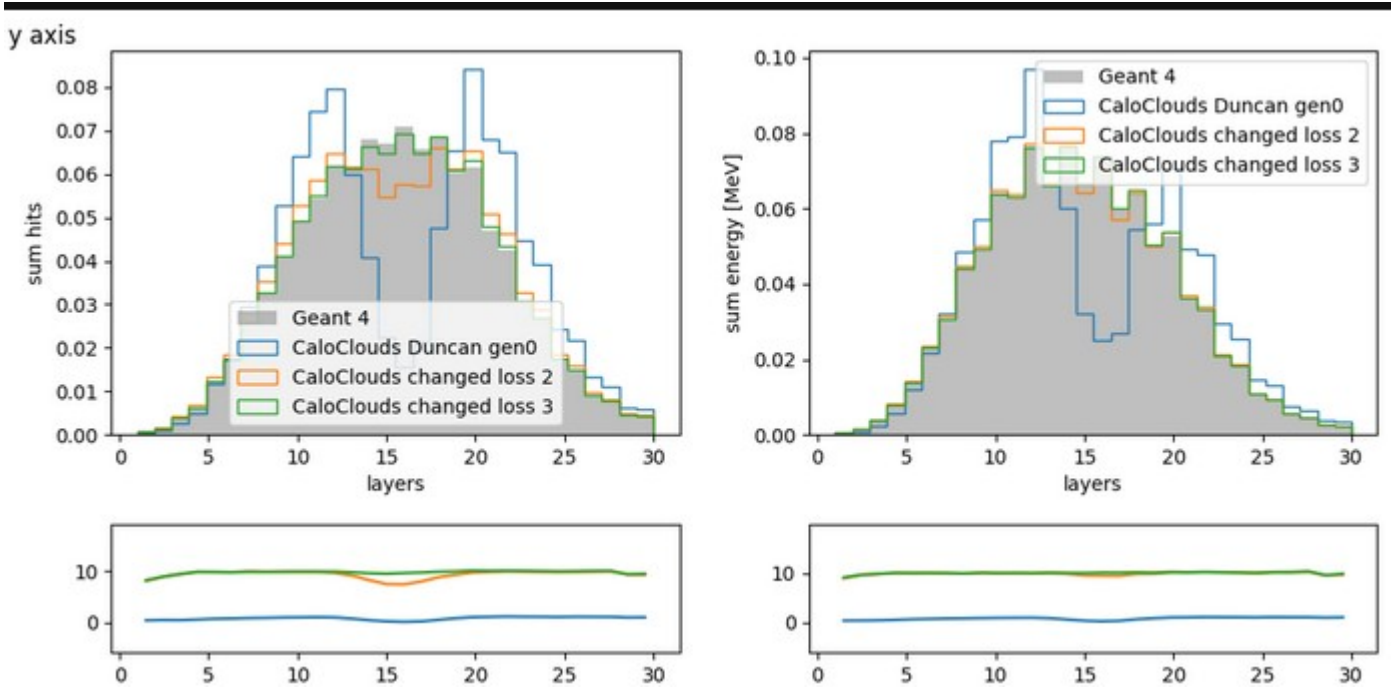
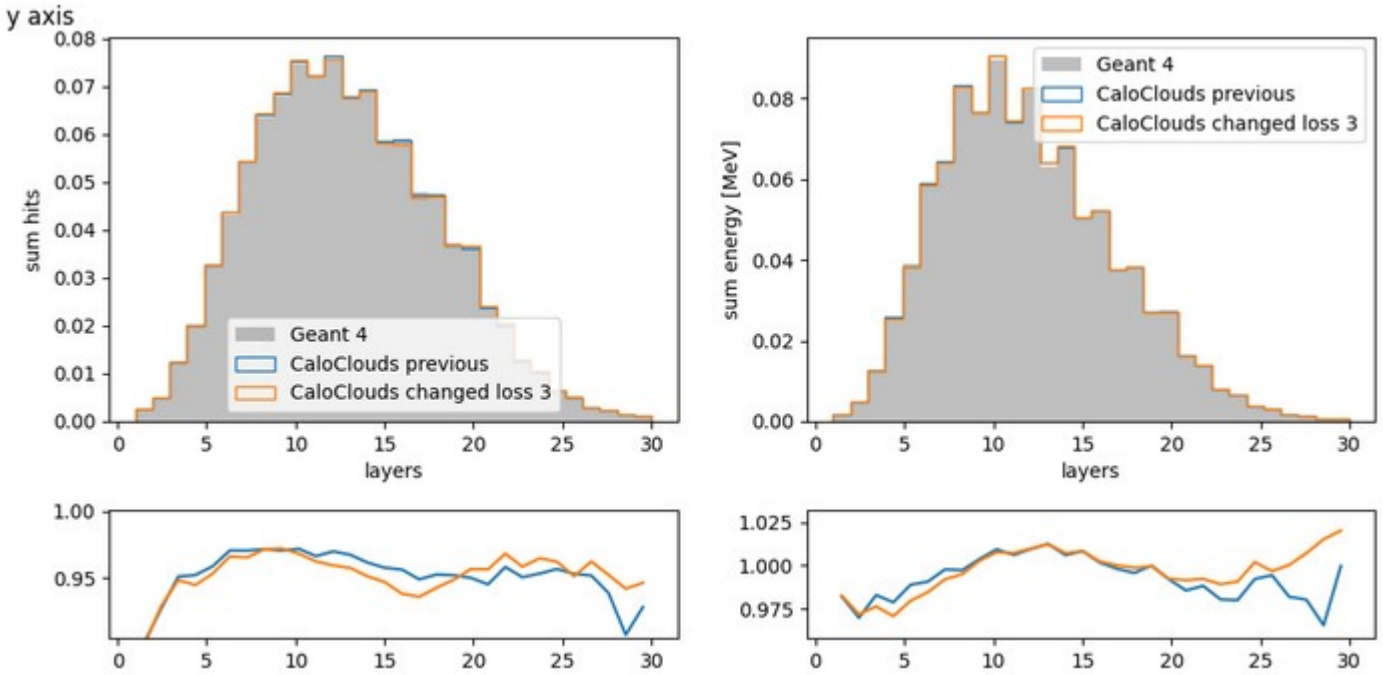
Working on inference

