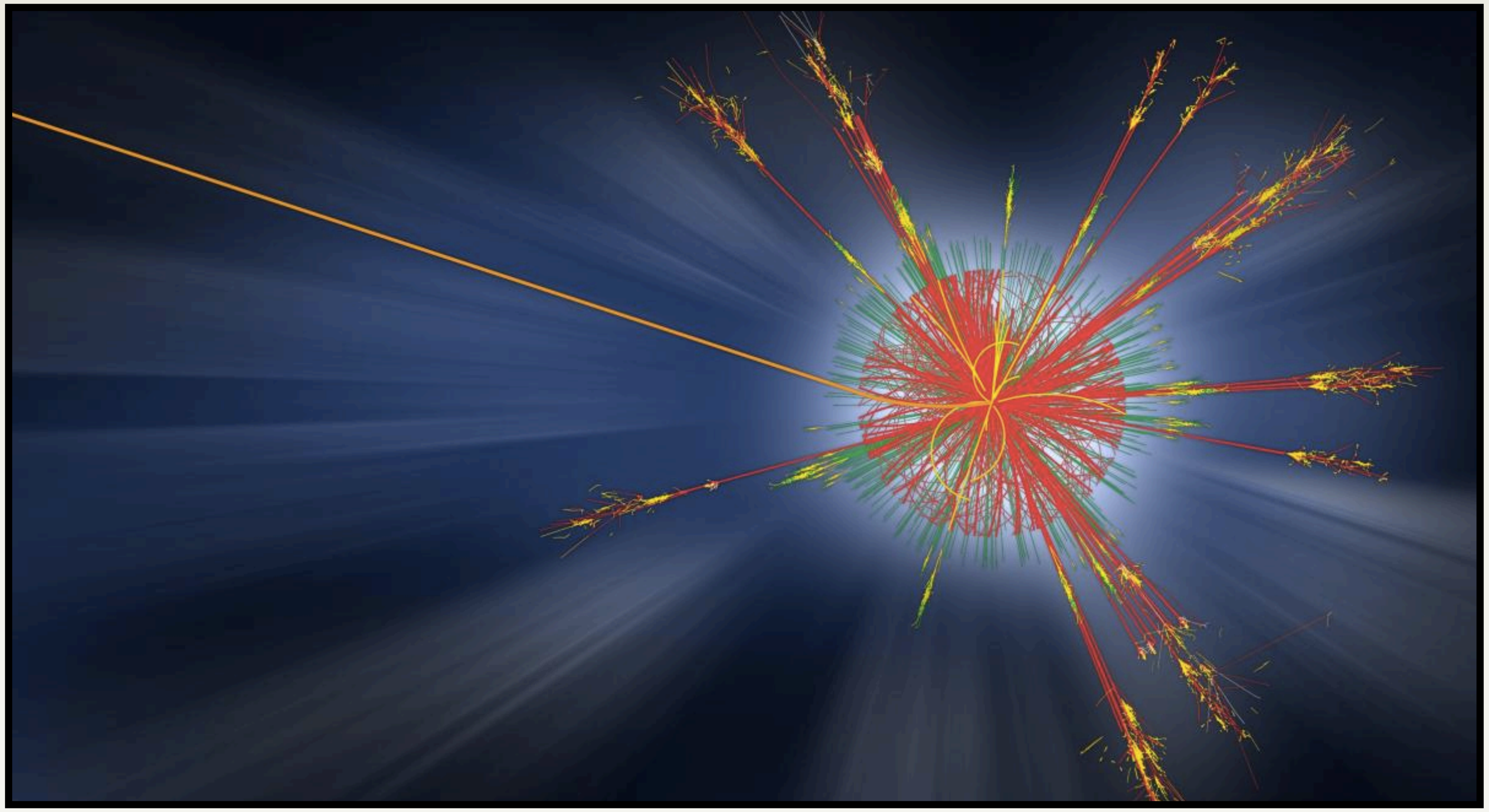
Object Condensation for Particle Flow

Jan Kieseler

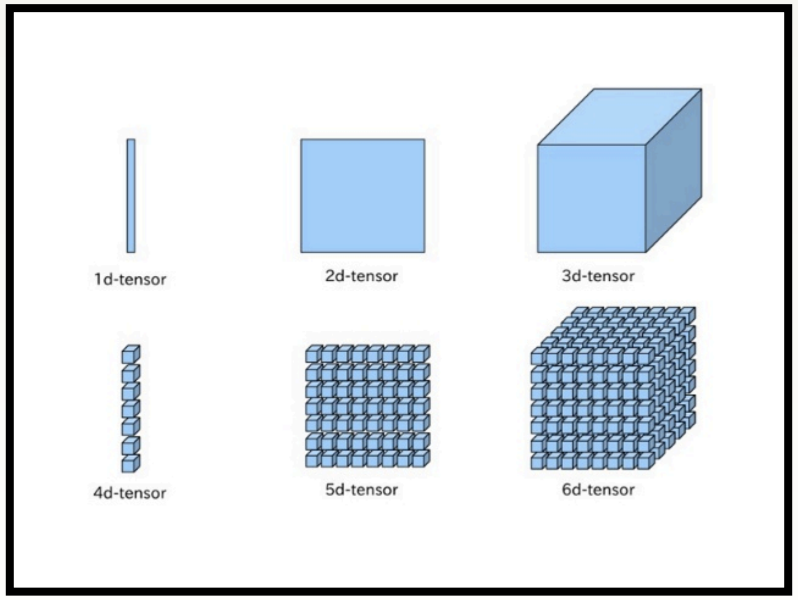
07.04.2020

N to K Problem for Reconstruction

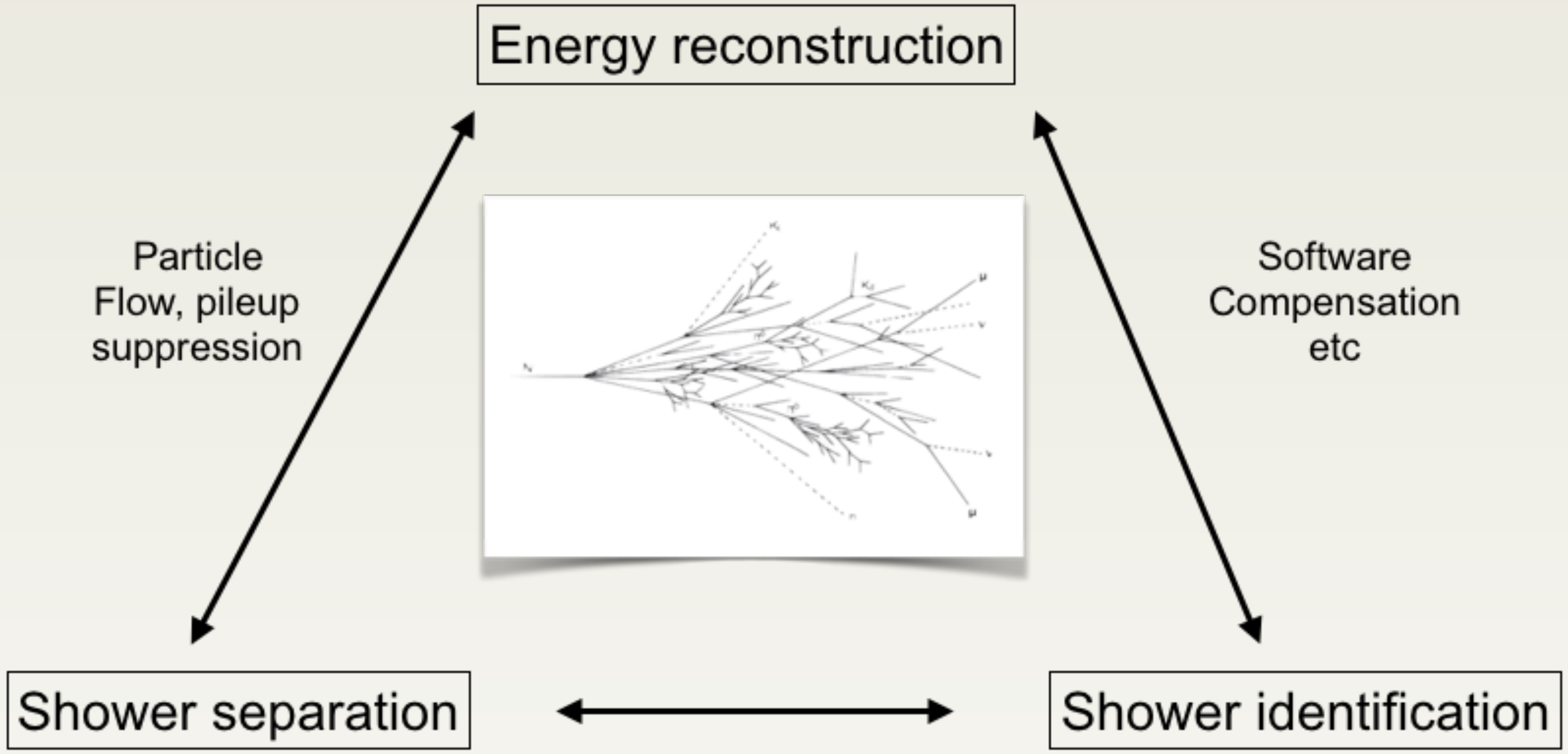


1 muon
3 few jets
....

- Each event has a different number of particles
- Detector hits need to be clustered/linked to physics objects
- DNNs prefer fixed-size outputs



What do we actually want



- What we actually want: particle ID, momentum, position
- ***Segmentation/clustering is 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

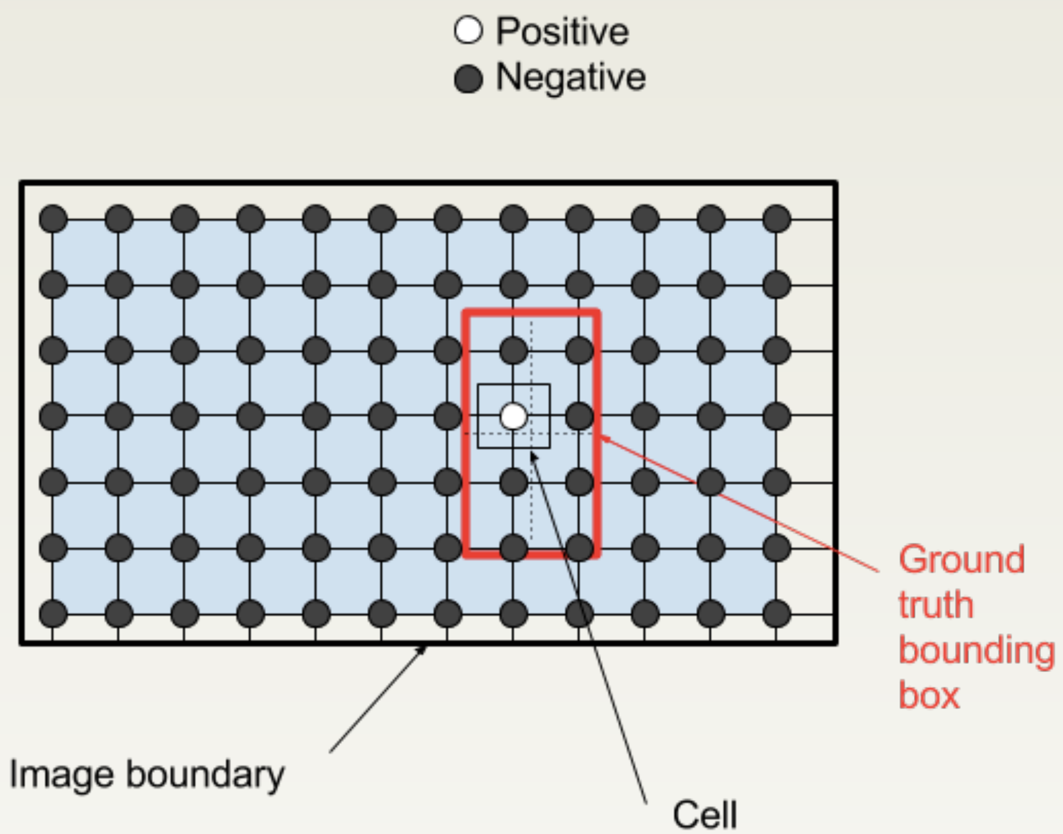
A look at computer vision

T. Lin et al, arXiv:1405.0312

- 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 x M per image)
- Assign object score/bounding box to anchor point
- Object can be found multiple times

