أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Unsupervised learning for robust clustering in irregular detector regions

Mozoon Alkhalis

mozoon.alkhlis@kaust.edu.sa

Abstract

This project investigates the robustness of CLUE clustering under irregular detector conditions in the context of the CMS experiment at the LHC. Synthetic detector data was modified to simulate V-shaped cracks, longitudinal gaps, added noise, and inactive detector regions. CLUE was evaluated directly on these datasets and then combined with UMAP to study its effect on clustering performance. UMAP + CLUE improved the clustering agreement across all tested conditions, motivating a further investigation into the changes introduced by UMAP and their effect on CLUE.

The investigation showed that changes in the local neighbourhood structure were more closely related to CLUE improvement than simply increasing cluster separation. Based on these findings, local adaptive scaling, locally weighted neighbourhood smoothing, and a combination of the two were tested as simpler preprocessing approaches. The combined approach improved the ARI on the ideal dataset from 0.830 to 0.916, compared with 0.972 for UMAP + CLUE. These preliminary results suggest that part of UMAP's benefit may be reproduced using a simpler preprocessing step.

Date: August 24, 2026

Code: https://github.com/Mozoonn/robust-clustering-irregular-detectors

Contents

1 Introduction **2**

2 Background **2**

2.1 CMS Detector and Clustering . . . 2

2.2 CLUEstering . . . 3

2.3 UMAP . . . 3

2.4 Clustering Evaluation Metrics . . . 4

3 Methodology **5**

3.1 Synthetic Detector Dataset Generation . . . 5

3.2 Detector Irregularity Modeling . . . 5

3.3 Experimental Pipeline . . . 7

3.4 Evaluation Procedure . . . 7

4 Results **8**

4.1 CLUE-only Performance . . . 8

4.2 UMAP + CLUE Performance . . . 9

5 Investigation of UMAP's Effect on CLUE **9**

5.1 Local Neighbourhood Scaling . . . 10

5.2 Cluster Separation and Neighbourhood Purity . . . 10

5.3 Connection to CLUE Improvement . . . 10

5.4 Preliminary UMAP-Inspired Preprocessing . . . 11

6 Limitations and Future Work **12**

7 Conclusion **13**

References **14**

1 Introduction

Particle detectors are not perfectly uniform systems and can contain gaps, cracks, or faulty modules. These irregular regions can make pattern recognition techniques less efficient. Clustering techniques such as CLUEstering [1] are suitable for these tasks because they can be robust to irregular and noisy data. Dimensionality reduction techniques can also transform the structure of the data representation in ways that may affect clustering performance. Therefore, this project investigates whether UMAP [2] can improve the robustness of clustering in irregular detector regions. The study evaluates CLUEstering both independently and in combination with UMAP using synthetic detector data under several controlled irregularities, including V-shaped cracks, longitudinal gaps, added noise, and inactive detector modules. The two approaches are compared using cluster reconstruction performance and multiple clustering validation metrics. Since UMAP was found to improve clustering performance across the tested conditions, the final stage of the project begins investigating how UMAP changes the data representation and which of these changes contribute to the improved CLUEstering results. The longer-term goal is to determine whether these beneficial effects can be reproduced through preprocessing suitable for use with CLUEstering in a production environment.

2 Background

2.1 CMS Detector and Clustering

The Compact Muon Solenoid (CMS) experiment [3] is one of the general-purpose particle physics experiments at the Large Hadron Collider (LHC). Particles produced in proton–proton collisions pass through different layers of the CMS detector, including the silicon tracker, electromagnetic calorimeter (ECAL), hadron calorimeter (HCAL), and muon system. These detector systems provide measurements that are used to reconstruct the trajectories and energies of particles produced in the collisions. Clustering plays an important role in particle reconstruction by grouping nearby energy deposits that are likely to originate from the same particle. However, detector geometries are not perfectly continuous and may contain inactive regions, gaps, cracks, or noise. As illustrated in Figure 1, these effects can modify the spatial distribution of detector measurements by removing expected energy deposits or introducing additional signals. Such irregularities can make the reconstruction of the underlying cluster structure more challenging and motivate the investigation of robust clustering methods.

