

UNIVERSIDAD DE OVIEDO

GRAPH NEURAL NETWORKS FOR DISPLACED MUON CLASSIFICATION

02/JUN/2026

Elena Aller
Andrea Cardini
Daniel Estrada
Santiago Folgueras
Pelayo Leguina
Miguel Obeso



Universidad de Oviedo



Funded by
the European Union

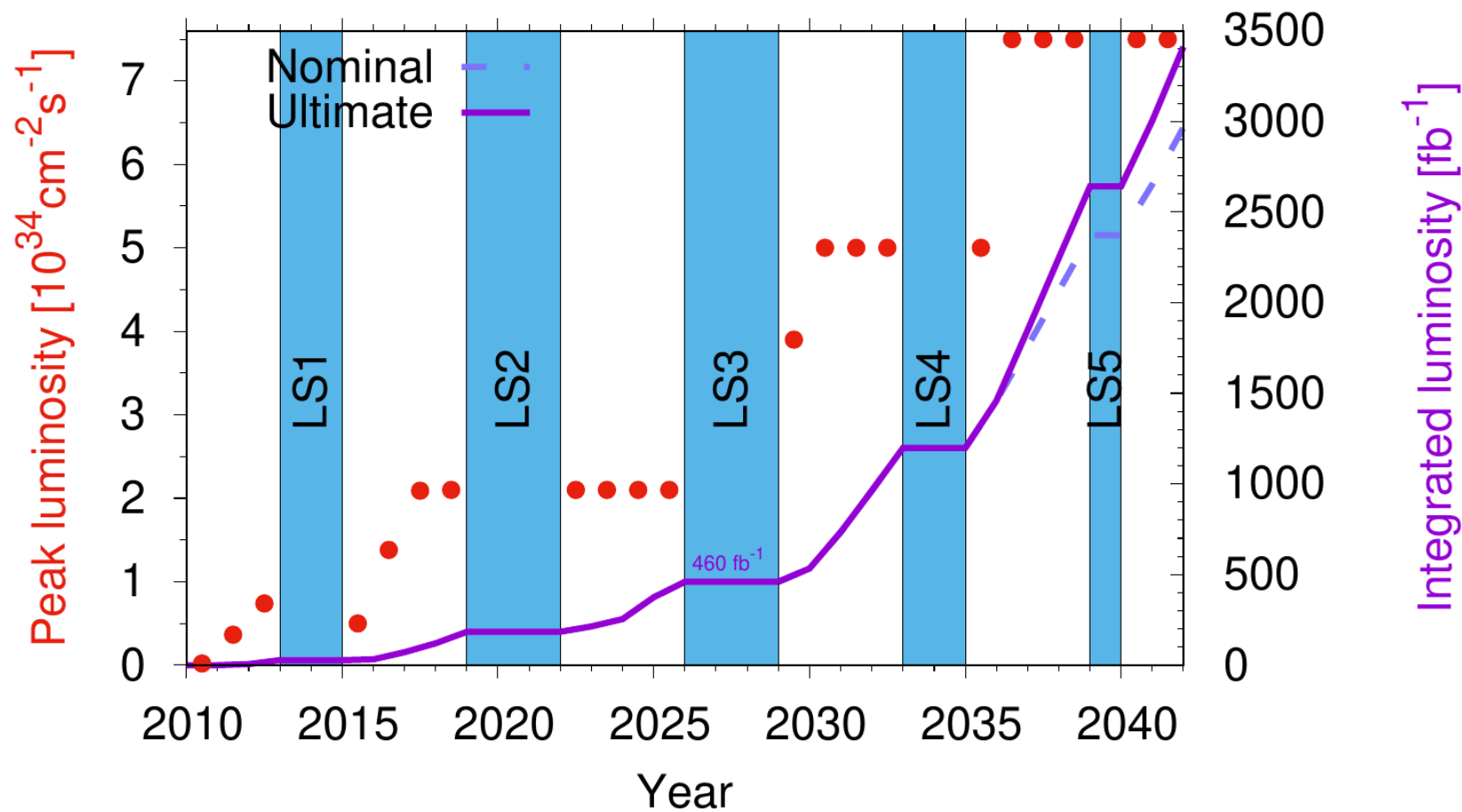


European Research Council
Established by the European Commission

Motivation

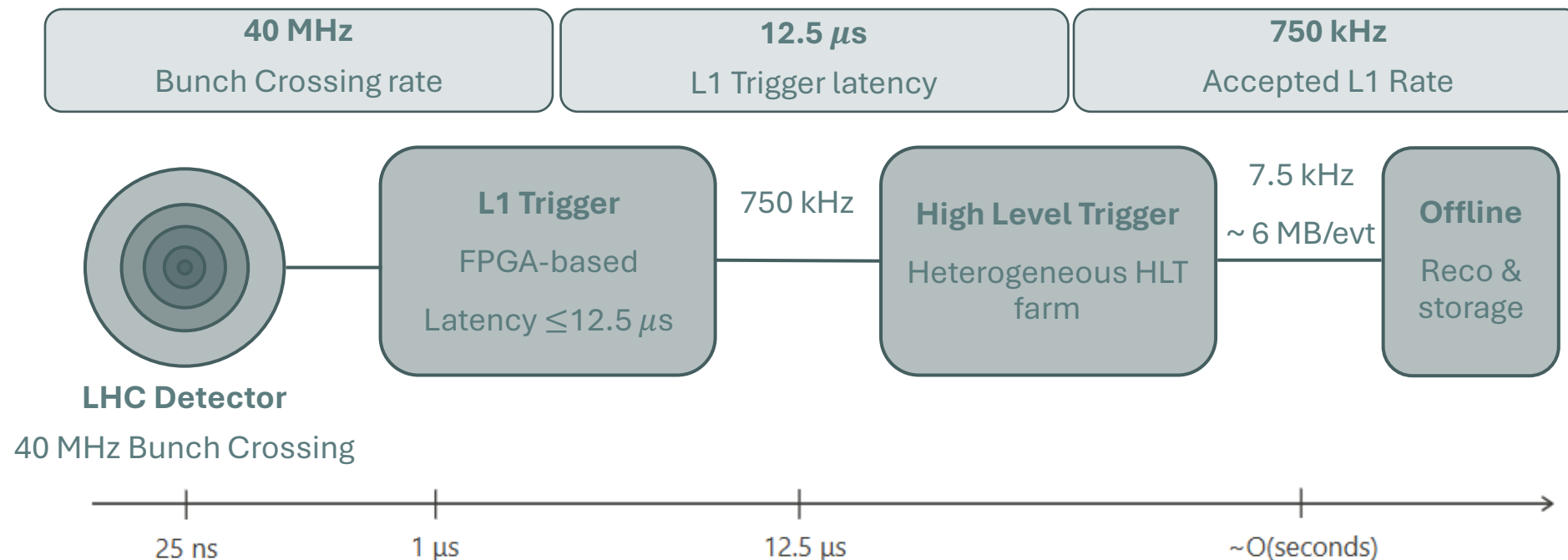
During the HL-LHC project, the collisions are expected to have **higher