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

Technical Note

Róbert Králik[1]

[1]University of Sussex, Brighton, UK

February 22, 2024

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) [1] 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

The data-based simulation of cosmic muons was initially developed by Teresa Lackey [2] 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 [3], 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.

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

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

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

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

The text file is then passed to the `Text File Generator`, which employs the information as a seed for a geant4-based [5] detector simulation. By incorporating additional simulation and reconstruction steps, an artdaq-stage simulation sample of cosmic muons is generated. To create calibration samples for stopping and through-going cosmic muons, we apply the same reconstruction and selection criteria as for any data sample. Further details are provided in Sec. 2.3.

A step-by-step summary is presented in the box below and can also be found on the DataDriven 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` to the grid;
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 them 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 event time relative to the time of the beam spill. For Test Beam the beam spill is 4.2 seconds long and we remove all events within a 5 seconds window from the start of the beam spill, as shown on Fig. 1. This should leave us with mostly cosmic events.

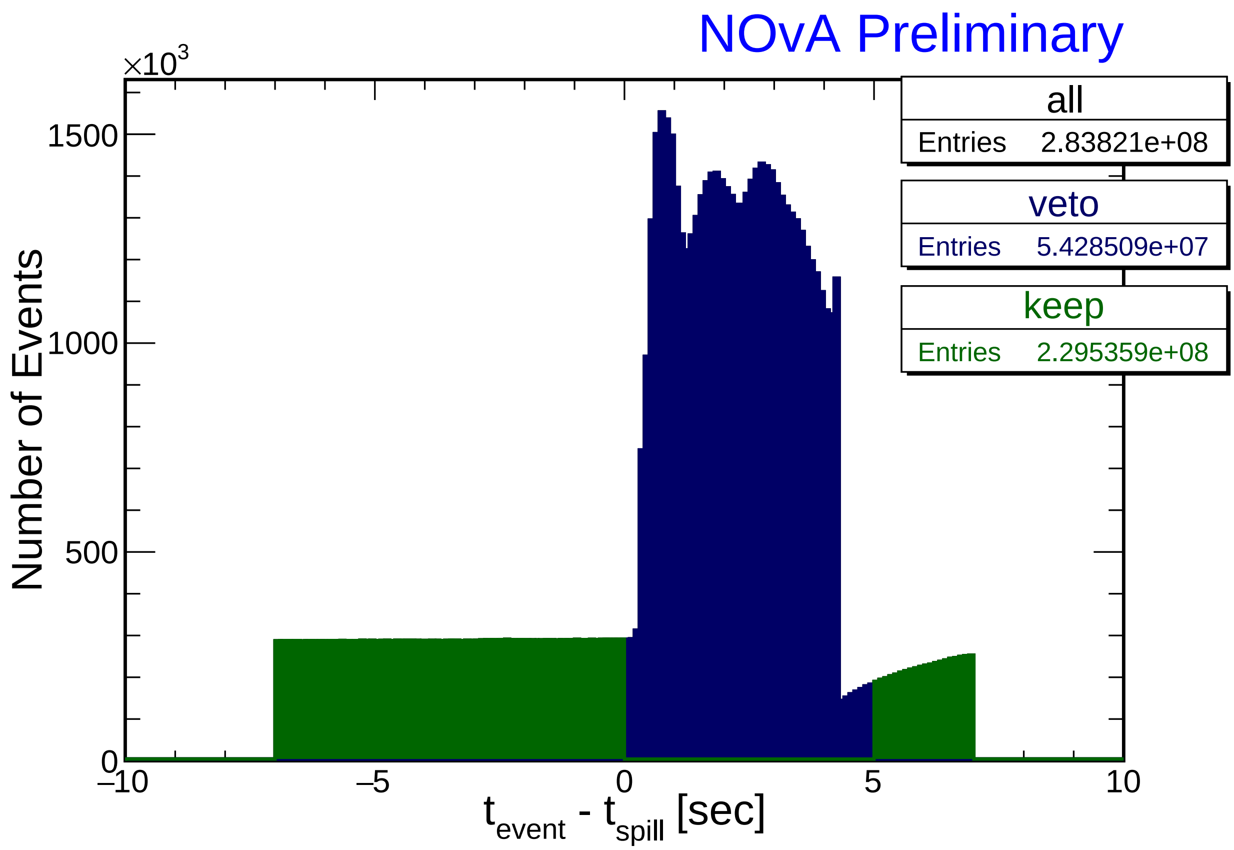


Figure 1: Test Beam beam spill events removed (blue) from the calibration samples. The remaining events (green) should mostly consist of cosmic particles. This example and the numbers of entries are for the full period 4 Test Beam sample.

2.1.2 Reconstruction

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

1. `CalHit` to create Calibrated Cell Hits from Raw Digits,
2. `Slicer` to group hits into slices,

3. `Window CosmicTrack` - a tracking algorithm for cosmic particles [6],

4. `CosmicRayVertex` to identify vertices for cosmic particles,

5. `FuzzyKVertex` to cluster hits into prongs and

6. `BreakPointFitter (BPF)` to identify muons (or muon-like tracks) and get their initial 4-momenta by fitting to the FuzzyK prongs and CosmicRayVertex vertices [7].

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 (Tab. 1 provides an overview of all cuts and their corresponding values):

1. We only use successfully reconstructed 3D BPF tracks with the muon assumption [7];

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 Fig. 2;

3. We remove all events whose track is parallel to the beam direction, by requiring the angle from the z (beam) axis to be $|\mathsf{Cos}_Z| \leq 0.98$. Figure 2 demonstrates the presence of events peaked at track lengths of approximately 410 cm and 200 cm, which correspond to the total and half length of the detector, respectively (or alternatively lengths of both modules and a single module). These events are strictly parallel to the beam direction and are likely remnants of beam events. Applying a cut on Cos_Z 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. This module 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 apply selection to tracks from the **Window cosmic track** algorithm instead of the Break Point Fitter algorithm, which yield different distributions as depicted on Fig. 3. Notably, the BPF tracks have a hard cut-off at the detector edges, whereas the Window cosmic tracks are allowed to start beyond these limits. Also, the BPF tracks have a rugged distribution in Cos_Z, which is not present for Window cosmic tracks. This is likely caused by the detector structure, as shown on Fig. 4, but it is not clear how. We concluded that he rugged shape does not have any impact on the resulting simulation. Given these differences between the tracking algorithms, 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 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 calibration cuts applied to the BPF tracks are listed in Tab. 1 and shown on Fig. 3. There we also show the data calibration sample, which was created by applying the full calibration cuts 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 position[1]. This information assists in correcting the energy of through-going muons, as outlined in Sec. 2.2.

[1] For Test Beam we say it is a stopping muon if its track ends at least 20 cm from any edge of the detector. For the far and near detector this is 50 cm. This value was chosen by Kevin Mulder [8] as 50 cm removed too many cosmic events from the Test Beam detector.