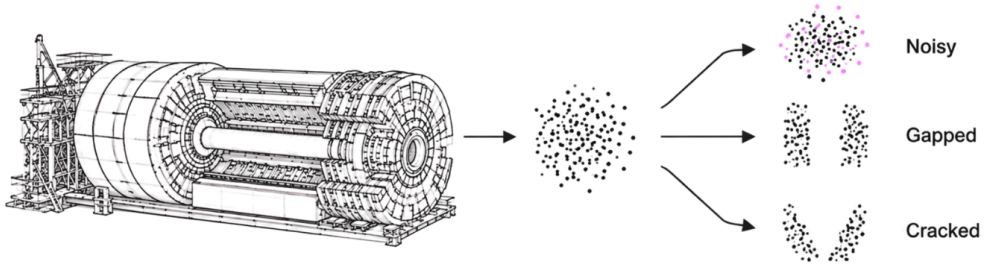


Figure 1 Illustration of some detector irregularities considered in this work, including added noise, gaps, and cracks.

2.2 CLUEstering

CLUEstering is a general-purpose weighted density-based clustering algorithm that can be applied to high-energy physics and detector reconstruction tasks [1]. It groups nearby data points into clusters based on their local density while identifying isolated points as outliers. The local density is calculated using the weights associated with neighbouring points, which in detector applications can represent quantities such as deposited energy. The algorithm is controlled by parameters including the density radius d_c and the minimum density threshold ρ_c. The parameter d_c defines the neighbourhood considered when estimating the local density around each point, while ρ_c determines the minimum density required when identifying cluster structures. These parameters influence the formation of clusters and the treatment of isolated or noisy points. An illustration of the CLUEstering algorithm is shown in Figure 2.

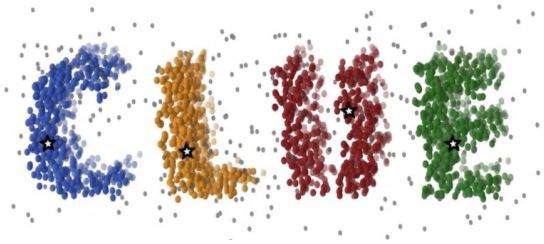


Figure 2 Official logo of the CLUEstering algorithm [1].

2.3 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a dimensionality reduction technique used to create a lower-dimensional representation of high-dimensional data [2]. Mathematically, UMAP builds a weighted graph from the high-dimensional data, where the edge weights represent how strongly points are connected based on their local neighbourhoods. It then creates another graph in the low-dimensional space and optimizes the positions of the points by minimizing the difference between the two graphs. The main parameters considered in this project are `n_neighbors`, which determines the neighbourhood size; `min_dist`, which controls how tightly points can be grouped; `n_components`, which determines the dimensionality of the reduced space; and `metric`, which determines how distances between data points are calculated. Figure 3 illustrates the transformation from the original data space to a lower-dimensional UMAP representation.

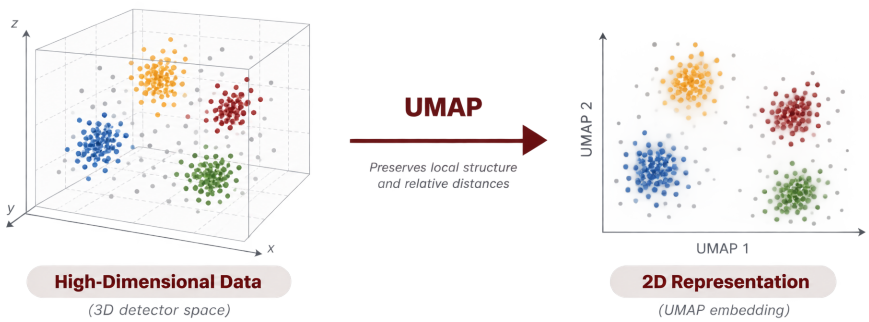


Figure 3 Illustration of UMAP transforming high-dimensional data into a lower-dimensional representation.

2.4 Clustering Evaluation Metrics

To quantitatively evaluate the clustering performance, several validation metrics were used throughout the project. The initial evaluation was based on three internal clustering metrics: the Silhouette Score, Dunn Index, and Davies–Bouldin Index. As the experimental methodology was further developed, the Adjusted Rand Index (ARI) was added to the evaluation process to compare the predicted clustering directly with the known ground-truth cluster labels.

