Machine Learning in the CMS Experiment

- **A general overview with special emphasis on local activities**

Dirk Krücker - DESY CMS group
2.12.2019

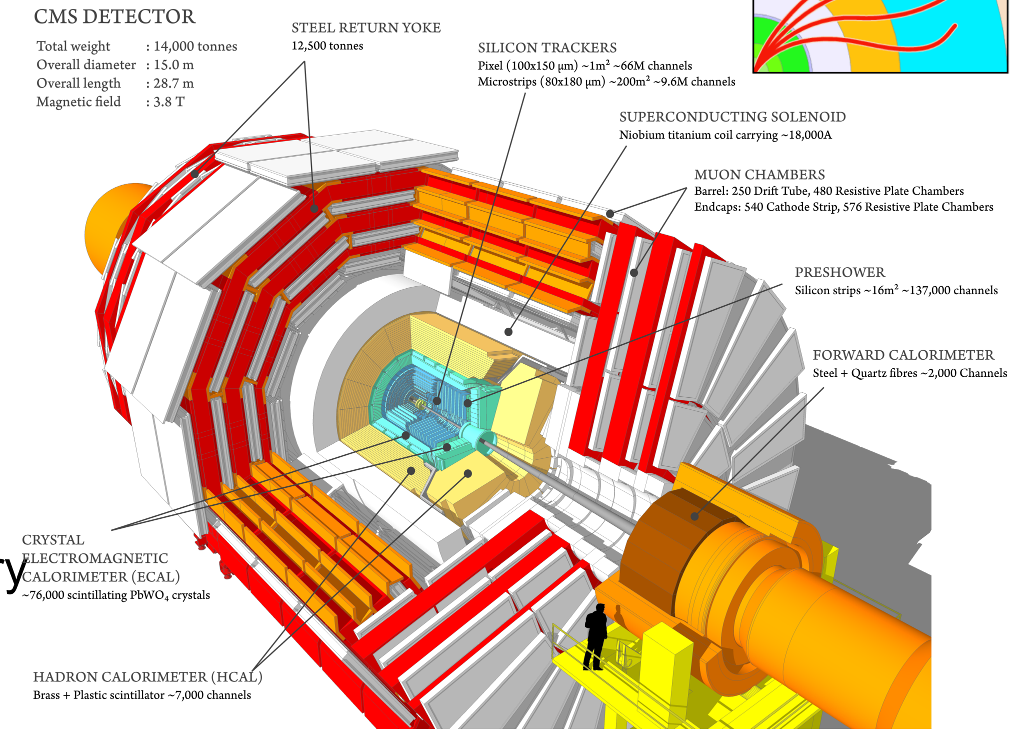
HELMHOLTZ RESEARCH FOR GRAND CHALLENGES

DESY.

CMS Experiment at the LHC collider

CMS Group HH: O(10^2)

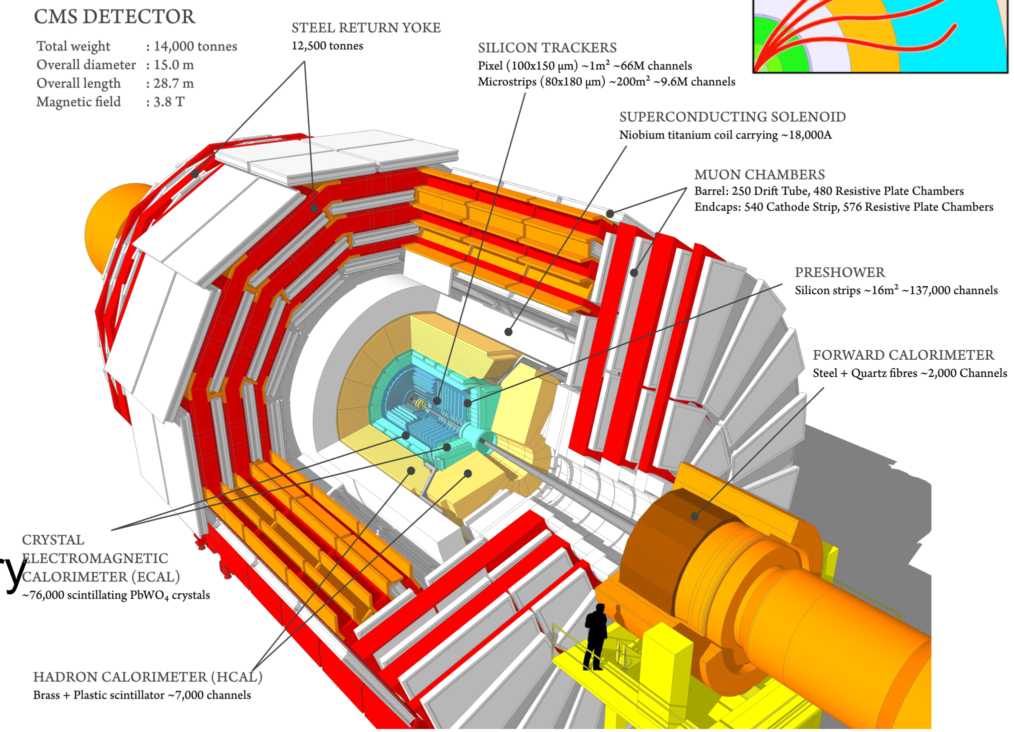
- CMS collaboration ~**4000** physicist worldwide
- Machine Learning and Deep Learning techniques are used for a long time in High Energy Physics
  - Domain specific tools, e.g. **TMVA**, mainly BDT
  - In recent years drift to common data science tools, e.g. **Tensorflow**
- **CERN** works as **hub** for the exchange of knowledge and industry contacts
  - **IML** - Inter-Experimental LHC Machine Learning Working Group
    - Monthly meetings
    - Yearly workshops: Tutorials; industry sessions, Google, Nvidia etc.
  - **CERN openLab,** especially for industry contacts and projects
  - **CMS ML forum** with monthly meeting (Exchange, expert advice and control)



CMS Experiment at the LHC collider

CMS Group HH: O(10^2)

- CMS collaboration ~**4000** physicist worldwide
- Machine Learning and Deep Learning techniques are used for a long time in High Energy Physics
  - Domain specific tools, e.g. **TMVA**, mainly BDT
  - In recent years drift to common data science tools, e.g. **Tensorflow**
- **CERN** works as **hub** for the exchange of knowledge and industry contacts
  - **IML** - Inter-Experimental LHC Machine Learning Working Group
    - Monthly meetings
    - Yearly workshops: Tutorials; industry sessions, Google, Nvidia etc.
  - **CERN openLab,** especially for industry contacts and projects
  - **CMS ML forum** with monthly meeting
    (Exchange, expert advice and control)



HEP traditionally uses full simulation of the analysis process:
Physics ⟶ Detector/Electronics ⟶ Reconstruction ⟶ Analyses
(Digital twin)
⟹ We can **train on simulation**

Overview

CMS activities

CMS is very active in applying Machine Learning and especially Deep Learning to a variety of subjects

- About ~25 dedicated ML contributions on conferences in 2019-present (not including physics analyses)
- Annual dedicated CMS Machine Learning workshops
- ML is transforming the field and the traditional workflows
  - New tools and new terminology

Main fields of applications

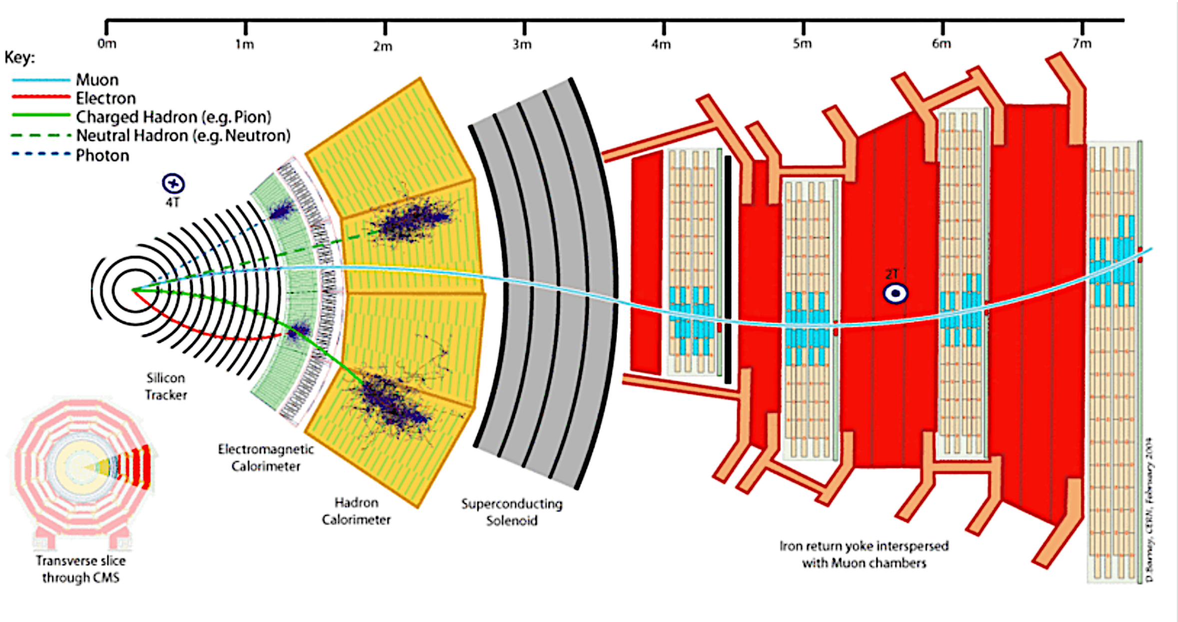
- **Object tagging** and object calibration
  especially Jets and taus are **flagship** applications
- **Physics analyses**
- In addition
  - Triggers (e.g.: NN on FPGAs)
  - Reconstruction/Calibration
  - Fast Simulation
    $\longrightarrow$ **GANs**
  - Data Quality Monitoring
  - Computing workflows

Disclaimer: some of the following slides are collect from other talks for illustration.
Here, I will not explain all details.

