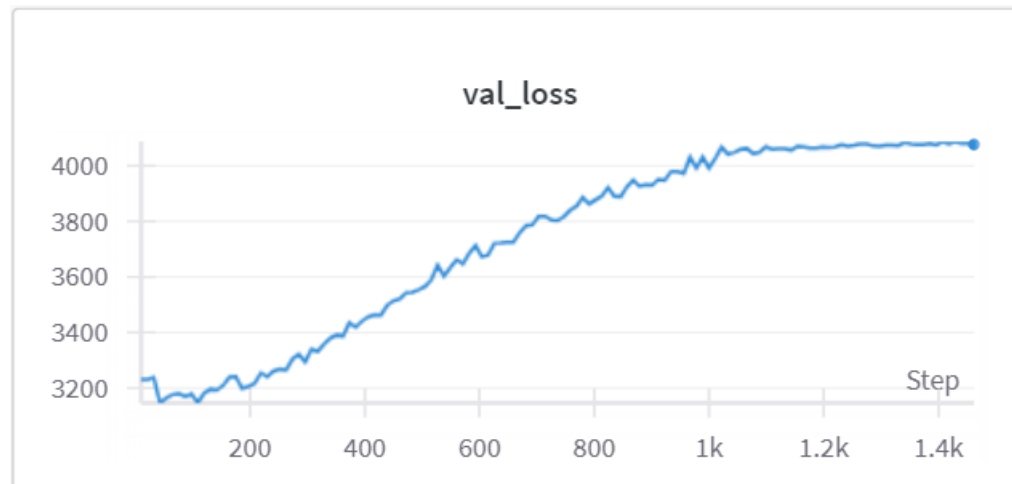


Weekly Update

17 Oct 2024

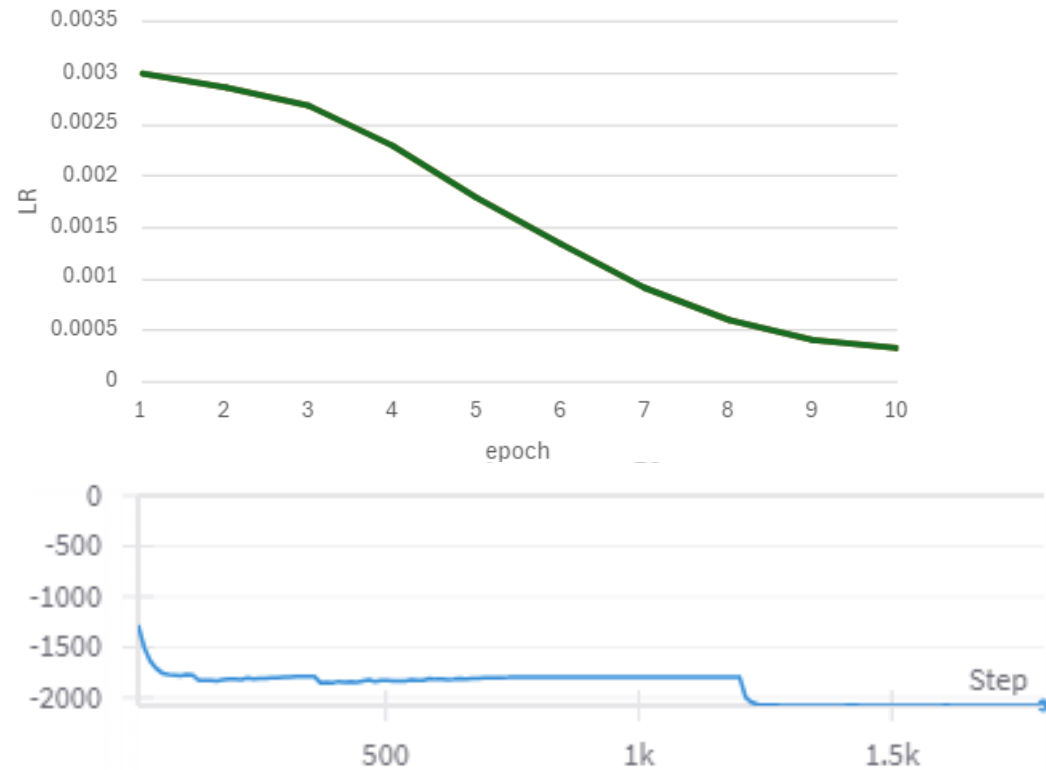
Quick Recap

- ↑ Decoders → ↑ **val_hit_loss slope**
- ↑ Dropout → ↓ **val_hit_loss slope**
- ↑ Weight Decay → ↓ **val_hit_loss slope**, ↑ **Volatility**
- ↑ LR → ↓ **val_hit_loss slope**, ↑ **Volatility**