2.4.1 Silhouette Score

The Silhouette Score evaluates both the compactness and separation of clusters [4]. For each point i, the Silhouette coefficient is calculated as

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{1}$$

where $a(i)$ is the average distance between point i and the other points in its own cluster, and $b(i)$ is the smallest average distance between point i and all points in any other cluster. The score ranges from -1 to 1. Values close to 1 indicate compact and well-separated clusters, values close to 0 indicate overlapping clusters or points near cluster boundaries, and negative values may indicate incorrect cluster assignments.

2.4.2 Dunn Index

The Dunn Index evaluates cluster separation and compactness by comparing the minimum distance between clusters with the maximum diameter of a cluster [5]. It is conceptually defined as

$$D = \frac{\text{minimum inter-cluster distance}}{\text{maximum intra-cluster diameter}}. \tag{2}$$

A higher Dunn Index indicates better clustering, meaning that the clusters are more separated relative to their internal spread. The index has a minimum value of zero and no fixed upper bound.

2.4.3 Davies–Bouldin Index

The Davies–Bouldin Index compares the internal spread of clusters with the distance between their centres [6]. For clusters i and j,

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}, \tag{3}$$

where S_i and S_j are the cluster spreads and M_{ij} is the distance between their centres. Lower values indicate more compact and well-separated clusters.

2.4.4 Adjusted Rand Index

The Adjusted Rand Index (ARI) measures the agreement between predicted and ground-truth cluster assignments while correcting for agreement expected by chance [7]. It is defined as

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}, \tag{4}$$

where RI is the Rand Index and $\mathbb{E}[\text{RI}]$ is its expected value under random cluster assignments. An ARI of 1 indicates perfect agreement, while values near 0 indicate agreement close to that expected by chance.

3 Methodology

The objective of this project is to evaluate the robustness of CLUEstering under irregular detector conditions and investigate whether dimensionality reduction using UMAP can improve its clustering performance. To achieve this, synthetic detector datasets were generated and modified with different controlled detector irregularities. CLUEstering was then evaluated both independently and in combination with UMAP using the same detector conditions and clustering validation metrics.

3.1 Synthetic Detector Dataset Generation

The detector data used in this work were synthetically generated using a data generation function provided in the CLUEstering repository [1]. The generated dataset represents detector energy deposits in a three-dimensional space and includes the spatial coordinates (x_0, x_1, x_2) together with an energy value for each point. The final dataset was generated with 200 clusters and a mean cluster size of 500 points. The spatial boundaries were defined as $(0, 300)$, $(0, 300)$, and $(0, 100)$ for the three dimensions, with an energy range of $(10, 20)$ and cluster standard deviations of $(10.0, 1.0, 1.0)$. A fixed random seed of 42 was used for reproducibility. The original data generation process was modified to additionally preserve the ground-truth cluster label associated with each generated point. This allowed the reconstructed CLUEstering labels to be compared directly with the original cluster assignments using the Adjusted Rand Index (ARI). The resulting synthetic detector dataset is shown in Figure 4.

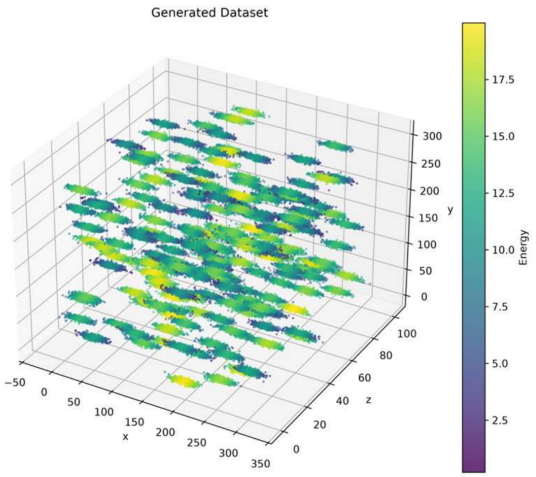


Figure 4 Generated synthetic detector dataset containing 200 clusters with energy visualization.

3.2 Detector Irregularity Modeling

To evaluate the robustness of the clustering methods under non-ideal detector conditions, four controlled detector irregularities were considered: a V-shaped crack, a longitudinal gap, added noise, and an inactive detector module. For each irregularity, a mathematical model was developed to define how the detector data should be modified. Each model was then implemented as a Python function and applied independently to the same synthetic detector dataset. This allowed the effect of each irregularity on the clustering performance to be evaluated under controlled conditions.

