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

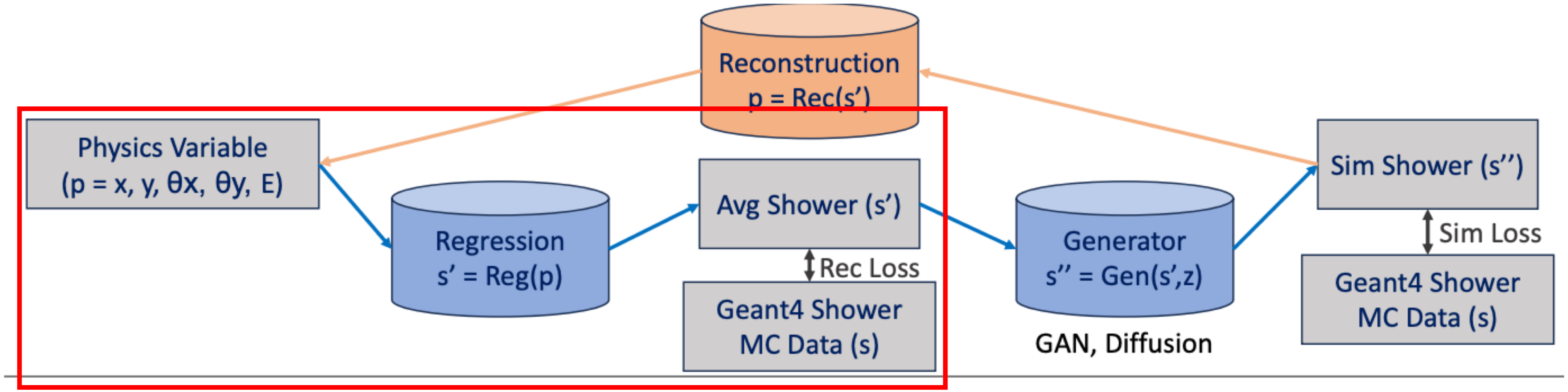
Status report

2025/06/27

ZDC Internal

WAI YUEN CHAN

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



Pre-processing (Data) → Input to → Model → Train by → Objective Function (Loss)

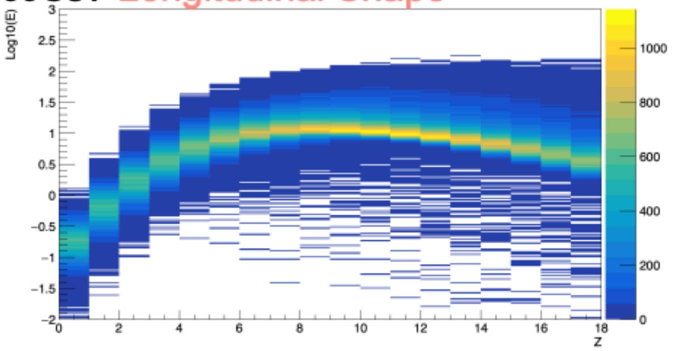
Model → Test by → Evaluation Methodology (Metric)

- Balanced samples (uniform)
- Data normalization ($\mu = 0, \sigma = 1$)

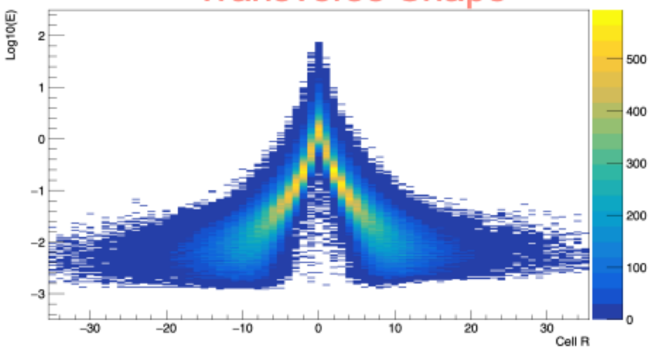
Evaluation Methodology (Metric): cell energy distribution

- GAN: Cross entropy loss
- REC: L2 loss

e-, 100GeV Longitudinal Shape



Transverse Shape



The longitudinal shower profile is described by the Gamma distribution:

$$\frac{dE}{dt}(t) = E_0 \frac{(\beta t)^{\beta T_0} \beta e^{-\beta t}}{\Gamma(\beta T_0 + 1)}, \quad (1)$$

using parameters described above. The scale parameter β is found to be constant over the wide energy range, therefore it is fixed in the present analysis. Individual shower parameters E_0 and T_0 are obtained from a fit to observed energy depositions in the ECAL cells of that shower. Fig. 7 shows the high quality of the description of electron showers over a wide energy range by the model based on Eq. (1).

