Machine learning for data reconstruction at the LHC

Joosep Pata (NICPB, Estonia)
joosep.pata@cern.ch

Overview

A high-level summary of various aspects of machine learning in LHC data reconstruction, mostly based on CMS examples.

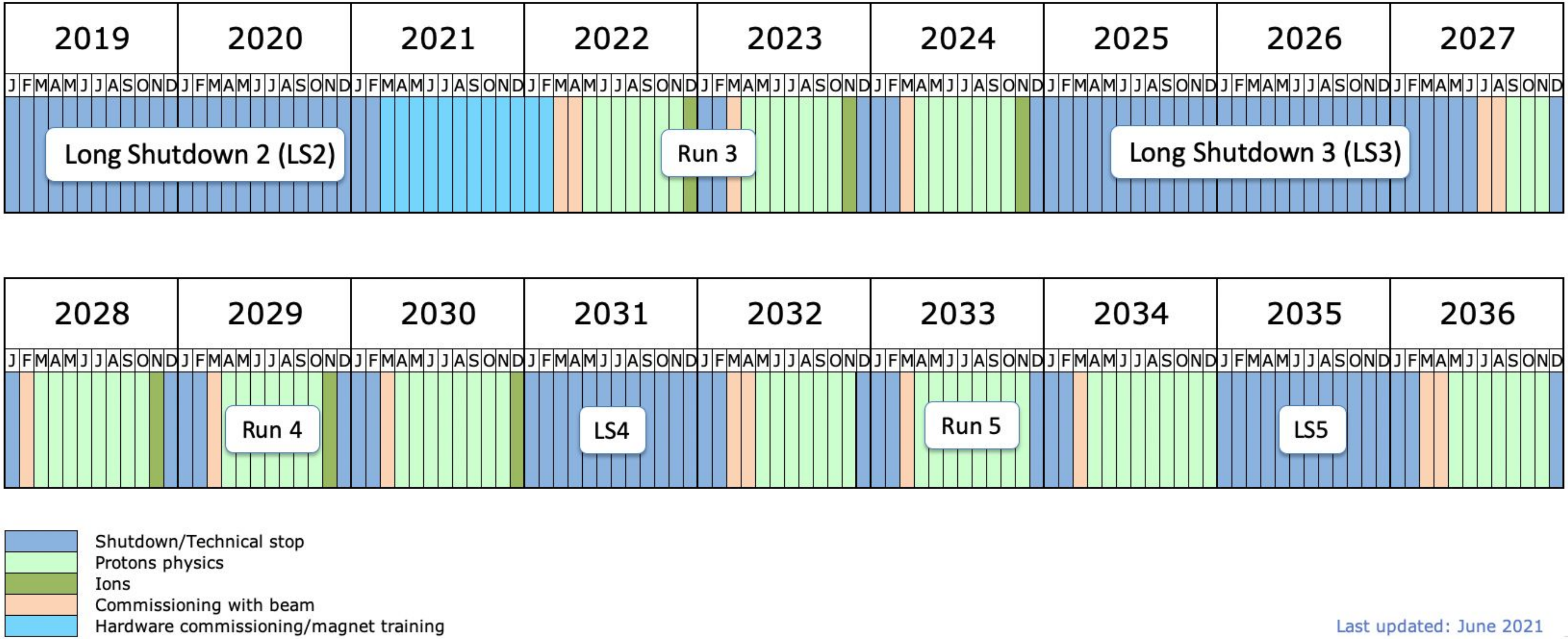
A short summary of a particular use case: ML for combining signals across detector subsystems with particle flow.

This talk is in personal capacity (not representing CMS or CERN), representing my biased views.

You can find a great and fairly complete overview of ML papers in HEP at https://iml-wg.github.io/HEPML-LivingReview/.

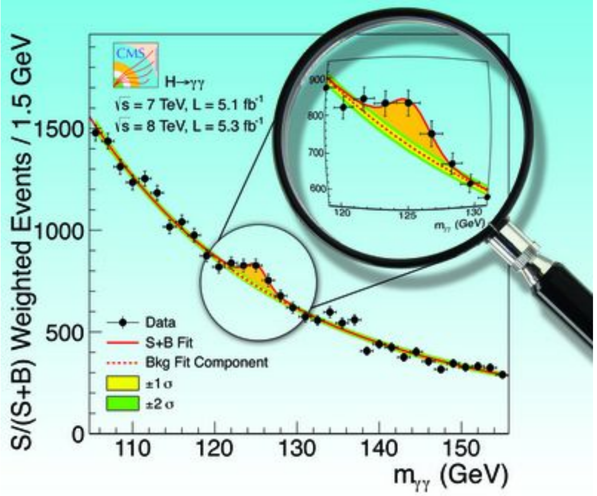
“The primary goal of the experiments at the CERN Large Hadron Collider (LHC) is to answer fundamental questions in particle physics.”

LHC schedule



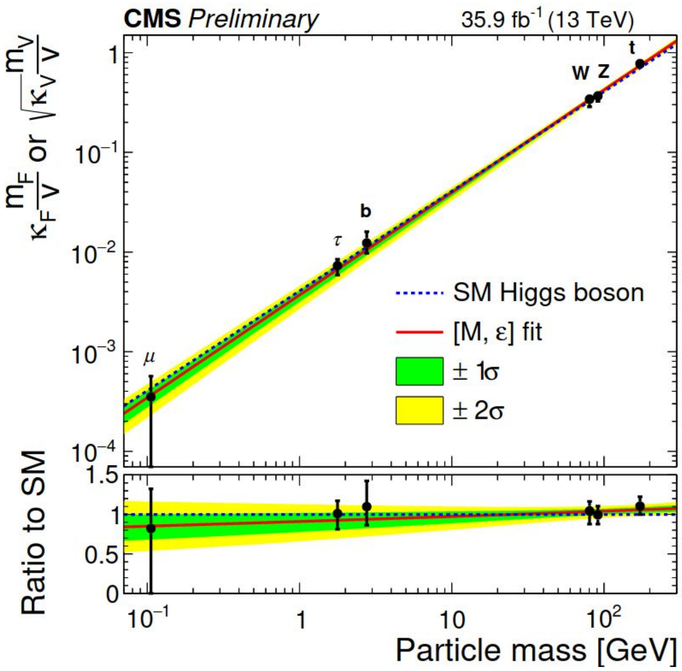
How do particles acquire mass?

Higgsdependence (2012)

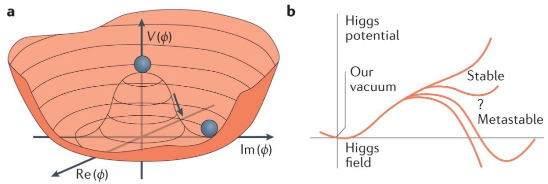


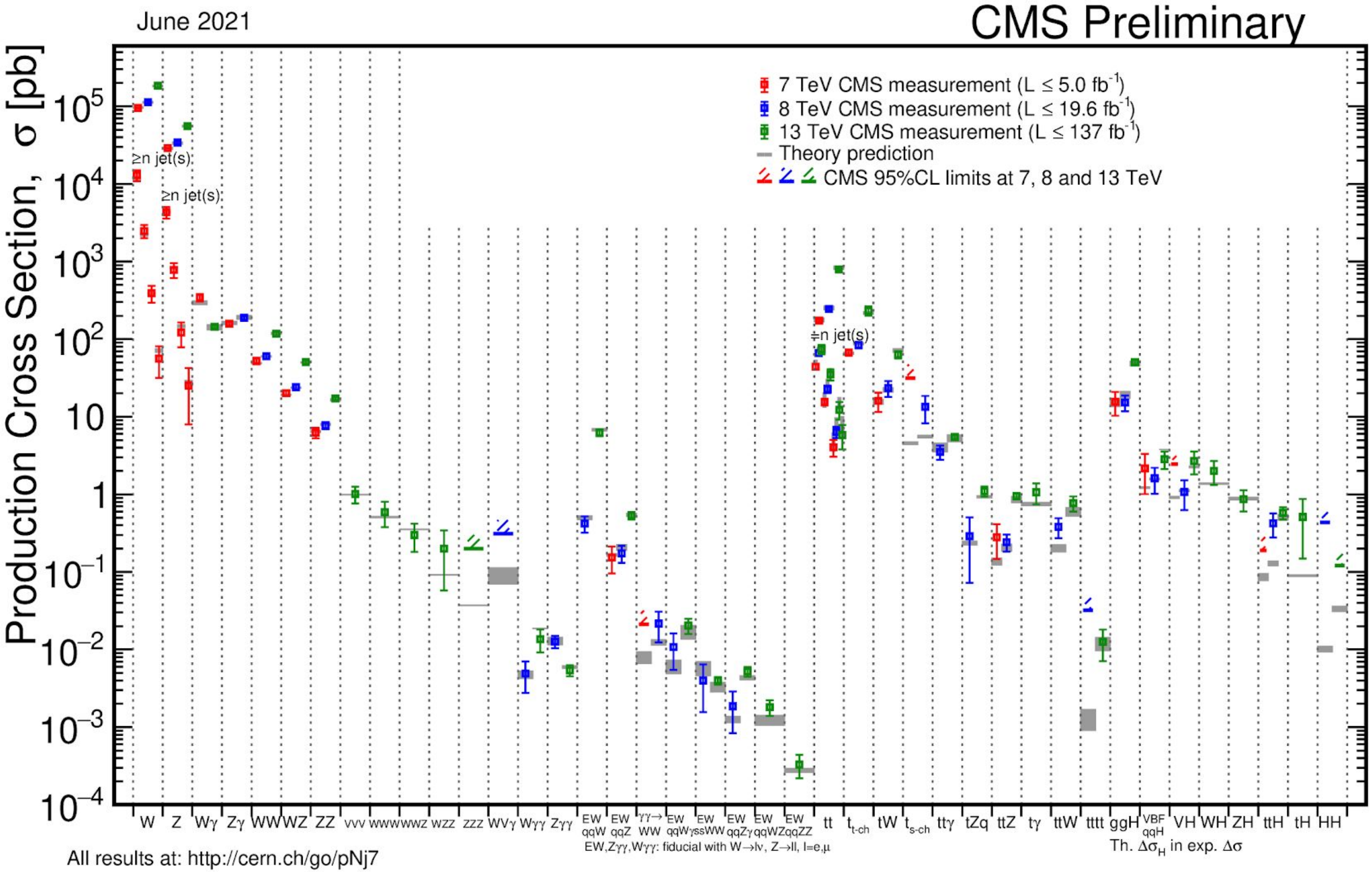
The first and so far only observed spinless elementary particle!

Higgs couplings (2019)

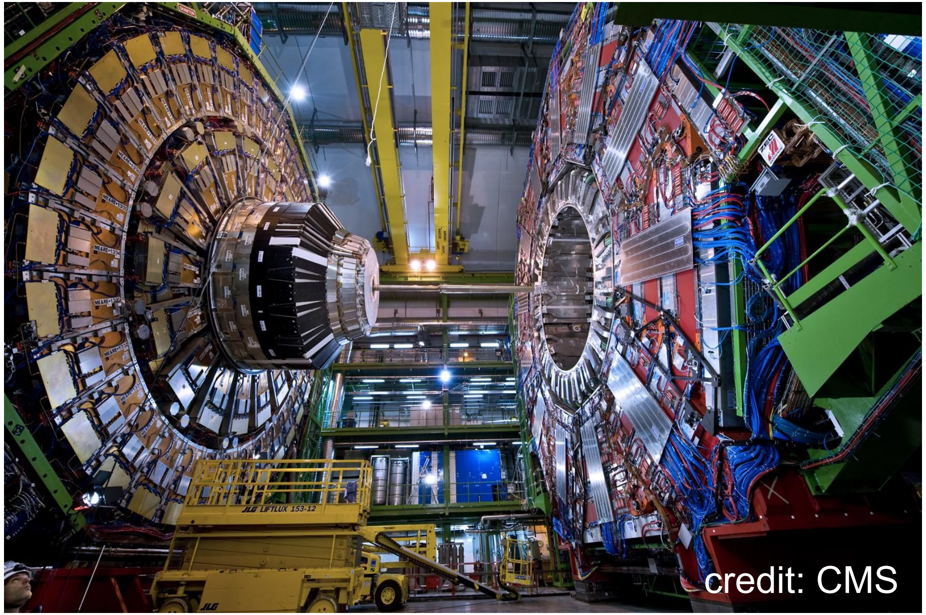


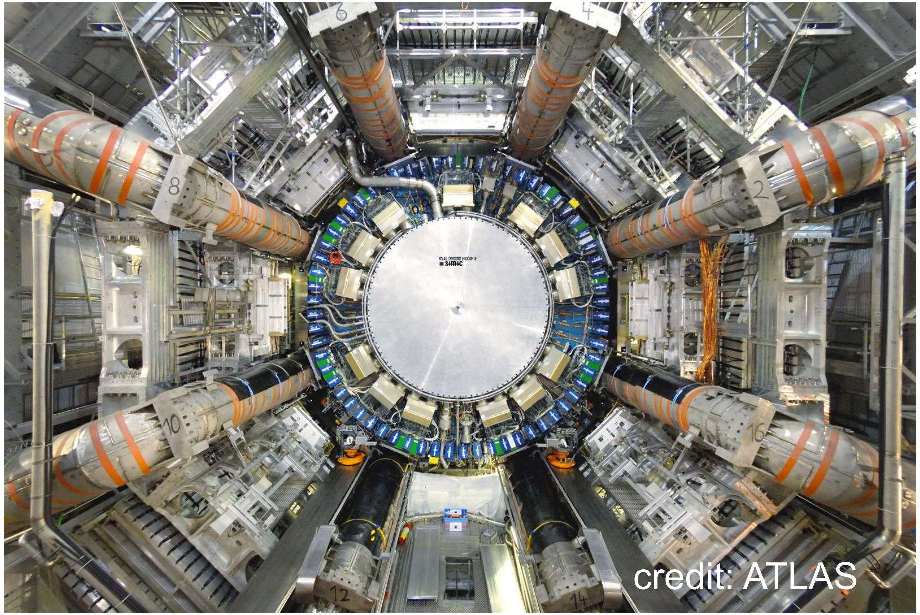
Higgs self-interaction (~2030)



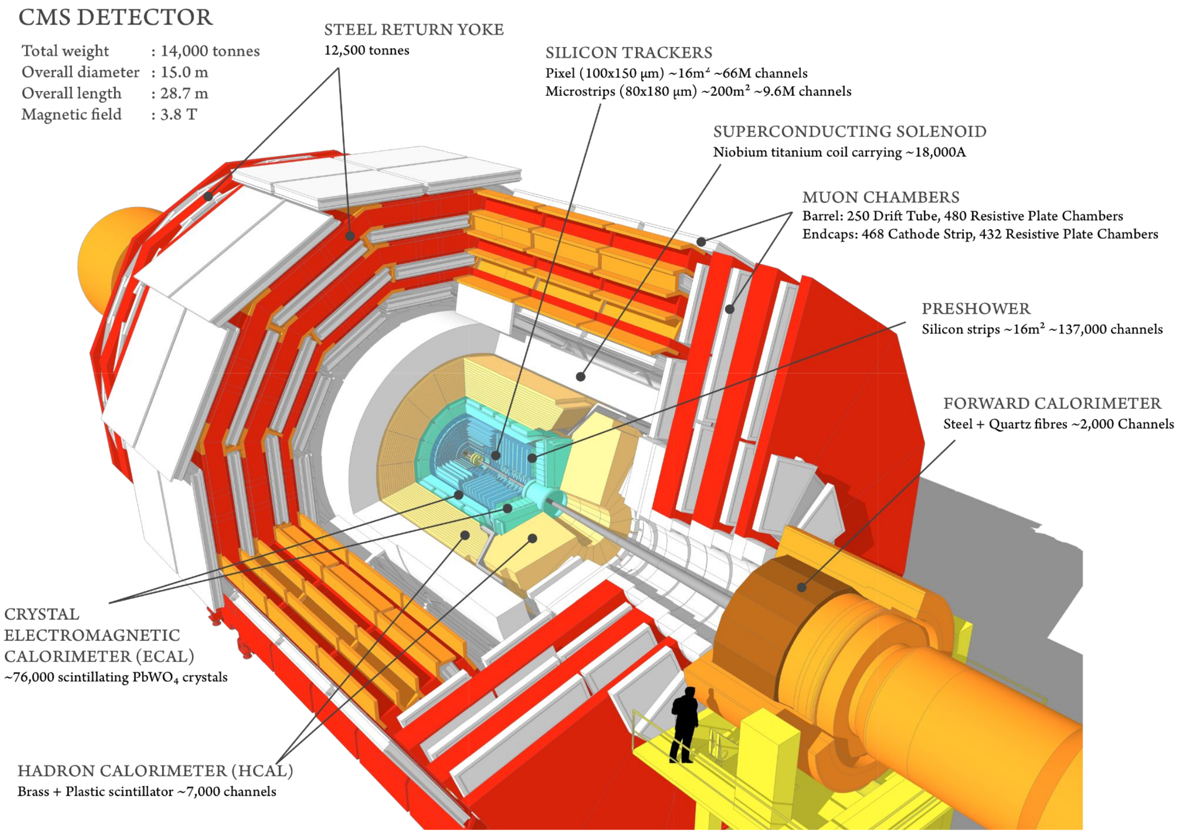

June 2021
CMS Preliminary
Production Cross Section, σ [pb]
7 TeV CMS measurement (L ≤ 5.0 fb^{-1})
8 TeV CMS measurement (L ≤ 19.6 fb^{-1})
13 TeV CMS measurement (L ≤ 137 fb^{-1})
Theory prediction
CMS 95%CL limits at 7, 8 and 13 TeV
≥n jet(s)
≥n jet(s)
=n jet(s)
W Z Wγ Zγ WW WZ ZZ VVV WWW WWZ WZZ ZZZ WVγ Wγγ Zγγ EW qqW EW qqZ γγ→WW EW qqWγ EW ssWW EW qqZγ EW qqWZ EW qqZZ tt $t_{t\text{-}ch}$ tW $t_{s\text{-}ch}$ ttγ tZq ttZ tγ ttW tttt ggH VBF qqH VH WH ZH ttH tH HH
EW,Zγγ,Wγγ: fiducial with W→lν, Z→ll, l=e,μ
Th. $\Delta\sigma_H$ in exp. Δσ
All results at: http://cern.ch/go/pNj7

General purpose detectors

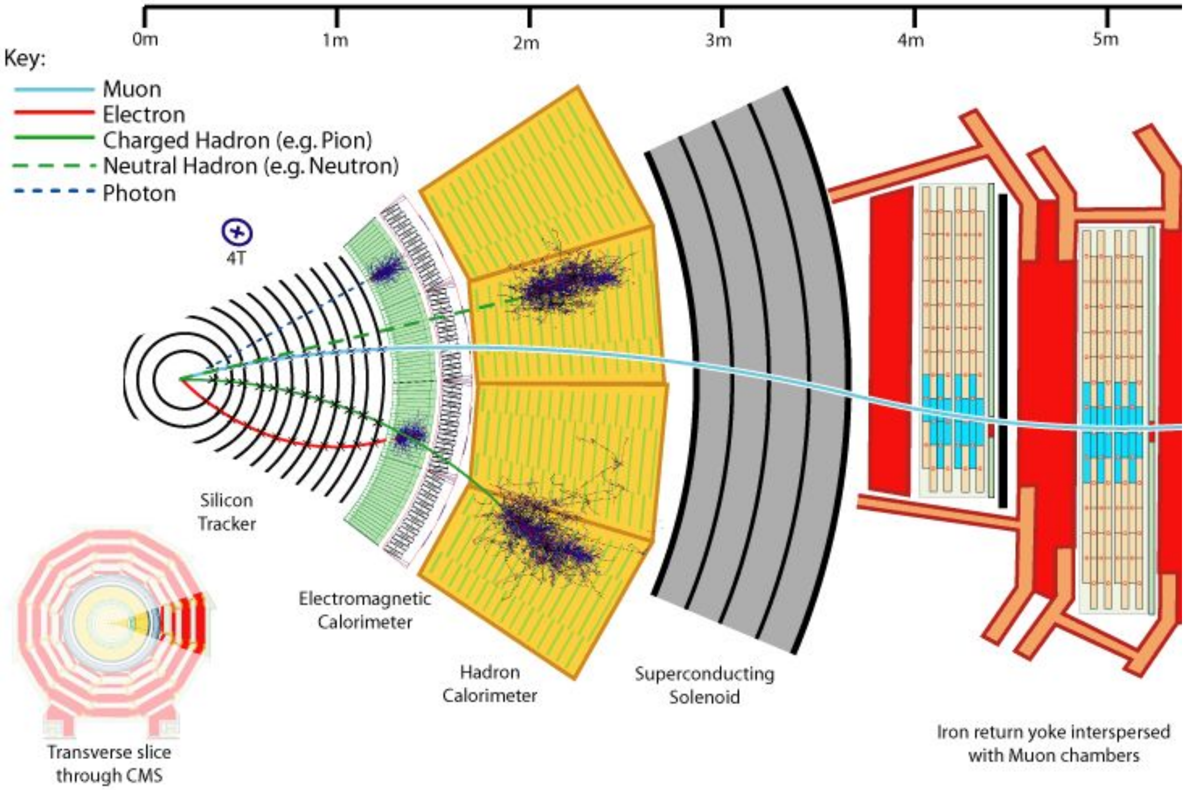




The cylindrical onion

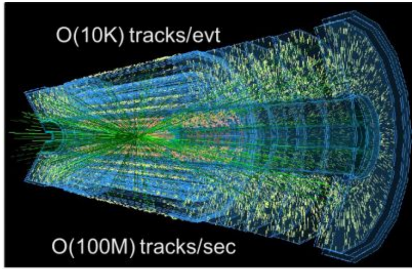


Particles in detector layers

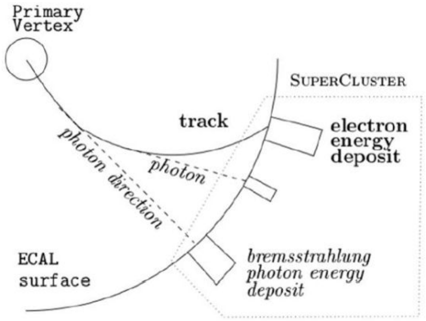


Reconstruction, simplified

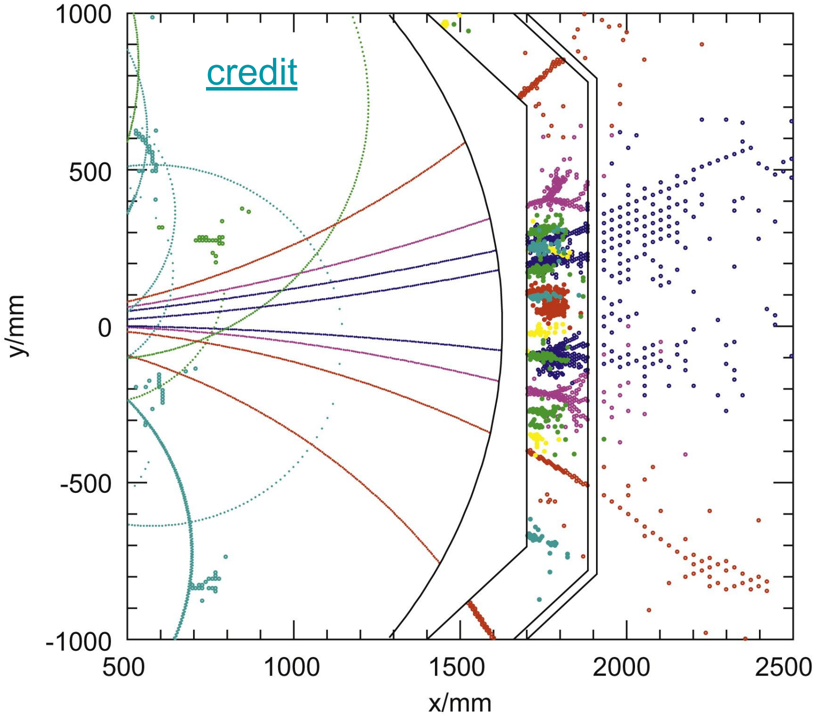
Charged particle tracking



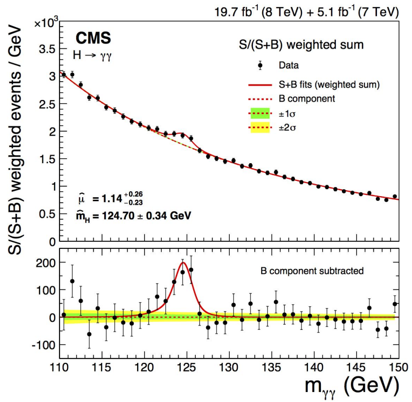
Energy clustering



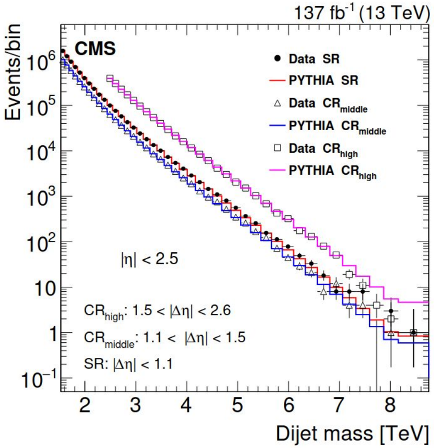
Combining detector subsystems



Photons



Jets

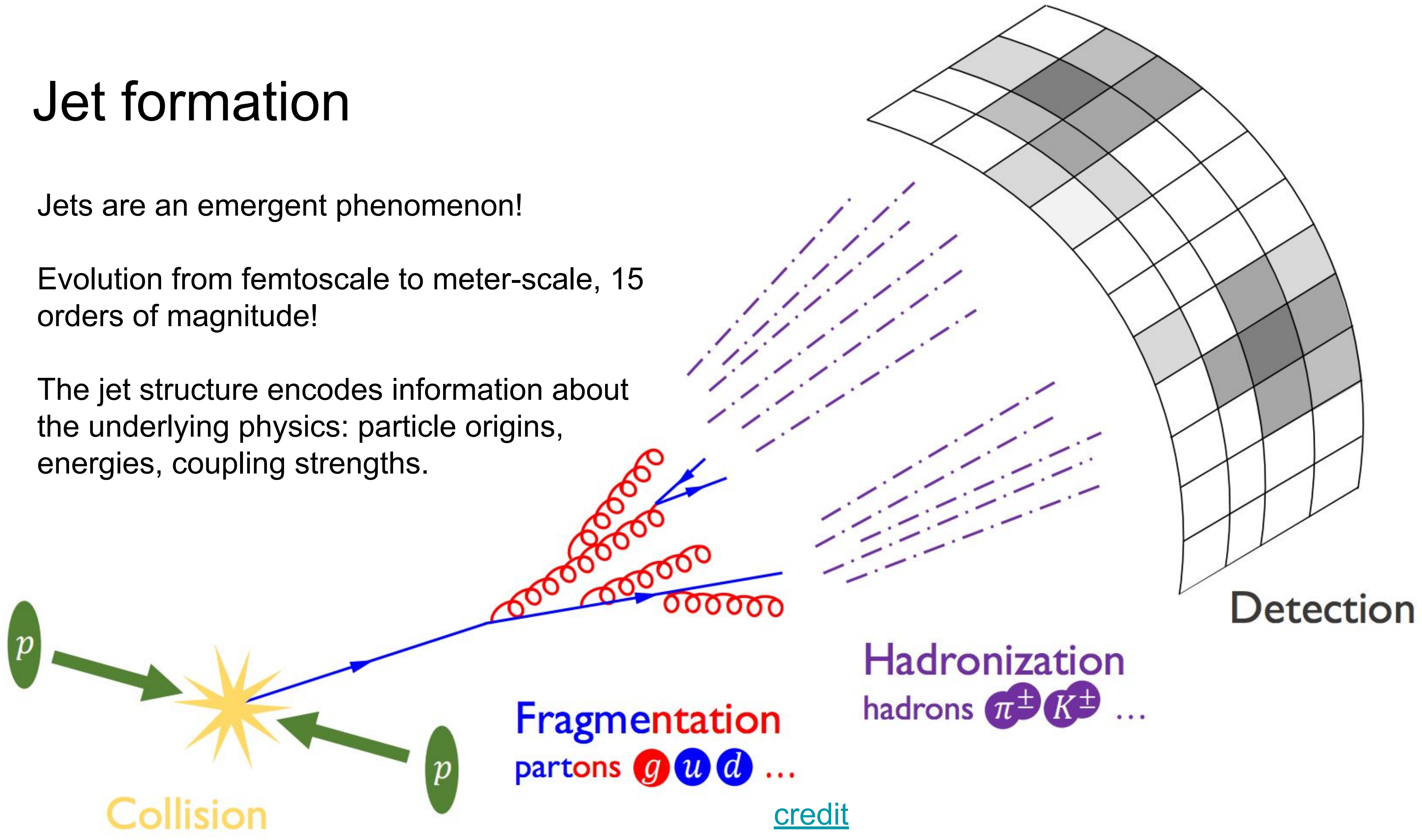


Jet formation

Jets are an emergent phenomenon!

