

Data-based simulation of cosmic muons (not only) for calibration

Technical Note

Róbert Králik

University of Sussex

August 18, 2023

Abstract

In NOvA we employ samples of stopping and through-going cosmic muons to calibrate our detectors. To validate the efficacy of this calibration procedure for both the simulated and real detectors, as well as to determine systematic uncertainties arising from calibration, we require a sample of simulated cosmic muons.

Originally we used (or still use) the Cosmic-Ray Shower Generator (CRY) [?] to create a large Monte-Carlo (MC) cosmic ray sample. However, the CRY simulation proved to be highly inefficient, with only a small fraction of the simulated cosmic-ray activity resulting in selected calibration hits, and the majority of particles failing to even hit the detector. This inefficiency consumed significant CPU resources, disk space, and file usage. Moreover, the momentum and angle distributions in CRY were not well suited to the NOvA sites, potentially impacting the calibration accuracy.

To overcome these challenges, we have implemented a data-based simulation method that eliminates the need for the CRY MC generator. Instead, we use the data-driven activity trigger (DDActivity) data sample, which is passed through the beam removal filter, reconstruction chain and a selection of high-quality cosmic muons. The selected cosmic ray muon events are then being used in the "Text File Generator" simulation to create an equivalent cosmic ray sample.

This approach results in near-perfect efficiency, ensuring that almost every simulated muon contributes to the final calibration sample, thus saving processing time, file size, and storage. Additionally, the simulated muon distributions are inherently consistent with the data. Given that the calibration chain itself is a time and CPU intensive process, the reduction in simulation files and their sizes has significant benefits downstream of the file generation.

Contents

1	Introduction	2
2	How does it work?	2
2.1	Reconstruction and selection of cosmic muon events from data	4
2.1.1	Remove Beam Spills	4
2.1.2	Reconstruction	5
2.1.3	Selection	6
2.2	Energy correction, charge assignment and smearing	11
2.2.1	Energy correction	11
2.2.2	Smearing	11
2.2.3	Charge assignment	13
2.2.4	Number of events to simulate	13
2.2.5	Print the outputs	13
2.3	Submitting the simulation jobs	15
3	Validation	16
4	Conclusions	17

1 Introduction

The data-based simulation of cosmic muons was initially developed by Teresa Lackey [?] in 2021 for Test Beam detector calibration. Teresa based the reconstruction and selection of data events on the `CerenkovSelection` module from light level tuning and created a simple Python script for event smearing and muon charge assignment. However, when tested by Robert Kralik in 2021 and 2022 [?], the simulation exhibited notable discrepancies when compared to the Test Beam data from period 2.

To address these disparities, Robert made improvements to the simulation throughout 2022, ultimately completing it in 2023. The enhancements involved modifications to event selection, charge assignment and energy correction of the through-going muons. The revised simulation was then employed and tested during the calibration of the Test Beam detector in 2023.

This technical note provides an explanation of the process to create simulation samples for calibration, with a primary focus on its application in Test Beam. However, the approach and underlying code have been designed to be easily adaptable for use for the Near and Far detectors, as well as to generate simulated samples of cosmic muons for purposes other than calibration.

2 How does it work?

All the code required to generate data-based simulations of cosmic muons is located within the `novasoft CosmicStudies` package.

63 The process of generating a new data-based simulation begins with a data sample containing
64 information on Raw Digits hits. For Test Beam we use the artdaq-stage DDActivity samples, but
65 the pid-stage samples should also contain all the necessary information. Pre-staging this sample
66 can be the most time-consuming part of the data-based simulation process, so it is advised to select
67 a data sample that is either already cached or that can be easily pre-staged. Section 2.2.4 provides
68 insights into estimating the required data volume.

69 It is necessary to choose a data sample that represents the detector in a fault-free state. For
70 Test Beam, we use the full period 4 data sample, as other periods had issues such as faulty FEBs,
71 underfilled cells, or similar complications. In the initial version of the data-based simulation,
72 Teresa used the period 2 Test Beam data, as it was the only complete Test Beam data sample
73 available. However, this data included the aforementioned effects, which could have had a non-
74 trivial impact on the simulation.

75 Once we have a selected data sample, we use the cosmicgenanajob ART job to apply a series
76 of filtering, reconstruction, and selection steps to obtain a ROOT TTree with vertex positions and
77 4-momenta of selected good quality cosmic muon events. This is outlined in section 2.1 of this
78 document.

79 Subsequently, the reconstructed information of each event is processed by a Python script
80 GenerateHEPEVTFromROOT.py. This script corrects the 4-momenta of the through-going muons
81 to account for the missing energy that was not deposited in the detector. Furthermore, it assigns a
82 charge to each cosmic muon based on a statistical distribution, smears the kinematic information
83 to reduce bias from the input data and prints the HEPEVT-styled [?] description of each event into a
84 text file. The details of this process are elaborated in section 2.2.

85 The text file is then passed to the Text File Generator, which employs the information as
86 a seed for a geant4-based [?] detector simulation. By incorporating additional simulation and re-
87 construction steps, an artdaq-stage simulation sample of cosmic muons is generated. To create
88 calibration samples for stopping and through-going cosmic muons, we apply the same reconstruc-
89 tion and selection criteria as for any data sample. Further details are provided in section 2.3.

90 A step-by-step summary is presented in the box below and can also be found on the [DataDriven](#)
91 [Cosmics Generation redmine wiki page](#).

1. Run `cosmicgenanajob.fcl` on detector activity `artdaq-stage` files;
2. Hadd the outputs and run
`GenerateHEPEVTFromROOT.py haddedOutput.root HEPEVTFileName.txt;`
3. Split the resulting txt file into files each containing a subset of lines (events);
4. Generate FHiCL files each sourcing a different text file
`CreateFclsForSimulation.sh TextFileGenjob_template.fcl
inDir outDir;`
5. Make a SAM definition from the FHiCL files with `sam_add_dataset`;
6. Include all the text files into `TextFileGen.cfg` with `-inputfile`;
7. Submit `TextFileGen.cfg`;
8. Move the newly created simulation `artdaq` files to a persistent area, declare them to SAM and create a definition;
9. Generate the calibration files (`pclist/pcliststop`);
10. Move the results into a persistent location, declare to SAM and make definitions.

2.1 Reconstruction and selection of cosmic muon events from data

Our goal is to examine the response of the **simulated detector** to realistic cosmic muons found in the **real data**. We therefore need to use well-reconstructed and selected cosmic muons from data to generate our simulation. If the selection of the reconstructed data does not accurately correspond to reality, either due to misreconstruction or incorrect selection criteria, it can introduce bias into our simulation.

Additionally, we want the simulation to primarily consist of events that will make it into the final simulation calibration sample and reject those that will not. Therefore, it is useful to employ a similar reconstruction and selection process to that used to create the data calibration samples. We also require the distributions of the selected events to be well-understood and to resemble those of the data calibration samples.

We use a single ART job located in `CosmicStudies/cosmicgenanajob.fcl` to filter, reconstruct, and select the desired events, which are then written to a ROOT TTree. The details of each step are described below.

2.1.1 Remove Beam Spills

The first step is to remove beam spill events using the `RemoveBeamSpills.fcl` job for the Near and Far Detectors, or `RemoveTBSpills.fcl` for the Test Beam detector. This is done based on the

110 event time relative to the time of the beam spill. For Test Beam the beam spill is 4.2 seconds long
 111 and we remove all events within a 5 seconds window from the start of the beam spill, as shown on
 112 figure 1. This should leave us with mostly cosmic events.

