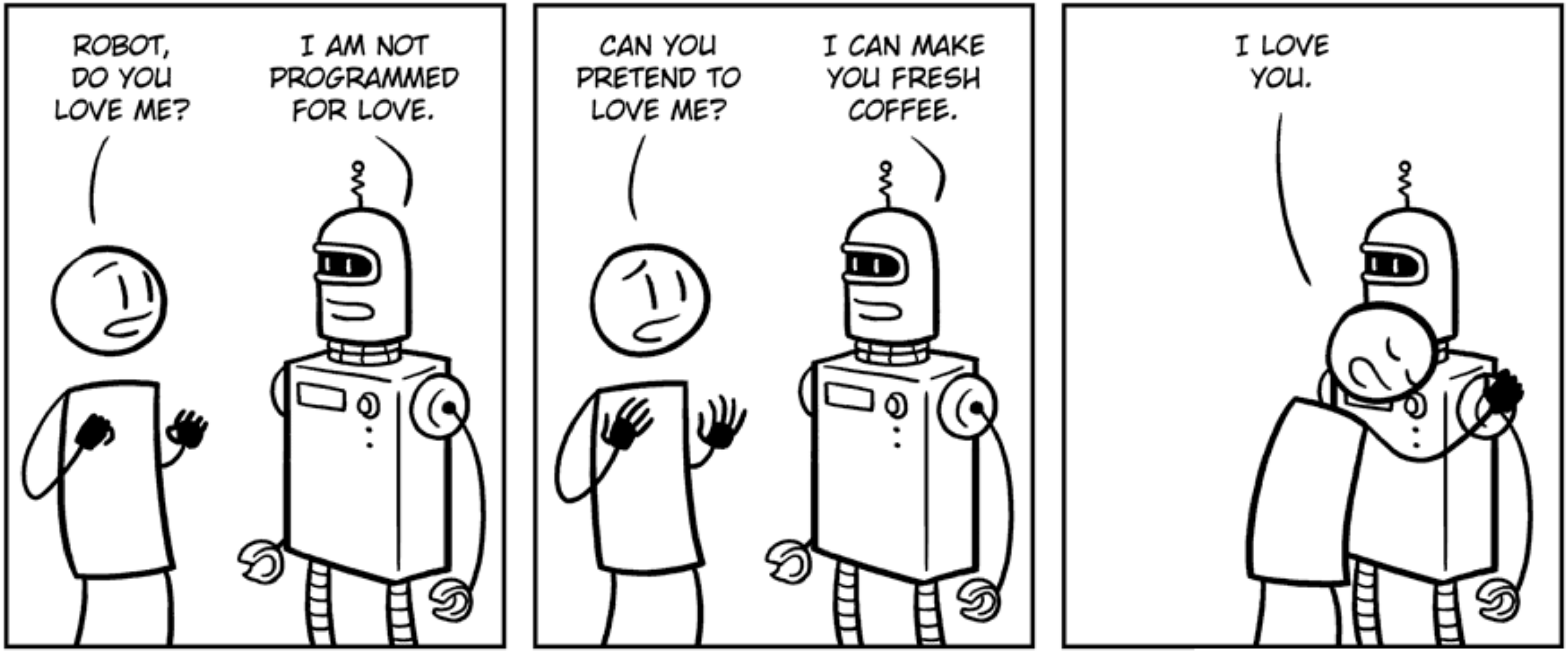


Evolution of AI approaches at the LHC

P. Harris (MIT,IAIFI,A3D3)

Overview of this talk

Origins of Deep Learning at LHC

Where Algorithms are Going

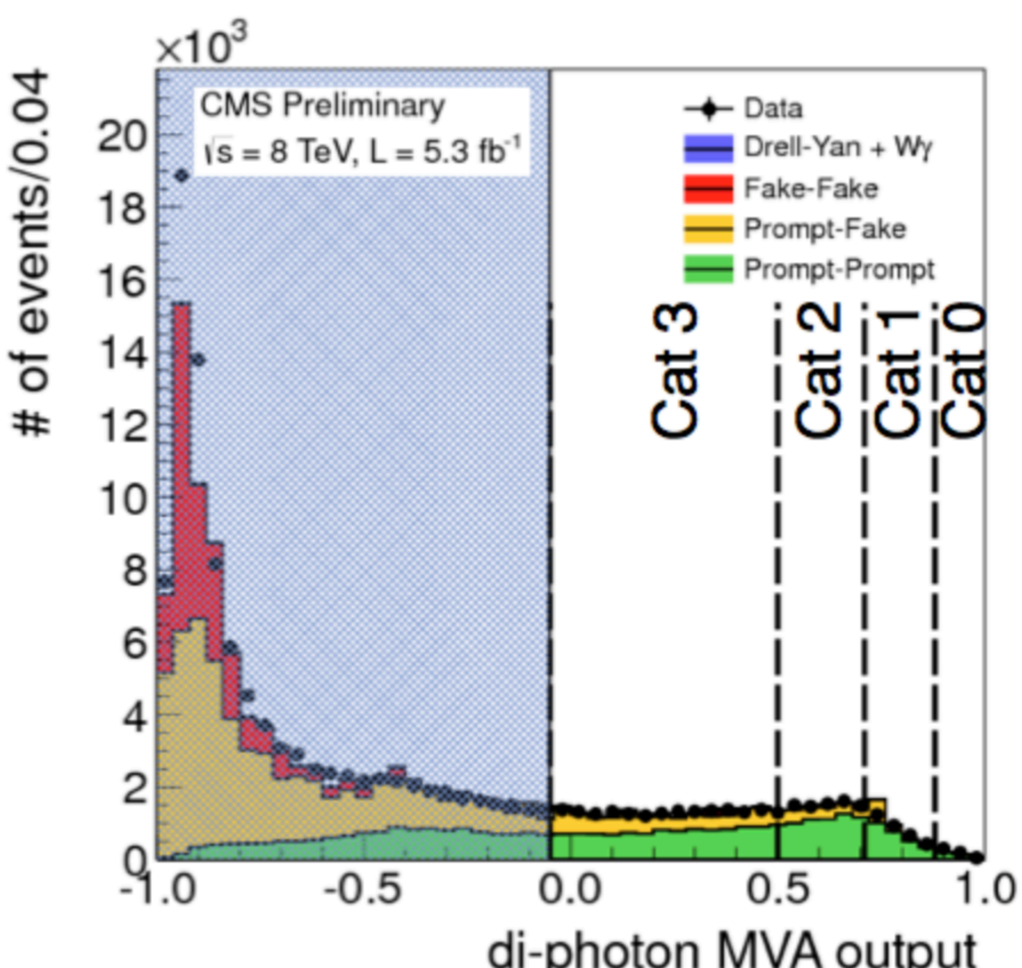
Where Experiments are Going

Deep Learning For Others (LIGO/Neut)

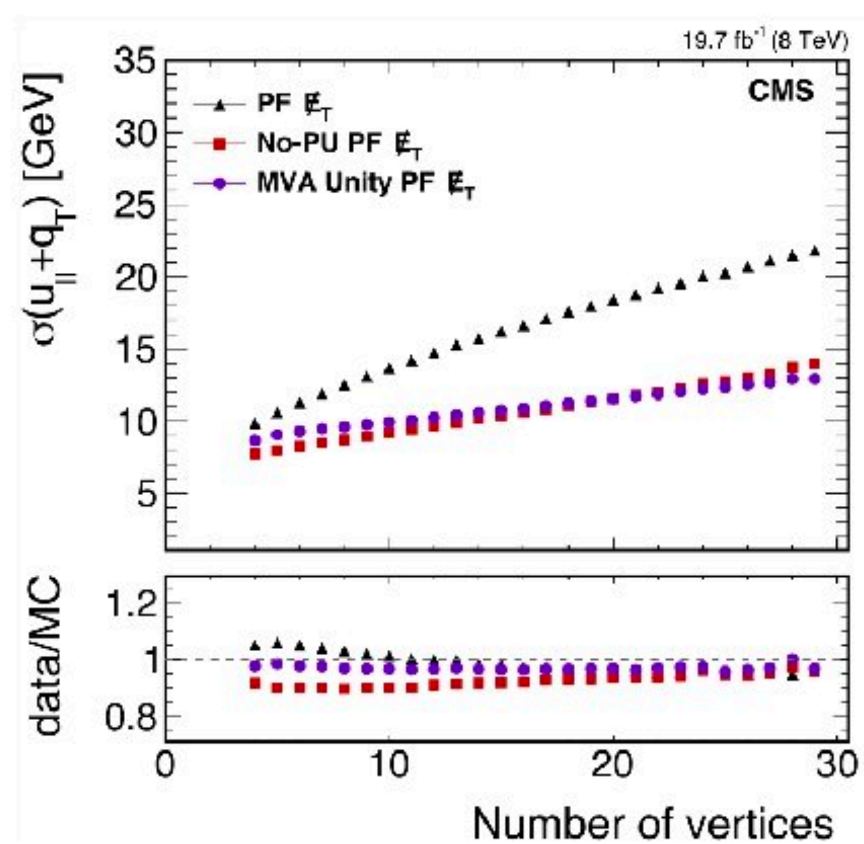
Anomaly Detection

Power of ML

- The LHC has long kept up with trends from ML
  - In the era of BDTs, many big advancements came
  - Many were critical for the observation of the Higgs boson



MVA MET



A Change in Past 5 years

- Deep Learning has heavily push progression to other arch
- Why was this case?
  - New DL frameworks dramatically changed flexibility
  - We can now train for arbitrary loss functions
- DL frameworks are very effective with GPUs
  - GPUs allow us to have many inputs > 100! (BDTs capped at 40-50)

TMVA Loss
Regression(MSE)
Classier(CCE)

Pytorch/Keras Loss

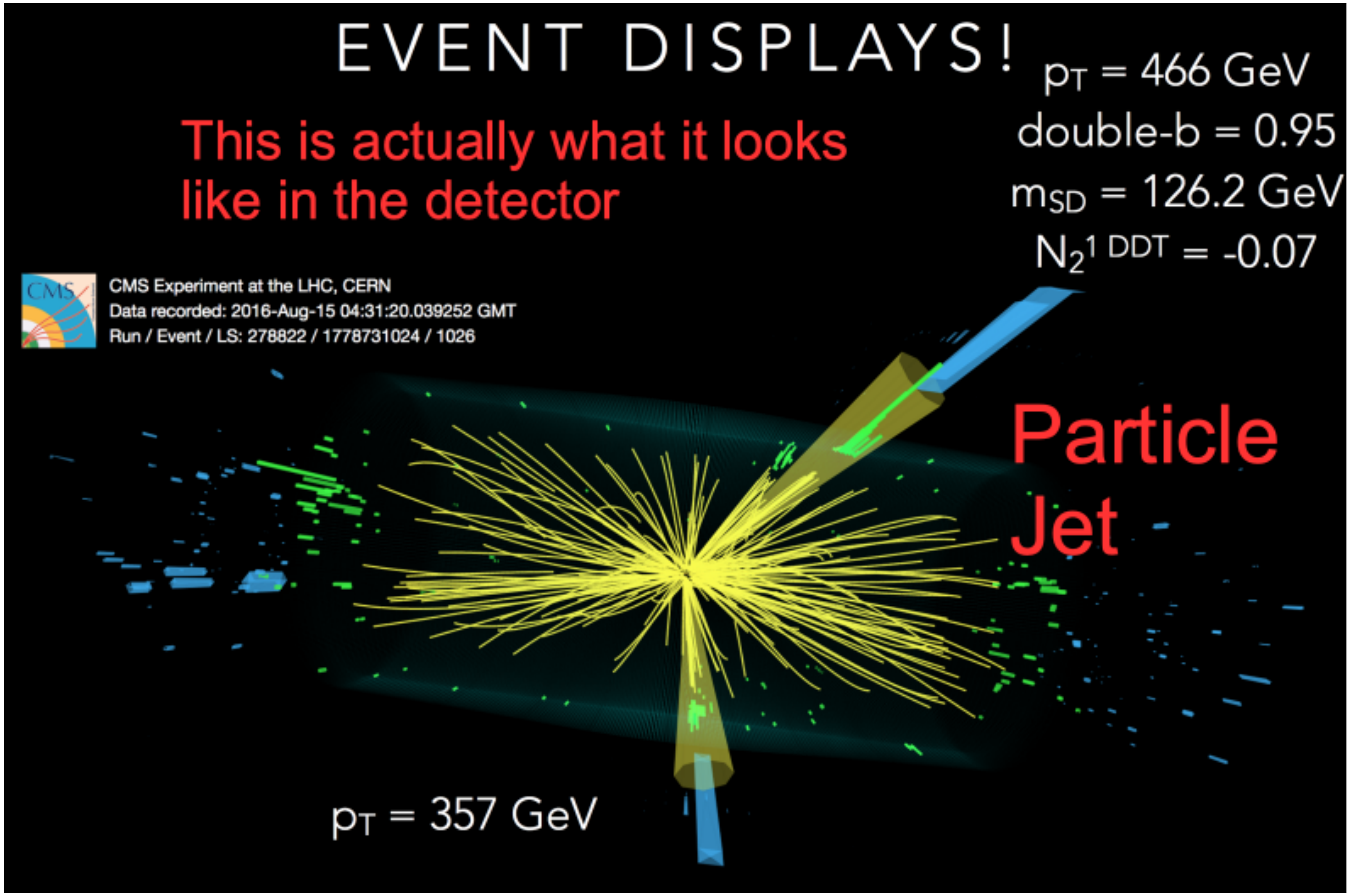
- Mean Absolute Error Loss
- Mean Squared Error Loss
- Negative Log-Likelihood Loss
- Cross-Entropy Loss
- Hinge Embedding Loss
- Margin Ranking Loss
- Triplet Margin Loss
- Kullback-Leibler divergence

+Custom Loss

Powerful Gradient Tools

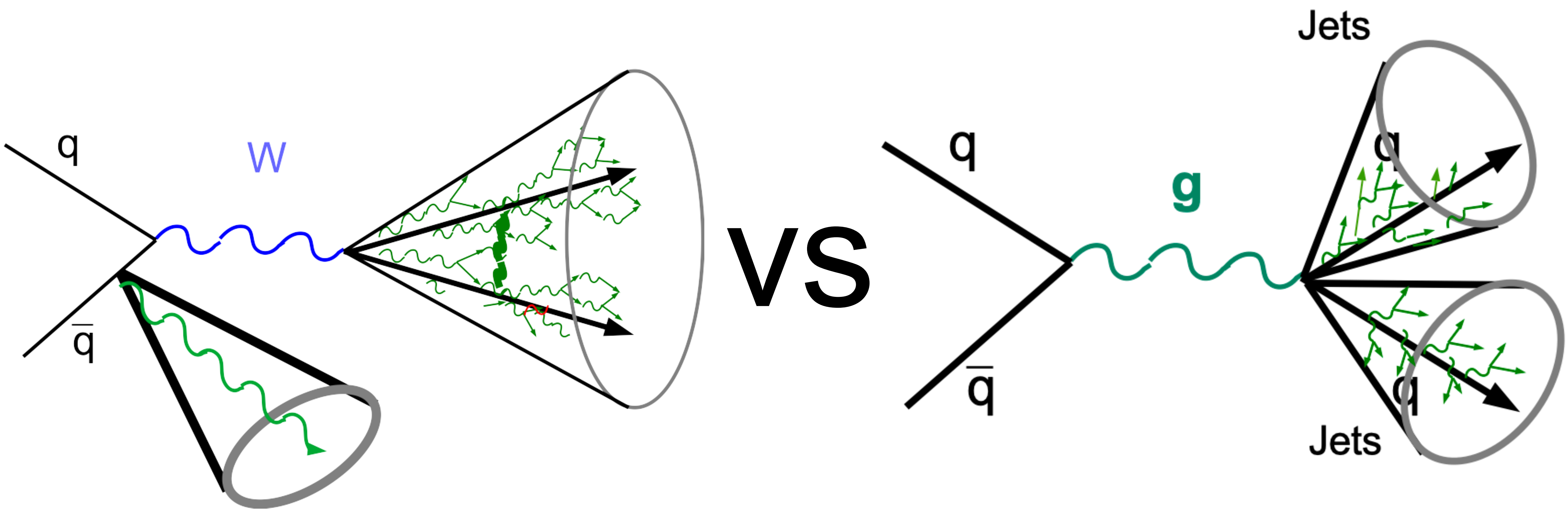
Native GPU support

For Jets



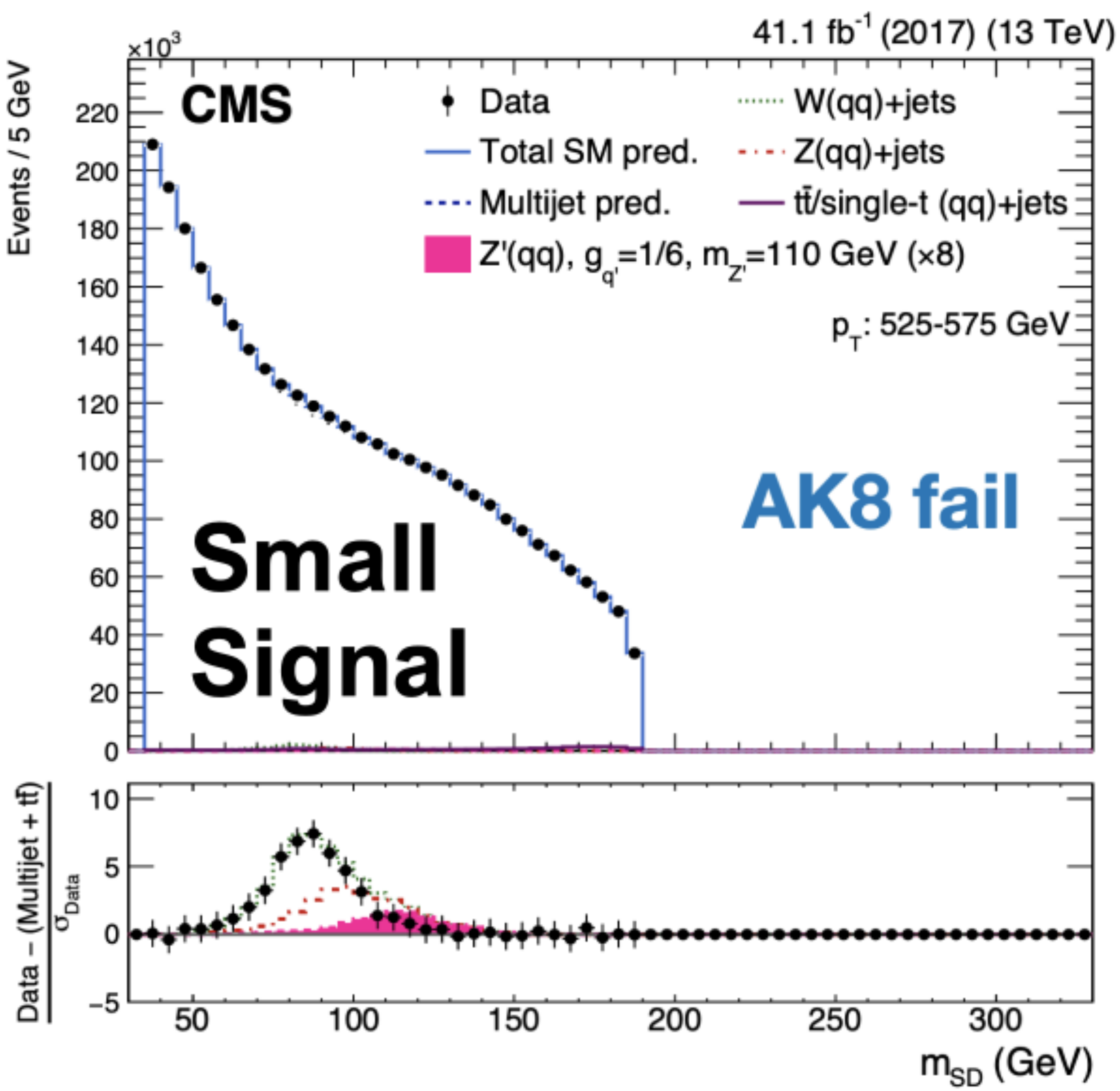
- Jets have the chance to benefit greatly from Deep Learning
  - There is a large variety of variables that we can construct

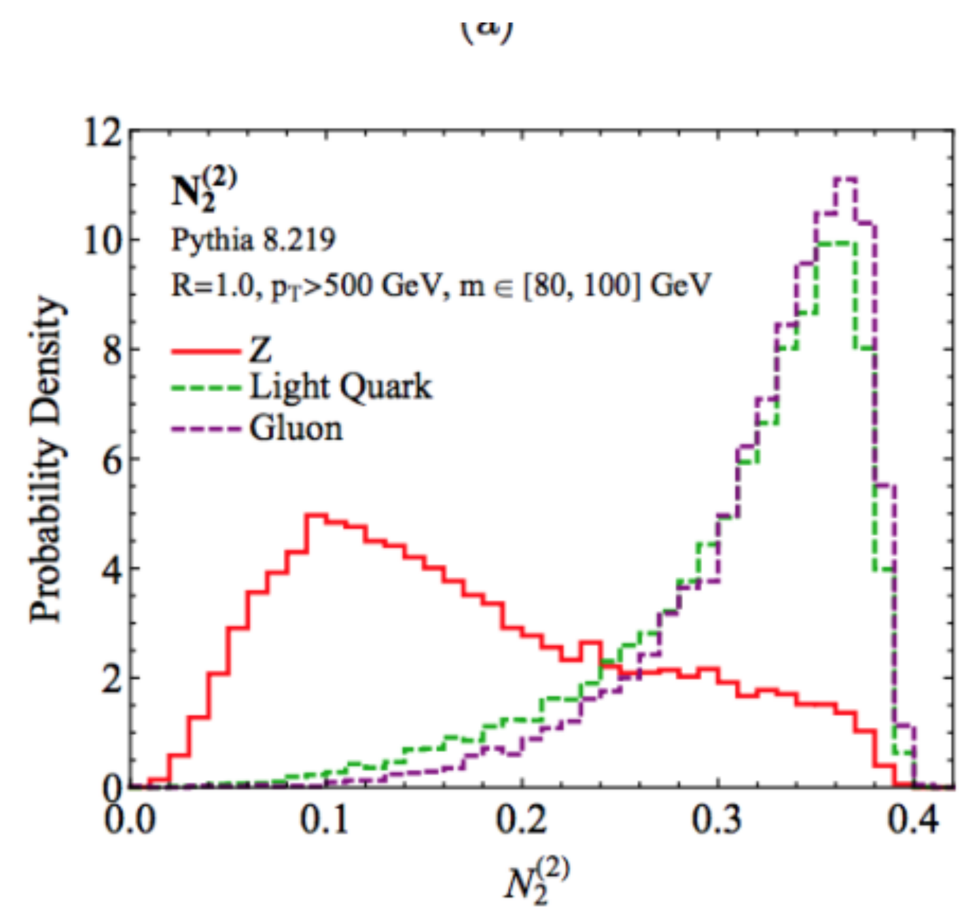
Selecting a Jet in data



- If you select a jet in data and look at the mass
- There is an enormous amount of background
- But, you can potentially find a W boson or a Higgs boson

Jets have 100s of particles



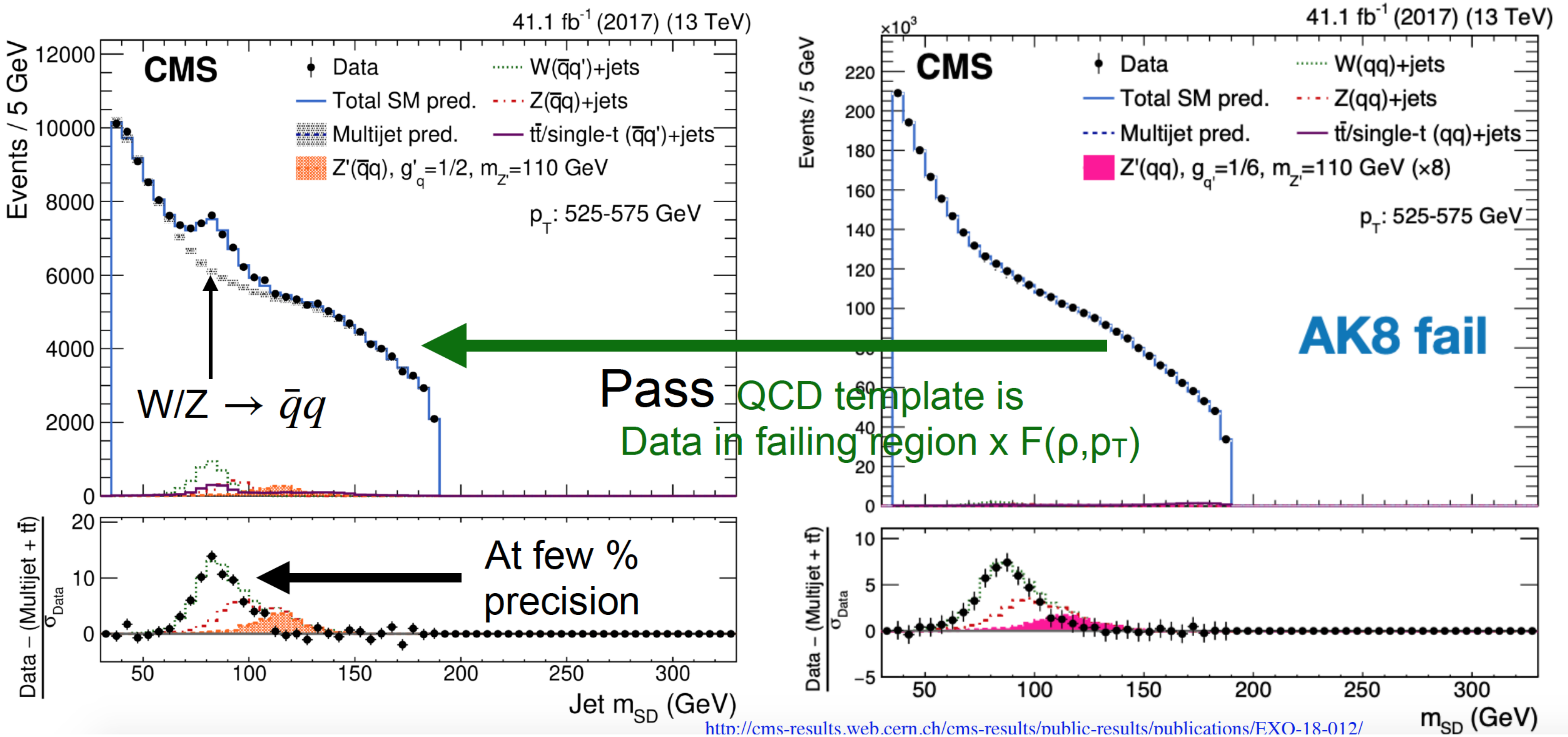


$$e_3^{(\beta=2)} = \sum_{i<j<k\in\text{jet}} z_i z_j z_k \theta_{ij}^2 \theta_{ik}^2 \theta_{jk}^2 \approx \sum_{i<j} z_i z_j \theta_{ij}^2 \theta_i^2 \theta_j^2$$

$$\approx \sum_{i<j} z_i z_j \max(\theta_i^2, \theta_j^2) \theta_i^2 \theta_j^2 \approx \sum_{i<j} \rho_i \rho_j \max(\theta_i^2, \theta_j^2),$$

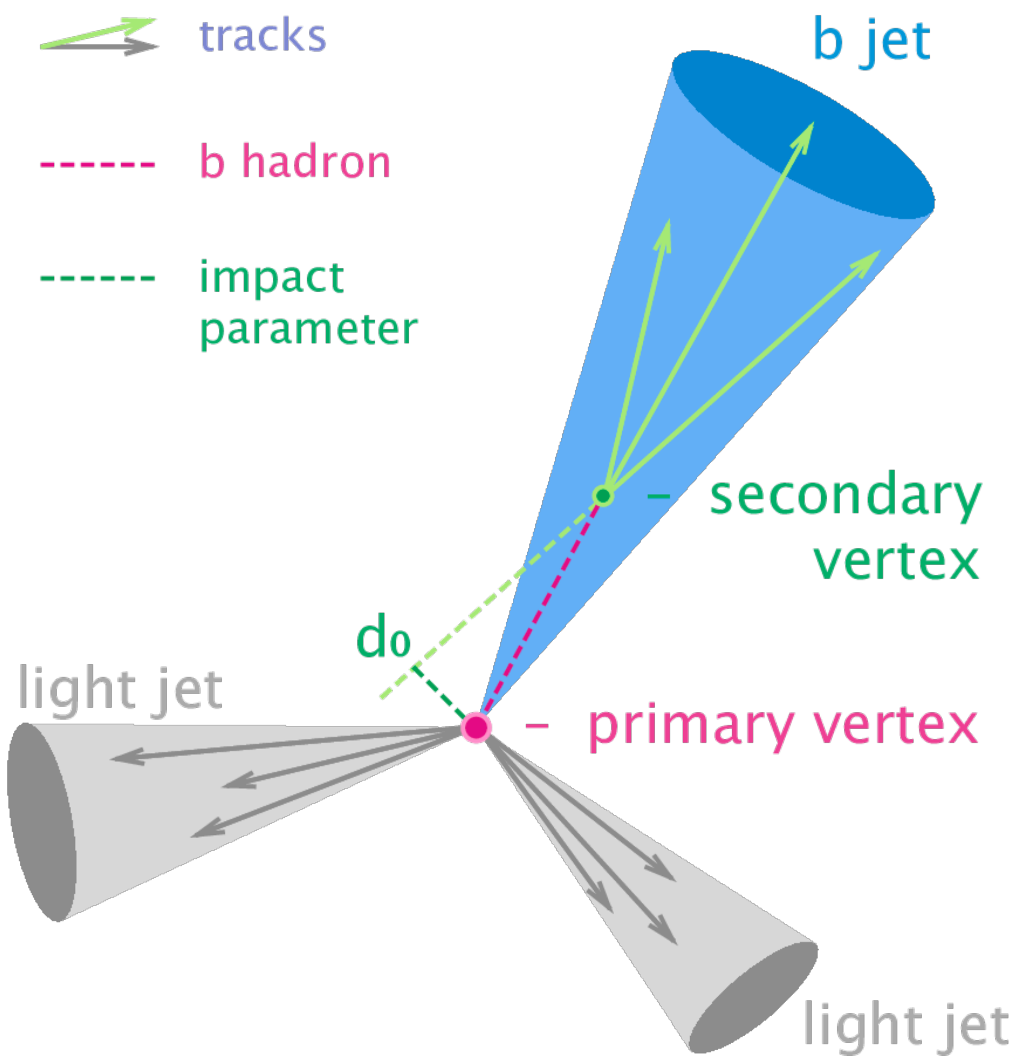
- Large backgrounds and many particles good ML problem
    - To see anything we need to reduce bkg by x10-100

Without Deep Learning



- Already with jet substructure we can start to see resonances
- But these analyses set the stage for a great deep learning

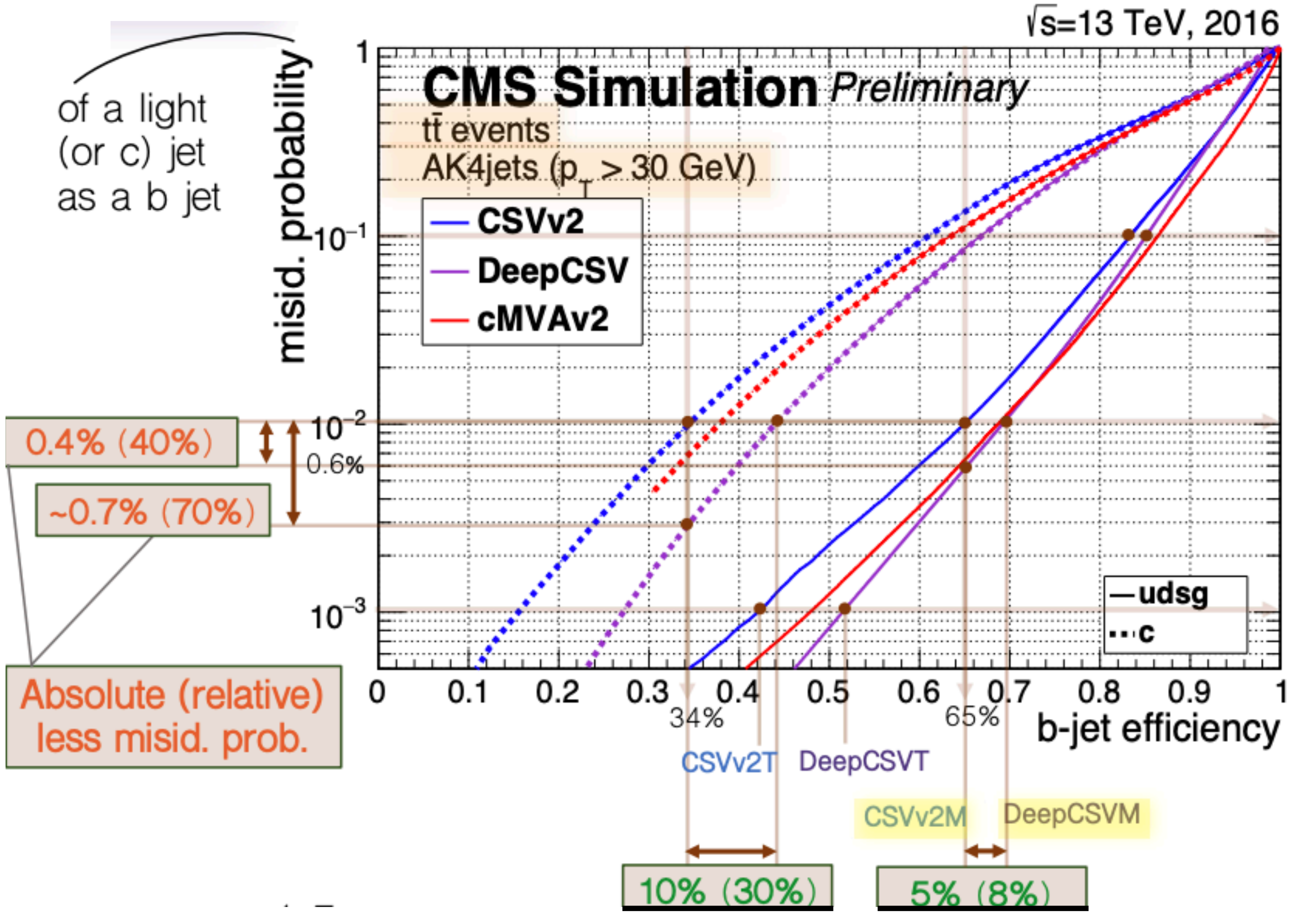
Where Deep Learning really started to help

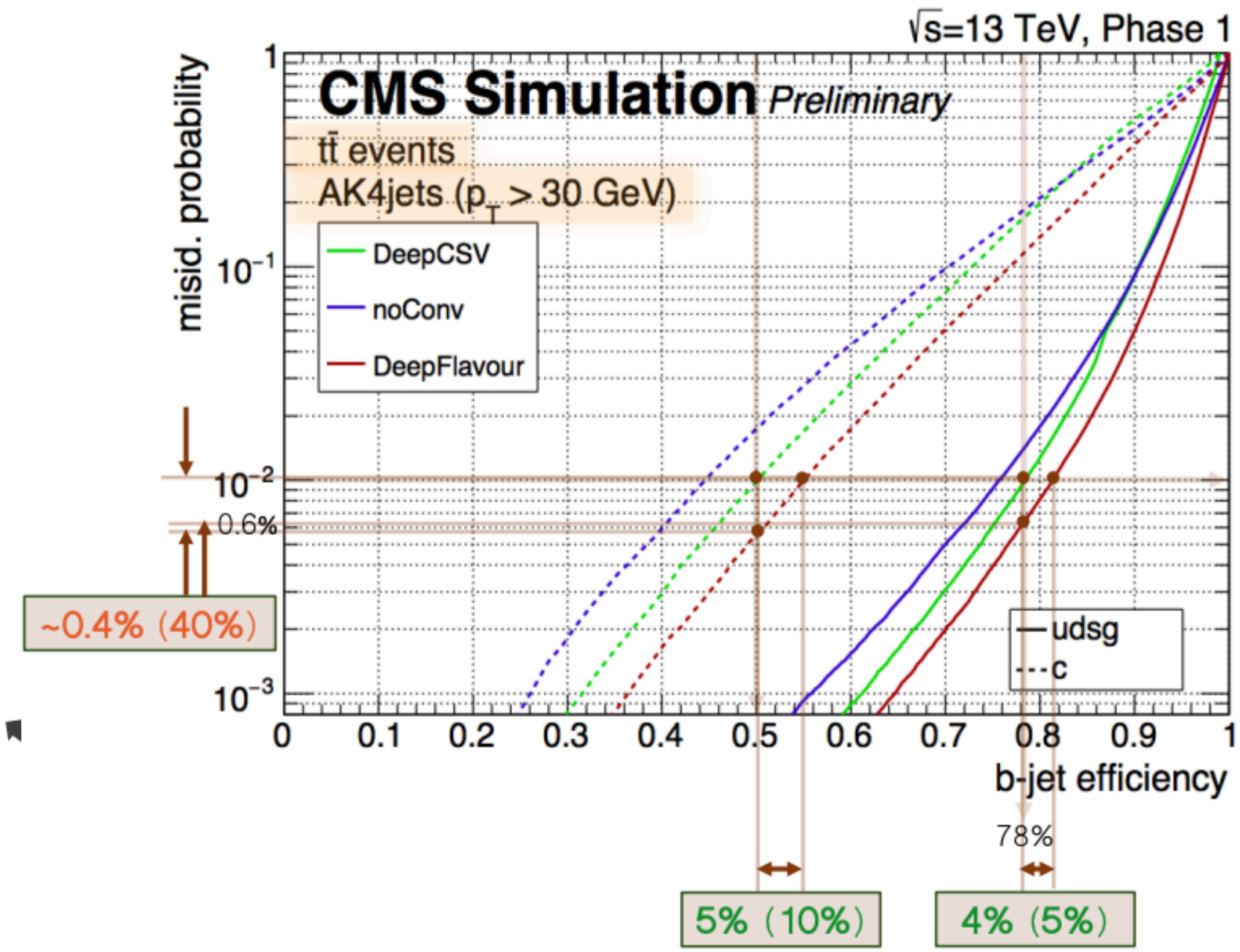


B-tagging has lots of handles
Also there is lots of background
Discrimination is key!

- For b-quark tagging Deep Learning brought a lot of gains
  - Part of these gains was from the fact that things were not tuned

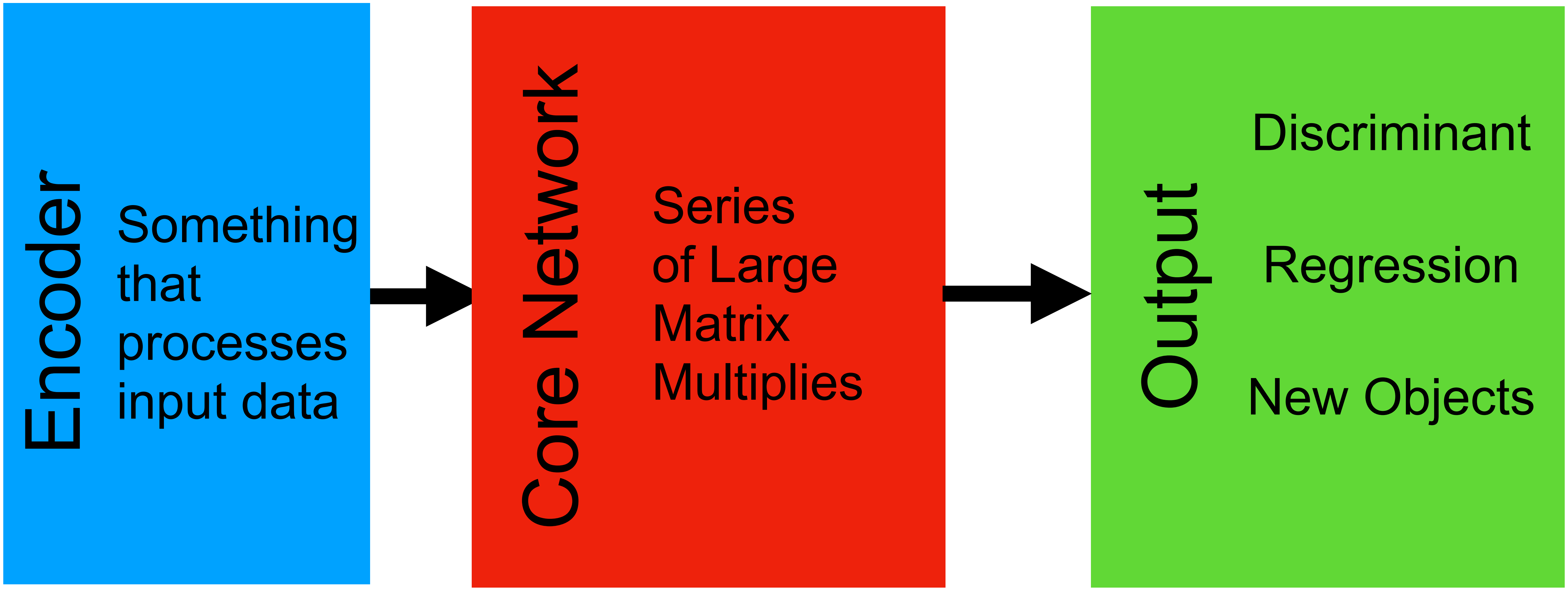
Where Deep Learning really started to help





- For b-quark tagging Deep Learning brought a lot of gains
  - Part of these gains was from the fact that things were not tuned

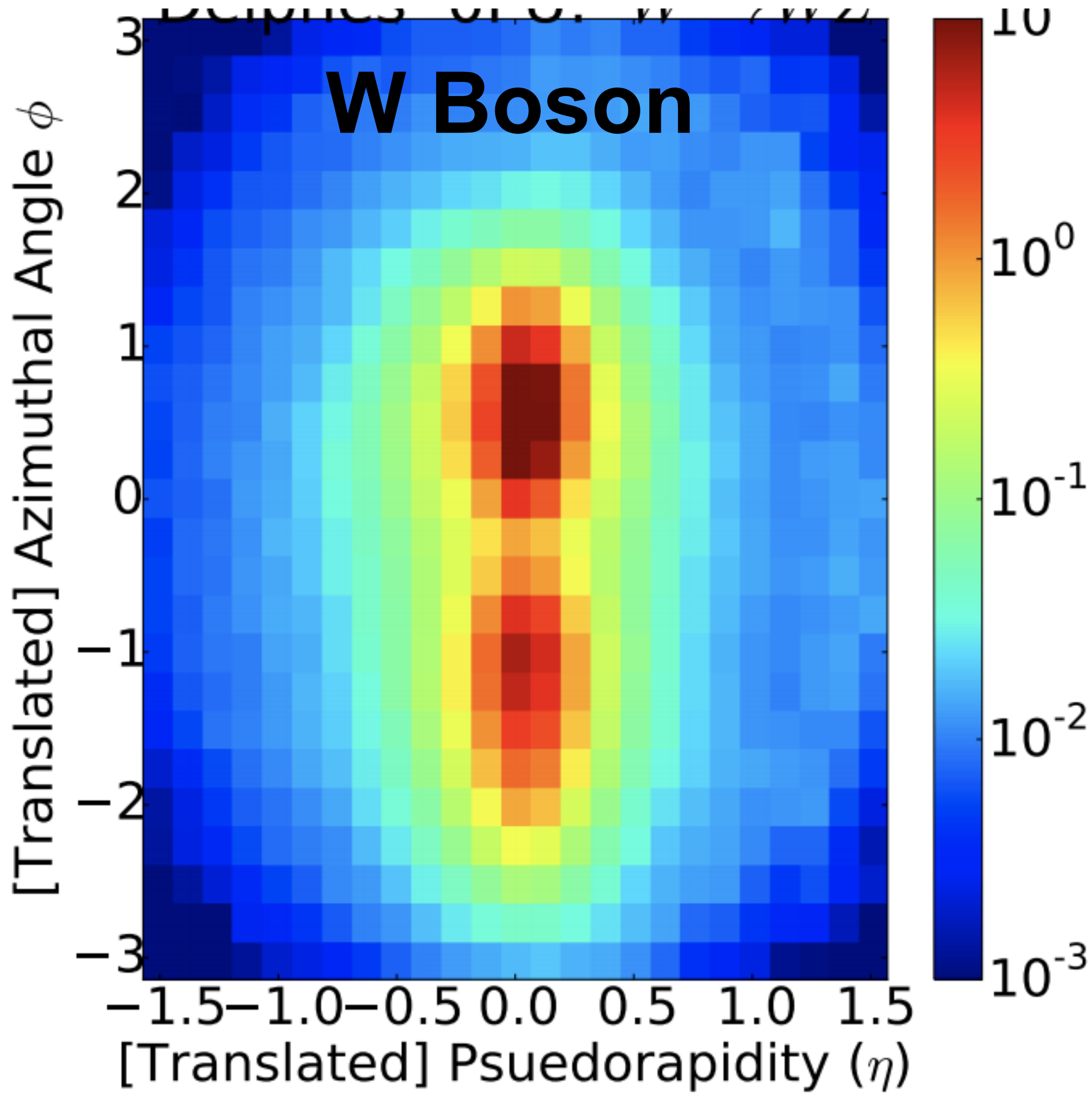
Neural Network Arch



- Encoders capture much of the physics to all for standard DL tools
  - Responsible for much of the big gains over the past few years

So what has happened?

- Big gains in deep learning have come from embedded data
- How can we take a complex object like a jet and process it

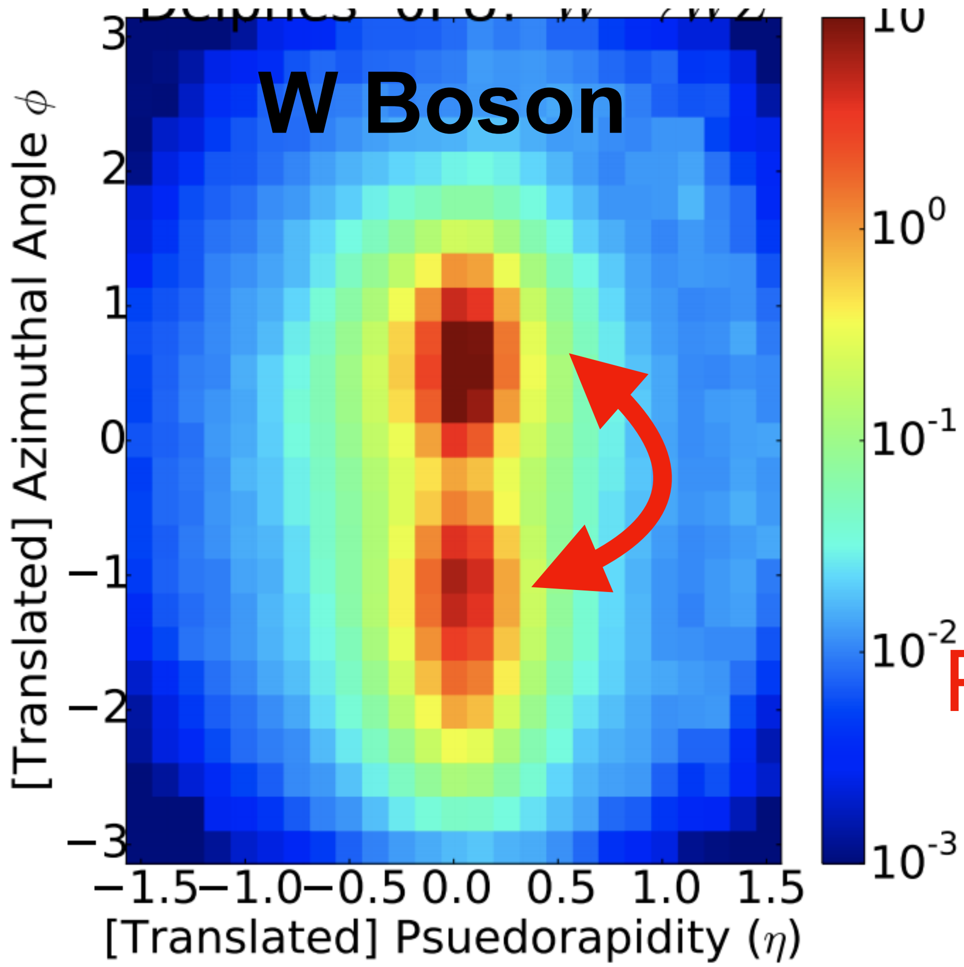


Jet Image
Take a jet and do an energy weighted sum of the particles centered about the jet axis

When we first tried this Convolutional Neural Networks for Imag Id were the new big thing!

So what has happened?

- Big gains in deep learning have come from embedded data
- How can we take a complex object like a jet and process it



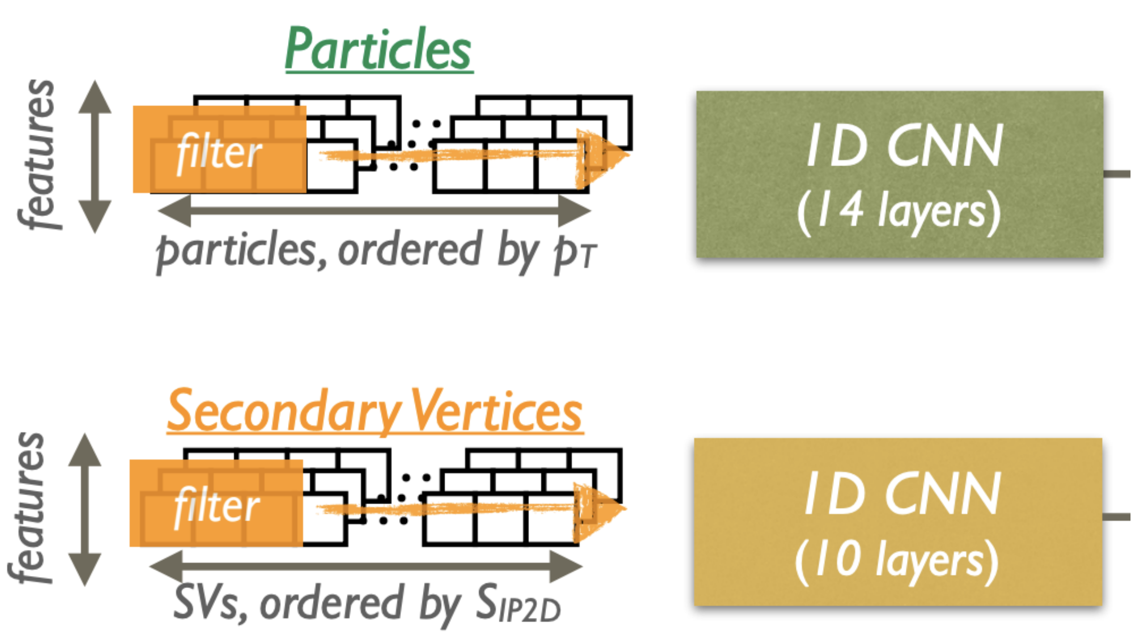
Jet Image
Take a jet and do an energy weighted sum of the particles centered about the jet axis

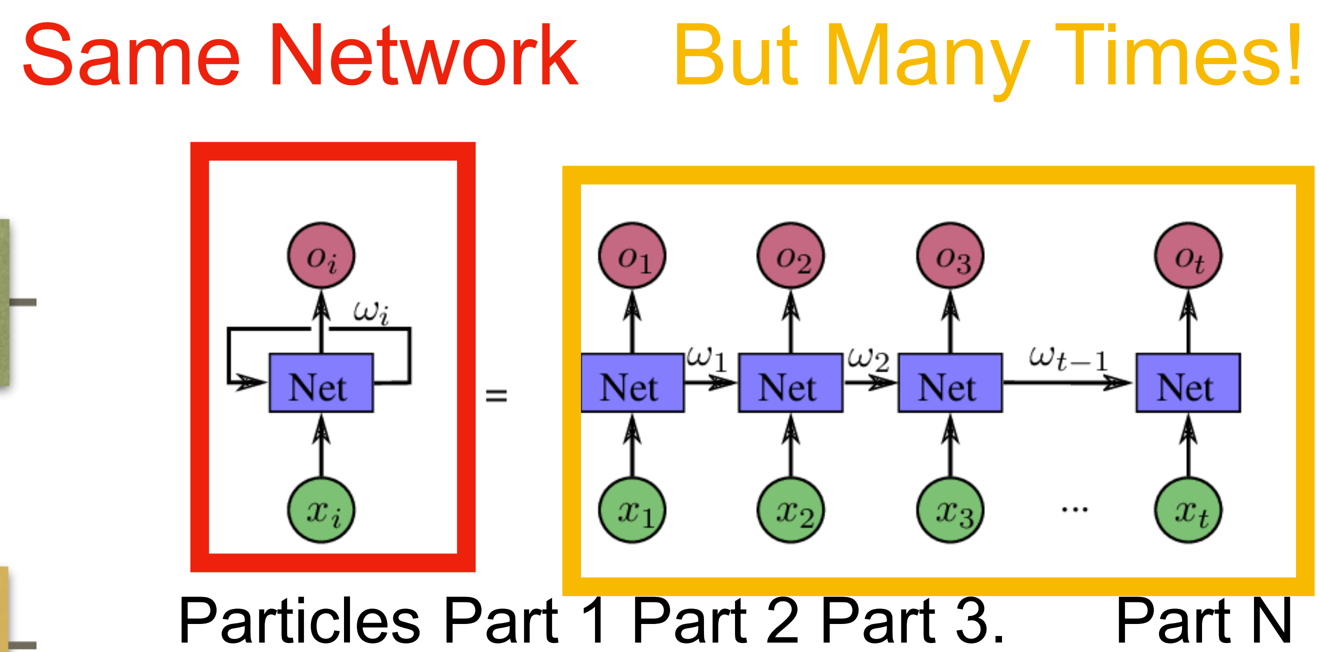
Problem! Image Not Lorentz Invariant

Jet p_T will change the overall position!

Improving the idea

- Instead we can consider sending in 4 vectors
  - Utilizing 4-vectors gives us a notion of lorentz invariance



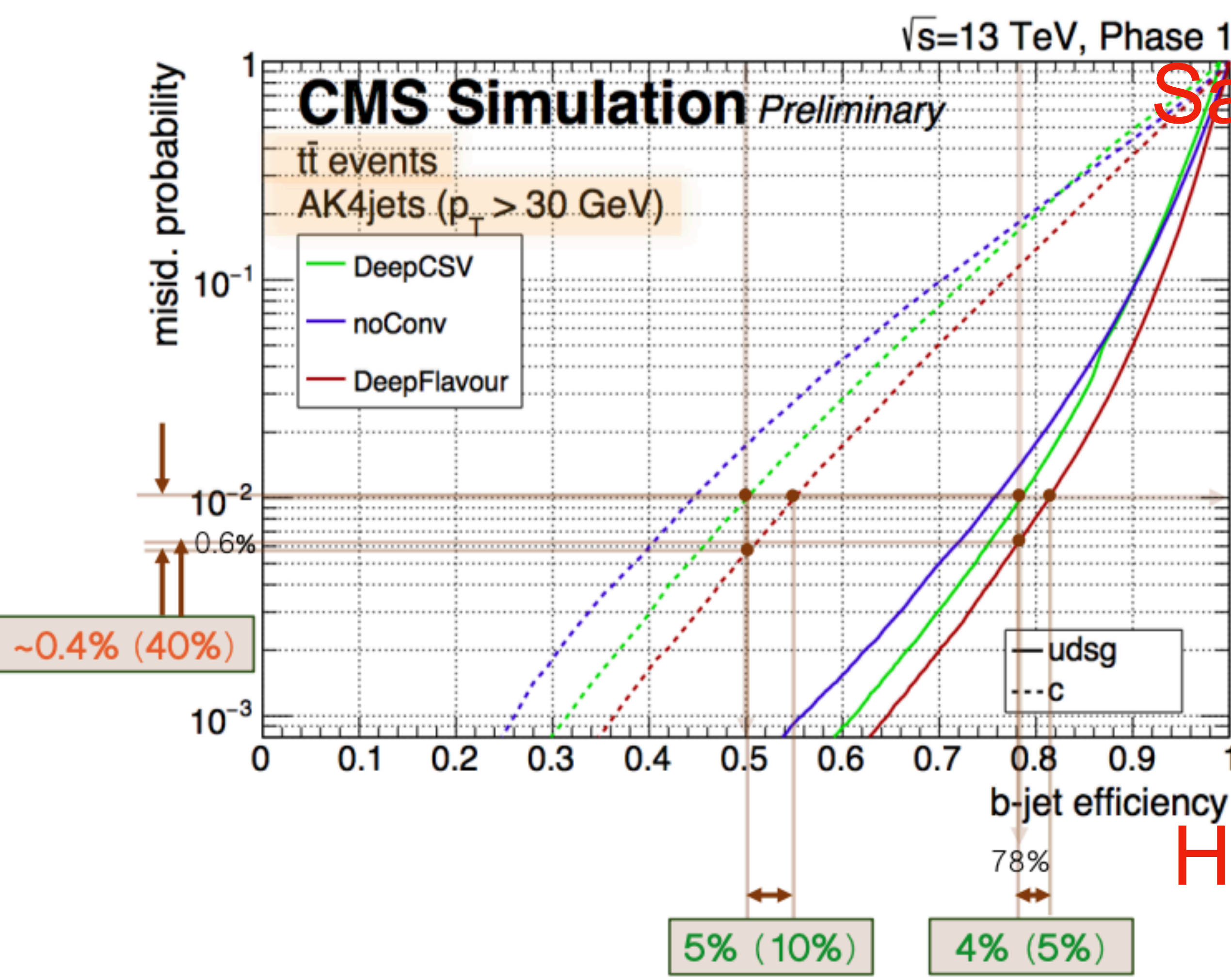


Take's a single Particle in at a time

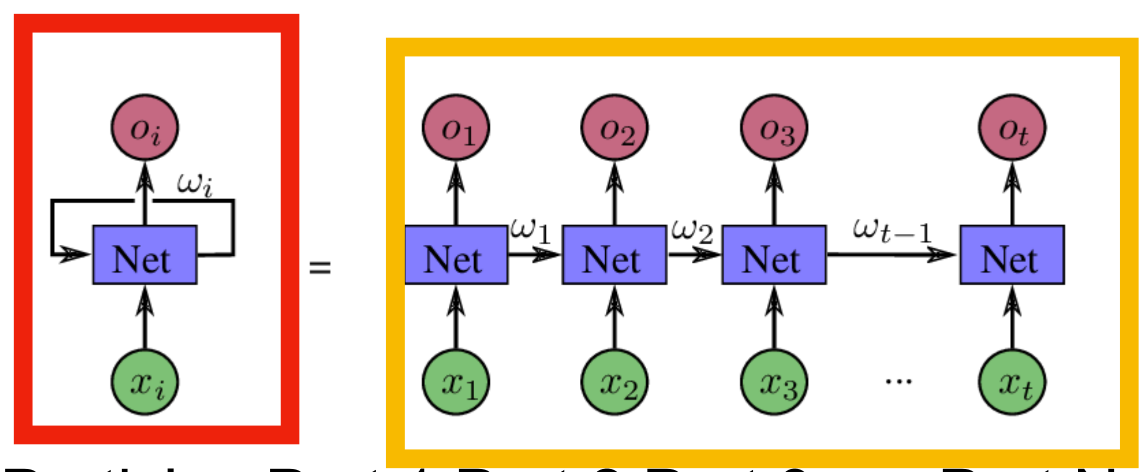
Popular in 2018 when Recurrent Neural Networks were the crazy

Improving the idea

- Instead we can consider sending in 4 vectors
  - Utilizing 4-vectors gives us a notion of lorentz invariance



Same Network   But Many Times!



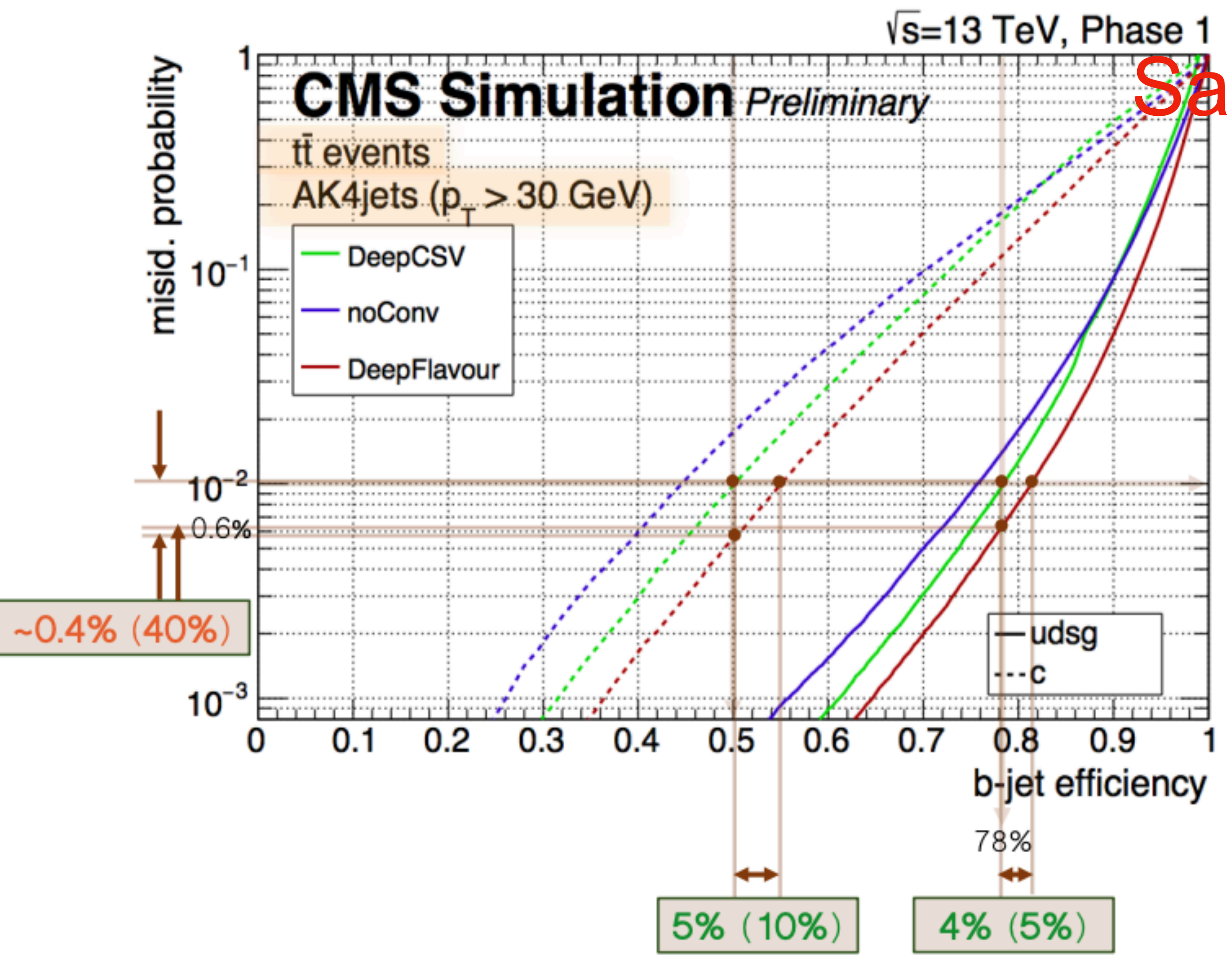
Particles Part 1 Part 2 Part 3.   Part N

However it lacks particle correlations

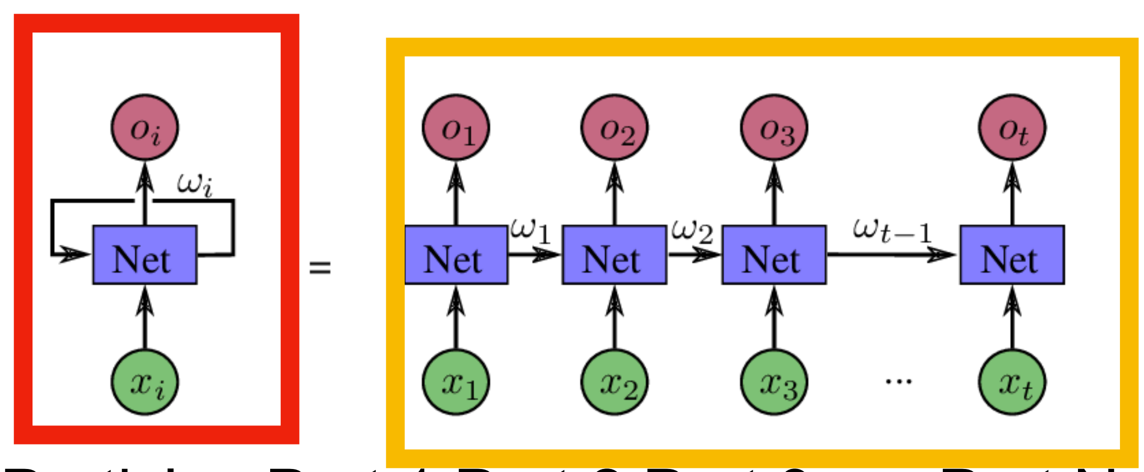
Lack or particle correlations limits Jet Identification abillity

Improving the idea

- Instead we can consider sending in 4 vectors
  - Utilizing 4-vectors gives us a notion of lorentz invariance



Same Network   But Many Times!



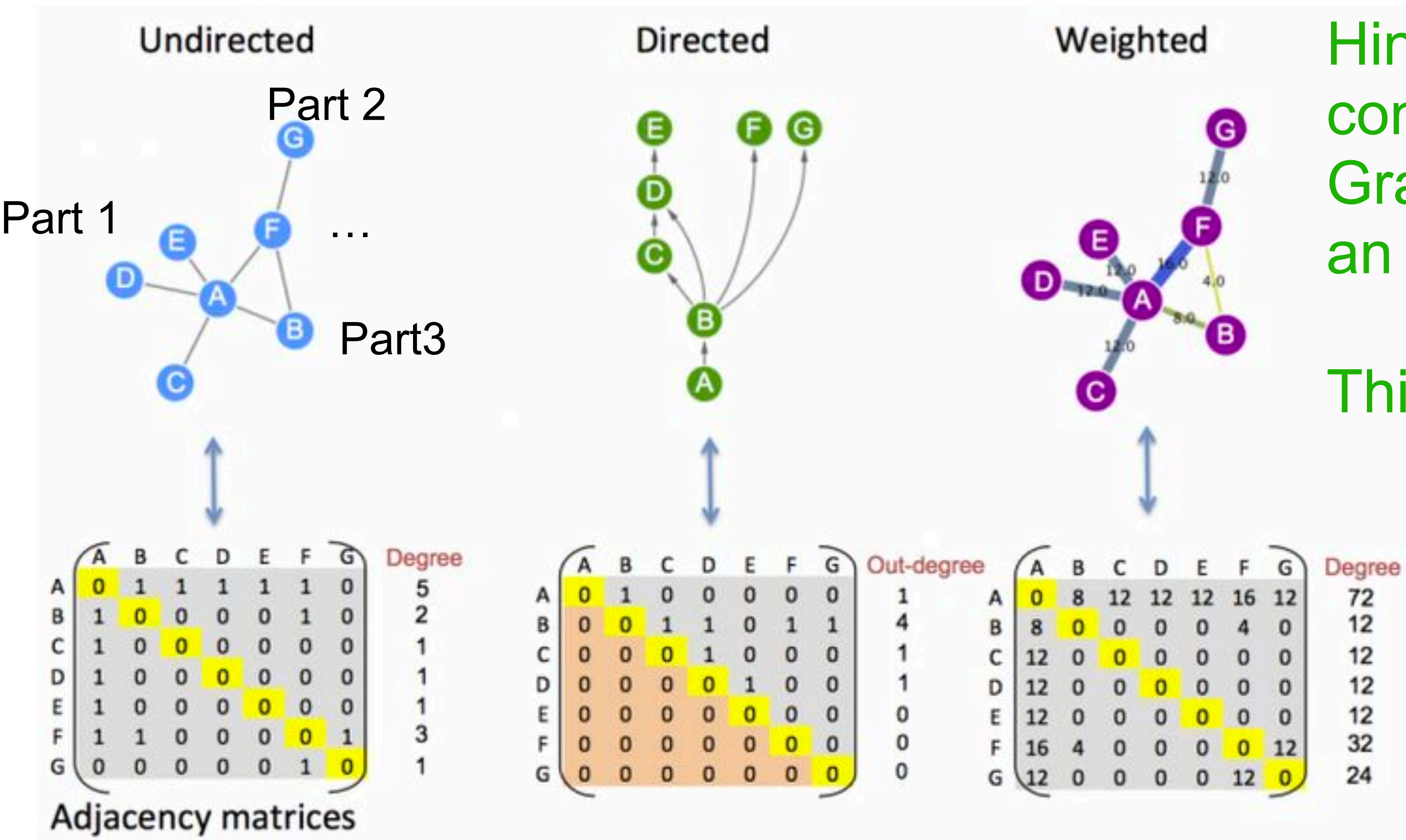
Particles Part 1 Part 2 Part 3.   Part N

Does not take into account particle Correlations

Gain from DeepCSV to DeepFlavor is from the Architecture choice

Current State of the Art

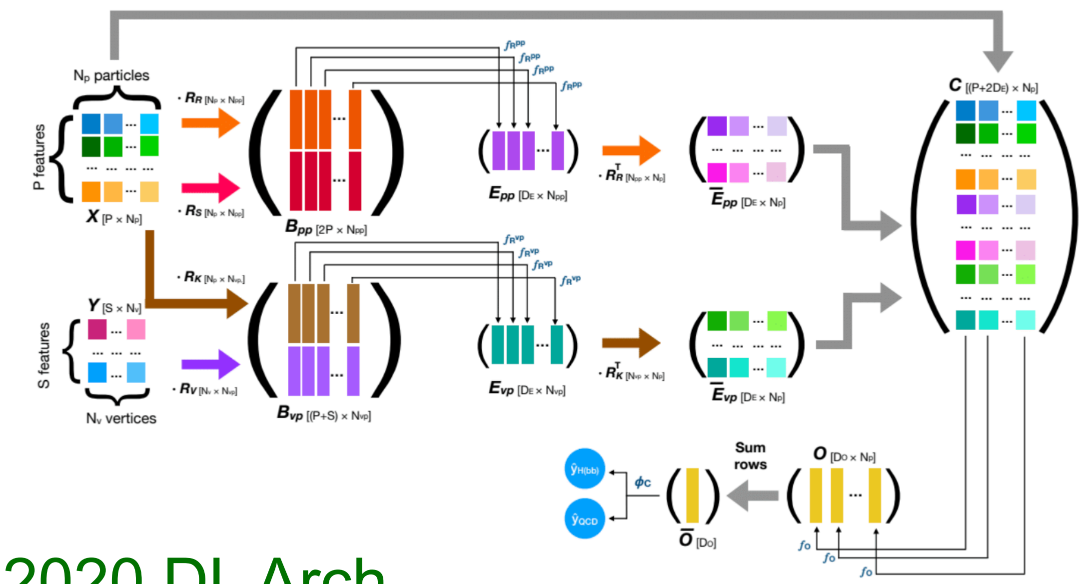
- We can take the same 4-vectors and features
  - Instead construct an NN that takes particles and correlations



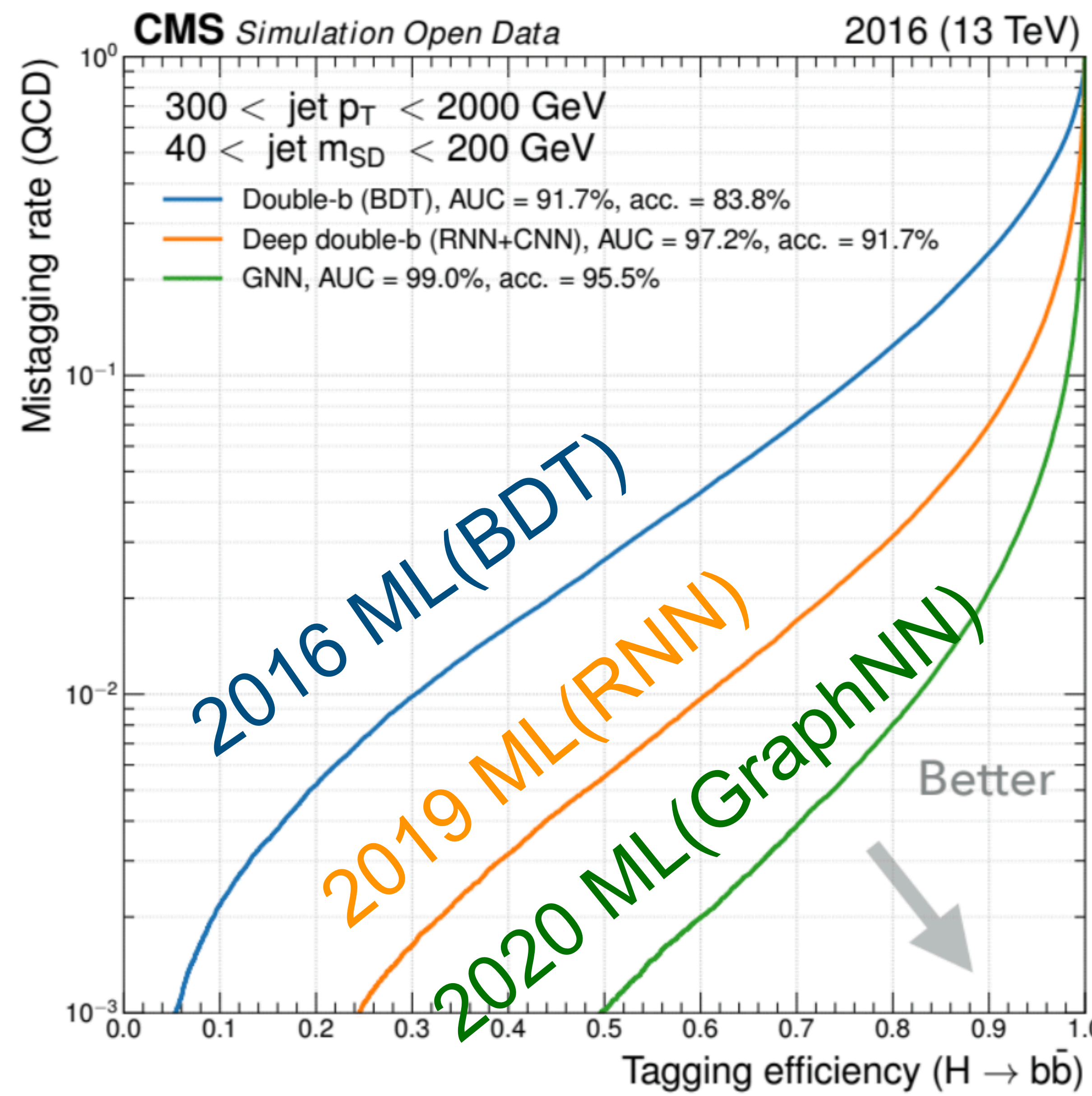
Hinges on constructing a Graph by building an adjacency matrix

This is a Graph NN

Observing Big gains



2020 DL Arch

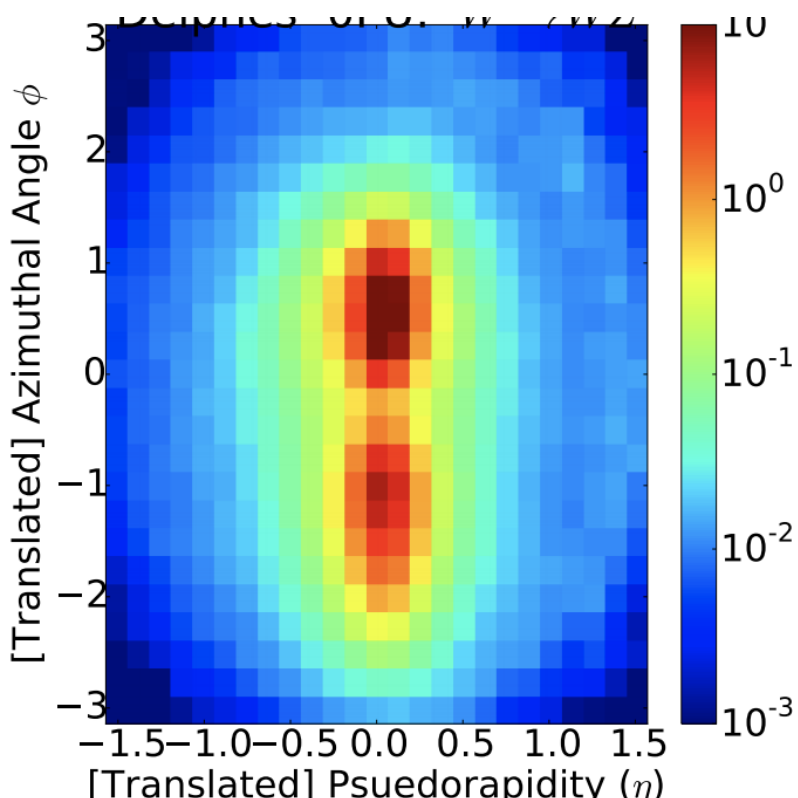


- For a Higgs boson at high energy
  - We have to rely on deep learning
- Deep learning is quickly leading to a major transformation
  - We can measure processes that we didn't think possible

arxiv:1909.12285

Encoder Progression

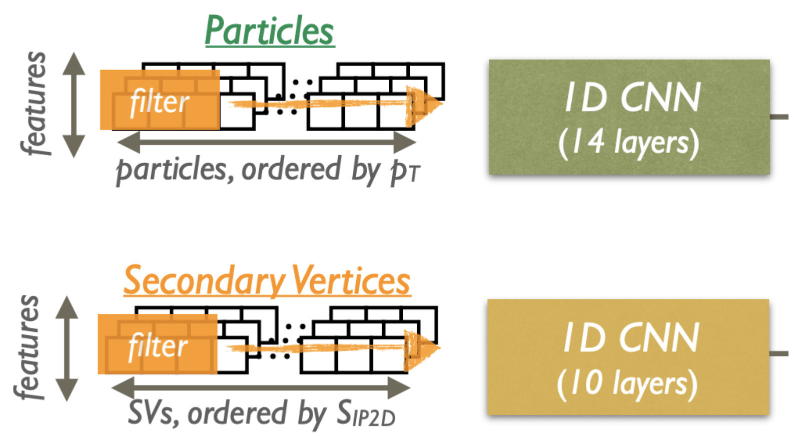
2016



Images

(not lorentz invariant)

2018

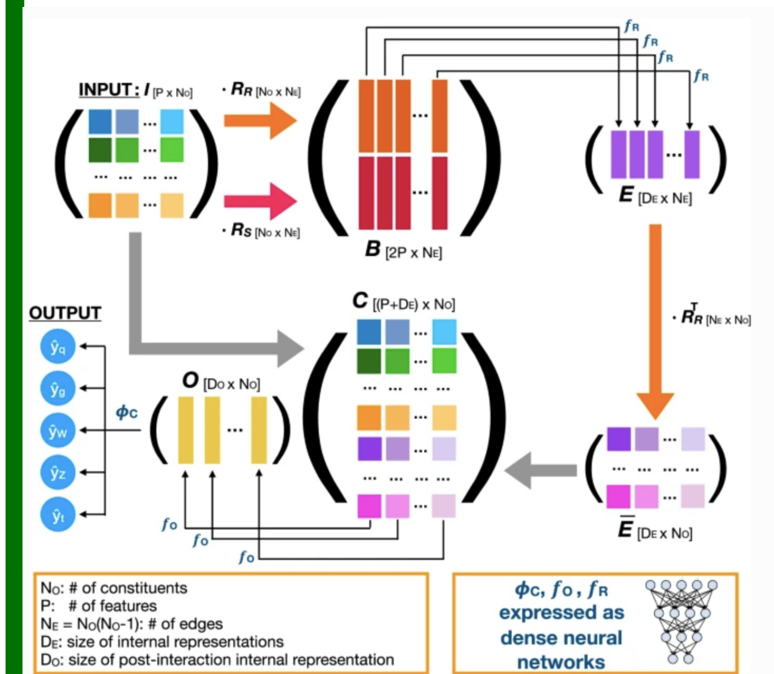


Particles and SVs
with 4-vectors+features

Particles

(limited correlations)

2020



Graphs

(Particles+correlations)

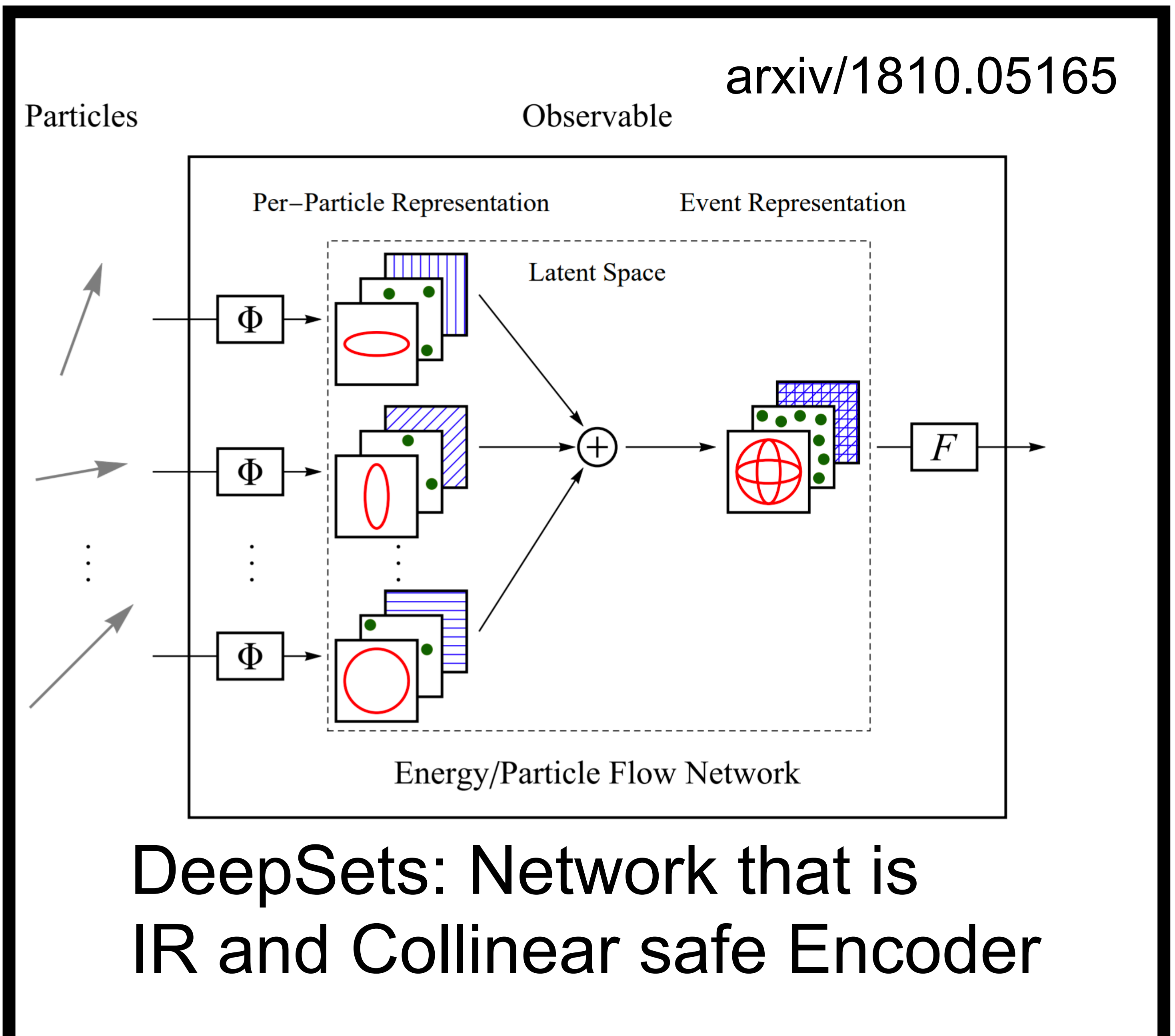
Current collaboration results

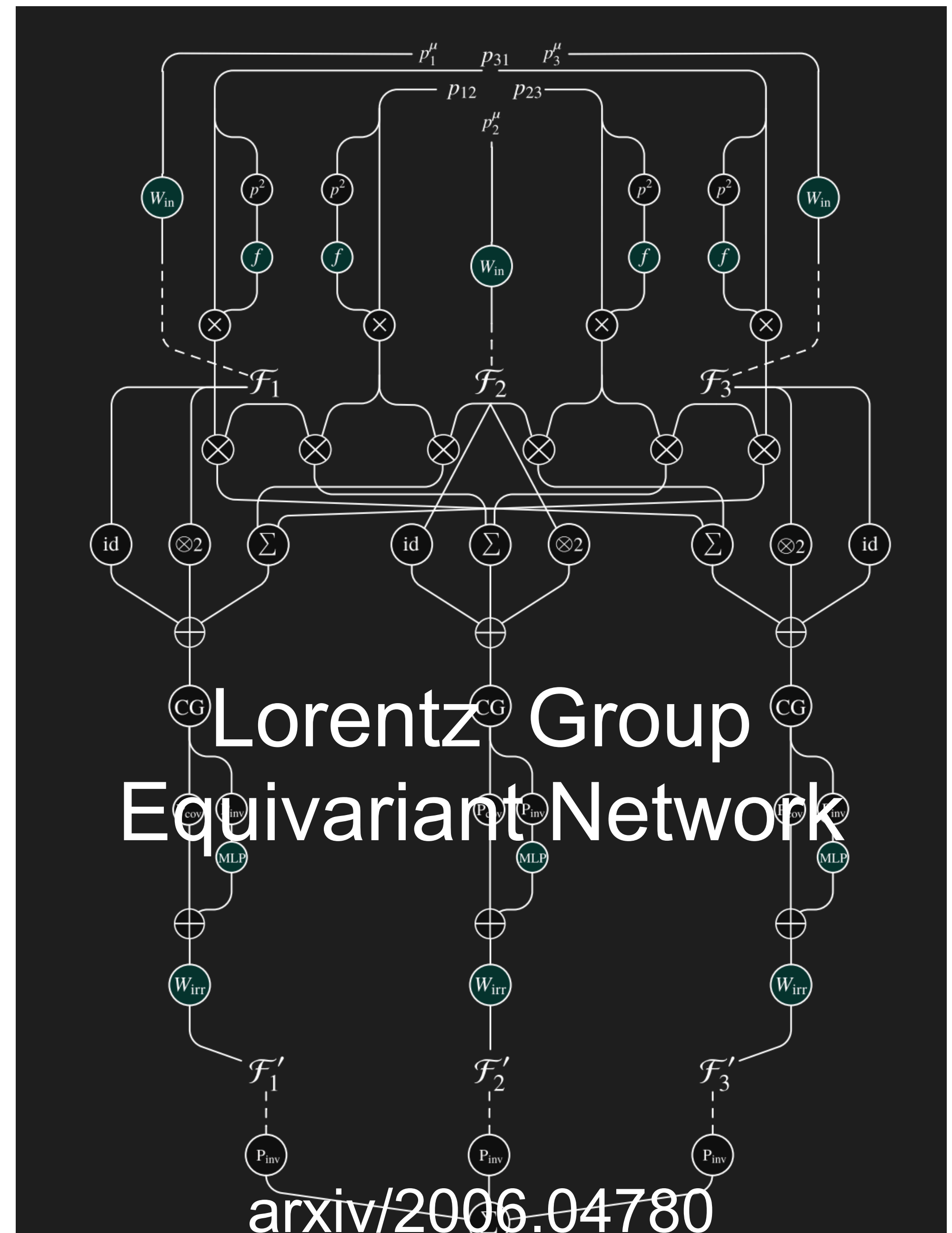
Expected Results Soon!