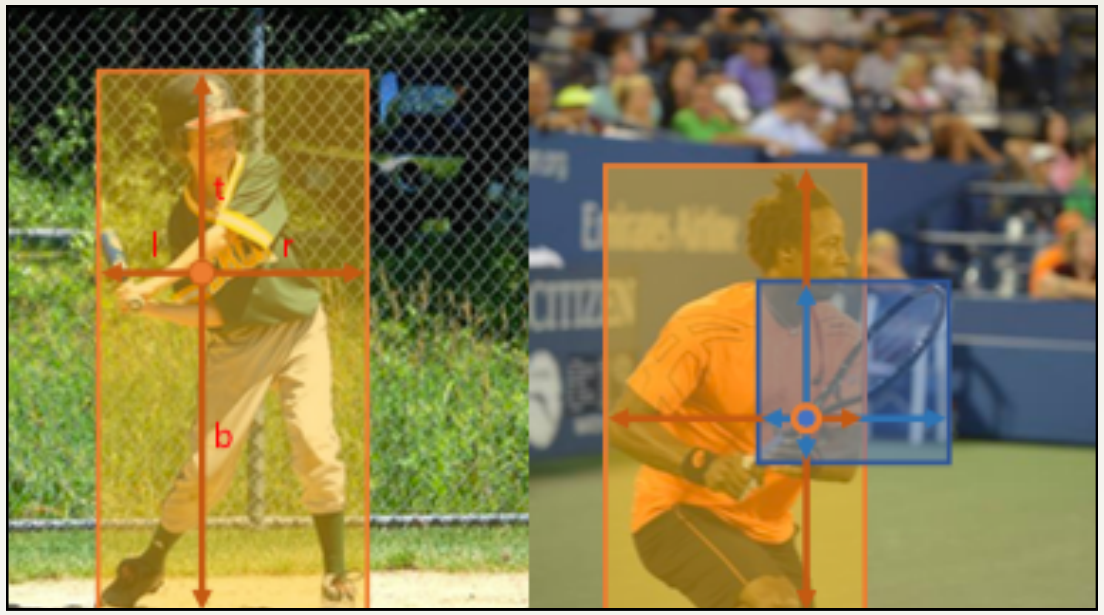
- Anchor points grow with with N^(dim), make implicit assumptions on object size

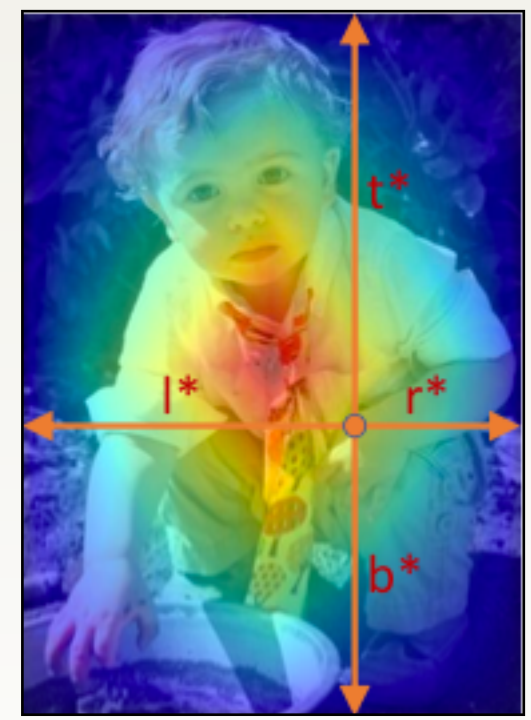
- *Not suitable for reconstruction based on high-dimensional detector signals*

Figures: towardsdatascience.com

Key point methods

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





Problem: identify the key points

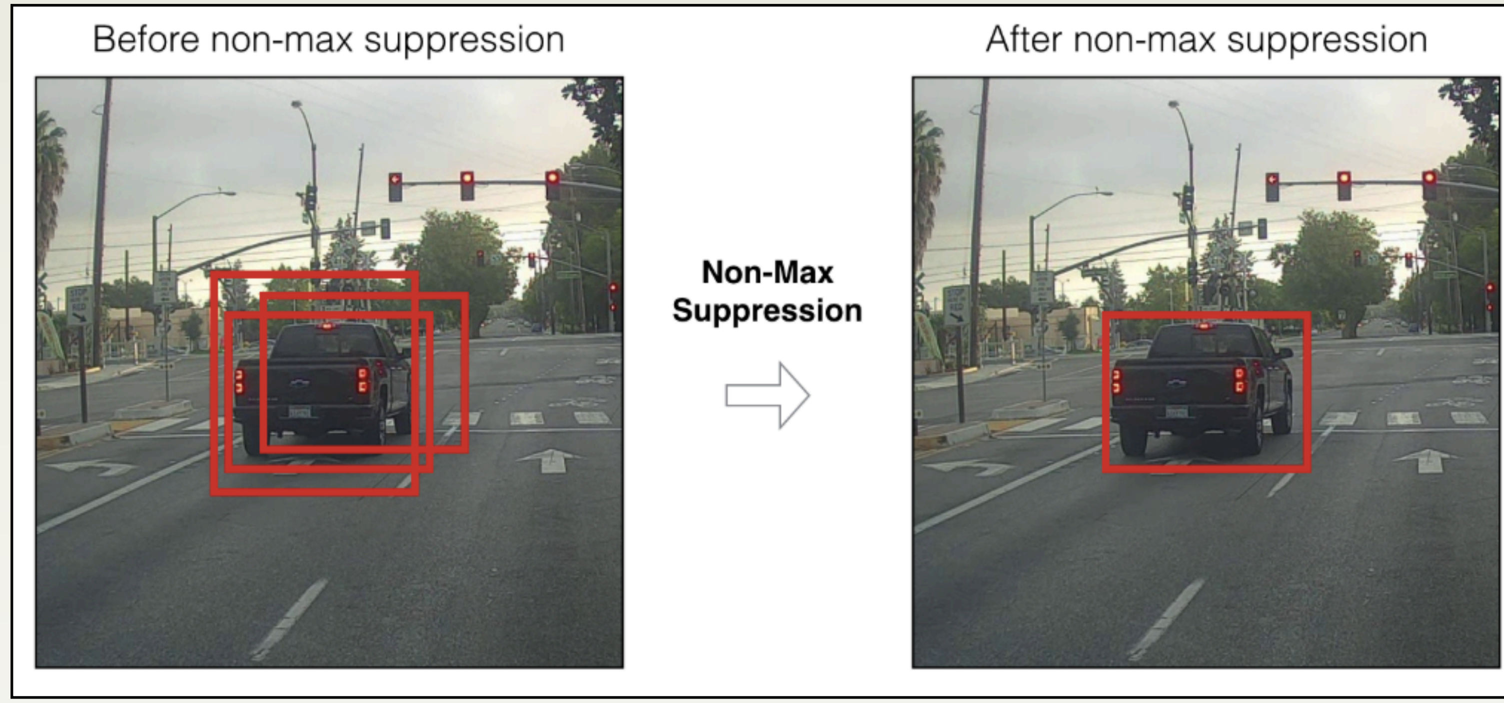
- Also predict 'center-score'
- Select highest score in the area as key point
  - ***Seed identification!***
  - Heavily relies on objects to have a center: problematic for a particle

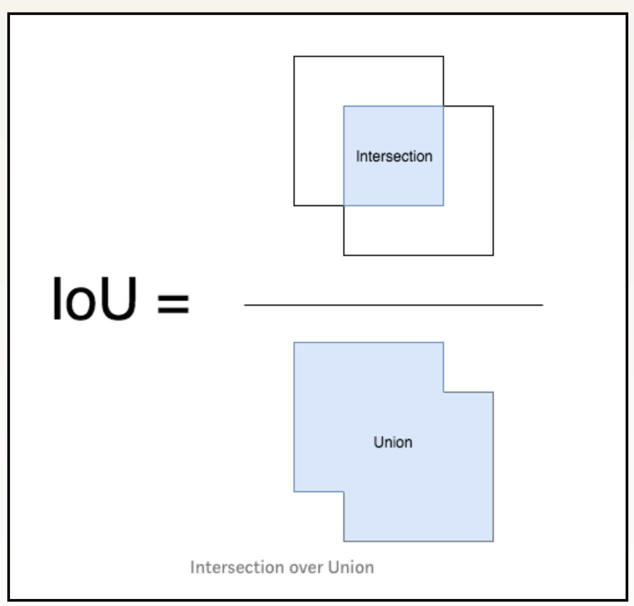
- Remaining ambiguities still need to be resolved

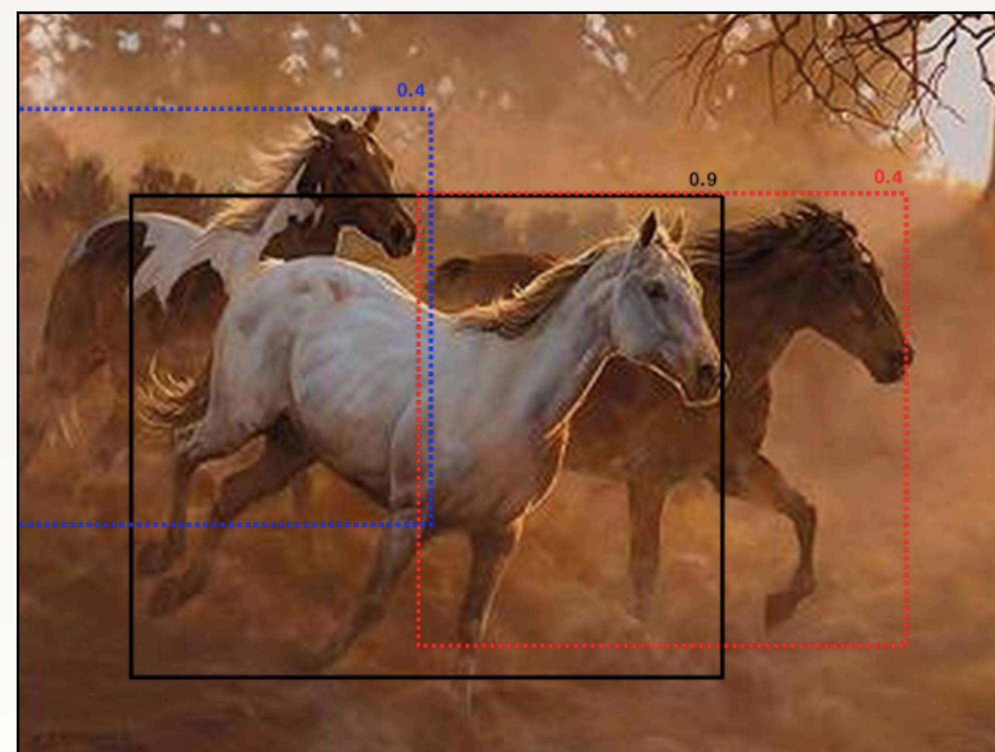
N. Wang et al, arXiv:1904.01355
X. Zhou et al, arXiv:1904.07850

