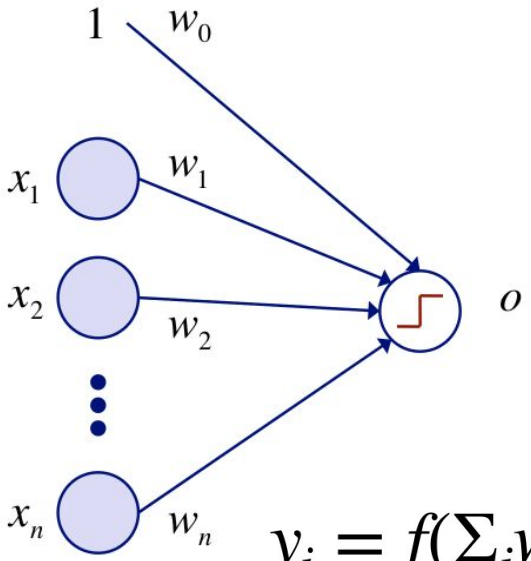
NN Introduction

- Artificial Neural Networks, connectionist models
- inspired by interconnected neurons in biological systems
  - First Mathematical model of neurons Pitts & McCulloch (1943)
  - 1986 **Backpropagation** reinvented: Rumelhart, Hinton et al.Nature
  - **1990s: Great success of SVM and graphical models almost kills the ANN**
  - **Yann LeCun (1998) developed deep convolutional neural networks**
    - LeNet-5, a pioneering 7-level convolutional network
  - 2006+: Deep learning is a rebranding of ANN research.
    - Convolutional neural networks running on GPUs





NN Introduction



$$y_i = f(\Sigma_j w_{ij} x_j + b_i)$$

given: network structure and a training set $D = \{(\boldsymbol{x}^{(1)}, y^{(1)}) \ldots (\boldsymbol{x}^{(m)}, y^{(m)})\}$

initialize all weights in $\boldsymbol{w}$ to small random numbers

until stopping criteria met do

for each $(\boldsymbol{x}^{(d)}, y^{(d)})$ in the training set

input $\boldsymbol{x}^{(d)}$ to the network and compute output $o^{(d)}$

calculate the error $E(\boldsymbol{w}) = \frac{1}{2}\left(y^{(d)} - o^{(d)}\right)^2$

calculate the gradient

$$\nabla E(\boldsymbol{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots, \frac{\partial E}{\partial w_n}\right]$$

update the weights

$$\Delta \boldsymbol{w} = -\eta \nabla E(\boldsymbol{w})$$

b: bias term

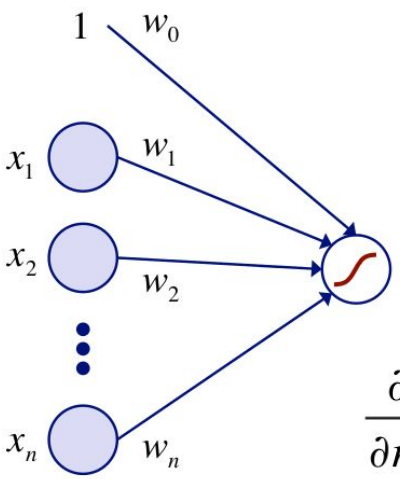
η: learning rate

Standard gradient descent (batch training)

Stochastic gradient descent (online training)

NN Introduction

如果不用激活函数，每一层输出都是上层输入的线性函数，无论神经网络有多少层，输出都是输入的线性组合，这种情况就是最原始的感知机



$$net^{(d)} = w_0 + \sum_{i=1}^{n} w_i x_i^{(d)}$$

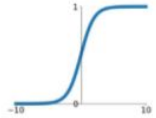
$$o^{(d)} = \frac{1}{1 + e^{-net^{(d)}}}$$

Sigmoid Function

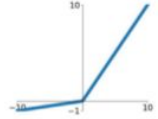
$$\frac{\partial o^{(d)}}{\partial net^{(d)}} = o^{(d)}(1 - o^{(d)})$$

Activation Functions

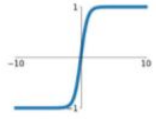
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



Leaky ReLU
$\max(0.1x, x)$

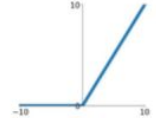


tanh
$\tanh(x)$

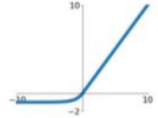


Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ReLU
$\max(0, x)$



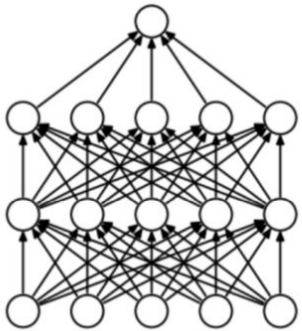
ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

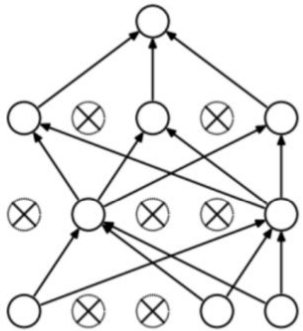


线性整流函数

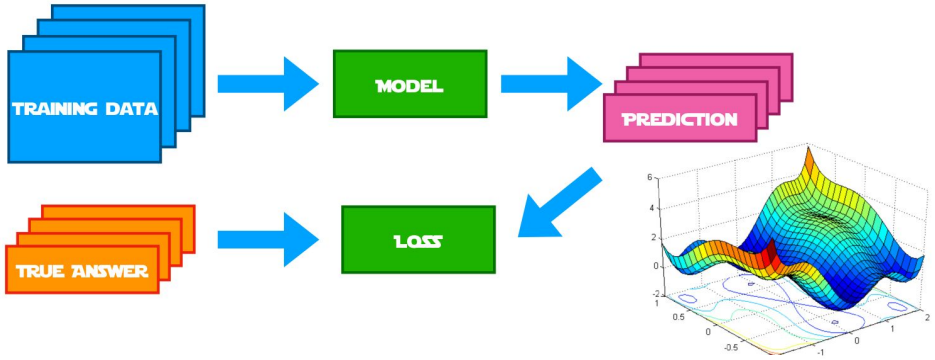
$$\frac{\partial E^{(d)}}{\partial w_i} = \frac{\partial}{\partial w_i}\frac{1}{2}\left(y^{(d)} - o^{(d)}\right)^2 = -\left(y^{(d)} - o^{(d)}\right)o^{(d)}(1 - o^{(d)})x_i^{(d)}$$

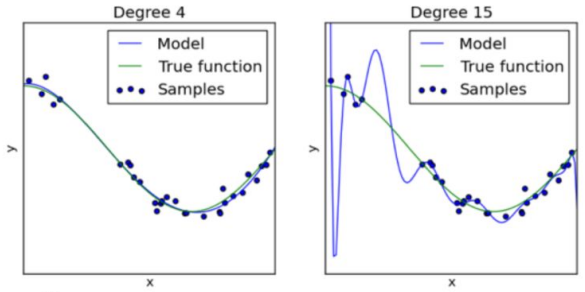
Dropout layer:
Randomly drop links between neurons, with probability p

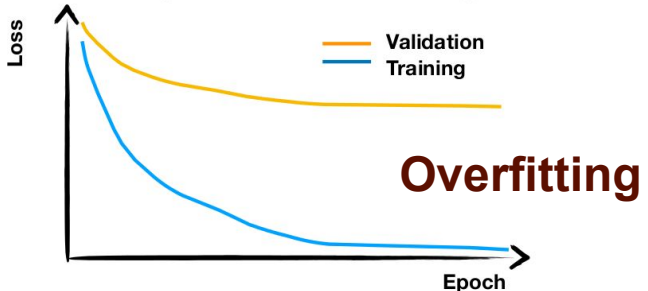

(a) Standard Neural Net


(b) After applying dropout.

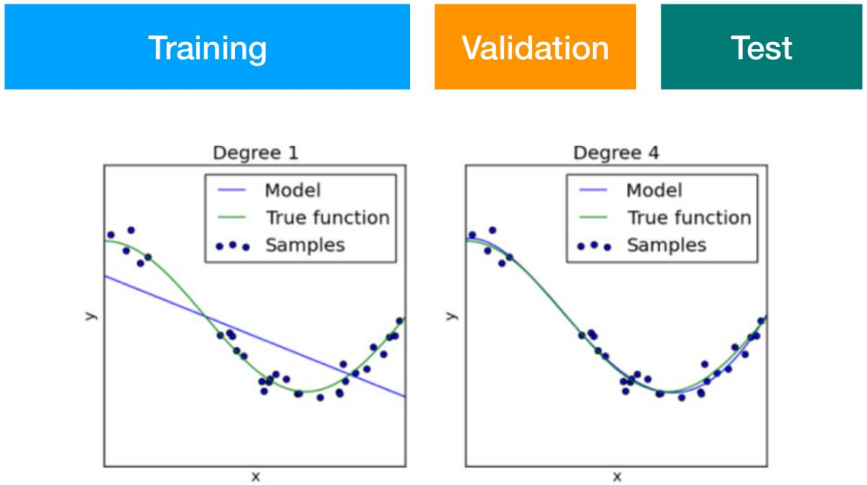
Supervised Learning



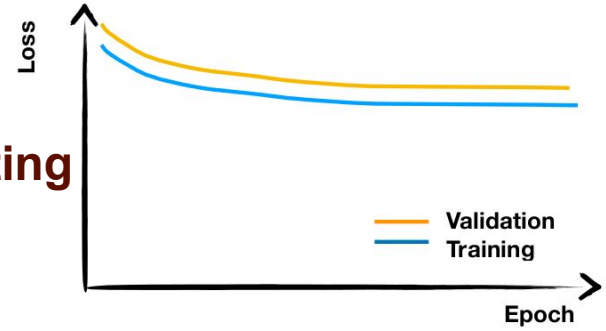




Overfitting



underfitting



A living Review of ML for Particle Physics

A Living Review of Machine Learning for Particle Physics

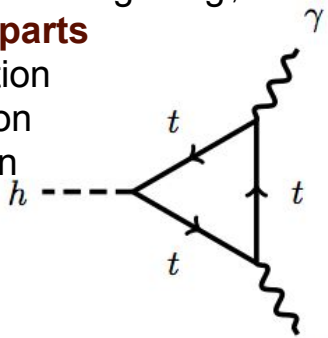
Modern machine learning techniques, including deep learning, is rapidly being applied, adapted, and developed for high energy physics. The goal of this document is to provide a nearly comprehensive list of citations for those developing and applying these approaches to experimental, phenomenological, or theoretical analyses. As a living document, it will be updated as often as possible to incorporate the latest developments. A list of proper (unchanging) reviews can be found within. Papers are grouped into a small set of topics to be as useful as possible. Suggestions are most welcome.

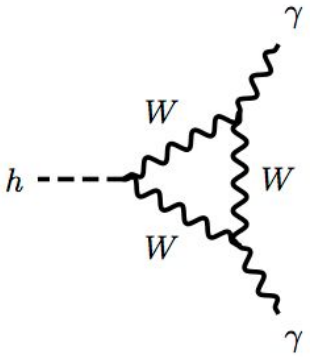
- Reviews
  - Modern reviews
    - Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in T
    - Deep Learning and its Application to LHC Physics [DOI]
    - Machine Learning in High Energy Physics Community White Paper [DOI]
    - Machine learning at the energy and intensity frontiers of particle physics
    - Machine learning and the physical sciences [DOI]
    - Machine and Deep Learning Applications in Particle Physics [DOI]
    - Modern Machine Learning and Particle Physics
  - Specialized reviews
    - The Machine Learning Landscape of Top Taggers [DOI]
    - Dealing with Nuisance Parameters using Machine Learning in High Energy Physic
    - Graph neural networks in particle physics [DOI]
    - A Review on Machine Learning for Neutrino Experiments [DOI]
    - Generative Networks for LHC events
    - Parton distribution functions
    - Simulation-based inference methods for particle physics
- Direct Dark Matter Detectors
  - Boosted decision trees approach to neck alpha events discrimination in DEAP-3600 experime
  - Improving sensitivity to low-mass dark matter in LUX using a novel electrode background mitig
  - Convolutional Neural Networks for Direct Detection of Dark Matter [DOI]
- Cosmology, Astro Particle, and Cosmic Ray physics
  - Detecting Subhalos in Strong Gravitational Lens Images with Image Segmentation
  - Mining for Dark Matter Substructure: Inferring subhalo population properties from strong lense [DOI]
  - Inverting cosmic ray propagation by Convolutional Neural Networks
  - Particle Track Reconstruction using Geometric Deep Learning
  - Deep-Learning based Reconstruction of the Shower Maximum X_{max} using the Detectors of the Pierre Auger Observatory
  - A comparison of optimisation algorithms for high-dimensional particle and astrophysics applic
  - Tackling the muon identification in water Cherenkov detectors problem for the future Southern Observatory by means of Machine Learning
  - Muon identification in a compact single-layered water Cherenkov detector and gamma/hadron Machine Learning techniques
  - A convolutional-neural-network estimator of CMB constraints on dark matter energy injection
  - A neural network classifier for electron identification on the DAMPE experiment
  - Bayesian nonparametric inference of neutron star equation of state via neural network
  - Novel null tests for the spatial curvature and homogeneity of the Universe and their machine l
  - Machine Learning the 6th Dimension: Stellar Radial Velocities from 5D Phase-Space Correlations
  - Via Machinae: Searching for Stellar Streams using Unsupervised Machine Learning
- Regression
  - Pileup
    - Pileup Mitigation with Machine Learning (PUMML) [DOI]
    - Convolutional Neural Networks with Event Images for Pileup Mitigation with the ATLAS Detector
    - Pileup mitigation at the Large Hadron Collider with graph neural networks [DOI]
    - Jet grooming through reinforcement learning [DOI]
  - Calibration
    - Parametrizing the Detector Response with Neural Networks [DOI]
    - Simultaneous Jet Energy and Mass Calibrations with Neural Networks
    - Generalized Numerical Inversion: A Neural Network Approach to Jet Calibration
    - Calorimetry with Deep Learning: Particle Classification, Energy Regression, and Simulation for High-Energy Physics
    - Per-Object Systematics using Deep-Learned Calibration [DOI]
    - A deep neural network for simultaneous estimation of b jet energy and resolution [DOI]
    - How to GAN Higher Jet Resolution
    - Deep learning jet modifications in heavy-ion collisions
  - Recasting
    - The BSM-AI project: SUSY-AI–generalizing LHC limits on supersymmetry with machine learning
    - Accelerating the BSM interpretation of LHC data with machine learning [DOI]
    - Bayesian Neural Networks for Fast SUSY Predictions [DOI]
  - Matrix elements
    - Using neural networks for efficient evaluation of high multiplicity scattering amplitudes [DOI]
    - (Machine) Learning Amplitudes for Faster Event Generation
    - $\textsf{Xsec}$: the cross-section evaluation code [DOI]
    - Matrix Element Regression with Deep Neural Networks – breaking the CPU barrier
    - Unveiling the pole structure of S-matrix using deep learning
    - Model independent analysis of coupled-channel scattering: a deep learning approach
  - Parameter estimation