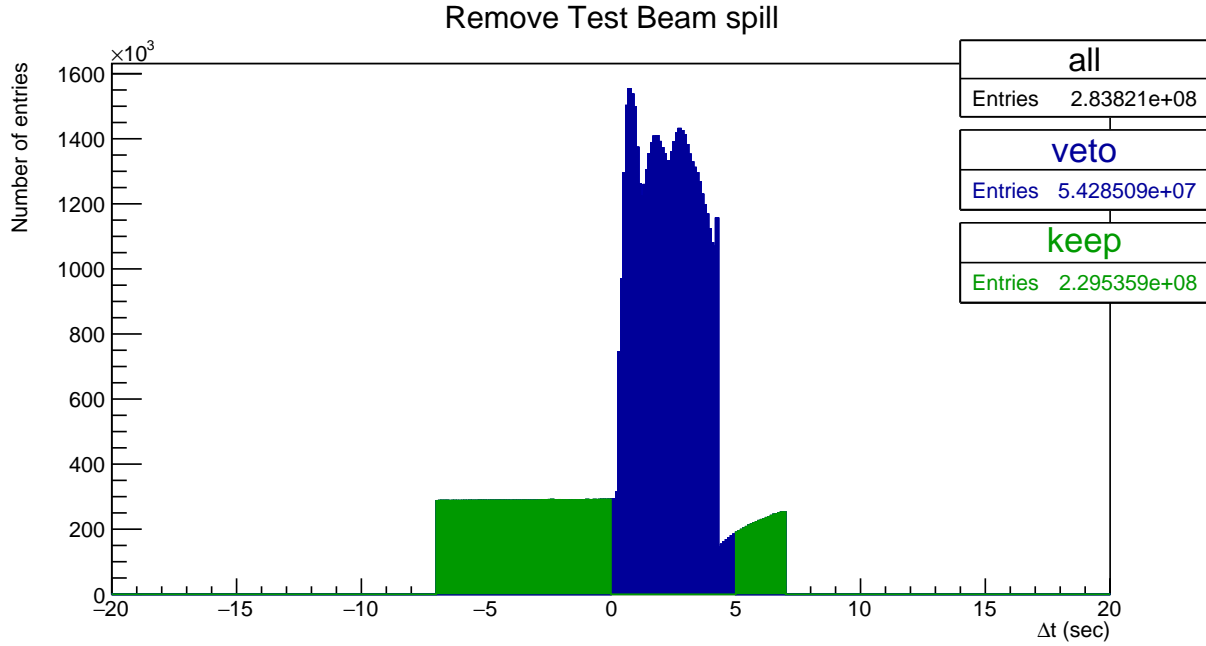


Figure 1: Test Beam beam spill events removed (blue) from the calibration samples. The remaining events (green) should mostly consist of cosmic particles. This example and the numbers of entries are for the full period 4 Test Beam sample.

113 2.1.2 Reconstruction

114 The Text File Generator requires the vertex position and the initial 4-momentum for each
 115 event. For that we use:

- 116 1. CalHit to create Calibrated Cell Hits from Raw Digits,
- 117 2. Slicer to group hits into slices,
- 118 3. Window CosmicTrack - a tracking algorithm for cosmic particles [?],
- 119 4. CosmicRayVertex to identify vertices for cosmic particles,
- 120 5. FuzzyKVertex to cluster hits into prongs and
- 121 6. BreakPointFitter (BPF) to identify muons (or muon-like tracks) and get their initial 4-
- 122 momenta by fitting to the FuzzyK prongs and CosmicRayVertex vertices [?].

The first three steps are identical to the full reconstruction applied to get the calibration samples. Since we do not need a 4-momentum information for calibration, we do not need to use Cosmic Ray Vertex, FuzzyK Vertex, or the Break Point Fitter to create calibration samples.

2.1.3 Selection

After the reconstruction process, we proceed to select events based on their slice and **BreakPoint-Fitter (BPF) track** properties, saving them in a Root TTree. This step is done using the CosmicStudies/CosmicGenAna module.

To select an event, the following conditions must be met (table 1 provides an overview of all cuts and their corresponding values):

1. We only use successfully reconstructed 3D tracks with the muon assumption from the BreakPointFitter.
2. As we aim to select cosmic events originating outside the detector, we apply a cut based on the distance of each track's start from the Top/Front/Back/Sides of the detector. This cut has a negligible impact on the BPF tracks, as indicated by the minimal difference between the red and the dotted azure lines in Figure 2;
3. We remove all events whose track is parallel to the beam direction. Figure 2 demonstrates the presence of events peaked at track lengths of approximately 410cm and 200 cm, which correspond to the total and half length of the detector (or alternatively lengths of both modules and a single module), respectively. These events are strictly parallel to the beam direction, with track's angle from the Z axis $\sim 0^\circ$ ($\text{Cos}_Z \sim 1$), and are likely remnants of beam events, or possibly from a Test Beam beam halo. Applying a cut on $|\text{Cos}_Z| > 0.98$ effectively removes these events without affecting the rest of the data. This cut might only be needed for Test Beam and not for the near and far detectors.
4. To ensure that only events contributing to the final calibration sample are simulated, we use a selection based on cuts from the Calibration/PCHitsList module, which is used for both data and simulation to create the calibration samples. Let's call these cuts the **calibration cuts**. However, there are two caveats we need to consider:
 - (a) First, to create calibration samples, we use tracks from the Window cosmic track algorithm instead of the Break Point Fitter algorithm, which yield different distributions as depicted in Figure 3. Applying the calibration cuts on BPF tracks could mistakenly remove events that would pass the same selection when applied to the window cosmic tracks.
 - (b) Second, each reconstruction algorithm has intrinsic deficiencies that can lead to misreconstructions. Applying the full calibration cuts may remove misreconstructed events that should have been included in the simulation, introducing a selection efficiency bias.

To address these concerns, we have loosened the full calibration cuts to create a "buffer" around the selected events, allowing for fluctuations of the reconstruction algorithms while maintaining track quality. The differences between the full calibration cuts and the employed loosened cuts are listed in table 1 and shown on Figure 3. Additionally, we present a comparison to the data calibration sample, which was created by applying the full PCHitsList selection on window cosmic tracks from the same artdaq data sample.

Cut		Full selection	Loose selection
Muon assumption and 3D track from BPF			
Max. track start distance from edge			50 cm
Max. Cos_Z			0.98
Calibration sample selection	Max. number of hits in X or Y		2
	Min. difference between Stop_Z and Start_Z	70 cm	50 cm
	Min. Cos_Z	0.2	0.15
	Min. frac. of slice hits in track in each view		0.8
	Max. number of cells per plane in each view	6	15
	Max. difference in X-Y for first (last) plane	3	5
	Max. plane asymmetry	0.1	0.2
	Max. step size to median step size ratio	3	5
	Max. vertex distance from edge		10 cm
	Max. track end distance from edge		10 cm

Table 1: Event selection for the data-based simulation (in green under Loose selection) and comparison to the Full calibration sample selection cuts in blue. The last two rows are not used for Test Beam, but are employed for the Near and Far detectors and should be studied before creating a data-based simulation for them.

During the selection process, we determine whether the muon is stopping inside the detector or passing through, based on the reconstructed track's end position¹. This information assists in correcting the energy of through-going muons to account for "lost energy," as outlined in section 2.2.

In the initial iteration of the simulation, Teresa implemented a cut requiring the total track length to be at least 300 cm. This was motivated by the Cerenkov selection for light level tuning but contributed significantly to the discrepancy between data and simulation in the calibration samples. To address this issue, Robert removed this cut and, to ensure the quality of selected events remained high, applied the loosened calibration sample selection cuts along with the Cos_Z cut.

¹For Test Beam we say it is a stopping muon if its track ends at least 20cm from any edge of the detector. For the far and near detector this is 50cm. This value was chosen by Kevin Mulder [?] as 50cm removed too many cosmic events from the Test Beam detector.