At a given shower depth, the transverse shower shape as a function of the distance from the shower axis r is described by the sum of a narrow core and a wide tail:

$$\frac{dE}{dr}(t,r) \propto Q_C \frac{2rR_C^2}{(r^2 + R_C^2)^2} + (1 - Q_C) \frac{2rR_T^2}{(r^2 + R_T^2)^2}, \quad (2)$$

Reminder: Network structure

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

Hidden dim = 32 (i.e. 32 neurons in each hidden layers)
Add layers into the default structure: "Extended 2"

Default

```
self.encoder = nn.Sequential(
        nn.Linear(1, hidden_dim*4),
        LinearBlock(hidden_dim*4, hidden_dim*4, 4),
        LinearBlock(hidden_dim*4, hidden_dim*9, 4),
        nn.Unflatten(1, (hidden_dim, 3, 3)), # (3, 3)
        nn.ConvTranspose2d(hidden_dim, hidden_dim,
        kernel_size = (3, 3), stride = (1, 1),
        padding = (0, 0)), # (5, 5)
        Conv2dBlockH3W3(hidden_dim, hidden_dim*2),
        Conv2dBlockH3W3(hidden_dim*2, hidden_dim*4),
        PixelShuffle2D(2, 2), # (10, 10)
        Conv2dBlockH5W5(hidden_dim, hidden_dim*2),
        Conv2dBlockH5W5(hidden_dim*2, hidden_dim*4),
        PixelShuffle2D(2, 2), # (20, 20)
        Conv2dBlockH5W5(hidden_dim, hidden_dim),
        Conv2dBlockH5W5(hidden_dim, hidden_dim)
        )
```

