# Enhancing Geant4 Monte Carlo Simulations through Machine Learning Integration

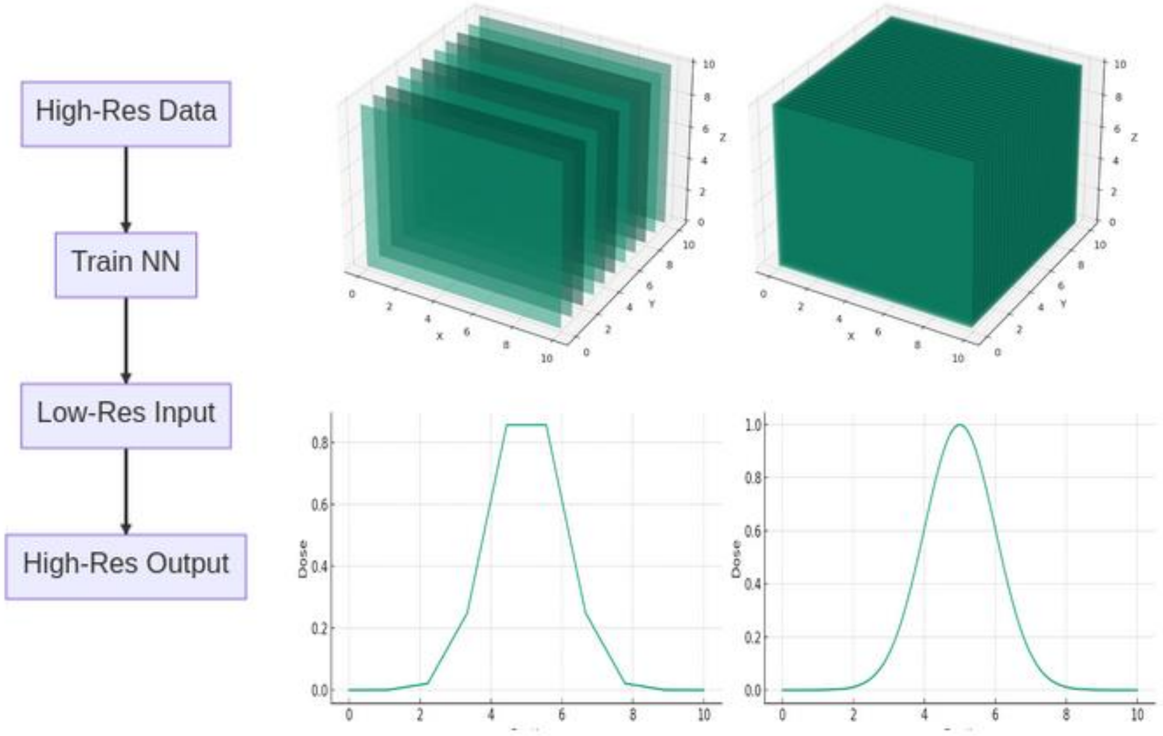
# Spoke 2 - WP6

G.A.P Cirrone, A. Tricomi, S.Fattori, A. Sciuto, G. Gallo, V. Ientile

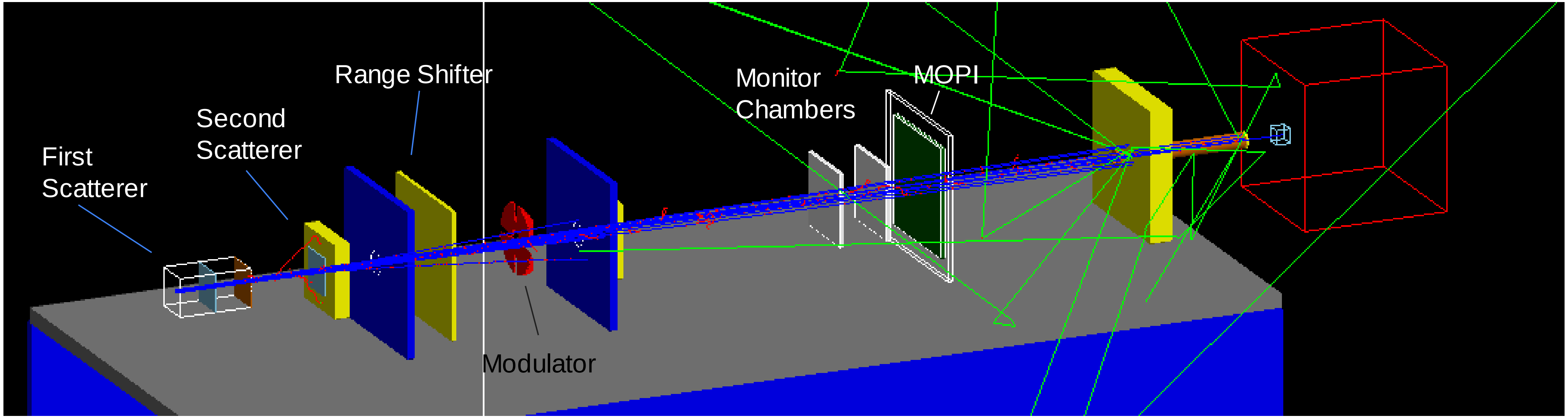
Meeting, 23-01-2024

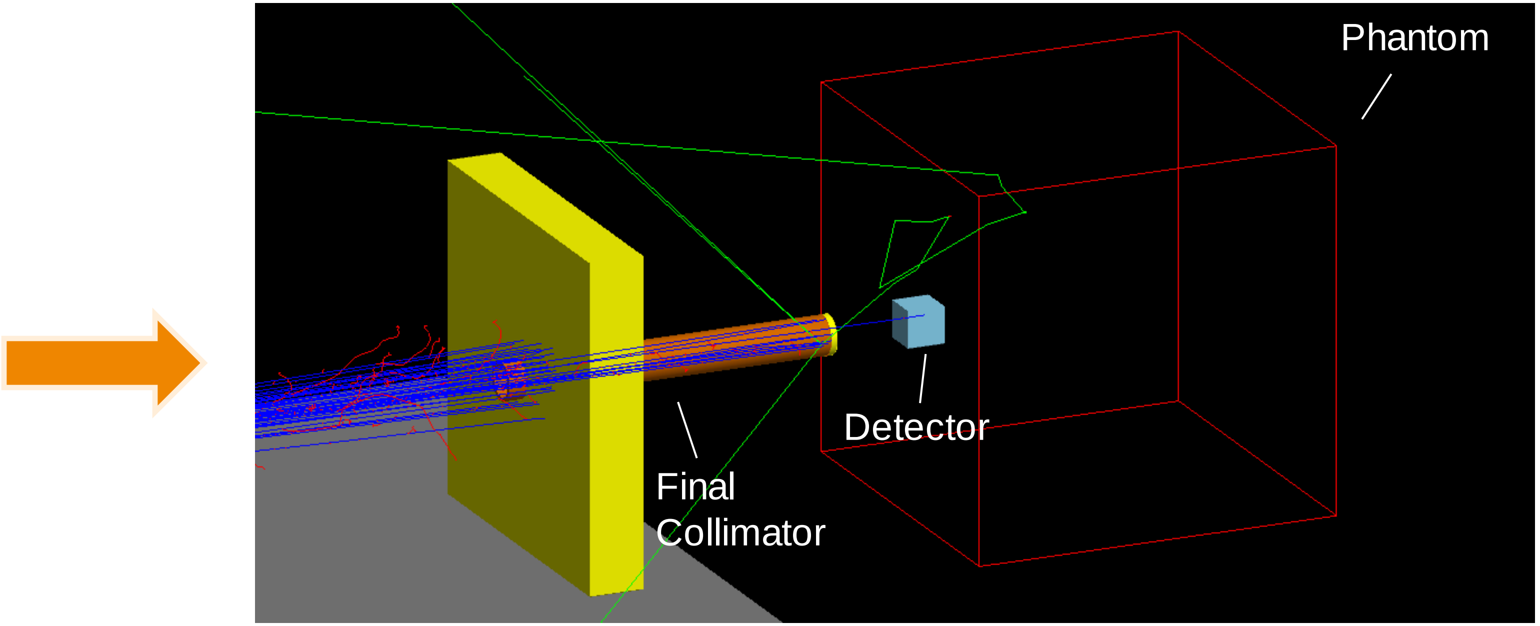
# Autoencoder to enhance low voxel density data

- **Step 1: Data Collection and Preprocessing**
  - Collect high-density and low-density data from detectors.
  - Convert raw data to machine-readable format (CSV, HDF5).
  - Normalize data values to a range (e.g., 0-1).
  - Split data into training, validation, and test sets.
- **Step 2: Install Required Libraries**
- **Step 3: Import Libraries and Load Data**
- **Step 4: Data Partitioning**
- **Step 5: Design the Autoencoder Architecture**
- **Step 6: Compile and Train the Autoencoder**
- **Step 7: Use the Autoencoder to Enhance Low-Density Data**
- **Step 8: Evaluation and Optimization**
- **Step 9: Deploy the Model**



# Hadrontherapy example G4




Geant4-based, open-source application, specifically developed for dosimetric and radiobiological studies with protons and ions beams.

# Execution time, CPU and memory usage

| Cut(um) | Voxelsize(um) | Exec Time | CPU(%) | MEM(%) |
|---|---|---|---|---|
| 1000 | 100 | 00h.00m.11s | 16.72 | 0.08 |
| 100 | 100 | 00h.00m.12s | 18.39 | 0.08 |
| 50 | 100 | 00h.00m.15s | 20.48 | 0.09 |
| 10 | 100 | 00h.00m.26s | 26.96 | 0.09 |
| 1 | 100 | 00h.02m.33s | 31.86 | 0.1 |