pileup and event rate**, therefore, a faster and more efficient algorithm is needed.

Furthermore, **Long-Lived Particles** signatures can be easily misinterpreted by conventional trigger logic.



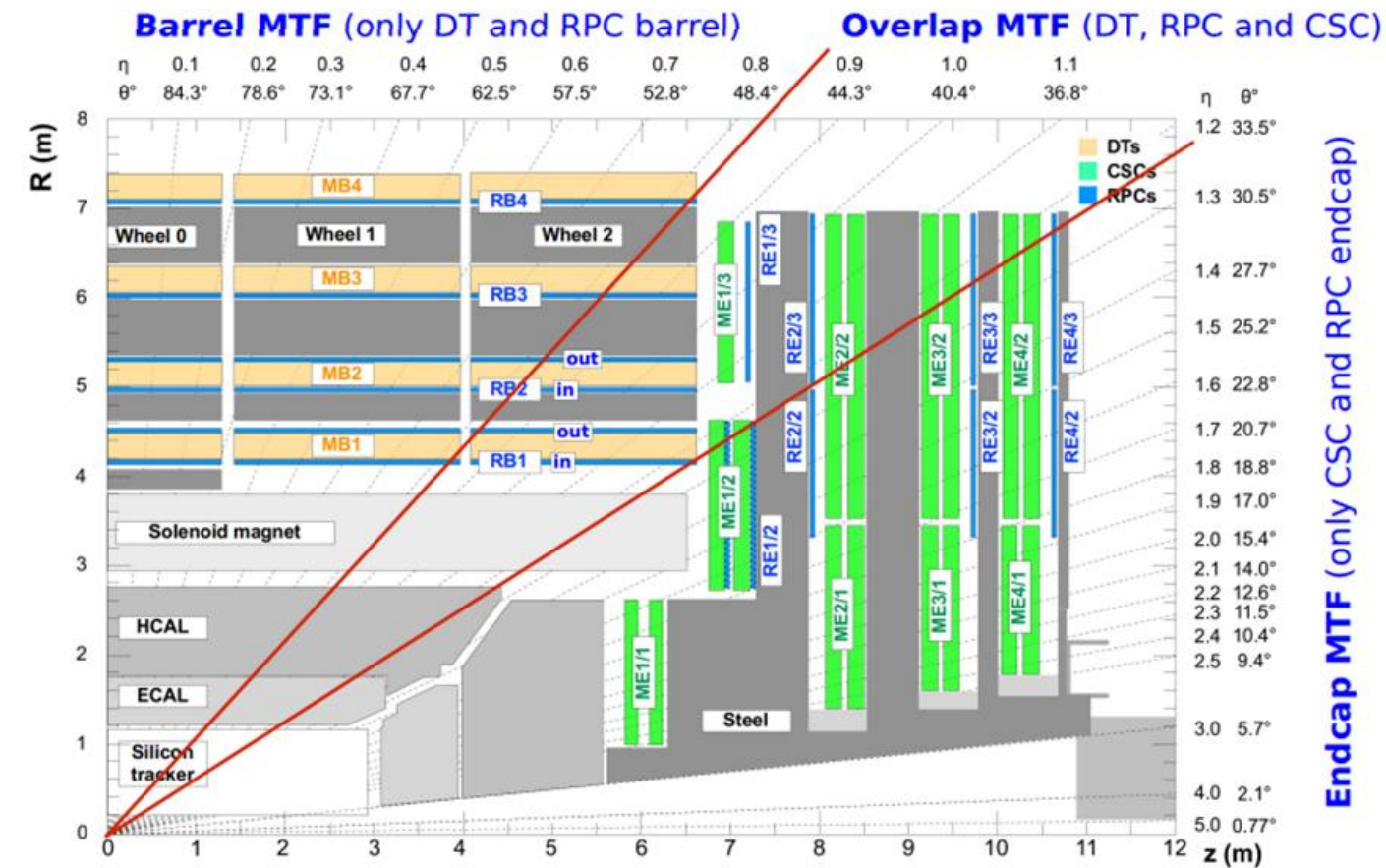
Introduction

The CMS Phase-2 experiment must process **high-throughput detector** data, between 140 – 200 collisions, every 25 ns, i.e., with a certain **latency**. In fact, decisions at **L1 trigger** must be taken in $\sim \mu\text{s}$.



