Investigation of a Data-Driven Fast Simulation Algorithm for AHCAL Test Beam Data with the Discrete Cosine Transform

André Wilhahn

II. Physikalisches Institut, Georg-August-Universität Göttingen
Supervised by Stan Lai

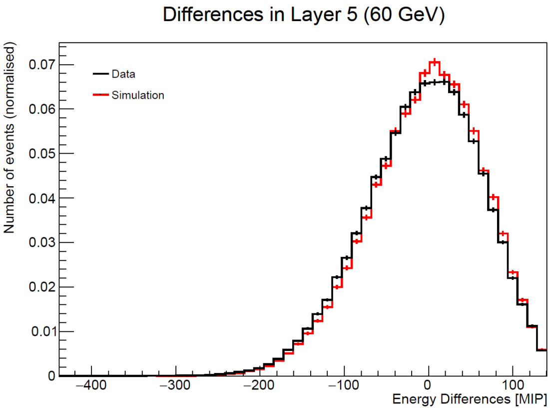
December 11th, 2024

Motivation

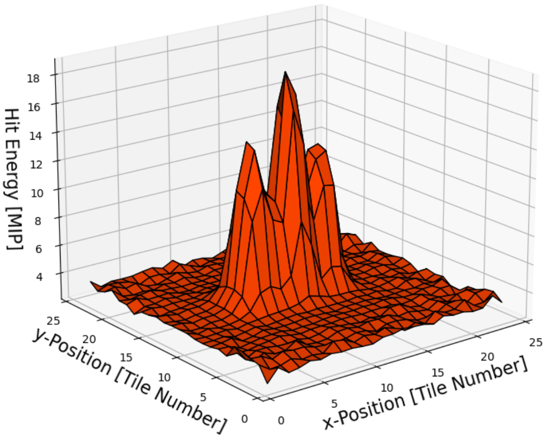
- Want to develop data-based fast simulation algorithm for hadron showers
- Decreased computation time as well as storage requirements
- Possible to simulate as accurately as full simulation (or even better)?
- Fast simulation based on pion shower test beam dataset from June 2018
  $\Rightarrow$ "Confined" to test beam conditions
- Combine with full simulation for more general test beam conditions
- Different approaches:
  1. Fast simulation with discrete cosine transform by myself
  2. Distance-based sorting algorithm by Zobeyer Ghafoor (next talk)

Longitudinal Simulation with Kernel Density Estimators

- Used Kernel Density Estimators for longitudinal simulation of pion showers
- Data and simulation in very good agreement
- Also developed inter- and extrapolation algorithms for longitudinal PDFs
- Want sooner or later also simulate:
  1. on hit level (also done by Zobi)
  2. timing
  3. under different incident angles
  4. different particles (briefly touched upon for Bachelor's thesis last summer semester)
  5. ...



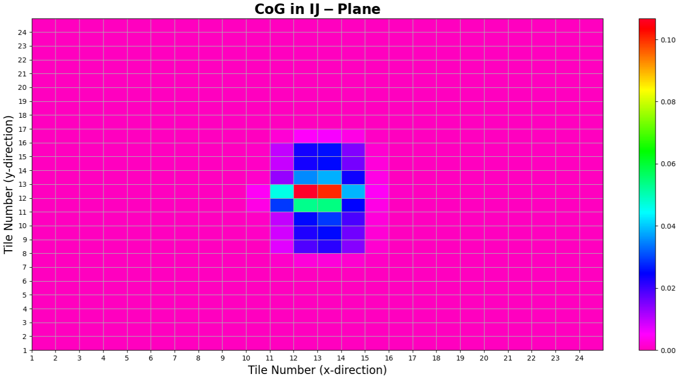
Compressing Test Beam Data?



