中央研究院物理研究所
INSTITUTE OF PHYSICS, ACADEMIA SINICA

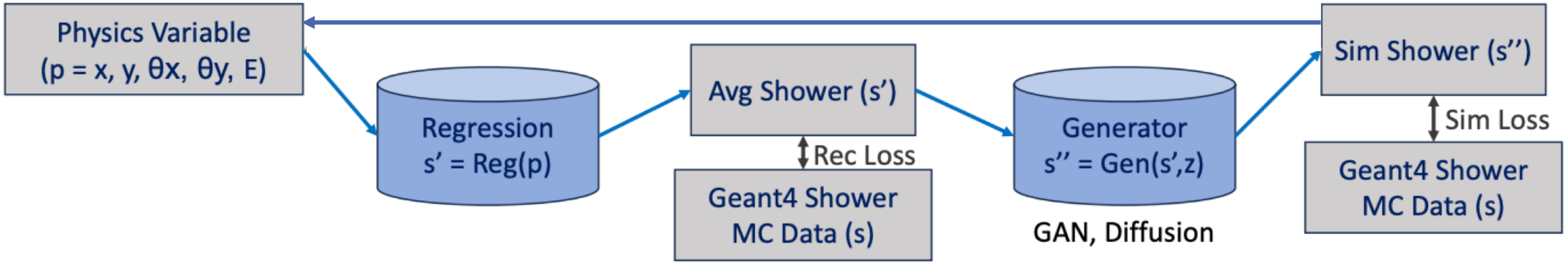
Status report

2025/09/01

ZDC Internal

WAI YUEN CHAN

Introduction

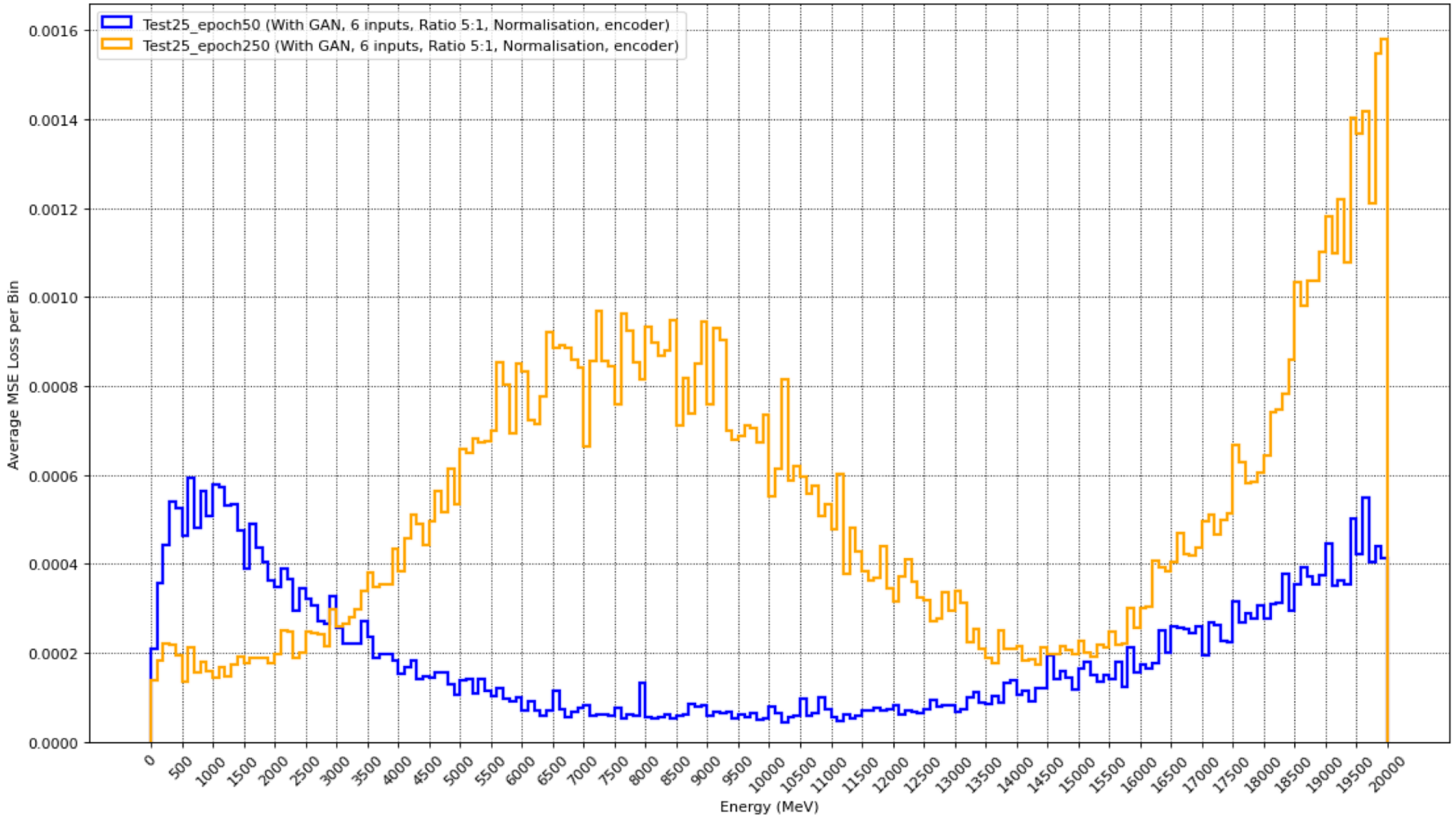


We have 2 parts in the algorithm:

1. Regression
   - We train to model to learn the general showering pattern in ECAL
2. Generator
   - We generate fake samples with noise to push the model to learn more details about the pattern (GAN)

