# Generative AI for BIB Simulation

Radha Mastandrea
rmastand@uchicago.edu
In collaboration with Matt LeBlanc, Shiyu Peng, Ben Rosser
AI4MuC Virtual Mini-Workshop
06/30/2026

# Motivation

# The challenge of beam-induced background

**Beam-induced background** (BIB) provides a large computational challenge for event reconstruction at a future muon collider.

BIB is produced when unstable muons in the beam line decay and interact with the detector and accelerator, leading to a slew (~10^8) of background particles ($\gamma$, $e$, $n$, $\nu$).

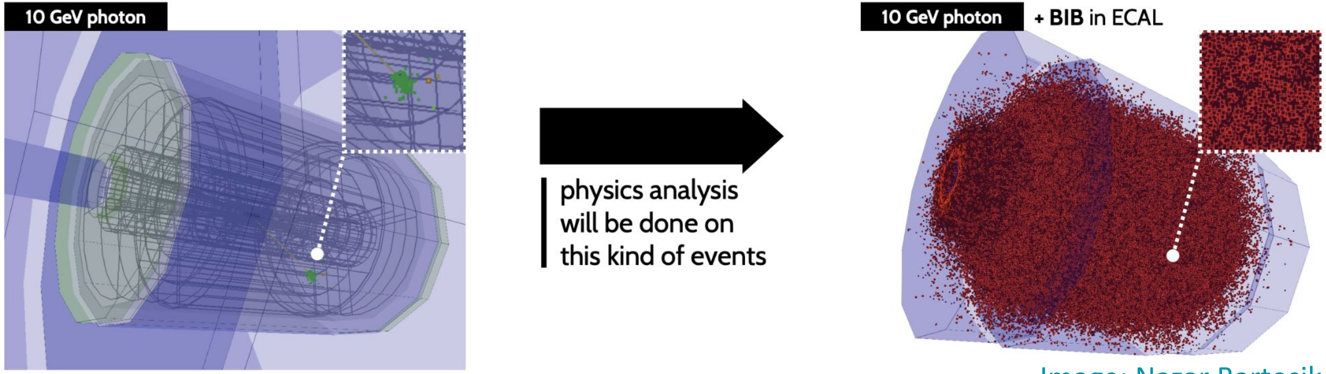



Image: Nazar Bartosik

BIB will overwhelm "interesting" tracks, so it is vital that we understand how BIB will show up in future detectors.

# Generating BIB for muon collider R&D

Understanding what BIB would look like at a future collider requires that we have large amounts of BIB simulation.

Currently, BIB is simulated with FLUKA and GEANT. These simulations are **CPU-intensive**, and we can often only generate a portion (~10%) of a bunch crossing at a time.

Table: Kevin Pedro

| Step | CPU | Memory | Disk |
|---|---|---|---|
| BIB simulation | up to 24 hours/event ($10^8$ particles) | | ~20 GB/event (BIB) |
| BIB overlay | 5 mins/event (before digitization!) | up to 32 GB/event (considering whole chain) | |
| Tracking | 5 mins/event up to hours/event (depends on lattice) | | ~1 MB/event (signal, w/o BIB) |

Geant alone!

# Generating BIB for muon collider R&D

Understanding what BIB would look like at a future collider requires that we have large amounts of BIB simulation.

Currently, BIB is simulated with FLUKA and GEANT. These simulations are **CPU-intensive**, and we can often only generate a portion (~10%) of a bunch crossing at a time.

Table: Kevin Pedro

| Step | CPU | Memory | Disk |
|---|---|---|---|
| BIB simulation | up to 24 hours/event ($10^8$ particles) | | ~20 GB/event (BIB) |
| BIB overlay | 5 mins/event (before digitization!) | up to 32 GB/event (considering whole chain) | |
| Tracking | 5 mins/event up to hours/event (depends on lattice) | | ~1 MB/event (signal, w/o BIB) |

Geant alone!

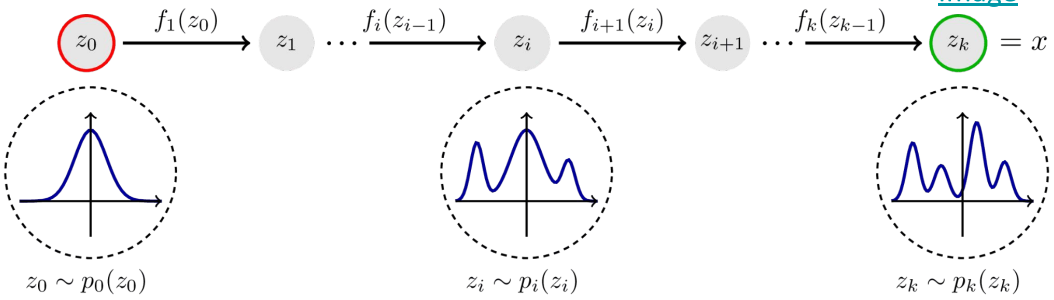
Given that we already have a small number of BIB events, we can use them to create a **fast BIB generator** by training generative machine learning networks.

# Methodology for
# ML BIB generation

# Baseline architecture: normalizing flows

- Normalizing flows are expressive neural networks that can be used to model high-dimensional probability densities
- The flow learns a series of invertible transformations $f_i$ from a standard normal distribution $z_0 \sim N(0,1)^D$ to the training data distribution $p_k$

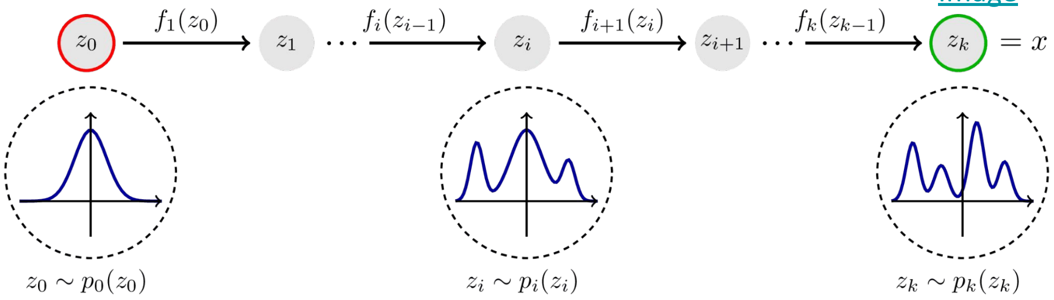
Image



# Baseline architecture: normalizing flows

- Normalizing flows are expressive neural networks that can be used to model high-dimensional probability densities
- The flow learns a series of invertible transformations $f_i$ from a standard normal distribution $z_0 \sim N(0,1)^D$ to the training data distribution $p_k$

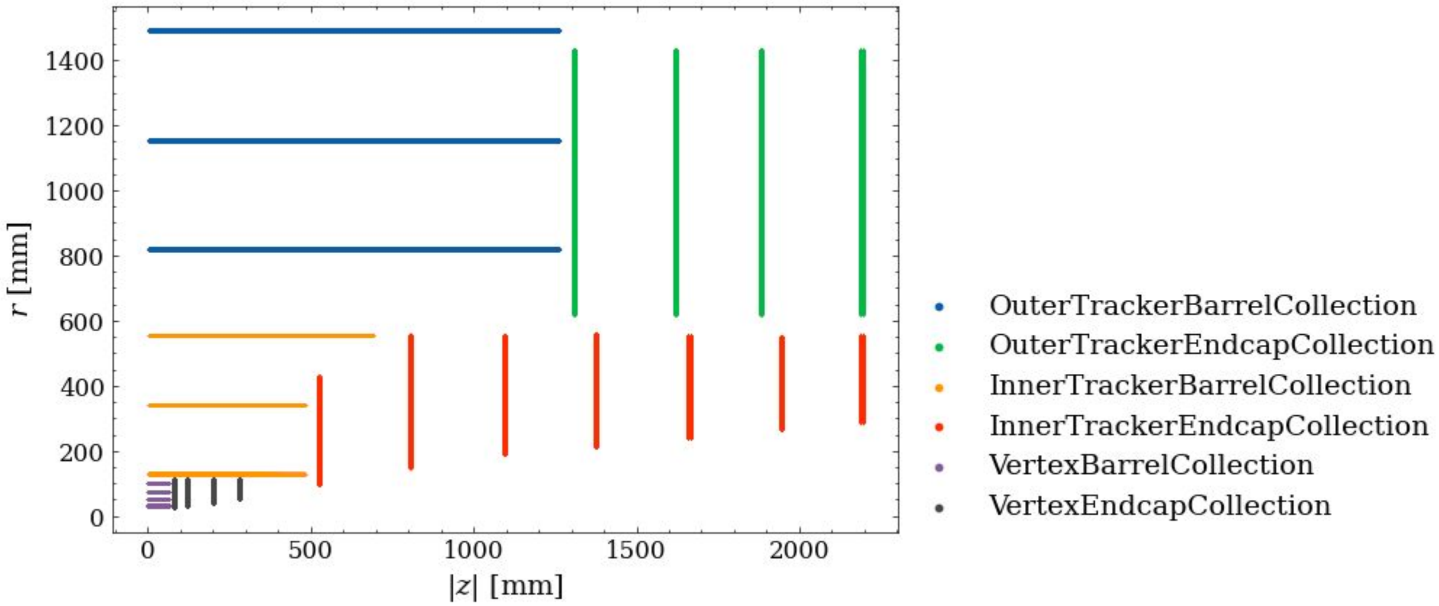
Image




- We can **quickly draw samples** from $p_k$ by sending standard normal samples $z_0$ through the flow — without having a closed form expression for $p_k$!
- We use a Neural Circular Spline Flow (NCSF) architecture [stat/2002.02428] which is good for observables on a circular domain (e.g. $\boldsymbol{\varphi}$)

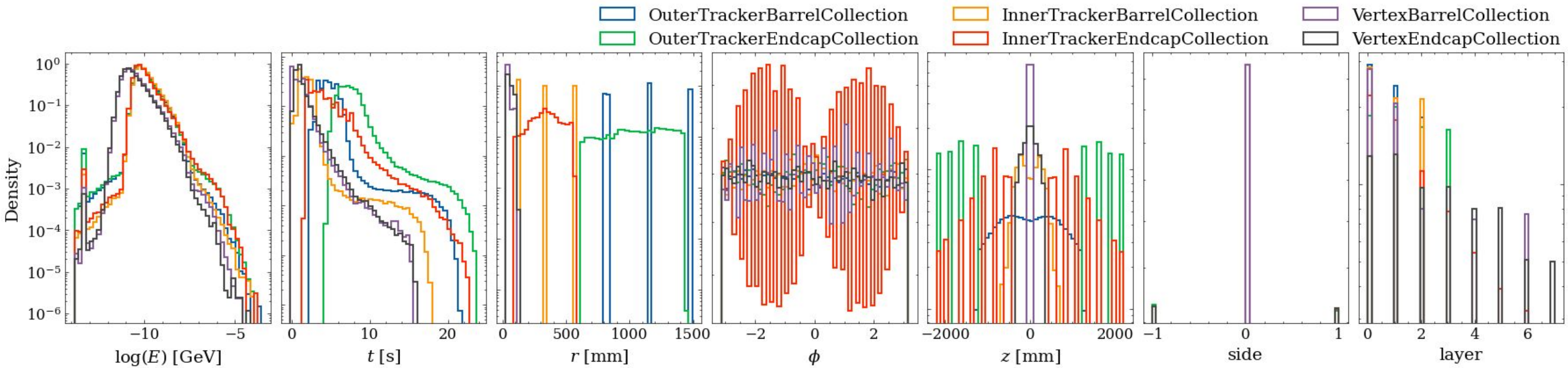
# Sim BIB training set: pre-digitized tracker hits

- We use 1 event from the latest MAIA BIB simulation (stored on the OSG)
  - $\nu$ gun samples with $p_T \in [0, 50]$ GeV
  - **MAIA detector** geometry [physics.ins-det/2502.00181] with 6 tracking subsystems