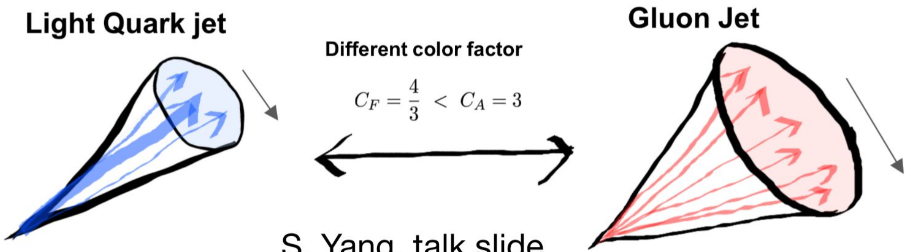
Evolution from femtoscale to meter-scale, 15 orders of magnitude!

The jet structure encodes information about the underlying physics: particle origins, energies, coupling strengths.

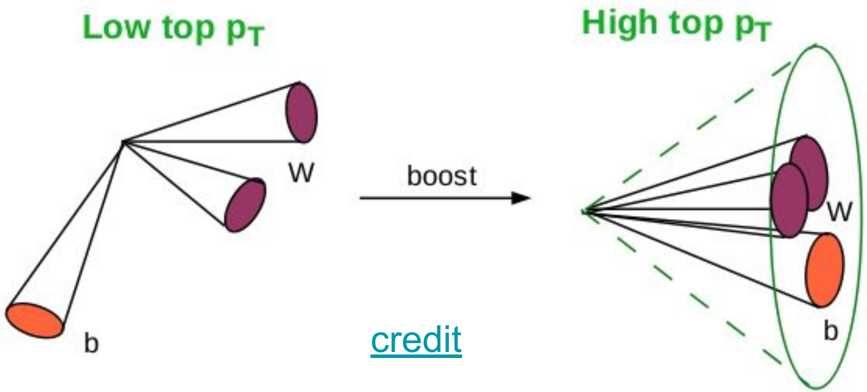


credit

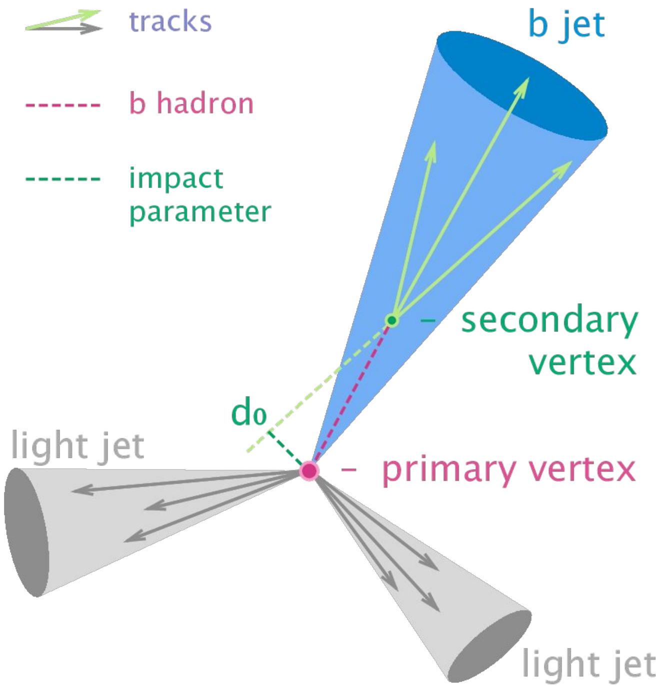
Jet origins



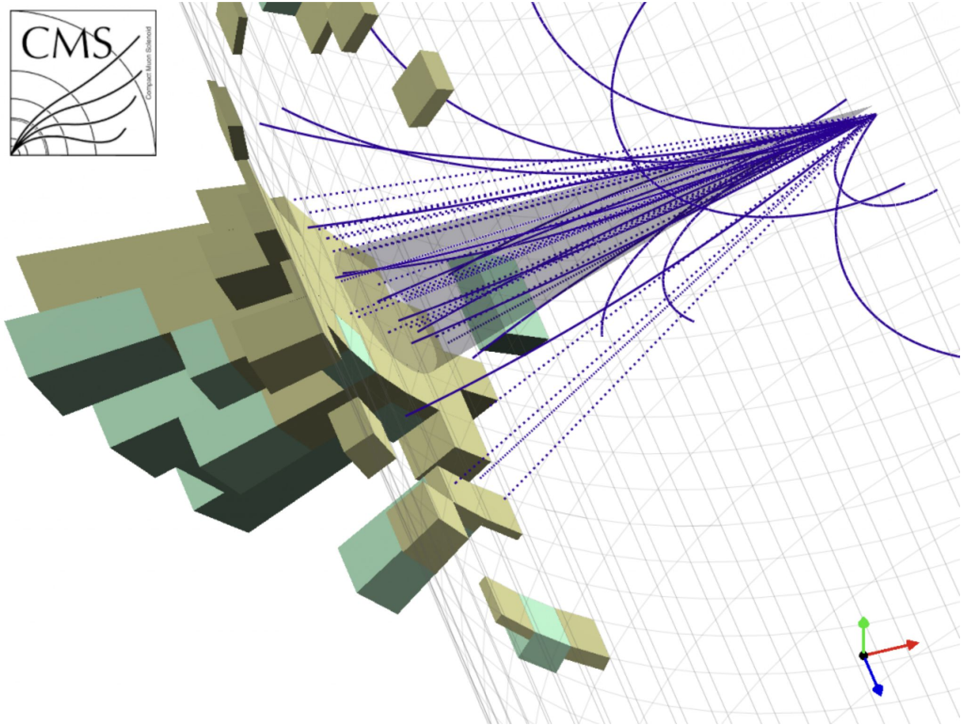
S. Yang, talk slide



[credit]

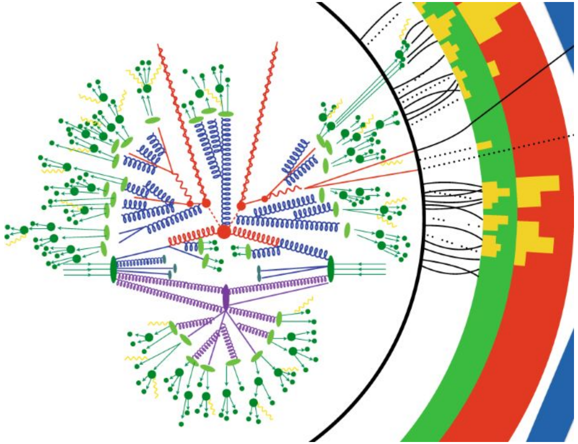


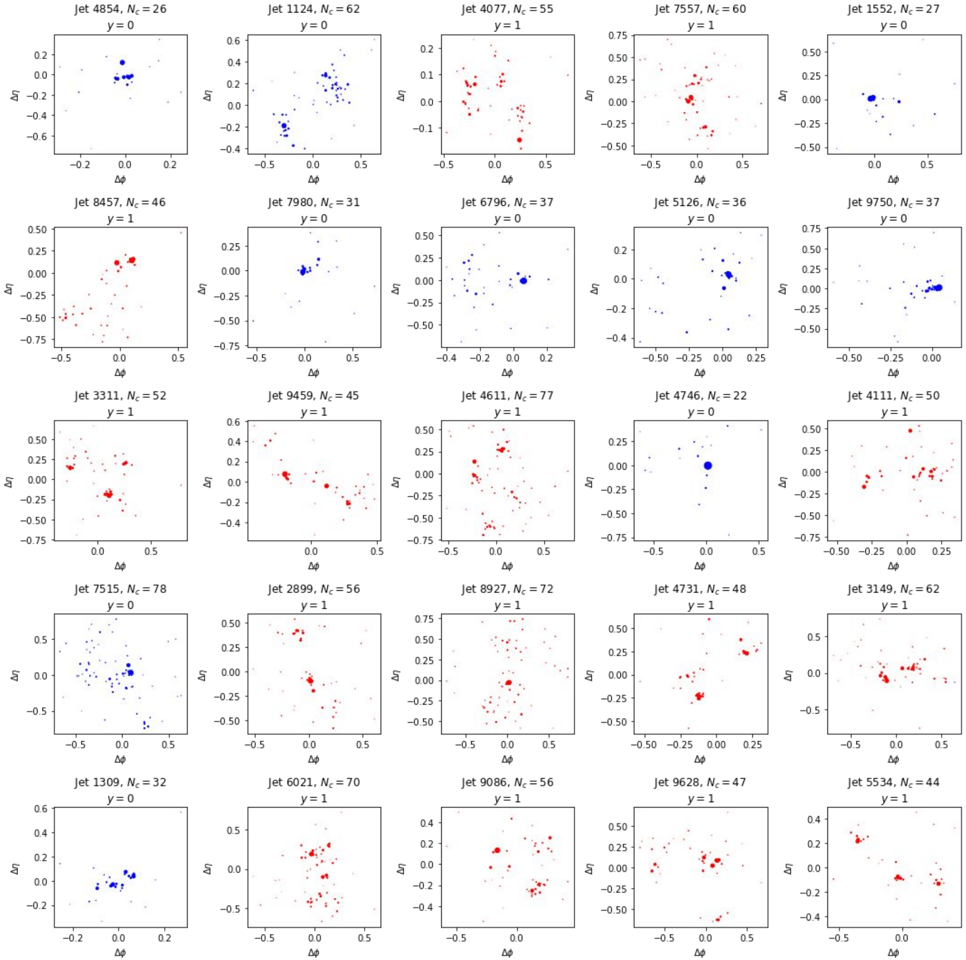
Observing jets



What is the originating particle of the jet?

Simulated datasets

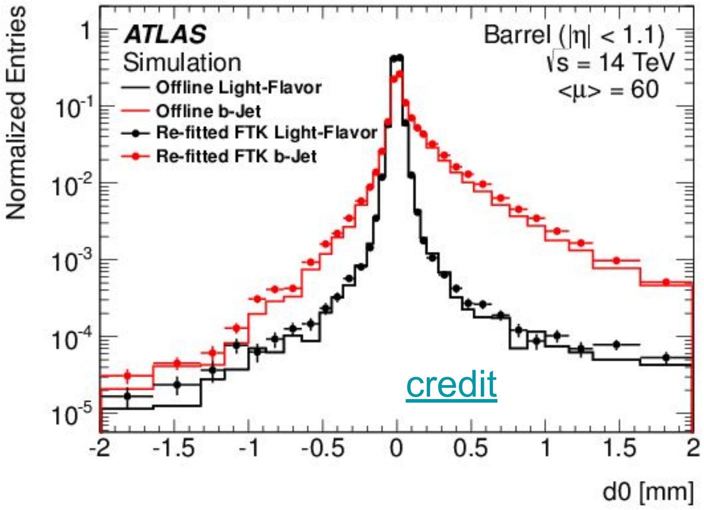




We can use the full Standard Model predictive machinery to simulate millions of examples with full detector interactions!

https://github.com/jpata/gnn-hep-lecture/blob/main/intro-to-gnns.ipynb

Discriminating with observables



credit

Construct observables based on theory or prior physics knowledge.

Things get complicated in the real world: systematics, irreducible backgrounds.

Always need to validate and calibrate on real data!

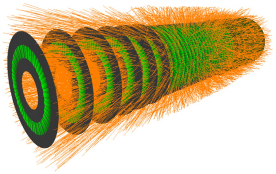
Supervised ML ≃ fitting nonlinear models on large datasets

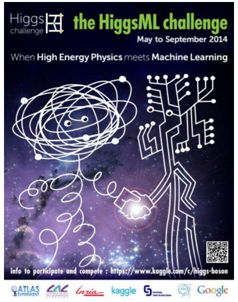
Dataset

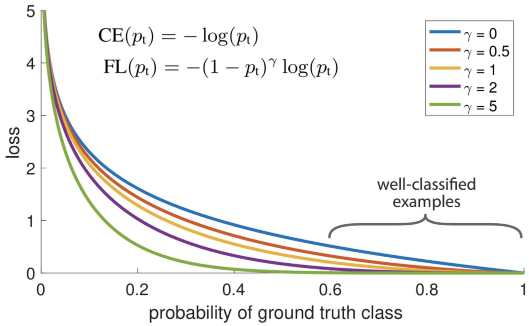
Loss

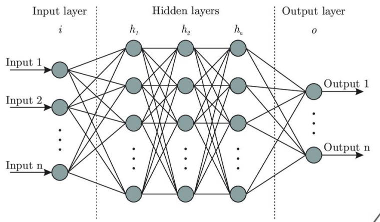
Model

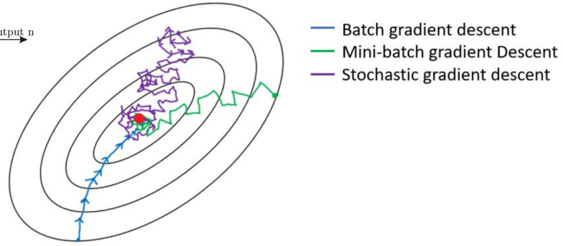
Optimizer



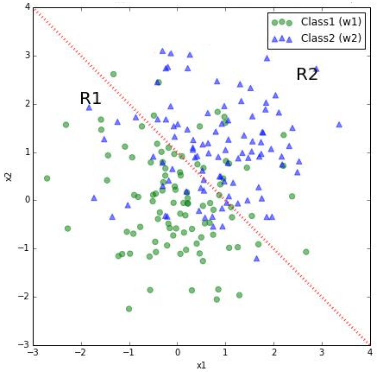


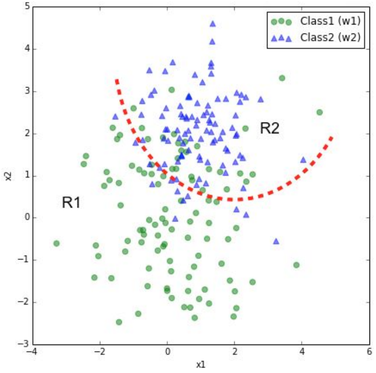






B-jet identification: multivariate classification

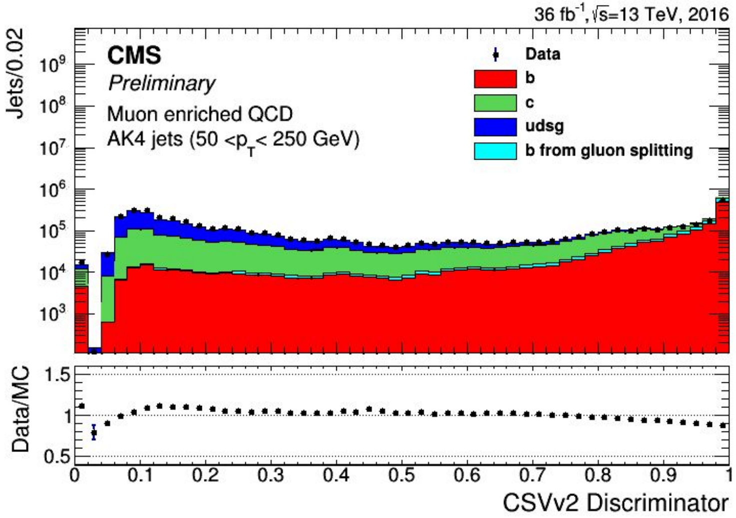




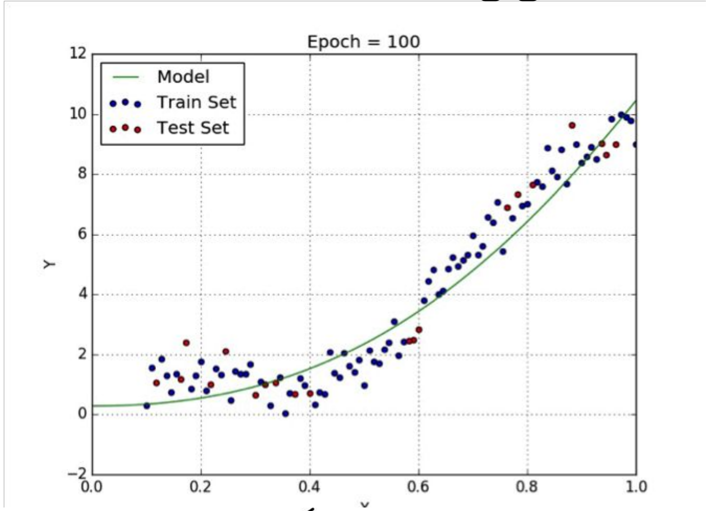
Combine individual observables nonlinearly, classify examples using a decision boundary.

Table 1: Input variables used for the Run 1 version of the CSV algorithm and for the CSVv2 algorithm. The symbol "x" ("—") means that the variable is (not) used in the algorithm

Input variable	Run 1 CSV	CSVv2
SV 2D flight distance significance	x	x
Number of SV	—	x
Track η_{rel}	x	x
Corrected SV mass	x	x
Number of tracks from SV	x	x
SV energy ratio	x	x
ΔR(SV, jet)	—	x
3D IP significance of the first four tracks	x	x
Track $p_{\mathrm{T,rel}}$	—	x
ΔR(track, jet)	—	x
Track $p_{\mathrm{T,rel}}$ ratio	—	x
Track distance	—	x
Track decay length	—	x
Summed tracks E_{T} ratio	—	x
ΔR(summed tracks, jet)	—	x
First track 2D IP significance above c threshold	—	x
Number of selected tracks	—	x
Jet p_{T}	—	x
Jet η	—	x

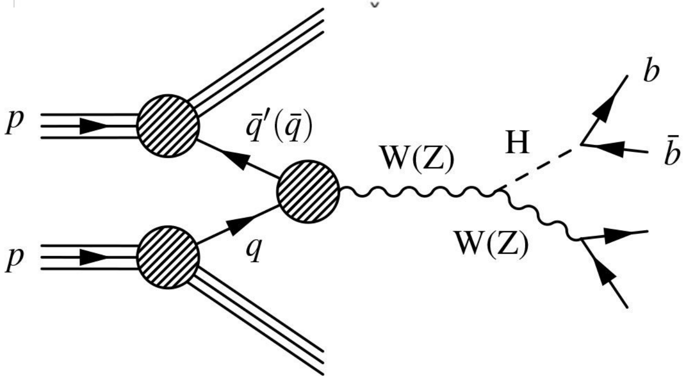


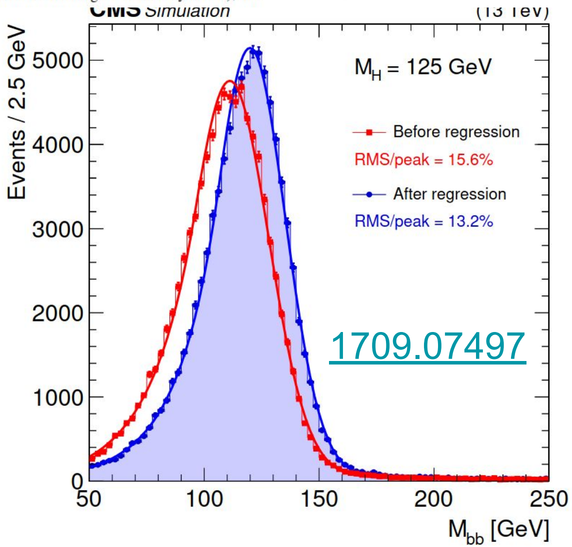
Observation of Higgs to b quarks: energy regression



- jet kinematics: jet p_T, η, mass, and transverse mass, defined as $\sqrt{E^2 - p_z^2}$;
- information about pileup interactions: the median energy density in the event, ρ, corresponding to the amount of transverse momentum per unit area that is due to overlapping collisions[35];
- information about semileptonic decays of b hadrons when an electron or muon candidate is clustered within a jet: the transverse component of lepton momentum perpendicular to the jet axis, the distance $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$, and a categorical variable that encodes information about the lepton candidate's flavor;
- information about the secondary vertex, selected as the highest p_T displaced vertex linked to the jet: number of tracks associated to the vertex, transverse momentum, and mass (computed assigning the pion mass to all reconstructed tracks forming the secondary vertex); the distance between the collision vertex and the secondary vertex computed in three-dimensional space with its associated uncertainty[36, 37];
- jet composition: largest p_T value of any charged hadron candidates, fractions of energy carried by jet constituents; namely charged hadrons, neutral hadrons, muons, and an electromagnetic component coming from electrons and photons. These fractions are computed for the whole jet, and separately in five rings of ΔR around the jet axis (ΔR = 0–0.05, 0.05–0.1, 0.1–0.2, 0.2–0.3, 0.3–0.4);
- multiplicity of PF candidates clustered to form the jet;
- information about jet energy sharing among the jet constituents computed as

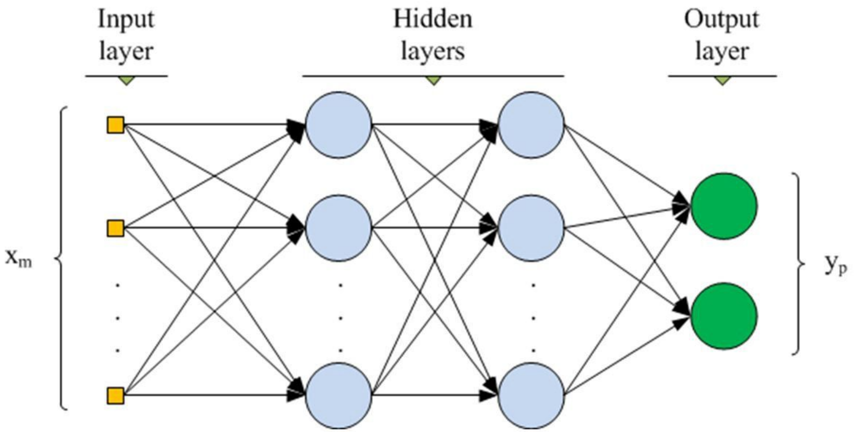
$$\frac{\sqrt{\sum_i p_{T,i}^2}}{\sum_i p_{T,i}}, \quad (1)$$

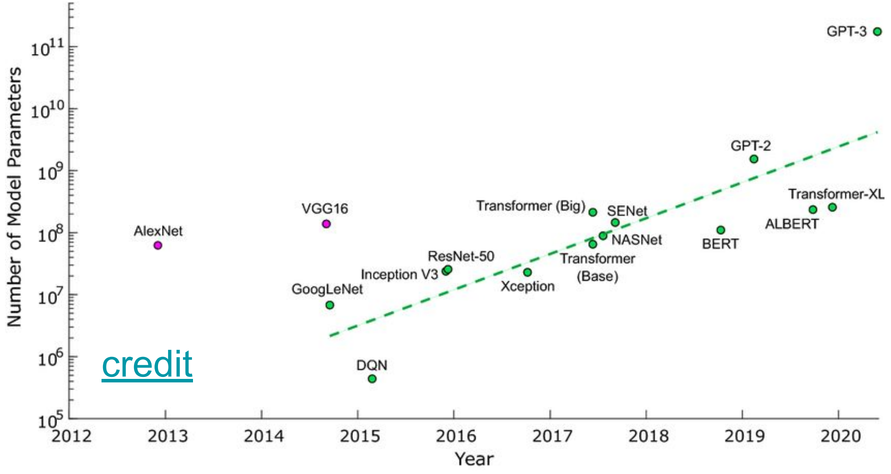




1709.07497

Neural networks





credit

Multivariate, nonlinear, highly over-parameterized functions.