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*







N. Bodla et al, arXiv:1704.04503
Figures: towardsdatascience.com

Segmentation and Clustering

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

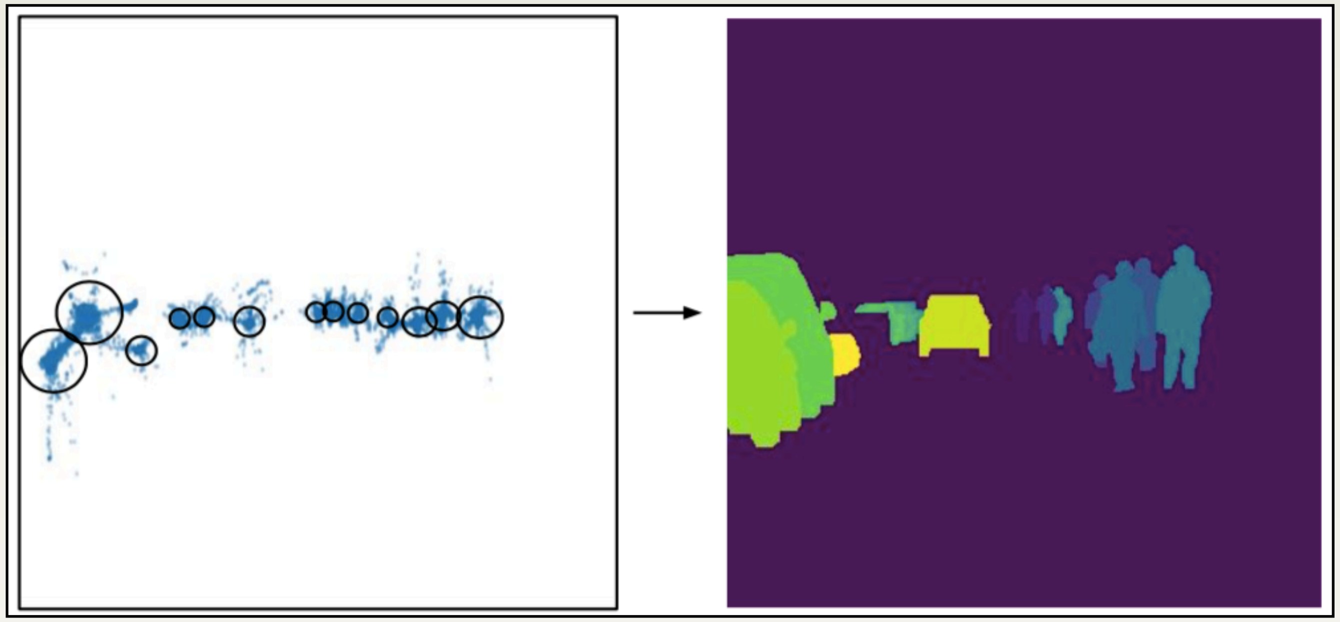
$$\phi_k(e_i) = \exp\left(-\frac{\|e_i - C_k\|^2}{2\sigma_k^2}\right)$$

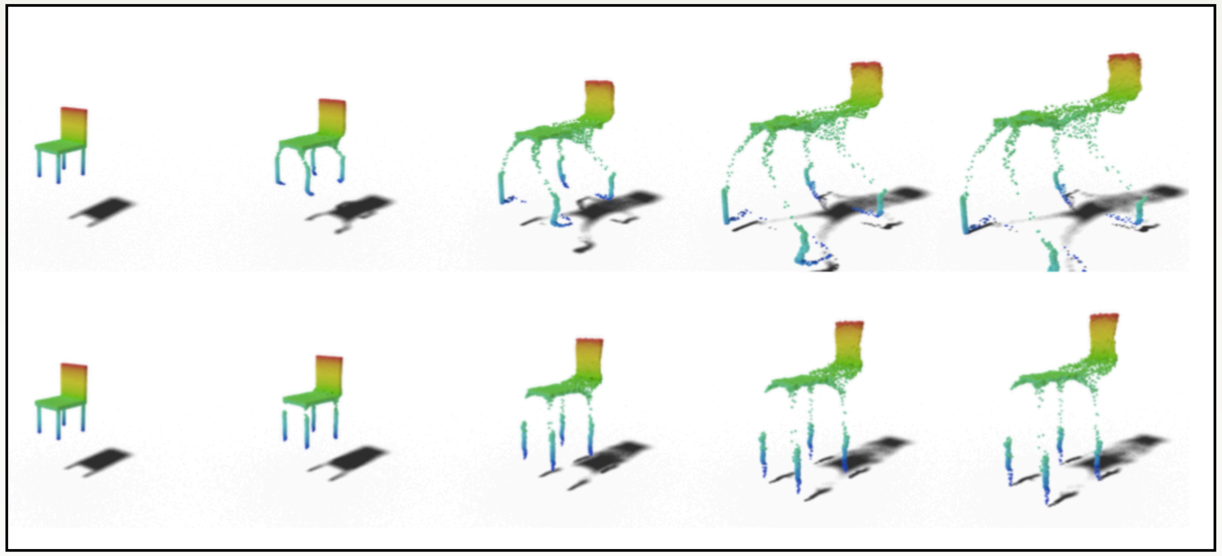
- Assign seed score

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

- Collect (from highest seeds score) around the seeds

- *'Only' performs segmentation*
- *Heavily relies on the center of an object*
  - *Problematic concept for particles*

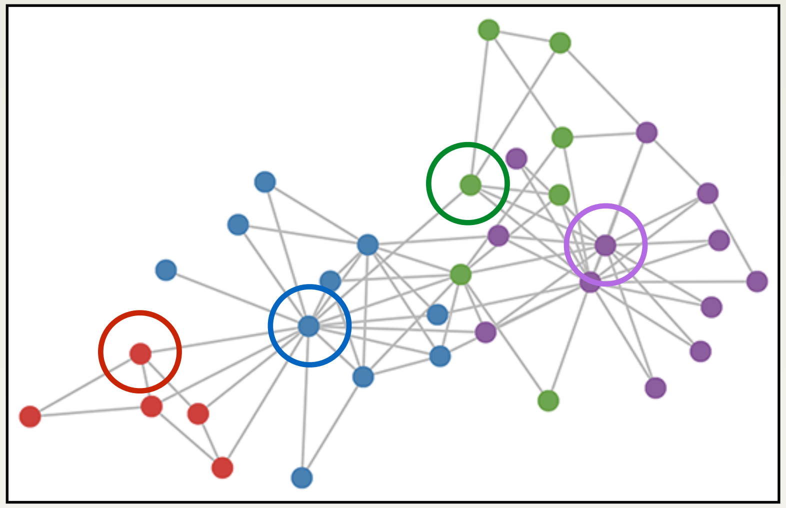




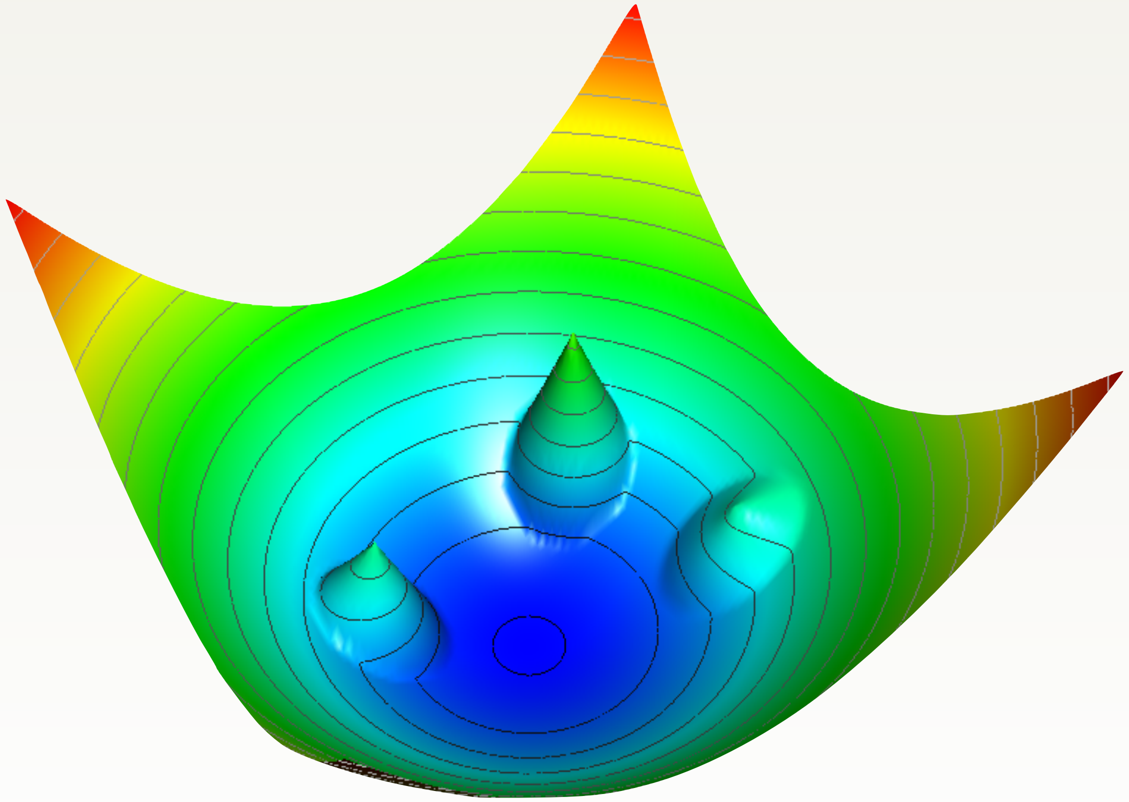
D. Neven et al, arXiv:1906.11109
B. Zhang, P. Wonka, arXiv:1912.00145

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 β (linked to a “charge” q)
  - Cluster coordinates x

$$q_i = \operatorname{arctanh}^2 \beta_i + q_{\min}$$

- Define charge, attractive and repulsive potential

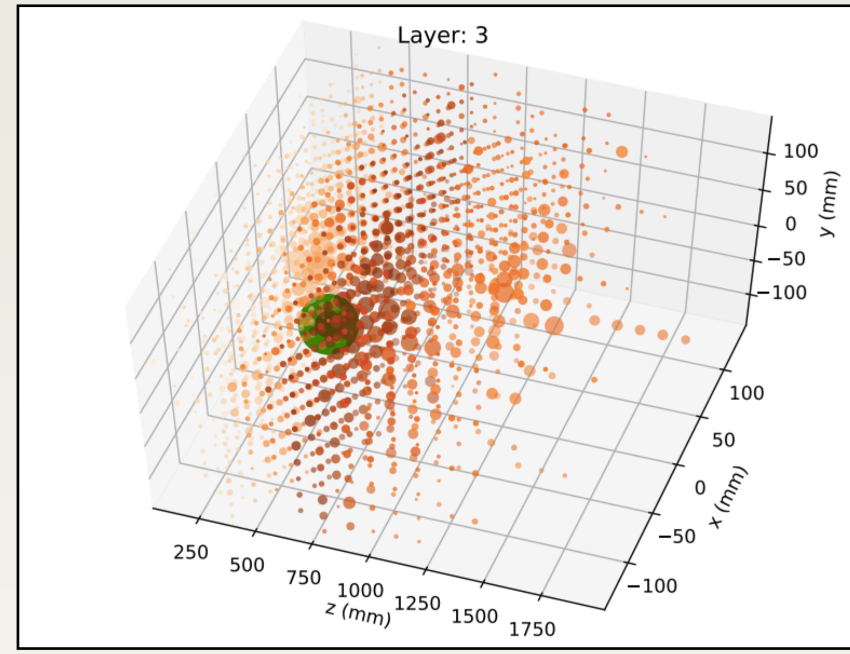
JK, arxiv:2020.03605

Condensate and predict

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

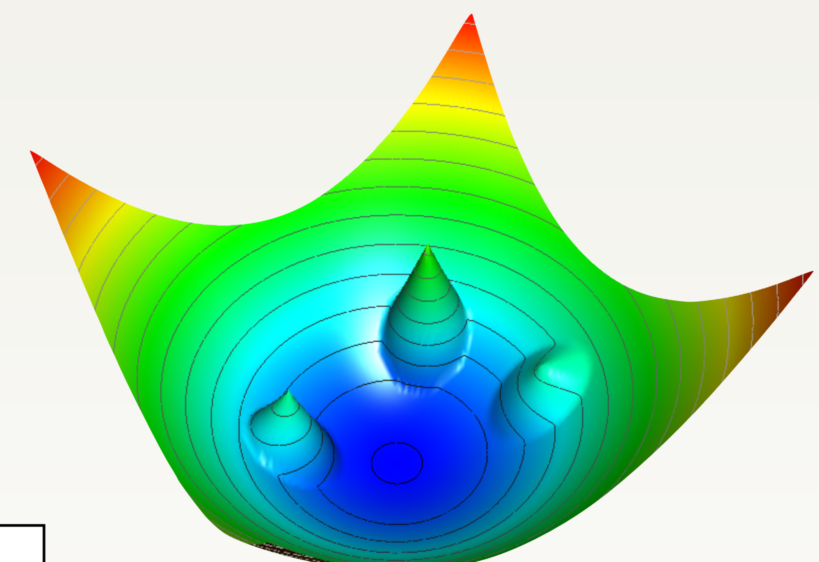
$$\hat{V}_k(x) = \max(0, 1 - ||x - x_\alpha||) q_{\alpha k}.$$

