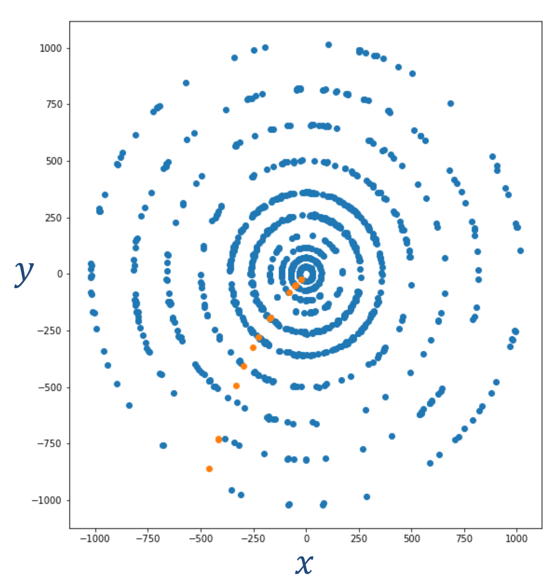
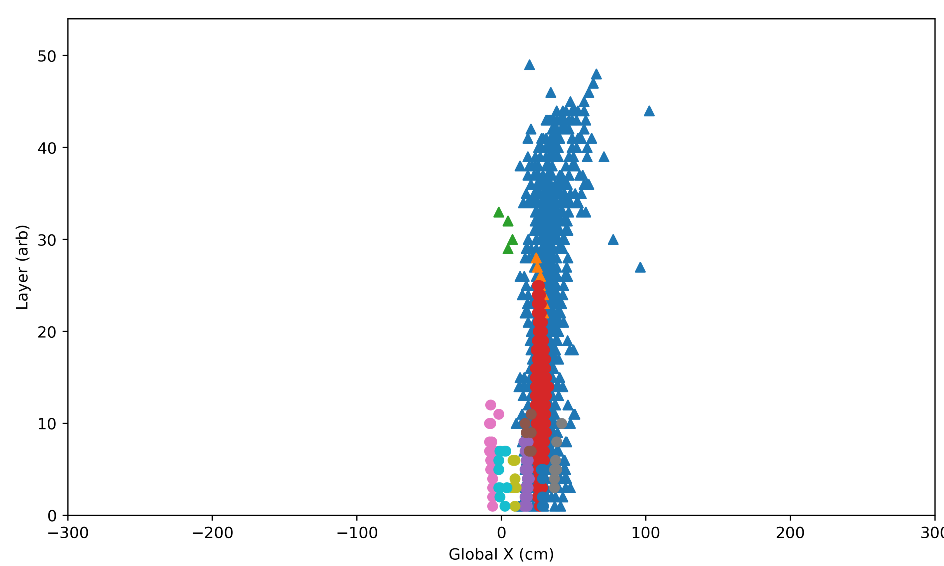
Graph Neural Networks for Pattern Recognition in Particle Physics

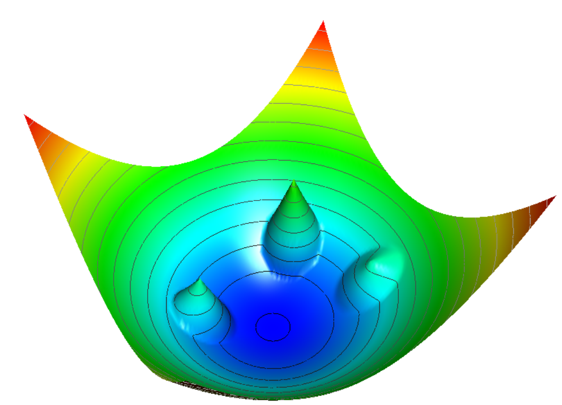
Lindsey Gray

MPML Seminar

14 October 2020







Overview

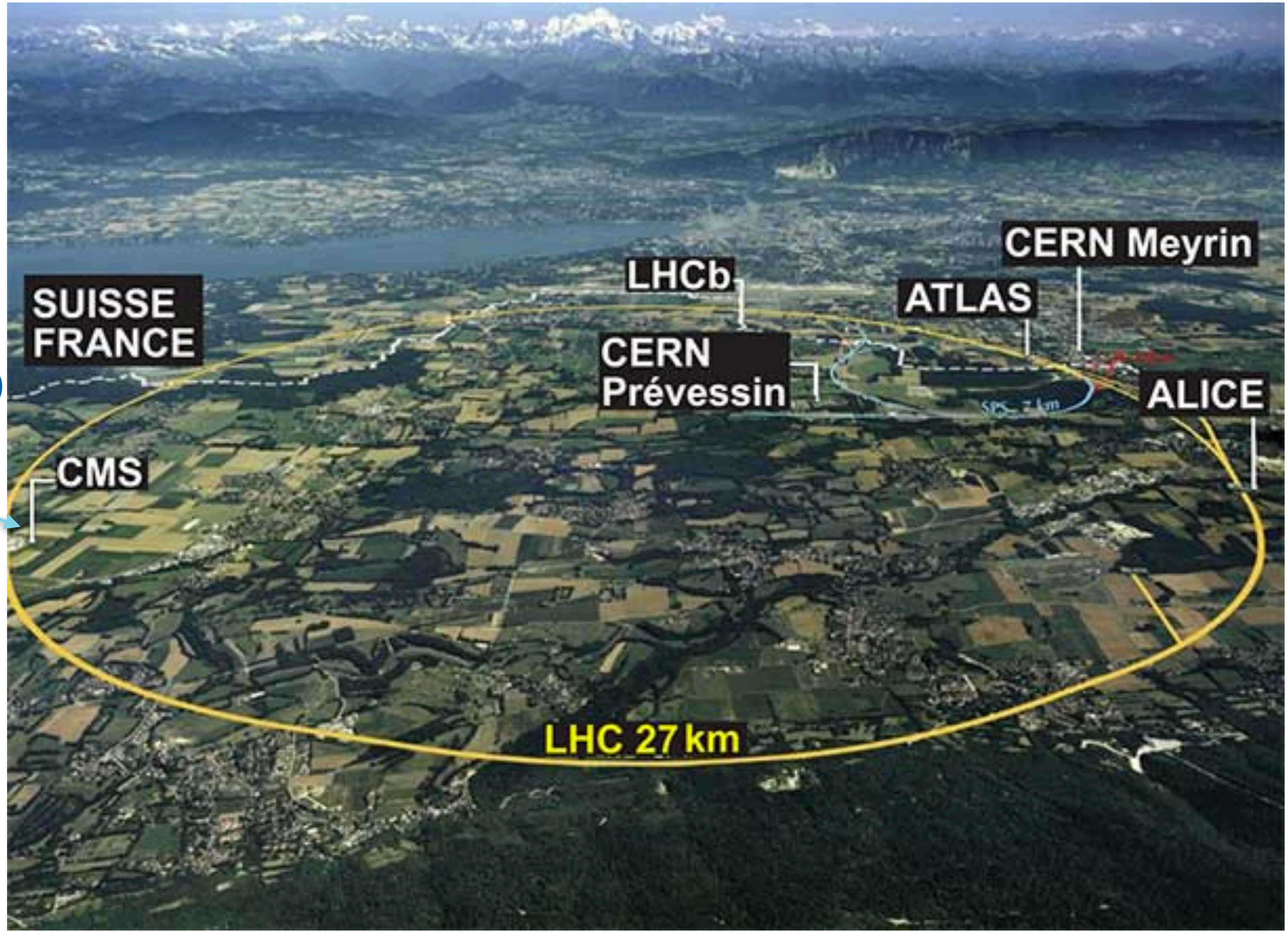
- Introduction
  - to the LHC the CMS Detector and their upgrade programs
  - to the basics of machine learning and a touch of its history within particle physics
  - to the pattern recognition problems that need to be solved in modern particle physics

- Turning particle physics pattern recognition into a learnable task
  - The relationship of pattern recognition algorithms to graphs
  - Graph neural networks as engines for pattern recognition
  - Turning operations on graphs into particle physics reconstruction tasks
  - Examples of successful differentiable reconstruction algorithms
  - A pause to discuss the limitations of these methods

- Where to go next?
  - Achieving tiny networks with meaningful loss constructions and activation functions
  - Avenues for applying this to particle physics pattern recognition
  - Concluding remarks

A Brief Introduction to the Large Hadron Collider (LHC)

Compact
Muon
Solenoid (CMS)
The experiment
I work on



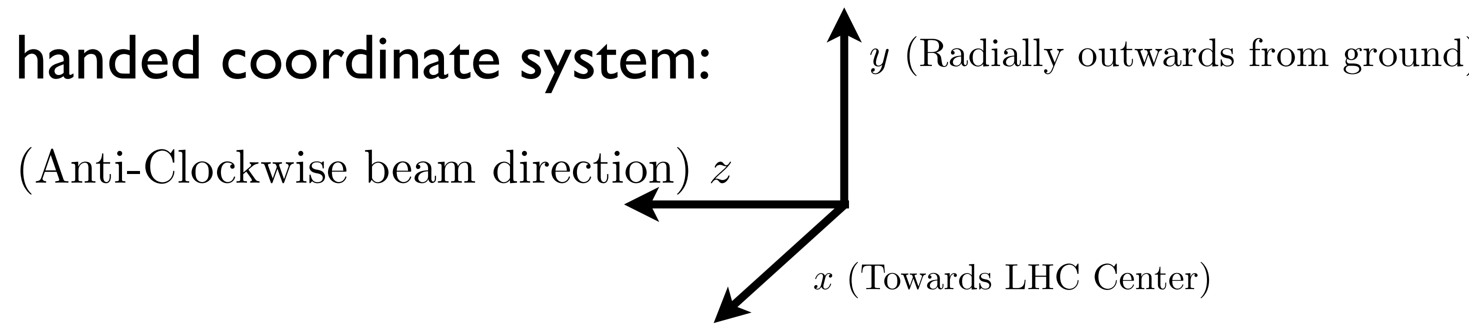
Two counter-rotating beams of bunched protons each with an energy of ~7 TeV.

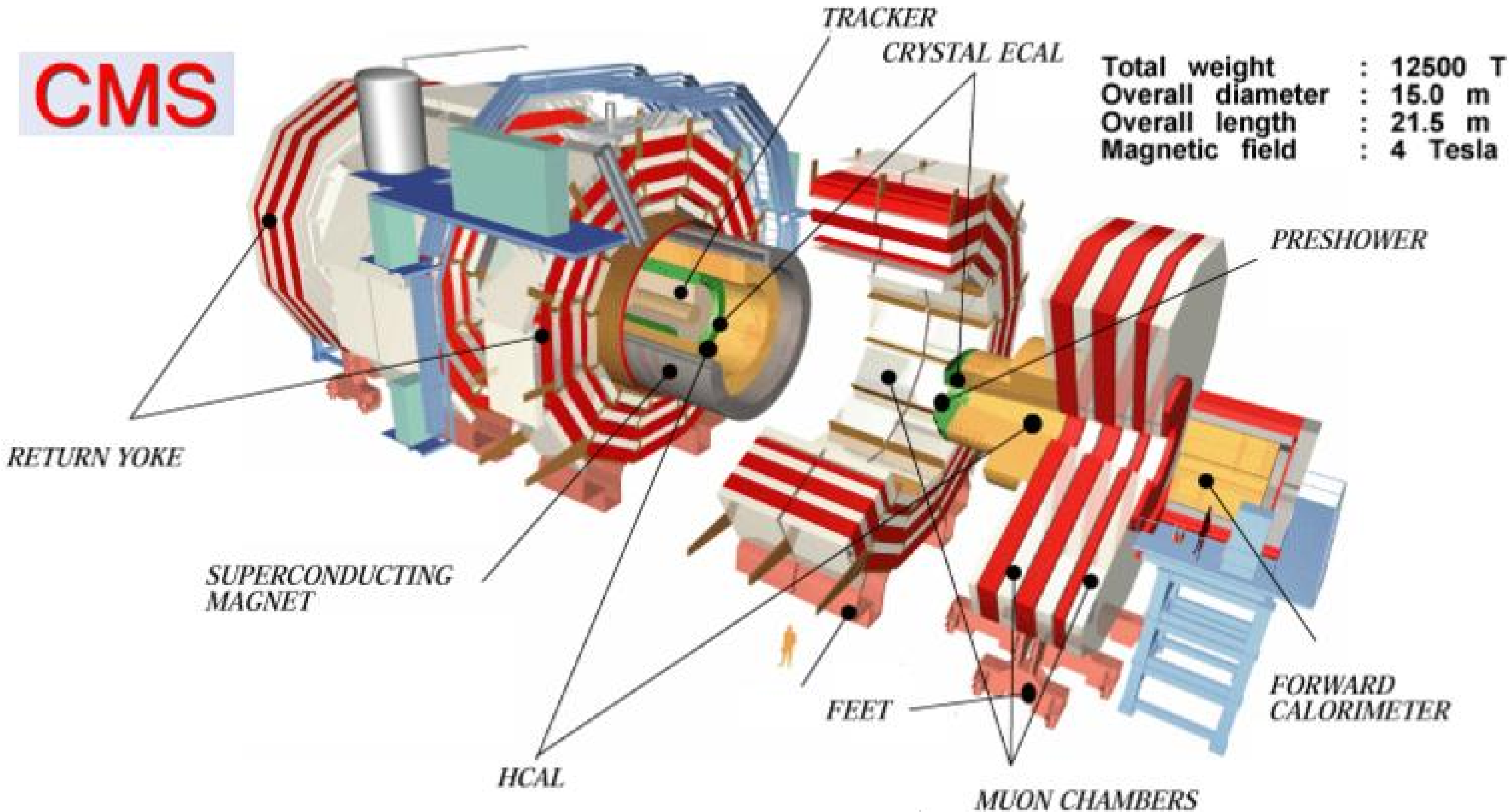
Bunches pass through each other at a rate of 40 MHz, we record a few kHz of that.

The CMS Detector

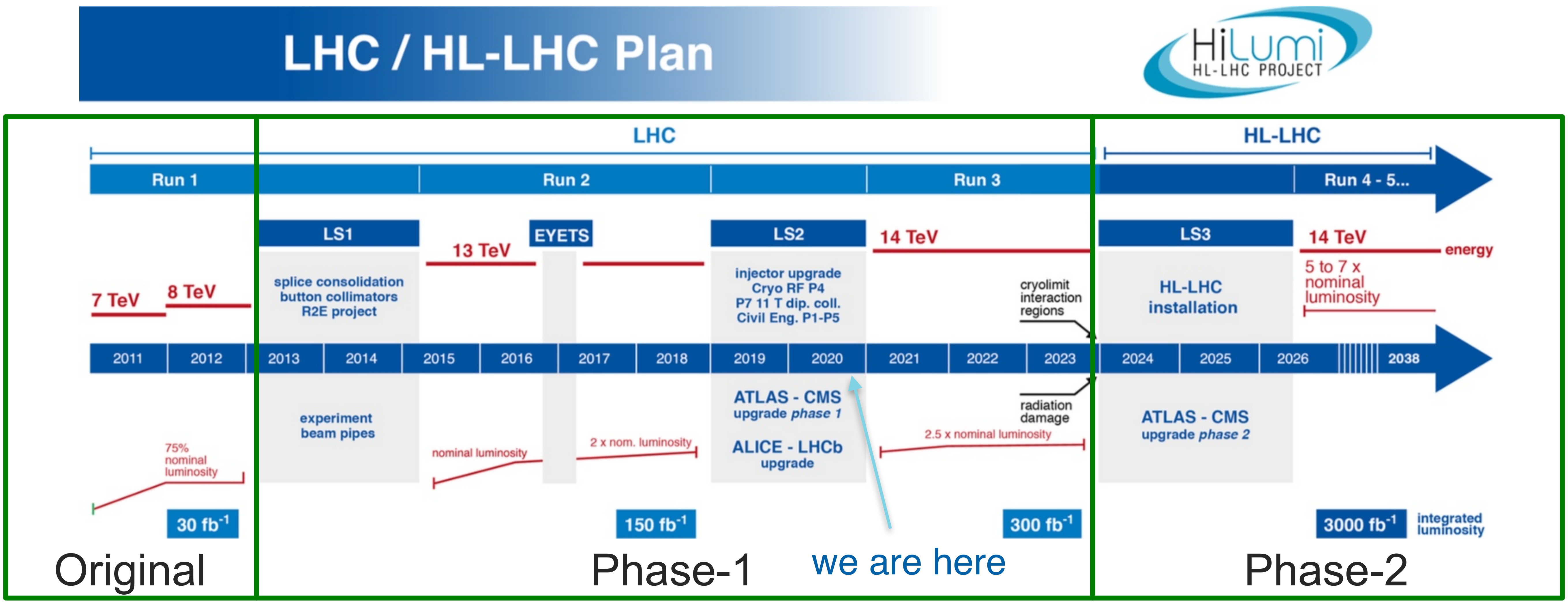
$$\eta = -\ln\tan\frac{\vartheta}{2}$$

Right handed coordinate system:



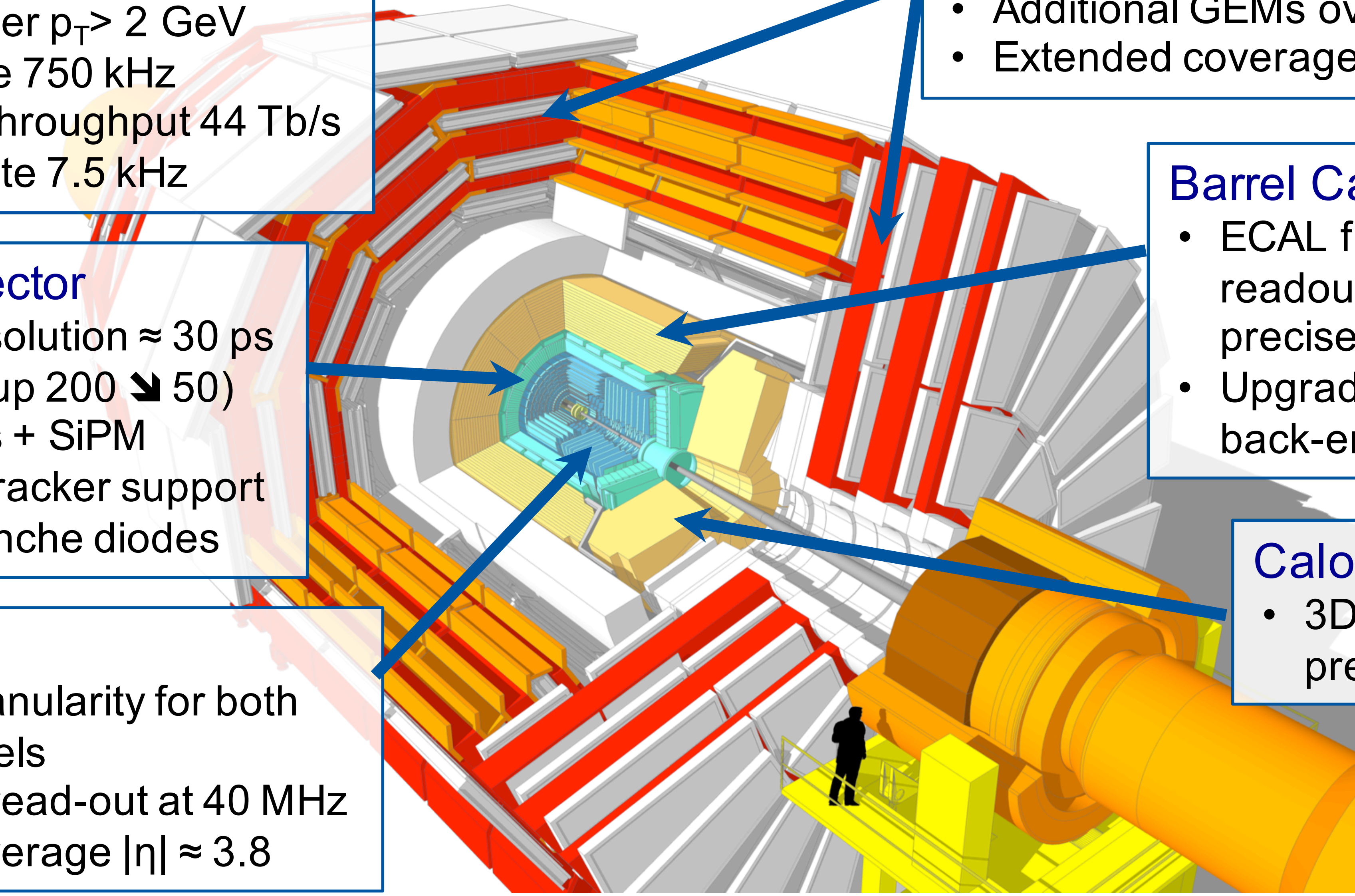


Upgrading the LHC to the High Luminosity LHC



- LHC will be upgraded to deliver 5-7x more luminosity in the mid 2020s
  - Driven by new physics objectives to measure detailed properties of the Higgs Boson,
  - By end of Phase-1 there will be significant radiation damage to sub-detectors throughout CMS, and the upgraded accelerator delivers an even more challenging environment
- The accelerator and experiments will all need to be retrofitted and upgraded to approach this challenging 10 year task.

Corresponding Upgrades to CMS



Trigger & DAQ
- L1 Track trigger $p_T > 2$ GeV
- L1 accept rate 750 kHz
- DAQ design throughput 44 Tb/s
- HLT output rate 7.5 kHz

MIP timing detector
- Target time resolution ≈ 30 ps (effective pile-up 200 ➘ 50)
- Barrel: crystals + SiPM embedded in tracker support
- Endcap: avalanche diodes

Tracker
- Increased granularity for both strips and pixels
- Strip tracker read-out at 40 MHz
- Extended coverage $|\eta| \approx 3.8$

Muon systems
- New DT & CSC front- & back-ends
- Additional GEMs over $1.6 < |\eta| < 2.4$
- Extended coverage to $|\eta| \approx 3$

Barrel Calorimeters
- ECAL full crystal granularity readout at 40 MHz with precise e/γ at 30 GeV
- Upgraded ECAL & HCAL back-ends

Calorimeter endcap
- 3D shower topology with precise timing

- To deal with this increased luminosity the CMS detector is being significantly upgraded, improving its radiation tolerance and granularity.
- Utilizing this massive amount of data demands even more of our algorithms.

ML and Neural Networks in a Nutshell

- Goal: find the parameters w of a function F that best maps x onto y
  - Do this by minimizing a chosen “loss function”
  - ML is the set of numerical algorithms that solve this problem

- Neural Networks are a subset of these algorithms that are defined recursively from inputs to outputs
  - Any mapping function F can be approximated by a sufficiently large NN
- Since inputs are variable and relationship to output is learn, including new information is very straightforward, as is scaling up computation to more data

particle four-momenta

signal or background?

$$\vec{y} = F_{\vec{w}}(\vec{x})$$

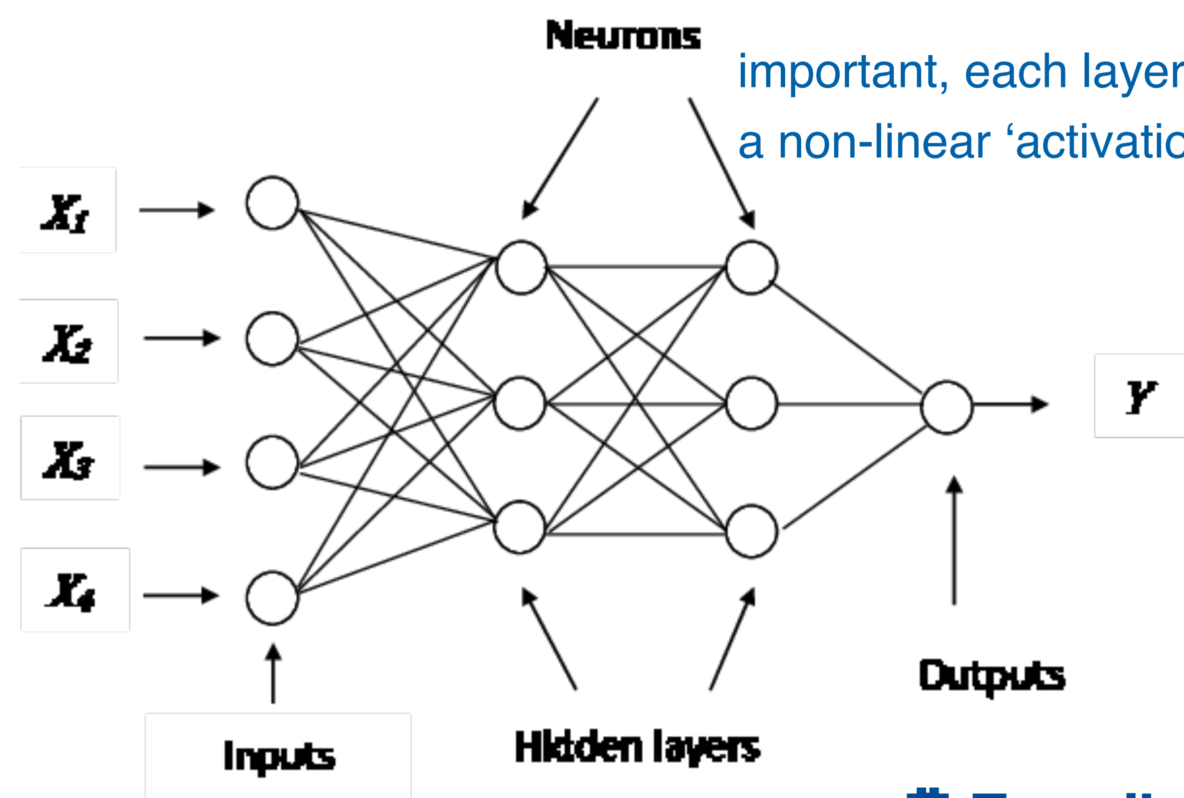
$\vec{x}$ input space

mapping function

weight space

$\vec{y}$ output space

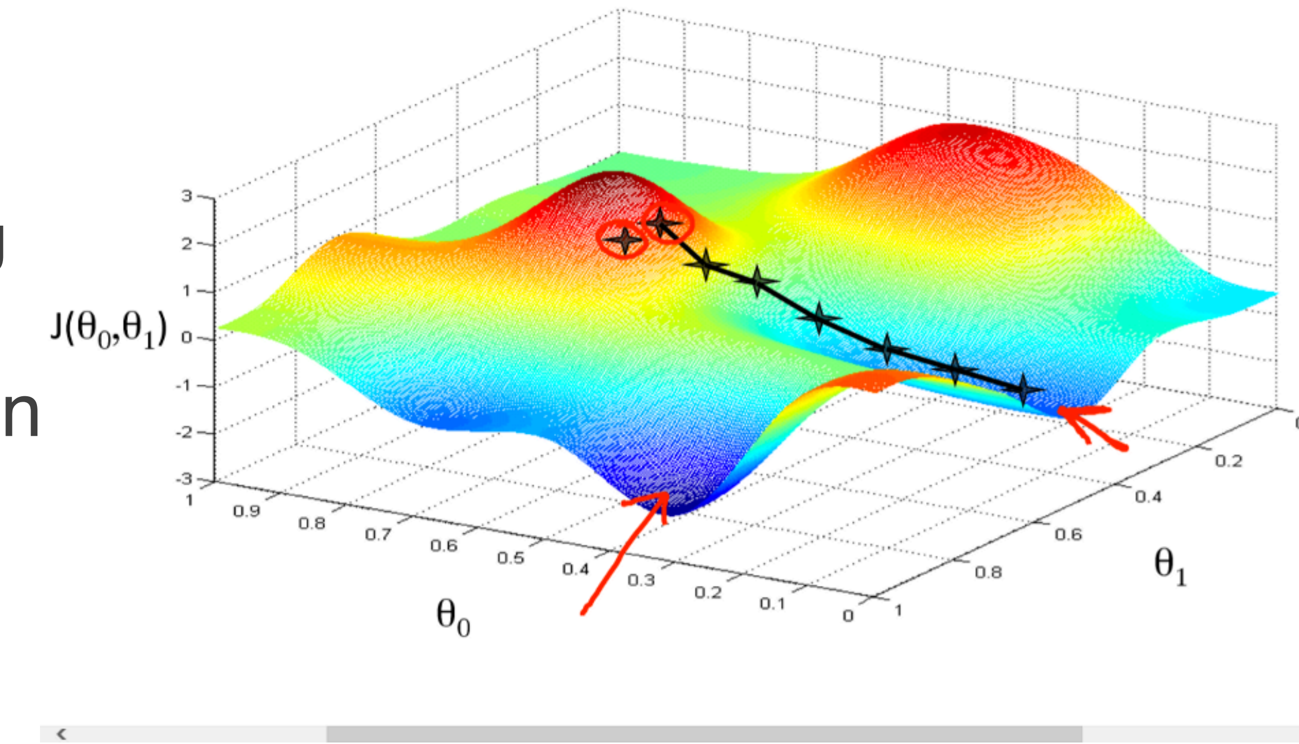
important, each layer has a non-linear ‘activation’!

