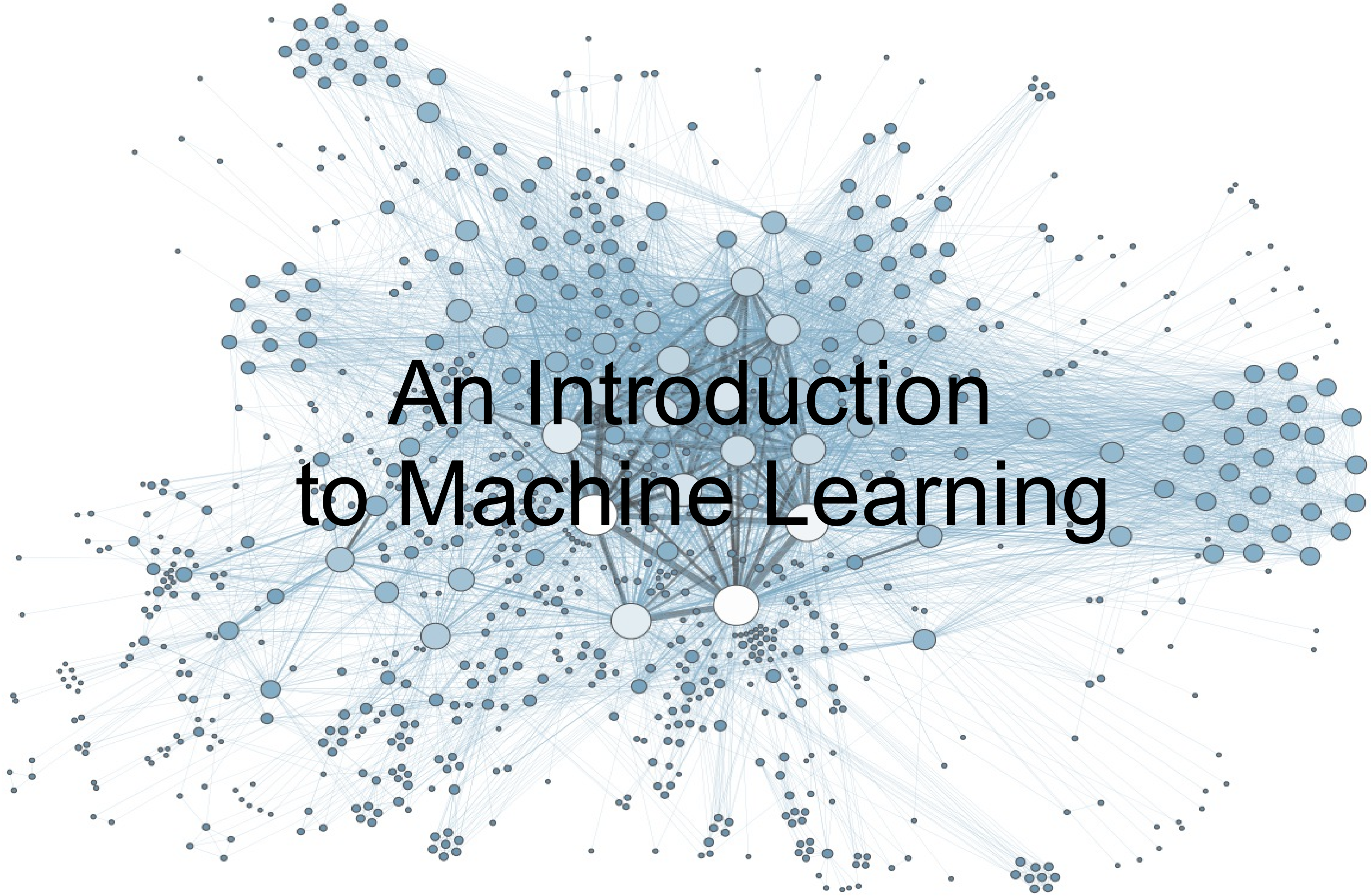
Machine Learning in High Energy Physics

CMS Induction Course
29-31 January 2020

Jean-Roch Vlimant
jvlimant@caltech.edu @vlimant

Outline

- An introduction to Machine Learning
- Motivations for Using ML in HEP
- 360º of pitfalls
- Usage in CMS
- R&D in CMS
- Outlooks



An Introduction to Machine Learning

What Is Machine Learning

“Giving computers the ability to learn without explicitly programming them” A. Samuel (1959).

Is fitting a straight line machine learning ?
Models that have enough capacity to define its own internal representation of the data to accomplish a task : **learning from data.**

In practice : a statistical method that can extract information from the data, not obviously apparent to an observer.

→ Most approach will involve a **mathematical model** and a cost/reward function that needs to be **optimized.**
→ The more **domain knowledge** is incorporated, the better.

Overview

Reinforcement Learning (cherry)
- The machine predicts a scalar reward given once in a while.
- **A few bits for some samples**

Supervised Learning (icing)
- The machine predicts a category or a few numbers for each input
- **10→10,000 bits per sample**

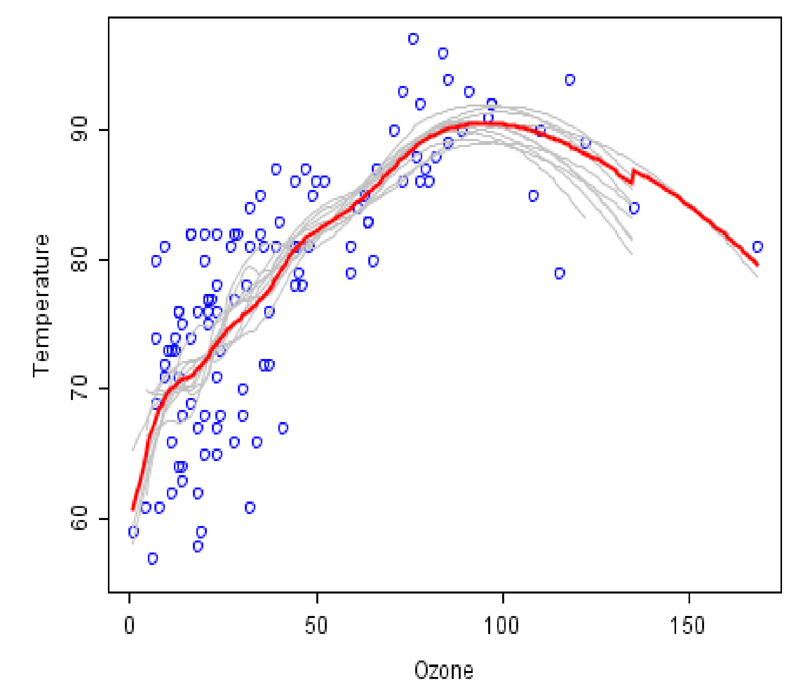
Unsupervised Learning (cake)
- The machine predicts any part of its input for any observed part.
- Predicts future frames in videos
- **Millions of bits per sample**

Yann Le cun, CERN 2016

Supervised Learning

- Given a dataset of samples, a subset of features is qualified as **target**, and the rest as **input**
- Find a **mapping from input to target**
- The mapping should **generalize to any extension** of the given dataset, provided it is generated from the same mechanism

$$dataset \equiv \{(x_i, y_i)\}_i$$
$$\textit{find function } f \textit{ s.t. } f(x_i) = y_i$$



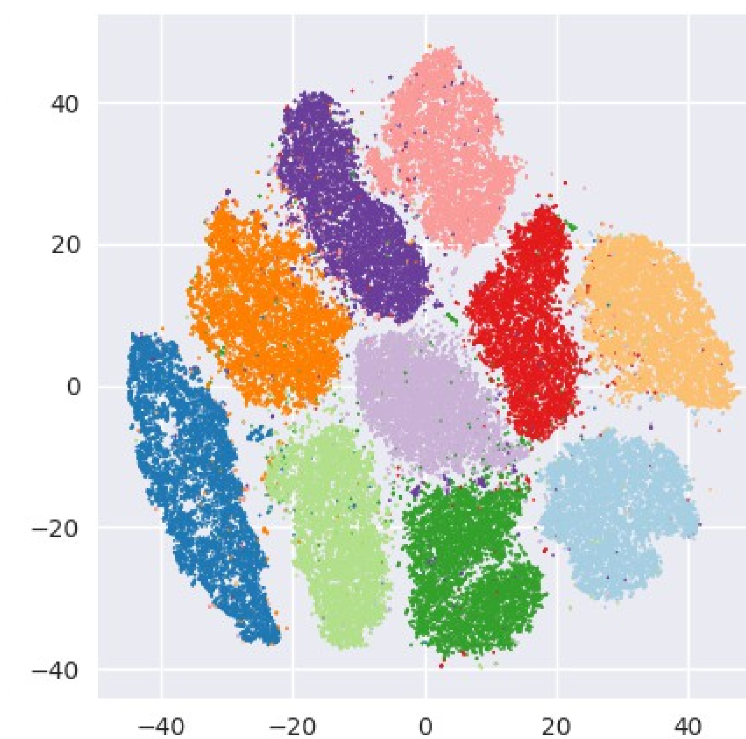
- Finite set of target values :
  - → **Classification**
- Target is a continuous variable :
  - → **Regression**

Unsupervised Learning

- Given a dataset of samples, but there is no subset of feature that one would like to predict
- Find mapping of the samples to a lower dimension manifold
- The mapping should generalize to any extension of the given dataset, provided it is generated from the same mechanism

$$dataset \equiv \{(x_i)\}_i$$
$$find\ \ f\ \ s.t.\ \ f(x_i) = p_i$$

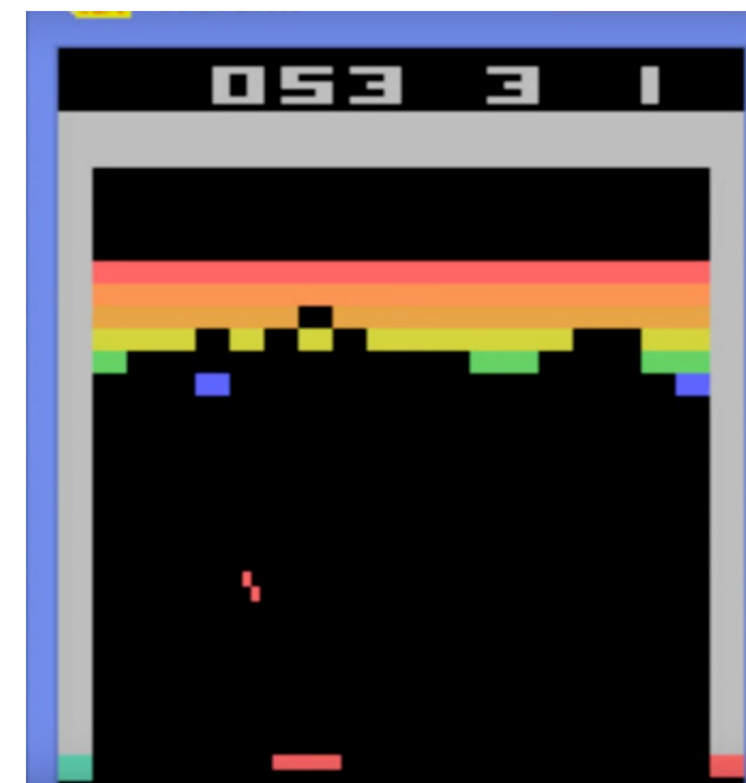
- Manifold is a finite set
  → **Clusterization**
- Manifold is a lower dimension manifold :
  → **Dimensionality reduction, density estimator**



Reinforcement Learning

- Given an **environment** with multiple states, given a reward upon action being taken over a state
- Find an **action policy to drive** the environment toward maximum cumulative reward

$$s_{t+1} = Env(s_t, a_t)$$
$$r_t = Rew(s_t, a_t)$$
$$\pi(a|s) = P(A_t = a | S_t = s)$$
$$find\ \pi\ s.t.\ \sum_t r_t\ is\ maximum$$



Motivation

Classical (read not deep-learning) machine learning has been around for long and **used at many level** in science.

Artificial neural network : a.k.a "Deep learning" is now very present in data-science thanks to :

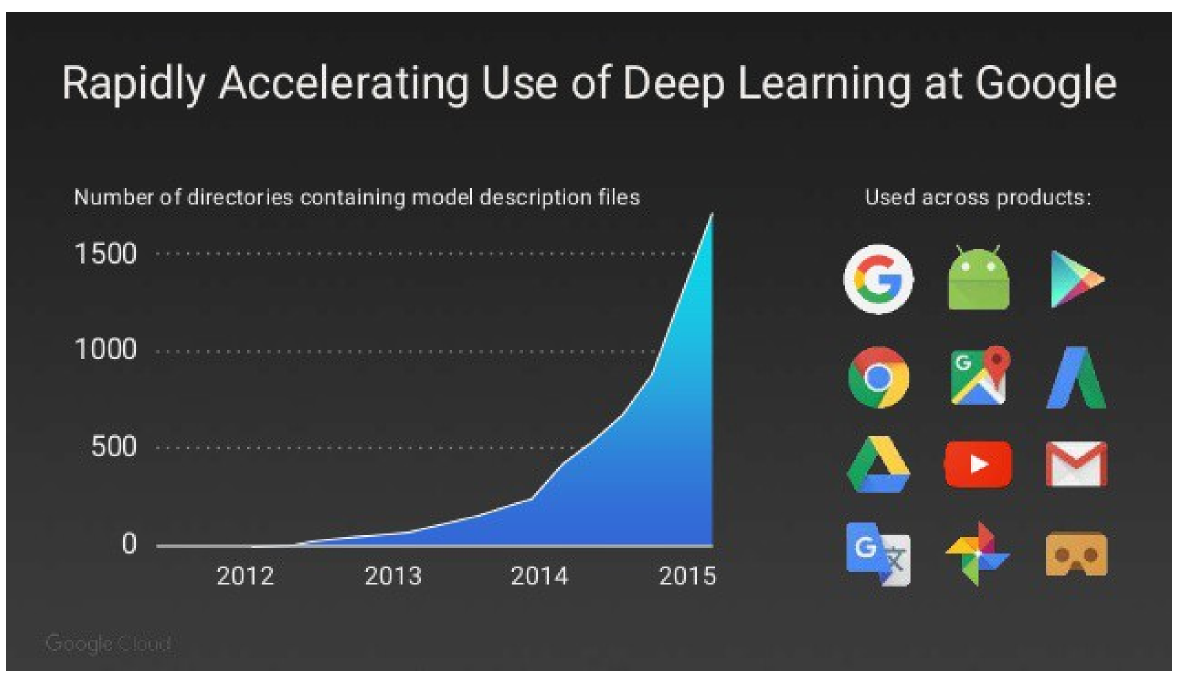
- Increased **computation power** through general purpose graphical processing units (GP-GPU)
- Increased **dataset size** through the internet-of-things (IOT)
- Improved **models architectures** (relu activation, convolution, …)

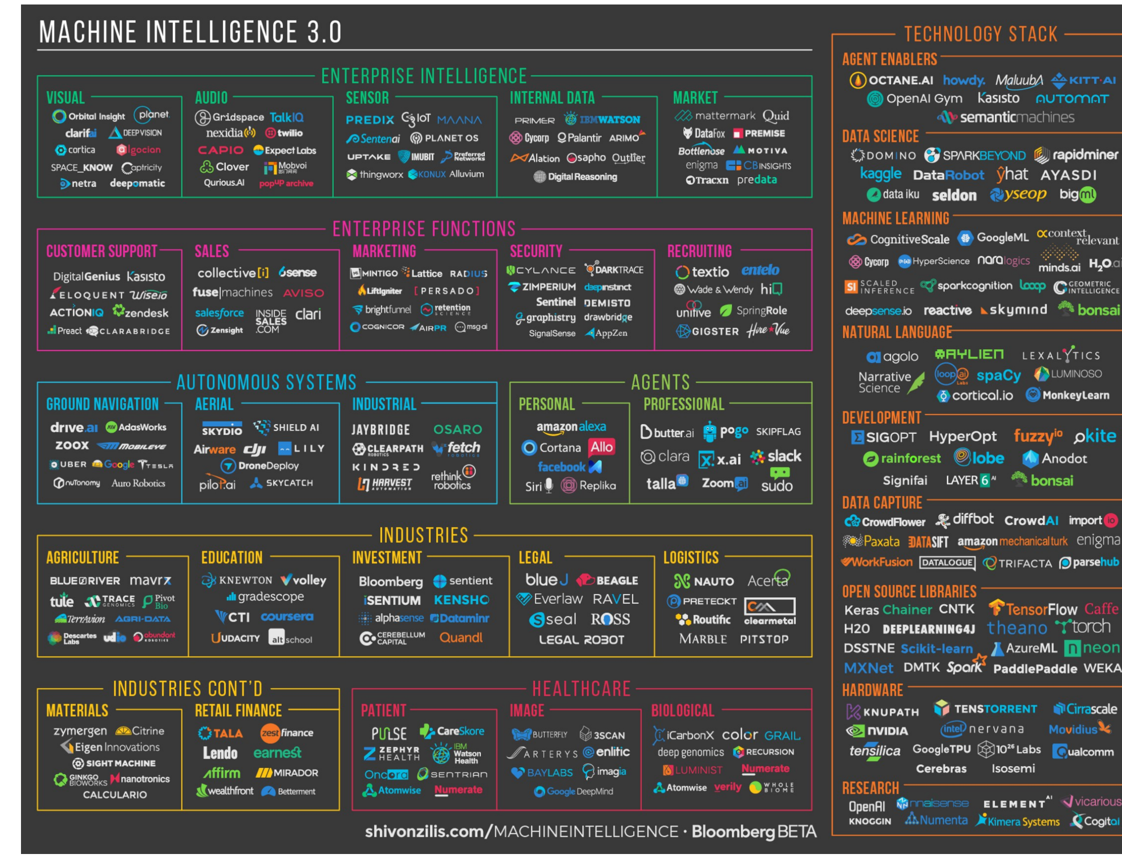
→ It became possible to **train models with millions of parameters** on dataset with **millions of samples**, each with **multiple thousands of pixels**

→ It became possible to **extract very complex correlations**, otherwise cumbersome to model.

Machine Learning in Industry

https://www.nvidia.com/en-us/deep-learning-ai/



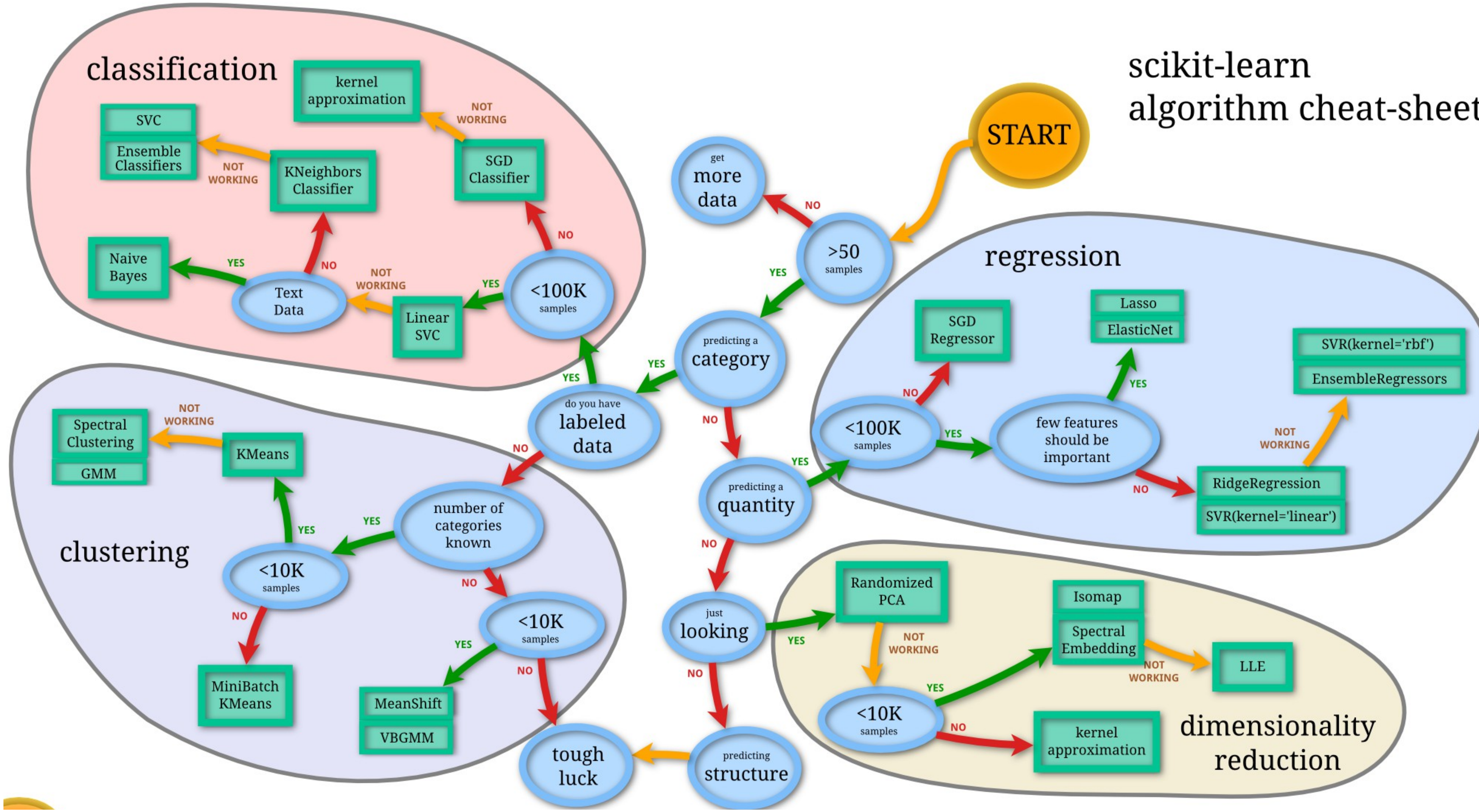


http://www.shivonzilis.com/machineintelligence

- Prominent field in industry nowadays
- Lots of data, lots of applications, lots of potential use cases, lots of money

➔ Knowing machine learning can open significantly your **career horizons**

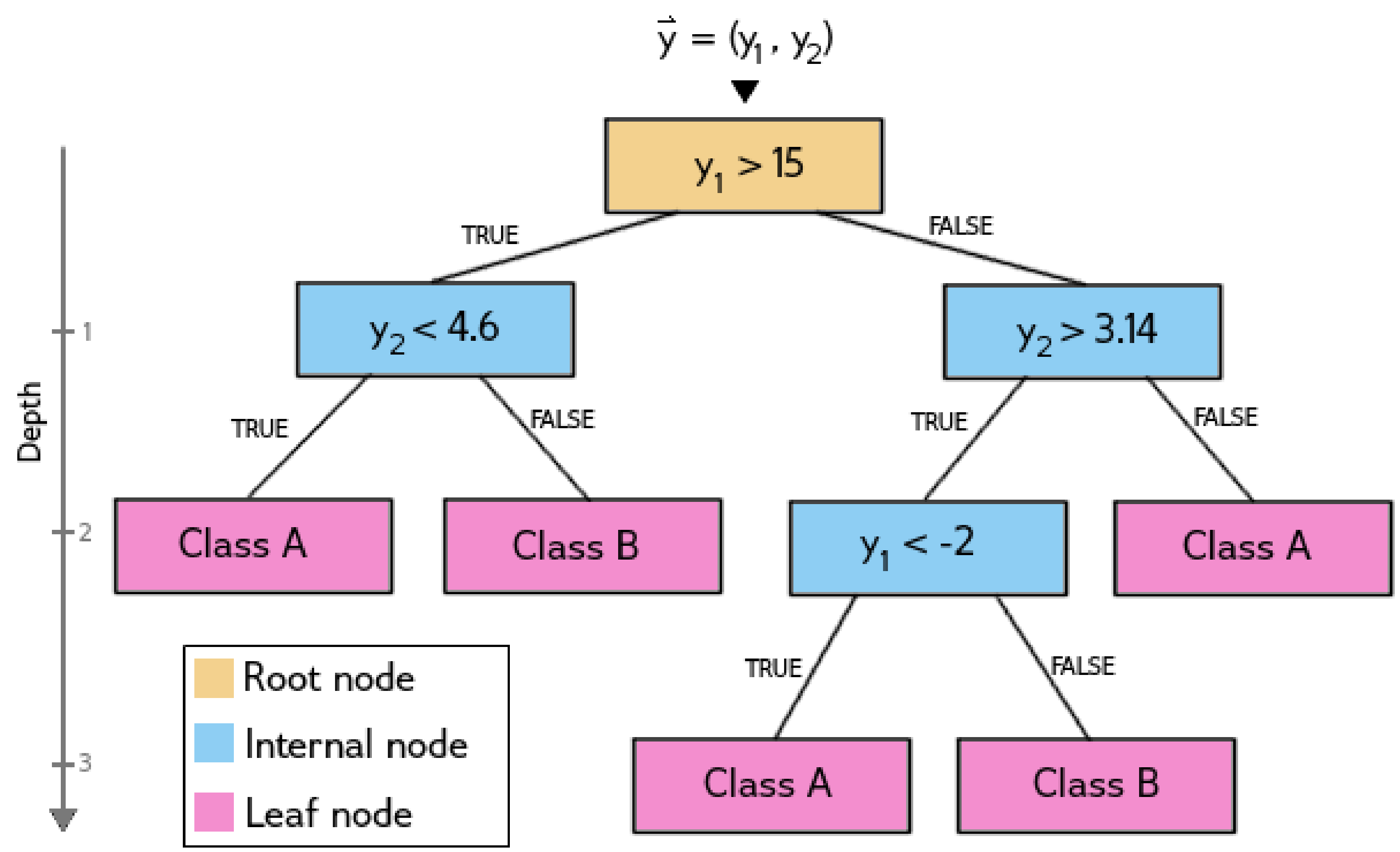
(Some) Machine Learning Methods



http://scikit-learn.org/stable/tutorial/index.html

Decision Tree

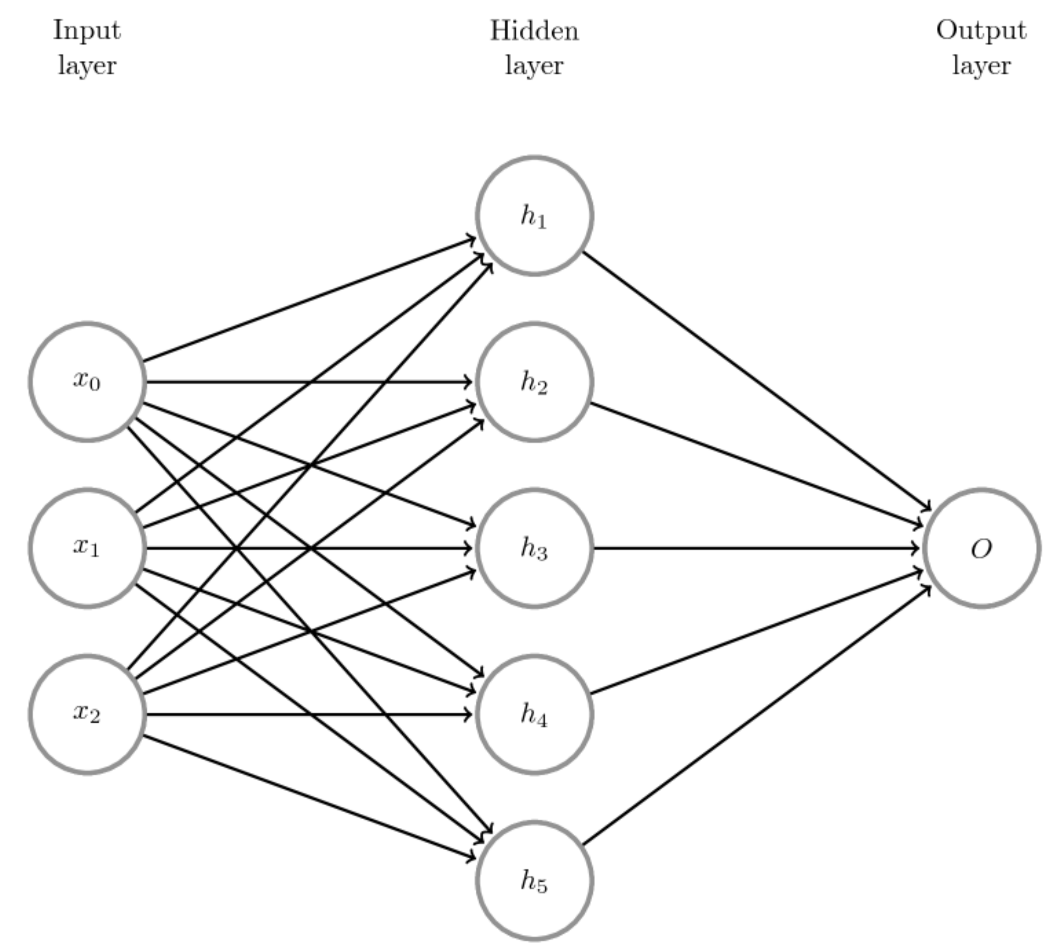
- Decision trees is a well known tool in supervised learning.
- It has the advantage of being easily interpretable
- Can be used for classification or regression



Artificial Neural Network

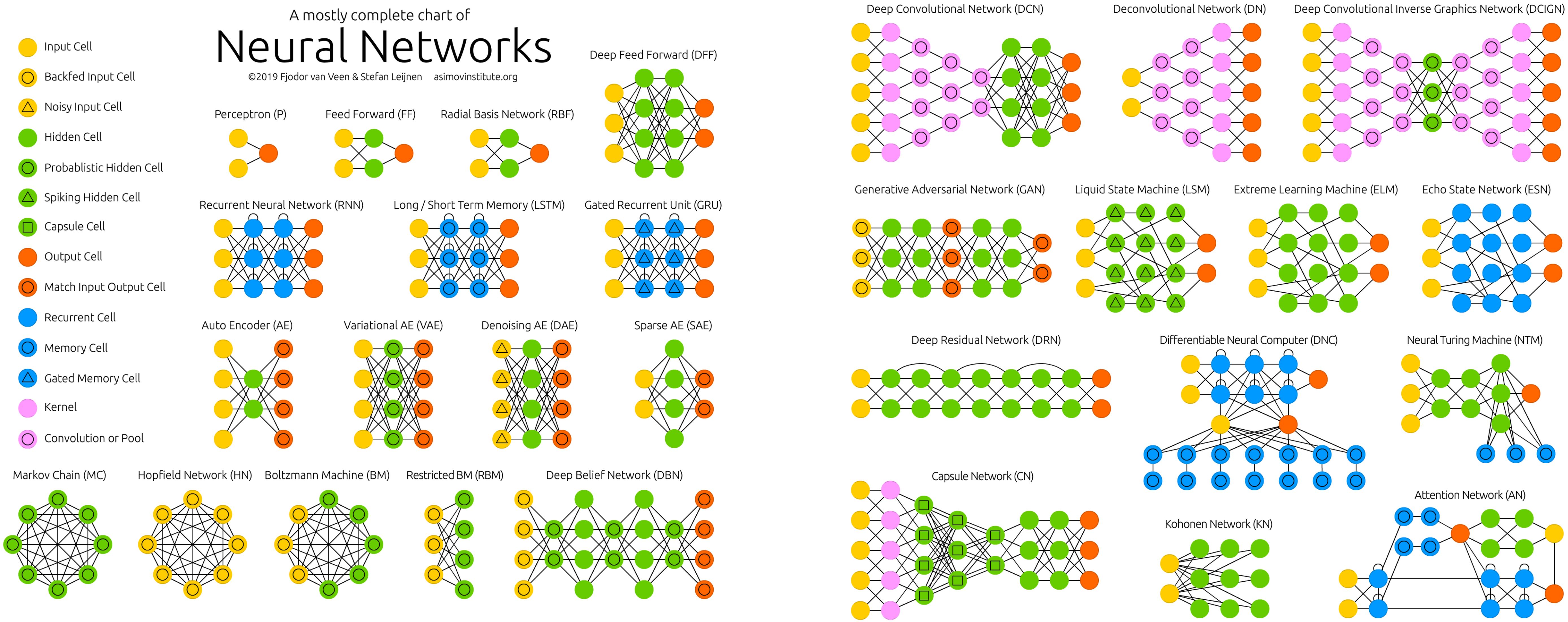
- **Biology inspired** analytical model, but **not bio-mimetic**
- Booming in recent decade thanks to large dataset, increased computational power and theoretical novelties
- Origin tied to logistic regression with change of data representation
- Part of any “deep learning” model nowadays
- Usually large number of parameters trained with stochastic gradient descent

$$h = \phi(Ux + v)$$
$$o(x) = \omega^T h + b$$
$$p_i \equiv p(y=1|x) \equiv \sigma(o(x)) = \frac{1}{1+e^{-o(x)}}$$
$$loss_{XE} = -\sum_i y_i \ln(p_i) + (1-y_i)\ln(1-p_i)$$



Neural Net Architectures

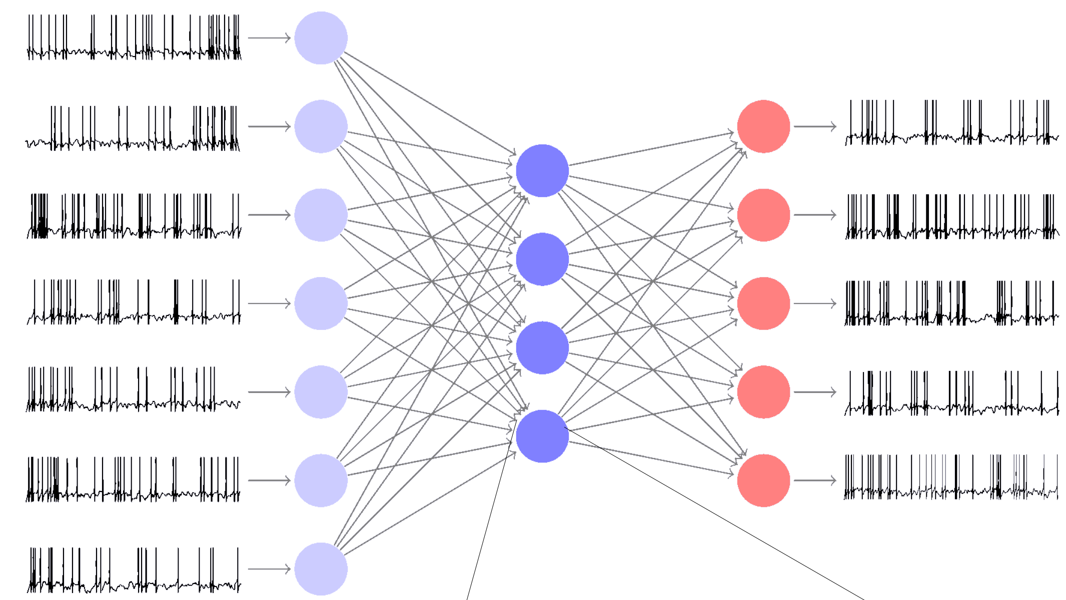
http://www.asimovinstitute.org/neural-network-zoo

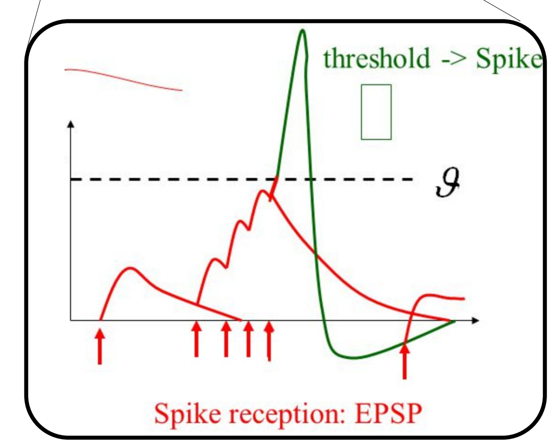


- Does not cover it all : densenet, graph network, ...

