Background reduction with HGTD for the search for LLPs

Leo Reynaud
Université de Grenoble-Alpes
Master 2 PSC

Supervised by Dr. Louie D.Corpe

Motivation

The main objective of these studies is to know if we can use HGTD to do an LLPs search in the region it covers.
For this, we need to see if it can be used for a satisfactory BIB reduction.

Long Lives Particles (LLPs) are particles with macroscopic lifetime found in SM and BSM models. The search of LLPs is a big part of the ATLAS exotic program, led by the subgroup UEH (Unconventional Signatures and Exotic Higgs). In our case, we have a benchmark model that involves a Hidden Sector representing our LLPs, and where SM and HS are connected *via* a scalar neutral boson Φ with low coupling.

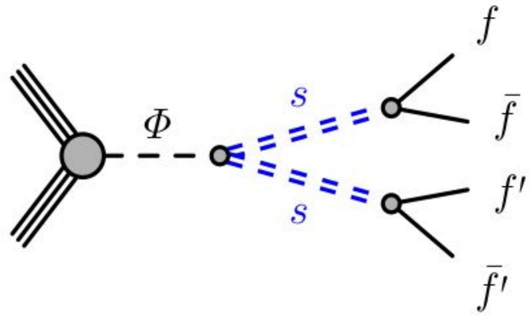


Diagram of $\Phi \rightarrow ss \rightarrow ff'f'$ decay used as the benchmark model.

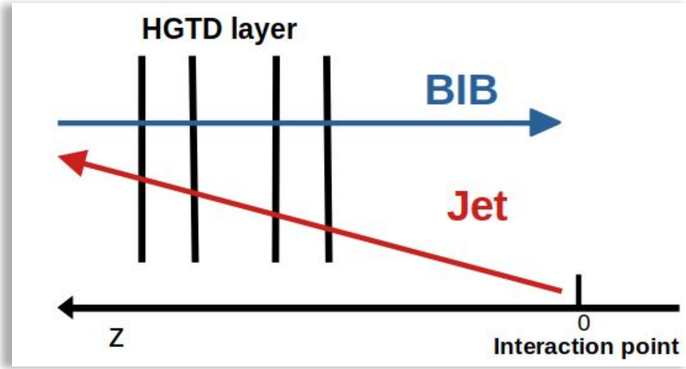
The **Beam Induced Background (BIB)** is a background coming from the interaction between the proton beam and its environment. There are two types of BIB:

- ➢ **BIB Gas** : Beam interaction with residual gas
- ➢ **BIB Halo** : Beam interaction with the LHC structure.

BIB is one of the major difficulties in the search for LLPs because it produces displaced activity similar to LLP signal.

Main objective

We want to know if we can differentiate BIB from jets from the interaction point using HGTD.



HOW?

We want to exploit the time resolution of HGTD hits to distinguish inward-going activity (BIB) from outward-going activity (SM or signal) using a linear for of z versus t.

We want to mix the BIB and the jet event by event, make a clustering in each event to separate the different tracks and make a linear fit to know the direction of each track

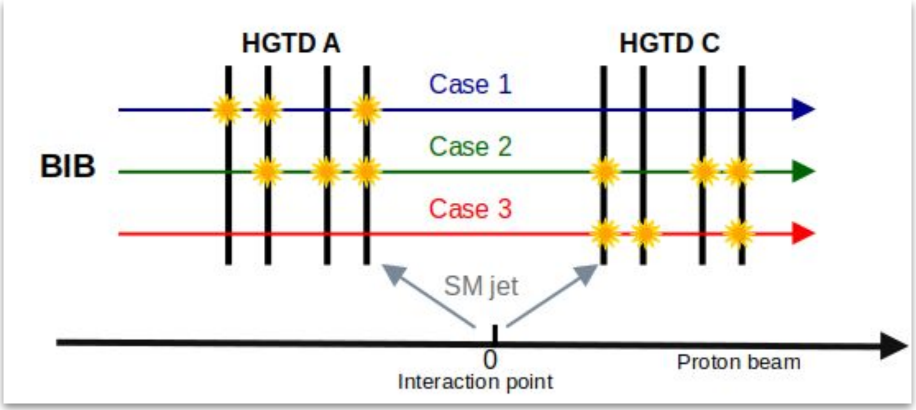
- ➢ Using **Anti-KT** algorithm with **FastJet** for spatial clustering.
- ➢ **DBSCAN** algorithm for further time-based clustering.
- ➢ Linear fit ***t = az + b*** with **Minuit** and know the direction with the sign of ***a***

Three different cases

With our method, we will only look at the side where BIB and SM jet come from different directions.