Jet Tagging

DeepJet

The latest approach on b-quark jet tagging

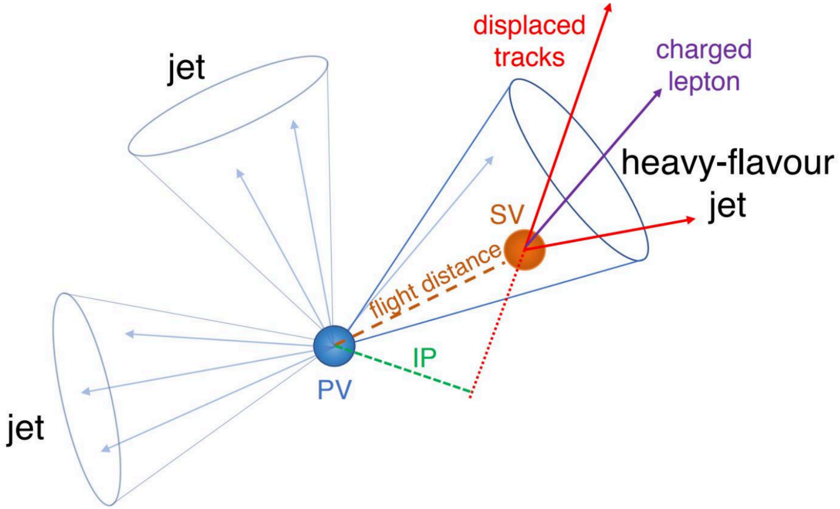


CMS uses a **Particle Flow** approach

- Identify each particle by the combined information of all sub-detector
- combine to jets of particles

Jet identification had been one of the first application of Deep Learning in HEP

- **b-quark jets** vs g/u/d/s-jets etc.
- The classical approach looks for secondary vertices
- There is plenty of information in the correlation between the individual tracks and vertices and global event characteristics

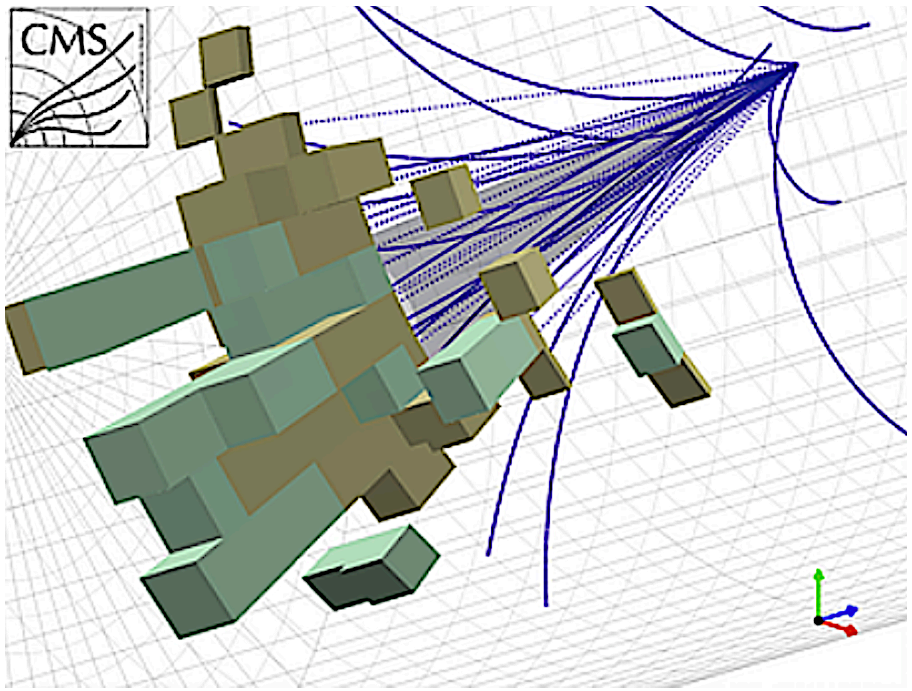


Number of tracks changes from collision to collision
*⟶ **input of changing length**, solution*

- *Recurrent networks (**RNN**)*
- *Convolution NNs (**CNN**)*

DeepJet

The latest approach on b-quark jet tagging

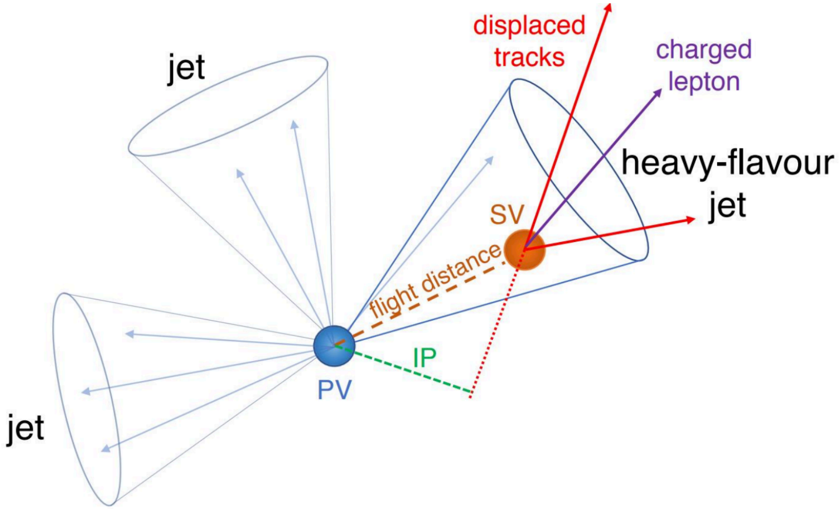


CMS uses a **Particle Flow** approach

- Identify each particle by the combined information of all sub-detector
- combine to jets of particles

Jet identification had been one of the first application of Deep Learning in HEP

- **b-quark jets** vs g/u/d/s-jets etc.
- The classical approach looks for secondary vertices
- There is plenty of information in the correlation between the individual tracks and vertices and global event characteristics

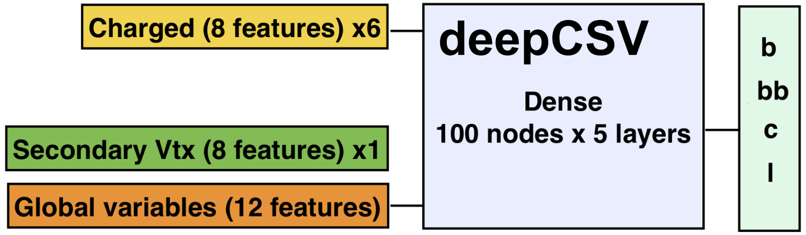


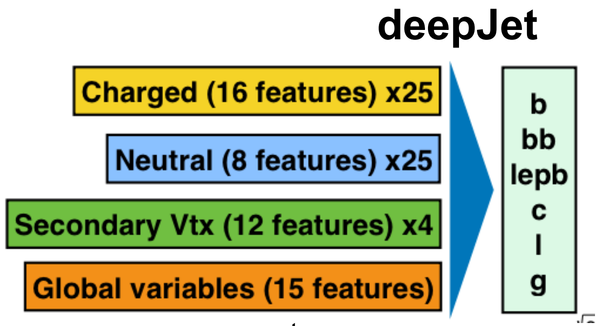
Number of tracks changes from collision to collision
*⟶ **input of changing length**, solution*
- *Recurrent networks (**RNN**)*
- *Convolution NNs (**CNN**)*

Evolution of CMS b-tagging

A short history (for more details **M. Verzetti** ML4Jets WS 2018)

