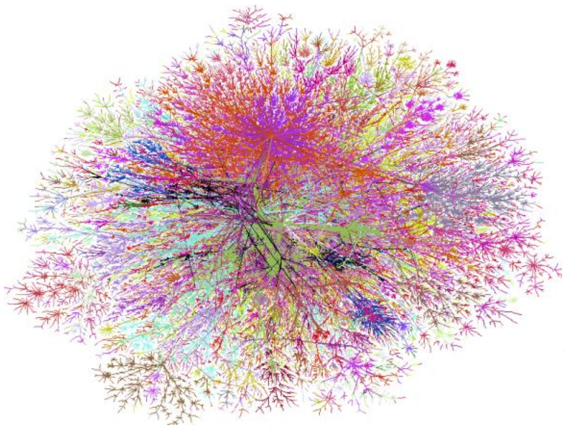
Scalable Clustering with Graph Neural Networks using DGL

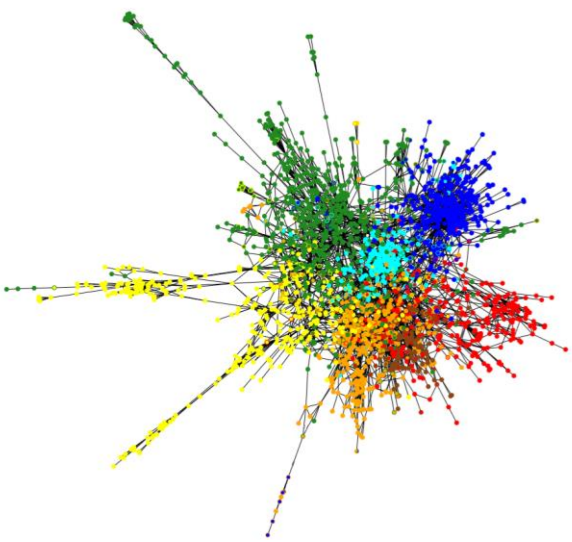
NICHOLAS CHOMA, JOAN BRUNA

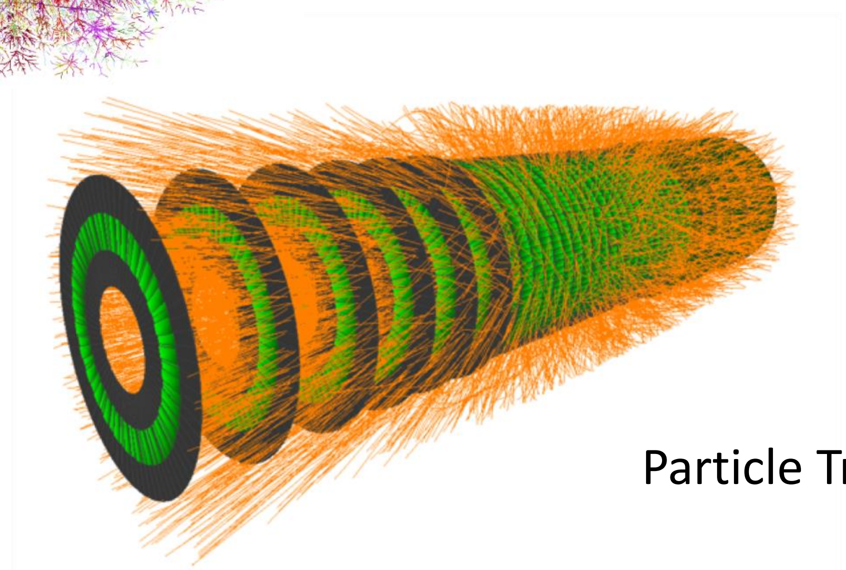
Clustering with GNNs



Sensor / Social Networks

Citation Networks

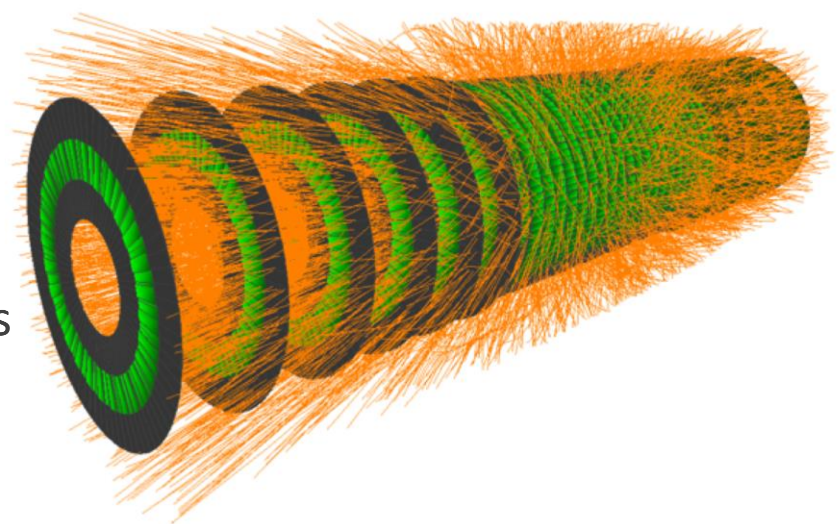




Particle Tracking

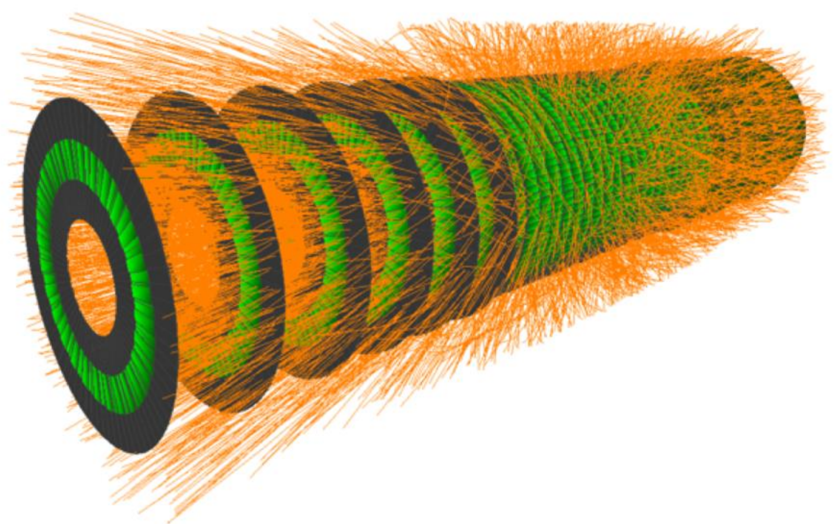
Particle Tracking

- Proton bunches circulate at LHC and collide at high energy
- Each collision produces many new particles, which spread outward in a shower
