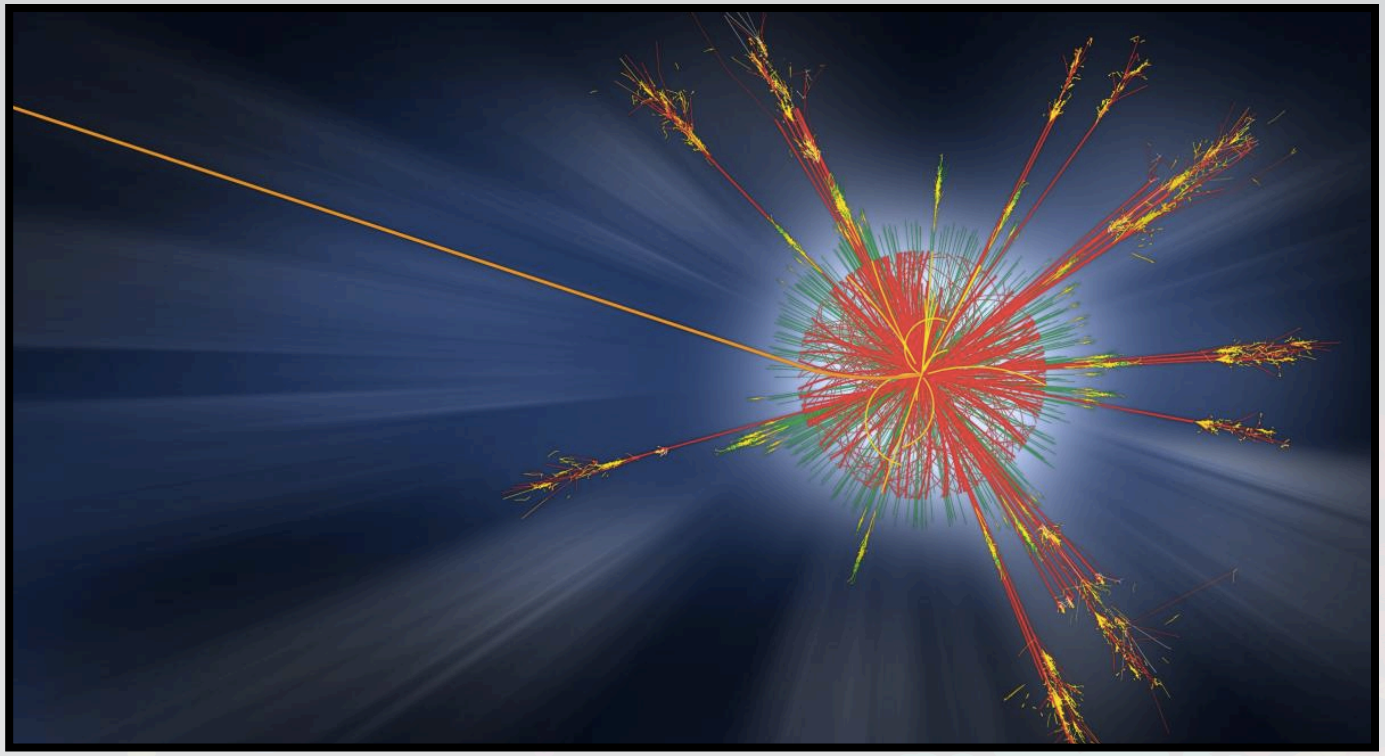
Object Condensation

one-stage grid-free multi-object reconstruction in physics detectors, graph, and image data

Jan Kieseler

23.10.2020

Reconstruction



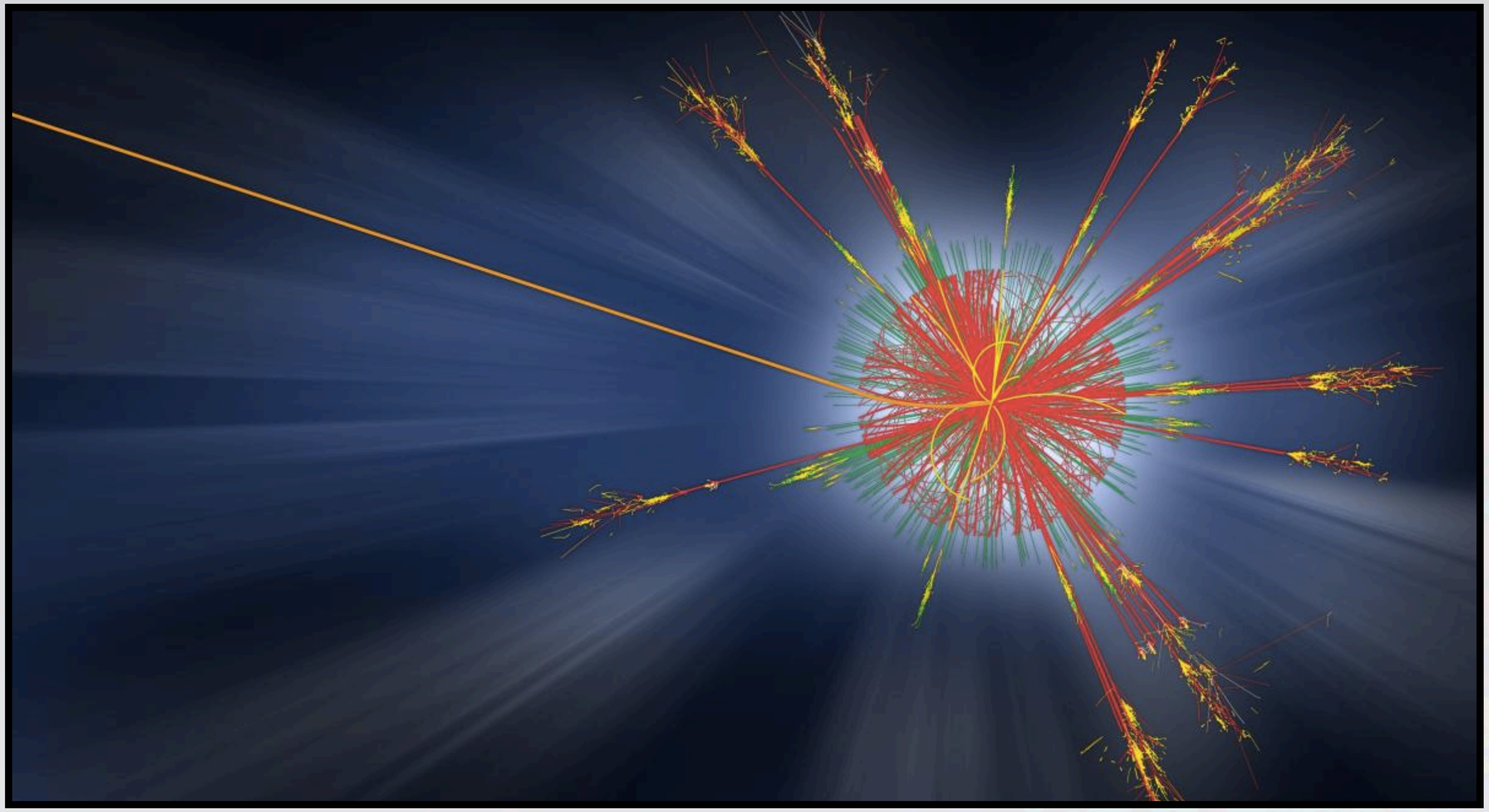
1 muon
3 few jets
....

- What we actually want: particle ID, momentum, position
- Standard chain has many redundancies
  - Seeding (pattern recognition)
  - Clustering (pattern recognition)
  - Software compensation (pattern recognition)
  - ID (pattern recognition)
  - PFlow (pattern recognition)

- Many specialised teams
- Binds a lot of person power
- Interdependencies not always clear
- Often impossible to properly optimise whole chain

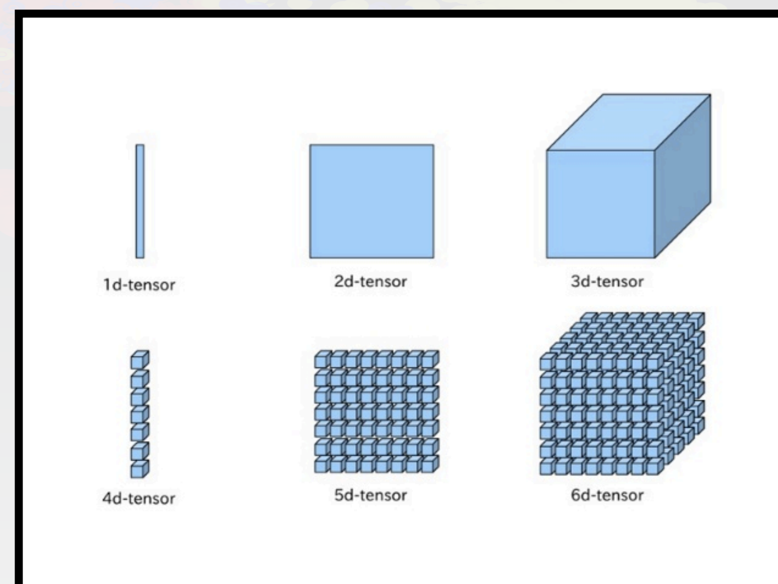
- Always the same patterns
- ***Segmentation/clustering is just a tool***
- Seedless one-stage approach can save resources and is easier to maintain
  - One objective function, fully differentiable, once setup requires O(1) physicists to retrain

