Imperial College London

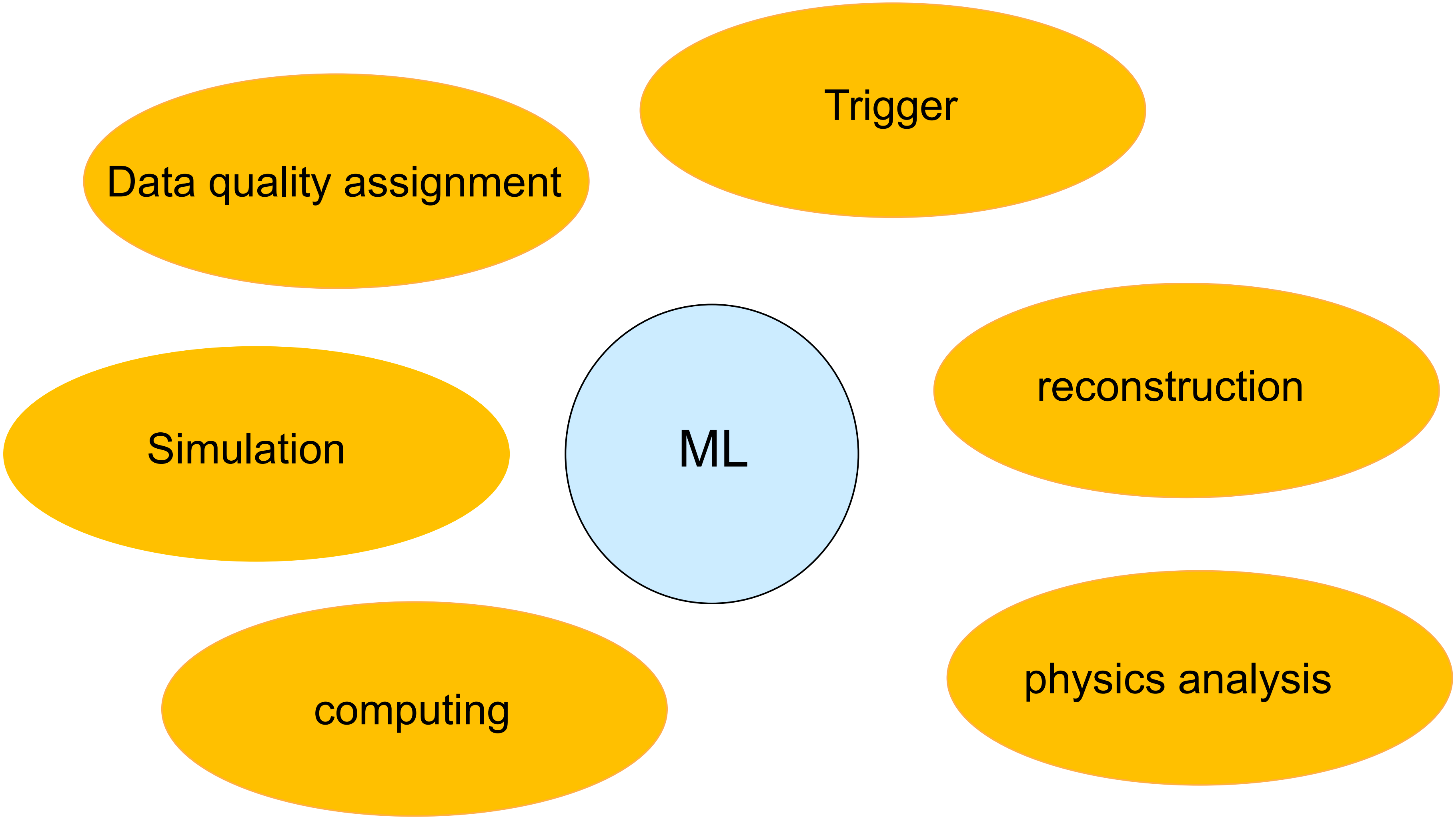
Data Science Institute

This Report is part of a project that has received funding from the **European Union's Horizon 2020 research and innovation programme under grant agreement N°675440**

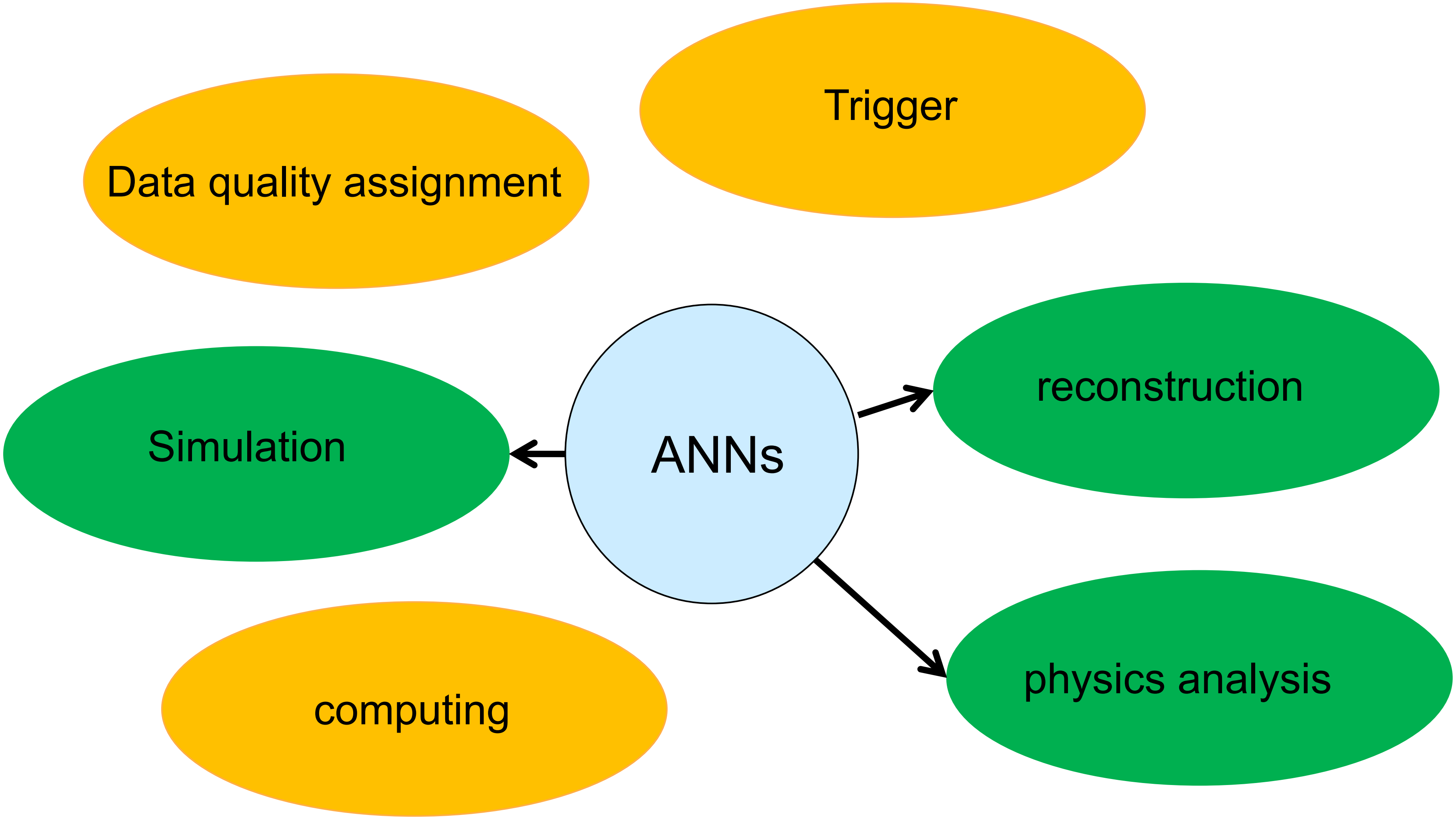
Machine learning at the LHC

Markus Stoye
Imperial College London, DSI, aMVA

Leiden, 17th , January 2018



- Machine learning (shallow) is essential at the LHC



- Machine learning (shallow) is essential at the LHC
- I will focus on physics relevant newer developments in context of (deep) **neural networks**

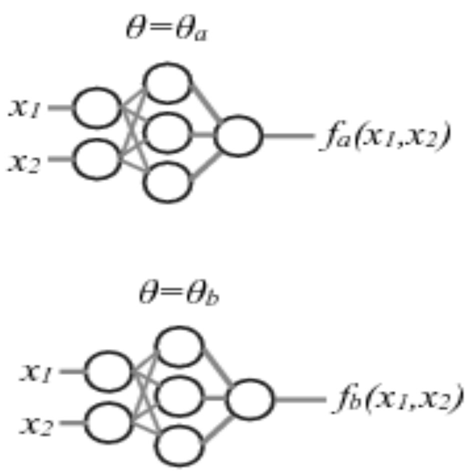
physics analysis

arXiv:1601.07913

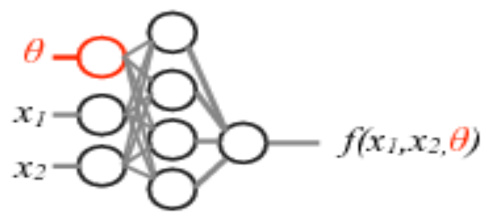
Parametric Neural Networks for scans

Situation: Simulation scans, a few discrete values of a real number physics parameter q are simulated
Question: How to best use this in training of classifier?

Train for different θ →



Parameterized neural networks with mass as input

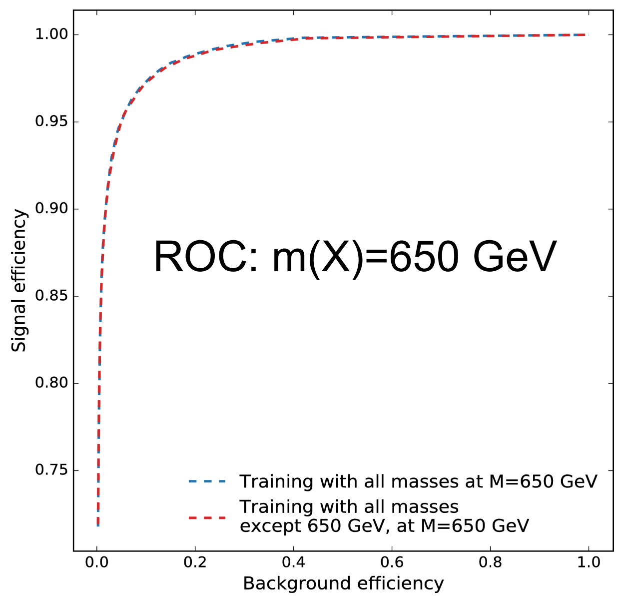


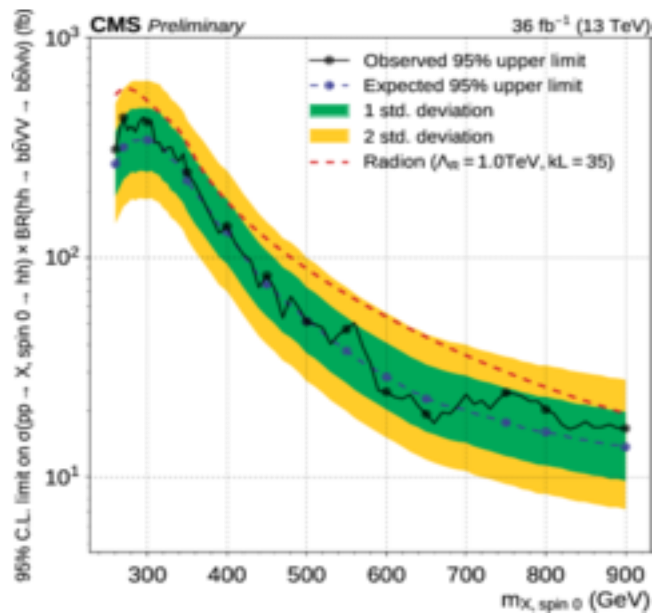
- Background is given random parameters q values samples from signal *PDF*
- PNN interpolates between masses and smoothly work for all masses