Warm restarts

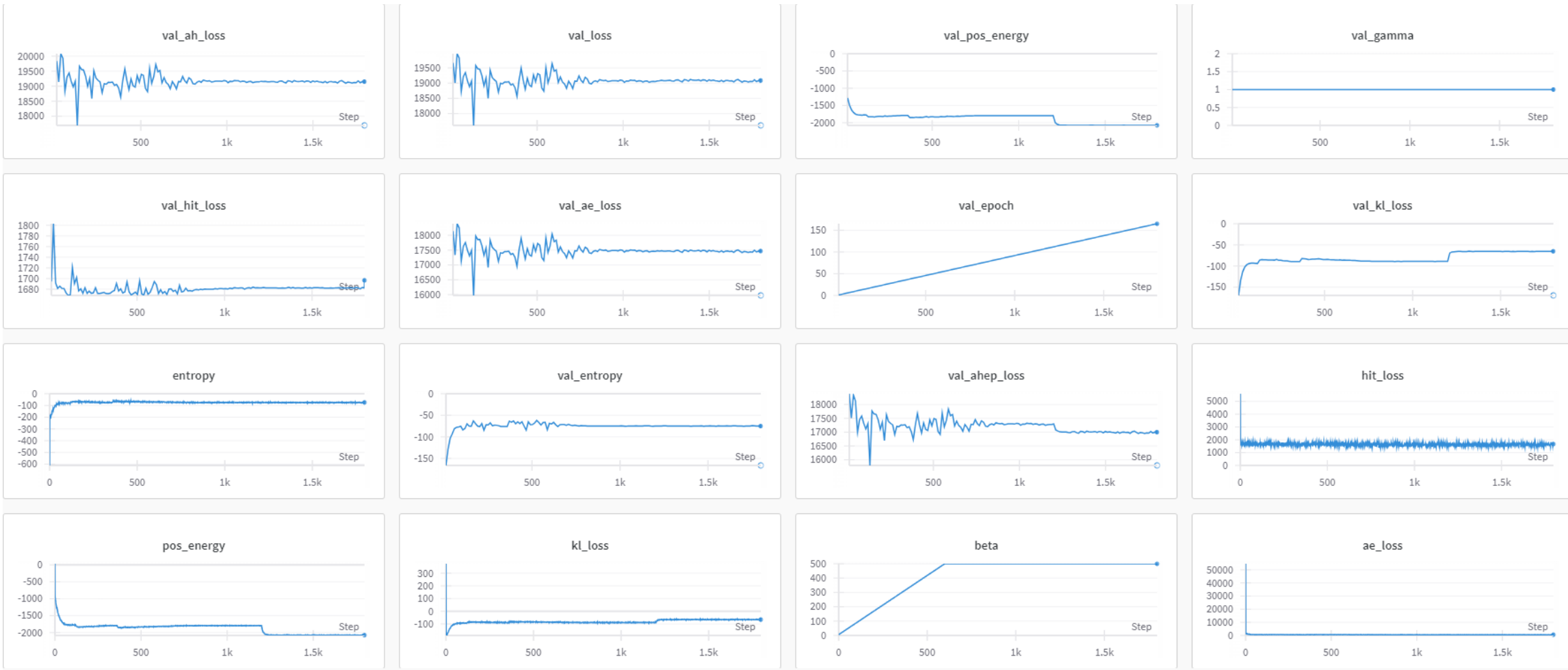
- $T_0 = 10$
- $T_{\text{mult}} = 2$
- 2 times



```
scheduler = torch.optim.lr_scheduler.CosineAnnealingWarmRestarts(  
    engine.optimiser,  
    T_0=config.engine.T_0, # The number of iterations for the first restart  
    T_mult=config.engine.T_mult # Factor for increasing the cycle length
```