Spiking Neural Network

- Closer to the actual biological brain
- Adapted to temporal data
- Hardware implementation with low power consumption
- Trained using evolutionary algorithms
- Economical models





	Deep Learning	Spiking
Training Method	Back-propagation	Not well established (here, genetic algorithms)
Native Input Types	Images/Arrays of values	Spikes
Network Size	Large (many layers, many neurons and synapses per layer)	Relatively small (fewer neurons and sparser synaptic connections)
Processing Abilities	Good for spatial	Good for temporal
Performance	Well understood and state-of-the-art	Not well understood

Take Home Message

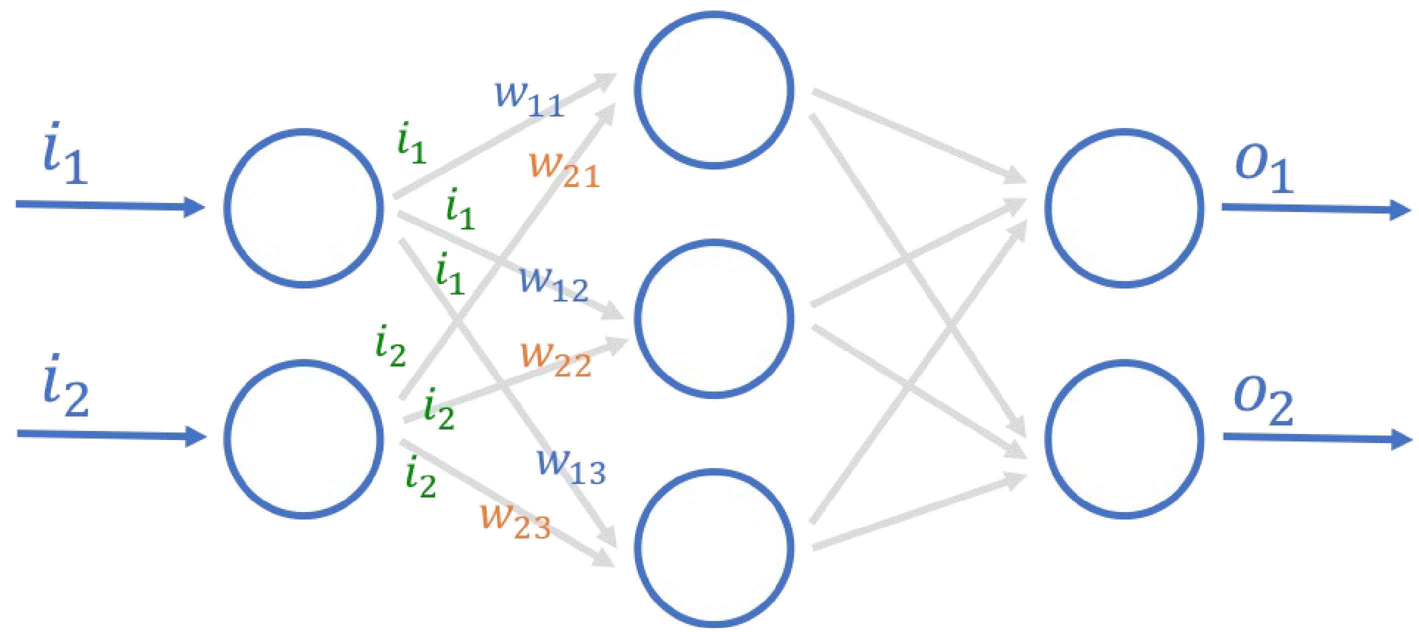
Machine learning helps with tasks on complex dataset, with complex/unknown correlations.

Large potential for applications in science.

Desired knowledge/skill for career path in industry.

Motivations for Using Machine Learning in High Energy Physics

Operation Vectorization

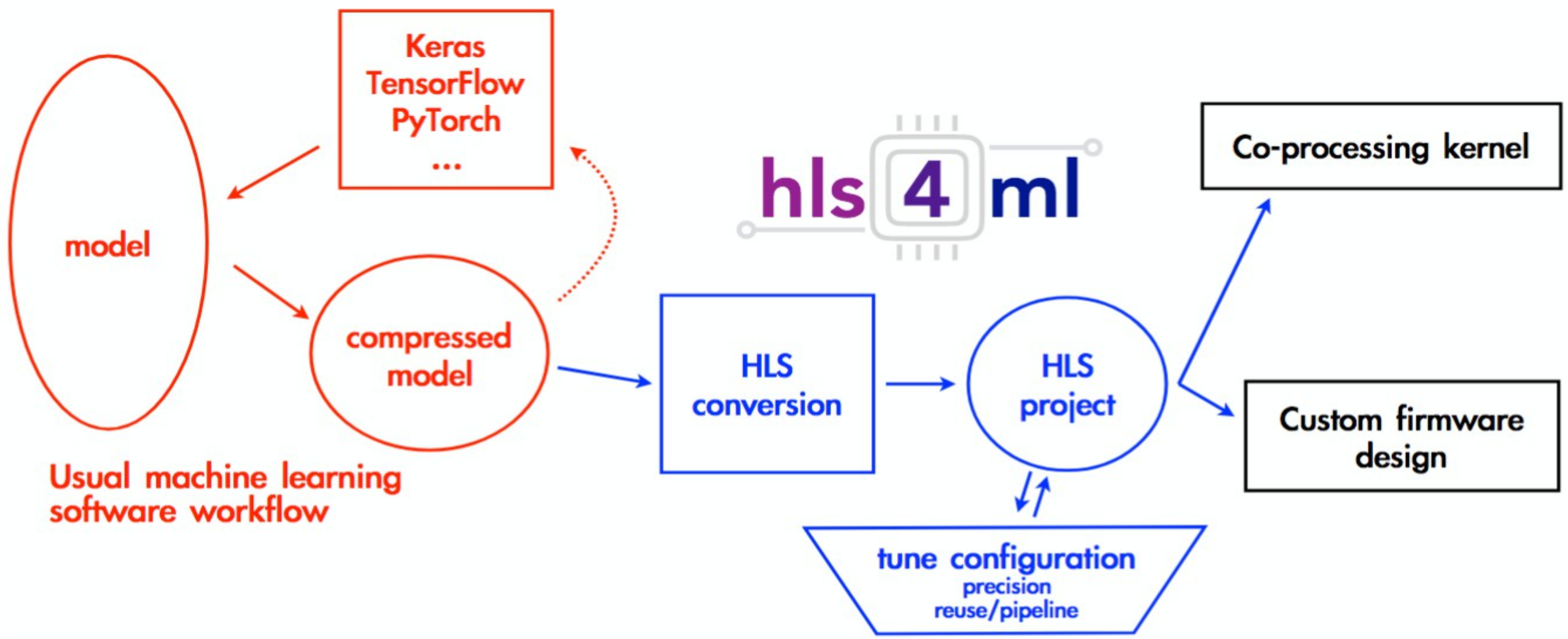


ANN ≡ matrix operations ≡ parallelizable

$$\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \\ w_{13} & w_{23} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} (w_{11} \times i_1) + (w_{21} \times i_2) \\ (w_{12} \times i_1) + (w_{22} \times i_2) \\ (w_{13} \times i_1) + (w_{23} \times i_2) \end{bmatrix}$$

Computation of prediction from artificial neural network model can be **vectorized to a large extend.**

Hyper-Fast Prediction



Synthesizing FPGA firmware from trained ANN

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

J. Duarte et al.https://arxiv.org/abs/1804.06913

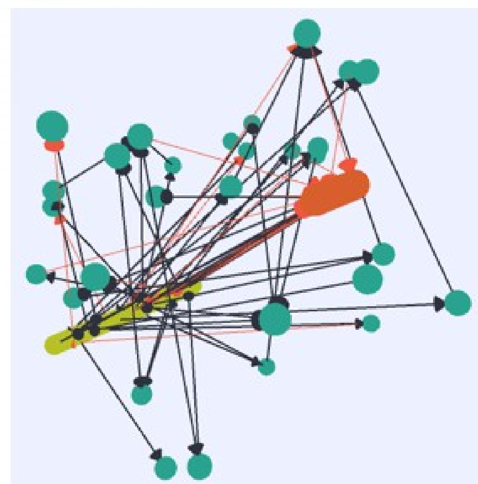
Prediction from artificial neural network model can be **done on FPGA**, GPU, TPU, ...

Low Power Prediction

Best Results: Single View

x-view (127x50)	conv1 (8x3)	pool1 (2x1)	conv2 (7x3)	pool2 (2x1)	conv3 (6x3)	pool3 (2x1)	conv4 (6x3)	pool4 (2x1)	fc1 (196)	drop out	fc2 (98)	drop out	fc3 (11)	classification

Convolutional Neural Network Result: ~80.42%



- 90 neurons, 86 synapses
- Estimated energy for a single classification for mrDANNA implementation: 1.66 μJ

Spiking Neural Network Result: ~80.63%

Source for CNN results: A. Terwilliger, et al. Vertex Reconstruction of Neutrino Interactions using Deep Learning. IJCNN 2017.

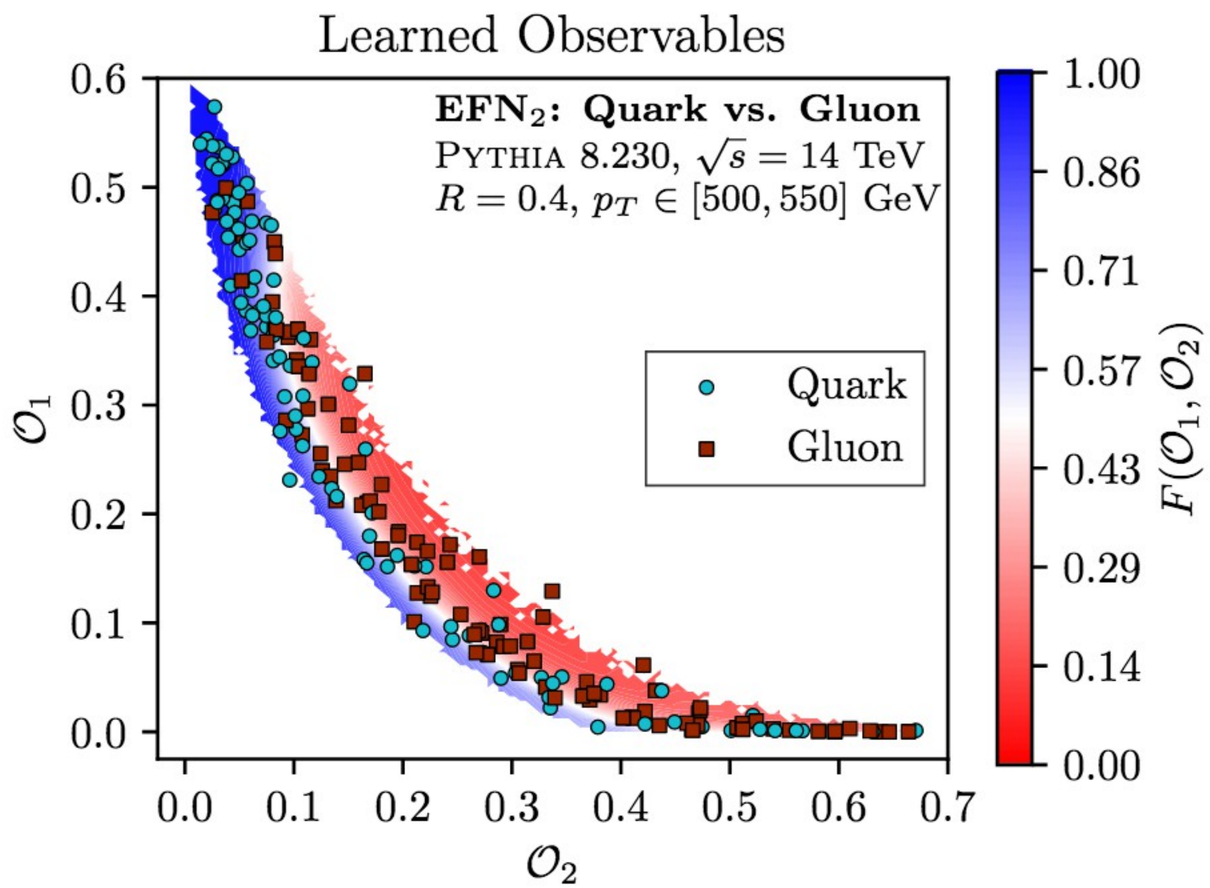
33 Programming Neuromorphic Computing Systems

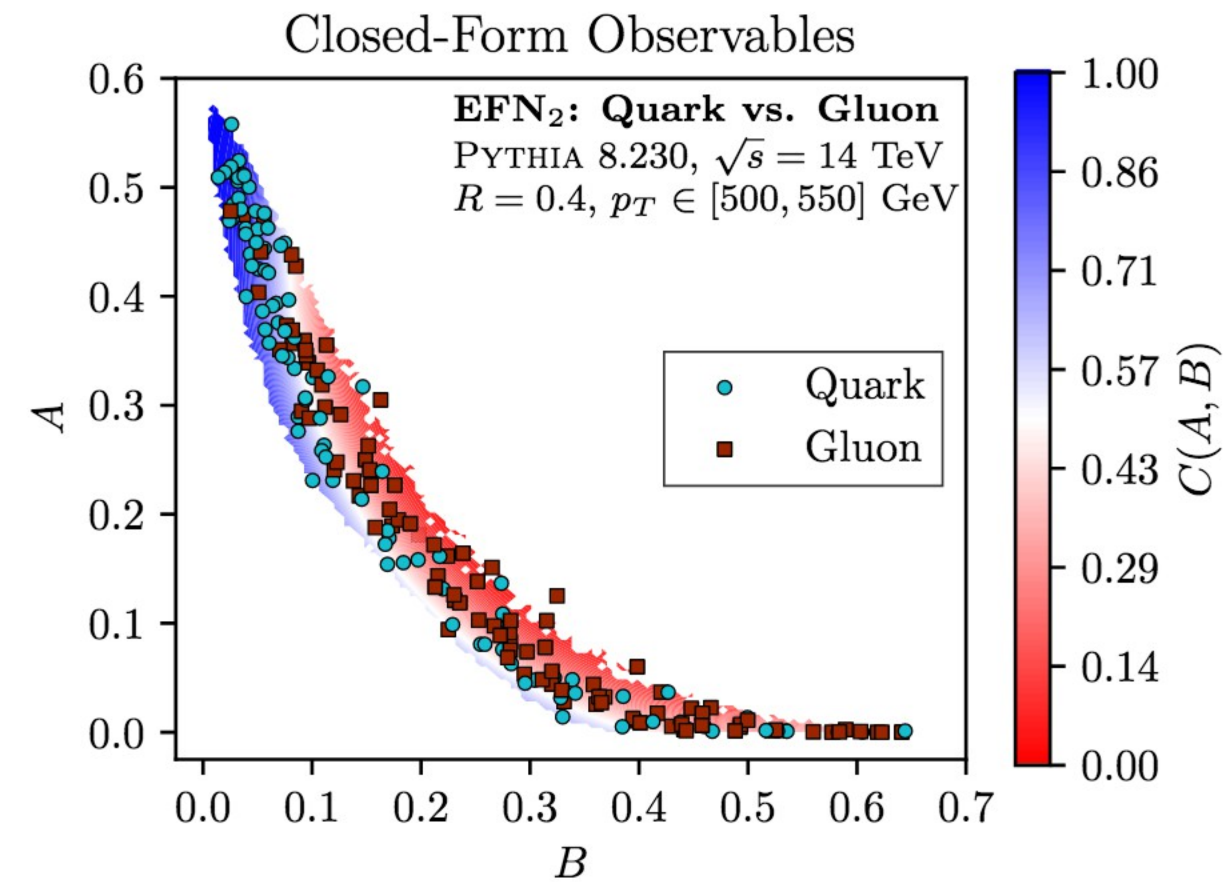
OAK RIDGE National Laboratory

https://indico.fnal.gov/event/13497/contribution/0 *Slide C. Schuman*

Neuromorphic hardware dedicated to spiking neural networks.
Low power consumption by design.

Physics Knowledge

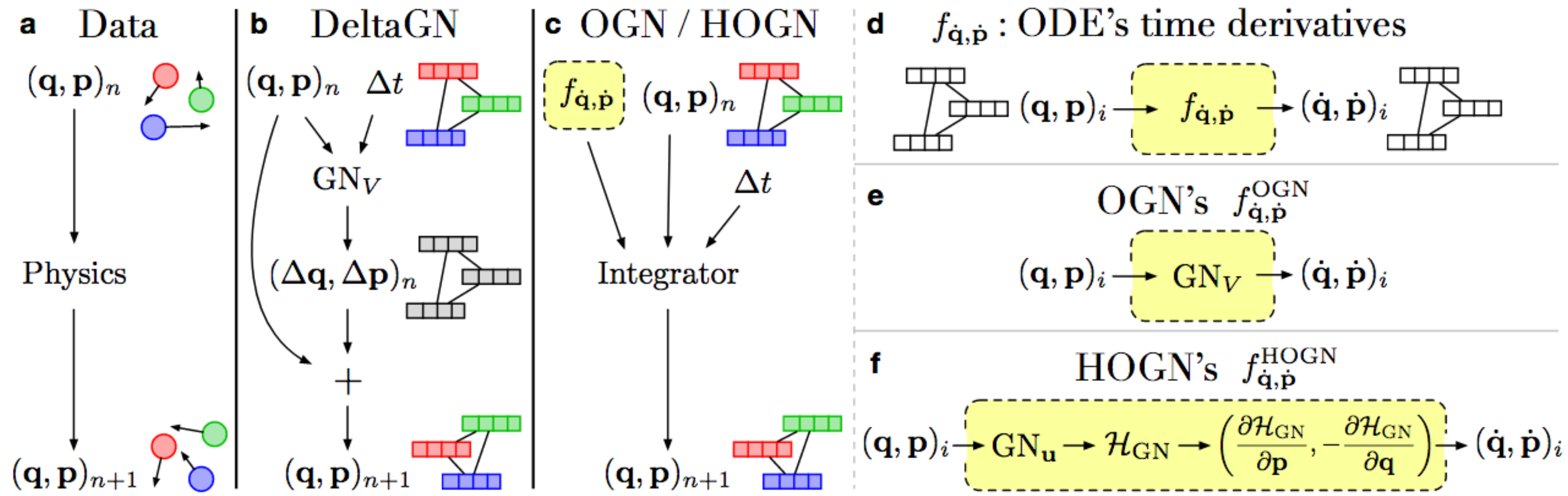


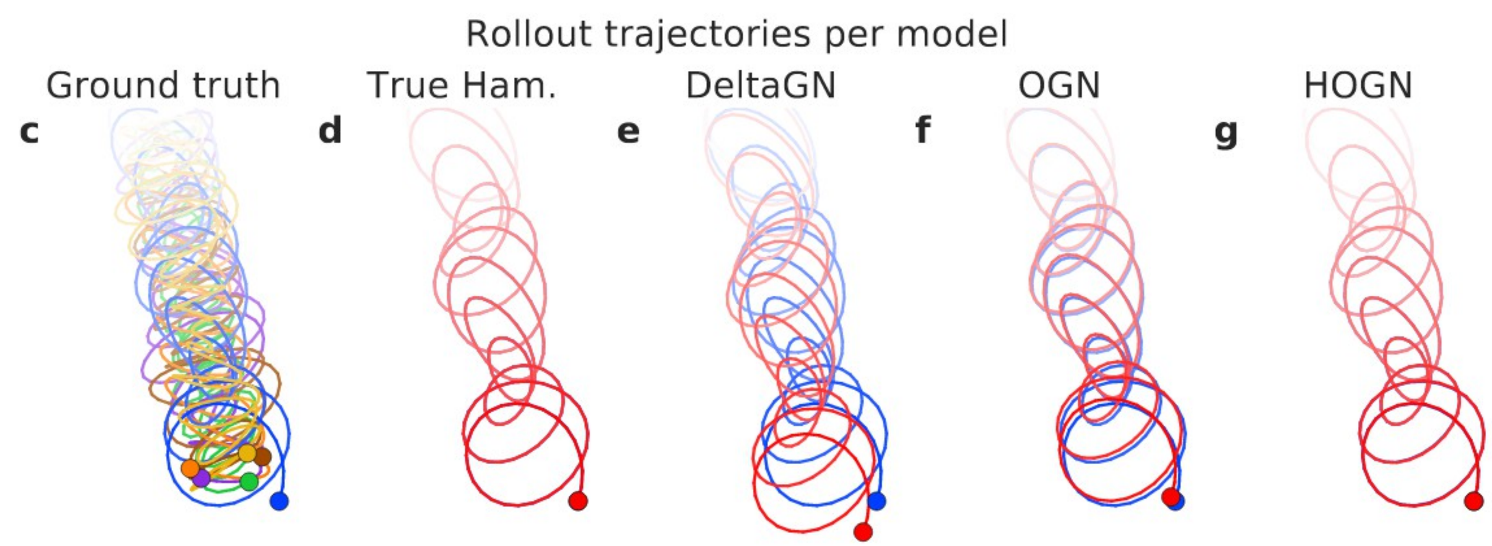


P. Komiske, E. Metodiev, J. Thaler, https://arxiv.org/abs/1810.05165

Machine Learning can **help understand Physics**.

Use Physics

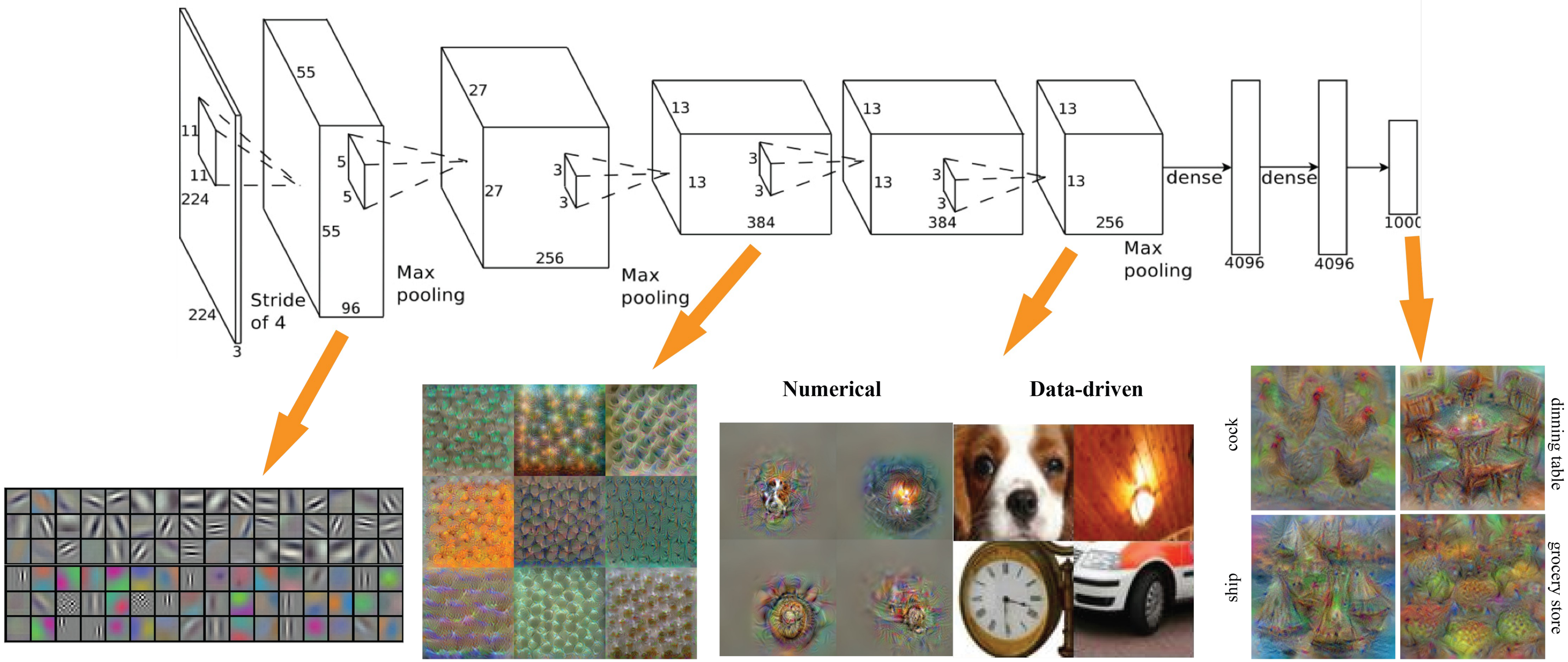




A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, P. Battaglia https://arxiv.org/abs/1909.12790

Let me model **include Physics principles** to master convergence

Learning from Complexity



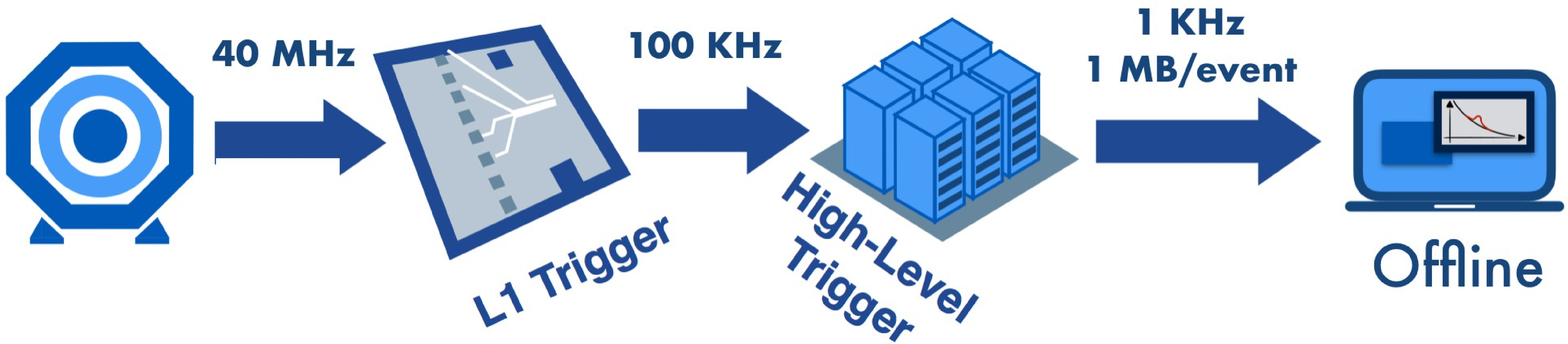
Conv 1: Edge+Blob Conv 3: Texture Conv 5: Object Parts Fc8: Object Classes

"Simple" machine learning model can **extract information from complex dataset.**

More classical algorithm counter part may take **years of development.**

Event Triggering

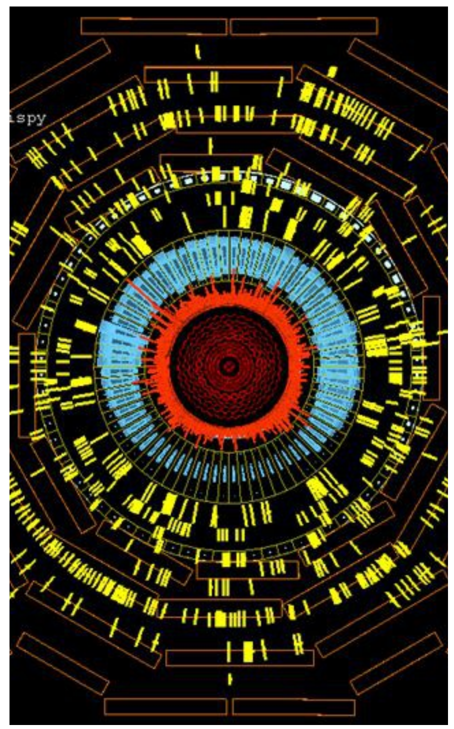
Select what is important to keep for analysis.
Ultra fast decision in hardware and software.

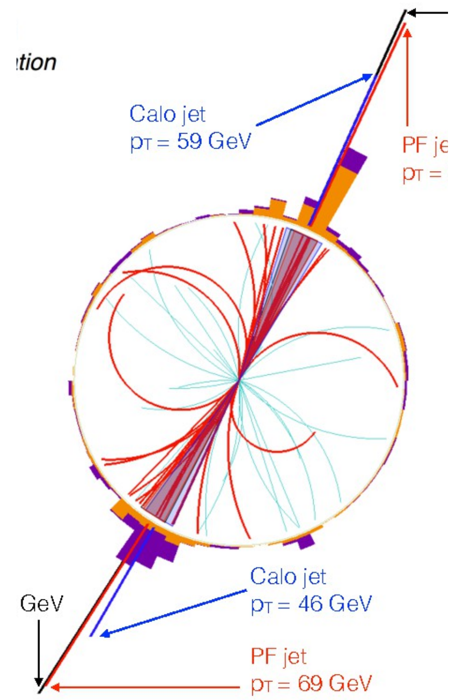


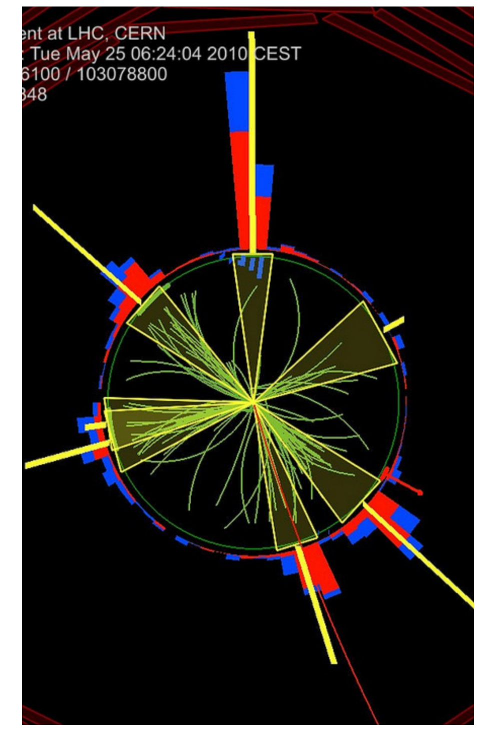
Reconstruction(s) of the event under limited latency.
Better resolution help lowering background trigger rates.
Enabling approximate, deep learning algorithms can help.

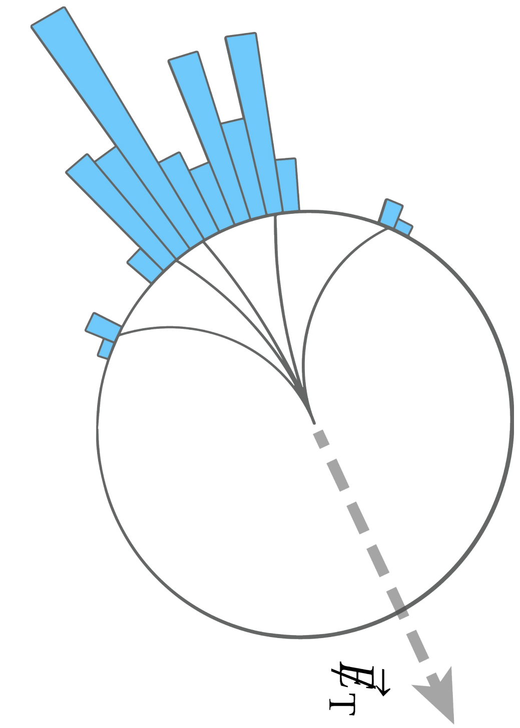
From RAW to High Level Features

Detector Data | Local reconstruction | Particle representation | Jet Clustering | High level features









Event Processing

Dimensionality reduction

Globalization of information

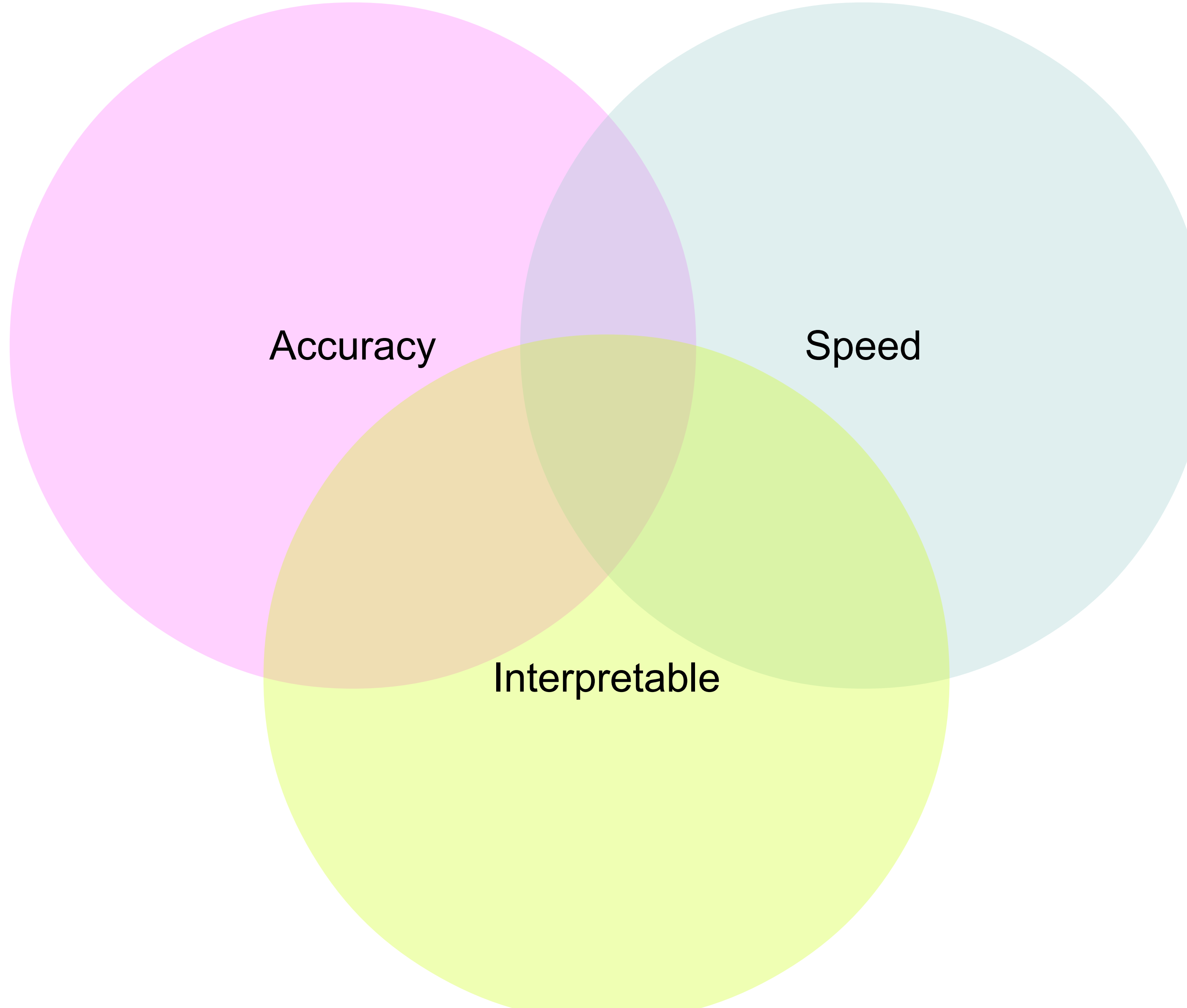
From digital signal, to local hits, to a sequence of particles, jets, and high-level features.
Complex and computing intensive task that could find a match in ML application.