# Sim BIB training set: pre-digitized tracker hits

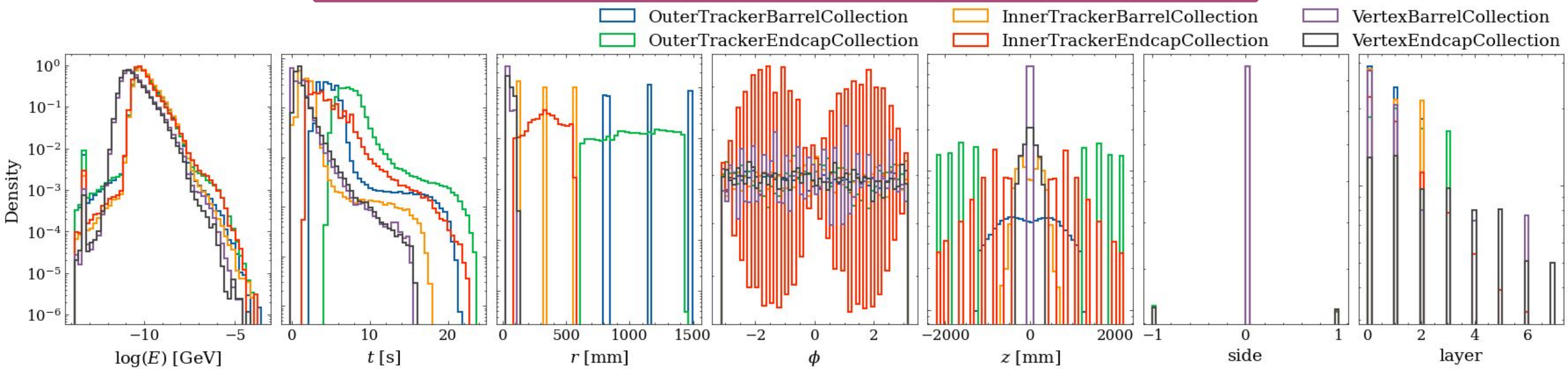
- We use 1 event from the latest MAIA BIB simulation (stored on the OSG)
  - $\nu$ gun samples with $p_T \in [0, 50]$ GeV
  - MAIA detector geometry [physics.ins-det/2502.00181] with 6 subsystems
- We train a separate flow model for each of the 6 tracker subsystems
- We train a NCSF to model 5 features: *log(E), t, r*, $\boldsymbol{\varphi}$, *z* for `SimTrackerHits` (i.e. pre-digitization)
- We condition the flows on the side and layer integer associated with each BIB hit



# Sim BIB training set: pre-digitized tracker hits

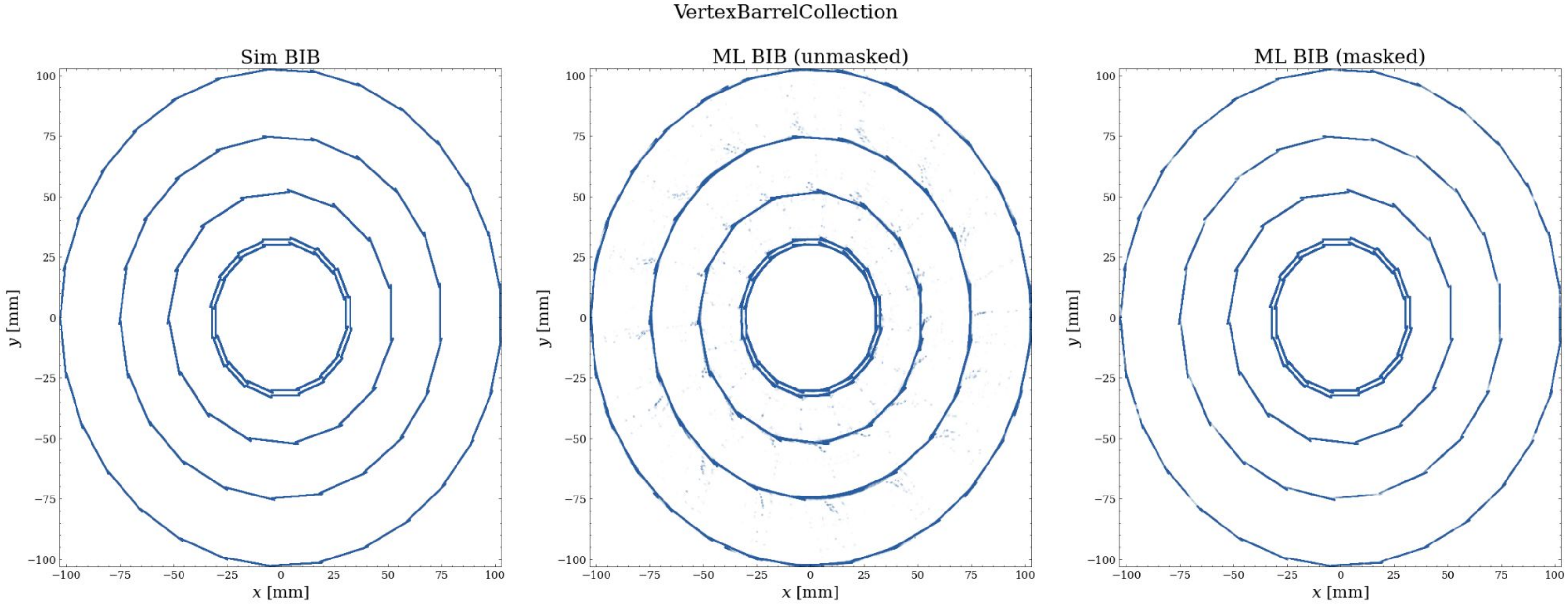
- We use 1 event from the latest MAIA BIB simulation (stored on the OSG)
  - $\nu$ gun samples with $p_T \in [0, 50]$ GeV
  - MAIA detector geometry [physics.ins-det/2502.00181] with 6 subsystems
- We train a se
- We train a N( pre-digitization)
- We condition hit

Due to a bug in its generation, the Sim BIB training set has a geometry where the sensitive layers of the detectors are too thin by several orders of magnitude



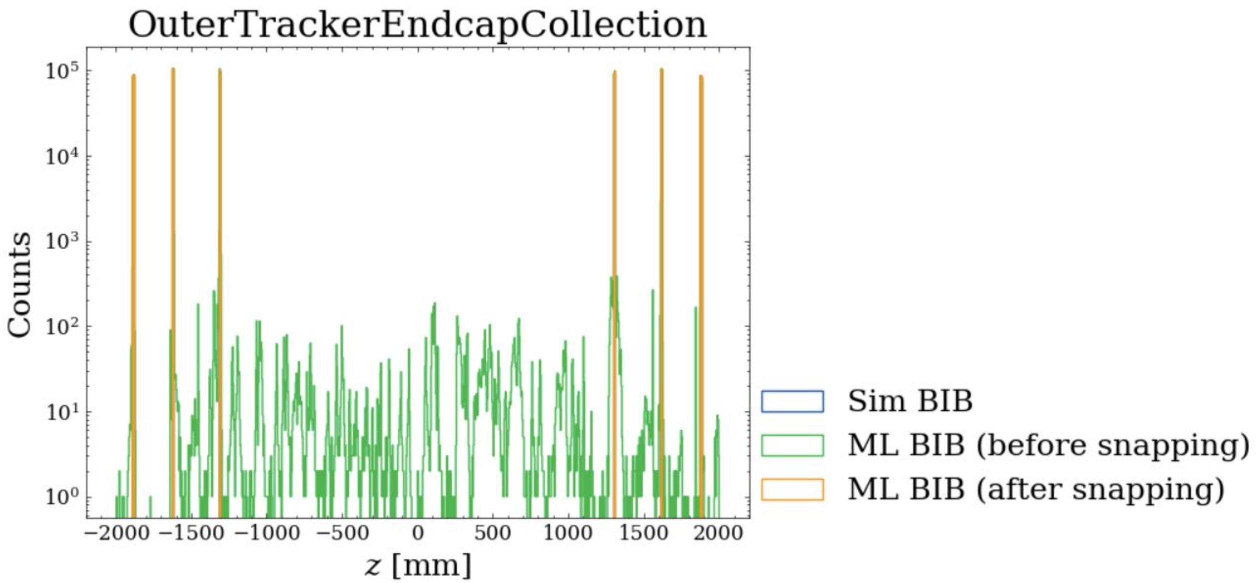
# [SKIP FOR TIME] ML BIB processing: detector geometry mask

- After we generate BIB samples from the flow, we apply a **detector-based geometry mask** to reject off-module hits
- The mask primarily targets off-layer hits in the barrel subsystems
  - Pass rate is >99% for the endcaps and 45-80% in the barrels



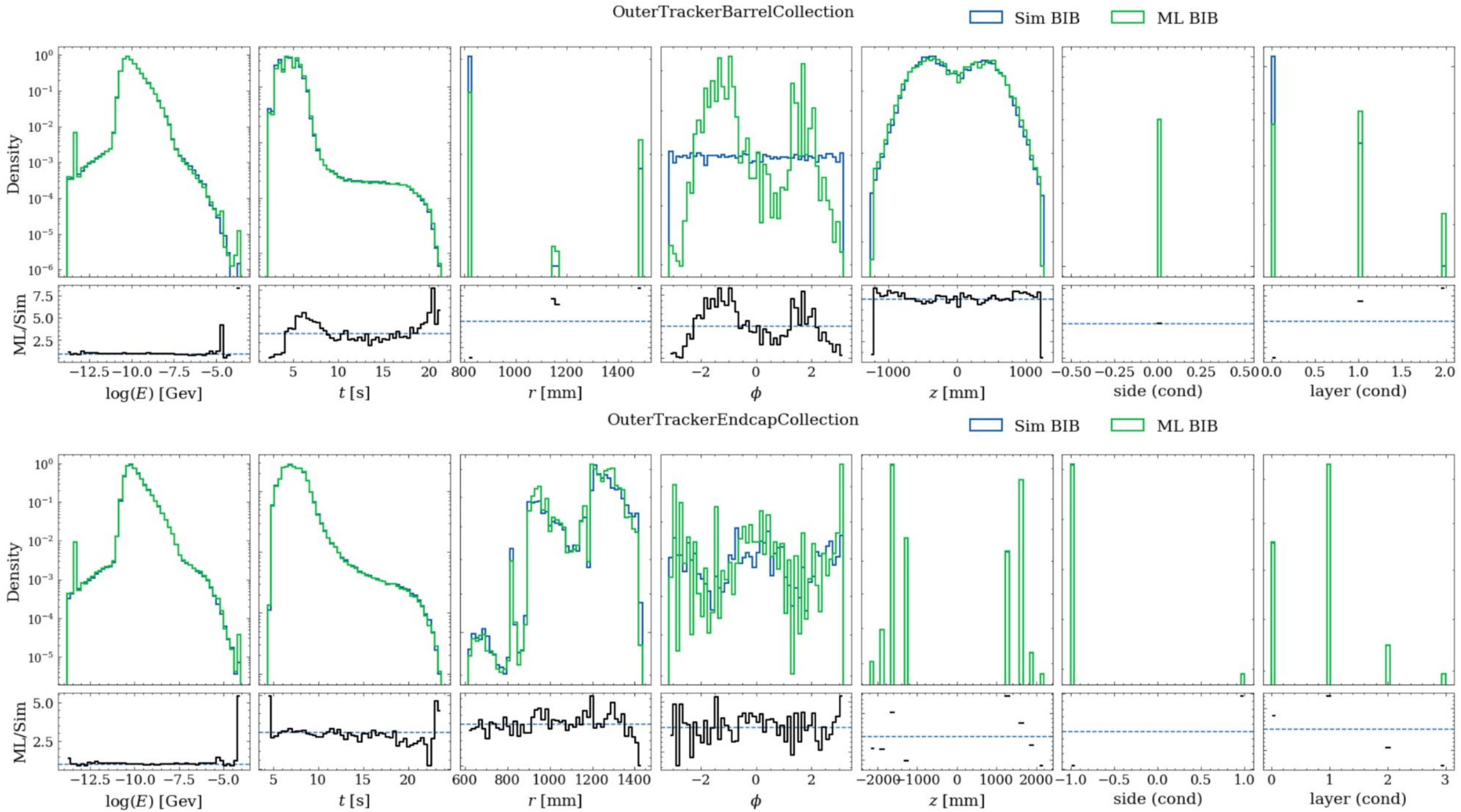
# [SKIP FOR TIME] ML BIB processing: endcap *z*-snapping