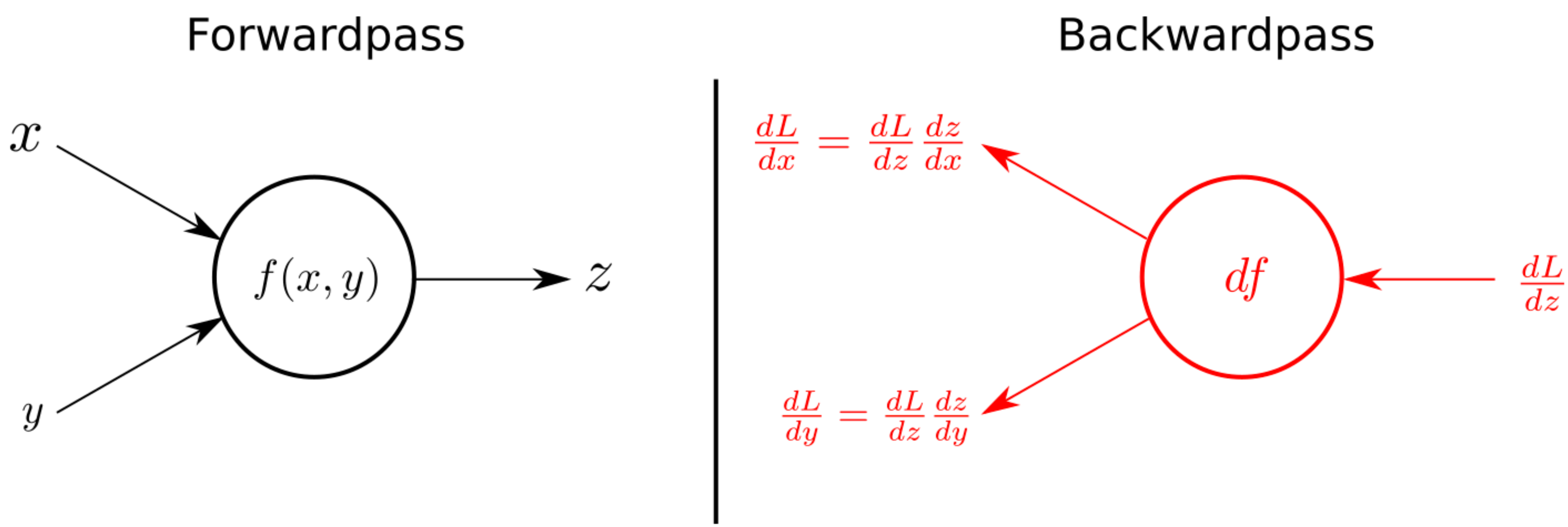


Fermilab

Backpropagation and training of neural networks

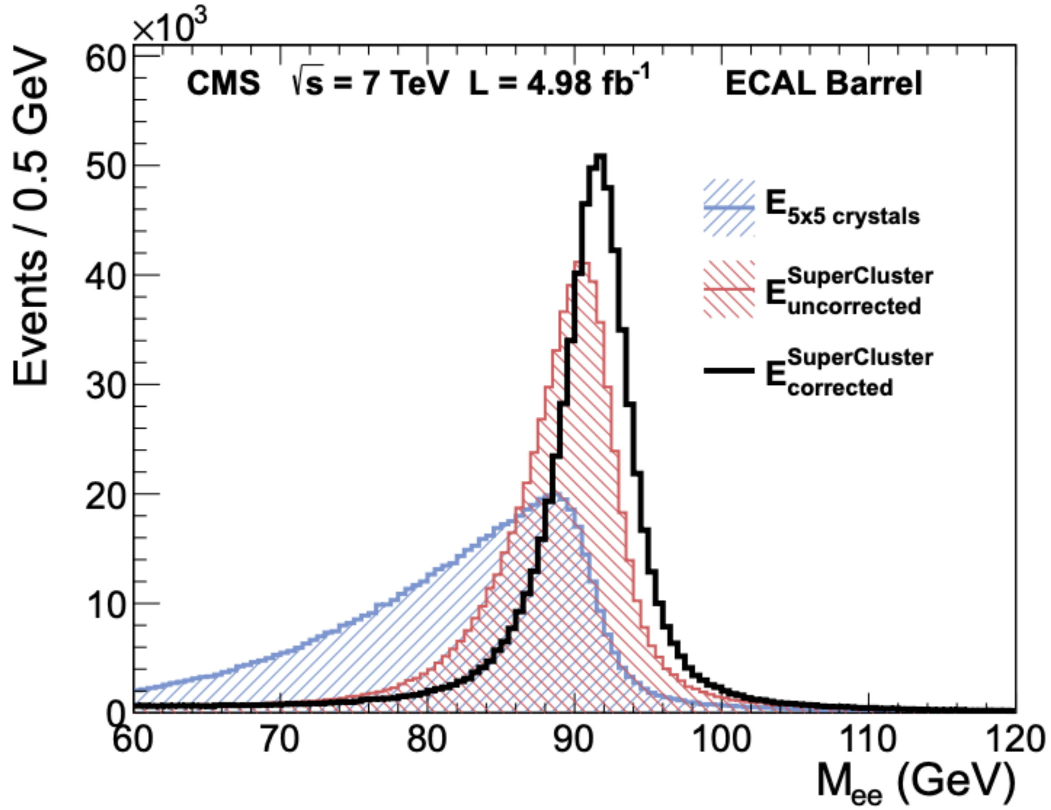
- Each node in a network makes some prediction
  - Each predictions error can be calculated using the chain rule
- Encode task to be done in a 'loss' function and minimize that loss
  - Recursively update parameters based on estimated error at each pass over the data

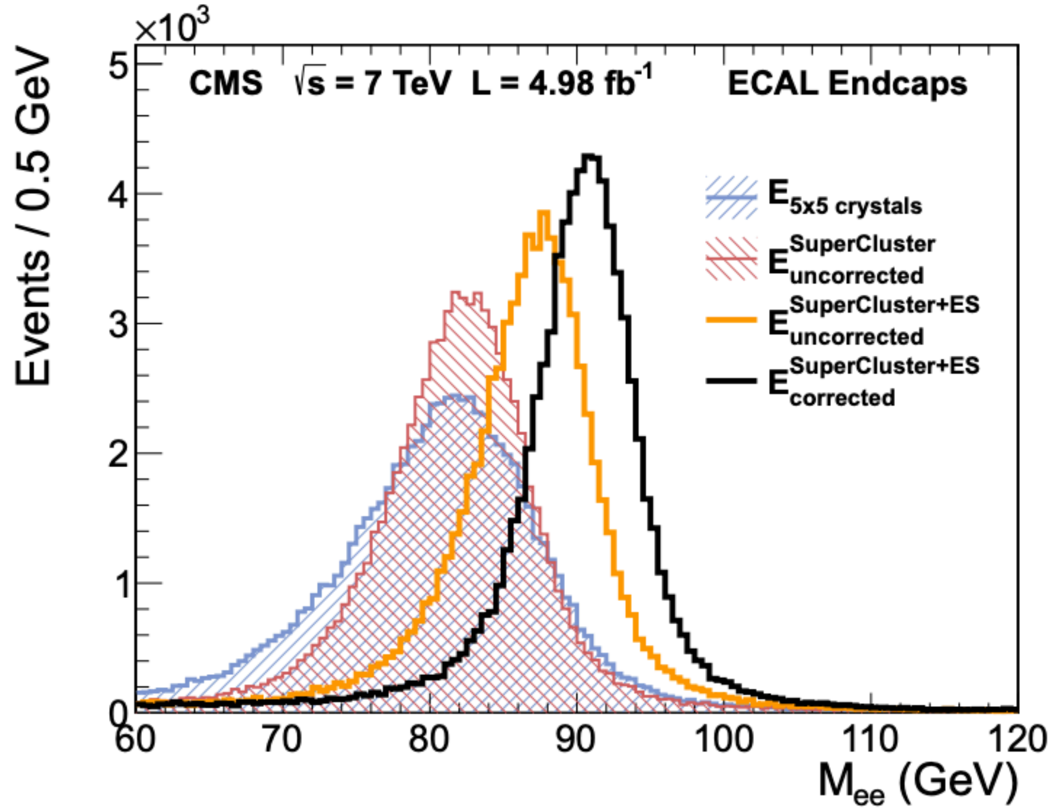




ML in Physics: Electromagnetic calorimeter energy corrections

- Boosted Decision Trees (BDTs) have had a long history of use within the CMS Collaboration
  - Relatively fast inference and training times before 2012
  - Functions by making progressively finer cuts in the input space
- Use average over numerous binary-cut based selections to generate a classifier
  - This can be used to discriminate categories or to regress quantities
  - Can handle position dependent corrections as in CMS ECAL, with enough data





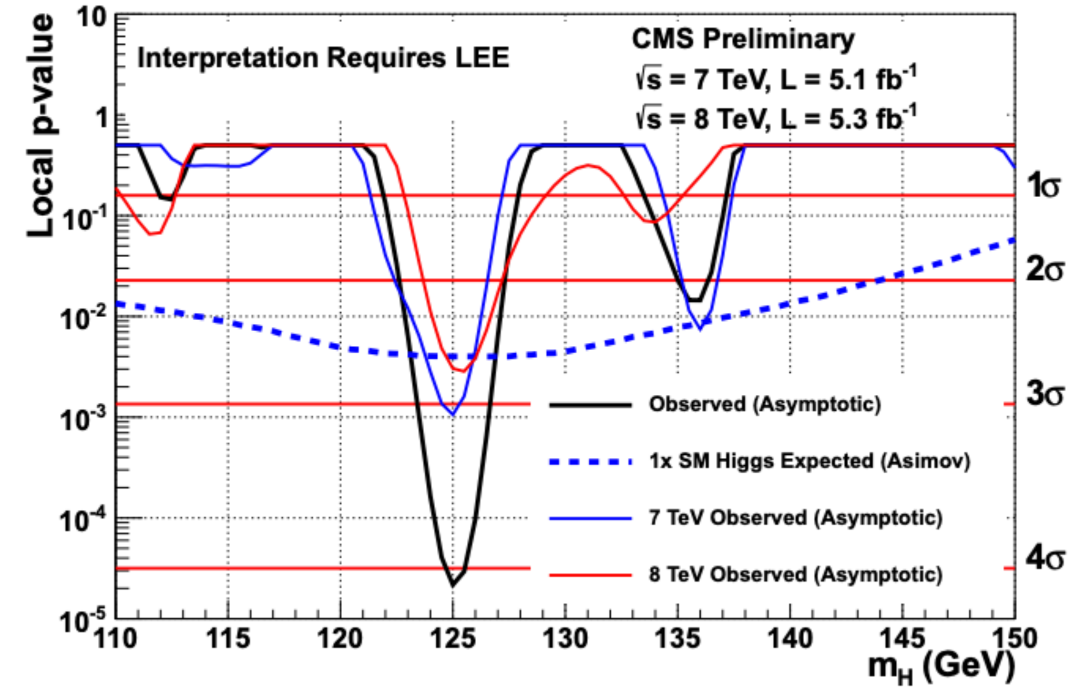
Fermilab

Higgs to γγ Discovery

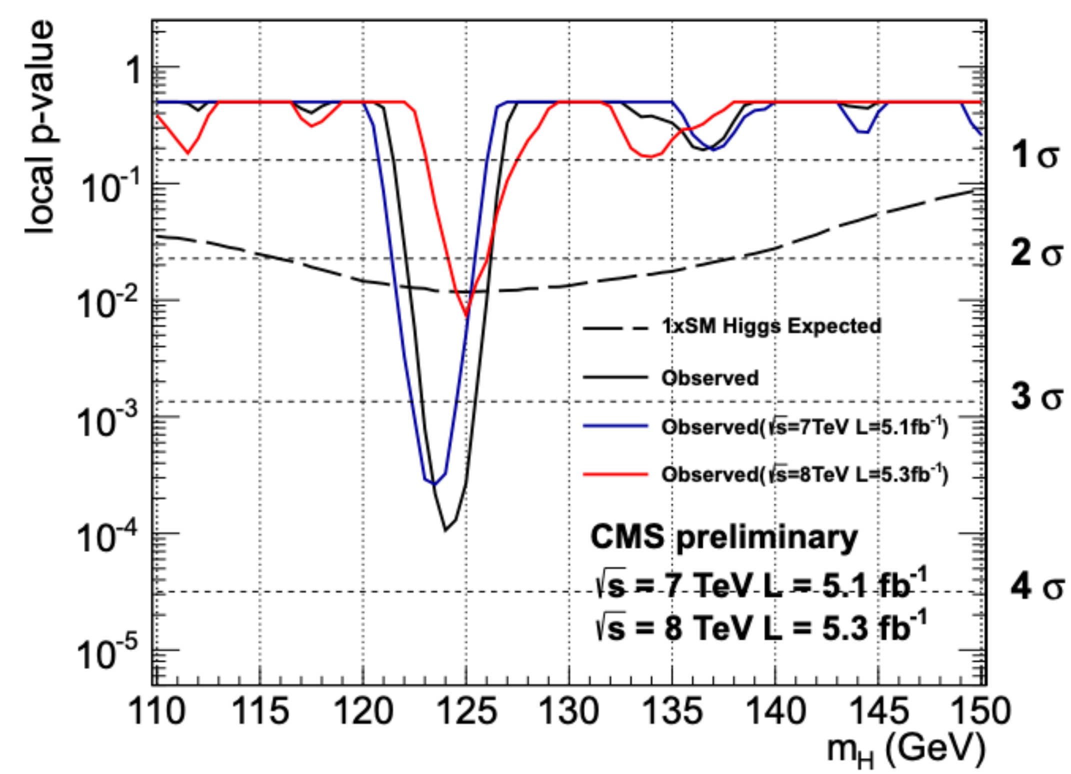
- Usage of ML techniques led to an analysis workflow that is easier to describe and maintain
  - Training based workflow instead of re-optimizing cuts by hand
  - Trade some abstraction for ease of use

- Improved sensitivity
  - At the cost of a lot of jokes about “BDTs all the way down”

- Demonstrable control of systematics related to multivariate modeling of the input data
  - This is now the status quo



(a) mass-fit MVA.



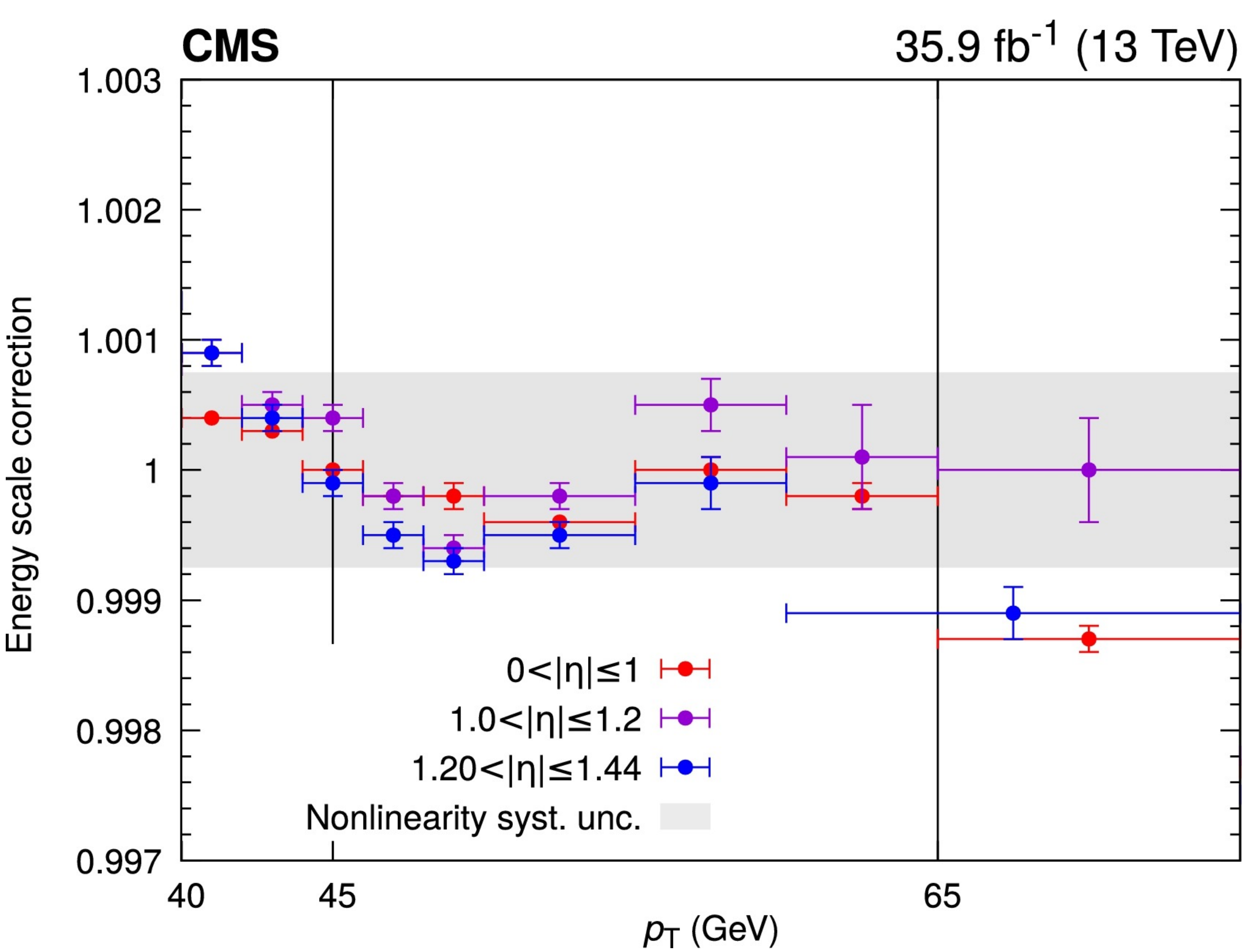
(b) Cut-based analysis.

http://cds.cern.ch/record/1460419

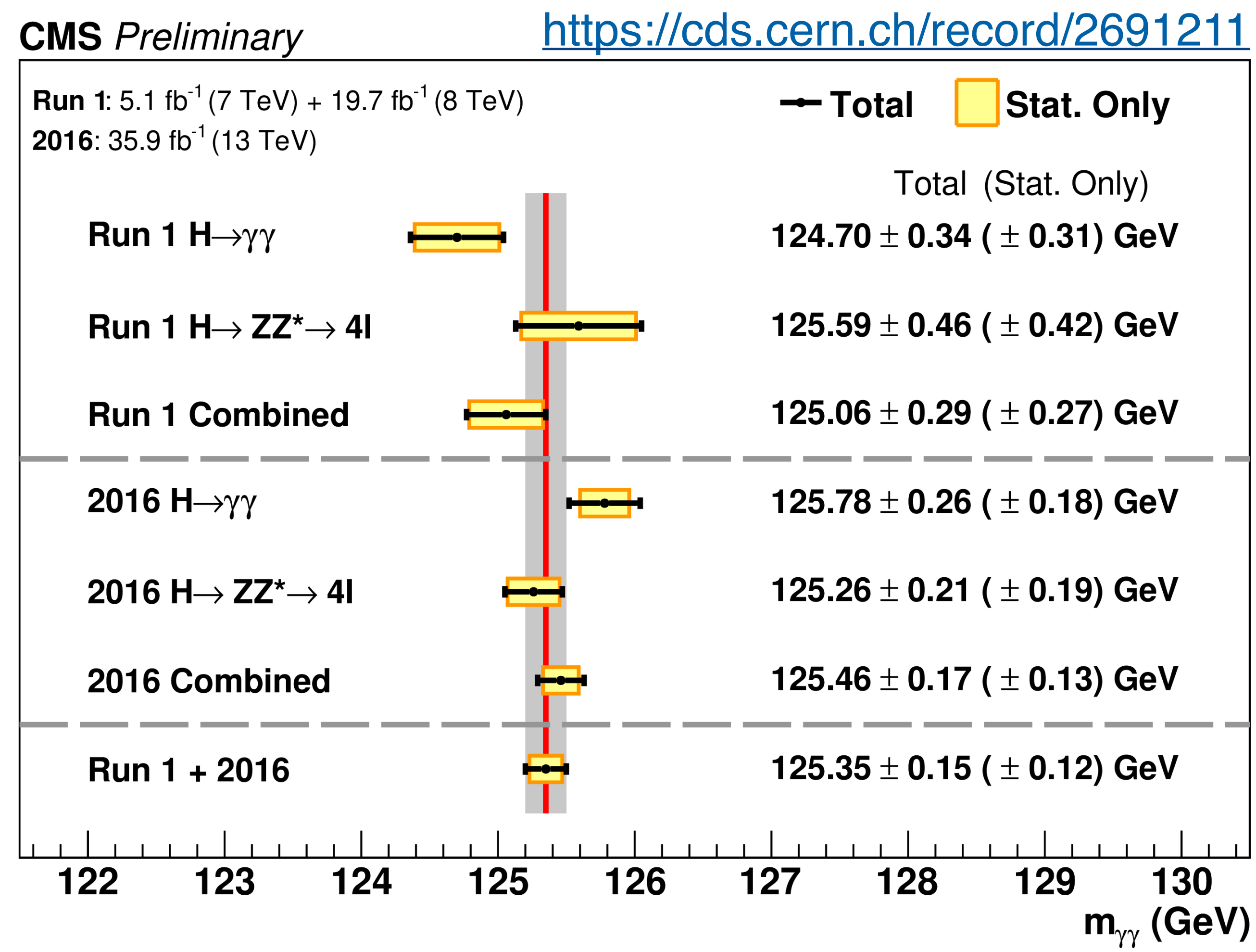
Fermilab

Current usage and performance of ML regression in CMS

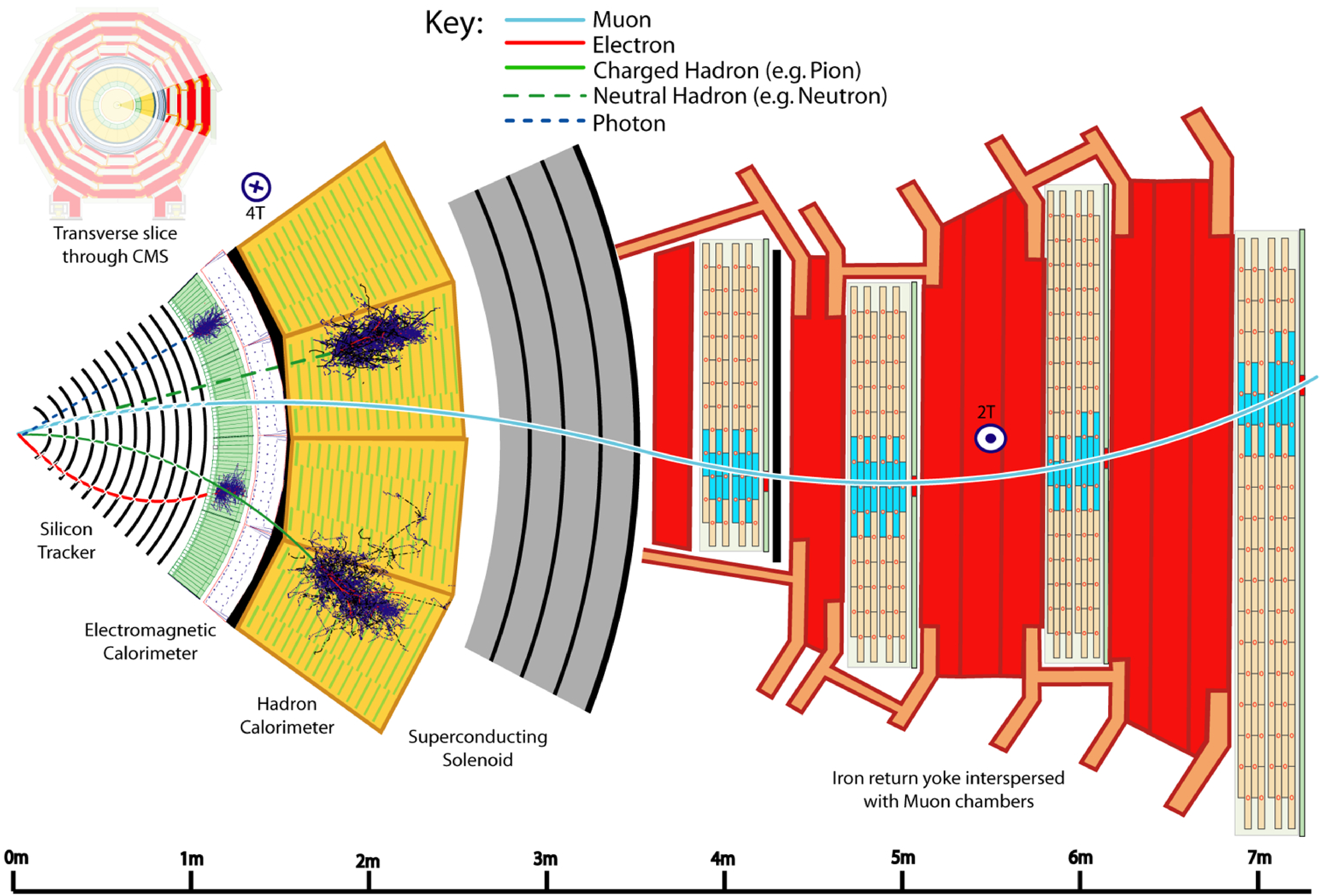
- Coming to modern times: the ML-based analysis and energy reconstruction is being used to perform precision measurements
  - Energy scale uncertainties for photons understood to ~0.1%
- ML-based regressions a critical piece for modern Higgs measurements!
- ML techniques are well-adopted in HEP, what more can we do with them?