Results with PNN

- Search for X → hh → bblνlν (CMS-PAS-HIG-17-006)
- Not a unique signal, θ = ***mass(X), unknown***

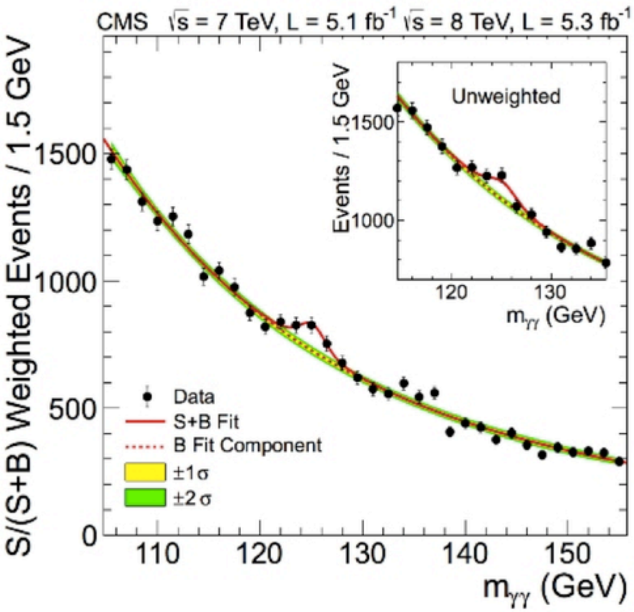
Blue: Only use m(X)=650 GeV
Red: trained not using m(X)=650 GeV





- Same ROC performance show that the interpolation works
- Smooth (no bumps) and good separation over full mass range

Independence of classifier of certain features



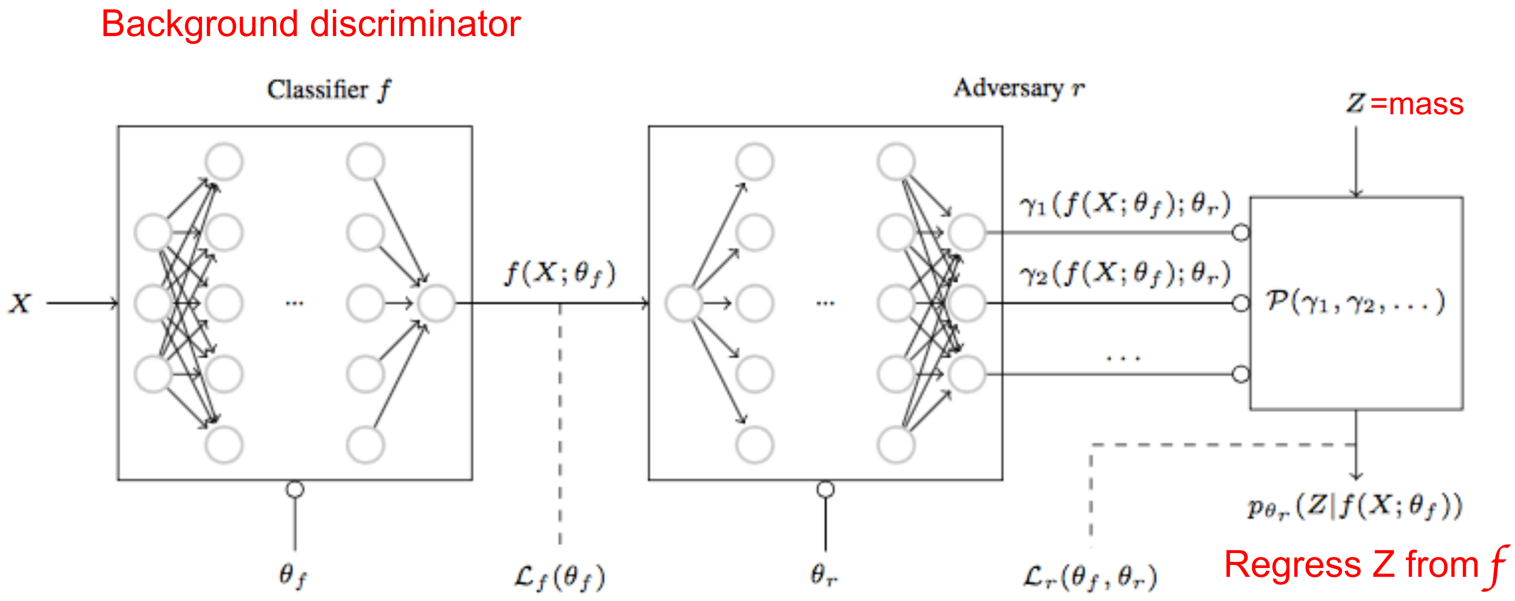
Simple bump-hunt:

- 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)

arXiv:1611.01046

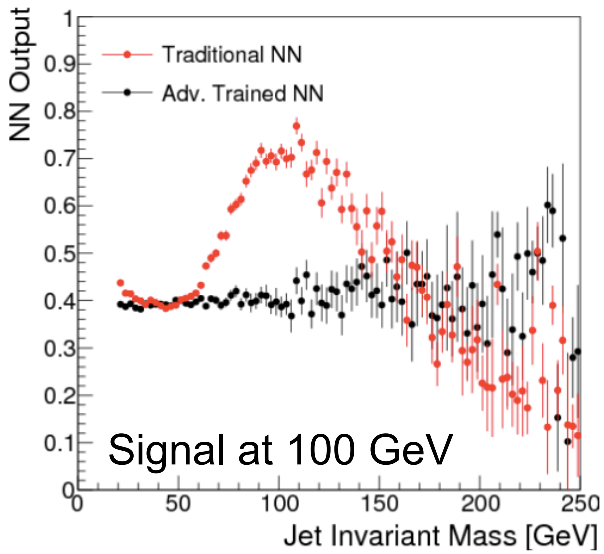
Adversarial training

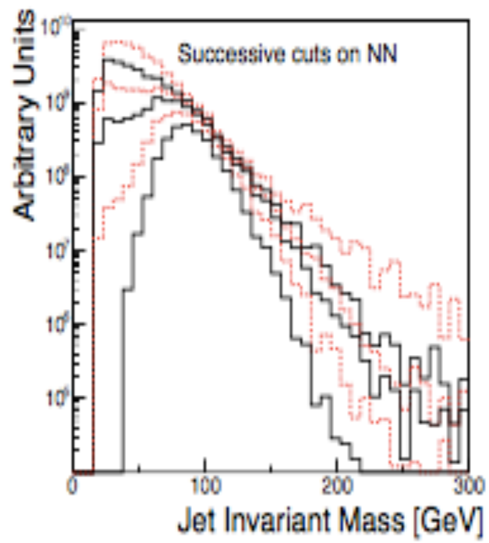


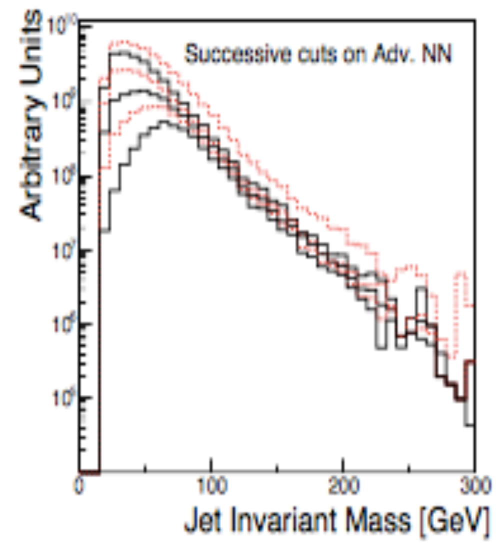
$$\hat{\theta}_f, \hat{\theta}_r = \arg\min_{\theta_f} \max_{\theta_r} \mathcal{L}_f(\theta_f) - \mathcal{L}_r(\theta_f, \theta_r)$$

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

Test of method on search with jet mass



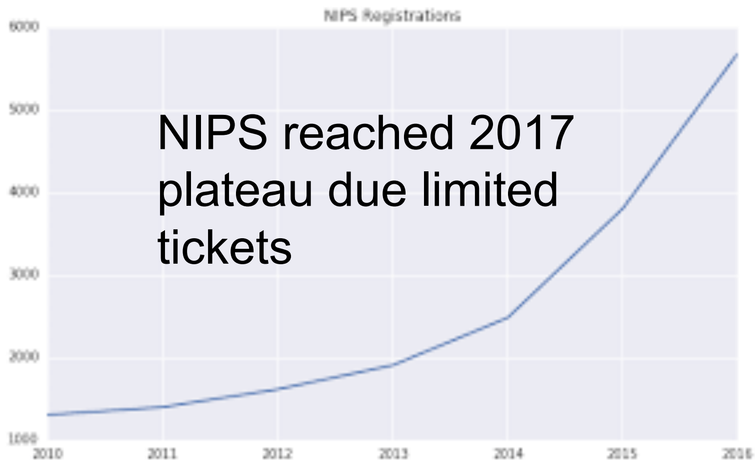


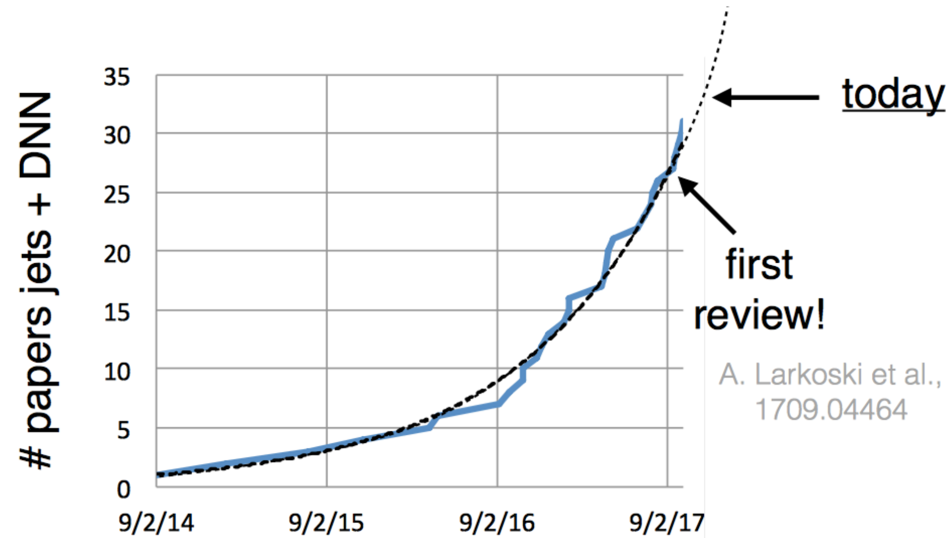


- Dependence of NN output on mass significantly reduced
- Mass shape less effected by cuts on discriminator

