

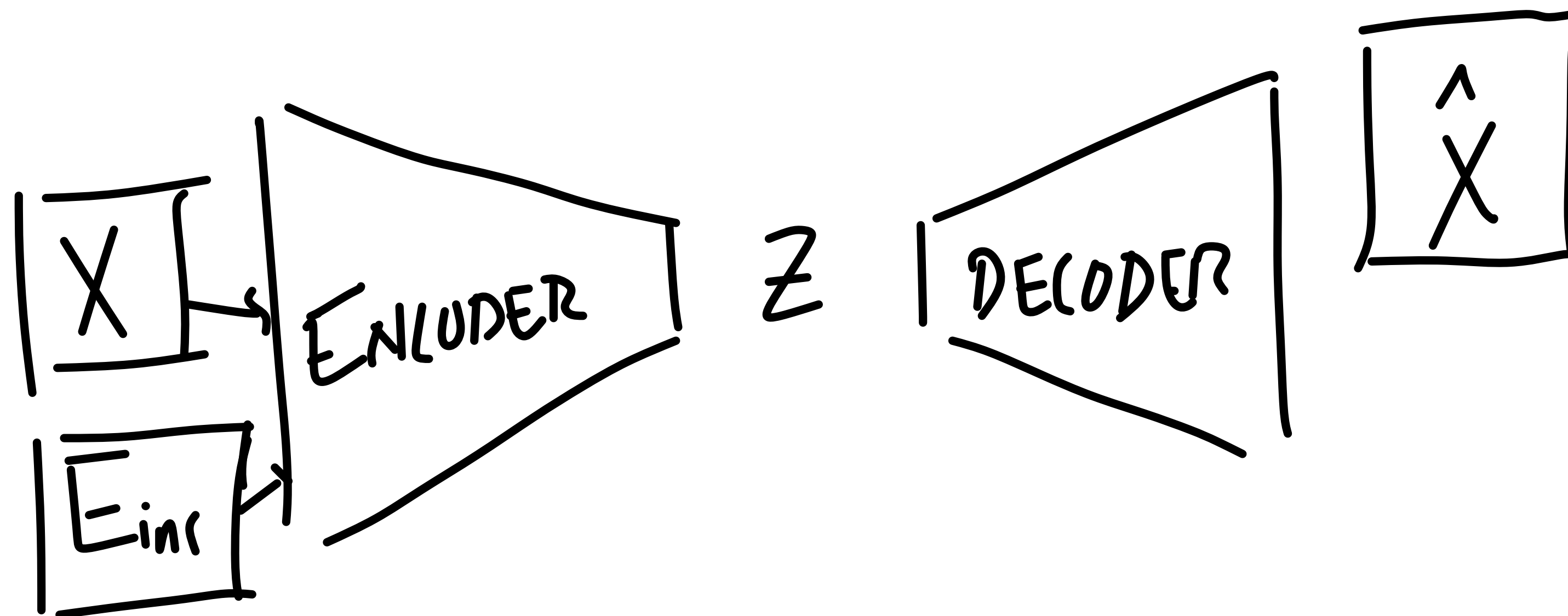
Calo4pQVAE: Progress and updates

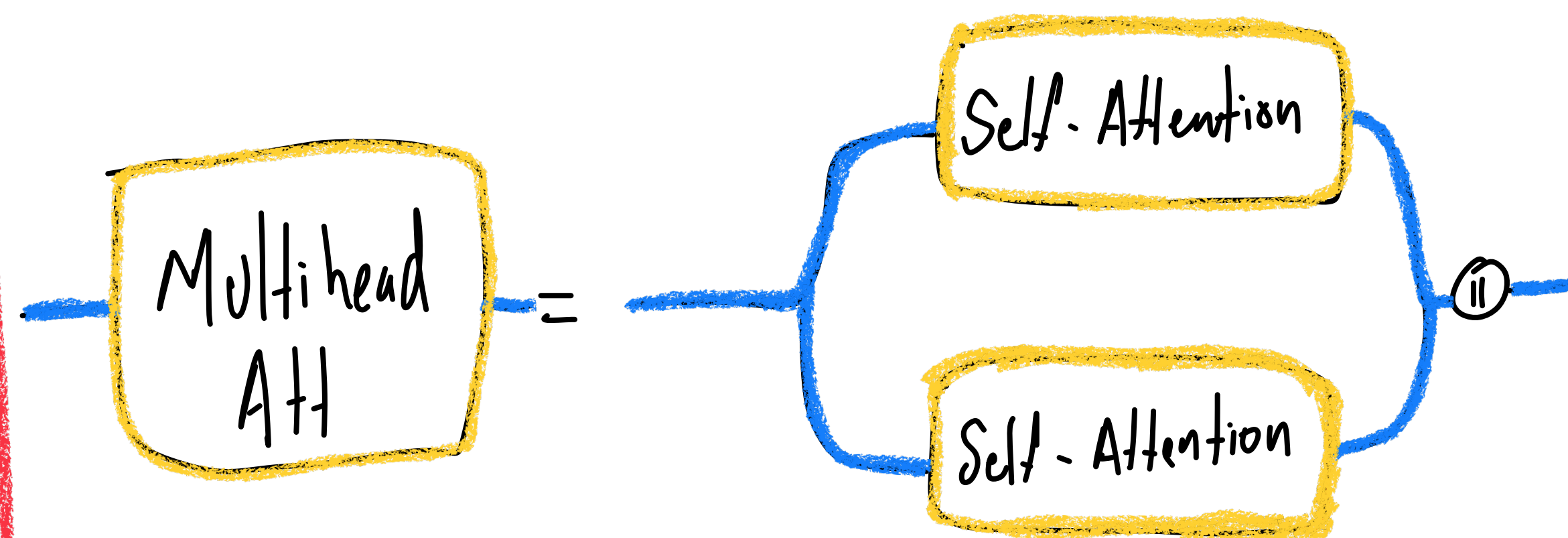
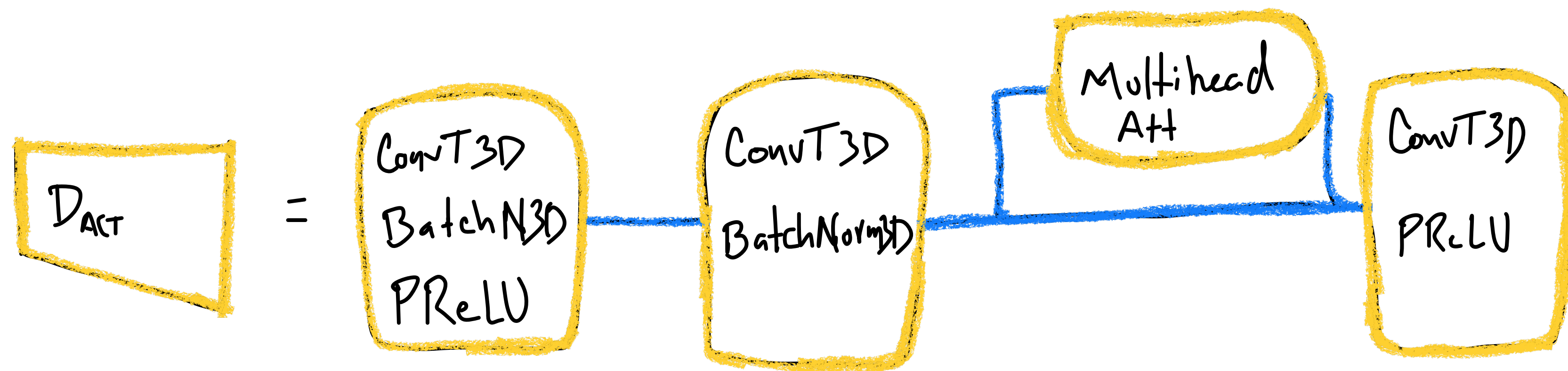
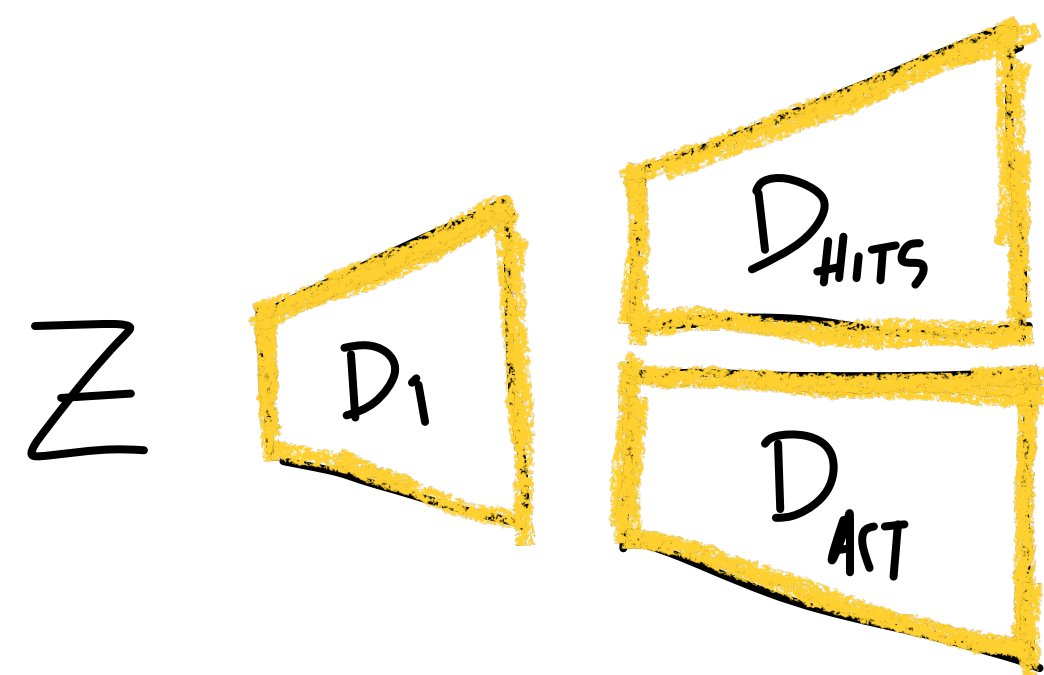


