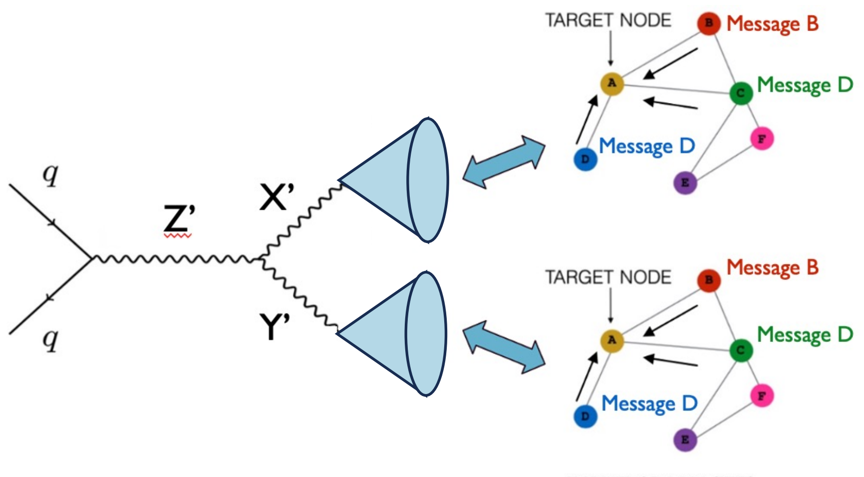
Analysis framework status report

Antonio D’Avanzo, DBL fully-hadronic anomaly detection meeting, 31/05/2024

The analysis

- **Anomaly Detection** in fully hadronic events with message passing based Graph Neural Netwoks (GNNs).
- Graphs representing the final states jets, the 2 pT leading jets per event, built from transformed constituents.
- Graph features contain **[pT frac, η and ϕ]** of the constituents as node features, nodes connected if ΔR < 0.2 with **1/ΔR** as the edge feature (dataset with ΔR < 0.1 also available).



- **Final goal:** Run 3 fully hadronic search
  - Completely model agnostic, 2 large-R jets per event
  - Signal region based on Anomaly Score cut.

toy model

- R&D LHC Olympics dataset
  - $Z' \longrightarrow XY \longrightarrow qqqq$ events
  - $m_{W'} = 3.5\text{TeV}$, $m_X = 500\text{GeV}$, $m_Y = 100\text{GeV}$
  - reconstructed with anti-k_T with $R = 1.0$

benchmarks

- ML tasks
  - **classification approach:** RF, transformers, GNN for graph-classification
  - **AD approach:** GNNs, transformer+AE

Nutple framework

- Production of ntuples from our run 3 LLJ1 DxAOD based on EasyJet framework.

- **News**:
  - Produced ntuple for data22, ~100k events.
  - Increased trigger list with new largeR-jet items, for both 2022 and 2023.
    - 2 items give problems with MC, can be commentated.

trigger list 2022

```
'2022':
- 'HLT_j360_a10t_lcw_jes_L1J100'
- 'HLT_j420_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j460_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j460_a10t_lcw_jes_L1J100'
#- 'HLT_j460_a10r_L1J100'
#- 'HLT_j460_a10_lcw_subjes_L1J100'
- 'HLT_j420_35smcINF_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j420_35smcINF_a10t_lcw_jes_L1J100'
- 'HLT_mu24_ivarmedium_L1MU14FCH'
- 'HLT_mu50_L1MU14FCH'
- 'HLT_mu60_0eta105_msonly_L1MU14FCH'
- 'HLT_mu60_L1MU14FCH'
- 'HLT_mu80_msonly_3layersEC_L1MU14FCH'
```

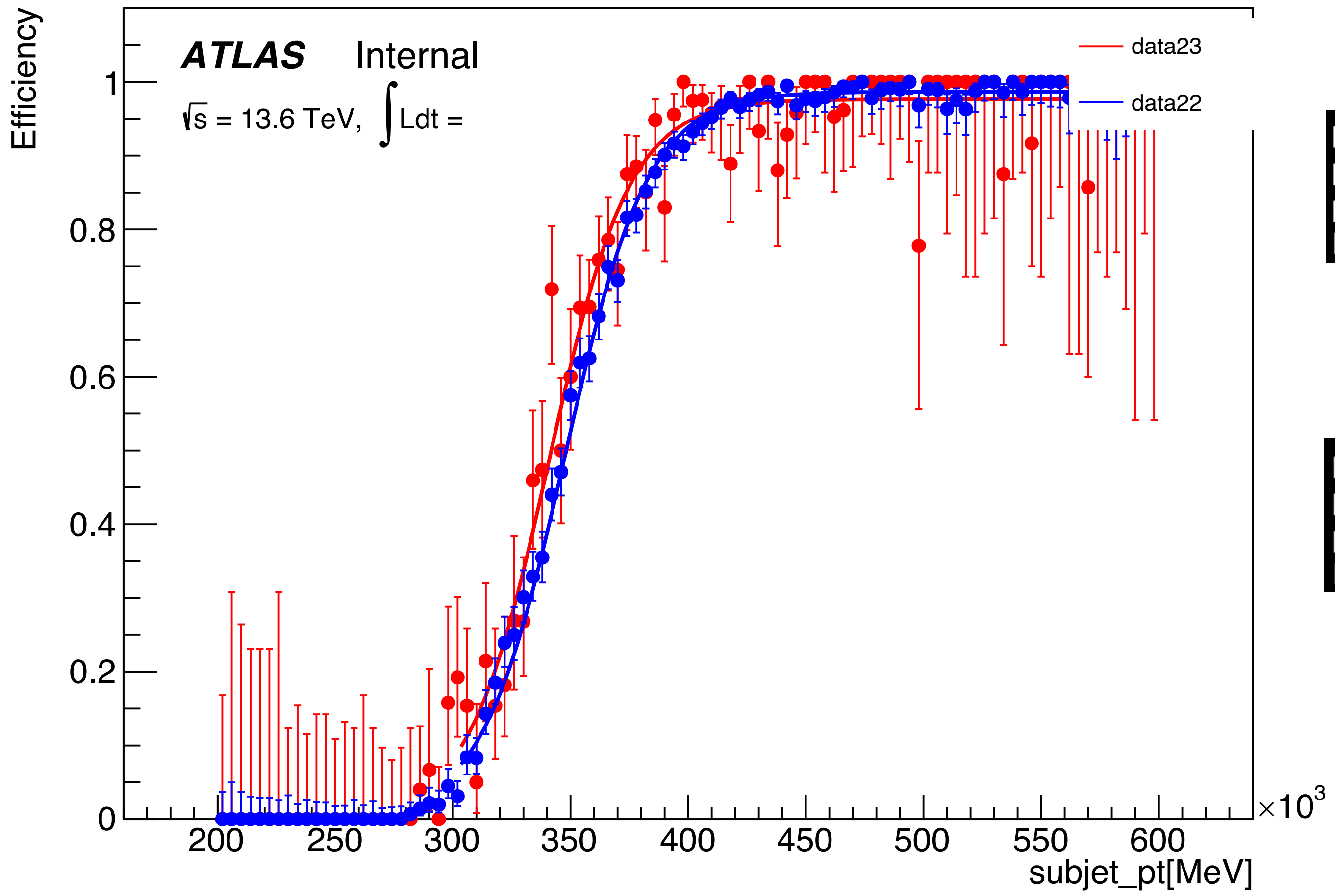
trigger list 2023

```
'2023':
- 'HLT_j360_a10t_lcw_jes_L1J100'
- 'HLT_j420_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j460_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j460_a10t_lcw_jes_L1J100'
#- 'HLT_j460_a10r_L1J100'
#- 'HLT_j460_a10_lcw_subjes_L1J100'
- 'HLT_j420_35smcINF_a10sd_cssk_pf_jes_ftf_preselj225_L1J100'
- 'HLT_j420_35smcINF_a10t_lcw_jes_L1J100'
- 'HLT_mu24_ivarmedium_L1MU14FCH'
- 'HLT_mu50_L1MU14FCH'
- 'HLT_mu60_0eta105_msonly_L1MU14FCH'
- 'HLT_mu60_L1MU14FCH'
- 'HLT_mu80_msonly_3layersEC_L1MU14FCH'
```

Unprescaled run 3 triggers twiki here

trigger study data

- Trigger item: HLT_j360_a10t_lcw_jes_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

p0	=	17117	+/-	1041.3
p1	=	340962	+/-	1502.88
p2	=	0.976353	+/-	0.00606332

Found cut off at 0.97 efficiency: 427033

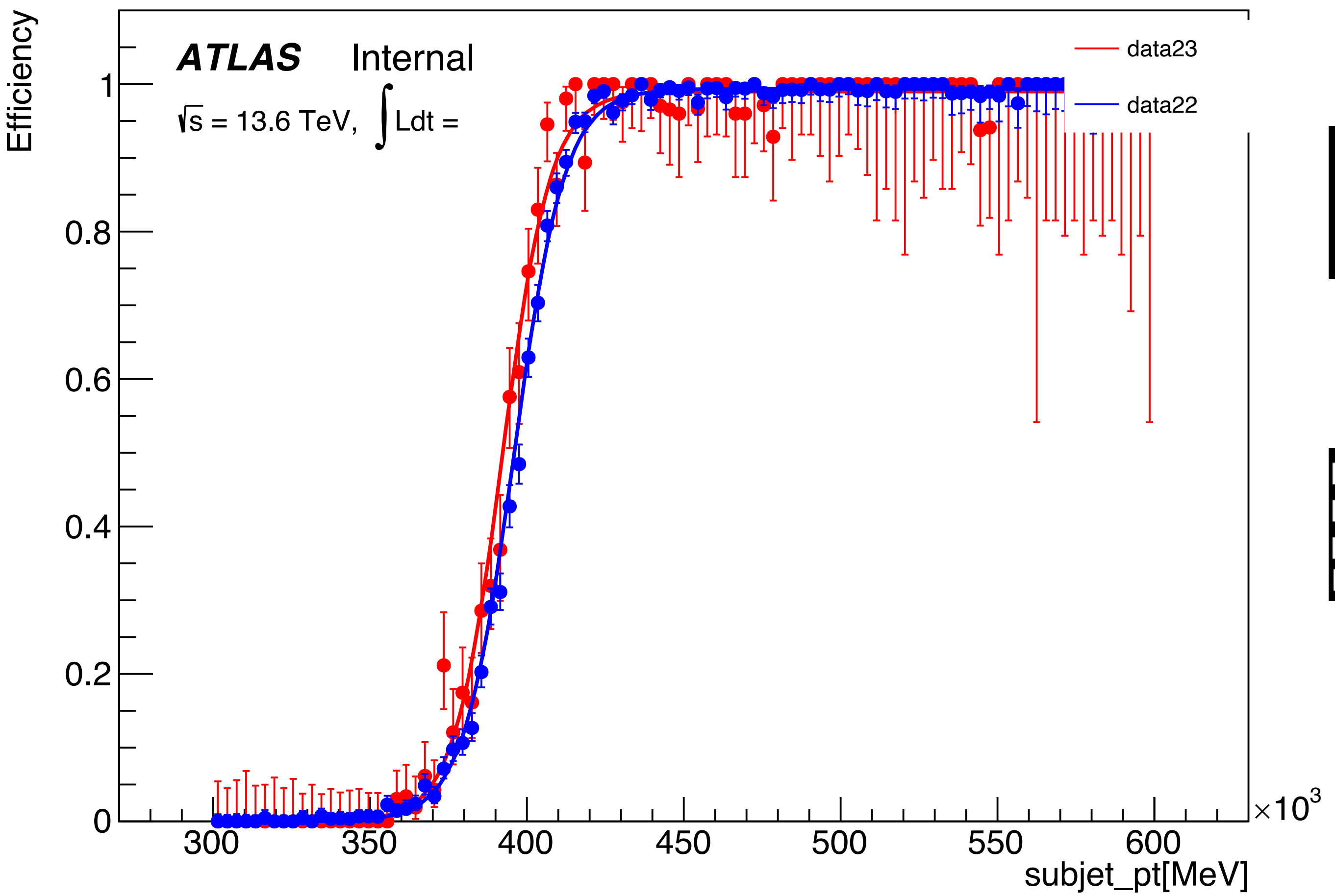
data22

p0	=	17620.7	+/-	392.2
p1	=	347667	+/-	590.338
p2	=	0.986714	+/-	0.00192562

Found cut off at 0.98 efficiency: 435477

trigger study data

- Trigger item: HLT_j420_a10sd_cssk_pf_jes_ftf_preselj225_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