CMS Phase-2 (HL-LHC Run 4, ≥ 2030) — 140–200 pileup interactions per bunch crossing

Introduction



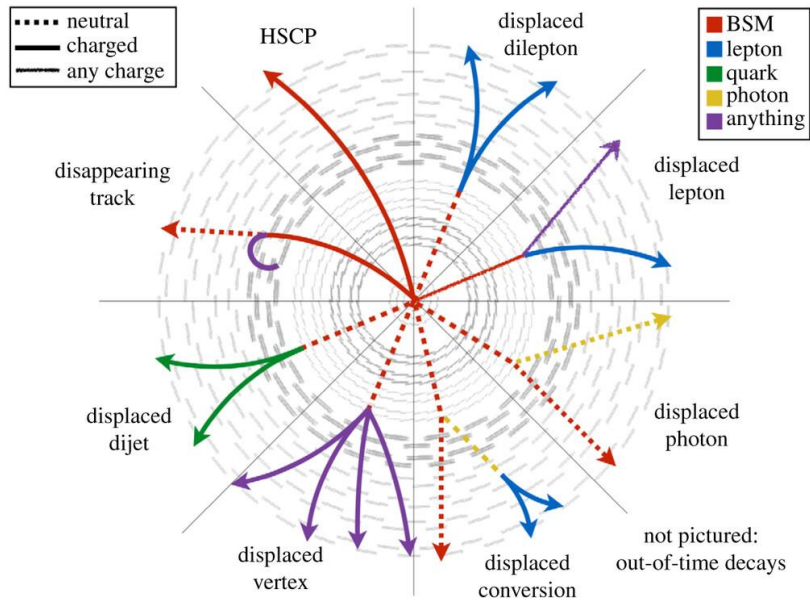
The main goal of this project is to develop a **Graph Neural Network (GNN)** for the muon reconstruction algorithms of the L1 trigger of the CMS experiment for the **HL-LHC**.

We are developing an **edge classifier** to determine whether a track segment is part of a specific trajectory.

Moreover, we are working on performing the **regression** of the charge and transverse momentum of displaced leptons.

However, the new algorithms should be compatible with **FPGA chips**.

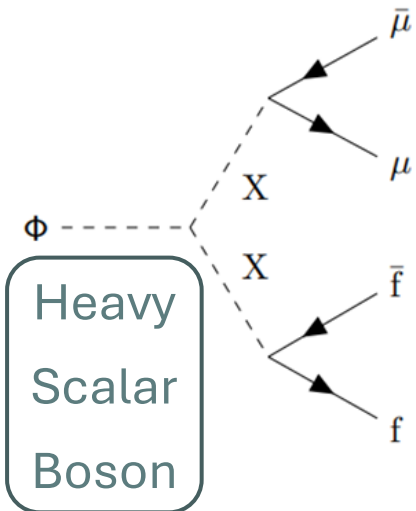
Long-Lived Particles



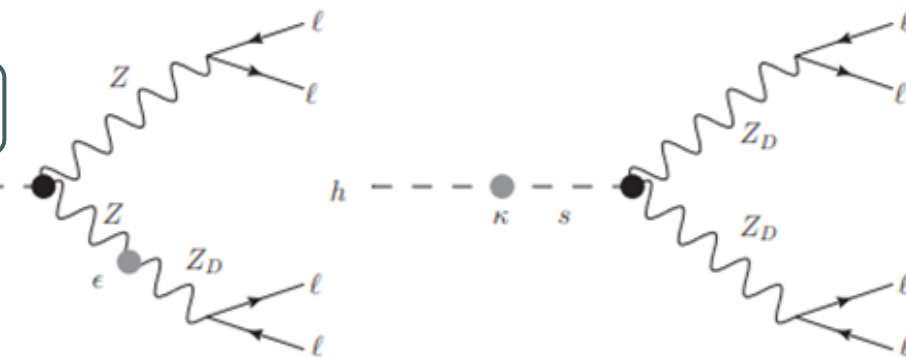
Long-Lived Particles (LLPs) are predicted by many Beyond Standard Model theories.

Experimentally, several **challenges**, e.g., object reconstruction, arise when performing LLP studies.

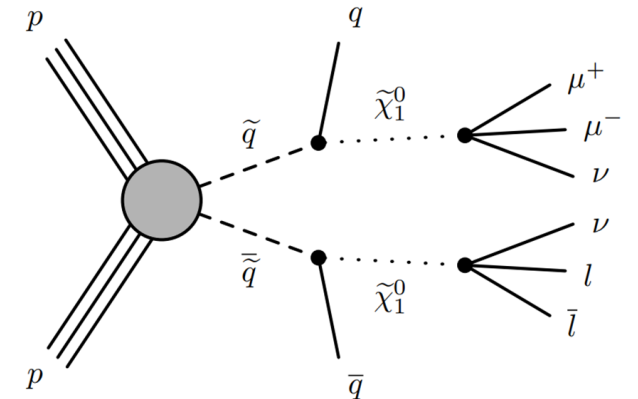
Due to the characteristics of the CMS experiment, **displaced dimuons** analyses stand out.