This method will not work for every BIB event. Indeed, we have three cases.

- Case 1: BIB only hits *HGTD A*. In this case we can make the rejection with our method without any problem.
- Case 2: BIB hits both *HGTD*. We could make the rejection for *HGTD A* and use the tracker to make the rejection on the whole track.
- Case 3: BIB only hits *HGTD C*. In this case we cannot make the rejection and we need to find another method, such as comparing flight times.



Number of hit in the case 1: ~ 77 %
Number of hit in the case 2: ~ 14 %
Number of hit in the case 3: ~ 9 %

Samples

BIB

Event generated by Monte Carlo simulation using FLUKA software. **Only a unidirectional** beam.

Sample name

mc15_13TeV.309680.BeamHaloGenerator_BeamGasB1_20MeV.evgen.EVNT.e6513
mc15_13TeV.309682.BeamHaloGenerator_BeamGasB1_20GeV.evgen.EVNT.e6513
mc15_13TeV.309681.BeamHaloGenerator_BeamHaloB1_20MeV.evgen.EVNT.e6513
mc15_13TeV.309683.BeamHaloGenerator_BeamHaloB1_20GeV.evgen.EVNT.e6513

→ We take only non-empty events. Between one or two particles (~ 5-25 hit) for each event

SM events

Events representing the decay of a top quark and anti-quark from the interaction point.

Sample name

EVNT_ttbar.32468304._002312.pool.root.1

→ ~ 500-4000 hit for each event

Pileup events

Pileup represents the decay of particles from the interaction point for low and high energies.

Sample name

mc15_14TeV.800380.Py8EG_A3NNPDF23LO_minbias_inelastic_low_keepJets.evgen.EVNT.e8205
mc15_14TeV.800381.Py8EG_A3NNPDF23LO_minbias_inelastic_high_keepJets.evgen.EVNT.e8205

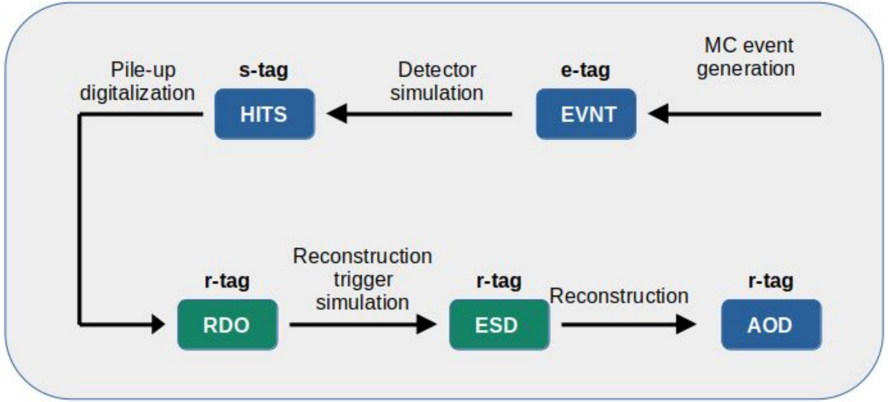
→ Low events with ~10-100 hit

High events with ~130-500 hit

Geometries

Simulation

- Using s4038 GEANT4 geometry to simulate hits with the HGTD (s-tag).
- Using SiHitAnalysis to convert the local coordinates to global coordinates.



The reconstruction gives us a lot of hit not linked to the pdg index and we don't know why there are so many. For example, for one top event, numbers are PDG index and represent real particles, *-9999.0* are “False particles”. We have **~13%** of real particles and **~87%** of other hit.

pdg index for top event

```
[-9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0,
 -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0,
 -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0,
 -9999.0, 11.0, 11.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999
.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -9999.0, -321.0, -321.0, -321.0]
```

For the rest of the study, we will consider every hit without distinction.