Reminder: MSE loss with Normalisation

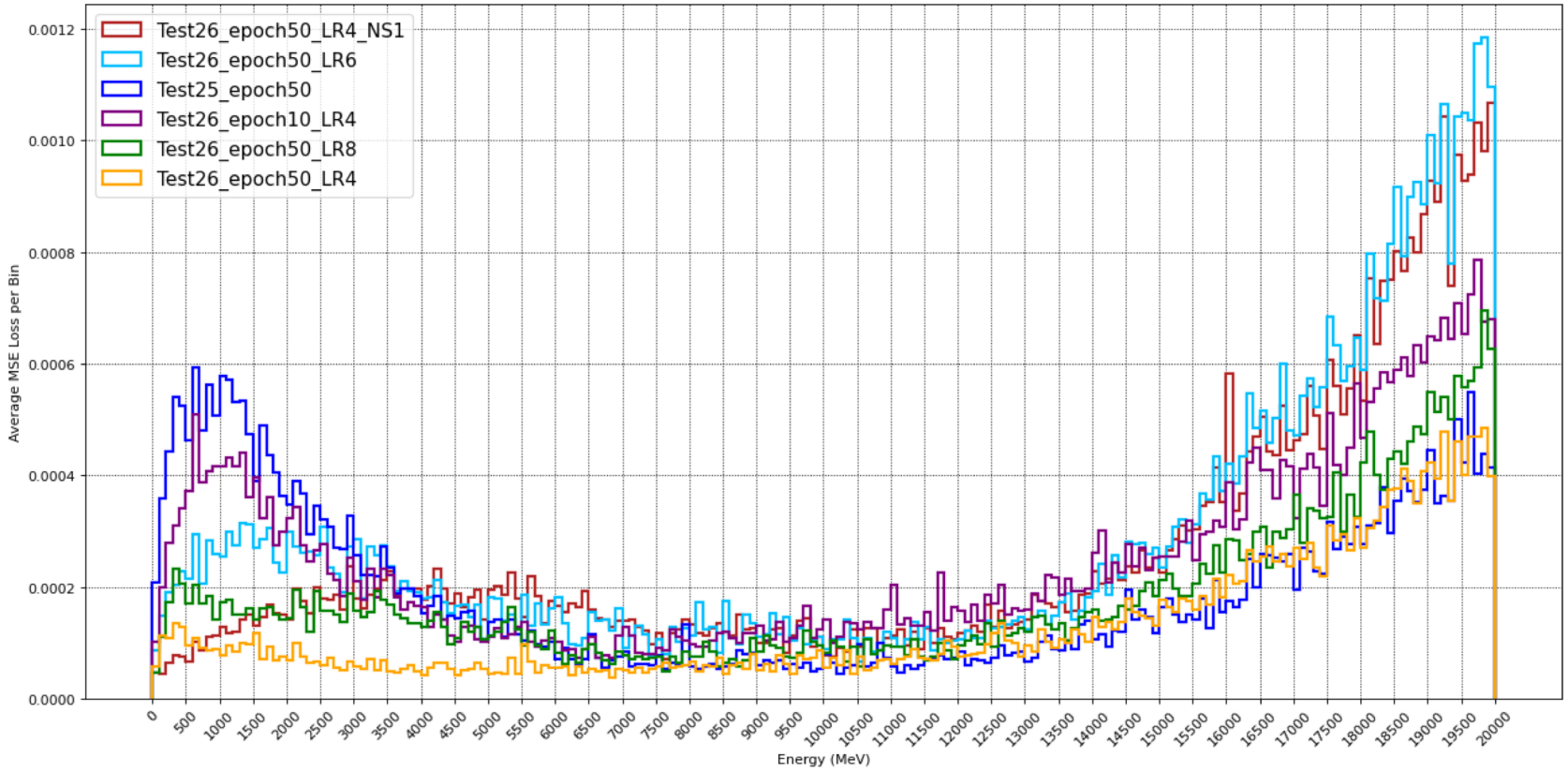
- Applying the normalization etc, we can see the improvement (Check the scale at Y-axis)



- However, with more iteration the MSE loss become worse, especially at high energy range.

Reminder: MSE loss with variable hyperparameters

- We try to add a hyperparameter to variable the noisy level of the Generated samples along the epochs during the training, multiplying the noise strength by 0.7 to 1.0.
- Using OneCycleLR to change the LR along the iteration (See next page)

