Machine Learning with Muon Collider Data

From .slcio files to Neural Networks in Hardware

Abhinav M
University of Hyderabad

Objectives

- Study differences between Beam Induced Background and Hard Scattering data
- Train a DL model to classify BIB/Hard based on these differences
- Implement the model on FPGA using hls4ml

Beam Induced Background (BIB)

Muons are unstable !

$$\underbrace{\text{In flight decays} + \text{decay products from interaction with matter}}_{\text{result in}}$$

Beam Induced Background

How can on-chip ML help ?

A good model can characterize BIB noise and eliminate it from the signal data.

- Avoids loss of data due to cutoff based filters.
- Can potentially discover new physics.
- Save resources by avoiding noise.
- Reduce hardware load.

Train a Deep Learning model to reject BIB minimizing loss of signal data.

Opportunities for on-chip ML

Exploit characteristic differences between BIB and signal.

- Pulse shape
- Arrival time distribution
- Hit distance distribution
- Energy profiles
- ...

GEANT4 Data → csv

The data

- GEANT4 simulation output files were received for BIB and Hard interactions.
- Files were in .slcio format.

BIB data→ /cvmfs/cms.hep.wisc.edu/mucol/reference/slomte_BIBsamples

Hard data → /scratch/slomte/analysis/inputfiles/sim_mumuHbb3TeV_100evts.slcio

- First these files were converted into plaintext to access the data.

The data

```
      Event  : 0 - run:  0 - timestamp 0 - weight 1

date:      01.01.1970  00:00:00.000000000
detector : CLIC_o3_v14_mod4
event parameters:

collection name : ECalBarrelCollection
parameters:

-------------- print out of SimCalorimeterHit collection --------------

 flag:  0xb0000000
parameter CellIDEncoding [string]: system:0:5,side:5:-2,module:7:8,stave:15:4,layer:19:9,submodule:28:4,x:32:-16,y:48:-16,
 -> LCIO::CHBIT_LONG   : 1
    LCIO::CHBIT_BARREL : 0
    LCIO::CHBIT_ID1    : 1
    LCIO::CHBIT_STEP   : 1

[   id   ] |cellId0 |cellId1 |  energy  |        position (x,y,z)          | nMCParticles
          -> MC contribution: prim. PDG |  energy  |   time   | length   | sec. PDG | stepPosition (x,y,z)
[00000029] |03671060|10289130|+1.688e-04|-6.717e+02, +1.388e+03, +7.956e+02|          +1
       id-fields: (system:20,side:0,module:8,stave:0,layer:7,submodule:0,x:-22,y:156)
          ->                          +0|+1.688e-04|+6.376e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000030] |03671060|10289129|+6.282e-04|-6.673e+02, +1.390e+03, +7.956e+02|          +2
       id-fields: (system:20,side:0,module:8,stave:0,layer:7,submodule:0,x:-23,y:156)
          ->                          +0|+1.428e-04|+6.376e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
          ->                          +0|+4.854e-04|+6.057e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000031] |03146516|10223608|+1.472e-03|-1.533e+03, +4.080e+01, +7.905e+02|          +2
       id-fields: (system:20,side:0,module:6,stave:0,layer:6,submodule:0,x:-8,y:155)
          ->                          +0|+4.745e-04|+6.344e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
          ->                          +0|+9.974e-04|+5.543e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000032] |00000660|-6750180|+1.418e-04|-1.230e+03, -8.749e+02, -5.253e+02|          +1
       id-fields: (system:20,side:0,module:5,stave:0,layer:0,submodule:0,x:28,y:-103)
          ->                          +0|+1.418e-04|+6.096e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000033] |02098196|-13762538|+1.930e-04|-8.585e+02, +1.263e+03, -1.071e+03|          +1
       id-fields: (system:20,side:0,module:8,stave:0,layer:4,submodule:0,x:22,y:-210)
          ->                          +0|+1.930e-04|+7.308e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000034] |20448660|-15269951|+2.738e-04|+1.632e+03, +5.714e+02, -1.193e+03|          +1
       id-fields: (system:20,side:0,module:11,stave:0,layer:39,submodule:0,x:-63,y:-234)
          ->                          +2|+2.738e-04|+9.607e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
```

Parsing the data

- Each file had around 500,000 lines and each line was in below format:

```
[   id   ] |cellId0 |cellId1 |  energy  |        position (x,y,z)         | nMCParticles
          → MC contribution: prim. PDG |  energy  |   time    | length   | sec. PDG | stepPosition (x,y,z)
───────────┼─────────┼─────────┼──────────┼──────────────────────────────────┼──────────────
[00000029] |03671060|10289130|+1.688e-04|-6.717e+02, +1.388e+03, +7.956e+02|          +1
       id-fields: (system:20,side:0,module:8,stave:0,layer:7,submodule:0,x:-22,y:156)
          →                           +0|+1.688e-04|+6.376e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000030] |03671060|10289129|+6.282e-04|-6.673e+02, +1.390e+03, +7.956e+02|          +2
       id-fields: (system:20,side:0,module:8,stave:0,layer:7,submodule:0,x:-23,y:156)
          →                           +0|+1.428e-04|+6.376e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
          →                           +0|+4.854e-04|+6.057e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
[00000031] |03146516|10223608|+1.472e-03|-1.533e+03, +4.080e+01, +7.905e+02|          +2
       id-fields: (system:20,side:0,module:6,stave:0,layer:6,submodule:0,x:-8,y:155)
          →                           +0|+4.745e-04|+6.344e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
          →                           +0|+9.974e-04|+5.543e+00|+0.000e+00| (+0, +0.000e+00, +0.000e+00, +0.000e+00)
```

- This data was loaded into a pandas dataframe in python for visualising and understanding the data.

Non-standard format

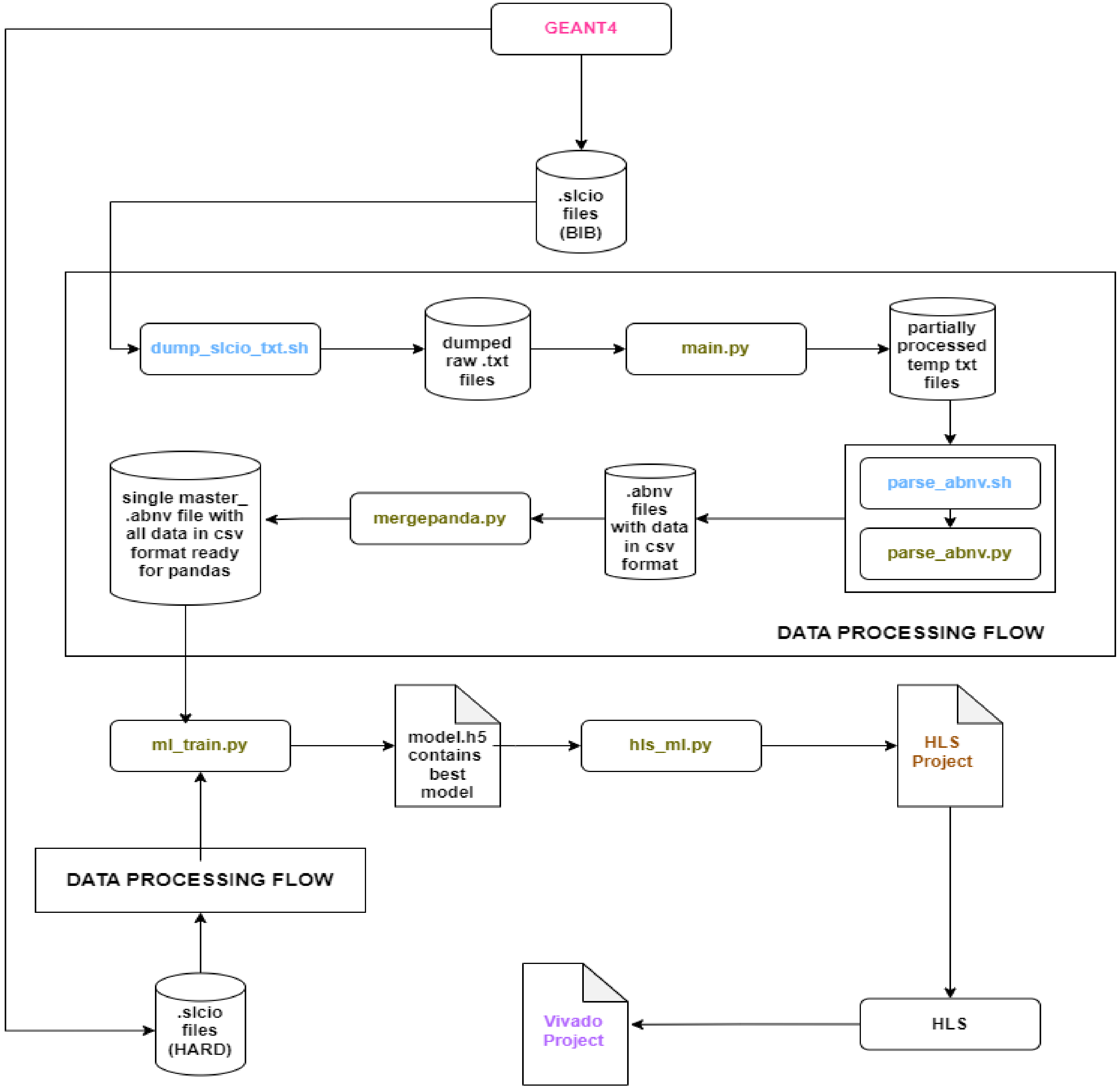
```
           ->                       +211|+2.105e-03|+8.827e+00|+7.079e-01| (+1000010020, +1.861e+02, +1.523e+03, +1.970e+03)
           ->                       +211|+6.138e-04|+8.805e+00|+5.636e-01| (+2212, +1.848e+02, +1.523e+03, +1.969e+03)
[00000036] |01048596|20447170|+6.233e-04|+1.512e+03, -3.162e+02, +1.586e+03|        +14
       id-fields: (system:20,side:0,module:0,stave:0,layer:2,submodule:0,x:-62,y:311)
           ->                       +211|+8.675e-05|+9.102e+02|+2.453e-01| (+11, +1.512e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+1.862e-04|+9.102e+02|+2.534e-01| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+4.839e-06|+9.102e+02|+2.400e-02| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+1.622e-05|+9.102e+02|+6.871e-02| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+1.303e-05|+9.102e+02|+6.028e-02| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+9.181e-05|+9.102e+02|+5.612e-02| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+1.833e-05|+9.102e+02|+7.146e-02| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+4.622e-05|+9.102e+02|+1.153e-01| (+11, +1.513e+03, -3.168e+02, +1.587e+03)
           ->                       +211|+4.085e-05|+9.102e+02|+1.327e-01| (+11, +1.512e+03, -3.167e+02, +1.587e+03)
           ->                       +211|+2.534e-05|+9.102e+02|+1.057e-01| (+11, +1.512e+03, -3.166e+02, +1.587e+03)
           ->                       +211|+4.071e-05|+9.102e+02|+6.560e-02| (+11, +1.512e+03, -3.166e+02, +1.587e+03)
           ->                       +211|+1.392e-05|+9.102e+02|+5.045e-02| (+11, +1.512e+03, -3.166e+02, +1.587e+03)
           ->                       +211|+2.336e-05|+9.102e+02|+6.216e-02| (+11, +1.512e+03, -3.165e+02, +1.587e+03)
           ->                       +211|+1.567e-05|+9.102e+02|+4.271e-02| (+11, +1.512e+03, -3.165e+02, +1.587e+03)
[00000037] |03671188|24444903|+7.703e-06|+1.275e+02, +1.538e+03, +1.897e+03|         +1
       id-fields: (system:20,side:0,module:9,stave:0,layer:7,submodule:0,x:-25,y:372)
           ->                       +211|+7.703e-06|+4.164e+01|+5.776e-01| (+2112, +1.276e+02, +1.538e+03, +1.897e+03)
[00000038] |06292628|24838110|+2.685e-04|+1.734e+02, +1.563e+03, +1.928e+03|         +2
       id-fields: (system:20,side:0,module:9,stave:0,layer:12,submodule:0,x:-34,y:378)
           ->                       +211|+5.211e-05|+1.524e+01|+8.163e-02| (+11, +1.713e+02, +1.563e+03, +1.927e+03)
           ->                       +211|+2.164e-04|+1.524e+01|+9.157e-02| (+11, +1.713e+02, +1.563e+03, +1.927e+03)
[00000039] |07340308|20709441|+5.952e-05|+1.074e+03, -1.197e+03, +1.612e+03|         +1
       id-fields: (system:20,side:0,module:2,stave:0,layer:14,submodule:0,x:65,y:316)
           ->                       +211|+5.952e-05|+1.762e+02|+5.263e-01| (+2112, +1.075e+03, -1.196e+03, +1.609e+03)
[00000040] |08388884|20578376|+2.951e-05|+1.110e+03, -1.187e+03, +1.601e+03|         +1
       id-fields: (system:20,side:0,module:2,stave:0,layer:16,submodule:0,x:72,y:314)
           ->                       +211|+2.951e-05|+1.791e+02|+2.988e-03| (+11, +1.110e+03, -1.188e+03, +1.602e+03)
[00000041] |05244052|24707042|+2.921e-04|+1.530e+02, +1.553e+03, +1.918e+03|         +9
       id-fields: (system:20,side:0,module:9,stave:0,layer:10,submodule:0,x:-30,y:376)
           ->                       +211|+2.782e-05|+1.576e+01|+9.234e-02| (+11, +1.521e+02, +1.553e+03, +1.919e+03)
           ->                       +211|+3.528e-05|+1.576e+01|+9.282e-02| (+11, +1.520e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+2.356e-05|+1.576e+01|+1.012e-01| (+11, +1.520e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+2.484e-05|+1.576e+01|+1.176e-01| (+11, +1.519e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+3.553e-05|+1.576e+01|+1.072e-01| (+11, +1.519e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+3.548e-05|+1.576e+01|+1.431e-01| (+11, +1.518e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+3.538e-05|+1.576e+01|+1.204e-01| (+11, +1.518e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+3.412e-05|+1.576e+01|+9.836e-02| (+11, +1.518e+02, +1.553e+03, +1.920e+03)
           ->                       +211|+4.004e-05|+1.576e+01|+1.189e-01| (+11, +1.517e+02, +1.553e+03, +1.920e+03)
[00000042] |05244052|24772578|+1.170e-04|+1.530e+02, +1.553e+03, +1.923e+03|         +4
```