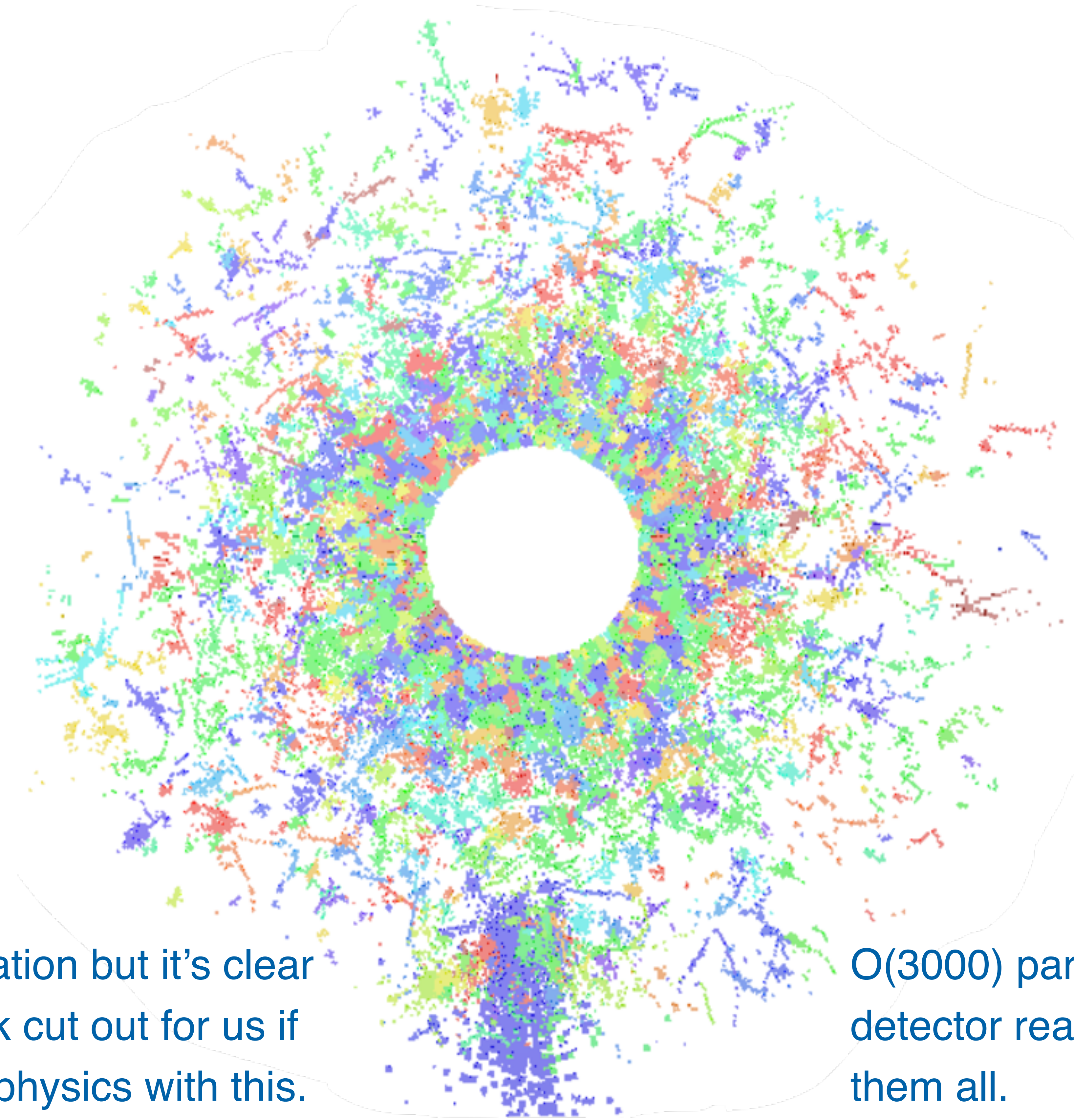
https://cds.cern.ch/record/2691211



Particle Detection in CMS



A view from the upgraded endcap calorimeter of the HL-LHC



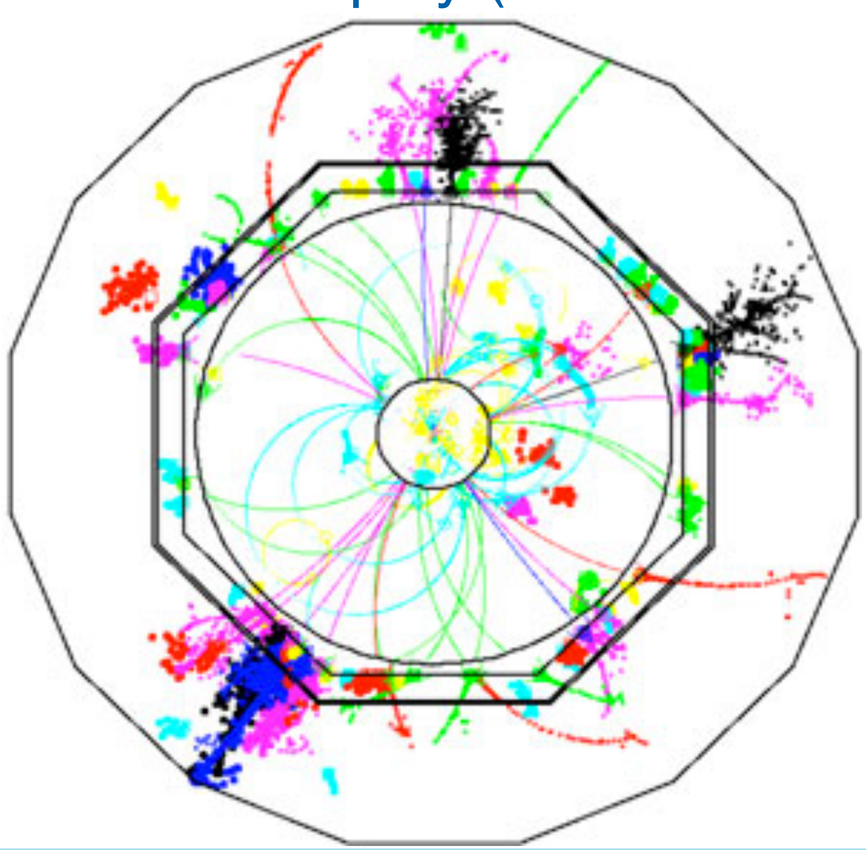
This is from simulation but it's clear we've got our work cut out for us if we're going to do physics with this.

O(3000) particles, 150k active detector readouts to describe them all.

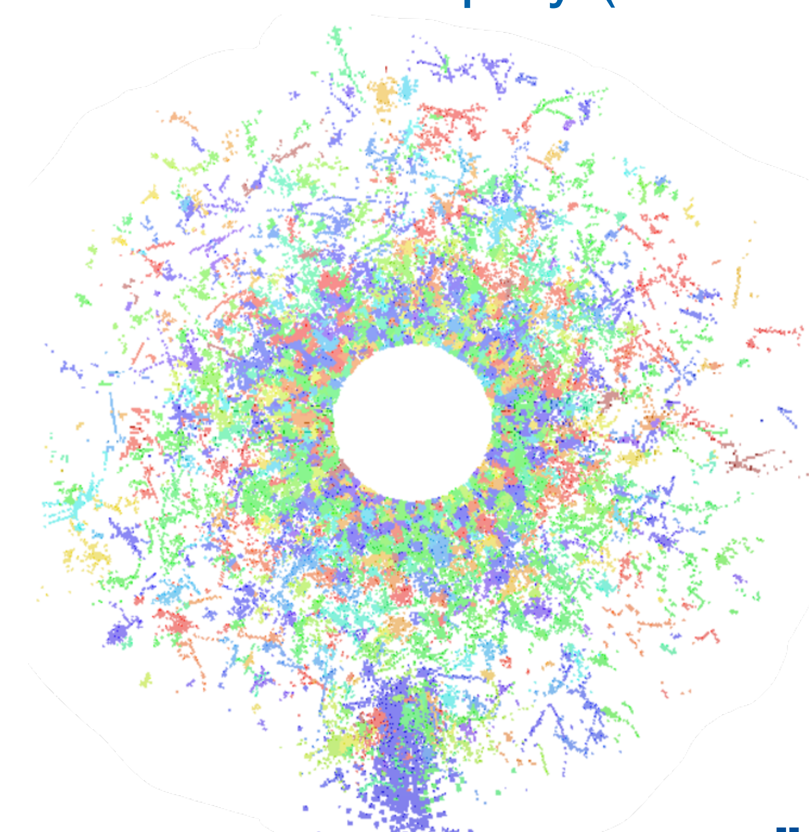
Modern detectors and data are significantly more complex

- Detectors are changing, they're becoming more larger, more granular
  - DUNE, the CMS High Granularity Calorimeter (HGCAL)
  - HL-LHC Trackers + Timing Detectors
- They're aiming for high performance in strenuous environments
  - ILD aiming for electron positron collider, HGCAL for HL-LHC
  - Readouts include precision timing information, but have to correlate x,y,z,t & E
  - Detector performance depends much more on algorithmic physics performance

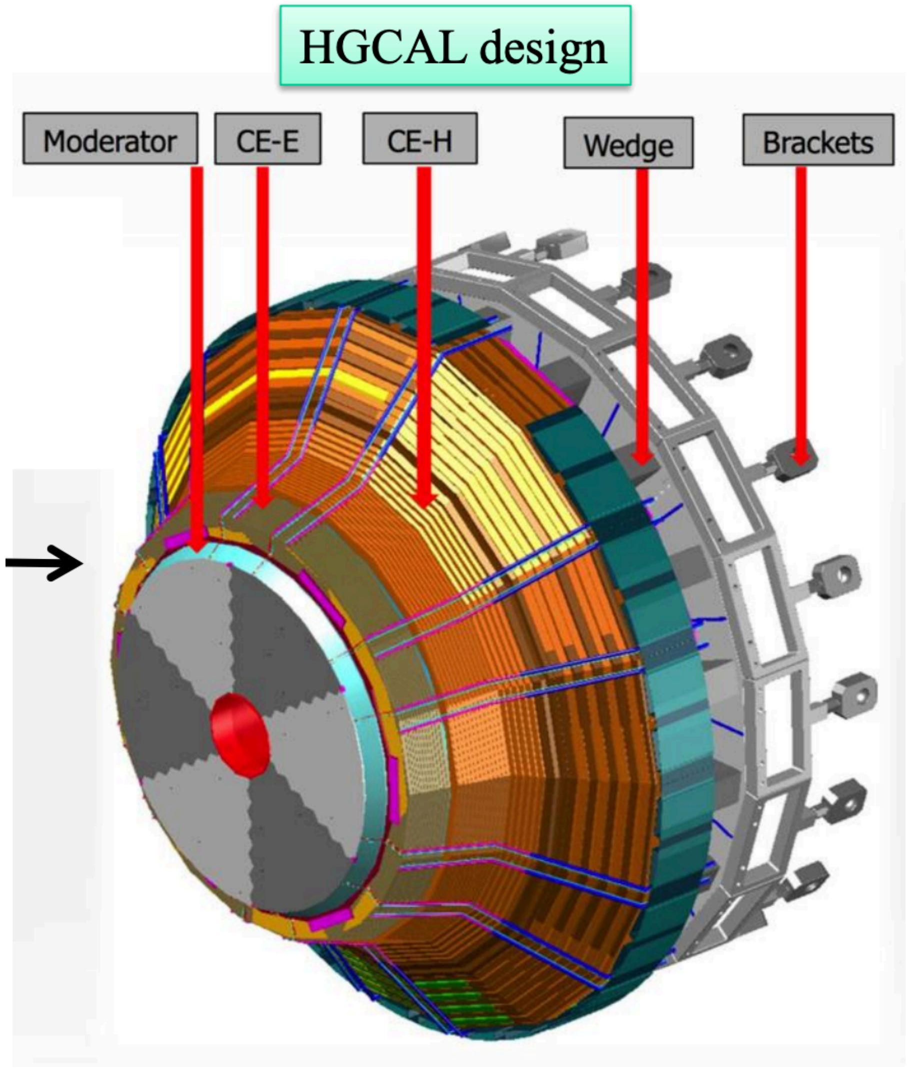
ILD Event Display (whole detector)

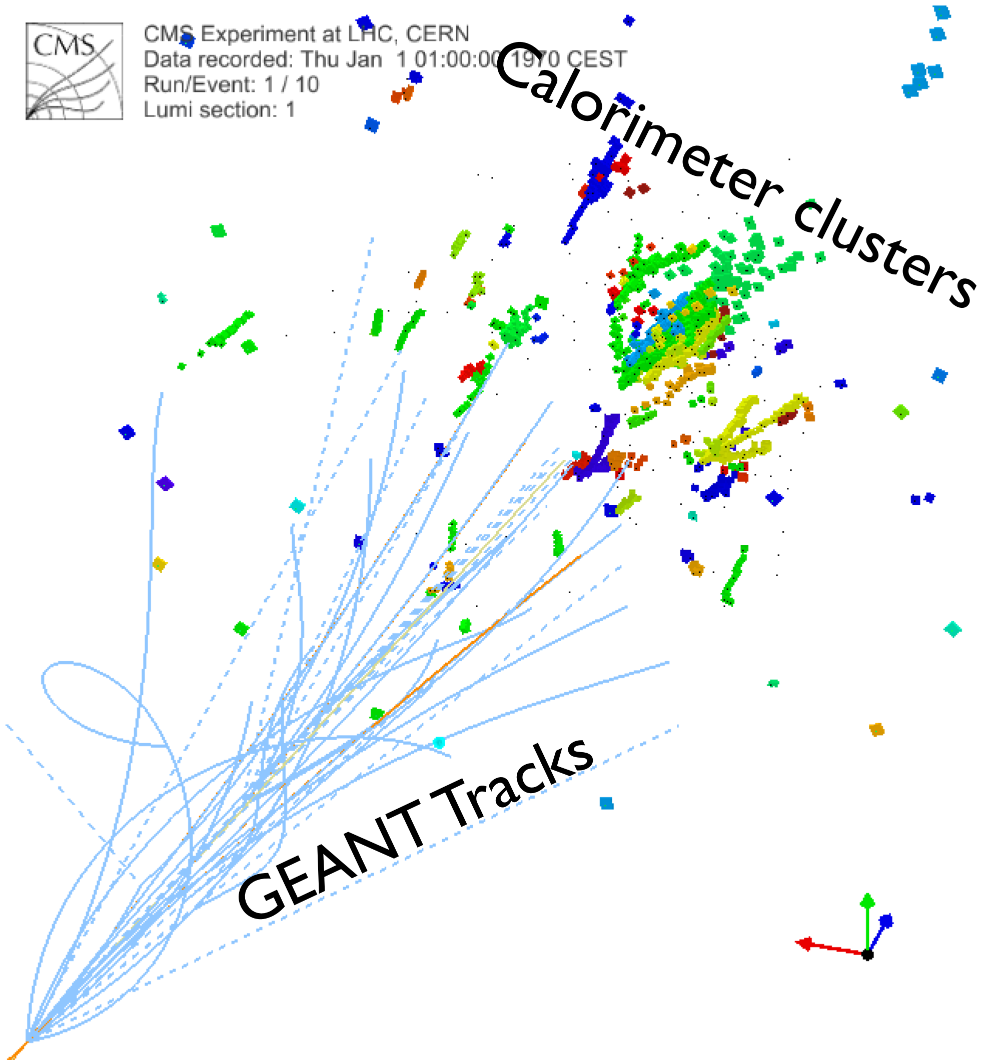


HGCAL Event Display (one endcap)



Imaging Calorimetry With HGCal



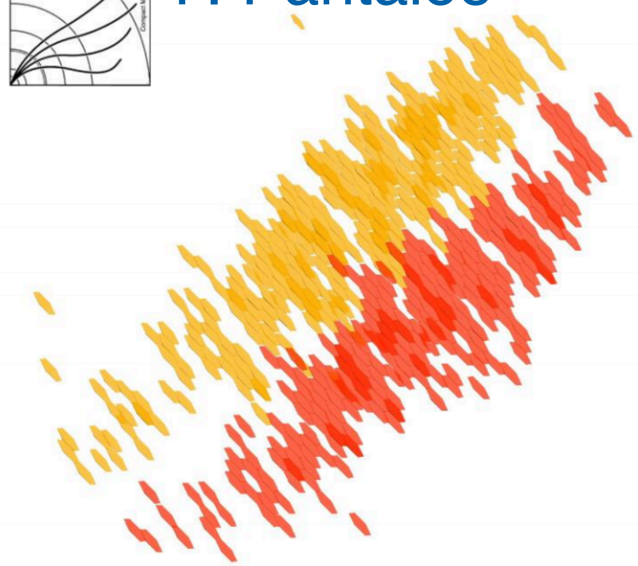


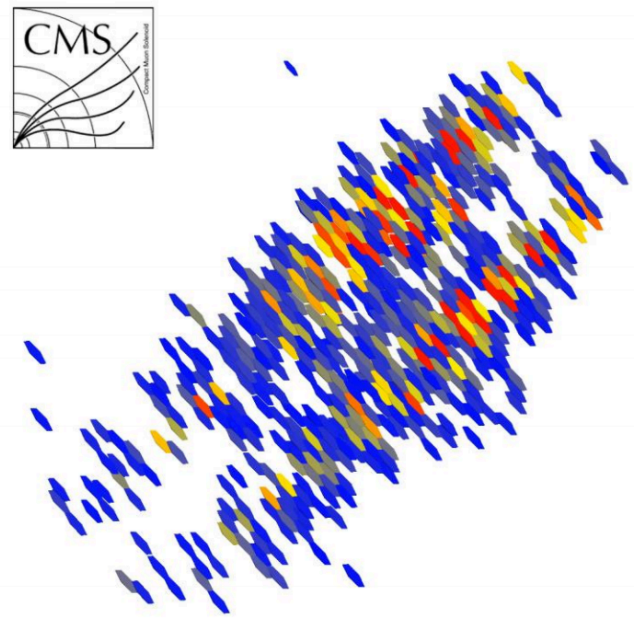
- Rough 6 million channels individually read out
  - Provides sampling calorimetry with 50 instrumented readout planes
  - Can capture the evolution of EM and hadron showers in space as well as time
    - Dedicated timing readout with excellent precision for large energy deposits
  - Higher-dimensional data leads to more easily discernible patterns
- Multiple reconstruction algorithms efforts ongoing to use this device

The cost of having more information:

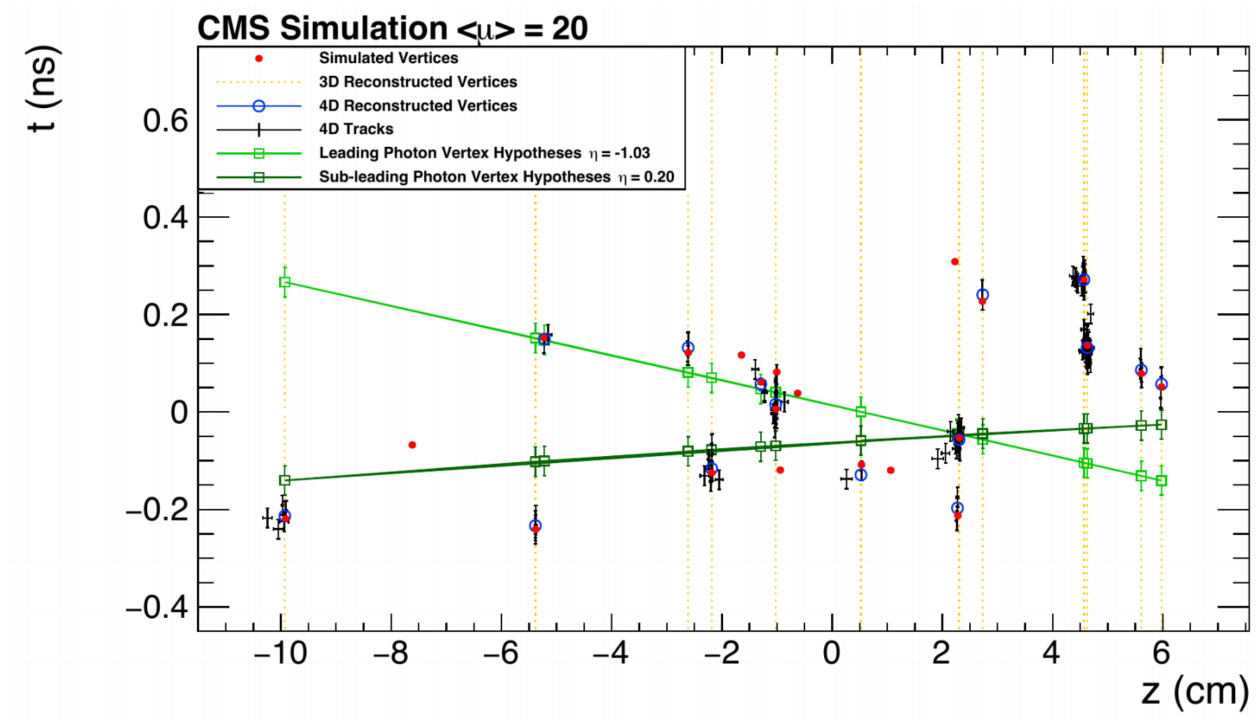
Overlapping photons in HGCal

F. Pantaleo





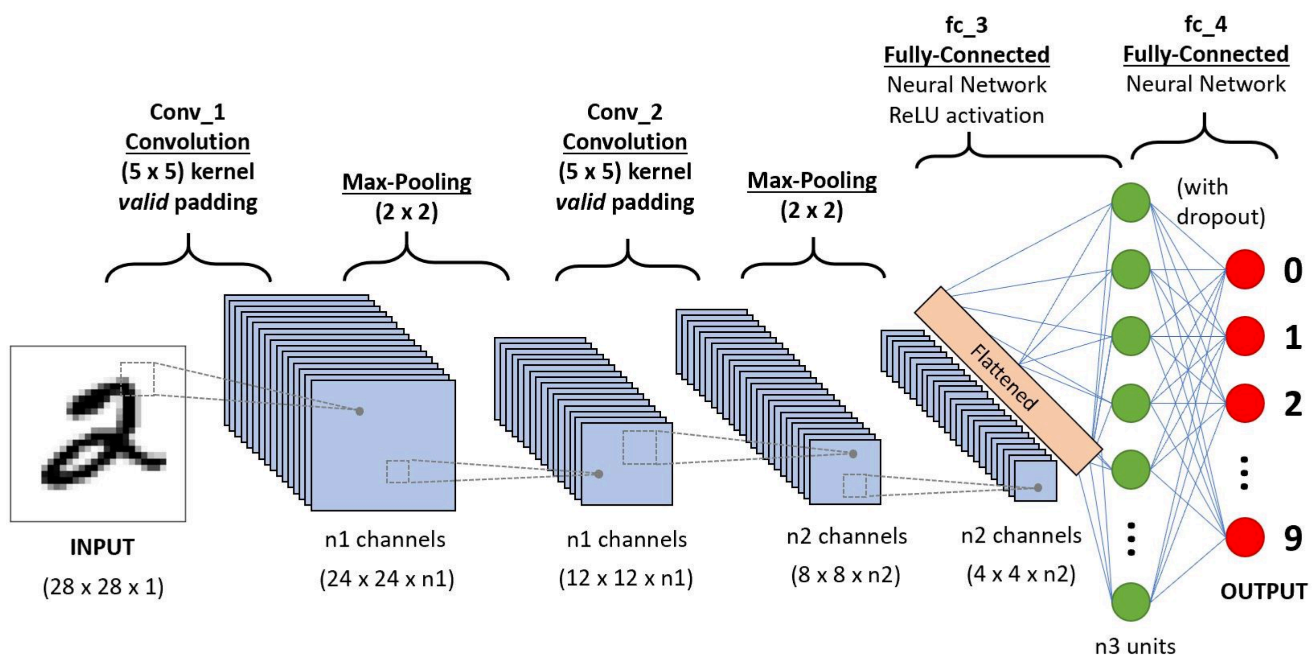
Cross-referencing neutral and charged precision timing information

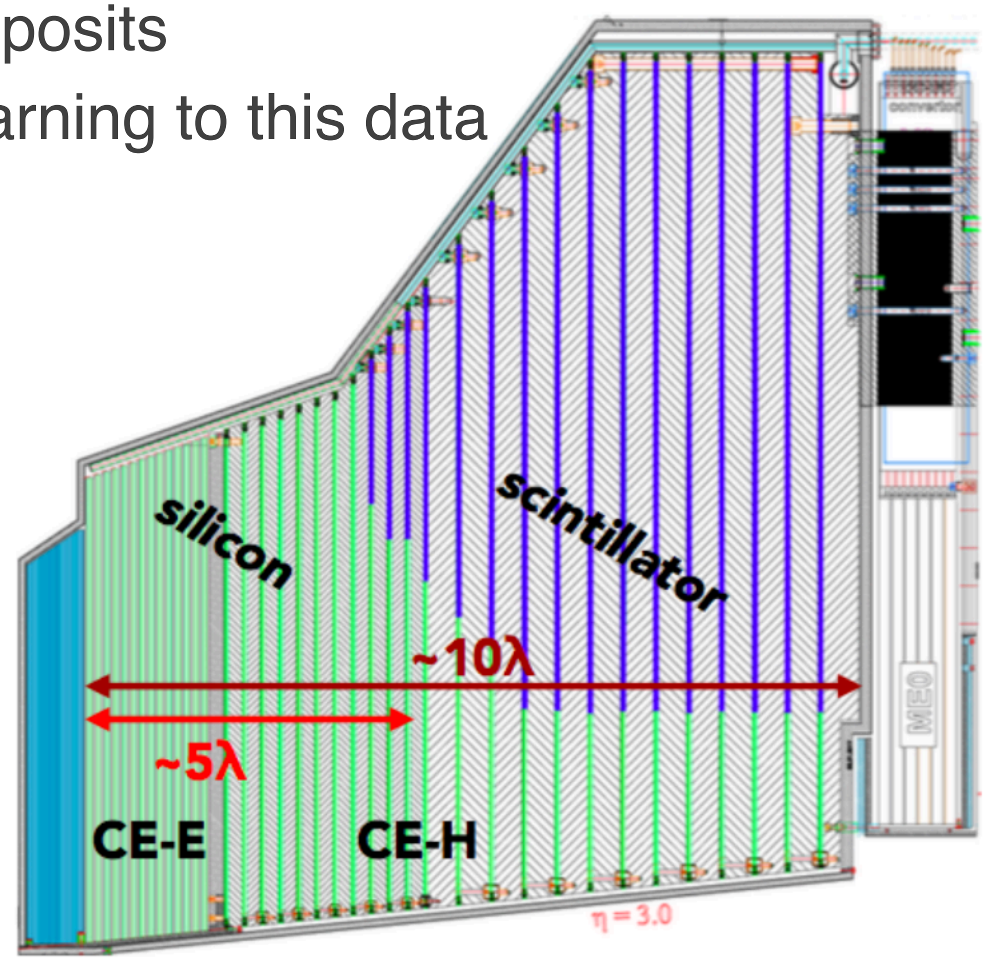


- While the computing costs of more data are clear, it takes significant human time to engineer algorithms that take advantage of more data
  - High dimensionality, while more sparse is far more difficult to reason with effectively
  - For instance: thinking in projections often leads to designing algorithms that mischaracterize some behavior
- The best approach by far, is to try to handle the detector information in its full dimensionality, but humans are not well equipped to do that above 3D
  - Moreover, each detector has its own unique geometry which has to be specifically accounted for

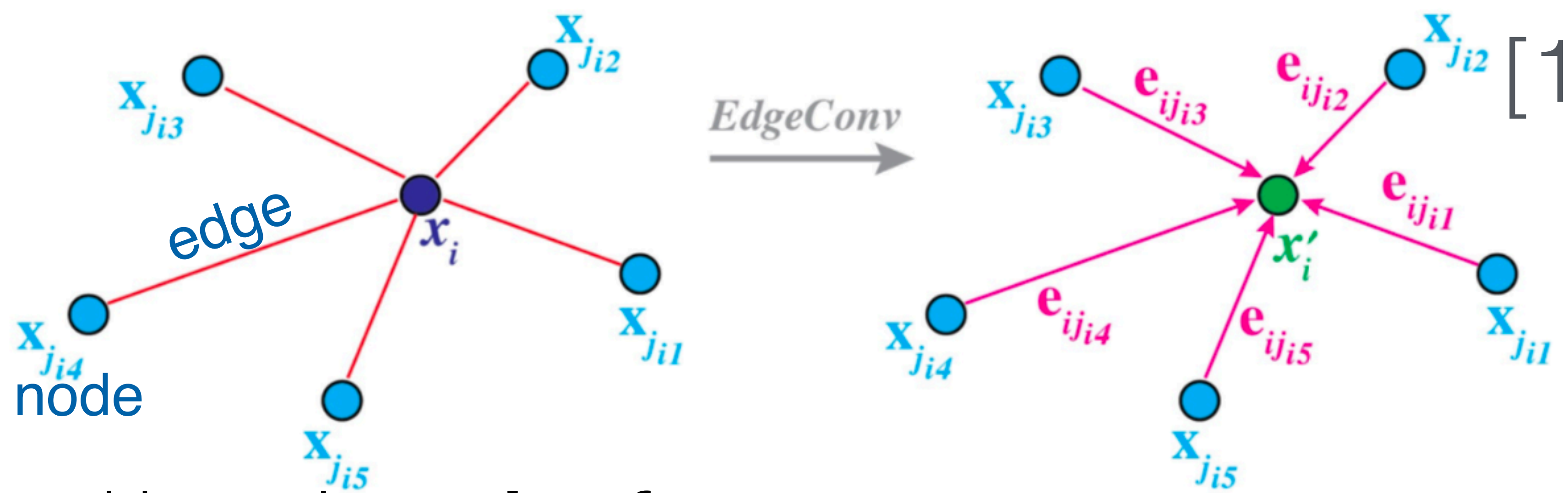
Exploiting granular information with machine learning

- Modern machine learning can determine important discriminating information in the course of training if the input 'shape' is fixed
  - Using convolutional neural networks for example, images are given as-is for training examples, discriminating features encoded in filters and high-dimensional 'latent spaces'
- However, many next generation particle physics detectors have irregular geometries with zero-suppressed outputs
  - Varying material with sparse sampling of energy deposits
  - Requires different approaches to apply machine learning to this data





