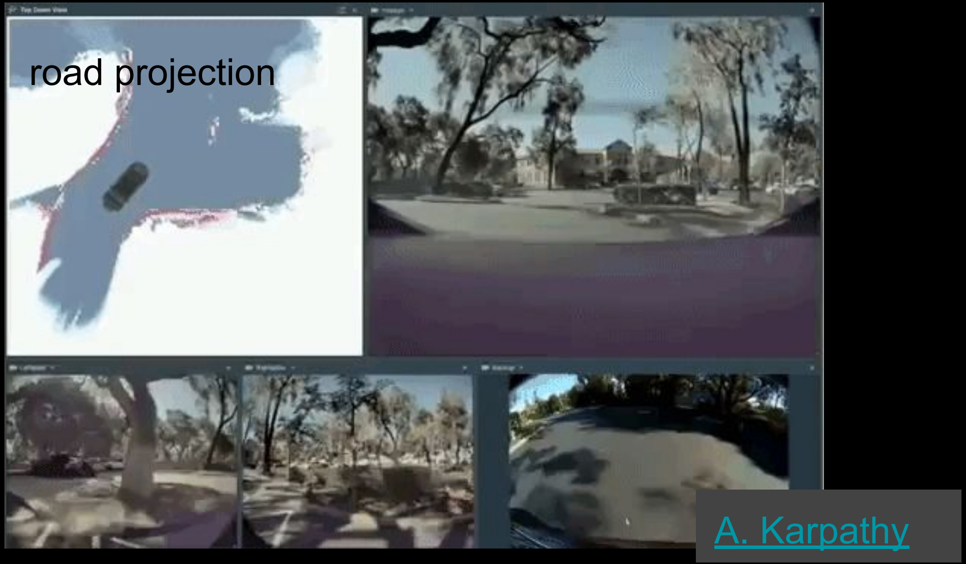
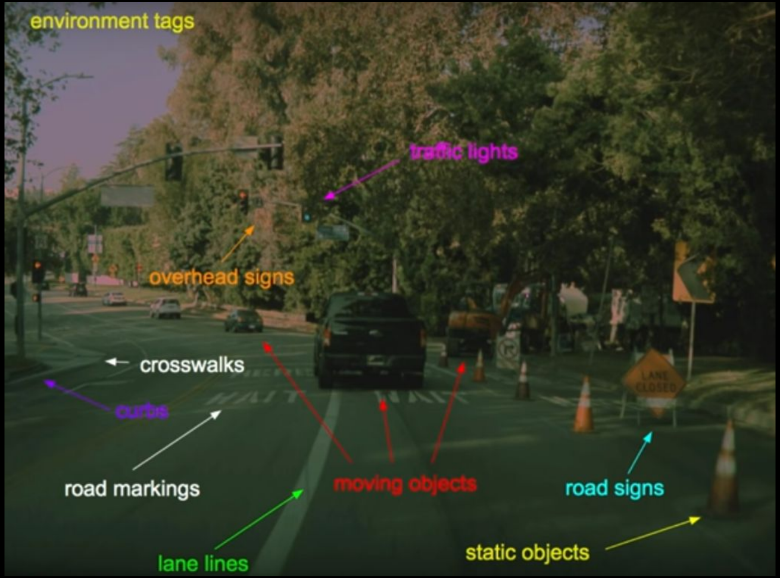
AI-supported algorithms for Streaming Readout

