Imperial College London

Data Science Institute

DeepJet: Jet classification with the CMS experiment

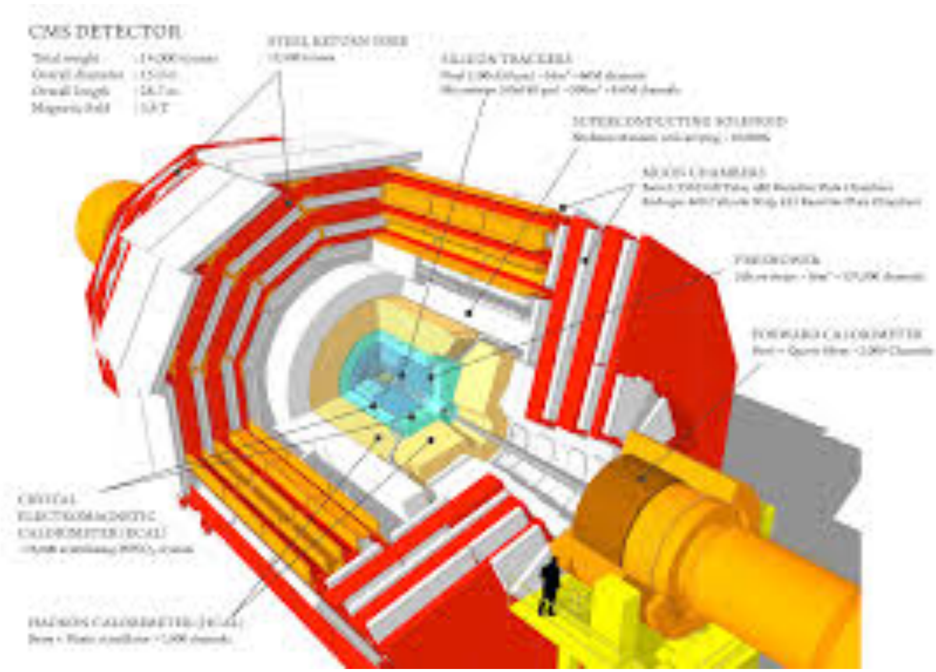
Markus Stoye
Imperial College London, DSI

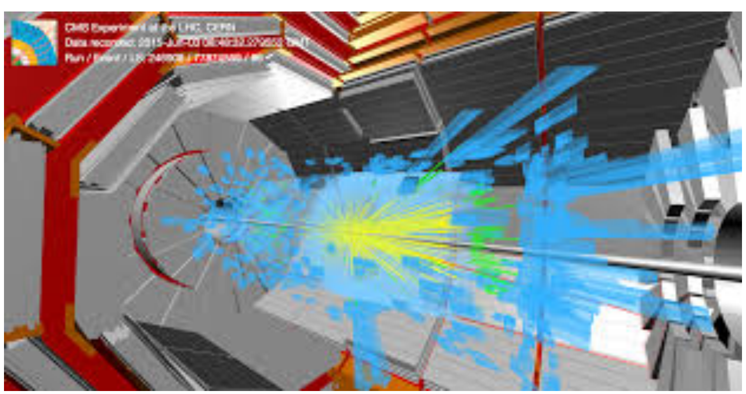
“Big data science in astroparticle physics”, HAP workshop, Aachen, Germany, 20th Feb. 2018

Content

- Introduction
- Revisited machine learning for flavor tagging
- Deep learning for jet tagging

Problems in CMS experiment invite for "predictive" machine learning





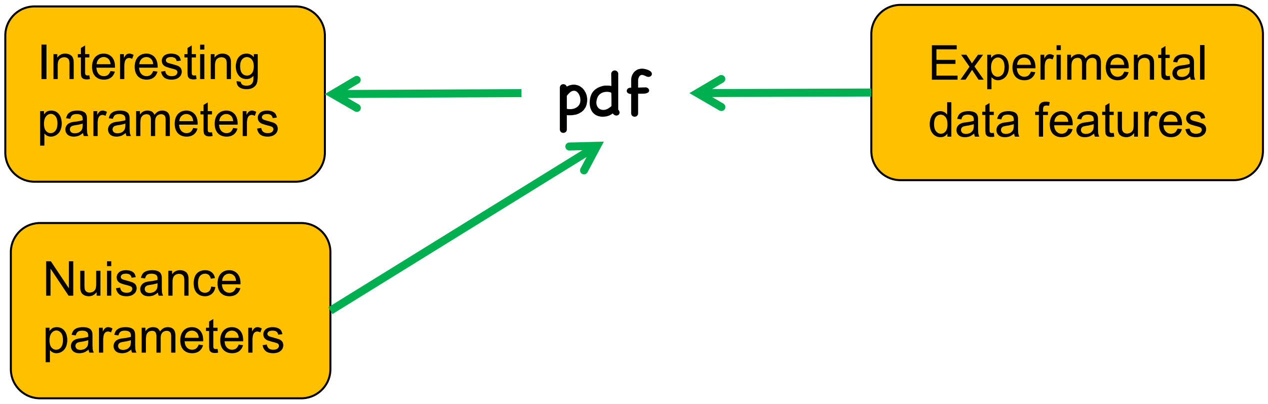
CMS experiment:

- Complex heterogeneous detector, 100M channels and100.000s nuisance parameters
- ***Very Good*** generator model (our simulation) already existing
- Billions of examples

Astrophysics?

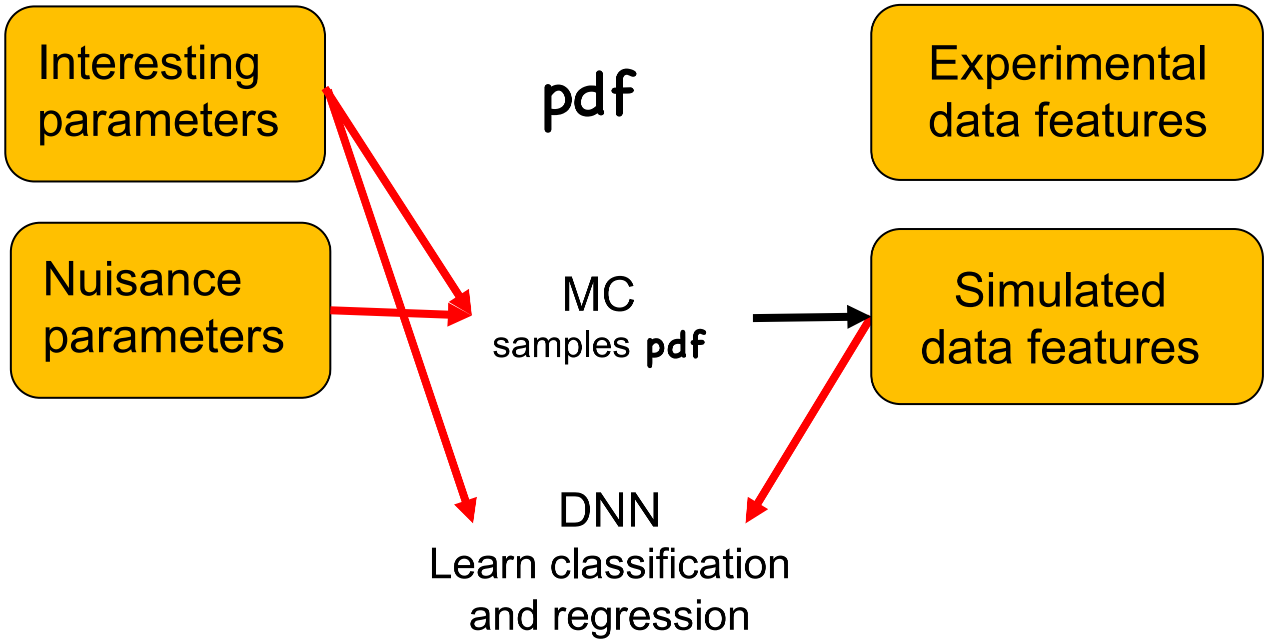
- Likely generative machine learning more important than in CMS

Infer interesting parameters from data in CMS



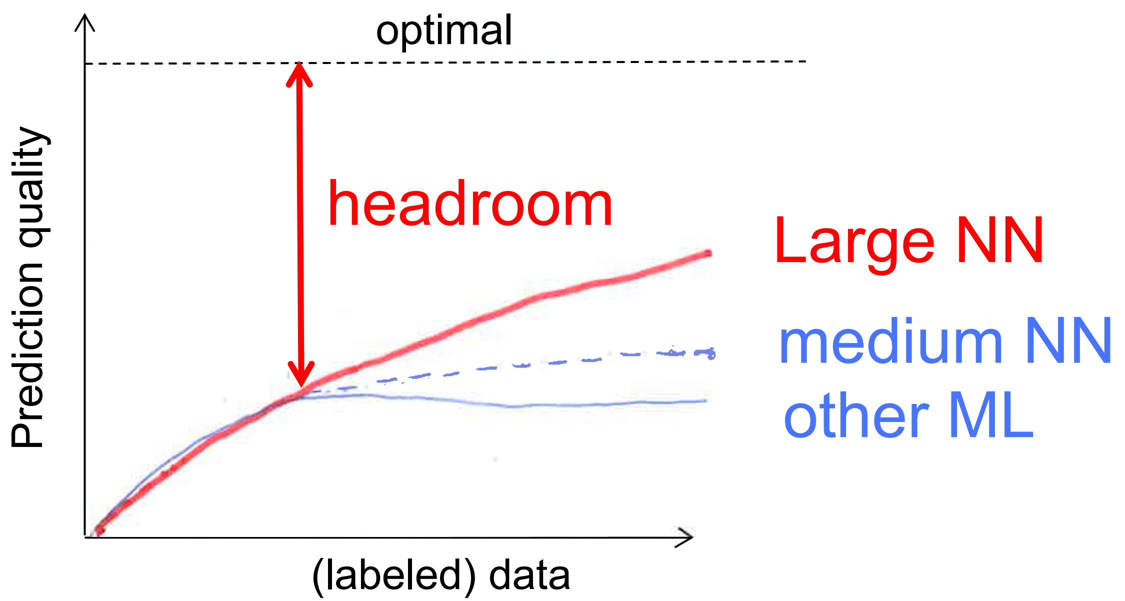
- *Ideally* we would have the **pdf** for likelihoods
- We can not write the **pdf** down analytically for our complex experiment (CMS)

Supervised deep learning to estimate parameters



- Practically we can make MC simulation
- We that we can try a ML to estimate interesting parameters

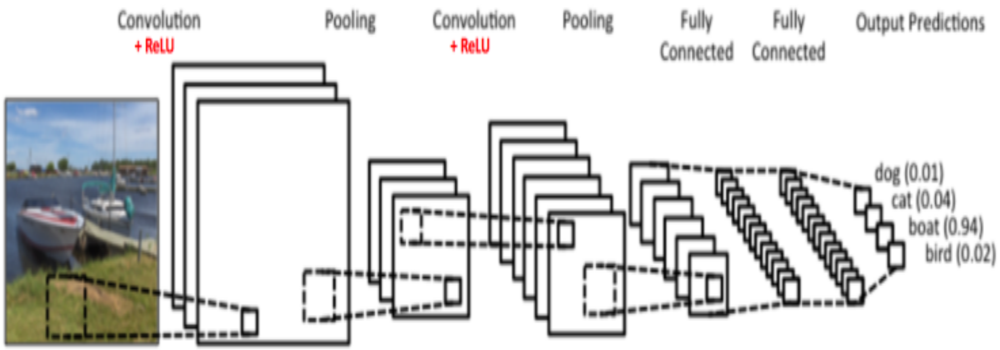
Deep learning: bigger data is better data



- High dimensional inputs with big dataset and a large Deep Neural Networks brought breakthroughs
- We have huge numbers of simulated samples with truth information 😀
- It is very hard to estimate the ***headroom*** left 😕

Neural network glossary

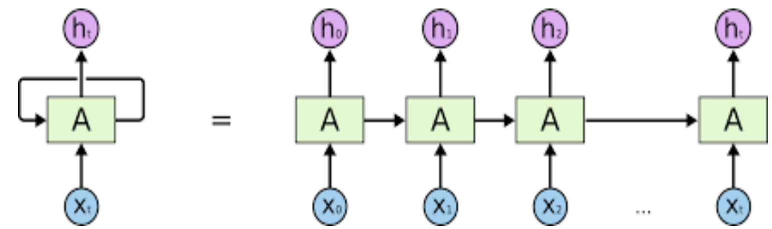
Convolutional neural network



Initially made for images:

- Discrete pixels (2D)
- Translation invariant (constant resolution)
- Local features need to be important
- . . .

