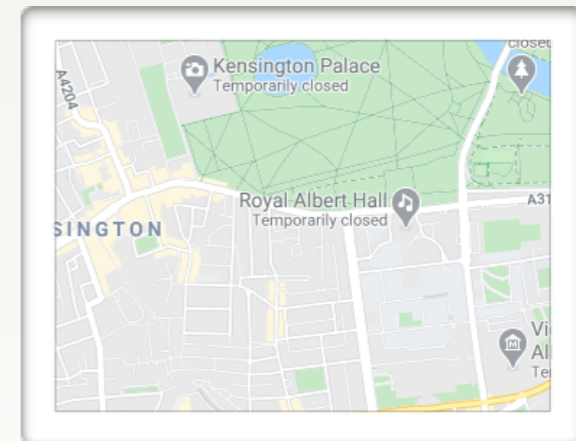
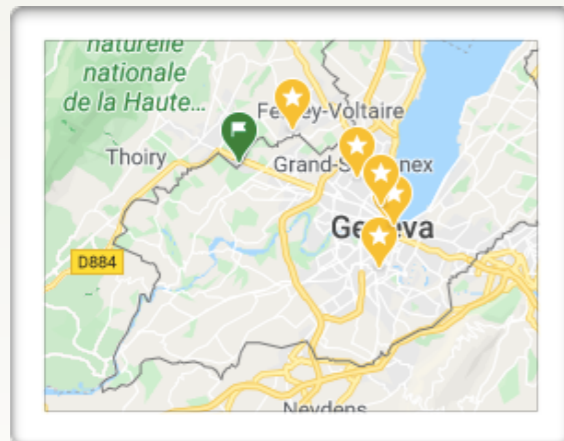


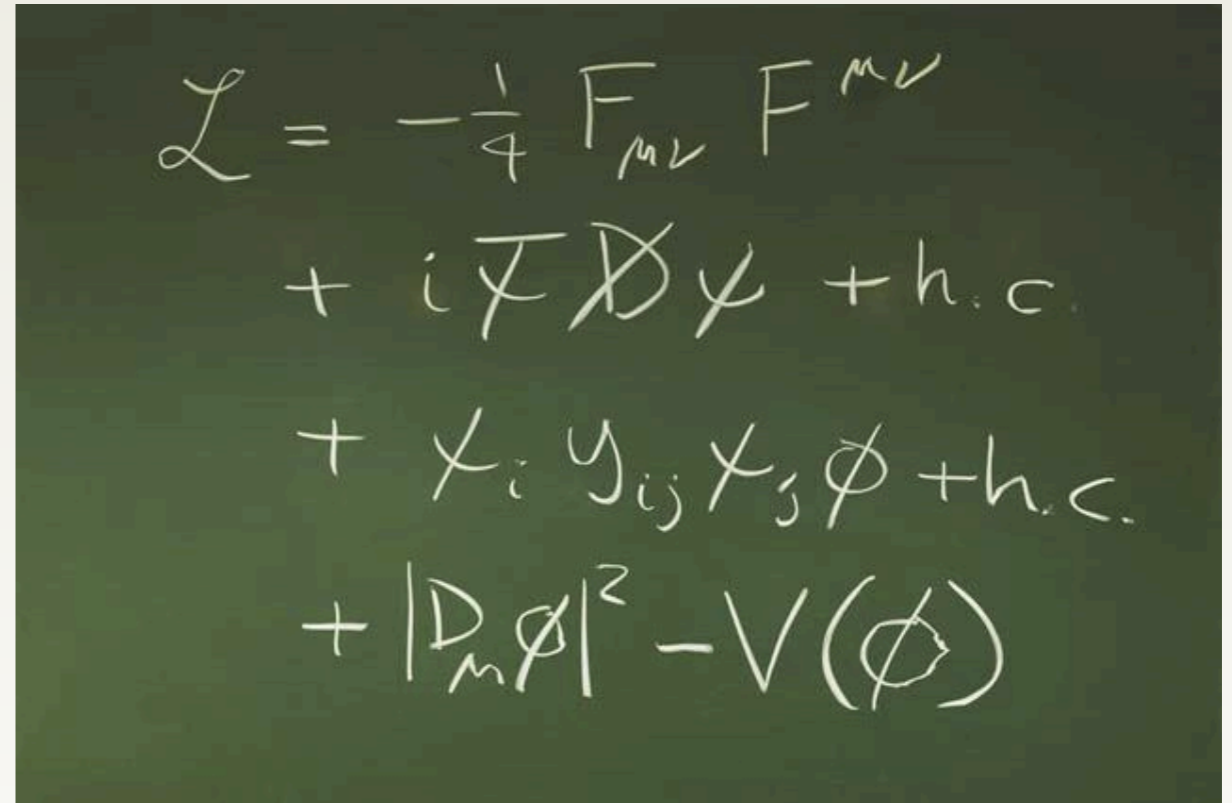
# Advanced deep neural networks for high-granularity calorimeters.

Jan Kieseler

22.05.2020



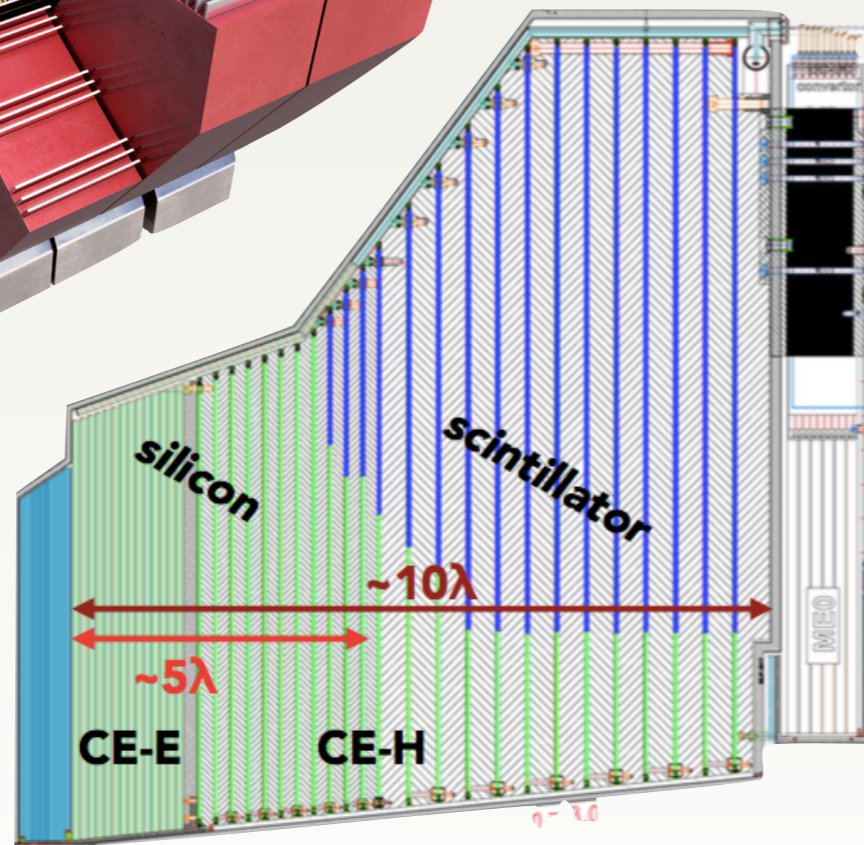
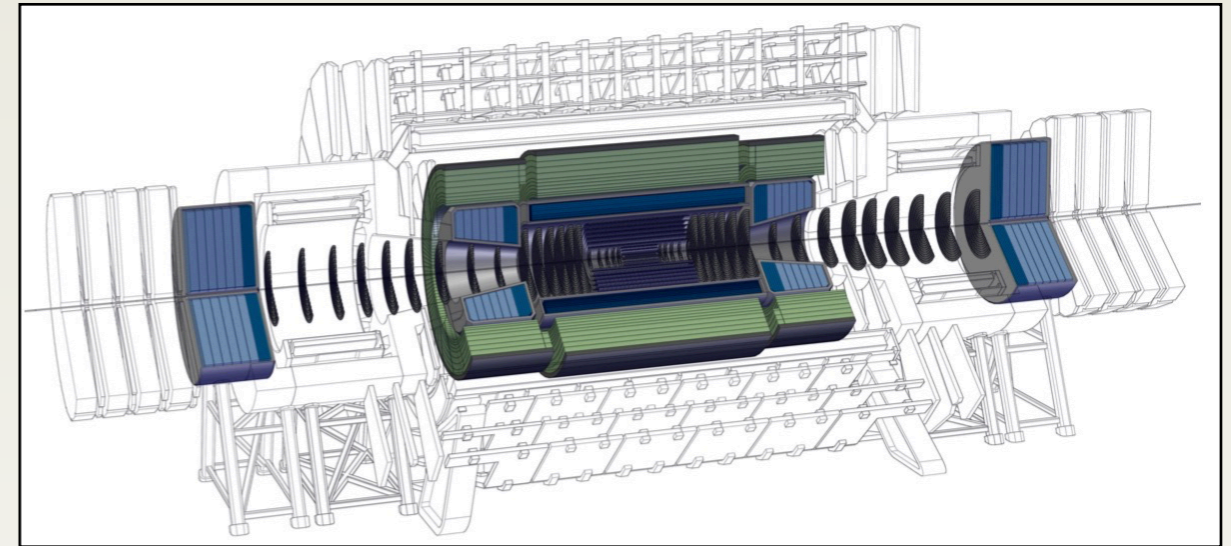
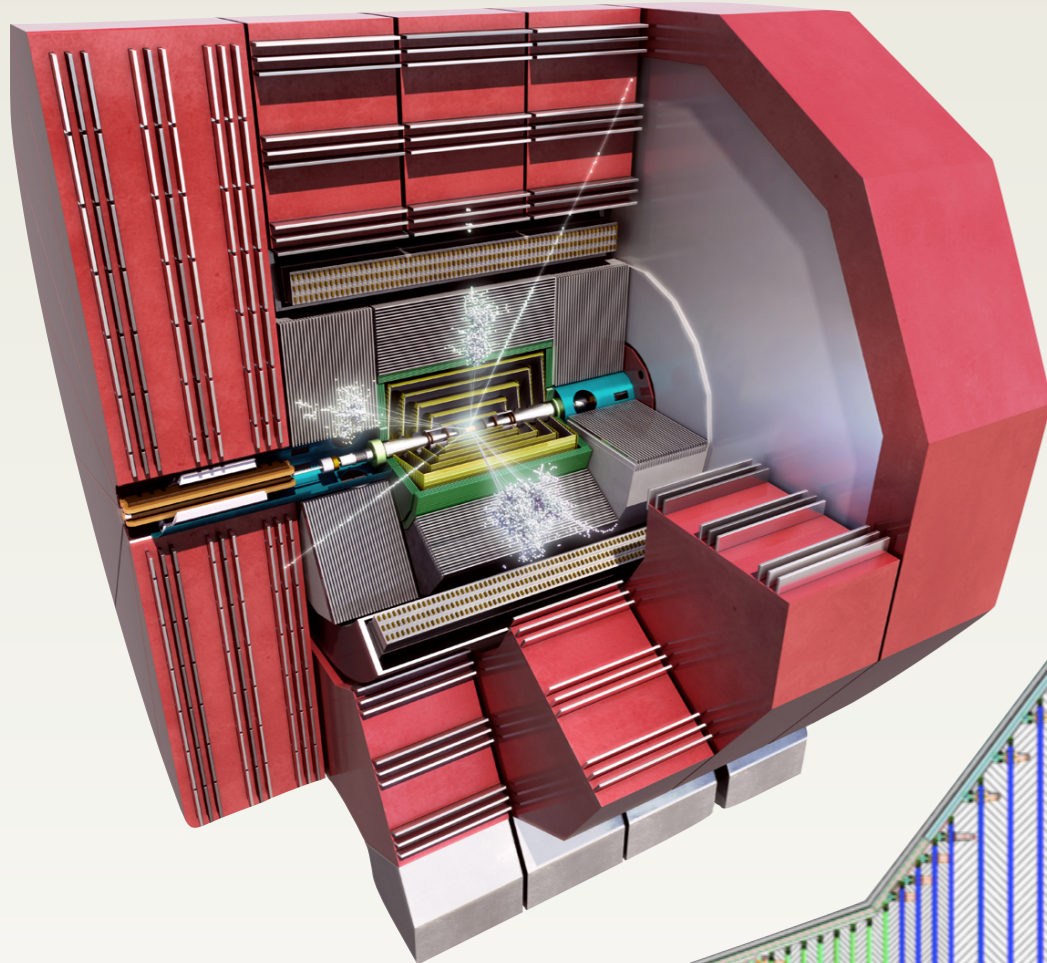
- Short introduction
- Exploiting shower shape variables
- Using hits in regular geometries
- Irregular geometries
- Seedless inference
- *More focus on techniques than on calorimeters or results*



$$\begin{aligned}
 \mathcal{L} = & -\frac{1}{4} F_{\mu\nu} F^{\mu\nu} \\
 & + i\bar{\psi} \not{D} \psi + \text{h.c.} \\
 & + \chi_i Y_{ij} \chi_j \phi + \text{h.c.} \\
 & + |D_\mu \phi|^2 - V(\phi)
 \end{aligned}$$

Image search using this talk's title  
(in February, on my laptop)

# High granularity calorimeters



CALICE, FCChh (barrel),  
CMS HGCal

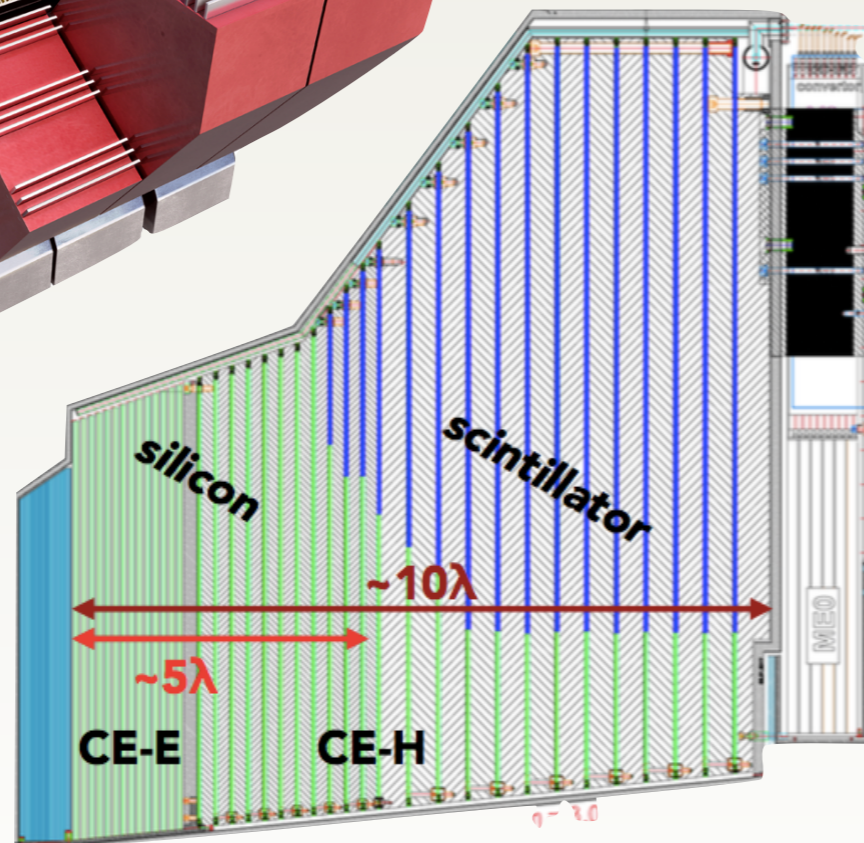
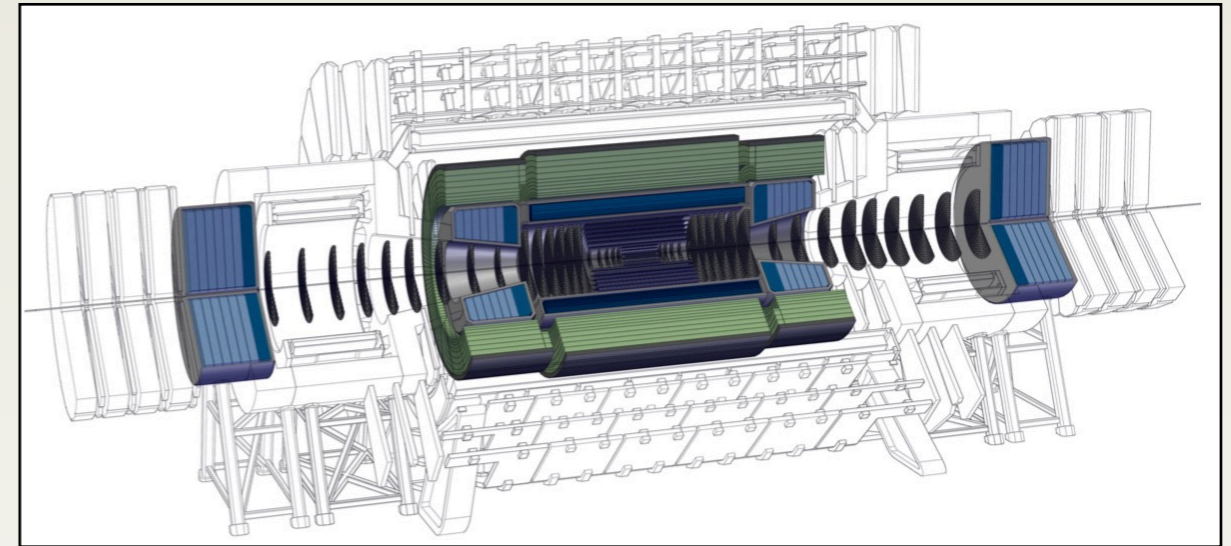
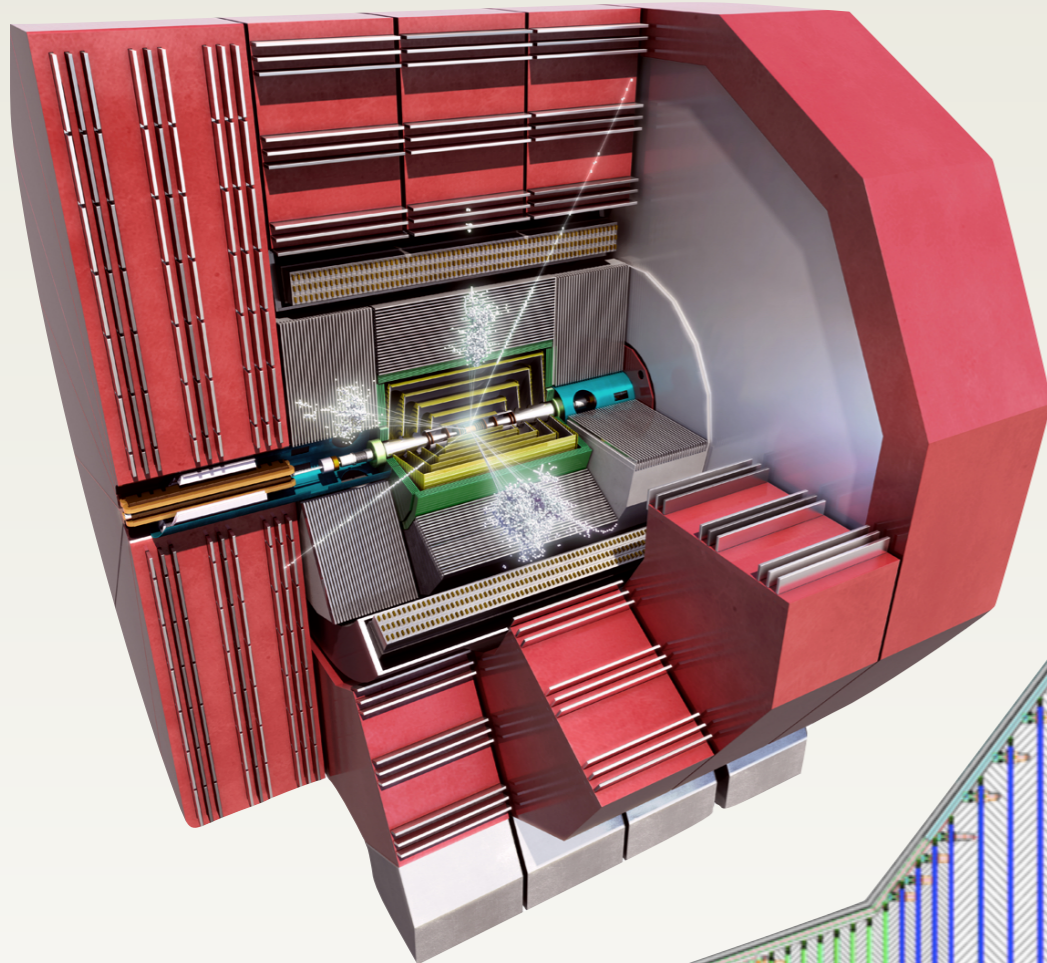
- In parts very different concepts
  - ▶ LAr,
  - ▶ Si (+SiPM)
  - ▶ SiPM
- However similar granularities
  - ▶ About 1cm x 1cm transversal (ECal)
  - ▶ > 10 layers longitudinal

M. Aleksa: <https://indico.cern.ch/event/838435>

F.Simon: <https://indico.cern.ch/event/838435>

CMS TDR 17-007

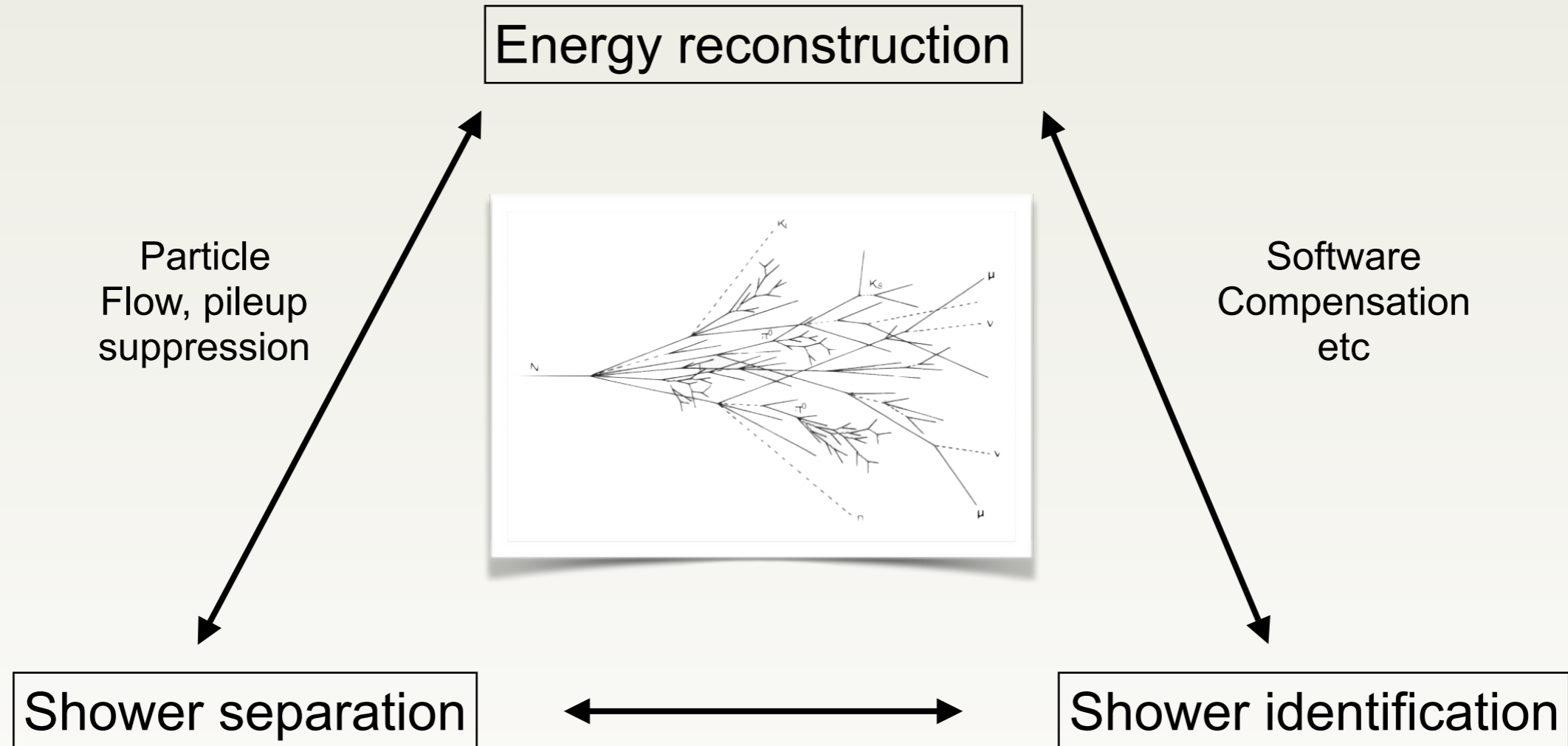
# High granularity calorimeters



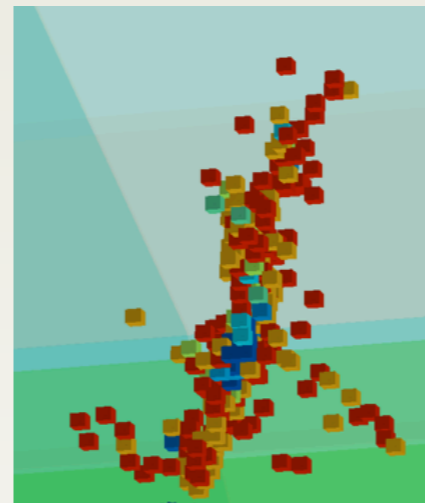
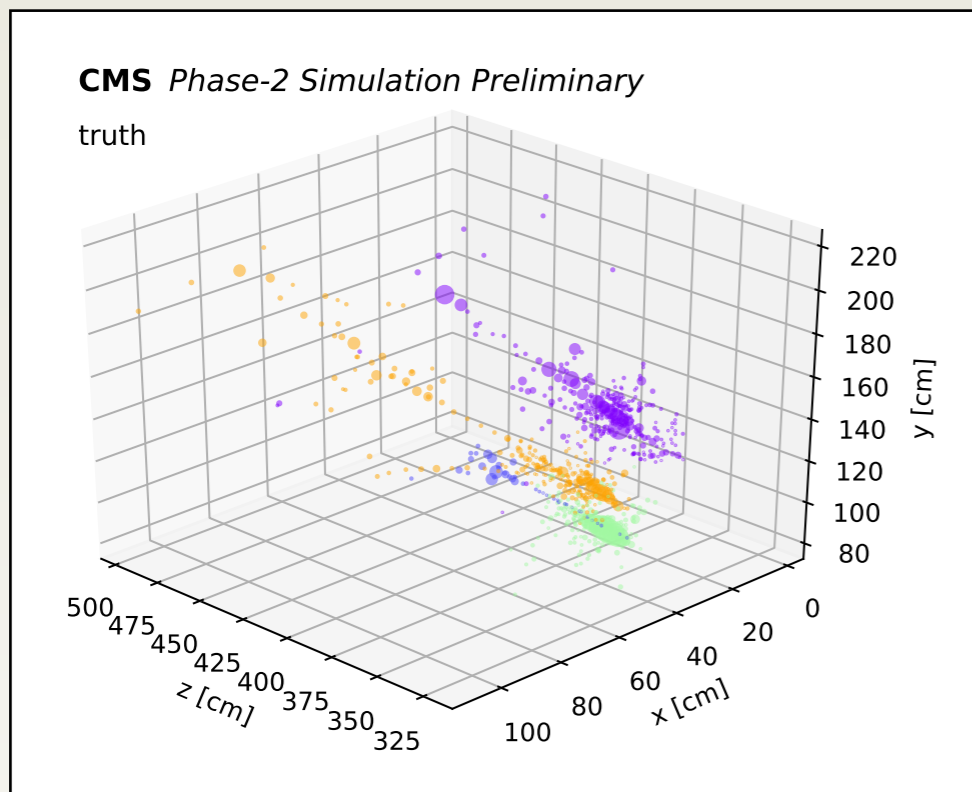
- Similar w.r.t. basic reconstruction concepts

