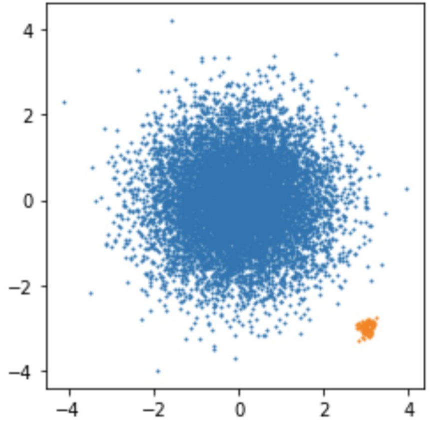
GRAPH NEURAL NETWORKS AND ANOMALY DETECTION IN RUN-III AT ATLAS

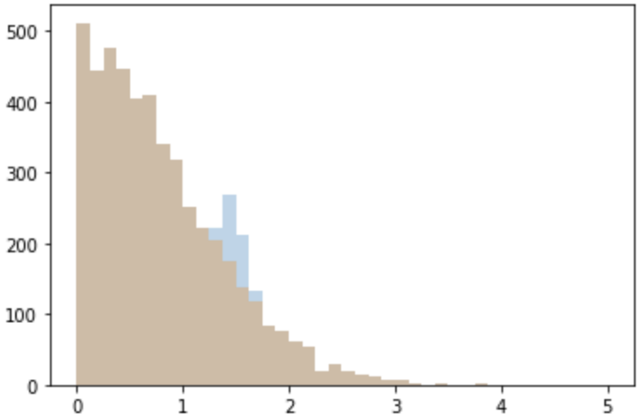
ANTONIO D'AVANZO, O.B.O. THE ATLAS GROUP OF NAPLES

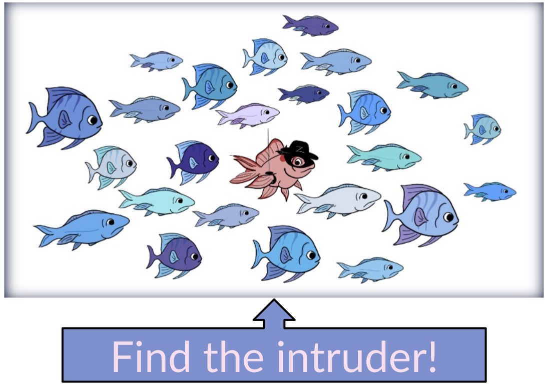
Riunione Gruppo 1, Napoli, 12/01/2023

WHAT IS ANOMALY DETECTION?

- Anomalies: abnormal objects significantly different from other members of a sample.
  - Anomaly Detection refers to ML techniques used to spot these outliers.
- Particle physics scenario → Identification of **model-independent** features of detector data inconsistent with the expected background.
  - Related works: PRL 125 131801, arXiv:2105.09274, ATLAS-CONF-2022-045.

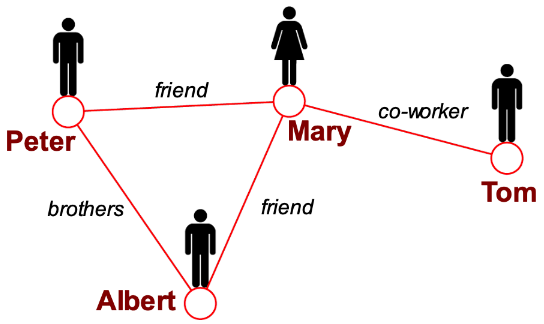




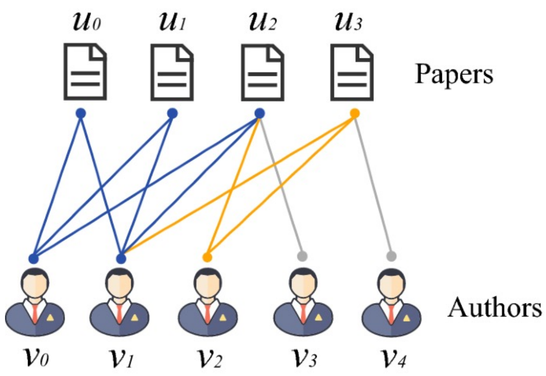


WHAT ARE GRAPHS?