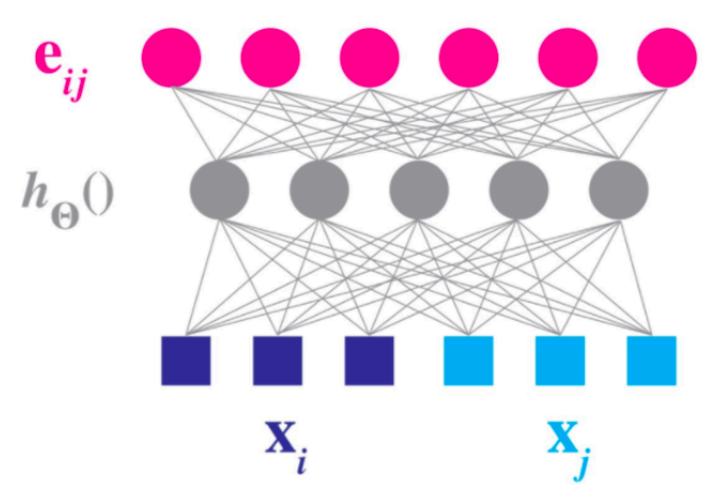
Graph Neural Networks: Edge Convolution



[1801.07829]

- Update $x_i \rightarrow x_i'$ by using **edge features**
  - i.e. learned features of the edges that connects x_i with its neighbors
  - Still independent of ordering of points, but uses **local geometry**
  - '**Convolutional**' as the operation is applied point by point to obtain **x**'
- These edge features and aggregation steps mimic the functionality of loops with if-statements in them (i.e. hand-written pattern recognition)

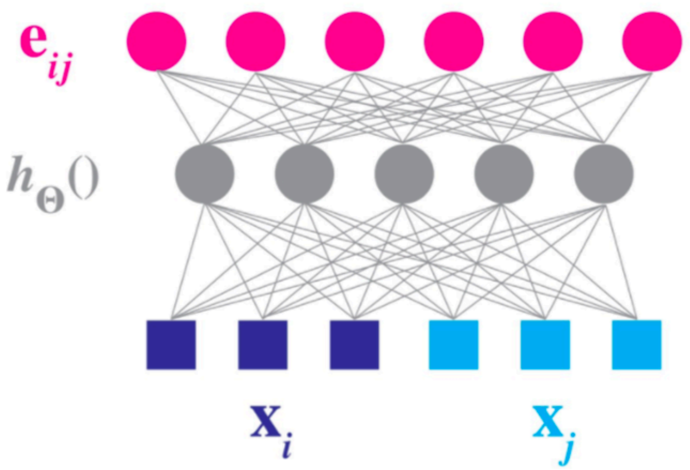
$$\mathbf{x}'_i = \square_{j:(i,j)\in\mathcal{E}} h_{\Theta}(\mathbf{x}_i, \mathbf{x}_j)$$



Can set dim of edge feature vector

Graph Neural Networks: Dynamic Graph Convolutions

[1801.07829]



$$\mathbf{x}_i' = \underset{j:(i,j)\in\mathcal{E}}{\square} h_\Theta(\mathbf{x}_i, \mathbf{x}_j)$$

$$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_i)$$ No neighborhood info (only global)

$$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = h_\Theta(\mathbf{x}_j - \mathbf{x}_i)$$ Only local information

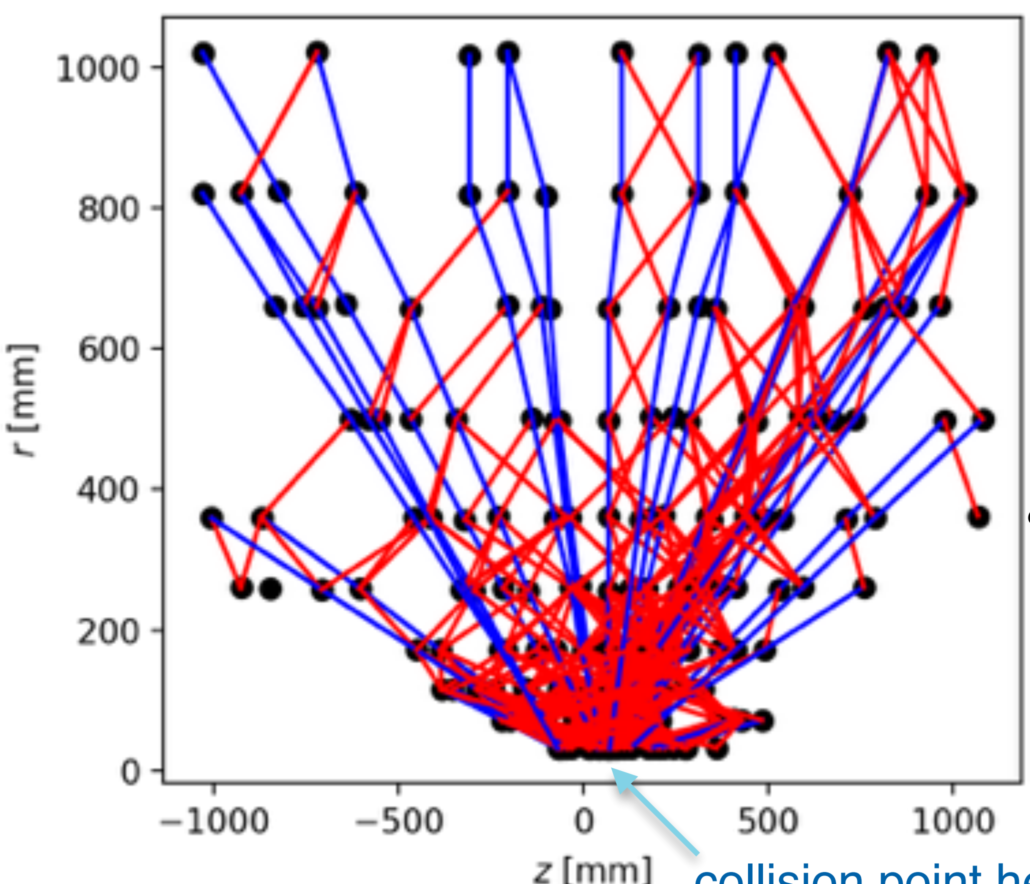
$$h_\Theta(\mathbf{x}_i, \mathbf{x}_j) = \bar{h}_\Theta(\mathbf{x}_i, \mathbf{x}_j - \mathbf{x}_i)$$ Combination of both

- **Dynamic:** Redo kNN after every update
  - The connectivity matrix changes after every update

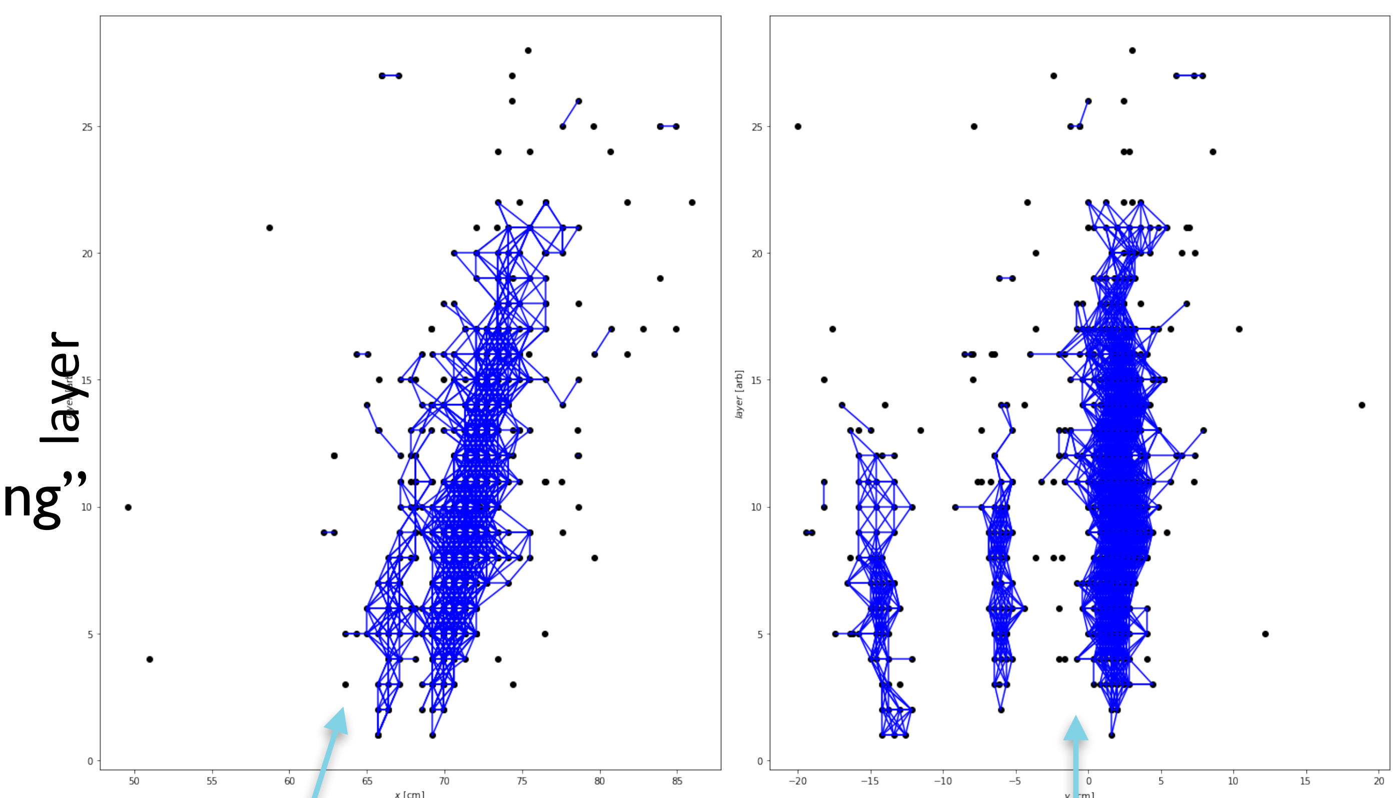
Looking at graphs on physics detector data

- Tracks and clusters can also be described as connections between points
  - We can then score these relationships between the detector data and select certain associations in the graph that we want to keep.

- This results in a useful abstraction: finding points comprising helices in tracks is the same as points in calorimeter clusters
  - Can we simplify our lives and find one algorithm which can handle these different cases?



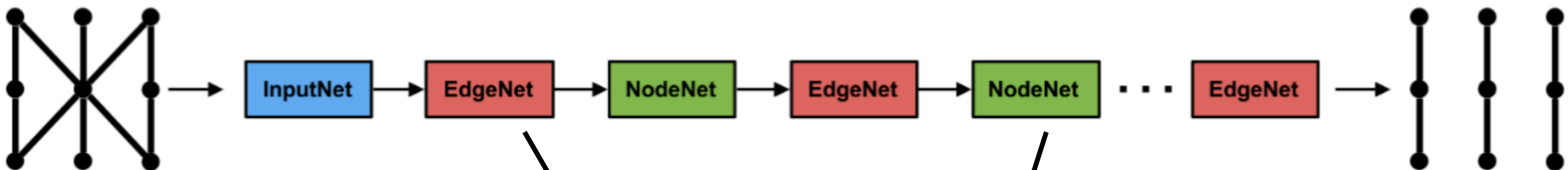
“same thing”



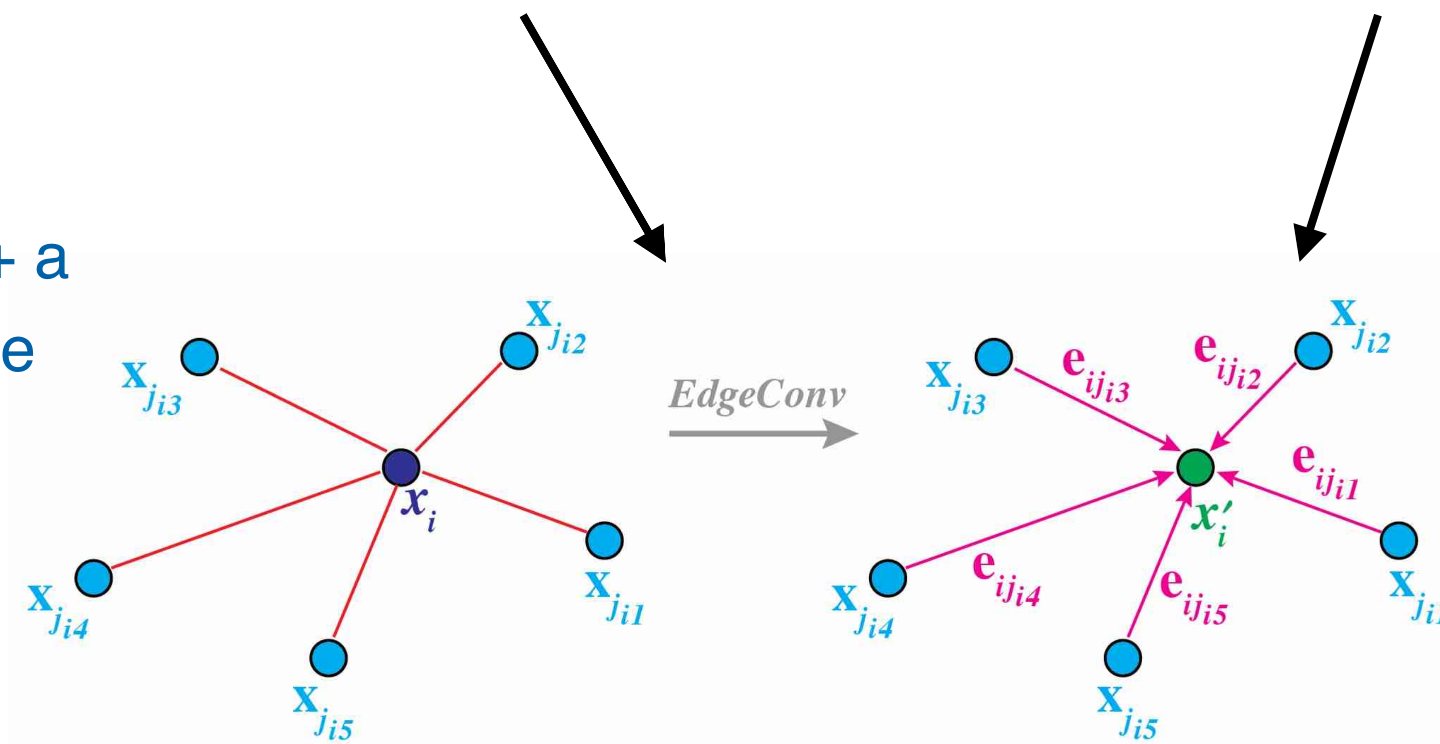
Putting it all together: a model for reconstruction

- With an preliminary model the answer seems to be “yes”
  - So long as we are willing to accept some light post processing

- Basic steps:
  - Define an input graph
  - train an ‘edge classifier’ based on information sharing on that graph
  - Apply edge classification scores to yield a subgraph of just the connections of interest



just raw hit information + a “guess” at the graph

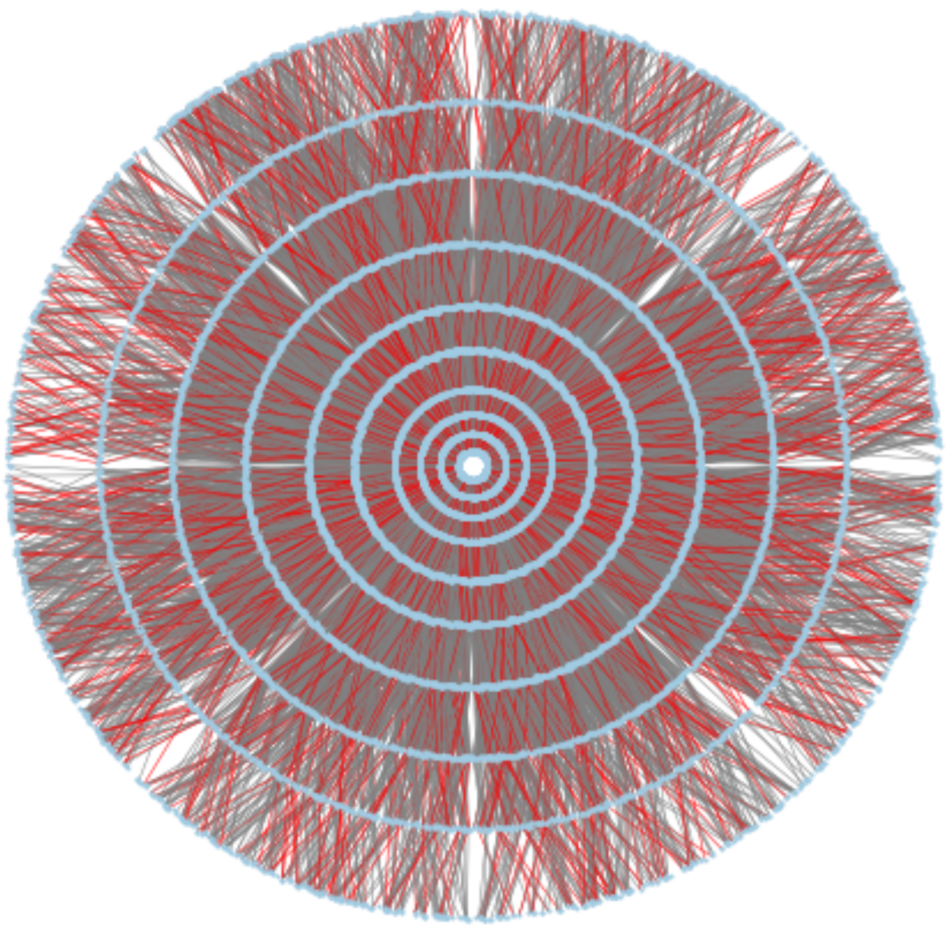


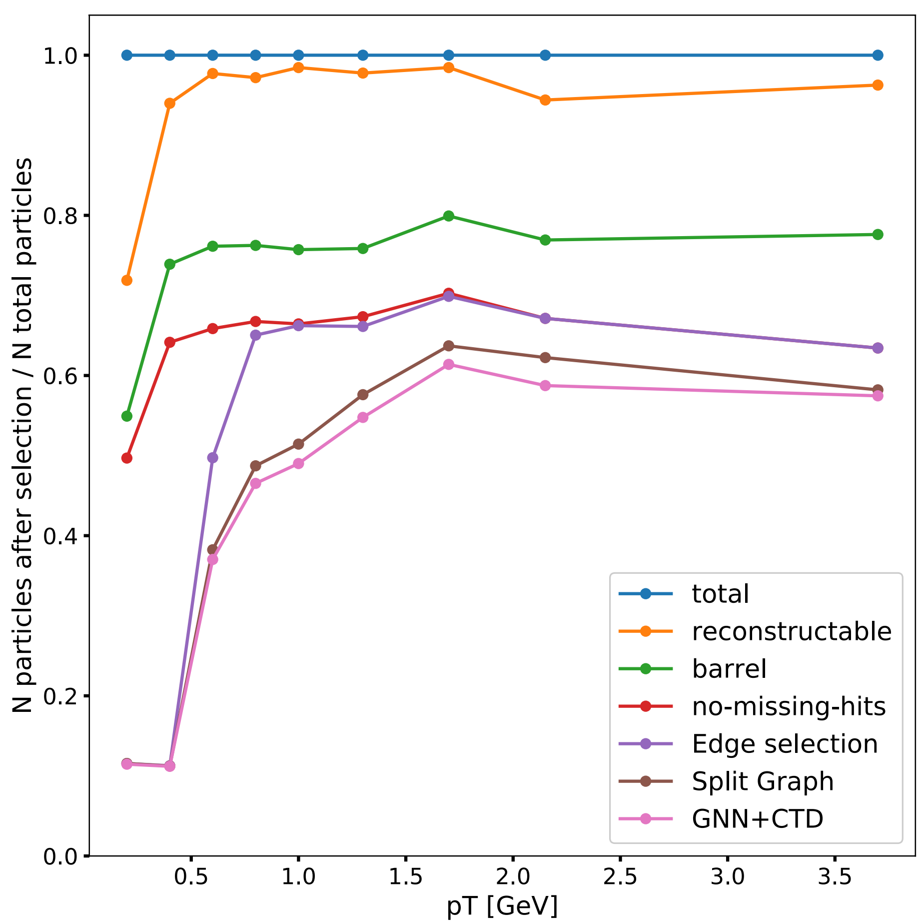
selected links between hits

https://arxiv.org/abs/2003.11603

Preliminary Tracking Results with a GNN

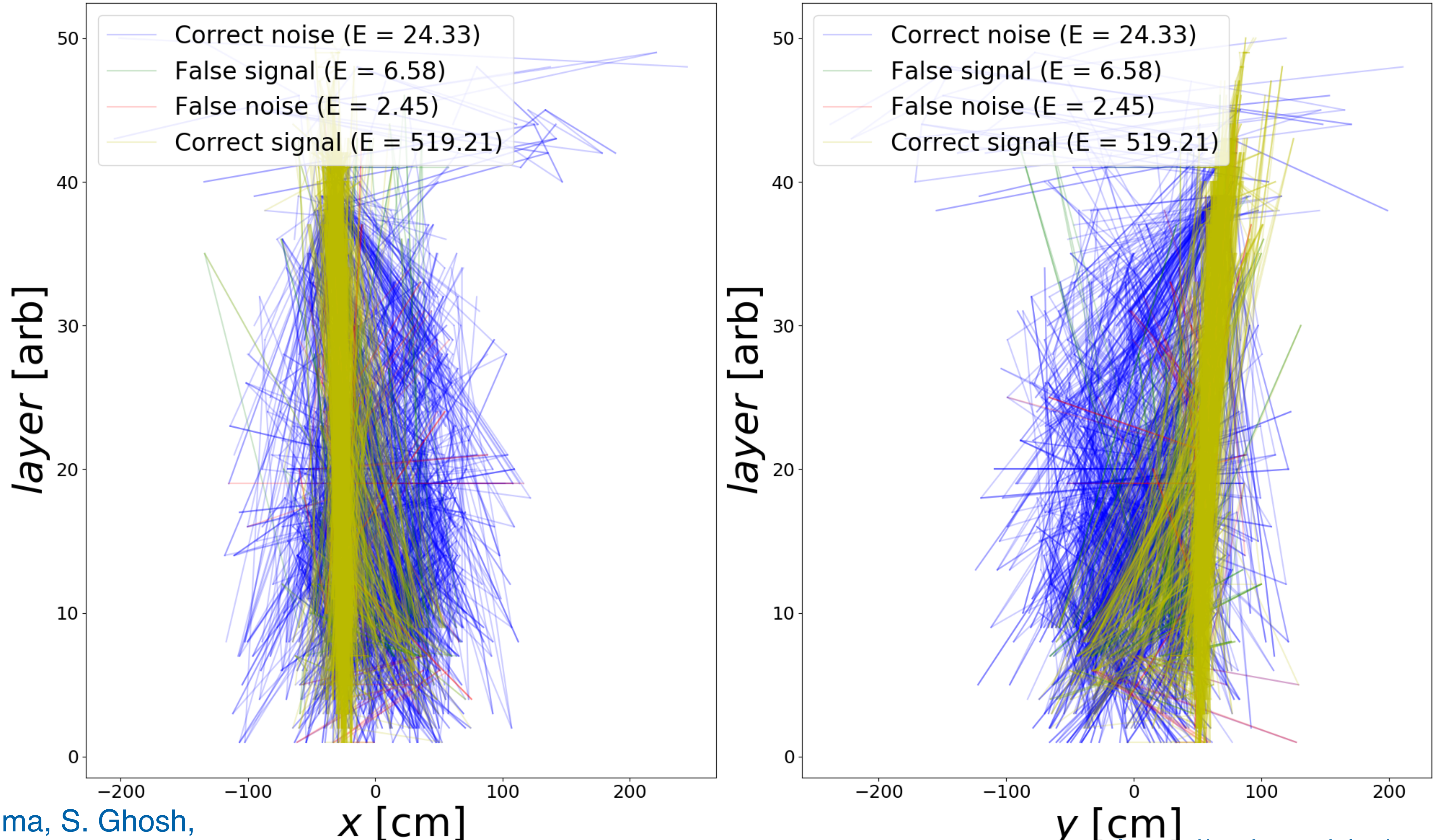
- Many selections applied to yield training set
  - Important: sectorization and no missing hits
- These are “easy” tracks but this also early days for the these kinds of network in HEP
  - Applying GNN, assembling tracks -> 97% efficient relative to preselection
  - Track-segment selection GNN executes significantly faster than Kalman filter





Reconstruction of a charged pion with edge classification

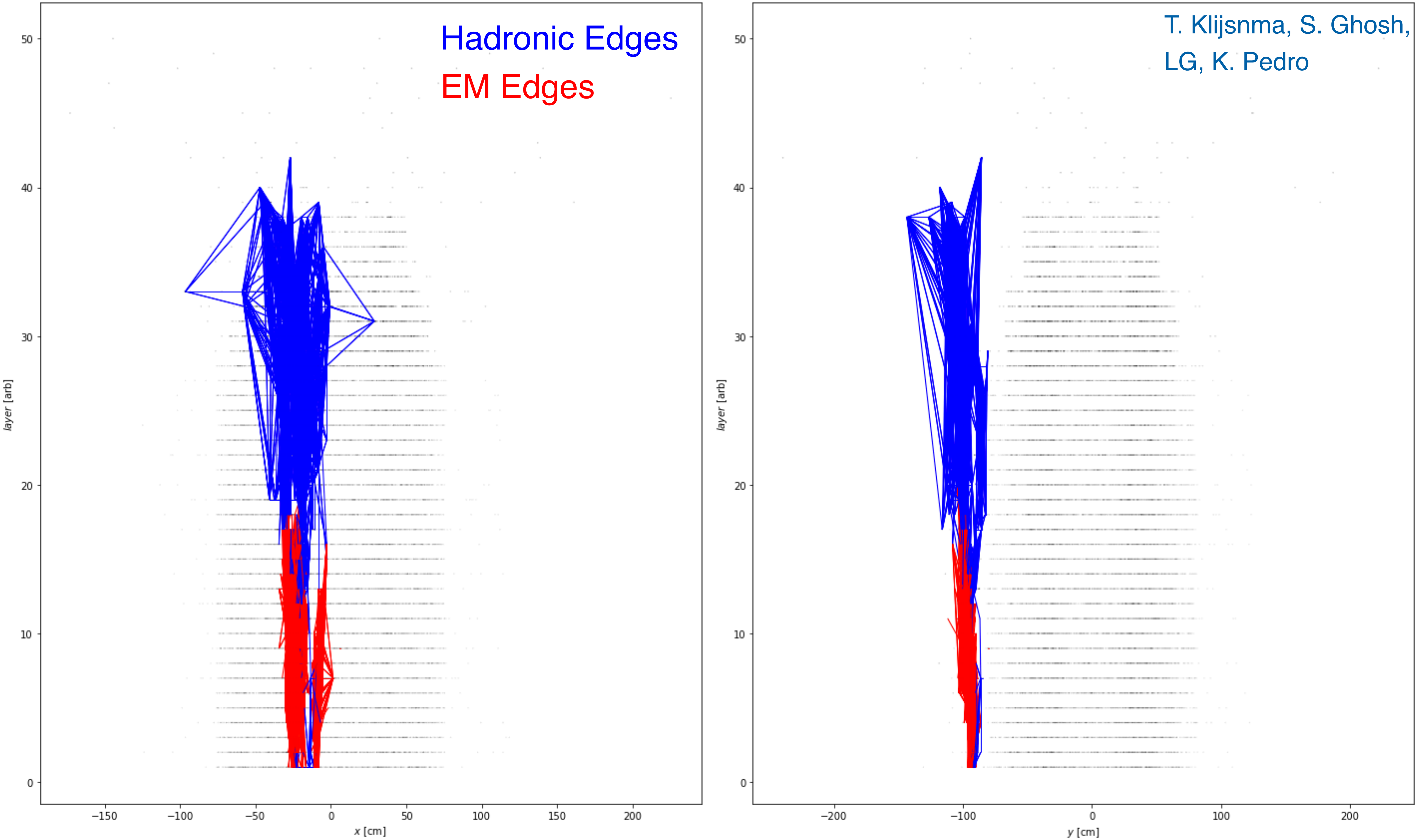
true negatives
true positives
false positives
false negatives