reconstruction

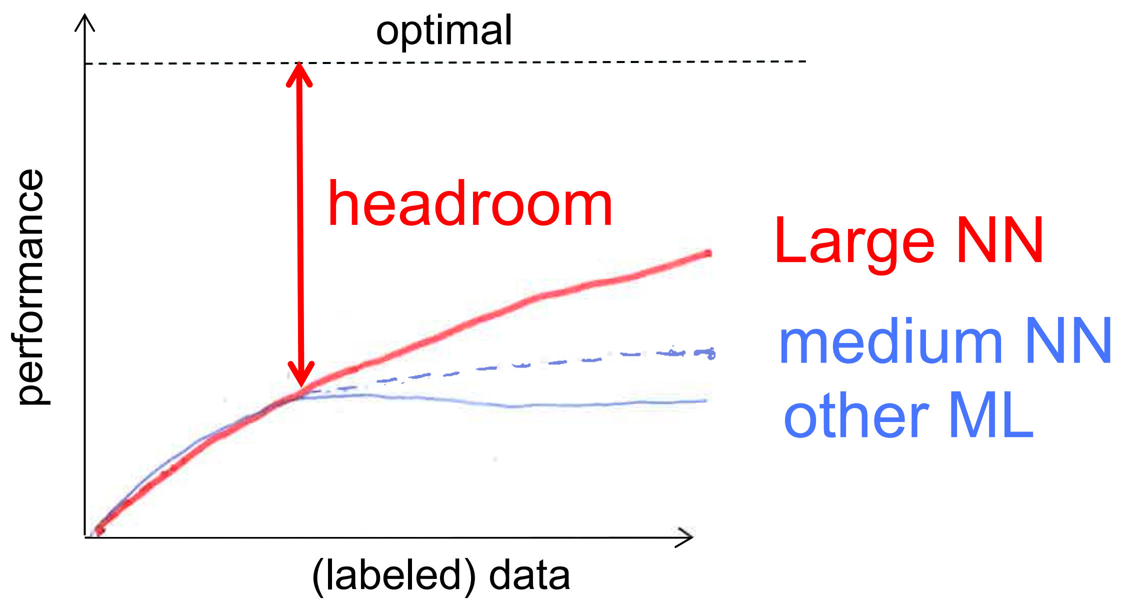
Deep learning at LHC





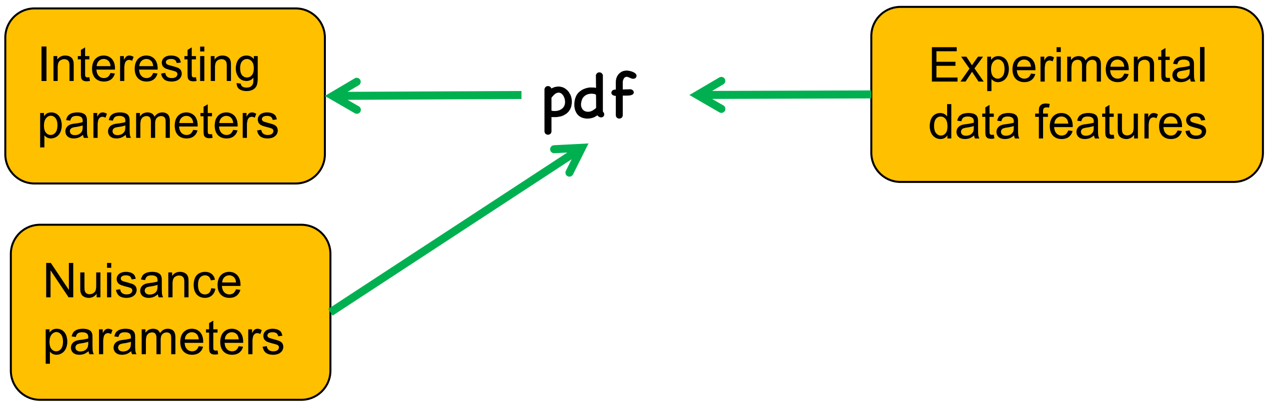
- Deep learning community continues grow at LHC and elsewhere
- NN toolkits improved as well
- Without higher energy collisions we need better data analysis to keep progressing in science

Deep learning: more is better



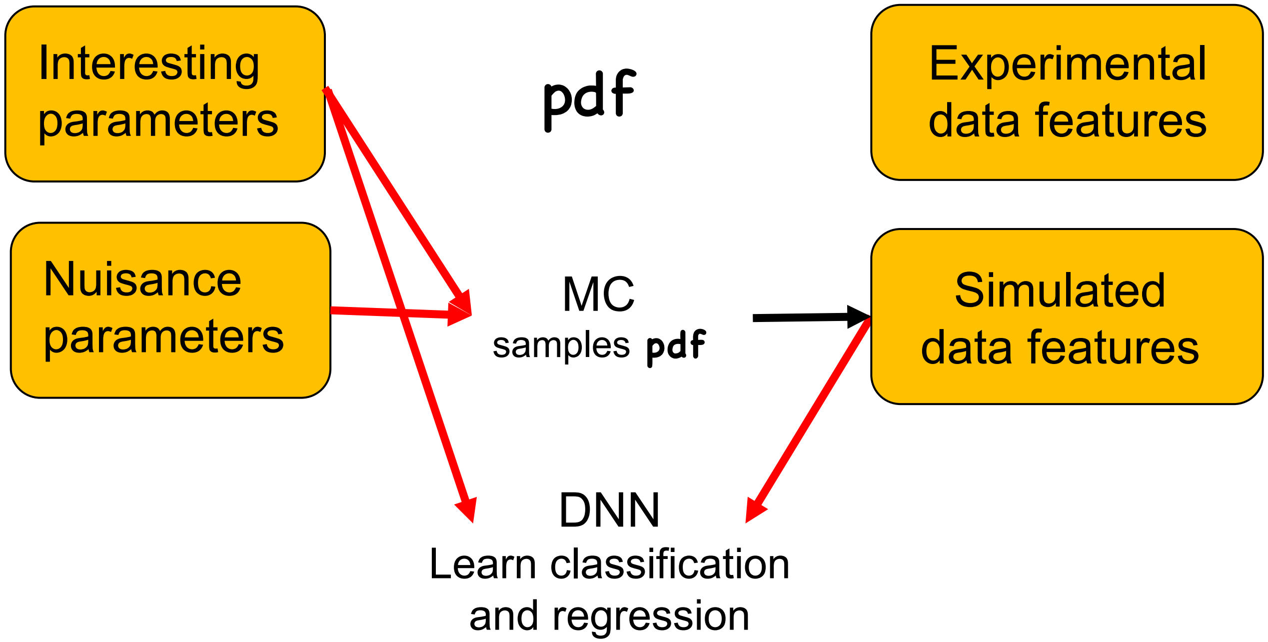
- High dimensional inputs with big dataset and a large Deep Neural Networks brought breakthroughs
- We have huge numbers of simulated samples with truth information 😀
- It is very hard to estimate the ***headroom*** left 😕

Infer SM or NP parameters from data



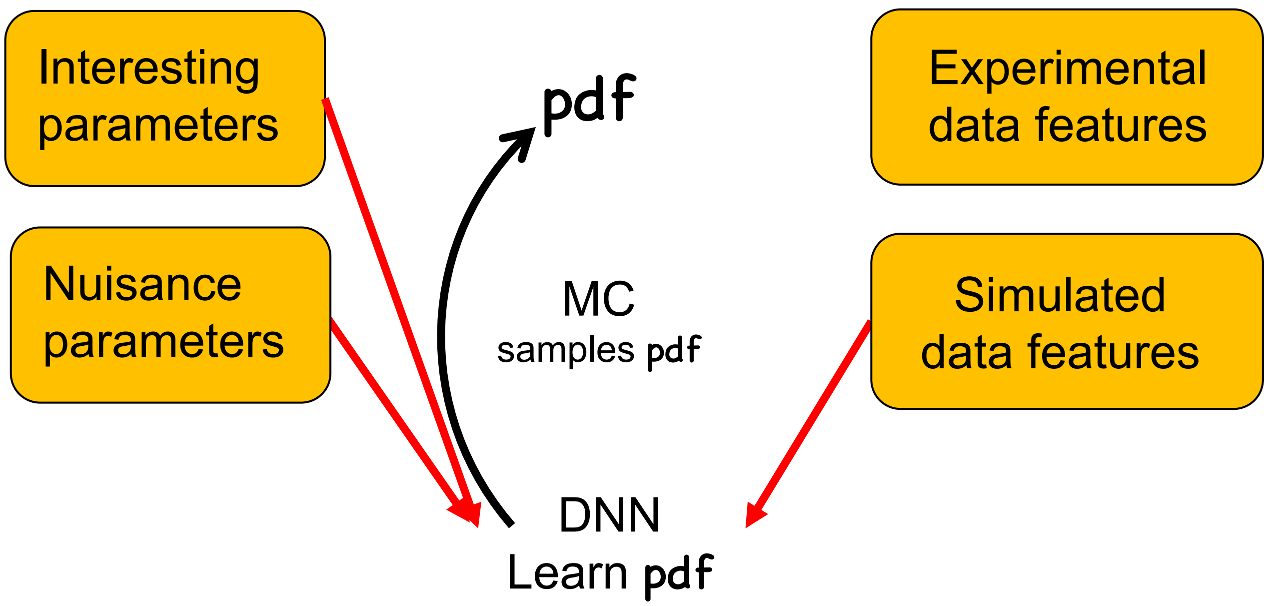
- *Ideally* we would have the **pdf** for likelihoods
- We can not write the **pdf** down analytically for our complex experiments

Supervised deep learning to estimate parameters



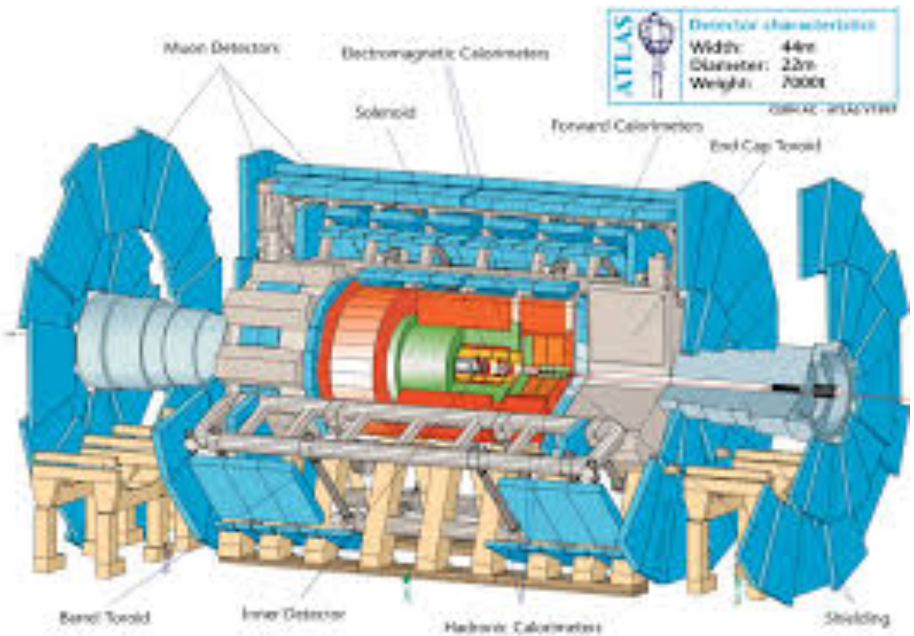
- Practically we can make MC simulation
- We that we can try a ML to estimate interesting parameters

Deep learning for distributions



Ultimately we could even learn the pdf

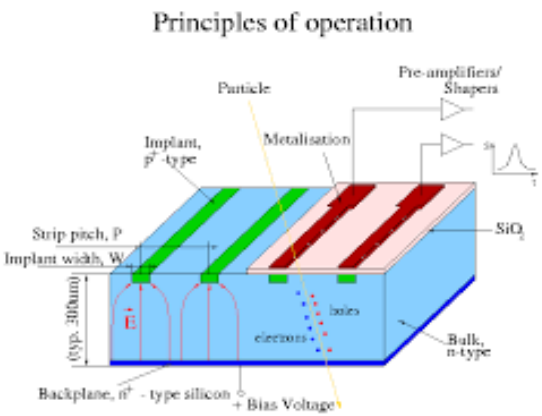
Overview of dimension involved



- **pdf**(X,SM+NP, q)
- Experimentel features: $X_{DIM} \sim$ O(100 M)
- Theory parameters: $(SM+NP)_{DIM} \sim$ few handfuls
- Calibration constants ect. $\theta_{DIM} \sim$ 1 M (nuisance parameters)