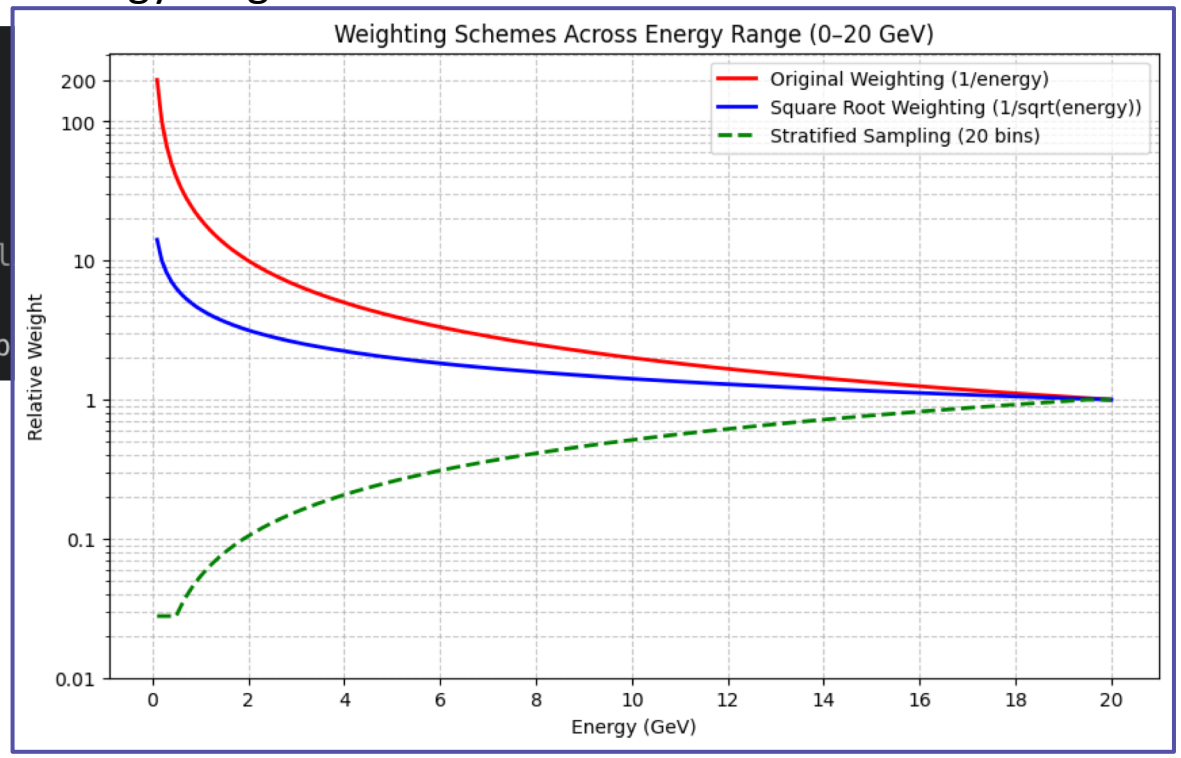


Sample Name	Hyperparameters
Test 25	Iterating DIS: GEN = 1:5
Test 26 LR4	Iterating DIS: GEN = 1:1 Max GEN LR = 9e-4 Max DIS LR = 1e-3 Pct_strat = 0.35
Test 26 LR4 NS1	Same as LR4 but noise strength always = 1.0
Test 26 LR6	Max GEN LR = 1e-3 Pct_strat = 0.8
Test 26 LR8	Max GEN LR = 7e-4

MSE loss at high energy range

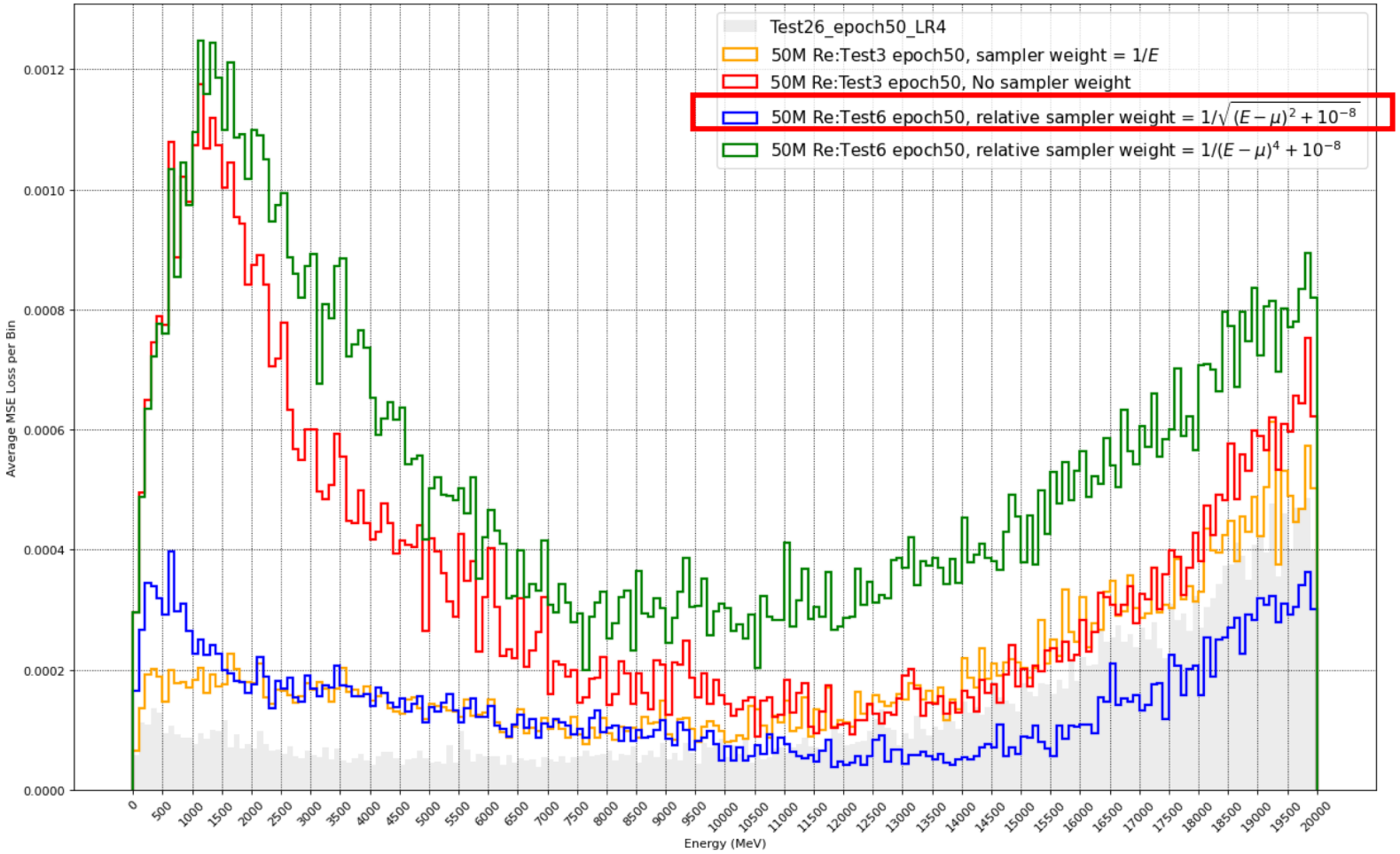
- Looks like the bias is coming from the weighting of the sampler in the dataloader.
- Since we shuffle our events during the training, a sampler ensure that events are allocated evenly.
- However, here our weights (= 1/E) suppressed the probability of having a high energy events during the training. Which means the lower energy events dominate.
- To improve, try to use $1/\sqrt{E}$ instead. Or design carefully for each energy range.

```
# Weighted sampling based on energy (mcPar[:, 0])
mcPar_np = np.array(dataset.mcPar).reshape(-1, 6)
energies = np.maximum(mcPar_np[:, 0], 1e-8)  # Clip to positive min
weights = 1.0 / energies  # Inverse energy weighting
weights = np.clip(weights, 0, np.percentile(weights, 99))  # Clip outl
weights = weights / weights.sum()  # Normalize (now all positive)
sampler = WeightedRandomSampler(weights, num_samples=len(weights), rep
```

