Weekly Update

August 15, 2025

Leo Zhu and Denaisha Kraft

New Data

File Structure: atlas_july31

- Eta (ie “eta_050”)
  - Binning type: (eta_000_default_binning or eta_000_regular_binning)
    - “Complete” datasets (dataset_combined_fine.hdf5 dataset_combined.hdf5 dataset_combined_positive.hdf5): 150,000 events each
    - Jobs (0 -19)
      - Each has its own set of the 3 datasets, 7,500 events each
    - Can build “rebuilt” datasets by combining datasets in job folders, 150,000 events total each

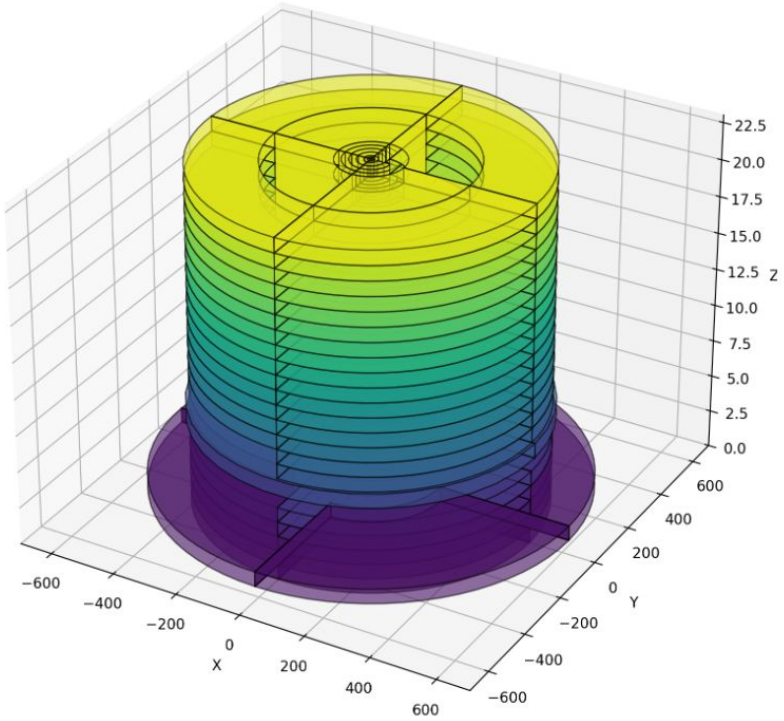
In each dataset

- Familiar set of keys (layer_num, incident energies, geometry)

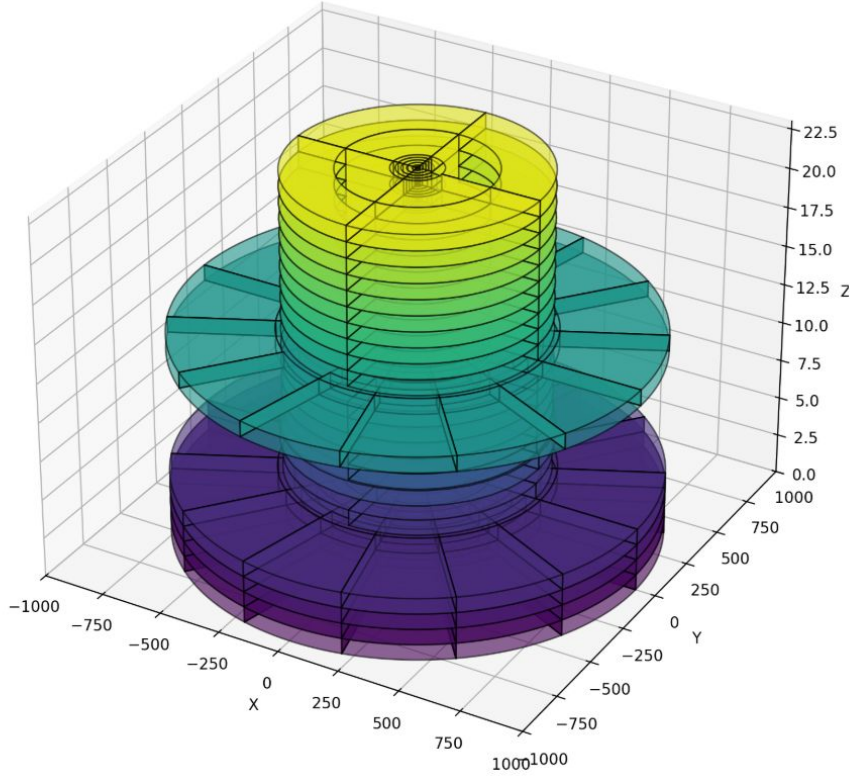
Cylinder Plots

Using dataset_combined and dataset_combined_positive:

Default binning:



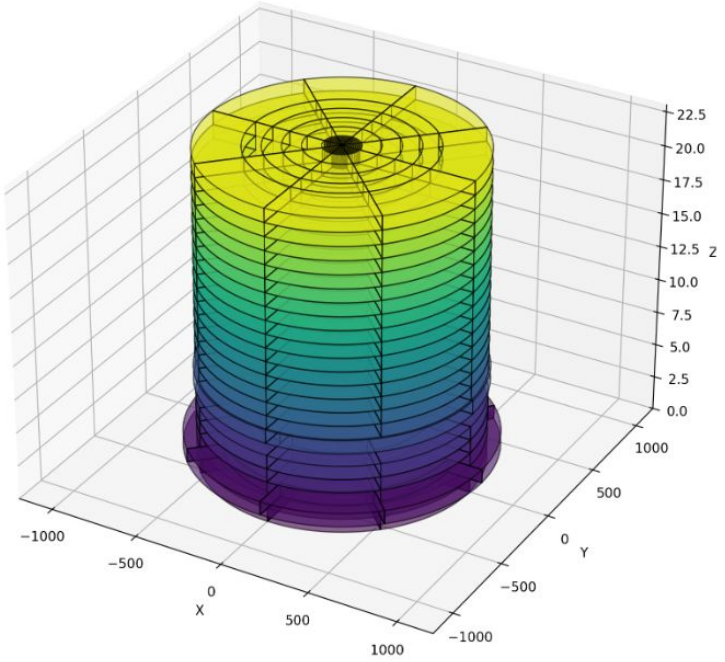
Regular binning:



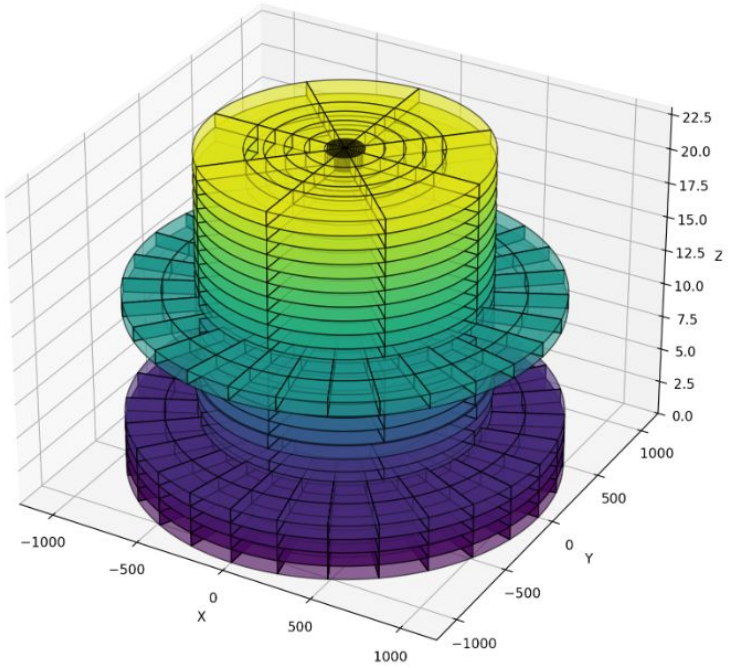
Fine Binning Cylinder Plots

Using dataset_combined_fine

Default binning:



Regular binning:



Voxels Per Layer

Default binning (events and voxels):

- combined and combined_positive

```
energy_layer_0: shape=(150000, 36),
energy_layer_1: shape=(150000, 164),
energy_layer_10: shape=(150000, 36),
energy_layer_11: shape=(150000, 36),
energy_layer_12: shape=(150000, 36),
energy_layer_13: shape=(150000, 36),
energy_layer_14: shape=(150000, 36),
energy_layer_15: shape=(150000, 36),
energy_layer_16: shape=(150000, 36),
energy_layer_17: shape=(150000, 36),
energy_layer_18: shape=(150000, 36),
energy_layer_19: shape=(150000, 36),
energy_layer_2: shape=(150000, 110),
energy_layer_20: shape=(150000, 36),
energy_layer_21: shape=(150000, 36),
energy_layer_22: shape=(150000, 36),
energy_layer_23: shape=(150000, 36),
energy_layer_3: shape=(150000, 36),
energy_layer_4: shape=(150000, 36),
energy_layer_5: shape=(150000, 124),
energy_layer_6: shape=(150000, 124),
energy_layer_7: shape=(150000, 60),
energy_layer_8: shape=(150000, 36),
energy_layer_9: shape=(150000, 36),
```

combined_fine

```
energy_layer_0: shape=(150000, 144),
energy_layer_1: shape=(150000, 656),
energy_layer_10: shape=(150000, 144),
energy_layer_11: shape=(150000, 144),
energy_layer_12: shape=(150000, 144),
energy_layer_13: shape=(150000, 144),
energy_layer_14: shape=(150000, 144),
energy_layer_15: shape=(150000, 144),
energy_layer_16: shape=(150000, 144),
energy_layer_17: shape=(150000, 144),
energy_layer_18: shape=(150000, 144),
energy_layer_19: shape=(150000, 144),
energy_layer_2: shape=(150000, 440),
energy_layer_20: shape=(150000, 144),
energy_layer_21: shape=(150000, 144),
energy_layer_22: shape=(150000, 144),
energy_layer_23: shape=(150000, 144),
energy_layer_3: shape=(150000, 144),
energy_layer_4: shape=(150000, 144),
energy_layer_5: shape=(150000, 496),
energy_layer_6: shape=(150000, 496),
energy_layer_7: shape=(150000, 240),
energy_layer_8: shape=(150000, 144),
energy_layer_9: shape=(150000, 144),
```

Voxels Per Layer

Regular binning (events and voxels):

- combined and combined_positive

```
energy_layer_0: shape=(150000, 336),
energy_layer_1: shape=(150000, 336),
energy_layer_10: shape=(150000, 36),
energy_layer_11: shape=(150000, 36),
energy_layer_12: shape=(150000, 336)
energy_layer_13: shape=(150000, 36),
energy_layer_14: shape=(150000, 36),
energy_layer_15: shape=(150000, 36),
energy_layer_16: shape=(150000, 36),
energy_layer_17: shape=(150000, 36),
energy_layer_18: shape=(150000, 36),
energy_layer_19: shape=(150000, 36),
energy_layer_2: shape=(150000, 336),
energy_layer_20: shape=(150000, 36),
energy_layer_21: shape=(150000, 36),
energy_layer_22: shape=(150000, 36),
energy_layer_23: shape=(150000, 36),
energy_layer_3: shape=(150000, 336),
energy_layer_4: shape=(150000, 36),
energy_layer_5: shape=(150000, 124),
energy_layer_6: shape=(150000, 124),
energy_layer_7: shape=(150000, 60),
energy_layer_8: shape=(150000, 36),
energy_layer_9: shape=(150000, 36),
```