- We snap the endcap subsystem hits to the *z*-position of the closest valid detector subcomponent
  - Flow models struggle with modeling delta functions (and the Sim BIB endcaps are already artificially too thin), so this snapping increases the mask pass rate without negatively affecting the ML BIB

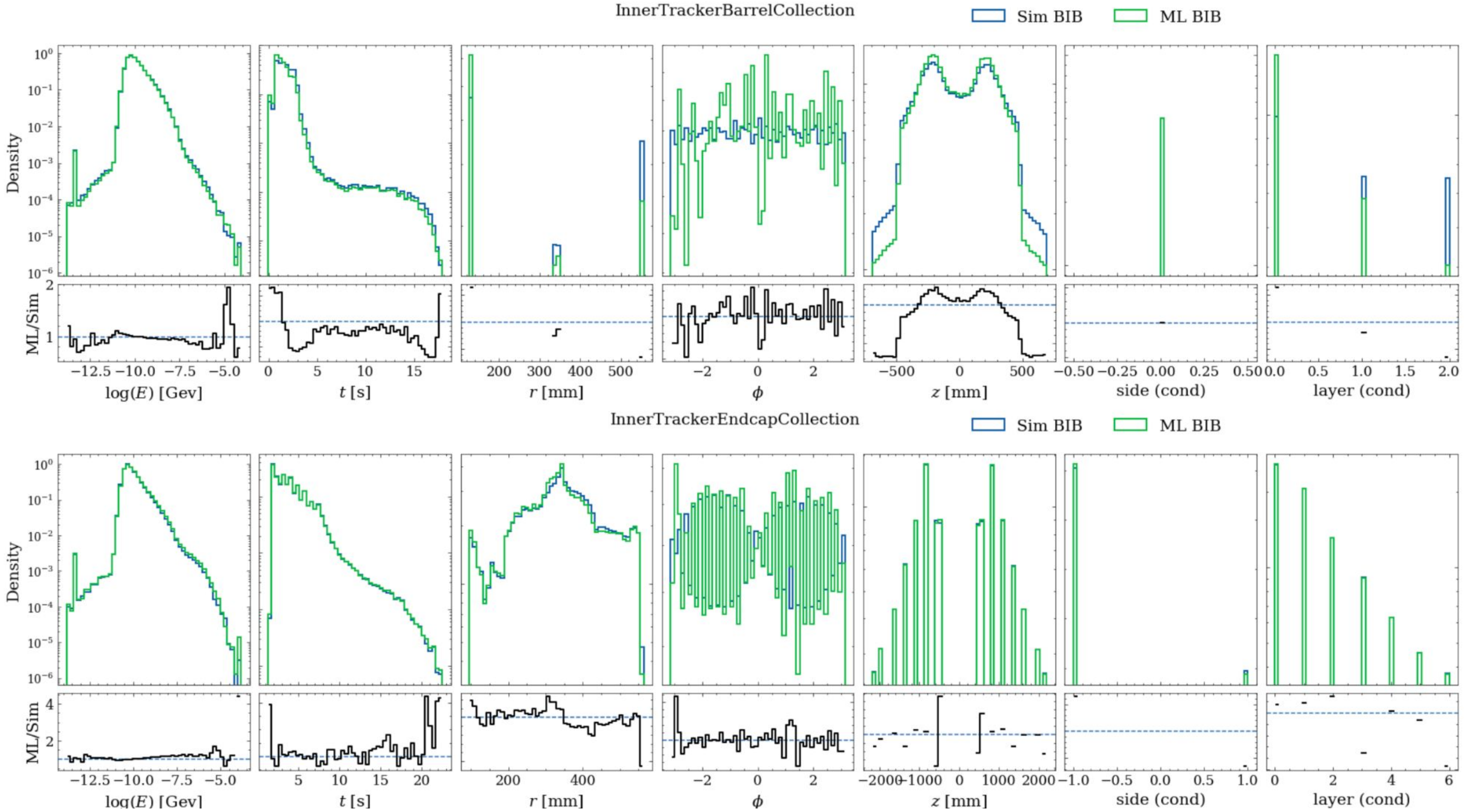


# Evaluation of the ML BIB

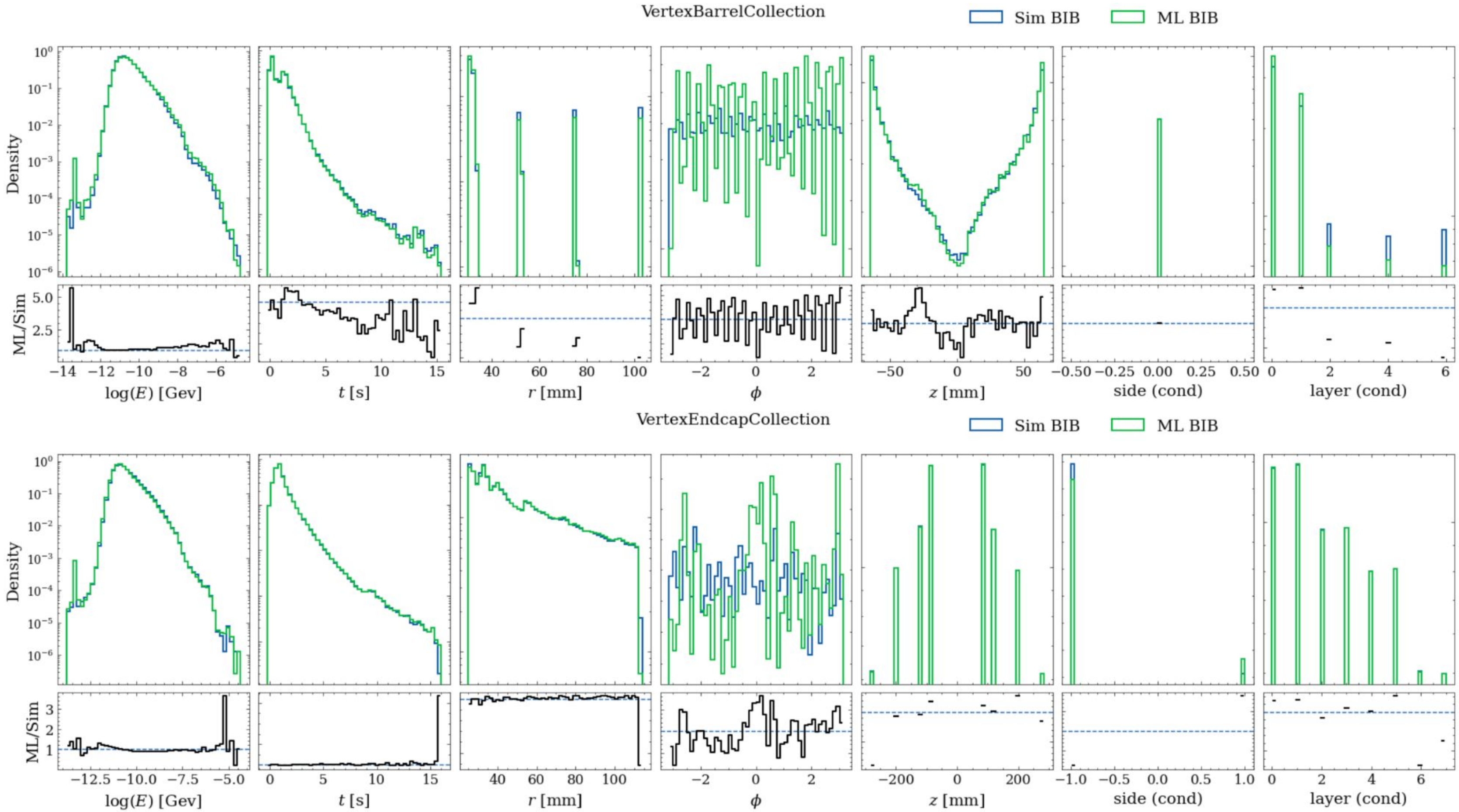
# Comparison of ML BIB to Sim BIB: 1D histograms



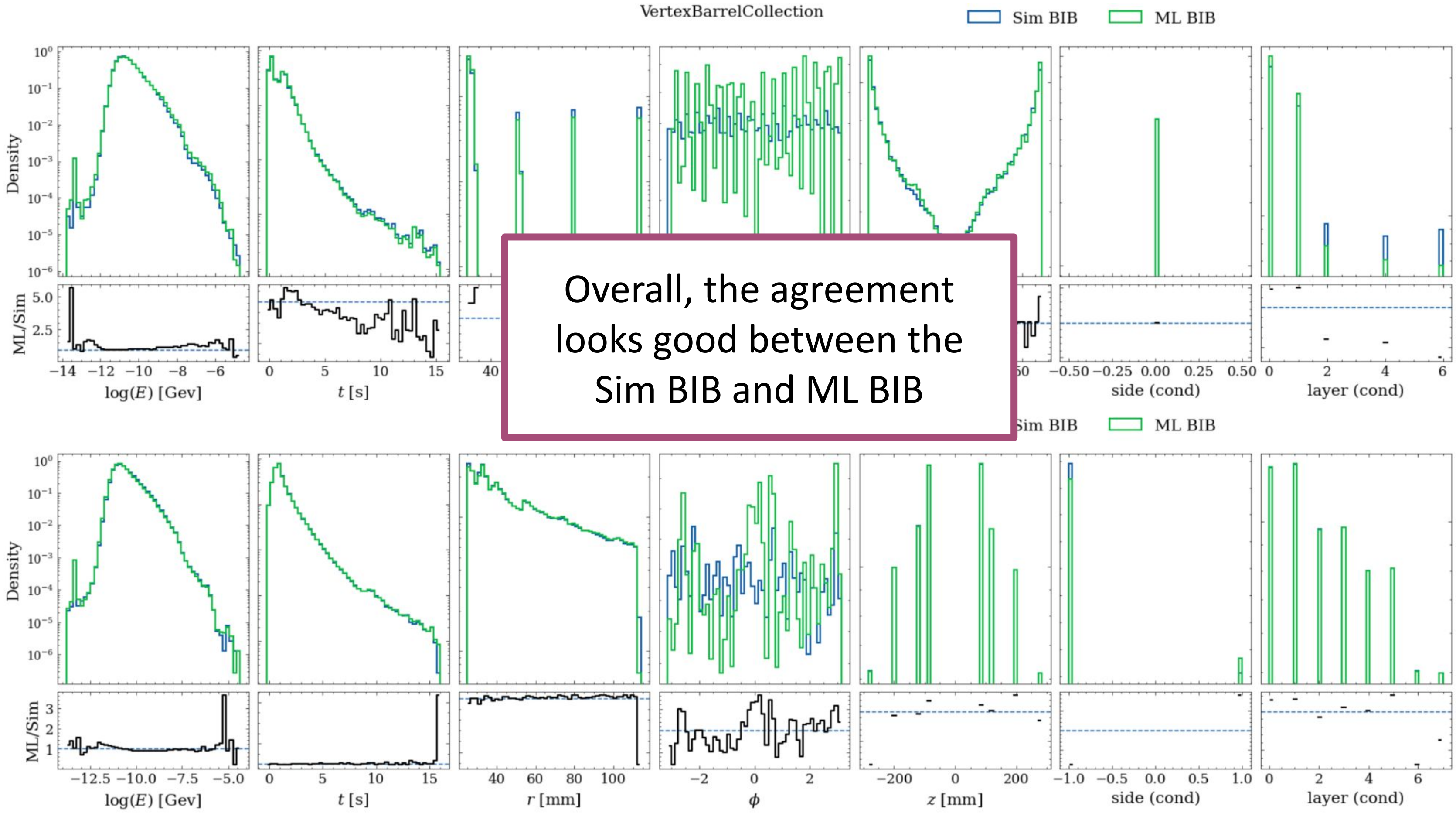
# Comparison of ML BIB to Sim BIB: 1D histograms