.abnv files
with data in .csv
for each event

```
CollectionID,PPDG,Energy,Time,Length,SPDG,X,Y,Z
ECBC,211,0.0003664,8.623,1.336,211,158.9,1502.0,1929.0
ECBC,211,0.0003844,8.669,1.391,211,166.2,1507.0,1940.0
ECBC,211,0.0003804,8.716,1.391,211,173.7,1512.0,1951.0
ECBC,211,0.0004748,8.763,1.406,211,181.0,1518.0,1962.0
ECBC,211,0.002105,8.827,0.7079,1000010020,186.1,1523.0,1970.0
ECBC,211,0.0006138,8.805,0.5636,2212,184.8,1523.0,1969.0
ECBC,211,8.675e-05,910.2,0.2453,11,1512.0,-316.8,1587.0
ECBC,211,0.0001862,910.2,0.2534,11,1513.0,-316.8,1587.0
ECBC,211,4.839e-06,910.2,0.024,11,1513.0,-316.8,1587.0
ECBC,211,1.622e-05,910.2,0.06871,11,1513.0,-316.8,1587.0
ECBC,211,1.303e-05,910.2,0.06028,11,1513.0,-316.8,1587.0
ECBC,211,9.181e-05,910.2,0.05612,11,1513.0,-316.8,1587.0
ECBC,211,1.833e-05,910.2,0.07146,11,1513.0,-316.8,1587.0
ECBC,211,4.622e-05,910.2,0.1153,11,1513.0,-316.8,1587.0
ECBC,211,4.085e-05,910.2,0.1327,11,1512.0,-316.7,1587.0
ECBC,211,2.534e-05,910.2,0.1057,11,1512.0,-316.6,1587.0
ECBC,211,4.071e-05,910.2,0.0656,11,1512.0,-316.6,1587.0
ECBC,211,1.392e-05,910.2,0.05045,11,1512.0,-316.6,1587.0
ECBC,211,2.336e-05,910.2,0.06216,11,1512.0,-316.5,1587.0
ECBC,211,1.567e-05,910.2,0.04271,11,1512.0,-316.5,1587.0
ECBC,211,7.703e-06,41.64,0.5776,2112,127.6,1538.0,1897.0
ECBC,211,5.211e-05,15.24,0.08163,11,171.3,1563.0,1927.0
ECBC,211,0.0002164,15.24,0.09157,11,171.3,1563.0,1927.0
ECBC,211,5.952e-05,176.2,0.5263,2112,1075.0,-1196.0,1609.0
ECBC,211,2.951e-05,179.1,0.002988,11,1110.0,-1188.0,1602.0
ECBC,211,2.782e-05,15.76,0.09234,11,152.1,1553.0,1919.0
ECBC,211,3.528e-05,15.76,0.09282,11,152.0,1553.0,1920.0
ECBC,211,2.356e-05,15.76,0.1012,11,152.0,1553.0,1920.0
ECBC,211,2.484e-05,15.76,0.1176,11,151.9,1553.0,1920.0
ECBC,211,3.553e-05,15.76,0.1072,11,151.9,1553.0,1920.0
```