Neural network models for various data types

Feedforward networks ~ simple feature vectors

Convolutional neural networks ~ 2D/3D images

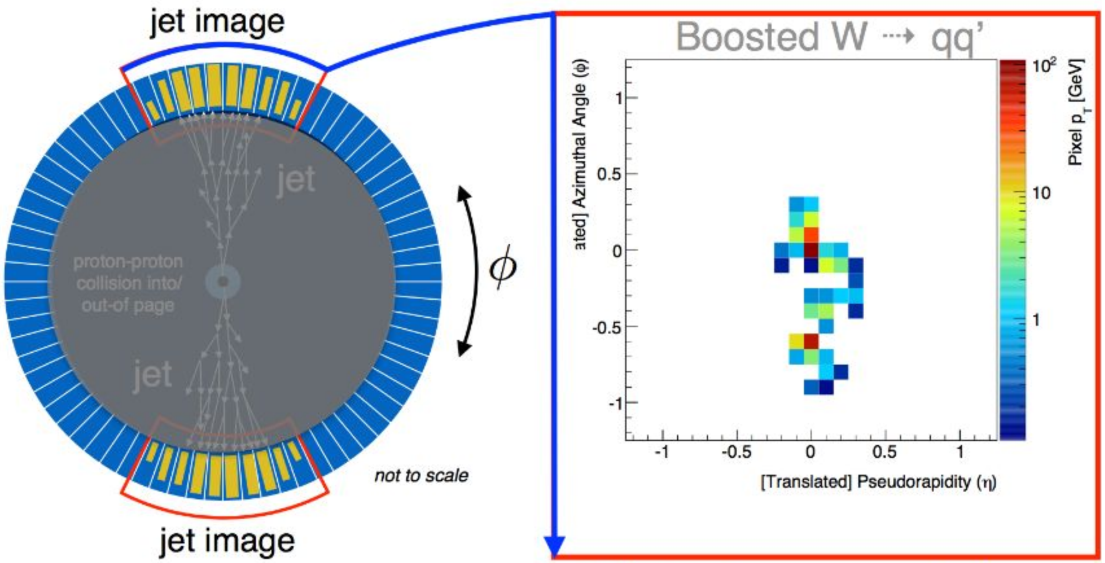
Recurrent neural networks / LSTM ~ ordered sequences

Graph neural networks ~ sets, graphs

Transformers ~ ordered sequences, sets

Jets as images

Unrolled slice of detector

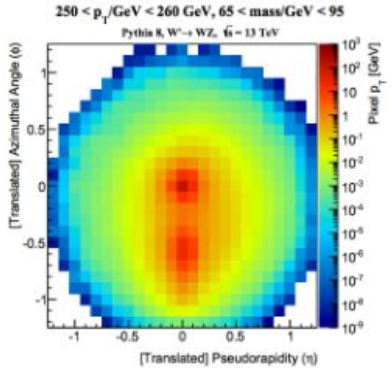


credit

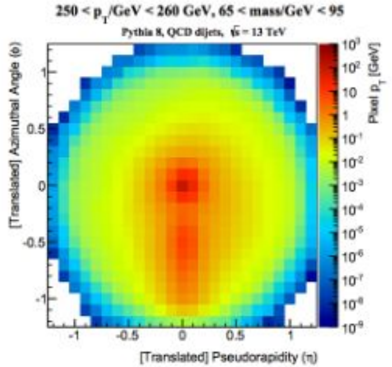
Calorimeter towers as pixels

Energy depositions as intensity

W-jets

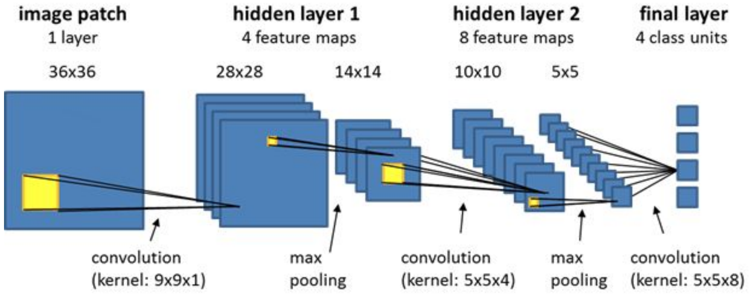


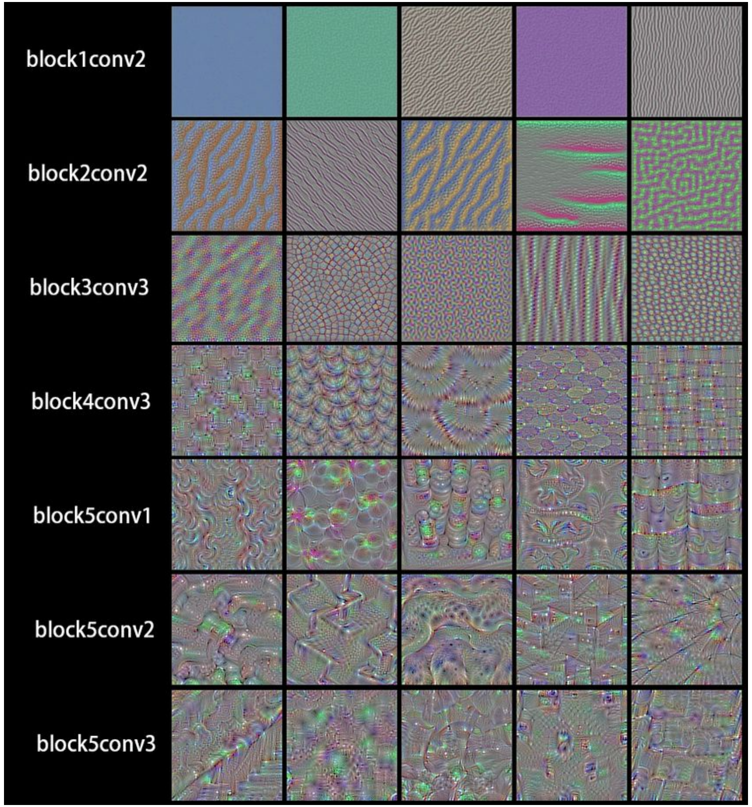
QCD-jets



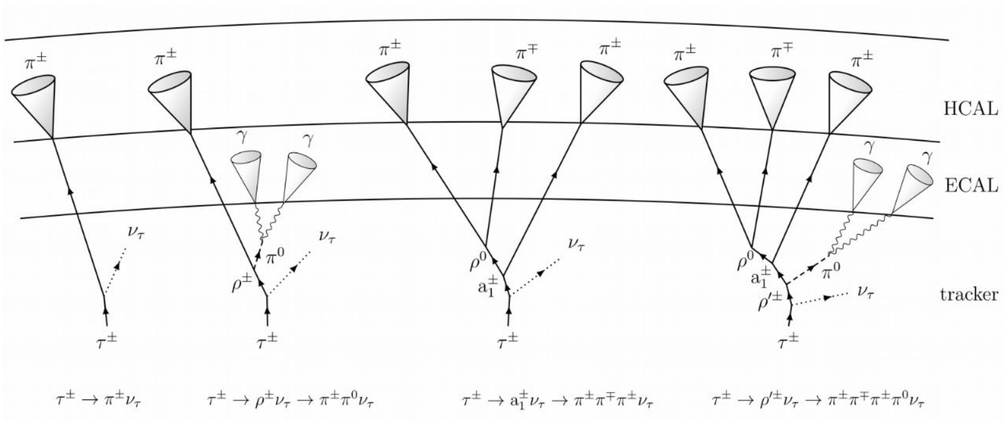
Encoding symmetries

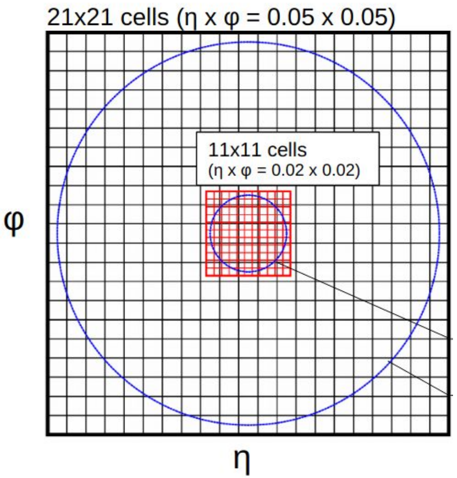
Process the image with learnable translation-invariant filters: Convolutional Neural Networks

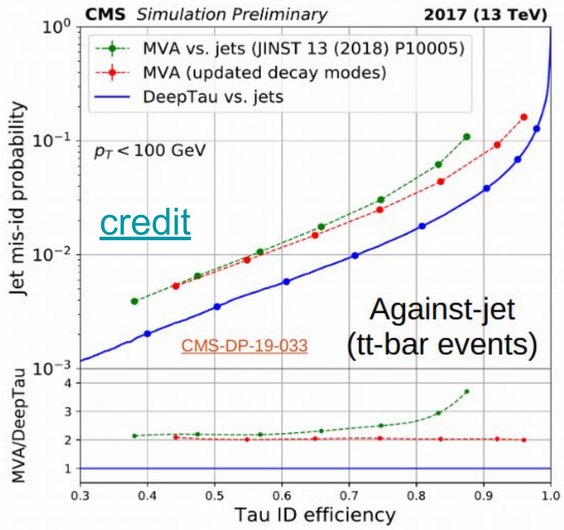




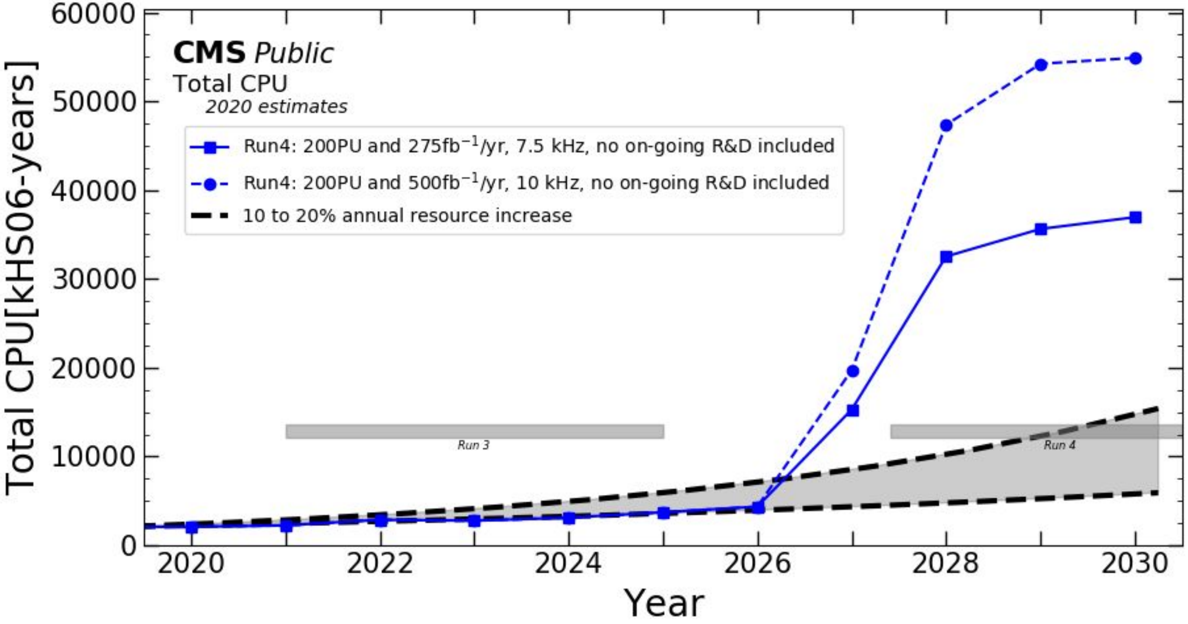
Tau identification with CNNs





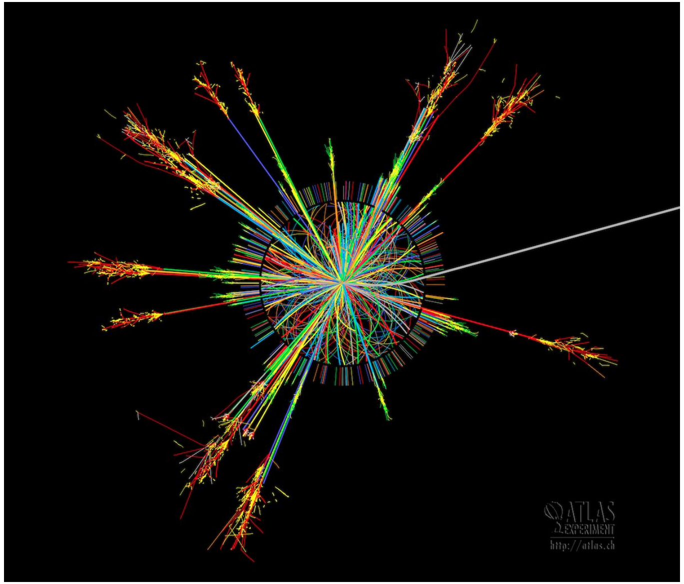


The computing challenge



LHC is compute-limited, we fight for every CPU cycle and kilobyte.

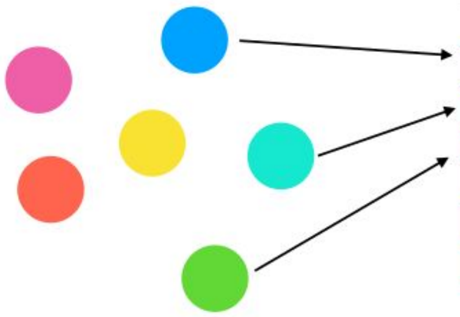
Sets of feature vectors



set of inputs with N constituents, M features

{..., (pT, η, ϕ, particle ID), ...}

feature matrix (N, M)

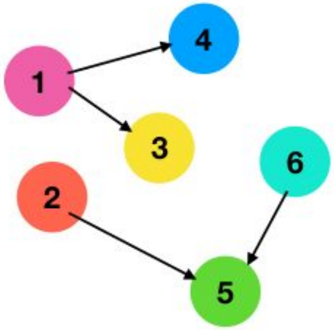


pT (GeV)	η	ϕ	particle ID
12.3	*1.2*	*0.5*	*pi+*
11.8	*1.24*	*0.45*	*K0*
10.4	*1.18*	*0.43*	*pi-*
9.8	*1.39*	...	*e-*
6.4	...	...	...
5.3	...	...	

jet constituents

Set to graph

graph = set of nodes/vertices/elements + edges between them

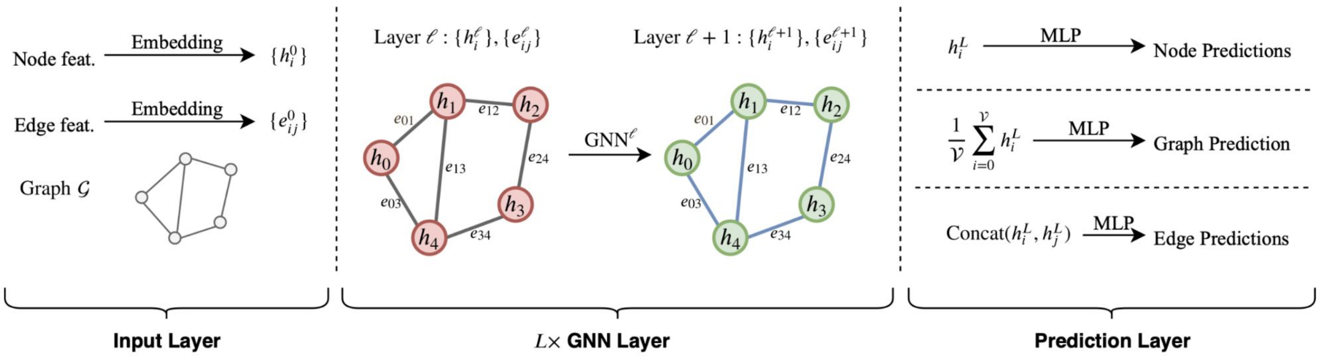


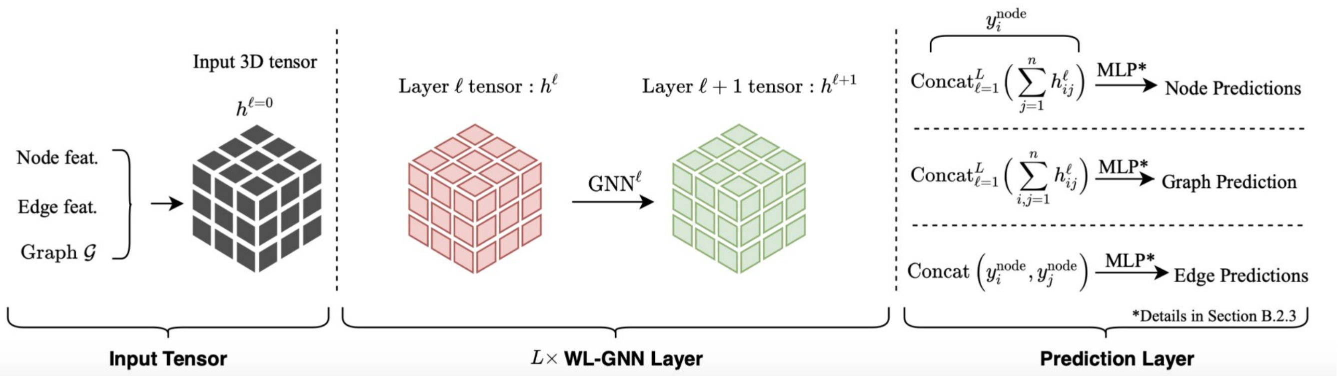
Edges represented as a index pairs
edges = [(1,4), (1,3), (2,5), (6,5)]

Or as an NxN adjacency matrix
(possibly sparse)

$$A_{ij} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Neural nets on graphs



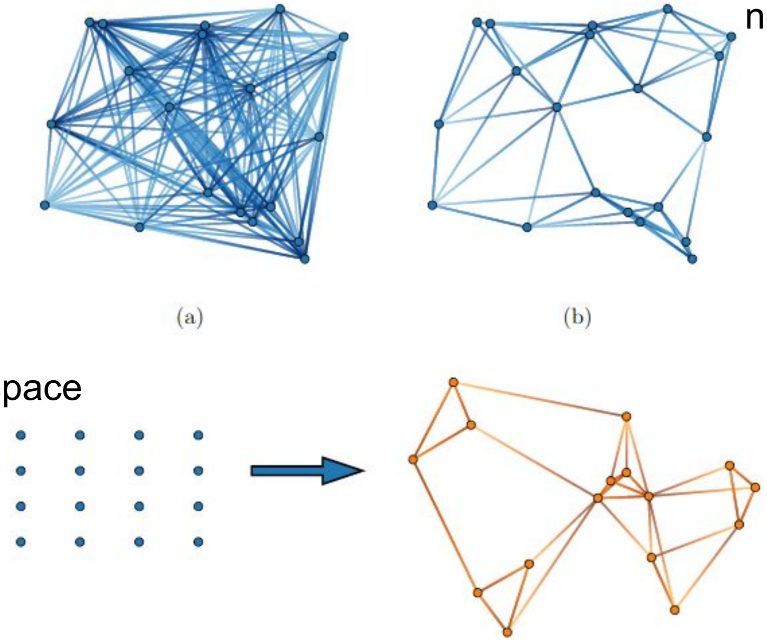


Constructing graphs

All-to-all

Predefined neighborhood

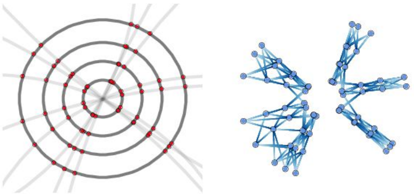
Learned feature space



(a) (b) (c)

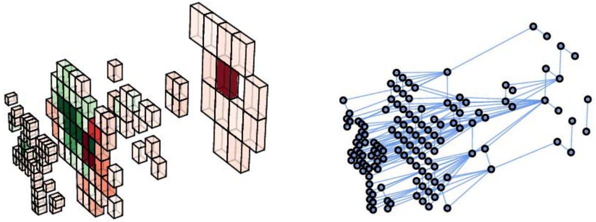
Physics data as graphs

Particle tracking (neighborhood)



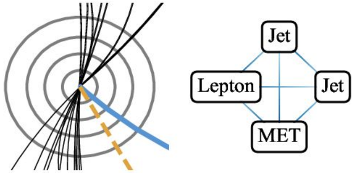
(a)

Calorimeter clustering (learned)



(b)

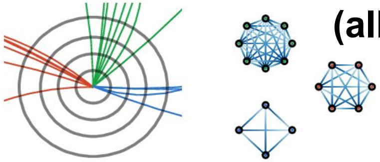
Event identification (all-to-all)



(c)

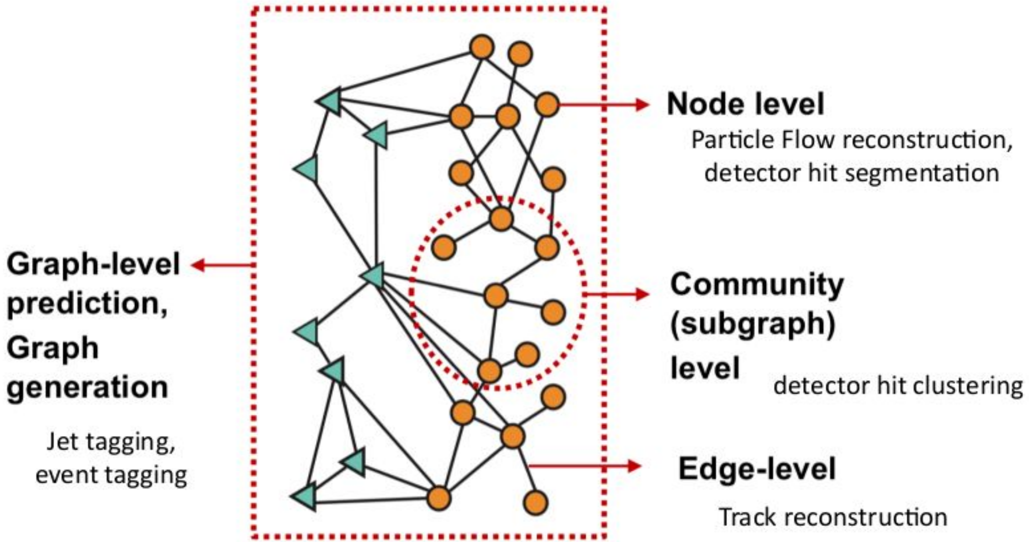
[2007.13681](https://arxiv.org/abs/2007.13681)

Jet constituents (all-to-all)



(d)

Graph nets in a nutshell



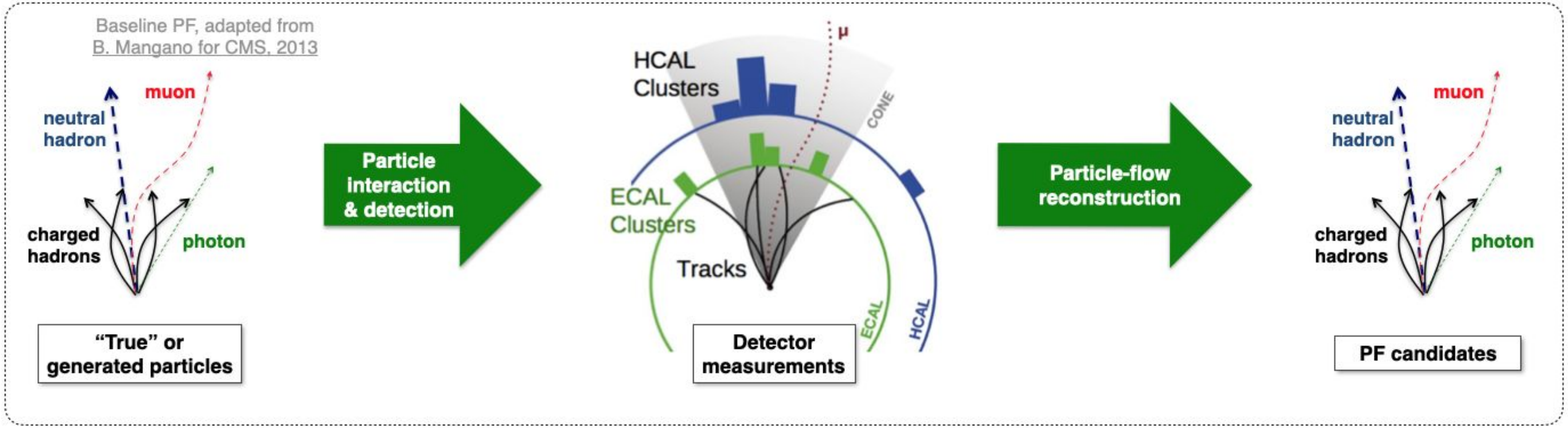
J. Leskovec et al [2021]

A concrete case: machine learned particle flow reconstruction

- Graphs
  - Neural Message Passing for Jet Physics
  - Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors
  - Probing stop pair production at the LHC with graph neural networks [DOI]
  - Pileup mitigation at the Large Hadron Collider with graph neural networks [DOI]
  - Unveiling CP property of top-Higgs coupling with graph neural networks at the LHC [DOI]
  - JEDI-net: a jet identification algorithm based on interaction networks [DOI]
  - Learning representations of irregular particle-detector geometry with distance-weighted graph networks [DOI]
  - Interpretable deep learning for two-prong jet classification with jet spectra [DOI]
  - Neural Network-based Top Tagger with Two-Point Energy Correlations and Geometry of Soft Emissions [DOI]
  - Probing triple Higgs coupling with machine learning at the LHC
  - Casting a graph net to catch dark showers [DOI]
  - Graph neural networks in particle physics [DOI]
  - Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics [DOI]
  - Supervised Jet Clustering with Graph Neural Networks for Lorentz Boosted Bosons [DOI]
  - Track Seeding and Labelling with Embedded-space Graph Neural Networks
  - Graph neural network for 3D classification of ambiguities and optical crosstalk in scintillator-based neutrino detectors [DOI]
  - The Boosted Higgs Jet Reconstruction via Graph Neural Network
  - Accelerated Charged Particle Tracking with Graph Neural Networks on FPGAs
  - Particle Track Reconstruction using Geometric Deep Learning
  - Jet tagging in the Lund plane with graph networks [DOI]
  - Vertex and Energy Reconstruction in JUNO with Machine Learning Methods
  - MLPF: Efficient machine-learned particle-flow reconstruction using graph neural networks
  - Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC
  - Deep Learning strategies for ProtoDUNE raw data denoising
  - Graph Neural Network for Object Reconstruction in Liquid Argon Time Projection Chambers
  - Instance Segmentation GNNs for One-Shot Conformal Tracking at the LHC
  - Charged particle tracking via edge-classifying interaction networks
  - Jet characterization in Heavy Ion Collisions by QCD-Aware Graph Neural Networks
  - Graph Generative Models for Fast Detector Simulations in High Energy Physics
  - Segmentation of EM showers for neutrino experiments with deep graph neural networks
  - Anomaly detection with Convolutional Graph Neural Networks
  - Energy-weighted Message Passing: an infra-red and collinear safe graph neural network algorithm
  - Improved Constraints on Effective Top Quark Interactions using Edge Convolution Networks
  - Particle Graph Autoencoders and Differentiable, Learned Energy Mover's Distance
- Sets (point clouds)
  - Energy Flow Networks: Deep Sets for Particle Jets [DOI]
  - ParticleNet: Jet Tagging via Particle Clouds [DOI]
  - ABCNet: An attention-based method for particle tagging [DOI]
  - Secondary Vertex Finding in Jets with Neural Networks
  - Equivariant Energy Flow Networks for Jet Tagging
  - Permutationless Many-Jet Event Reconstruction with Symmetry Preserving Attention Networks
  - Zero-Permutation Jet-Parton Assignment using a Self-Attention Network
  - Learning to Isolate Muons
  - Point Cloud Transformers applied to Collider Physics
  - SPANet: Generalized Permutationless Set Assignment for Particle Physics using Symmetry Preserving Attention
  - Particle Convolution for High Energy Physics
  - Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS

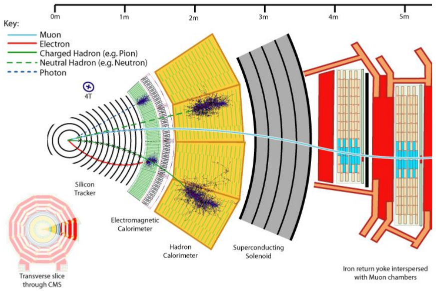
https://iml-wg.github.io/HEPML-LivingReview/

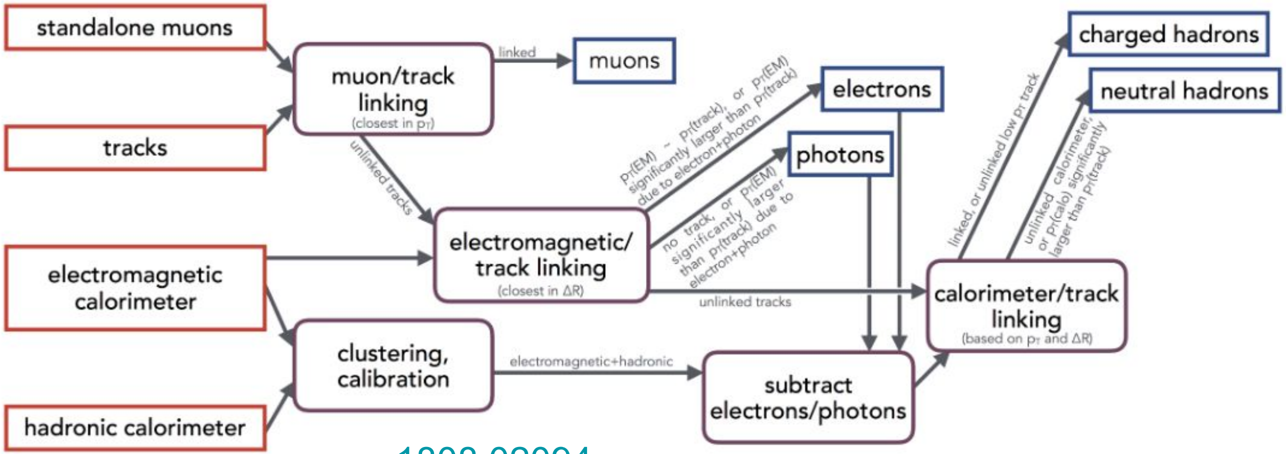
Particle reconstruction



The particle flow algorithm aims to identify and reconstruct individually all of the particles produced in a collision, through an optimal combination of the information from the entire detector.