HAHM



RPV SUSY



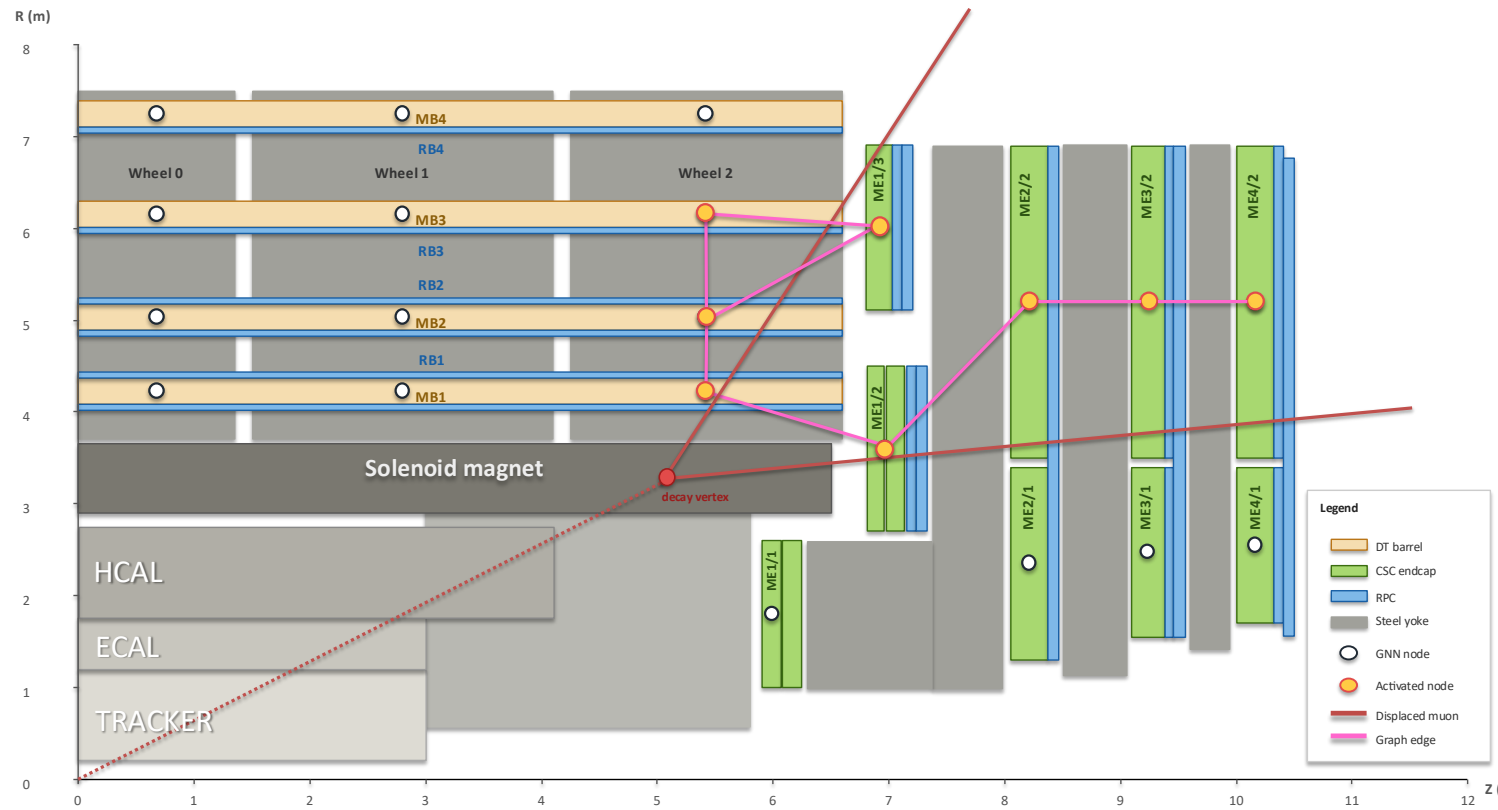
Samples

To **optimize** our models and cover different possible scenarios, we include various kind of samples for the training and the evaluation of results.