.slcio → pandas df

.py files take command line arguments for plotting

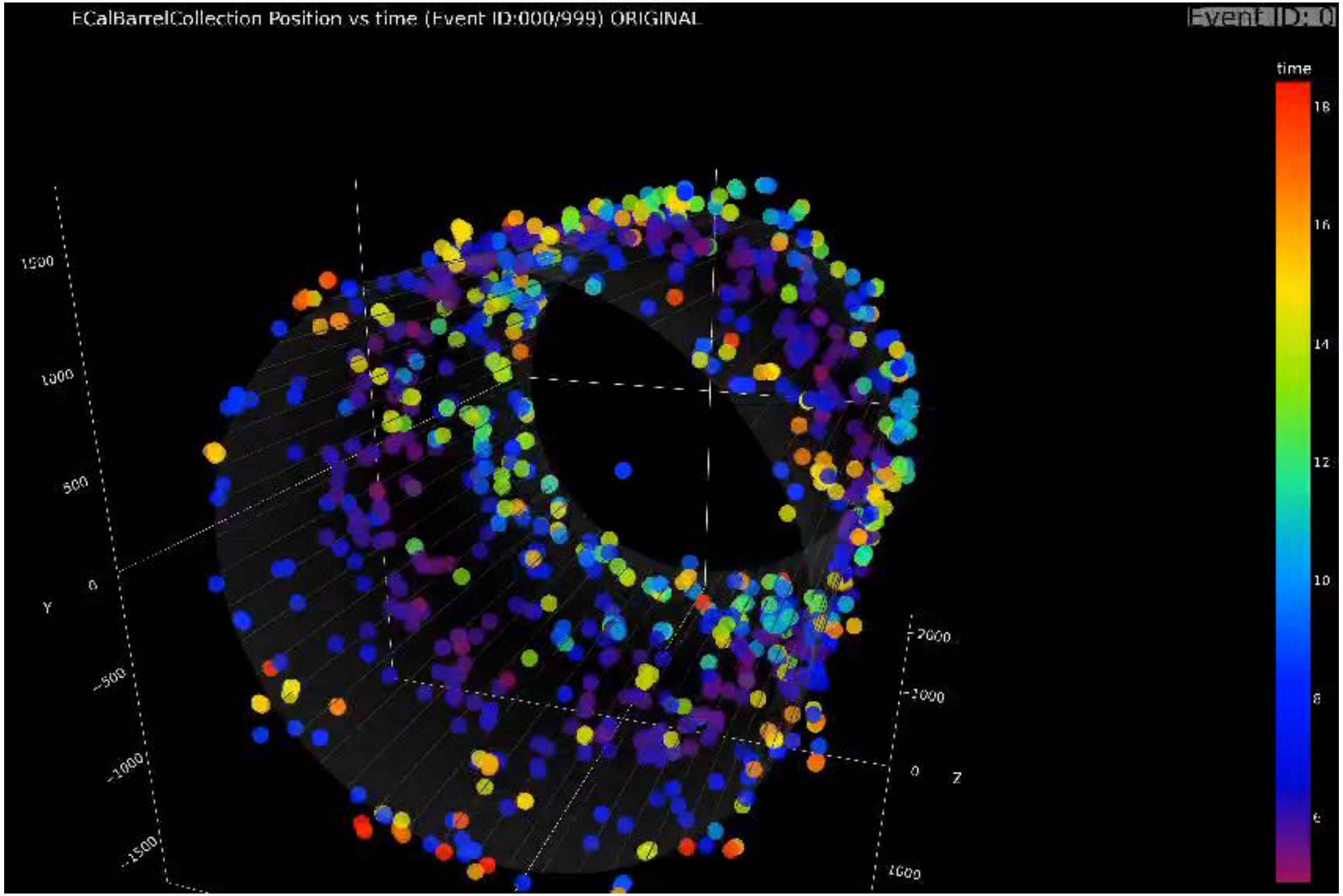


The data was correctly extracted into a pandas df

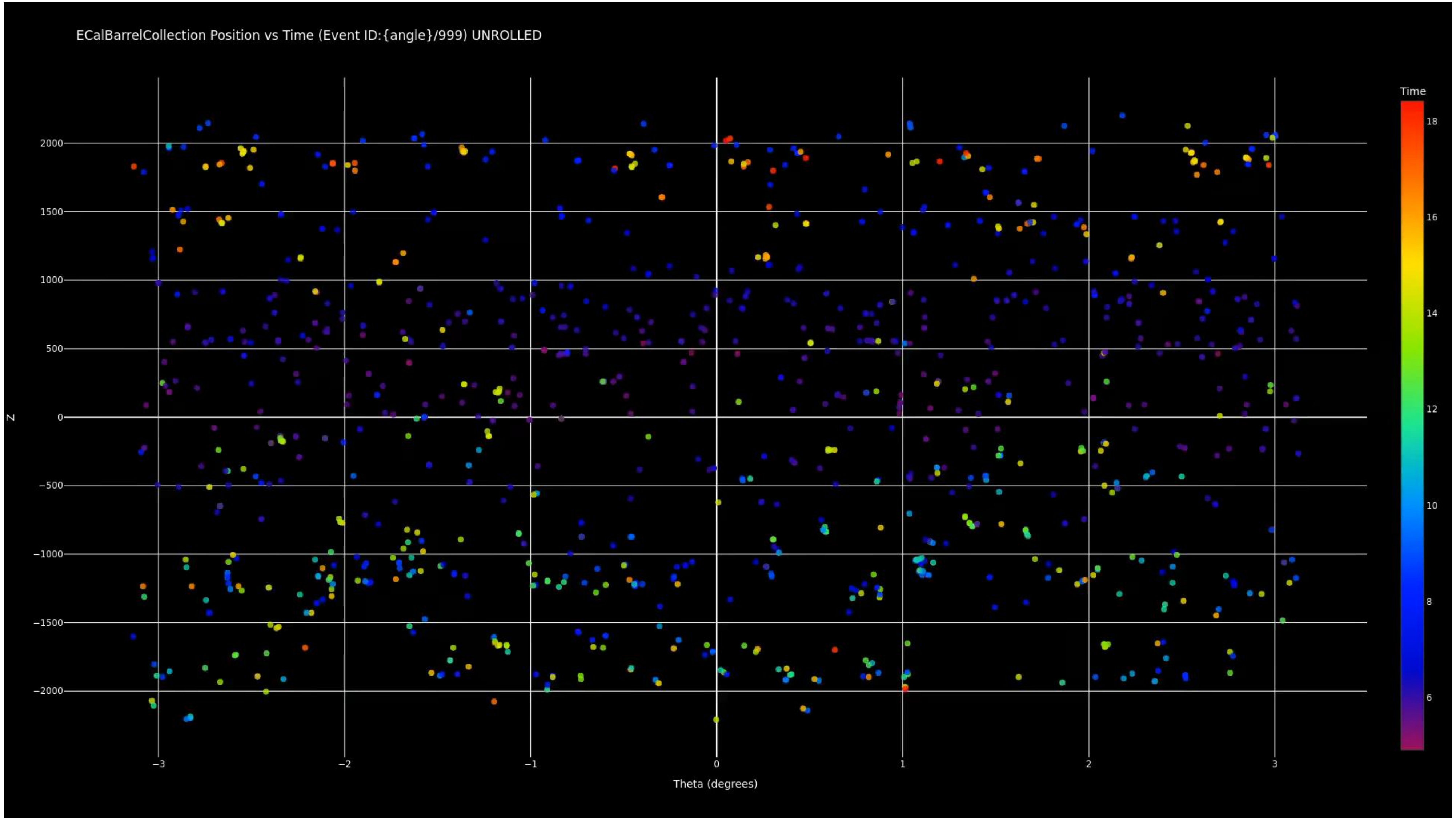
	CollectionID	PPDG	Energy	Time	Length	SPDG	X	Y	Z	Dataset
0	ECBC	-321	2.381000e-04	863.7	0.73080	11	1502.0	87.12	-144.1	Hard
1	ECBC	-321	2.997000e-04	863.7	0.70900	11	1502.0	87.69	-144.5	Hard
2	ECBC	-321	9.577000e-05	863.7	0.36890	11	1502.0	88.00	-144.8	Hard
3	ECBC	-321	7.138000e-05	1376.0	0.01415	11	-373.5	1588.00	-1571.0	Hard
4	ECBC	-321	7.435000e-06	396.8	0.02096	11	1503.0	317.30	695.0	Hard
...	...	...	...	...	...	...	...	...	...	...
5470951	HCEC	2112	3.184000e-08	519.7	3.15500	2112	1205.0	1491.00	-434.4	BIB
5470952	HCEC	2112	9.368000e-09	556.3	7.48300	2112	1209.0	1489.00	-431.4	BIB
5470953	HCEC	2112	6.501000e-08	570.0	2.73100	2112	1212.0	1486.00	-428.8	BIB
5470954	HCEC	2112	6.400000e-09	579.1	1.11300	2112	1213.0	1485.00	-427.8	BIB
5470955	HCEC	2112	8.332000e-09	1304.0	0.42920	2112	1243.0	1498.00	-421.7	BIB

Around 5.47M rows were available with 50-50 Hard-BIB
Of this, I focused on the ECalBarrelCollection

Some BIB visualizations: Arrival time vs XYZ



Some BIB visualizations: Arrival time vs XYZ

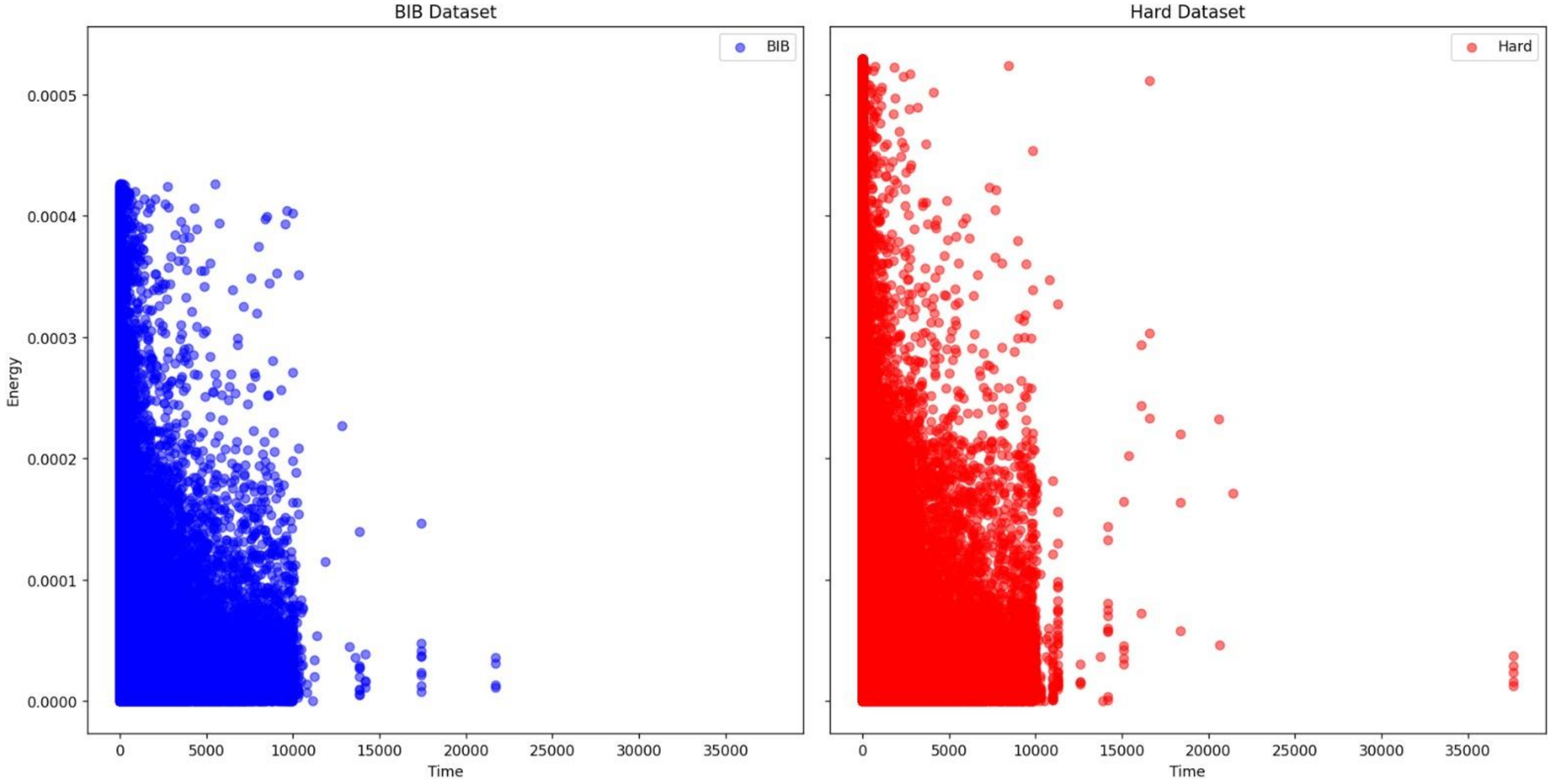


BIB vs Hard for ECalBarrelCollection

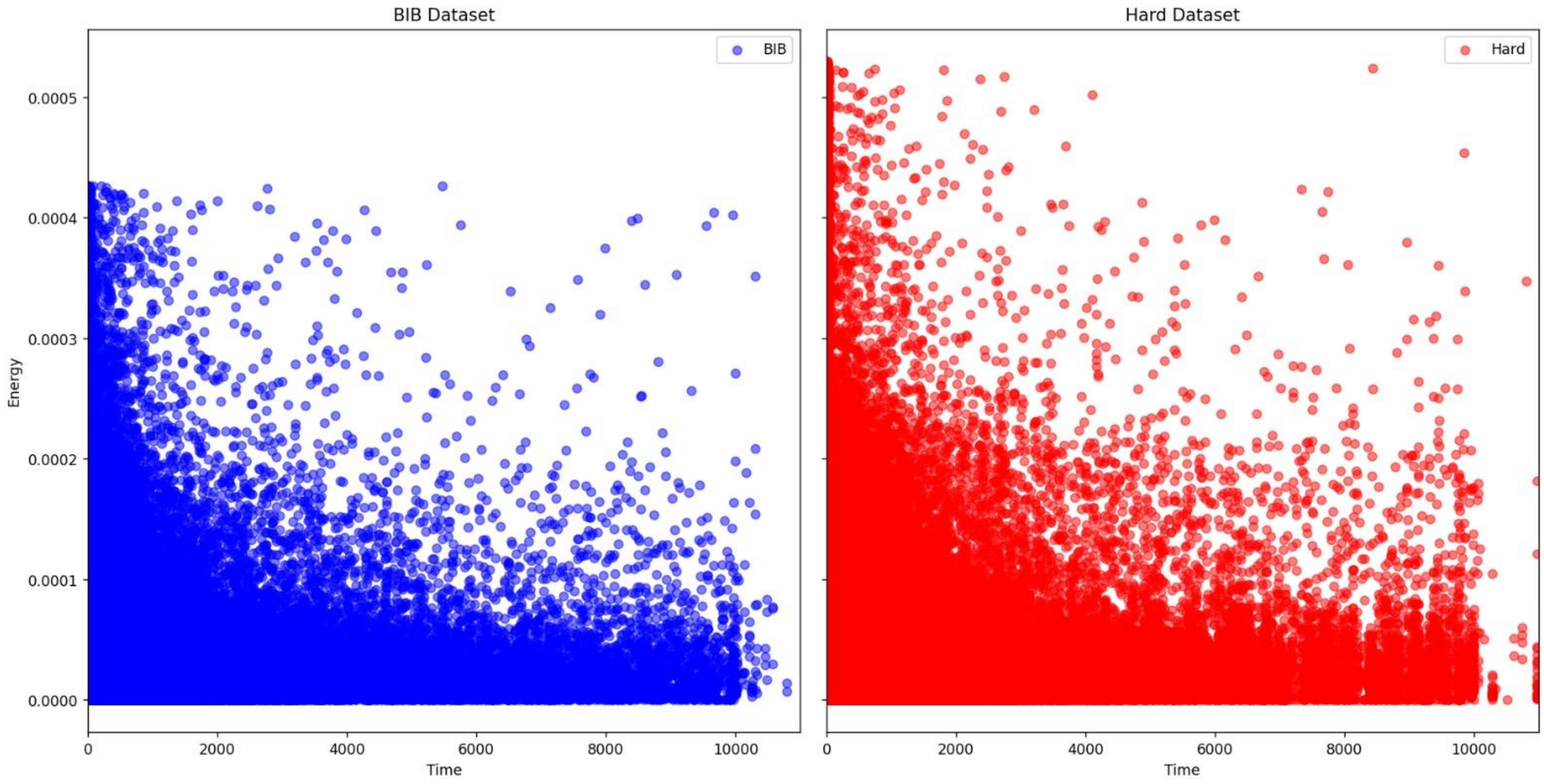
BIB vs Hard Dataset

- What was expected:
  - Some difference in energy distribution and time distribution as hinted in literature.
  - Other variables were X,Y,Z coordinates and PDG ID

BIB vs Hard Dataset



BIB vs Hard Dataset



Primary PDG ID is highly correlated with deciding Hard/BIB

- PDG ID gives the type of particle.