Maximum charge vertex for object k



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

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



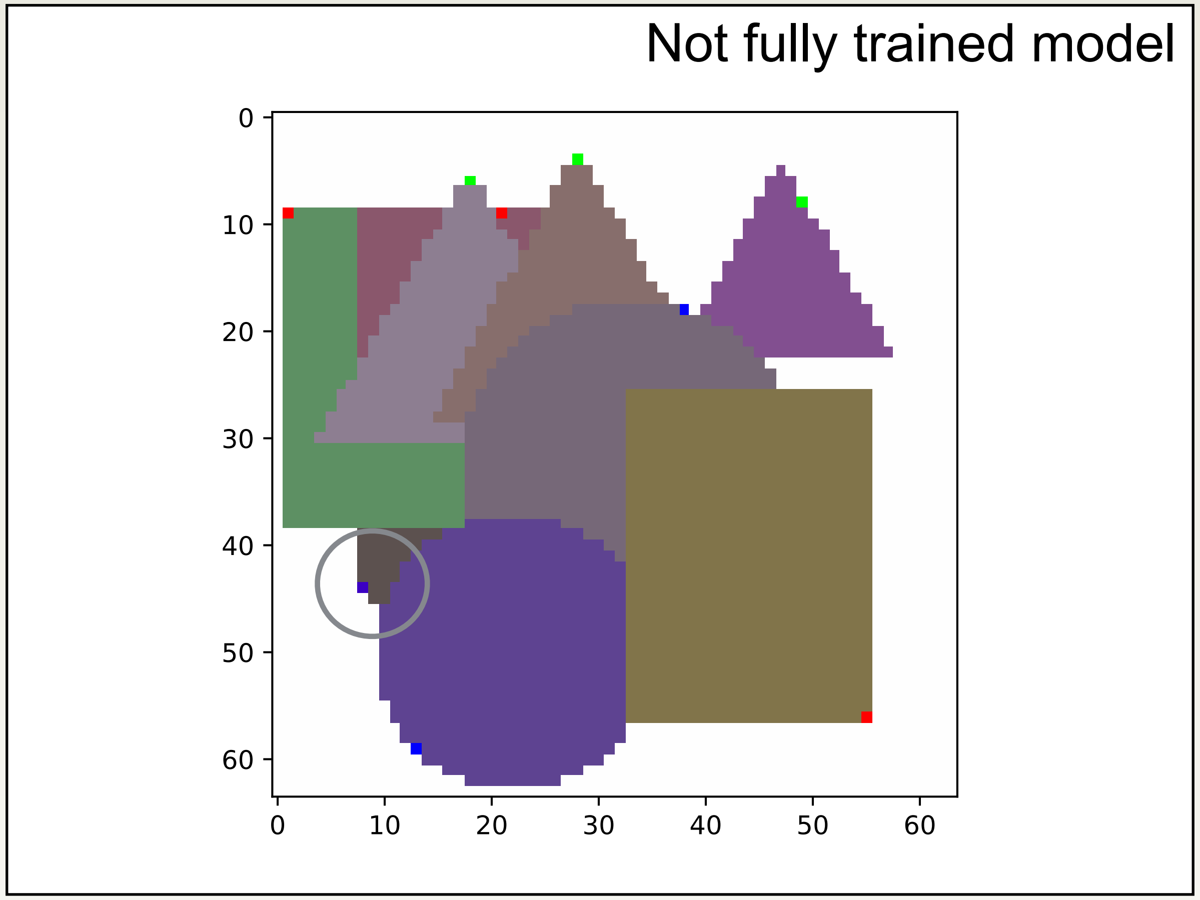
- Also weight object property loss with β

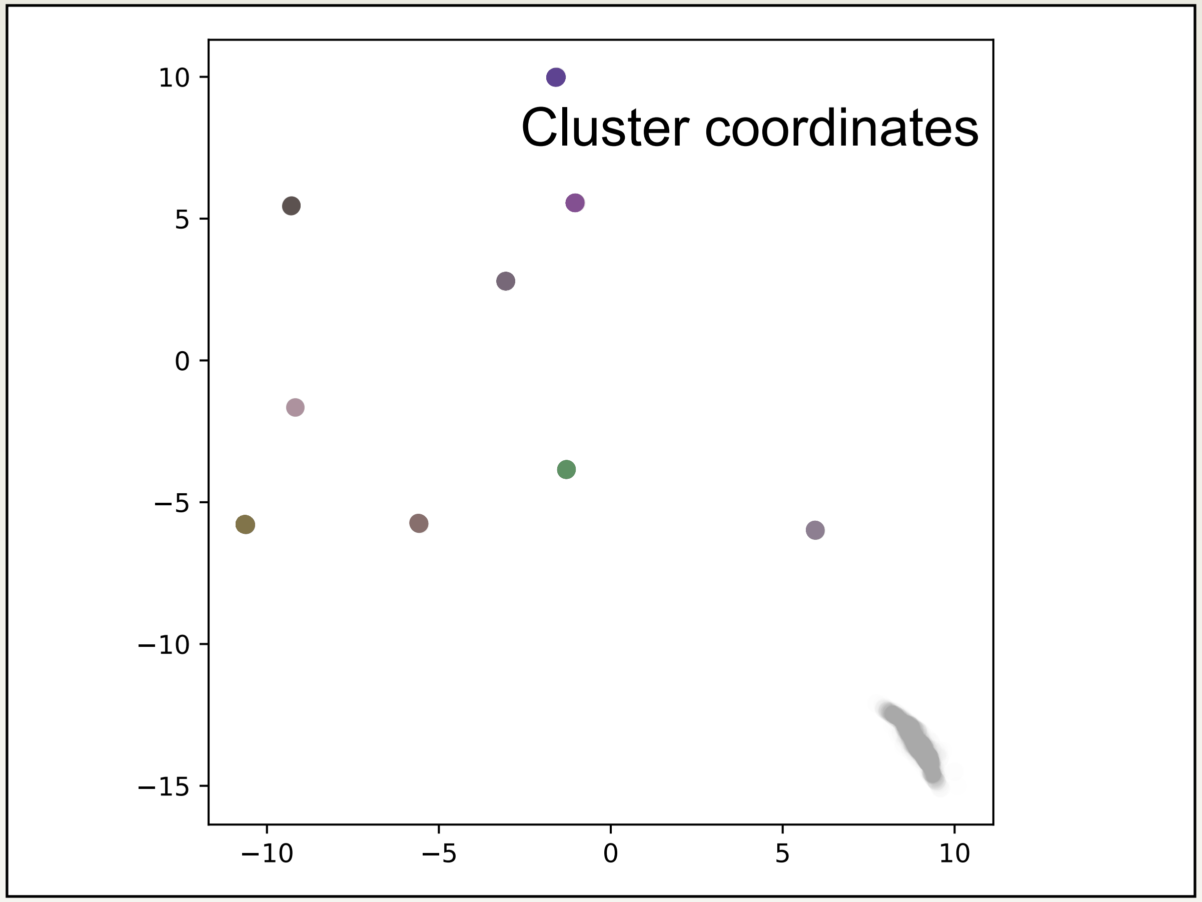
$$L_p = \frac{1}{\sum_{i=0}^N (1 - n_i) \operatorname{arctanh}^2 \beta_i} \sum_{i=0}^N L(t_i, p_i)(1 - n_i) \operatorname{arctanh}^2 \beta_i$$

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

*NB: Removes saddle point for large N

Image data





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

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

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

- 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

Particle Flow

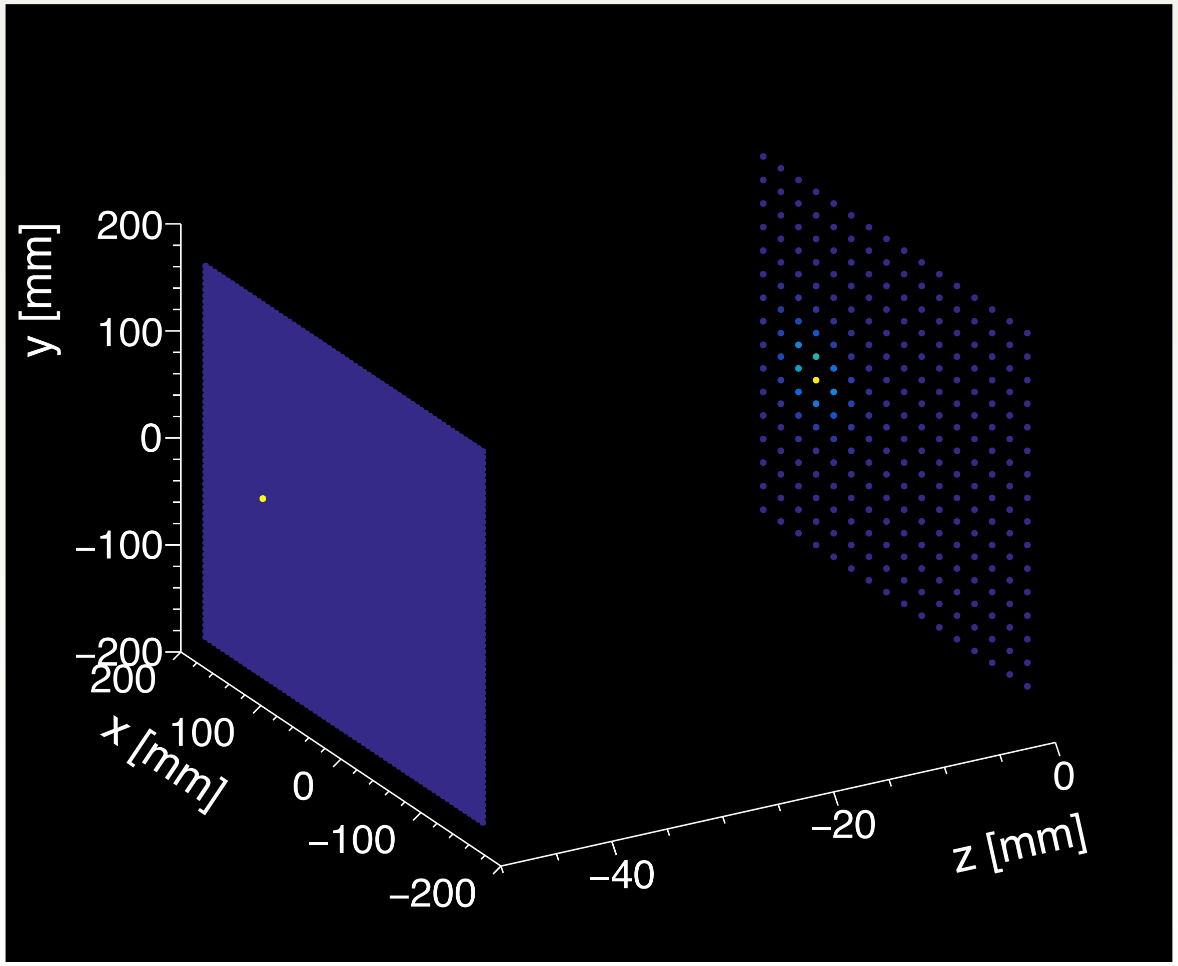
- Simplified detector
  - "Flat" in x,y: not curved
  - ECal: 16 x 16 cells, each 22 x 22 mm² x 26 cm lead tungstate (CMS ECal)
  - No magnetic field
  - "Tracker": 300μm silicon 5.5 x 5.5 mm² sensors, placed 5 cm in front of calorimeter
    - Assign Gaussian smeared track momentum to highest energy 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

* for evaluation up to 15
for cluster calibration 1
for jets up to 25
(later)

Geometry



Baseline: CMS PFAlgo

- Taken only the EM part, stripped of Brem recovery etc

All numbers from PF paper [arxiv:1706.04965]