Recurrent neural network

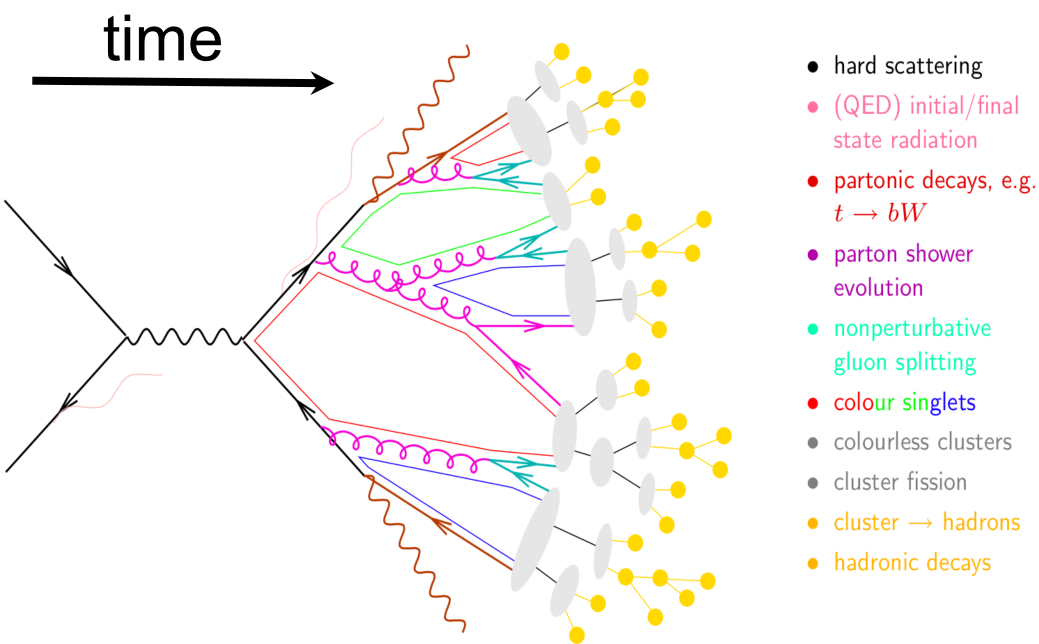


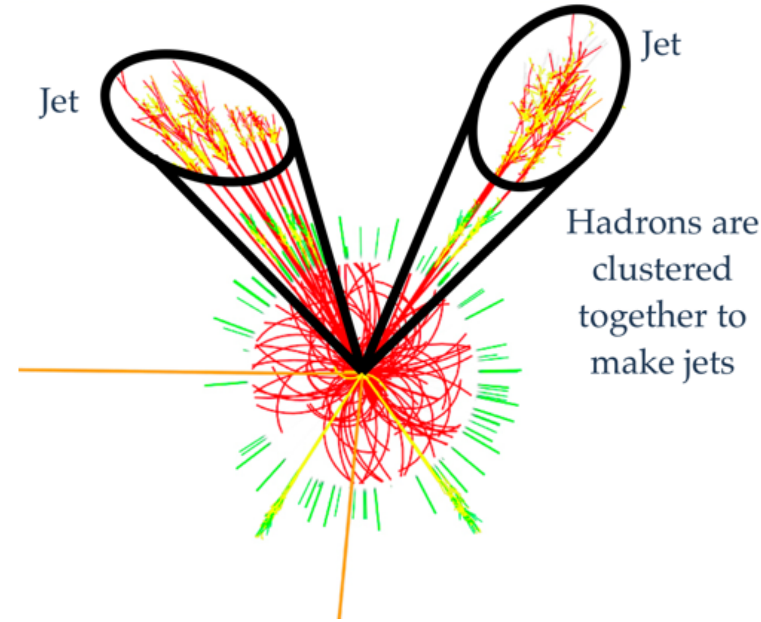
Often used in natural languages or time series:

- Flexible length sequence as input, output always the size
- Long-short term memory RNN (LSTM) avoids e.g. zero impact of early elements in sequence
- . . .

Revisited machine learning for heavy flavor tagging in CMS

Collecting particles from one hard scatter particle

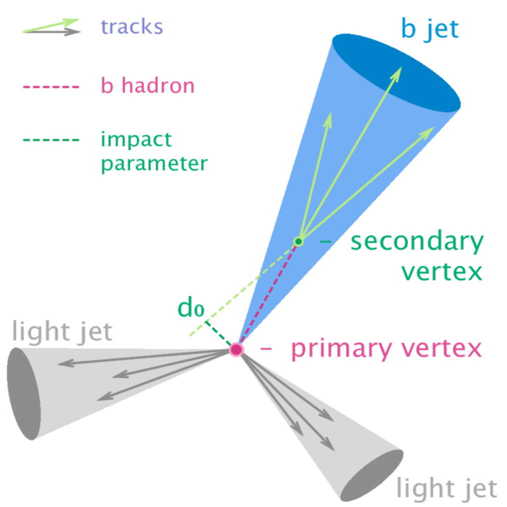




- We are interested in the properties particle Id) of the “black”; but in the detector we see the loose ends on the right.
- We use a clustering algorithm (anti-k_T) to collect particle candidates and than secondary vertices that might belong to one particle from the hard scatter.

Jet tagging

Task to find the particle ID of a jet, e.g. b-quark



Key features:

- Long lifetime of heavy flavor quarks
- Displaced tracks, …
- Usage of ML standard for this problem

Revisited machine learning part from scratch

Changed to multi-class classification

Jet flavor tagging is intrinsically a multi-class classification problem

4 exclusive ***flavor*** categories:

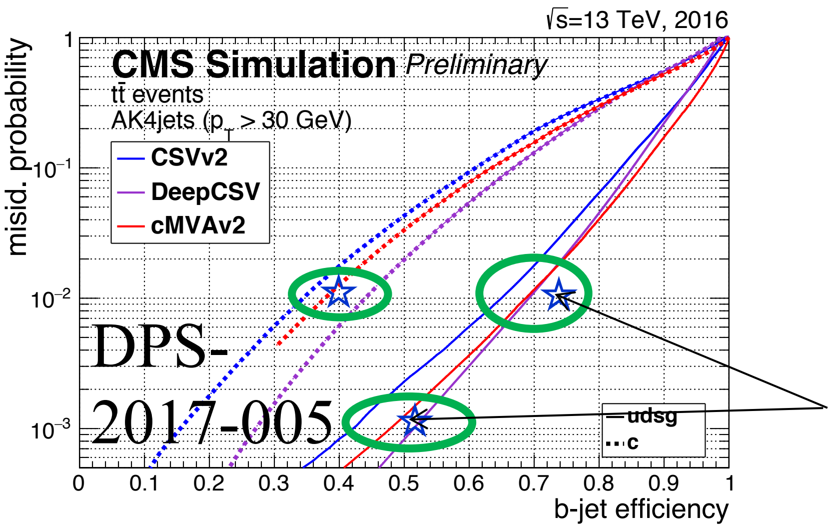
- Exactly **one b hadron** in the jet
- Exactly **one c hadron**, with no b-hadron in the jet
- **Two** or more **b hadrons** in jet
- Light quark/gluon jets (udsg)

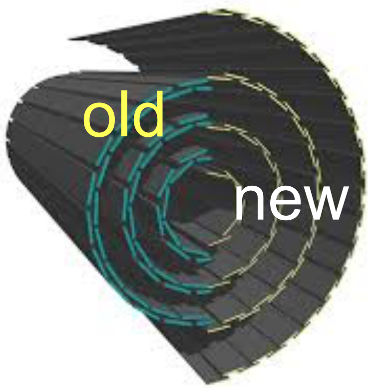
Generic jet tagging has even more classes: light quark, gluons, hadronic τ, pile up

$\rightarrow$ Using many classes is important for a robust taggers. In real data the tagger will see all possible classes

Changes of training strategy

- More diverse samples
  - QCD **and tt**
- Bigger samples
  - 50M jets!
- Use complete **standard CSV b-tag "Tag info"** (from ~30→60)
- Dense Deep Neural Network (Dense)

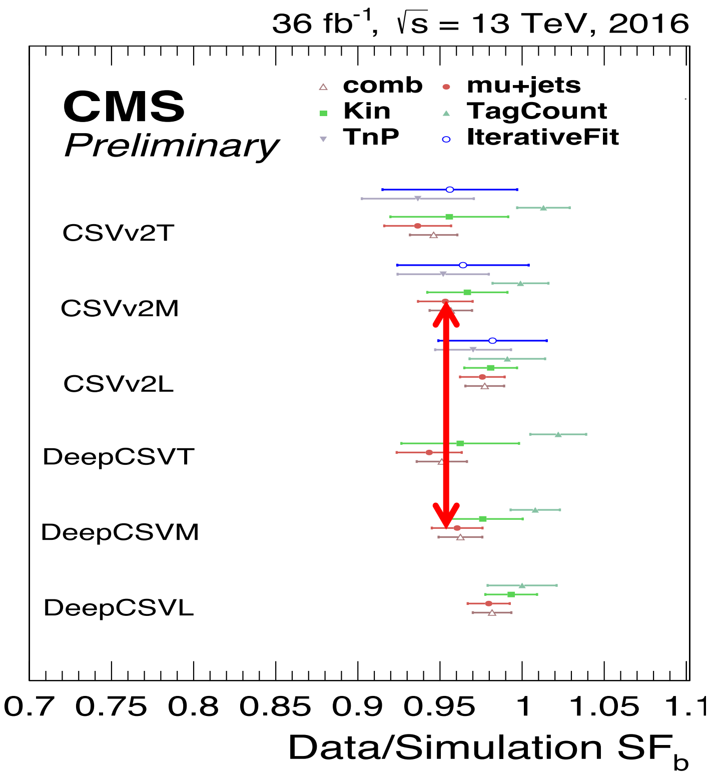




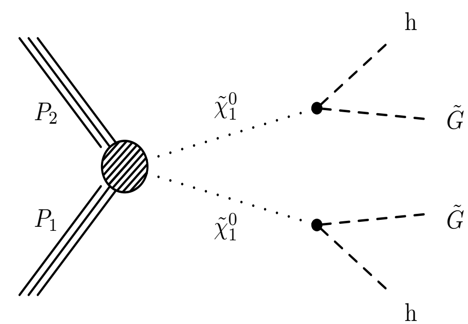
☆ Old tagger with new pixel detector in simulation

Similar impact as the new inner pixel!