T. Klijsnma, S. Ghosh, LG, K. Pedro

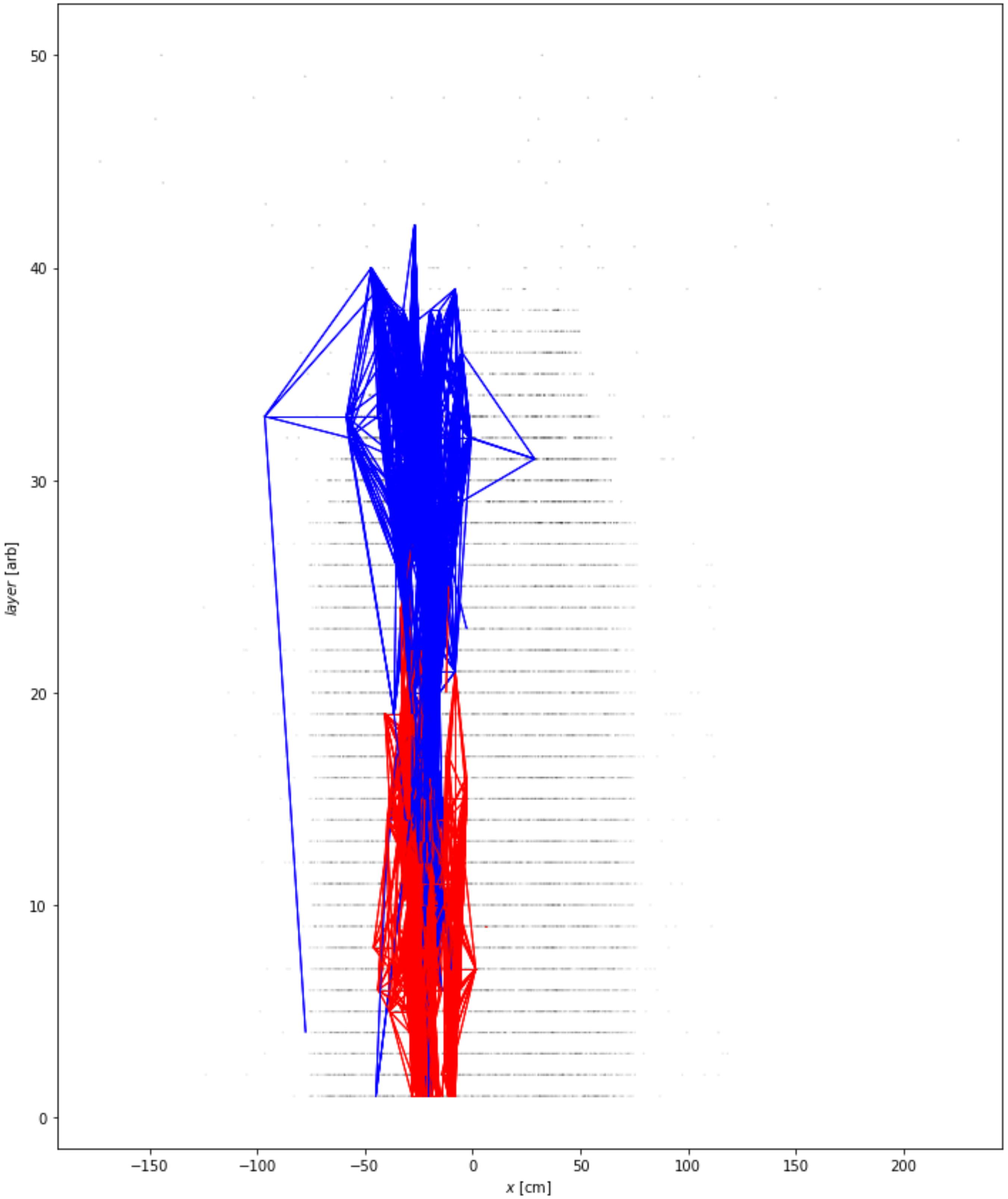
https://arxiv.org/abs/2003.11603

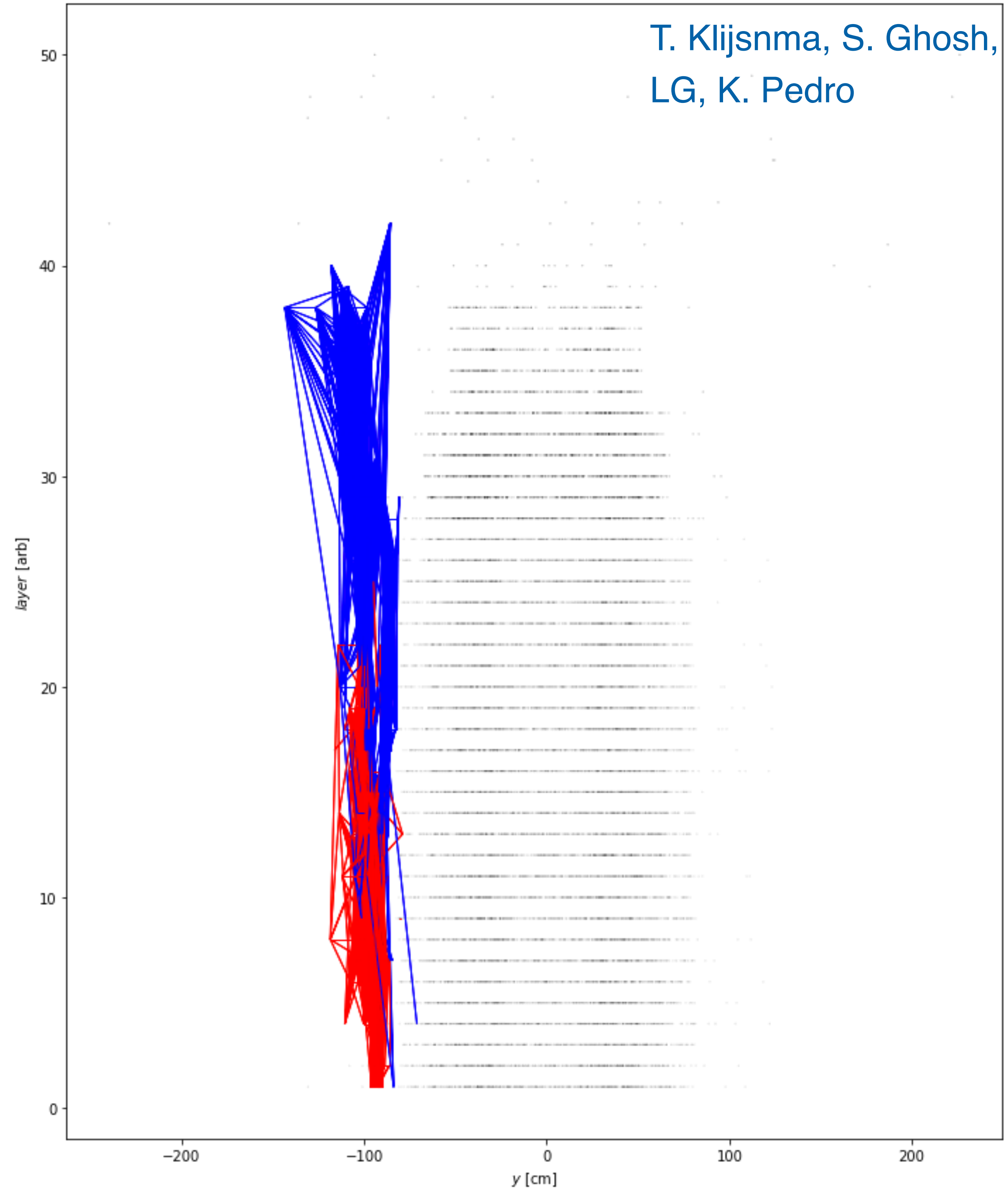
Simultaneous Reco & ID: Tau Lepton Example Prediction



T. Klijsnma, S. Ghosh, LG, K. Pedro

Simultaneous Reco & ID: Tau Lepton Example Truth



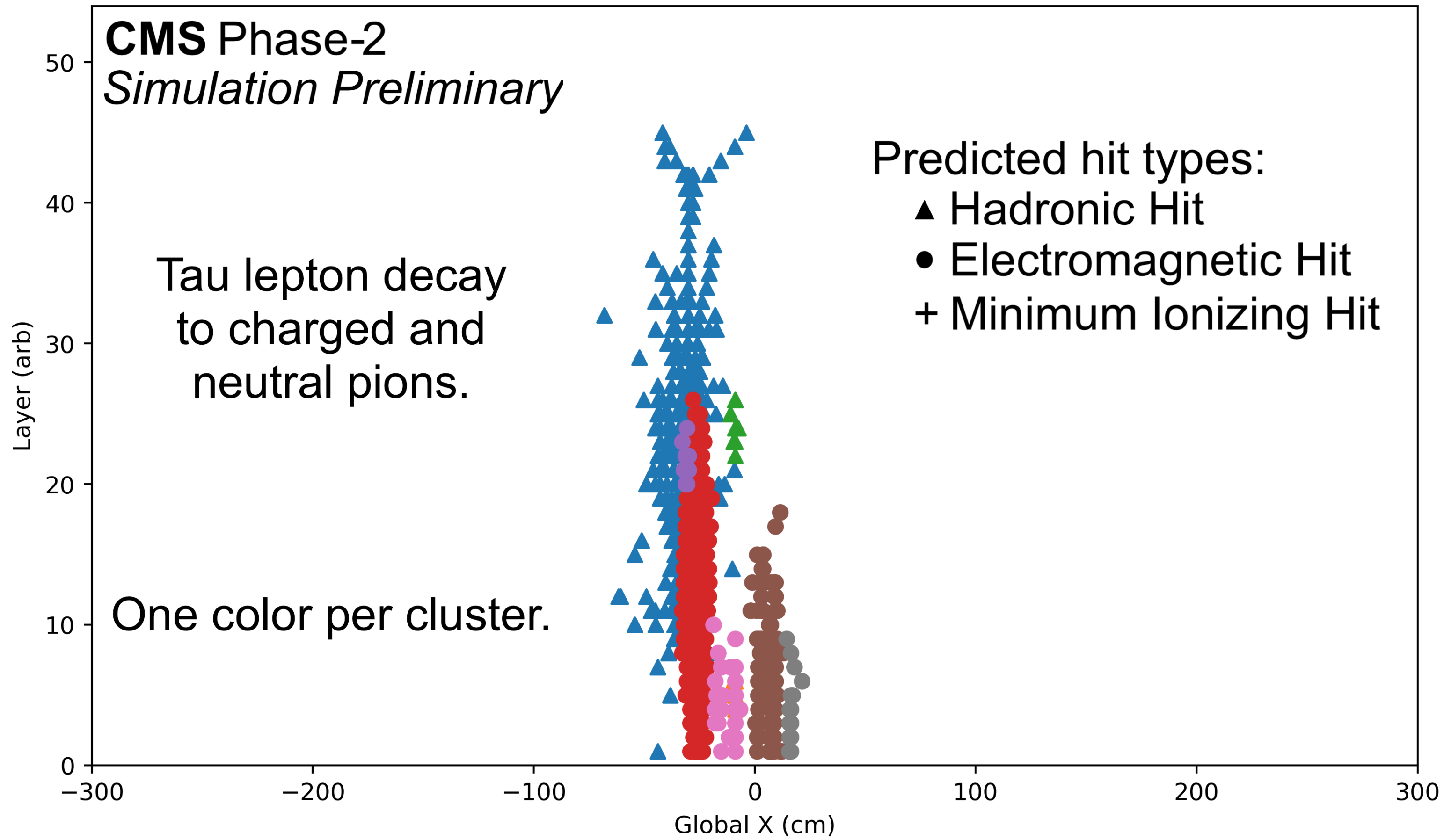


T. Klijsnma, S. Ghosh, LG, K. Pedro

Edge Classification: Making a Clustering (I)

- In order to get calorimeter clusters, need to take the edges and convert to groups of points
  - In this case we just make a union of all the points with common edges of the same type
  - It does a reasonable job already segmenting hadronic energy from electromagnetic
  - We can reconstruct very close-by photons and hadrons effectively

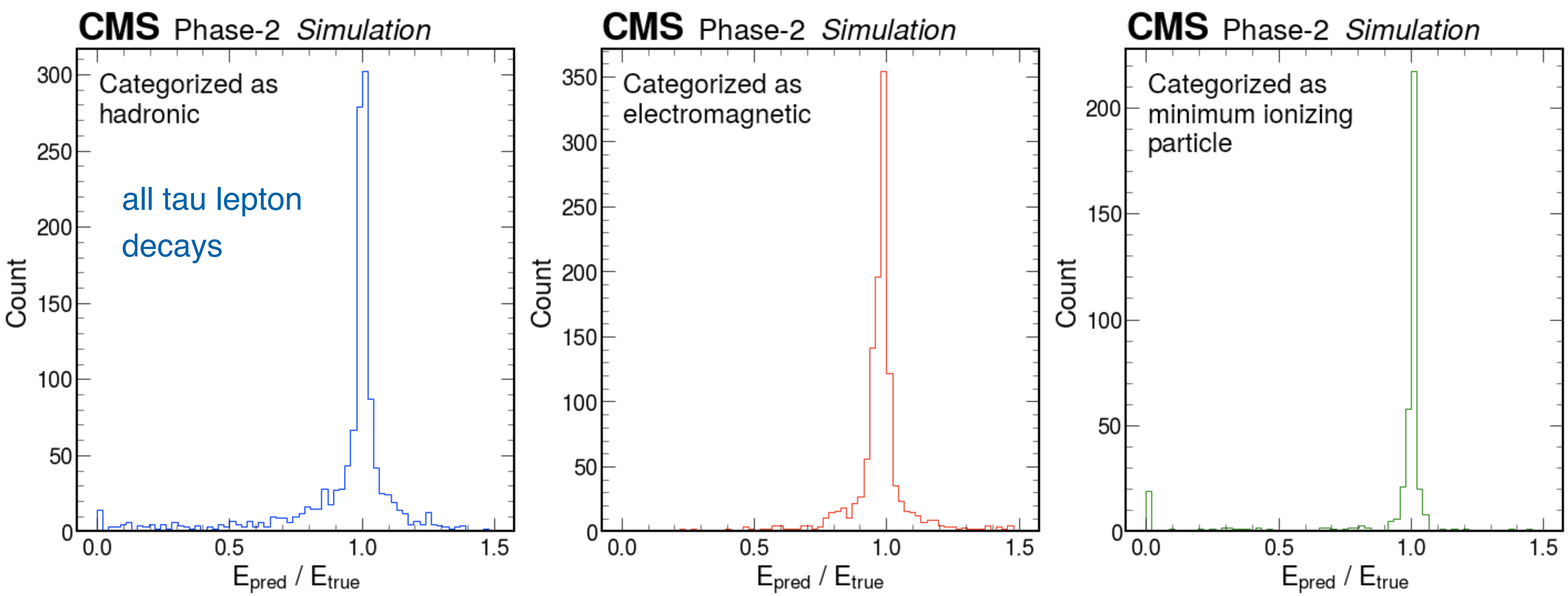
- The same network and processing can also be used on tracking



T. Klijnsma, S. Ghosh, LG, K. Pedro

Edge Classification: Making a Clustering (II)

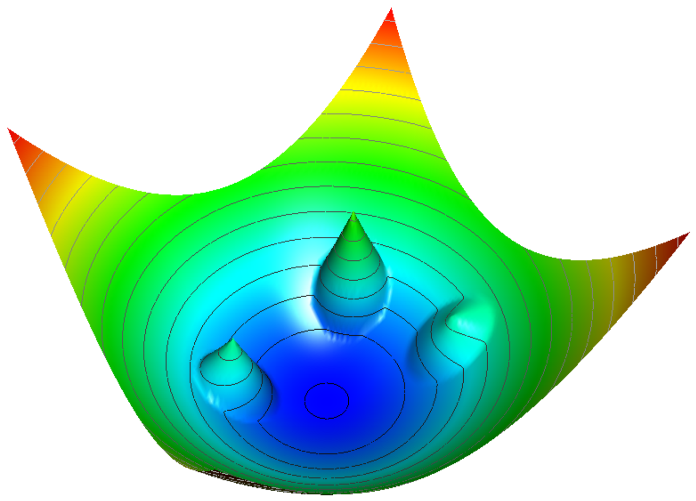
- In order to get calorimeter clusters, need to take the edges and convert to groups of points
  - In this case we just make a union of all the points with common edges
  - It does a reasonable job already segmenting hadronic energy from electromagnetic
  - We can reconstruct very close-by photons and hadrons effectively
  - Proof of concept achieved
- The same network and processing can also be used on tracking



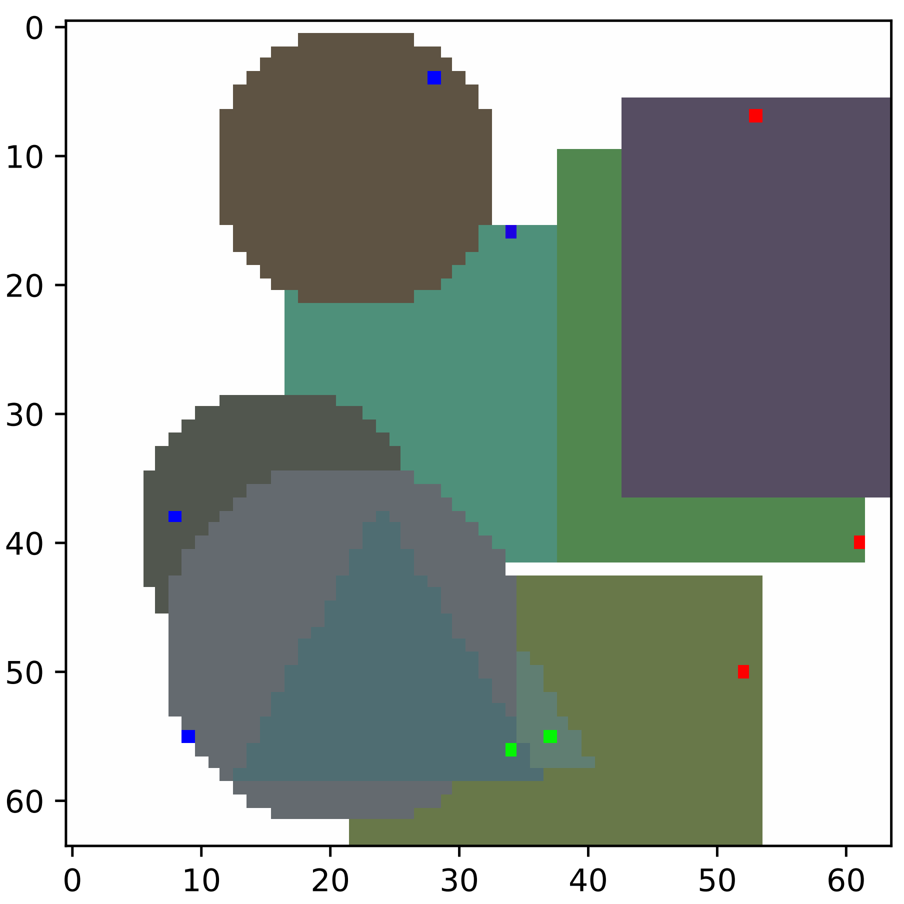
T. Klijnsma, S. Ghosh, LG, K. Pedro

Object Condensation: a loss function for reconstruction

- Physics motivated loss function
  - Potentials with charges
  - like charges attract, opposites repel
  - points that should be associated attract each other
  - variable number of inputs and outputs



- The network is trained to predict the 'condensation points' of the input data
  - Points within the data that are representative of a whole object



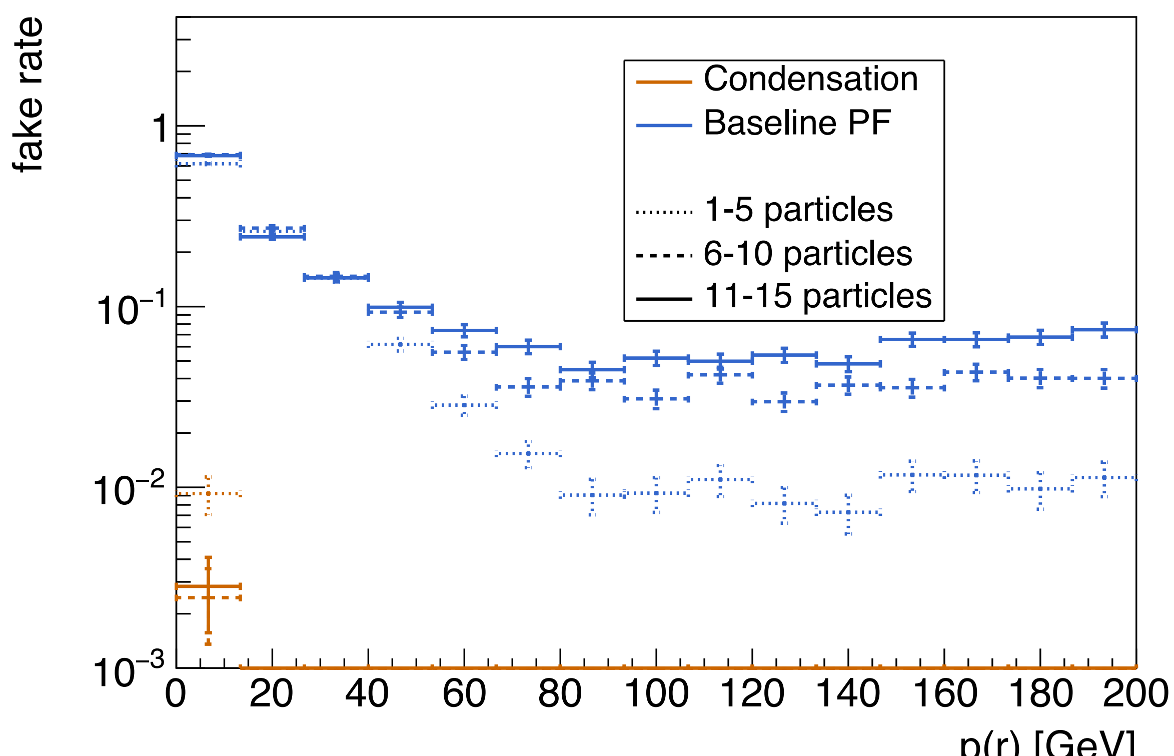
- The condensation points can then be used to collect points around them into 'segmented' objects
  - at this point we have created particles in an event or clusters in a calorimeter

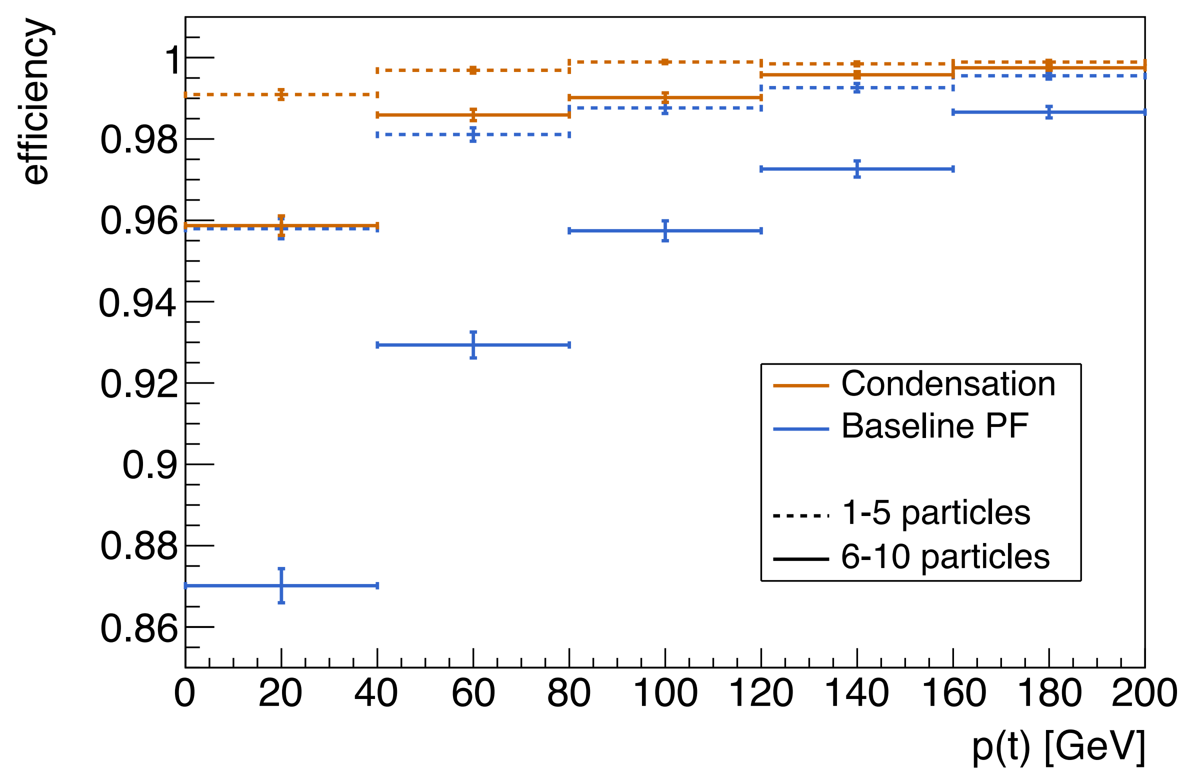
https://arxiv.org/abs/2002.03605

Object Condensation: Results

- A first reconstruction model has been developed and benchmarked
  - Using a toy detector and comparing to a simplified implementation of particle flow
  - Specifically - only a tracker and only an electromagnetic calorimeter

- Particle reconstruction efficiencies significantly improved for object condensation
  - Improved purities and resolutions (backup) across a range of multiplicities as well





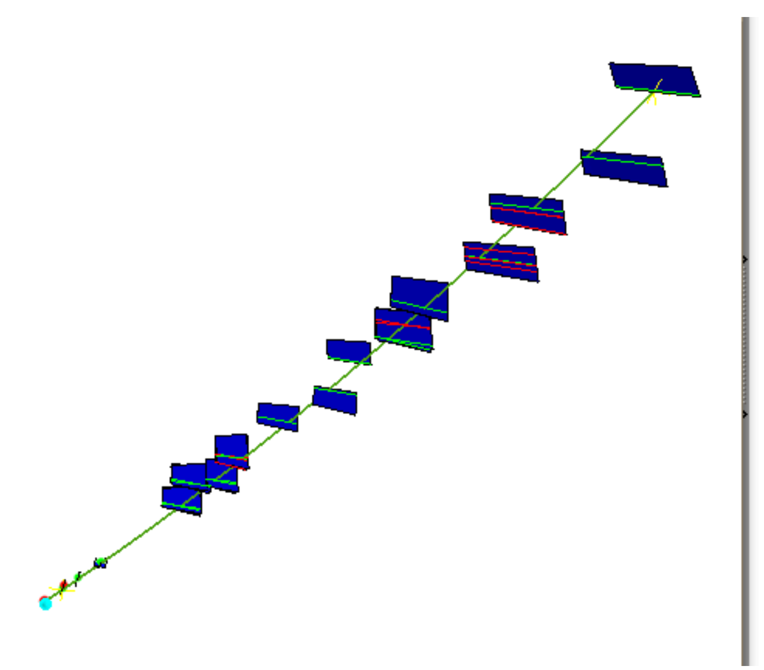
https://arxiv.org/abs/2002.03605

Other methods for one-shot graph pattern recognition

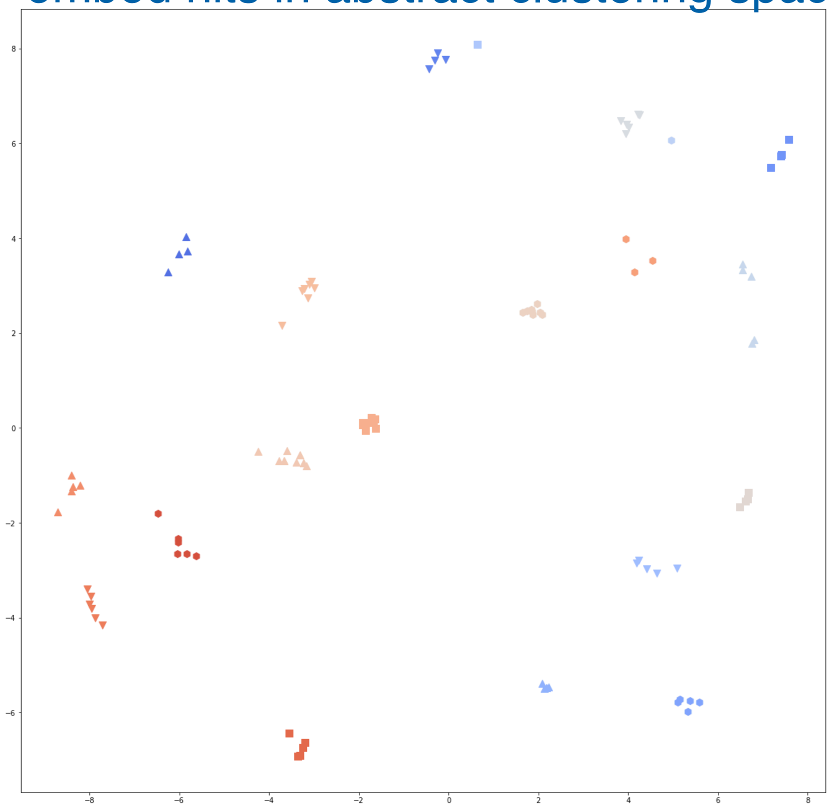
- Taking inspiration from object condensation's embedding
  - Make a network construction that attempts to predict groups of hits correctly
  - Still based on using relational structure between hits
  - But at no point is information concentrated to one point, less 'hierarchy' and sets are predicted rather than output properties (below, examples with a small number of tracks)