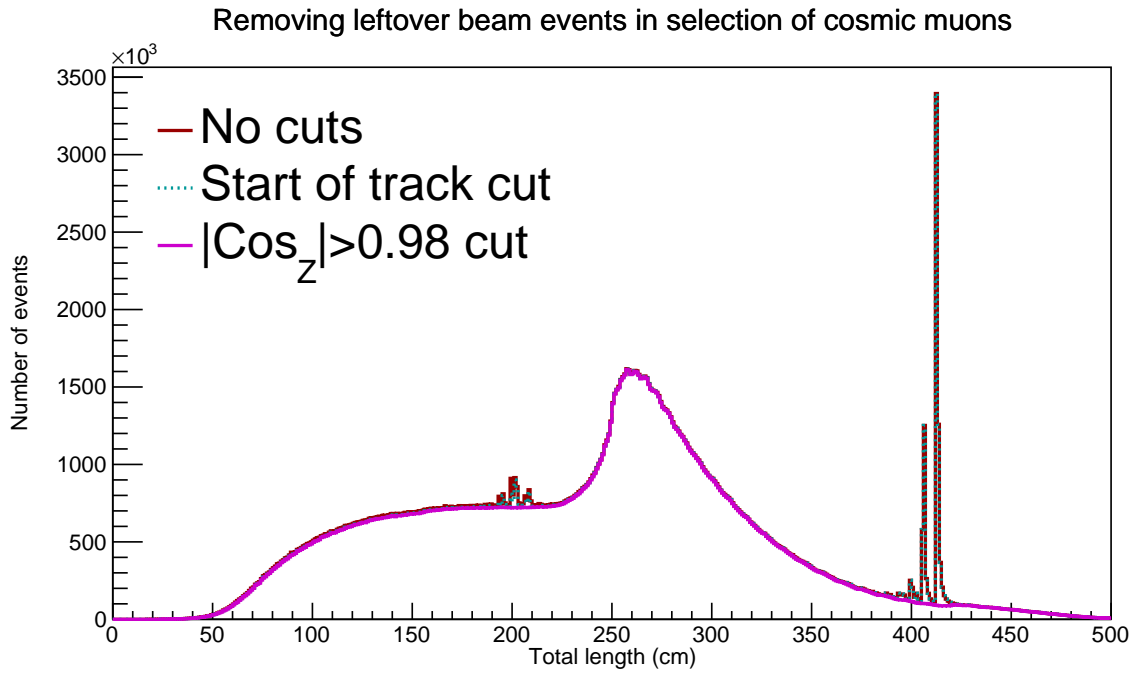
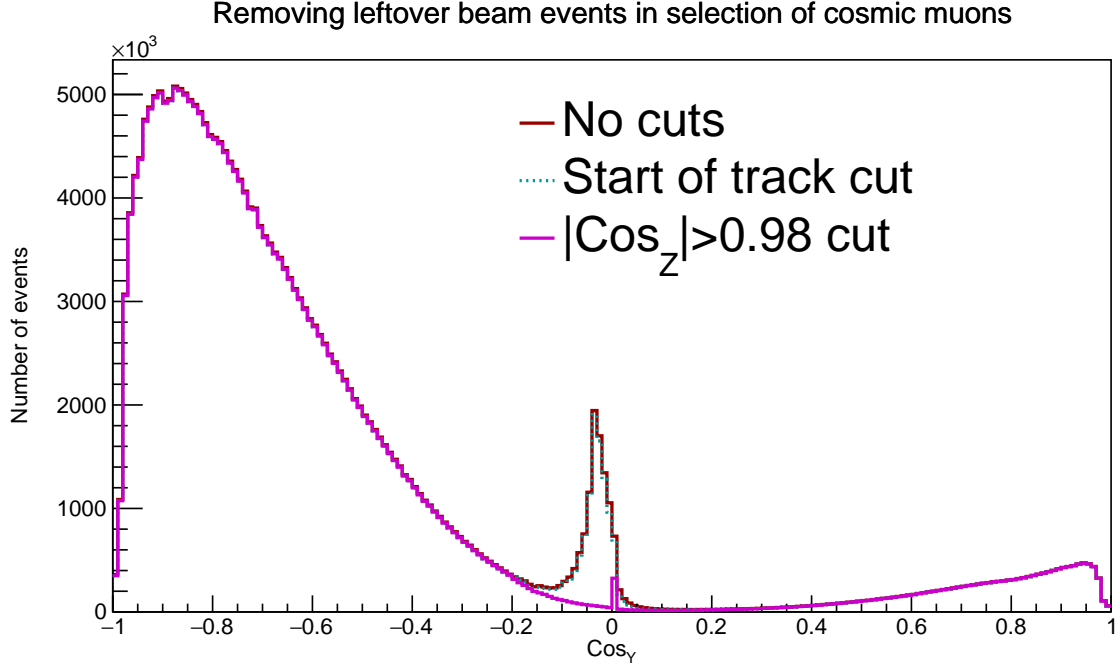


Figure 2: Impact of track start and maximum track angle from the z axis (Cos_Z) cuts on the Test Beam data for the data-based simulation of cosmic muons. The track start cut has only negligible effect. The maximum Cos_Z cut effectively removes sharp peaks in the total track length distribution and events perpendicular to the Y axis. These events are all parallel with the Z axes and are most likely leftover beam events. All of the distributions are made from the period 4 Test Beam data.

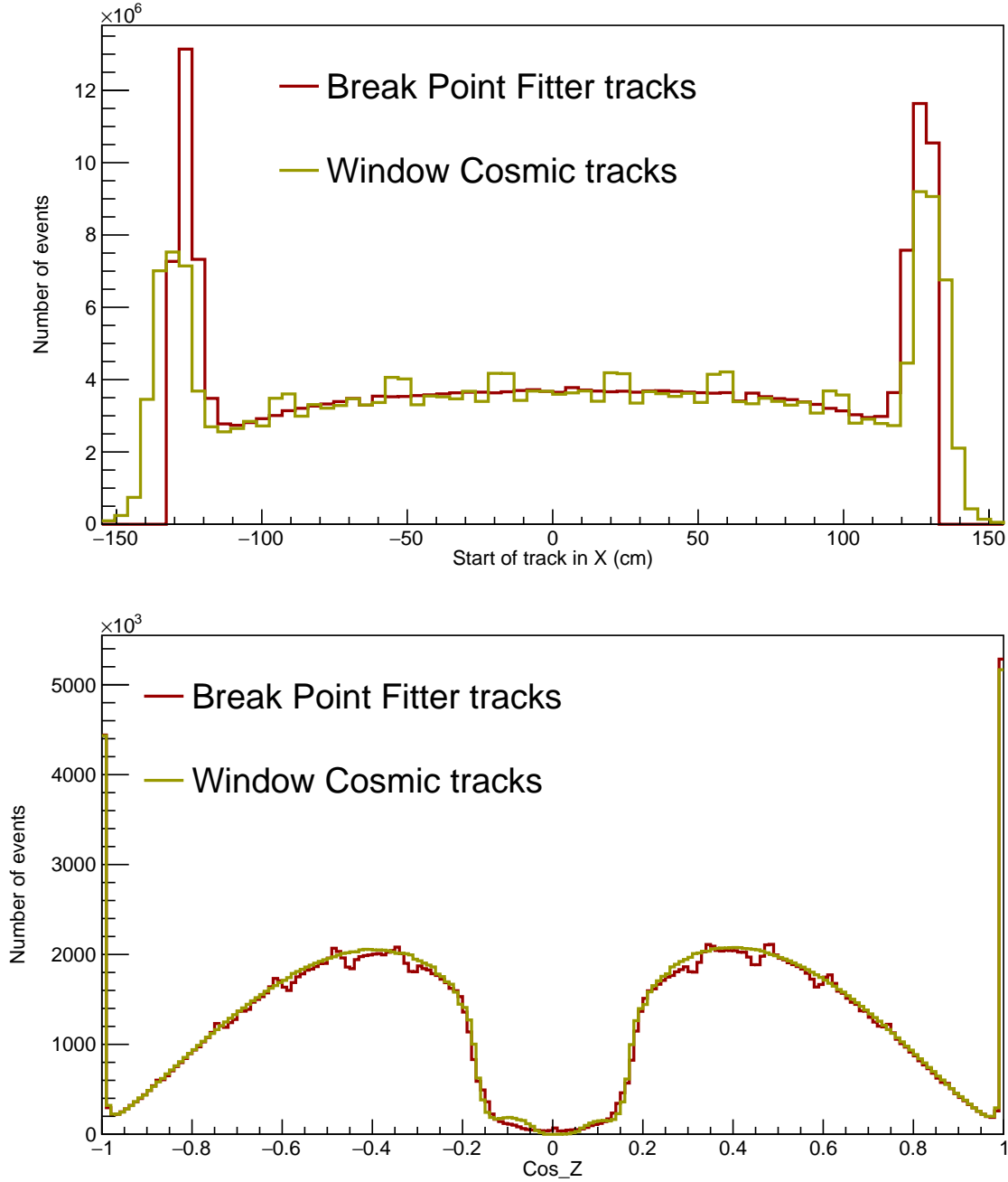


Figure 3: Difference between the tracks reconstructed with the break point fitter (BPF - used for data-based simulation) and with the window cosmictrack (WT - used for generating the calibration samples) algorithms. Both distributions are for the period 4 Test Beam data (with removed beam spill) without applying any selection. The BPF tracks have a hard cut-off at the detector edges, whereas the WT tracks are allowed to start beyond these limits. Also, the BPF tracks have a ragged distribution in Cos_Z , which is not present for WT tracks. It is not yet known where does this raggedness come from.