Progressively moving towards use of more info

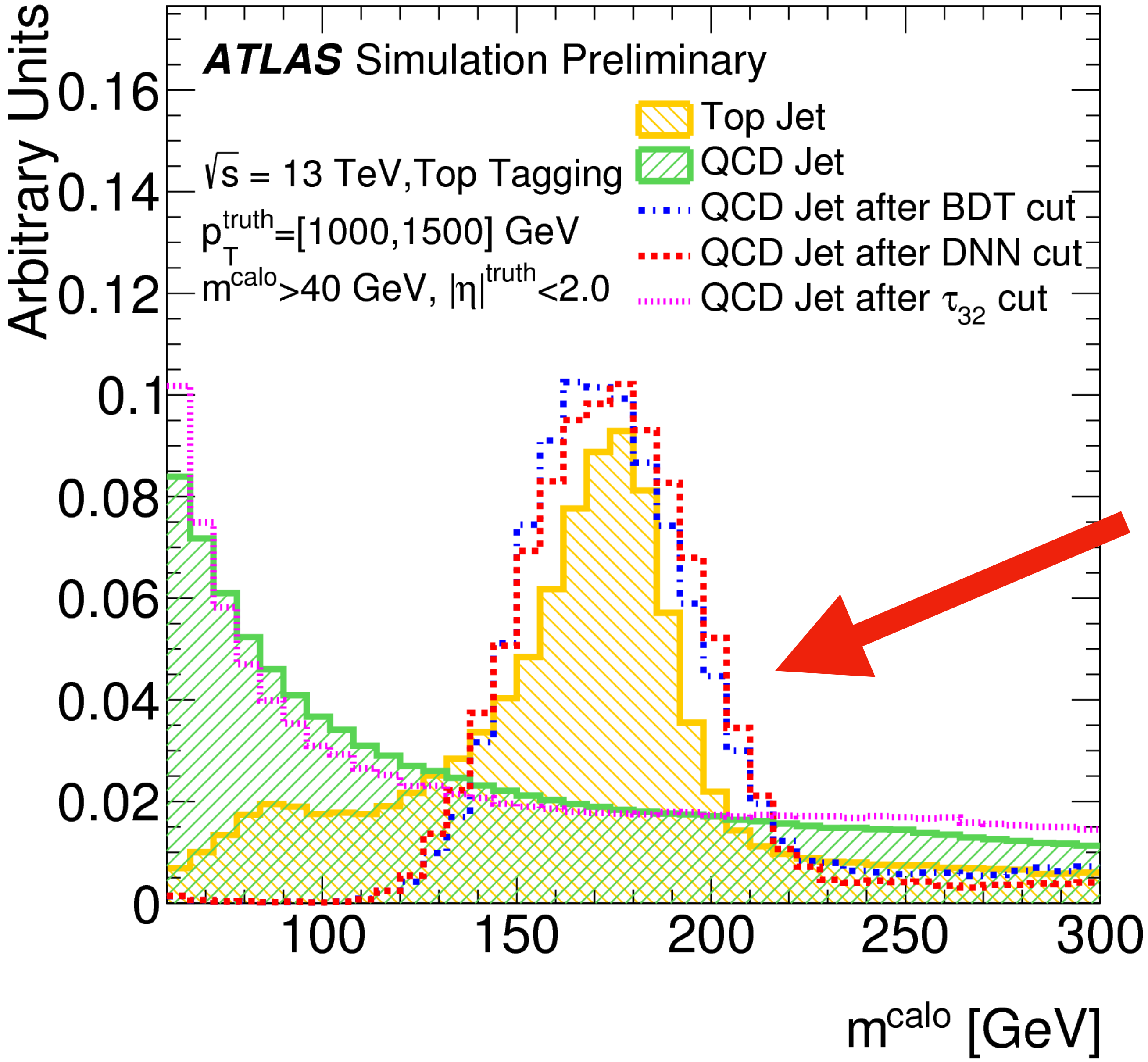
Extensions of these Ideas

- There are many ways to make encoders better





Finding a resonance

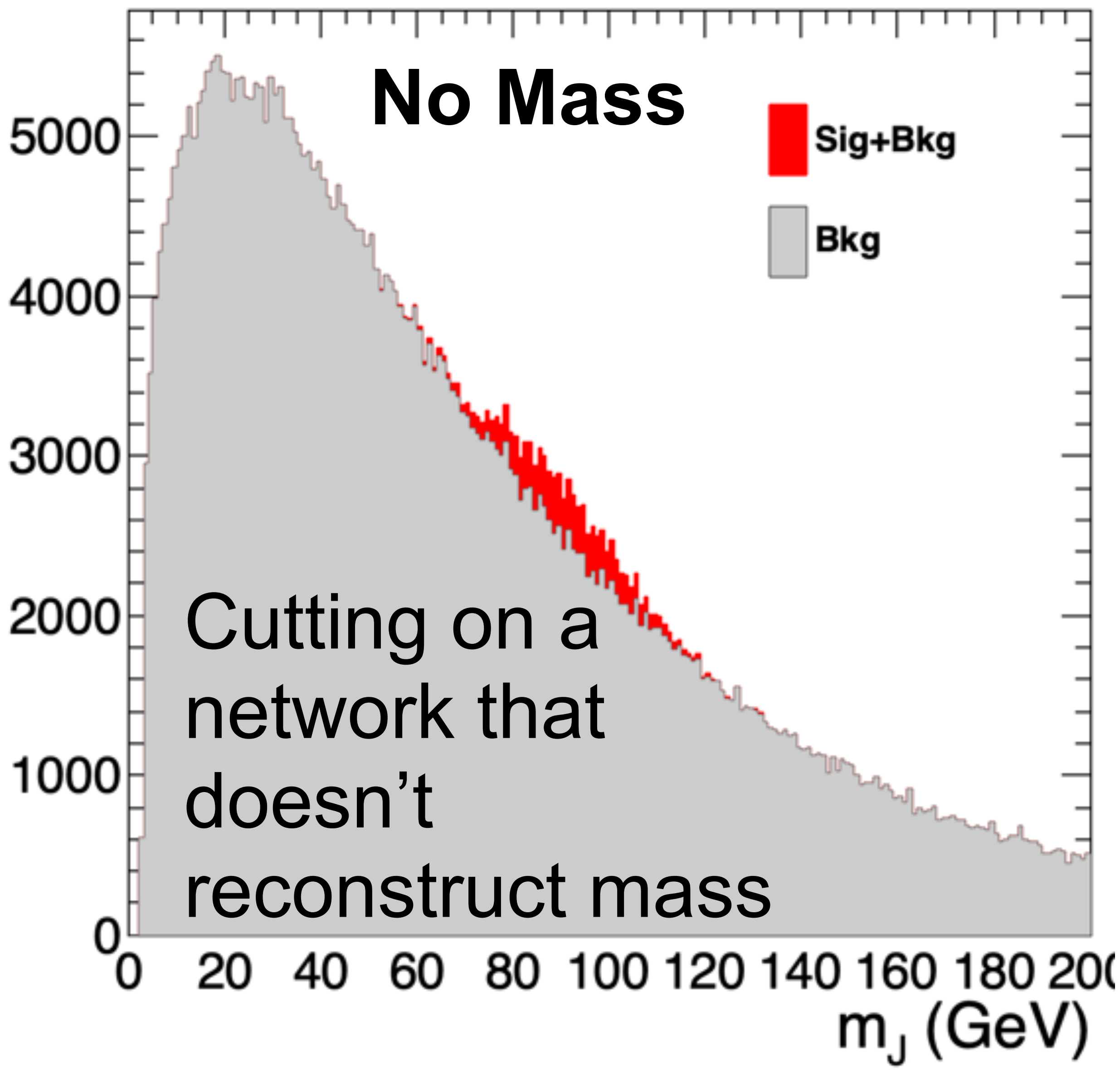


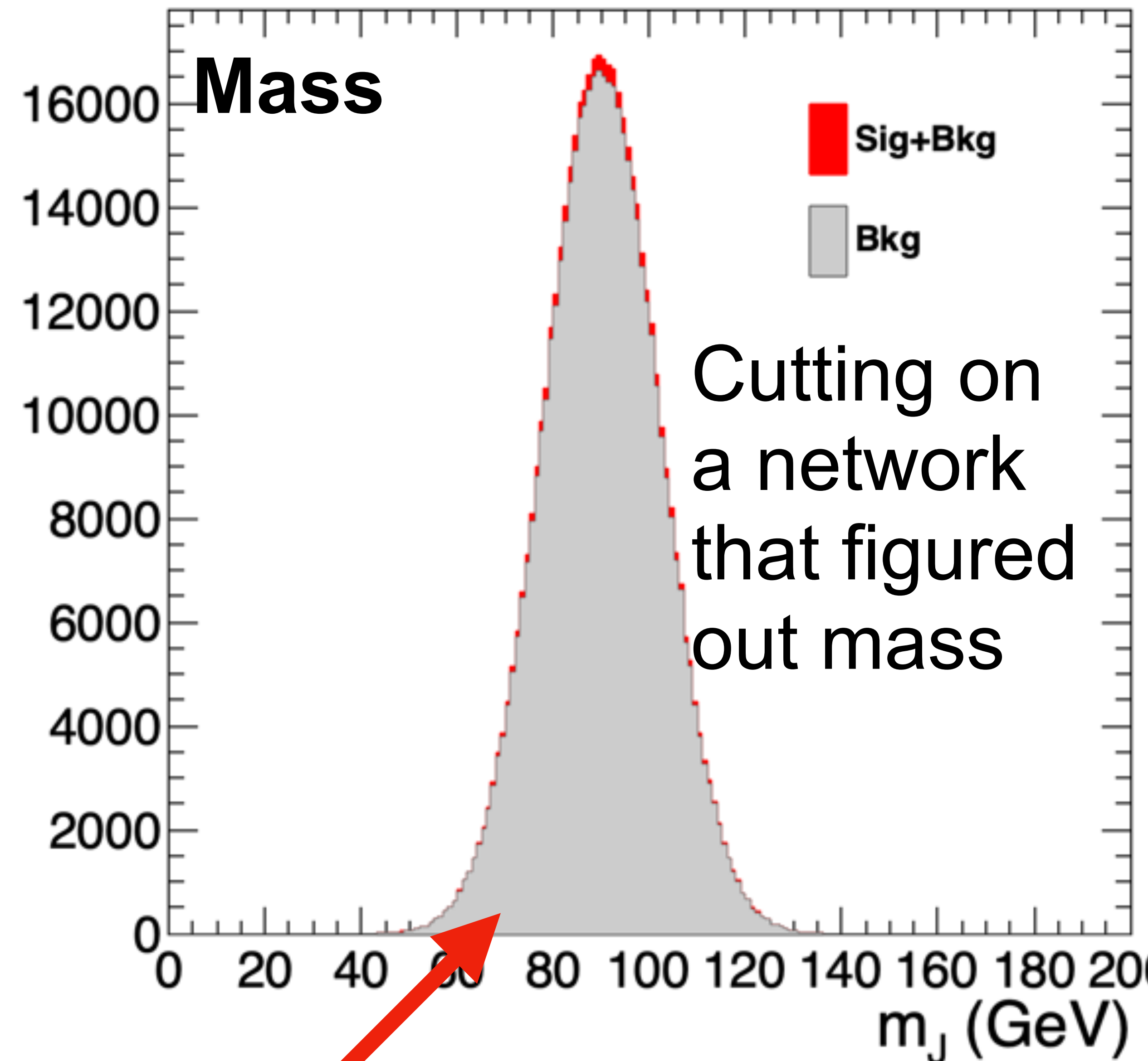
Selecting on a well trained Neural Network

Network will reconstruct mass

- To find a resonance, we don't just need a good DNN
  - We also need a way to extract it

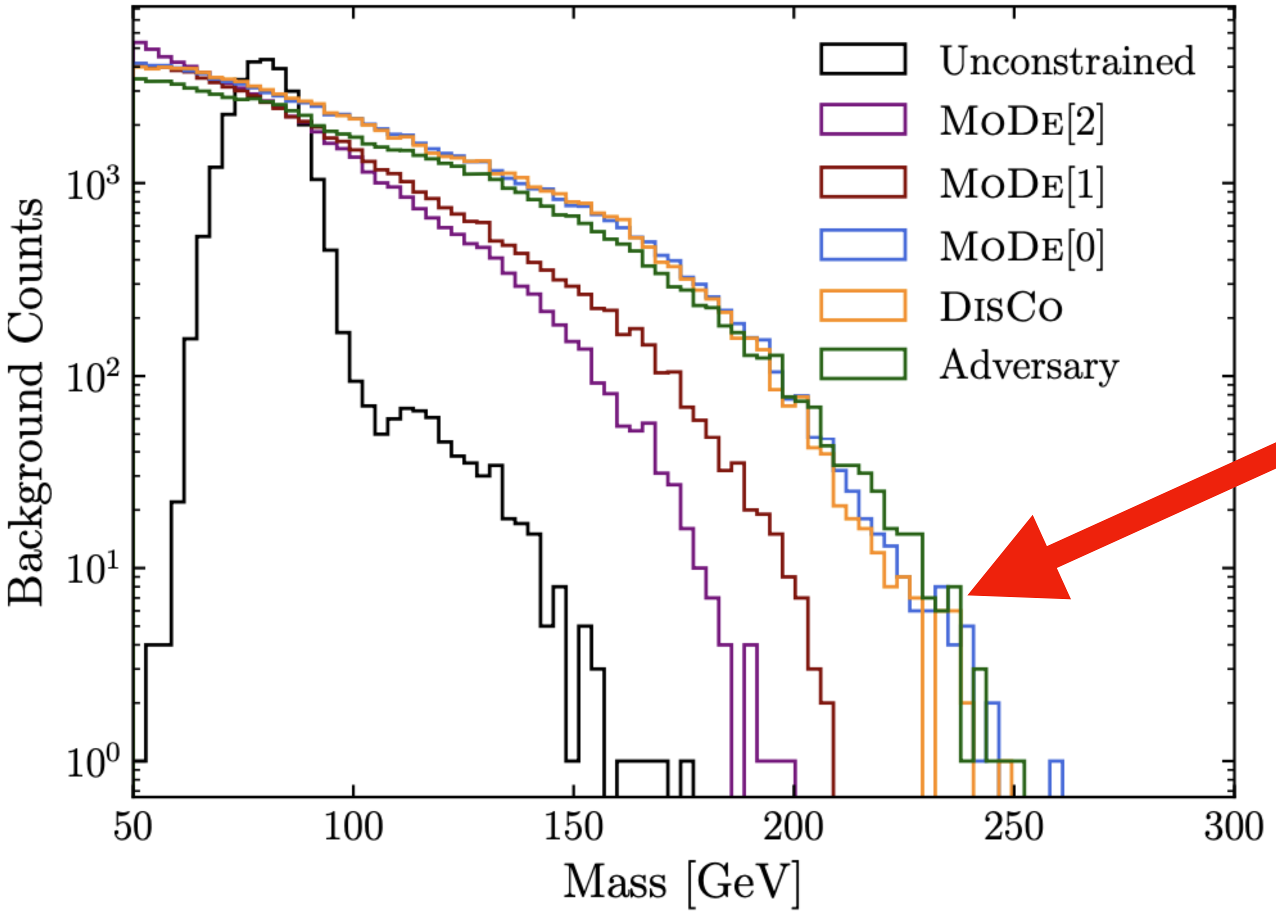
Finding a resonance





- You can't find a bump on a bump!
  - Being able to control background is essential in data

One Method To Control

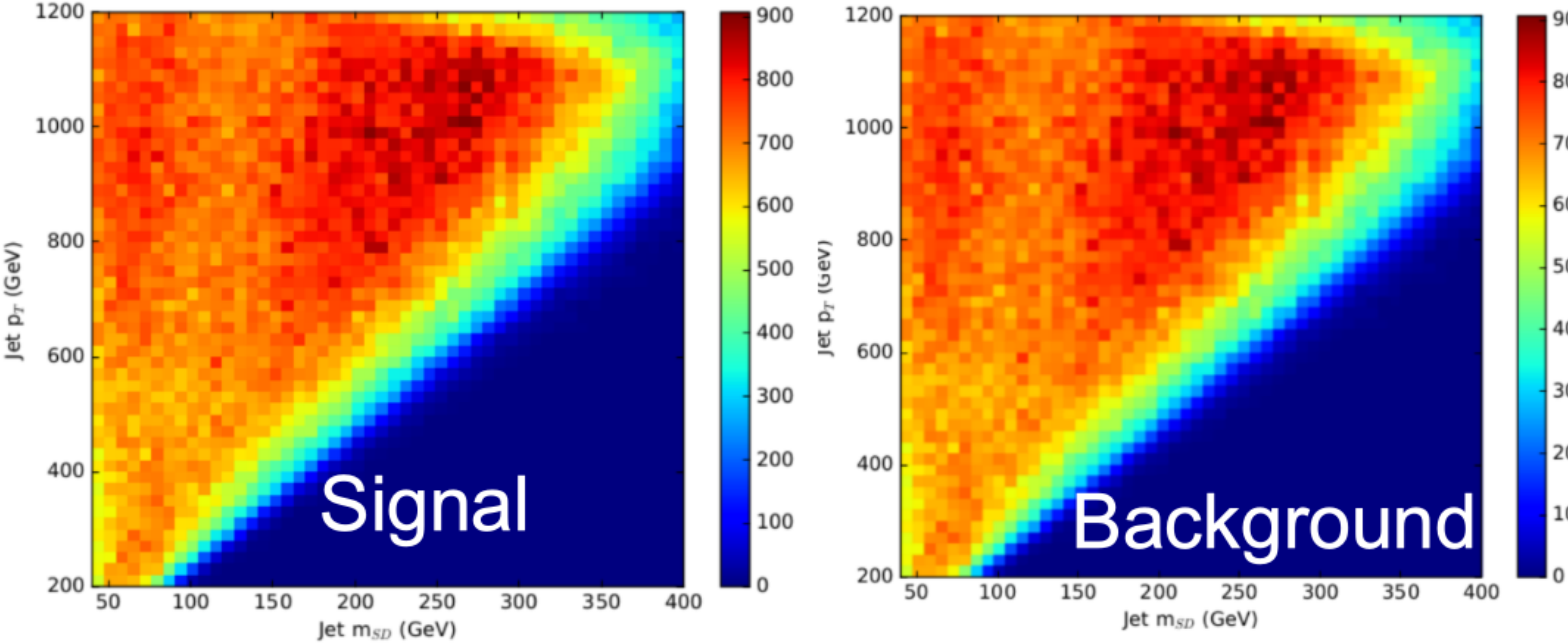


Invent a way to penalize the NN so that it can't reconstruct mass

New Loss = Loss + Penalty term

- Adding a penalty can force the network to go the other way
- This requires a bit of tuning
- However there is lots of literature doing this

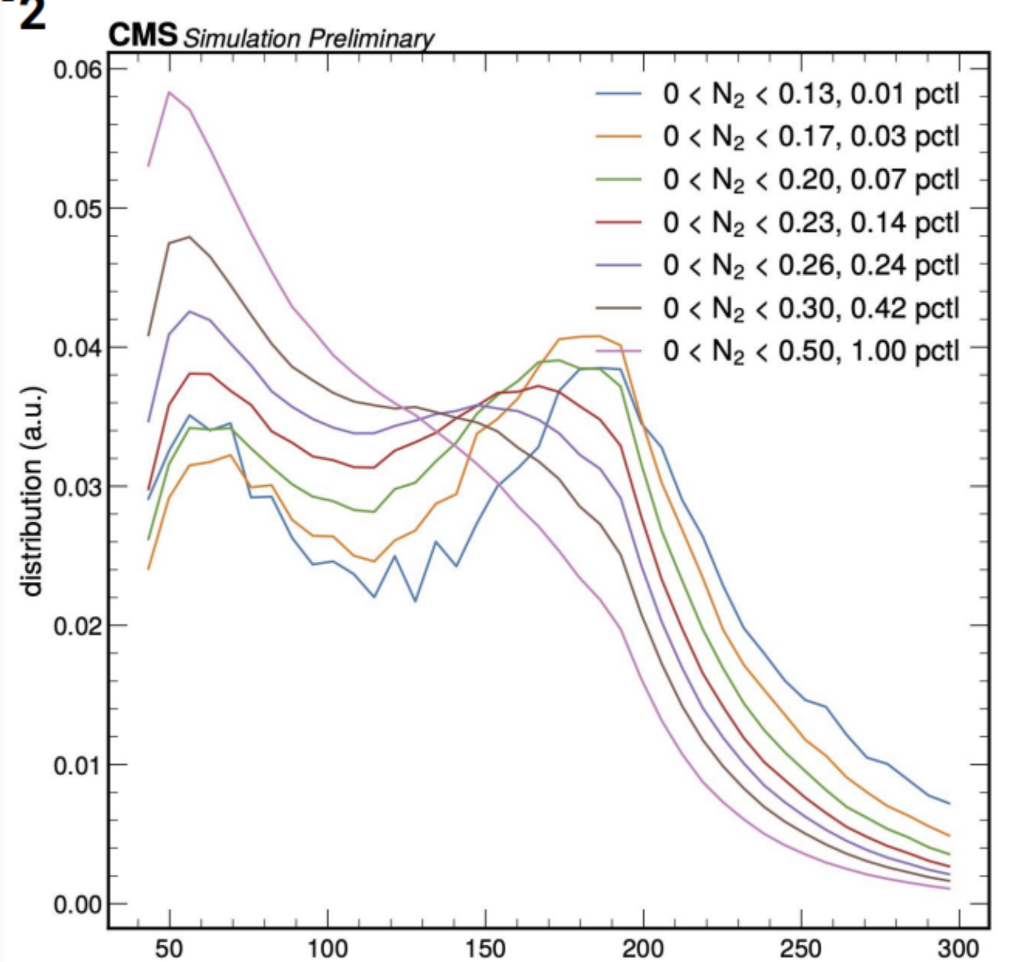
A More Robust Approach



- Modifying Matrix elements so signal and background are the same
- This solution turns out to be very powerful, but "Old School"

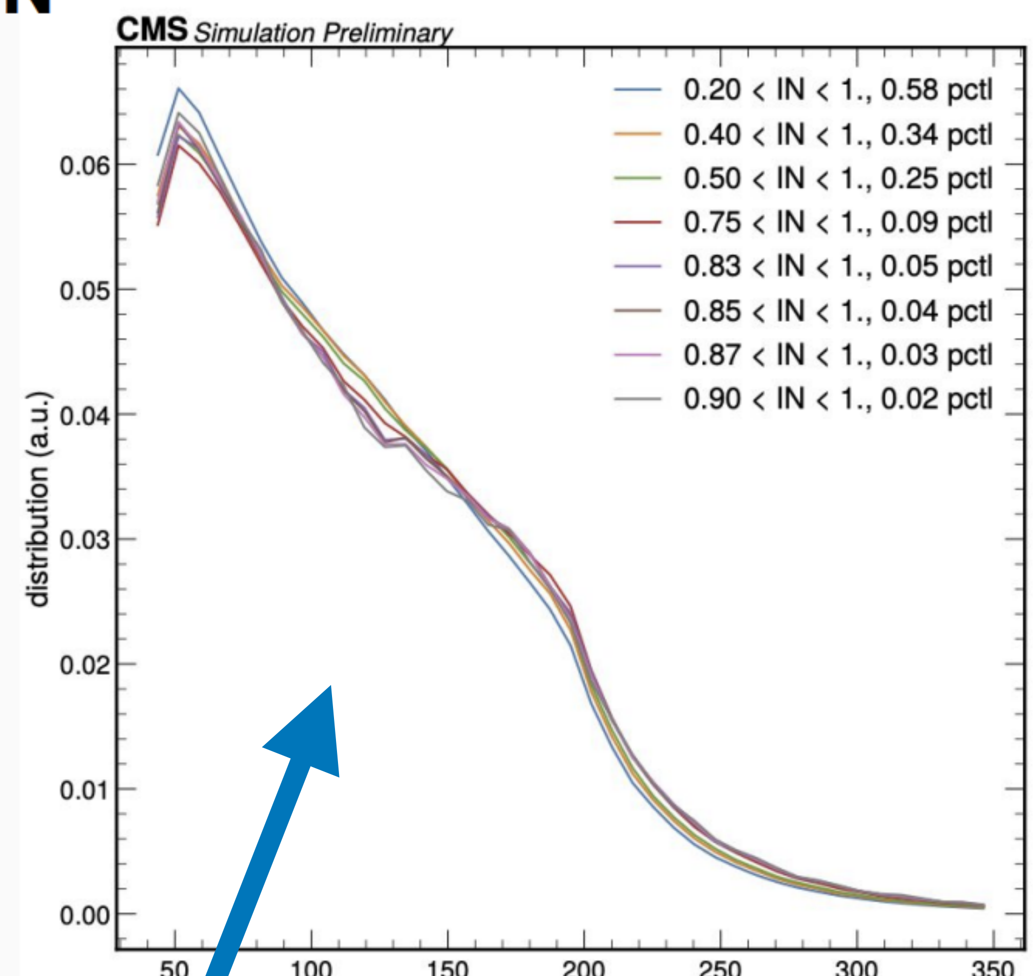
A More Robust Approach

N_2



Jet Mass (Gev)

IN

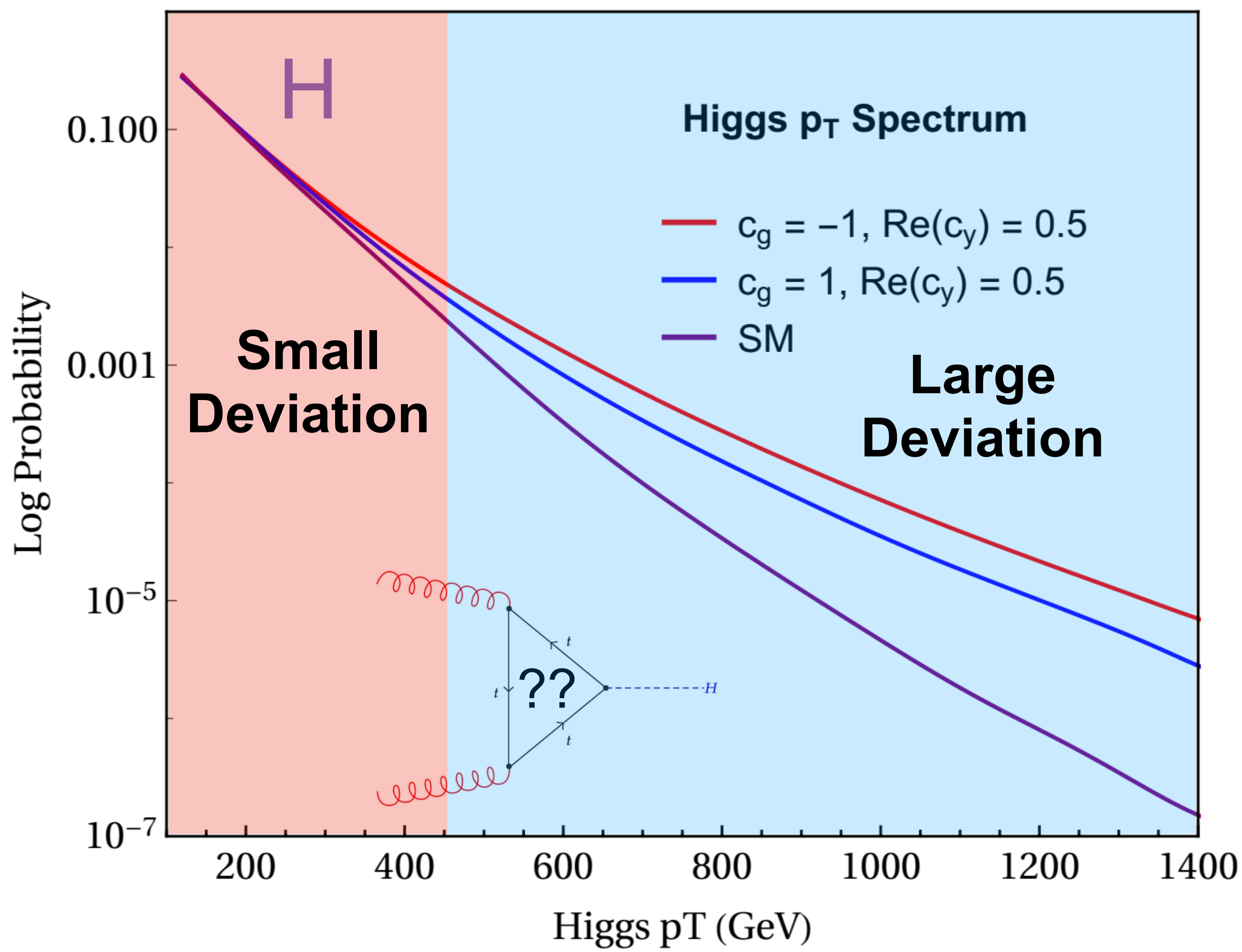


Jet Mass (Gev)

No Variation with the cut!

Boosted Higgs Result

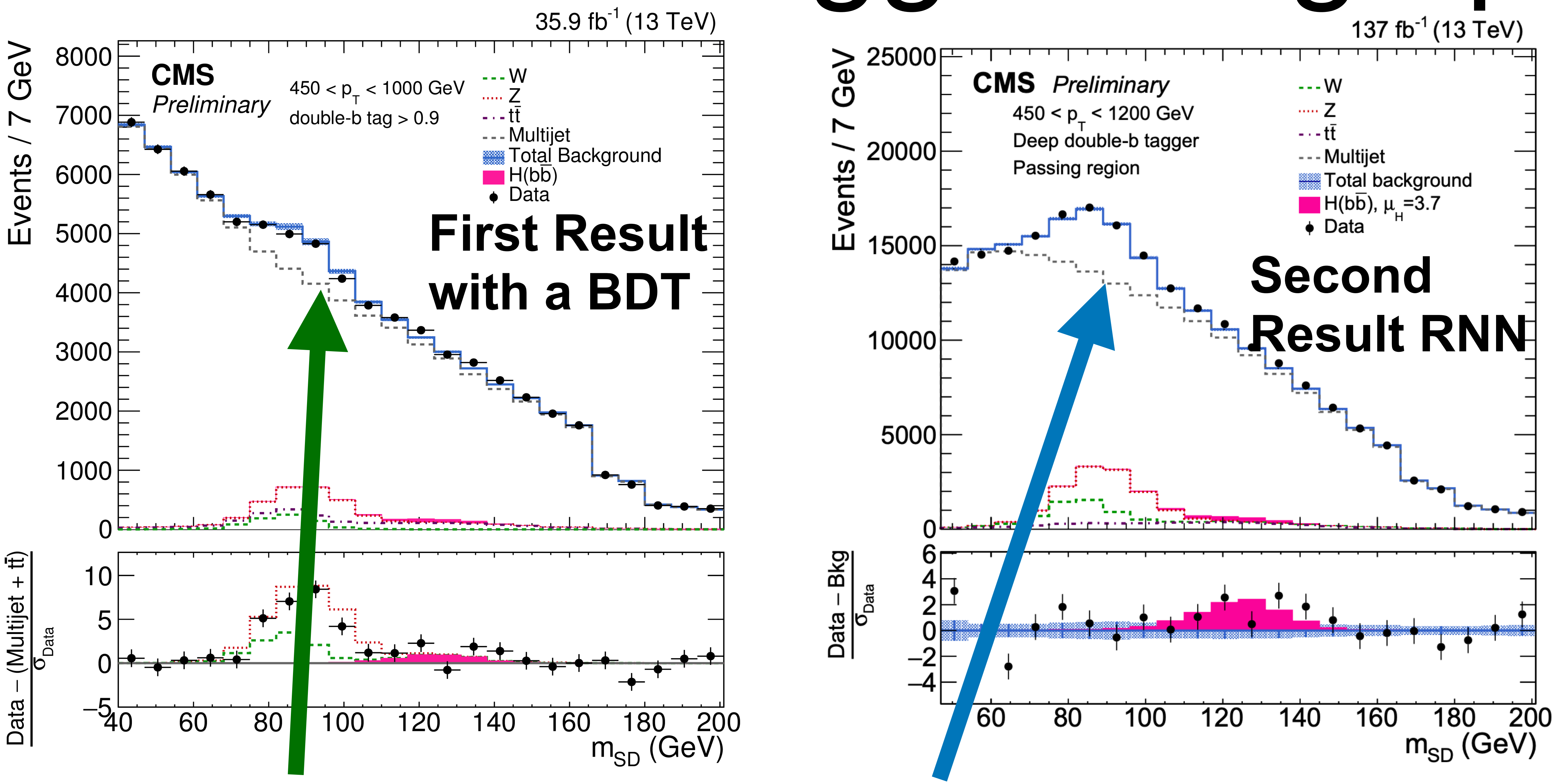
Can we build a new Higgs boson result with deep learning?



Deep learning is effective at isolating overlapping b-quarks

With deep learning we were able to reduce background by 2

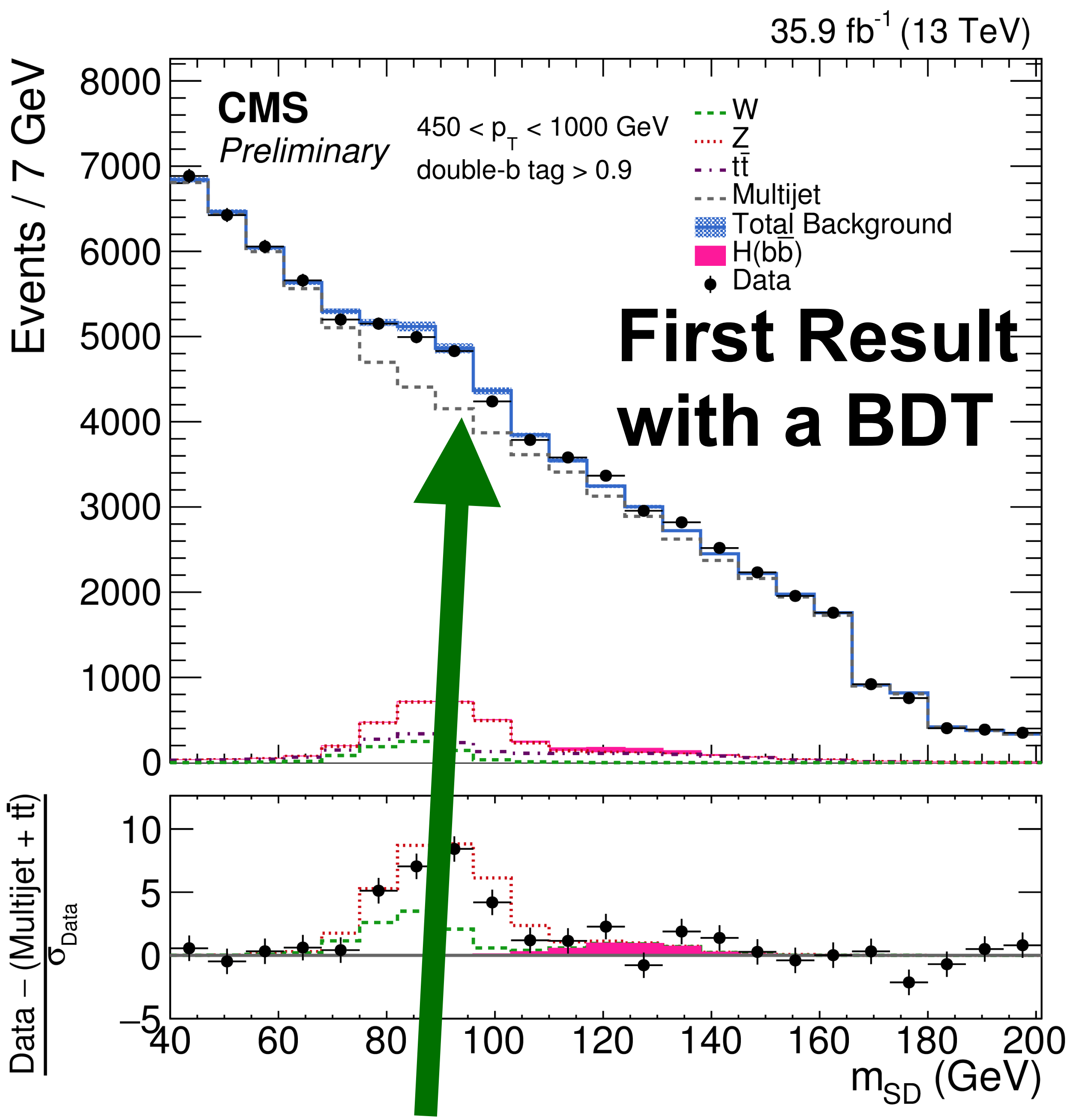
Higgs at high p_T

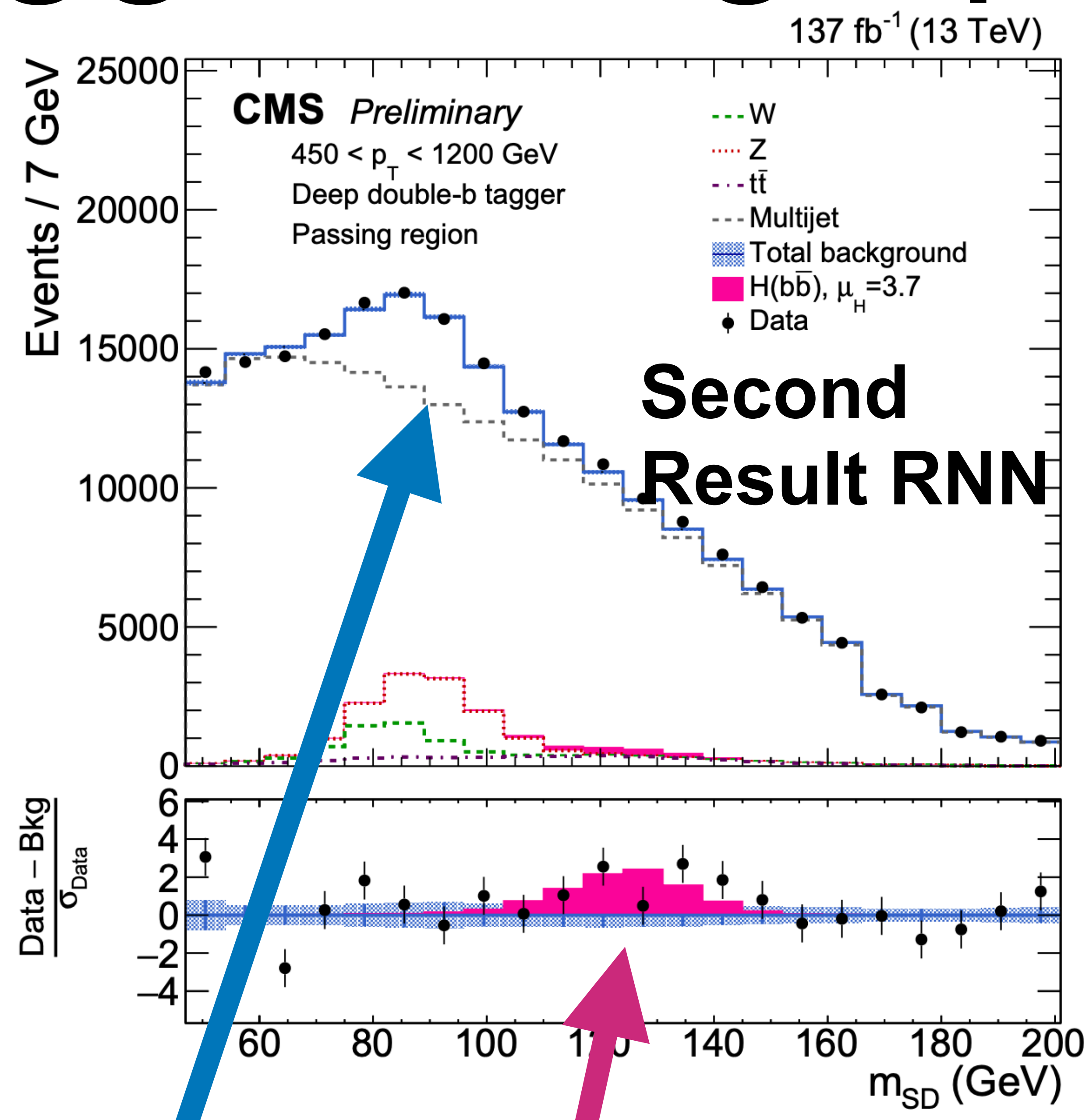


Only with an ML Algorithm can we see. Z→bb

With an NN peak is dramitically larger

Higgs at high p_T





Only with an ML Algorithm can we see. Z→bb

With an NN peak is dramitically larger

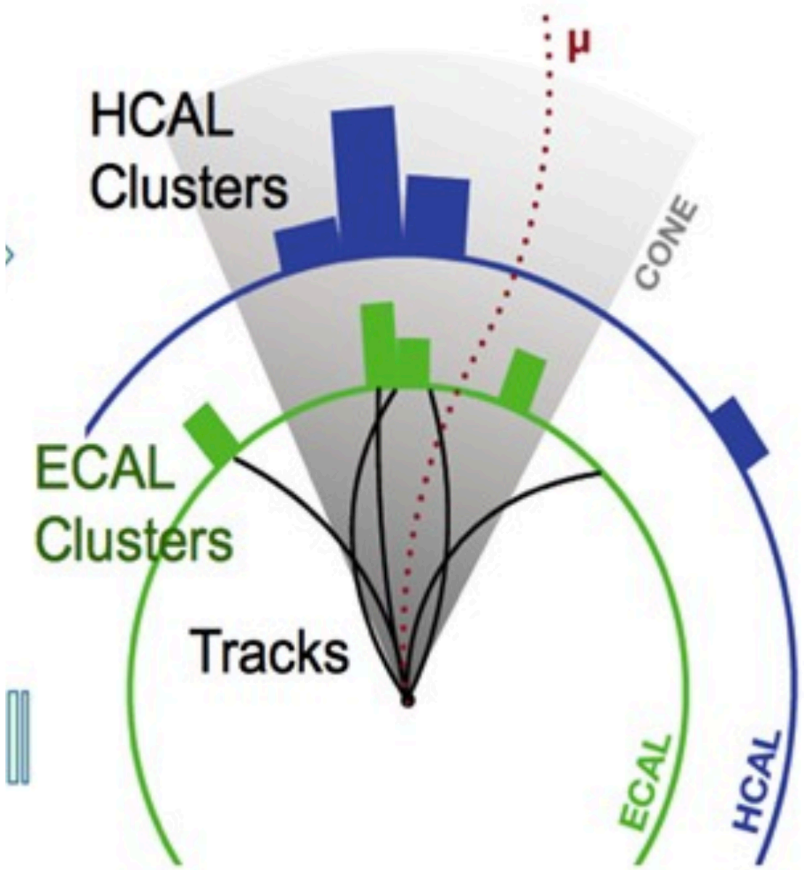
And there are signs of a Higgs Peak

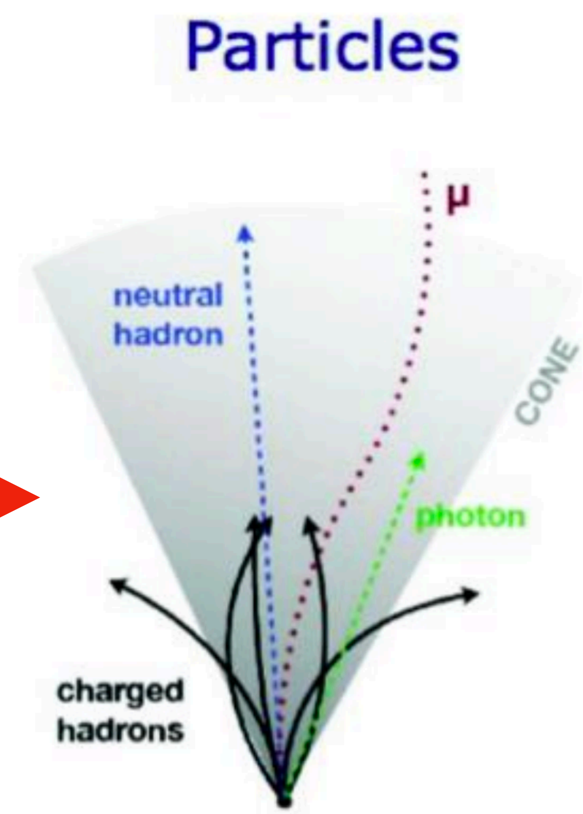
What LHC physicists have been doing during COVID

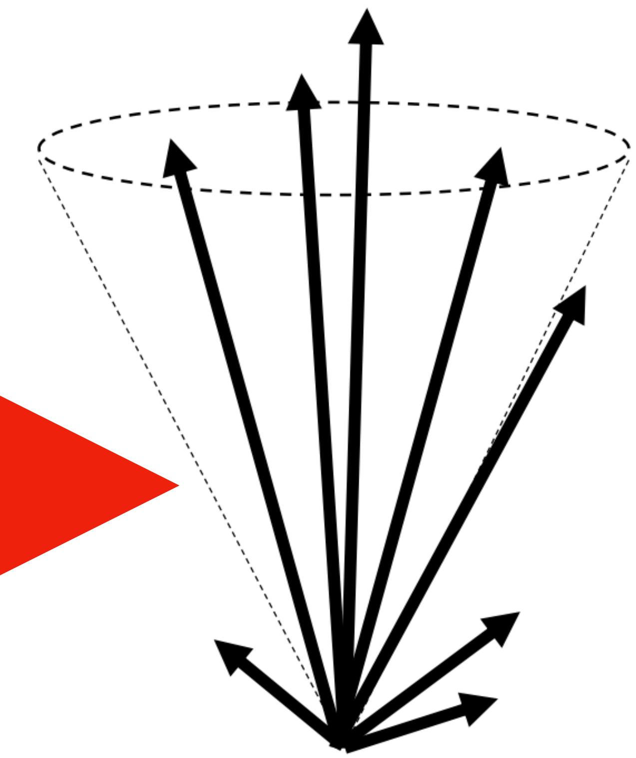
Deep Learning Evolution

Reconstruction flow

quark/gluom
aka Jet
(cluster of particles)



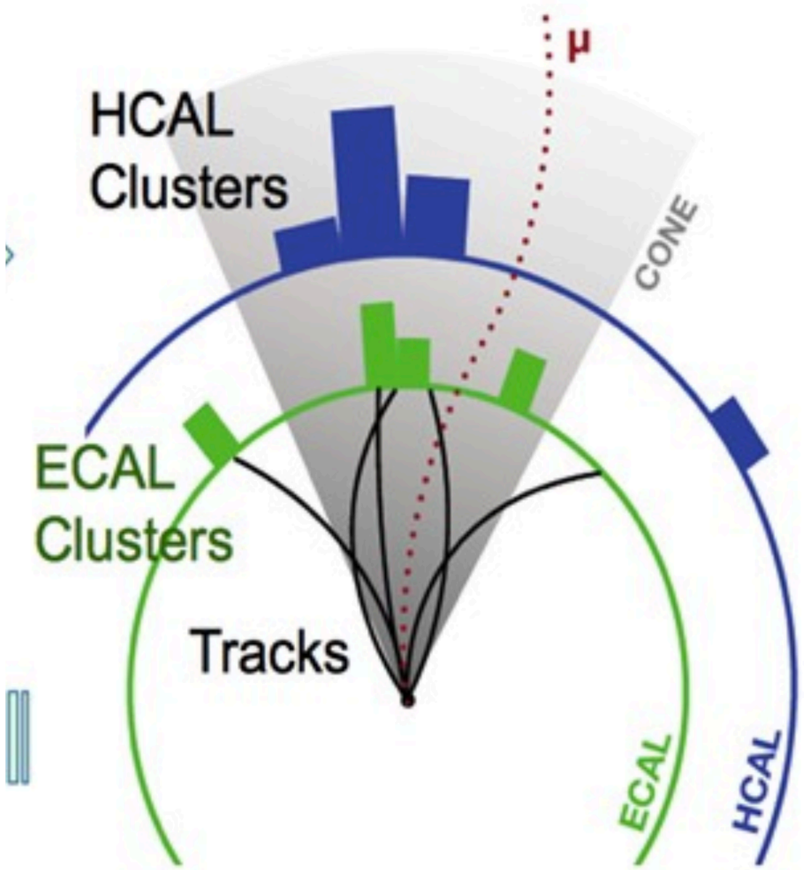


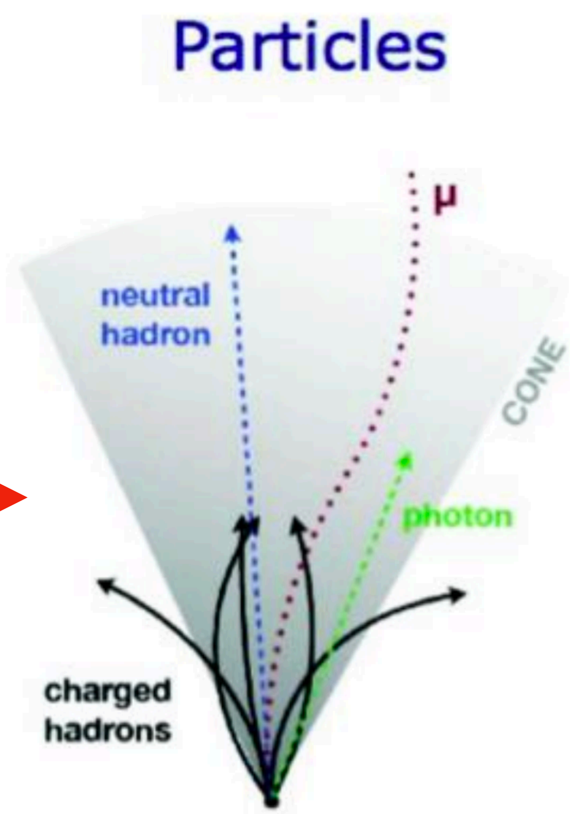


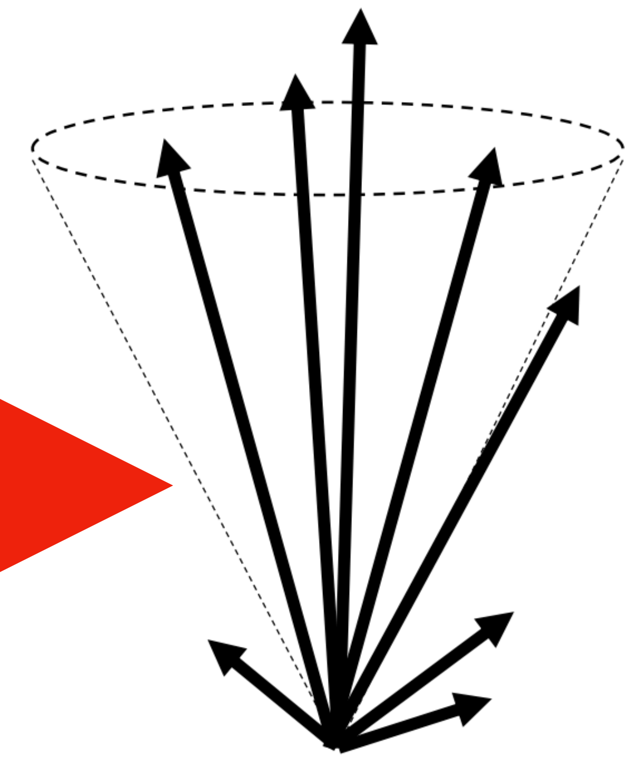
Deep Learning Evolution

Reconstruction flow

quark/gluom
aka Jet
(cluster of particles)





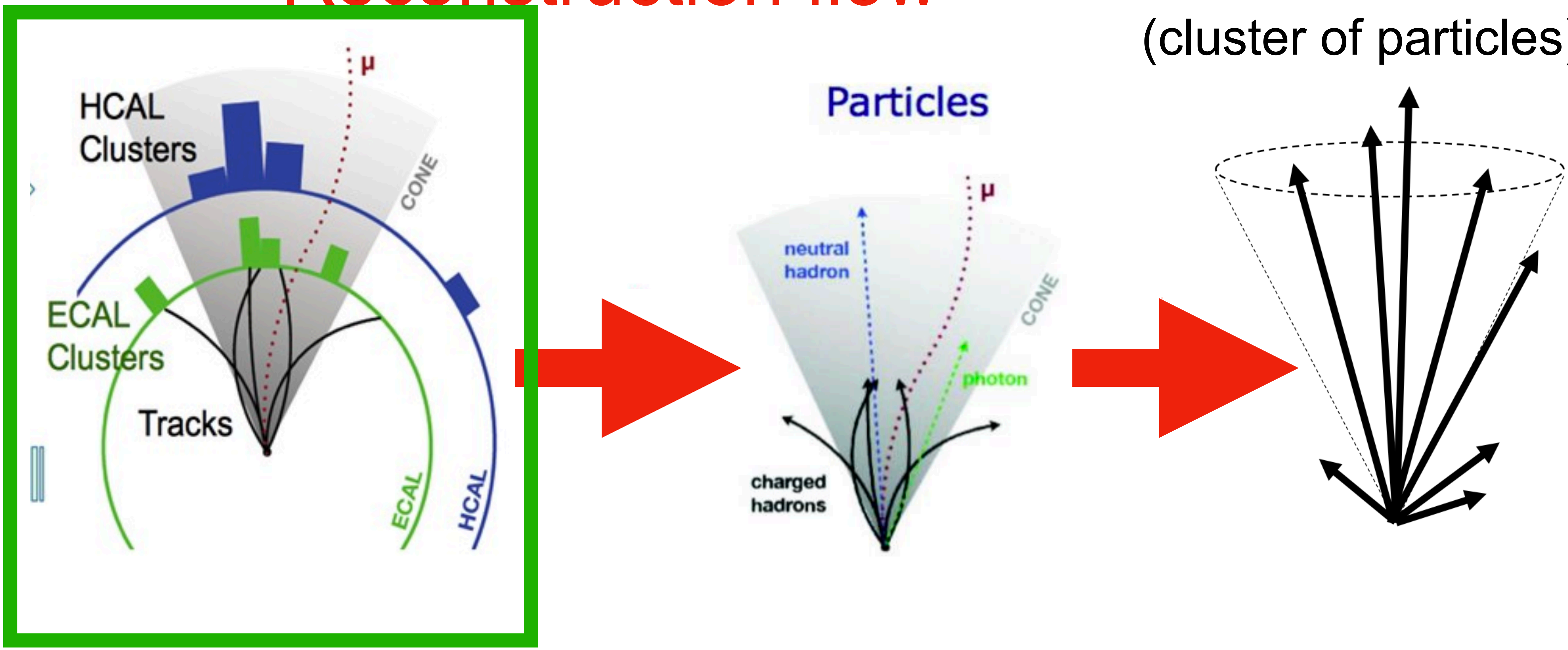


Deep Learning Migration

Deep Learning Evolution

Reconstruction flow

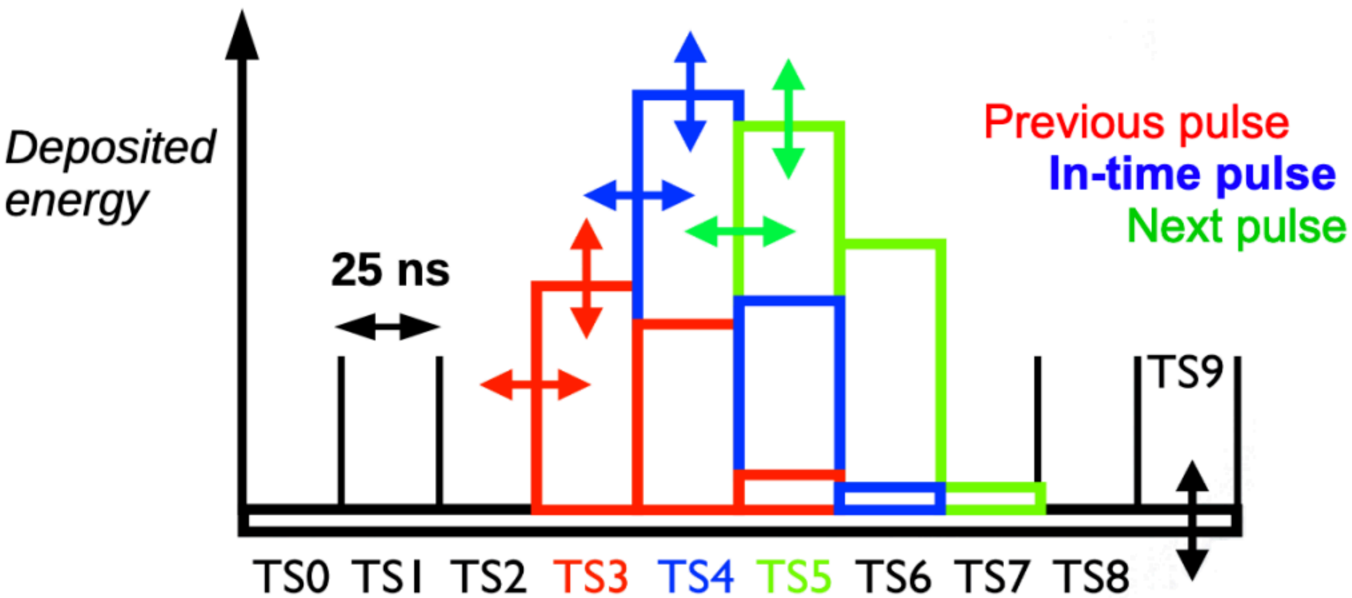
quark/gluom
aka Jet
(cluster of particles)

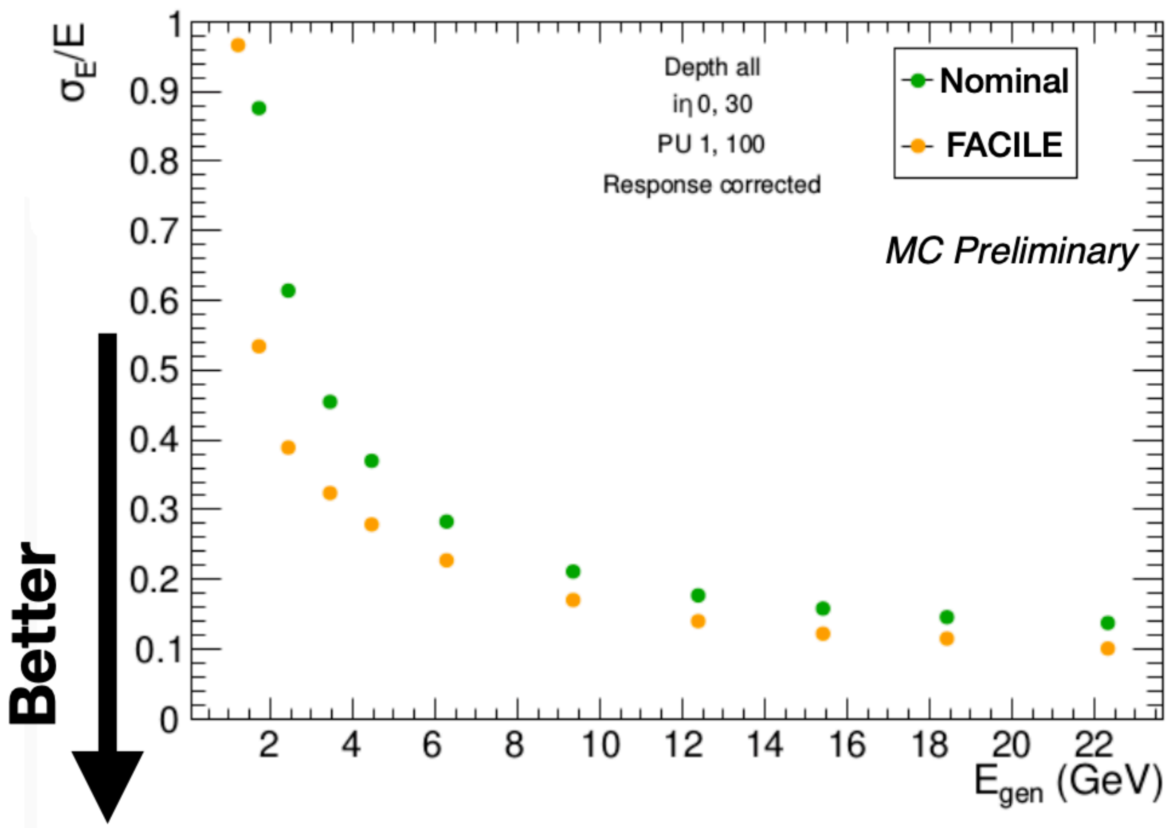


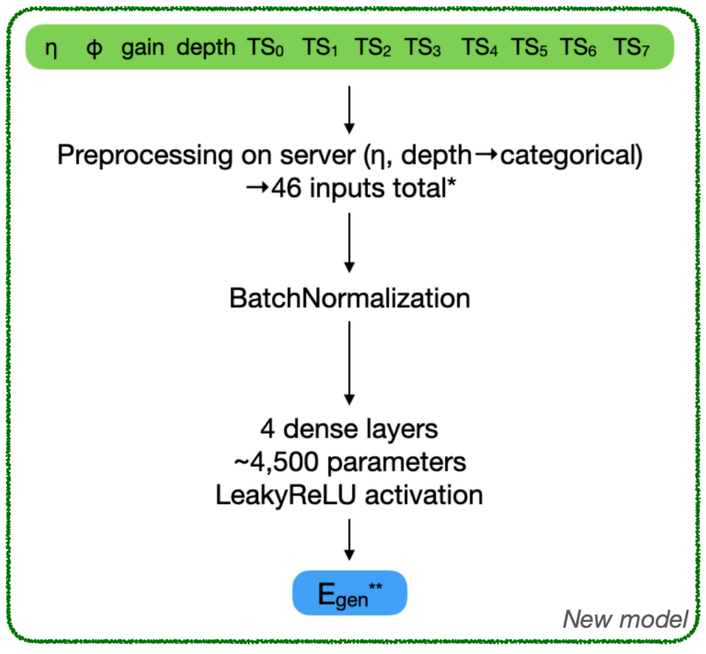
Challenge:
Can you go from Raw inputs to reco?

Simple Example

- Reconstructing a single calorimeter tower
  - FACILE Algorithm: Reconstruct integral of in-time pulse
  - Up to 5 overlapping pulse





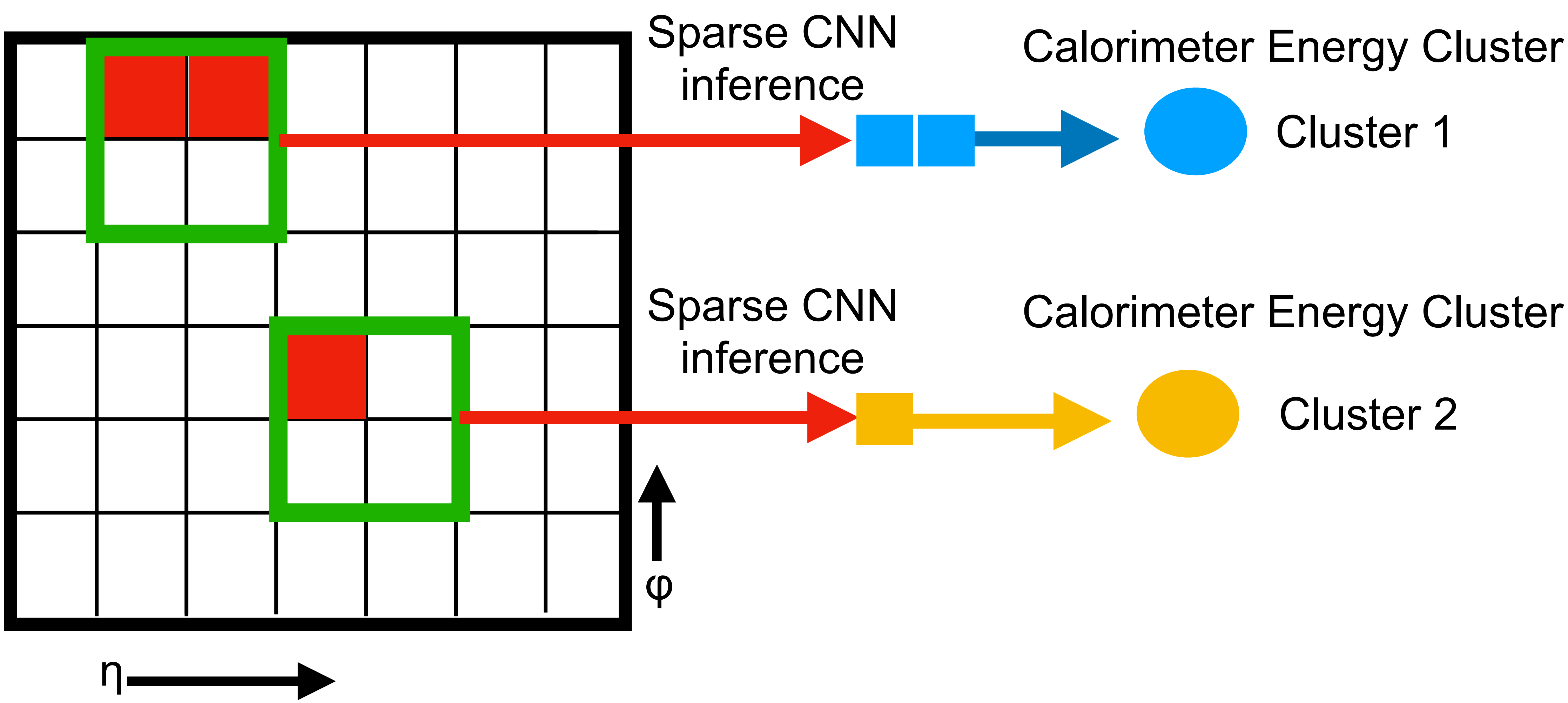


Simple NN
Can run fast

LeakyRelu
Critical to regression

From Single to Collection

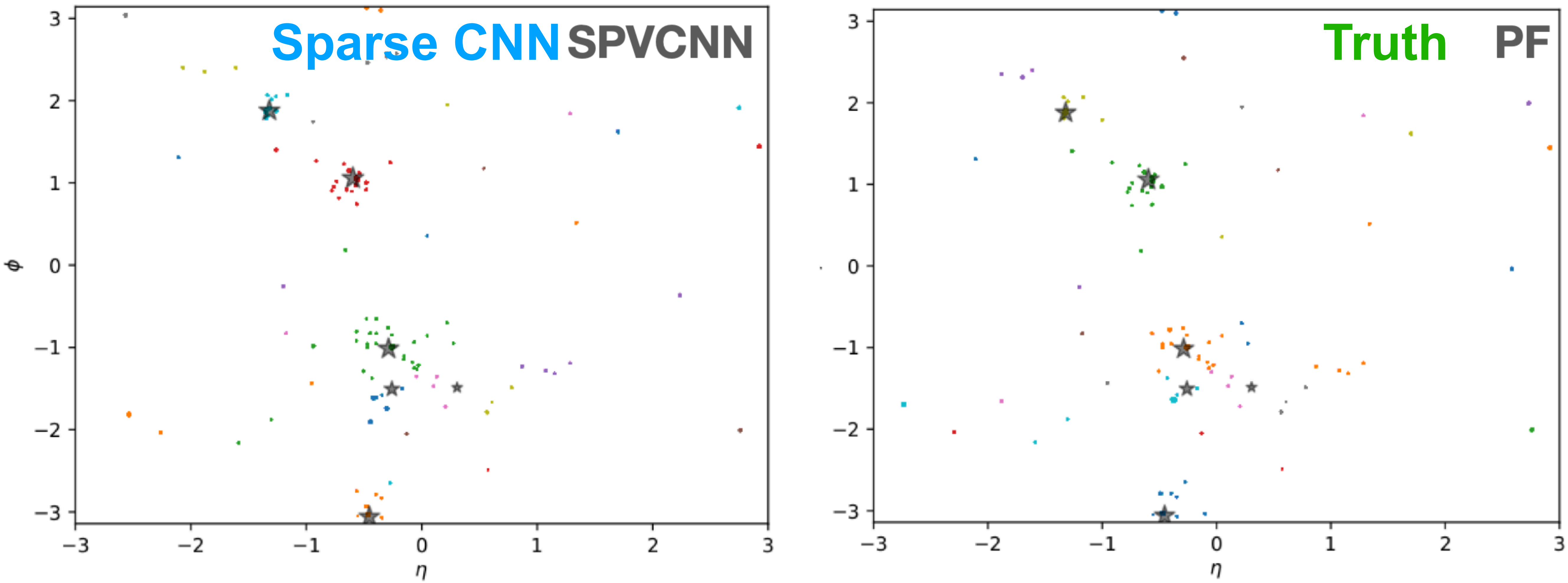
- Facile runs reconstruction on a single channel
  - We can envision an algorithm that takes in all channels
  - One way is to use a sparse CNN for graph-like inputs



By taking the grid geometry of calorimeter can deploy Sparse CNN to Infer whole calo at once

Can compare to Reality

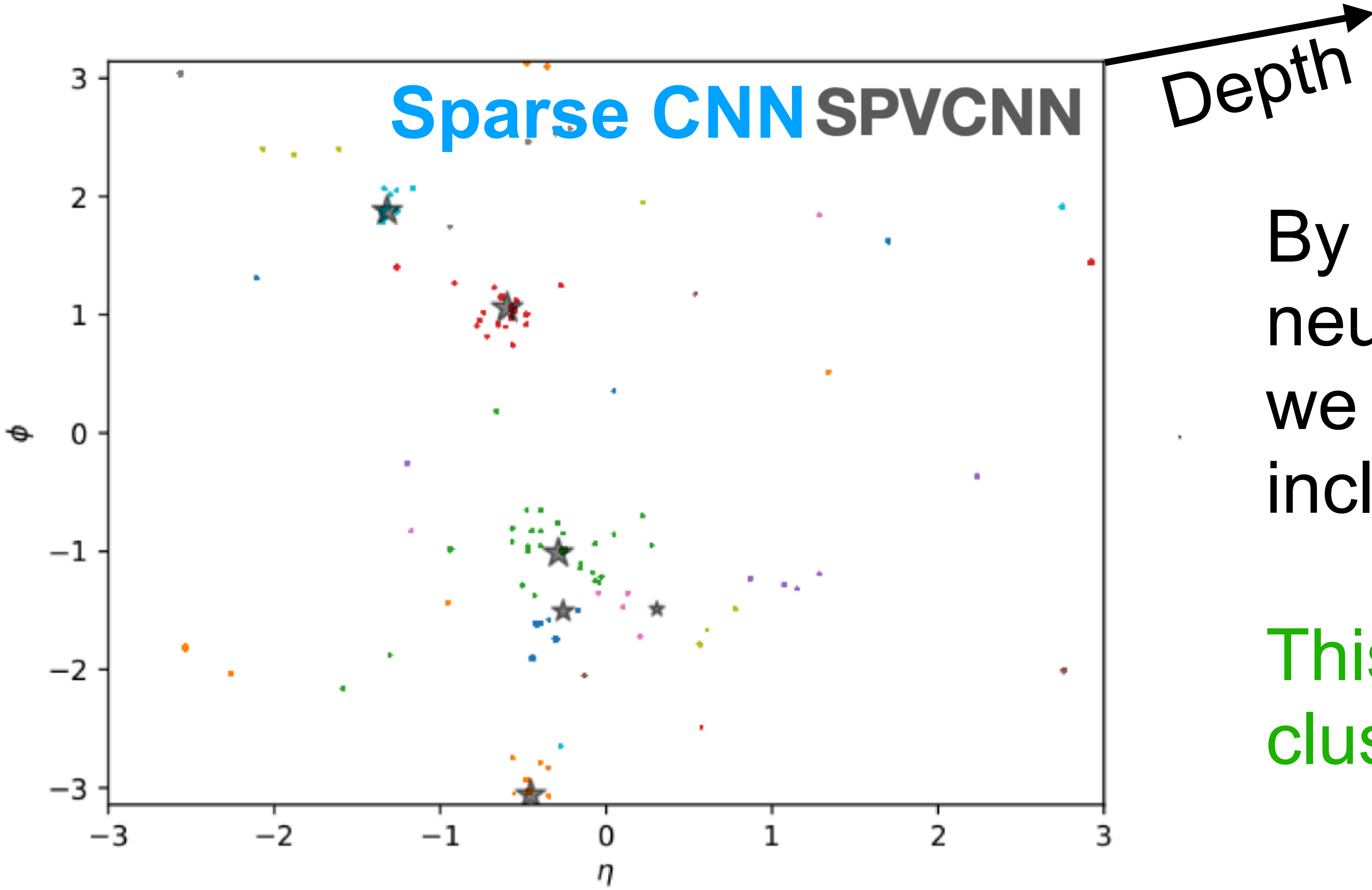
- A single algorithm is doing all of the clustering



- Clustering algorithm produces very similar results to truth
  - Single algorithm that takes in whole detector at once

So what do we gain?

- A single algorithm is doing all of the clustering

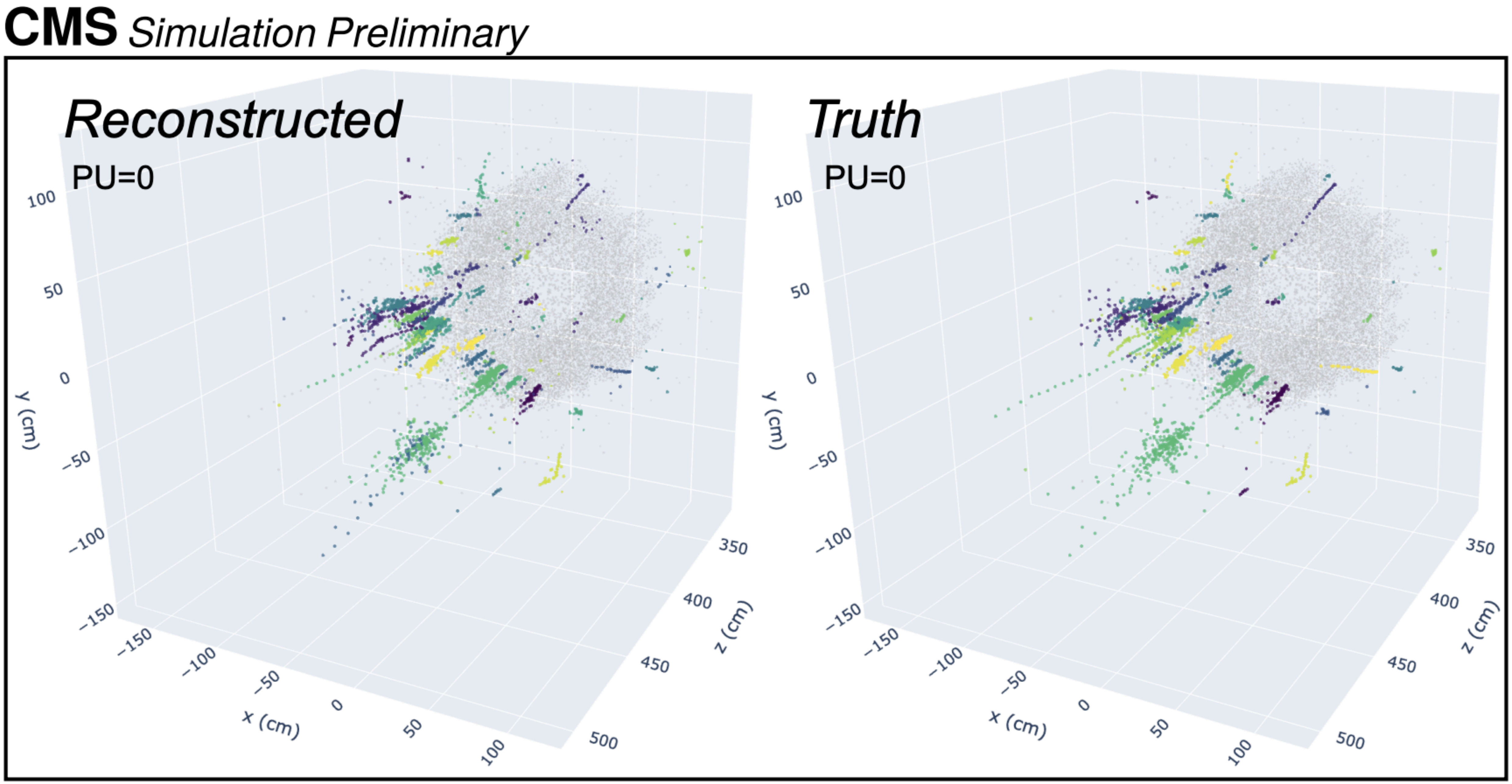


By embedding this in a neural network
we can extend algo to include more info

This is 1st algorithm to cluster with depth info

- Moreover this algorithm can now look at whole event to perform clustering
  - Awarness of the event can allow for dynamic thresholds/interpretations
- Finally, this algorithm is highly parallelize → Can Run it Fast!

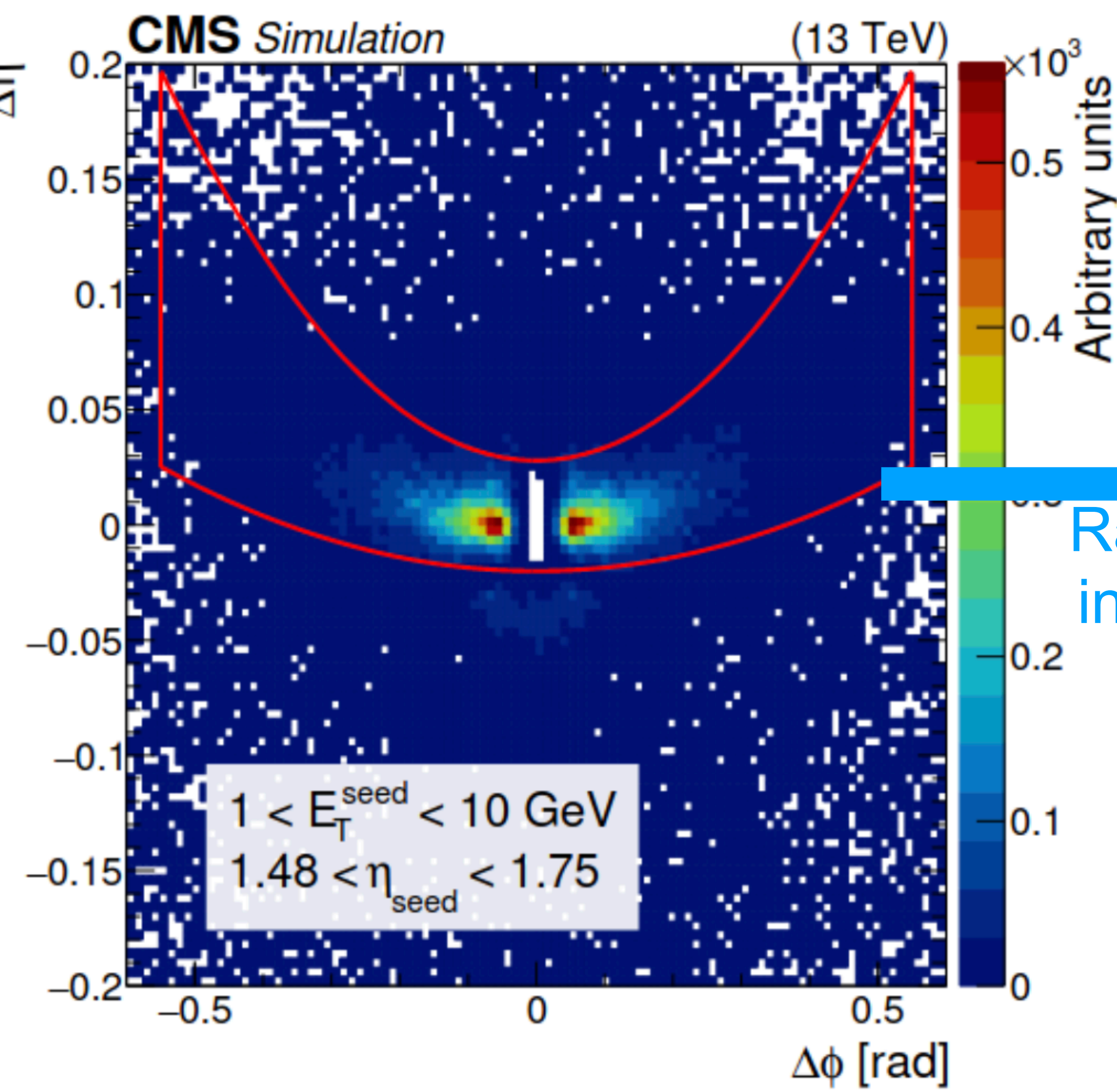
A more Extreme Example



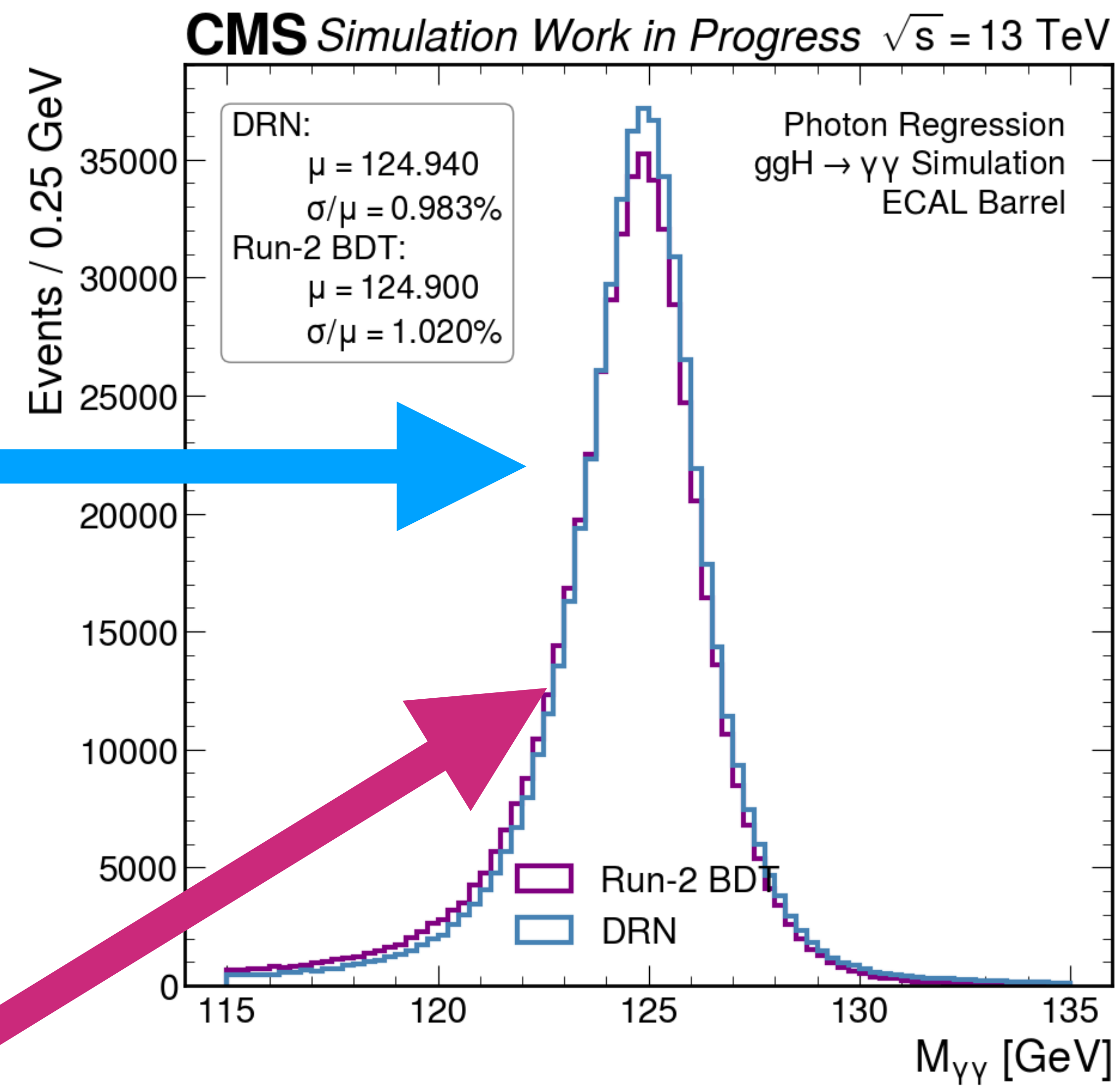
- Algorithm effective a reconstructing new complex topologies

Another Example

- Electron and Photon energy regression with an NN
  - Raw inputs to make an NN gives significant improvements

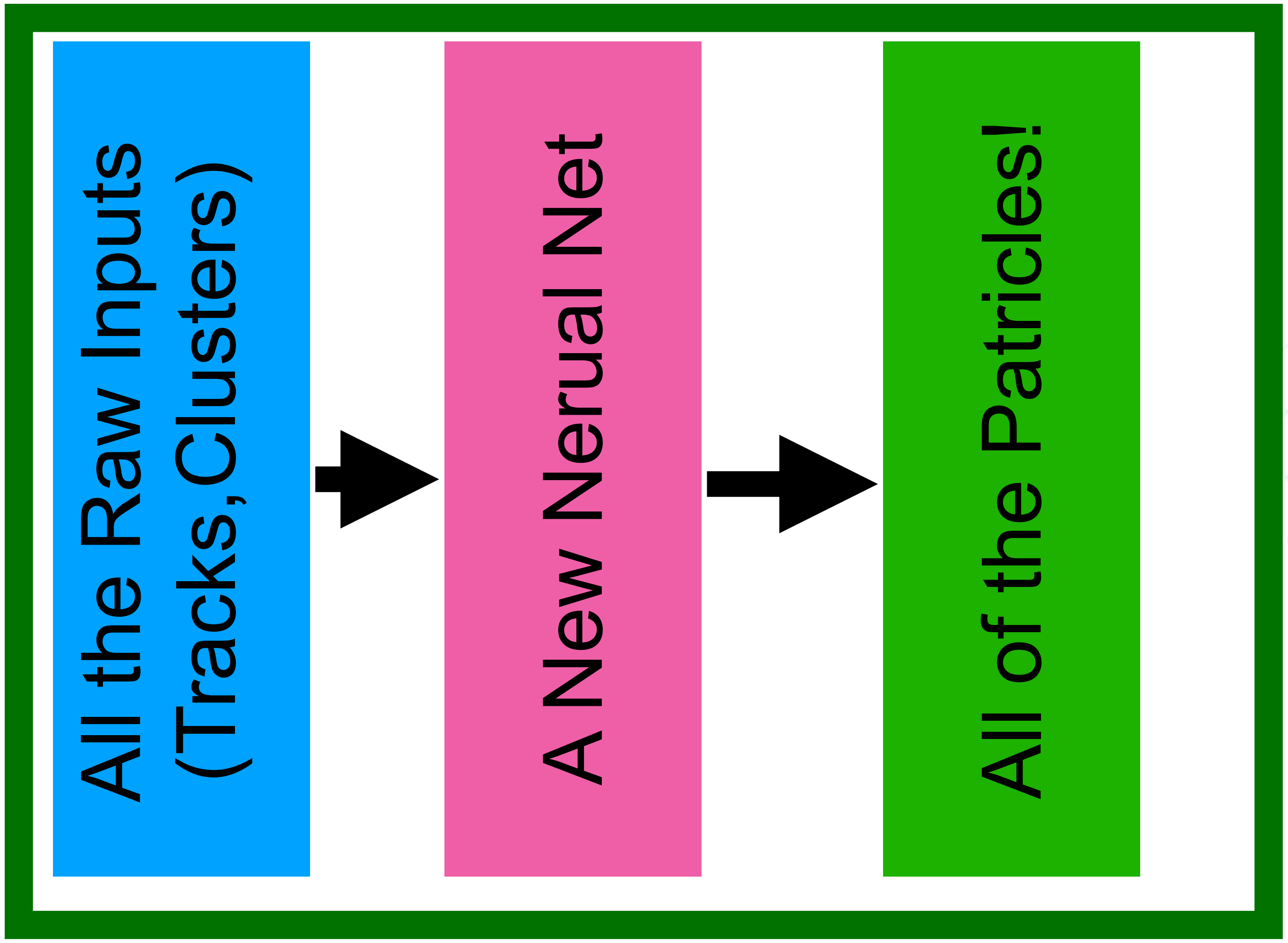


Raw Inputs into an NN

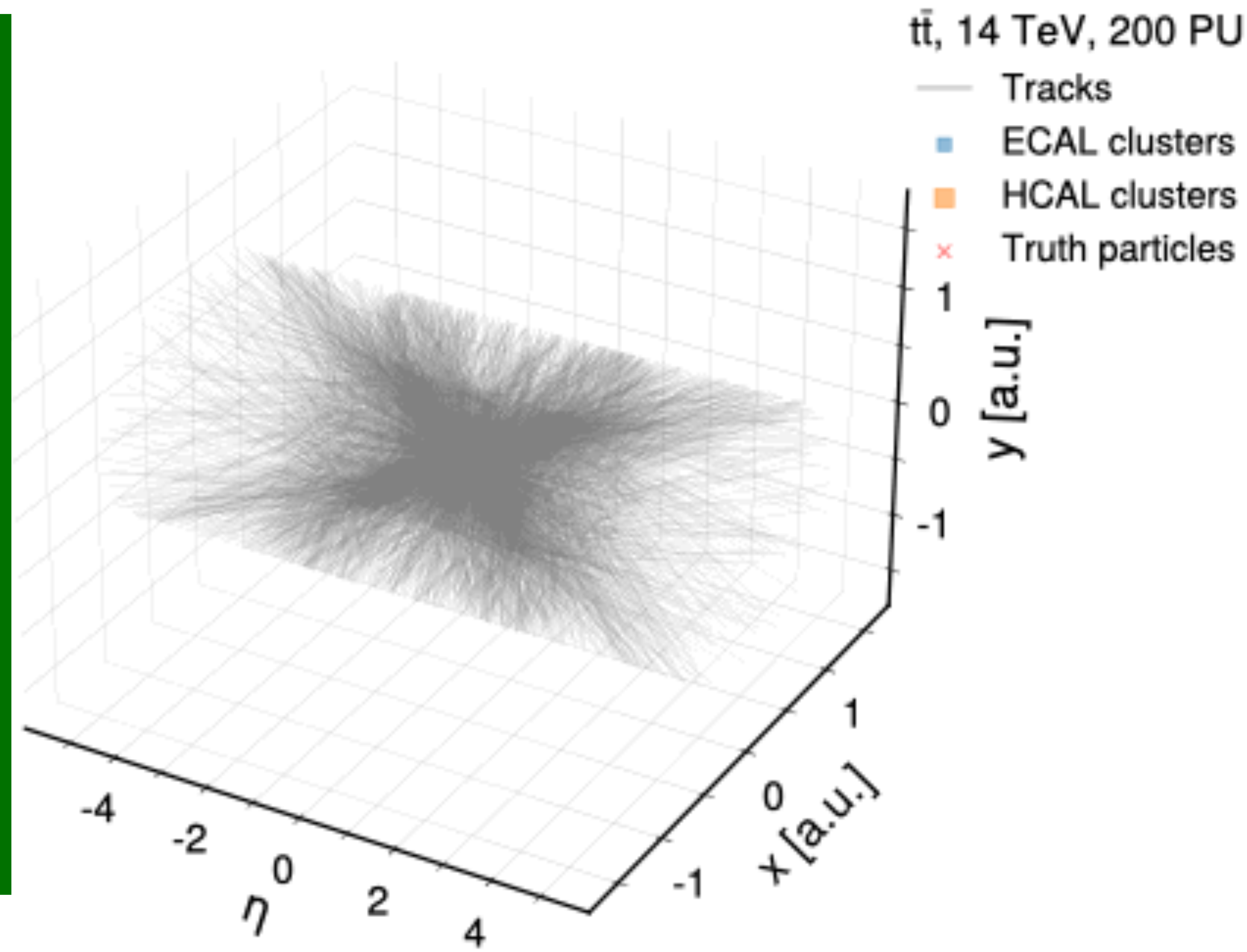


Previous Version used pre-reconstructed variables based on raw inputs eg. $\langle \Delta\phi^2 \rangle_{crystals}, \langle \Delta\eta^2 \rangle_{crystals}$

Success of Deep Learning



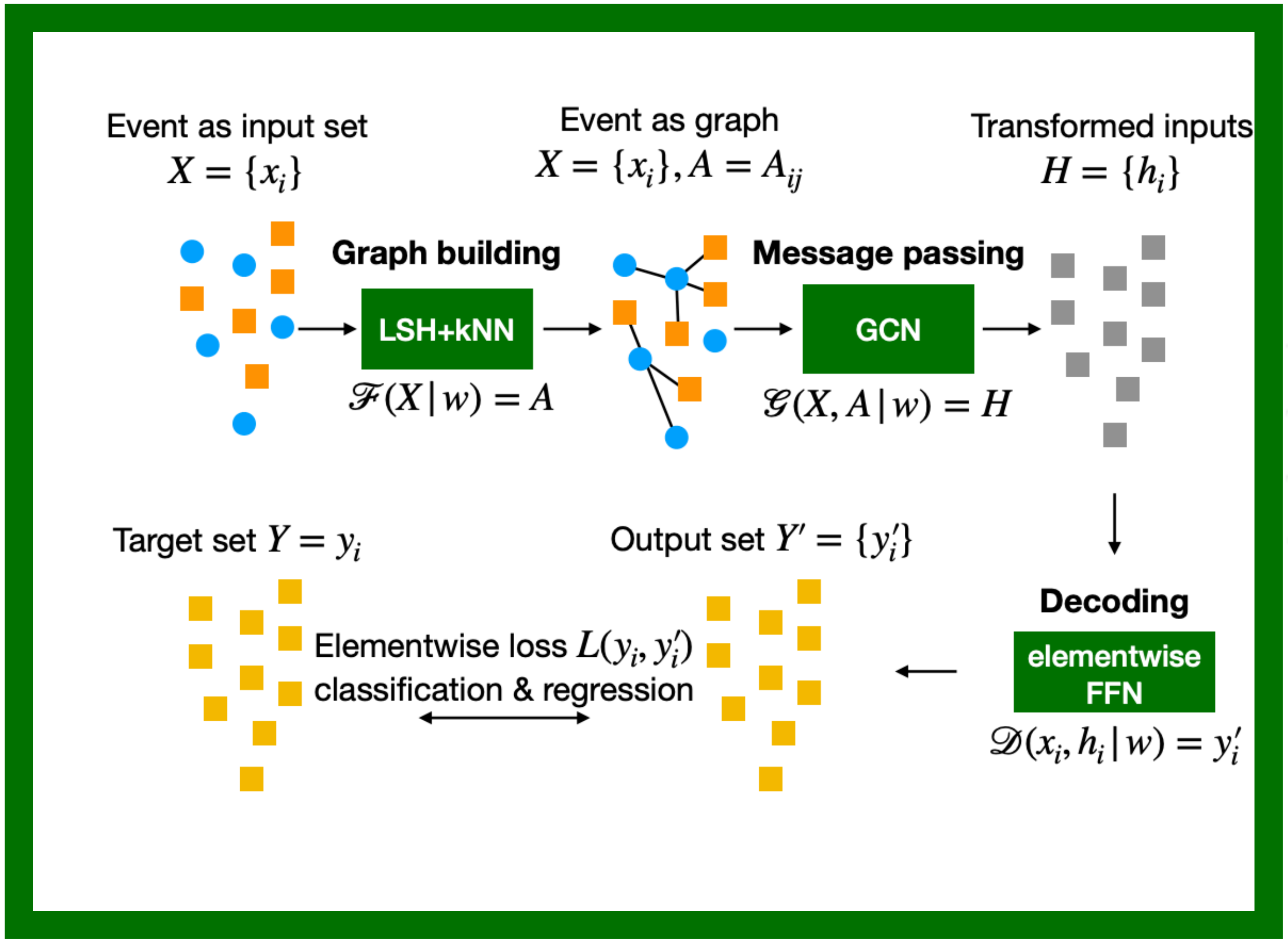
All particles in on fell swoop

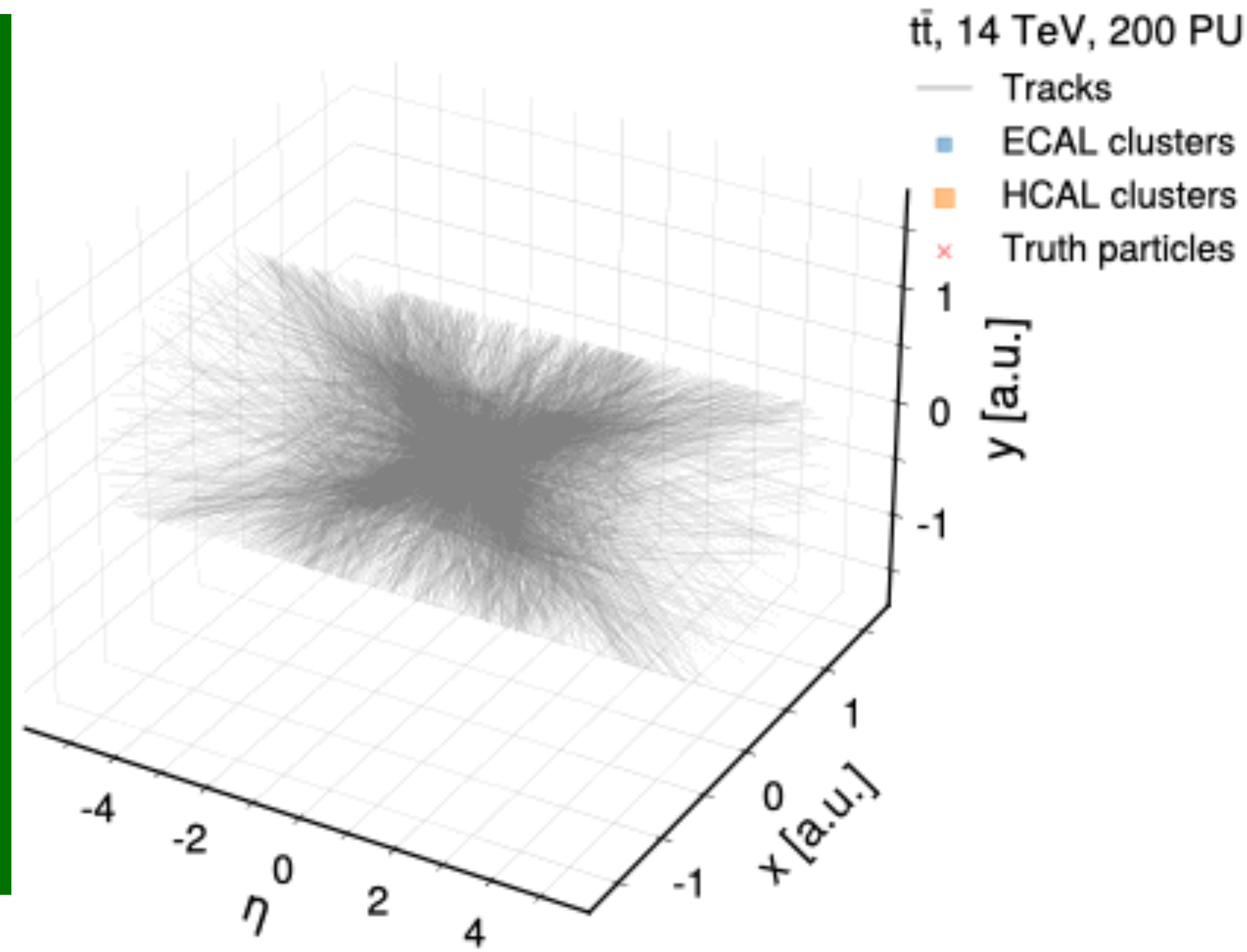


- First ideas of full particle based reconstruction are emerging
- Tools are emerging to do particle reconstructeion in one go

arxiv:2101.08578

Success of Deep Learning

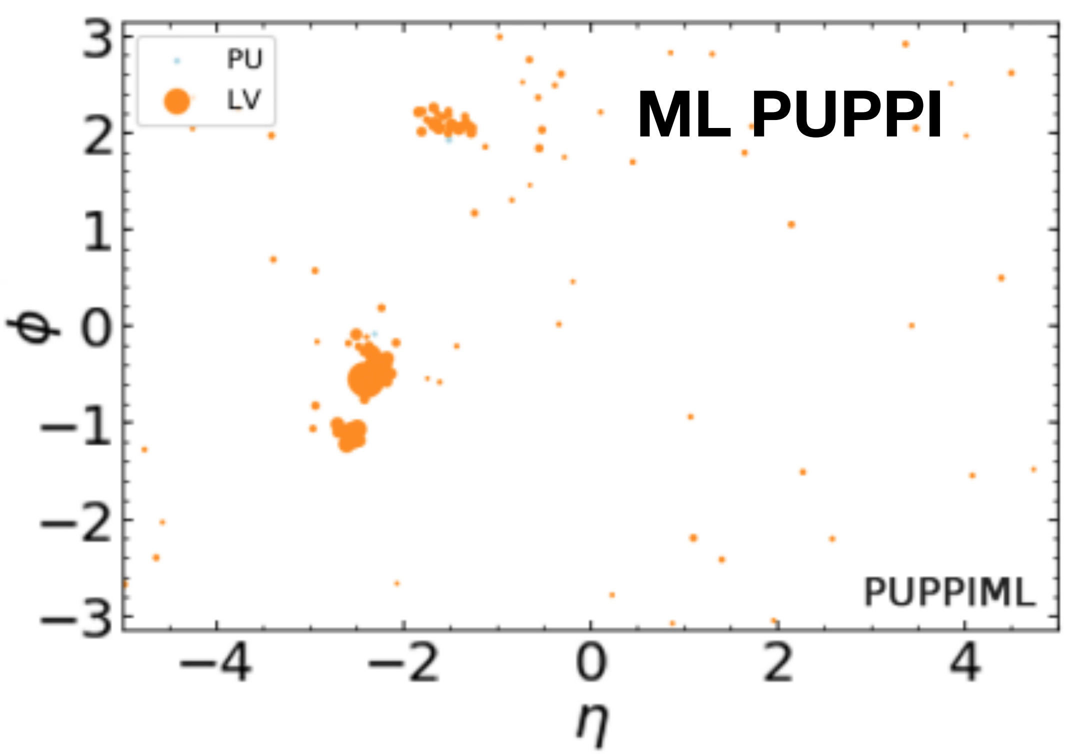




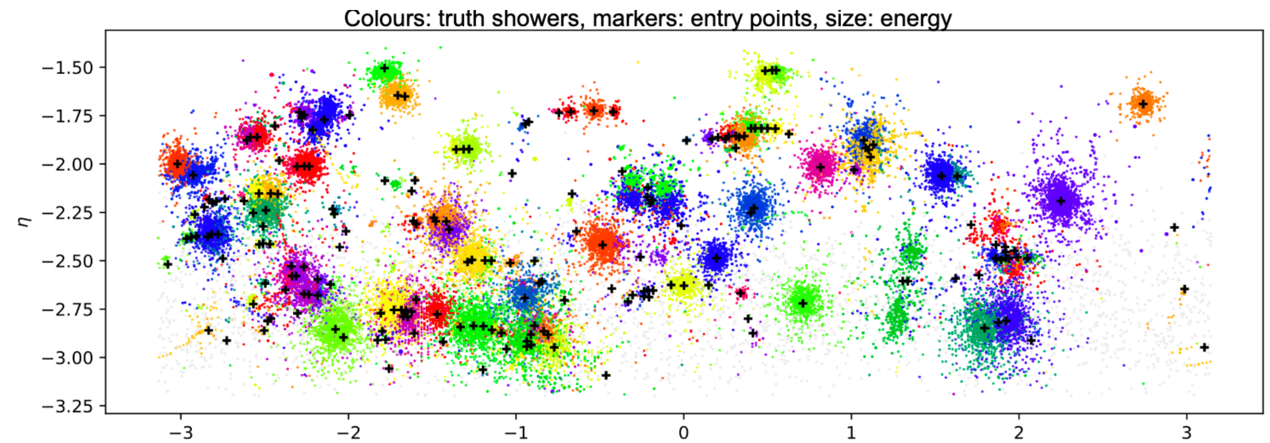
- First ideas of full particle based reconstruction are emerging
- LHC is a great place for DL because we have fantastic simulation

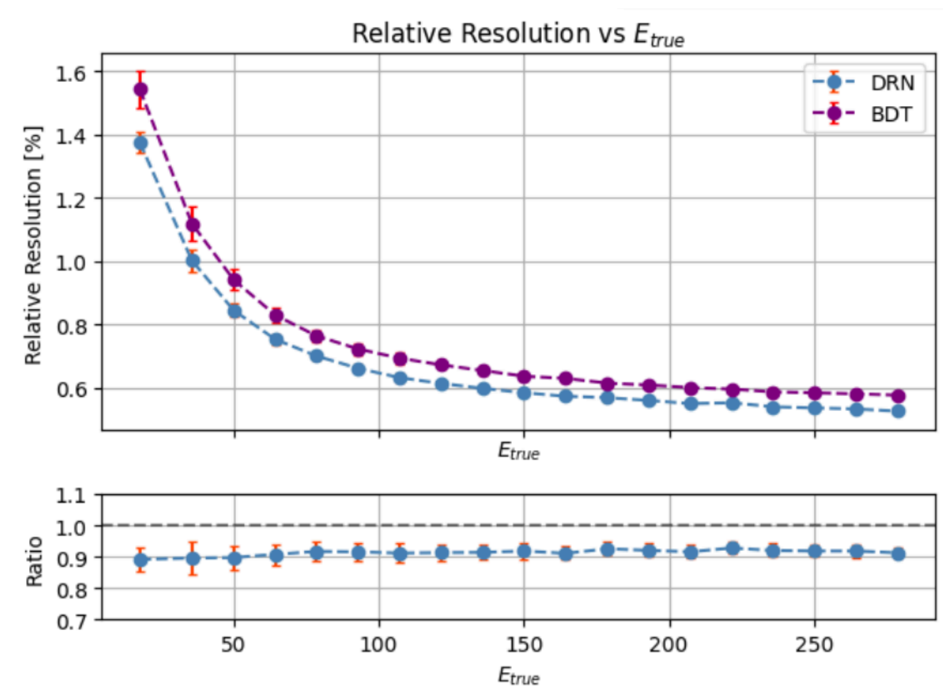
arxiv:2101.08578

Success of Deep Learning



Clustering: Graph NNs for HGCAL



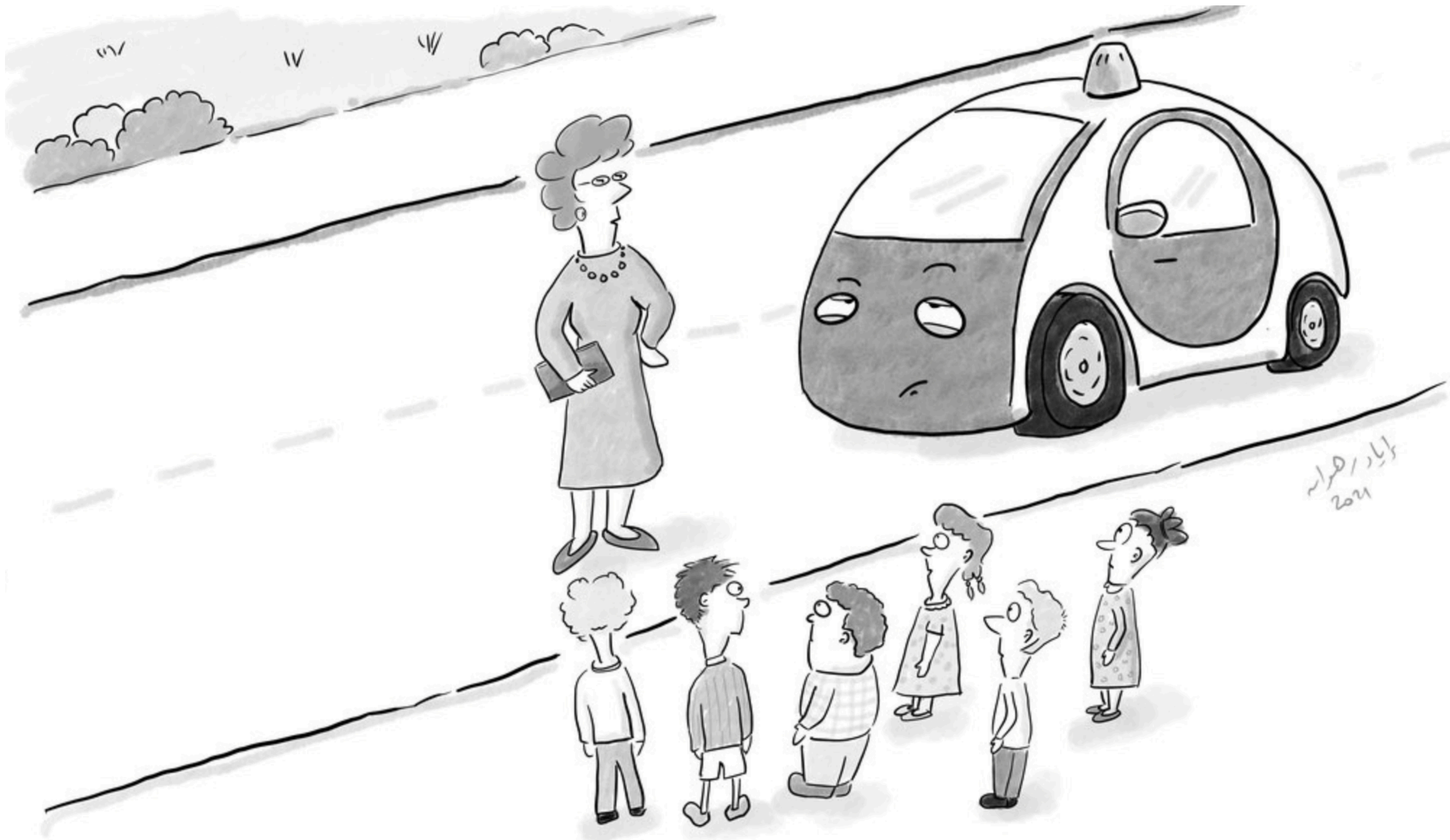


Dynamic reduction network for EGamma regression

S. Rothman

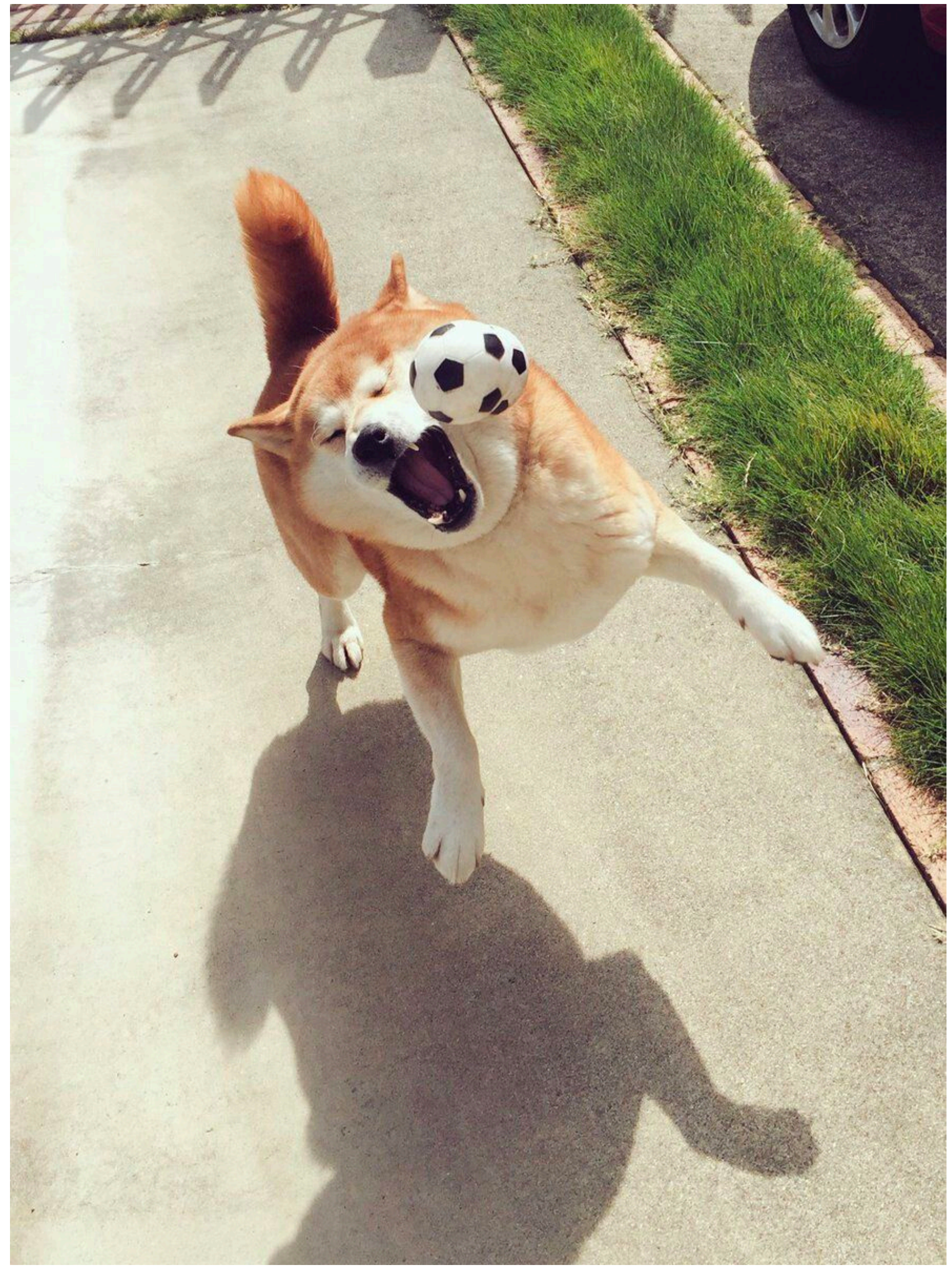
- Networks are emerging to do calorimeter clustering
- Additionally networks are emerging to identify all objects

Taking a Leap of Faith



“Remember kids, you should never look before you cross, because driverless cars will always stop for you!”