- Handle Pileup
  - 200 (CMS) - 1000 (FCChh)
- High precision energy measurements
  - Missing energy/precision resolution
- Fully consistent Particle Flow
- Particle ID
  - Also part of software compensation
- Fully utilise timing

# Calorimeter Reconstruction



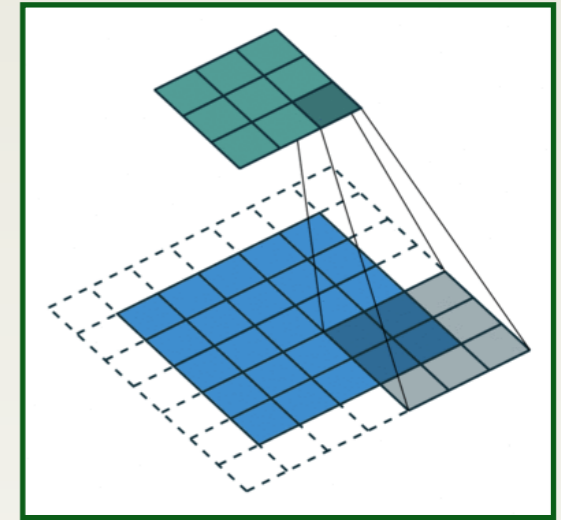
# HG calorimeters and ML



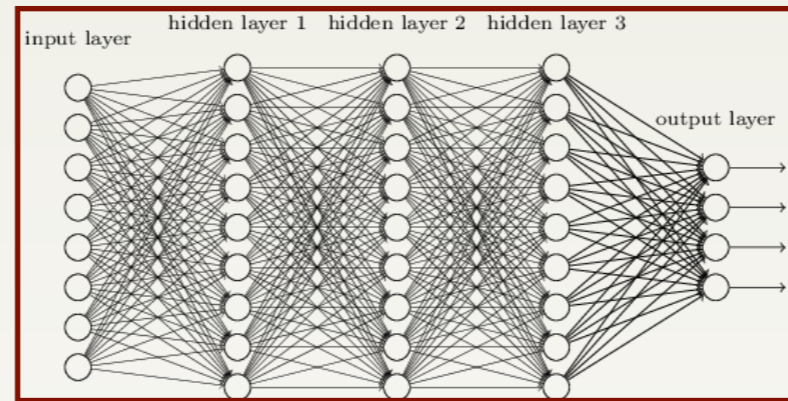
- High granularity calorimeters produce 3D/4D images of showers
- Deep neural networks have made many advances possible in the last years
  - ▶ Image classification, face recognition, ....., self-driving cars, ...
  - ▶ More and more applications in HEP (jet-tagging,...)
- Very powerful where 'things get messy': e.g. real cows versus spheric cows in vacuum

# Basic DNN building blocks

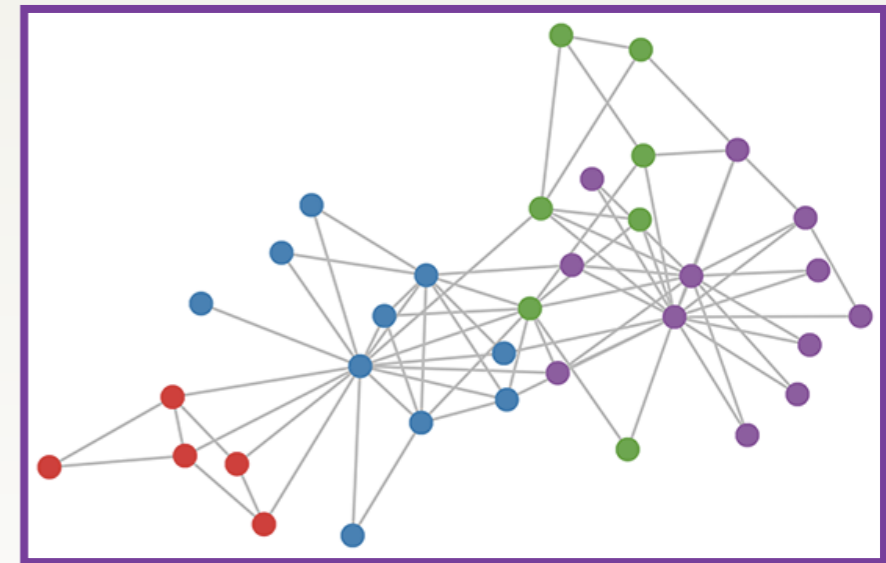
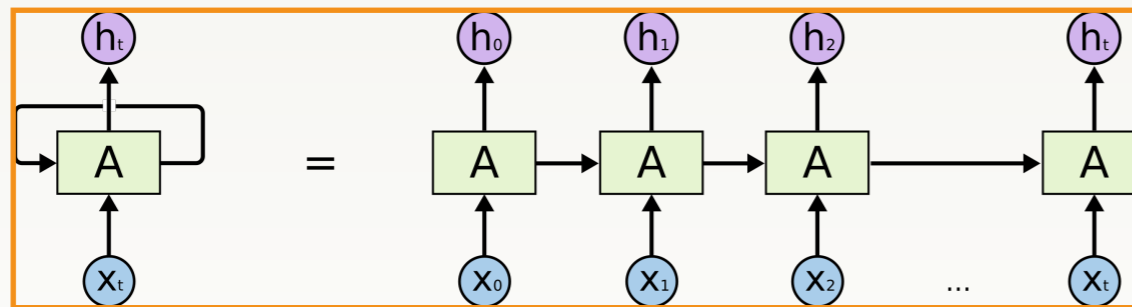
- Three off-the-shelf DNN types / building blocks
  - ▶ Fully connected 'dense' (very powerful but many parameters)
  - ▶ **Recurrent** ('time' series, good for sparsity, less parallelisable)
  - ▶ **Convolutional** (translation invariant structures, *key to image processing*)



- Rather recent developments: **Graph** neural networks
  - ▶ Will cover details later



- All mostly matrix multiplications
  - ▶ Fast and parallelisable
- Approximate an unknown function: *structure is the key!*

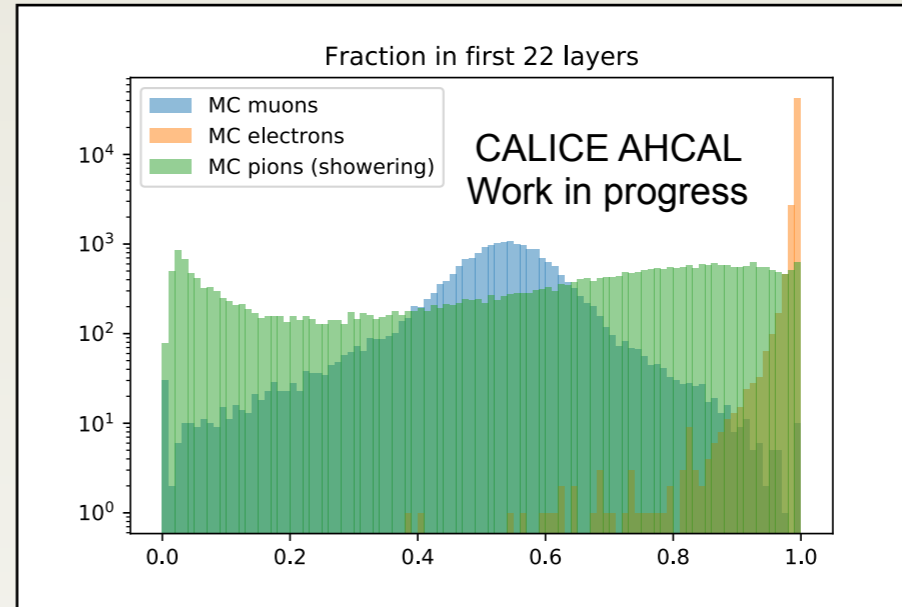
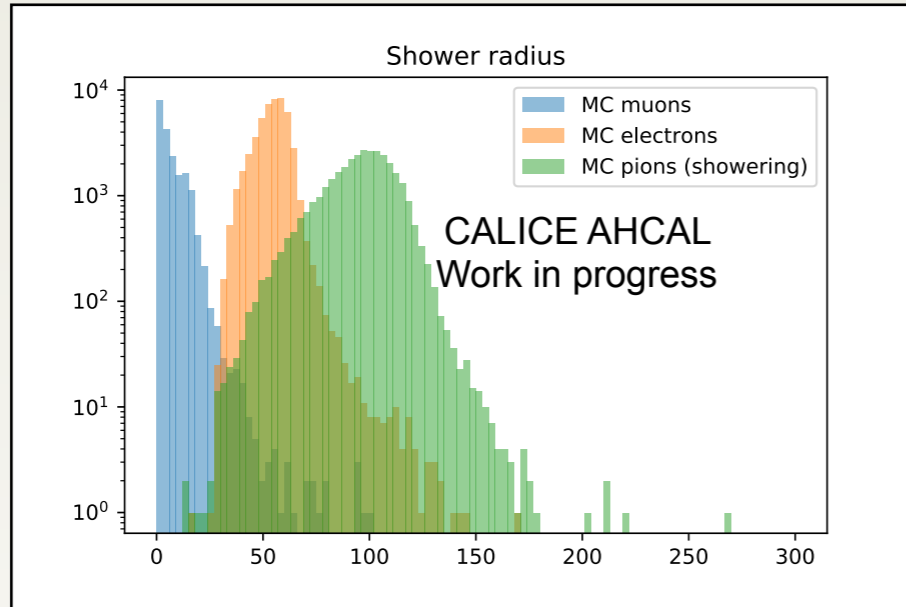


- Trained by minimising a loss function

Adam: D. Kingma, J. Ba, arXiv:1412.6980, conf. paper  
 AdaGrad: J. Duchi, E. Hazan, Y. Singer (2011)  
 RMSProp: T. Tieleman, G. Hinton (2012)  
 Stochastic gradient descent: H. Robbins; S. Monro (1951)

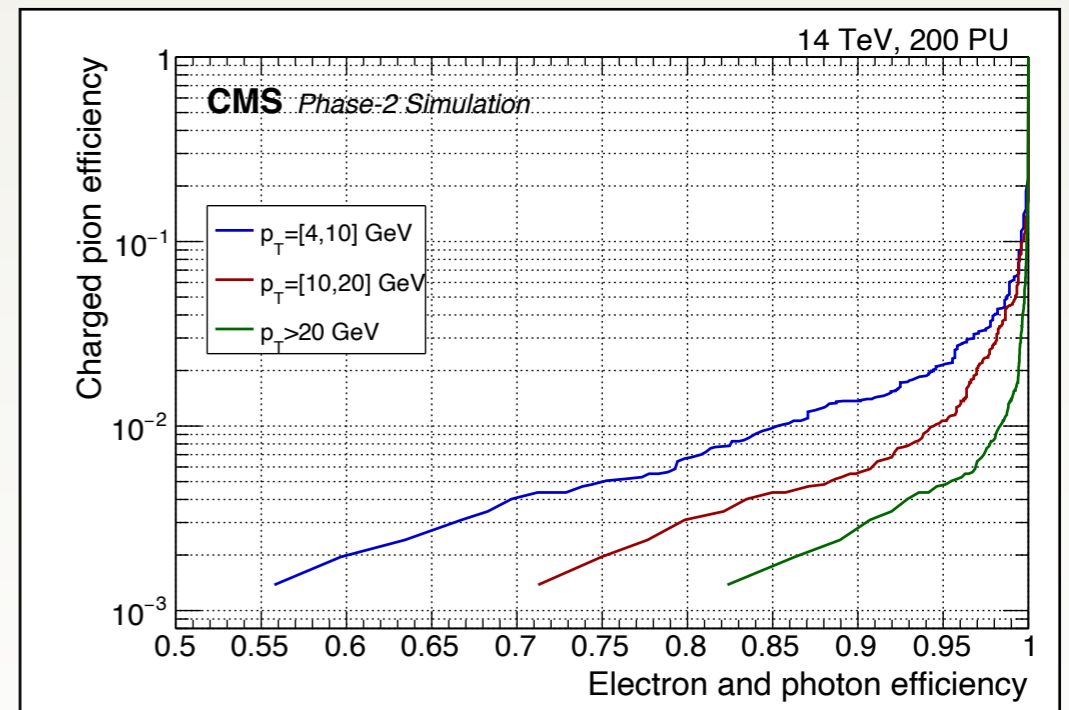
# Particle Identification

- Most important: separate EM showers from hadronic showers
  - ▶ Utilise global shower shape variables



- ▶ Process *individual hits* with DNNs based on off-the shelf convolutional layers as used for computer vision

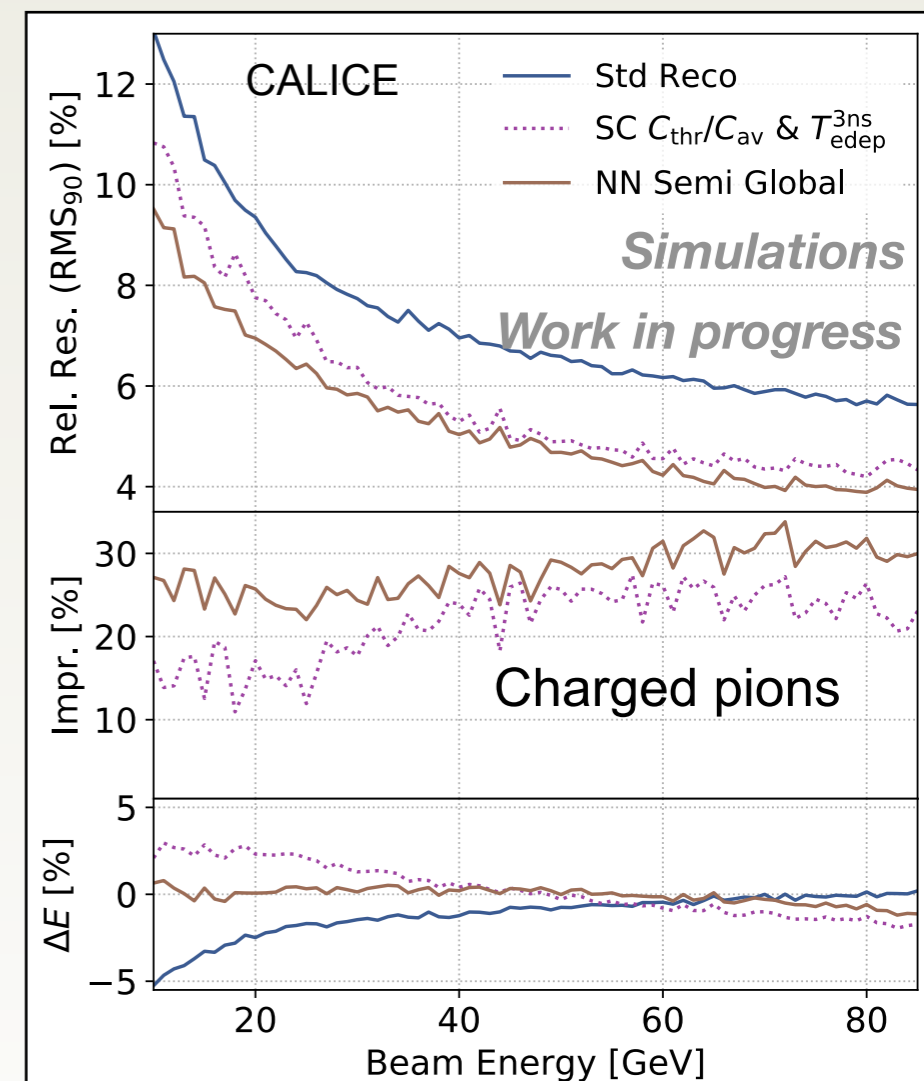
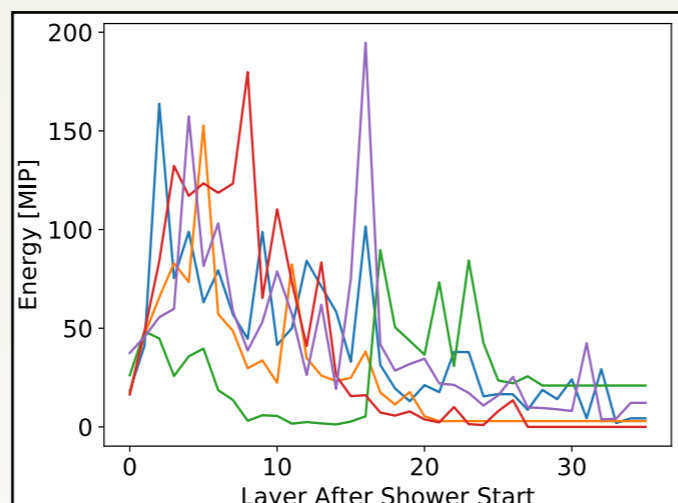
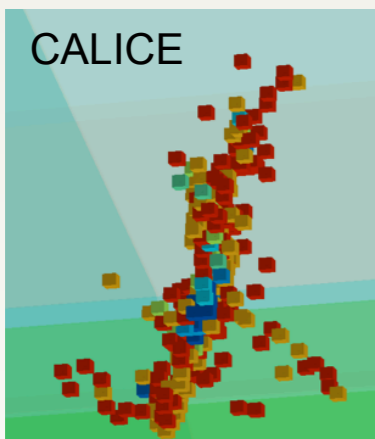
- High performance particle classification even in high pileup environments is possible already using off-the shelf architectures



Plots: V. Bocharnikov,  
CMS TDR-17-007

# Software compensation

- Separate electromagnetic and hadronic components
  - ▶ Strongly increased resolution for hadron showers
- Human engineered:
  - ▶ weight EM components less than hadronic components
  - ▶ Identify EM components by local energy density



- Machine-learning based
  - ▶ Consider shower shapes, in particular longitudinal
  - ▶ Feed in dense NNs

plots: C. Graf

# Software compensation

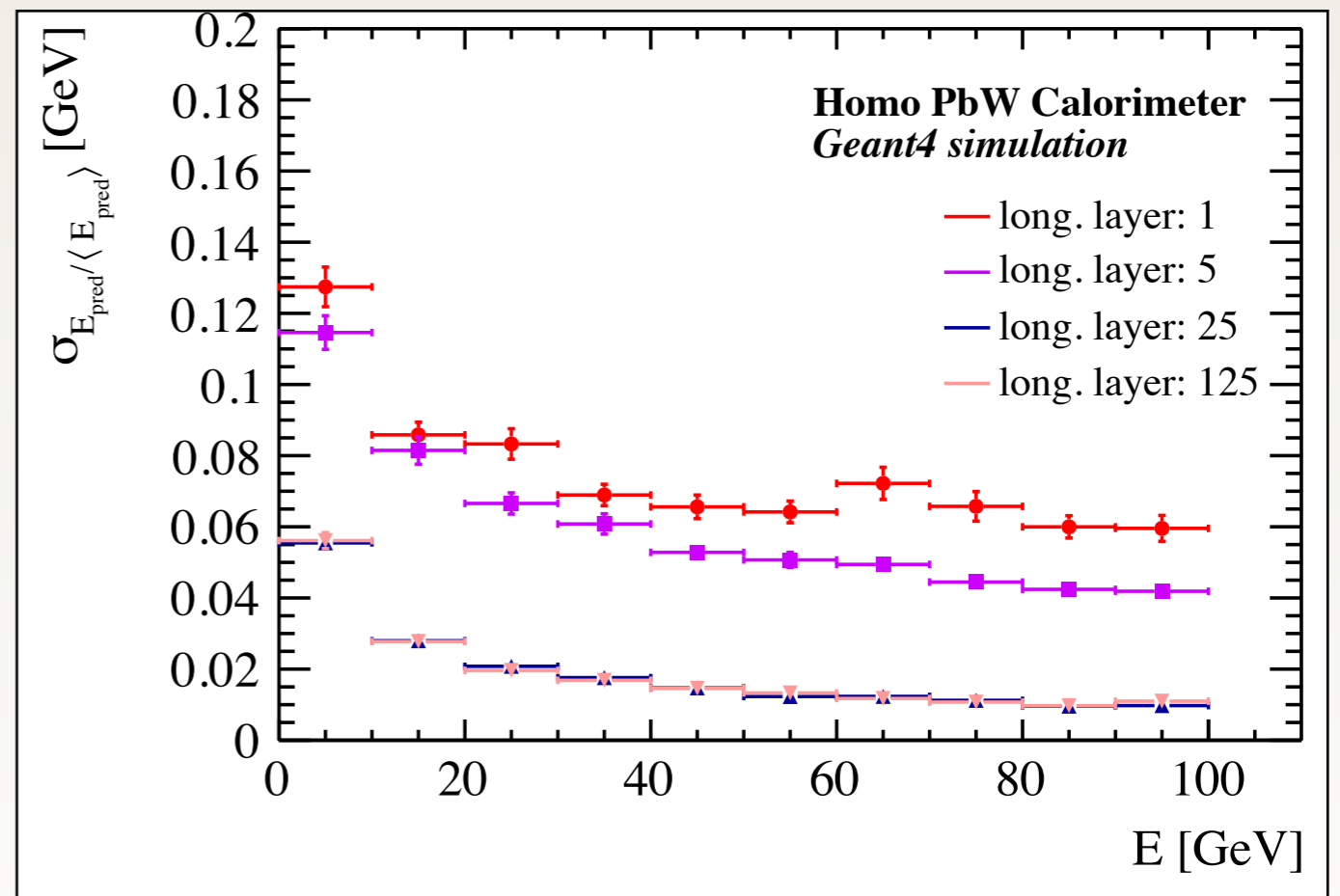
- How does calorimeter segmentation impact software compensation
- Turn it around: use DNN as a tool for (almost) optimal reconstruction

- Consider lead tungsten calorimeter

- ▶ Factorise out sampling and electronics effects
- ▶ 1m x 1m x 2.5m
- ▶  $10 \lambda$ ,  $200 X_0$

- Compare different longitudinal segmentations

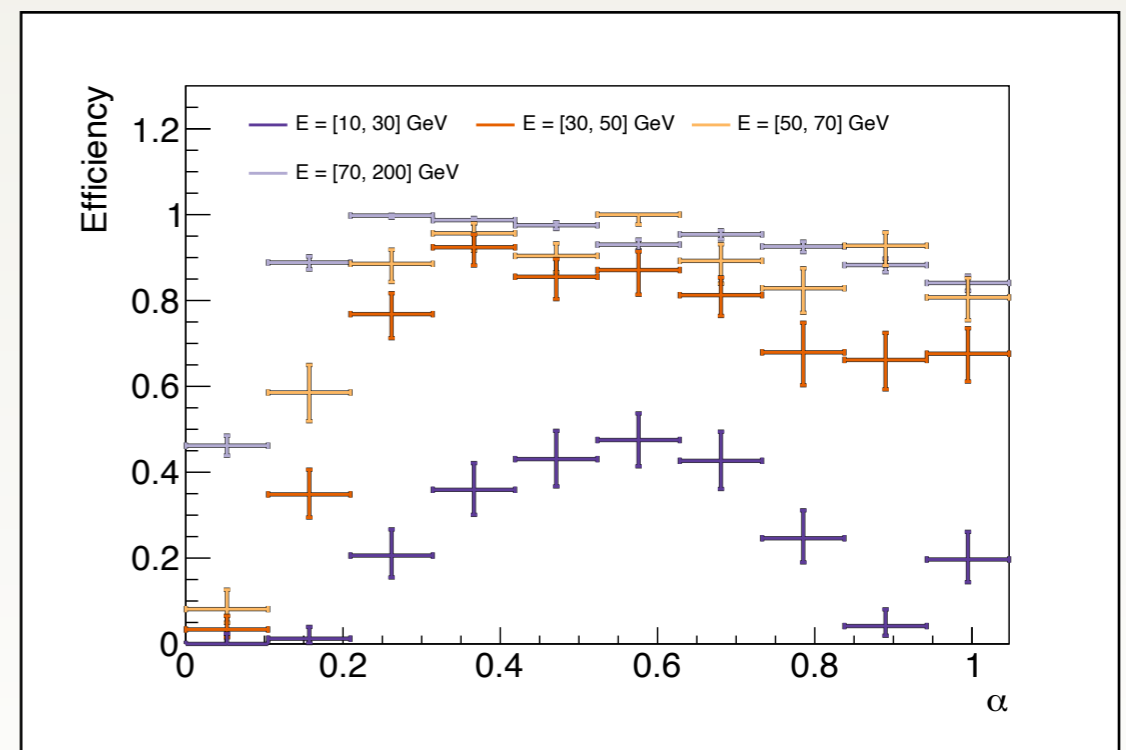
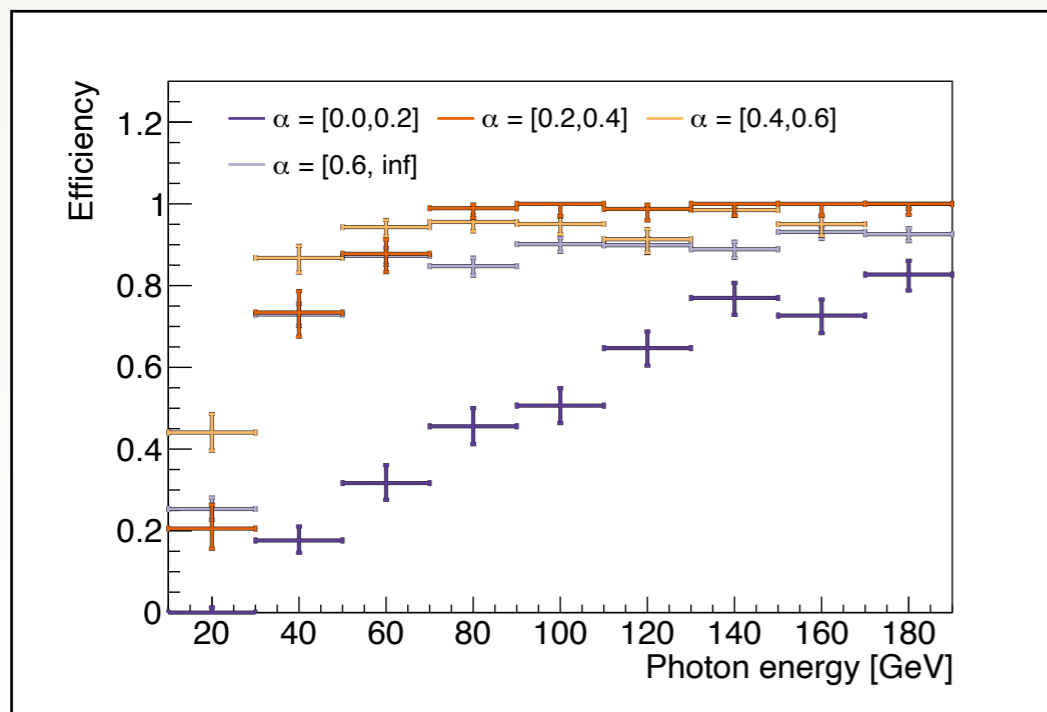
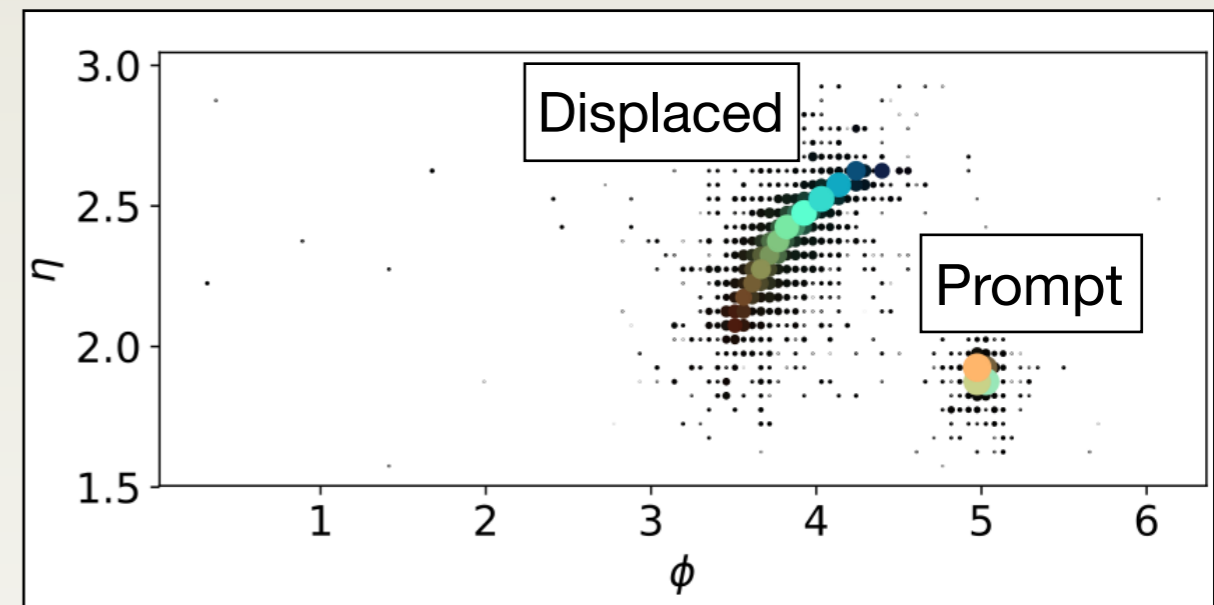
- ▶  $10 \lambda$
- ▶  $2 \lambda \sim 40 X_0$
- ▶  $0.4 \lambda \sim 8 X_0$
- ▶  $0.08 \lambda \sim 1.6 X_0$