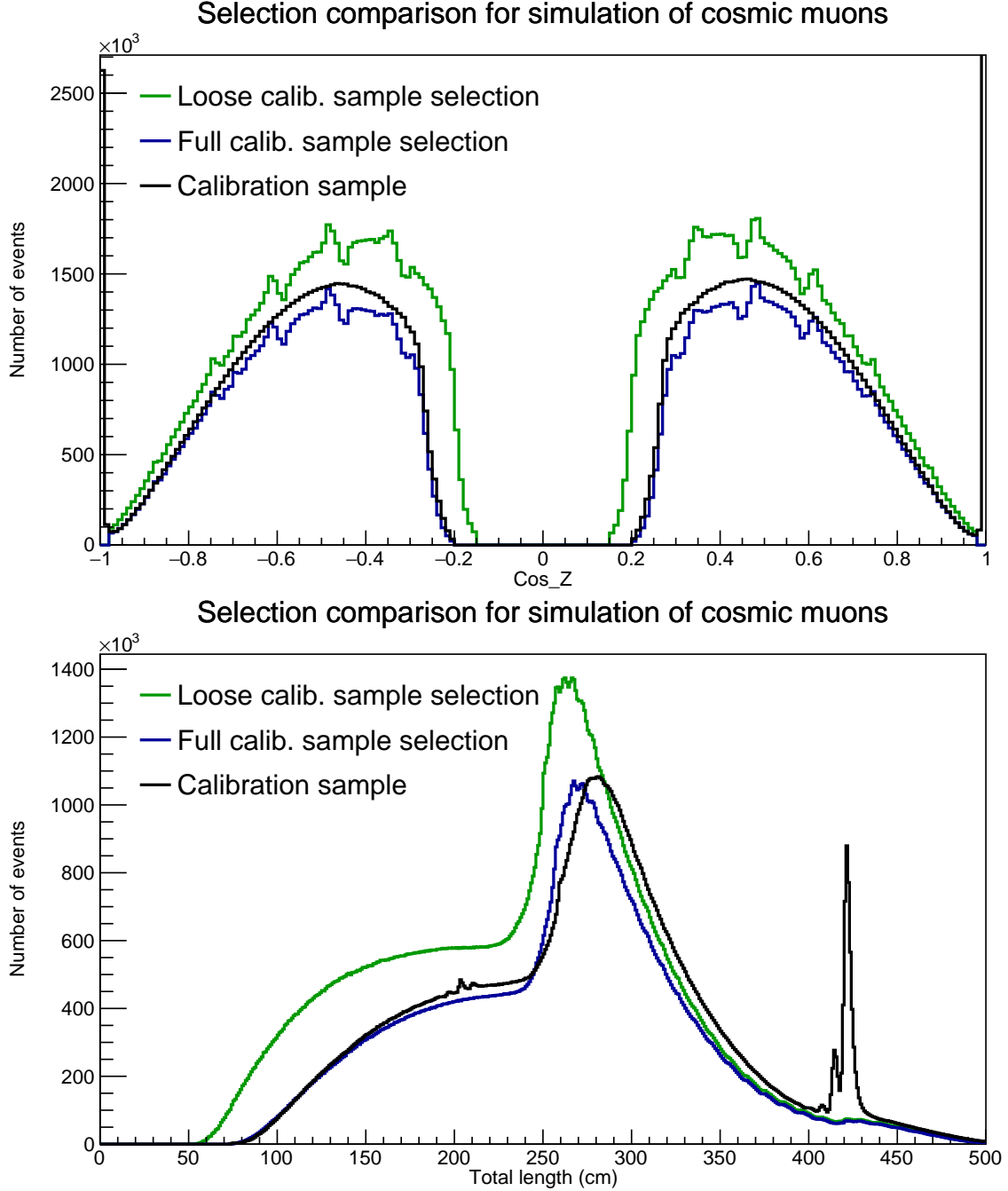


Figure 4: Comparison of event selections for the data-based simulation and of the corresponding data calibration sample in black. The green line represents the final selection used for the simulation, while the blue line shows the distributions with full calibration sample cuts. The data calibration sample was made with the same full calibration sample cuts as the blue line (without the track start cut and maximum Cos_Z cuts), but applied to the window cosmic tracks instead of the Break Point Fitter tracks. We loosened the calibration sample cuts in an attempt to mitigate the differences between the blue and the black distributions. All of the distributions are made from the period 4 Test Beam data.

2.2 Energy correction, charge assignment and smearing

Once we have the TTree with kinematic information for the selected events we run a python script CosmicStudies/GenerateHEPEVTFromROOT.py. This script performs several tasks, including correcting energies of the through-going muons, assigning a charge to each muon, applies smearing and converting the information into a HEPEVT format, which will be used in the Text File Generator.

To load the TTree ntuples into a dictionary of numpy arrays, the script utilizes the uproot library. It is important to note that the machine being used may not have uproot installed. If necessary, the command `pip install -user uproot` can be executed to install the library.

During a detector systematics planning session in Summer 2021, Mark Messier and Teresa Lackey presented an overview and a strategy for data-based simulation of cosmic muons for calibration [?]. They discussed potential improvements to the energy estimation of through-going muons, charge assignment based on energy distribution and a plan to implement data-based simulation in the Near and Far detectors.

To run the python script do:

```
python generate_hepevt_cosmic.py inFile.root outFile.txt\  
                                --niter Niterations
```

2.2.1 Energy correction

Muons that do not stop inside the detector carry away the energy not deposited in the detector and their true initial energy is therefore larger than the reconstructed energy (E_R) we get from the Break Point Fitter. In general, the energy spectrum of cosmic muons can be approximately described by a power law $E^{-\alpha}$, with $\alpha \approx 2.7$ [?, ?]. The expectation value for the "true" energy of the through-going muons can be calculated as

$$\langle E \rangle = \frac{\int_{E_R}^{E_C} E \cdot E^{-\alpha}}{\int_{E_R}^{E_C} E^{-\alpha}} = \left(\frac{\alpha - 1}{\alpha - 2} \right) \left(\frac{E_C^{2-\alpha} - E_R^{2-\alpha}}{E_C^{1-\alpha} - E_R^{1-\alpha}} \right) \quad (1)$$

where E_C is the critical energy chosen to be 300 GeV, as we do not expect muons with higher energies to be selected due to large showers along their paths. Plot 5 shows the corrected energy distribution of our selected events and demonstrates that the choice of the critical energy does not significantly change the correction of the true energy.

To improve the energy correction we can also include angular dependence for the energy correction as described in the PDG [?]; This has not been explored in the scope of the work presented here.

2.2.2 Smearing

Smear the events to avoid any bias from the data. Smearing is done by randomly changing the total momentum within 2%, azimuthal angle uniformly, polar angle within 4mrad and the X/Y and Z vertex positions within the width or depth of the cell respectively.