R-HADRONS	
$R^0_{\tilde{g}g}$	1000993
$R^0_{\tilde{g}d\overline{d}}$	1009113
$R^+_{\tilde{g}u\overline{d}}$	1009213
$R^0_{\tilde{g}u\overline{u}}$	1009223
$R^0_{\tilde{g}d\overline{s}}$	1009313
$R^+_{\tilde{g}u\overline{s}}$	1009323
$R^0_{\tilde{g}s\overline{s}}$	1009333
$R^-_{\tilde{g}ddd}$	1091114
$R^0_{\tilde{g}udd}$	1092114
$R^+_{\tilde{g}uud}$	1092214
$R^{++}_{\tilde{g}uuu}$	1092224
$R^-_{\tilde{g}sdd}$	1093114
R^0 [illegible]	1093214

SUSY PARTICLES	
$\tilde{d}_L$	1000001
$\tilde{u}_L$	1000002
$\tilde{s}_L$	1000003
$\tilde{c}_L$	1000004
$\tilde{b}_1$	1000005[a]
$\tilde{t}_1$	1000006[a]
$\tilde{e}^-_L$	1000011
$\tilde{\nu}_{eL}$	1000012
$\tilde{\mu}^-_L$	1000013
$\tilde{\nu}_{\mu L}$	1000014
$\tilde{\tau}^-_1$	1000015[a]
$\tilde{\nu}_{\tau L}$	1000016
$\tilde{d}_R$	2000001
$\tilde{u}_R$	2000002
$\tilde{s}_R$	2000003

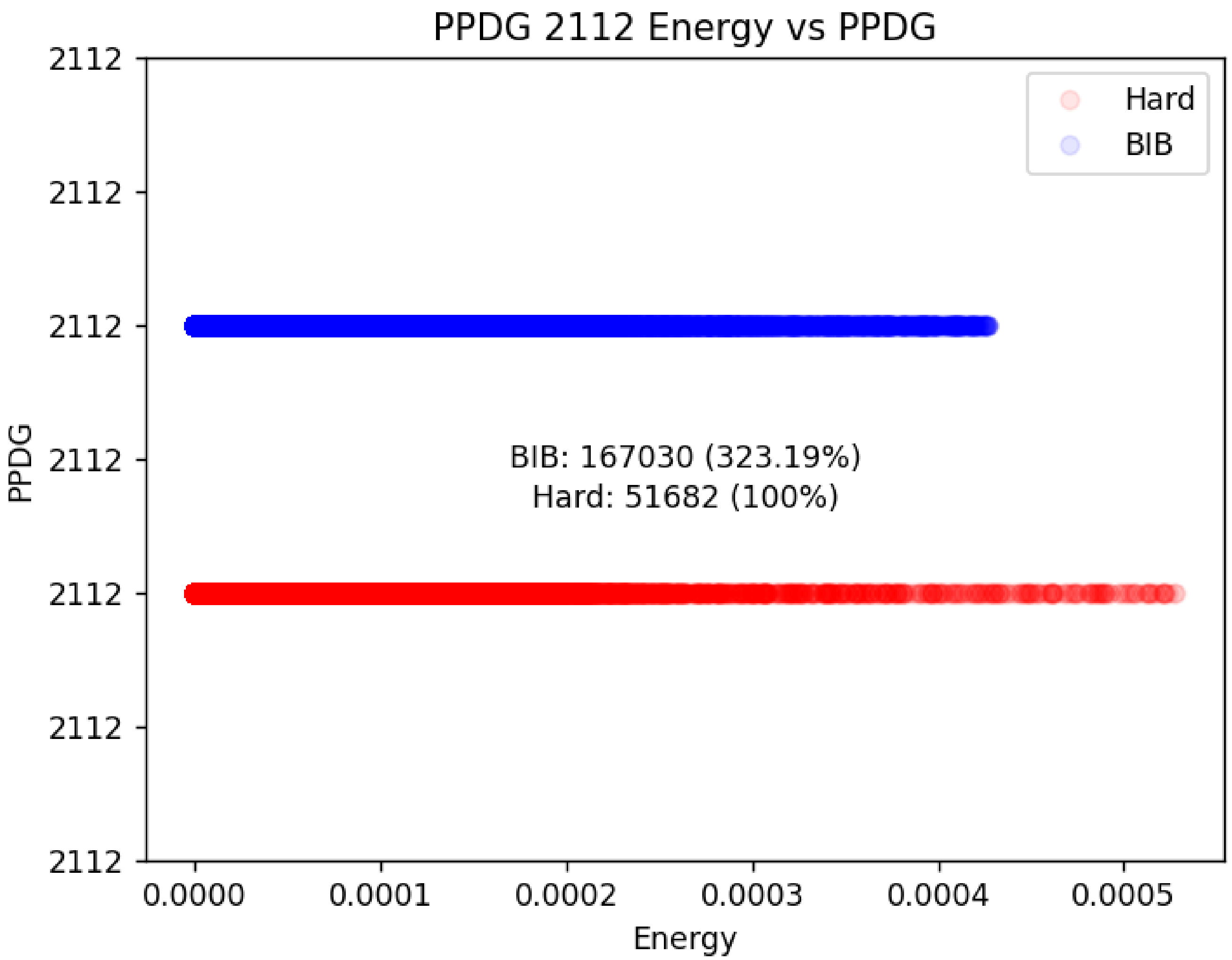
LIGHT $I = 1$ MESONS	
π^0	111
π^+	211
$a_0(980)^0$	9000111
$a_0(980)^+$	9000211
$\pi(1300)^0$	100111
$\pi(1300)^+$	100211
$a_0(1450)^0$	10111
$a_0(1450)^+$	10211
$\pi(1800)^0$	9010111
$\pi(1800)^+$	9010211
$\rho(770)^0$	113
$\rho(770)^+$	213
$b_1(1235)^0$	10113
$b_1(1235)^+$	10213
$a_1(1260)^0$	20113
$a_1(1260)^+$	20213

LIGHT $I = 0$ MESONS ($u\overline{u}$, $d\overline{d}$, and $s\overline{s}$ Admixtures)	
η	221
$\eta'(958)$	331
$f_0(600)$	9000221
$f_0(980)$	9010221
$\eta(1295)$	100221
$f_0(1370)$	10221
$\eta(1405)$	9020221
$\eta(1475)$	100331
$f_0(1500)$	9030221
$f_0(1710)$	10331
$\eta(1760)$	**9040221***
$f_0(2020)$	**9050221***
$f_0(2100)$	**9060221***
$f_0(2200)$	**9070221***
[illegible]	[illegible]

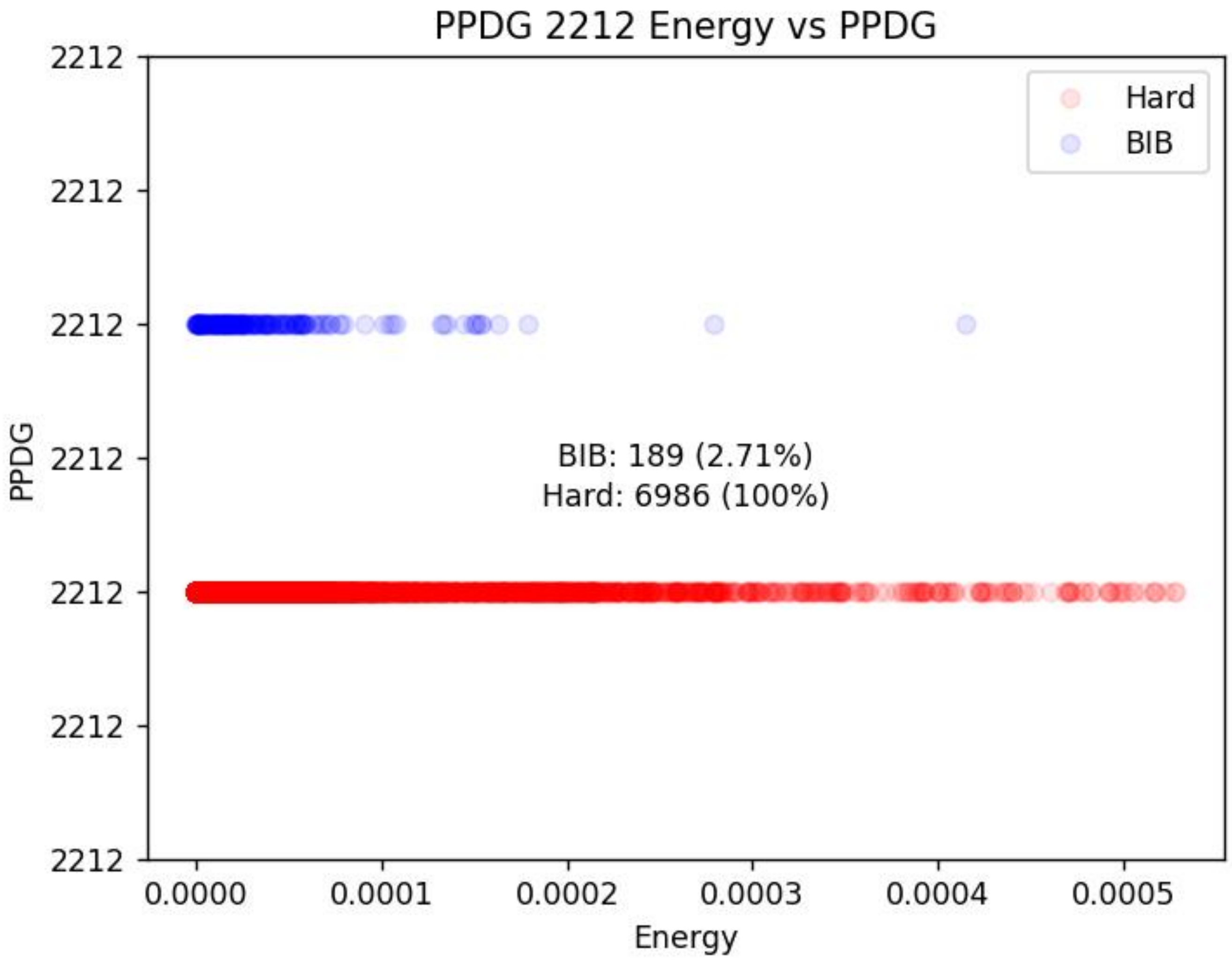
Monte Carlo particle numbering scheme (Rev.)