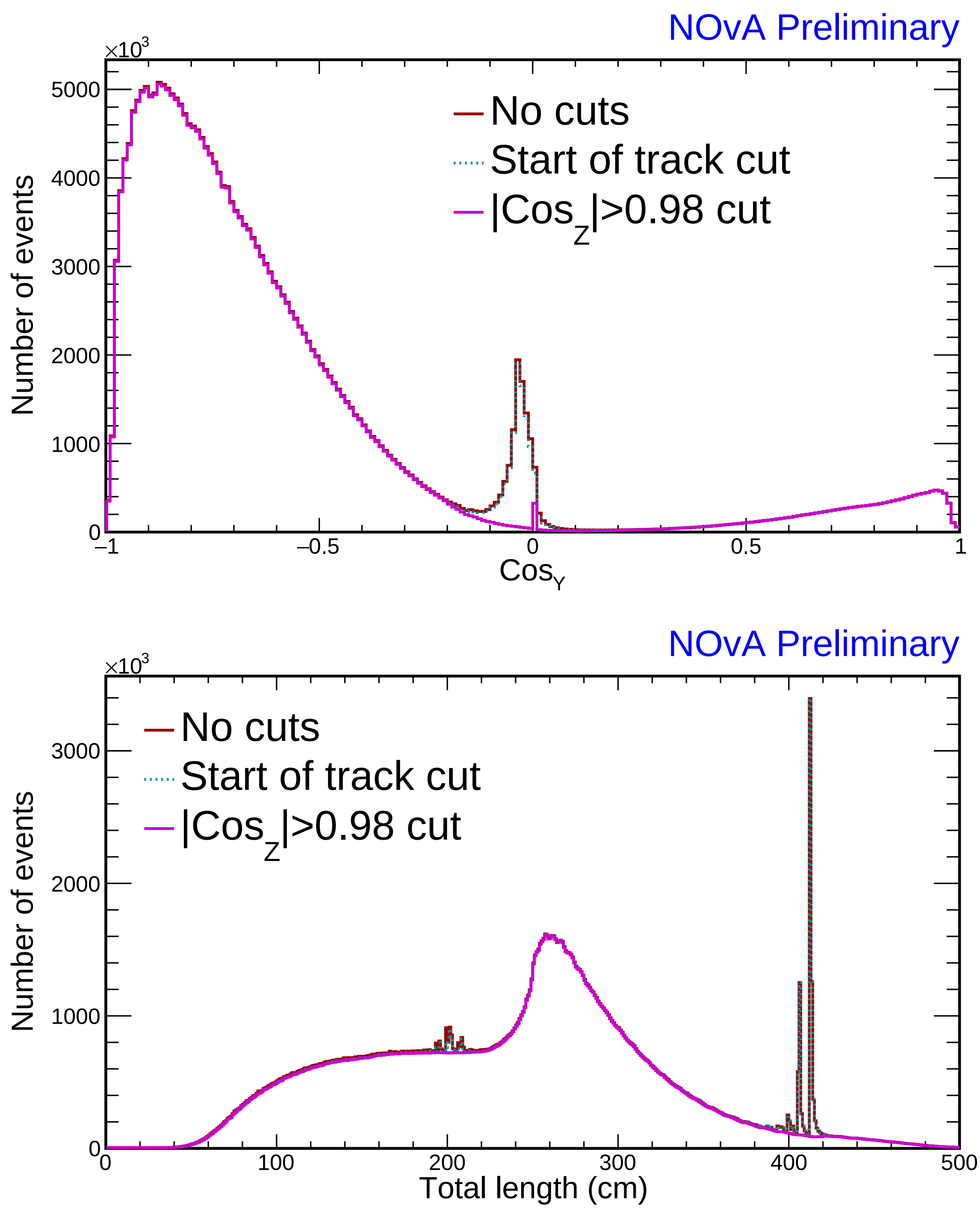


Figure 2: Impact of track start and maximum track angle from the z axis (Cos_Z) cuts on the Test Beam data for the data-based simulation of cosmic muons. The track start cut has only negligible effect. The maximum Cos_Z cut effectively removes sharp peaks in the total track length distribution and events 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.