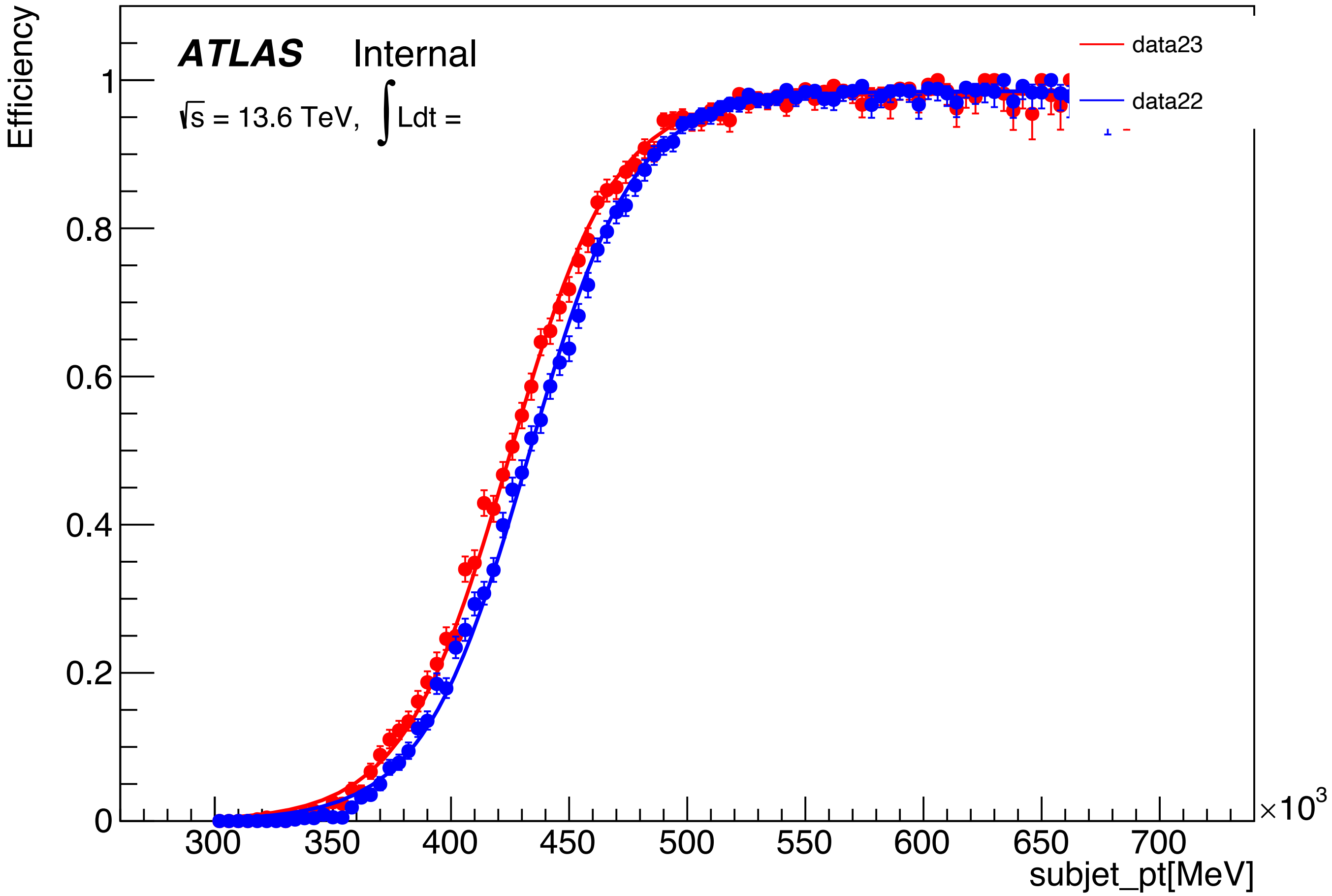
```
p0                          =      7655.11    +/-    374.917
p1                          =       392160    +/-    629.843
p2                          =     0.990003    +/-    0.00297264
Found cut off at 0.99 efficiency: 490289
```

data22

```
p0                          =       8065.4    +/-    158.41
p1                          =       395856    +/-    264.661
p2                          =     0.993721    +/-    0.000981642
Found cut off at 0.99 efficiency: 440891
```

trigger study data

- Trigger item: HLT_j460_a10r_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

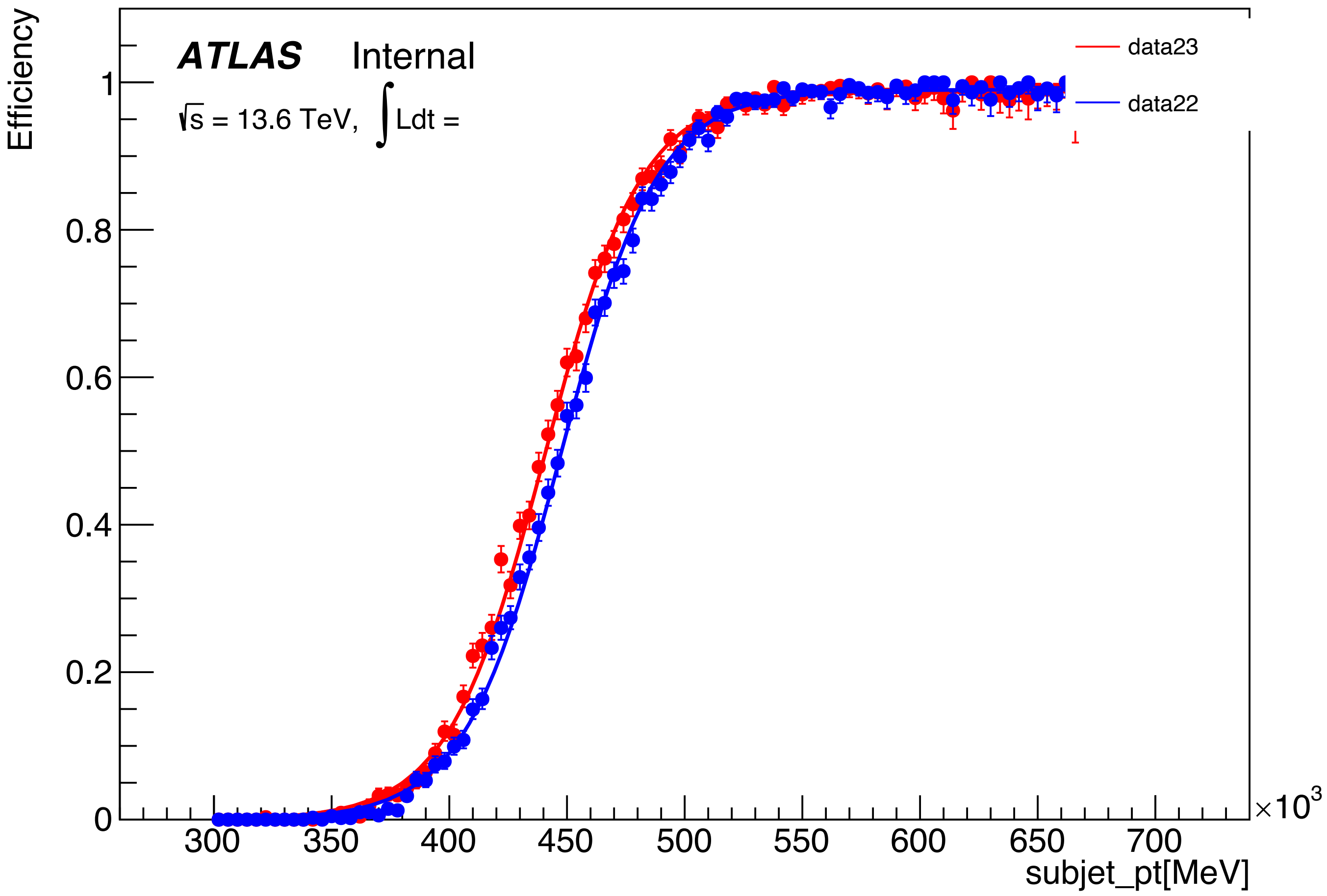
```
p0                          =      22423.4    +/-    267.65
p1                          =       424553    +/-    391.524
p2                          =     0.981706    +/-    0.00166781
Found cut off at 0.97 efficiency: 523602
```

data22

```
p0                          =      22383.8    +/-    245.315
p1                          =       432760    +/-    358.861
p2                          =     0.985058    +/-    0.00142138
Found cut off at 0.98 efficiency: 550647
```

trigger study data

- Trigger item: HLT_j460_a10t_lcw_jes_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

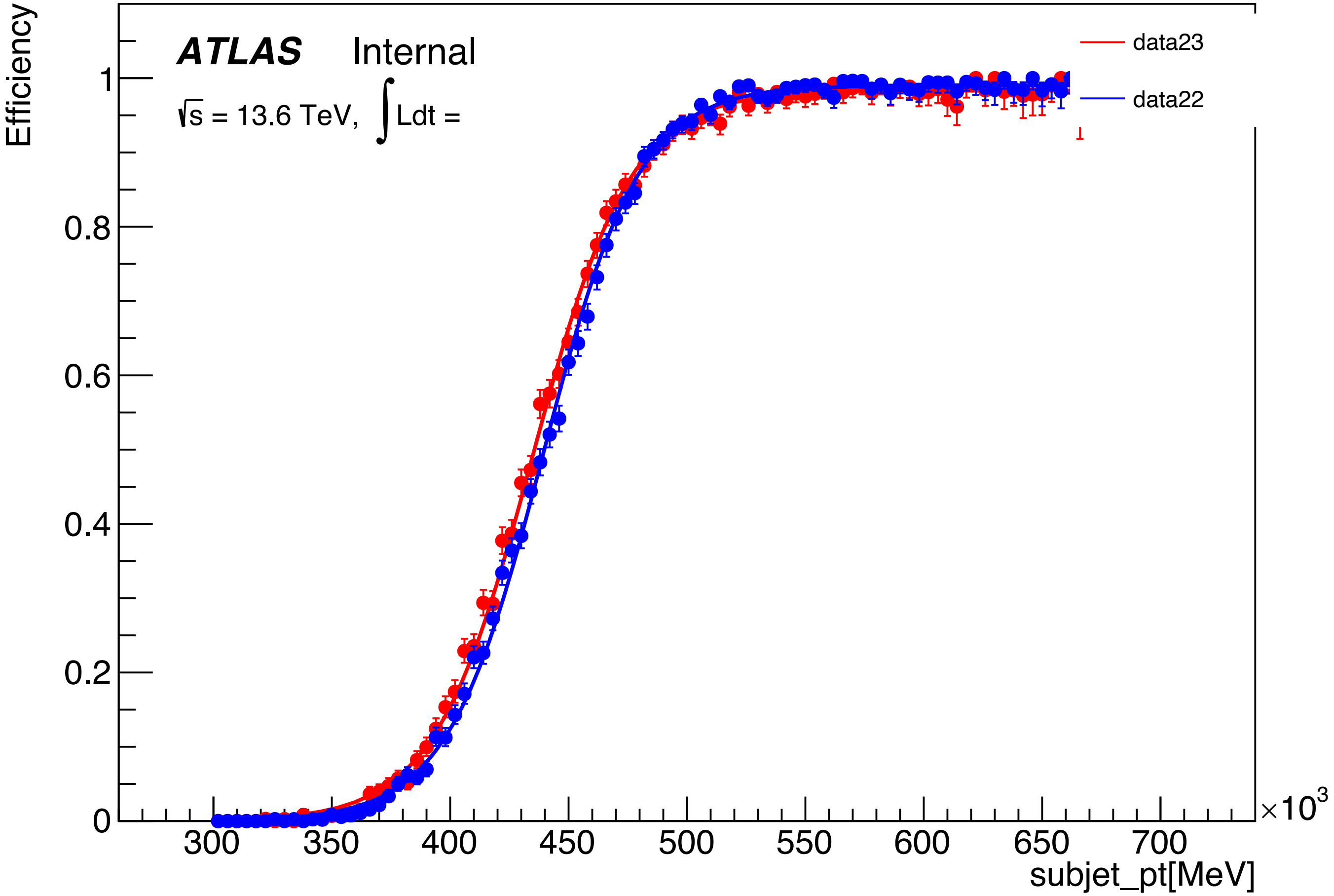
```
p0                                   =     20584.8    +/-    259.742
p1                                   =      440484    +/-    390.226
p2                                   =    0.987447    +/-    0.00150964
Found cut off at 0.98 efficiency: 540932
```

data22

```
p0                                   =     20580.3    +/-    239.521
p1                                   =      447298    +/-    363.356
p2                                   =     0.99001    +/-    0.00127226
Found cut off at 0.98 efficiency: 541637
```

trigger study data

- Trigger item: HLT_j460_a10_lcw_subjes_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