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

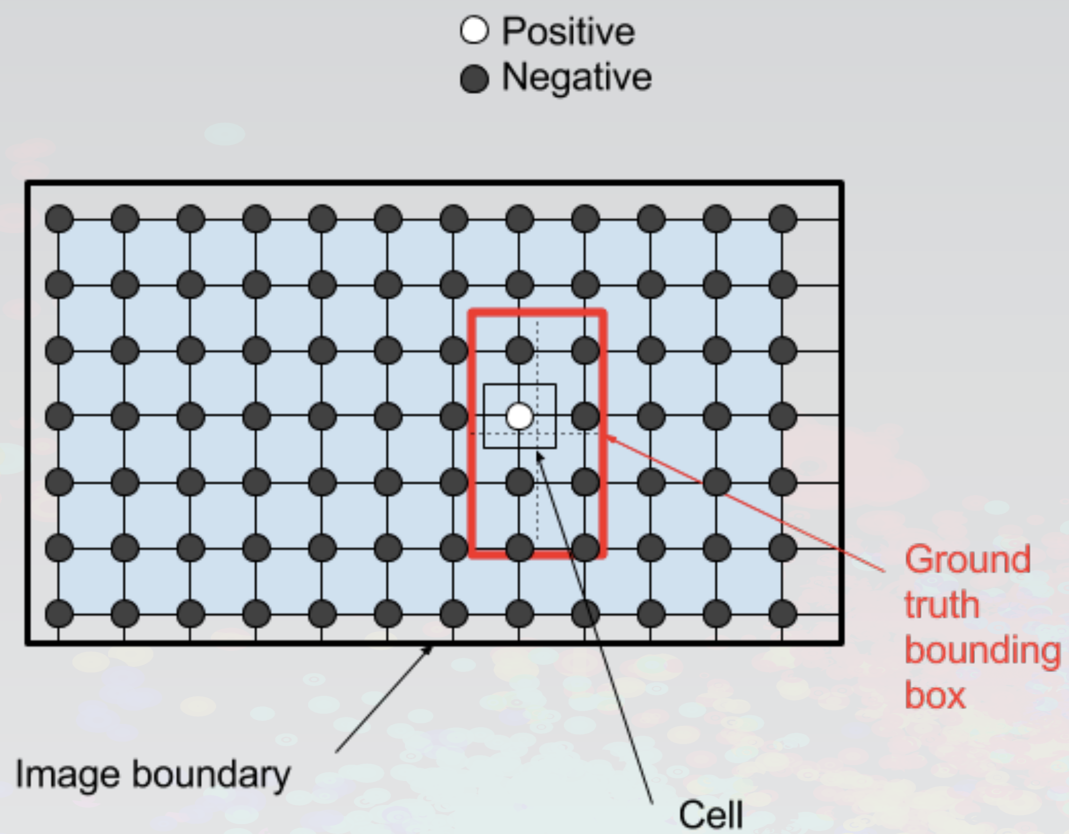


A look at computer vision

- Well known from object detection in images
- Two main approaches:
  - “Traditional’ anchor / bounding box 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



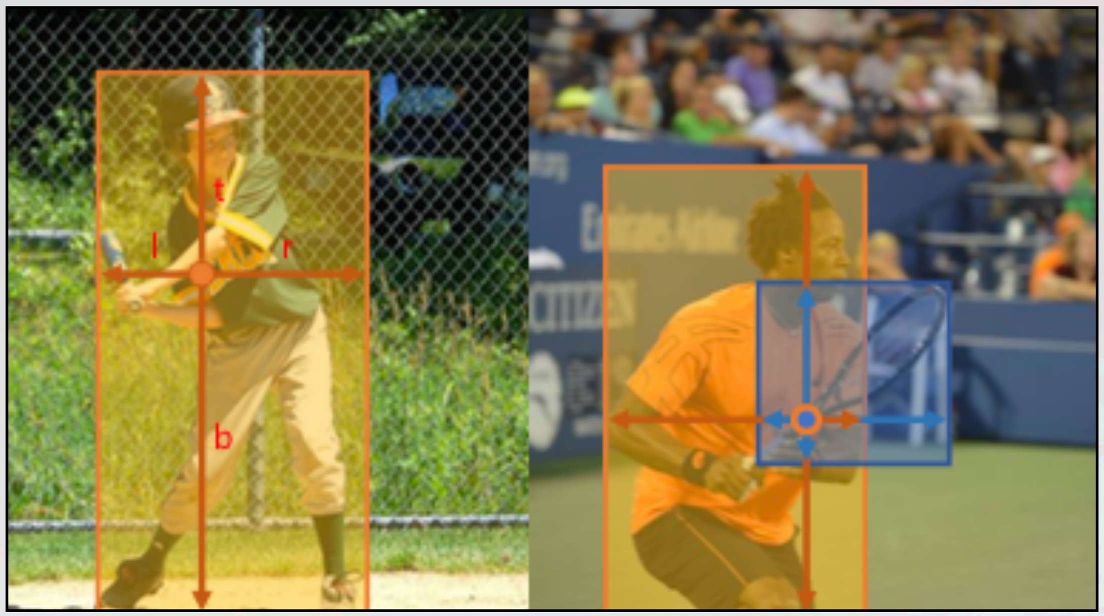
- Anchor points (X x Y per image)
- Assign object score/bounding box to anchor point
- Can also carry other object properties, or IDs follow a different grid (e.g. YOLO)
- Object can be found multiple times

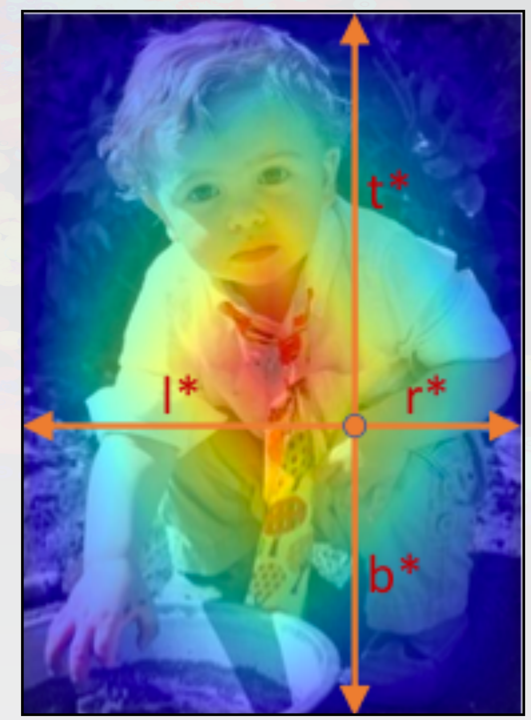
- Anchor points grow with with N^(dim), make implicit assumptions on object size
- A minimal regular grid is assumed
- *Not really optimal for reconstruction based on high-dimensional detector signals*

Figures: towardsdatascience.com

Key point methods

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





Problem: identify the key points

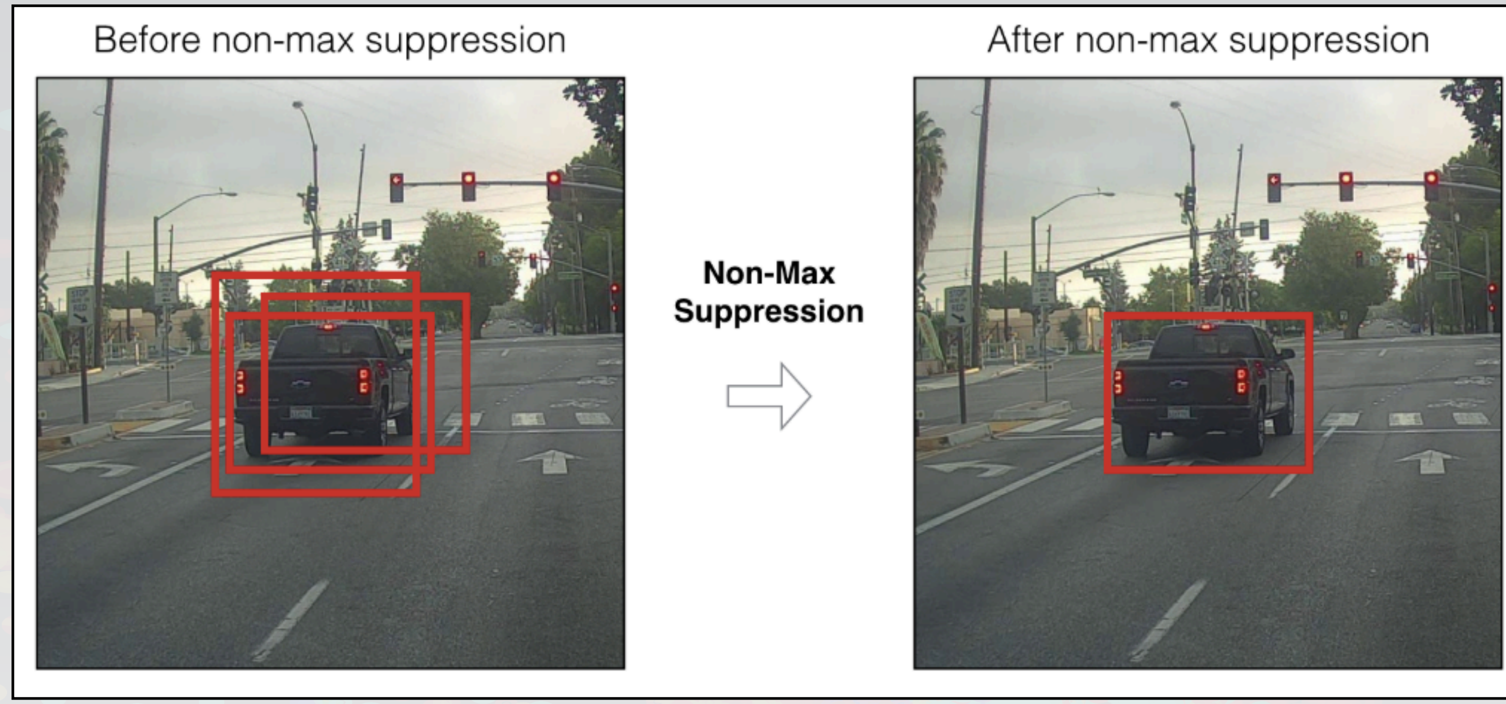
- Also predict ‘center-score’ + bounding box
- Select highest score in the area as key point
    - ***Seed identification!***
    - Heavily relies on objects to have a center: problematic for a particle