- To identify the types of particles and their kinematic properties, an applied magnetic field bends their trajectories
- These particles are recorded having passed through the detector cells
- From the recorded hits, the goal is to cluster them such that each cluster is associated with one particle



Particle Tracking

Given a set of $\approx 10^5$ points created by $\approx 10^4$ particle tracks, cluster hits such that each cluster is associated with one track.



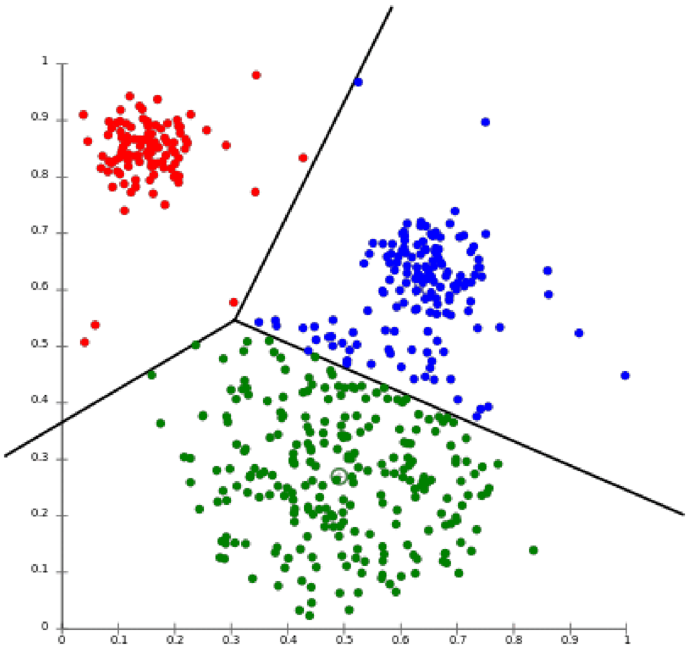
Input:
- N hits (x, y, z and detector ID)
- Detector cell pattern for each hit