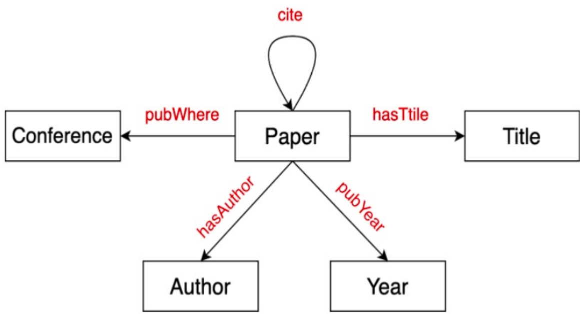
- Structured objects composed of entities used to describe and analyze relations and interactions (edges) between such entities (nodes).
  - Nodes and edges typically contain features specific to each element and each pair.
  - Many types of graphs based on the relations: directed, heterogeneous, bipartite, weighted ecc.



Undirected graph

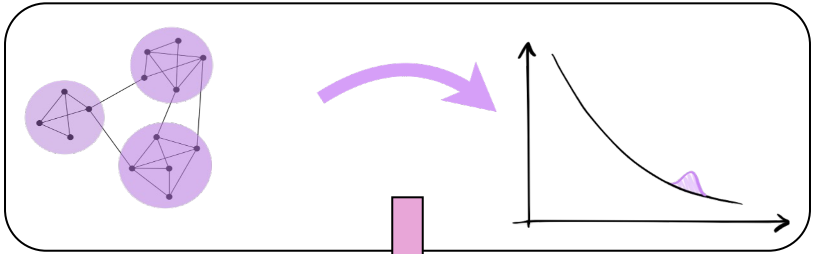


Bipartite graph



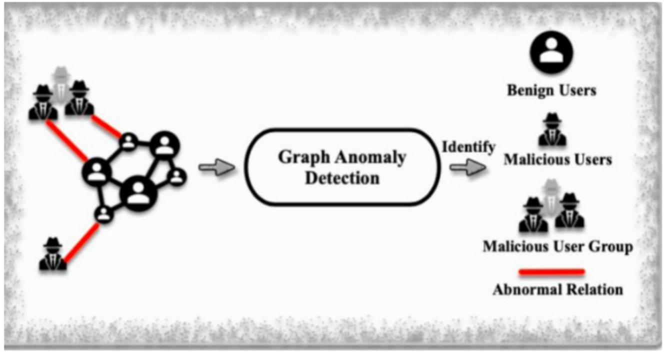
Directed graph

COMBINE: GRAPH ANOMALY DETECTION IN HEP



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

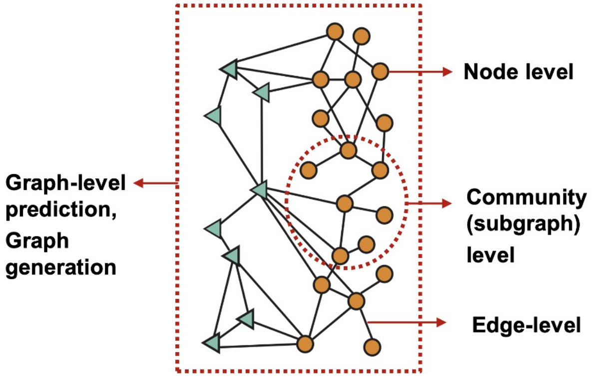
- Graph Anomaly Detection (GAD) applied to many research fields (social networks, e-commerce, medicine, and telecommunications) where graph representation is more natural than classic data sequencing.
  - Many successful results obtained.
  - Yet to be applied in High Energy Physics analysis.