Final samples

```python
        low = 209.2692
        high = 0.725172
        for i in range(NbPileup):
            PileupUse = []
            low_high = random.uniform(0, low + high)
            if low_high < high:
                EventPileup = random.randint(0,len(x_highP)-1)
                if EventPileup not in PileupUse:
                    x1.extend(x_highP[EventPileup])
                    y1.extend(y_highP[EventPileup])
                    z1.extend(z_highP[EventPileup])
                    t1.extend(t_highP[EventPileup])
                    R1.extend(R_highP[EventPileup])
                    TruePart1.extend(True_highP[EventPileup])
                    pdg1.extend(pdg_highP[EventPileup])
                    PileupUse.append(EventPileup)
            else:
                EventPileup = random.randint(0,len(x_lowP)-1)
                if EventPileup not in PileupUse:
                    x1.extend(x_lowP[EventPileup])
                    y1.extend(y_lowP[EventPileup])
                    z1.extend(z_lowP[EventPileup])
                    t1.extend(t_lowP[EventPileup])
                    R1.extend(R_lowP[EventPileup])
                    TruePart1.extend(True_lowP[EventPileup])
                    pdg1.extend(pdg_lowP[EventPileup])
                    PileupUse.append(EventPileup)
```

Proportion of low and high

Number of pileup for each event

Random selection between low and high

Random selection between our 2 000 high pileup event

Random selection between our 20 000 low pileup event

For each final event, we combine randomly one BIB event, one top event and N pileup event among 3 000 BIB events, 1 000 Top events and 22 000 pileup events.

For the pileup, we dispose of 20 000 low events and 2 000 high events. The low and high pileups are added randomly according to a certain proportion, ~99.65% of low and ~0.35% of high
We are careful not to take the same pileup event twice.

Choose randomly

Choose randomly with proportion

[[BIB event + Top event + N Pileup event], ← Final event 1
[BIB event + Top event + N Pileup event],

.
.
.

[BIB event + Top event + N Pileup event]] ← Final event M

Final samples

Clustering with Anti-KT

For the first clustering, we use Anti-KT algorithm with FastJet python library

Anti-KT use quadri-vector energy-momentum and radius for the clustering.
The radius will be described in the next slide
For the momentum, we use ***p=md/t*** with the approximation of the mass ***m = 1***
For the total energy, we use that deposited in the layer.
It's a lot of approximation, but it works well!

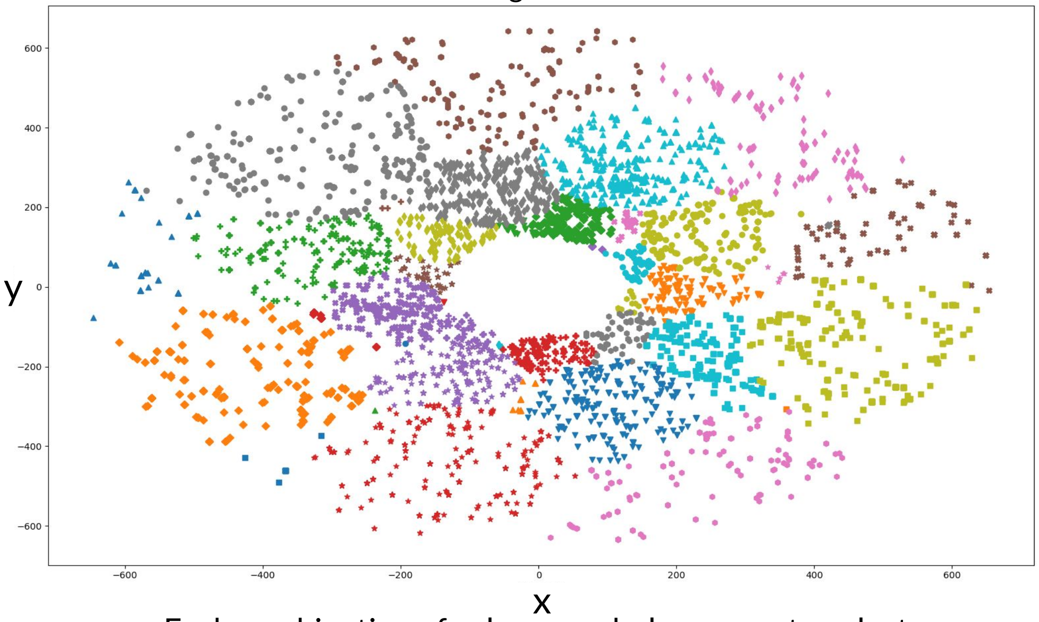
```python
def FastJetCluster(vector,radius):
    constituent_index = []
    for event in vector:
        jetdef = fastjet.JetDefinition(fastjet.antikt_algorithm, radius)
        cluster = fastjet.ClusterSequence(event, jetdef)
        constituent_index1 = cluster.constituent_index()
        constituent_index1 = constituent_index1.to_list()
        constituent_index.append(constituent_index1)
    return constituent_index
```