- Resolution saturates between 2 and  $0.4 \lambda$  for full energy range

# Fast Pattern Recognition

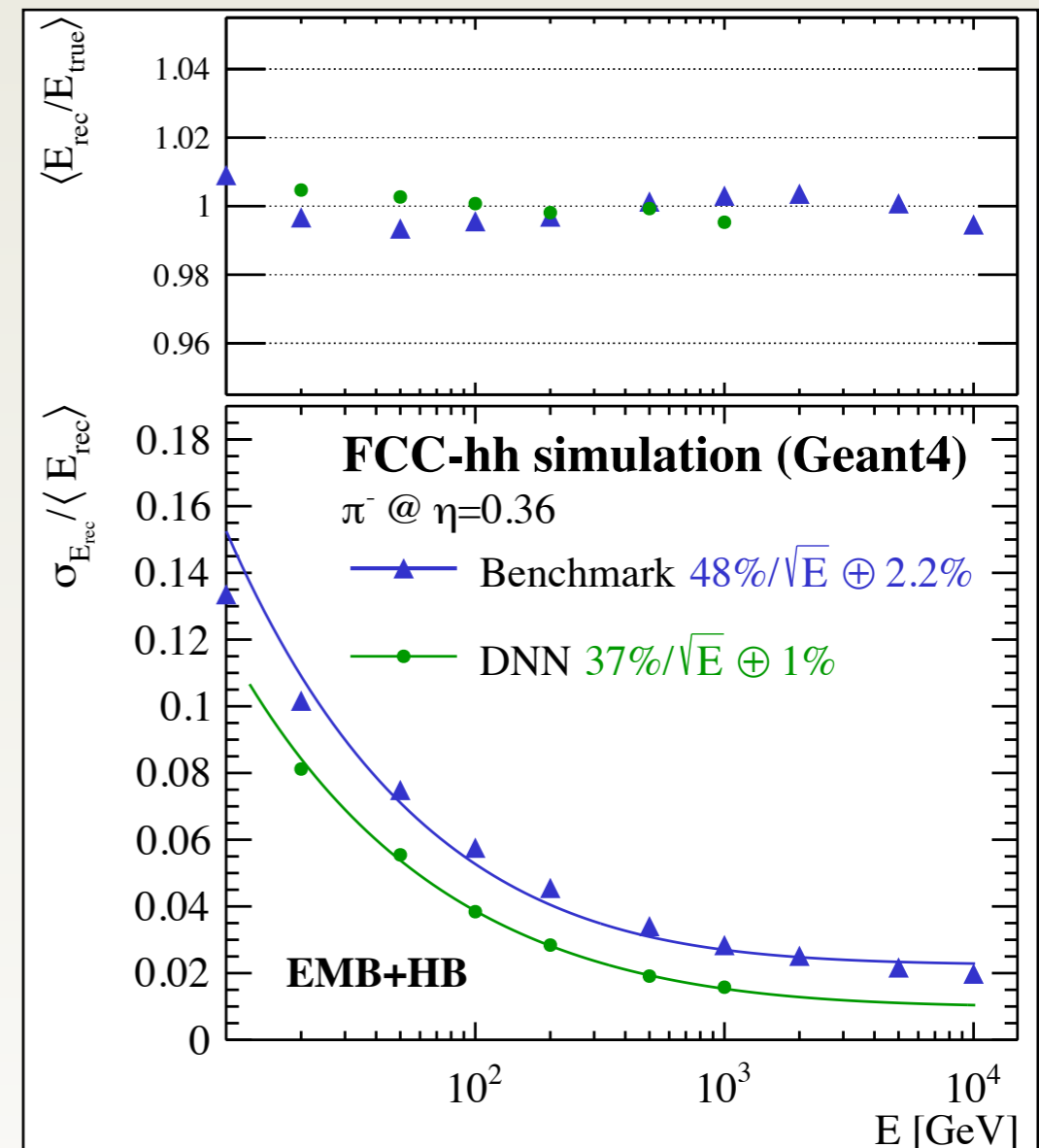
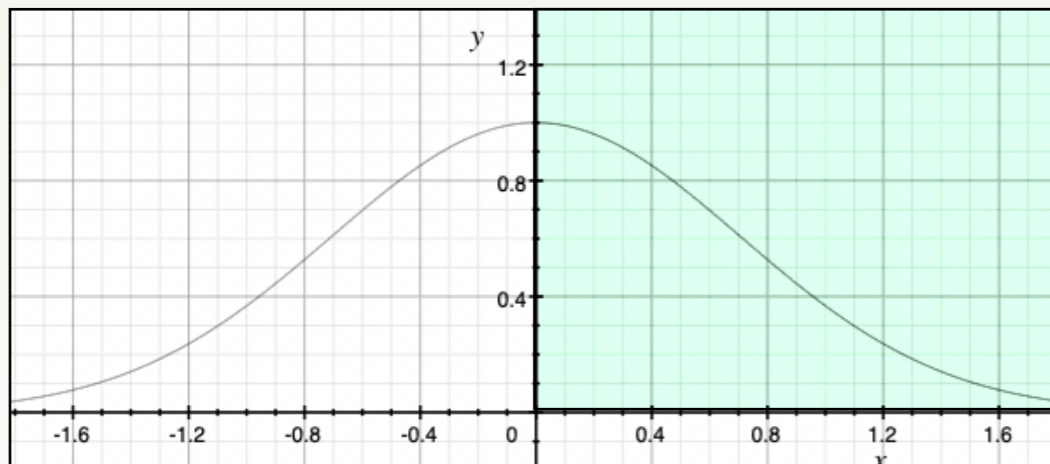
- Identify non pointing showers from BSM long-lived particle decays
- Study performed in the forward region (similar to CMS HGCal geometry)
- Aim for first-level trigger
  - ▶ Needs to be very fast
  - ▶ Rather “simple” low-parameter CNN
- Proof of concept is promising
  - ▶ Working on optimising it for FPGA implementation (with HLS4ML [2] team)



J. Alimena, Y. Iiyama, JK; arxiv:2004.10744  
 [2] arxiv:1804.06913

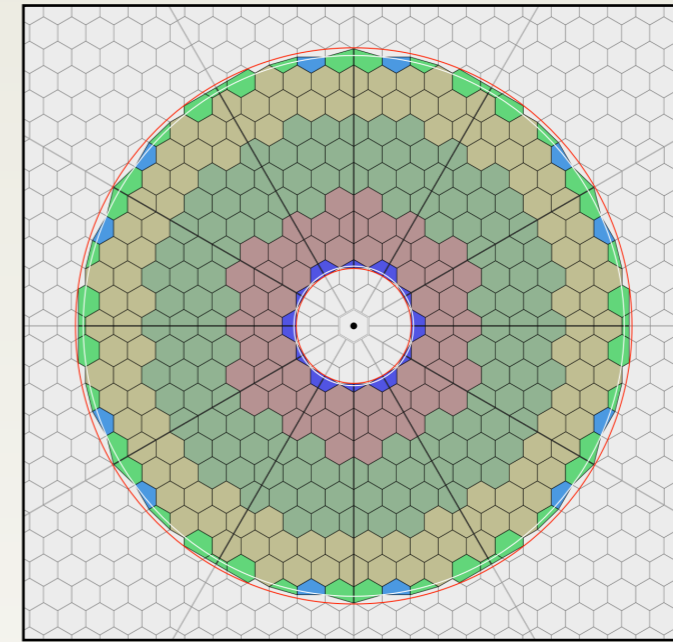
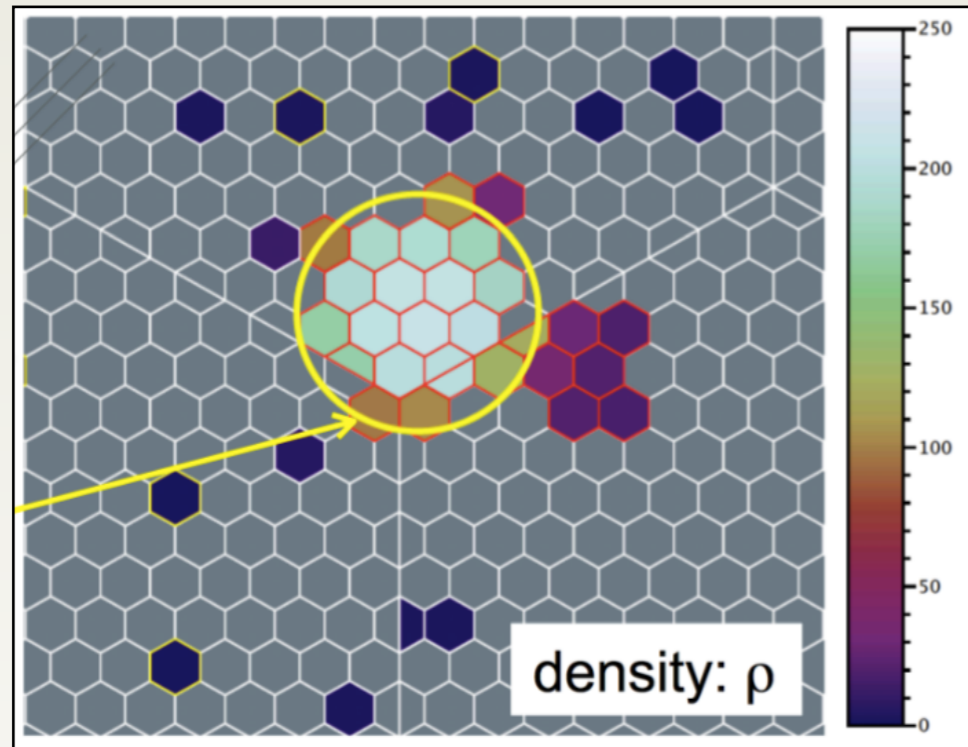
# Results and linearity

- High gain from ML based approach
  - ▶ Customised CNN layers with energy pass-through including 'compensating' (small) corrections to it (ResNet inspired)
- Sampling term of only 37%
- Linearity at edges not optimal  $\rightarrow$  *very common*
  - ▶ Network learns quickly:  $E > 0$
  - ▶ Expectation value and mean differ

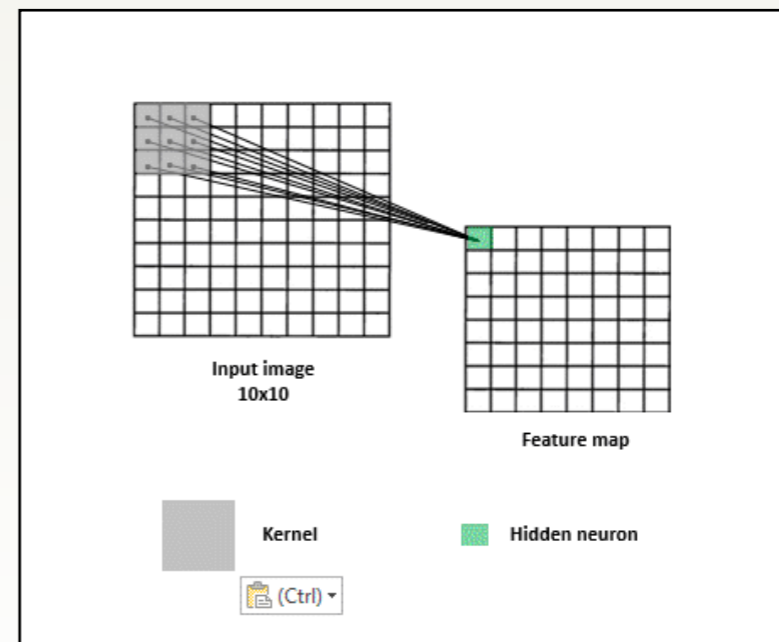


C. Neubüser, et al, arXiv:1912.09962  
 More details will be in C. Neubüser, JK, paper in prep.

# Going beyond regular geometries



- Detectors are not regular grids
- E.g. CMS HGCal
  - ▶ Hexagonal sensors
  - ▶ Size changes with depth and  $\eta$



# Representation of showers

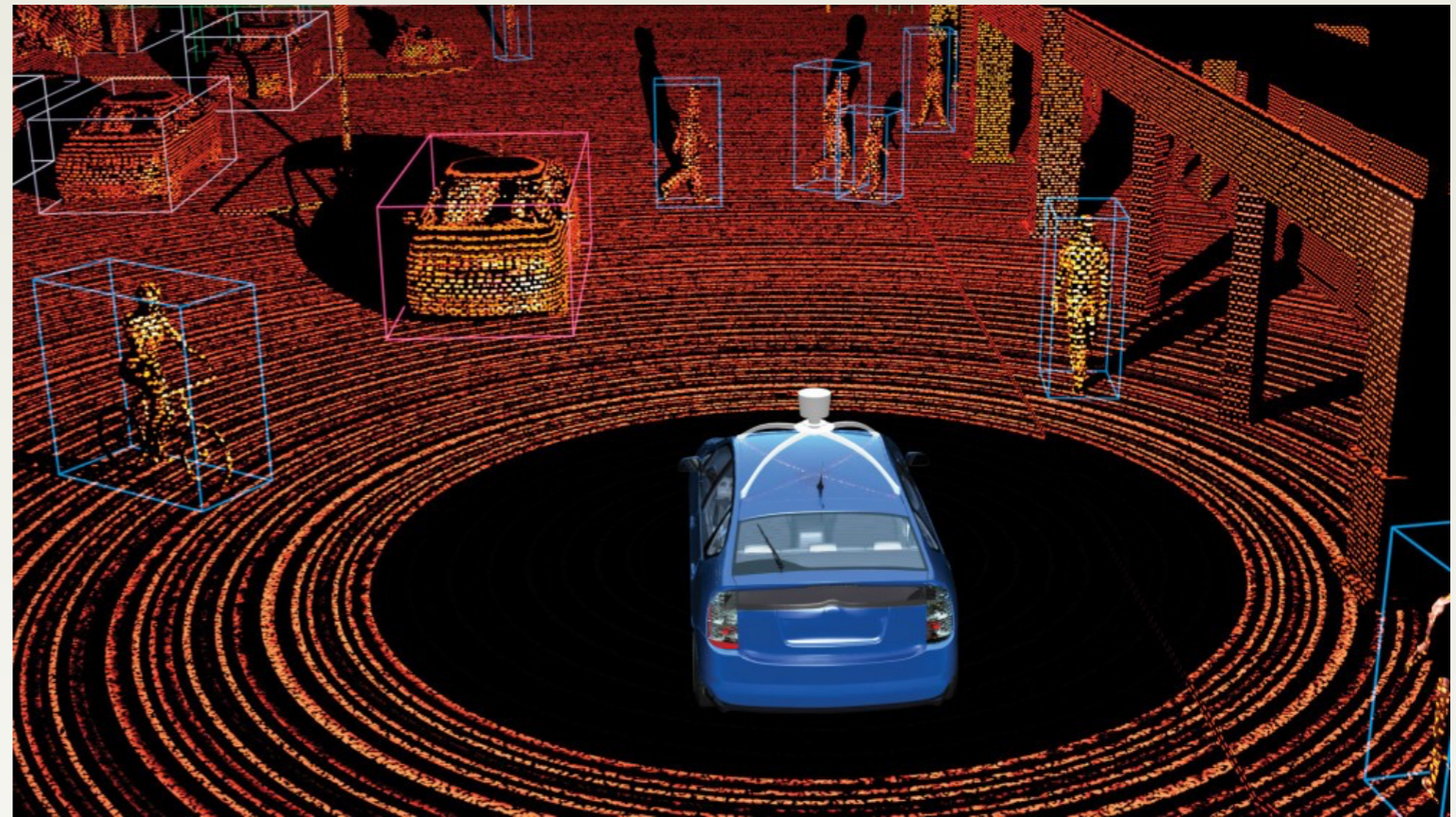
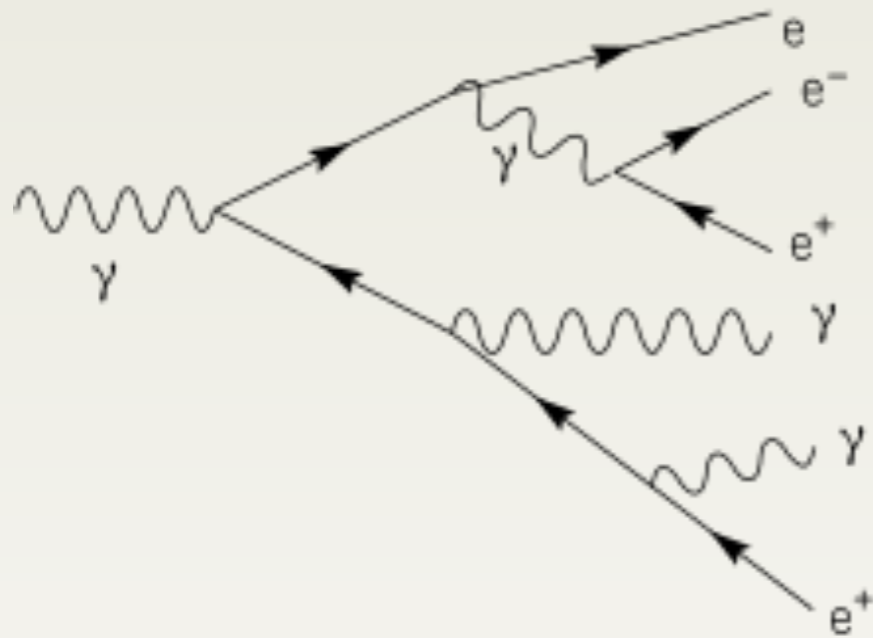
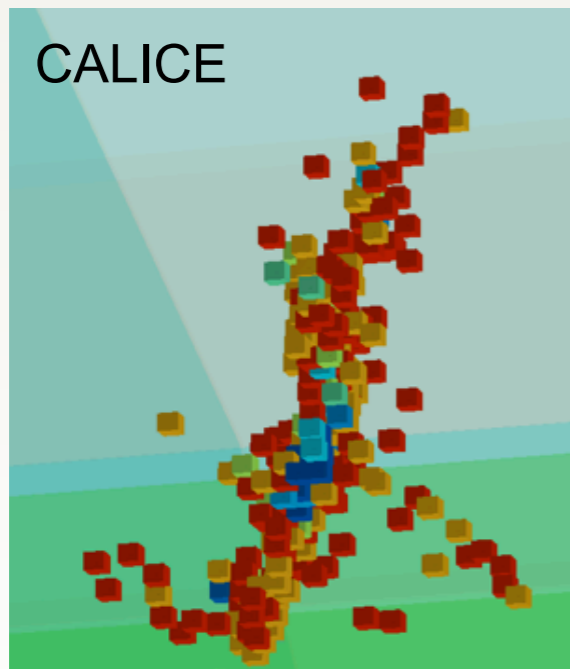


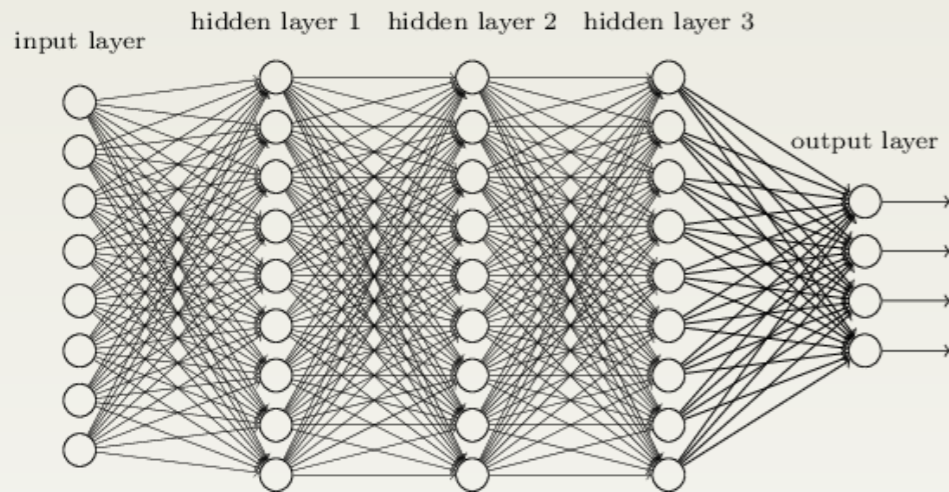
Image from <https://news.voyage.auto/an-introduction-to-lidar-the-key-self-driving-car-sensor-a7e405590cff>



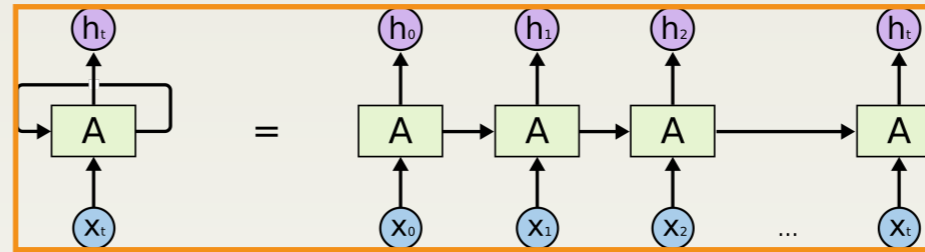
- Dense energy deposits
- Deposits connected by tracks
- ➔ Showers have physical graph like structure
- ➔ Hits can be represented by point clouds

# Irregular Structures

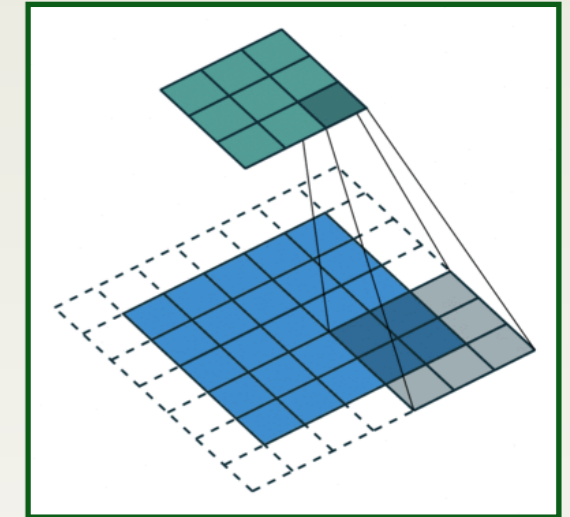
- Off-the shelf architectures...



Low input dimensionality



Clear sense of sorting / sequences



Regular grid

- ..do not represent particles or most sensor arrays in a detector

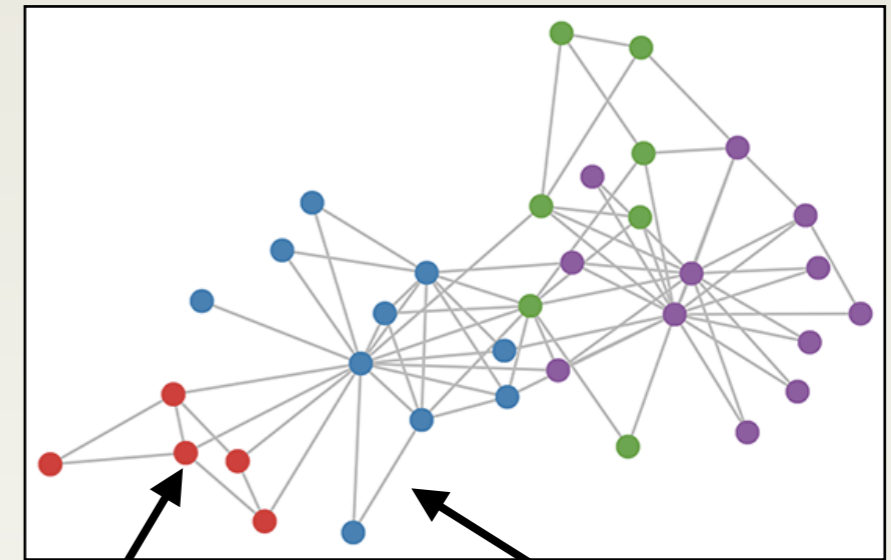
- Graph networks

- ▶ No sorting required
- ▶ No grid
- ▶ Sense of connection
- ▶ Basic principle: information exchange through edges (connections)
- ▶ Very active area of research in CS



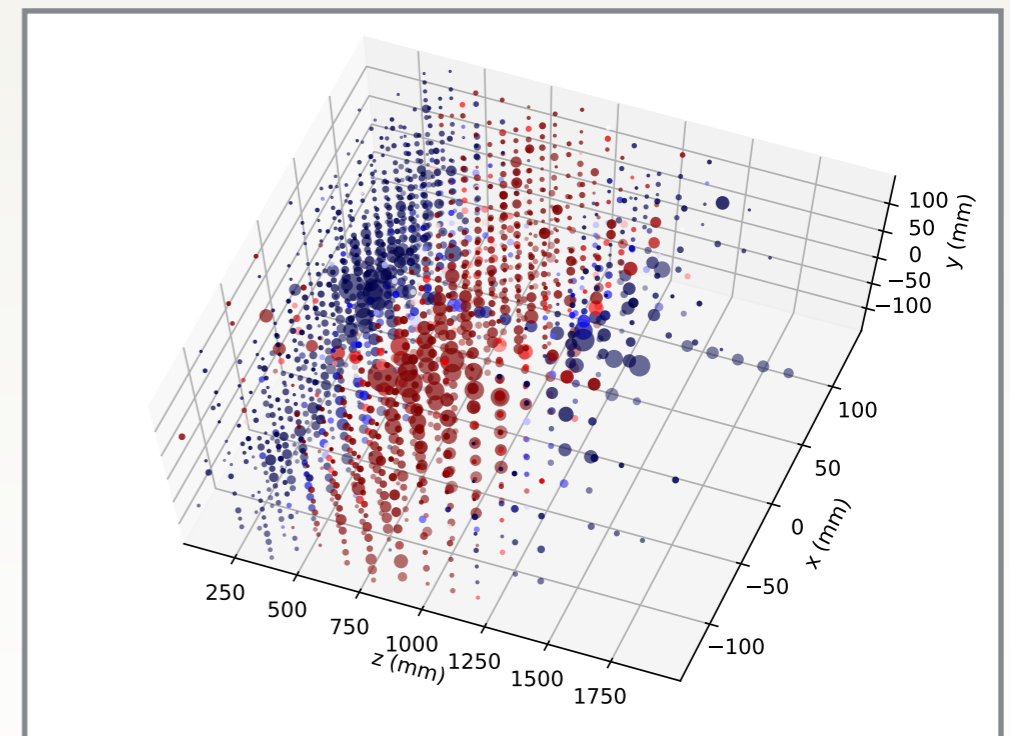
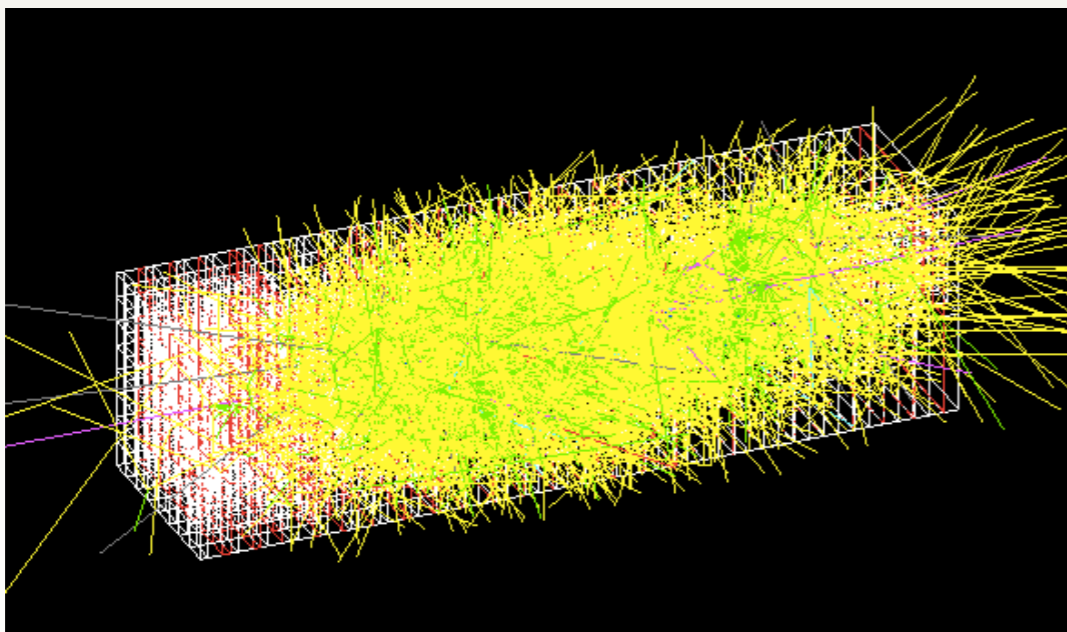
# Going beyond CNNs