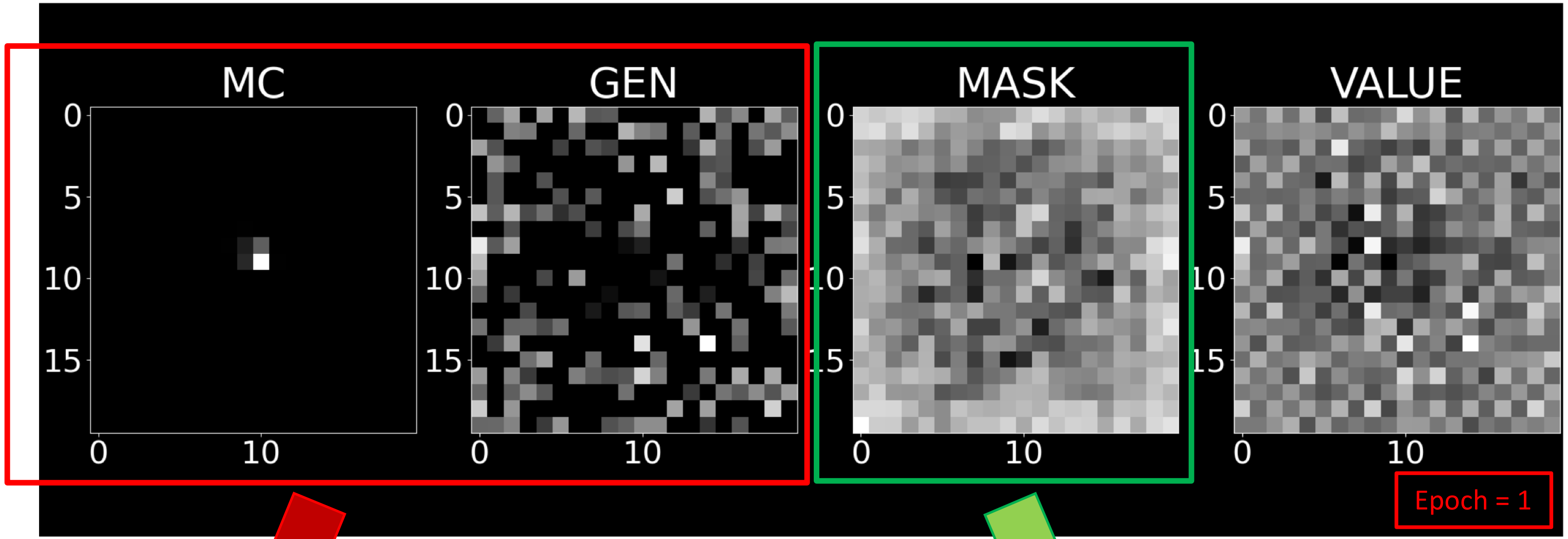
Extended 2

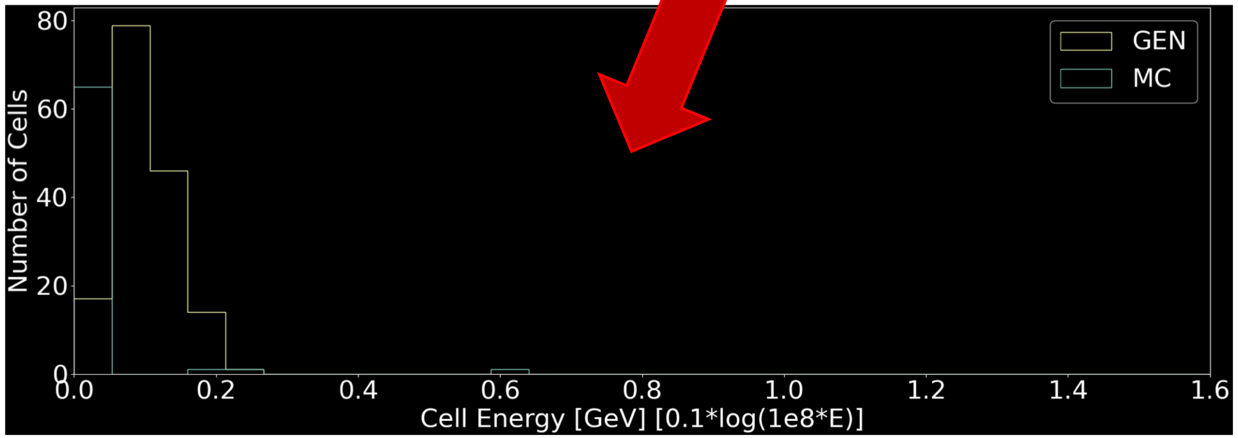
```
self.encoder = nn.Sequential(
        nn.Linear(1, hidden_dim*4), #0
        LinearBlock(hidden_dim*4, hidden_dim*4, 4), #1
        LinearBlock(hidden_dim*4, hidden_dim*4, 4), #2
        LinearBlock(hidden_dim*4, hidden_dim*4, 4), #3
        LinearBlock(hidden_dim*4, hidden_dim*9, 4), #4
        nn.Dropout(0.1), #5
        nn.Unflatten(1, (hidden_dim, 3, 3)), # (3, 3) #6
        nn.ConvTranspose2d(hidden_dim, hidden_dim, k
        ernel_size = (3, 3), stride = (1, 1),
        padding = (0, 0)), # (5, 5) #7
        Conv2dBlockH3W3(hidden_dim, hidden_dim*2), #8
        Conv2dBlockH3W3(hidden_dim*2, hidden_dim*2), #9
        Conv2dBlockH3W3(hidden_dim*2, hidden_dim*2), #10
        Conv2dBlockH3W3(hidden_dim*2, hidden_dim*4), #11
        nn.Dropout(0.1), #12
        PixelShuffle2D(2, 2), # (10, 10) #13
        Conv2dBlockH5W5(hidden_dim, hidden_dim*2), #14
        Conv2dBlockH5W5(hidden_dim*2, hidden_dim*2), #15
        Conv2dBlockH5W5(hidden_dim*2, hidden_dim*2), #16
        Conv2dBlockH5W5(hidden_dim*2, hidden_dim*4), #17
        nn.Dropout(0.1), #18
        PixelShuffle2D(2, 2), # (20, 20) #19
        Conv2dBlockH5W5(hidden_dim, hidden_dim), #20
        Conv2dBlockH5W5(hidden_dim, hidden_dim), #21
        Conv2dBlockH5W5(hidden_dim, hidden_dim), #22
        Conv2dBlockH5W5(hidden_dim, hidden_dim) #23
        )
```