# Comparison of ML BIB to Sim BIB: 1D histograms



# Comparison of ML BIB to Sim BIB: 1D histograms




Overall, the agreement looks good between the Sim BIB and ML BIB

# Comparison of ML BIB to Sim BIB: discrimination test

- We train a simple binary classifier (BDT) to distinguish Sim BIB from ML BIB
- Ideally the classifier would fail to distinguish the two sets (ROC AUC = 0.5)

------------------ BDT feature importances ------------------

| | ROC AUC | *log(E)* | *t* | *r* | ***φ*** | *z* |
|---|---|---|---|---|---|---|
| Outer Tracker Barrel | 0.706 | 0.0151 | 0.0226 | 0.8131 | 0.0979 | 0.0513 |
| Inner Tracker Barrel | 0.636 | 0.0476 | 0.0610 | 0.7025 | 0.1524 | 0.0365 |
| Vertex Barrel | 0.627 | 0.0858 | 0.0496 | 0.5963 | 0.2334 | 0.0349 |
| Outer Tracker Endcap | 0.533 | 0.2133 | 0.1827 | 0.1509 | 0.1486 | 0.3044 |
| Inner Tracker Endcap | 0.530 | 0.1417 | 0.2184 | 0.2059 | 0.2171 | 0.2168 |
| Vertex Endcap | 0.538 | 0.1943 | 0.1272 | 0.2010 | 0.1402 | 0.3373 |
| Overall | 0.550 | 0.1117 | 0.1614 | 0.4166 | 0.1590 | 0.1514 |

# Comparison of ML BIB to Sim BIB: discrimination test

- We train a simple binary classifier (BDT) to distinguish Sim BIB from ML BIB
- Ideally the classifier would fail to distinguish the two sets (ROC AUC = 0.5)

----------------- BDT feature importances -----------------

| | ROC AUC | *log(E)* | *t* | *r* | ***φ*** | *z* |
|---|---|---|---|---|---|---|
| Outer Tracker Barrel | 0.706 | 0.0151 | 0.0226 | 0.8131 | 0.0079 | 0.0513 |
| Inner Tracker Barrel | 0.636 | 0.0476 | [illegible] | [illegible] | [illegible] | [illegible] |
| Vertex Barrel | 0.627 | 0.0858 | [illegible] | [illegible] | [illegible] | [illegible] |
| **Outer Tracker Endcap** | **0.533** | **0.2133** | **0.1827** | **0.1509** | **0.1486** | **0.3044** |
| **Inner Tracker Endcap** | **0.530** | **0.1417** | **0.2184** | **0.2059** | **0.2171** | **0.2168** |
| **Vertex Endcap** | **0.538** | **0.1943** | **0.1272** | **0.2010** | **0.1402** | **0.3373** |
| Overall | 0.550 | 0.1117 | 0.1614 | 0.4166 | 0.1590 | 0.1514 |

The endcaps look good, with minor *z* mismodeling

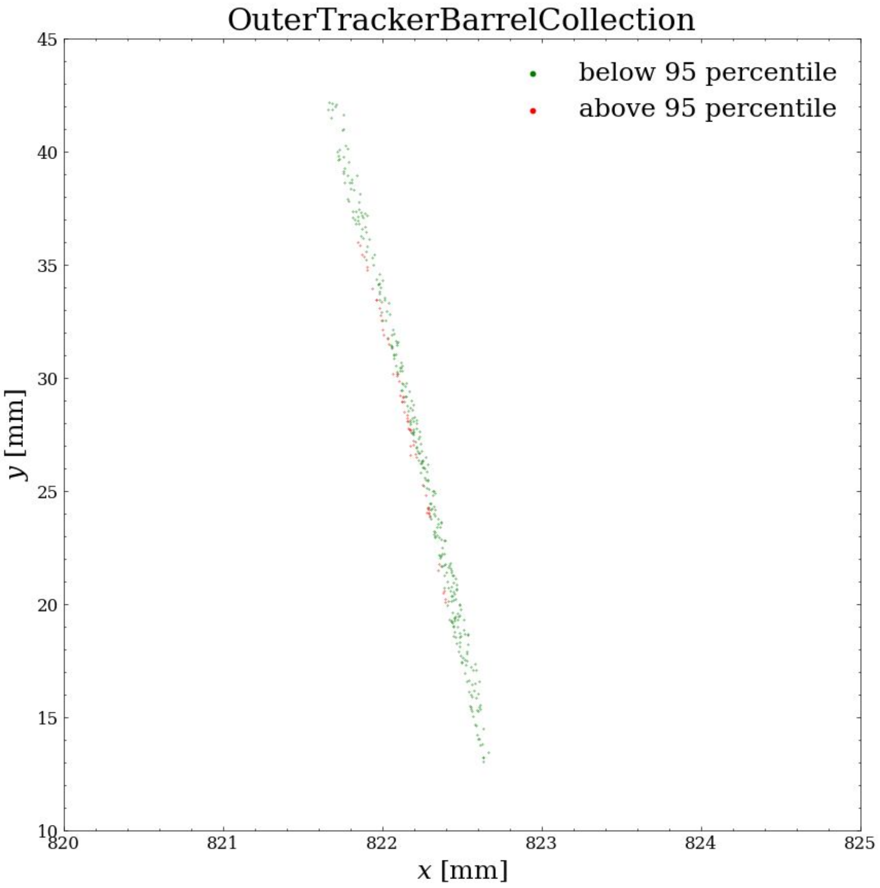
# Comparison of ML BIB to Sim BIB: discrimination test

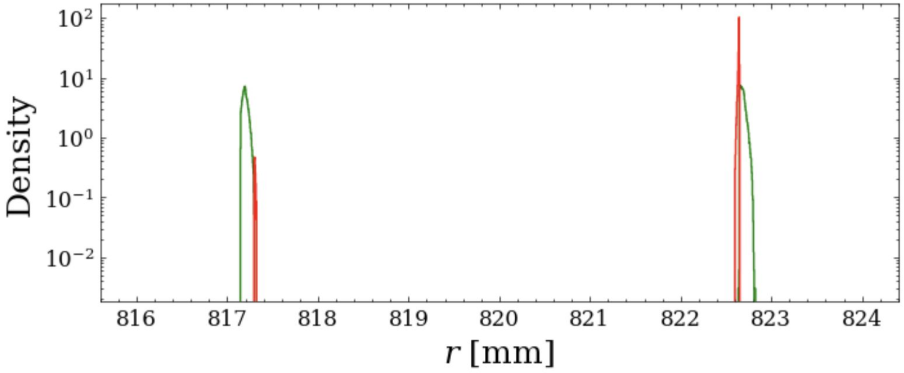
- We train a simple binary classifier (BDT) to distinguish Sim BIB from ML BIB
- Ideally the classifier would fail to distinguish the two sets (ROC AUC = 0.5)

| | ROC AUC | ----------------- BDT feature importances ----------------- | | | | |
|---|---|---|---|---|---|---|
| | ROC AUC | *log(E)* | *t* | *r* | $\varphi$ | *z* |
| **Outer Tracker Barrel** | **0.706** | **0.0151** | **0.0226** | **0.8131** | **0.0979** | **0.0513** |
| **Inner Tracker Barrel** | **0.636** | **0.0476** | **0.0610** | **0.7025** | **0.1524** | **0.0365** |
| **Vertex Barrel** | **0.627** | **0.0858** | **0.0496** | **0.5963** | **0.2334** | **0.0349** |
| Outer Tracker Endcap | 0.533 | [illegible] | [illegible] | [illegible] | [illegible] | 0.3044 |
| Inner Tracker Endcap | 0.530 | [illegible] | [illegible] | [illegible] | [illegible] | 0.2168 |
| Vertex Endcap | 0.538 | [illegible] | [illegible] | [illegible] | [illegible] | 0.3373 |
| Overall | 0.550 | 0.1117 | 0.1614 | 0.4166 | 0.1590 | 0.1514 |

The barrels suffer from sizable *r* mismodeling

# [SKIP FOR TIME] Why is *r* so poorly modeled in the barrels?




The most “anomalous” (i.e. non Sim BIB-like) hits are at the edges of the sensitive layers of the barrel modules

# Comparison of ML BIB to Sim BIB: discrimination test

- We train a simple binary classifier (BDT) to distinguish Sim BIB from ML BIB
- Ideally the classifier would fail to distinguish the two sets (ROC AUC = 0.5)

------------------ BDT feature importances ------------------

| | ROC AUC | *log(E)* | *t* | *r* | ***φ*** | *z* |
|---|---|---|---|---|---|---|
| Outer Tracker Barrel | 0.706 | 0.0151 | 0.0226 | 0.8131 | 0.0979 | 0.0513 |
| Inner Tracker Barrel | 0.636 | 0.0476 | 0.0610 | 0.7025 | 0.1524 | 0.0365 |
| Vertex Barrel | 0.627 | 0.0858 | 0.0496 | 0.5963 | 0.2334 | 0.0349 |
| Outer Tracker Endcap | 0.533 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Inner Tracker Endcap | 0.530 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Vertex Endcap | 0.538 | 0.1943 | 0.1272 | 0.2010 | 0.1402 | 0.3373 |
| **Overall** | **0.550** | **0.1117** | **0.1614** | **0.4166** | **0.1590** | **0.1514** |

Overall, the ML BIB is quantitatively similar to the Sim BIB

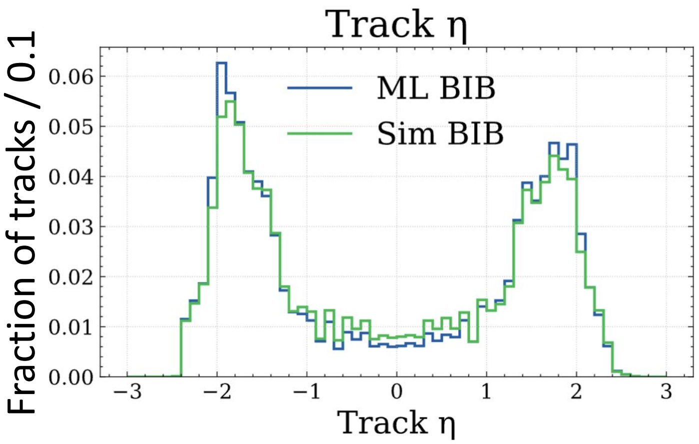
# Comparison of ML BIB to Sim BIB: track observables

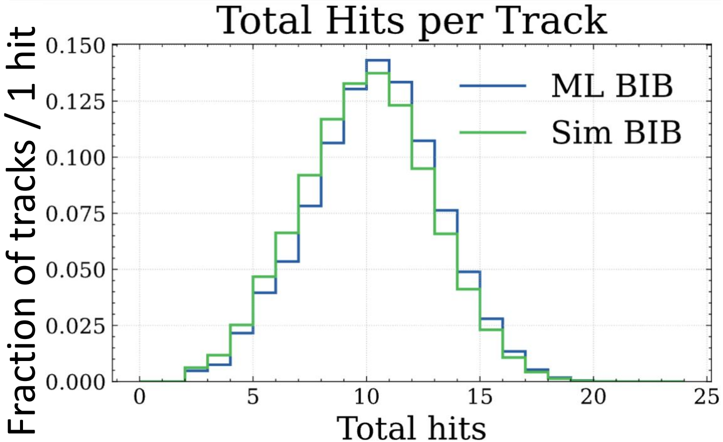
- A more important test of the ML BIB is a phenomenological one – to compare ML BIB tracks with Sim BIB tracks