Simulating Collisions

Madgraph, Pythia, Sherpa, ...	**Event Generator**: compute predictions of the standard models to several orders of expansion in coupling constants (LO, NLO, NNLO, ...) using proton density functions.
Pythia, ...	**Hadronization**: phenomenological model of the evolution of hadrons under the effect of QCD.
GEANT 4, GEANT V	**Material simulator**: transports all particles throughout meters of detector, using high resolution geometrical description of the materials.
Homegrown software	**Electronic emulator:** converts simulated energy deposits in sensitive material, into the expected electronic signal, including noise from the detector.

Non-differentiable sequence of complex simulators of the signal expected from the detectors.
Computing intensive task, exceeding budget for reconstruction.

Possible Utilizations



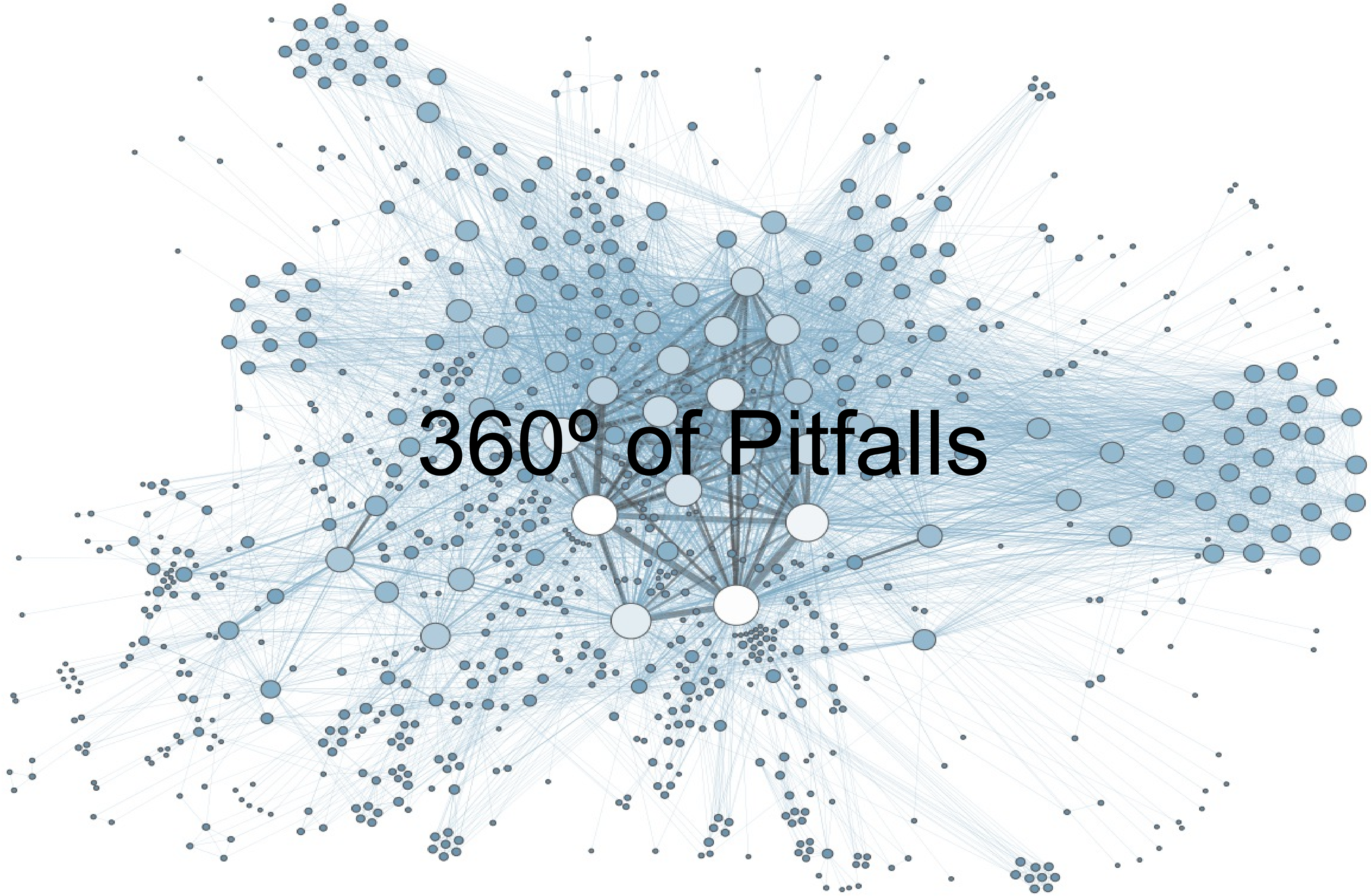
- ➔ **Fast surrogate** models (trigger, simulation, ...) for computing restricted algorithms.
- ➔ Model **more accurate** than existing algorithms (tagging, ...)
- ➔ Model performing **otherwise impossible tasks** (operations, ...)

Take Home Message

Model prediction can be fast and help with computing restrictions.

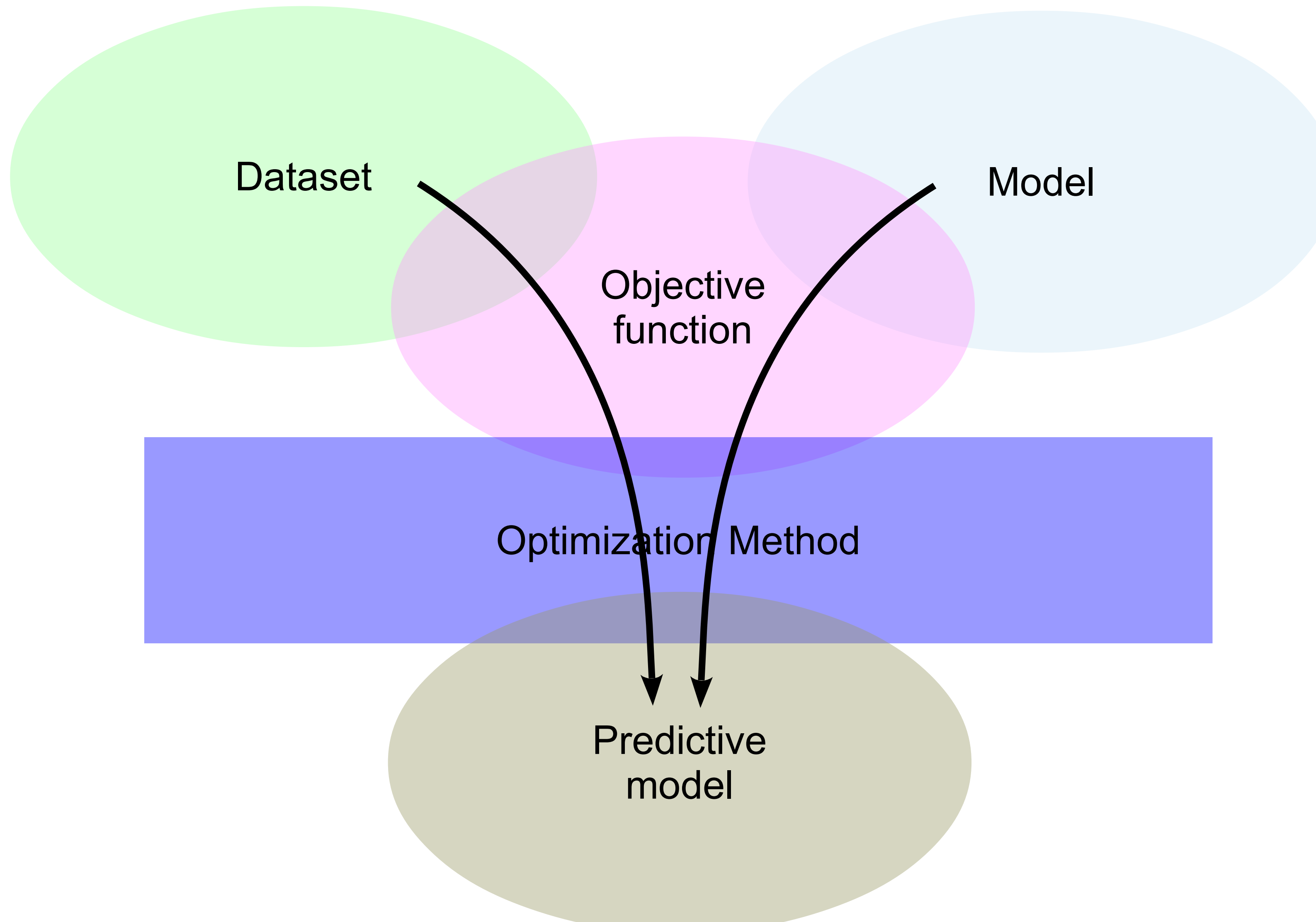
HEP data representation is multi-trait and match with wide range of existing tools.

Machine learning can help with extracting better physics knowledge from data.



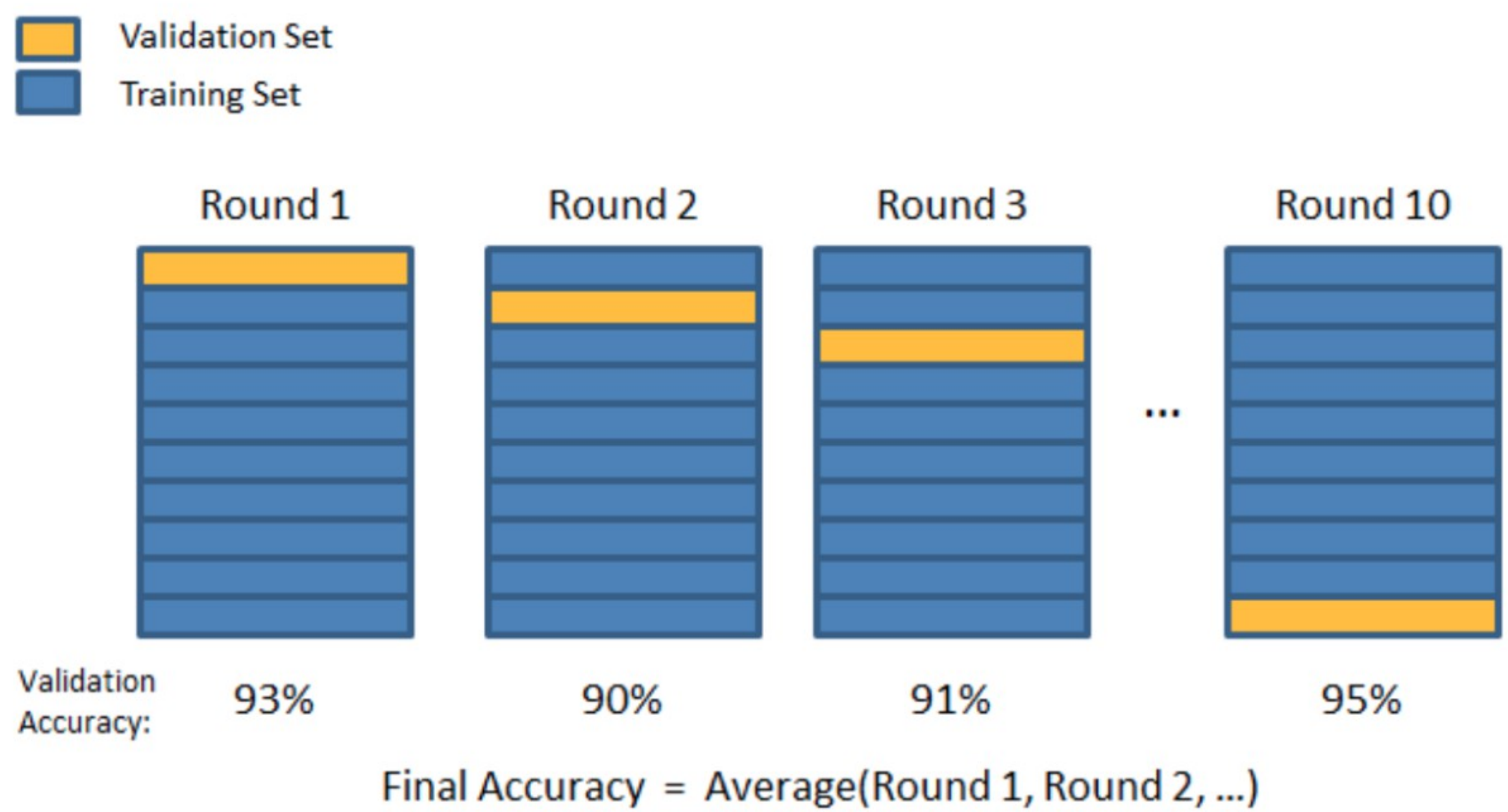
360º of Pitfalls

Machine Learning Concept



All comes down to an optimization problem.
What follows are some of the things to keep an eye on when developing a machine learning solution

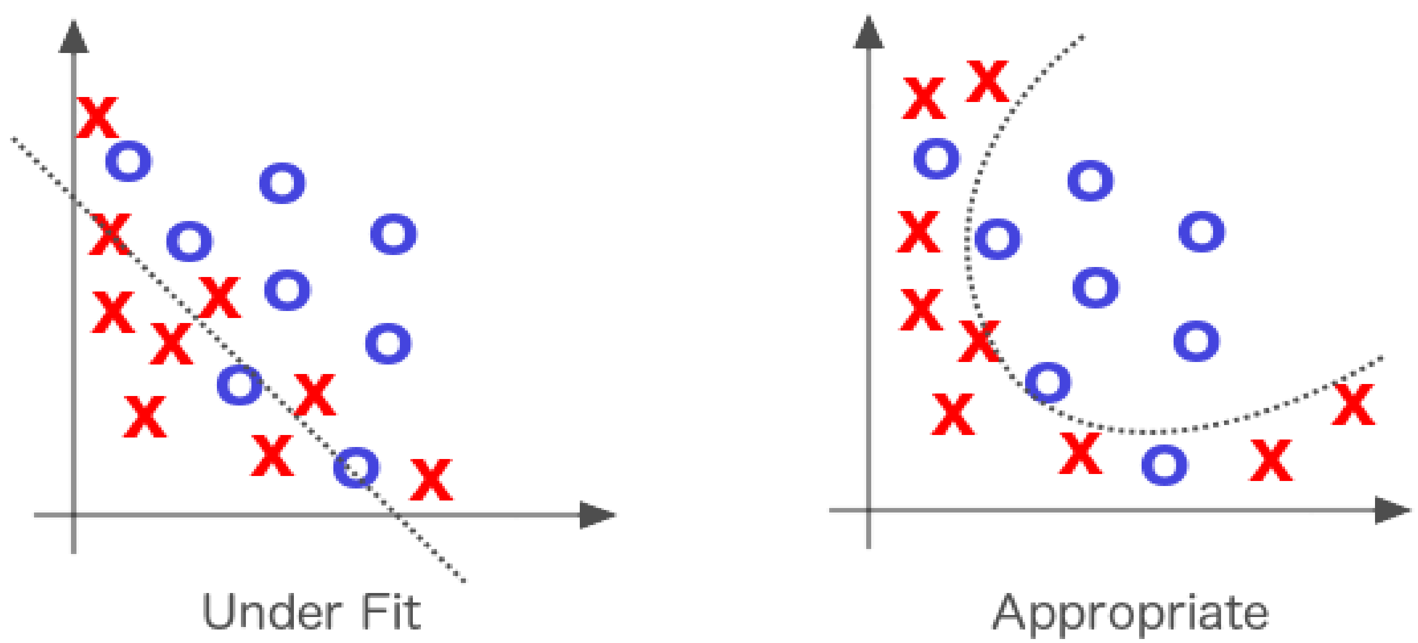
Cross Validation



- Model selection requires to have an estimate of the uncertainty on the metric used for comparison
  - K-folding provides an un-biased way of comparing models
- Stratified splitting (conserving category fractions) protects from large variance coming from biased training
- Leave-one-out cross validation : number folds ≡ sample size

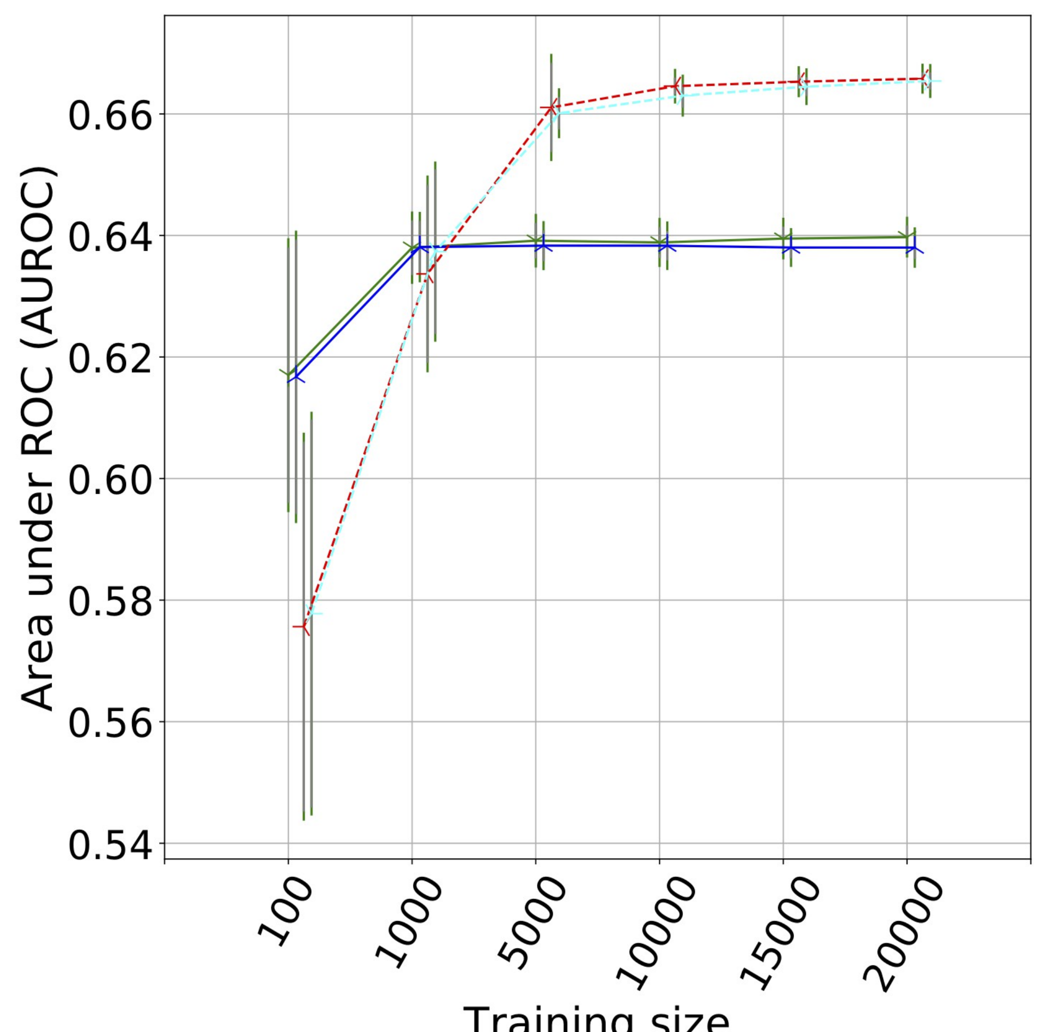
Under-fitting

- Poor model performance can be explained
  - Lack of modeling capacity (not enough parameters, inappropriate parametrization, …)
  - Model parameters have not reached optimal values



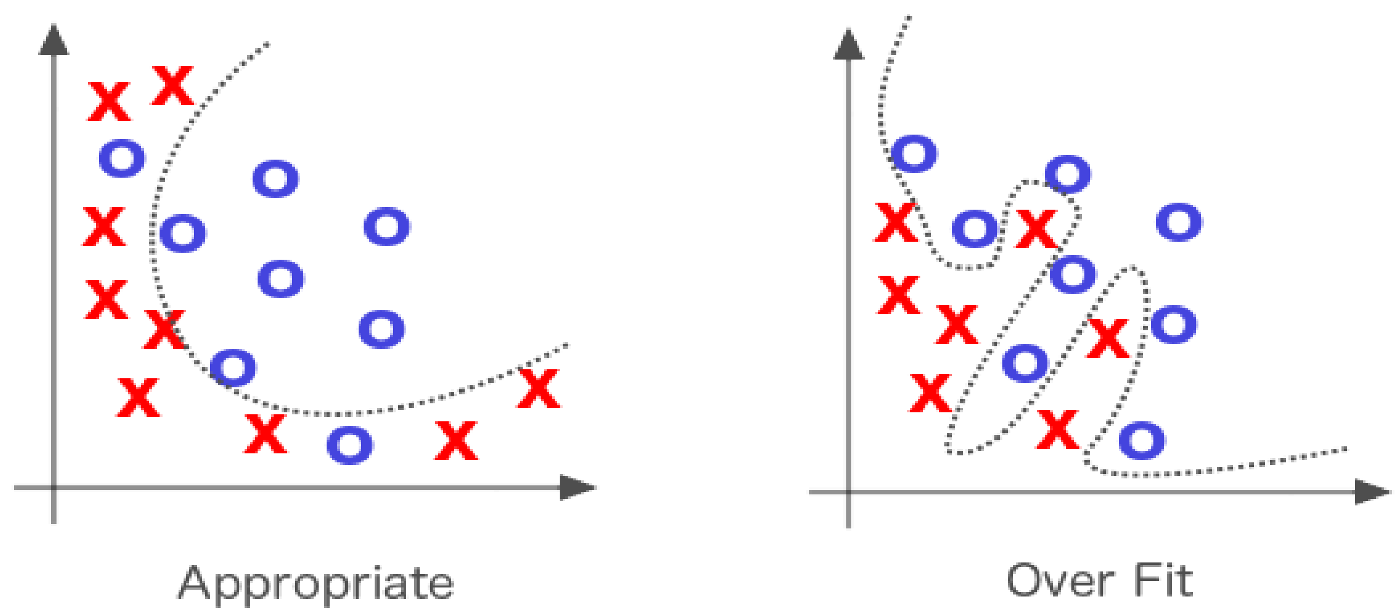
Need for Data

- “What is the **best performance one can get** ?” rarely has an answer
- When comparing multiple models, one can answer “what is the **best of these models, for this given dataset** ?”
- It does not answer “what is the **best model at this task** ?”



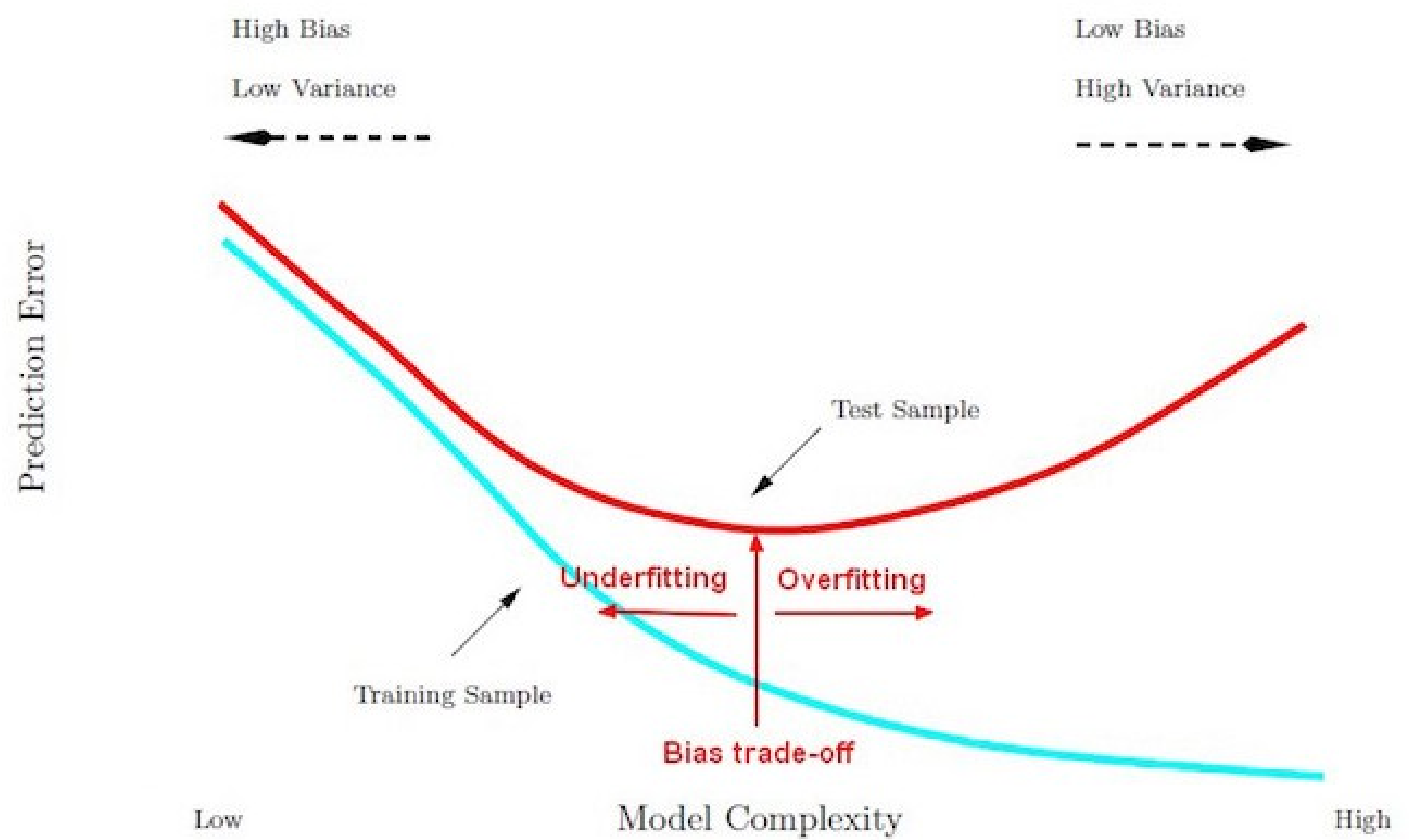
Over-fitting

- “Too good to be true” model performance can be explained
  - Excessive modeling capacity (too many parameters, parametrization is too flexible, ...)
  - Model parameters have learn the trained data by heart
- Characterized by very good performance on the training set and (much) lower performance on unseen dataset



Generalization

- Systematic error ≡ bias
- Sensitivity of prediction ≡ variance
- A good model is a tradeof of both
  - Early stopping can help with halting the model



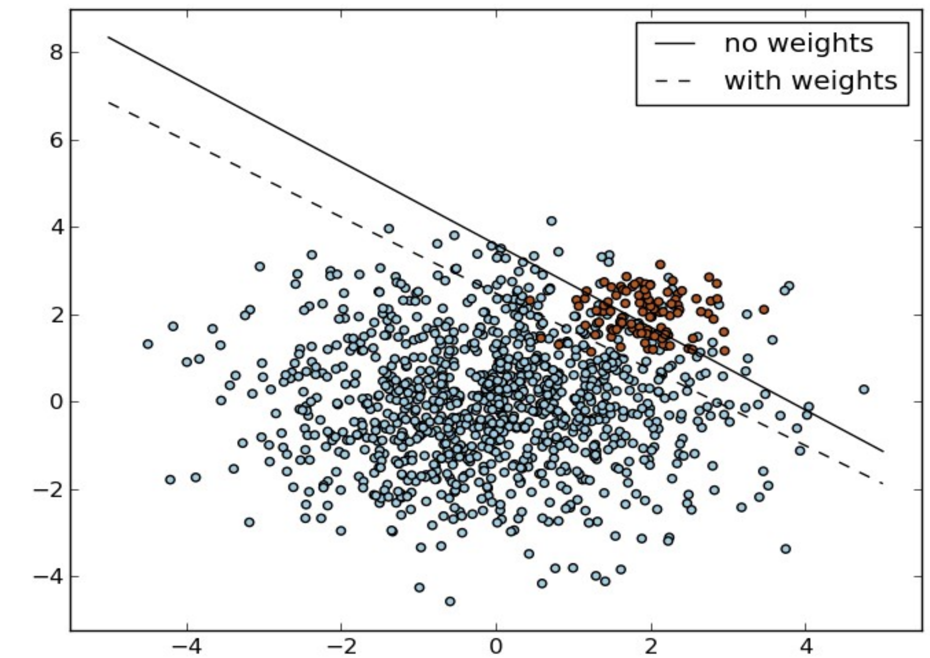
Figure(s) of Merit(s)

- Objective function in optimization might be chosen for computational reason (differentiable, …)
- **Objective function might only be a proxy** to the actual figure of merit of the problem at hand
- Multi-objective optimization is subject to trade-off between objectives

➢ While model optimization is based on the loss function over the training set, following the evolution of a more interesting metric over the validation can help selecting models that are better for the use case

Class Imbalance

- In many cases the number of samples varies significantly from class to class
- Class imbalance biases the performance on the minority class
- Multiple ways to tackle the issue
  - Over-sample the minority class
  - Synthetic minority over-sampling
  - Under-sample the majority class
  - Weighted loss function
  - Active learning



- NB: metrics can be sensitive to class imbalance and be misguiding if not correct : e.g. 99% accuracy with 0% recall

Training

- Training phase or learning phase is when the parameters of the model are adujsted to best solve the problem
- For some model/technique (especially deep learning) this can become computationally prohibitive
- General purpose graphical processing units (GP-GPU) offer an enormous amount of parallel compute power, applicable to specific numerical problems
- Matrix calculation, minibatch computation, deep learning, … can get a significant boost from GP-GPU.
- Further parallelization can be obtained across multiple nodes/GPU using

Hyper-parameter Optimization

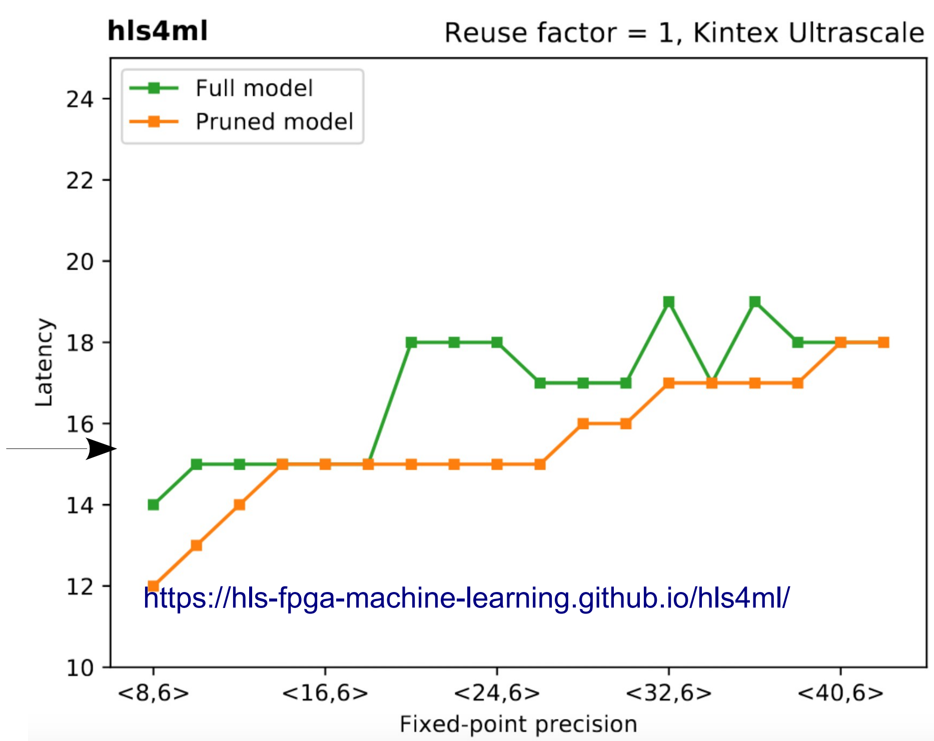
- Most optimization methods and models require hyperparameters
    - number of layers in an ANN, number of leaves in a decision tree, learning rates, …
- In most cases these parameters cannot be optimized while the model is trained
- Their values can however significantly influence the final performance

- These can be optimize in various ways
  - Simple grid search
  - Bayesian optimization
  - Evolutionary algorithm
- Model comparison should be done very carefully
  - K-folding is a “must”

Cost of Running the Model

- Contrary to training, making prediction from a trained model is usually rather fast, even on CPU
- However fast is may be, it might still not be fast enough for the particular application
- Faster inference can be obtained on specialized hardware GP-GPU, TPU, FPGA, neuromorphic, … when the application allows it (trigger, onboard electronics, ...)