- So far: **longitudinal** simulation with **Kernel Density Estimators**
- AHCAL has $24 \times 24 \times 38 = 21\,888$ readout channels
- If describing positions relative to centre of gravity and shower start, then technically $47 \times 47 \times 38 = 83\,942$ channels in analysis
  $\Rightarrow$ Too much for KDEs
- Goal thus: reduce ("compress") number of values to more manageable size for hit-wise simulation

Centre-of-Gravity Cuts

- Centre of gravity on average close to detector center
- Only consider events with CoG within 8×8 block around detector center
  $\Rightarrow$ Reduces input values to $31 \times 31 \times 38 = 36\,518$ (still too much for KDEs)



Discrete Cosine Transform

- Consider 1D-array of N real numbers: $\{x_0,\ x_1,\ ...,\ x_{N-1}\}$
- Discrete Cosine Transform (DCT) will transform these N real numbers into another set of N real numbers: $\{X_0,\ X_1,\ ...,\ X_{N-1}\}$, where

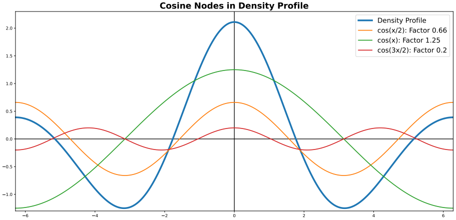
$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]$$

- DCT has exact inverse defined as

$$x_n = \frac{2}{N}\left(\frac{1}{2}X_0 + \sum_{k=1}^{N-1} X_k \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right]\right)$$

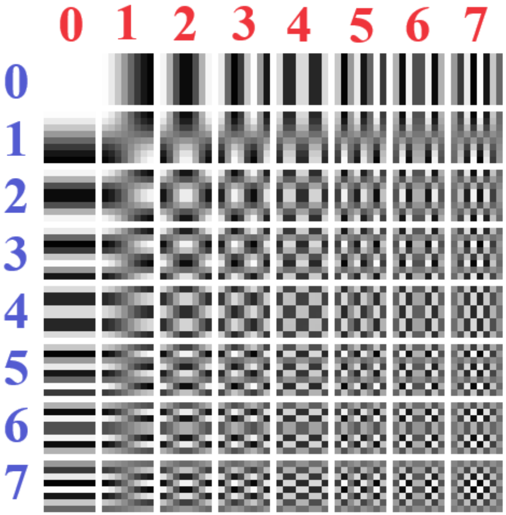
Discrete Cosine Transformation

- Split three-dimensional hit distributions into single cosine waves
  $\Rightarrow$ In principle real Fourier transform
- Transformed values quantify how strongly specific cosine nodes are represented in original PDF
- AHCAL has three dimensions
  $\Rightarrow$ Use three-dimensional DCT
- If blue curve was hit distribution, expect cosine nodes to fall off towards edge of layer



Two-Dimensional Discrete Cosine Transform

- From **left to right**: lowest to highest **x-node**
- From **top to bottom**: lowest to highest **y-node**
- Superpositions of x- and y-nodes
- High (white) and low (black) intensities
- Expect combinations with odd-numbered nodes to vanish
  $\Rightarrow$ Correspond to energy increases at edges of active layers



Complexity of Discrete Cosine Transform

- One-dimensional DCT requires $\mathcal{O}(n^2)$ operations: n outputs, each of which requires a sum over n terms
- Becomes even worse for AHCAL test beam data in three dimensions
  $\Rightarrow$ Runtime $\sim \mathcal{O}(n^6)$
- Fast Fourier Transform (FFT) comes into play which has runtime of $\mathcal{O}(n \log n)$
  $\Rightarrow$ Technically Fast Cosine Transform (FCT), but mathematics between FFT and FCT are equivalent

Radix-2 Decimation-in-time FFT (Cooley-Tukey Algorithm)

- DFT defined as: $X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N}nk}$ with $k = 0, 1, ..., N-1$
- Split sum into two sums running over even/odd indices, respectively:

$$X_k = \underbrace{\sum_{m=0}^{\frac{N}{2}-1} x_{2m} e^{-\frac{2\pi i}{N/2}mk}}_{\text{even-indexed part } E_k} + e^{-\frac{2\pi i}{N}k} \underbrace{\sum_{m=0}^{\frac{N}{2}-1} x_{2m+1} e^{-\frac{2\pi i}{N/2}mk}}_{\text{odd-indexed part } O_k}$$