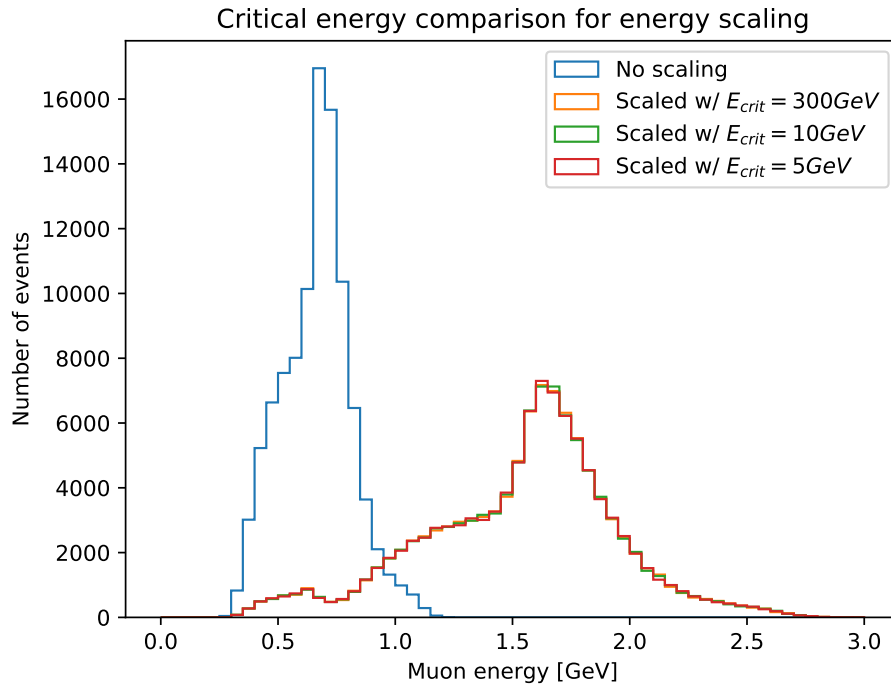


Figure 5: The effect of energy correction for through-going muons with various critical energies. No significant difference can be seen when using different critical energies.

2.2.3 Charge assignment

Text File Generator requires the charge of each muon event. However we do not reconstruct the charge of the muons, so we have to assign the charge based on a statistical distribution from external measurements [?]:

$$P_+ \simeq 0.539 + \frac{x}{34.5} - \left(\frac{x}{9.48}\right)^2 + \left(\frac{x}{8.27}\right)^3 \quad (2)$$

2.2.4 Number of events to simulate

The python script offers options to adjust the statistics of the generated simulation. We can specify the number of iterations (niter) or choose to skip events (stride). It is crucial to strike a balance between simulating an adequate number of events for thorough calibration and avoiding unnecessary computational and memory requirements.

To ensure a reliable calibration, we aim to have a sufficient number of through-going muons in each cell, view, and fiber brightness bin, with a target $\chi^2 < 0.2$.

Based on observations, approximately 50,000 tricell hits per cell, view, and brightness bin provide reliable results. This figure remains the same for the Near and Far detectors, since the binning for position inside a cell is identical for all detectors (100 bins per cell). However, we must consider the discrepancy between horizontal and vertical cells. Vertical cells, parallel to cosmic muons, receive approximately 5 times fewer hits than horizontal cells. To adequately populate the vertical cells, we multiply the target number (50,000) by 6 (1 horizontal + 5 vertical views), then by the number of fiber brightness bins (12). Finally, we multiply this by the number of cells: 390 for FD, 100 for ND, and 64 for TB. Thus, the minimum total tricell hits required is calculated as:

Detector	$50,000 \times 6 \times 12 \times \text{NCells}$
Test Beam	230,400,000
Near Detector	360,000,000
Far Detector	1,404,000,000

Since we are simulating events not hits and each event can have a vastly different number of successful tricell hits, we need to estimate the average number of tricell hits per event. This number will be different for each detector.

For Test Beam events have on average 5 tricell hits (and 37 cell hits), so we need about 46×10^6 events in our simulation calibration sample. The simulation and calibration selection processes have about 90% efficiency. **So for Test Beam we need to simulate at least 51×10^6 events.** From period 4 we get 159,153,260 events with loose selection and 132,420,230 events with full selection. We decided to divide this sample in half to get a smaller simulation sample in the end.

2.2.5 Print the outputs

Write the pdg code, 4-momentum components and vertex positions into a HEPEVT-format [?] text file which will then be fed to the Text File Generator.

237 In the HEPEVT format each event is described in two lines. The header line contains the event
 238 number (which is ignored in ART) and a non-negative integer number of particles in the event. The
 239 second line contains 15 entries to describe each particle in the following order:

- 240 1. status code (set to 1 for any particle to be tracked)
- 241 2. pdg code for this particle
- 242 3. entry for the first mother of this particle in the event (0 means no mother)
- 243 4. entry for the second mother of this particle in the event
- 244 5. entry for the first daughter of this particle in the event
- 245 6. entry for the second daughter of this particle in the event
- 246 7. x component of the particle momentum
- 247 8. y component of the particle momentum
- 248 9. z component of the particle momentum
- 249 10. energy of the particle
- 250 11. mass of the particle
- 251 12. x position of the particle initial position (vertex)
- 252 13. y position of the particle initial position (vertex)
- 253 14. z position of the particle initial position (vertex)
- 254 15. time of the particle production (relative to the beginning of the event)

255 The momenta and masses are in GeV, positions in centimeters and time in nanoseconds.

256 An example description of a single event with a single anti-muon, with momentum $P_{x,y,z} =$
 257 $(0.149320, -1.561071, 0.653841)$ GeV and vertex position $V_{x,y,z} = (40.476409, 121.044924, 120.964778)$ cm.
 258 The mass is the same for all particles since we assume they are all muons, energy is calculated from
 259 the three momenta and the mass. The time of the particle production is chosen to be $50\mu s$ for all
 260 particles.

	26375015	1				
261	1	-13	0	0	0	(no newline)
	0.149320	-1.561071	0.653841	1.702346	0.106	(no newline)
	40.476409	121.044924	120.964778	50000		

262 More details can be found in the comment block of the Text File Generator module in
 263 [novasoft/EventGenerator/TextFileGen_module.cc](https://github.com/novasoft/EventGenerator/TextFileGen_module.cc).

2.3 Submitting the simulation jobs

1. The output of `generate_hepevt_cosmic.py` script is a single large `.txt` file (let's call it `TextGen_inFile.txt`). To run it on the grid, split it into multiple subfiles, each will be sourced by a separate FHiCL file. Don't forget that each event is written on 2 lines in the `txt` file, so split it into an even number of lines. From experience, 125,000 events (250,000 lines) are optimal, where each job runs for only a few hours. 250,000 events could be considered if the number of created subfiles would be >1000 .

```
split -d -l 250000 --additional-suffix=.txt\  
TextGen_inFile.txt /path/to/new/files/TextGen_inFile_
```

2. Then create the same amount of FHiCL files, each sourcing a different text file. There is a template called `TextFileGenTBjob_template.fcl`. Take a look at it and check (not only):

- `maxEvents`
- `firstRun`, `firstSubRun`
- `physics.producers.photrans.nd/fd/tb.BrightnessFile` (which brightness file is used for the simulation)

The brightness file describes the relative differences in energy response across the different cells and planes. These differences mainly arise from the variability of each fiber's brightness, and specifically for Test Beam, also from the different scintillators used. Since we want the simulated detectors to be functional copies of the ideal versions of the real detectors, it is important to provide a correct brightness file without any defects. More information about the brightness files and how to create them can be found on the [Test Beam calibration redmine wiki page](#).

In the first iteration of the data-based simulation, Teresa used a test beam brightness file created from period 2 data, which contains faulty FEBs and underfilled cells, resulting in a simulation also containing these defects. In the second iteration Robert created a new brightness file from period 4 test beam data, which are free from any irregularities and supplied that to the simulation

```
bash CreateFclsForSimulation.sh  
TextFileGenjob_template.fcl  
/path/to/TextGen_infiles_directory/  
/path/to/output/TextGen_fcl_directory
```

3. Then create a SamWeb definition out of these FHiCL files

```
sam_add_dataset -n username_CosmicGen_description  
               -t username_date  
               -d /path/to/FHiCL/directory
```

You can also use the option `-no-rename` if you've named your FHiCL files uniquely enough.