Implementation of the algorithm

Creation of cluster

Return list of cluster

Clustering with Anti-KT



Each combination of color + symbol represents a cluster

Eventually, this step would be replaced by the standard clustering at AOD level

Performance of Anti-KT

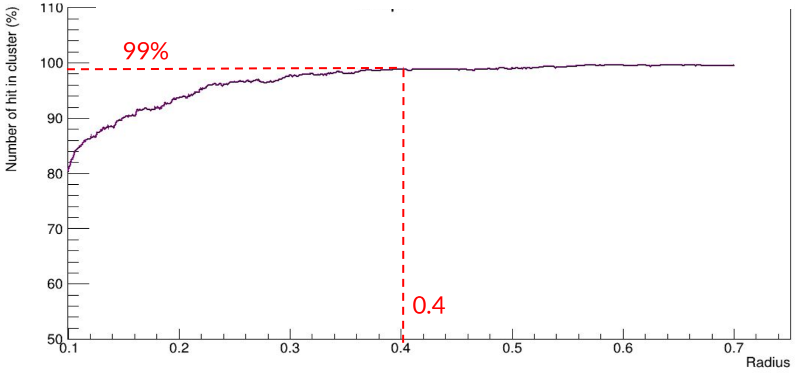
Best radius

We know that the radius used by ATLAS is 0.4, we will try to look if it's a good radius for our studies using two factors:

- ➢ The total number of clusters that we manage to differentiate, whether it will be BIB, top or pileup.
- ➢ The total number of hit that our clustering covers.

If we take these two factors into account, 0.4 is indeed a good radius. Giving us an efficiency on the differentiation of clusters of ~80.7% for a number of hit included of ~99%.

Percentage of hits included in the clustering



However, the rate of differentiation is not very high and we can see one of the obvious causes for this. Clustering appears to pack together particles that hit the HGTD at the same location and with similar energy but with different times.
Here is one cluster of BIB event with Anti-KT where we can see three different particles at three different times (red, blue, and green).

Time (ns) of a BIB event

```
[-10.731889724731445, -10.69443130493164, -10.630202293395996,
-11.531757354736328, -11.447617530822754, -11.531797409057617,
-11.447623252868652, -10.334186553955078, -10.259435653686523]
```

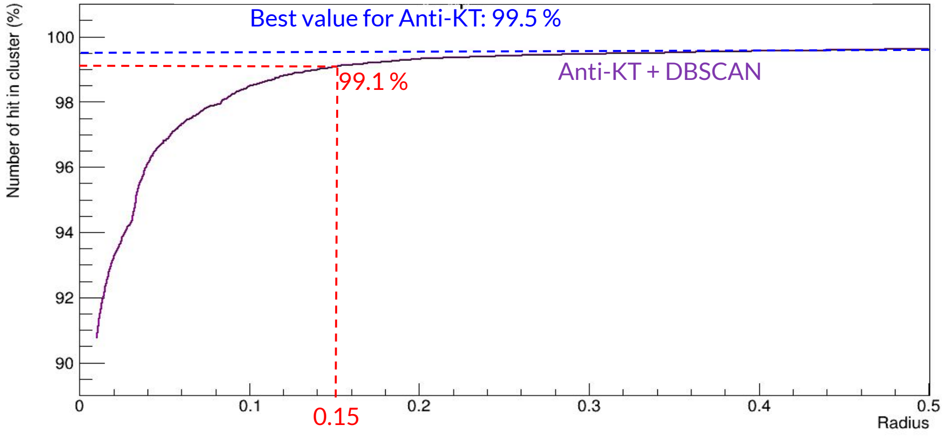
Temporal clustering with DBSCAN

Therefore, to improve our clustering, we will do a second clustering using DBSCAN inside each Anti-KT cluster

DBSCAN is a clustering algorithm that uses two main parameters, "***eps***" (epsilon) for the radius and "***min_samples***" for the minimum number of neighbors that a point must have to be considered as a kernel.