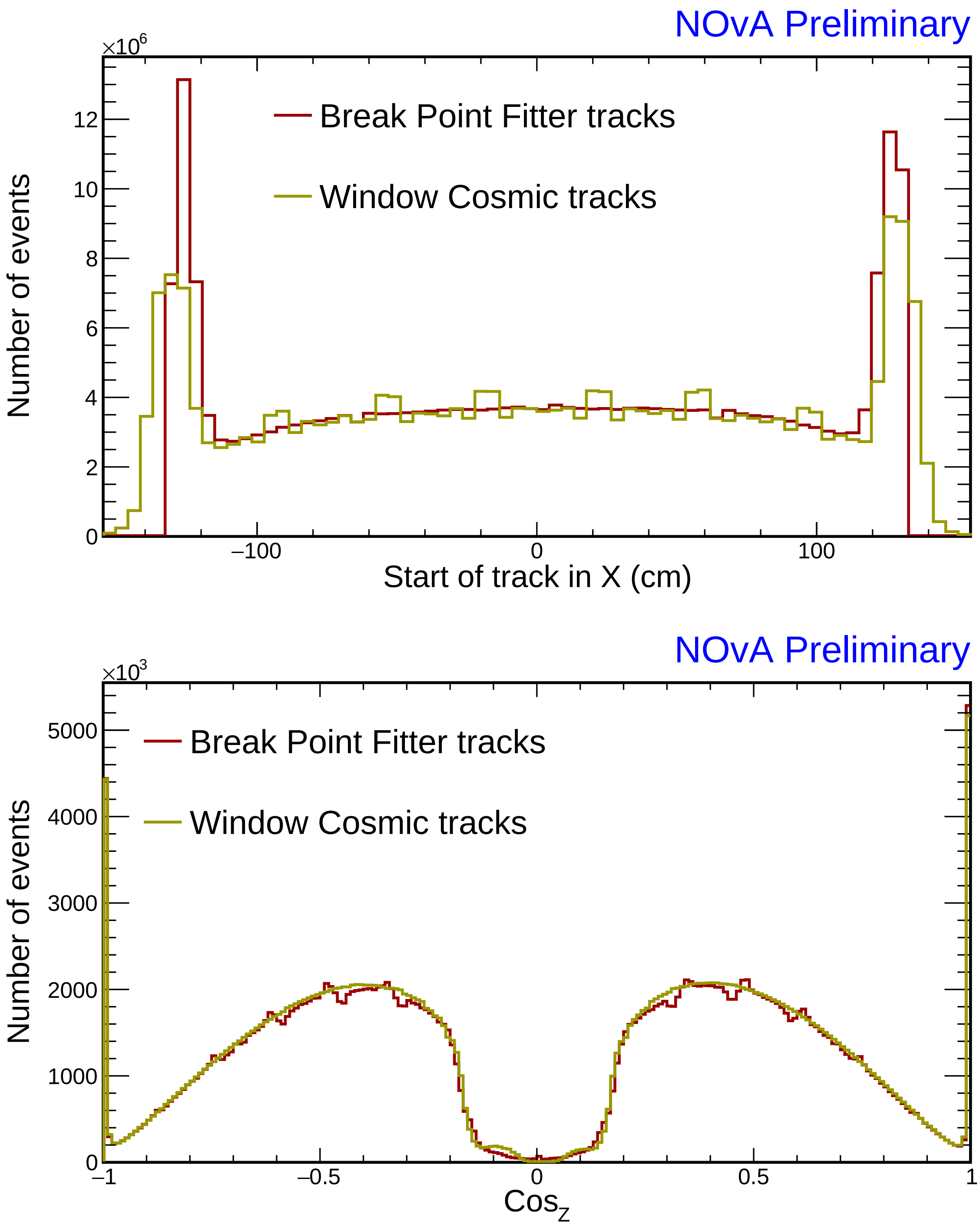
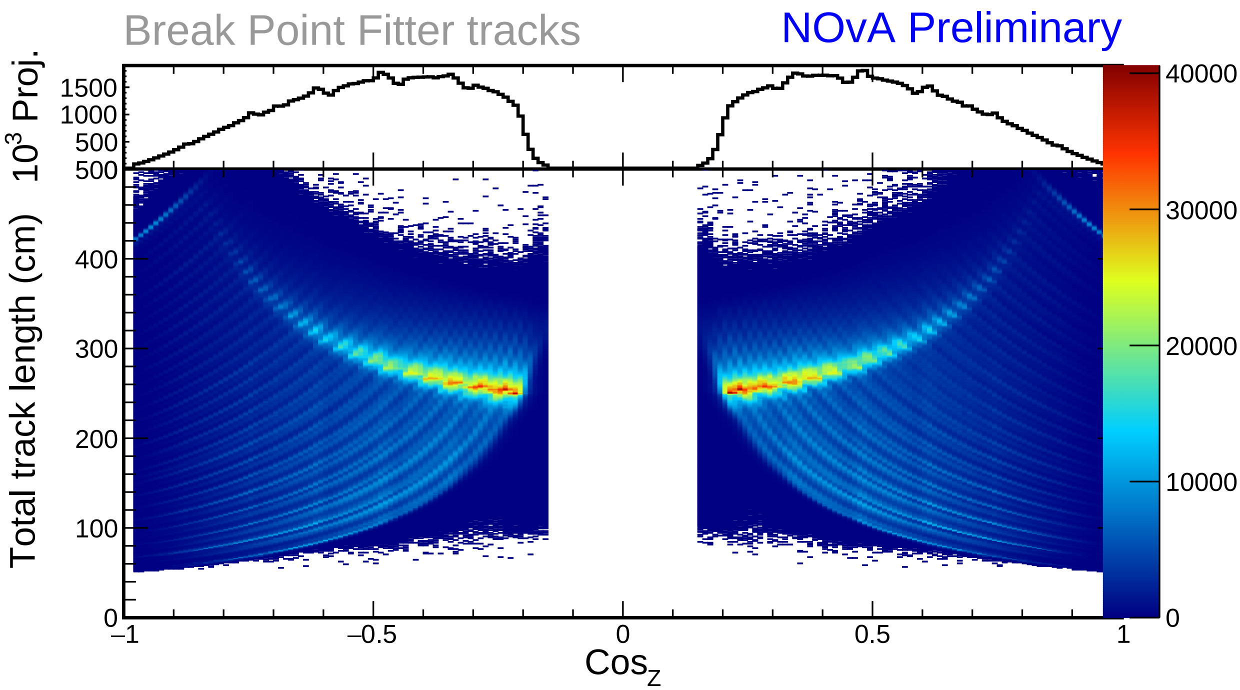


Figure 3: Difference between the tracks reconstructed with the break point fitter and with the Window cosmic track algorithms. Both distributions are for the period 4 Test Beam data (with removed beam spill) without applying any selection.



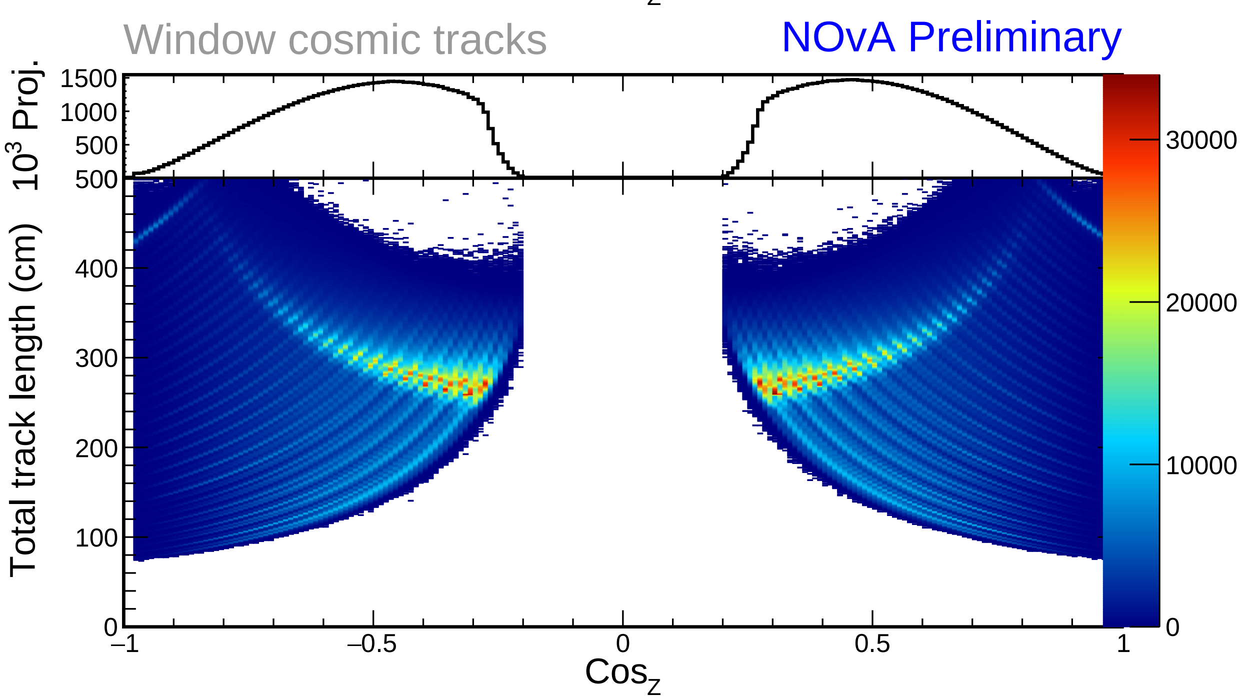


Figure 4: Investigating the origin of the rugged shape in the Cos$_Z$ distribution of Break Point Fitter tracks. The top plot is created with the loose calibration cuts and the bottom plot with full calibration cuts. This difference in selection shouldn't matter. The long lines on the 2D plots are likely the effects of the detector structure. We can see that for the Break Point Fitter tracks, each Cos$_Z$ angle corresponds to a specific track length, whereas for the Window cosmic tracks there is multiple track length for each angle. This could cause the resulting shape in the Cos$_Z$ distribution of Break Point Fitter tracks.