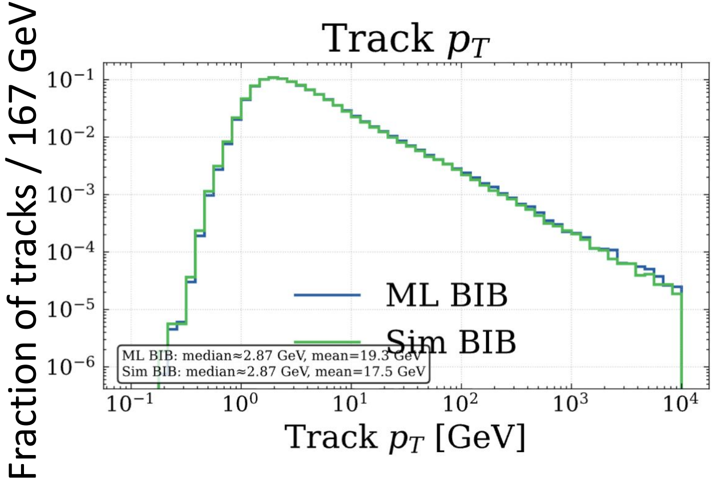
- We compare digitized and reconstructed ML BIB to Sim BIB (both at 100%)
- We generate ML BIB such that it has the same number of hits in each tracking detector as the Sim BIB

- We find:
  - 1_329_032 ML BIB tracks (`SiTrackDeduped`)
  - 1_069_261 Sim BIB tracks (`SiTrackDeduped`)

# Comparison of ML BIB to Sim BIB: track observables

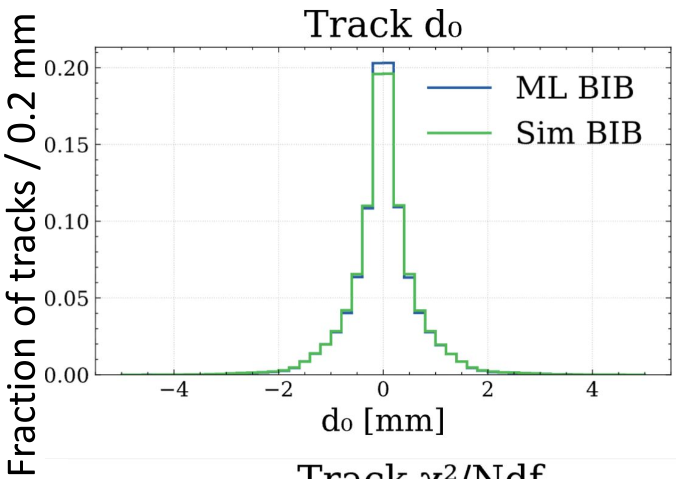


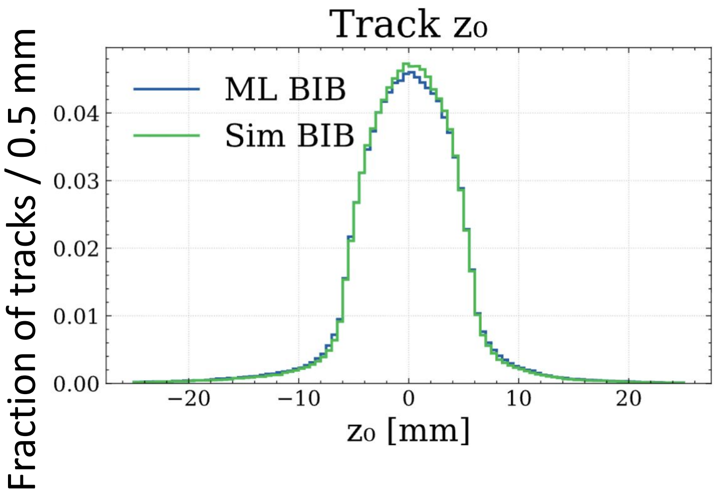



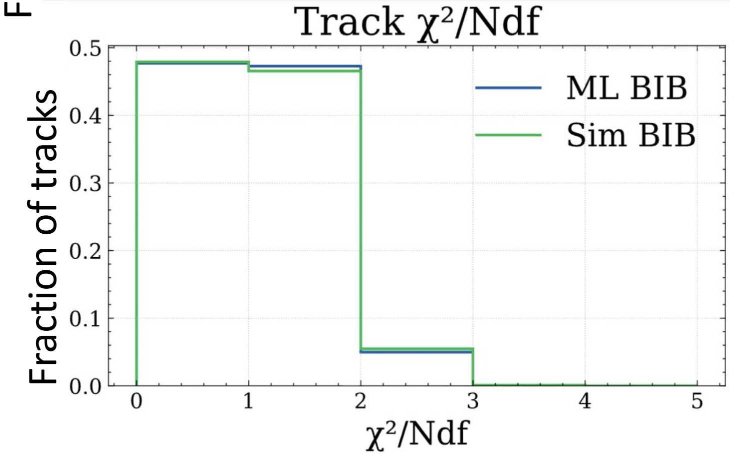




Overall we see good agreement in track kinematic observables, with ML bib having slightly more hits per track.

# Comparison of ML BIB to Sim BIB: track observables




Overall we see good agreement in track fitting parameters.

# Conclusions and next steps

We have developed a generative machine learning model that allows us to **quickly generate ML BIB** (pre-digitized tracker) hits that are qualitatively and phenomenologically **similar to Sim BIB**.

Next steps

- Targeting and minimizing the differences between ML BIB and Sim BIB at the hit and track level
- Exploring other network architectures (we are currently working on a tabular diffusion model, led by Shiyu Peng at Brown)
- Developing a model for post-digitized hits
- Developing a model for calorimeter hits

# Backup slides

# Number of training samples and material mask pass rates

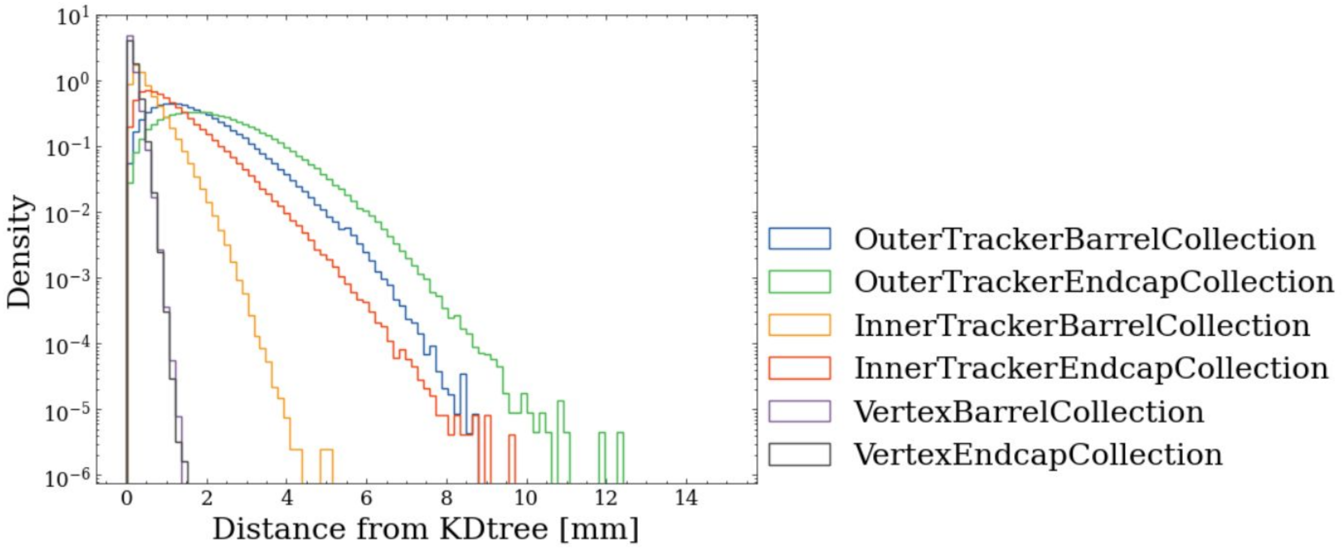
| | Number of training samples | Mask pass rate |
|---|---|---|
| Outer Tracker Barrel Collection | 3_280_678 | 47.0 |
| Outer Tracker Endcap Collection | 1_465_825 | 99.8 |
| Inner Tracker Barrel Collection | 3_926_196 | 68.2 |
| Inner Tracker Endcap Collection | 1_614_679 | 99.6 |
| Vertex Barrel Collection | 2_129_166 | 78.6 |
| Vertex Endcap Collection | 4_096_371 | 99.2 |

# Network training specifications

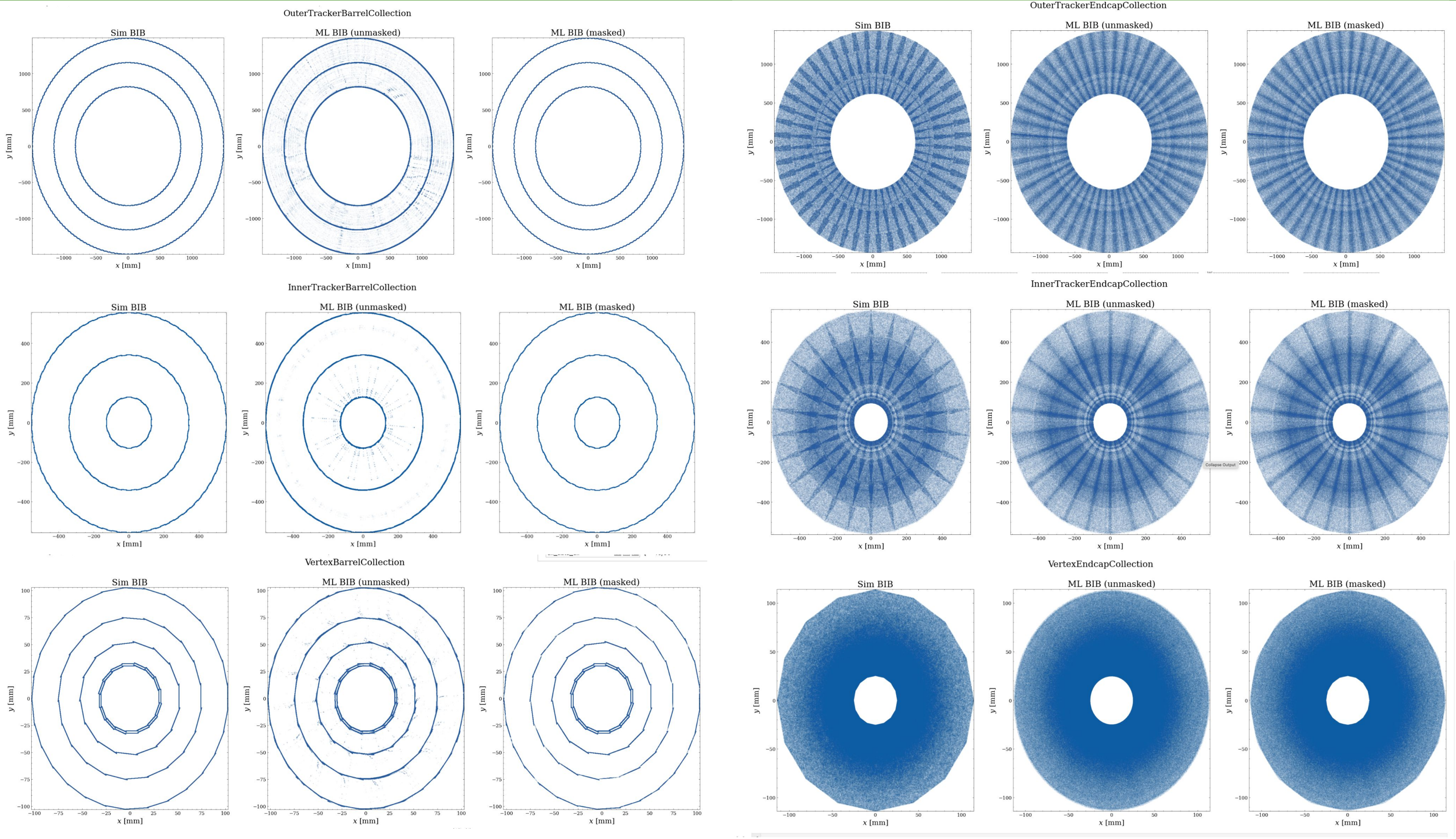
- NCSF with 5 transform blocks, each with 3 layers with 128 hidden features
- Network is implemented with the zuko package in PyTorch
- We train for 500 epochs with a batch size of 2048 and a learning rate of 1e-4 (ADAM optimizer)