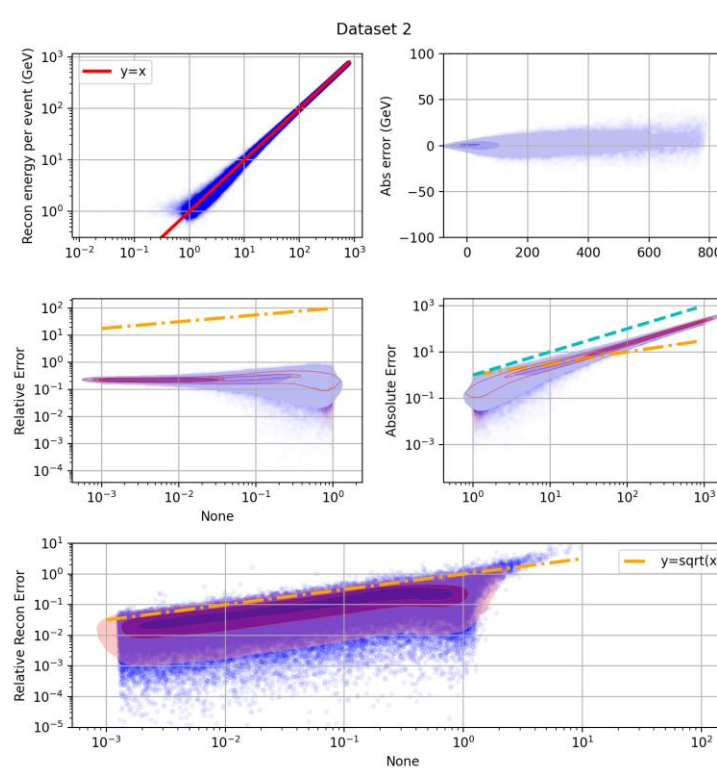
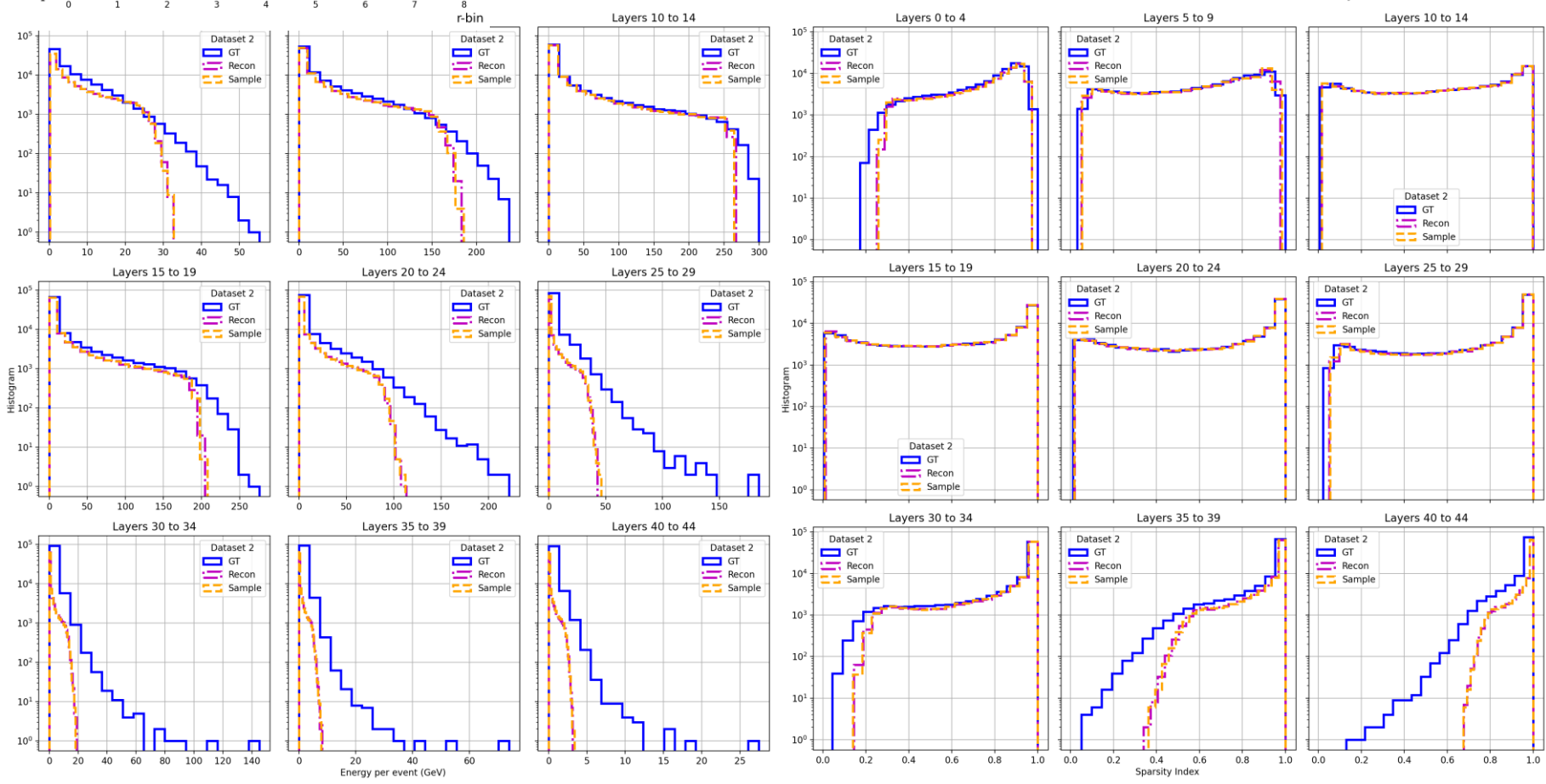