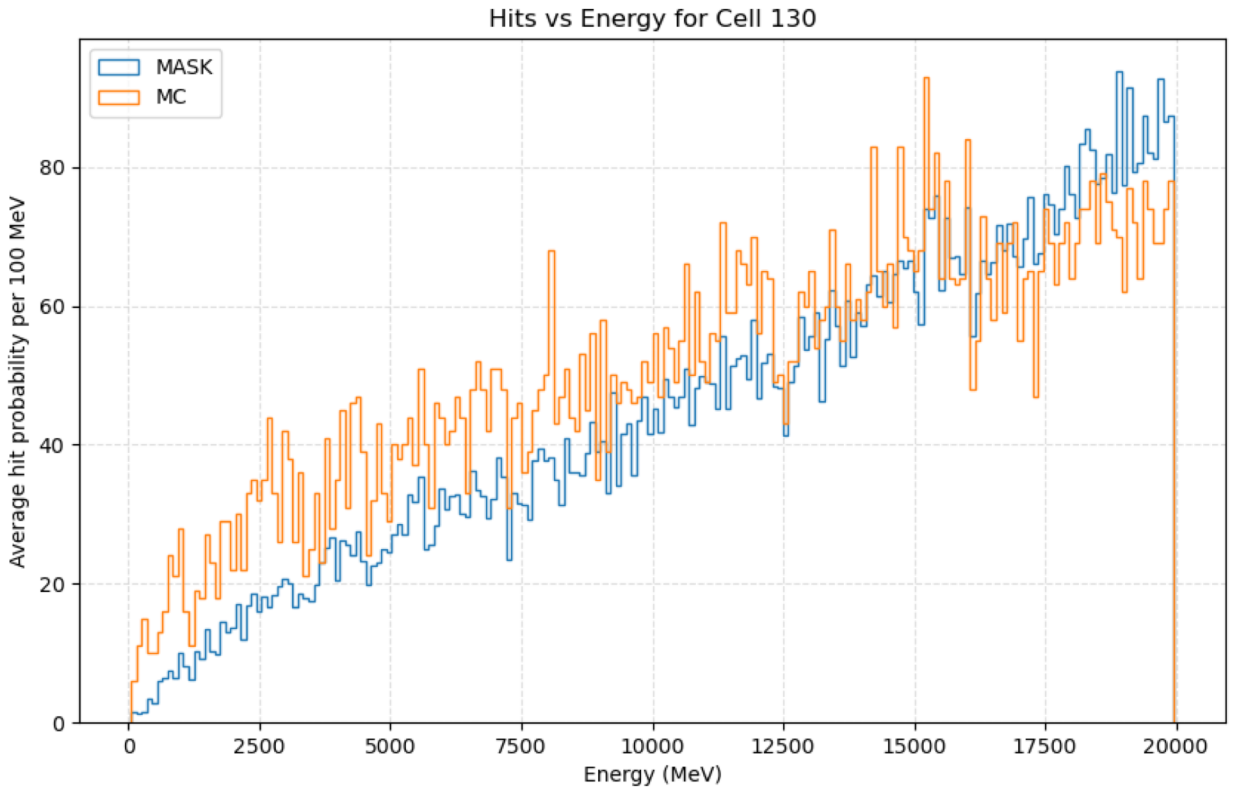


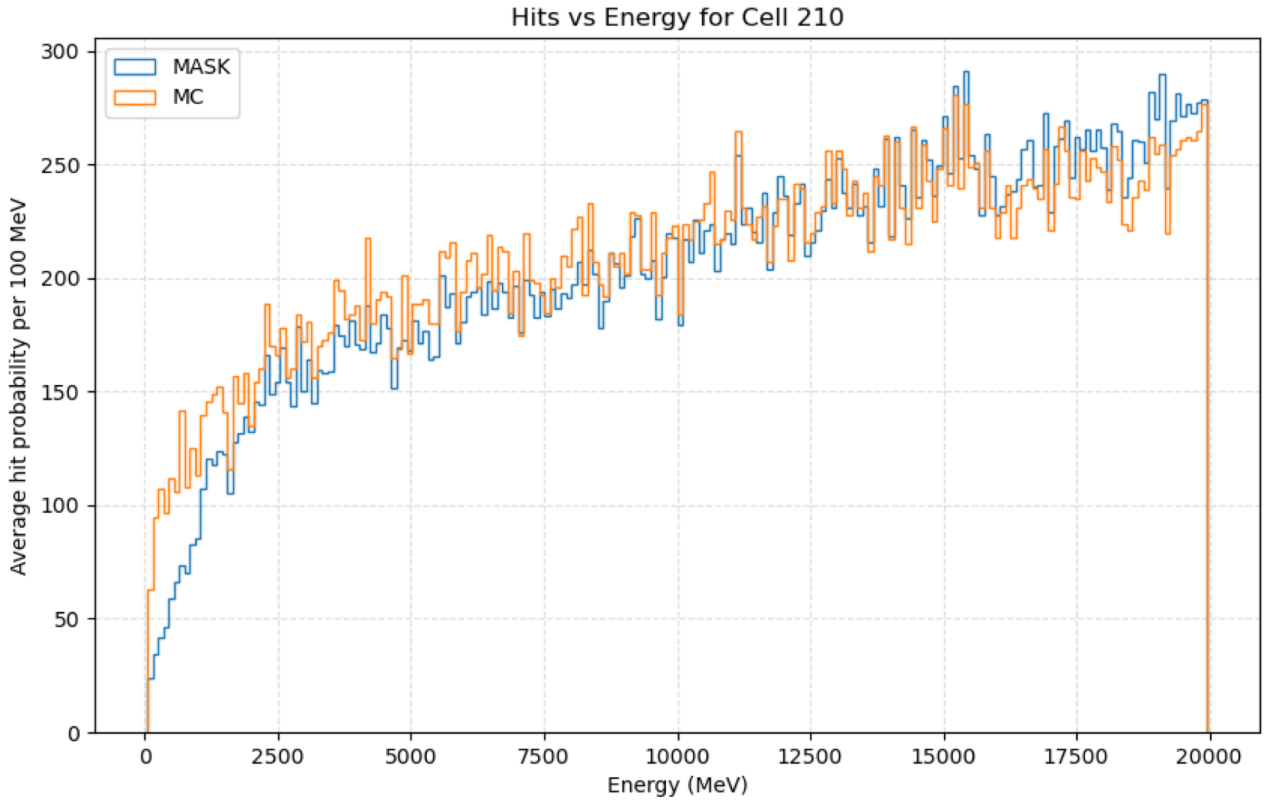
Remind

- Last time we tried to apply a sampler weight related to the energy range.
- Eventually the one with $^{1}/_{\sqrt{(E-\mu)^2}}$, where μ = mean energy, give the best performance.