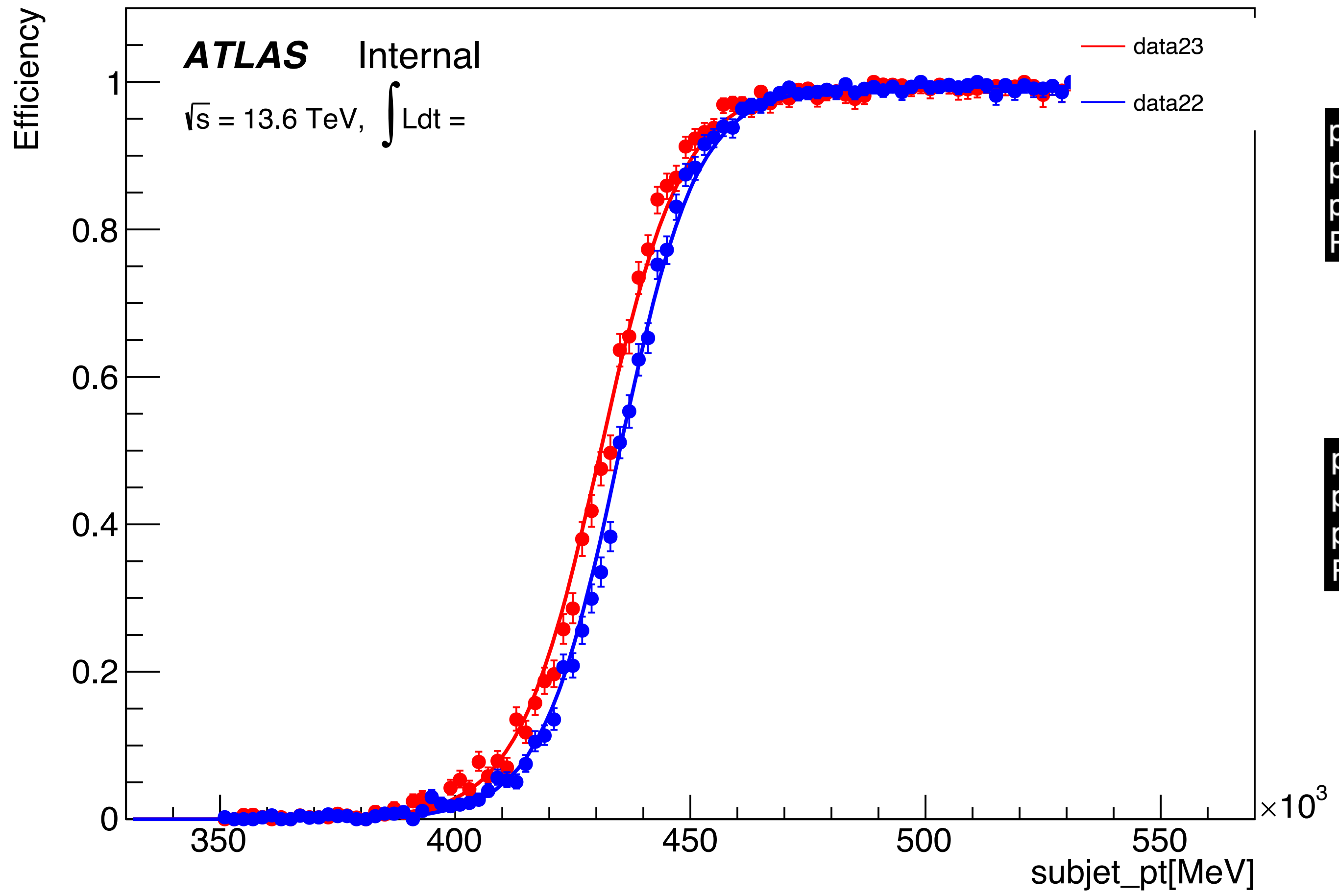
```
p0                       =     20702.3    +/-    264.489
p1                       =      434846    +/-    390.602
p2                       =    0.983288    +/-    0.00165146
Found cut off at 0.98 efficiency: 552791
```

data22

```
p0                       =     20236.8    +/-    226.398
p1                       =      439348    +/-    343.221
p2                       =    0.990331    +/-    0.00115468
Found cut off at 0.99 efficiency: 601296
```

trigger study data

- Trigger item: HLT_j460_a10sd_cssk_pf_jes_ftf_preselj225_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



data23

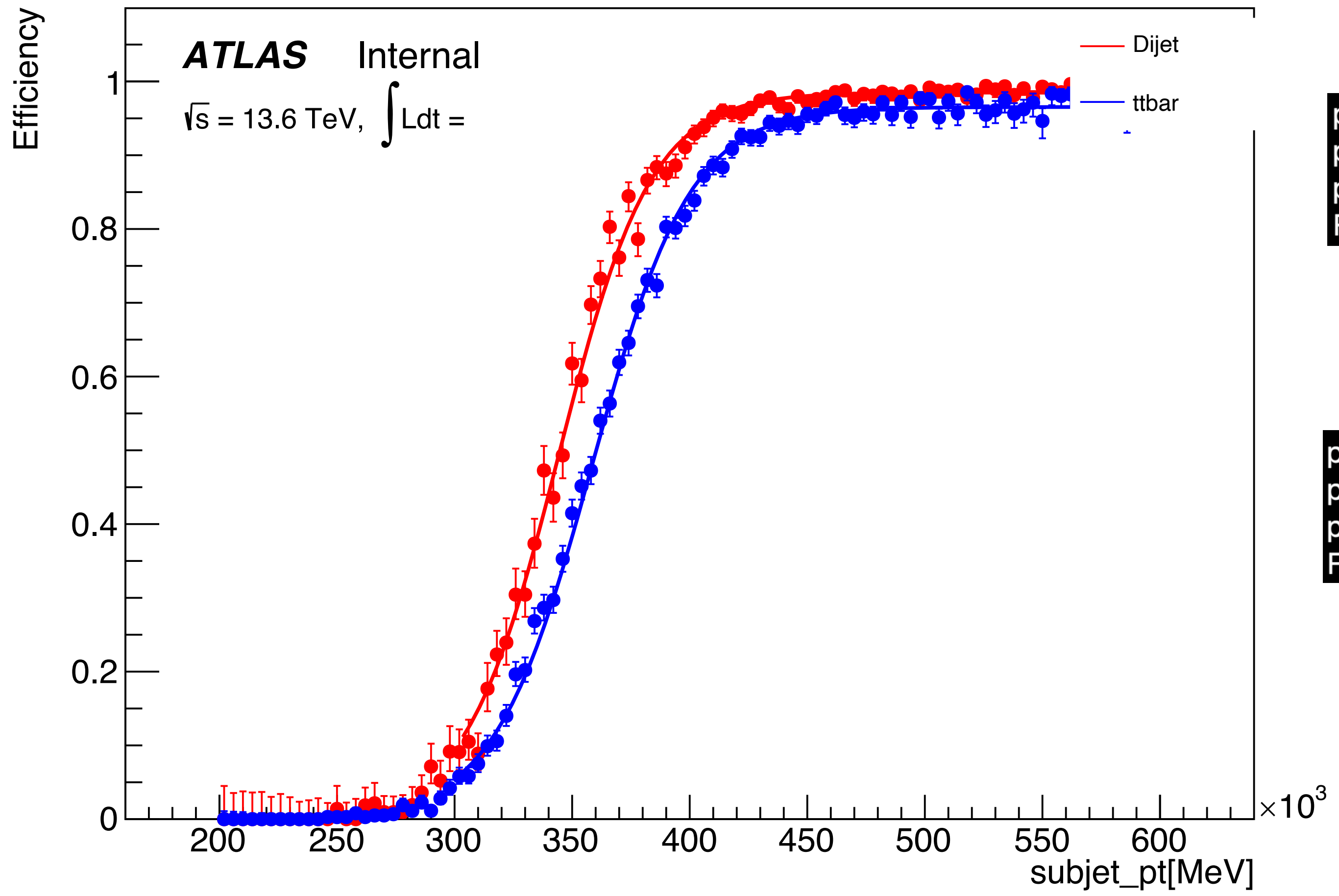
```
p0                        =      8737.24    +/-    119.947
p1                        =       430779    +/-    196.361
p2                        =     0.993001    +/-    0.000923239
Found cut off at 0.99 efficiency: 481444
```

data22

```
p0                        =      8290.88    +/-    105.462
p1                        =       434886    +/-    176.388
p2                        =     0.993693    +/-    0.000797063
Found cut off at 0.99 efficiency: 481243
```

trigger study MC

- Trigger item: HLT_j360_a10t_lcw_jes_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



Dijet

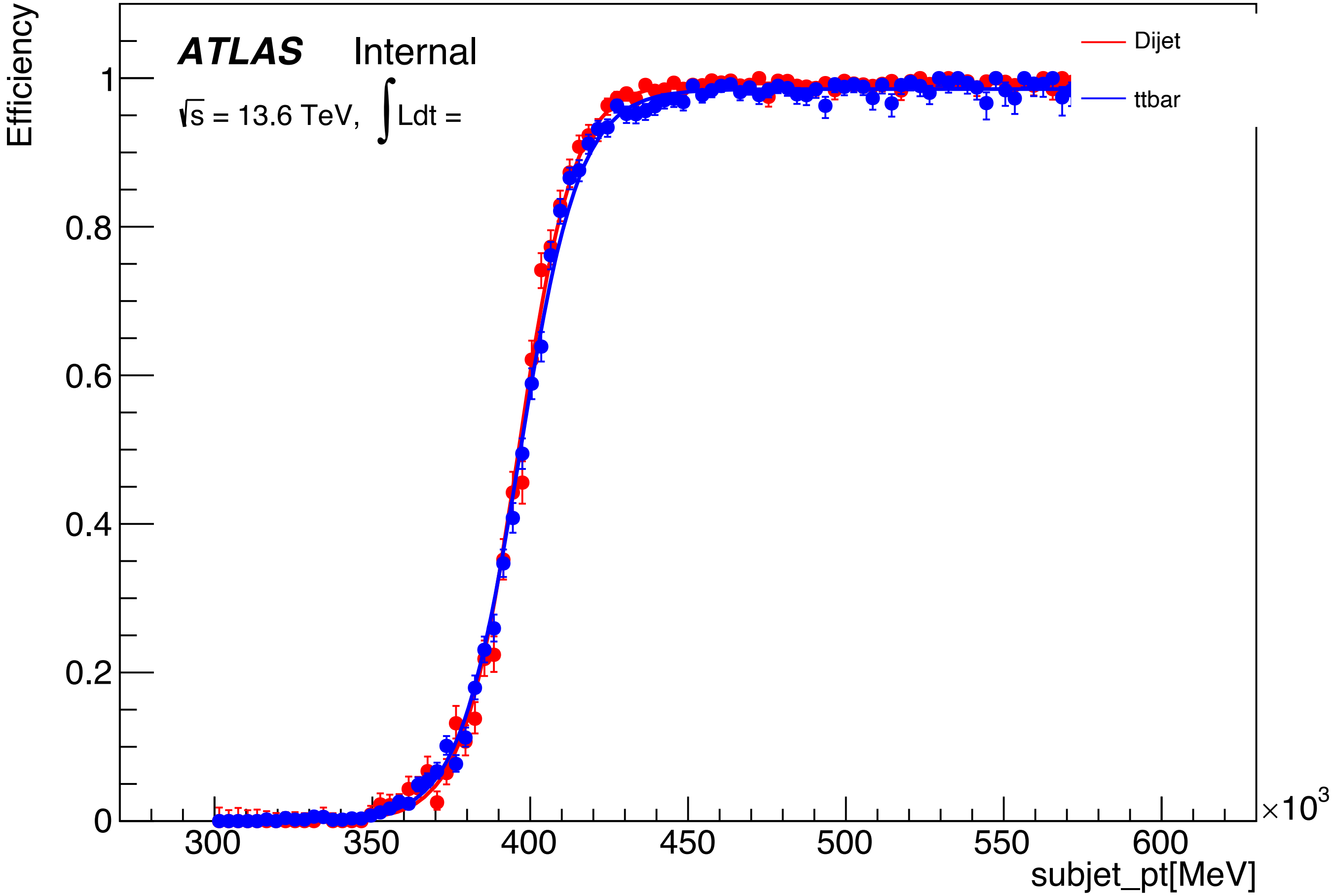
```
p0                                  =      19757.6    +/-    390.512
p1                                  =       344209    +/-    571.932
p2                                  =      0.98499    +/-    0.00111923
Found cut off at 0.98 efficiency: 448530
```

ttbar

```
p0                                  =      20831.2    +/-    267.153
p1                                  =       358738    +/-    388.853
p2                                  =     0.965405    +/-    0.00183146
Found cut off at 0.96 efficiency: 466636
```

trigger study MC

- Trigger item: HLT_j420_a10sd_cssk_pf_jes_ftf_preselj225_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



Dijet

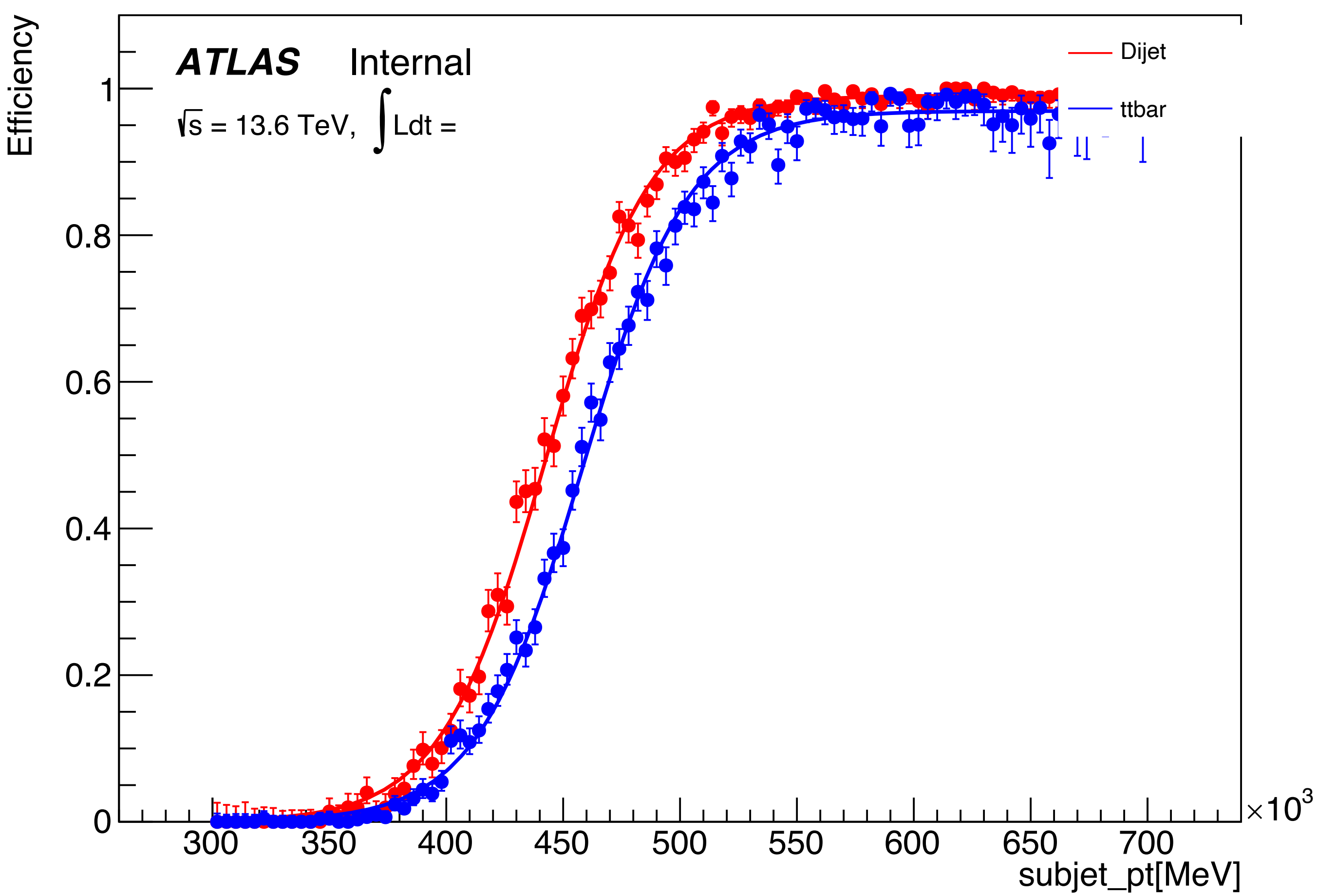
```
p0                              =      8756.98    +/-    170.396
p1                              =       396203    +/-    281.215
p2                              =     0.993021    +/-    0.000715131
Found cut off at 0.99 efficiency: 446924
```

ttbar

```
p0                              =       9527.8    +/-    137.616
p1                              =       396632    +/-    226.778
p2                              =     0.985301    +/-    0.0011313
Found cut off at 0.98 efficiency: 446364
```

trigger study MC

- Trigger item: HLT_j460_a10t_lcw_jes_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