| 0.5 | 100 | 00h.02m.36s | 31.83 | 0.1 |
| 1000 | 10 | 00h.00m.43s | 28.99 | 0.1 |
| 100 | 10 | 00h.00m.45s | 29.27 | 0.1 |
| 50 | 10 | 00h.00m.51s | 29.42 | 0.1 |
| 10 | 10 | 00h.01m.28s | 30.98 | 0.1 |
| 1 | 10 | 00h.04m.12s | 32.23 | 0.1 |
| 0.5 | 10 | 00h.04m.10s | 32.26 | 0.1 |
| 1000 | 1 | 00h.06m.14s | 31.72 | 0.1 |
| 100 | 1 | 00h.06m.25s | 31.7 | 0.1 |
| 50 | 1 | 00h.07m.10s | 31.62 | 0.1 |
| 10 | 1 | 00h.08m.47s | 31.85 | 0.1 |
| 1 | 1 | 00h.15m.04s | 32.3 | 0.2 |
| 0.5 | 1 | 00h.15m.00s | 32.25 | 0.2 |

Dell PowerEdge C6525 -
2 x AMD EPYC 7413 24-Core CPU
512 GB RAM

N=32 Threads

run/beamOn 50000

# Table of exec time and CPU usage

| Cut (u) | Voxelsize (u) | Exec Time | CPU (%) | MEM (%) |
|---|---|---|---|---|
| 1000 | 100 | 00h.00m.54s | 78.96 | 0.11 |
| 100 | 100 | 00h.01m.08s | 80.97 | 0.10 |
| 50 | 100 | 00h.01m.37s | 84.47 | 0.12 |
| 10 | 100 | 00h.04m.25s | 88.46 | 0.18 |
| 1 | 100 | 00h.33m.15s | 92.03 | 0.20 |
| 0.5 | 100 | 00h.33m.20s | 92.32 | 0.20 |
| 1000 | 10 | 00h.07m.27s | 89.84 | 0.18 |
| 100 | 10 | 00h.08m.02s | 89.81 | 0.19 |
| 50 | 10 | 00h.09m.08s | 88.72 | 0.19 |
| 10 | 10 | 00h.18m.13s | 90.50 | 0.20 |
| 1 | 10 | 00h.55m.52s | 93.28 | 0.30 |
| 0.5 | 10 | 00h.56m.01s | 92.78 | 0.30 |
| 1000 | 1 | 01h.17m.07s | 90.07 | 0.20 |
| 100 | 1 | 01h.19m.27s | 89.92 | 0.20 |
| 50 | 1 | 01h.24m.57s | 91.63 | 0.22 |
| 10 | 1 | 01h.48m.07s | 92.10 | 0.30 |
| 1 | 1 | 03h.14m.48s | 93.31 | 0.50 |
| 0.5 | 1 | 03h.15m.00s | 93.55 | 0.50 |