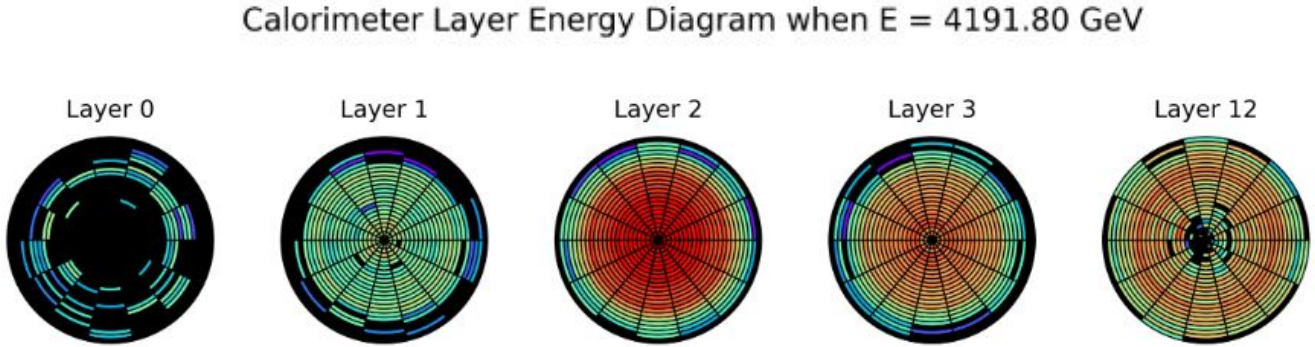
combined_fine

```
energy_layer_0: shape=(150000, 1344),
energy_layer_1: shape=(150000, 1344),
energy_layer_10: shape=(150000, 144),
energy_layer_11: shape=(150000, 144),
energy_layer_12: shape=(150000, 1344)
energy_layer_13: shape=(150000, 144),
energy_layer_14: shape=(150000, 144),
energy_layer_15: shape=(150000, 144),
energy_layer_16: shape=(150000, 144),
energy_layer_17: shape=(150000, 144),
energy_layer_18: shape=(150000, 144),
energy_layer_19: shape=(150000, 144),
energy_layer_2: shape=(150000, 1344),
energy_layer_20: shape=(150000, 144),
energy_layer_21: shape=(150000, 144),
energy_layer_22: shape=(150000, 144),
energy_layer_23: shape=(150000, 144),
energy_layer_3: shape=(150000, 1344),
energy_layer_4: shape=(150000, 144),
energy_layer_5: shape=(150000, 496),
energy_layer_6: shape=(150000, 496),
energy_layer_7: shape=(150000, 240),
energy_layer_8: shape=(150000, 144),
energy_layer_9: shape=(150000, 144),
```

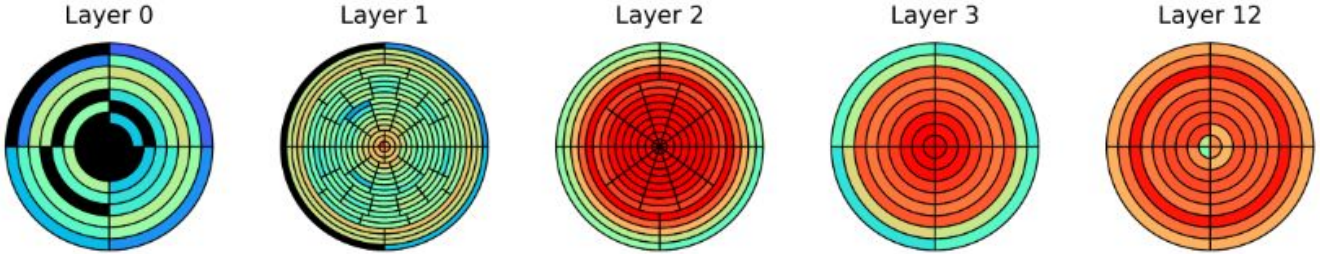
Cylinder Slices

Example for eta_000:

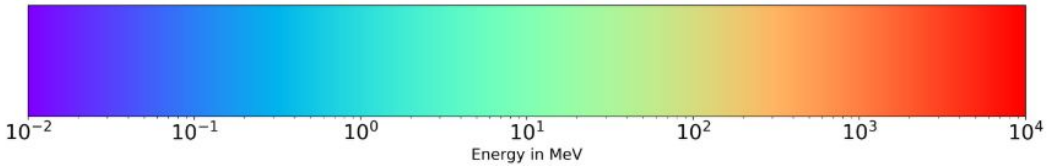
Regular binning:



Default binning:



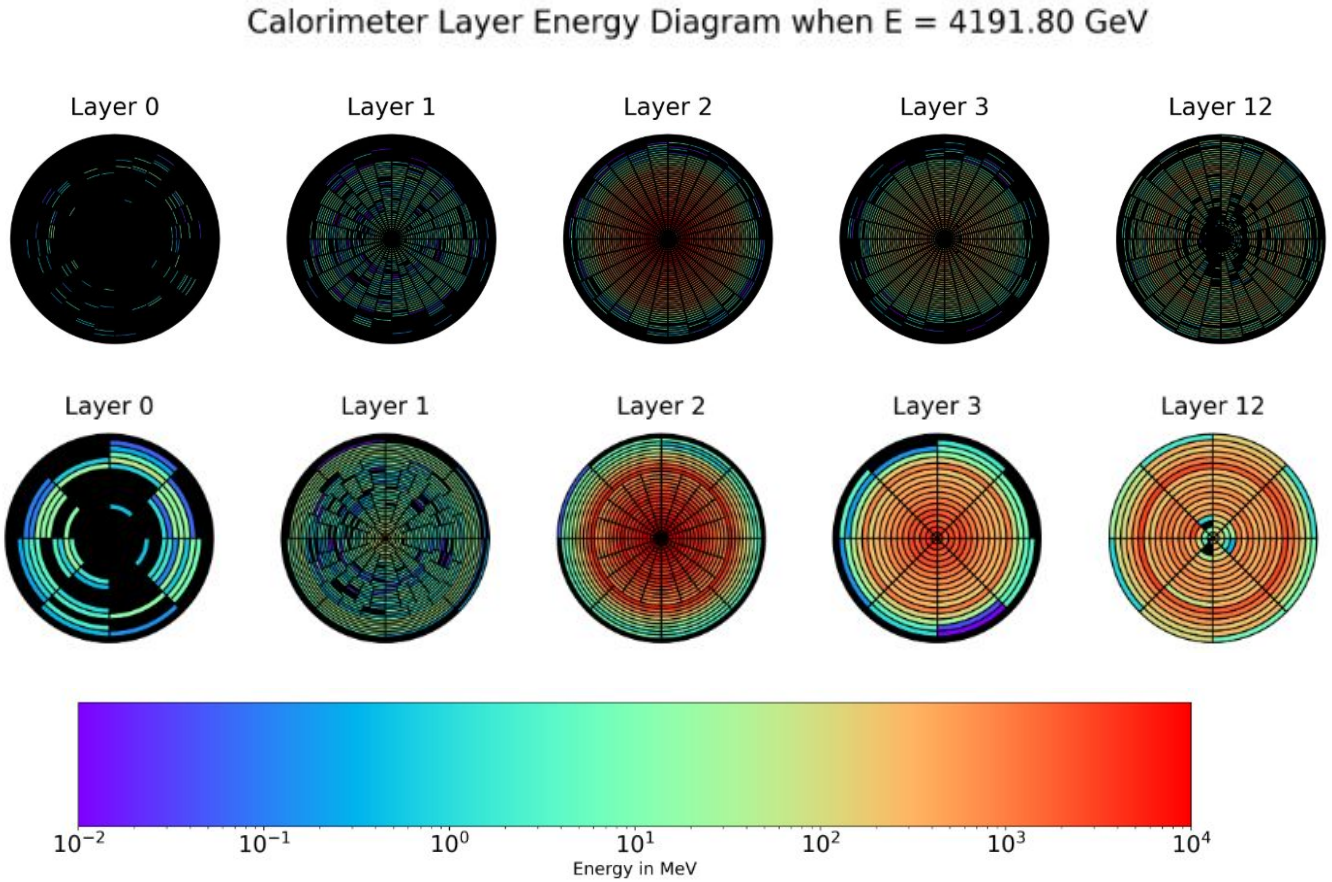
Dataset_combined and dataset_combined_positive have the same binning



Fine Cylinder Slices

Regular binning:

Default binning:



File Binning

Using regular_binning for active layers (layers with events that have non-zero voxel energy deposits)

- This is the same for all eta

```
Layer  0: radial bins =  24, angular bins =  14
Layer  1: radial bins =  24, angular bins =  14
Layer  2: radial bins =  24, angular bins =  14
Layer  3: radial bins =  24, angular bins =  14
Layer 12: radial bins =  24, angular bins =  14
```

Note previous dataset that was being trained on had 24 radial bins and 14 angular bins for all layers

For all layers (regular binning):

- This is for dataset_combined_positive and dataset_combined (eta_000)

```
Layer  0: radial bins = 24, angular bins = 14
Layer  1: radial bins = 24, angular bins = 14
Layer  2: radial bins = 24, angular bins = 14
Layer  3: radial bins = 24, angular bins = 14
Layer  4: radial bins =  9, angular bins =  4
Layer  5: radial bins = 31, angular bins =  4
Layer  6: radial bins = 31, angular bins =  4
Layer  7: radial bins = 15, angular bins =  4
Layer  8: radial bins =  9, angular bins =  4
Layer  9: radial bins =  9, angular bins =  4
Layer 10: radial bins =  9, angular bins =  4
Layer 11: radial bins =  9, angular bins =  4
Layer 12: radial bins = 24, angular bins = 14
Layer 13: radial bins =  9, angular bins =  4
Layer 14: radial bins =  9, angular bins =  4
Layer 15: radial bins =  9, angular bins =  4
Layer 16: radial bins =  9, angular bins =  4
Layer 17: radial bins =  9, angular bins =  4
Layer 18: radial bins =  9, angular bins =  4
Layer 19: radial bins =  9, angular bins =  4
Layer 20: radial bins =  9, angular bins =  4
Layer 21: radial bins =  9, angular bins =  4
Layer 22: radial bins =  9, angular bins =  4
Layer 23: radial bins =  9, angular bins =  4
```

Fine Binning

Using regular_binning for active layers

- This is the same for all eta

```
Layer  0: radial bins =  48, angular bins =  28
Layer  1: radial bins =  48, angular bins =  28
Layer  2: radial bins =  48, angular bins =  28
Layer  3: radial bins =  48, angular bins =  28
Layer 12: radial bins =  48, angular bins =  28
```

Compared to dataset_combined and dataset_combined_positive, the dataset_fine files have 2x as many radial bins and 2x as many angular bins for each active layer

For all layers (regular binning):

```
Layer  0: radial bins = 48, angular bins = 28
Layer  1: radial bins = 48, angular bins = 28
Layer  2: radial bins = 48, angular bins = 28
Layer  3: radial bins = 48, angular bins = 28
Layer  4: radial bins = 18, angular bins =  8
Layer  5: radial bins = 31, angular bins = 16
Layer  6: radial bins = 31, angular bins = 16
Layer  7: radial bins = 30, angular bins =  8
Layer  8: radial bins = 18, angular bins =  8
Layer  9: radial bins = 18, angular bins =  8
Layer 10: radial bins = 18, angular bins =  8
Layer 11: radial bins = 18, angular bins =  8
Layer 12: radial bins = 48, angular bins = 28
Layer 13: radial bins = 18, angular bins =  8
Layer 14: radial bins = 18, angular bins =  8
Layer 15: radial bins = 18, angular bins =  8
Layer 16: radial bins = 18, angular bins =  8
Layer 17: radial bins = 18, angular bins =  8
Layer 18: radial bins = 18, angular bins =  8
Layer 19: radial bins = 18, angular bins =  8
Layer 20: radial bins = 18, angular bins =  8
Layer 21: radial bins = 18, angular bins =  8
Layer 22: radial bins = 18, angular bins =  8
Layer 23: radial bins = 18, angular bins =  8
```

Rebuilt File Binning

The rebuilt files have more active layers

In these files, certain active layers have different r and phi binning

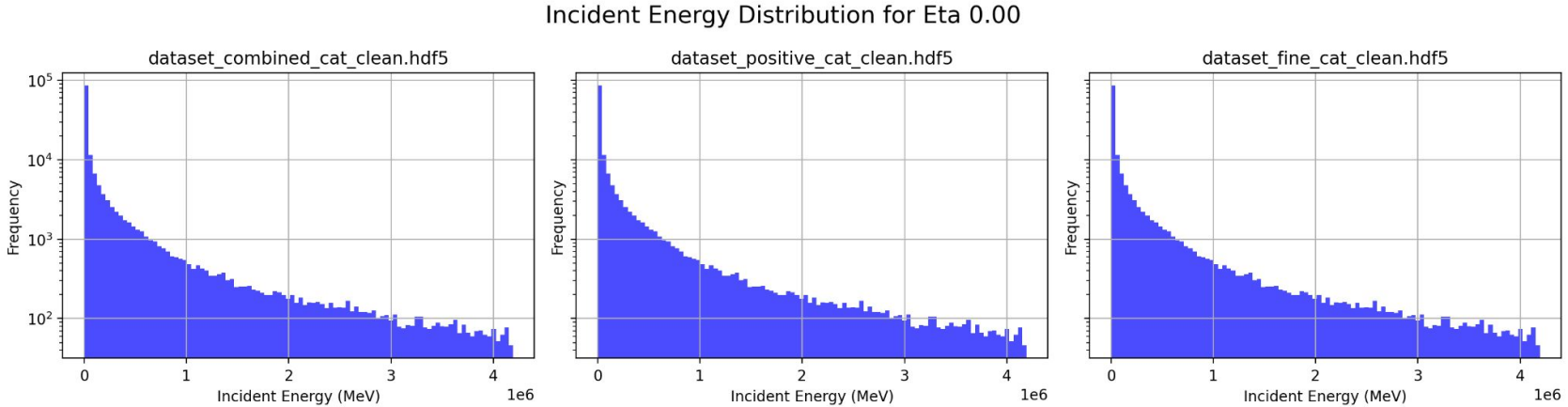
- Example (r, phi) binning for eta_130 combined and combined_positive:

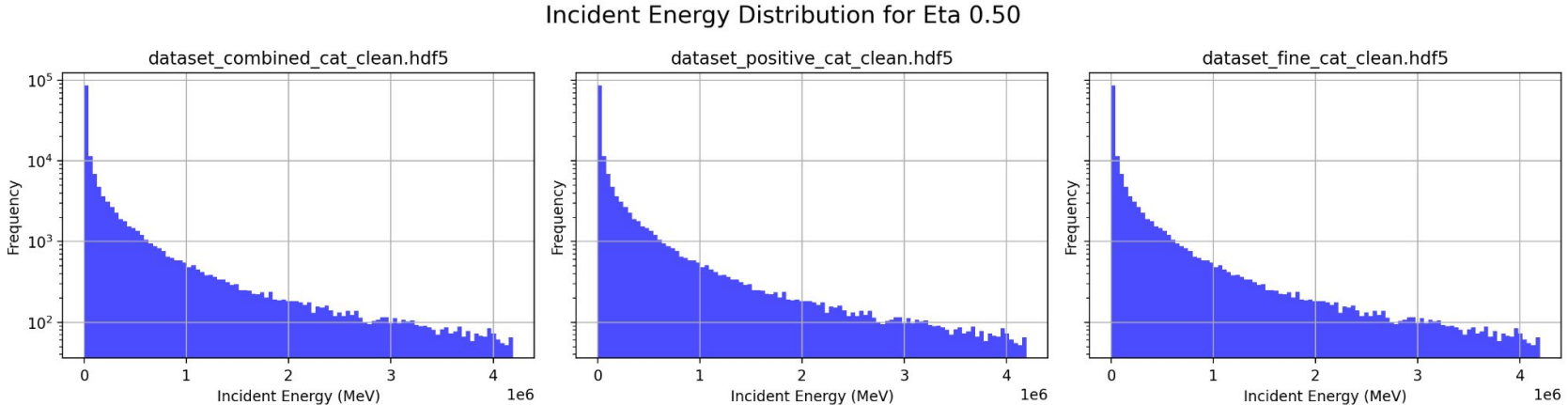
```
eta_130 | Layer  0: Match | Binning = (24, 14)
eta_130 | Layer  1: Match | Binning = (24, 14)
eta_130 | Layer  2: Match | Binning = (24, 14)
eta_130 | Layer  3: Match | Binning = (24, 14)
eta_130 | Layer  4: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer  5: Extra in split (inactive in main) | Binning = {(31, 4)}
eta_130 | Layer  6: Extra in split (inactive in main) | Binning = {(31, 4)}
eta_130 | Layer  7: Extra in split (inactive in main) | Binning = {(15, 4)}
eta_130 | Layer  8: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer  9: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer 12: Match | Binning = (24, 14)
eta_130 | Layer 15: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer 17: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer 18: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer 19: Extra in split (inactive in main) | Binning = {(9, 4)}
eta_130 | Layer 20: Extra in split (inactive in main) | Binning = {(9, 4)}
```

For the active layers in common, the (r, phi) binning is the same

Note combined_fine has 2x these values for r and phi binning

Comparing the 3 complete datasets (Incident Energies)





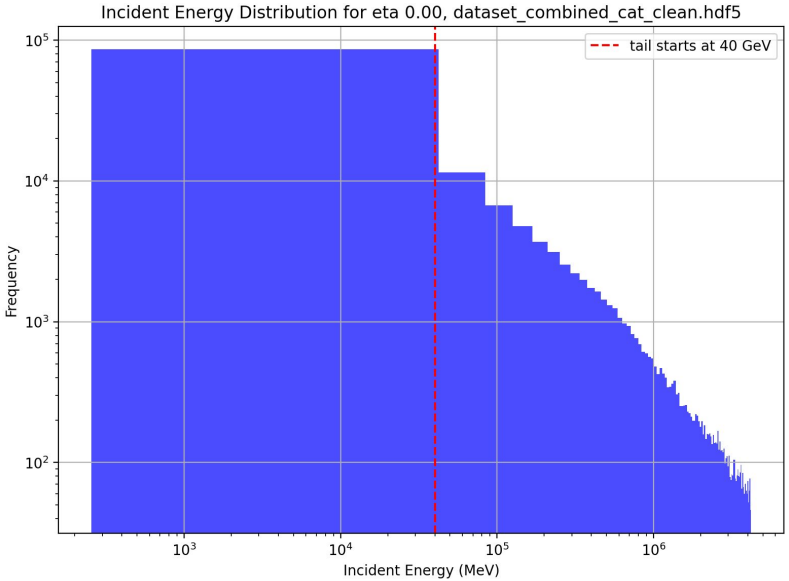
Confirming incident energies are identical across the 3 datasets

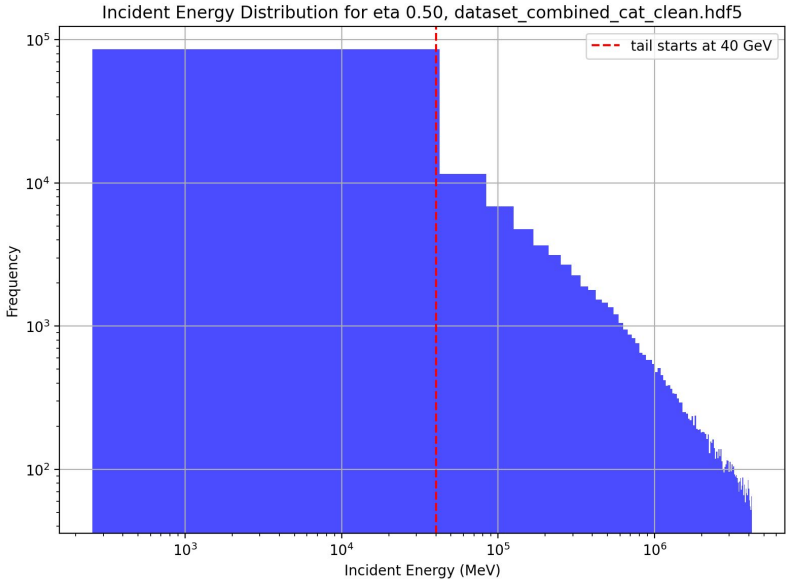
```python
# Check if incident energies are identical for each eta in original datasets
base_dir = "/fast_scratch_1/caloqvae/data/atlas_july31_cat"
for eta in range(0, 135, 5):
    file_names = ["dataset_combined_cat.hdf5", "dataset_positive_cat.hdf5", "dataset_fine_cat.hdf5"]
    incident_energies = []
    showers_list = []
    for file_name in file_names:
        with h5py.File(f"{base_dir}/eta_{eta:03d}/eta_{eta:03d}_regular_binning/{file_name}", 'r') as f:
            incident_energy = torch.from_numpy(f['incident_energy'][:])
            incident_energies.append(incident_energy)
            showers = torch.from_numpy(f['showers'][:]).sum(dim=1)
            showers_list.append(showers)


    # Check if all incident energies are the same
    if all(torch.equal(incident_energies[0], ie) for ie in incident_energies):
        print(f"Incident energies are identical for eta {eta}")
    else:
        print(f"Incident energies differ for eta {eta}")
```

```
Incident energies are identical for eta 0
Incident energies are identical for eta 5
Incident energies are identical for eta 10
Incident energies are identical for eta 15
Incident energies are identical for eta 20
Incident energies are identical for eta 25
Incident energies are identical for eta 30
Incident energies are identical for eta 35
Incident energies are identical for eta 40
Incident energies are identical for eta 45
Incident energies are identical for eta 50
Incident energies are identical for eta 55
Incident energies are identical for eta 60
Incident energies are identical for eta 65
Incident energies are identical for eta 70
Incident energies are identical for eta 75
Incident energies are identical for eta 80
Incident energies are identical for eta 85
Incident energies are identical for eta 90
Incident energies are identical for eta 95
Incident energies are identical for eta 100
Incident energies are identical for eta 105
Incident energies are identical for eta 110
Incident energies are identical for eta 115
Incident energies are identical for eta 120
Incident energies are identical for eta 125
Incident energies are identical for eta 130
```

Incidence Energy Distribution

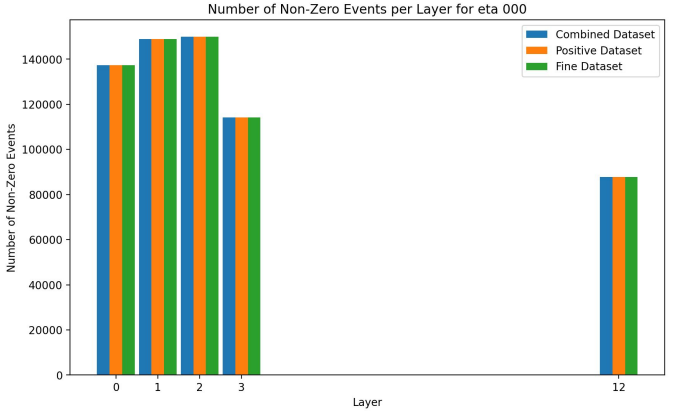
- Uniform on log-log axes in the beginning (power law distribution)
- Tail starting at 40 GeV

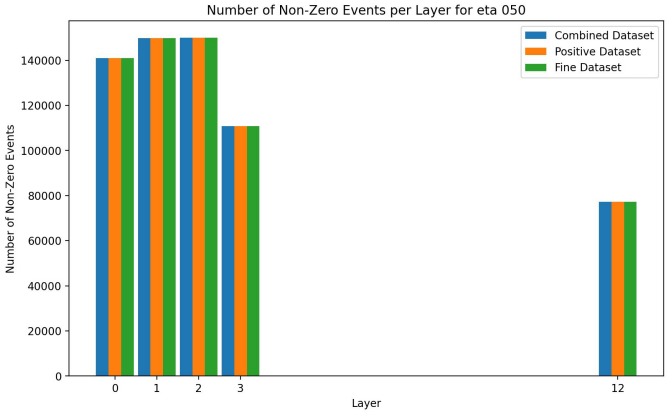


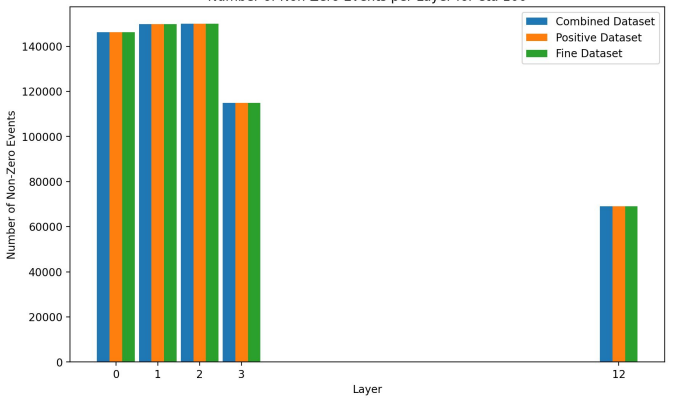


Comparing the 3 complete datasets (activated layers)

- Across all eta, same 5 layers [0, 1, 2, 3, 12] are activated
- Activations are identical across the 3 datasets