- Using graph neural networks for reconstruction
  - Represent showers as point clouds
  - In particularly interesting: dynamic graph networks learning space transformations (no human engineered edges)
- Here in a simplified irregular calorimeter
  - PbW, 35 cm x 35 cm x 2.2 m
- Predict fractions per hit for 2 overlapping charged pion showers
  - Energy: 10-100 GeV



Sensor → vertex

Connection → edge

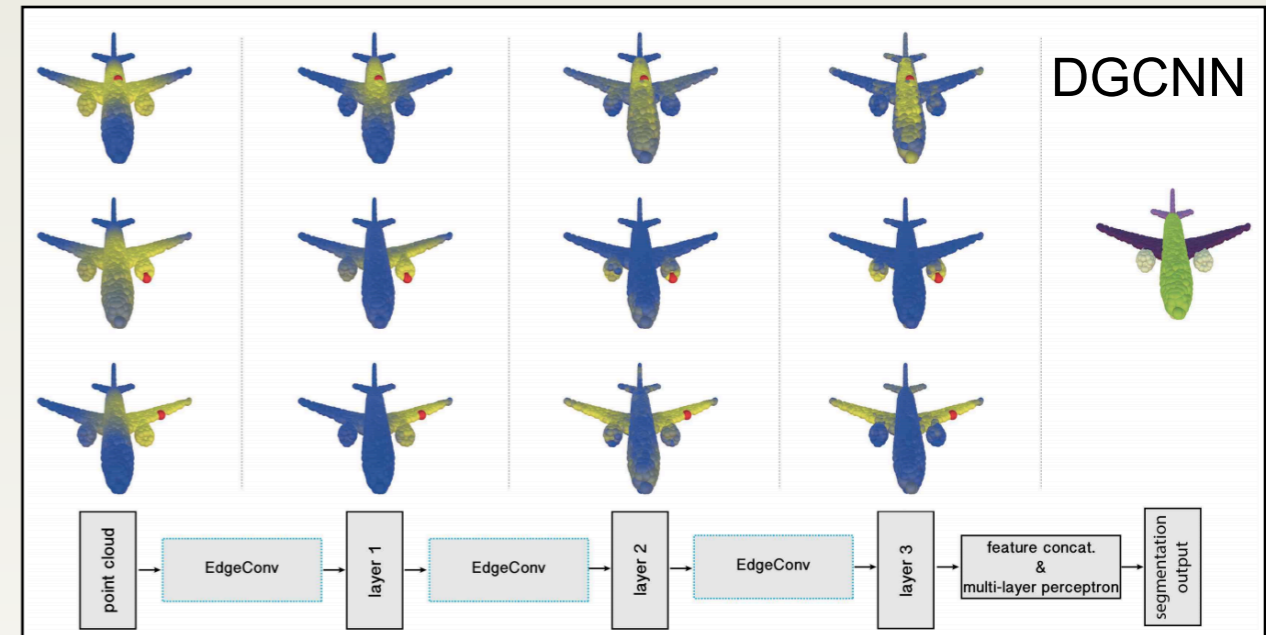


# Applicable dynamic graph networks

- Proposal for 3D segmentation of point clouds: EdgeConv/DGCNN [1] similar to our problem

- ▶ Transform features per vertex (sensor) (64)
- ▶ Calculate distances in new feature space
- ▶ Collect K neighbours
- ▶ *Transform edge features* (distance vectors between sensors)
- ▶ Collect maxima to determine new vertex properties

- Proven very powerful for segmentation
- Also successfully used for jet identification [2]



- Fractional assignment is not 'just' segmentation

Multiple operations scale:  
 $V \times K \times F$

- Very resource demanding network architecture

- ▶ Realistically > 100k hits in a calorimeter, and **billions of events to process**

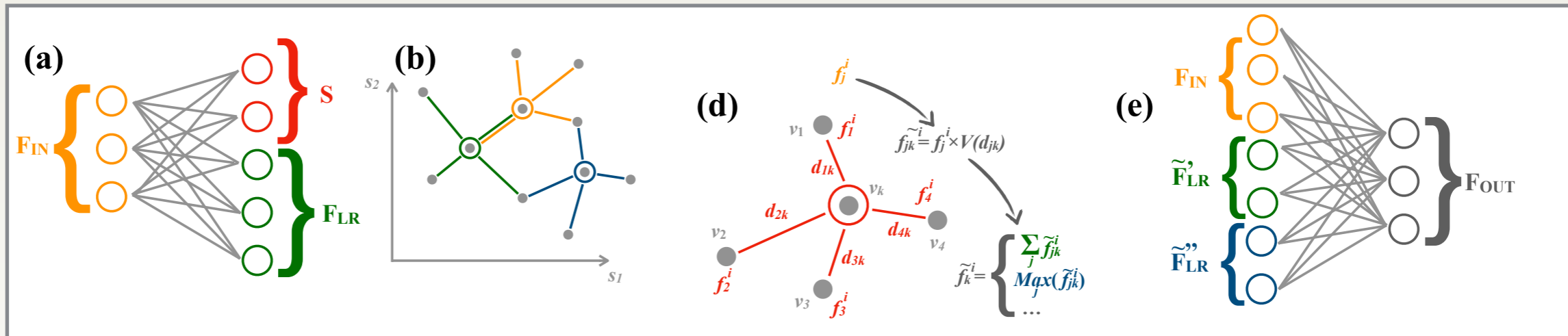
- Can we do better?

[1] Y. Wang, et al, arXiv:1801.07829  
 [2] H. Qu, L. Gouskos, arXiv:1902.08570

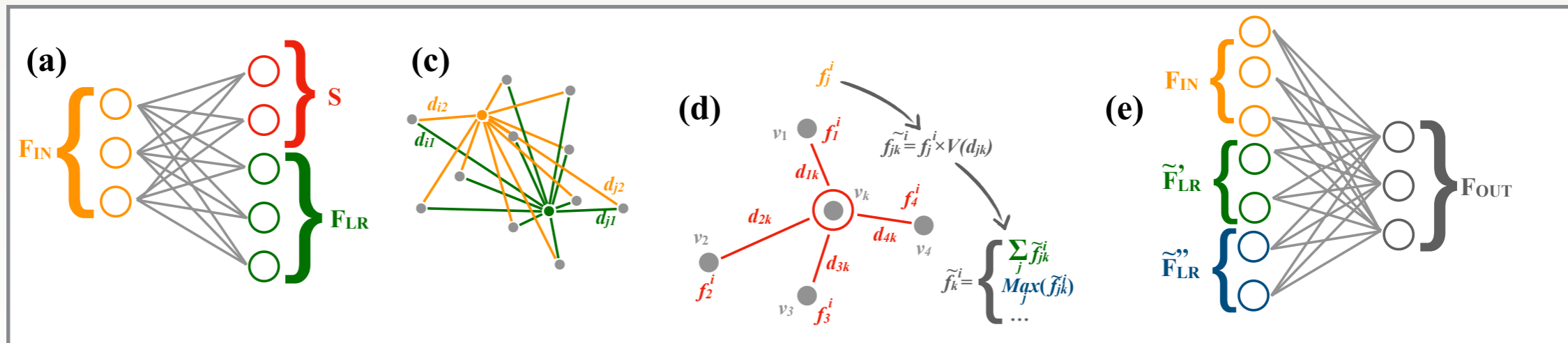
# GravNet/GarNet

- Most resource demanding operation in DGCNN
  - Determine neighbours in  $F_{IN}$  dimensions
  - Iteration over edges between  $K$  neighbours in  $F_{IN}$  dimensions
- GravNet/GarNet circumvent this problem
  - Split coordinate and feature space

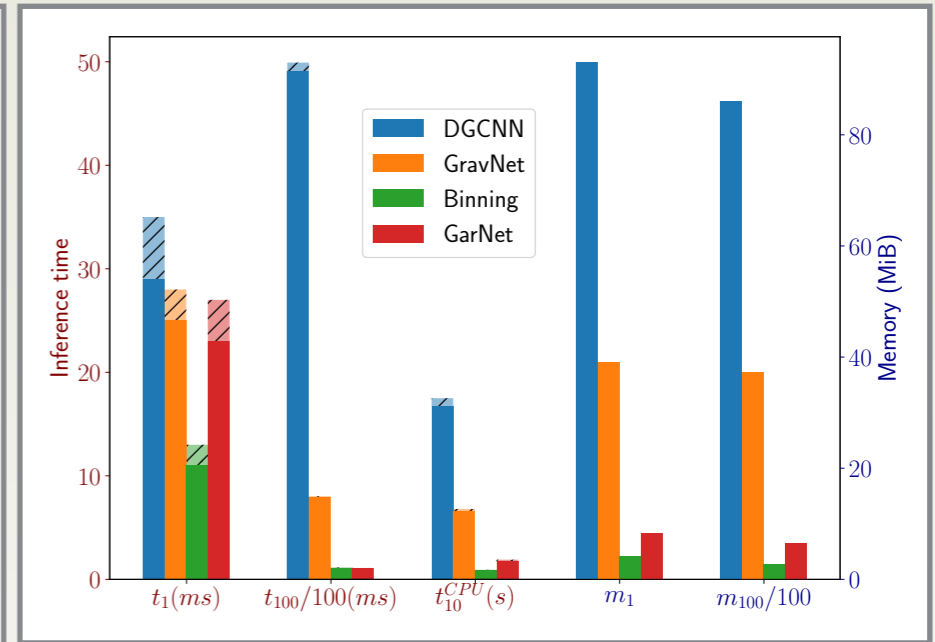
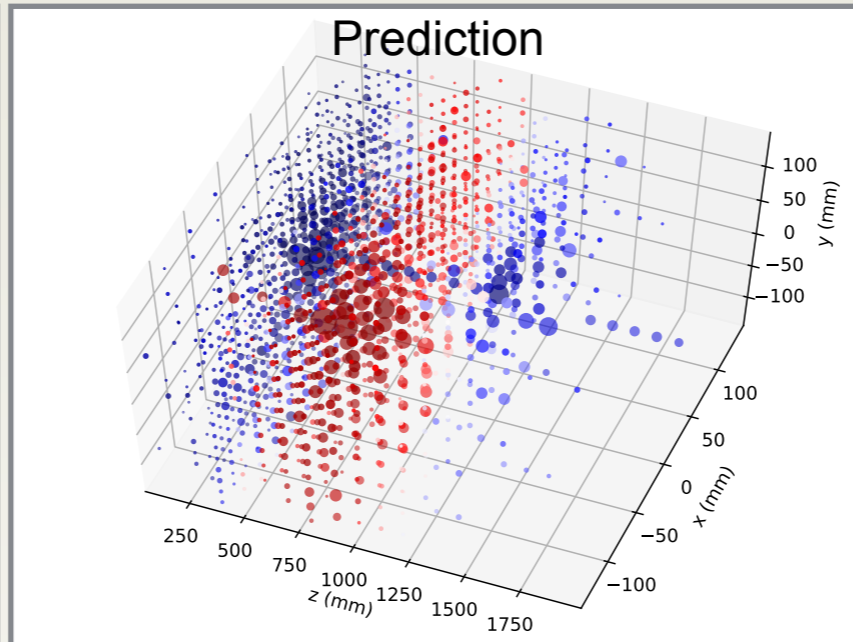
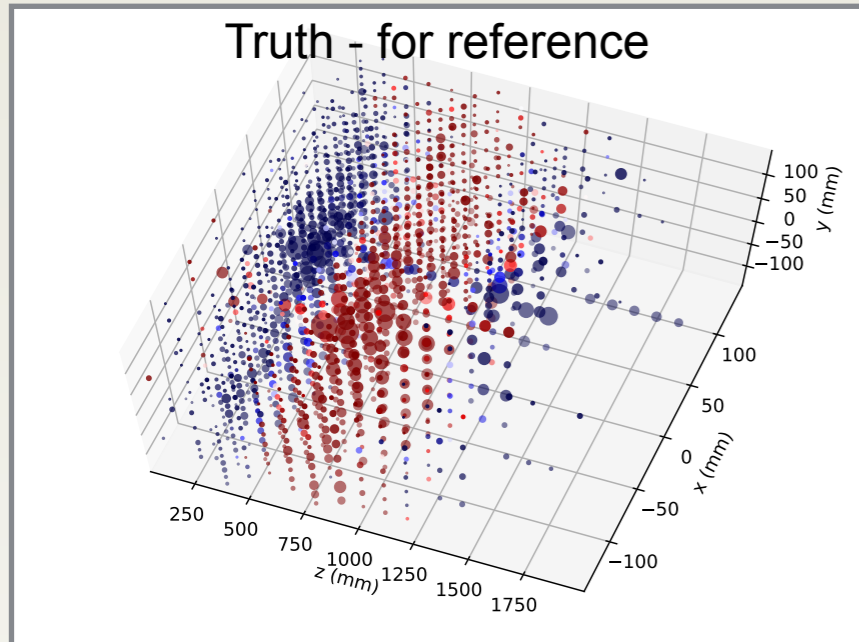
## • GravNet



## • GarNet



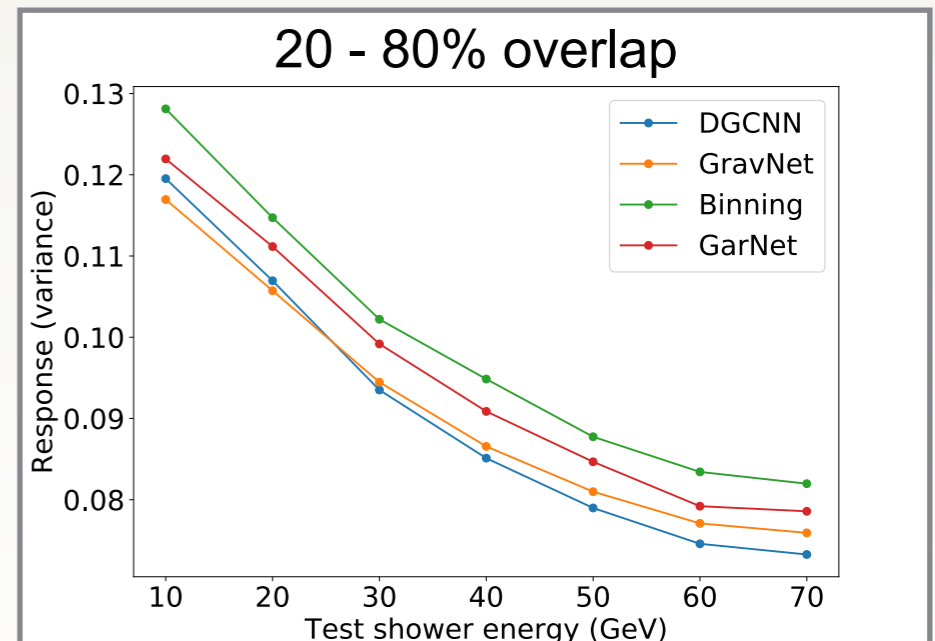
# Results



$$L = \sum_k \frac{\sum_i \sqrt{E_i t_{ki}} (p_{ki} - t_{ki})^2}{\sum_i \sqrt{E_i t_{ki}}}$$

$$R_k = \frac{\sum_i E_i p_{ik}}{\sum_i E_i t_{ik}}$$

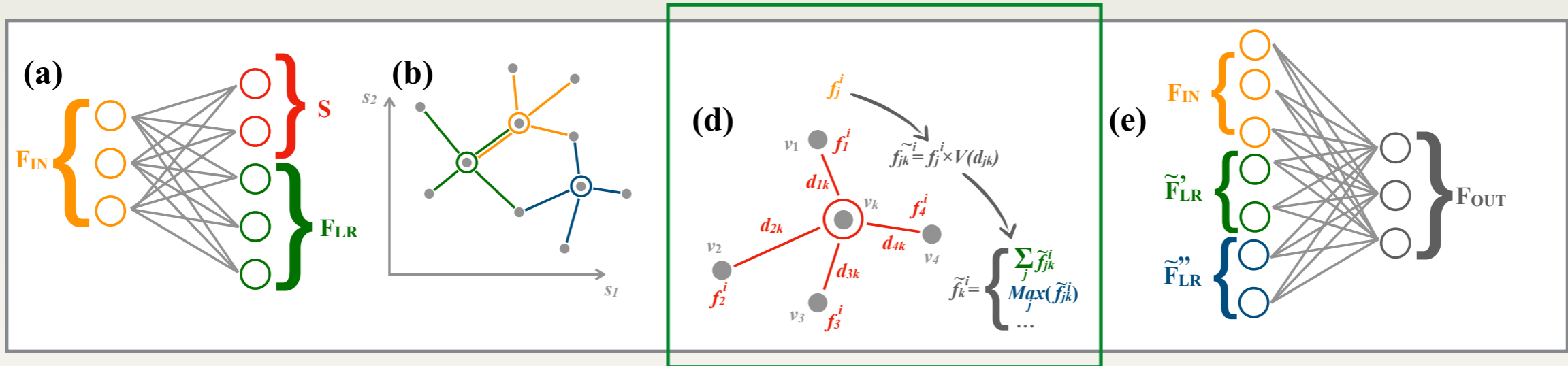
- Compare layers embedded in very similar architectures
- The graph network architectures outperform the CNN approach
- Similar performance but **lower resource requirements** of GravNet versus DGCNN
- Competitive performance and very low resource requirements for Garnet
- These architectures are applicable to (sparse) data with any structure, e.g. tracking, jets, ...



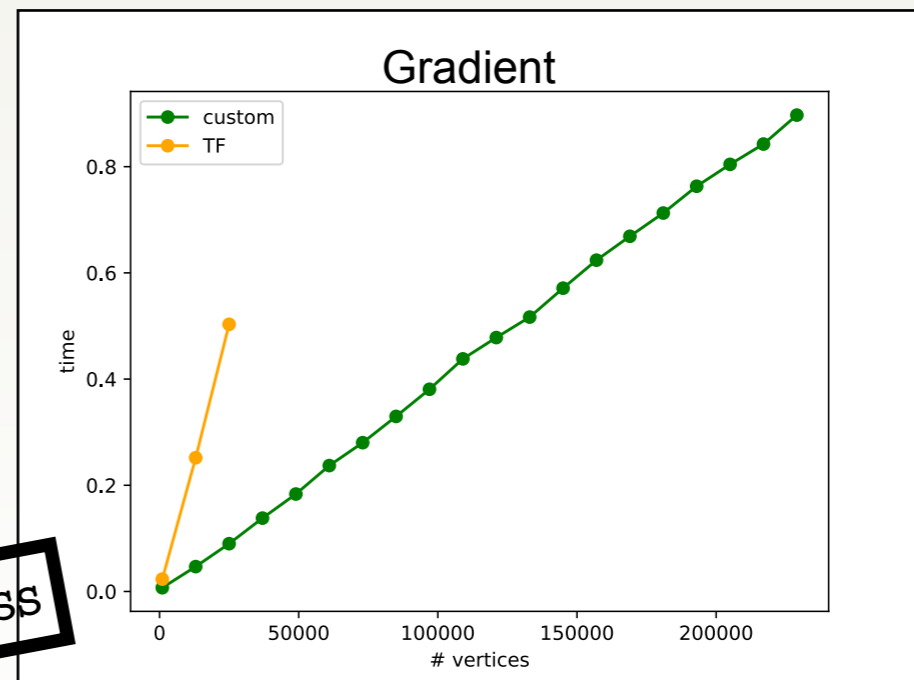
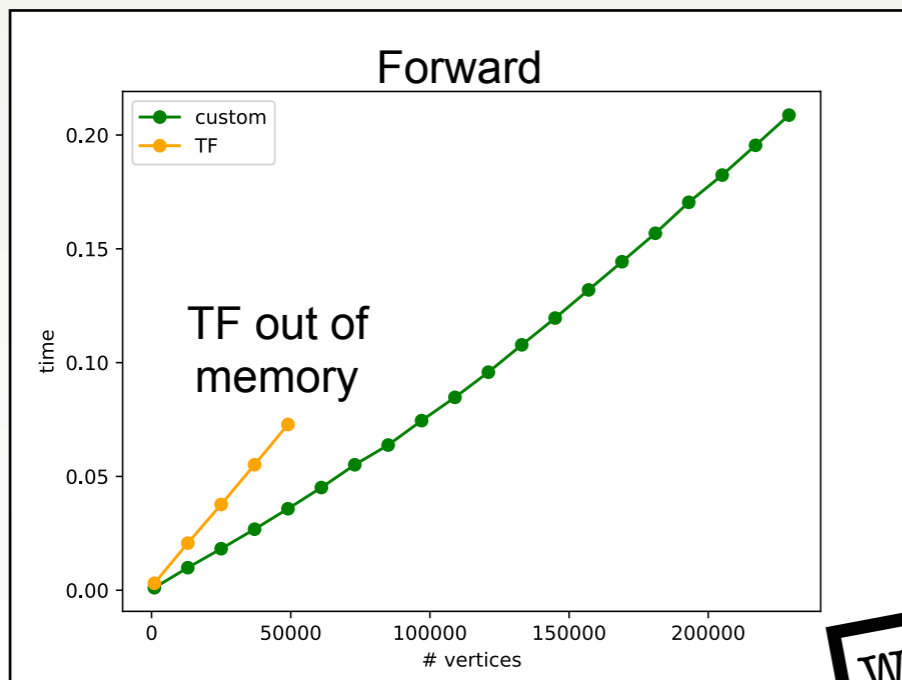
S.R. Qasim, J. K, Y. Iiyama, M Pierini arXiv:1902.07987, EPJC

# Customisations

- Operations in  $V \times K$  are expensive, also for memory.
- But: all information is already in memory



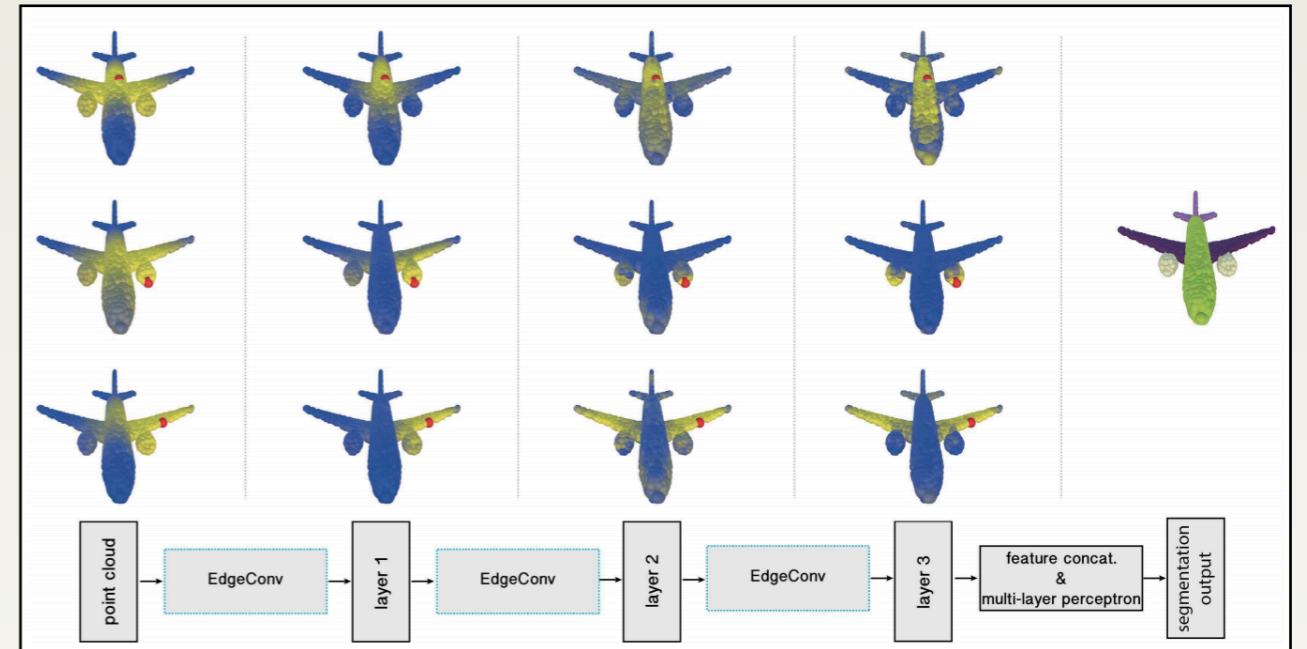
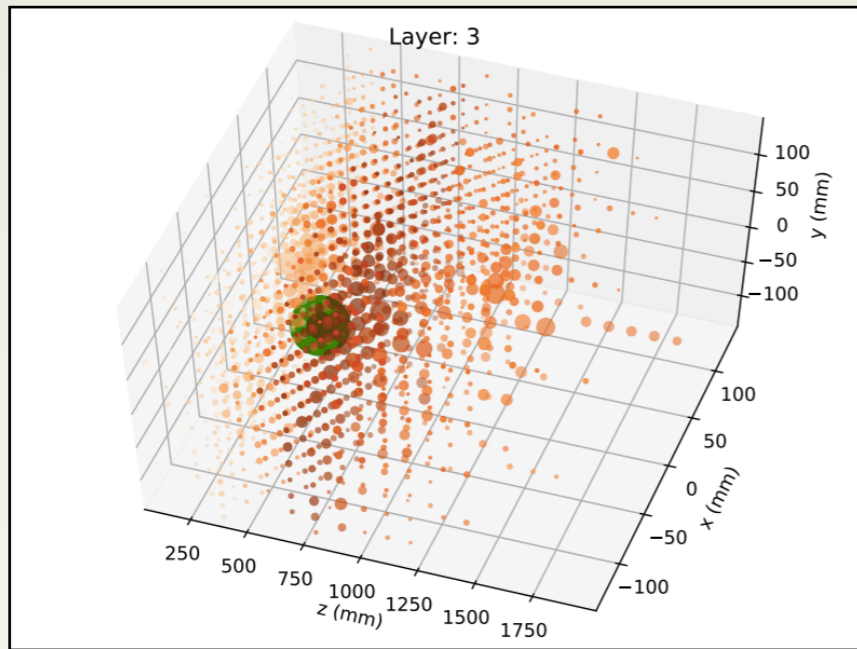
- Building custom CUDA kernels for fast inference/training
  - No memory scaling with  $K$  nearest neighbours