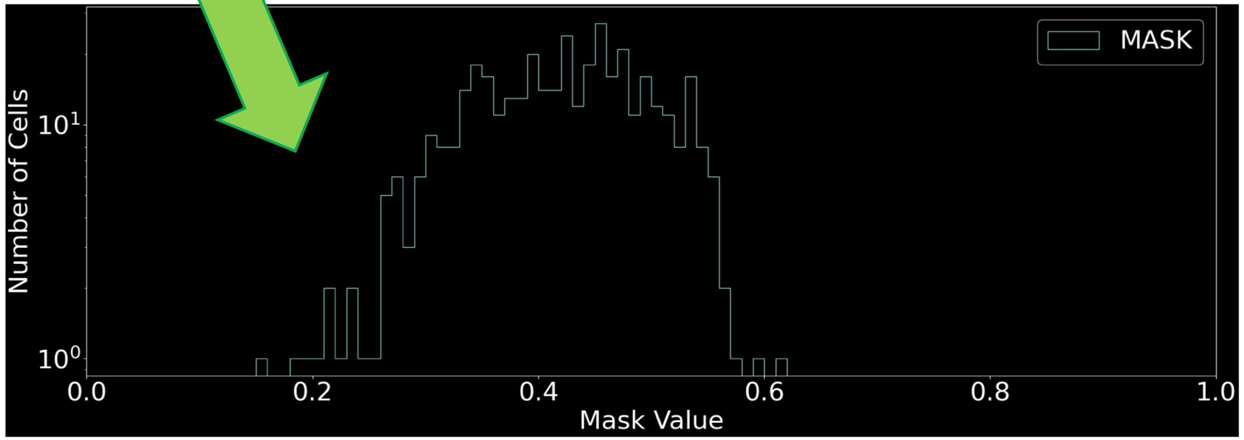
Reminder: Test result (image)

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

Besides the image itself, we have 2 histogram to study the model.