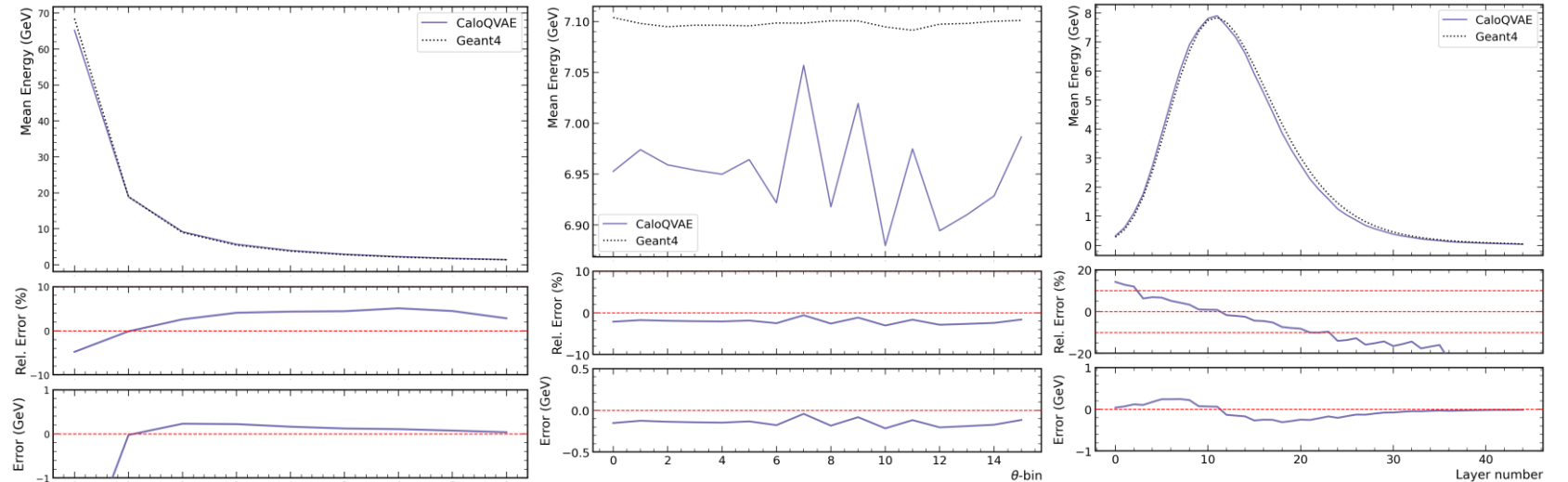
Model 389