Particle Flow algorithm





1808.02094

Figure 2. Schematic of particle flow algorithm for CMS Level-1 trigger correlator.

Particle-flow reconstruction and global event description with the CMS detector #1

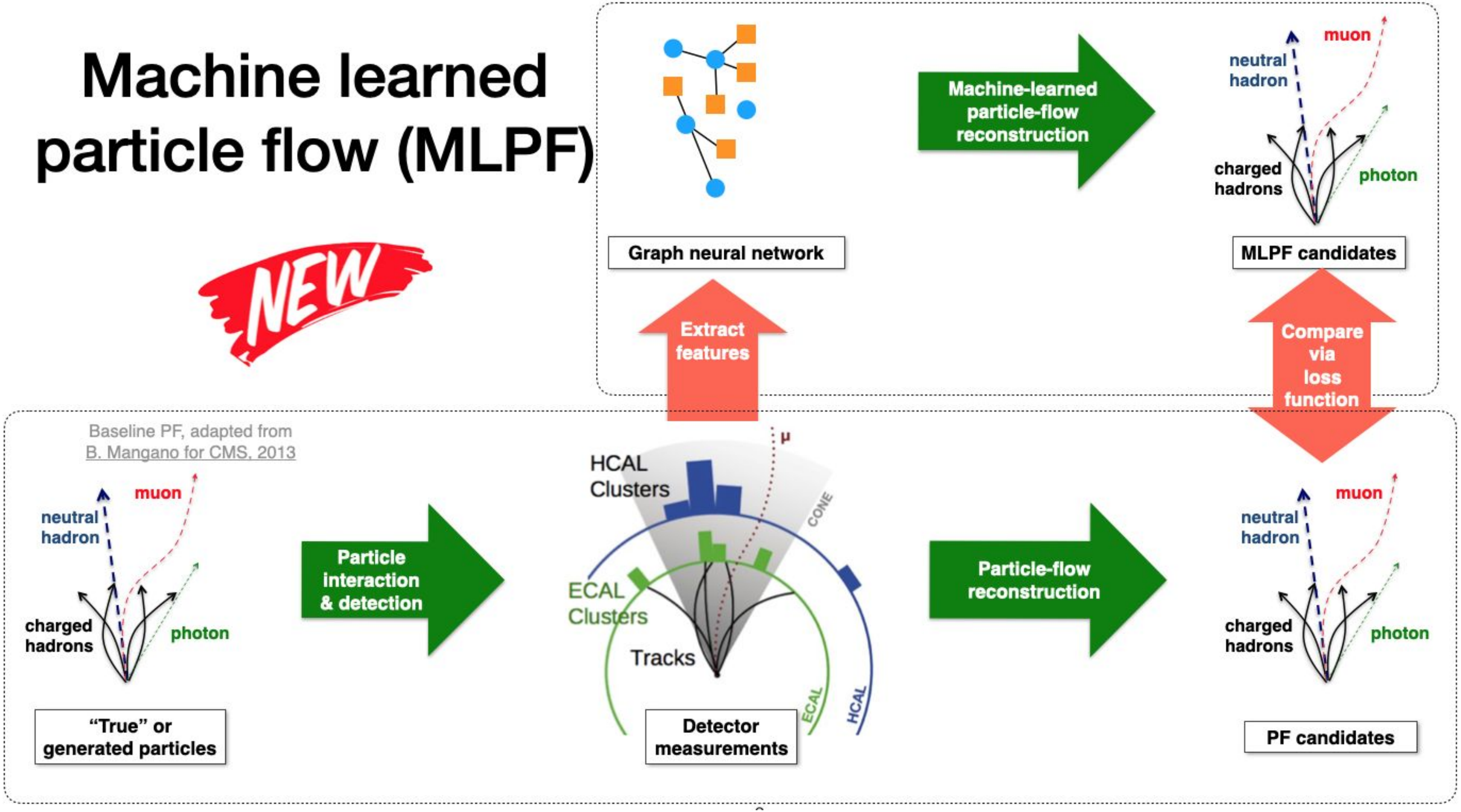
CMS Collaboration • A.M. Sirunyan et al. (Jun 15, 2017)

Published in: *JINST* 12 (2017) 10, P10003 • e-Print: 1706.04965 [physics.ins-det]

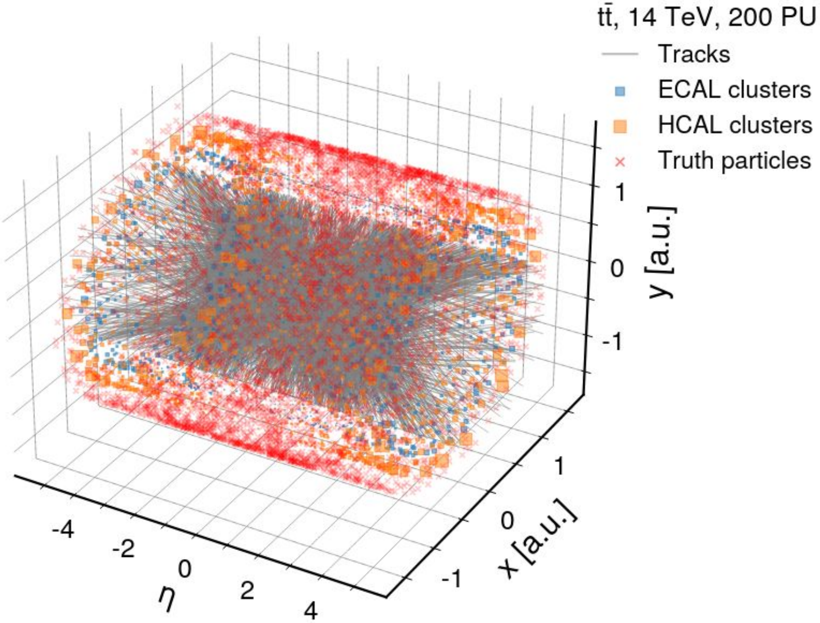
pdf links DOI cite 1,135 citations

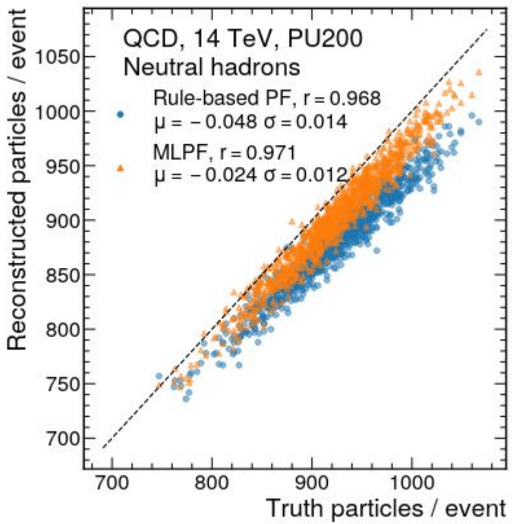
Machine learned particle flow (MLPF)

NEW



Machine learned particle flow reconstruction





“Efficient machine-learned particle-flow reconstruction using graph neural networks”; **JP**, J. Duarte, J-R Vlimant, M. Pierini, M. Spiropulu; Eur. Phys. J. C (2021) 81: 381

Open benchmark dataset for particle flow reconstruction

February 24, 2021

Dataset Open Access

Simulated particle-level events of ttbar and QCD with PU200 using Pythia8+Delphes3 for machine learned particle flow (MLPF)

Pata, Joosep; Duarte, Javier Mauricio; Vlimant, Jean-Roch; Pierini, Maurizio; Spiropulu, Maria

374 views

18,349 downloads

See more details...

Publication date:
February 24, 2021

DOI:
DOI 10.5281/zenodo.4559324

Keyword(s):
particle physics | high-energy physics | machine learning

Communities:
Machine Learning for Particle Physics

License (for files):
Creative Commons Attribution 4.0 International

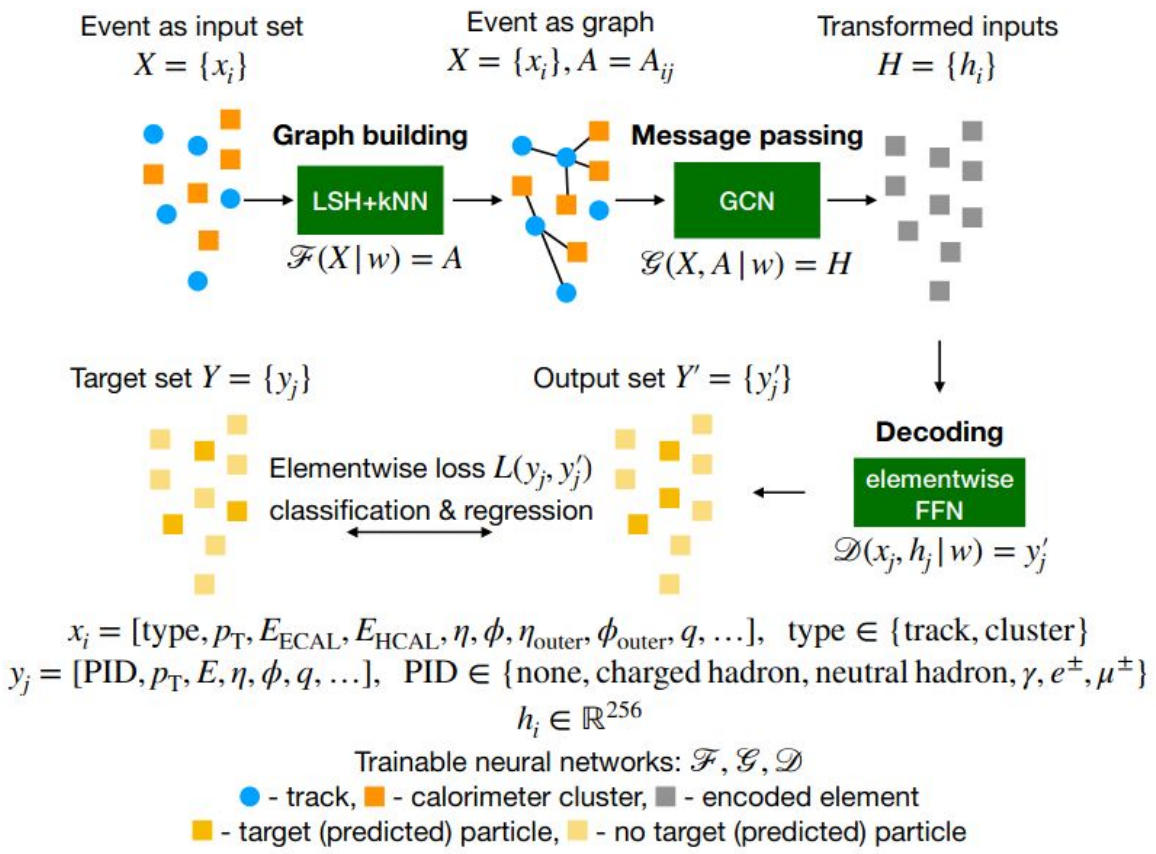
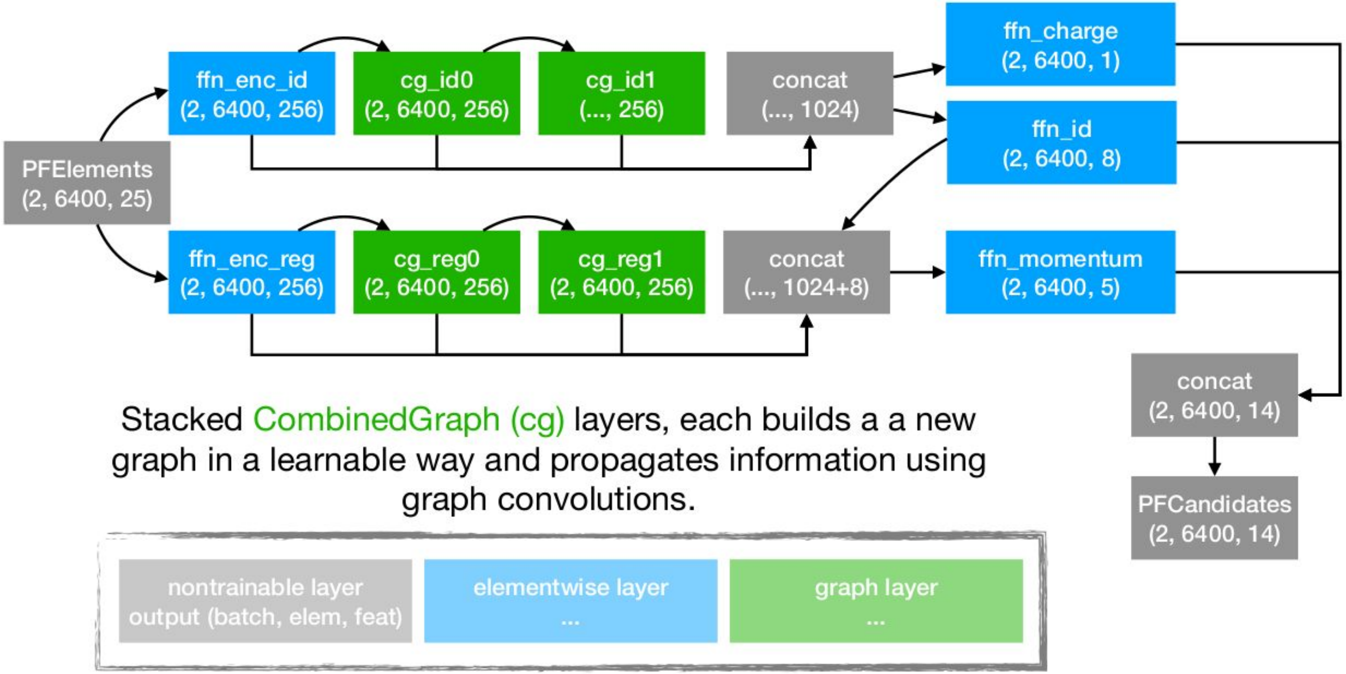


Fig. 3 Functional overview of the end-to-end trainable MLPF setup with GNNs. The event is represented as a set of detector elements x_i. The set is transformed into a graph by the graph building step, which is implemented here using an locality sensitive hashing (LSH) approximation of kNN. The graph nodes are then encoded using a message passing step, implemented using graph convolutional nets. The encoded elements are decoded to the output feature vectors y_j using elementwise feedforward networks.

Model implementation

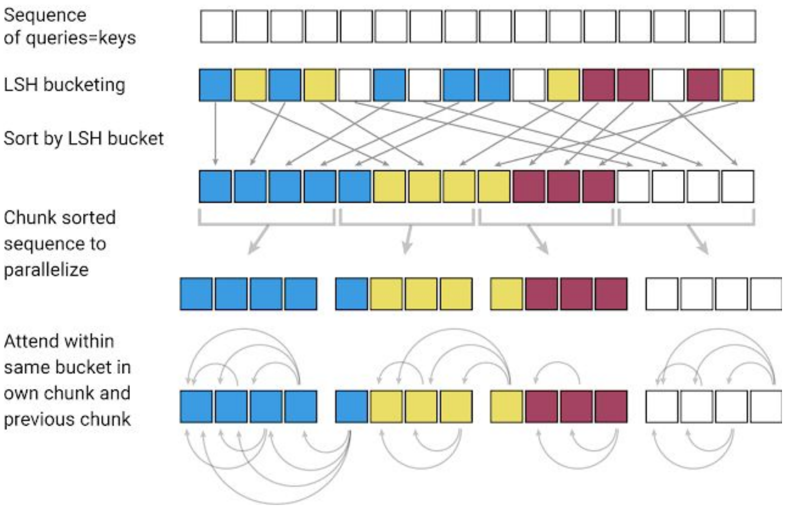
As an example (batches, elements, features) = (2, 6400, 25)



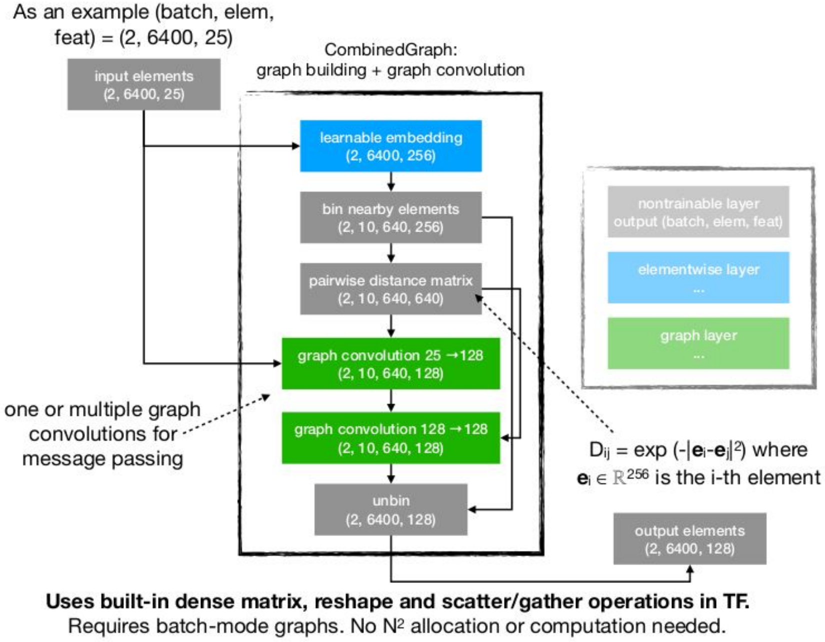
Stacked CombinedGraph (cg) layers, each builds a a new graph in a learnable way and propagates information using graph convolutions.

Scalable graph building

Avoid a quadratic bottleneck with locality-sensitive hashing.

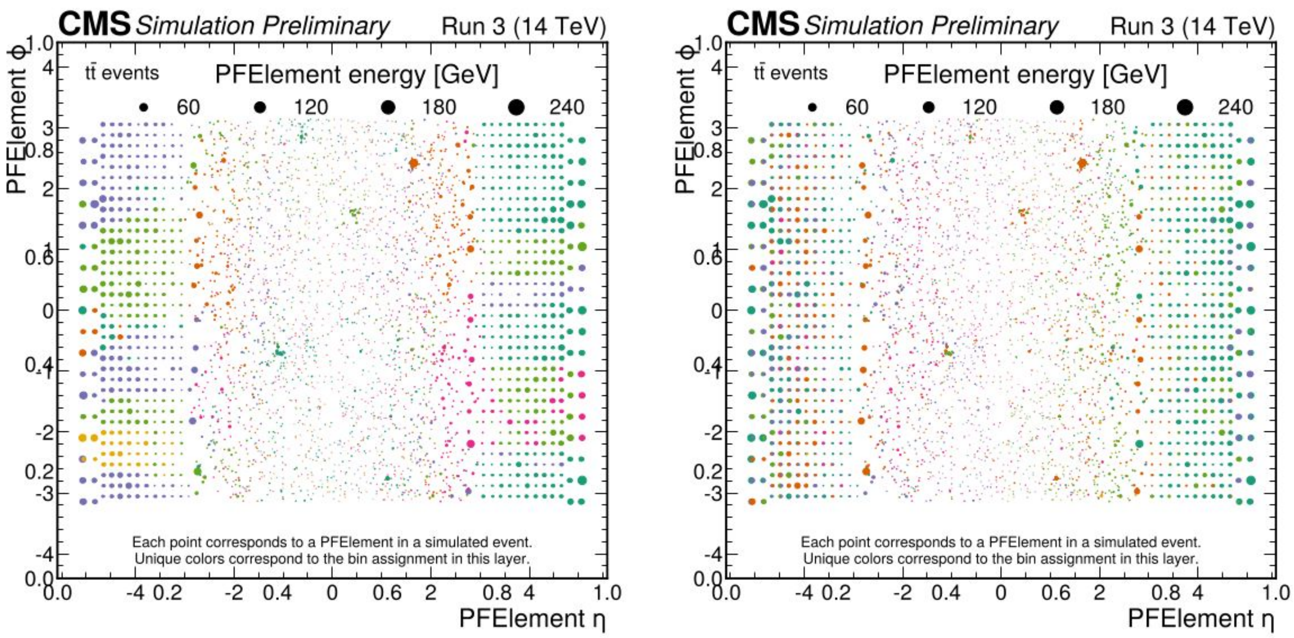


credit



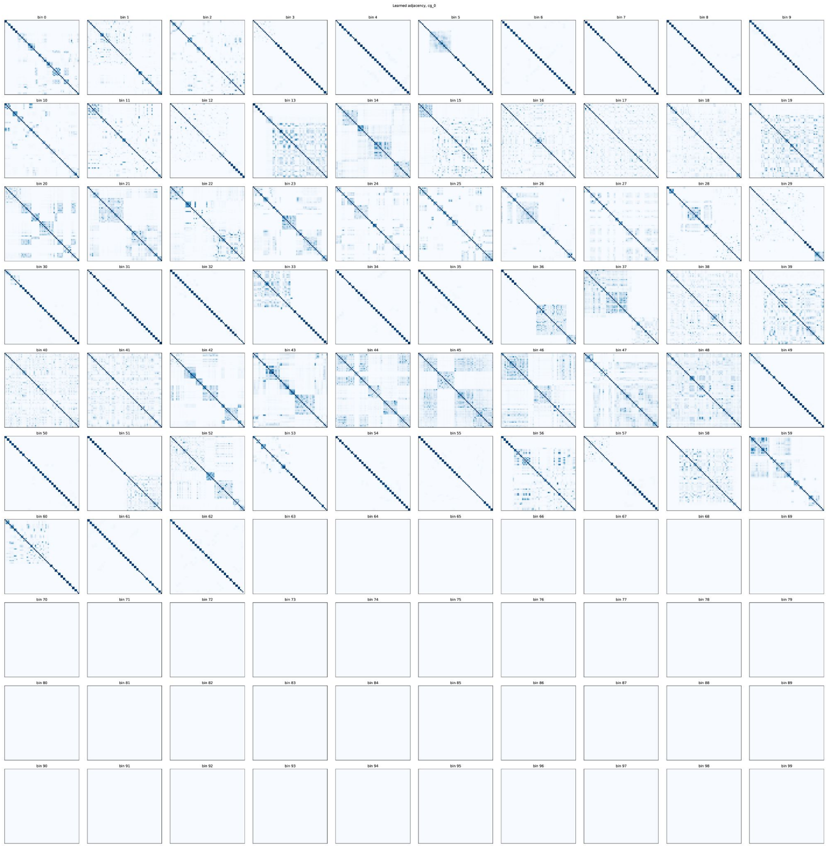
One scalable combined graph layer. The input elements are projected into a learnable embedding space. Nearby elements in the embedding space are binned to fixed-size bins. A fully-connected graph is built in each bin, which is used for one or multiple graph convolutions that are used to transform the input elements. Finally, the transformed elements are unbinned.

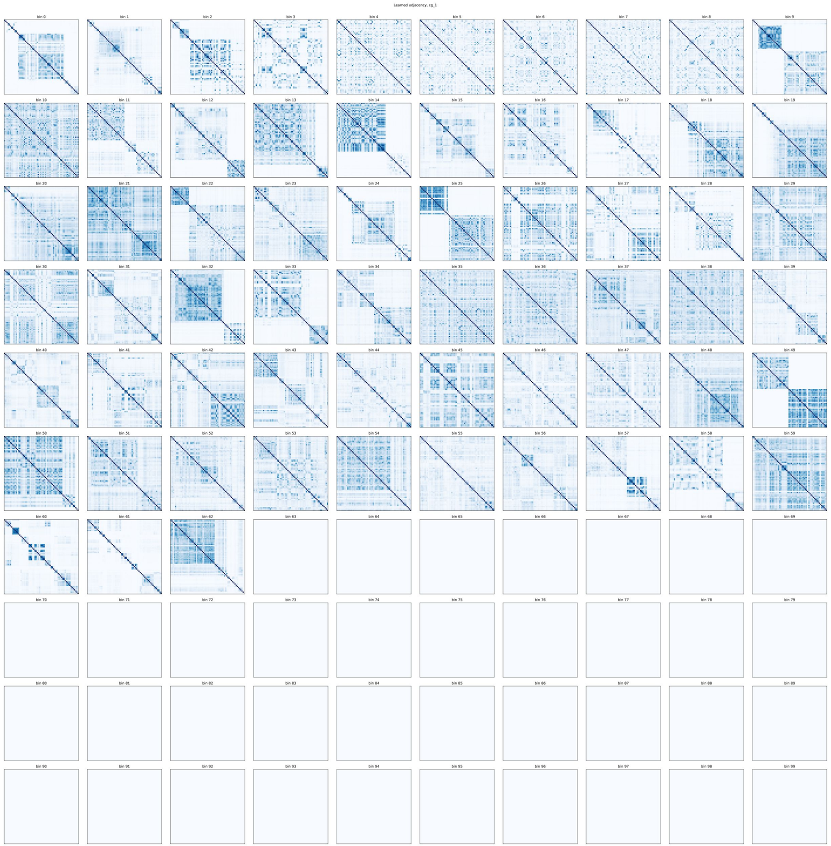
Learned binning



The learned binning structure in the first two layers of the model. We show one simulated ttbar event, with each point corresponding to a PFElement in the event. The colors correspond to the assignment of the PFElements into the bins in each layer.