Output:
- k clusters of the N hits

Challenges:
- Variable number of tracks which is not *a priori* known
- Inference must be efficient

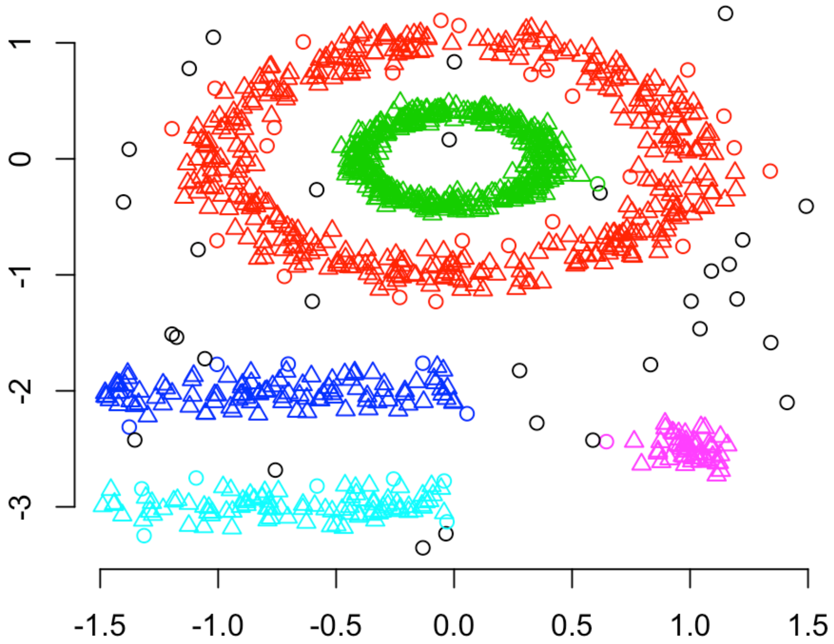
Traditional method, k-means

- **Input:** $X = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$
- **Output:** k clusters of X
- **Hyperparameters**
  - k, the number of clusters
  - i, the number of iterations for convergence
- **Algorithm**
  1. **Assign** points to current cluster centroids
  2. **Update** cluster centroids from assigned points
- **Time complexity**
  - $O(ndki)$