3.2.1 V-shaped Crack

A V-shaped crack was modeled by removing detector points located inside a V-shaped region:

$$D' = D \setminus \left\{(x_i, y_i, z_i, w_i) : y_i \leq y_v + m|z_i - z_c|\right\}, \tag{5}$$

where y_v is the crack vertex, z_c is the centre in the z direction, and m controls the shape of the crack. The model was implemented as a Python function and applied to the synthetic dataset.

3.2.2 Longitudinal Gap

A longitudinal gap was modeled by removing detector points within a straight region of width g:

$$D' = D \setminus \left\{(x_i, y_i, z_i, w_i) : |z_i - z_0| < \frac{g}{2}\right\}, \tag{6}$$

where z_0 is the centre of the gap and g is its width. The model was implemented as a Python function that removes points within the defined gap.

3.2.3 Added Noise

Noise was modeled by adding randomly generated points to the detector dataset:

$$D' = D \cup N_{\text{noise}}, \qquad N_{\text{noise}} = \{n_j\}_{j=1}^{K}, \tag{7}$$

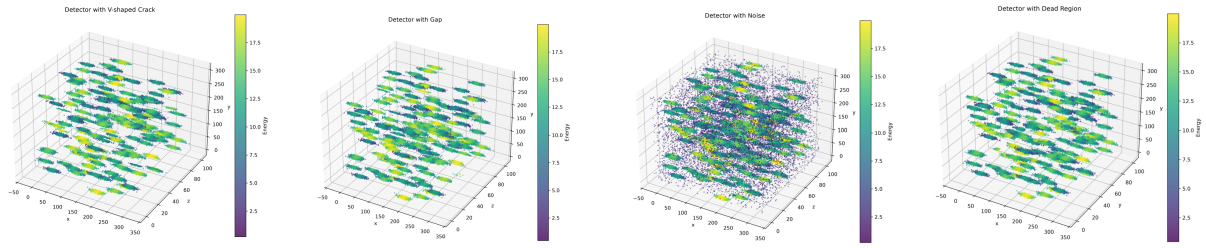
where K is the number of added noise points and n_j represents a randomly generated noise point. These points do not belong to any ground-truth cluster. The model was implemented as a Python function that adds uniformly distributed noise to the dataset.

3.2.4 Inactive Detector Module

An inactive detector module was modeled by removing all points within a three-dimensional cuboid:

$$D' = D \setminus \left\{(x_i, y_i, z_i, w_i) : |x_i - x_c| < \frac{L_x}{2},\ |y_i - y_c| < \frac{L_y}{2},\ |z_i - z_c| < \frac{L_z}{2}\right\}, \tag{8}$$

where (x_c, y_c, z_c) defines the centre of the inactive region and L_x, L_y, and L_z define its dimensions. The model was implemented as a Python function that removes all points within this region. Figure 5 shows the resulting synthetic detector datasets after applying each irregularity. The V-shaped crack and longitudinal gap introduce structured missing regions, while the added noise introduces random points throughout the detector space. The inactive detector module creates a localized three-dimensional region with missing measurements.



(a) V-shaped crack **(b)** Longitudinal gap **(c)** Added noise **(d)** Inactive module

Figure 5 Extended detector datasets under the tested irregular conditions.

3.3 Experimental Pipeline

Two clustering pipelines were evaluated under the same detector conditions. The first applied CLUEstering directly to the original detector data, while the second applied UMAP before CLUEstering. The two approaches were evaluated on the ideal dataset and on each of the irregular detector datasets.

3.3.1 CLUE-only Pipeline

In the first pipeline, CLUEstering was applied directly to the original three-dimensional detector data. The final CLUE configuration used a density radius of $d_c = 8.3$ and a minimum density threshold of $\rho_c = 5.0$.

3.3.2 UMAP + CLUEstering Pipeline

In the second pipeline, UMAP was first applied to transform the detector data before clustering. The UMAP configuration used `n_neighbors`=50, `min_dist`=0.1, and `n_components`=2. CLUEstering was then applied to the resulting representation using $d_c = 0.6$ and a minimum density threshold of $\rho_c = 5.0$. The final configurations of both clustering pipelines are summarized in Table 1.