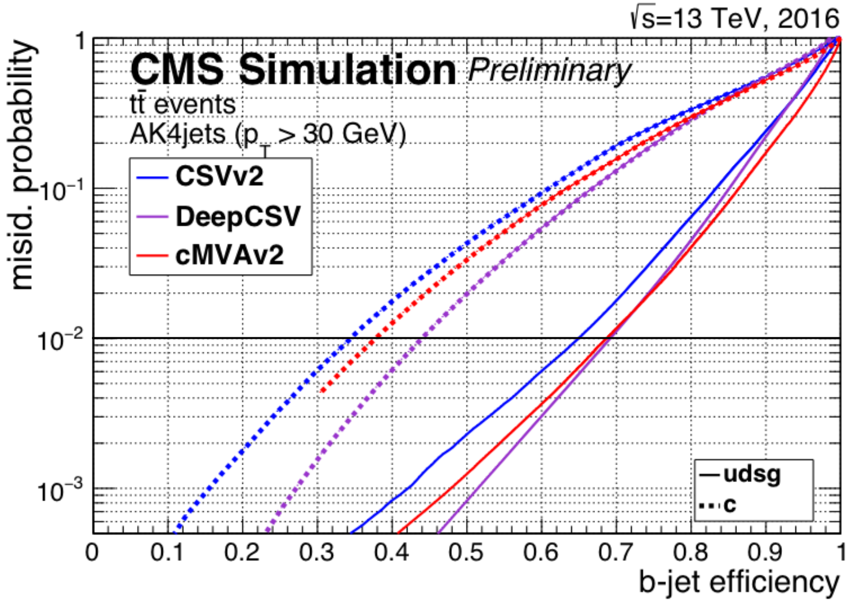
- 2015 handcrafted **CSVv2(nn)** and cMVAv2(BDT)
  - Mainly the same information
- 2017 **deepCSV**
- mid 2017 **deepJet** (DeepFlavour)
  - The gain is due to the usage of all available particle information in a deep network