Process needs to be factorized in a chain

From raw data to intermediate physics features



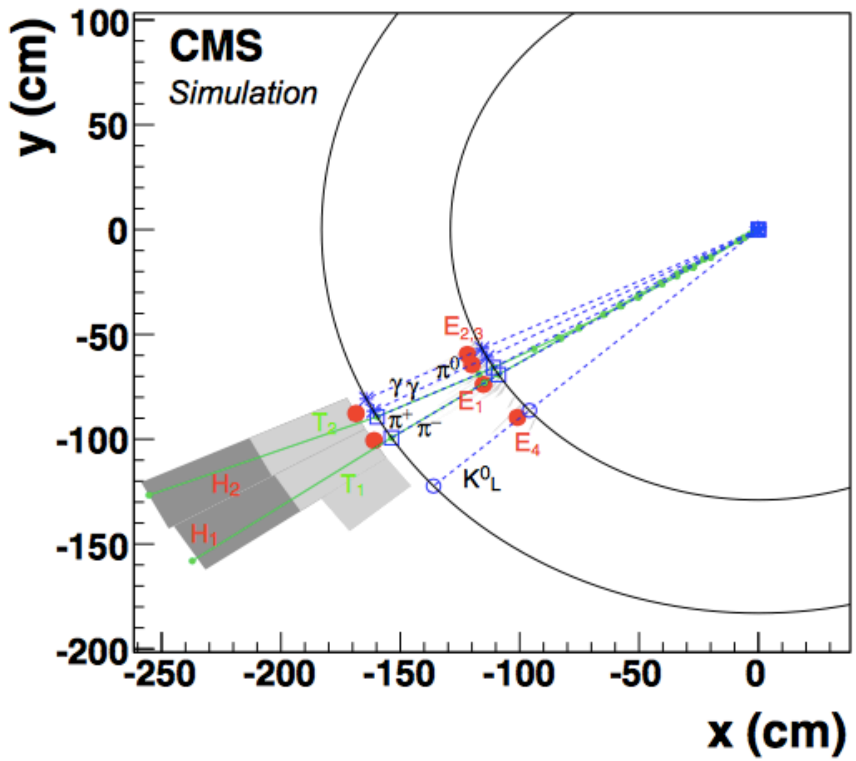
1) Analyze raw data per sensor
2) Find other sensors with signals of track(s)
3) Robustly fit track parameter, MLE (momenta), (higher level)
4) …

- We start by transforming raw data to a physically meaningful (lower dimension) representation

Higher features, particles that left signals in the detector

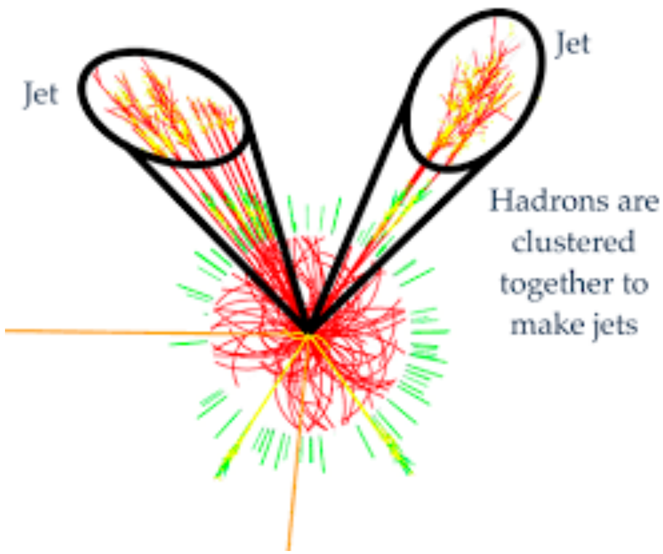
a) Match intermediate information from different detectors to built particle features, e.g. particle ID (muon), momenta, …

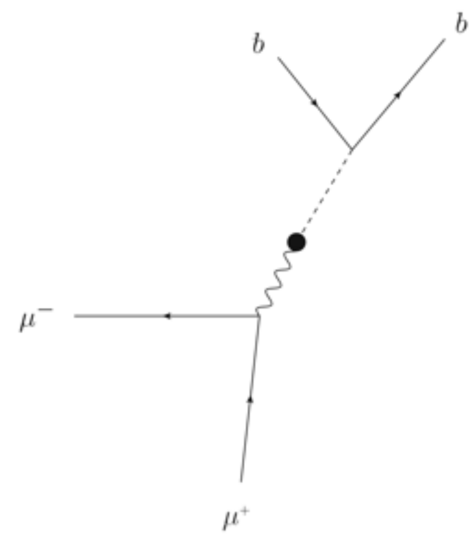
b) **Particle flow** assigns all intermediate physics features to particles features (particle candidates)



Building highest level features for final data analysis

Parton level picture





1) Assign (we cluster) particles to a partons we can calculate
2) Estimate hard particle's features
3) Parton's features analyzed allows using shallow ML, MLE, …

Reconstruction chain

Forward chain with increasingly high level features

Improve individual pieces of the chain

- Deep learning already in standard reconstruction chains
- Jet classification
- Silicon sensor hit reconstruction
- tracking
- …

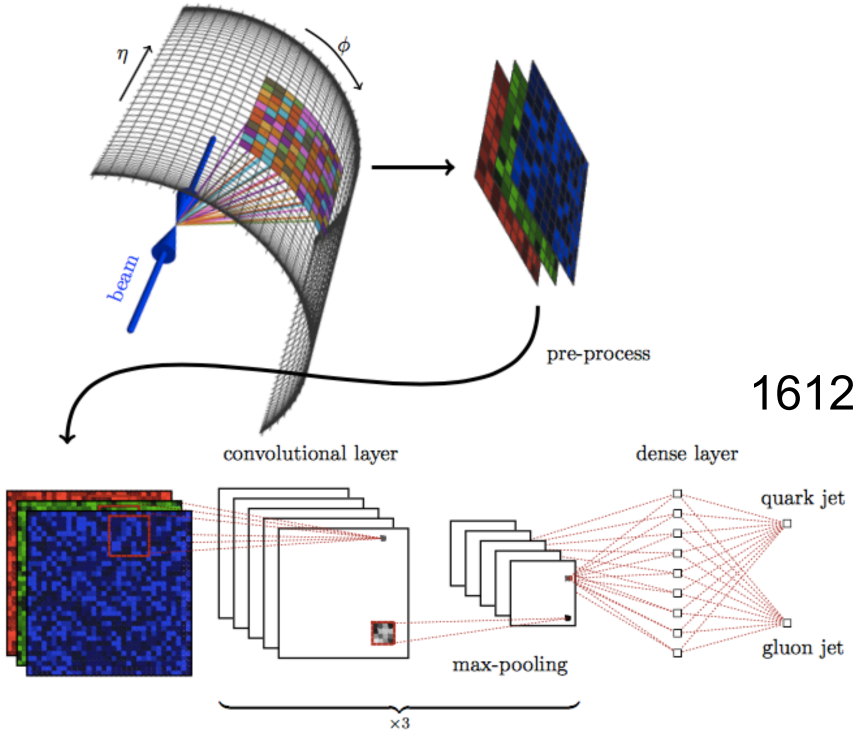
Do a few of the chain's pieces simultaniously

- First positive feasibility studies

end-to-end-learning (f* the chain)

- Mostly an idea at this point

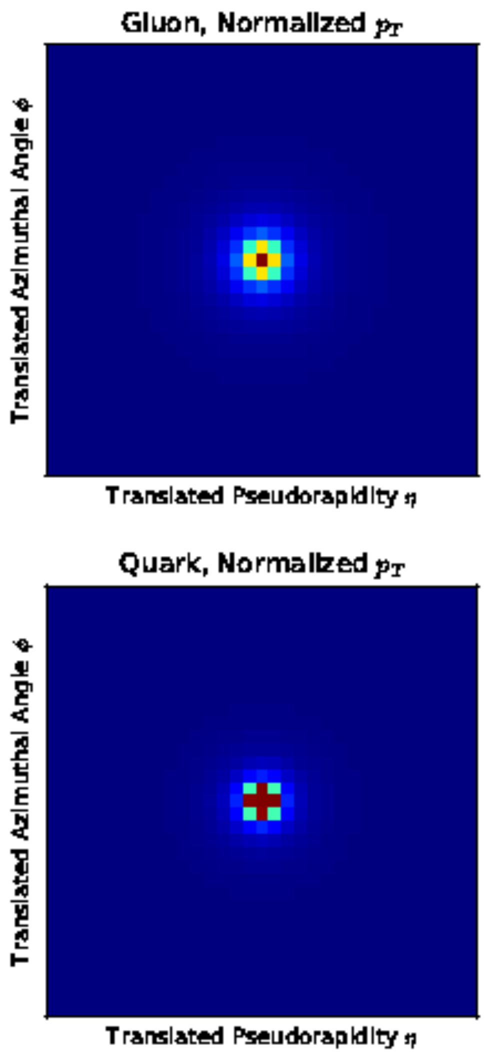
Jet images



1612.01551

- 2D **convolutional**, e.g. from calorimeter cells
- A natural representation of pure calorimeter information
- Not all inductive biases, e.g. translational invariance, of convolutional networks apply in real detectors!

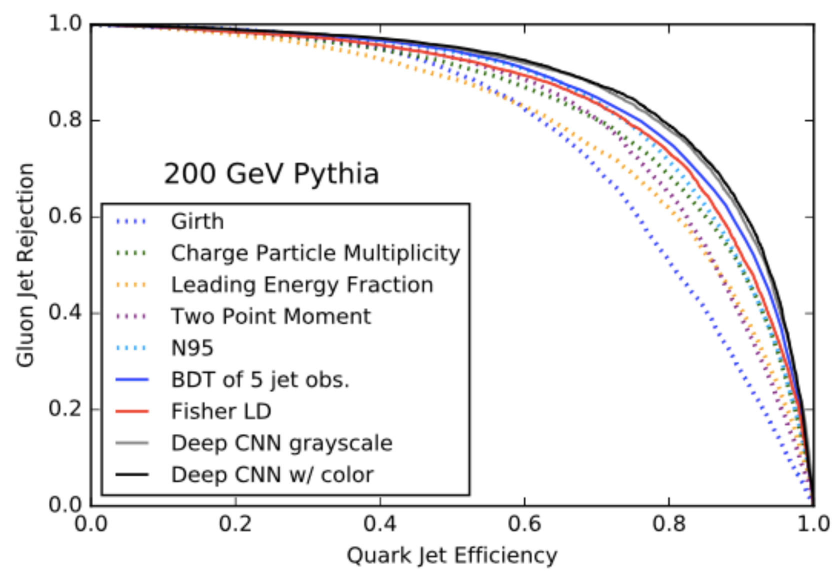
Quark vs. gluon jet classification



Gluon radiate more:

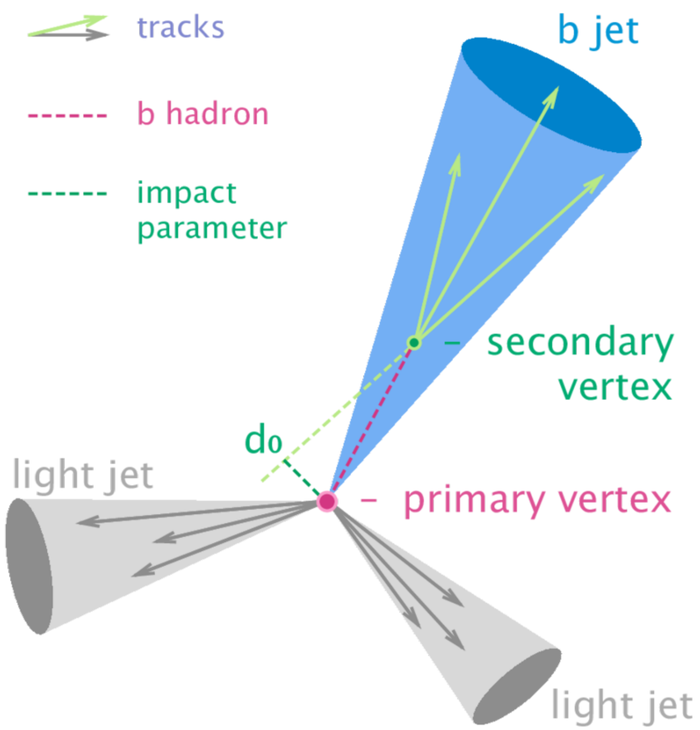
- Typically wider spread and softer particles
- Thinner and harder particles

Energy densities captures by calorimeter!