Table 1 Final configurations used for the two clustering pipelines.

Parameter	CLUE-only	UMAP + CLUE
UMAP `n_neighbors`	–	50
UMAP `min_dist`	–	0.1
UMAP `n_components`	–	2
CLUE d_c	8.3	0.6
CLUE ρ_c	5.0	5.0

3.4 Evaluation Procedure

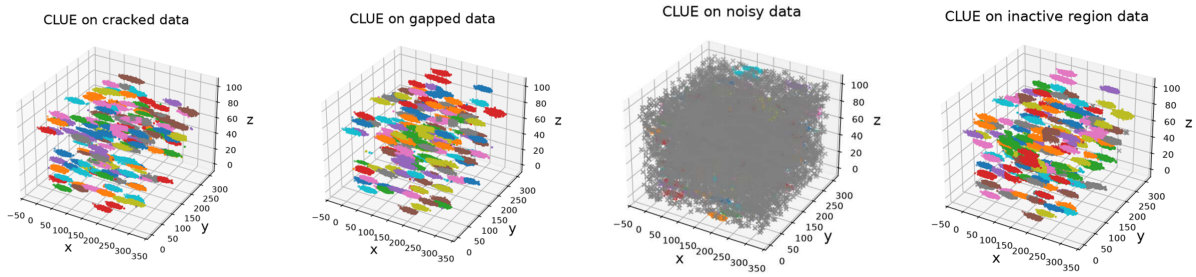
The clustering results were evaluated using the Silhouette Score, Dunn Index, Davies–Bouldin Index, and Adjusted Rand Index (ARI), as described in Section 2.4, together with the number of reconstructed clusters. These metrics were calculated for each detector condition and for both the CLUE-only and UMAP + CLUE pipelines. For the internal clustering metrics, points classified by CLUEstering as outliers were excluded before calculating the Silhouette Score, Dunn Index, and Davies–Bouldin Index. The ARI was calculated by comparing the predictedcluster assignments with the ground-truth cluster labels preserved during synthetic data generation. For the added-noise dataset, artificially generated noise points were excluded from the ARI calculation so that the reconstruction of the original detector clusters could be evaluated independently. The reconstructed cluster count was also compared with the number of ground-truth clusters remaining after each detector irregularity was applied. This provided an additional measure of how closely each pipeline recovered the expected cluster structure.

4 Results

The performance of the two clustering pipelines was evaluated across the ideal dataset and the different irregular detector conditions. The results are presented first for the CLUE-only pipeline, followed by the UMAP + CLUE pipeline and a direct comparison between the two approaches.

4.1 CLUE-only Performance

Figure 6 shows the clustering produced by the CLUE-only pipeline for the V-shaped crack, longitudinal gap, added-noise, and inactive-module datasets.



(a) V-shaped crack **(b)** Longitudinal gap **(c)** Added noise **(d)** Inactive module

Figure 6 CLUE clustering results for the irregular detector datasets: (a) V-shaped crack, (b) longitudinal gap, (c) added noise, and (d) inactive module.

Table 2 summarizes the performance of the CLUE-only pipeline across the ideal and irregular detector conditions.

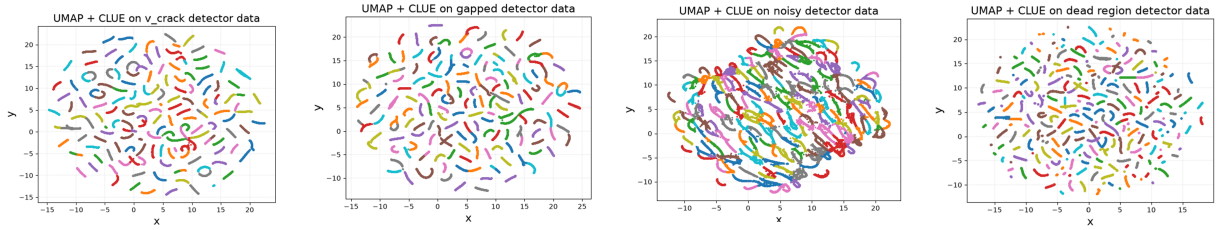
Table 2 CLUE-only clustering performance across the tested detector conditions.