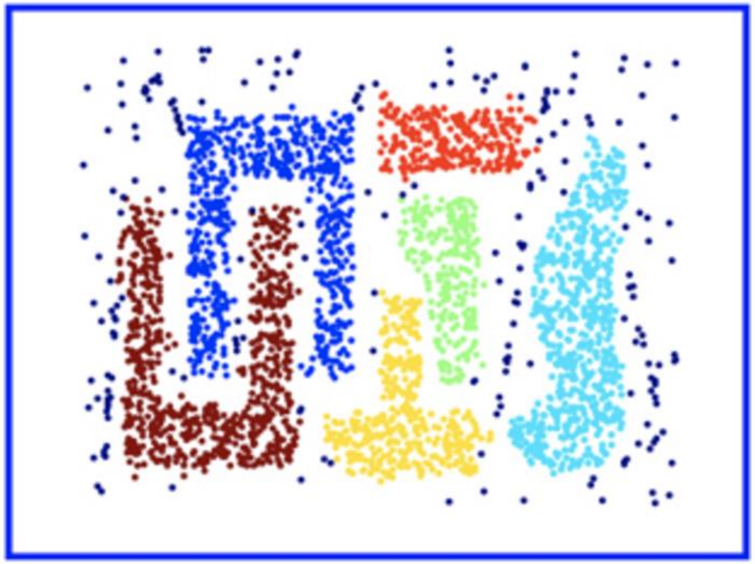
Traditional method, DBSCAN

- **Input:** $X = \{x_1, \dots, x_n\}, x_i \in \mathbb{R}^d$
- **Output:** k clusters of X
- **Hyperparameters**
  - $\boldsymbol{\epsilon}$, the maximum distance between two points to be considered within the same neighborhood
  - $\boldsymbol{m}$, the minimum number of points per cluster
- Related to spectral clustering
  - But no need to pre-specify number of clusters