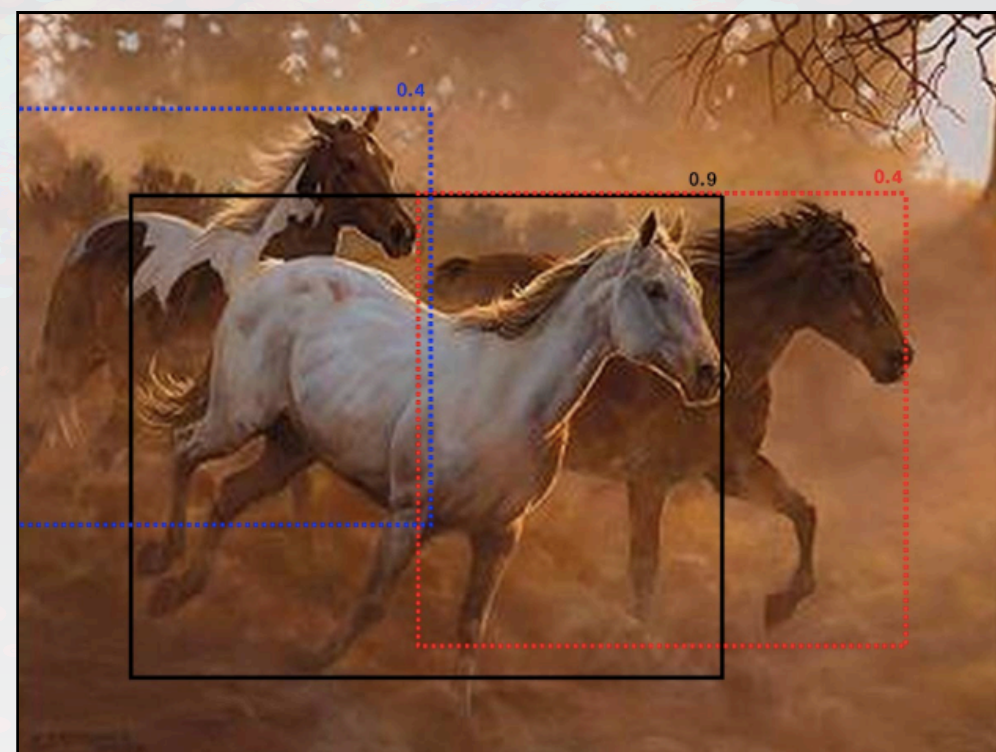
- Remaining ambiguities still need to be resolved

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 generic particles in detectors*





- *Need something more generic, applicable to N dimensions and non-regular geometries (point clouds)*

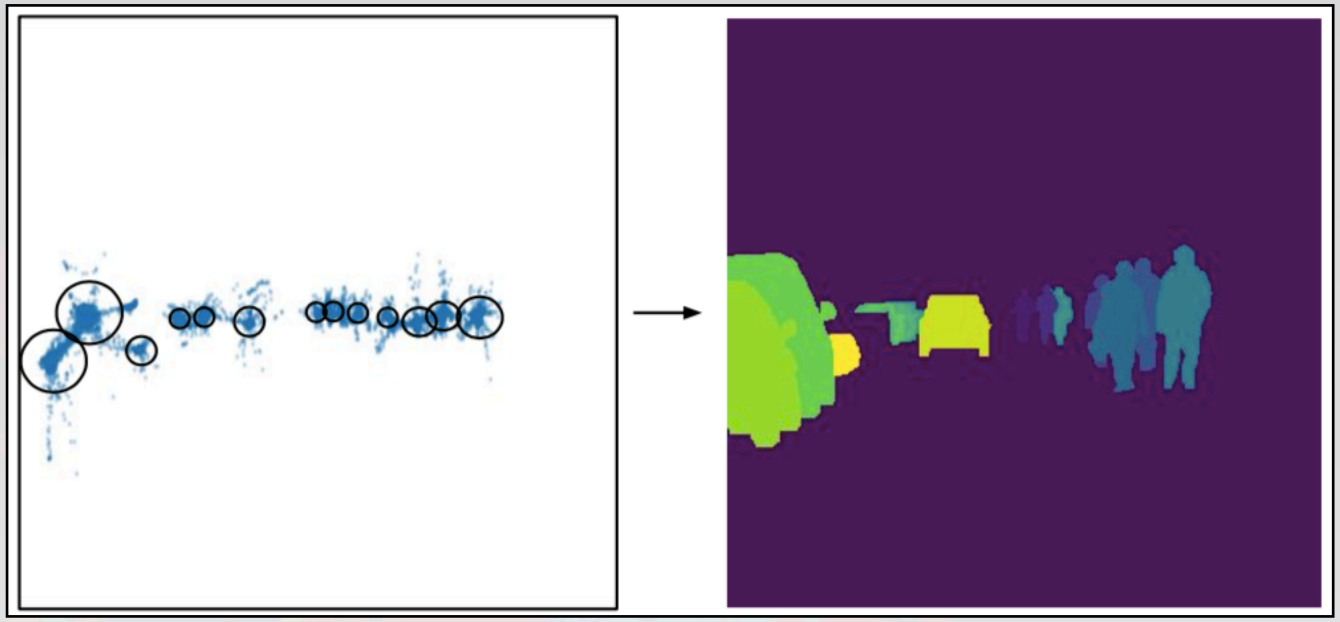
N. Bodla et al, arXiv:1704.04503
Figures: towardsdatascience.com

Segmentation and Clustering

- Maximum number of objects per image/point cloud:
  number of pixels/vertices

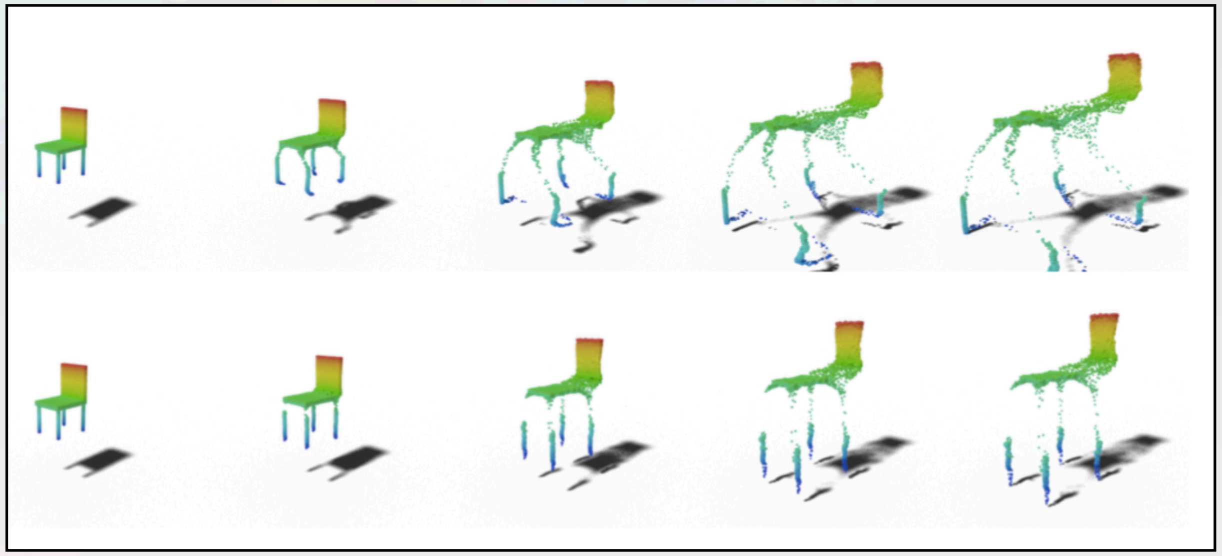
- Learn to move pixels towards the object center
- Map to Gaussian probability

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



- Assign seed score

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

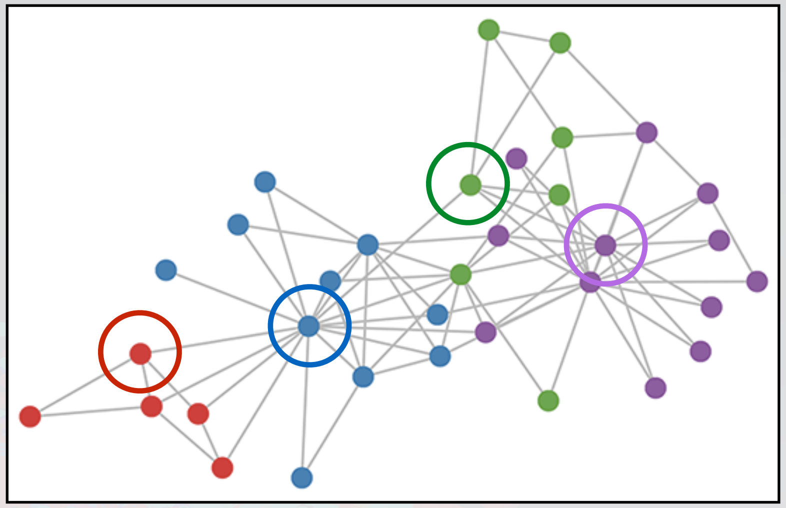


- Collect (from highest seeds score) around the seeds
- *'Only' performs segmentation*
- *Might be problematic if objects don't have distinct centre*
- *Seems to work for the neutrino reconstruction chain → Kazuhiro's IML talk*

