Practical ML Examples in HEP*

* HEP = biased towards my interested, collider experiments and jets…

Karol Krizka

September 22, 2021

BERKELEY LAB
Lawrence Berkeley National Laboratory

Particle Physics

Glossary

Tagger: Classifier used to identify an object.

ie: b-tagger identifies a jet as containing a b-hadron or not

Fake: Object of class *x* tagged as class *y*

Signal: What you are looking for

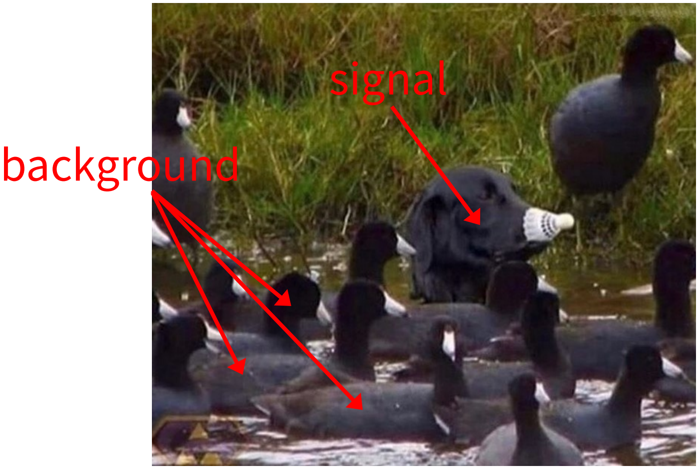
aka the needle

Background: What you want to reduce

aka the haystack

irreducible = cannot be reduced because it looks exactly like the signal

Search for the dogduck



Machine Learning Quick Start

BDT or NN or ...

trained **parameters**

$$y = f(\vec{x}; \vec{\theta})$$

output
ie: likelihood event
contains h→bb

input
ie: jets in an event

Uses of Machine Learning in HEP

Classification (Supervised)

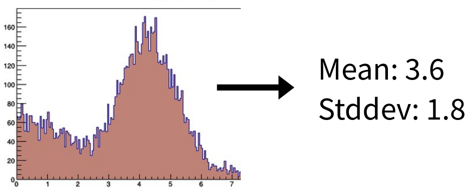
does this event contain a Higgs boson or QCD background?

Classification (Unsupervised)

does this event look like new physics?

Regression

value of a funtion without a (simple) analytical form



Event Generation

generate new events without the need for complicated simulation

photon in nature

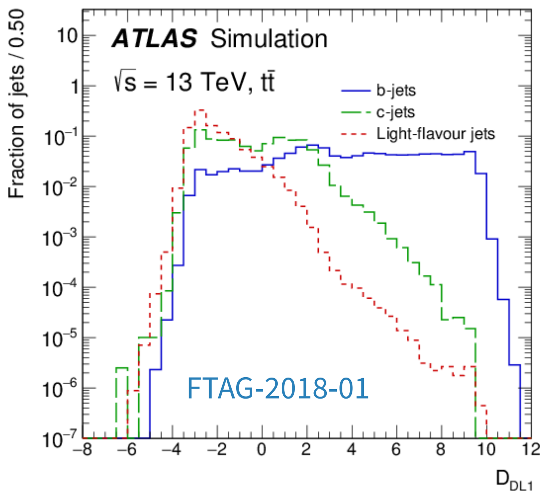
photon in calorimeter

and constantly growing...

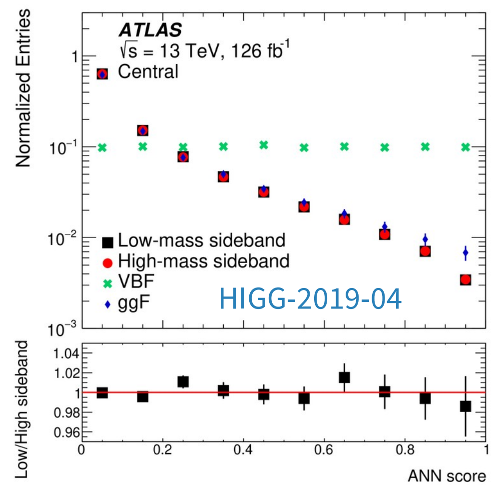
Object/Event Classification

The classical use of ML in High Energy Physics…

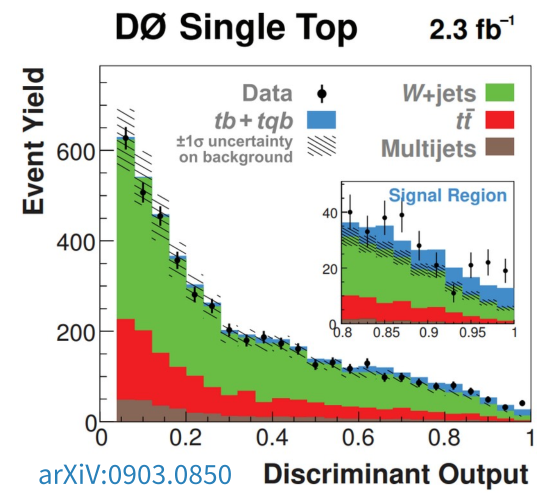
NN on reconstructed secondary vertices to identify b-jets



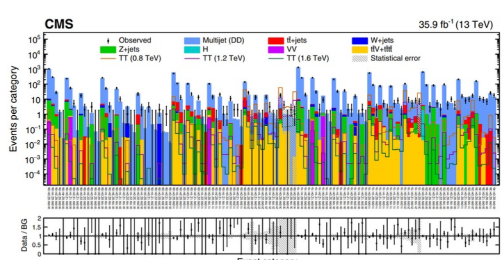
ANN for identifying VBF h→bb events from QCD



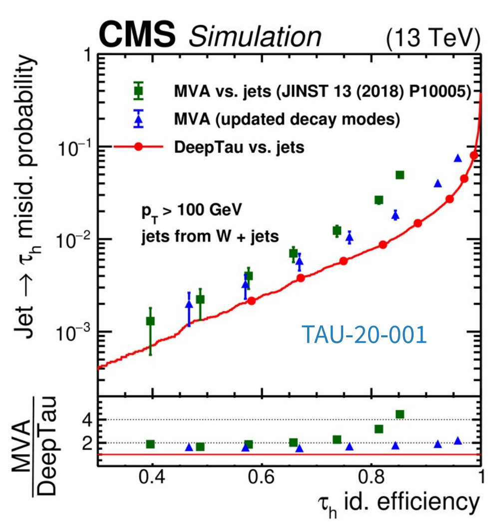
BDT used for observation of single top quark production



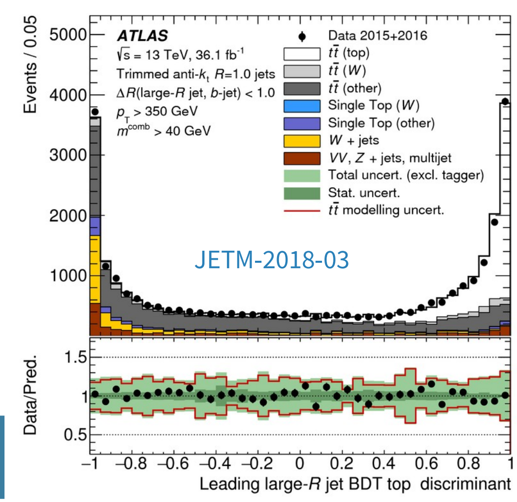
NN classifies jets as W, Z, H, t, b, light and categories in event are counted for vector-like quarks search B2G-18-005



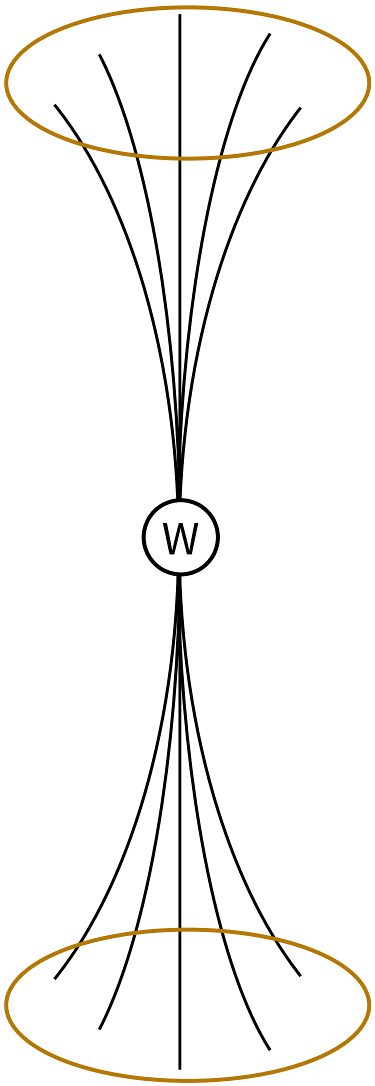
NN for identifying taus



BDT for identifying W vs top jets

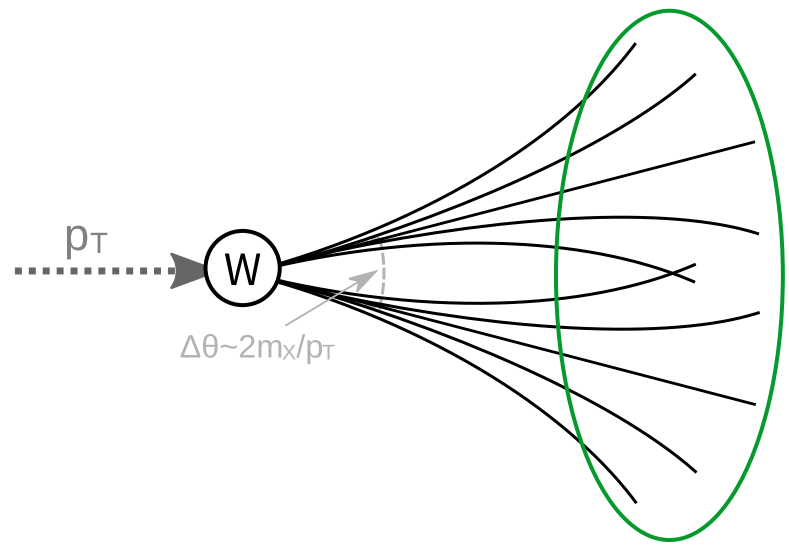


What are boosted objects?



A hadronically decaying particle *W* at rest can be reconstructed using two anti-K_T R=0.4 jets.

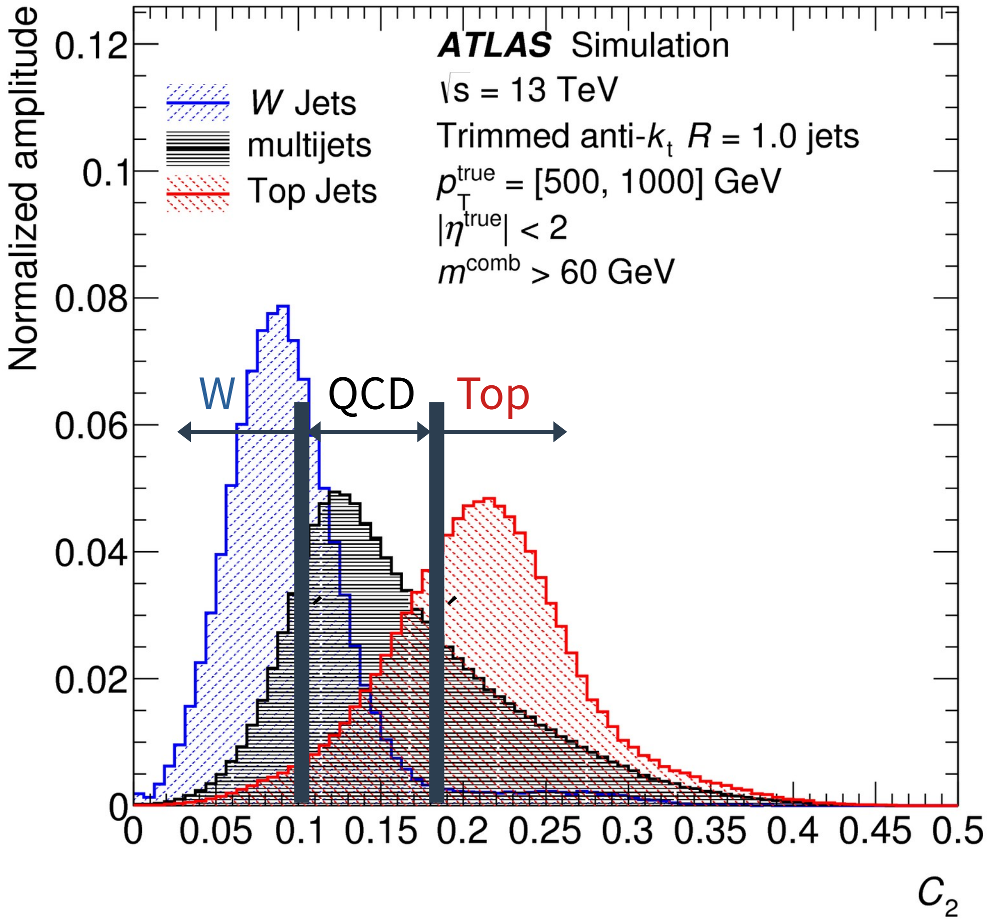
But if *W* is boosted, then anti-k_T R=0.4 will not be able to resolve two separate jets.



Solution: reconstruct a single large-R jet and look at the radiation pattern of the constituents (substructure).