Condition	Expected	Detected	Silhouette	Dunn	DB	ARI
Ideal	200	201	0.4480	0.00552	0.6816	0.8295
V-shaped crack	164	174	0.4797	0.00827	0.6257	0.8713
Longitudinal gap	177	184	0.4673	0.00565	0.6494	0.8593
Added noise	200	196	0.4485	0.00423	0.7183	0.8294
Inactive module	181	164	0.5062	0.00574	0.6564	0.8499

The CLUE-only pipeline produced cluster counts that remained reasonably close to the expected values across the tested conditions. The ideal dataset produced 201 clusters compared with 200 ground-truth clusters, while the V-shaped crack and longitudinal gap produced 174 of 164 and 184 of 177 expected clusters, respectively. The added-noise condition reconstructed 196 of the expected 200 clusters, and the inactive-module condition reconstructed 164 of 181. ARI values remained between 0.8294 and 0.8713 across all conditions, with the highest value obtained for the V-shaped crack. For the added-noise condition, CLUE rejected 8,146 of the 9,985 artificially added noise points, corresponding to a noise rejection rate of approximately 81.6%.

4.2 UMAP + CLUE Performance

Figure 7 shows the clustering produced by the UMAP + CLUE pipeline for the V-shaped crack, longitudinal gap, added-noise, and inactive-module datasets.



(a) V-shaped crack **(b)** Longitudinal gap **(c)** Added noise **(d)** Inactive module

Figure 7 UMAP + CLUE clustering results for detector datasets with irregularities: (a) V-shaped crack, (b) longitudinal gap, (c) added noise, and (d) inactive module.

Table 3 presents the corresponding results after applying UMAP before CLUEstering.

Table 3 UMAP + CLUE clustering performance across the tested detector conditions.

Condition	Expected	Detected	Silhouette	Dunn	DB	ARI
Ideal	200	199	0.6306	0.00289	0.5398	0.9715
V-shaped crack	164	160	0.6213	0.00206	0.5425	0.9654
Longitudinal gap	177	170	0.6199	0.00351	0.5531	0.9520
Added noise	200	198	0.4122	0.00125	0.8575	0.8772
Inactive module	181	253	0.6109	0.00330	0.4681	0.8722

The UMAP + CLUE pipeline achieved strong agreement with the ground-truth clustering for most of the tested detector conditions. The ideal, V-shaped crack, and longitudinal gap datasets achieved ARI values of 0.9715, 0.9654, and 0.9520, respectively, with reconstructed cluster counts of 199 of 200, 160 of 164, and 170 of 177. The added-noise condition reconstructed 198 of the expected 200 clusters with an ARI of 0.8772. For the inactive-module condition, an ARI of 0.8722 and a Silhouette Score of 0.6109 were obtained; however, the number of detected clusters increased to 253 compared with 181 expected clusters, indicating increased cluster fragmentation under this detector irregularity. Overall, UMAP + CLUE maintained high clustering agreement, although its performance in terms of cluster reconstruction was less robust for the inactive-module condition.

5 Investigation of UMAP's Effect on CLUE

The previous results showed that adding UMAP before CLUE improved the clustering performance across the tested detector conditions. This improvement was also observed under the inactive-module condition, where a complete region of the detector data was removed. However, using UMAP before CLUE would add a computationally expensive step to a production pipeline. This raised the question of whether the full UMAP transformation is necessary, or whether the part that helps CLUE could be understood and reproduced using a simpler preprocessing method. The investigation therefore focused on understanding what UMAP changes in the data and which of these changes could explain the improvement in CLUE. The ideal dataset with 200 ground-truth clusters was used as a controlled baseline, where each point in the original data could be directly compared with its corresponding point after UMAP.

5.1 Local Neighbourhood Scaling

The first step was to investigate how UMAP changes the local neighbourhood around each point. The distance to the 50th nearest neighbour was measured before and after UMAP and compared across the 200 clusters. The results showed that UMAP strongly changes the local neighbourhood distance scale, producing much smaller neighbourhood distances in the transformed space. This is important for CLUE because it uses a fixed neighbourhood radius, meaning that changes in the local distance scale can directly affect how CLUE identifies neighbouring points and estimates local density. The change in local neighbourhood scale before and after UMAP is shown in Figure 8.

```
ORIGINAL
Mean: 1.8800952250934824
Std: 1.0524556641512786
CV: 0.5597884884256052

UMAP
Mean: 0.030960087296860773
Std: 0.019703712497259704
CV: 0.6364230277625083
```

Figure 8 Local neighbourhood scaling before and after UMAP using the 50th-nearest-neighbour distance.