Dijet

p0	=	22491.1	+/-	384.568
p1	=	442669	+/-	558.174
p2	=	0.990664	+/-	0.00121457

Found cut off at 0.98 efficiency: 544344

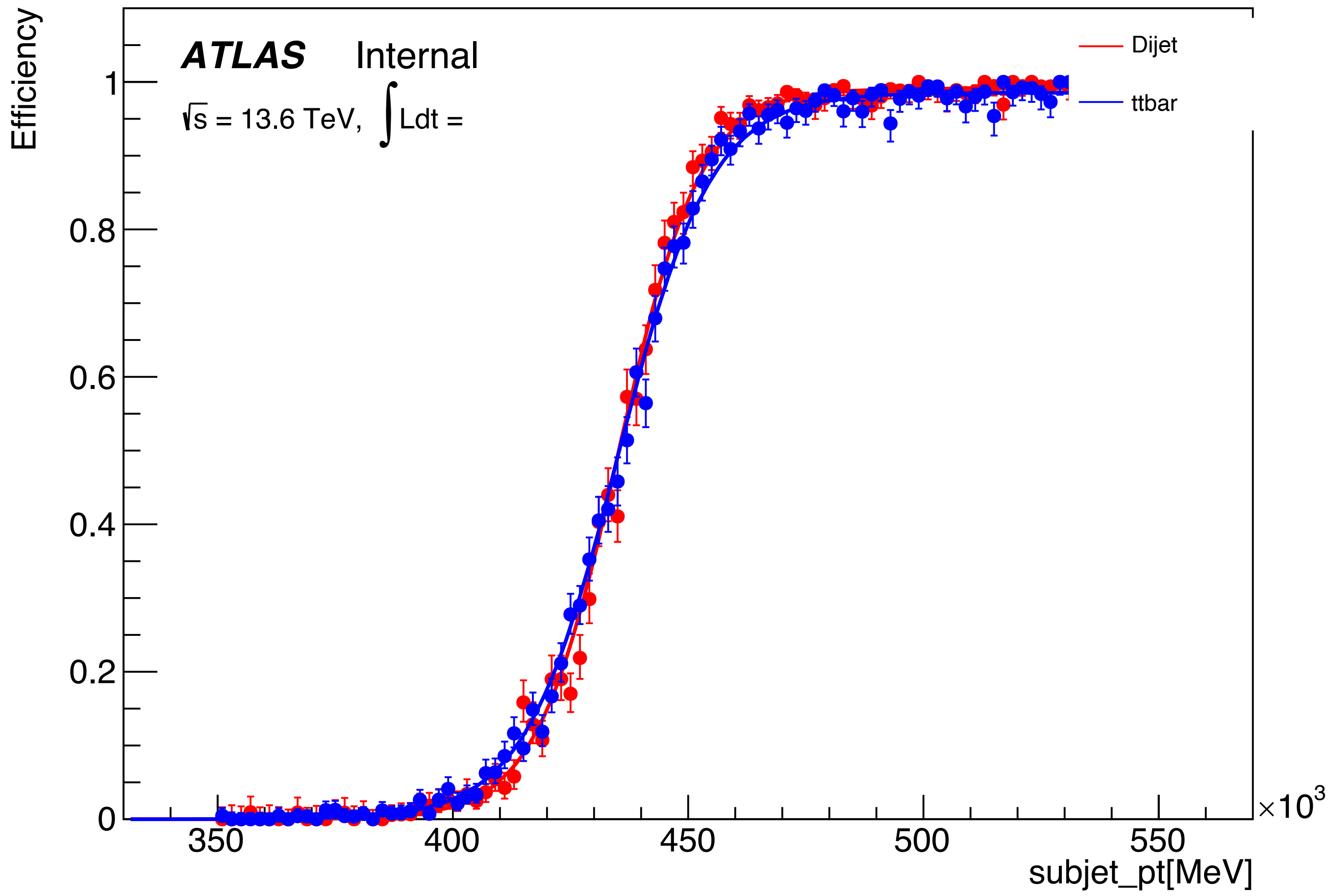
ttbar

p0	=	22799.2	+/-	390.005
p1	=	458500	+/-	611.372
p2	=	0.96943	+/-	0.00292106

Found cut off at 0.93 efficiency: 530561

trigger study MC

- Trigger item: HLT_j460_a10sd_cssk_pf_jes_ftf_preselj225_L1J100
- Sigmoid = p2/(1+TMath::Exp(-(x – p1)/p0))



Dijet

p0	=	8704.49	+/-	184.957
p1	=	435131	+/-	294.569
p2	=	0.991279	+/-	0.00122803

Found cut off at 0.99 efficiency: 493027

ttbar

p0	=	9872.54	+/-	175.105
p1	=	435088	+/-	290.259
p2	=	0.985346	+/-	0.00170914

Found cut off at 0.98 efficiency: 486535

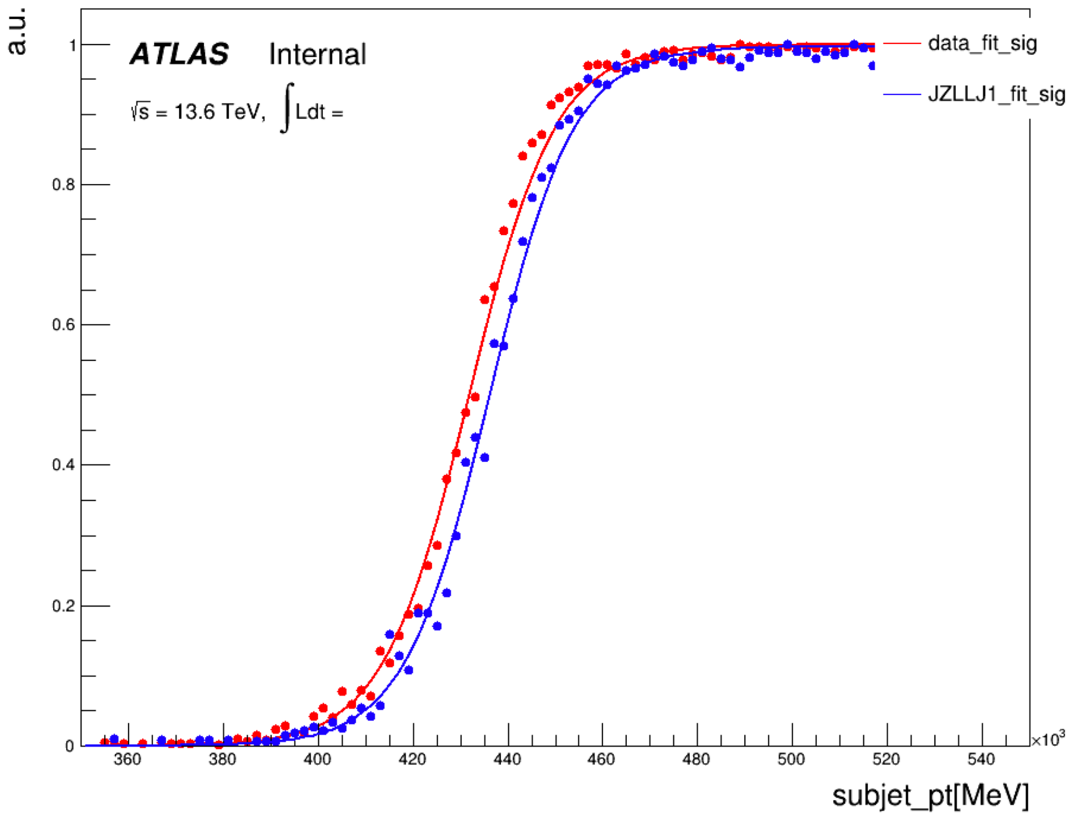
BACK-UP

trigger study

- Tag and probe method: the efficiency is computed on the subleading jet pT by requesting that the leading jet is matched with the trigger item and then checking the subleading matching

- The efficiency curve is fitted with sigmoid function: L/(1+TMath::Exp(-(x - a)/b))

$$\text{efficiency} = \frac{\text{leading\_isTrigMatched\_trigger\_item} == 1\ \&\&\ \text{subleading\_isTrigMatched\_trigger\_item} == 1}{\text{leading\_isTrigMatched\_trigger\_item} == 1}$$



	Data	Z+jets
b	8737.24 ± 119.947	9860.62 ± 18.0499
a	430779 ± 196.361	434942 ± 29.8561
L	0.993001 ± 0.000923239	0.985701 ± 0.000179415

New format definition

Francesco's slides

- Format on gitlab
- Starting from DAOD_PHYS

- Added constituents for UFO jets

```
### 1 jet object skimming
sel_1jet_template = "((count (abs({0}eta) < 2.8 && {0}pt > 150*GeV && {0}m > 30*GeV)  >= 1))"
topology_selection_1jet = "({})".format(
    " || ".join([sel_1jet_template.format(j) for j in largeRJetsForSkimming])
)
```

- Added event skimming
  ↪ At least one large-R with $|\eta| < 2.8,\ p_T > 150\ GeV,\ m > 30\ GeV$

```
### trigger skimming
TriggersList = [
    ### baseline run-2
    'HLT_j360_a10_lcw_sub_L1J100',
    'HLT_j420_a10_lcw_L1J100',
    'HLT_j460_a10t_lcw_jes_L1J100',
    ### new run-3
    'HLT_j460_a10sd_cssk_pf_jes_ftf_preselj225_L1J100',
    'HLT_j460_a10_lcw_subjes_L1J100',
    'HLT_j460_a10r_L1J100',
    ### new run-3 mass cut
    'HLT_j420_35smcINF_a10sd_cssk_pf_jes_ftf_preselj225_L1J100',
    'HLT_j420_35smcINF_a10t_lcw_jes_L1J100',
]
```

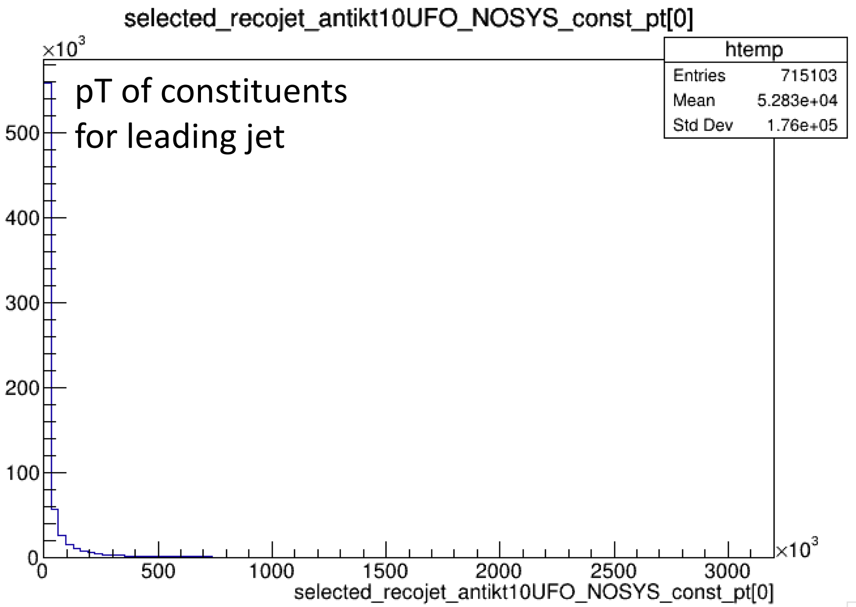
- Added trigger skimming (new Run-3 Large-R jet trigger, still ongoing)