# ML BIB processing: cell ID assignment

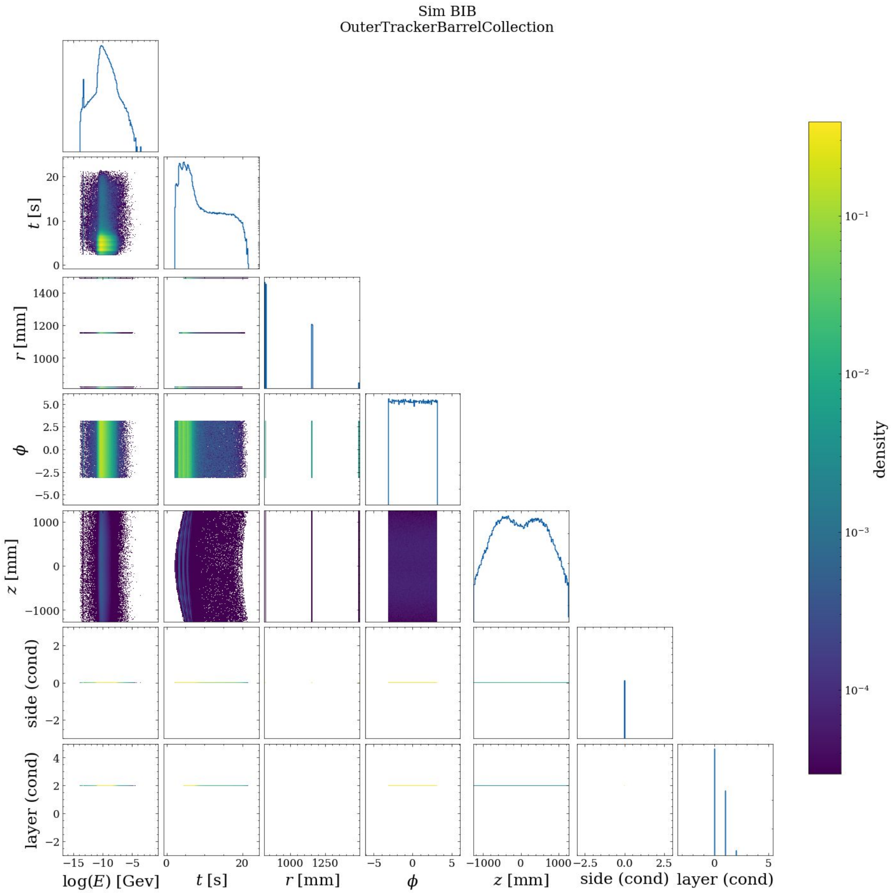
- ML BIB hits need a cell ID for digitization and reconstruction
- Ideally we could use a function that maps from $(x, y, z) \longrightarrow$ cell ID
  - This function seems to be implemented only for the calorimeter for LCIO processing
- Instead, we define a `scipy` KDTree in $(x, y, z)$ from the Sim BIB, and we assign ML BIB hits the cell ID of the closest Sim BIB hit

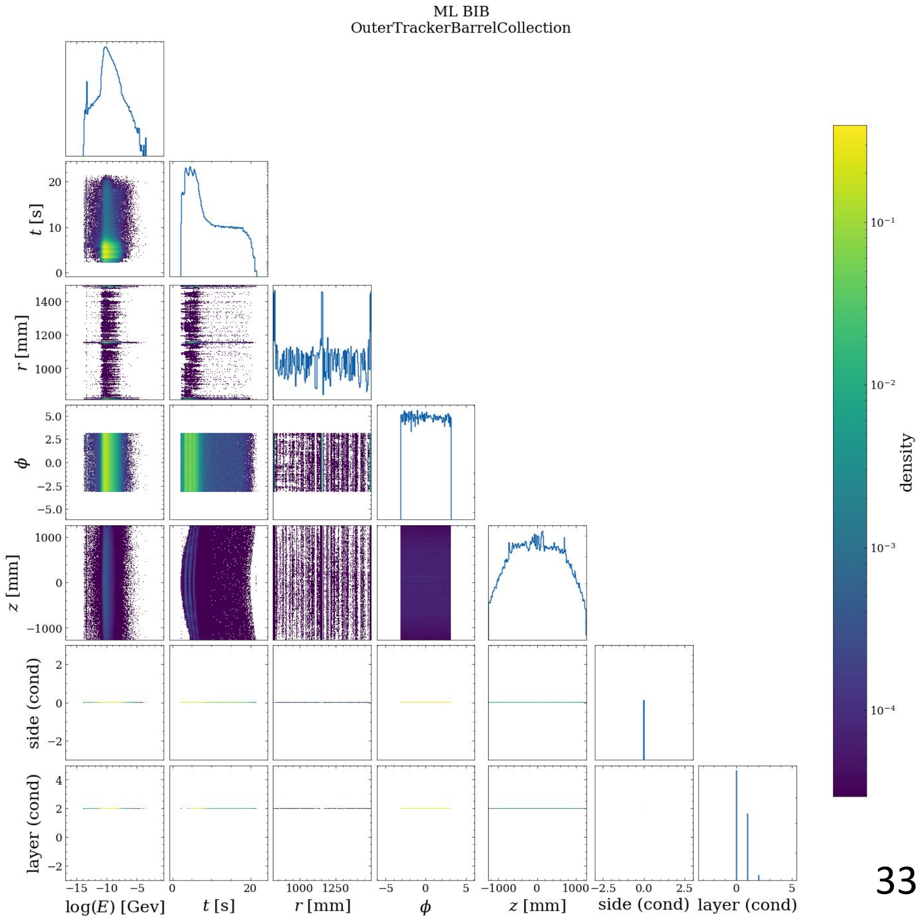


# Effect of masking in the *x*-*y* plane

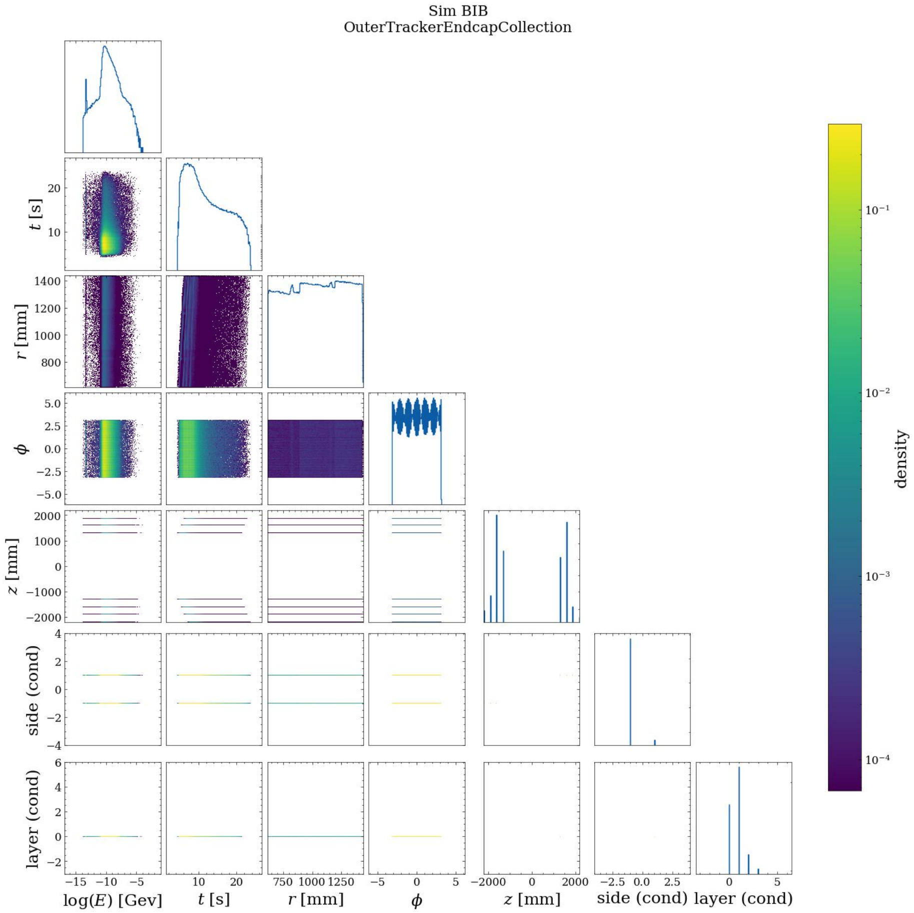


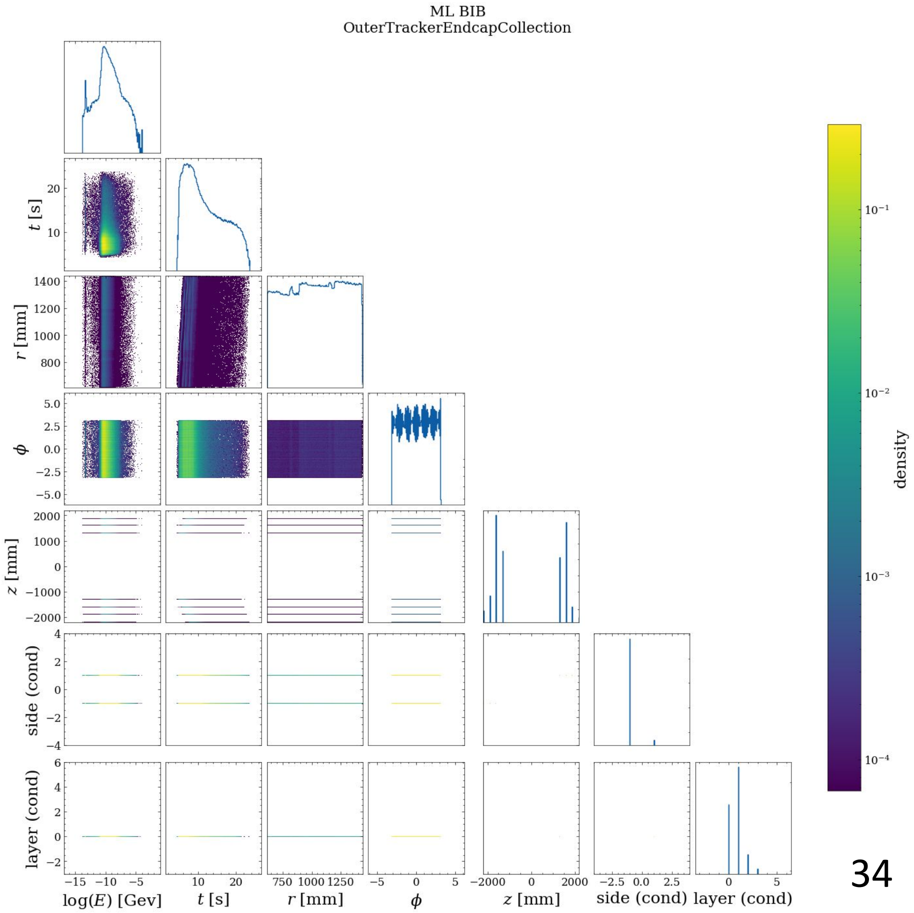
# Corner plots: Outer Tracker Barrel Collection