- 15 decoders
- 0.5 Dropout
- 0.005 Weight decay
- 0.003 Learning rate (High, Warm Restarts)

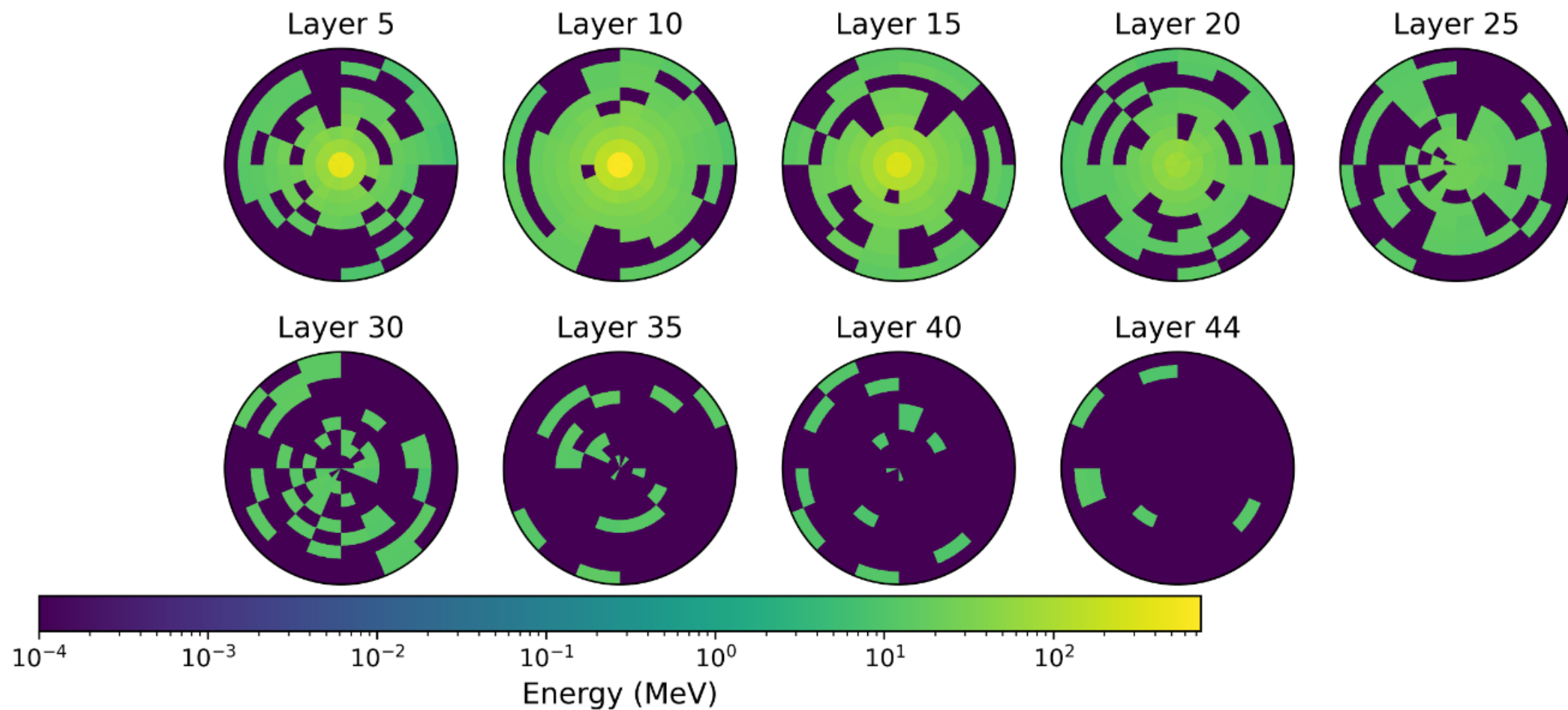


Model 389

- 15 decoders
- 0.5 Dropout
- 0.005 Weight decay
- 0.003 Learning rate (High, Warm Restarts)