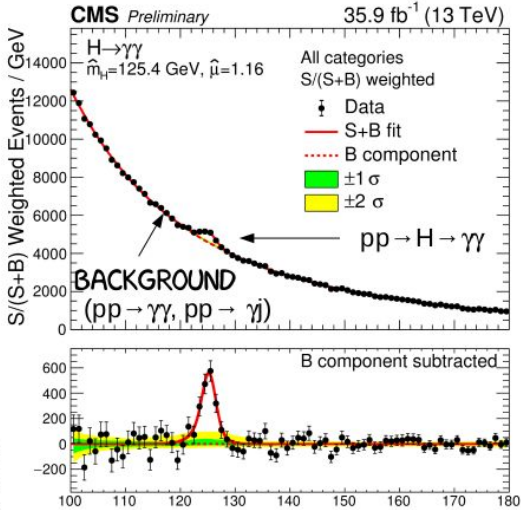
Application example 1: Higgs discovery

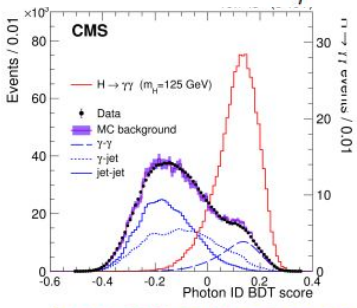
Higgs to diphoton in CMS:

- BR~10^{-3}: small signal over huge bkg;
- **BDT applied in many parts**
  - Photon identification
  - Event classification
  - Energy regression
  - Diphoton vertex



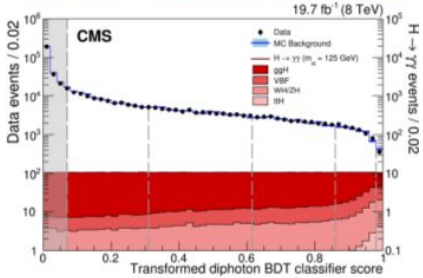




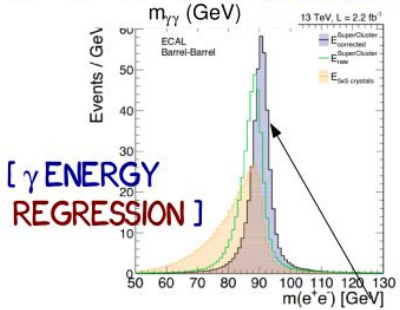


[PARTICLE-ID: SEPARATE PROMPT γ FROM HADRONIC JETS]

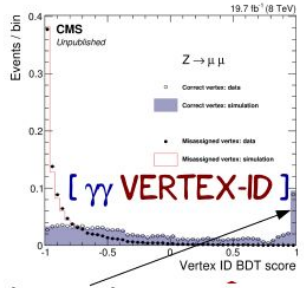
[EVENT CLASSIFICATION]



[INVARIANT MASS ESTIMATION]



[γ ENERGY REGRESSION]



[γγ VERTEX-ID]

$m^2 = E_1 E_2 (1-\cos\alpha)$

Application example 2 : NNPDF

ANNs provide **universal unbiased interpolants** to parametrize the non-perturbative dynamics that determines the **size and shape of the PDFs** from experimental data

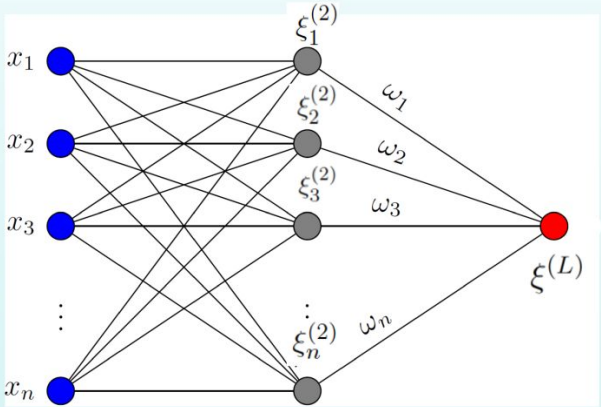
not from QCD!

Traditional approach

$$g(x,Q_0) = A_g(1-x)^{a_g}x^{-b_g}\left(1 + c_g\sqrt{s} + d_g x + \ldots\right)$$

NNPDF approach

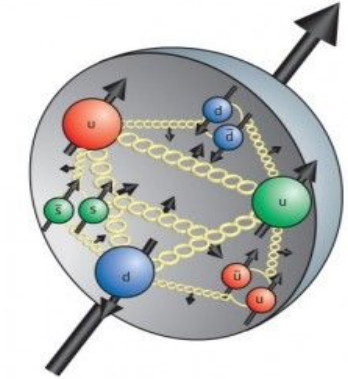
$$g(x,Q_0) = A_g \mathrm{ANN}_g(x)$$

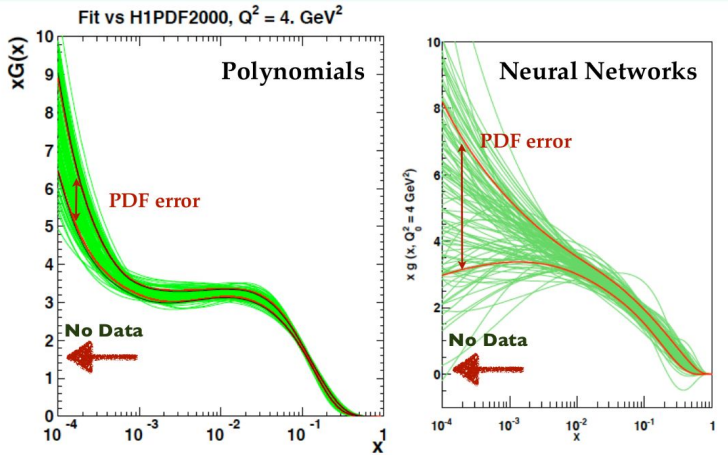


$$\mathrm{ANN}_g(x) = \xi^{(L)} = \mathcal{F}\left[\xi^{(1)}, \{\omega_{ij}^{(l)}\}, \{\theta_i^{(l)}\}\right]$$

$$\xi_i^{(l)} = g\left(\sum_{j=1}^{n_{l-1}} \omega_{ij}^{(l-1)}\xi_j^{(l-1)} - \theta_i^{(l)}\right)$$

- ANNs eliminate **theory bias** introduced in PDF fit from choice of *ad-hoc* functional forms
- NNPDF fits used **O(400) free parameters**, to b compared with O(10-20) in traditional PDFs. Result stable if **O(4000) parameters used!**

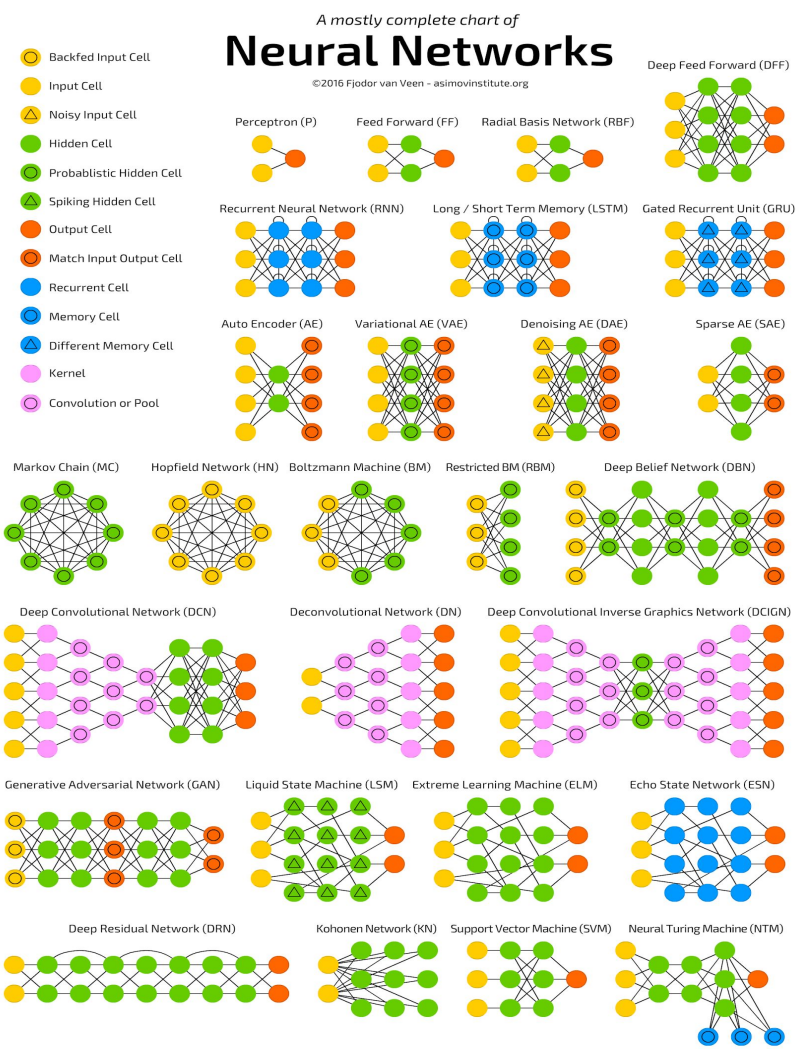




ANNs avoid biasing the PDFs, faithful extrapolation at small-x (very few data, thus error blow up)

Deep NN

- Deep neural networks are those with >1 inner layer
- Thanks to GPUs, it is now possible to train them efficiently, which boosted the revival of neural networks in the 2000s
- In addition, **new architectures** emerged, which better exploit the new computing power



Universal approximation theorem:

The standard multilayer feed-forward networks with a single hidden layer that contains finite number of hidden neurons, and with arbitrary **activation** function are universal approximators in C(R^m).

DNN: frameworks

ROOT

- Data analysis framework for HEP, developed mainly at CERN
- Written in C++ (fully interpreted)

TMVA

- **T**oolkit for **M**ulti**v**ariate **A**nalysis
- Includes several machine learning algorithms such as :
  - *Likelihood, KNN, Fisher, MLP, SVN, Neural Networks, BDT, etc...*

TensorFlow

PyTorch

mxnet

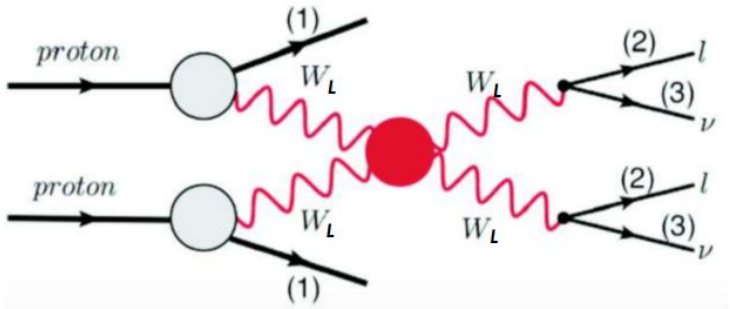
NumPy

scikit learn

Keras

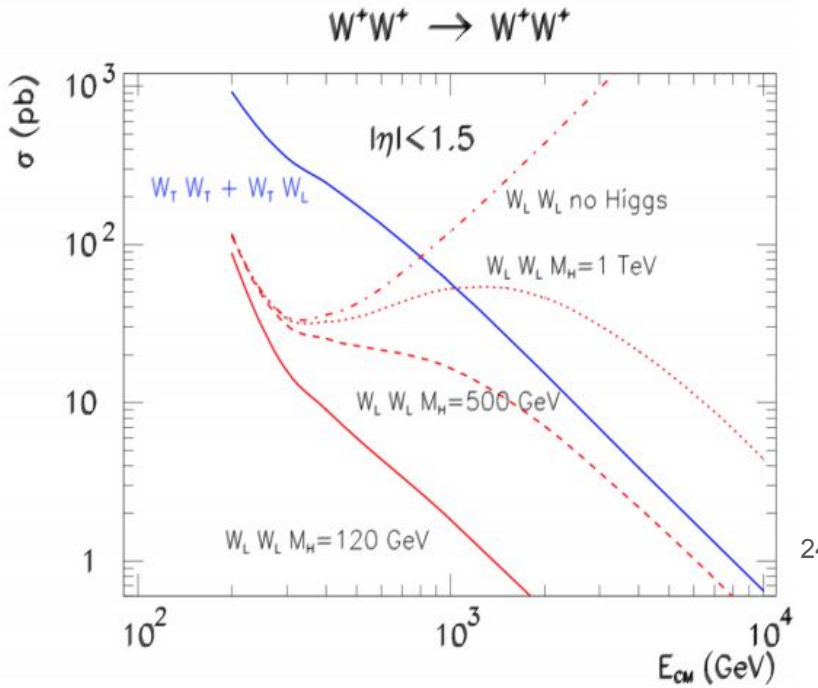
Application example 3 : Vector Boson Scattering

Same-Sign W W-> Polarized scattering, ~200 events (2016-2018)!



Boson scattering and Interaction

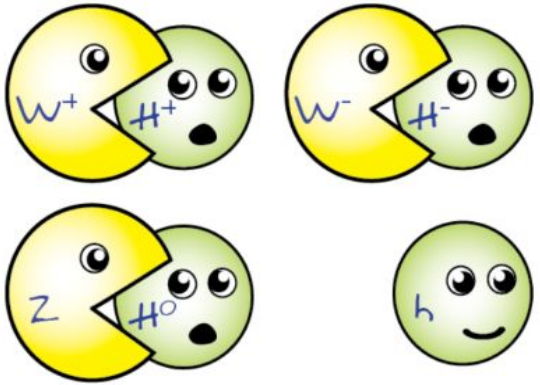
- Yang-Mills Non-Abelian interactions
  Anomalous coupling, EFT
- Electroweak symmetry breaking
  Higgs Unitarization Scheme
- Tev scale new Physics
  Boosted Boson

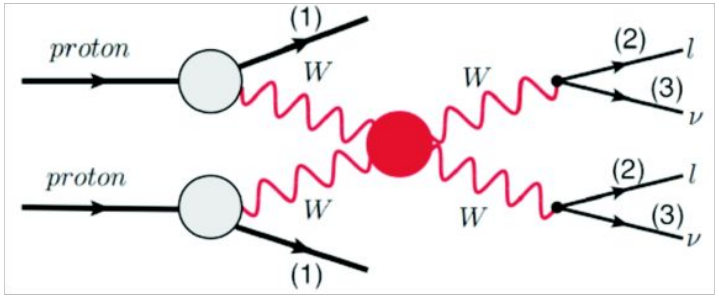


WW->WW behavior on scattering energy

24

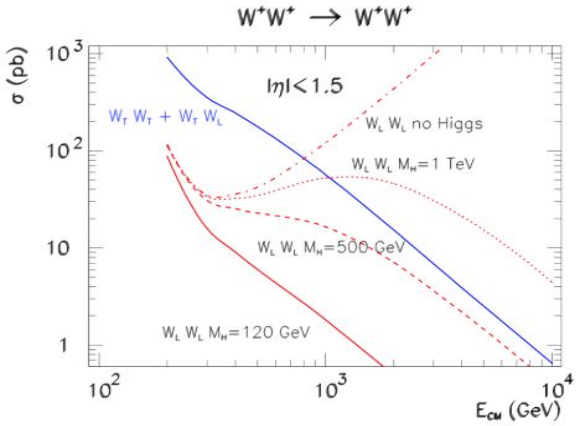
Application example 3 : Vector Boson Scattering

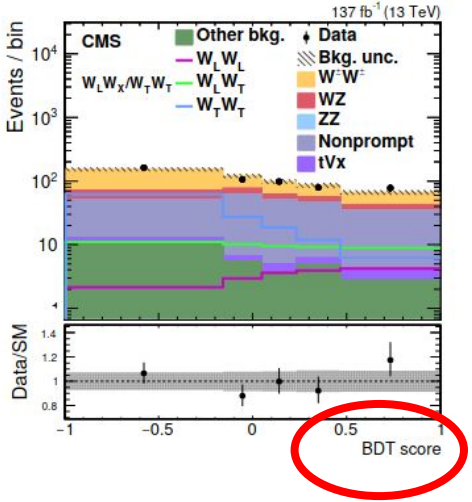




Phys. Lett. B 812 (2020) 136018

First probe on Longitudinal polarized VBS from CMS in 2020 with BDT. It is the beginning of the game, though sensitivity ~ 1 standard deviation.