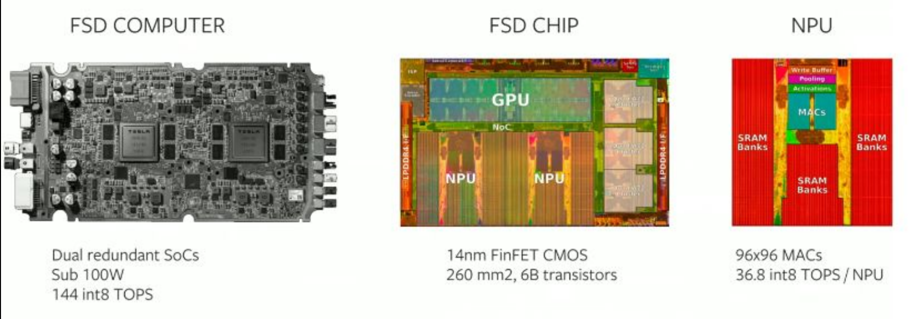
C. Fanelli





stream of data

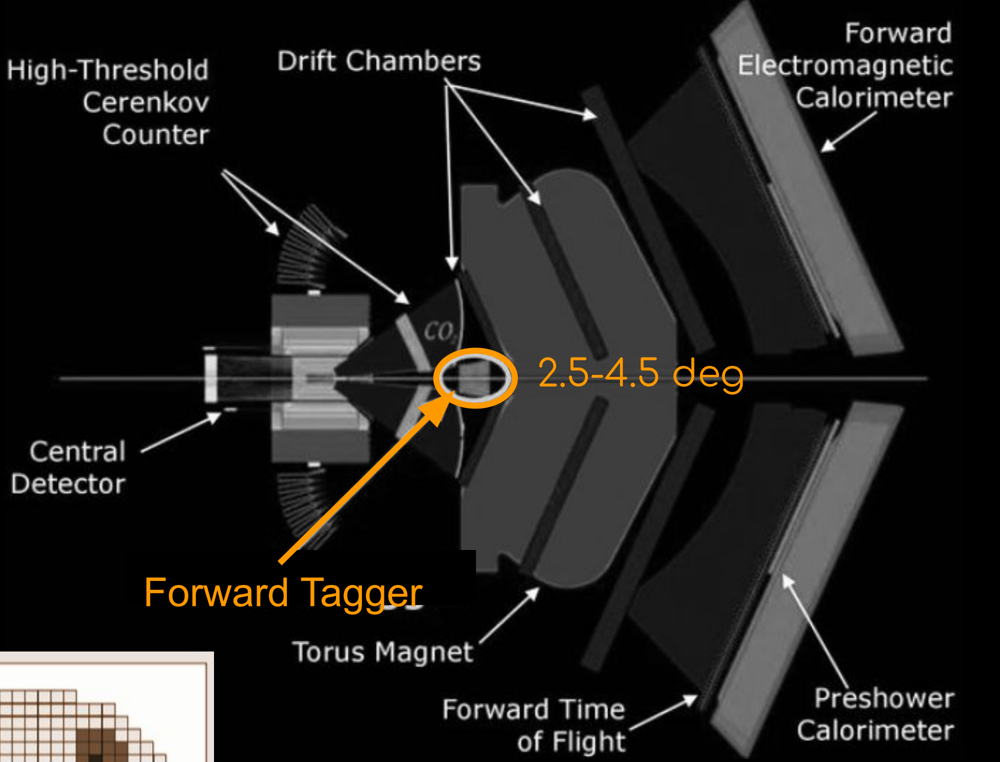
Custom architecture and hardware

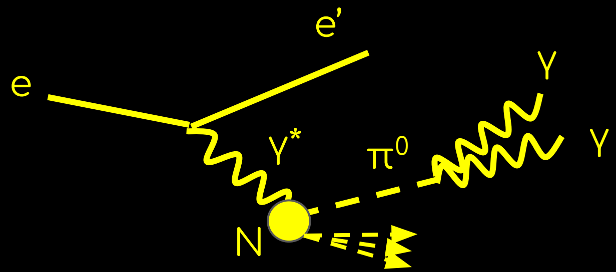


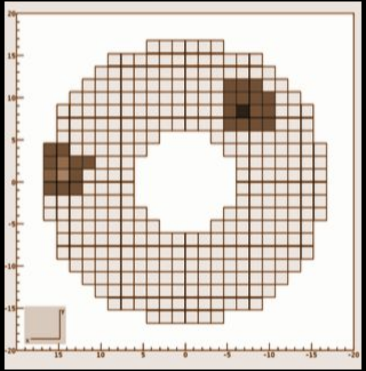
Self Driving
TESLA Example

See slides of D. Romanov, Streaming Readout V

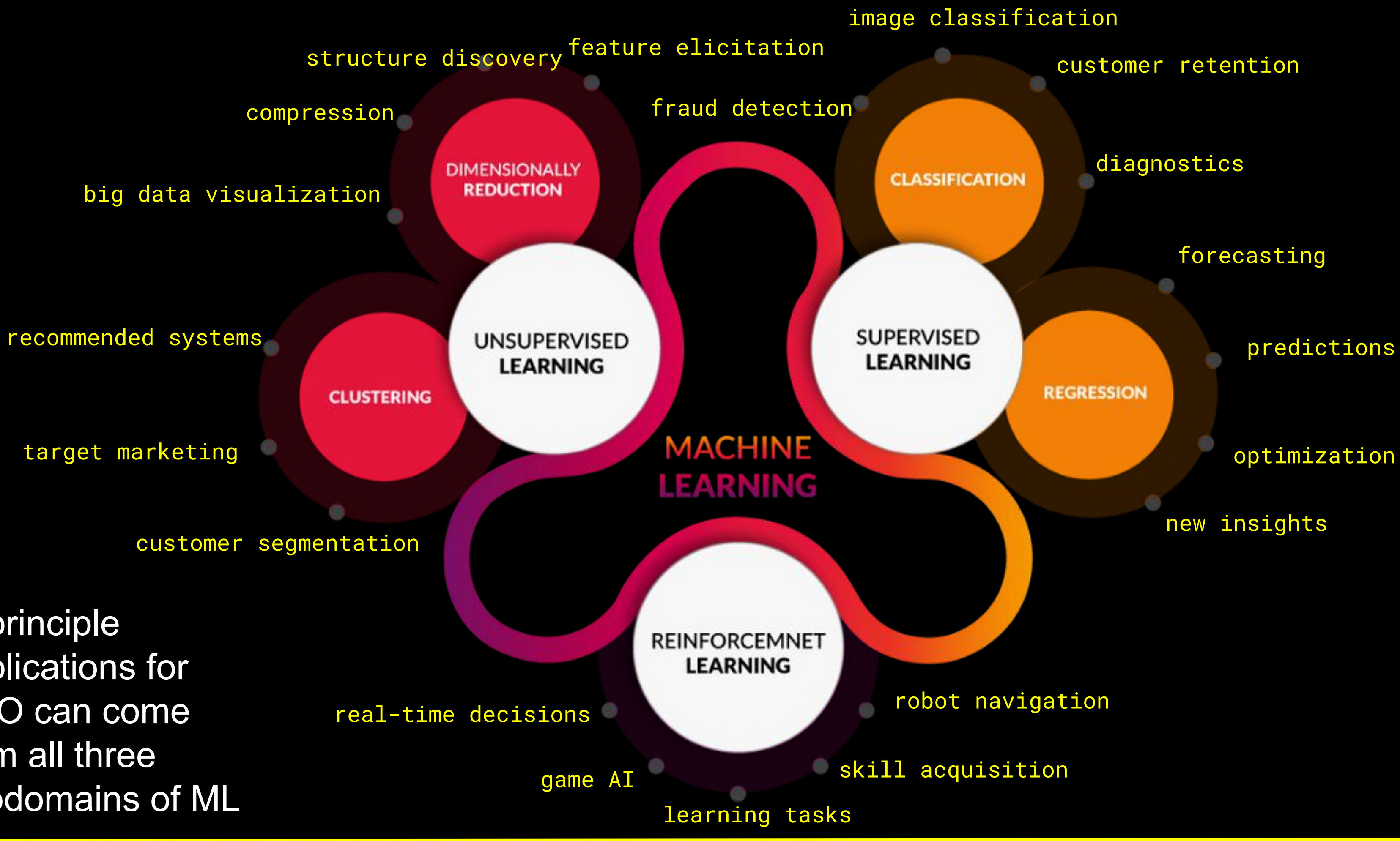
- CLAS12 SRO setup
  - S. Boyarinov, M. Battaglieri talks
- TriDAS SR backend
  - See slides of T. Chiarusi
- JANA2 reconstruction framework
  - See slides of D. Lawrence
- Analysis of π^0 electro-production