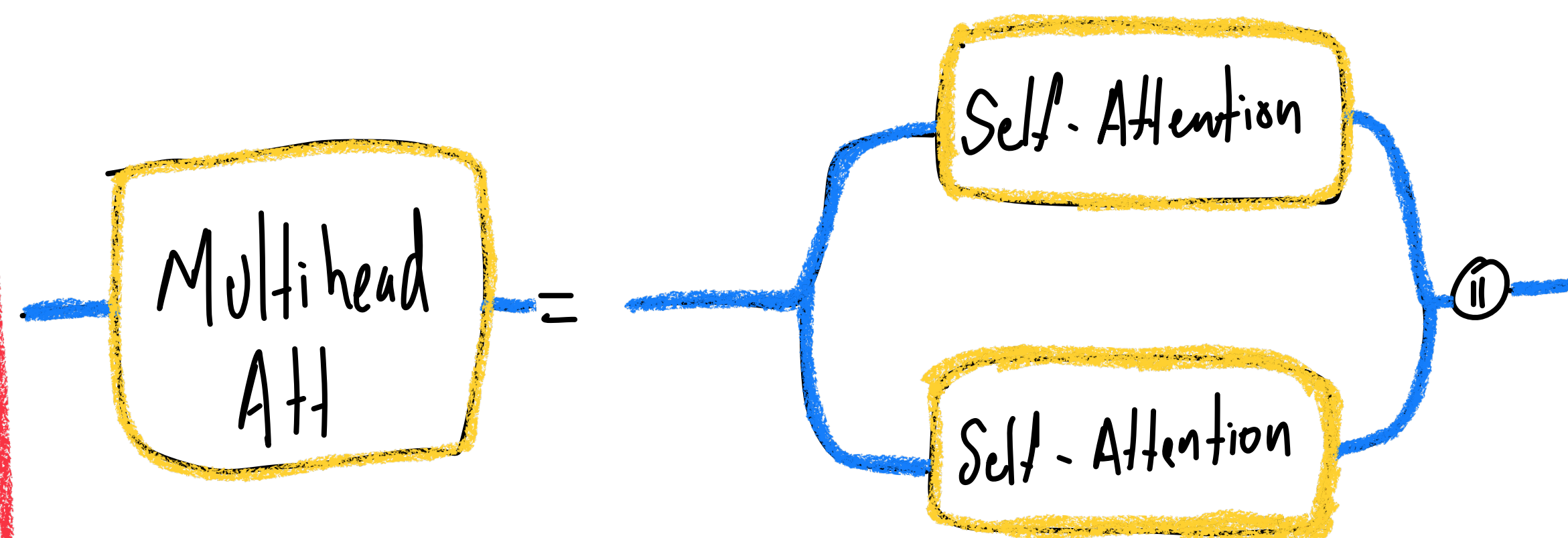
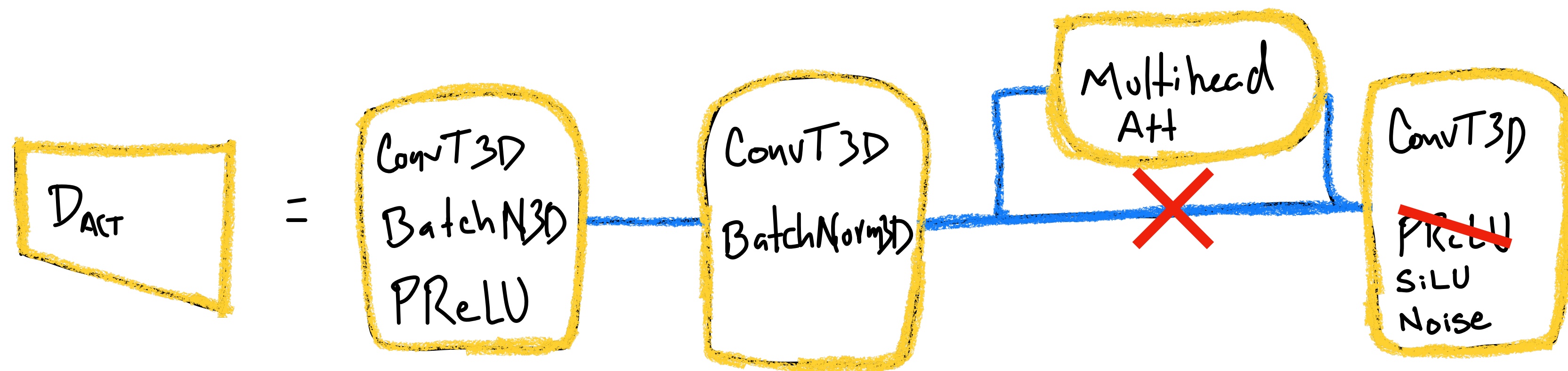
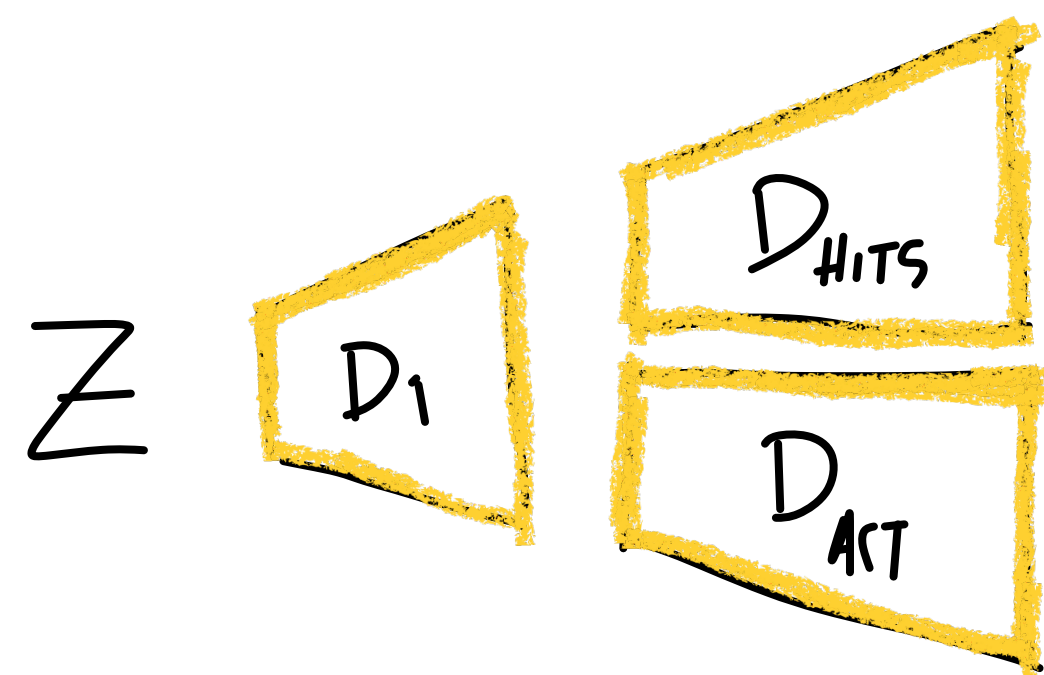
Feb 21 2025

New models

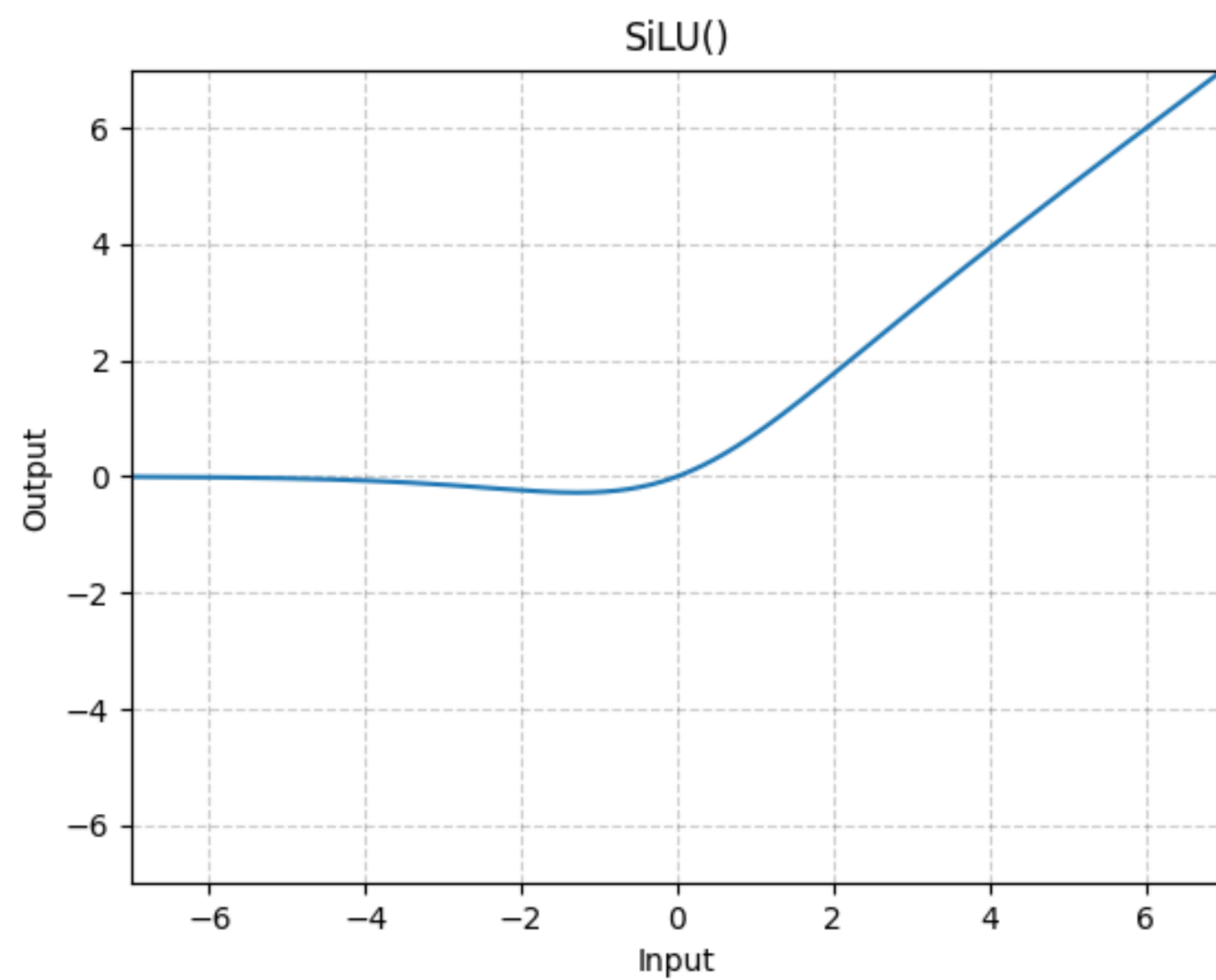
Self-attention







$\text{silu}(x) = x * \sigma(x)$, where $\sigma(x)$ is the logistic sigmoid.