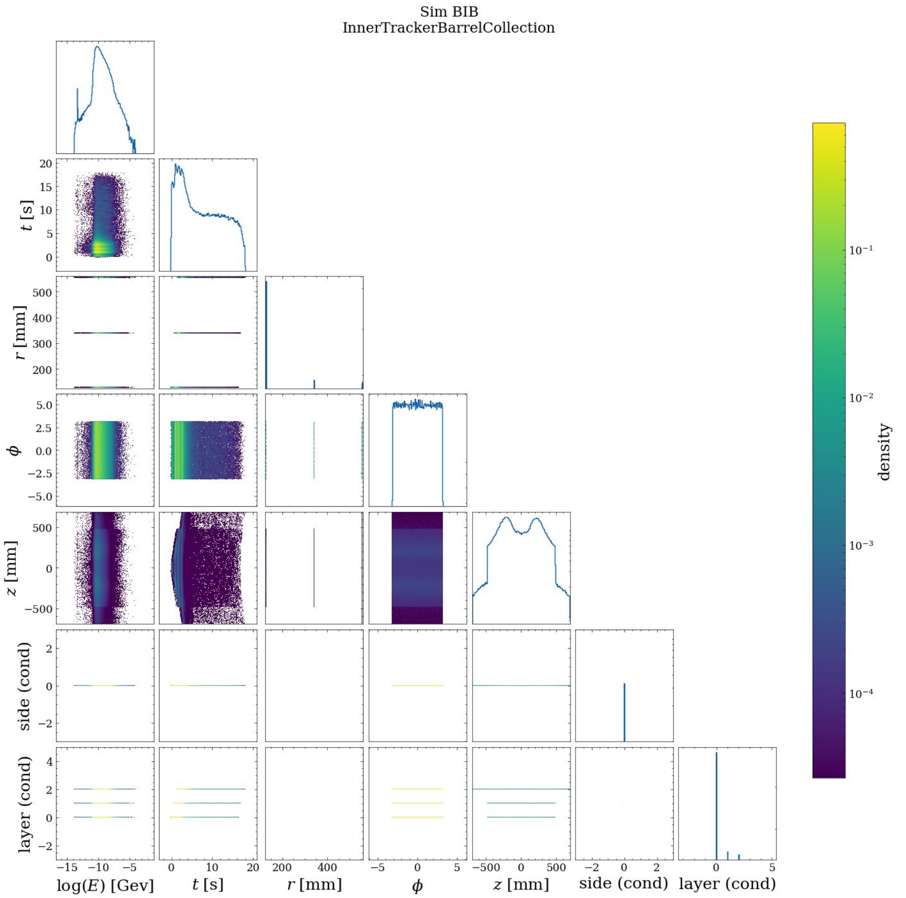



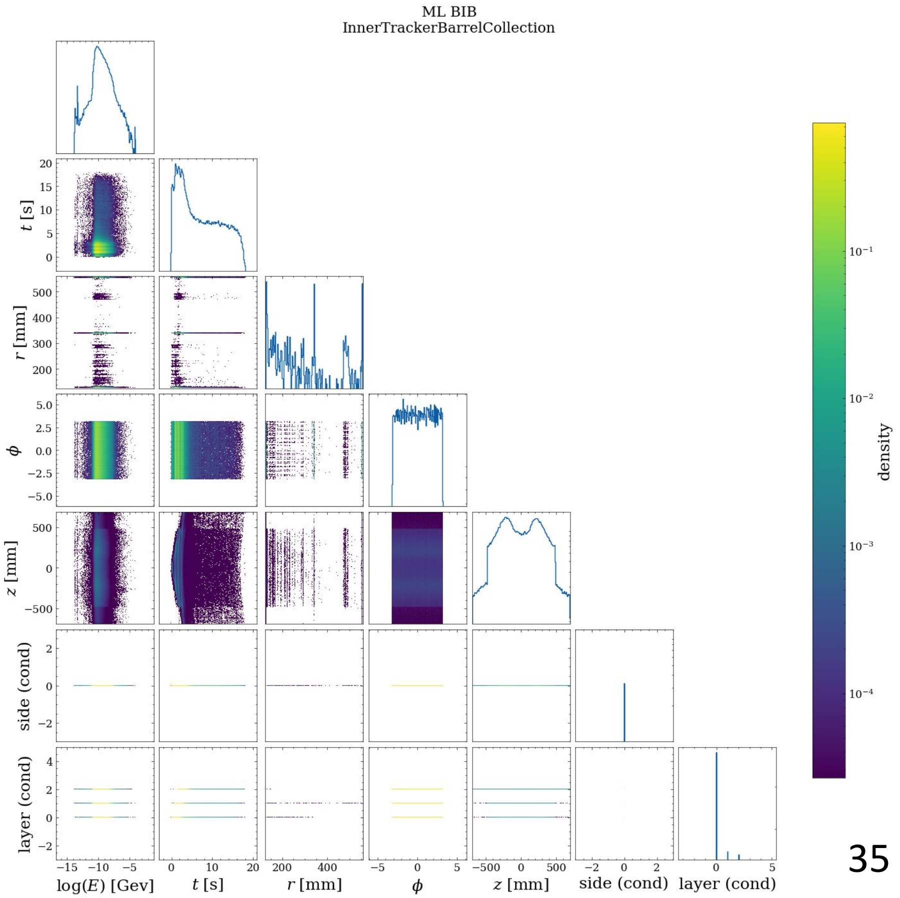
# Corner plots: Outer Tracker Endcap Collection



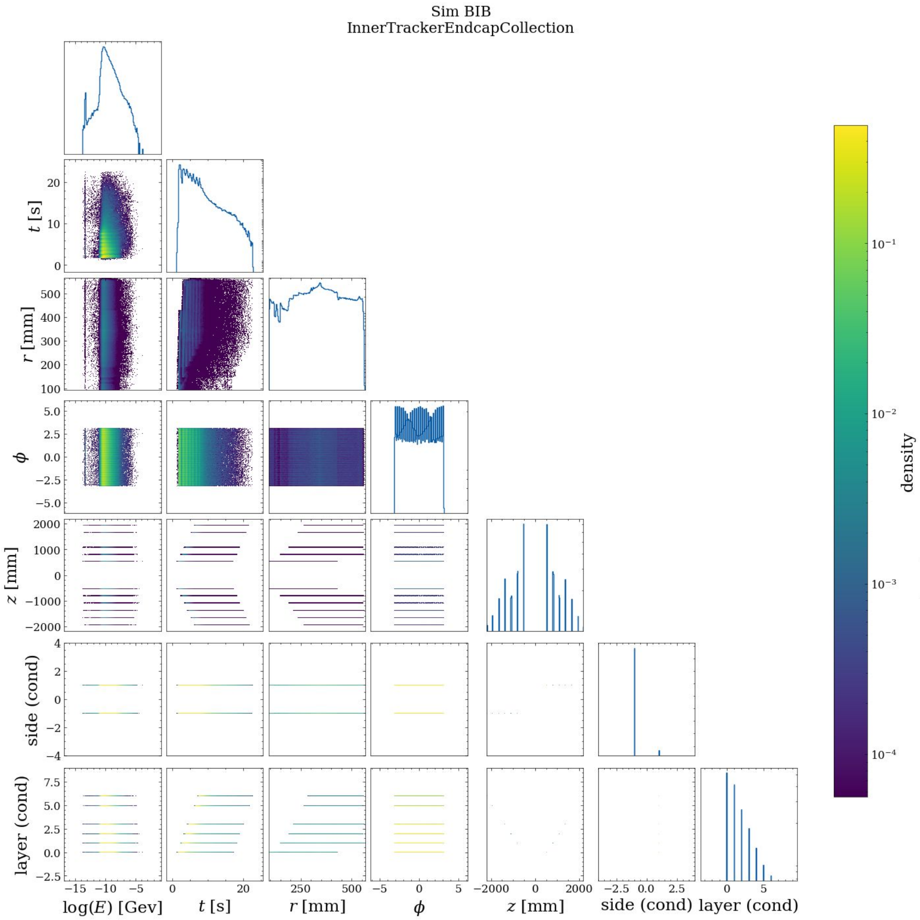



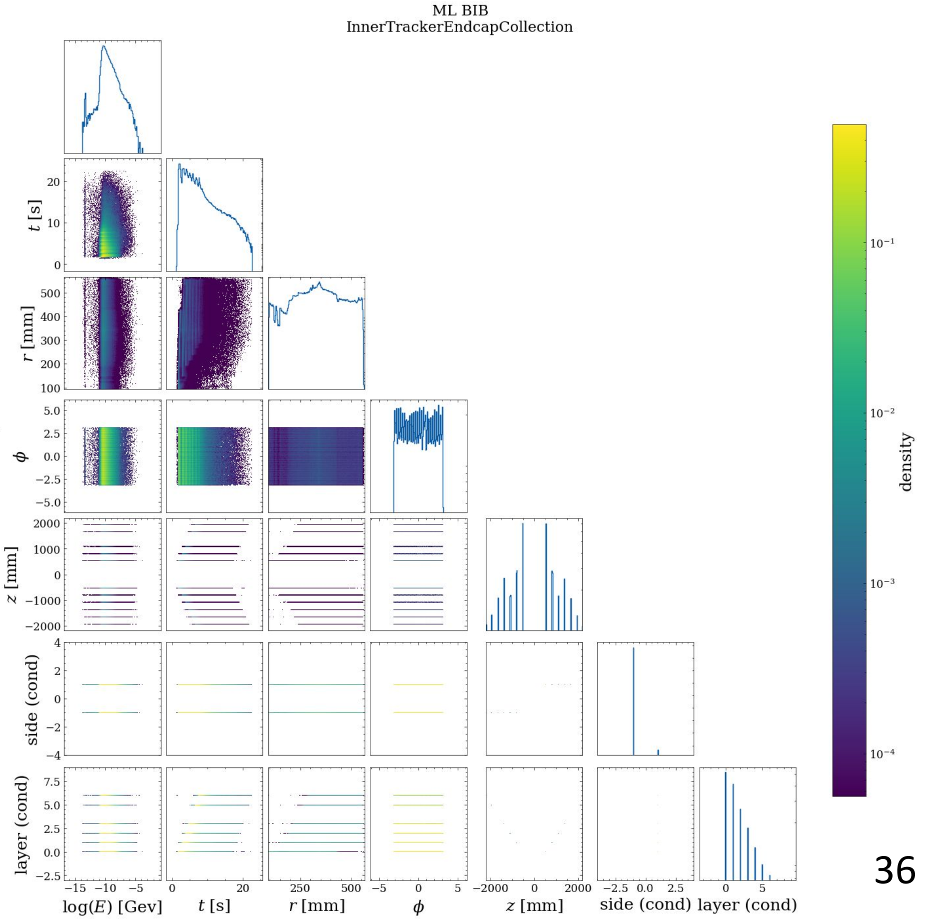
# Corner plots: Inner Tracker Barrel Collection