Can we really trust AI to work from scratch well? always?

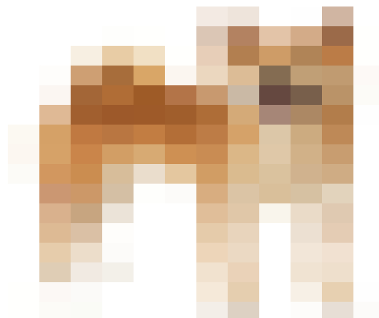


and Thinking Fast!
(NN Inference)

Spanning Frequencies

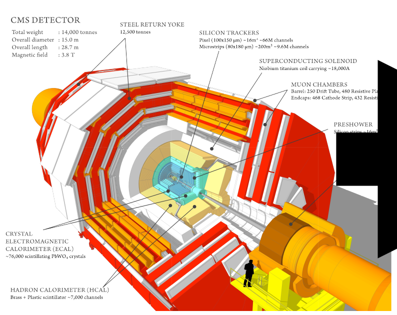
40 MHz **1 kHz**

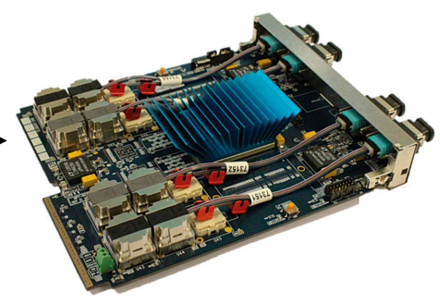
25ns **1ms**



Radiation Hard ASICs

FPGA Boards





Select 1 event in 400

The rest is thrown away **Forever**!

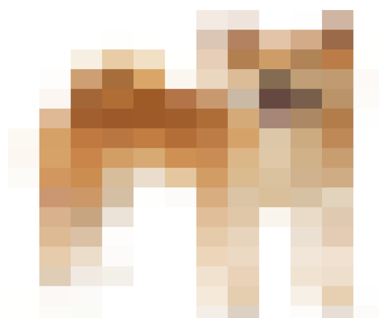
320 tb/s

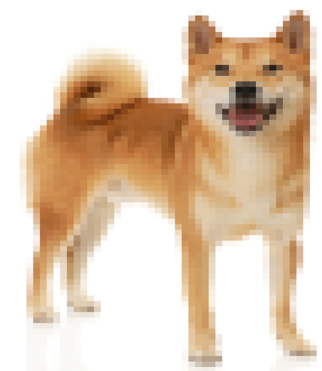
Fast
40 MHz Collisions
10 μs window
L1Trigger

Spanning Frequencies

40 MHz **1 kHz**

25ns **1ms**

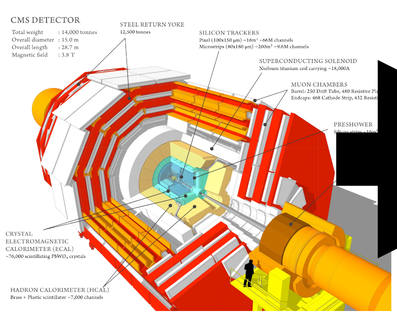


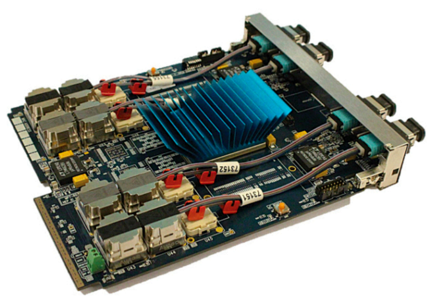


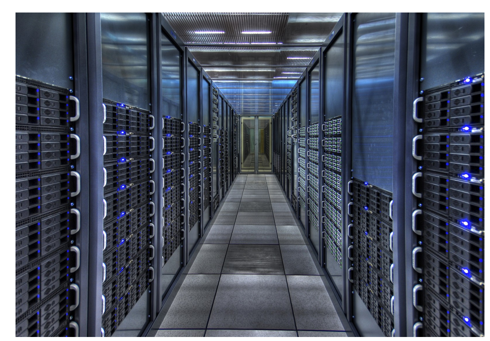
Radiation Hard ASICs

FPGA Boards

Local CPU Cluster







320 tb/s

1 tb/s

Fast
40 MHz Collisions
10 μs window
L1Trigger

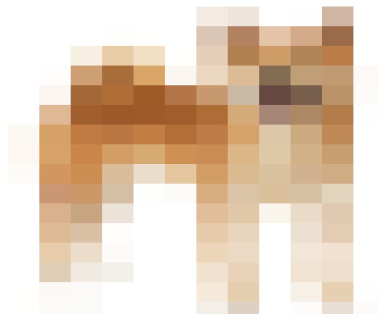
Intermediate
100 kHz Collisions
<500 ms window
High Level Trigger

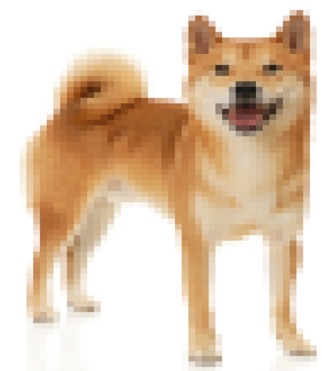
Select 1 in 100

Spanning Frequencies

40 MHz **1 kHz**

25ns **1ms**



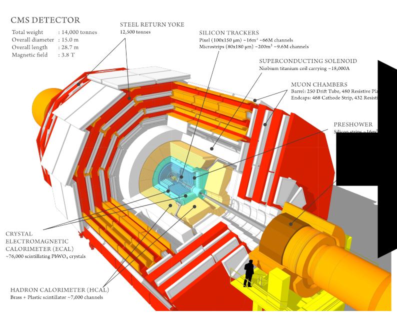


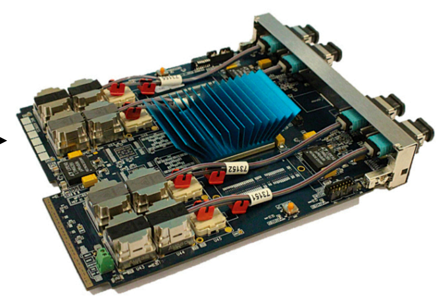
Radiation Hard ASICs

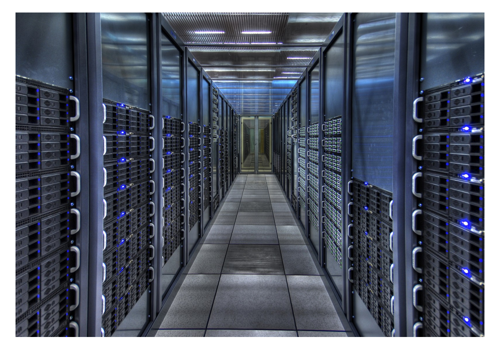
FPGA Boards

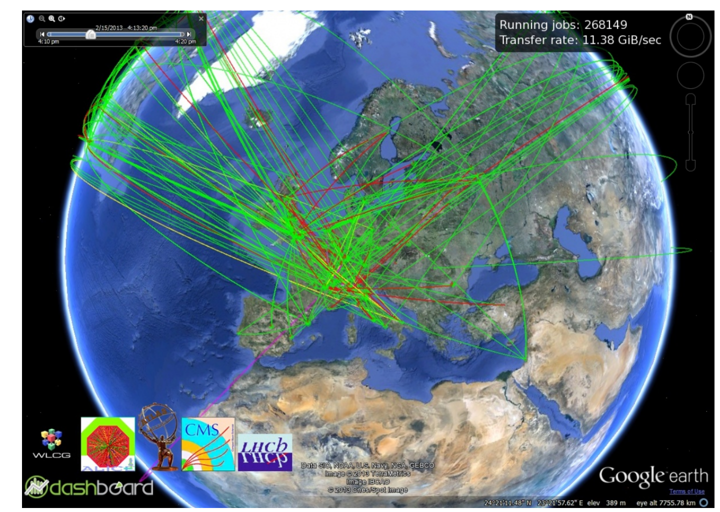
Local CPU Cluster

CPU Grid









320 tb/s

1 tb/s

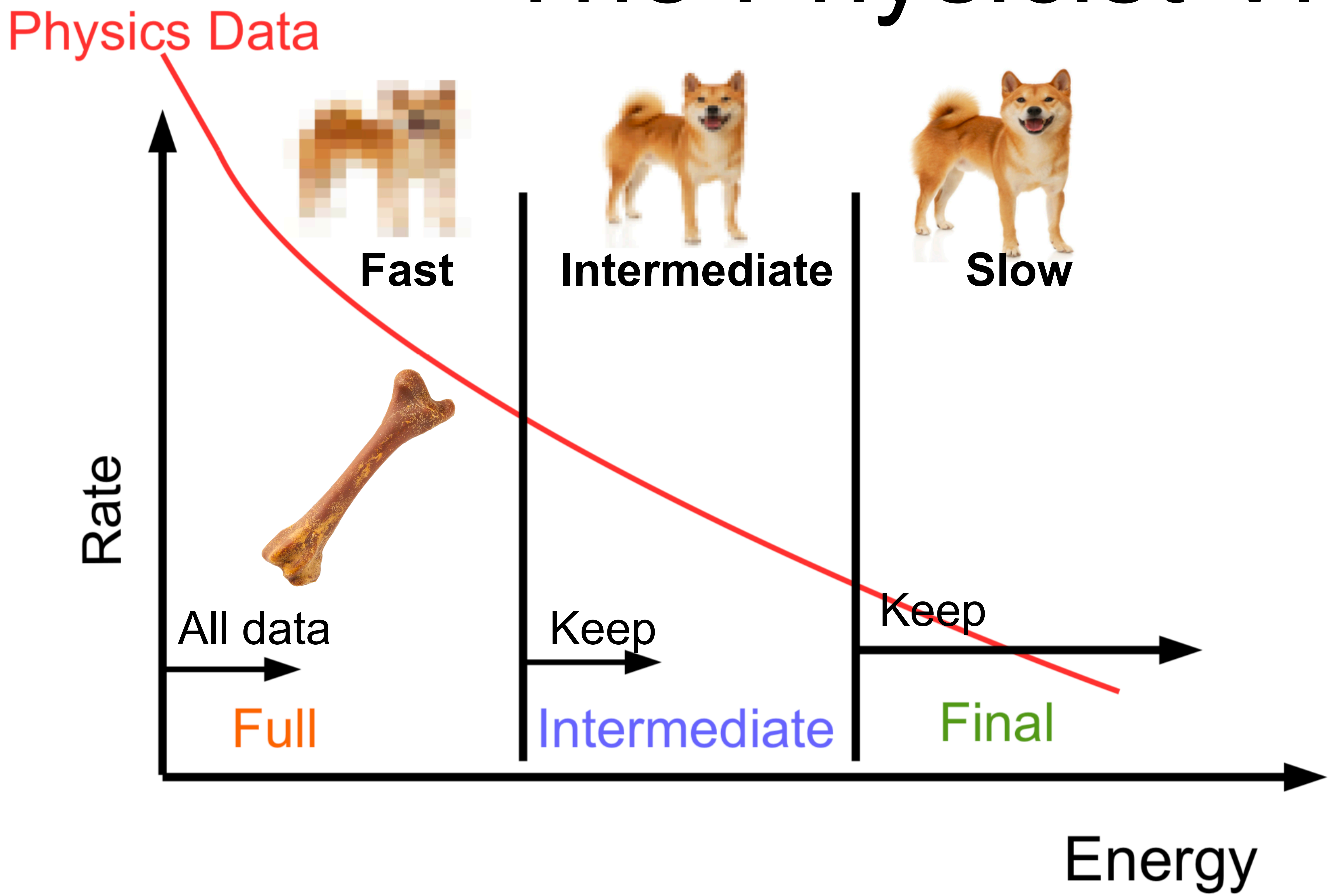
10 Gb/s

Fast
40 MHz Collisions
10 µs window
L1Trigger

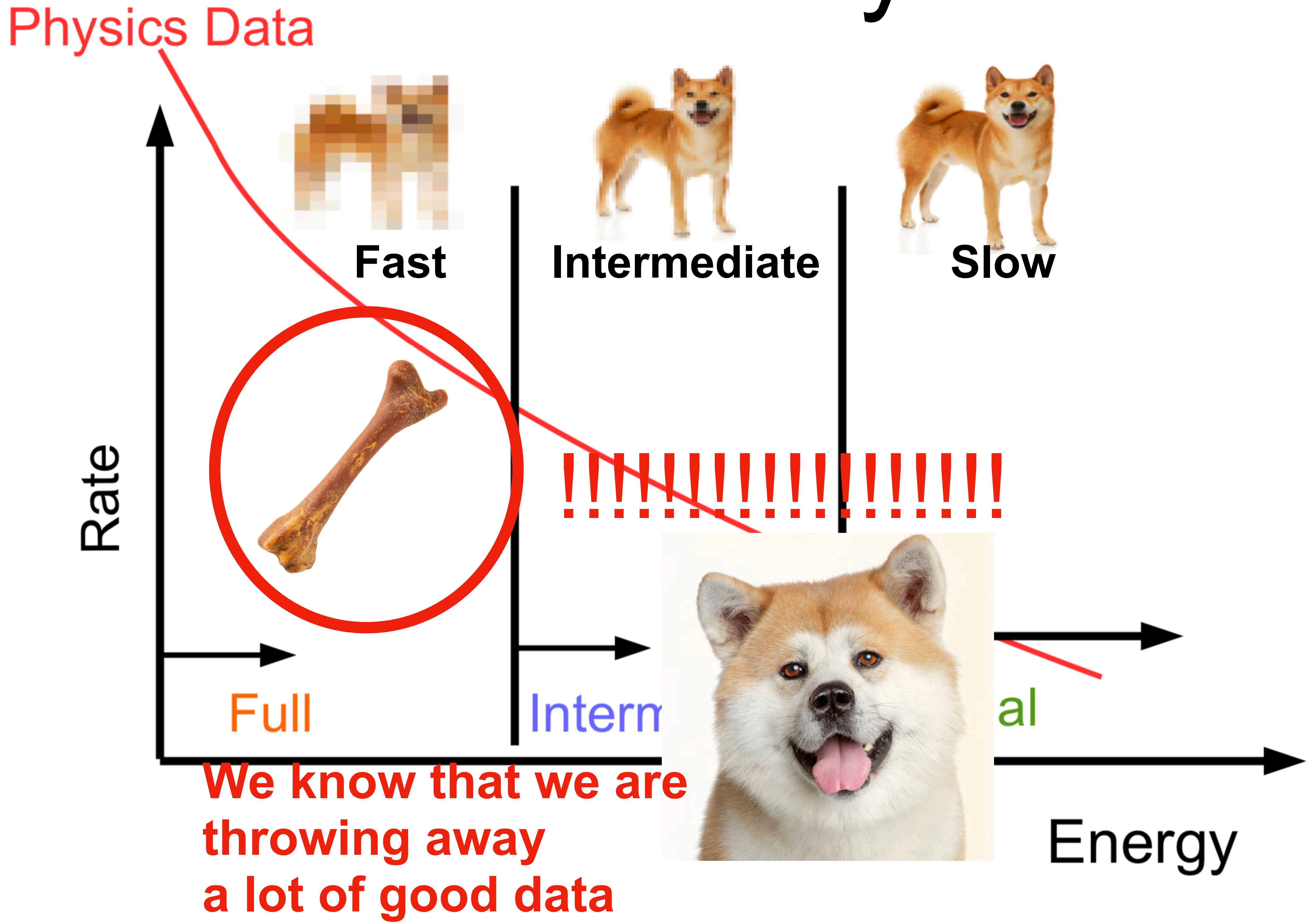
Intermediate
100 kHz Collisions
<500 ms window
High Level Trigger

Slow
1 kHz Collisions
10 s window
Offline Cluster

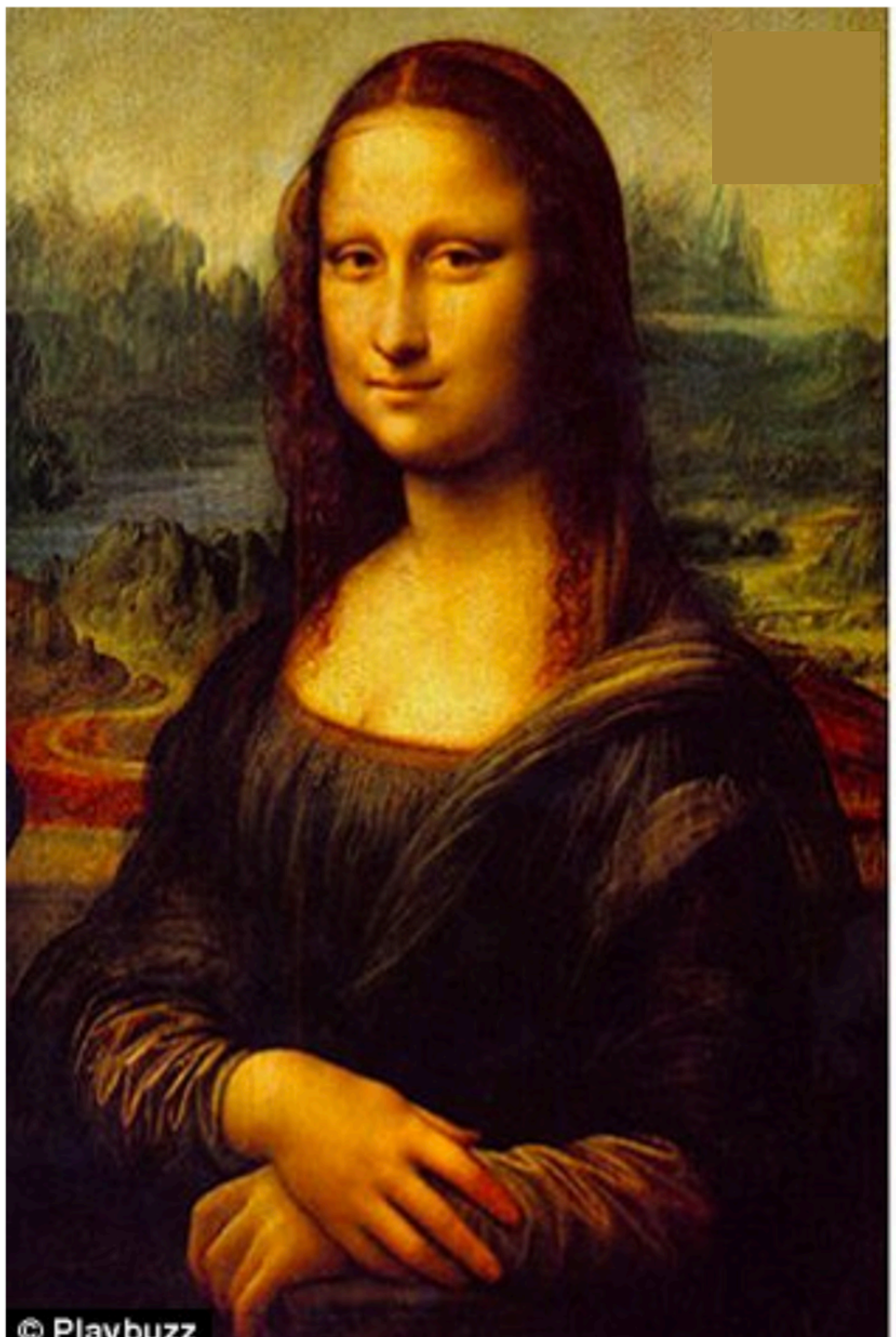
The Physicist View



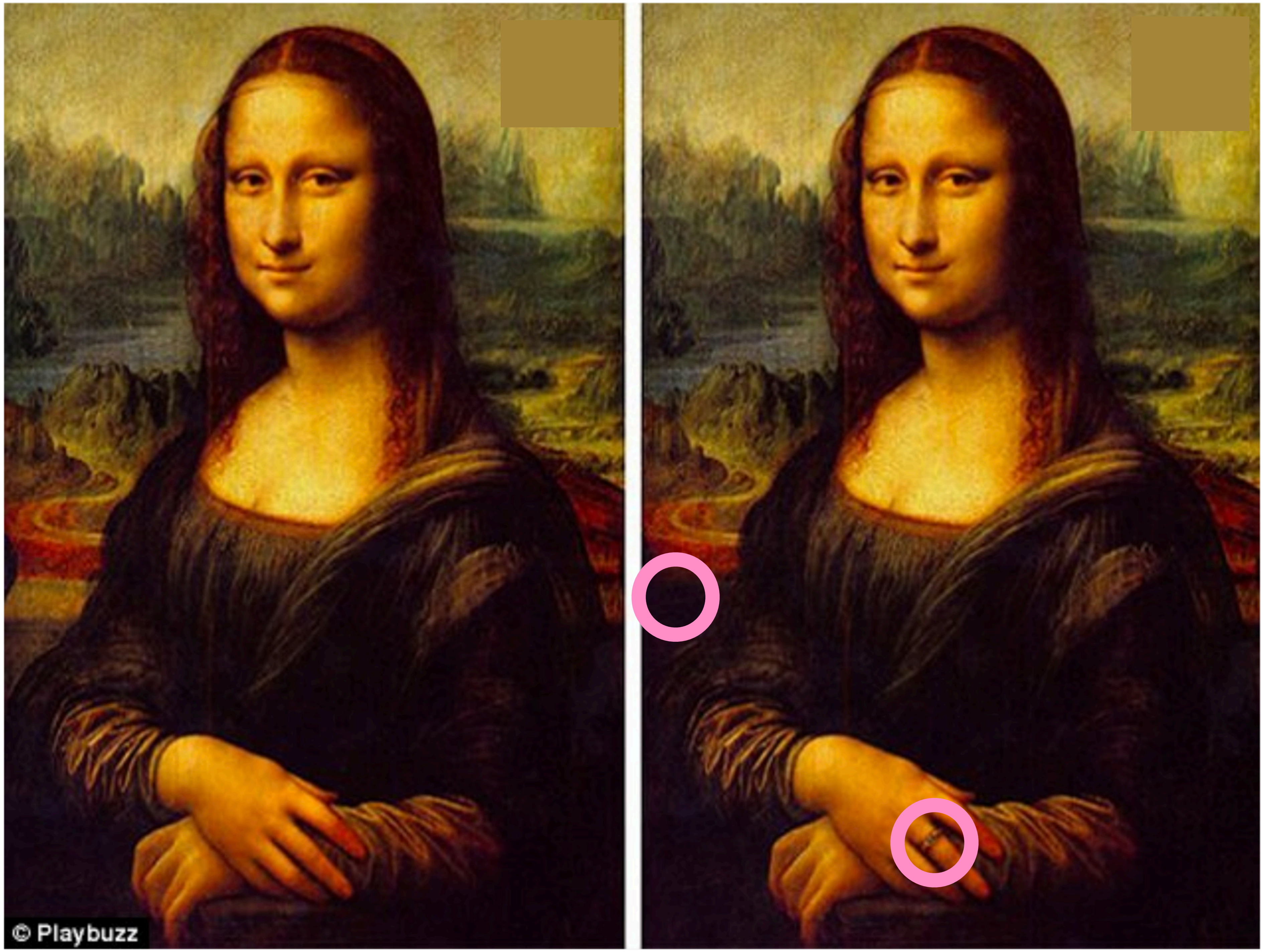
The Physicist View



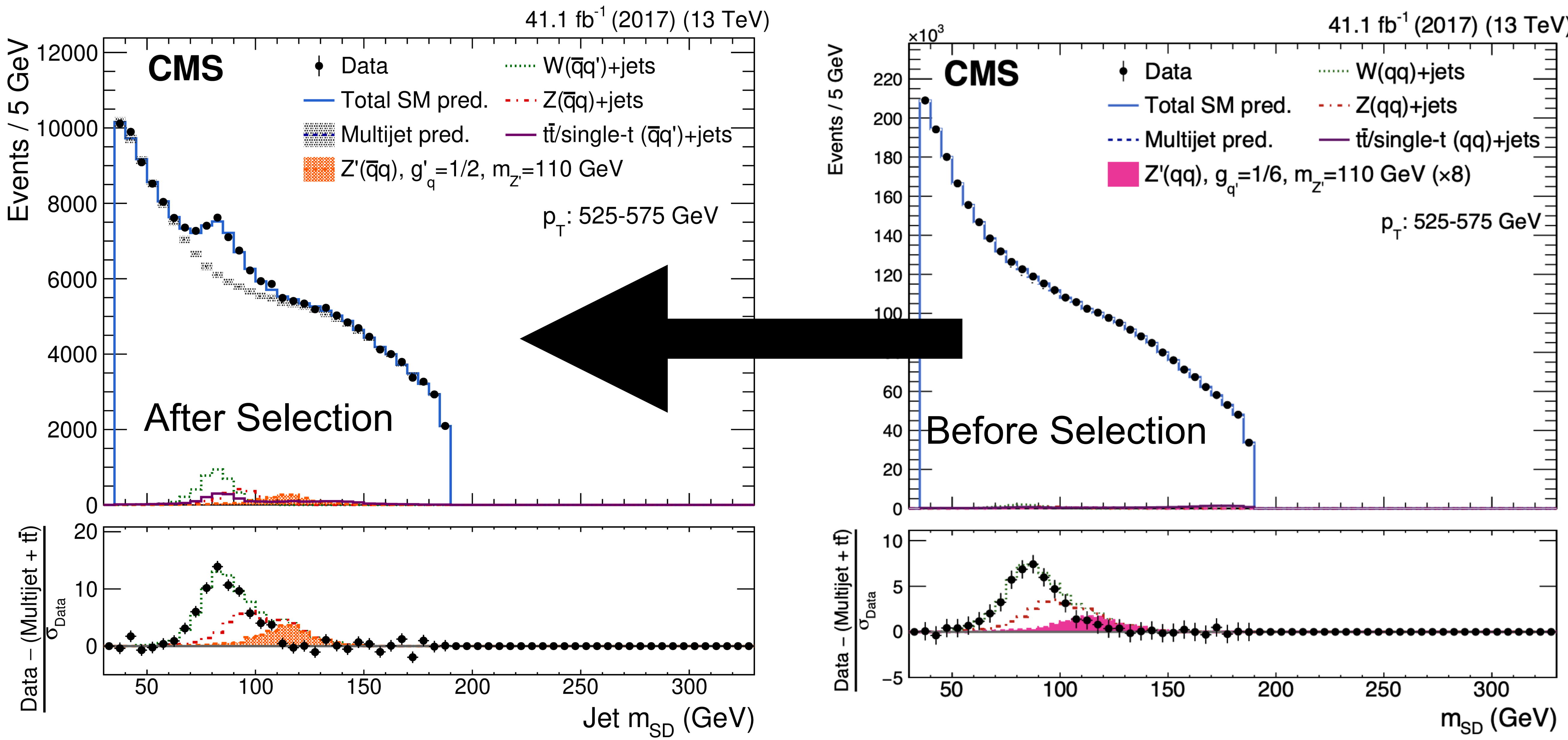
What is different w/Left and Right?



The Need for Subtlety



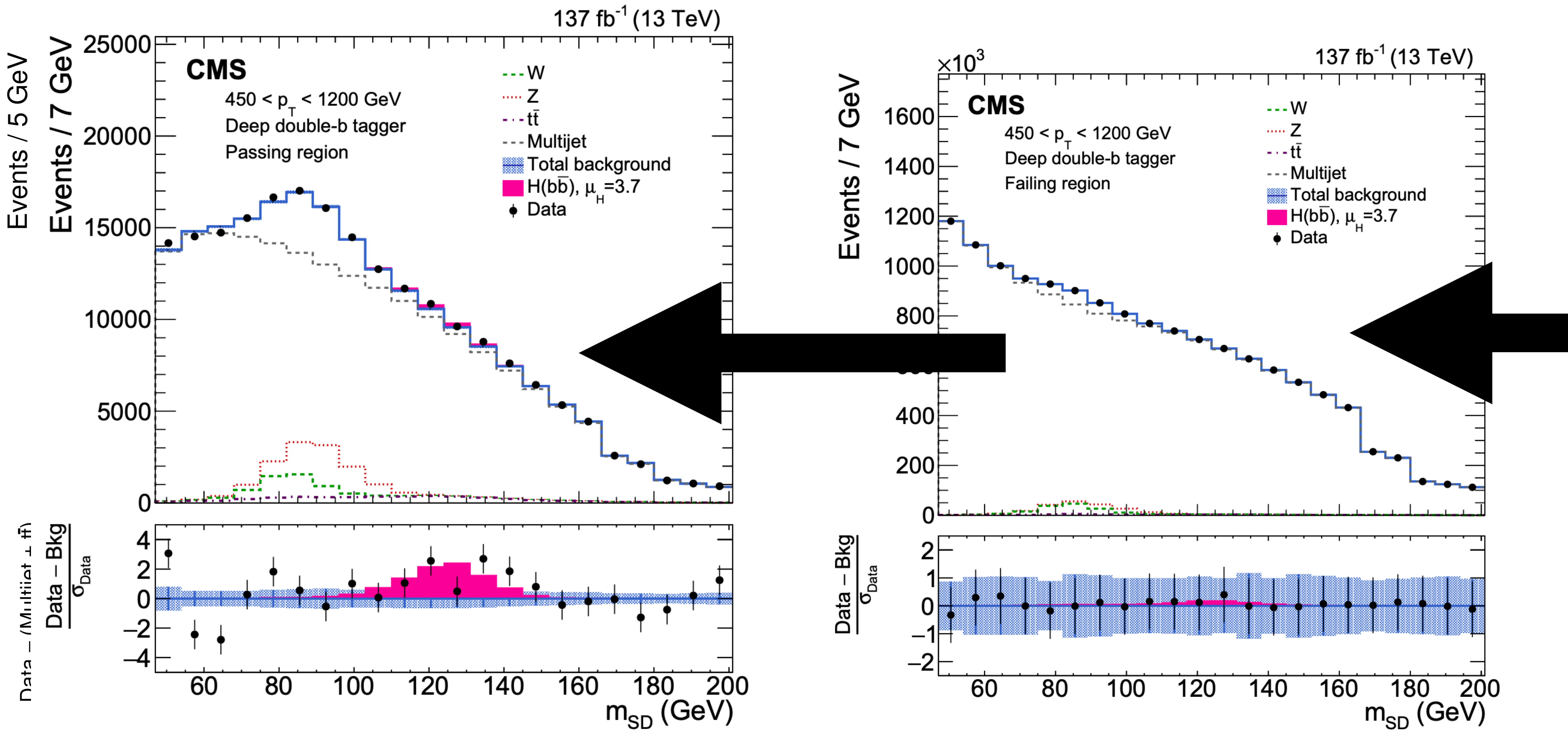
Looking for small signals



There is still a wealth of unexplored physics at the LHC

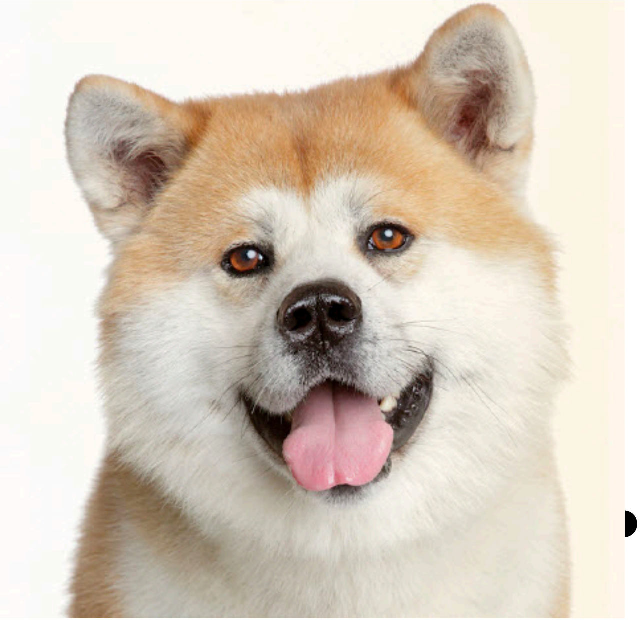
Its just a bit harder to find

Looking for small signals



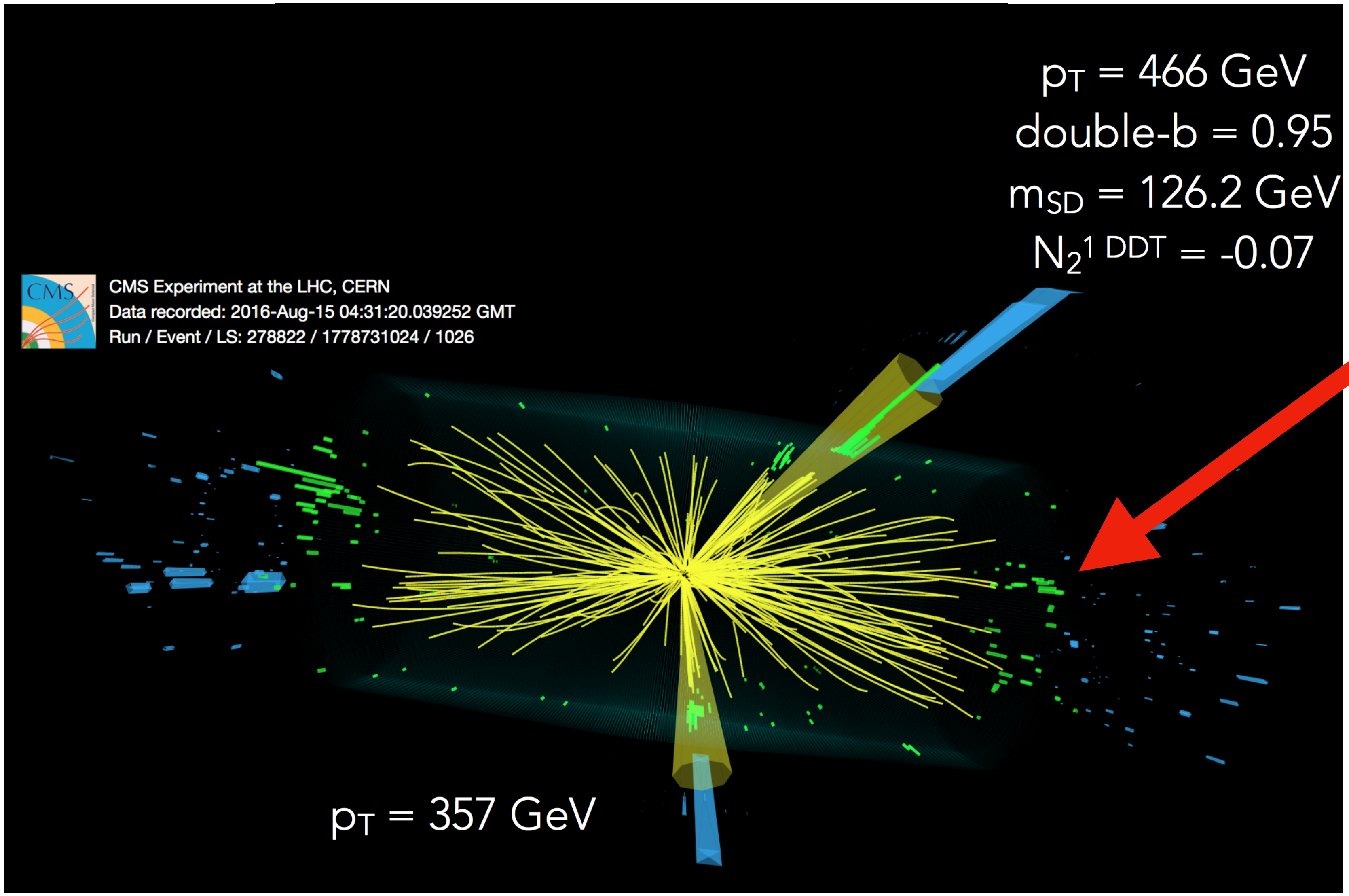
There is still a wealth of unexplored physics at the LHC

Its just a bit harder to find



Hidden gems?

- There is a plethora of physics that we throw out



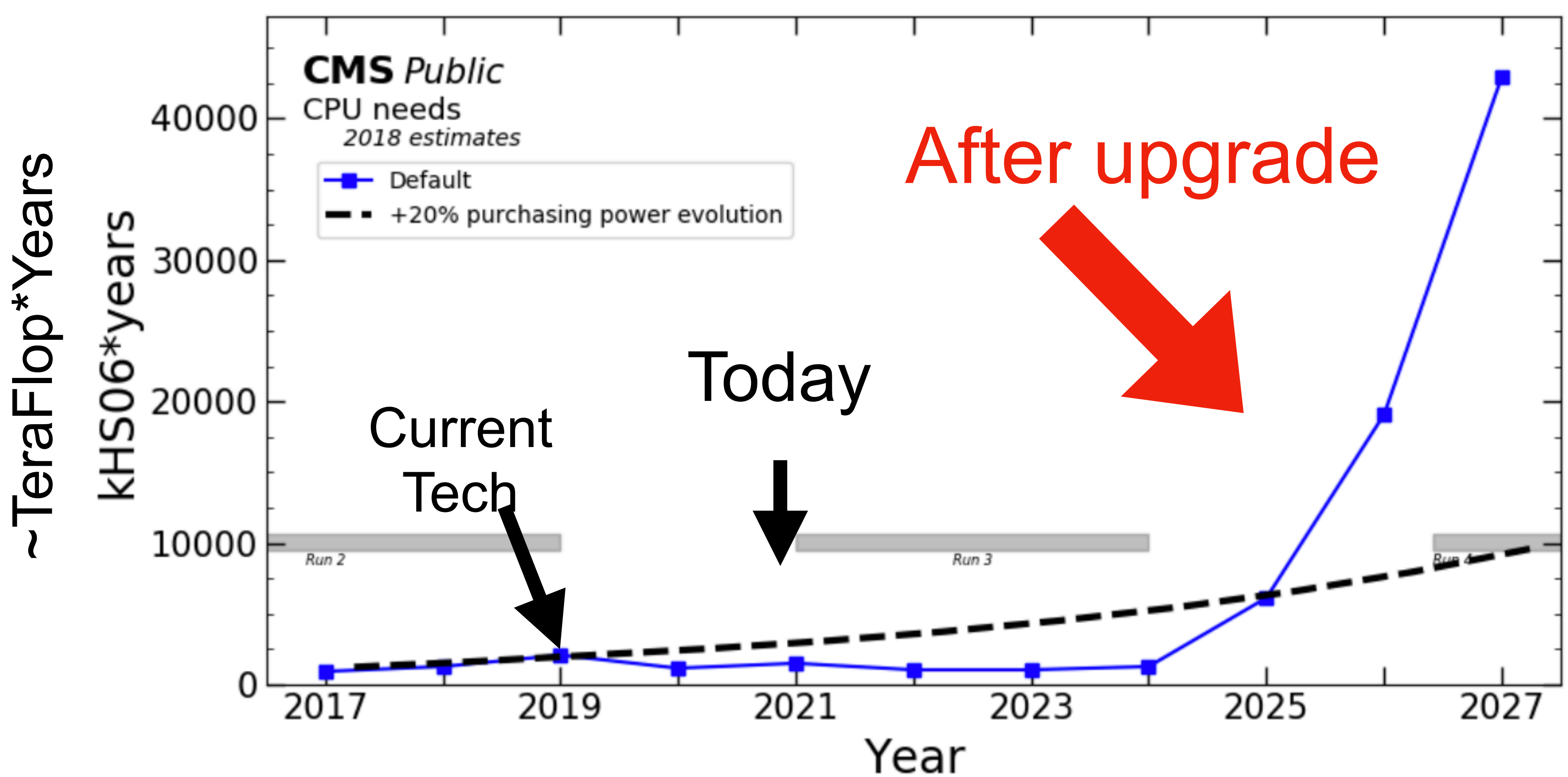
Higgs boson right on the cusp of being thrown out

The dream

- At the moment:
  - We only get a full data of one in 40,000 collisions
  - There is interesting physics that we have to throw away
- We would like to analyze every collision at the LHC
  - To deal with this we need to increase our throughput
  - Ultimately this means going to 100s of Tb/s

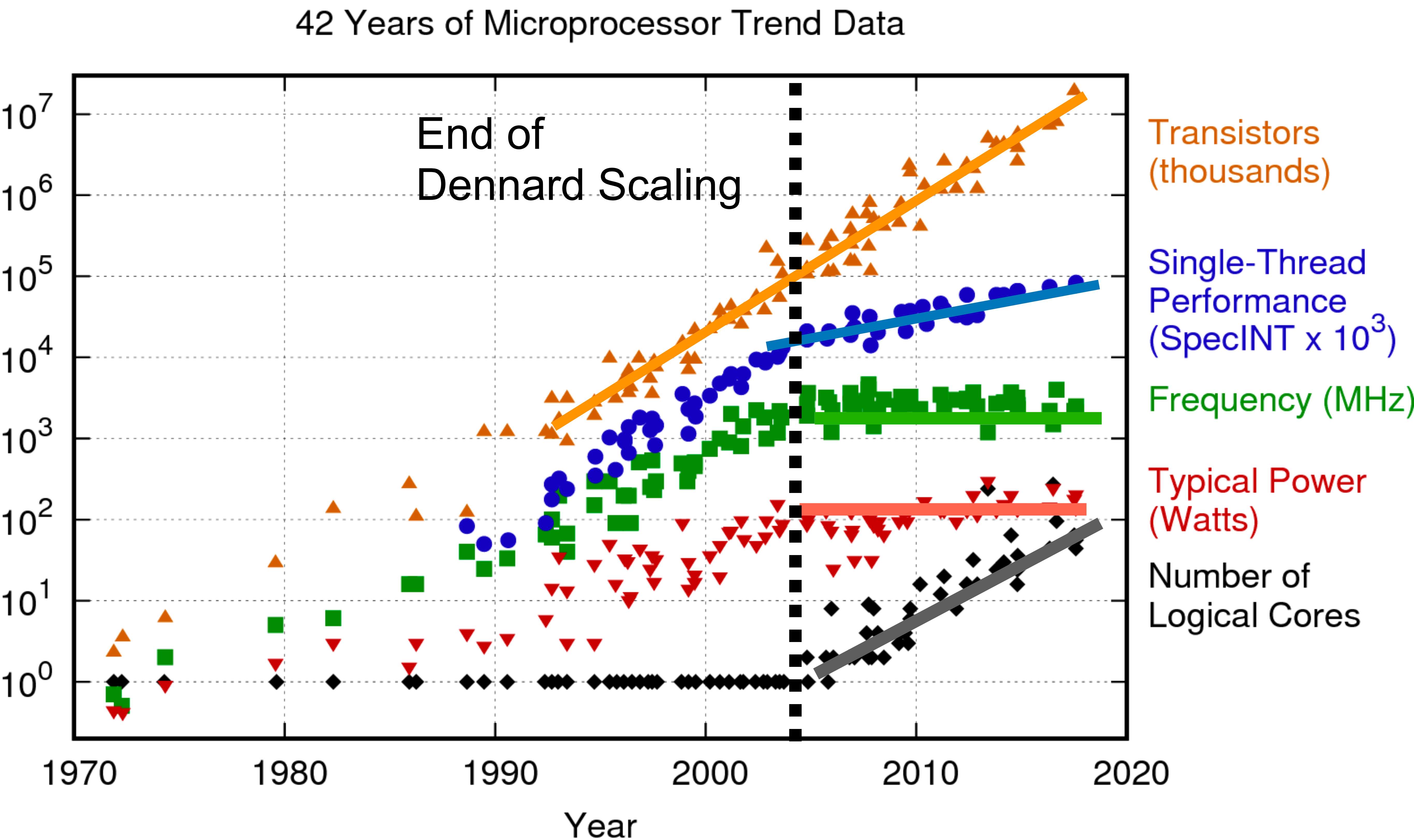
The Challenge

- To deal with the upgraded LHC intensity
- To preserve current physics we are upgrading the system
  - Our event size will have to be 10x larger
  - We will have to take data at 5 times the current rate



Results

The Crises



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Processor Technology

Will we be able to handle the future upgrades?

Modern Processing

- Multi cores CPU:
  - Your standard CPU with split all up (you know this)
- GPU :
  - Effectively many multi-cores with simplified instructions
    - Many cores in parallel (O(500)) with addition and mult.
  - Power hungry (better, but not too different from a CPU)
- FPGA:
  - Pre-programed the chip to do the operation you want
  - Every switch and multiplier assigned to a fixed patter
  - Energy efficient and hyper parallel (5000 parallel)
- ASIC:
  - FPGA but with inability to be programmed

Processing Tech

CPU	GPU	FPGA
		
1 player	A few at same time	The whole chip
A soloist	A group	An orchestra
Whatever	Main theme and some freedom to improv	Score has to be known perfectly beforehand

Processing Tech

Past	Present	Future
CPU	GPU	FPGA

Past

Speed: 1

Single complex operation per clock

Speed: 20-50

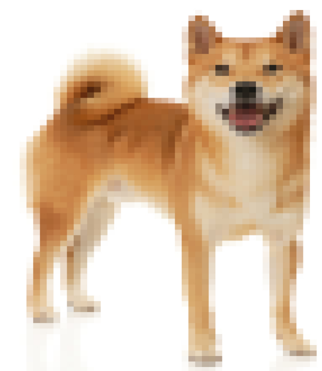
Multiple simpler operations in parallel commonly available

Speed: 200-1000

Efficient packing of operations highly parallelized (low power)

40 MHz **1 kHz**

25ns **1ms**

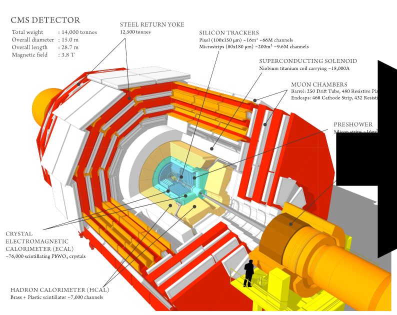


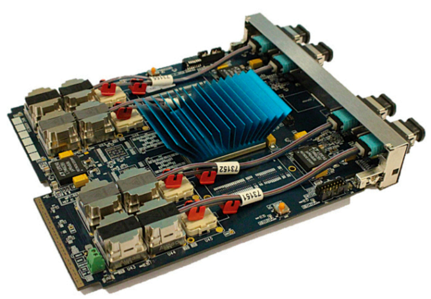
Radiation Hard ASICs

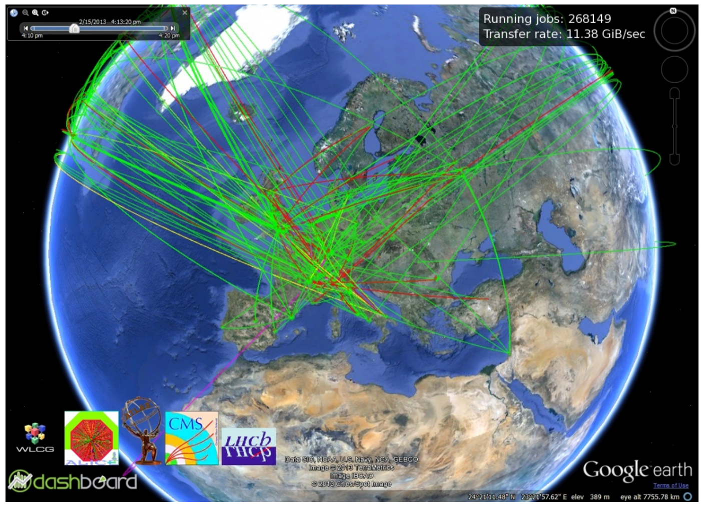
FPGA Boards

Local CPU Cluster

CPU Grid







320 tb/s

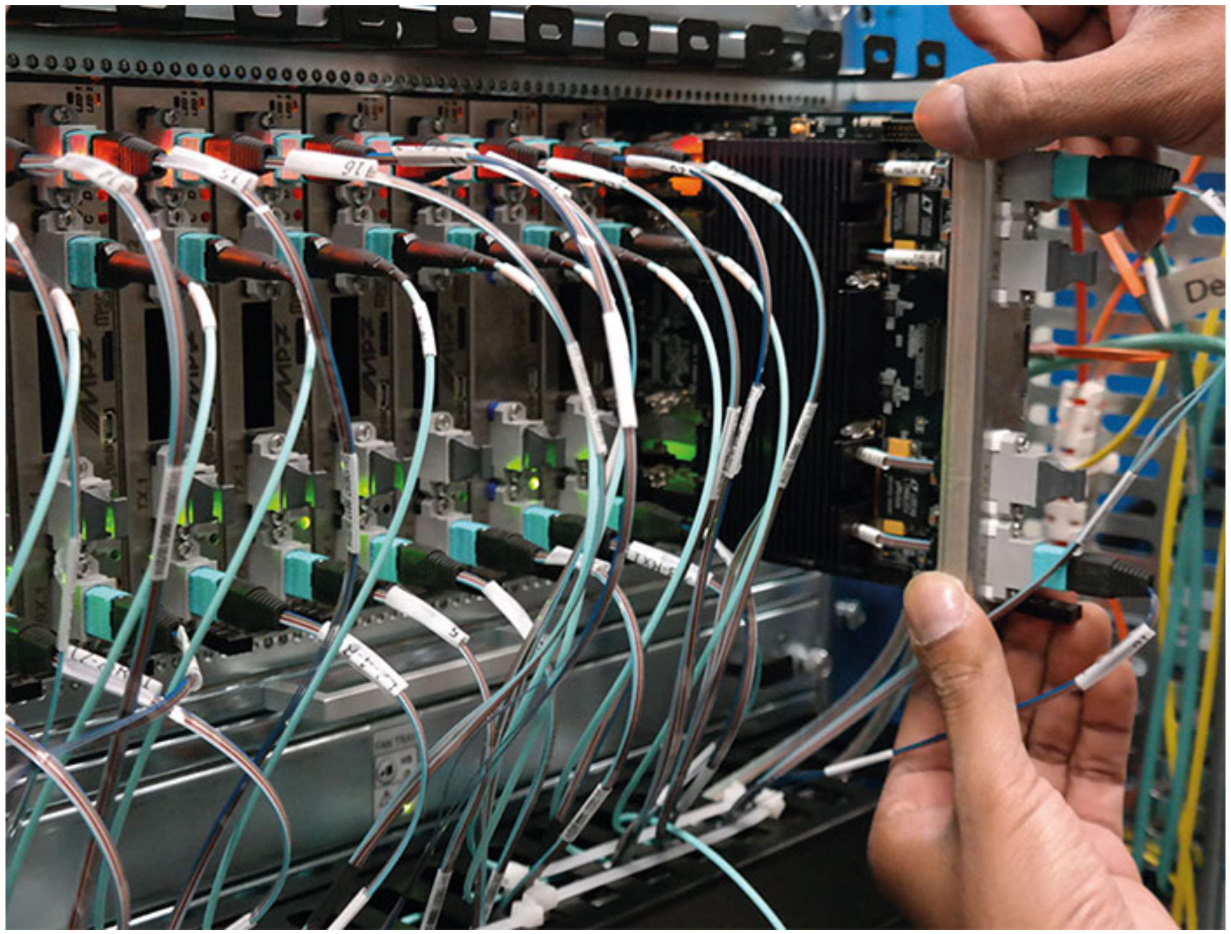
1 tb/s

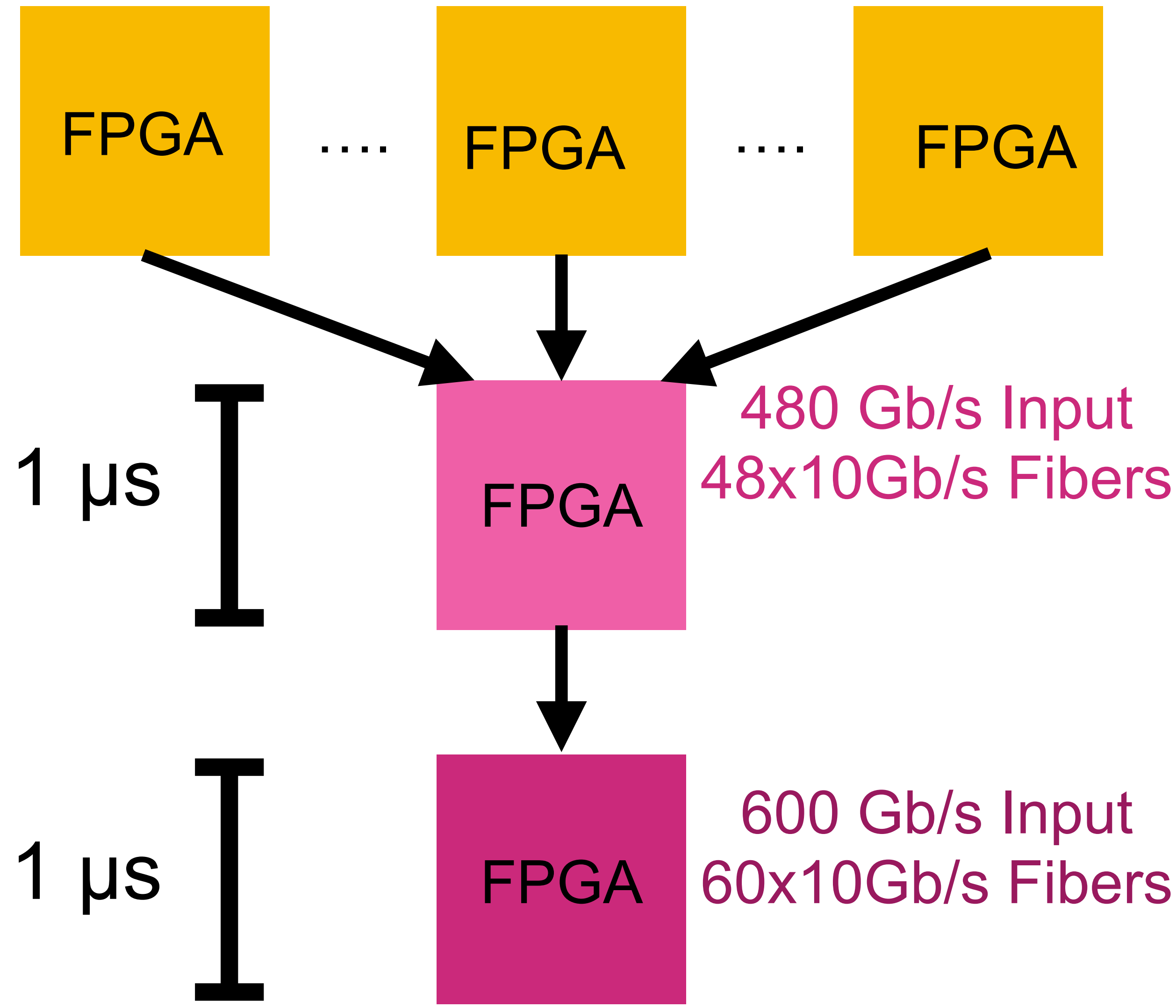
10 Gb/s

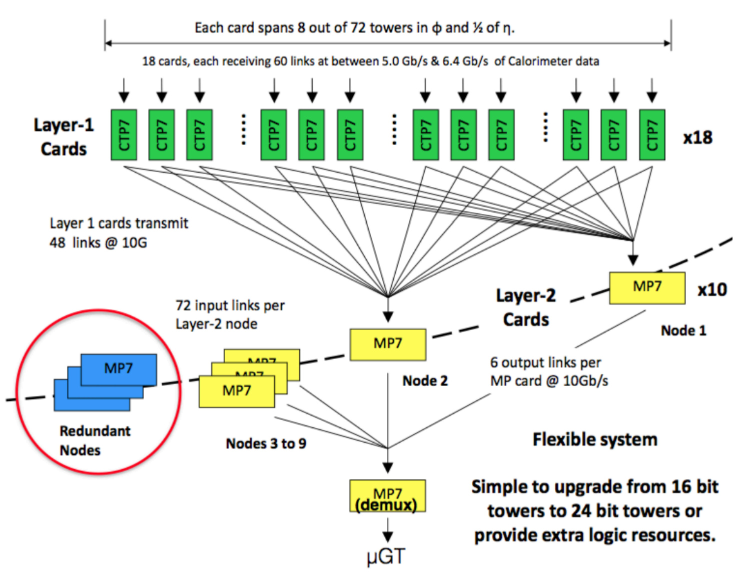
Real-time AI on every LHC Collisions

To process this data we need Deep Neural Networks on FPGAs in Nanoseconds!

Current (Old) Tech



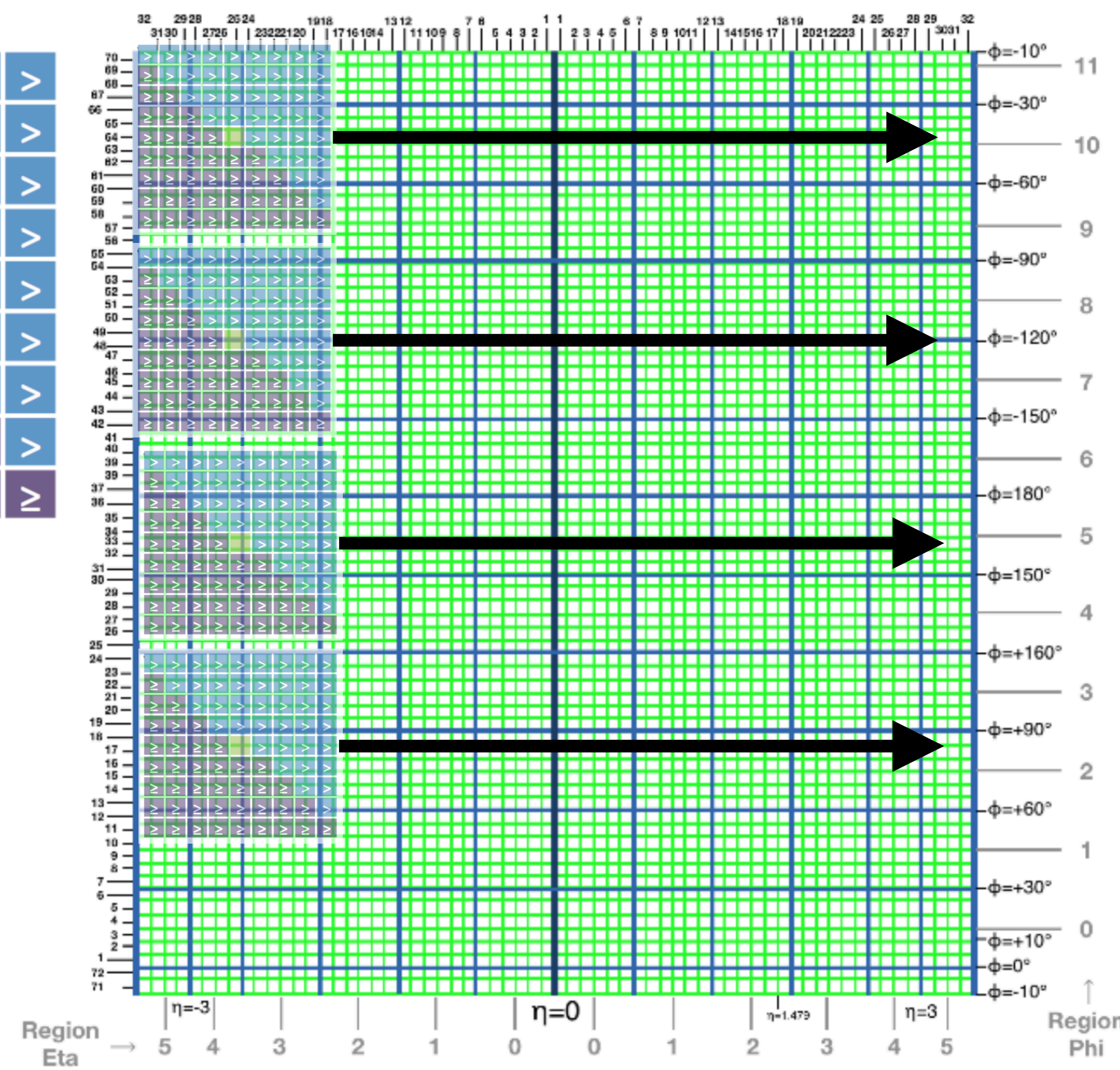




Current System is roughly 100 Virtex7 FPGAs interconnected with Fibers

Current Algos

Algo



Simultaneously scan over calorimeter region a very simple algorithm

FPGA is essential to parallelize & deal w/ enormous bandwidth

Algorithms have traditionally been simple due to the size of the FPGAs + RTL code

Real-Time Deep Learning

- We only have 1μs or less for the inference time
  - We need to run the networks at a rate > 40 MHz (II < 25ns)
  - Forced us to re-think DNN hardware implementations
- This work led us to the project:

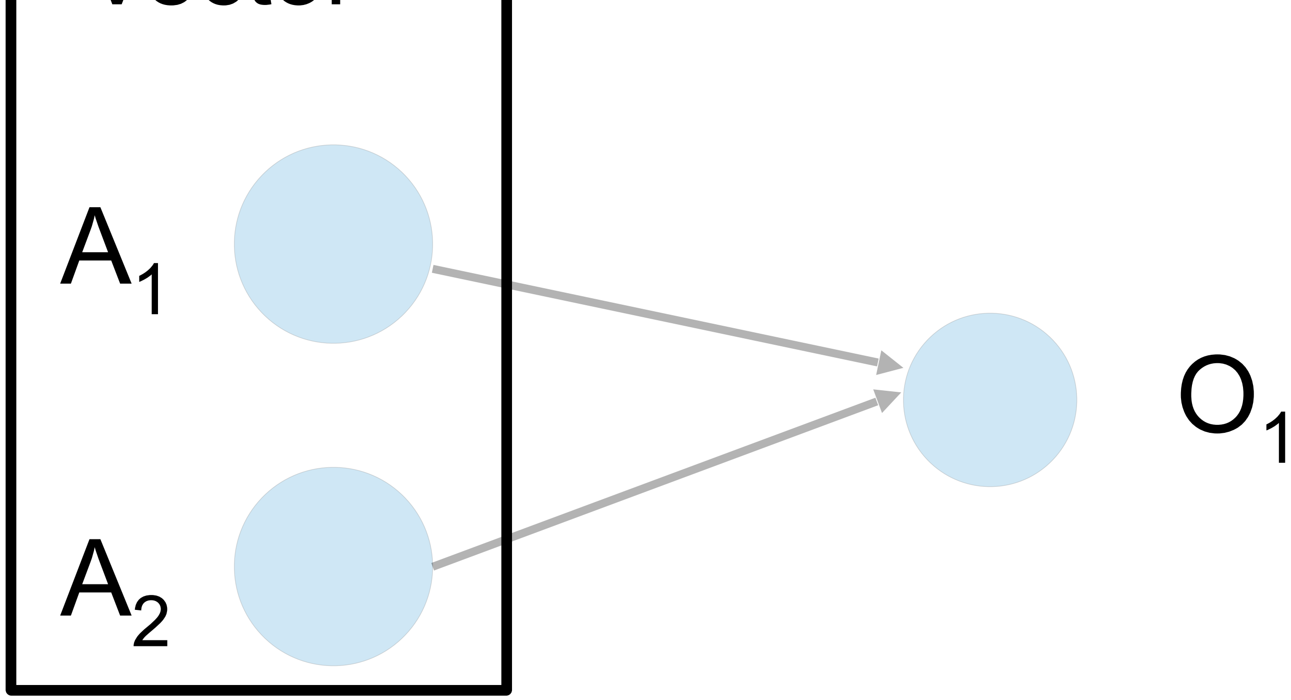
D. Rankin

S. Han

hls4ml

arxiv:1804.06913

Matrix Mult in Math



How can we parallelize this?

$$\varphi(A_1W_{11}+A_2W_{21}+B_1)=O_1$$

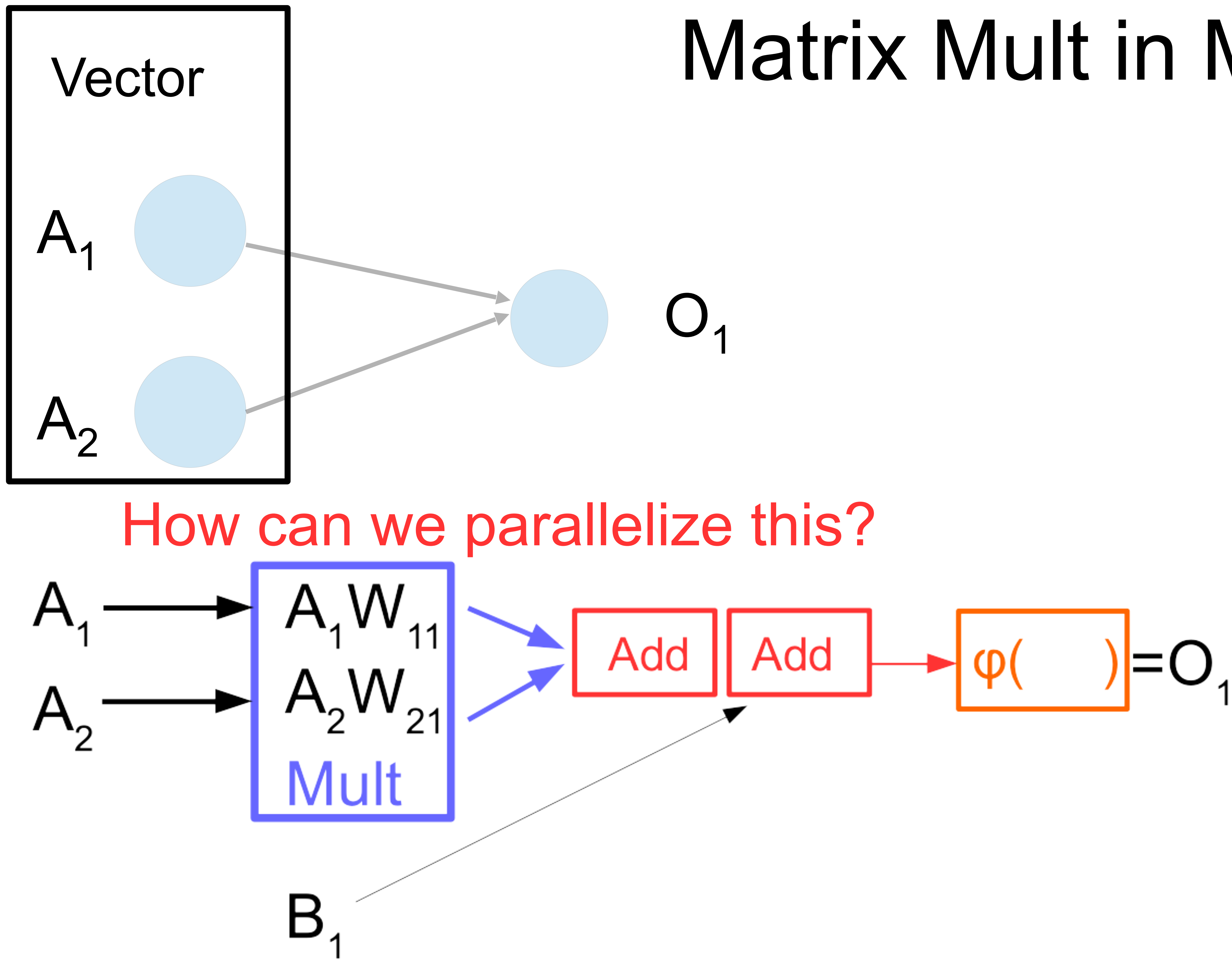
Vector Addition

Matrix Multiplication

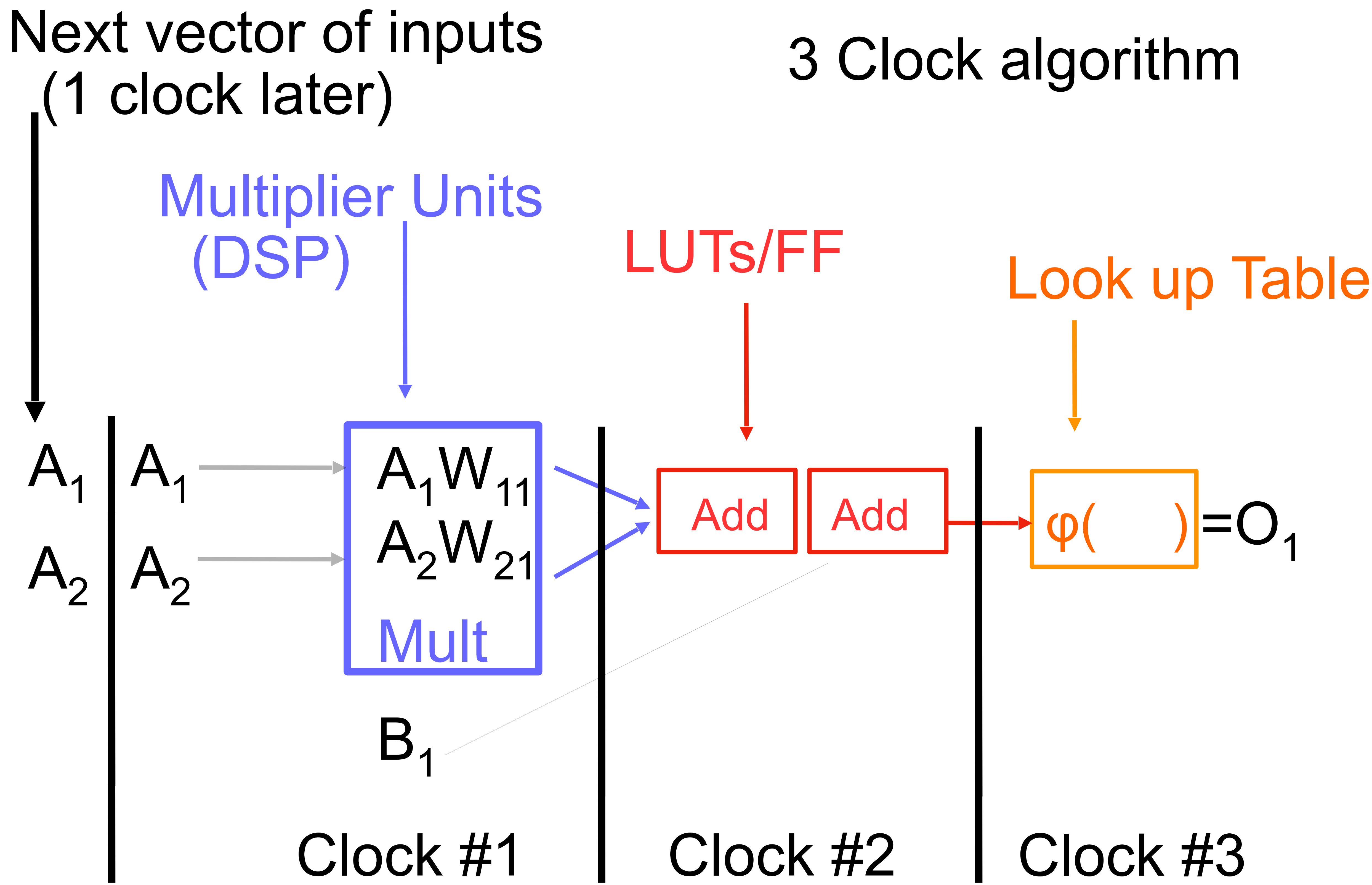
Activation function

arxiv:1804.06913

Matrix Mult in Math

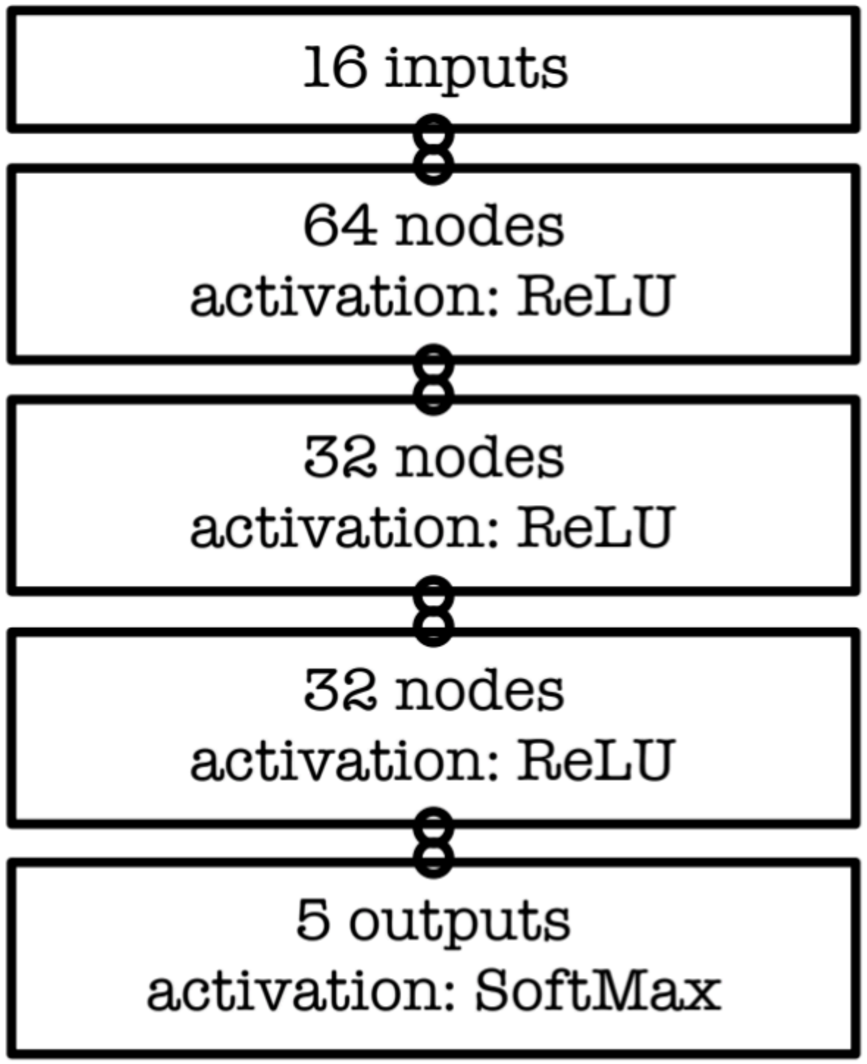


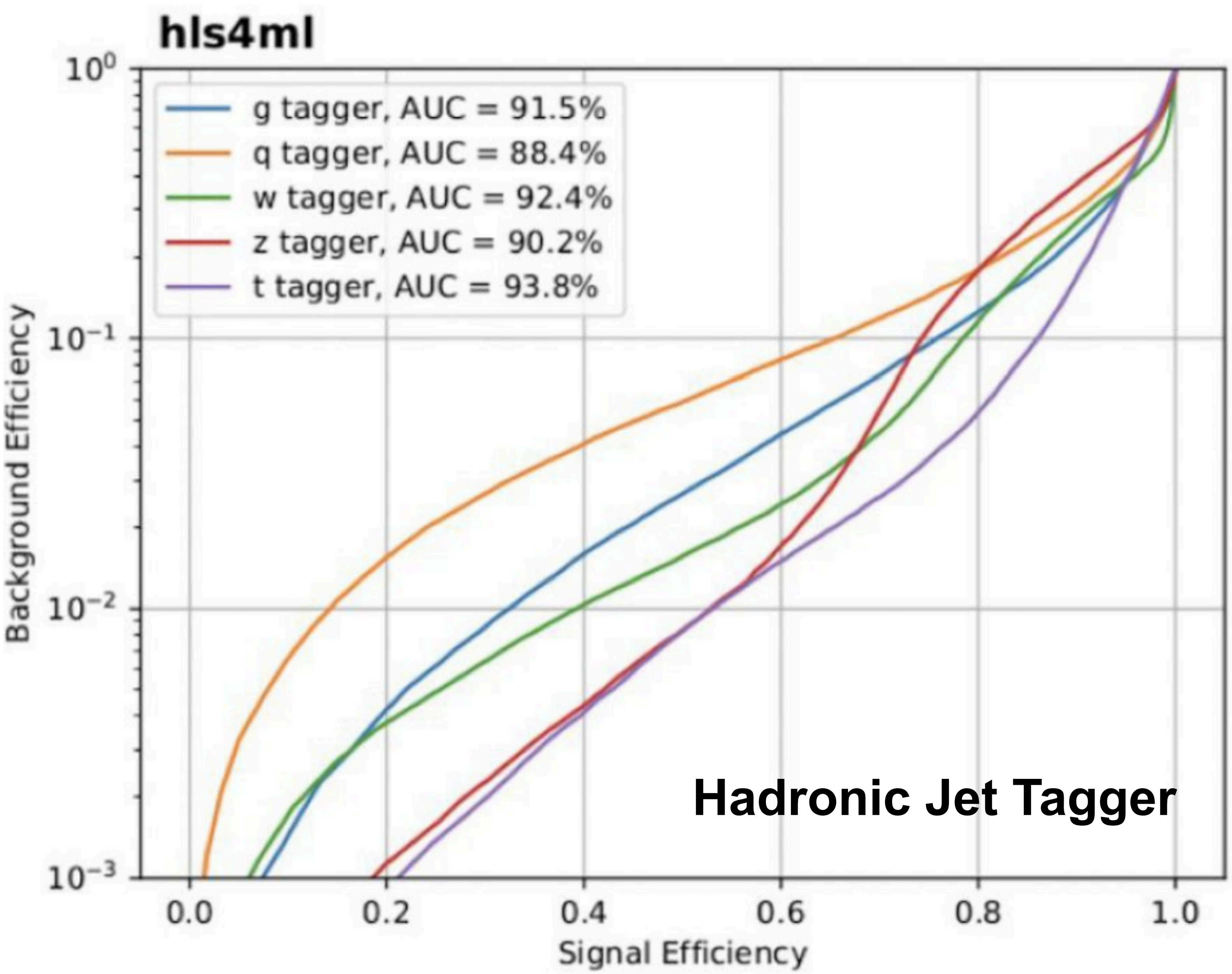
Matrix Mult in an FPGA



Results subject to precision outputs

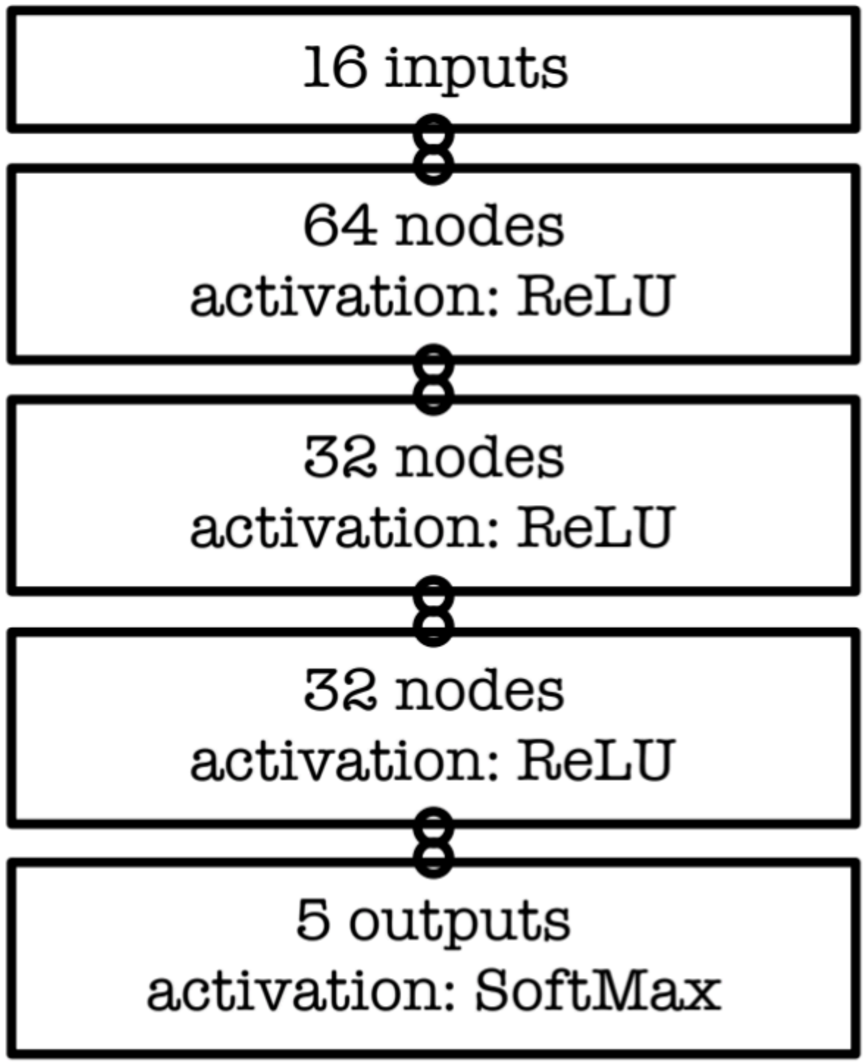
A full benchmark example

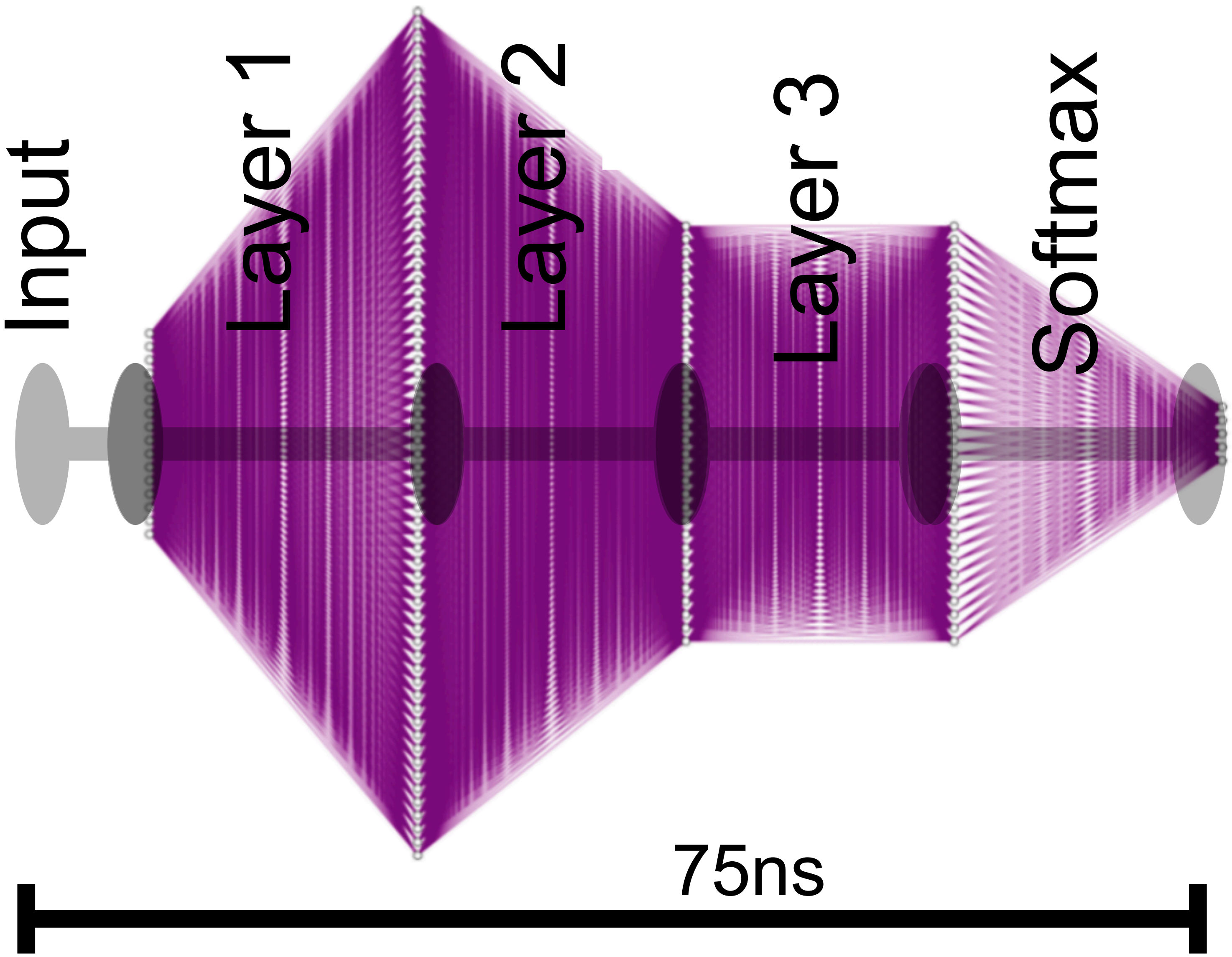




This network has an II of 1 clock, being run constantly
It has 4.3k weights and 4.3k DSPs at II=1

A full benchmark example





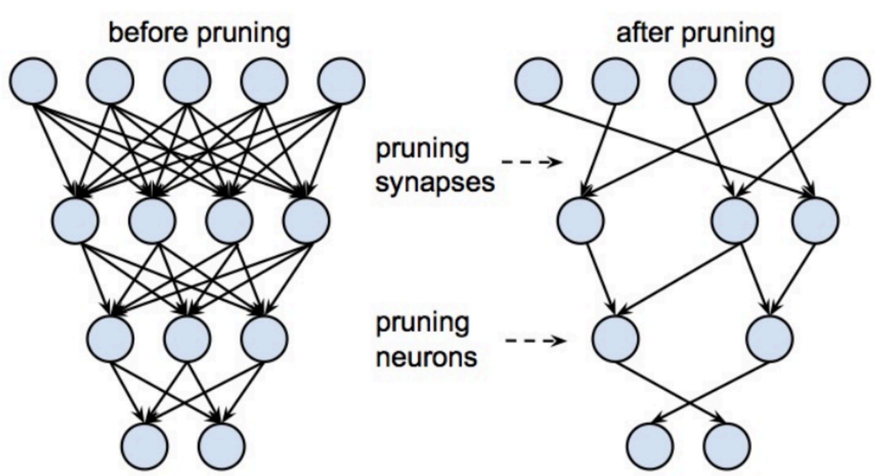
This network has an II of 1 clock, being run constantly
It has 4.3k weights and 4.3k DSPs at II=1

How can we reduce resources?

Focus on 3 ways to cut down resources

Is our algorithm overly complex?

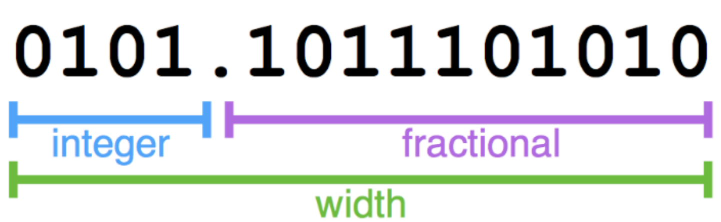
Algorithmic Compression



Are we too precise?

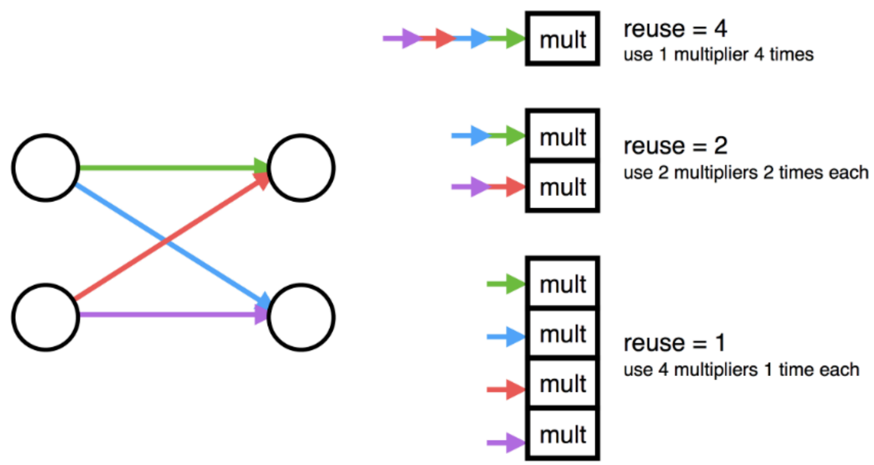
Quantization

ap_fixed<width,integer>



Does it really need to be this fast?