75 ns



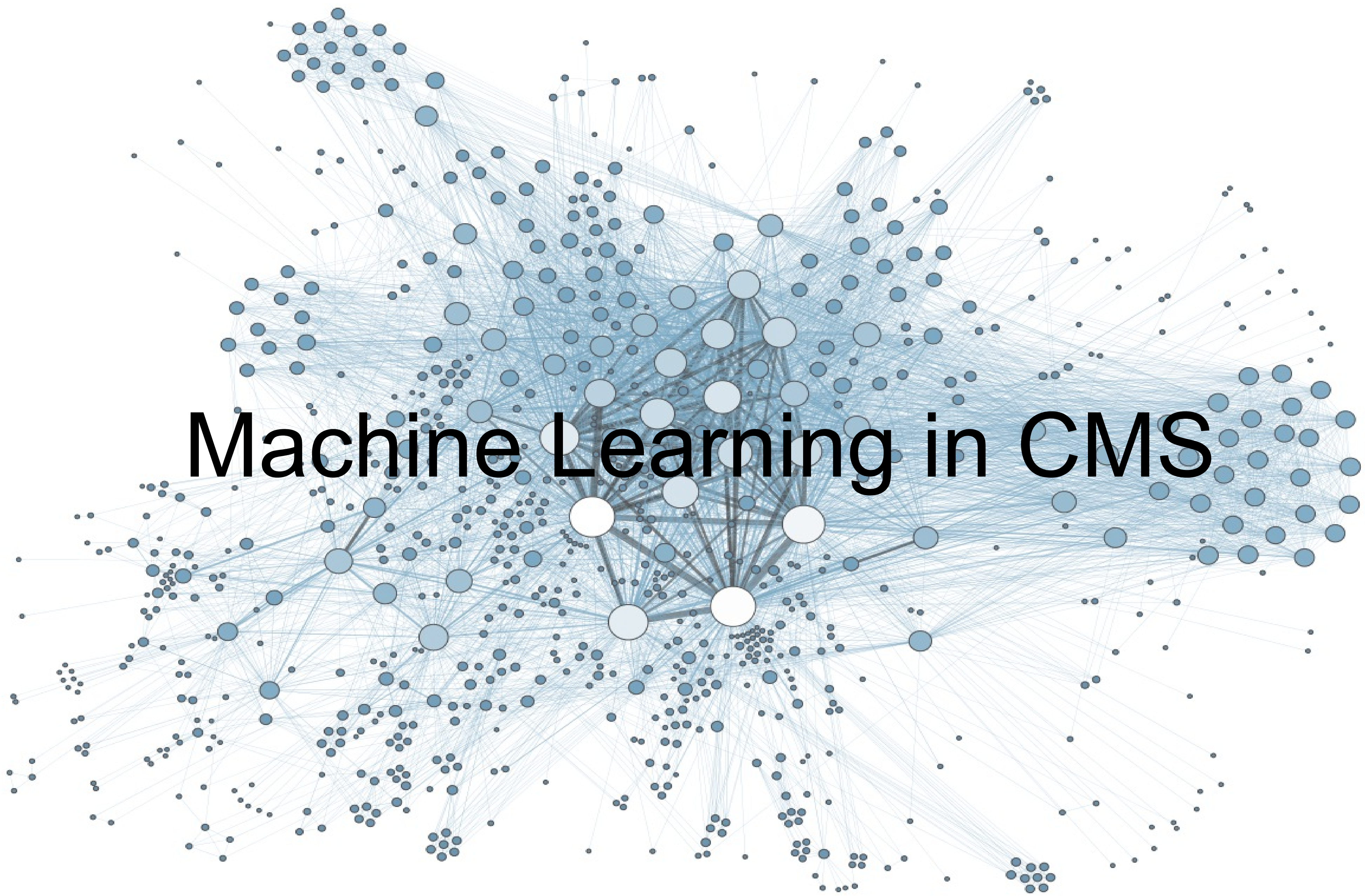
Take Home Message

Machine learning applications need to be developed with scientific rigor.

Lots of interesting studies possible on statistics/theory of learning.

Keep an eye on cost of prediction.

Machine Learning in CMS



Take Home Message

Model prediction can be fast and help with computing restrictions.

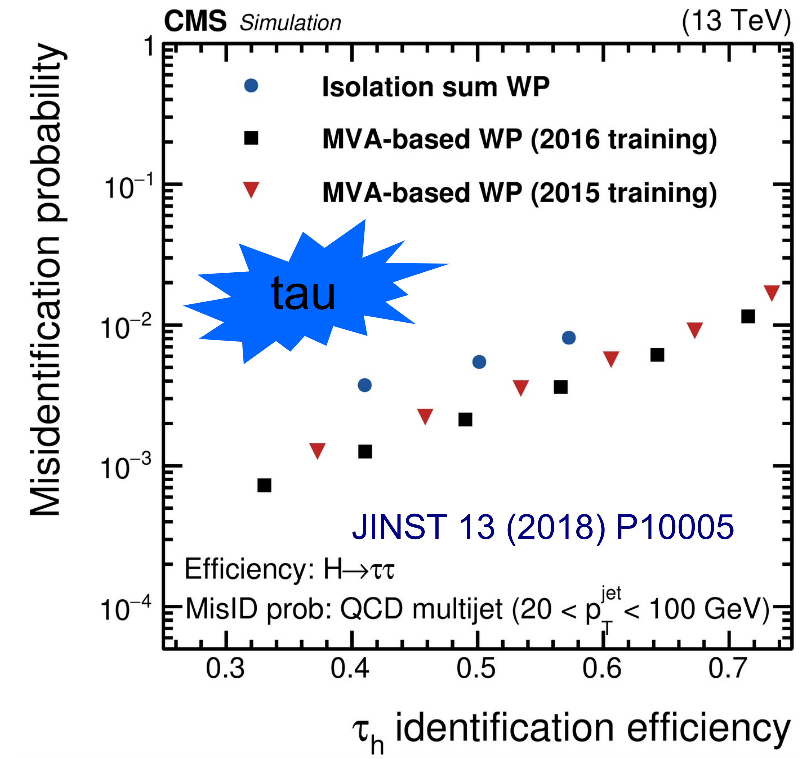
HEP data representation is multi-trait and match with wide range of existing tools.

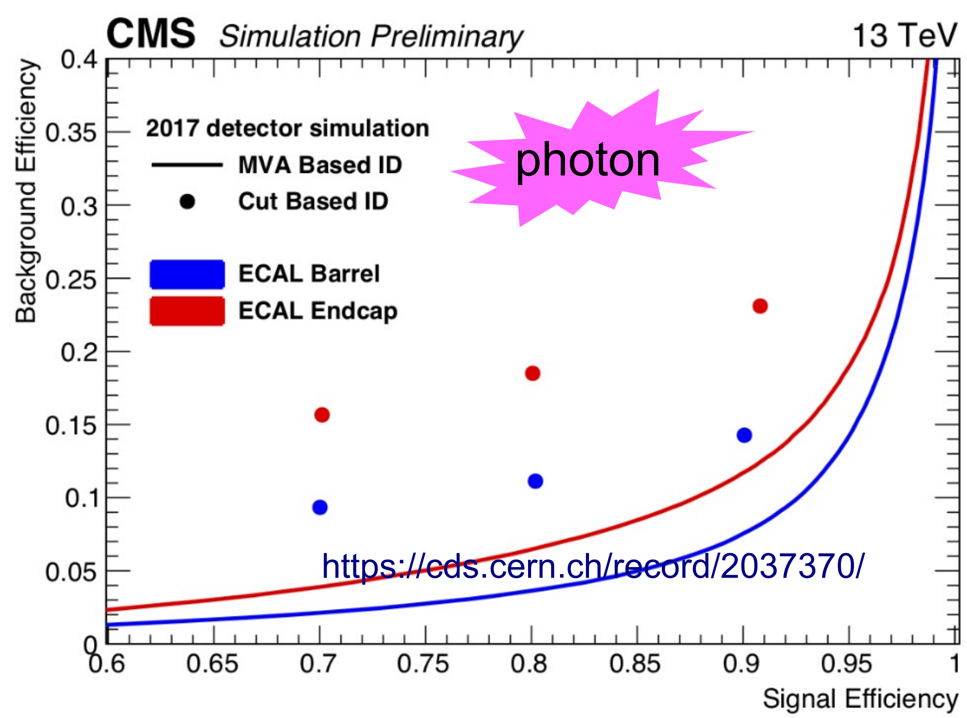
Machine learning can help with extracting better physics knowledge from data.

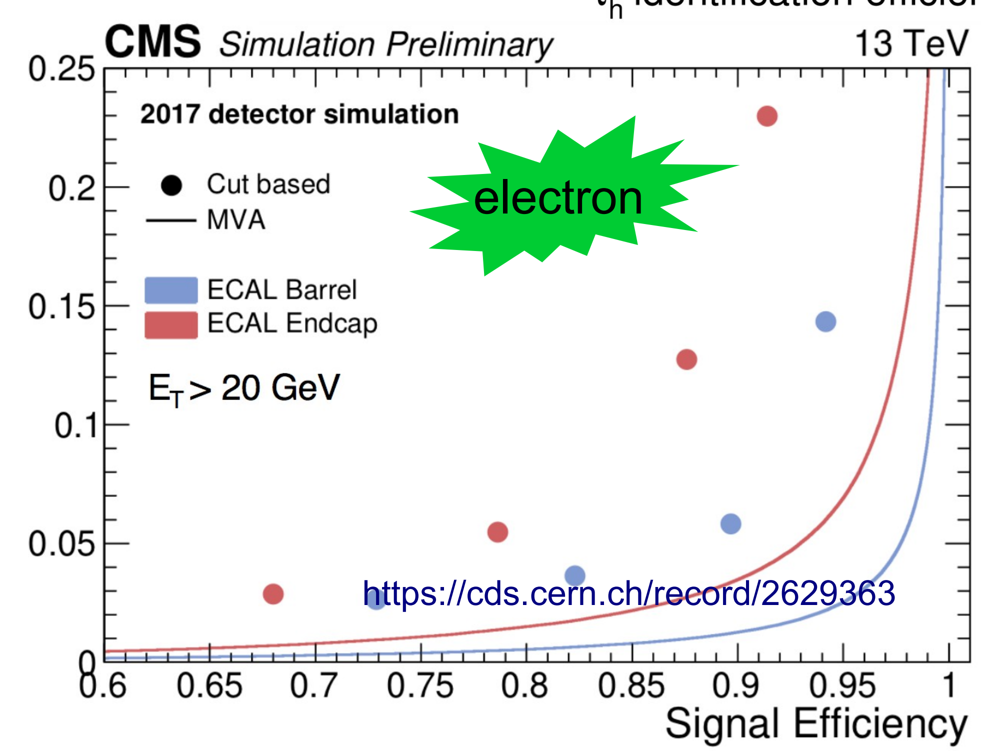
Particle Identification with ML

Particle Id

Object-level features boosted decision tree classification.
Analysis specific Muon identification with MVA.

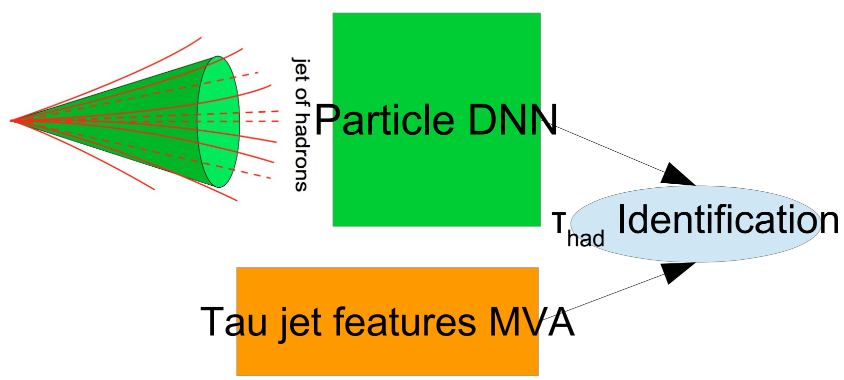


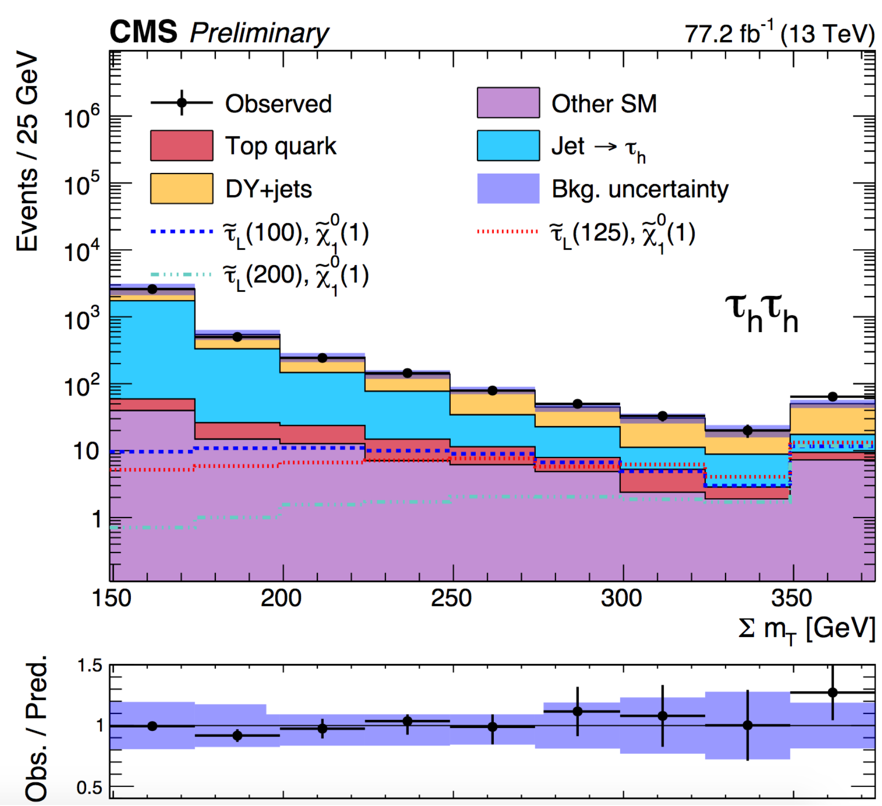




Tau-id with DNN

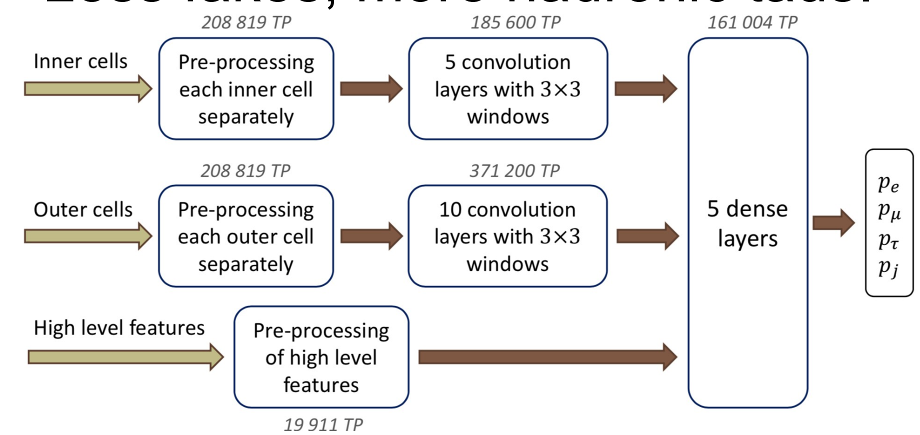
Combine the jet and particle features.
Reduction of fake hadronic taus.

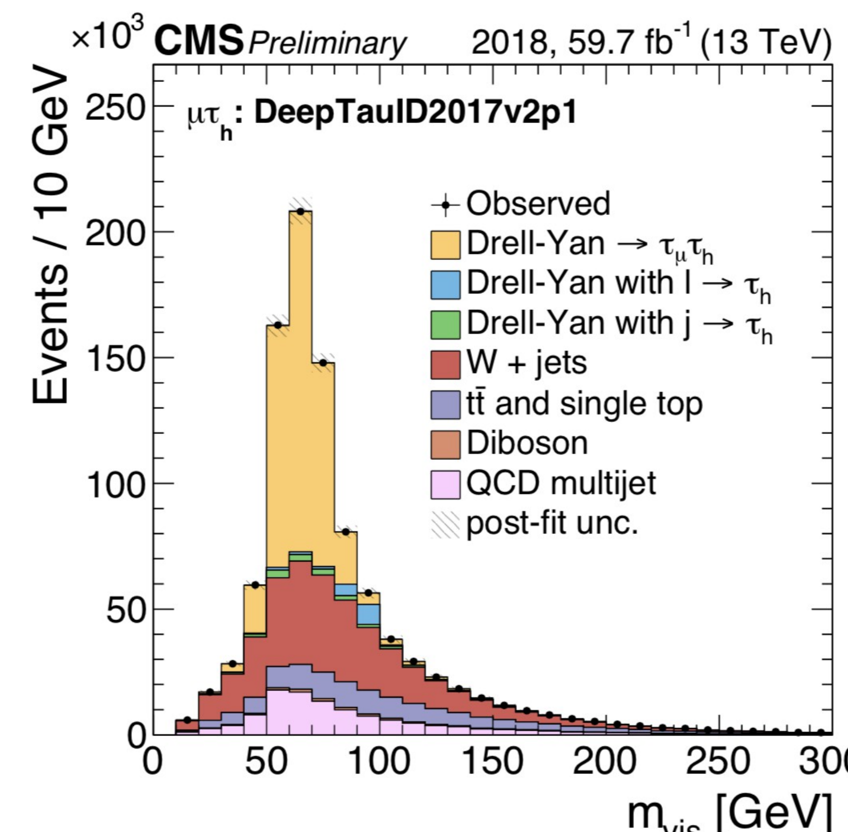




http://cds.cern.ch/record/2669190?ln=en

Combines jet and particle-image features.
Less fakes, more hadronic taus.



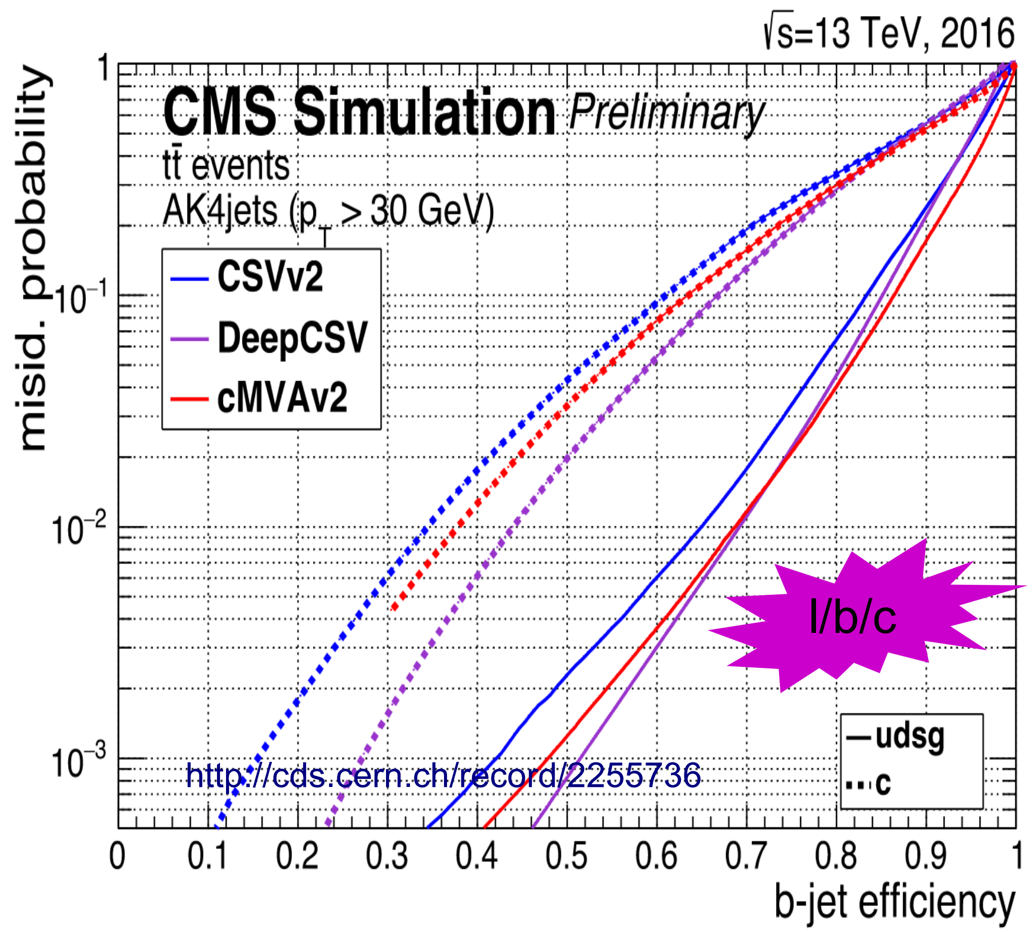


http://ceur-ws.org/Vol-2507/84-88-paper-13.pdf

Jet Tagging with ML

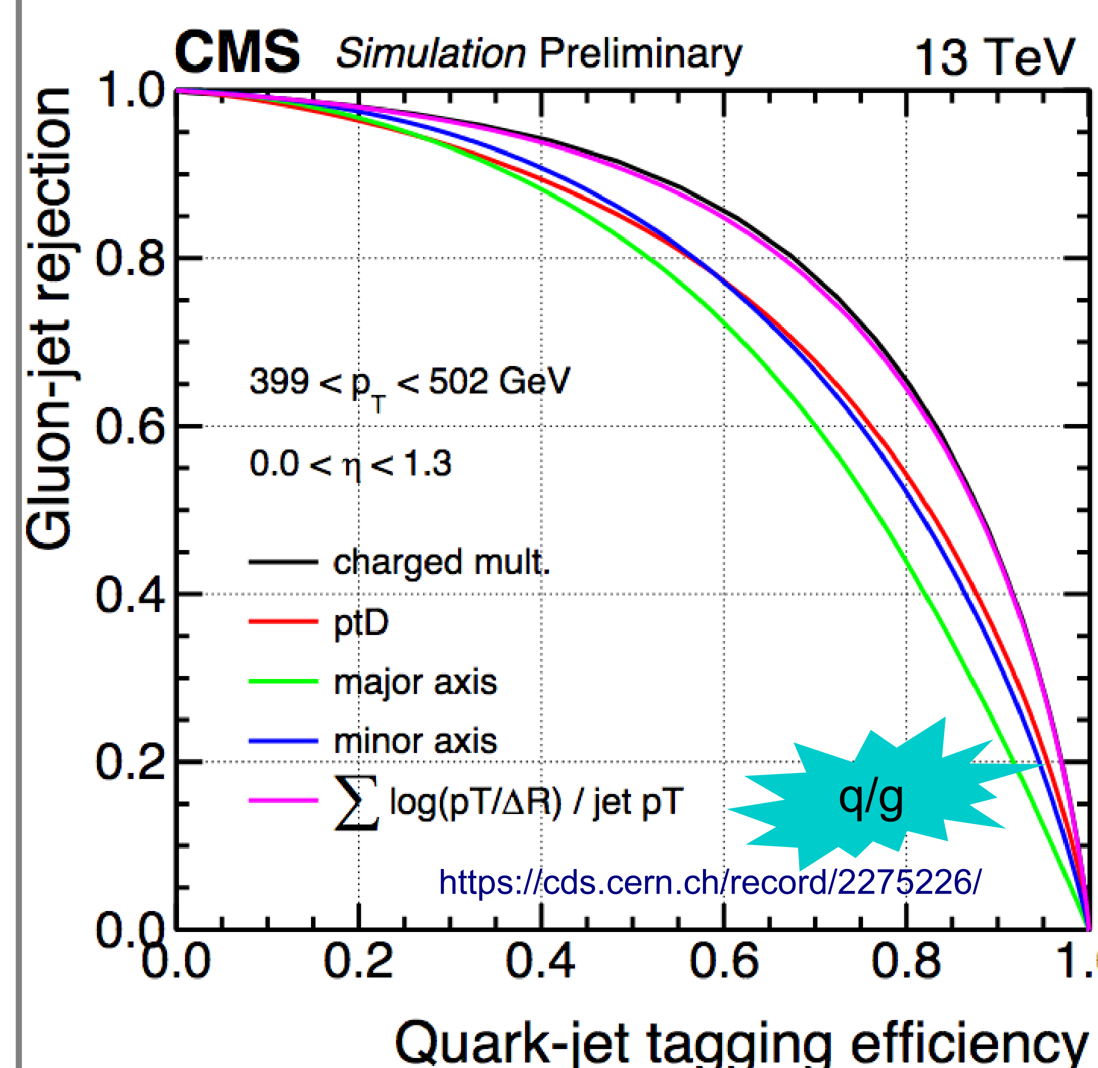
Extended studies in
https://cds.cern.ch/record/2683870

Jet Tagging

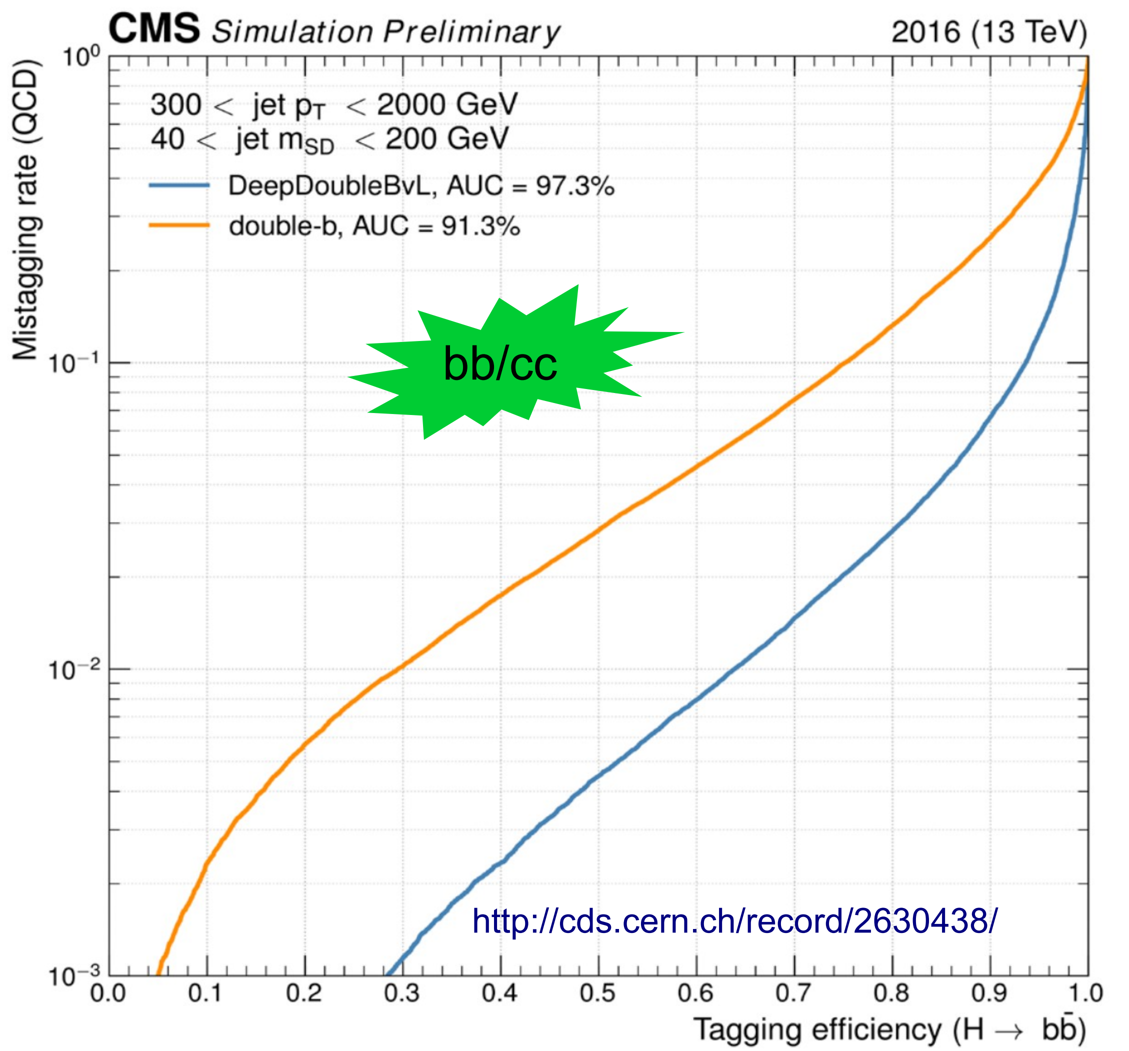


Combining jet-level, vertex-level and track-level features in FC neural net.

Jet-level features in a boosted decision tree for Quark/Gluon discrimination.

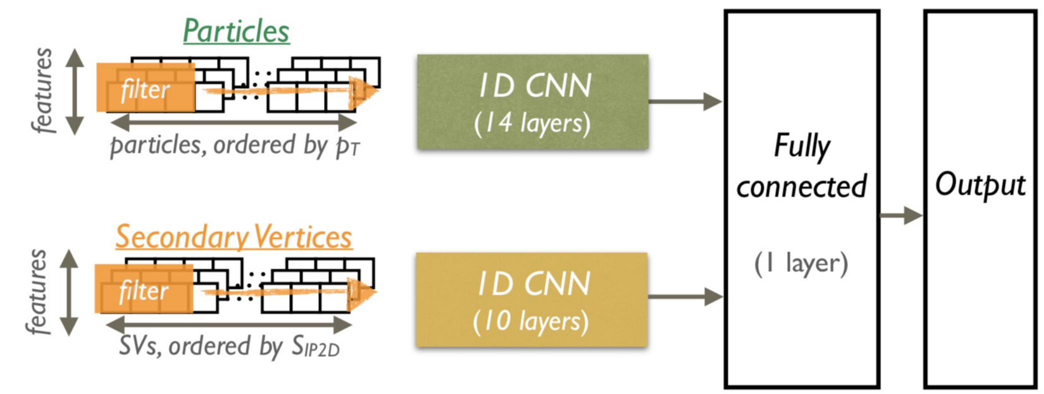


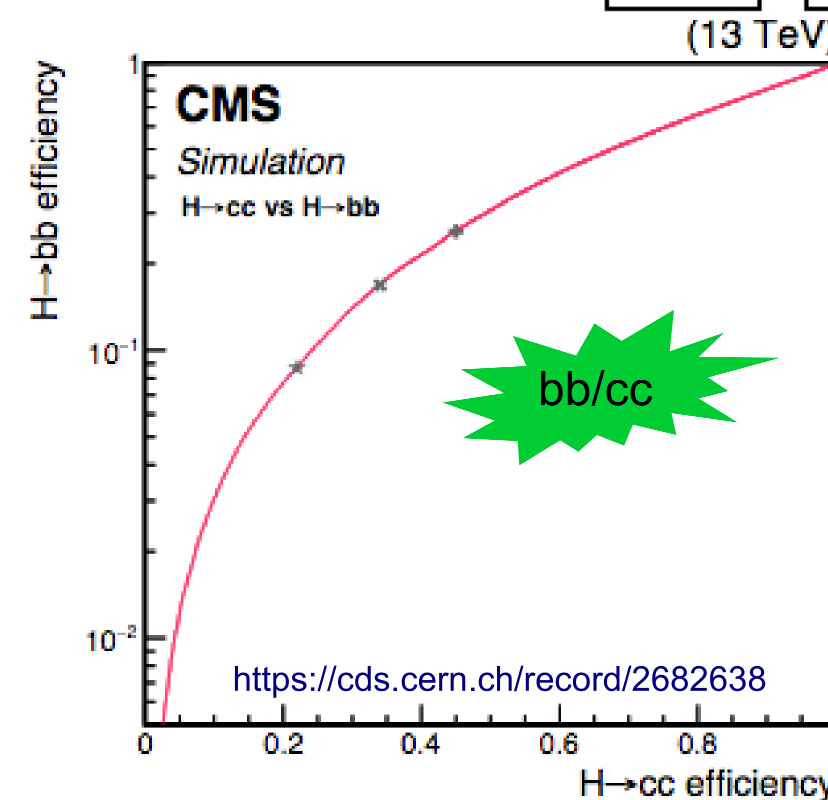
Higgs Tagging



Combining jet-level, track-level and vertex-level features in Conv1D/GRU/FC neural net.

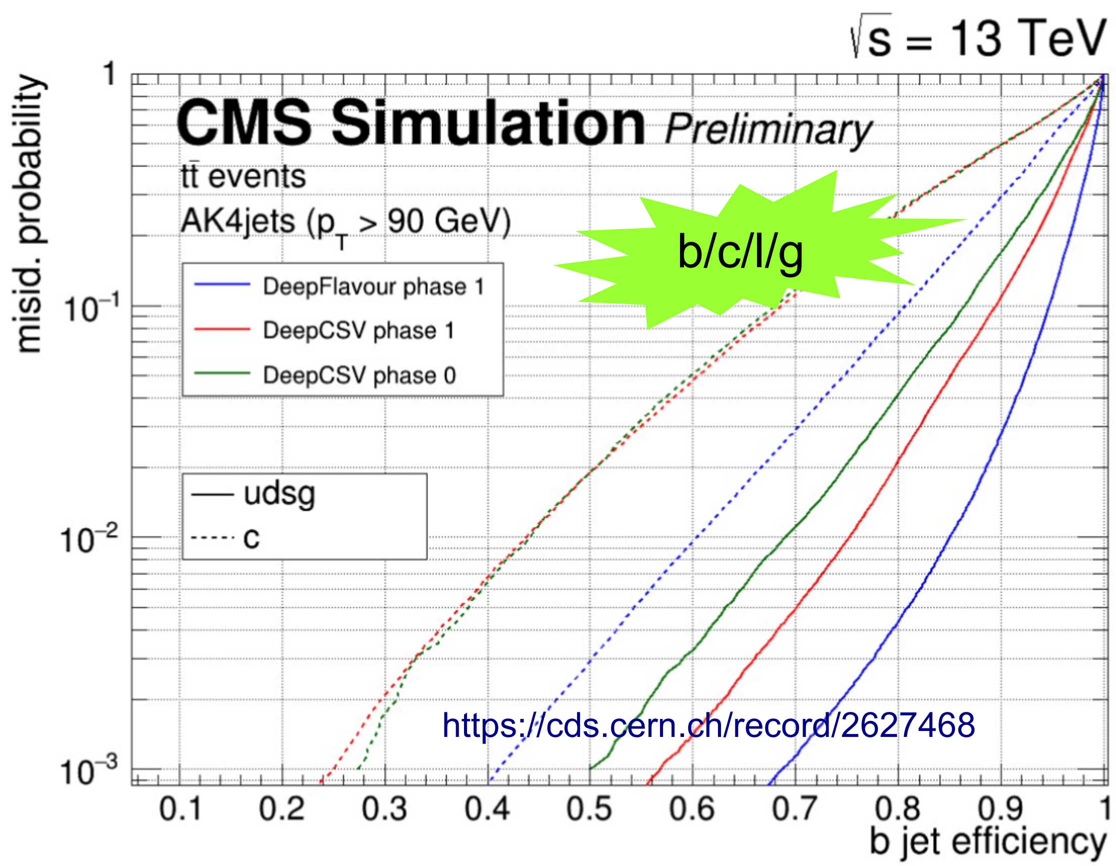
Combines particle and secondary vertex features in Conv1D/FC

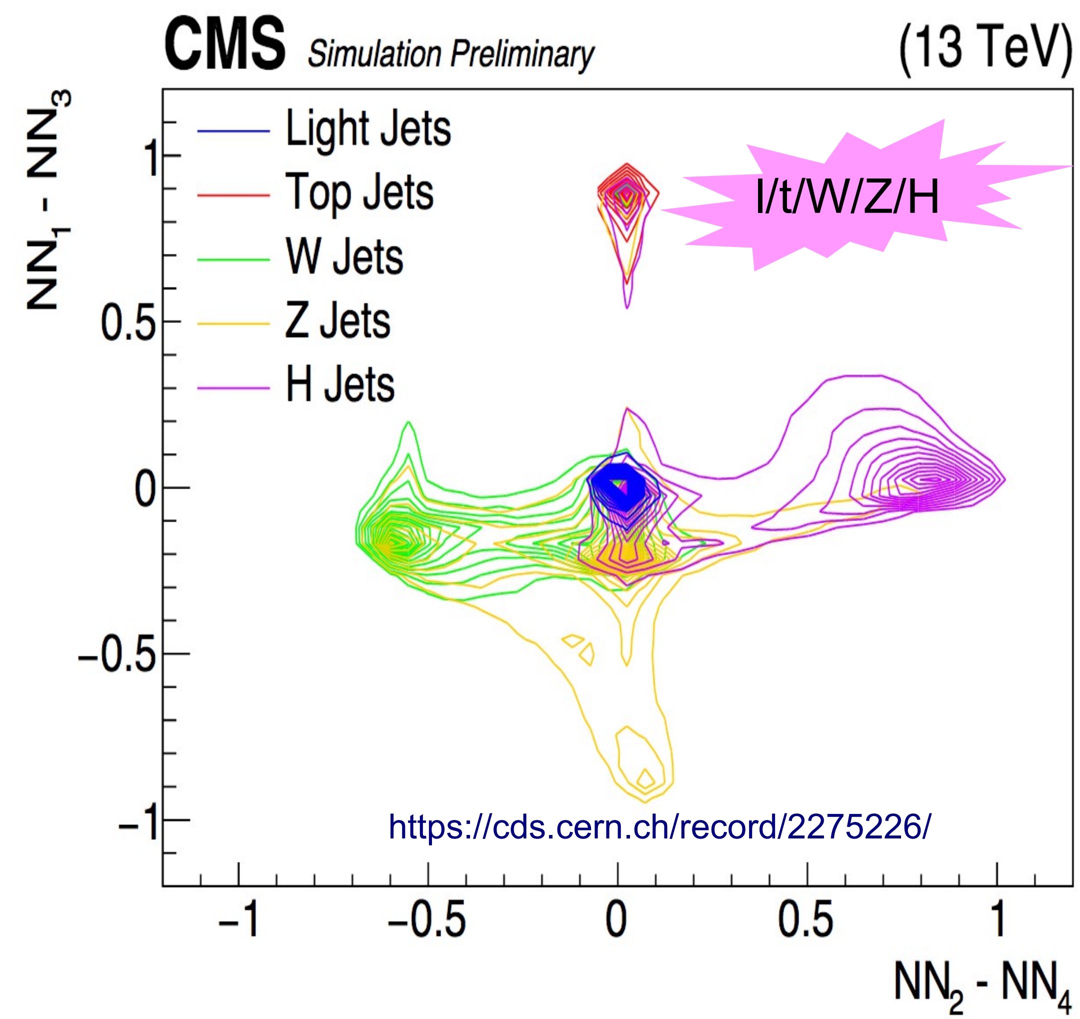




Jet x-Tagging

Jet-level, vertex-level and particle-level features in Conv1D/LSTM/FC multi-class neural net.