4. Adjust the configuration script accordingly (`njobs`, `defname` for your `fcls`, `dest`,...) and include all the text files with:

```
for file in $(ls -l /path/to/TextGen_inputfiles/); do
echo --inputfile $file >> TextFileGen.cfg; done
```

If you need to remove the previously included text files, you can do

```
sed -i '/^--inputfile/d' TextFileGen.cfg
```

5. Submit. First with the `-test_submission` flag and if everything looks all right, comment it out.

```
submit_nova_art -f TextFileGen.cfg
```

6. This results in `artdaq-stage` simulation files. Move them to a suitable area for long storage (for example persistent, ask production) and investigate them. You can re-use the `Cosmic-GenAna` module from section 2.1.

7. To create the calibration samples you can ask production to create them for you or to point you to a corresponding job to use. For Test Beam you can use [this](#) script

```
novaproduct/novaproduct/fcl/testbeam/
prod_tb_ddactivity1_pclist_mc_job.fcl
```

This creates the simulation calibration samples. You should move them a suitable area (like persistent) and create new definitions from them. You can again ask production for help.

3 Validation

To validate whether our simulation is working as expected, we

1. compare the new simulation calibration sample with the data calibration sample and
2. use the new simulation `artdaq-stage` sample as "fake data" and repeat the simulation process to create a "re-simulation".

The data simulation comparisons on figures 6 and 7 show that the angular distributions of the new simulation are basically the same as distributions of break point fitter tracks with full calibration sample selection (blue dashed lines). This means that loosening up the calibration sample selection did not help with compensating for the underlying differences between the break point fitter tracks and window cosmic tracks. This can also be seen on the total track length distribution on figure 7.

343 The start of track comparison between data and simulation on figures 8 and 9 indicates that the
344 starts of tracks of the new simulation are spread out more into the inside of the detector. This is
345 likely the result of the event smearing.

346 Adding the distributions for the re-simulation calibration sample shown on figure 10, we can
347 see that the tracks' starts are systematically shifted even more towards the inside of the detector.
348 This would support the theory that this effect is caused by the smearing of the events. This is
349 also likely directly related to the loss of events with longer track lengths also shown on figure 10.
350 If tracks start a few centimeters later in the detector their tracks would get shorter by the same
351 amount.

352 4 Conclusions

353 We created a new version of data-based simulation of cosmic muons for the Test Beam calibration.
354 The new simulation is performing way better than the old version of the data-based simulation and
355 the CRY based simulations used in the earlier stages. The results of the new simulation inside the
356 Test Beam calibration process are described in the Test Beam calibration technical note [cite the
357 docdb for the TB calibration technote].

358 The SAMWEB definitions for the new simulation samples are:

359 **artdaq:**

360 rkralik_artdaq_testbeam_databasedsim_R23-04-05-testbeam-production.a

361 **pclist:**

362 rkralik_pclist_testbeam_databasedsim_R23-04-05-testbeam-production.a

363 **pcliststop:**

364 rkralik_pcliststop_testbeam_databasedsim_R23-04-05-testbeam-production.a

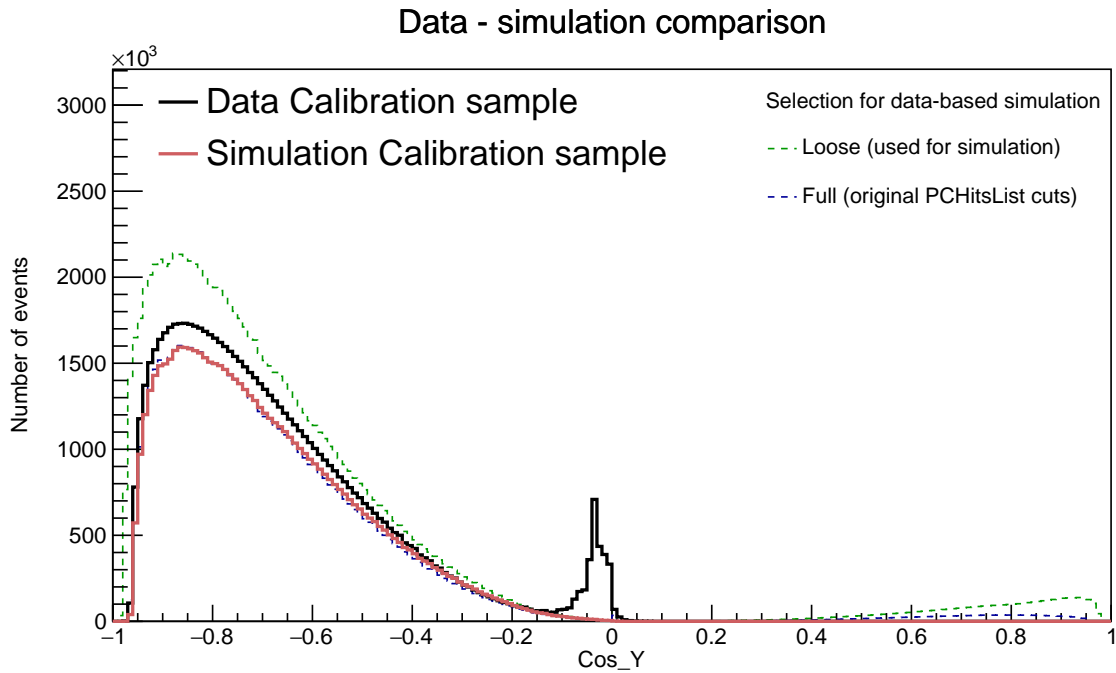
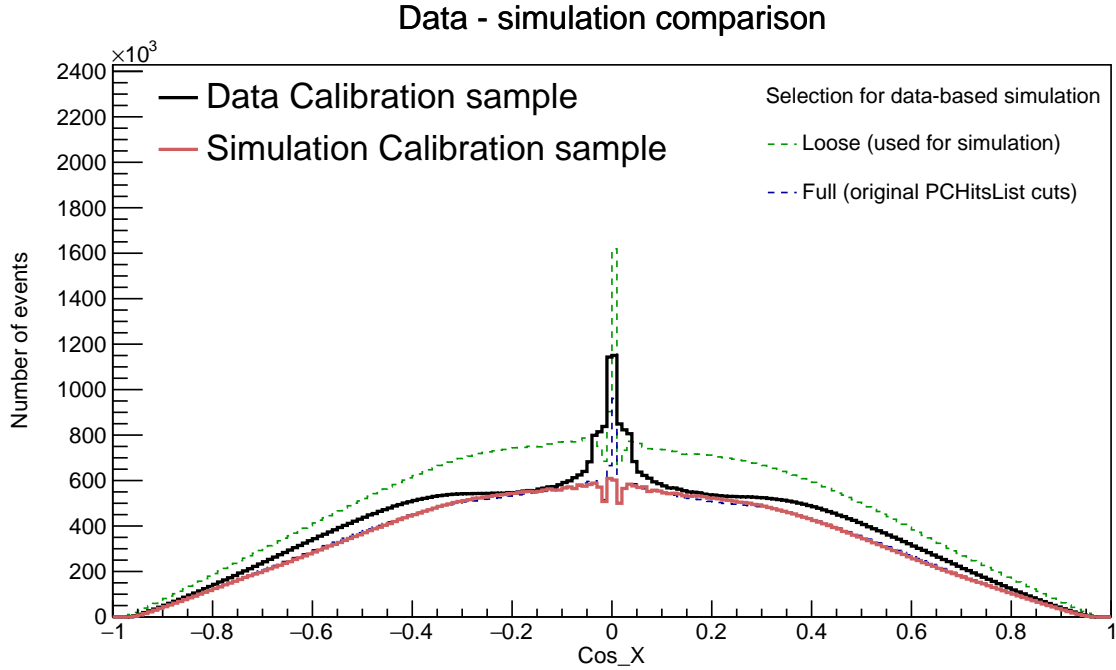


Figure 6: Angular distribution comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the data selection used for the new simulation in green and a "full calibration sample" selection (same as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of window cosmic tracks) in blue.

