KIT-Fakultät für Physik
Institut für Experimentelle Teilchenphysik

Vorlesung: Prof. Dr. M. Klute, Dr. P. Goldenzweig
Übung: Dr. G. De Pietro
Assistenz: J. Eppelt, S. Giappichini, L. Reuter, J. Saffer, D. Wong

Übungen zu Teilchenphysik I

Wintersemester 2024/25

Exercise 3 To be worked on until November 21, 2024

To start with the exercise, please login to the jupytermachine, start the standard `Datenanalyse` container and update your `tp1_forstudents` repository, e.g. by navigating to the directory in the file browser on the left and then selecting `Git -> Pull from Remote` from the menu bar. A new subfolder `Exercise03` should appear with a Jupyter notebook `Exercise03.ipynb`. Open this notebook and work though the different tasks that are given inside.

But... before you navigate through the notebook, please read the following sections, which will give you some fundamental information to help you complete the exercise successfully.

1 Electromagnetic cascades in a calorimeter

1.1 Energy loss of electrons/positrons and photons

Bremsstrahlung is the primary energy loss process for electrons/positrons above ~ 10 MeV, while photons lose energy mainly through the production of electron–positron pairs. High-energy photons, electrons, and positrons create a cascade of secondary particles—known as an "electromagnetic shower." The following quantities (*Thomson's approximation*) are defined:

- **Radiation Length** X_0: Average distance over which the electron energy reduces by $1/e$:

$$X_0 \approx \frac{1}{4\,\alpha\,n\,Z^2\,r_e^2 \ln\left(\frac{287}{\sqrt{Z}}\right)}$$

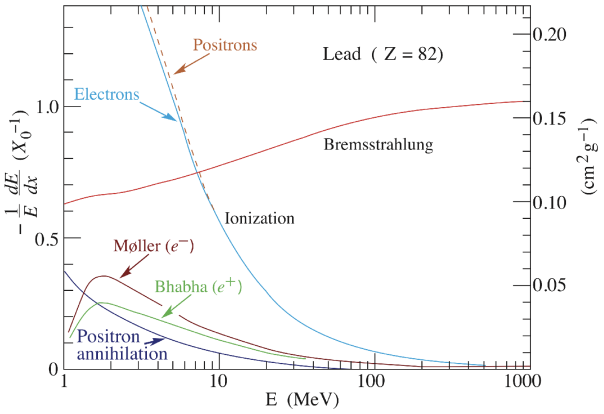

Figure 1: Energy loss of electron as a function of energy.

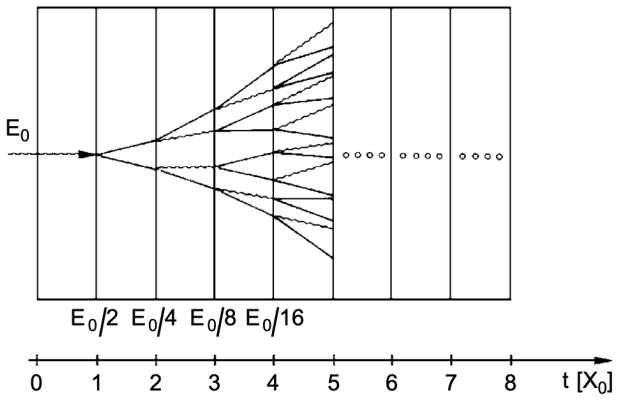

Figure 2: Electromagnetic cascade

where α is the fine-structure constant, n is the number density of the nucleus, Z is the atomic number of the nucles, and r_e is the classical electron radius.

- **Critical Energy** E_c: Energy at which ionization becomes the dominant energy loss:

$$E_c \approx \frac{800\text{MeV}}{Z}$$

- **Molière Radius** R_M: Represents the lateral spread of an electromagnetic shower, mainly due to multiple scattering:

$$R_M = \frac{21\,\text{MeV}}{E_c} X_0\,(\text{g/cm}^2)$$

  Approximately 95% of the shower energy is contained within $2R_M$ (transverse width).

- **Maximum Particle Count Length** $x_{\max}$: The shower reaches the maximum particle count after $x_{\max}$ radiation lengths, given by:

$$x_{\max} = \frac{\ln(E/E_c)}{\ln 2} X_0$$

  where E is the initial energy. $x_{\max}$ is also called longitudinal depth.