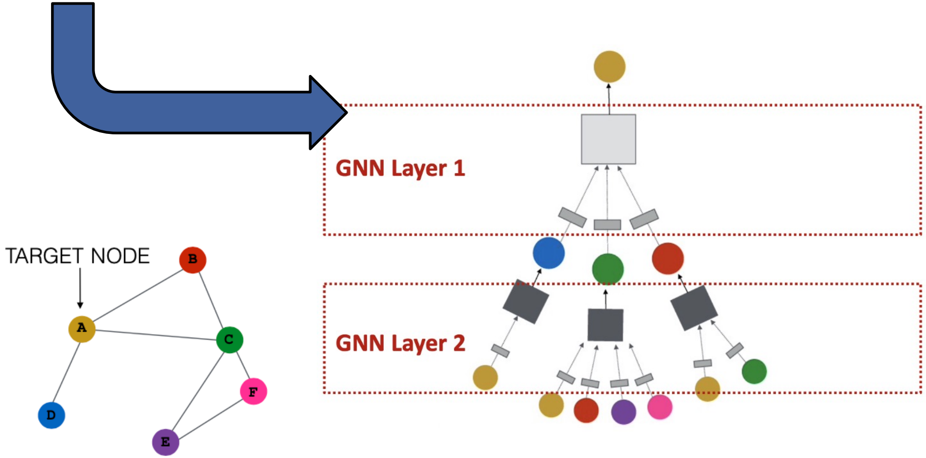
Our strategy: represent jets in heavy diboson resonance searches with hadronic final states as graphs using LHC **run-III** data

HOW? 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.

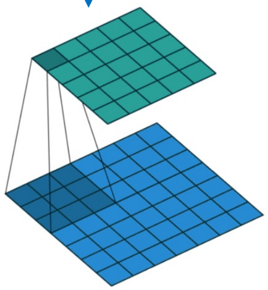


INPUT GRAPH

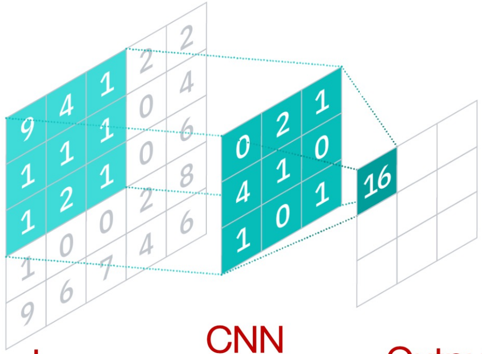
Each layer of GNN extends the neighbour range

CNN VS GNN

- CNNs are special GNNs with fixed neighbour size and nodes ordering of the input graphs.
  - Heterogeneous objects can be treated as nodes
  - Graphs typically have arbitrary number of connections between nodes, as opposed to images.
  - Possibility to assign any kind of information to nodes and edges (structural and features).
- GNN message passing formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\mathbf{W}_l \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \dots, L-1\}$
- CNN convolution formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v) \cup \{v\}} \mathrm{W}_l^u \mathrm{h}_u^{(l)}), \forall l \in \{0, \dots, L-1\}$
  - Rewritten as: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v)} \mathbf{W}_l^u \mathrm{h}_u^{(l)} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \dots, L-1\}$



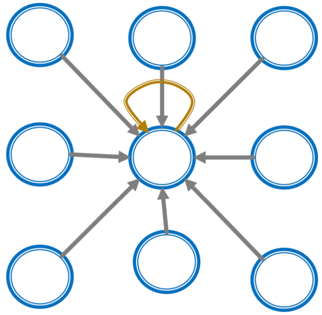
Image



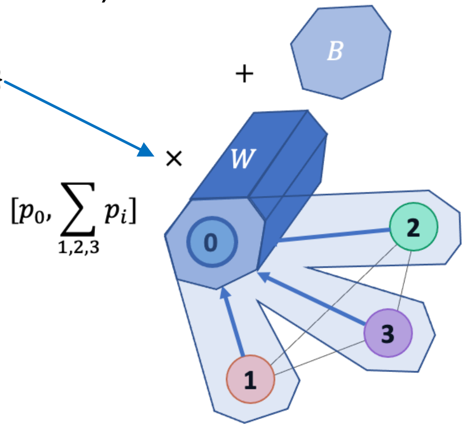
Image

CNN weights

Output



Graph



B and W: weight parameters
N(v): set of neighbours of node v
σ: non-linear activation function
$h_u^{(l)}$: embedding at layer l of node u

ON-GOING WORK

- Development of a GAD algorithm for the discovery of diboson resonances decaying in fully hadronic final states.

- Collaborative effort between the Napoli and Roma I ATLAS groups.
  - Active since after summer, we are in R&D phase.
  - Objective: obtain results on the full run-III dataset (2-3 years).