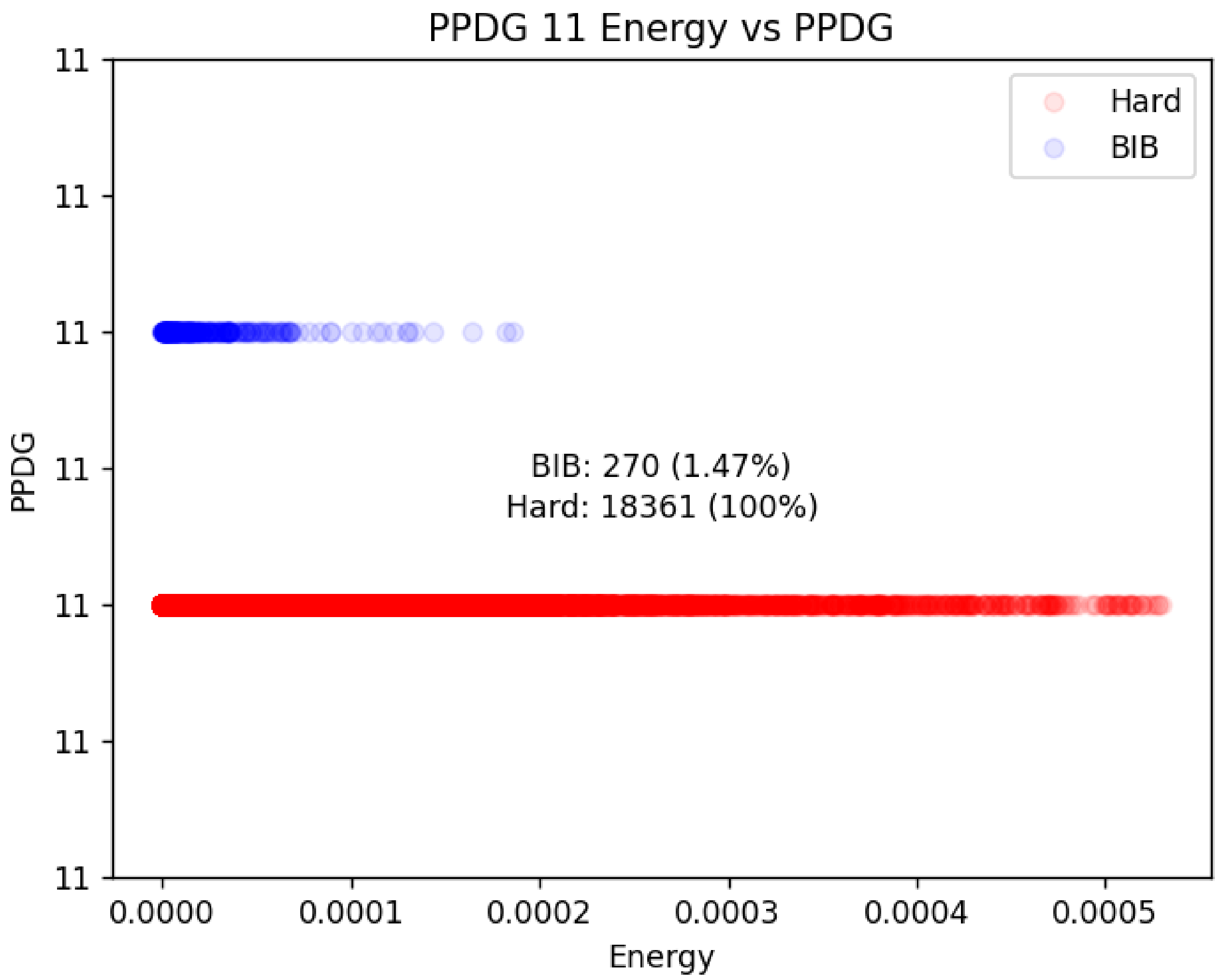
L. Garren (Fermilab), I.G. Knowles (Edinburgh U.), T. Sjostrand (Lund U.), T.G. Trippe (LBL, Berkeley)

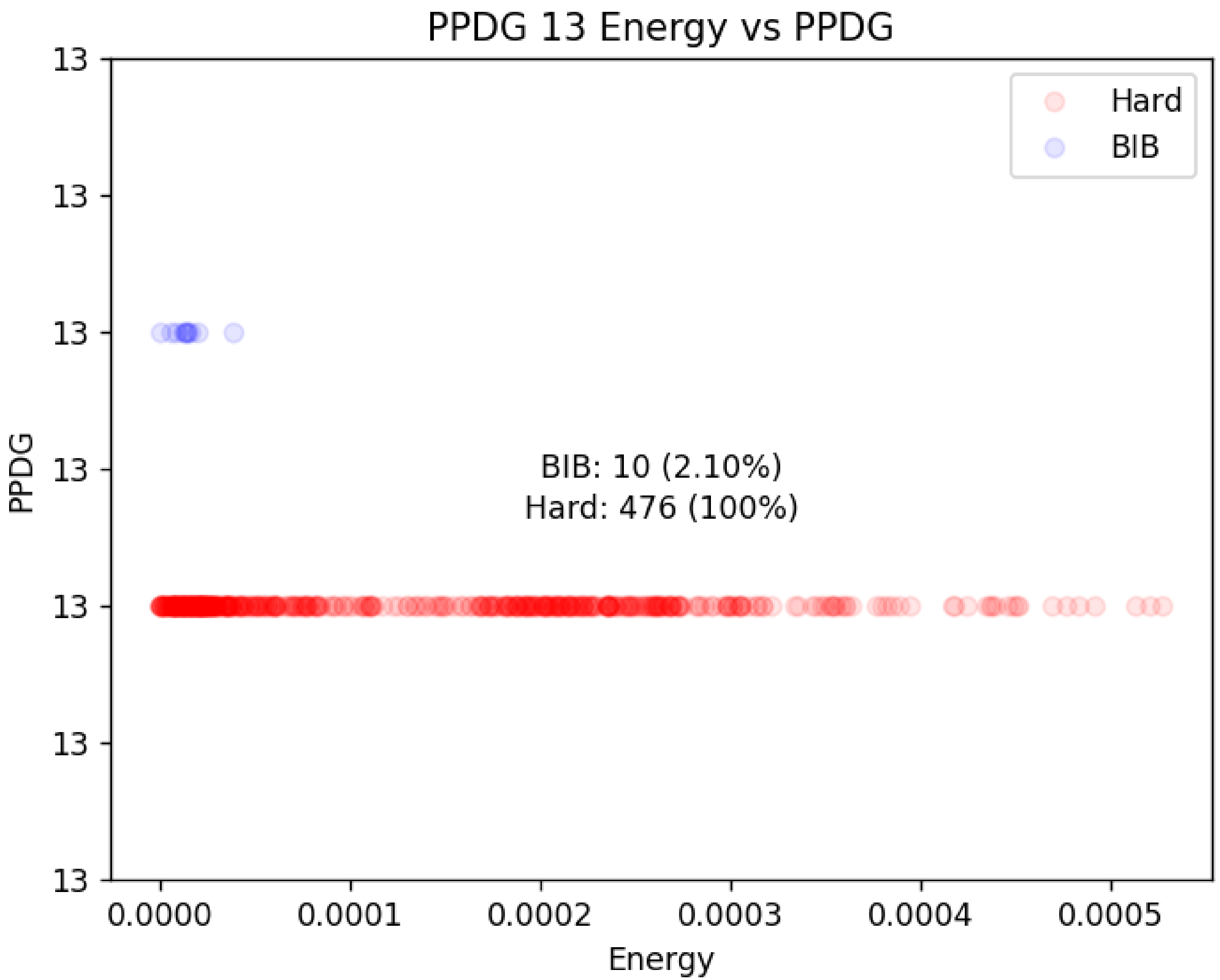
Same particle, in Both BIB and Hard how the energy composition is…

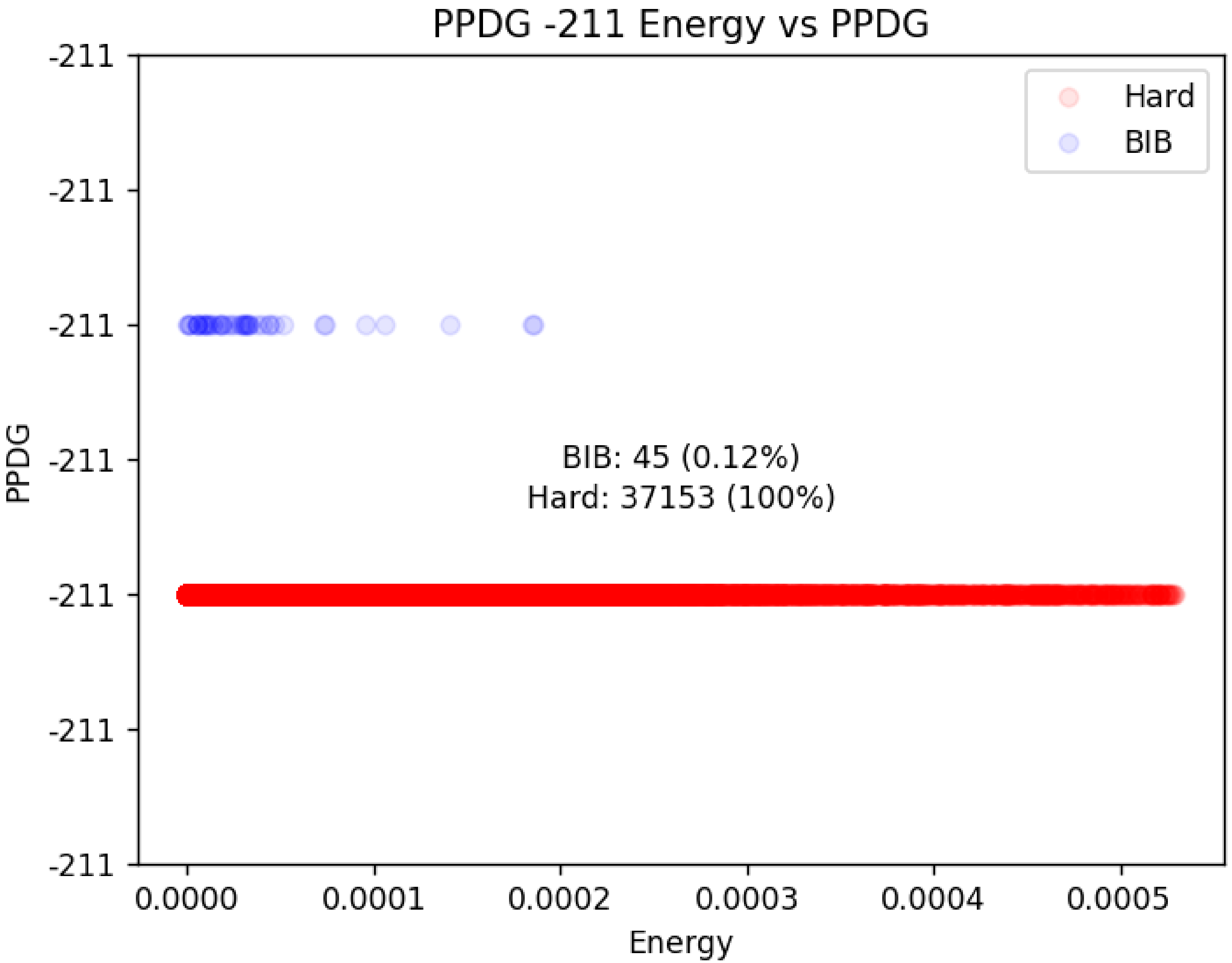


Y axis is the same, just shifted to give some offset otherwise the dots will overlap since same particle ID