Count vs energy per cell (Test6)

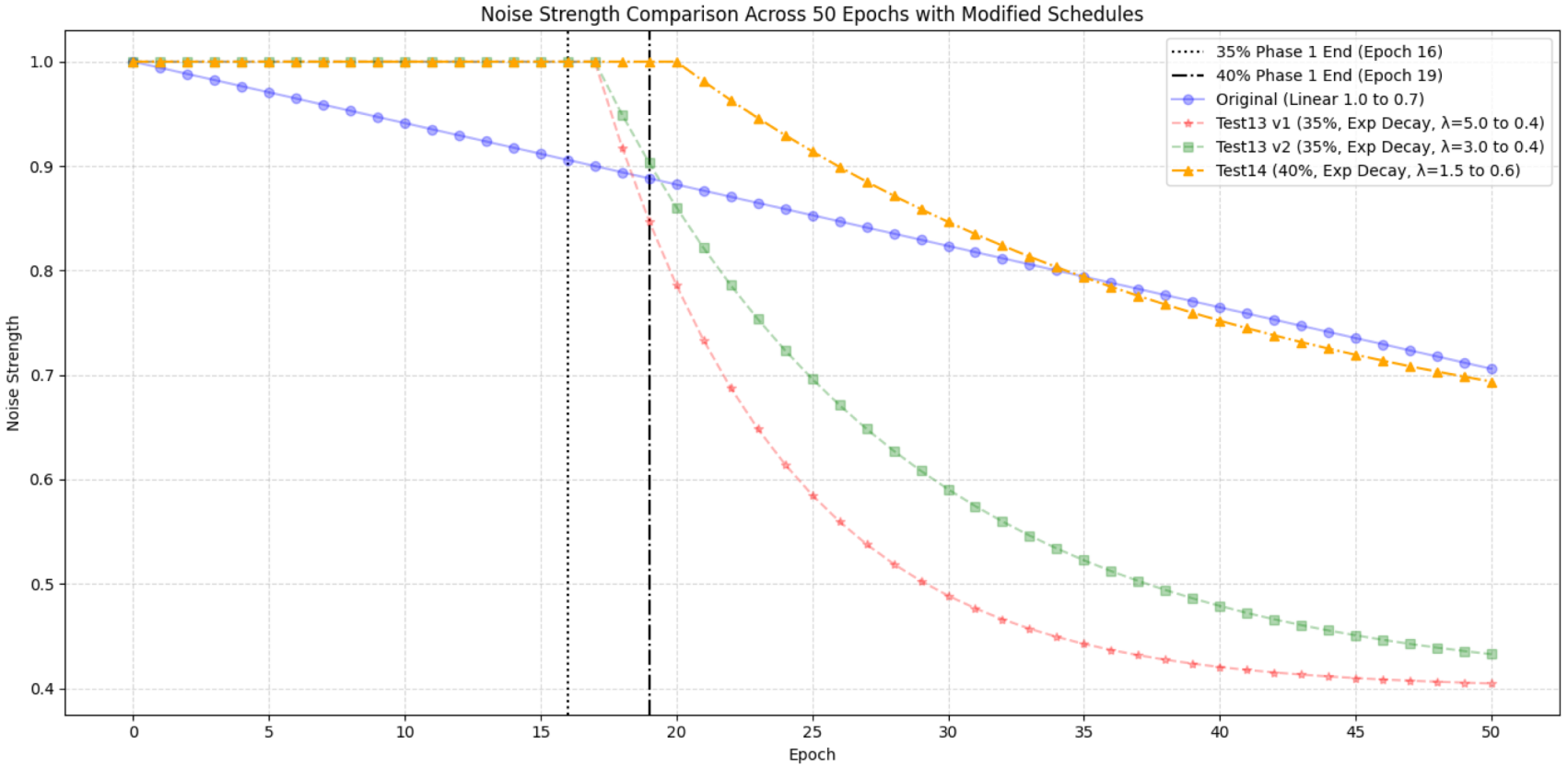




- Looks like the simulation at the center of the ZDC is overall mimicking the distribution from the MC sample.
- However at the outer cells, it gives a little bit different shape.