- Data are from https://www.kaggle.com/c/trackml-particle-identification
  - Go give it a try yourself :-)

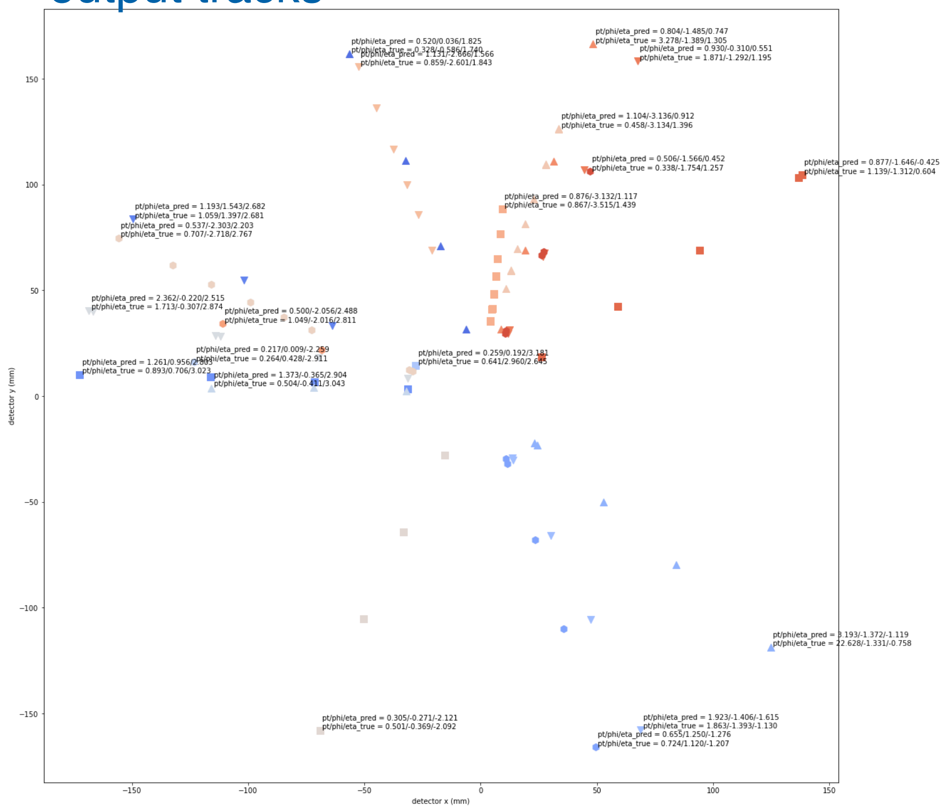
Tracking data in:



embed hits in abstract clustering space

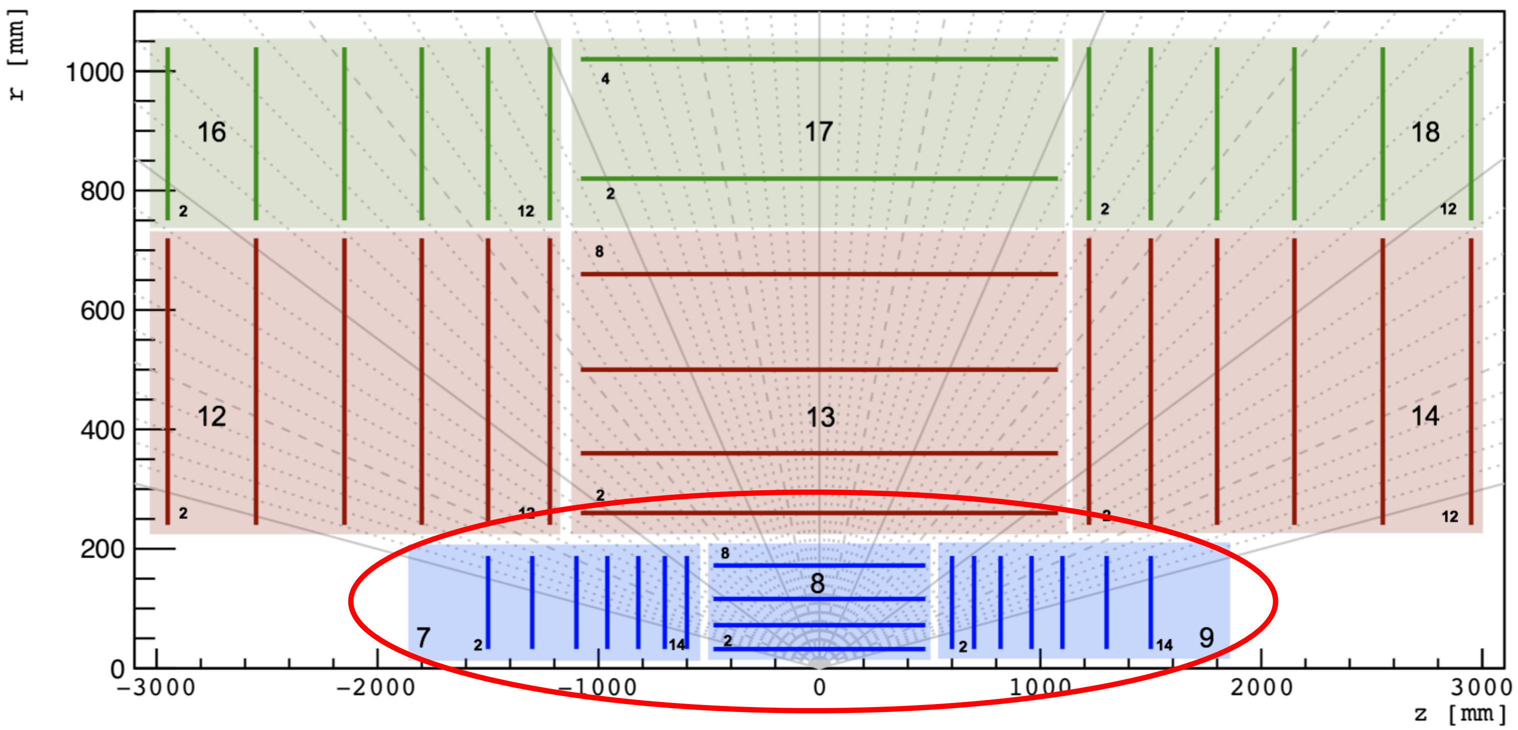


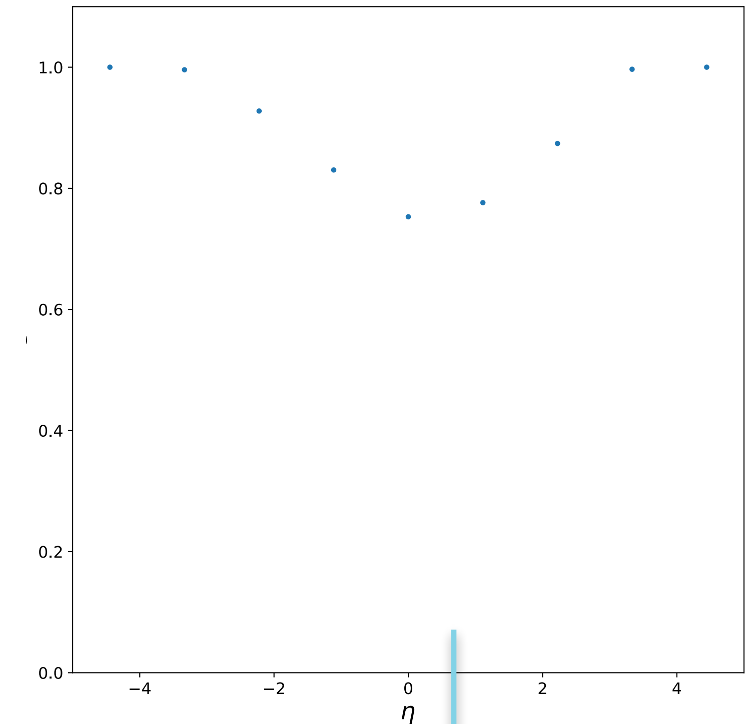
output tracks

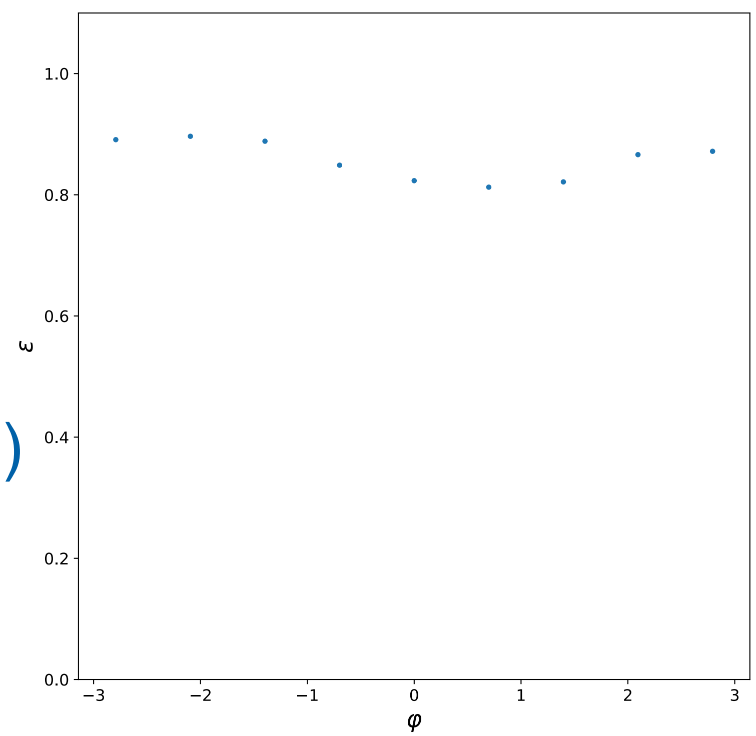


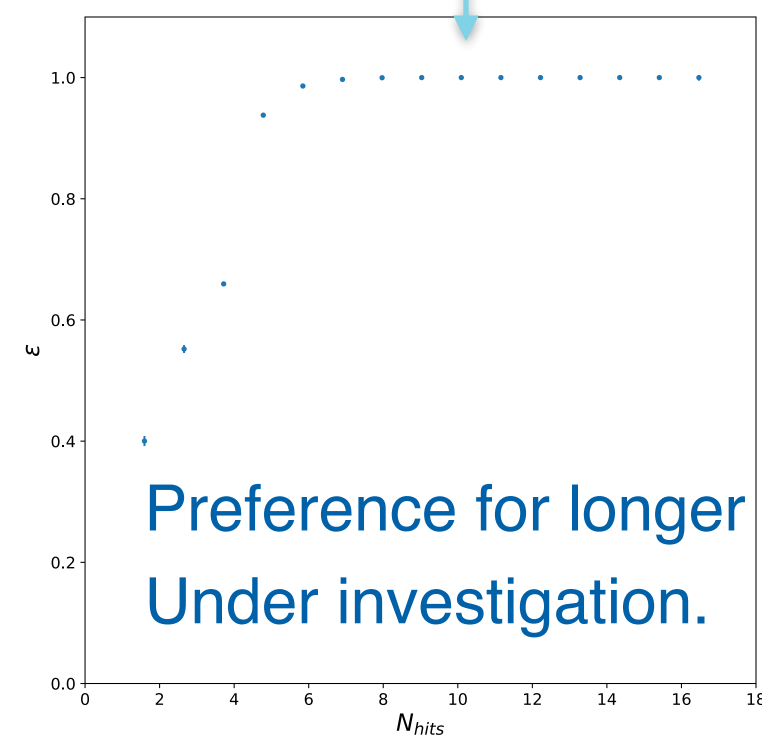
Chhavi Sharma, Thomas Klijnsma, LG

Very preliminary results on pixel tracking









Pardon the dust this is really a work in progress. :)

Preference for longer tracks. Under investigation.

Here we are reconstructing 600 +/- 25 tracks per event.

Limitations of these methods

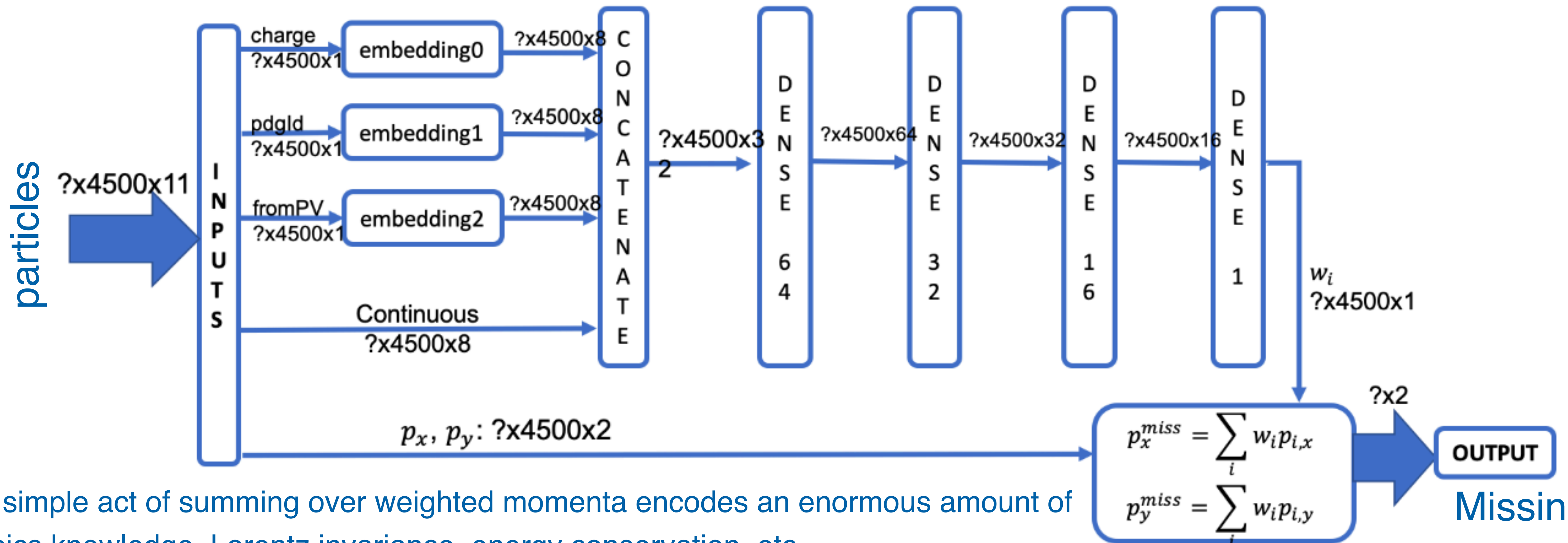
- These methods require repeated recalculation of a dynamically determined graph
  - Within these networks are multiple layers of graph networks where the structure depends on the observed data, and the feature spaces are often 32 dimensional or greater
  - So intrinsically there is a computational bottleneck in the determination of the graphs
  - Typically it is possible to find some clever algorithm to ease this, but the scale of particle physics is enormous and the problem remains.

- Graph networks only ensure permutation invariance
  - Permutation invariance encodes very little information about physics!
  - These networks need mountains of data to achieve the best performance because they need many millions of examples of data that follow similar underlying patterns
  - Training takes weeks

- These two things together make the maintenance burden of these networks quite high, and it is worth thinking about if we want to deal with it

Getting around limitations with loss functions

- Taking as example the calculation of missing energy in an event
  - Having good precision and accuracy for missing energy, and its direction, is important
  - Related to important systematic uncertainties for precision measurements in particle physics
- Past attempts tried to use complex models to map various measurements together, requires enormous amounts of training data (millions)
  - A simpler model, below, does better with 60k events and 5000 parameters. How?



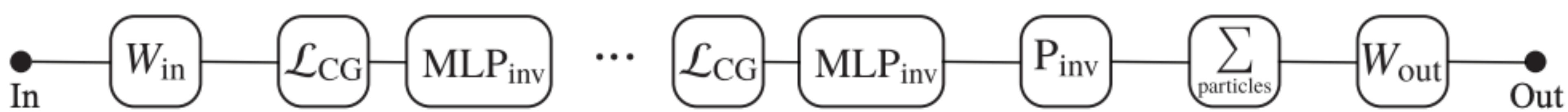
The simple act of summing over weighted momenta encodes an enormous amount of physics knowledge. Lorentz invariance, energy conservation, etc.
With so much information in the loss construction, the network becomes very simple.

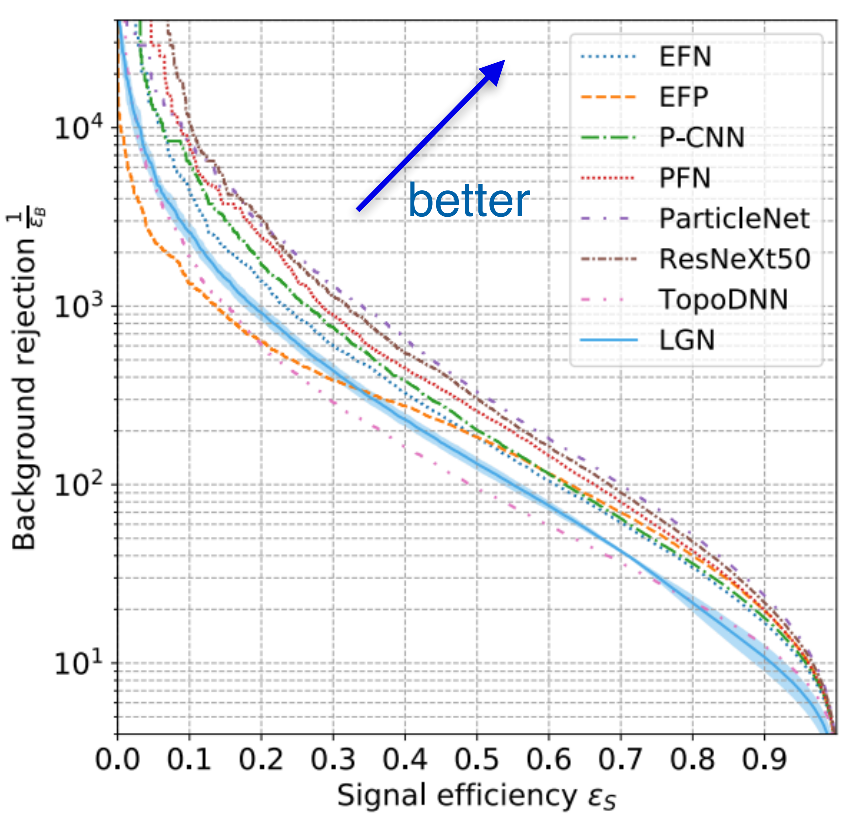
Yongbin Feng, Jan Steggeman

Fermilab

Getting around limitations with equivariant activations

- The loss function trick can be done for a limited class of problems
  - If we to use the known symmetries of the data in an abstract way we must change how the non-linear activations in neural networks
  - One direction: enforce those activations to behave in a why that is isomorphic to the discrete symmetry group your data obeys, in this case SO+(1,3)
  - Since SL(2,C) is a double cover of SO+(1,3) you can create sums of activations that obey the lorentz transformations, and enforce a network to learn only equivariant quantities!





Architecture	Accuracy	AUC	$1/\epsilon_B$	#Param
ParticleNet	0.938	0.985	1298 ± 46	498k
P-CNN	0.930	0.980	732 ± 24	348k
ResNeXt	0.936	0.984	1122 ± 47	1.46M
EFP	0.932	0.980	384	1k
EFN	0.927	0.979	633 ± 31	82k
PFN	0.932	0.982	891 ± 18	82k
TopoDNN	0.916	0.972	295 ± 5	59k
LGN	0.929 $\pm .001$	0.964 ± 0.018	435 ± 95	4.5k

https://arxiv.org/abs/2006.04780

Mapping these ideas onto pattern recognition

- All of the data we reconstruct obeys some useful symmetries in the lab's frame of reference
  - Current options for ML-based reconstruction do not include any of this information
  - We've been focused on the equally difficult task of encoding the reconstruction algorithms as differentiable programs in the first place.

- Knowing that a charged particle in a magnetic field follows a helix, or that particle showers are mediated by a splitting process are similar examples of rules that could be embedded in the basic operation of a neural network
  - Then the job of the neural network becomes learning and exploiting relationships in the context of those rules rather than needing to encode those rules as well
  - This also means that the behavior of the network is tied to rules that we understand as humans and greatly improves our ability to understand the performance and estimate systematic uncertainties.

- As we integrate these two lines of research together it will yield new powerful, compact, and understandable networks that can accurately perform the pattern recognition tasks we need.
  - We are just at the beginning here and the near future is very exciting!

Conclusions and Outlook

- Machine Learning is being used for more and more fundamental tasks in HEP
  - Adoption of ML techniques has led to simplification in analysis definition
  - We have also demonstrated that we can control the process of training and applying these techniques to yield *precision results*

- ML techniques have been evolving to become more dynamic and particle physics is following along
  - We are now at the point where we can make differentiable versions of iterative algorithms, which was not possible 4 years ago
  - We can now implement and use complex reconstruction algorithms end-to-end in ML

- The recent advent of enforcing symmetries in a general way in neural networks will improve and simplify the designs of these networks
  - The will lead to learned pattern recognition machines with concise, understandable descriptions of what they are doing and significantly reduced burden on those maintaining it

Extras

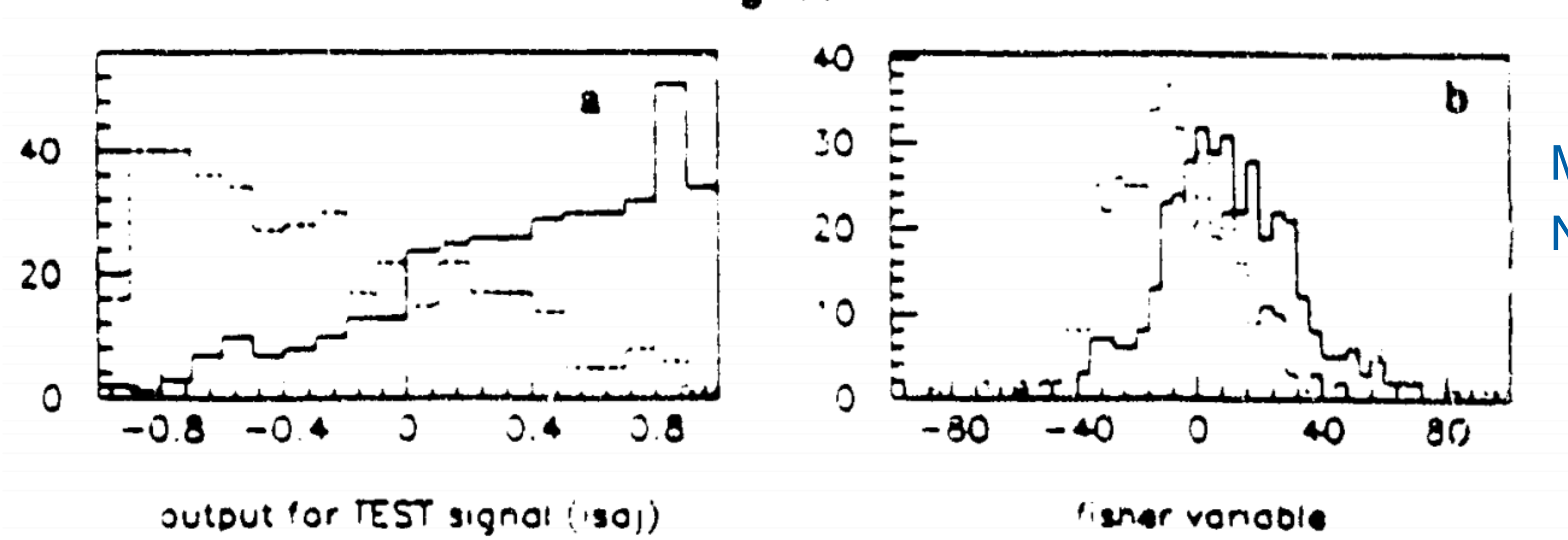
How did we get to where we are going?