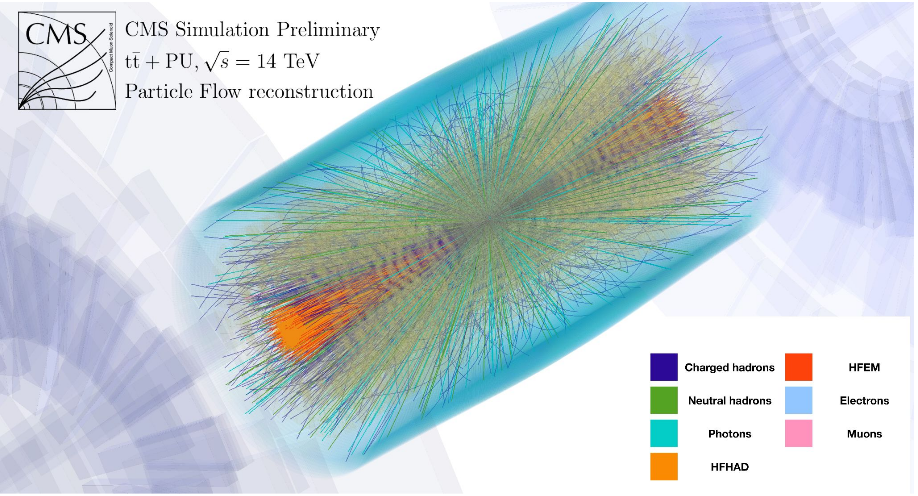
Learned graph structure



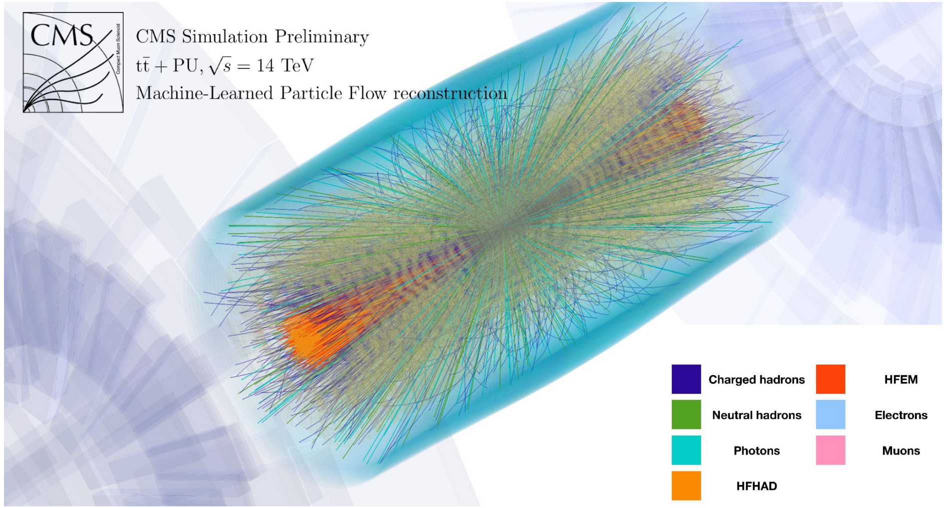


Application in the CMS experiment

Reconstructed with standard particle flow.



Machine-learned particle flow



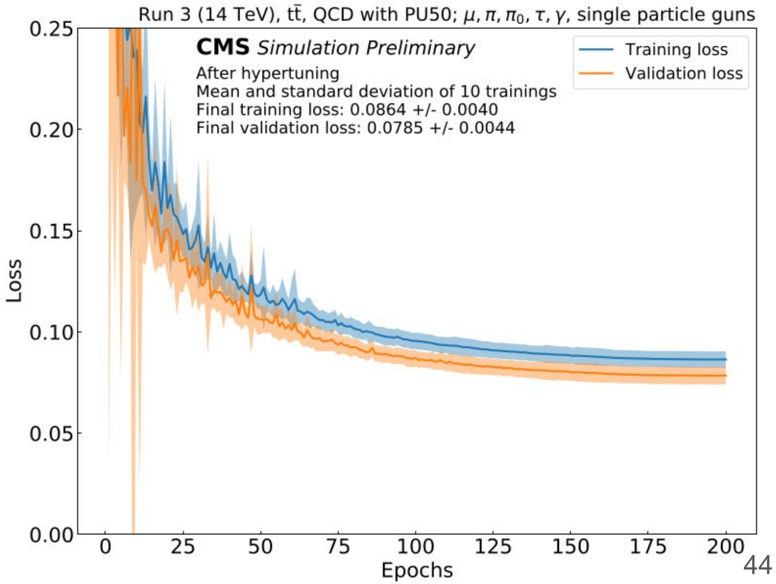
"Machine Learning for Particle Flow Reconstruction at CMS", CMS Collaboration, 2021
[JP on behalf of CMS, ACAT2021, Daejeon, South Korea; CERN CDS]

Training on simulation

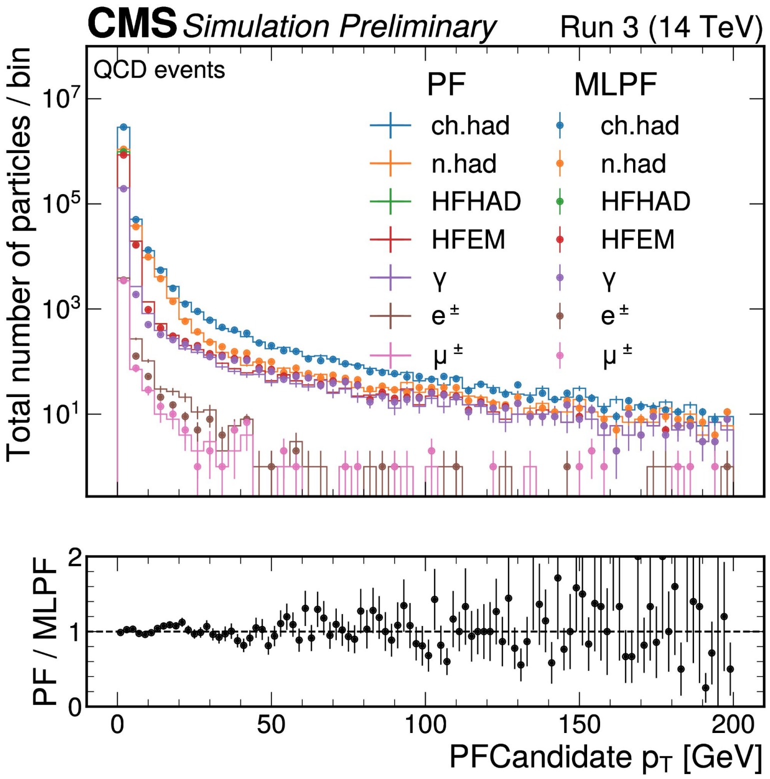
- Trained on 40k events with pileup + 2.4M single-particle events
- ~5 days on 5 GPUs in the KBFI cluster
- Hyperparameter optimization at the Julich supercomputing center [E. Wulff]

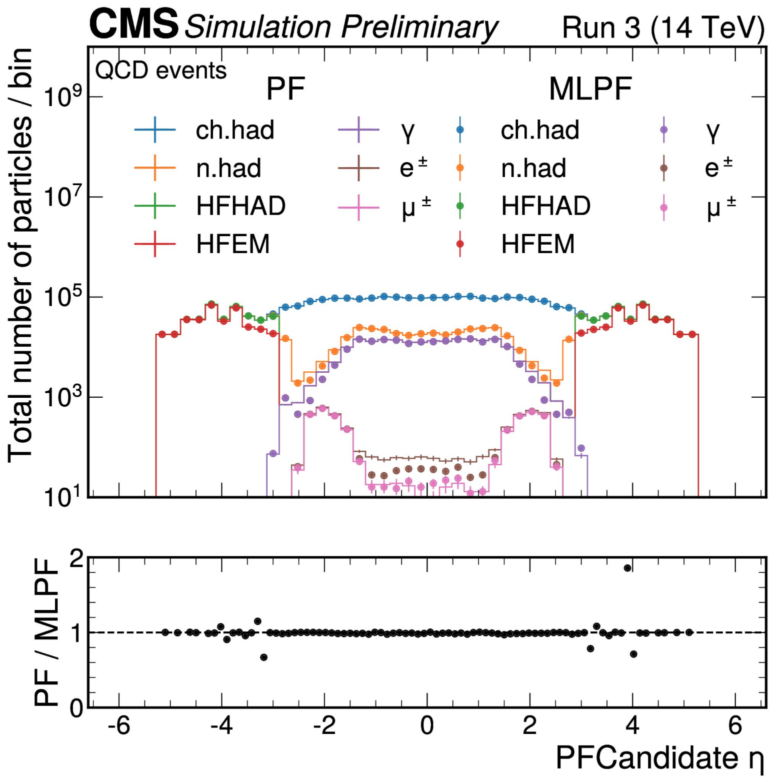
sample fragment	PU configuration	MC events
top-antitop pairs	flat 55-75	20k
$Z \to \tau\tau$ all-hadronic	flat 55-75	20k
single electron flat $p_T \in [1, 100]$ GeV	no PU	400k
single muon flat $p_T \in [0.7, 10]$ GeV	no PU	400k
single π^0 flat $p_T \in [0, 10]$ GeV	no PU	400k
single π flat $p_T \in [0.7, 10]$ GeV	no PU	400k
single τ flat $p_T \in [2, 150]$ GeV	no PU	400k
single γ flat $p_T \in [10, 100]$ GeV	no PU	400k

Table 1: MC simulation samples used for optimizing the MLPF model.

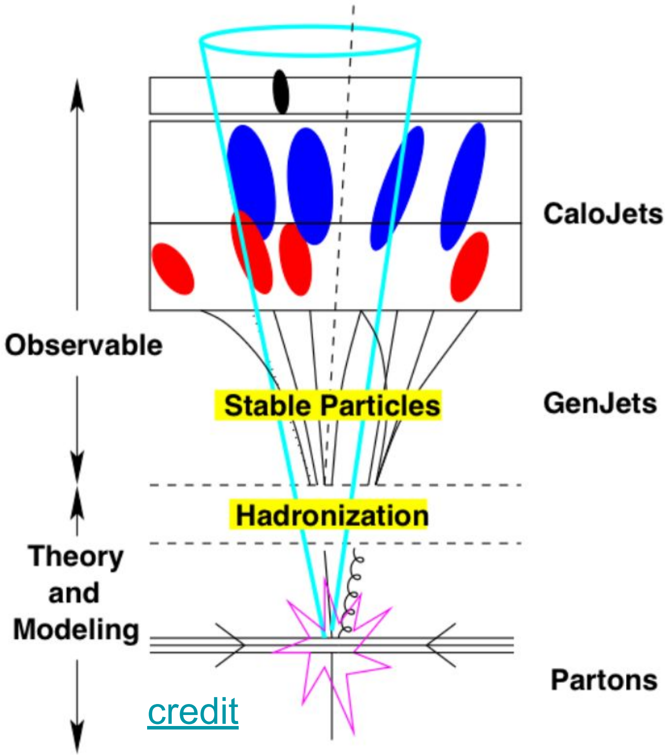


Particle distributions

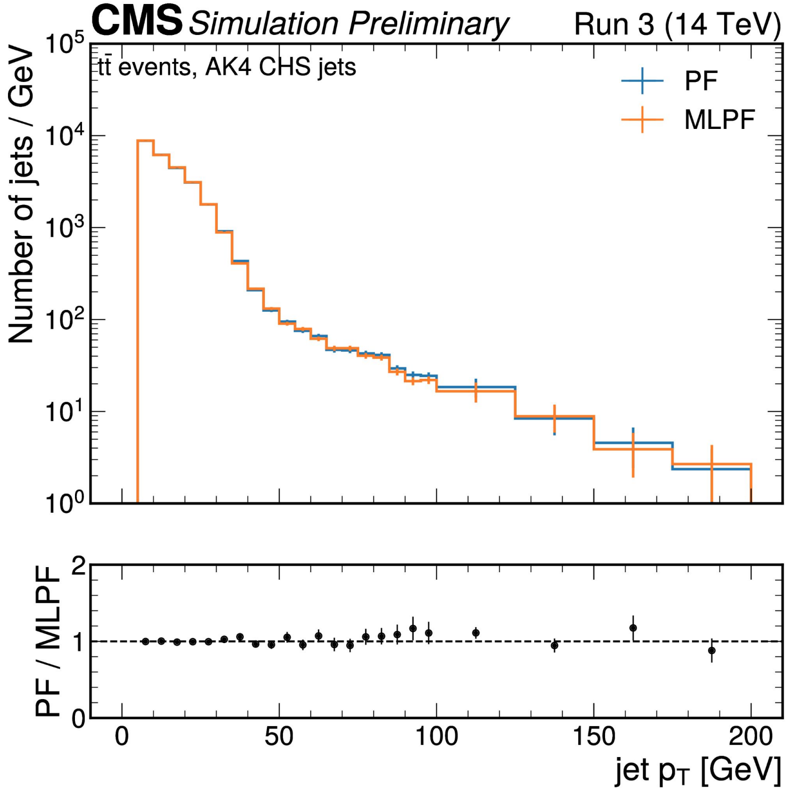




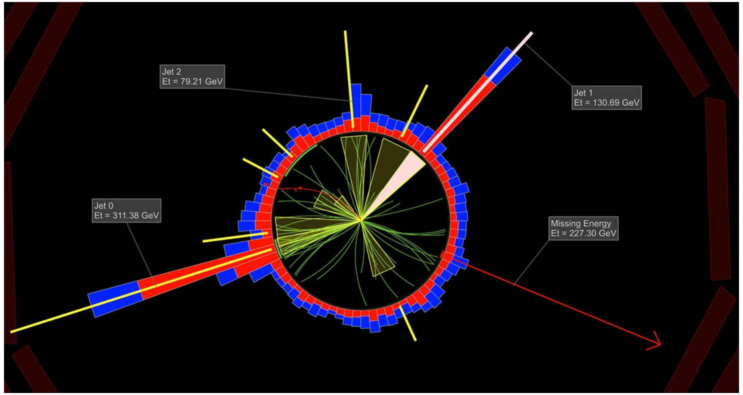
Jets in full reconstruction

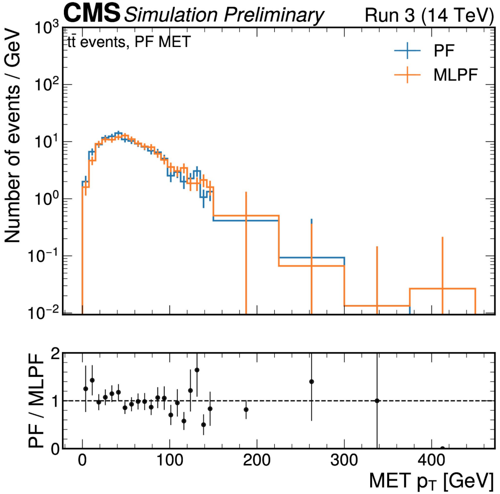


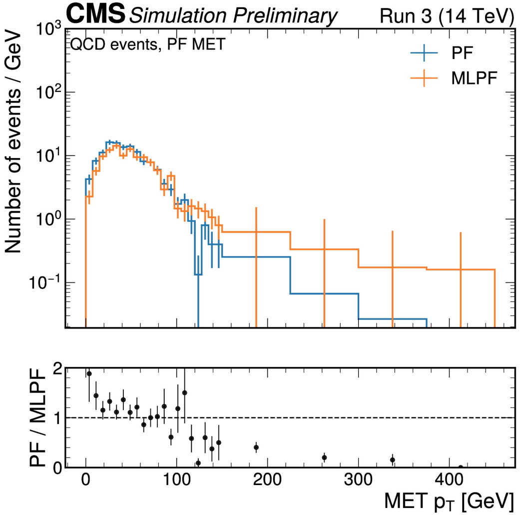
credit



Missing transverse energy

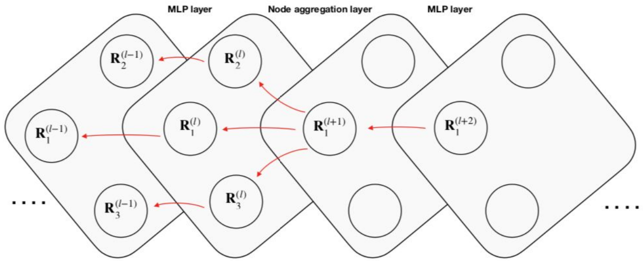






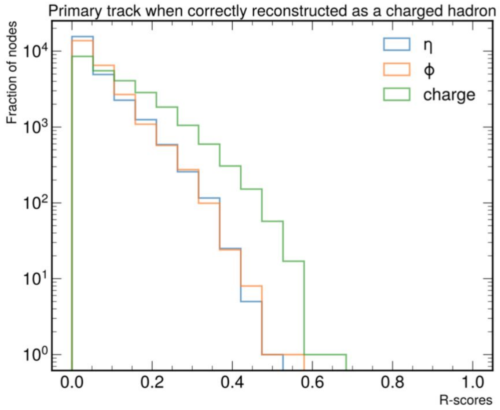
Validate performance under different physics conditions (=datasets).
We found that we need to augment the training dataset with more high-energy neutral hadrons for better generalization to e.g. QCD events.

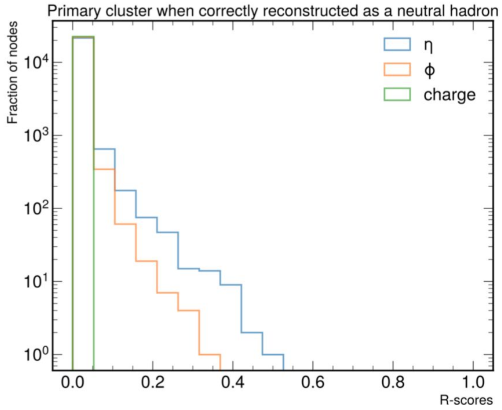
Interpreting ML models



- What inputs are relevant for a particular model output?
- Compute layerwise relevance scores
- Aggregate along the graph structure

$$\mathbf{R}_j^{(l)} = \sum_k \frac{x_j A_{jk}}{\sum_m x_m A_{mk}} \mathbf{R}_k^{(l+1)}$$

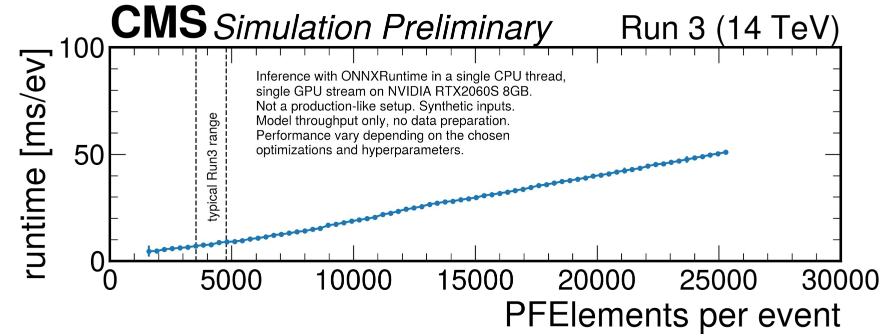


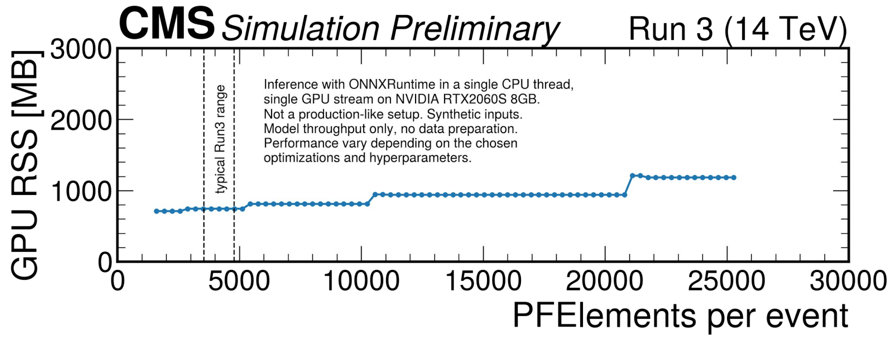


“Explaining machine-learned particle-flow reconstruction”; Farouk Mokhtar, Raghav Kansal, Daniel C Diaz Javier Duarte, **JP**, Maurizio Pierini, Jean-Roch Vlimant [NeurIPS 2021, Machine Learning and the Physical Sciences](), 2111.12840 [physics.data-an]

Speeding up reconstruction

- Besides good physics performance, reconstruction needs to be fast and computationally efficient
- Neural nets are well-suited for GPUs & other parallel processors
- Important to avoid a quadratic scaleup with occupancy
- Next steps are to test the MLPF algorithm on real data in CMS in Run3
- Also looking into extending this for FCC reconstruction





Next steps on MLPF

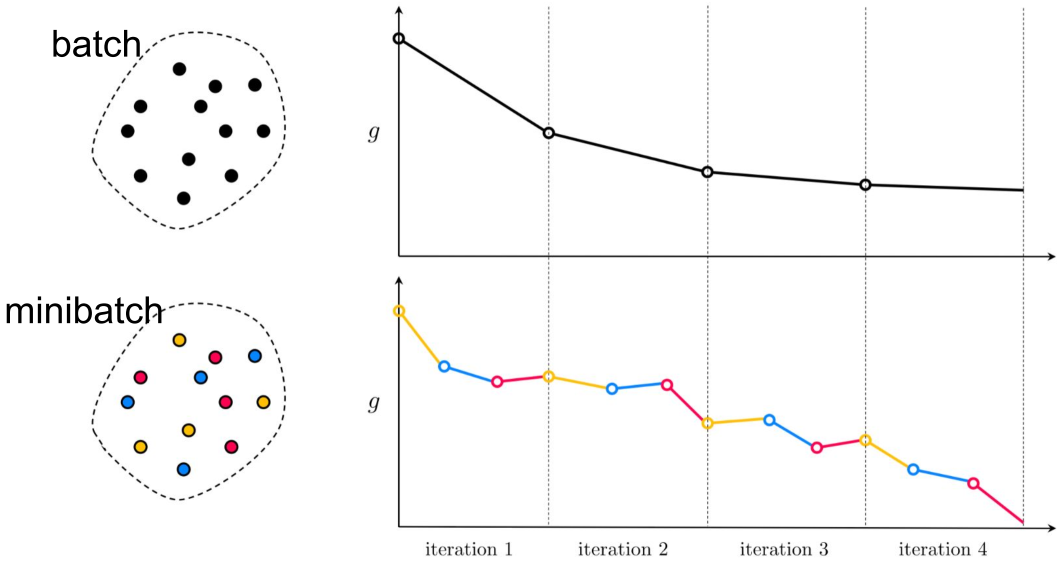
- Improve the training statistics, additional validation in the tails of distributions
- Improve GPU inference integration in CMS reconstruction software
- LHC Run 3 is an opportunity to test machine-learned particle flow reconstruction on real data!
- The algorithm is generic - possible feasibility studies for future detectors
- This dataset for further studies on interpretability
- Integrate with machine-learned tracking and clustering from upstream reconstruction

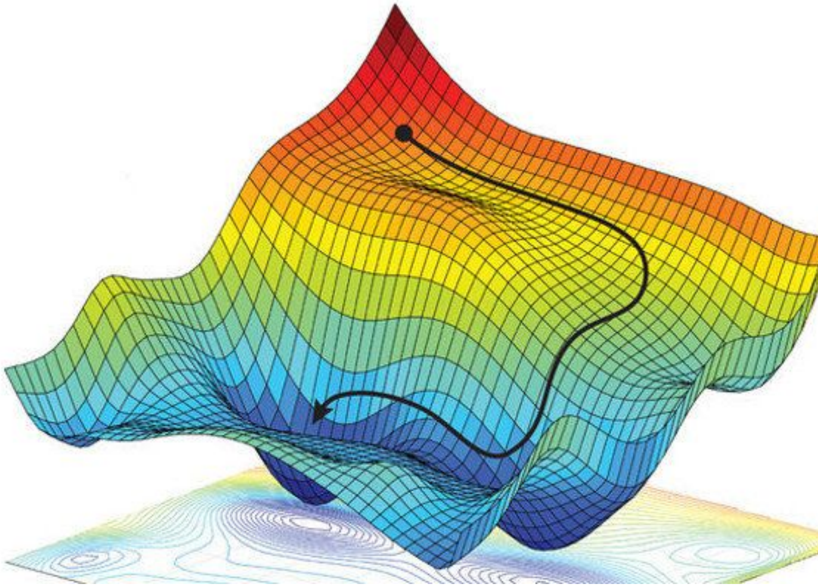
Summary

- **Fundamental physics + ML**: a unique combination of large datasets, accurate underlying quantitative models and hard physics problems
- The **LHC is a rich area for applied ML methods**, hundreds of models running in production at any given time
- Machine learning is about **fitting distributions on data** with numerical optimization, can **augment imperative algorithms**
- Data r**econstruction at the LHC is a challenging problem**, well-suited to differentiable, machine-learned algorithms:
  - "Efficient machine-learned particle-flow reconstruction using graph neural networks"; **JP**, J. Duarte, J-R Vlimant, M. Pierini, M. Spiropulu; Eur. Phys. J. C (2021) 81: 381
  - "Machine Learning for Particle Flow Reconstruction at CMS", CMS Collaboration, 2021 [JP on behalf of CMS, ACAT2021, Daejeon, South Korea; CERN CDS]
  - "Explaining machine-learned particle-flow reconstruction"; Farouk Mokhtar, Raghav Kansal, Daniel C Diaz Javier Duarte, **JP**, Maurizio Pierini, Jean-Roch Vlimant; NeurIPS 2021, Machine Learning and the Physical Sciences
- Encoding **physics priors (=symmetries) can improve the representation power** of neural networks