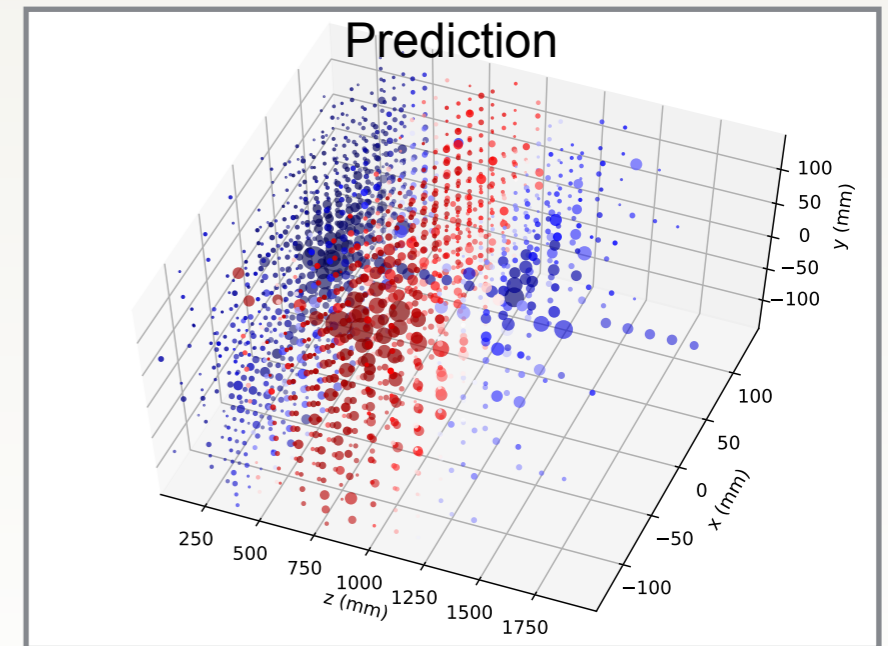
**Work in Progress**

# Interpretation

- Visualise distances in the latent coordinate space



- *Without direct supervision, the networks tend to cluster vertices belonging to the same shower*
- Seems to be a common feature of distance based dynamic graph networks

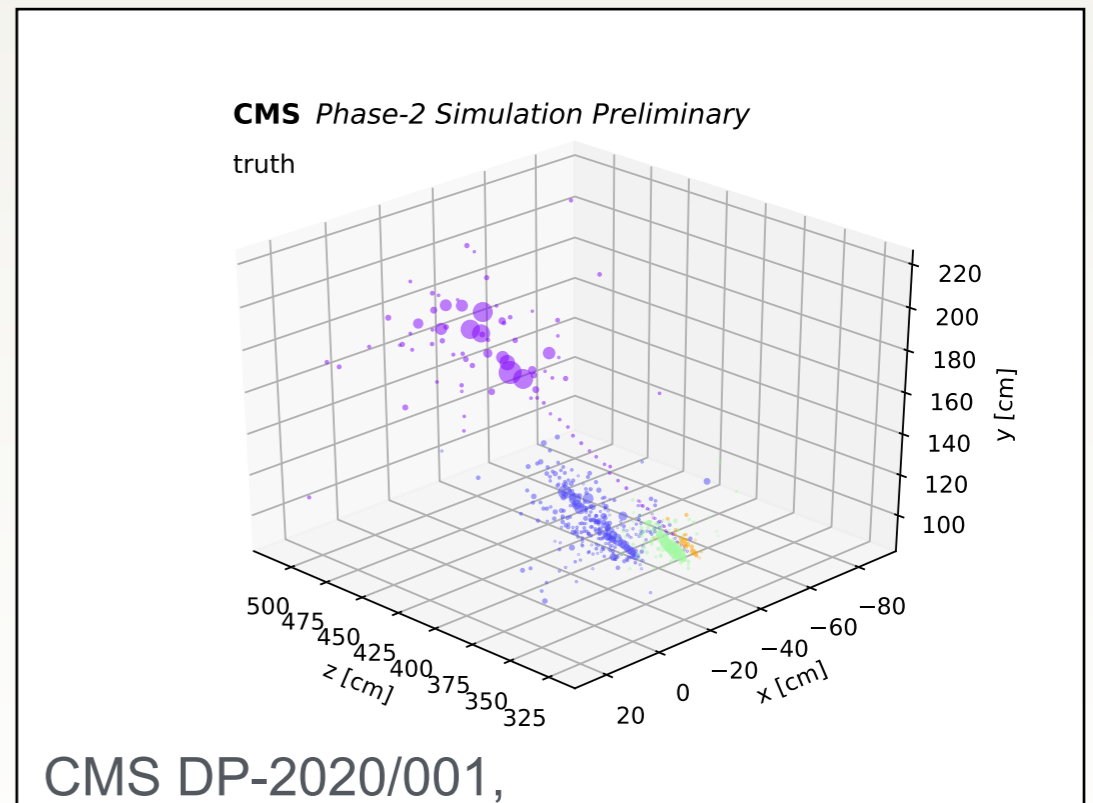
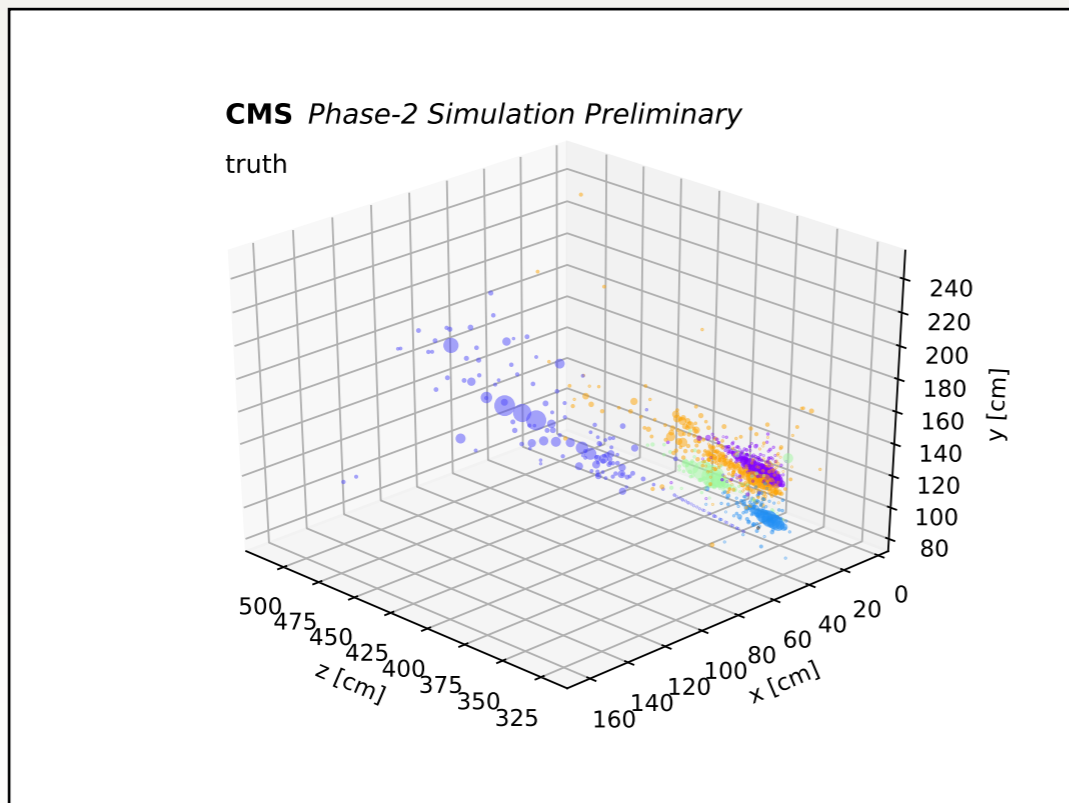
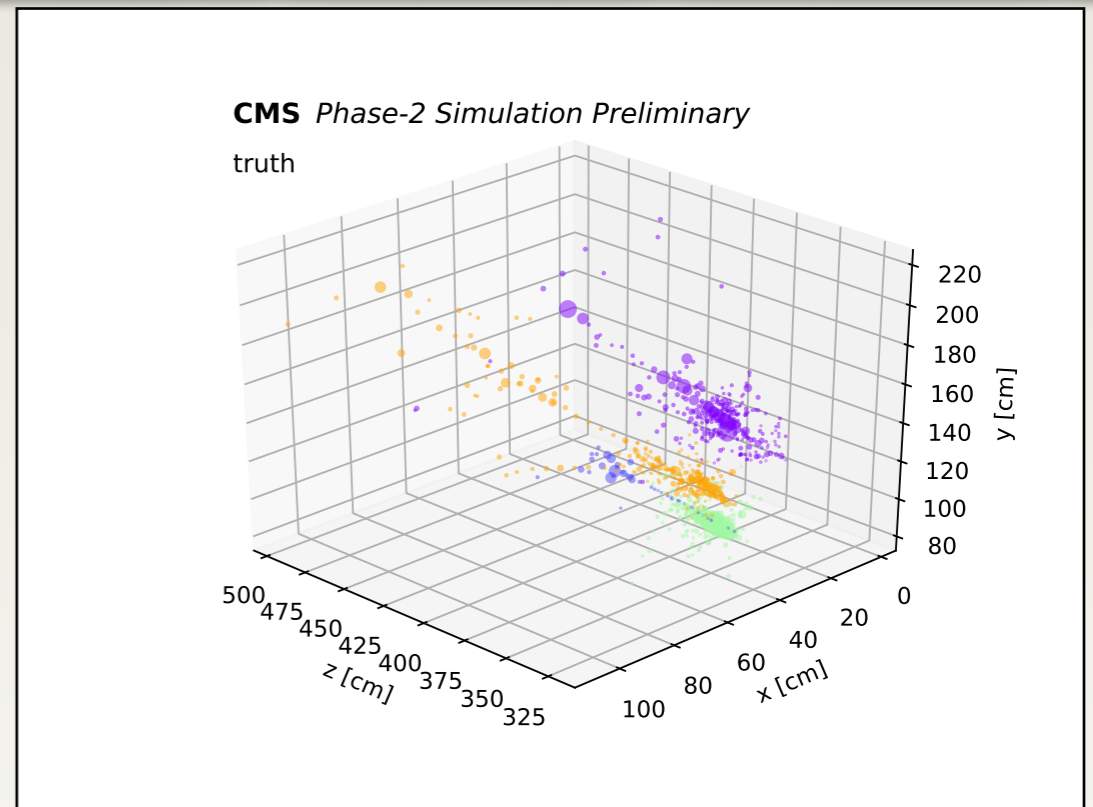


**GravNet**  
Now in torch\_geometric!

[https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch\\_geometric.nn.conv.GravNetConv](https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.GravNetConv)

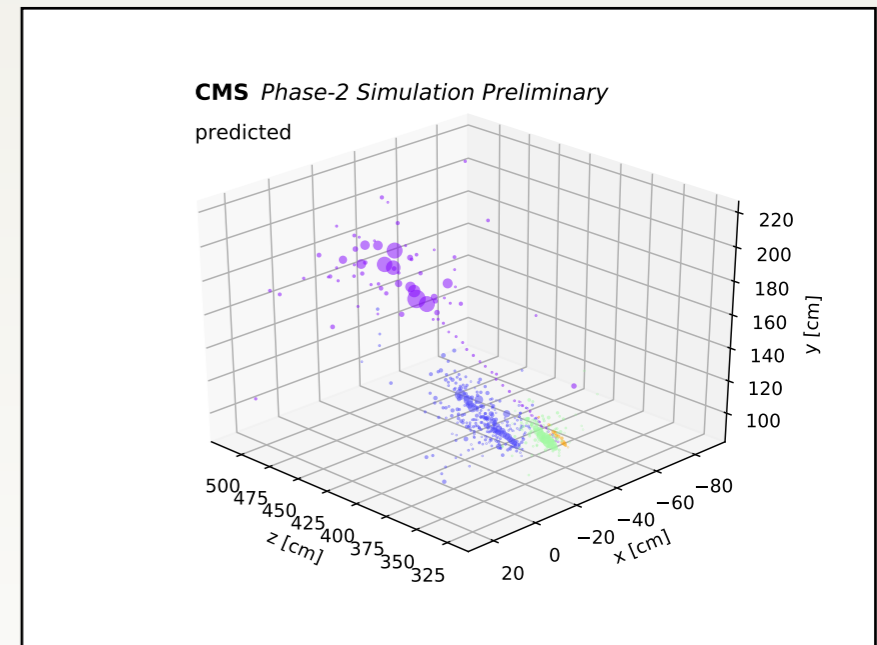
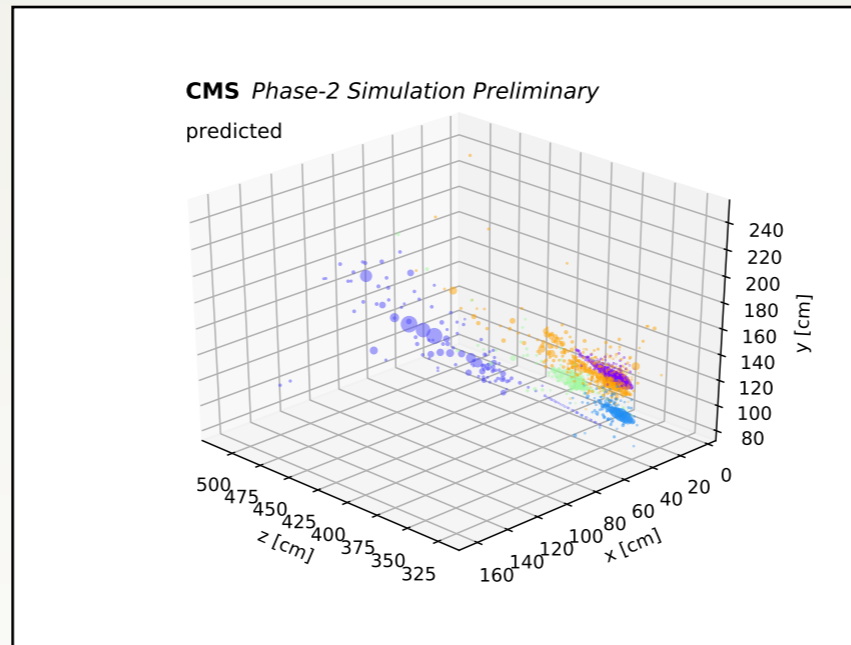
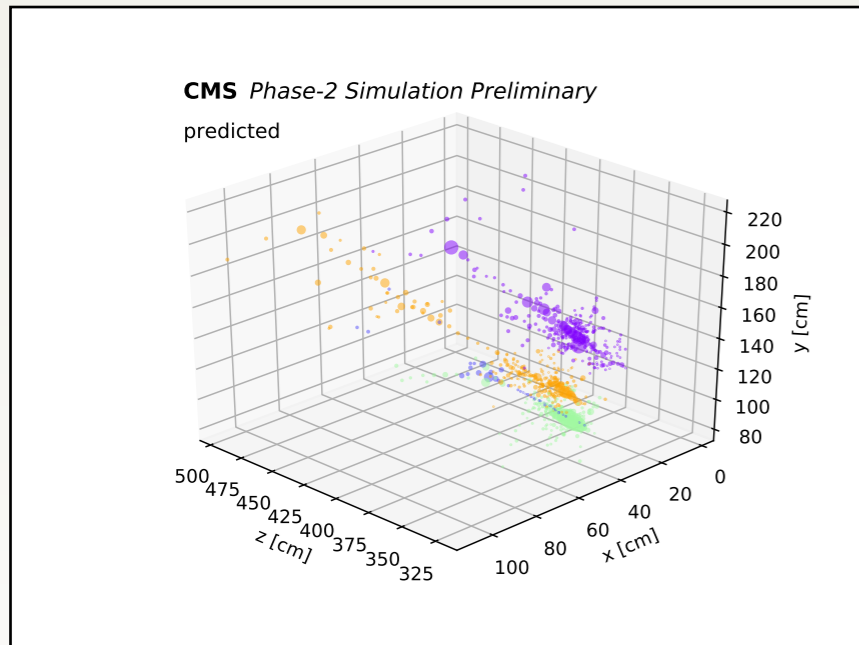
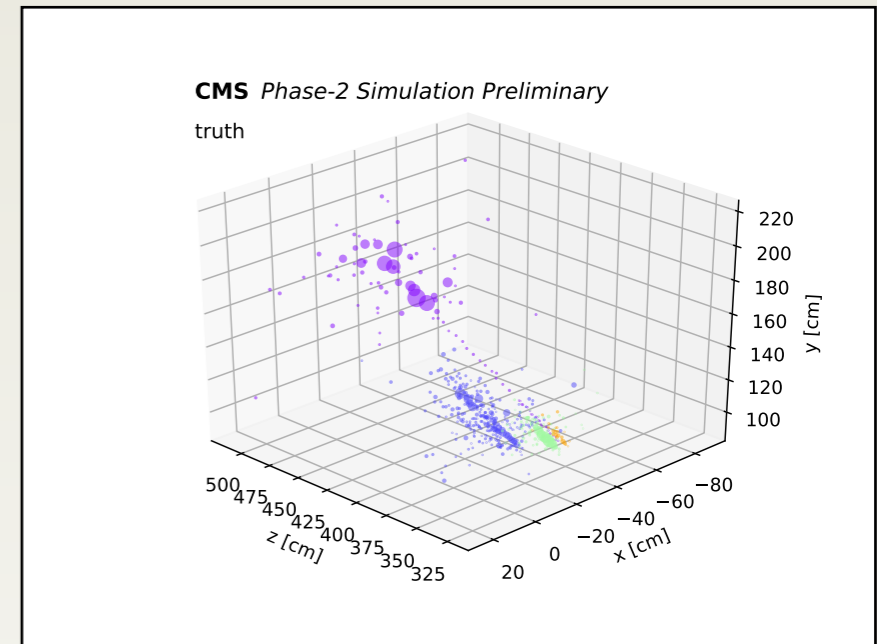
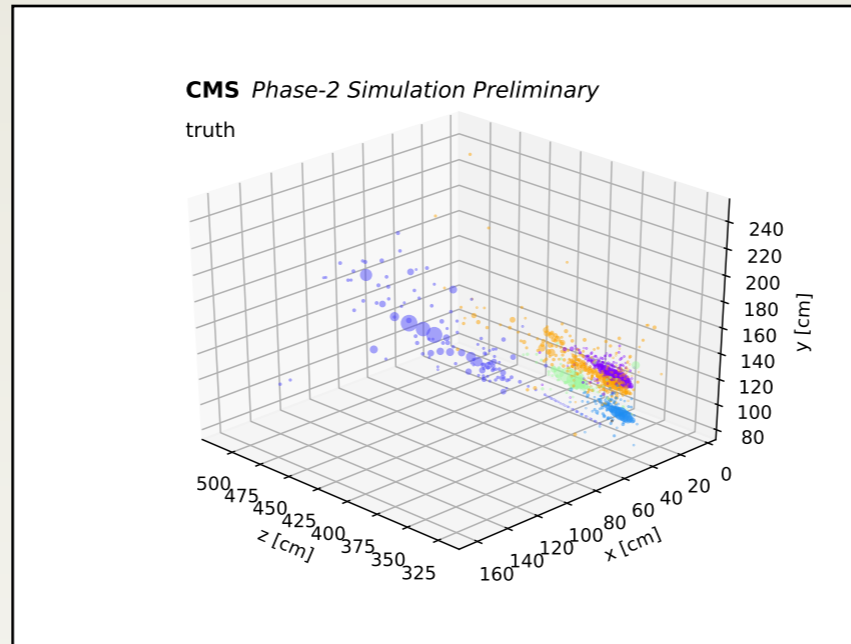
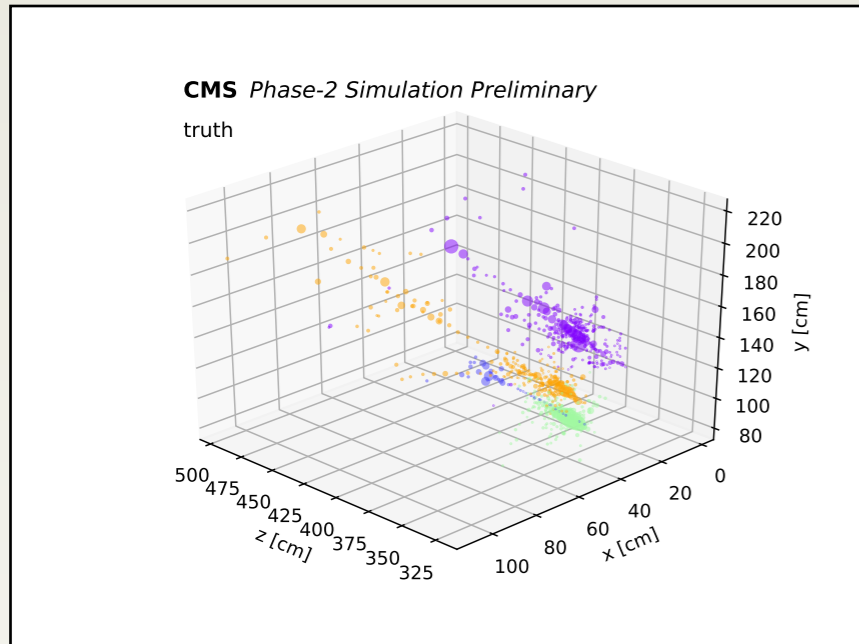
# Application to CMS HGCal

- Dataset
  - ▶ Full CMS HGCal simulation
  - ▶ 1-5 showers from electrons, photons, muons, charged pions within DR=0.5
  - ▶ 10-100 GeV
  - ▶ About 500k events
  - ▶ Hits pre-clustered on each layer (less inputs)
- Use GravNet with small adjustments
  - ▶ 5 output nodes, predicting shower fractions
  - ▶ 2 additional message passing layers in latent space



CMS DP-2020/001,  
NeurIPS 2019

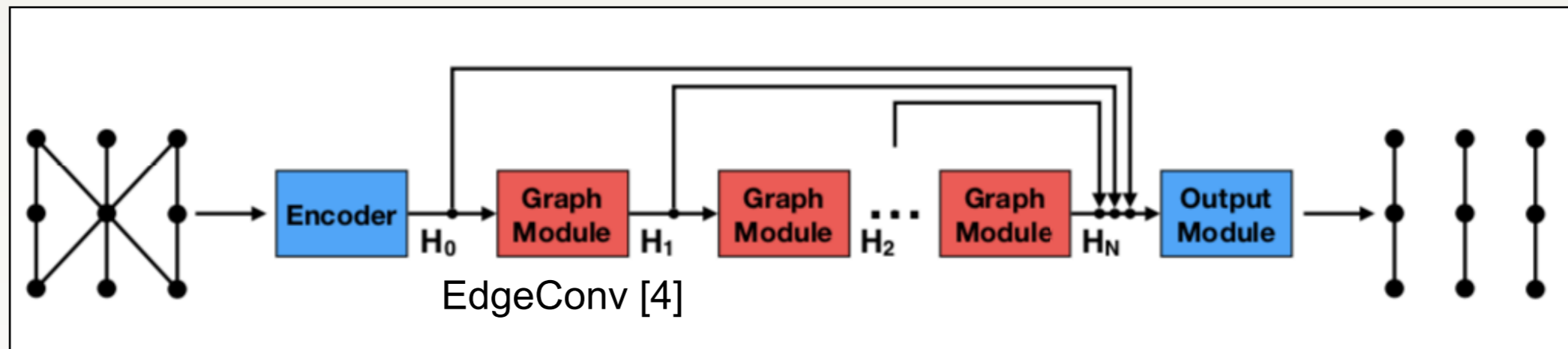
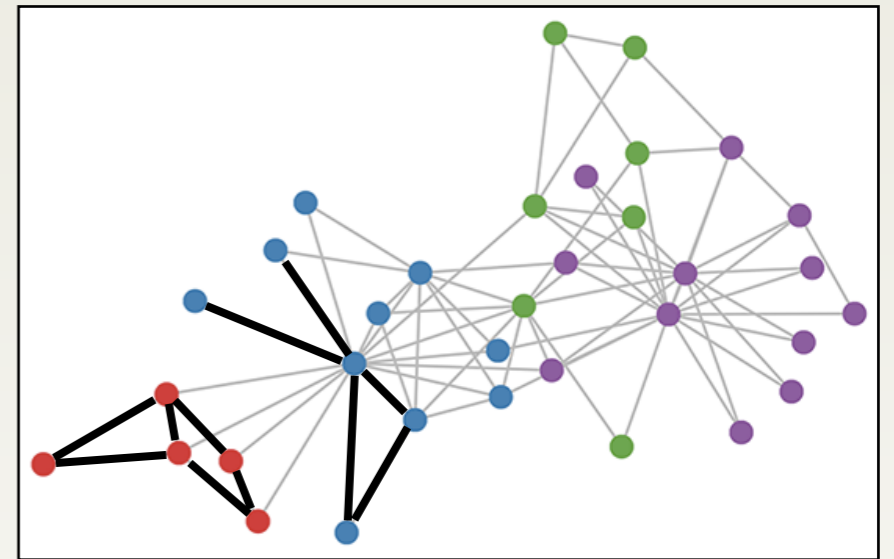
# Results



- *Excellent shower reconstruction*
- But what if there are more than 5 particles?