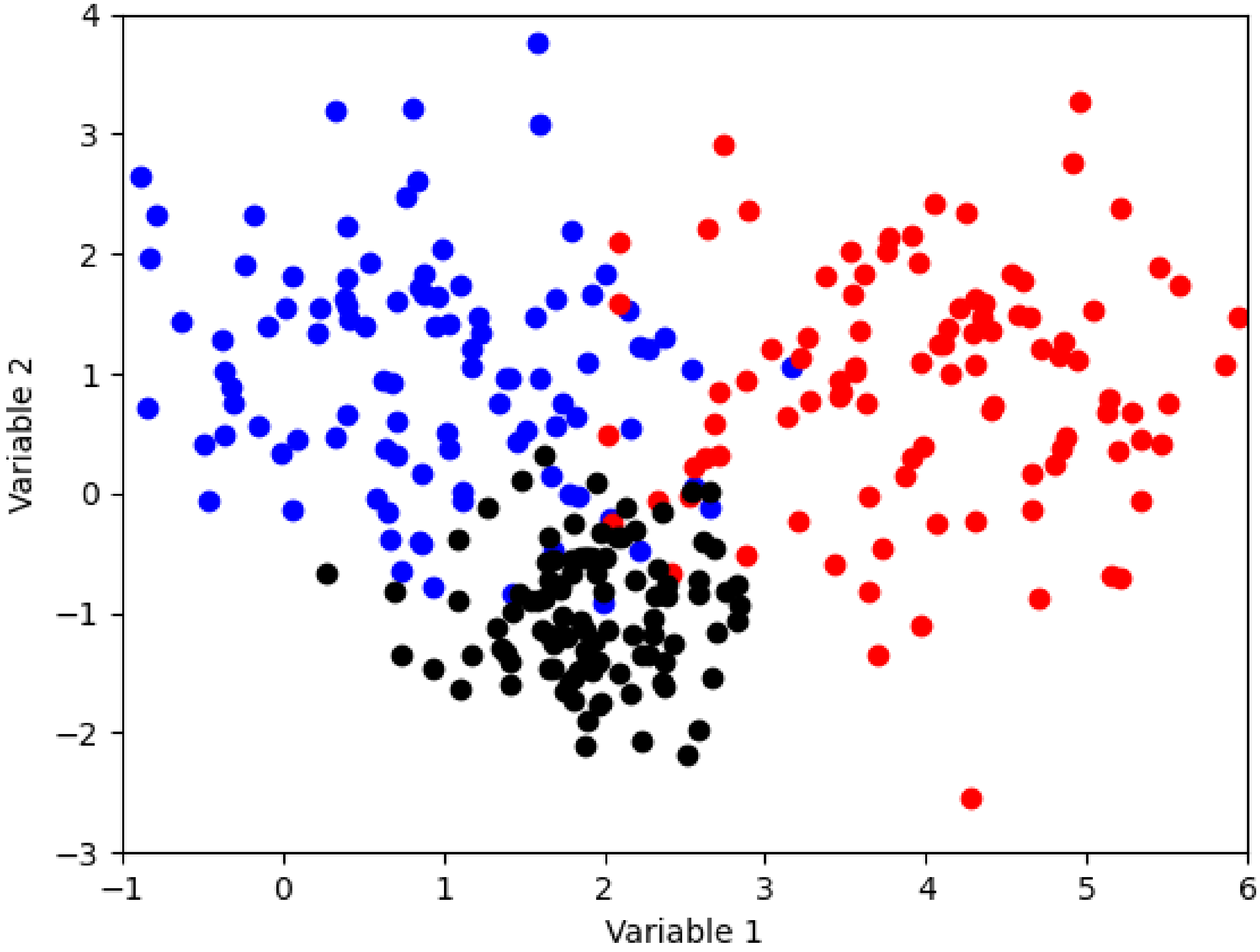
- Invariant mass of constituents?
- How many hard prongs?
- How many b-tagged track jets?

Some Jet Moments

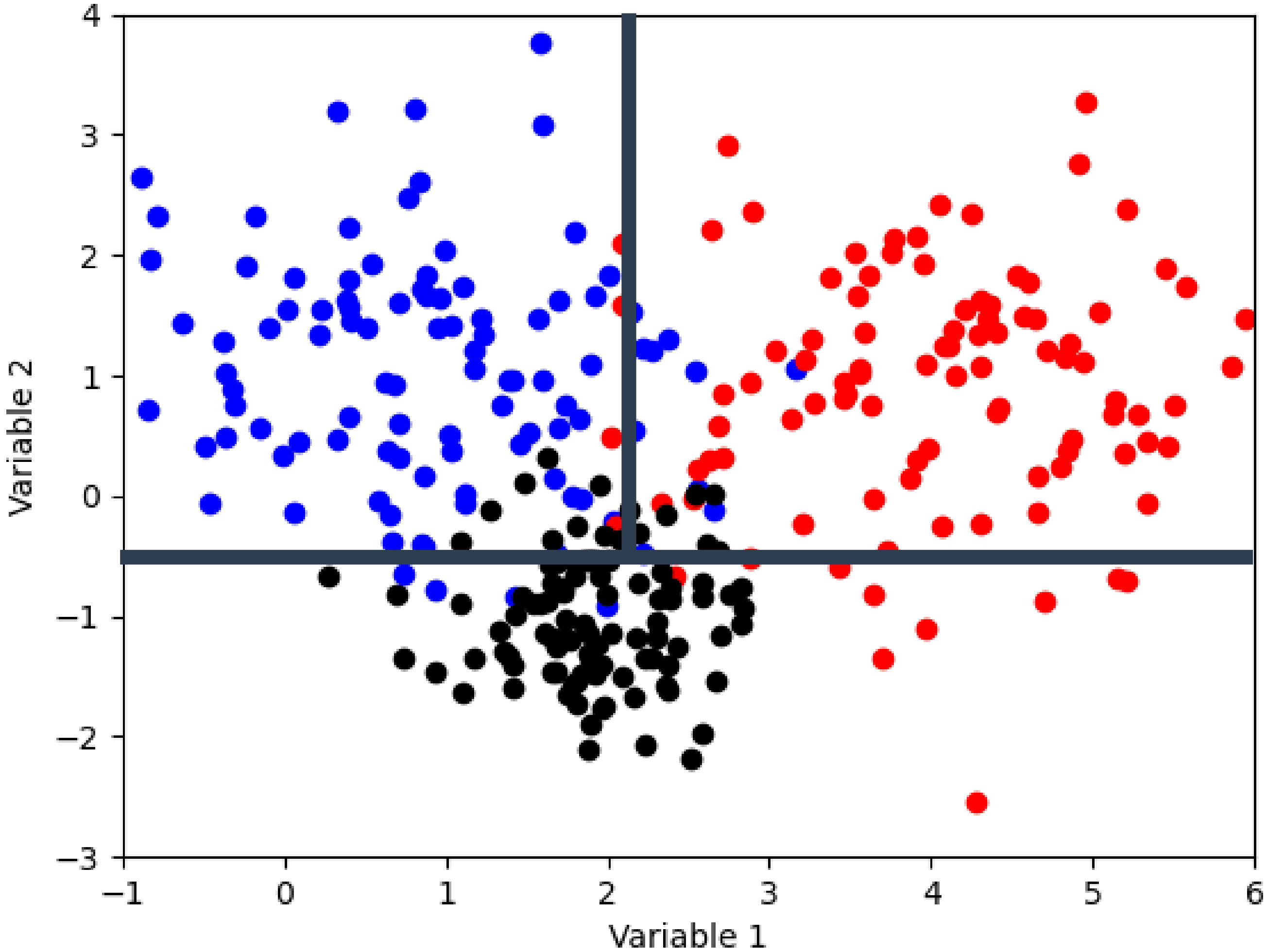


Observable	Variable	Used for
Calibrated jet kinematics	p_T, m^{comb}	top, W
Energy correlation ratios	e_3, C_2, D_2	top, W
N-subjettiness	$\tau_1, \tau_2, \tau_{21}$	top, W
	τ_3, τ_{32}	top
Fox–Wolfram moment	R_2^{FW}	W
Splitting measures	z_{cut}	W
	$\sqrt{d_{12}}$	top, W
	$\sqrt{d_{23}}$	top
Planar flow	$\mathcal{P}$	W
Angularity	a_3	W
Aplanarity	A	W
KtDR	$KtDR$	W
Qw	Q_w	top

Multivariate Classification With ML

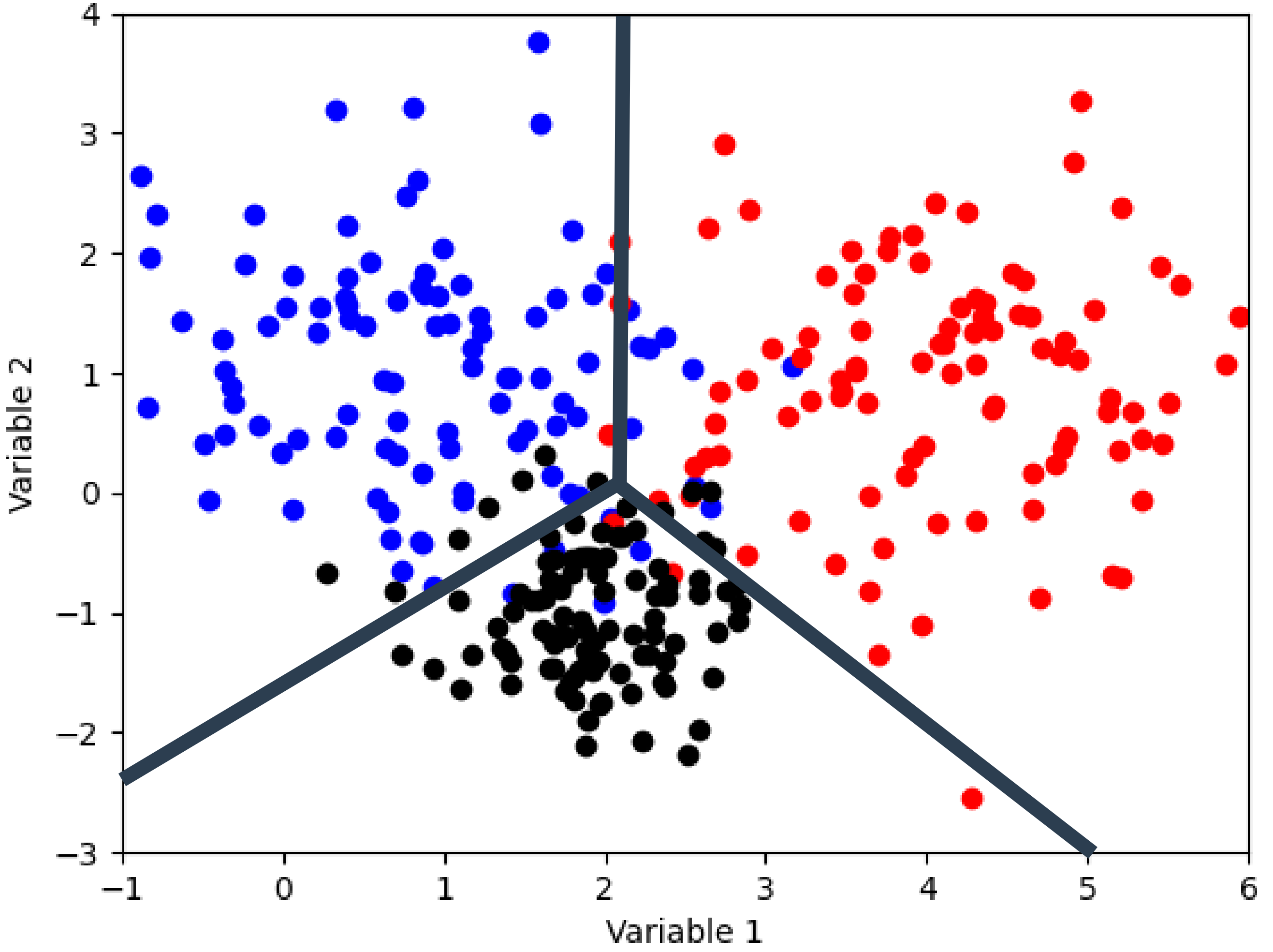


"Cut-Based" Classification



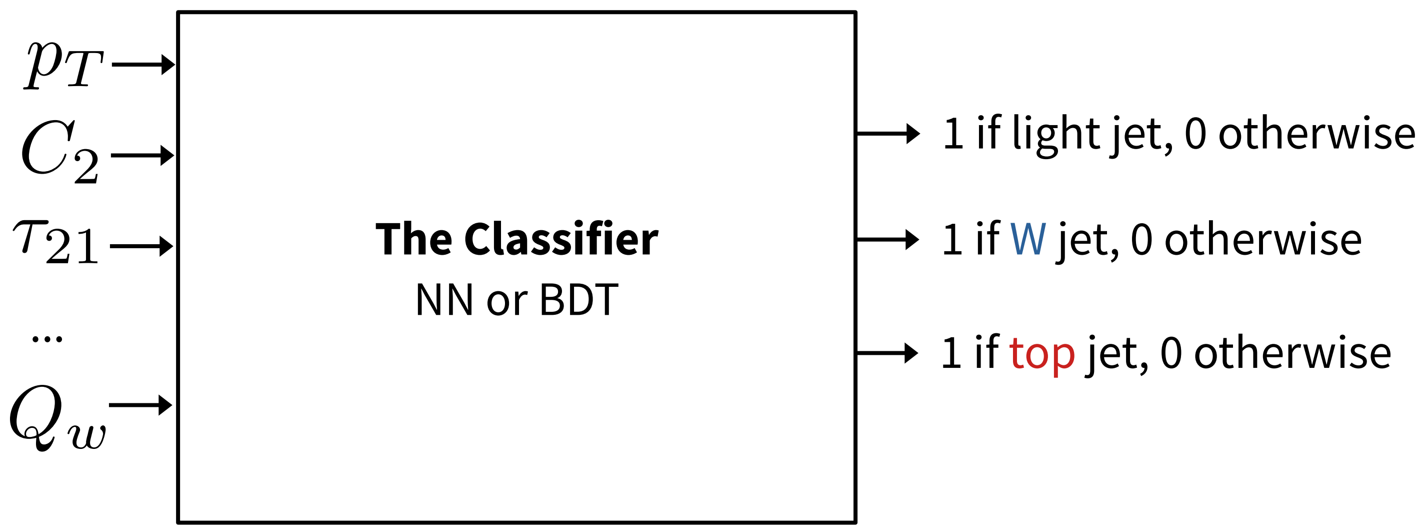
Multivariate Classification

Neural network or BDT also allows to draw curly boundaries!