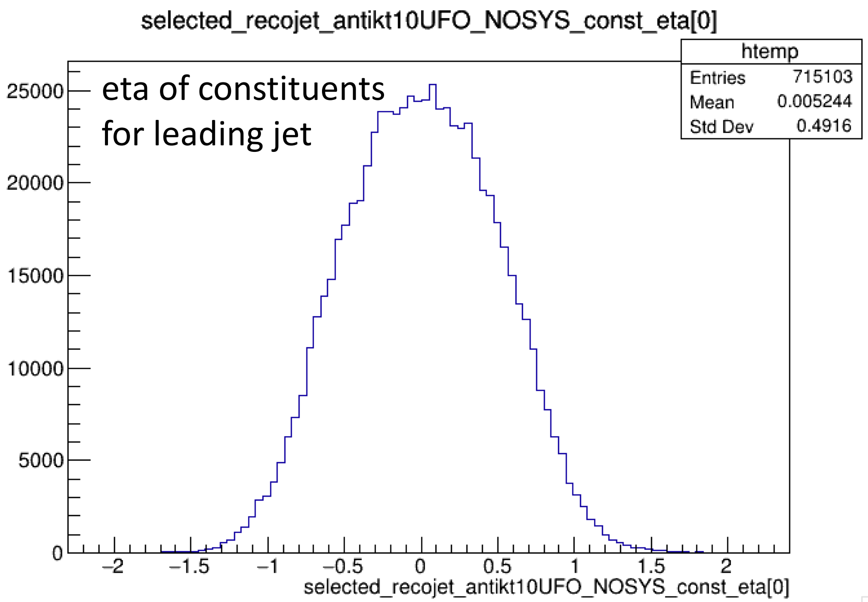
ntuple maker

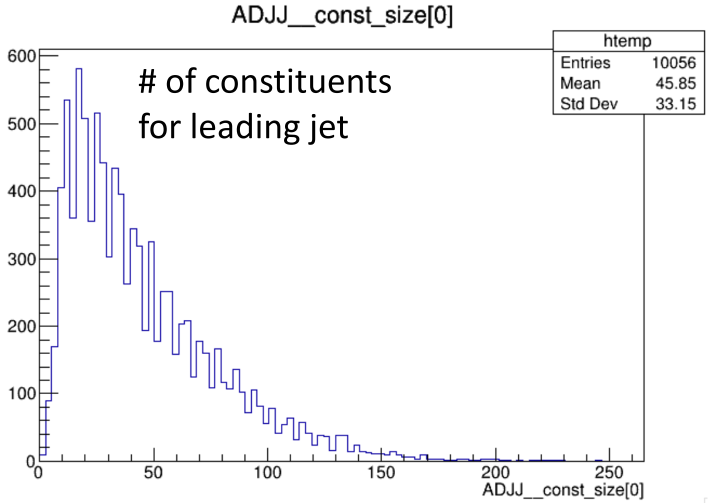
- Production of ntuples from our run 3 LLJ1 DxAOD based on EasyJet framework.

- Achievements:
  - Disabling b-tagging on large-R jets;
  - Customization of list of applied triggers;
  - Computation of new variables from base ones included in DxAOD;
  - **Addition of constituents variables to the final ntuple, also systematic aware**.

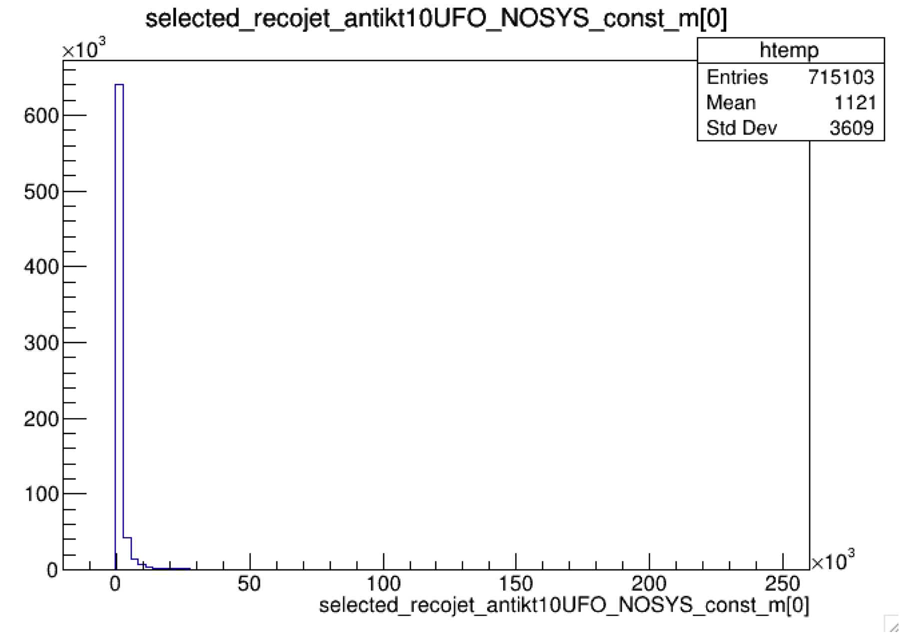
- Running on **LLJ1 MC background** (JZ8 splice) with 20k events.
  - Selections applied on jets pT > 200 GeV and |eta| < 2, 2 jets selected per event.
  - About ~70% of size of ntuple consists of constituents info.