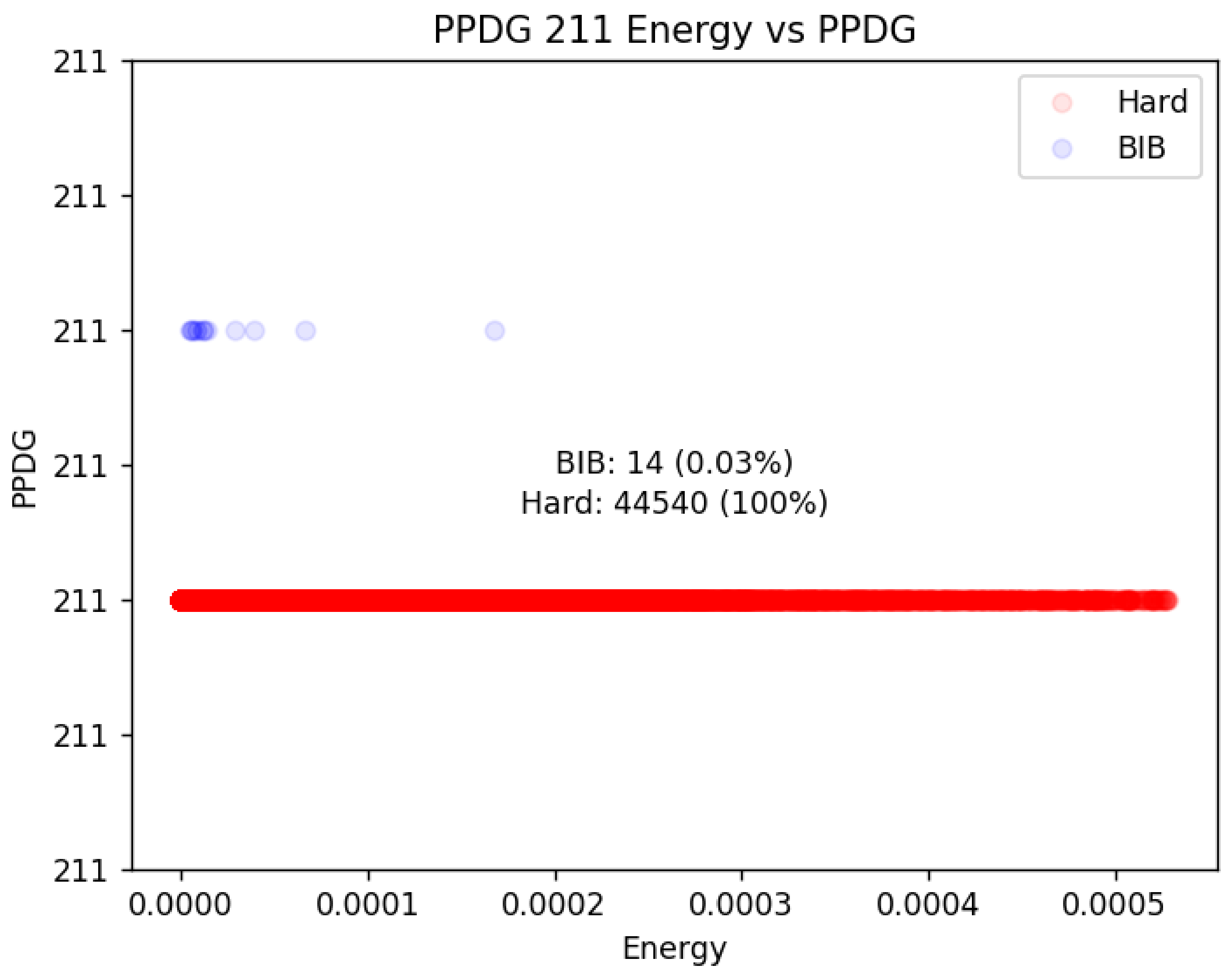

PPDG 2212 Energy vs PPDG
Hard
BIB
BIB: 189 (2.71%)
Hard: 6986 (100%)
PPDG
2212
2212
2212
2212
2212
2212
2212
Energy
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005

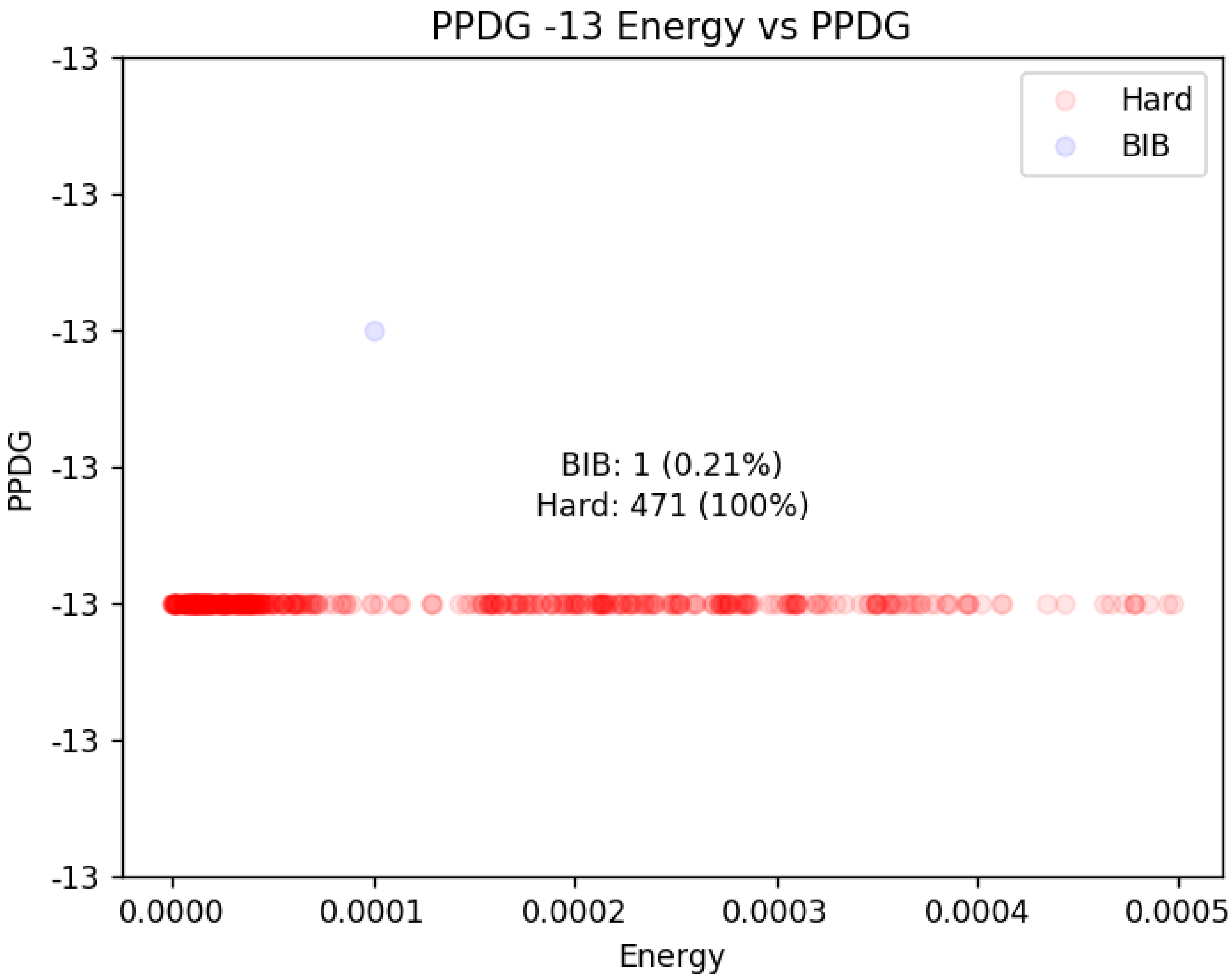

PPDG 11 Energy vs PPDG
Hard
BIB
BIB: 270 (1.47%)
Hard: 18361 (100%)
PPDG
11
11
11
11
11
11
11
Energy
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005

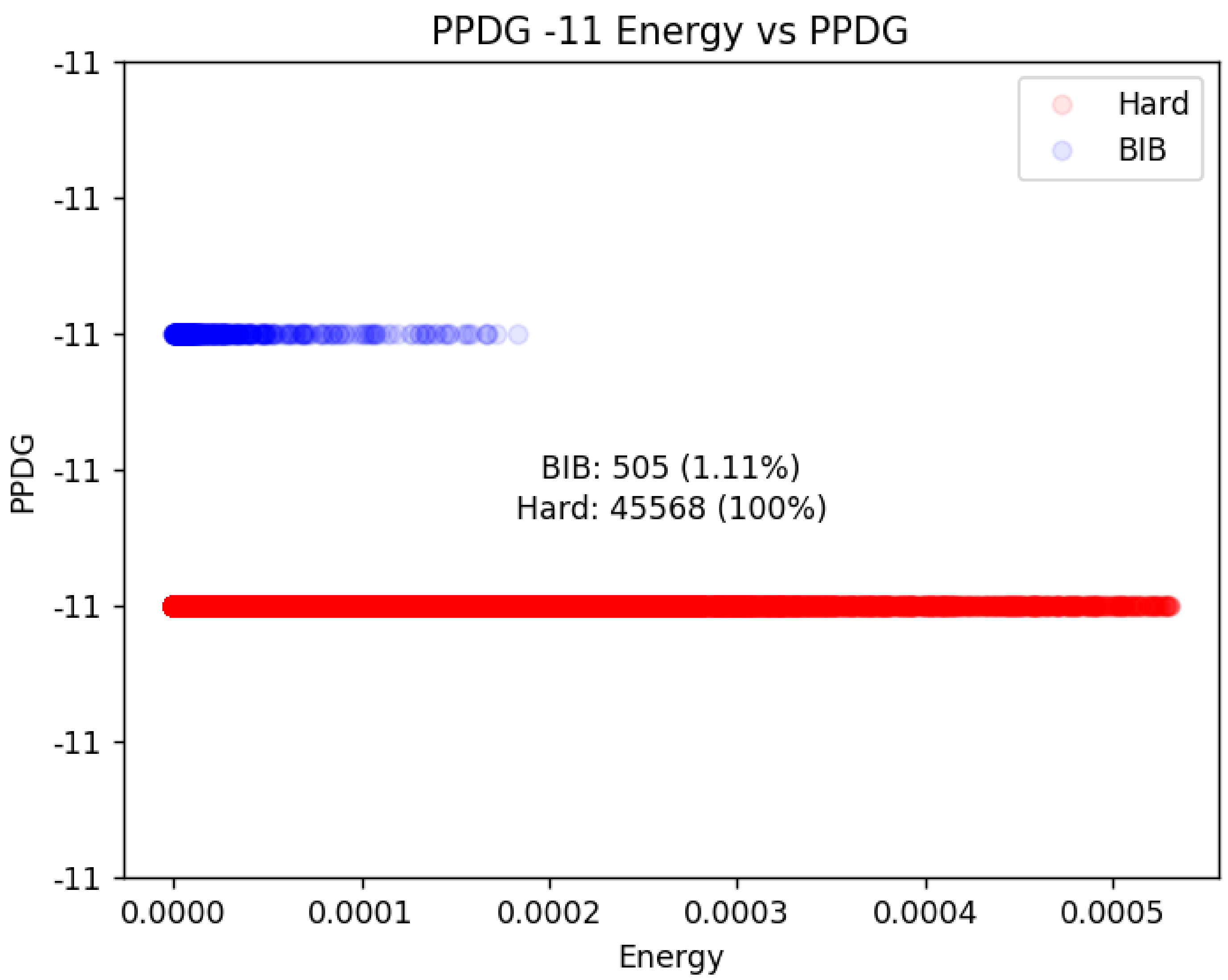

PPDG 13 Energy vs PPDG
Hard
BIB
BIB: 10 (2.10%)
Hard: 476 (100%)
PPDG
13
13
13
13
13
13
13
Energy
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005



PPDG -211 Energy vs PPDG
-211
-211
-211
-211
-211
-211
-211
PPDG
Hard
BIB
BIB: 45 (0.12%)
Hard: 37153 (100%)
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
Energy


PPDG 211 Energy vs PPDG
Hard
BIB
BIB: 14 (0.03%)
Hard: 44540 (100%)
211
211
211
211
211
211
211
PPDG
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
Energy


PPDG -13 Energy vs PPDG
Hard
BIB
BIB: 1 (0.21%)
Hard: 471 (100%)
PPDG
Energy
-13
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005


PPDG -11 Energy vs PPDG
Hard
BIB
BIB: 505 (1.11%)
Hard: 45568 (100%)
PPDG
-11
-11
-11
-11
-11
-11
-11
0.0000
0.0001
0.0002
0.0003
0.0004
0.0005
Energy

Primary PDG ID is highly correlated with deciding Hard/BIB

- Was able to get >90% model accuracy with PDG ID included in features

- But this cannot be used as a feature for training the model because this information will not be available in real-time !

- Excluding this PDG ID and training model resulted in 50% accuracy, which is same as random guess.