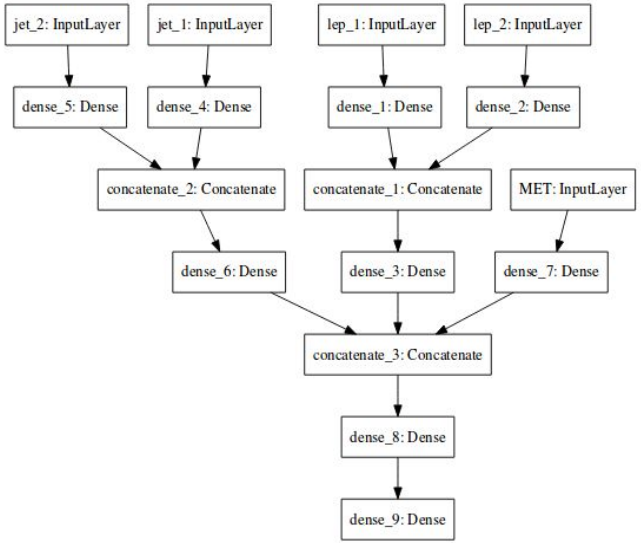


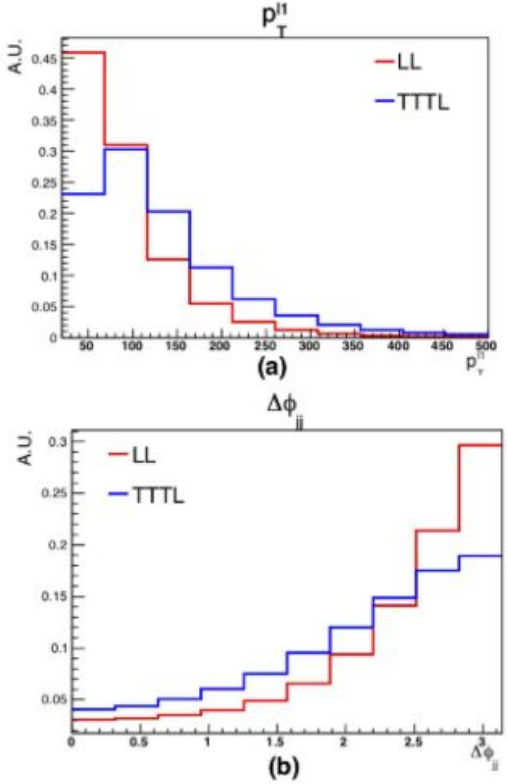
"Longitudinal weak boson scattering… is one of the most important processes to be studied at the Superconducting Super Collider and the CERN Large Hadron Collider."

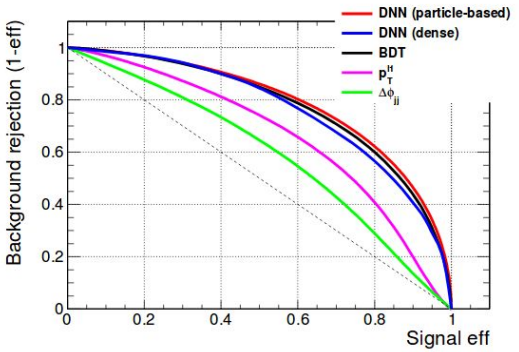
Can verify Higgs unitarization scheme directly!

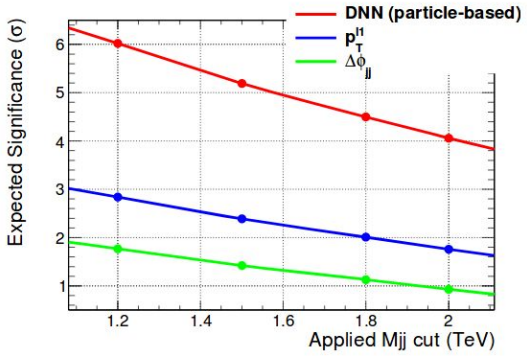
Process	$\sigma\mathcal{B}$ (fb)	Theoretical prediction (fb)
$W_L^{\pm}W_L^{\pm}$	$0.32^{+0.42}_{-0.40}$	0.44 ± 0.05
$W_X^{\pm}W_T^{\pm}$	$3.06^{+0.51}_{-0.48}$	3.13 ± 0.35
$W_L^{\pm}W_X^{\pm}$	$1.20^{+0.56}_{-0.53}$	1.63 ± 0.18
$W_T^{\pm}W_T^{\pm}$	$2.11^{+0.49}_{-0.47}$	1.94 ± 0.21

Application example 3 : Vector Boson Scattering









Classification

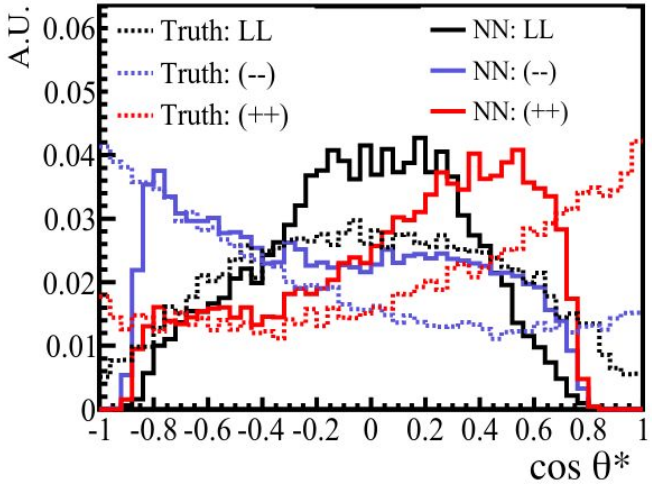
PKU group tried DNN using a specific particle based input structure, which shows improvement over BDT. PRD 99, 033004 (2019), PRD 100, 116010 (2019)

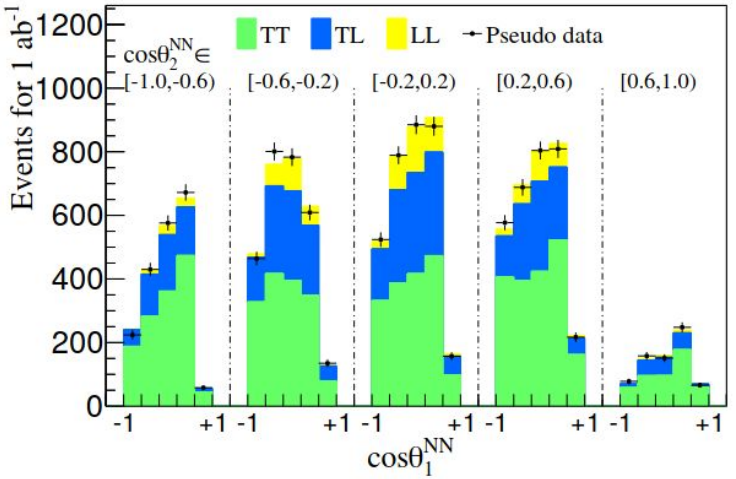
Application example 3 : Vector Boson Scattering

$$\frac{d\sigma}{d\cos\theta^*} \propto \frac{3}{8} f_-(1 \mp \cos\theta^*)^2 + \frac{3}{8} f_+(1 \pm \cos\theta^*)^2 + \frac{3}{4} f_L(1 - \cos^2\theta^*), \text{for } W^\pm$$

$$\mathcal{C} = \frac{1}{N}\sum_{i=1}^{N}[(\cos\theta^*_{1,i} - \cos\theta^{NN}_{1,i})^2 + (\cos\theta^*_{2,i} - \cos\theta^{NN}_{2,i})^2],$$

Two neutrinos: using DNN regression to get the angular distributions

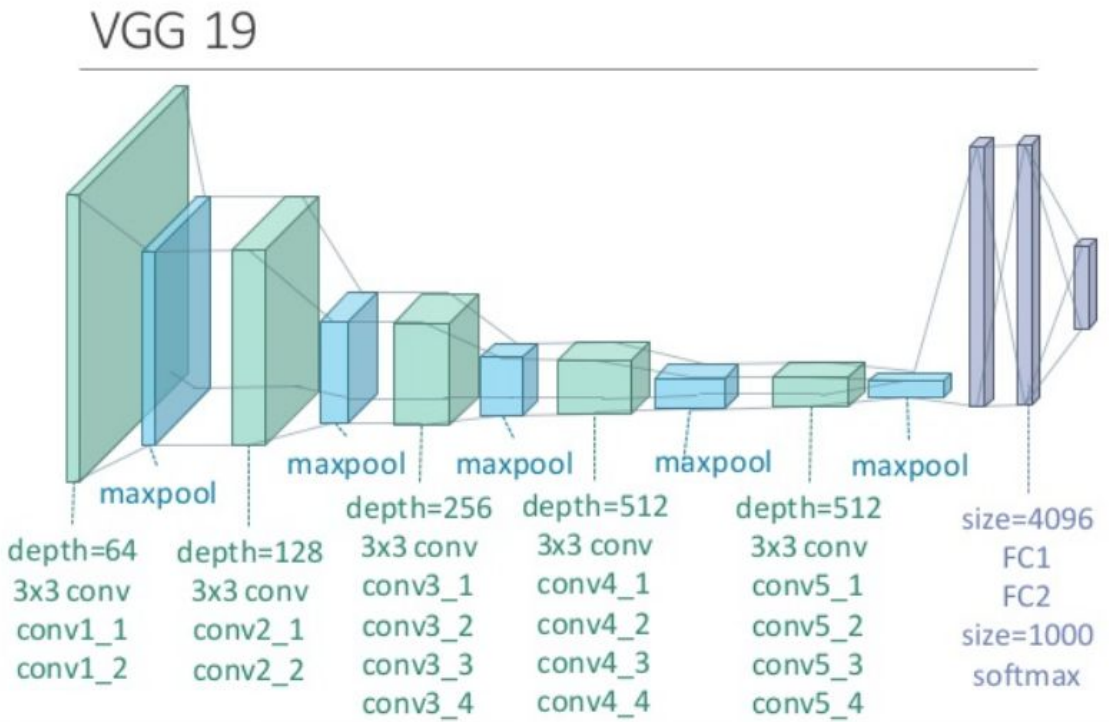




DNN regression shows also promising results to be tried at the LHC measurements. PRD 93, 094033 (2016)

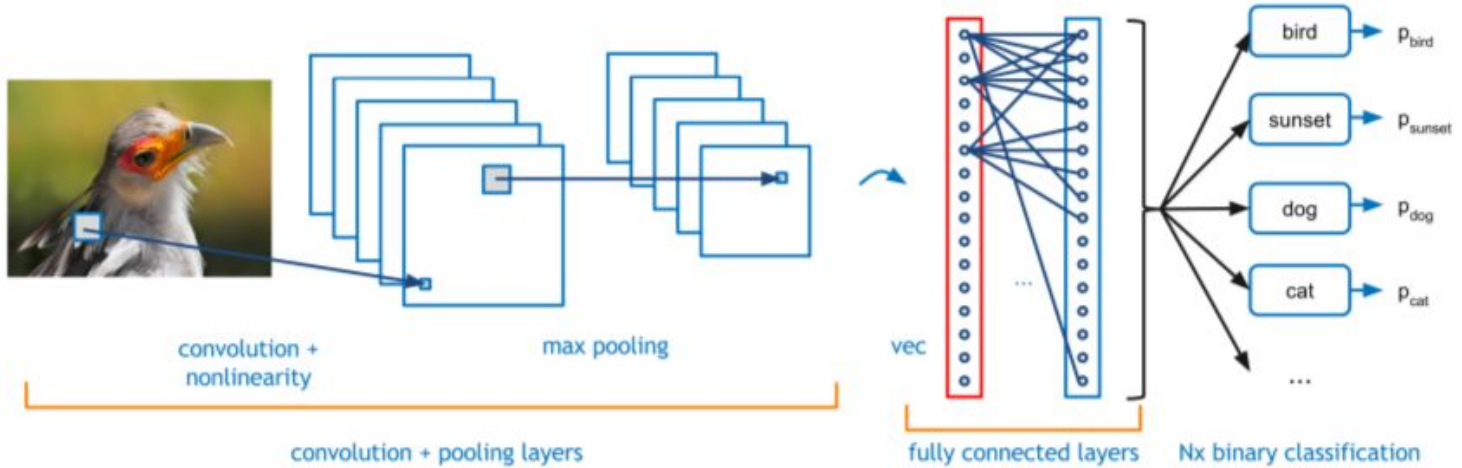
Convolution NN overview

- *A full ConvNN is a sequence of Con2D+Pooling (+BatchNormalization+ Dropout) layers*
- *The Conv+Pooling layer reduces the 2D image representation*
- *The use of multiple filters on the image make the output grow on a third dimension*
- *Eventually, flattening occurs and the result is given to a dense layer*



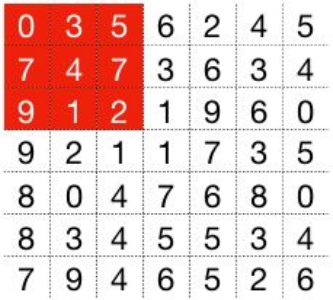
CNN overview

- *Special architectures read the raw information (e.g., images) and convert them into "smart variables" (high-level features) to accomplish the task*
- *Typical example: convolutional neural networks for image processing & computing vision*

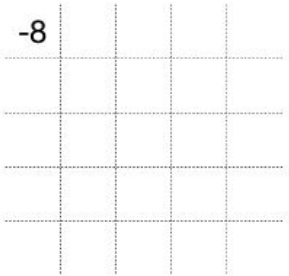


CNN introduction: convolution

- *The main ingredient of ConvNN is a filter, a k x k' matrix of weights*
- *The filter scans the image and performs a scalar product of each image patch*
- *This results into a new matrix of values, with different dimensionality*

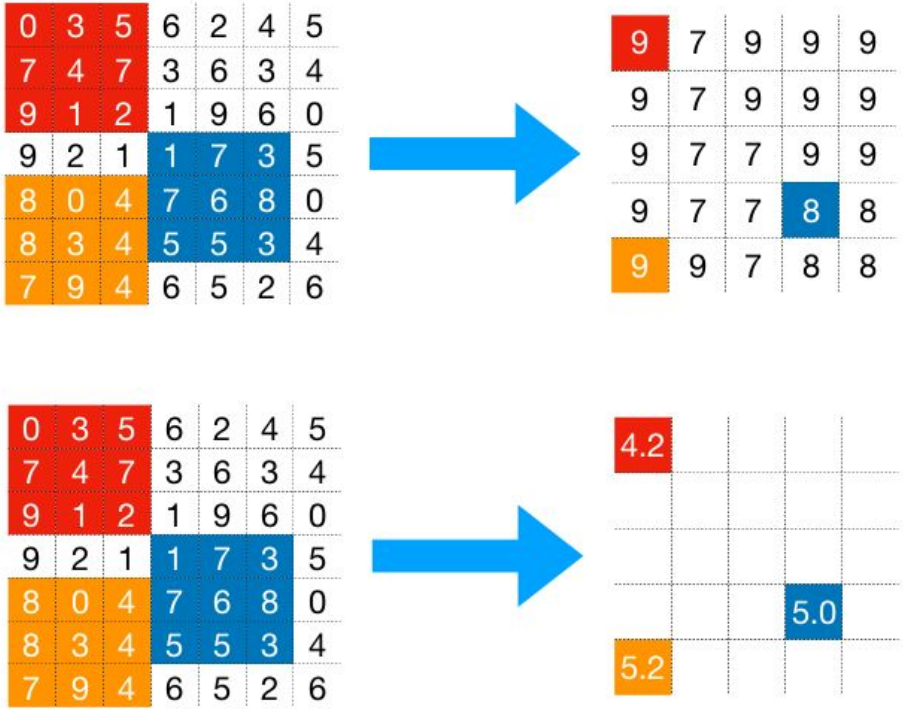


0x4 - 3x1 + 5x4 +
-7x2 + 4x2 - 7x5 +
9x3 + 1x1 - 2x6 = -8



CNN introduction: pooling

- *MaxPooling: Given an image and a filter of size k x k', scans the image and replaces each k x k' patch with its maximum*
- *AveragePooling: Given an image and a filter of size k x k', scans the image and replaces each k x k' patch with its average*
- …

