

# Context

This doc intends to capture the exact definitions of all metrics used to quantify the performance of the GNN tracking workflow. It is primarily to standardize comparisons across the Princeton/UIUC collaboration, but can also be shared with other groups (eg ExaTrkX).

Google docs is likely not the best place to record this information, but I'll start here.

Some additional measurements (degree distributions, clustering coefficients):

<http://web.stanford.edu/class/cs224w/slides/02-gnp-smallworld.pdf>

# Graph Construction

## Graph Sizes

**Description:** Number of hits, number of segments, and size (in MB) per pt cut

**Definition:**  $n\_nodes = X.shape[0]$ ,  $n\_segments = Ri.shape[i]$ ,  
 $size = \sum[\text{sys.getsizeof}(i)]/10^{**6}$  for  $i = \{X, Ri, Ro, y, Ra\}$

**Pointer to implementation:**

- Number of hits and segments in each graph:  
[https://github.com/savvy379/princeton\\_gnn\\_tracking/blob/master/analyses/prepare\\_measurements.py#L49-L54](https://github.com/savvy379/princeton_gnn_tracking/blob/master/analyses/prepare_measurements.py#L49-L54)
- Total size (in MB) of each graph:  
[https://github.com/savvy379/princeton\\_gnn\\_tracking/blob/master/analyses/prepare\\_measurements.py#L101-L105](https://github.com/savvy379/princeton_gnn_tracking/blob/master/analyses/prepare_measurements.py#L101-L105)

## Degree Distribution

**Description:** Probability that a randomly chosen node has degree  $k$

**Definition:**  $P(k) = N_k/k$  ( $N_k = \#$  nodes with degree  $k$ )

**Pointer to implementation:** not yet implemented

## Network Diameter

**Description:** The maximum distance between any pair of nodes in a graph

**Definition:** distance=shortest path between two nodes, exclude non-connected pairs

**Pointer to implementation:** not yet implemented

## Clustering Coefficient

**Description:** Measures how connected node  $i$ 's neighbors are to each other

**Definition:**  $C_i = (2e_i) / (k_i(k_i - 1))$  for  $e_i$  = number of edges between neighbors of node  $i$ ,  $k_i$  = degree of node  $i$

**Pointer to implementation:** not yet implemented

## Connectivity

**Description:** Size of the largest connected component of the graph

**Definition:** Start from random node and perform BFS, label nodes that BFS visits, if all nodes are visited, network is fully connected, otherwise find an unvisited node and repeat.

**Pointer to implementation:** not yet implemented

## Graph Efficiencies

**Description:** Measure of how many false edges are created by the graph construction process and what percentage of the edges that should exist are created by the graph construction process. \*Note:\* you can also break this into multiple steps (ie raw truth graph, with pt and duplicate hits cut, after segment construction)

**Definition:** True edges captured in the graph/all edges in the graph and true edges captured in the graph/all true edges that exist in the actual event file

**Pointer to Implementation:**

[https://github.com/savvy379/gnn\\_track\\_challenge/blob/master/measurements/graph\\_efficiency.py](https://github.com/savvy379/gnn_track_challenge/blob/master/measurements/graph_efficiency.py)

## Graph Networks

### Edge Classification Efficiency:

**Description:** The values of and a plot of the confusion matrix parameters (true positive, false positives, true negatives, false negatives) as a function of discriminant cut. Can also present as a ROC curve (true positive vs false positive as discriminant cut changes).

**Definition:**

**Pointer to implementation:**

[https://github.com/GageDeZoort/interaction\\_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot\\_menu.py#L171](https://github.com/GageDeZoort/interaction_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot_menu.py#L171)

## Training Loss

**Description:** A plot of the overall network loss function by epoch for both training and test sets overlaid. Can have variations such as loss by batch.

**Definition:** There are two loss functions currently used by the group

- Root Mean Squared Error
- Weighted Binary Cross Entropy (weighted by real and fake weights, defined by portion of real edges in graph)

**Pointer to implementation:**

Unweighted BCE: [https://github.com/GageDeZoort/interaction\\_network/blob/master/train\\_IN.py](https://github.com/GageDeZoort/interaction_network/blob/master/train_IN.py)

Weighted BCE:

<https://github.com/savvy379/heptrkx-gnn-tracking/blob/c622ed038244c12915c784109b1088dbffb64b53/trainers/gnn.py#L46>

## Edge Weights

**Description:** A histogram distribution of the edge weights for all graphs in the test set. Separate histograms for real and false edges.

**Definition:**

**Pointer to implementation:**

[https://github.com/GageDeZoort/interaction\\_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot\\_menu.py#L133](https://github.com/GageDeZoort/interaction_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot_menu.py#L133)

## Track Finding

## Visualizations

### Draw Input Graph

**Description:**

**Definition:**

**Pointer to Implementation:**

### Draw Processed Graph

**Description:** Draw the a graph after it's been processed through a GNN with the True Positives in green, False Negatives in blue, and False Positives in red

**Definition:**

**Pointer to Implementation:**

[https://github.com/GageDeZoort/interaction\\_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot\\_menu.py#L52](https://github.com/GageDeZoort/interaction_network/blob/78c3f7d95a3bd52f30bc04b59035d5bf81dc700b/plots/plot_menu.py#L52)

Other