BIB vs Hard Dataset

- What was expected:
  - Some difference in energy distribution and time distribution as reported in literature.
  - Other variables were X,Y,Z coordinates and PDG ID

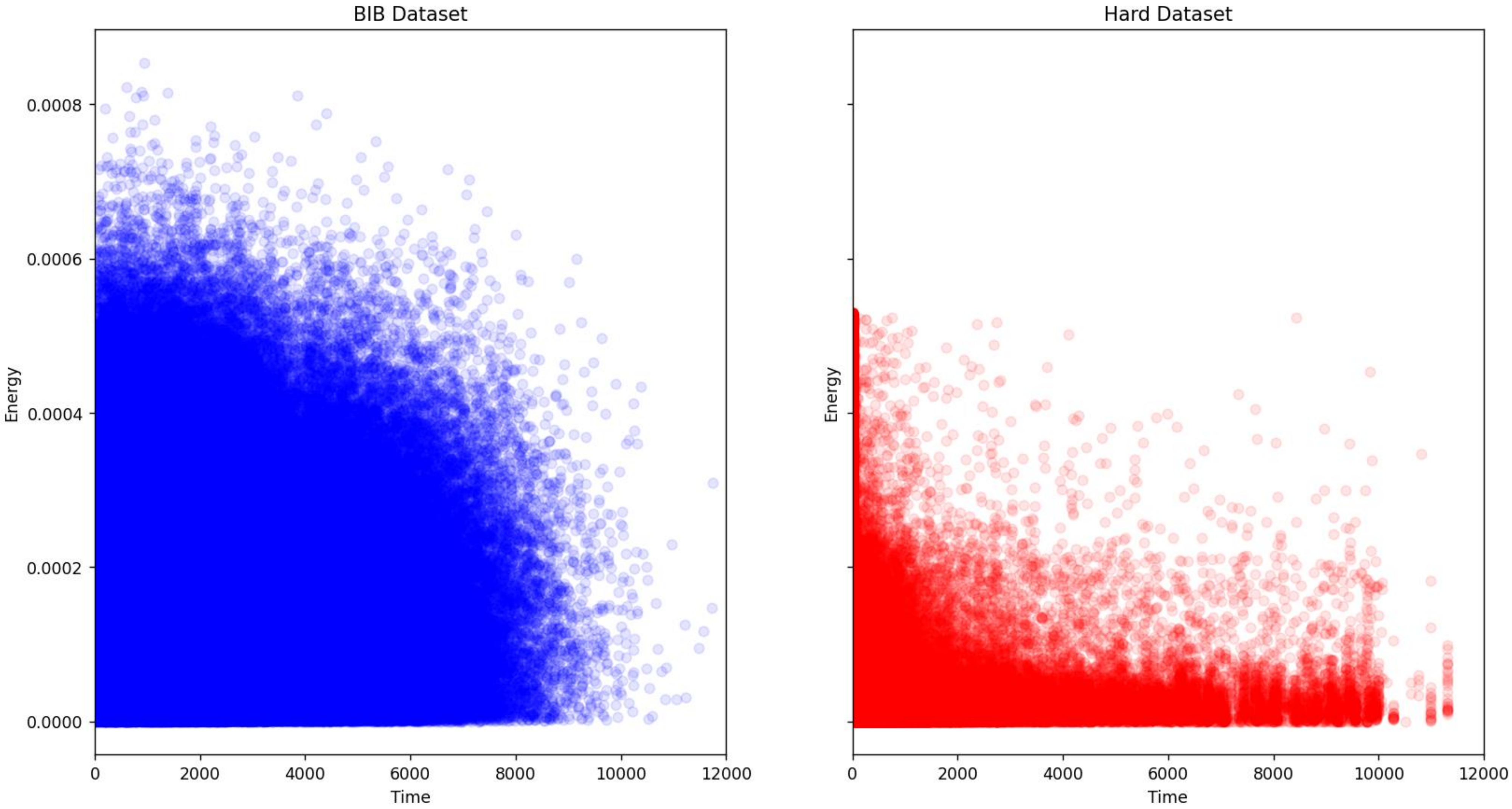
  - BIB had no distinguishing characteristics in Energy or Arrival time distribution as expected.

Generating a Synthetic BIB Dataset

Synthetic BIB vs Hard Dataset

- BIB had no distinguishing characteristics in Energy or Arrival time distribution as expected.
- So a synthetic database was generated for BIB with some noise-like characteristics.

Synthetic BIB vs Hard Dataset



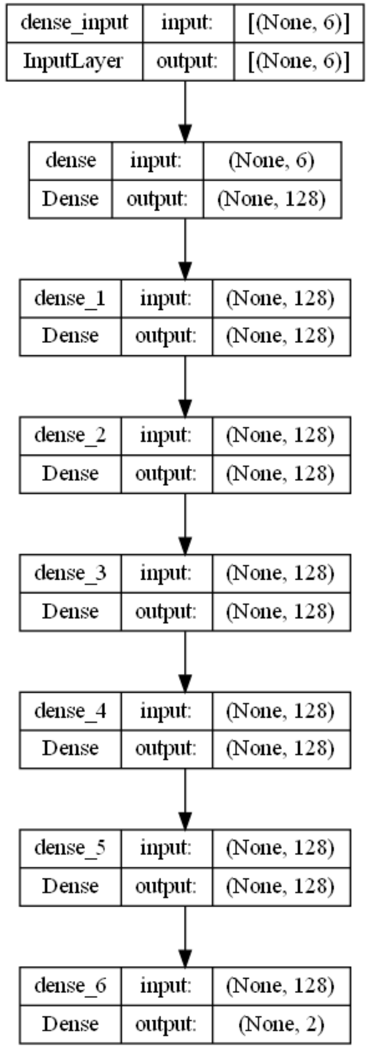
Finding best NN architecture

Finding best NN architecture

- FF neural networks were used for simplicity.
- Grid search technique was used to find the best architecture.
- Iteratively trains different configurations from the search space and finds the most accurate one.

```
# Define the search space for different architectures
# search_space = [
#     {'hidden_layers': [6],
#      'units': [16,32, 64, 128],
#      'activation': ['relu', 'tanh', 'sigmoid','softmax']}
# ]
```

Finding best NN architecture



- Using 500k rows with 50-50 train-test split, >98% accuracy was obtained at classifying BIB.

```
Accuracy on the entire dataset: 0.98913375
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 128)               896

 dense_1 (Dense)             (None, 128)               16512

 dense_2 (Dense)             (None, 128)               16512

 dense_3 (Dense)             (None, 128)               16512

 dense_4 (Dense)             (None, 128)               16512

 dense_5 (Dense)             (None, 128)               16512

 dense_6 (Dense)             (None, 2)                 258

=================================================================
Total params: 83,714
Trainable params: 83,714
Non-trainable params: 0
```

hls4ml → HLS Project

Performance Estimates

⊟ Timing

⊟ Summary

Clock	Target	Estimated	Uncertainty
ap_clk	10.00 ns	6.673 ns	3.00 ns

⊟ Latency

⊟ Summary

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
15	15	0.150 us	0.150 us	1	1	function

⊟ Detail

⊞ Instance

⊞ Loop

hls4ml → HLS Project

Utilization Estimates

Summary

Name	BRAM_18K	DSP48E	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	6	-
FIFO	-	-	-	-	-
Instance	98	4169	49	174346	-
Memory	-	-	-	-	-
Multiplexer	-	-	-	36	-
Register	-	-	3224	-	-
Total	98	4169	3273	174388	0
Available	5376	12288	3456000	1728000	1280
Available SLR	1344	3072	864000	432000	320
Utilization (%)	1	33	~0	10	0
Utilization SLR (%)	7	135	~0	40	0

hls4ml → HLS Project

Resource Usage

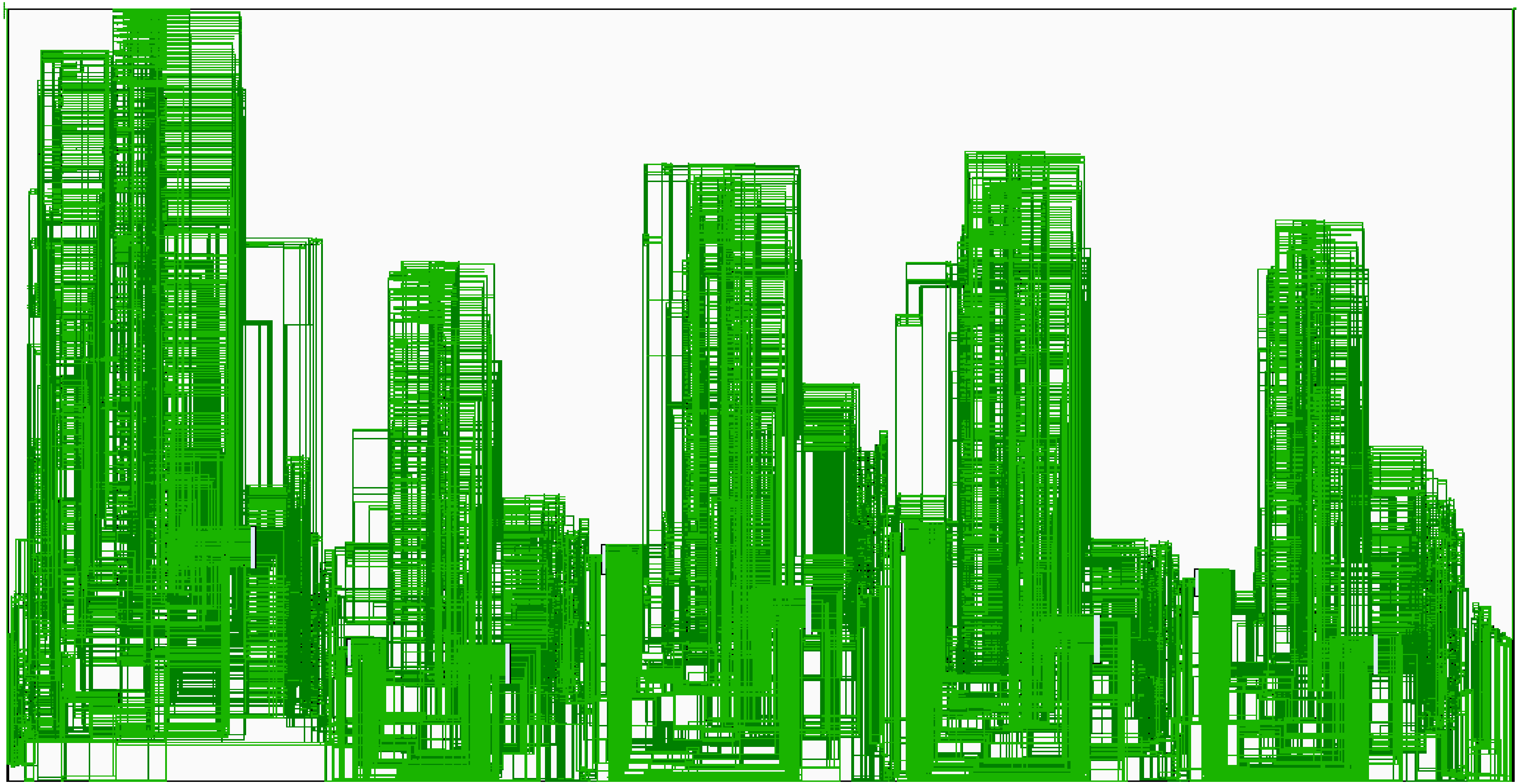
	Verilog
CLB	11157
LUT	62400
FF	3224
DSP	4161
BRAM	99
SRL	0
URAM	0

Final Timing

	Verilog
CP required	10.000
CP achieved post-synthesis	6.201
CP achieved post-implementation	9.933