Meet the full team

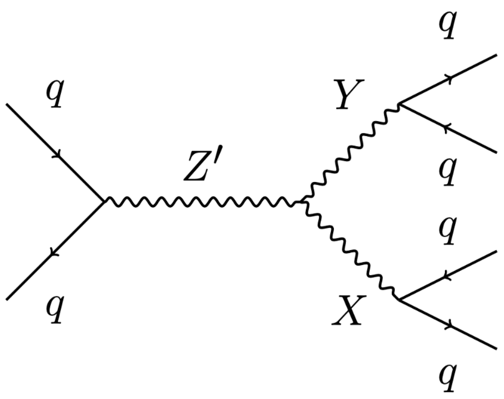
Who	Where	Role
Valerio Ippolito	Roma 1	Faculty
Stefano Giagu	Roma 1	Faculty
Graziella Russo	Roma 1	PhD (expected 2025)
Francesco Conventi	Napoli	Faculty
Elvira Rossi	Napoli	Faculty
Francesco Cirotto	Napoli	Post-doc
Antonio D’Avanzo	Napoli	Post-master fellow

DATASET

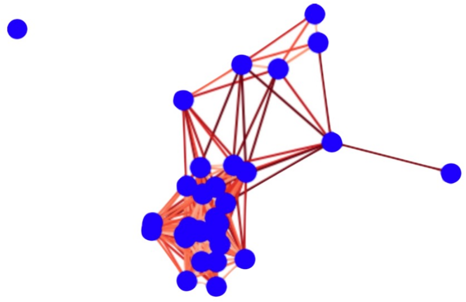
- Benchmark application with LHC Olympics 2020 R&D dataset.
  - MC generated dataset built specifically for anomaly detection.
  - 1.1M total events, 1M background and 100k anomalous signal.

- Events signature
  - Background: QCD di-jet.
  - Signal: $Z' \rightarrow XY \rightarrow qqqq$, particles reconstructed as single jets (Fatjets) with large radius R = 1.

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

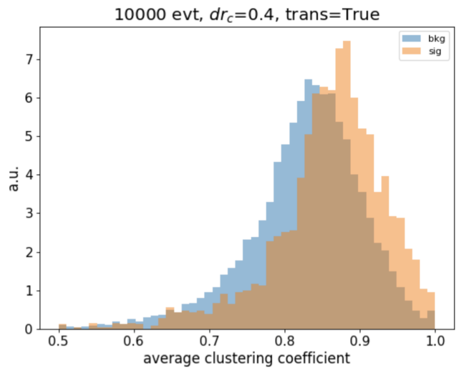


GRAPH REPRESENTATION OF JETS

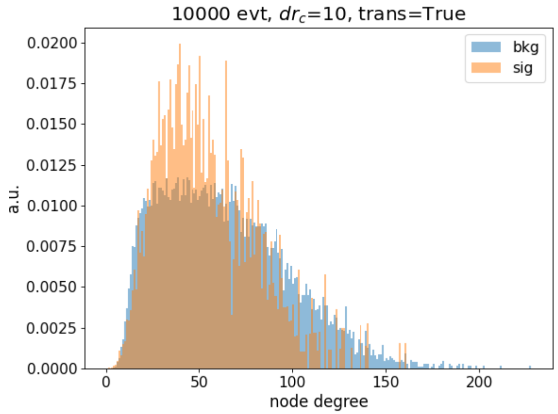


- **Current definition of a jet**
  - Entites:
    - Nodes → topoclusters contained in each jet reconstructed with anti-k_t algorithm
    - Edges → Created only if ΔR < 0.4 between two topoclusters, no self-loops
  - Features:
    - Nodes → pT fraction, η, ϕ.
    - Edges → $1/(\Delta R + \varepsilon)$

- **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.