Mild performance gain with respect to traditional methods (BDT)

Flavor tagging (b,c, tagging)

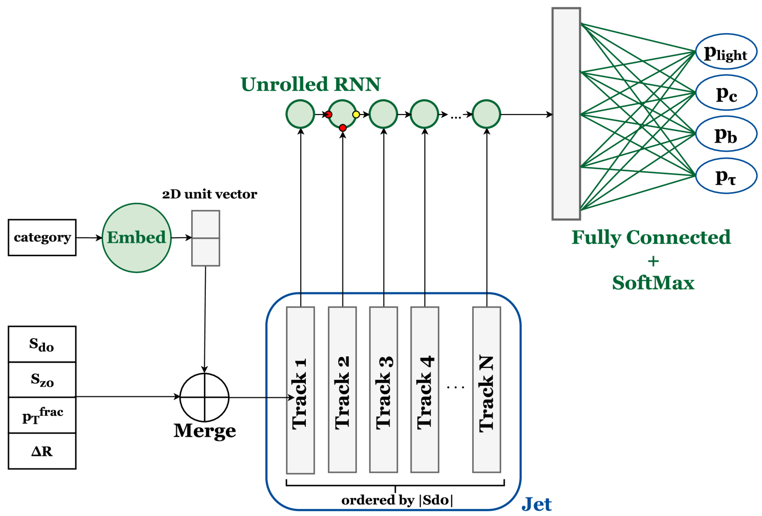


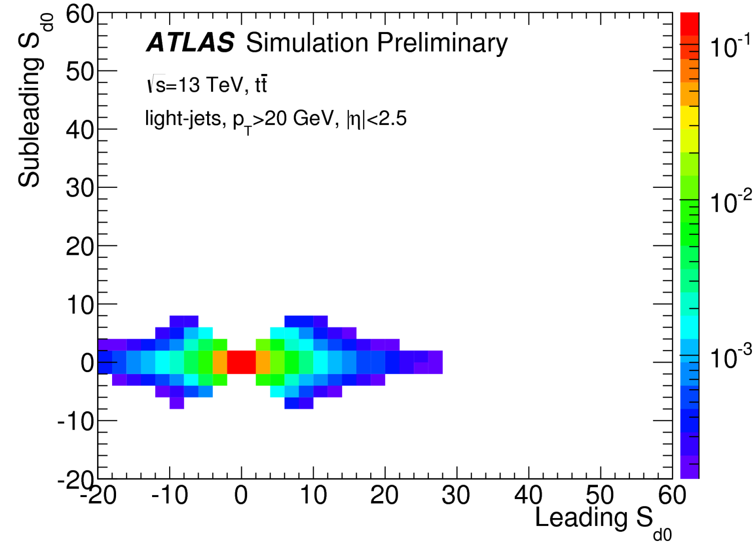
Key features:

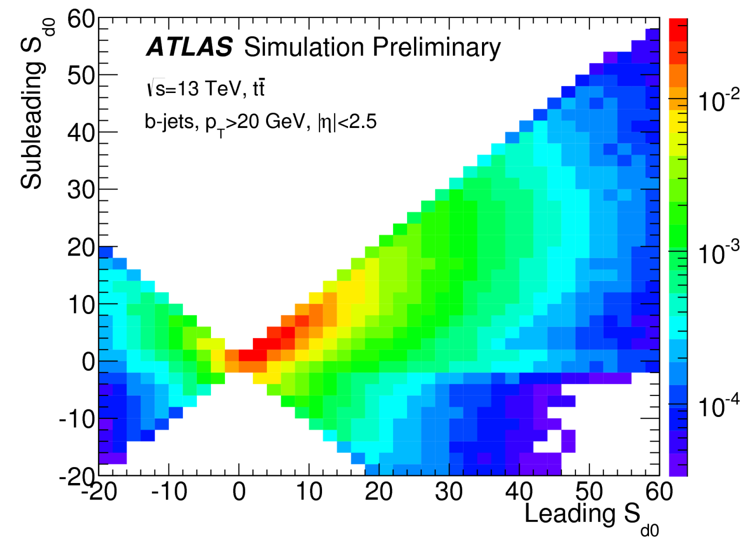
- Displaces tracks (d_0) or secondary vertices
- Tracks and vertices more complex structure than calorimeter
- Number of tracks vary

Image not a solution for flavor tagging

Recurrent network for tracks

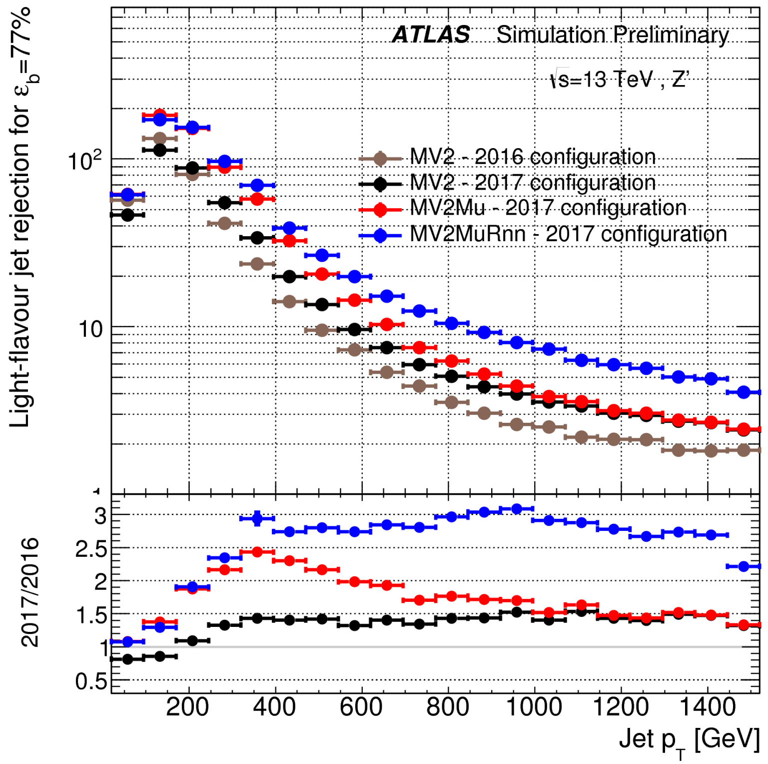






- **15** most displaced Tracks fed into recurrent network
- Takes for example correlation between tracks displacements (d_0) into account

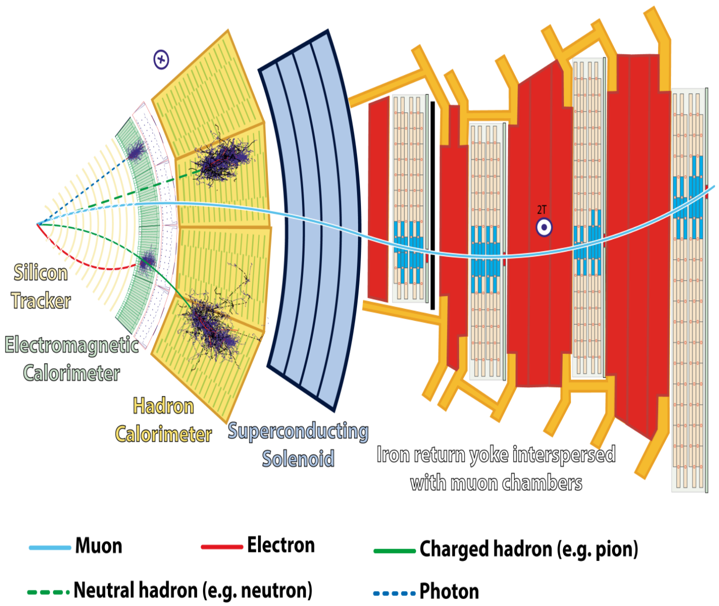
Recurrent network for tracks



Significantly better results at high momentum

Complete jet: particle flow candidates and physics objects

CMS particle candidates contain "most" information originating from a "particle"



"Complete" jet information

- All particle candidates of a jet and many features per particle
- Add in addition vertices aligned to the jet
- Jet PT, h and number of vertices in events for PNN