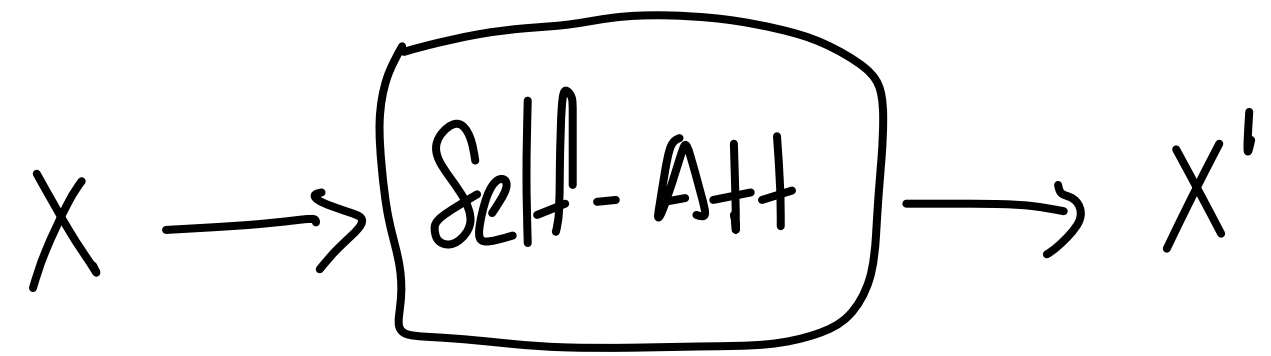
Noise($\vec{X}$):

return $\vec{X} = \vec{X} + \vec{\zeta}$

where $\langle \zeta_i \zeta_j \rangle = \delta_{ij}$

$\langle \zeta_i \rangle = 0$

How Does Self-Attention Work?



① X : tensor of Batch $\times$ Channels $\times$ Depth $\times$ Width $\times$ Height = $B \times C \times l \times h \times w$ (Rank 5)

```
class Head(nn.Module):
```

```
...
```

```
Self-attention block
```

```
...
```

```
def __init__(self, dim, head_size=16):
```

```
    super().__init__()
```

```
    self.head_size = head_size
```

```
    self.key = nn.Linear(dim, head_size, bias=False)
```

```
    self.query = nn.Linear(dim, head_size, bias=False)
```

```
    self.value = nn.Linear(dim, head_size, bias=False)
```

```
    # self.ln1 = nn.LayerNorm(dim)
```

```
    # self.ln2 = nn.LayerNorm(dim)
```

```
def forward(self, x):
```

```
    b, c, l, h, w = x.shape
```

```
    x = rearrange(x, "b c l h w -> b (l h w) c")
```

```
    # x = self.ln1(x)
```

```
    k = self.key(x)
```

```
    q = self.query(x)
```

```
    wei = q @ k.transpose(-2, -1) * self.head_size**-0.5
```

```
    wei = F.softmax(wei, dim=-1)
```

```
    v = self.value(x)
```

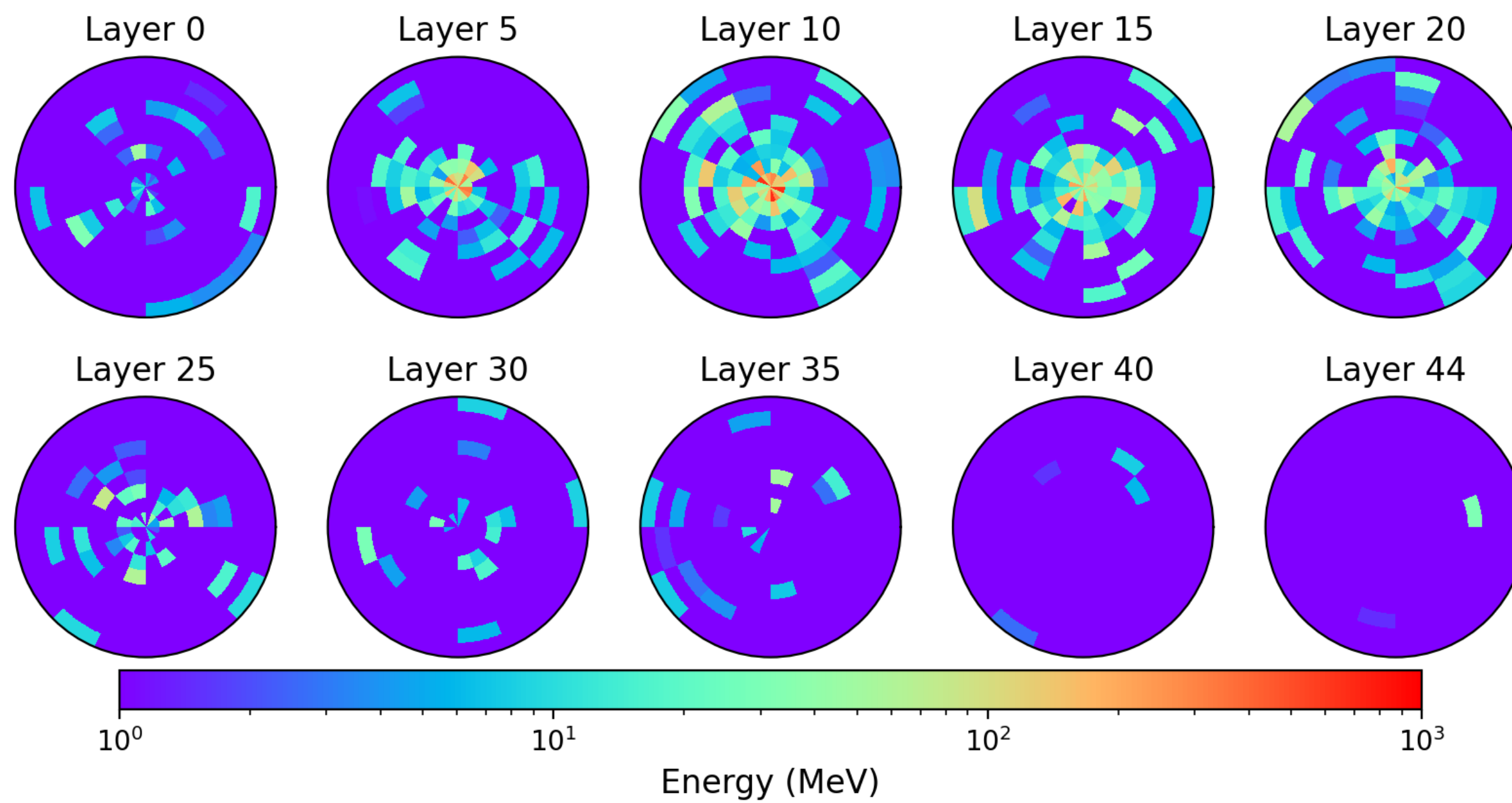
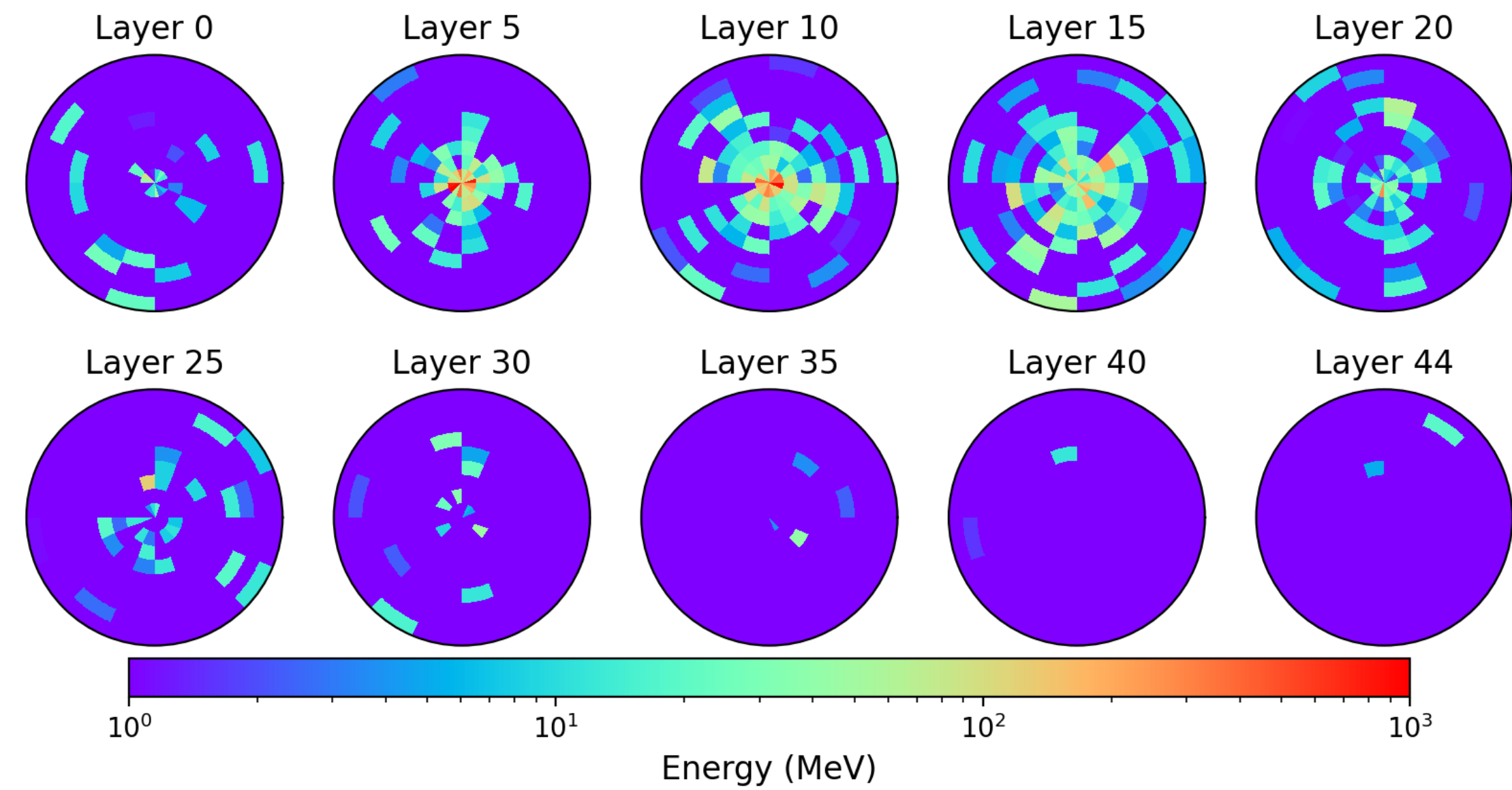
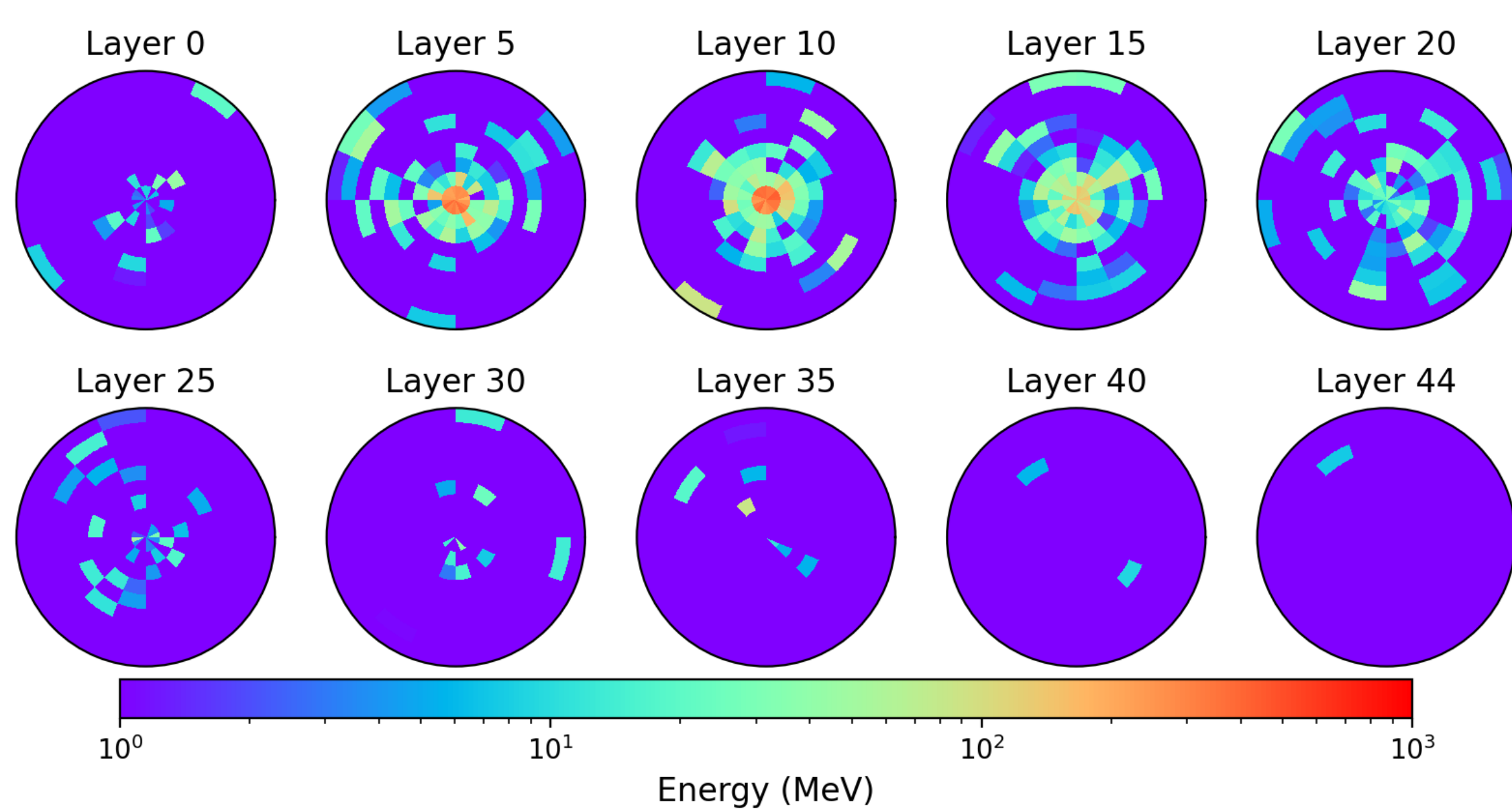
```
    # out = self.ln2(wei @ v)
```