Min_samples will always be set up at two hits minimum and we will use the same method as for anti-kt to find the best value of *eps*.

Percentage of hits included in the clustering



Radius Anti-KT: 0.4
Radius DBSCAN: 0.15

- ~ 89.5% Differentiable Clusters
- ~ 99% Hits included

DBSCAN multiplies the number of clusters by 5 and isolates hits with improbable times.

Fit with Minuit

Minuit python library is used for the linear fit

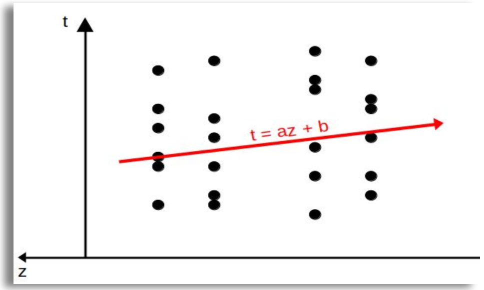
Fit: $t = az + b$

The goal is to accurately fit each cluster, which is depicted as a cloud of dots, and determines the overall orientation of the cluster along the z-axis. The sign of ***a*** gives us the direction of the track and allows us to determine if it's BIB or top/pileup.

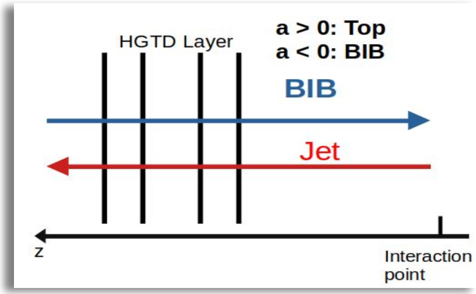
- ***a* > 0 = Top**
- ***a* < 0 = BIB**

Minuit uses the least squares method for the fit, so we can make an estimate of the reliability of the fit using the covariance matrix, but we have not exploited it yet.

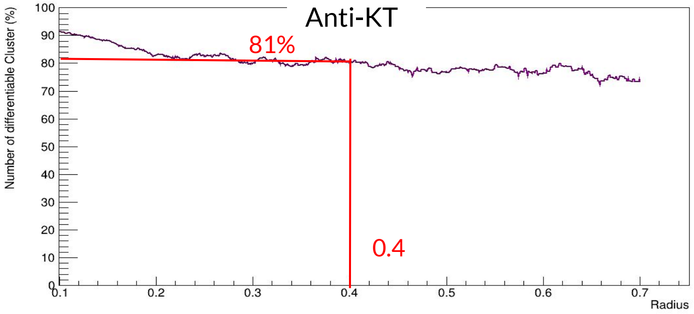
Cluster fit

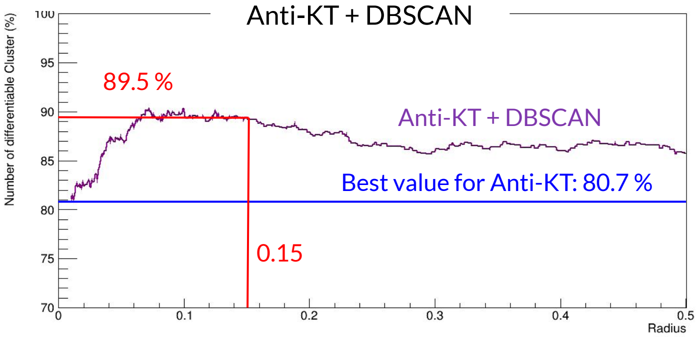


a value



These two graphs represent the percentage of clusters that we manage to differentiate, whether it will be BIB, top or pileup with Anti-KT and Anti-KT + DBSCAN.





Analysis 1

The analysis will be led by a normal way, using a confusion matrix.

We will run our program for five different number of pileup event: 0, 10, 25, 50, 100, 150, 200.
One run is composed of 500 final samples, and each of them is made out of 400 “final events”.

For each final sample, we record a confusion matrix and present the average of all the matrix, as well as their statistical variation.