# One approach: Edge classifiers

- Inspired by HEP.TrkX [1,2], edge classifiers can overcome the problem
- Objects appear as vertices that are connected to each other, but not connected to others
  - ▶ Separates showers
- Edges can carry additional information like particle ID
- Recipe [3]:
  - ▶ Pre-define a graph containing all possibly true edges (e.g. neighbours within a sphere)
  - ▶ Train the network and perform inference



- ▶ Select edges with a predicted probability of more than 0.5 to be true as connections

[1] S. Farrel et al, arxiv:1810.06111,

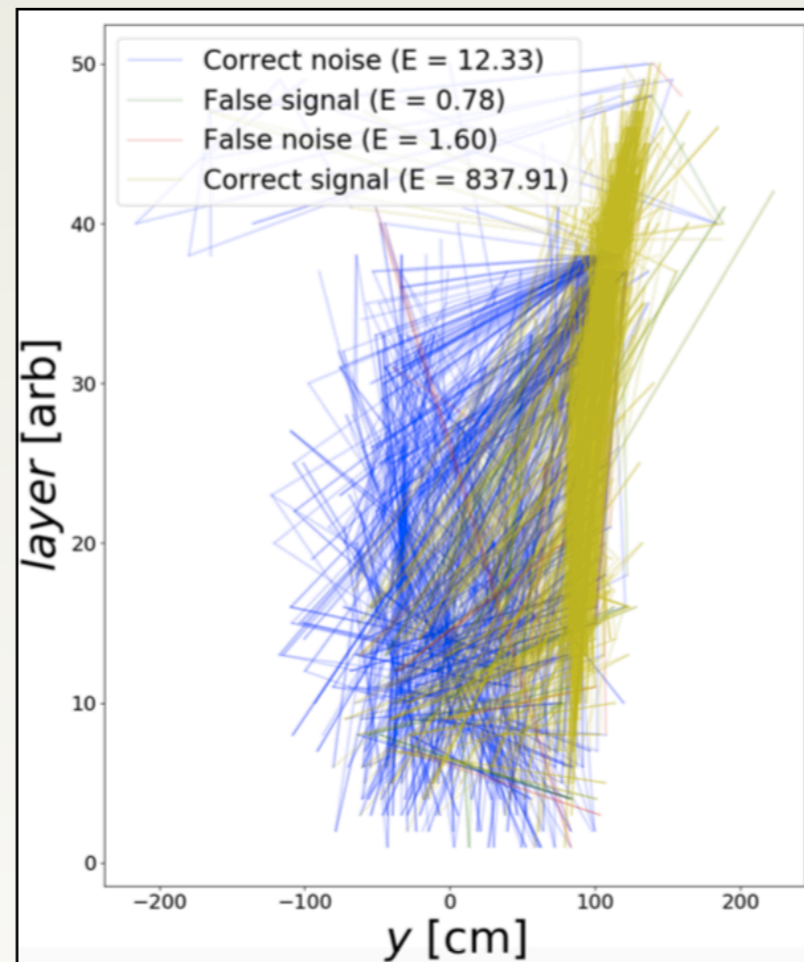
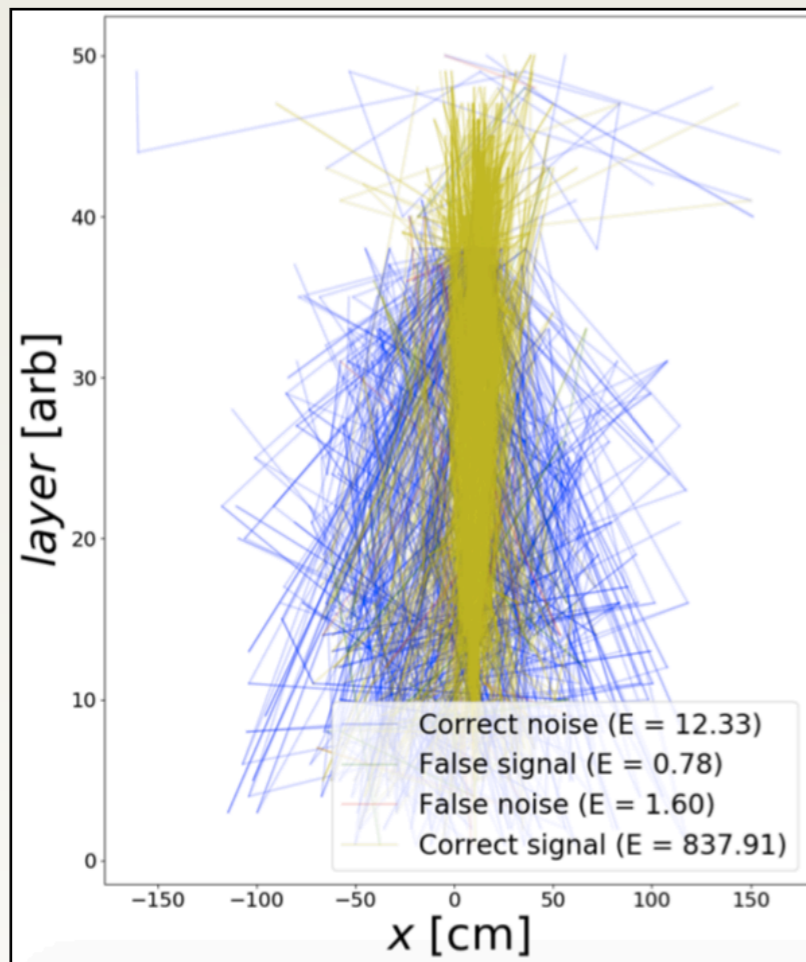
[2] 10.1051/epjconf/201715000003

[3] X. Ju et al, [https://ml4physicalsciences.github.io/files/NeurIPS\\_ML4PS\\_2019\\_83.pdf](https://ml4physicalsciences.github.io/files/NeurIPS_ML4PS_2019_83.pdf)

[4] Y. Wang, et al, arXiv:1801.07829. (DGCNN)

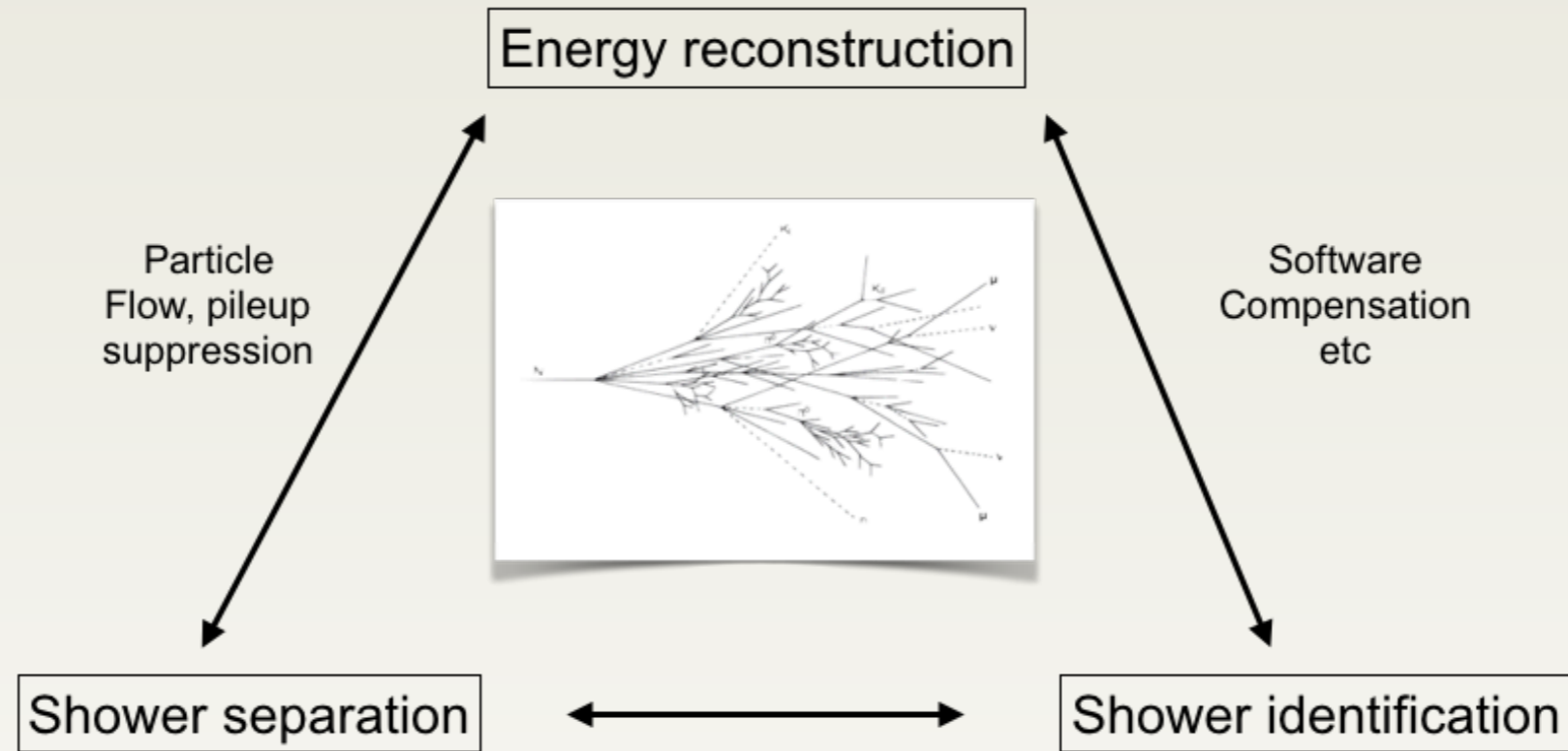
# Edge Classifier in calorimeter

- CMS HGCal
- Single charged pions in 0 PU



- Excellent discrimination between noise and signal
- Needs more developments for fractional assignments, very small objects
- $N \times K$  edges need to be evaluated to determine object and its properties
  - Mean over edges for properties or e.g. weight with edge score

# Take a step back



- What we actually want: particle ID, momentum, position
- Segmentation just a tool
- Standard chain has many redundancies
  - ▶ Seeding (pattern recognition)
  - ▶ Clustering (pattern recognition)
  - ▶ Software compensation (pattern recognition)
  - ▶ ID (pattern recognition)
  - ▶ PFlow (pattern recognition)
- Always the same patterns
- One-stage approach can save resources and is easier to maintain
- Look at *fast and graph-compatible* computer vision approaches

# Segmentation and Clustering: CV

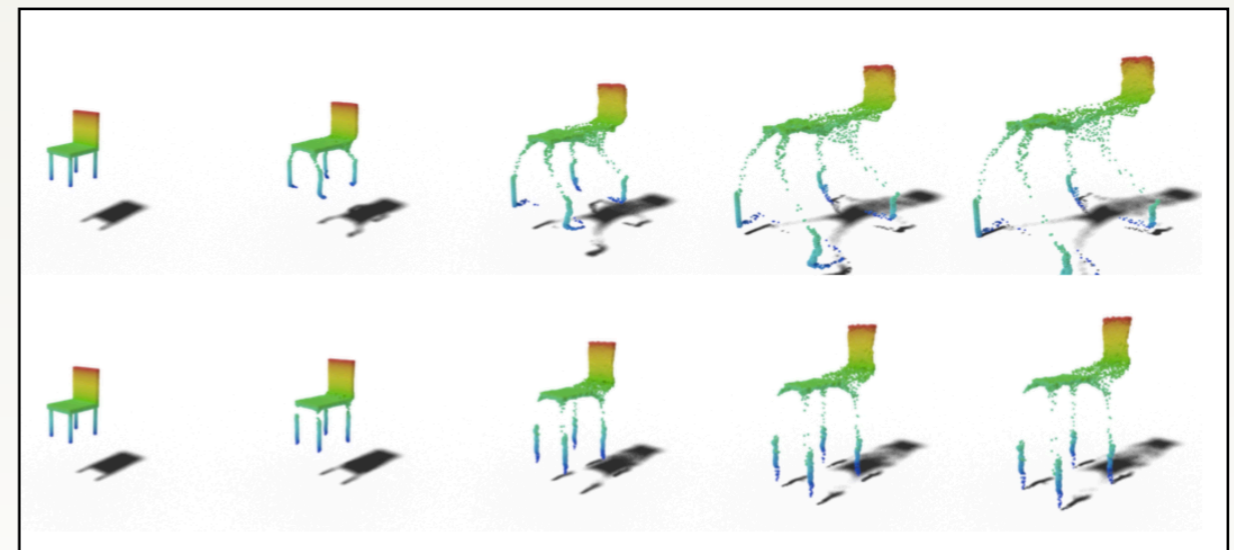
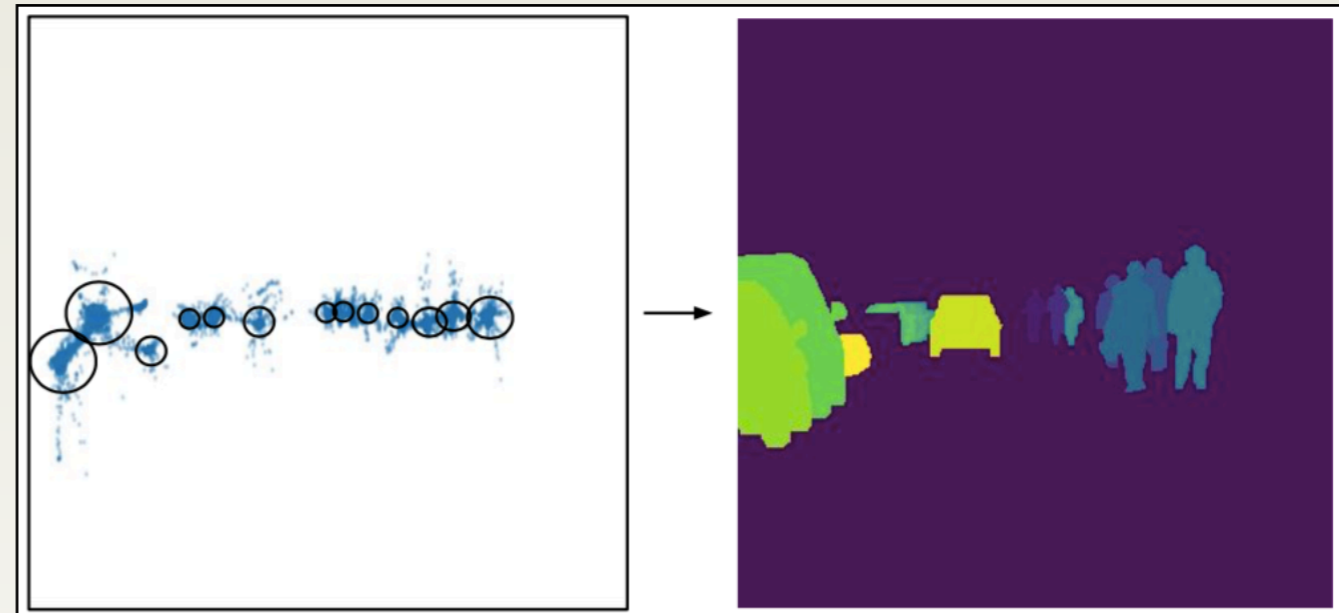
- Maximum number of objects per image/point cloud:  
number of pixels/vertices
- Learn to move pixels towards the object center
- Map to Gaussian probability

$$\phi_k(e_i) = \exp\left(-\frac{\|e_i - C_k\|^2}{2\sigma_k^2}\right)$$

- Assign seed score

$$\mathcal{L}_{\text{seed}} = \frac{1}{N} \sum_i \mathbb{1}_{\{s_i \in S_k\}} \|s_i - \phi_k(e_i)\|^2 + \mathbb{1}_{\{s_i \in \text{bg}\}} \|s_i - 0\|^2$$

- Collect (from highest seeds score) around the seeds
- *'Only' performs segmentation*
- *Heavily relies on the center of an object*
  - *Problematic concept for particles*

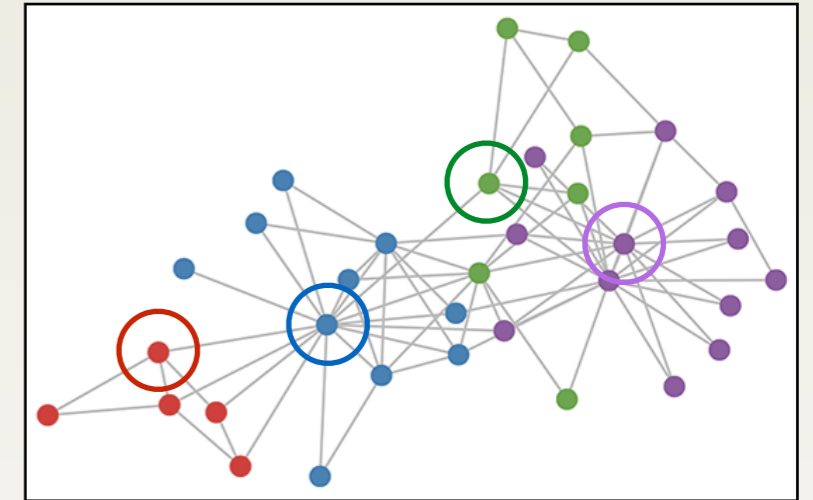


D. Neven et al, arXiv:1906.11109  
B. Zhang, P. Wonka, arXiv:1912.00145

# Object condensation

- Aim

- ▶ Determine object properties (e.g. particle 4 momenta, ID) (graphs, images, ...)
- ▶ Aggregate all object properties in representative 'condensation point'
- ▶ *Detach* input space (3D/4D/5D) from output space
- ▶ Resolve ambiguities without IoU (boxes) concept
- ▶ Allow for fractional/ambiguous assignments



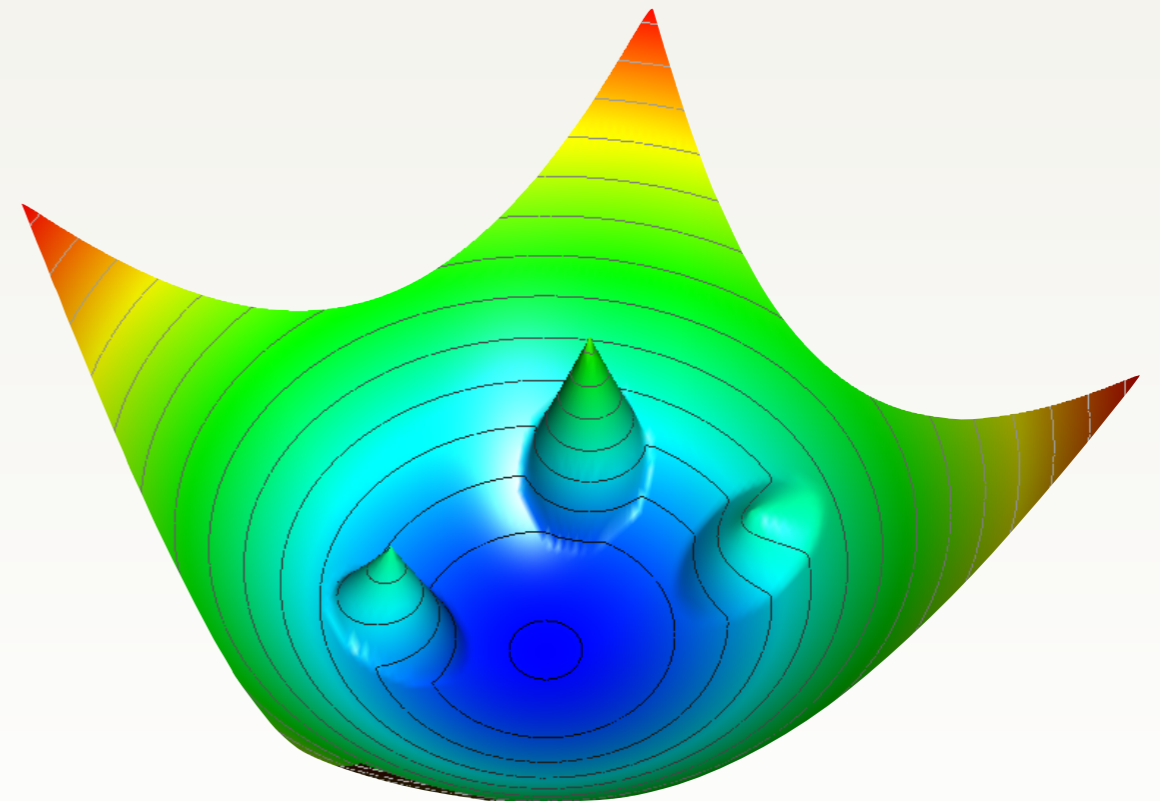
- Define truth:

- ▶ Assign each vertex to one object (e.g. highest fraction)
- ▶ Assign all object properties to each assigned vertex

- Predict per vertex

- ▶ Object properties
- ▶ Confidence  $\beta$
- ▶ Cluster coordinates  $x$

- Define charge, attractive and repulsive potential



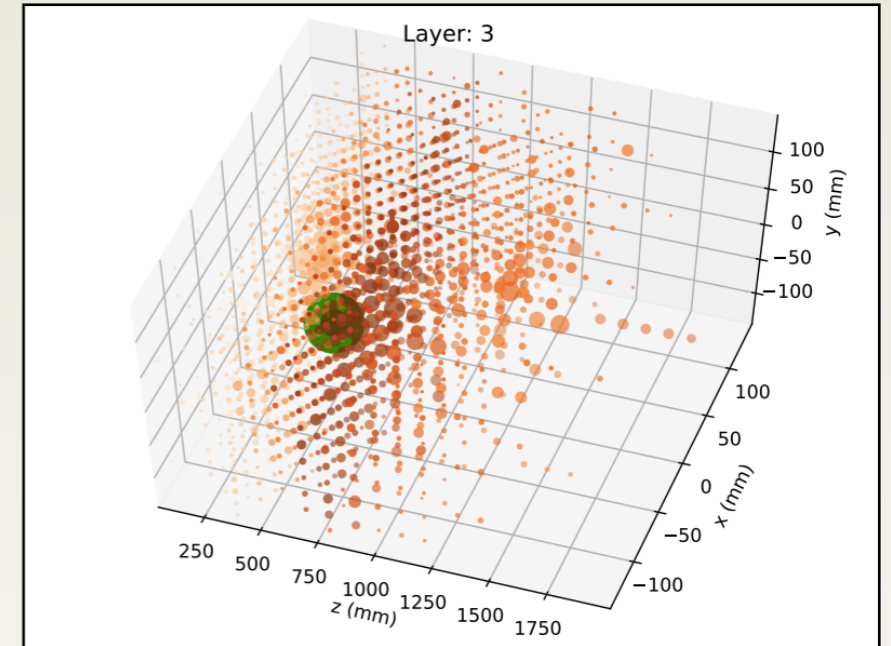
# Condensate and predict

$$\check{V}_k(x) = ||x - x_\alpha||^2 q_{\alpha k}, \text{ and}$$

$$\hat{V}_k(x) = \max(0, 1 - ||x - x_\alpha||) q_{\alpha k}.$$

Maximum charge vertex for object k

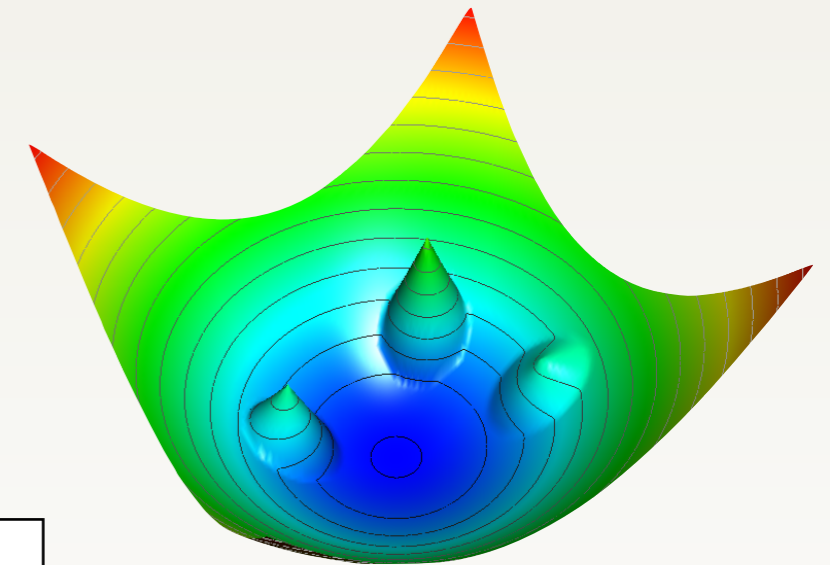
- Maximum  $\beta$ /charge vertices *are* center points \*
- Encourage network to select one representative point per object k



$$L_\beta = \frac{1}{K} \sum_k (1 - \beta_{\alpha k}) + s_B \frac{1}{N_B} \sum_i^n n_i \beta_i,$$

- Also weight object property loss with  $\beta$

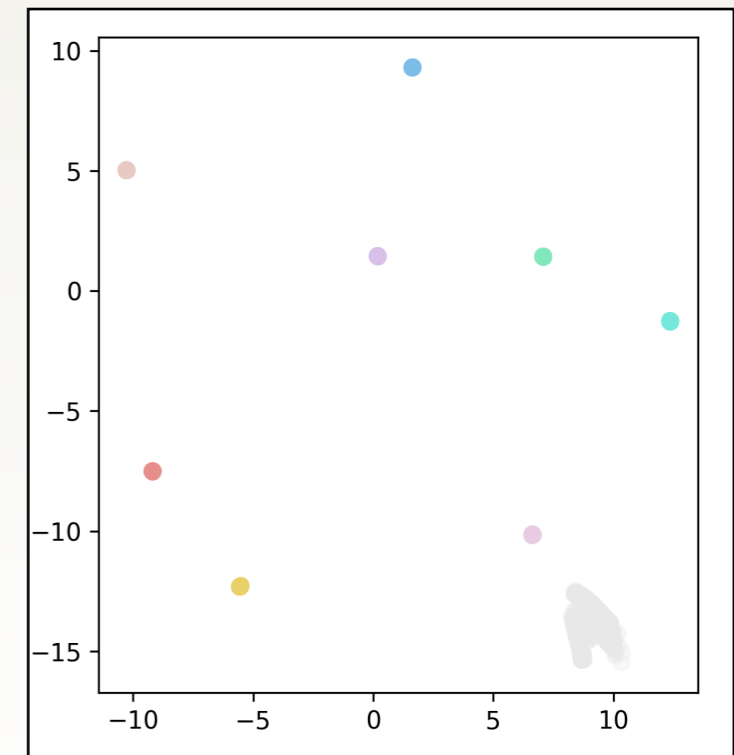
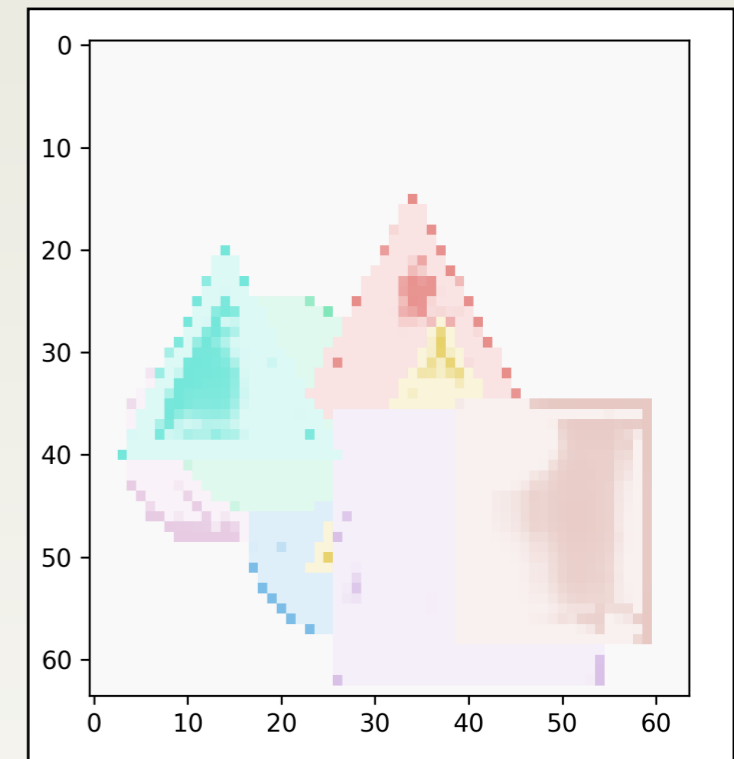
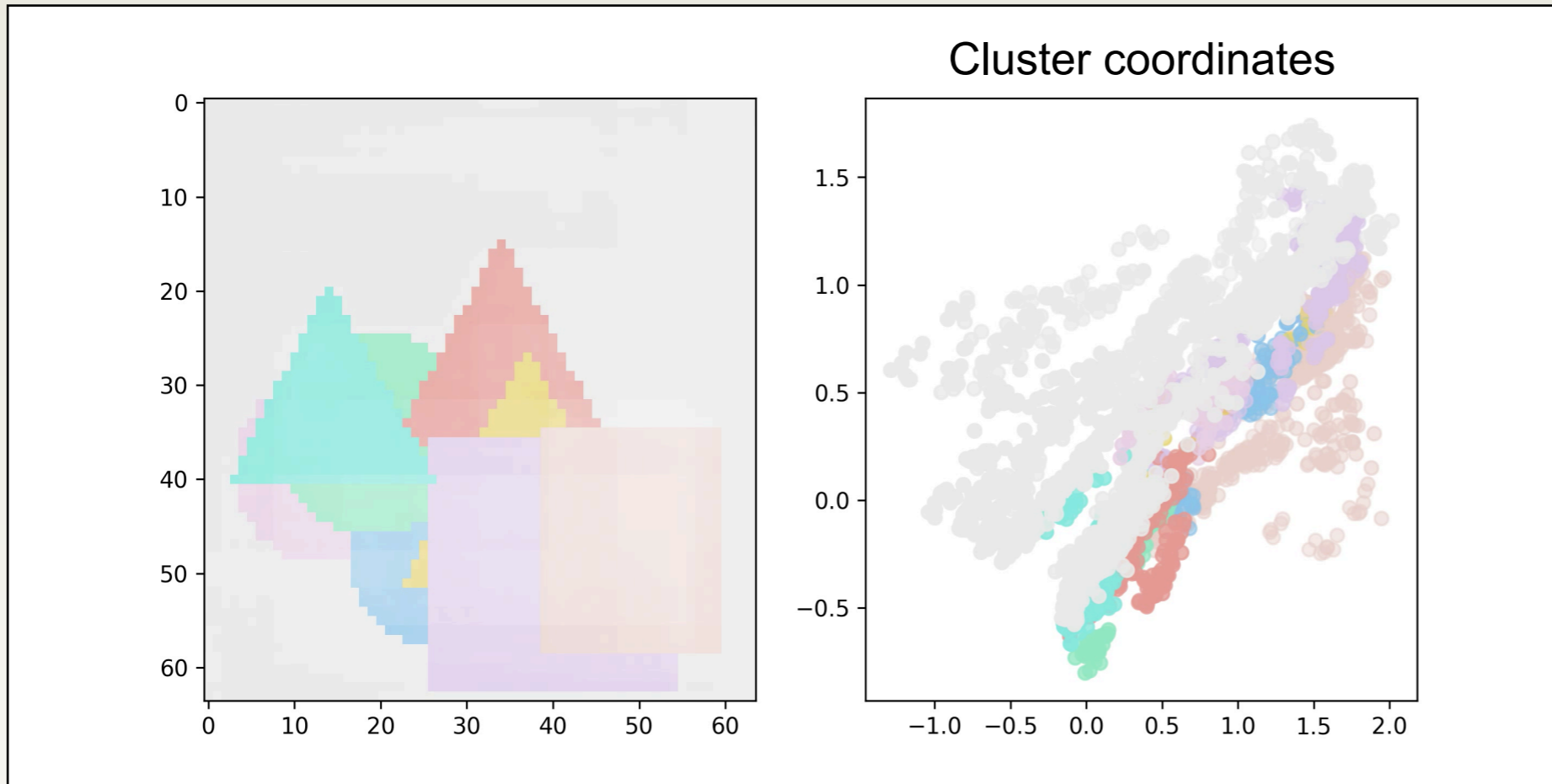
$$L_p = \frac{1}{\sum_{i=0}^N (1 - n_i) \operatorname{arctanh}^2 \beta_i} \sum_{i=0}^N L(t_i, p_i) (1 - n_i) \operatorname{arctanh}^2 \beta_i$$