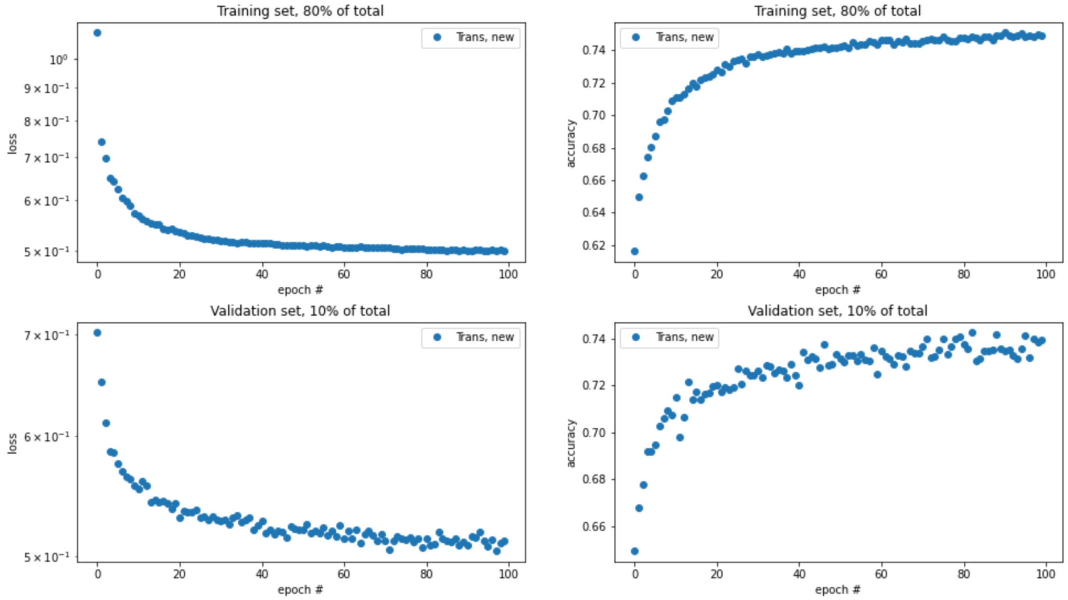
Measure of the connection between neighboring nodes of node ν



nodes connected to node ν

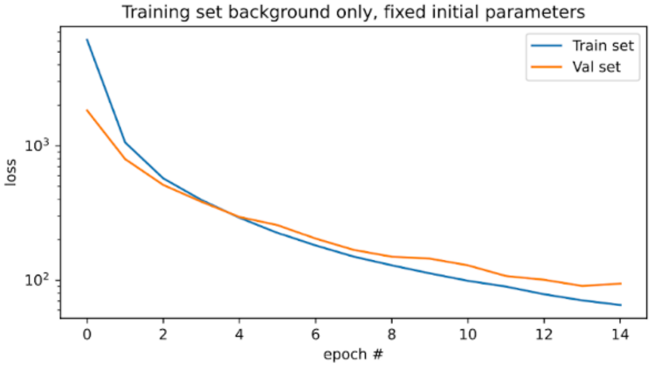
PRELIMINARY FIRST APPROACH : CLASSIFICATION

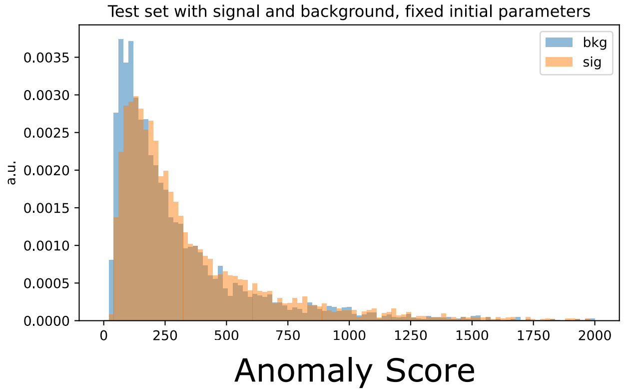
- Graph Isomorphism Network (GIN) model used as GNN layers for message passing.
- Jet-level signal vs background classification with GNNs.
  - Supervised optimization of cross entropy with Adam optimizer, results interpreted with predicted labels.

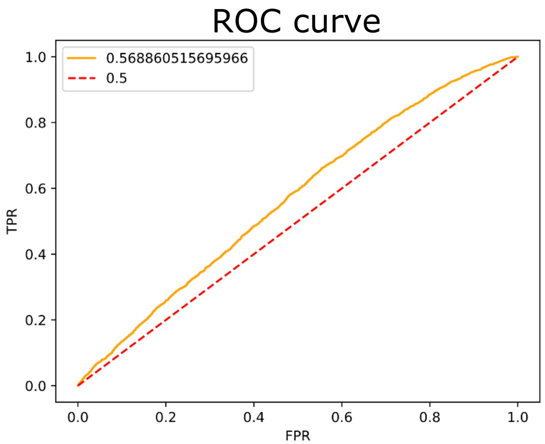


PRELIMINARY SECOND APPROACH: ANOMALY DETECTION

- Graph Isomorphism Network (GIN) model used as GNN layers for message passing.
  - Implementation of pre- and post-processing MLP layers.
- Jet-level anomalous signal identification among background with GNNs.
  - Unsupervised optimization of DeepSVDD objective with Adam optimizer, results interpreted with anomaly score.