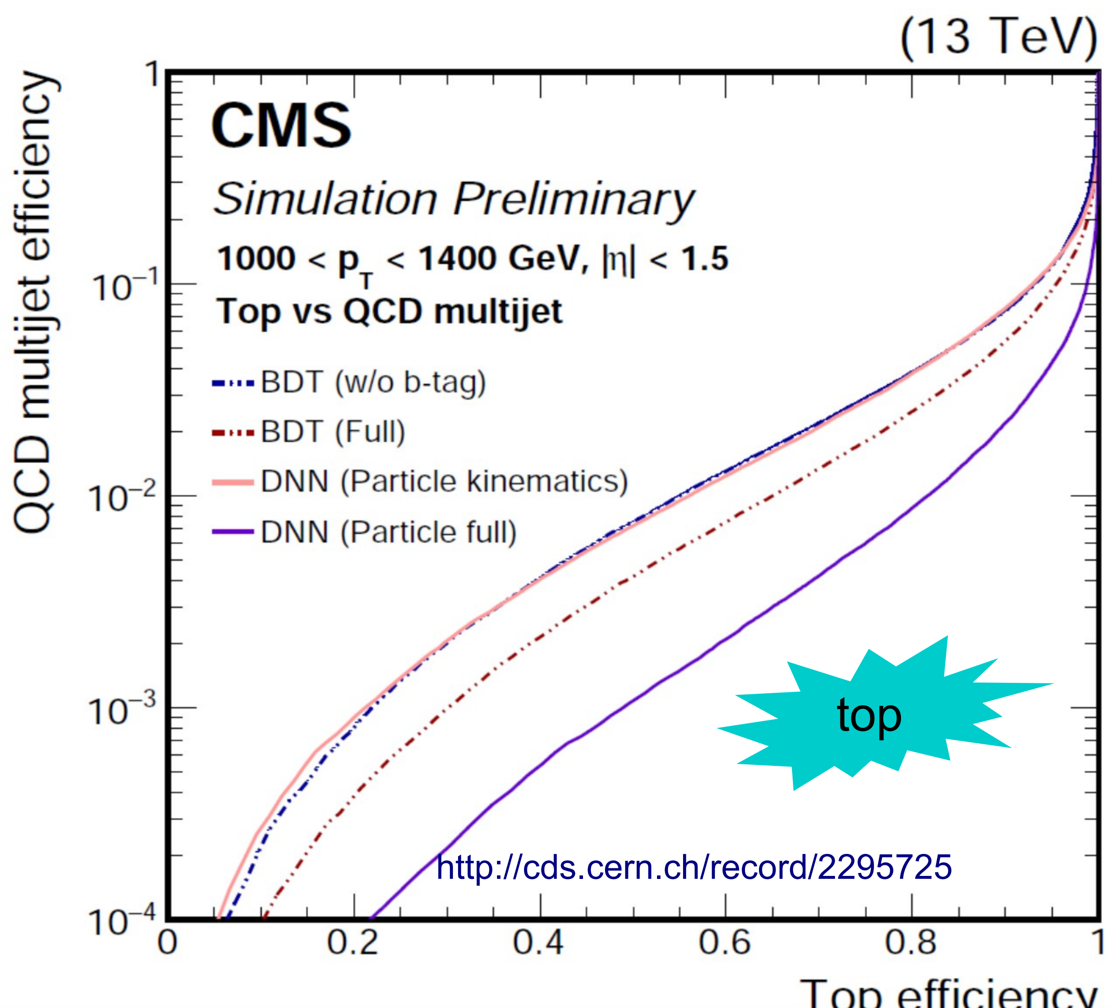




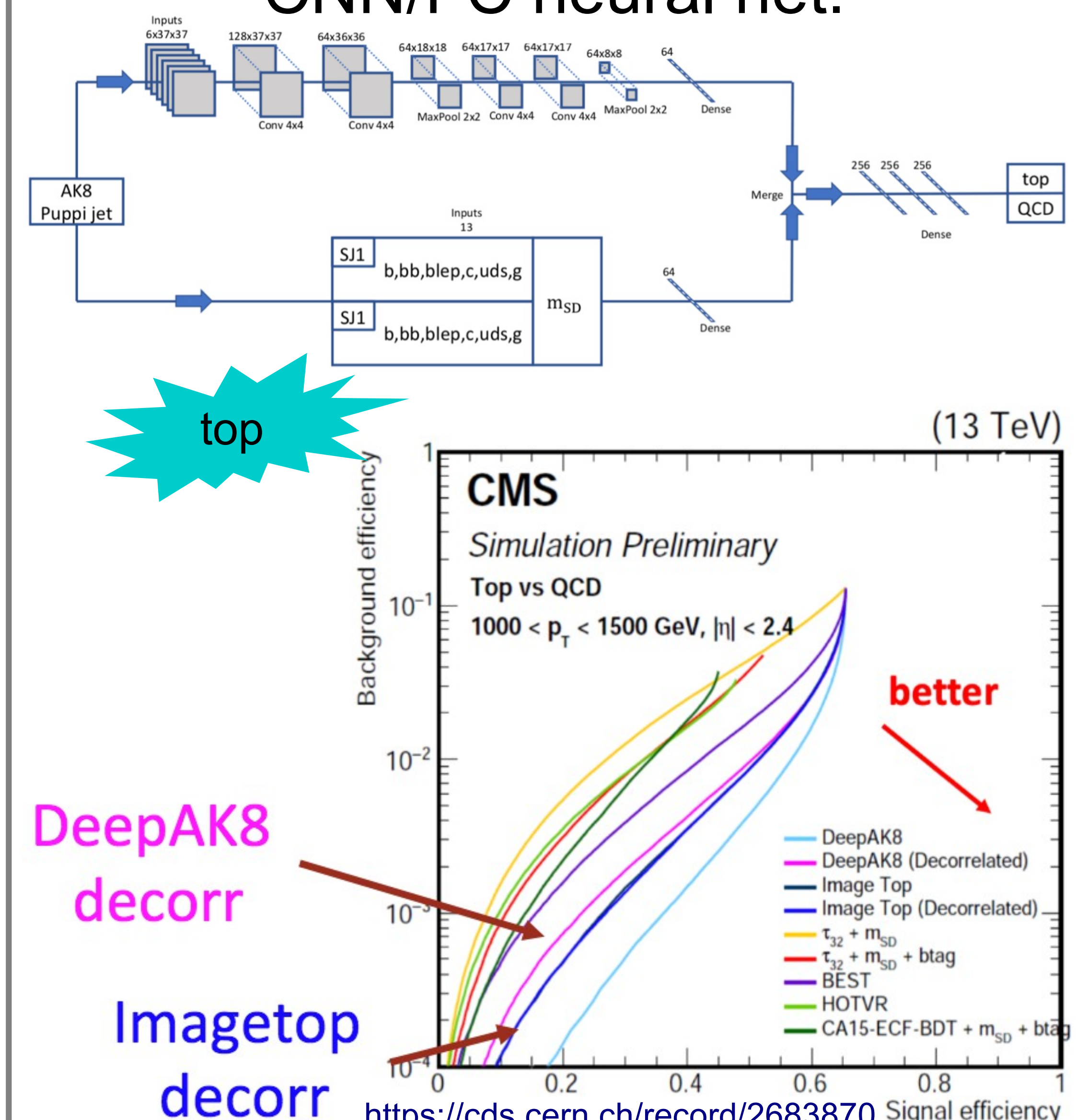
Jet-level features computed in hypothesized rest frame in dense multi-class neural net.

Top Tagging

Combining particle-level, track-level and vertex-level features in Conv1D/FC neural net.

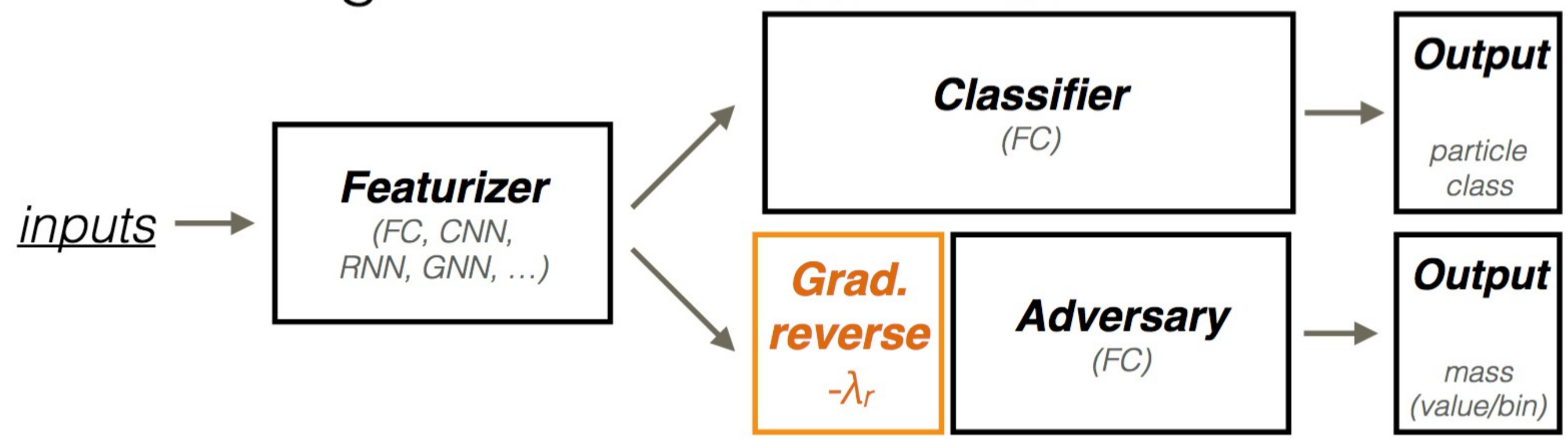


Combining particle-image and subjet tagging features with CNN/FC neural net.

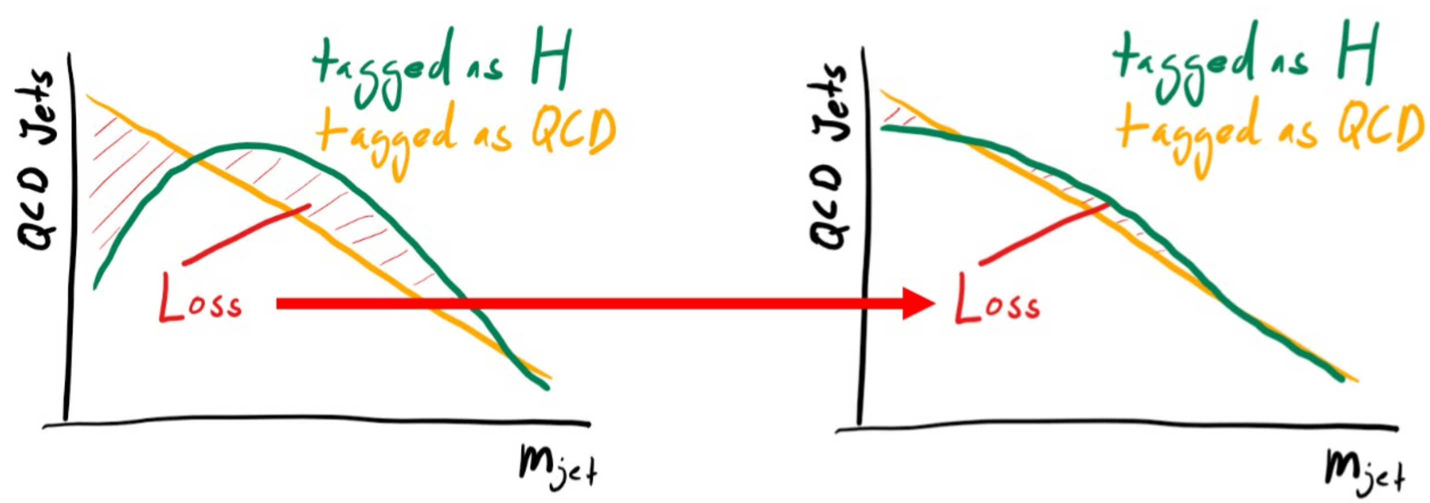


Mass Decorrelation

Adversarial training $L = L_{\text{disc}} + \lambda L_{\text{adv}}$



Dedicated "penalty term" $L = L_{\text{disc}} + \lambda D_{\text{KL}}$



https://arxiv.org/abs/1611.01046
https://arxiv.org/abs/1409.7495

Slide J. Duarte

Decorrelates the model output from targeted quantities.

Decorrelation Performance

The "simpler" approach: **Sample reweighting**

- Reweight the QCD jet mass distribution to match the signal one

The "brute force" approach: "**D**esigning **D**ecorrelated **T**agger" (DDT)

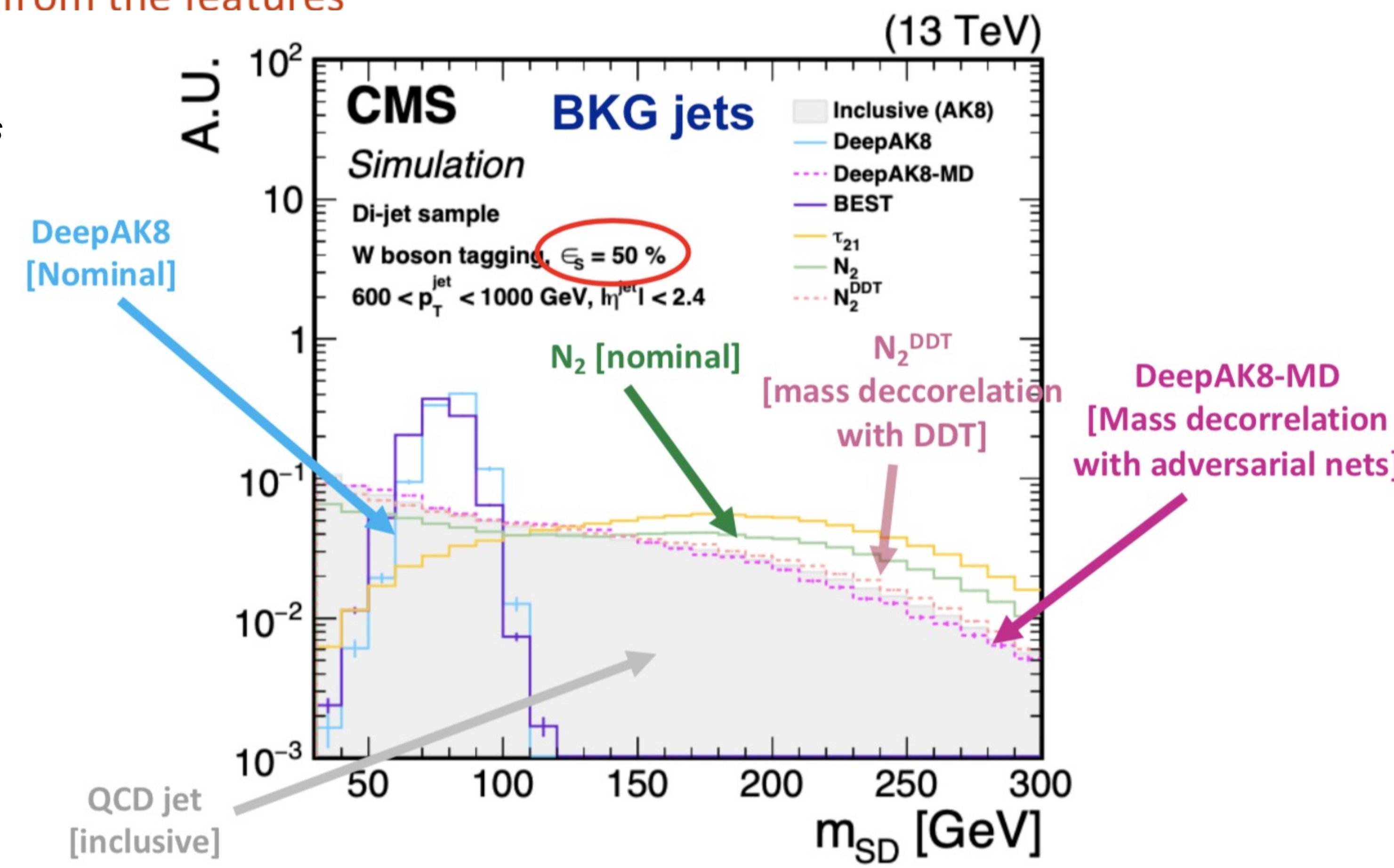
- Define a metric e.g., $\boxed{\rho = \ln(m_{\mathrm{SD}}^2/p_{\mathrm{T}}^2)}$ to capture tagger's correlation with m(jet)
- **Then:** transform tagger's response to preserve constant BKG rejection across **jet mass** and $\mathbf{p_T}$: **Tagger$^{\mathrm{DDT}}$ = Tagger – $\mathrm{X}_{(\#\%)}$**

The "painful" approach: **Adversarial networks**

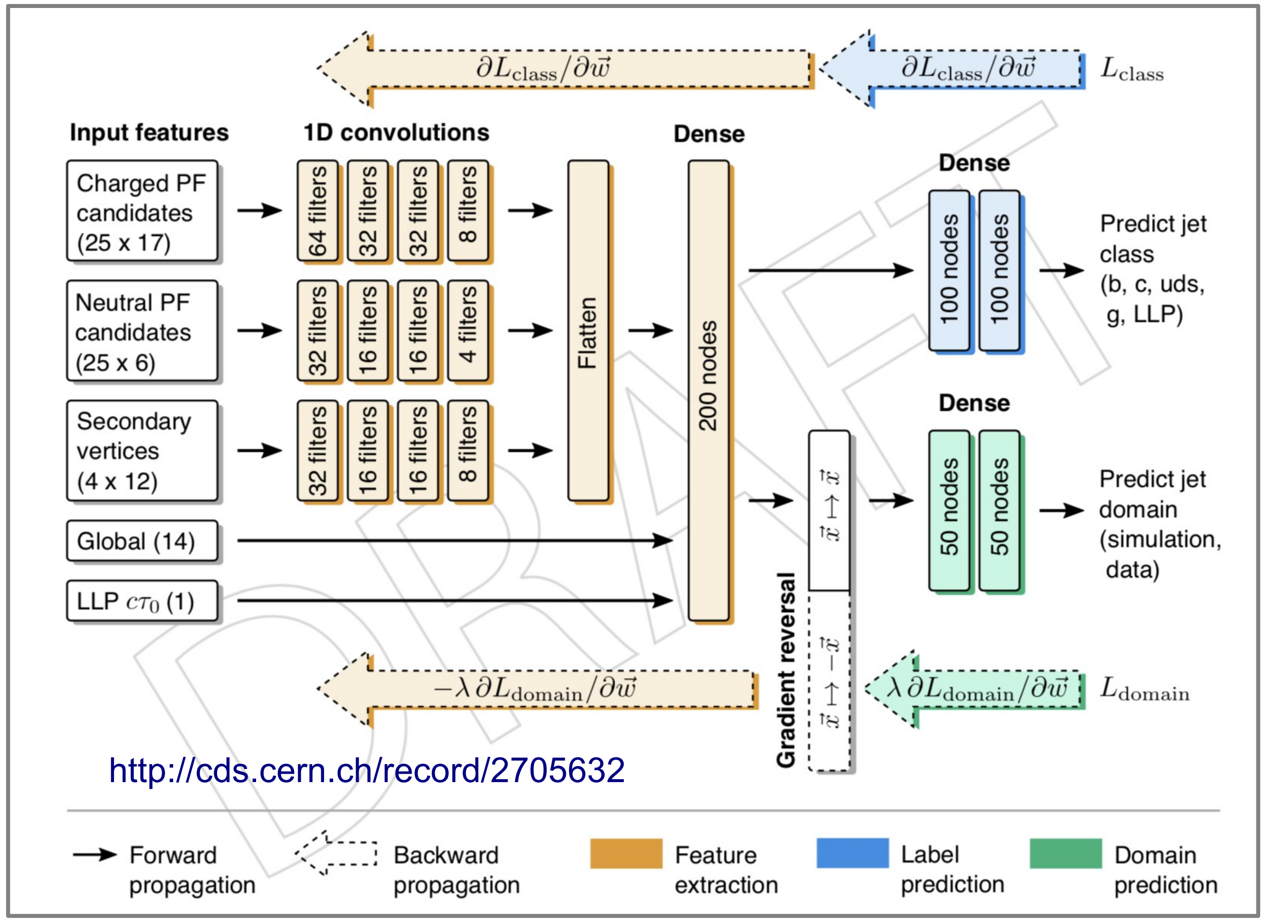
- Introduce a mass prediction network to predict m(jet) from the features extracted by the algorithm (DNN)

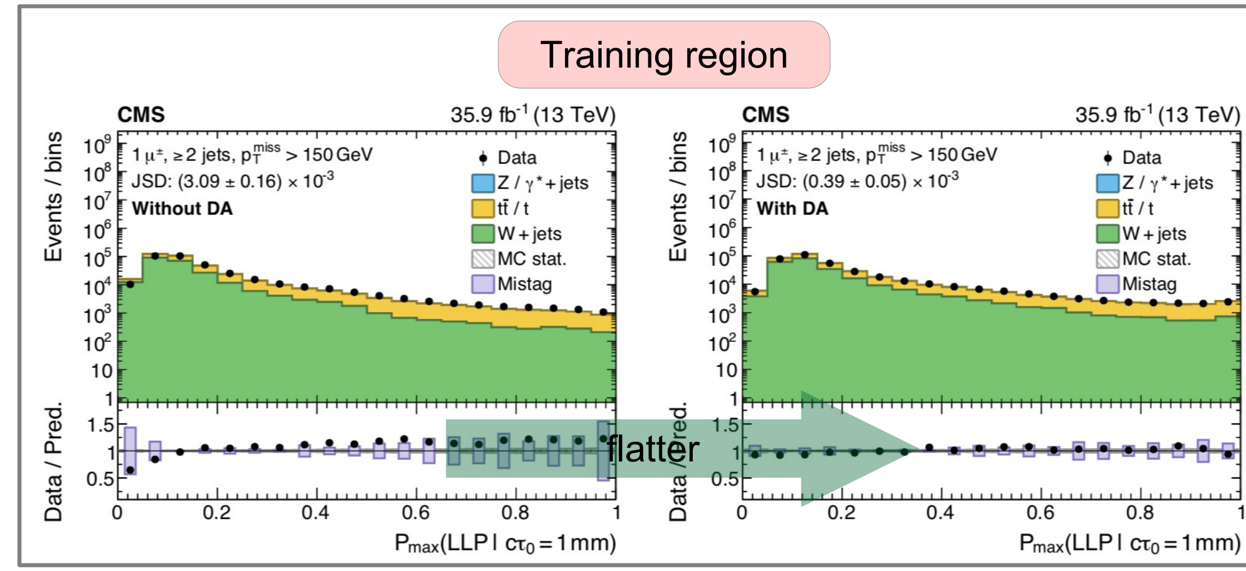
Slide L. Gouskos

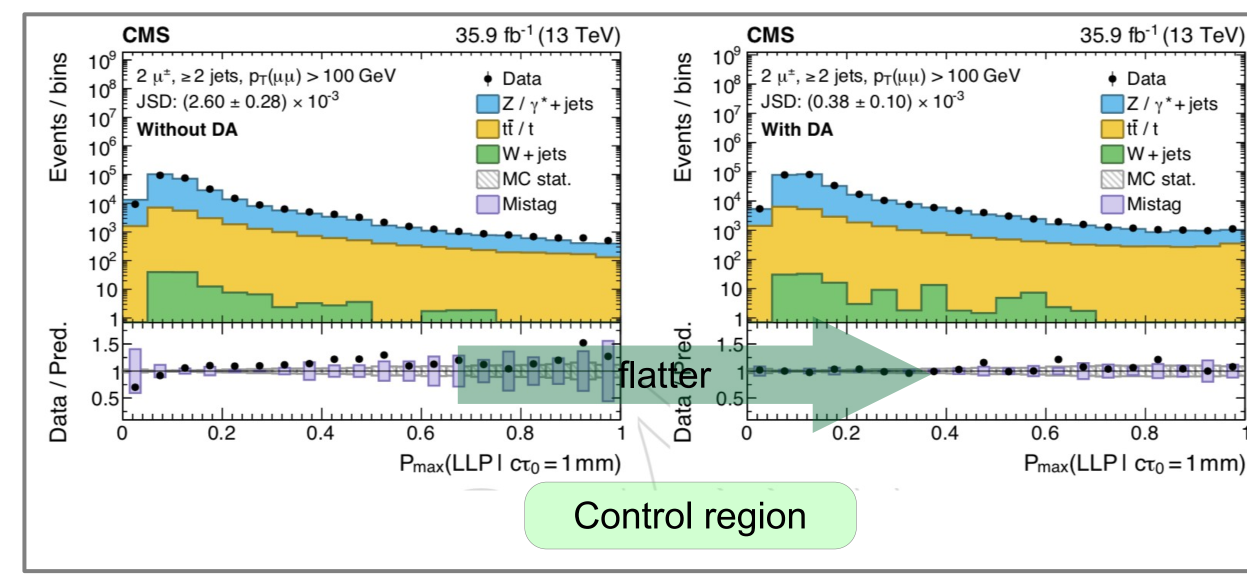
http://inspirehep.net/record/1746161
https://indico.cern.ch/event/836360/



Data/MC Agreement



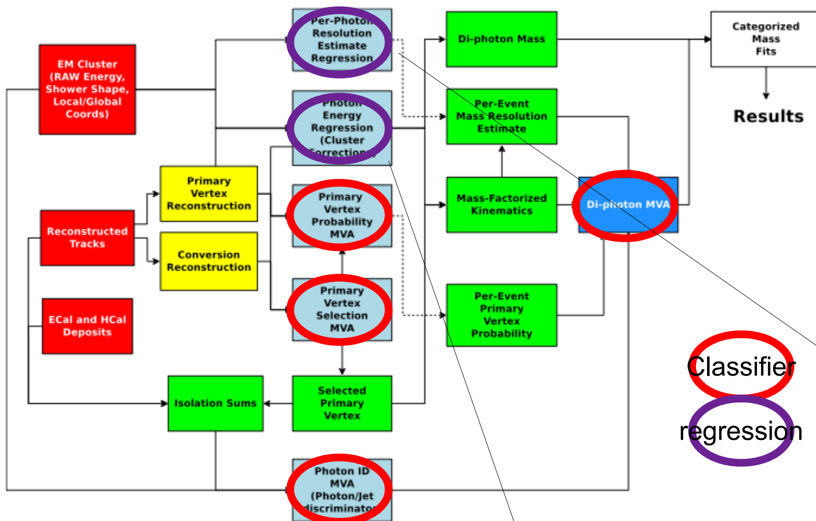




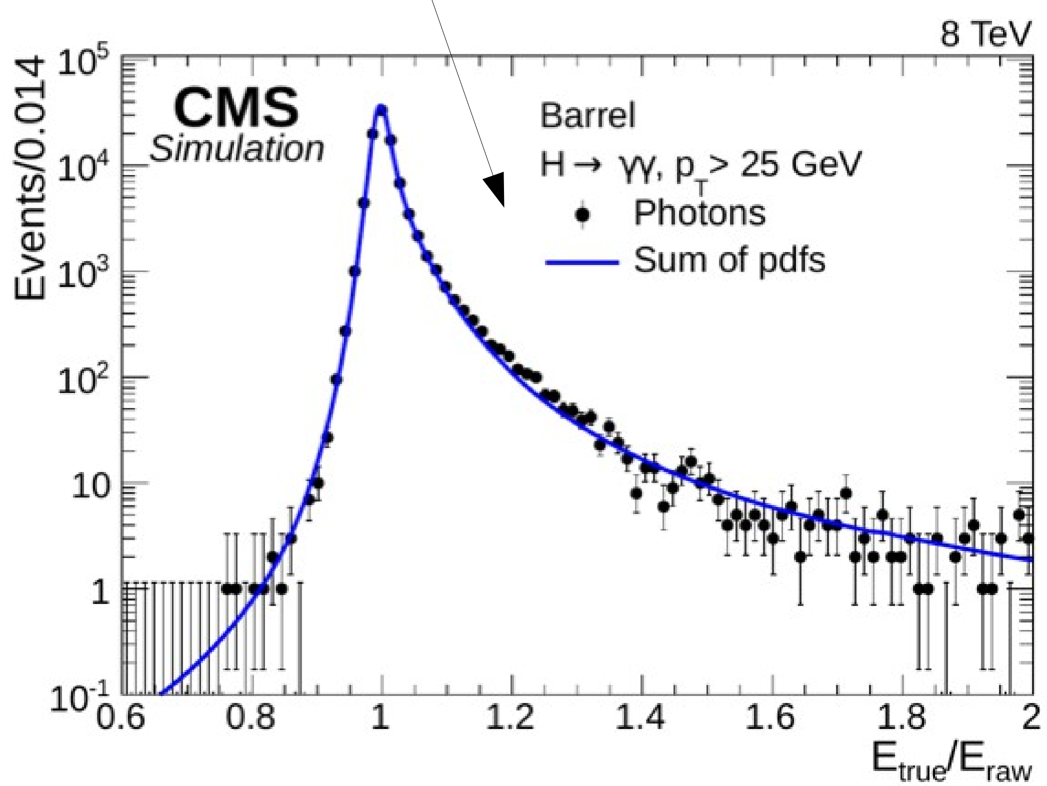
Mitigates the modeling discrepancies of discriminant between data and simulation

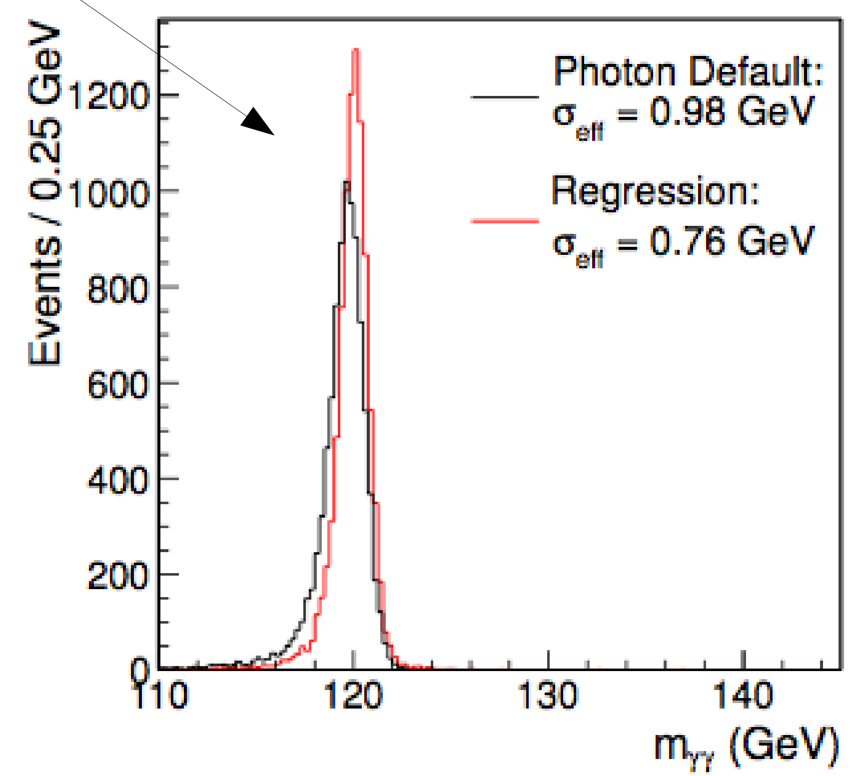
Energy Regression with ML

Higgs to gamma^2



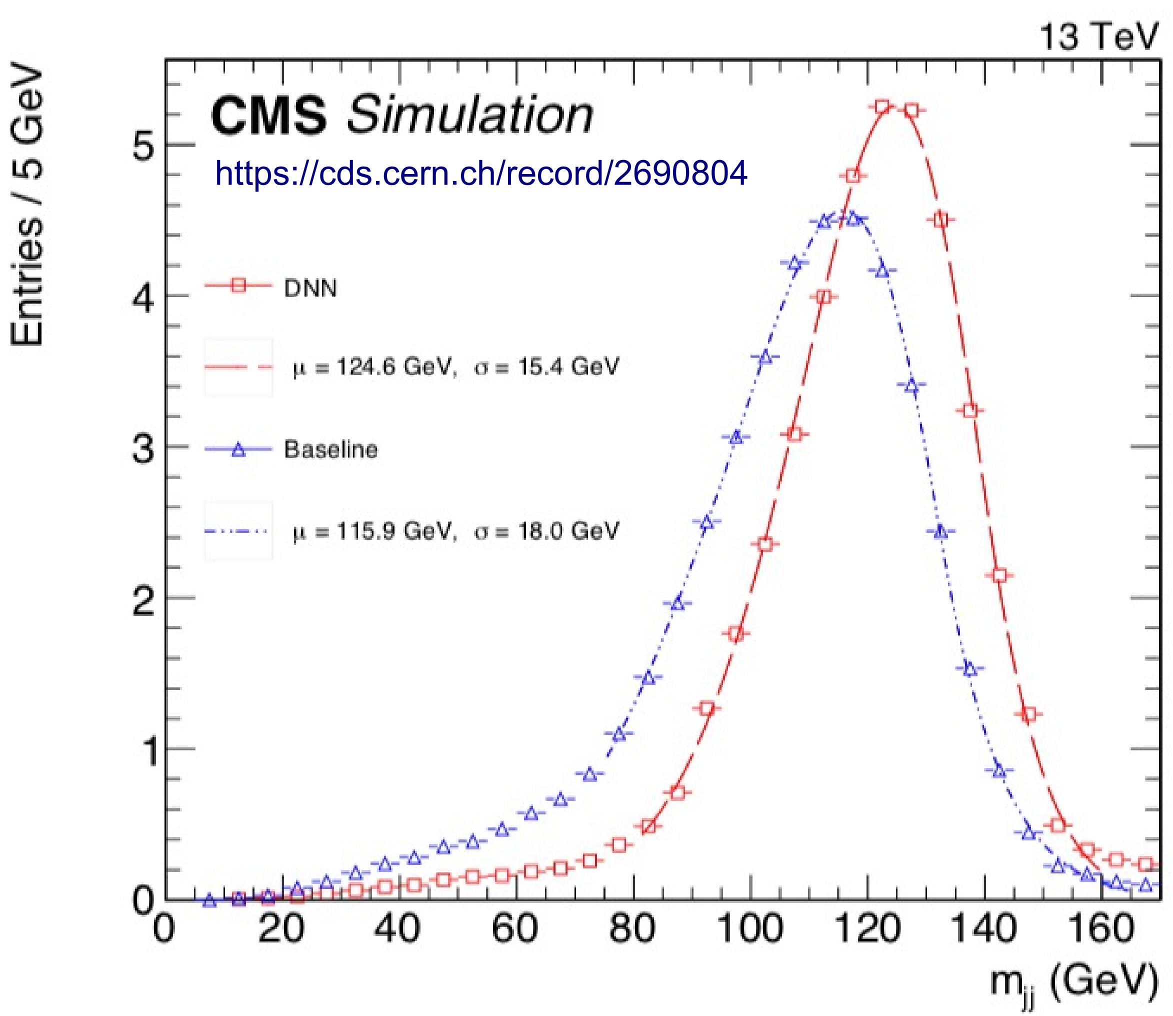
Per photon boosted decision tree based prediction of energy and resolution with semi-parametrized loss function.