N.threads= 96

run/beamOn 1000000

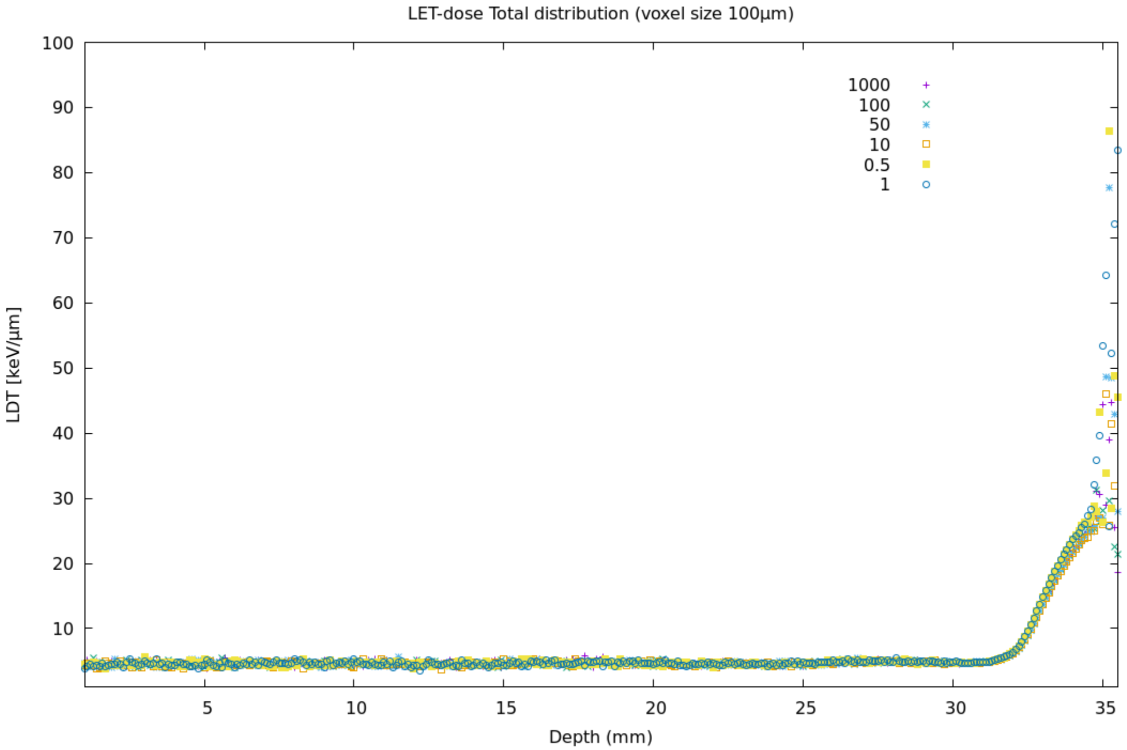
Dose.out

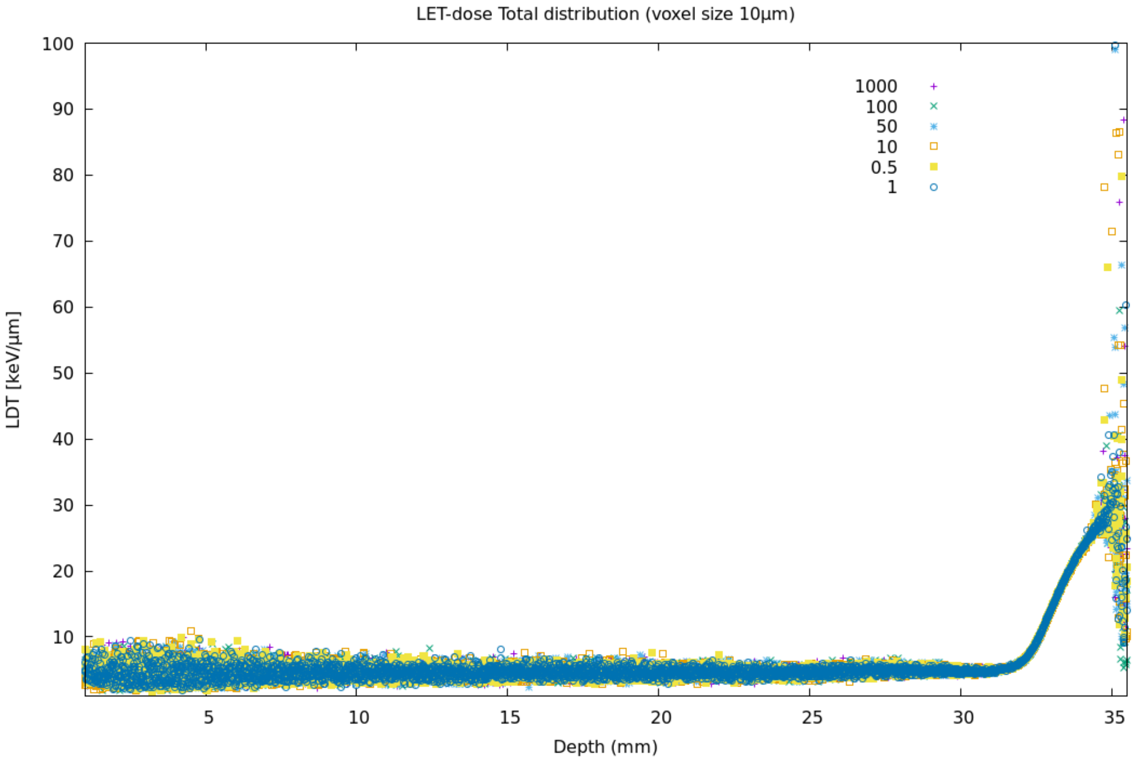
| i | j | k | Dose(Gy) | proton_1 | proton_1_f | proton_1 | proton_1_f | proton_1 | proton_1_f | proton_1 | proton_1 | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 8.92835e-05 | 6.09049e-13 | 0 | 4.6249e-11 | 0 | 4.17558e-11 | 0 | 6.30847e-11 | 0 | 4.28119e-11 | 0 | 3.00401e-11 | | | | | | | |
| 1 | 0 | 0 | 9.34978e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6.99616e-05 | 536605 | 2.77714e-07 | 248 |
| 2 | 0 | 0 | 9.59535e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00421e-05 | 536752 | 2.84383e-07 | 257 |
| 3 | 0 | 0 | 9.66963e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6.99924e-05 | 536858 | 2.96087e-07 | 255 |
| 4 | 0 | 0 | 9.69344e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00438e-05 | 536906 | 2.91541e-07 | 257 |
| 5 | 0 | 0 | 9.73492e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00207e-05 | 536940 | 3.30737e-07 | 265 |
| 6 | 0 | 0 | 9.79645e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00983e-05 | 536935 | 3.29459e-07 | 261 |
| 7 | 0 | 0 | 9.82922e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00245e-05 | 536987 | 3.12056e-07 | 263 |
| 8 | 0 | 0 | 9.86328e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00288e-05 | 537025 | 3.15954e-07 | 267 |
| 9 | 0 | 0 | 9.9158e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.01276e-05 | 537054 | 4.16769e-07 | 271 |
| 10 | 0 | 0 | 9.93448e-05 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7.00375e-05 | 537127 | 4.08863e-07 | 272 |