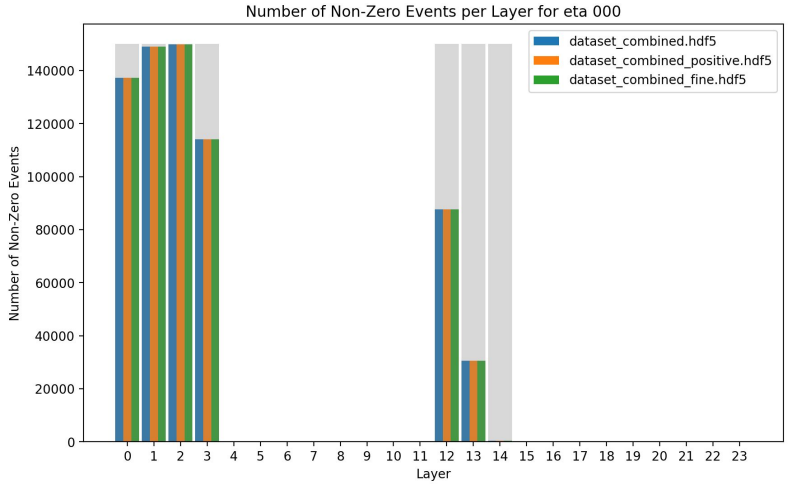


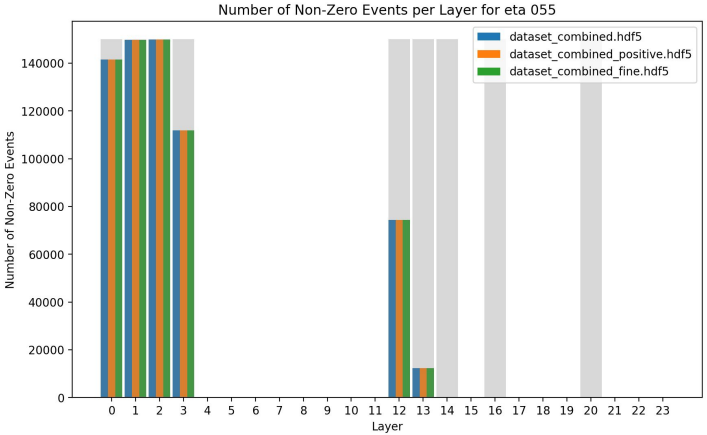


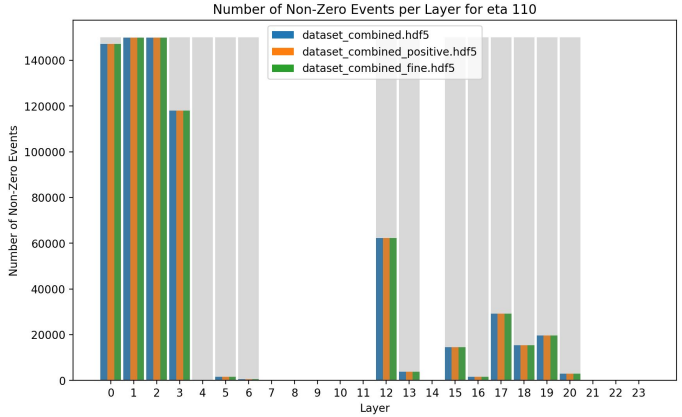


Rebuilt Datasets (Activated Layers)

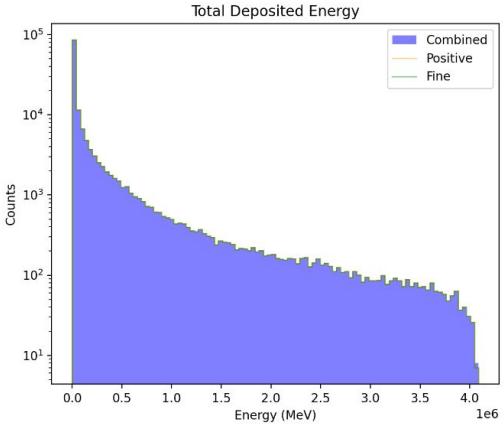
- More activated layers than the complete datasets
- Number of activated layers increases with eta
- Identical for the 3 dataset types

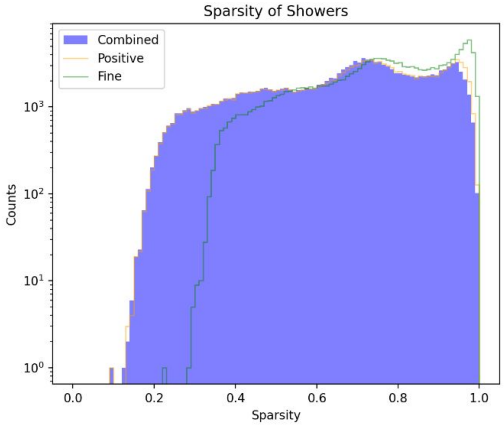


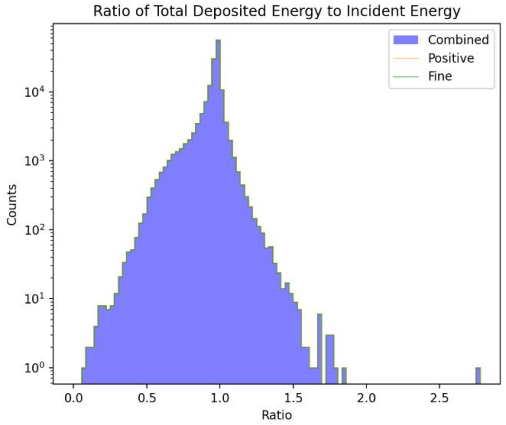


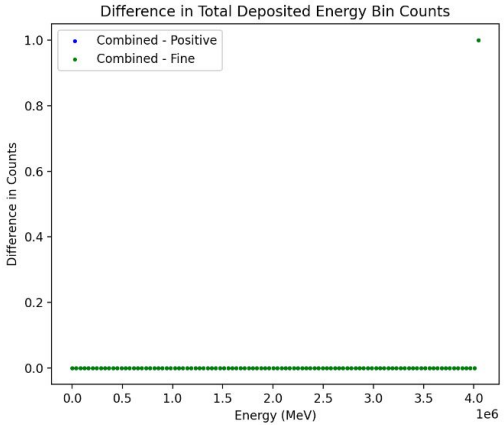


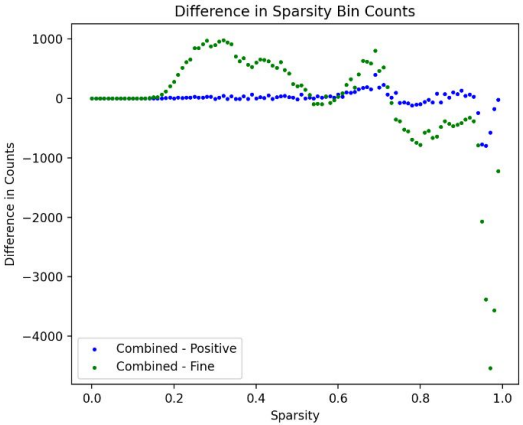
Comparing the 3 complete datasets (showers)

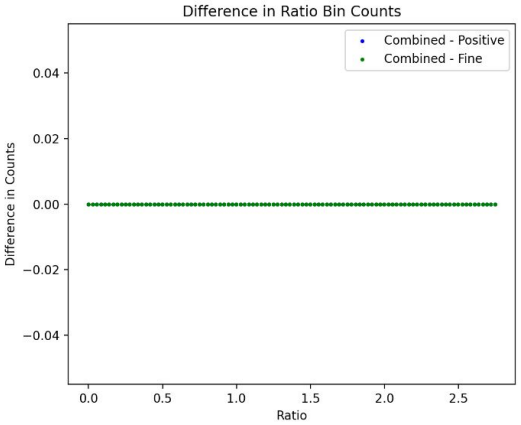






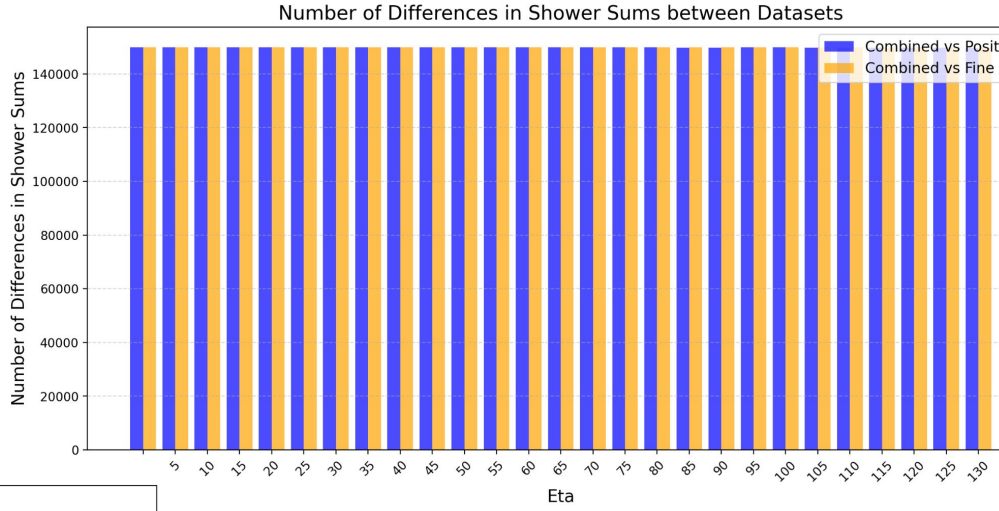


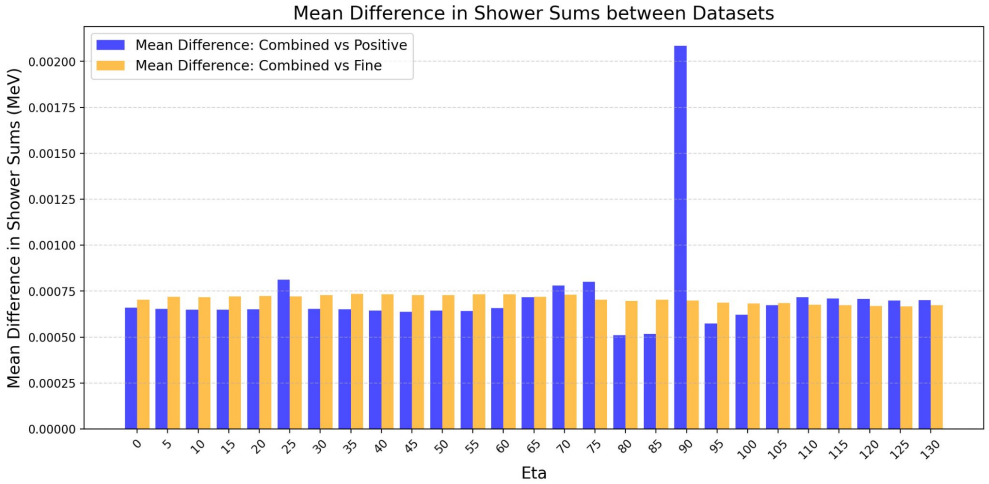




Comparing Energy Sums

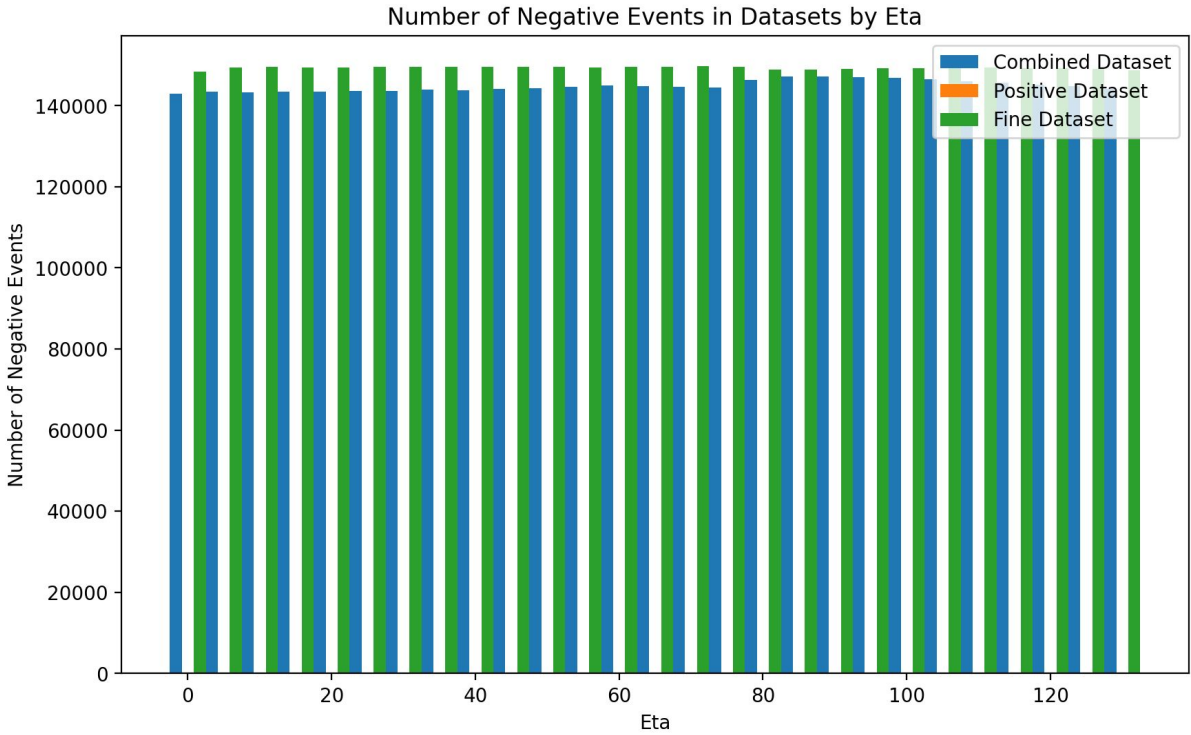
- Distribution is the same, but sums are subtly different