09/01/2025

Resolved bug affecting datasets with old axis config



Also didn’t break current axis config



Plus new unit tests to cover axis varients

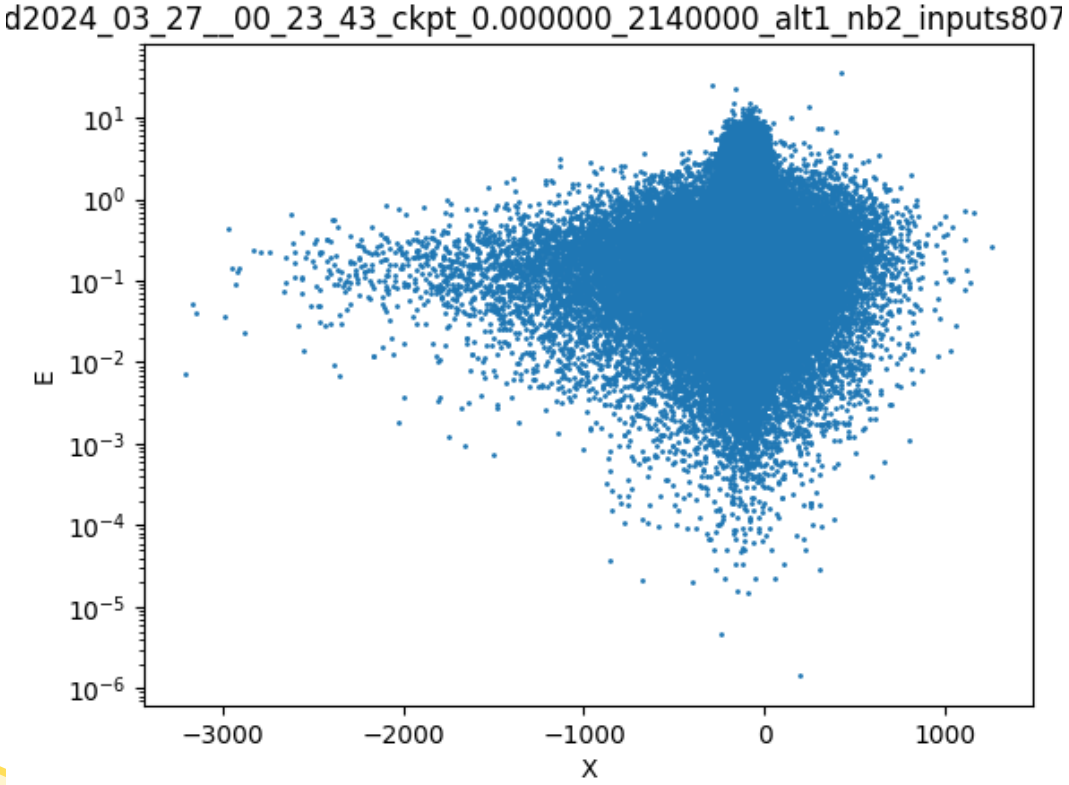
Error messages from jit

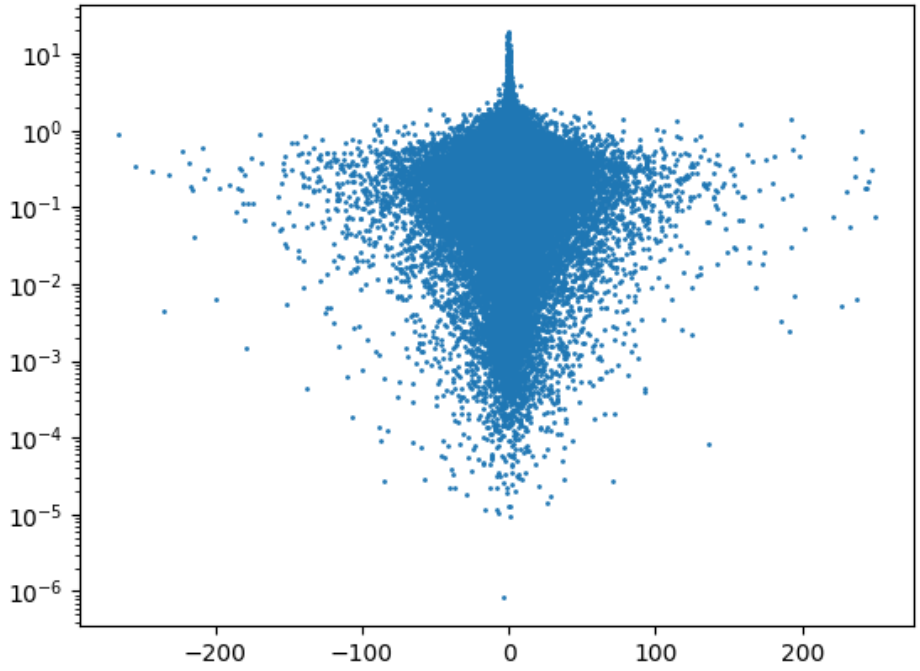
```
python3 inference_clean.py
… (not relevant) …
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/pyro/distributions/transforms/spline.py:535: TracerWarning: Converting a tensor
to a Python boolean might cause the trace to be incorrect. We can't record the data flow of Python values, so this value will be treated as a
constant in the future. This means that the trace might not generalize to other inputs!
  if w.shape[-1] == self.input_dim:
… (ditto previous error) …
  if w.shape[-1] == self.input_dim:
… (ditto previous error) …
  if min_bin_width * num_bins > 1.0:
… (ditto previous error) …
  if min_bin_height * num_bins > 1.0:
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/jit/_trace.py:1306: TracerWarning: Trace had nondeterministic nodes. Did
you forget call .eval() on your model? Nodes:
       %eps : Float(1, 62, strides=[62, 1], requires_grad=0, device=cpu) = aten::normal(%63, %71, %72) #
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/distributions/utils.py:61:0
This may cause errors in trace checking. To disable trace checking, pass check_trace=False to torch.jit.trace()
  _check_trace(
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/jit/_trace.py:1306: TracerWarning: Output nr 1. of the traced function
does not match the corresponding output of the Python function. Detailed error:

Tensor-likes are not close!


Mismatched elements: 58 / 62 (93.5%)
Greatest absolute difference: 2.2353057861328125 at index (0, 1) (up to 1e-05 allowed)
Greatest relative difference: 10.41143919737338 at index (0, 52) (up to 1e-05 allowed)
```