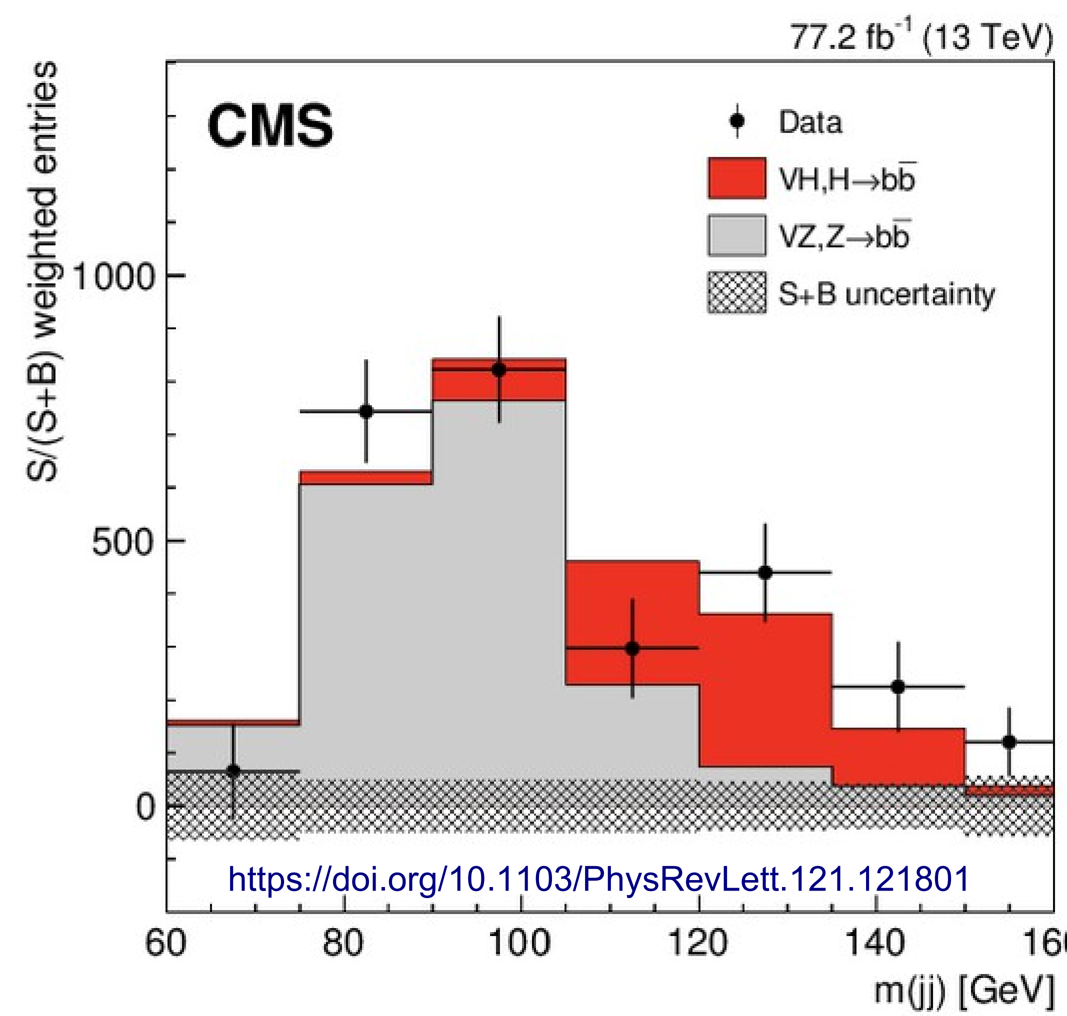




b-jet Energy & Resolution

Fully connected jet-level features neural network predicts the jet energy correction and resolution using quantile regression .

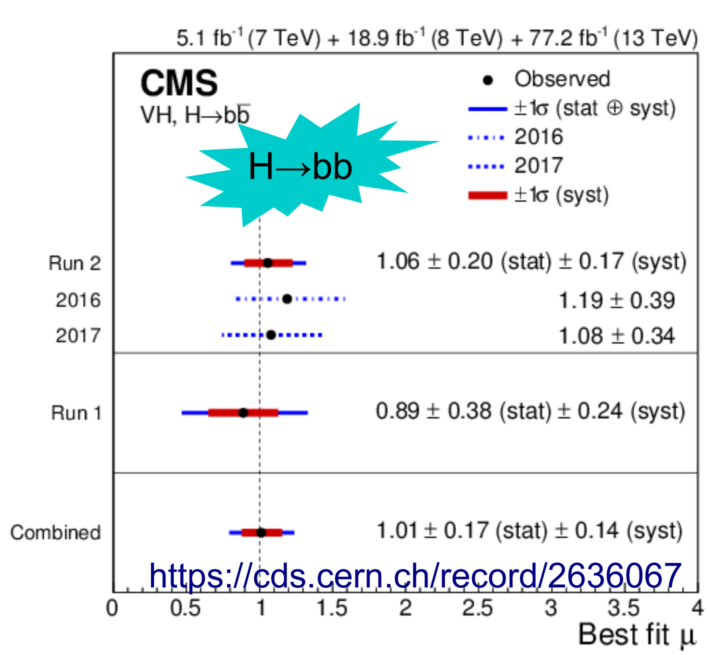


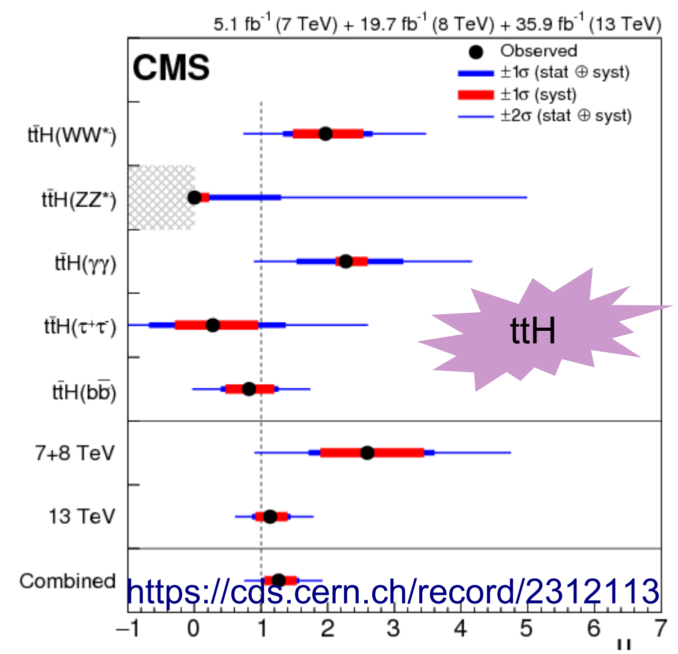


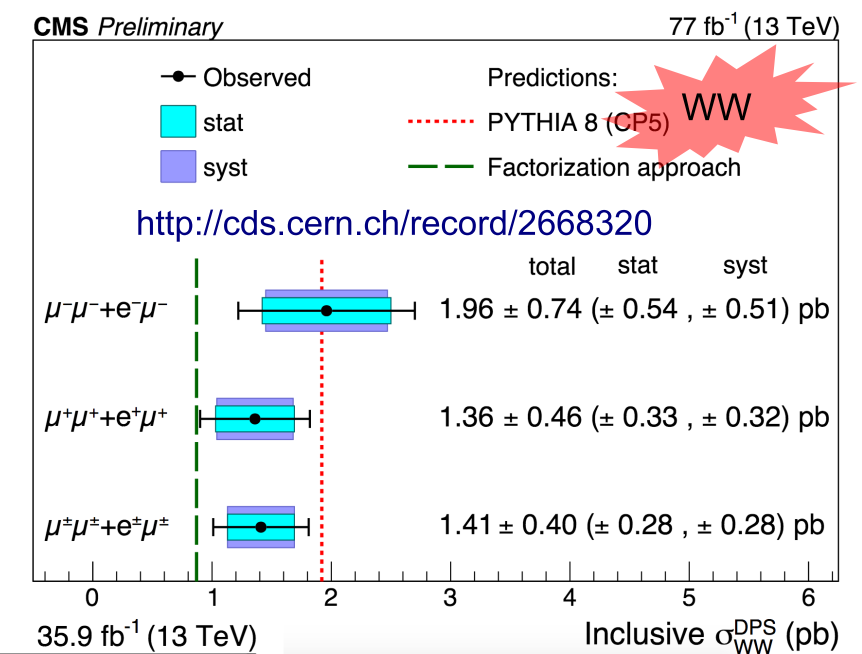
~20% improvement on Higgs mass resolution.

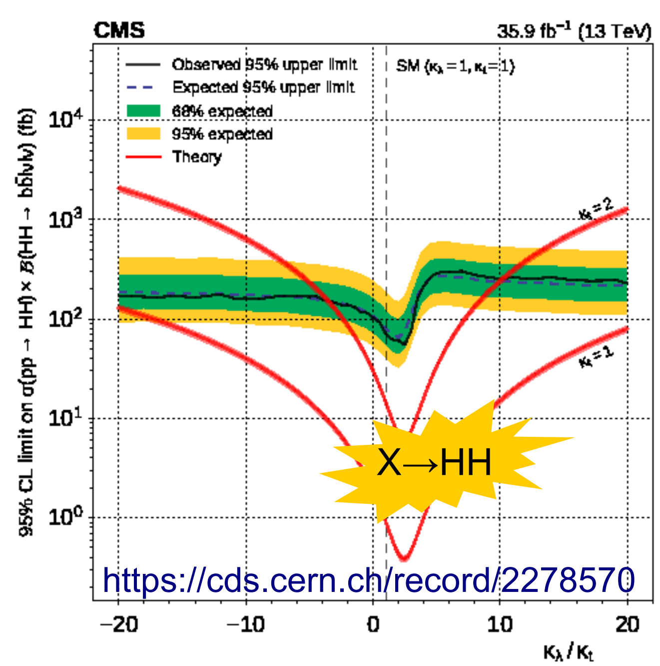
Analysis with ML

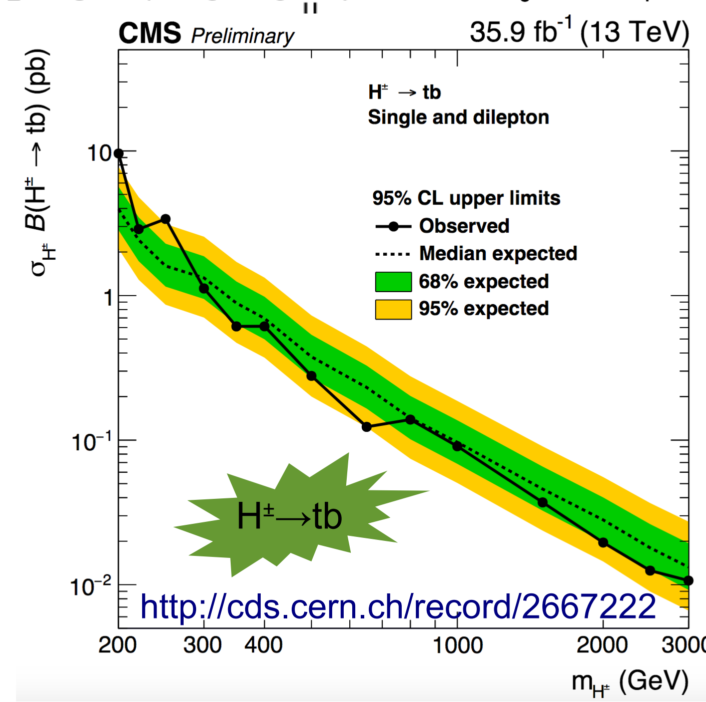
Increased Sensitivity





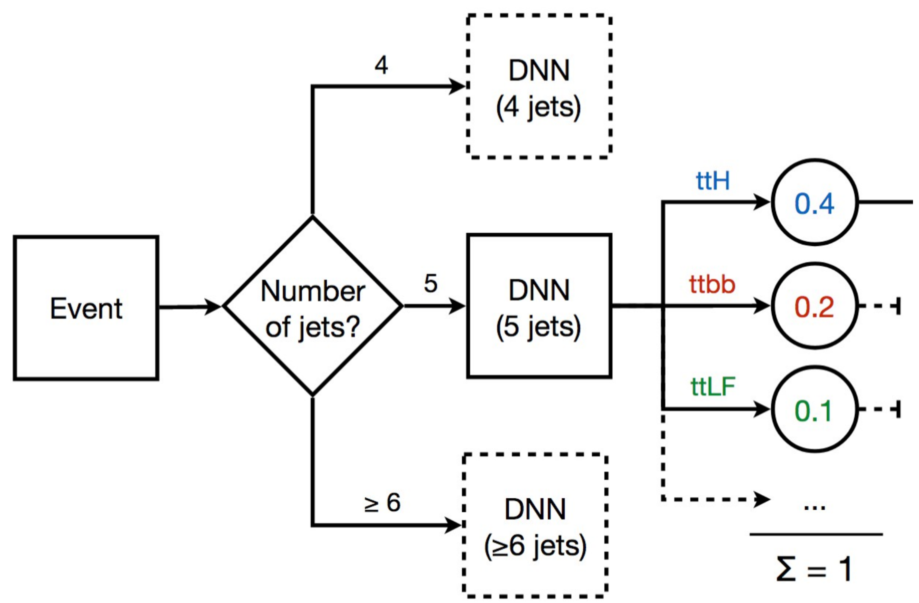




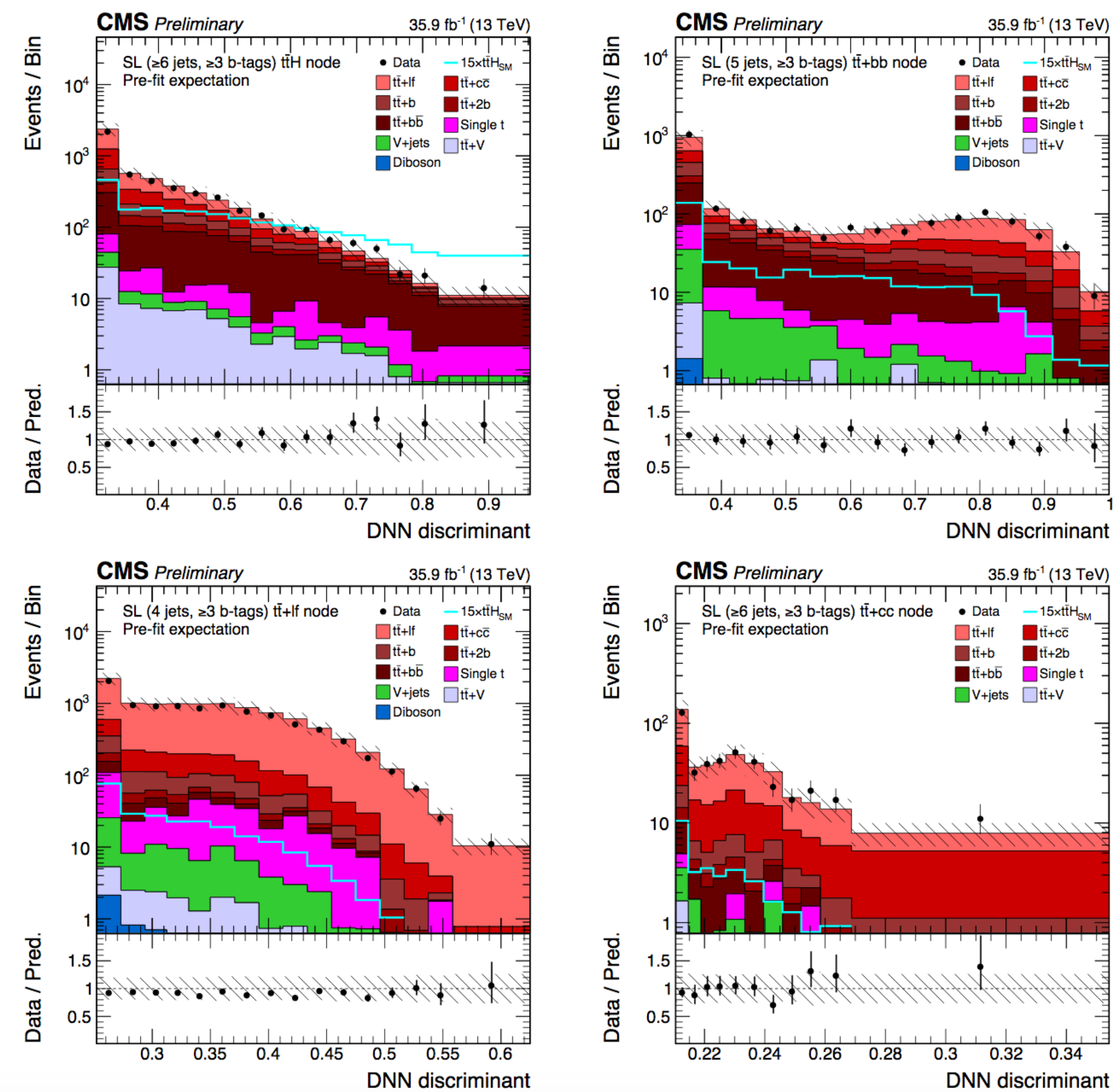


Increased sensitivity of analysis with BDT/NN signal extraction.
Would require more data otherwise.

Multi-category Classification



Slide M. Rieger

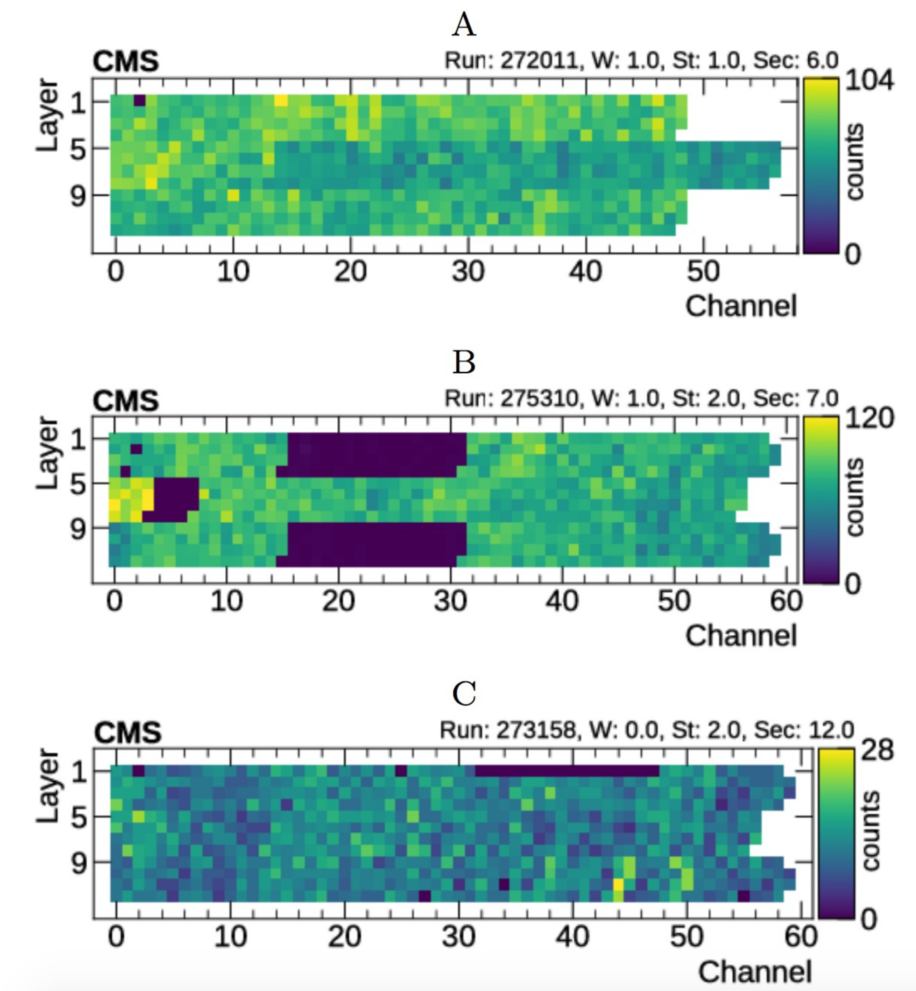


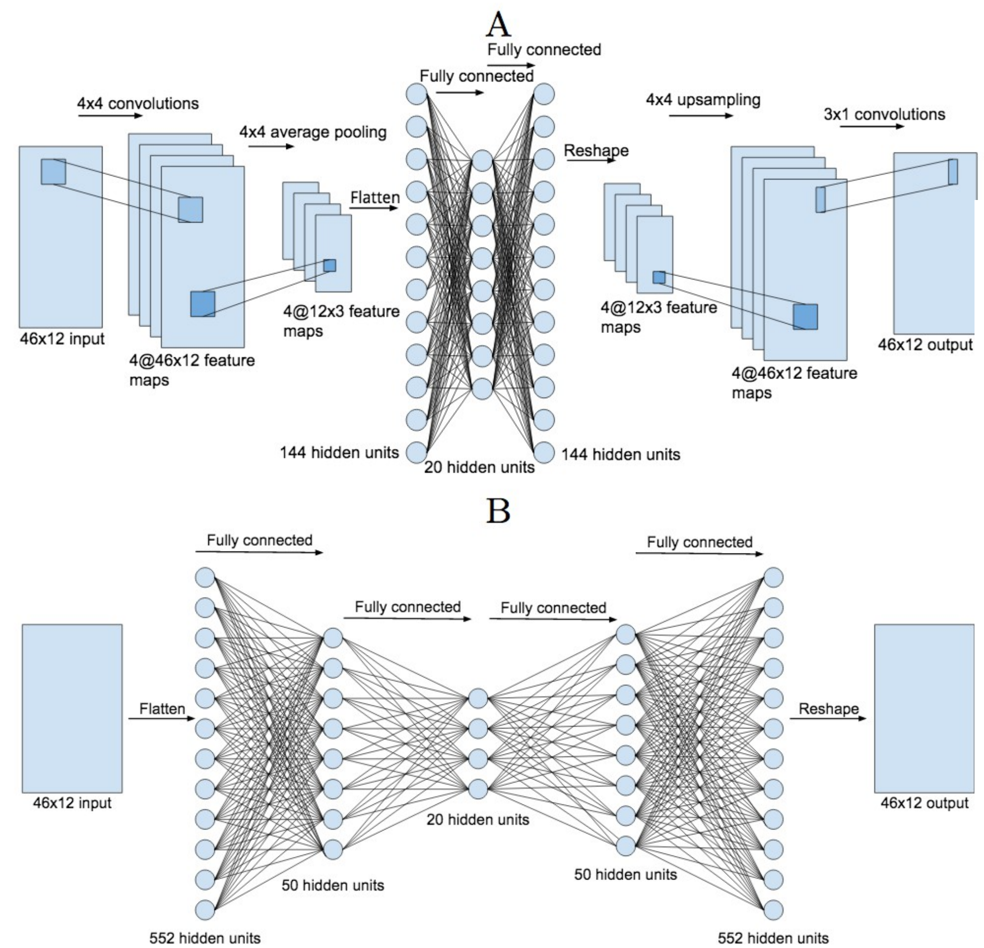
https://cds.cern.ch/record/2308267

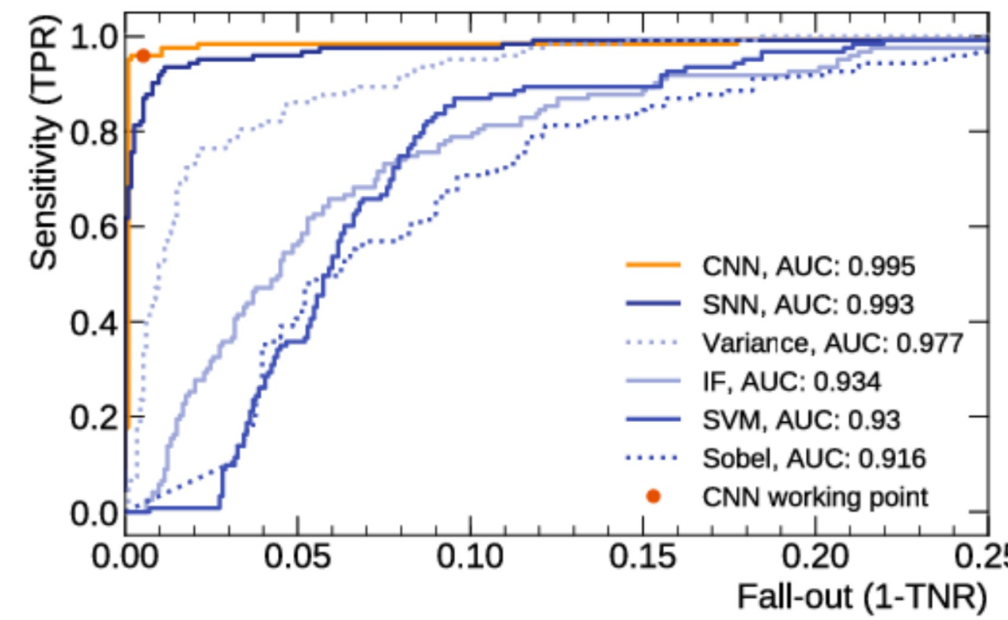
Regular analysis fit categories sub-divided using DNN output nodes for added sensitivity.

Monitoring with ML

Muon DT Quality Monitor







Unsupervised and supervised methods to identify alarming patterns in the muon drift tubes chambers.

https://arxiv.org/abs/1808.00911

Take Home Message

Deep Learning primarily deployed in Jet tagging within CMS.

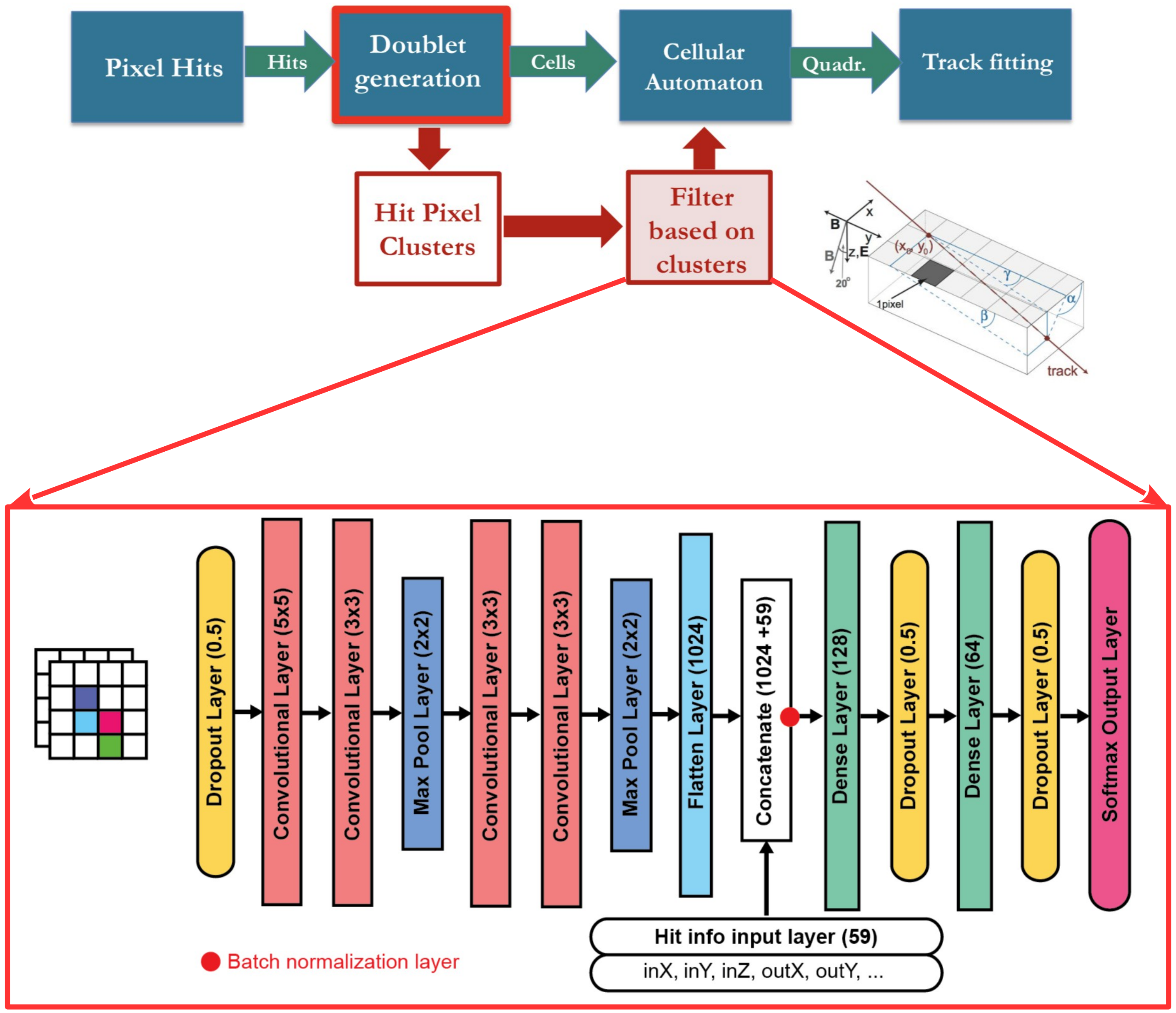
Increased use of neural networks in analysis.

Emerging applications in other areas of doing physics at CMS.