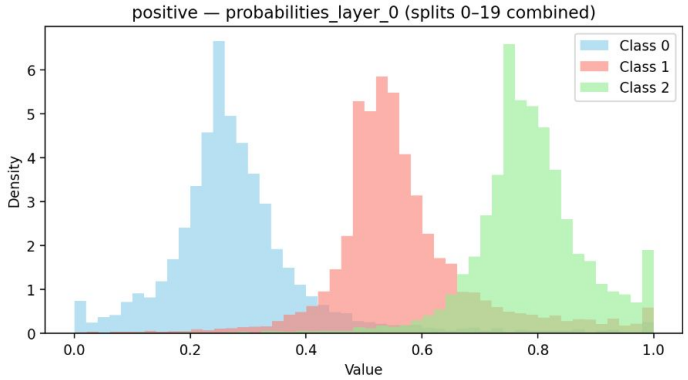


combined vs combined_positive

- Key difference is negative voxel counts
- Negative voxels present in nearly all events in the 2 non-positive datasets
- Masking process is unknown

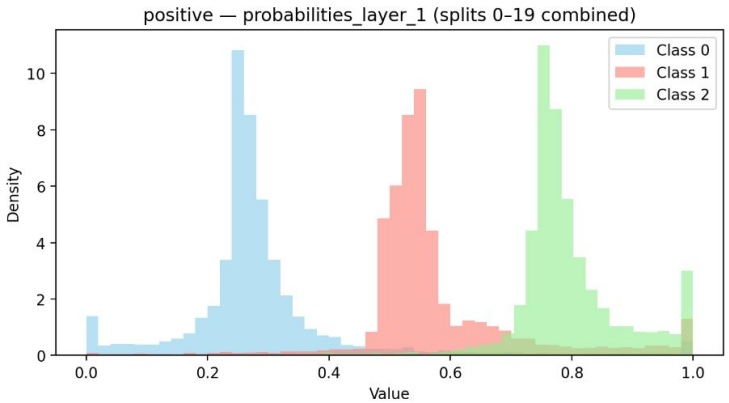


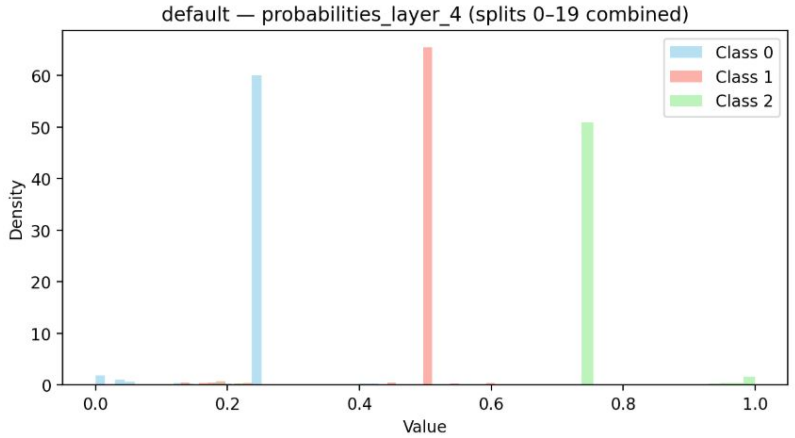
Layer Probabilities



- Additional information in the rebuilt files with layer probabilities
- Not in the fine binning files
- Non active layers are mainly 0.25, 0.50, and 0.75

```
probabilities_layer_0: shape=(336, 3),
probabilities_layer_1: shape=(336, 3),
probabilities_layer_10: shape=(36, 3),
probabilities_layer_11: shape=(36, 3),
probabilities_layer_12: shape=(336, 3)
probabilities_layer_13: shape=(36, 3),
probabilities_layer_14: shape=(36, 3),
probabilities_layer_15: shape=(36, 3),
probabilities_layer_16: shape=(36, 3),
probabilities_layer_17: shape=(36, 3),
probabilities_layer_18: shape=(36, 3),
probabilities_layer_19: shape=(36, 3),
probabilities_layer_2: shape=(336, 3),
probabilities_layer_20: shape=(36, 3),
probabilities_layer_21: shape=(36, 3),
probabilities_layer_22: shape=(36, 3),
probabilities_layer_23: shape=(36, 3),
probabilities_layer_3: shape=(336, 3),
probabilities_layer_4: shape=(36, 3),
probabilities_layer_5: shape=(124, 3),
probabilities_layer_6: shape=(124, 3),
probabilities_layer_7: shape=(60, 3),
probabilities_layer_8: shape=(36, 3),
probabilities_layer_9: shape=(36, 3),
```

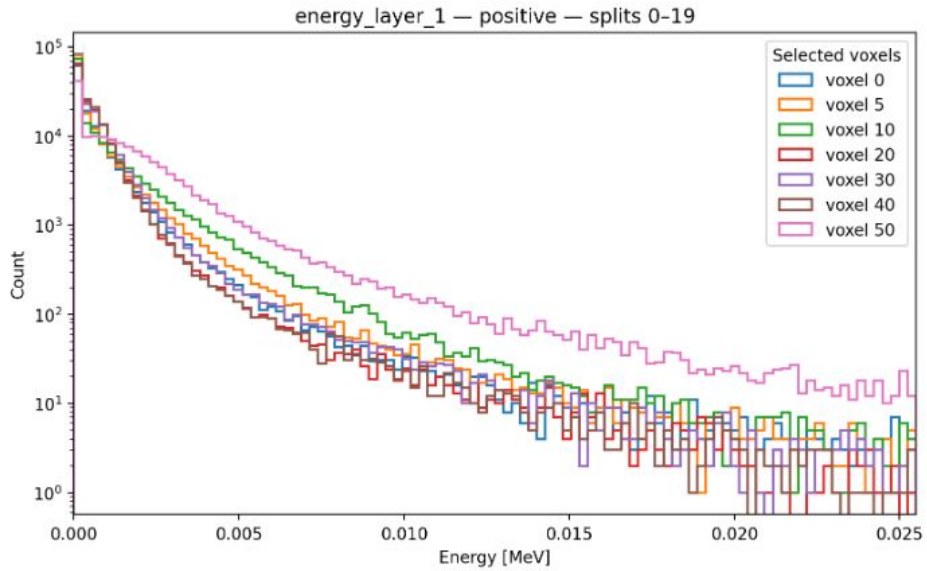


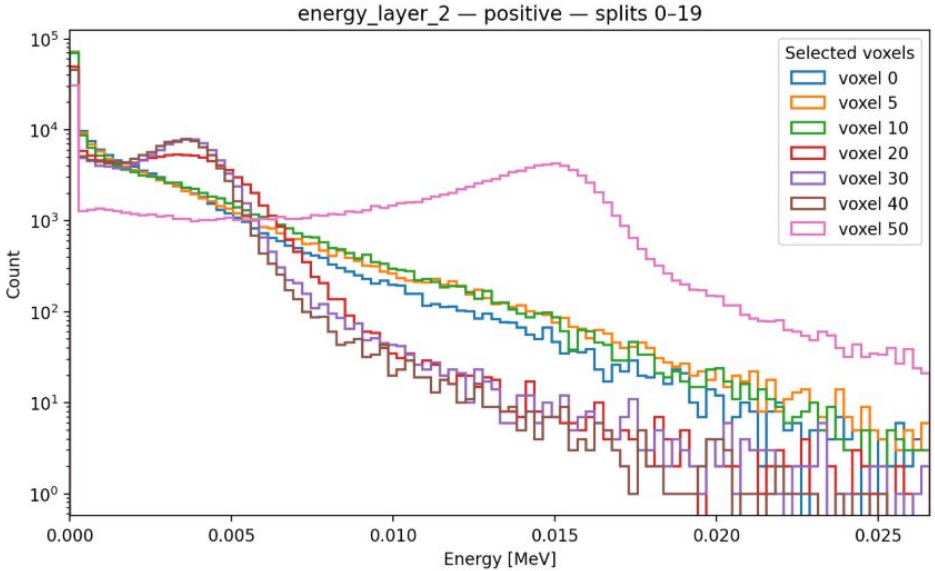


Specific Layers

- Looking at specific voxels and the probabilities

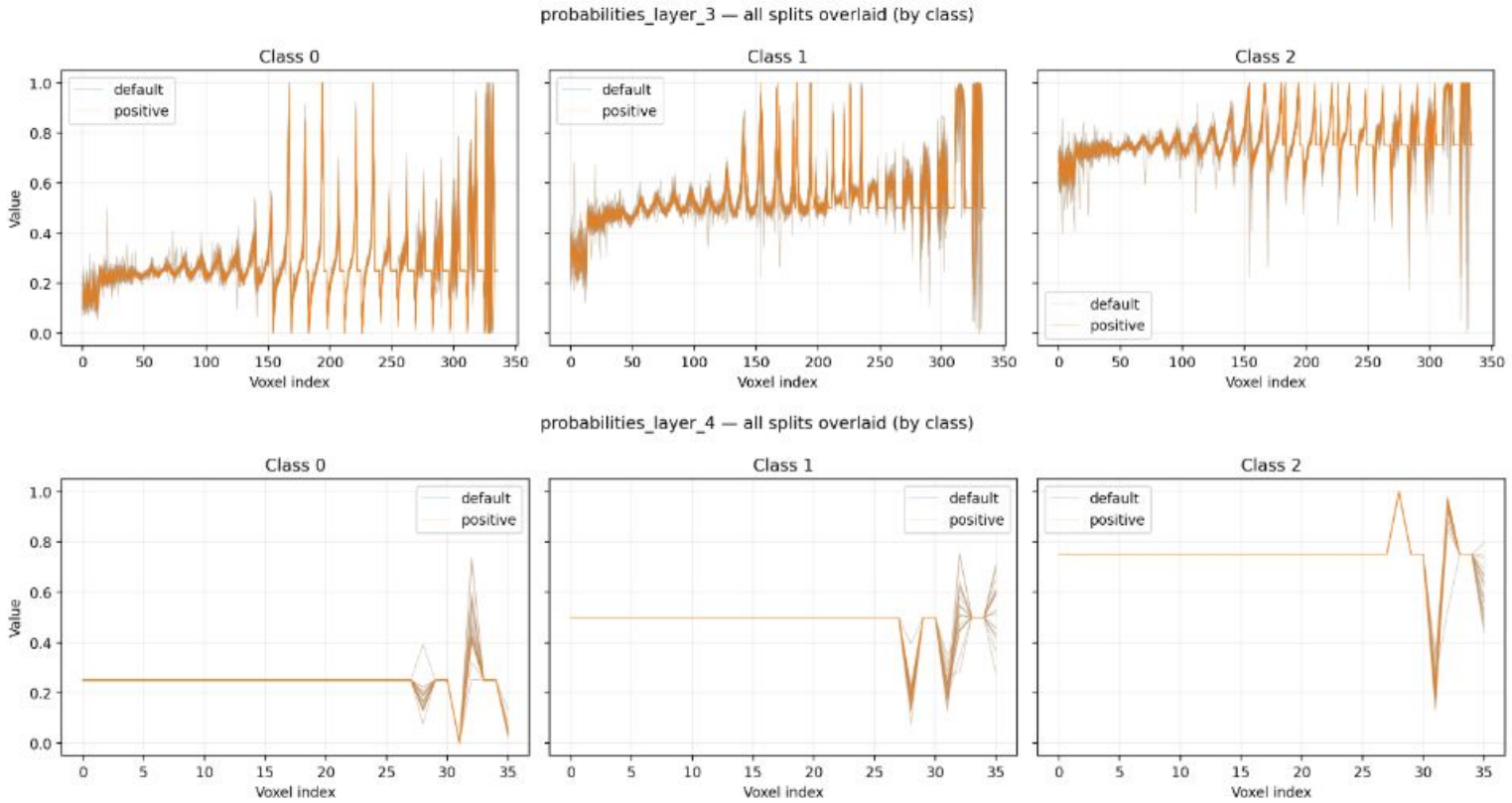
```
energy_layer_2:
  voxel    0  --   [0.1378, 0.387728, 0.703464] |
  voxel    5  --   [0.0698482, 0.212591, 0.603662]
  voxel   10  --   [0.129619, 0.278331, 0.629237]
  voxel   20  --   [0.295808, 0.465479, 0.691706]
  voxel   30  --   [0.239064, 0.443037, 0.72888] |
  voxel   40  --   [0.24554, 0.481176, 0.752892] |
  voxel   50  --   [0.226929, 0.443161, 0.734117]
```