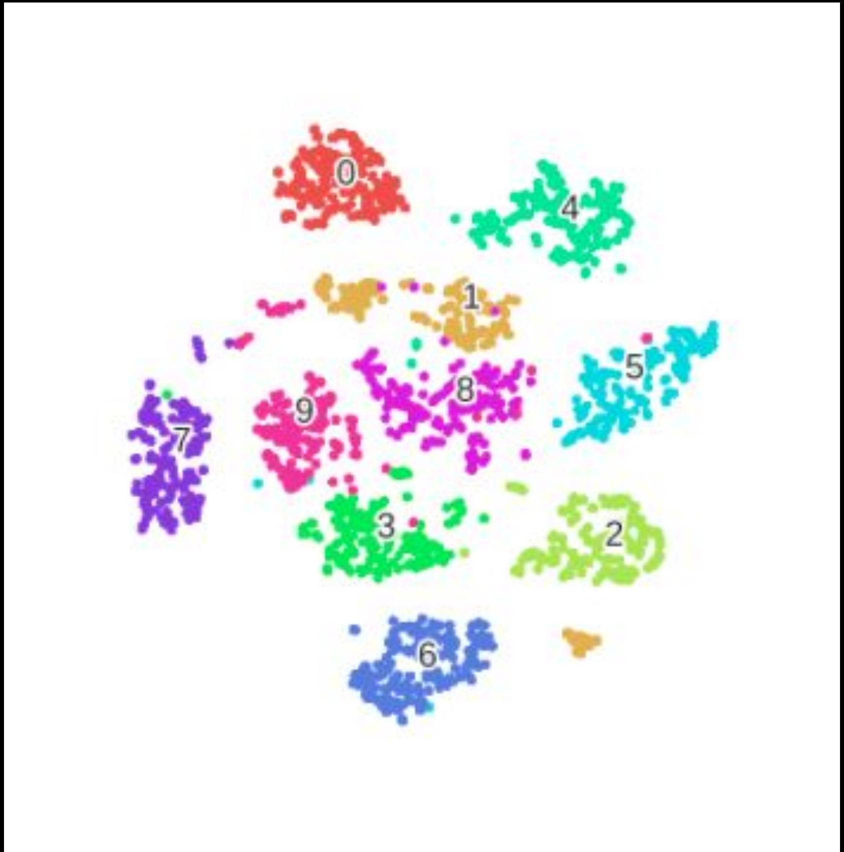


Photons detected in the FT-Calorimeter



In principle applications for SRO can come from all three subdomains of ML

Unsupervised



NIPS 2016: “If intelligence is a **cake**, the bulk of the **cake** is **unsupervised** learning, the icing on the **cake** is supervised learning, and the cherry on the **cake** is reinforcement learning (RL).”

LeCun, Turing award 2018
VP and Chief AI Scientist, Facebook

Clustering

- Clustering needs to work in SRO mode: AI as an alternative to standard clustering.
- AI-based algorithms look at all the available information in the calorimeter at the hit-level, x, y, t, E, to learn correlations: clusters of objects share common features
  - Need to define a metric in a space.
  - The observables of a cluster can be determined by weighing the corresponding quantities of the hits belonging to that cluster.
- Tests on minimum bias trigger data and then on SRO.
- When using machine learning one typically introduces hyperparameters:
  - No optimization done
  - The following preliminary results which can be further improved.
- Going to show results based on semi-supervised and unsupervised.

Semi-supervised Clustering: e.g., K-means

STEP 1: Choose the number K of clusters

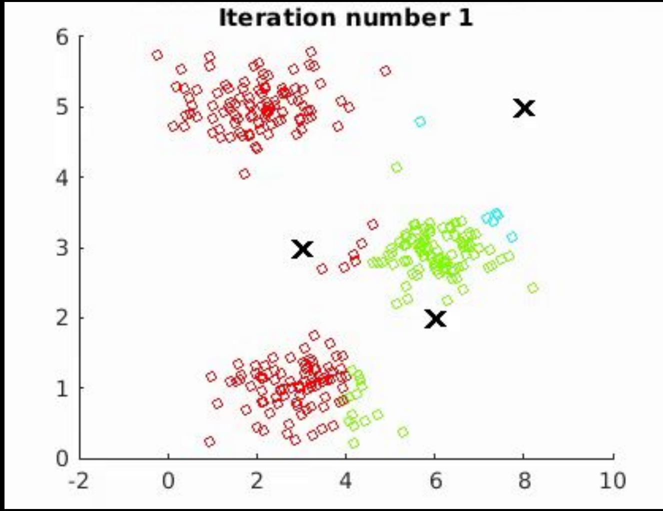
STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

STEP 3: Assign each data point to the closest centroid (That forms K clusters)

STEP 4: Compute and place the new centroid of each cluster

STEP 5: Reassign each data point to the new closest centroid
If any reassignment took place, go to STEP 4,
otherwise go to FIN.