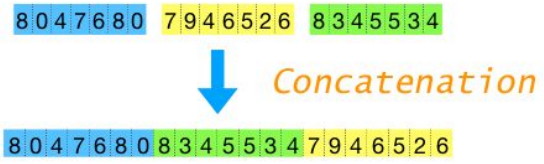


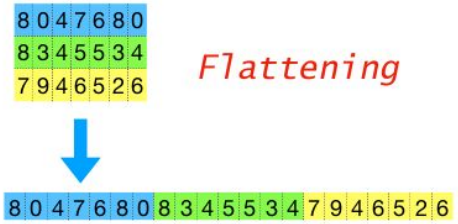
CNN introduction: Padding, Flattening, Inception

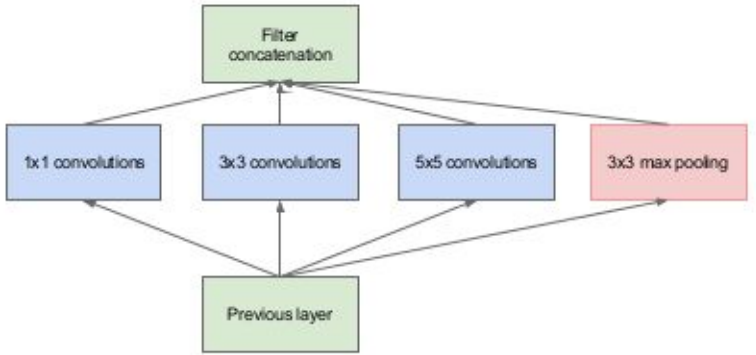
- *When the filter arrived at the edge, it might exceeds it (if n/k is not an integer)*
- *In this case, a padding rule needs to be specified*
  - *Same: repeat the values at the boundary*
  - *Zero: fill the extra columns with zeros*

0	3	5	6	2	4	4
7	4	7	3	6	3	3
9	1	2	1	9	6	6
9	2	1	1	7	3	3
8	0	4	7	6	8	8
8	3	4	5	5	3	3
8	3	4	5	5	3	3

0	3	5	6	2	4	0
7	4	7	3	6	3	0
9	1	2	1	9	6	0
9	2	1	1	7	3	0
8	0	4	7	6	8	0
8	3	4	5	5		
0	0	0	0	0		





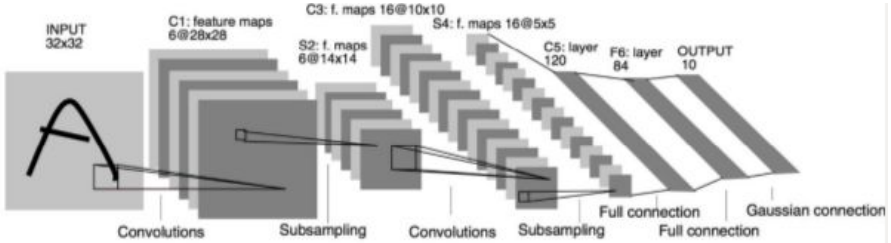


(a) Inception module, naïve version

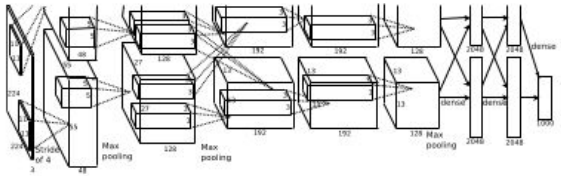
Inception:
Several conv layers, with different filter size, process the same inputs

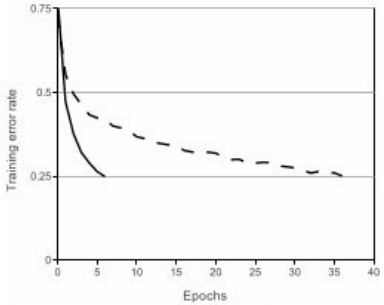
CNN history

- *LeNet (1990s): the very first ConvNN, designer for digit recognition (ZIP codes)*

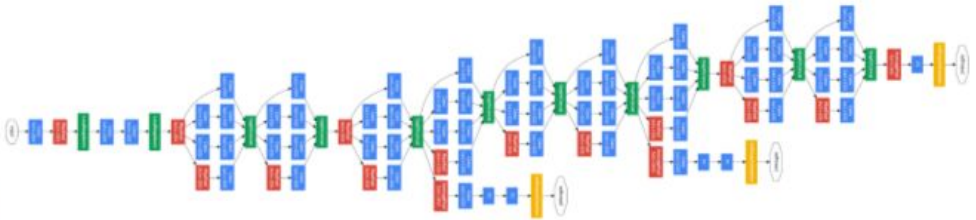


- *AlexNet (2012): the first big ConvNN (60M parameters, 650K neurtons), setting the state of the art: trained on GPUs, using ReLU and Dropout*

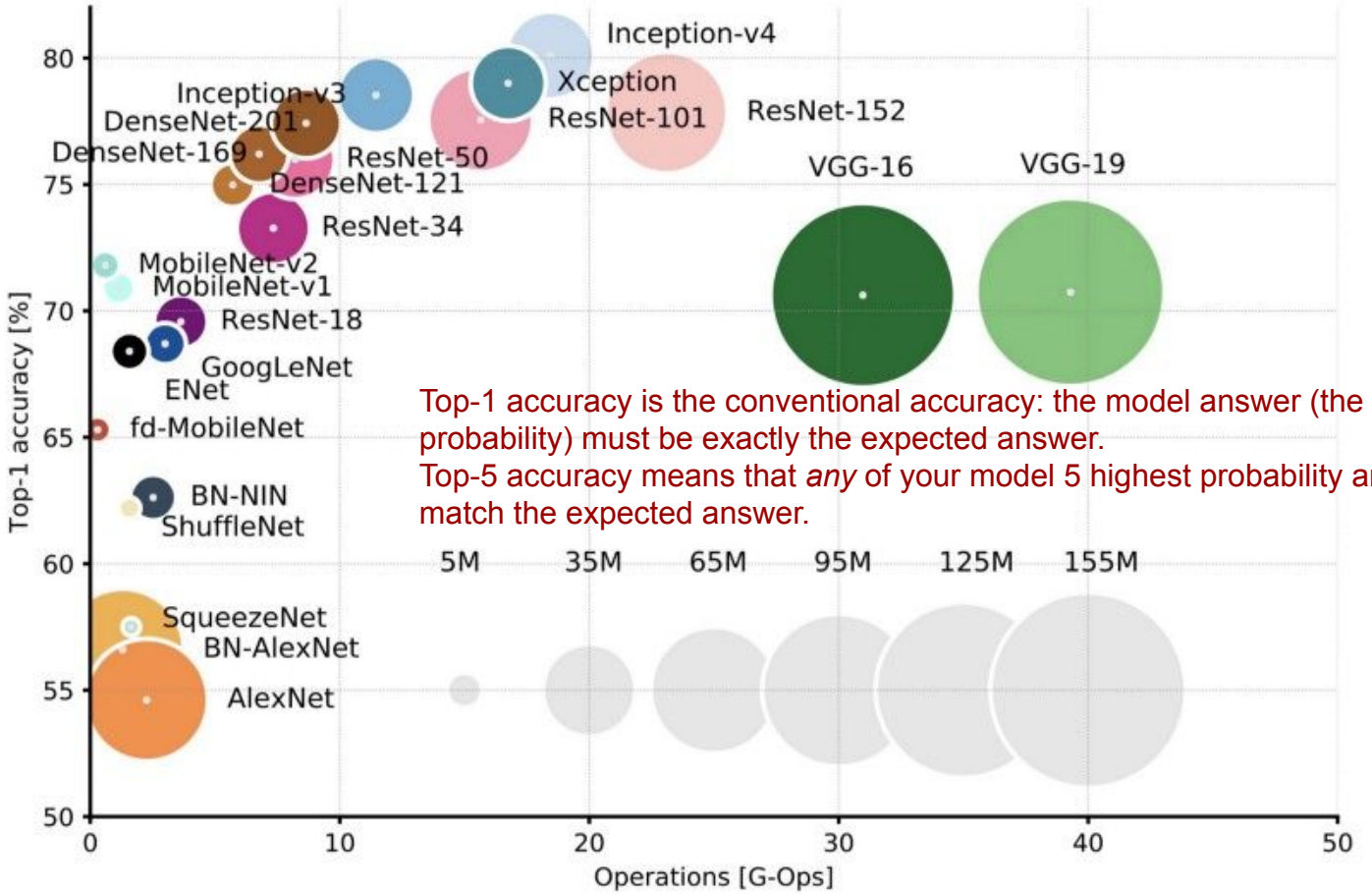




- *GoogleNet (2014): built on AlexNet, introduced an inception model to reduce e the number of parameters*

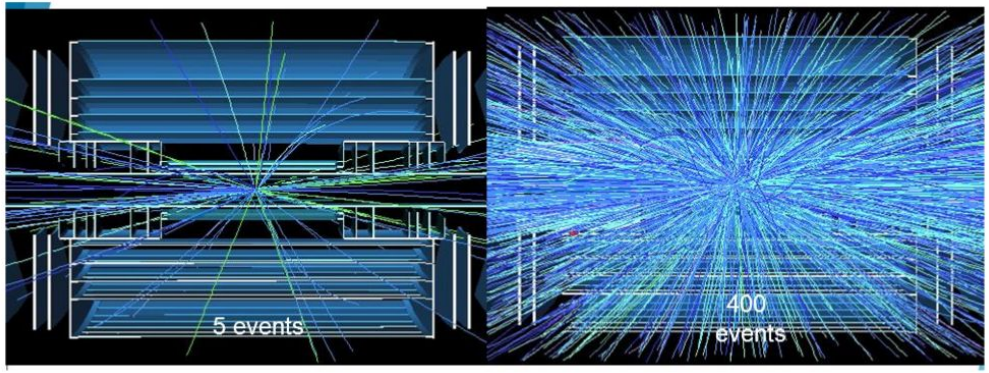


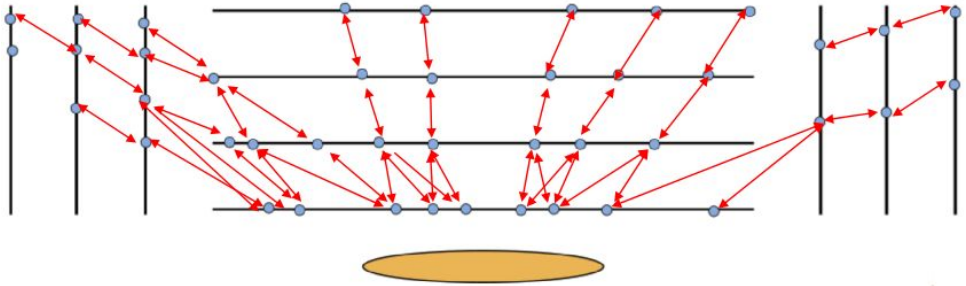
CNN history



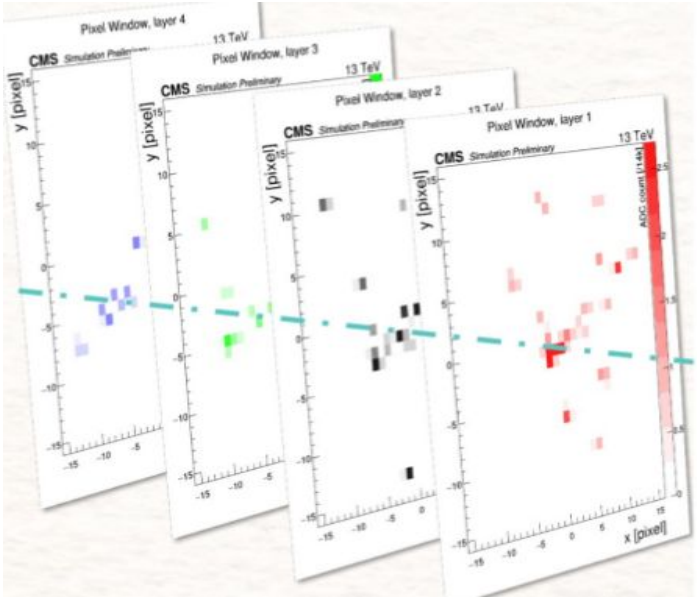
Top-1 accuracy is the conventional accuracy: the model answer (the one with highest probability) must be exactly the expected answer.
Top-5 accuracy means that *any* of your model 5 highest probability answers must match the expected answer.

Application example 4: Tracking reconstruction





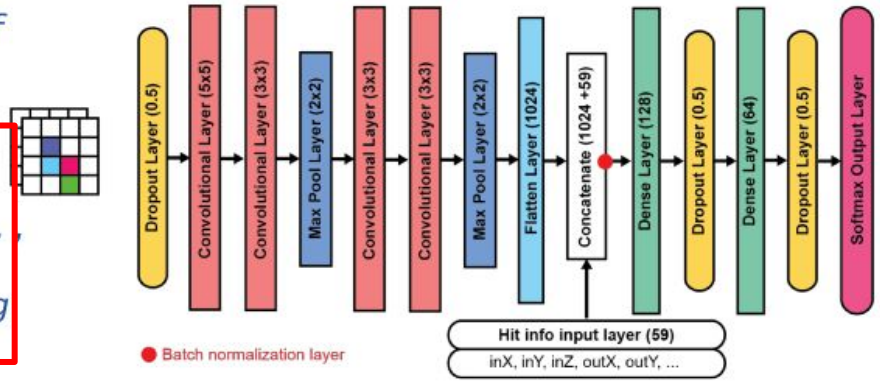
Quite challenging to reconstruct charged particle's track in dense environment: combinatorial complexity, fake seeds...