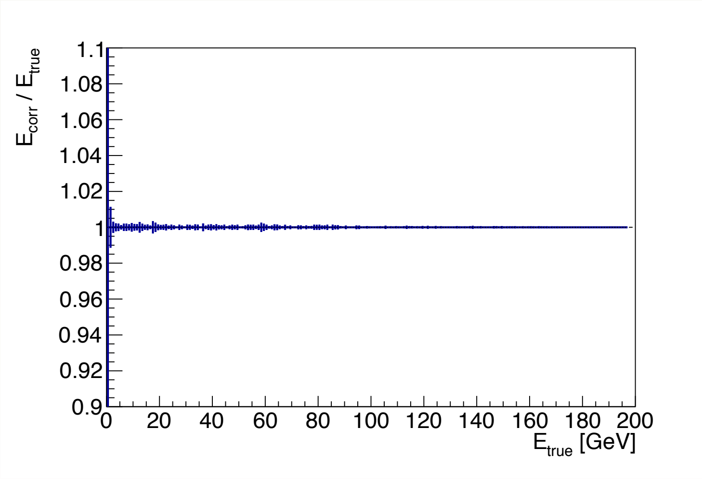
- Calo seeding
  - Discard depositions below 80 MeV
  - Find deposits over 230 MeV
  - Set as seed if all adjacent 8 deposits have lower energy
  - Additional track driven seeds
    - Set any calorimeter cell with a track in the cell area as seed

$$f_{ji} = \frac{A_i e^{-(\vec{c}_j - \vec{\mu}_i)^2/(2\sigma^2)}}{\sum_{k=1}^{N} A_k e^{-(\vec{c}_j - \vec{\mu}_k)^2/(2\sigma^2)}}.$$

$$A_i = \sum_{j=1}^{M} f_{ji} E_j, \vec{\mu}_i = \sum_{j=1}^{M} f_{ji} E_j \vec{c}_j \frac{1}{\sum_k E_k}$$

- No topo clustering!
- Calo clustering for each seed
  - Discard depositions below 80 MeV
  - Iterative likelihood (σ=15mm) maximisation
    - Energy
    - Position

- Calo cluster calibration
  - Use a sample of 100k single particles and calibrate in bins of 1 GeV for perfect response (by construction)



PFAlgo : Linking and truth matching

- Linking **
  - For each **track** find the cluster with the closest distance
  - Hard cut-off: 22 mm (cell size) **
  - Can lead to multiple tracks per cluster

Tracks here are never fakes

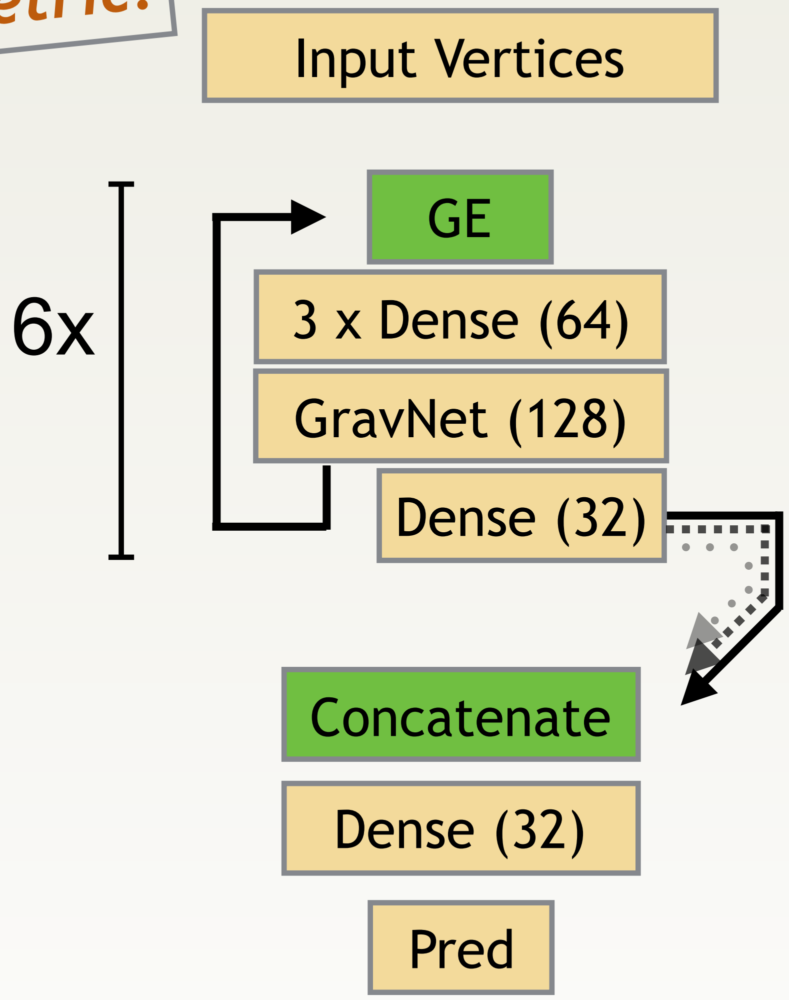
- Built PF candidates from calo clusters
  - If no track linked: use calo cluster energy and position
  - Track linked
    - Momentum compatible within 1 sigma (Δcalo $\oplus$ Δtrack):
      Momentum: resolution weighted mean
      Position: 67% track position, 33% cluster position
    - Not compatible and calo excess > 500 MeV:
      assign track momentum/position,
      generate **new cluster** from excess at calo cluster position
    - If another track linked to remaining new cluster: repeat,
      otherwise generate photon **

** different from base PFAlgo,
because it led to better results

Object Condensation approach

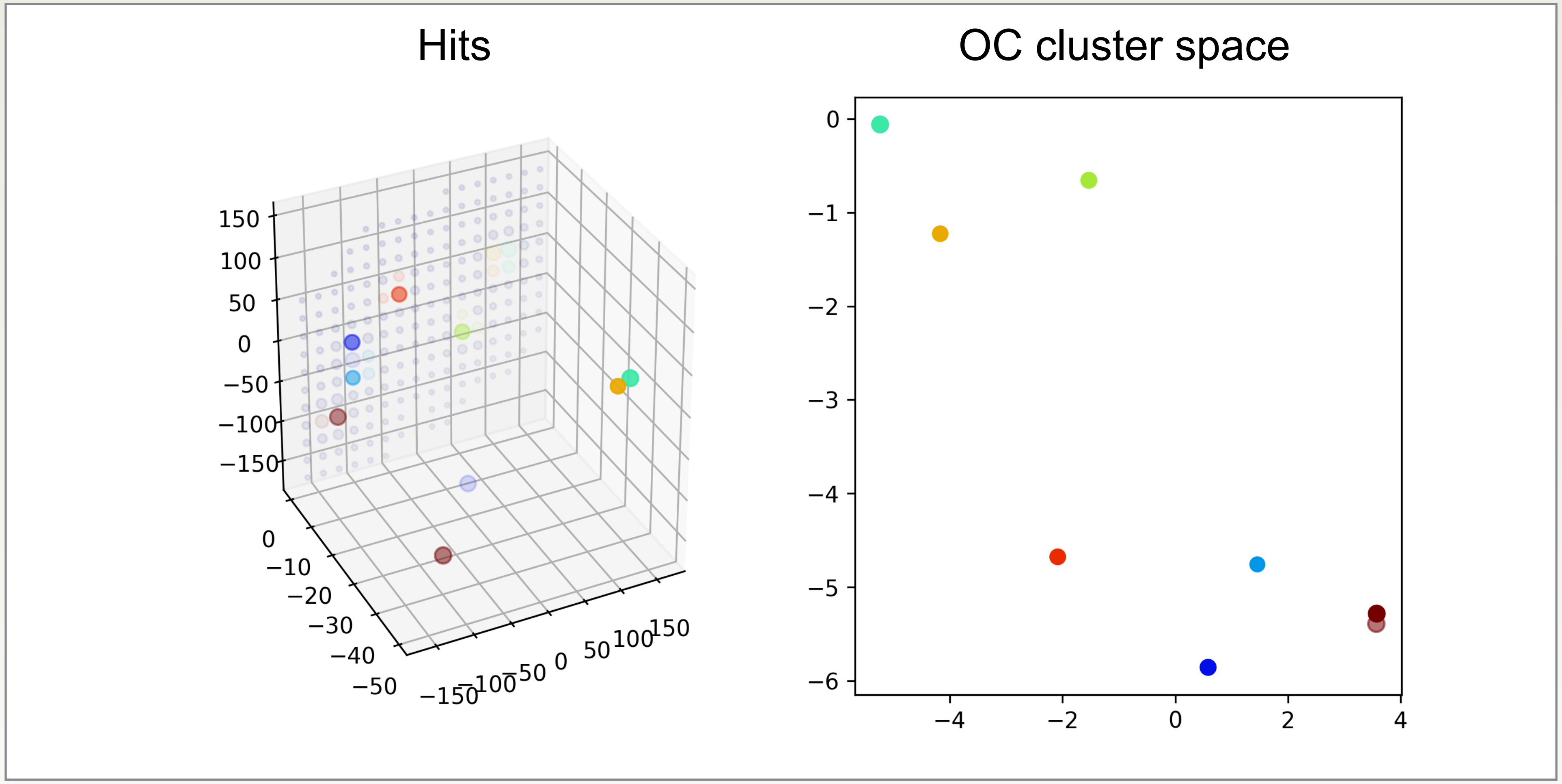
- Truth:
  - Assign particle properties to vertex with highest fraction
- Select 200 highest energy deposits/tracks
- Use rather standard GravNet [1]
  - 10 neighbours, 4 space dimensions, 64 features to be exchanged
  - Nothing ragged
- Predict:
  - OC Clustering space
  - OC Confidence beta
  - Position (offset w.r.t. sensor position)
  - Energy = Momentum (correction factor w.r.t. sensor energy)
- Sample: 1.7M events, 1-9 particles per event
  - Trained for 110 epochs, learning rate decrease after 20 epochs
- Set minimum OC clustering charge to 0.1 (less segmentation focus)

Now in torch_geometric!



[1] S.Qasim, JK, et al, 1902.07987, EPJC (2019)
https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html#torch_geometric.nn.conv.GravNetConv

Segmentation / Postprocessing



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

Truth matching

- For object condensation: match remaining with ground truth on cell-by-cell/track-by-track basis