Your Model is Ready



Hyperparameters and metrics

Table 2. The different metrics used for k-means.

metric	description
$(X_{hit} - X_{mean})^2 + (Y_{hit} - Y_{mean})^2$	squared 2D space distance
$\frac{(X_{hit}-X_{mean})^2}{L^2_{cell}} + \frac{(Y_{hit}-Y_{mean})^2}{L^2_{cell}} + \frac{(t_{hit}-t_{mean})^2}{(50\ ns)^2}$	squared 3D space-time distance
$\frac{(X_{hit}-X_{mean})^2}{L^2_{cell}} + \frac{(Y_{hit}-Y_{mean})^2}{L^2_{cell}} + \frac{(t_{hit}-t_{mean})^2}{(50\ ns)^2} + (E_{hit} - E_{mean})^2$	squared 4D space-time-energy distance

Table 3. The main parameters of the k-means algorithm are described and their values reported. For each parameter, the last column shows when it intervenes, either if in the pre-processing or in the clustering phase.

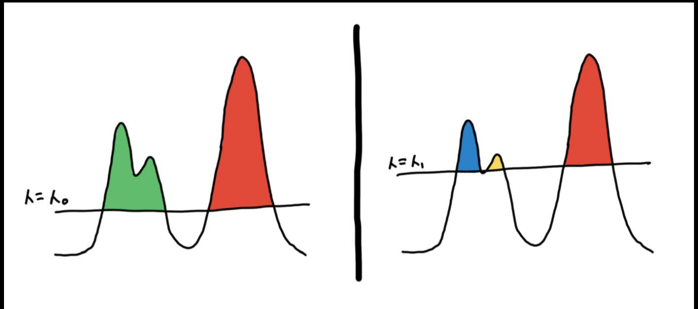
parameter	**description**	**value** [units]	**phase**
t threshold	minimum time of hits	0. ns	preprocessing
E threshold	minimum energy of hits	0. GeV	preprocessing
time_window	time difference between hits	50 ns	preprocessing
count_cells	active neighbor cells for each hit	≥ 1	preprocessing
iterations	k-means updates	10 (30)	clustering
bad_distance	max distance hit-cluster	not used	clustering
bad_time	max time difference hit-cluster	not used	clustering
norm_space	normalization space distance hit-cluster	L_cell (cell length, see Tab. 2)	clustering
norm_time	normalization time difference hit-cluster	50 ns (see Tab. 2)	clustering
norm_ene	normalization energy difference hit-cluster	not used	clustering

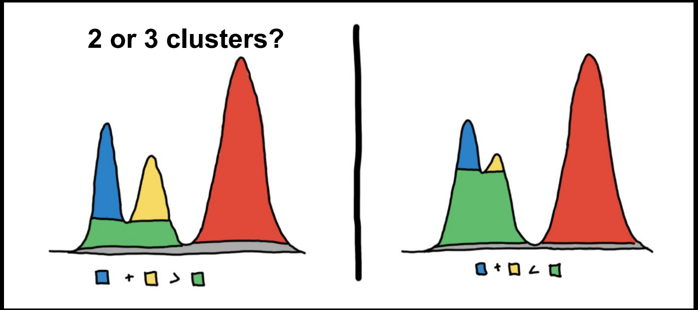
$$bool = \Delta t < 50\ ns\ \&\&\ \Delta X \le 1\ \&\&\ \Delta Y \le 1\ \&\&\ (\Delta X + \Delta Y) > 0 \tag{3.1}$$

For K-means we need to make some assumptions, in particular we need to provide the seeds.