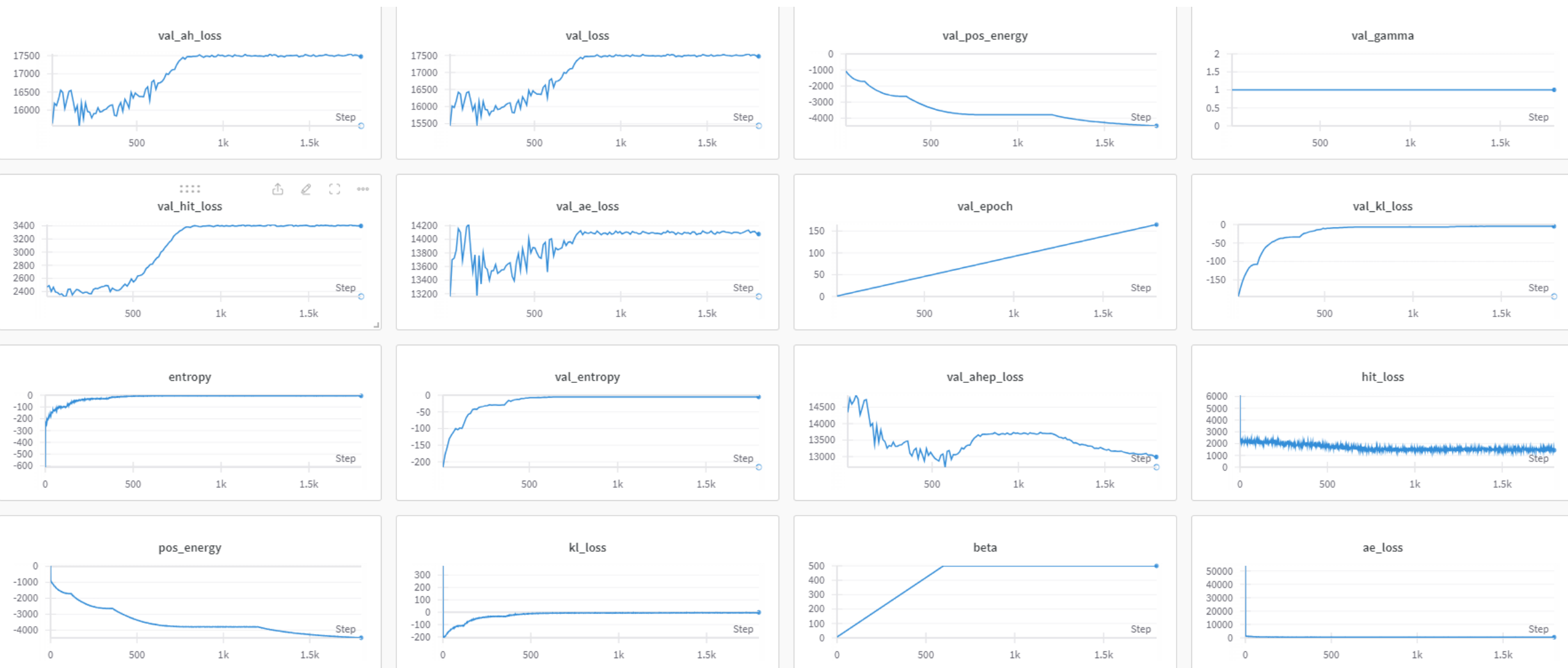


Model 389



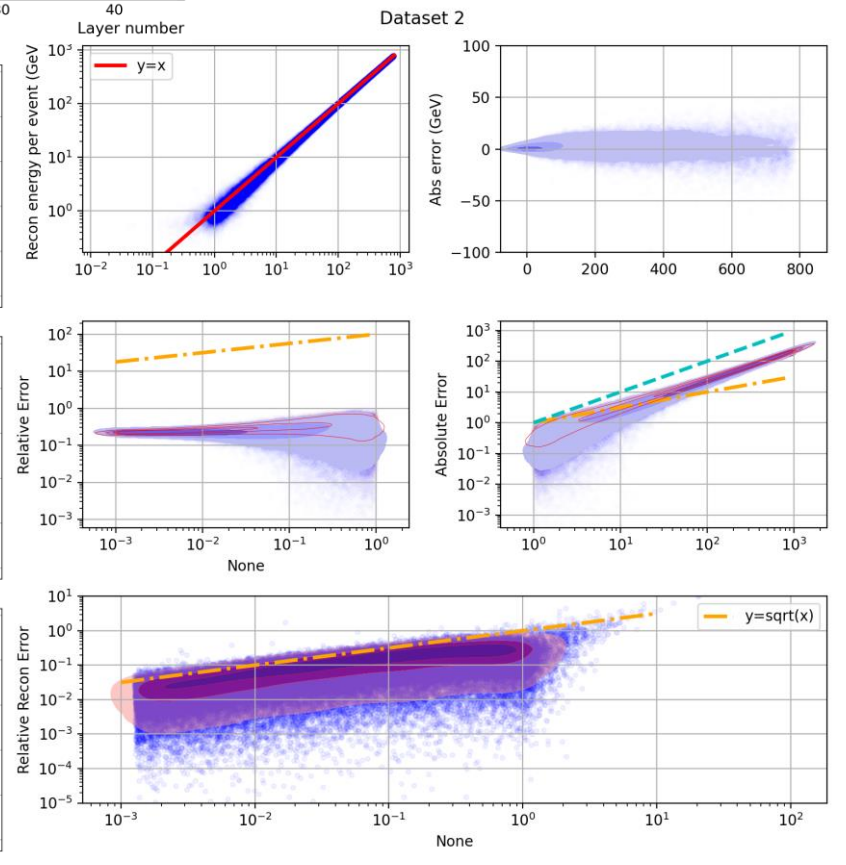