Multiplicity

Displaced
Candidates

Isolation



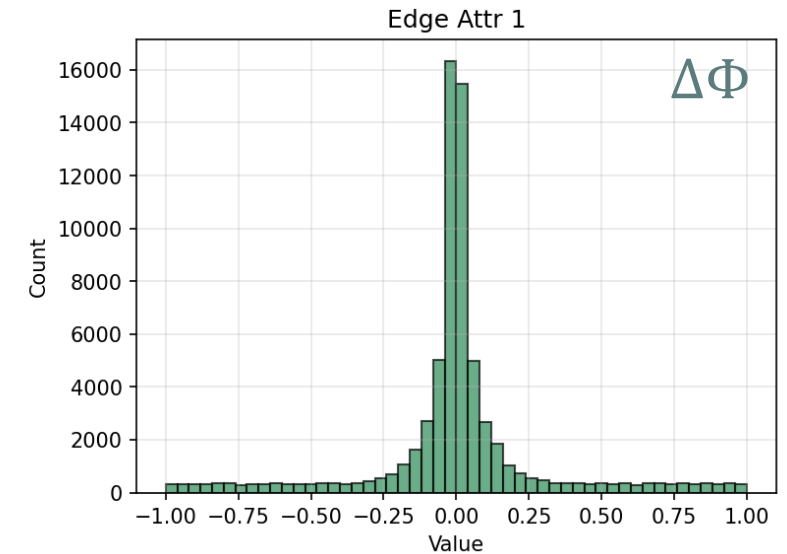
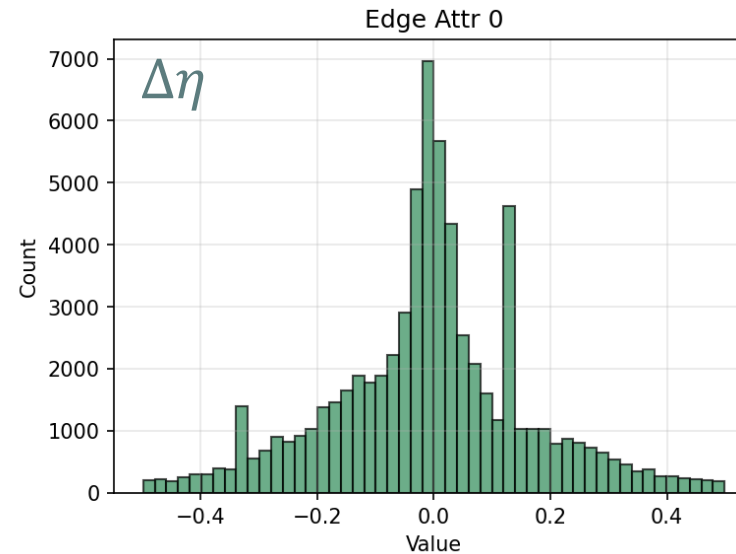
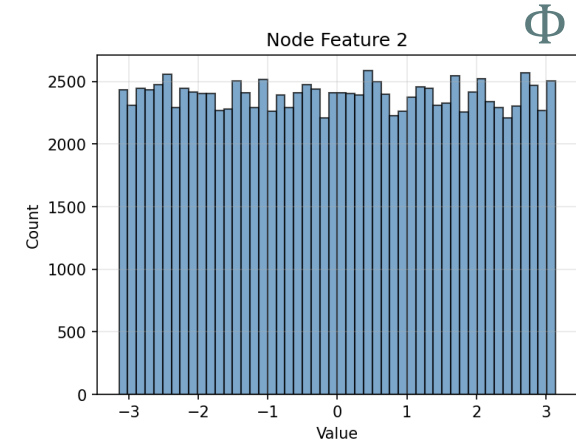
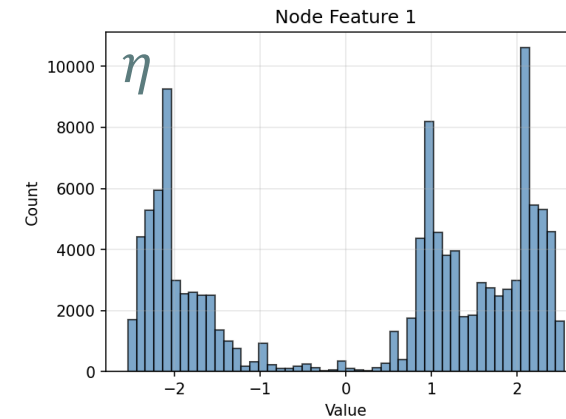
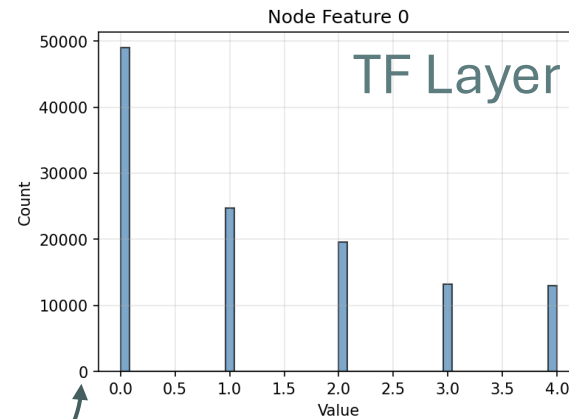
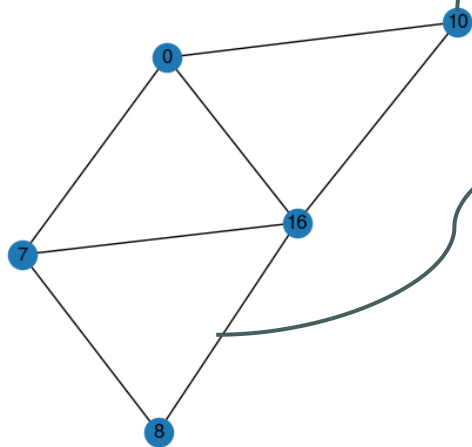
Overlap
Region

High
Pseudorapidity

Graph Generation

Not only the stub properties are used as **node features**, but also some **edge attributes** are computed.

These are the variables employed during the training after **normalized**.