Unsupervised: e.g., Hierarchical Clustering

Two different clusterings based on two different level-sets



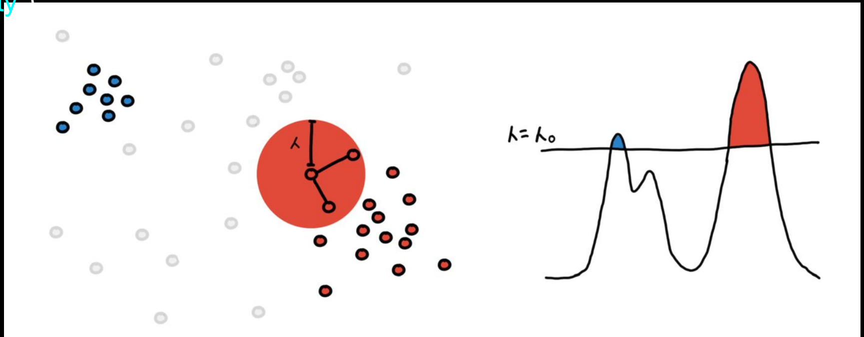


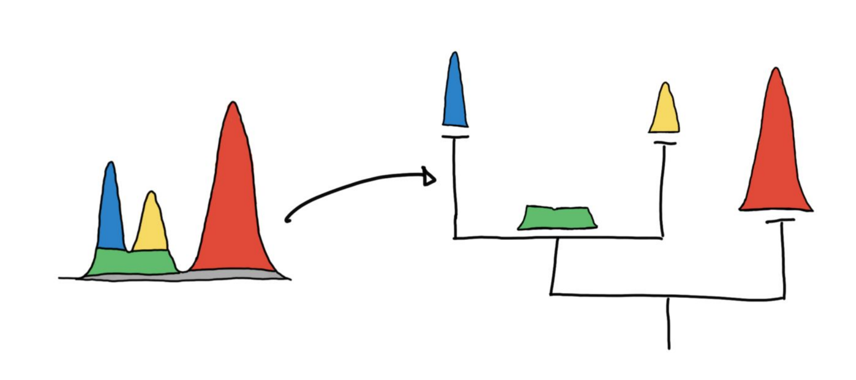
The area of the regions is the measure of “persistence”.

Maximize the persistence of the clusters under the constraint that they do not overlap.

Core distance (defined by a required # of neighbors) as estimate of density

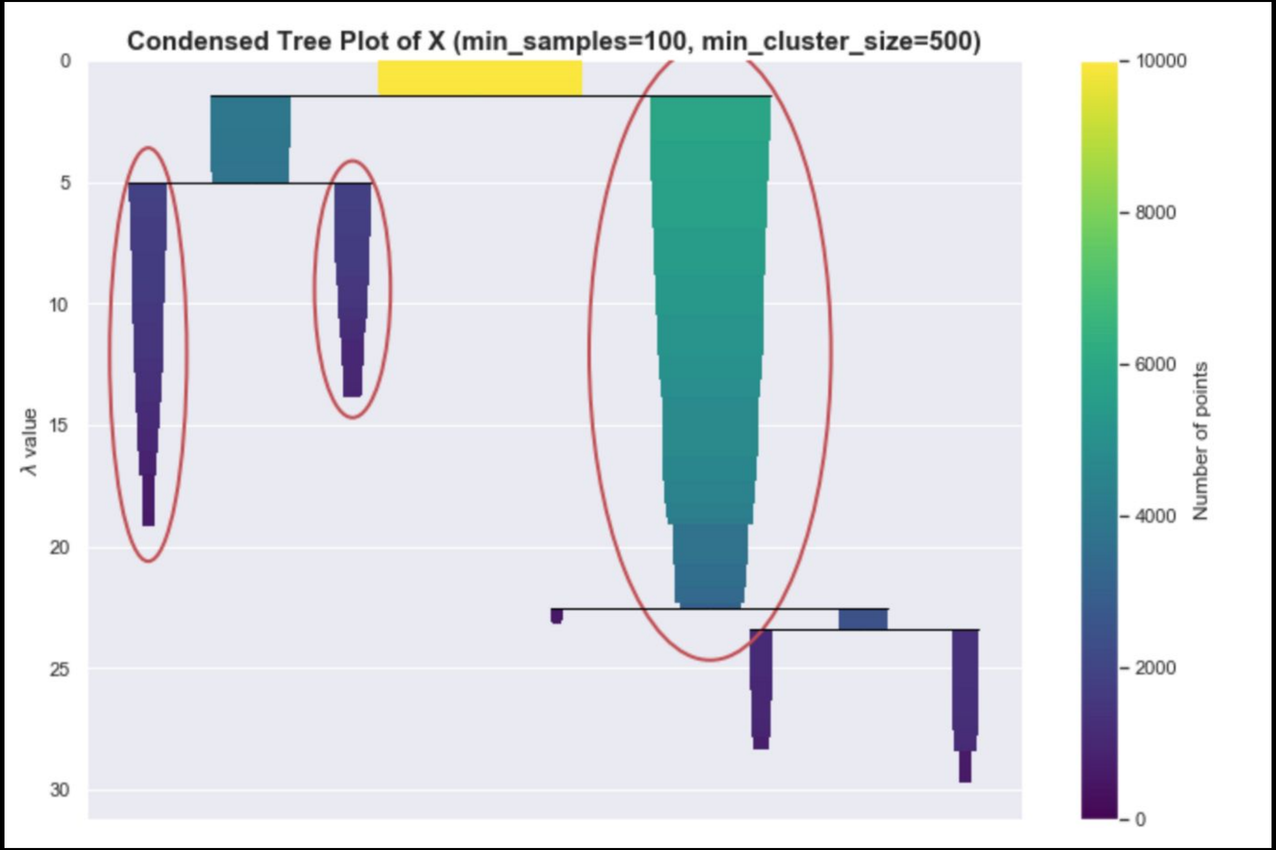
Points have to be in a high density region and close to each other (“mutual reachability”)





clusters are more likely regions separated by less likely regions -> densities

Hierarchical clustering



A hierarchy of multiple level-sets is obtained by varying the density threshold

Visualization of the tree top-down as in the literature

hdbscan vs K-means

K-means is a semi-supervised parametric algorithm parameterized by the *K cluster centroids* (aka K seeds). Can perform if the underlying assumptions on the shape of the clusters are not met. Clusters have to be:

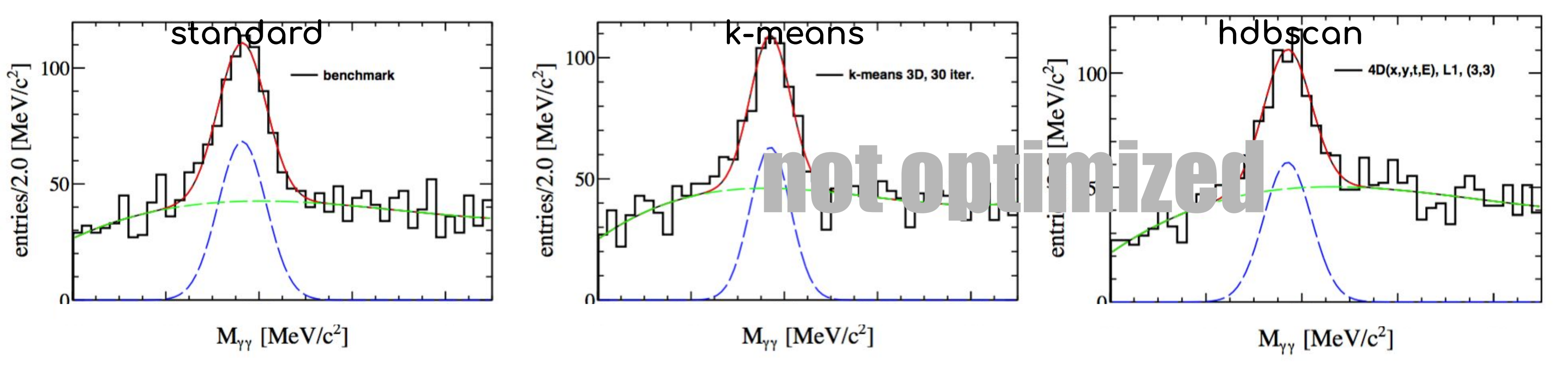
- “round” or “spherical”
- equally sized, dense
- typically most dense in the center
- not contaminated by noise and outliers

Hdbscan on the other hand is an unsupervised hierarchical clustering which excels when data has:

- arbitrarily shaped clusters
- clusters with different sizes and densities
- noise

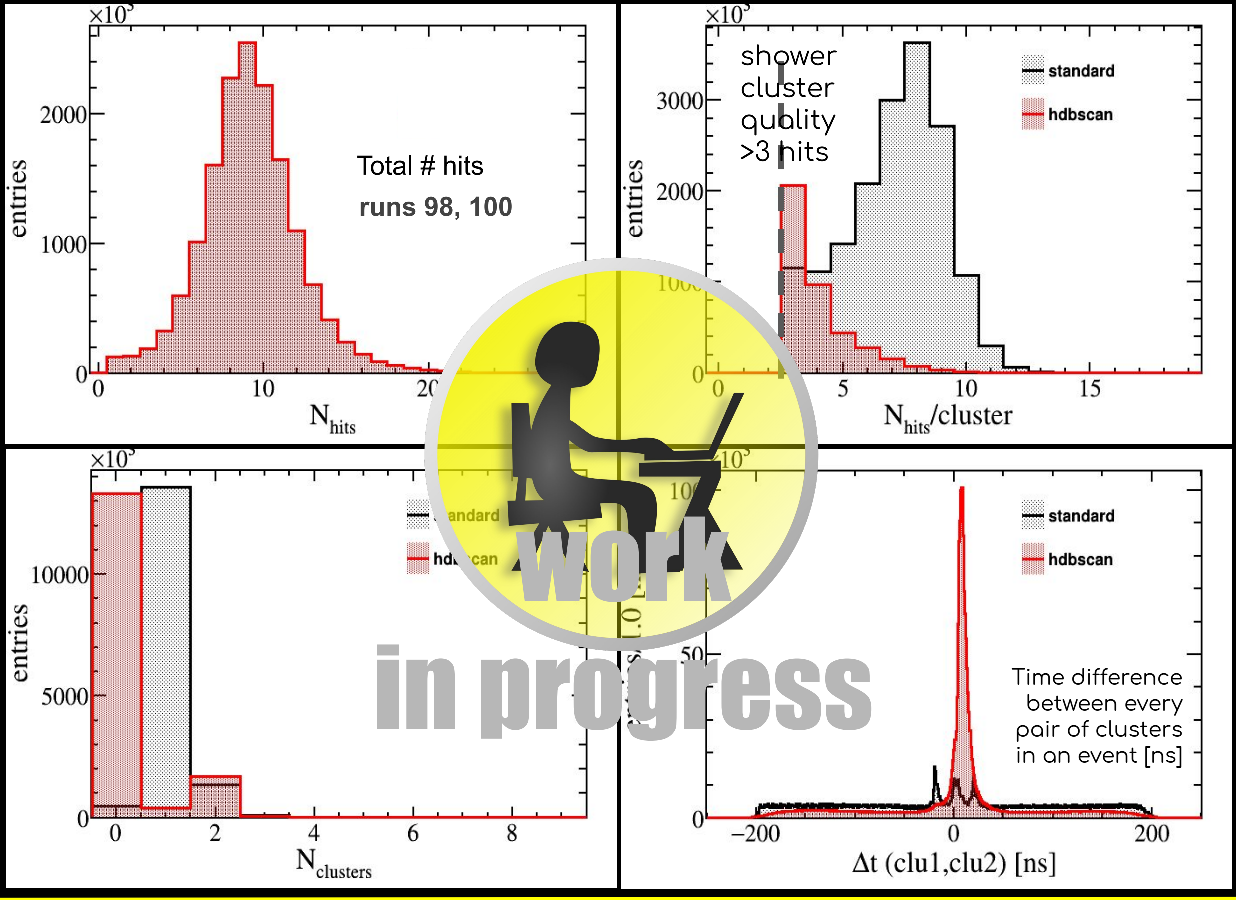
Offline tests on triggered data

- Implementation of AI-algorithms as plugins in the JANA2 reconstruction framework tested on real data (reconstruction of π^0 peak): results below are not optimized.
- Main ingredients: define a metric (N-dim) for the distance and the hyperparameters (two main (only) for hdbscan).
  - For unsupervised clustering need only few hyperparameters and no other cuts.
  - How do you optimize? Look at ‘candles’ in the calorimeter.