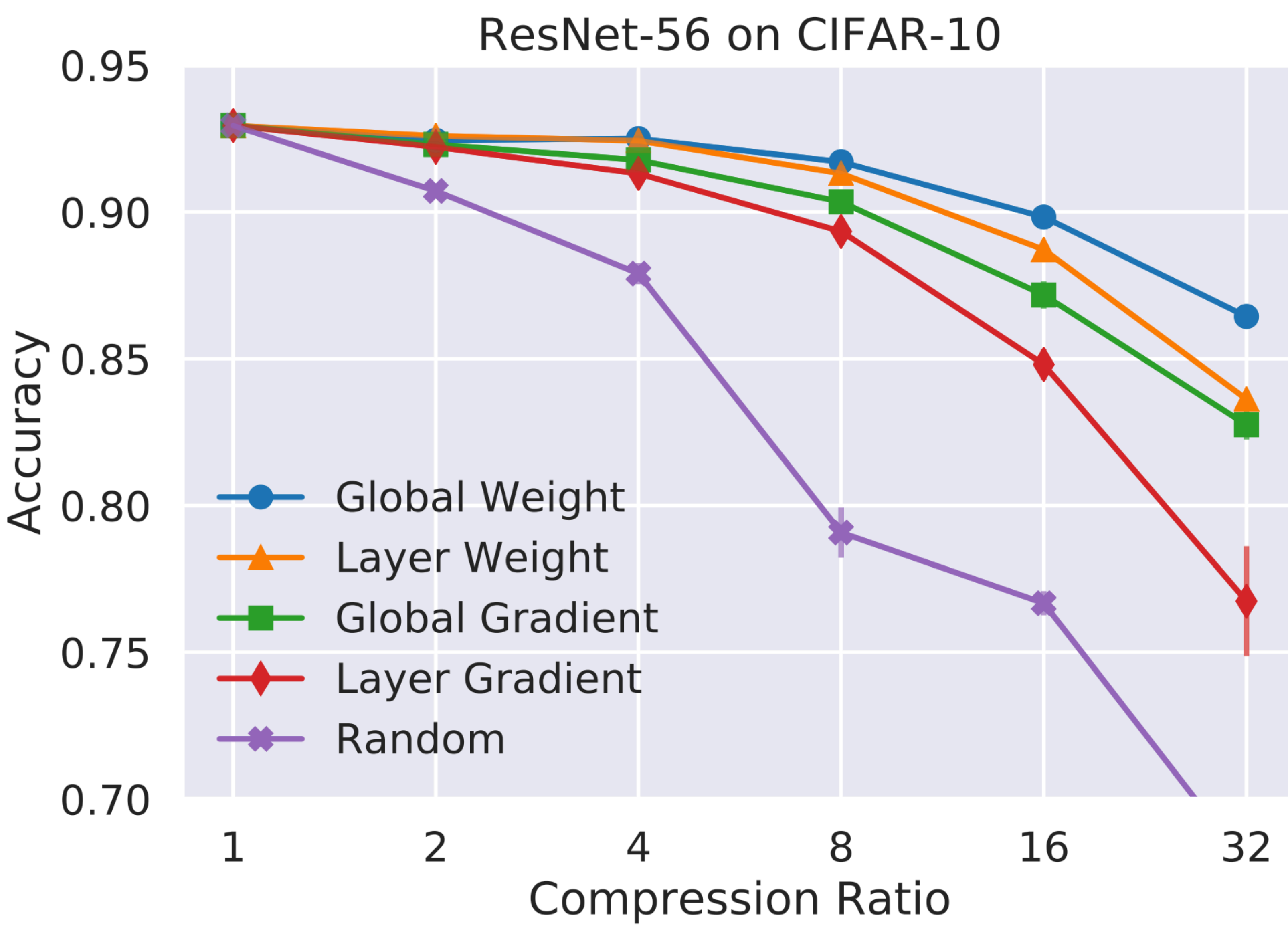
Reuse Factor



arxiv:1804.06913

Algorithm Compression

- Compression is a critical aspect to reduce ML
- A suprising amount of weights in an NN are irrelevant



Model	Mult(DSP)	LUTs
Before	15%	13%
After	0%	1%

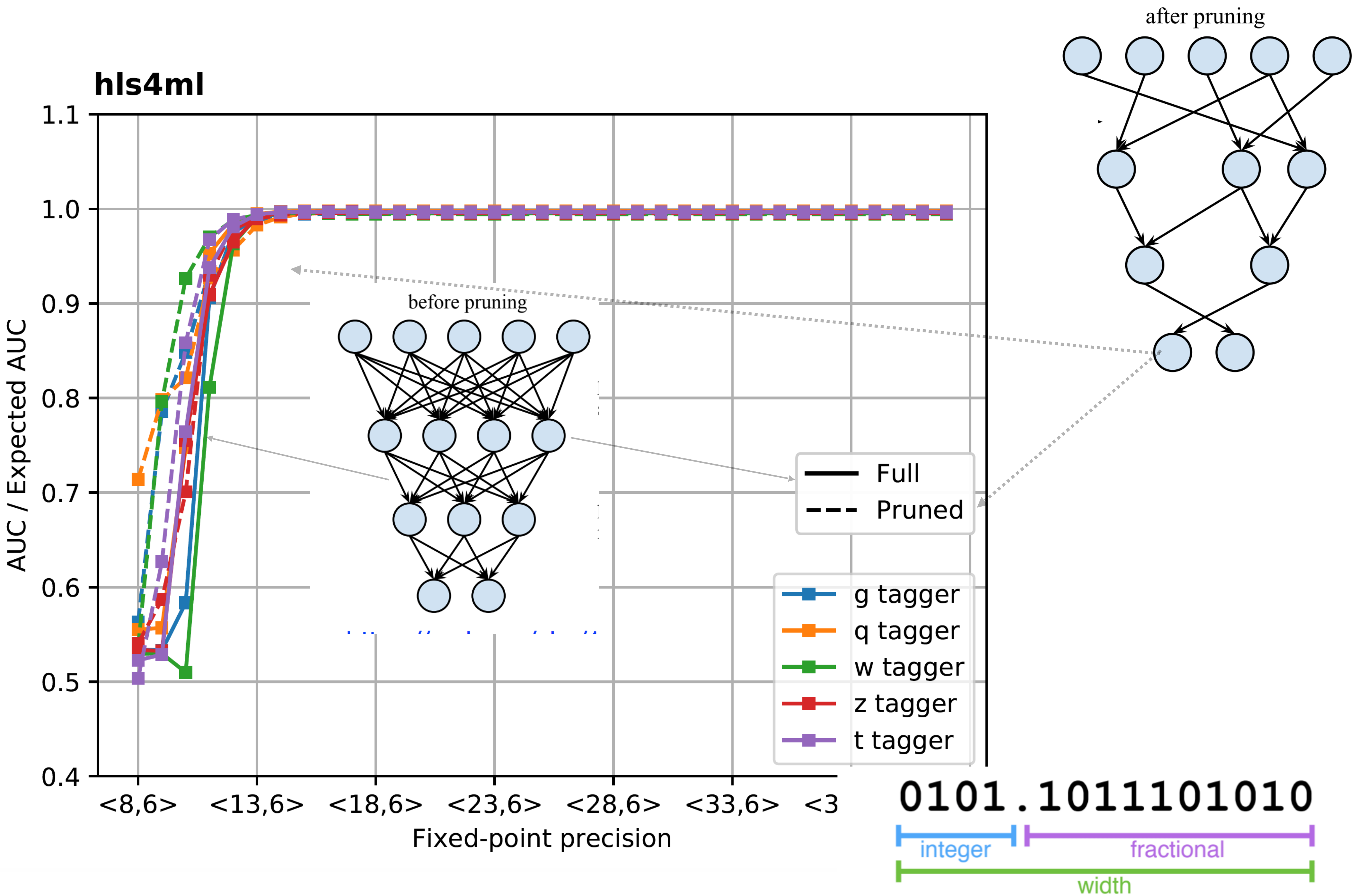
Same Performance

Smaller Latency (50→40ns)

Dramatic Compression

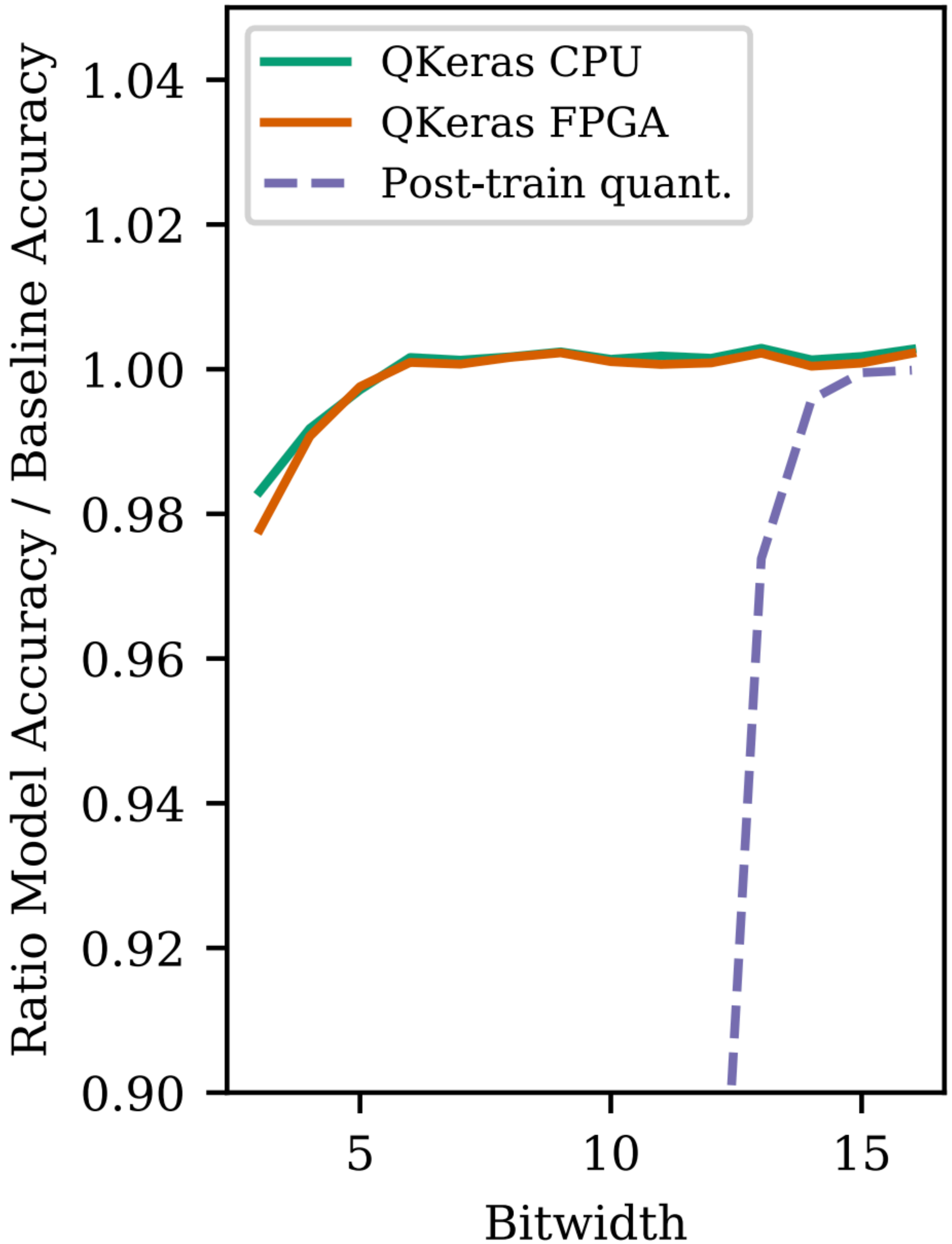
arxiv:1804.06913

Quantization



<Total bit width, integer bits above decimal>

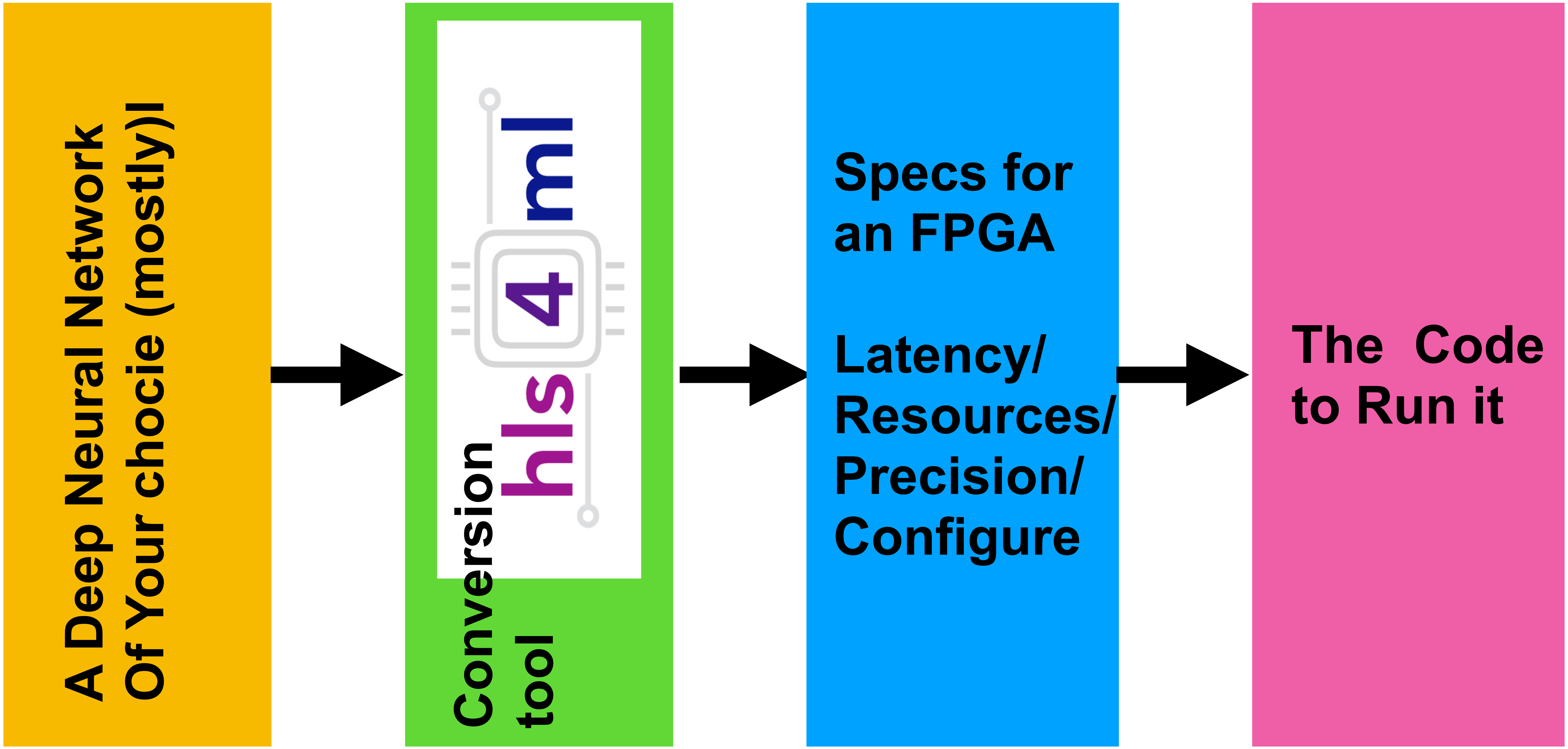
Algorithm Compression



Fixed precision training
Weight pruning shrinks networks

arxiv:2103.05579

A Compiler than can do it

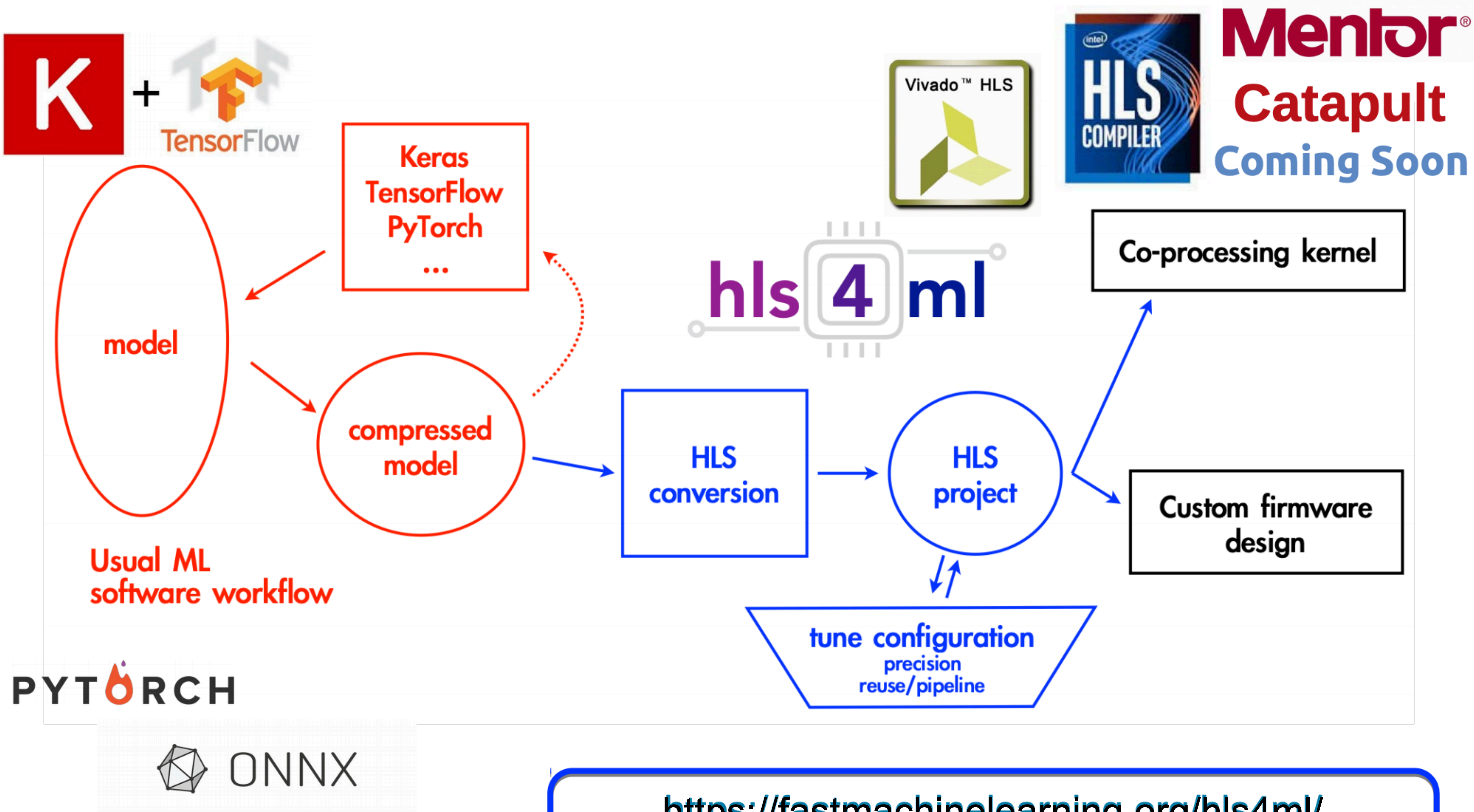


There are now a few tools
See Tae Min's Talk for another tool!

https://fastmachinelearning.org/hls4ml/

A Compiler than can do it

```
python keras-to-hls.py -c keras-config.yml
```



https://fastmachinelearning.org/hls4ml/

Flexibility

- Many different types of collisions are analyzed at LHC
  - A diverse set of algorithms are required
  - There is no one size fits all NN that will solver our problems
- With HLS4ML we have continued to expand options
  - HLS has allowed for quick development

Algorithms

MLPs

CNNs

RNNs(LSTM/GRU)

Binary & Ternary NNs

Graph NNs(MPNN/GravNet/GarNet)

BDTs

arxiv:2003.06308
arxiv:2002.02534
arxiv:2008.03601
arxiv:2006.10159

Not yet in official release

Backends

Xilinx Vitis HLS

Intel HLS Quertus

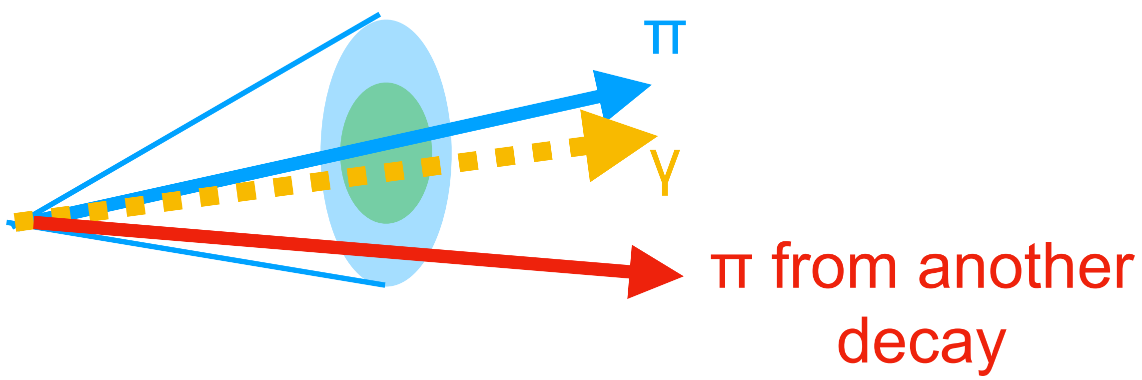
Mentor Catapult HLS

Intel OneAPI

Not yet in official release

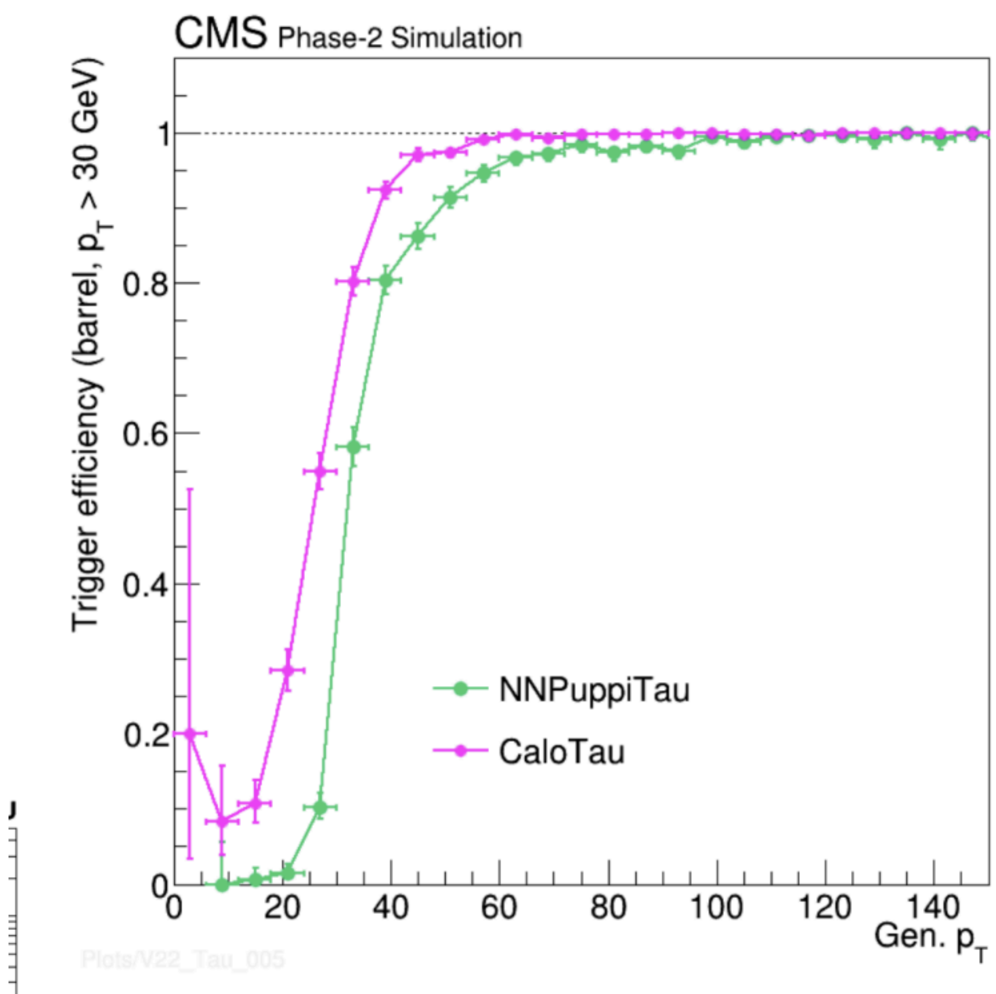
Example #1 Tau Tagging

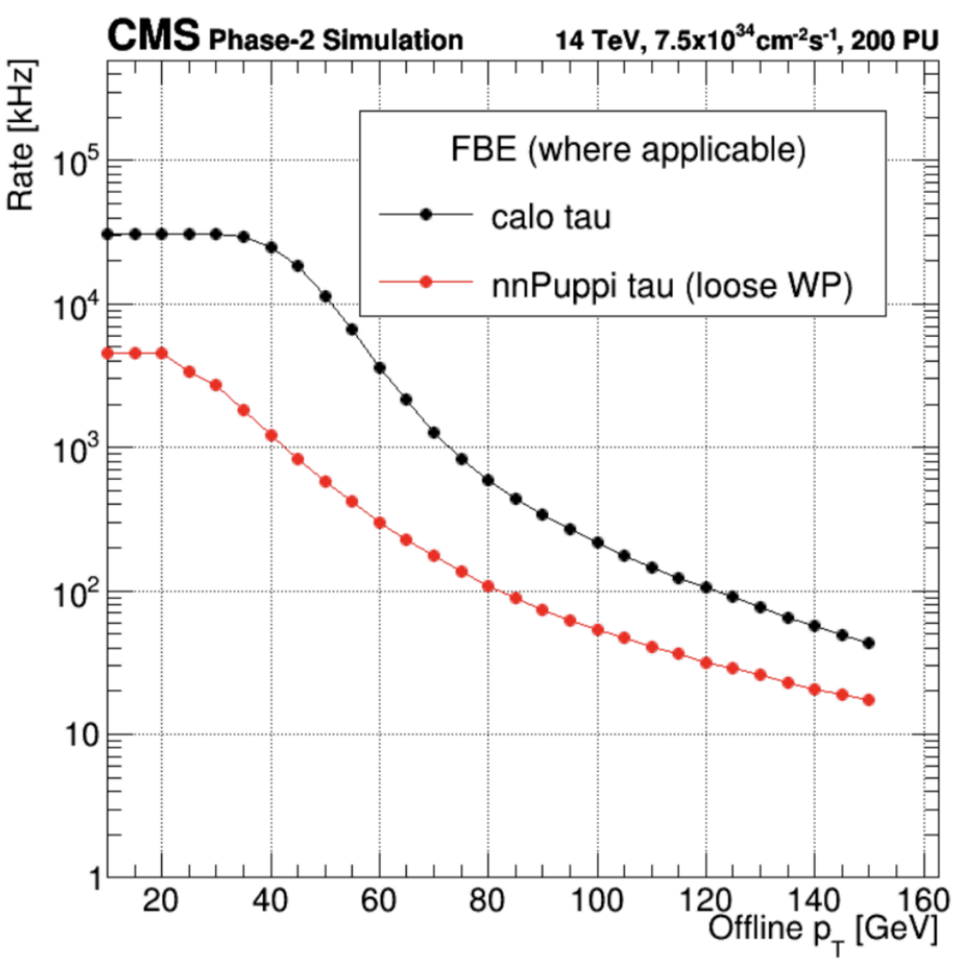
Tau Leptons have complex final states



Tau Lepton can decay to as many as 10 different particles
Background can decay to many more

Neural Network has long been the algorithm of choice to identify Taus

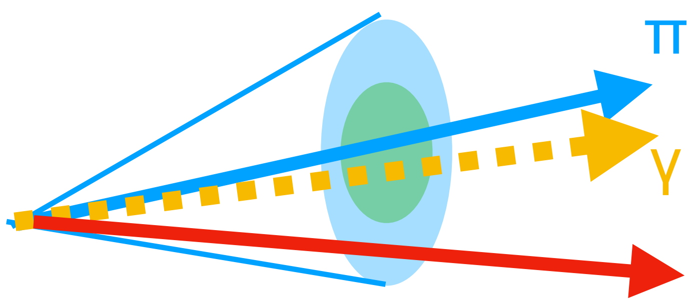




Example #1 Tau Tagging

Algorithm Takes 10 top particles in a cone and runs NN

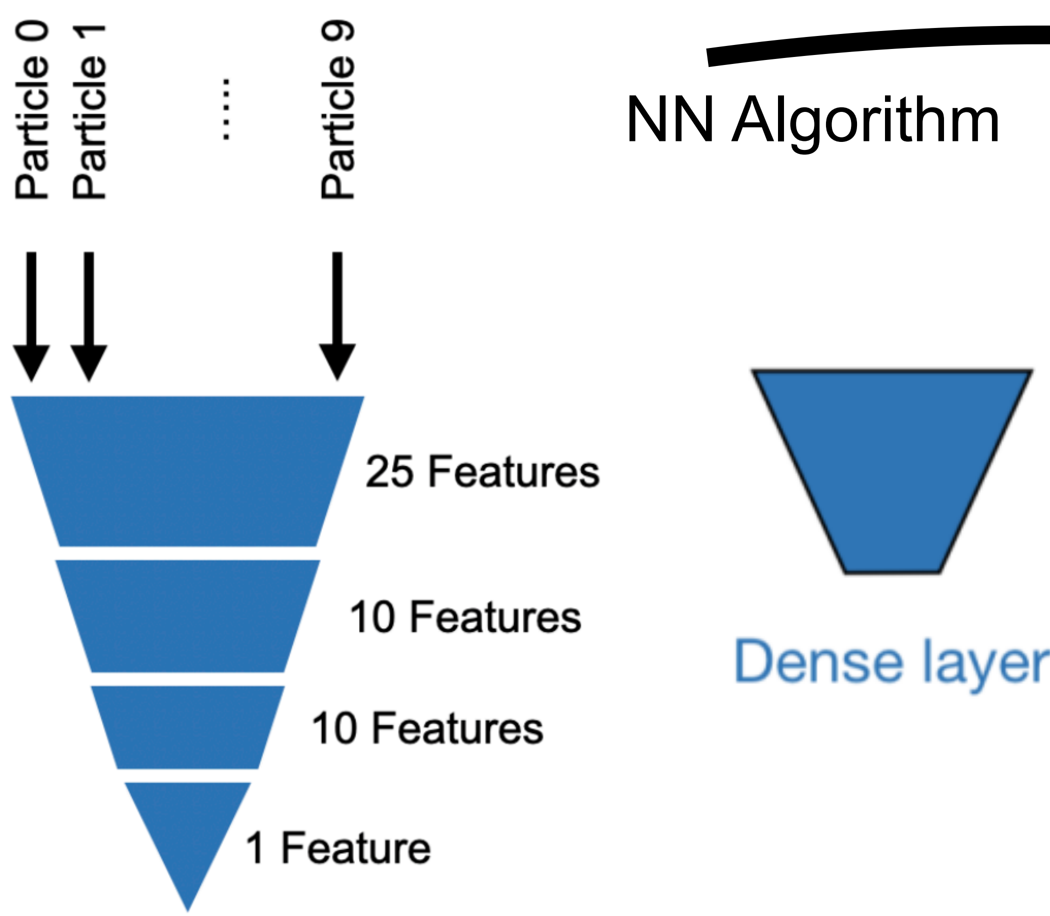
With HLS4ML we can run this algorithm in 70ns



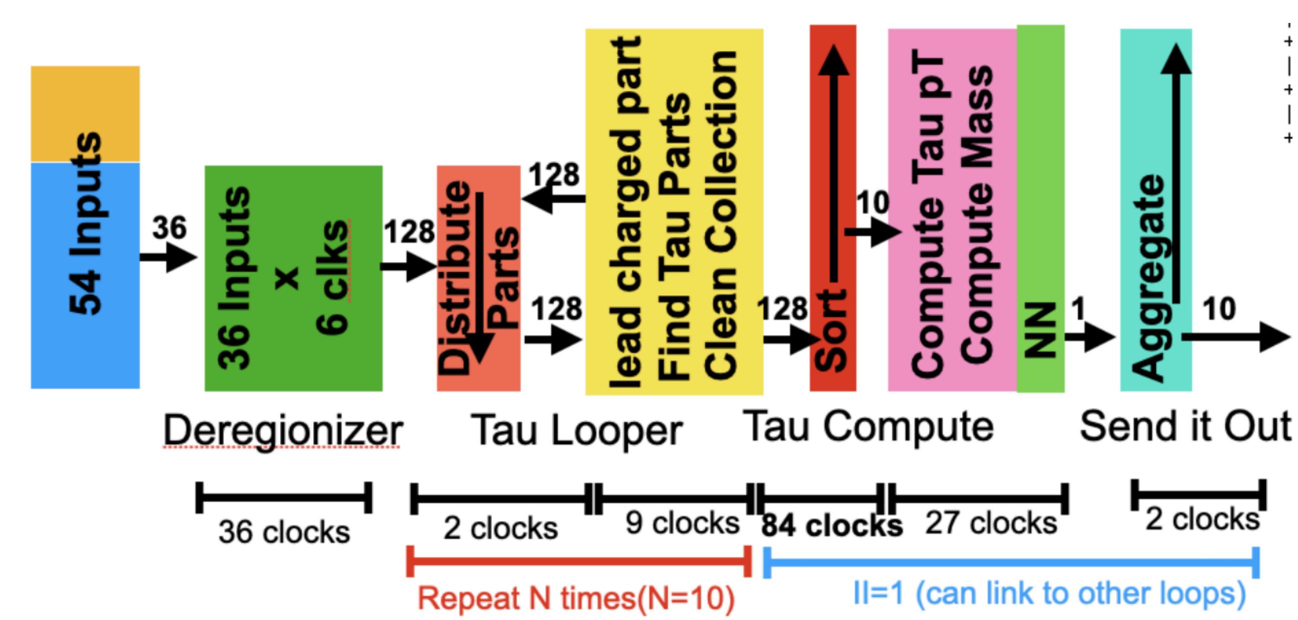
VU9P	DSP	FF	LUTs	BRAM
NNTau	11%	12%	18%	16%

NN alone is <10% of the FPGA

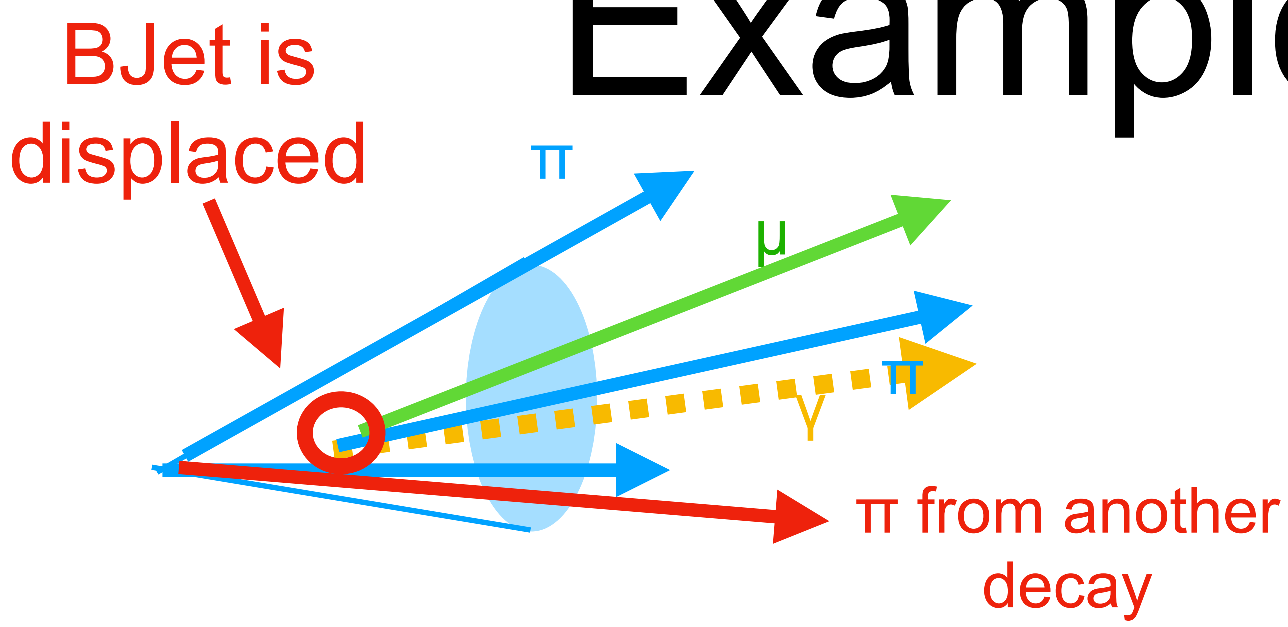
Whole Algorithm Resources



Whole Algorithm on Board

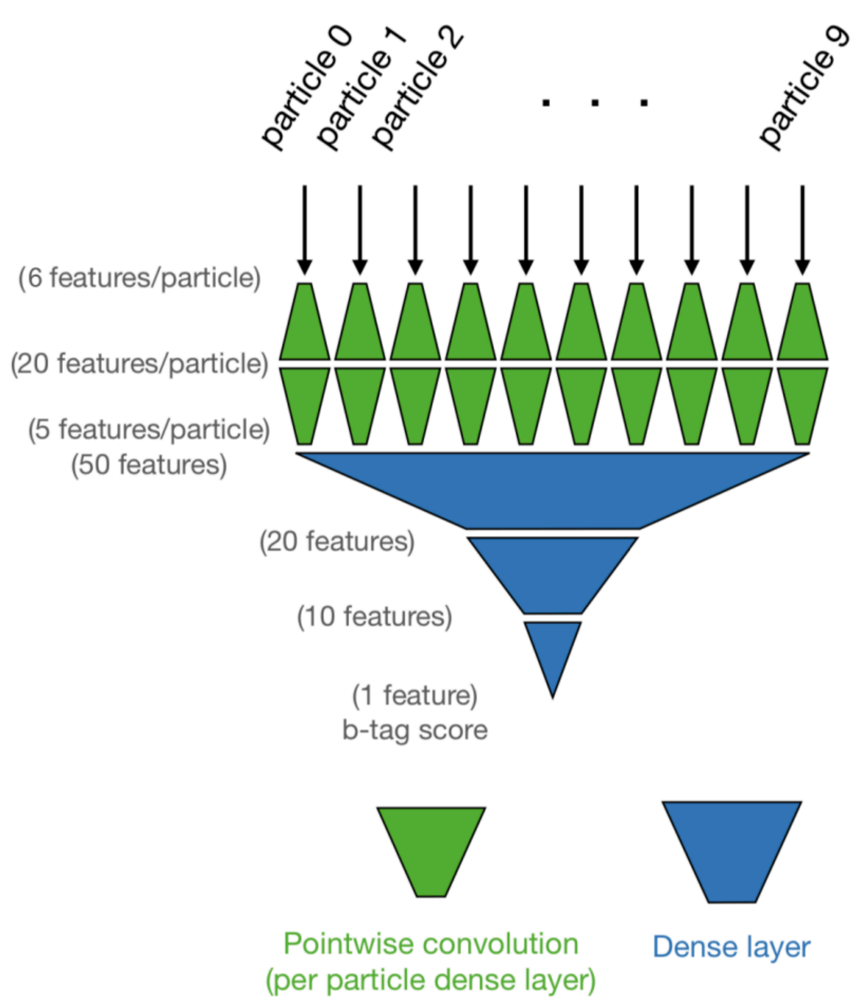


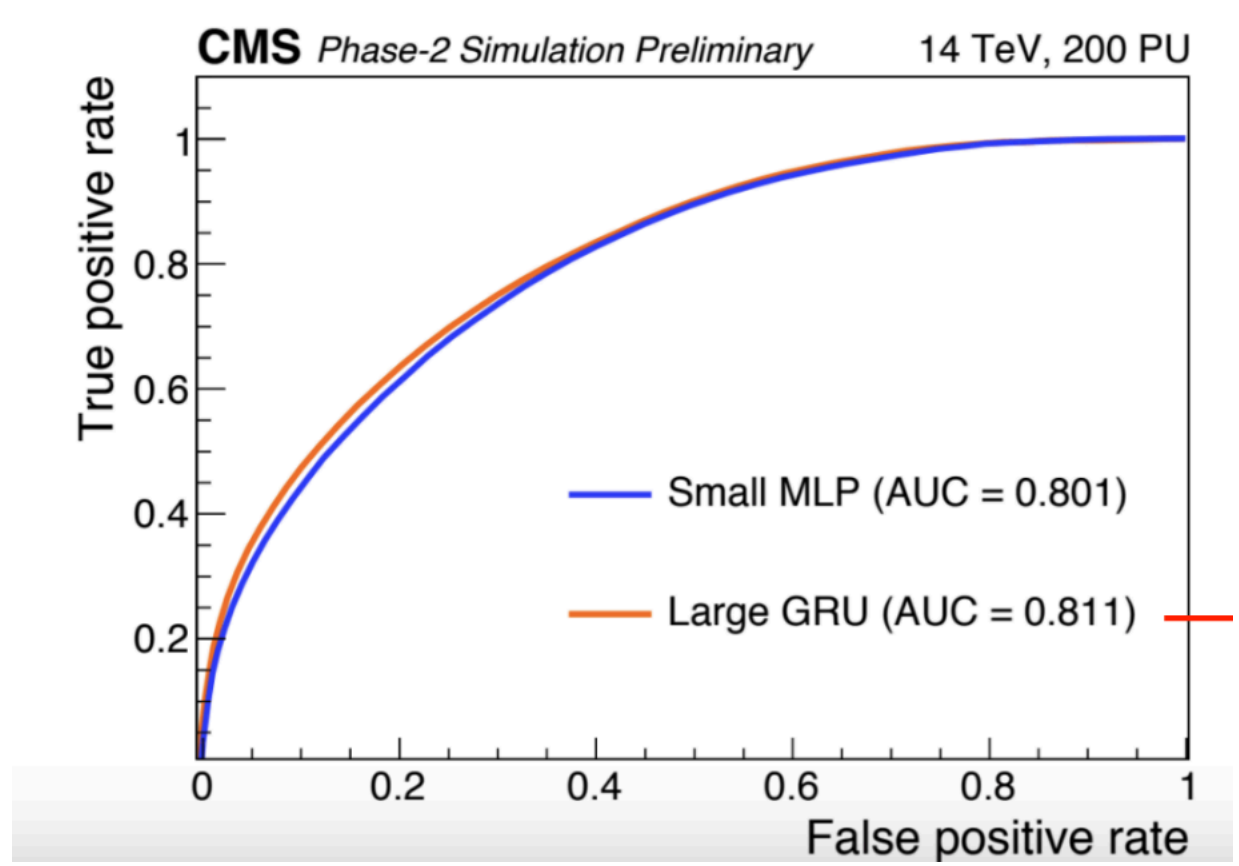
Example #2 BTagging



In addition to taus
B-tagging good ML candidate

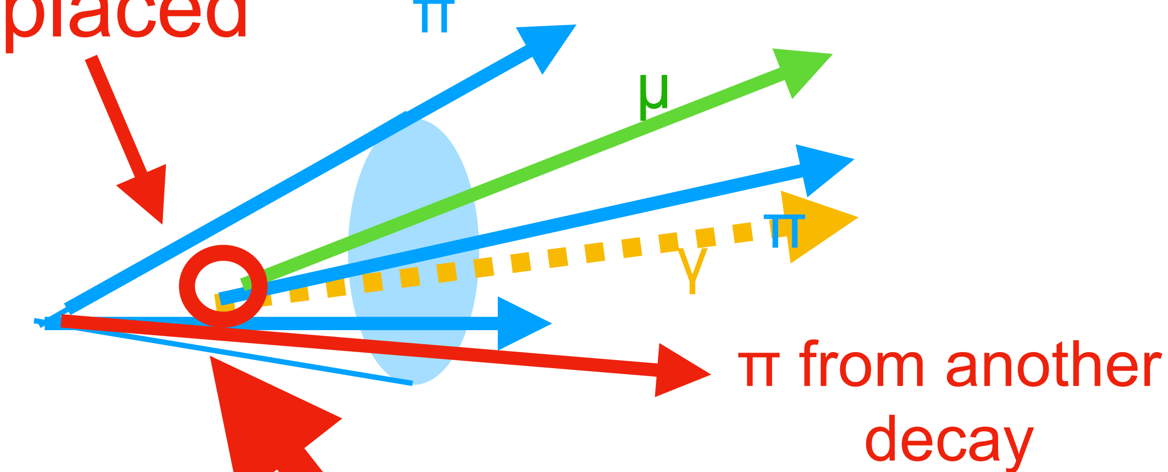
Not obvious CMS Trigger
vertex resolution is large





Example #2 BTagging

BJet is displaced

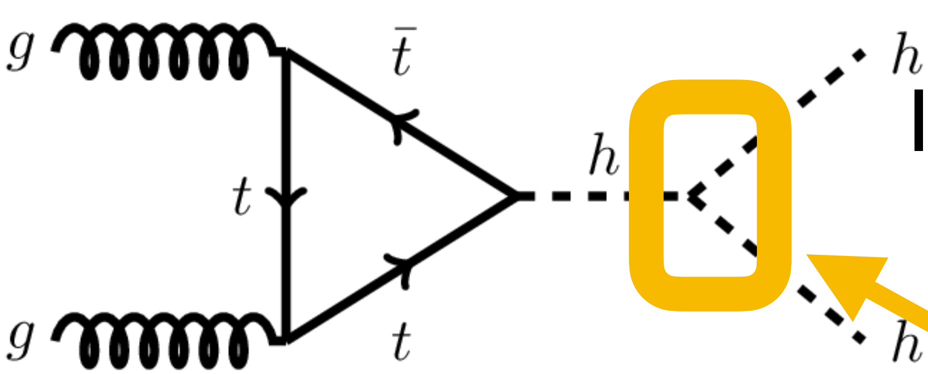


Hard

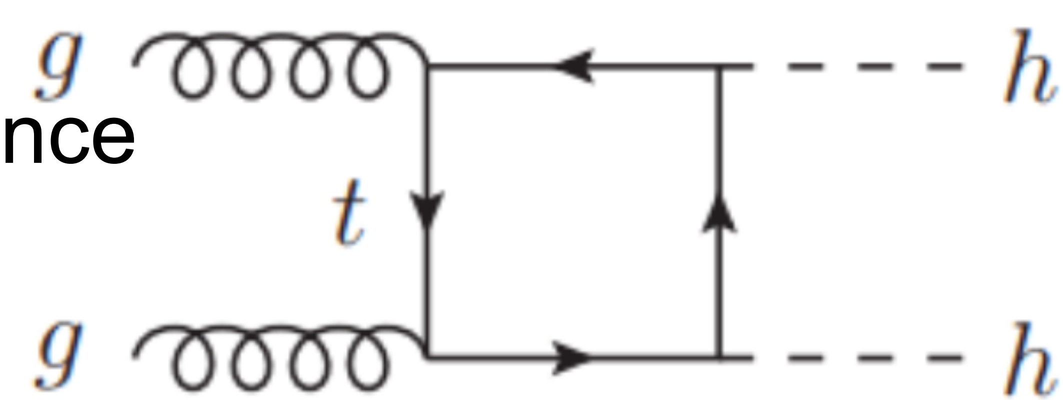
Resolution in Trigger is worse

In addition to taus
B-tagging good ML candidate

Not obvious CMS Trigger
vertex resolution is large



Interference
+

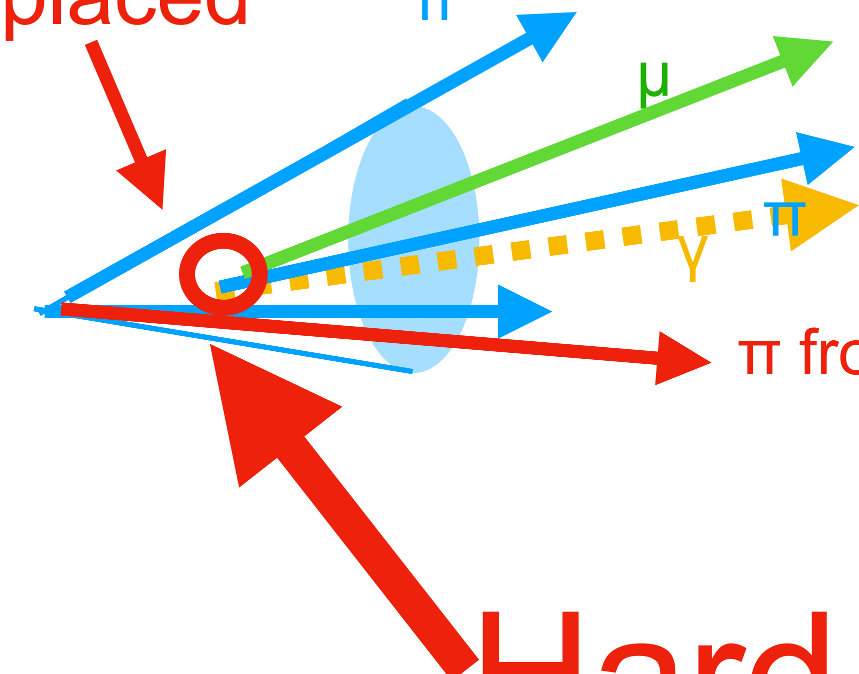


Di Higgs Boson Production

Higgs Self Coupling Term

Example #2 BTagging

BJet is displaced



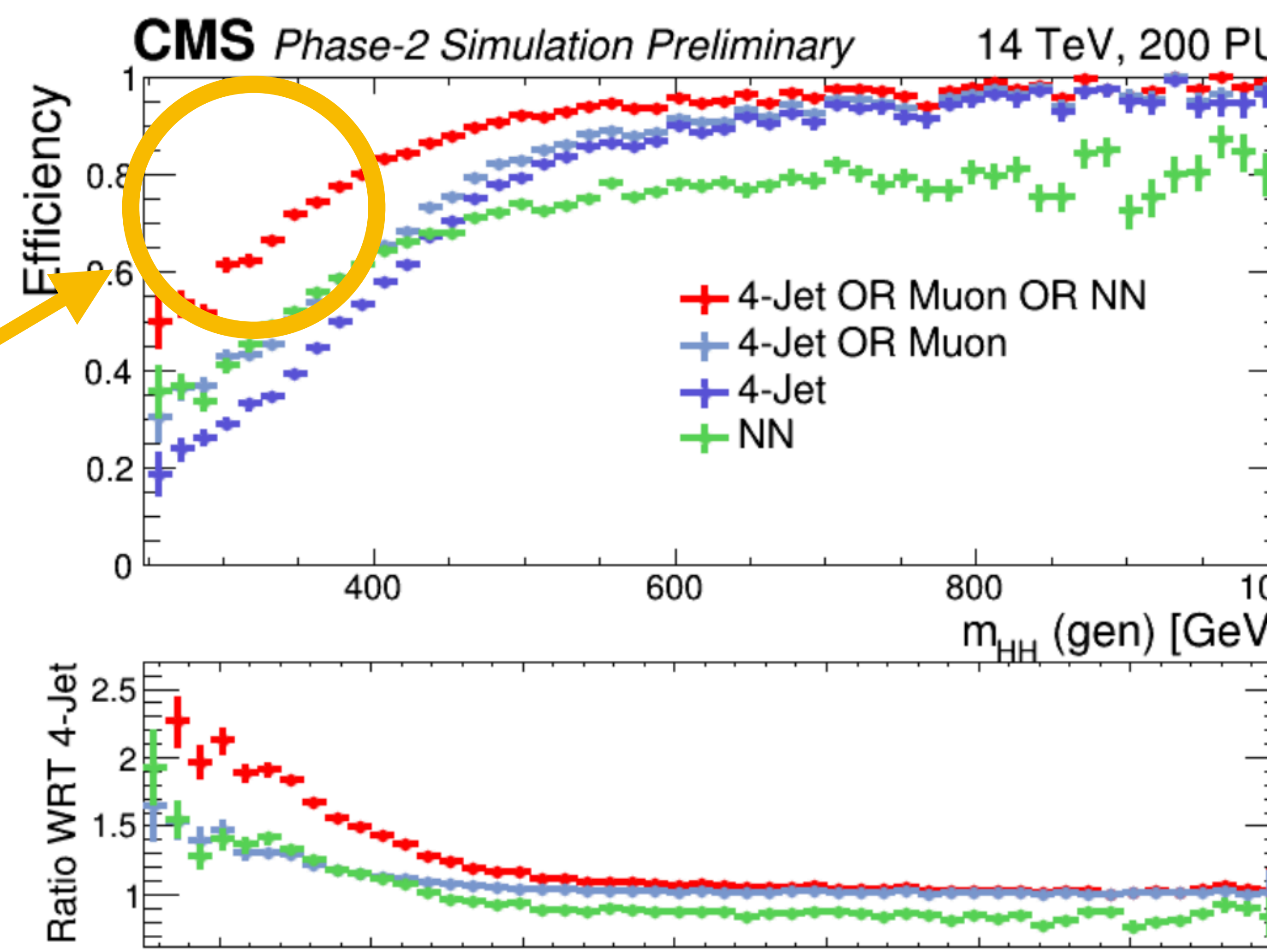
π from another decay

Hard

Resolution in Trigger is worse

In addition to taus
B-tagging good ML candidate

Not obvious CMS Trigger vertex resolution is large

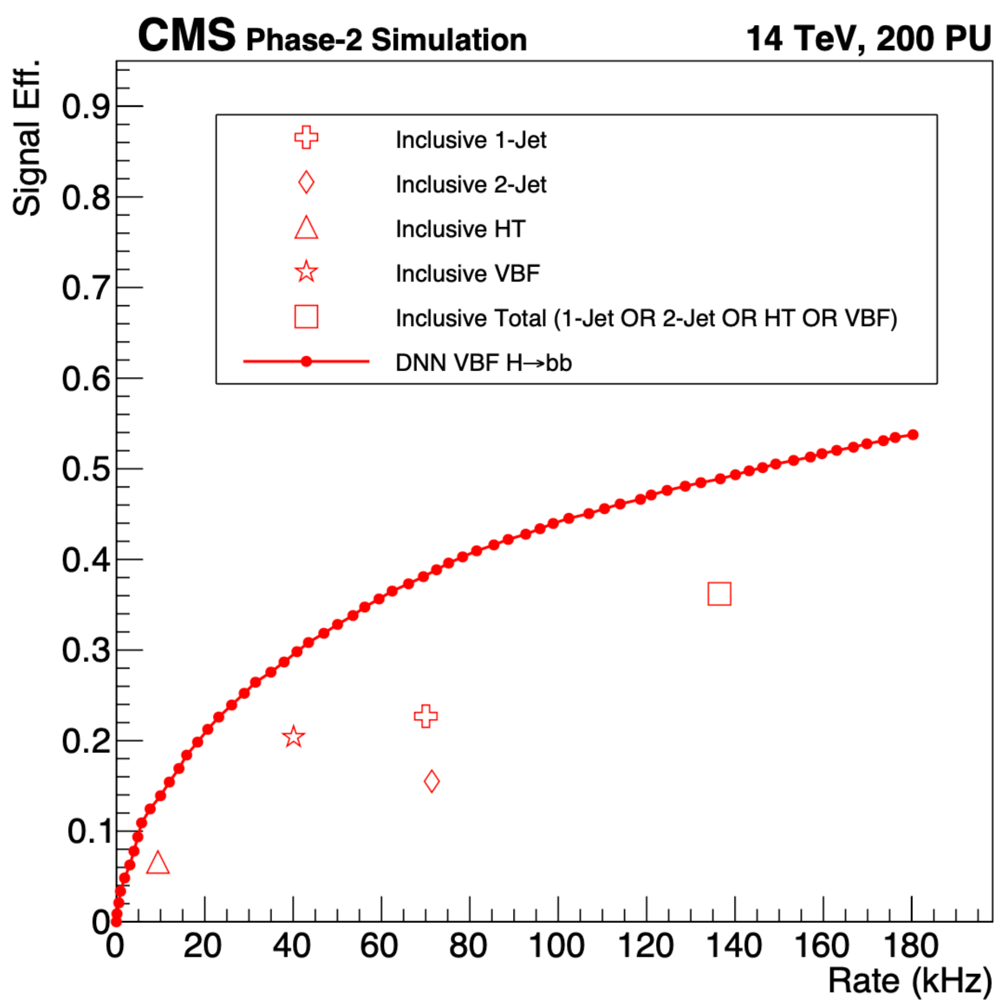


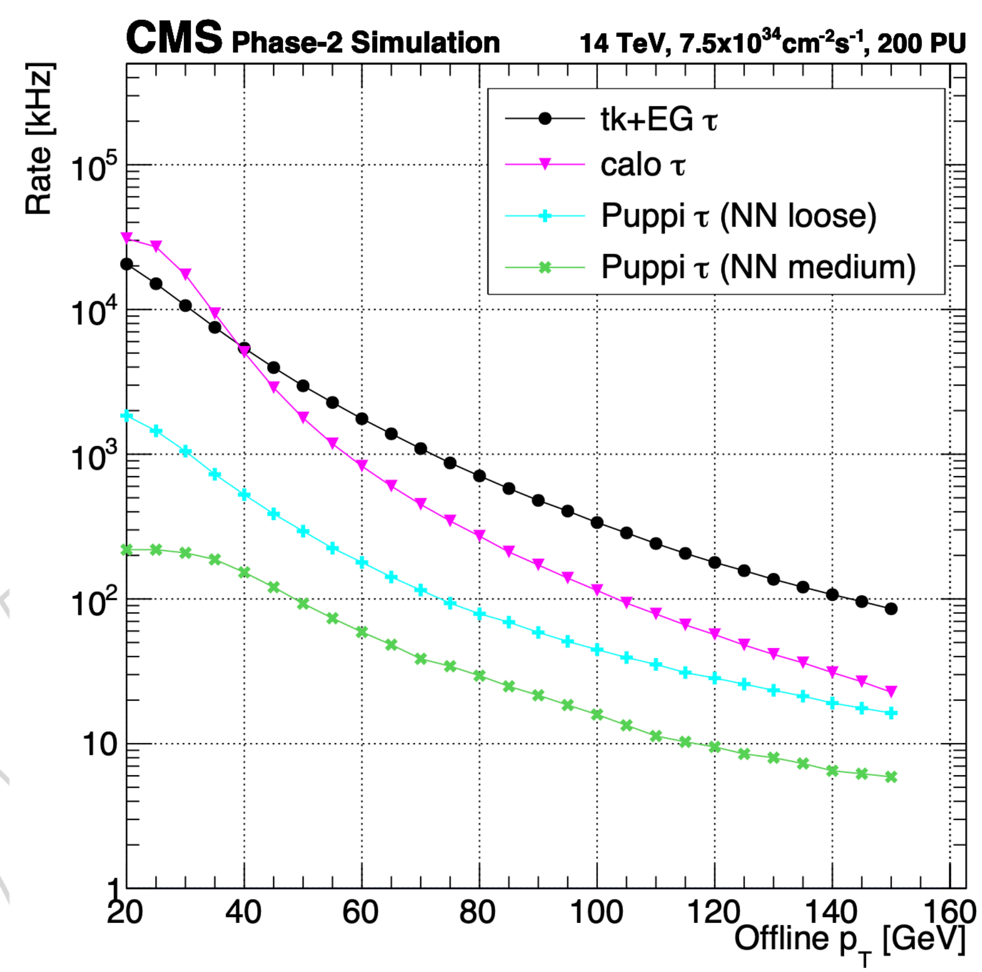
Critical Region
For Self Coupling

Accomplishments

- HLS4ML is rapidly being adopted in our trigger system
  - Will be used in the next running at the LHC
- We already see a number of substantial improvement

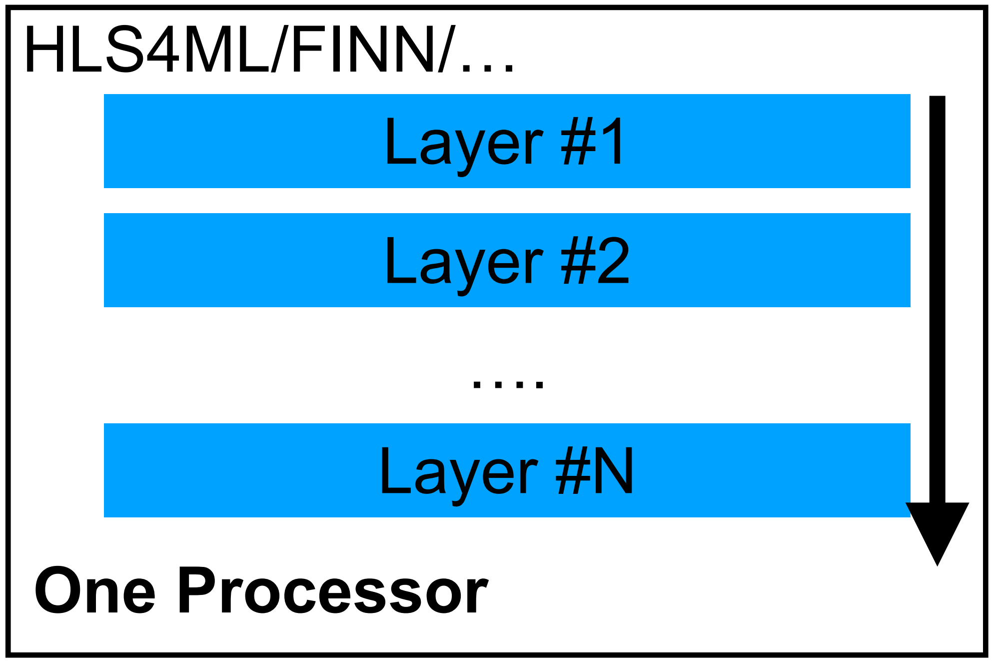
2-5 times More Higgs bosons with the same data rates



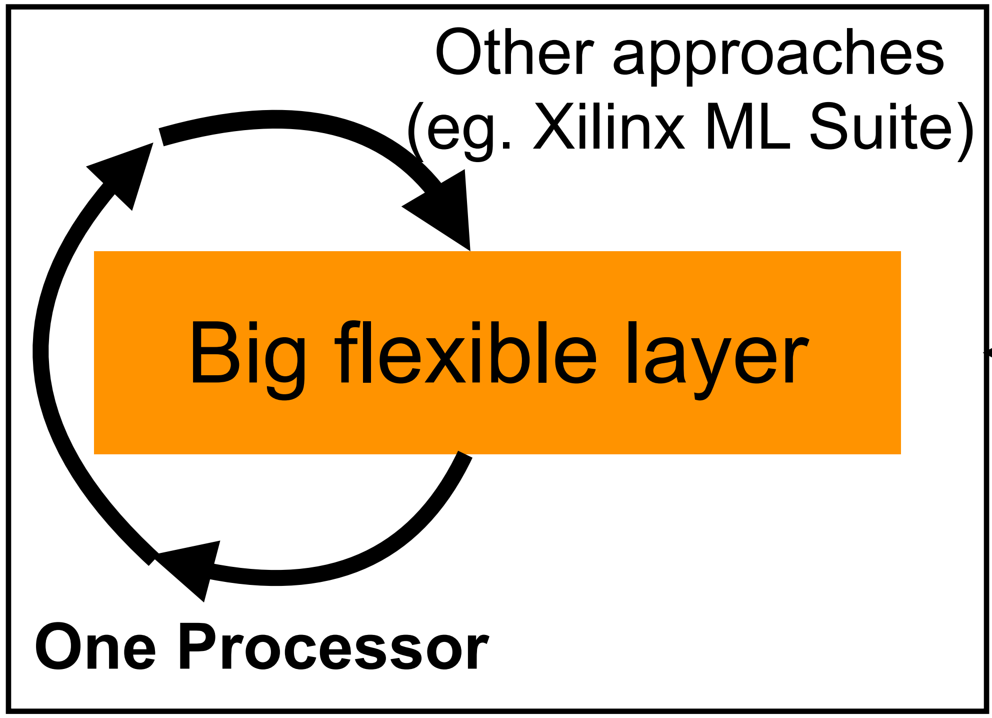


Other Deep Learning Models

- HLS4ML differs from other ML models



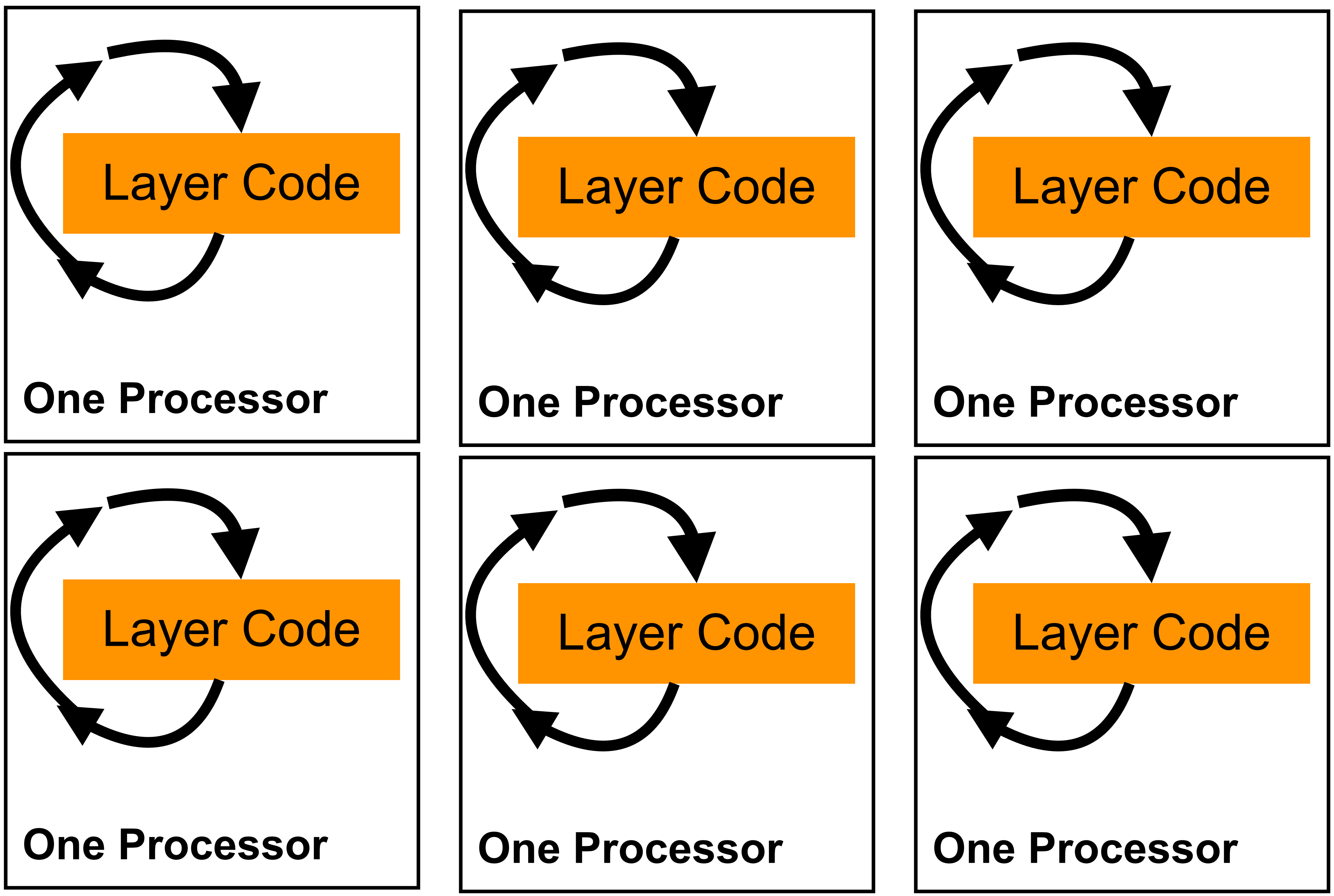
Good for small models where you need ultra low latency and ultra high throughput



Good for very large models where you can’t fit the whole algorithm on the processor logic

How does a GPU do this?

- GPU is about even more standardization

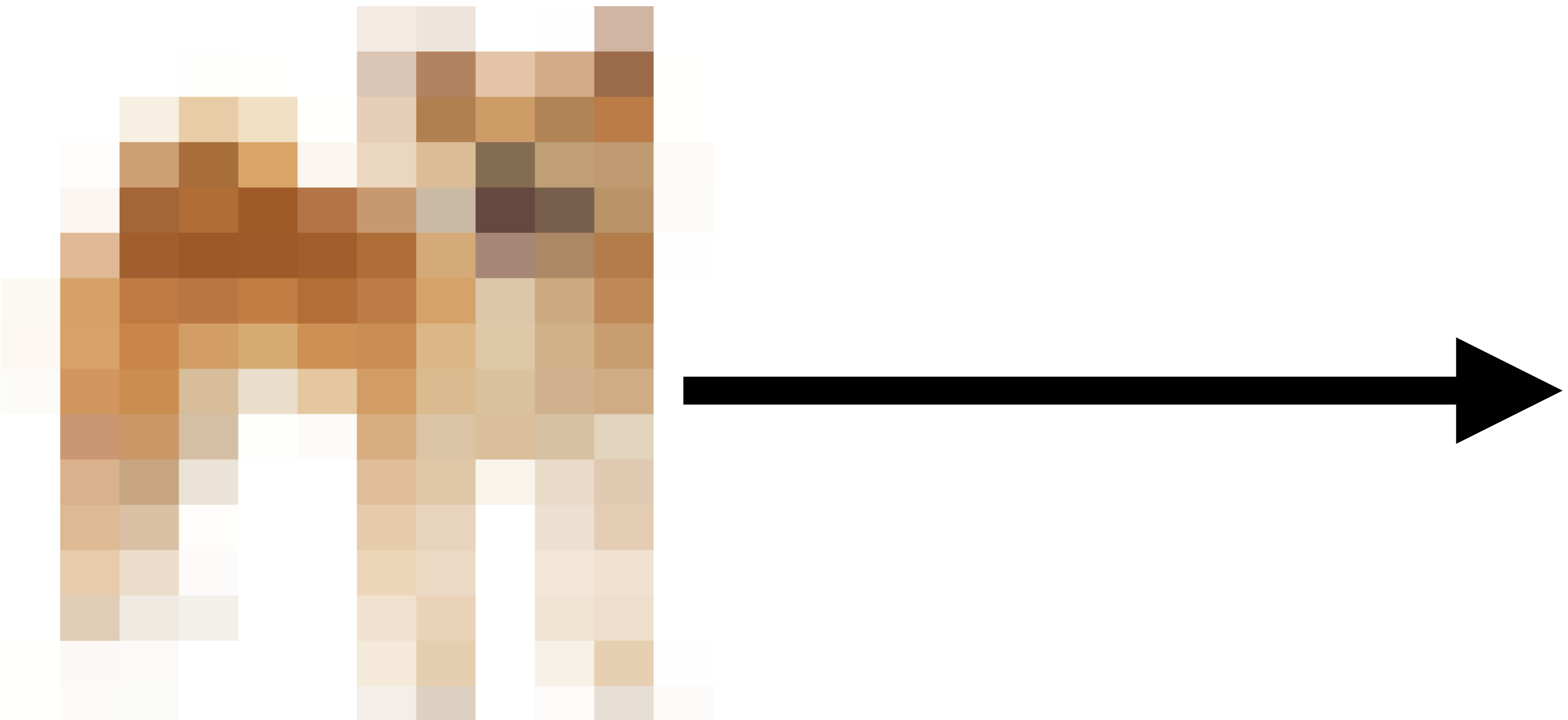


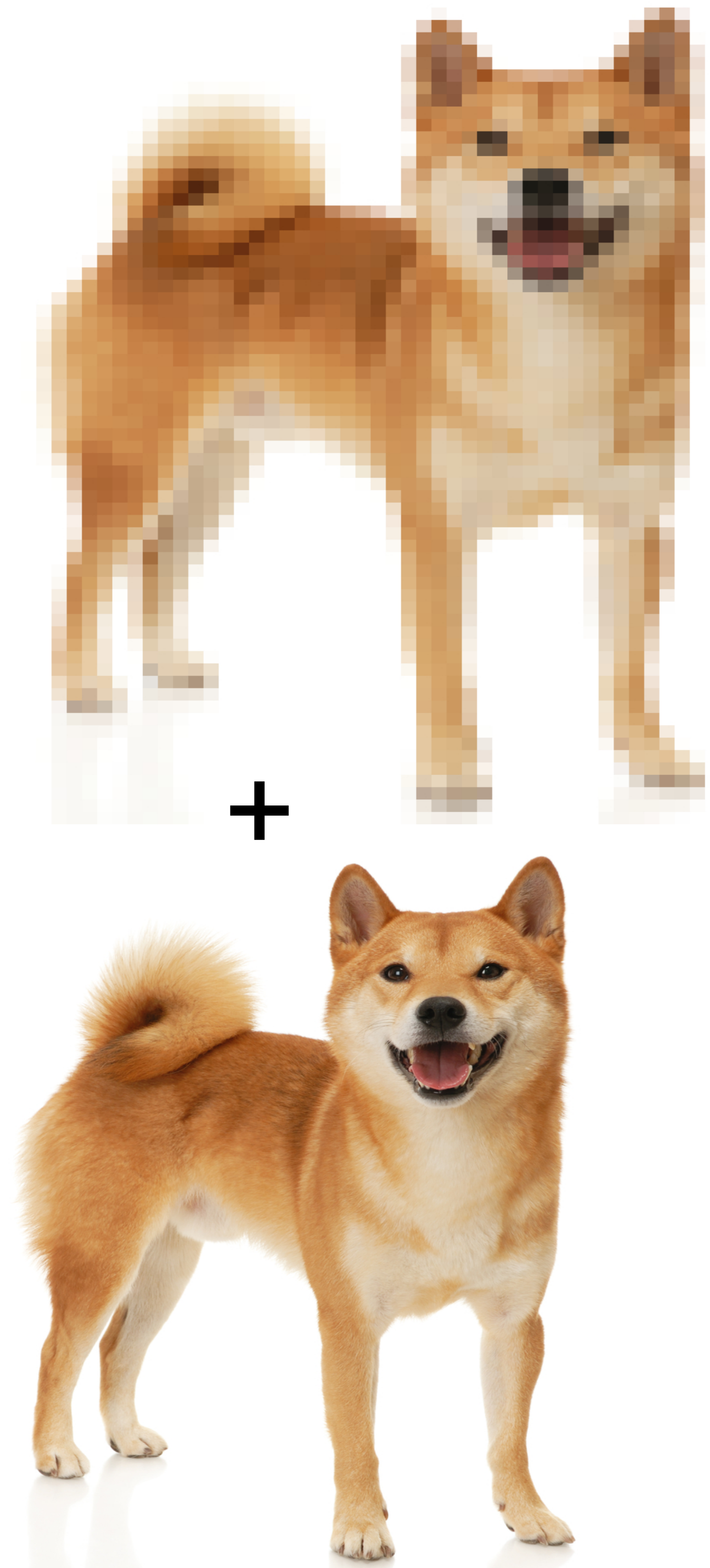
Great for many many evaluations of a big network

Not Great for a small network

…..

…..

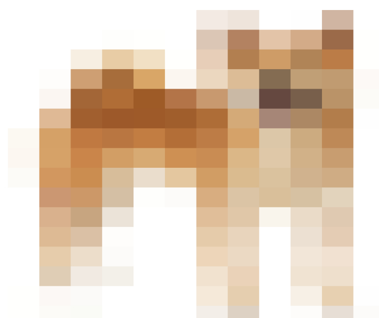


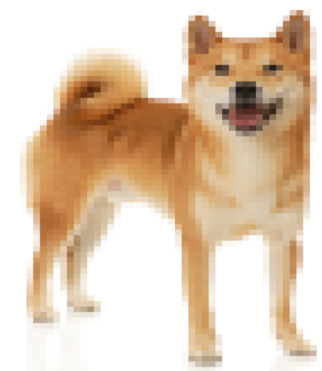


Running @
Longer latencies

HLT Trigger+Offline Reco

40 MHz **1 kHz**



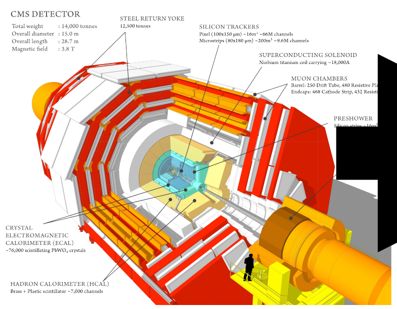


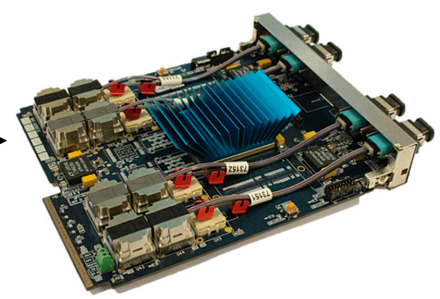
Radiation Hard ASICs

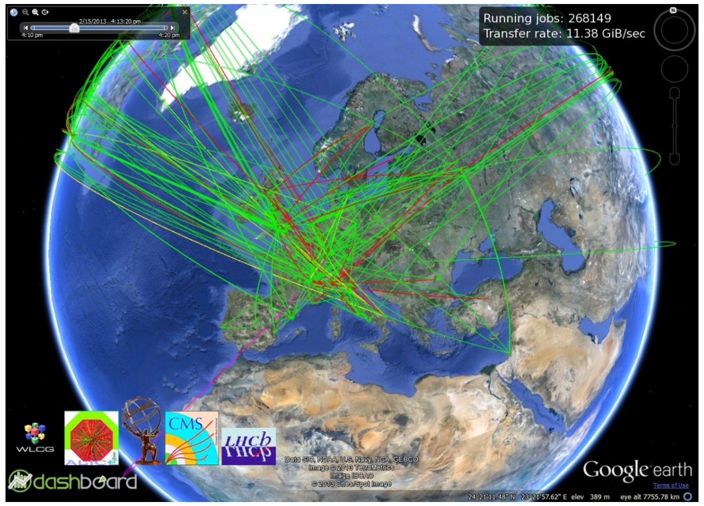
FPGA Boards

Local CPU Cluster

CPU Grid





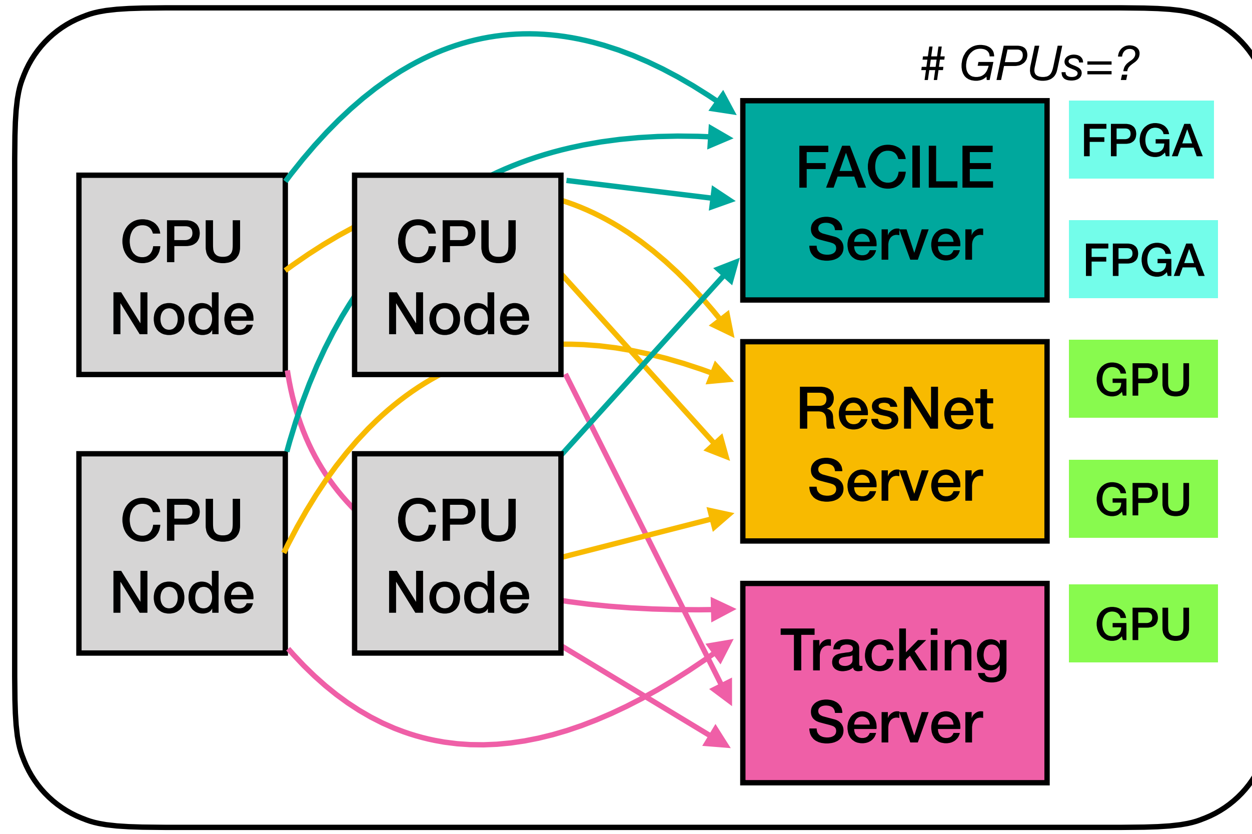


Both Tiers are CPU
similar algos(different scales)

Talking to GPUs

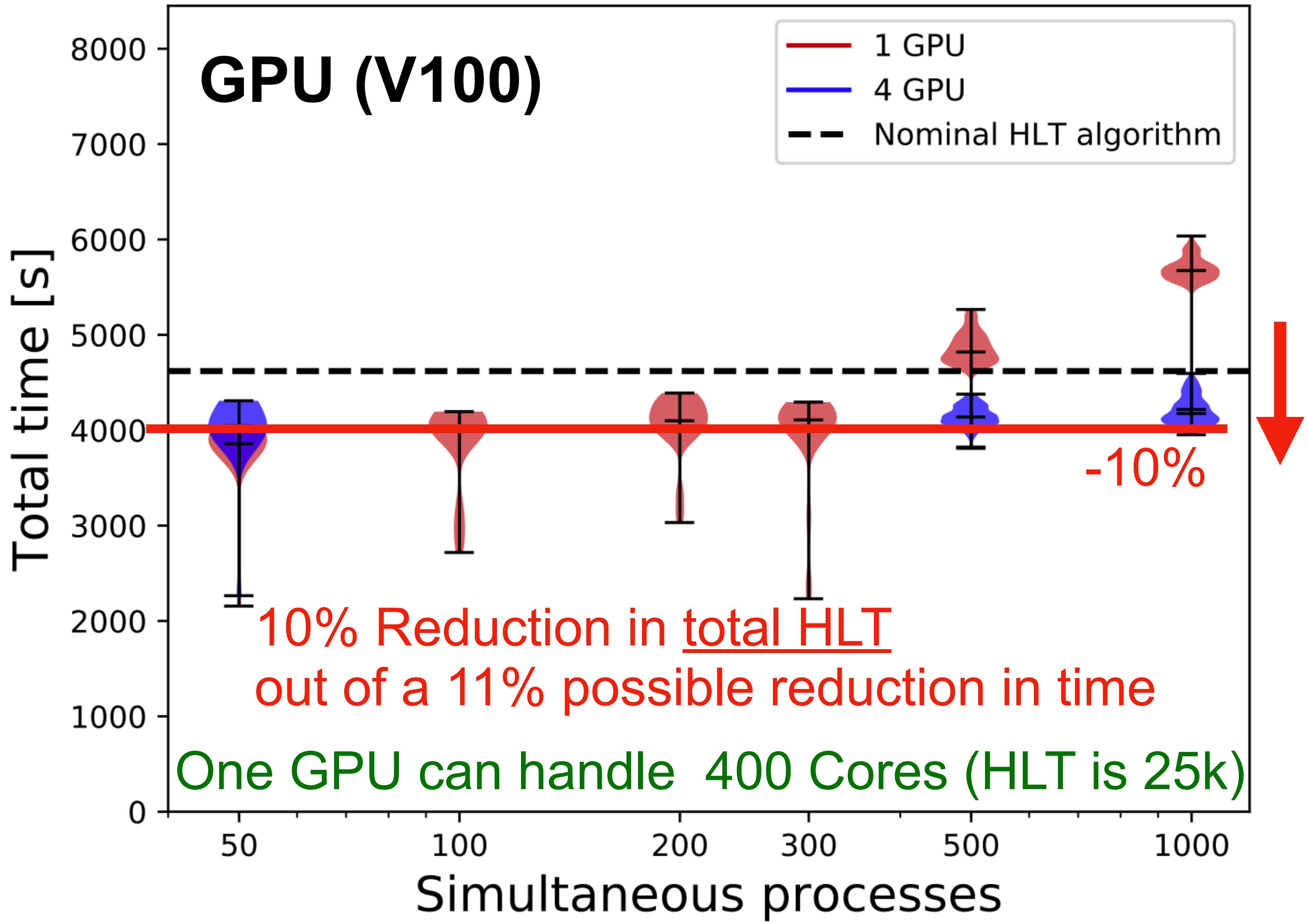
Algo	Per Event
CPU	1.75s
GPU Batch 1	7ms
GPU Batch 32	2ms
FPGA	1.7ms

… as a service



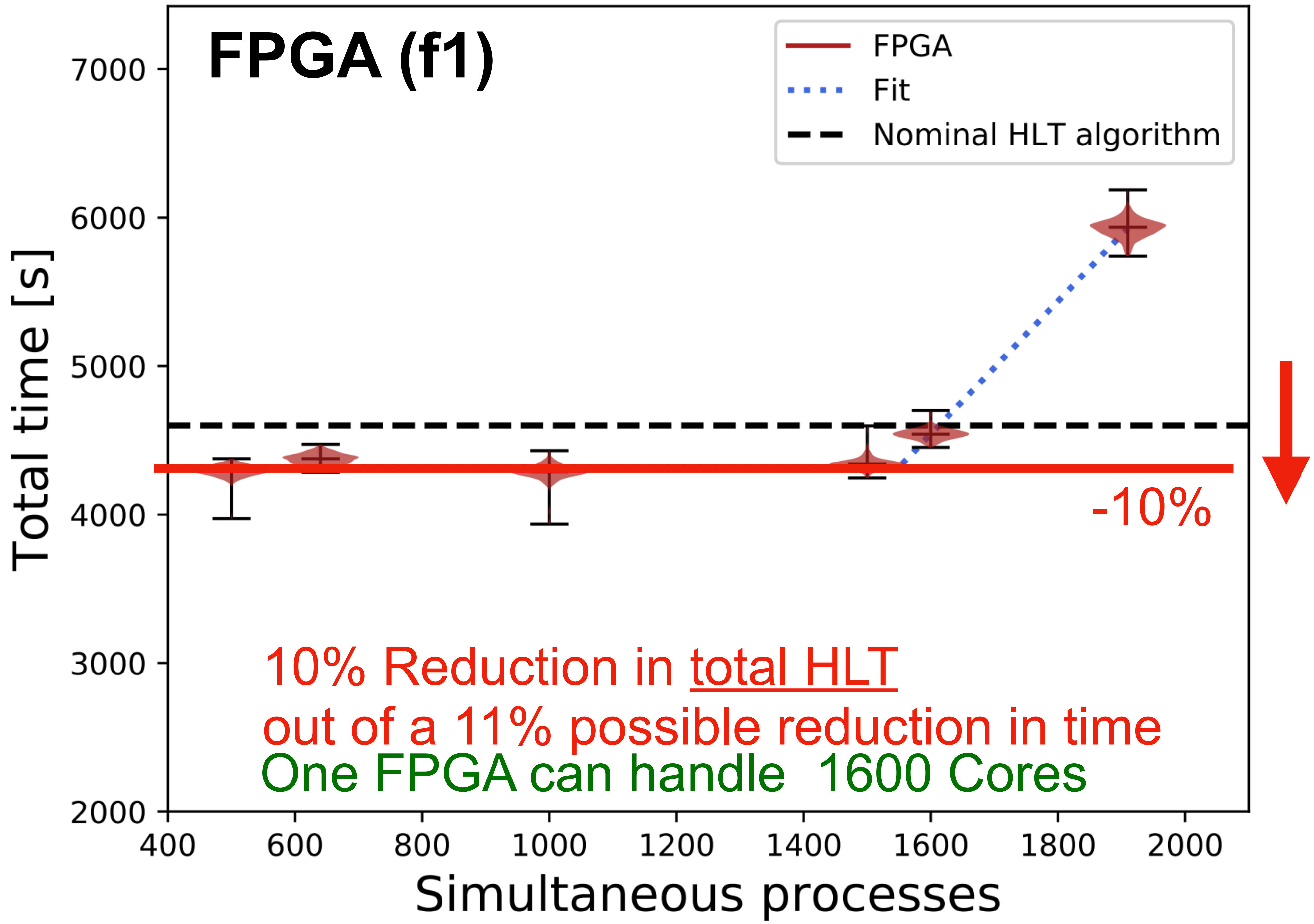
- Deep learning + GPUs or FPGAs can help to speed up systems
  - Deep Learning's regular arch makes GPU/FPGA speedups large
- There are a few ways to integrate these systems
  - My preference is to the right (some connect GPUs directly)

4 GPUs can reduce a 1000 CPUs systems time by 10%



arxiv:2007.10359

1 FPGA can reduce a 1500 CPU systems time by 10%



In fact the limit here is not from the FPGA its network (25 Gbps)

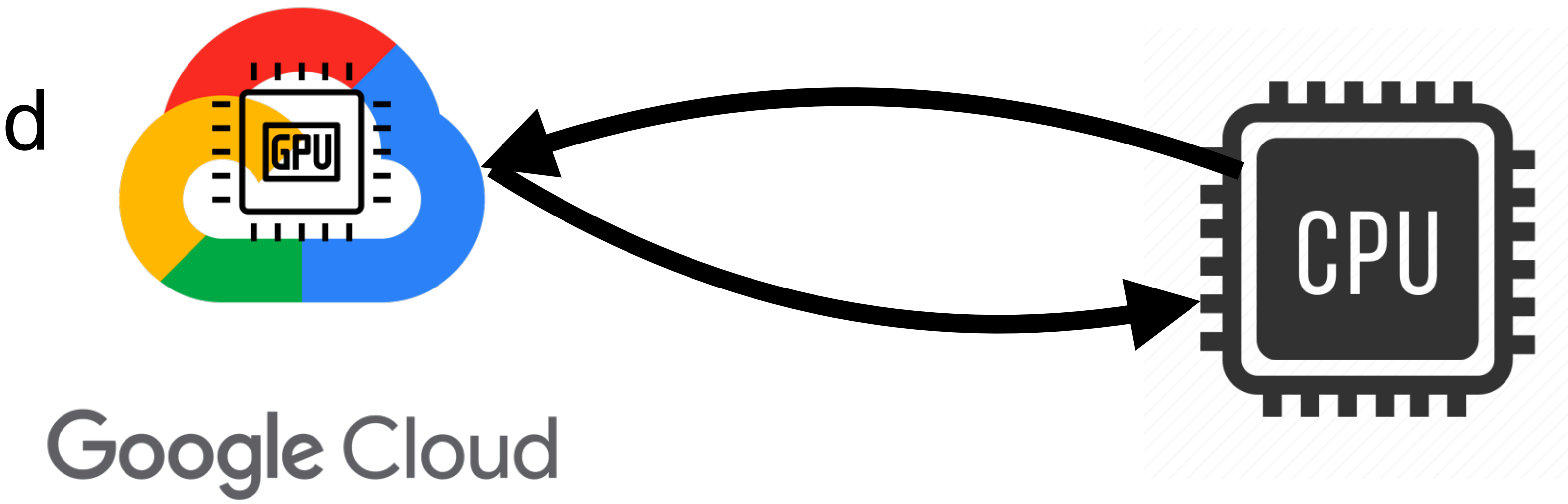
arxiv:2010.08556

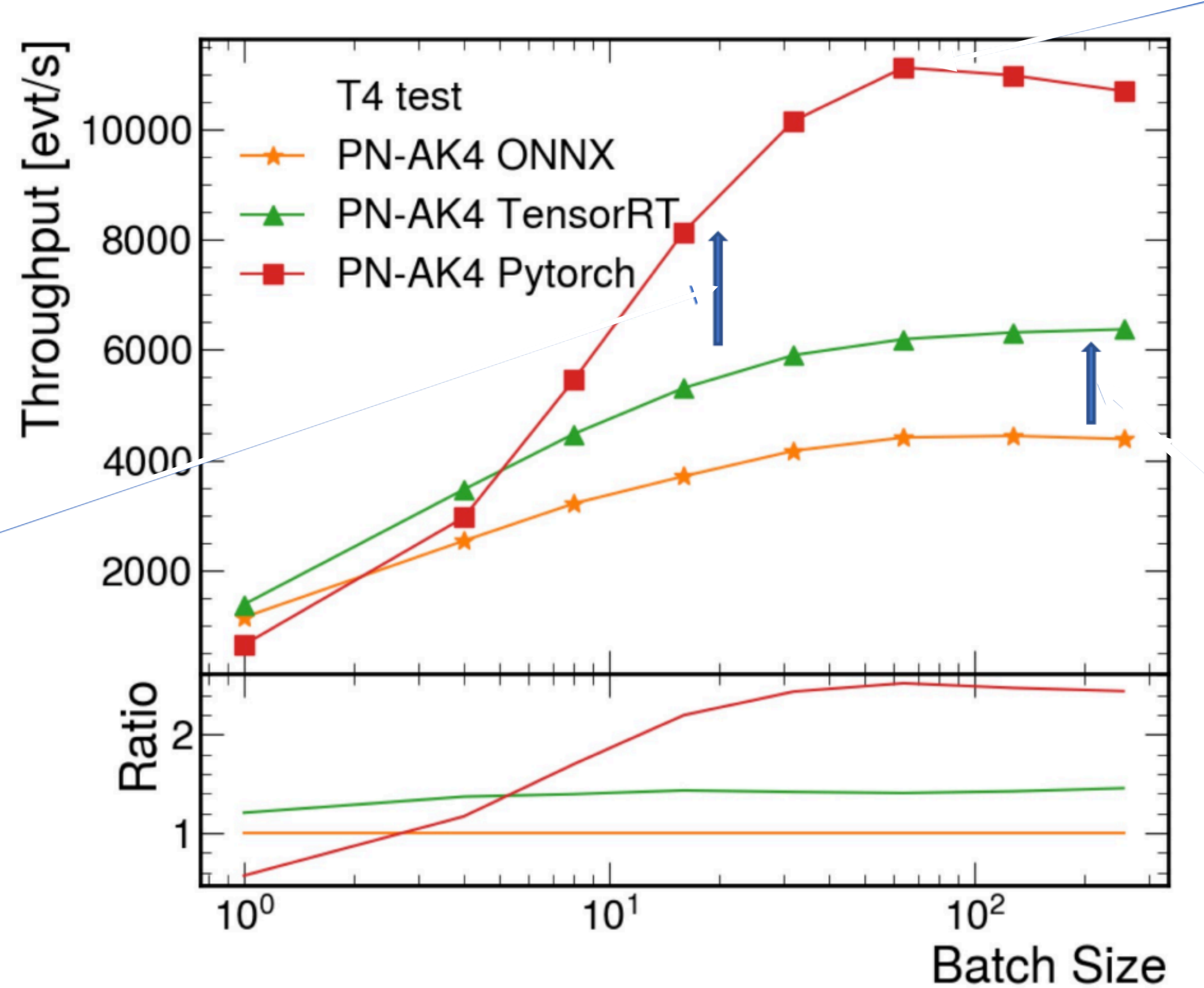
Running To Scale

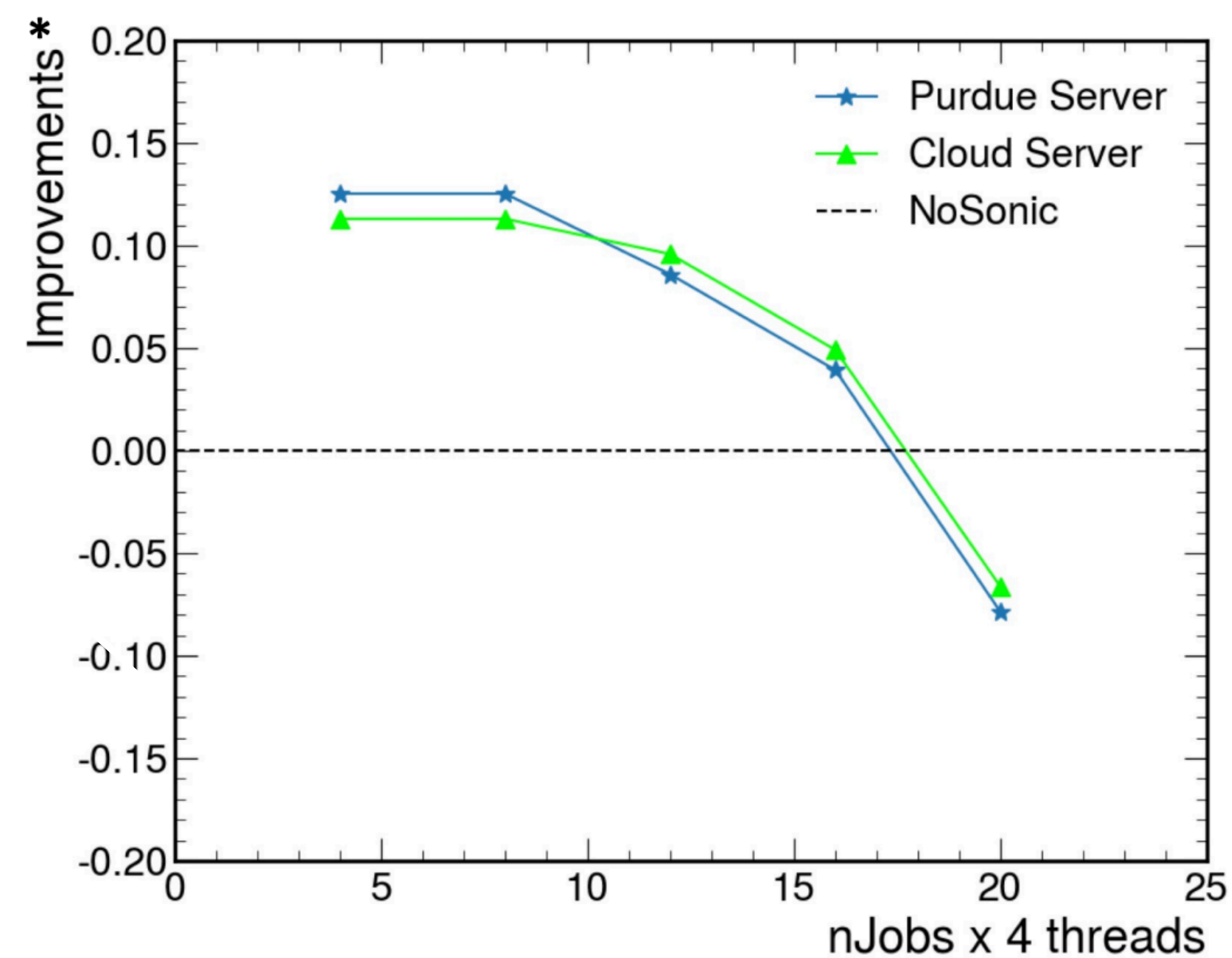
- In addition we have been able to run this work to scale

By Using Google Cloud

Sped up 3 algos currently in use gave 15% reco speedup

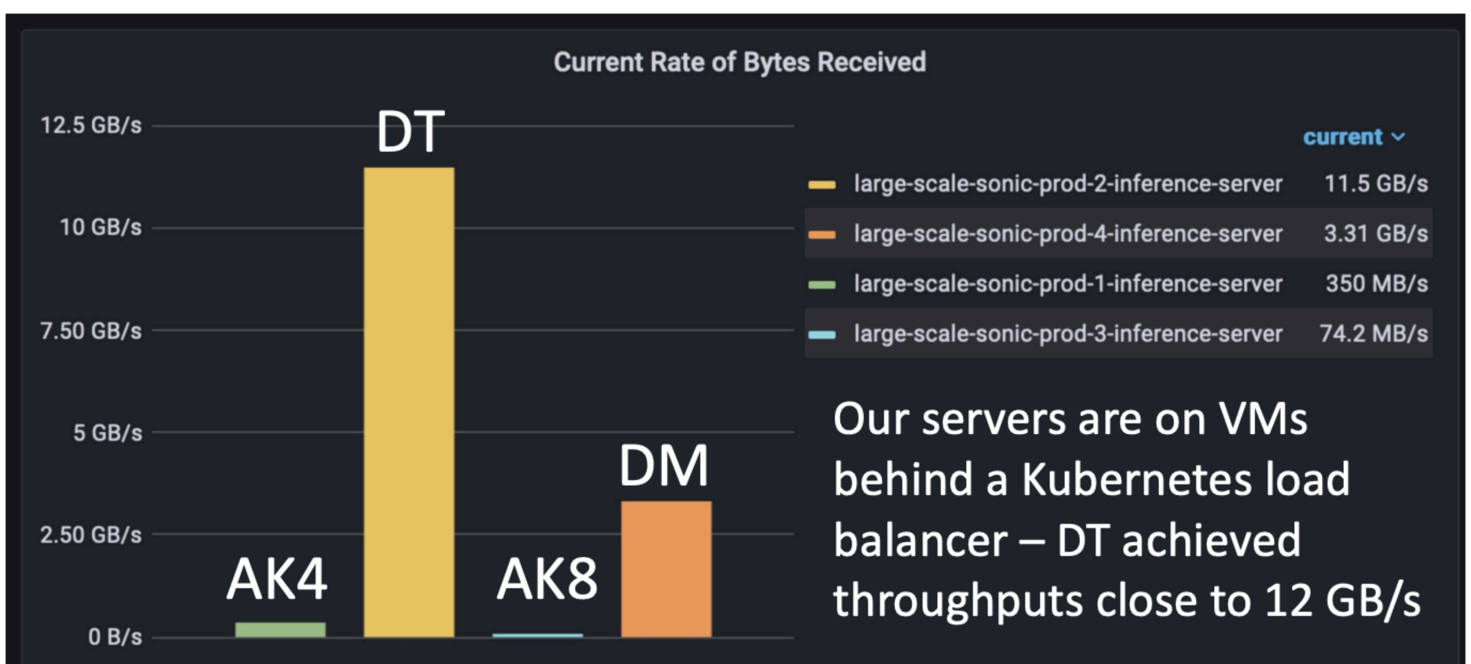


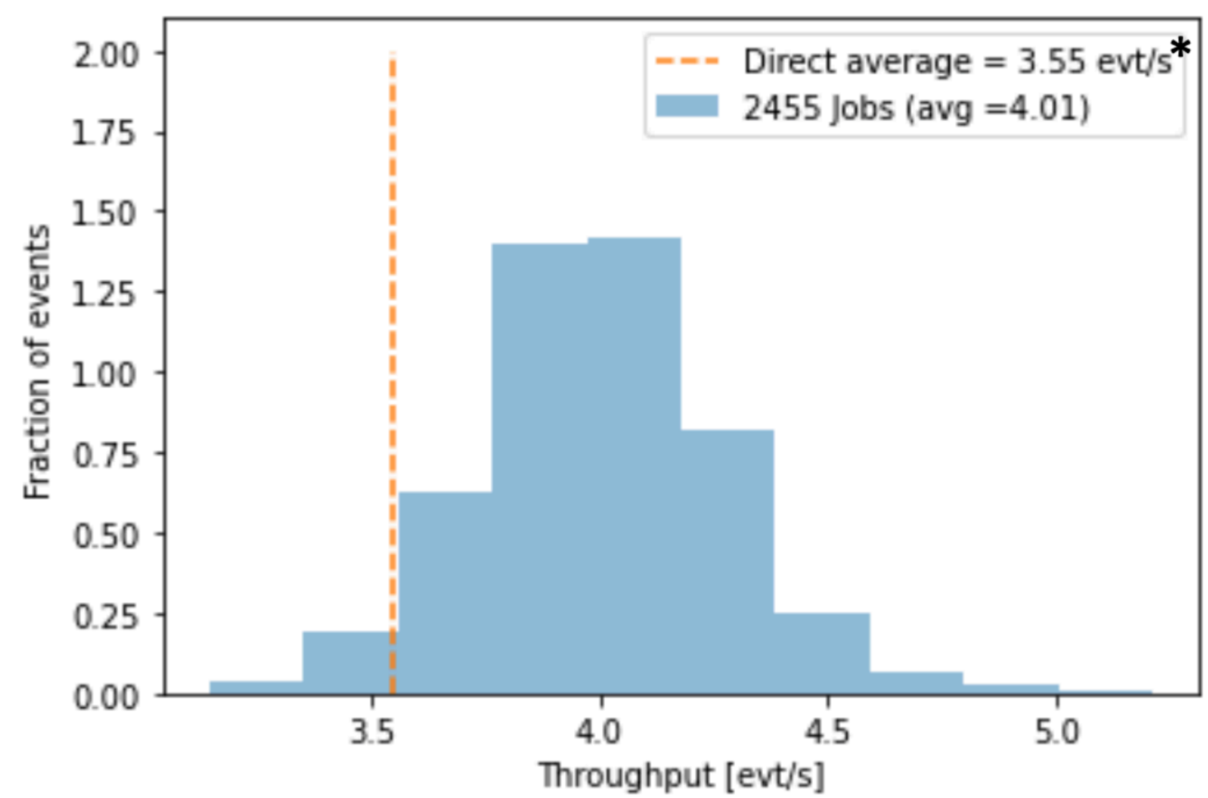




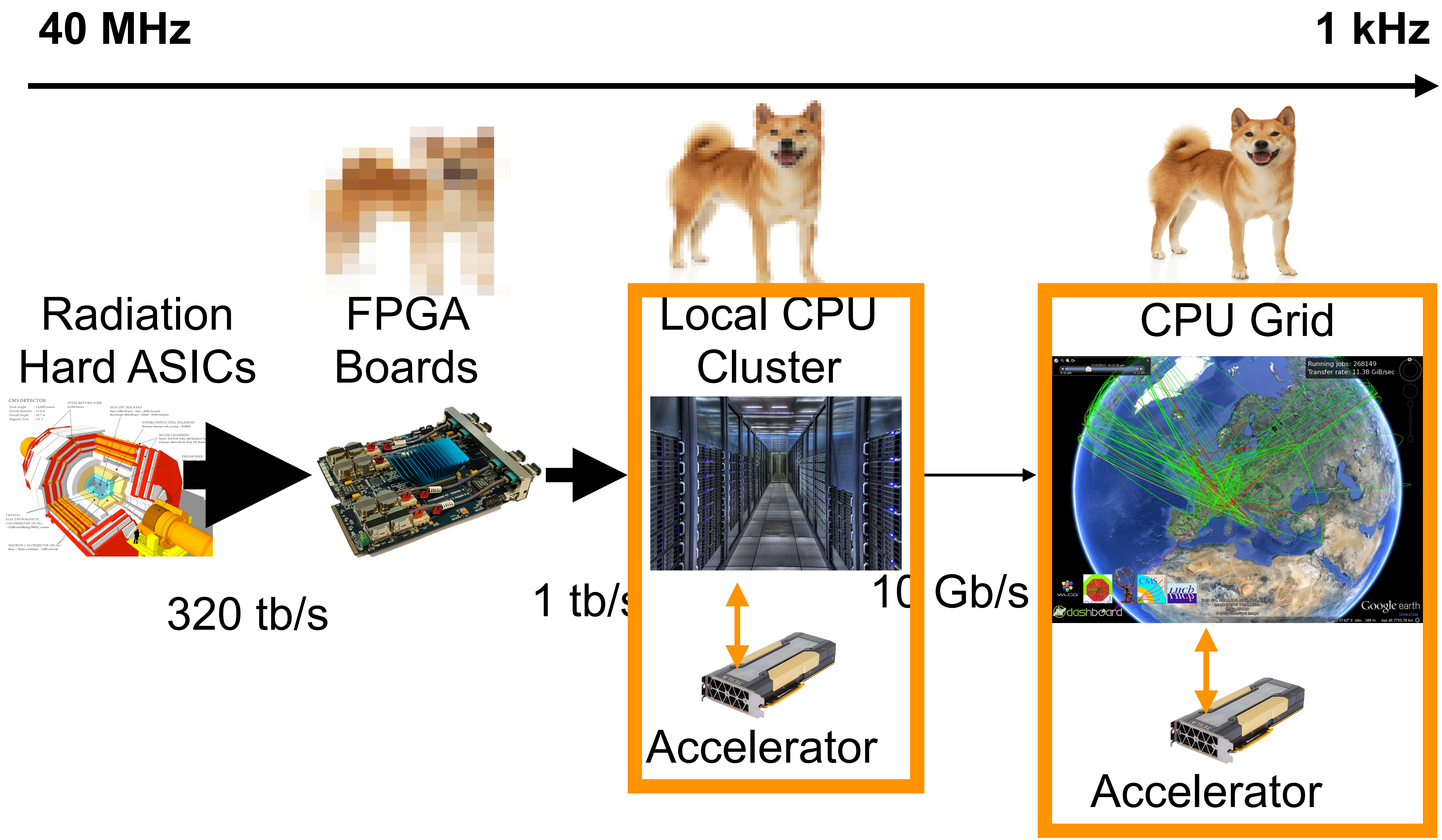
Running To Scale

- In addition we have been able to run this work to scale
  - Ran a test with 10000 CPU cores and 150 GPUs
  - Processes a realistic 150 TB sample
  - Demonstrated this paradigm works to scale!