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

Object condensation

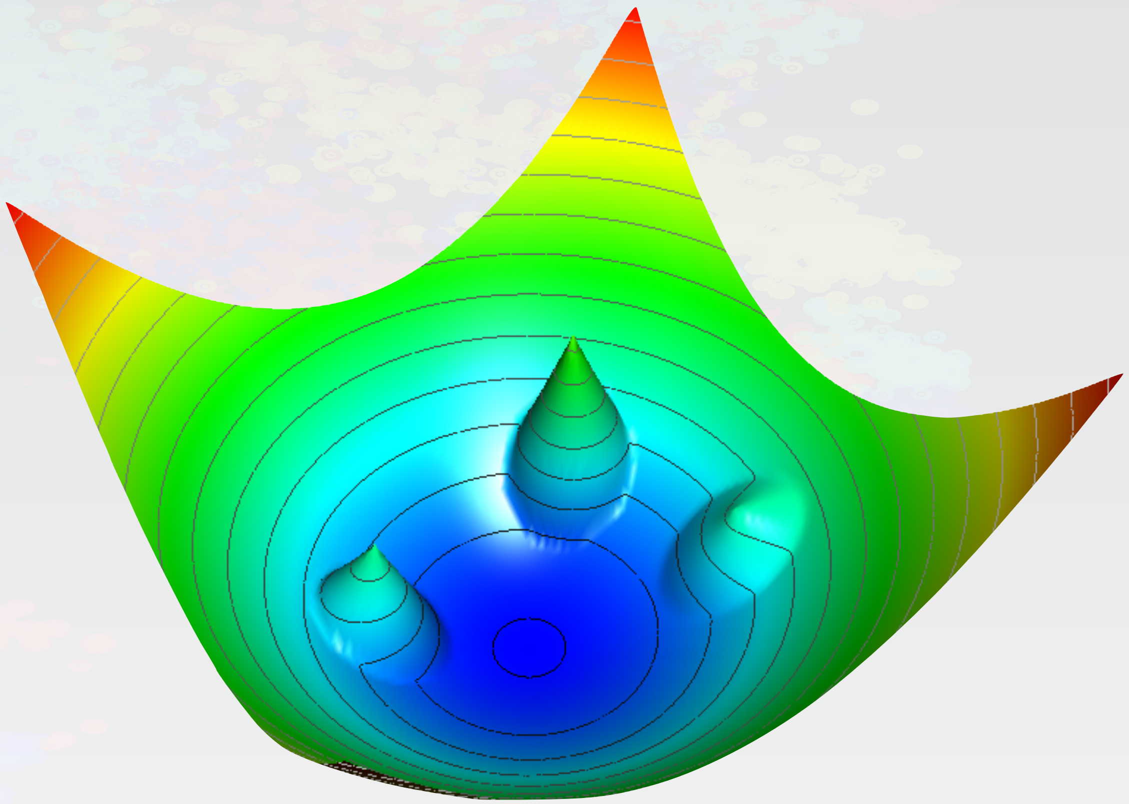
- Aim
  - **Directly** determine object properties (e.g. particle 4 momenta, ID) (graphs, images, …)
  - **Aggregate all object properties** in representative 'condensation point'
  - Resolve ambiguities without IoU (boxes) concept
  - Also perform a clustering/segmentation but:
    - **Detach** input space (3D/4D/5D) from output space
    - Allow for **fractional/ambiguous** assignments
    - Just a tool to **resolve ambiguities** (and for validation)



- 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 (dim(x) > 1)

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

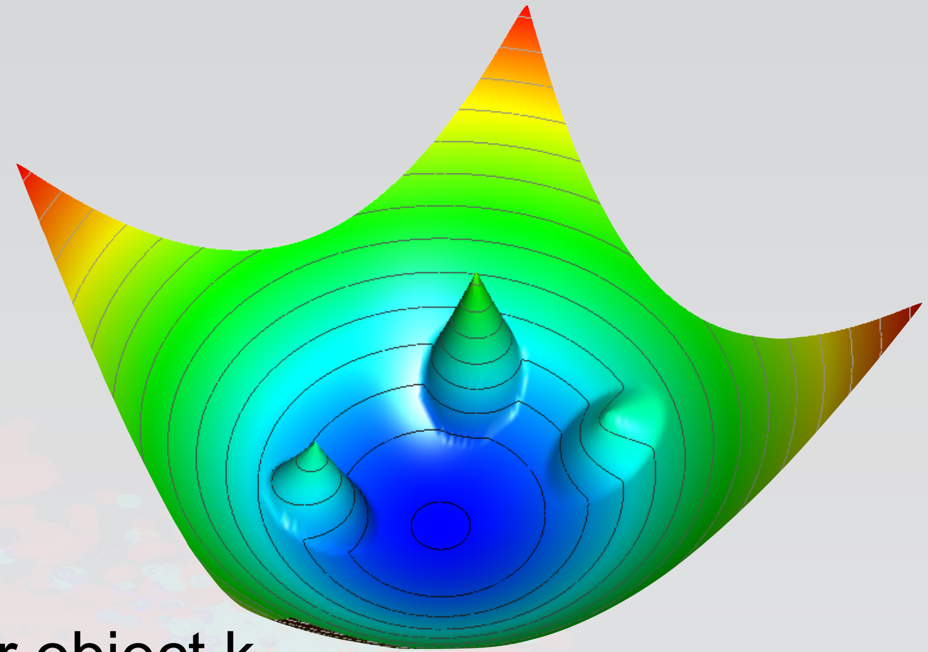


JK, arxiv:2020.03605

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

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

Maximum charge vertex for object k



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

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

x^2 dependence allows detaching from input space, Gradient does not vanish at large Δx unlike for a Gaussian mapping

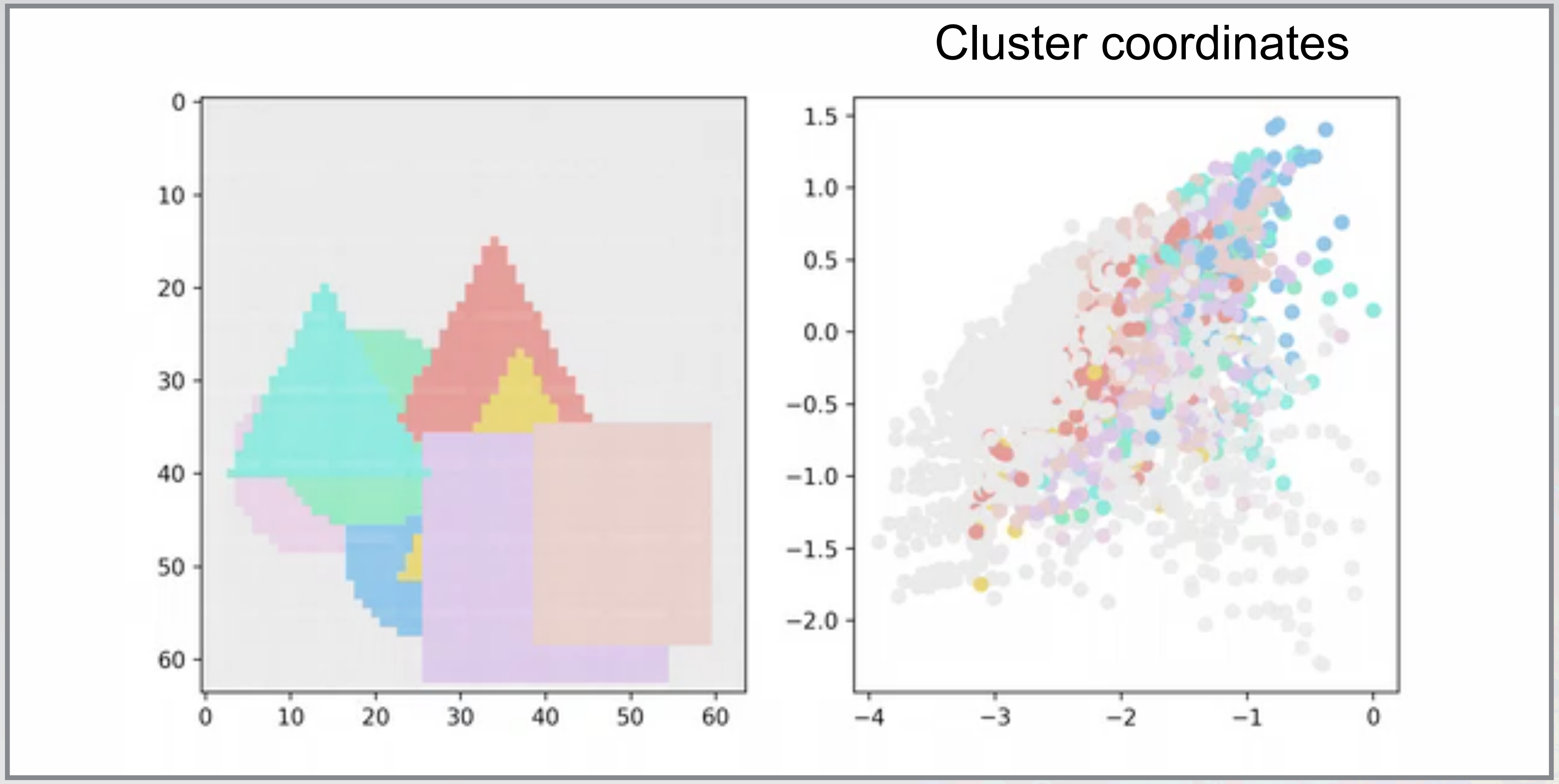
- Also weight object property loss with β