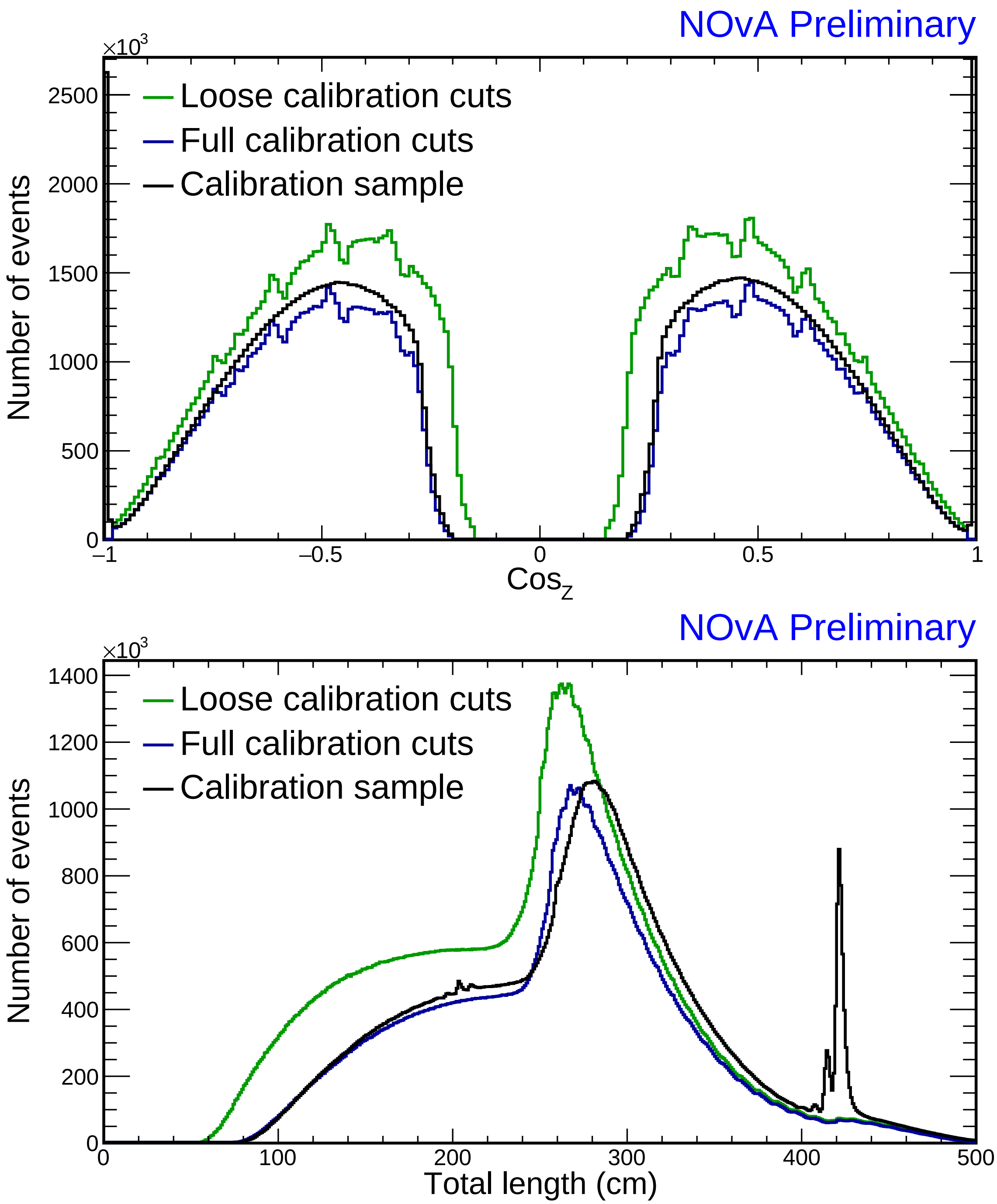


Figure 5: Comparison of event selections for the data-based simulation and of the corresponding data calibration sample in black. The green line represents the final selection used for the simulation, using the loosened calibration cuts, while the blue line shows the distributions with full calibration cuts. The data calibration sample was made with the same full calibration 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. 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, and smearing and converting the information into a `HEPEVT` format (see Sec. 2.2.5), which is required by the `Text File Generator`.

To load the TTree ntuples into a dictionary of numpy arrays, the script utilizes the `uproot` library. If the machine being used does not have `uproot` installed, we can use `pip install -user uproot` 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 [9]. 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. Work discussed in this section directly follows this discussion.

To run the python script do:

```
python GenerateHEPEVTFromROOT.py inFile.root outFile.txt\
                                 --niter NIterations\
                                 --stride IStride
```

2.2.1 Energy correction

Through-going muons do not deposit all of their energy inside the detector. From data we cannot reliably calculate their initial energies, but we can estimate an energy that would leave the same track. In general, the energy spectrum of cosmic muons can be approximately described by a power law $E^{-\alpha}$, with $\alpha \approx 2.7$ [9, 10]. The expectation value for the "true" initial energy of through-going muons can be therefore calculated as

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

where E_R is the reconstructed energy from the Break Point Fitter. 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.

We use this corrected initial energy for all muons that do not stop inside the detector (as identified during selection), as described in Sec. 2.1.3. Plot 6 shows the corrected energy distribution of our selected events and demonstrates that the choice of the critical energy does not significantly change the correction.

This corrected energy is **not** a good representation of the true energy spectrum of cosmic muons on surface level and getting a correct energy distribution from data would require a much more dedicated effort. The corrected energy would also be different for different NOvA detectors, since the reconstructed energy is calculated from the track length. For example, the corrected energy of

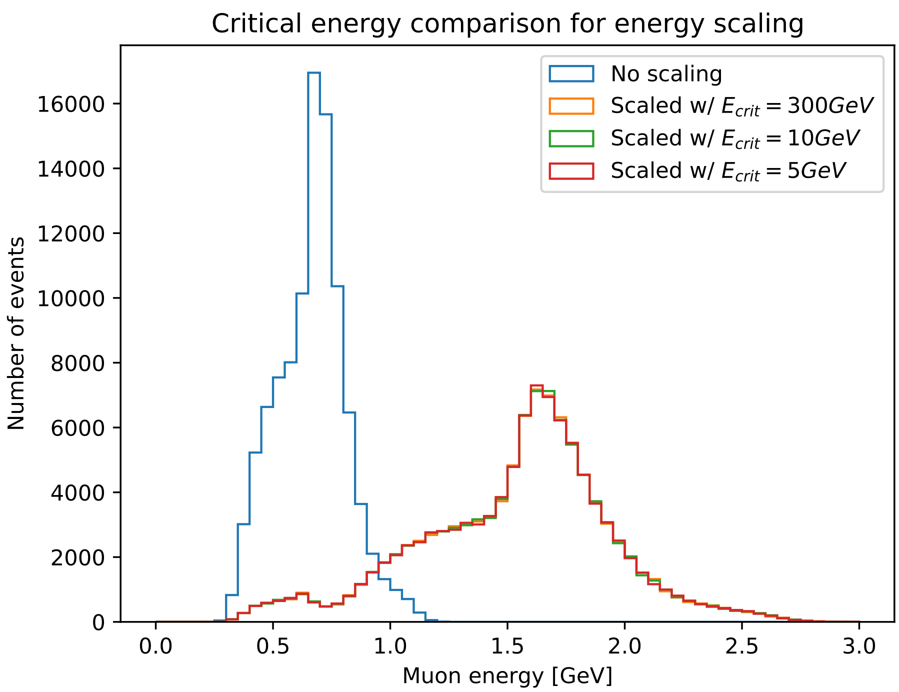


Figure 6: The effect of energy correction for through-going muons with various critical energies. No significant difference can be seen when using different critical energies.

cosmic muons when entering the detector would be larger for the bigger near detector than for Test Beam, even though the near detector is underground.