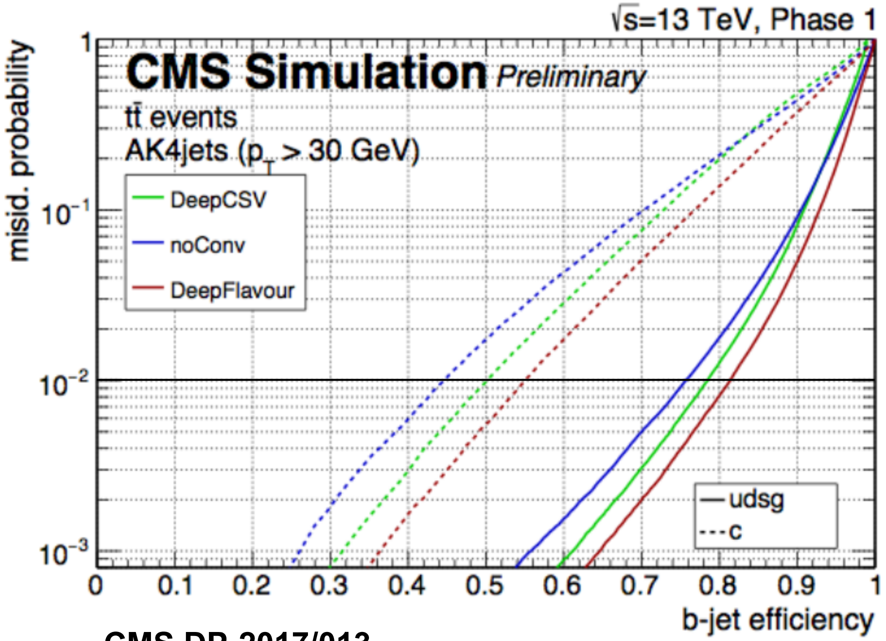


see next page

Steady development of DL taggers over the last 4 years



CMS DP-2017/005



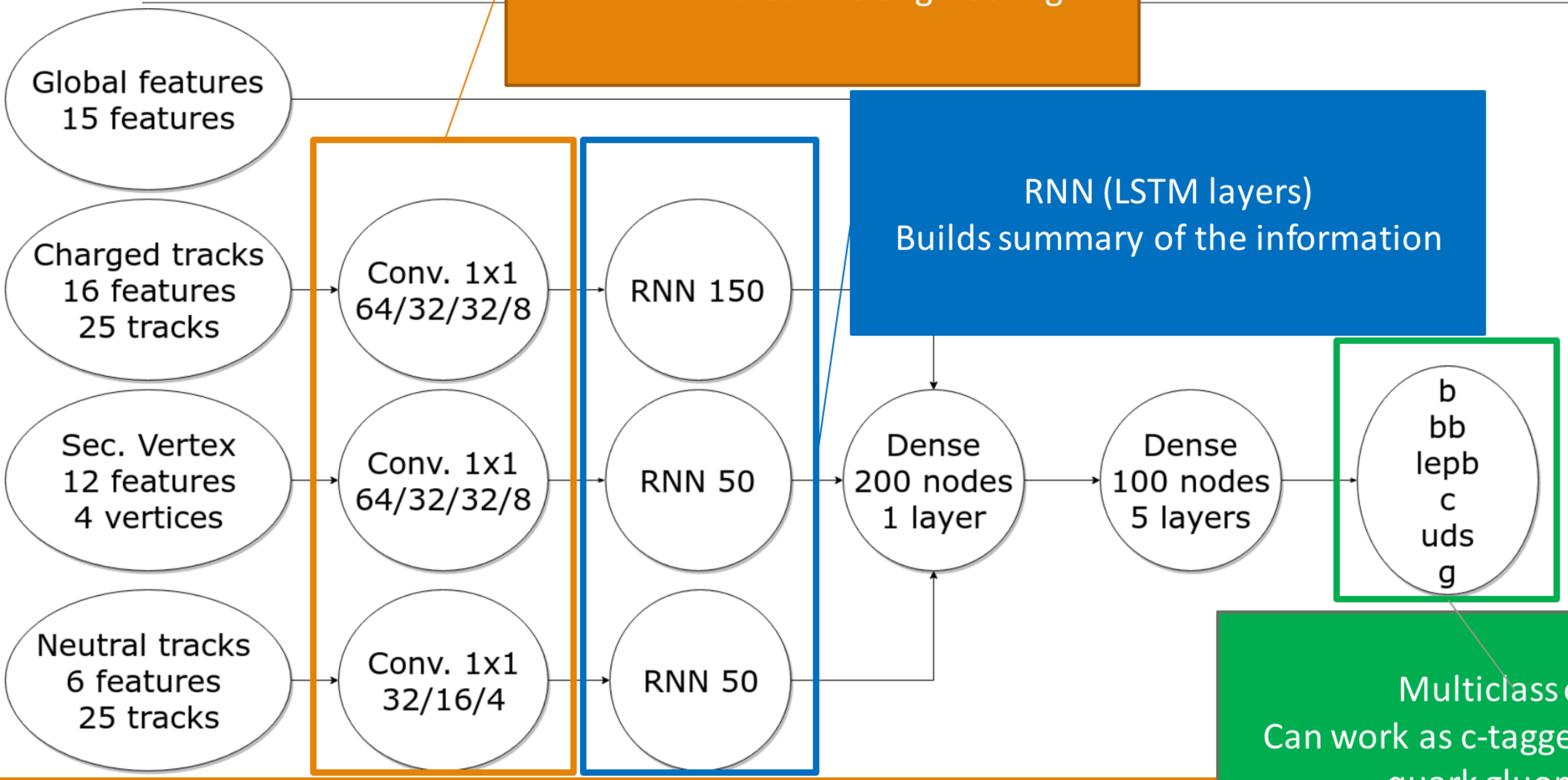
CMS DP-2017/013

DeepJet

1x1 Convolutional layers
Automatic feature engineering

RNN (LSTM layers)
Builds summary of the information

Complex machinery
with ~500k of parameters
613 input features

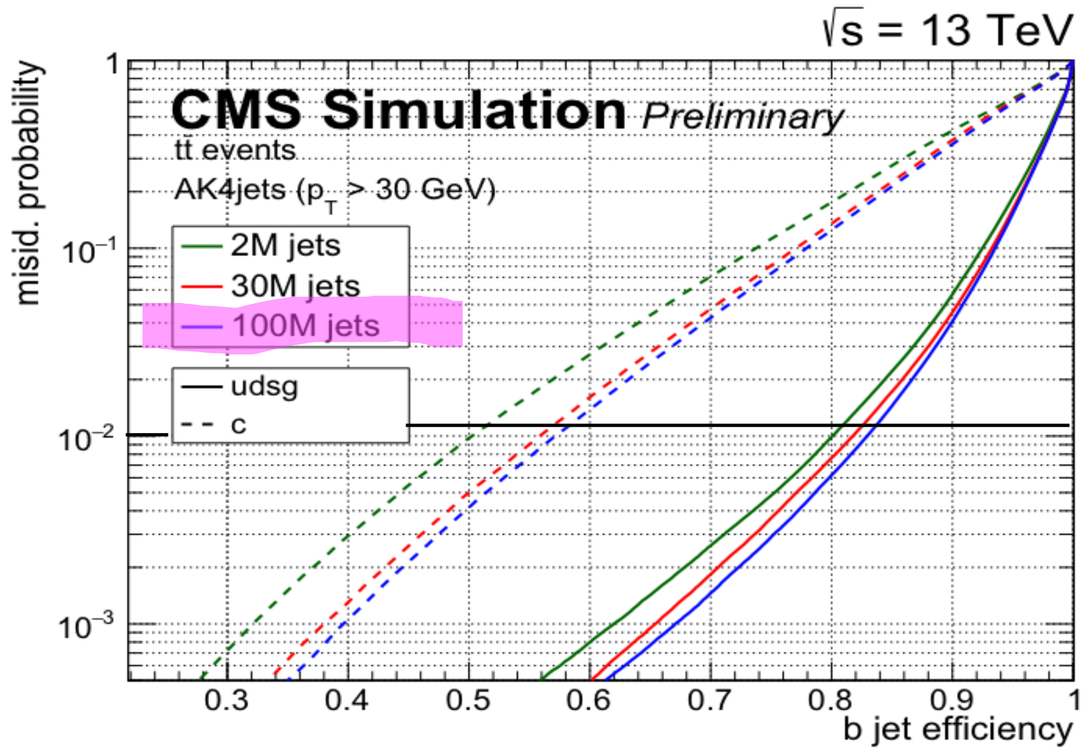


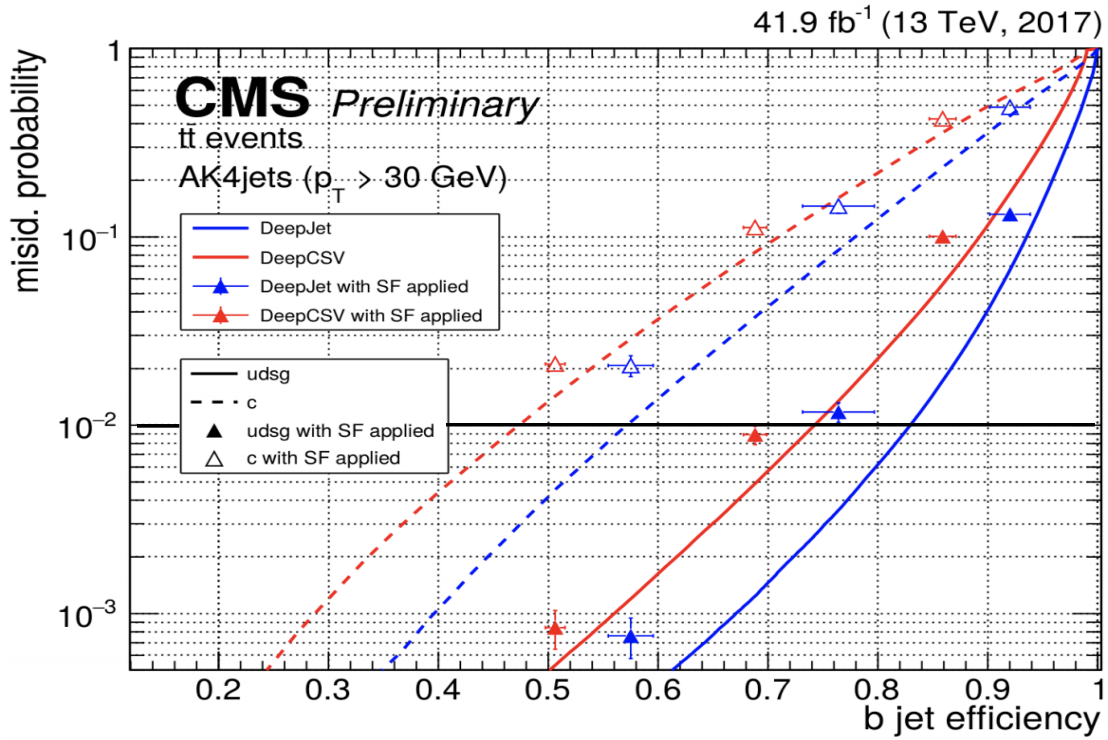
Multiclass output
Can work as c-tagger, b-tagger and quark gluon tagger

EMIL BOLS (VUB)

Training DeepJet

Big data





Training needs large amount of data

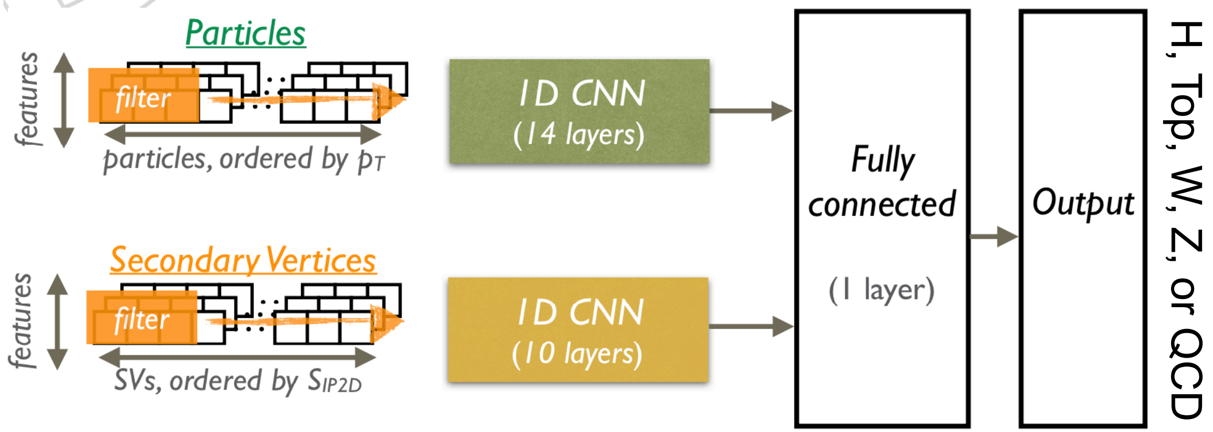
***Training on simulation data**,*
Good but smaller performance on real data

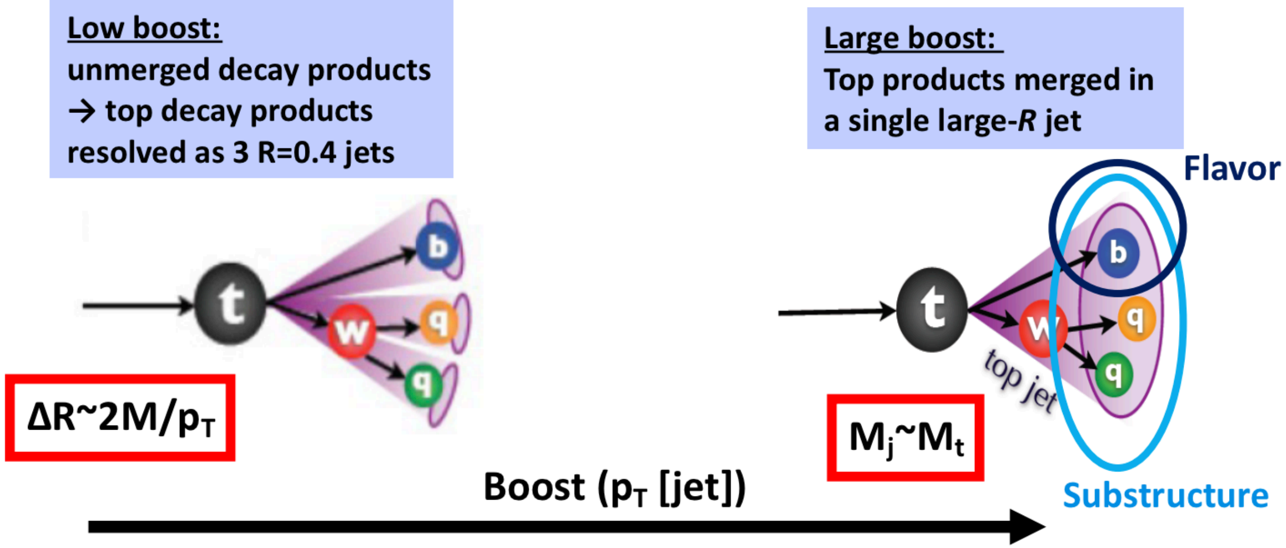
Top etc. tagging

Highly-boosted top quarks

- A top decays into a b quark and a W ⟶ lep,had
- If the momentum is large the form a common jet
  - **Boosted topology in fat jets** (AK8)
- Top-tagging ⟺ identify such jets