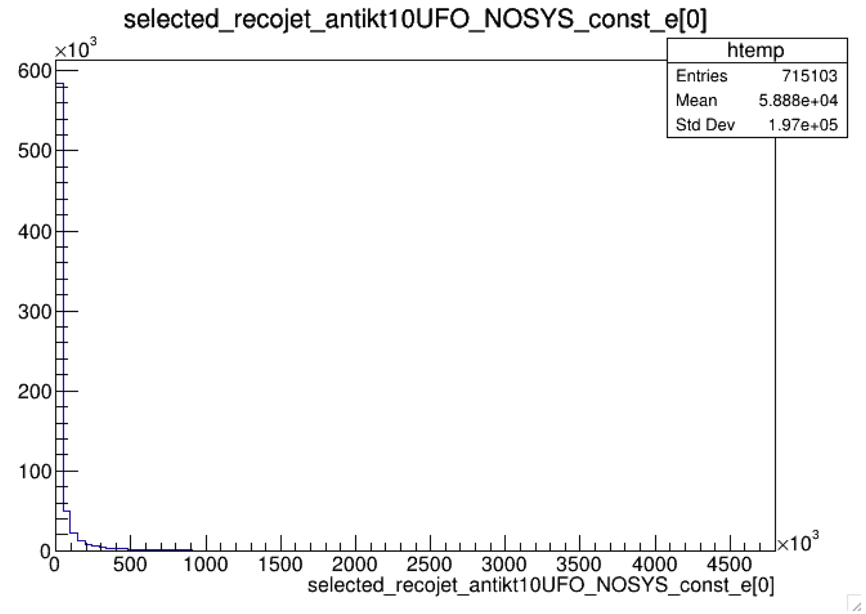


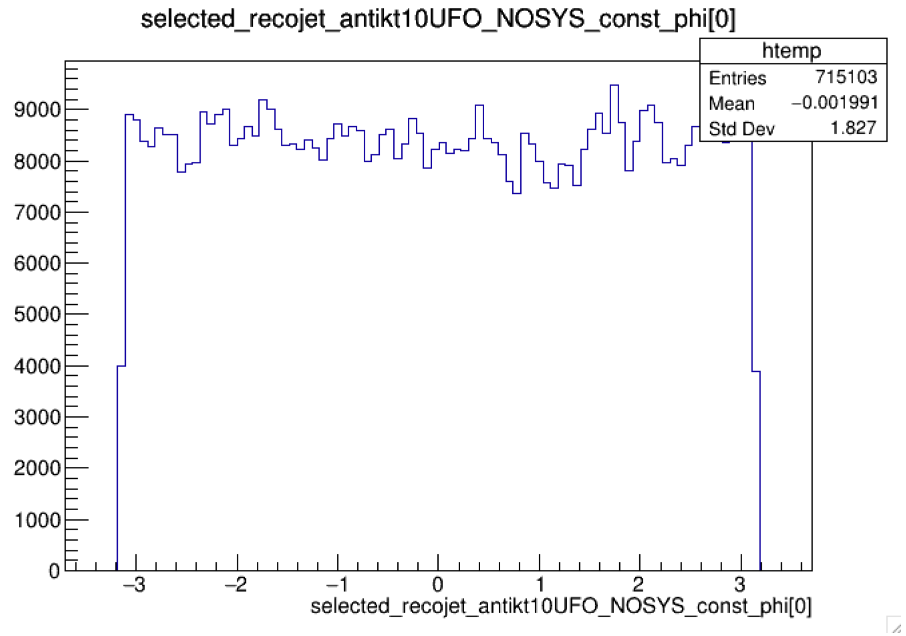


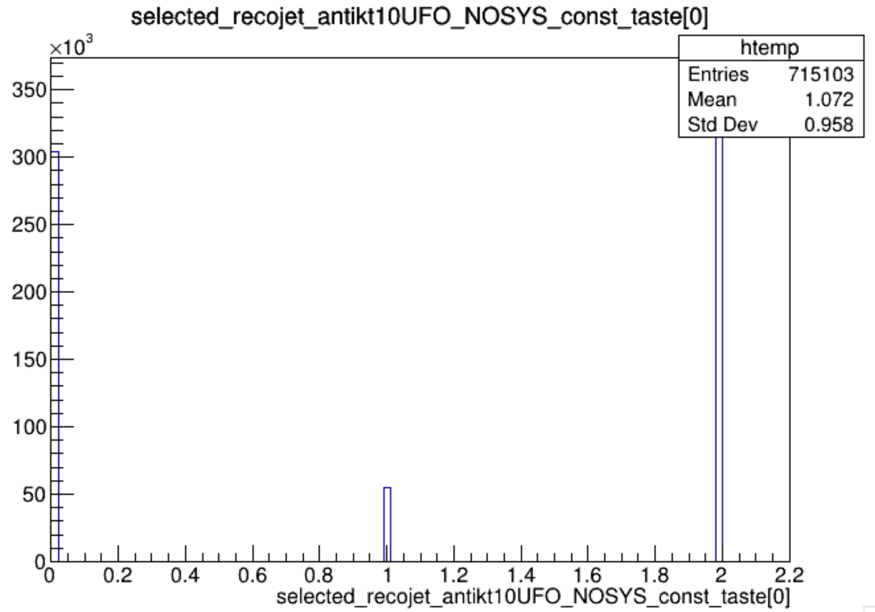


More plots: leading jet

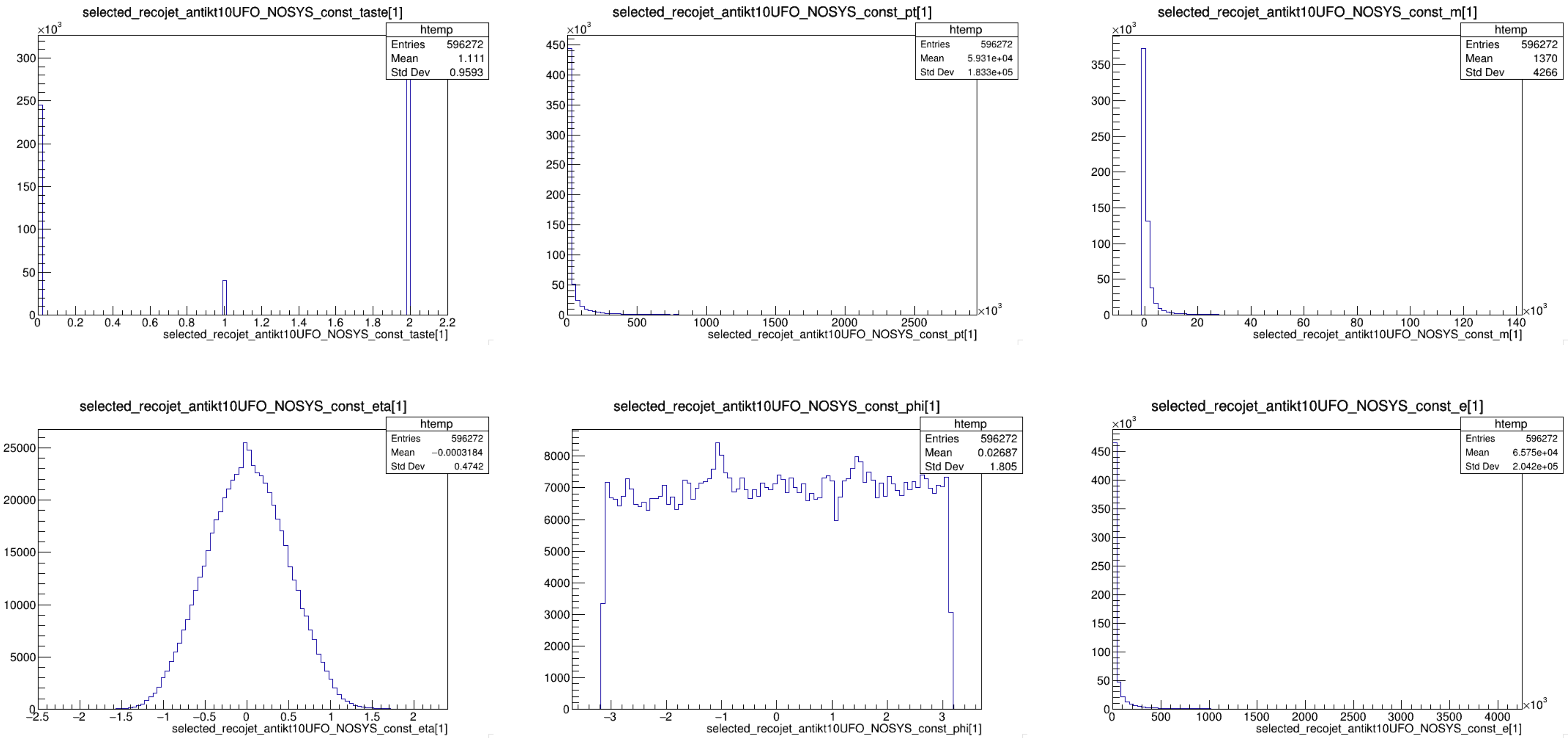








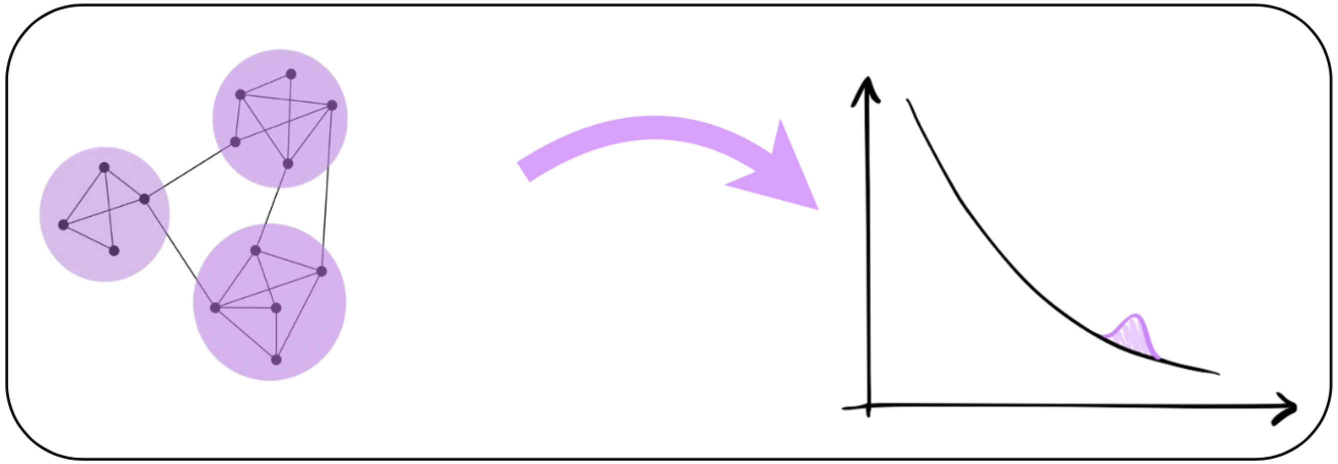
More plots: subleading jet



INTRODUCTION

- **The aim**:
  - **Graph Anomaly Detection** algorithm for the discovery of diboson resonances decaying in fully hadronic final states with the ATLAS detector in run-III.
    - Anomaly Detection: **model-independent** approach, sensitive to more than one signal hypothesis as it only detects «anomalies» w.r.t. background.

- **The strategy**:
  - Reconstructed jets at each event are represented as graphs.
  - This dataset of graphs is then used to train a Graph Neural Network (GNN) model in order to classify anomalous signal vs background by means of an anomaly score.

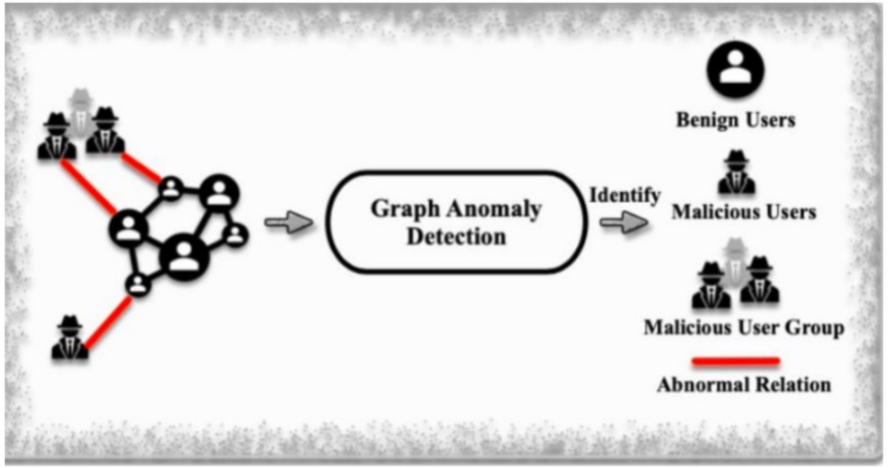


GRAPH ANOMALY DETECTION IN HEP

- Graph: Structured objects composed of entities used to describe and analyze relations and interactions (edges) between such entities (nodes).
  - Nodes and edges typically contain features specific to each element and each pair.

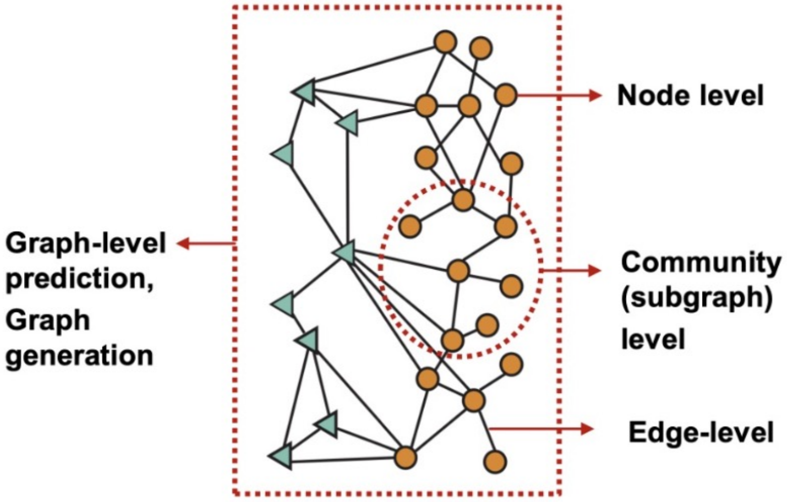
Application of machine learning to build an anomaly detection algorithm on graphs

- Graph Anomaly Detection (GAD) applied to many research fields (social networks, e-commerce, medicine, and telecommunications) where graph representation is more natural than classic data sequencing.
  - Many successful results obtained.
  - Yet to be applied in High Energy Physics analysis.

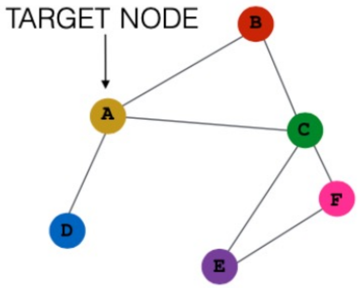


GRAPH NEURAL NETWORKS

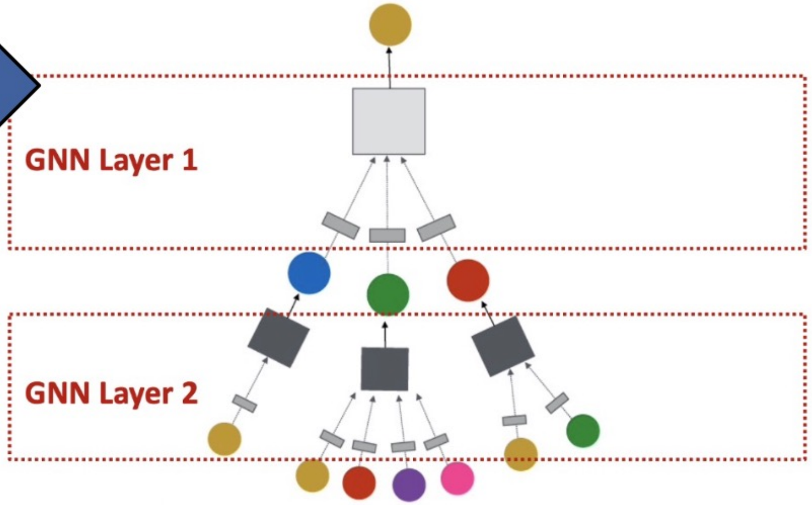
- Graph Neural Networks (GNNs) are ML architectures built specifically to make predictions on graphs, exploiting their relational nature.
  - Based on learnt vector representation (embedding) of each node of the input graphs.
- The embeddings are updated at each layer by aggregating the information passed between the target node and the nodes from its closest neighbourhood → **message passing**



Several task levels, carried out by processing the final node embeddings in certain ways.



INPUT GRAPH

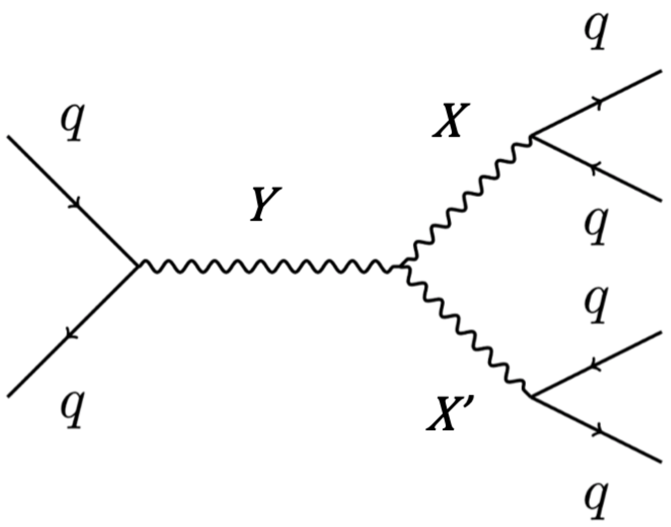


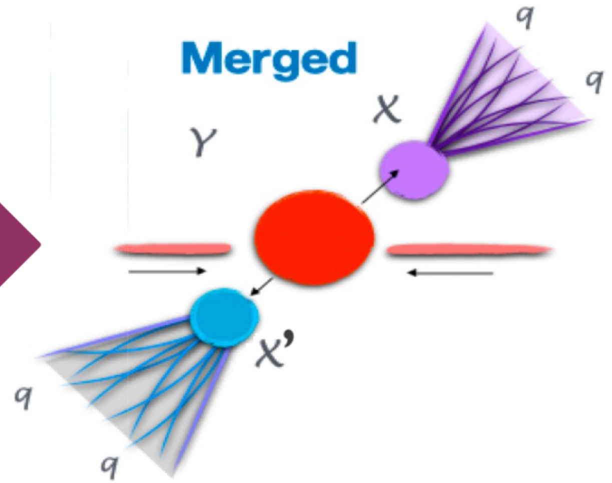
Each layer of GNN extends the neighbour range

ANALYSIS CHANNEL OF INTEREST

- Events signature
  - Background: QCD, to be estimated with data-driven approach (typical of unsupervised tasks).
  - Signal: $Y \rightarrow XX' \rightarrow qqqq$, where X or X' can be either a SM or BSM boson.

- Aiming at the kinematic region where X and X' are produced with significant Lorentz boost due to large mass difference between parent and daughters particles.
  - Y reconstructed with two large-R jets (fatjets) of radius ΔR, denoted as **merged regime**.





Mass of decaying particle

$$\boxed{\Delta R \cong \frac{2m}{p_T}}$$

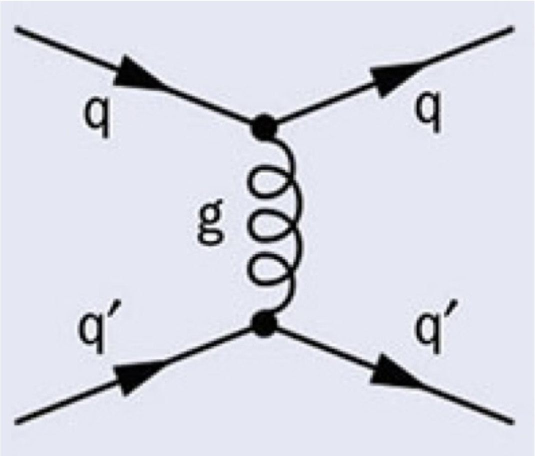
p_T of decaying particle

CURRENT BENCHMARK DATASET

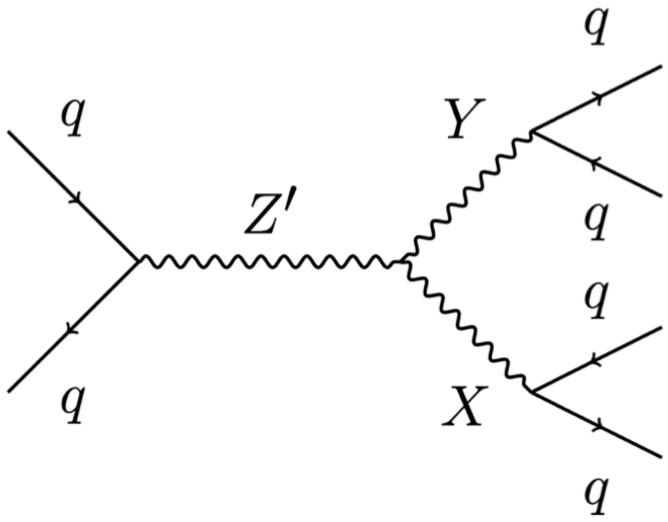
- Benchmark application with LHC Olympics 2020 R&D dataset.
  - MC generated dataset built specifically for anomaly detection.
  - 1.1M total events, 1M background and 100k anomalous signal.

- Events signature
  - Background: QCD di-jet.
  - Signal: Z' → XY → qqqq, particles reconstructed as fatjets with large radius R = 1.