Let.out

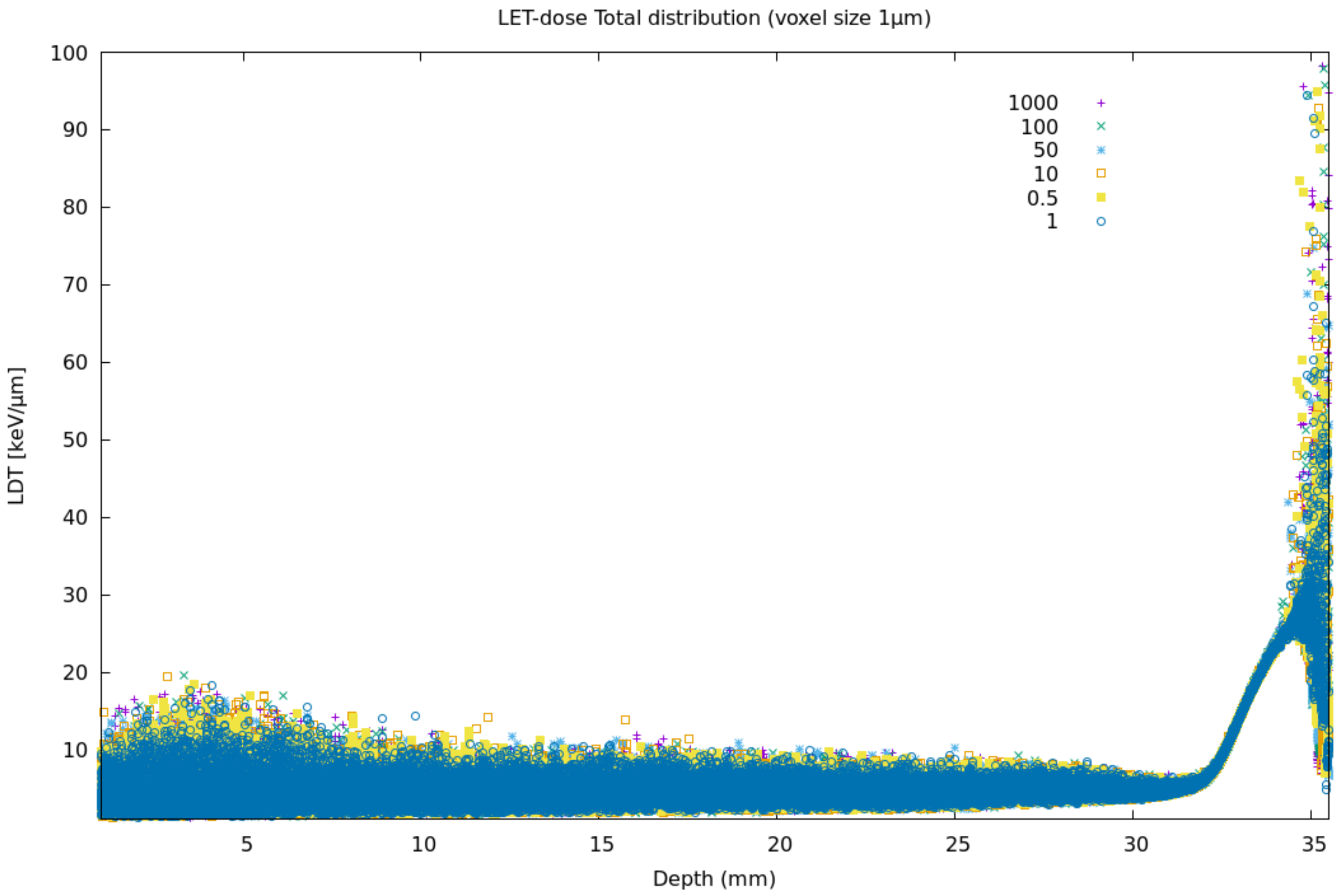
| i | j | k | LDT | LTT | proton_1_D | proton_1_T | proton_D | proton_T | proton_D | proton_T | proton_1_D | proton_1_T | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 2.14327 | 1.04647 | 0 | 0 | 0 | 0 | 9.75109 | 5.40327 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 4.66173 | 1.04903 | 0 | 0 | 0 | 0 | 9.43159 | 5.30463 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 7.60975 | 1.057 | 0 | 0 | 0 | 0 | 10.2542 | 5.36868 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 7.01166 | 1.05542 | 0 | 0 | 0 | 0 | 11.0134 | 5.33326 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 5.12687 | 1.05351 | 0 | 0 | 0 | 0 | 9.89301 | 5.29435 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 3.59234 | 1.05015 | 0 | 0 | 0 | 0 | 15.0083 | 5.8176 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 3.11049 | 1.05016 | 0 | 0 | 0 | 0 | 13.3732 | 5.64901 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 3.53447 | 1.05128 | 0 | 0 | 0 | 0 | 11.3773 | 5.45393 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 5.62656 | 1.05068 | 0 | 0 | 0 | 0 | 10.1942 | 5.46316 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 3.88274 | 1.05236 | 0 | 0 | 0 | 0 | 8.24793 | 4.06994 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 4.85283 | 1.05073 | 0 | 0 | 0 | 0 | 10.2107 | 4.71098 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

# LDT plots


LET-dose Total distribution (voxel size 100µm)
1000
100
50
10
0.5
1
LDT [keV/µm]
Depth (mm)
100
90
80
70
60
50
40
30
20
10
5
10
15
20
25
30
35