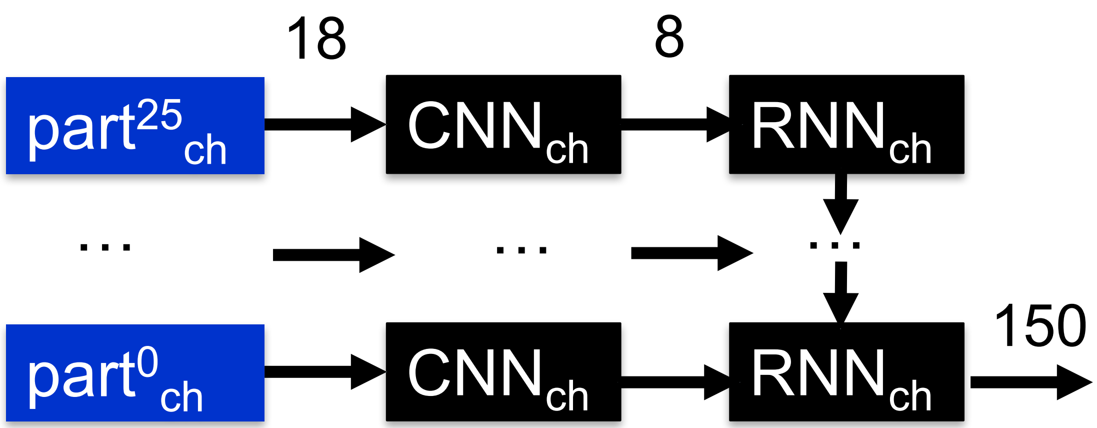
About 1000 features

"Complete" jet input can be used for multi-class classification or regression

Physics object based NN architecture for jet input

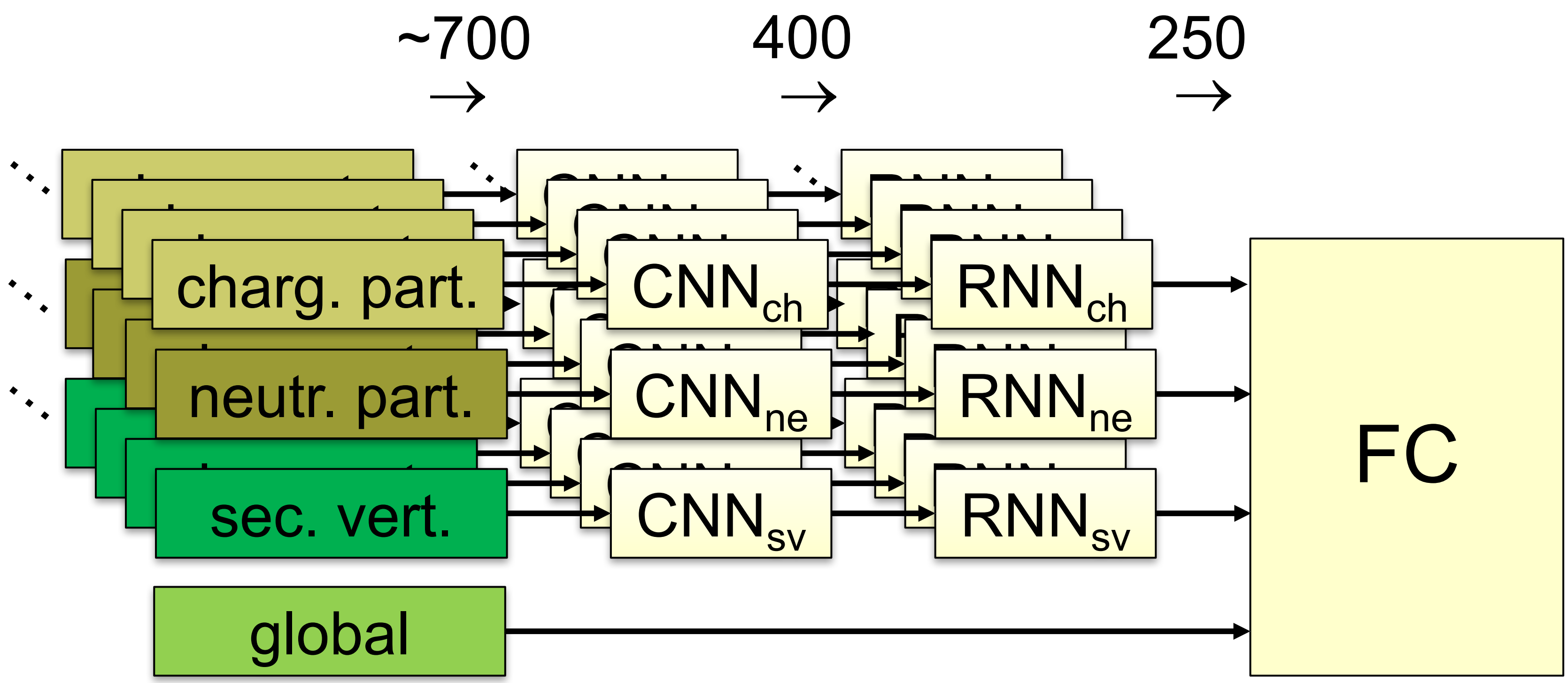
Example: charged particle candidates

- Four 1x1 1D CNN layers reduces 18 to 8 features (feature engineering) or you can see it as non linear (4 layers) particle embedding



- 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

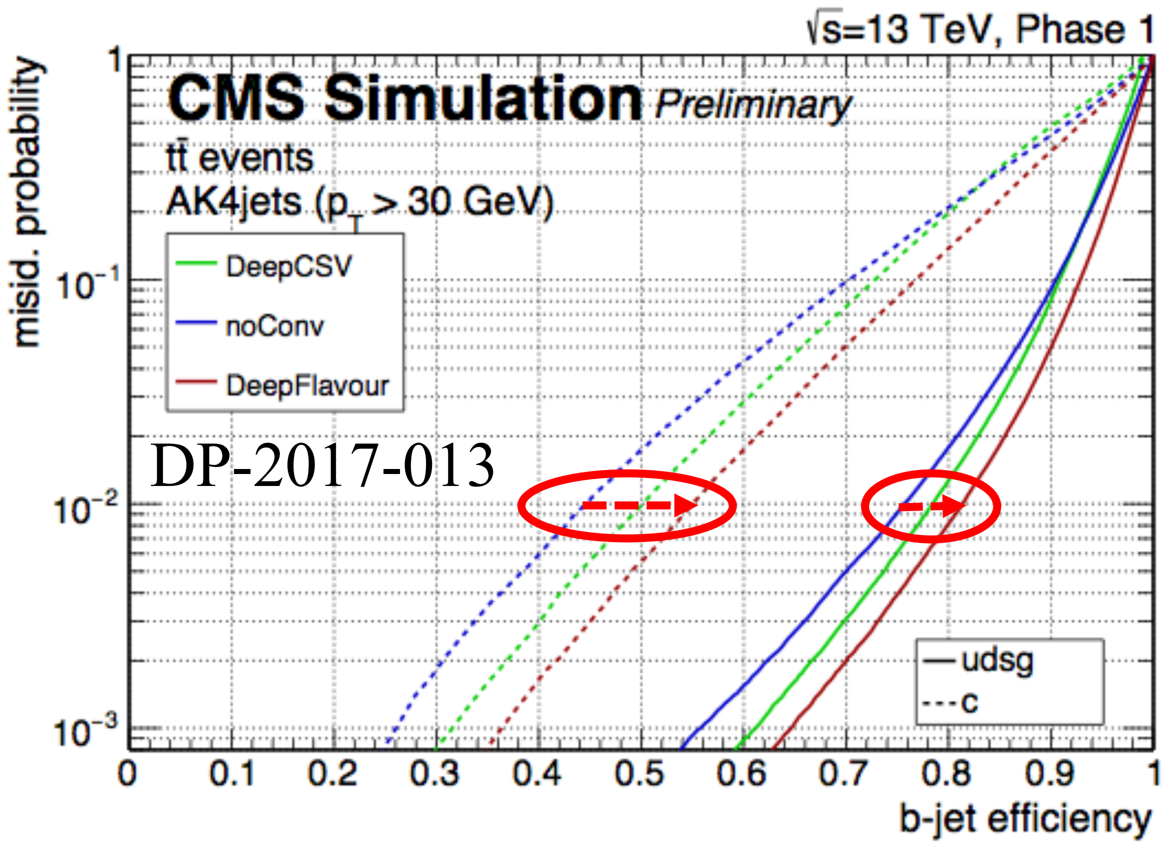
Particle and vertex based DNN: DeepJet



~ 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

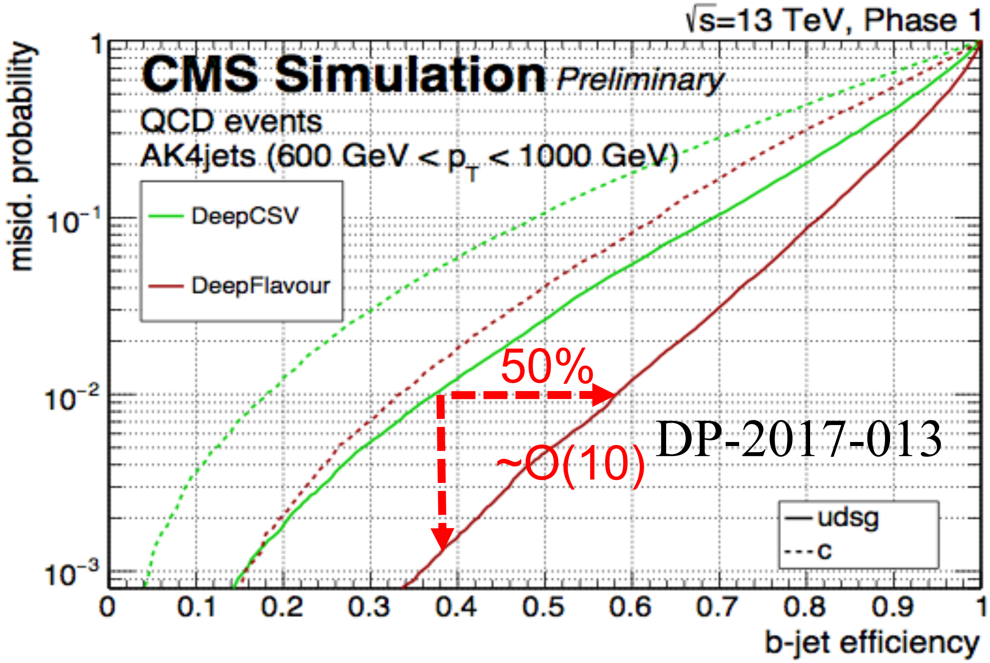
Impact of DNN architecture



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

Particle and vertex based DNN performs best

DeepJet results



Very significant gain at high p_T

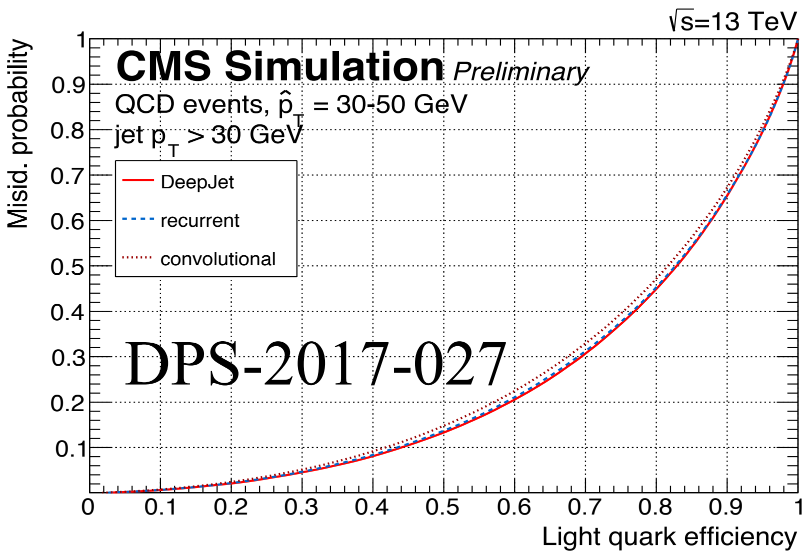
Increase input step by step from DeepCSV:

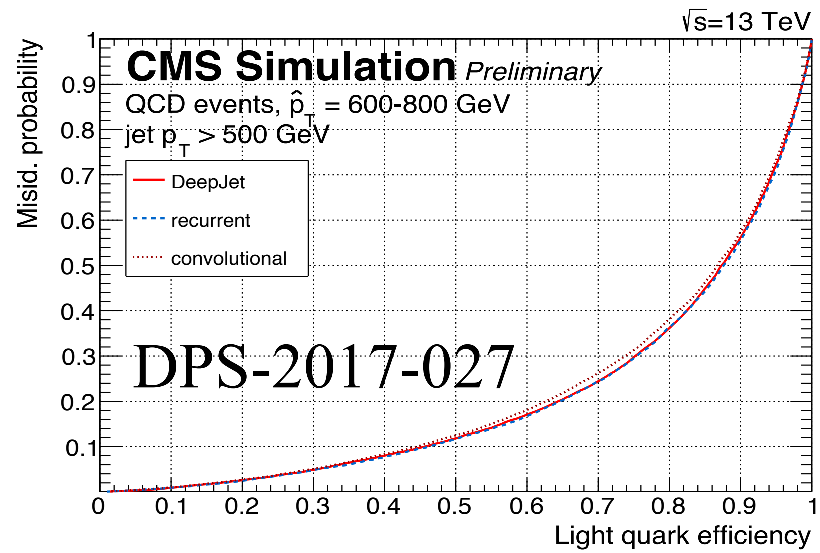
- Not applying former track selection helped
- More features helped

Not yet confirmed in data, validation ongoing

Comparisons of DNNs

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

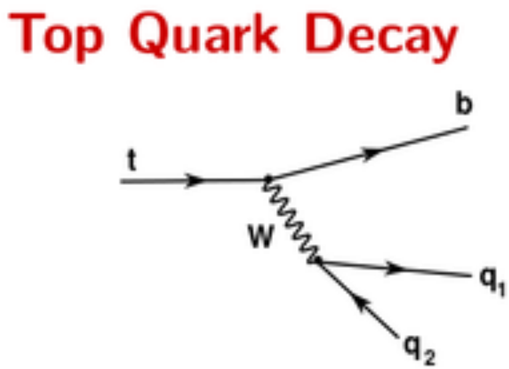


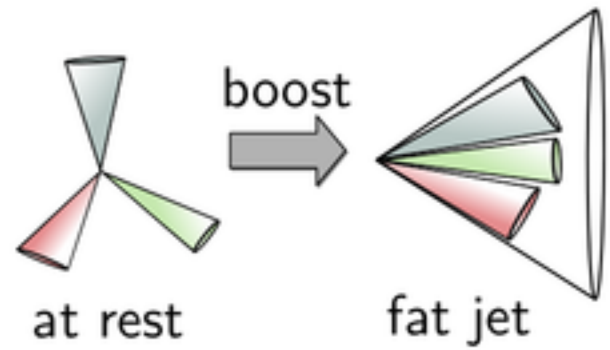


→ Generic DeepJet and custom quark vs. gluon DNN (2D convolutions) gave very similar results!