Application of new tagger in data



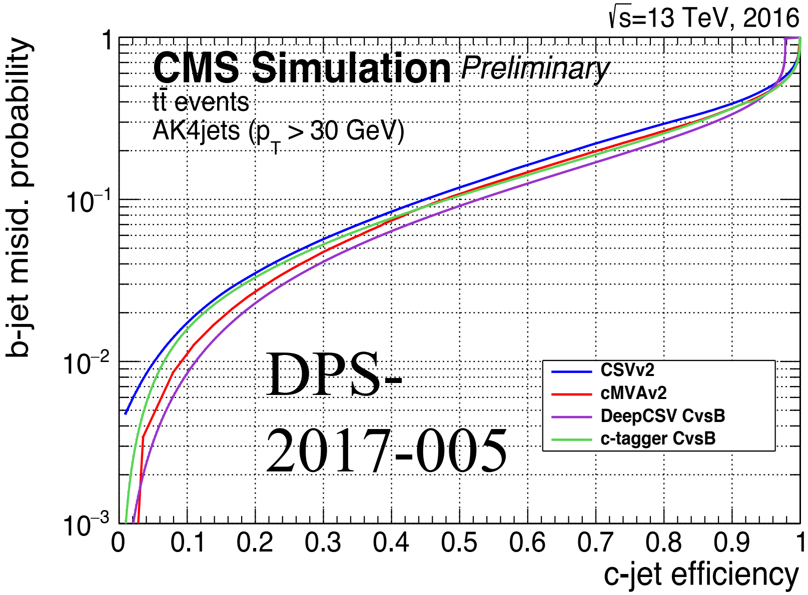
hh->bbbb and MET

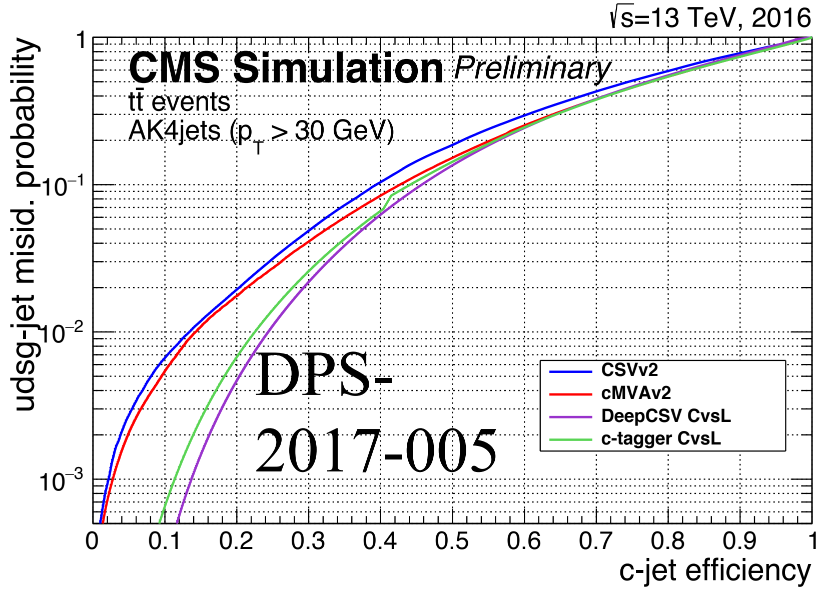


- Up to 50% more signal with 15% more bkg.
- Gained 150 GeV in m($\tilde{\chi}_1^0$)

This new flavor tagger officially ***recommended*** since 2017 in CMS!

ROC for c vs b an light

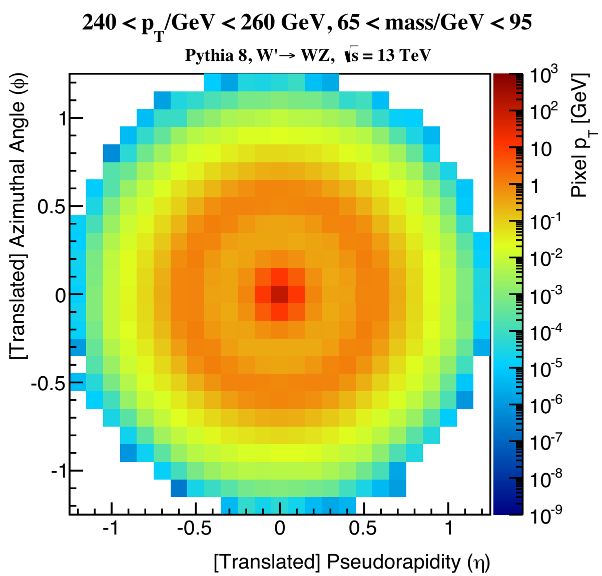


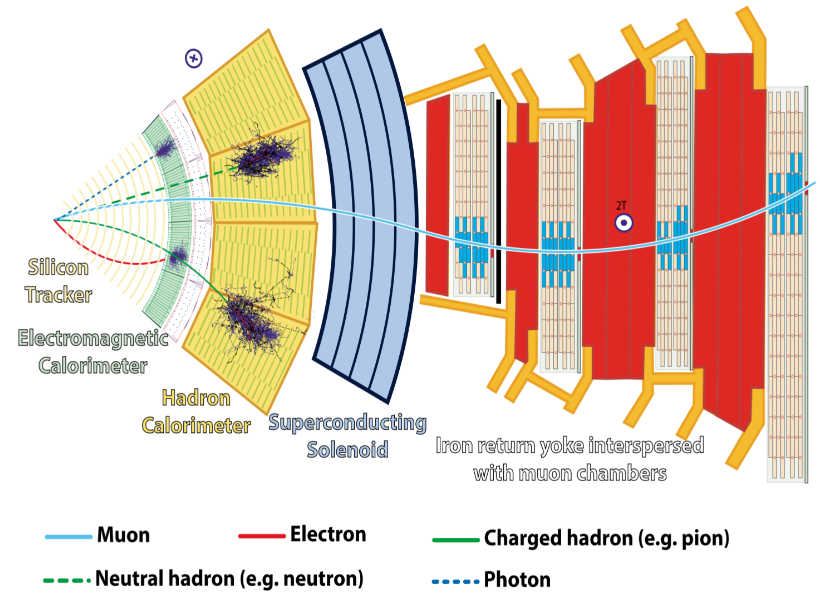


DeepCSV best c tag performance

DeepJet: jet tagging by physics object based deep learning

CMS is a complex detector





- Convolutional networks propose for jet images and shown to work for some problems
- In general the CMS detector is more complex, e.g. not translational invariant

CMS not “image” like, 2D CNN less easy to use

What is a charged particle in the detector?

Charged particles flow candidates

- Particle flow candidates combine the information of all subdetector
- p_T, η, ϕ, and and particle ID
- Estimated of probability to be from the primary vertex
- Provides links to rawer objects like tracks
- Via particle tracks access to “BTV” features and others
- ***Maybe a DeepParticle candidate would be interesting***

feature	offset	lower bound	upper bound	comment
trackEtaRel	-	-5	15	BTV
trackPtRel	-	-	4	BTV
trackPPar	-	-10^5	10^5	BTV
trackDeltaR	-	-5	5	BTV
trackPParRatio	-10	100	-	BTV
trackSip2dVal	-	-	70	BTV
trackSip2dSig	-	-	$4 \cdot 10^4$	BTV
trackSip3dVal	-	-	10^5	BTV
trackSip3dSig	-	-	$4 \cdot 10^4$	BTV
trackJetDistVal	-	-20	1	BTV
trackJetDistSig	-	-1	10^5	BTV
$p_T(cPF)/p_T(j)$	-1	-1	0	
$\Delta R_m(cPF, SV)$	-5	-5	0	
fromPV	-	-	-	
VTXass	-	-	-	
$w_p(cPF)$	-	-	-	
χ^2	-	-	-	
Npixel hits	-	-	-	

More features of particle jets

Neutral particles candidates

feature	offset	lower bound	upper bound
$p_T(nPF)/p_T(j)$	-1	-1	0
$\Delta R_m(nPF, SV)$	-5	-5	0
isGamma	-	-	-
hadFrac	-	-	-
$\Delta R(nPF)$	-0.6	-0.6	0
$w_p(cPF)$	-	-	-

Secondary vertices

feature	offset	lower bound	upper bound
$p_T(SV)$			
$\Delta R(SV)$	-0.5	-2	0
$m_{SV})$	-	-	-
$N_{tracks}(SV)$	-	-	-
$\chi^2(SV)$			
$\chi^2_n(SV)$	0	-1000	1000
$d_{xy}(SV)$	-	-	-
$S_{xy}(SV)$	-	-	800
$d_{3D}(SV)$	-	-	-
$S_{3D}(SV)$	-2	-2	0
$\cos\theta(SV)$	-	-	-
$E_{rel}(SV)$	-	-	-

global features

feature	comment
$p_T(j)$	
$\eta(j)$	
N_{cPF}	
N_{nPF}	
N_{SV}	
N_{PV}	
trackSumJetEtRatio	BTV
trackSumJetDeltaR	BTV
vertexCategory	BTV
trackSip2dValAboveCharm	BTV
trackSip2dSigAboveCharm	BTV
trackSip3dValAboveCharm	BTV
trackSip3dSigAboveCharm	BTV
jetNSelectedTracks	BTV
jetNTracksEtaRel	BTV

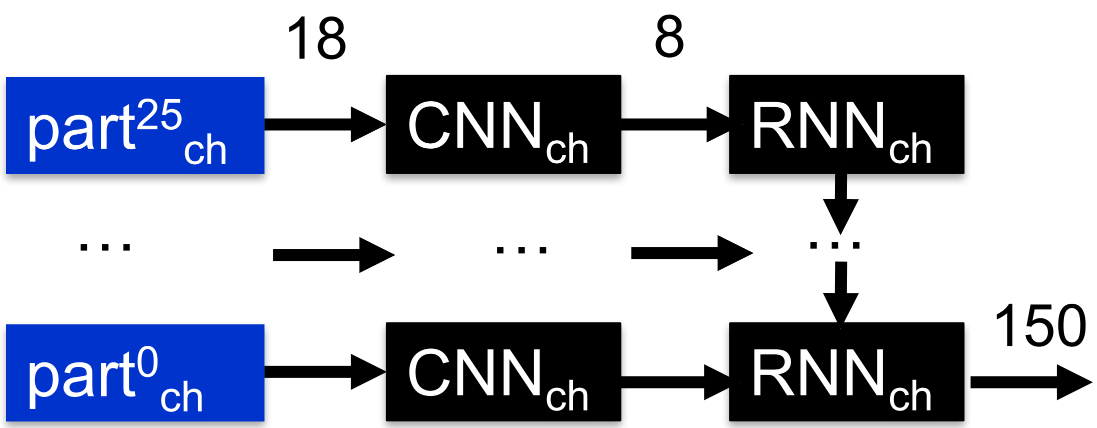
Strategy:

- Add quite extended information of jets
- Build a DNN that can deal with many and potentially low information features

Physics object based NN architecture

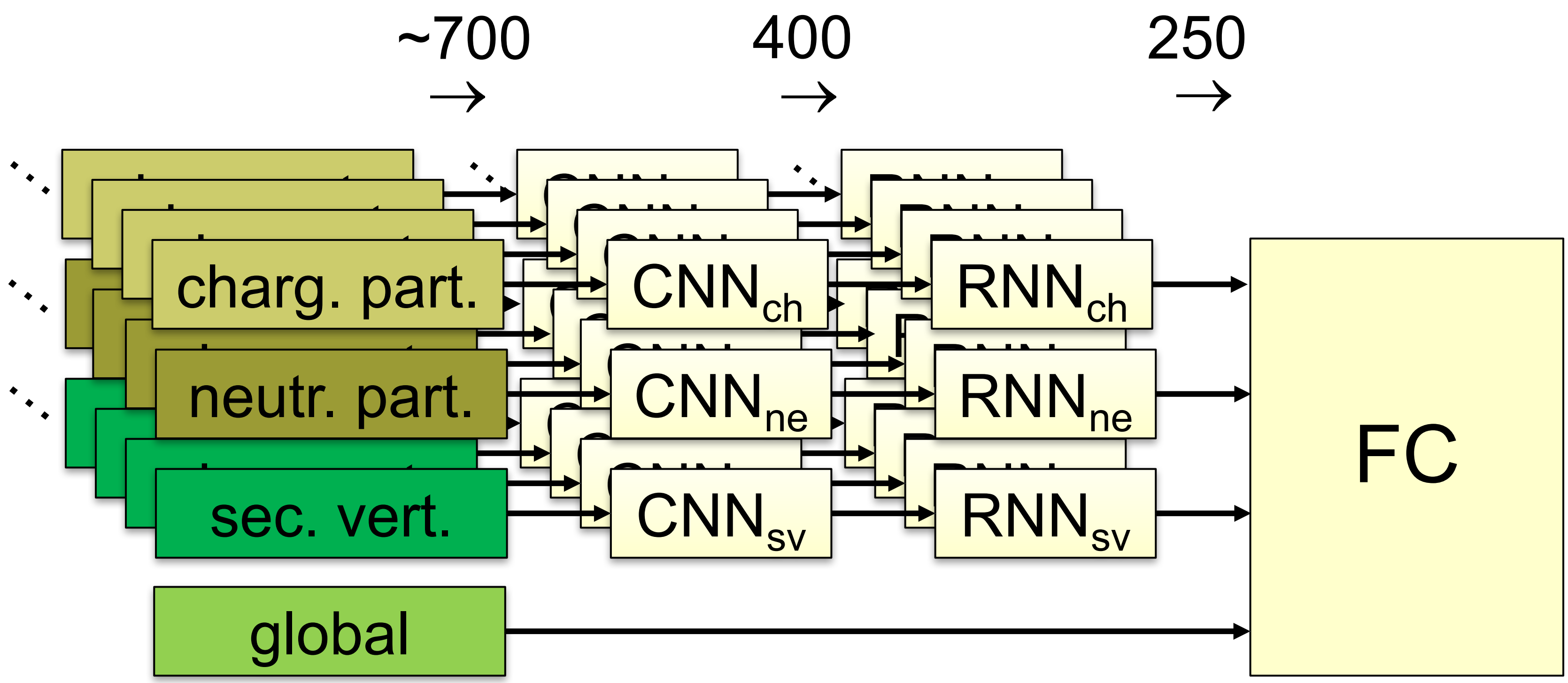
Example: charged particle candidates

- Four 1x1 1D CNN layers reduces 18 to 8 features (feature engineering)



- A recurrent NN (LSTM) represents the sequence of charged particles that is sorted by impact parameter significance
- A constant length vector is than given to the next layers

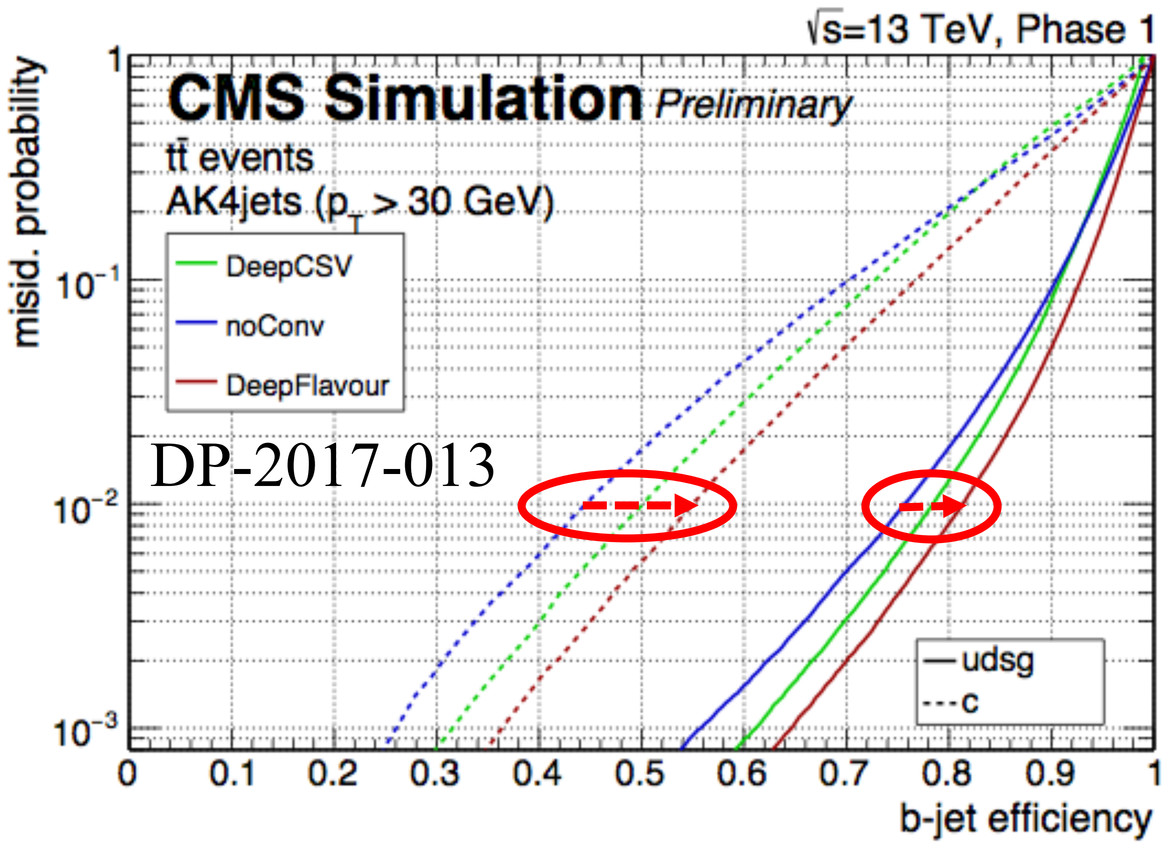
Particle and vertex based DNN: DeepJet



~ 700 inputs and 250.000 model parameters

- Particle and vertex based DNN has factor 10 less free parameters than a generic Dense DNN would have
- 100M jets used for training, overtraining is not an issue

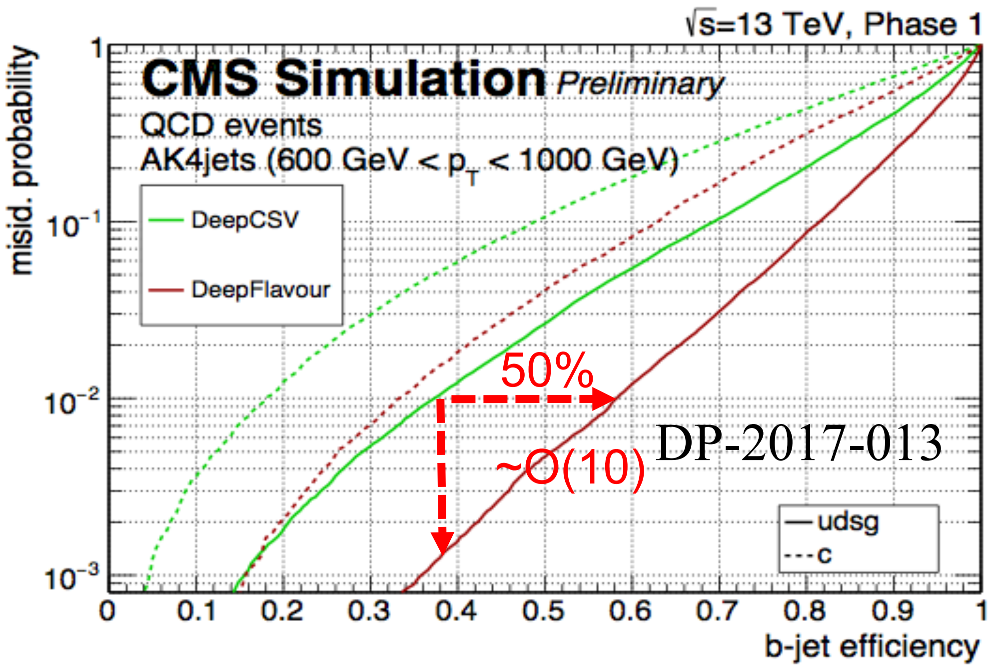
Impact of DNN architecture



Blue: generic DNN (650 inputs)
Green: CMS tagger (~65 human made inputs)
Red: Physics inspired DNN (650 inputs)

Physics object based DNN performs best

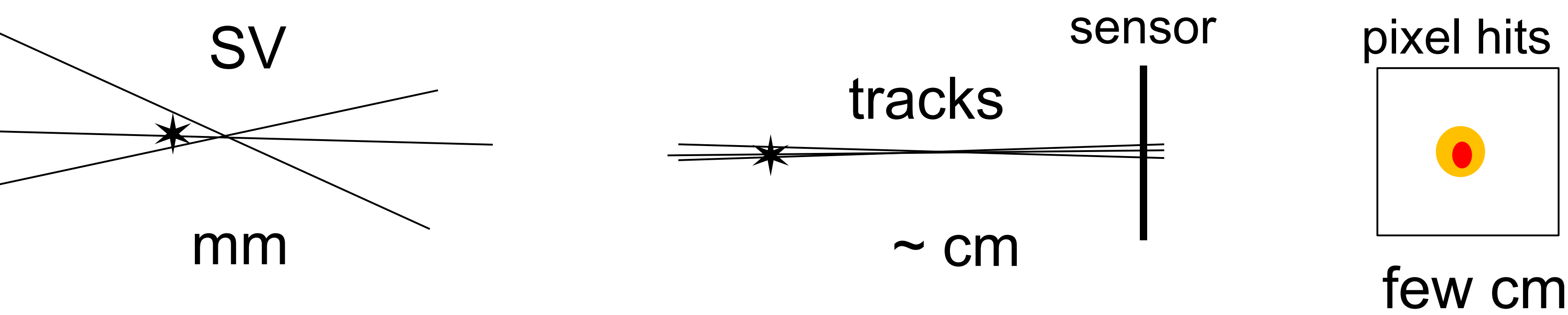
DNN reveals true CMS potential



Very significant gain at high p_T

- With DeeJet network can reproduce DeepCSV if for same inputs
- Increase input step by step:
  - ***Not applying track selection*** (lost valuable information in past)
  - ***More features help***, e.g. number of Pixel hits
- Past human features track selection procedure a bottleneck of performance
- DNN allows more automated evaluation of which information is needed