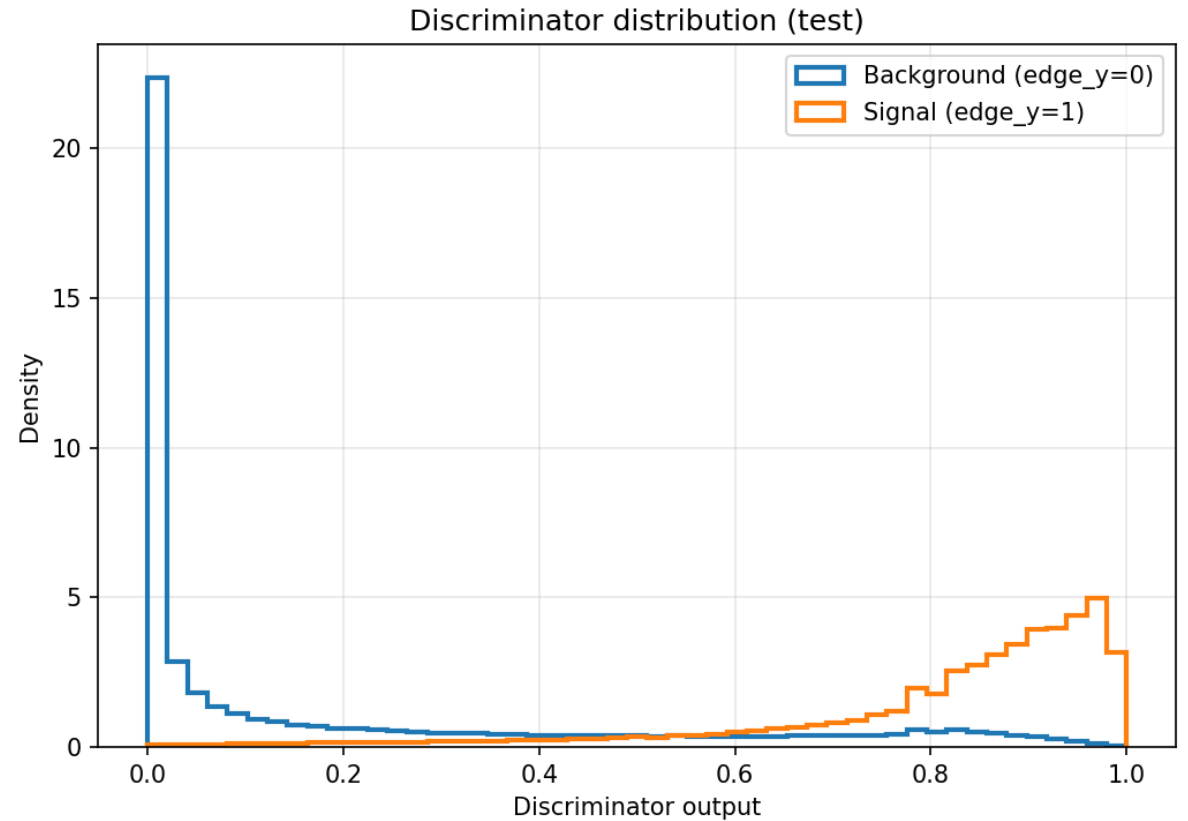
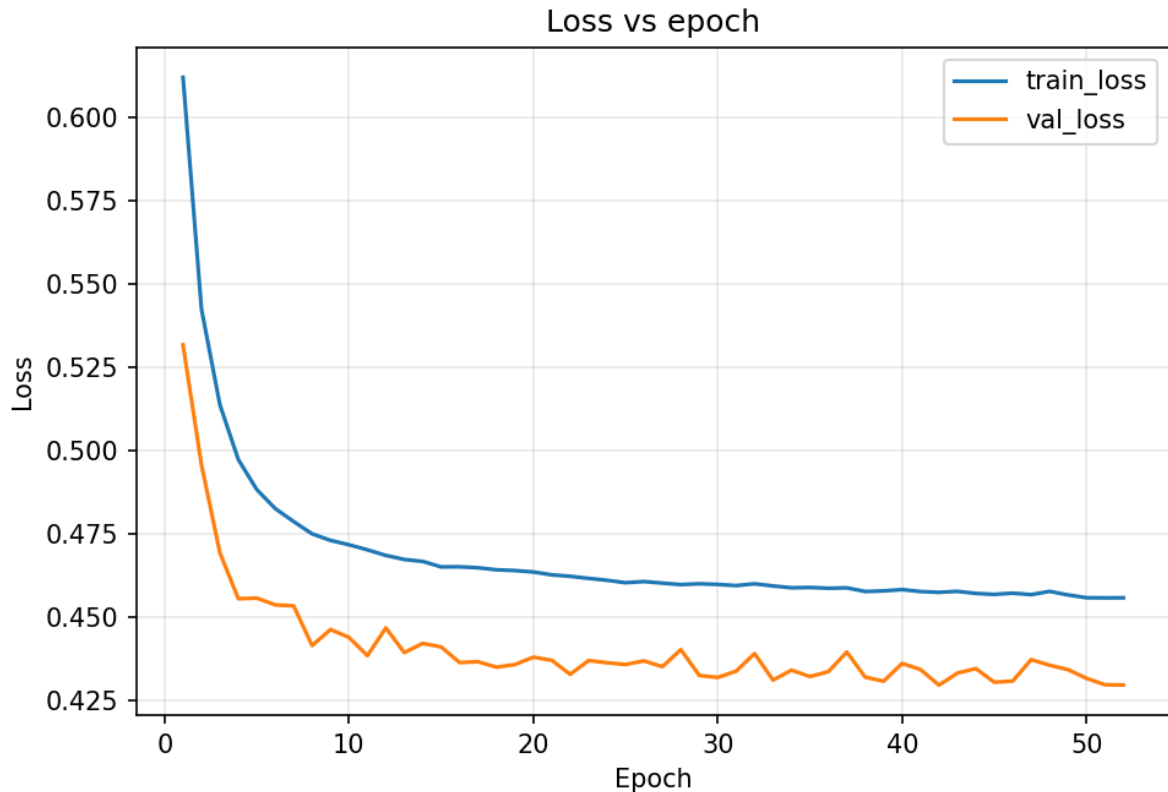
TransformerConv

Three-layer Graph **Transformer** model that combines source and target nodes, as well as edge attributes to build **edge embeddings**. The binary edge classification is performed by an **MLP**.