- *Condensation points will carry all object properties*
- *Very natural approach for dynamic graph NN*

\*NB: Removes saddle point for large N  
JK, arxiv:2002.03605

# Example on image data



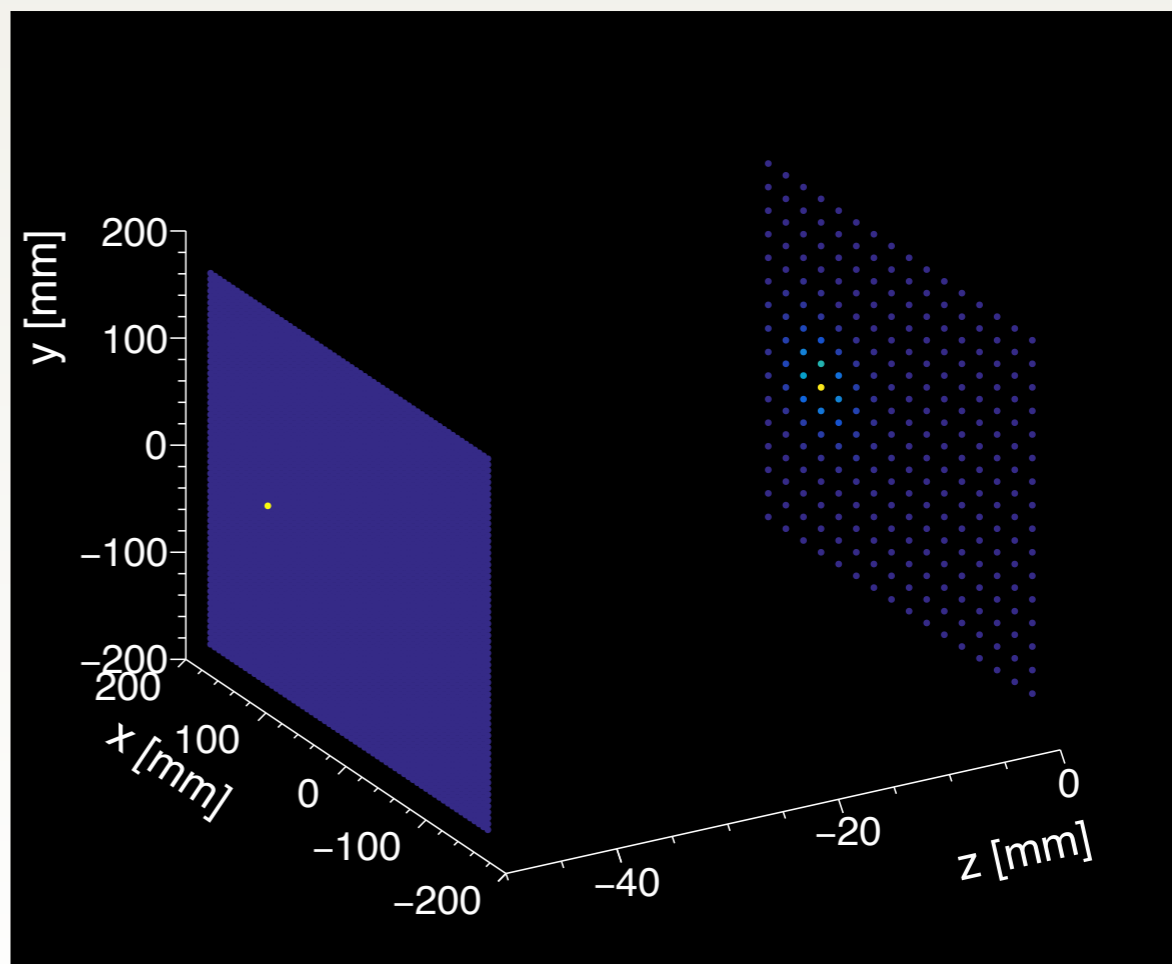
- Proof of principle using images with large overlaps
  - Condensation, object ID
  - Rather simple CNN
- Inference
  - Start with highest  $\beta$  vertex, collect points in  $t_d \cong 0.9$ 
    - Get object properties
  - Repeat until  $\beta_{\min} \cong 0.1$

JK, arxiv:2002.03605

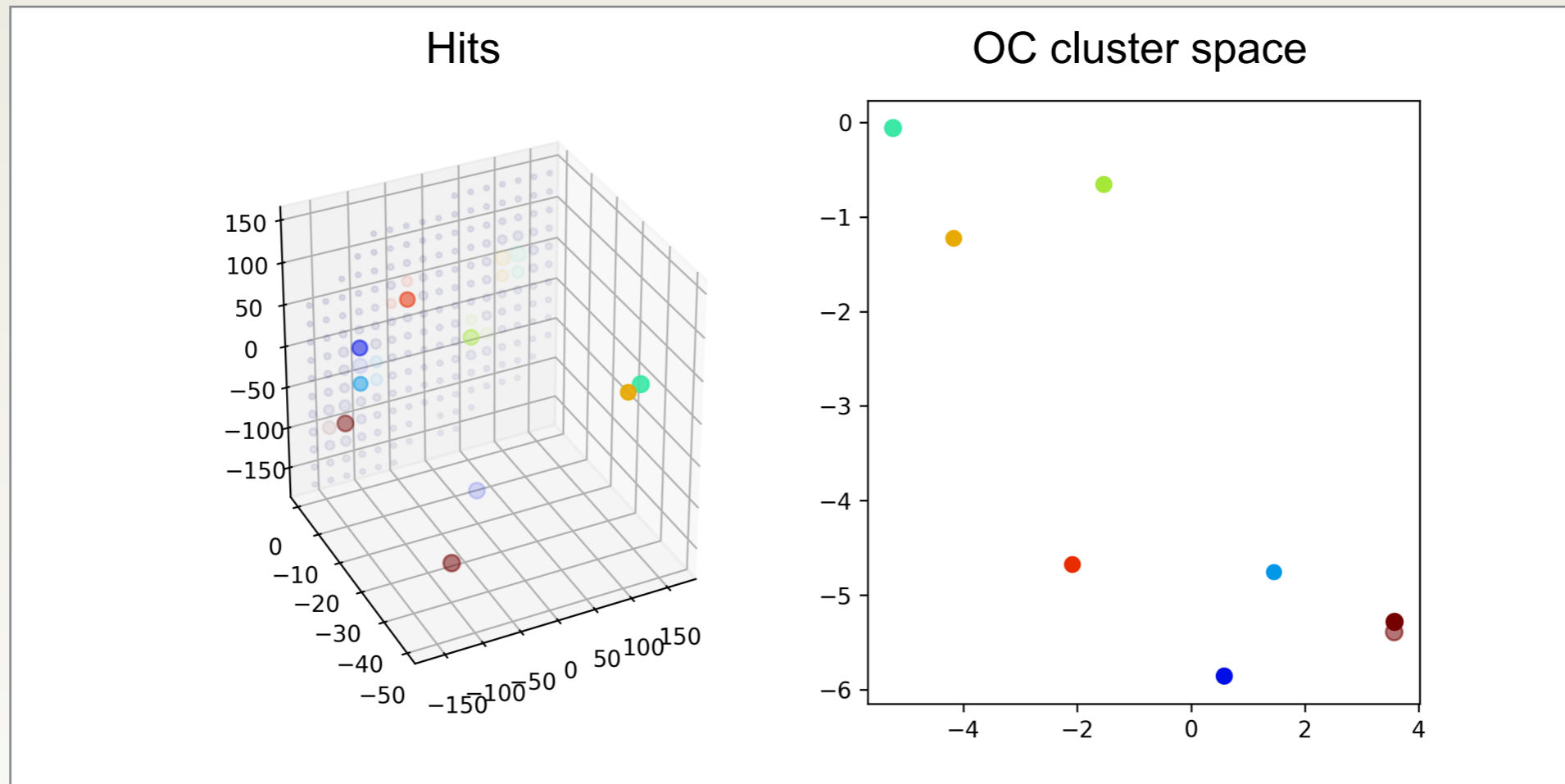
# Application to Particle Flow

- Simplified detector
  - ▶ “Flat” in x,y: not curved
  - ▶ ECal: 16 x 16 cells, each 22 x 22 mm<sup>2</sup> x 26 cm lead tungstate (CMS ECal)
  - ▶ No magnetic field
  - ▶ “Tracker”: 300μm silicon 5.5 x 5.5 mm<sup>2</sup> sensors, placed 5 cm in front of calorimeter
    - ▶ Assign Gaussian smeared track momentum to highest energy hit
    - rel. resolution =  $((p/100.)*(p/100.)*0.04 + 0.01)$
  
- Shoot electrons and photons (50/50)
  - ▶ E = 1 - 200 GeV
  - ▶ x,y random between -14 and 14 cm
  
- 1-9\* particles per event
  - ▶ Discard particle if no sensor can be found where it leaves the highest fraction
  
- Use GravNet

Geometry

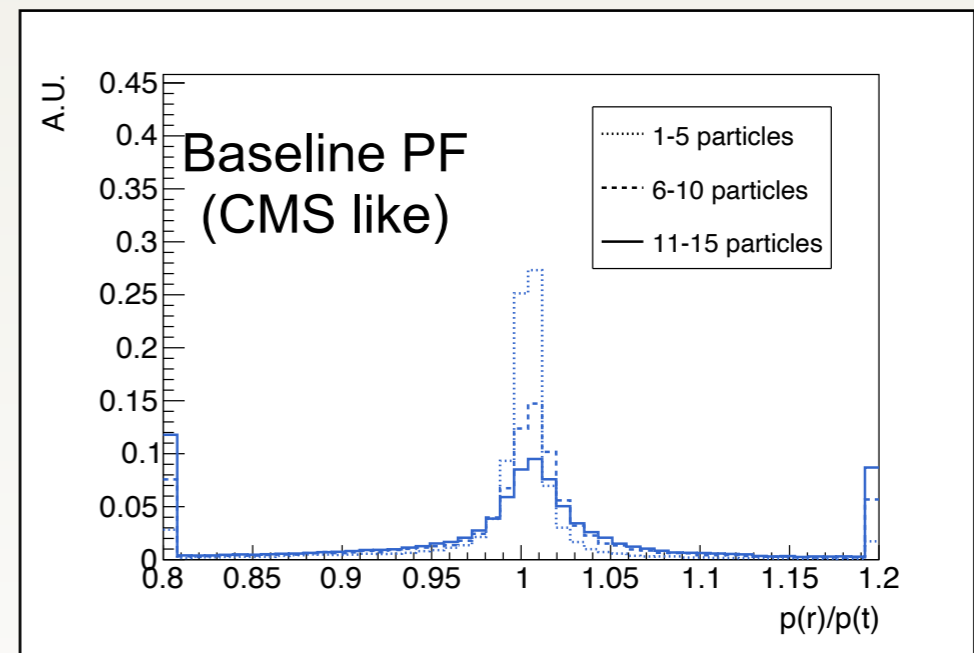
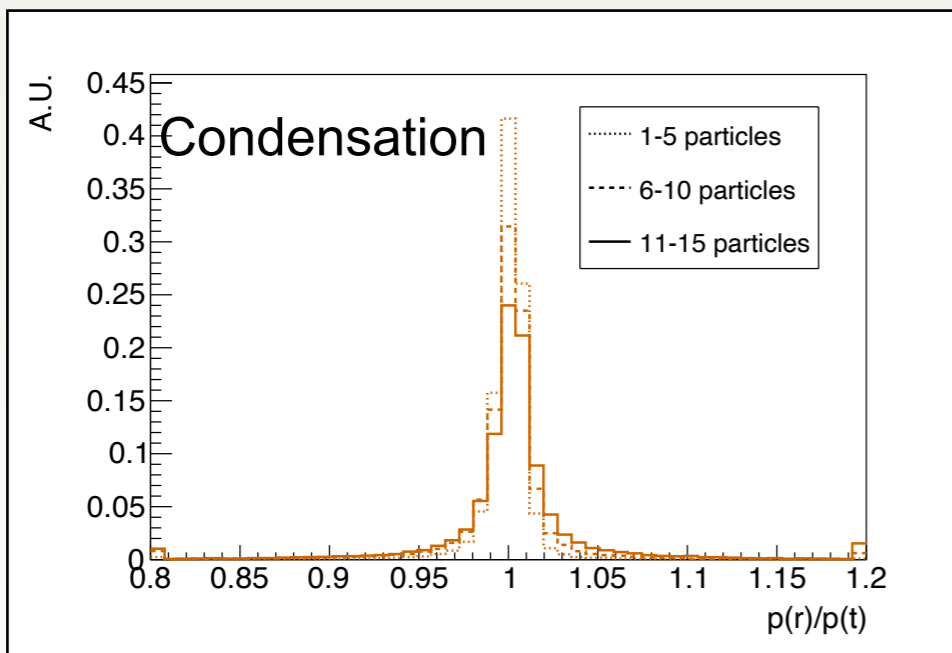
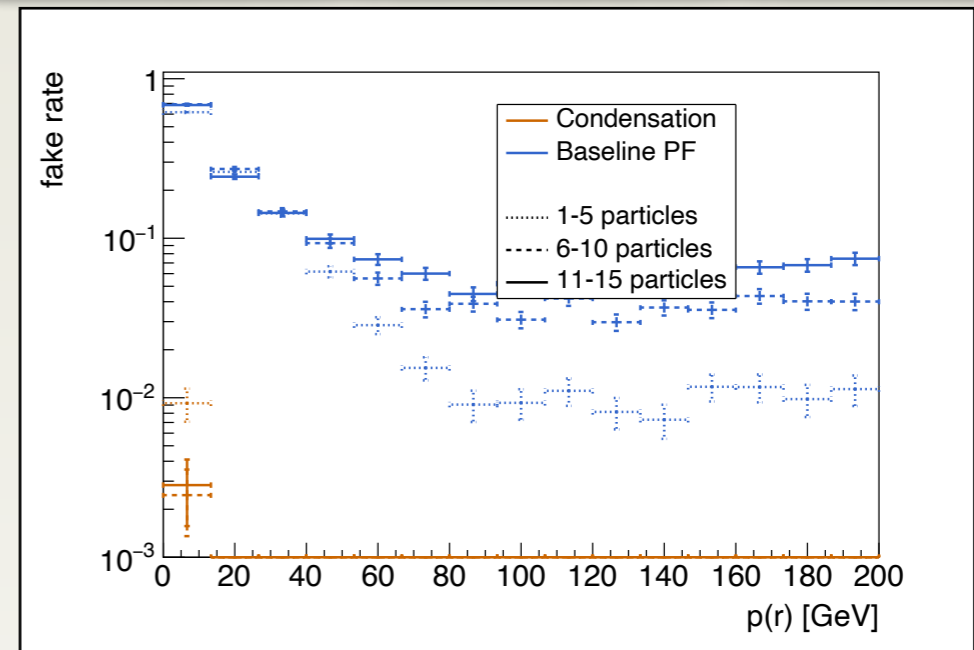
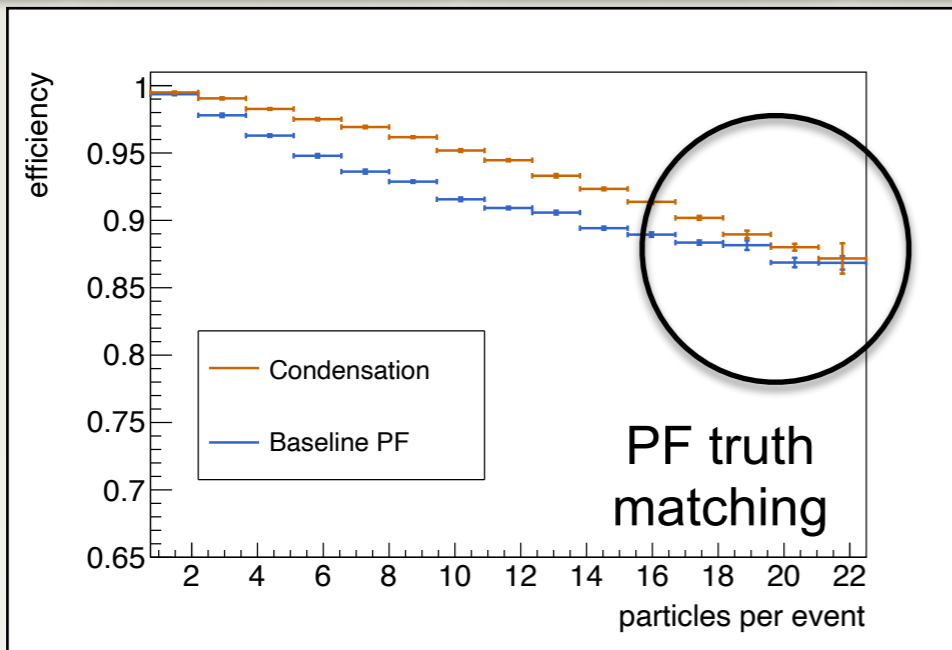


# Segmentation / Postprocessing



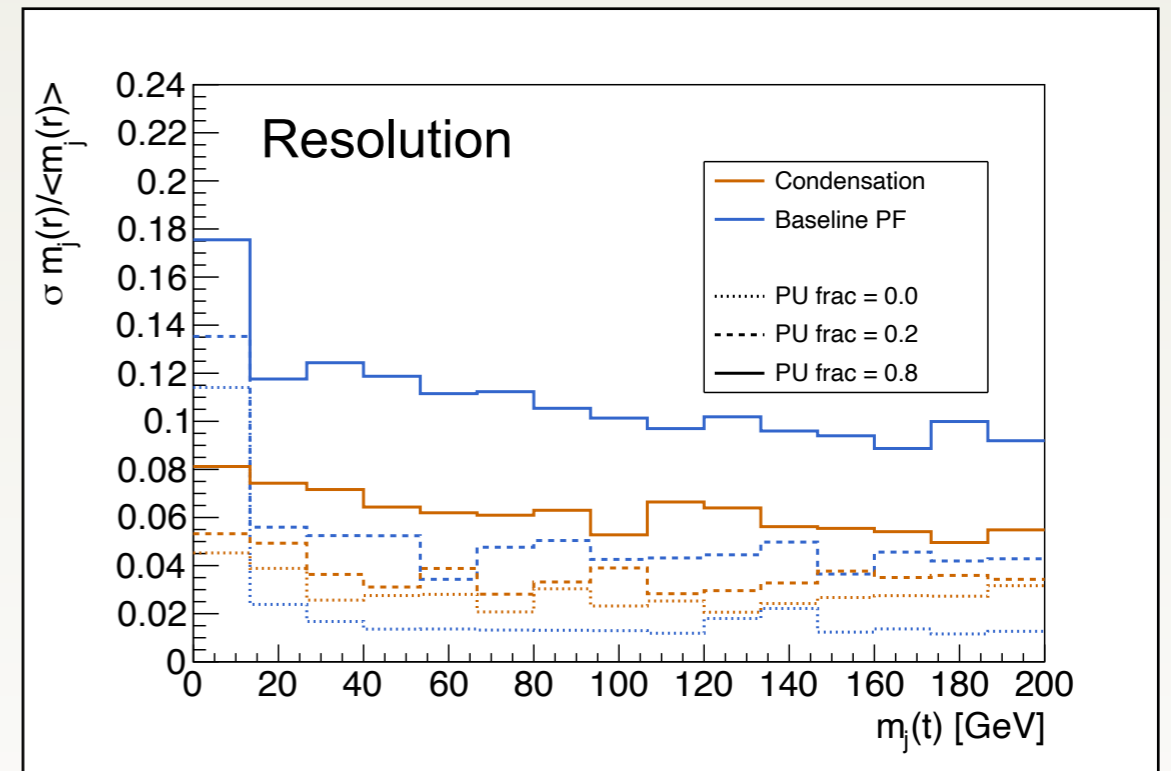
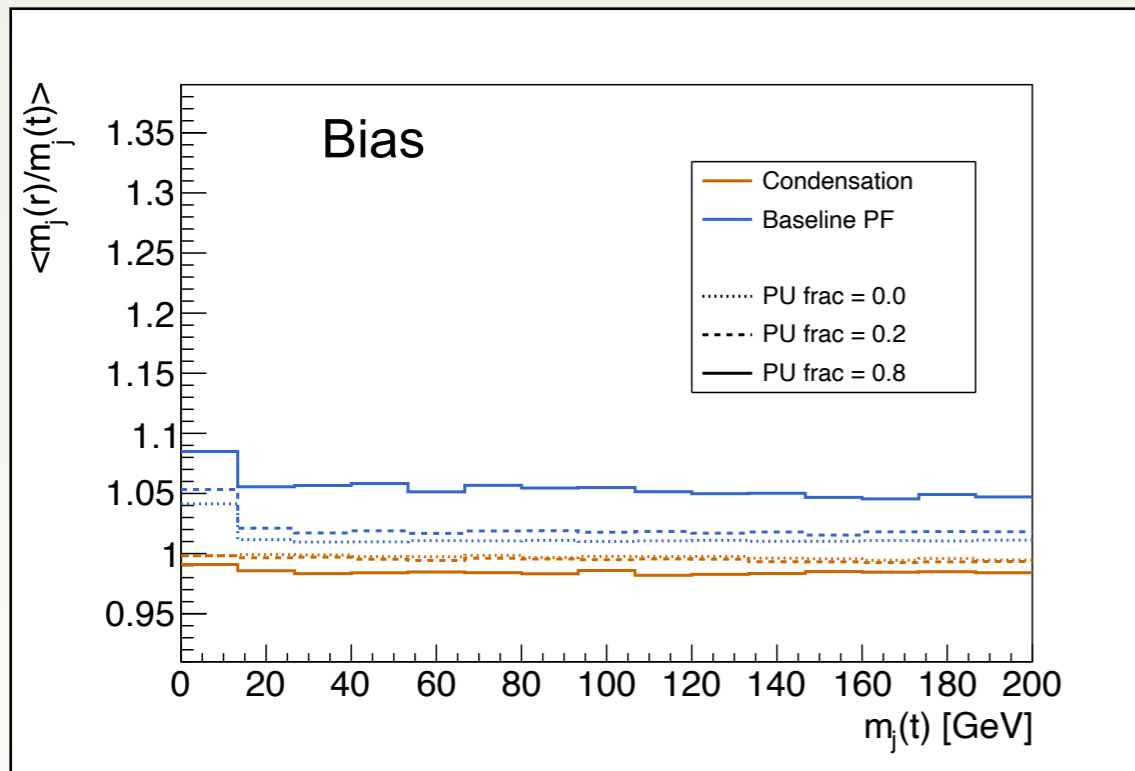
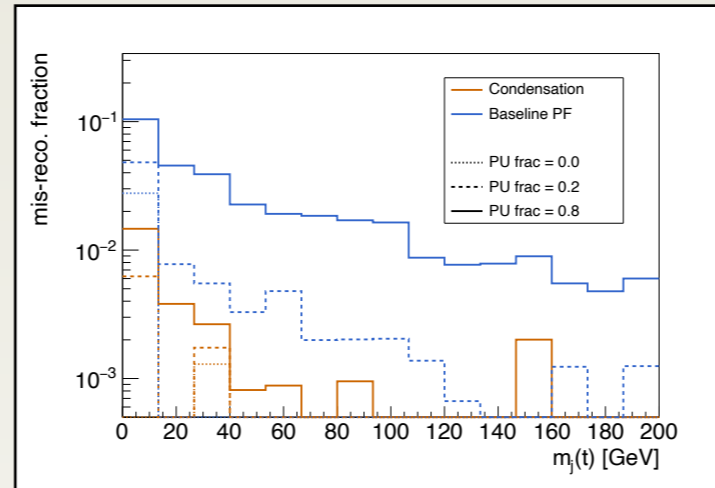
- Start with highest  $\beta$  vertex, collect points in  $td \cong 0.8$
- Get object properties
- Repeat until  $\beta_{\min} \cong 0.1$

# Particle Efficiency and Response



- Excellent extrapolation properties beyond training conditions
- Low fake rate, and fakes only at low energies
- Improved single particle resolution