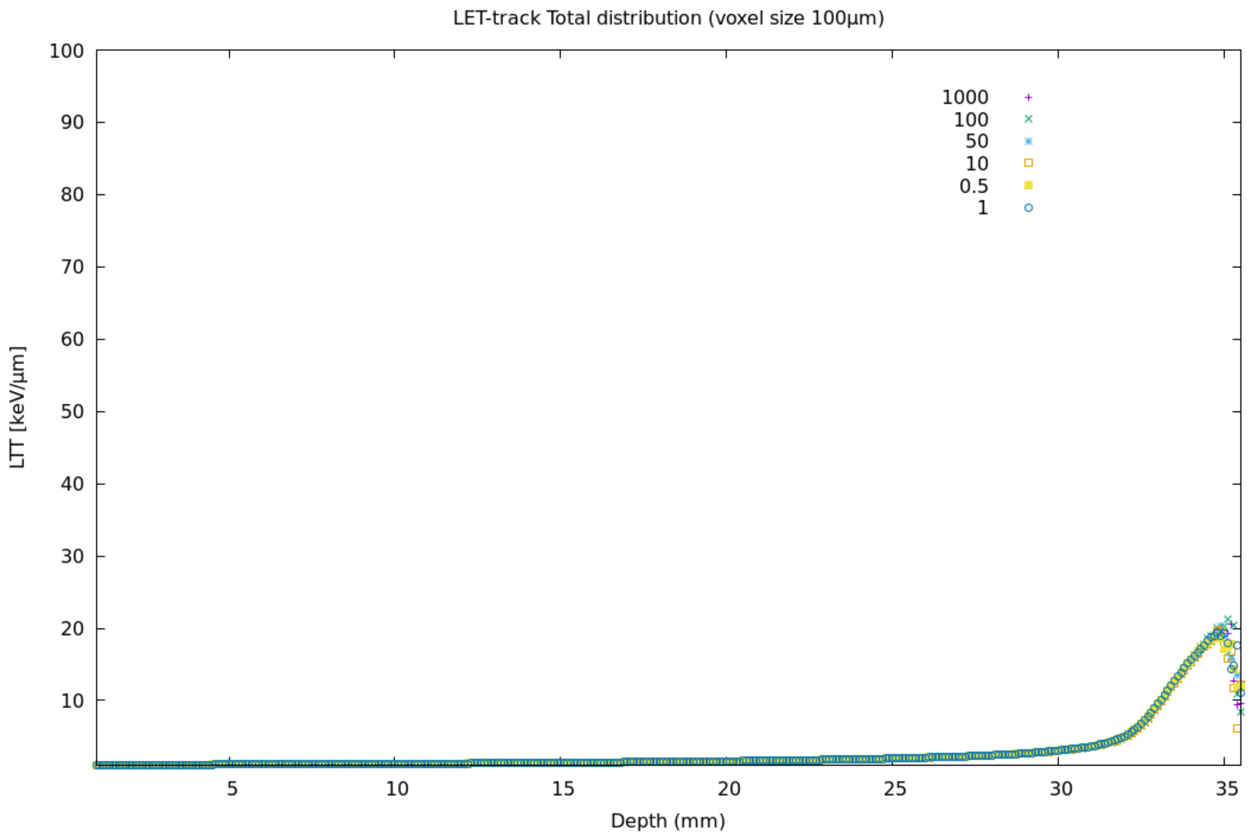

LET-dose Total distribution (voxel size 10µm)
1000
100
50
10
0.5
1
LDT [keV/µm]
Depth (mm)
100
90
80
70
60
50
40
30
20
10
5
10
15
20
25
30
35

# LDT plots

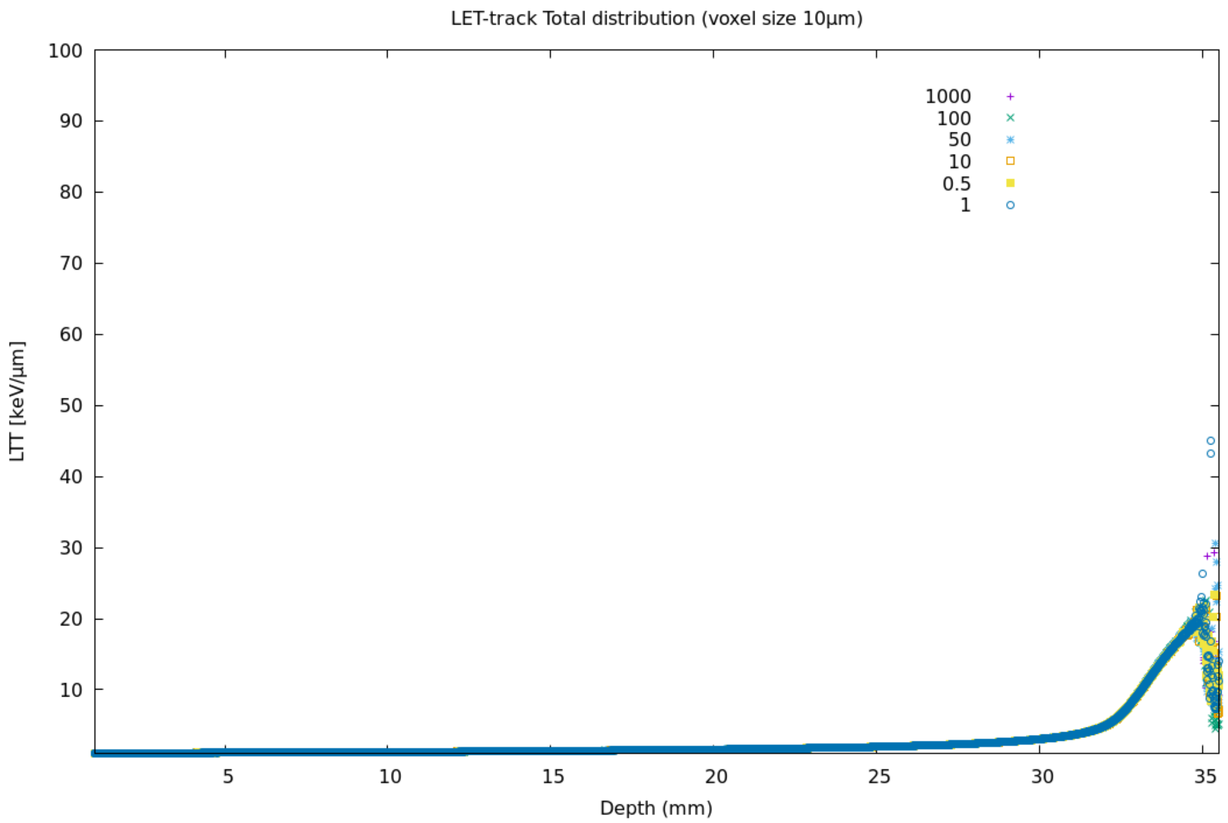


LET-dose Total distribution (voxel size 1µm)
LDT [keV/µm]
Depth (mm)
1000
100
50
10
0.5
1
100
90
80
70
60
50
40
30
20
10
5
10
15
20
25
30
35