Timing met

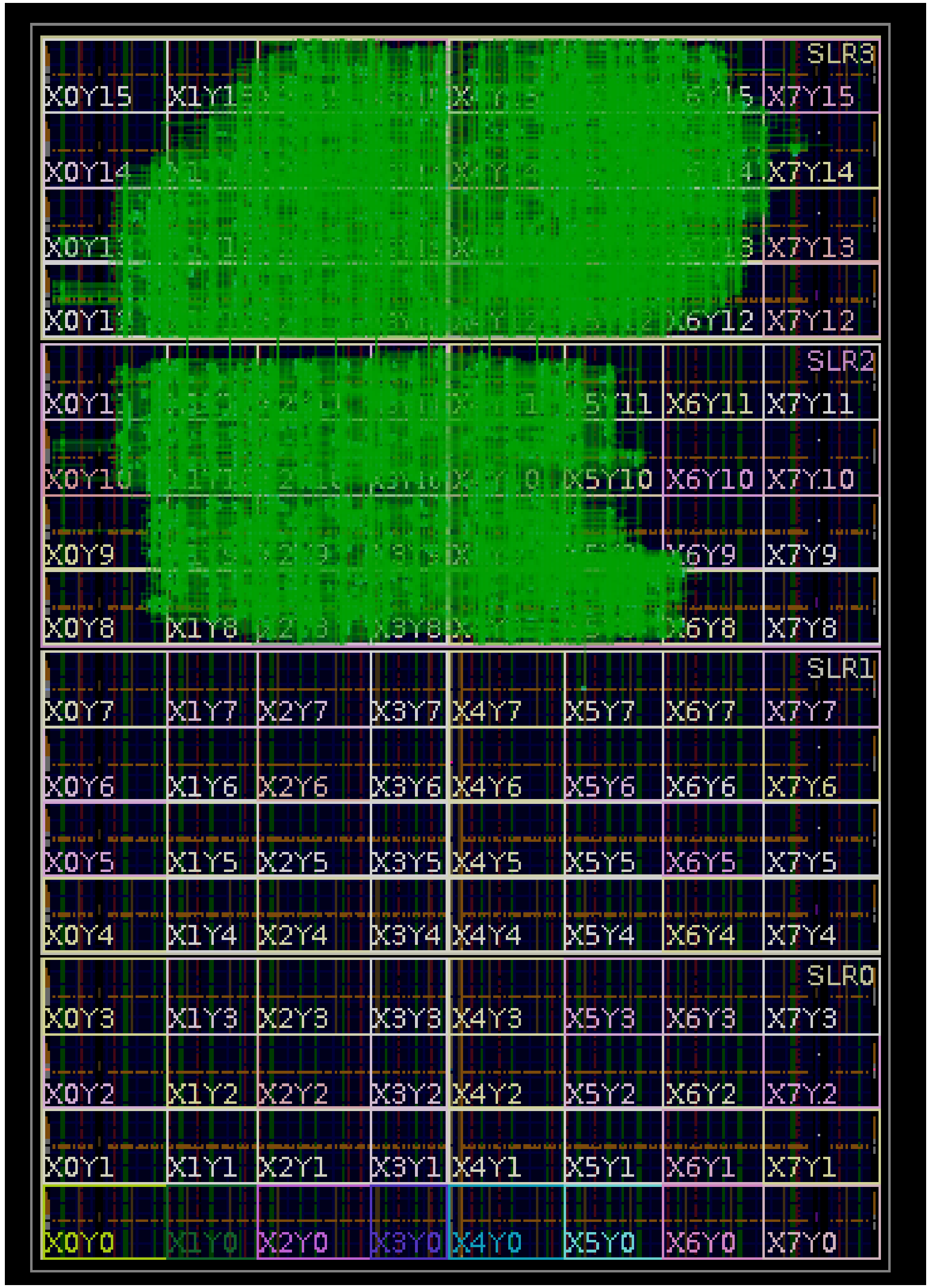
HLS Project → Vivado



HLS Project → Vivado

Part: xcu250-figd2104-2L-e

ULTRASCALE+ FPGA



HLS Project → Vivado

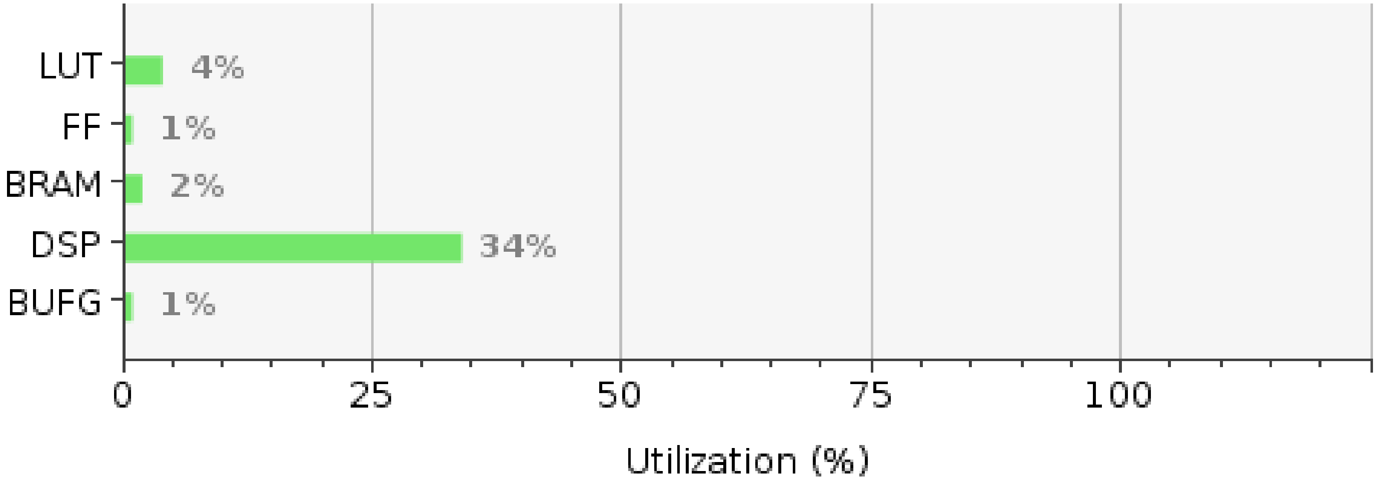
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.067 ns	Worst Hold Slack (WHS):	0.053 ns	Worst Pulse Width Slack (WPWS):	4.458 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	8427	Total Number of Endpoints:	8427	Total Number of Endpoints:	3419

All user specified timing constraints are met.

HLS Project → Vivado

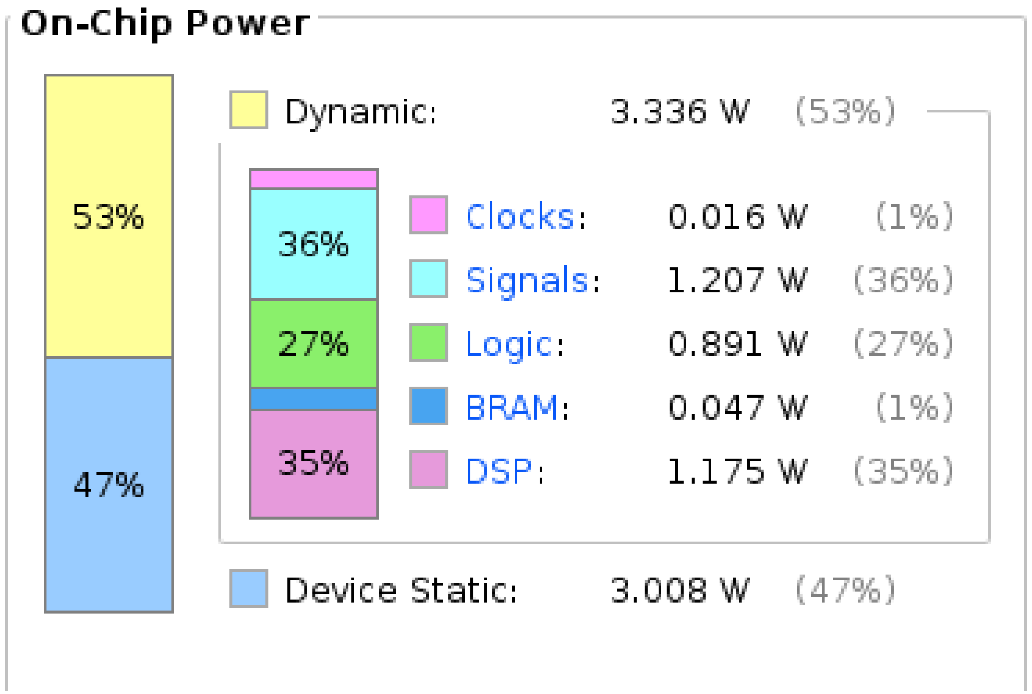
Resource	Utilization	Available	Utilization %
LUT	62400	1728000	3.61
FF	3224	3456000	0.09
BRAM	49.50	2688	1.84
DSP	4161	12288	33.86
BUFG	1	1344	0.07



HLS Project → Vivado

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**6.344 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**28.3°C**
Thermal Margin:	71.7°C (124.6 W)
Effective ϑJA:	0.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium



Key Findings

1. Training deep learning models and their hardware implementation via hls4ml, starting with GEANT4 simulation data was demonstrated.

2. It was seen that no trivial relationships exist between the Energy, arrival time and position data between the BIB and Hard data for the ECalBarrelCollection. A model trained directly on this data was shown to not work.

3. It was seen that there is a clear difference in the datasets in terms of particle composition although this data is not available in real-time, it goes to show that the BIB and Hard datasets are not identical.

4. In the case that there are distinguishing characteristics between the datasets, as in the case with synthetic BIB vs Hard, it was shown that the trained NNs could predict the data with a very high degree of accuracy and with <10ns latency for inference.

Future Directions

1. Bring out the distinguishing characteristics between Hard and BIB dataset using some mathematical techniques.

2. Test the other 12 collections to see if there are distinguishing features between Hard and BIB.

4. Try this methodology with other GEANT4 simulations.

5. Develop a standard python package to convert or parse .slcio files into .csv format and integrate it into the LCIO python package.

6. Restructure the data to be compatible for other deep-learning architectures such as CNNs, RNNs etc. and implement those.

References

[1] C. Accettura and et al., “Towards a muon collider,” *arXiv preprint arXiv:2303.08533*, pp. 1–119, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2303.08533

[2] F. Fahim and et al., “hls4ml: An open-source codesign workflow to empower scientific low-power machine learning devices,” *arXiv preprint arXiv:2103.05579*, pp. 1–10, 2021. [Online]. Available: https://arxiv.org/abs/2103.05579

[3] O. Calin, *Deep Learning Architectures: A Mathematical Approach*, ser. Springer Series in the Data Sciences. Springer Nature Switzerland AG, 2020.

[4] M. f. t. C. C. Lorusso, “Implementing machine learning inference on fpgas from software to hardware using hls4ml,” The Compact Muon Solenoid Experiment, CMS CERN, CH-1211 GENEVA 23, Switzerland, Conference Report CMS CR-2023/019, February 2023, v2, 22 March 2023.

[5] T. Aarrestad and et al., “Fast convolutional neural networks on fpgas with hls4ml,” *Machine Learning: Science and Technology*, vol. 2, no. 4, p. 045015, 2021.

References

[6] F. Barbosa, L. Belfore, N. Branson, C. Dickover, C. Fanelli, D. Furletov, S. Furletov, L. Jokhovets, D. Lawrence, and D. Romanov, "Development of ml fpga filter for particle identification and tracking in real time," *IEEE Transactions on Nuclear Science*, vol. 70, no. 6, June 2023.

[7] A. Sznajder, "Nanosecond jet classification at lhc," *Journal of Physics: Conference Series*, vol. 2438, no. 1, p. 012047, 2023.

[8] B. Ramhorst, G. A. Constantinides, and V. Loncar, "Fpga resource-aware structured pruning for real-time neural networks," *arXiv preprint arXiv:2308.05170*, August 2023.