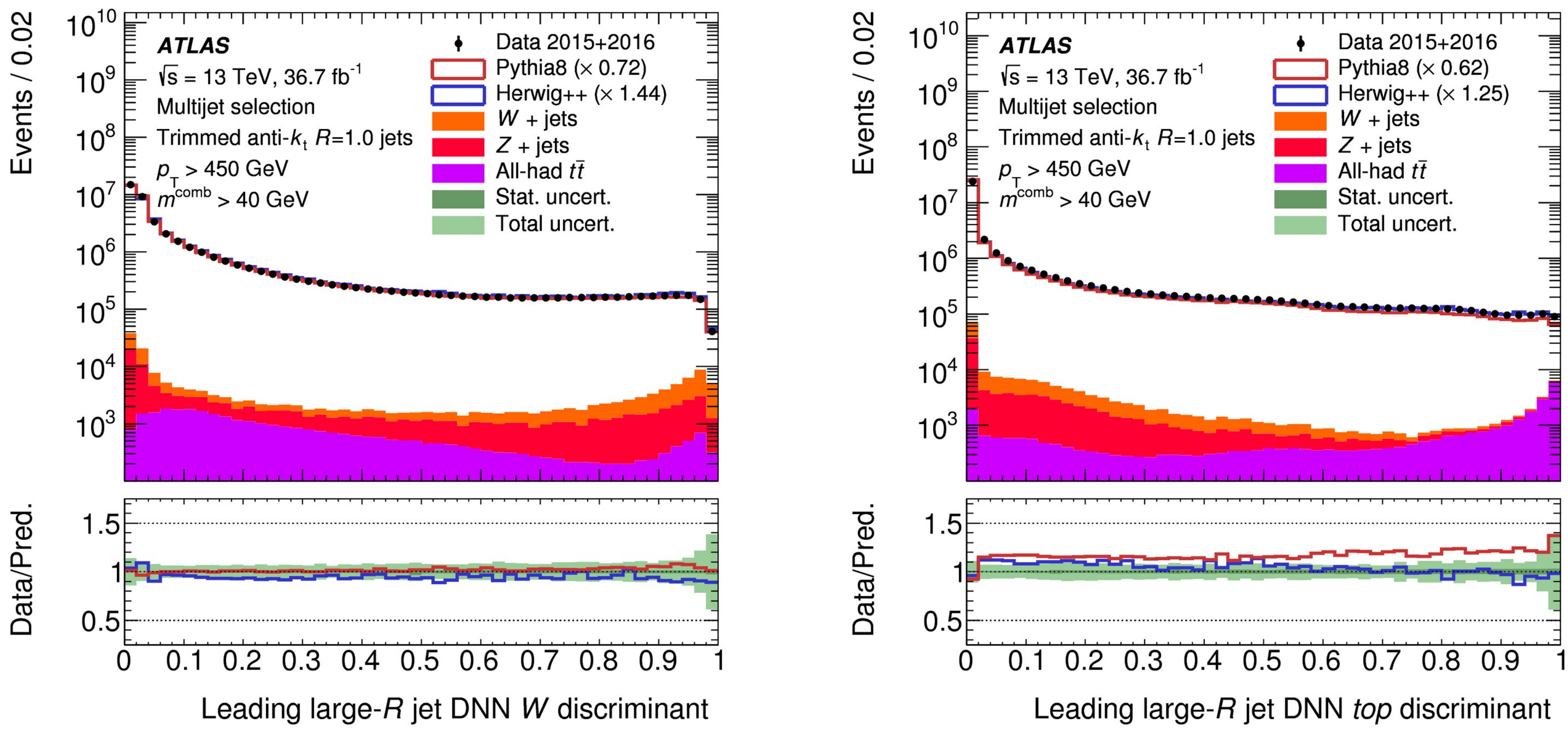


Architecture

"one hot encoding"
truth label of W jet = (0, 1, 0)



Results of ATLAS Classifier



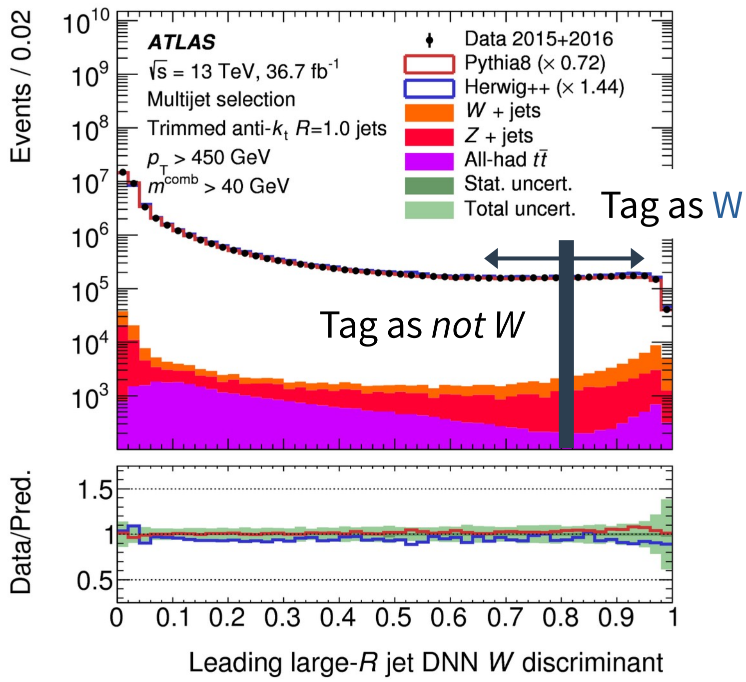
How to Choose Class?

In most ML tutorials, you will see:

$$\text{class} = \text{argmax}\{[p_W, p_{QCD}, P_t]\}$$

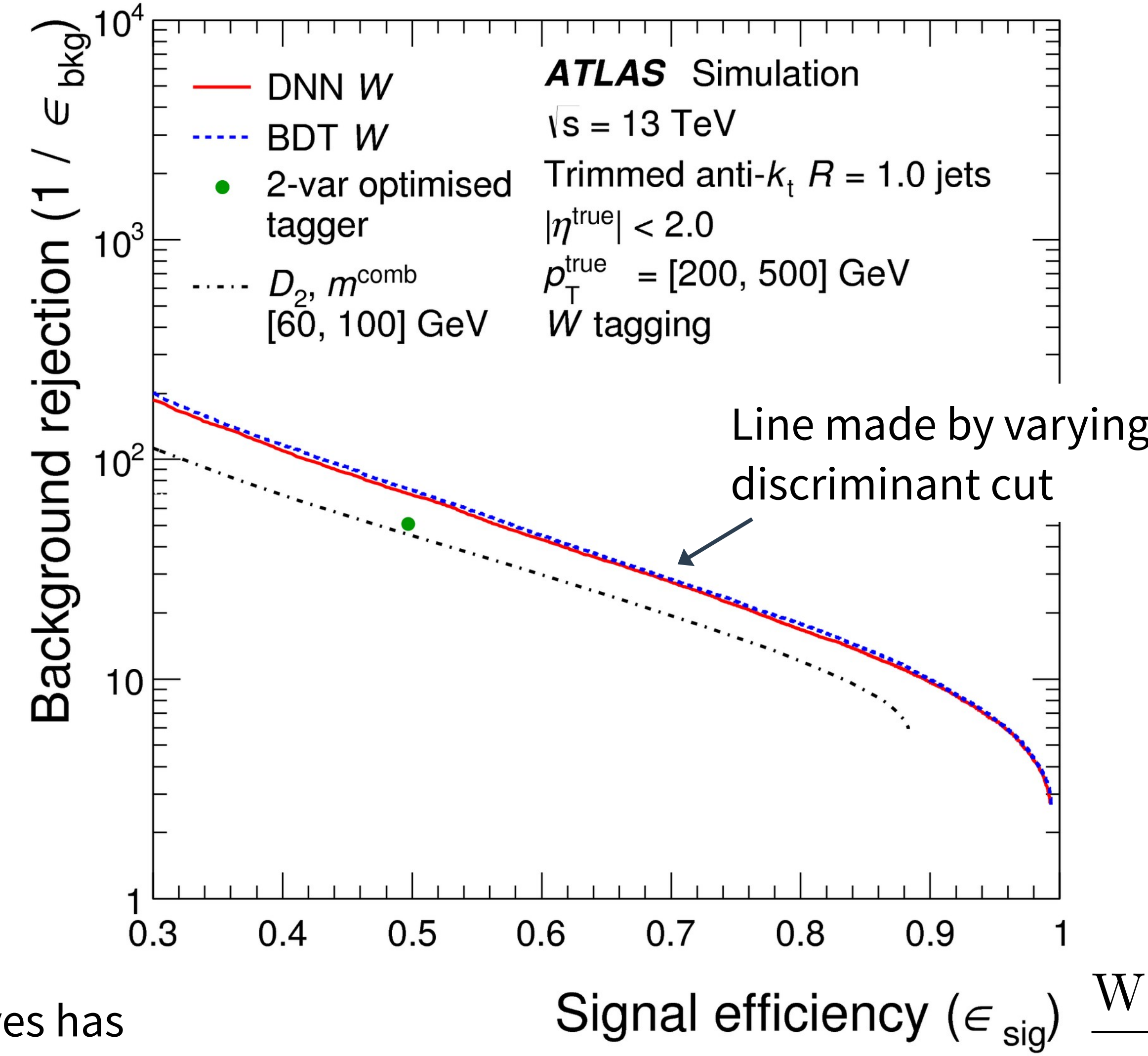
In HEP, we often cut on discriminator variable:

- Tight cut → very pure sample
  - Low on statistics
  - Used for measurements
- Loose cut → many signal events
  - Good for searches
- Chosen value is part of analysis optimization
  - Sometimes only few *calibrated* cuts allowed



Receiver Operating Characteristic (ROC) Curve

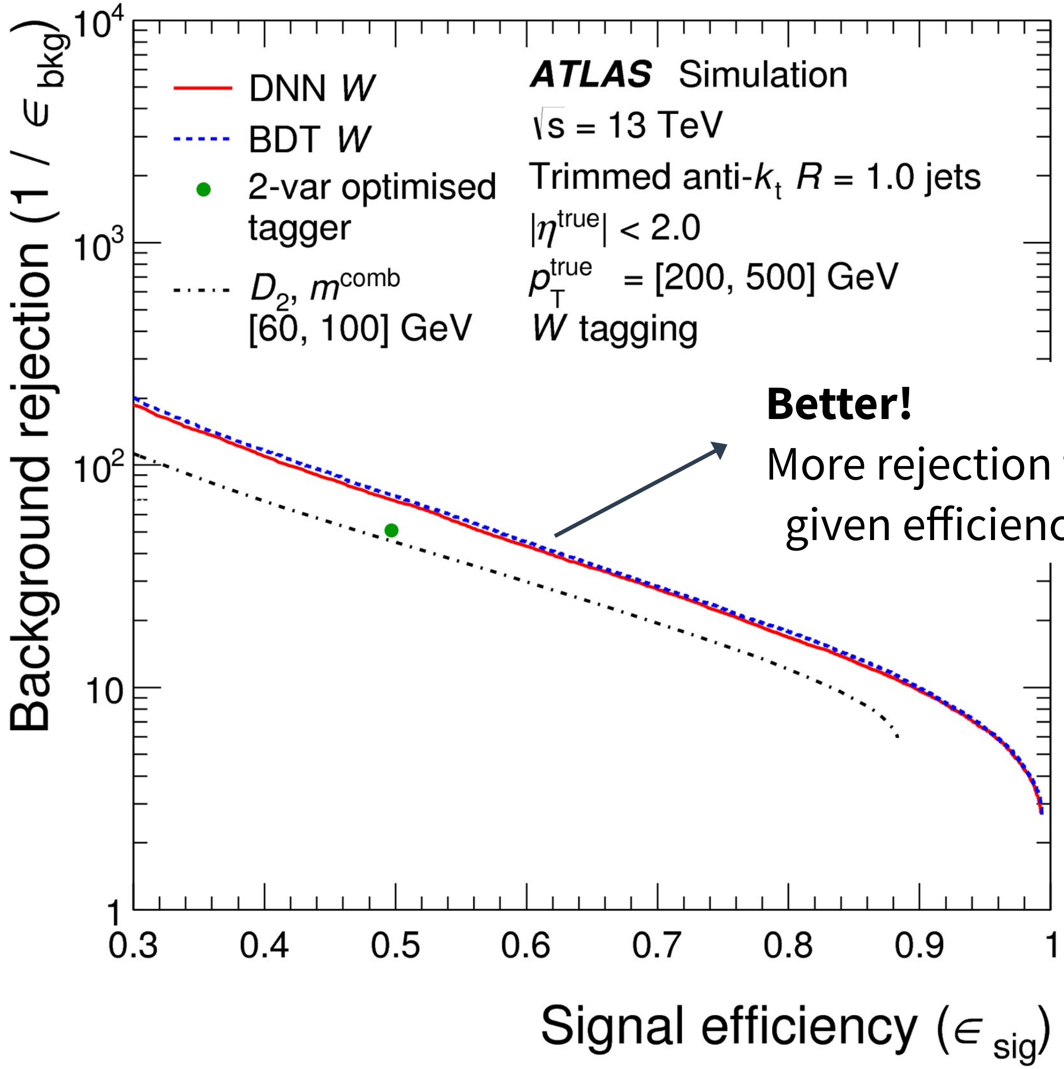
$$\frac{\text{all background}}{\text{bkg passing cut}}$$



$$\frac{\text{W's passing cut}}{\text{all W's}}$$

Q: Which of the curves has the best performance?