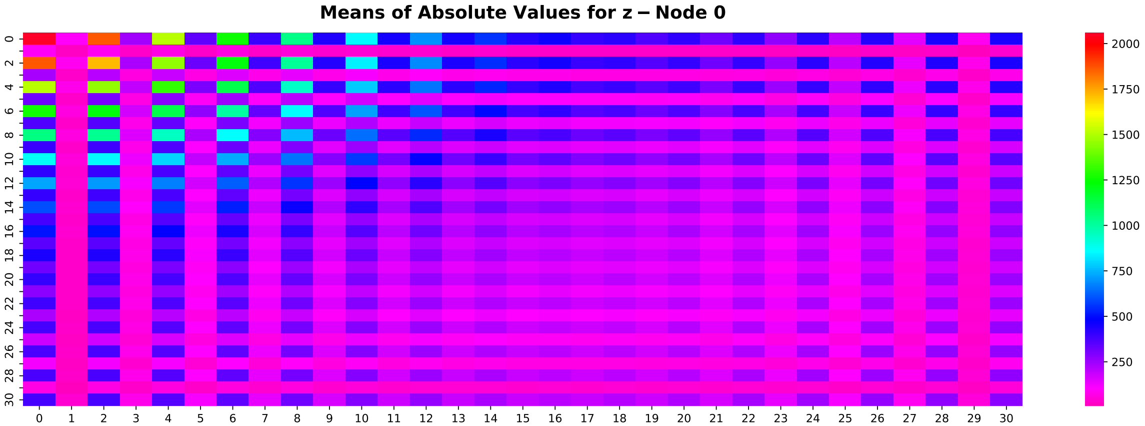
$$= E_k + e^{-\frac{2\pi i}{N}k} O_k$$

- DFT of N terms has been split into two DFTs of only $\frac{N}{2}$ terms
  $\Rightarrow$ Apply algorithm recursively
- Because of periodicity of exponential function, we have:

$$X_{k+\frac{N}{2}} = E_k - e^{\frac{2\pi i}{N}k} O_k$$

Mean Absolute Cosine Nodes

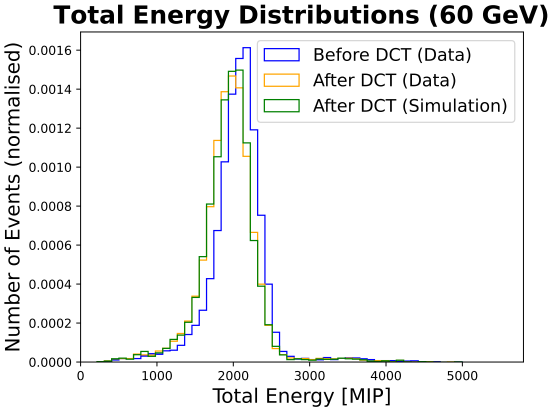
- On average, most nodes do not carry large coefficients
- "Even-even" nodes dominate, "odd-odd" nodes the smallest



Simulate Only Even-Even Nodes

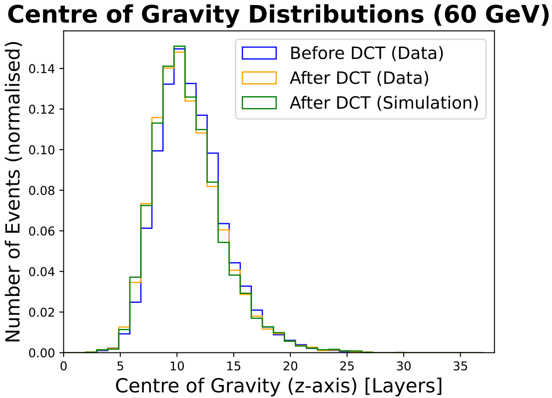
- Only simulate even-even nodes for $z \leq 24$ (everything else set to zero)
- Transform simulated nodes back into hit energies
- Create PDFs of shower variables to compare data before FCT with data after FCT (only even-even nodes) and simulated even-even nodes
- Expect simulation to match data with only even-even nodes, and (small) deviations from unaltered dataset
  $\Rightarrow$ Visible in PDFs