Layer	Operation	Input Shape	Output Shape	Description
1	Node Encoder	(N, in_channels)	(N, hidden_dim)	Maps node features to hidden embeddings
2	Edge Encoder	(M, edge_dim)	(M, edge_dim)	Encodes edge attributes
3	Transformer Layer 1	(hidden_dim, edge_dim, heads)	(N, 2 hidden_dim)	Adds neighbour information
4	Transformer Layer 2	(2 hidden_dim, edge_dim, heads)	(N, hidden_dim)	Expands neighborhood context
5	Transformer Layer 3	(hidden_dim, edge_dim, heads)	(N, hidden_dim)	Refines node encoding
6	Residual Block	(N, hidden_dim)+(N, hidden_dim)	(N, hidden_dim)	Preserves previous information
7	Edge Representation	(hidden_dim, hidden_dim, edge_dim)	(M, 2 hidden_dim + edge_dim)	Concatenates edge attributes
8	MLP 1	(2 hidden_dim, edge_dim)	(2 hidden_dim)	Learn high-level edge attributes
9	MLP 2	(2 hidden_dim)	(hidden_dim)	Compresses edge attributes
10	MLP 3	(hidden_dim)	(hidden_dim / 2)	Refines edge attributes
11	Output	(hidden_dim / 2)	(1)	Produces the edge logit

Classifier

To **optimize** the model, we apply different techniques, e.g., **early stopping**.

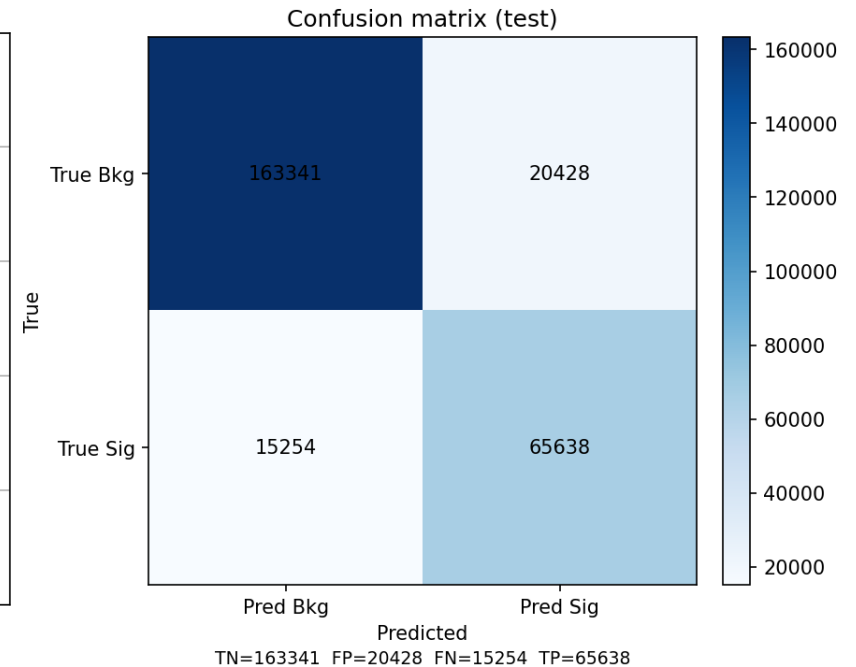
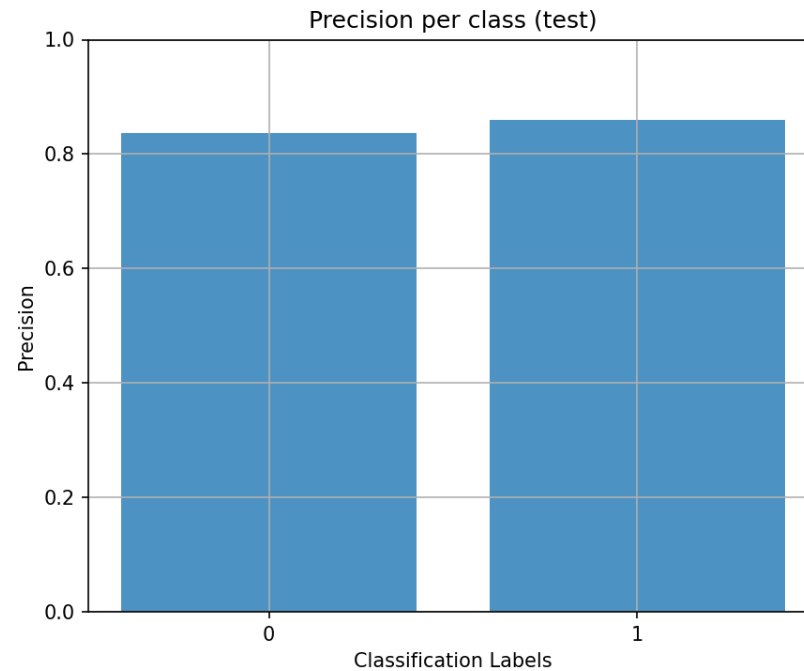
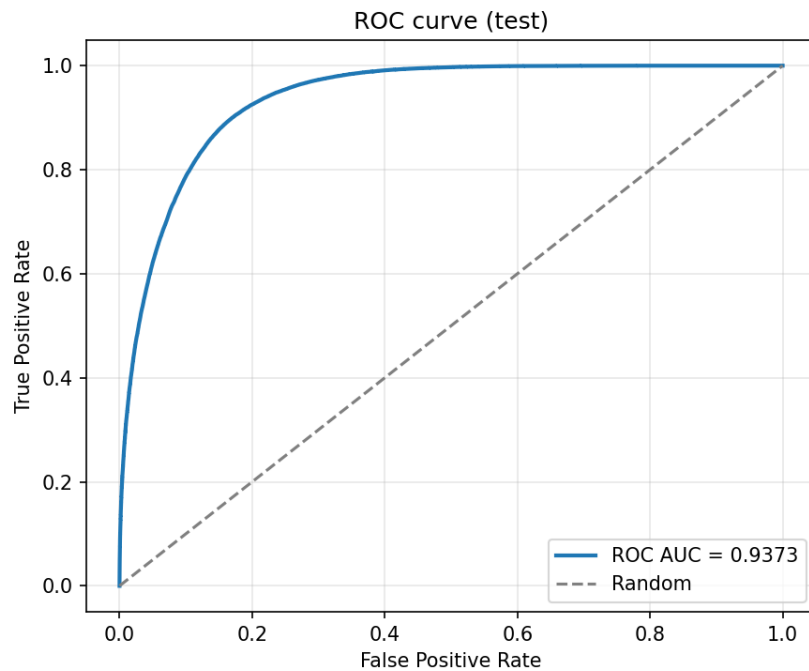


For the **evaluation** of the model, we are including all the different samples.