Receiver Operating Characteristic (ROC) Curve



Better!
More rejection for given efficiency

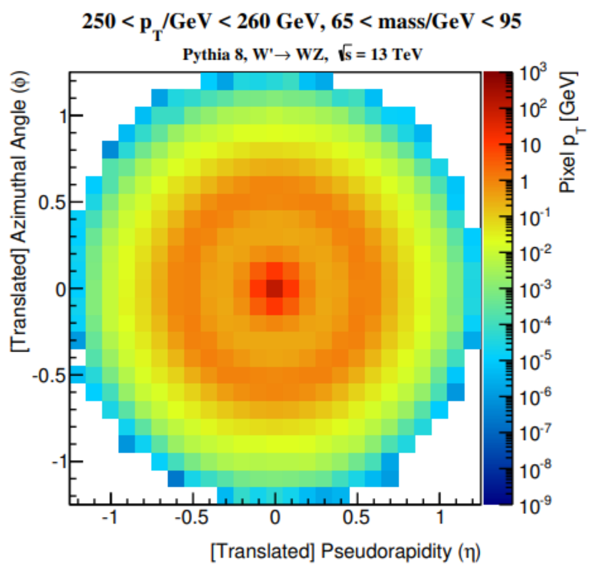
Jet Images

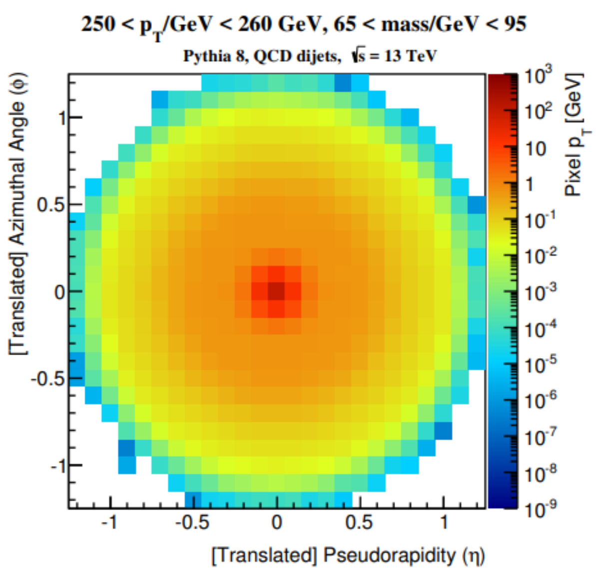
1511.05190

Calorimeter ≈ an image

- x,y = φ, η
- color = energy deposition

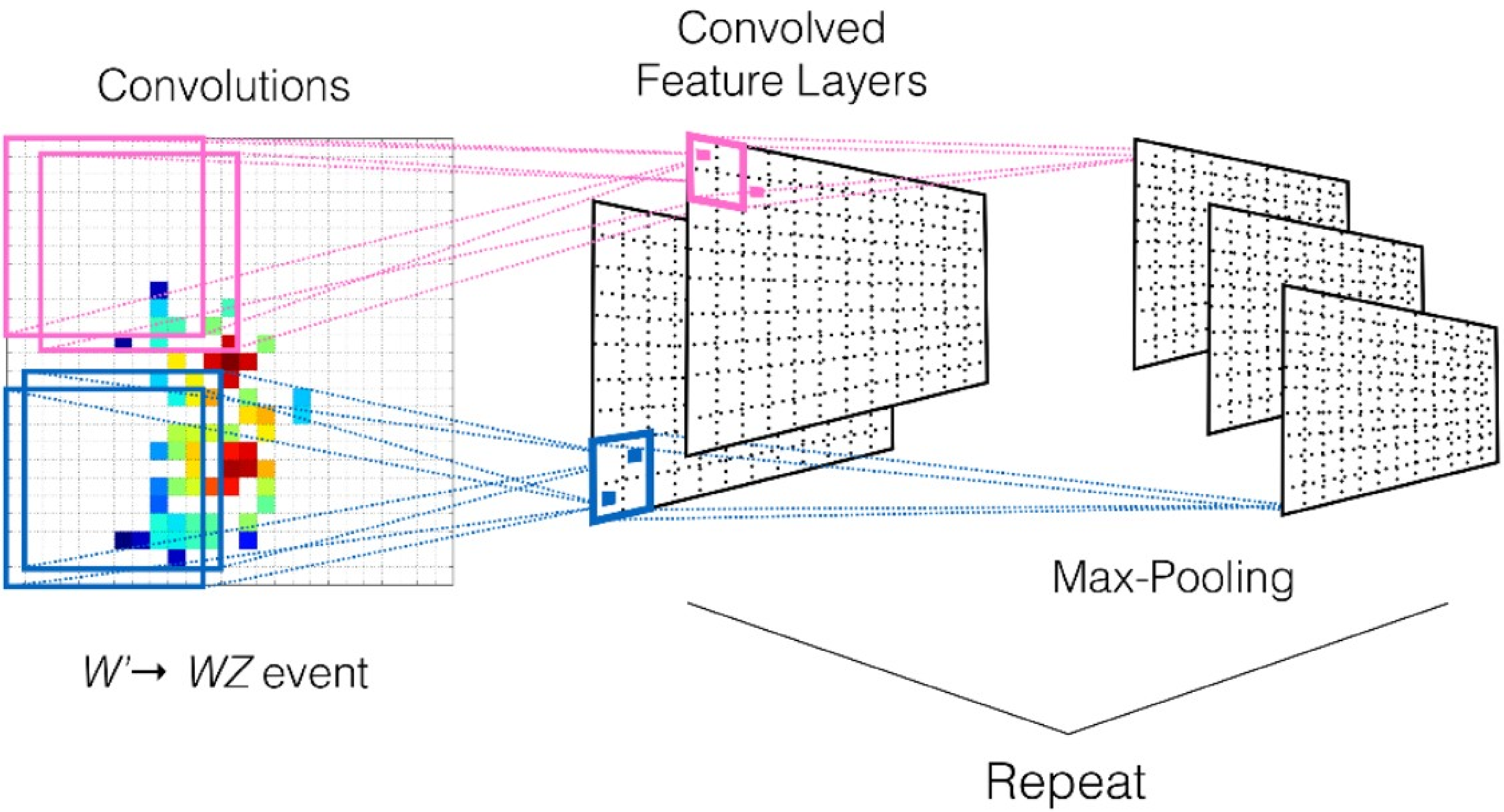
- **Image classification is a key part of ML in industry**
- **Can a NN learn the calculation of "jet moments" from images?**
  - And can it calculate jet moments we didn't think of?



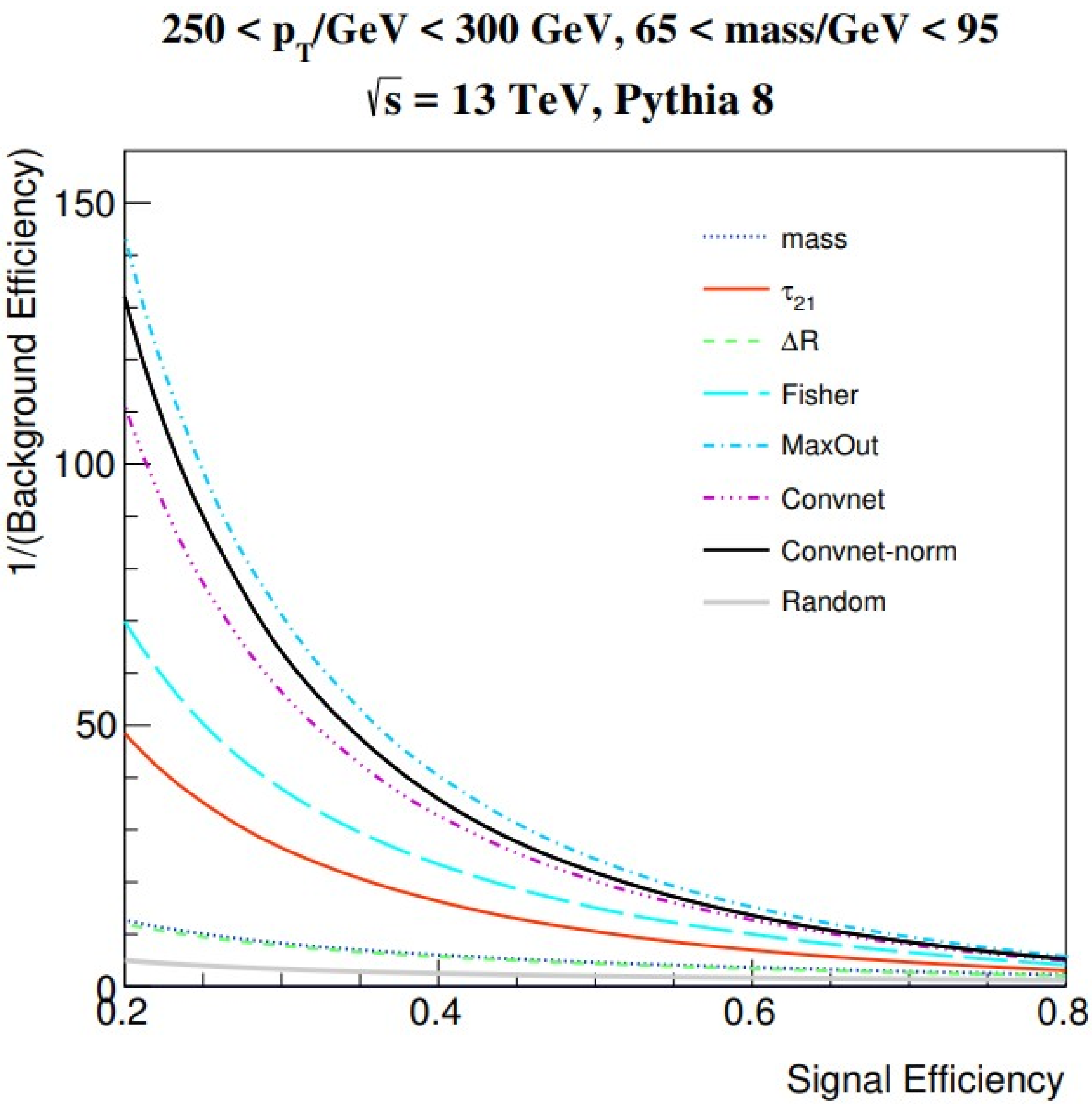


Convolutional Neural Networks

CNN's are a great architecture for processing images



Result of Jet Images



Calibrating a Classifier

- **Classifier is a black box… no idea what it is learning.**
- **Training is (mostly) done on simulated events.**
  - What if our simulation is wrong? NN can be sensitive to random corners…
- **Calibration: ensure same efficiency in MC as in data**

1) Run classifier on data that is pure in signal via selection

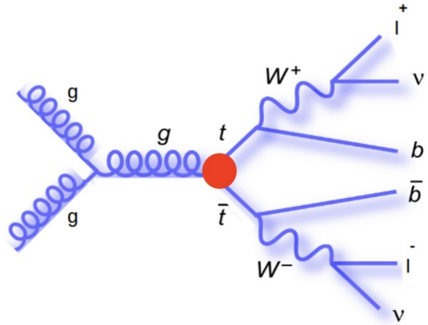
2) Run classifier on MC (pure in signal)

3) Scale Factor = correction to MC = $\epsilon_{data}/\epsilon_{MC}$

4) Repeat on background sample second SF

THIS IS THE HARDEST PART TO PUT A NEW CLASSIFIER INTO PRODUCTION!!!

Example of pure b-jet sample



Any jets in a dileptonic ttbar event will be b-jets

Important Classifier Figures

- Thoughts?

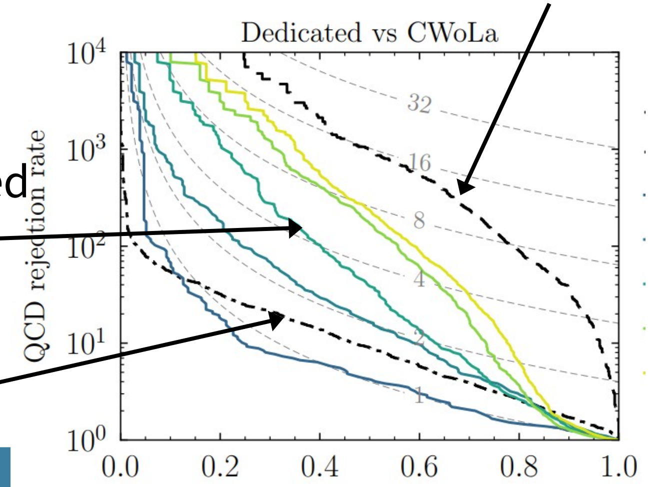
Important Classifier Figures

- **Plot your input variables**
  - Teaches you what is important. Even better if you include correlations
- **ROC curves are great way to compare classifiers**
  - Great overview of the performance. Not just a single point.
- **Always include a simple cut-based for comparison**
  - If your NN does not outperform cut-based, then don’t use it (Keep It Simple).

Anomaly Searches

Given collision events, find ones that you can't exaplain.

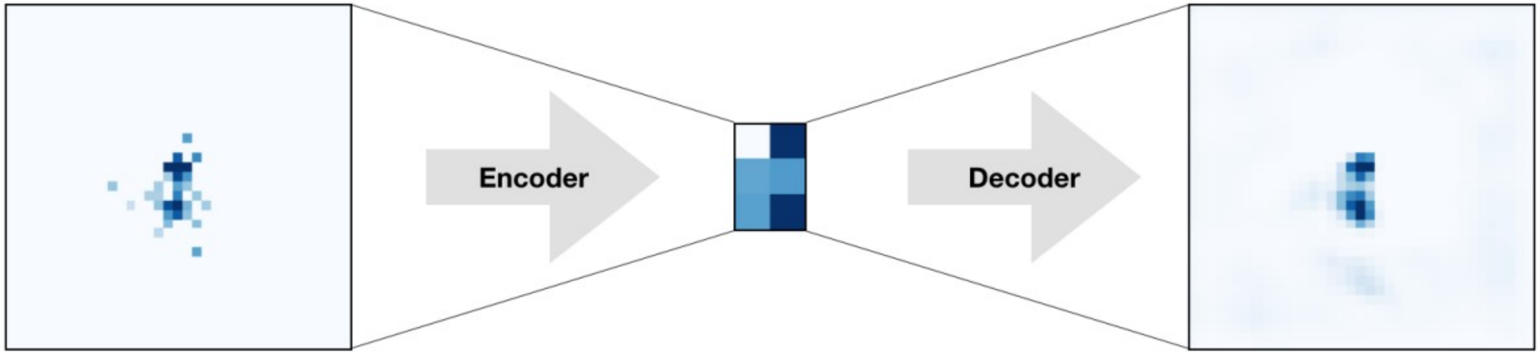
- **Supervised: train on background and signal**
  - Most powerful, but model dependent. *Not anomaly search*.
- **Weakly Supervised: train with imperfectly labeled data**
  - Hard to tell if "anomaly" is new physics *or bad modeling*
- **Unsupervised: no simulation as input to training**
  - The "holy grail" of ML in collider experiments



arXiv:1902.02634

Auto Encoders

Learn to encode and decode your data to a representation



If trained on class of *x*

$$x \approx d(e(x, \theta_e), \theta_d))$$

If ***not*** trained on class of *x*

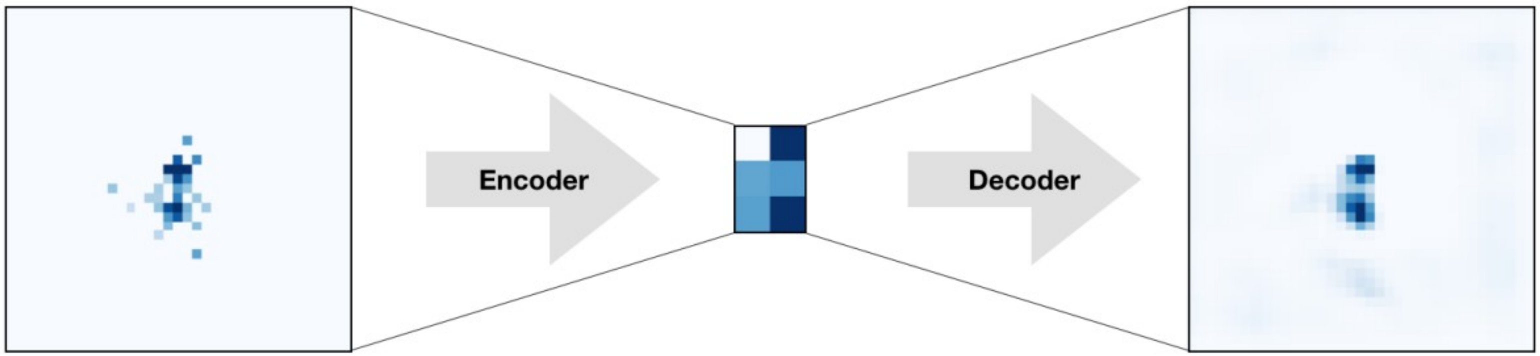
$$x \not\approx d(e(x, \theta_e), \theta_d))$$

Important figure of merit (loss function)

????? question

Auto Encoders

Learn to encode and decode your data to a representation



If trained on class of *x*

$$x \approx d(e(x, \theta_e), \theta_d))$$

If ***not*** trained on class of *x*

$$x \not\approx d(e(x, \theta_e), \theta_d))$$

Important figure of merit (loss function)