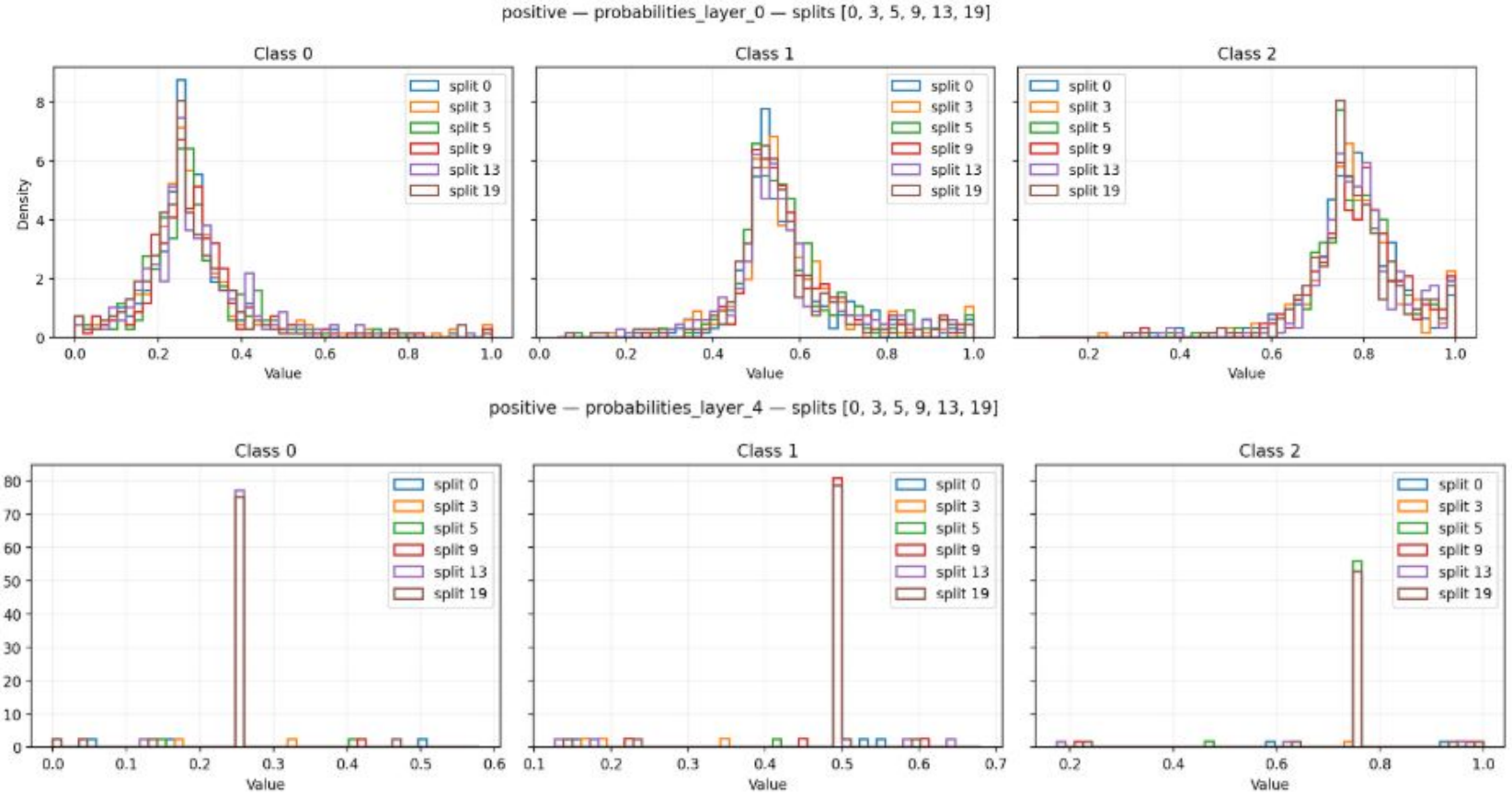
Probabilities and Voxel Indices

Inactive layers have probabilities of mainly 0.25, 0.50, and 0.75



Probabilities

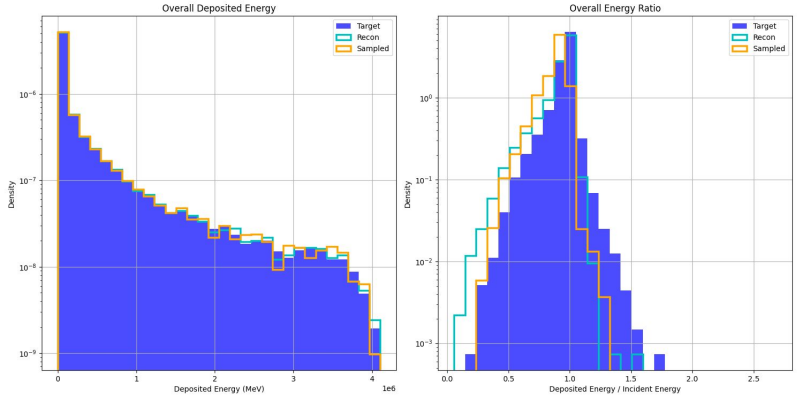
Rebuilt files 0-19 have similar but slightly different probabilities for voxels

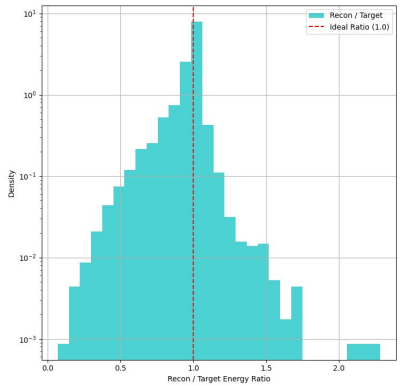


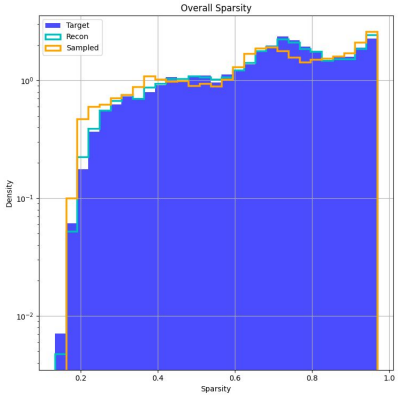
Processing/Work Done on New Dataset

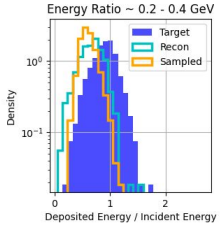
- Filtered out events where voxel energy > incident energy
- Adapted decoder and encoder to handle z=5
- Training successfully on new dataset