5.2 Cluster Separation and Neighbourhood Purity

The next step was to check whether UMAP simply moves clusters farther apart. Instead of only measuring the distance between clusters, the separation was measured relative to the local neighbourhood size. The local neighbourhoods were already highly pure in the original data, meaning that most neighbouring points belonged to the same true cluster. After UMAP, the neighbourhood purity remained high, but the relative separation increased considerably. This suggests that UMAP is not mainly fixing strongly mixed clusters. Instead, it increases the separation between neighborhoods that were already mostly correct and the surrounding clusters. Relative separation and neighborhood purity before and after UMAP are shown in Figure 9.

```
================ ORIGINAL ================
Mean relative separation: 4.128096312776707
Median relative separation: 3.5489429680572657
Mean neighborhood purity: 0.9830485899695562

================ UMAP ====================
Mean relative separation: 11.878797200173896
Median relative separation: 11.593212274597782
Mean neighborhood purity: 0.9891663996154464
```

Figure 9 Relative cluster separation and neighborhoods purity before and after UMAP.

5.3 Connection to CLUE Improvement

The previous results showed that UMAP changes both the local neighbourhood scale and the relative separation between clusters. The next step was to determine which of these changes was actually related to the improvement observed in CLUE. The reconstruction performance was therefore evaluated for each ground-truth cluster using the F1 score. The mean reconstruction

score increased from 0.923 with CLUE-only to 0.983 with UMAP + CLUE, showing that the individual clusters were reconstructed more accurately after UMAP. The reconstruction scores and correlation analysis are shown in Figure 10.

```
Mean CLUE reconstruction score:
CLUE only: 0.9229818489380053
UMAP + CLUE: 0.9828552638247768

Spearman correlation with CLUE improvement:
Local-scale CV change: 0.32431138457839437 p = 2.8001850206513613e-06
Relative-separation change: 0.03645186985822766 p = 0.6083386099126017
Neighborhood-purity change: 0.602073539897059 p = 4.1087681423896446e-21
```

Figure 10 Mean CLUE reconstruction score and correlation between the changes introduced by UMAP and the improvement in cluster reconstruction.

The change in reconstruction performance was then compared with the changes in local-scale variation, relative separation, and neighbourhood purity. Neighbourhood-purity change showed the strongest correlation with CLUE improvement ($\rho = 0.602$), followed by local-scale variation ($\rho = 0.324$). Both correlations were statistically significant. In contrast, relative-separation change showed almost no correlation with CLUE improvement ($\rho = 0.036$, $p = 0.608$). This was interesting because UMAP considerably increased the overall separation between clusters, as shown in the previous section, but this increase did not explain which clusters improved. Instead, the results suggest that the improvement is more closely related to changes in the local neighbourhood structure.

5.4 Preliminary UMAP-Inspired Preprocessing

Based on the previous findings, the next step was to test whether some of the useful effects of UMAP could be reproduced using simpler preprocessing before CLUE. Three candidates were explored on the ideal dataset.

Candidate 1: Local adaptive scaling. The first candidate was motivated by the change in local neighbourhood scale observed after UMAP. For each point, the 50 nearest neighbours were found and the distance to the 50th neighbour was used as its local scale. The point was then rescaled relative to its neighbourhood centre using the median local scale of the dataset as a common reference. The aim was to make the local distance scales more consistent before applying CLUE.

Candidate 2: Locally weighted neighbourhood smoothing. The second candidate focused on the local neighbourhood structure. Nearby points were given larger weights based on their distance, and a weighted neighbourhood centre was calculated for each point. The point was then moved partially towards this centre using a smoothing factor of $\alpha = 0.2$. This was intended to strengthen the local neighbourhood structure while keeping the original geometry largely unchanged.

Candidate 3: Combined preprocessing. Finally, the two approaches were combined. Local adaptive scaling was applied first, after which the neighbourhoods were recalculated using the transformed coordinates. The locally weighted smoothing was then applied to the scaled data. A simplified version of the combined preprocessing is shown in Listing 1.