Typical problem how to handle a varying number of particles, here: 1d CNN

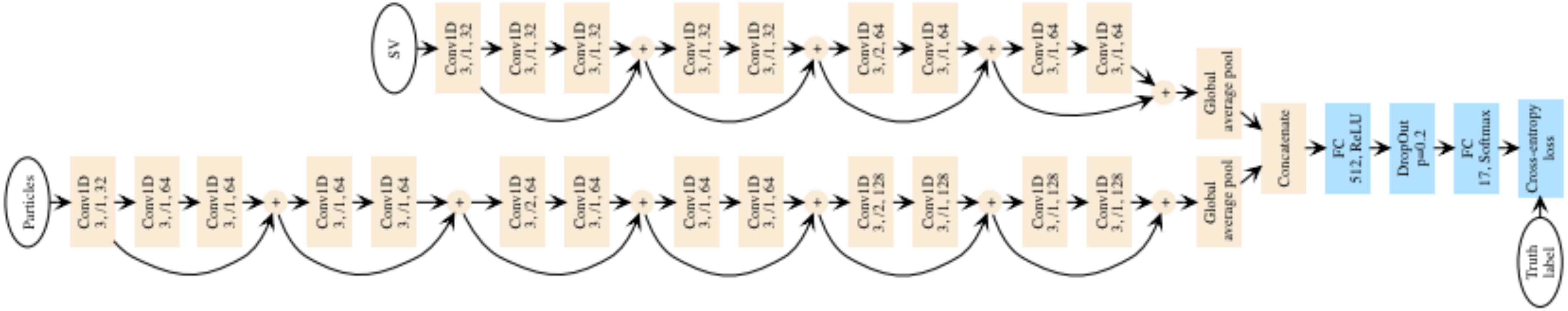




DeepAK8

- Up to 100 particles
- Each particle comes with 42 features
- 14 layers is indeed deep
- Up to 7 SV with 7 features
- Trained with 50 million jets
- **Residual net** (short cuts between layers)

CMS-PAS-JME-18-002



Convolution allows to handle a varying number of particle within a jet

Our networks are typically in the order of 10^5-10^6 parameters

How to feed in a changing number of objects

A common subject

Many Deep Learning approaches have been developed for imaging, i.e. dense regular grid of points; or text processing, i.e. sequences

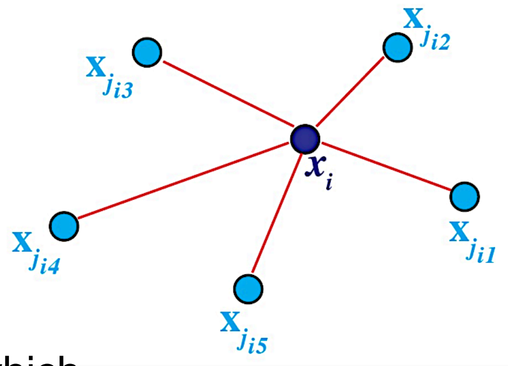
- Not straightforward how to connect to typical HEP problems
  - Jet images in (η, ϕ) plane are sparse
  - Varying number of particles in cone from jet to jet
- Standard solution so far
  - Recurrent nets: p_T ordered particles
  - Convolution: particles as 1d string of inputs
- New approach: Geometric learning or
  - **Graph networks**
    (Hot subject on this year NIPS dozens of papers)

- H. Qu and L. Gouskos.
  „ParticleNet: Jet Tagging via Particle Clouds“
  https://arxiv.org/abs/1902.08570
  - Bsed on Y. Wang et al.
    „Dynamic Graph CNN for Learning on Point Clouds“
- M. Fey and J.E. Lenssen.
  „Fast Graph Representation Learning with PyTorch Geometric“
  https://github.com/rusty1s/pytorch_geometric

Our data are not images – no regular grid

EdgeConv network

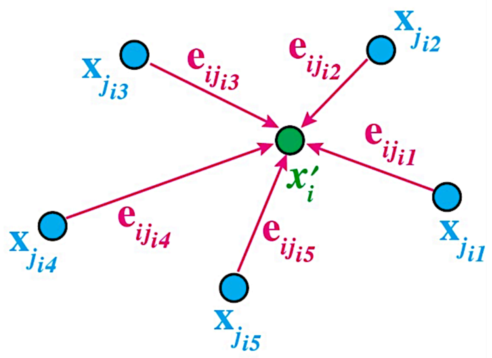
Particle Cloud



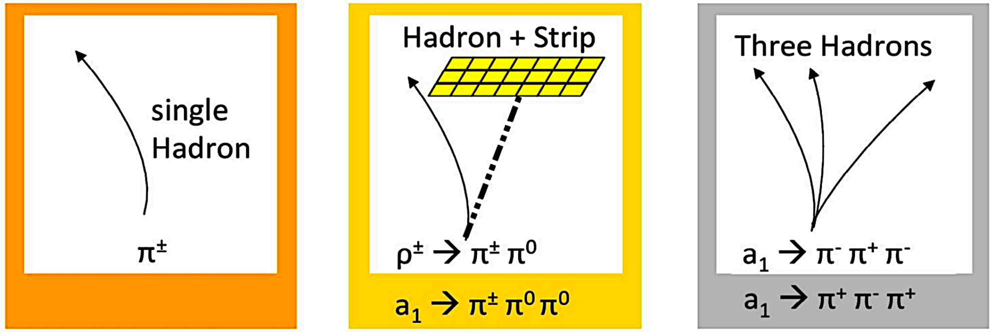
- Compute k-nearest neighbors graph
  - Defines a local neighborhood on which a convolution operation work
  - For the first layer this is the $(\Delta\eta, \Delta\phi, p_T)$ plane wrt. jet axis
- Apply EdgeConv, here

$$e'_{ijm} = \mathrm{ReLU}(\boldsymbol{\theta}_m \cdot (\mathbf{x}_j - \mathbf{x}_i) + \boldsymbol{\phi}_m \cdot \mathbf{x}_i)$$

$$x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm}$$



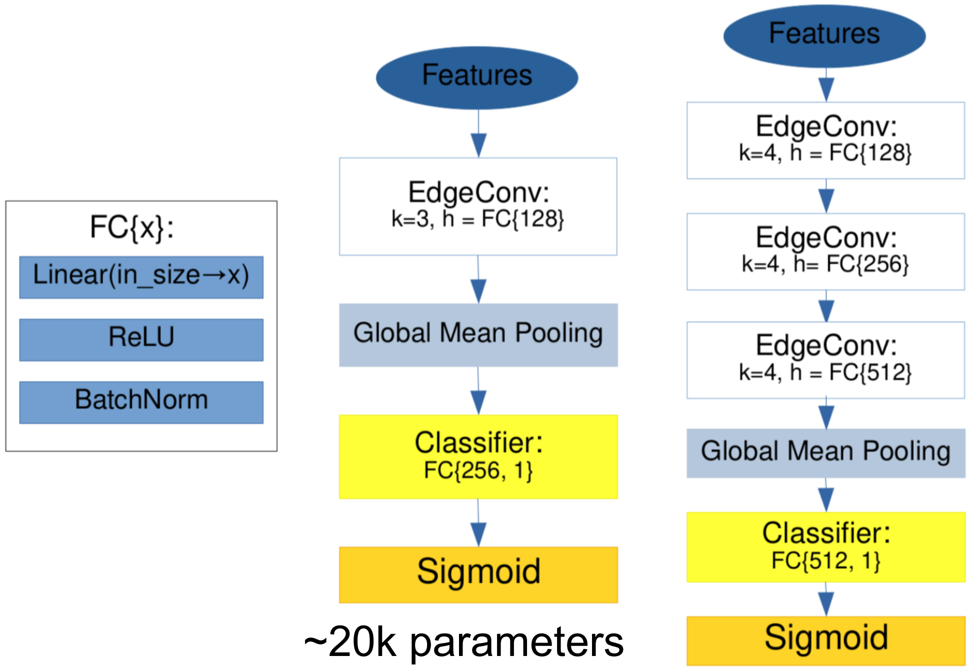
- Dynamic Graph CNN means the recalculation of the k-nn in the x' space



Tau-lepton tagging

A newly started project with Graph Networks – no official CMS results yet

- CMS investigates deep NN for tau-lepton identification
- Similar complexity as the shown b tagger
- Here at DESY, we started to investigate a Graph Network approach
- Promising results similar performance with a model that contains **10 times less parameters**
  - **Time advantage in interference**



~20k parameters

~50k parameters

Solves 3 problems:
- varying number of features
- representing non-image-like data
- fast interference