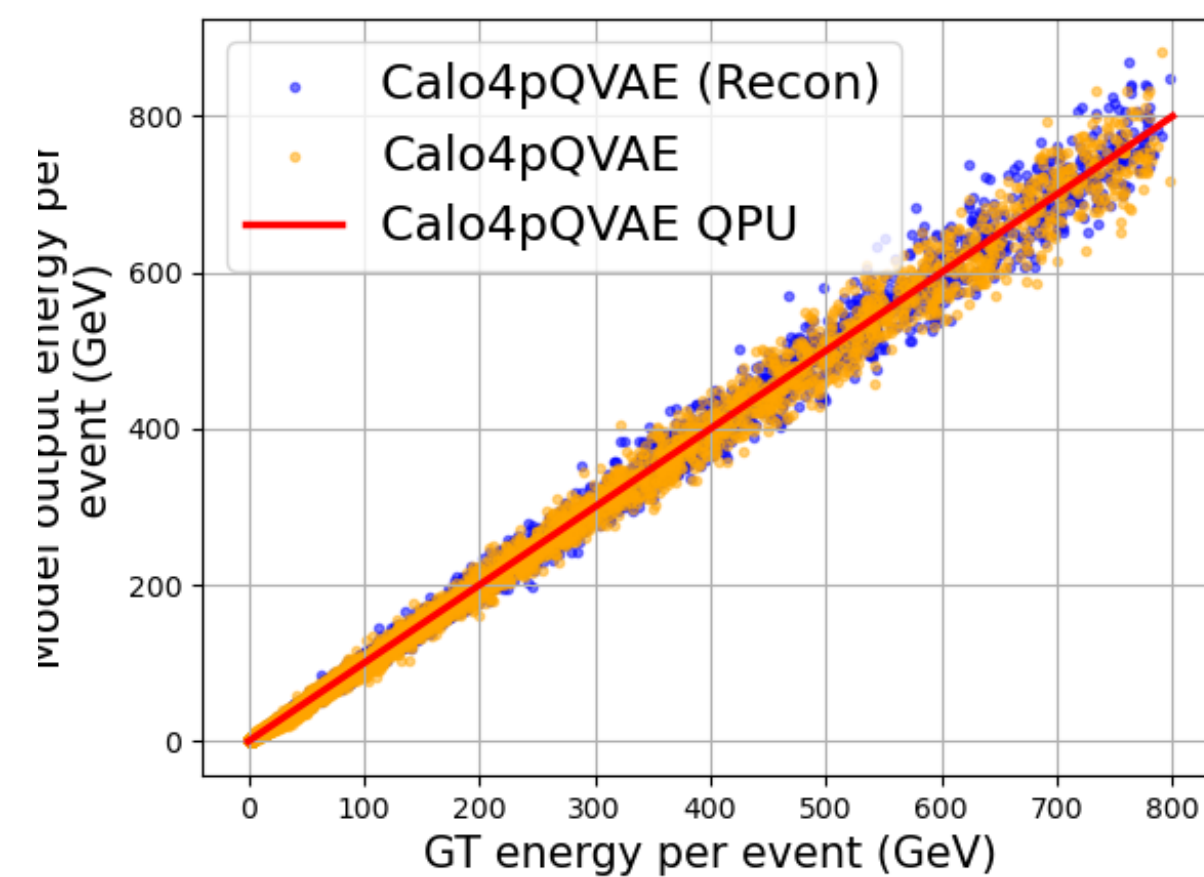
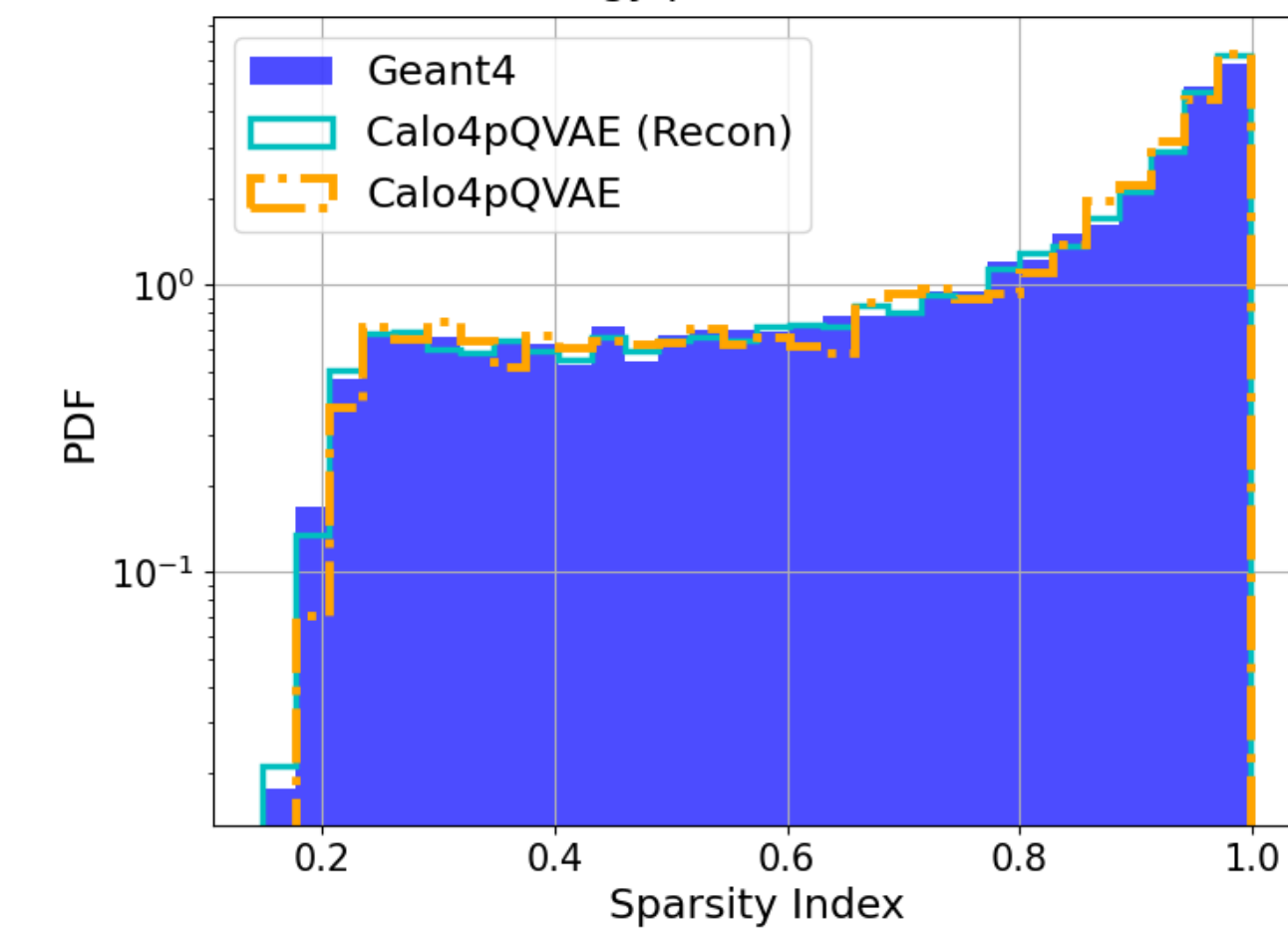
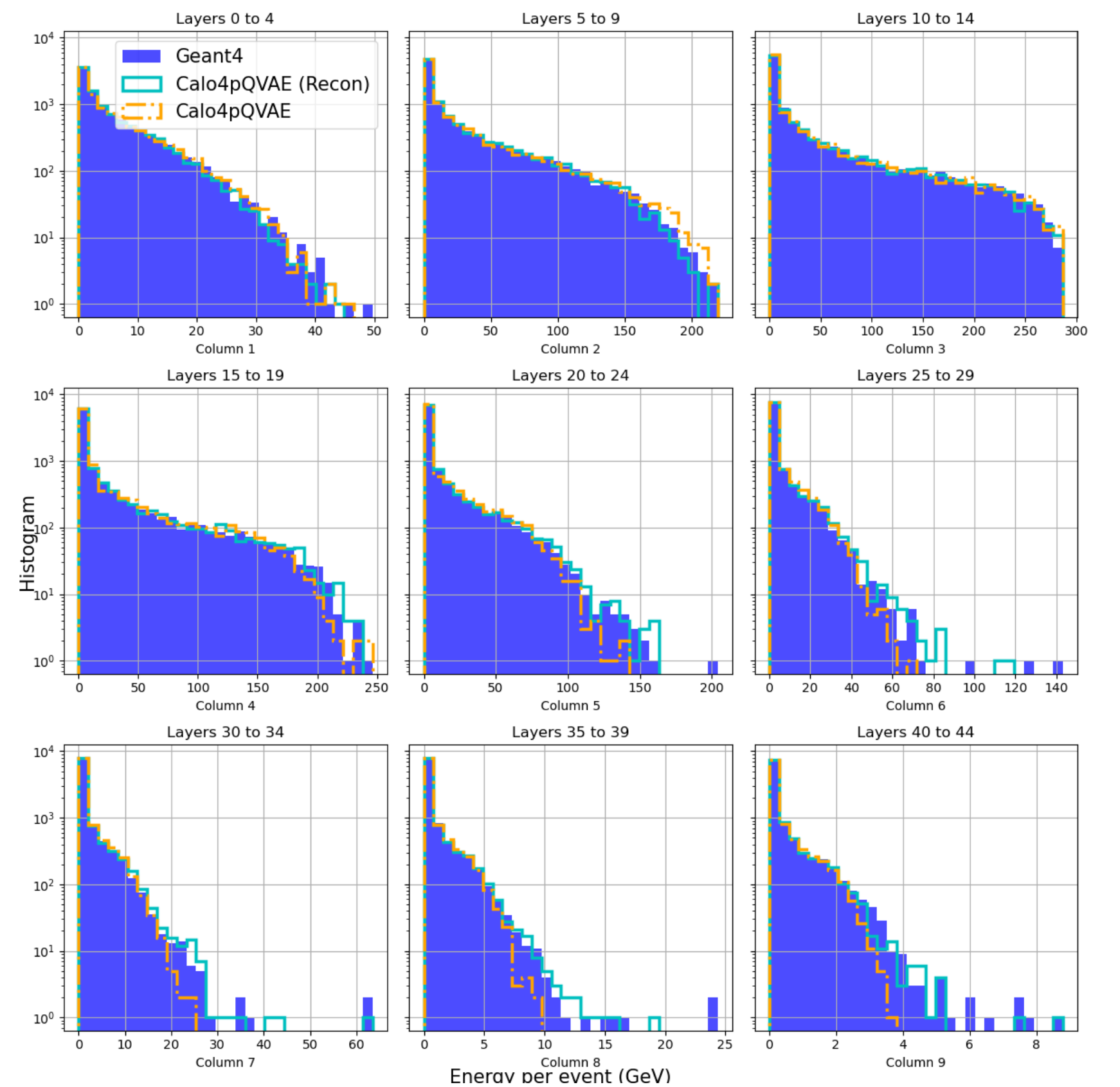