1.2 Energy loss of muons

For muons with energies less than 100 GeV, the ionization is the dominant energy-loss process. Muons travel significant distances in dense materials: for example, a 10 GeV muon loses order of 10 MeV/cm in iron and has a range of several meters.

A particle with a momentum close to the minimum ionization point is called a minimum ionizing particle (MIP). A particle with a much larger momentum but with an energy loss comparable to that of the minimum ionization point is also called a MIP.

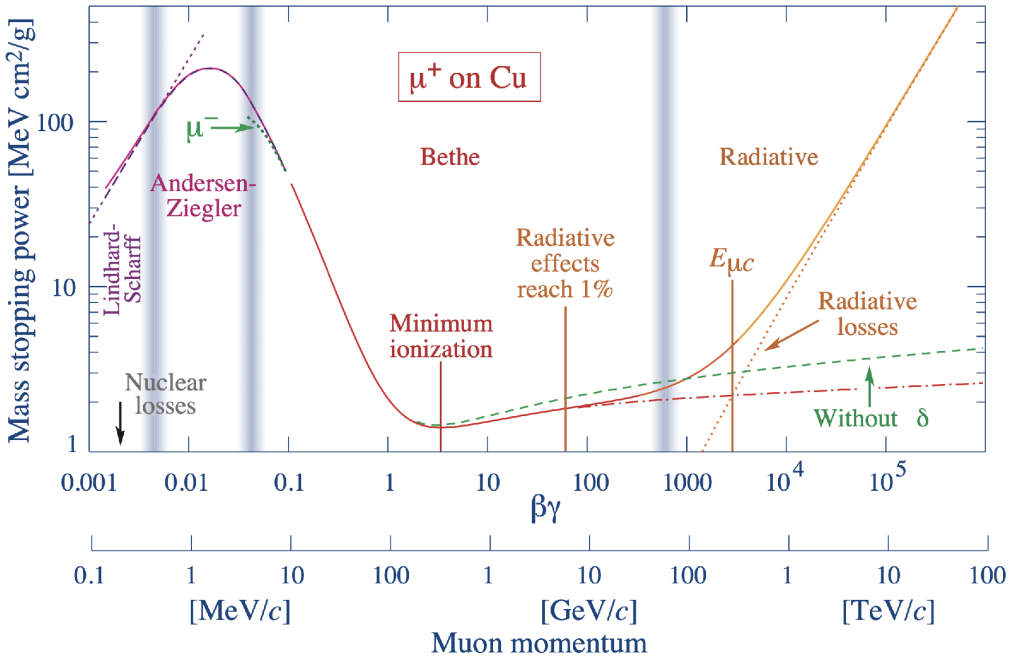

Figure 3: Energy loss of muons as a function of the muon momentum.

2 Calorimetry and reconstruction

We have prepared ROOT n-tuples containing hit information from EMCal simulations at the PHENIX experiment at Brookhaven National Laboratory (https://www.phenix.bnl.gov/). The data were generated using a particle gun with a gaussian energy distribution, ranging from 5 GeV to 80 GeV. The EMCal at the PHENIX experiment covers an area of 2 m $\times$ 4 m perpendicular to the beamline. This corresponds to 3 $\times$ 6 EMCal super modules where:

- 1 EMCal super module = 6 $\times$ 6 = 36 EMCal modules.
- 1 EMCal module is 11 cm $\times$ 11 cm and contains 2 $\times$ 2 = 4 towers/channels, with a granularity of approximately 5.5 cm.

In total, there are 2592 towers or channels to read out the energy deposits from the EMCal.

For practical purposes, consider the PHENIX EMCal as a 2D matrix of 2592 channels (72 horizontal x 36 vertical), where each channel has a dimension of (5.535cm x 5.535cm x 36.96cm).