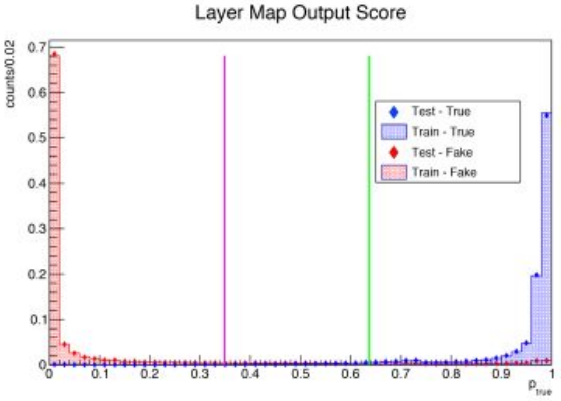
Application example 4: Tracking reconstruction

PixelSeed ConvNN

- The final model uses two sets of inputs:
  - the hit images
  - a set of expert features (e.g., position of the hits in the detector) to help the learning process



- The trained model shows a good separation of true vs fake seeds

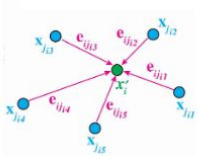


- **One can reduce the fake rate by one order of magnitude with a few % loss in efficiency**

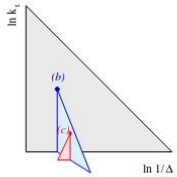
```
Efficiency (tpr) @ fake rejection
tpr @ rej 50%: 0.998996700259
tpr @ rej 75%: 0.990524391331
tpr @ rej 90%: 0.922210826719
tpr @ rej 99%: 0.338669401587
```

Application example 5: Jet tagging

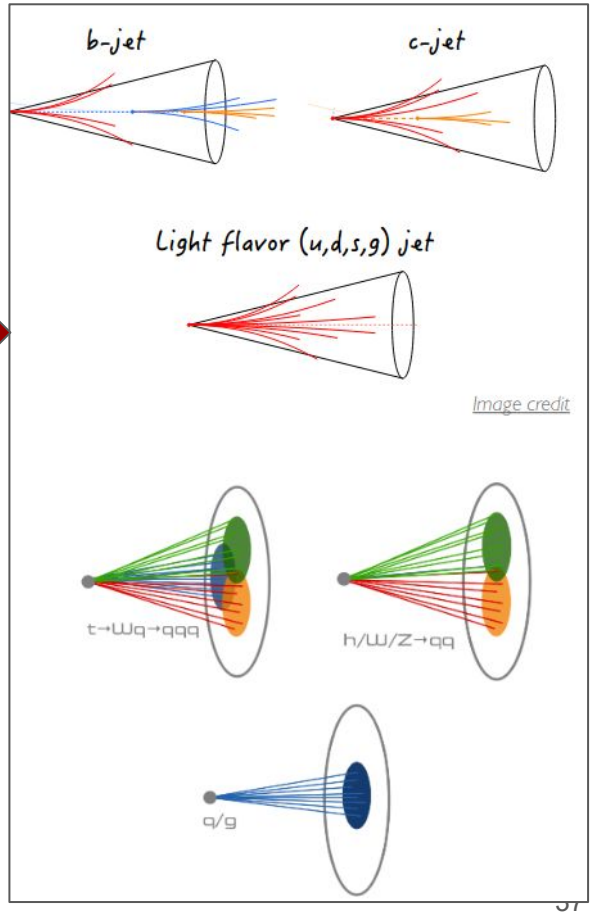
DeepAK8: ConvNet on particle-flow candidates



ParticleNet: jet tagging via particle clouds



LundNet: graph neural network in the Lund plane



See more details from Huilin Qu's talk

Application example 5：Jet tagging

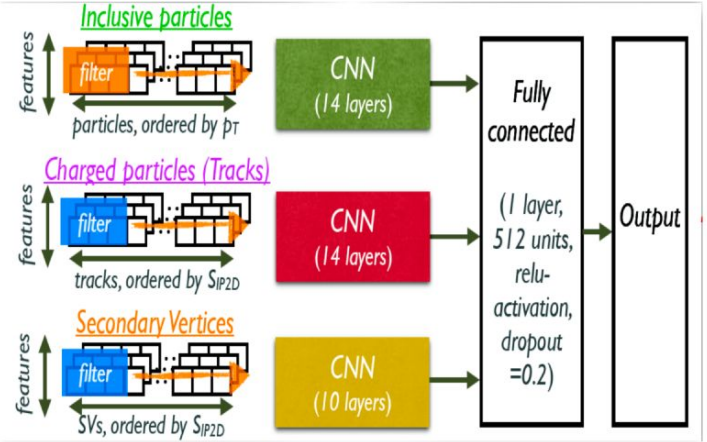
https://indico.cern.ch/event/783781/contributions/3389493/attachments/1832744/3001915/Deep_Heavy_Resonance_Tagging_HEP2019_Kontaxakis.pdf

DeepAK8 and ParticleNet are now the CMS standards for H/W/Z Jet tagging

DeepAK8

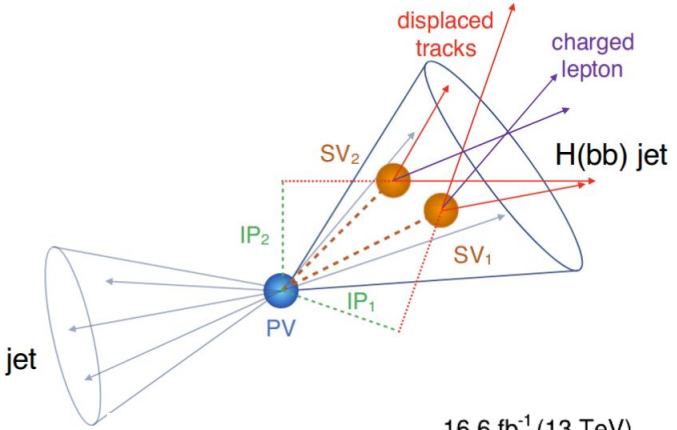
CMS-DP-2017-049
10.1007/JHEP10(2017)005

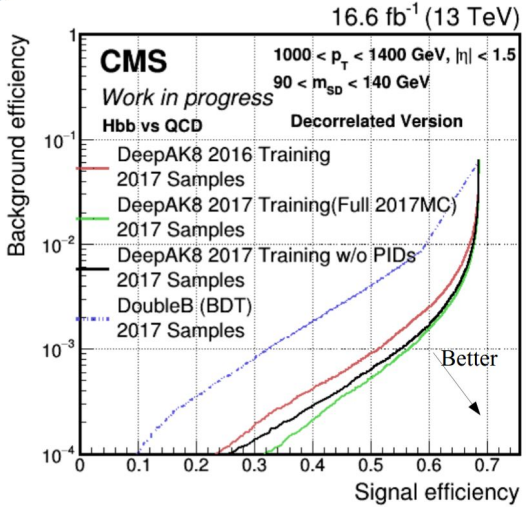
- Deep AK8 takes advantage of this additional information
  - Includes particle and detector-level quantities (tracking, vertex formation)
  - Individual jet constituents as inputs
- Uses convolutional NNs to take advantage of nearby correlations



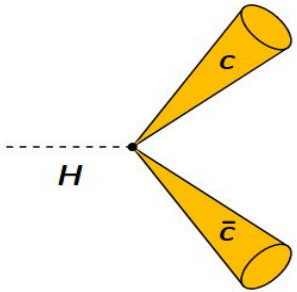
Many output categories!

Category	Label
Higgs	H (bb)
	H (cc)
	H (VV*→qqqq)
Top	top (bcq)
	top (bqq)
	top (bc)
	top (bq)
W	W (cq)
	W (qq)
Z	Z (bb)
	Z (cc)
	Z (qq)
QCD	QCD (bb)
	QCD (cc)
	QCD (b)
	QCD (c)
	QCD (others)





Application example 5 : Hcc



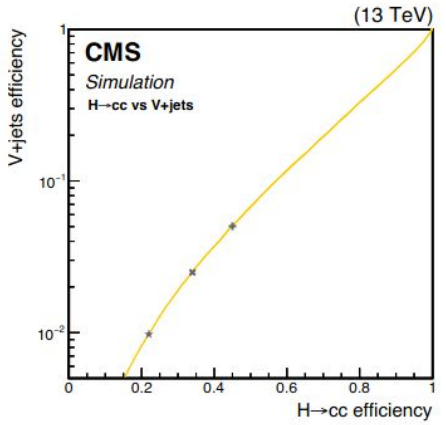
Higgs coupling with 2nd generation fermion

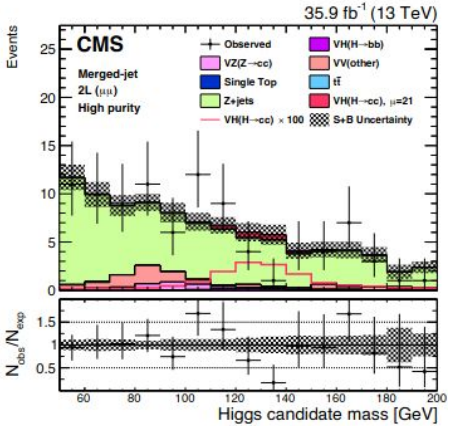
- DeepAK8(-MD) has become the standard boosted jet tagging algorithm in CMS and has been used in several high-profile analyses
- First direct search for H→cc in CMS
  - VH channel: V (W, Z) → ll, lv, vv
  - H→cc: resolved-jet topology + merged-jet topology
- DeepAK8-MD used in the merged-jet topology
  - adapted to R=1.5 jets (instead of R=0.8 jets) to increase acceptance at lower pT (> ~200 GeV)
  - the DeepAK8-MD cc-tagging discriminant used to select cc-jet and suppress light-/bb-flavor jets
  - fit to the soft-drop jet mass distribution to extract the H→cc signal
- Most stringent direct limit on H→cc to date

95% CL exclusion limit on $\mu_{VH(H\to c\bar{c})}$

	Resolved-jet	Merged-jet	Combination			
	$(p_T(V) < 300\,GeV)$	$(p_T(V) \geq 300\,GeV)$	0L	1L	2L	All channels
Expected	45^{+18}_{-13}	73^{+34}_{-22}	79^{+32}_{-22}	72^{+31}_{-21}	57^{+25}_{-17}	37^{+16}_{-11}
Observed	86	75	83	110	93	70

- cf. ATLAS [PRL 120 (2018) 211802]: $\mu_{ZH(H\to cc)}$ < 110 (150) obs. (exp.)





CMS [JHEP 03 (2020) 131]

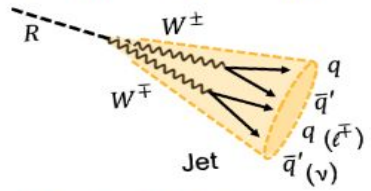
Application example 5：WWW resonance

- First search of TeV scale resonances decaying to three W bosons
  - cascade decay: W_{KK} -> W(lv) + Radion

Analysis led by the PKU group

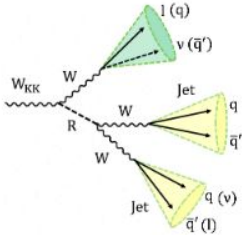
[CMS-PAS-B2G-20-001]

$R \approx R^{4q} + R^{3q} + R^{lqq}$

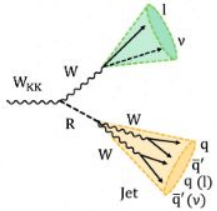


We do NOT have standard candle in SM to calibrate all these

Resolved Radion



Merged Radion

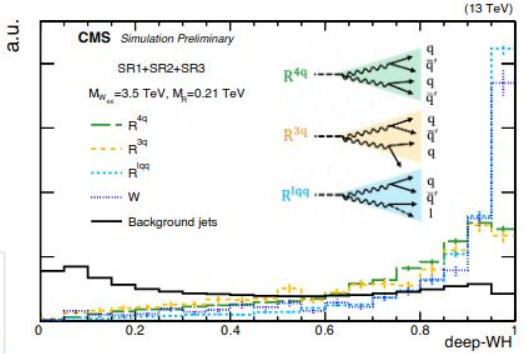


W-tagging w/ DeepAK8-MD

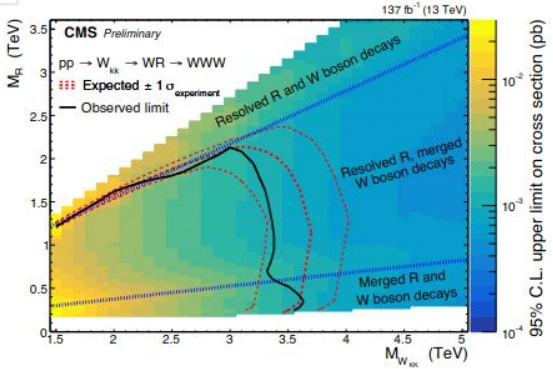
$$\frac{score(W \to cq, qq)}{score(W \to cq, qq) + score(QCD)}$$

Hybrid tagging w/ DeepAK8-MD

$$\frac{score(W \to cq, qq) + score(H \to 4q)}{score(W \to cq, qq) + score(H \to 4q) + score(QCD)}$$



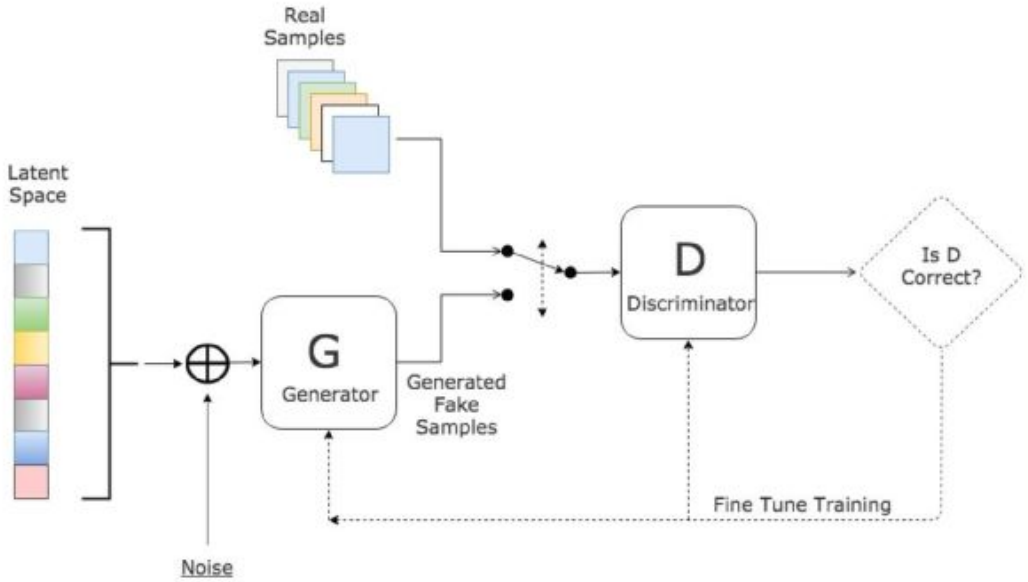
- innovative usage of the DeepAK8-MD tagger to identify merged radion decay
- new approach developed for the calibration of the DeepAK8-MD tagger
- final signal extraction based on the invariant mass of the lv+jet(s) system



B2G-20-001

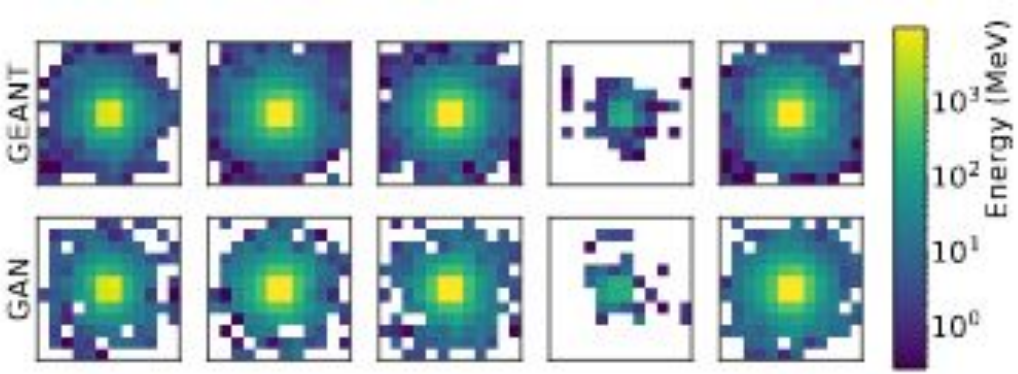
Generative Adversarial Network (GAN)