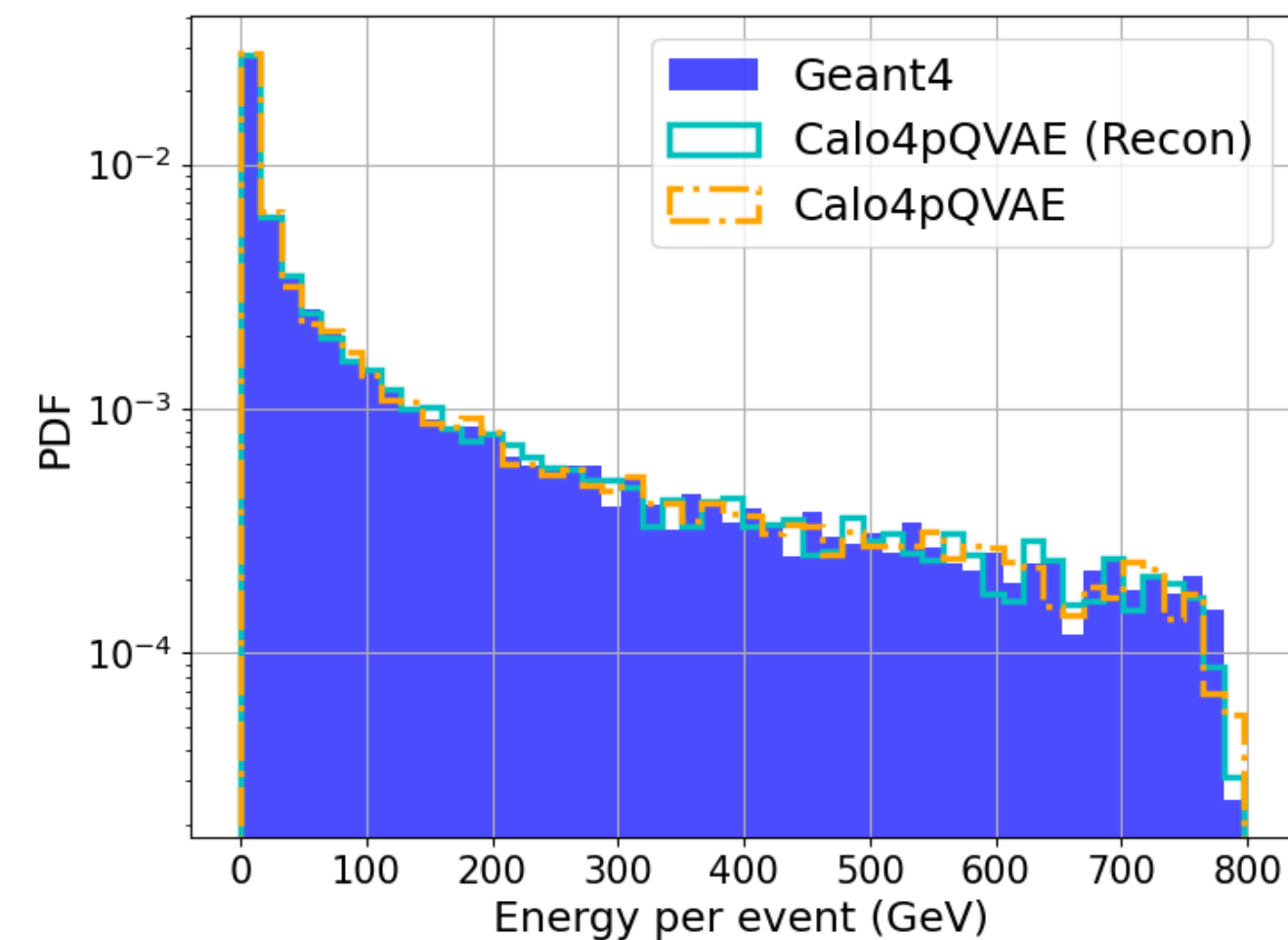
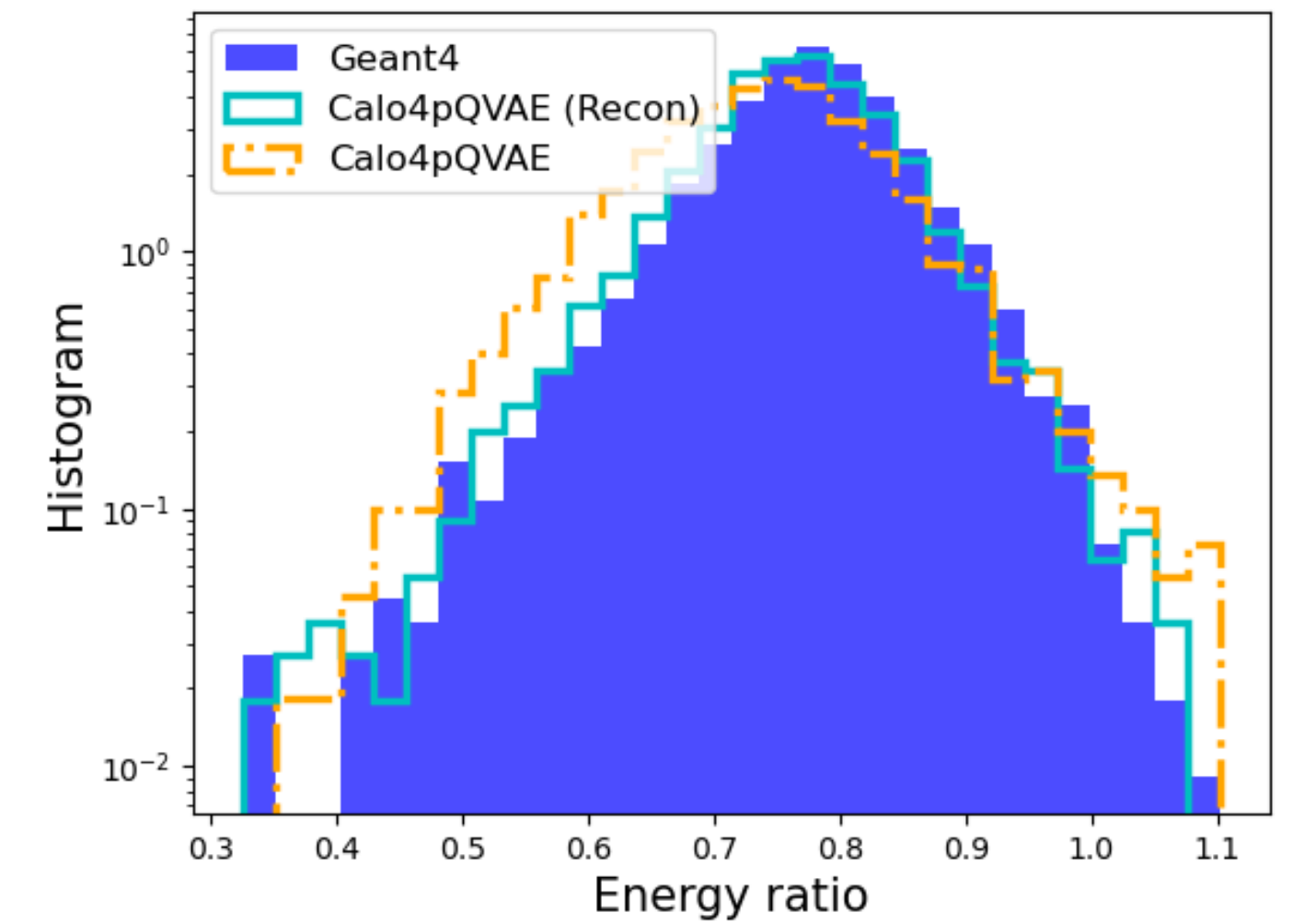
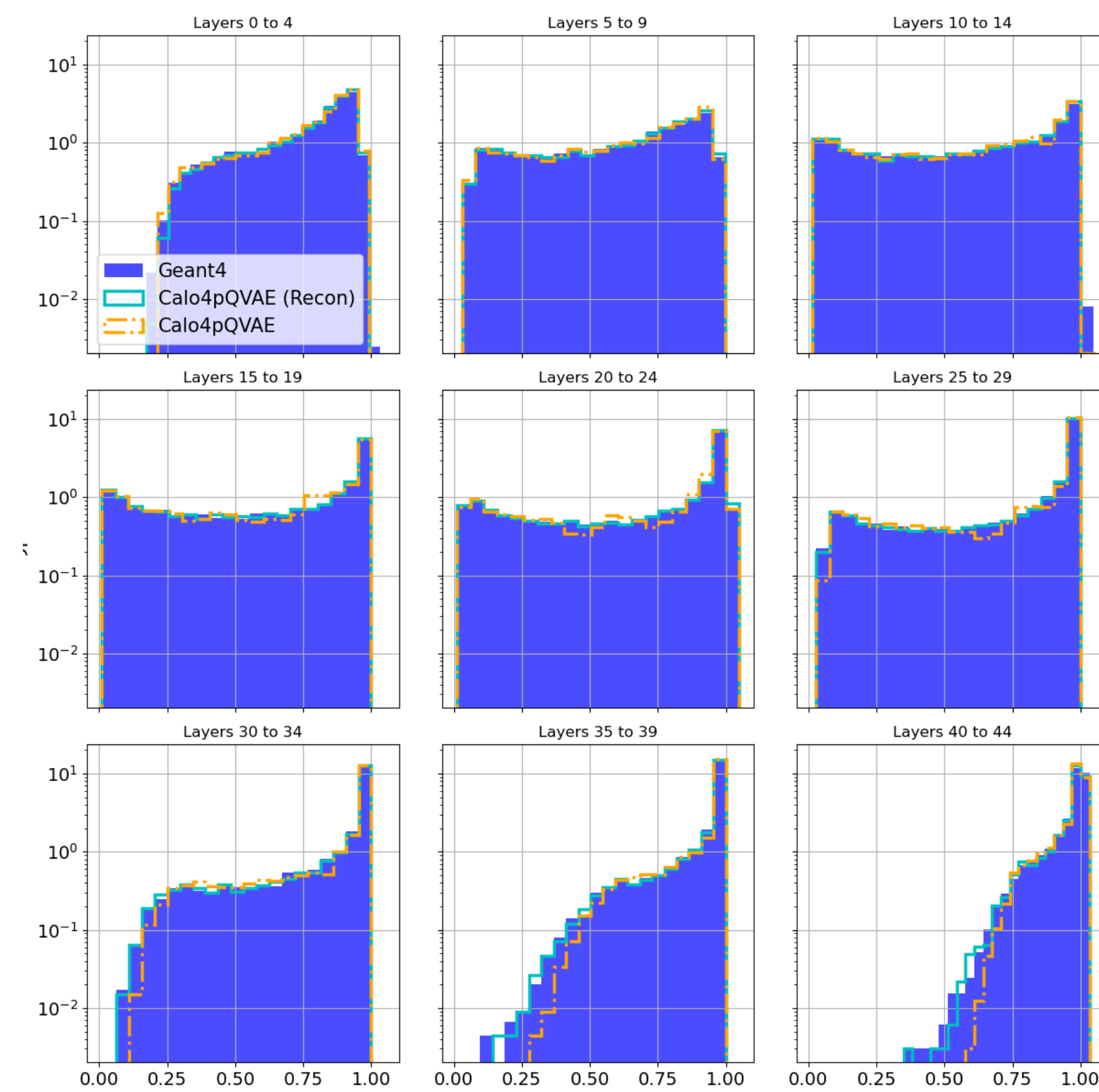