Physics analyses

Searches

Examples for ML methods in searches

Using ML methods (BDTs, NNs) has become the default for new analyses

Two example analyses from DESY

- Supersymmetry
- Higgs

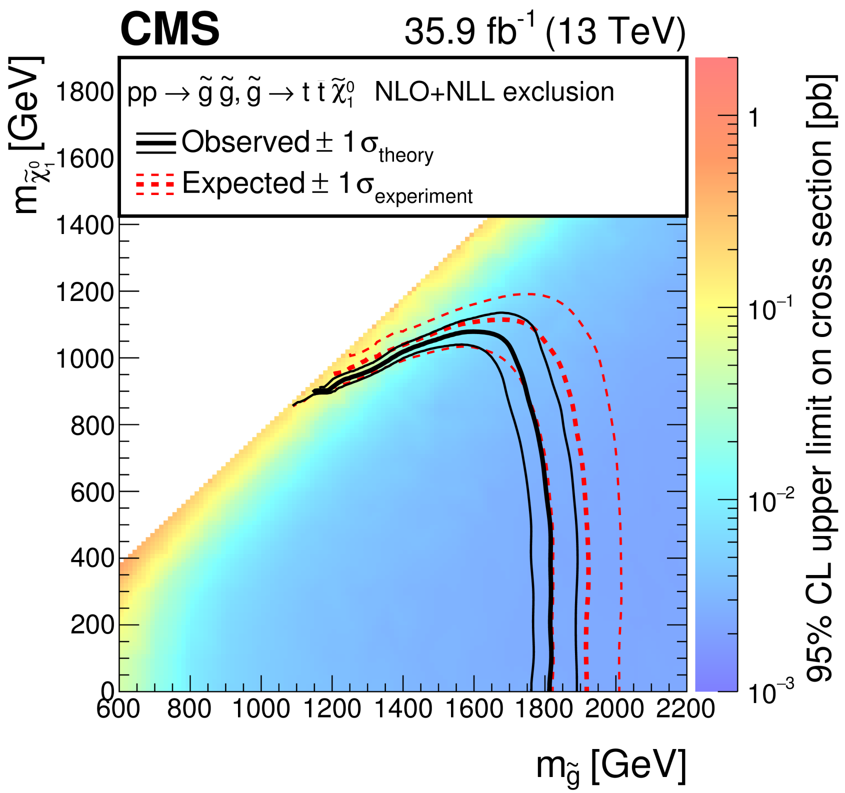
Please note!

- These are ongoing analyses
- Results are not approved
- Therefore no numbers, no data plots

- *Access to data (and simulations) is restricted and controlled by the collaboration*
- *CMS has recently relaxed the rules under certain conditions (mostly driven by possible ML publications)*

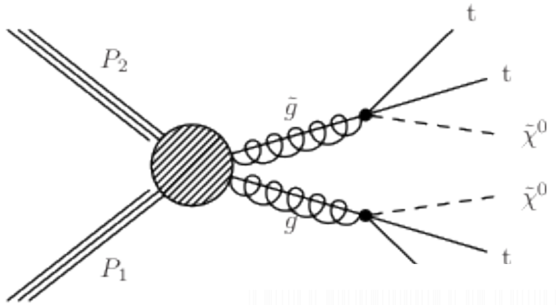
Ex.1: Search for supersymmetry in events with one lepton

Search for supersymmetry in events with one lepton and multiple jets exploiting the angular correlation between the lepton and the missing transverse momentum in proton-proton collisions at √s = 13 TeV

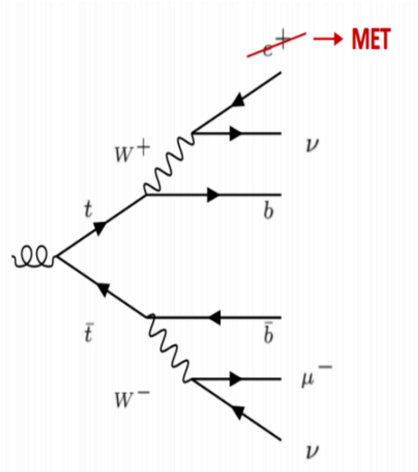


published Run II result on 2016 data

- **SUSY** signal: T1tttt
  2 **gluinos** that decay
  to 4 top quarks
  + 2 **neutralinos**
  (**Dark matter** candidate)



- Important **SM** background
  top-anti-top pair (**ttbar**) with
  2 leptons where
  one lepton is missed in the reconstruction



Can we do better with a neural network? – Sure,but …

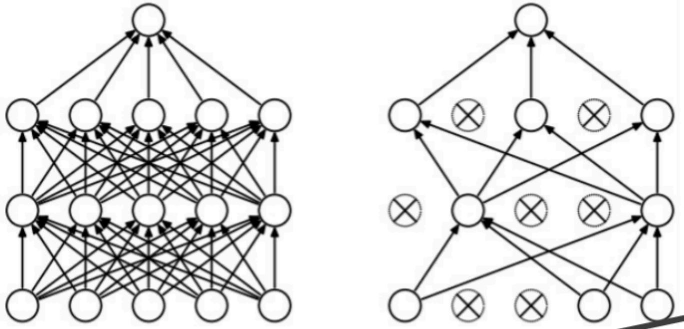
- How to connect the NN response to the observed data to estimate the background in the signal region?

DNN DETAILS

100 epochs, batch size = 2048
Dropout with rate of 0.01

Early stopping to stop training once the model performance stops improving on a hold out validation dataset with patience of 10 epochs.

Preselect events :
- I Lep , no Veto, LT > 250 , HT > 500, nJ > 3 nbJ >= 1, iso_track veto



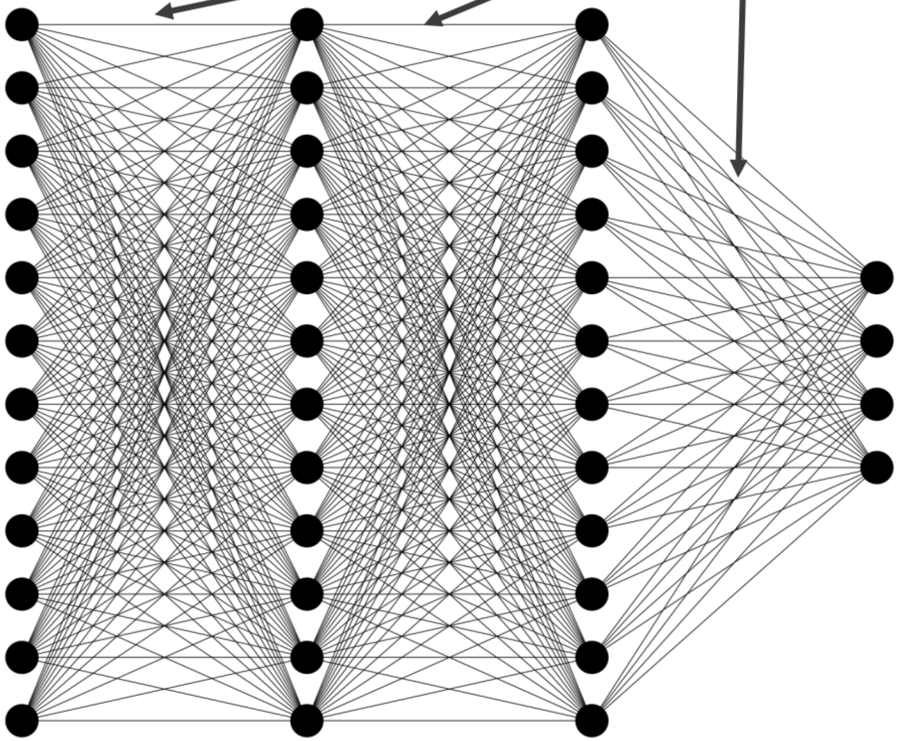
regularisation

Dropout layer

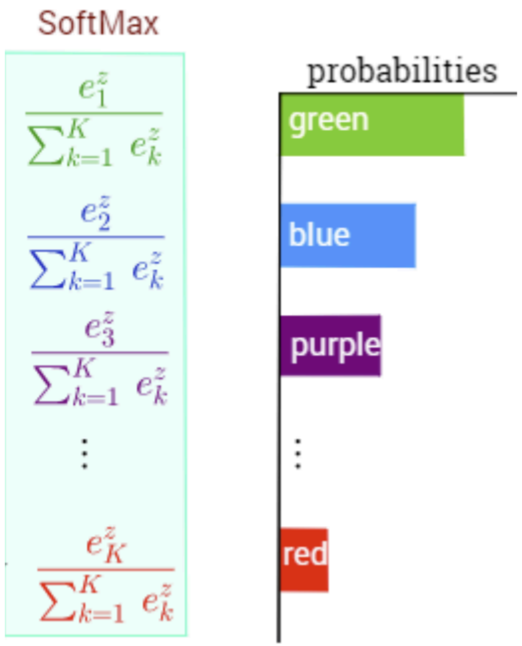
Input Features

$$MET, MT, Jet_{p_T}^{1/2}$$
$$Lep_{p_T}, L_T, H_T, n_{bJets}$$
$$n_{Top}, n_{jets}, \Delta\phi, Lep_{relIso}$$
$$Lep_{MiniIso}, IsoTrack_{p_T}$$
$$IsoTrack_{MT_2}$$
$$m_{\tilde{g}}, m_{\chi_1^0}$$