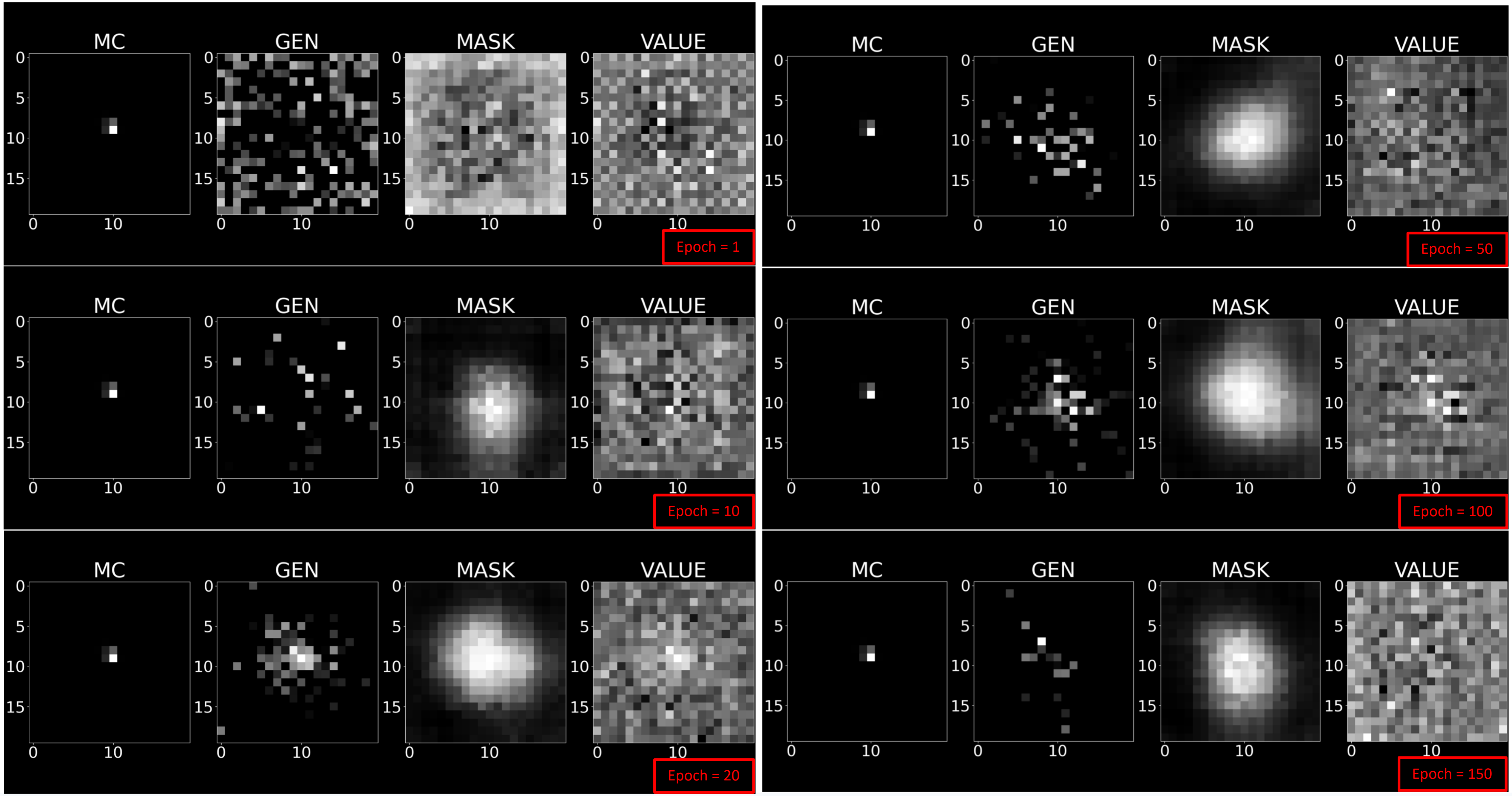






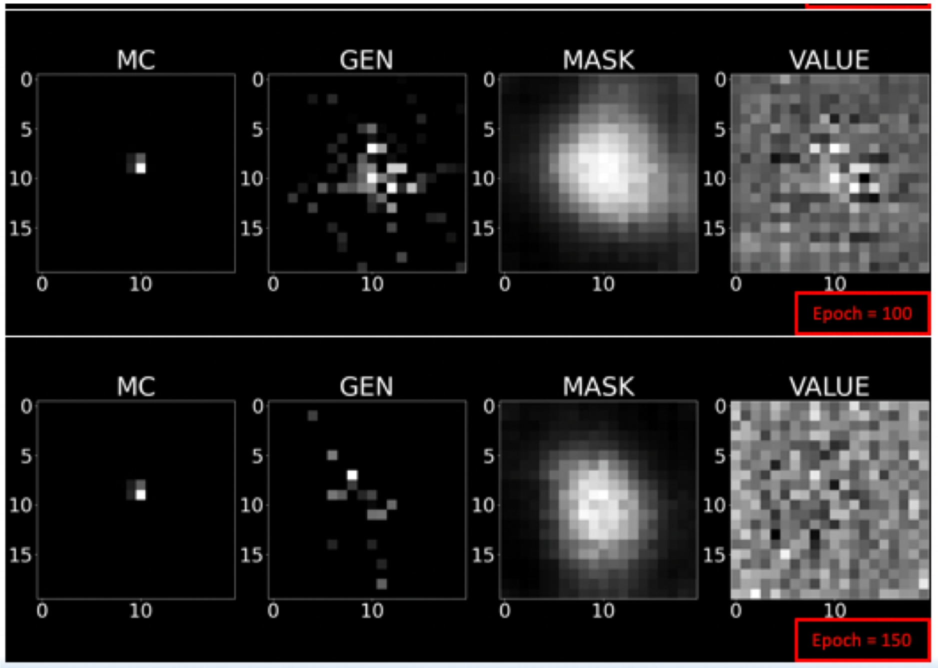
Reminder: Test result (image)