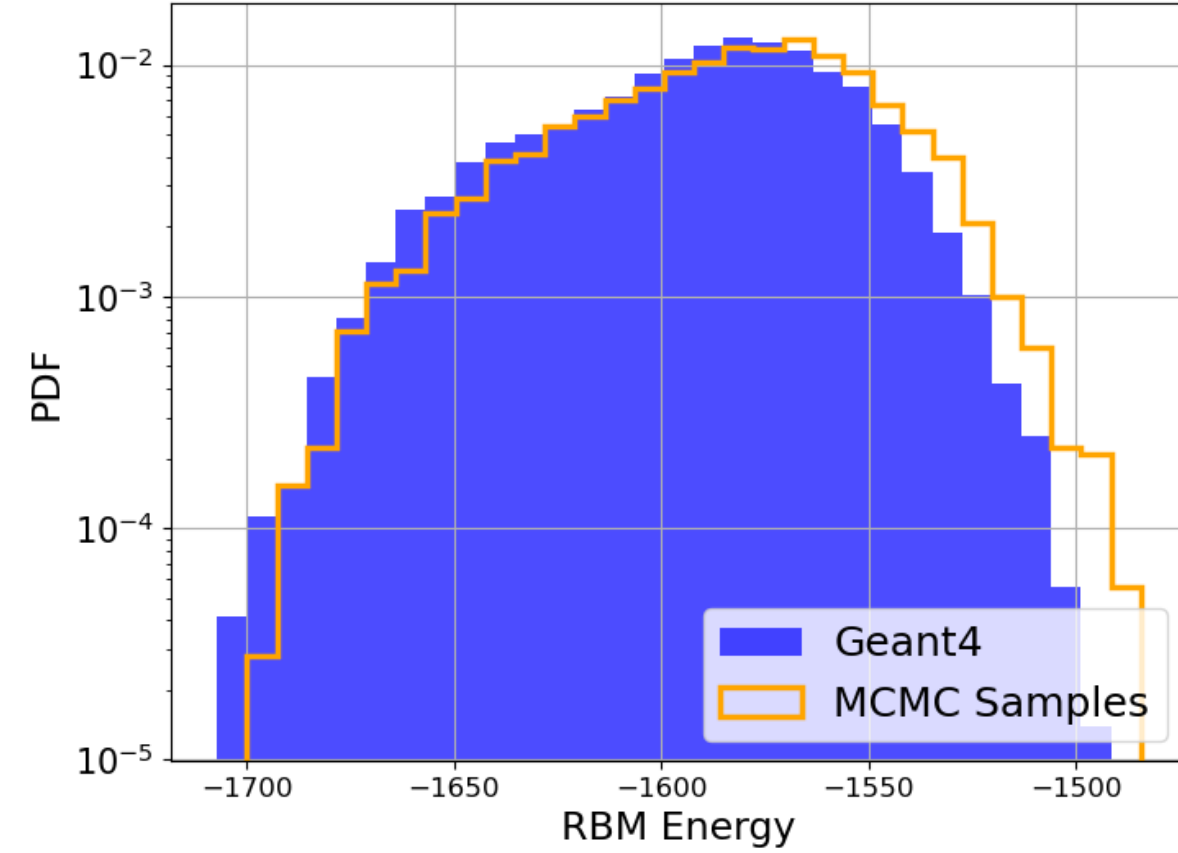
```
    out = wei @ v
```

```
    return rearrange(out, "b (l h w) c -> b c l h w", l=l, h=h, w=w)
```