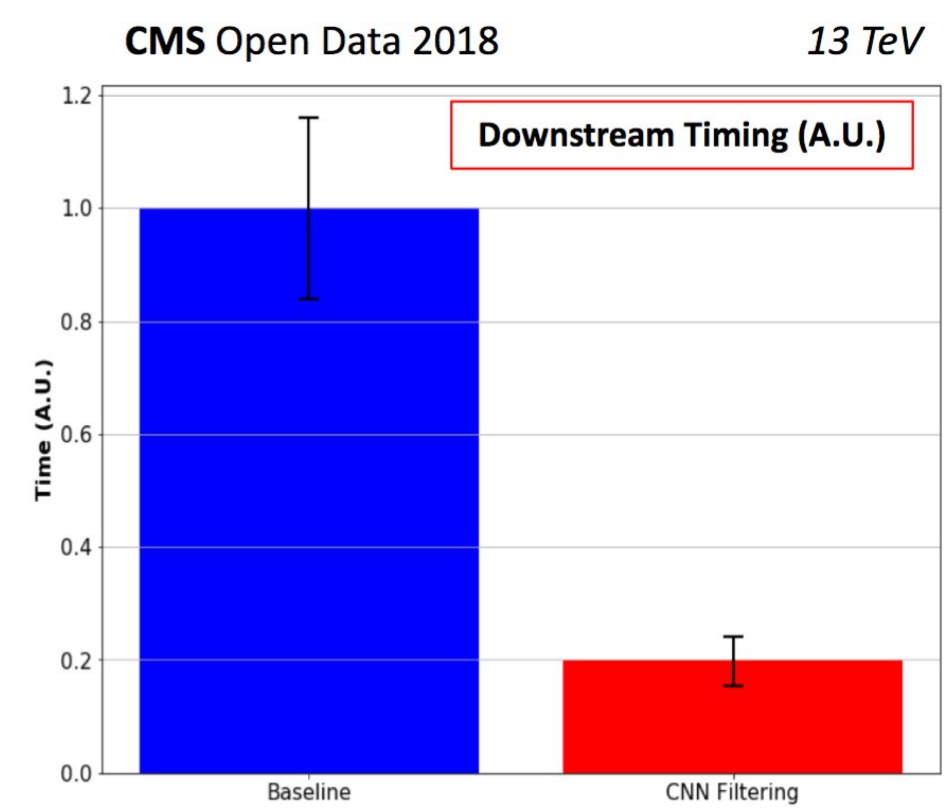
Machine Learning R&D in CMS

Charged Particle Tracking R&D

Seed Cleaning

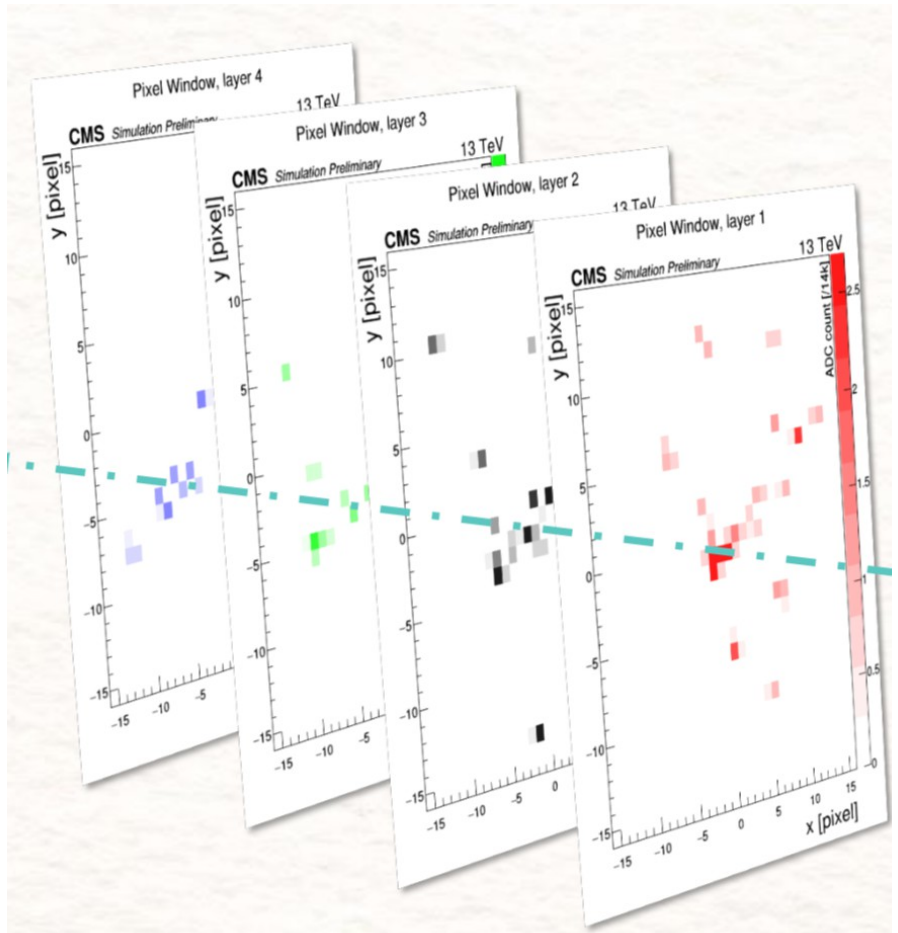


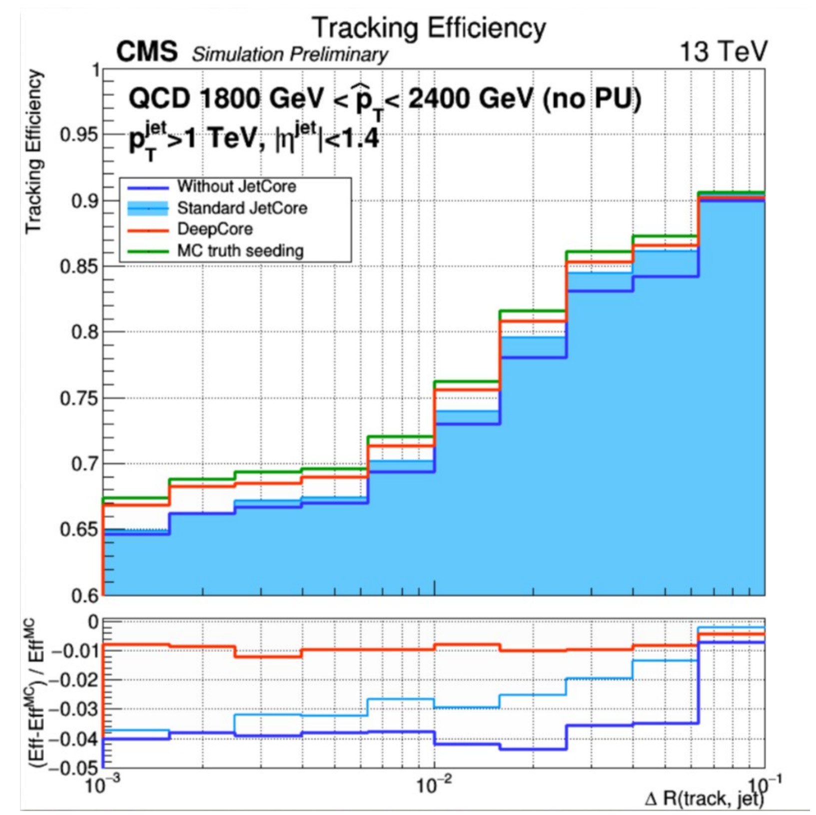
- Categorization of hits doublet using the pixel cluster shapes as input
- Significantly reduce timing in pattern recognition



https://indico.cern.ch/event/742793/contributions/3298727

Seed Finding in Jets

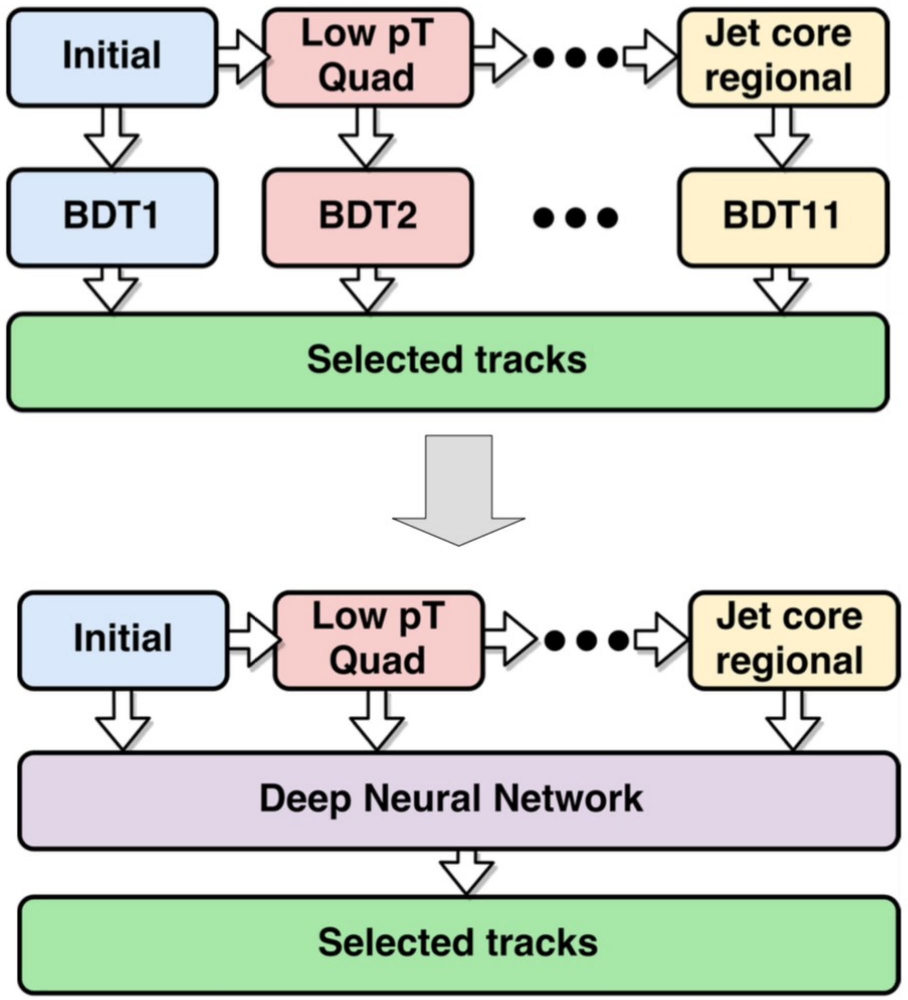


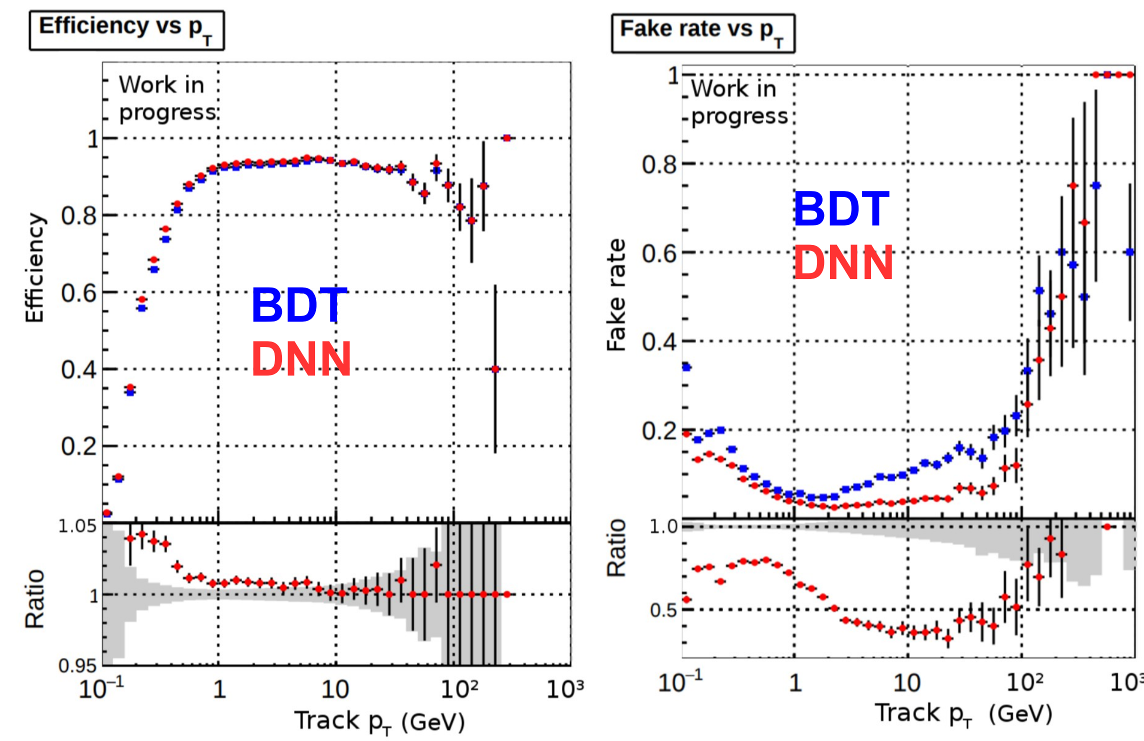


- Predict tracklets parameters from raw pixels using CNN
- Approaching the maximum performance

https://indico.cern.ch/event/742793/contributions/3274301/

Track Quality with DNN



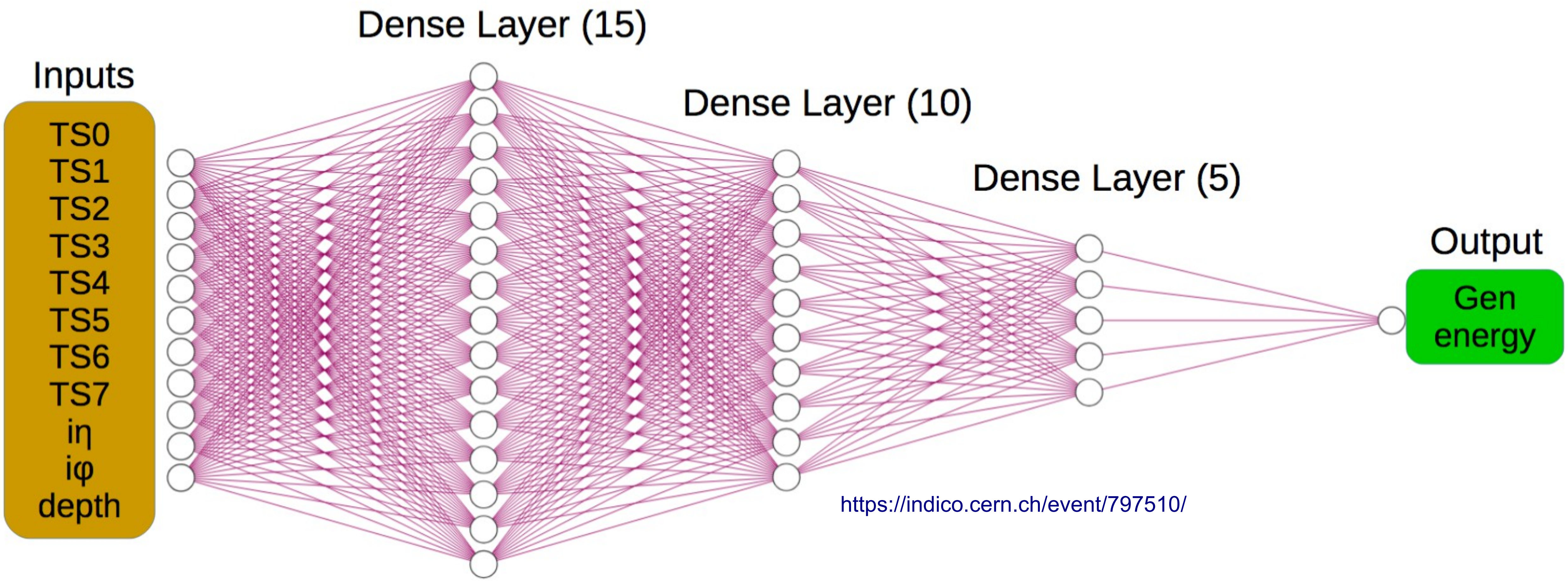


Simplifies and improves track selection within the scope of CMS iterative tracking

https://indico.cern.ch/event/658267/contributions/2813693/

Calorimeter – Jet R&D

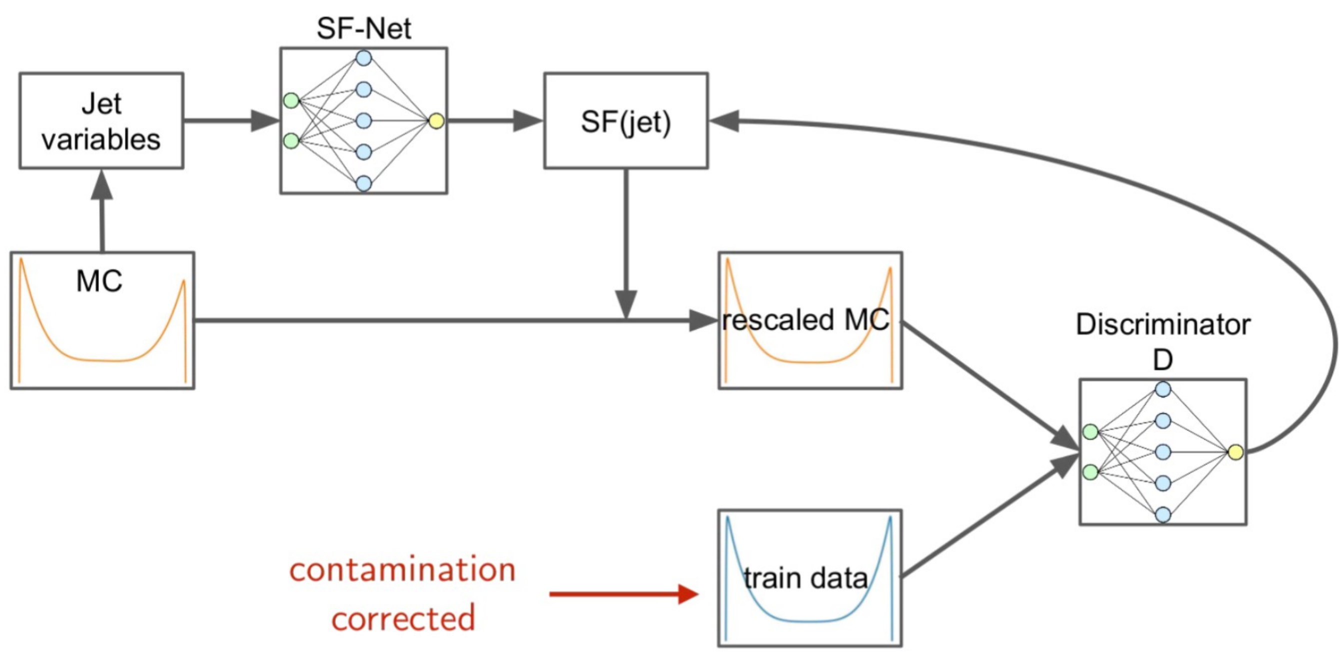
HCAL Energy



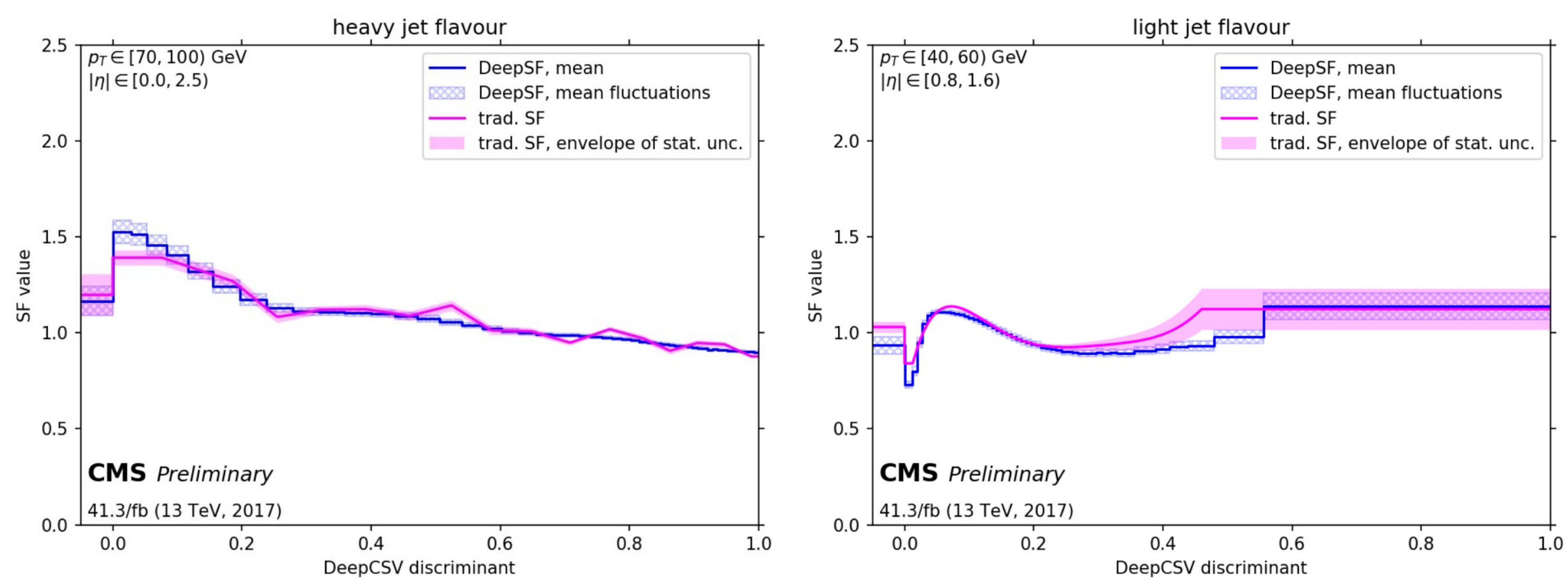
https://indico.cern.ch/event/797510/

Learn the pre-pileup energy deposition in a regression from the sampled pulse shape.

Tagging Scale Factor



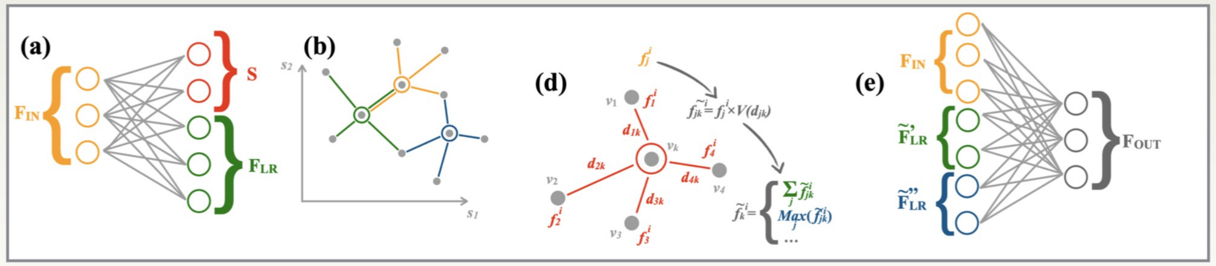
Learn per-jet data/MC scale factor using adversarial technique.



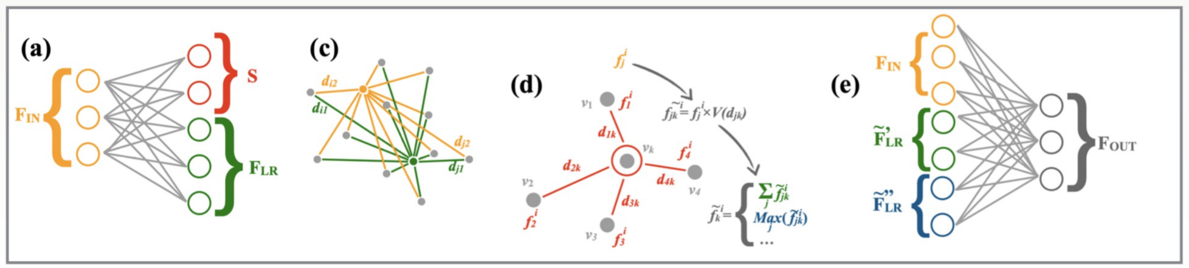
https://cds.cern.ch/record/2666647

HGCal Reconstruction

- GravNet

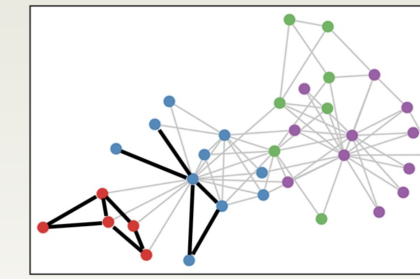


- GarNet

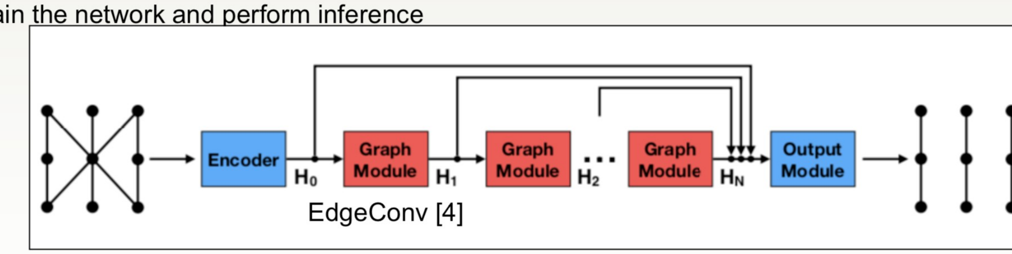


S.R. Qasim, J.K, Y. Iiyama, M Pierini arXiv:1902.07987, EPJC

- Inspired by HEP.TrkX [1,2], edge classifiers can overcome the problem
- Objects appear as vertices that are connected to each other, but not connected to others



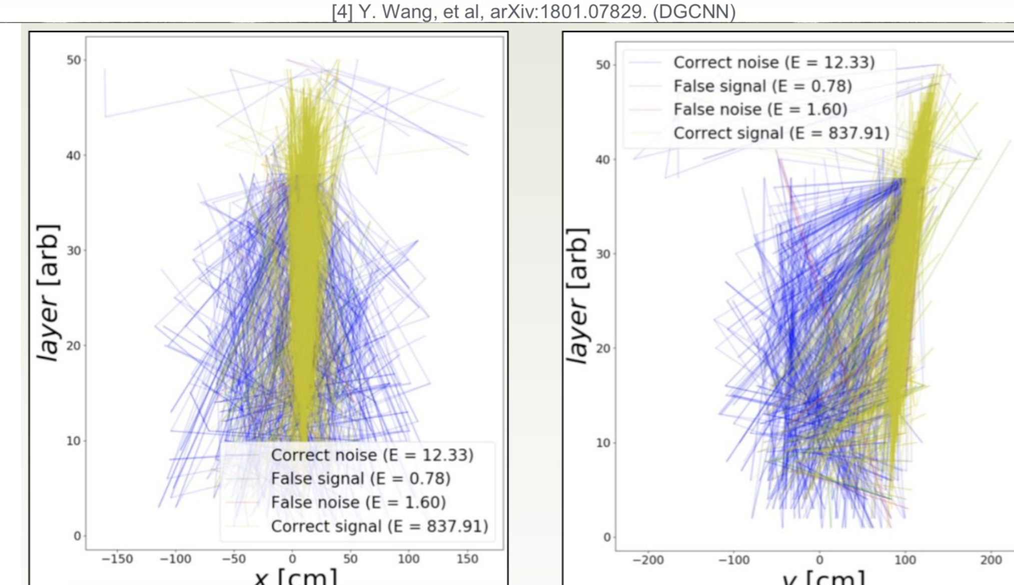
- Edges can carry additional information like particle ID
- Recipe [3]:
  - Pre-define a graph containing all possibly true edges (e.g. neighbours within a sphere)
  - Train the network and perform inference



  - Select edges with a predicted probability of more than 0.5 to be true as connections

[1] S. Farrel et al, arxiv:1810.06111,
[2] 10.1051/epjconf/201715000003
[3] X. Ju et al, https://ml4physicalsciences.github.io/files/NeurIPS_ML4PS_2019_83.pdf
[4] Y. Wang, et al, arXiv:1801.07829. (DGCNN)

Use of graph models to perform reconstruction in the high granularity calorimeter.
Node clustering, Edge classification, node segmentation, ...

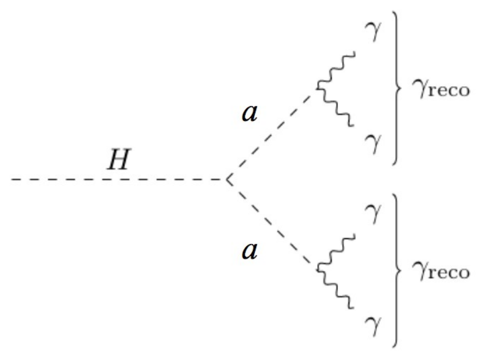


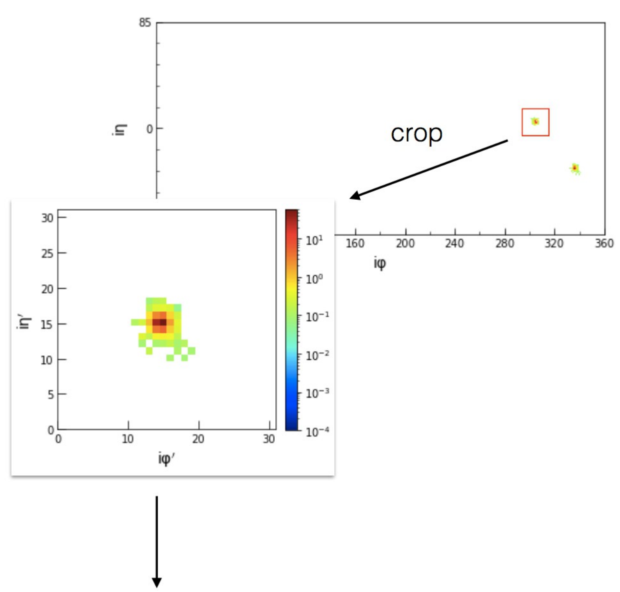
https://indico.cern.ch/event/847990/ *Slide J. Kieseler*

end-2-end Mass Regression

Search for exotic Higgs decay to light pseudoscalar, *a*

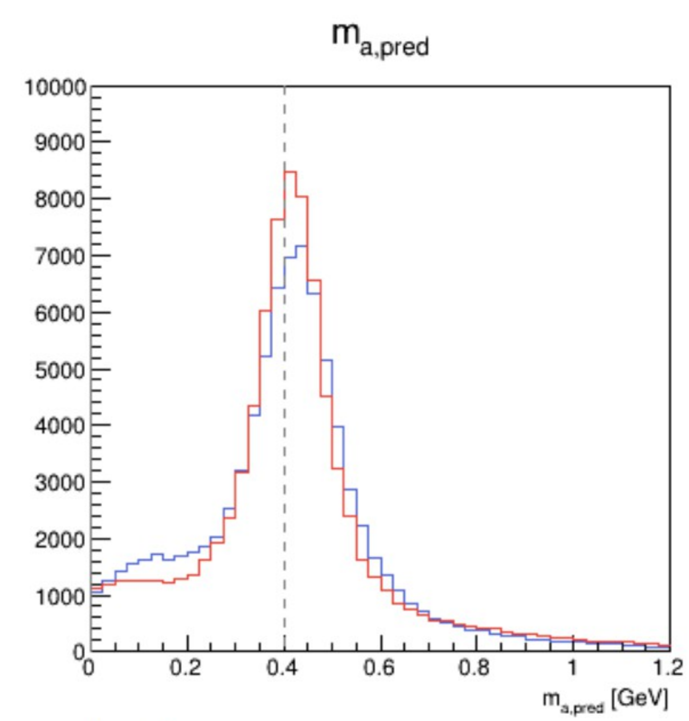
- H➝a+a➝2γ+2γ
- For sufficiently low m_a (≲ 1 GeV) each of the 2γ can be reconstructed as 1 merged γ, giving similar signature as SM H➝γγ

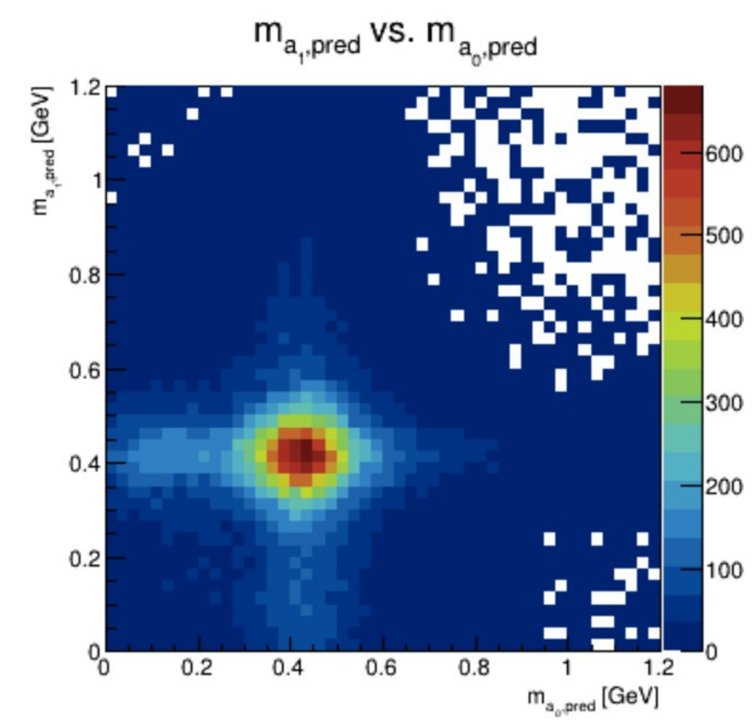




Slide M. Andrews

Learn the a/di-photon mass from the energy deposition at the Ecal surface. Unprecedented reach at low mass.