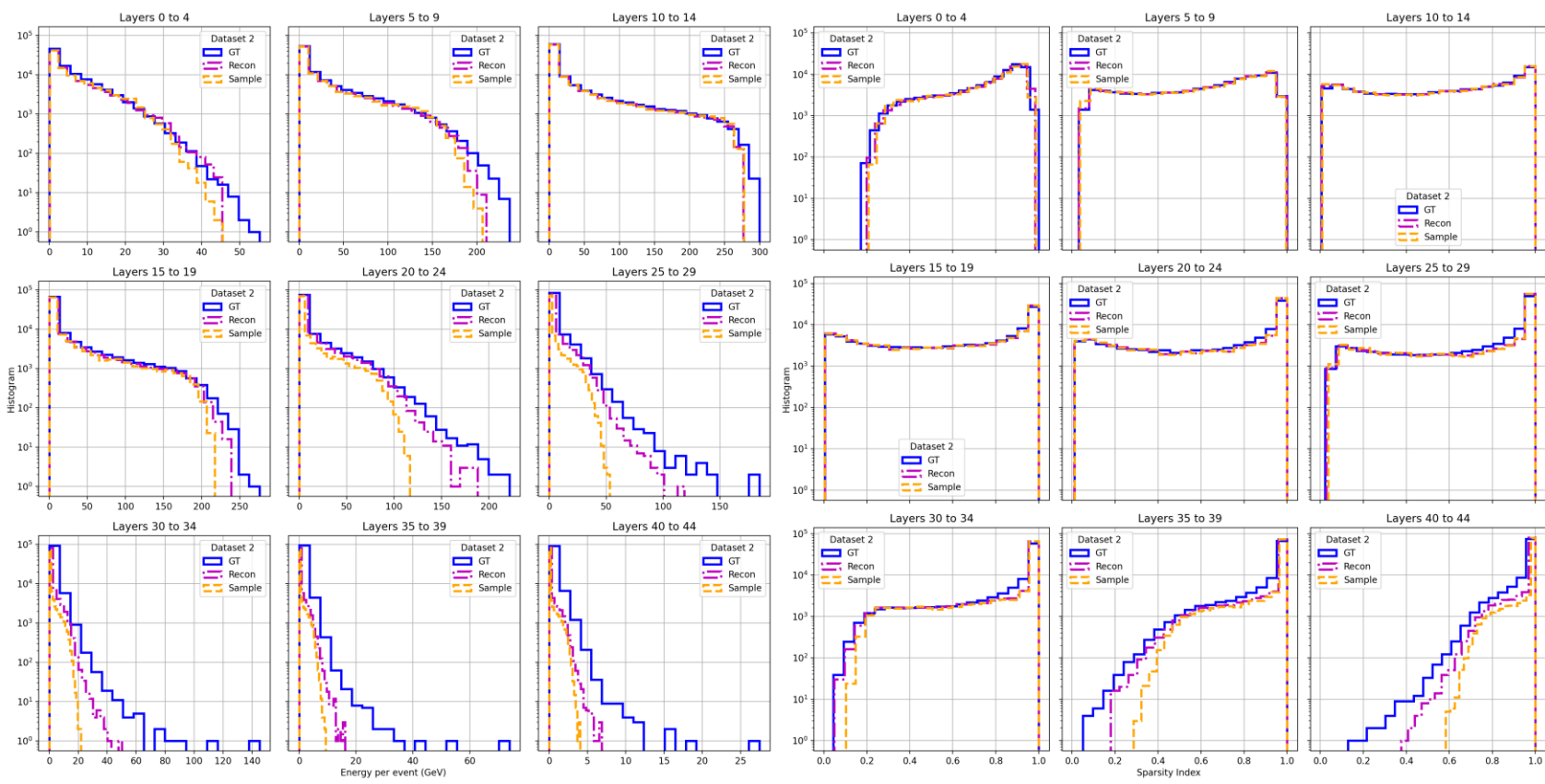
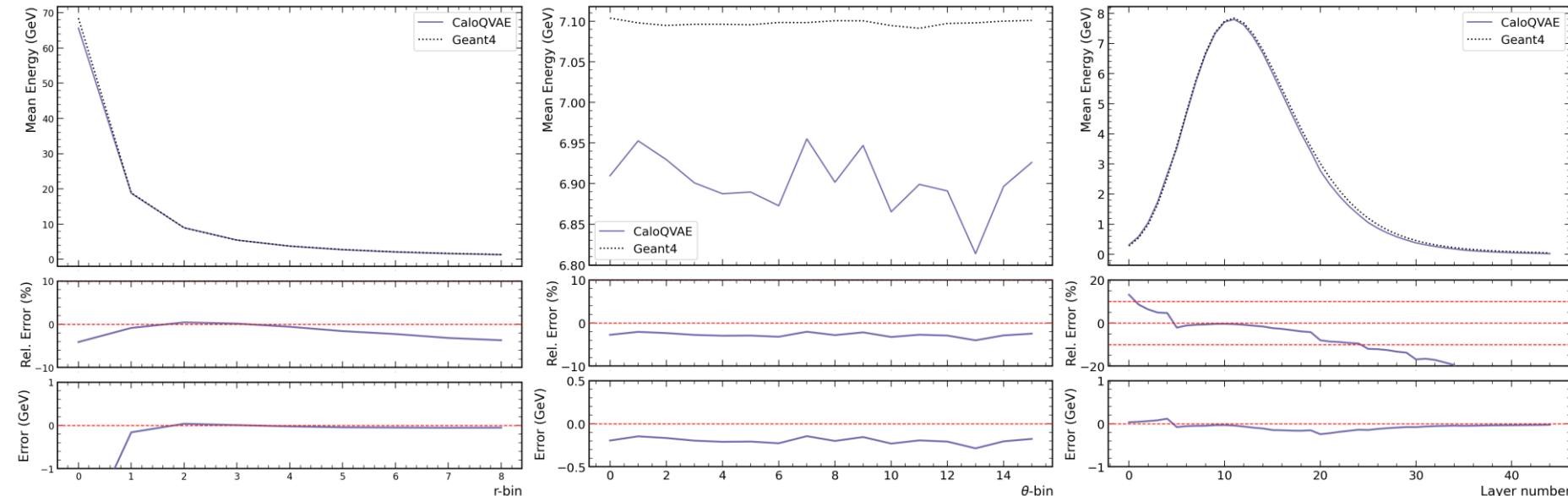
Model 398