Sample	BIB	Top
a > 0	B1	T1
a < 0	B2	T2
total	B3	T3

B2 and T1 are the number of BIB and top clusters that we manage to differentiate.

B1 and T2 are the number of BIB and top clusters that we wrongly differentiate.

Table of efficiency (confusion matrix)

True positive %	$100 \times B2/B3$
False positive %	$100 \times T2/T3$
True negative %	$100 \times B1/B3$
False negative %	$100 \times T1/T3$

True positive is the rate of BIB well differentiate.

False positive is the rate of top/pileup differentiate as BIB.

True negative is the rate of BIB differentiate as top.

False negative is the rate of top/pileup well differentiate.

Analysis 2

Example for the case with 50 Pileup.

500 Sample

Sample

Event

[1 BIB event + 1 Top event + 50 Pileup event]

Choose randomly

Choose randomly with proportion of low and high

x 400

We reproduce the creation of an event 400 times to produce a sample. The BIB and top events used once will not be reused in a sample, unlike the pileup event, which will be reused several times.
We will do clustering, fit and confusion matrix for an entire sample.

x 500

We reproduce the creation of one sample 500 times. We obtain 500 confusion matrixes for each pileup value, which allows us to average and have an uncertainty.

We will reproduce all this step for a number of pileups of 0, 10, 25, 50, 100, 150 and 200.

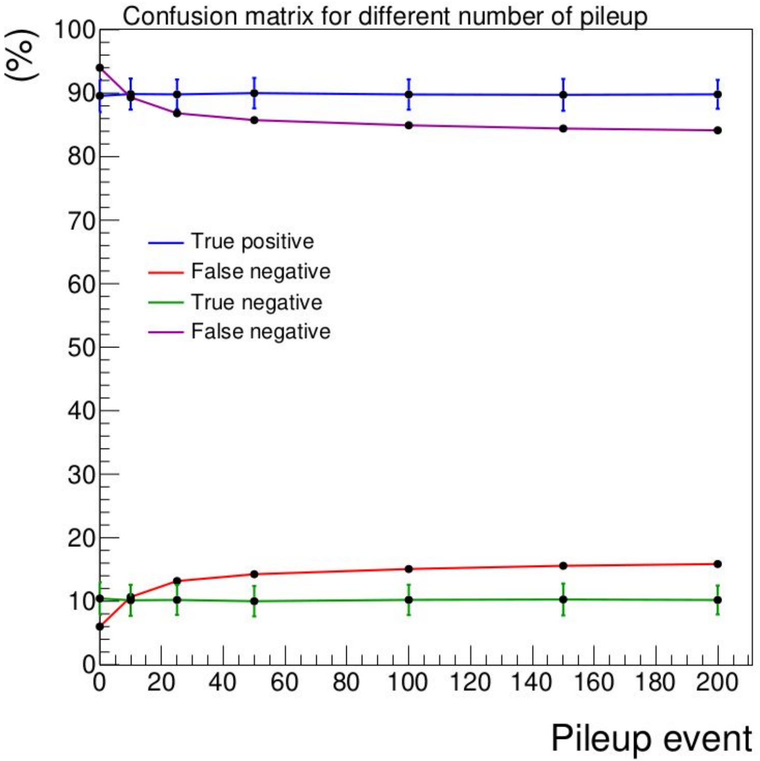
Results

Results for different numbers of pileups

Pileup per event	0	50	100	200
True positive %	$89.6 \pm 2.5\%$	$90.0 \pm 2.4\%$	$89.8 \pm 2.3\%$	$89.8 \pm 2.3\%$
False positive %	$6.0 \pm 0.2\%$	$14.2 \pm 0.2\%$	$15.1 \pm 0.2\%$	$15.9 \pm 0.2\%$
True negative %	$10.4 \pm 2.5\%$	$10.0 \pm 2.4\%$	$10.3 \pm 2.5\%$	$10.2 \pm 2.3\%$
False negative %	$94.0 \pm 0.2\%$	$85.8 \pm 0.2\%$	$84.4 \pm 0.2\%$	$84.1 \pm 0.2\%$

The uncertainties are statistical, calculated considering the standard deviation.

We can see that BIB is not influenced by the presence of pileup. It comes from a different point than the top or the pileup and Anti-KT can easily isolate it.
The jet event is more sensitive to the presence of pileup, causing a perturbation with a small number of pileup events. The efficiency loss is approximately 8% when adding 50 pileup events, but it undergoes minimal changes beyond that point.