Error messages from jit

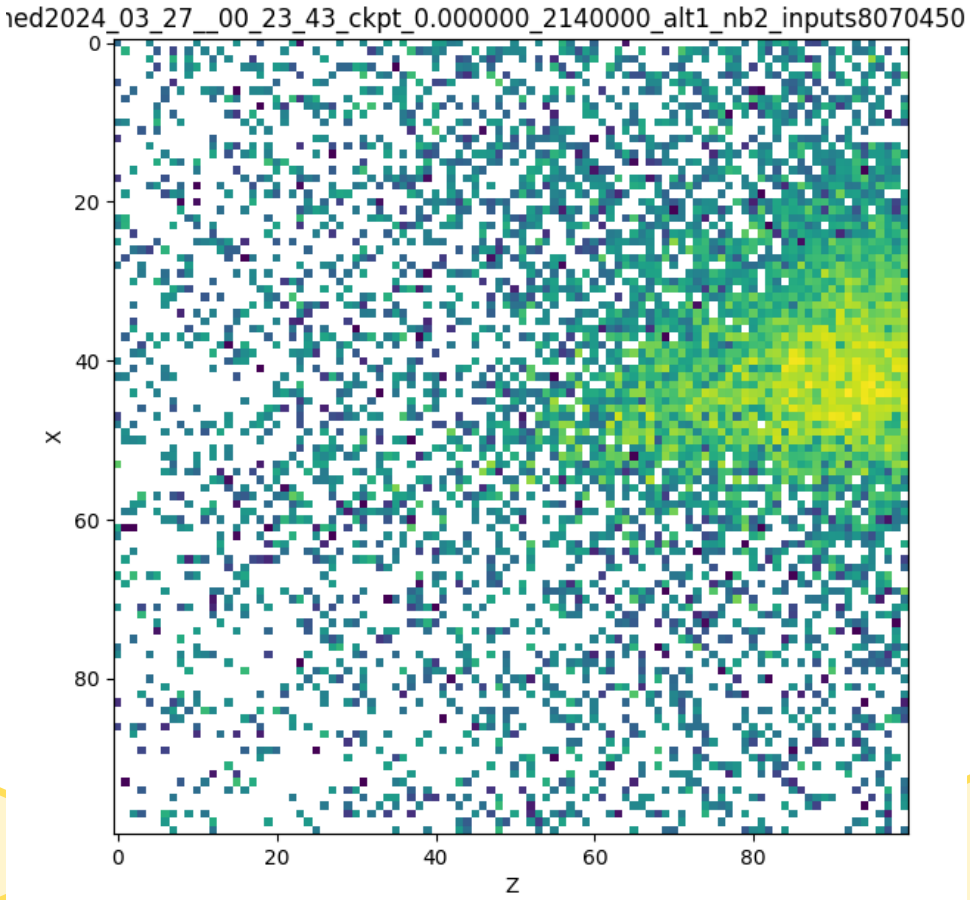
```
(calogpu) max-wn049|11:51:13|physical_recos|$ python3 inference_clean.py
/home/dayhallh/training/physical_recos/inference_clean.py:851: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It
 is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future r
elease, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode
unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded
file. Please open an issue on GitHub for any issues related to this experimental feature.
  loaded = torch.load(sf_model_path, map_location=cfg.device)
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/pyro/distributions/transforms/spline.py:535: TracerWarning: Converting a tensor to a Python boolean might cause the trace to be incorrect. We ca
n't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace might not generalize to other inputs!
  if w.shape[-1] == self.input_dim:
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/pyro/distributions/transforms/spline.py:552: TracerWarning: Converting a tensor to a Python boolean might cause the trace to be incorrect. We ca
n't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace might not generalize to other inputs!
  if w.shape[-1] == self.input_dim:
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/pyro/distributions/transforms/spline.py:111: TracerWarning: Converting a tensor to a Python boolean might cause the trace to be incorrect. We ca
n't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace might not generalize to other inputs!
  if min_bin_width * num_bins > 1.0:
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/pyro/distributions/transforms/spline.py:113: TracerWarning: Converting a tensor to a Python boolean might cause the trace to be incorrect. We ca
n't record the data flow of Python values, so this value will be treated as a constant in the future. This means that the trace might not generalize to other inputs!
  if min_bin_height * num_bins > 1.0:
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/jit/_trace.py:1306: TracerWarning: Trace had nondeterministic nodes. Did you forget call .eval() on your model? Nodes:
        %eps : Float(1, 62, strides=[62, 1], requires_grad=0, device=cpu) = aten::normal(%63, %71, %72) # /data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/distributions/utils.py:61:0
This may cause errors in trace checking. To disable trace checking, pass check_trace=False to torch.jit.trace()
  _check_trace(
/data/dust/user/dayhallh/envs/calogpu/lib/python3.12/site-packages/torch/jit/_trace.py:1306: TracerWarning: Output nr 1. of the traced function does not match the corresponding output of the Python function. Det
ailed error:
Tensor-likes are not close!

Mismatched elements: 58 / 62 (93.5%)
Greatest absolute difference: 2.2353057861328125 at index (0, 1) (up to 1e-05 allowed)
Greatest relative difference: 10.41143919737338 at index (0, 52) (up to 1e-05 allowed)
  _check_trace(
/home/dayhallh/training/physical_recos/inference_clean.py:892: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It
 is possible to construct malicious pickle data which will execute arbitrary code during unpickling (See https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future r
elease, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode
unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded
file. Please open an issue on GitHub for any issues related to this experimental feature.
```