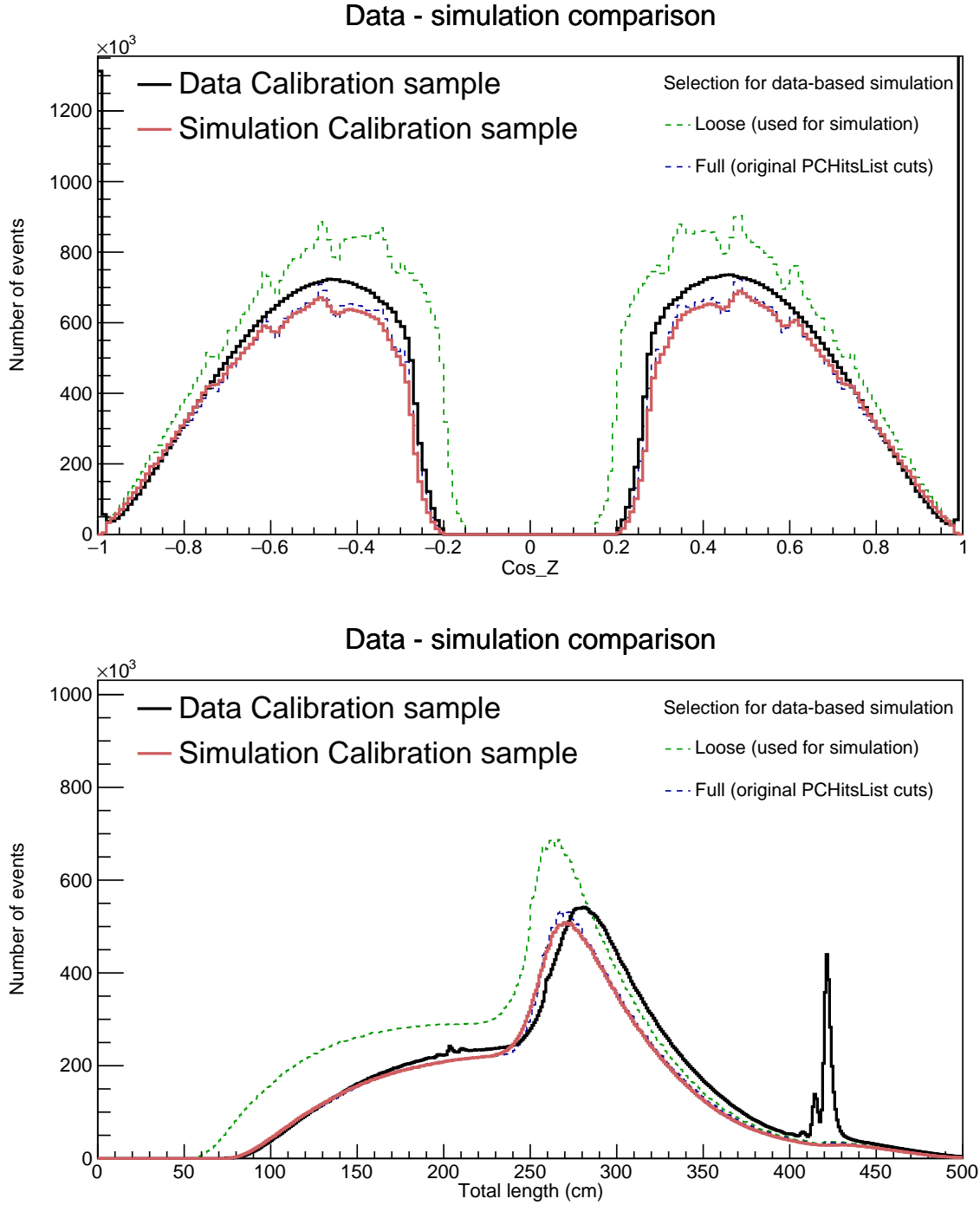


Figure 7: Angular distribution and total track length distribution comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the data selection used for the new simulation in green and a "full calibration sample" selection (same as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of window cosmic tracks) in blue.

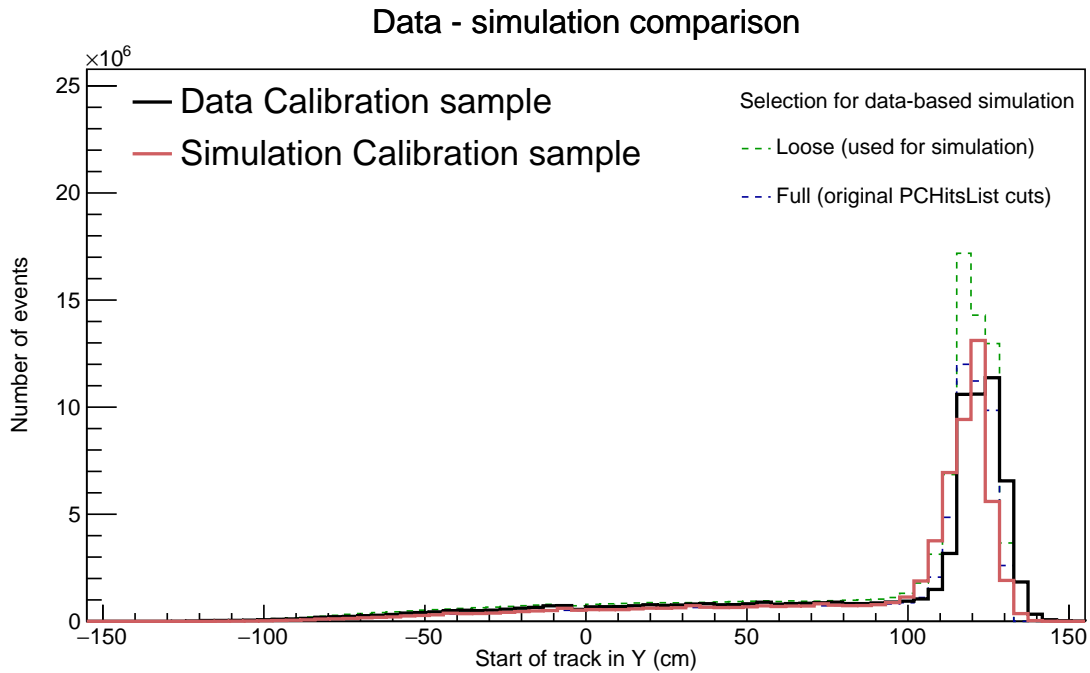
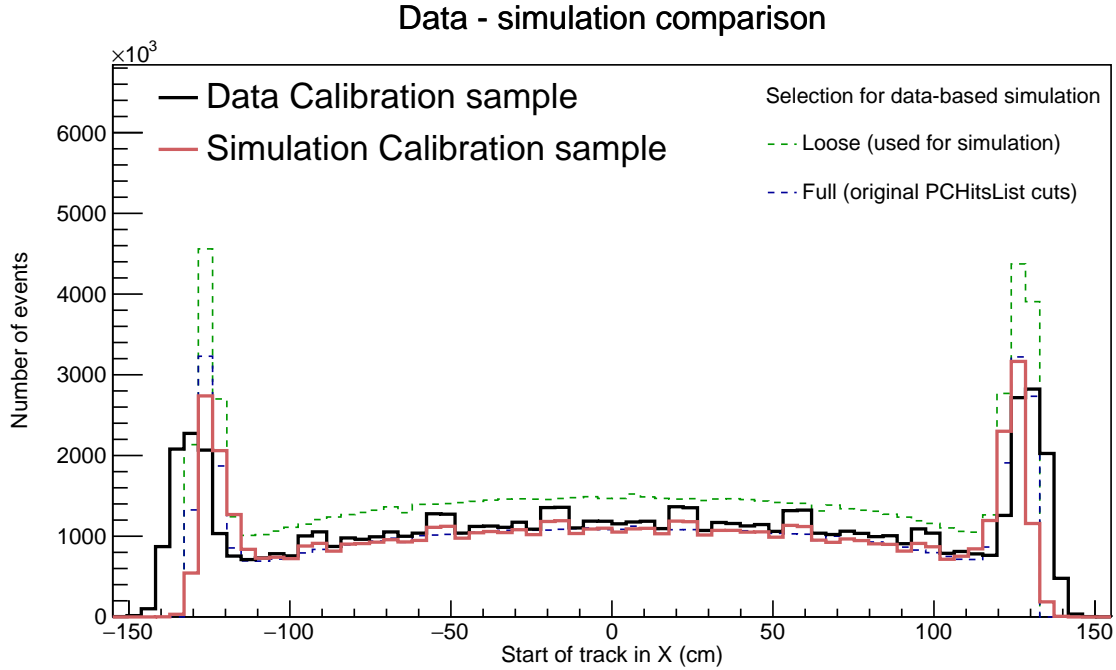


Figure 8: Start of tracks comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the data selection used for the new simulation in green and a "full calibration sample" selection (same as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of window cosmic tracks) in blue.