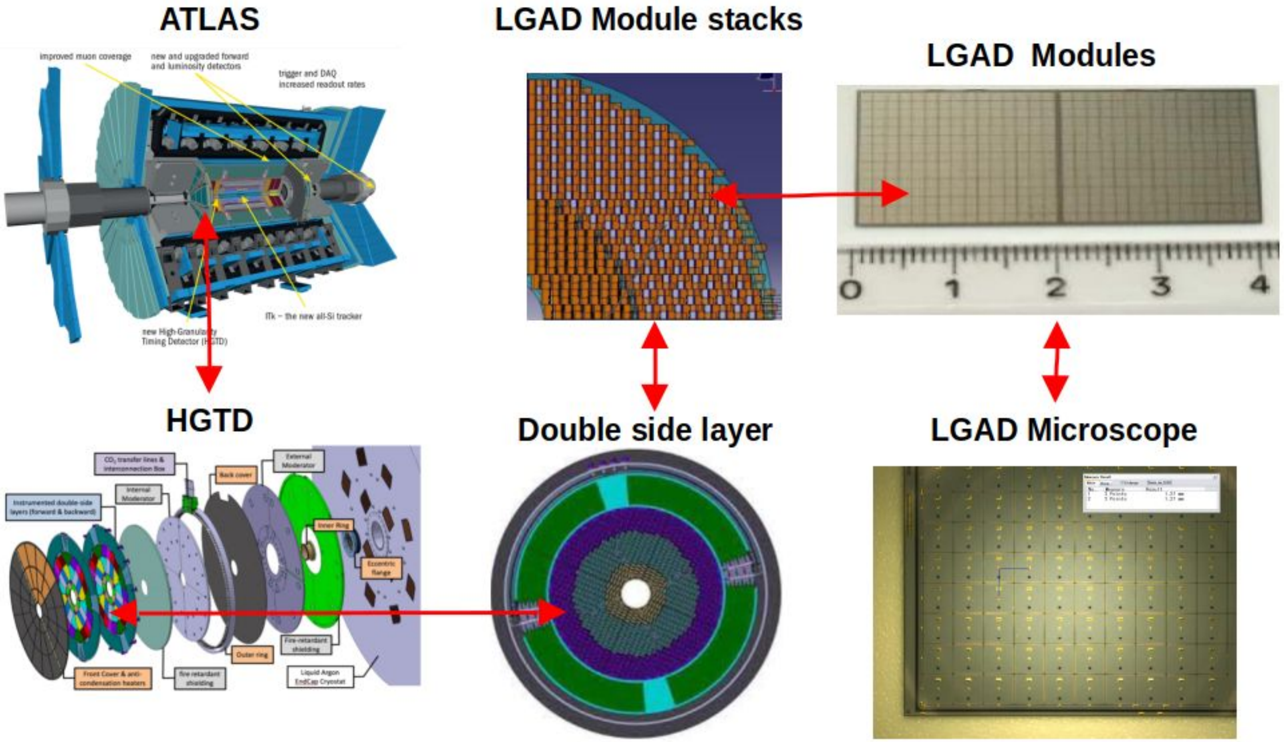
Conclusion

The main objective was to conduct a preliminary study to determine if the HGTD could contribute to reducing the BIB and, consequently, aid in the search for LLPs.
The method used involves several approximations that could be improved, particularly in the creation of the energy-momentum four-vector, the search for an ideal radius for DBSCAN, or the simulation of the samples themselves, which appears to generate too many "false hits."
However, we can still conclude that such a method can be effective in rejecting the BIB, and the HGTD has the potential to become a relevant tool in the search for LLPs in the long term.

Project GitLab: https://gitlab.cern.ch/lreynaud/background-eduction-with-hgtd
Main: - fast.py
- fast_library.py

High Granularity Timing Detector (HGTD)

Composed of two double-sided layers, each of which containing thousands of silicon modules, the HGTD stands out for its high temporal resolution.



Characteristics

$2.4 < |\eta| < 4.0$
$3500\text{mm} < |z| < 4600\text{mm}$

Resolution

$1.3 \times 1.3\ \text{mm}^2$
50 *ps*