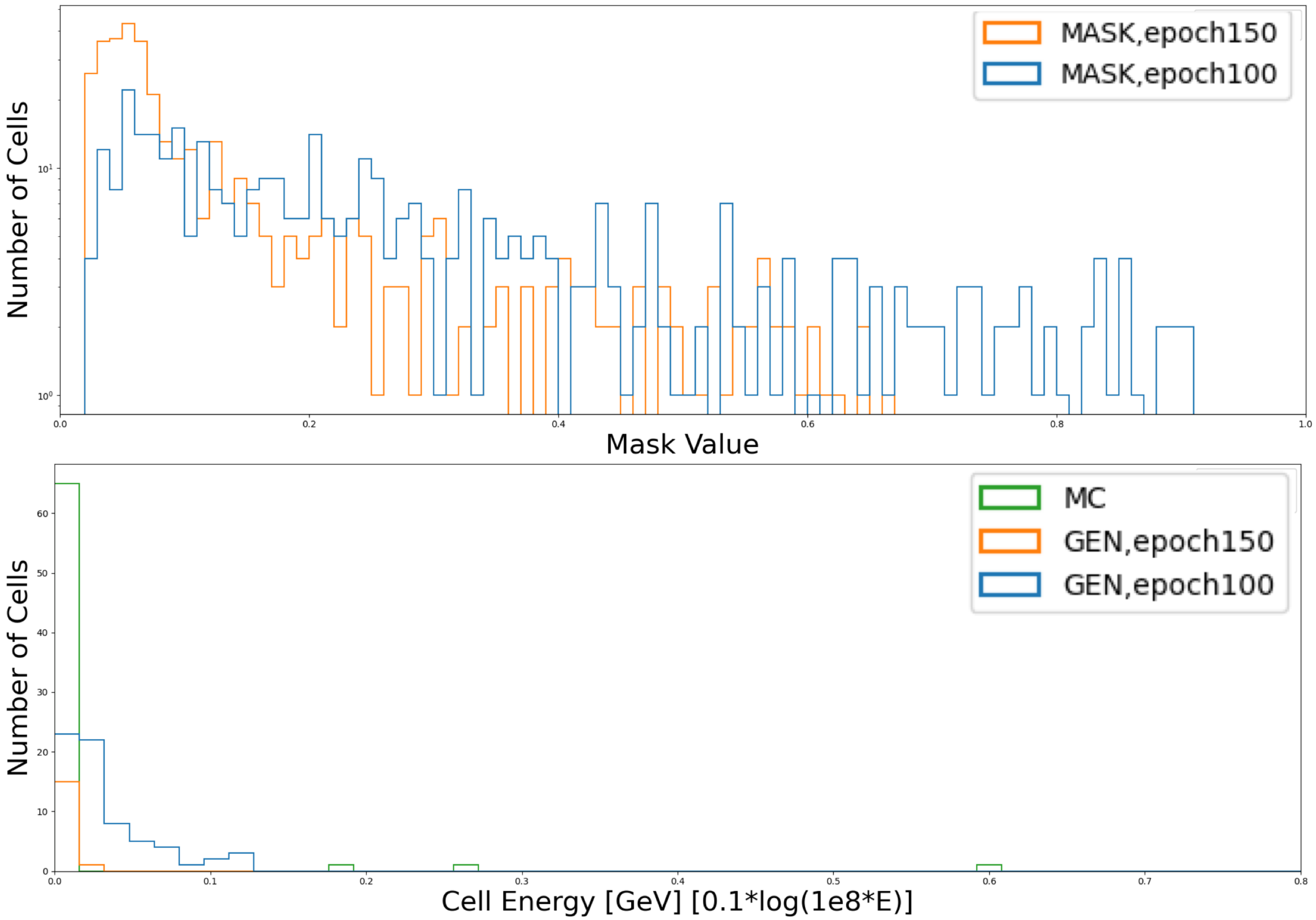
中央研究院物理研究所
INSTITUTE OF PHYSICS, ACADEMIA SINICA



Reminder: What do we learn from these plots

- We can see that the model is trying to balance the MC-mimicking and the mask range from the hist.
- Which is very clear if we overlay the hist.