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

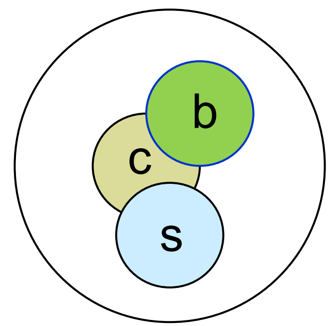
Fat jets





Key features of tops:

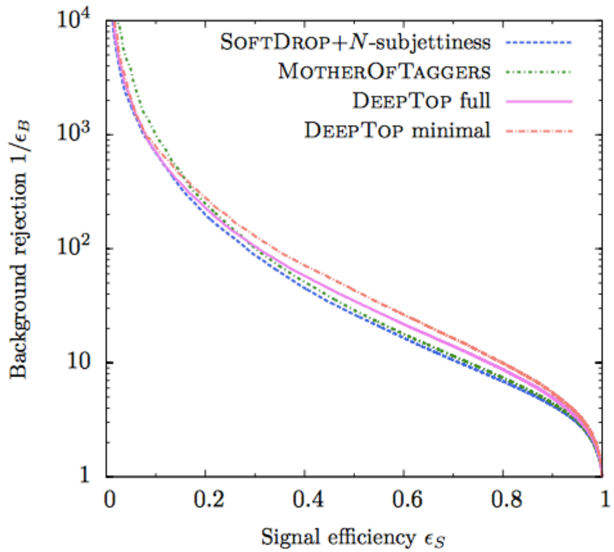
- Masses W, t, W polarization
- 3 "prong"
- b-subjet and 50% with c-subjet



Not obvious if these key features factorize or need to be addressed simultaneous.

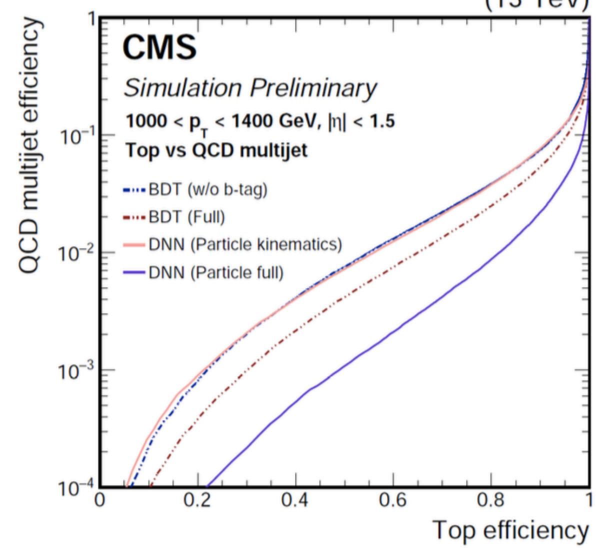
Large cone jets for boosted objects

1701.08784



- 2D convolution
- No flavor tagging

CMS-DP-17-049

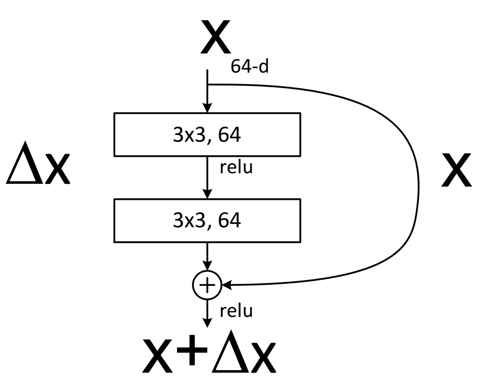


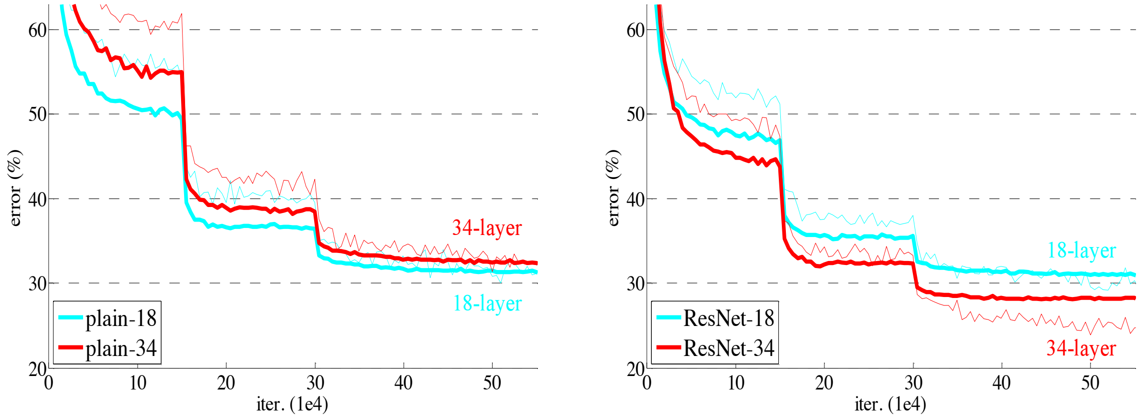
- DeepJet (using all particles + vertices) with and without flavor tagging

- Modest gain w.r.t. state of the art features + BDT
- Simultaneous flavor and structure tagging show improvements

Residual deep neural networks used in fat DeepJet

- 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



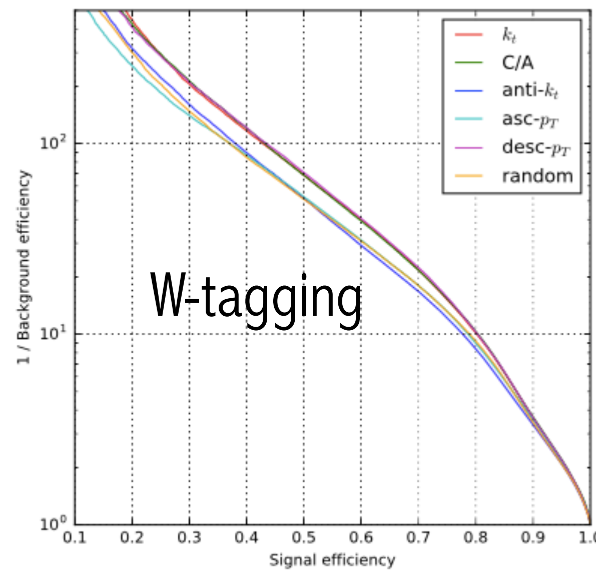


RESNETs useful for to make deep convolutional networks

1512.03385

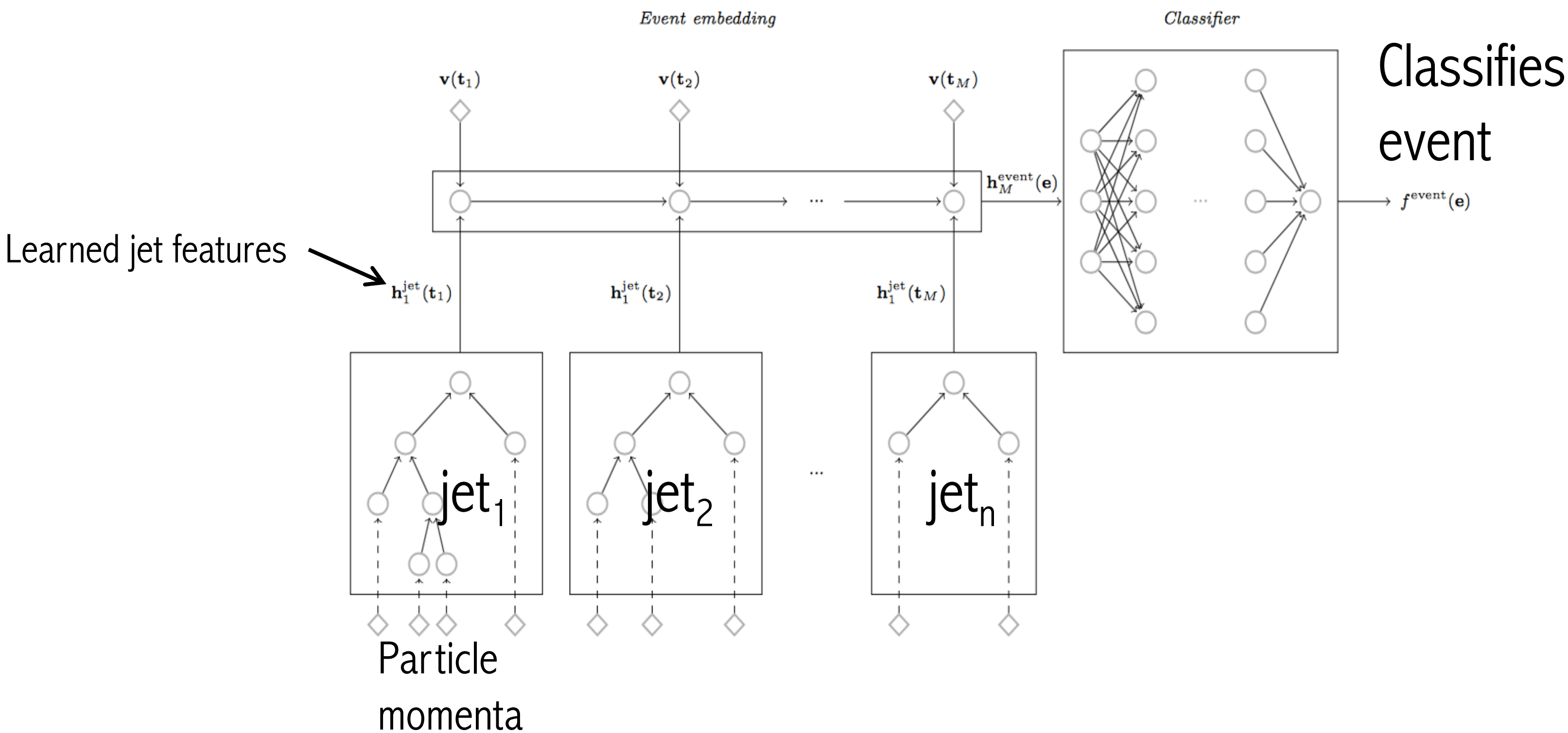
Recursive Neural Networks

- Use QCD inspires clustering to built a tree of jet particles
- Use this for recursive NN



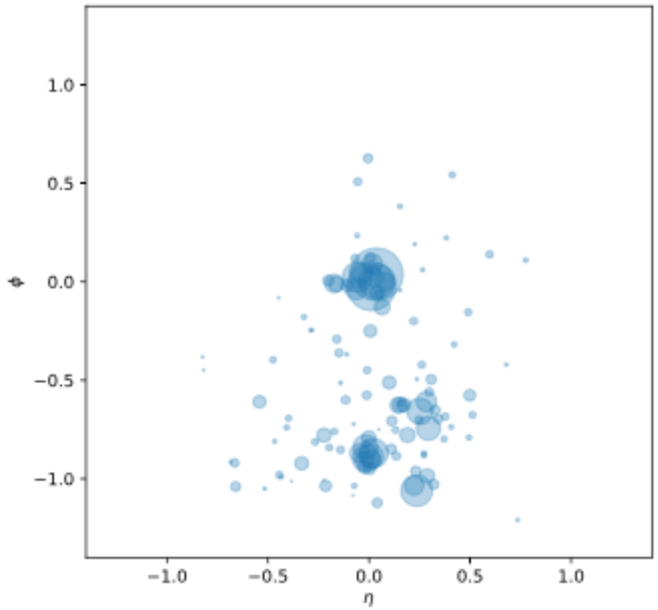
- Similar performance as simple p_T ordering RNN
- Potentially more stable w.r.t. uncertainties from theory