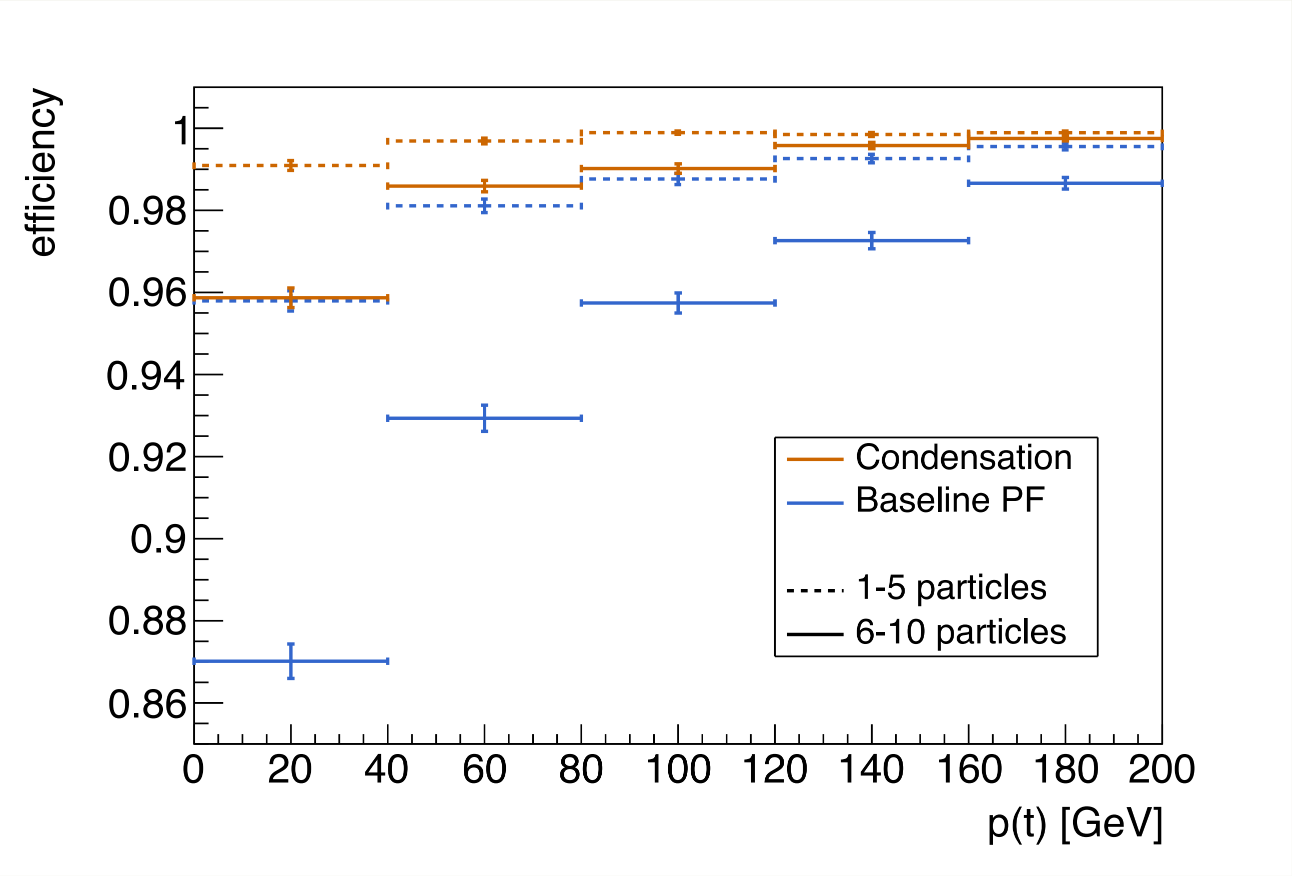
- For PF:
  - If particle has track associated match via track truth, match directly, mask candidate and truth
  - For remaining truth try to find a reconstructed candidate
    - Within distance of 3 cells (3*22mm)
    - Within | Er / Et - 1 | < 0.9
    - Match reco candidate that has smallest d

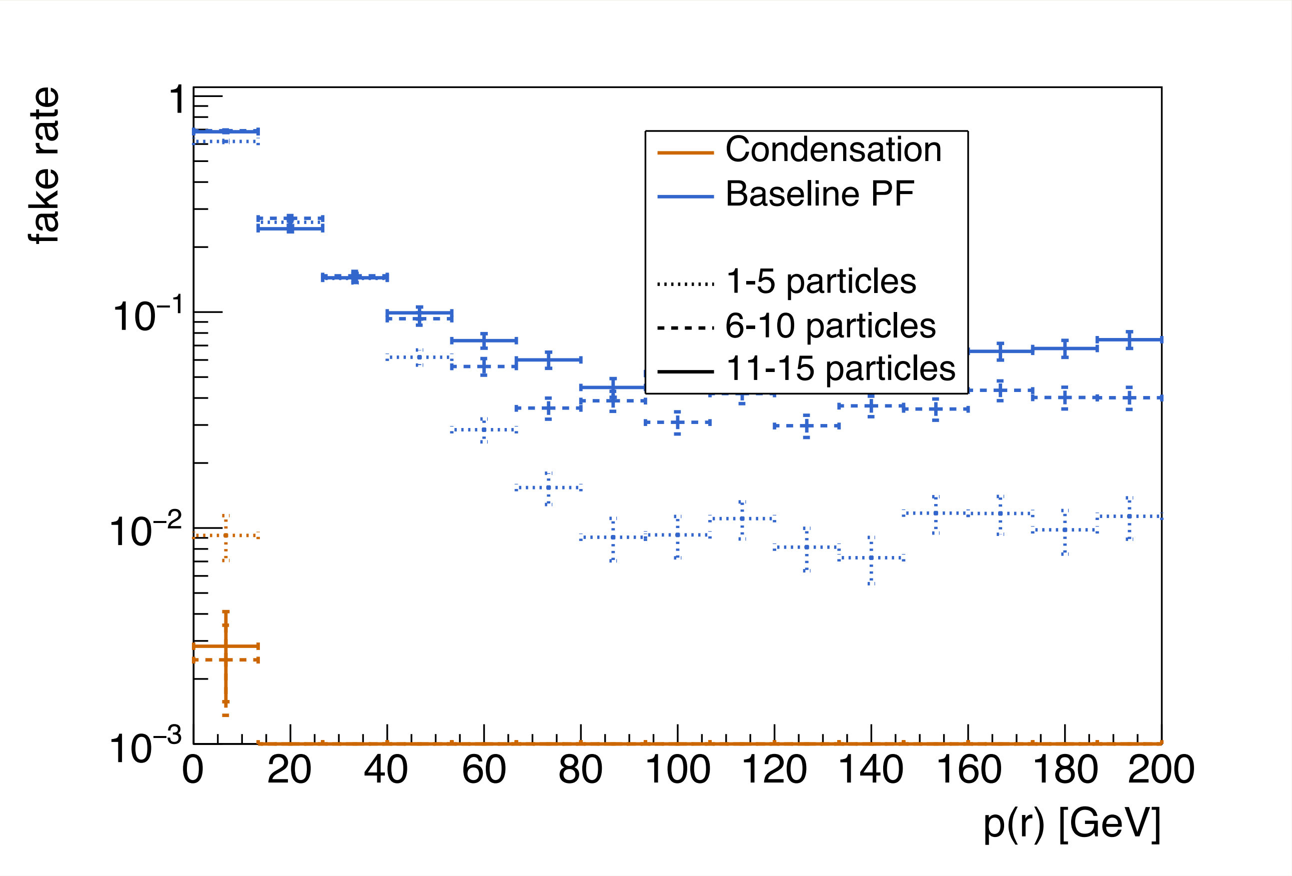
$$d = \Delta x^y + \Delta y^2 + \frac{22}{0.05}\left(\frac{E_r}{E_t} - 1\right)^2$$

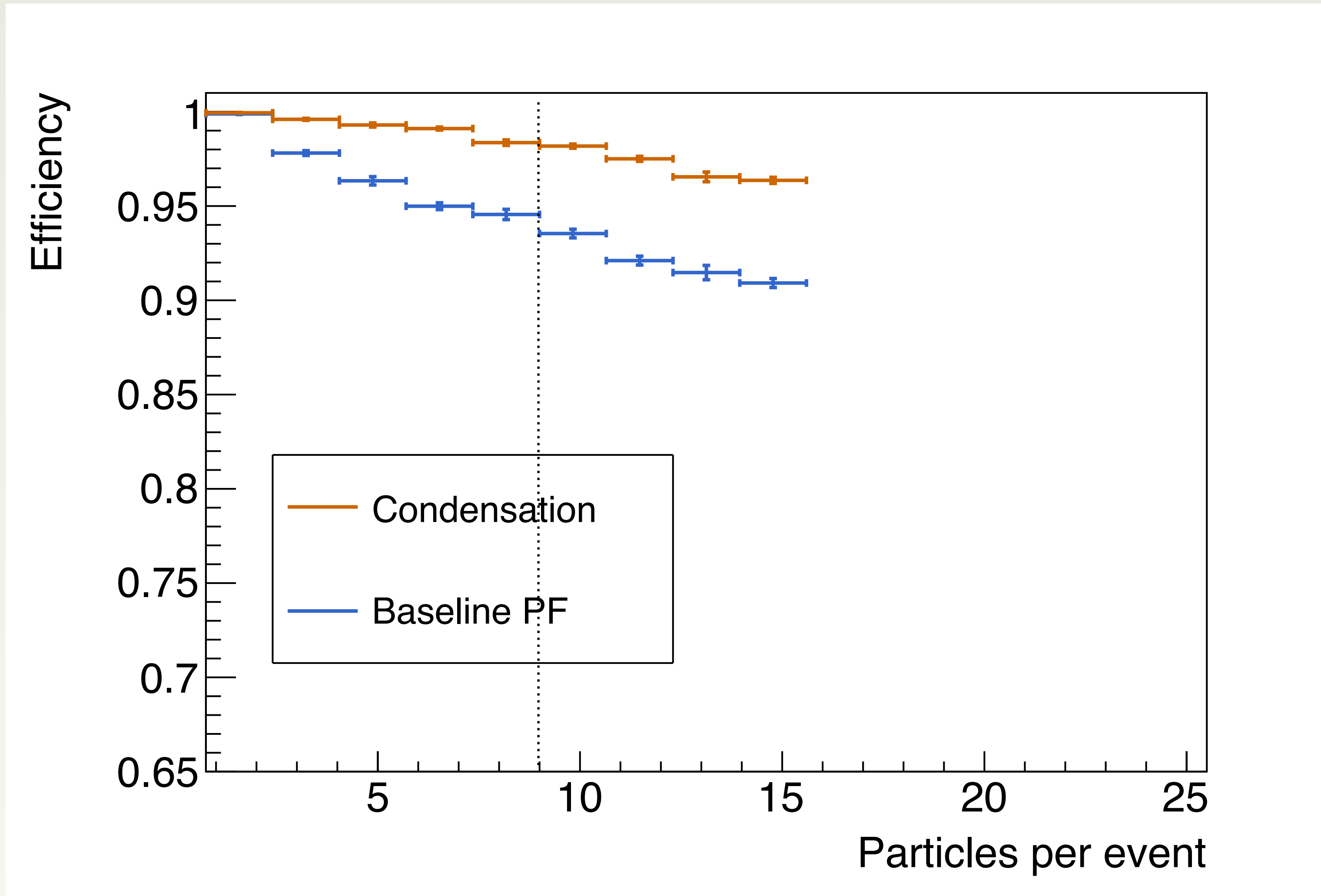
- For all unmatched remaining truth particles record properties (efficiency)
- Same for reconstructed candidates (fake rate)

- Test sample 1: same configuration as training sample but up to 15 particles per event