However, since this simulation is intended to be used for calibration, where we use through-going muons only for relative calibration, we do not need a perfect representation of cosmic muon energy spectrum. Not including more energetic cosmic muons into the simulation does bias the energy deposition towards lower average, but this is corrected for during absolute calibration which only uses stopping muons. See Test Beam calibration technical note for further information [11].

If someone were to use this simulation for something other than calibration, it would be necessary to rethink the energy correction, either by changing the energy estimation from track based algorithms to energy deposition, or by including information from external sources. It would also be necessary to include angular dependence for the energy correction as described in the PDG [10].

2.2.2 Smearing

The reconstructed data is influenced by the detector structure, reconstruction efficiencies and other effects that can bias the simulation. To avoid this influence, we apply smearing. Smearing is done by randomly changing

- total momentum within 2%,
- azimuthal angle uniformly,
- polar angle within 4 mrad,
- and the X/Y and Z vertex positions within the width or depth of the cell respectively.

The size of the smearing has been decided as the best estimate of variations of these variables for cosmic muons.

2.2.3 Charge assignment

We need to tell the `Text File Generator` whether to simulate a muon or an anti-muon. However we do not reconstruct the charge of the muons, so we have to randomly assign it based on a statistical distribution from external measurements [9]:

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

where x is the logarithm of total momentum in GeV.

2.2.4 Number of events to simulate

The `GenerateHEPEVTFromROOT.py` script offers options to adjust the statistics of the generated simulation. We can specify the number of iterations (niter) to loop each data event, or we can choose to skip events (stride). We want to avoid unnecessary computational and memory requirements by simulating too many events, but also require enough events for a successful calibration.

Calibration is successful when there is sufficient number of through-going muons in each cell, view, and fibre brightness bin, so that each attenuation fit has a $\chi^2 < 0.2$. This is an external condition set for NOvA calibration.

Here are our considerations, based purely on observation and experience. The intention is solely to help the reader avoid fine tuning the statistics when making a simulation of their own. Laid back reader can skip this section.

Attenuation fits require approximately 50,000 calibration (tricell) hits per cell, view, and brightness bin. This Fig. remains the same for the near and far detectors, since the binning for position inside a cell (w) is identical for all detectors (100 bins per attenuation profile). 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 original number (50,000) by 6 (1 for horizontal + 5 for vertical view), then by the number of fibre 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	$50,000 \times 6 \times 12 \times$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, with far detector events having many more hits than Test Beam events.

From the calibration files we got that 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 data we get 159,153,260 events after the selection. We decided to divide this sample in half with `stride 2` to get about 80×10^6 events, which was sufficient for a successful calibration. Originally we tried to divide the sample into a third, but this resulted in a few uncalibrated cells and a few days worth of work lost. We therefore recommend to rather use more events than less.

2.2.5 Output format

For each event, we write the pdg code, the 4-momentum components and the vertex positions in a HEPEVT-format [4] into a text file, which will then be used as an input for the `Text File Generator`.

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

1. status code (set to 1 for any particle to be tracked)

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

Table 2: Example HEPEVT-style kinematic description of a single anti-muon particle.

2. pdg code for this particle
3. entry for the first mother of this particle in the event (0 means no mother)
4. entry for the second mother of this particle in the event
5. entry for the first daughter of this particle in the event
6. entry for the second daughter of this particle in the event
7. x component of the particle momentum
8. y component of the particle momentum
9. z component of the particle momentum
10. energy of the particle
11. mass of the particle
12. x position of the particle initial position (vertex)
13. y position of the particle initial position (vertex)
14. z position of the particle initial position (vertex)
15. time of the particle production (relative to the beginning of the event)

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

Table 1 shows an example description for a single anti-muon particle, with momentum $P_{x,y,z} = (0.15, -1.56, 0.65)$ GeV and vertex position $V_{x,y,z} = (40.48, 121.04, 120.96)$ cm. The energy is calculated from the momentum and the mass. The time of the particle production is chosen to be 50μs and is same for all particles. The second line of the description is divided into three rows in the table below to fit on this page, but needs to be in a single line inside the text file.

More details can be found in the comment block of the Text File Generator module in 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, where each will be sourced by a separate `FHiCL` file. Since each event is written on two lines in the `txt` file, it's necessary to 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 fibre'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 -1 /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 CosmicGenAna module from section 2.1 for validation.

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

```
novaprod/novaproduction/fcl/testbeam/
prod_tb_ddactivity1_pclist_mc_job.fcl
```

   This creates the simulation calibration samples. You should move them to 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 with the original data and

2. use the new simulation (specifically the artdaq-stage sample) as "fake data" and feed it into the exact same simulation process to create a "re-simulation" sample.

For the data-simulation comparisons we use the events from the calibration (pclist) samples, which are equivalent to the Window cosmic tracks with full calibration cuts from the CosmicGenAna module, as described in the selection Sec. 2.1.3. We are expecting the new simulation to be similar to the data calibration sample, without a bias from the original data used for the simulation.

Figures 7 and 8 show that the angular distributions of the new simulation (pink lines) are almost the same as distributions of Break Point Fitter tracks with full calibration cuts (blue dashed lines).

This means that loosening up the calibration cuts (green dashed lines) did not help as expected with compensating for the underlying differences between the Break Point Fitter tracks and the Window cosmic tracks. This can also be seen on the total track length distribution on Fig. 8. This is disappointing, but the distributions of the new simulation look reasonable and are close enough to the data, that we've decided to proceed with this simulation and use it in the Test Beam calibration. It is unlikely that we could mitigate these differences by changing the selection even more.

The start of track comparison between data and simulation on Fig. 9 and 10 show that there are fewer events that start at the edge of the detector. This is likely the result of the event smearing.

Adding the distributions for the re-simulation calibration sample shown on Fig. 11, we can see that the tracks' starts are systematically shifted even more towards the inside of the detector. This would support the theory that this effect is caused by the smearing of the events. This is also likely directly related to the loss of events with longer track lengths as shown on Fig. 11. If tracks start a few centimetres later in the detector their tracks would get shorter by the same amount.

4 Conclusions

We created a new version of data-based simulation of cosmic muons for the Test Beam calibration. The new simulation is performing better than the old version of the data-based simulation and also than the CRY based simulations used in the earlier stages. The results of the new simulation inside the Test Beam calibration process are described in the Test Beam calibration technical note [11].