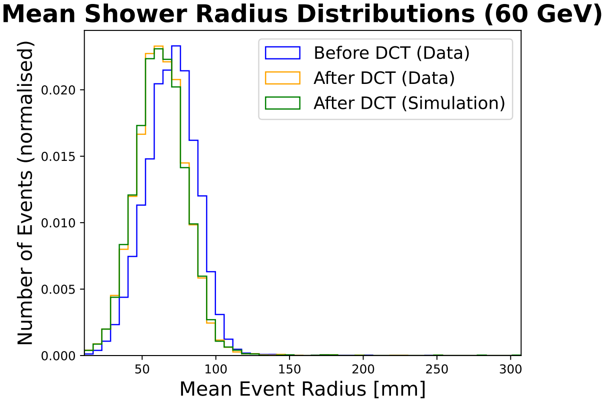
$$E_{\text{total}} = \sum_{\text{hits}} E_{\text{hit}}$$



Centre of Gravity and Mean Shower Radius

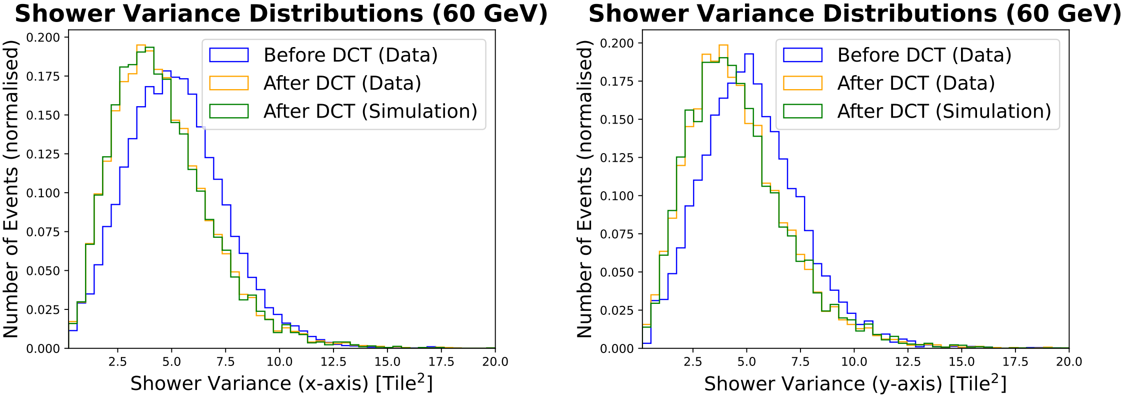
$$\mathrm{CoG}_z = \frac{1}{E_{\mathrm{total}}} \sum_{\mathrm{hits}} E_{\mathrm{hit}} \cdot z_{\mathrm{hit}} \qquad R = \frac{1}{E_{\mathrm{total}}} \sum_{\mathrm{hits}} E_{\mathrm{hit}} \cdot r_{\mathrm{hit}}$$





Shower Variances (x- and y-axis)

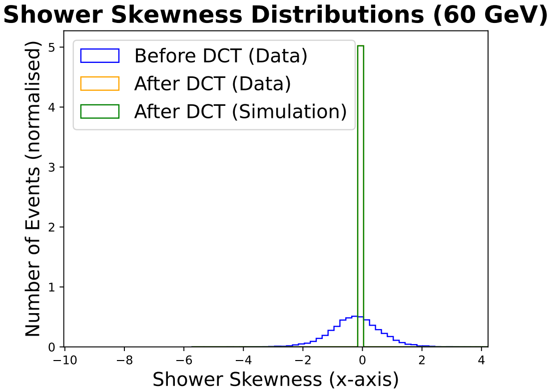
$$\mathrm{Var}(i) = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot (i_{\text{hit}} - \mathrm{CoG}_i)^2 \text{ for } i \in [x, y]$$