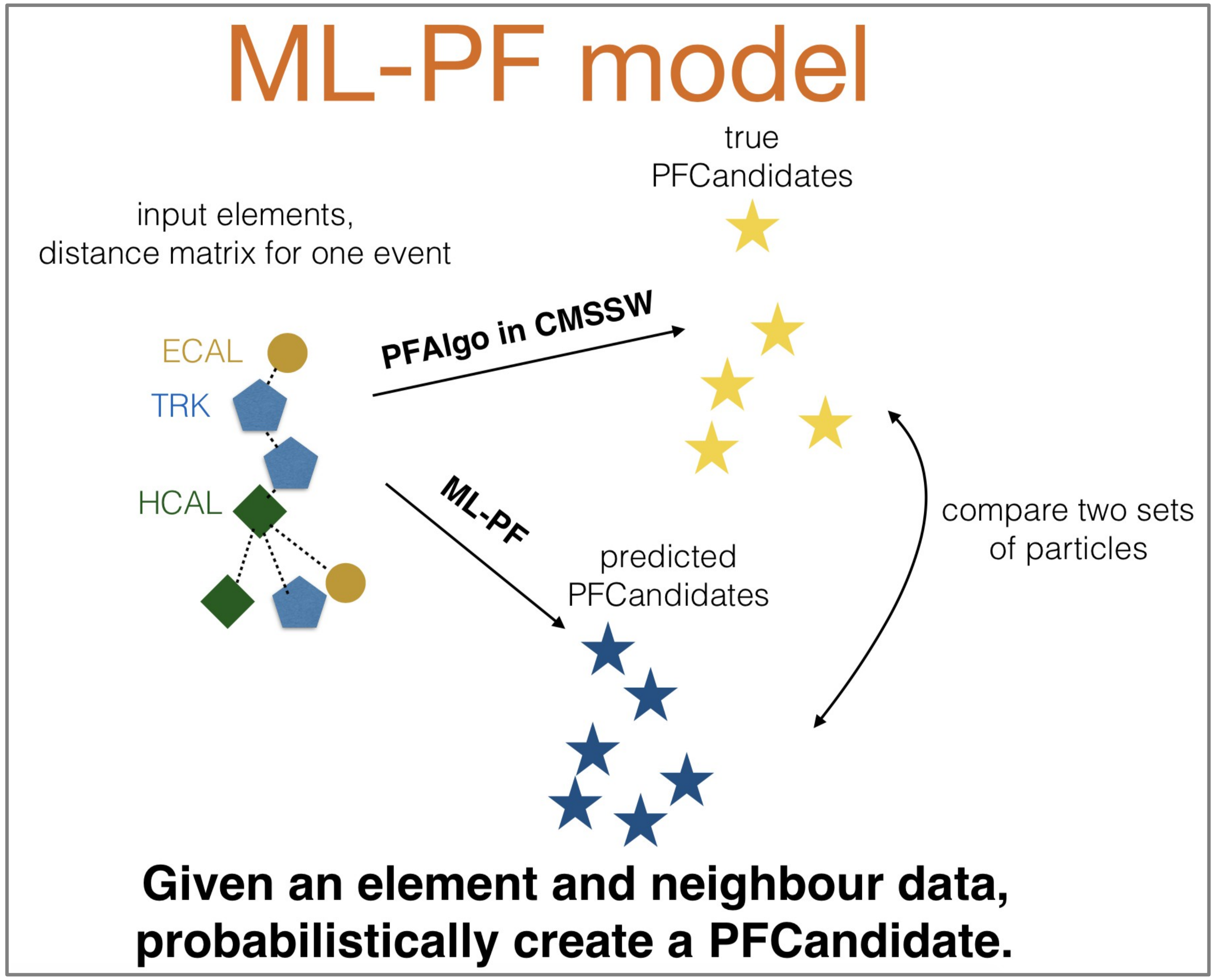




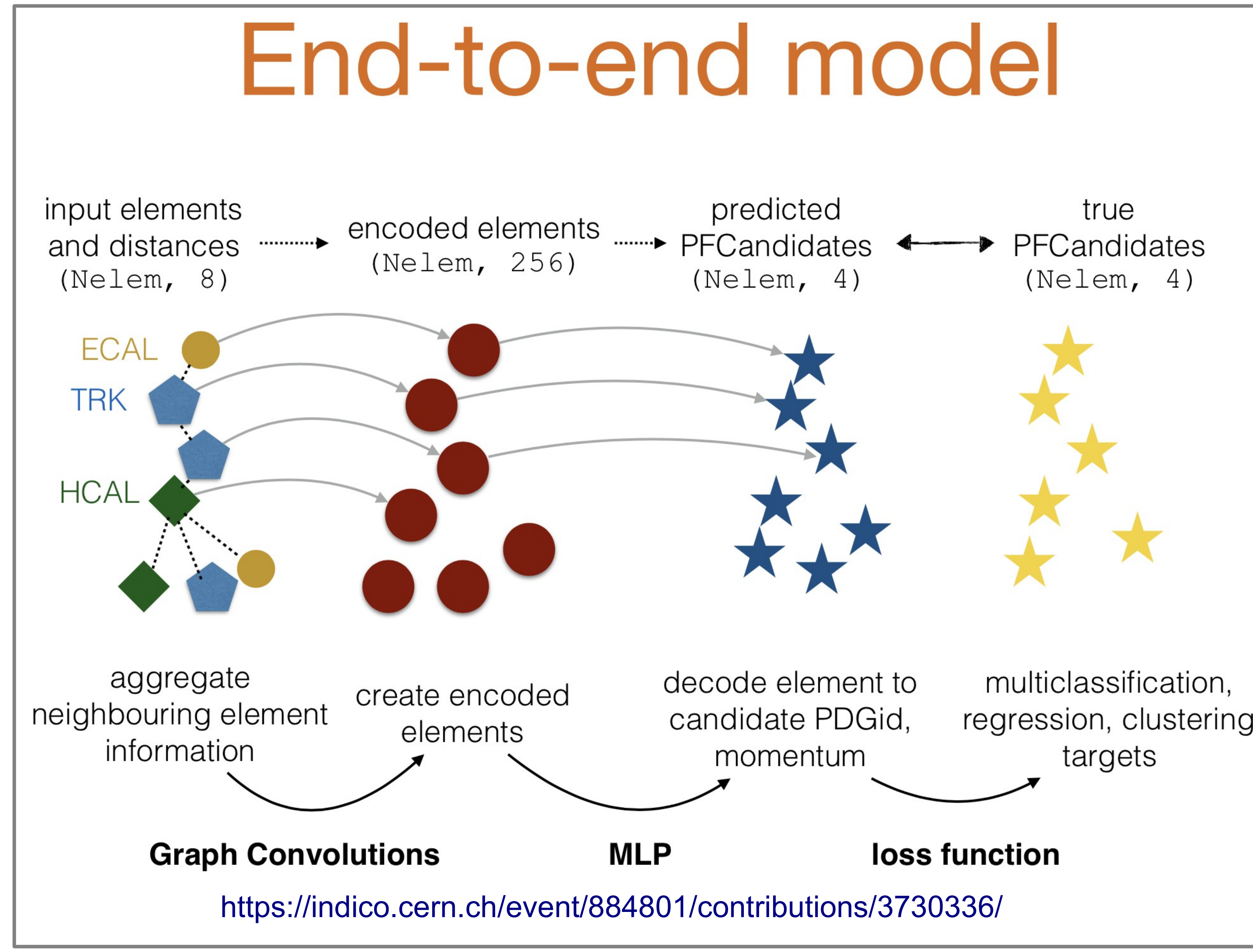
leading-p_T γ
sub-leading-p_T γ

https://indico.cern.ch/event/798721/contributions/3464782/

Particle Flow Reconstruction



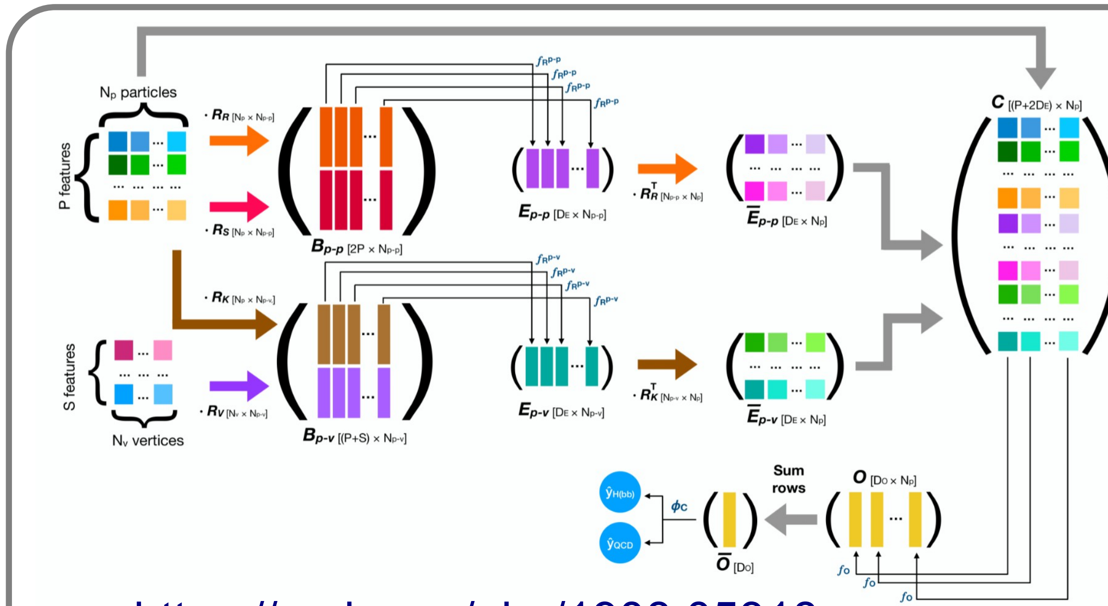
Slide J. Pata



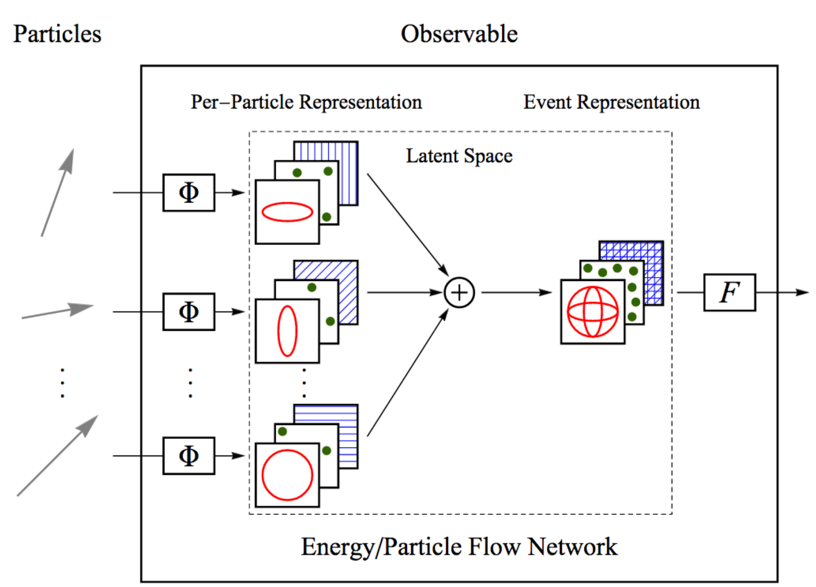
Multiple possible objective for applying machine learning for particle flow.
Graph network appears to be the most appropriate.

Particle-Cloud Jets

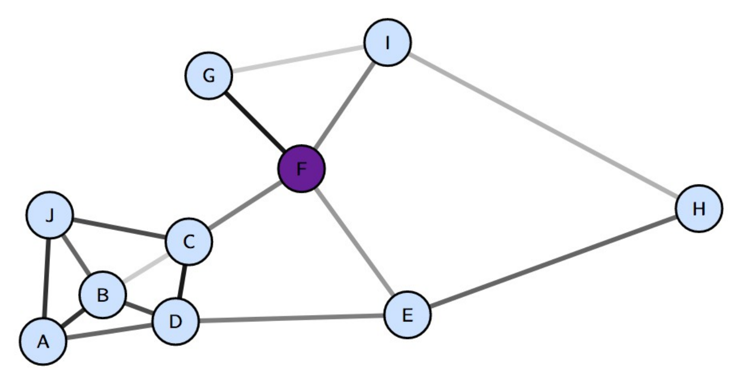
- Particle-flow jets are collection of reconstructed particles
- Graph / point-cloud representation is rather natural
- Connectivity of the graph depends on the model



https://arxiv.org/abs/1908.05318
https://arxiv.org/abs/1909.12285

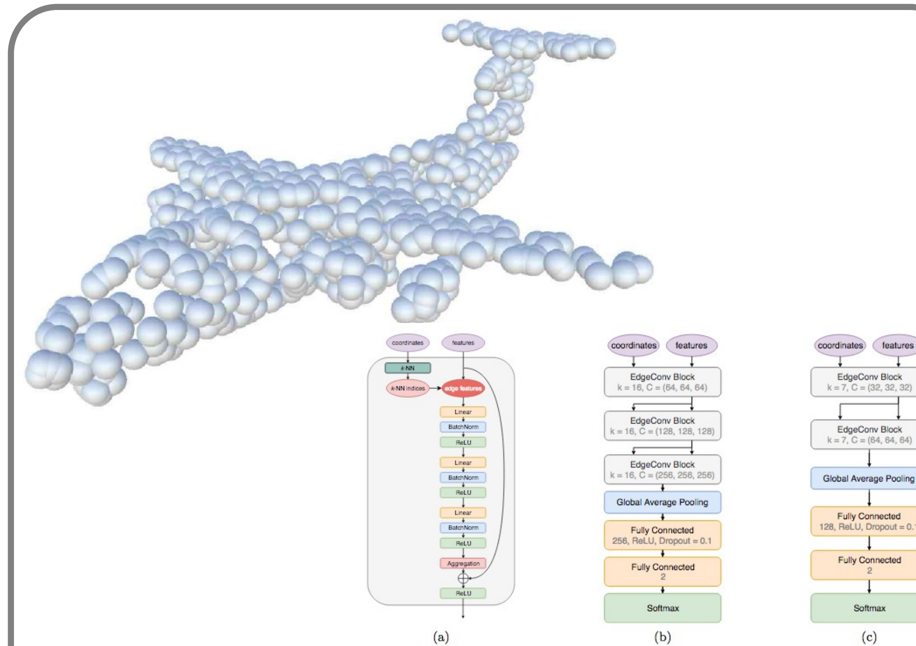


https://arxiv.org/abs/1810.05165



$$\tilde{m}_j^t = f(h_j^{t-1})$$
$$m_{j\to i}^t = \sigma(A_{ij}\tilde{m}_j^t)$$
$$h_i^t = \mathrm{GRU}(h_i^{t-1}, \Sigma_j m_{j\to i}^t)$$

https://tinyurl.com/y44ggt5b



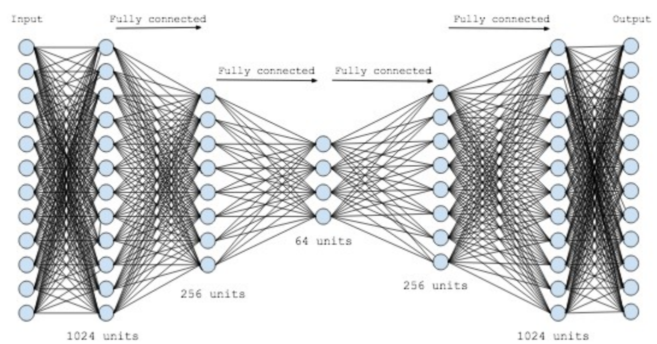
https://arxiv.org/abs/1902.08570

Monitoring R&D

Data Quality Monitoring

Chosen Autoencoder Architecture

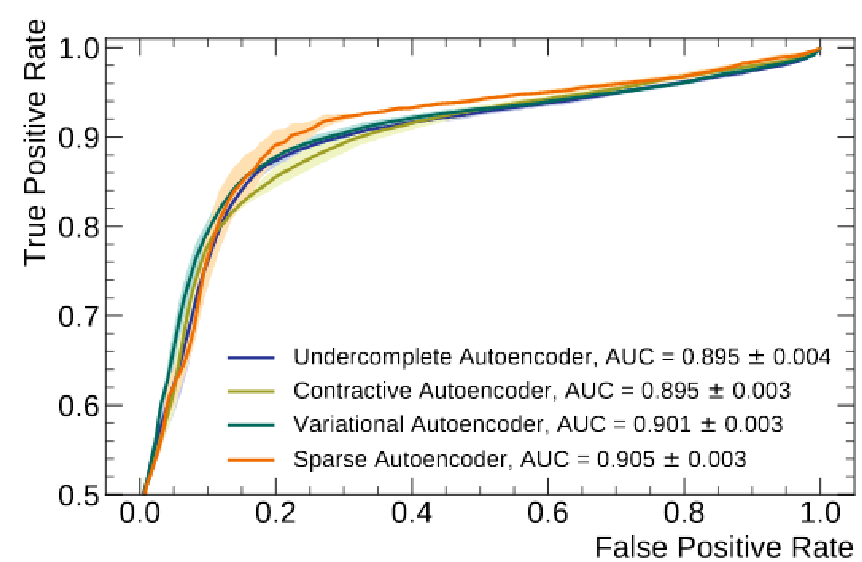
- Trained with *Keras/TensorFlow*.
- *Adam optimizer* (lr$=0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.7$) and *early stopping* (patience $=$ 32 epochs).
- Trained to minimize *mean squared error* between input vector and the output one: $\frac{1}{n}\sum_{i=0}^{n}(X_i - \hat{X}_i)^2$.
- Activations: *parametric rectified linear units*.



Proposed autoencoder architecture

Semi-supervised AD: Results

- Test set chosen gives representative values for ROC AUC.
- Anomaly score is the average reconstruction error squared over 100 worst reconstructed features $TOP100 = \frac{1}{100}\sum_{i=1}^{100} sorted(X_i - \hat{X}_i)^2$.
  - Contributions from well behaving features are irrelevant.



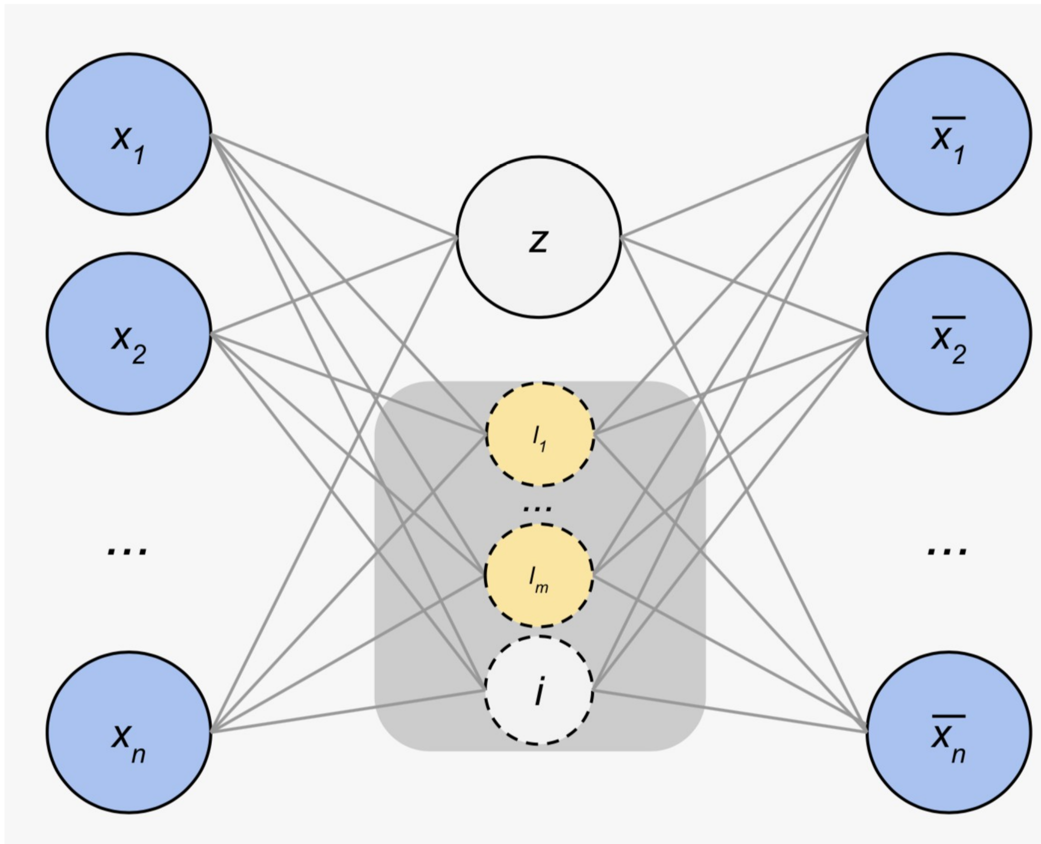
Slide A. Pol

Performance of different AEs

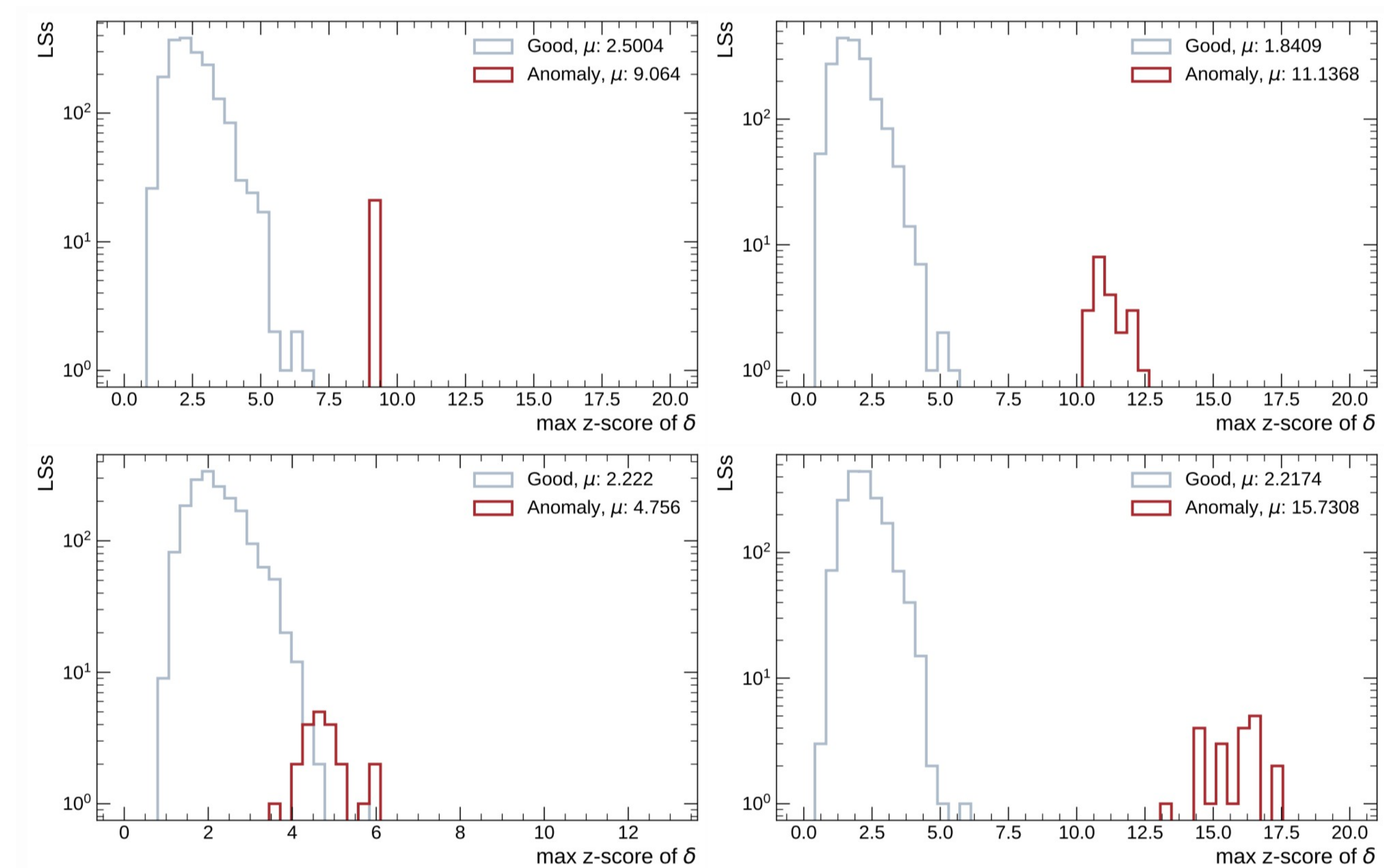
Catch anomalies in data taking using auto-encoder of hundreds of features

https://indico.cern.ch/event/708041/contributions/3276189/

Trigger Rate Prediction



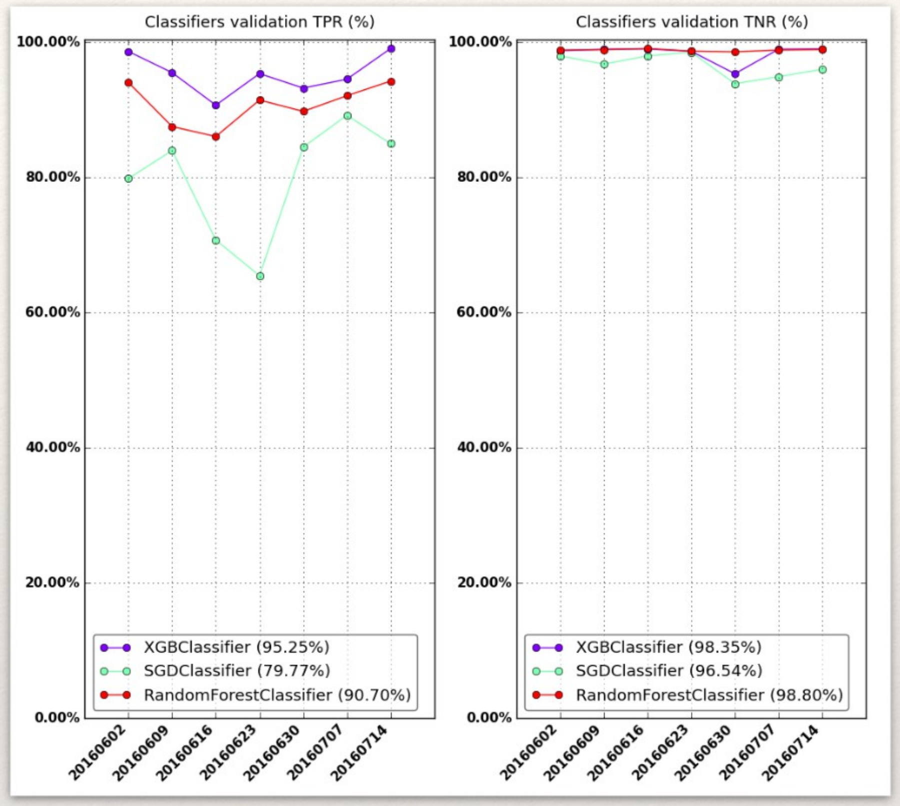
Detect deviation of trigger rate using variational auto-encoder on high level trigger rate, and L1 trigger rate in latent space



https://indico.cern.ch/event/708041/contributions/3276197/

Operation
R&D

Data Popularity



Performed studies with various classifiers:

RandomForest, SGD, XGboost

Found similar results with

SparkML

TPR=TP/(TP+FN)
TNR=TN/(TN+FP)

MINIAOD were introduced

in mid 2014

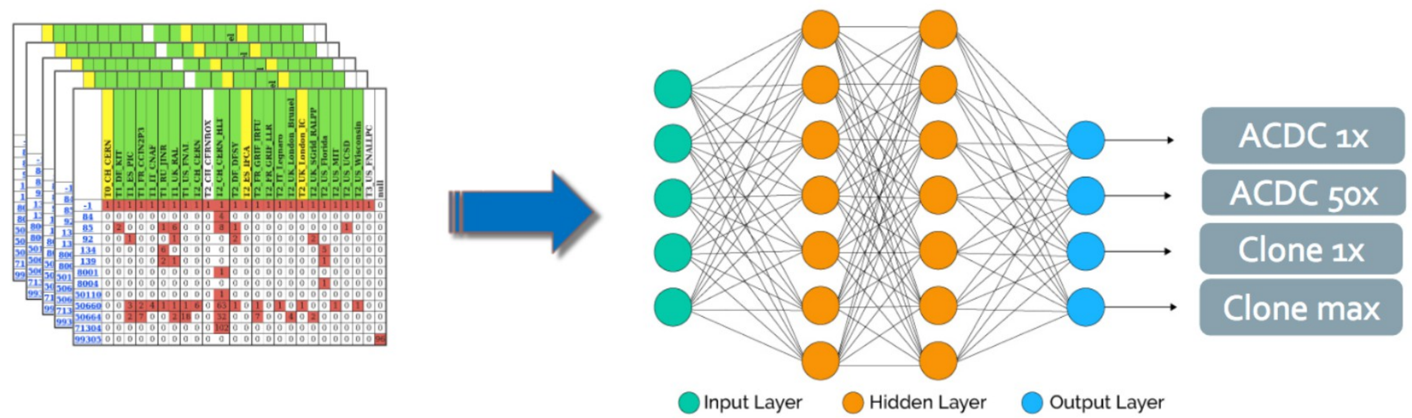
https://arxiv.org/abs/1602.07226

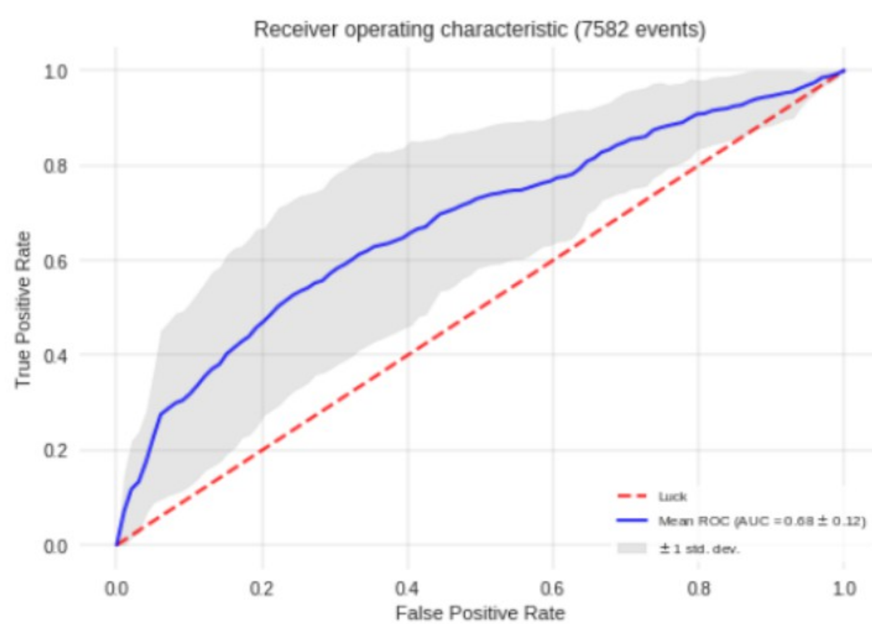
Data tier	TPR	TNR	FP	FN
AOD	0.97±0.05	0.99±0.02	0.005±0.011	0.015±0.029
AODSIM	0.93±0.13	0.99±0.02	0.008±0.016	0.021±0.045
MINIAOD	0.11±0.32	0.99±0.02	0.014±0.026	0.001±0.007
MINIAODSIM	0.49±0.48	0.99±0.02	0.009±0.016	0.007±0.031
USER	0.93±0.15	0.98±0.02	0.014±0.021	0.011±0.023

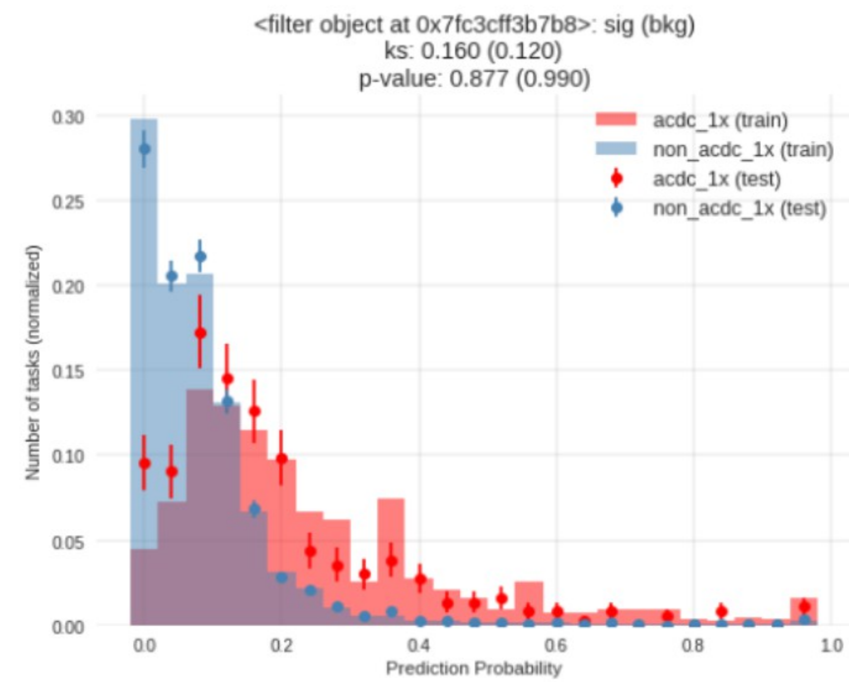
Slide V. Kuznetsov

R&D on predicting popularity of analysis datasets, in a view to a more efficient data placement.

Predicting Operator's Action







Challenging task of predicting the operator's action from the information they are provided with.

https://indico.cern.ch/event/587955/contributions/2937424/

Outlooks

ML in Operation

- Potential application of machine learning to operation and reduce manpower needs, burden on operation, expedite production of data from detector and simulation
  - Detector control
  - Data quality
  - Computing operation
  - ...

ML in Trigger

- Potential application of machine learning in data acquisition and triggering
  - Anomaly detection in data taking
  - Unsupervised new physics mining
  - Signal specific trigger paths
  - Background and trigger rate reduction
  - ...

ML in Reconstruction

- Event reconstruction, online and offline may be in part replaced by surrogate models (approximate and faster) or by novel algorithm (improved performance)
  - Charged particle tracking (GNN, vertexing, ...)
  - Calorimeter reconstruction (local, clustering, ...)
  - Particle flow (GNN, ...)
  - Particle identification (boosted jets, isolation, ...)
  - Pileup mitigation
  - Energy regression (end-2-end, ...)
  - ...

ML in Simulation

- Producing events through full/fast simulation is extremely computing intensive, and limiting somehow how the Physics reach of the experiment. ML may help reducing the load
  - Calorimeter shower surrogate simulator
  - Analysis level simulator
  - Pile-up overlay generator
  - Monte-carlo integration
  - ML-fast-simulation
  - Invertible full-simulation (probprog, ...)
  - ...

ML in Analysis

- ML has entered analysis long ago. Novel technics have been published in HEP outside of experiments. Hard work of bringing proof of concept to analysis
  - Likelyhood free inference
  - Background modeling technics
  - Kinematic reconstruction
  - Unsupervised new physics search
  - ...

ML Methodology

- ML as tool works very well empirically. Lots of understanding is still required at the theoretical level. Science and physicists may help several items
  - Network compression
  - Transfer learning
  - Uncertainty quantification
  - Interpretability
  - Mechanism of convergence
  - Incorporating domain knowledge
  - ...

Take Home Message

Many applications of deep learning for high energy physics.

Many exciting projects to be done within CMS.

Keep the analysis/physics in line of sight.

Summary

→Machine learning as a personal asset to acquire

→Field developing very rapidly

→Powerful tool to boost and full the science

→Increasing amount of deep learning applications for HEP

→Get involved with CMS Software, THEN apply ML

...

→Do not loose sight of the physics/analysis

Resource

Groups
- CML Machine Learning Forum : https://indico.cern.ch/category/9376/
- Hypernews : https://hypernews.cern.ch/HyperNews/CMS/get/machine-learning.html
- IML : https://iml.web.cern.ch/
- CERN Openlab : https://indico.cern.ch/category/3169/
- Fermilab machine learning : https://machinelearning.fnal.gov/
- Fast machine learning lab : https://fastmachinelearning.org/
- AMVA4NP : https://amva4newphysics.wordpress.com
- Dark Machines : http://darkmachines.org/
- NNPDF : http://nnpdf.mi.infn.it/

Conference
- Data Science in HEP series : https://indico.fnal.gov/event/13497/ (last)
- Hammers and Nails : http://www.weizmann.ac.il/conferences/SRitp/Aug2019/ (last)
- ML4JET Workshop series : https://indico.cern.ch/event/809820 (last)
- ACAT : https://indico.cern.ch/category/7679/
- Machine Learning and the Physical Science : https://ml4physicalsciences.github.io (last)
- CERN DS-IT Seminars : https://indico.cern.ch/category/9320/

Training
- CMS Data Analysis School : https://lpc.fnal.gov/programs/schools-workshops/cmsdas.shtml
- mPP deep learning training : https://indico.cern.ch/category/10066/
- Machine learning in High Energy Physics Schools : https://indico.cern.ch/event/838377/ (last)

Tools
- Scikit Learn : https://scikit-learn.org
- Keras : https://keras.io/
- Tensorflow : https://www.tensorflow.org/
- Pytorch : https://pytorch.org/

Journals
- Computing and Software for Big Science (CSBS) : https://www.springer.com/journal/41781
- Machine Learning: Science and Technology (MLST) : https://iopscience.iop.org/journal/2632-2153
- Big data and AI for HEP (BDAI) : https://frontiersin.org/big-data-and-ai-in-high-energy-physics

Model Prediction in CMSSW

Chronologically

- Lightweight Trained Neural Network (LWTNN)
  https://github.com/lwtnn/lwtnn
- Tensorflow (google)
  https://www.tensorflow.org/
- Apache mxnet (opensource)
  https://mxnet.apache.org/
- ONNX (Facebook and Microsoft)
  https://onnx.ai

Prediction from already trained models can be made directly in CMS Software Framework.

ONNX on the fast side of things.
TORCH api considered