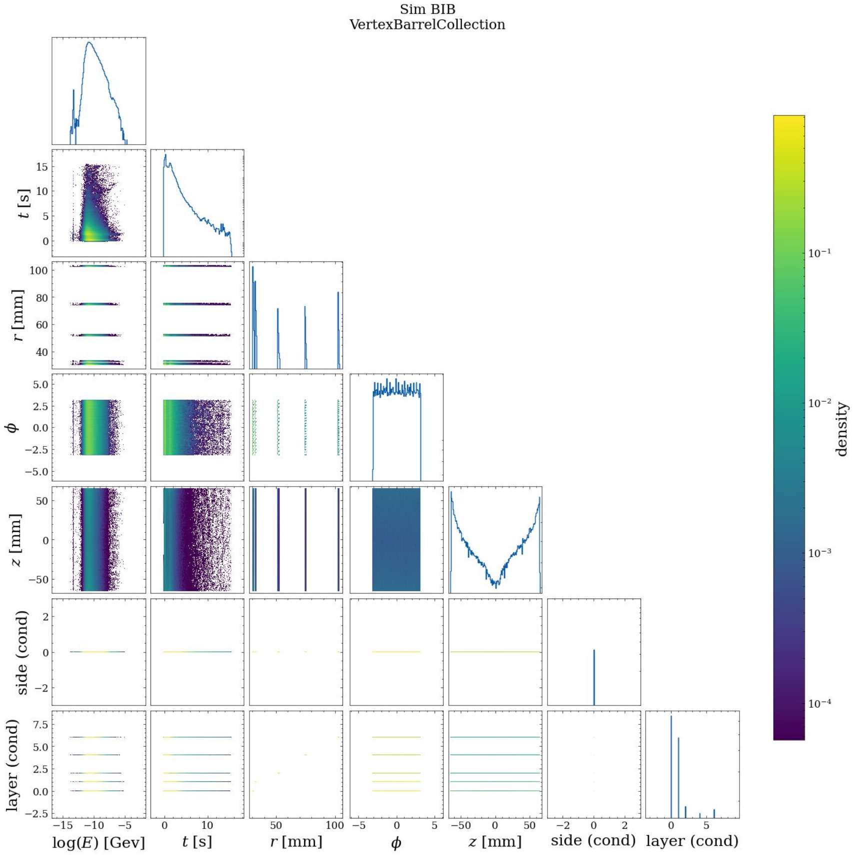



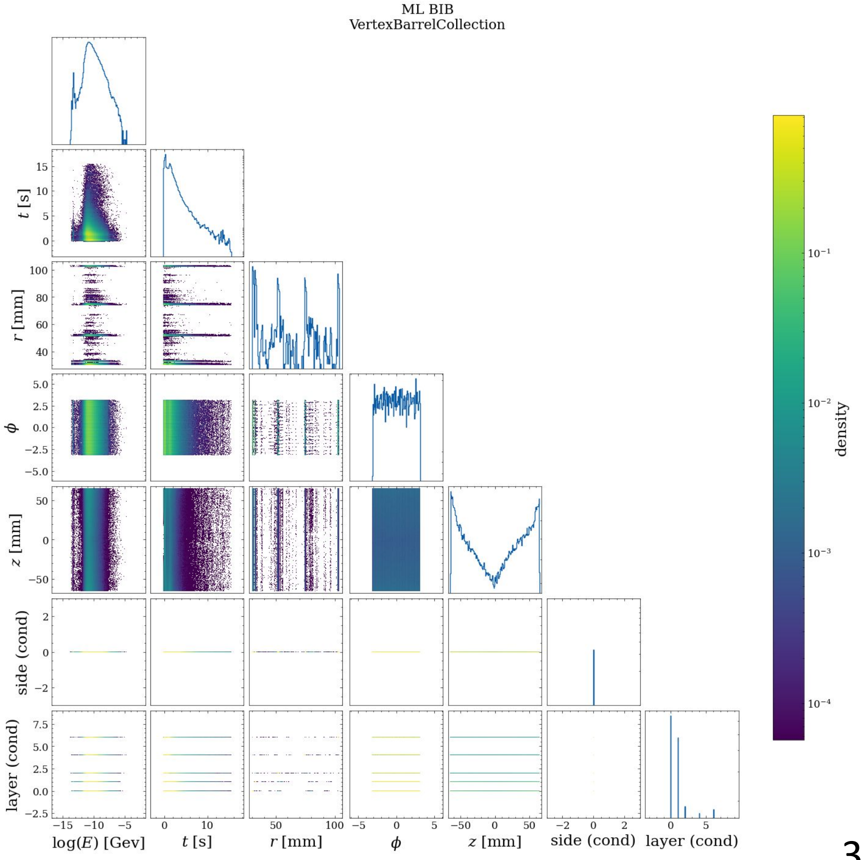
# Corner plots: Inner Tracker Endcap Collection



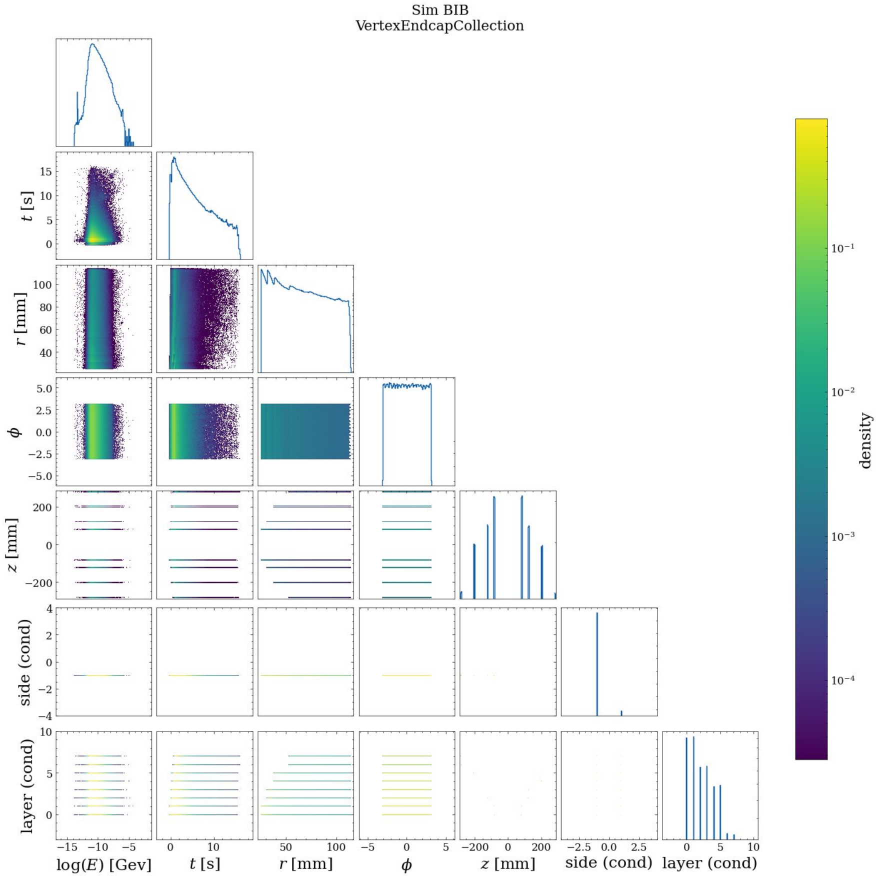



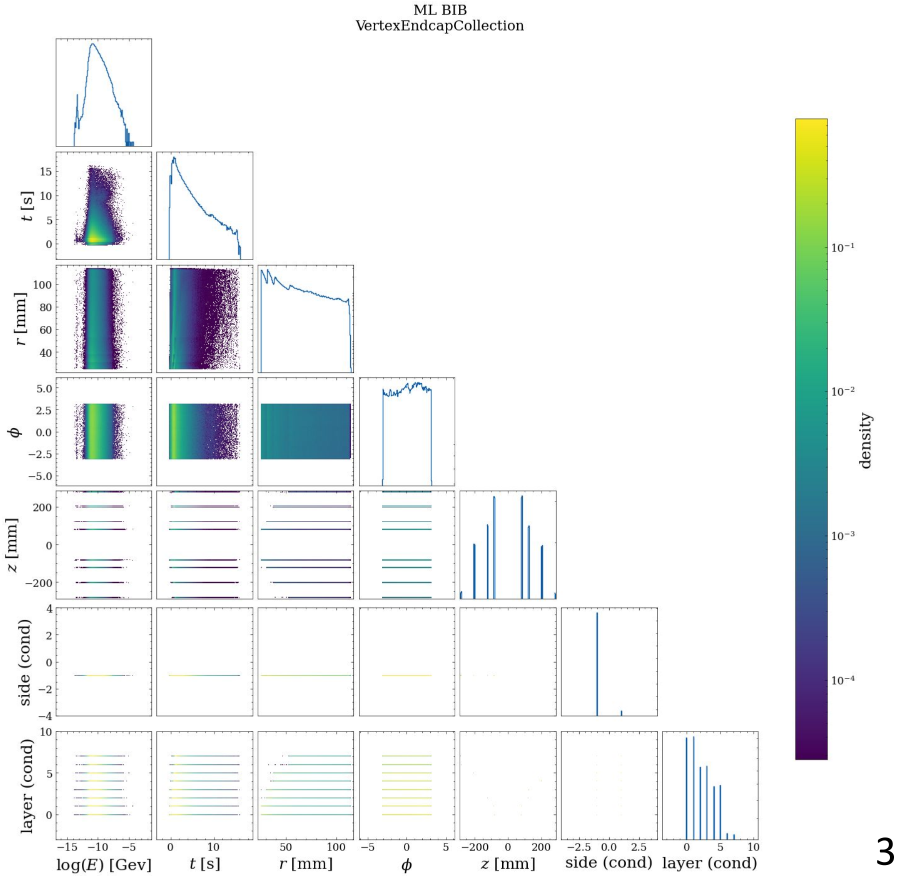
# Corner plots: Vertex Barrel Collection



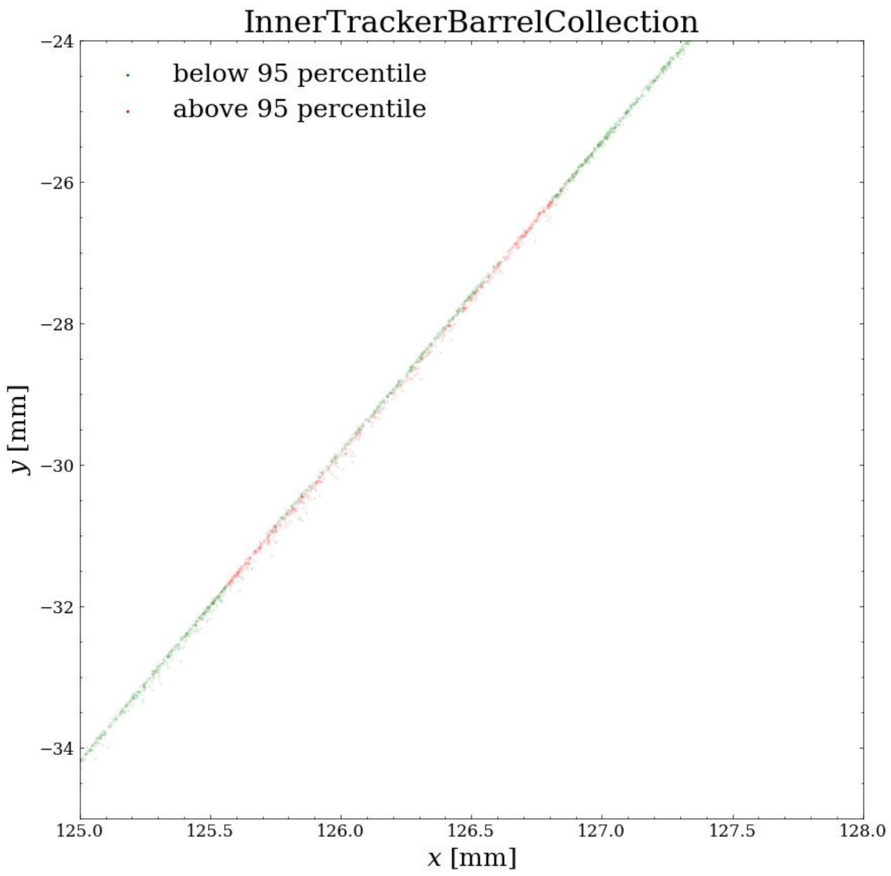



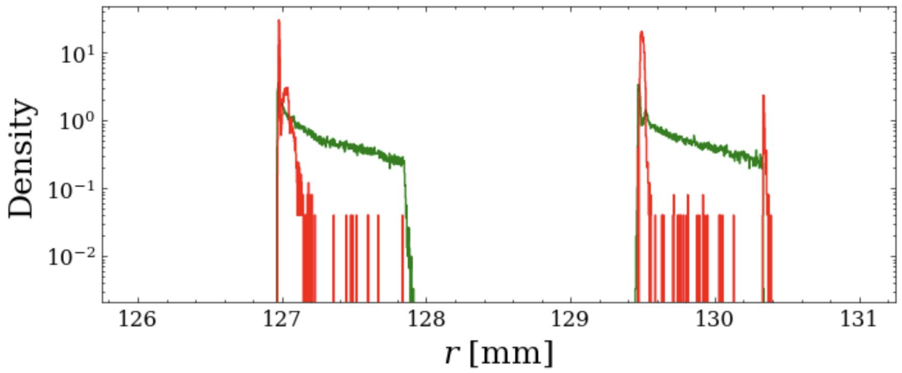
# Corner plots: Vertex Endcap Collection



# Deep dive: why is *r* so poorly modeled in the barrels?




The most “anomalous” (i.e. non Sim BIB-like) hits are at the edges of the sensitive layers of the barrel modules