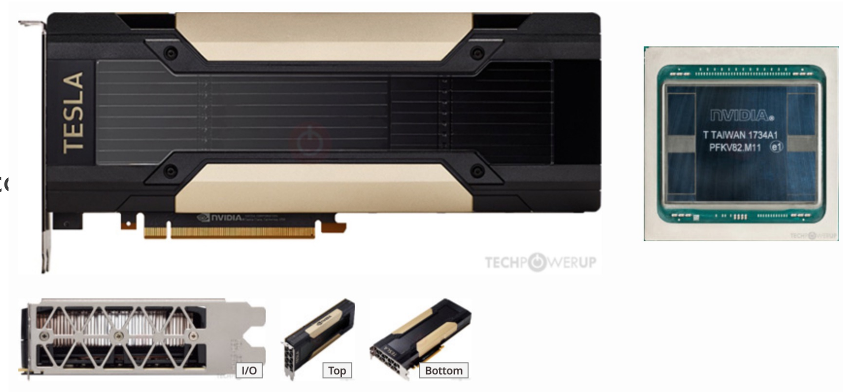


HARDWARE SUPPORT

- Tests run on hardware provided with GPUs, since the input to GNNs is given as data tensors and transformed in other tensors by each layer.
  - Allows for parallelization of model training.

- Common online tools for ML turned out insufficient for our task (free version of AWS, Google Colab, Kaggle), also hard to run on personal hardware not dedicated to ML.
  - Currently run on INFN I.Bi.S.Co cluster.
  - GPUs: 2x NVIDIA Tesla V100S PCIe 32 GB, 5120 cores.

- GNNs are memory- and time-hungry.
  - Training takes about 1 hour - 1 day based on dataset.
  - Favored GPUs with dedicated ML chip architecture, more c
    and more RAM depending on the task.



CONCLUSION

- Graphs neural networks combined with Anomaly Detection have shown great expressive power in many research fields, with positive results w.r.t. standard techniques.
    - **First application** in heavy diboson resonance searches with hadronic final states.

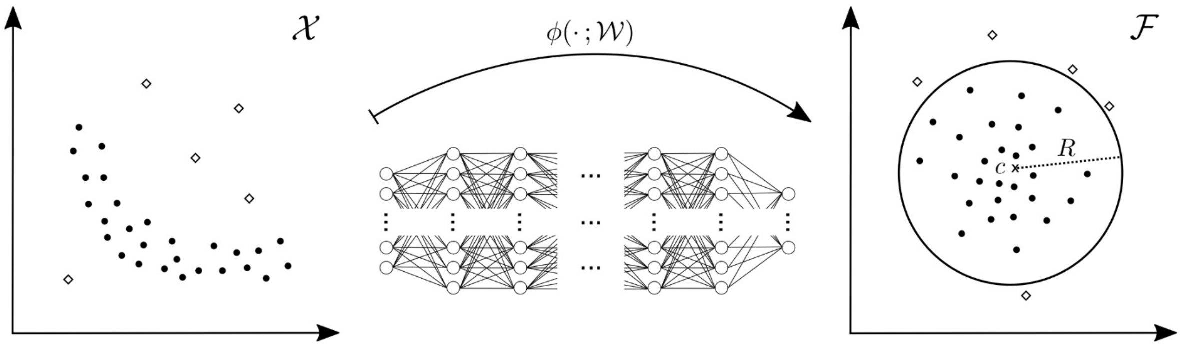
- Our work is still in a preliminary phase, future developments:
    1. Test of graph definition on benchmark models (transformers, autoencoders).
    2. Optimization of GNN models.
    3. Inclusion of full detector info and event-based anomaly score.
    4. Migration to real dataset to explore run-III data gathered by the ATLAS detector for new searches (preferably DBL) or rediscovery of known resonances!

Thank for your attention!

BACKUP

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:

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

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

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.)