Listing 1 Simplified implementation of the combined preprocessing.

```
# Candidate 1: local adaptive scaling
local_scale = neighbor_distances[:, K - 1]
local_centers = np.mean(X[neighbor_indices], axis=1)

X_scaled = local_centers + (
    (X - local_centers) / local_scale[:, None]
) * np.median(local_scale)

# Candidate 2: weighted neighbourhood smoothing
weights = np.exp(-adjusted_distances / sigma[:, None])
weighted_center = weighted_neighbourhood_center(weights)

X_combined = (1 - ALPHA) * X_scaled + ALPHA * weighted_center
```

Candidate 3 gave the best preliminary result. On the ideal dataset, it reconstructed 197 of the 200 expected clusters and achieved an ARI of 0.916. For comparison, CLUE-only achieved an ARI of 0.830, while UMAP + CLUE achieved 0.972. Although the combined preprocessing did not fully reproduce the UMAP result, it recovered a considerable part of the improvement using a much simpler transformation. These results are promising, but further investigation is still needed. The current preprocessing was tested only on the ideal dataset, and it is not yet clear whether the same approach will work under detector irregularities. The next step is therefore to test and refine the method on the V-shaped crack, longitudinal gap, added-noise, and inactive-module conditions, and to determine whether the useful effect of UMAP can be reproduced consistently without including UMAP in the final pipeline.

6 Limitations and Future Work

The main limitation of this work is that the UMAP-inspired preprocessing was only tested on the ideal dataset. The first results are promising, with the ARI improving from 0.830 with CLUE-only to 0.916 with the combined preprocessing, but more testing is needed to understand whether the same improvement can be achieved under detector irregularities. The next step is to test the combined approach on the V-shaped crack, longitudinal gap, added-noise, and inactive-module datasets. This will help determine whether the effects identified in the ideal data also explain UMAP's improvement under more difficult conditions. Further work should also compare the clustering performance and computational cost of CLUE-only, UMAP + CLUE, and the simplified preprocessing + CLUE. The final goal is to determine whether the useful effect of UMAP can be reproduced with a simpler and faster preprocessing step that is more suitable for a production pipeline.

7 Conclusion

In this project, I investigated how CLUE clustering performs when detector data contains different types of irregularities in the context of the CMS experiment at the LHC. I tested V-shaped cracks, longitudinal gaps, added noise, and inactive detector regions, and compared CLUE-only with UMAP + CLUE. I found that CLUE remained reasonably robust even under the more difficult conditions, while adding UMAP improved the clustering agreement across all the tested datasets. After observing this improvement, I investigated why UMAP helps CLUE. Since running UMAP as part of a production pipeline would be computationally expensive, the goal was to understand which parts of its transformation were actually useful. The results showed that changes in the local neighbourhood structure were more closely related to the improvement in CLUE than simply increasing the separation between clusters. Based on these findings, I tested local adaptive scaling, locally weighted neighbourhood smoothing, and finally a combination of the two as simpler alternatives. The combined approach improved the ARI on the ideal dataset from 0.830 to 0.916. Although further investigation is still needed, this result is promising and shows that part of UMAP's benefit may be reproduced using a simpler preprocessing step before CLUE. Overall, this project allowed me to test CLUE under challenging detector conditions, investigate why UMAP improves its performance, and take the first steps towards reproducing this improvement in a way that could be more practical for future use in the CMS clustering pipeline.

References

[1] Simone Balducci et al. *CLUEstering Documentation*. 2026. URL: https://cms-patatrack.github.io/CLUEstering/ (visited on 07/2026).

[2] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[3] CERN. *CMS Experiment*. 2026. URL: https://home.cern/science/experiments/cms/ (visited on 07/2026).

[4] Peter J. Rousseeuw. "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis". In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65.

[5] J. C. Dunn. "Well-Separated Clusters and Optimal Fuzzy Partitions". In: *Journal of Cybernetics* 4.1 (1974), pp. 95–104.

[6] D. L. Davies and D. W. Bouldin. "A Cluster Separation Measure". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1.2 (Apr. 1979), pp. 224–227.

[7] Lawrence Hubert and Phipps Arabie. "Comparing Partitions". In: *Journal of Classification* 2.1 (1985), pp. 193–218. DOI: 10.1007/BF01908075.