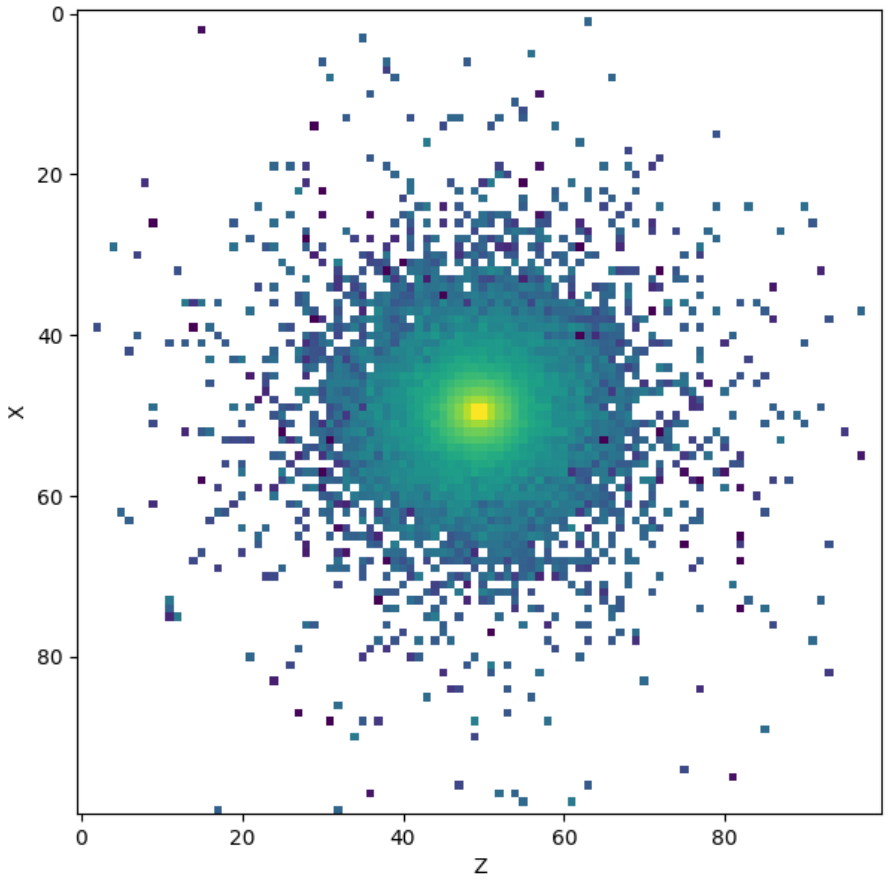
Plots with some diffusion models look weird