In the notebook, you have the option to get readout for electron and dielectron events from the EMCal. The readout has the following format:

```
'elmID': array([1207, 1208, 1243, 1244, 1245, 1246, 1279, 1280, 1281, 1282,
       1283, 1314, 1315, 1316, 1317, 1318, 1352, 1387, 1388, 1389, 1390],
       dtype=int32),
'edep': array([2.1672759e-03, 2.3865167e-03, 6.9637364e-03, 6.5519638e-02,
       5.9960117e-03, 1.5223619e-03, 1.5519783e-02, 1.3763729e+00,
       1.7959915e-02, 1.1316873e-03, 2.6412117e-03, 1.1183993e-03,
       3.0373151e-03, 1.8506199e-02, 5.9680296e-03, 1.2358509e-03,
       1.5430434e-03, 1.2675102e-03, 8.8297541e-04, 8.7783835e-04,
       1.5175857e-03],
       dtype=float32)
```

where `elmID[X]` is the identifier of the channel that measured a given `edep[X]`. The values of `elmID[X]` vary from 0 to 2591 (corresponding to the total number of channels).

2.1 Moments of a distribution

The moments of a distribution provide useful information about its shape and spread. Below are the formulas for different moments:

- **1st order (raw)**:

$$\text{Mean: } \mu := \mathbb{E}[X] = \frac{\mu_1}{1}$$

- **2nd order (central)**:

$$\text{Variance: } \sigma^2 = \mathbb{E}[(X-\mu)^2] = \frac{\mu_2}{1^2} = \mu_2$$

- **3rd order (standardized)**:

$$\text{Skewness}\gamma = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\mathbb{E}[(X-\mu)^3]}{(\mathbb{E}[(X-\mu)^2])^{3/2}}$$

- **4th order (standardized)**:

$$\text{Kurtosis } g = \mathbb{E}\left[\left(\frac{X-\mu}{\sigma}\right)^4\right] = \frac{\mathbb{E}[(X-\mu)^4]}{(\mathbb{E}[(X-\mu)^2])^2}$$

They are helpful to analyze the 2D distribution of the EMCal hits.

3 Calorimetry and clustering algorithms

Due to the complex patterns of particle interactions, EMCal readouts can become dense and chaotic. In these situations, clustering algorithms are essential to group hits that originate from a single particle, helping to reconstruct particle flows or analyze jets.

3.1 Clustering using the K-means algorithm

The K-means algorithm is a popular unsupervised machine learning technique used to partition data into K distinct clusters based on feature similarity.

1. **Initialize centroids**: Randomly select K hits; these hits will be the starting, K, centroids
2. **Assign clusters**: Assign hits to the nearest centroid, forming K clusters
3. **Update centroids and reassign hits**: Calculate the centroid (mean of the coordinates) of each cluster; check if a hit in a cluster is closer to another centroid; if so, reallocate the hits to the closest cluster.
4. **Iterate**: Repeat step 3 until no more hits change cluster (or until a certain tolerance).
5. **Output**: The final centroids represent the centers of the K clusters, and each hit is assigned to the cluster of its nearest centroid.

3.2 Finding the optimal number of clusters

K-means is an efficient method that works well with large datasets, making it ideal for quickly organizing data into clusters based on similarity. However, one limitation is that the number of clusters, K, must be predetermined, which can be challenging without prior knowledge of the data structure. To address this, techniques such as the Elbow and the Silhouette methods are often used to find the optimal number of clusters, helping to ensure that the chosen K best captures the natural groupings within the data. In any case, for a given event, one must run the K-means algorithm multiple times, scanning different values of K and identifying the optimal value of K.

3.2.1 Elbow method

The Elbow Method helps find the optimal number of clusters K by computing the within-cluster sum of squares (WCSS_K):

$$\text{WCSS}_K = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

where K is the number of clusters, C_i is the i-th cluster, x is a hit in the cluster, μ_i is the centroid of the cluster C_i, and $\|x-\mu_i\|^2$ is the squared Euclidean distance between the hit and the centroid. If you plot WCSS_K as a function of K, you will notice that as K increases, WCSS_K decreases. The "elbow" point, where the decrease slows down "sharply", indicates the optimal K.

To determine the elbow point more precisely, you can: calculate the second derivative of WCSS with respect to K and identify the point with the largest change, indicating the most significant slowdown in the rate of decrease; use algorithms such as the "Kneedle" algorithm, which automatically detects the "knee" or "elbow" in the curve by analyzing the curvature.