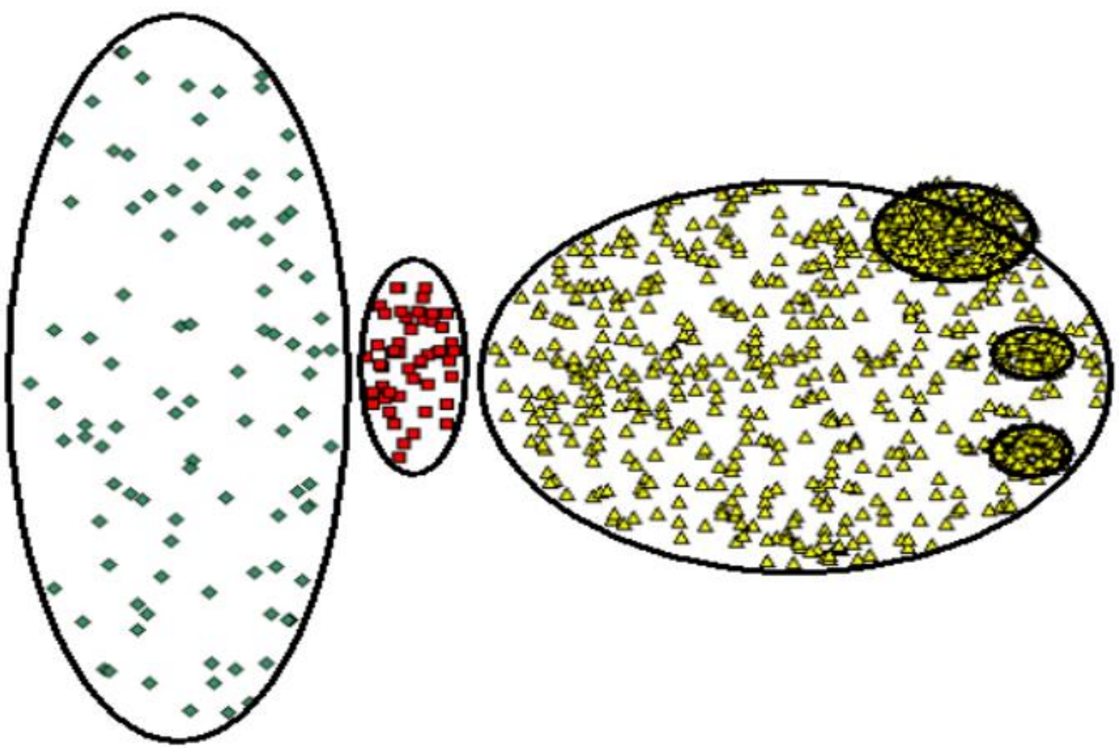
k-means vs. DBSCAN

- DBSCAN succeeds where k-means fails

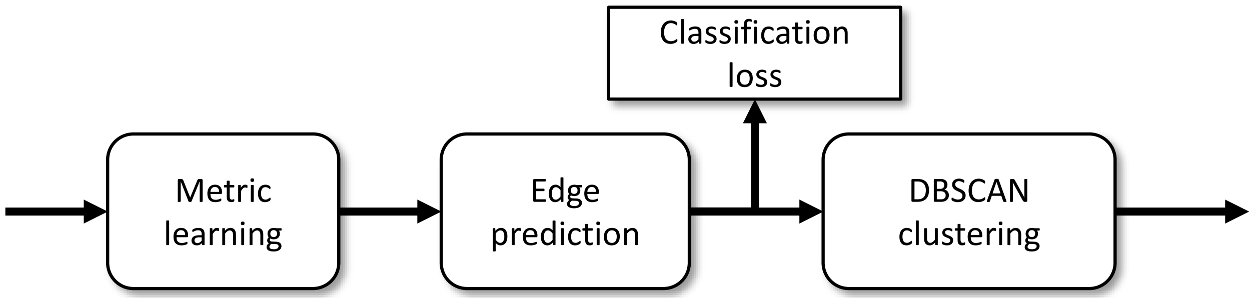


k-means vs. DBSCAN

- DBSCAN fails when cluster density varies drastically



Clustering with GNNs

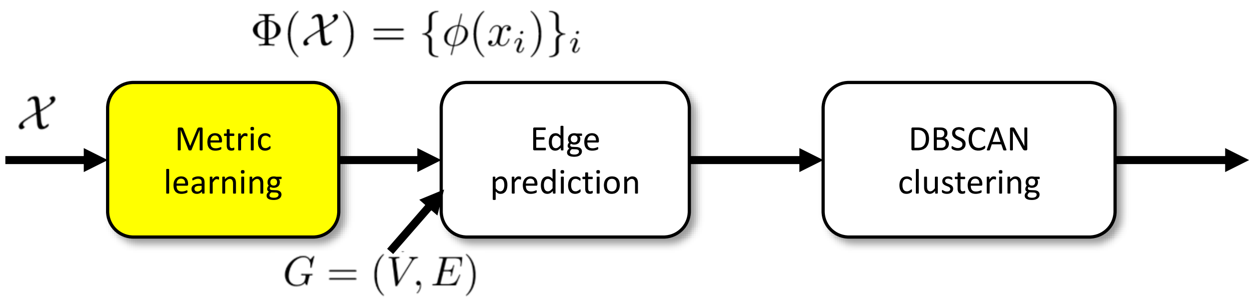


- Graph neural network model is trained in stages

1. Construct graph by pre-selecting potential hit pairs, which become edges (hits are vertices)
2. Embed hits using proxy goal of classifying whether hit pairs belong to same track
3. Cluster embedded hits into tracks

Construct Graph

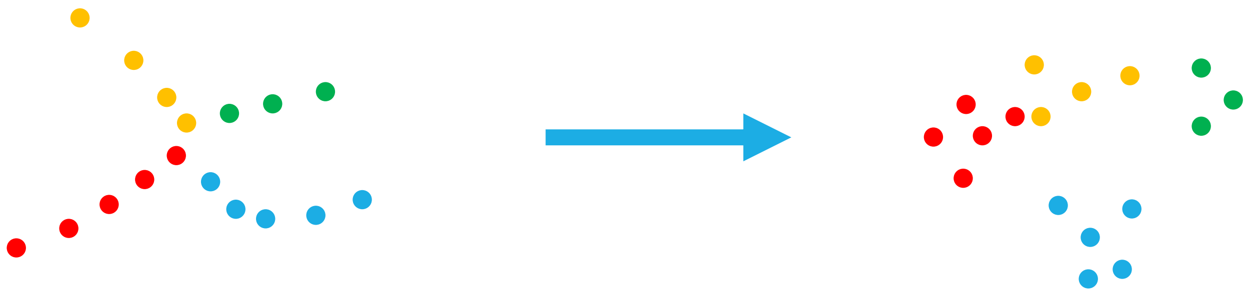
Construct graph by pre-selecting potential hit pairs, which become edges



- Relevant hit pairs are selected in stages
  1. Embed hits into new space with Euclidean distance metric
  2. Build k-d tree using embedded hits
  3. Find all nearby hits within ϵ-neighborhood