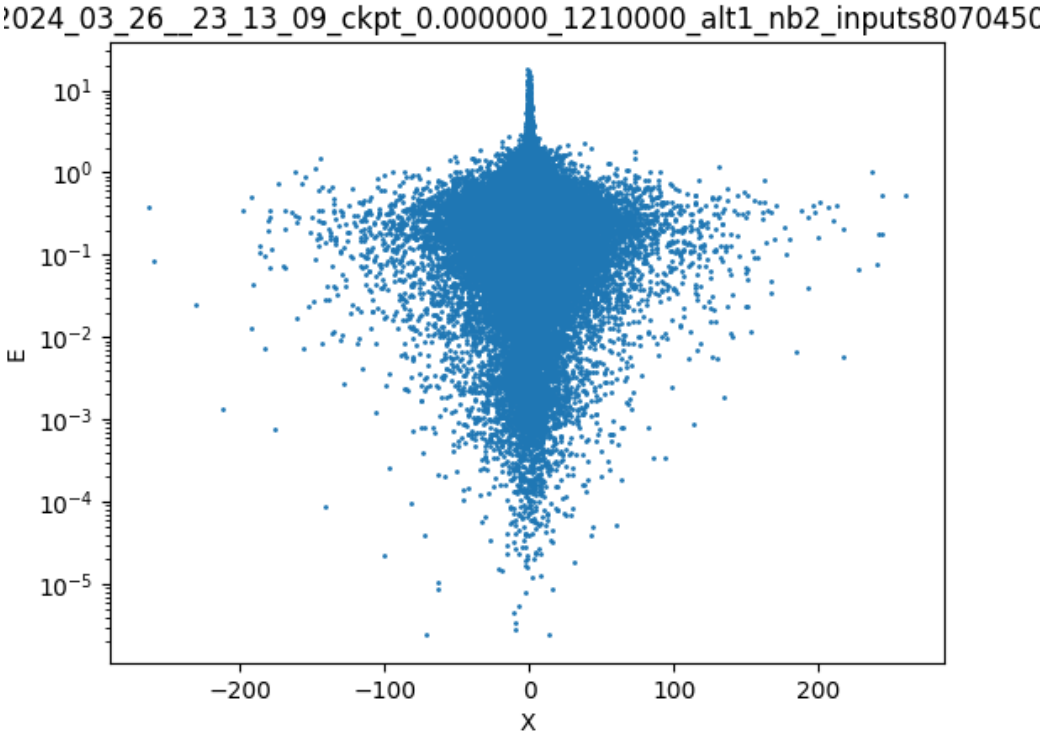


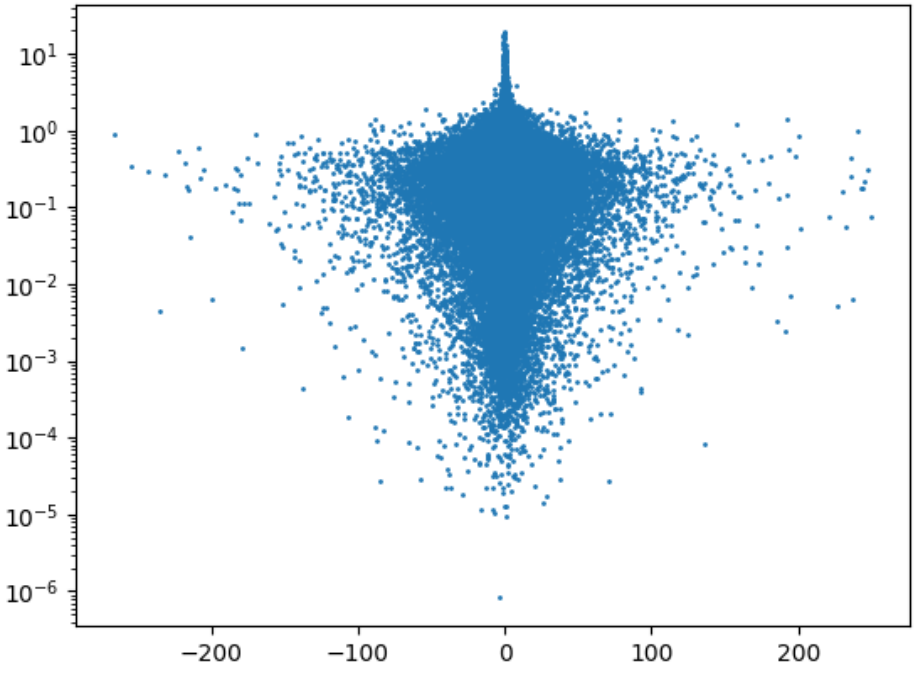
Plots with some diffusion models look weird