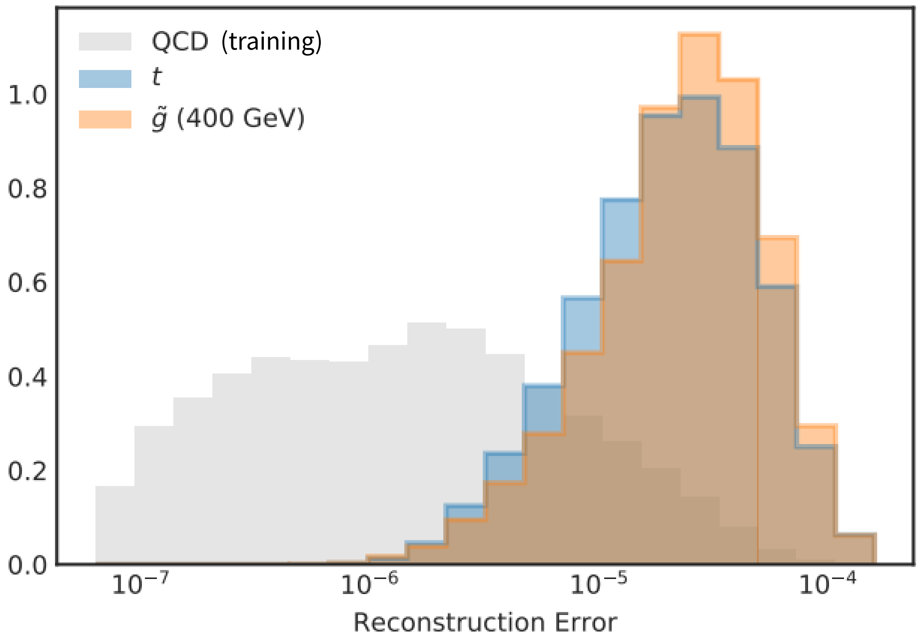
$$L = |x - d(e(x, \theta_e), \theta_d)|$$

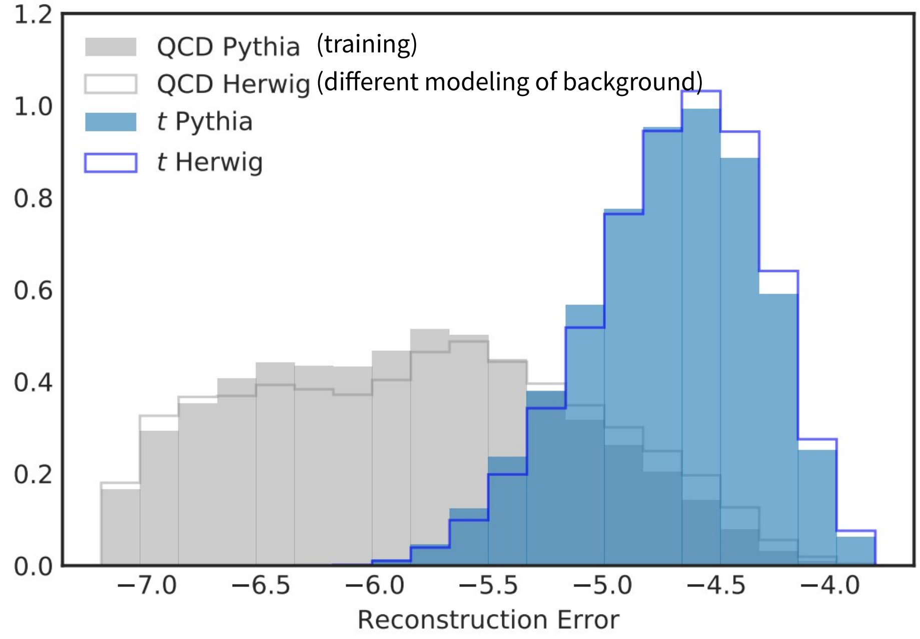
arXiv:1808.08992

Searching for New Physics with Deep Autoencoders*

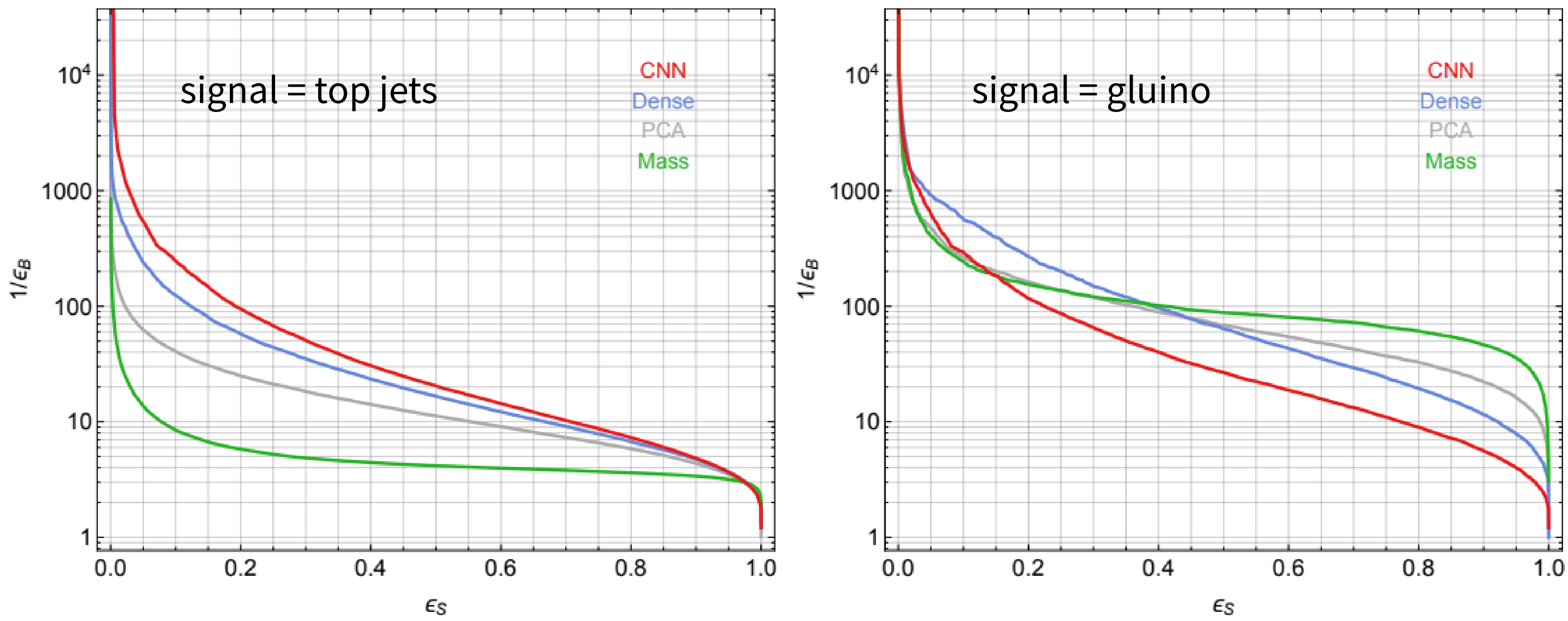
* Marco Farina, Yuichiro Nakai, David Shih

- **Proposed for a (weakly) supervised jet substructure search**
- ***Input* to the autoencoder is a *jet image***
- **Three architectures tried**
  - Principal Component Analysis
  - Deep Neural Network
  - Convolutional Neural Network
- **Two approaches to training**
  - Weakly supervised: train on simulated background
  - Unsupervised: train on (simulated) background with signal injected

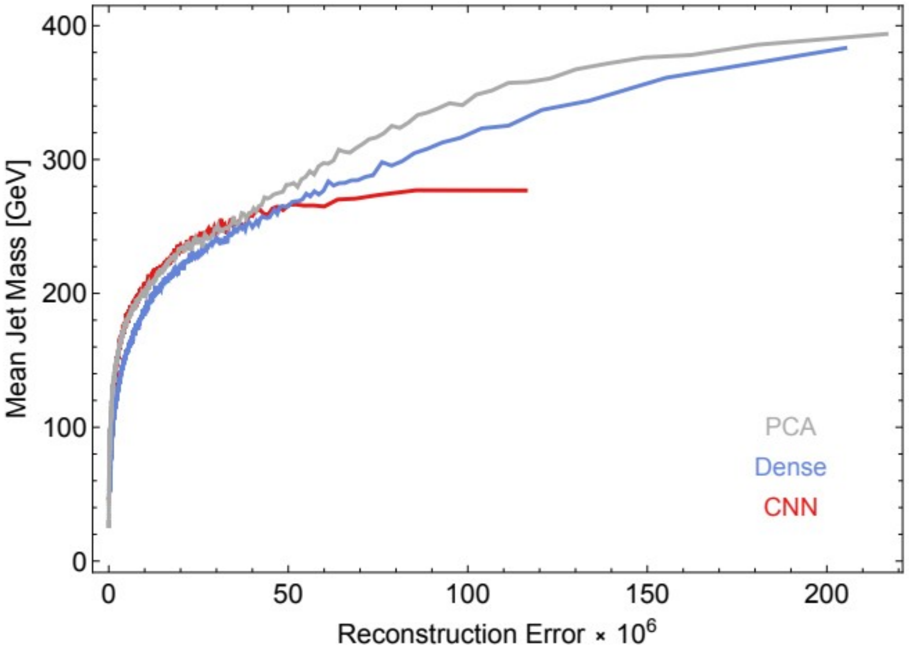

QCD (training)
t
$\tilde{g}$ (400 GeV)
1.0
0.8
0.6
0.4
0.2
0.0
10^{-7}
10^{-6}
10^{-5}
10^{-4}
Reconstruction Error

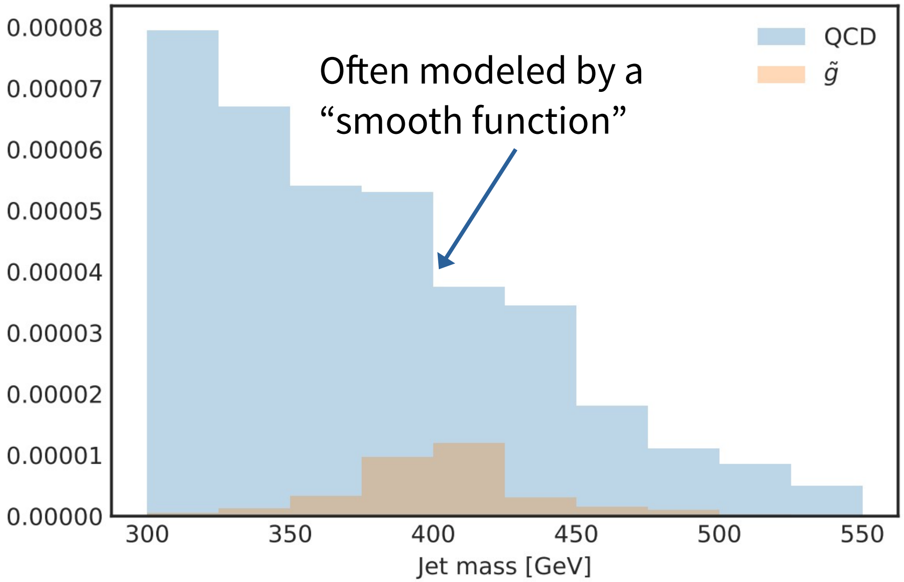

QCD Pythia (training)
QCD Herwig (different modeling of background)
t Pythia
t Herwig
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−7.0
−6.5
−6.0
−5.5
−5.0
−4.5
−4.0
Reconstruction Error

Performance



Be Careful of Black Boxes





Typical analysis flow:

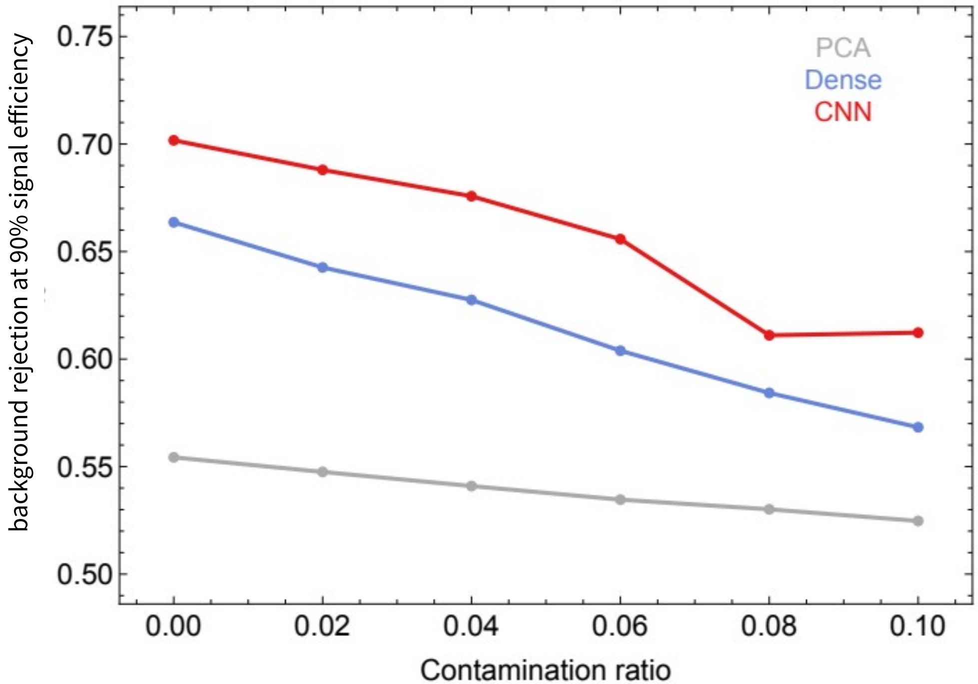
1) Reduce background (ie: classifier)
2) Estimate remaining background

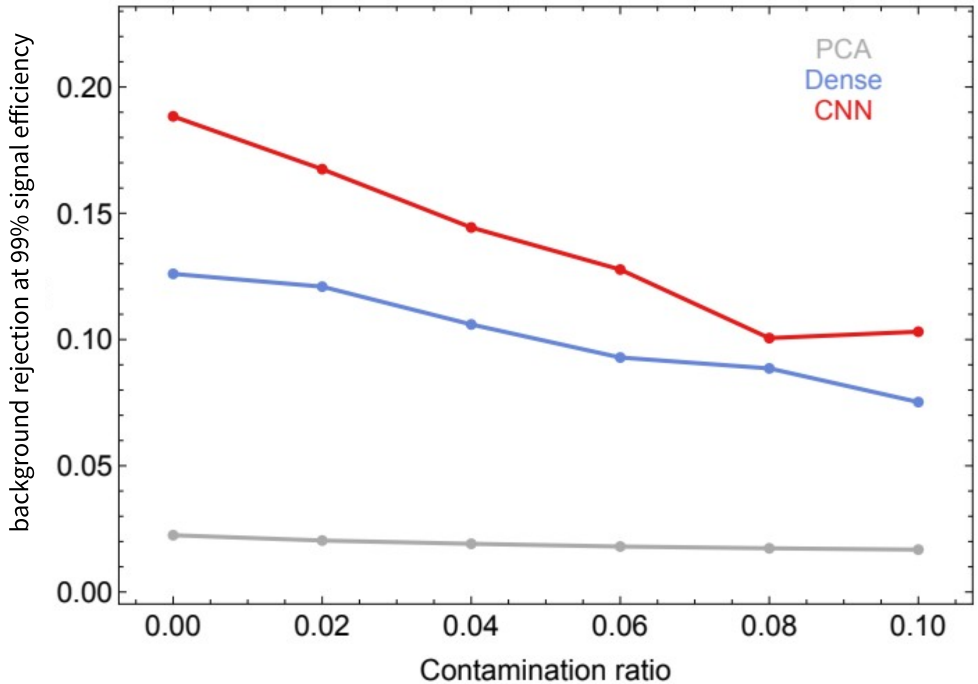
You need to make sure that 1) does not make 2) harder.

ie: do not sculpting of background!