- The detectors and challenges, and the tools to address them are the result of a long story in particle physics
  - We always want better discriminators that utilize more information

- HEP Physicists have to demonstrate control over methodologies
  - We can't just separate categories of data from one another
  - Error models and confidence regions are required in order to report our results

- Using ML techniques as reconstruction algorithms is the result of decades of accumulated knowledge within HEP

Figure 2



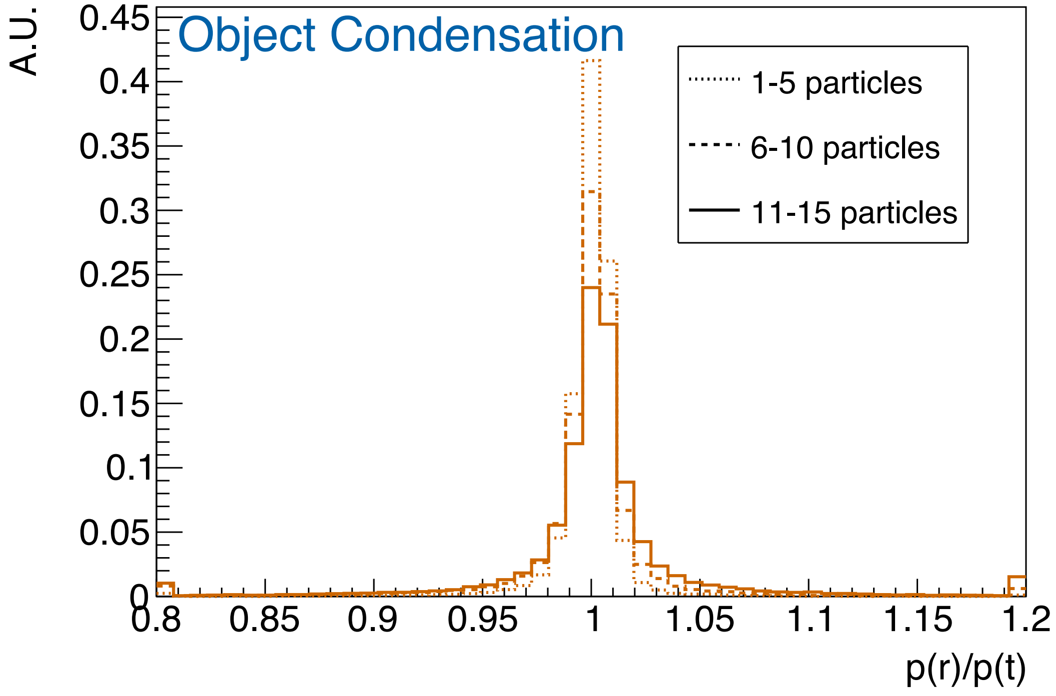
Multijet Top discriminant from 1992

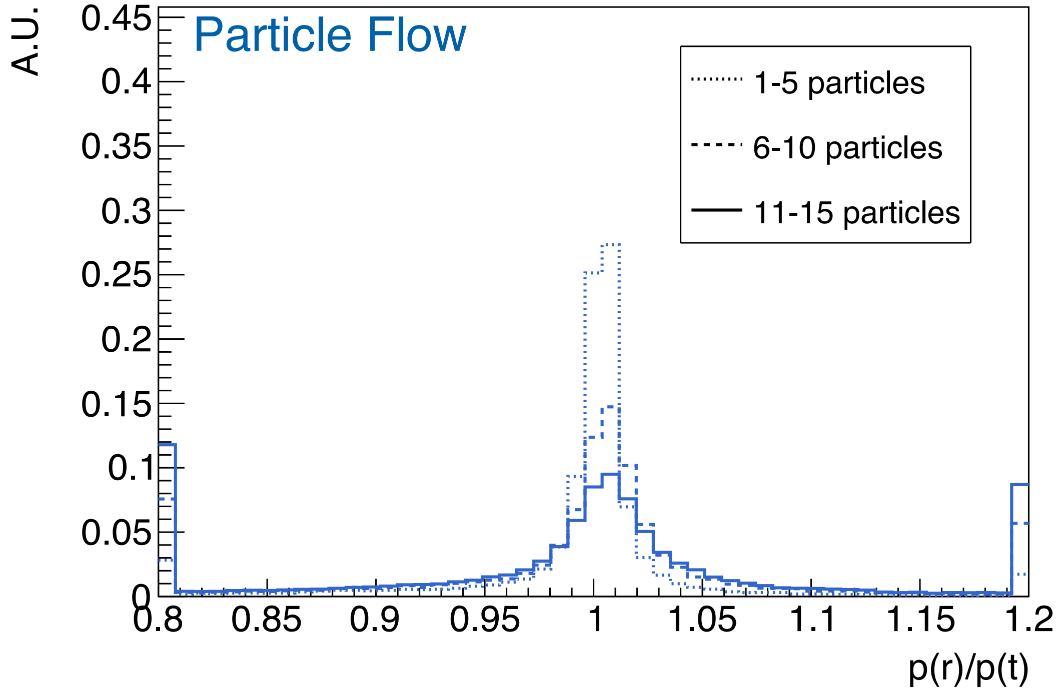
Neural Network (left), Fisher discriminant (right)

https://www.osti.gov/biblio/10110749

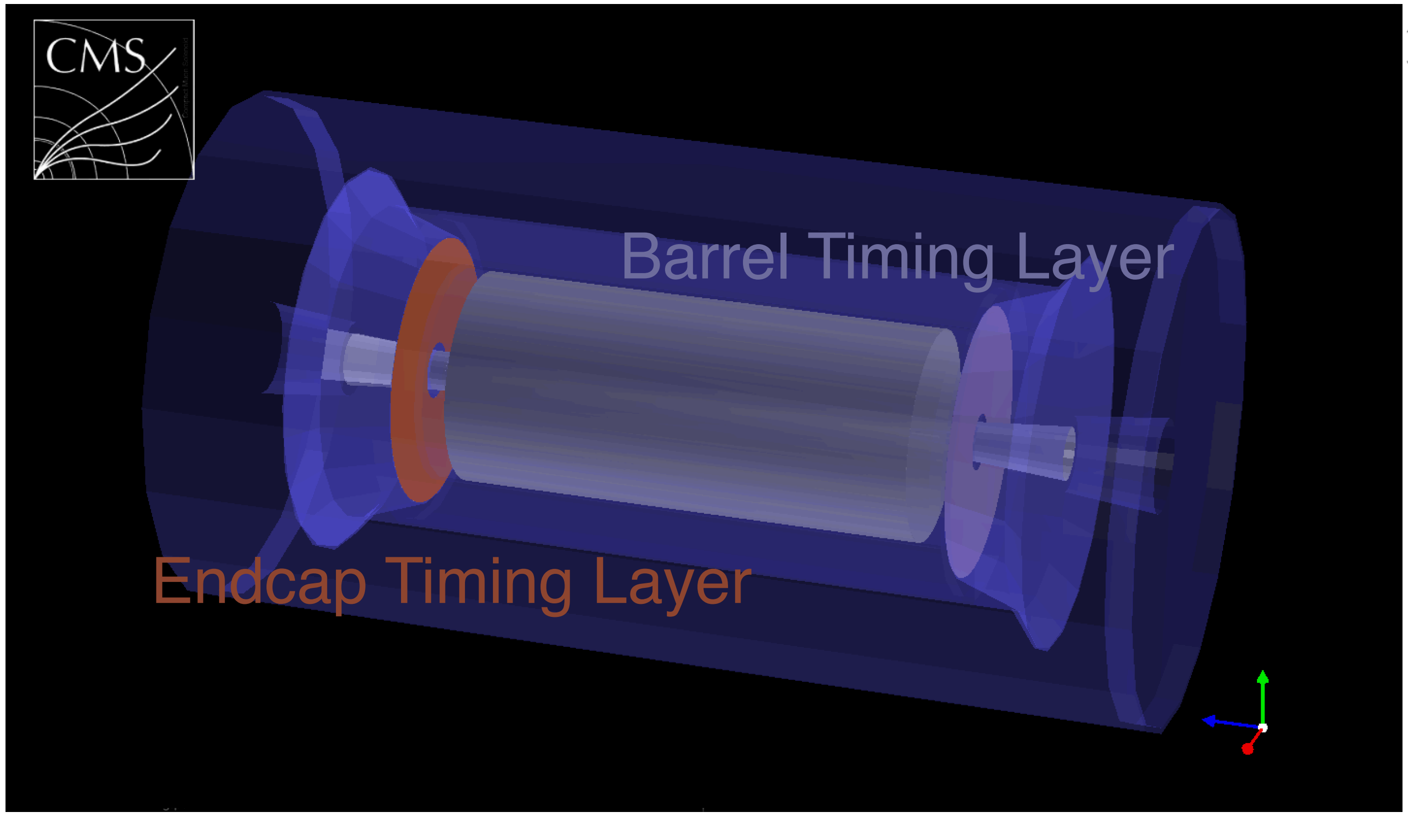
Object Condensation Performance

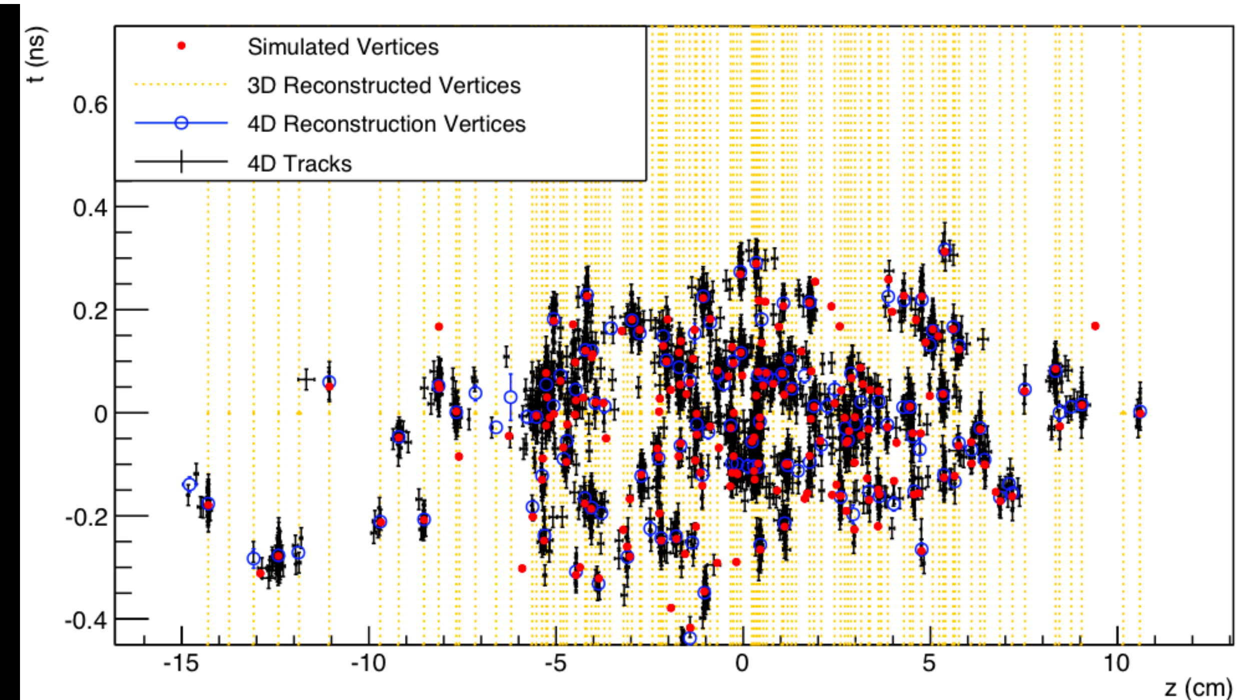
- Object condensation reconstructs individual particles significantly better
  - Even in dense multi-particle environments
  - Significant reduction in outliers





Timing in Tracking for HL-LHC





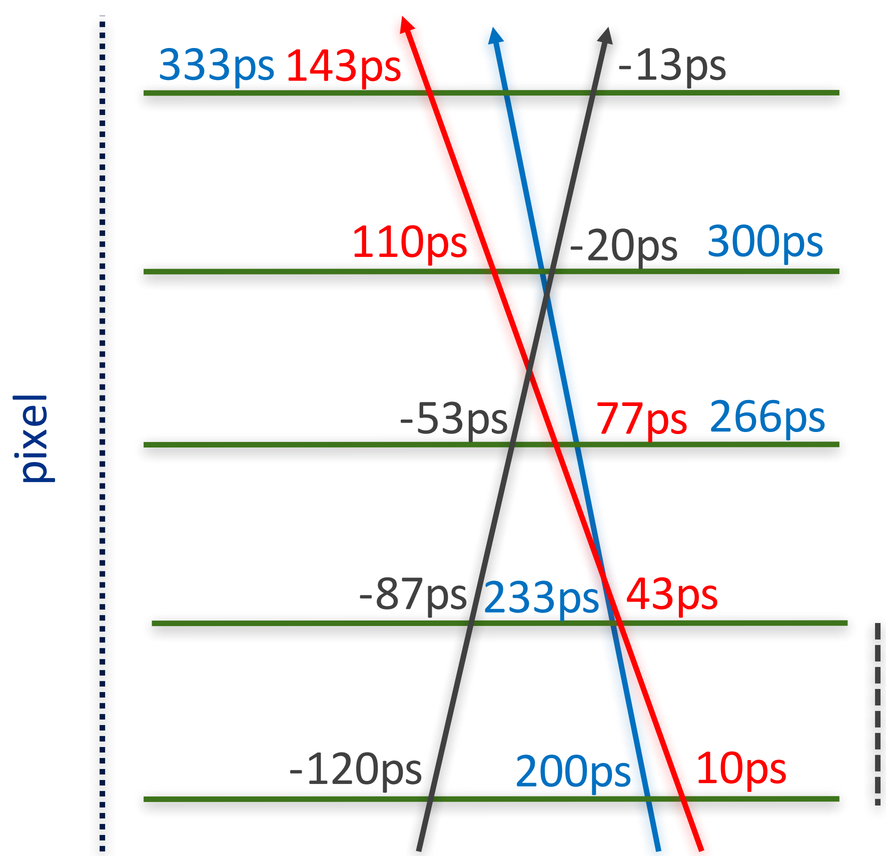
MTD TDR CMS-TDR-020

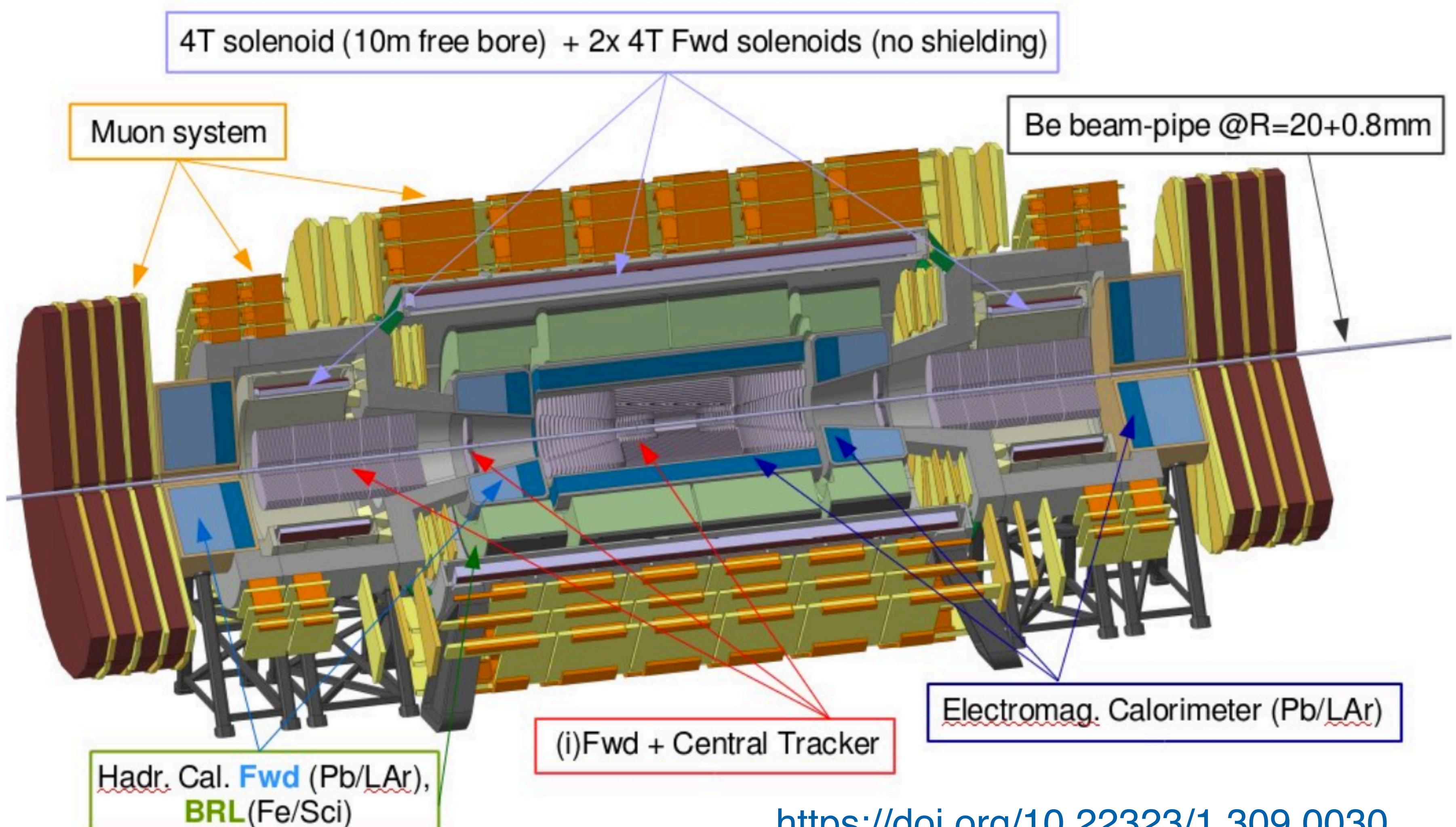
- CMS - MIP Timing detector
  - Strategy is to match precision timing hits to inner-detector tracks, back propagate time to vertices with ~30ps precision at beginning of life
  - Results in pileup removal in isolation cones, particle ID capabilities, excellent sensitivity to a variety of long-lived particles
  - Being integrated into general CMS tracking algorithms to make most informed choices
  - Higher-dimensional data leads to more easily discernible patterns
- Forward-only detector in ATLAS - HGTD to bolster forward tracking

Where granularity and timing in detectors is going:

- FCC-hh concept designs include timing in most or all layers of trackers extending to $|\eta| < 6$
  - Precision timing capabilities anticipated in all calorimetry as well, necessary to make sense of neutral particles in 1000 pileup, many billions of channels
- Timing pixel research ongoing, possible LHCb VELO upgrade

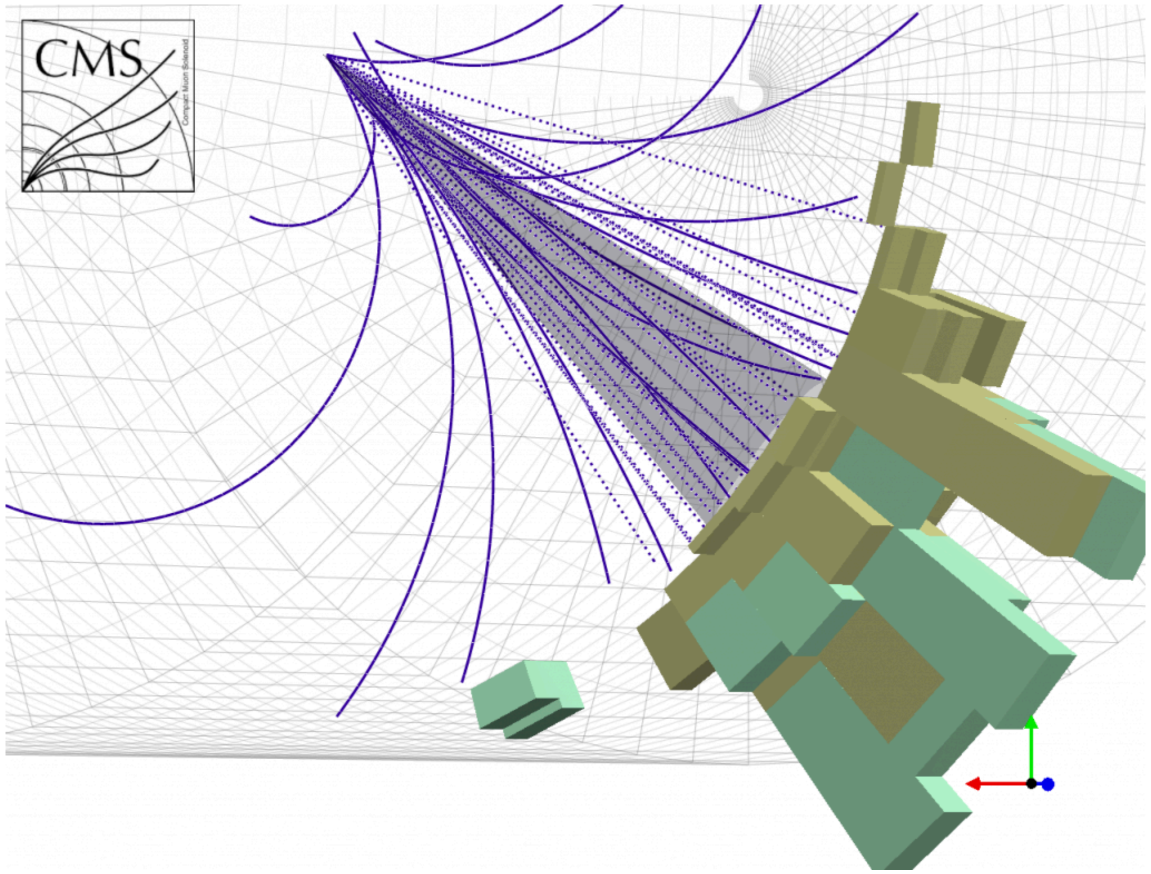
Assume σ_t = 30 ps per layer

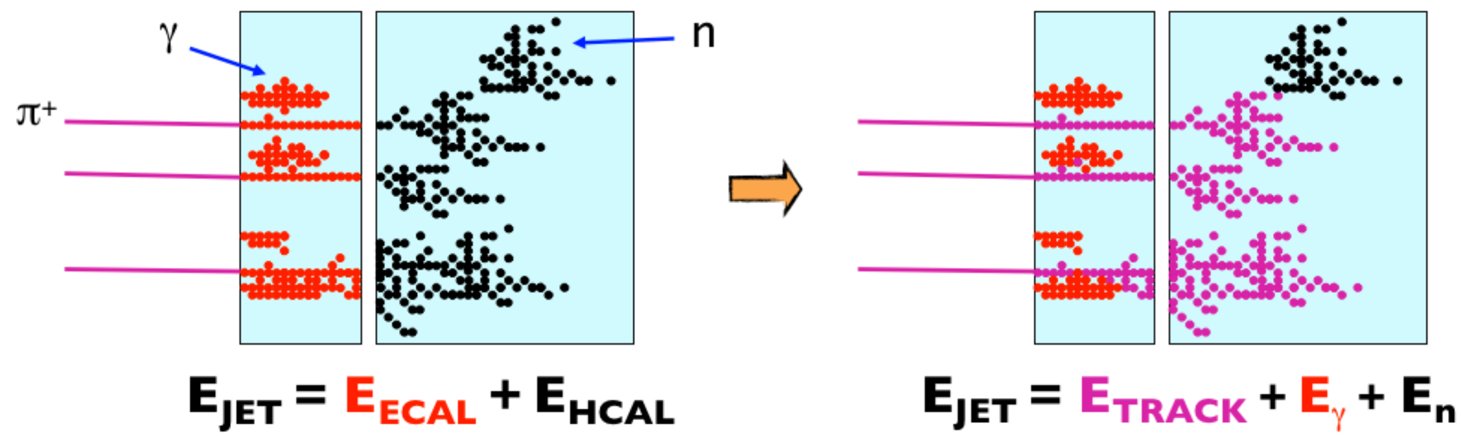




https://doi.org/10.22323/1.309.0030

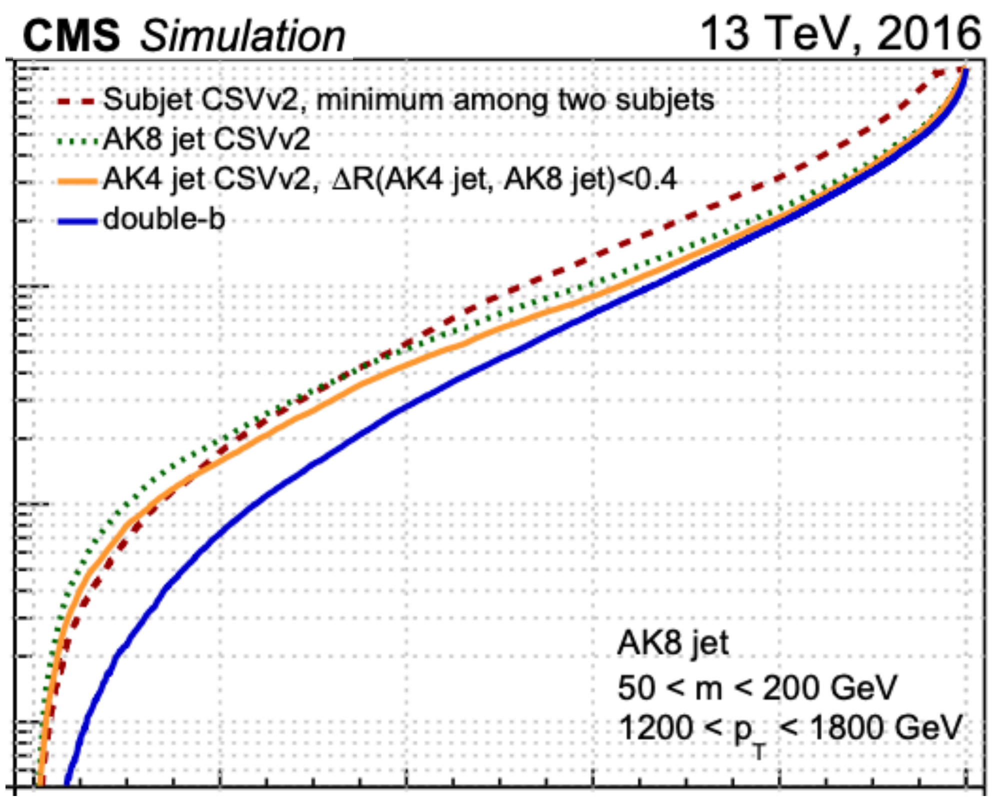
Algorithms govern detector performance more and more





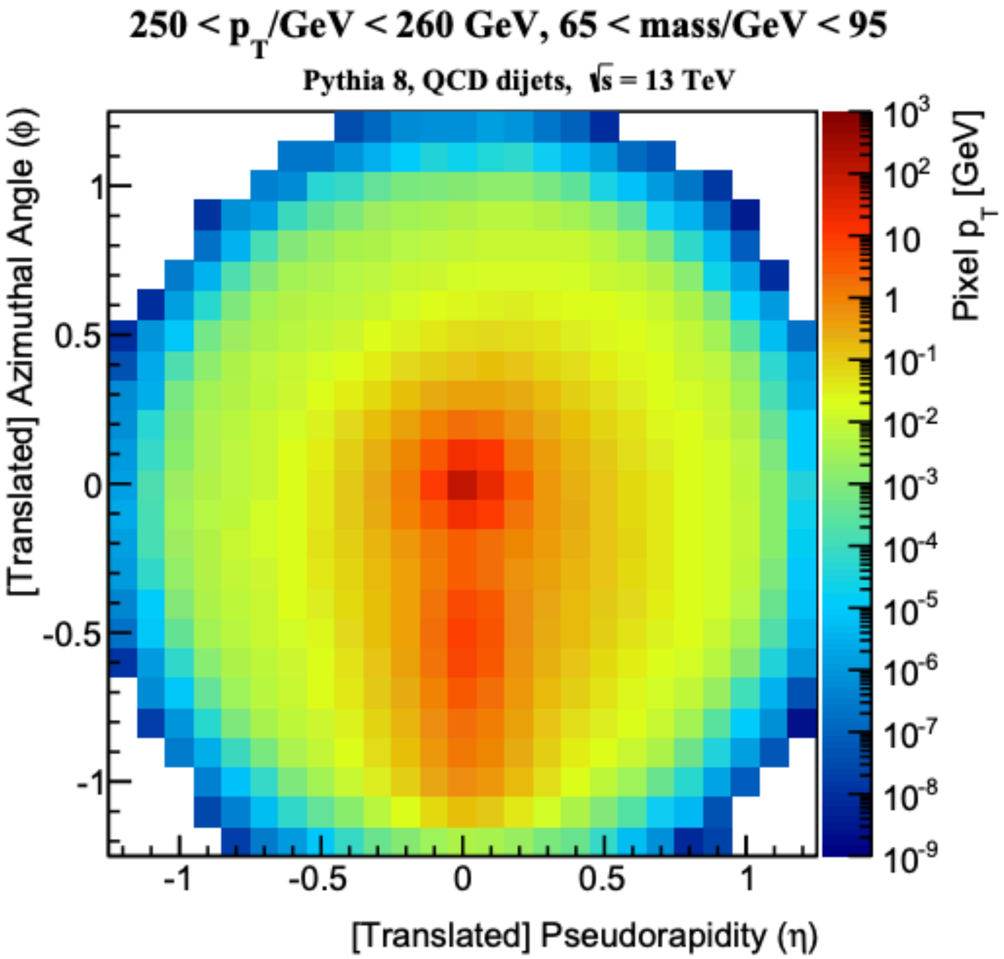
- Finer granularity yields more precise shower identification
  - At the cost of easily conceptualized energy summing rules (e.g. summing towers top left)
  - Now need algorithms to define bounding volumes, etc…
- Particle Flow algorithms help by associating tracking with calorimetry
  - Can use tracking information to bring additional topological information to clustering
  - Further identification of particles allows precise calibrations to be applied (top right)

Jet tagging and Jets as images



https://arxiv.org/abs/1712.07158

https://arxiv.org/abs/1511.05190



- Given the complex nature of jets, ML techniques have been commonplace
  - Every particle in a jet has some information about that jet’s nature

- Common uses in b-tagging, and more recently merged-jet tagging
  - Evolution from using the original jet clustering very rigidly to allowing the ML algorithm to pick out what relationships are interesting
  - Only possible with modern ML techniques like deep networks or CNNs

Jets aren't really images though!

- What is seen in the distribution of calorimeters and tracks is the outcome of relationships between hadrons and the QCD fluctuations that made them

- Jet formation is modeled well by a series of nested branchings of QCD splitting functions
  - This is where the real information about the jet "lives"

- It would be better to try to learn classifying information using this tree of splittings
  - It is more fundamentally related to the physics, a more clear "representation" of the data

- This is a graph, not an image!
  - Moreover, this graph can vary from event to event and jet to jet
  - How can we get around this since the techniques we have so far are static?