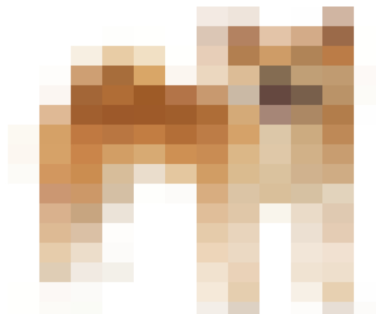


A Broader Vision of DAQ



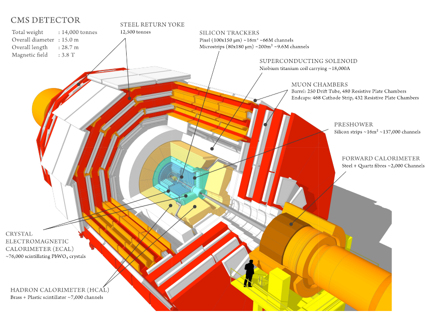
A Broader Vision of DAQ

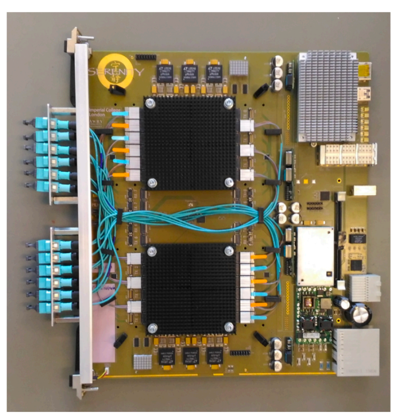
40 MHz **100 kHz**

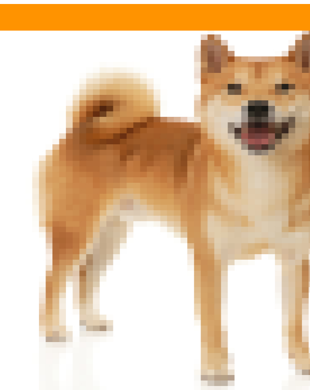


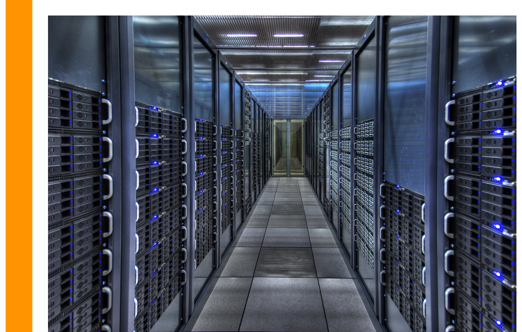
Radiation Hard ASICs

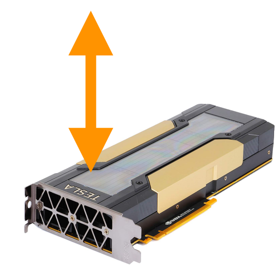
FPGA Boards







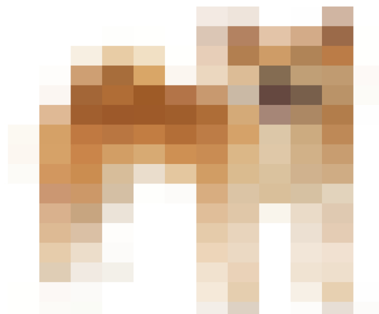




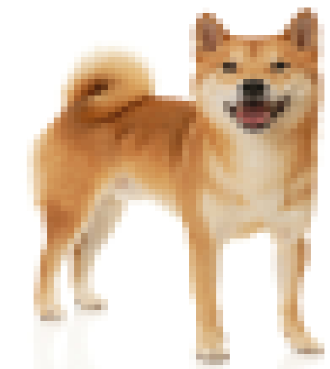
Now Lets Zoom In on our system

A Broader Vision of DAQ

40 MHz **100 kHz**

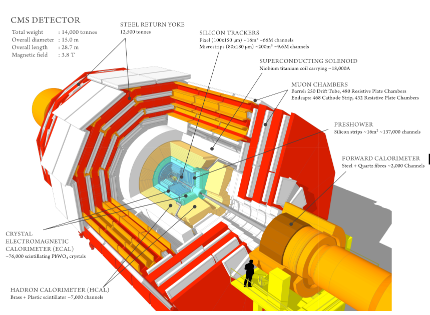


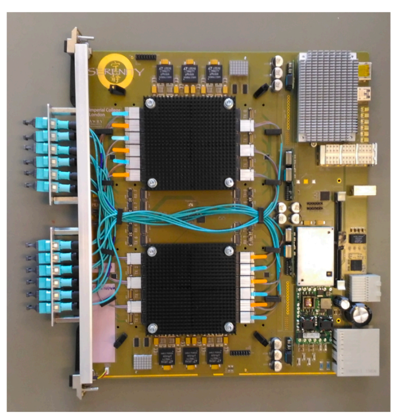
hls4ml

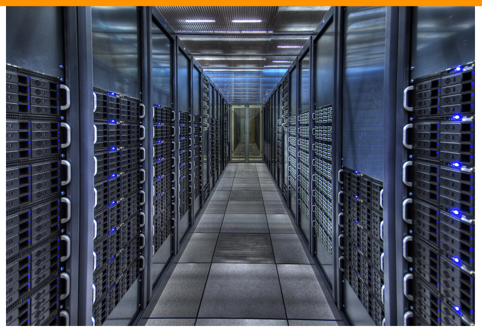


Radiation Hard ASICs

FPGA Boards

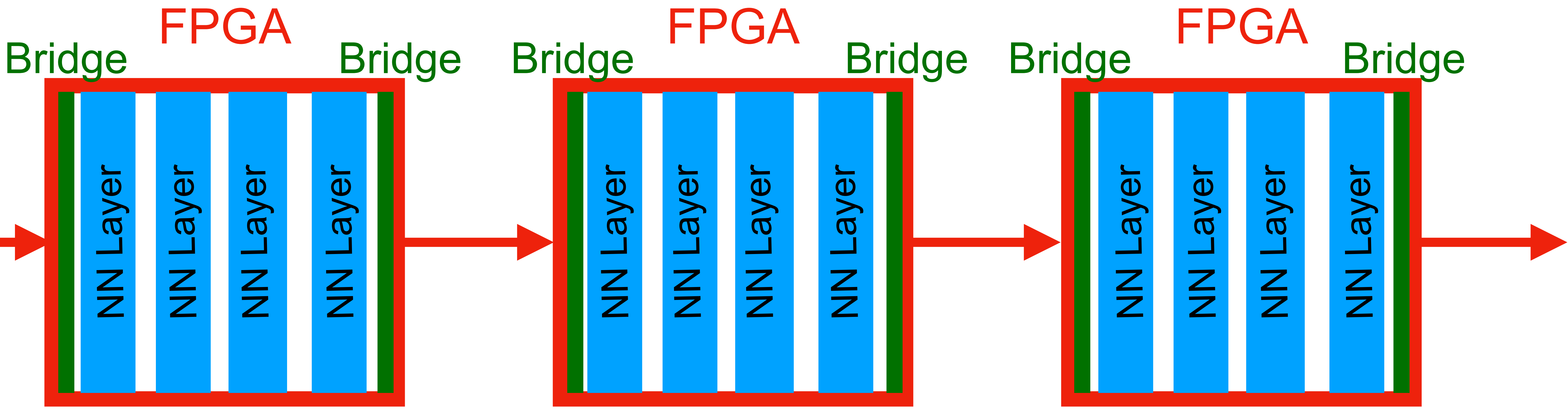






We can actually envision merging these systems

Algean



With Algean we can stretch out networks across many FPGAs

100 Gb/s protocol between FPGAs (can go to CPUs)

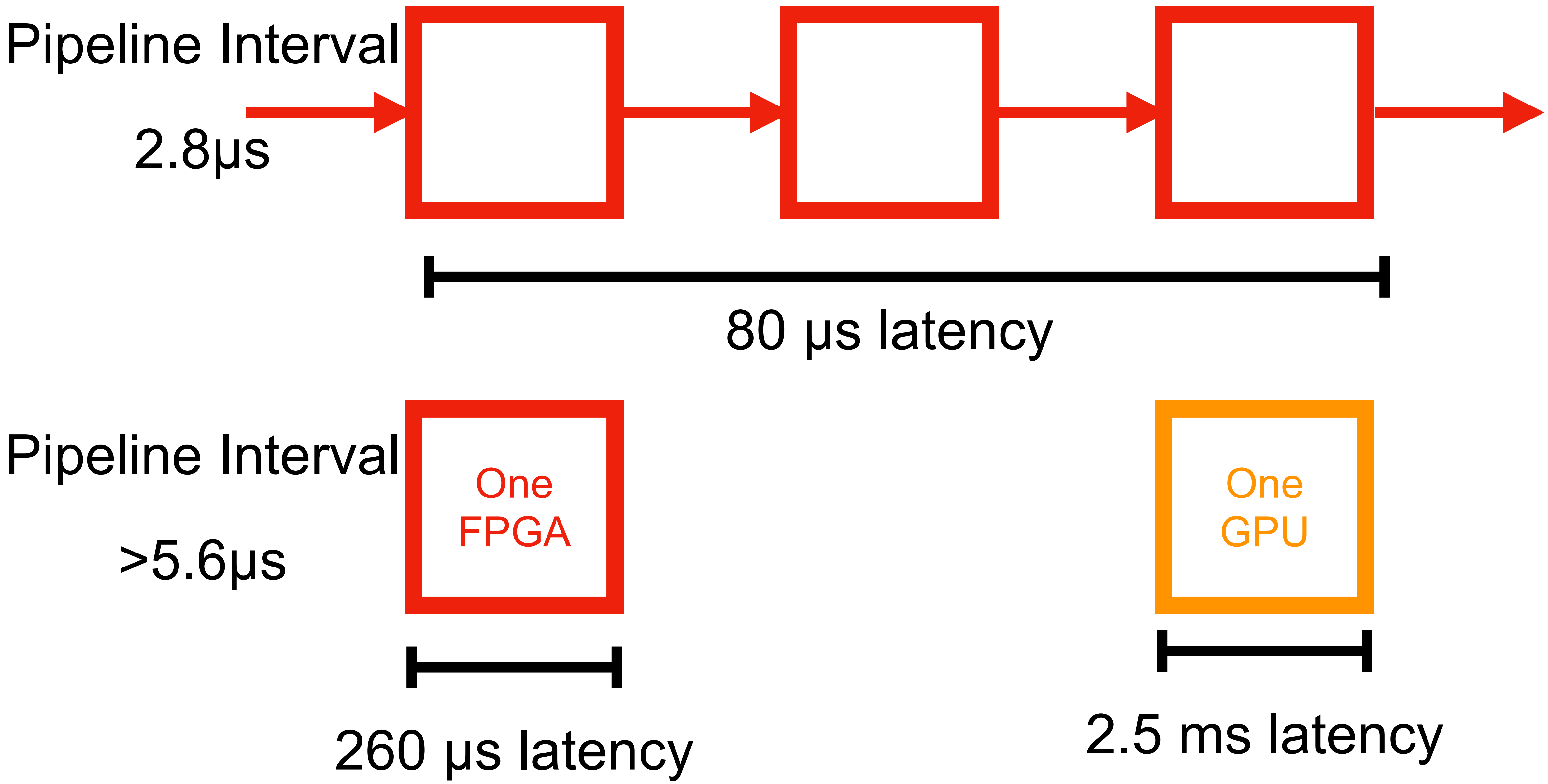
This allows us to run inference for very large networks

Very Fast

Tune our network to the resources we have

Example Autoencoder

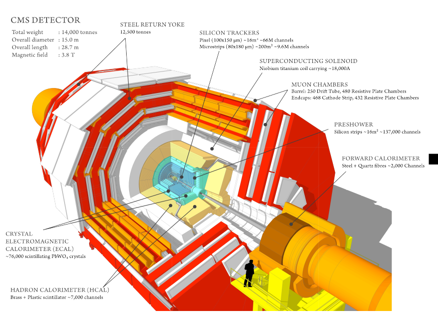
Anomaly detection algorithm



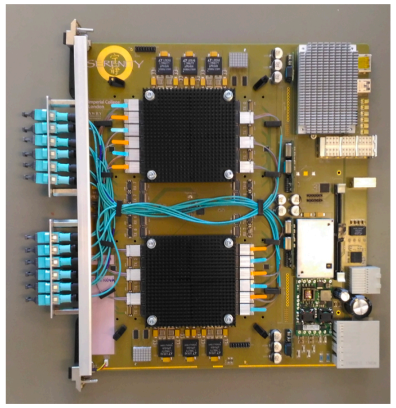
A Broader Vision of DAQ

40 MHz **100 kHz**

Radiation Hard ASICs

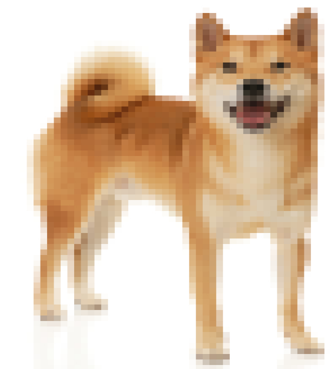


FPGA Boards



Local Algorithms

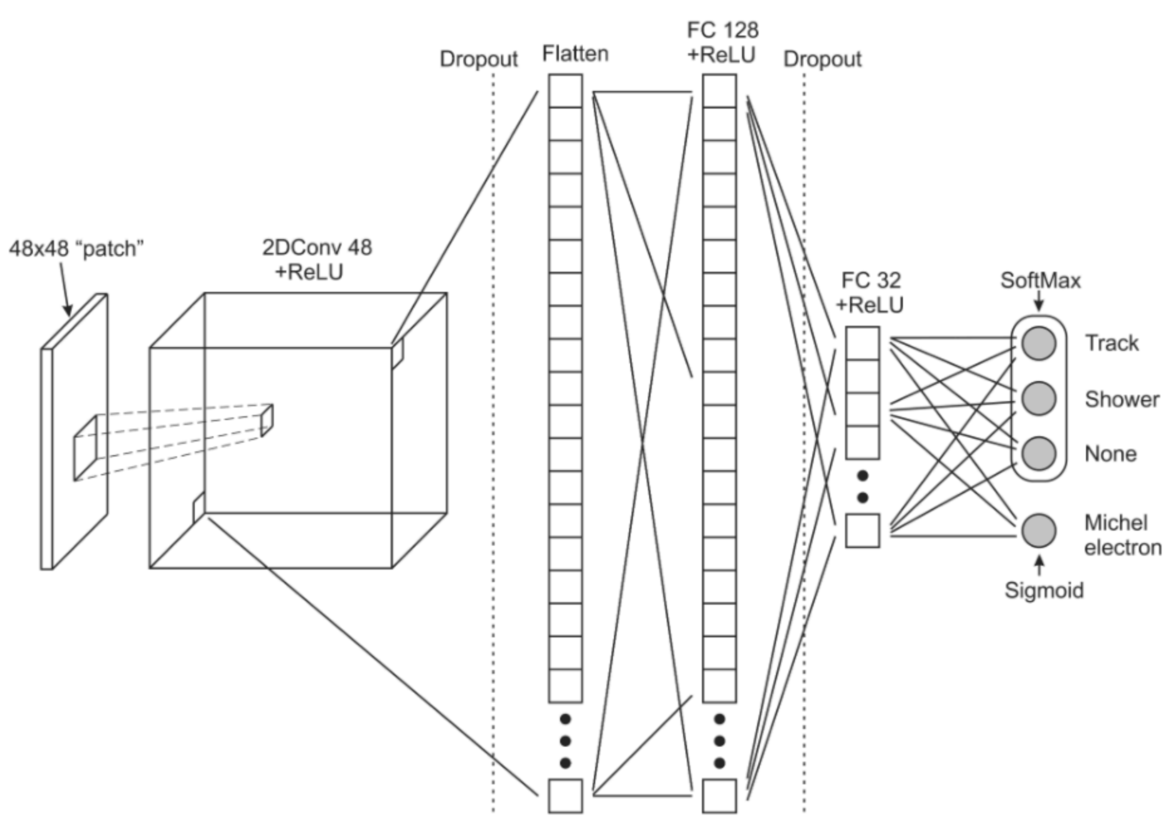
Global Algorithms



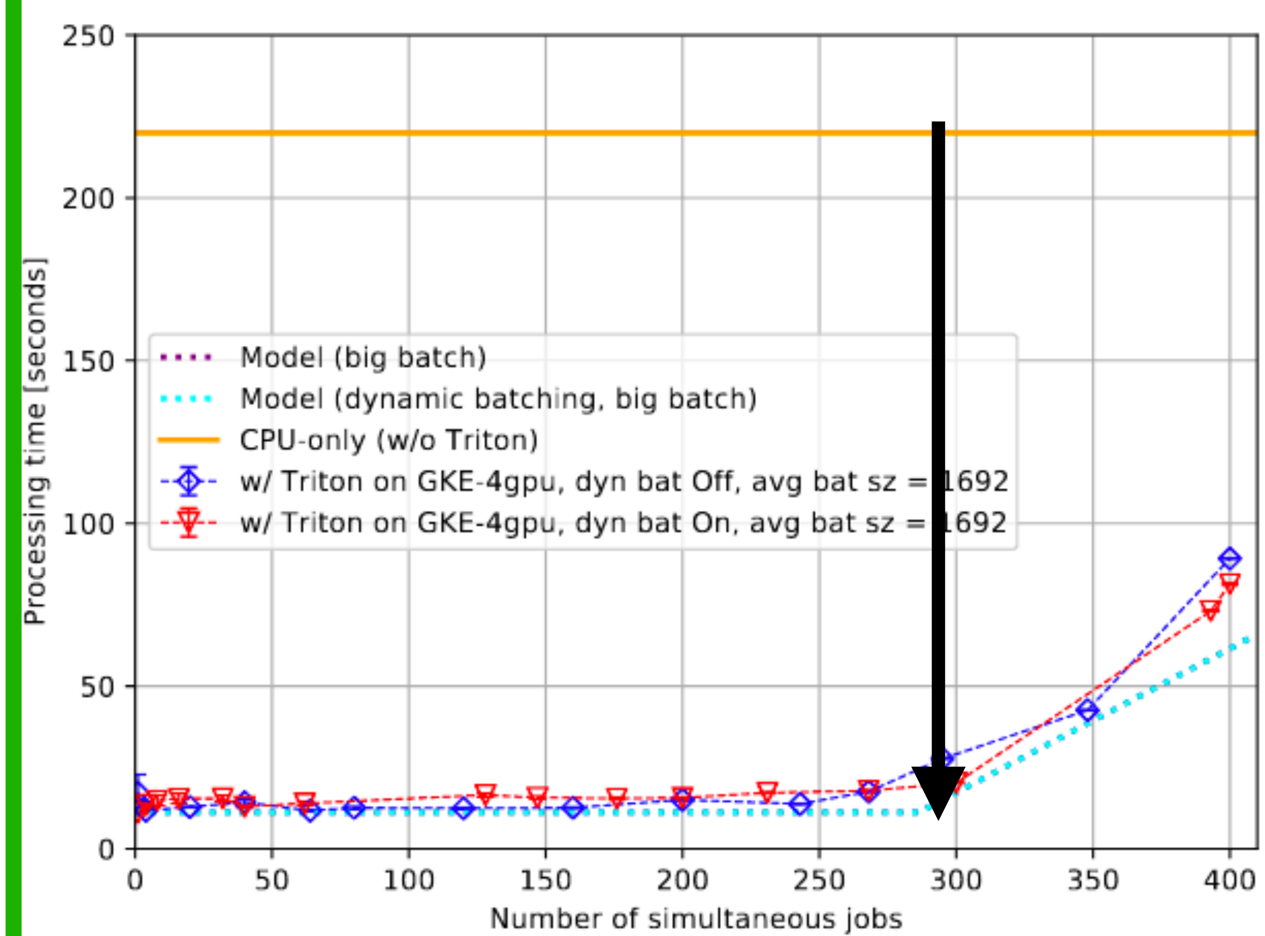
There are new ideas for 40 MHz (partial) processing of all data

Neutrino Physics

- We are pursuing the same idea in Neutrino physics



Michel Electron Id NN

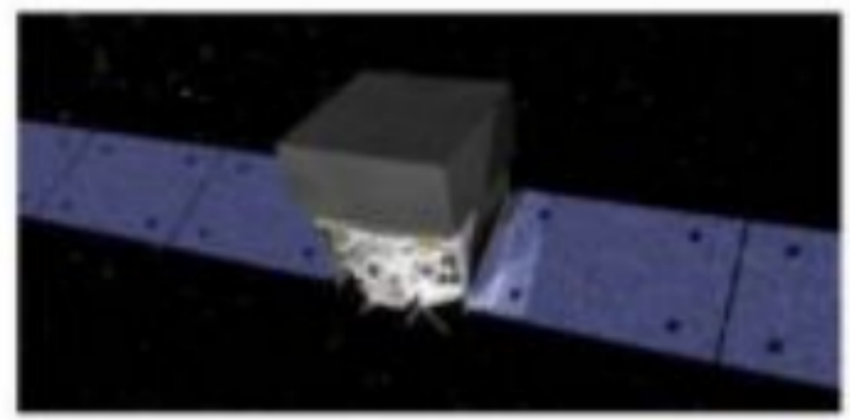


Large Factor in speed up

Gravitational Waves

- Aiming to identify Gravitatoinal waves fast to do MMA
  - Correlating GW and Optical observations is powerful

See a Gravitational Wave

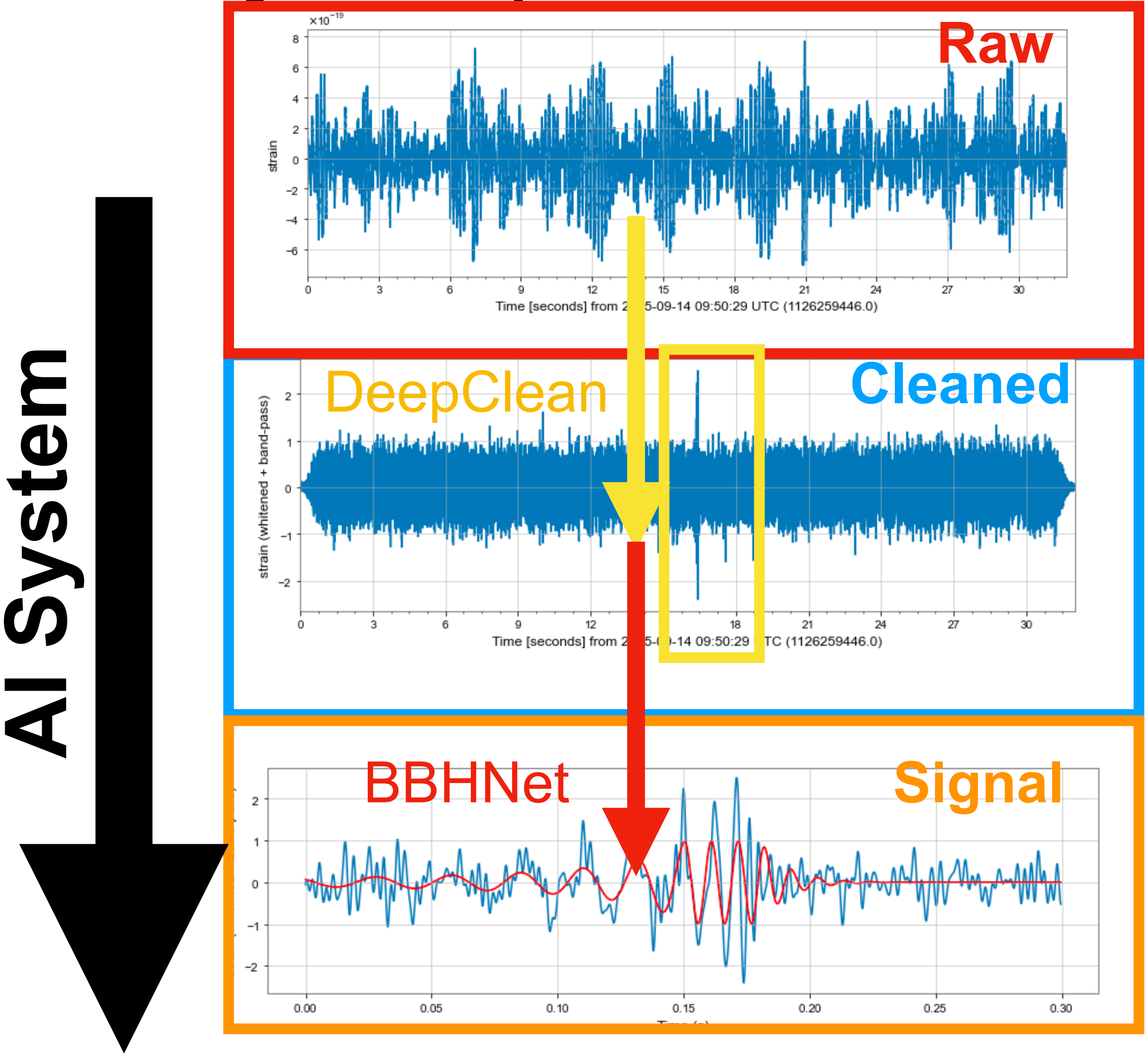


Alert a Telescope

Can we make the GW reconstruction fast enough to be real-time?

Gravitational Waves

- Aiming to identify Gravitatoinal waves fast to do MMA



Current Non-AI Chain
Takes a long time
This Whole chain in < 1s

Arsenal of telescopes

Once you have found the GW event
have to send the coordinates to a huge network

A3D3

- An institute to unite real-time AI
  - Quickly looking for people to be part of extended team

Accelerated AI
Algorithms for
Data-Driven
Discovery

Fast ML and now A3D3

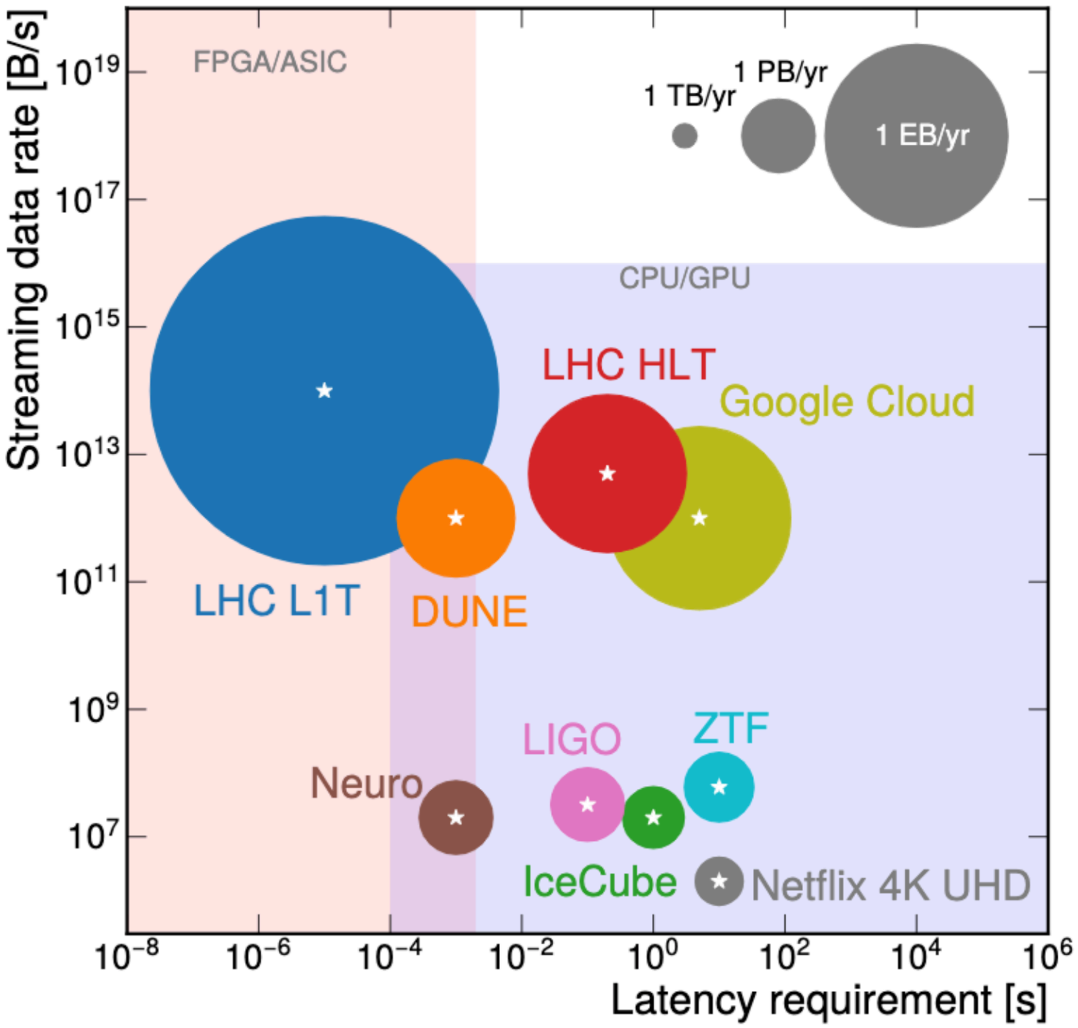


Photo from our first Fast ML workshop!

fastmachinelearning.org

- We make AI run fast :
  - Our goal is to use AI to speed up processing of experiments
  - Additionally we are developing new ways to speed up AI

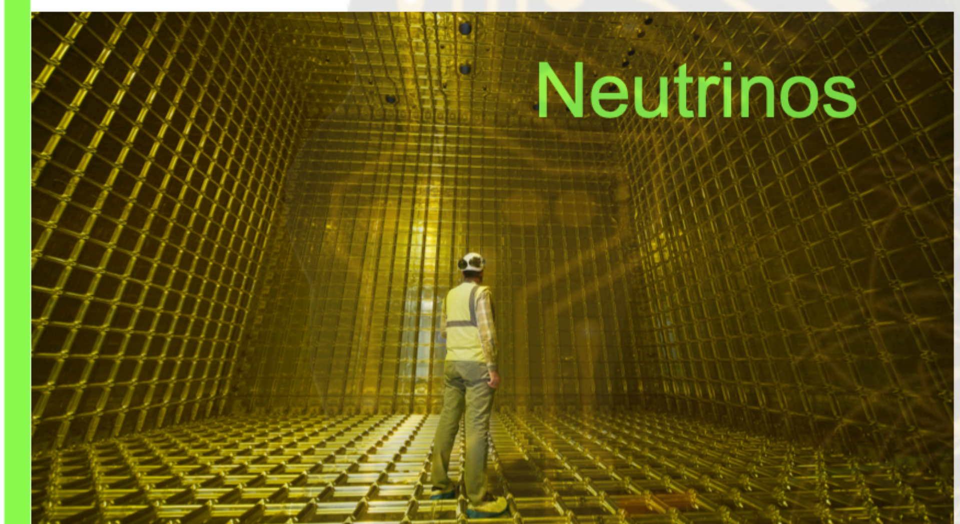
IAIFI Colloquium FPGA Keynote Talk

hls4ml → Deep Learning Compiler for FPGAs/ASICs

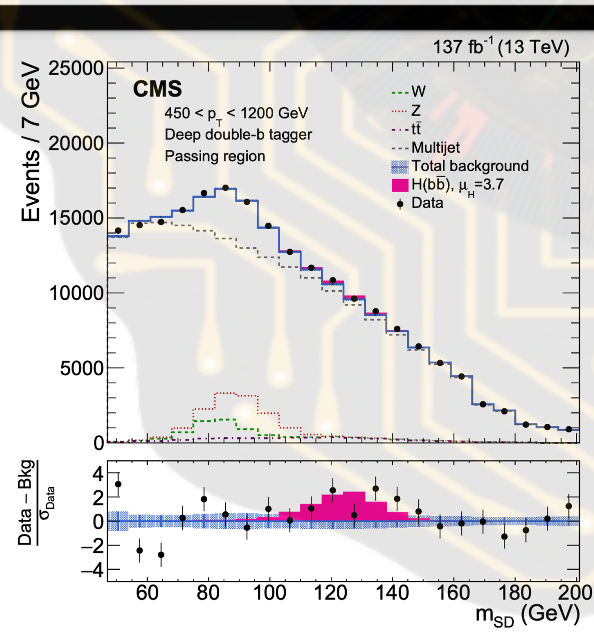
A New Institute: **A3D3**

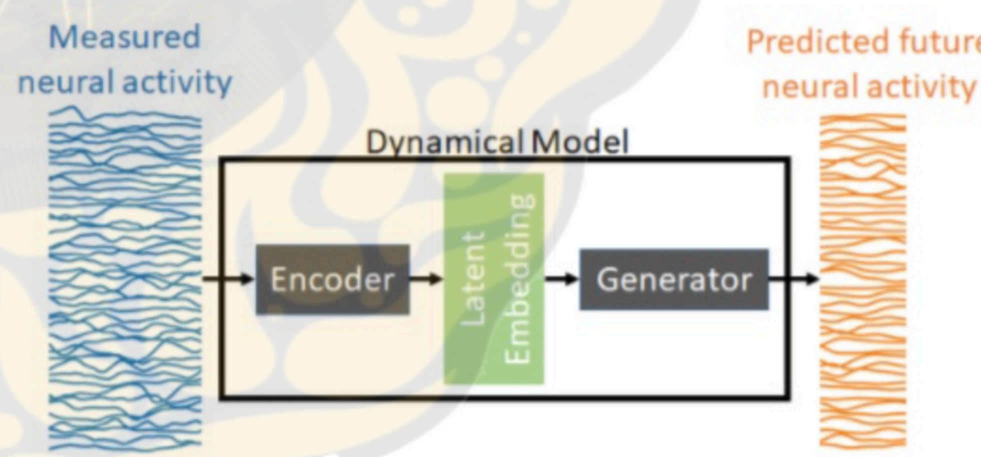
- We have been awarded a new institute to explore real-time AI
  - Accelerated AI Algorithms for Data Driven Discovery (A3D3)

New Types of Computing



LHC
Physics





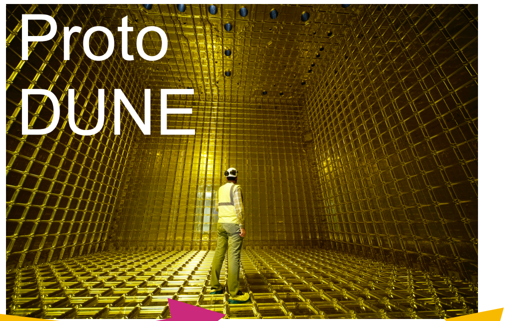
NeuroScience

Overview Venn Diagram

ATLAS EXPERIMENT

CMS

sPHENIX



LIGO VIRGO KAGRA

Fast Machine Learning Lab

Real-Time
Heavy Flavor
Tagging @ sPHENIX

a3d3 Accelerated AI Algorithms for Data-Driven Discovery

Real-time Multi-messenger
Alert

Exploring Clouds
to Accelerate Science

INTERNET2

AI based compression
For Silicon calorimeter
Readout (DOE ASCR)

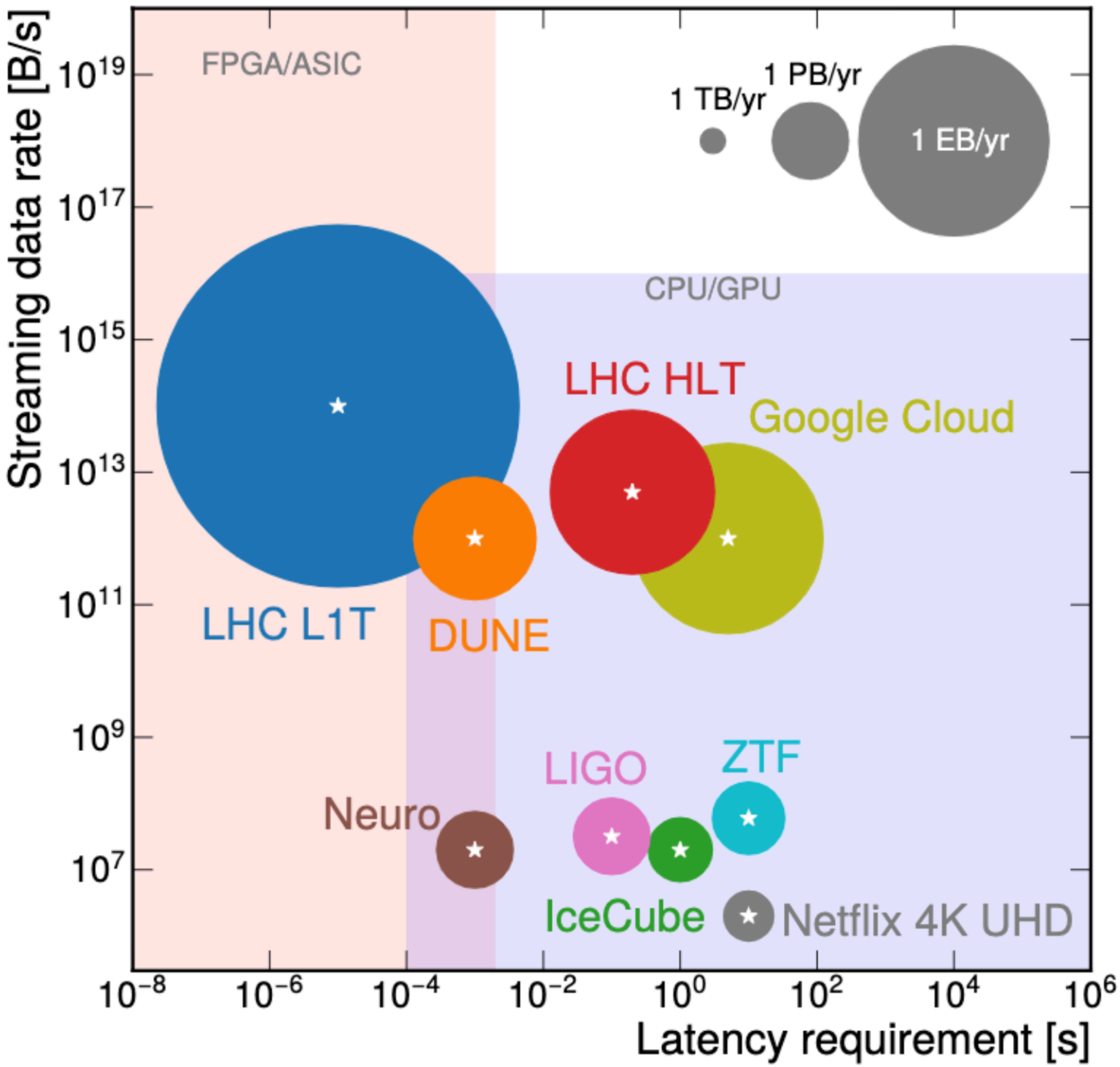
hls4ml

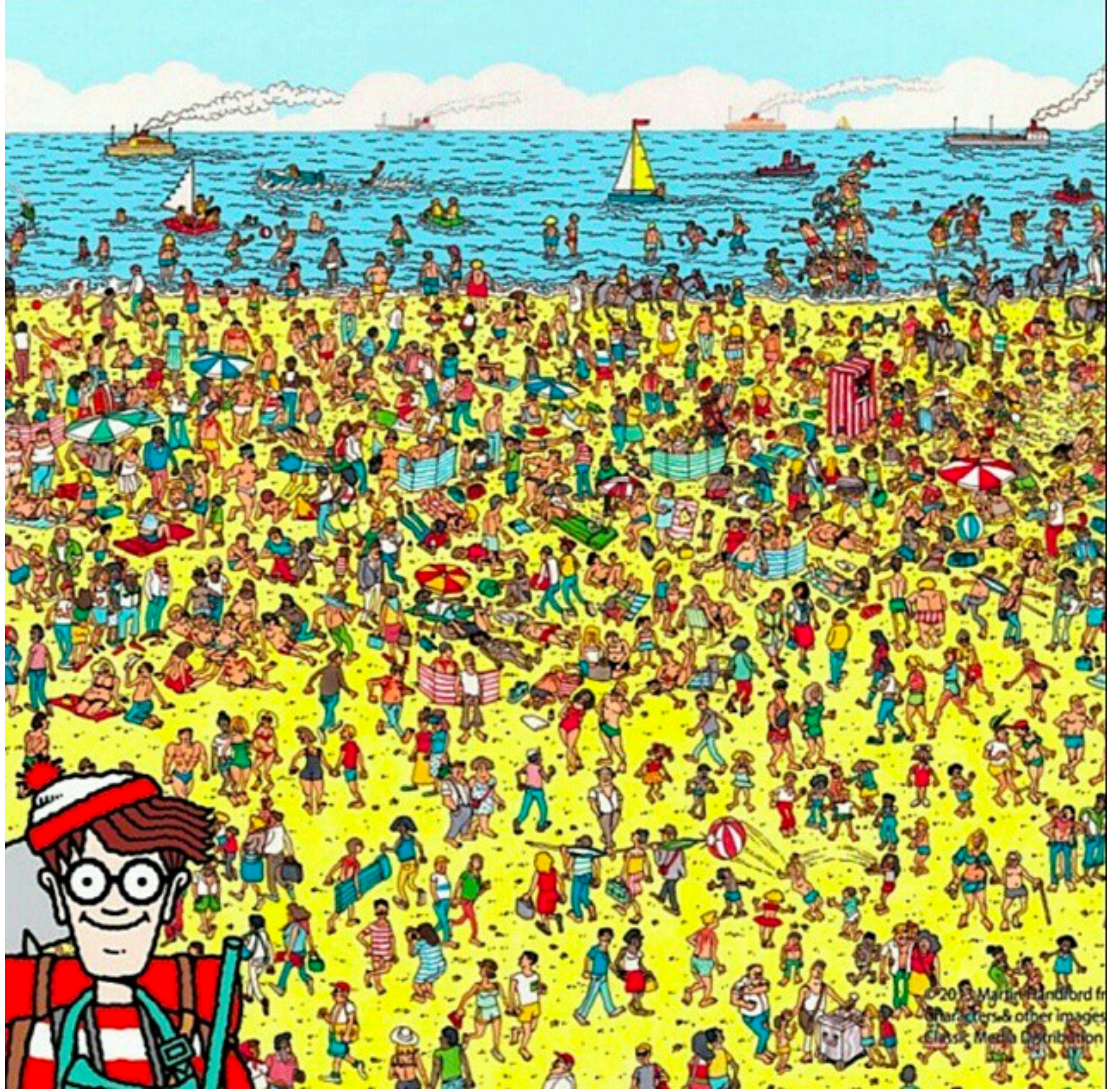
FAIR4HEP

AI Algorithms (AI²)

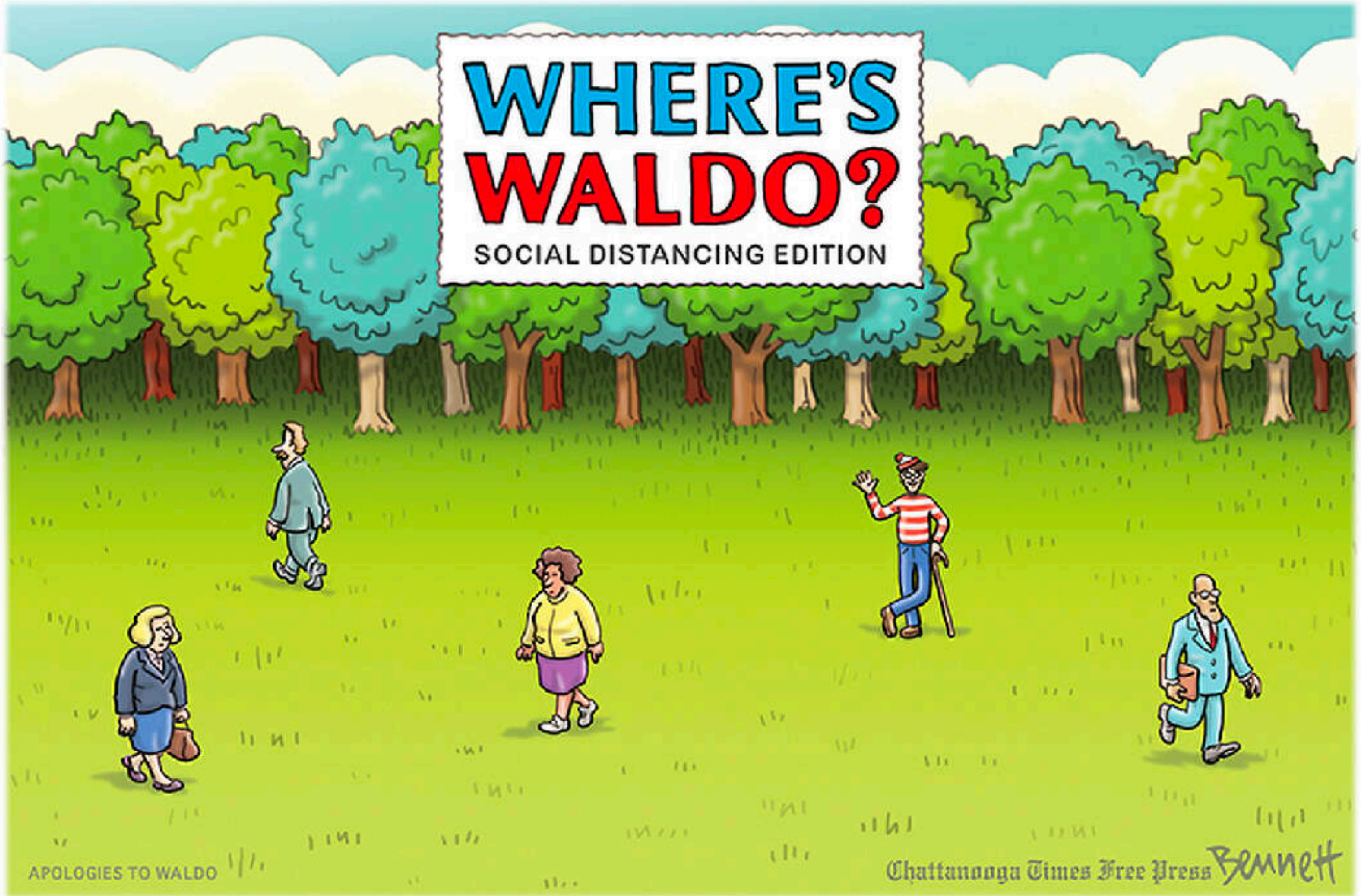
IAIFI

Preparing for the future





Anomaly Detection

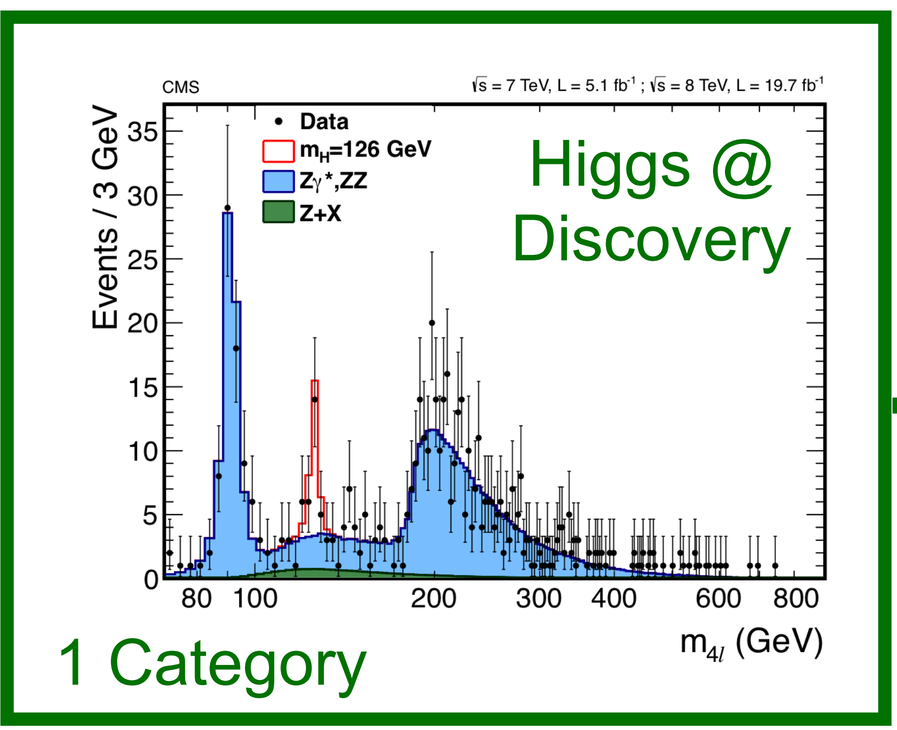


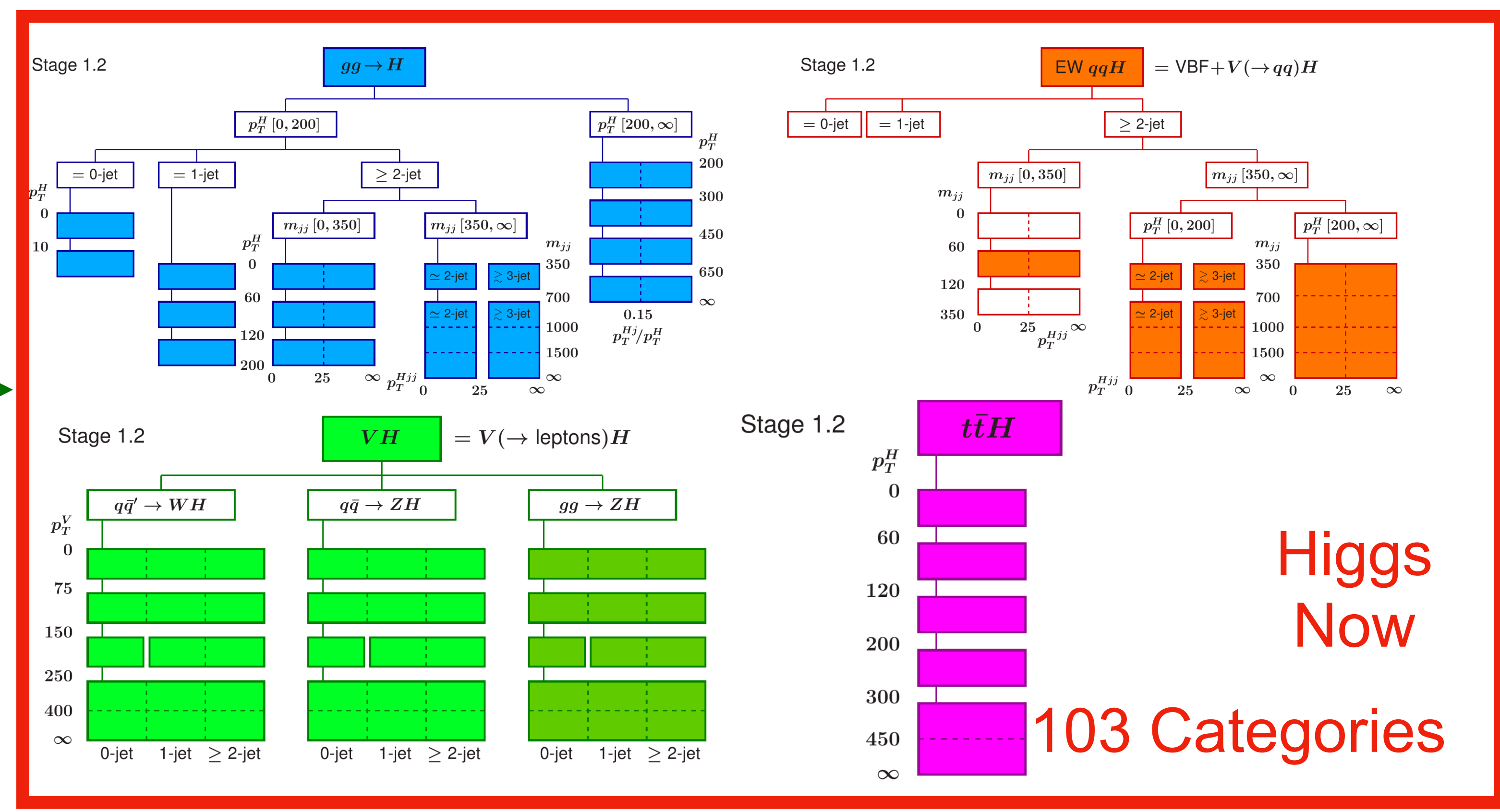
Anomaly Detection

Another Fun thing to do during COVID

Ageing Analyses @LHC

- Data analyses at the LHC are changing
  - Analyses are becoming much more complex
    - Many categories and many final states
- General trend towards more complicated analyses



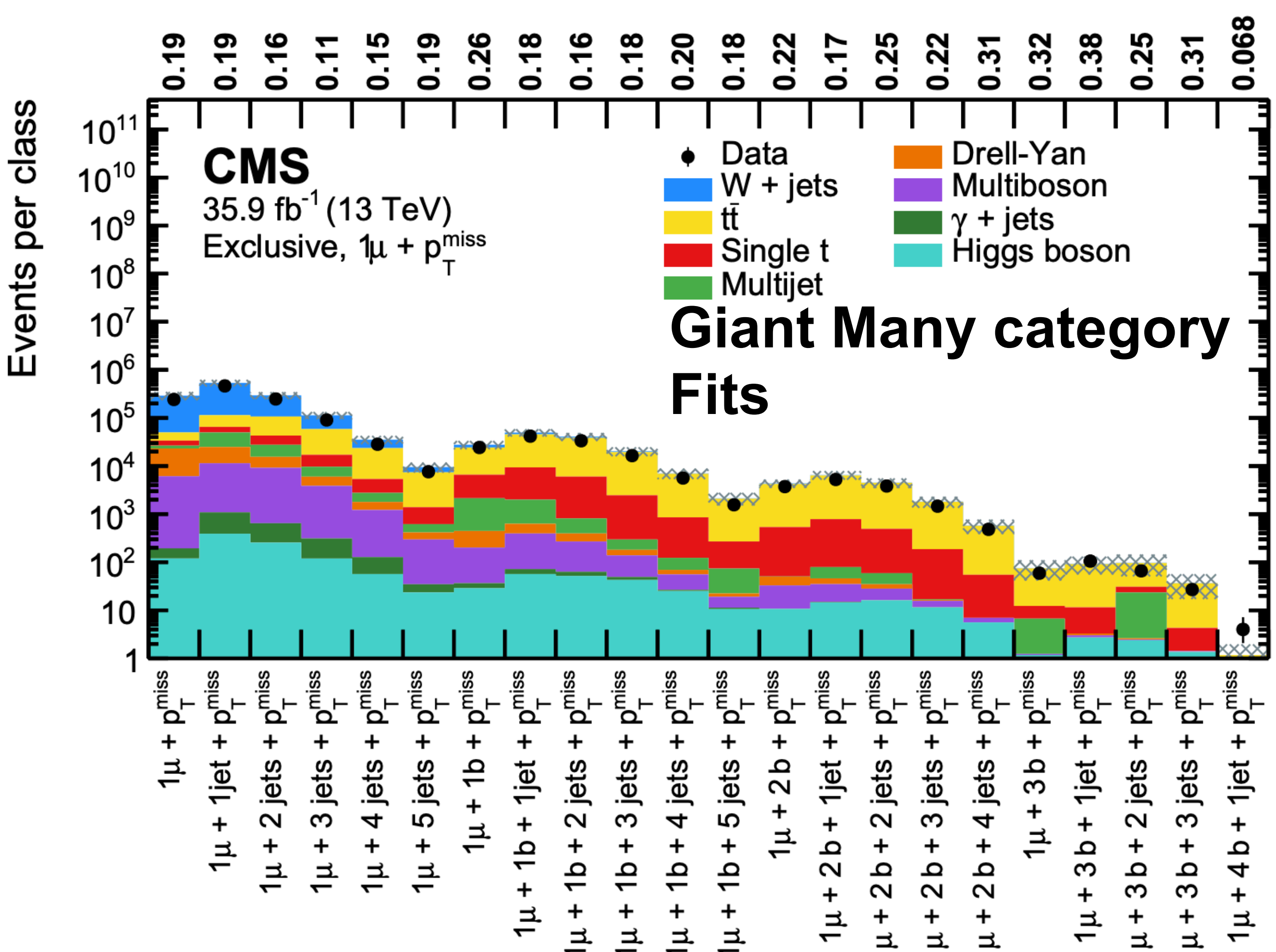


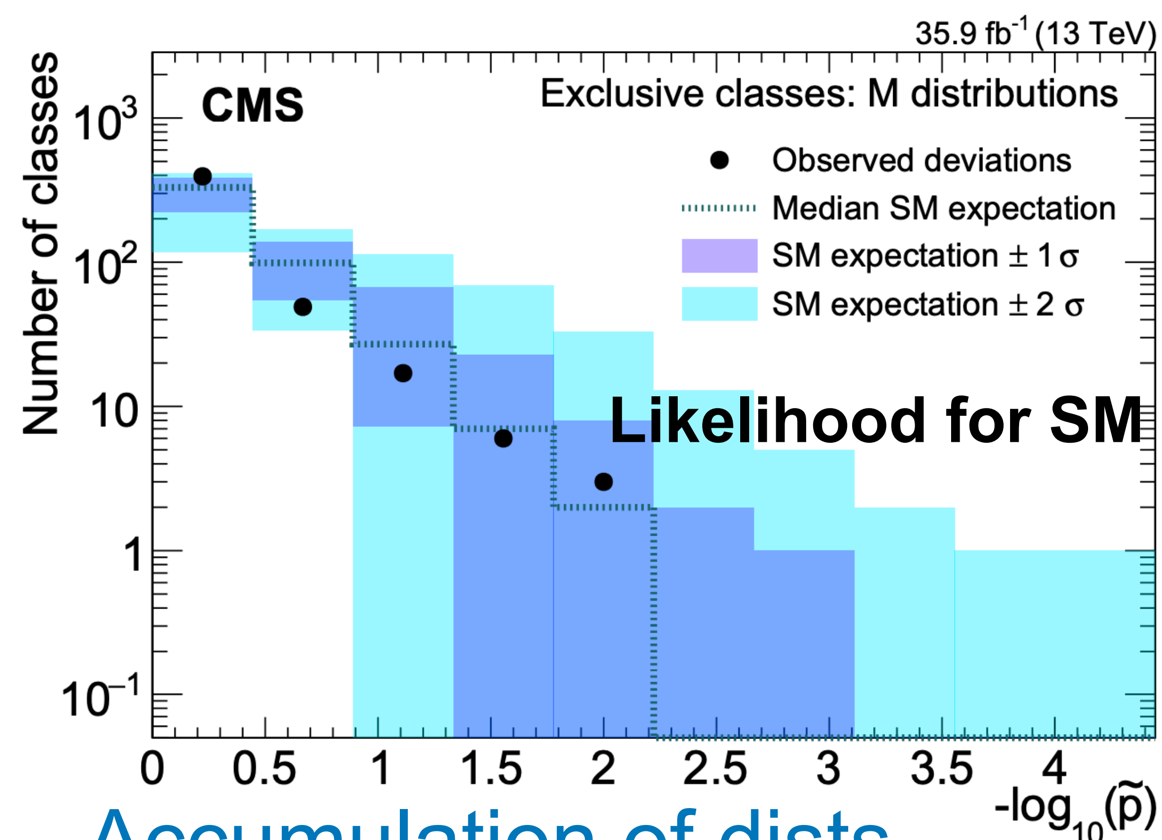
What has caused trend?

- The power of computing
  - Complex many parameter fits run much faster these days
  - Newer optimization strategies that are proven to be robust
  - Along with the ease of use of complex fitting tools
    - Many tools now auto build likelihood and minimize
- A better understanding of our simulation
  - Many processes are understood
  - Steps to making categories has become progressively simpler
- Encroaching on a general philosophy to do more at the same time

From this trend

- Some old ideas are starting to be taken more seriously
  - Can we perform analyses on a broad range of data at once





Accumulation of dists compared to SM

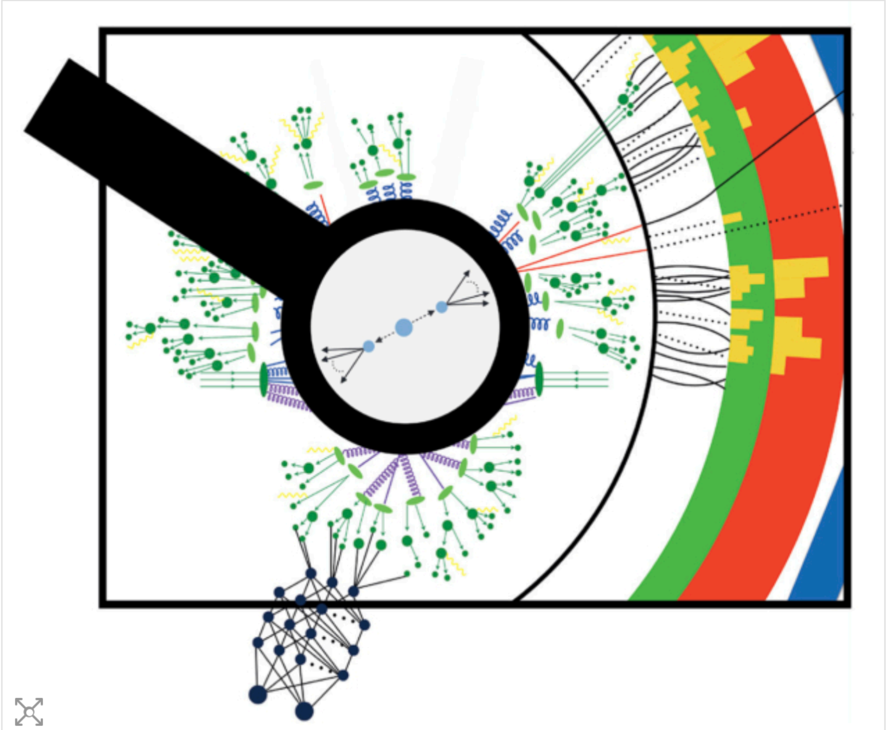
Two Anomaly Challenges

LHC Olympics 2020

arxiv/2101.08320

David Shih, Ben Nachman, Gregor Kasieczka

Dark Machines

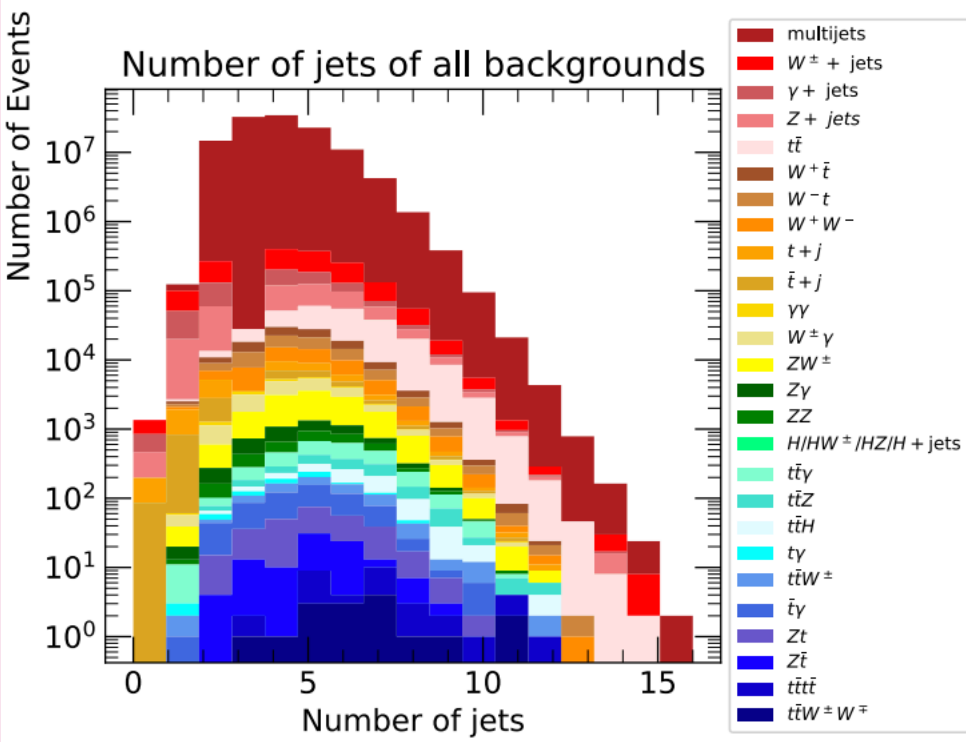


arxiv/2105.14027

Challenge: Hide signal(s) in a lot of data
See if the community can find it

Anomaly Data

Dark Machines



Channel 1: 214,185 SM events
- $H_T \geq 600$ GeV
- $MET \geq 200$ GeV
- $\mathrm{MET/H_T} \geq 0.2$
- at least 4 (b)-jets with pT > 50 GeV
- at least 1 (b)-jet with pT > 200 GeV

Channel 2a: 20,005 SM events
- $MET \geq 50$ GeV
- at least 3 μ/e with pT > 15 GeV
- at least 1 (b)-jet with pT > 200 GeV

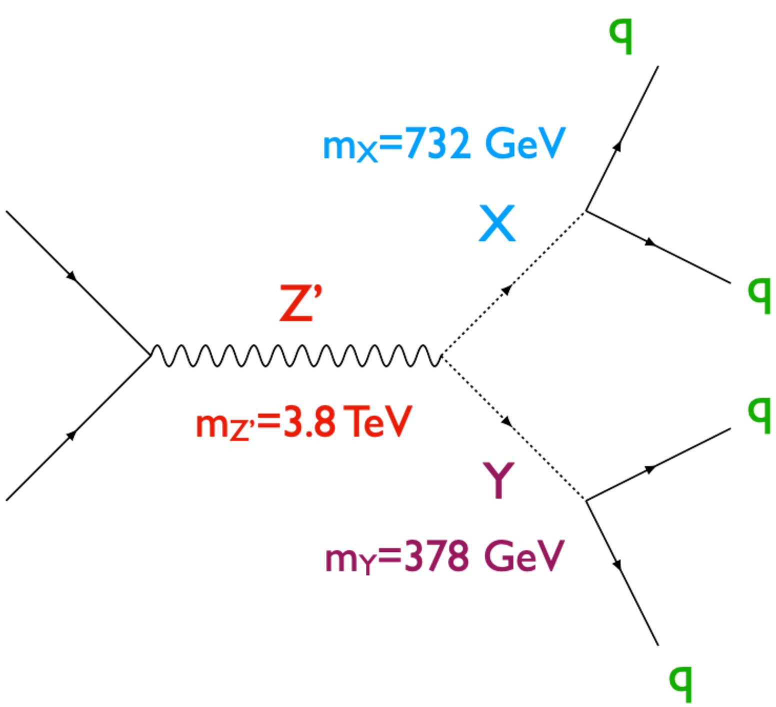
Channel 3: 8,544,111 SM events
- $H_T \geq 600$ GeV
- $MET \geq 100$ GeV

Channel 2b: 340,268 SM events
- $MET \geq 50$ GeV
- $H_T \geq 50$ GeV
- at least 2 μ/e with pT > 15 GeV

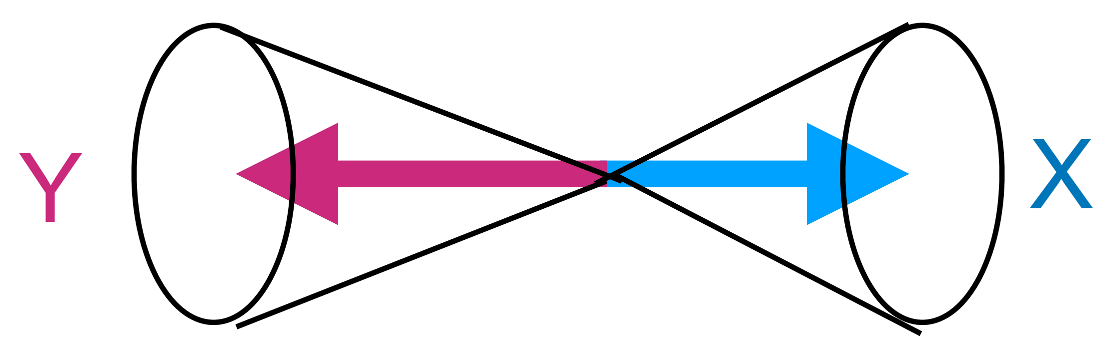
Complicated Signals
Many final states

Black Box #1

LHC Olympics



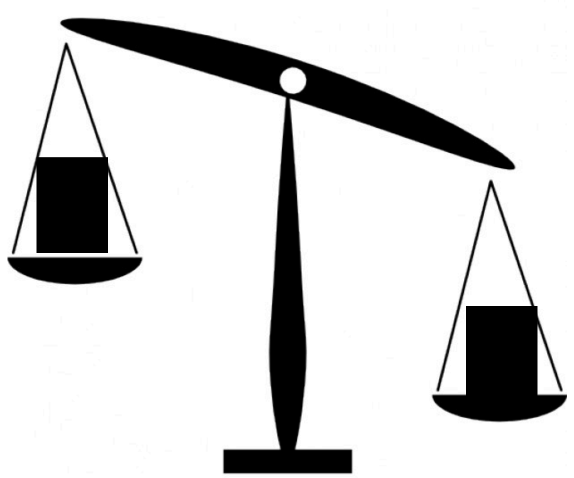
Single Signal
With a Dijet (or trijet) topology



Anomaly Strategies@LHC

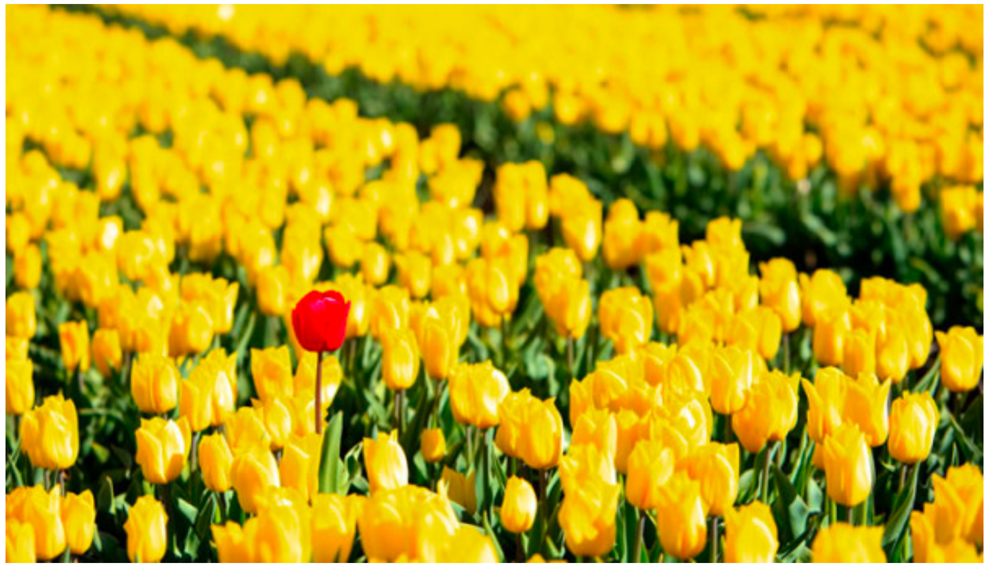
- Anomaly Strategies at LHC fall into two categories

I know regions where new physics does not exist



I want to leverage those regions against other parts of the data to find differences

I know how to predict all collisions



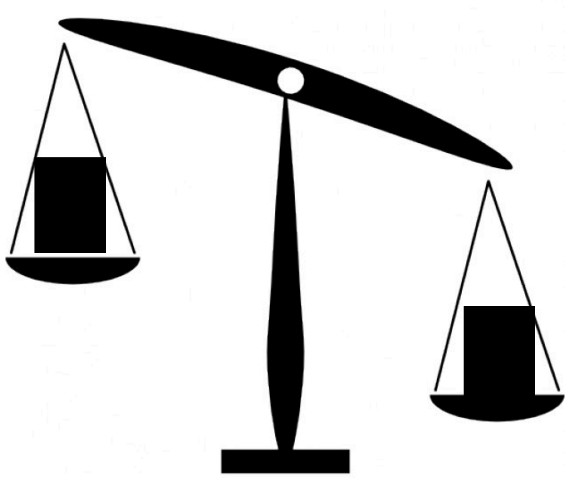
Are there any collisions that I cannot predict?

Anomaly Strategies@LHC

- Anomaly Strategies at LHC fall into two categories

Weakly-Supervised

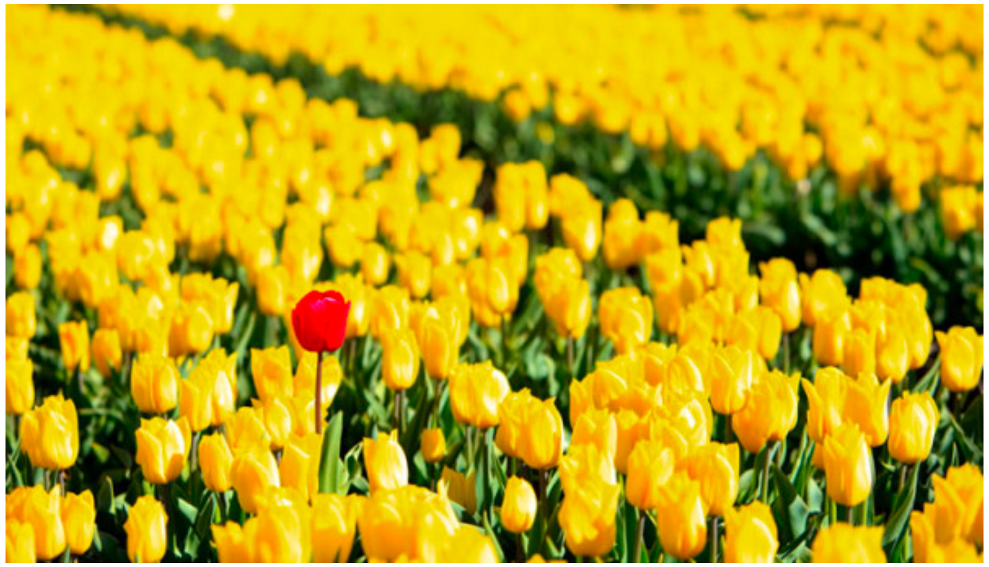
I know regions where new physics does not exist



I want to leverage those regions against other parts of the data to find differences

Autoencoders

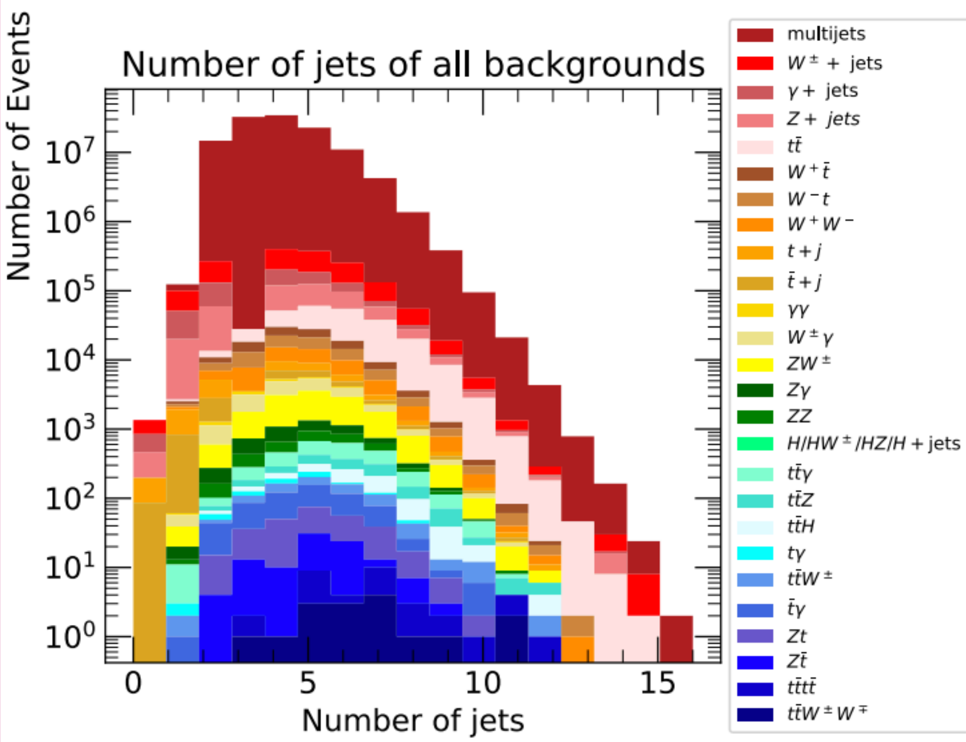
I know how to predict all collisions



Are there any collisions that I cannot predict?

Anomaly Data

Dark Machines



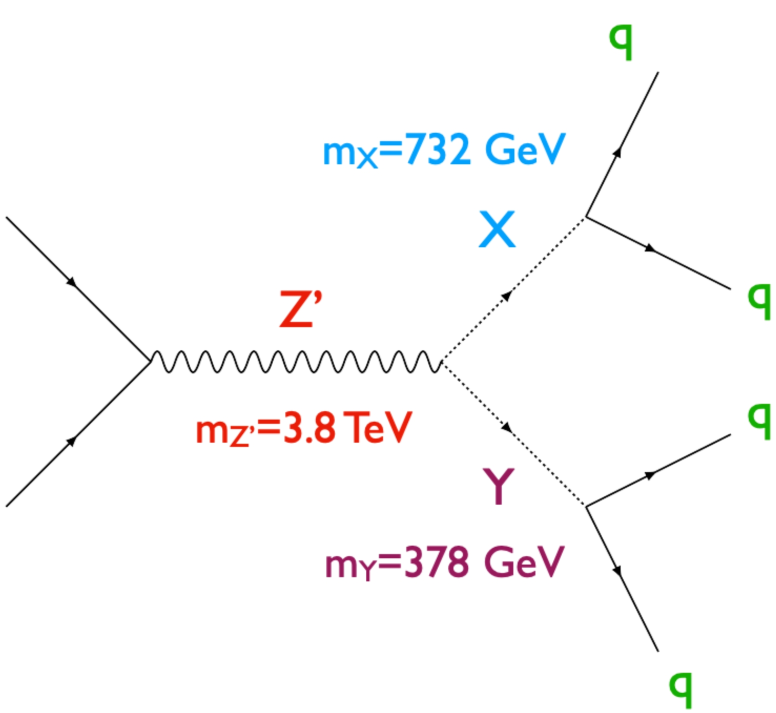
General emphasis was on Signal Prior free approaches

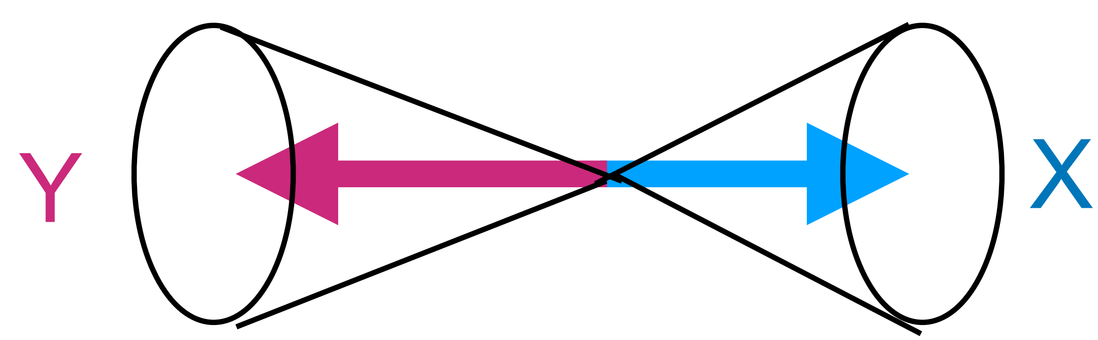
Many different types of Autoencoders

Black Box #1

LHC Olympics

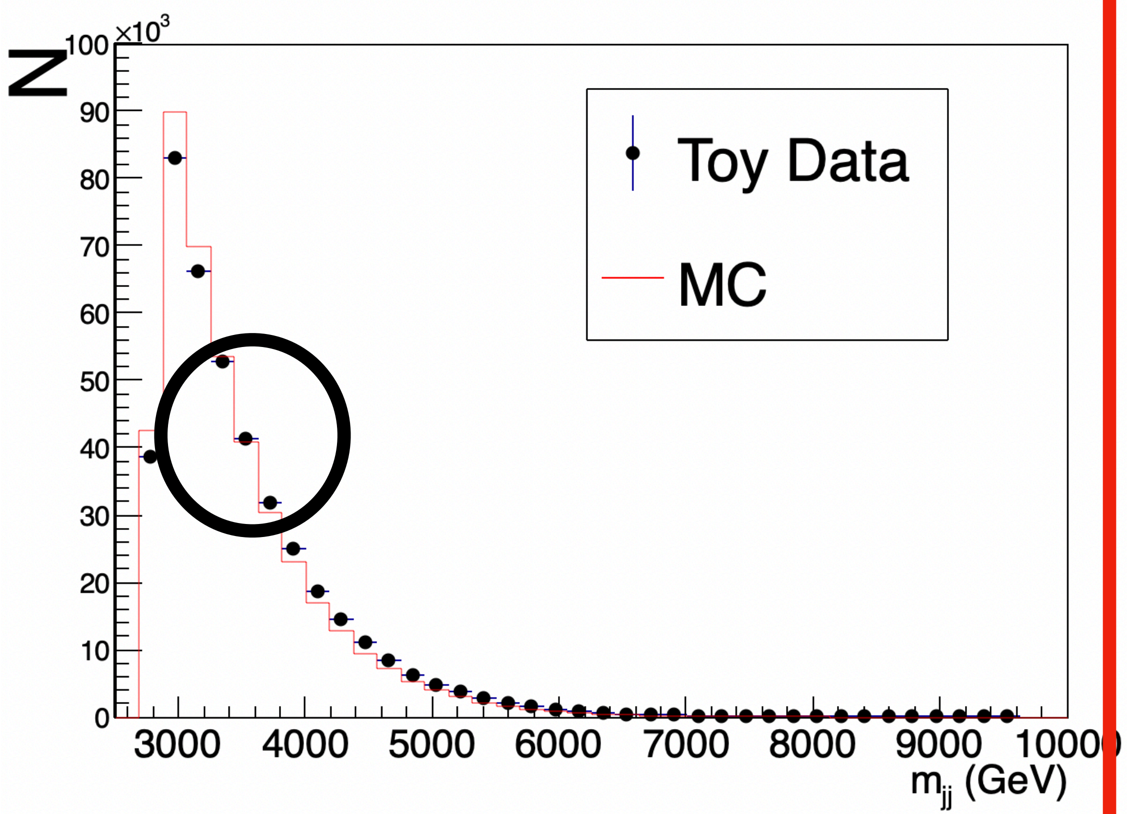
Weak-Supervision and other Signal assumptions were put in Due to dijet topology



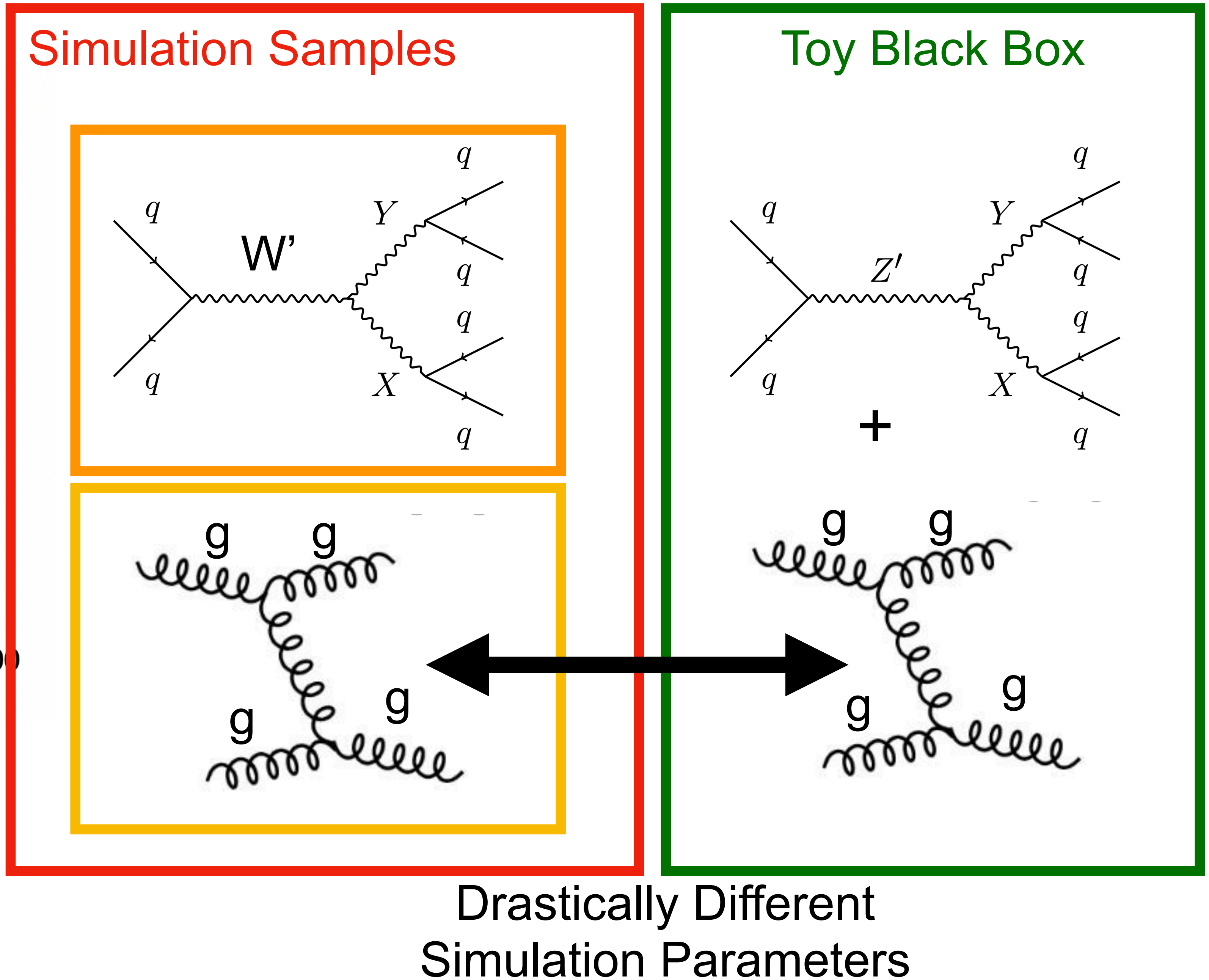


Simulation

Samples



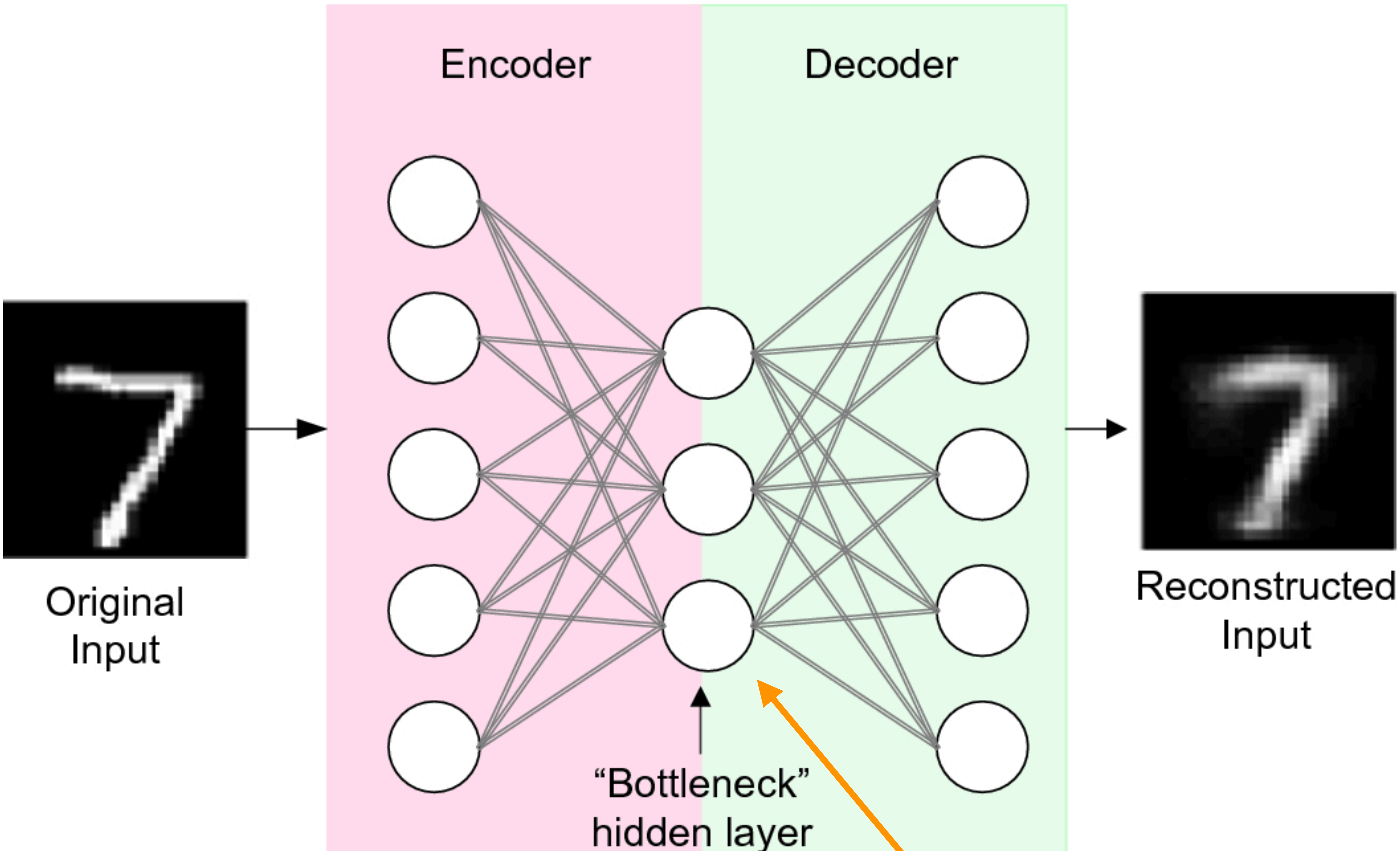
Data and simulation shower parameters had differences



- Aim was to emulate a real search as much as as possible
  - Simulation and Toy Data are released (Sim and Data different)

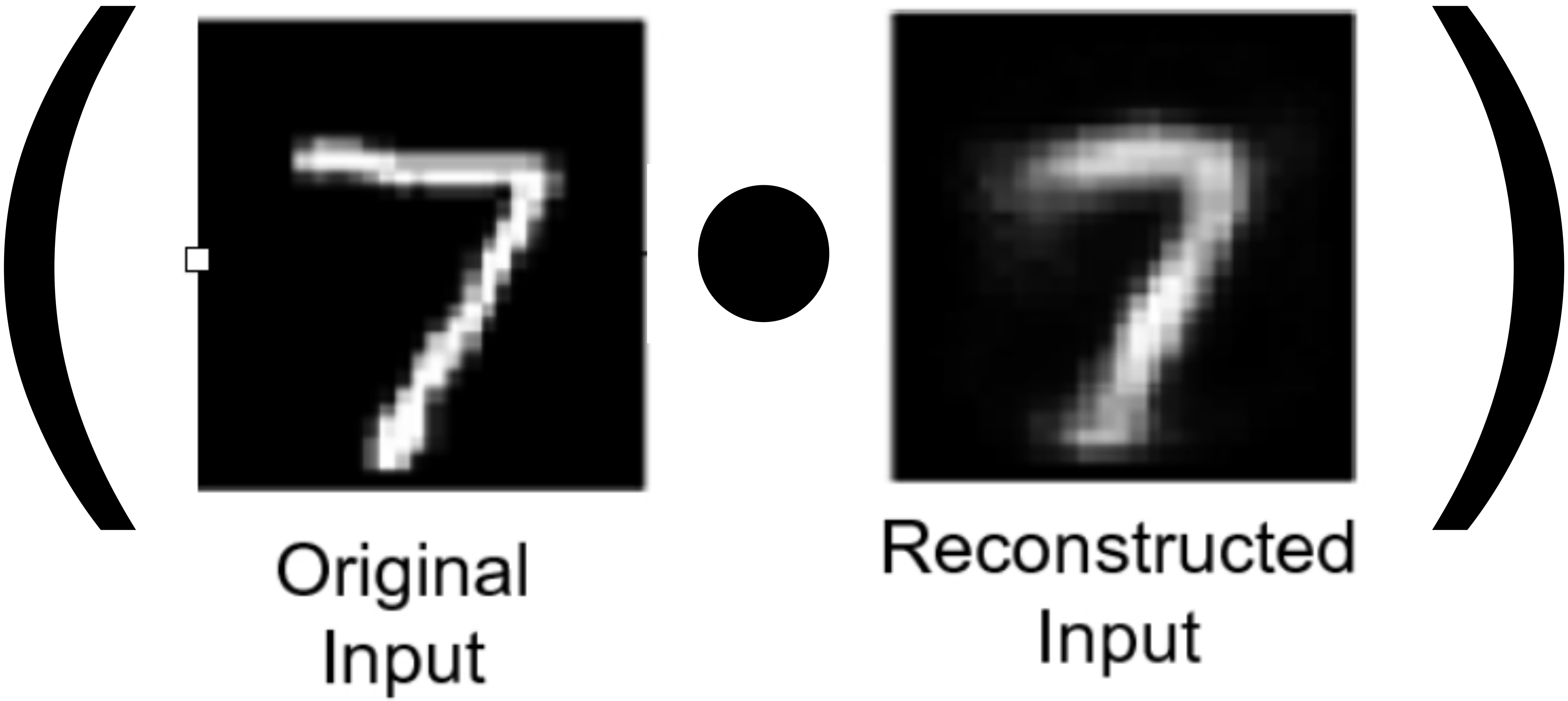
What are people thinking about to find anomalies?