- 9 decoders
- 0.4 Dropout
- No Weight decay
- 0.001 Learning rate (Warm Restarts)
- Removed hit mask from forward function

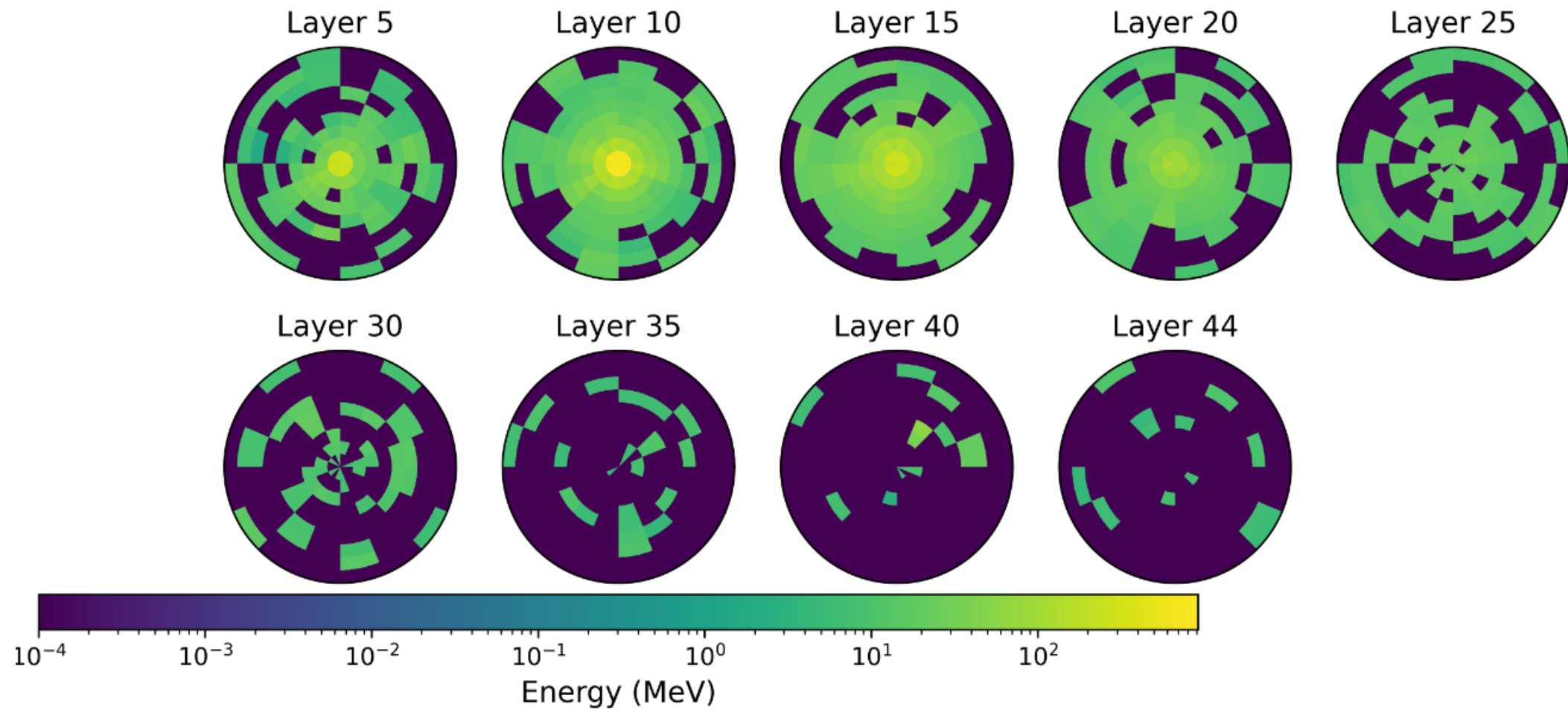


Model 398

- 9 decoders
- 0.4 Dropout
- No Weight decay
- 0.001 Learning rate (Warm Restarts)
- Removed hit mask from forward function



Model 398



Next Steps

- BCE Weight

```
hit_loss = binary_cross_entropy_with_logits(fwd_out.output_hits, torch.where(input_data > 0, 1., 0.), weight= (1+input_data).pow(self._config.model.bce_weights_power), reduction='none')
```

- Warm restart periods
- Ways to improve overfitting