Unsupervised Method

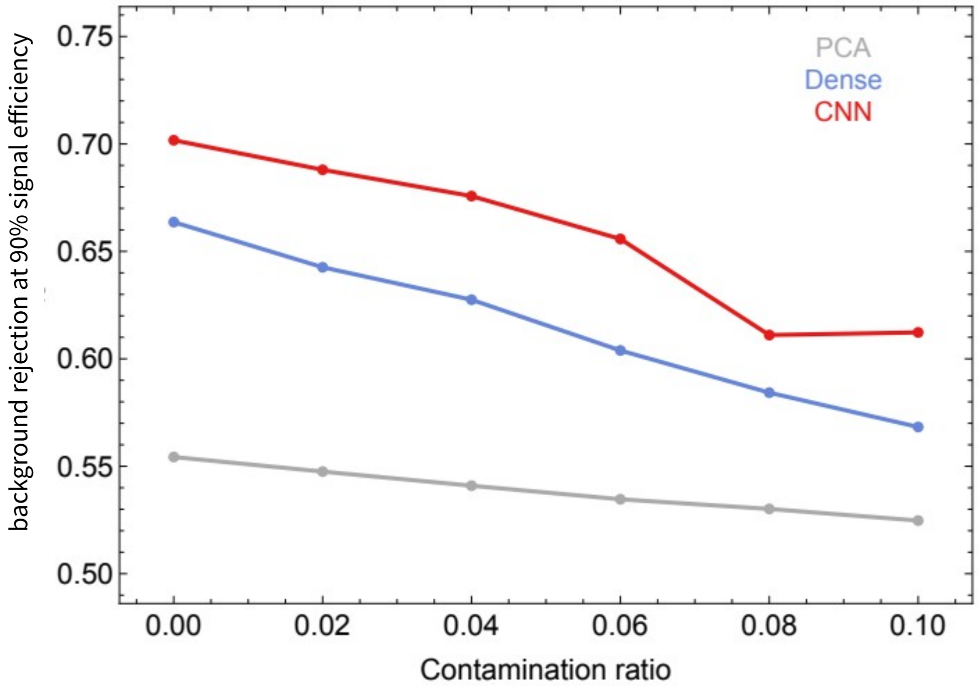
Training dataset is contaminated with top sample

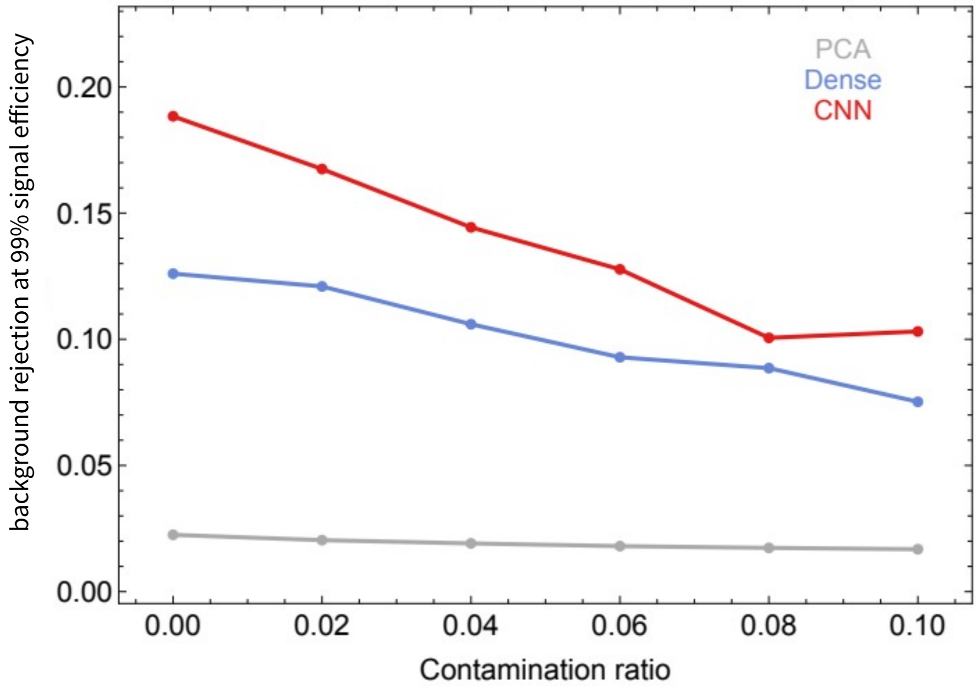




Unsupervised Method

Training dataset is contaminated with top sample





Regression

Estimate quantity given properties.

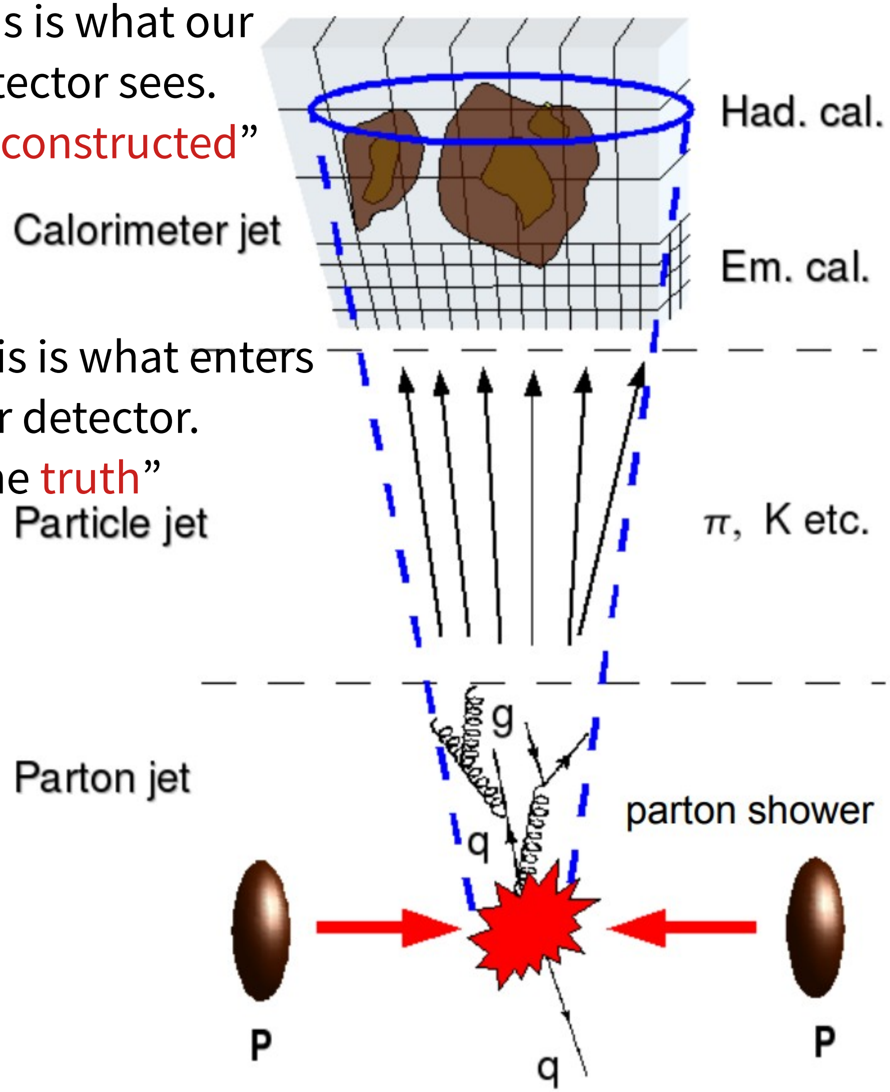
$$y = f(\vec{x}; \vec{\theta})$$

- Example: calibrated jet energy given calorimeter measurements
- ML gives opportunity to handle correlations

Calibrating Jets

This is what our detector sees.
“reconstructed”

This is what enters our detector.
“the truth”



Calibration: $f(x) \equiv < p_T^{\text{reco}} | p_T^{\text{true}} = x > = x$

reconstructed energy = true energy *on average*

Why is this hard?

- Calorimeter response not gaussian
- Calorimeter response not fully known
- Different jets respond differently
  - Depends on jet content (ie: b- vs light jet)

Calibration depends on many variables. NN is a convenient way to account for their correlation.

Numerical Inversion

- **Learning to predict p_T^{true} given p_T^{reco} does not close**
  - Due to assumptions on learning sample (ie: p_T^{true} distribution)
- **Numerical inversion to the rescue:**

1) Learn calorimeter response "$f(p_T^{true})$"

2) Invert function $f(p_T^{true})$ to get calibration $p_T^{reco,cal}=f^{-1}(p_T^{reco})$

3) Apply multiple $f^{-1}(x;\theta_i)$ sequentially to correct for θ_i

$$p_{\mathrm{T}}^{\mathrm{reco}} \mapsto \hat{p}_{\mathrm{T}}^{\mathrm{reco}} = f_{\theta_n}^{-1}\left(\cdots f_{\theta_2}^{-1}\left(f_{\theta_1}^{-1}\left(p_{\mathrm{T}}^{\mathrm{reco}}\right)\right)\cdots\right)$$

- **Neural Net can be used to create a single $f^{-1}(x;\theta_{0..n})$**
  - Generalized Numerical Inversion: A Neural Network Approach to Jet Calibration

Numerical Inversion With NN

1)**Learn detector response by training NN *L(x, θ)***

- Approximation of $f_\theta(x) = \langle p_{\mathrm{T}}^{\mathrm{reco}} | p_{\mathrm{T}}^{\mathrm{true}} = x, \theta \rangle$

2)**Learn inversion by training NN *C(L(x, θ), θ)* to predict *x***

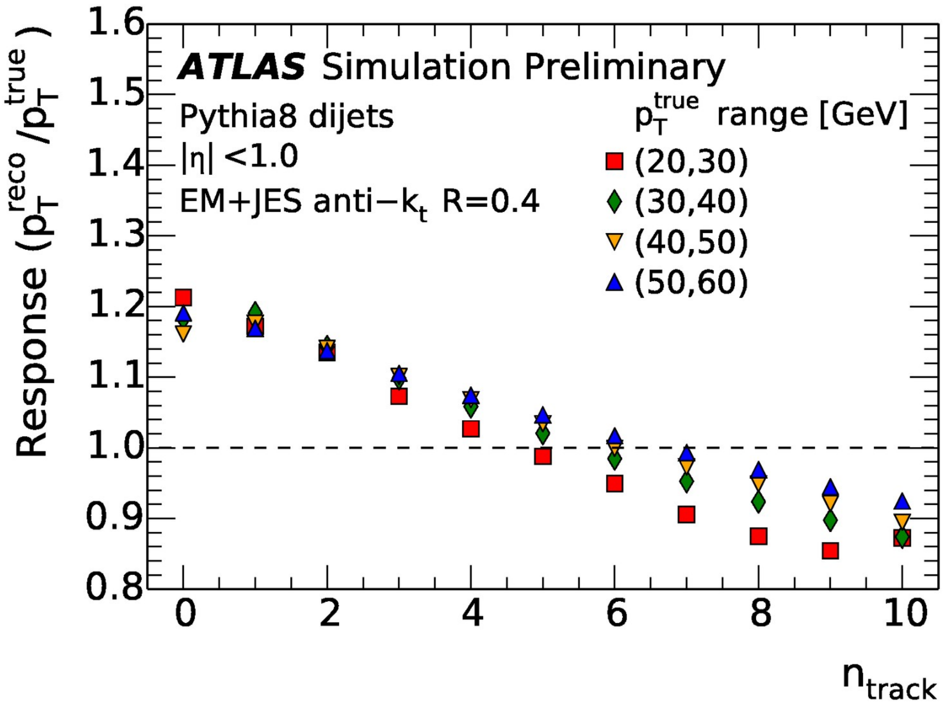
3)**Apply calibration**

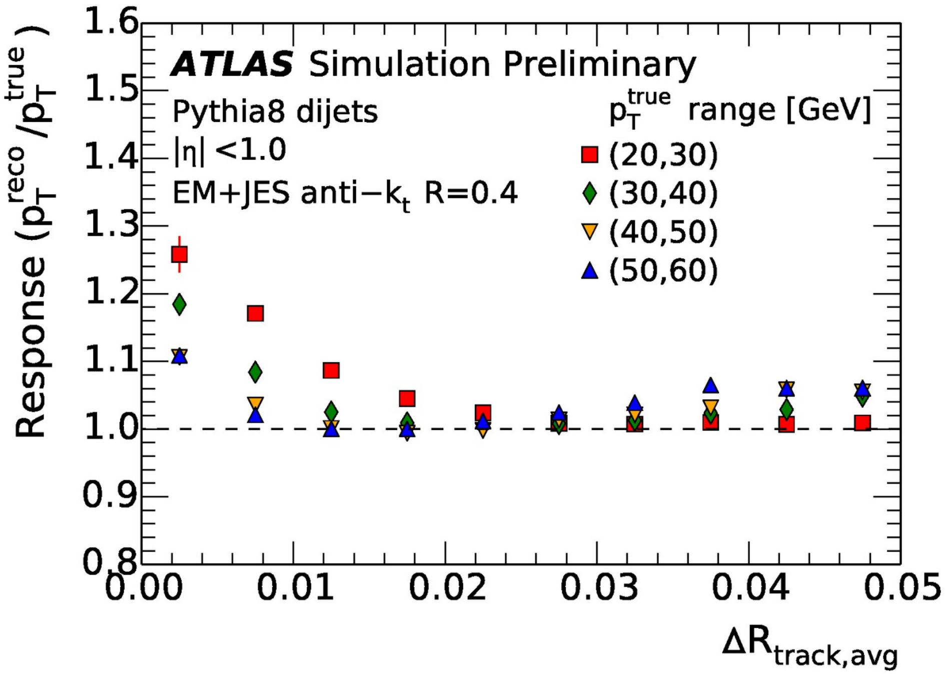
$$p_{\mathrm{T}}^{\mathrm{reco}} \mapsto \hat{p}_{\mathrm{T}}^{\mathrm{reco}} = C(p_{\mathrm{T}}^{\mathrm{reco}}, \theta)$$

Note:

- θ are jet moments that predict jet type (ie: radiation pattern of quark vs gluon)

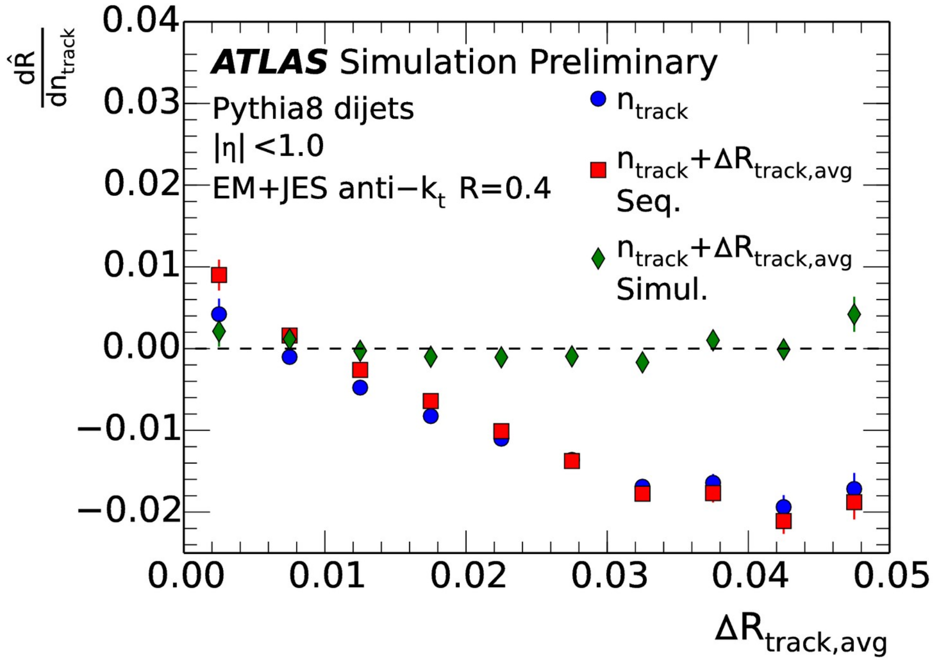
Response vs θ

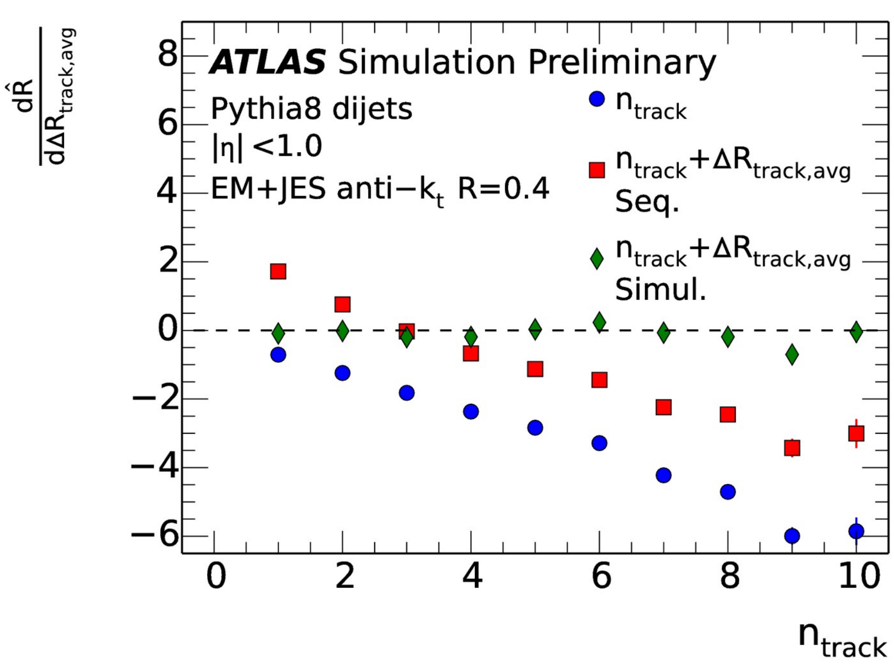




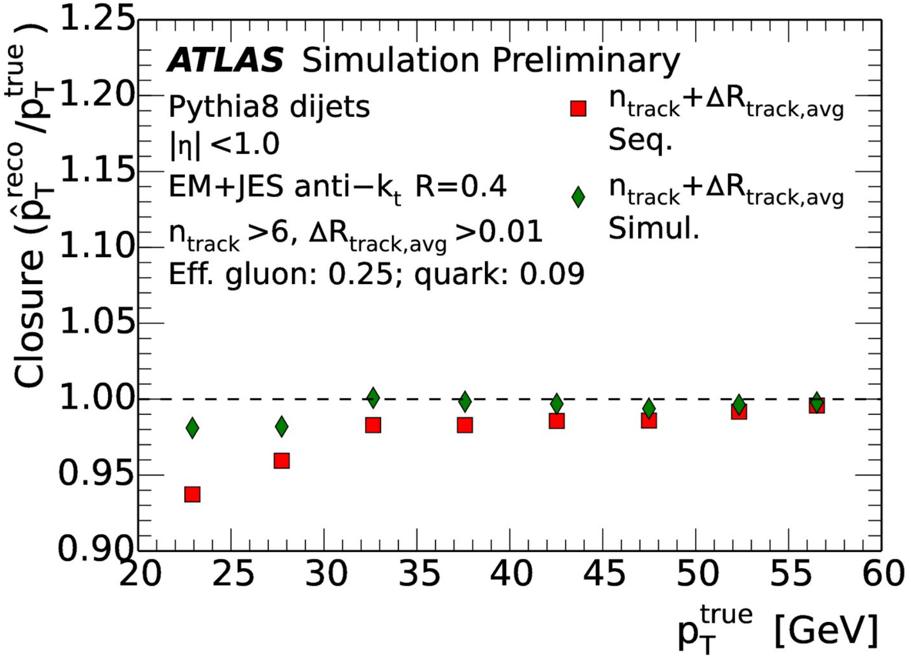
Removing Dependence on θ

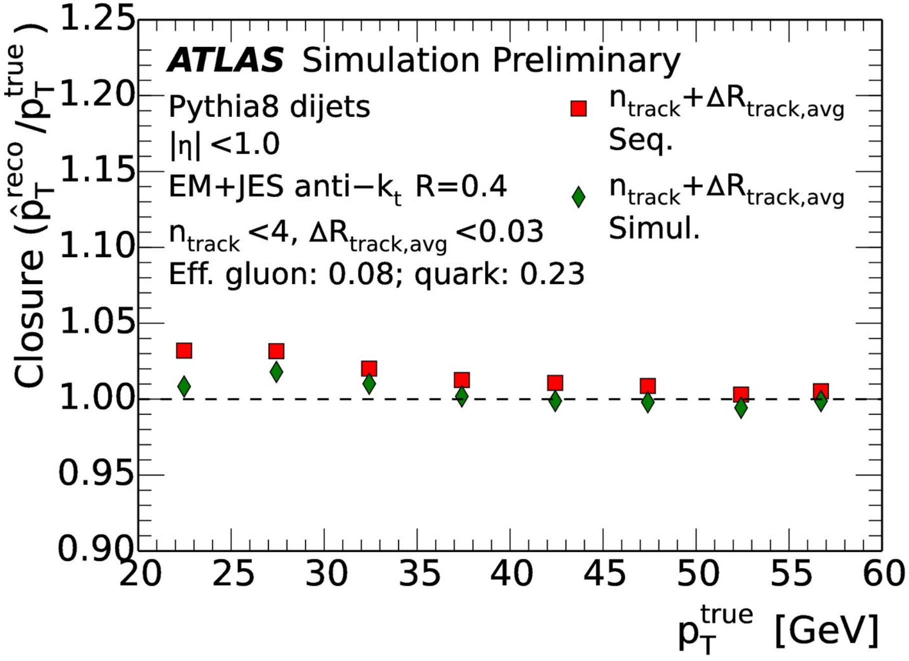
$$\hat{R} \equiv < \hat{p}_T^{\mathrm{reco}} / p_T^{\mathrm{true}} | p_T^{\mathrm{true}} = x >$$





Closure of Method

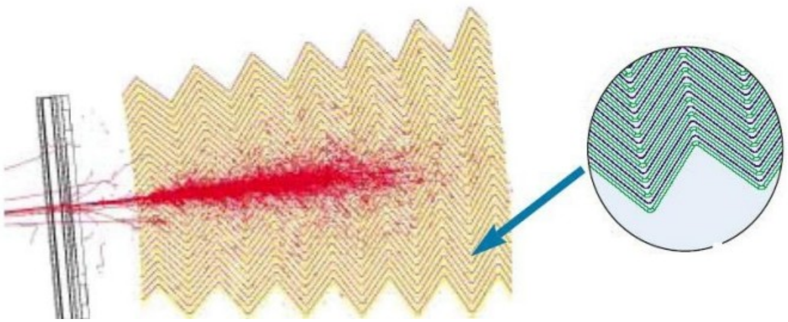




Generative Models

SIMU-2018-04

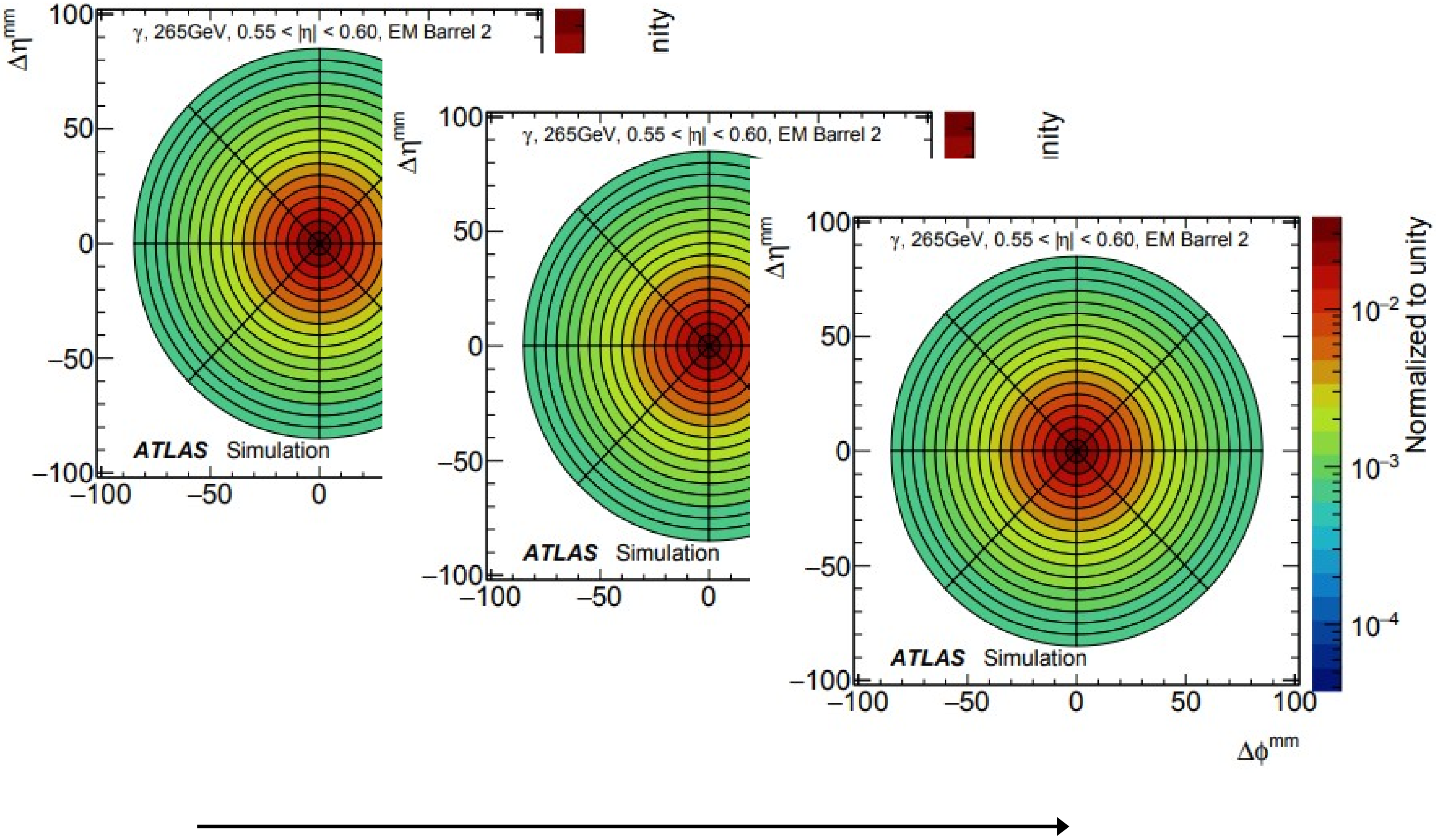
- **Simulating our detector is very computationally expensive**
  - Especially harmonica structure of our Ecal



- **Fast Simulation is very useful**
  - PCA: database of frozen showers, assigned randomly
  - GAN: Showers generated via neural network (inputs: particle and rand num)