Backup

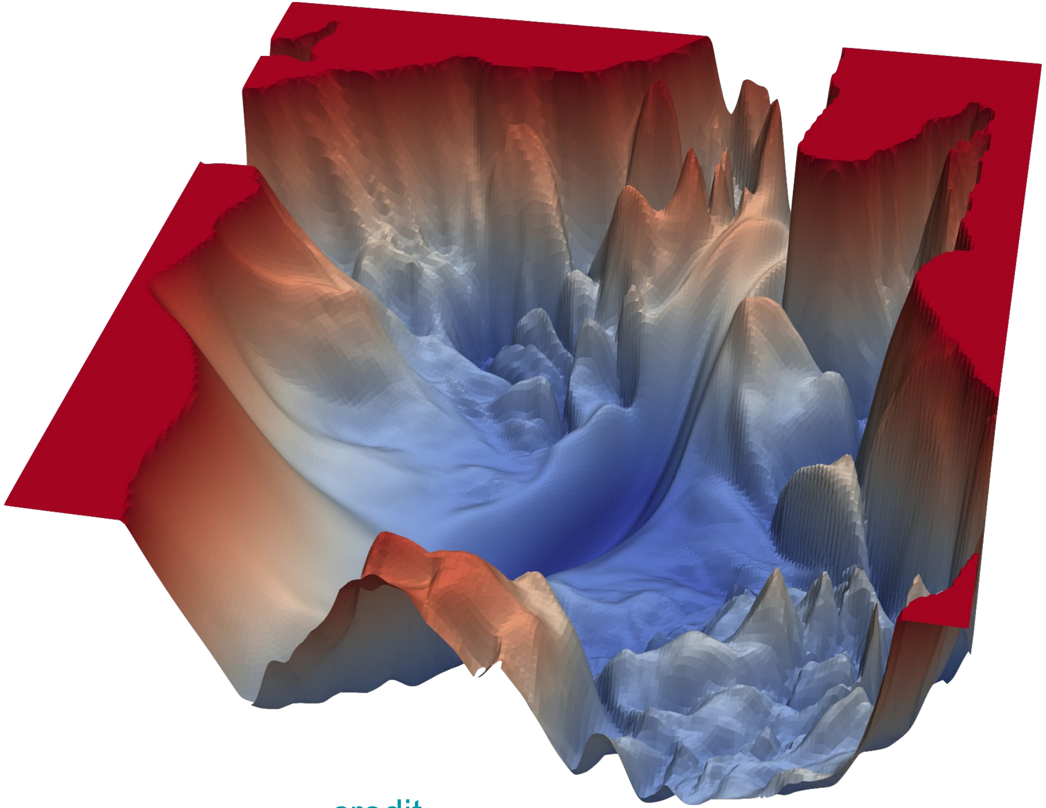
Optimization over large datasets





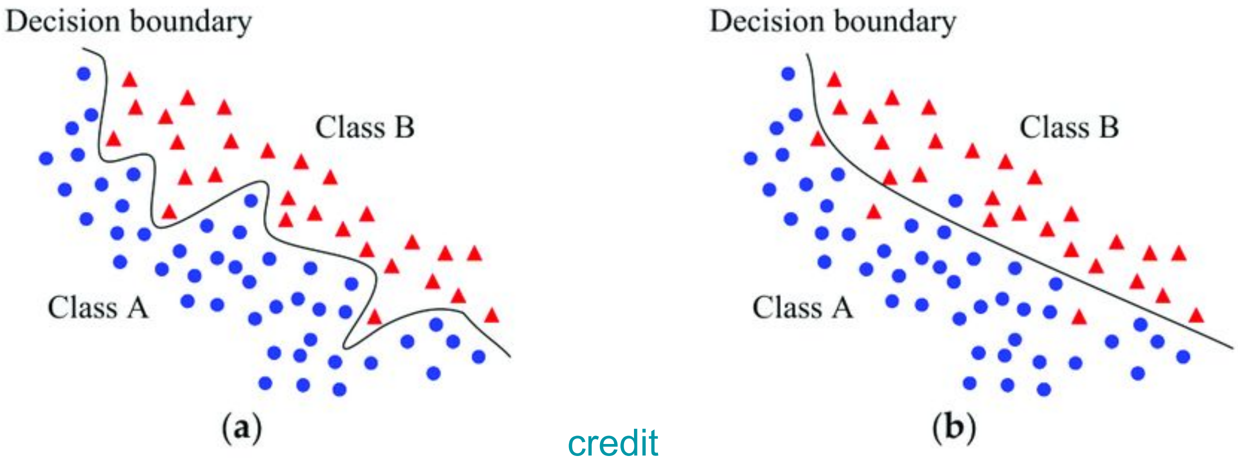
credit

Loss surface of ANN-s



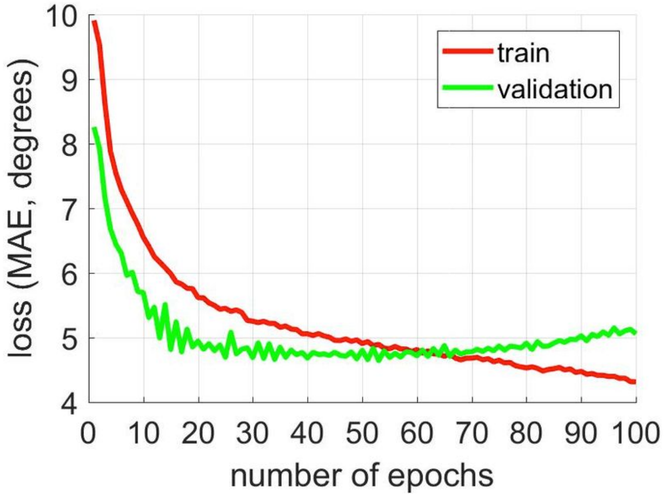
credit

Overfitting



credit

Train, test and validate

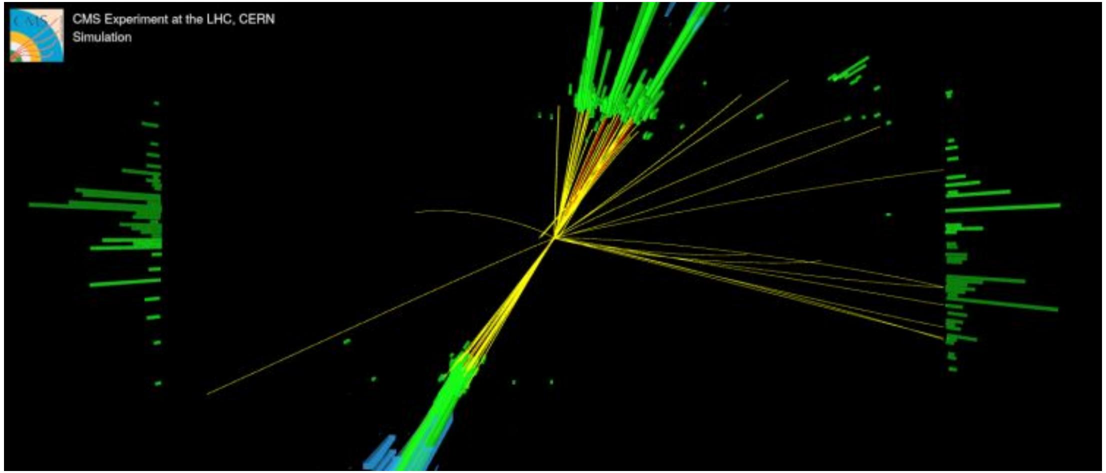


Lessons learned

- It’s all about the dataset!
- Set up a simple baseline method and simple performance metrics!
- Decouple the model and how you measure the performance of a model.
- Change one thing at a time.
- Visualize a few predictions, understand where and why they fail.
- Visualize the learning dynamics. What is learnt quickly, what takes time?
- Don’t try fancy methods before you get a simple method to work.
- Try to reuse existing models before inventing your own.
- ML can only attempt to answer to questions that you can pose quantitatively.
- **All models are wrong, some models are useful.**

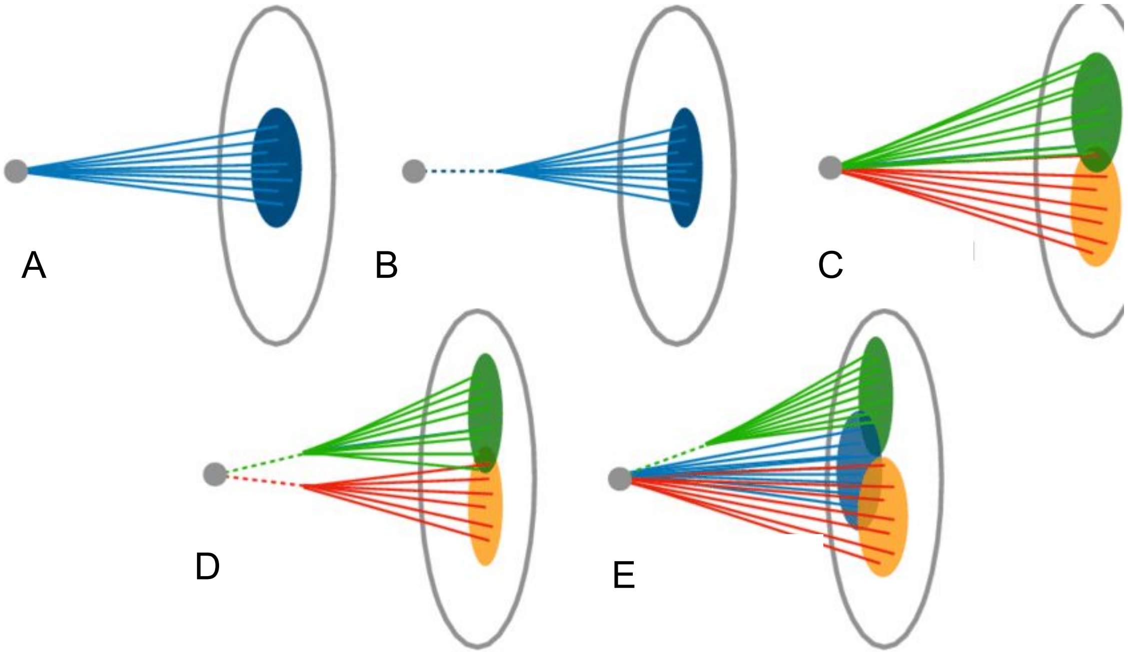
Top quark / Higgs jets

credit



One top jet, one W jet. Credit: CMS.

Jet substructure to jet identification

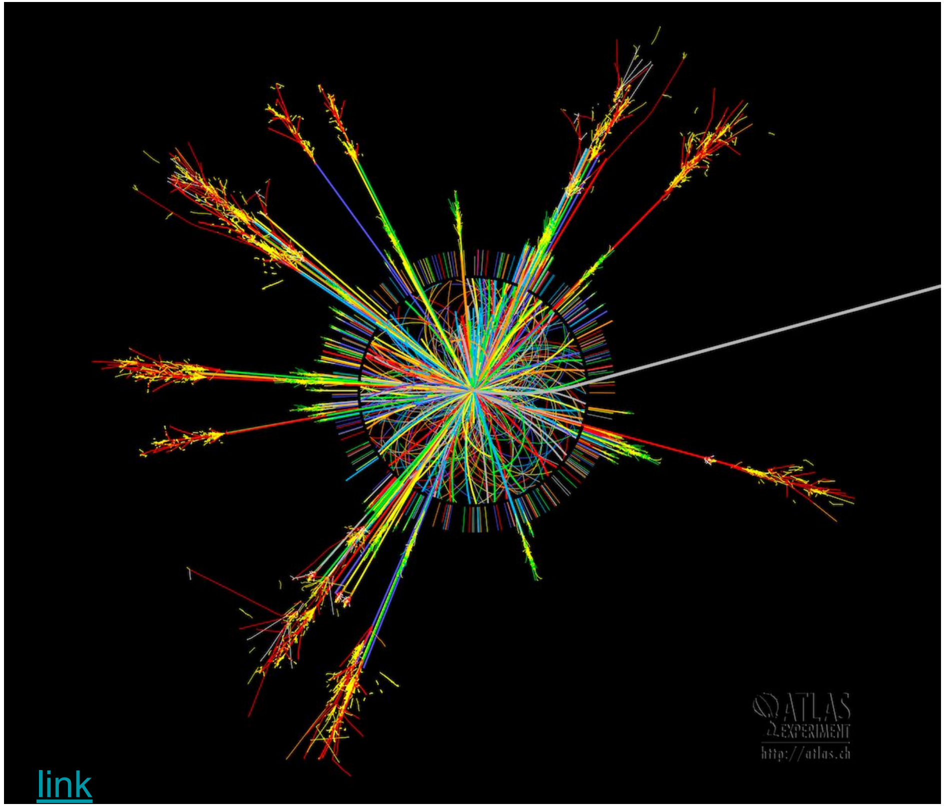


link

Which is which?

- q/g
- H->bb
- t->bW->bqq
- b
- W->qq

A full event



link

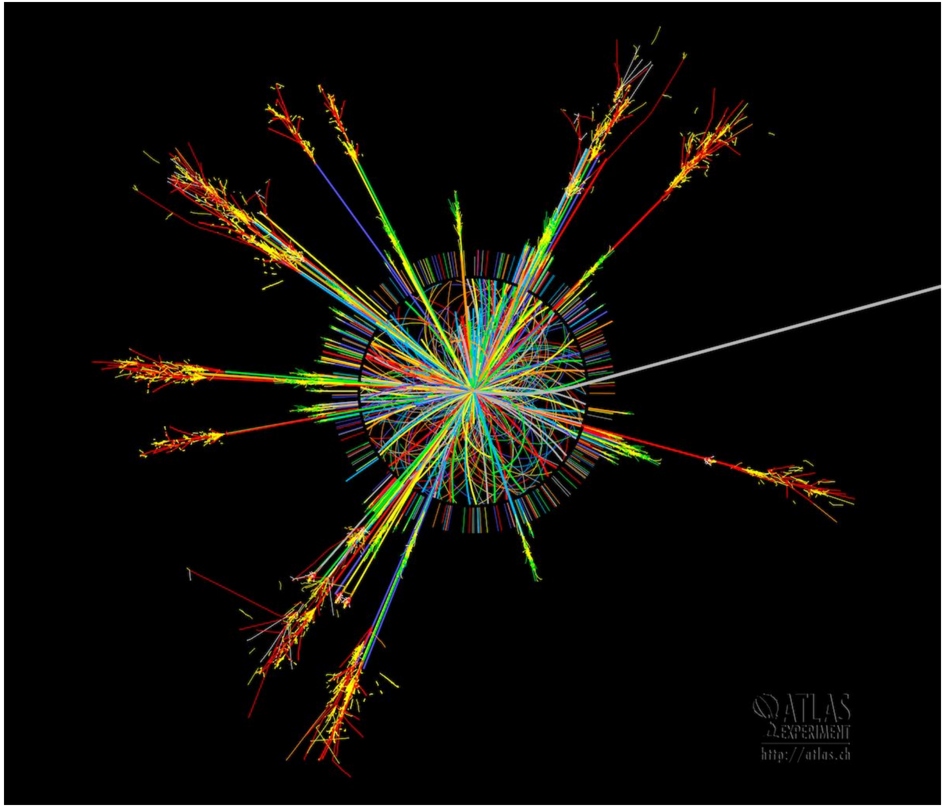
This event has:

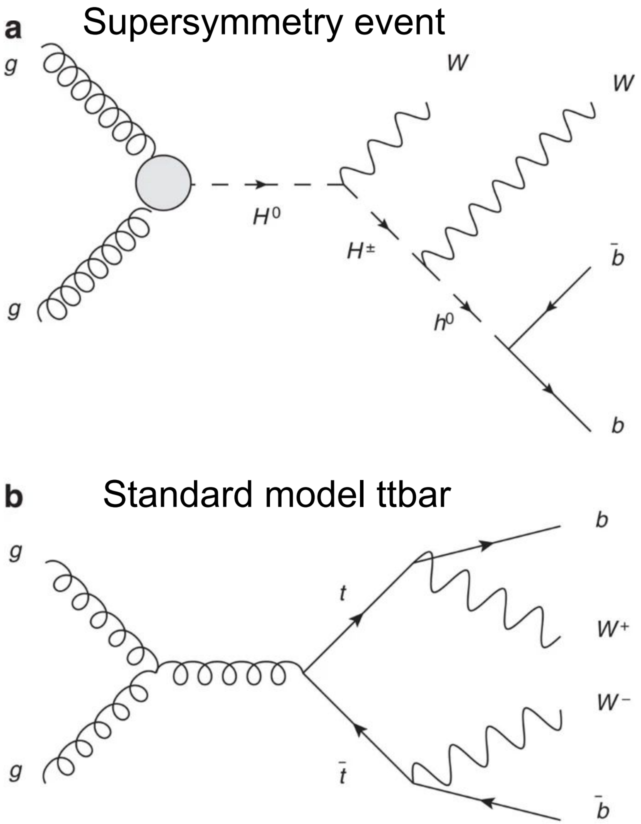
A. ~5 jets
B. ~10 jets
C. ~20 jets
D. ~50 jets

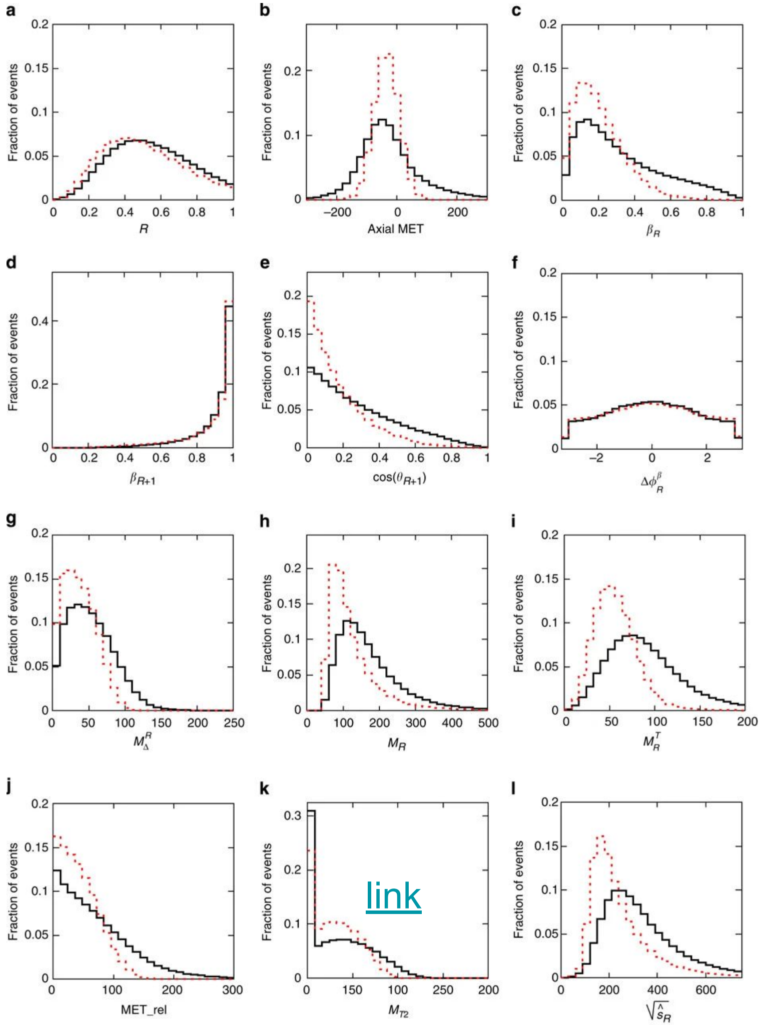
We need X numbers to represent this event at the level of momentum vectors of the jets and leptons:

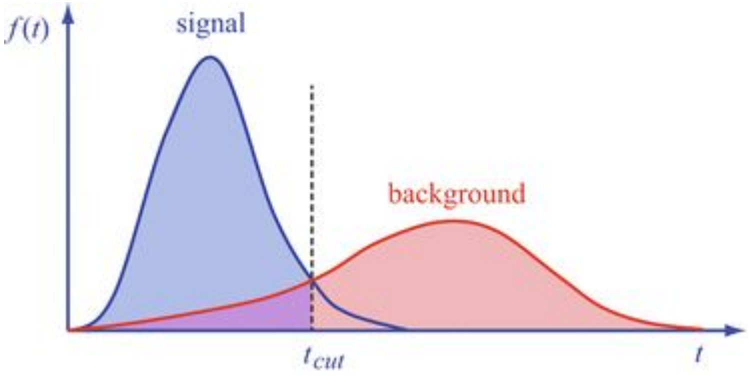
A. X~10
B. X~50
C. X~500
D. X~5000

Identifying collision events



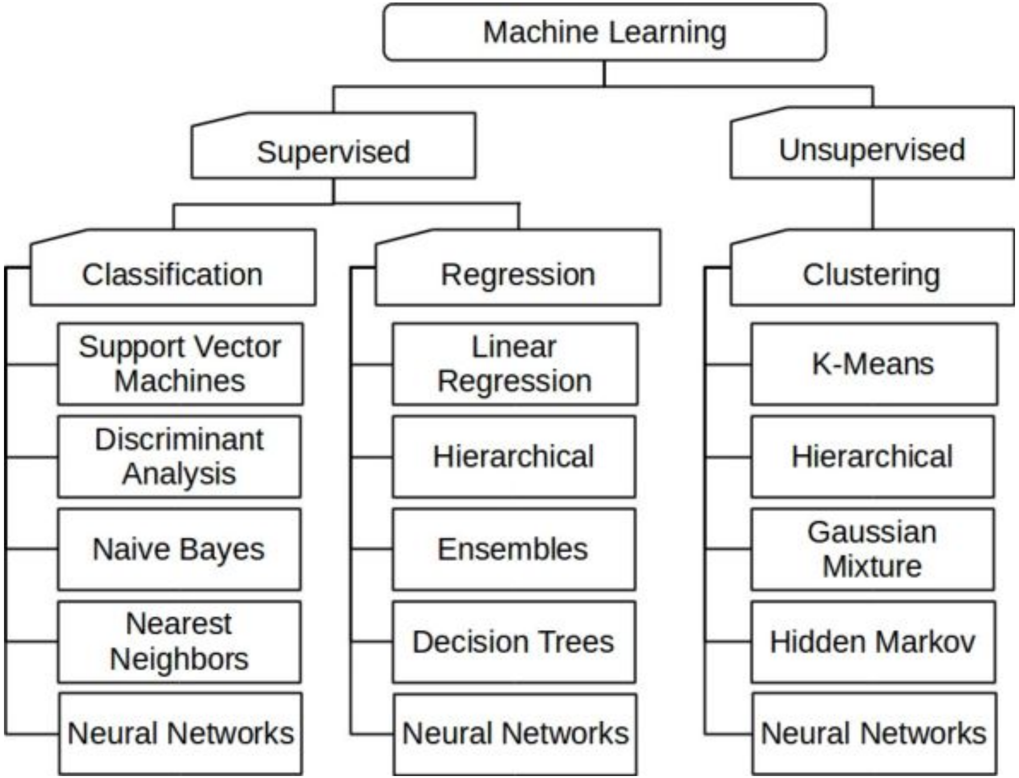






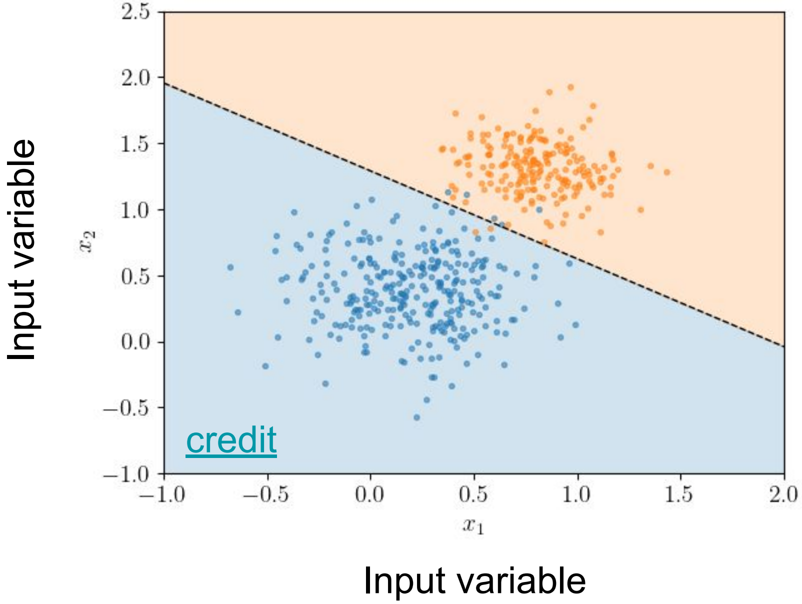
Statistical discriminator

Machine learning: mathematical models optimized on data.



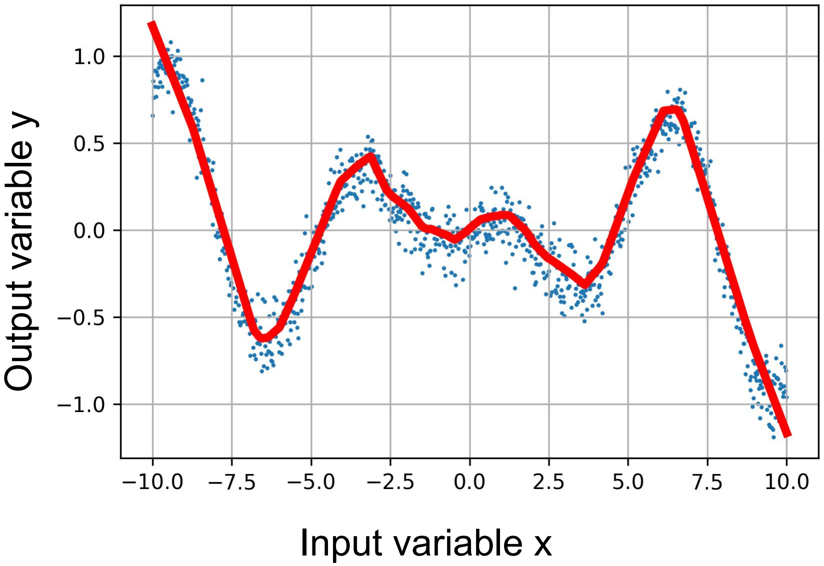
credit

Classification



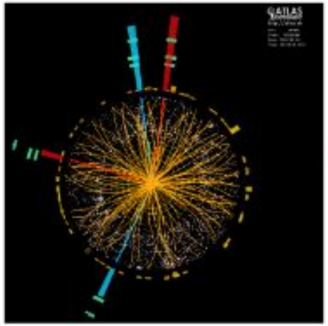
x1	x2	label	Label as number
0.5	0.5	blue	0
1.0	0.7	blue	0
1.0	1.0	orange	1
...	...	...	

Regression



x	y
0.0	0.1
0.0	0.15
0.2	0.02
...	...

Supervised learning



h(**x**; **w**)
Function with adjustable parameters

Loss Function

Compare prediction with true label

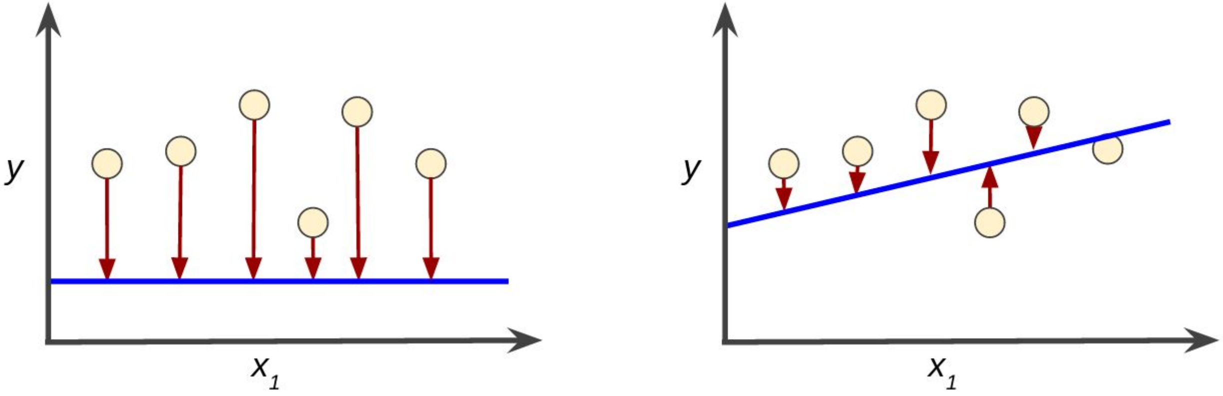
Loss

<u>True labels:</u>
Higgs = 1
Bkg = 0

credit

Optimization

Our model: *y* = *mx* + *c*



Mean squared error $E = \frac{1}{n}\sum_{i=0}^{n}(y_i - \bar{y}_i)^2$

Which model has a lower overall error?

A. Left
B. Right

How many parameters does the linear model (blue line) have?

A. One
B. Two
C. Three
D. Undefined

What are the units of the mean squared error E?