Simplified p_T evolution of b-tagging



- Vertexing and tracking increasingly difficult at high p_T
- Tracks and e.g. number of pixel hits or even pixel images become more interesting
- Track selection at high p_T was suboptimal in CMS

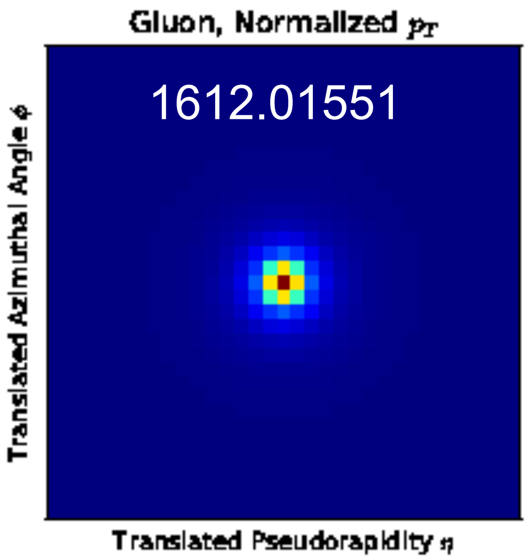
Quark gluon separation

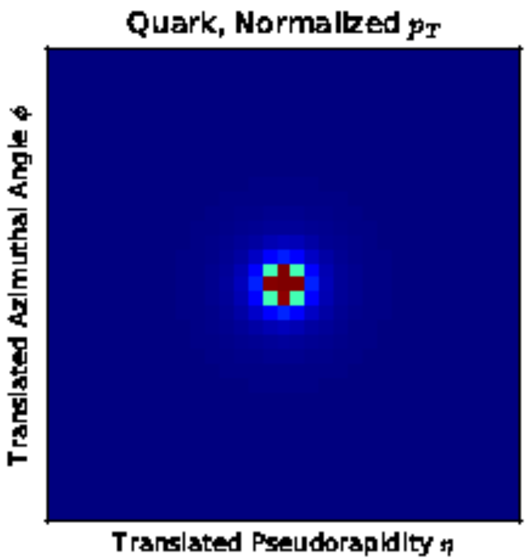
Gluon radiate more:

- Typically wider spread and softer particles

Both, quark and gluon have are prompt, i.e. displaced particles and vertices are not relevant

- Image approach proposed in 1612.01551





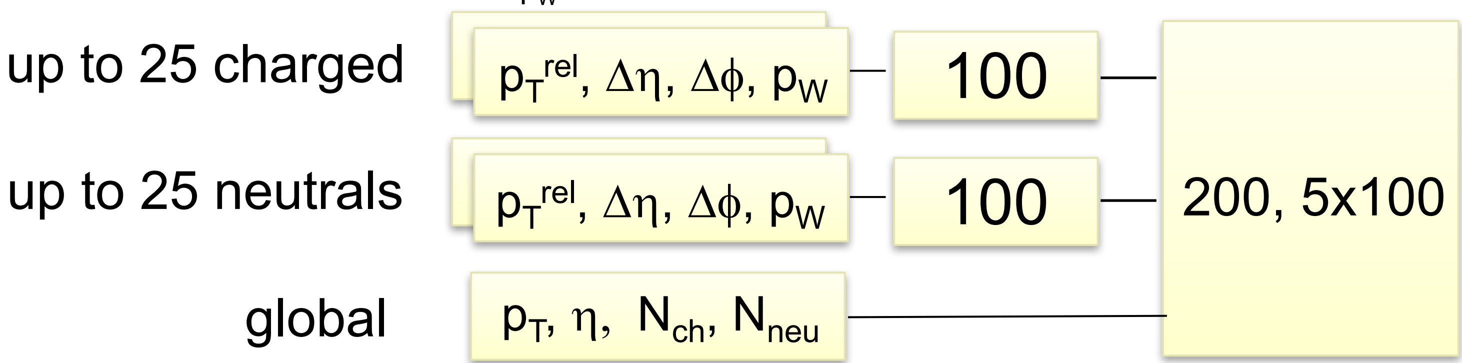
Quark gluon separation

Investigate a few custom DNN q/g tagging:

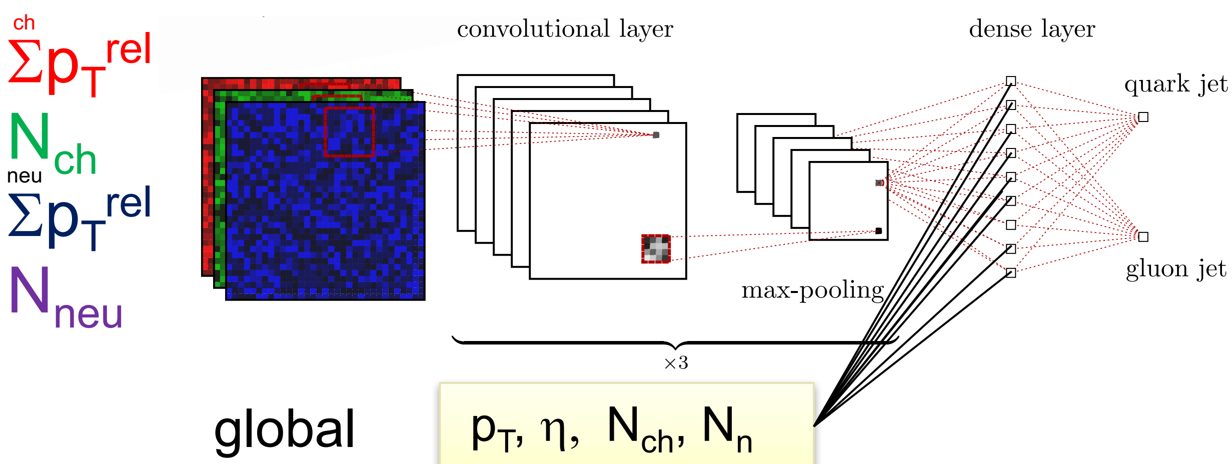
Recurrent for q/g:

Input features, p_T descending: RNN(LSTM) Dense

p_W as in 1407.6013

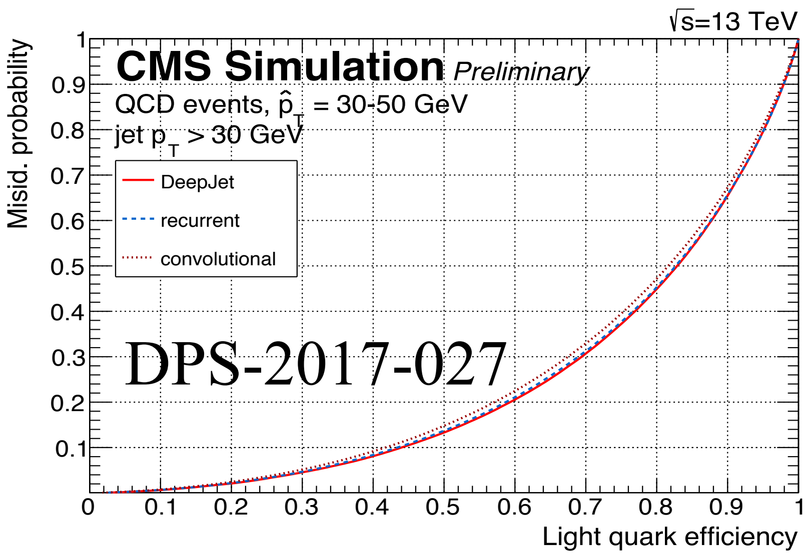


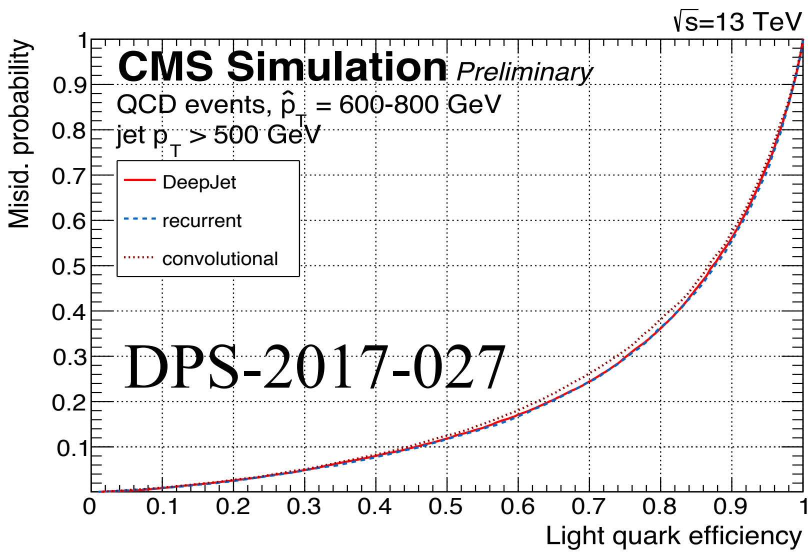
2D **convolutional**, four channels (CNN as in 1612.01551):



Comparisons of DNNs

- We filter on ***generator*** level only light quarks and gluons that did **NOT** split to heavy flavor.



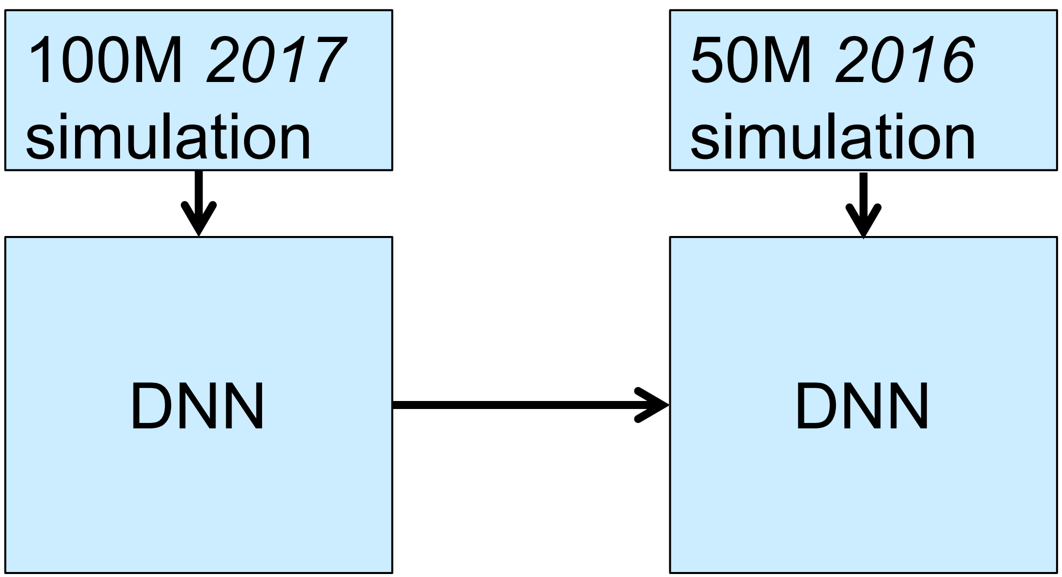


→ Generic DeepJet and custom q/g DNN gave very similar results!