QCD di-jet



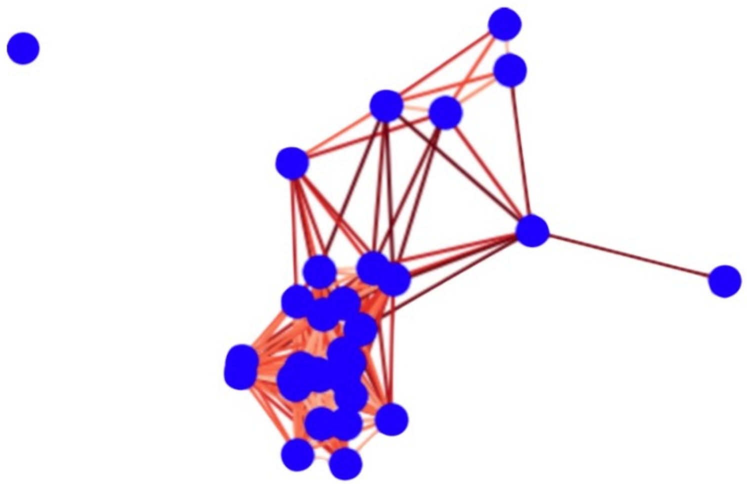
Particle	Mass [GeV]
Z'	3500
X	500
Y	100

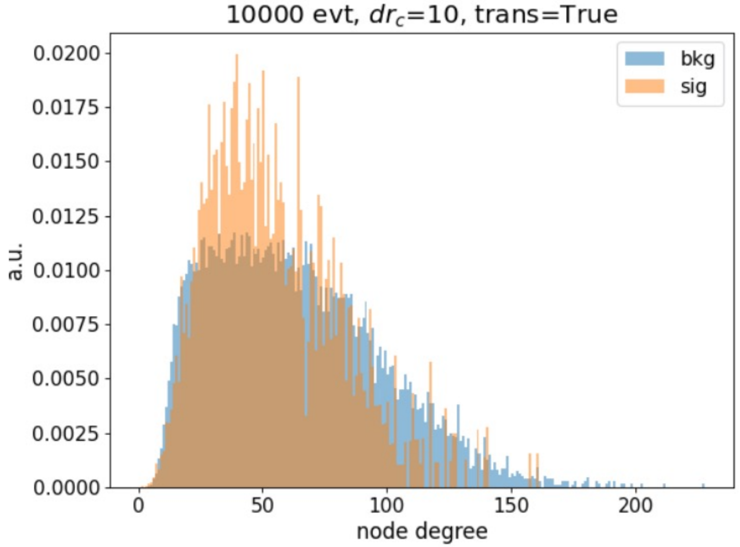


GRAPH REPRESENTATION OF JETS

- **Current definition of a jet**
  - Entites:
    - Nodes → topoclusters contained in each jet reconstructed with anti-k_t algorithm
    - Edges → Created only if ΔR < 0.2 (previously 0.4) between two topoclusters, no self-loops
  - Features:
    - Nodes → pT fraction, η, ϕ.
    - Edges → $1/(\Delta R + \varepsilon)$

- **Transfomation** applied for data augmentation and model robustness reasons (arXiv:1903.02032, arXiv:2105.09274).
  - Rescaling of the four momenta (m_0 = 0.25 GeV) → boost so that the energy is E_0 = 1 GeV → further rotation of constituents along jet axis.

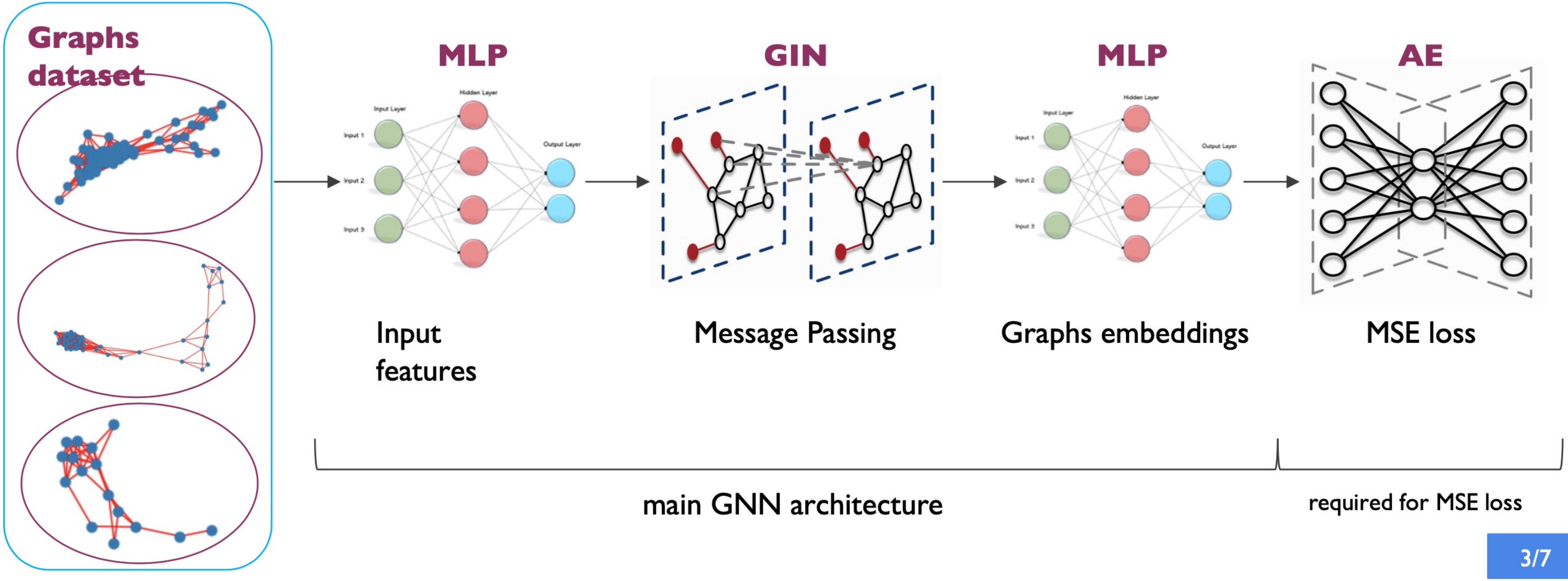




nodes connected to node v

Current architecture in a nutshell

- ➢ Main work on a GNN only model, so we need a suitable optimization task.
- ➢ A popular anomaly detection loss function, the **Mean Squared Error** (MSE), could be used, but an autoencoder (AE) must be connected to compare each graph representation with the AE output.

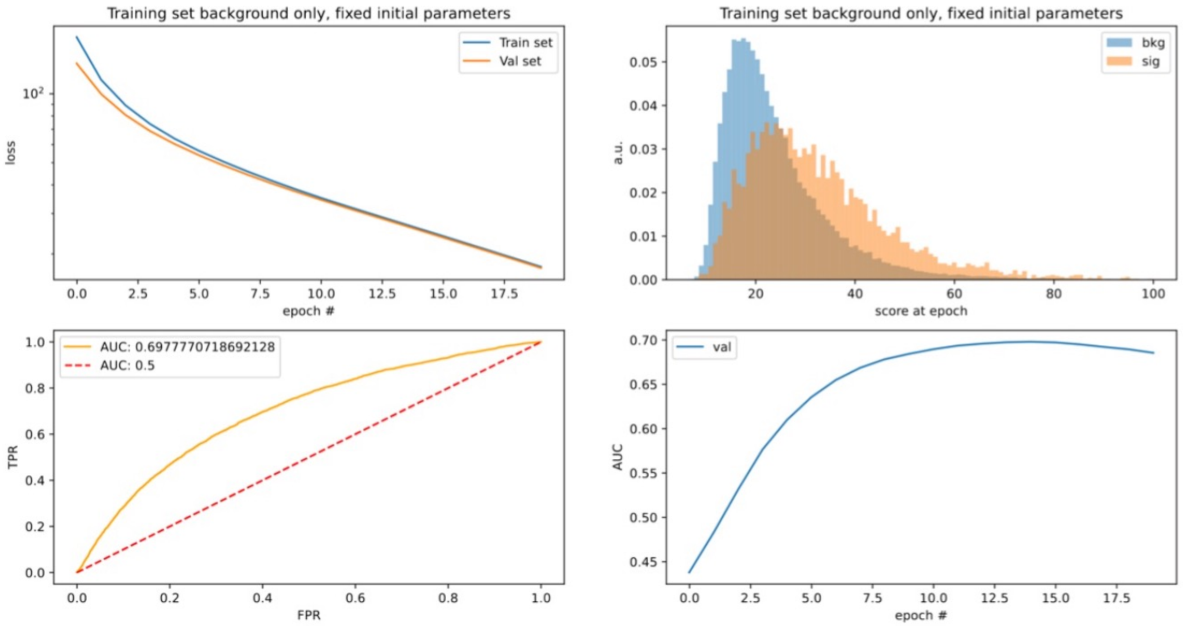


Previous state of the art

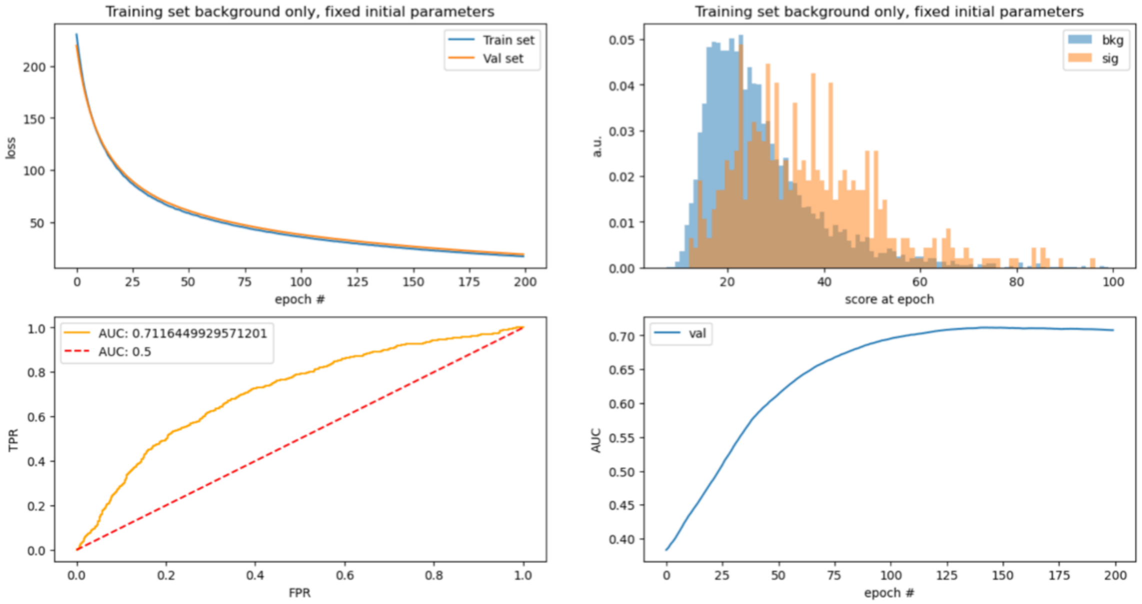
- DeepSVDD model, GIN model with MLP layers applied before and after.

$$L = \lambda_L \frac{1}{N}\sum_{i}^{N} \|GIN(x_i; W_{GIN}) - c_{GIN}\|^2 + \lambda_G \frac{1}{N}\sum_{i}^{N} \|MLP(X_i; W_{MLP}) - c_{MLP}\|^2 + weight\ decay$$

100k events



10k events



EGATConv

- EGAT extends on GAT model by implementing edge features in a different way and by allowing updating of the edge weights tensor between each layer of GNN (edge embedding).

GATConv

class **dgl.nn.pytorch.conv.GATConv**(*in_feats, out_feats, num_heads, feat_drop=0.0, attn_drop=0.0, negative_slope=0.2, residual=False, activation=None, allow_zero_in_degree=False, bias=True*) [source]

Bases: `torch.nn.modules.module.Module`

Graph attention layer from Graph Attention Network

$$h_i^{(l+1)} = \sum_{j\in\mathcal{N}(i)} \alpha_{i,j} W^{(l)} h_j^{(l)}$$

where α_{ij} is the attention score bewteen node i and node j:

$$\alpha_{ij}^l = \mathrm{softmax_i}(e_{ij}^l)$$
$$e_{ij}^l = \mathrm{LeakyReLU}\left(\vec{a}^T [W h_i \| W h_j]\right)$$

Returns:

- *torch.Tensor* – The output feature of shape $(N, *, H, D_{out})$ where H is the number of heads, and D_{out} is size of output feature.
- *torch.Tensor, optional* – The attention values of shape $(E, *, H, 1)$, where E is the number of edges. This is returned only when `get_attention` is `True`.

EGATConv

class **dgl.nn.pytorch.conv.EGATConv**(*in_node_feats, in_edge_feats, out_node_feats, out_edge_feats, num_heads, bias=True*) [source]

Bases: `torch.nn.modules.module.Module`

Graph attention layer that handles edge features from Rossmann-Toolbox (see supplementary data)

The difference lies in how unnormalized attention scores e_{ij} are obtained:

$$e_{ij} = \vec{F}(f'_{ij})$$
$$f'_{ij} = \mathrm{LeakyReLU}\left(A[h_i \| f_{ij} \| h_j]\right)$$

where f'_{ij} are edge features, A is weight matrix and

Returns:

- *pair of torch.Tensor* – node output features followed by edge output features The node output feature of shape (N, H, D_{out}) The edge output feature of shape (F, H, F_{out}) where:

  H is the number of heads, D_{out} is size of output node feature, F_{out} is size of output edge feature.