C. The units of y
D. The units of y^2
E. Unitless

Optimization game

Compute total error

$$E = \frac{1}{n} \sum_{i=0}^{n} (y_i - (mx_i + c))^2$$

Compute derivatives of dE/dm, dE/dc

$$D_m = \frac{1}{n} \sum_{i=0}^{n} 2(y_i - (mx_i + c))(-x_i)$$

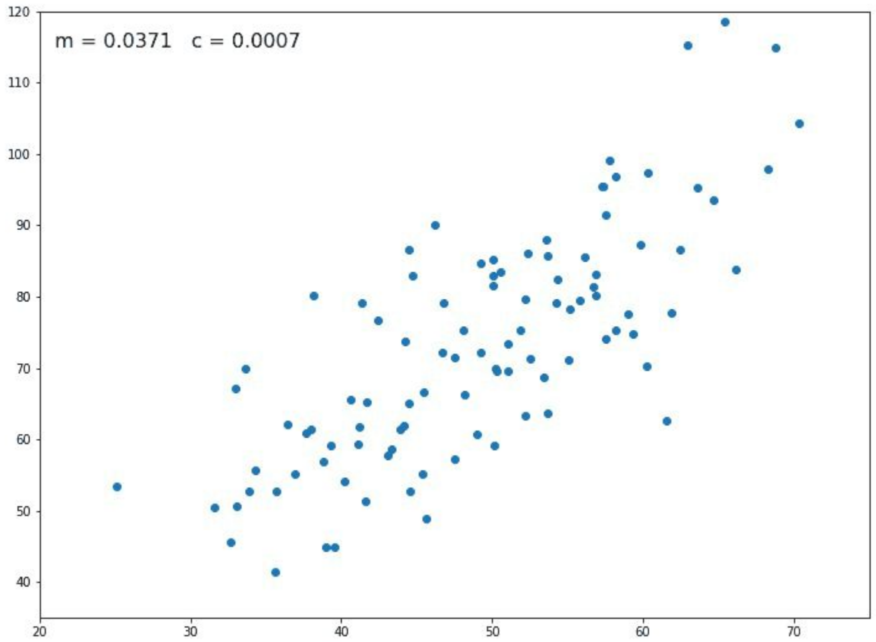
$$D_m = \frac{-2}{n} \sum_{i=0}^{n} x_i (y_i - \bar{y}_i)$$

$$D_c = \frac{-2}{n} \sum_{i=0}^{n} (y_i - \bar{y}_i)$$

Update m, c

$$m = m - L \times D_m$$

$$c = c - L \times D_c$$

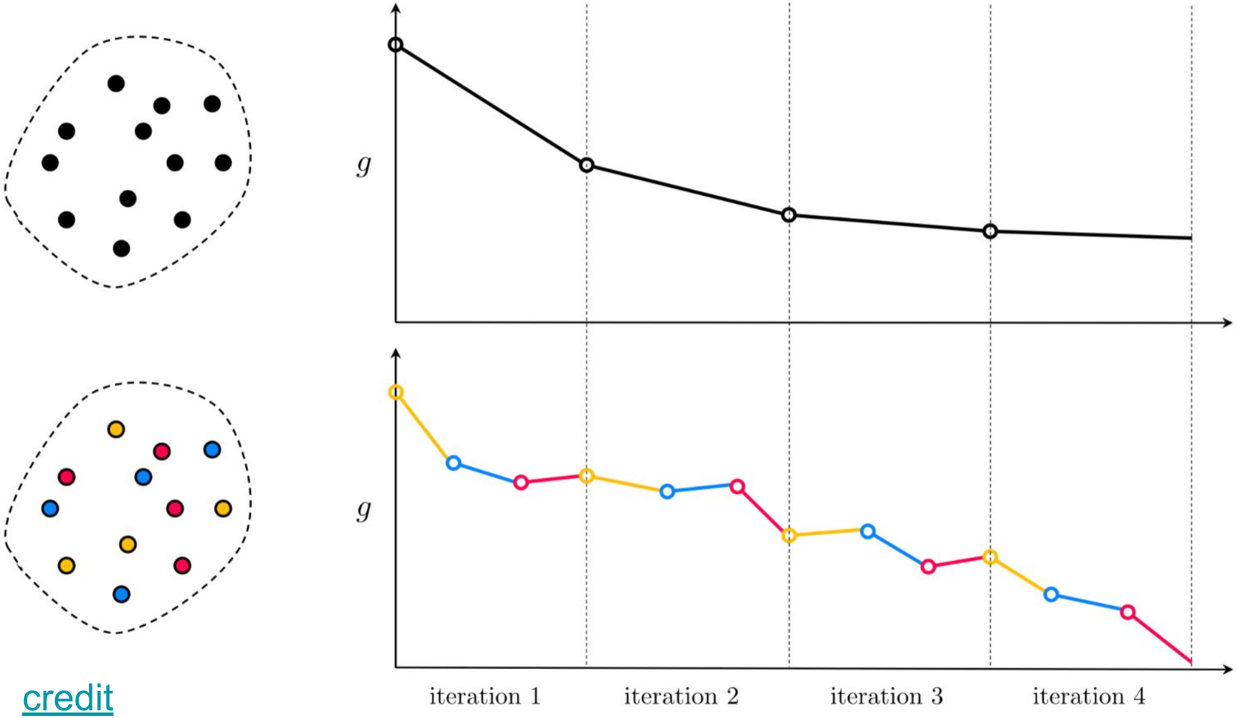


credit

Batched gradient descent

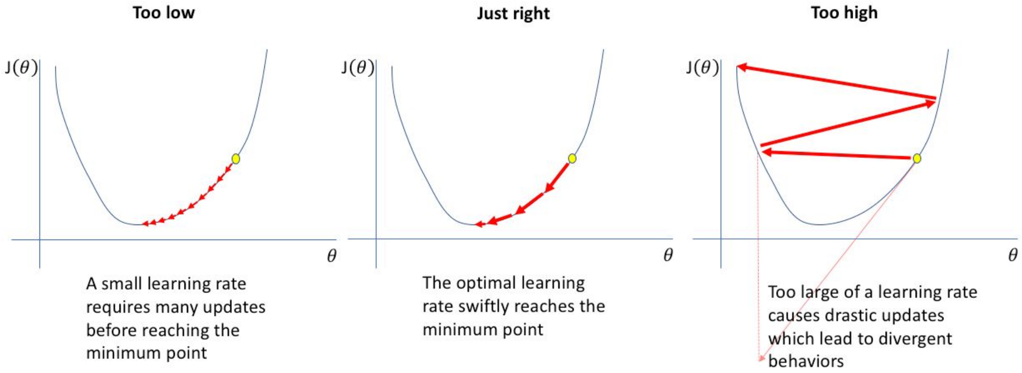
batch

minibatch

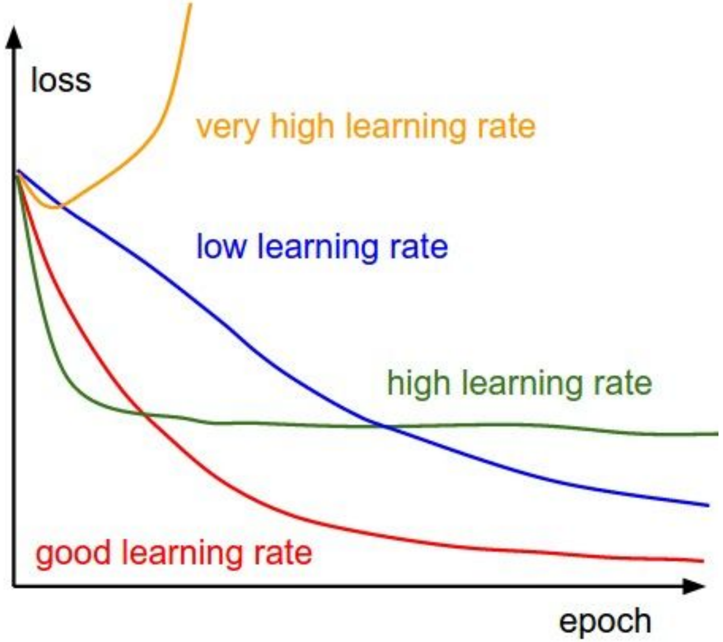


credit

Choosing the right learning rate

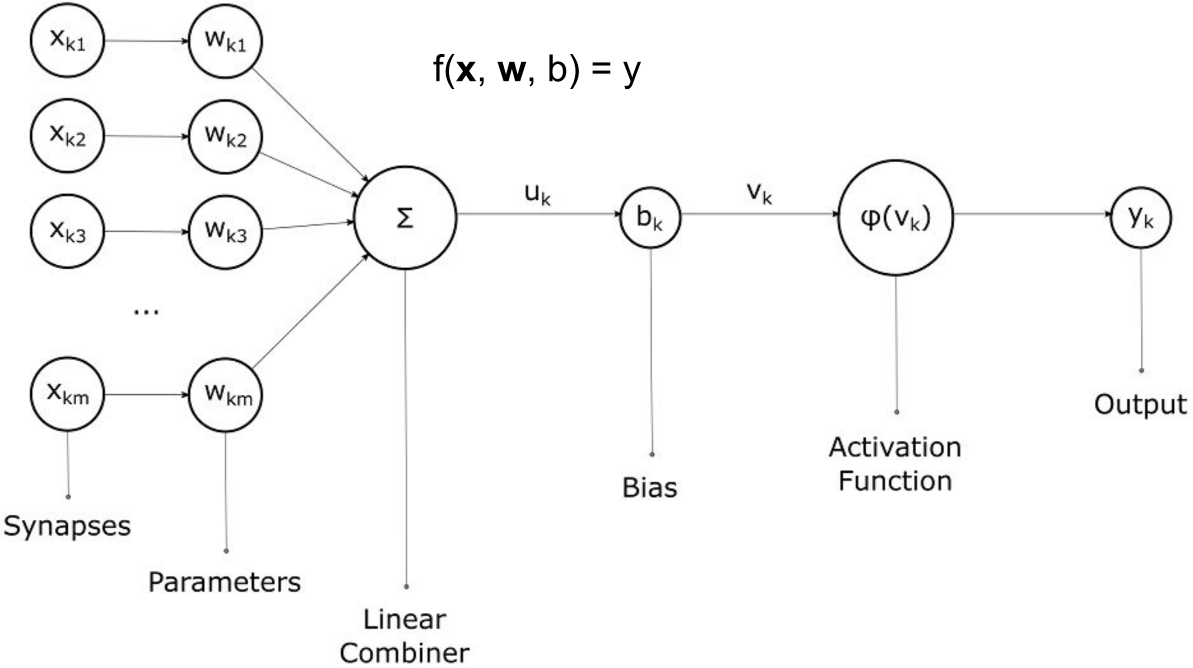


credit



credit

Artificial/deep neural network



f(**x**, **w**, b) = y

How many tunable parameters does this model have?

A. k_m+1
B. $k_m (k_m + 1)$
C. $2 k_m$

In this image, u_k, v_k, y_k are

A. Scalars (single numbers)
B. Vectors (1D lists of numbers)
C. Tensors (nD matrices)

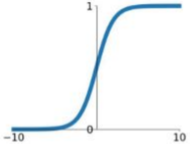
k_m inputs

1 output

Activation Functions

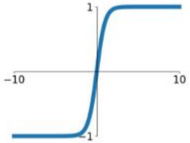
Sigmoid

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



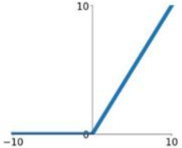
tanh

$\tanh(x)$



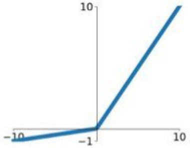
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

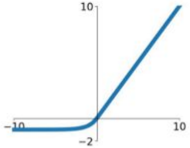


Maxout

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

ELU

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

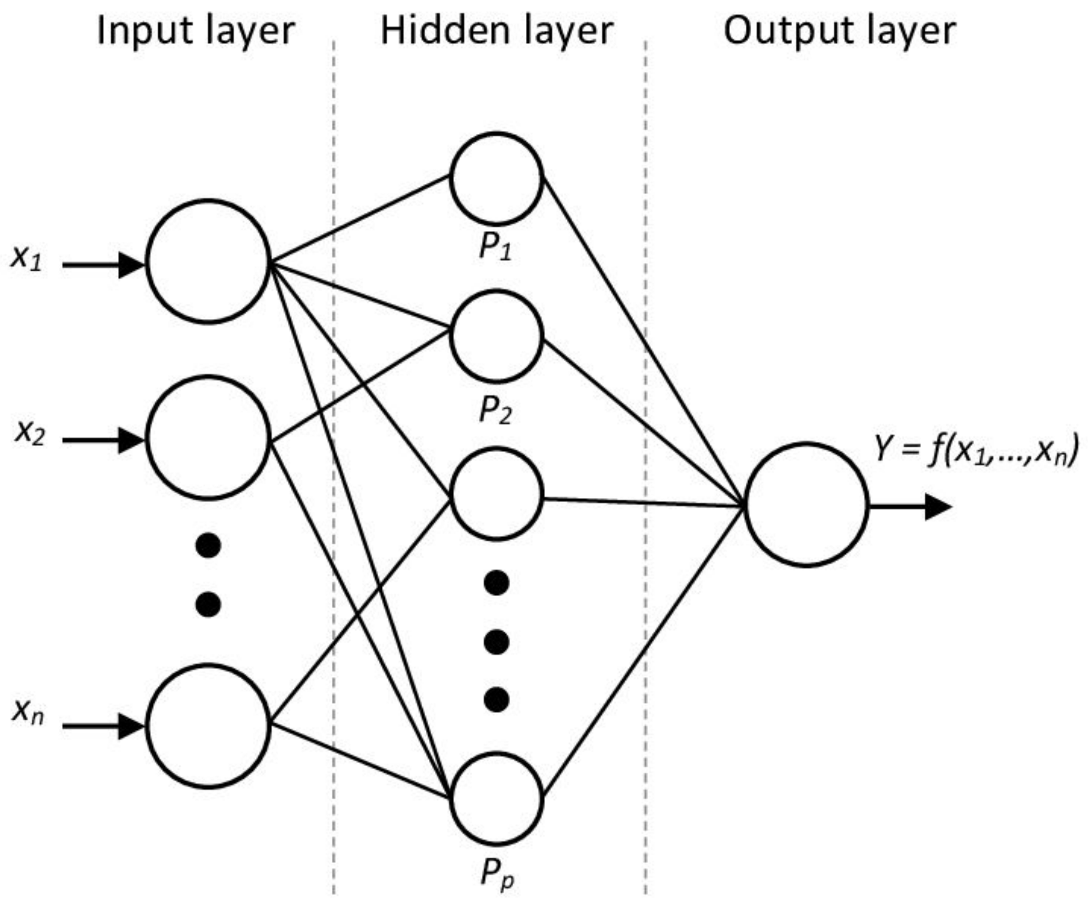


What would be a suitable activation function for binary classification, with output values from 0...1

A. Sigmoid
B. ReLU
C. Linear

What would be a suitable output activation function for regression, where the output domain is 0...500 (e.g. reconstructed mass)

A. Sigmoid
B. ReLU
C. Linear

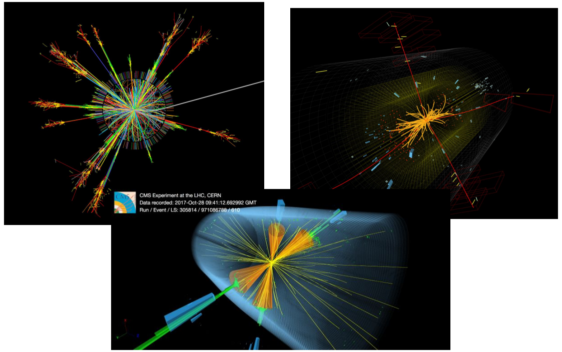


n inputs $\quad$ *p* hidden units $\quad$ 1 output

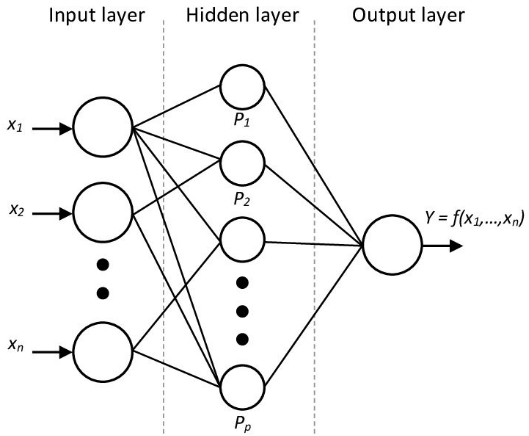
The number of nodes in the hidden layer *p* should be

A. Larger than the number of inputs *n*
B. Smaller than the number of inputs *n*
C. Exactly the same as the number of inputs *n*
D. Is not fixed and can be chosen as needed

Representing data



Collider events contain a variable number of particles of various types.



Typical DNNs require a fixed size *n* input.

How to map events $\longrightarrow \mathbb{R}^n$?

Categorical variables

One-hot encoding.

Color
Red
Red
Yellow
Green
Yellow

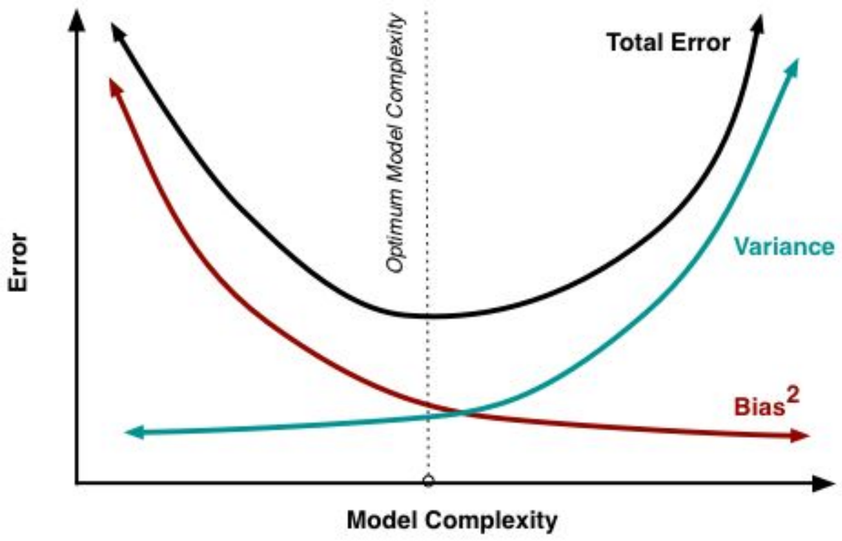
Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1
0	1	0

credit

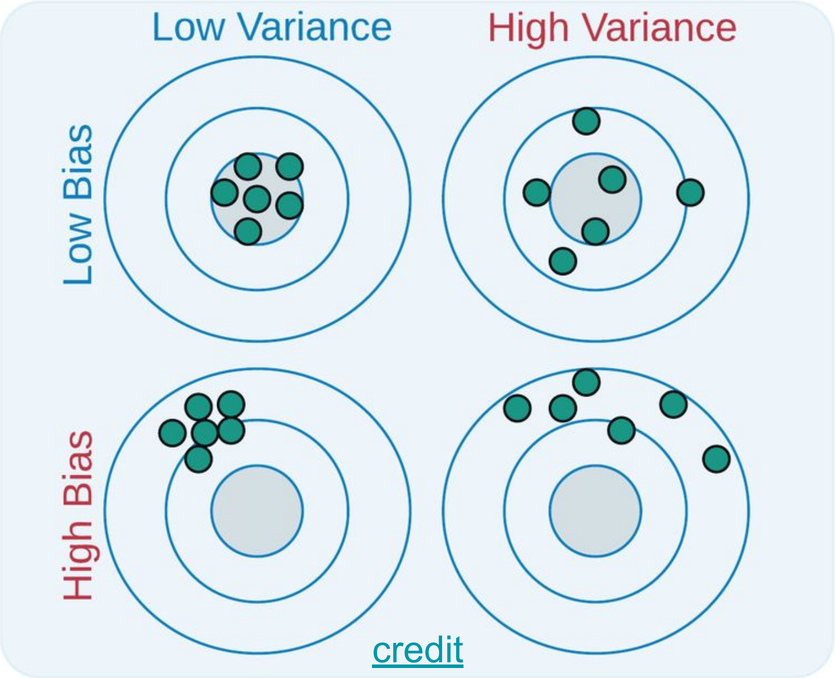
Your MC simulation can contain jets, electrons, muons and photons. How many bits are required to represent objects of all classes?

A. One
B. Two
C. Three
D. Four
E. Five

Bias vs. variance

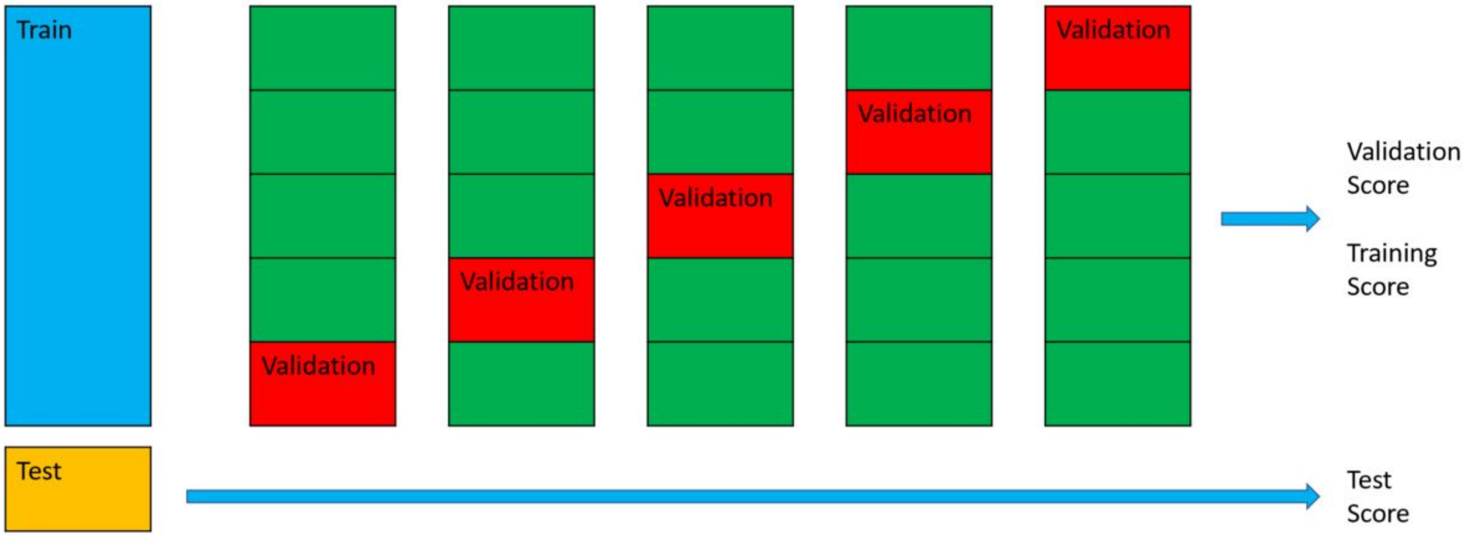


credit



credit

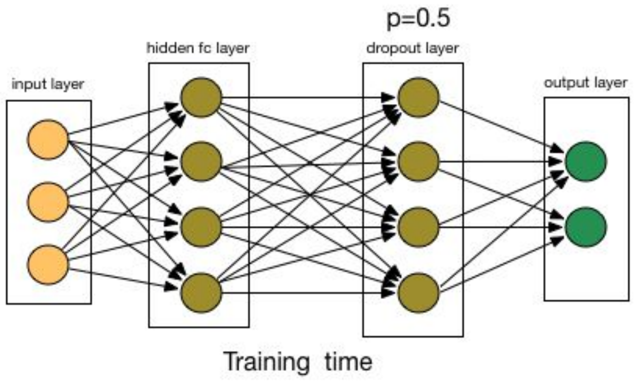
Training and validation datasets



Regularization

Building robust models with respect to fluctuations in the input dataset.

Dropout: randomly disable neural network nodes at training time.

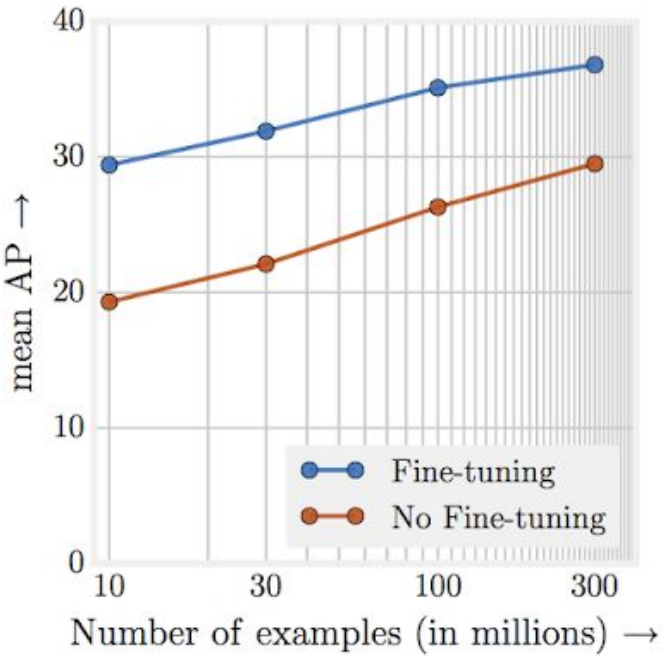


credit

What would happen if the dropout was applied not only in training, but also during inference?

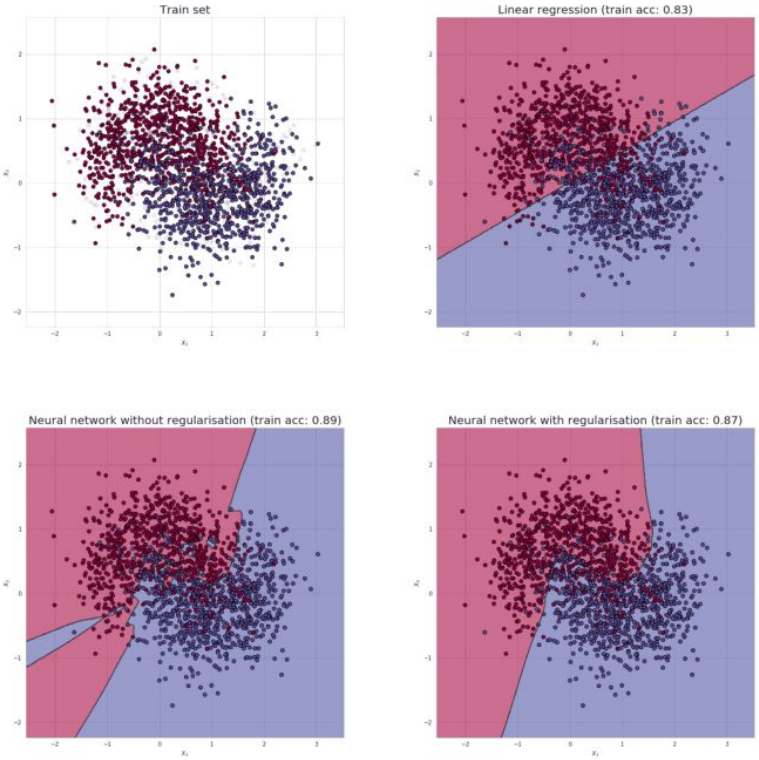
A. The network output would always be zero
B. The network output would be even more regularized
C. The network output would be more noisy from one prediction to the next.