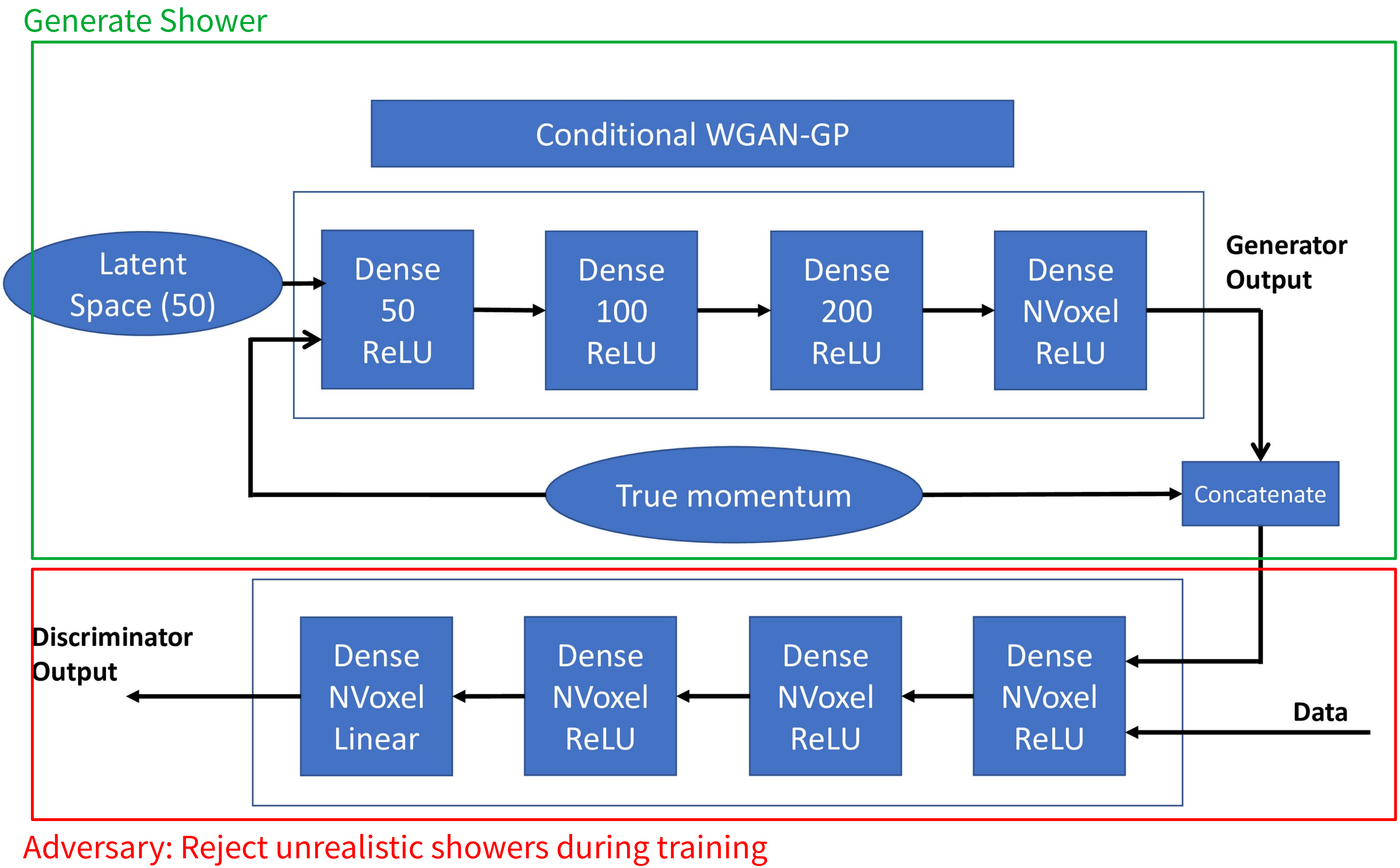
Voxelizting a Shower

Think of it as a 3D image.



calorimeter depth

Generative Adversarial network



Track Reconstruction

https://exatrkx.github.io/

All possible hit combinations

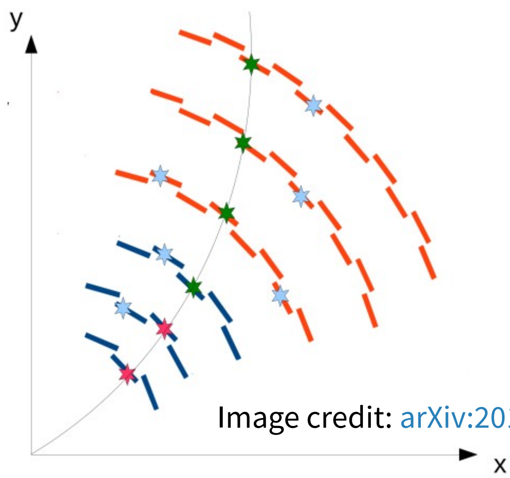
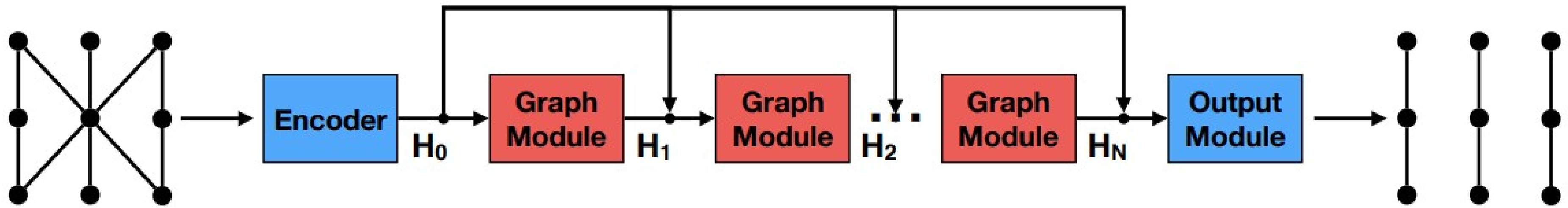


Image credit: arXiv:2010.13557

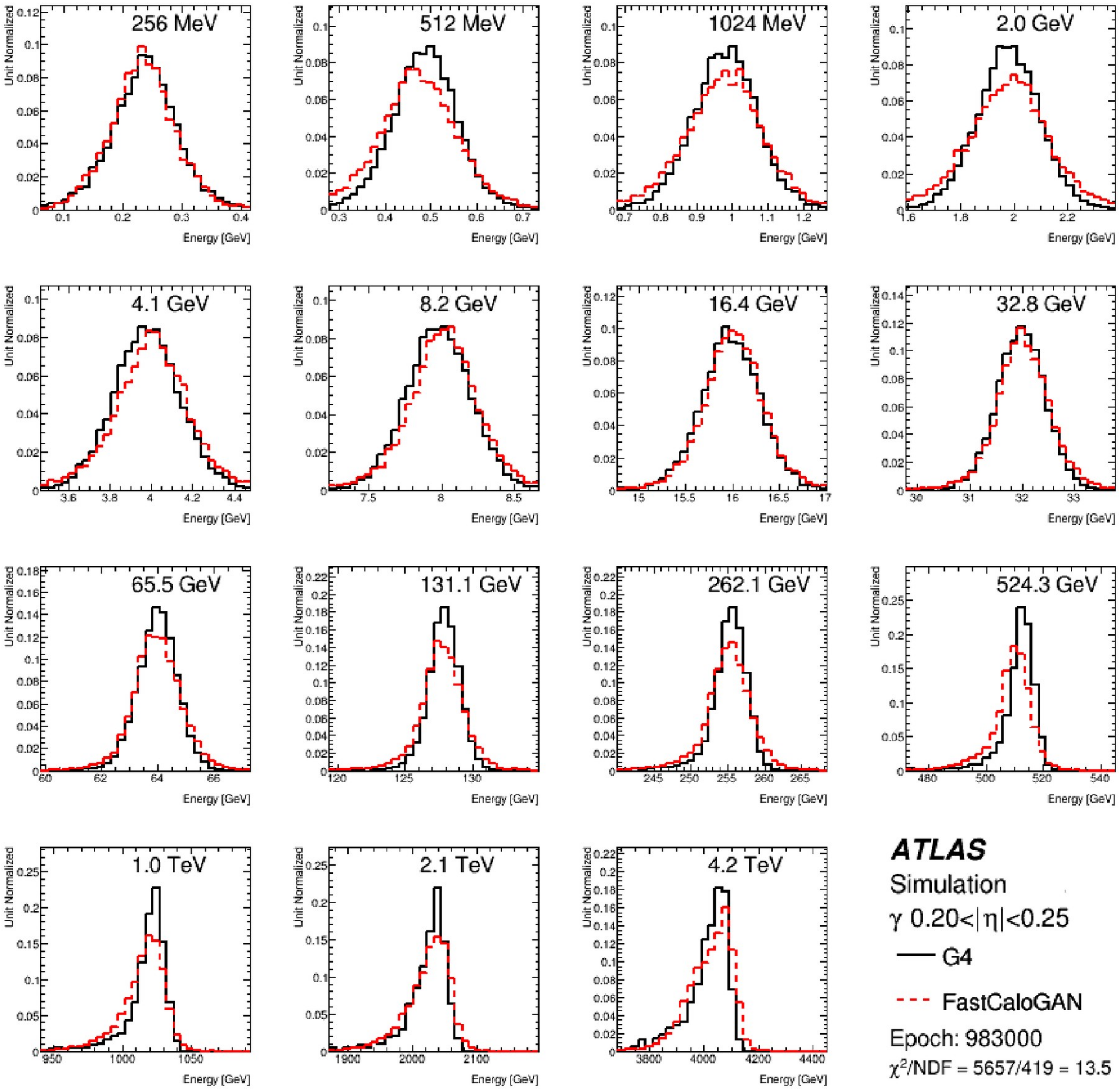
Hit combinations corresponding to a single track



arXiv:2003.11603: bunch of people at LBL

Great intro to GNN's: Relational inductive biases, deep learning, and graph networks

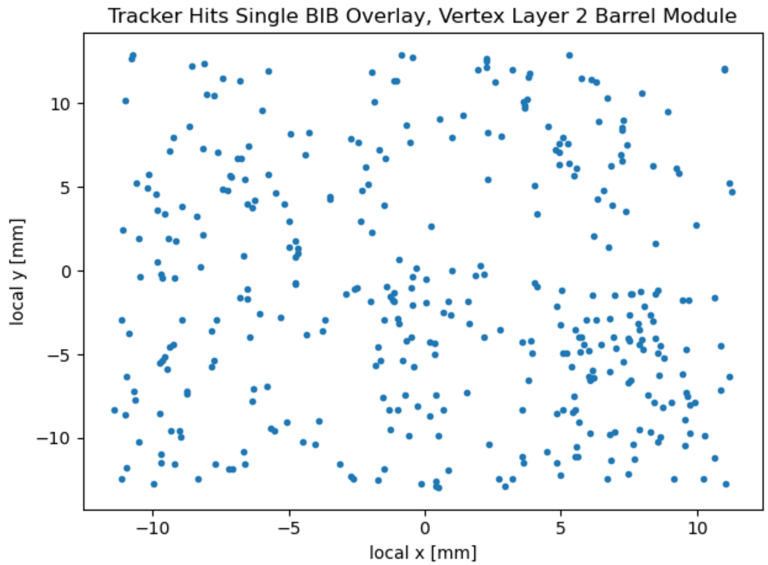
GAN Results


256 MeV
512 MeV
1024 MeV
2.0 GeV
4.1 GeV
8.2 GeV
16.4 GeV
32.8 GeV
65.5 GeV
131.1 GeV
262.1 GeV
524.3 GeV
1.0 TeV
2.1 TeV
4.2 TeV
Unit Normalized
Energy [GeV]
ATLAS
Simulation
γ 0.20<|η|<0.25
G4
FastCaloGAN
Epoch: 983000
χ²/NDF = 5657/419 = 13.5

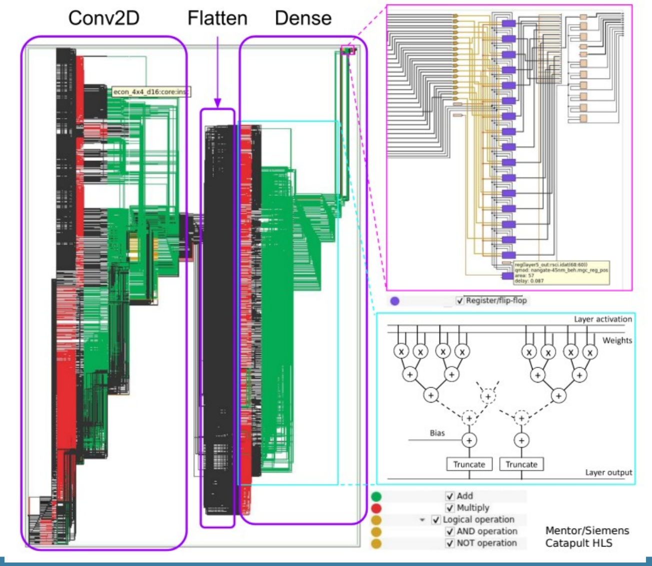
Detector Data Compression

arXiv:2105.01683

Example "pixel data"



Logic implementation using HLS4ML



- **Pixel detectors (tracking)**
  - x/y positions with hits
  - Many hits! Lots of data…
  - Very sparse
- **Compression**
  - Lossless already good
  - Can you improve via lossy? (use NN)
- **Caveats to NN "in detector"**
  - Need to be energy efficient
  - Need to be radiation hard
    - Are NN's inherently reliant to SEUs?

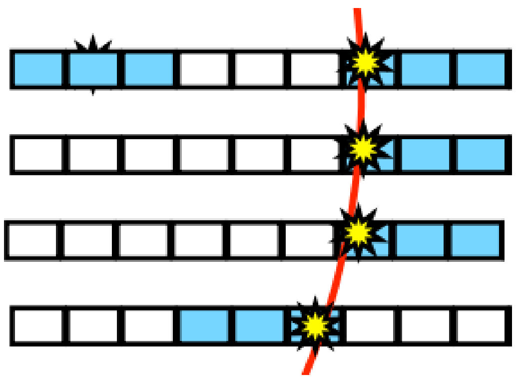
Pattern Recognition via Associative Memory

arXiv:2101.05078

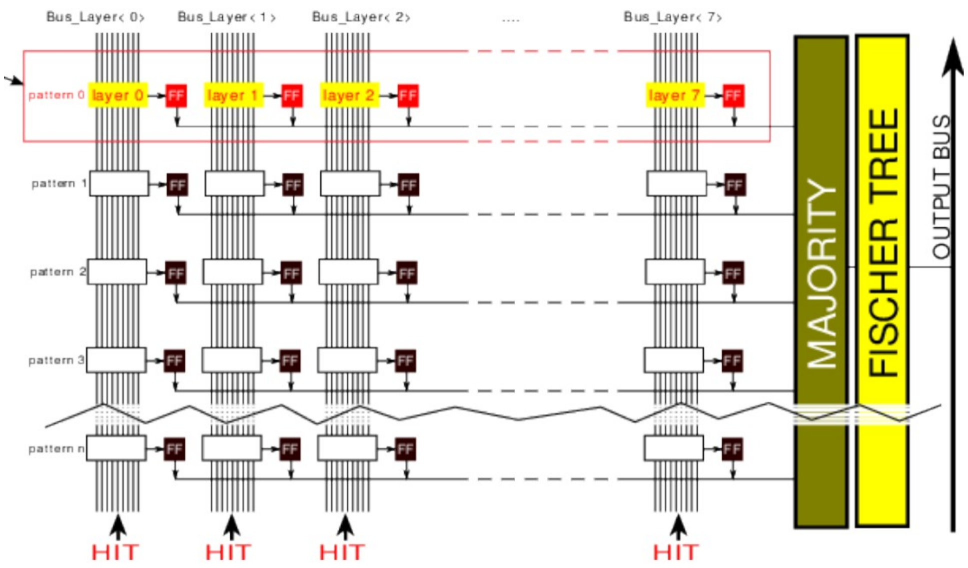
Hardest part of track reconstruction is pattern recognition.

Pattern recognition = which hits came from the same particle

Pattern: possible pattern a track leaves in the tracking detector.



Associative Memory: simultaneous matching against I patterns



Creating pattern bank of N most likely patterns is ML.

Proposed Short Projects

Classification (Supervised)

Boosted Top vs W/Z vs QCD Tagger

Classification (Unsupervised)

Anomalies in Hadronic Resonances using Auto-Encoders

Regression

Improving Jet Energy Scale/Resolution

Event Generation

Event Simulation Using Adversarial Networks

Note: Limitation in project ideas = generating data (currently my responsibility)