# LTT plots

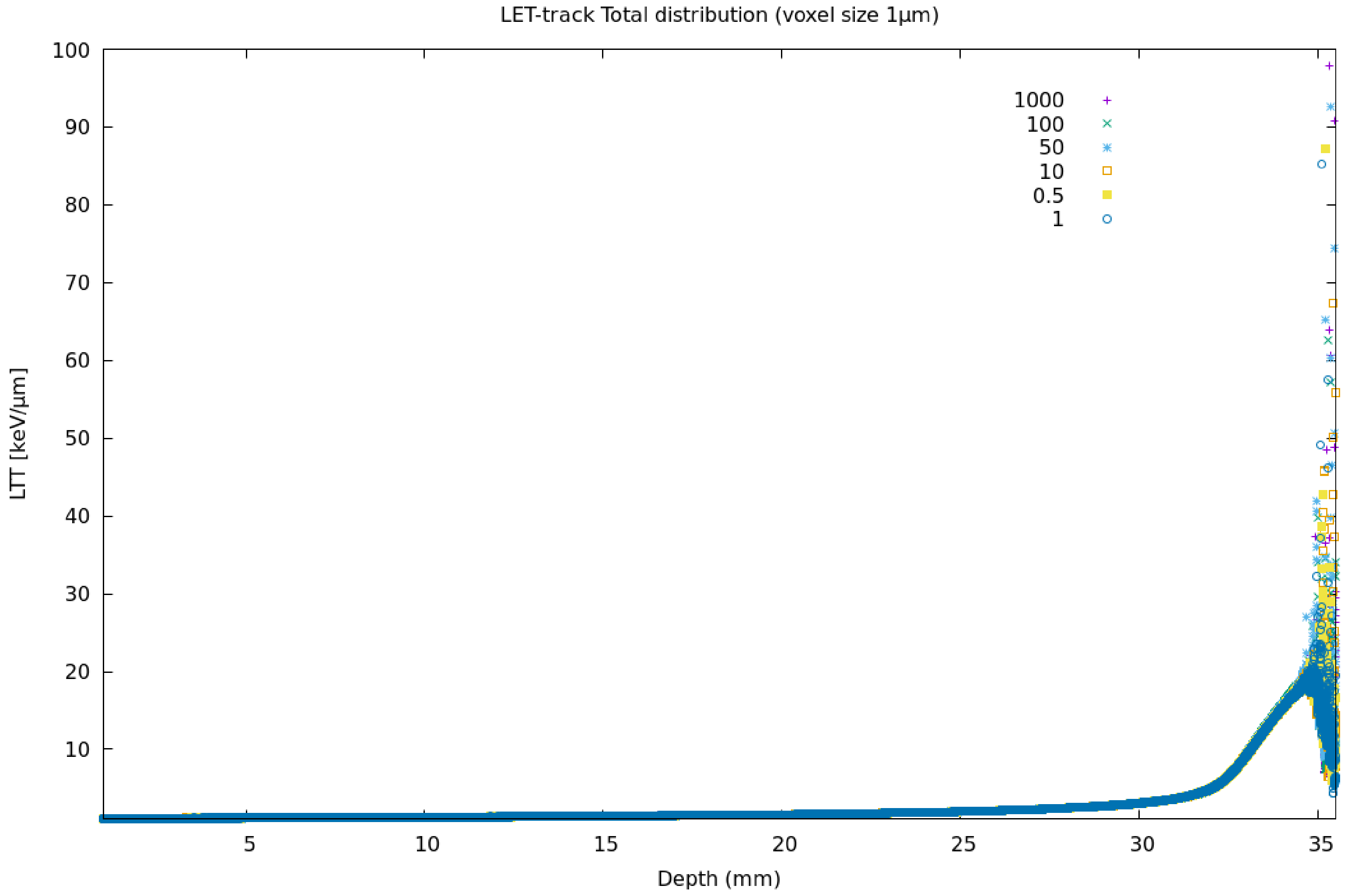


LET-track Total distribution (voxel size 100µm)
1000
100
50
10
0.5
1
LTT [keV/µm]
Depth (mm)

LET-track Total distribution (voxel size 10µm)
1000
100
50
10
0.5
1
LTT [keV/µm]
Depth (mm)

# LTT plots



LET-track Total distribution (voxel size 1μm)
LTT [keV/μm]
100
90
80
70
60
50
40
30
20
10
5
10
15
20
25
30
35
Depth (mm)
1000
100
50
10
0.5
1

GRAZIE

# Enhancing Geant4 Monte Carlo Simulations through Machine Learning Integration

G.A.P Cirrone, S.Fattori, G. Gallo, V. Ientile, A. Sciuto, A. Tricomi

# Initial framework definition

**Contextualized Explanation:**

Training on High-Density Data

The autoencoder would first be trained on the high-density voxelized data, learning to compress this high-quality data into a lower-dimensional latent space and then decompress it back into the original high-density data. During this process, the autoencoder learns the mappings between the high-dimensional space and the lower-dimensional latent space, effectively learning the most important features or "essence" of the high-density data.

Applying to Low-Density Data

Once trained, you can then use the encoder part of the autoencoder to transform your low-density data into the same latent space. Even though the low-density data lacks some features, the encoder will project this data into a compressed form that still captures the most important features based on what it learned from the high-density data. The decoder part will then expand this compressed form back into the original dimensionality.

In this process, the "gaps" in the low-density data are effectively "filled in" by the decoder, as it reconstructs the low-quality data based on the learned important features of the high-quality data. This results in an enhanced version of your low-density data, making it more comparable to the high-density data it was trained on.