The SAMWEB definitions for the new simulation samples are:

artdaq:

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

pclist:

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

pcliststop:

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

References

[1] Chris Hagmann, David Lange, and Douglas Wright. Cosmic-ray shower generator (cry) for monte carlo transport codes. volume 2, pages 1143 – 1146, 01 2007. doi:10.1109/NSSMIC.2007.4437209.

[2] Teresa Megan Lackey. *Proton Scattering in NOvA Test Beam*. PhD thesis, Indiana U., 7 2022.

[3] Robert Kralik. Data-based cosmics simulation for (not only) test beam calibration. NOVA Document 54417-v1, April 2022. NOvA internal document. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=54417.

[4] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. Pythia 6.4 physics and manual. *Journal of High Energy Physics*, 2006(05):026, may 2006. Section 5.4. URL: https://dx.doi.org/10.1088/1126-6708/2006/05/026, doi:10.1088/1126-6708/2006/05/026.

[5] S. Agostinelli et al. GEANT4–a simulation toolkit. *Nucl. Instrum. Meth. A*, 506:250–303, 2003. doi:10.1016/S0168-9002(03)01368-8.

[6] Brian Rebel. Window tracking algorithm for cosmic ray muons. NOVA Document 15977-v1, August 2016. NOvA internal document. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=15977.

[7] Michael Baird. Break point fitter technical note. NOVA Document 32455-v1, September 2018. NOvA technical note. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=32455.

[8] Kevin Mulder. Testbeam calibration update. NOVA Document 39244-v1. NOvA internal document. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=39244.

[9] Mark Messier and Teresa Lackey. Data driven cosmic generation. NOVA Document 51327-v3, July 2021. NOvA internal document. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=51327&version=3.

[10] R. L. Workman and Others. Review of Particle Physics. *PTEP*, 2022:083C01, 2022. doi:10.1093/ptep/ptac097.

[11] Robert Kralik. NOvA Test Beam detector calibration. NOVA Document 60592, November 2023. NOvA technical note. URL: https://nova-docdb.fnal.gov/cgi-bin/sso/ShowDocument?docid=60592.

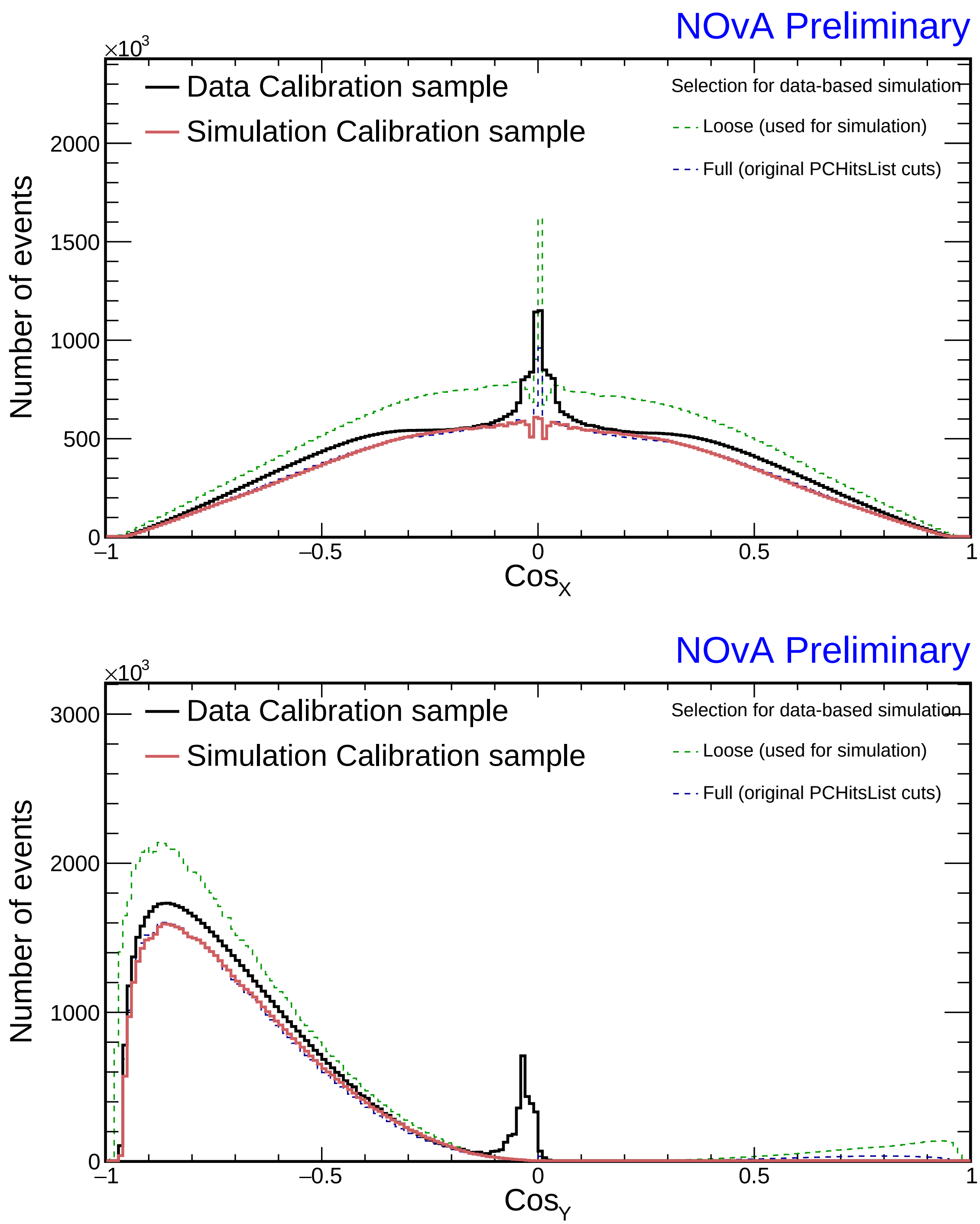


Figure 7: Angular distribution comparison of the newly created simulation calibration sample and the corresponding data calibration sample. We are also showing the selection of data used to create the new simulation in green and a "full calibration cuts" selection (same cuts as used for the simulation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