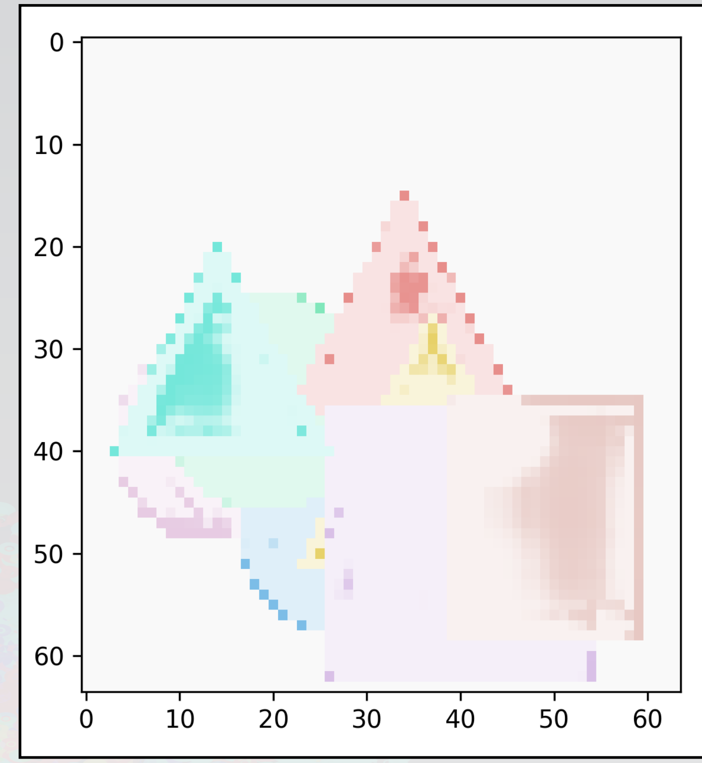
$$L_p = \frac{1}{\sum_{i=0}^N (1 - n_i)\,\mathrm{arctanh}^2\,\beta_i} \sum_{i=0}^N L(t_i, p_i)(1 - n_i)\,\mathrm{arctanh}^2\,\beta_i$$

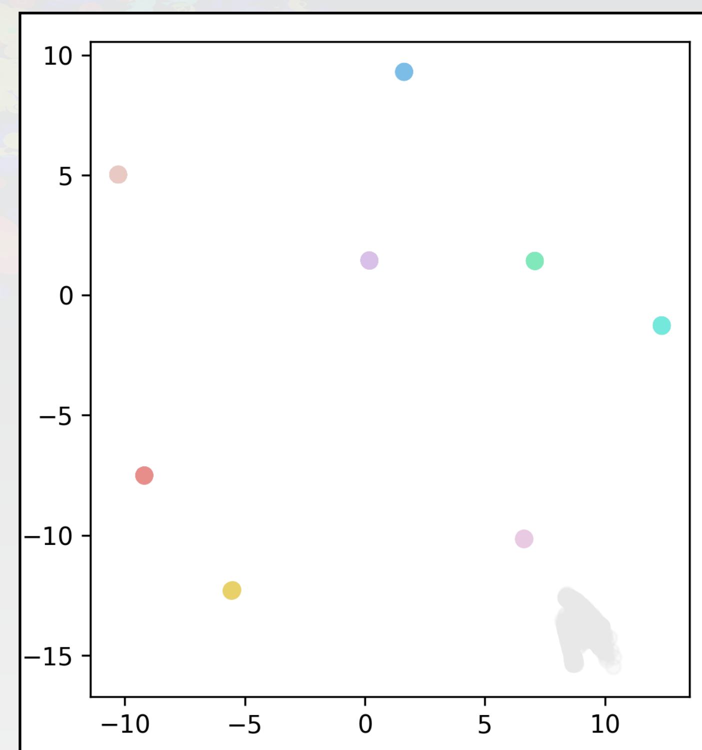
- *Condensation points will carry all object properties*
- *Very natural approach for dynamic graph NN*

*NB: Removes saddle point for large N

Example on image data







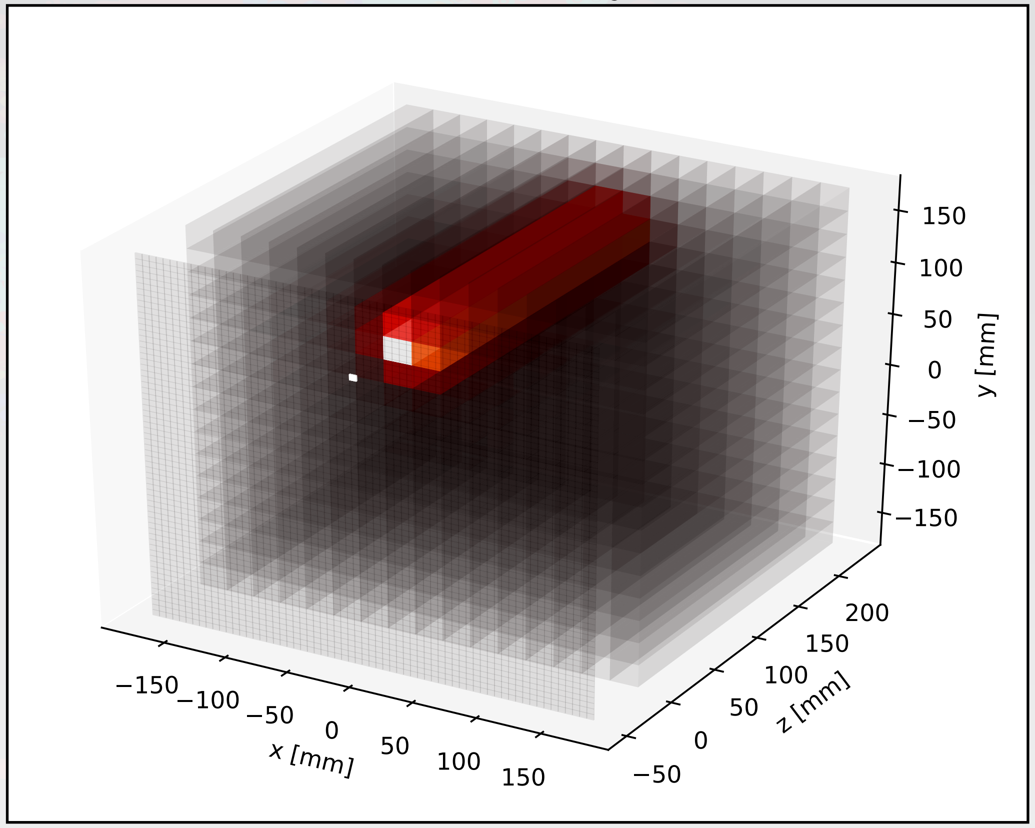
- Proof of principle using images with large overlaps
  - Condensation, object ID
  - Rather simple CNN
- Inference
  - Start with highest β vertex, collect points in $t_d \cong 0.9$
    - Get object properties
  - Repeat until $\beta_{min} \cong 0.1$

JK, arxiv:2002.03605, EPJC

Application to Particle Flow

- Simplified detector
  - "Flat" in x,y: not curved
  - ECal: 16 x 16 cells, each 22 x 22 mm^2 x 26 cm lead tungstate (CMS ECal)
  - No magnetic field
  - "Tracker": 300µm silicon 5.5 x 5.5 mm^2 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
- Track information can be incorporated very naturally (just another point in the cloud)