# Dataset Overview

DataFrame with loaded data from file: runbeam_1e6/Let_1-1.out

| | i | LDT | LTT | proton_1_D | proton_1_T | proton_1_D.1 | proton_1_T.1 | proton_D | proton_T | proton_1_D.2 | ... | O18_D | O18_T | F17_D | F17_T | F18_D | F18_T | F19_D | F19_T | Ne20_D | Ne20_T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2.63090 | 1.04760 | 0.0 | 0.0 | 1.04081 | 1.04065 | 11.7570 | 6.01532 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 1 | 1.95003 | 1.04636 | 0.0 | 0.0 | 1.04082 | 1.04070 | 12.9513 | 6.09598 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 2 | 2.41322 | 1.04725 | 0.0 | 0.0 | 1.04092 | 1.04075 | 13.8869 | 6.15894 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 3 | 2.15345 | 1.04629 | 0.0 | 0.0 | 1.04089 | 1.04075 | 13.5346 | 6.05996 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 4 | 4.35478 | 1.04927 | 0.0 | 0.0 | 1.04090 | 1.04074 | 12.2653 | 5.99763 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 5 | 4.31605 | 1.05018 | 0.0 | 0.0 | 1.04091 | 1.04077 | 11.9284 | 5.86857 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 6 | 4.44597 | 1.05099 | 0.0 | 0.0 | 1.04095 | 1.04077 | 14.2888 | 6.16782 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 7 | 3.77041 | 1.04961 | 0.0 | 0.0 | 1.04090 | 1.04078 | 13.0829 | 6.10534 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 8 | 2.90049 | 1.04771 | 0.0 | 0.0 | 1.04094 | 1.04082 | 11.4468 | 5.95522 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 9 | 9 | 3.81664 | 1.04859 | 0.0 | 0.0 | 1.04095 | 1.04082 | 10.9422 | 5.84279 | 0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

10 rows × 129 columns

DataFrame with loaded data from file: runbeam_1e6/Let_1-100.out

| | i | LDT | LTT | proton_1_D | proton_1_T | proton_1_D.1 | proton_1_T.1 | proton_1_D.2 | proton_1_T.2 | proton_D | ... | O17_D | O17_T | O18_D | O18_T | F17_D | F17_T | F18_D | F18_T | Ne20_D | Ne20_T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 4.07849 | 1.05314 | 0.0000 | 0.0000 | 0 | 0 | 1.04243 | 1.04177 | 13.1367 | ... | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 1 | 4.18778 | 1.05676 | 0.0000 | 0.0000 | 0 | 0 | 1.04595 | 1.04335 | 13.3717 | ... | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 2 | 4.15443 | 1.05973 | 0.0000 | 0.0000 | 0 | 0 | 1.04869 | 1.04472 | 13.6434 | ... | 364.684 | 364.684 | 0.000 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 3 | 4.28368 | 1.06016 | 0.0000 | 0.0000 | 0 | 0 | 1.05032 | 1.04619 | 12.4763 | ... | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 4 | 4.40955 | 1.06272 | 1.0467 | 1.0467 | 0 | 0 | 1.05011 | 1.04759 | 11.8088 | ... | 0.000 | 0.000 | 255.106 | 234.794 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 5 | 4.95228 | 1.06526 | 0.0000 | 0.0000 | 0 | 0 | 1.05650 | 1.04930 | 11.9446 | ... | 0.000 | 0.000 | 157.720 | 157.720 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 6 | 4.25197 | 1.06573 | 0.0000 | 0.0000 | 0 | 0 | 1.05811 | 1.05069 | 11.2372 | ... | 0.000 | 0.000 | 0.000 | 0.000 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 7 | 4.65704 | 1.06751 | 0.0000 | 0.0000 | 0 | 0 | 1.05795 | 1.05199 | 11.4878 | ... | 0.000 | 0.000 | 192.027 | 192.027 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 8 | 4.19457 | 1.06913 | 0.0000 | 0.0000 | 0 | 0 | 1.06176 | 1.05355 | 10.8126 | ... | 278.604 | 278.604 | 390.294 | 390.294 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 9 | 9 | 4.38071 | 1.07270 | 0.0000 | 0.0000 | 0 | 0 | 1.06316 | 1.05500 | 10.4814 | ... | 0.000 | 0.000 | 508.197 | 396.495 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