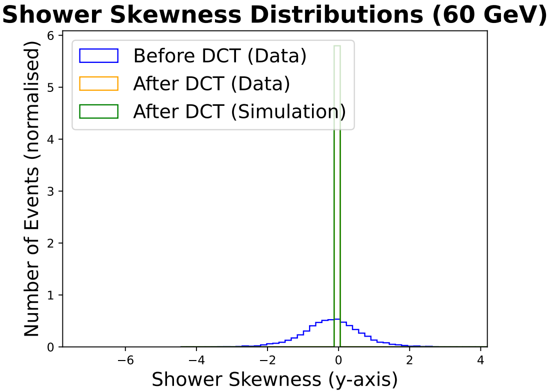


⇒ Showers are a bit too narrow

Shower Skewnesses (x- and y-axis)

$$\text{Skew}(i) = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot \left(\frac{i_{\text{hit}} - \text{CoG}_i}{\sigma_i} \right)^3 \quad \text{for } i \in [x, y] \text{ and } \sigma_i = \sqrt{\text{Var}(i)}$$

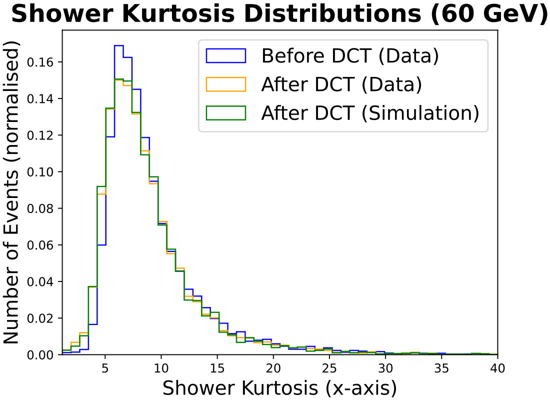


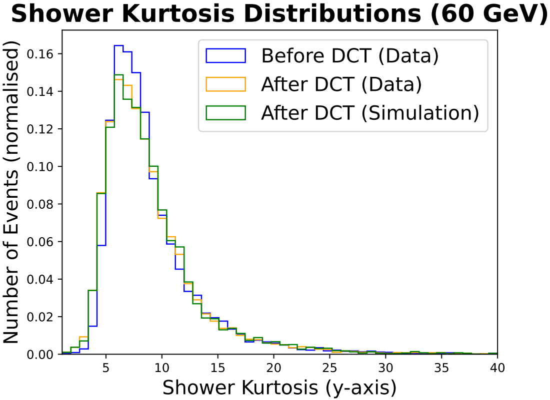


⇒ Showers are “too symmetric”

Shower Kurtoses (x- and y-axis)

$$\mathrm{Kurt}(i) = \frac{1}{E_{\mathrm{total}}} \sum_{\mathrm{hits}} E_{\mathrm{hit}} \cdot \left(\frac{i_{\mathrm{hit}} - \mathrm{CoG}_i}{\sigma_i} \right)^4 \quad \text{for } i \in [x, y] \text{ and } \sigma_i = \sqrt{\mathrm{Var}(i)}$$





Summary

Simulation of Cosine Nodes with KDEs:

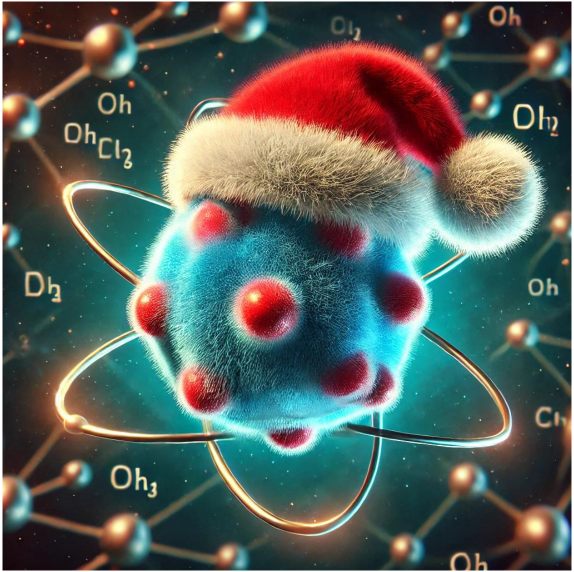
- Reduced number of input values by only simulating even-even nodes for $0 \leq z \leq 24$
- Down to $\frac{16\times16\times25}{31\times31\times38} \approx 17.5\,\%$ of original number of input values
- KDEs more “stable” because PDFs of even-even coefficients do not peak around zero