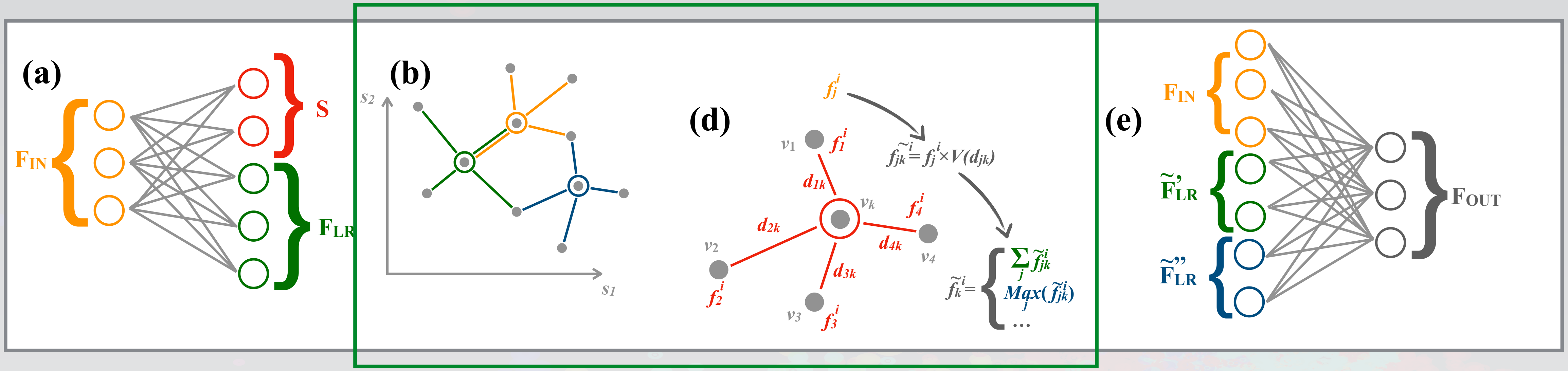
Geometry



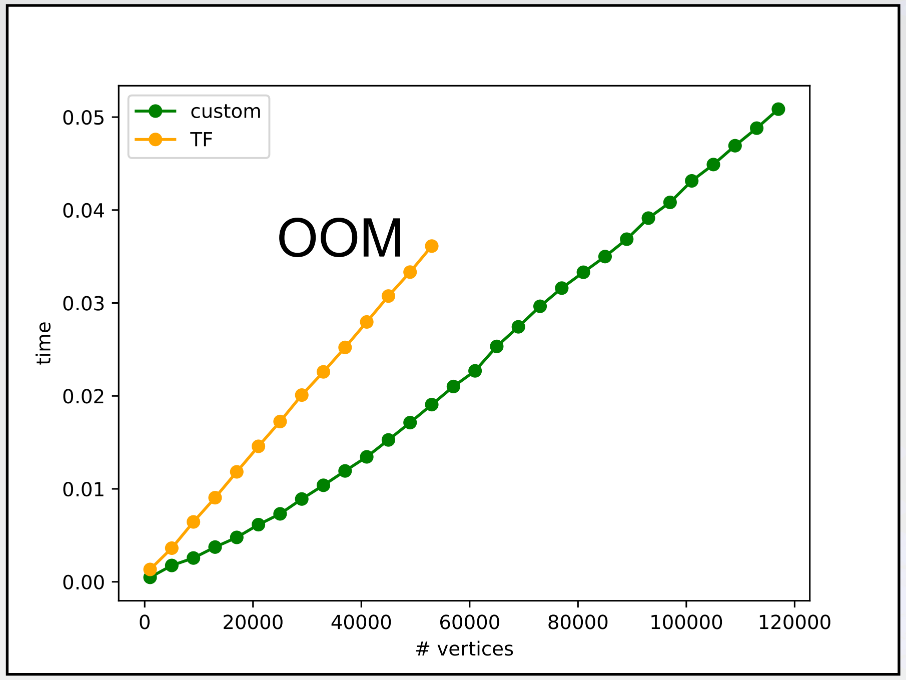
GravNet for High Dimensional Inputs

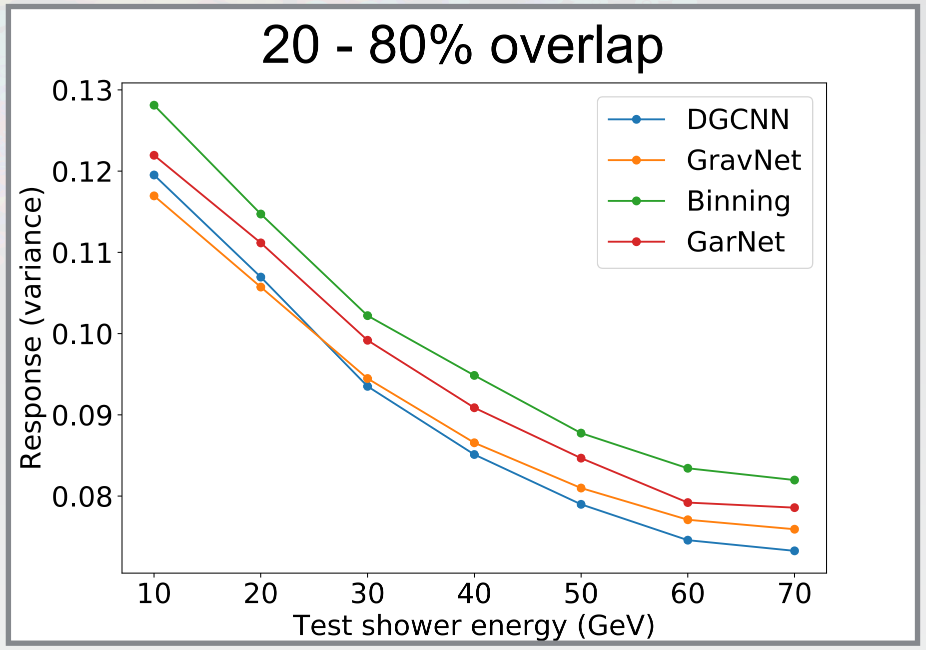
in torch_geometric!

- Non-sparse adjacency not feasible
- Operations in V x K (as e.g. in EdgeConv) are expensive, also for memory.



- Custom CUDA kernels for fast inference/training
  - (Almost) memory scaling with K nearest neighbours





[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

Object Condensation PF

- Truth:
  - Assign particle properties to vertex with highest fraction
- Select 200 highest energy deposits/tracks

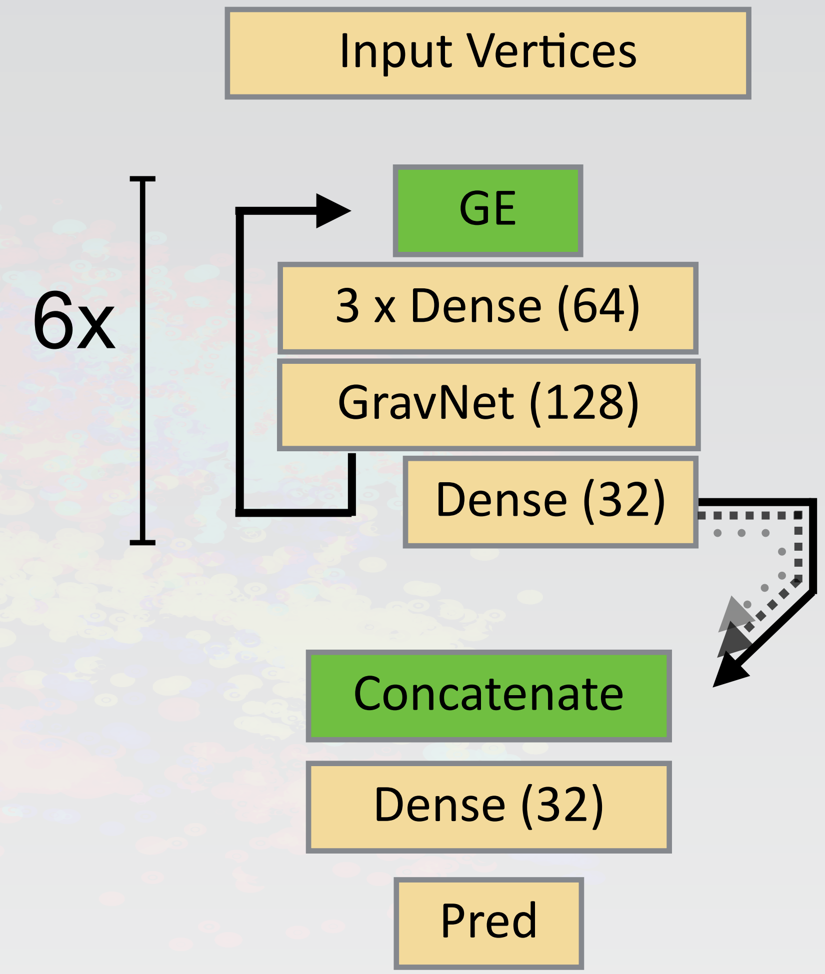
- Use rather standard GravNet
  - 10 neighbours, 4 space dimensions, 64 features to be exchanged

- Predict:
  - OC Clustering space
  - OC Confidence beta
  - Position (offset w.r.t. sensor position)
  - Energy = Momentum (correction factor w.r.t. sensor energy)

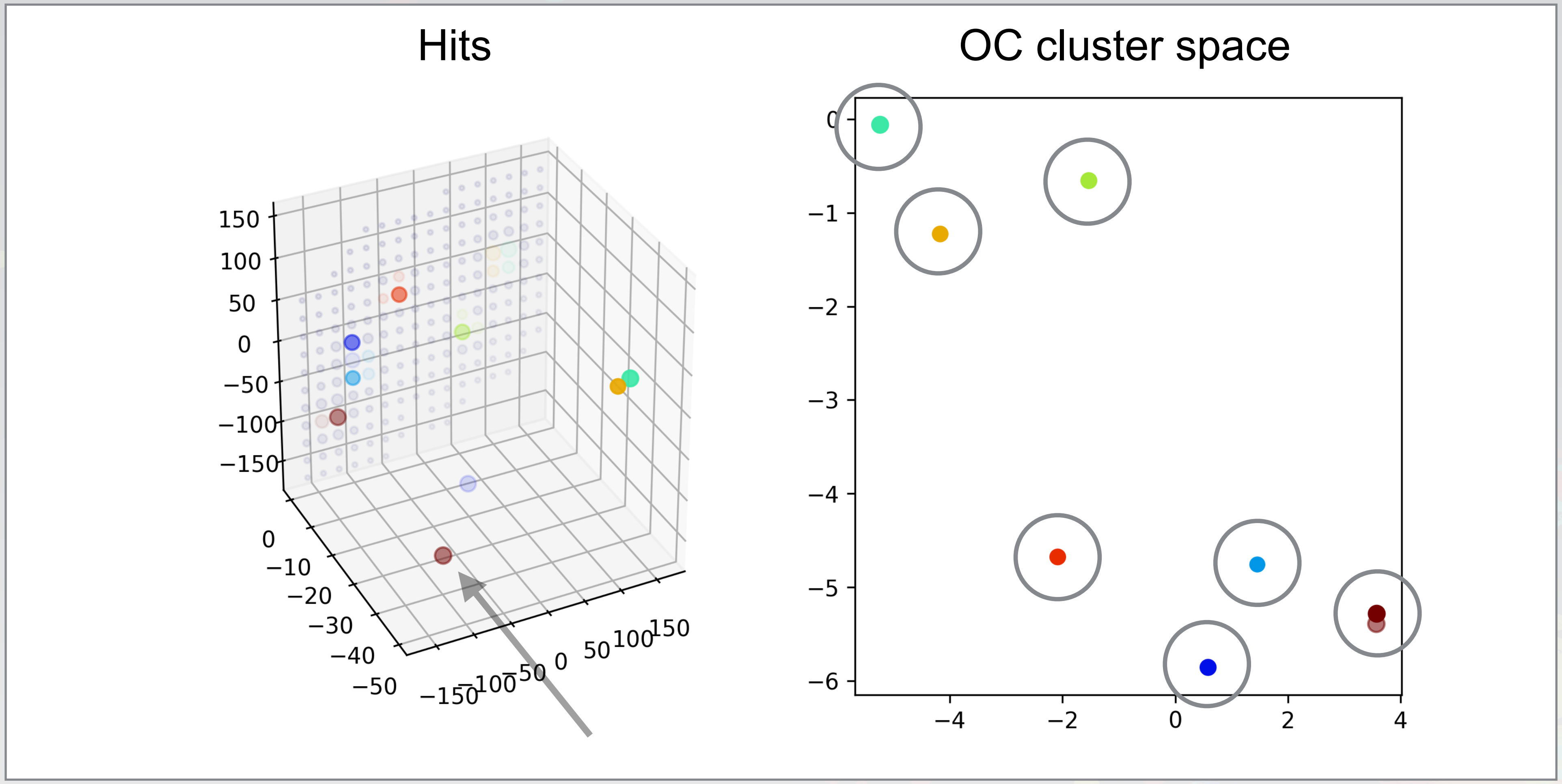
- Sample: 1.7M events, 1-9 particles per event
  - Trained for 110 epochs, learning rate decrease after 20 epochs