Regularization with more data



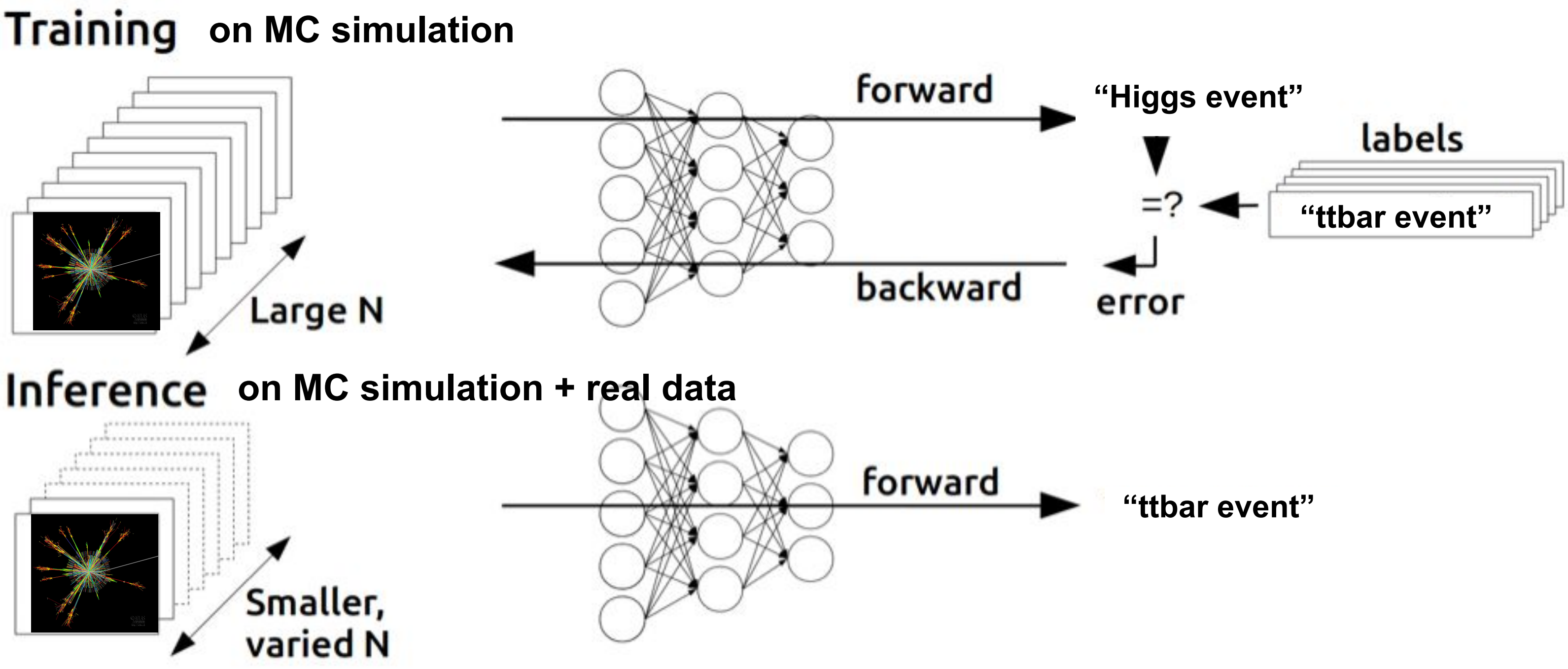
credit

Example of overfitting and regularization



credit

Recap

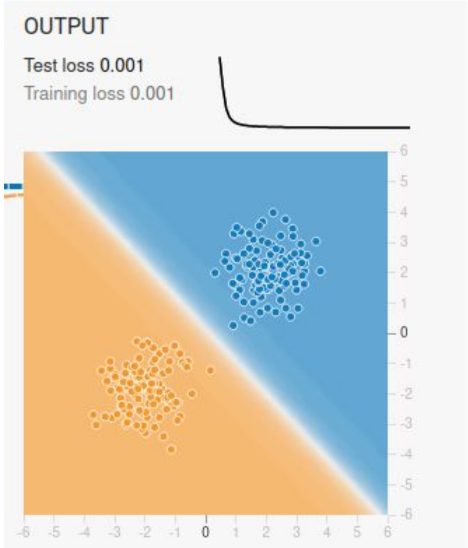


credit

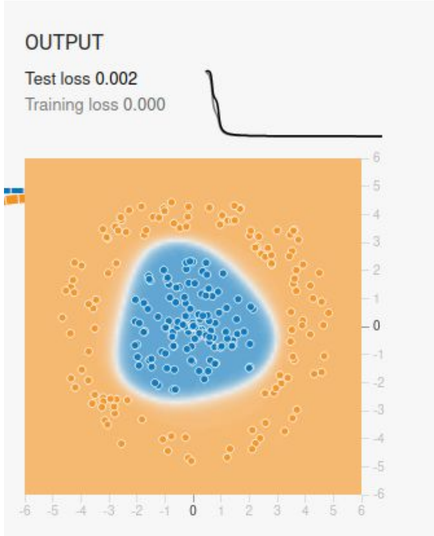
Tensorflow playground

https://playground.tensorflow.org

Two blobs



Circle



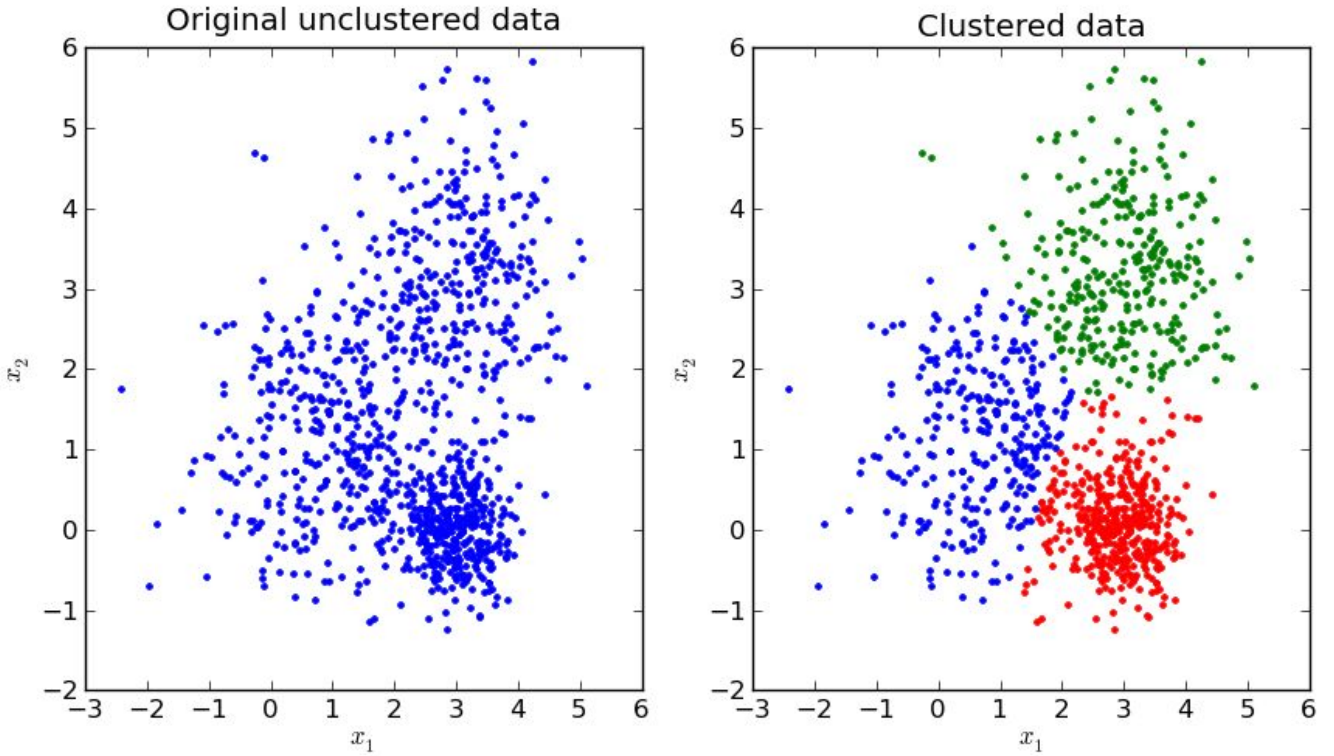
How many hidden neurons required to fit the “two blobs” dataset?

A. One
B. Two
C. Three

How many hidden neurons required to fit the circle dataset?

D. One
E. Two
F. Three

Clustering



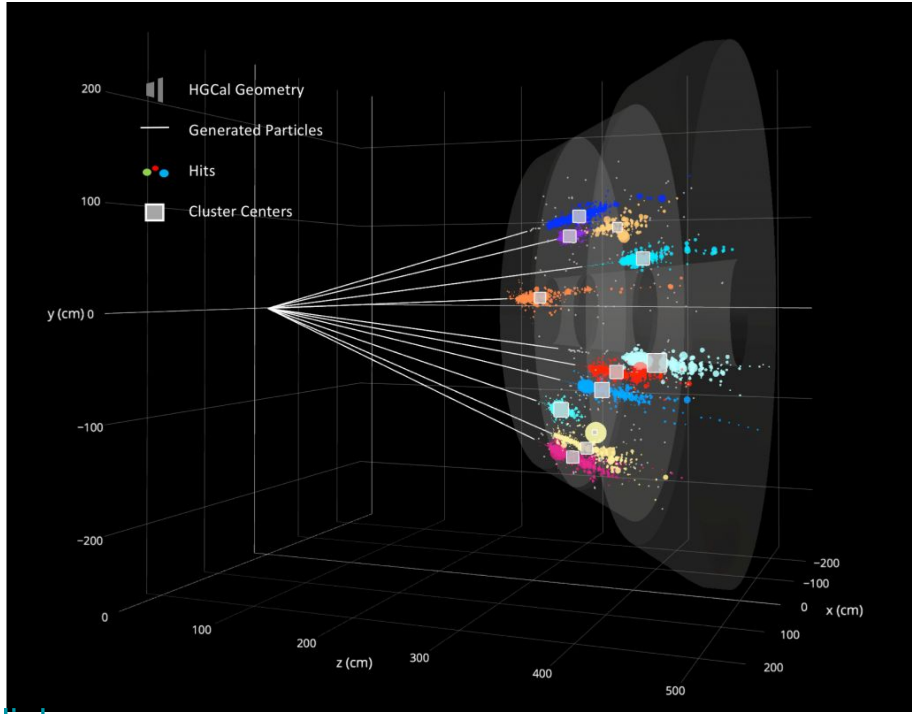
Reconstructing particle showers

Hits → clusters → particle candidates.

How many clusters do we expect?

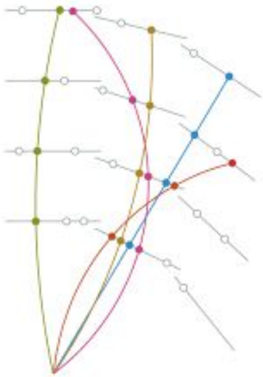
What's the "cost" of incorrectly merging/splitting a cluster?

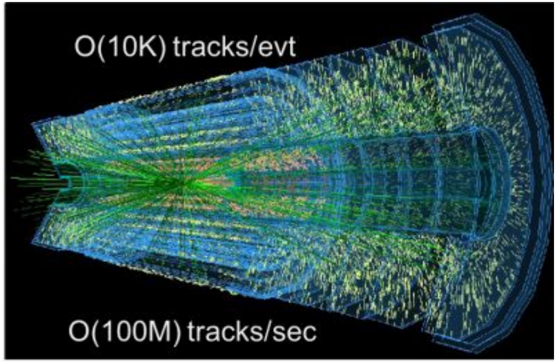
How do you determine the particle properties from the cluster?



link

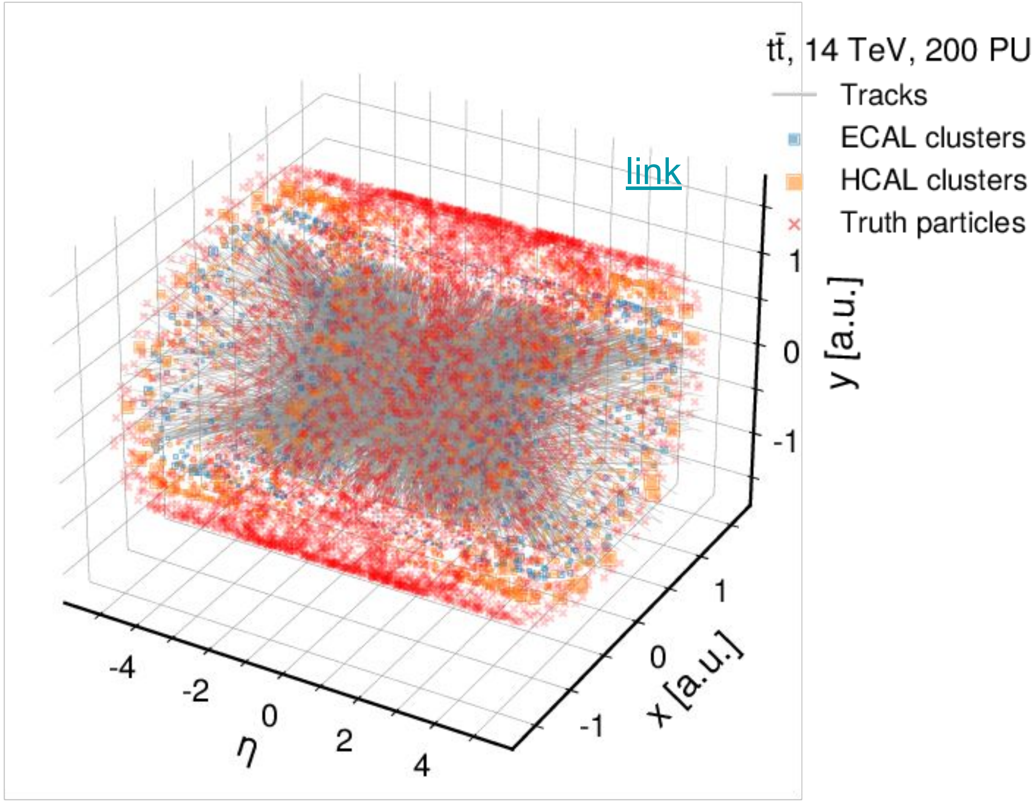
Particle tracking



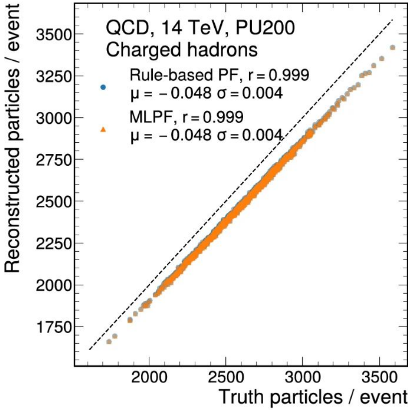


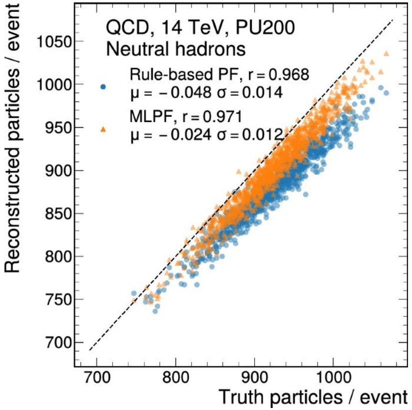
[link]

Reconstruction across different detector systems

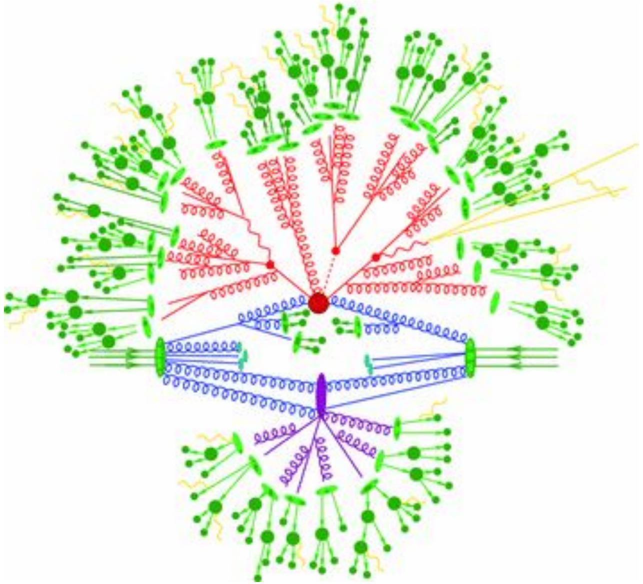


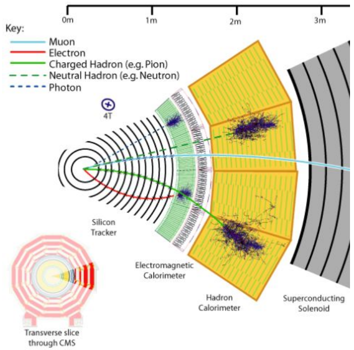
link

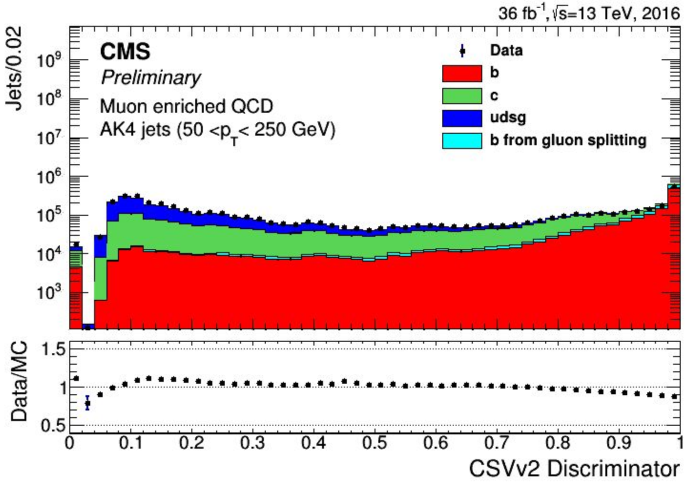




Generative modelling





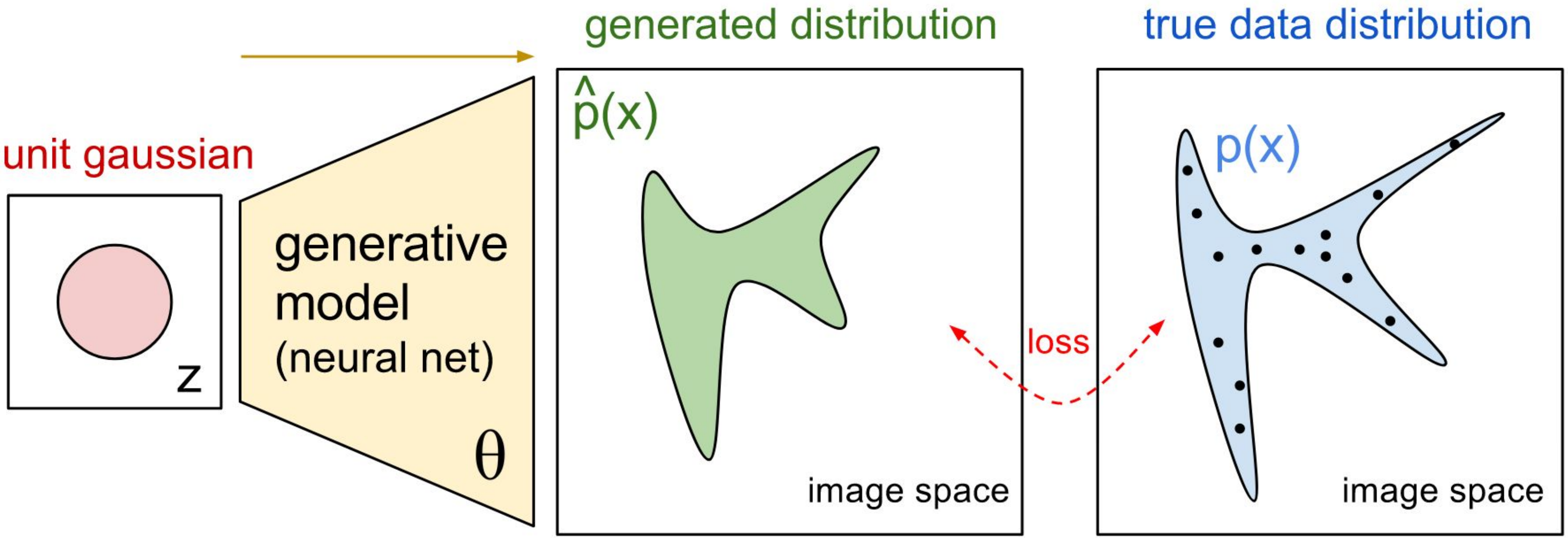


MC generation

Detector simulation

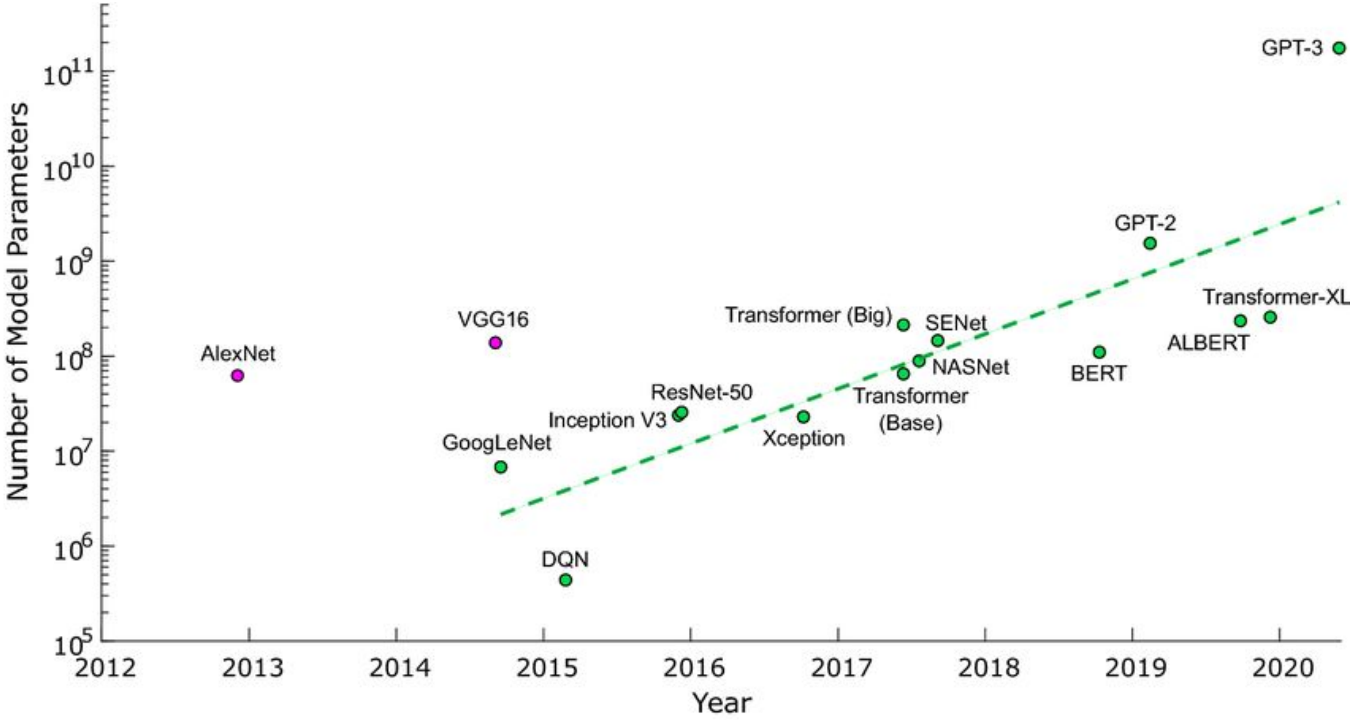
Comparison with data

Generative modeling with ML



https://openai.com/blog/generative-models/

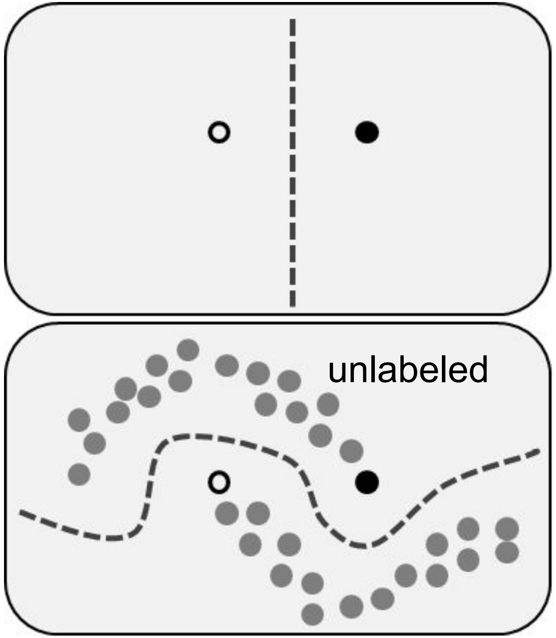
Overparameterization

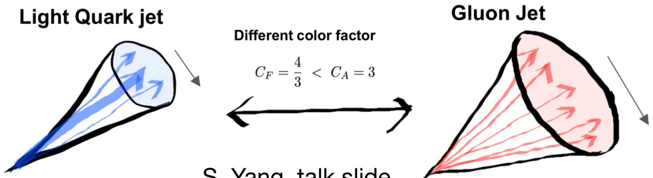


credit

Semi-supervised methods

Labeled examples
(e.g. simulation)





S. Yang, talk slide

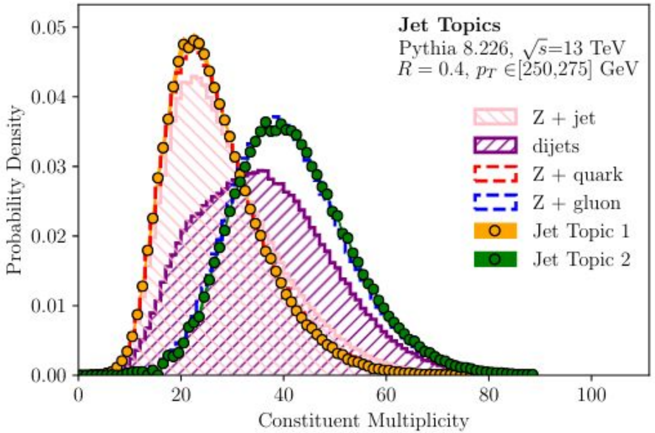


FIG. 2. The jet topics method applied to constituent multiplicity, starting with Z+jet (pink) and dijet (purple) distributions from PYTHIA 8.226. There is good agreement between the two extracted jet topics (orange and green) and pure Z+quark and Z+gluon distributions (red and blue).

1802.00008