Distributions of Kinematic Variables:

- Kinematic variables in good agreement, but not perfect
- Biggest problems with skewnesses in x- and y-direction (simulated showers “too symmetric”)

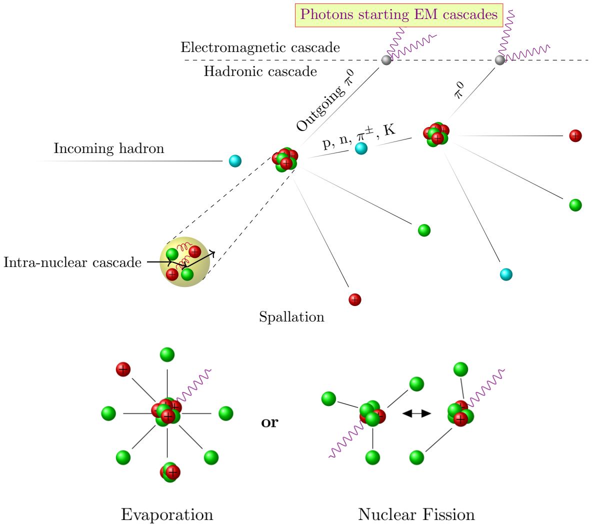
Future Steps

- Fix shower variances and in particular shower skewnesses
- So far tried to:
    1. include odd nodes into simulation again to make shower more asymmetric
    2. use fudge factors for hit radii
- Adding odd nodes back into simulation especially difficult
  $\Rightarrow$ Needle in the hay stack
- Help Zobi with his distance-based algorithm and compare results to those of FCT
- Eventually extend investigation to whole pion shower dataset (so far only 60 GeV for FCT)
- Long-term goals: timing, extra-/interpolation on hit level, EM showers, ...



Thanks for your attention!

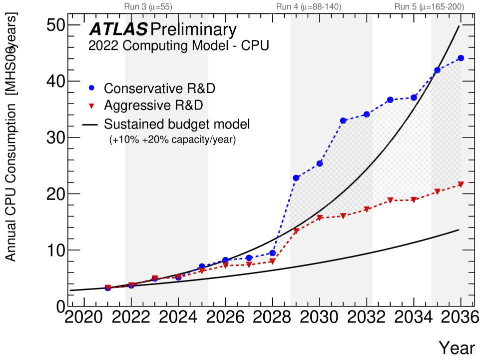
Hadronic Showers

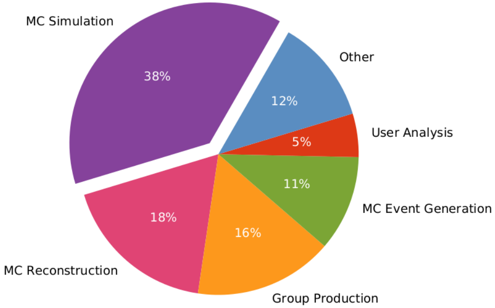


- Hadronic showers are very chaotic
- Energy resolution is limited in hadronic showers
  $\Rightarrow$ Limited by strongly varying electromagnetic fraction
- Study of single showers helps to understand behaviour of hadronic showers in highly granular calorimeters
- Can also develop fast simulation by studying single showers

Fast Simulations for Calorimetry?

- CPU consumption of MC simulations increases with occupancy/granularity
- Up to 90 % of calculation time is needed for the calorimeter (i.e. in ATLAS)
- Saving of computational resources will become necessary sooner or later
  ⇒ Data-driven fast simulation possible for highly granular calorimeters?