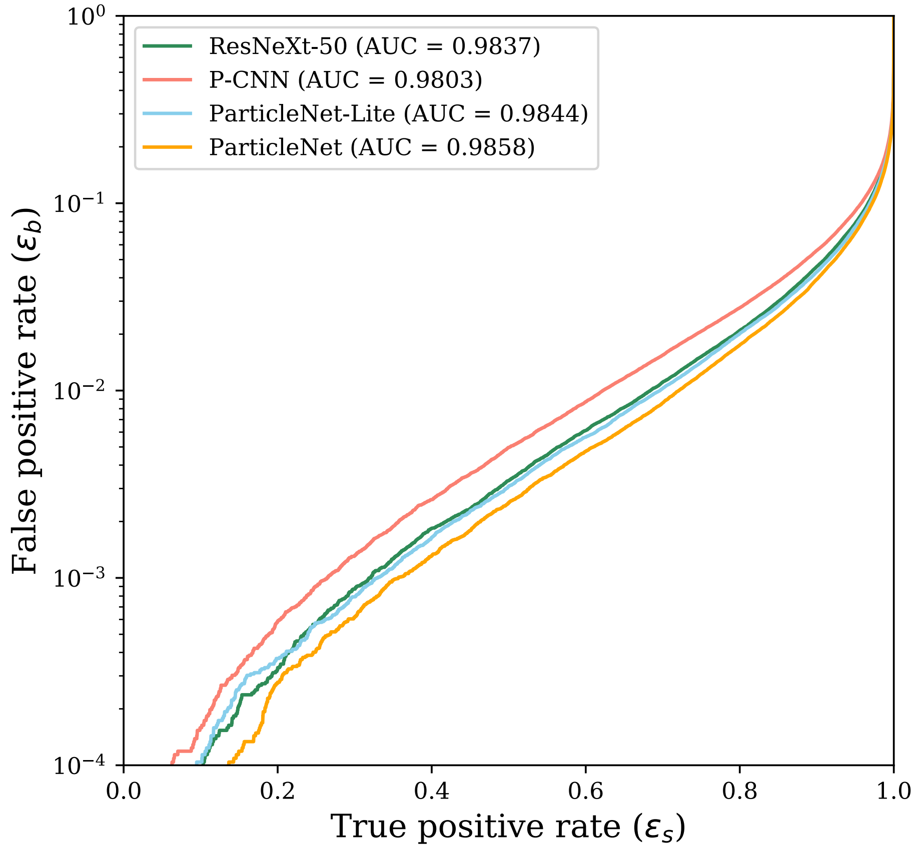
- A solution lies in Graph Neural Network (GNN) techniques

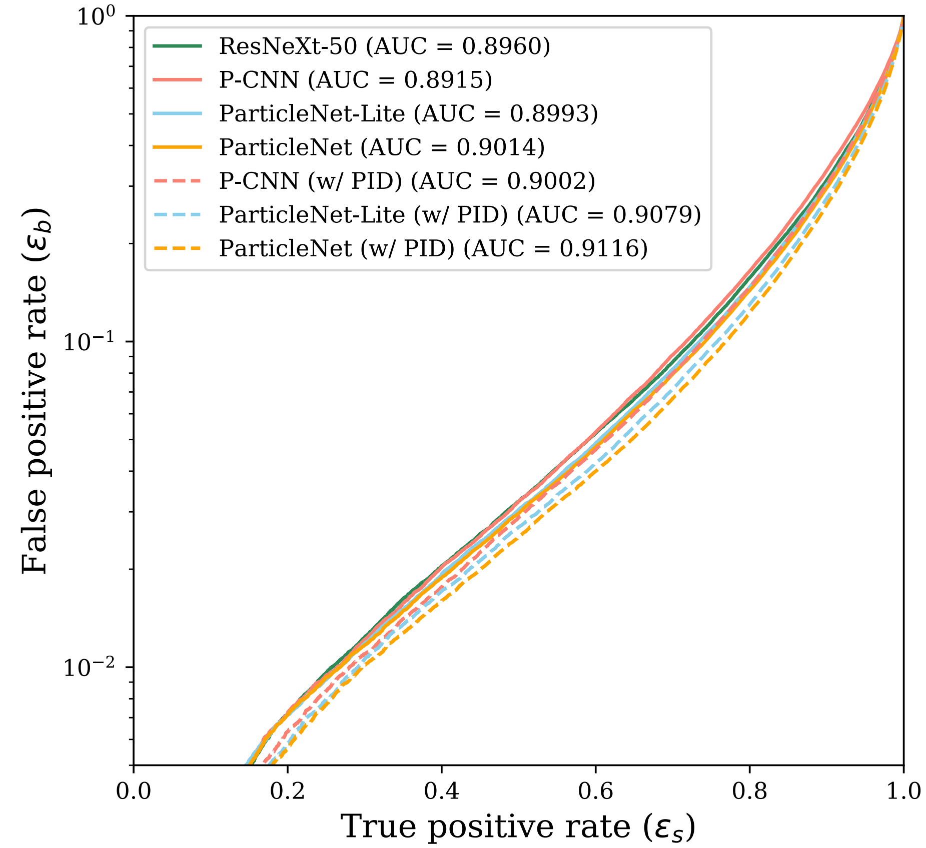
How graph neural networks are more like general programming

- From the two prior slides - GNNs provide a way to dynamically encode relationships between pieces of data
  - This is the equivalent of loops with nested if-statements, compared to more static fully connected or convolutional networks
- Each operation on the graph drives a new set of decisions based on a ruleset that is learned by the neural network
  - Specifically the network within a GNN making themessages which are passed
- This means that significantly more complex processes can be encoded more precisely by representing recurrent relationships in the structure of the model itself
  - Rather than having to learn it by example through training.

Jet tagging using graphs

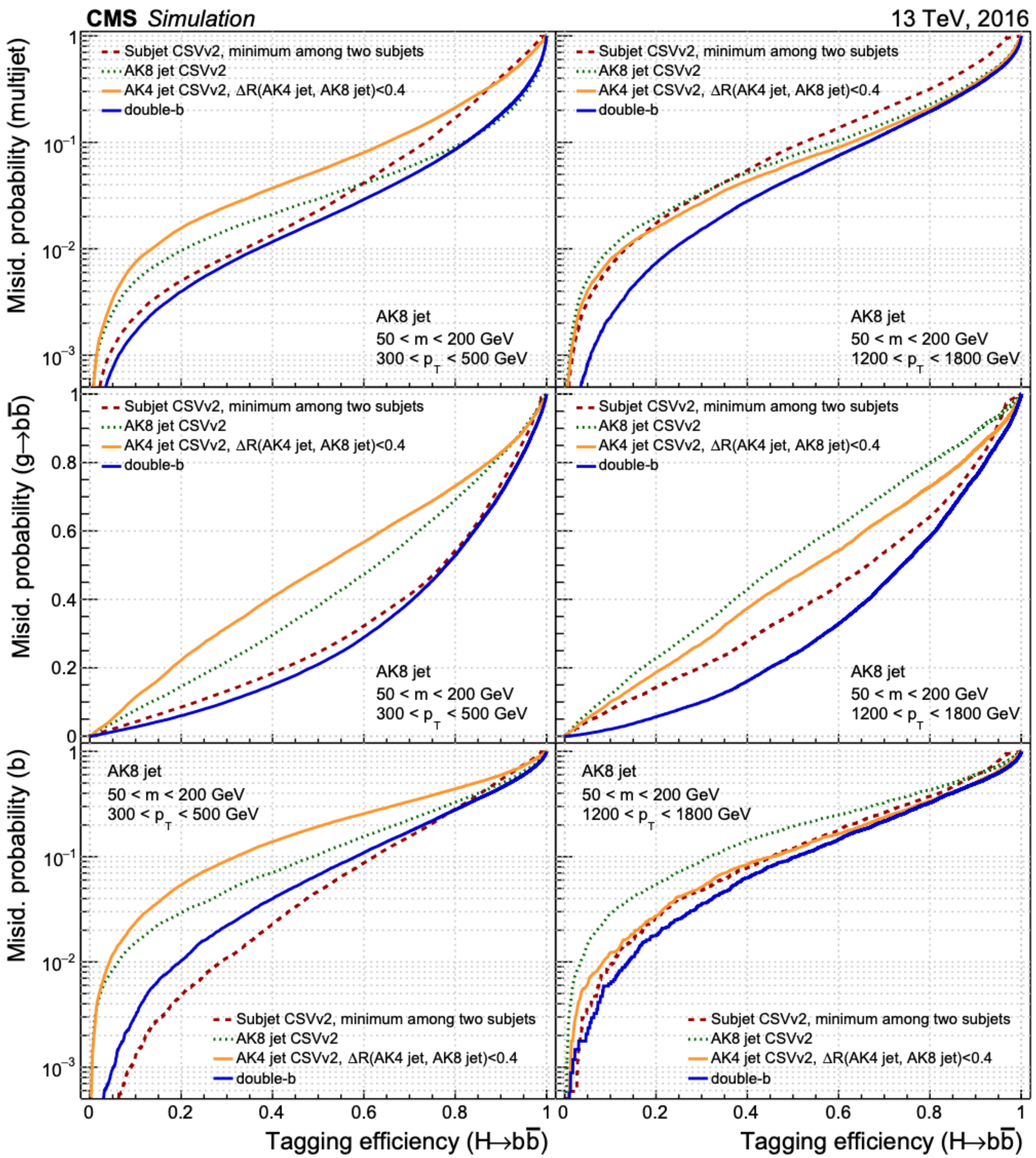
- Below are results for using a graph to describe associations in jets
  - Improved performance with respect to image-based architectures!
- Train the classifier to learn what connections between particles are important
- Less need to pre-process the input properties for classification
  - Little to no transformation of data about jet constituents





https://arxiv.org/pdf/1902.08570.pdf

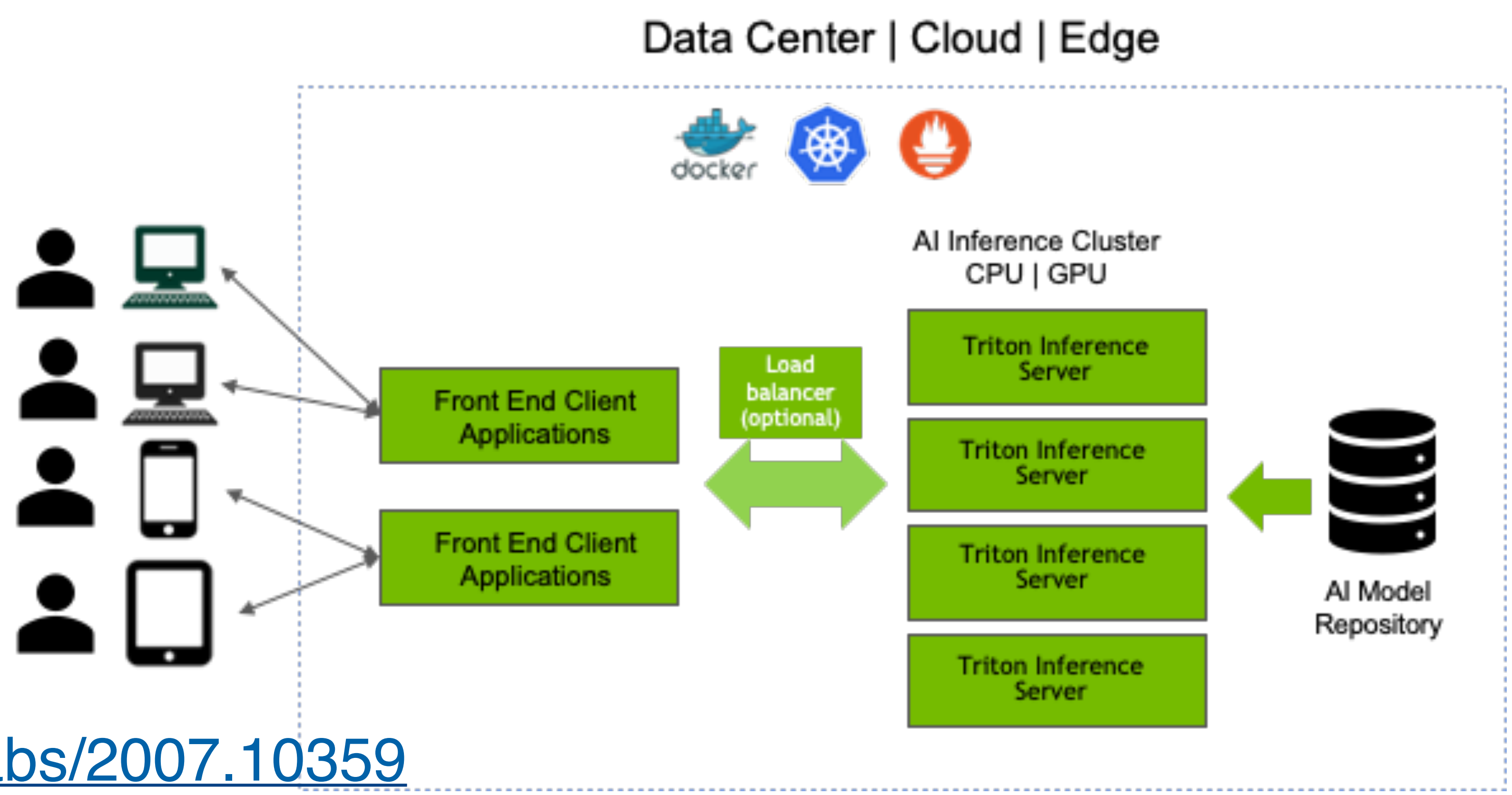
Full results for deep double b-tag



Fermilab

Deploying these techniques in the experiments (I)

- Experiment software stacks are often difficult to deal with, but are requiring more and more ML inference as part of their standard operation
  - Difficult to update to cutting edge software without expert knowledge
  - Already many moving parts, difficult to maintain
  - Adding machine learning frameworks to this means even more complexity!
- We are exploring decoupling this by using machine learning *as a service*
  - Experiment framework then makes standardized, lightweight api calls to a separate server running a big stack of GPUs and all the models one may require for reconstruction



https://arxiv.org/abs/2007.10359

Deploying these techniques in the experiments (II)

- GNNs, being rather new, were not readily compatible with inference as a service frameworks
  - Often required some contortion or limitation on how you were defining the model
  - Makes the process of model development and maintenance a pain!
- Together with authors of a GNN package for pytorch we developed a way to make the models immediately deployable
  - With no code changes, access to these powerful models to experiments is fairly easy
  - First large-scale tests of GNNs in CMSSW using models from this talk underway

```
import torch
import torch.nn.functional as F
from torch_geometric.nn import GCNConv

class Net(torch.nn.Module):
    def __init__(self, in_channels, out_channels):
        super(Net, self).__init__()
        self.conv1 = GCNConv(in_channels, 64)
        self.conv2 = GCNConv(64, out_channels)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index)
        x = F.relu(x)
        x = self.conv2(x, edge_index)
        return F.log_softmax(x, dim=1)

model = Net(dataset.num_features, dataset.num_classes)
```

change to

```
def __init__(self, in_channels, out_channels):
    super(Net, self).__init__()
    self.conv1 = GCNConv(in_channels, 64).jittable()
    self.conv2 = GCNConv(64, out_channels).jittable()
```

forward stays the same

```
model = torch.jit.script(model)
```

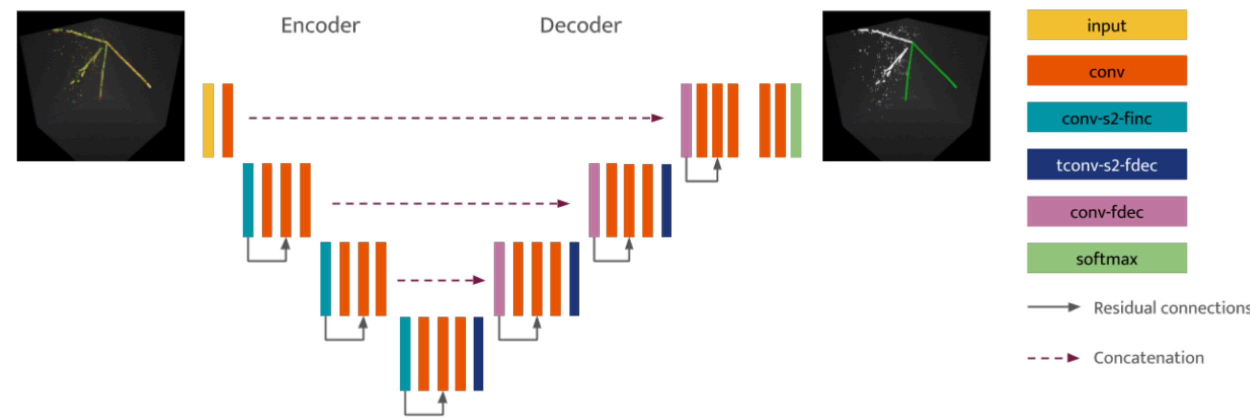
serialize jittable model

https://pytorch-geometric.readthedocs.io/en/latest/notes/jit.html

Other methodologies of tackling irregular detectors

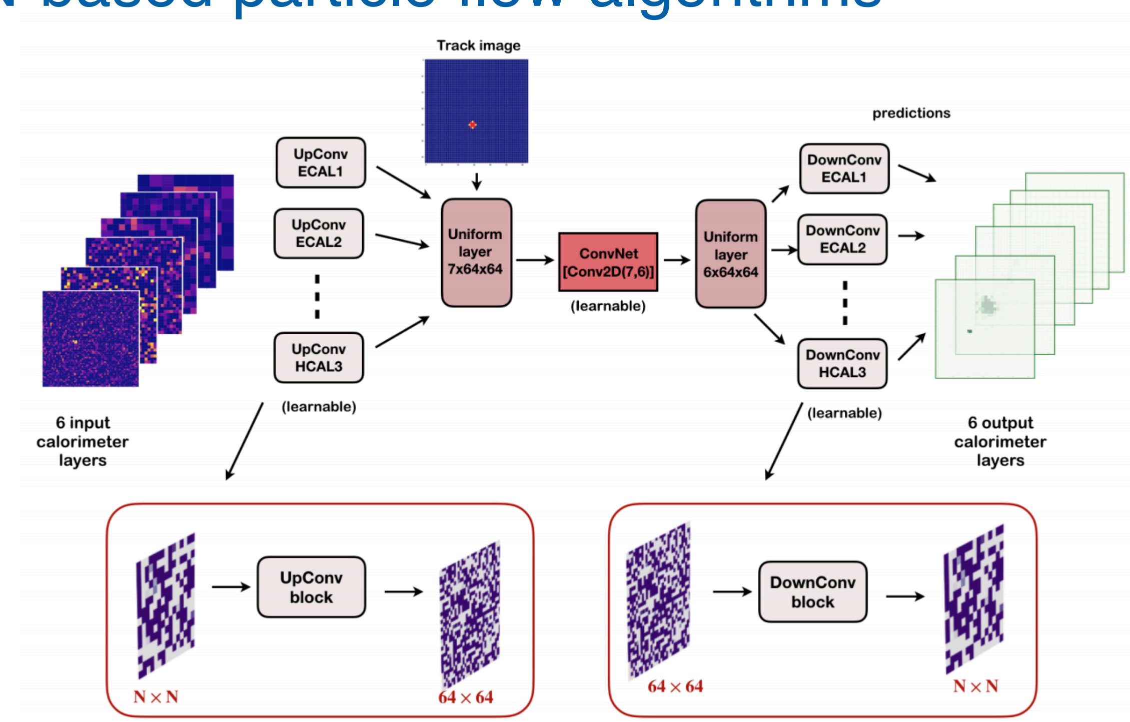
- LArTPCs and current calorimeters are being used to test ML based reconstruction as well
  - Early successes for convolutional based approaches
- ML based approaches starting to take the lead in neutrino physics
- Collider detectors exploring the use of CNNs and Graph techniques to reconstruction particle-level information
  - Similar or improved performance
  - Some issues still left: variable sized outputs

Sparse-CNN clustering for LArTPCs



https://arxiv.org/abs/1903.05663

CNN-based particle flow algorithms

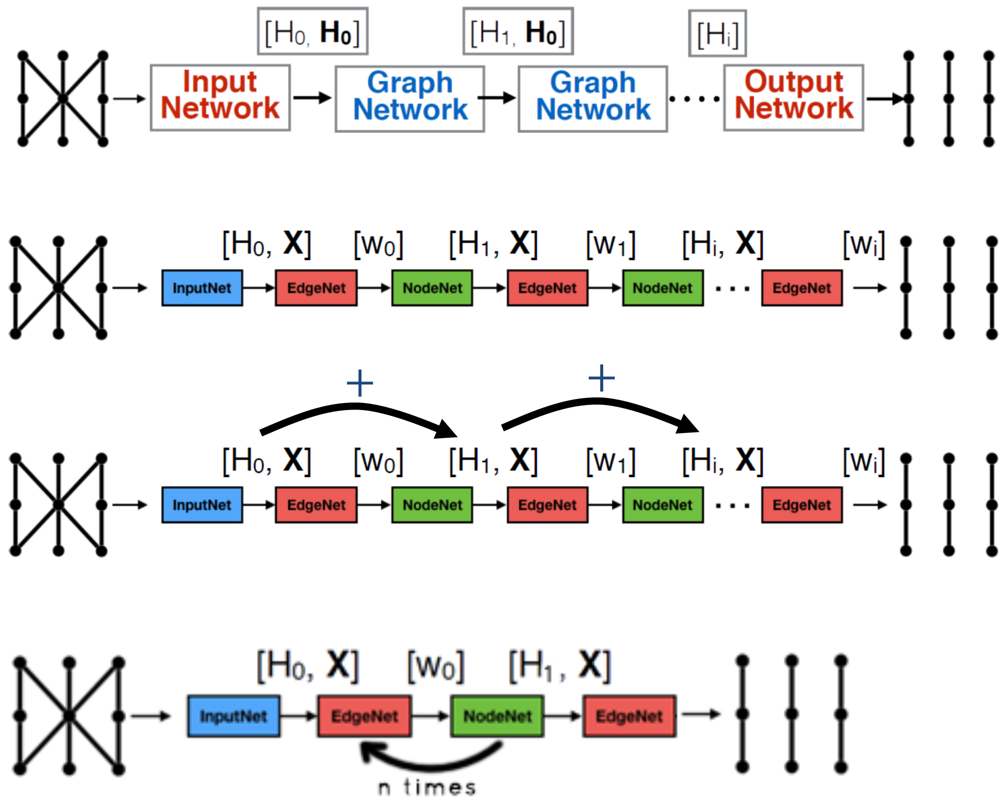


https://arxiv.org/abs/2003.08863

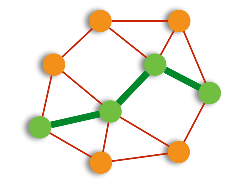
More methods yet to try and refine!

https://arxiv.org/abs/2003.08013

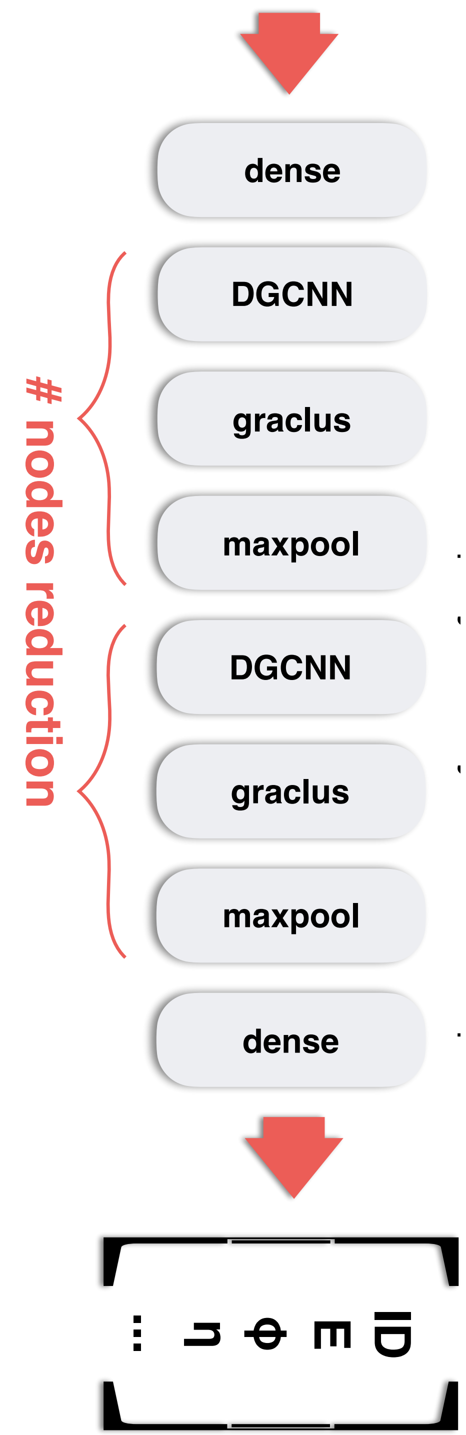
A variety of new message passing schemes (Exa.TrkX)



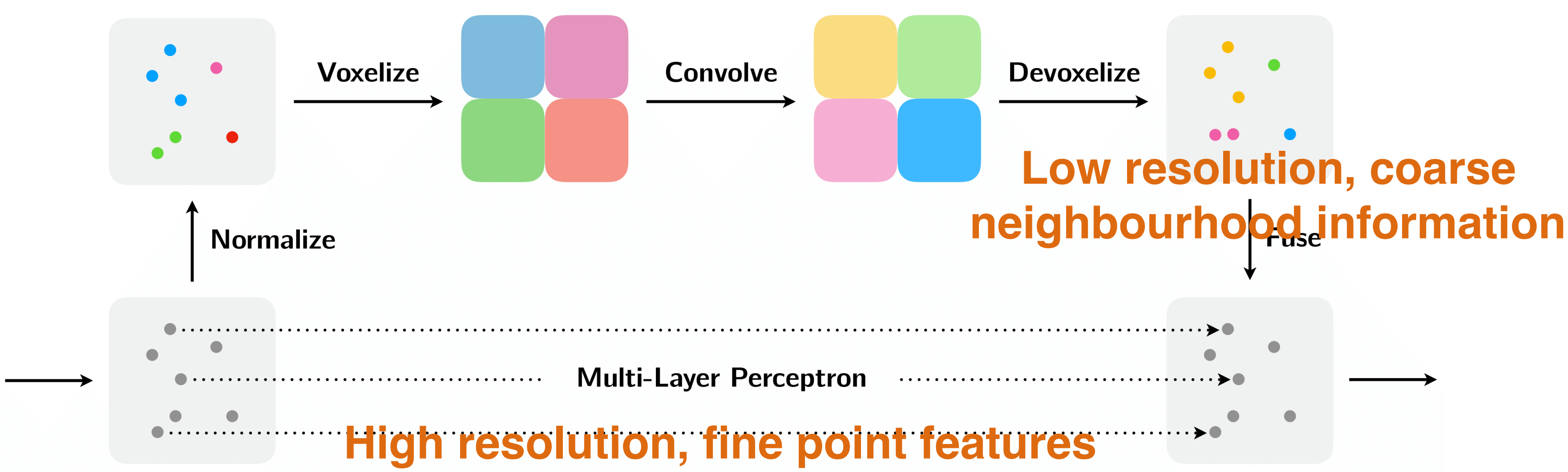
- Message Passing
- Attention Message Passing
- Attention Message Passing with Residuals
- Attention Message Passing with Recursion



Latent-space clustering



Voxelization for computing efficiency: https://arxiv.org/abs/1907.03739



Fermilab

Thoughts on using these techniques in analysis

- These GNN-based techniques will eventually be used directly in analysis
- In a very general sense we get the most statistical sensitivity in an analysis by choosing some number of physics-rule based categories
  - e.g. VBF, VH, boosted Higgs
  - Or, as in Higgs to 2 photons, having primary categories based off ML discriminator scores that tend to reflect reconstruction quality and detector performance

- However as the final states become more complex (6-8 jets) it becomes less and less efficient and accurate to do that sorting by hand using some heuristics
  - The probability to choose an improper combination of jets or leptons explodes as final state multiplicity increases

- Using a GNN in analysis to encode relationships of kinematic structure to play off kinematics vs. reconstruction quality vs. sensitivity to then create categories would help mitigate these combinatorial effects
  - For at worst a linear increase in the background (so ~sqrt(2) improvements)