Efficiency / Fakes



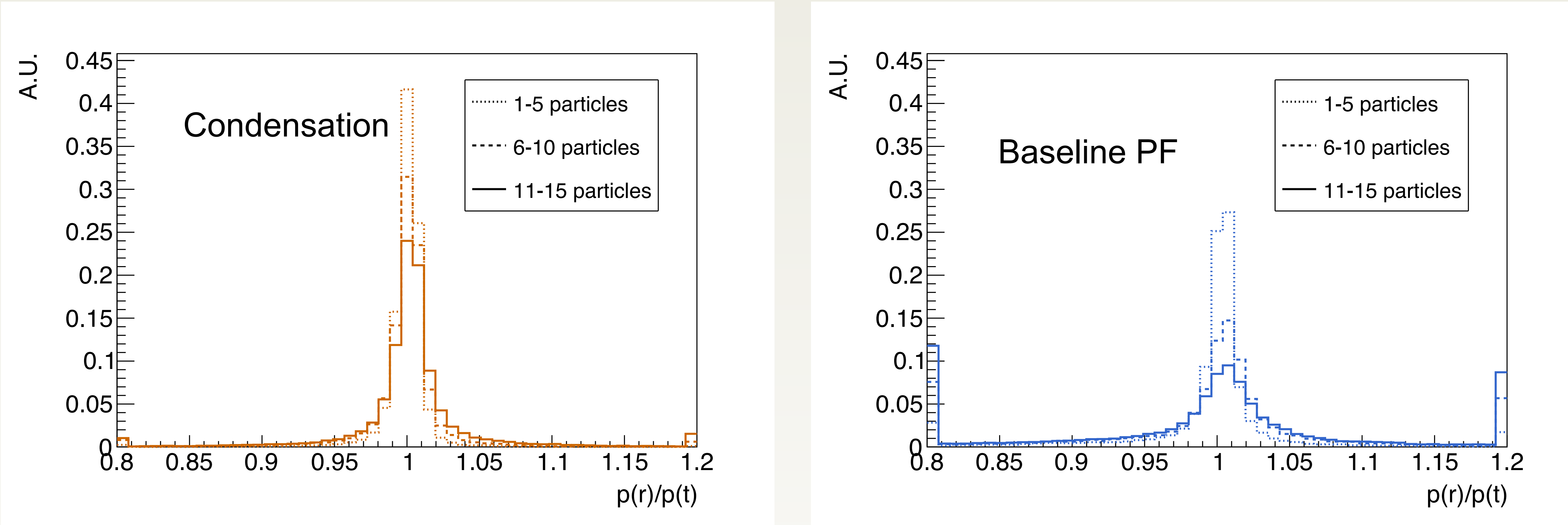




- Efficiency decreases with number of particles per event

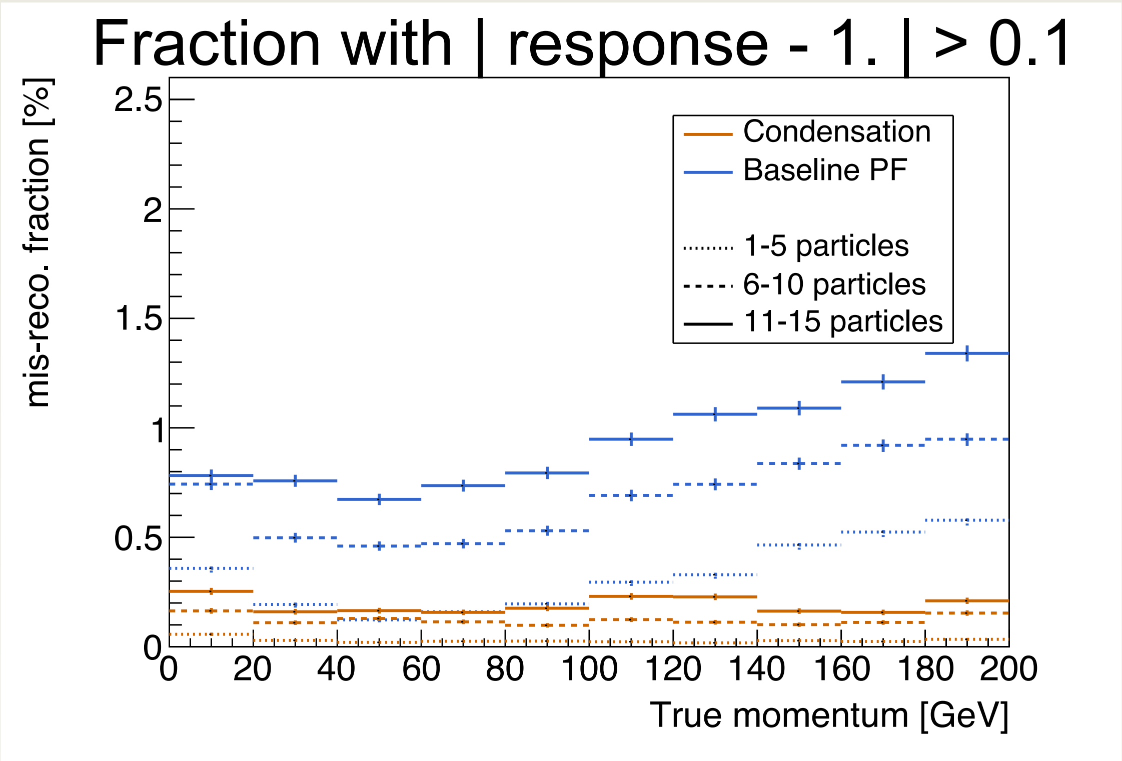
- **The extrapolation beyond the training conditions (9 particles) works perfectly!**
    - OC loss and GravNet are inherently *local*, no ‘number of particles per event’ was learnt

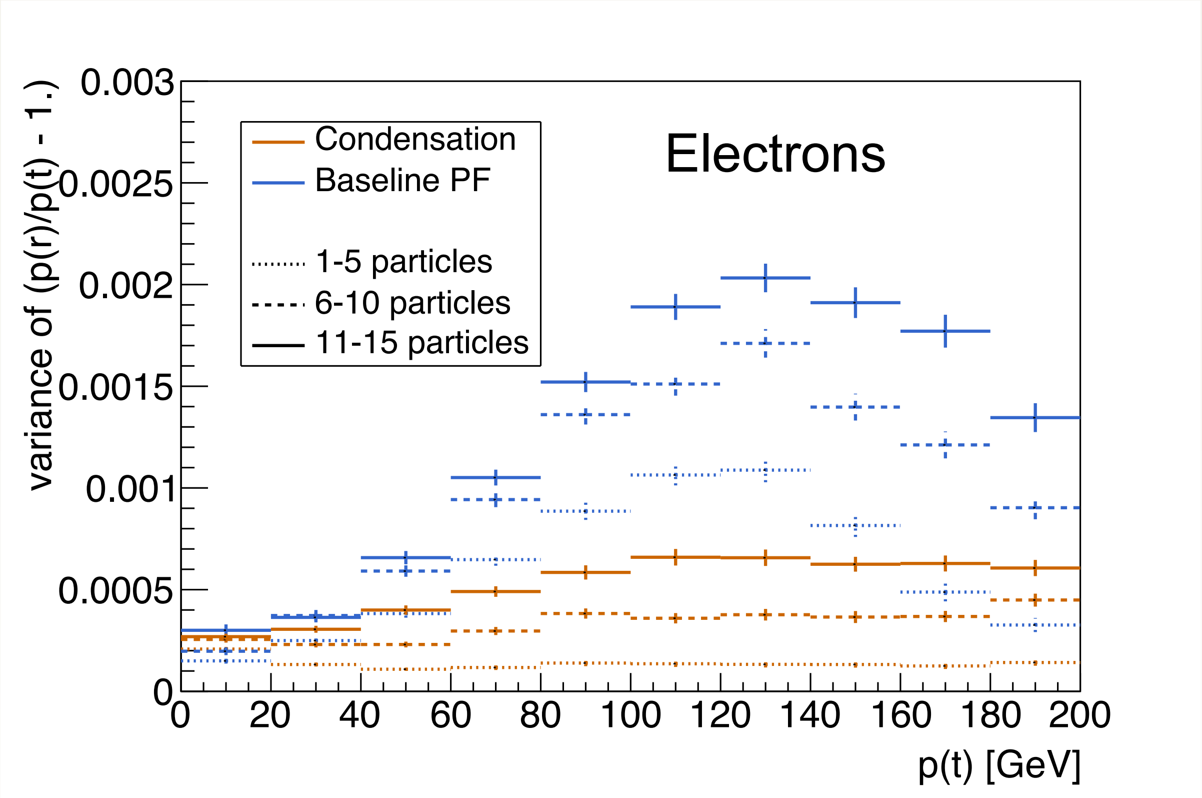
Response

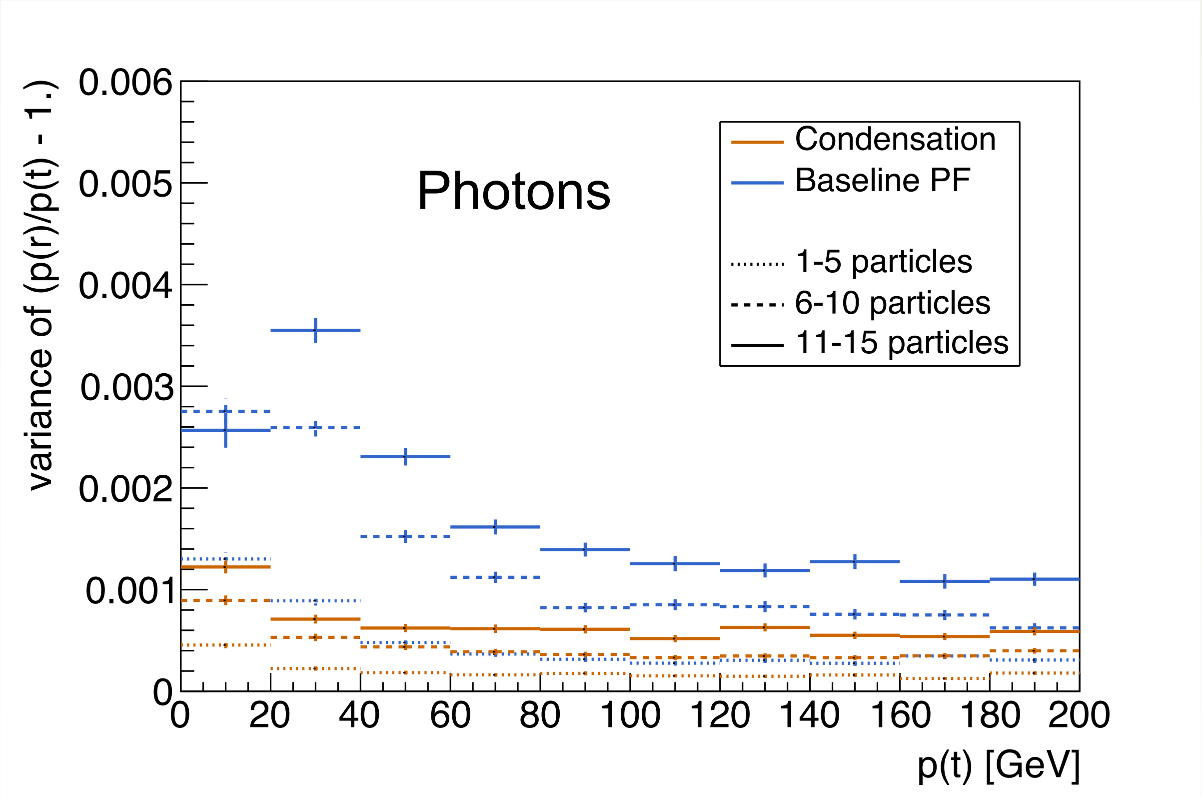


- For variance (next plots) just take into account | response - 1. | < 0.1

Response variance

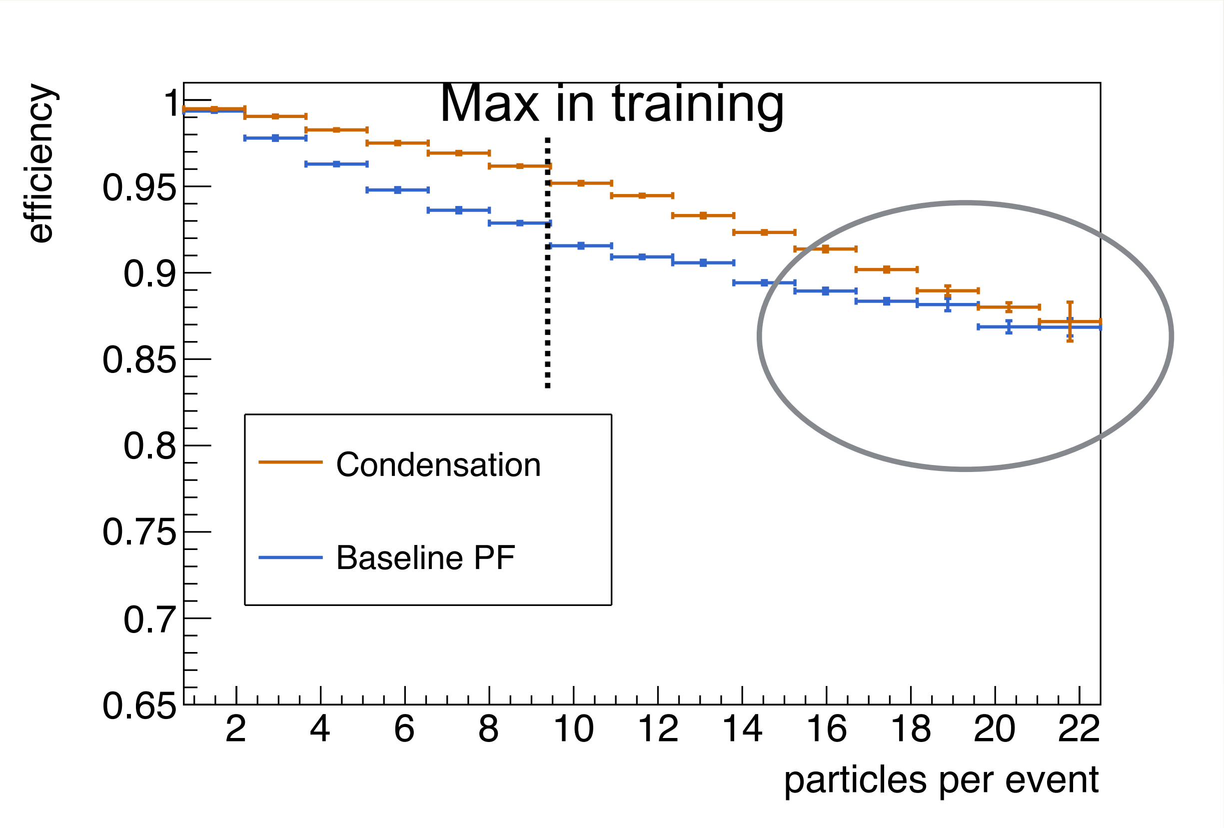






"Jet" properties

- Generate jet-like sample
  - Create particles (50/50 photons/electron) using exponentially falling energy spectrum (exp(-ln(300) * E/GeV)
  - Randomly pick N particles, with N being Poisson distributed around an average of M (M being drawn from a uniform distribution with M <= 15)
  - Gives jets proxies between a few GeV up to about 300 GeV, with a poisson distributed number of particles for fixed energy