Kernel Density Estimators

- Want to find PDF of dataset x_1, x_2, ..., x_n
- Define Kernel Density Estimator (KDE) with bandwidth h as:

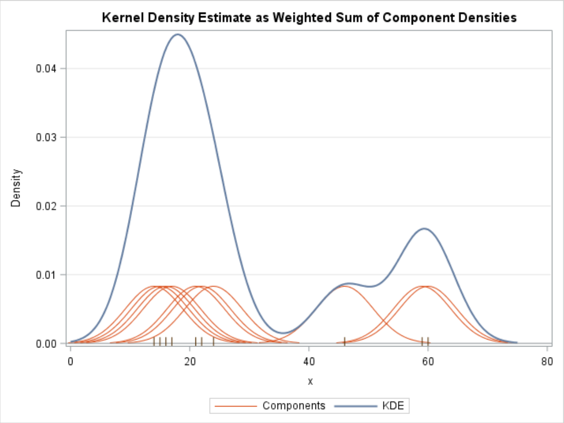
$$f(t) = \frac{1}{nh} \sum_{j=1}^{n} k\left(\frac{t - x_j}{h}\right)$$

with

$$k(t) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}t^2\right)$$

- PDF = sum of all (Gaussian) kernels
- Choice of bandwidth determines smoothness of PDF

Kernel Density Estimators



- Generalise to d dimensions:

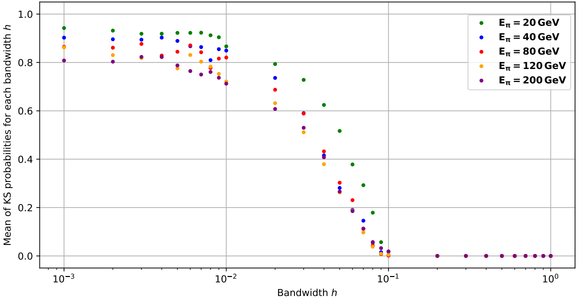
$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{H}|^{-1/2} K\left(\mathbf{H}^{-1/2}(\mathbf{x}-\mathbf{x}_i)\right)$$

$\Rightarrow$ **x**: d-dimensional data vector
$\Rightarrow$ **H**: $d \times d$ bandwidth matrix

- In Python, $\mathbf{H} = h^2\mathbf{C}$ where $\mathbf{C}$ is the covariance matrix of the dataset
- Have to choose h carefully for fast simulation

Bandwidth Optimisation

- Find optimal bandwidth in range of values
- Quantify differences between data and simulation PDFs with Kolmogorow-Smirnow test for various pion energies



$\Rightarrow$ **Bandwidths below 0.01 MIP yield best results**

Multidimensional Discrete Cosine Transform

- In d dimensions, we now have d arrays of lengths N_1, N_2, ..., N_d
- Multidimensional DCT is just product of individual 1D-DCTs, e.g. for three dimensions:

$$X_{k_1,k_2,k_3} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \sum_{n_3=0}^{N_3-1} x_{n_1,n_2,n_3} \times \\ \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right] \cos\left[\frac{\pi}{N_3}\left(n_3+\frac{1}{2}\right)k_3\right]$$

- Inverse of multidimensional DCT (with $\epsilon_i = \frac{1}{2}$ if $k_i = 0$ and else $\epsilon_i = 1$):

$$x_{n_1,n_2,n_3} = \frac{8}{N_1 N_2 N_3} \epsilon_1 \epsilon_2 \epsilon_3 \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} X_{k_1,k_2,k_3} \times \\ \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right] \cos\left[\frac{\pi}{N_3}\left(n_3+\frac{1}{2}\right)k_3\right]$$