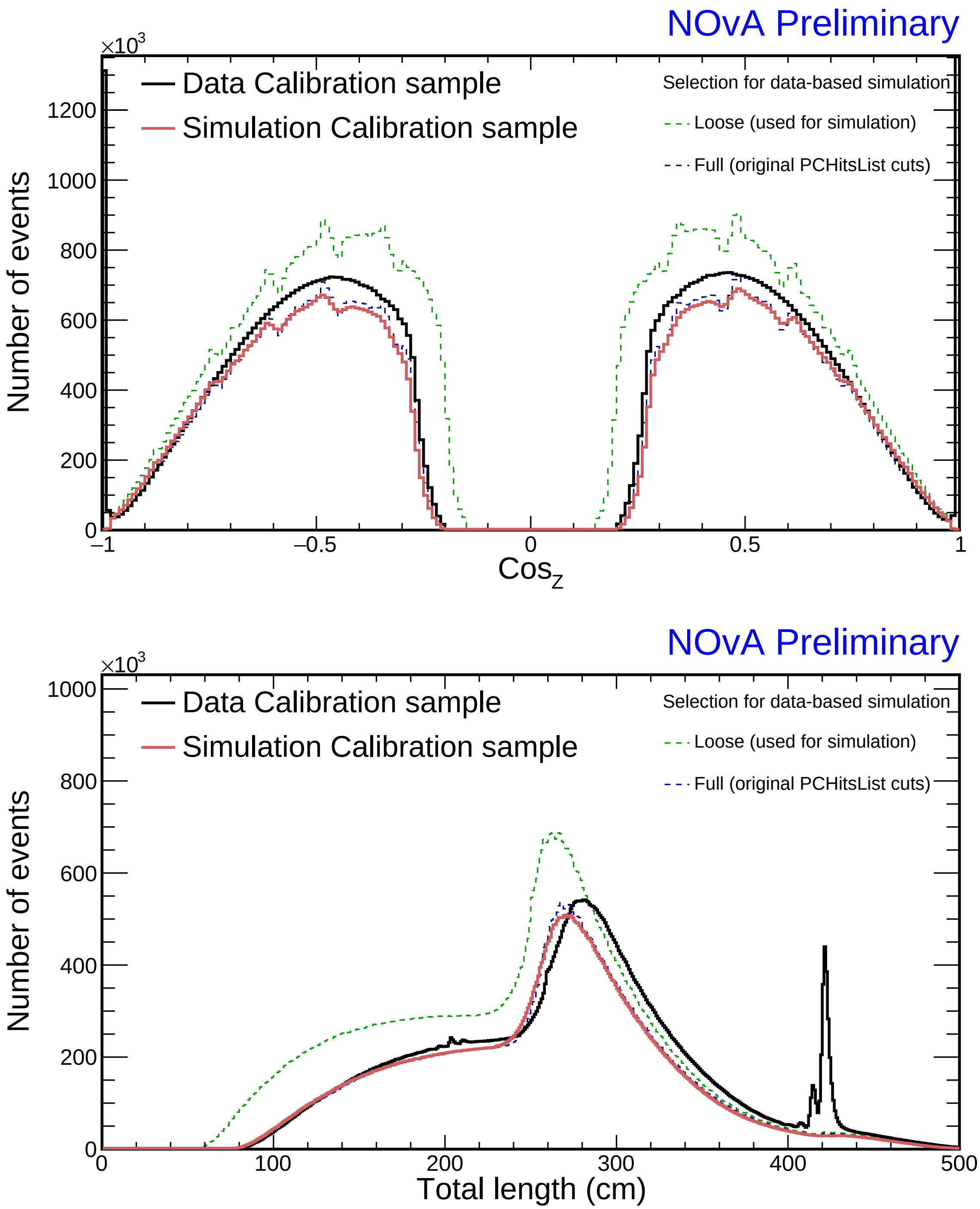


Figure 8: Angular and total track length distribution comparison of the newly created simulation calibration sample and the corresponding data calibration sample. We are also showing the selection of data used to create the new simulation in green and a "full calibration cuts" selection (same cuts as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of Window cosmic tracks) in blue.

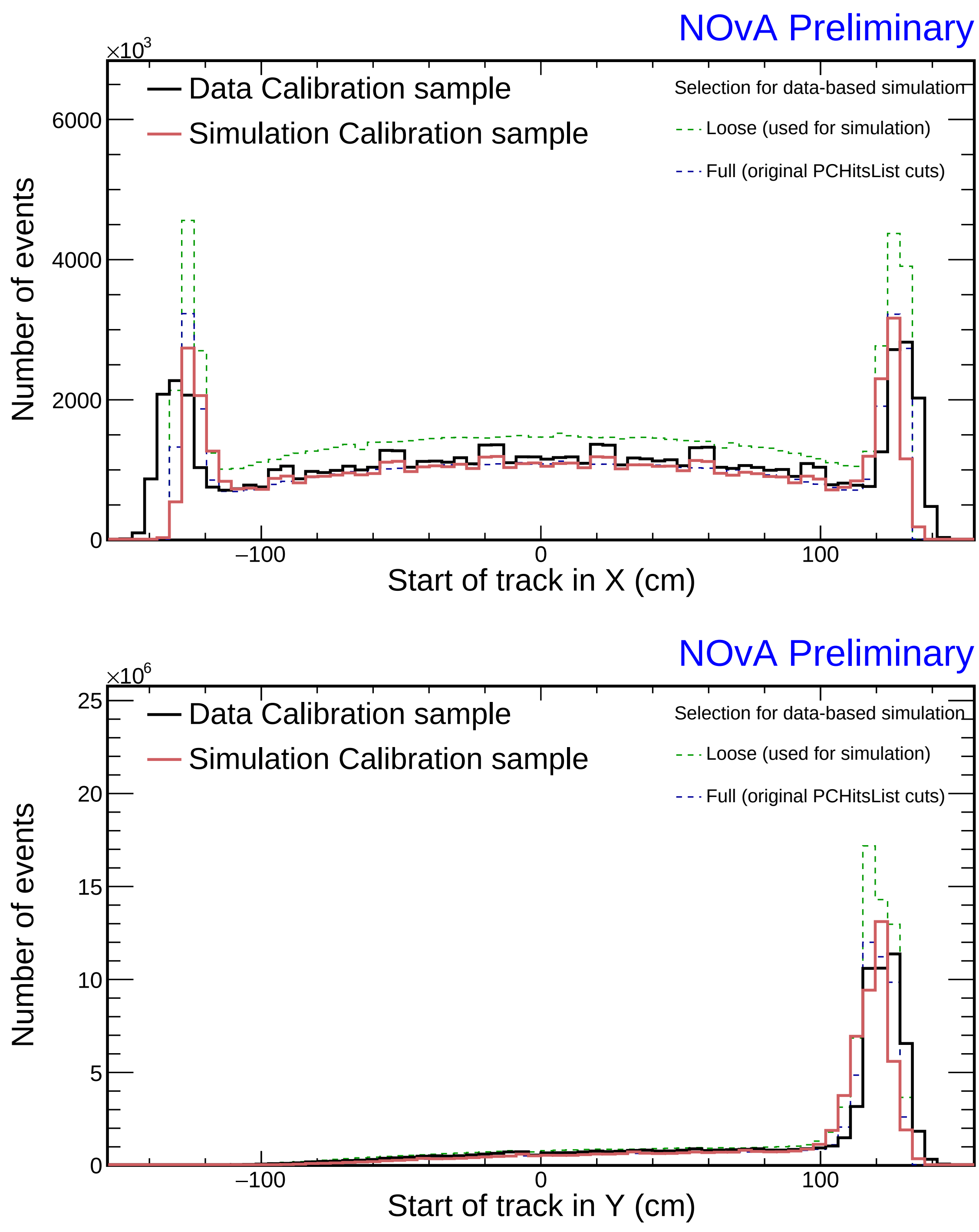


Figure 9: Start of tracks comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the selection of data used to create the new simulation in green and a "full calibration cuts" selection (same cuts as used for the simulation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