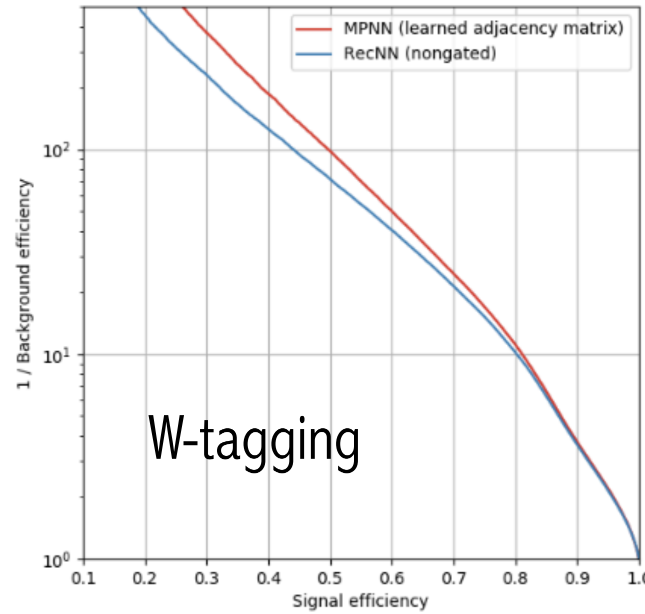
Recursive Neural Networks for event classification



Also shown as event classifier, i.e. merger steps of the traditional analysis chain

Message Passing Neural Networks





- Learn the adjacency matrix!

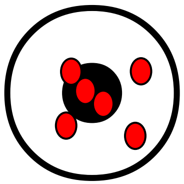
(see Kyle's talk)

- Some gain, e.g. 10% signal efficiency at 1% false positives
- Very data efficient!

https://dl4physicalsciences.github.io/files/nips_dlps_2017_29.pdf

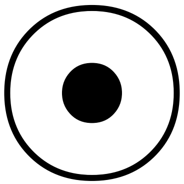
Definition of the target (loss)

Current target



Optimal performance in simulation

Desired target



Optimal and known performance in data

We teach ML to hit the wrong target

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

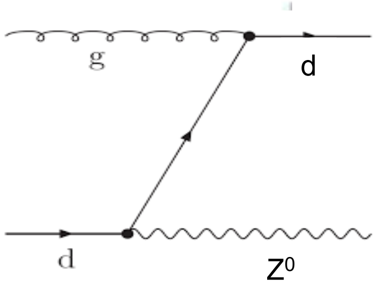
If you have several sets with know label proportions, this is enough for learning.

Just using sets with different label proportions

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

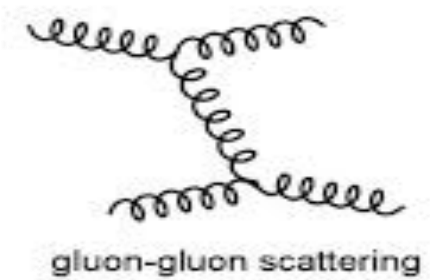
Indeed, it is sufficient to have different, but unknown label proportions

Z^0+jets:



many quark jets

Dijet:

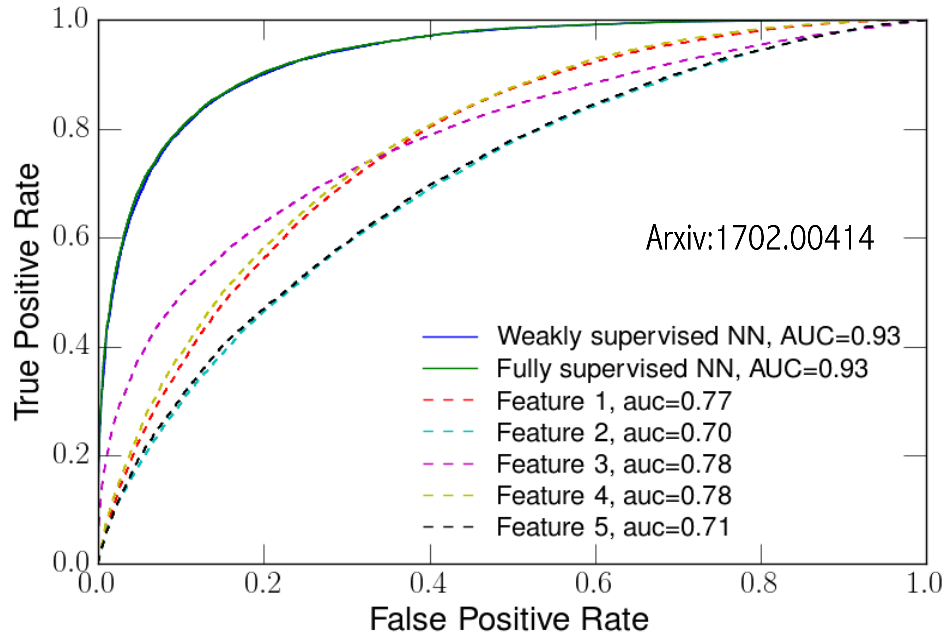


many gluon jets

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)

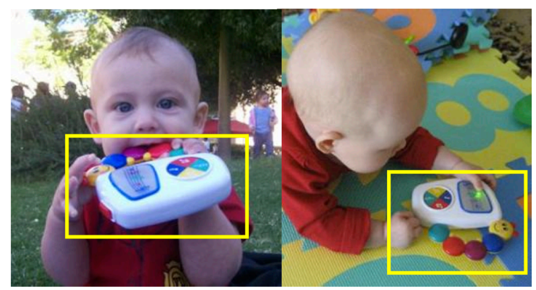
Target domain (real data)

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

digital SLR camera

low-cost camera, flash

amazon.com



consumer images

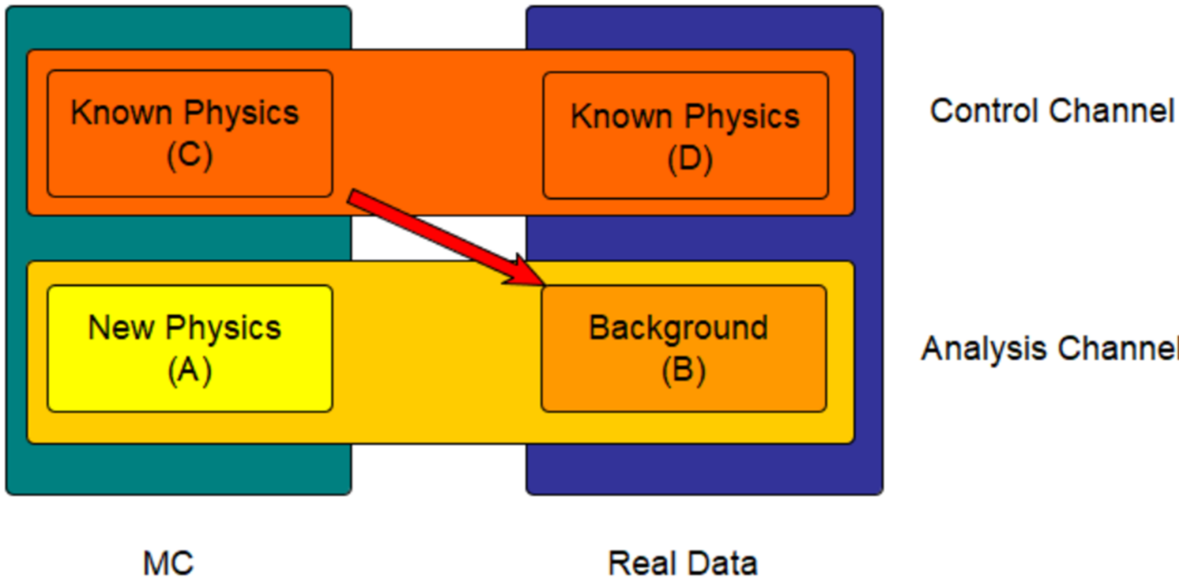
User samples to apply the training, **no labels** available

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

arXiv:1702.05464v1

LHCb example, $\tau \rightarrow \mu\mu\mu$ kaggle challenge

Bests NP discrimination with constraints:



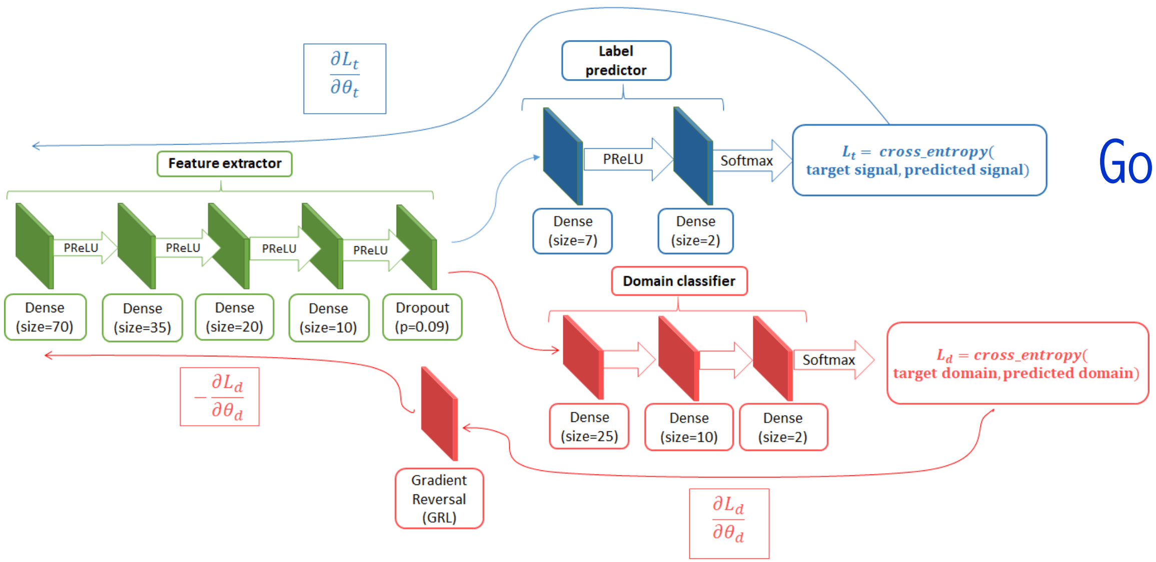
a) Discriminator should be same for real data and MC

b) Discriminator should be independent of mass

Background is data for computational reasons

1.https://www.kaggle.com/c/flavours-of-physics

Domain adaptation to get same data and simulation output for "known physics"



Good classification

Intermediate features same for data and MC

Metric / Model	Mass-aware Classifier	Data Doping	Domain-adaptation
AUC (truncated)	**0.999**	0.9744	0.979
KS (< 0.09)	0.18	0.087	0.06

Good classifier and small KS test between real data and MC outputs

https://indico.cern.ch/event/567550/contributions/2629724/attachments/1513629/2361286/Ryzhikov_poster_v6.pdf#search=Andrey%20Ustyuzhanin%20AND%20EventID%3A567550

Conclusion

- Plenty simulated labeled data for supervised learning available
- Headroom difficult to estimate
- Flavor tagging showed improvements
- First advanced DNNs (DeepJet) implemented in CMS for reconstruction
- Validation in real data are ongoing
- Still many parts in the reconstruction/data analysis chain that can be improved (not only tagging & jet energy regression)
- Use increasingly real data for training and not only the validation process