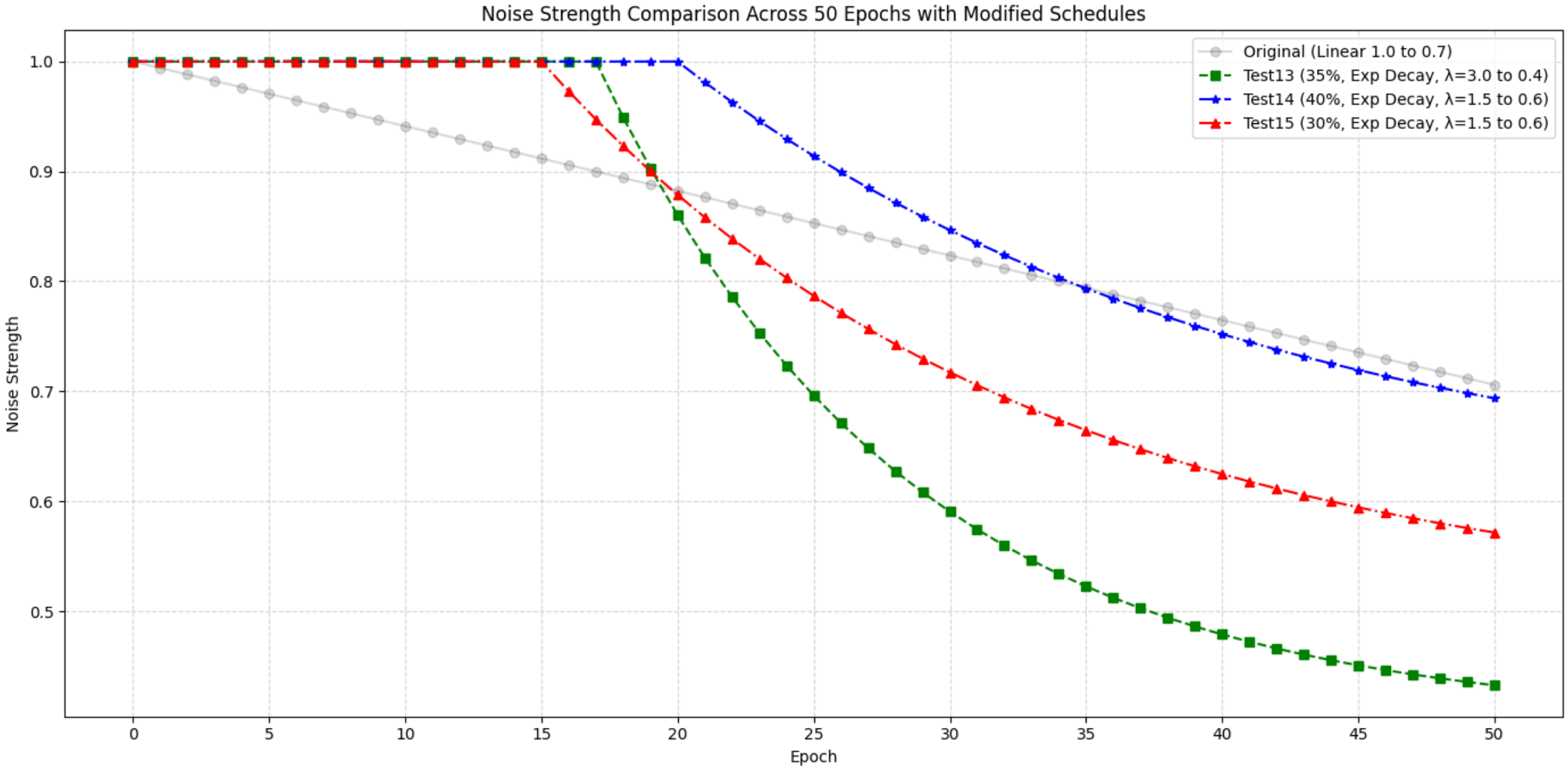
Attempts to improve the MSE loss

- We have try to add energy-depending penalty terms into the loss function but didn’t works.
- Therefore we try a different direction, try to change the noise strength that we have added into the generated image.