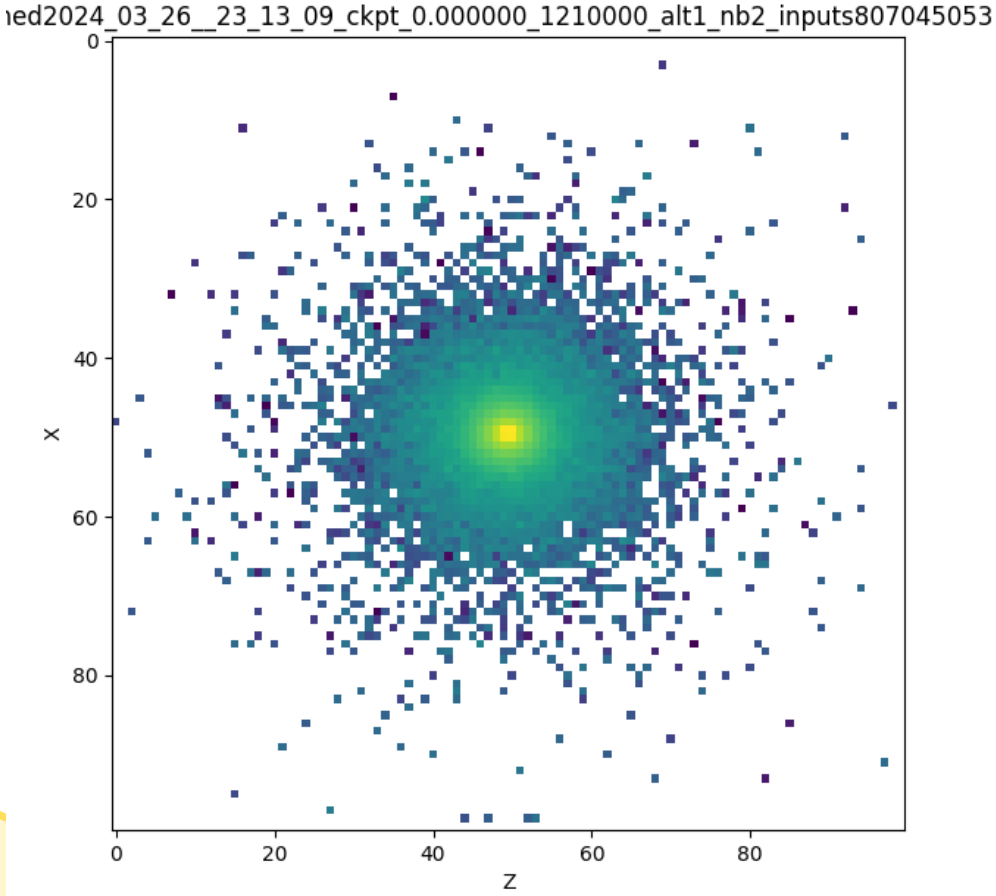


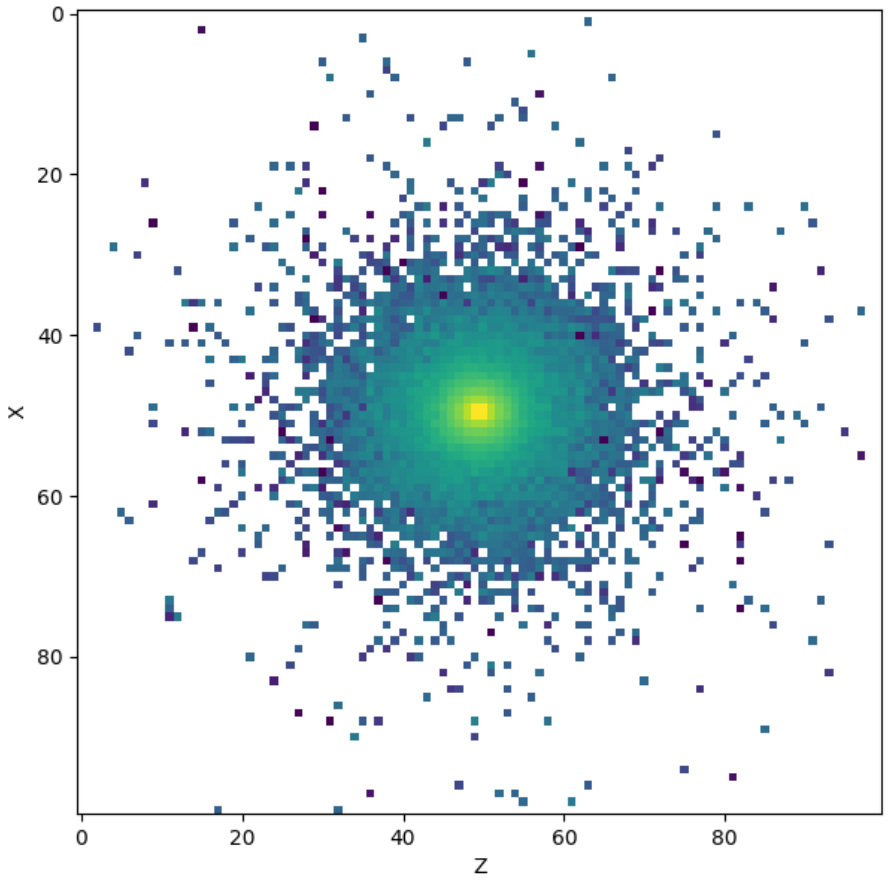
Older diffusion model still seems fine





Older diffusion model still seems fine





Getting a seg fault in run sim though

```
LCIOFileReader   INFO  Created file reader. Try to open input /user_data/inputs/ILD-gen-E2020pdg22maxSep30centralPosZ50-p1.slcio
GenerationInit   INFO  +++ Initializing event 1. Within run:0 event 1.
LCIOFileReader   INFO  read collection MCParticle from event 0 in run 0
LCIO4            INFO  +++ Particle interaction with 2 generator particles and 2 vertices ++++++++++++++++++++++++
PrimaryHandler   INFO  +++++ G4PrimaryVertex at (-4.65e+00,+1.80e+03,+5.84e+01) [mm] -1.09e-04 [ns]
PrimaryHandler   INFO  +++++ G4PrimaryVertex at (-4.65e+00,+1.80e+03,+5.90e+01) [mm] -1.09e-04 [ns]
ParticleHandler  INFO  +++ Event 0 Begin event action. Access event related information.
  PolyhedraBarrelGeometry::localDirection  - symmetry = 8 pos0 = (-4.64739,1804.7,58.9755) - dir = (0.00699994,0.999976,0) - E =  - localDir = (-0,-0.00699994,0.999976)20000.5
  PolyhedraBarrelGeometry::localDirection  - phi = -90   theta : 0.40107
 *** Break *** segmentation violation



============================================================
There was a crash.
This is the entire stack trace of all threads:
============================================================
```

It’s a problem regardless of the model used (new or old, for both CC and SF), so it’s an issue in the inference process.