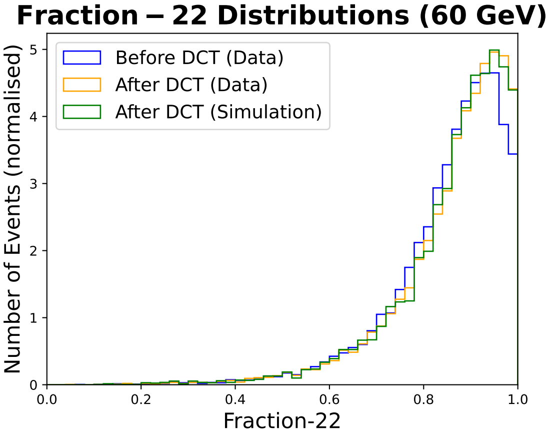
Kinematic Shower Variables

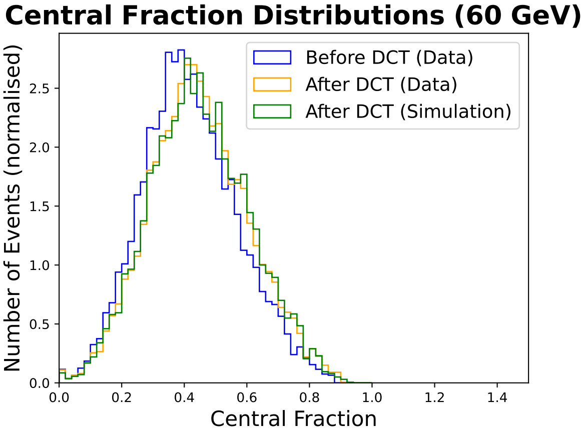
- Total Energy: $E_{\text{total}} = \sum_{\text{hits}} E_{\text{hit}}$
- Centre of Gravity: $\text{CoG}_i = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot i_{\text{hit}}$ for $i \in [x, y, z]$
- Shower Radius: $R = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot r_{\text{hit}}$ with $r_{\text{hit}} = \sqrt{(x_{\text{hit}} - \text{CoG}_x)^2 + (y_{\text{hit}} - \text{CoG}_y)^2}$
- Fraction-22: $f_{22} = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}}$ if $z_{\text{hit}} \leq 22$
- Central Fraction: $f_{\text{central}} = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}}$ if $r_{\text{hit}} \leq 30\,\text{mm}$
- Detached Fraction: $f_{\text{detached}} = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}}$ if hit has at most one neighbouring hit within $\pm 1I, \pm 1J$, and $\pm 2K$

Shower Moments

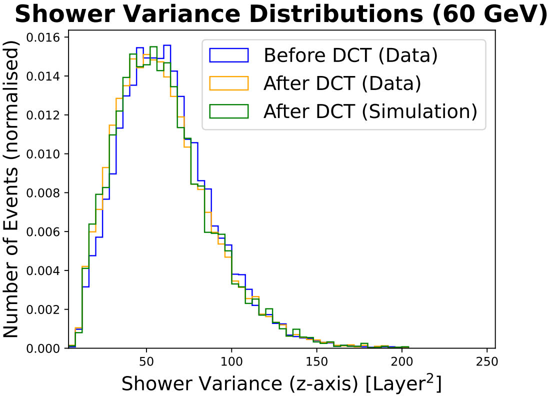
- Shower Variance: $\mathrm{Var}(i) = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot (i_{\text{hit}} - \mathrm{CoG}_i)^2$ for $i \in [x, y, z]$
- Shower Skewness: $\mathrm{Skew}(i) = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot \left(\frac{i_{\text{hit}} - \mathrm{CoG}_i}{\sigma_i}\right)^3$ with $\sigma_i = \sqrt{\mathrm{Var}(i)}$ and $i \in [x, y, z]$
- Shower Kurtosis: $\mathrm{Kurt}(i) = \frac{1}{E_{\text{total}}} \sum_{\text{hits}} E_{\text{hit}} \cdot \left(\frac{i_{\text{hit}} - \mathrm{CoG}_i}{\sigma_i}\right)^4$ with $\sigma_i = \sqrt{\mathrm{Var}(i)}$ and $i \in [x, y, z]$

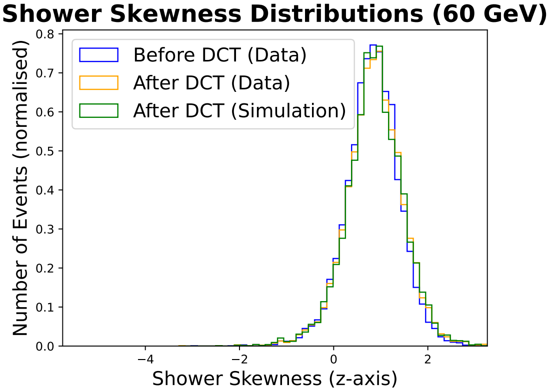
Fraction-22 and Central Fraction





Shower Variance and Skewness (z-axis)





Shower Kurtosis (z-axis)