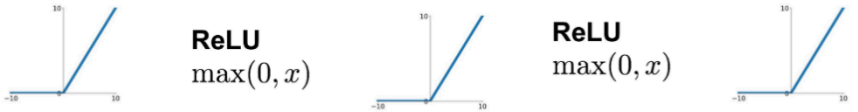
17 input variable



Class 0 → DiLep Ttbar
Class 1 → SemiLep Ttbar
Class 2 → Wjets + others
Class 4 → Signal



ReLU $\max(0, x)$ **ReLU** $\max(0, x)$ **ReLU** $\max(0, x)$

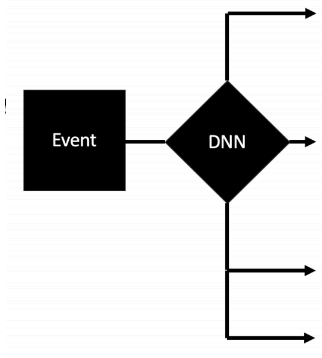


Multiclassification

Data augmented background estimation

- 4 classes
  - 3 class represent different backgrounds
  - 1 class is the search region
  - The simulation is normalized to the data in the background classes by solving the equation system

background	α	β	γ
1500_1000	0.84± 0.01	1.0± 0.05	0.72 ± 0.03
1500_1200	0.84 ± 0.01	0.98 ± 0.04	0.72 ± 0.03
1600_1100	0.84 ± 0.01	0.98 ± 0.05	0.73 ± 0.03
1700_1200	0.84 ± 0.01	0.98 ± 0.05	0.73 ± 0.03
1800_1300	0.84 ± 0.01	0.98 ± 0.04	0.71 ± 0.03
1900_100	0.84 ± 0.01	0.97 ± 0.05	0.71 ± 0.03
1900_1000	0.84 ± 0.01	0.96 ± 0.04	0.71 ± 0.03
1900_800	0.84 ± 0.01	0.94 ± 0.05	0.72 ± 0.03
2200_100	0.84 ± 0.01	0.96 ± 0.05	0.73 ± 0.03
2200_800	0.84 ± 0.01	0.96 ± 0.04	0.71 ± 0.03



$t\bar{t} \rightarrow l$

$\alpha \times (t\bar{t} \rightarrow l)_1 + \beta \times (t\bar{t} \rightarrow ll)_1 + \gamma \times (WJ + others)_1 = (data)_1$

$t\bar{t} \rightarrow ll$

$\alpha \times (t\bar{t} \rightarrow l)_2 + \beta \times (t\bar{t} \rightarrow ll)_2 + \gamma \times (WJ + others)_2 = (data)_2$

$WJ + others$

$\alpha \times (t\bar{t} \rightarrow l)_3 + \beta \times (t\bar{t} \rightarrow ll)_3 + \gamma \times (WJ + others)_3 = (data)_3$

$T1t^4$

use α, β and γ

$$Y_i = A_{ij} X_i$$

$Y_i \rightarrow$ data counts

$A_{ij} \rightarrow$ MC counts

$X_i \rightarrow$ scale factor for each background

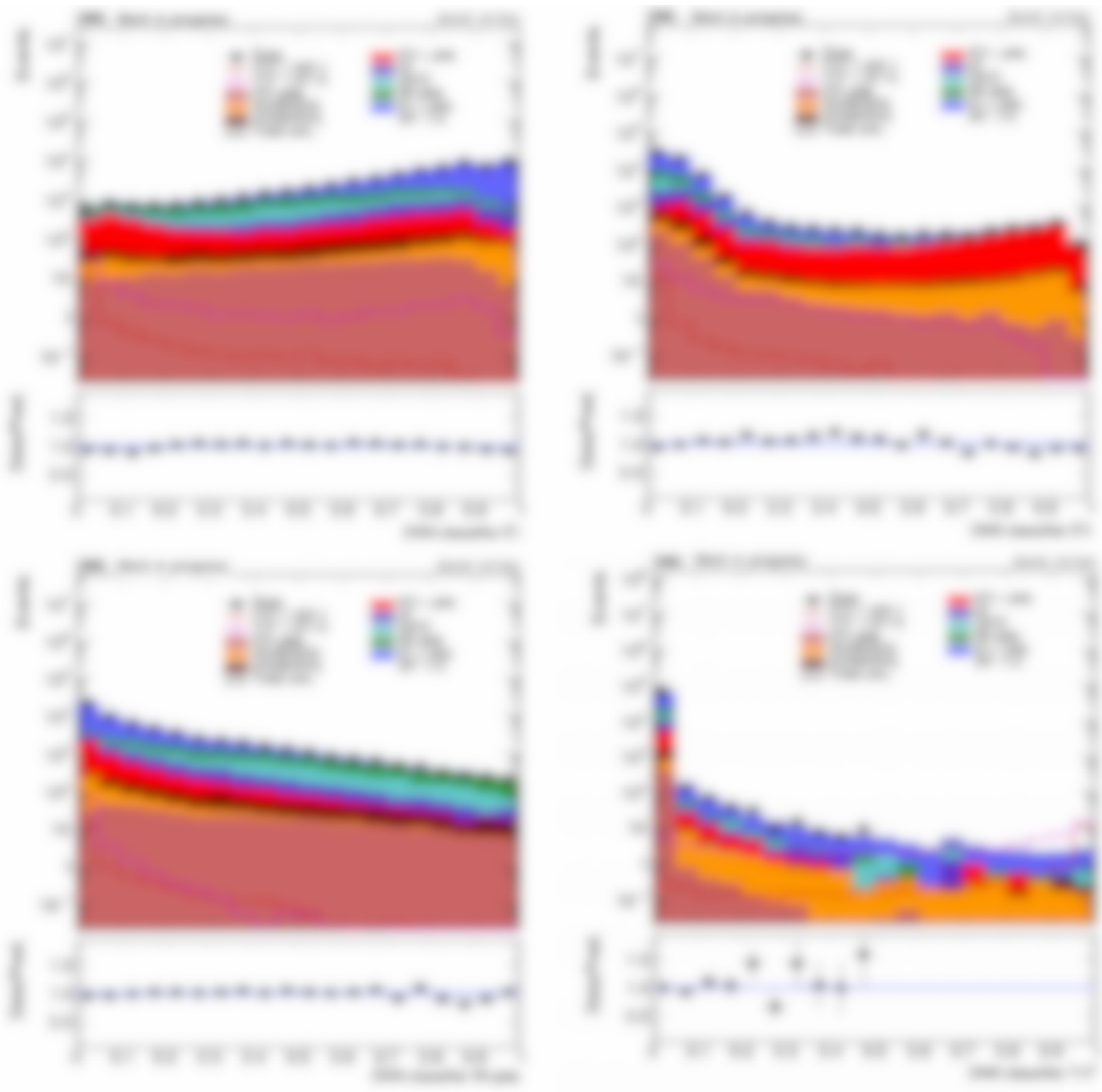
$i \rightarrow$ number of CRs

$j \rightarrow$ background number

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \times \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} data_1 \\ data_2 \\ data_3 \end{bmatrix}$$

- **Creates a nice data/MC agreement -> estimate Background in signal region**
  (not shown here)
- Independent of the signal point

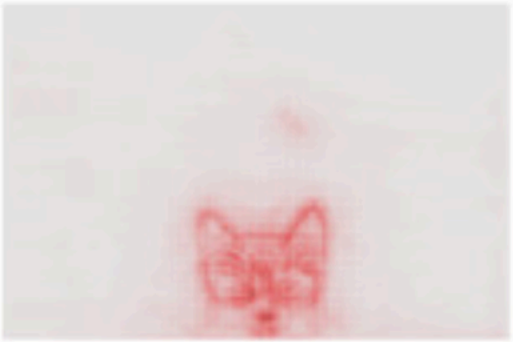
*Typically, physics analyses itself uses smaller networks (or BDTs). Main problem here how to calibrate the data performance aka **data driven background estimation**.*



Ex2. How to know what is relevant?