- Everything shown can work for online reconstruction in SRO. For data exploration we want a clustering algorithm with as few assumptions as possible so that the preliminary insights are useful!

Clustering with data taken in SRO

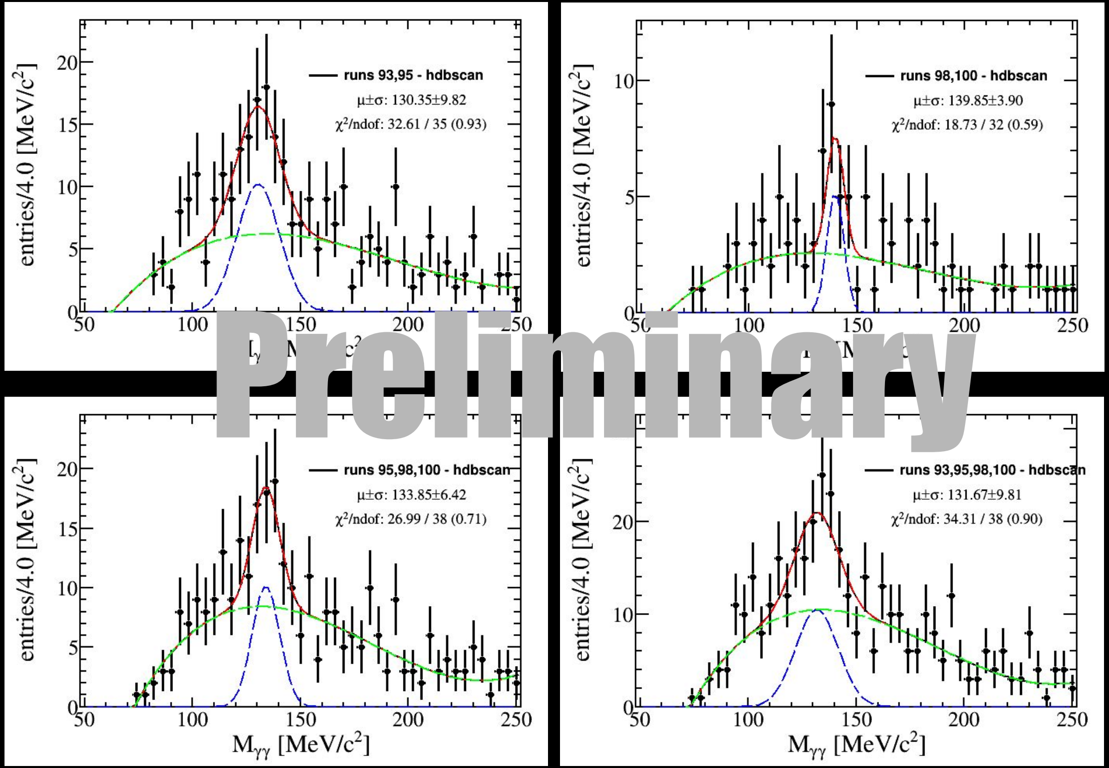


- SRO tests (TriDAS build a physics event in a time window of 400 ns if L1 finds > 2 GeV in the FTCal) with several physics triggers in parallel with a scaler that saves any data

run	calo (ON)	L2 config.
93	all	At least 1 cluster with E>3 GeV
95	all	At least 2 clusters with E>3 GeV
98	1/2	At least 3 clusters with E>2 GeV
100	all	At least 3 clusters with E>2 GeV

- hdbscan looks at the local density in the entire (x,y,t,E) space.
- Work in progress/under investigation:
  - Could reconstruct clusters for a fraction of data. Recall hdbscan can “filter” noise hits though.
  - Standard clustering seems to aggregate around the first seed most of the hits and at present is performing poorly [?].
  - Need to look at topologies of clusters event by event.

Preliminary results with SRO data



SRO tests with several physics triggers in parallel with a scaler that saves any data

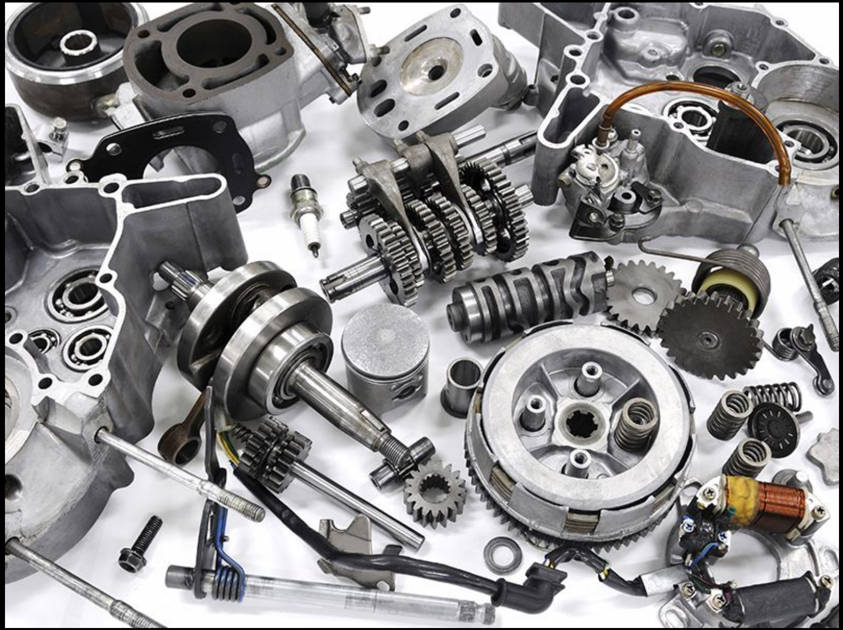
run	calo (ON)	L2 config.
93	all	At least 1 cluster with E>3 GeV
95	all	At least 2 clusters with E>3 GeV
98	1/2	At least 3 clusters with E>2 GeV
100	all	At least 3 clusters with E>2 GeV