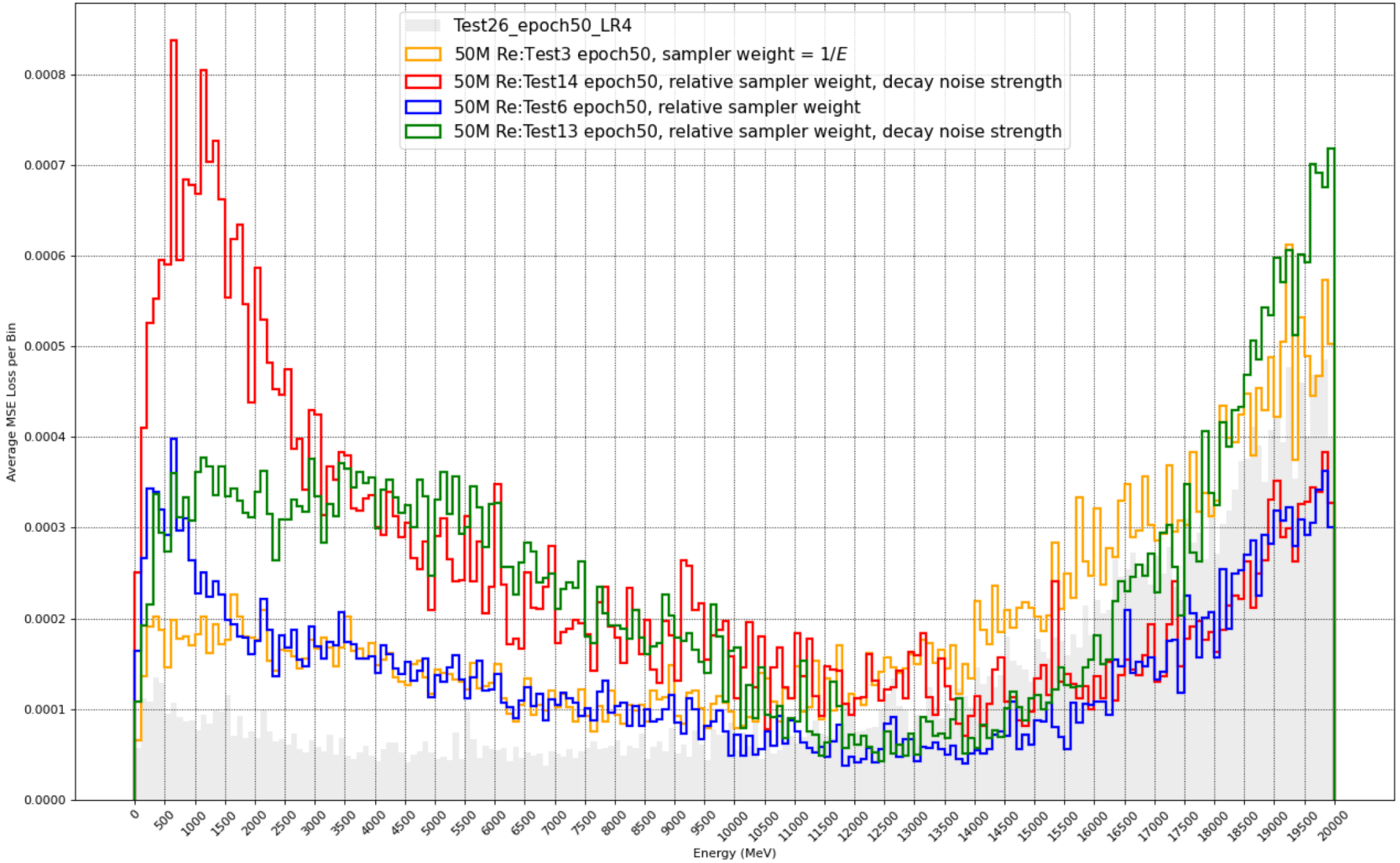


- In principle, the discriminator should be easier to distinguish the generated samples and the MC samples if the generated samples are much noisier.
- On the other hand, if we keep reducing the noise level, the generated sample would be much “sharper”.
- However, the linear schedule would be easy for the discriminator to predict how the generated sample change.
- Therefore we can try to use the decay schedule. Also we try to keep the noise strength = 1 in order to keep the randomness at the beginning.