Opening the neural net black box in a Higgs search

- Complex neural nets do not tell us how they come to a decision
  - 10th thousands of parameters
  - Several dozens of input variables
- Important to understand what is driving the decision
  - If the network is just a chain of matrix multiplication and function mappings why not just do a Taylor expansion
  - arXiv:1512.02479



Deep Taylor Decomposition

- **Higgs $\longrightarrow \tau\ \tau$**
  - Observation published last year (CMS) *Phys. Lett. B* **779** (2018) 283
  - New study with a multiclass NN ongoing

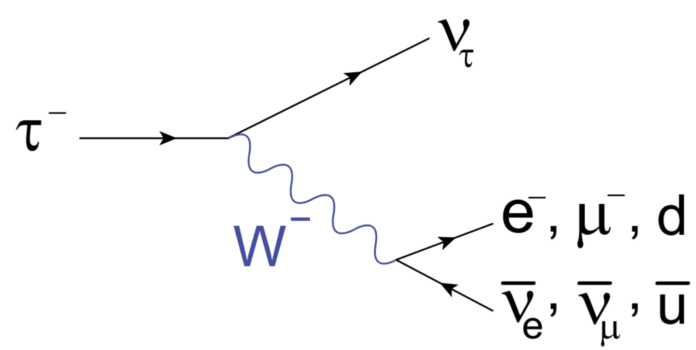
Relevance propagation of variables as applied in H $\rightarrow \tau\tau$

Teresa Lenz, Mareike Meyer, Alexei Raspereza for the HIG-18-032 team

S.Wunsch, R. Friese, R.Wolf, and G. Quast, Computing and Software for Big Science 2 (Sep, 2018) 5, doi:10.1007/s41781-018-0012-, arXiv:1803.08782

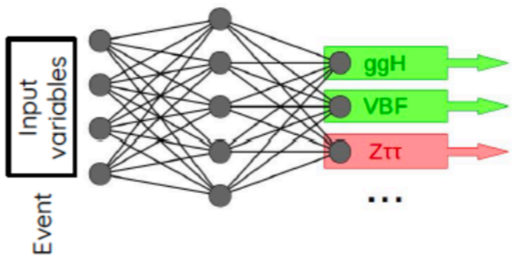
2nd order Taylor coefficients as sensitivity metric

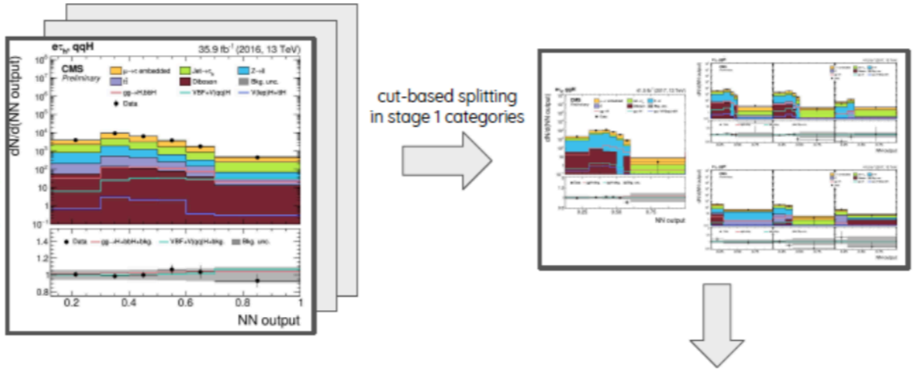
Analysis strategy



- four most sensitive final states of $\tau\tau$-pair studied: **eμ, eτ_h, $\mu\tau_h$, $\tau_h\tau_h$**
- loose baseline selection (trigger requirements, suppression of large backgrounds)
- **multi-class NN** with **2 signal classes (ggH, qqH) & several background classes** (control regions)
- selection & validation of NN input variables based on 1D and 2D GoFs → **relevance propagation of variables**

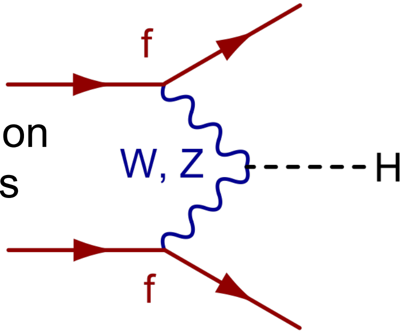
2 hidden layer with 200 nodes each

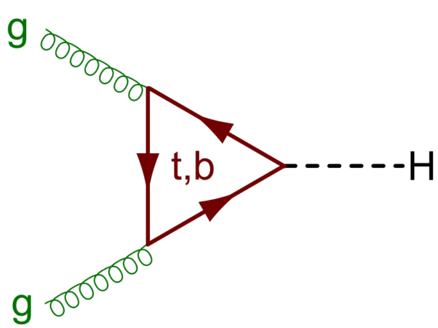




cross section

production modes





K + TensorFlow

Application to H → ττ: 1st order coefficients

- we can learn which variables were important for the NN to identify the respective process

2nd order coefficients

- 50 highest ranked variables or combinations

Understandable Deep Learning becomes important if we want to claim discoveries

Fast Simulation

Computing Load

From LHC to HL-LHC

- The LHC will start its 3rd data taking period in 2020
- Already now it is not obvious that the computing resources scale with the double amount of data
- The High Luminosity LHC will create a 10 times larger dataset
- **Large datasets implies even larger simulation datasets**: about 85% CPU resources had been used here
- Our fast (parametrized) simulation are often **not sufficient for ML based analyses**

CMS	Run I	Run II	Run III	HL-LHC
	2010-2012	2015-2018	2020-2023?	2026-203x
Int. Lumi	25/fb	150/fb	**300/fb**	**3000/fb**
Raw Data	3 PB	**36 PB**	?	??
PU		35	55	200

New compact (50x smaller) data formats are available for analyses

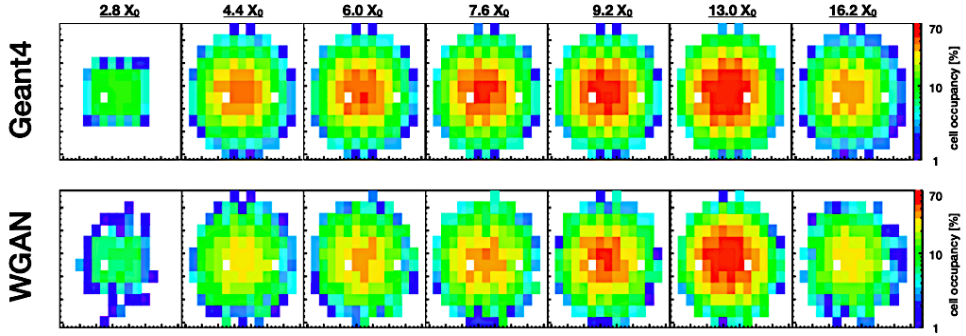
GANs for fast simulation

And other generative methods

W-GAN

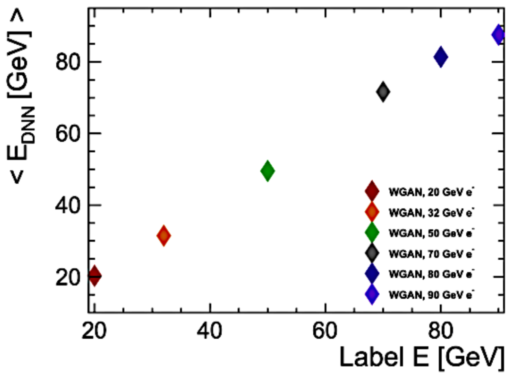
- Several CMS related groups studying different approaches
- To my knowledge: No official CMS approach
- Main interest in HGCAL for the HL-LHC upgrades
  - But also full event simulation
- GANs based on GraphNeural Networks for HGCAL (hexagonal structures)
  - Reconstruction 1902.07987

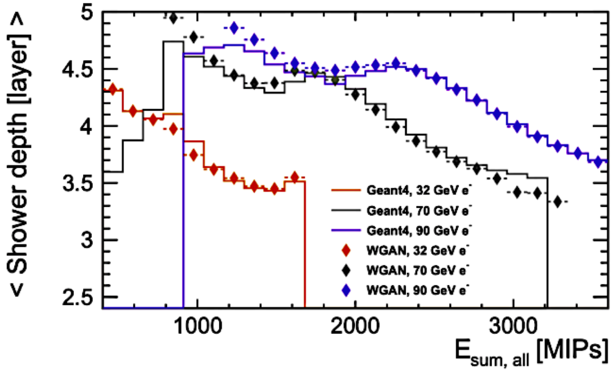
Use Wasserstein GAN to simulate electromagnetic calorimeter



$$L = D(x) - D(G(z)) -$$
$$-\lambda(||\nabla D||_2 - 1)^2 - 1) -$$
$$-\sum_i \kappa_i |(y_i - a_i(x))^2 - (\alpha_i - a_i(G(z)))^2|$$

D: Discriminator
G: Generator
x: True inputs
z: Random inputs
y: True labels
α: Random labels
a: Constrainer outputs
i: Position and Energy





Martin Erdmann, Jonas Glombitza, Thorben Quast: *Precise simulation of electromagnetic calorimeter showers using a Wasserstein Generative Adversarial Network* (1807.01954)

Method	Hardware	20 GeV e^-	Speed-up	90 GeV e^-	Speed-up
GEANT4	any CPU	O (500 ms)	-	O (2000 ms)	-
WGAN	Intel© Xeon© CPU E5-1620	52 ms	x10	52 ms	x40
WGAN	NVIDIA© Quadro© K2000 GPU	3.6 ms	x140	3.6 ms	x560
WGAN	NVIDIA© GTX™ 1080 GPU	0.3 ms	x1660	0.3 ms	x6660

CHEP 2019

Computing in High Energy, 4-8 Nov. Adelaide, Australia

- CMS ☹️
- ATLAS: GANs and VAEs (for electromagnetic calo) (here)
- CERN openlab: 3DGAN (here)