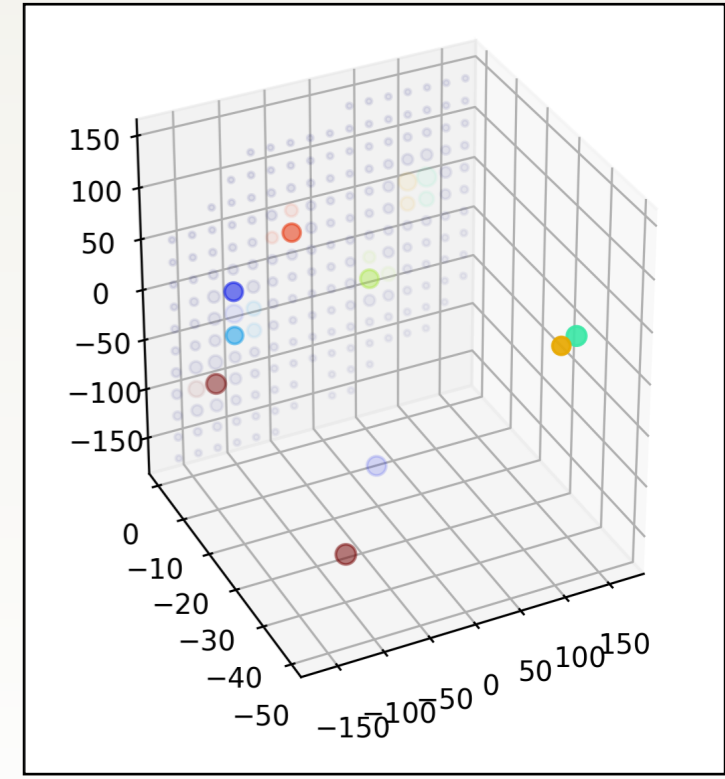
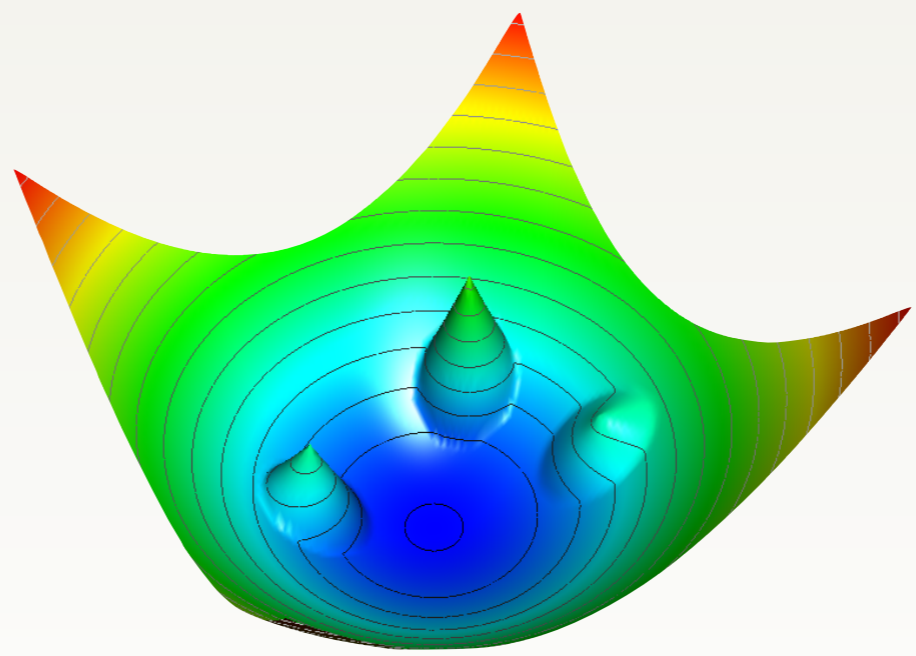
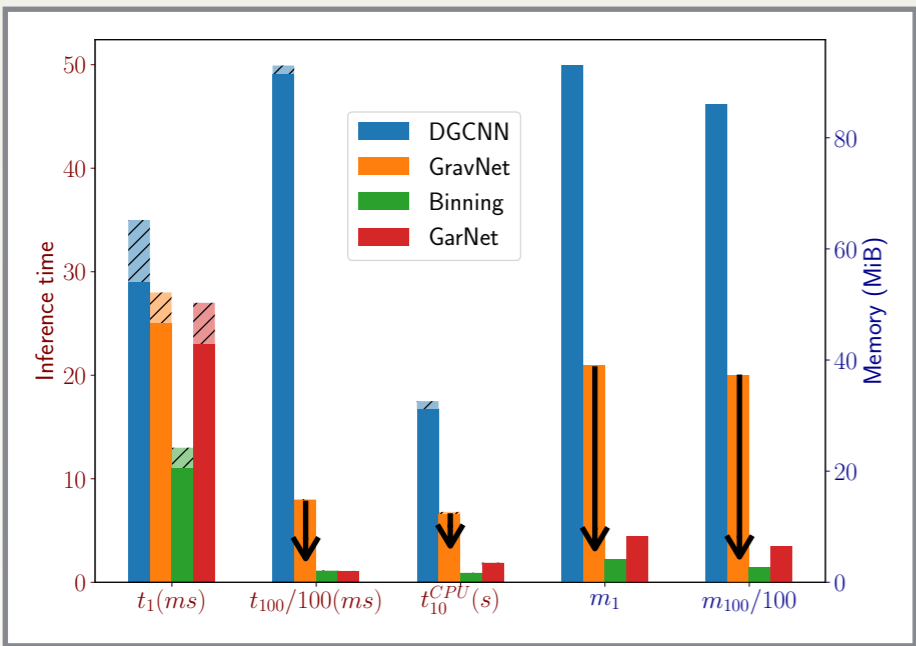
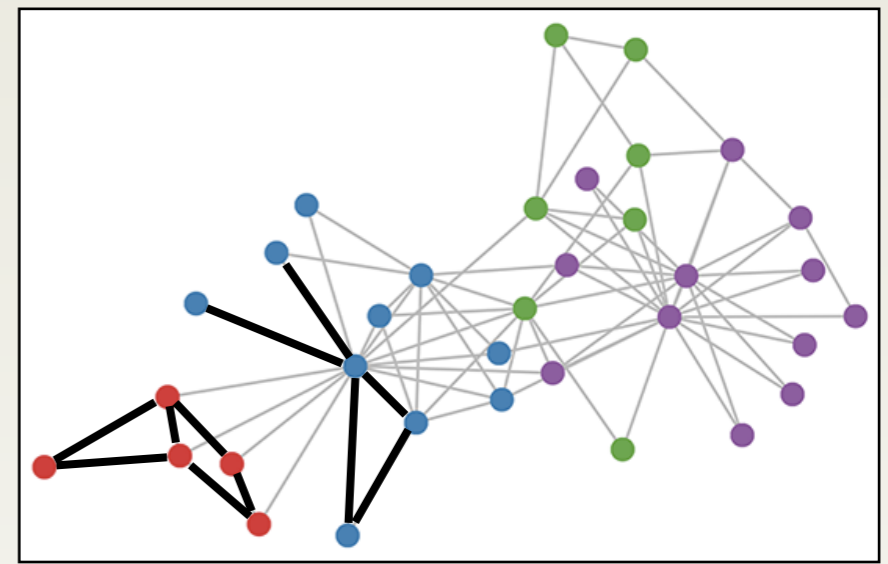
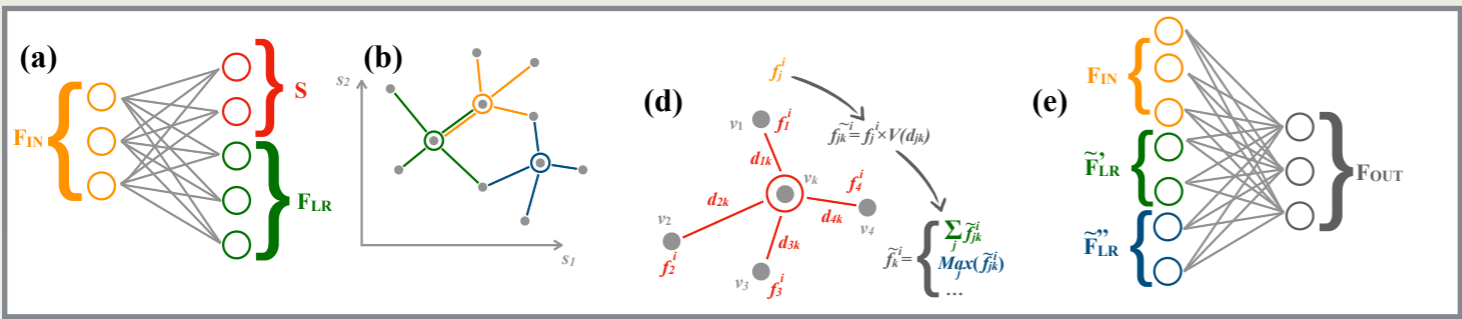
# “Jet” mass resolution



- Standard PF does very well for 0 PU fraction (built-in energy conservation)
- With higher PU fraction identification of individual particles way more important: **object condensation starts to be better**, in particular at low momenta

JK, arxiv:2002.03605

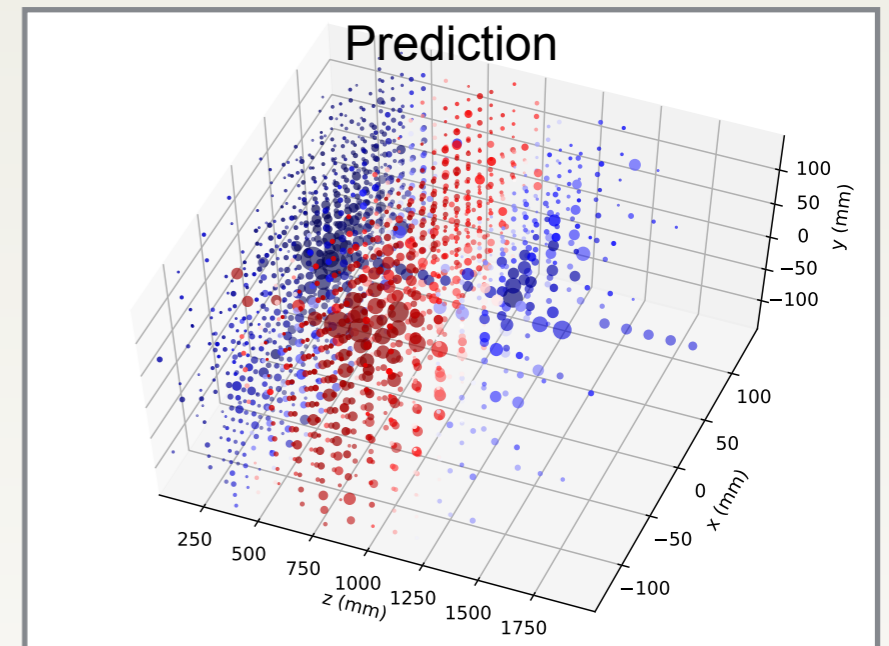
# It all comes together



- All tools at hand
- Near future will be exciting

# Summary

- High granularity calorimeters are widely accepted in HEP to control backgrounds, pileup and precise particle flow
- Very promising performance of ML algorithms in high granularity calorimeters
  - ▶ Direct link to particle flow
- Pushing forward developments for particle reconstruction
- Pushing forward new machine learning approaches





# Backup

# A look at computer vision



- Well known from object detection in images
- Two main approaches:
  - ▶ “Traditional’ anchor point based approaches [1-4], ...
  - ▶ Anchor-free approaches, using each pixel [5,6, ...]

[1] J. Redmond et al, arXiv:1506.02640

[2] Y. Hu et al, arXiv:1803.11187

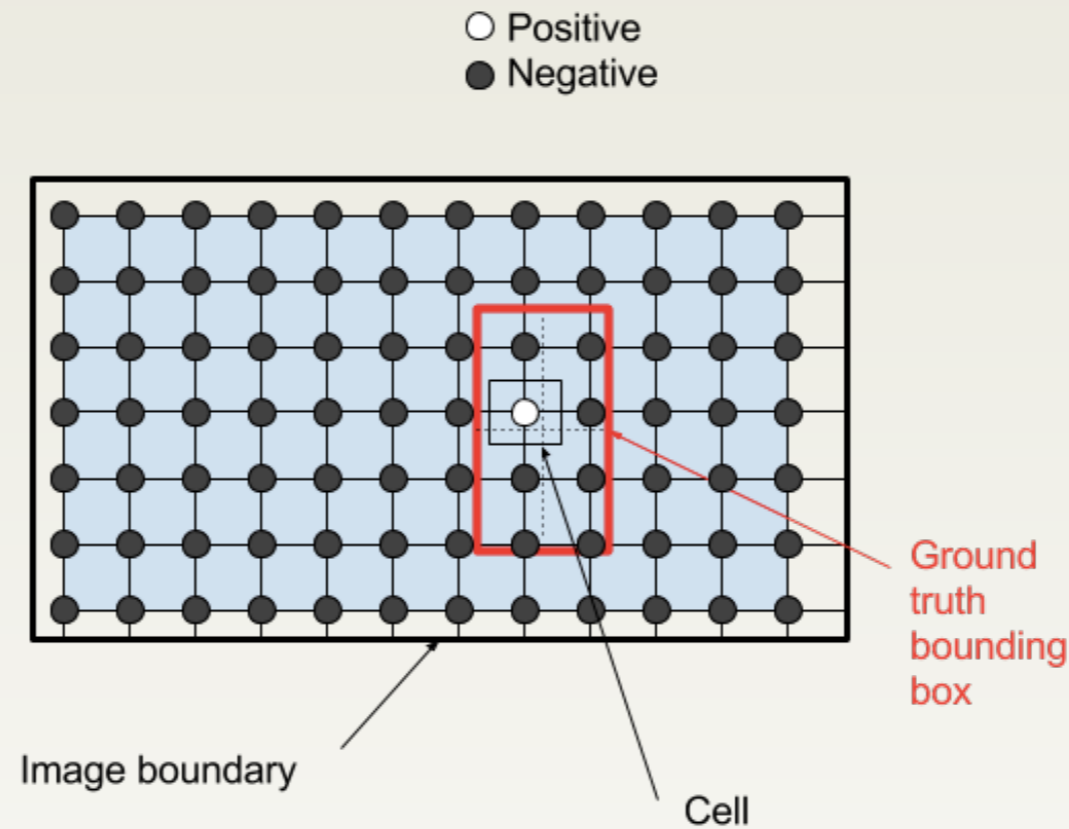
[3] R. Girshick, arXiv:1504.08083

[4] T. Lin et al, arXiv:1708.02002

[5] N. Wang et al, arXiv:1904.01355

[6] X. Zhou et al, arXiv:1904.07850

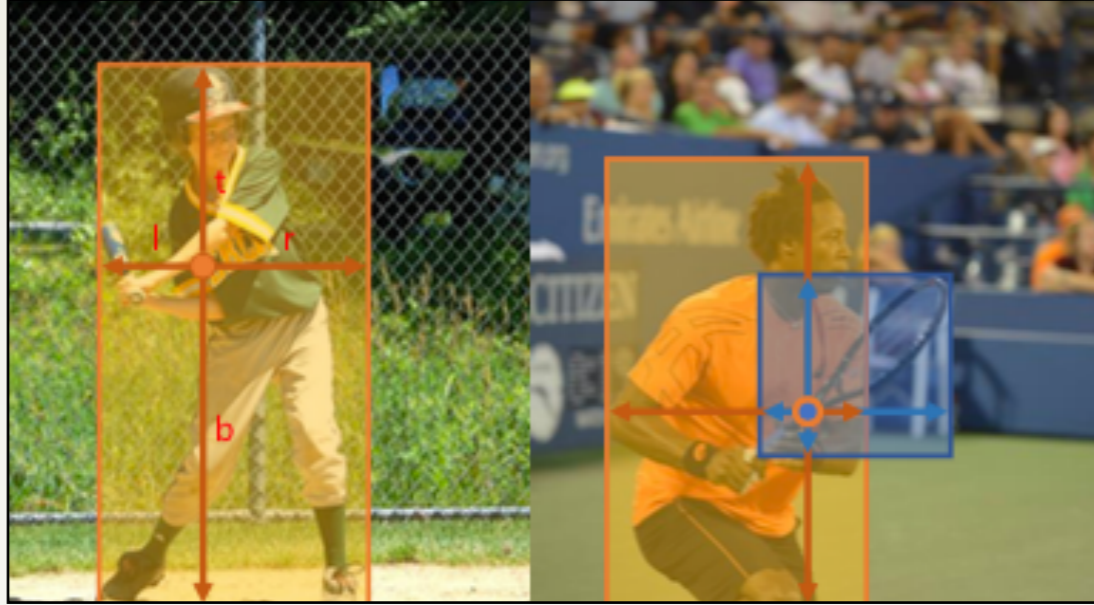
# Anchor point based methods



- Anchor points ( $M \times M$  per image)
- Assign object score/bounding box to anchor point
- Object can be found multiple times
- Anchor points grow with with  $N^{(\text{dim})}$ , make implicit assumptions on object size
- *Not suitable for reconstruction based on high-dimensional detector signals*

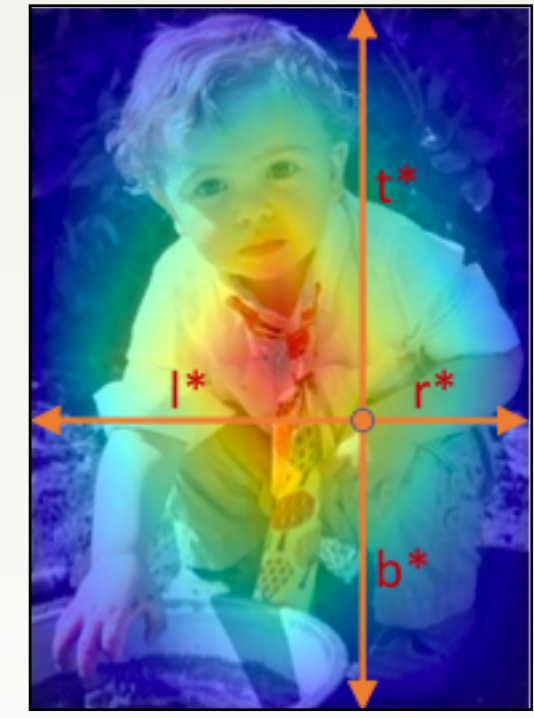
# Key point methods

- Identify key points of the object
- Predict object properties from key points



Problem: identify the key points

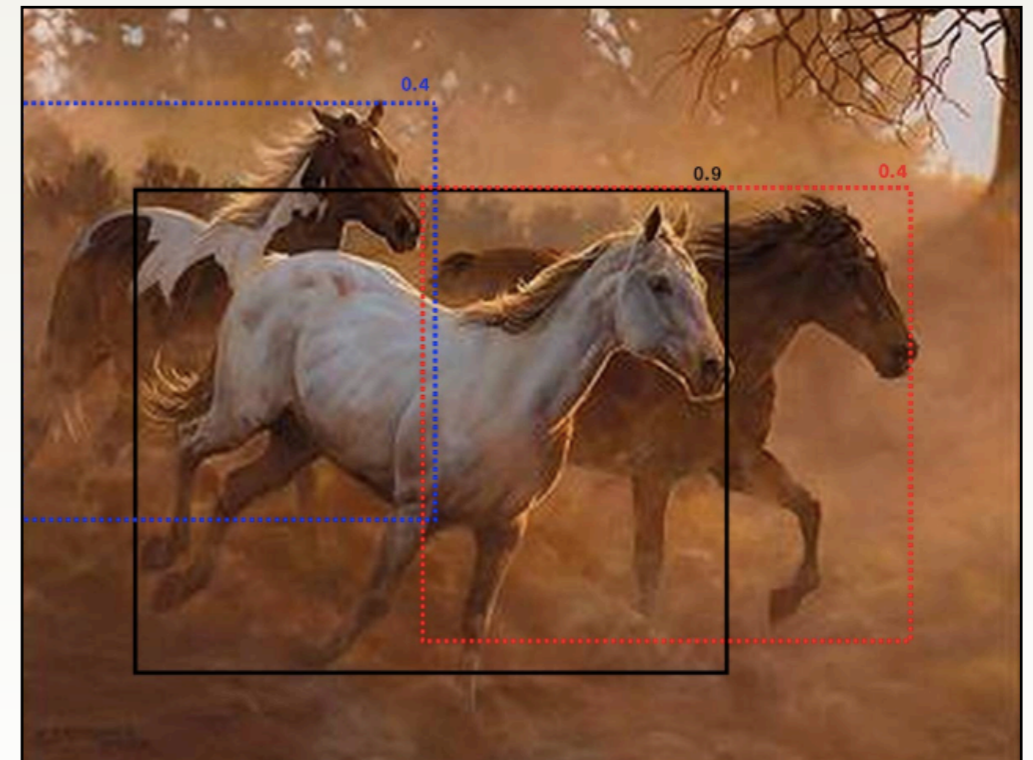
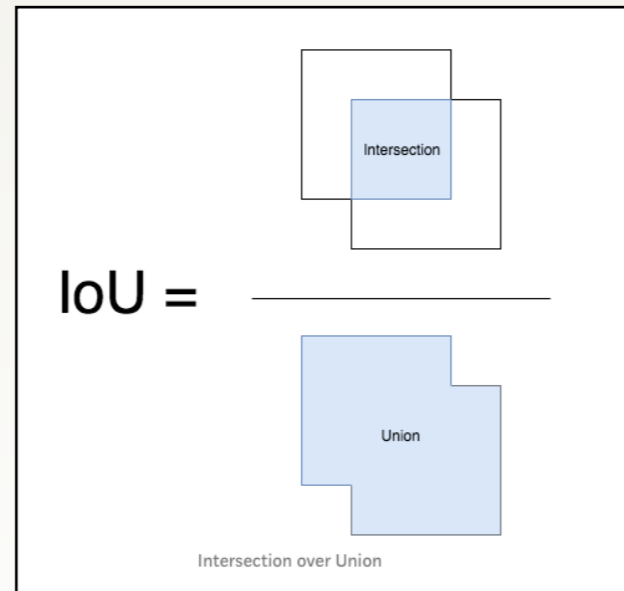
- Also predict 'center-score'
- Select highest score in the area as key point
  - ▶ **Seed identification!**
  - ▶ Heavily relies on objects to have a center: problematic for a particle
- Remaining ambiguities still need to be resolved

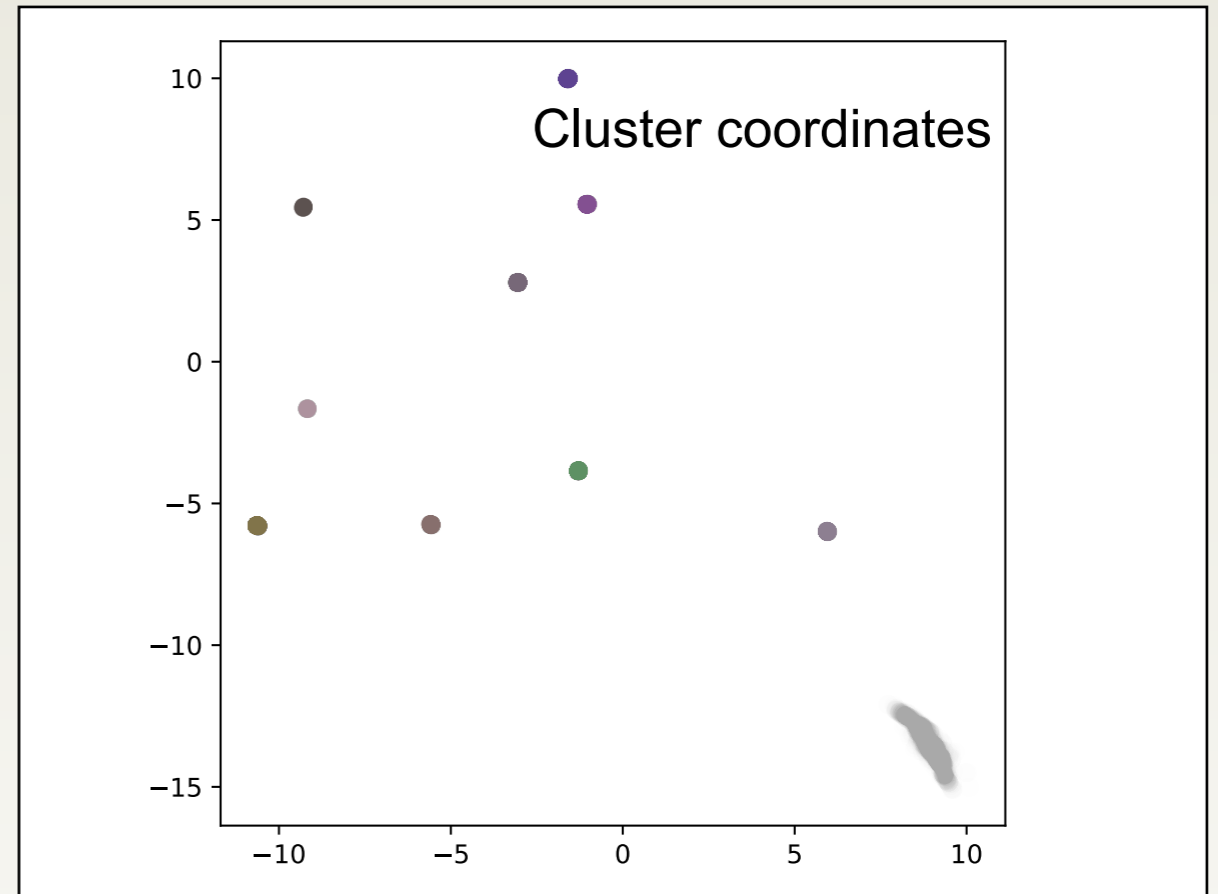
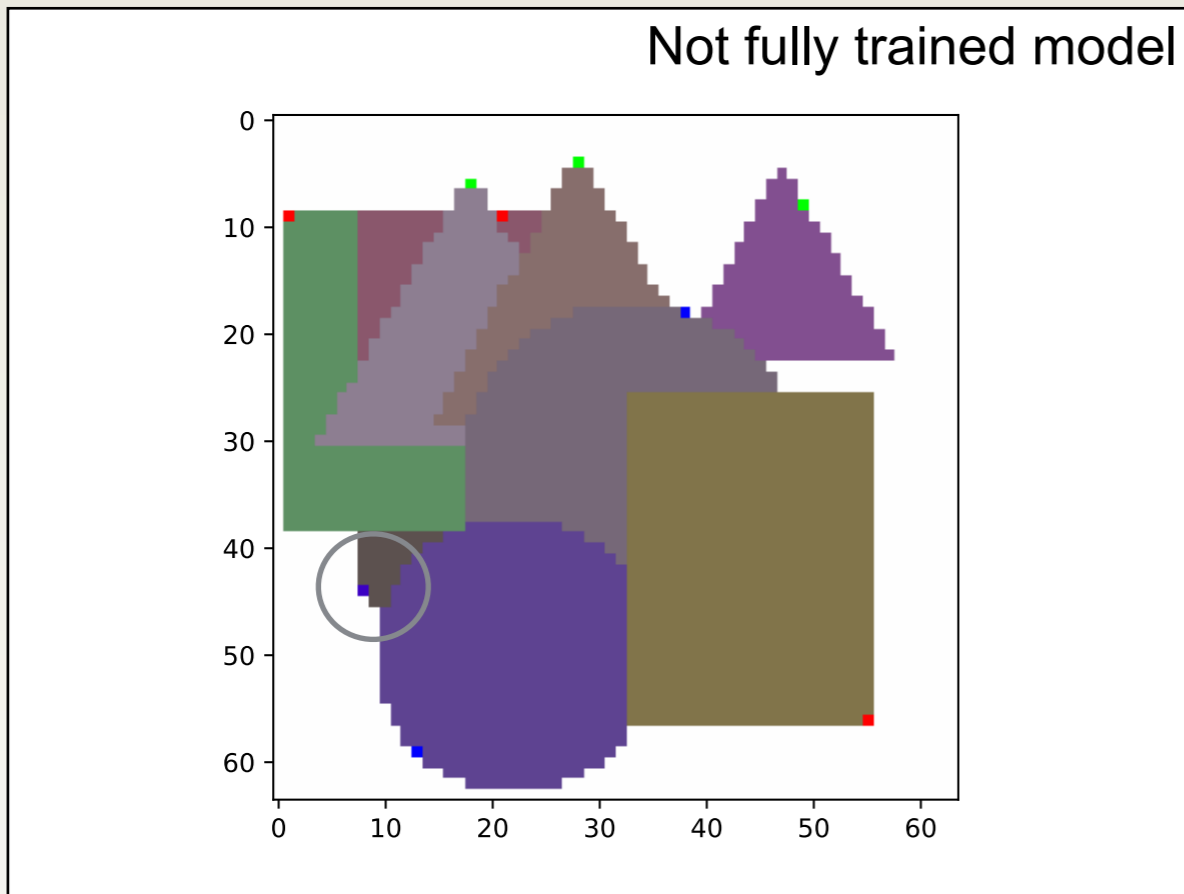


N. Wang et al, arXiv:1904.01355  
 X. Zhou et al, arXiv:1904.07850

# Non maximum suppression

- Start with highest score
- Downweight 'close' by objects using IoU (Soft NMS)
- Relies on bounding boxes
- *Not easily adaptable to particles in detectors*





- Inference

- ▶ Start with highest  $\beta$  vertex, collect points in  $t_d \cong 0.9$ 
  - ▶ Get object properties
- ▶ Repeat until  $\beta_{\min} \cong 0.1$

$$\check{V}_k(x) = ||x - x_\alpha||^2 q_{\alpha k}, \text{ and}$$

$$\hat{V}_k(x) = \max(0, 1 - ||x - x_\alpha||) q_{\alpha k}.$$

- Object condensation allows to predict K particles from N detector inputs

- ▶ Paves the way for one-stage approaches in reconstruction
- ▶ 'Just' needs to be combined with the networks proven to work well

JK, arxiv:2002.03605