→ **Data is multi-class, without heavy flavor removed DeepJet was clearly best**

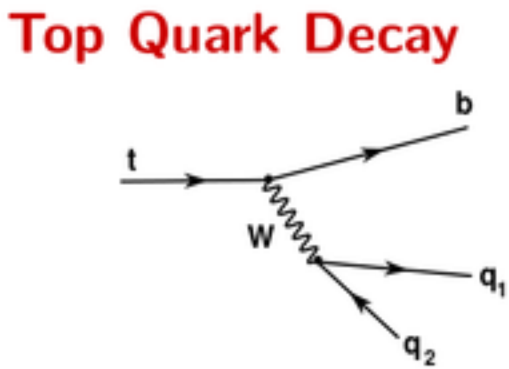
Pretraining

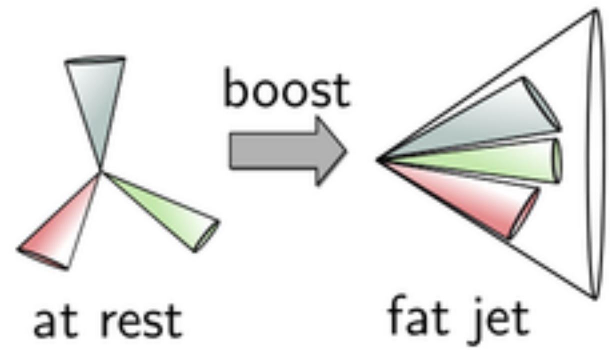
- New condition (PU or new geometry) require retraining of the network
- Use "similar" training sample with huge statistics to "pre-train"
- Increases effectively your data-sets



Used 2017 DNN as start or fixed some inner layers for 2016 DeepJet

Fat jets





Key features of tops:

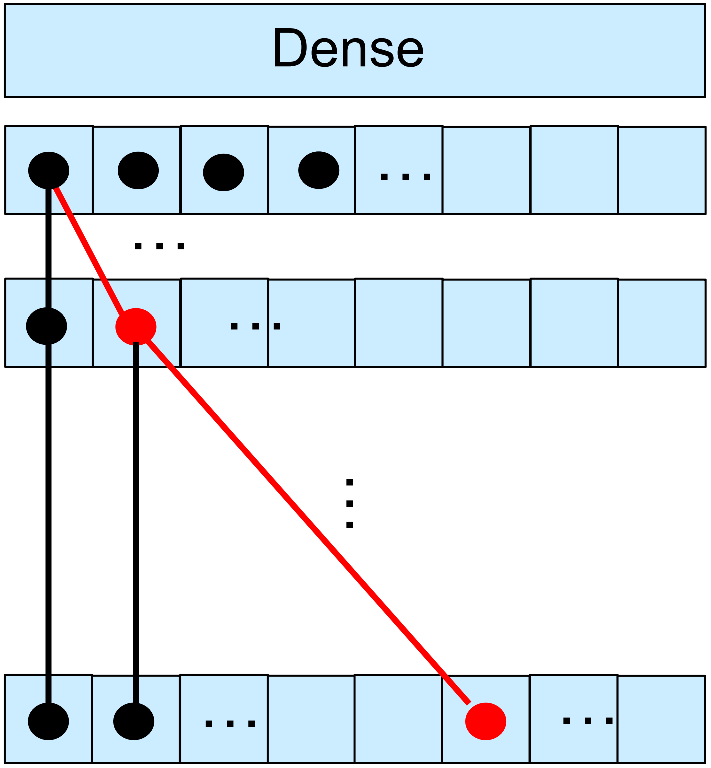
- M(W), M(t), W polarization
- 3 "prong"
- b-subjet and 50% with c-subjet

Top tagging is a combined problem of flavor tagging and substructure, masses with pileup, …

Good place for DeepJet approach starting from physics objects

Fat jet vs. slim jet tagging

1) More particles
2) More to learn:
   - Flavor tagging
   - quark vs. gluon
   - Mass of subjet combinations
   - All mixed if sub-jet merged

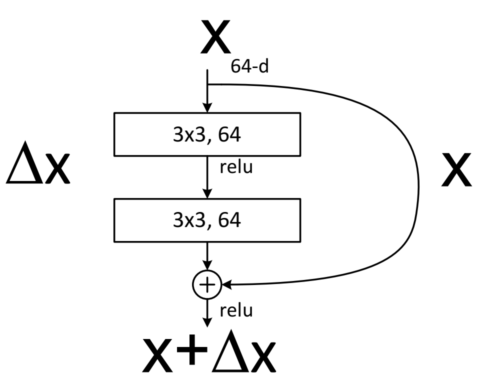


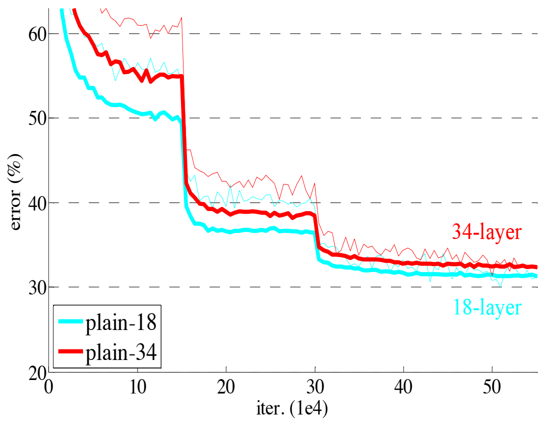
The slim jet DeepJet method slow for fat jets if RNN output and more particles are increased

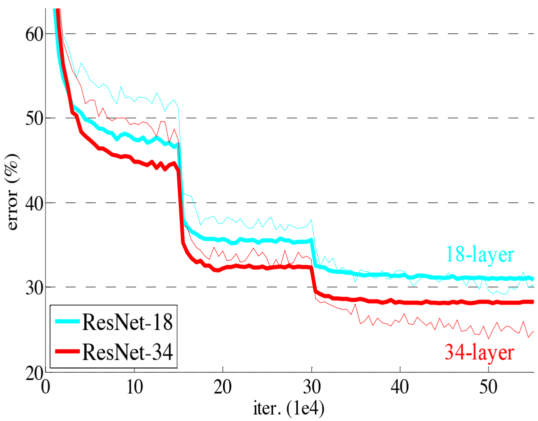
→ Keep concept of particles and vertices
→ Convolutional layer with kernel 3 to allow for long range correlation with *increasing* depth replaces slower recurrent network
→ Many more convolutional layers

Residual deep neural networks

- Adding more layers can degrade the result
- Later layers have to learn to not change x (identity) and add a correction (Δx)
- RESNETs only learn adding a residual Δx, not identity

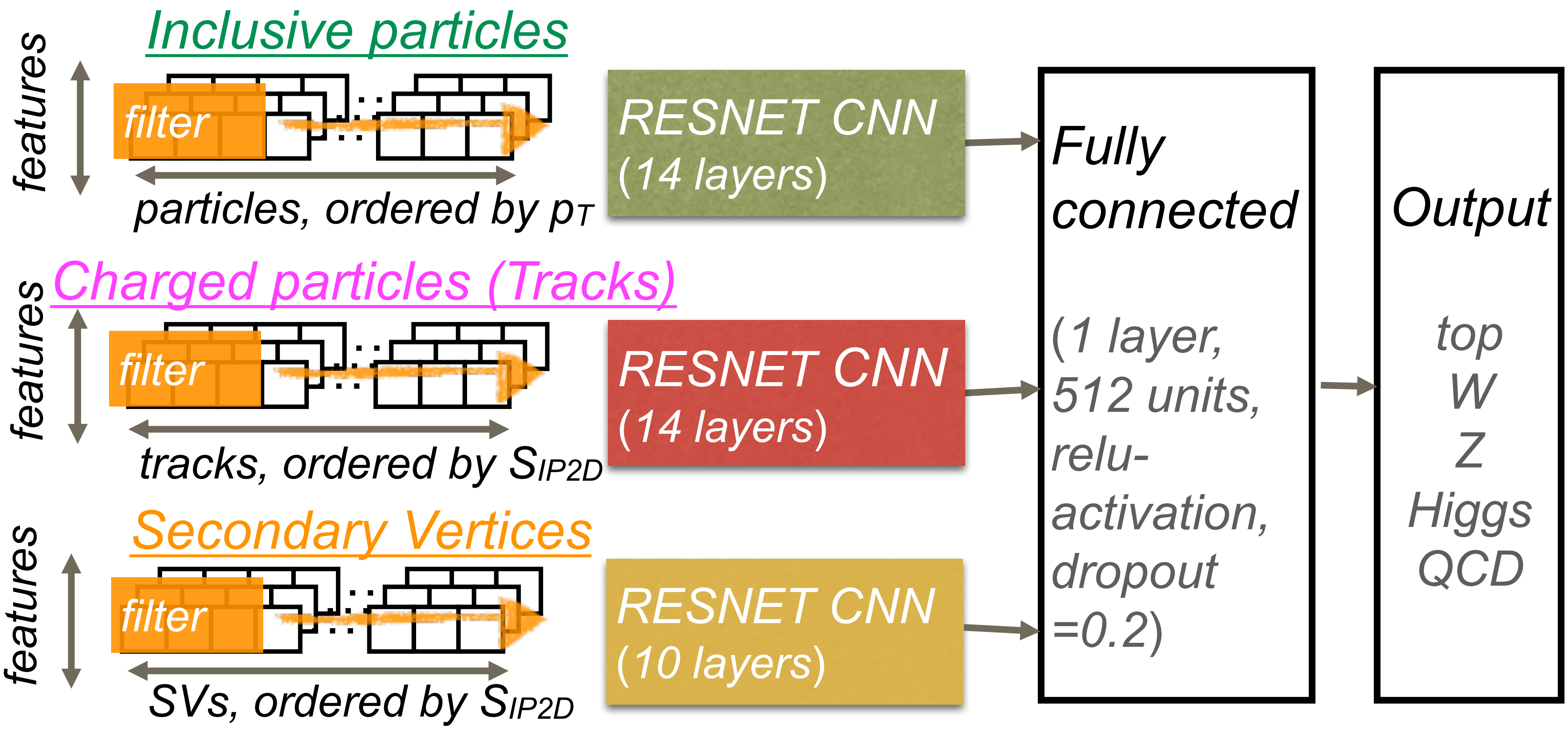






RESNETs useful for to make deep convolutional networks

DeepJet for fat jets



Kinematic: Only 3 vectors of particles → substructure , …

Full: all inputs → flavor tagging, substructure, …

BDT reference tagger

- BDT (full) using high-level features
  - Based on the top/W taggers used in SUS-16-049
    - inputs: jet kinematics, Nsubjettiness ratios, soft drop mass, subjet mass, subjet Q/G discriminator, and CSV b tag
  - added variables used by the boosted double-b tagger [BTV-15-002]
  - trained with the same samples as DeepJet