- *torch.Tensor, optional* – The attention values of shape $(E, H, 1)$. This is returned only when :attr: *get_attention* is `True`.

- **Selfloop** is required because of how the node representation is updated.

GLocalKD

- Article: [Deep Graph-level Anomaly Detection by Glocal Knowledge Distillation].

- Employs a variation of Knowledge Distillation (KD) technique, where the initial goal is to train a simple model that syntetize the knowledge of a large model while maintaining similar accuracy as the large model.
  - Random Knowledge Distillation for joint distillation at node-level and graph-level.

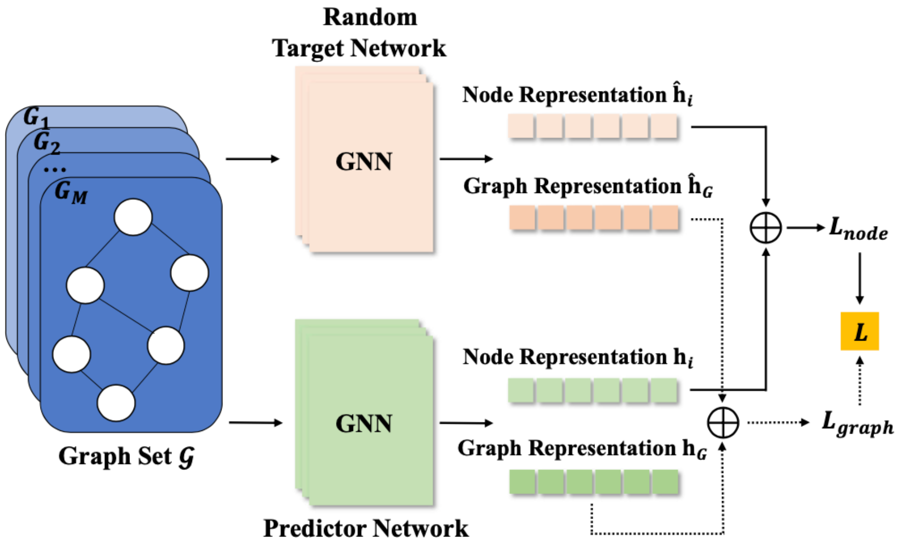
- Implements two GNNs:
  - Random Target Network, not-trained and randomly initialized, used as reference to learn the normal patters of our dataset.
  - Predictor Network, trained by comparing its node and graph representations ($\mathrm{h_G}$, $\mathrm{h_i}$) with the ones from the above network ($\widehat{\mathrm{h_G}}$, $\widehat{\mathrm{h_i}}$) through a KD function.

Loss function for unsupervised learning:

$$L = L_{graph} + \lambda L_{node}$$

$$L_{graph} = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} KD\left(\mathbf{h}_G, \hat{\mathbf{h}}_G\right),$$

$$L_{node} = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \left(\frac{1}{|G|} \sum_{v_i \in \mathcal{V}_G} KD\left(\mathbf{h}_i, \hat{\mathbf{h}}_i\right) \right)$$



GLocalKD paper model

- KD function in L chosen as error between the two networks output.

$$L = L_{graph} + L_{node}$$

$\lambda = 1$

$$L_{graph} = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \|\mathbf{h}_G - \hat{\mathbf{h}}_G\|^2,$$

$$L_{node} = \frac{1}{|\mathcal{G}|} \sum_{G \in \mathcal{G}} \left(\frac{1}{|G|} \sum_{v_i \in \mathcal{V}_G} \|\mathbf{h}_i - \hat{\mathbf{h}}_i\|^2 \right)$$

- Anomaly Score computed for test dataset:

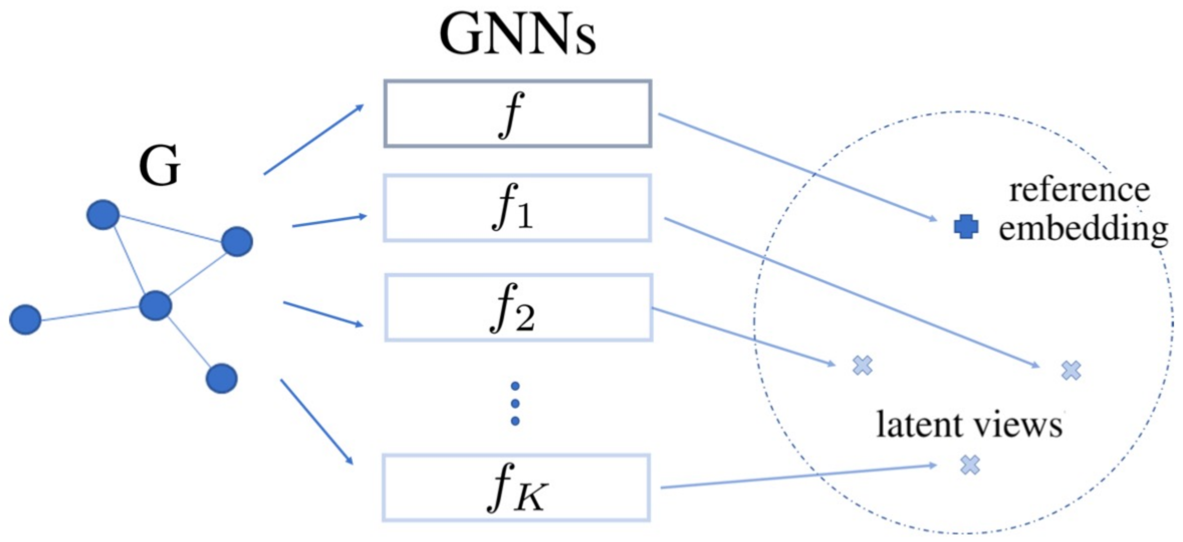
$$f(G; \hat{\Theta}, \Theta^*) = \left\|\mathbf{h}_G - \hat{\mathbf{h}}_G\right\|^2 + \frac{1}{|G|} \sum_{v_i \in \mathcal{V}_G} \|\mathbf{h}_i - \hat{\mathbf{h}}_i\|^2$$

G → number of nodes for graph $\mathcal{G}$

- Things to note:
  - Node degree used as node feature for training.
  - Max pooling to obtain graph representation.

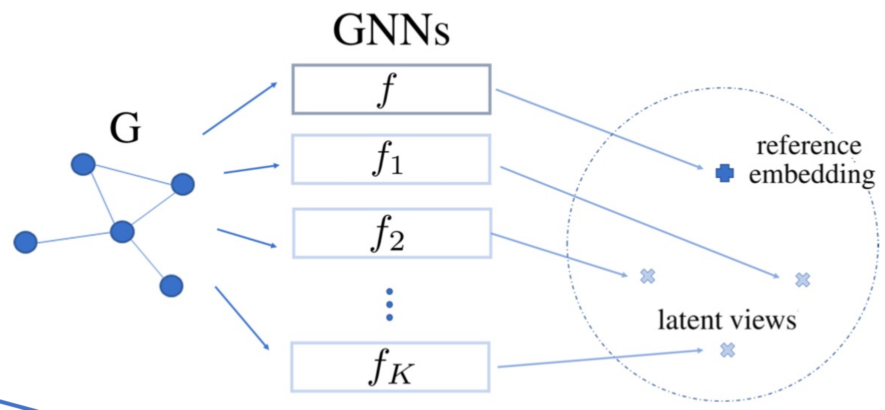
OCGTL

- Article: [Raising the Bar in Graph-level Anomaly Detection].
- Combines one-class classification of OCGIN and neural transformation lerning.
- One reference GNN and K additional GNNs are trained together in order to detect anomalies.
  - Each representation obtained now is used to learn the optimal hypersphere radius of non-anomalies region.
- Advantage w.r.t. DeepSVDD objective:
  - No hypersphere collapse, center can be treated as learnable parameter.
  - More robust training.
  - No performance flip.



OCGTL loss function

- Consists of two terms:



$$\mathcal{L}_{\mathrm{OCGTL}}(\mathrm{G}) = \mathcal{L}_{\mathrm{OCC}}(\mathrm{G}) + \mathcal{L}_{\mathrm{GTL}}(\mathrm{G})$$

$$\boxed{\mathcal{L}_{\mathrm{OCC}}(\mathrm{G}) = \sum_{k=1}^{K} \|(f_k(\mathrm{G}) - \theta)\|_2}$$

One class contribute

$$\mathcal{L}_{\mathrm{GTL}}(\mathrm{G}) = -\sum_{k=1}^{K} \log \frac{c_k}{C_k}$$

$$c_k = \exp\left(\frac{1}{\tau}\mathrm{sim}(f_k(\mathrm{G}), f(\mathrm{G}))\right),$$

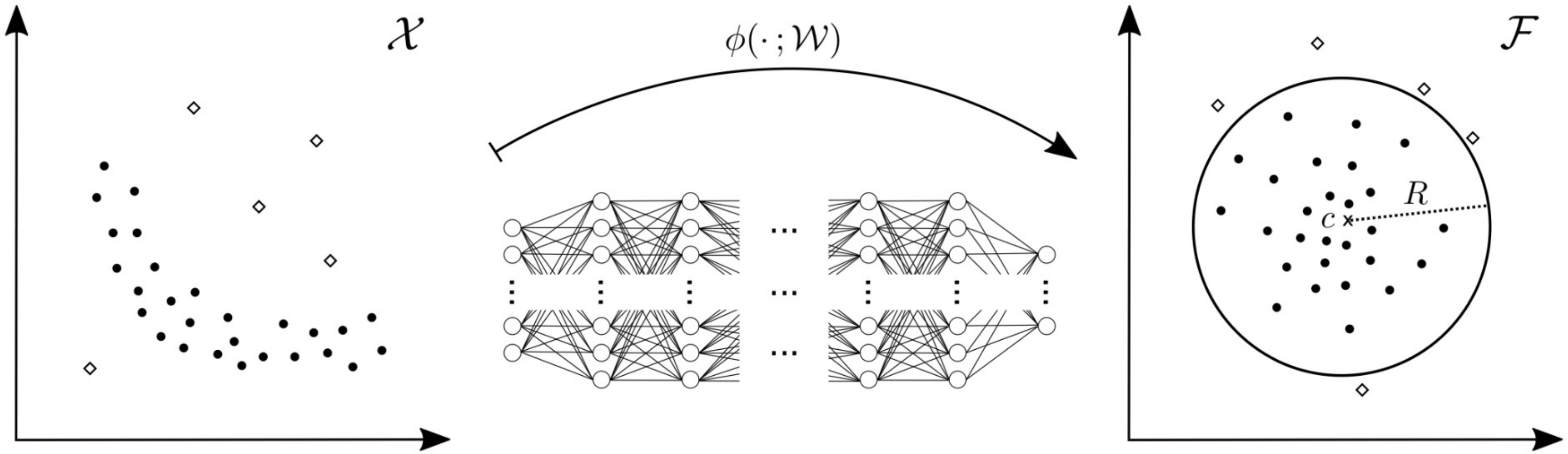
$$C_k = c_k + \sum_{l \neq k}^{K} \exp\left(\frac{1}{\tau}\mathrm{sim}(f_k(\mathrm{G}), f_l(\mathrm{G}))\right)$$

Transformation learning contribute

- $\theta \rightarrow$ center of the hypersphere, sim chosen as cosine similarity $\rightarrow z^T z' / \|z\| \|z'\|$
- τ temperature parameter, final loss on training dataset at each epoch computed as $\mathbb{E}_{\mathrm{G}}\left[\mathcal{L}_{\mathrm{OCGTL}}(\mathrm{G})\right]$

DEEP SUPPORT VECTOR DATA DESCRIPTION (DEEP SVDD)

- Deep SVDD works by minimizing an objective in order to learn and optimize the radius R of a hypersphere in the output space F which only cointains outputs from non-anomalous data features X.
  - Output space defined by the output of the considered ML architecture (NN, MLP, GNN, ecc.)
  - Output from anomalies falls outside of the hypersphere and is identified by its distance from the center c.



objective

$$\min_{W} \frac{1}{N}\sum_{i=1}^{N} \|\mathrm{GIN}(G_i; W) - \mathrm{c}\|^2 + \frac{\lambda}{2}\sum_{l=1}^{L} \|W^l\|_F^2 \qquad W = \{W^1, \ldots, W^l\}$$

$$s(\boldsymbol{x}) = \|\phi(\boldsymbol{x}; \mathcal{W}^*) - \mathbf{c}\|^2$$

Anomaly Score

GRAPH ISOMORPHISM NETWORK (GIN)

- GIN formulation employs both message passing and MLPs, making it the most expressive GNN:

$$\mathrm{MLP}_{\Phi}\left((1+\epsilon)\cdot \mathrm{MLP}_f(c^{(k)}(v)) + \sum_{u\in N(v)} \mathrm{MLP}_f(c^{(k)}(u))\right)$$

learnable parameter

$$\boxed{c^{(k)}(u) \leftrightarrow h_j^{(l)}}$$

Embedding of node u al layer (k)

- This expression can be rewritten in a more general way, also allowing for edge weights to be considered in the graph convolution.

$$\boxed{h_i^{(l+1)} = f_\Theta\left((1+\epsilon)h_i^l + \mathrm{aggregate}\left(\left\{e_{ji}h_j^l, j\in\mathcal{N}(i)\right\}\right)\right)}$$

- Aggregate can be any permutation invariant function (Sum, Mean, Max ecc.)