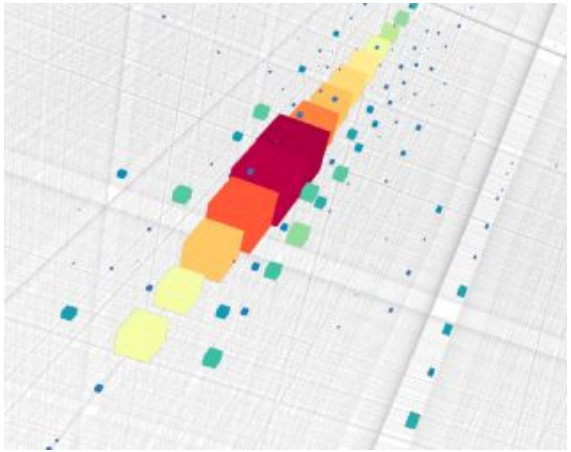
- *Two networks trained against each other*
  - *Generator: create images (from noise, other images, etc)*
  - *Discriminator: tries to spot which image comes from the generator and which is genuine*

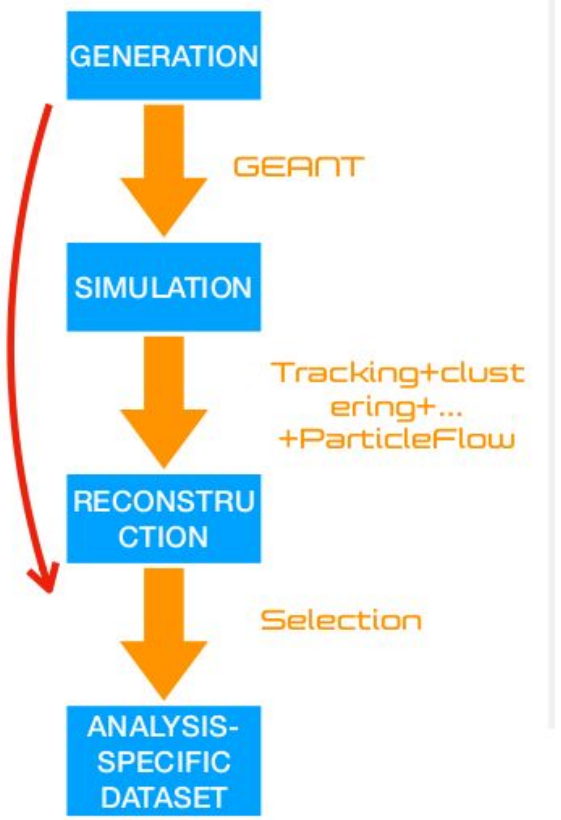


- ***Loss function to minimise: Loss(Gen)-Loss(Disc)***
  - ***Better discriminator -> bigger loss***
  - ***Better generator -> smaller loss***
- ***Trying to full the discriminatore, generatore learns how to create more realistic images***

Application example 6: fast simulation and reconstruction

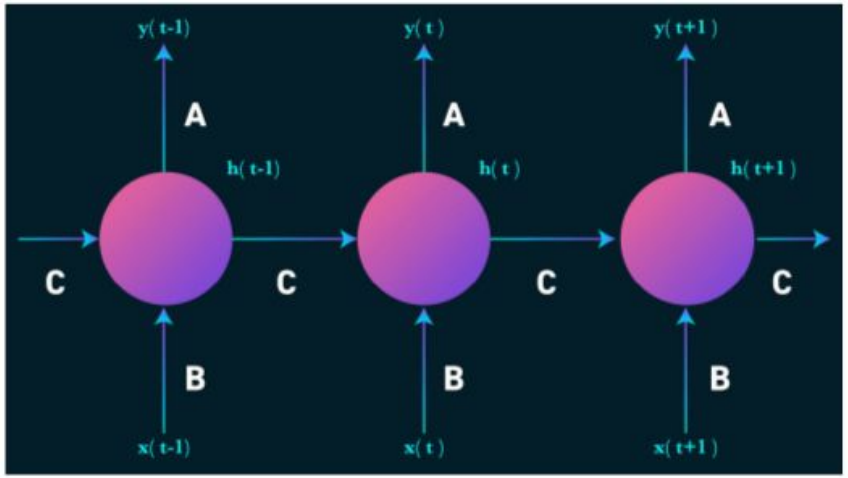






Recurrent NN

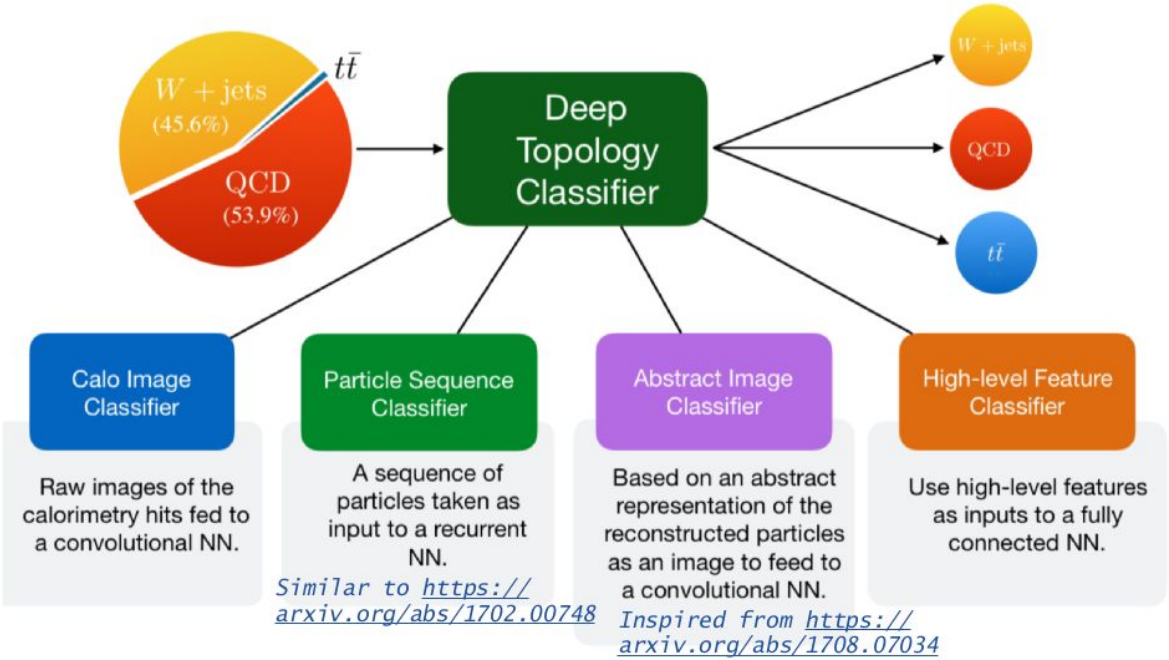
- Recurrent architectures are designed to process sequences of data
- Then idea is to have information flowing in the network while the sequence is sequentially processed
- Through this idea, recurrent networks mimic memory persistence
- Advantages
  - the input is not fixed-sized



Application example 7: RNN for classification

- particles as words in a sentence
- QCD is the grammar

four-momenta are like words and the clustering history of sequential recombination jet algorithms is like the parsing of a sentence.

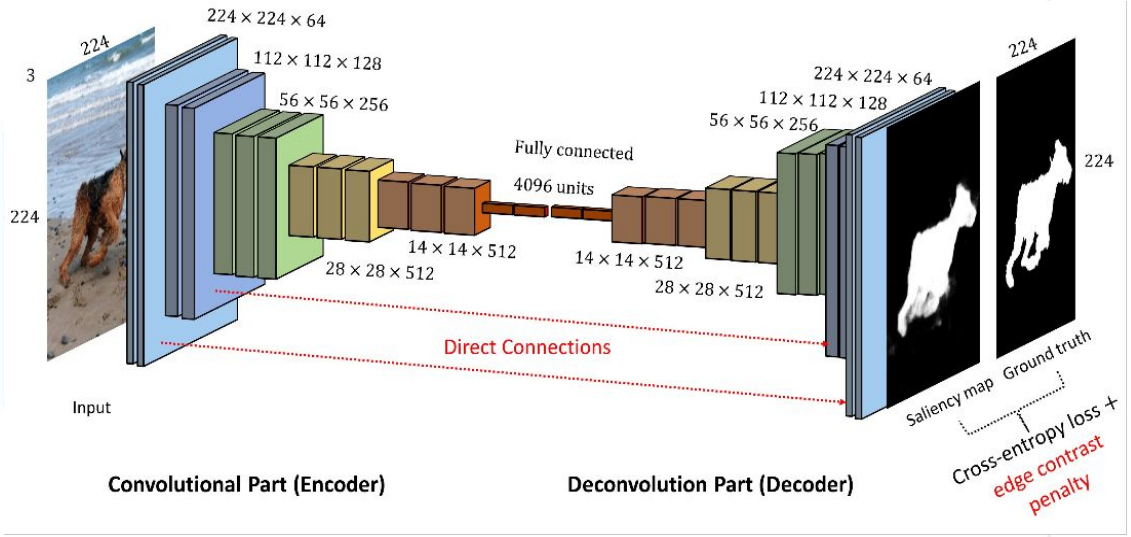


Autoencoders

- Autoencoders are networks with a typical "bottleneck" structure, with a symmetric structure around it
  - They go from $\mathbb{R}^n \rightarrow \mathbb{R}^n$
  - They are used to learn the identity function as $f^{-1}(f(x))$

where $f: \mathbb{R}^n \rightarrow \mathbb{R}^k$ and $f^{-1}: \mathbb{R}^k \rightarrow \mathbb{R}^n$

- Autoencoders are essential tools for unsupervised studies

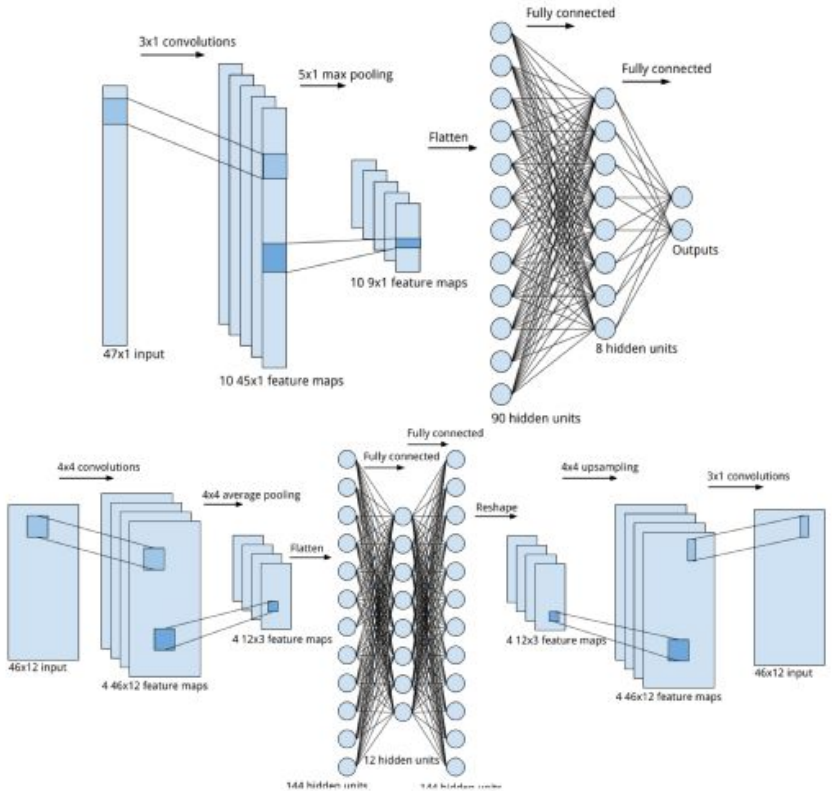


https://github.com/arthurmeyer/Saliency_Detection_Convolutional_Autoencoder

Application example 8: Data Quality Monitoring

- *Given the nature of these data, ConvNN are a natural analysis tool. Two approaches pursued*
  - *Classify good vs bad data. Works if failure mode is known*
  - *Use autoencoders to assess data "typicality". Generalises to unknown failure modes*

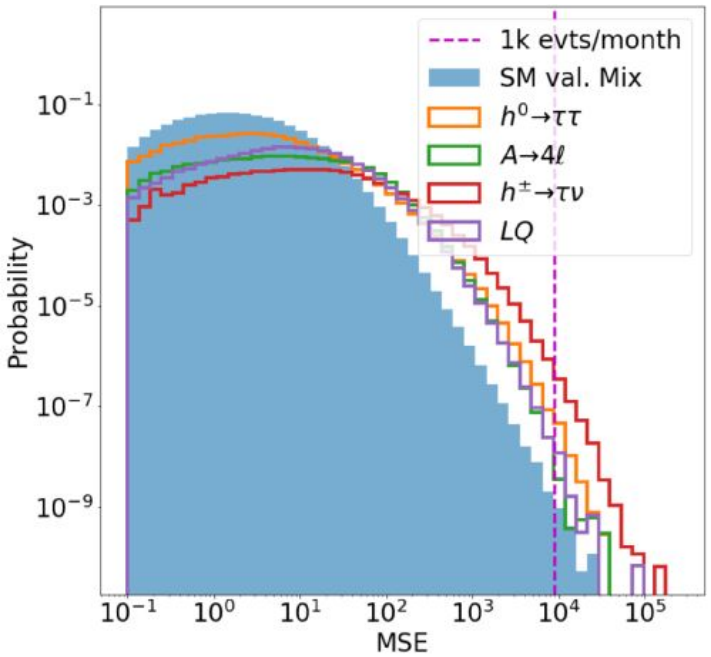
A. Pol et al., to appear soon



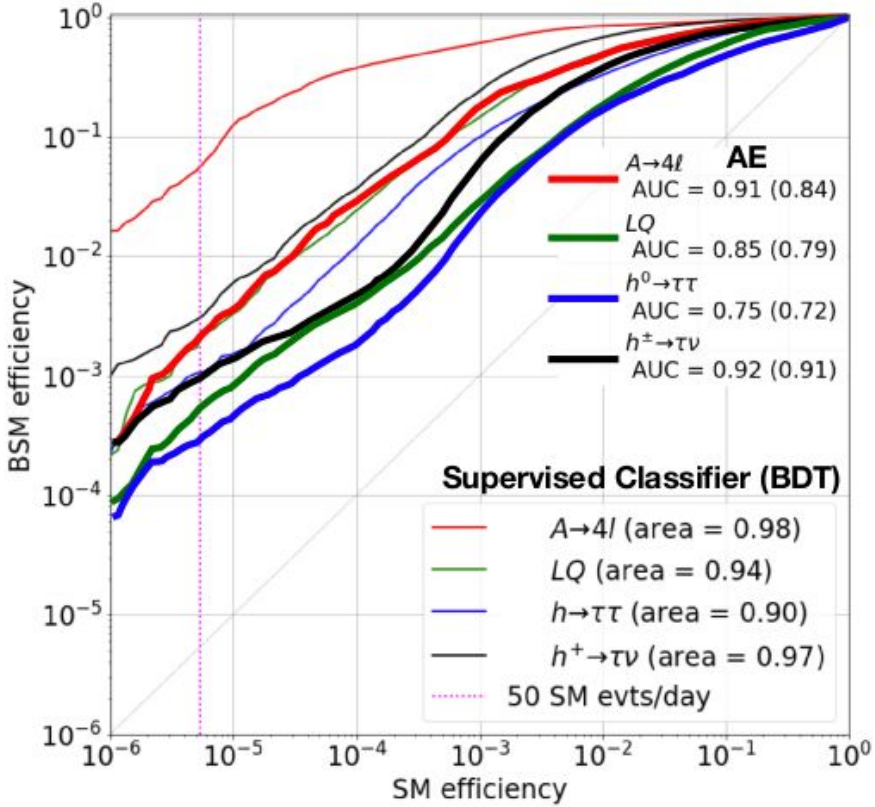
Pol, G. Cerminara, C. Germain, MP and A. Seth arXiv:1808.00911