- BDT (w/o b-tag info):
  - all input variables, except for subjet CSV b tag

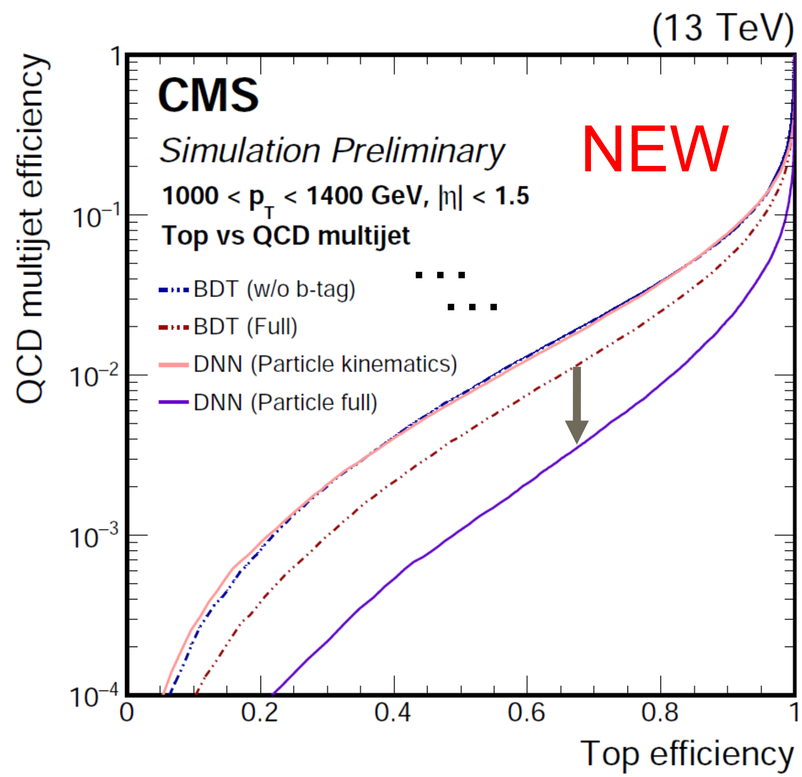
Very competitive tagger to compare with DeepJet

Comparing fat DeepJet vs. BDT

CMS DP-2017/049

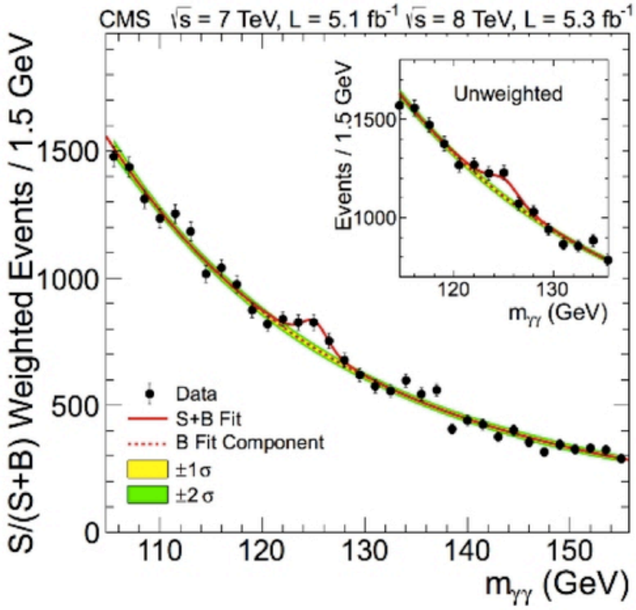
- DeepJet "kinematics" similar to BDT without b tag

- With full information for BDT and DeepJet perform much better (factor 3-4 @ 1% BKG)



- Big gain not in sub-structure, but combining structure, PU, and flavor
- Previous DNN proposals focused only on structure (image)

Independence of classifier of certain features

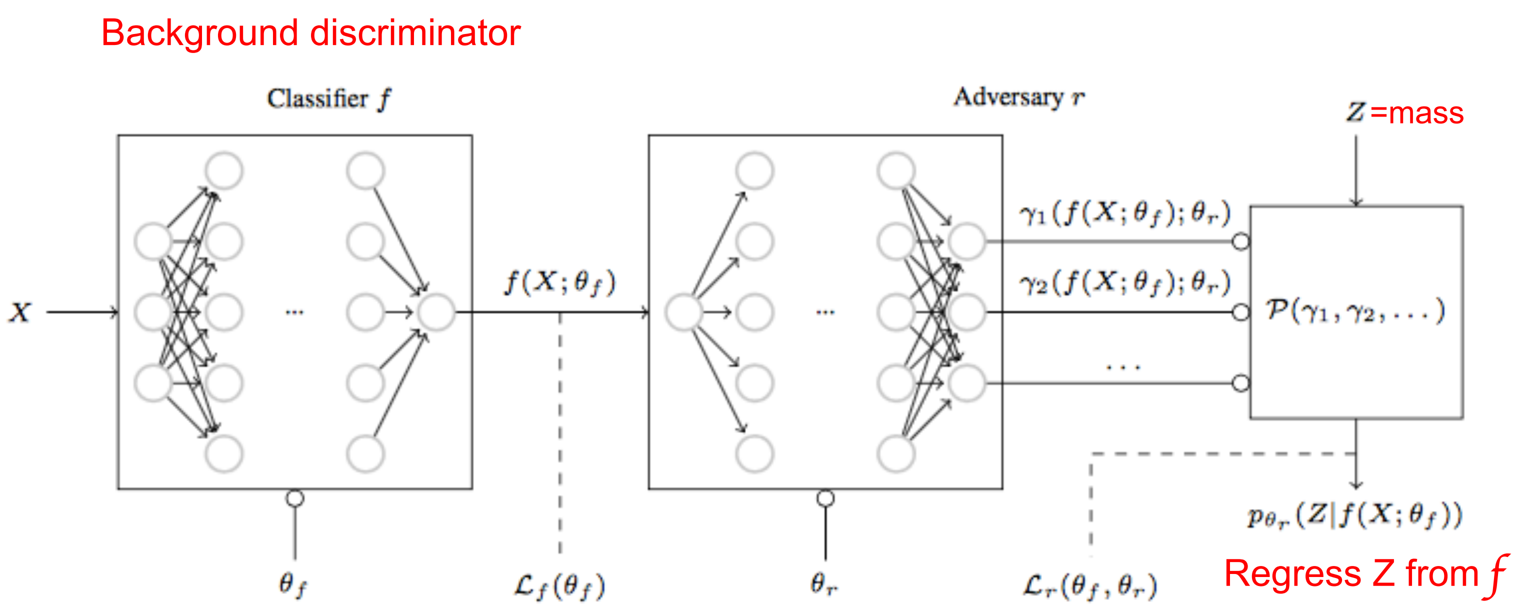


Simple bump-hunt:

- Fit a function to "side-band" to estimate background
- Check for bump

- Used a classifier threshold to increase signal fraction in sample, but want to avoid **artificial** bump in background
- Many features depend on mass (X), i.e. classifier likely as well even without adding the mass
- Enforce independence of classifier on mass (X)

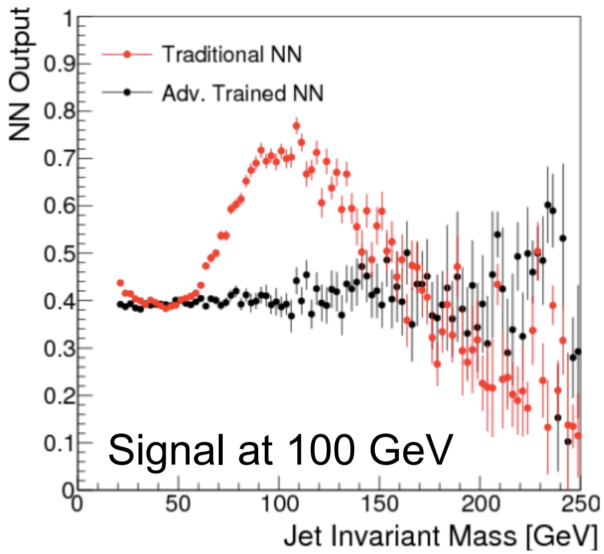
Adversarial training

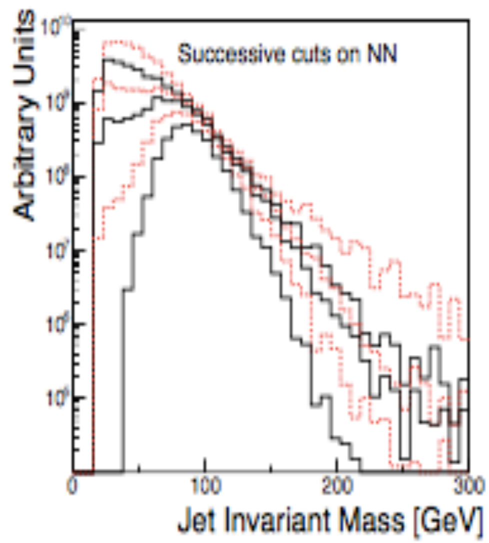


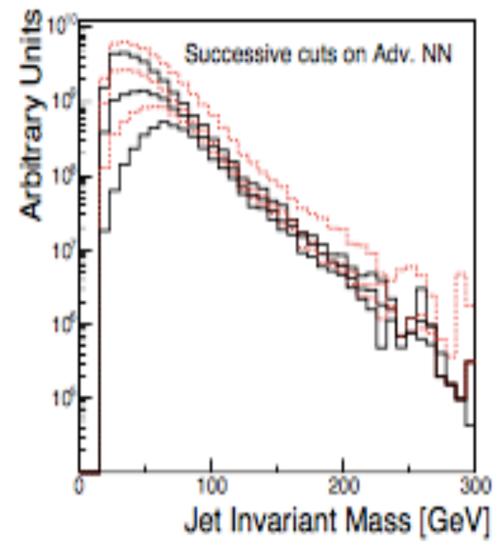
$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r)$$

Intuition: enforce that you cannot infer the “mass” from the discriminator output arXiv:1703.03507

Test of method on search with jet mass







- Dependence of NN output on mass significantly reduced
- Mass shape less effected by cuts on discriminator
- Tested also for DeepJet top tagger!

Summary: DeepJet in CMS

- Deep learned jet tagger for different cones sizes
- Custom DNN architectures and big datasets used
- Best performance:
  - Slim jets b, c, uds, g
  - Fat jets: top, W, Z, H (heavy flavor), QCD tagging
- Fat jet tagging version with mass independence existing

Use data only?

Learning by label proportion (semi supervised)

https://papers.nips.cc/paper/5453-almost-no-label-no-cry.pdf

"Small prints apply", e.g. some constraints on loss functions, …

Mean pred. prob.