- Set minimum OC clustering charge to 0.1 (less segmentation focus)

- Very similar approach now also being applied to CMS HGCal reconstruction

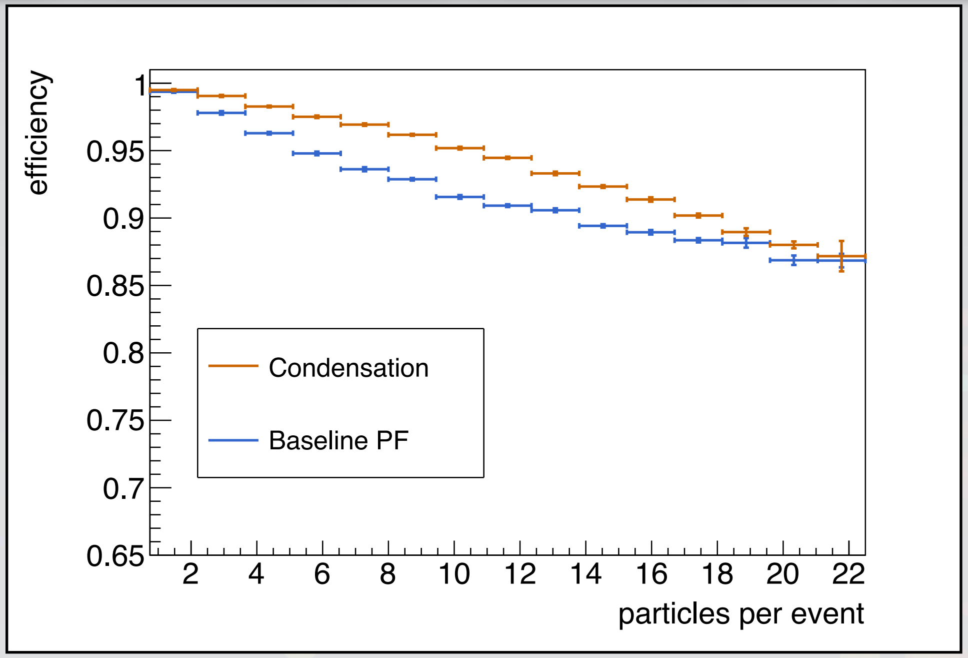


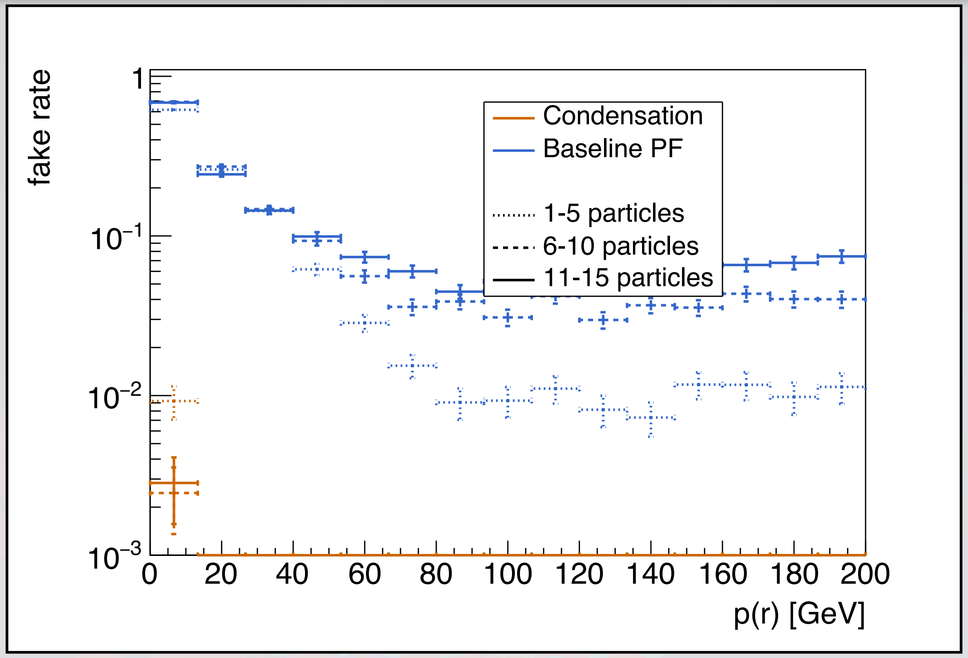
Segmentation / Postprocessing

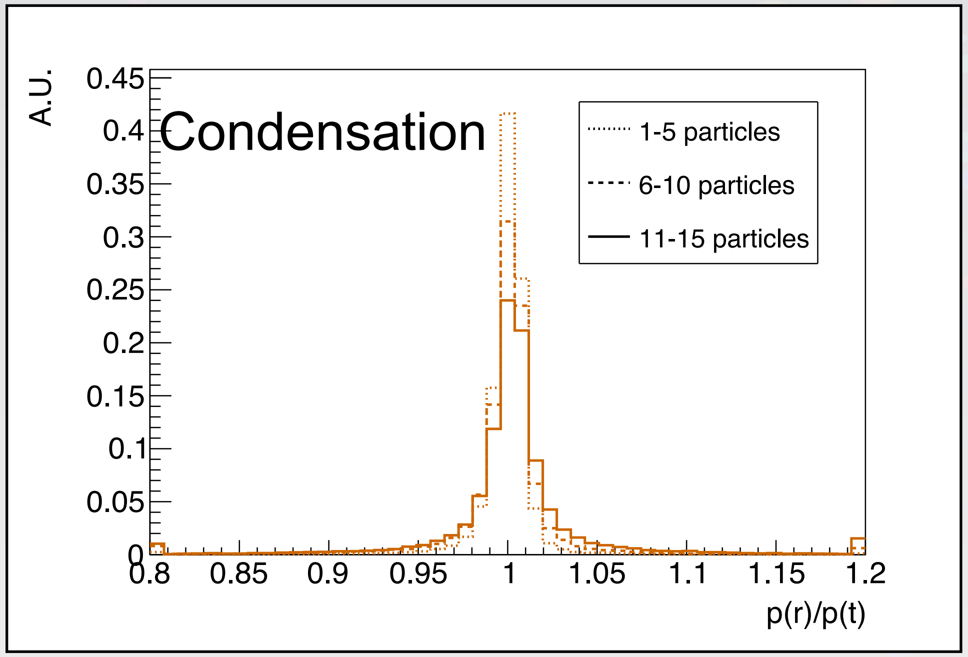


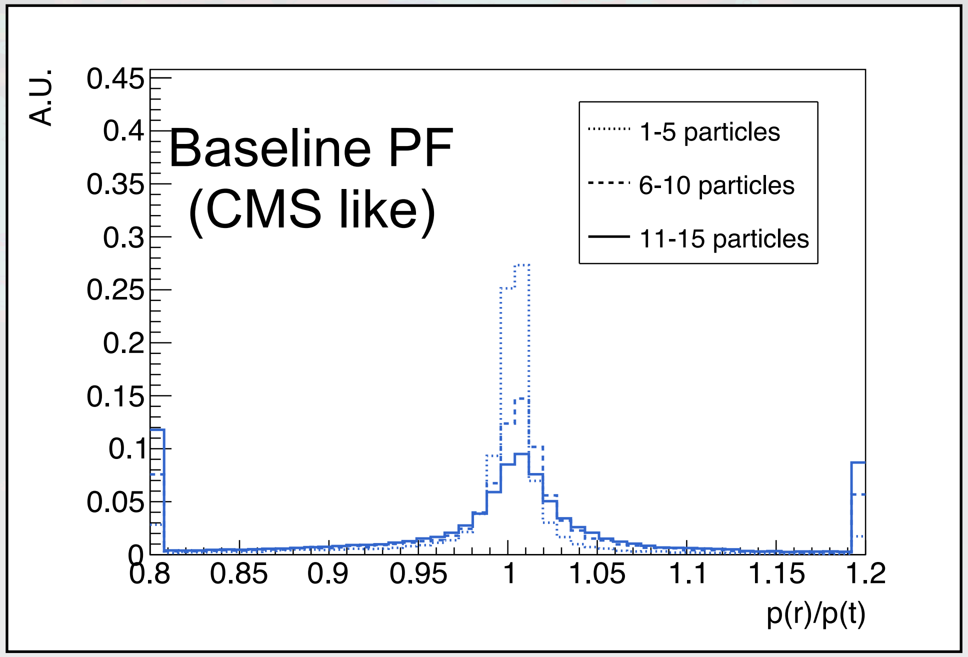
- Start with highest β vertex, collect points in td ≅ 0.8
- Get object properties
- Repeat until $\beta_{min} \cong 0.1$