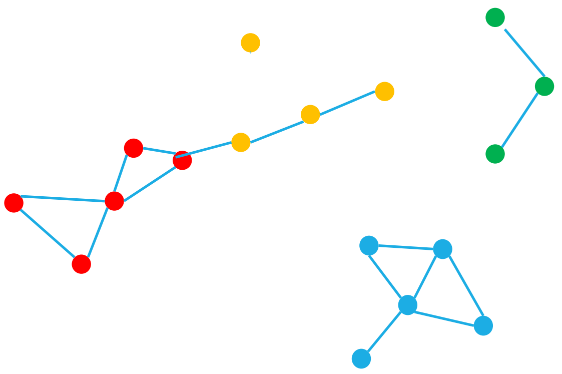
Construct Graph

- Hits are embedded from original feature space to new space with Euclidean distance metric
  - Hits belonging to same track are nearby
  - Hits belonging to different tracks are far apart



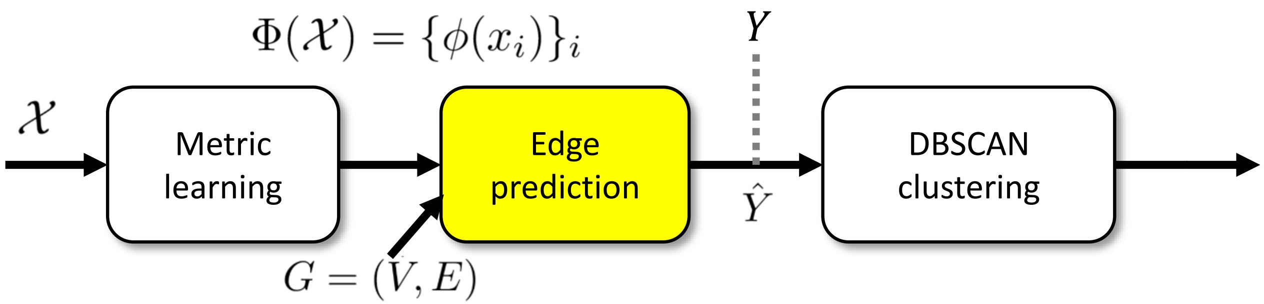
Construct Graph

- From embedded hits, construct k-d tree with Euclidean distance
  - Efficient querying for fast graph construction
- Construct graph by finding hits within ϵ-neighborhood of each hit



GNN Edge prediction

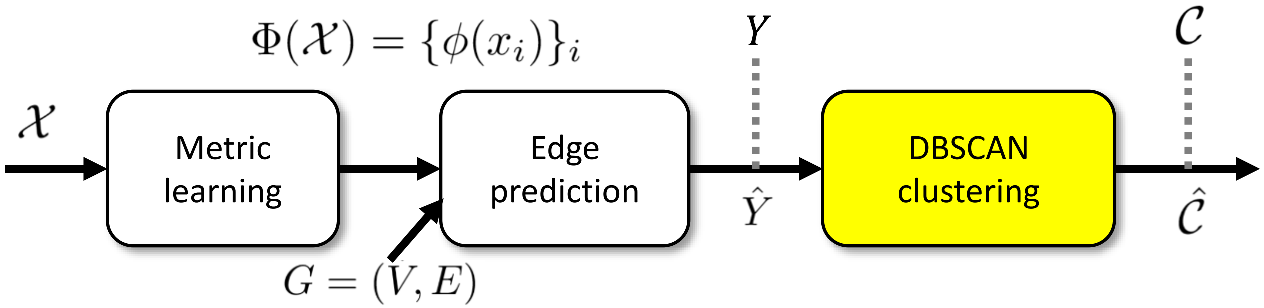
Embed hits by classifying whether hit pairs belong to same track



- Train graph neural network (GNN) model with sparse graph constructed by metric learning stage
- GNN improves hit embedding by using information from each hit's local neighborhood
- Model is a message-passing GNN, where each layer re-computes weighted edges

Clustering

Cluster embedded hits into tracks



- Run GNN-embedded hits through DBSCAN algorithm, fine-tuned for optimal ϵ-neighborhood and minimum cluster size
- Evaluate final clusters (tracks) on TrackML scoring funciton