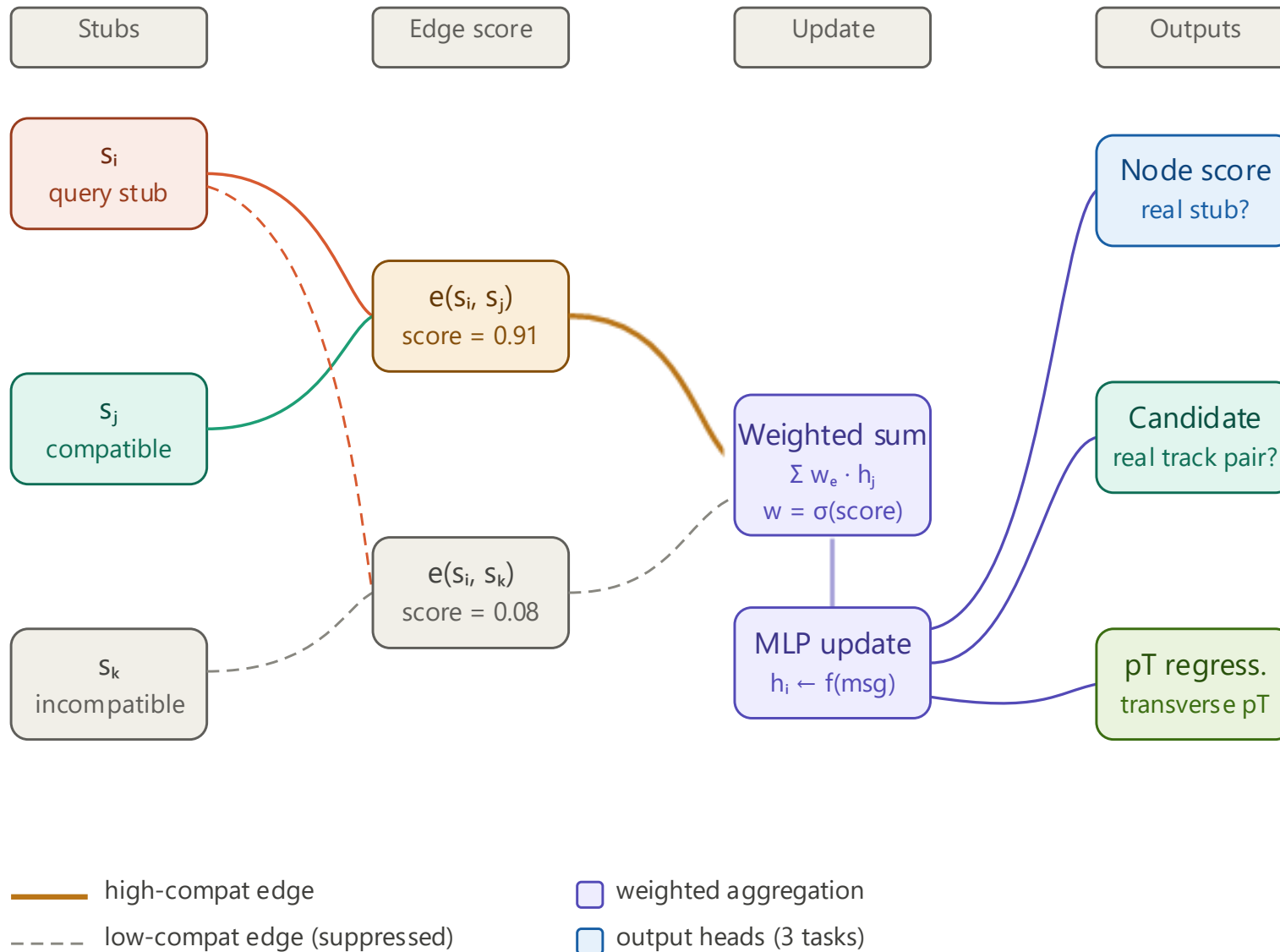
Classifier

To study the **performance** of the model we rely on different **metrics**: ROC curve, AUC value, precision per class, etc.

Accuracy	Precision	Recall	F1
0.8652	0.7626	0.8114	0.7863



Outlook



- Optimization of the model, focusing on **displaced signatures**.
- Include **charge and transverse momentum regression**, specifically for low p_T muons.
- **Implementation on FPGAs** → see [Santiago's talk](#).
- **Deploy** of the algorithm on real detectors. Test the method at the restart of data-taking to, later, **integrate** it in the L1 trigger.

THANK YOU
FOR THE ATTENTION!

BACKUPS

Graph Generation

Sample *	Description
G1	One isolated prompt candidate in the overlap
G2	One prompt candidate with pileup
G3	One isolated displaced candidate in the overlap
G4	One displaced candidate with pileup
G5	Two displaced candidates with pileup
G6	Three prompt candidates with pileup
G7	No overlap candidates (only barrel hits)
G8	No overlap candidates with pileup
G9	High eta region (endcap)
G10	High eta region (endcap) with pileup
B4	Only pileup

* Most of the samples, except G7 and G8, are divided in positive and negative η region.