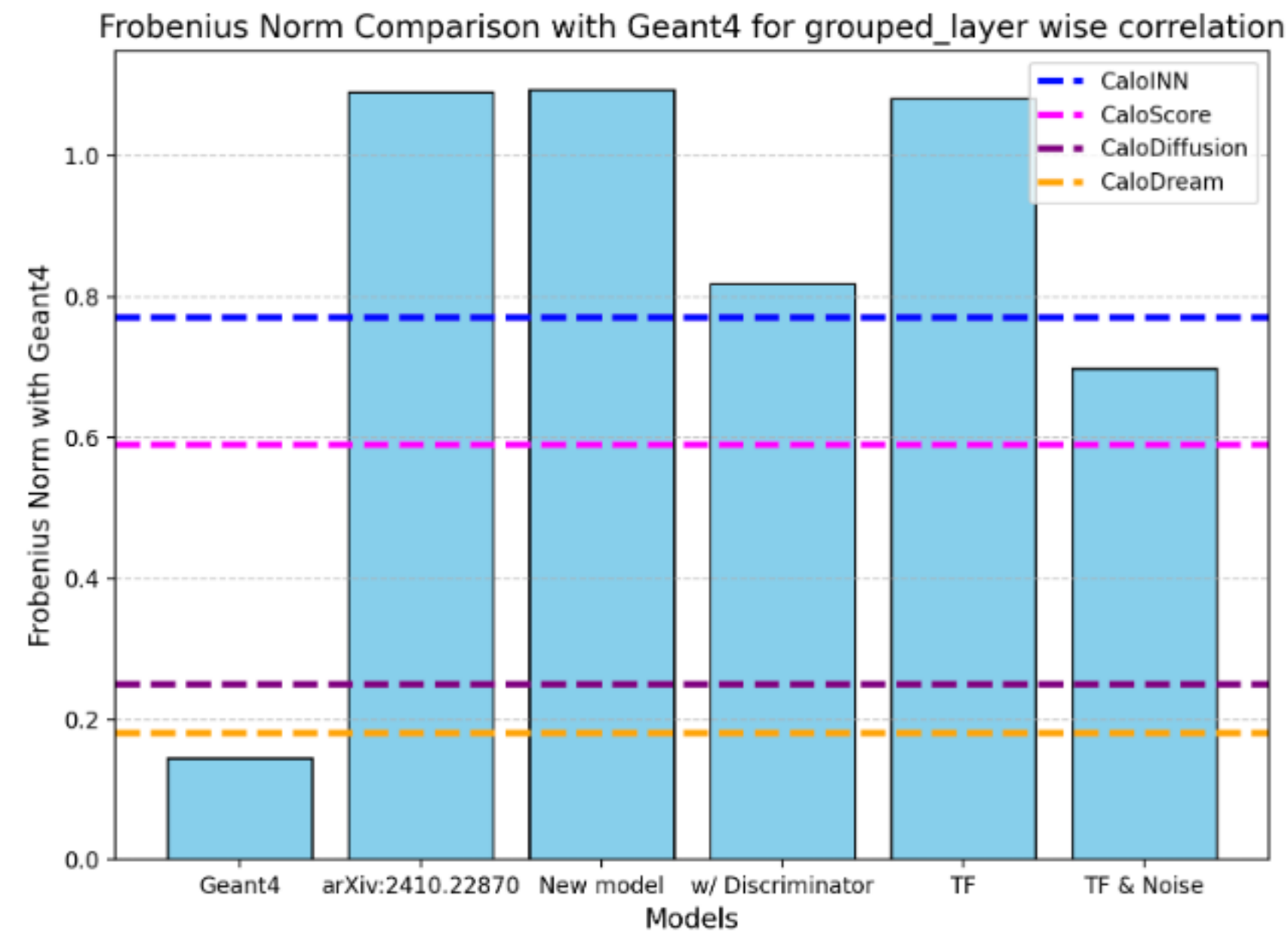
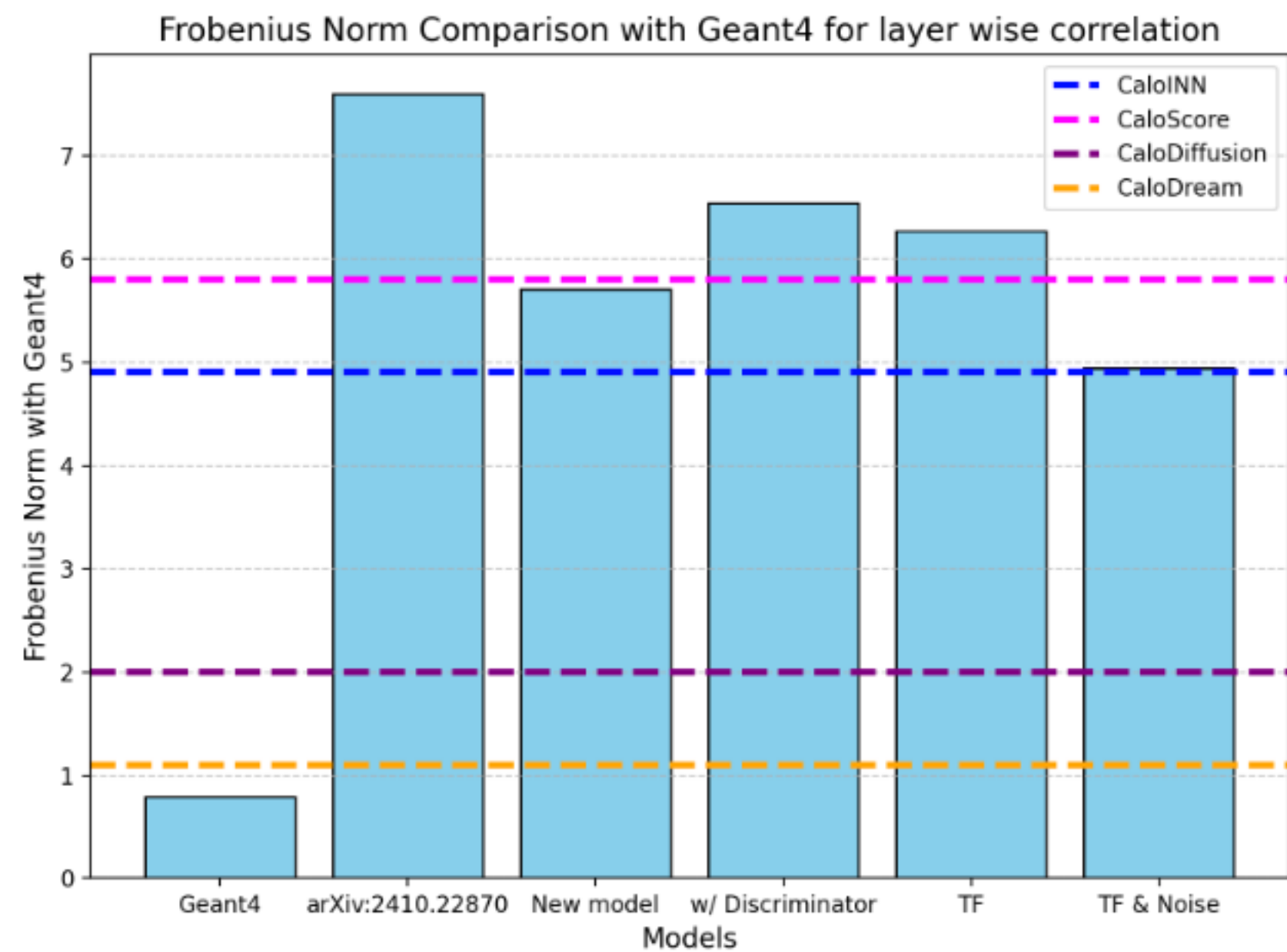
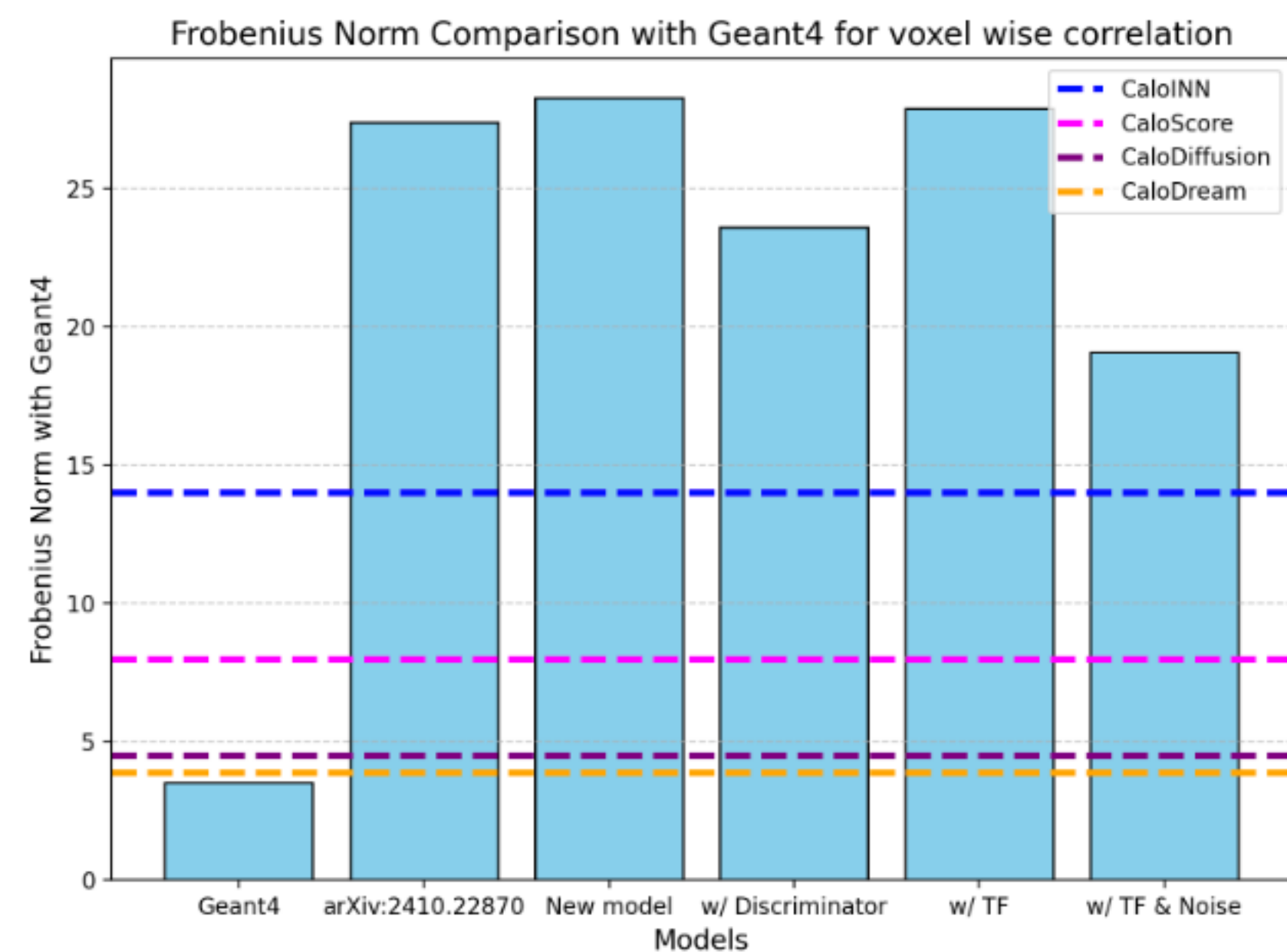
Permutes dimensions in X to $b \times (l \cdot h \cdot w) \times c$ (Rank 3)
Think of $X[i, j, :]$ as the features of voxel j in batch i
Builds the "weight matrix between tokens" Wei