Autoencoders



Strategy is to create a space in the middle that embodies all features of physics

Autoencoders



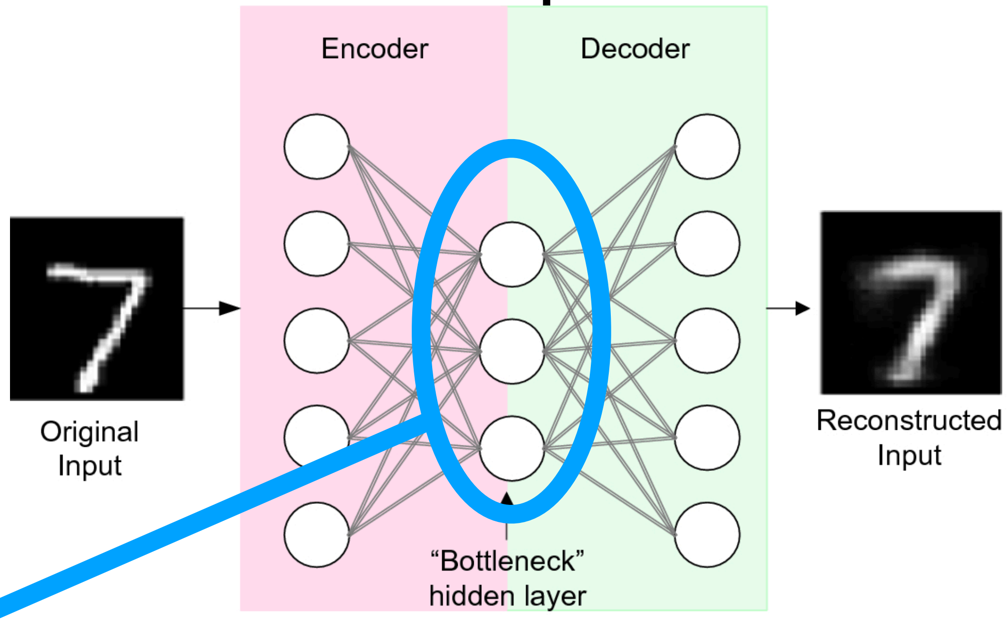
Dot product the input and output

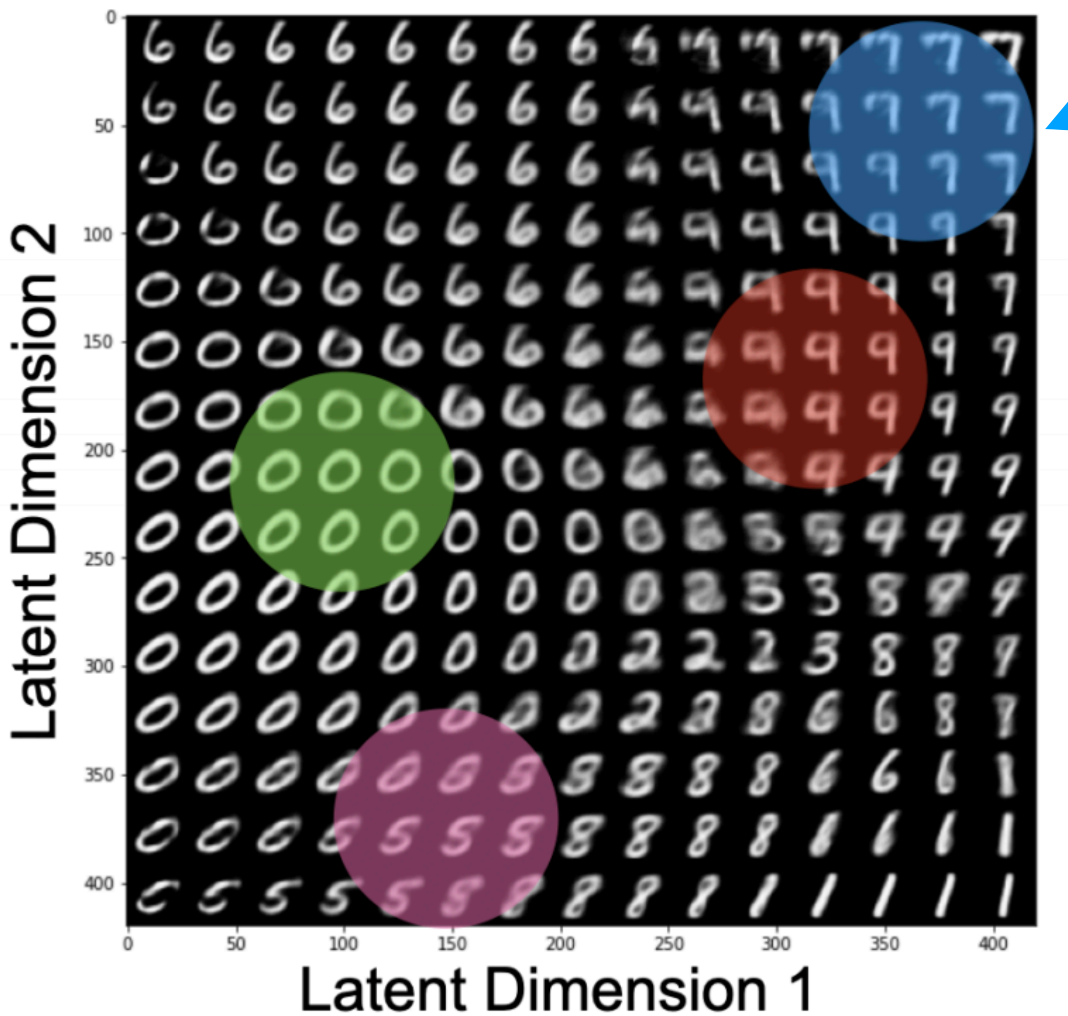
Large Value : Good

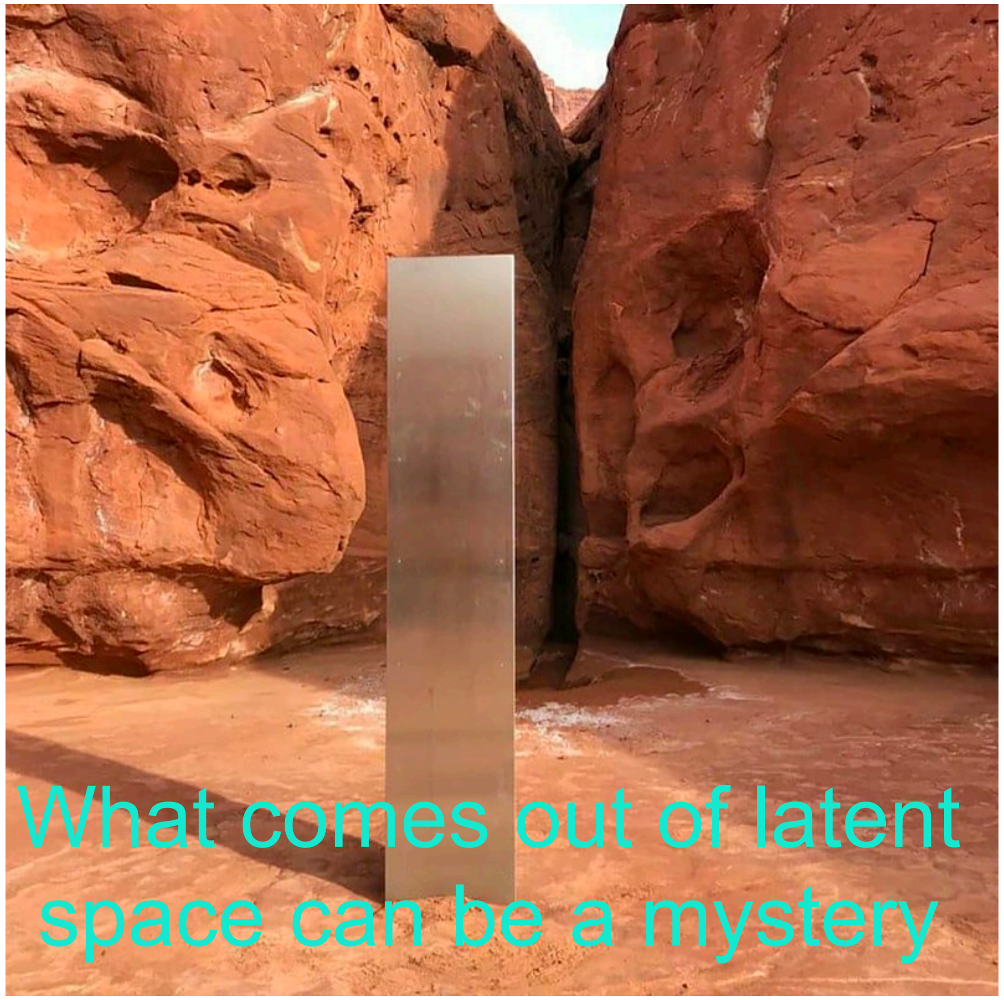
Small Value : Anomaly

The Latent Space

- Deep learning algos tend to focus on the latent space
- What is the latent space?
  - Its whatever you want it to be

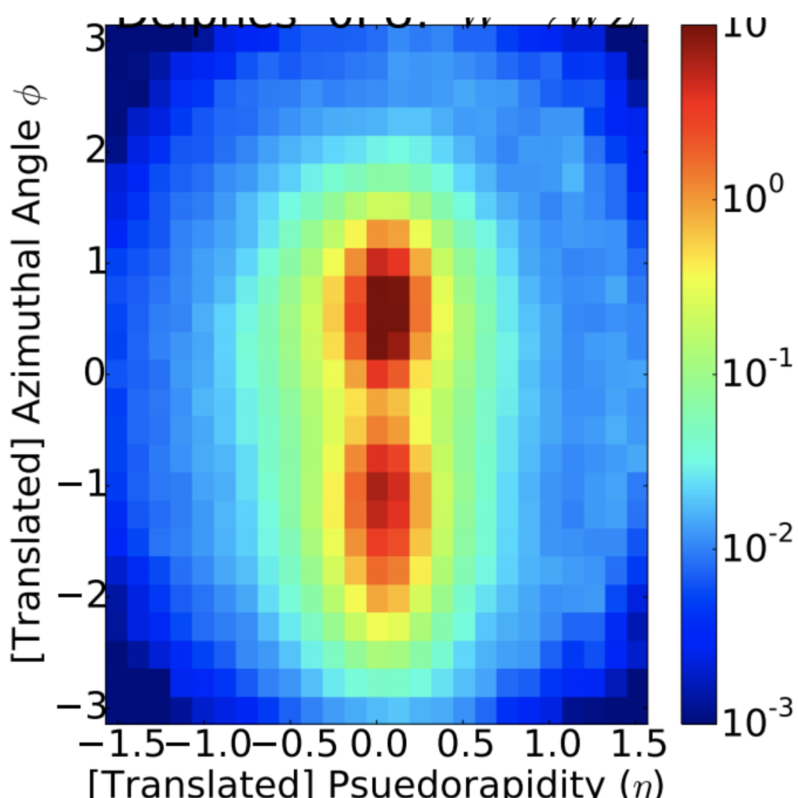






Encoder Progression

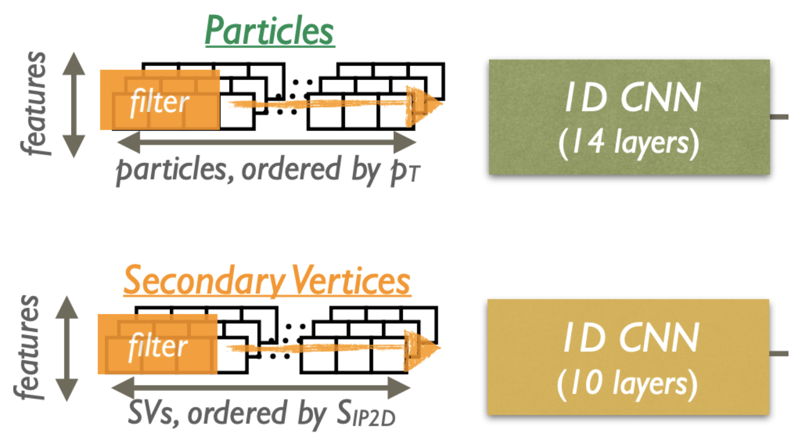
2016



Images

(not lorentz invariant)

2018

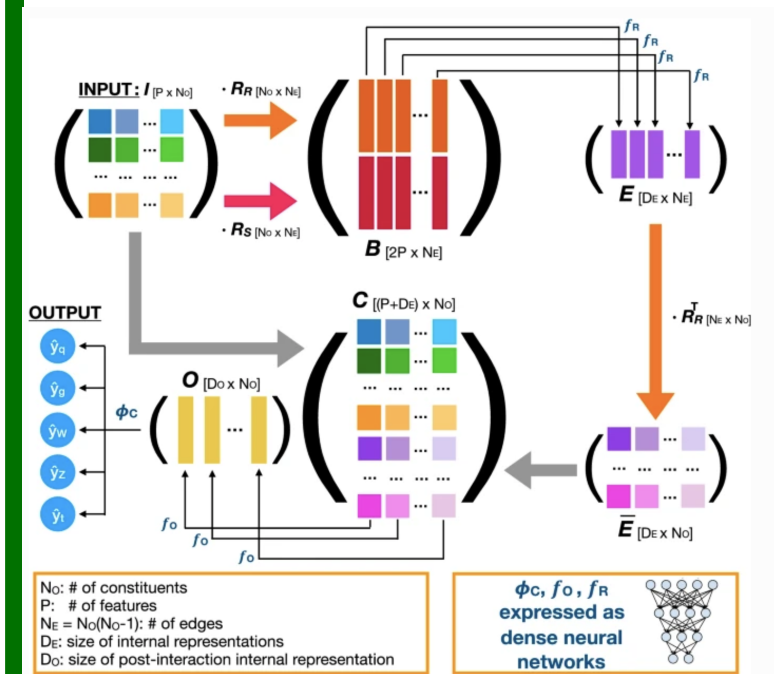


Particles and SVs
with 4-vectors+features

Particles

(limited correlations)

2020



Graphs

(Particles+correlations)

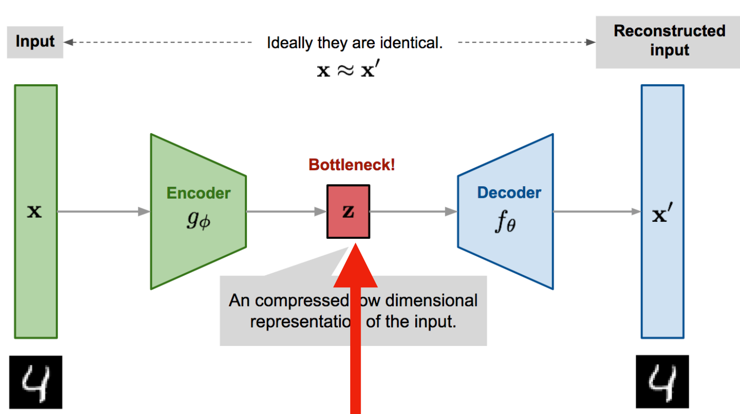
Current collaboration results

Progressively moving towards use of more info

Autoencoder Progression

- Autoencoders are gaining popularity in HEP just now

Dawn of Time

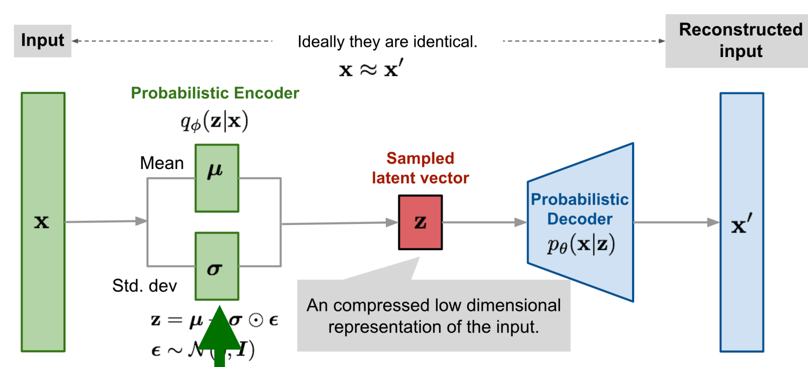


Small latent space
that encodes physics

Autoencoder

Not smooth

2015

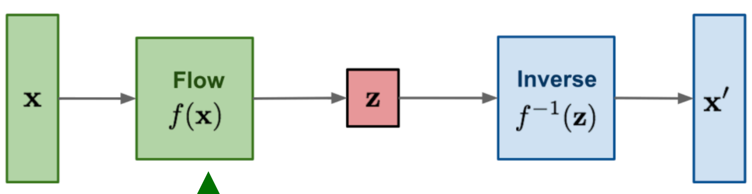


Inputs smeared w/gaussian
before latent space

Variational AE/GAN

Smooth AE

2017



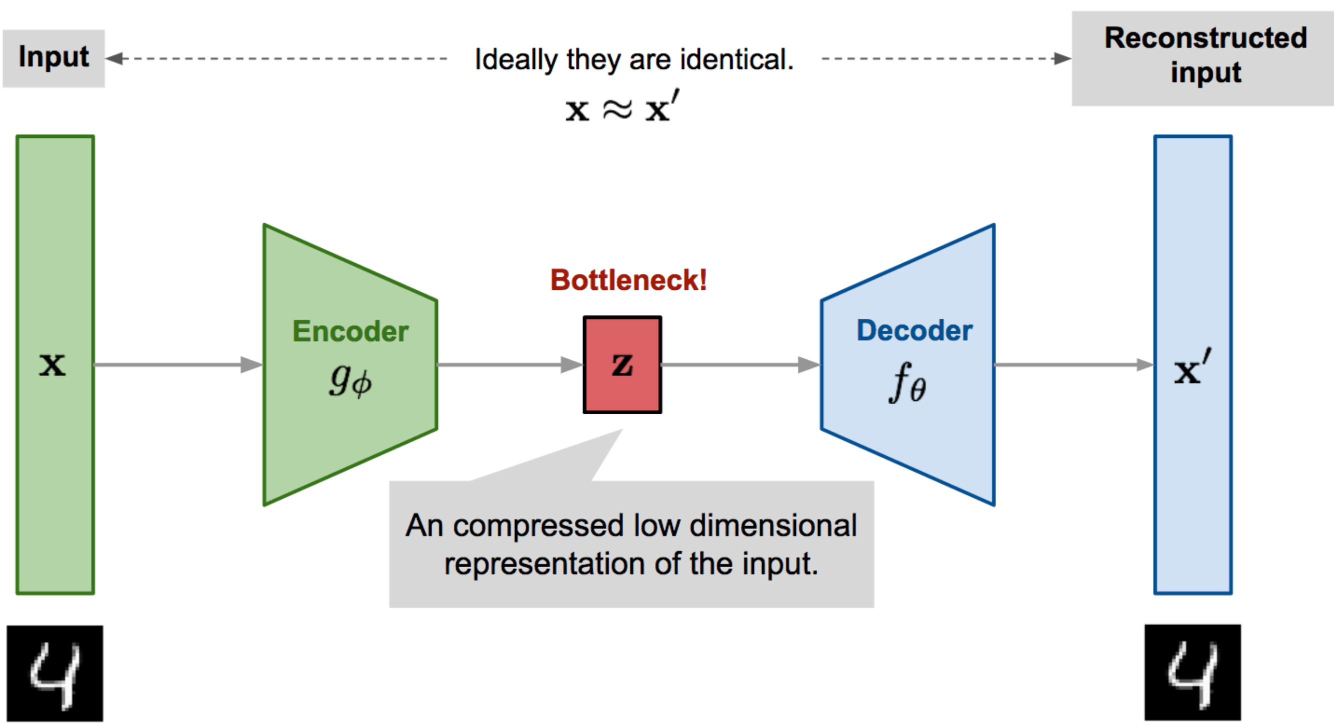
Inputs transformed before
entering latent space

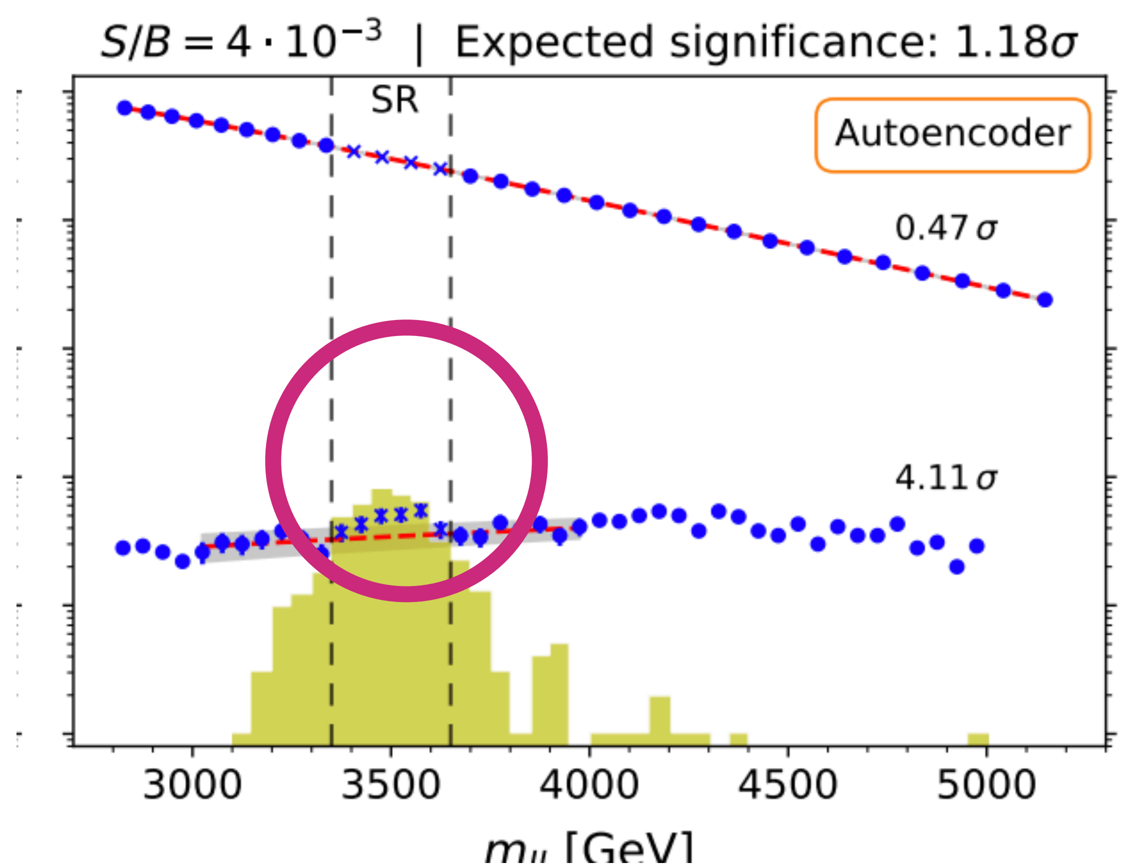
Normalizing Flow

Non-Gaussian VAE

We started with AEs

- Try to repeat the inputs with the outputs



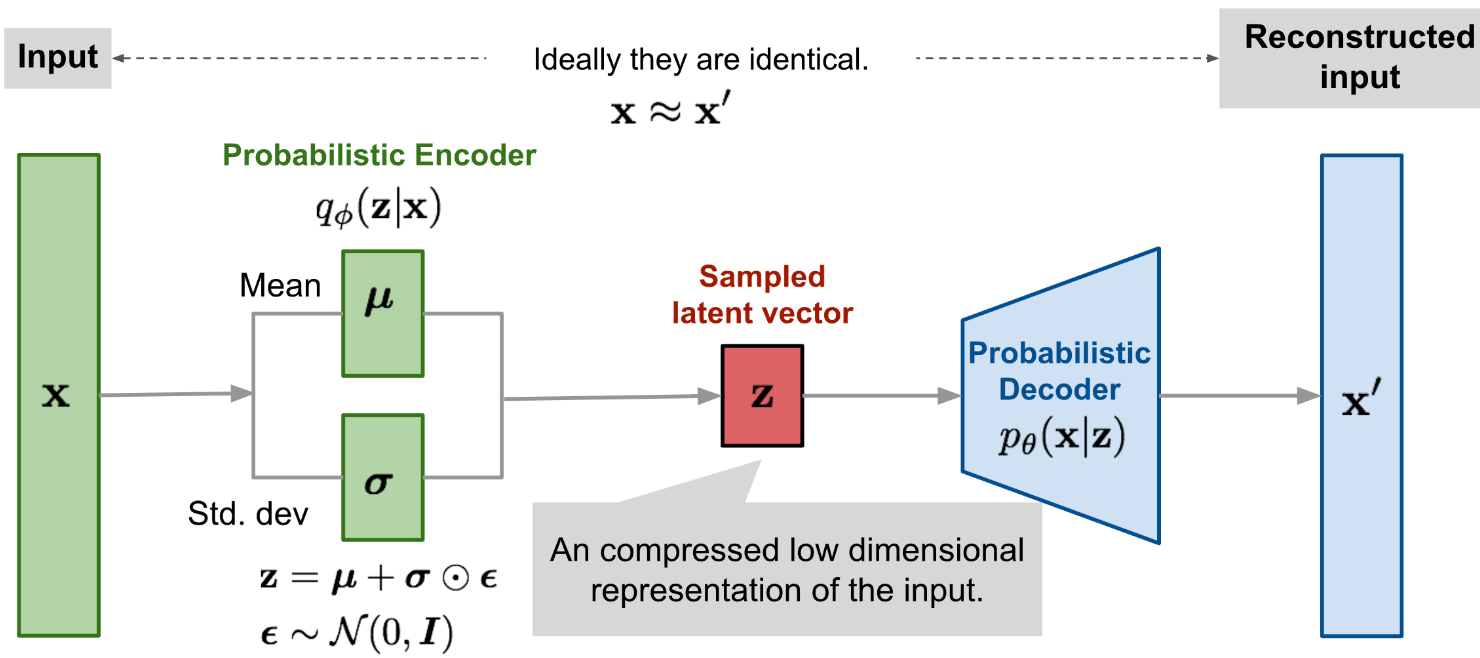


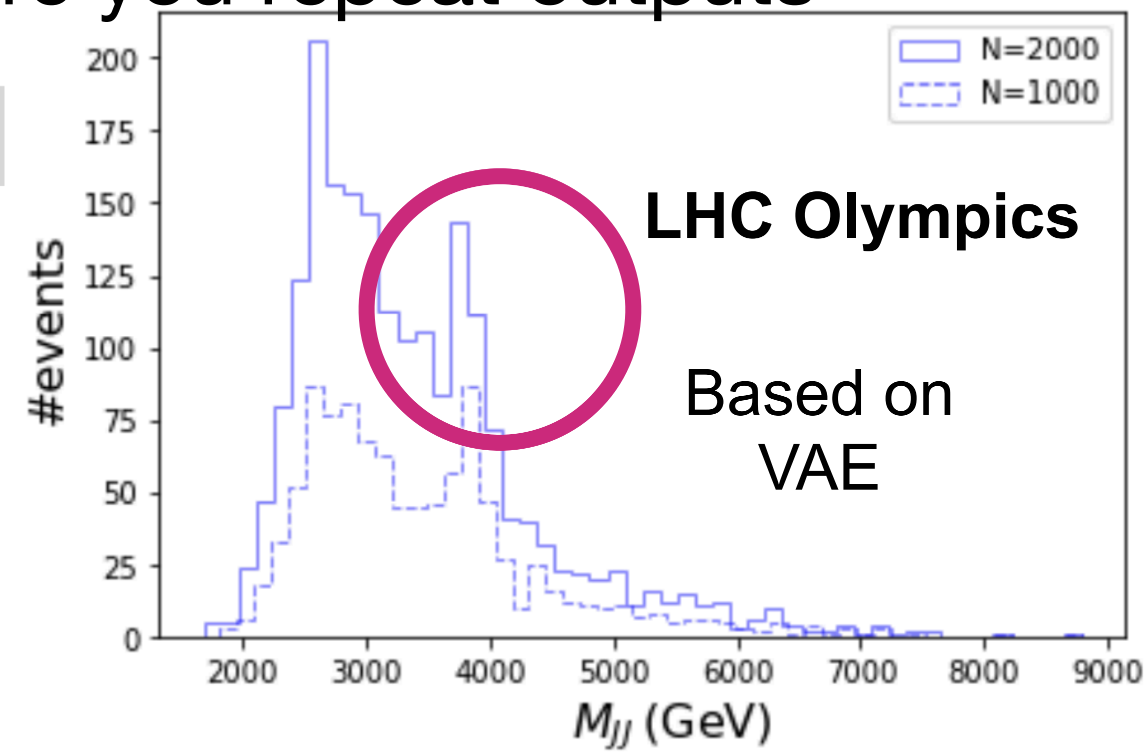
Anomaly Defined by how well reproduced the input is

An anomaly will not reconstruct the input well

We updated with VAEs

- Try to repeat the inputs with the outputs
  - but Smear with gaussians before you repeat outputs





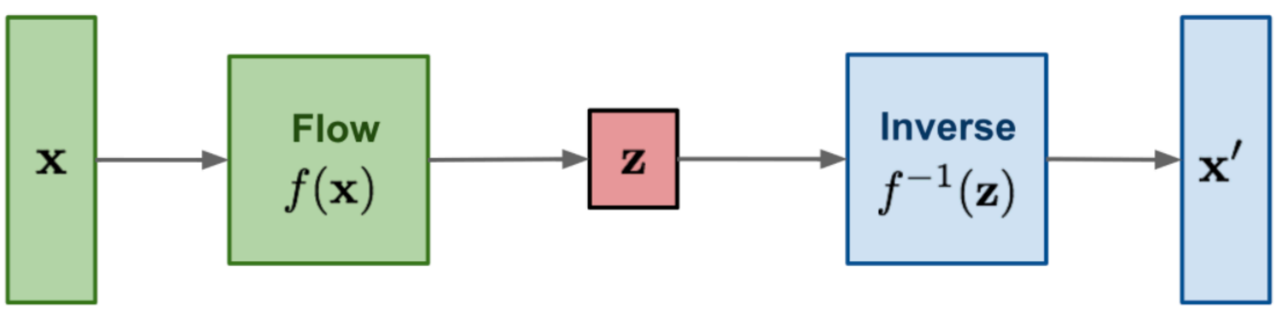
VAE makes latent space continuous which improves performance

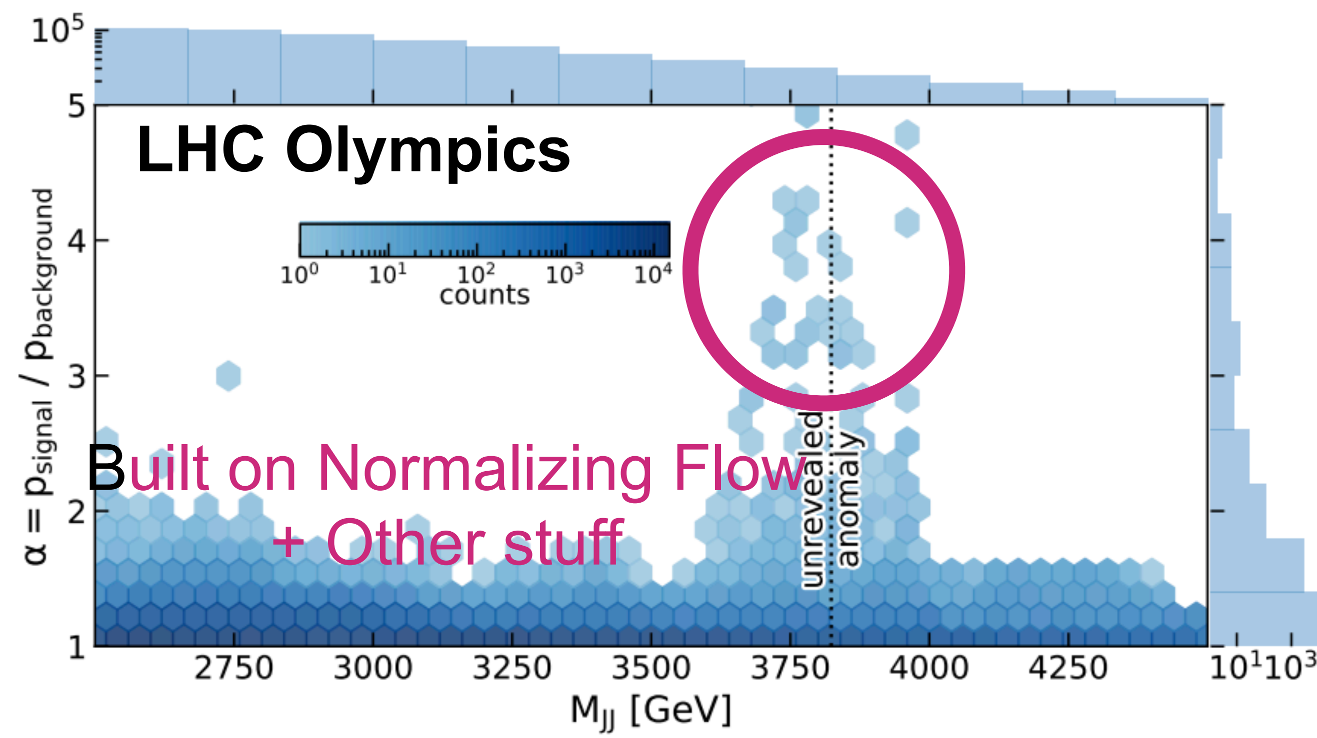
Found to be very effective (dark machines)

Particularly when adding tight constraints on μ and σ

added Normalizing Flows

- Try to repeat the inputs with the outputs
    - But transform (and smear) outputs



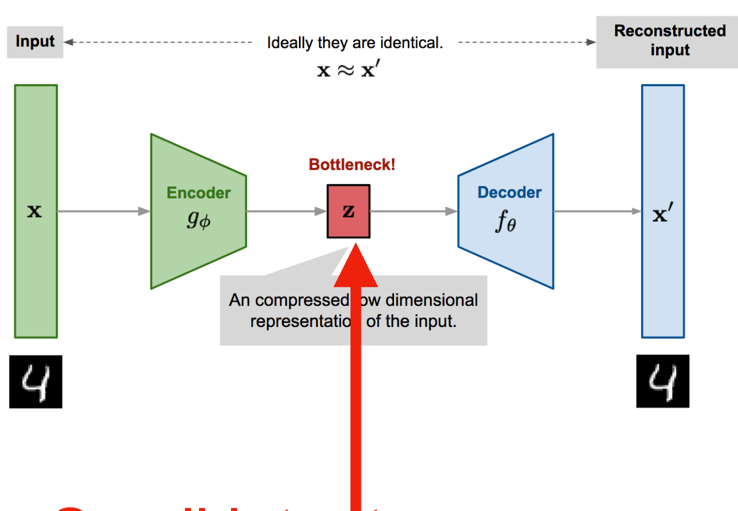


NF transforms the latent space so it has a lot more fexibiity
Gaussian smearing and motion in space can capture physics
These tend to perform the best in terms of anomaly detection

Autoencoder Progression

- Autoencoders are gaining popularity in HEP just now

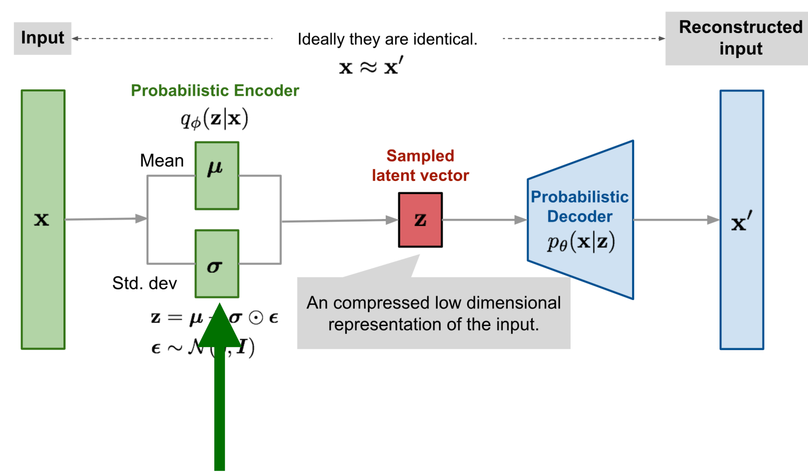
Dawn of Time



Small latent space
that encodes physics

Autoencoder
Not smooth

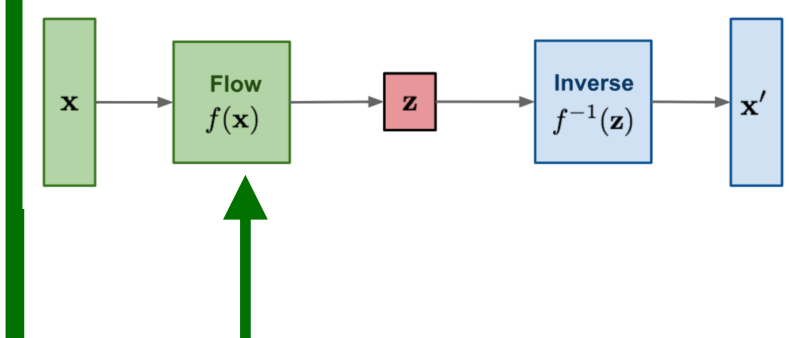
2015



Inputs smeared w/gaussian
before latent space

Variational AE/GAN
Smooth AE

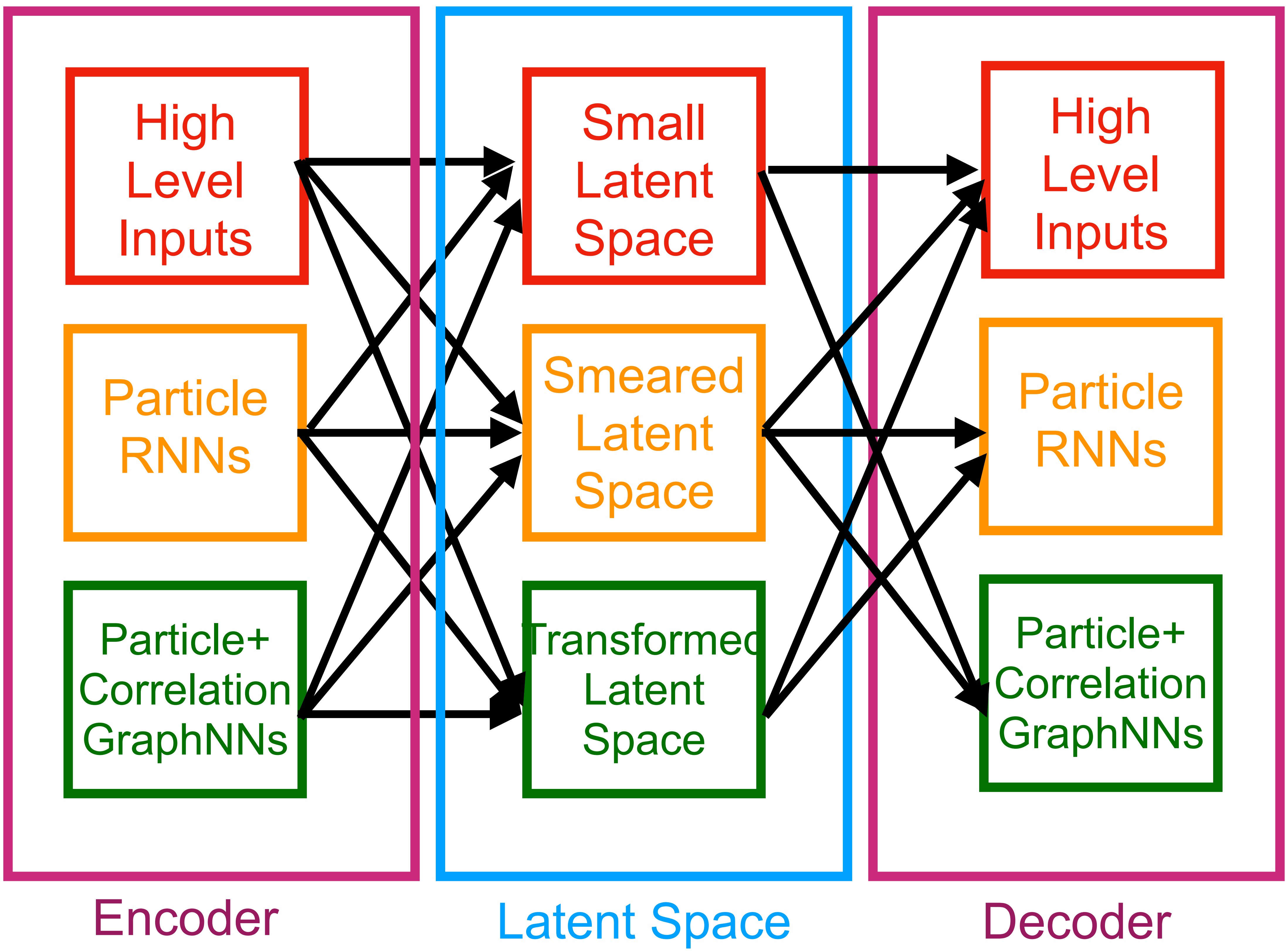
2017



Inputs transformed before
entering latent space

Normalizing Flow
Non-Gaussian VAE

Combinations



Weak Supervision

How do we separate two samples (one with anomalies)

Sample A

VS

Sample B

Difference:

Strategy: Train the data in A agains B

Challenge:

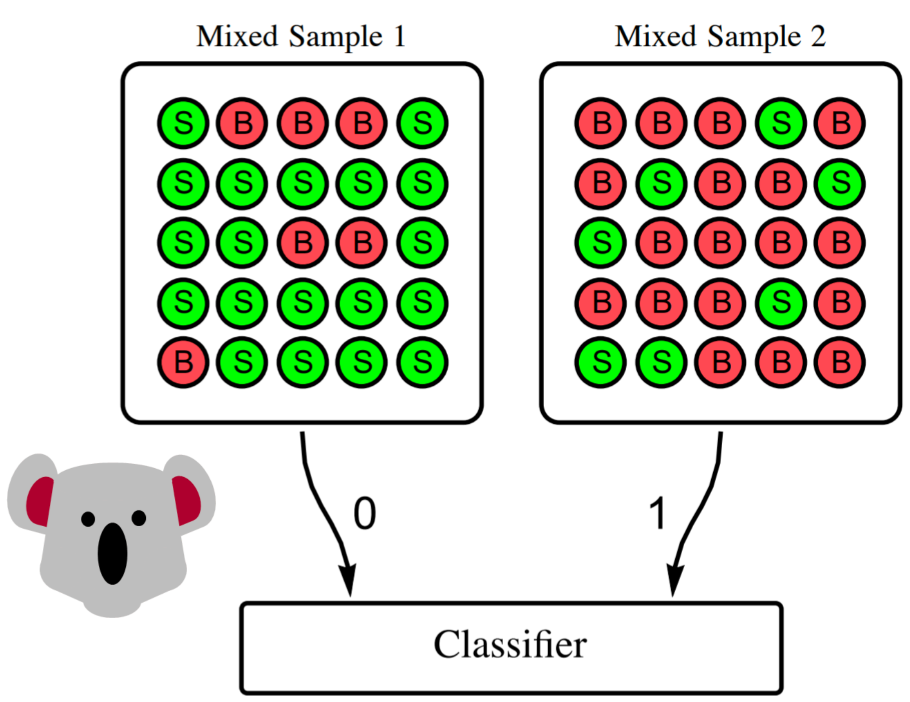
Must all be same in A and B

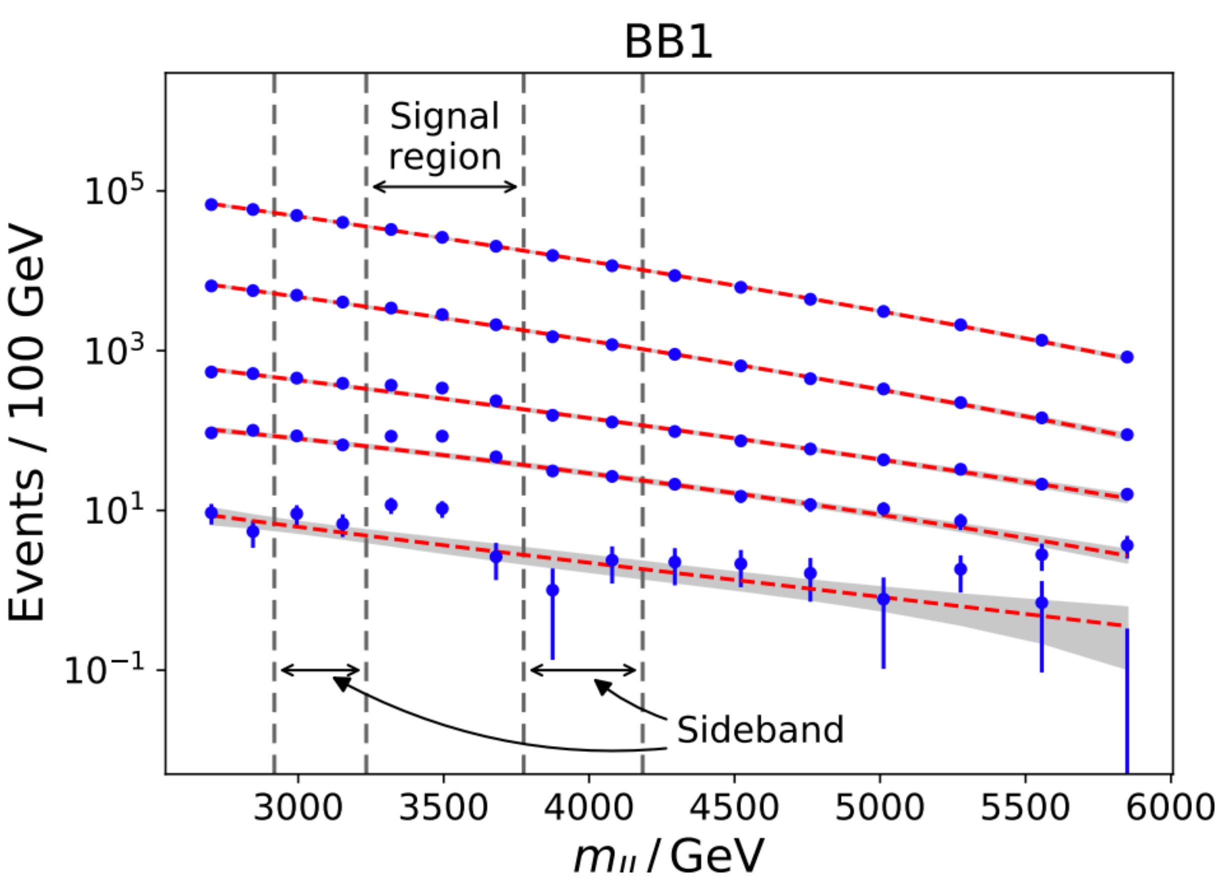
More realistic example

How do we train samples with variations of populations of an anomaly

Classification w/o Labels

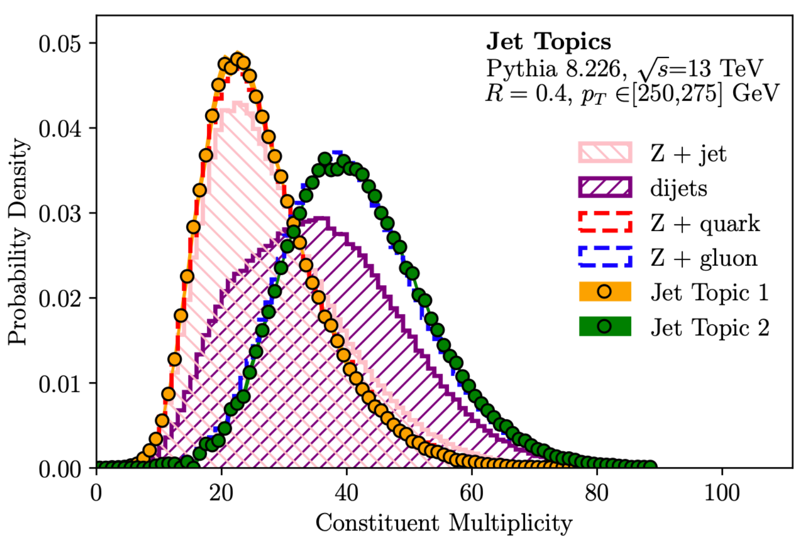
- CWoLA approach aims to exploit differences in datasets
  - Can play one region of data off the other
  - Provided you can separate out the two approaches





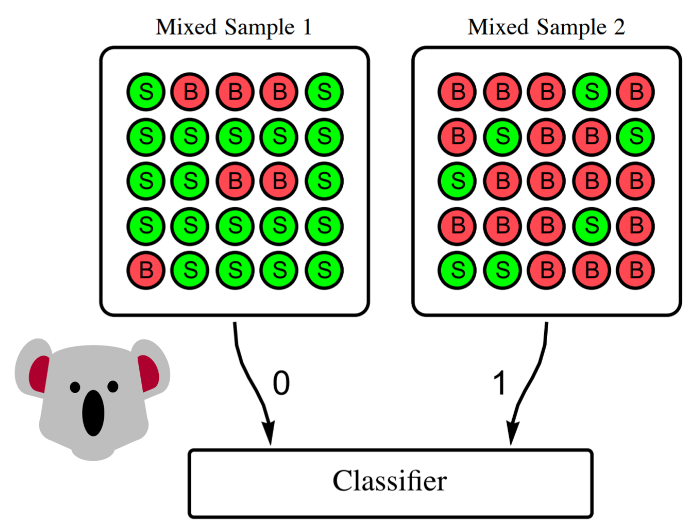
Training Strategies

Topic Modeling/ Clustering



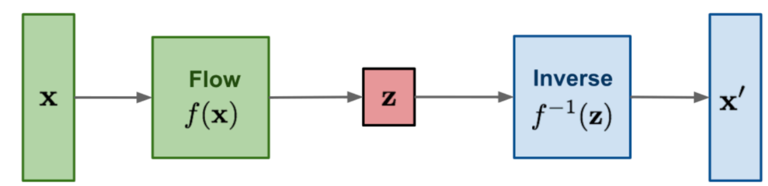
Split a histogram into multiple distributions by looking for separate regions

Classification W/O Labels



Separate out Sample 1 from Sample 2 by hidden signal

Likelihood Discrimination

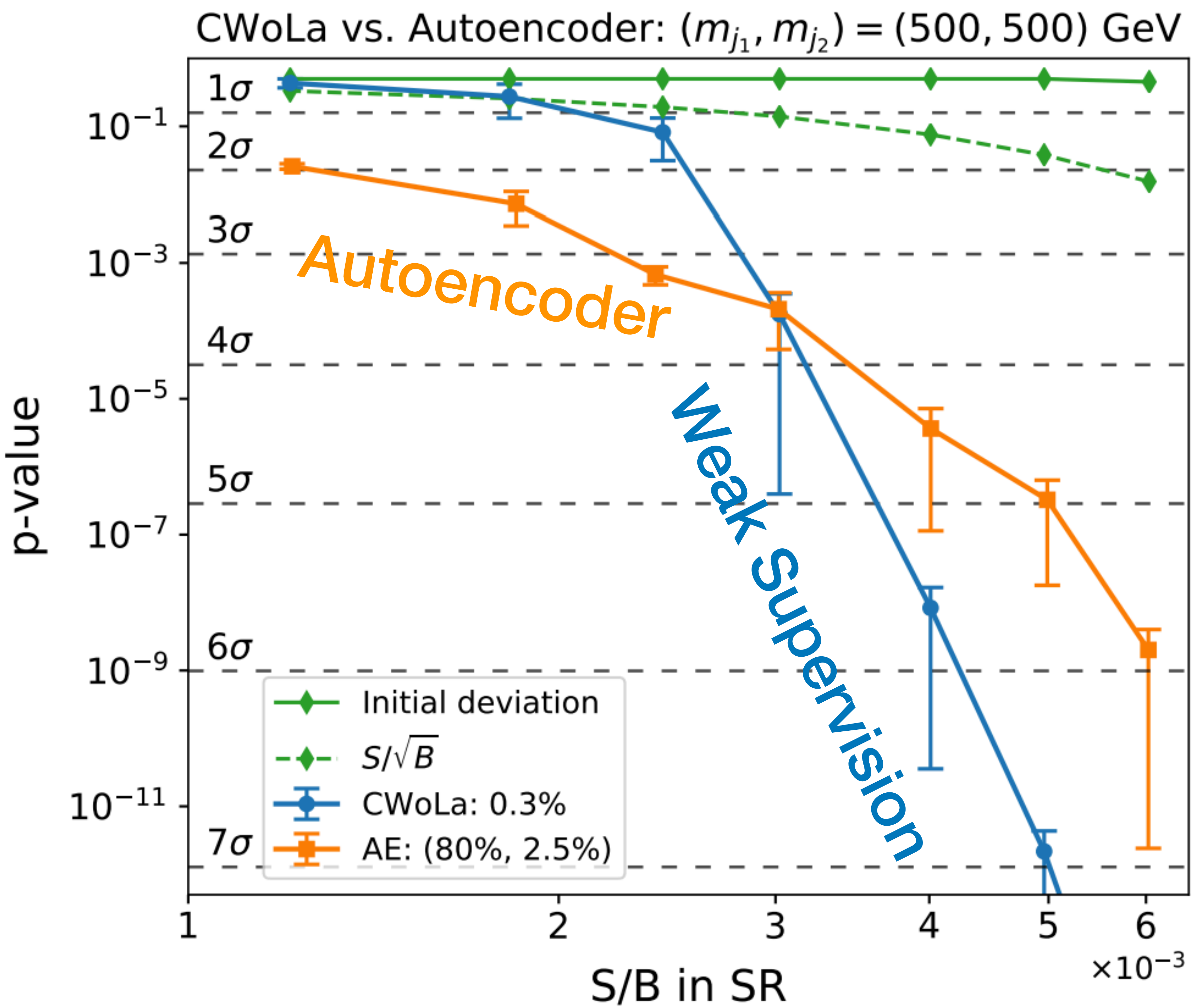


$$p(x|x_c) = \pi(f_{x_c}(x)) \left|\det\left(\frac{\partial f_{x_c}(x)}{\partial x}\right)\right|$$

$\boxed{p(x \,|\, x \in A)}$ $\boxed{p(x \,|\, x \in B)}$

$$R(x|m) = \frac{p_{\text{data}}(x|m)}{p_{\text{background}}(x|m)}$$

Performance Observations

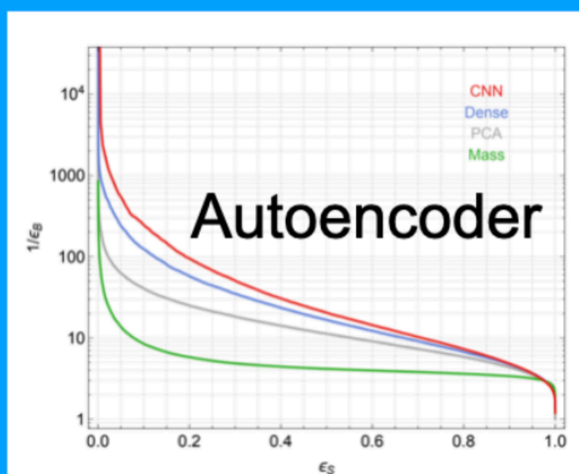


Normalizing Flow approaches stood out

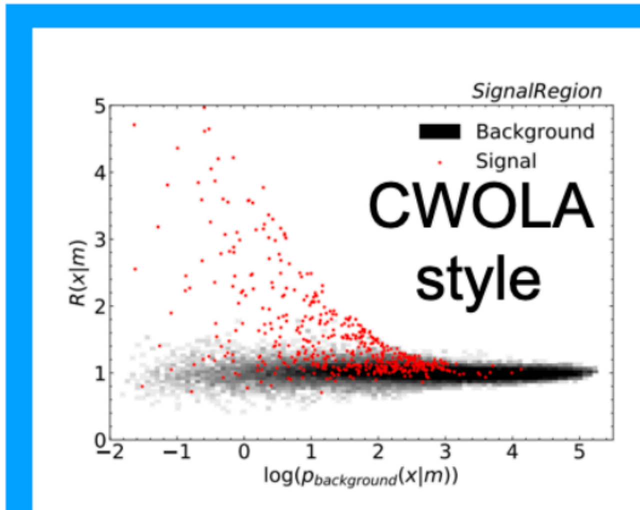
So did Observable based encoding(not sure why)

Anomaly Searches Spectrum

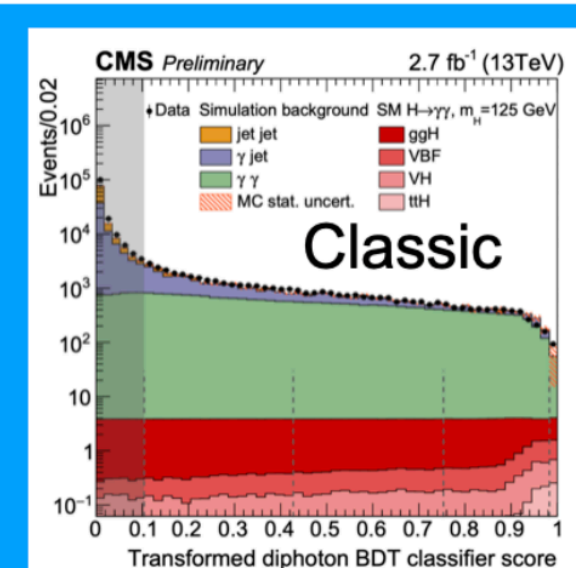
Prior Free → Fully Supervised



Knowledge of Background



Signal in (Mass) Window



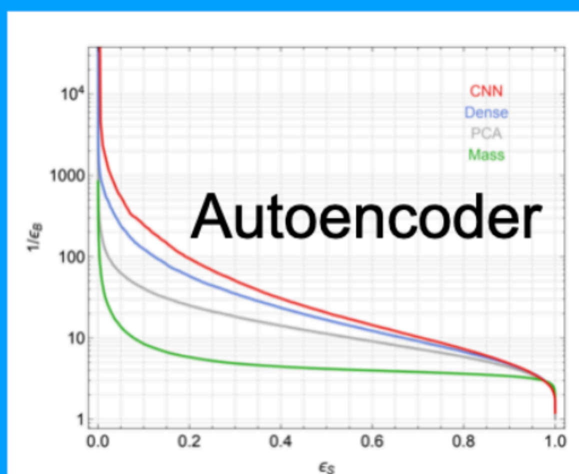
Fully Supervised

Gain in sensitivity by assuming a mass peak

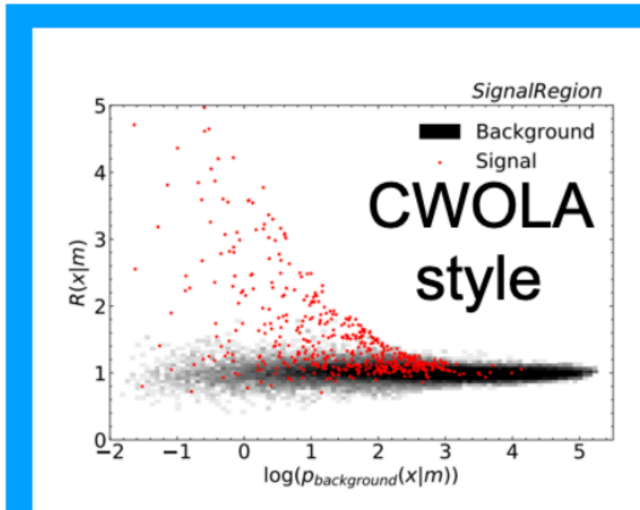
Adding assumptions about the signal

Playing with Prior

Prior Free → Fully Supervised



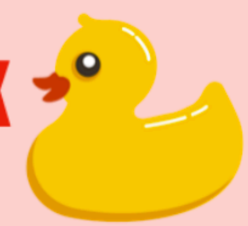
Knowledge of Background



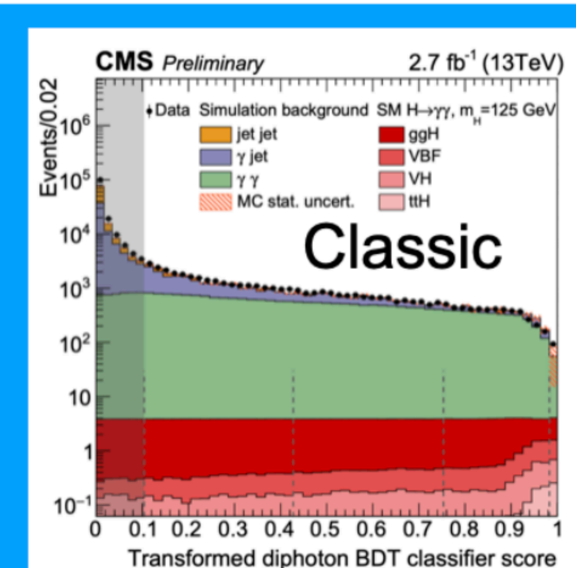
Signal in (Mass) Window

Semi-Supervised Approaches

QUAK



If it quaks like a duck?



Fully Supervised

Gain in sensitivity by assuming a mass peak

Adding assumptions about the signal

What if we decide to add more signal assumptions?

Can we make a robust construction?

Semi-Supervision

Autoencoder

A large amount of unlabelled data

Supervised Training

A small amount labeled data

- Use supervised training to catch  and not 

(i.e. Find anamalous tulips not anomalous something else in LHC a detector glitch)

Training Strategies

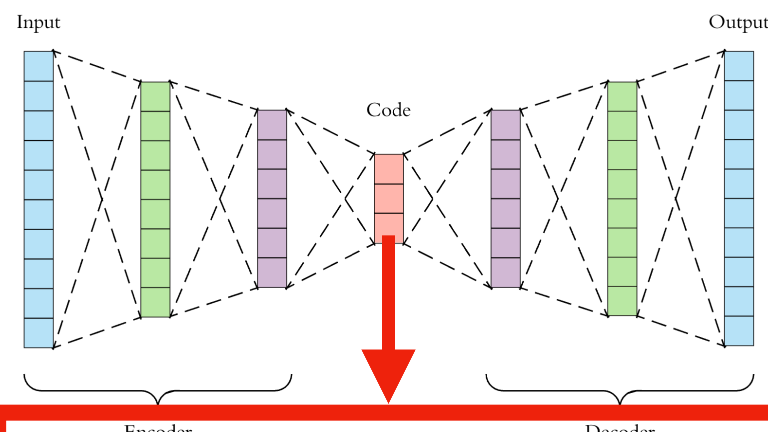
Just do a supervised training

Wrong Signals vs Background

Search for new physics by using an incorrect signal

Use classifier to isolate

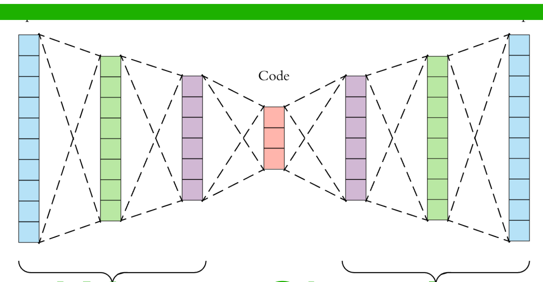
Use the latent space for autoencoder/ supervised



Background vs Wrong Signals

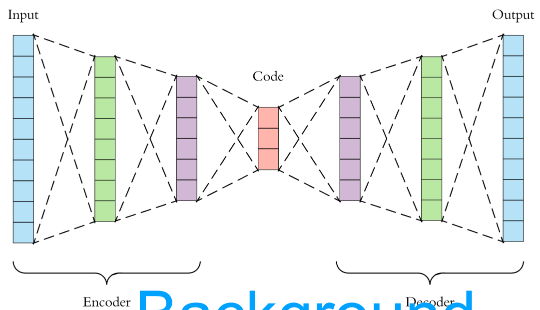
Use classifier loss for search

Construct Space from autoencoders on sig/bkg



Wrong Signals

y-axis



Background

x-axis

One-Shot Learning

One-shot learning aims to build a space of similar objects

Normalizing Flow → Similar

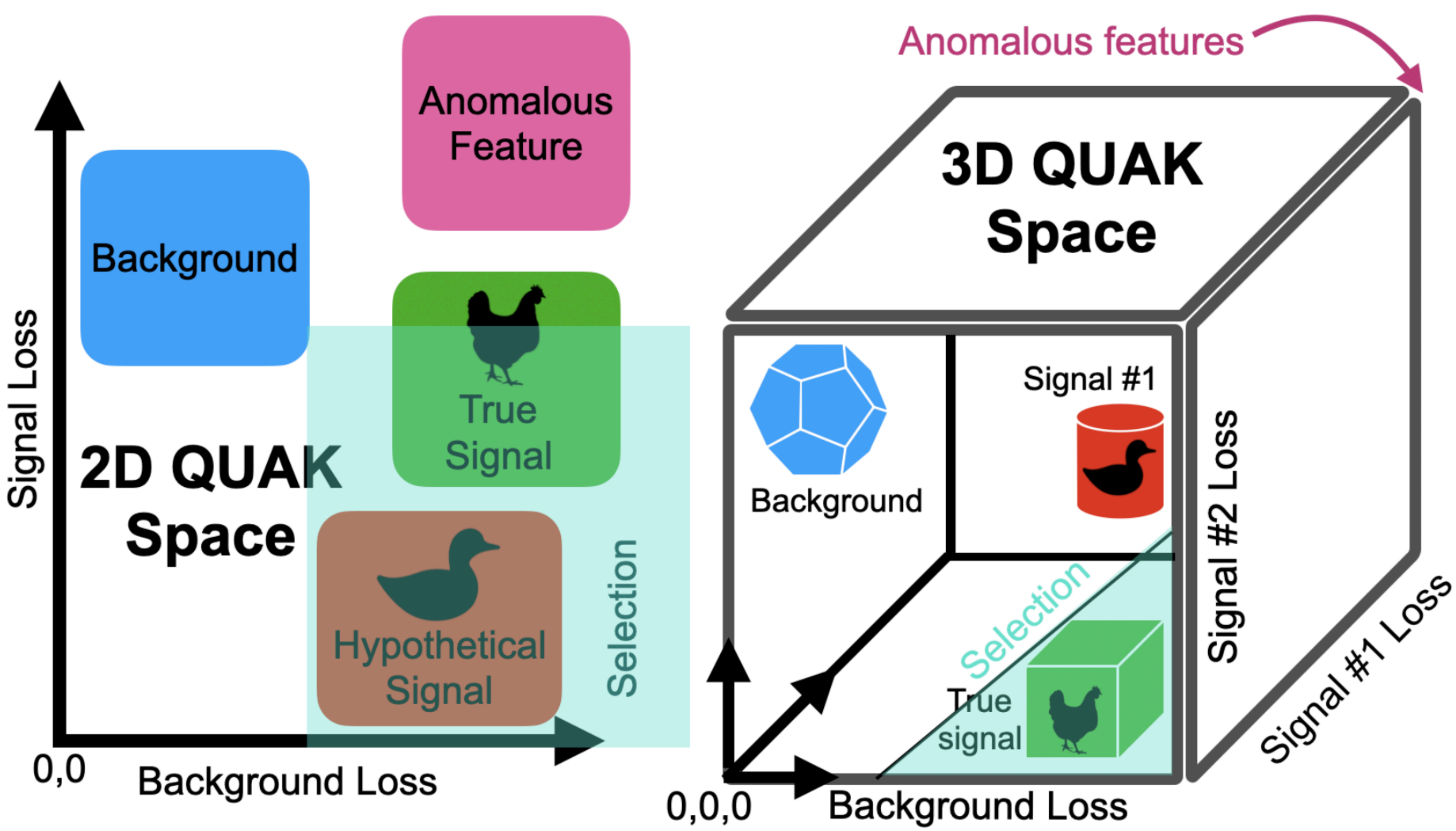
Our idea:
Normalizing Flow to build
a latent space of physics objects

QUasi-Anomalous Knowledge(QUAK)

Strategy: Train autoencoders on background and Signals

Choose a broad range of signals that capture physics of interest

Probe the result space for physics-like anomalies

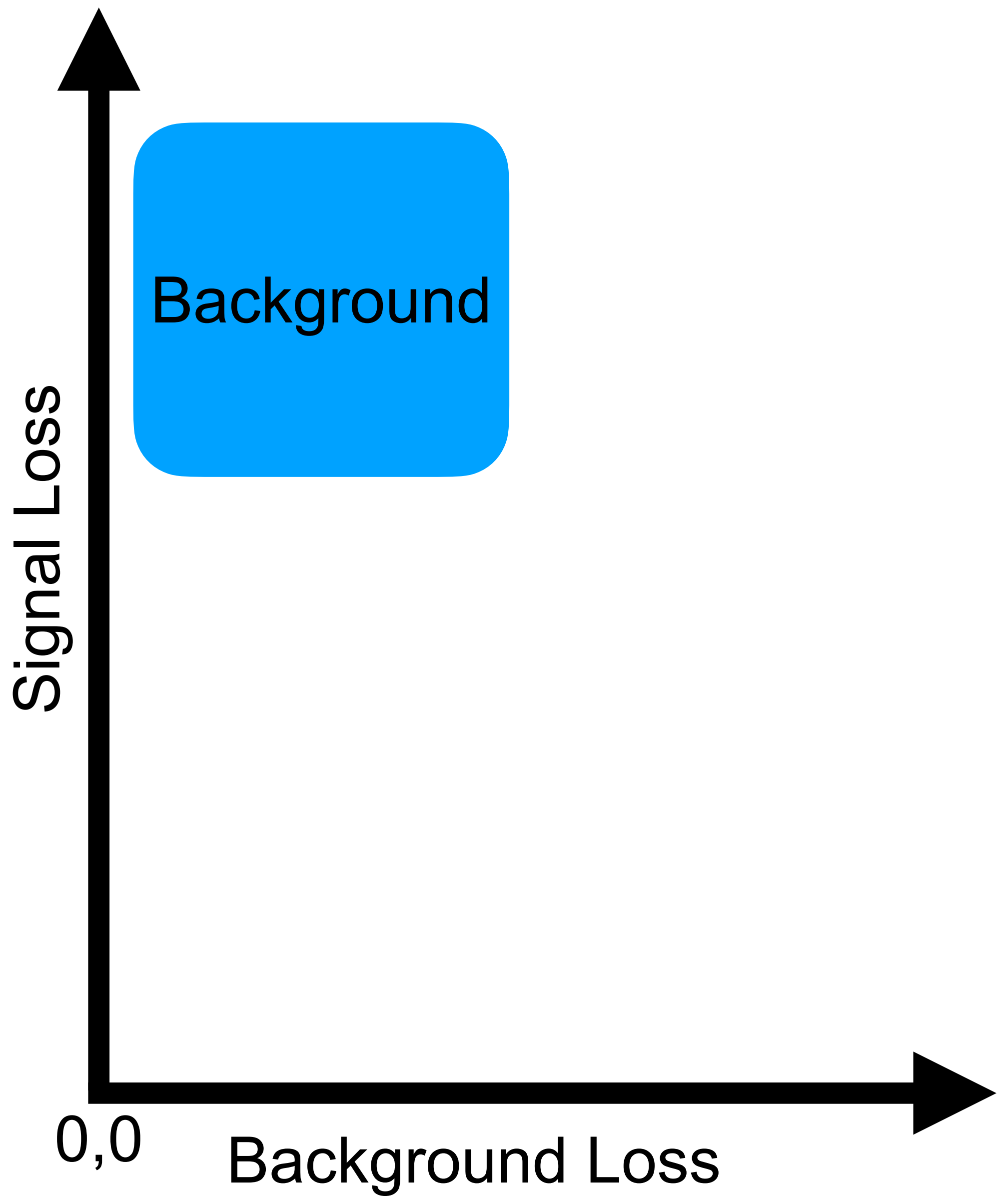


Use Normalizing Flow Autoencoders

arxiv/2011.03550

QUasi Anomalous Knowledge

Normalizing
Flow
Trained
On signals

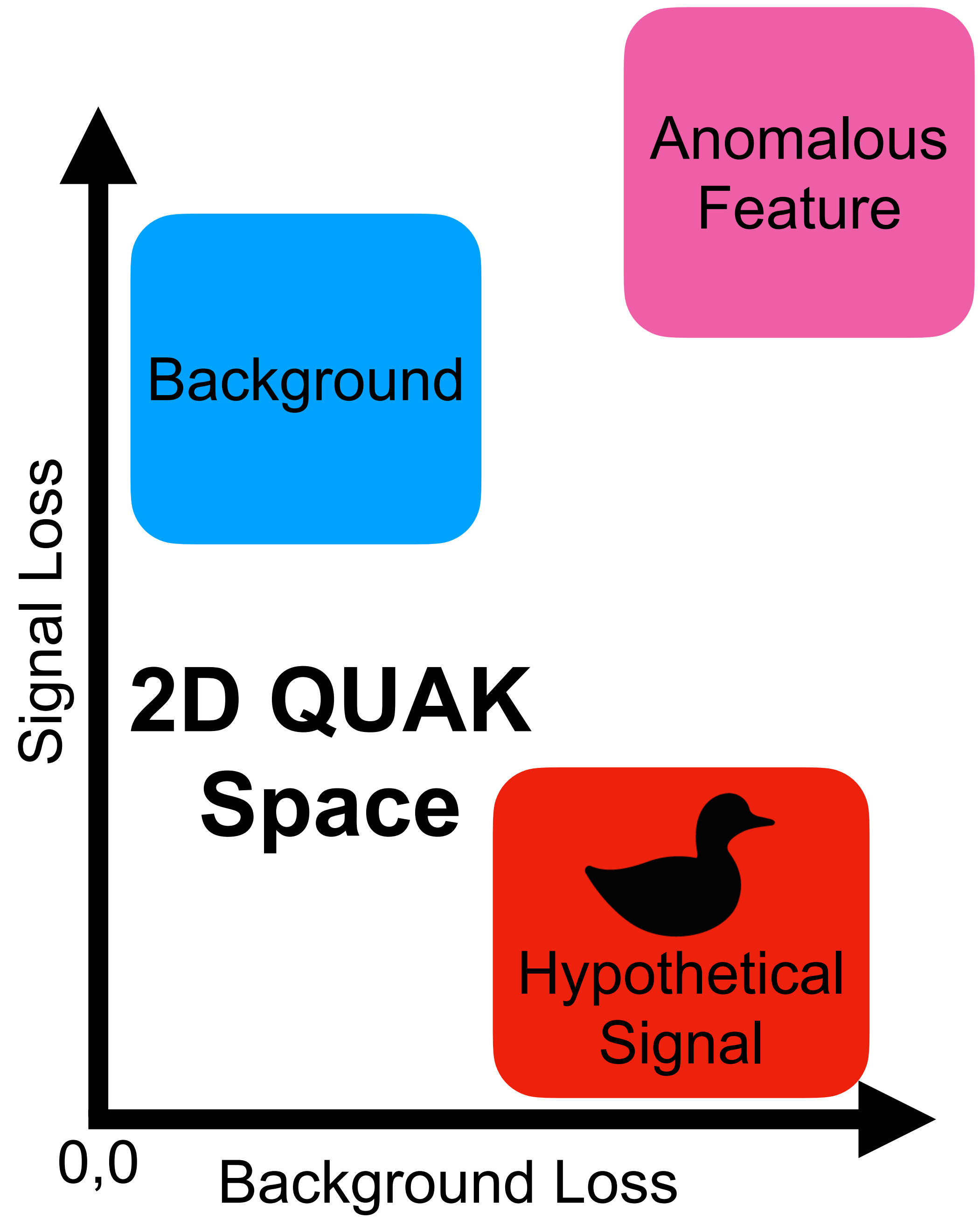


Normalizing Flow Trained On Backgrounds

arxiv/2011.03550

QUasi Anomalous Knowledge

Normalizing
Flow
Trained
On signals

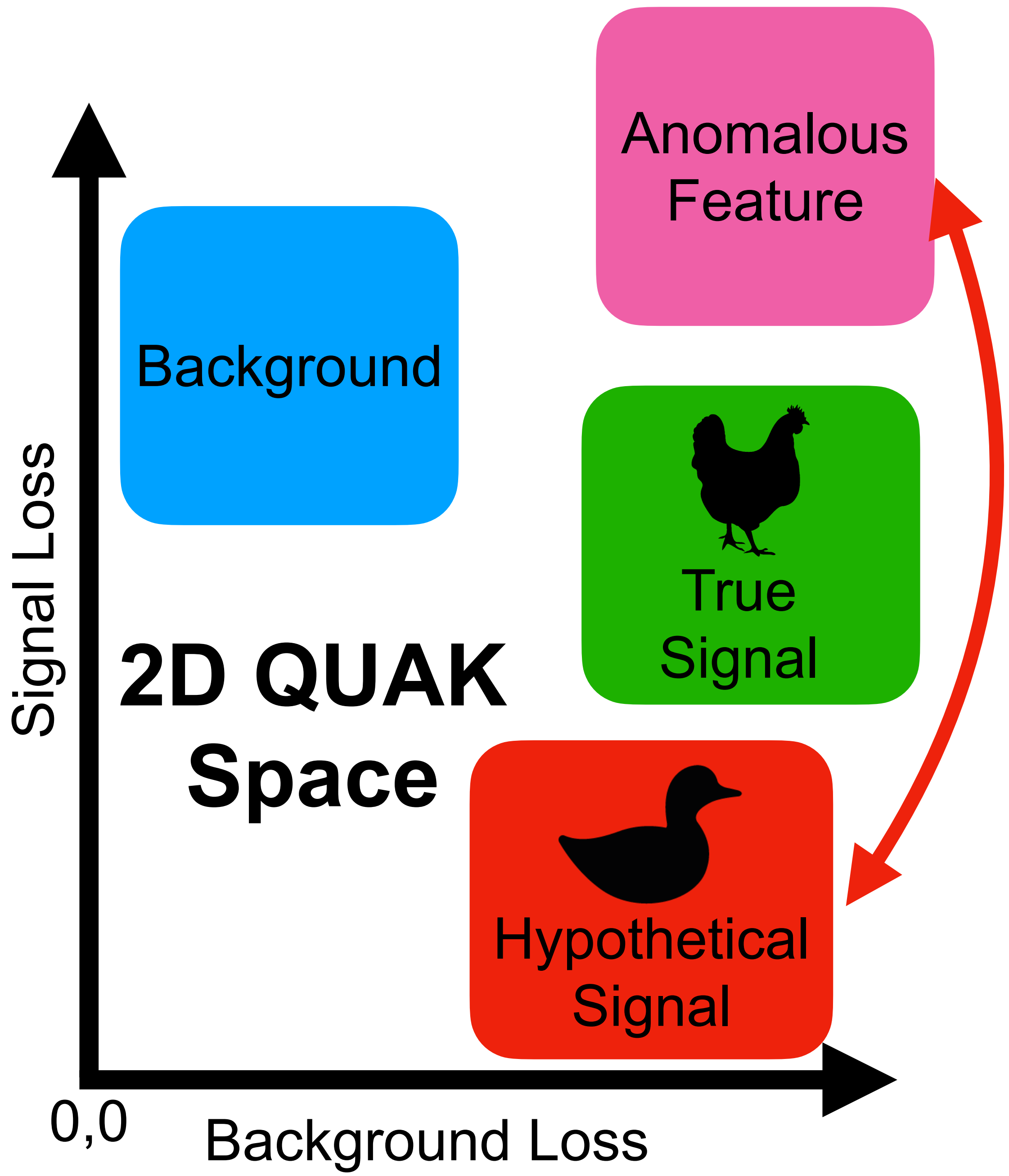


Normalizing Flow Trained On Backgrounds

arxiv/2011.03550

QUasi Anomalous Knowledge

Normalizing
Flow
Trained
On signals



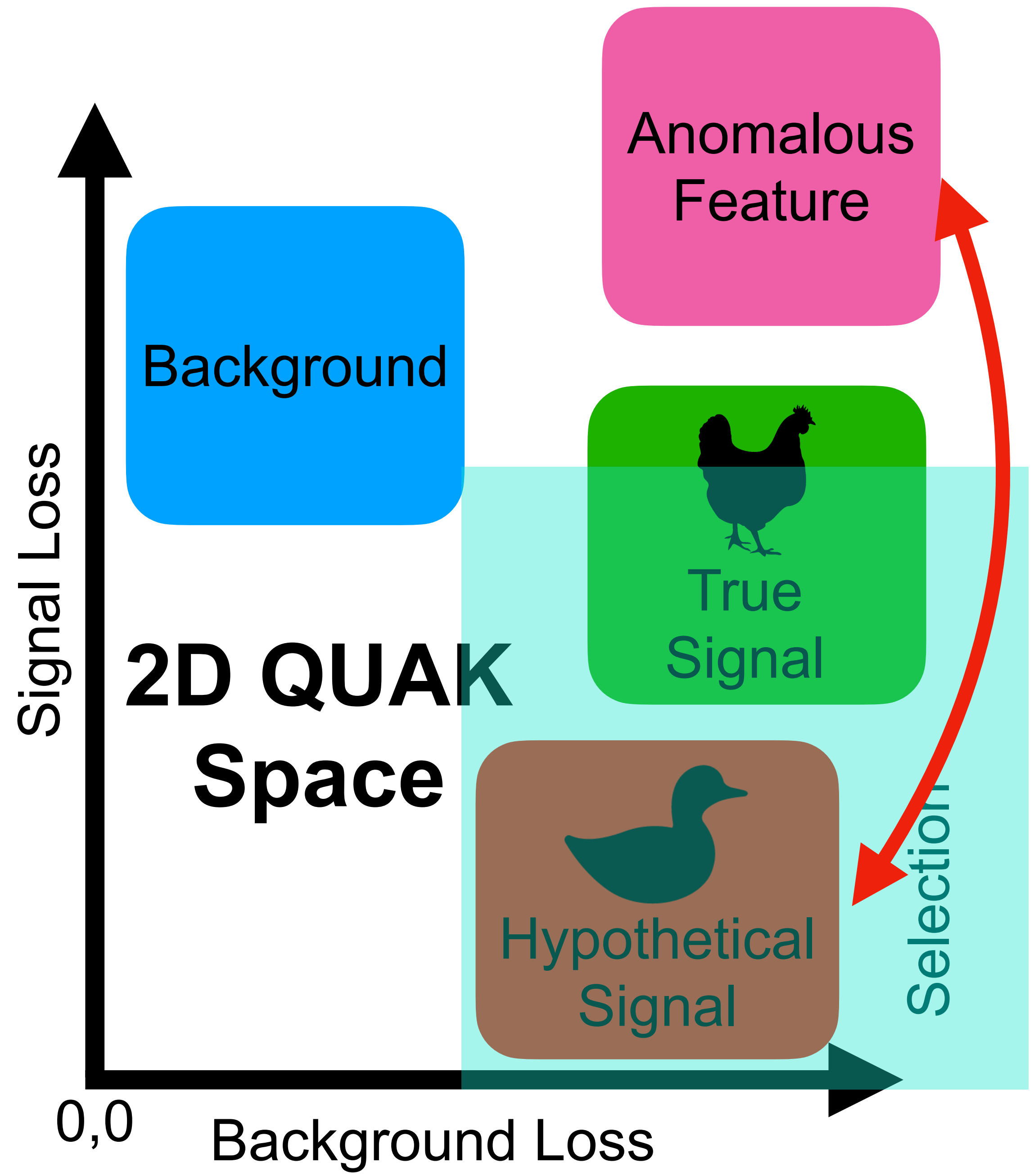
Adding (incorrect)
Signals splits
anomalous signals
From other features

Normalizing Flow Trained On Backgrounds

arxiv/2011.03550

QUasi Anomalous Knowledge

Normalizing
Flow
Trained
On signals



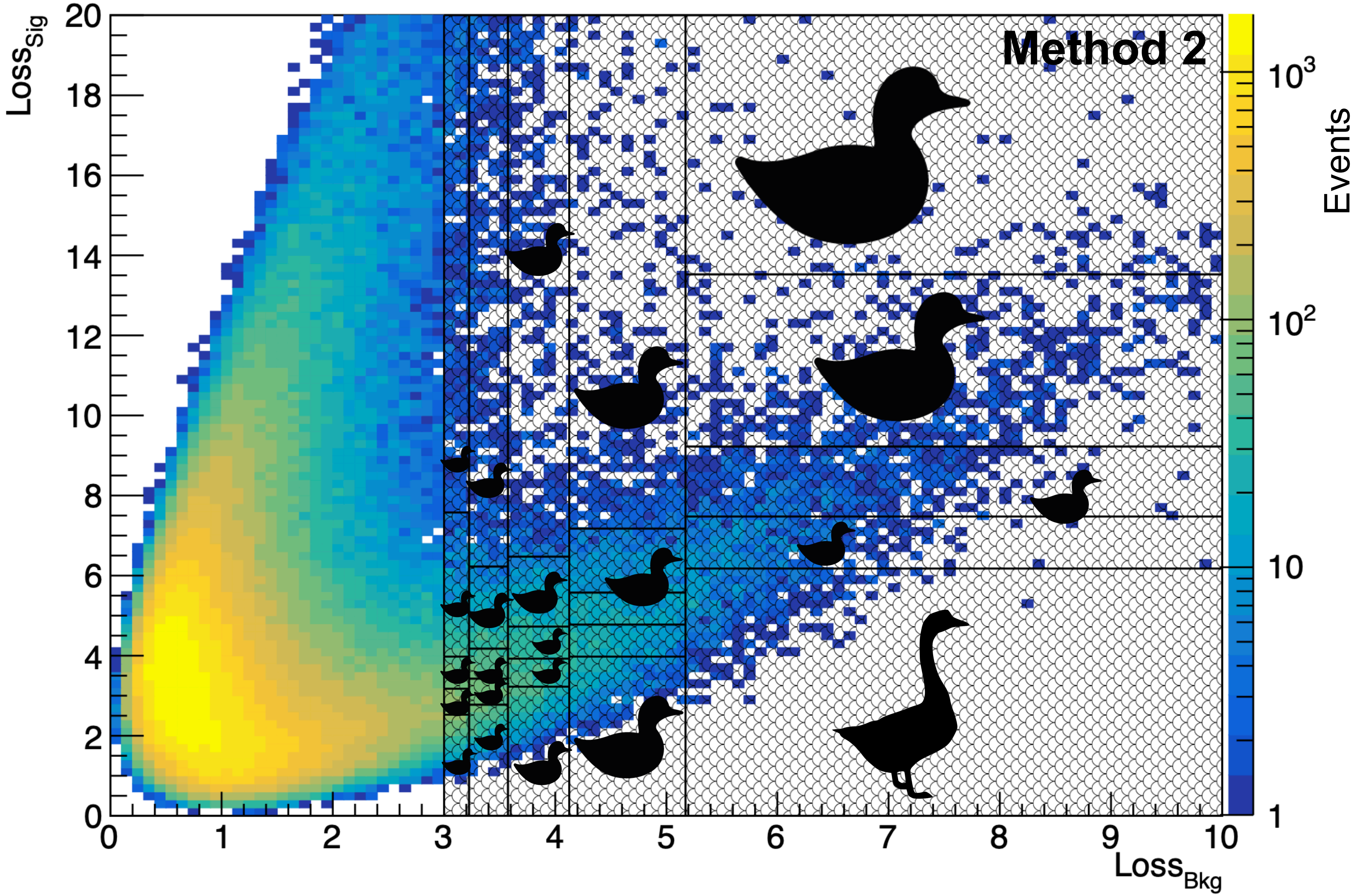
Adding (incorrect)
Signals splits
anomalous signals
From other features

Normalizing Flow Trained On Backgrounds

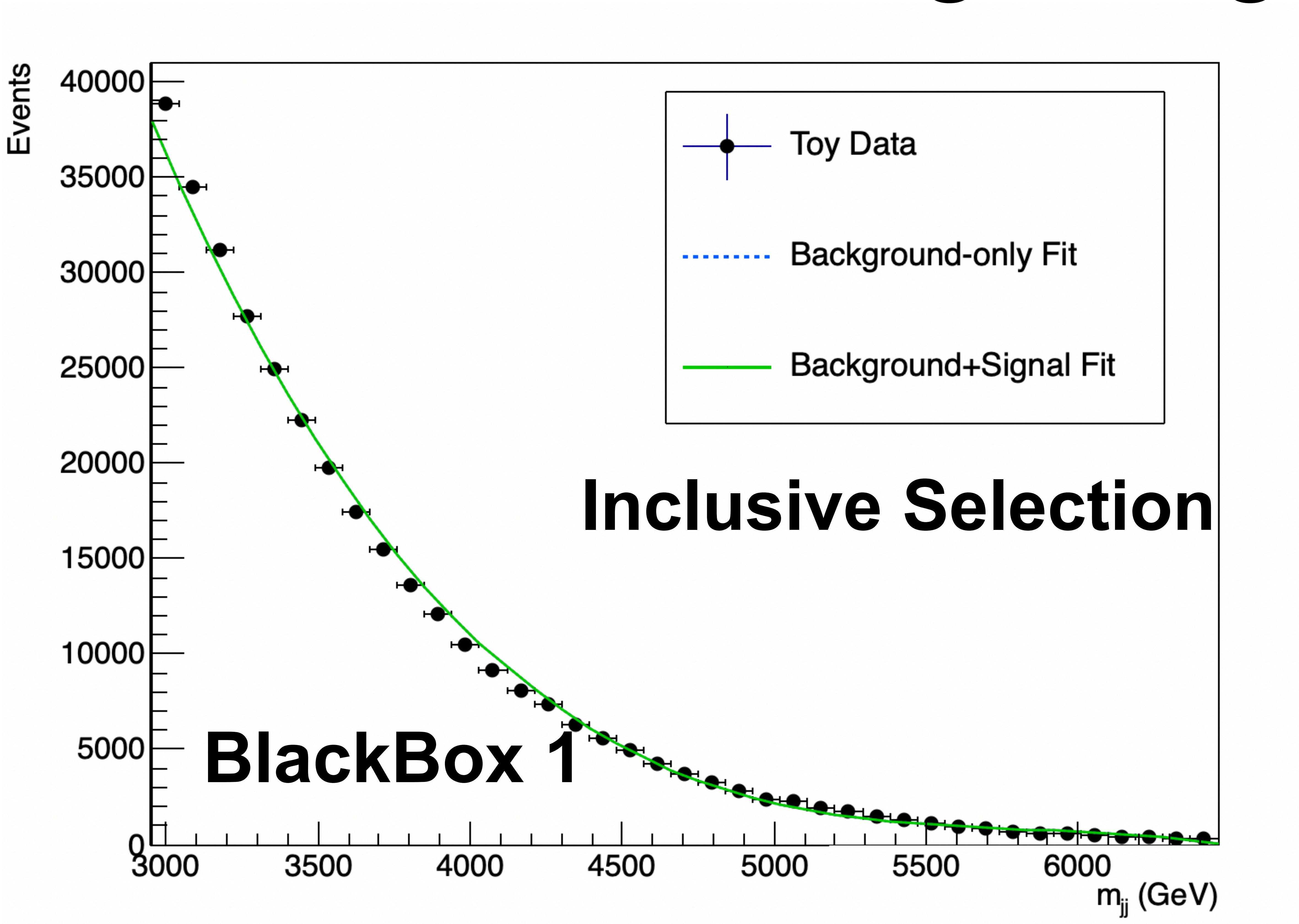
arxiv/2011.03550

Duck Duck Goose!