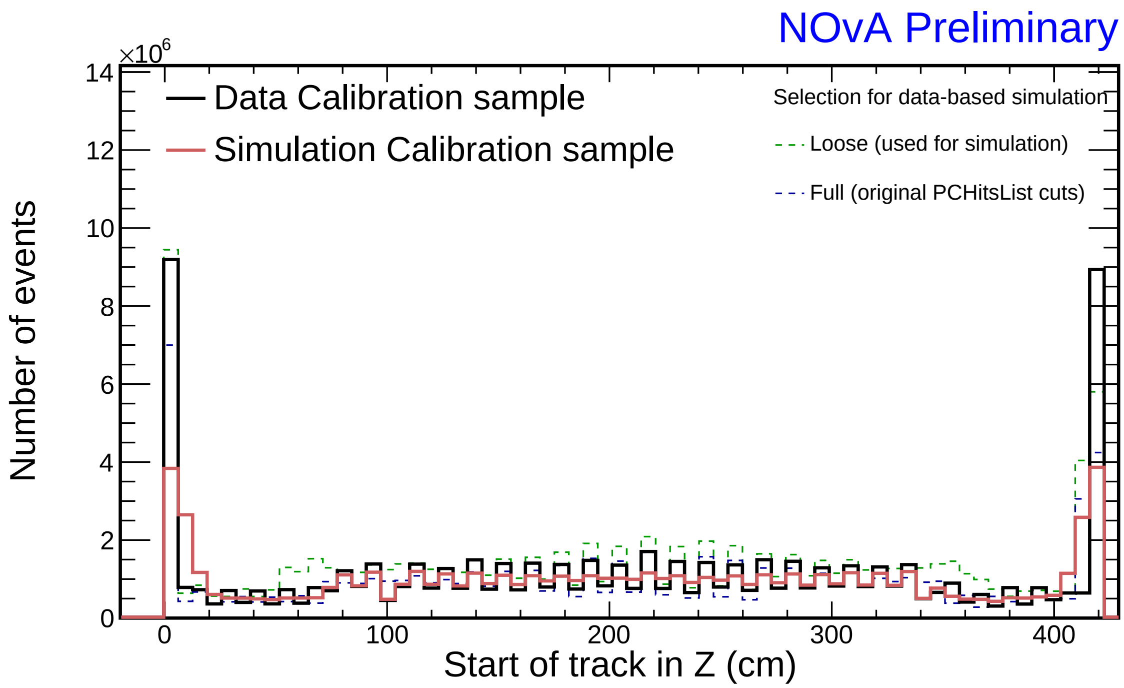


Figure 10: Start of tracks comparison of the newly created simulation calibration sample and the corresponding (period 4) data calibration sample. We are also showing the selection of data used to create the new simulation in green and a "full calibration cuts" selection (same cuts as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of Window cosmic tracks) in blue.

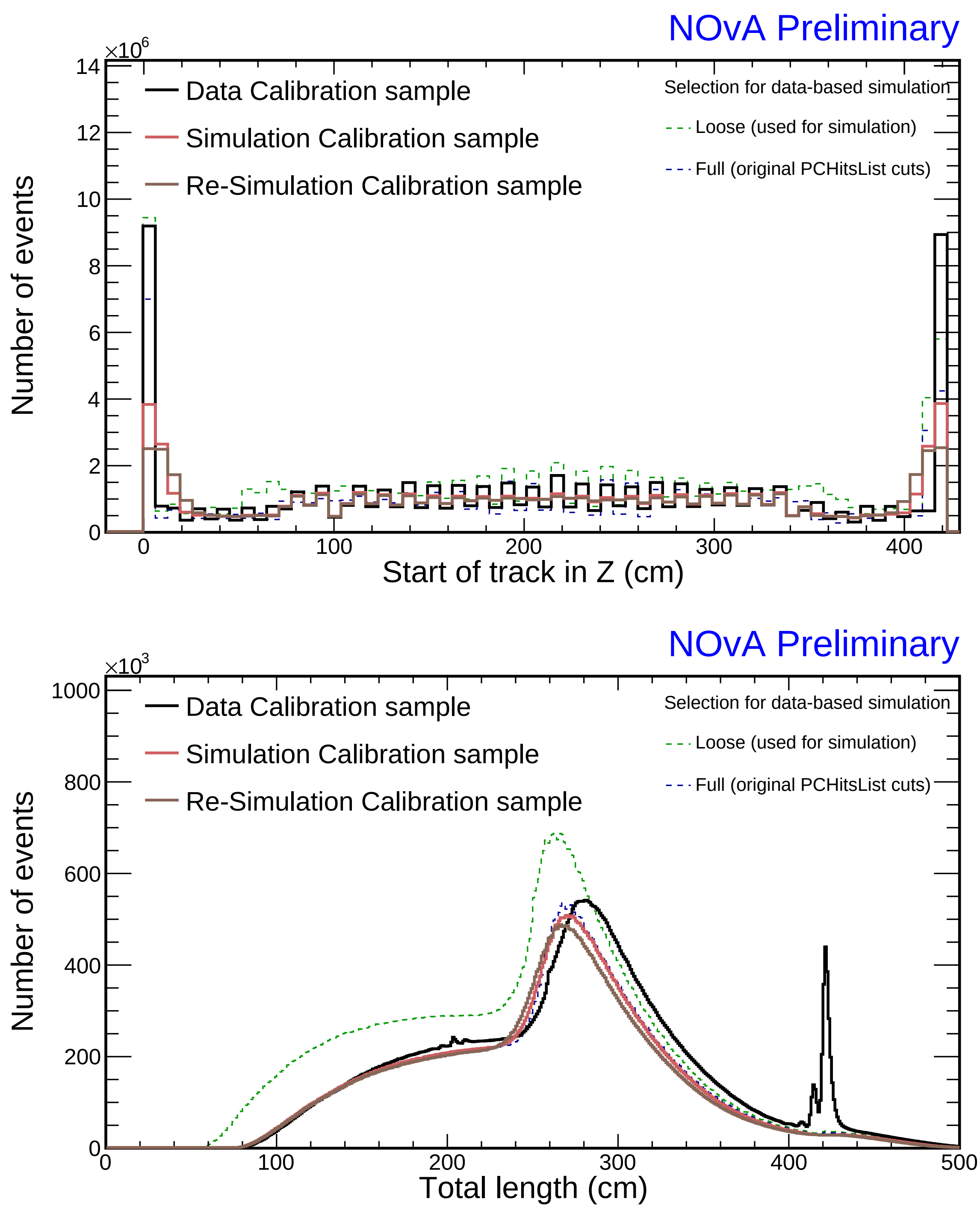


Figure 11: Distribution of the re-simulated events, when the new simulation's artdaq sample is used as "fake data" for a new iteration of the simulation process discussed in this document. This is compared to the newly created simulation calibration sample and the corresponding data calibration sample. We are also showing the selection of data used to create the new simulation in green and a "full calibration cuts" selection (same cuts as used for the simulation calibration sample but applied to Break Point Fitter tracks instead of Window cosmic tracks) in blue.