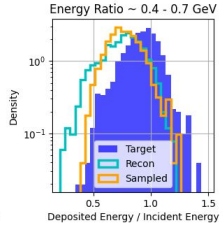
Model Performance on New Dataset

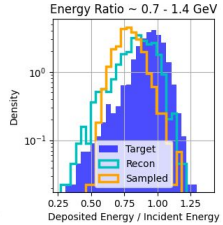


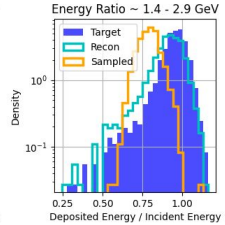


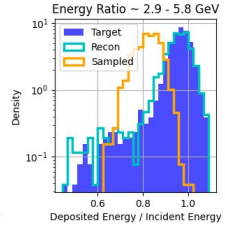


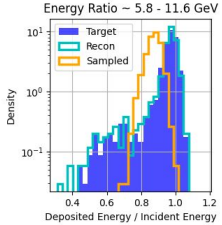


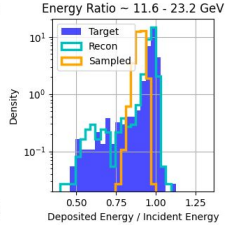


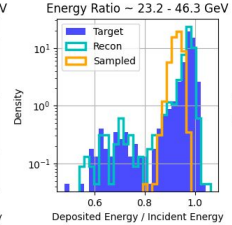


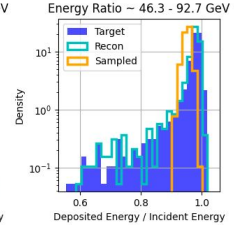


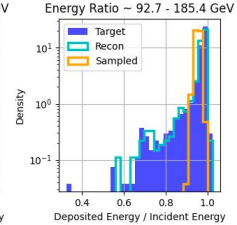


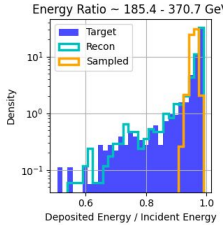


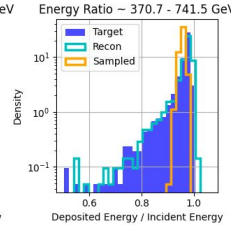


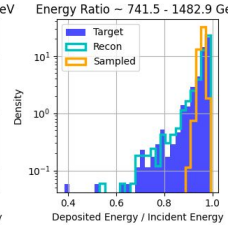


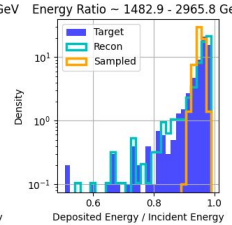


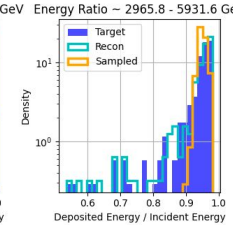






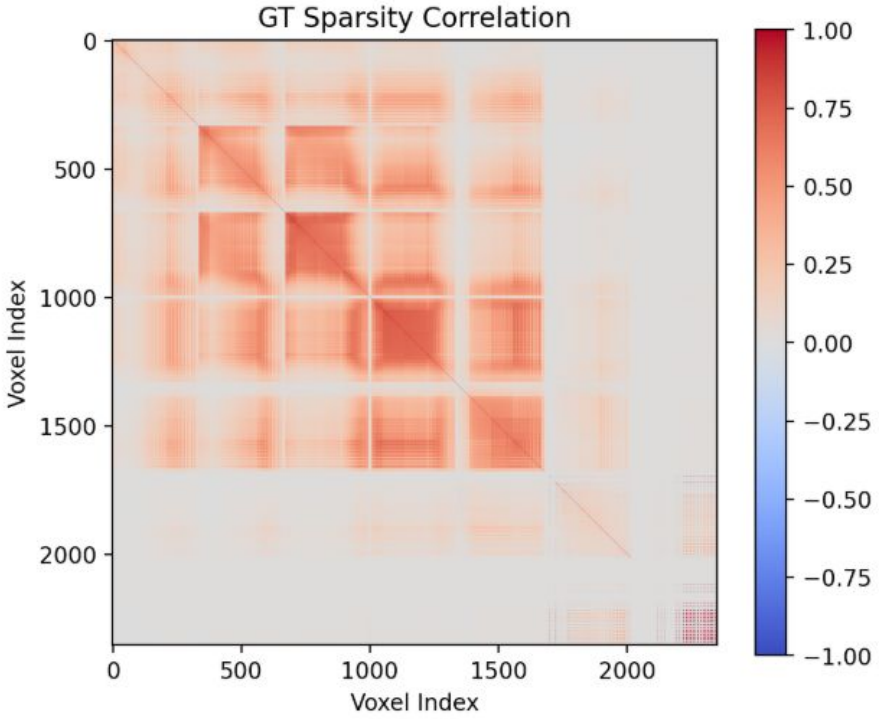


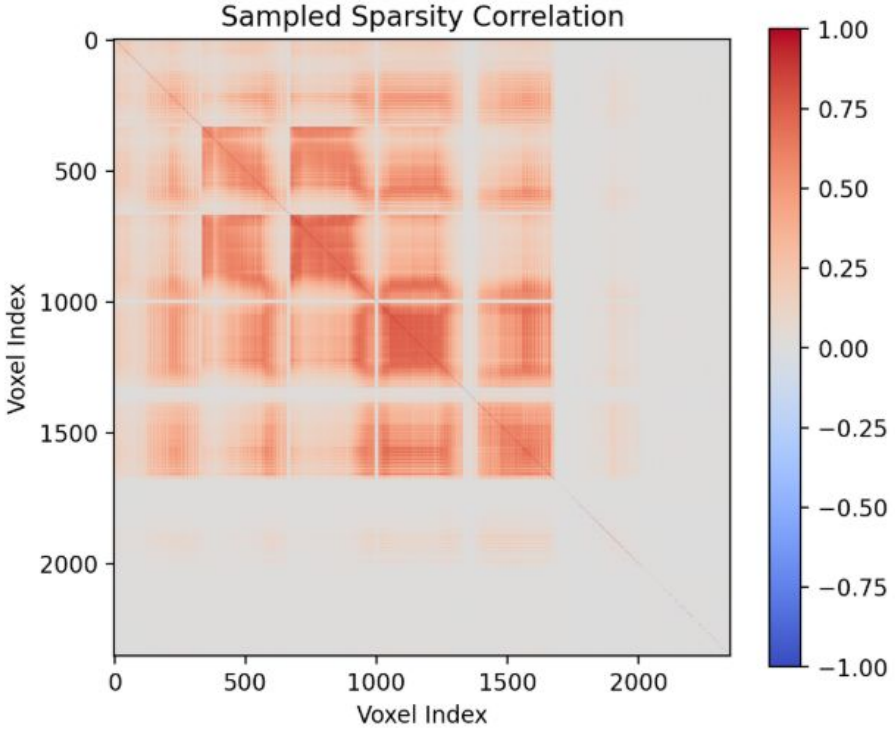




Sparsity Correlation/Frobenius

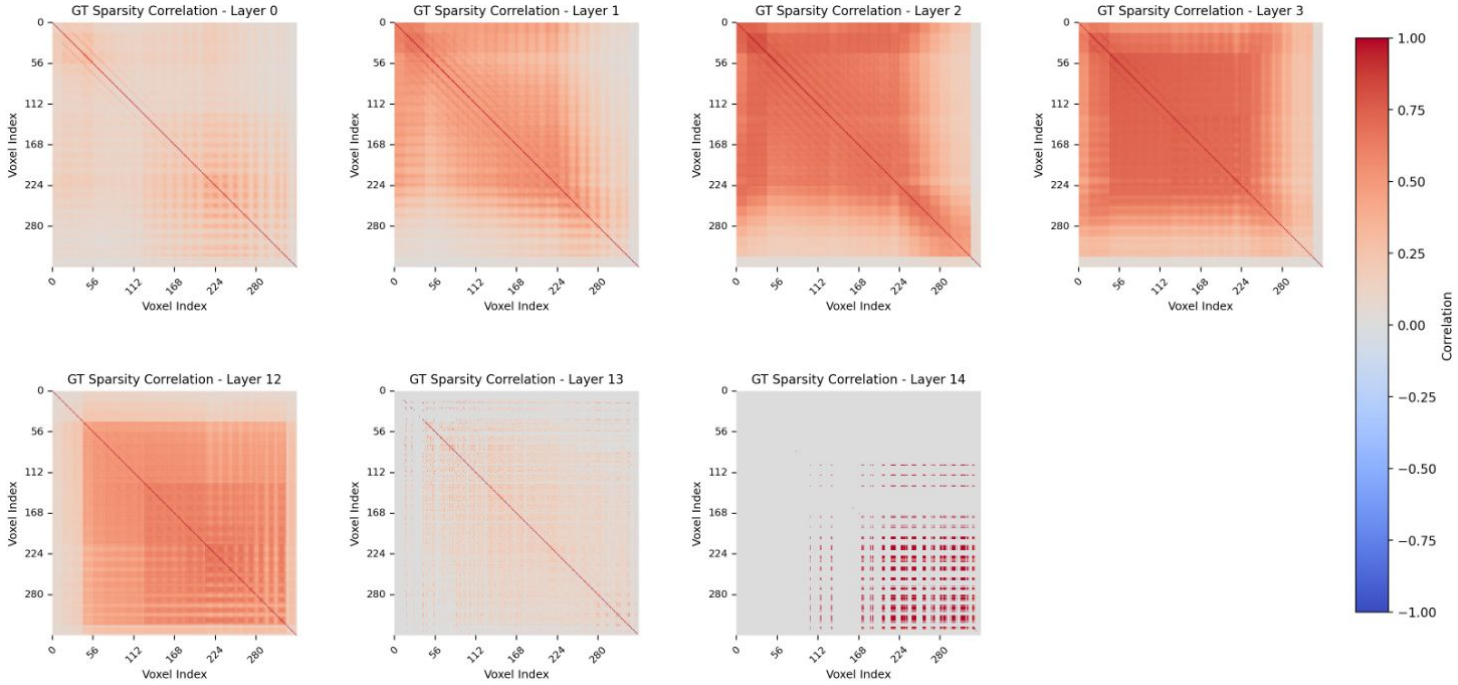
Now implemented for sparsity and voxel energy
Computed voxel-wise
Epoch 239 - Computing one Frobenius metric





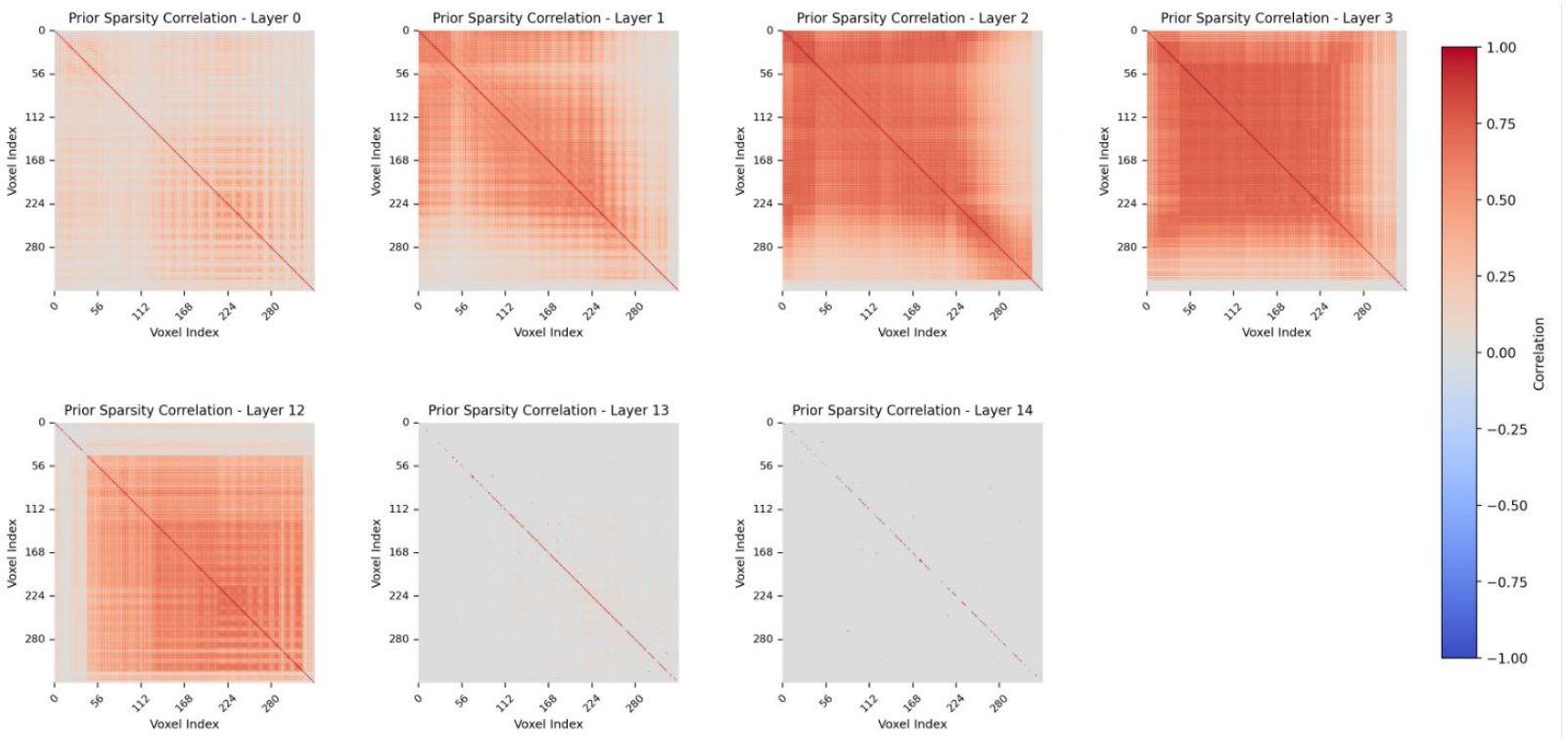
Sparsity Correlation (Layer Evaluation)

Also split up by layer and compute a Frobenius metric for each layer



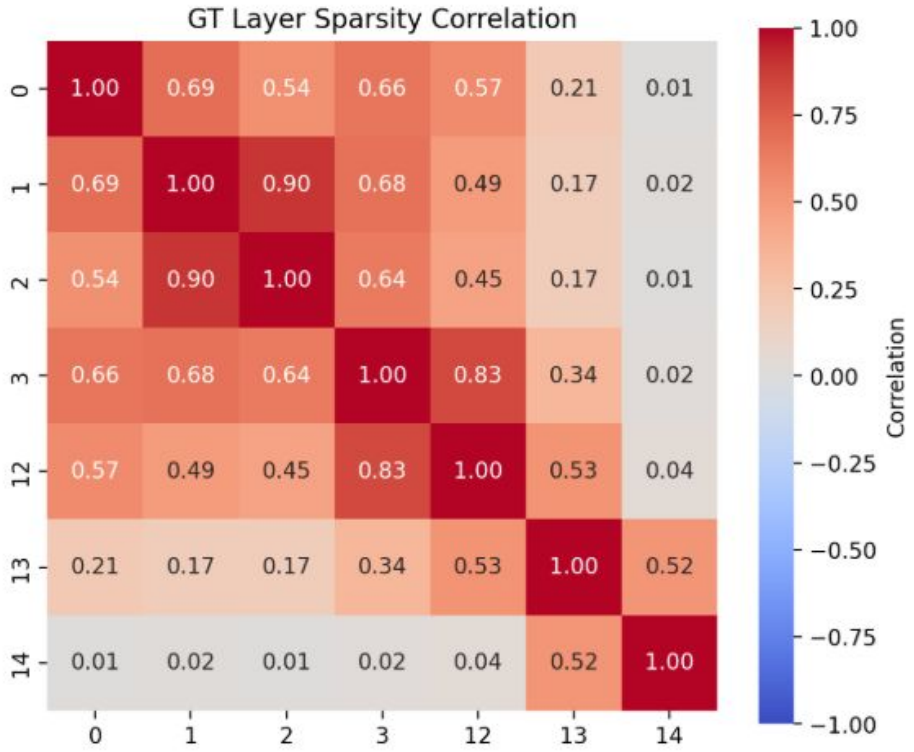
Sparsity (Sampled)