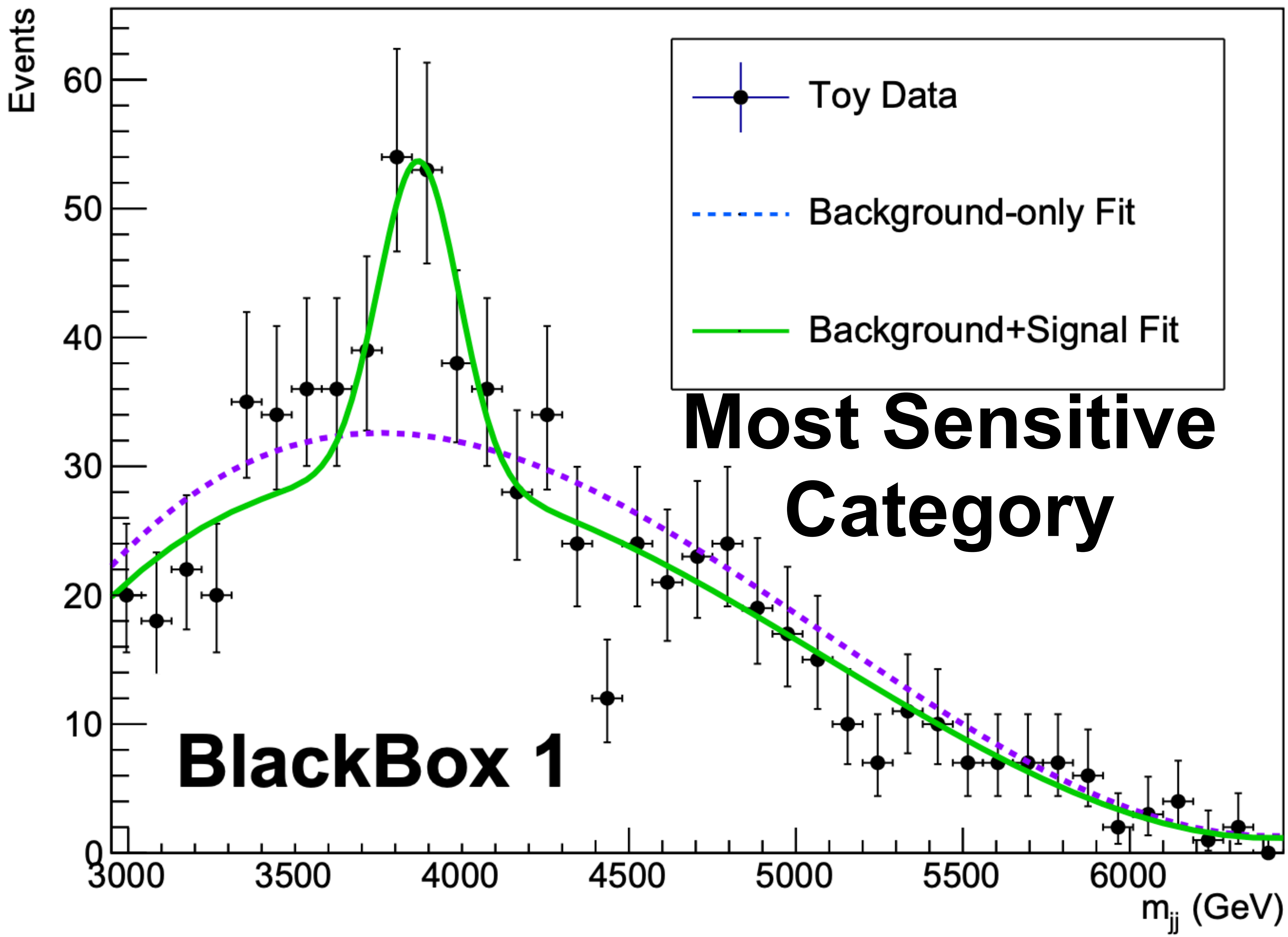
Search all of the regions **one big simultaneous fit**



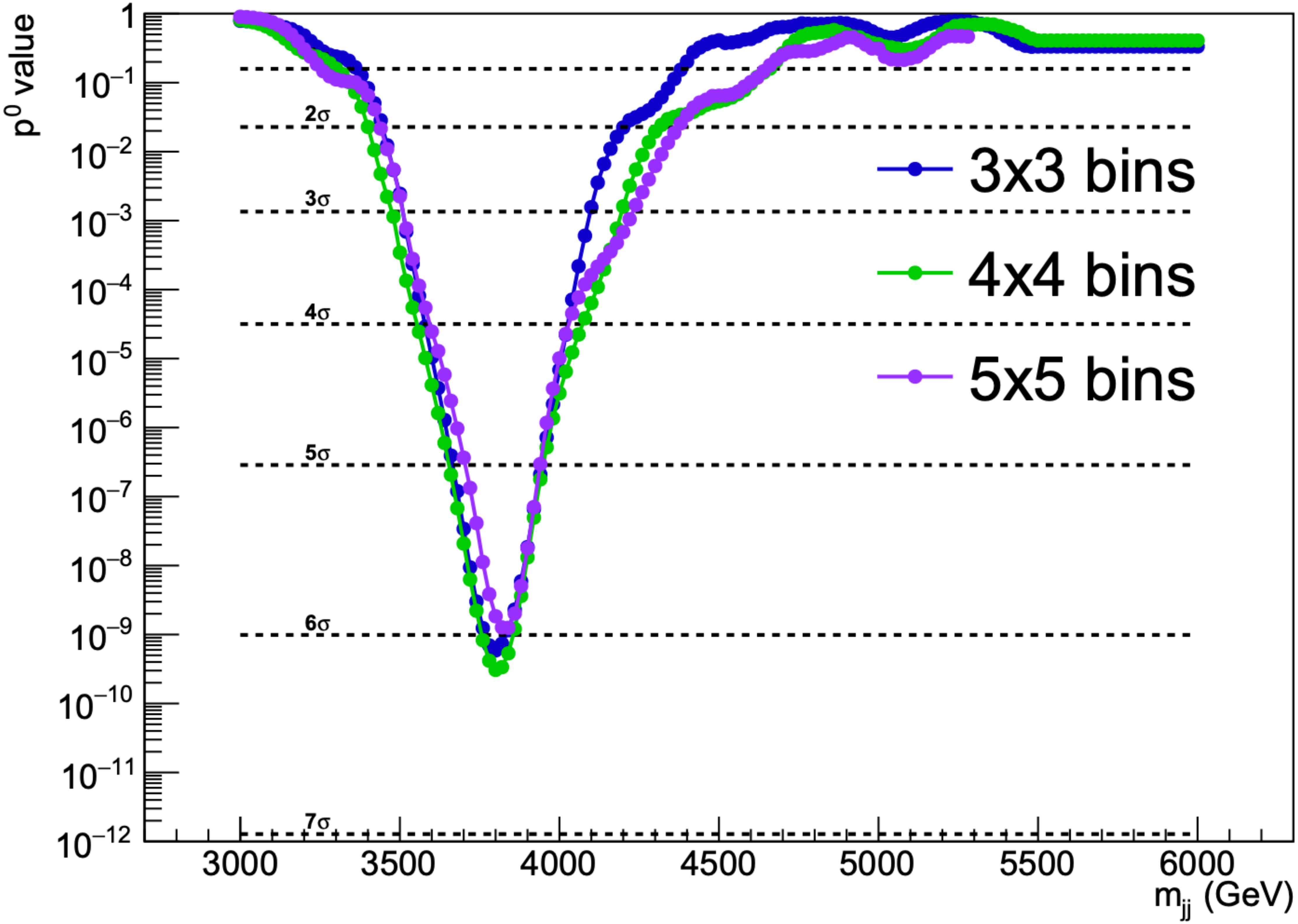
Seeing a Signal



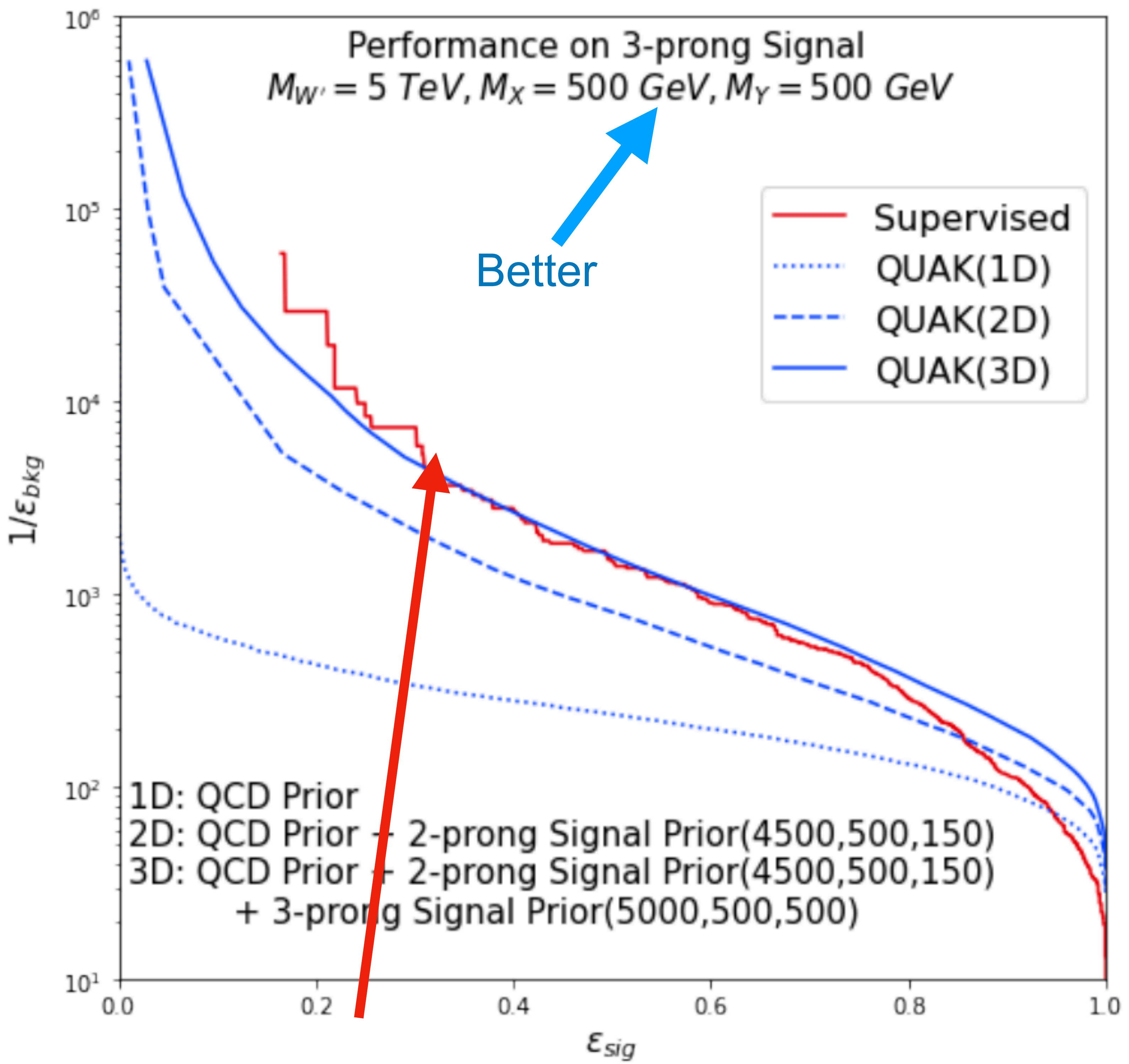
Seeing a Signal



Applying to Anomaly



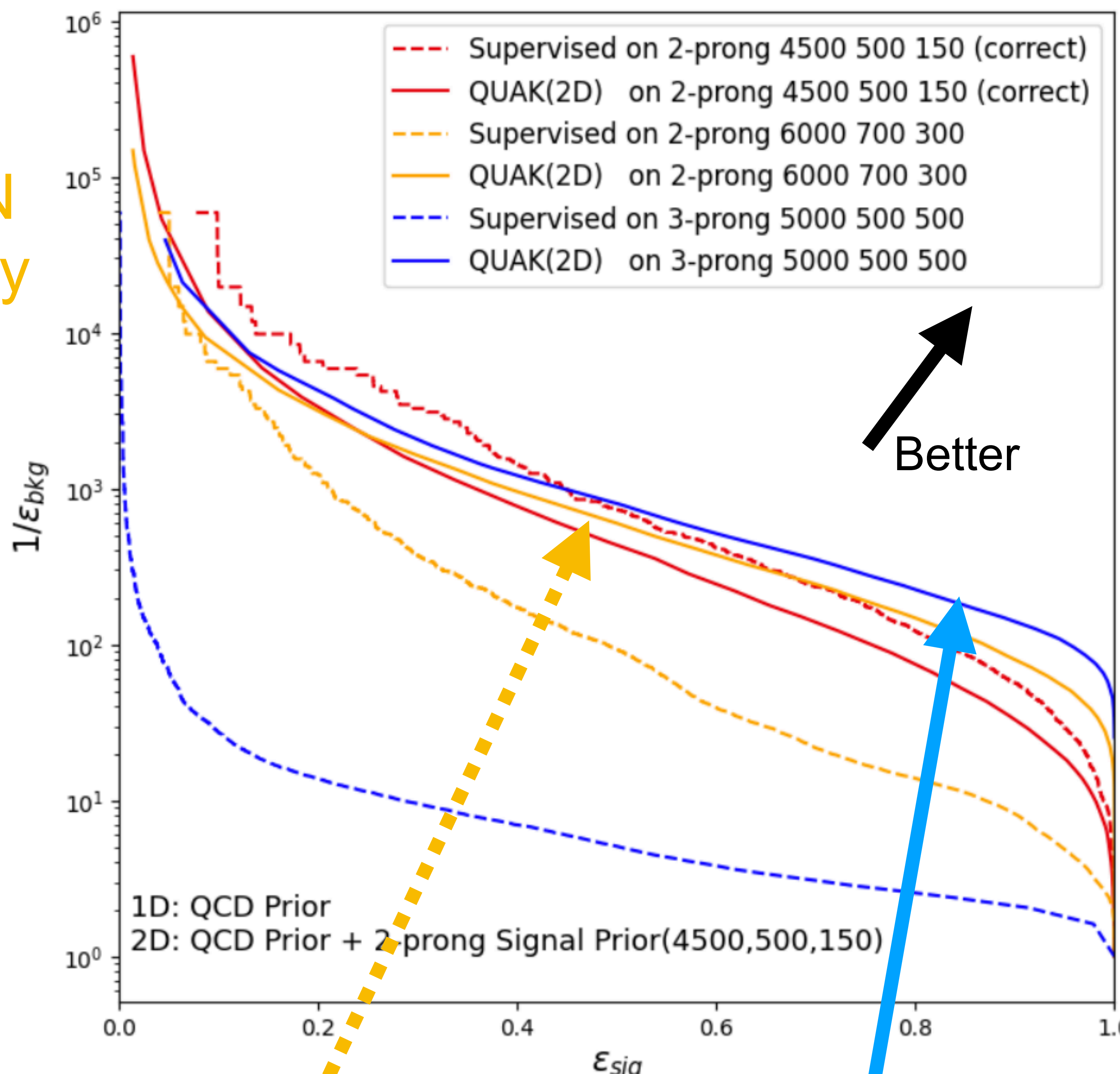
How Close to Optimal?



QUAK can outperform a supervised network
When signals are the same

How Close to Optimal?

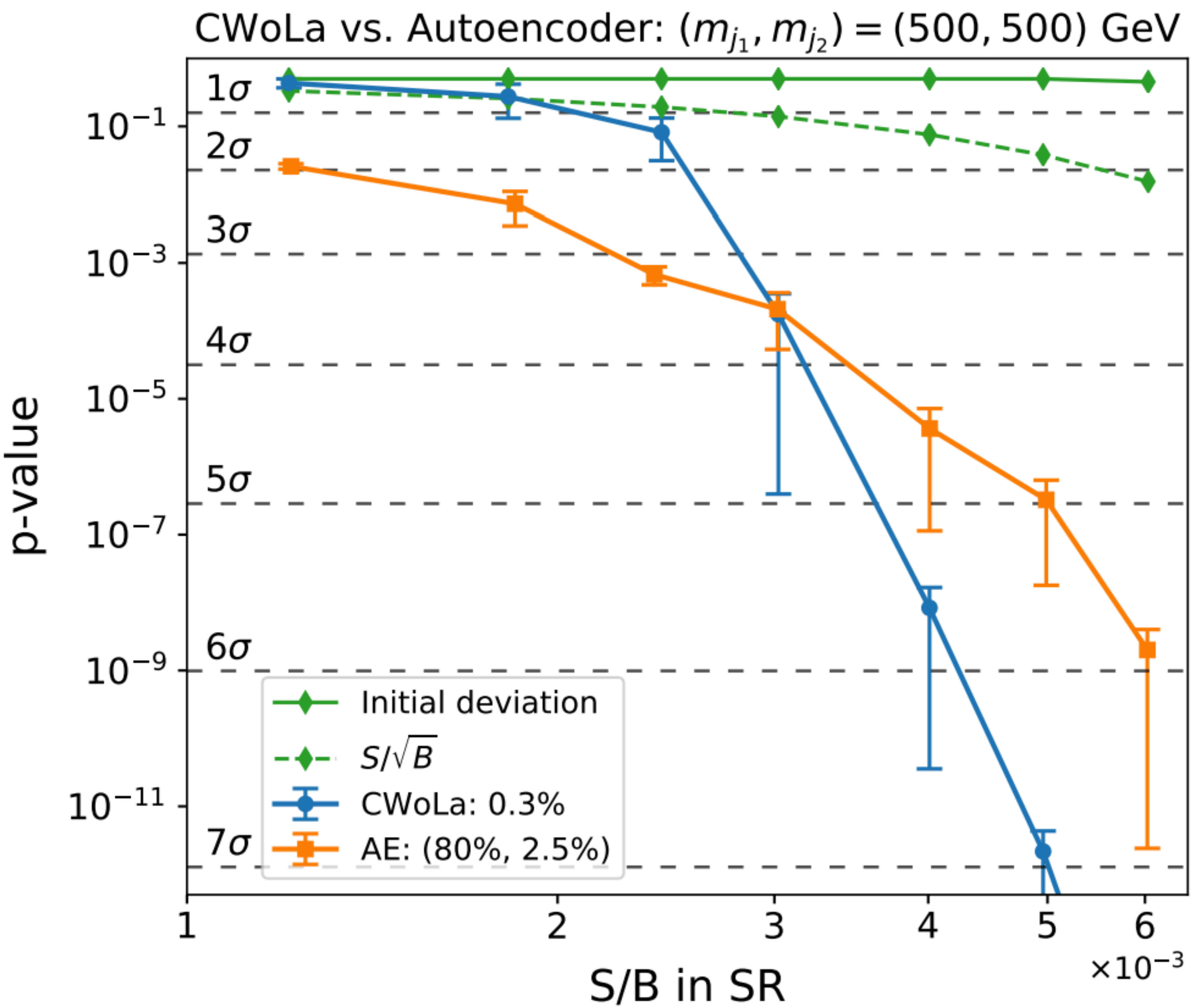
Relies on NN self-assembly to build a continuous space



One Supervised Network

One QUAK Network

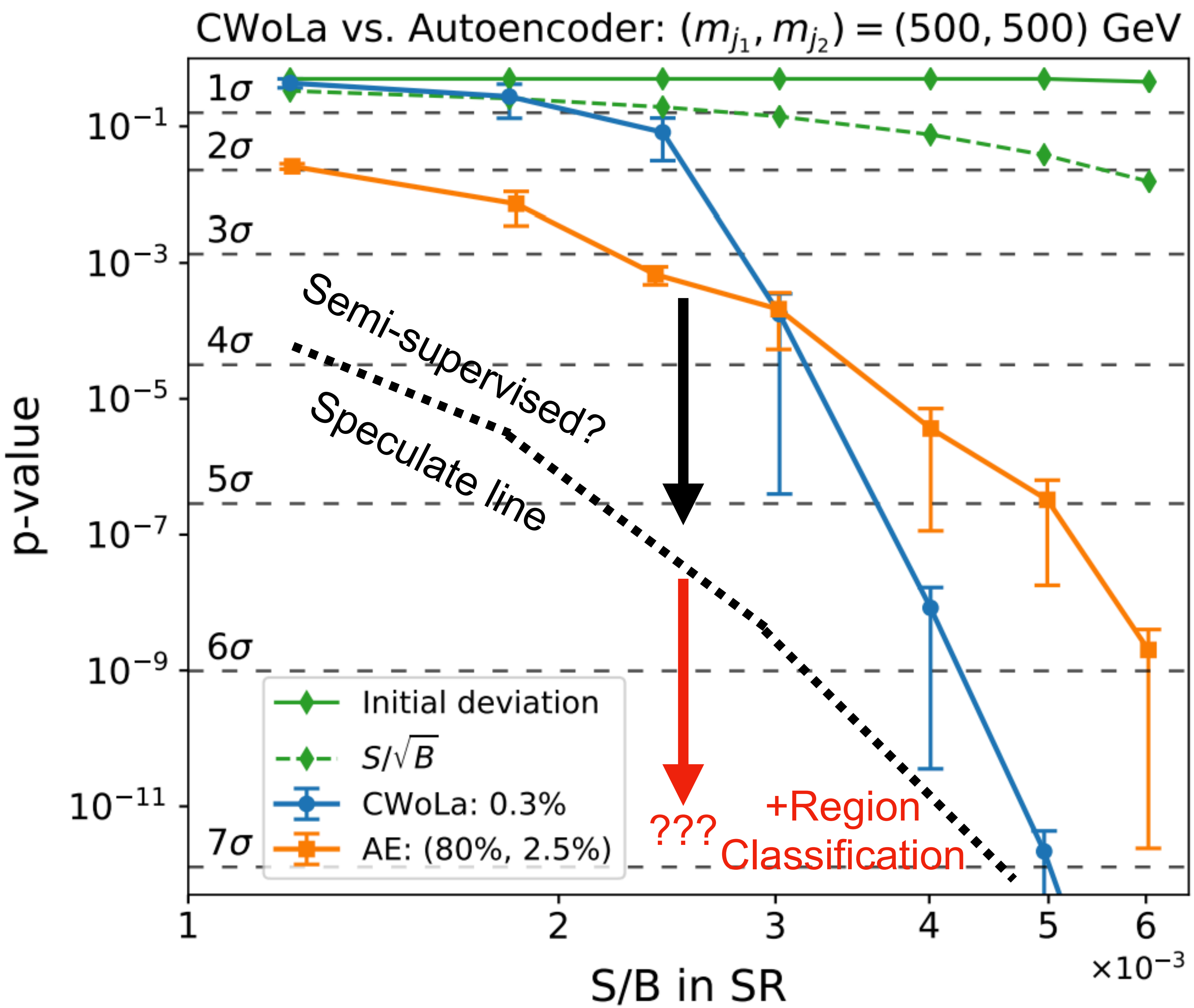
Performance Observations



Normalizing Flow approaches stood out

So did Observable based encoding(not sure why)

Performance Observations



Normalizing Flow approaches stood out

So did Observable based encoding(not sure why)

What will the future be?

- Deep learning is helping us to look at things in finer detail
  - It lets us go deeper and make sense of things

Did we find all the Higgs bosons in there?

Towards The Future

What are all the hidden signals in there?

Deep Learning can help Elucidate

- AI is helping us to look at things in finer detail
  - It lets us go deeper and make sense of things

Did we find all the Higgs bosons in there?

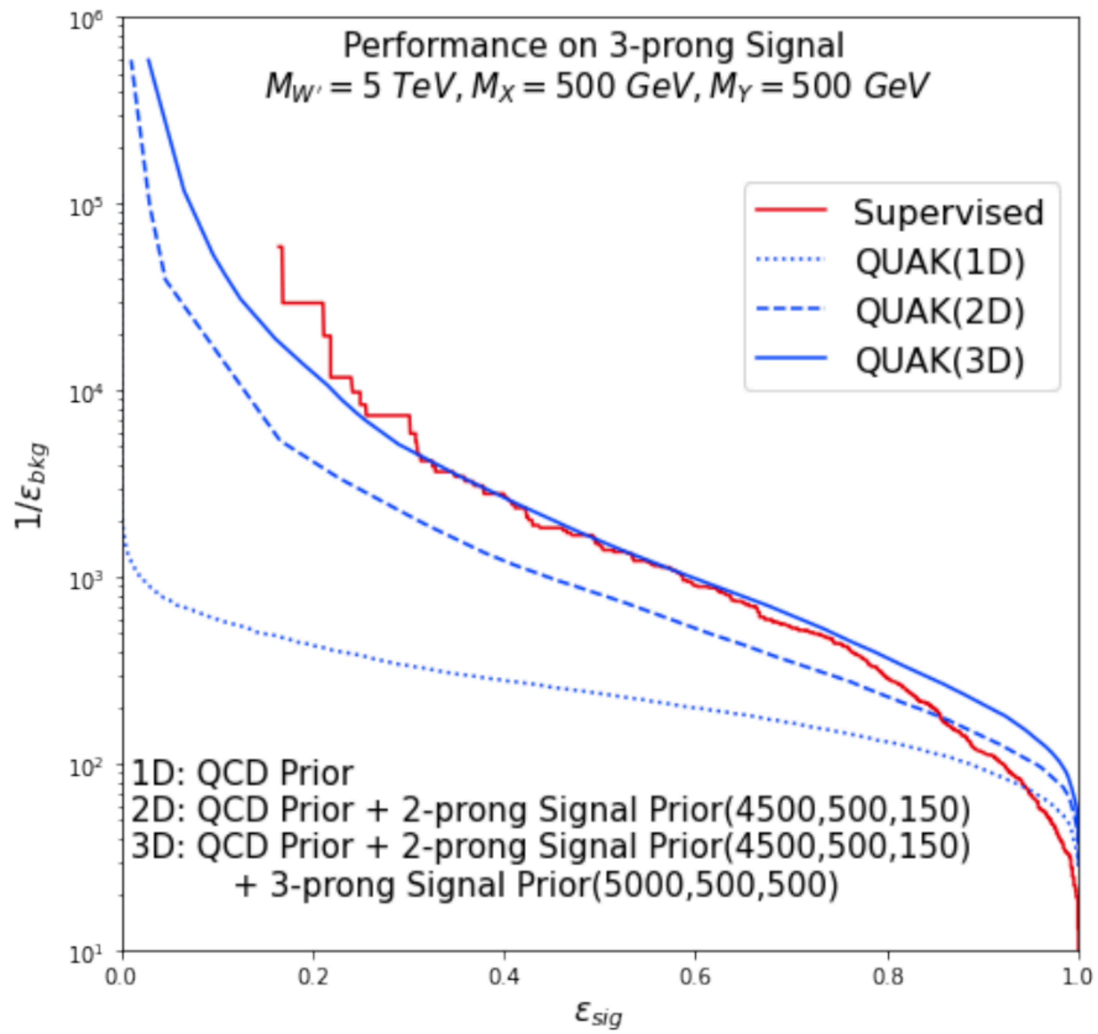
Towards The Future

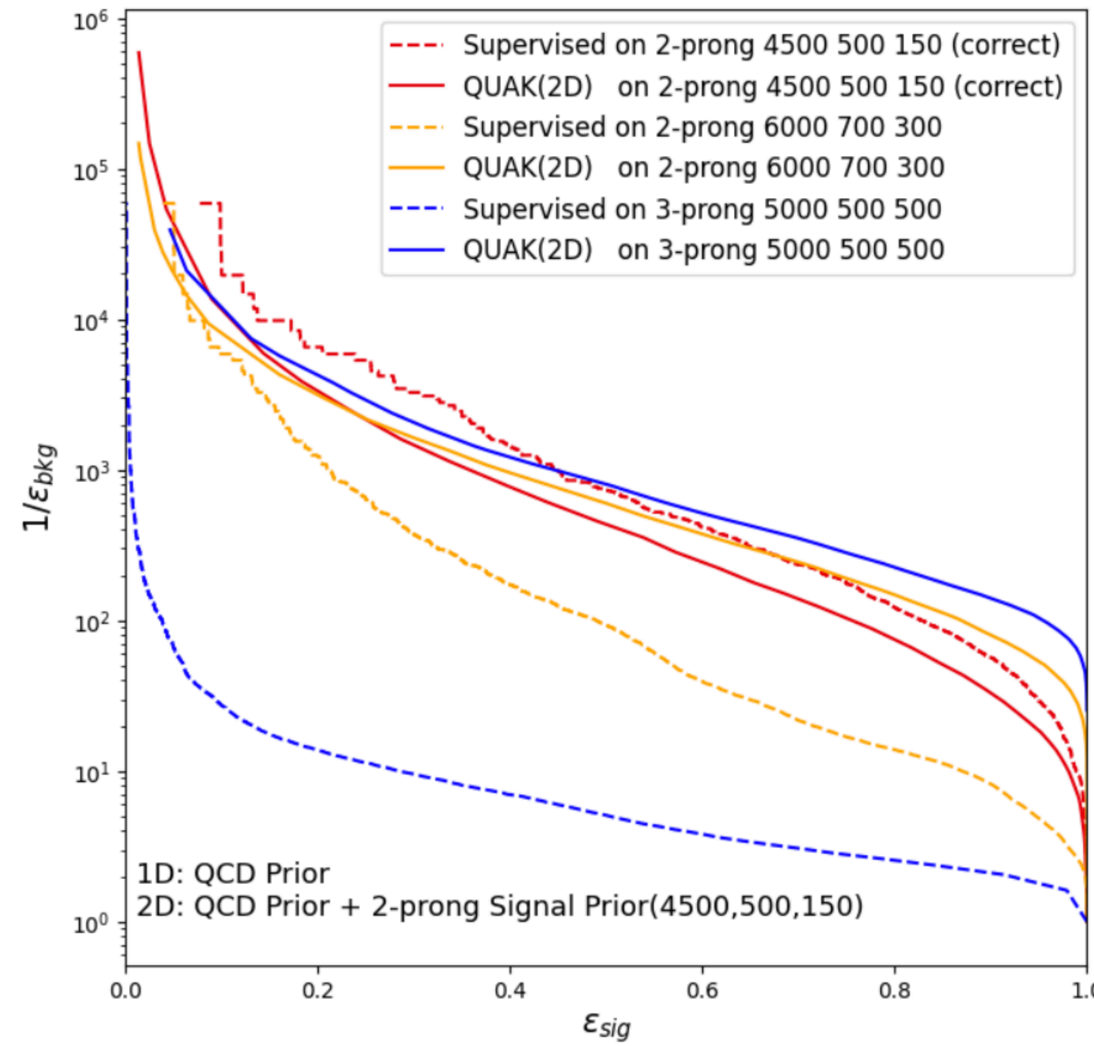
What are all the hidden signals in there?

Perhaps there is a hidden Discovery

Thanks to the organizers for inviting me!

QUAK





QUAK approaches or beats supervised NNs when signal is similar

Has been observed in literature with similar type of constructions
Relies on NN self-assembly to build a continuous space
Space starts to classify regions of algorithms

https://arxiv.org/pdf/2011.03550.pdf

Overview of this talk

- Strategy for this talk
  - I will do a broad overview of ideas about deep learnig
  - The idea is to discuss various general trends
  - Would like to tie this in to broad vision of AI
  - Mostly this will showcase work from my group
    - Don’t consider this a full survey of methods
    - Even though title says LHC I will go beyond at times

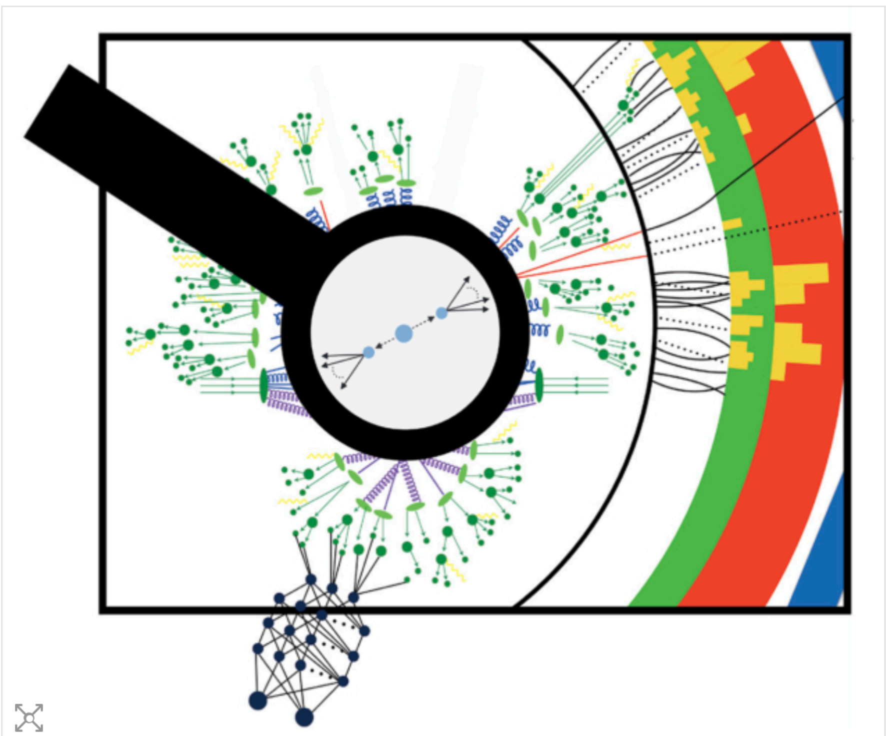
Two Anomaly Challenges

LHC Olympics 2020

arxiv/2101.08320

David Shih, Ben Nachman, Gregor Kasieczka

Dark Machines



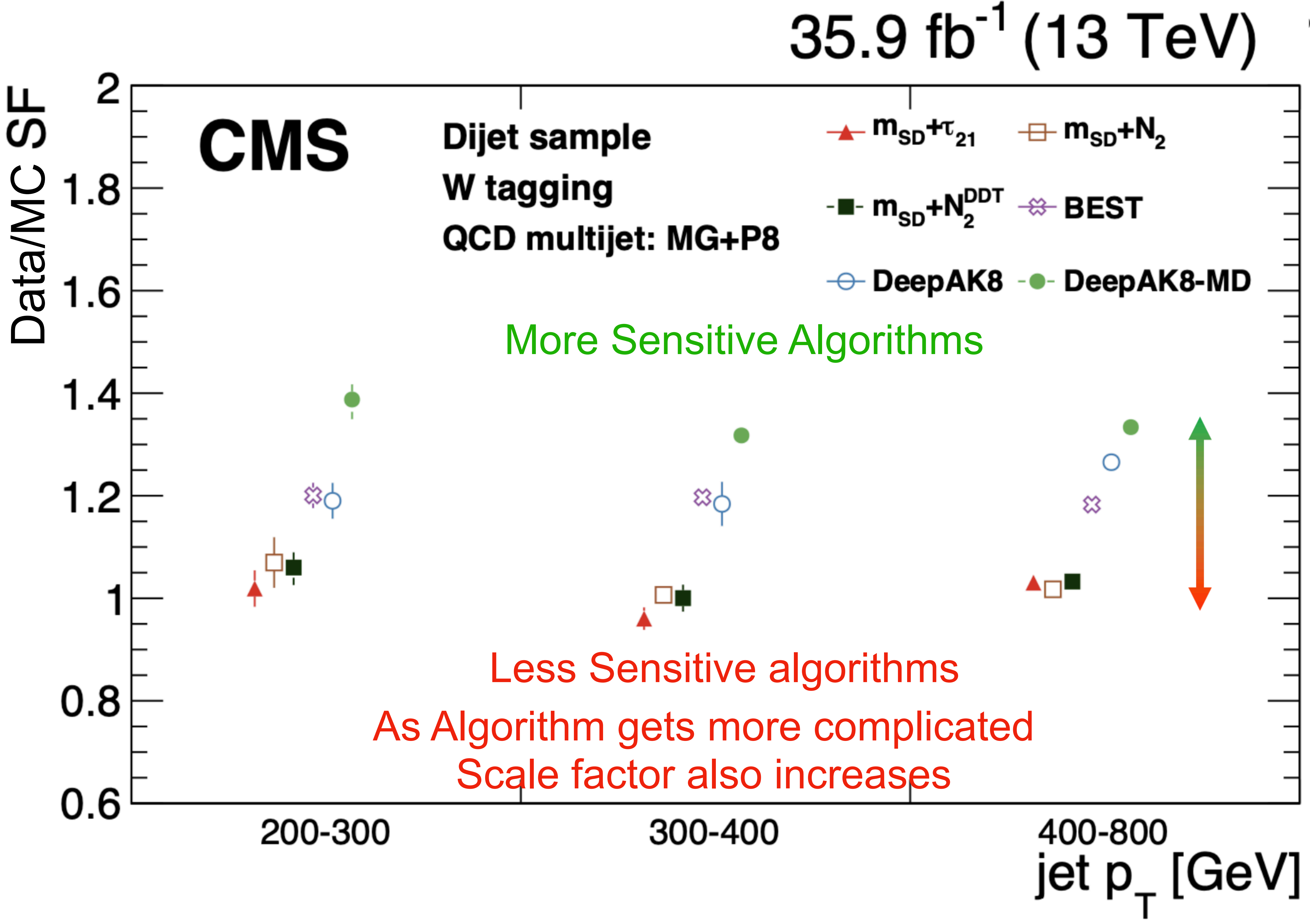
arxiv/2105.14027

- LHC Olympics focused on find a single di-jet resonant model
- DarkMachines focused on searching for a broad range of models

LHC Olympics 2020

- Over the past year there were two competitions
- In each setup a signal/signals wer hidden in pseudo data
  - The challenge was to "Find the hidden signal"
  - Emulate a realistic analysis as much as possible
  - Challenge : use deep learning to find an anomaly
- A number of different strategies are used for this approach
  - We will review the core concepts of these strategies

hep-ph/2101.08320



Next generation of taggers would benefit from **next generation of MC**

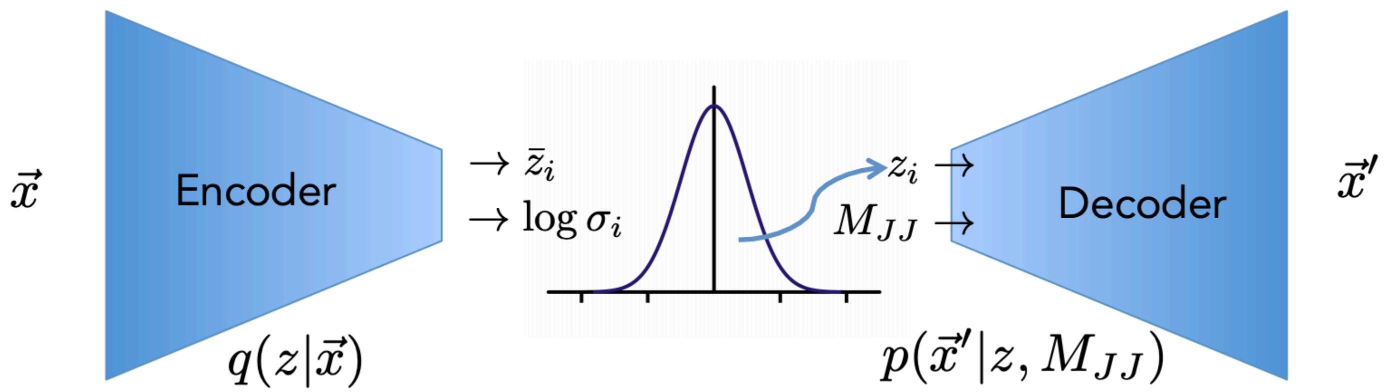
Training on Data

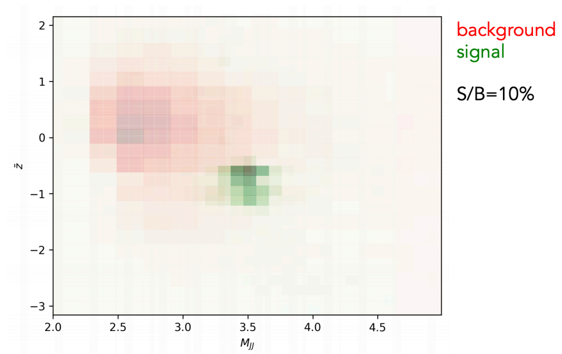
- Generally with anomaly approaches
    - There has been an emphasis to train on data
- Training on data simplifies our ability to process data
    - No need to correct for simulation/data disagreements
    - Regions where data/simulation don't agree can be probed
    - No fancy methods to probe these regions w/complicated fits
- Training on data throws away some interpretability of result
    - Not clear what features may drive an access

BuHuLaSpa

Inputs: High Level Features (Nsubjettiness/Jet masses/…)

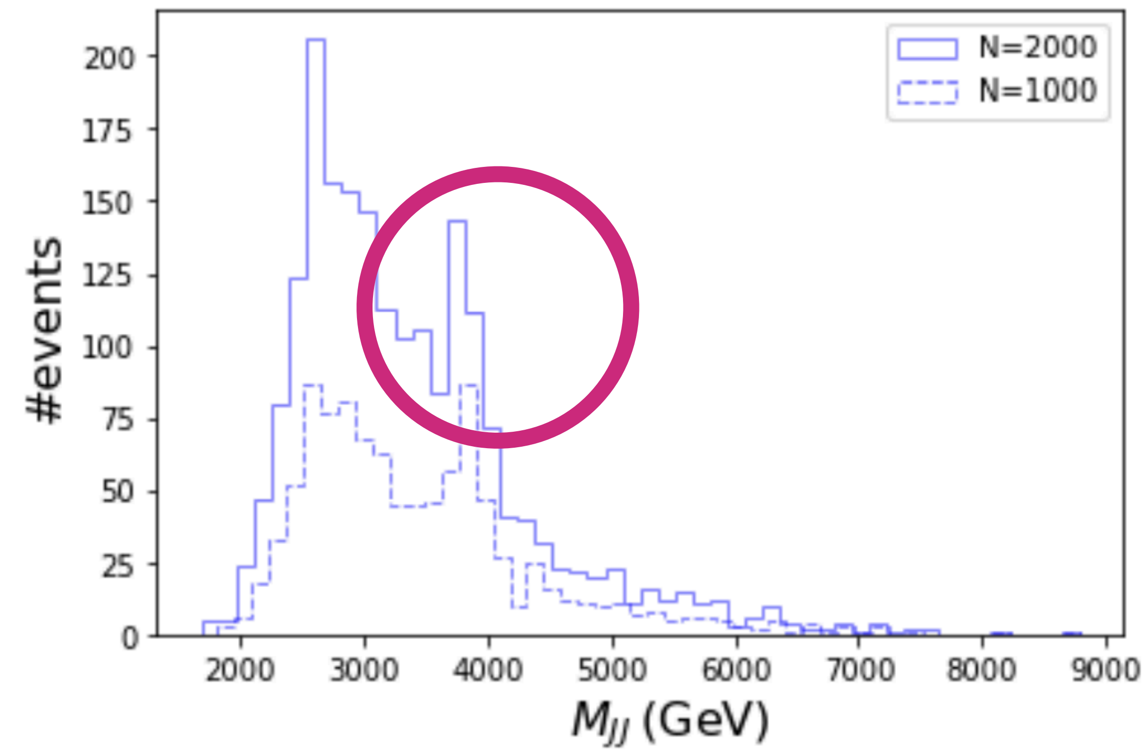
- Bump hunting in the latent space





Encoder output

BB1 Dataset



Autoencoder with 1D latent space
Latent space forced to be decorrelated with mass

$$\mathcal{L} = -D_{\mathrm{KL}}(q_\phi(\vec{z}_i|\vec{x}_i)|p(\vec{z}_i)) + \beta_{\mathrm{reco}} \log p_\theta(\vec{x}_i|\vec{z}_i)$$

Signal Extraction : None

Take Away:Training is critical to ensure good performance

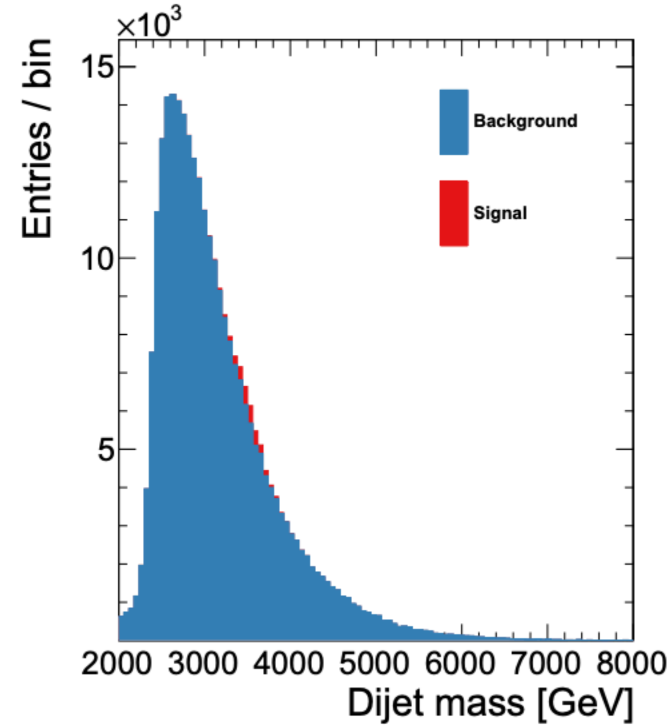
UCluster

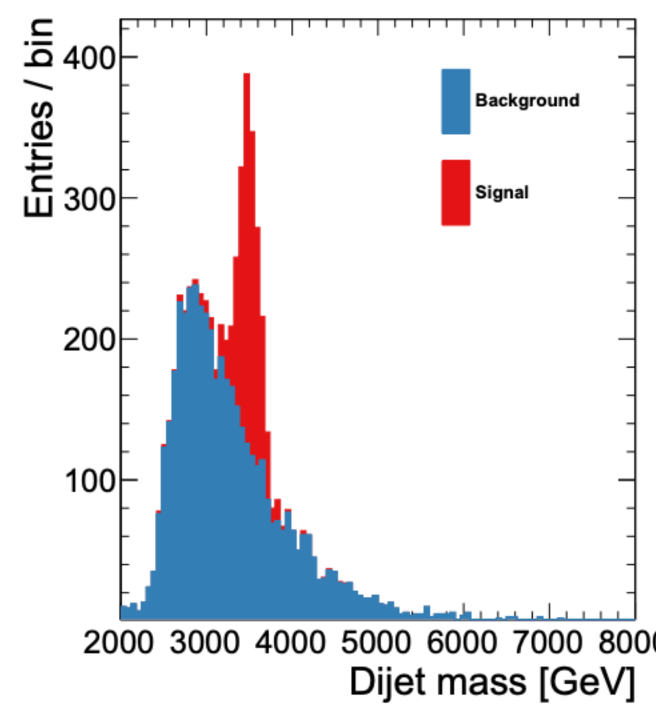
Inputs: Particle Objects

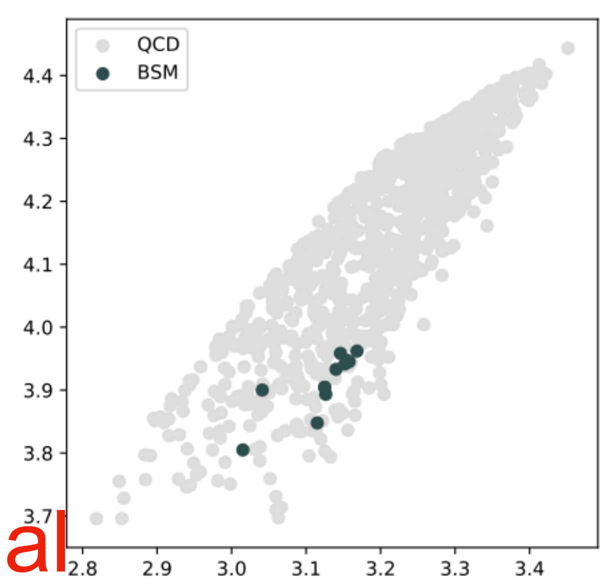
R&D Dataset

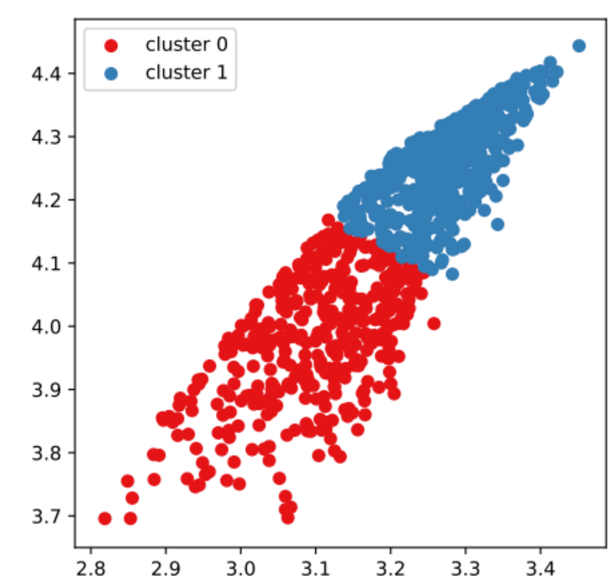
Train a supervised network for jet classification

Cluster in the latent space
Scan clusters for anomaly









Signal Extraction : No signal

Take Away: Hard with small signal

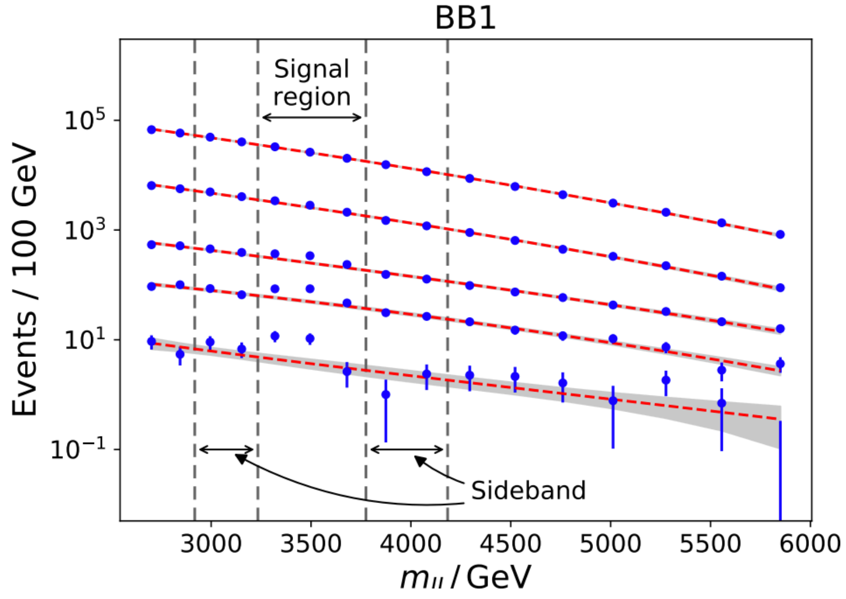
https://arxiv.org/abs/2010.07106

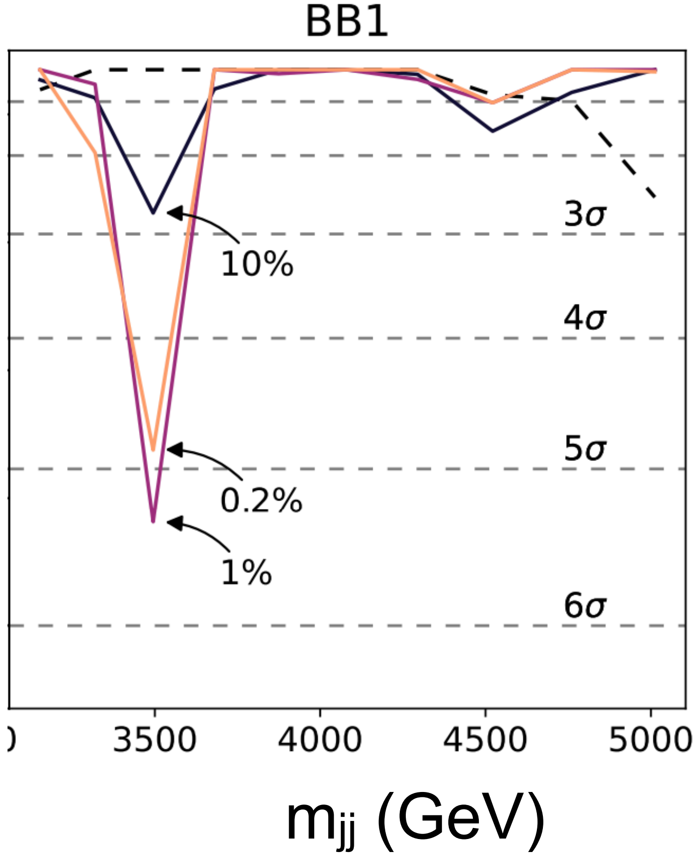
CWOLA

Inputs: High level features

- CWOLA modified from original paper
  - Mass inputs dimensionless

BB1 Dataset





Signal Extraction : Bump fit(5σ)

Take Away: Works but needed to correct dimension

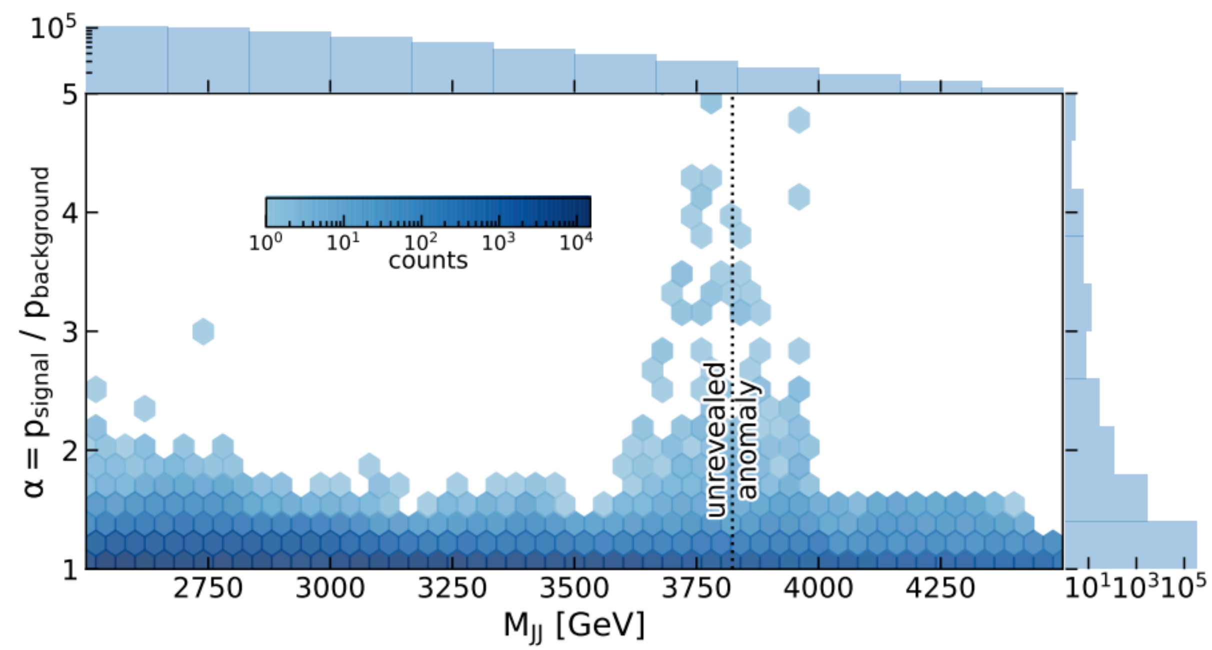
GIS(CWOLA+NF)

Inputs: High level features

GIS normalizing flows trained conditional on the mass distribution

Scan mass window (250 GeV)
Compute likelihood ratio (below)

BB1 Dataset



Large and significant signal

Signal mass | Mass Side band

$p(x \,|\, x \in A)$ | $p(x \,|\, x \in B)$

$$R(x|m) = \frac{p_{\mathrm{data}}(x|m)}{p_{\mathrm{background}}(x|m)}$$

Signal Extraction : Note, but large signal

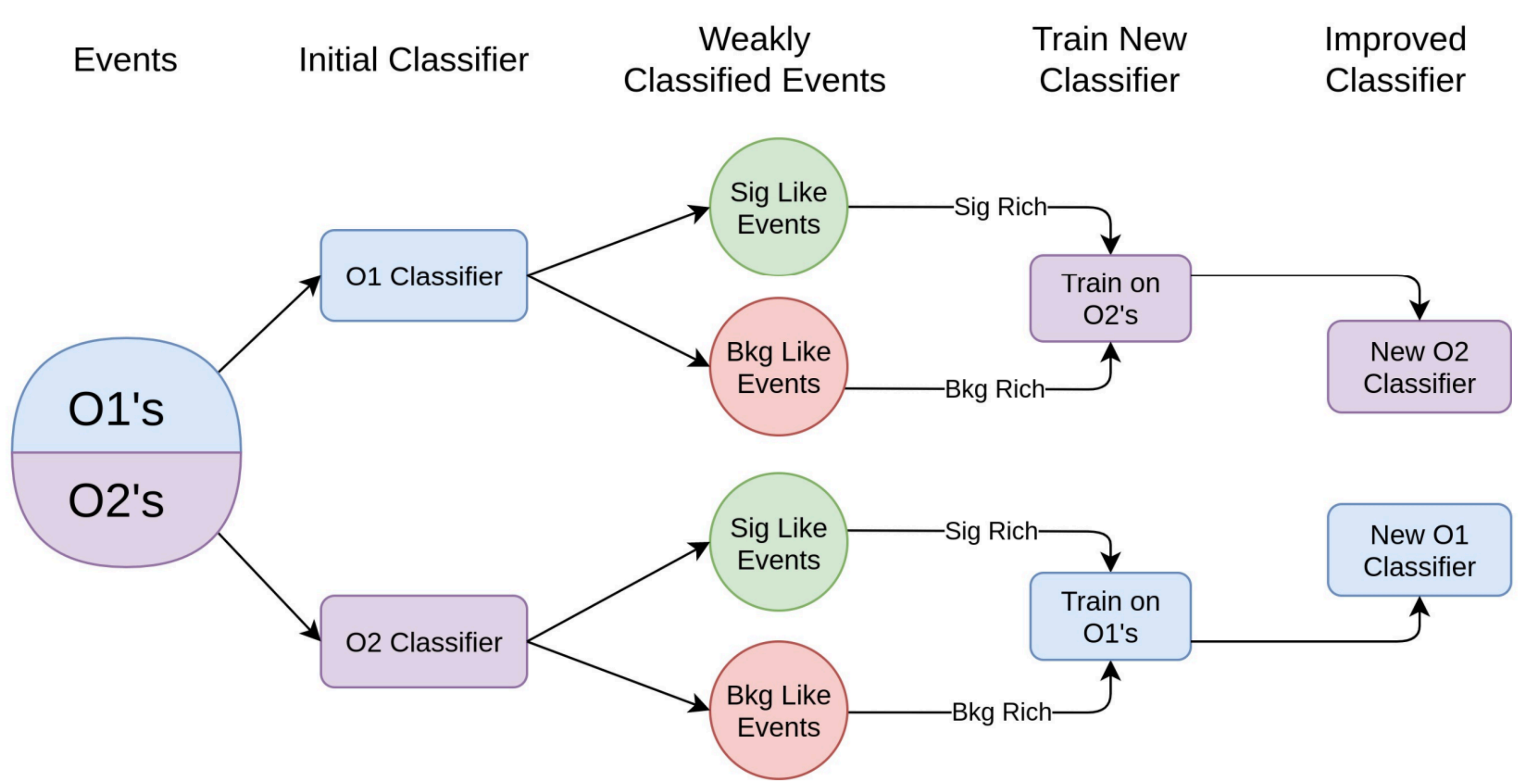
Take Away: Normalizing Flow can help CWOLA style approach

https://arxiv.org/pdf/2001.04990.pdf

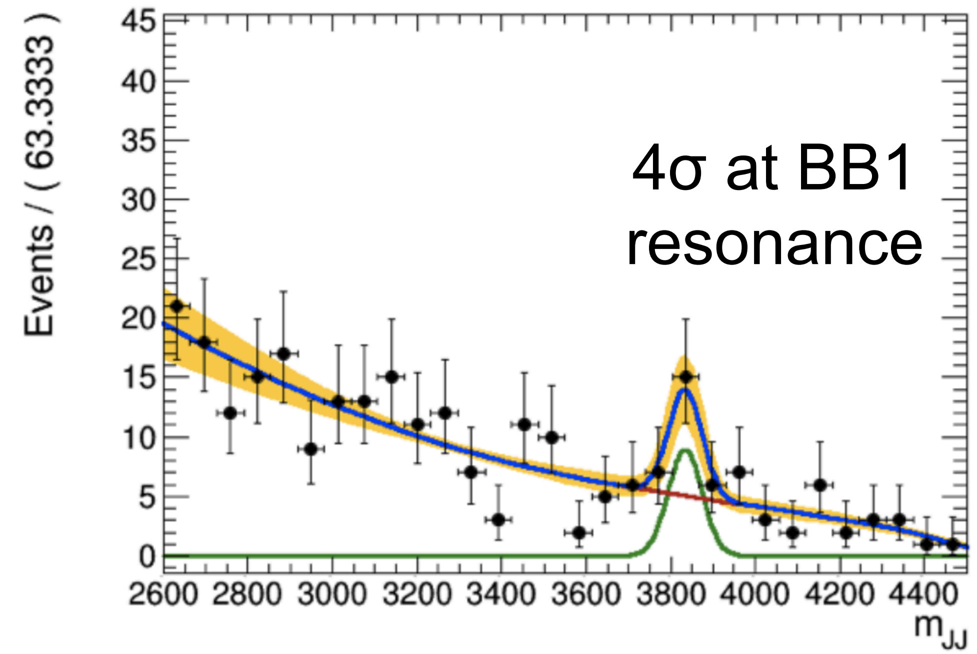
Tag N'Train

Inputs: High level features

Use dijet signature play one jet off the other
Start with an autoencoder on jet to split sample
Run CWOLA on other jet with split sample



BB1 Dataset



Would benefit more from mass decorrelation

Signal Extraction : Bump Fit

Take Away: Avoid mass windows by relying on the different jets

https://arxiv.org/abs/2002.12376

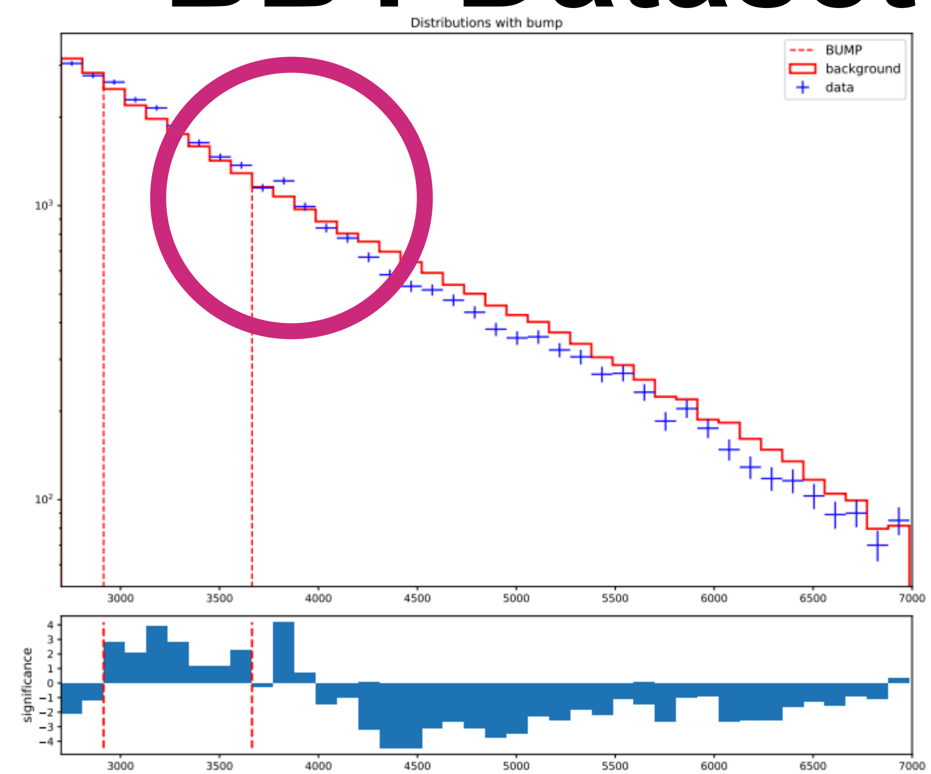
GAN supported AE

Inputs: High Level Features (Nsubjettiness/Jet masses/…)

BB1 Dataset

- Build an auto encoder (AE)
  - Add an GAN to help AE
  - Additionally decorrelate with mass

$$\mathrm{loss}_{\mathrm{AE}} = \mathrm{BC} + \varepsilon \times \mathrm{MED} + \alpha \times \mathrm{DisCo}$$



ıtent space forced to be decorrelated with mass

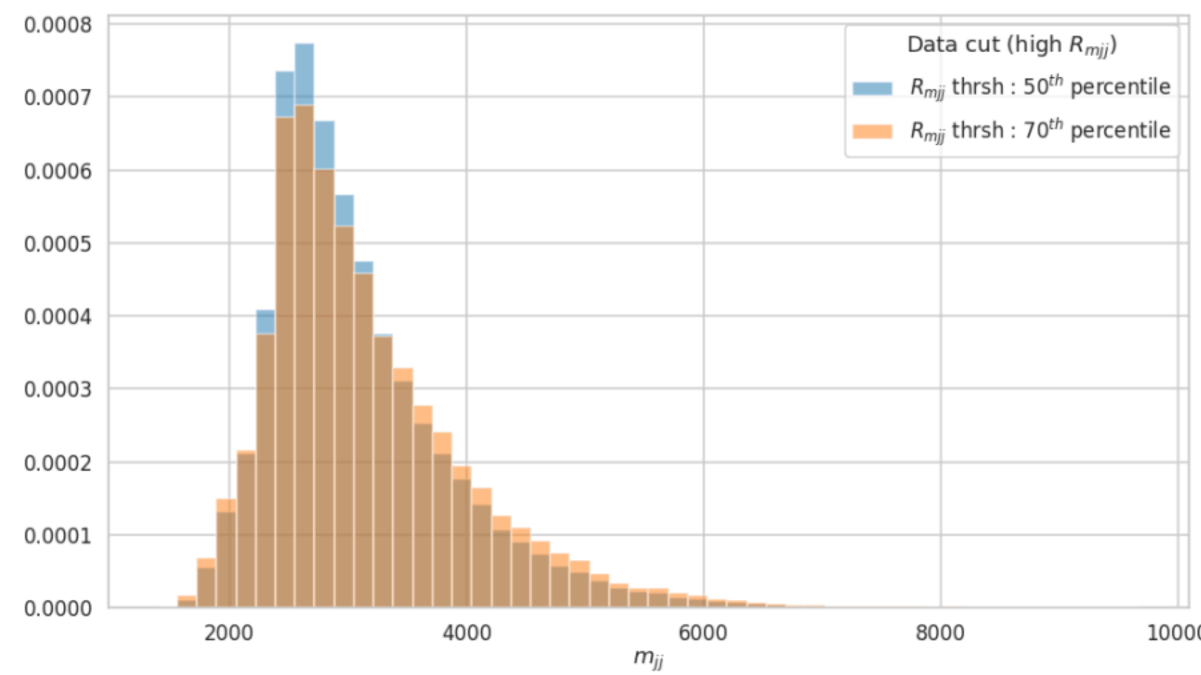
Signal Extraction : Bump Hunter (it Failed)

Take away: Mass Decorrelation+Good Simulation needed

Normalizing Flow

Inputs: High Level Features (Nsubjettiness/Jet masses/…)

BB1 Dataset



- Use a normalizing flow
  - Cut on high loss
  - Decorrelate loss with mass

$$\mathcal{R}_{m_{jj}}(x) = \frac{||x - g(g^{-1}(x))||^2}{1 + \frac{p_u(g^{-1}(x))}{p_{KDE}(m^x_{jj})}}$$

Cut is too loose (may actually work)

Signal Extraction : None (No signal)

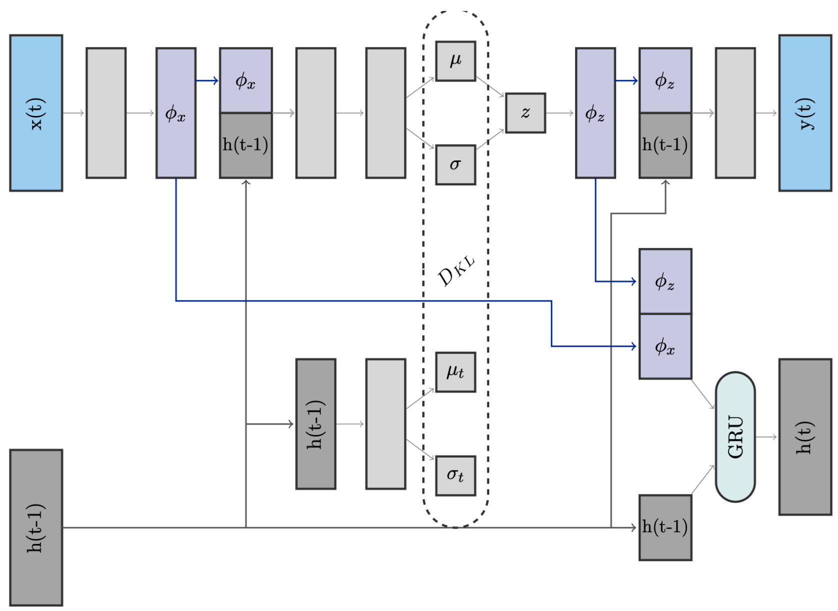
Take Away: single auto encoder even with NF is not enough
too many anomalies (no clear signal)

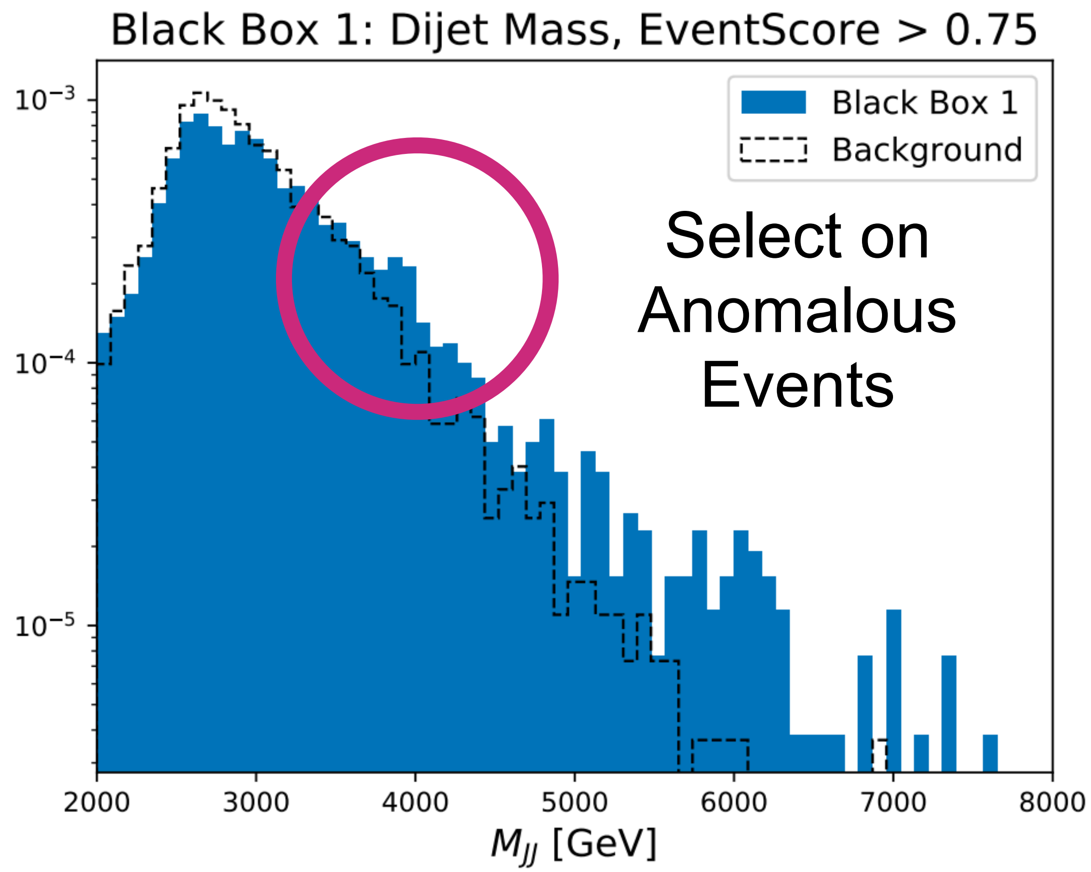
Particle VAE

Inputs: Particle four vectors of the jet

- VAE using particle inputs (RNN)

BB1 Dataset





$$\mathcal{L}(t) = \mathrm{MSE} + 0.1 \times \overline{p_T}(t) D_{\mathrm{KL}}$$

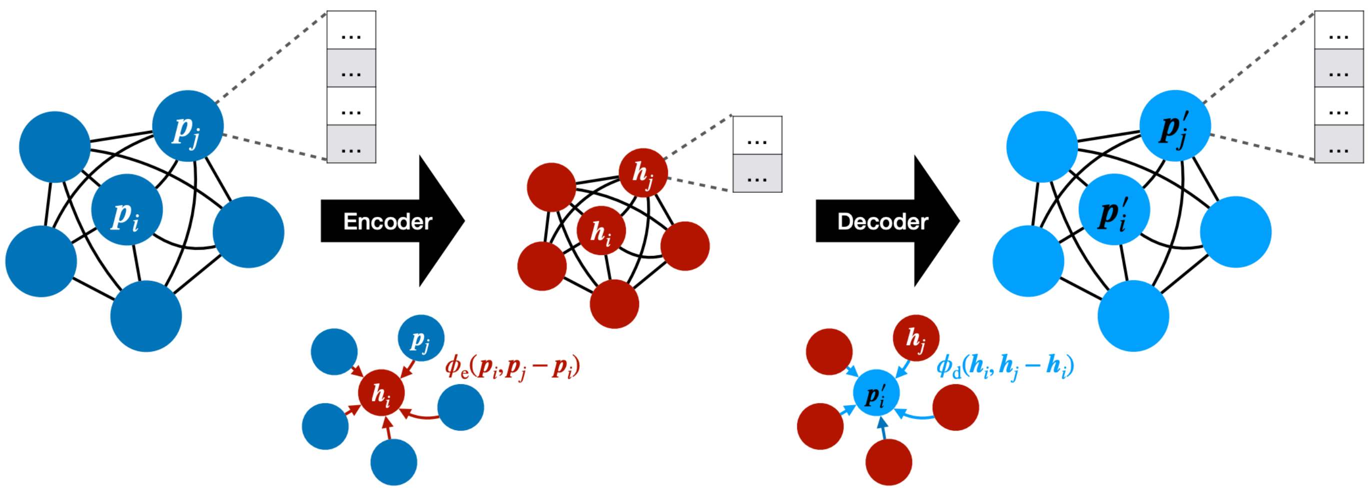
$$\text{Anomaly Score} = 1 - e^{-\overline{D_{\mathrm{KL}}}}$$

Signal Extraction : None

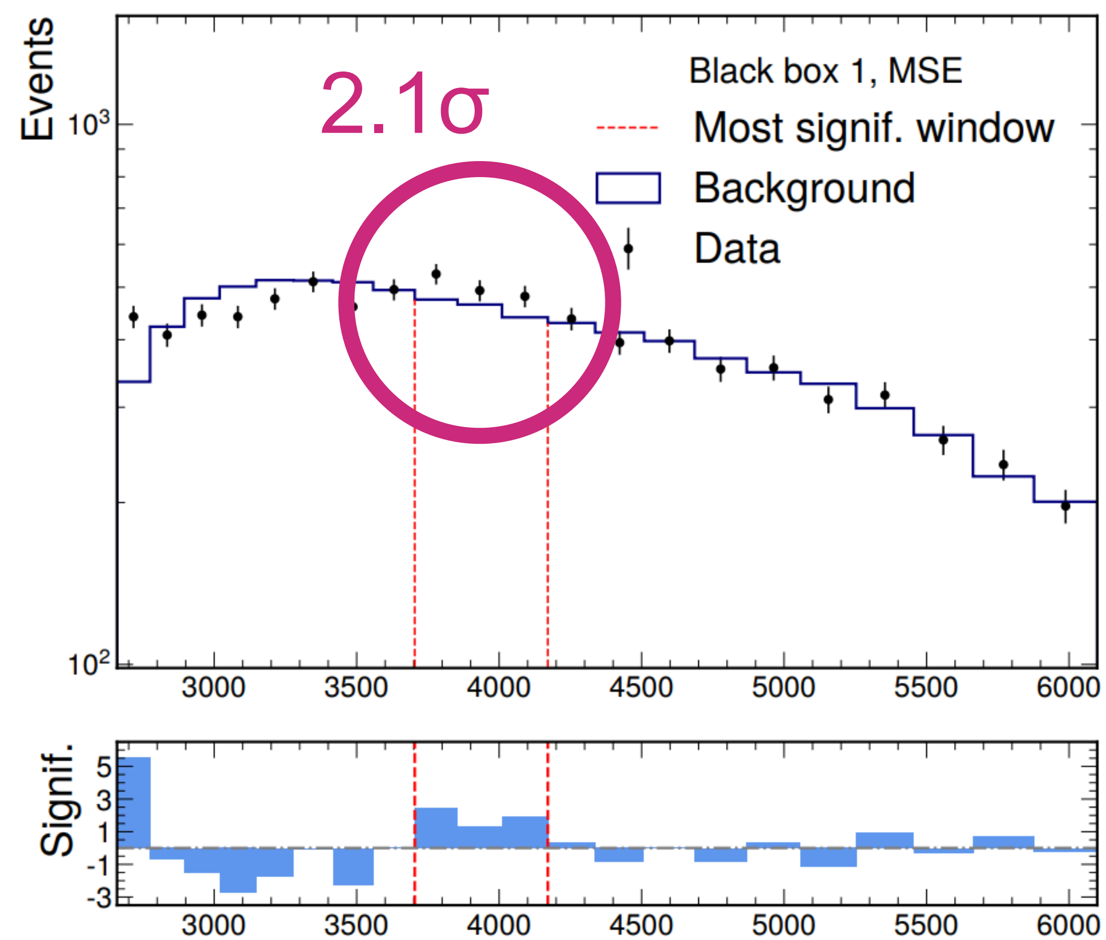
Take Away: Works but preparation of inputs is critical

Particle Graph AE

Inputs: Particle four vectors of the jet (Graph w/correlations)



BB1 Dataset



- Build a GraphNN Autoencoder
  - Try with mean squared error loss

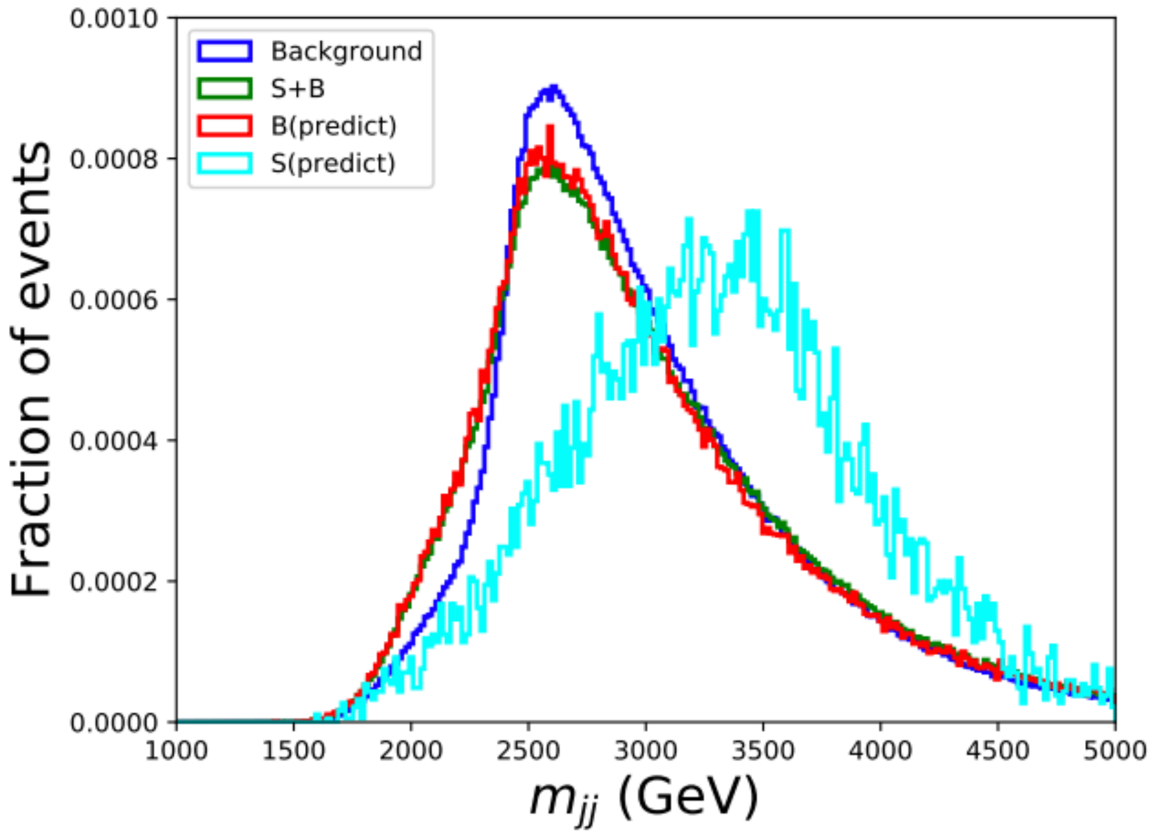
Signal Extraction : Bump Hunter Algo

Take Away: No good handle on loss

Just Training

Inputs: High level features

BB1 Dataset



Use R&D dataset and do a fully supervised training

Use the output discriminator

Try to see a signal from that

Two submissions tried

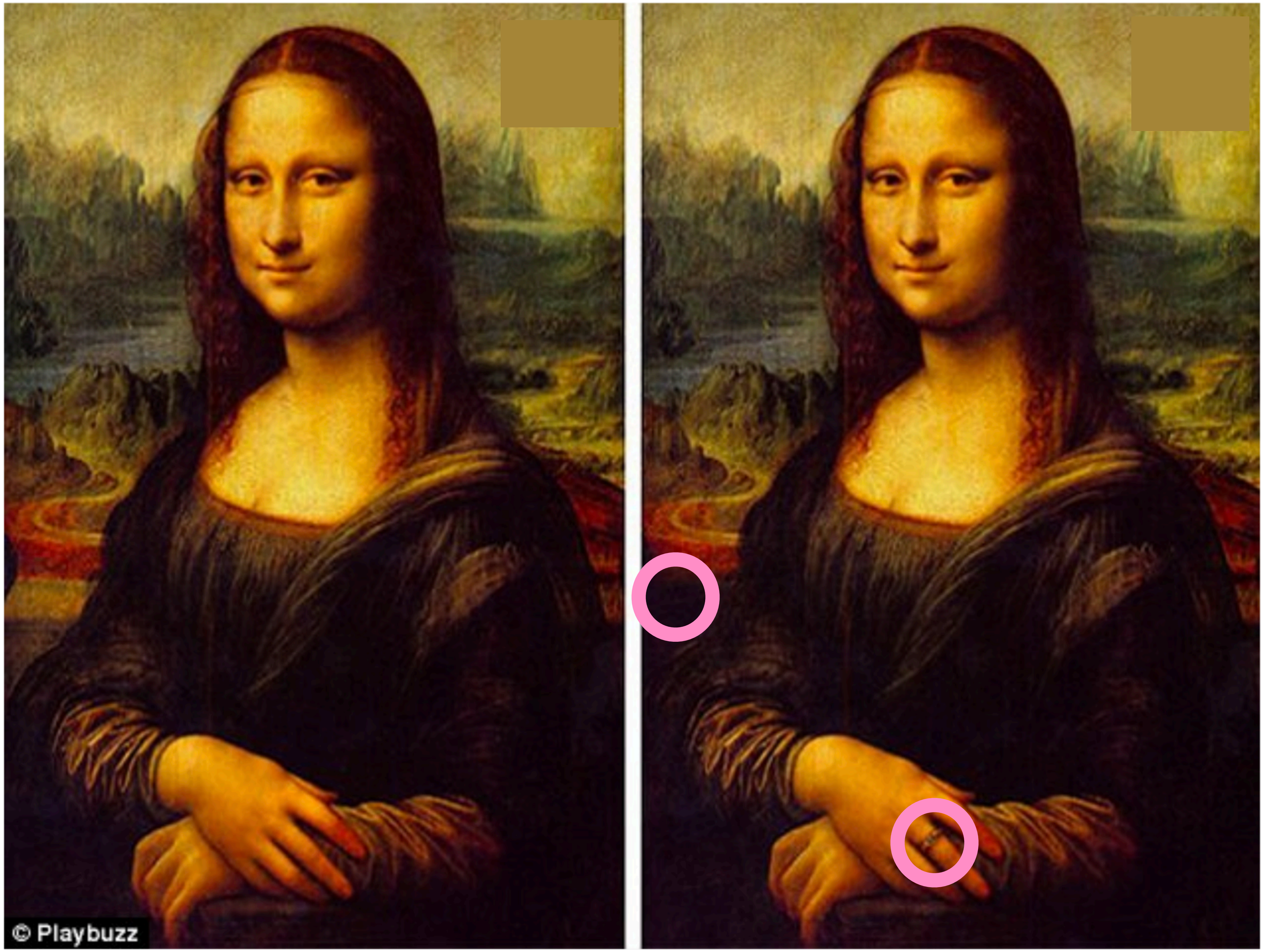
No Significant excess in either

Signal Extraction : None

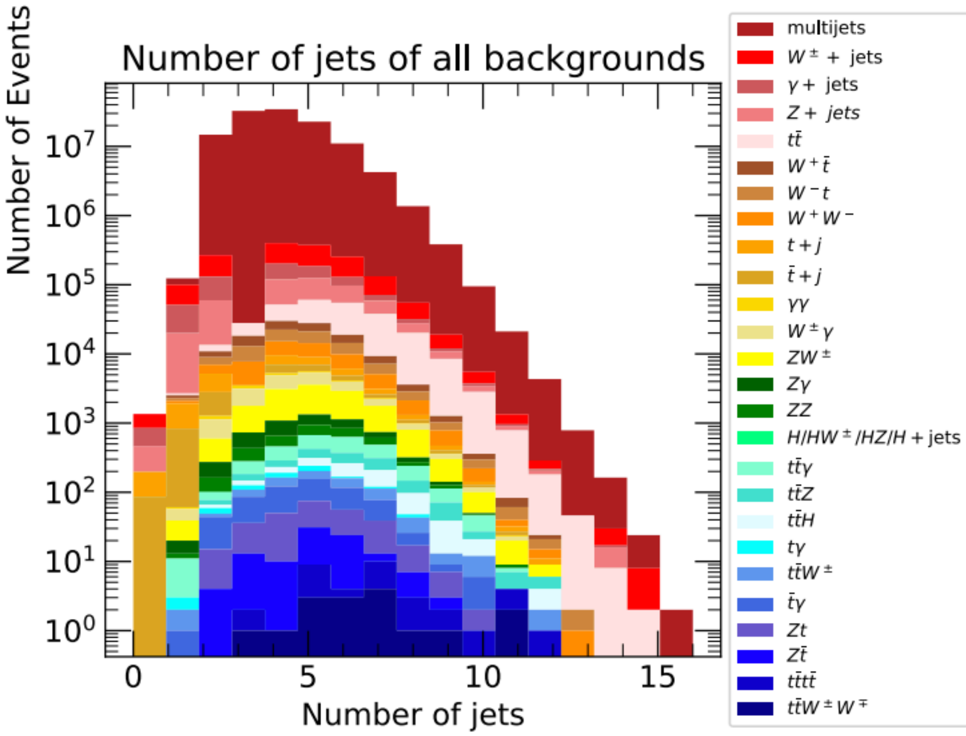
Take Away: Signal needs to be close to the hidden signal

What is different w/Left and Right?

The Need for Subtlety

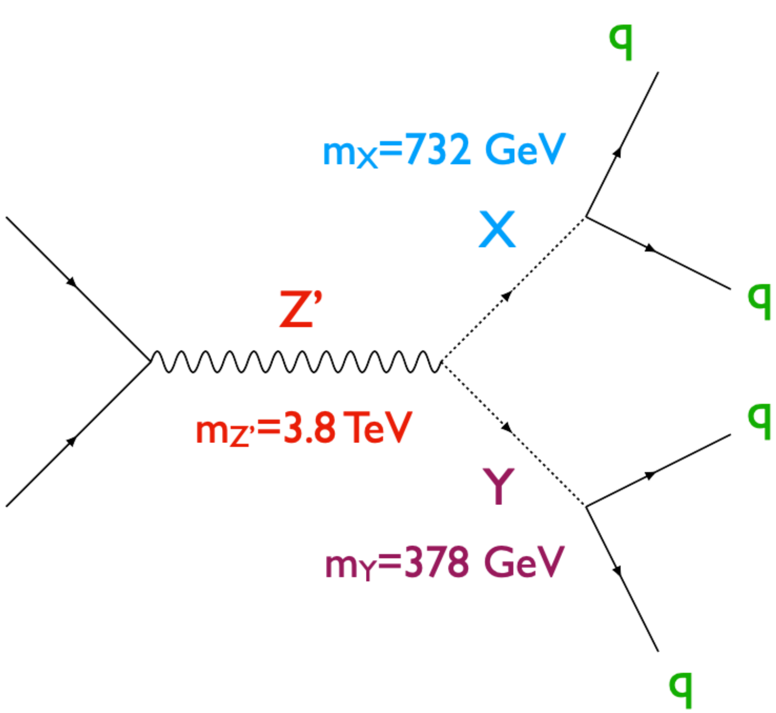


LHC Olympics Data



- ...mpics:
- ...signal and hide it in toy data
- ...k boxes split to emulate true data

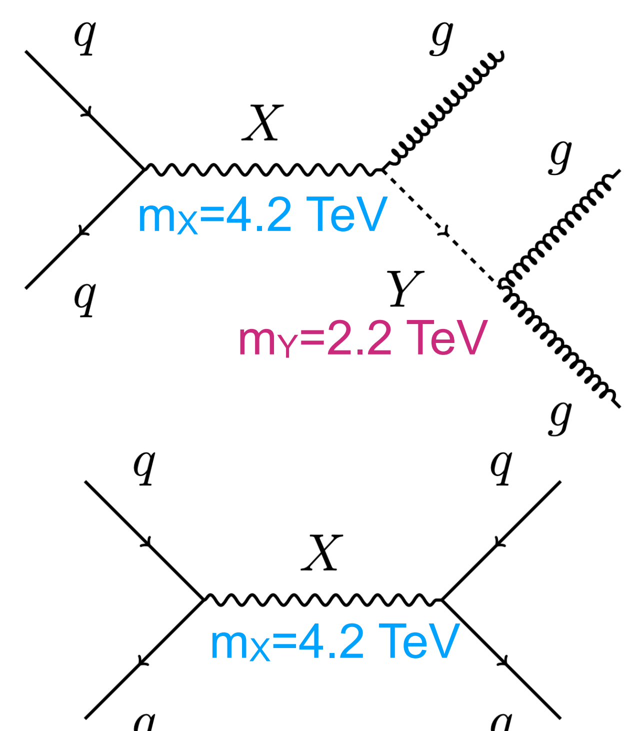
Black Box 1



Black Box 2

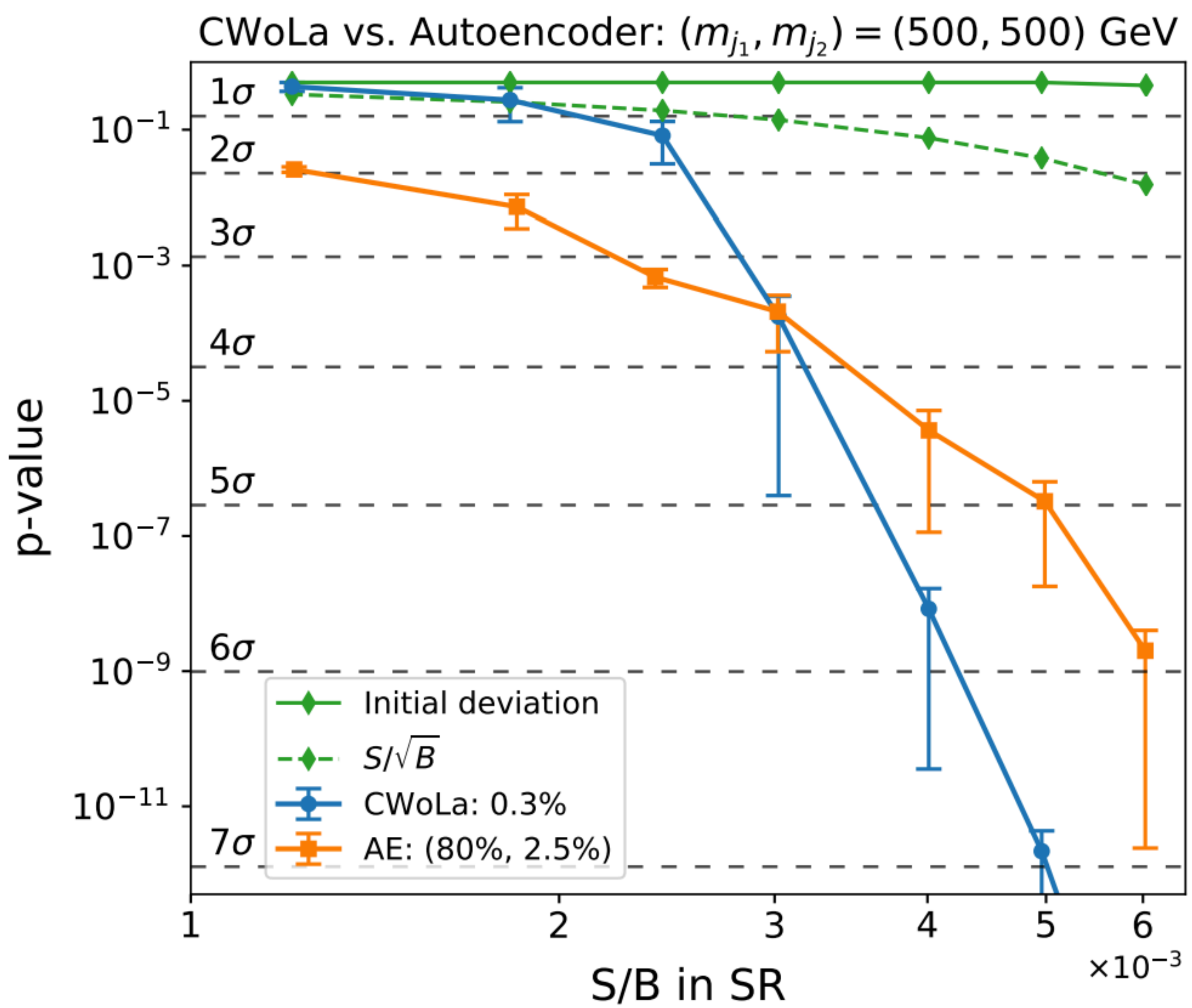
Nothing!