tensor shape Batch $\times$ head size $\times l \times h \times w$





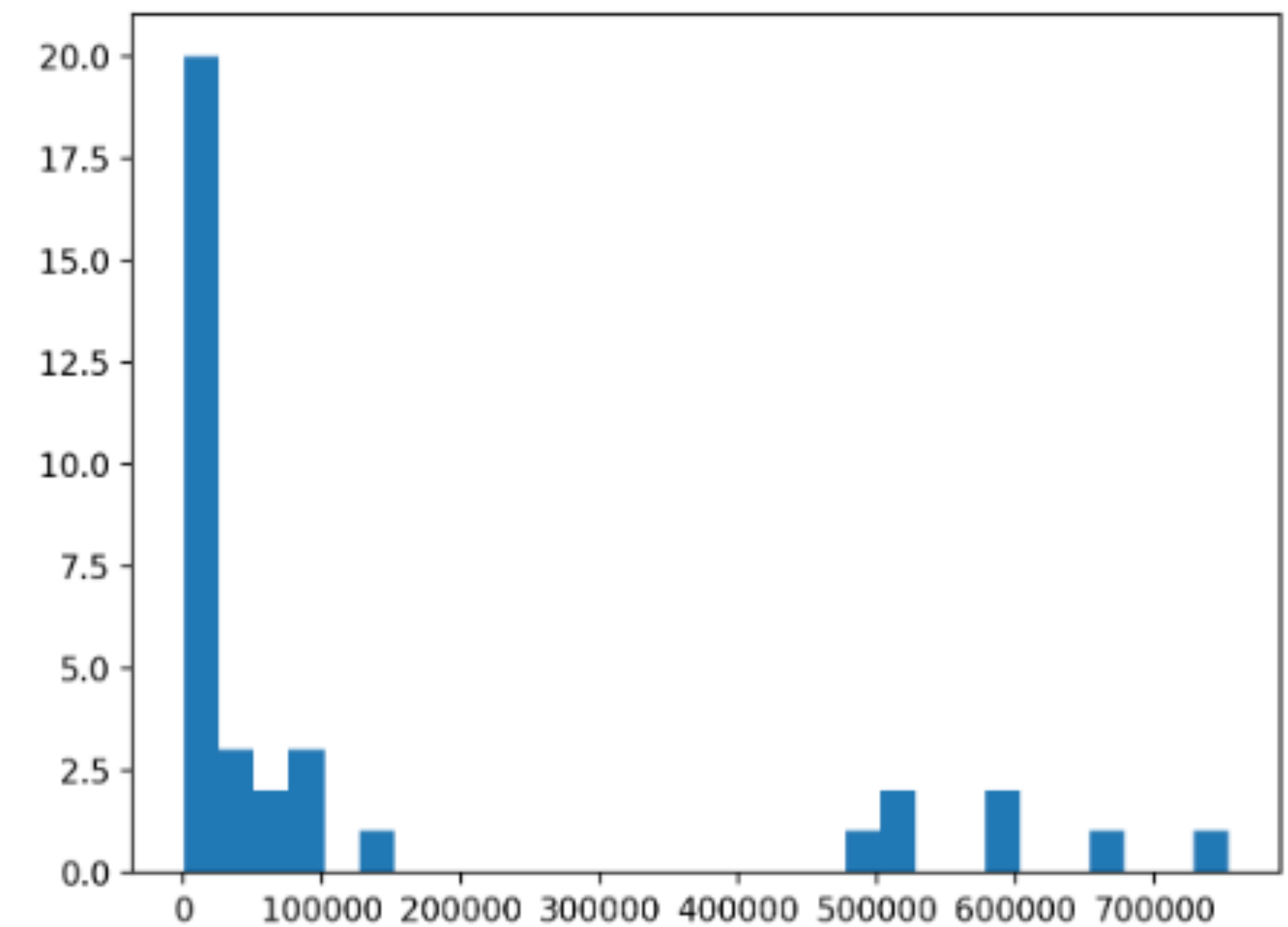
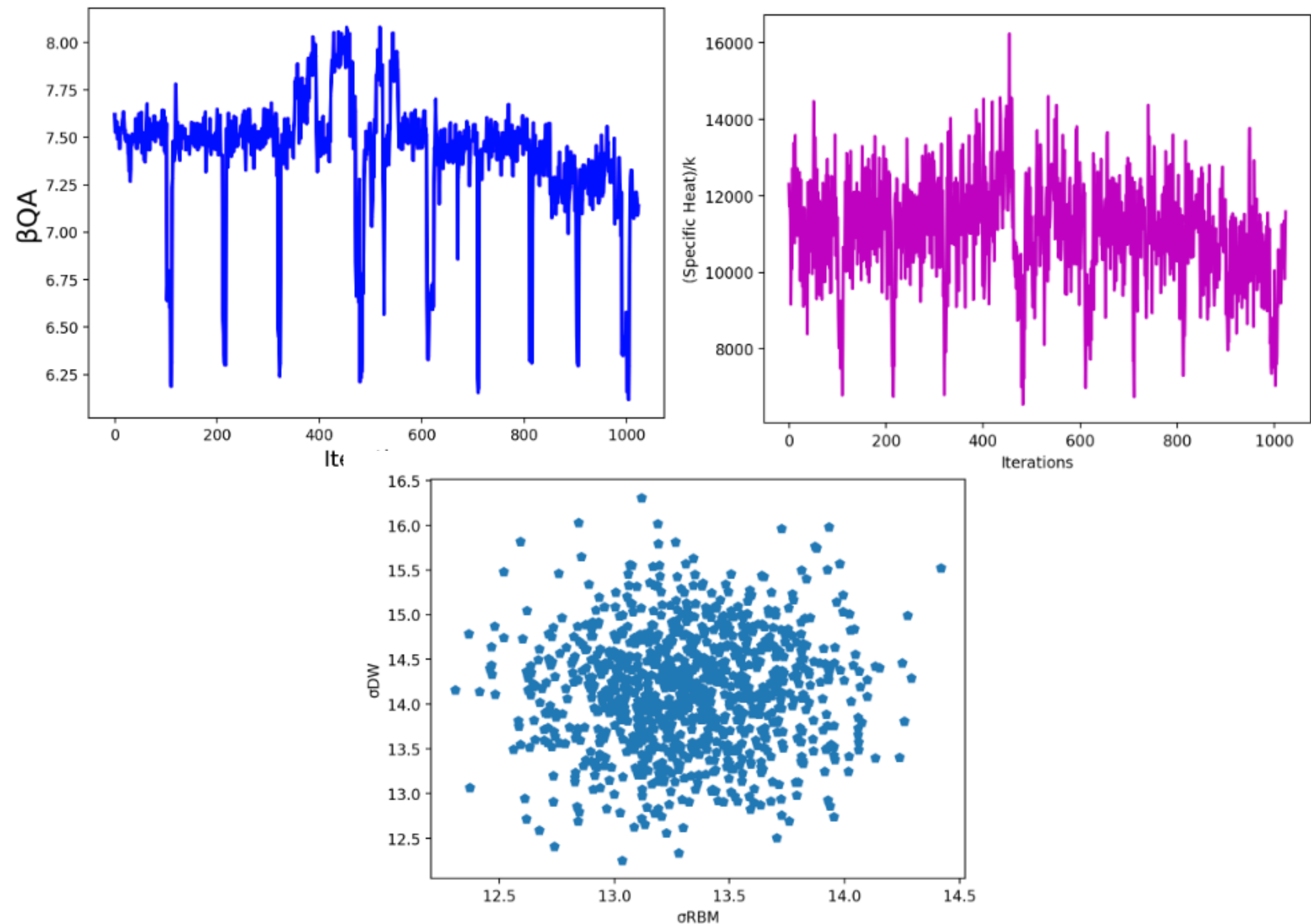
Model Comparison



KPD=0.65

FPD=334.72

Beta fluctuations



```
array([ 1264.0272,  1456.9666,  1499.0931,  1768.4874,  1801.2064,  
        1867.7544,  2467.3972,  2804.266 ,  3670.4797,  3892.2317,  
        4115.3667,  5956.2124,  8030.5303,  8574.8545, 10937.099 ,  
        11819.745 , 14311.695 , 14798.838 , 18881.361 , 25574.432 ,  
        35726.89 ,  41525.562 , 45929.805 , 55716.09 ,  64658.297 ,  
        78381.39 ,  83983.07 ,  99652.47 , 139697.72 , 483544.4 ,  
        522026.7 , 526698.8 , 587573.6 , 592855.4 , 655100.94 ,  
        753235.1  ], dtype=float32)
```