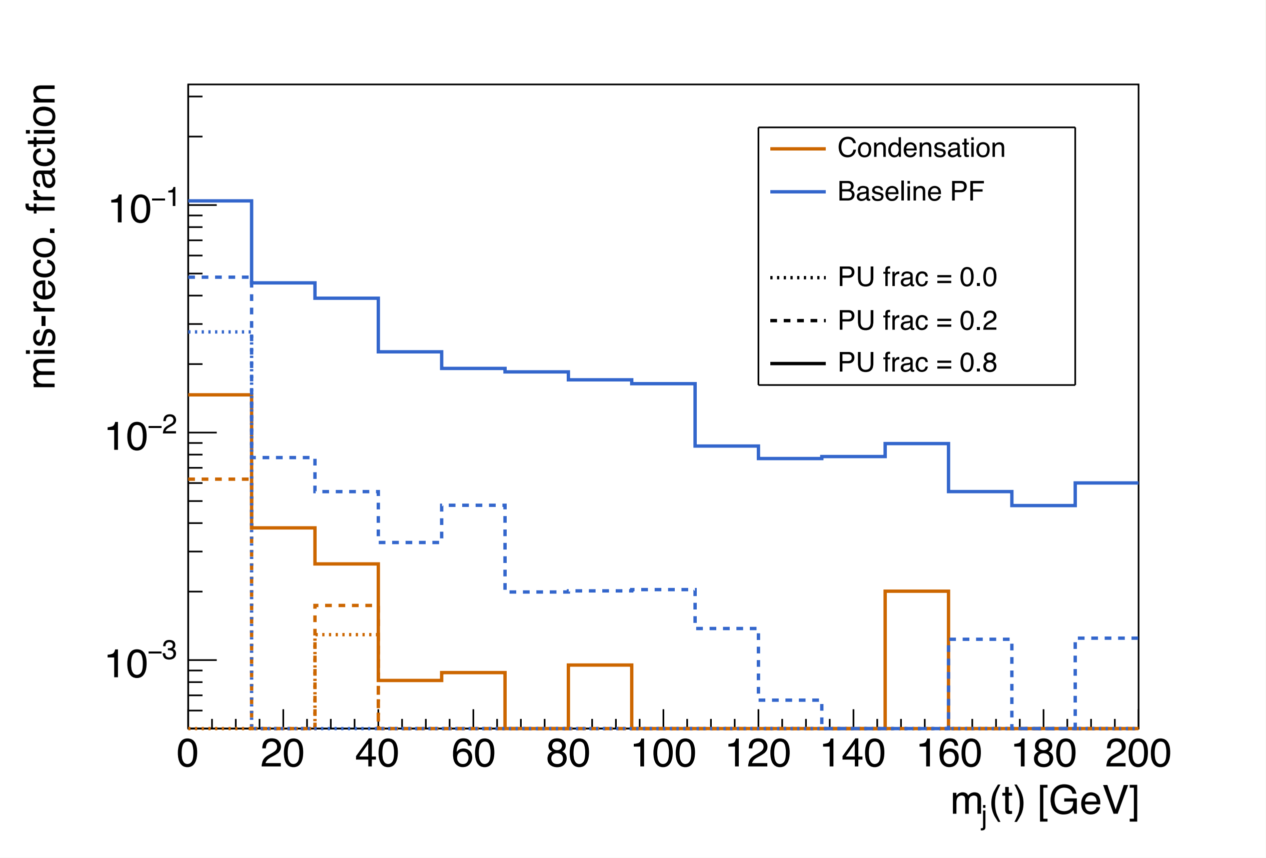


For PF, truth matching starts to fail -> will look at "jet" properties so doesn't matter

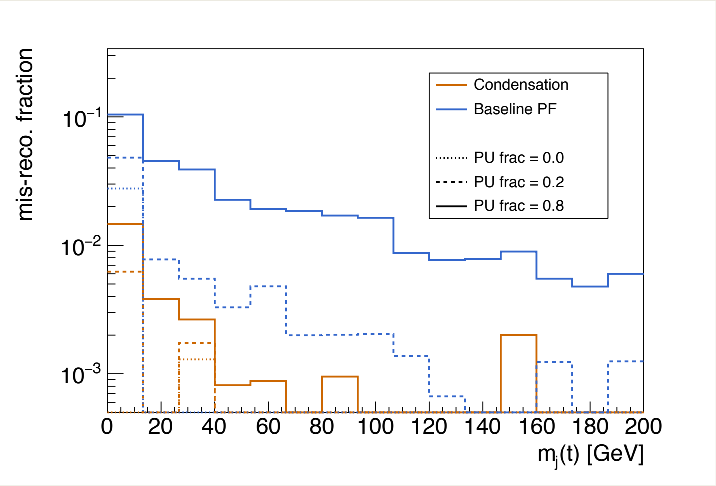
- Excellent behaviour for significantly larger particle densities than seen in the training!
  - Also true for other distributions

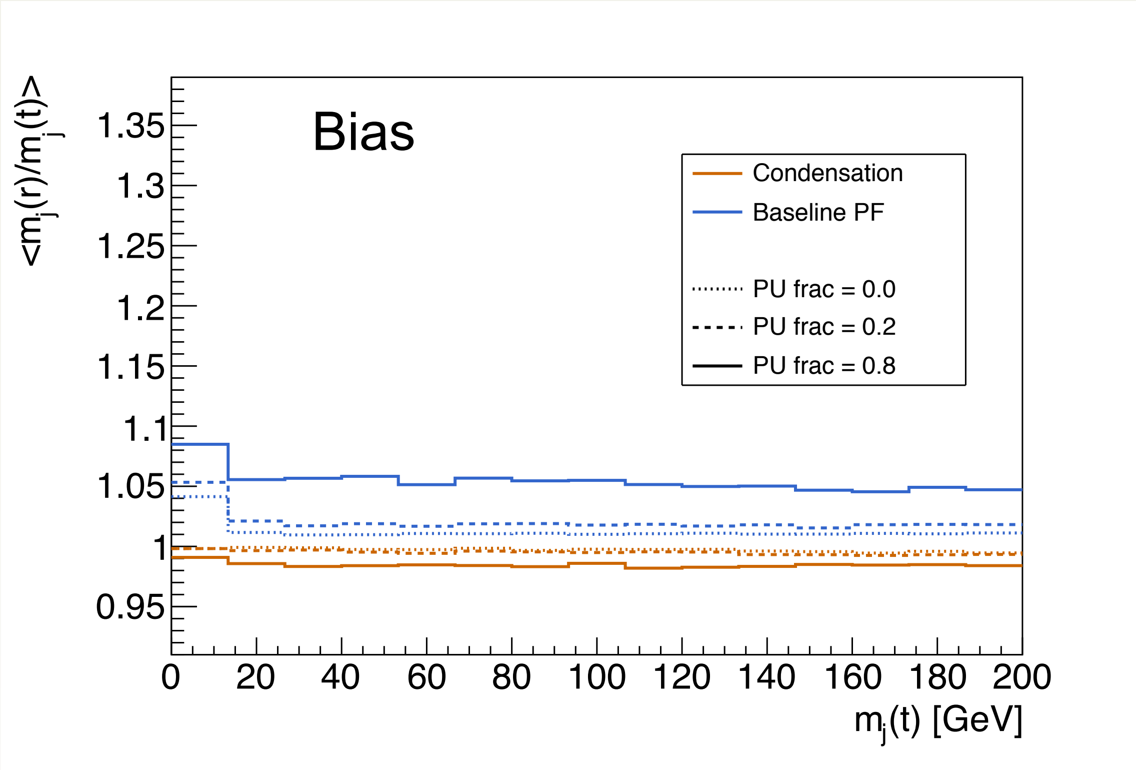
Jet mass response

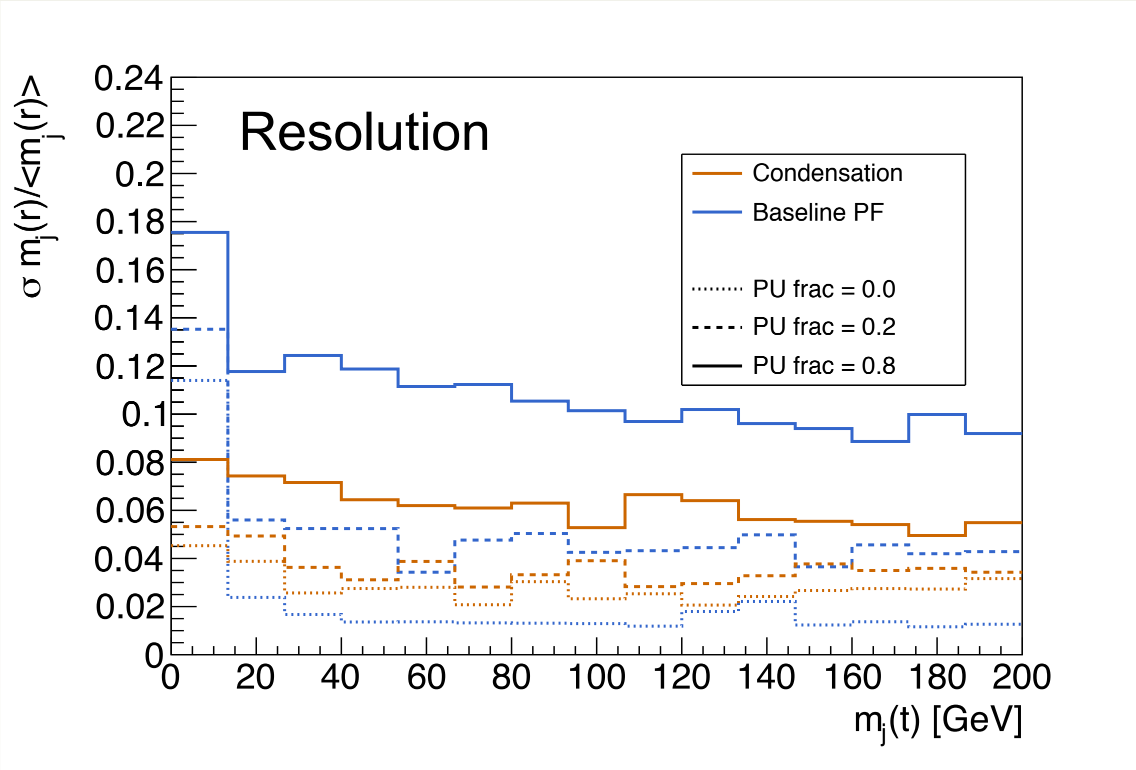
- Do not consider misreconstructed events with | response - 1. | > 0.5
- Simulate pileup subtraction (CHS) by removing a fraction of electrons (truth matching works because of track) = “pileup fraction"



Jet mass resolution







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

Summary

- **Object condensation allows to predict properties of an unknown number of object in image, point clouds, graphs, … with *one-stage* detectors in detector data**
  - Removes redundancies and dependencies

- No overhead at inference time

- Particle flow application very promising compared to classic approaches, even in almost ideal environment (most convenient for the classic approach)

- **Excellent extrapolation beyond the training conditions for GravNet + object condensation model**

- **More details: JK, arxiv:2020.03605**