Black Box 3



Observation

Inputs: High level features



CWOLA works really well for large signals

But for small signals Autoencoders tend to win

You need enough events in your data to separate them

Signal Extraction : Bump fit

Take Away: Works but needed to correct dimension

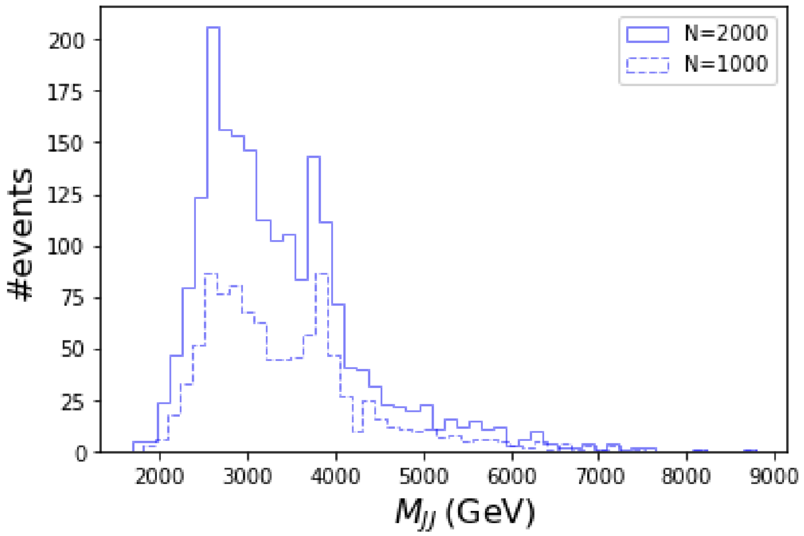
Variation of Encoder

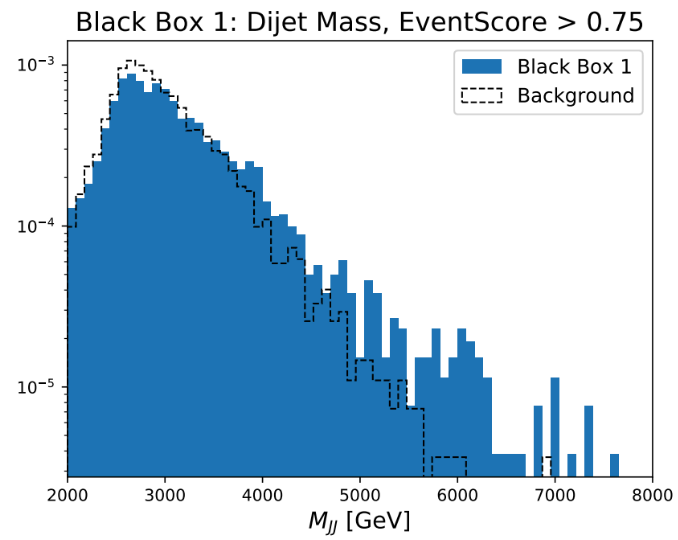
Varying the encoder architecture
can allow for a broad range of possibilities

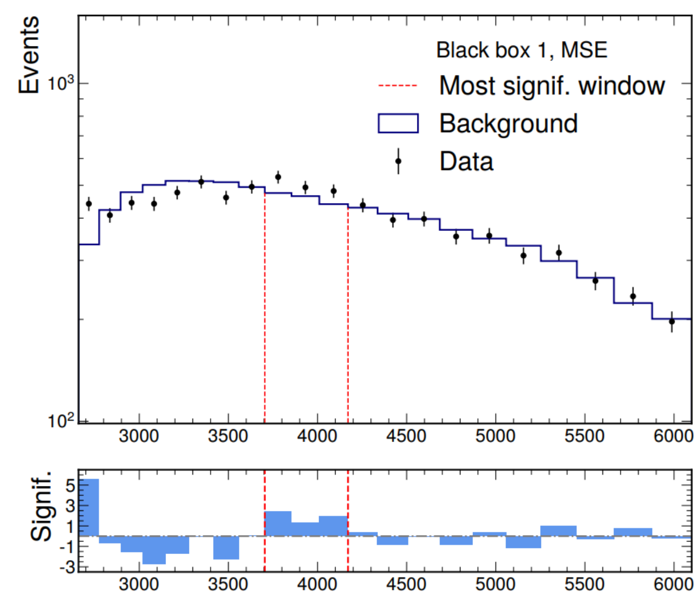
Derived Inputs

Particle Inputs

Graphs







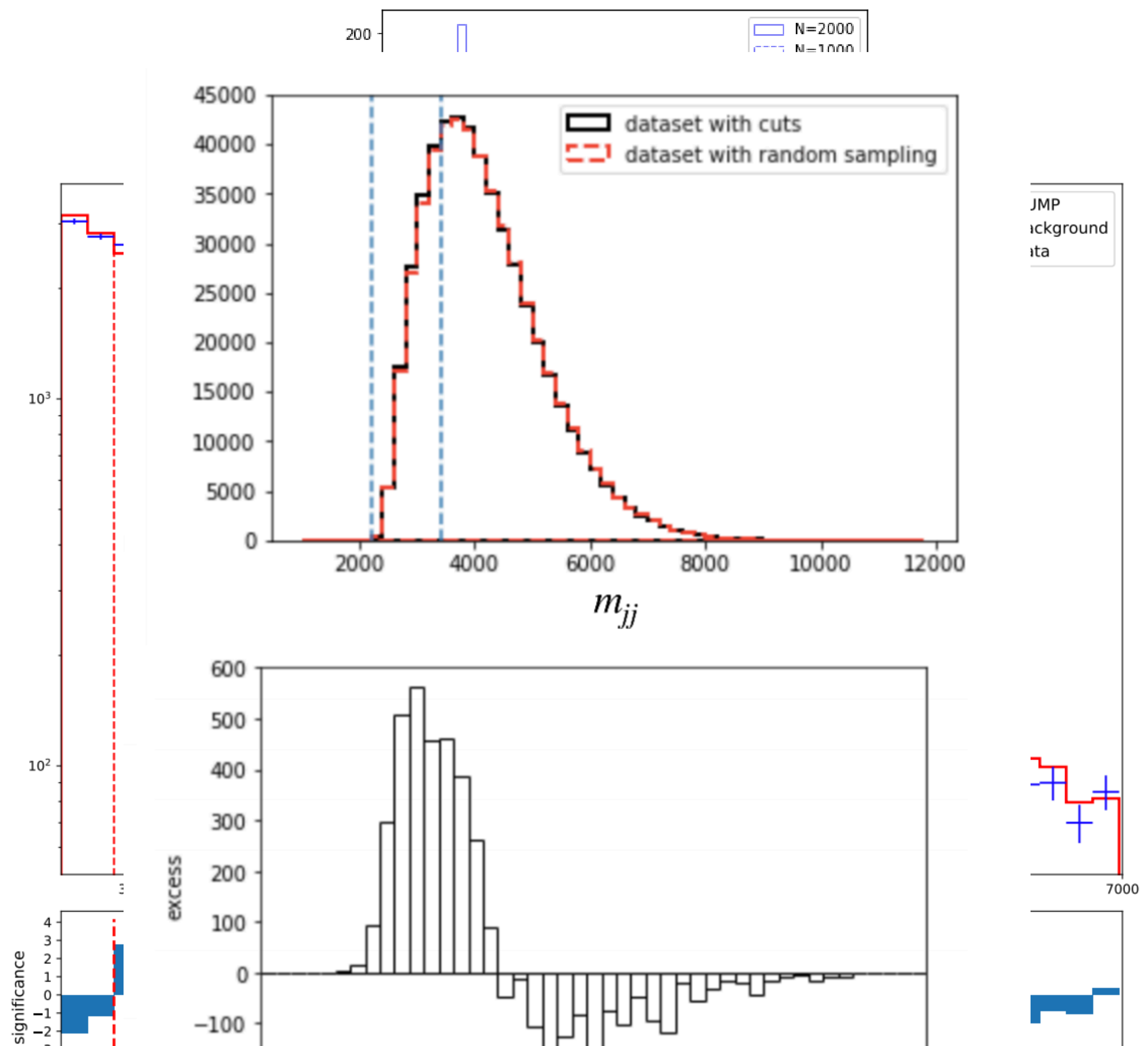
Variation of Architecture

Varying the encoder architecture
can allow for a broad range of possibilities

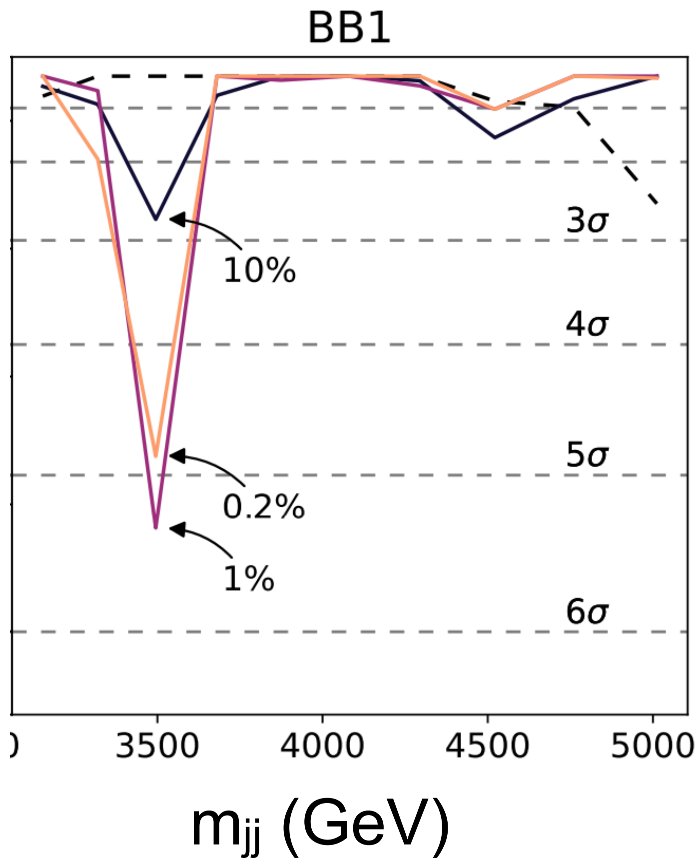
Autoencoder

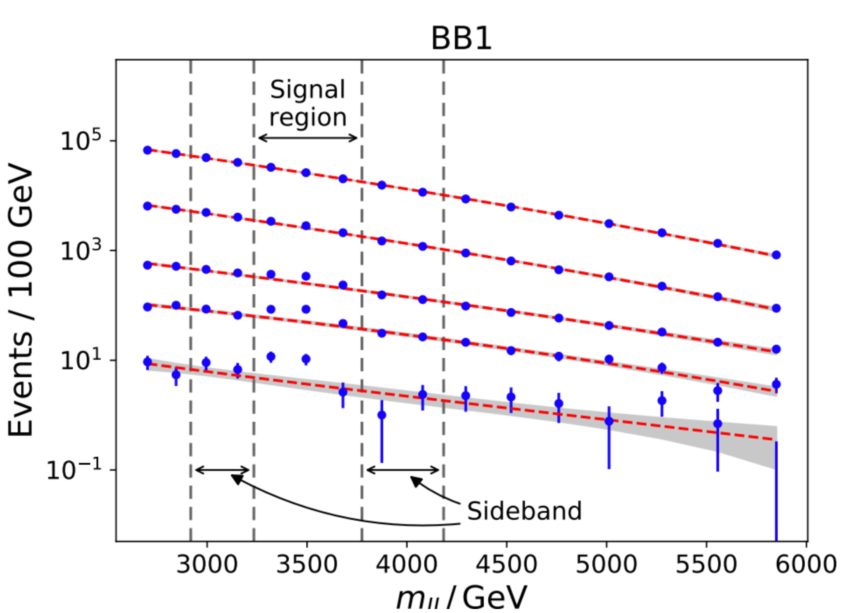
Variational Autoencoder

Normalizing Flow



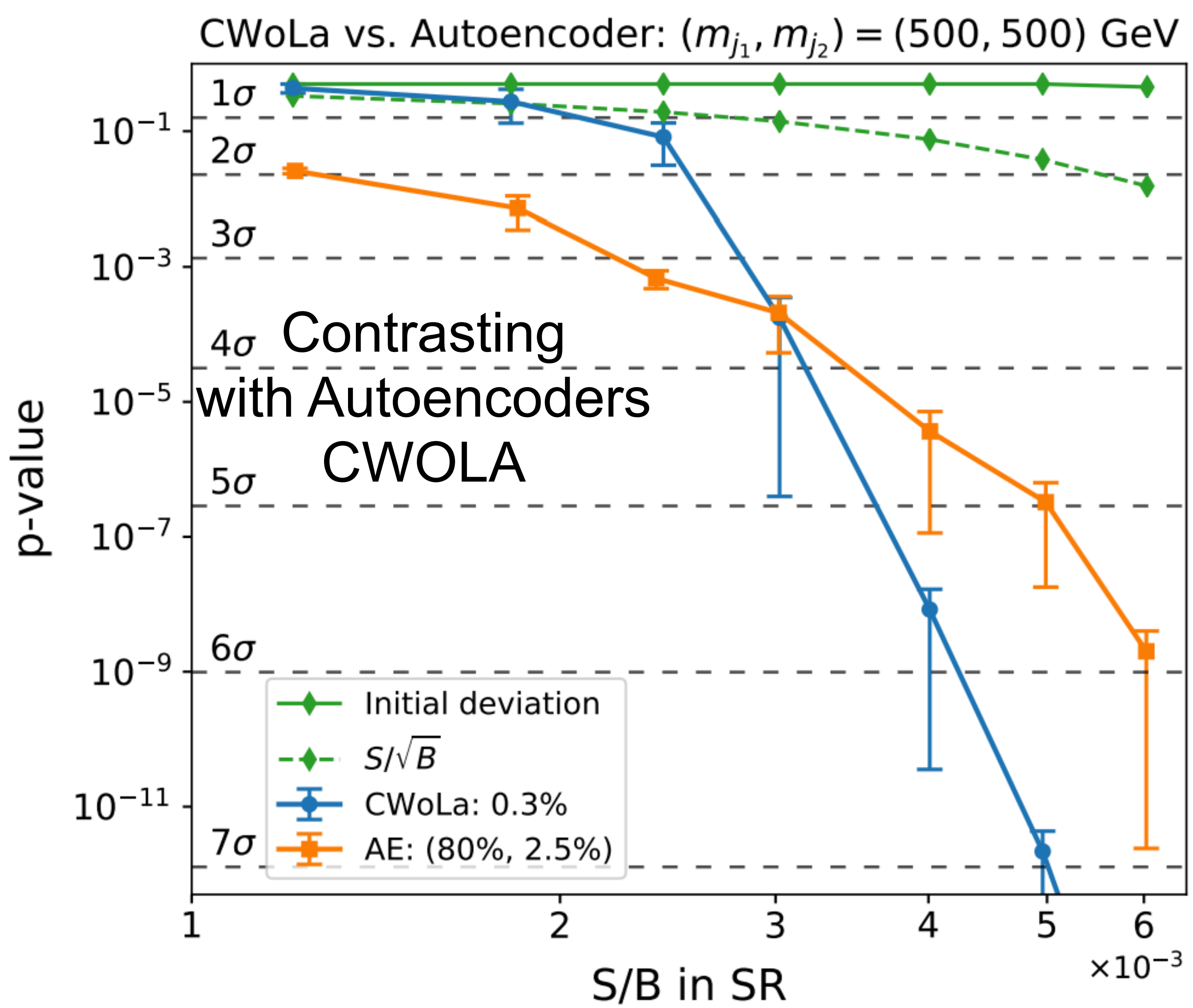
CWOLA style approach





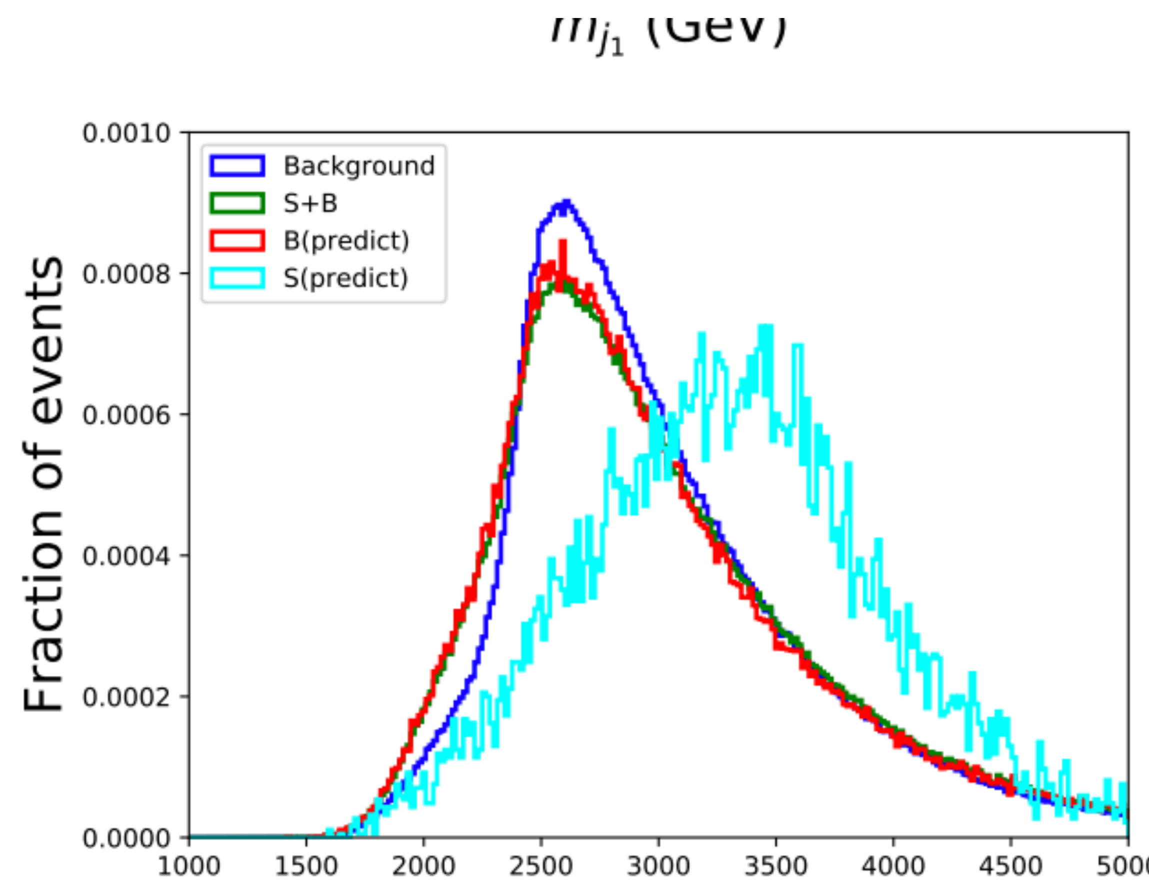
Excess at 3500 instead of 3800

- Running just a training got it to work
  - Was able to observe 5 standard deviations



Method 12:Deep Ensemble

- Use R&D dataset and do a fully supervised training
  - Use the output discriminator
  - Try to see a signal from that



- Try with both a CNN on jet images and BDT on observables

Observation:Low noise robust density estimation is key

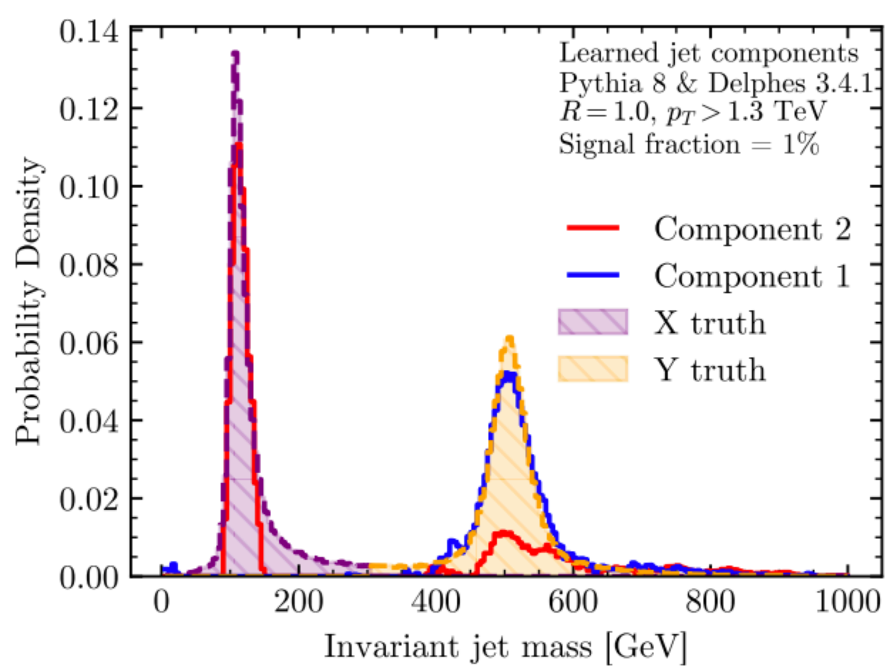
Method13:Factorized Topics

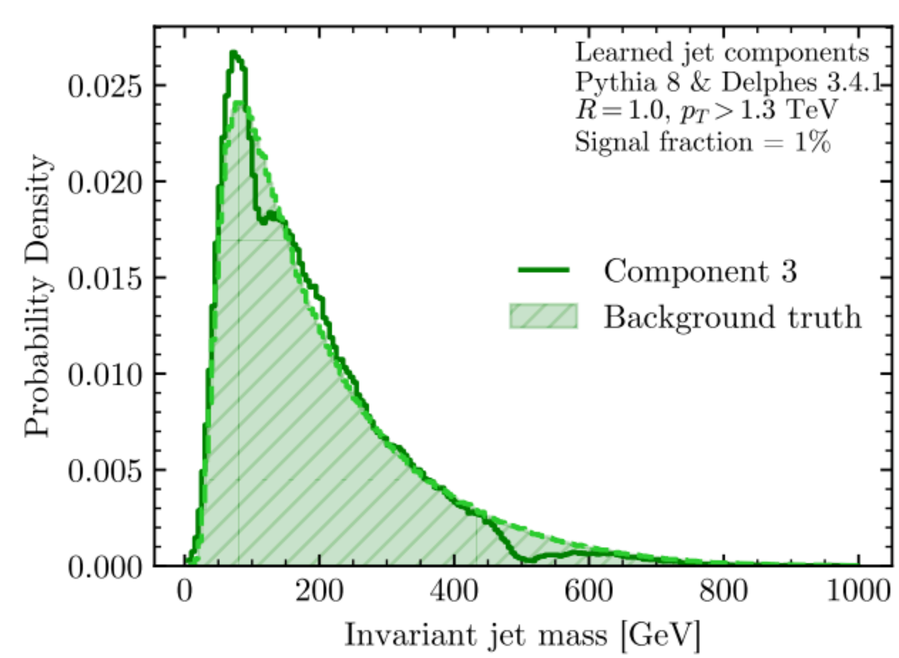
- Sample independence: each jet of a dijet can be treated as independent and for QCD its composition is the same for leading and subleading

- Factorization: jet mass distributions can be factorized

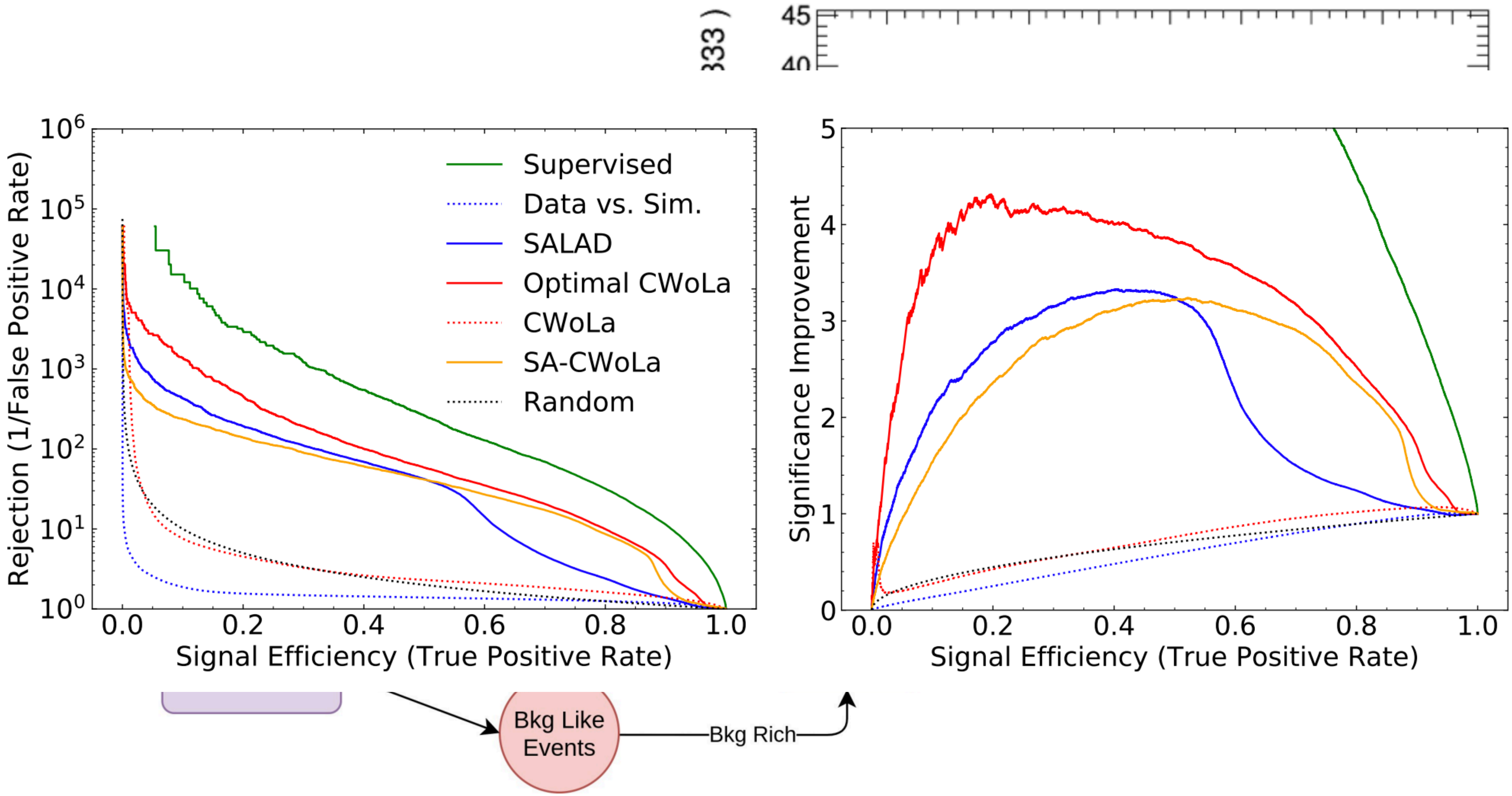
-

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = \frac{f(\text{signal}) \cdot p_{\text{signal}}(\mathbf{x}_1, \mathbf{x}_2)}{f(\text{background}) \cdot p_{\text{background}}(\mathbf{x}_1, \mathbf{x}_2)}$$
$$= \frac{f(X, Y)\, p_X(\mathbf{x}_1)\, p_Y(\mathbf{x}_2) + f(Y, X)\, p_Y(\mathbf{x}_1)\, p_X(\mathbf{x}_2)}{f(\text{QCD, QCD})\, p_{\text{QCD}}(\mathbf{x}_1)\, p_{\text{QCD}}(\mathbf{x}_2)}$$





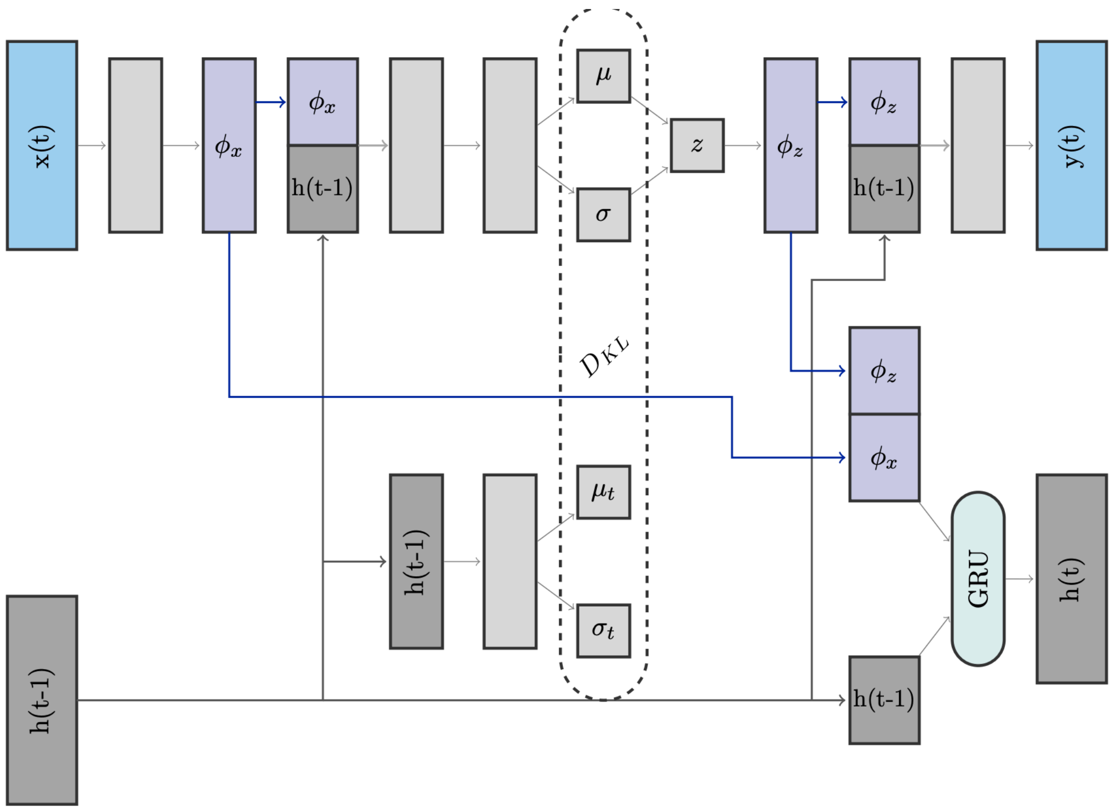
Method 10: Salad+CWOLA

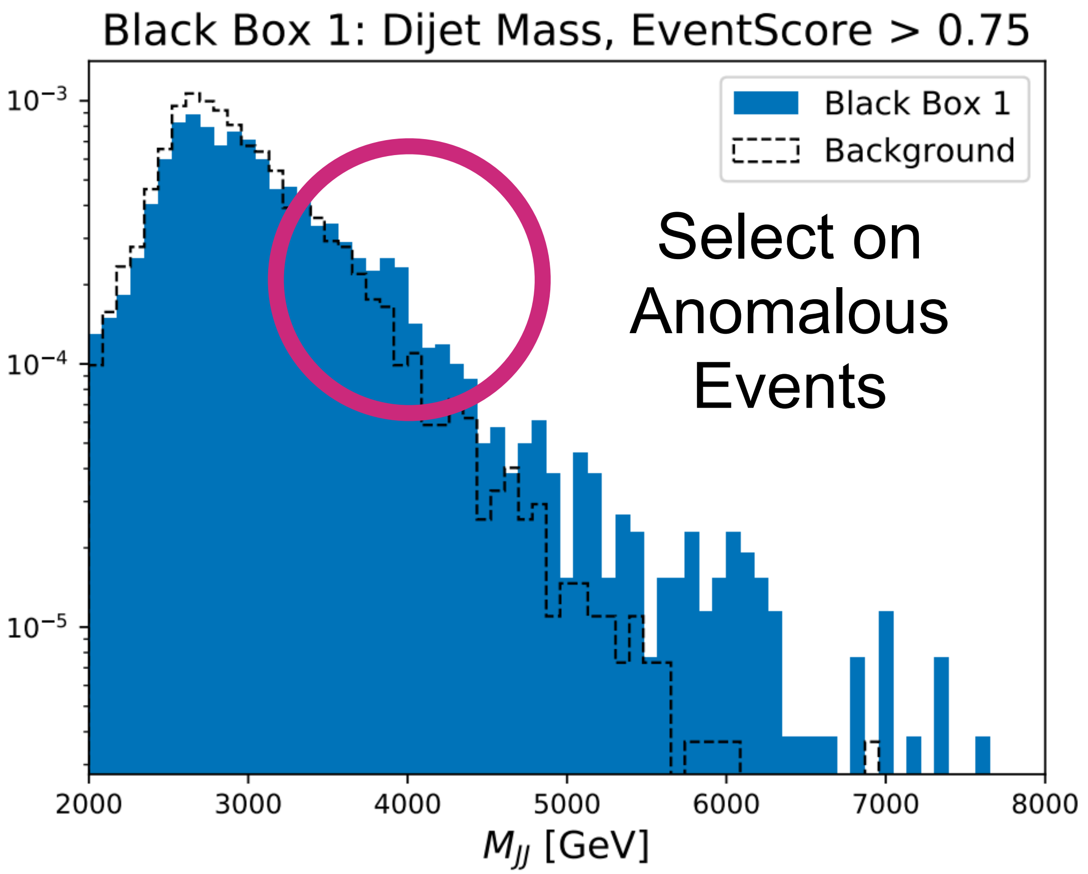


Observation:Works well on jets, some limiations from using jet images
Would benefit more from mass decorrelation

Method1:VRNN

- Variational Autoencoder using particle inputs (RNN)





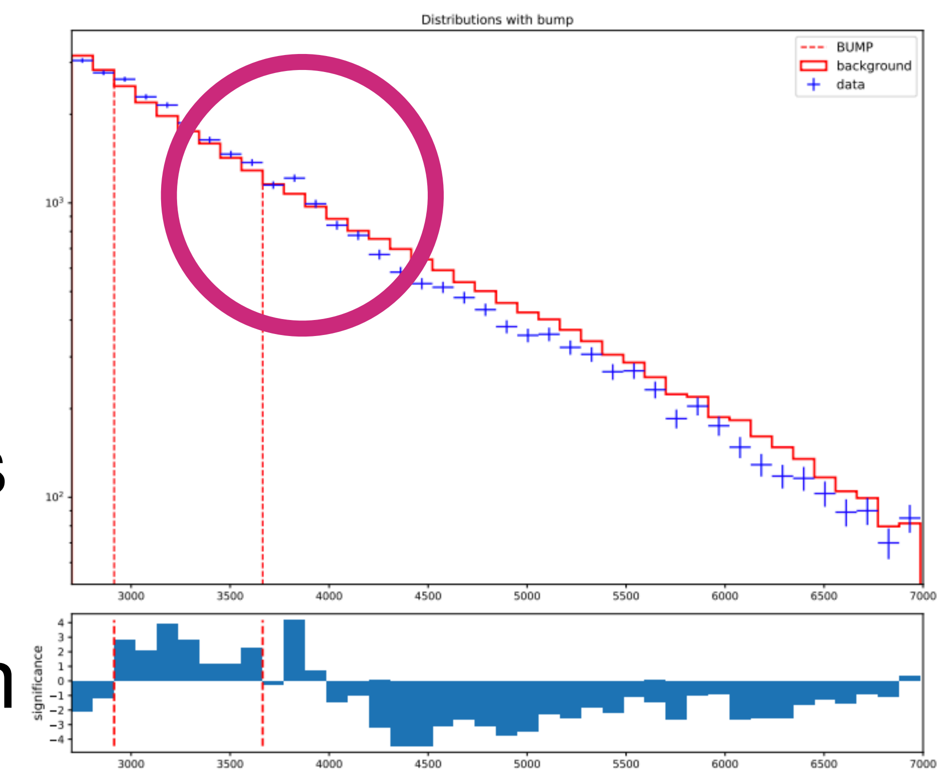
$$\mathcal{L}(t) = \mathrm{MSE} + 0.1 \times \overline{p_T}(t) D_{\mathrm{KL}}$$

$$\text{Anomaly Score} = 1 - e^{-\overline{D_{\mathrm{KL}}}}$$

Observation: Works but preparation of inputs is critical

Method 3:GAN-AE

- Build an auto encoder (AE)
  - Add an GAN to help AE
  - Additionally decorrelate with mass
  - Compute a distance (ED) for anom



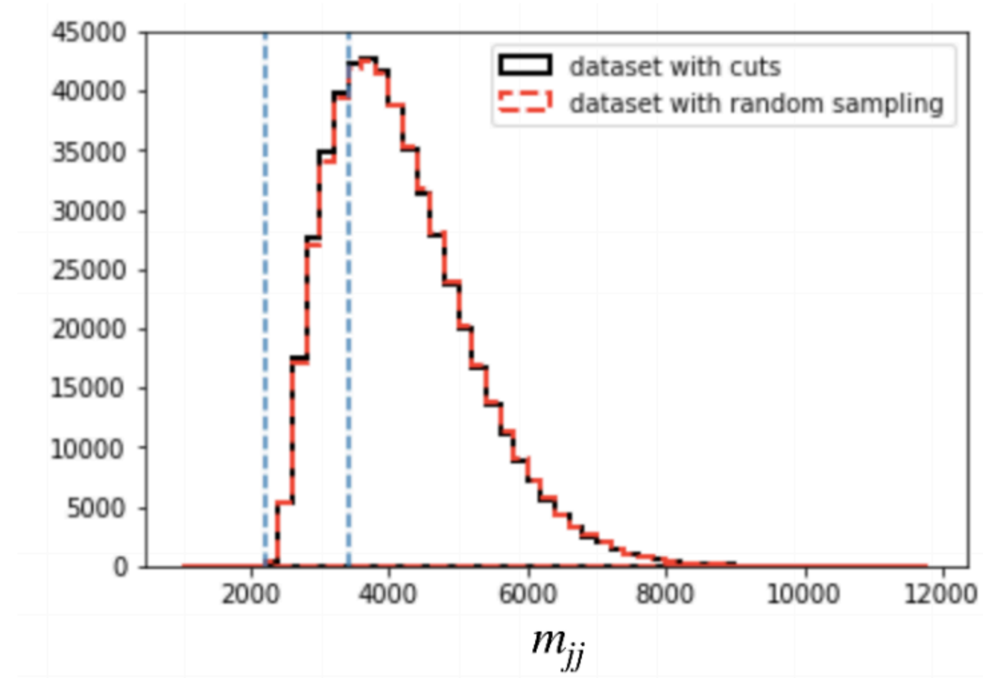
Autoencoder with 10D latent space
Latent space forced to be decorrelated with mass

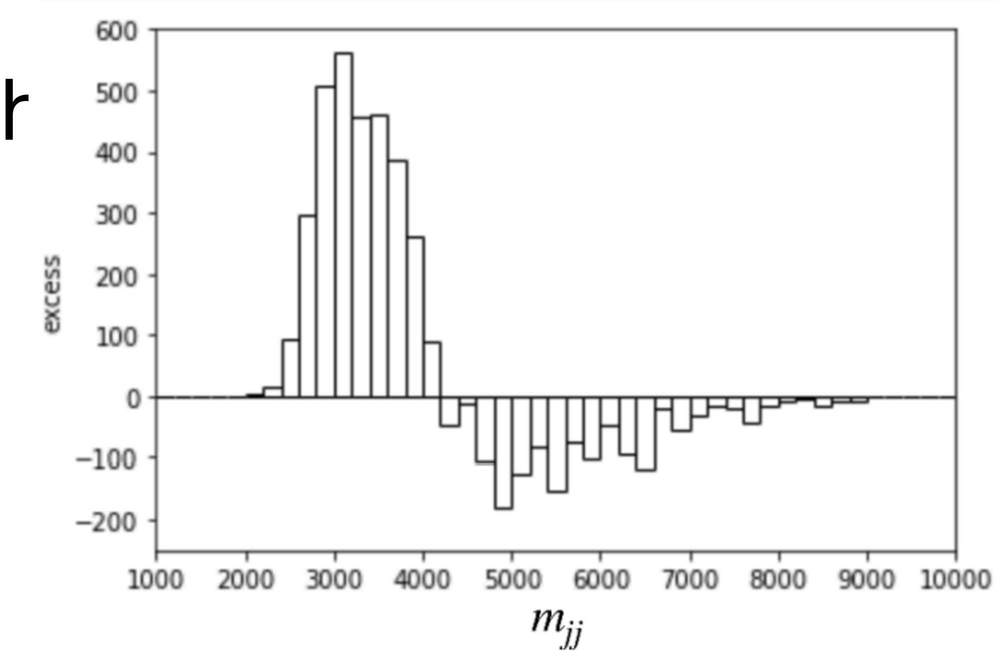
$$\mathrm{loss}_{\mathrm{AE}} = \mathrm{BC} + \varepsilon \times \mathrm{MED} + \alpha \times \mathrm{DisCo}$$

Observation: Mass Decorrelation+Good Simulation needed

Method 4:LDA

- Latent Dirichlet Allocation (LDA)
  - Decluster jet and use splitting info
  - Construct 2 hypotheses in data
    - Generated through LDA approach



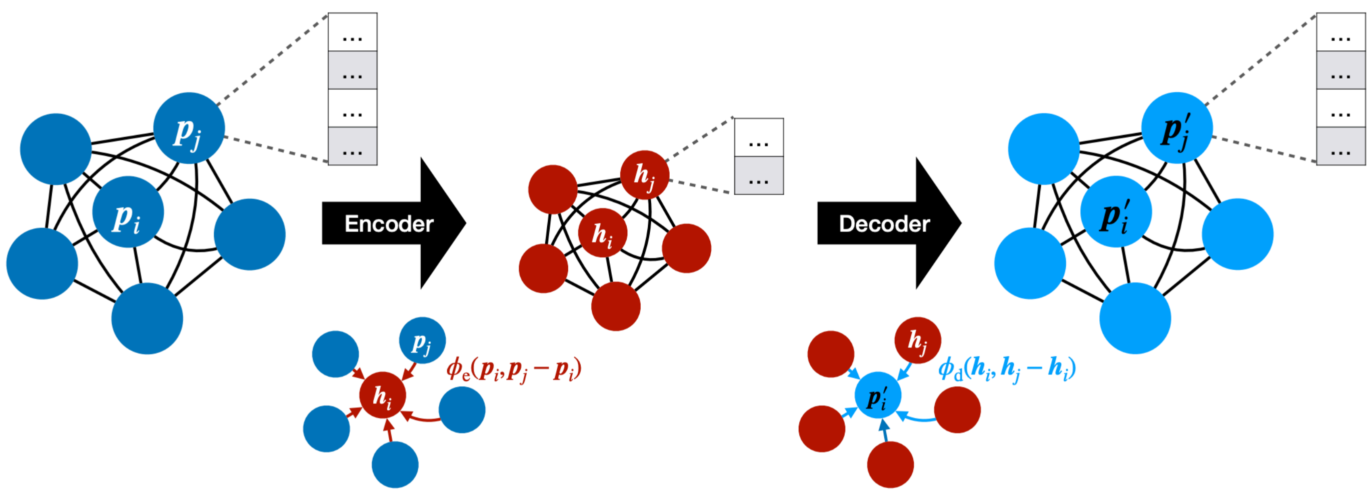


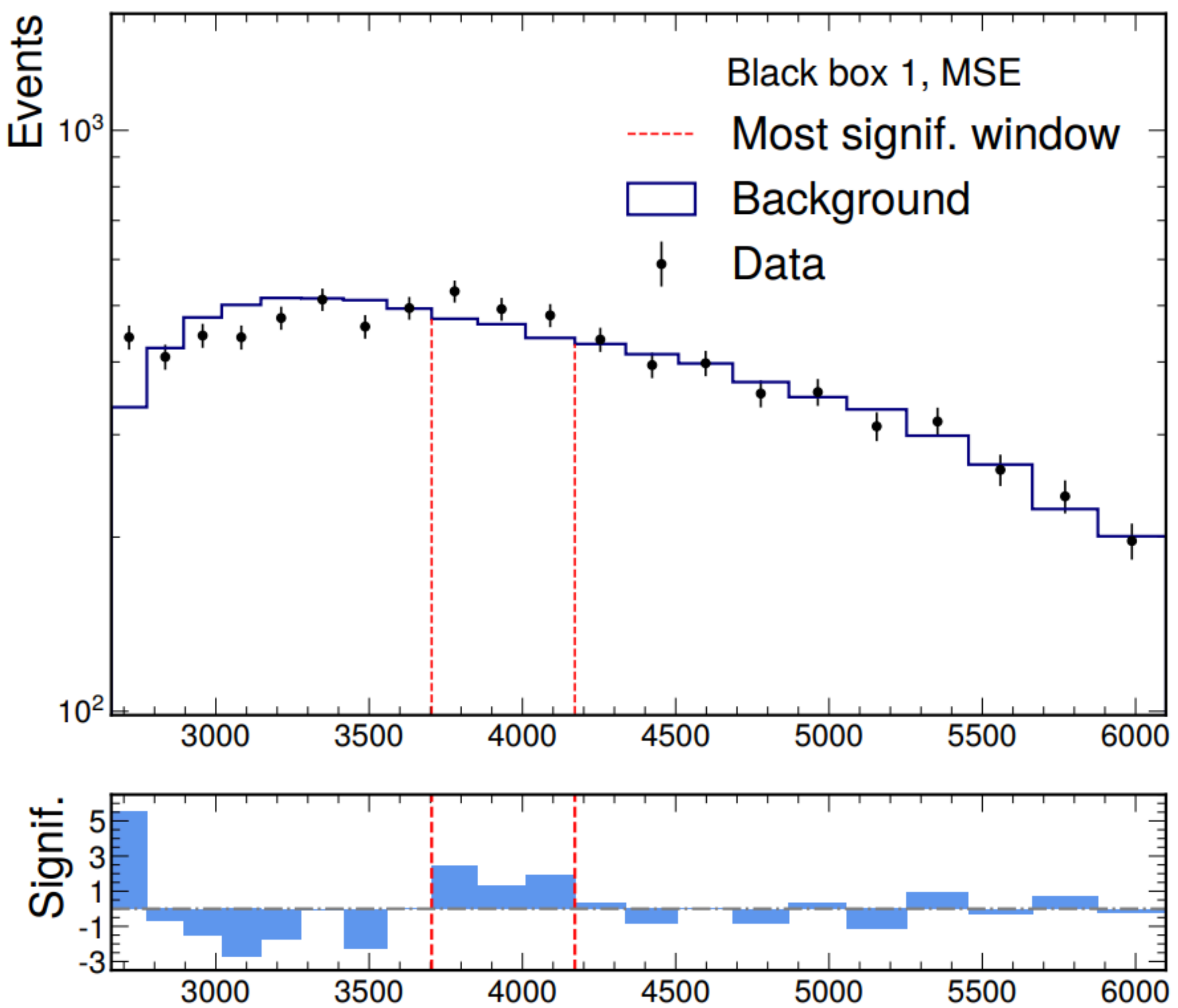
Compute likelihood of two hypoth to be consistent

$$L(o_1, \ldots, o_N|\alpha) = \prod_{i=1}^{N} \frac{p(o_i|\hat{\beta}_1(\alpha))}{p(o_i|\hat{\beta}_2(\alpha))} \, .$$

Observation: LDA benefits from many observables

Method 5: Particle Graph AE



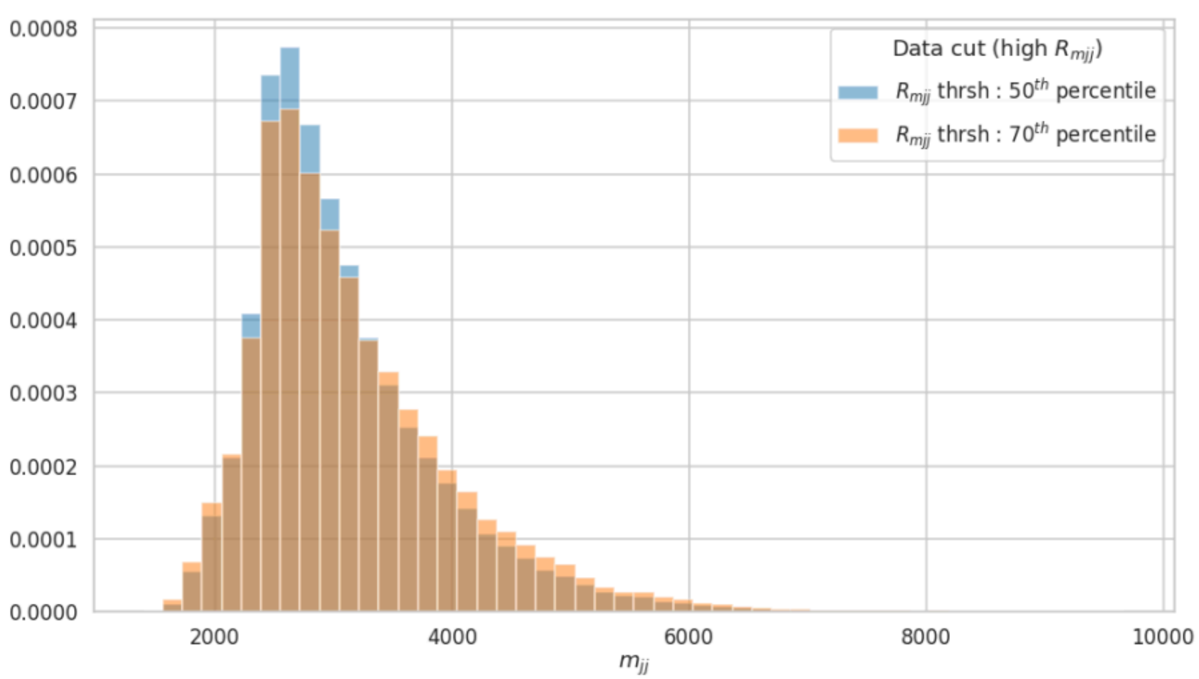


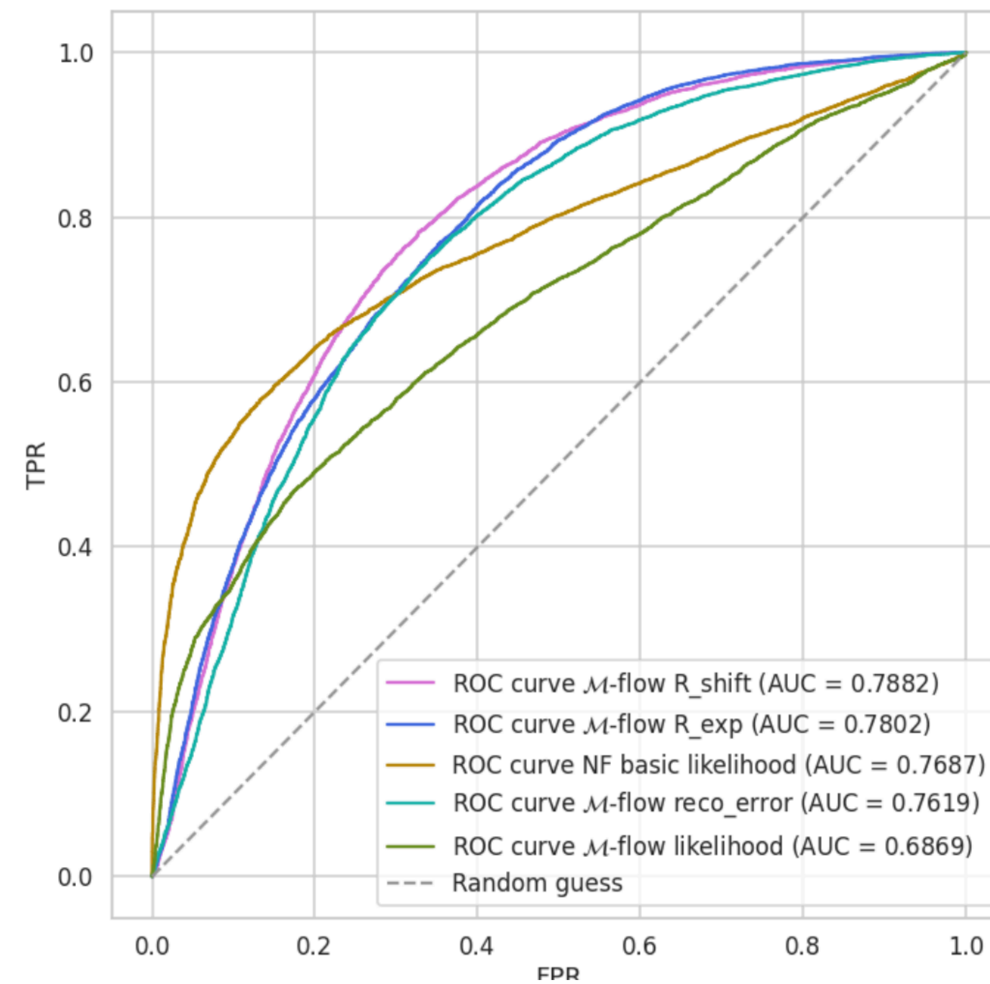
- Build a GraphNN Autoencoder
  - Try with mean squared error loss
  - Try with a permuation invariant loss (robust against physics)

Observation: No good handle on loss

Method 6: Regularized Likelihood

- Use a normalizing flow
  - Cut on high loss
  - Decorrelate loss with mass



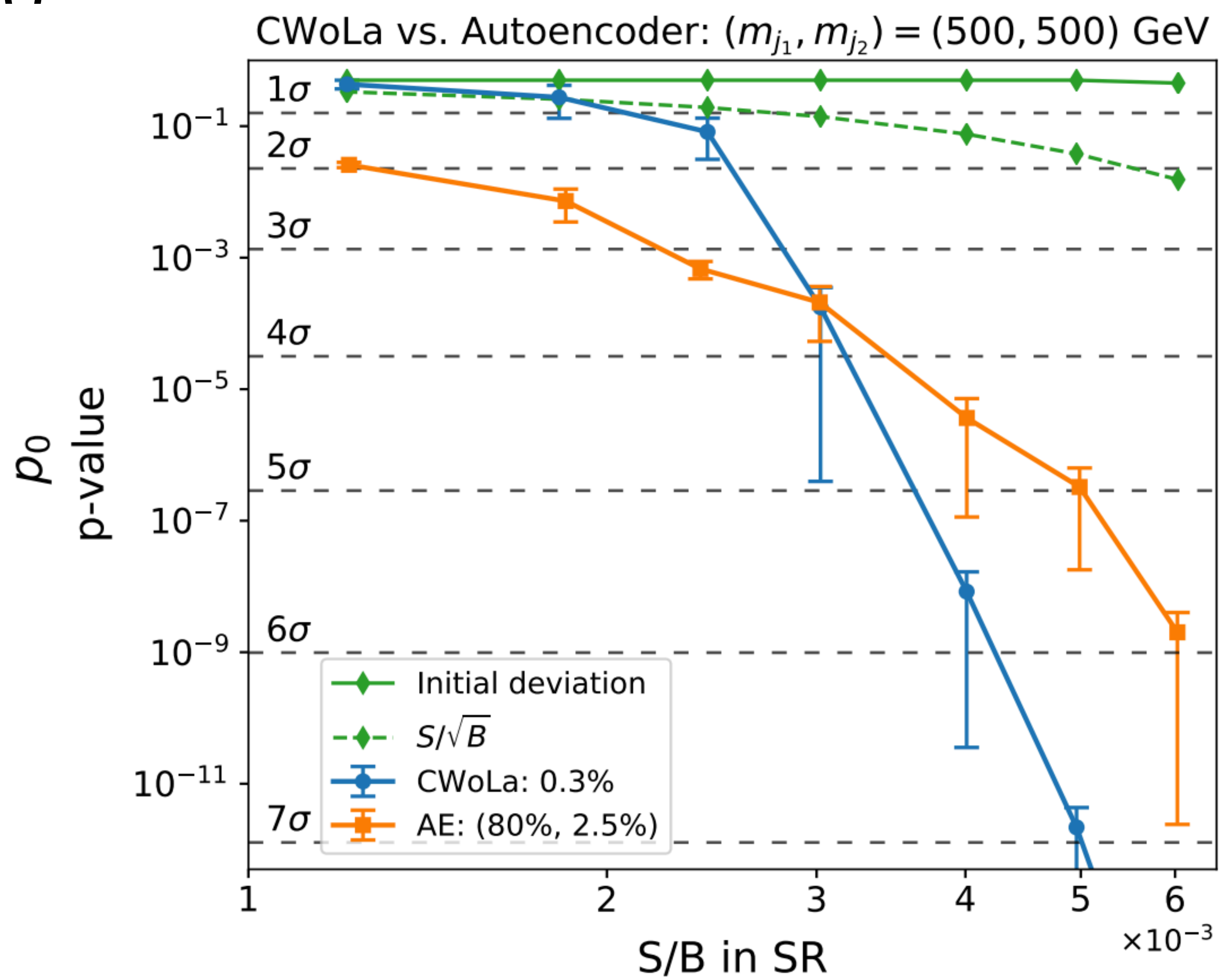


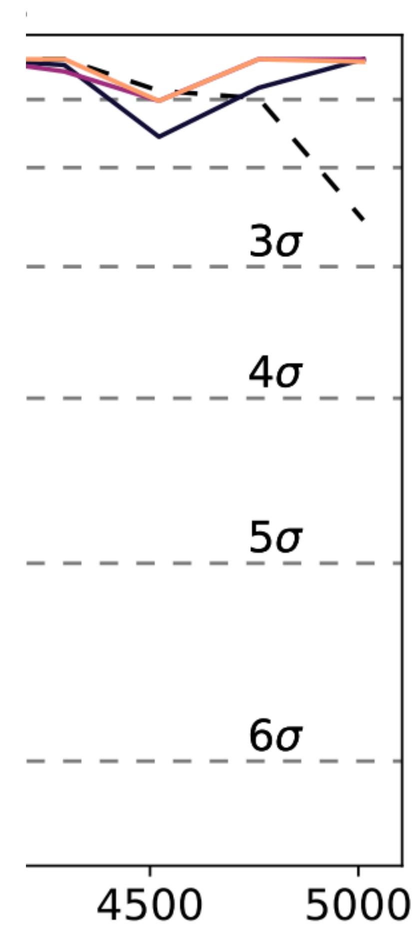
$$\mathcal{R}_{m_{jj}}(x) = \frac{||x - g(g^{-1}(x))||^2}{1 + \frac{p_u(g^{-1}(x))}{p_{KDE}(m^x_{jj})}}$$

Observation: A single auto encoder even with NF is not enough too many anomalies (no clear signal)

Method 8: CWoLa

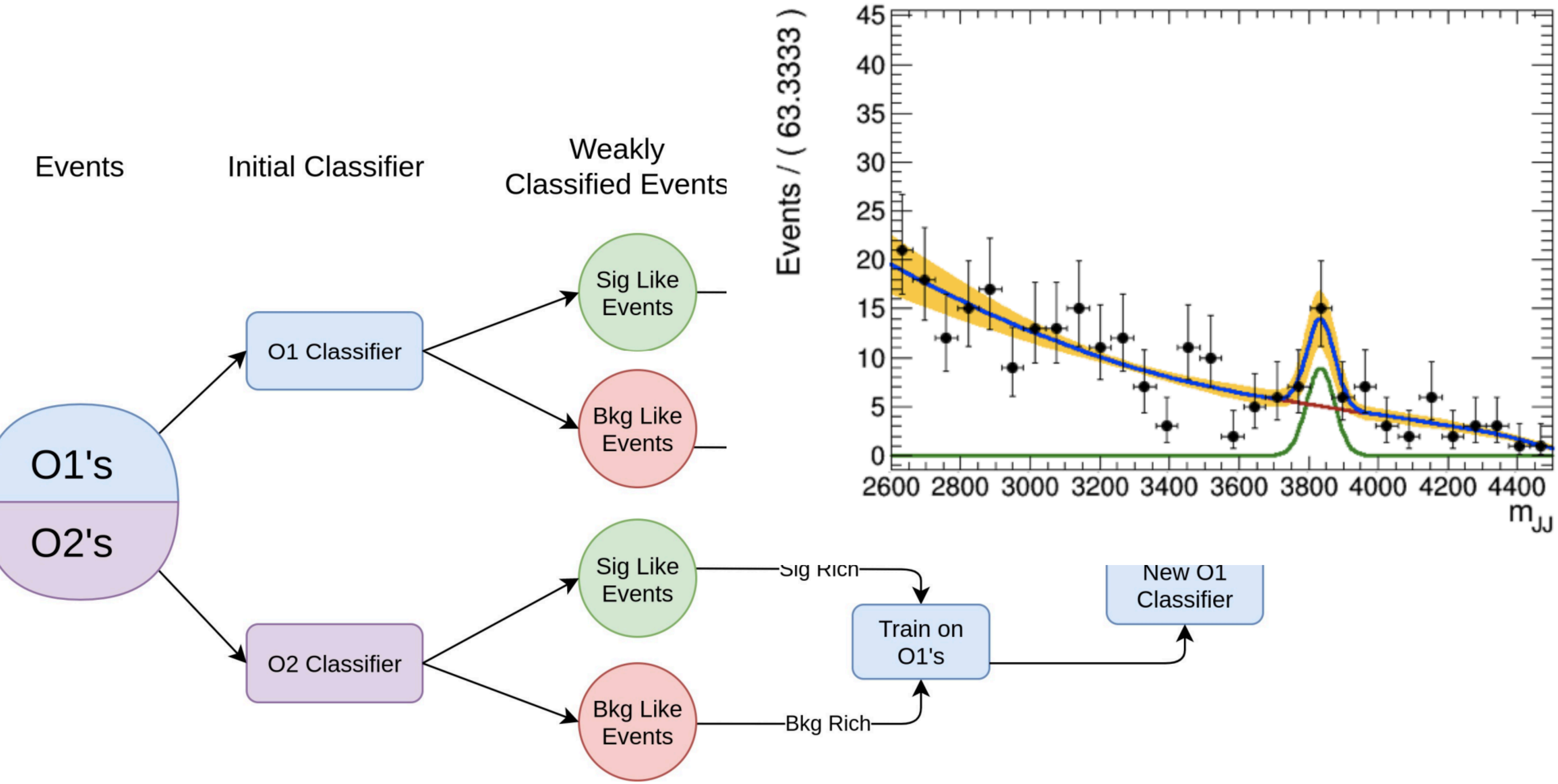
- Use a normalizing flow
  - Cut on high lo
  - Decorrelate





Observation:Approach works for single jet resonances

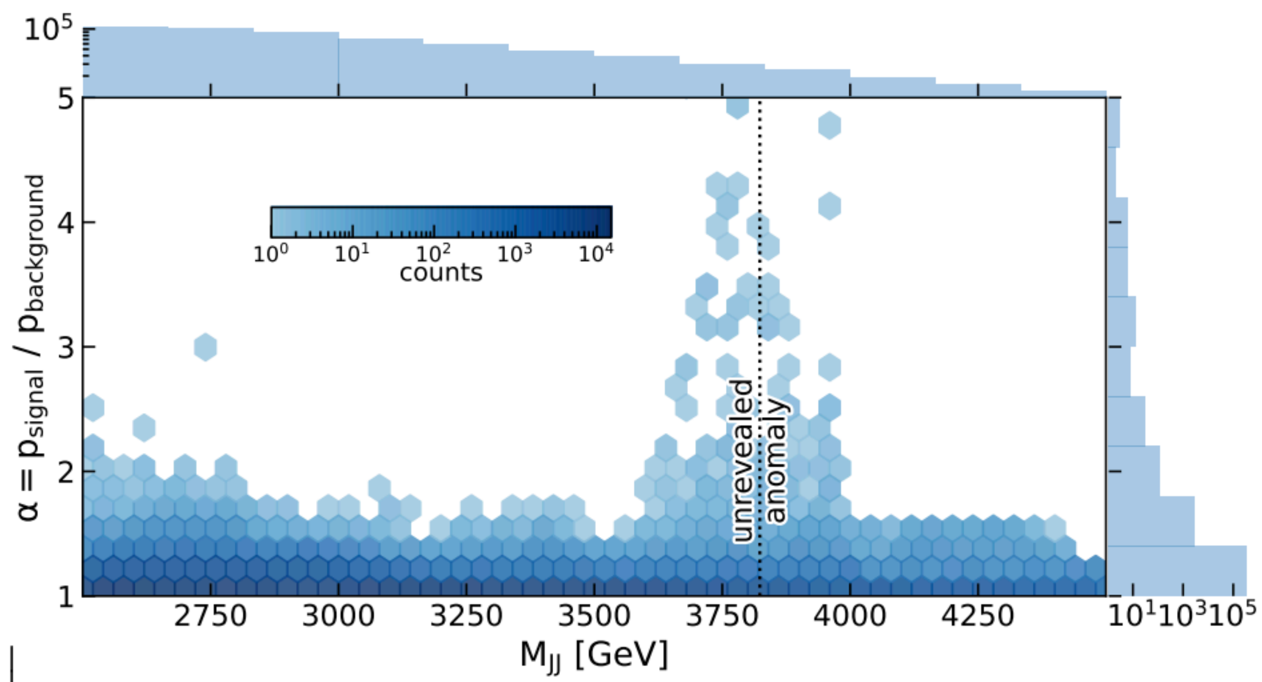
Method 9: Tag N'Train



Observation:Works well on jets, some limiations from using jet images
Would benefit more from mass decorrelation

Method 11:GIS

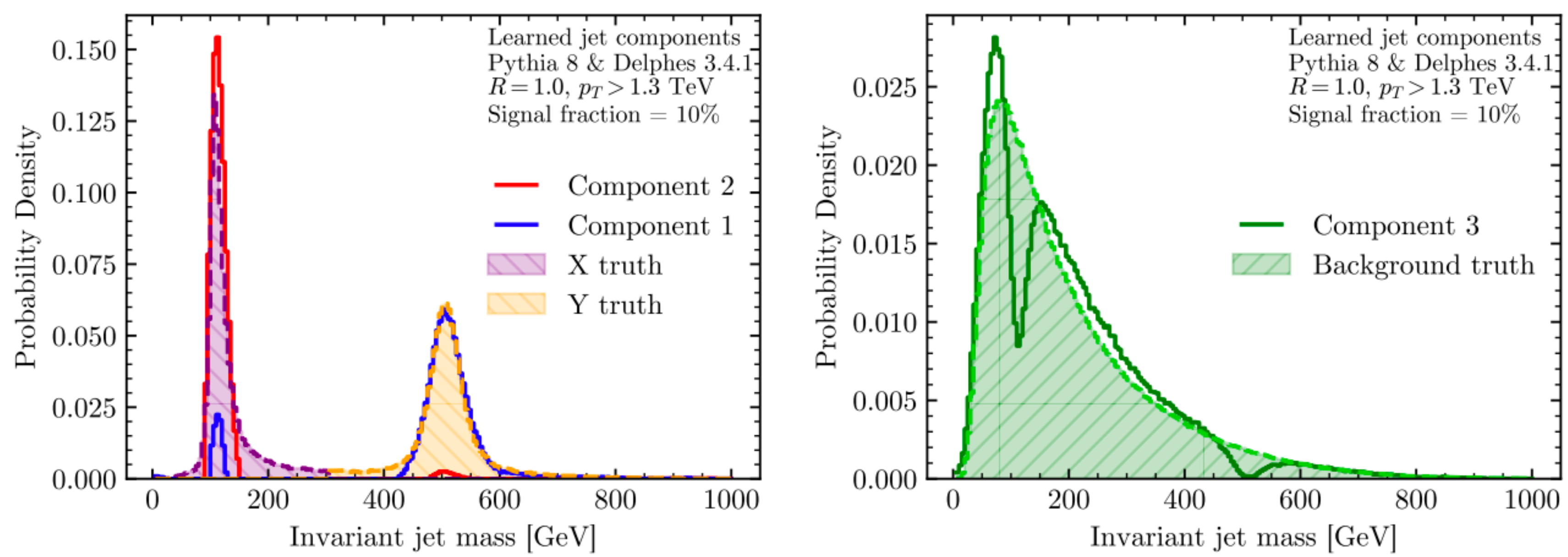
- Guassian Iterative Slicing
  - Cut on high loss
  - Decorrelate loss with mas



$$p(x|x_c) = \pi(f_{x_c}(x)) \left|\det\left(\frac{\partial f_{x_c}(x)}{\partial x}\right)\right| = \pi(f_{x_c}(x)) \prod_{i=1}^{i=N} \left|\det\left(\frac{\partial f_{x_c,i}(x)}{\partial x}\right)\right|.$$

Observation:Low noise robust density estimation is key

Method14:QUAK



Data Format

- Data released in h5 format
    - Standard python format using h5py and pandas
    - Easy to process tools that allow for quick turnaround

Entrée [1]:
```python
import numpy as np
import matplotlib.pyplot as plt
import h5py
import pandas as pd
```

Entrée [2]:
```python
file='events_anomalydetection_Z_XY_qqq.h5'
#f_sig = h5py.File(file,'r')
pd.read_hdf(file)
```

Out[2]:

	0	1	2	3	4	5	6	7	8	9	...	2091	2092	2093	2094	2095	2096
0	18.283588	-0.903479	0.060979	.316431	-0.784941	-0.008755	9.464178	-0.812918	-0.037386	4.578035	...	0.0	0.0	0.0	0.0	0.0	0.0
1	17.661003	-0.446288	-1.379160	.478683	-0.458125	-1.373650	8.452606	-0.455308	-1.375457	99.440353	...	0.0	0.0	0.0	0.0	0.0	0.0

Particle #1 Particle #2 Particle #3

Search for Non-Standard Sources of Parity Violation in Jets at $\sqrt{s} = 8$ TeV with CMS Open Data

Christopher G. Lester[a] Matthias Schott[b,c]

[a] *Cavendish Laboratory, University of Cambridge, UK*
[b] *Massachusetts Institute of Technology, Cambridge, USA*
[c] *Johannes Gutenberg-University, Mainz, Germany*

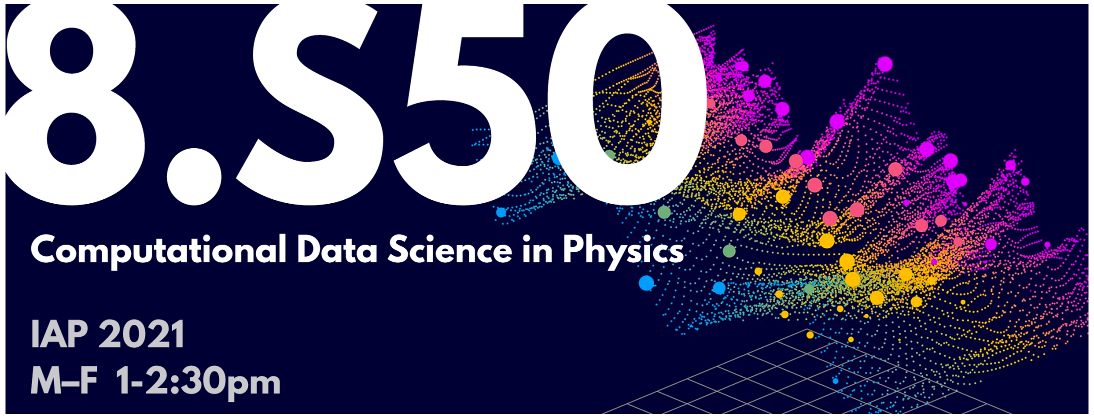
E-mail: lester@hep.phy.cam.ac.uk, matthias.schott@cern.ch

Opportunities and Challenges of Sta
Production Cross Section Measurem
Proton–Proton Collisions at $\sqrt{s}$=8 TeV using CMS Open Data

Aram Apyan[a] William Cuozzo[b] Markus Klute[b] Yoshihiro Saito[b] Matthias Schott[1b,c] Bereket Sintayehu[b]

[a] *Fermilab, USA*
[b] *Massachusetts Institute of Technology, Cambridge, USA*
[c] *Johannes Gutenberg-University, Mainz, Germany*

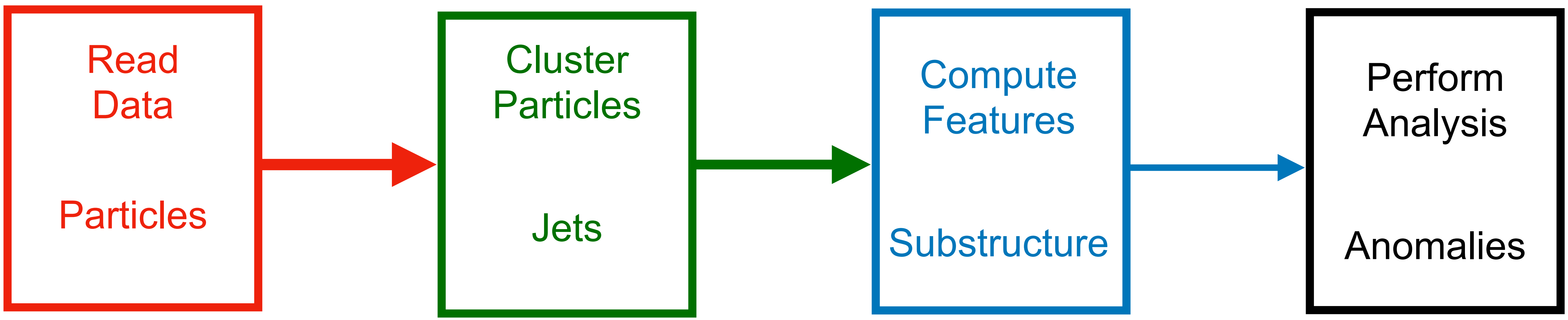
E-mail: matthias.schott@cern.ch



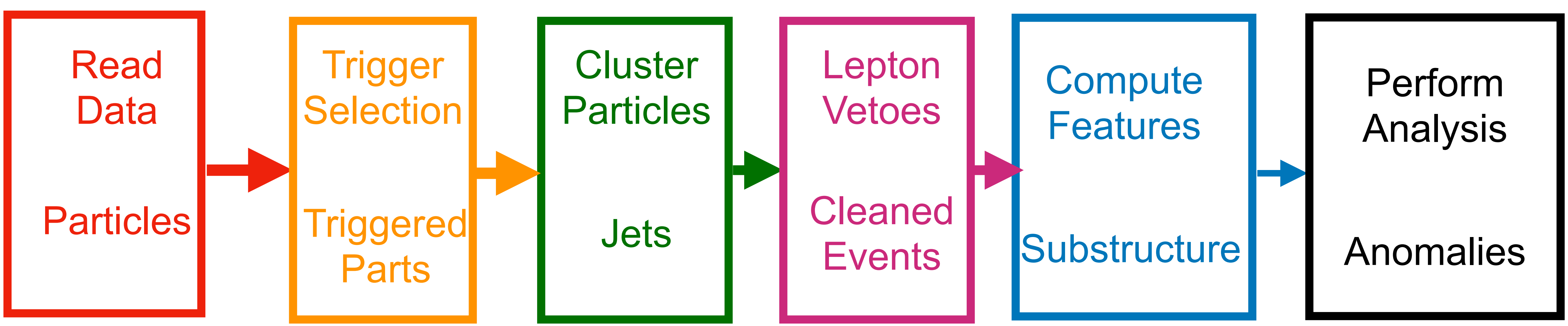
An Aside on Open Data

Processing Data

- To get from particles to analysis follow standard tool flow



Toy Data : Olympics Workflow



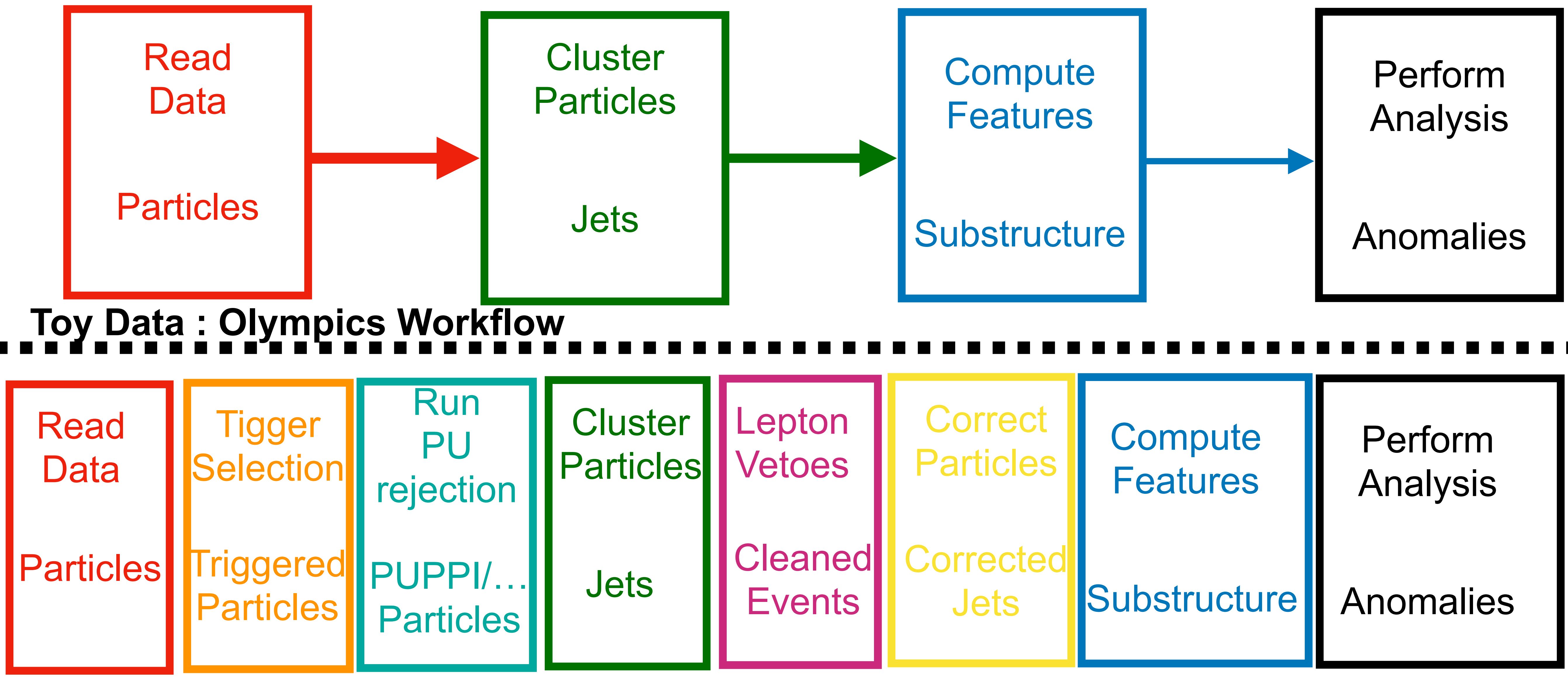
Real Data : Minimum Workflow

Why the extra steps?

- Going to real data a number of effects need to be considered
  - Data needs to pass a well defined/measured trigger
    - Bias or inclusive selection can introduce peaks
  - Sample needs to be close to pure QCD to emulate toy data
    - Processes like ttbar, W+jets will contribute significantly
- In reality, there are several more steps
  - Above steps constitute a minimum to emulate olympics

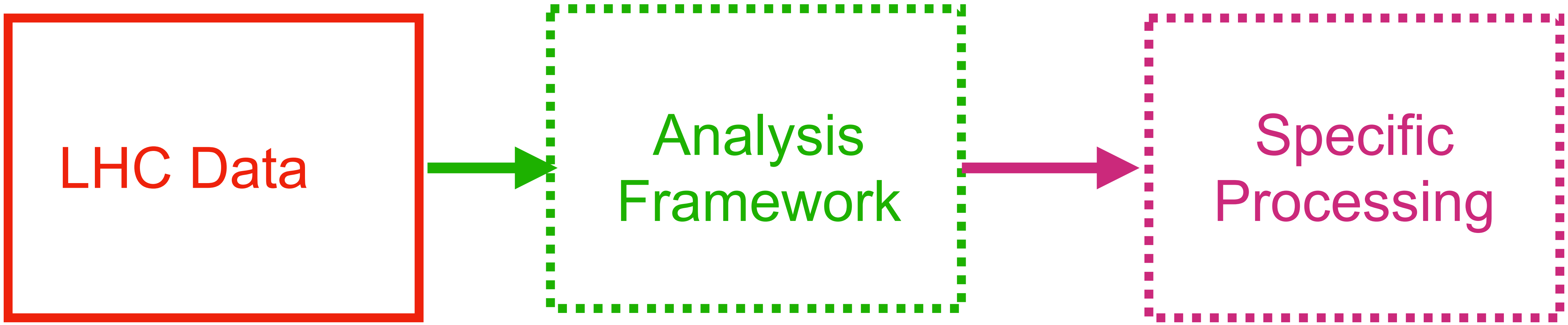
Processing Data

- To get from particles to analysis follow standard tool flow

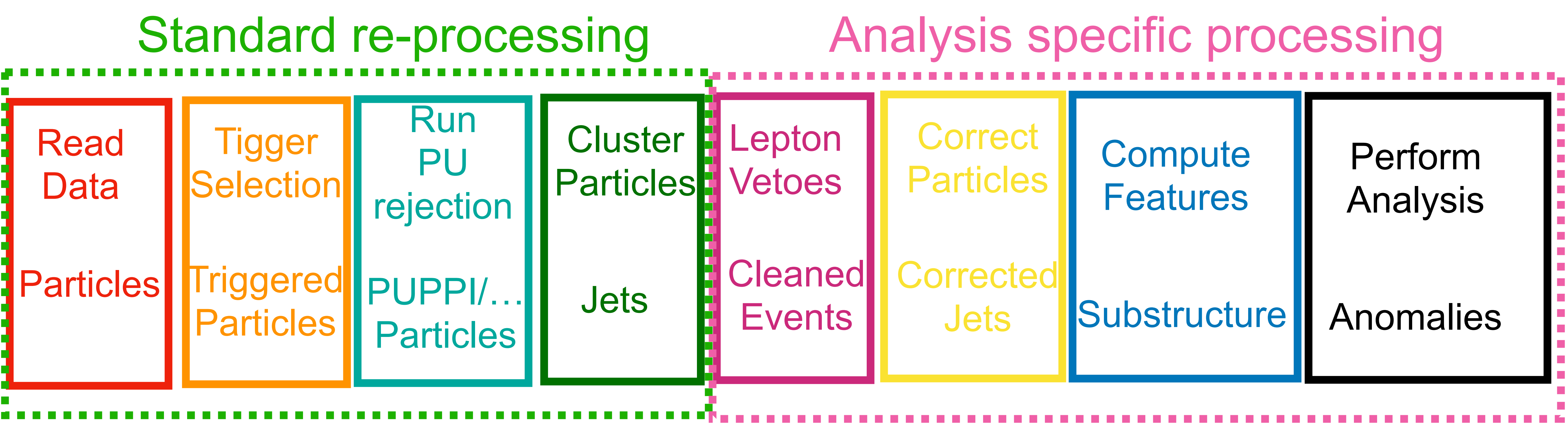


Real Data : Minimum Workflow

How is this usually done?



- Split is typically done to limit the amount of re-computing



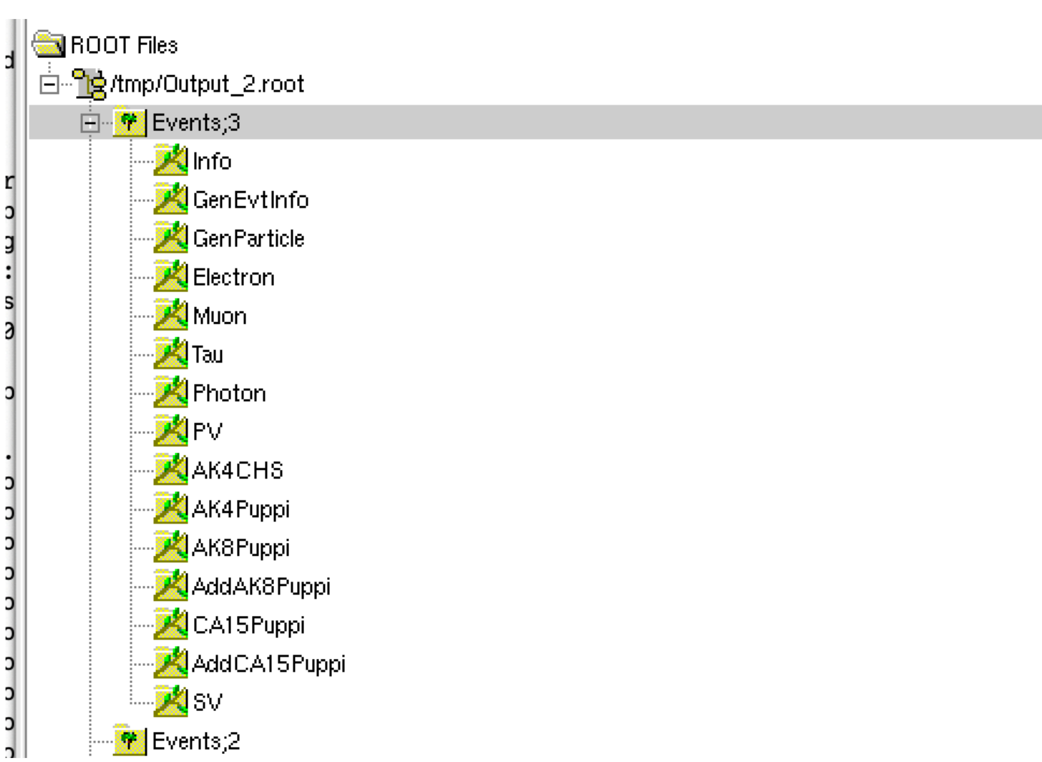
Real Data : Minimum Workflow

Building an Analysis FWK

- Frameworks take a long time to build
  - Complicated steps to follow careful curation of the data
  - Many iterations to avoid bugs in code
  - Data formatting what to keep a complex decision
- When preparing data for open analysis worked to get flat ntuple
- Collaborations have taken steps to centralize this
  - Newer data formats embed standard corrections
  - These data formats starting to be avaible in open data

Towards Regularization

- Bigger biases/corrections eventually embedded in software
  - In CMS: MiniAOD => NanoAOD
  - These are light smaller frameworks that lead to fast analysis
  - Still don’t solve all problems



Other things Lost

- Certain aspects in the data requires insider knowledge
  - Trigger preparation/Trigger biases
  - Which detectors were misfired
  - Details to address these issues are often complicated
- How do you deal with understanding inside knowledge?
  - Talk to others doing data analysis
  - Inside the collaboration many of these are well known

Examples Approaching

- Example sample approaching toy data
    - Special MC simulation sample used for Higgs tagging here
- Discussion on FAIRness of CMS open data here
    - Consensus is that this is close, but could be better
- Samples are are converted to h5 inputs

Dataset semantics

Variable	Type	Description
event_no	*UInt_t*	Event number
npv	*Float_t*	Number of reconstructed primary vertices (PVs)
ntrueInt	*Float_t*	True mean number of the poisson distribution for this event from which the number of interactions in each bunch crossing has been sampled
rho	*Float_t*	Median density (in GeV/A) of pile-up contamination per event; computed from all PF candidates of the event
sample_isQCD	*Int_t*	Boolean that is 1 if the simulated sample corresponds to QCD multijet production

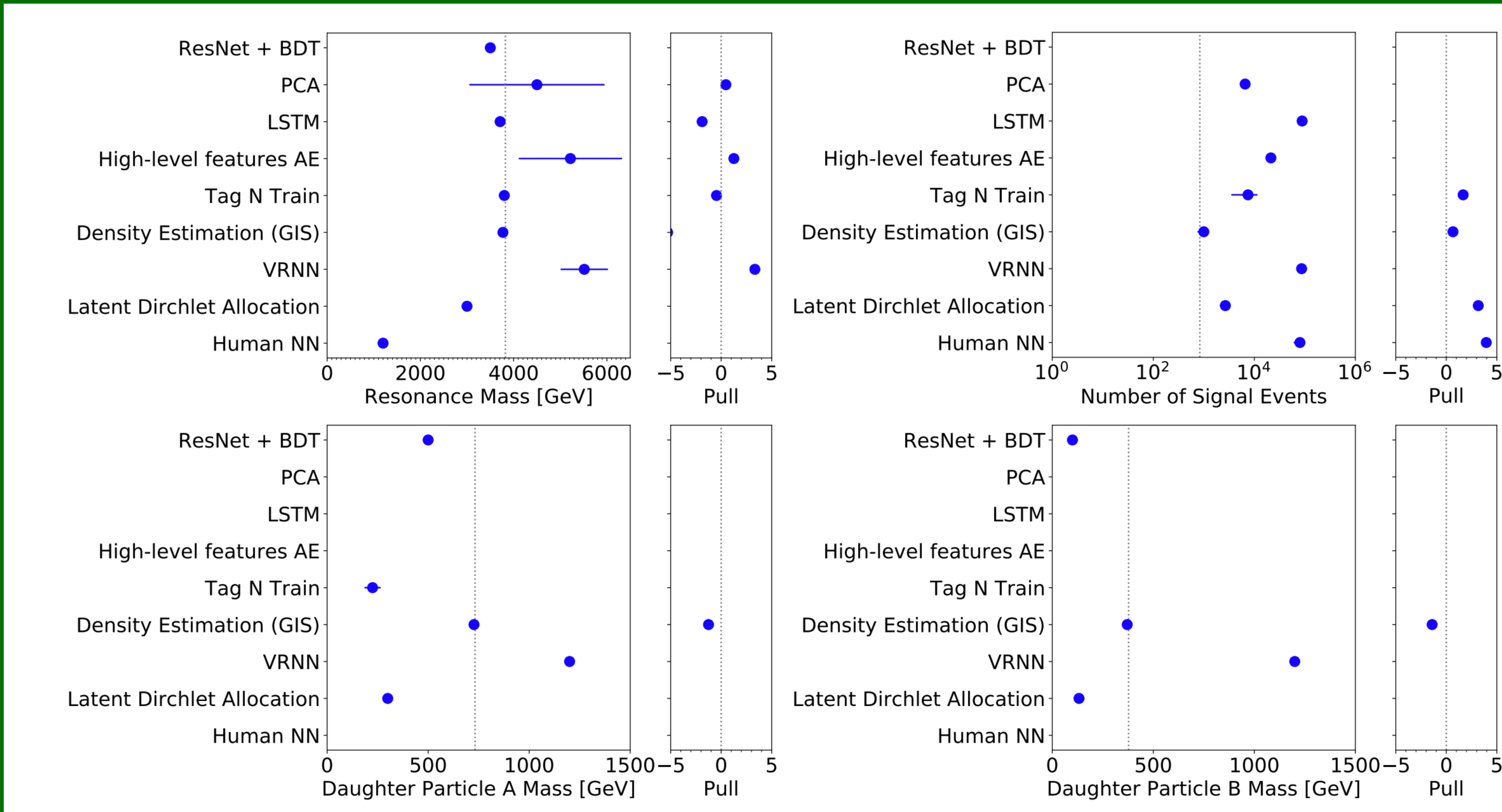
Future of Datasets is the FAIR convention

FAIR

- **F**indable
  - Resources easy to find to by both humans+computers
  - Metadata readily available; allows for the discovery of interesting data
- **A**ccessible
  - Resource and metadata can be easily accessed and downloaded
  - Both locally by a human, but also machines using standard protocols
- **I**nteroperability
  - Metadata should be ready to be exchanged, interepreted and combined in a semiautomated way with other datasets by humans and computers
- **R**euseability
  - Data and metadata are sufficiently well described to allow data to be reused
  - Proper citation must be facilitated and conditions should be valid to machines

Black Box 1

A fun look at results



- Nobody found an excess in black box 3
- Black box 2 was empty

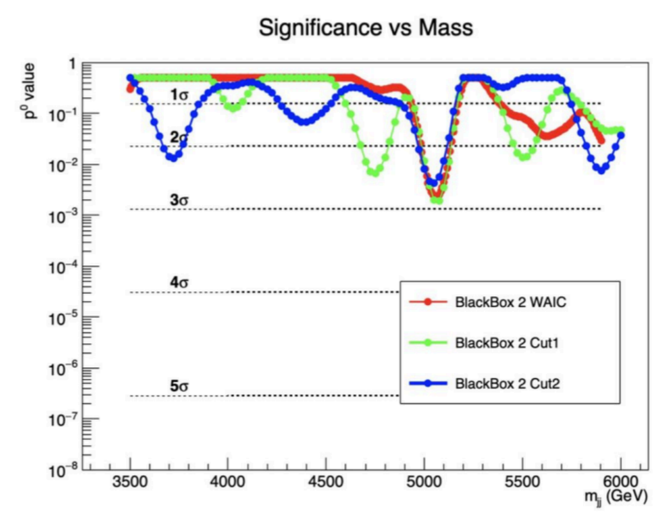
Black Box 2

Black Box 2 - Predictions

- **PCA on high-level features (old):**
  A > BC with B > jj and C > jj
  m(A)=**4800** +- 100 GeV, m(B)=725 GeV, m(C)=125 GeV
  p-value / Signal events: 0.00764 / 89
- **VRNN (old):**
  A > BC with B > jj and C > jj
  m(A)=**4422** +- 722 GeV
  p-Value: 0.229181609 / Signal events: < 12k
- **Embedding clustering:**
  Z' resonance with mass **4600** GeV +- 17 GeV decaying to 2 jets
  p-Value: 0.0396 (1.8 sigma) / Event count: 76 +- 28
- **Latent Dirichlet Allocation (old)**
  Our method extracts signal descriptions which appear convincing, however the classifier does nc identify a bump in the invariant mass spectra. Without this we were unable to determine that signal was present. The di-jet description extracted consisted of one jet of mass 350-400 GeV an another of mass 150-200 GeV. If the production of these states was non-resonant, we would b unable to find the signal with our method. Or if more than just di-jets were relevant to reconstruc the invariant mass, we would also not be able to find it. Otherwise, ***we determine that no signa was present in the data.***

Reminder no signal

- **QUAK:**
  BB2 3sigma local evidence for resonance at ~ 5 TeV



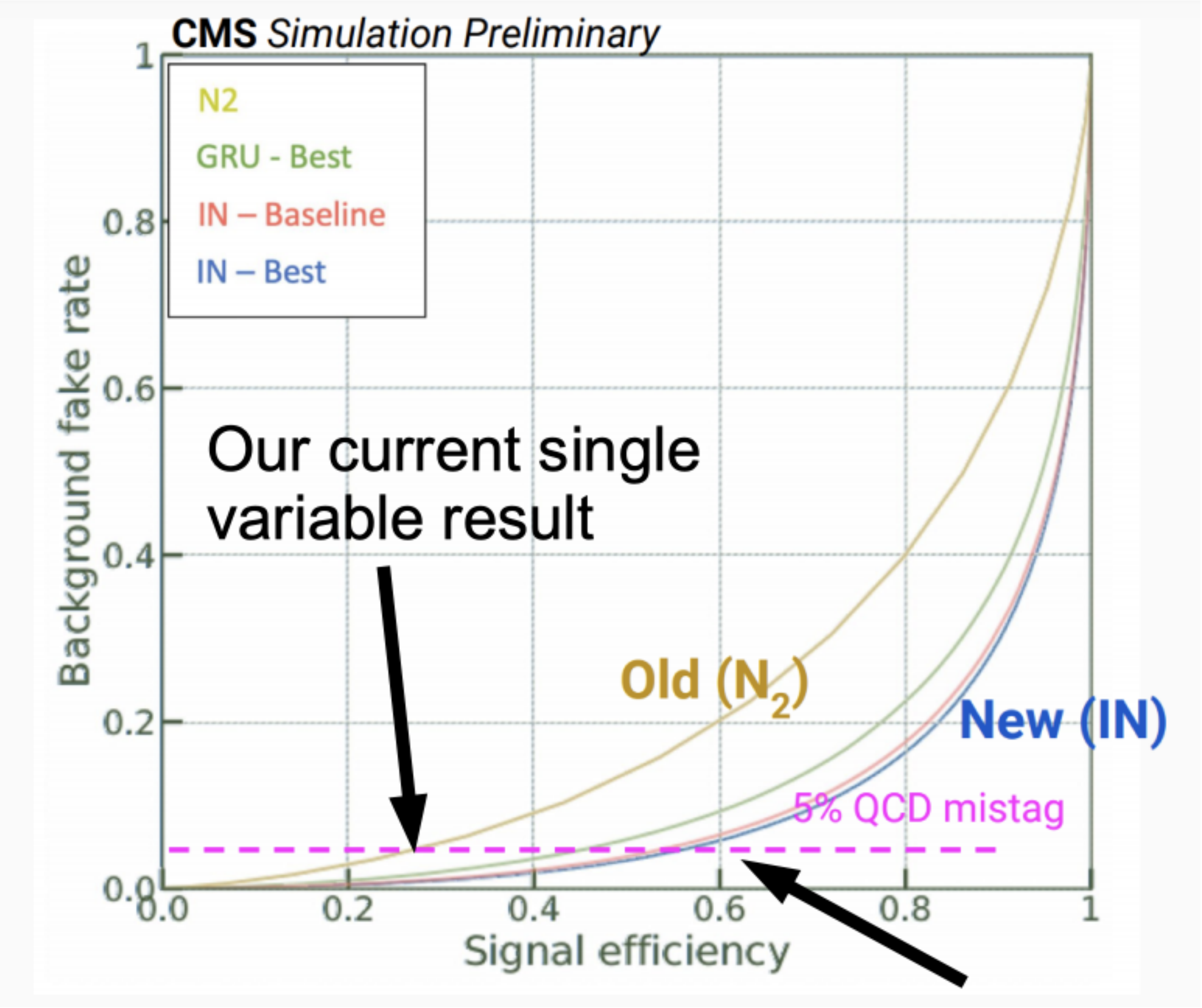
Sang Eon Park, LHCO 2020

- **M-flows and GAN-AE:**
  work in progress (inconclusive)
- **VRNN (new):**
  Hint of an excess at 4.2 TeV

Observations

- There is no catch all solution
  - Many of the best approaches combine multiple ideas
  - A diversity of approaches helps robustness
- LHC Olympics focused on resonant processes
  - Non-resonant processes make background extraction harder
  - Can we deal with complex topologies (such as black box 3)
- Data processing pipeline is assumed to be offline reconsturction
  - Could envision some approach in the triggers
- How can we actually compare sensitivities if we don't have a model?

Edit me

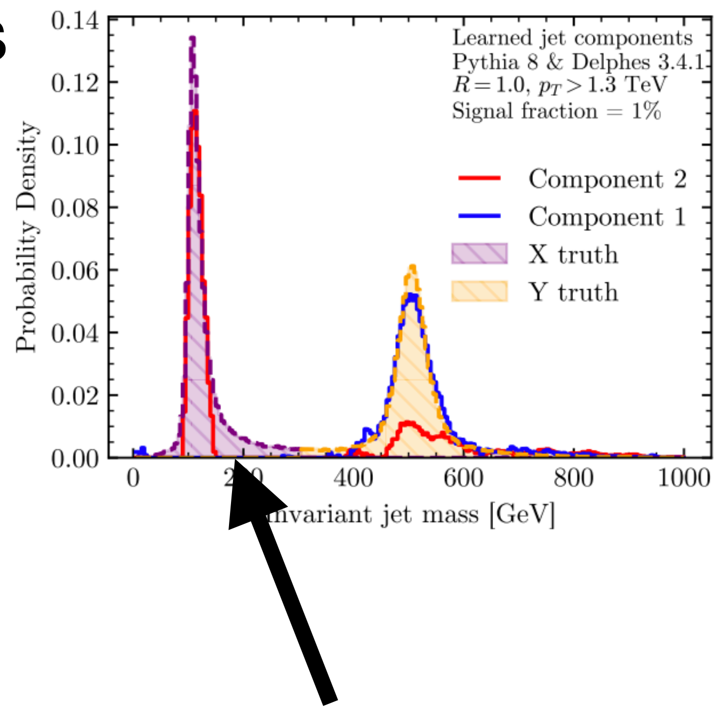

CMS Simulation Preliminary
N2
GRU - Best
IN – Baseline
IN – Best
Our current single variable result
Old (N_2)
New (IN)
5% QCD mistag
Background fake rate
Signal efficiency
0.0
0.2
0.4
0.6
0.8
1

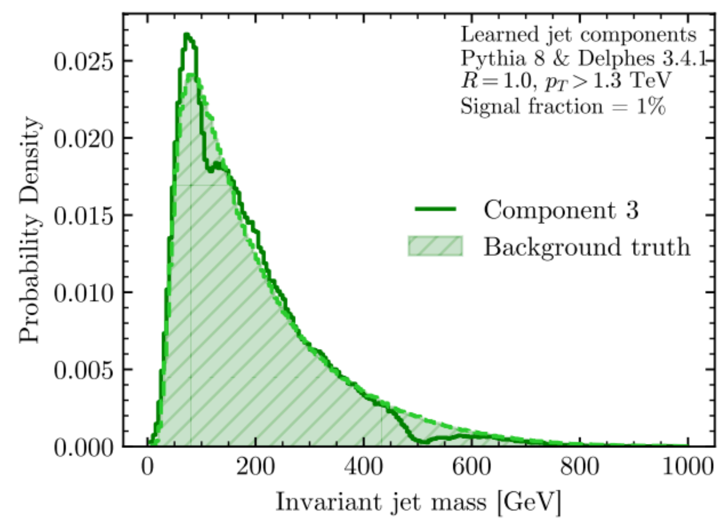
Factorized Topics

Inputs: Jet mass of each jet

R&D Dataset

- Factorization: each jet mass distributions can be factorized
- QCD composition is the same for leading and subleading





Use leading and trailing jet masses to make “topics”

Solve for the jet mass 1 and 2 that yield 3 distinct categories

Signal Extraction : None (did not work on BB1)

Take Away: Breaks down with small signal

LDA

Inputs: Jet splittings from declustering

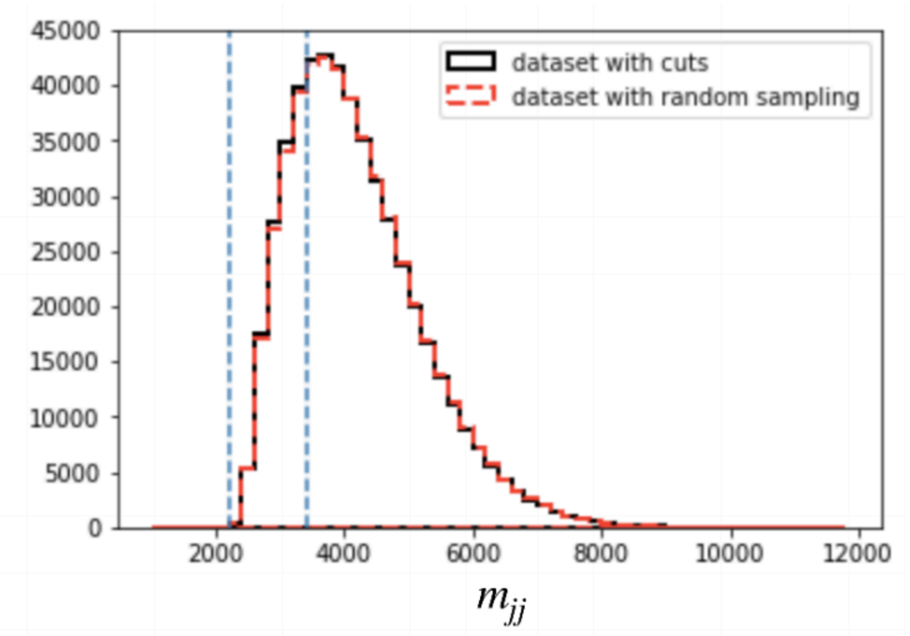
- Latent Dirichlet Allocation (LDA)
  - Decluster jet and use splitting info
  - Construct 2 hypotheses in data
    - LDA minimization to get 2

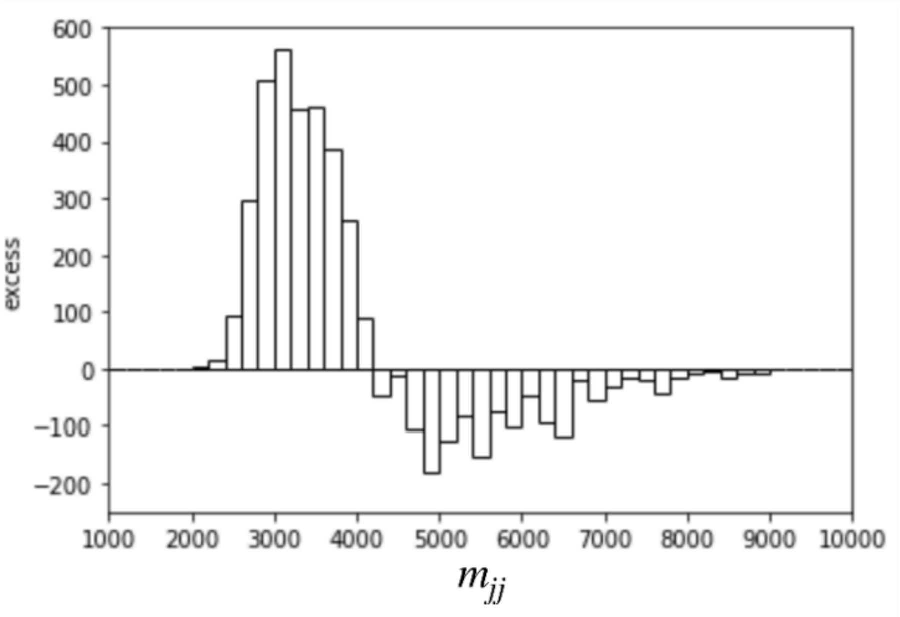
Compute likelihood of two hypothesis to be consistent

$$L(o_1, \ldots, o_N|\alpha) = \prod_{i=1}^{N} \frac{p(o_i|\hat{\beta}_1(\alpha))}{p(o_i|\hat{\beta}_2(\alpha))} .$$

Signal Extraction : None (did not work)

BB1 Dataset





Take Away: LDA benefits from many event observables