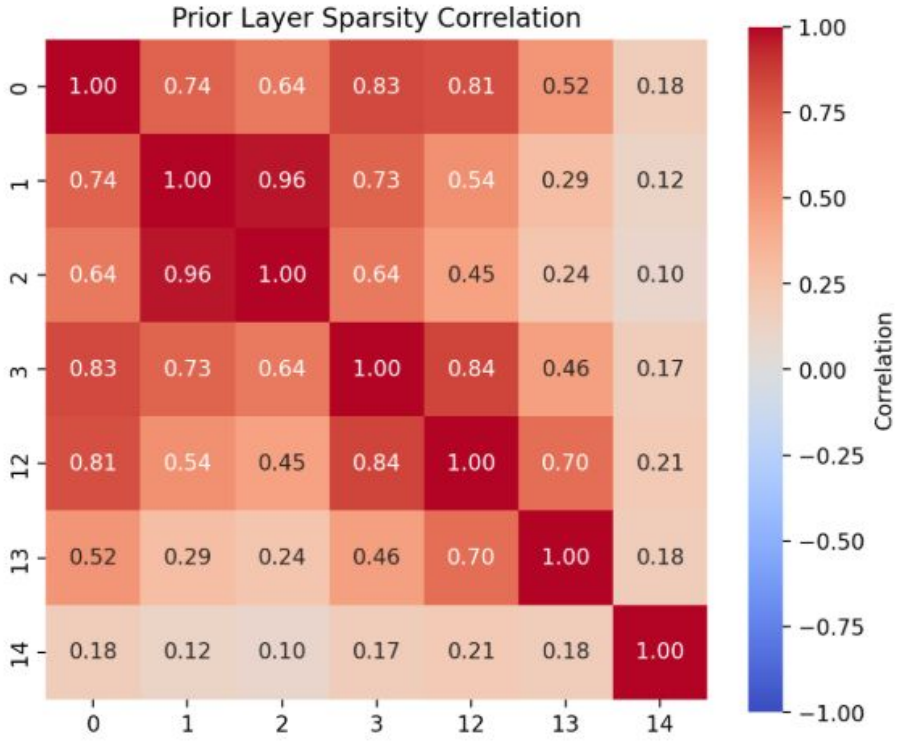
For prior samples:



Layer Sparsity

Taking average sparsity per layer and obtaining a layer-wise correlation matrix to compute a Frobenius metric (this is also done for voxel energy)

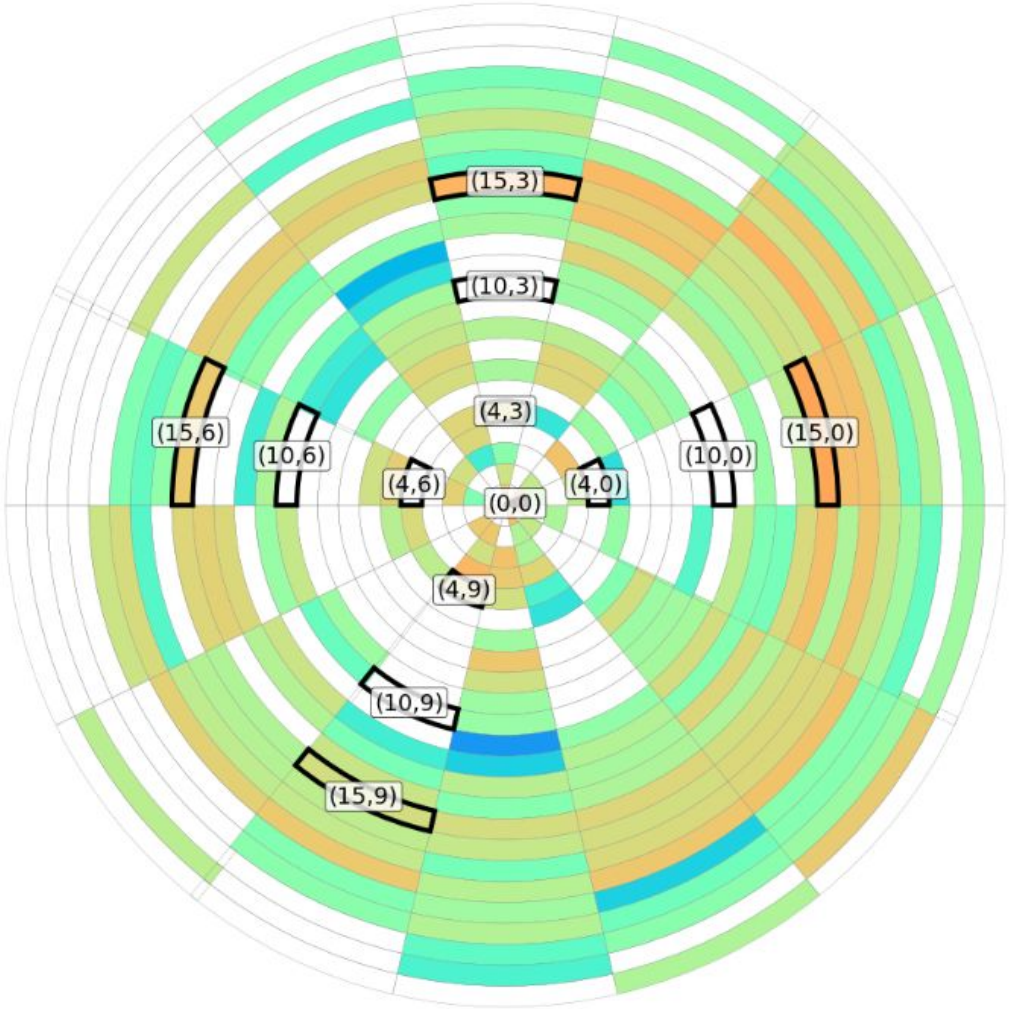




Patches

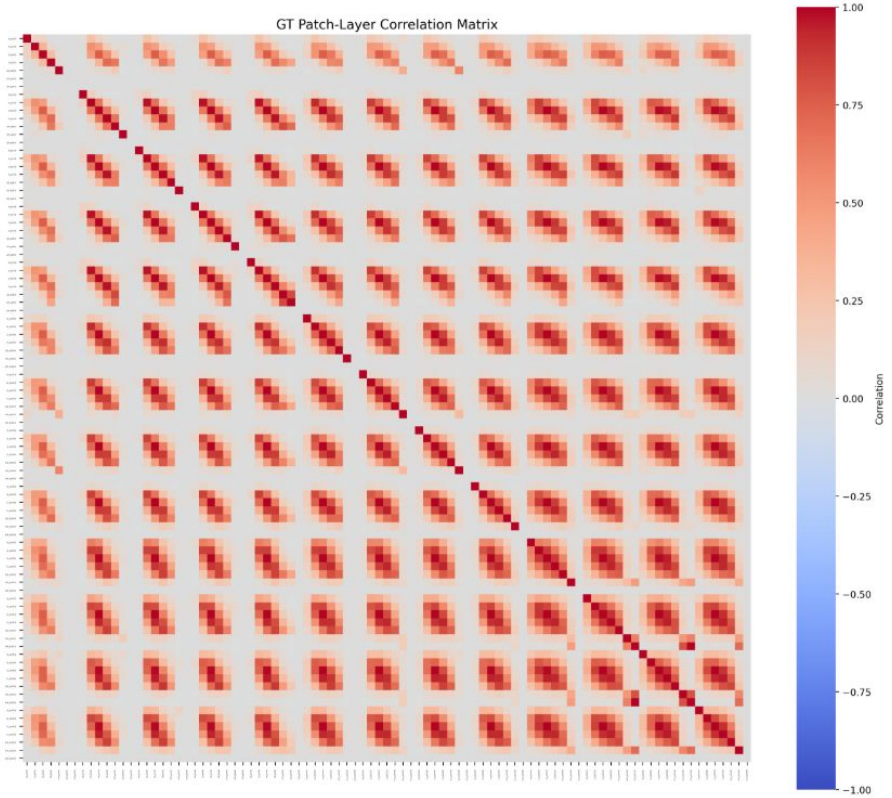
Performing correlations between patches and layers

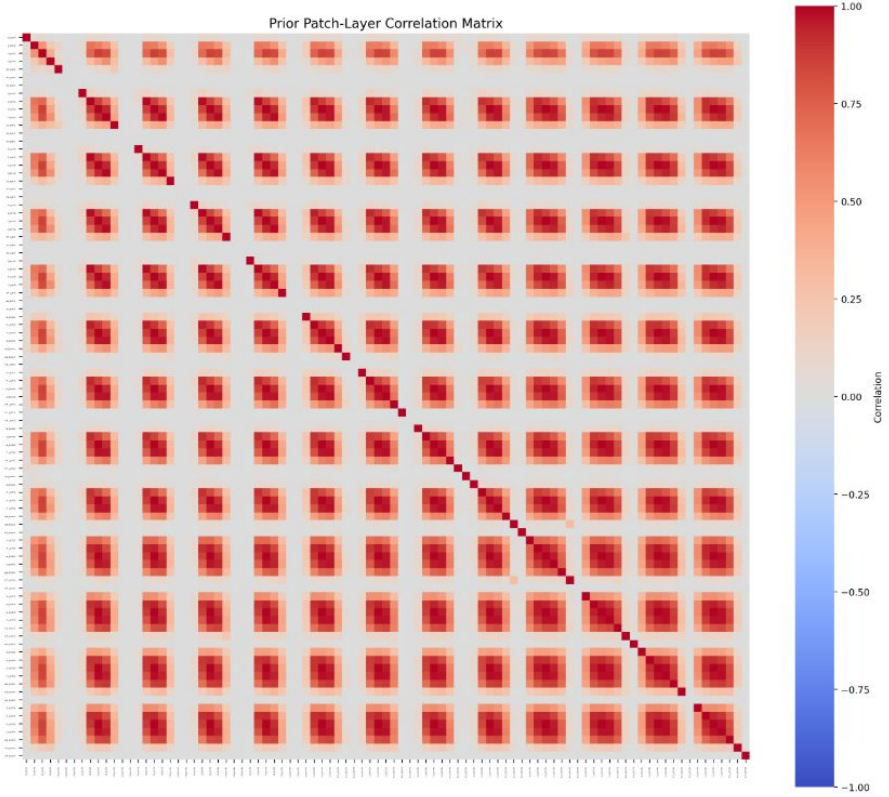
Selected patches shown here:



Patches (Correlations)

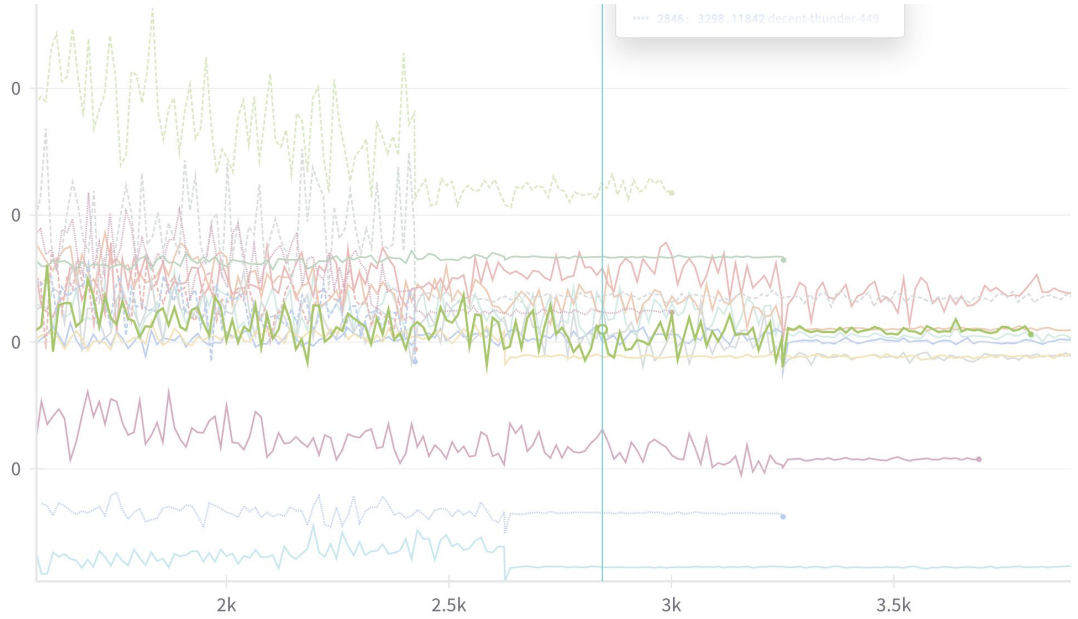
Correlations between patches and layers
Frobenius metric computed





Updates on Decoder: Two Heads are Better than One

- Removed heads in all subdecoders except for last
- No regularization performance on par with old models with regularization



GeoDecoder Updates

- Added second head to first subdecoder
- Regularized performance best so far
- TODO: KPD and FPD