Decay noise strength



Decay noise strength: Result

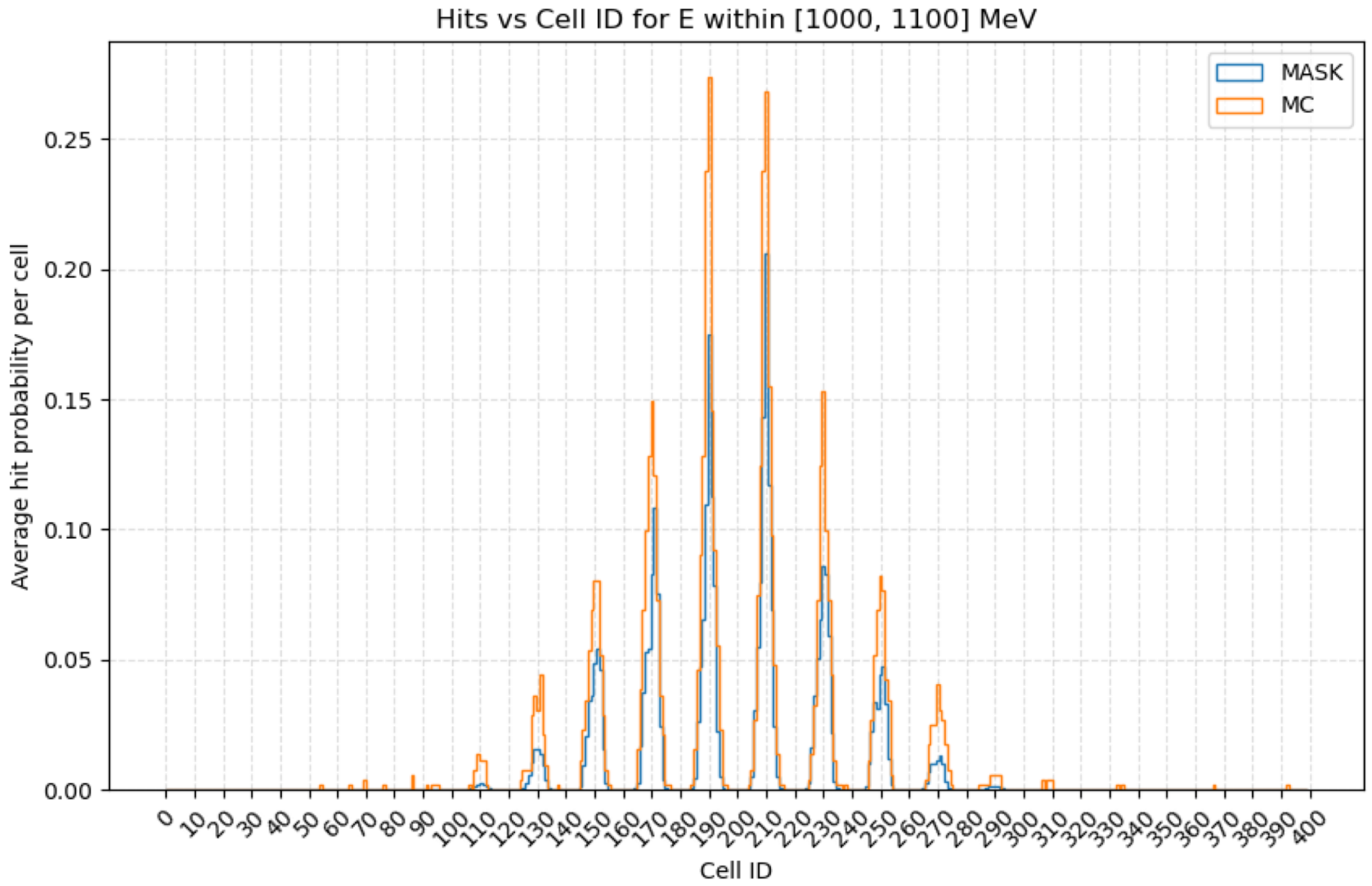


- Didn't show Test 15 here but also failed.
- At the moment nothing can beat the best performance (Test6).
- We can also try something else in the noise strength schedule (e.g. smoother functions)

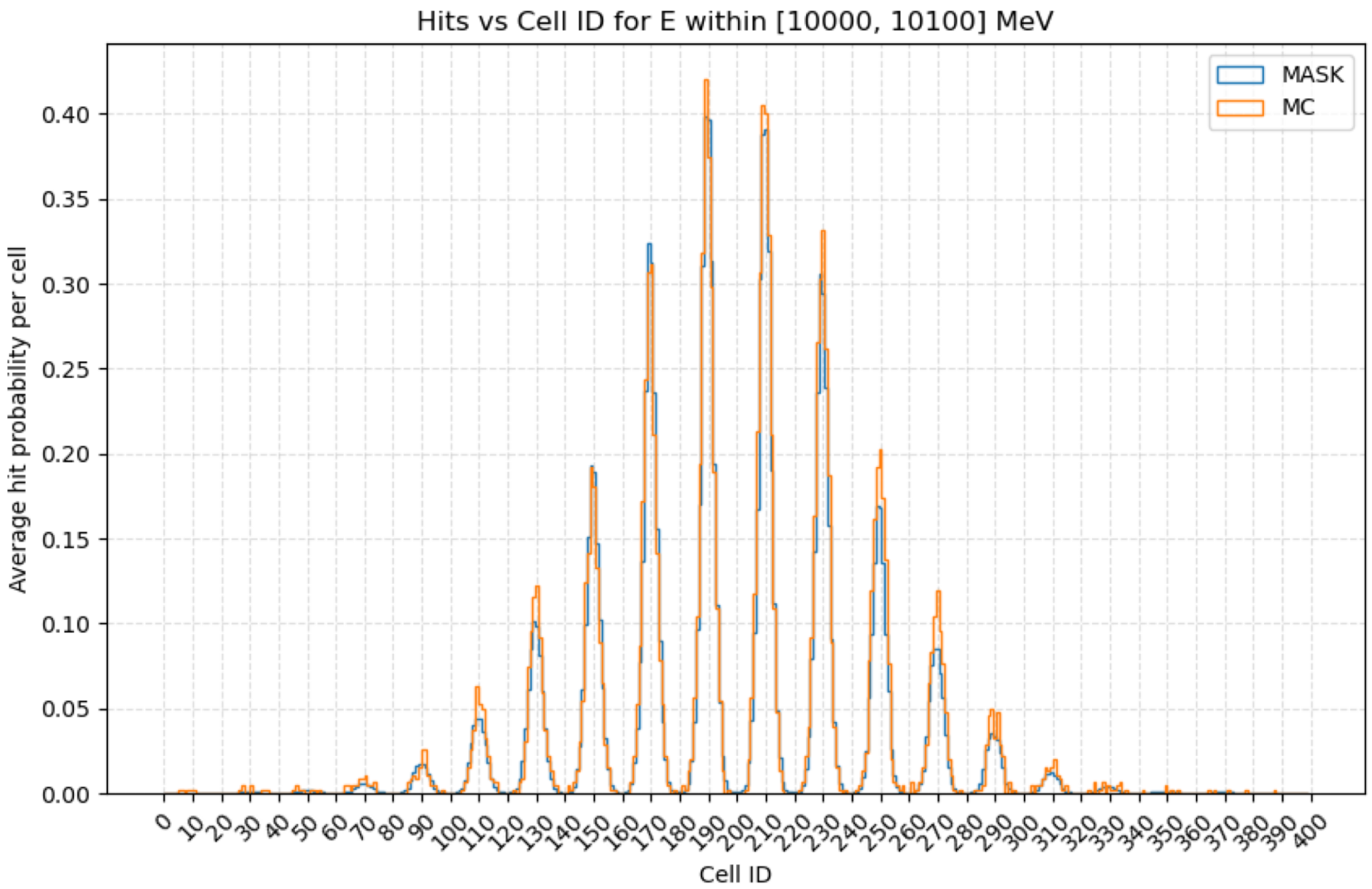
Conclusion and Outlook

- Looks like we are trying to improve the MSE loss which is already good enough (~ 0.0003 with Test6)
- Rethink about the target and we shall move on and try to add HCAL structure with neutron.
  - We can use [E, phi, theta] as the input, and use 1 ECAL layer [20 x 20] + 64 HCAL layers [40 x 40 (?)].
  - Already generated 5M train neutron sample and 1M test neutron sample.
  - Still have to redefine the HCAL structure to become a grid.

Count vs energy per cell (Test6): Low E



Count vs energy per cell (Test6): mid E



Count vs energy per cell (Test6): high E