Particle Efficiency and Response







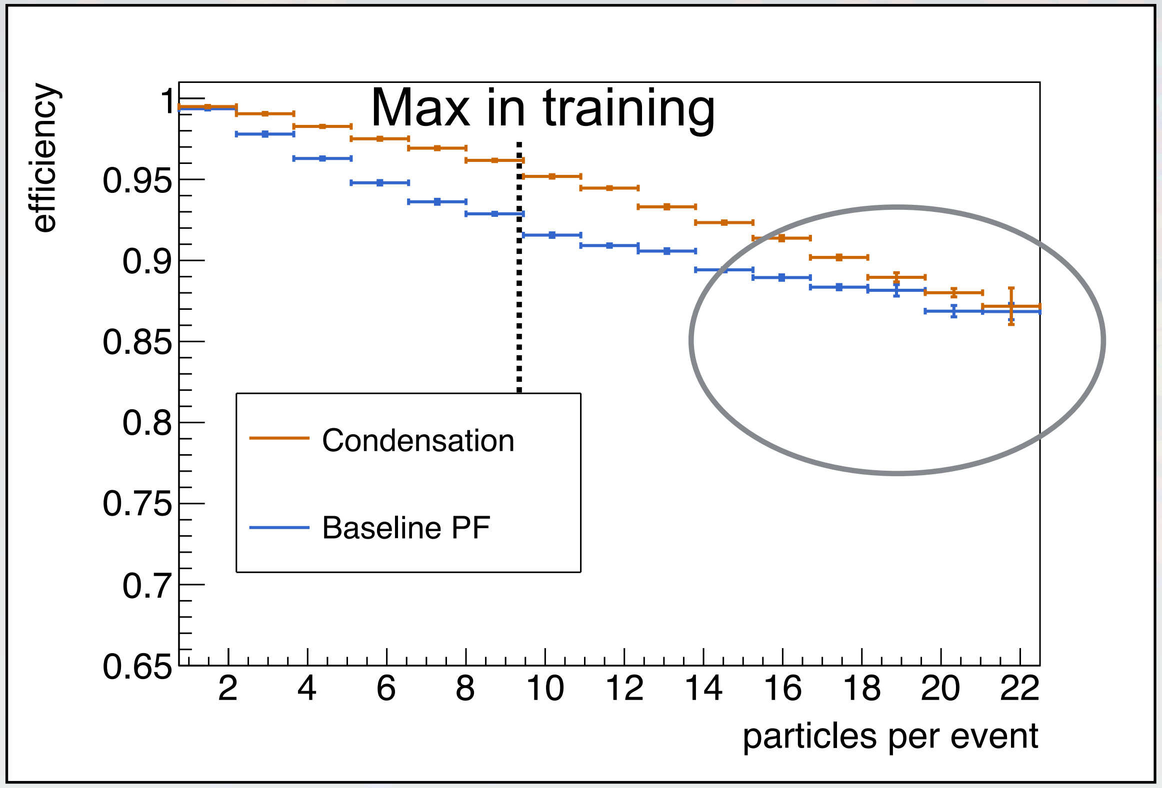


- Low fake rate, and fakes only at low energies
- Improved single particle resolution

JK, arxiv:2002.03605, EPJC

“Jet” properties

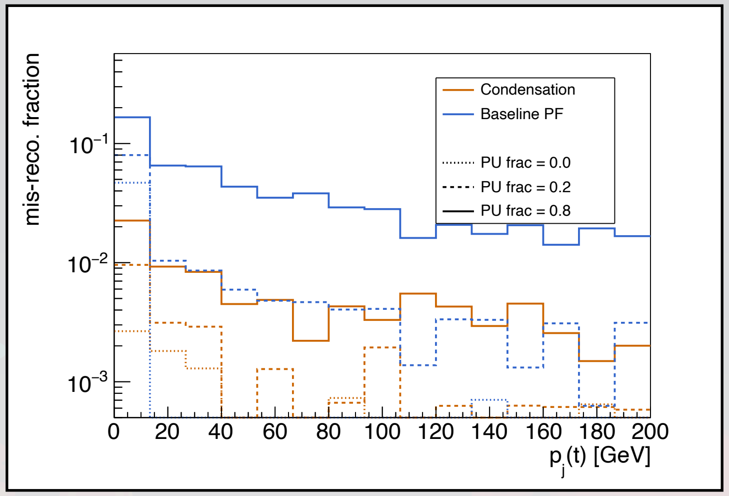
- Generate jet-like sample
  - Create particles (50/50 photons/electron) using exponentially falling energy spectrum (exp(-ln(300) * E/GeV)
  - Randomly pick N particles, with N being Poisson distributed around an average of M (M being drawn from a uniform distribution with M <= 15)
  - Gives jets proxies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

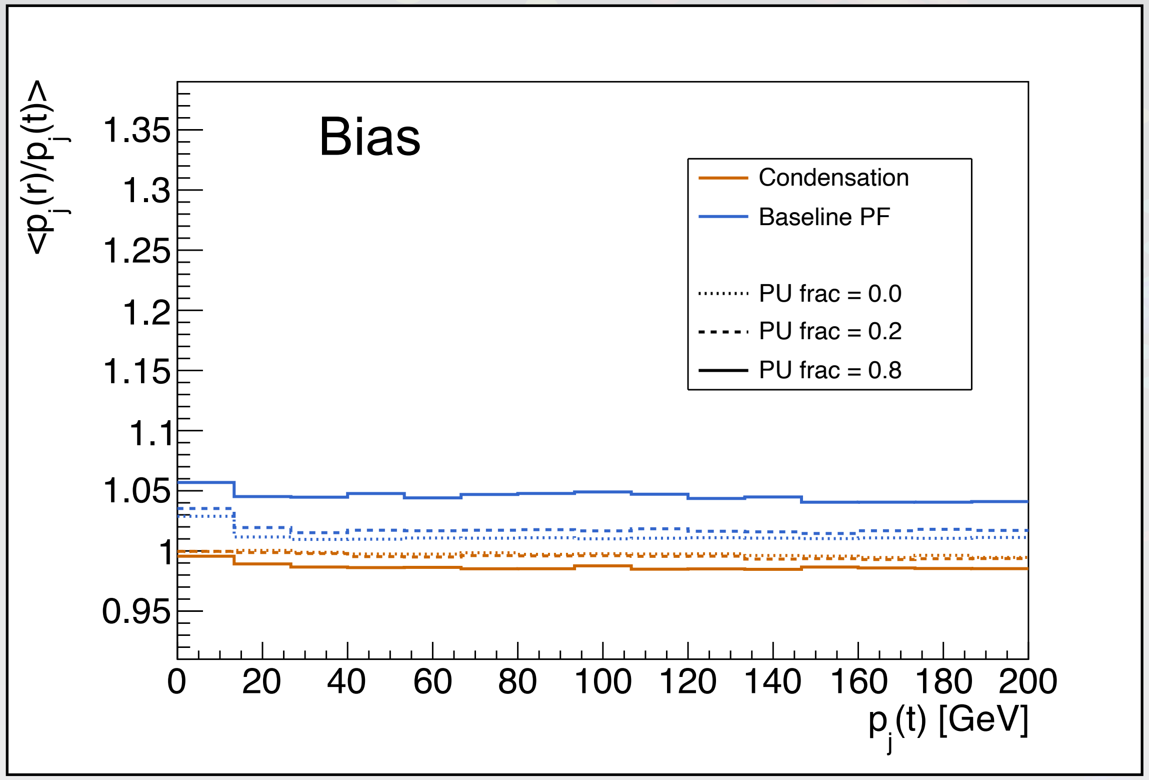


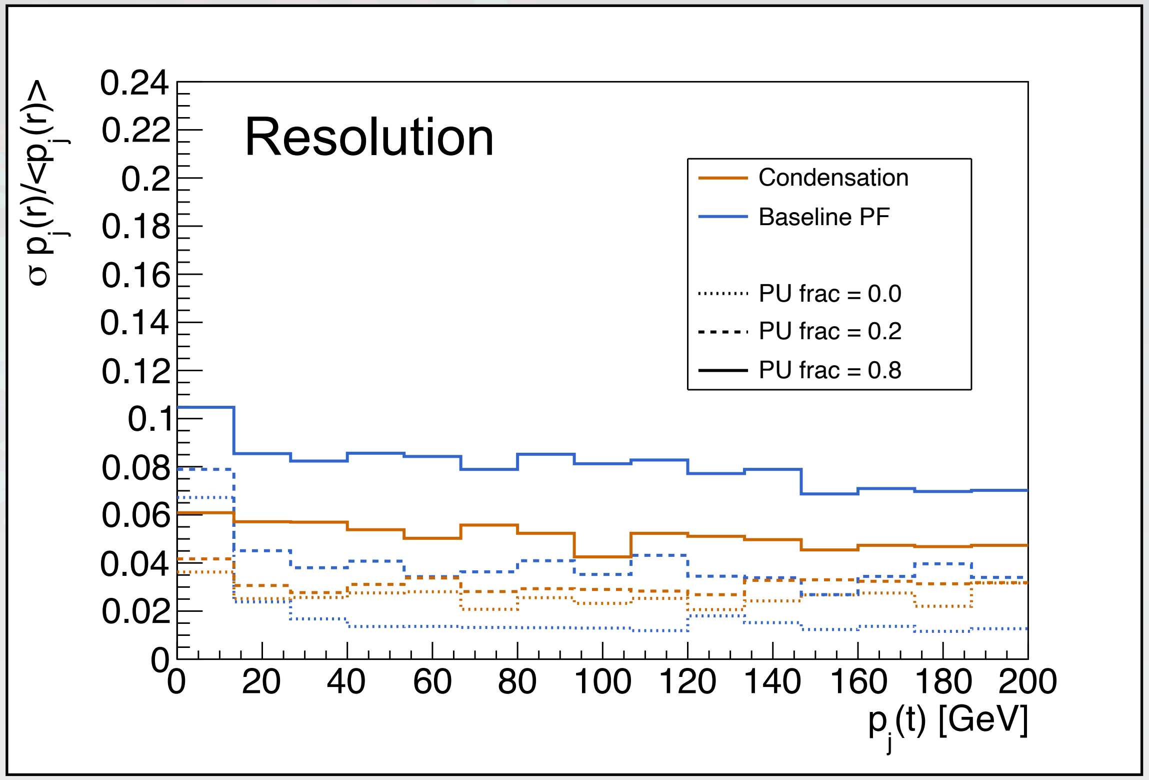
For PF, truth matching likely starts
 to fail
-> will look at “jet” properties so doesn’t matter

- **Excellent extrapolation behaviour** for significantly larger particle densities than seen in the training!
  - Both GravNet and OC are *local*

"Jet" momentum 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, EPJC

Summary

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

- No significant 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**

- **Application to more realistic environments is ongoing (e.g. CMS HGCal)**