Loss function

$$f_{\text{weak}} = \text{argmin}_{f':\mathbb{R}^n \to [0,1]} \ell\left(\sum_{i=1}^{N} \frac{f'(x_i)}{N} - y\right)$$

Known prob. to be of a class

In words: DNN output mean = label proportion

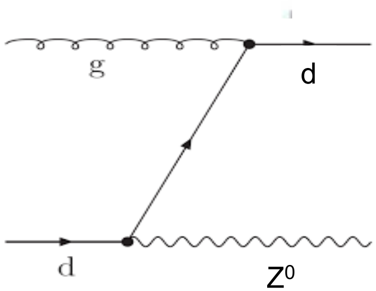
If you have several sets with know label proportions, this is enough for learning.

Just using sets with different label proportions

https://arxiv.org/pdf/1702.00414.pdf

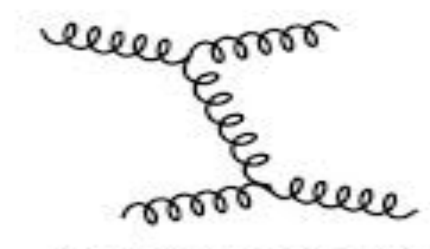
Indeed, it is sufficient to have different, but unknown label proportions

Z^0+jets:



many quark jets

Dijet:

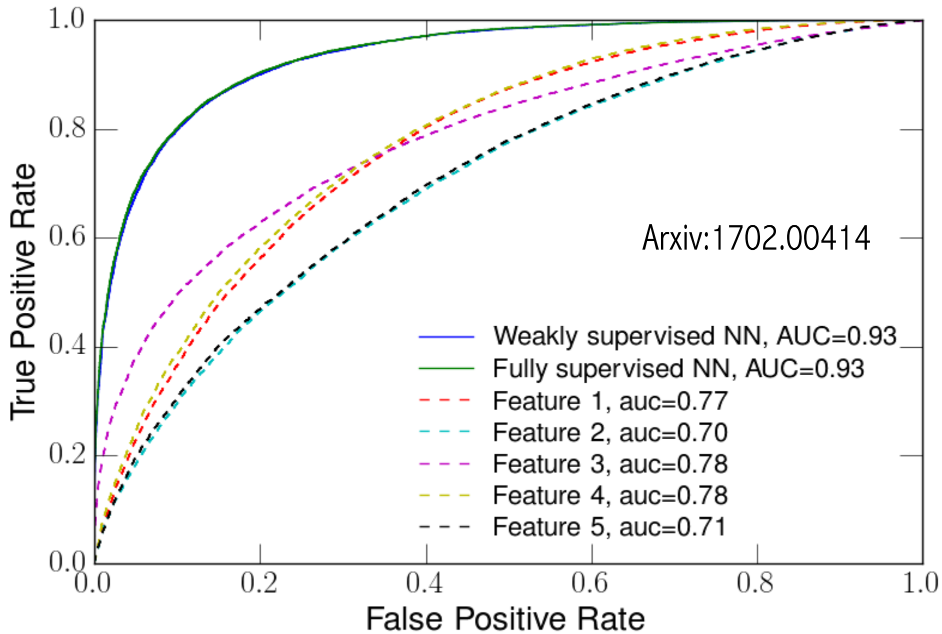


many gluon jets

Need more than **ONE** data set

Quark gluon data only example

Test in simulation with known labels and a simple neural network:
→ Weakly and fully supervised lead to same performance



Very interesting approach with a few caveats:

- Limited statistics in data in tails → tricky for deep learning
- Assumes that quark gluon is the ***ONLY*** difference, e.g. color reconnections are different and many classes present
- You cannot make a ROC curve, i.e. do not know the performance

Use data and MC?

Domain adaptation

Source domain (MC)

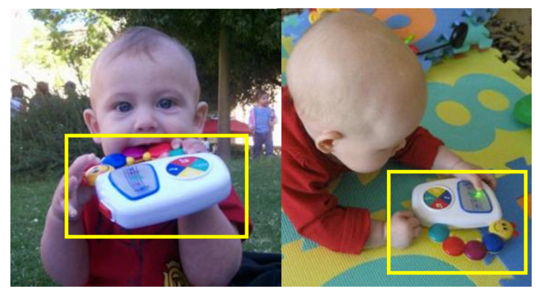
Good samples with **labels** for training a classifier

digital SLR camera

amazon.com

Target domain (real data)

low-cost camera, flash



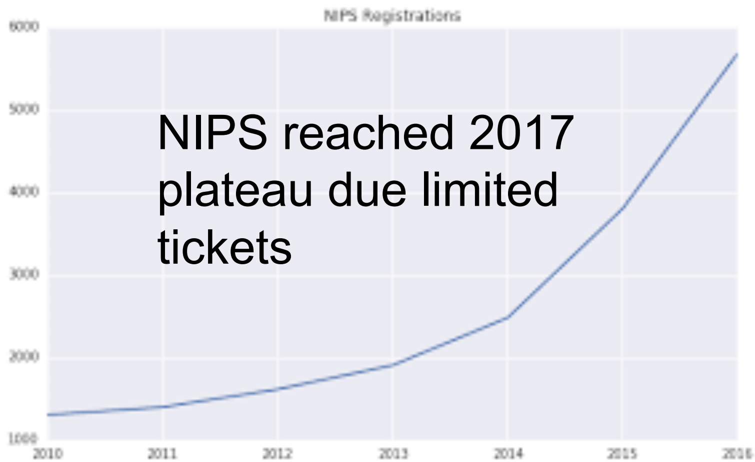
consumer images

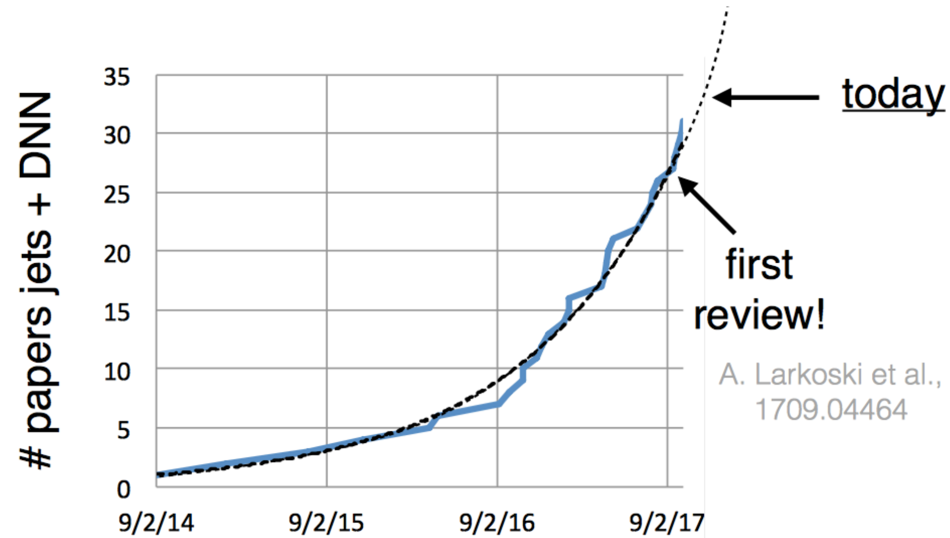
User samples to apply the training, **no labels** available

Much literature; mainly aimed to have good performance of classifier in target domain.

arXiv:1702.05464v1

Deep learning at LHC



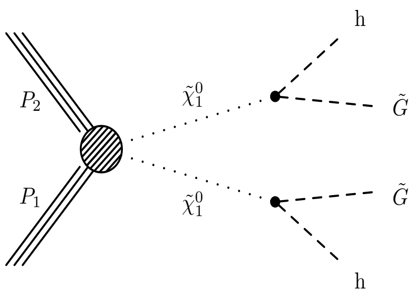


- Deep learning community continues grow at LHC and elsewhere
- NN toolkits improved as well
- Without higher energy collisions we need better data analysis to keep progressing in science

Application in physics analysis

SUS-16-044:

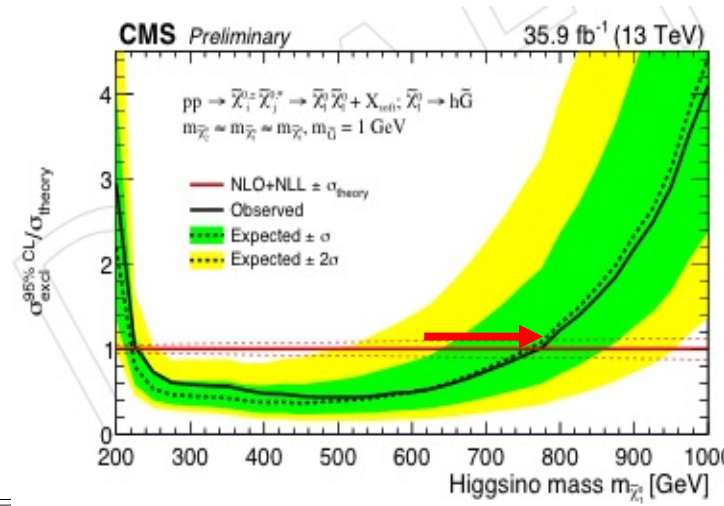
Search for events with two h->bb and MET



2b ≡ $N_{b,T} = 2$, $N_{b,M} = 2$

3b ≡ $N_{b,T} \geq 2$, $N_{b,M} = 3$, $N_{b,L} = 3$

4b ≡ $N_{b,T} \geq 2$, $N_{b,M} \geq 3$, $N_{b,L} \geq 4$



CSVv2 $\mathcal{L} = 35.9$ fb^{-1}	All SM bkg.	TChiHH (225,1)	TChiHH (700,1)
$\geq$ 2b	–	3761.5	33.7
$\geq$ 3b	–	1999.1	19.0
4b	–	860.0	9.3
Baseline, $\geq$ 2b	2600.1±101.0	75.6	7.7
Baseline, $\geq$ 3b	276.9±5.5	49.6	5.4
Baseline, 4b	72.2±4.1	30.9	3.6
Baseline, $p_T^{miss} > 300$, $\geq$ 2b	104.2±2.4	2.8	6.0
Baseline, $p_T^{miss} > 300$, $\geq$ 3b	12.9±0.8	2.4	4.2
Baseline, $p_T^{miss} > 300$, 4b	4.0±0.4	1.7	2.8

DeepCSV $\mathcal{L} = 35.9$ fb^{-1}	All SM bkg.	TChiHH (225,1)	TChiHH (700,1)
$\geq$ 2b	–	4625.6	39.7
$\geq$ 3b	–	2548.7	24.1
4b	–	1149.1	12.7
Baseline, $\geq$ 2b	3650.5±90.2	95.1	9.9
Baseline, $\geq$ 3b	385.2±9.0	68.6	7.4
Baseline, 4b	94.3±5.3	43.4	5.1
Baseline, $p_T^{miss} > 300$, $\geq$ 2b	144.8±2.8	4.0	7.7
Baseline, $p_T^{miss} > 300$, $\geq$ 3b	16.3±0.8	2.2	5.7
Baseline, $p_T^{miss} > 300$, 4b	4.6±0.4	2.5	4.0

Significant Improvement: e.g. up to ~50% more signal for 15% more bkg

→ Significantly improved lower mass limit (150 GeV in Higgsino mass)