Application example 8: Anomaly detection

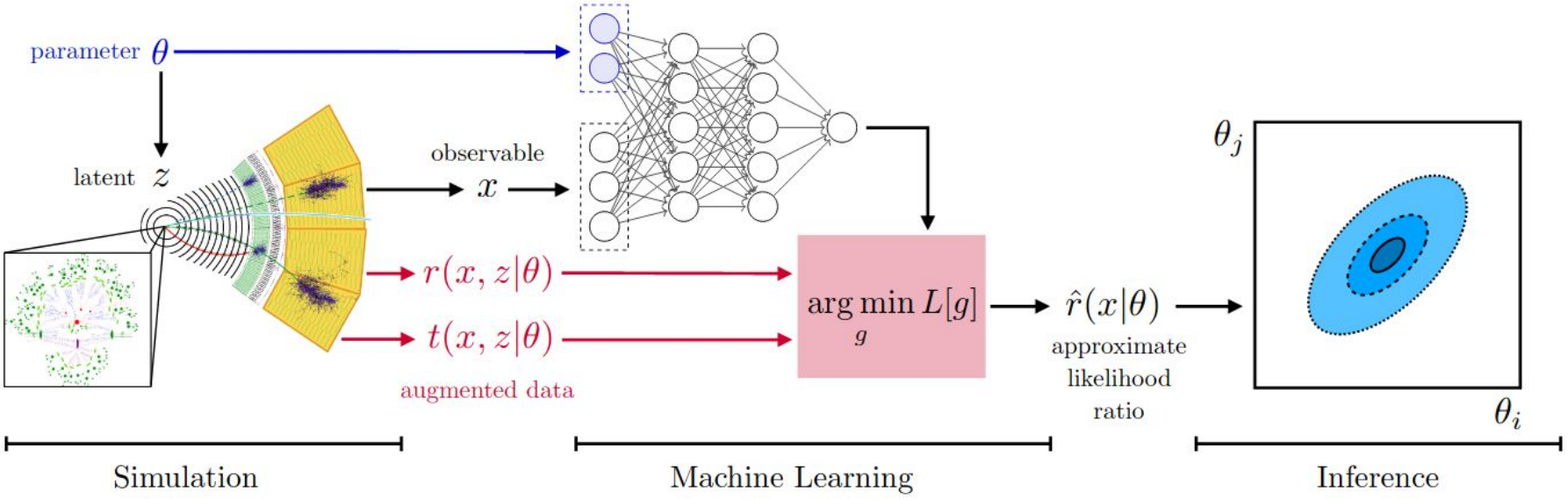
- Train on standard events
- Run autoencoder on new events
- Consider as anomalous
  all events with loss > threshold



Worse than Supervised but results encouraging



Others: ML for EFT

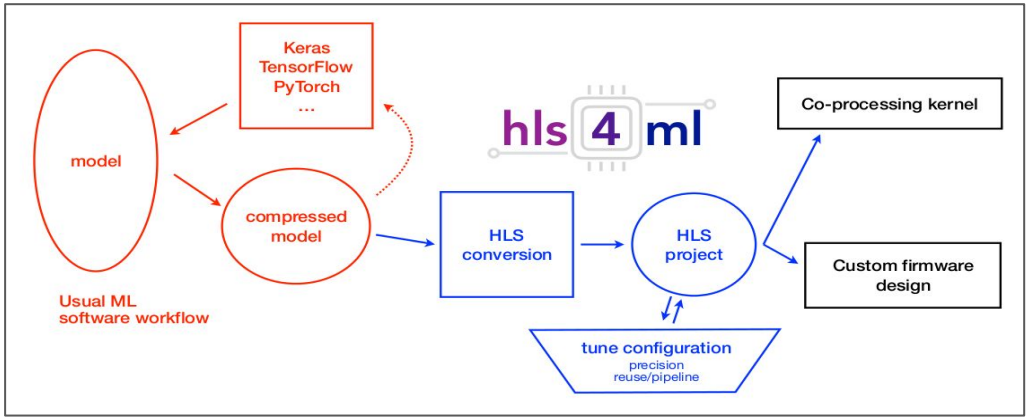


$$p(x|\theta) = \int \mathrm{d}z_{\mathrm{detector}} \int \mathrm{d}z_{\mathrm{shower}} \int \mathrm{d}z \underbrace{p(x|z_{\mathrm{detector}})\, p(z_{\mathrm{detector}}|z_{\mathrm{shower}})\, p(z_{\mathrm{shower}}|z)\, p(z|\theta)}_{=p(x, z_{\mathrm{detector}}, z_{\mathrm{shower}}, z|\theta)} .$$

EFT is continuous; many latent variables; using special DNN and loss function to regress likelihood ratio.

Others: Firmware/hardware e.g. FPGA

Map DNN nicely into an FPGA



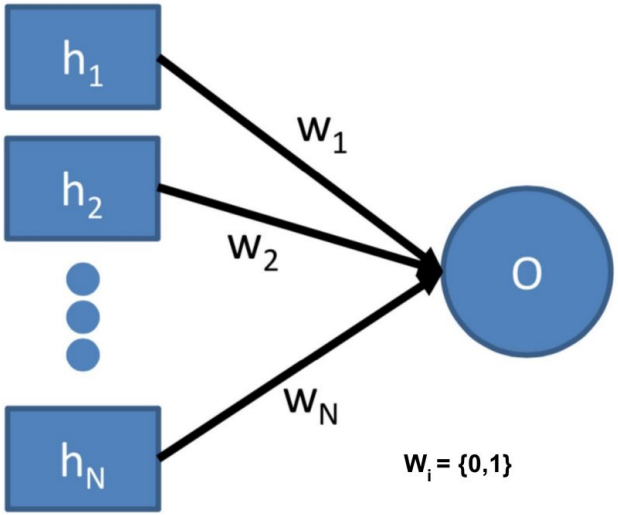
hls 4 ml

https://github.com/hls-fpga-machine-learning/hls4ml

DOI 10.5281/zenodo.1204445

A package for machine learning inference in FPGAs. We create firmware implementations of machine learning algorithms using high level synthesis language (HLS). We translate traditional open-source machine learning package models into HLS that can be configured for your use-case!

Others: ML with quantum computing



quantum annealing

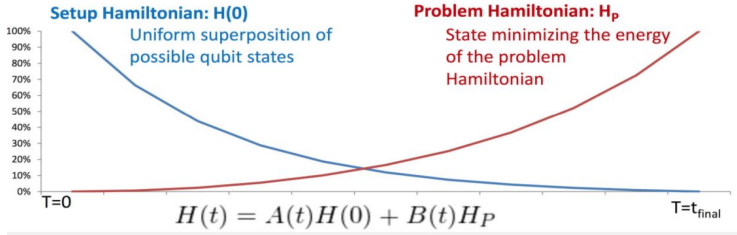
$H(0) = \sum_i \sigma_i^x$

$H_P = \sum_{i,j} J_{ij} s_i s_j + \sum_i h_i s_i$

H_P is effectively $\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_{iy}) w_i$

σ_i^x has a ground state of proportional to $|0\rangle + |1\rangle$

H(0) has no interactions, so cools to ground state quickly, and the total ground state is an equal superposition over all bitstrings



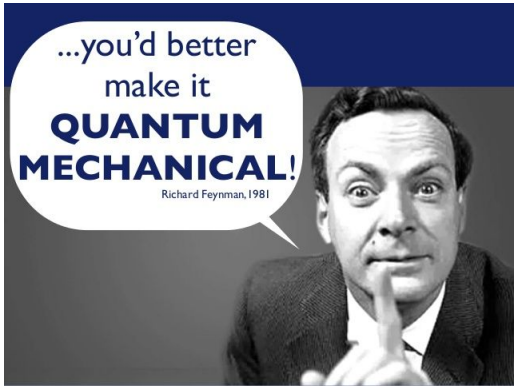
nature
International journal of science

Letter | Published: 18 October 2017

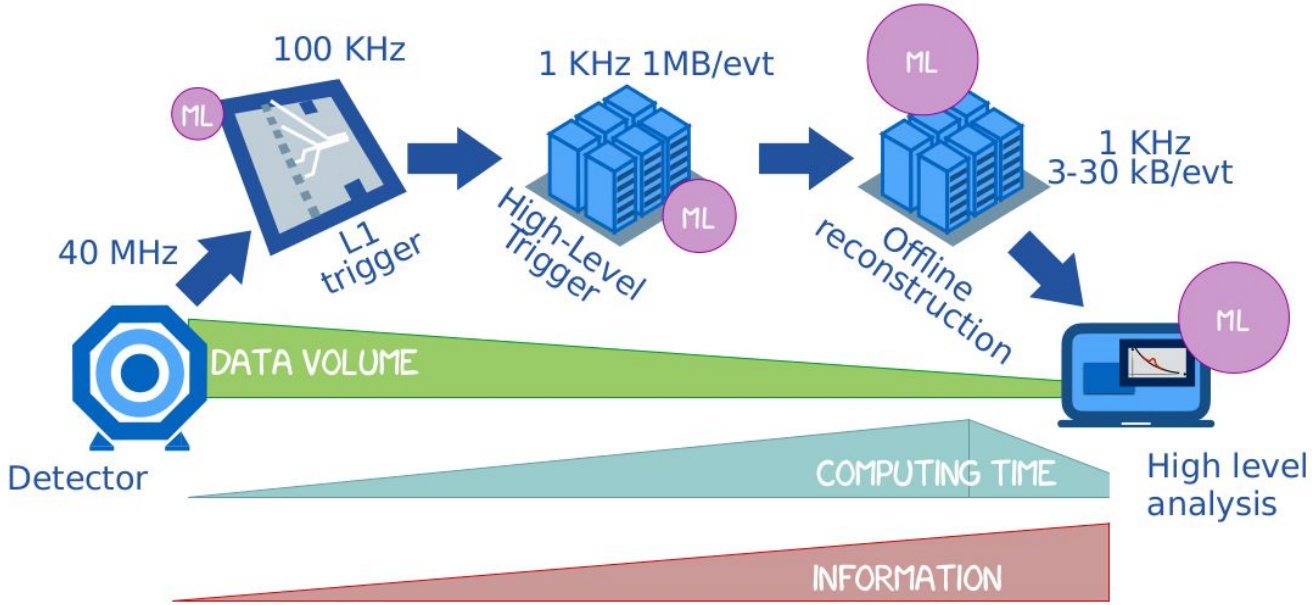
Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott, Joshua Job, Jean-Roch Vlimant, Daniel Lidar & Maria Spiropulu

Nature **550**, 375–379 (19 October 2017) | Download Citation



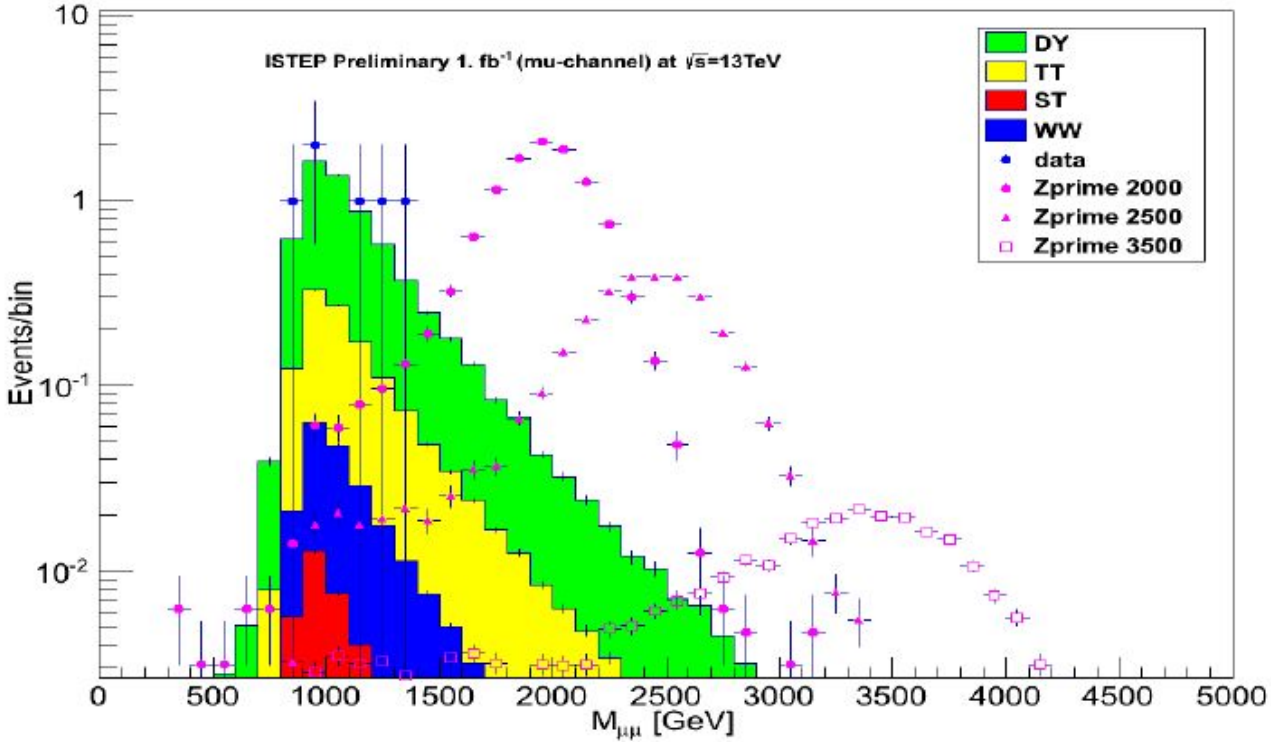
Summary



Deeper and Deeper in HEP

Tutorial： BDT, DNN

Z’ search: https://pan.baidu.com/s/1b54D2m

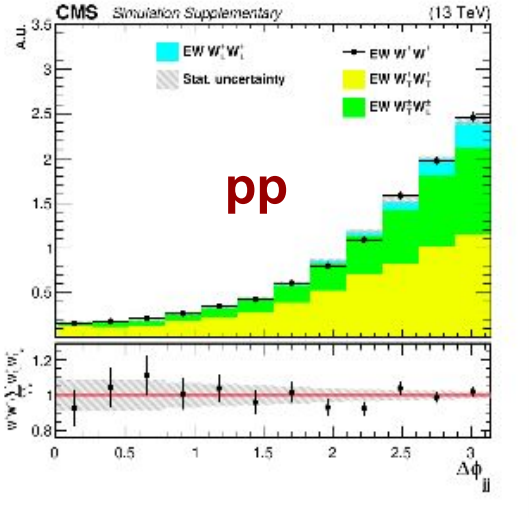


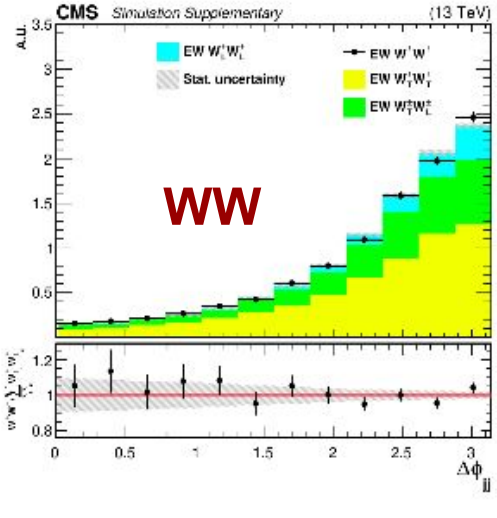

P
+
BACKUP

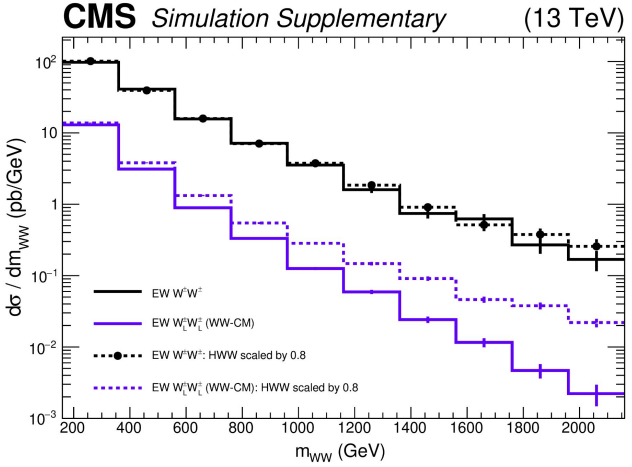
First Probe from CMS on Polarized VBS

- Signal sample simulated in **WW/pp center-of-mass frame**
- Simultaneous fit in bins of two BDT discriminant variables:

MADGRAPH5_aMC@NLO2.7.2







Approval of SMP-20-006 : Measurements of the scattering of polarized same-sign WW bosons

Speakers: Aram Apyan (Fermi National Accelerator Lab. (US)) , Mr Jie Xiao (Peking University (CN))

Phys. Lett. B 812 (2020) 136018

First Probe from CMS on Polarized VBS

Inclusive BDT: **Isolate VBS against non VBS background**

Variables	Definitions
m_{jj}	Dijet mass
$\lvert\Delta\eta_{jj}\rvert$	Difference in pseudorapidity between the leading and subleading jets
$\Delta\phi_{jj}$	Difference in azimuth angles between the leading and subleading jets
p_T^{j1}	p_T of the leading jet
p_T^{j2}	p_T of the subleading jet
$p_T^{\ell_1}$	Leading lepton p_T
$p_T^{\ell\ell}$	Dilepton p_T
$z^*_{\ell_1}$	Zeppenfeld variable of the leading lepton
$z^*_{\ell_2}$	Zeppenfeld variable of the subleading lepton
p_T^{miss}	Missing transverse momentum

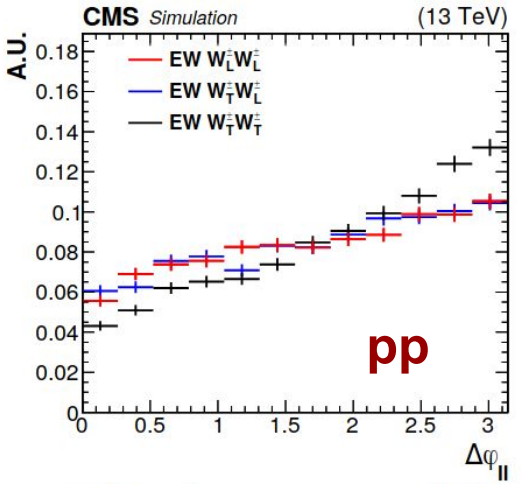
Process	Yields in $W^\pm W^\pm$ SR
$W^\pm_L W^\pm_L$	16.0 ± 18.3
$W^\pm_L W^\pm_T$	63.1 ± 10.7
$W^\pm_T W^\pm_T$	110.1 ± 18.1
QCD $W^\pm W^\pm$	13.8 ± 1.6
Interference $W^\pm W^\pm$	8.4 ± 0.6
WZ	63.3 ± 7.8
ZZ	0.7 ± 0.2
Nonprompt	213.7 ± 52.3
tVx	7.1 ± 2.2
Other background	26.9 ± 9.9
Total SM	522.9 ± 60.7
Data	524

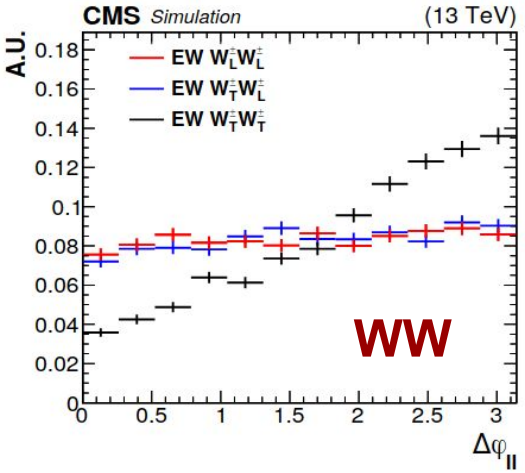
First Probe from CMS on Polarized VBS

Signal BDTs to improve the sensitivity to polarized scattering

Train LL against (LT+TT) and train (LL+LT) against TT

Variables	Definitions
$\Delta\phi_{jj}$	Difference in azimuthal angle between the leading and subleading jets
p_T^{j1}	p_T of the leading jet
p_T^{j2}	p_T of the subleading jet
$p_T^{\ell_1}$	Leading lepton p_T
$p_T^{\ell_2}$	Subleading lepton p_T
$\Delta\phi_{\ell\ell}$	Difference in azimuthal angle between the two leptons
$m_{\ell\ell}$	Dilepton mass
$p_T^{\ell\ell}$	Dilepton p_T
m_T^{WW}	Transverse WW diboson mass
$z^*_{\ell_1}$	Zeppenfeld variable of the leading lepton
$z^*_{\ell_2}$	Zeppenfeld variable of the subleading lepton
$\Delta R_{j1,\ell\ell}$	ΔR between the leading jet and the dilepton system
$\Delta R_{j2,\ell\ell}$	ΔR between the subleading jet and the dilepton system
$(p_T^{\ell_1} p_T^{\ell_2})/(p_T^{j1} p_T^{j2})$	Ratio of p_T products between leptons and jets
p_T^{miss}	Missing transverse momentum



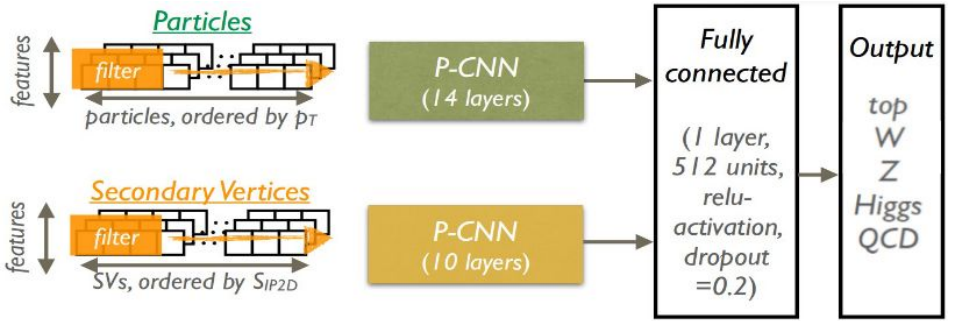


Deep Learning Tagger in CMS

DeepAK8 JINST 15 (2020) P06005

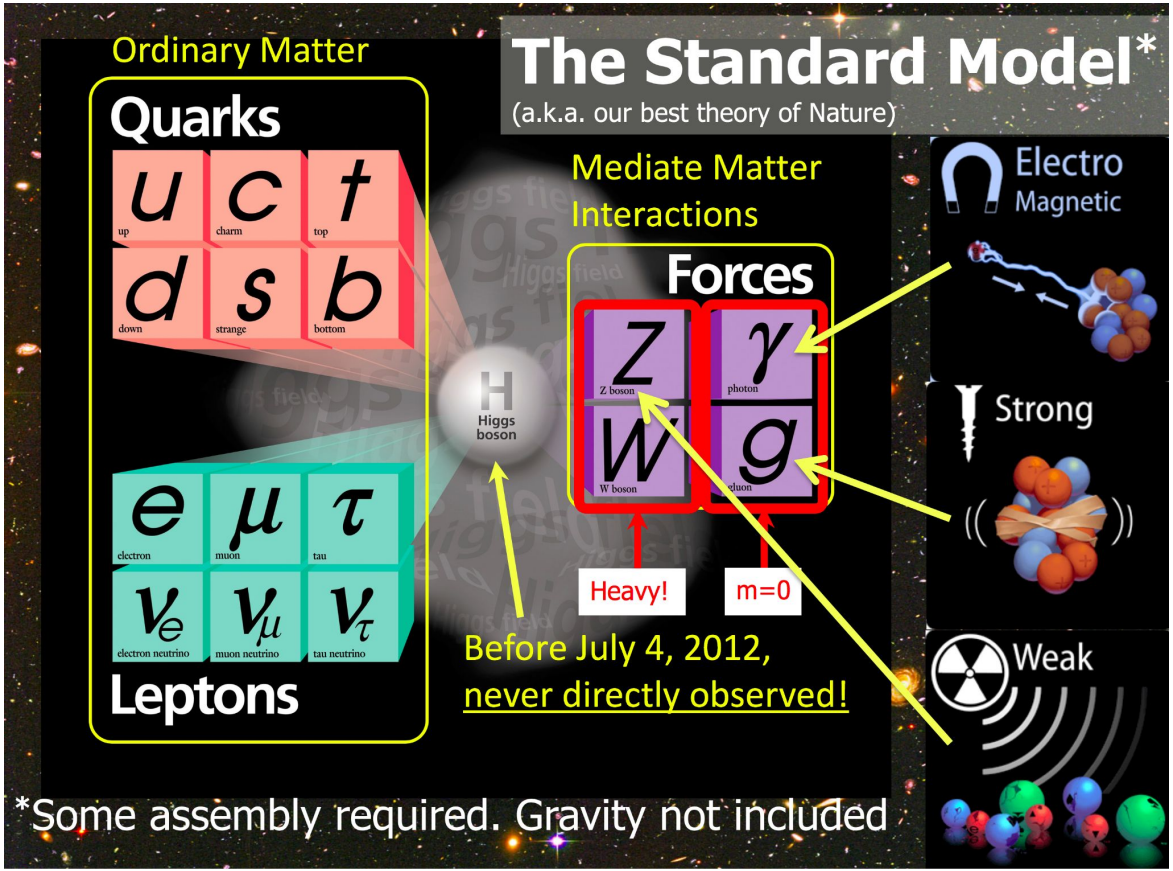
- **multi-class classifier** for t/W/Z/H tagging
  - categories subdivided based on decay modes
- **directly uses jet constituents**
  - **PF candidates / secondary vertices**
- 1D CNN based on the ResNet architecture
- **mass-decorrelated version** using **adversarial** training techniques
  - signal and background samples reweighted to yield flat distributions in both pT and mSD to aid the training.

new standards

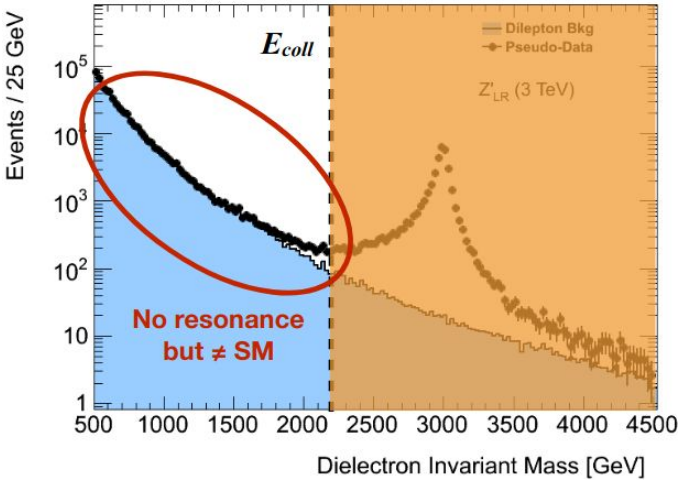


Output

Category	Label
Higgs	H (bb)
	H (cc)
	H (VV*→qqqq)
Top	top (bcq)
	top (bqq)
	top (bc)
	top (bq)
W	W (cq)
	W (qq)
Z	Z (bb)
	Z (cc)
	Z (qq)
QCD	QCD (bb)
	QCD (cc)
	QCD (b)
	QCD (c)
	QCD (others)

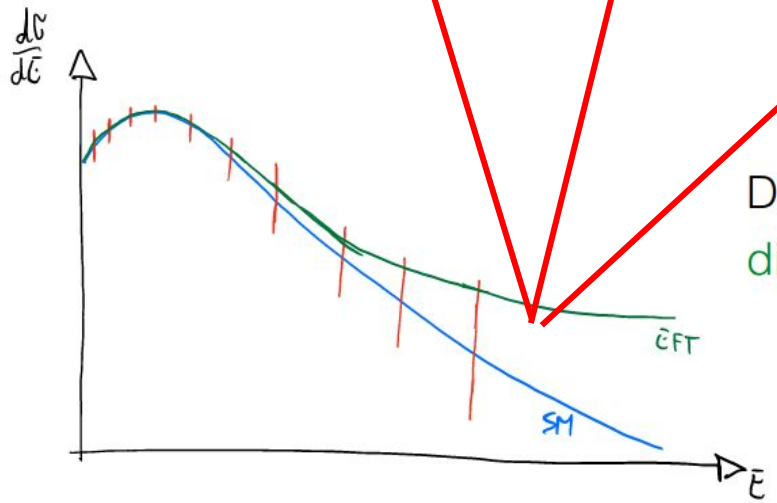


Indirect or direct search for new Particle



- Assuming linear representation for the Higgs, no new light particles, SM symmetries, etc:

$$\mathcal{L}_{SMEFT} = \mathcal{L}_{SM} + \frac{c^{(5)}}{\Lambda}\mathcal{O}^{(5)} + \frac{1}{\Lambda^2}\sum_i c_i^{(6)}\mathcal{O}_i^{(6)} + \sum_j\sum_k \frac{1}{\Lambda^{2+k}} c_j^{(6+k)}\mathcal{O}_j^{(6+k)}$$



Deviations in the tails of differential distributions

传统方法：直方图

Ref

Consider the L^2 squared loss functional for functions $\hat{g}(x)$ that only depend on x, but which are trying to approximate a function $g(x, z)$,

$$L[\hat{g}(x)] = \int \mathrm{d}x\, \mathrm{d}z\; p(x, z|\theta)\; |g(x, z) - \hat{g}(x)|^2$$
$$= \int \mathrm{d}x\; \underbrace{\left[\hat{g}^2(x) \int \mathrm{d}z\; p(x, z|\theta) - 2\hat{g}(x) \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) + \int \mathrm{d}z\; p(x, z|\theta)\, g^2(x, z)\right]}_{F(x)} .$$

Via calculus of variations we find that the function $g^*(x)$ that extremizes $L[\hat{g}]$ is given by [53]

$$0 = \left.\frac{\delta F}{\delta \hat{g}}\right|_{g^*} = 2\hat{g} \underbrace{\int \mathrm{d}z\; p(x, z|\theta)}_{=p(x|\theta)} - 2 \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) ,$$

therefore

$$g^*(x) = \frac{1}{p(x|\theta)} \int \mathrm{d}z\; p(x, z|\theta)\, g(x, z) .$$

We can make use of this general property in our problem in two ways. Identifying $g(x_e, z_e)$ with the joint likelihood ratios $r(x_e, z_{\mathrm{all},e}|\theta_0, \theta_1)$ (which we can calculate!) and $\theta = \theta_1$, we find

$$g^*(x) = \frac{1}{p(x|\theta_1)} \int \mathrm{d}z\; p(x, z|\theta_1)\, \frac{p(x, z|\theta_0)}{p(x, z|\theta_1)} = r(x|\theta_0, \theta_1) . \tag{24}$$

By minimizing the squared loss

$$L[\hat{r}(x|\theta_0, \theta_1)] = \frac{1}{N} \sum_{(x_e, z_e) \sim p(x, z|\theta_1)} |r(x_e, z_{\mathrm{all},e}|\theta_0, \theta_1) - \hat{r}(x_e|\theta_0, \theta_1)|^2 \tag{25}$$