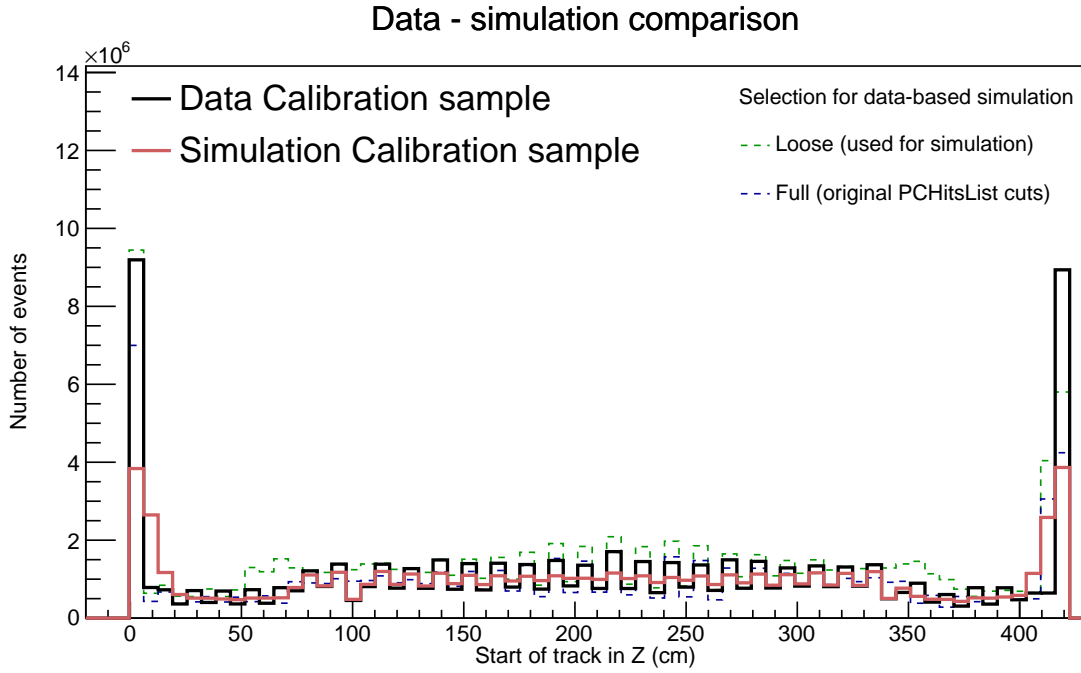


Figure 9: Start of tracks comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the data selection used for the new simulation in green and a "full calibration sample" selection (same as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of window cosmic tracks) in blue.

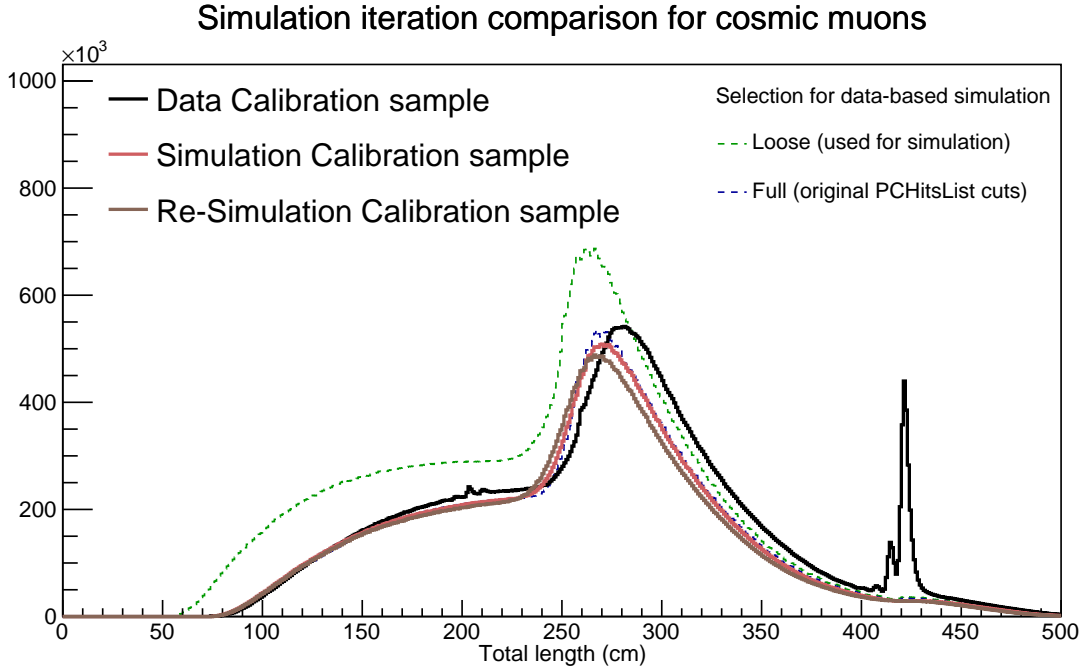
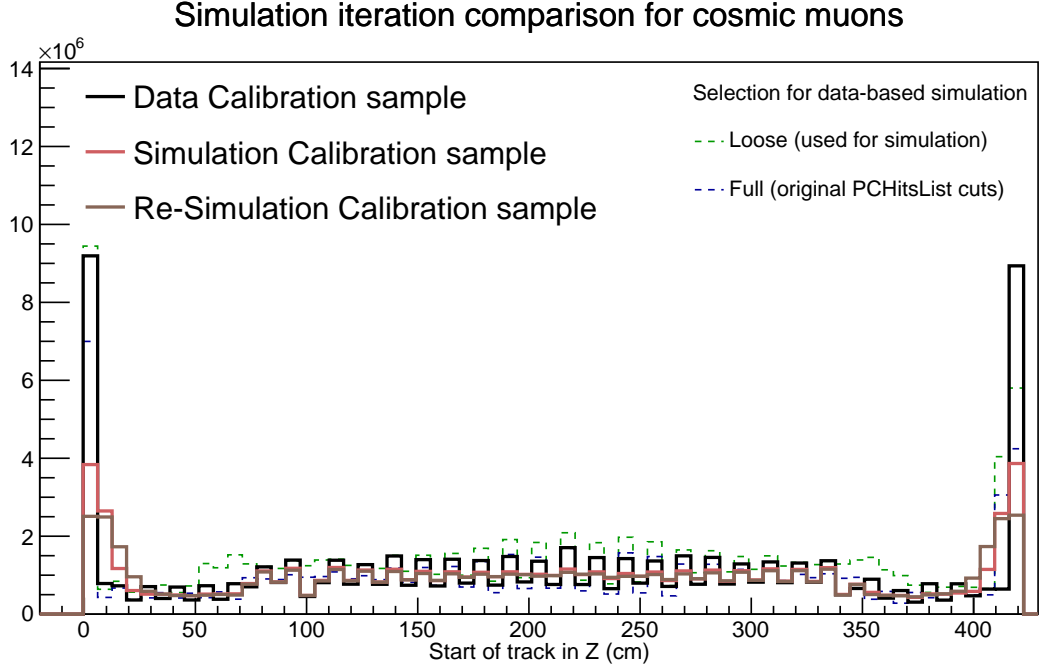


Figure 10: Distribution of the re-simulation, when the new simulation's artdaq sample is used as "fake data" for a new iteration of the simulation process discussed in this document. This is compared to the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the data selection used for the new simulation in green and a "full calibration sample" selection (same as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of window cosmic tracks) in blue.