3.2.2 Silouette method

The Silhouette method evaluates the quality of clustering by calculating a "silhouette score" for each hit, based on how similar it is to hits in its own cluster compared to hits in the nearest other cluster. For a given number of clusters K, the average silhouette score S ranges from -1 to 1: S near 1 indicates well-clustered hits, close to their own cluster and far from others; S near 0 indicates the presence of hits on the boundary between clusters (overlapping clusters); a negative S indicates that many hits are assigned to the wrong cluster.

To determine the optimal K, calculate the average silhouette score for different K values and choose the K that maximizes this score, indicating the best-defined clusters. Here is a method to calculate the silhouette score S:

- Compute the intra-cluster distance a_i: for each hit i in a cluster, compute the average Euclidean distance to all other points within the same cluster.
- Compute the nearest cluster distance b_i: for each hit i in a cluster, compute the average euclidean distance to all points of another cluster; repeat for all clusters and take the smallest average distance computed.
- Compute silhouette hit score s_i: for each hit i in a cluster, compute

  $$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

  and repeat it for all hits in the event.
- Calculate the average silhouette score S of the event, i.e. the average of s_i over all hits in the event.

3.2.3 Energy seeding

We are future data scientists, but also future physicists! Let's approach the determination of K from the perspective of physics. The EMCal readouts provide not only the hit position information, but also the energy deposition measured for each hit.

"Energy seeds" in calorimetry refer to localized points in the detector where the energy deposition is significantly higher than the surrounding areas. These seeds act as initial markers for clustering algorithms, helping to identify and group hits that are likely to belong to the same cluster. By starting with these high-energy regions, energy seed-based clustering improves accuracy in reconstructing particle trajectories and separating overlapping showers.

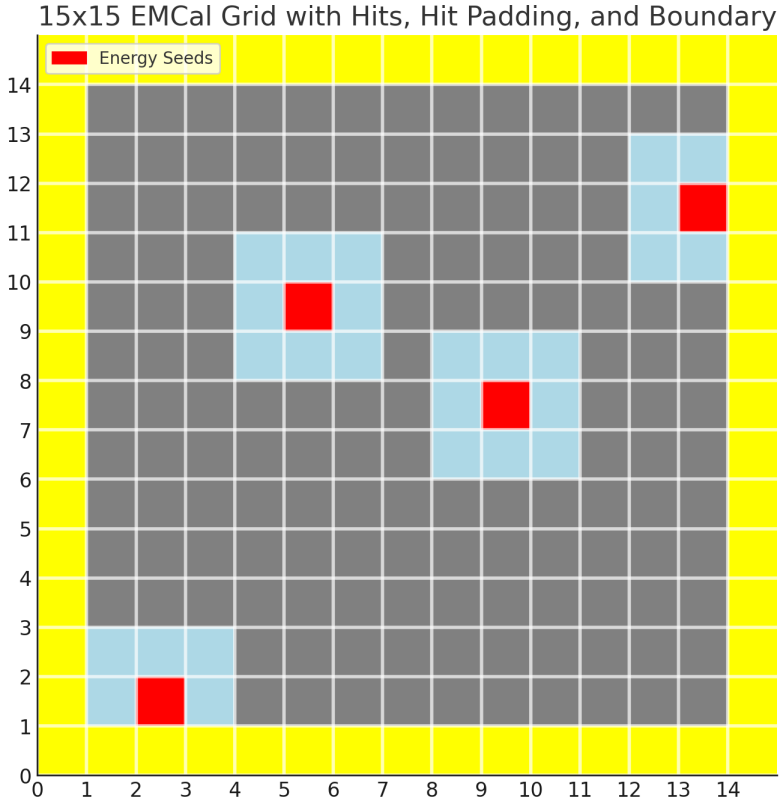

Figure 4: A possible logic for seed searching

Here there is an idea for a seed-searching algorithm:

- Suppose we have a 15×15 EMCal grid and create a $(15+1) \times (15+1)$ 2D array initialized with zeros.
- Assign energy values to the cells based on their `elmID`.
- Apply a minimum energy mask to identify potential energy seed candidates.

- Sort the energy seeds in ascending order.
- Add a padding region for each seed candidate. If any hit within the padding region has a higher energy than the candidate, discard the candidate. Loop through the list of energy seed candidates.

The found seed candidates can be used as a starting point for the K-means clustering algorithm.