SELECTION
d(clu1,clu2) >2 cell diag. & E(clu1)>3 GeV & E(clu2)>3 GeV & nclus≥2; Clustering pars: (3,3), Δt(clu1,clu2)<50 ns

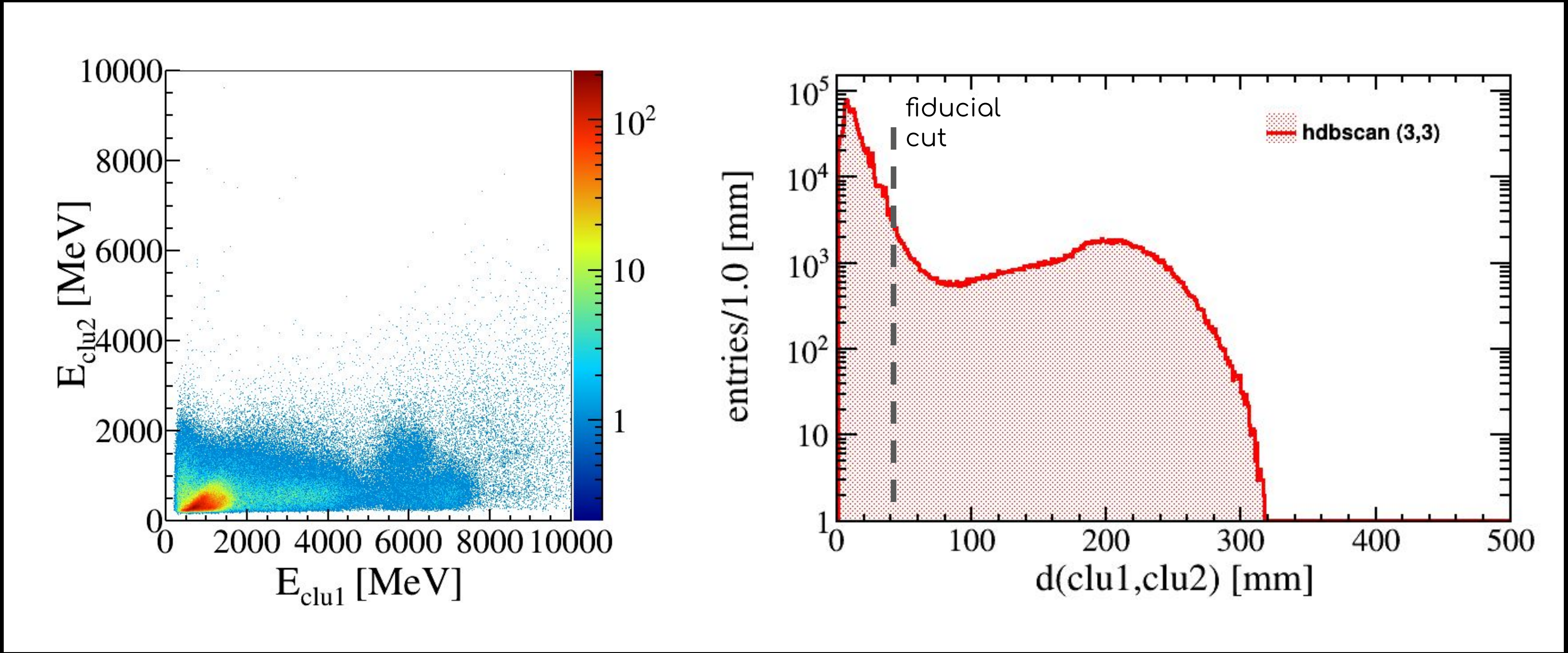
Conclusions

- AI-based algorithms have been developed within JANA framework and can run online.
  - Hierarchical clustering is an elegant and practical approach to deal with clustering introducing just few hyperparameters (which have not been optimized yet); supports different metrics.
  - It recovers the main limits of K-means (e.g., cluster shape, noise environment, etc.).
  - Nice additional features like determination of “outlier” and “membership” scores of hits which can further refine the clustering (not used yet).
- AI seems a “natural” approach to clustering in streaming readout:
  - Promising (though preliminary) results using data taken on Feb 2020 for the FT-Cal of CLAS12 in SRO show presence of π^0 candidates.
  - Ongoing work to consolidate/interpret these results:
    - Understand if we are losing events and/or there were issues with the physics trigger (based on standard algorithm).
    - Need to look in more detail at event topologies on an event by event basis.
    - Comparison to expected yields is underway.

SPARES



Preliminary results with SRO data



hdbscan (3,3)
Runs 98 & 100

Relative Performance k-means and hdbscan