10 rows × 79 columns

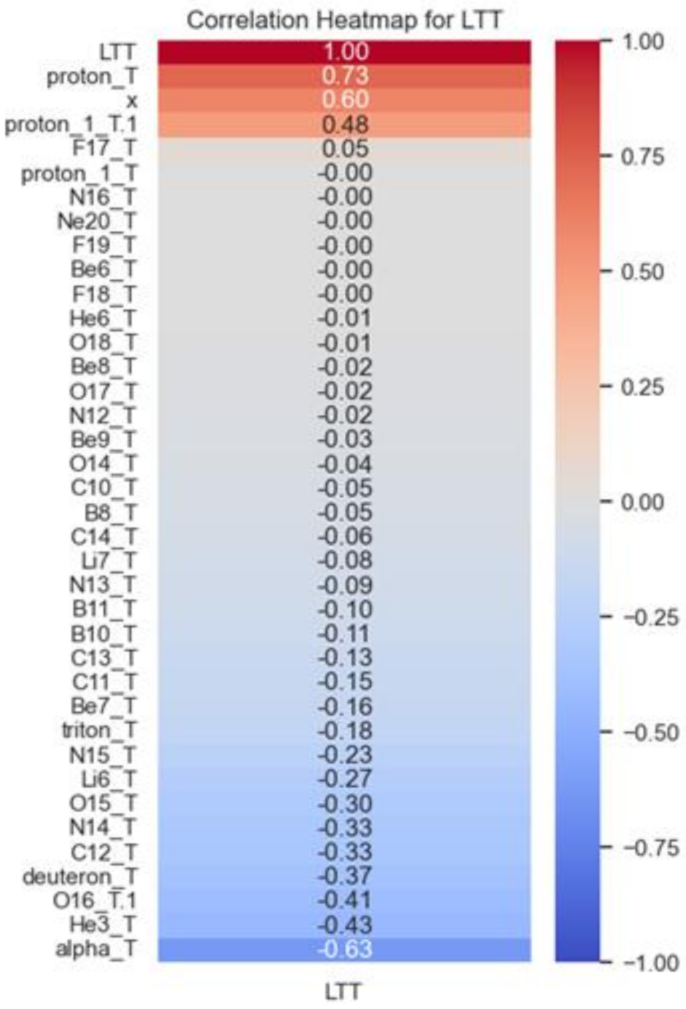
# Build the Dataset

- Use the LET and contributions to LET from primaries and fragments as features

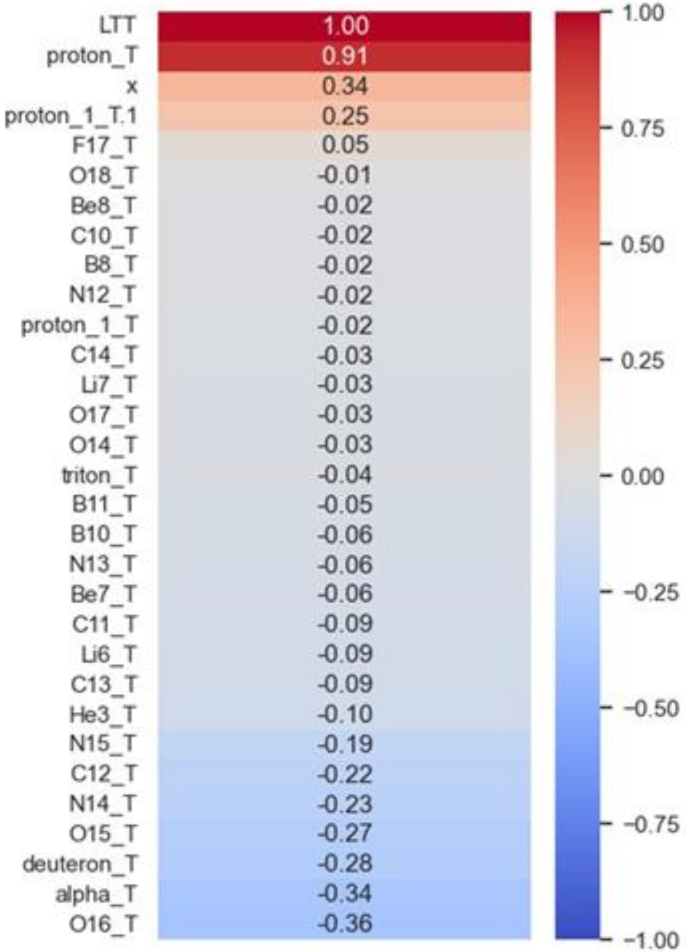
- Create a dataset consisting solely of LTT or LDT vectors (similar to image dataset)

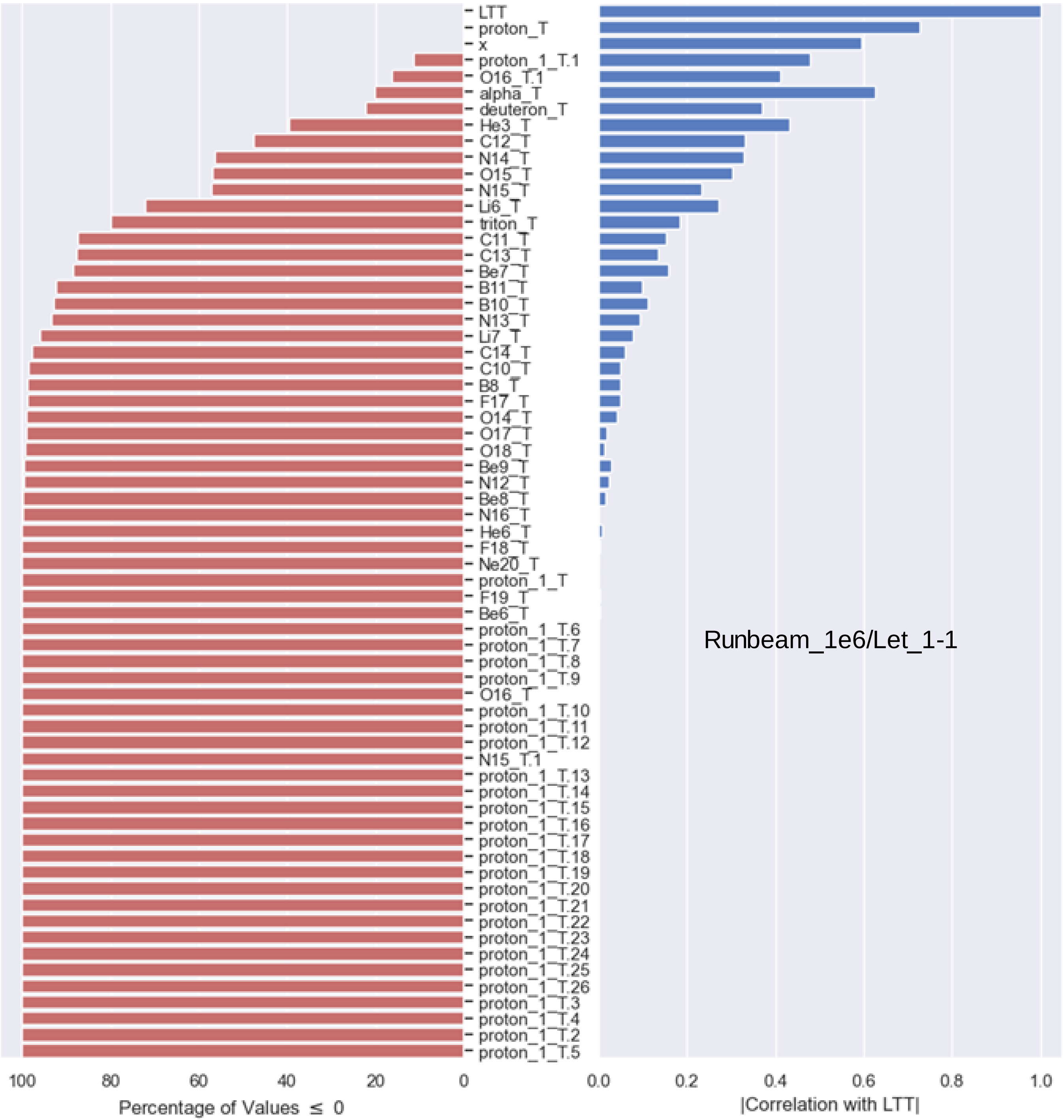
# Feature Correlation

Runbeam_1e6/Let_1-1




Runbeam_5e4/Let_1000-100






# Feature selection

- Feature correlation appears to be related to the percentage of zero values in each feature.
- In principle, features that are not correlated could be dropped from the data set.