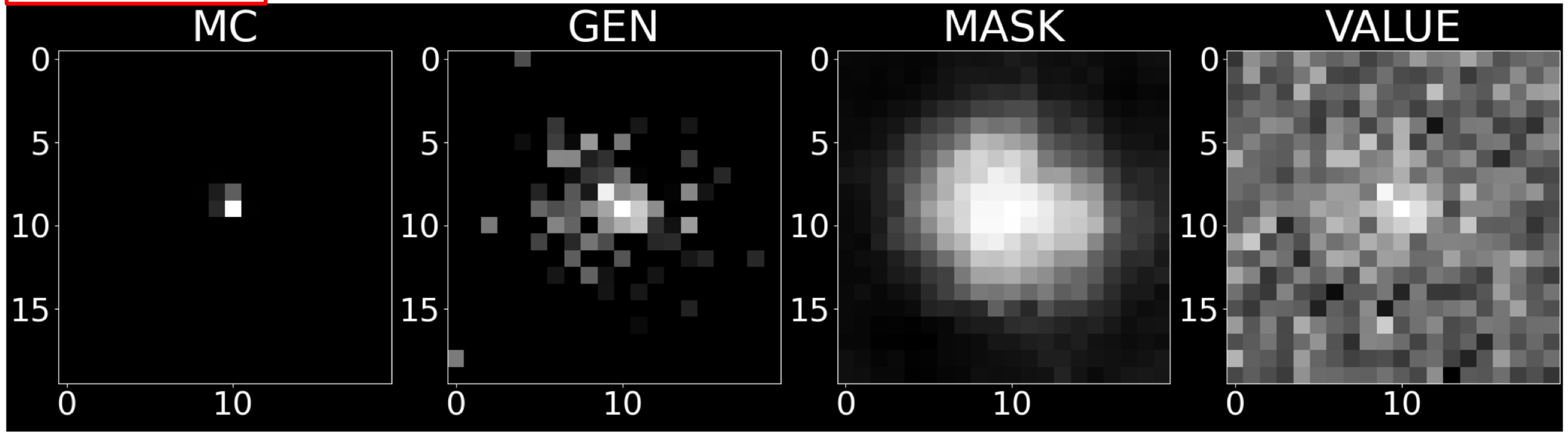


Progress

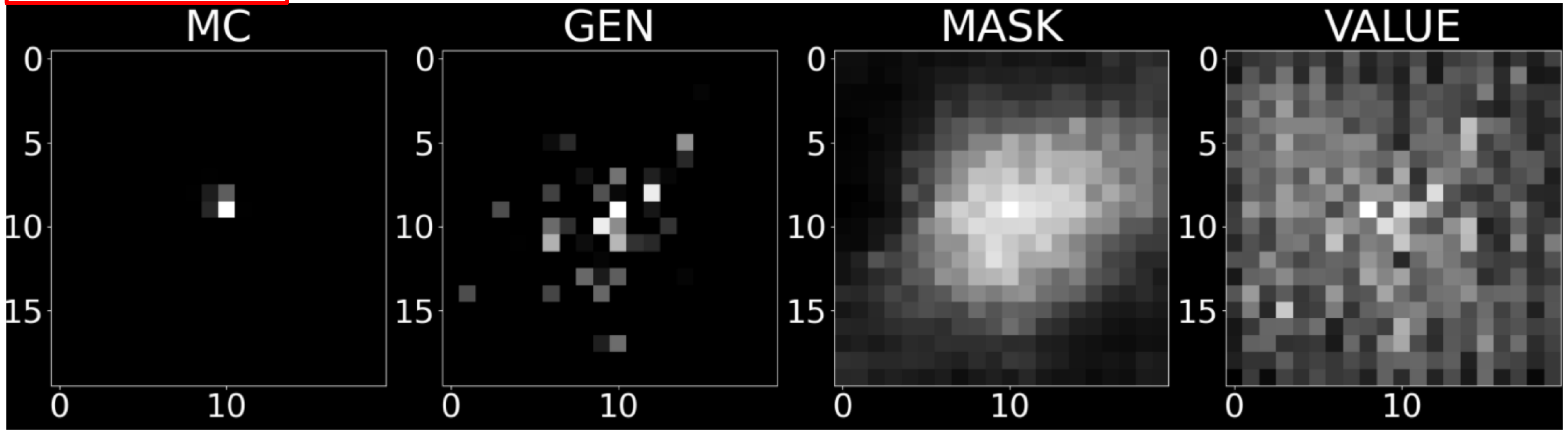
- Produced image with default CNN
- Add weighting into the Loss
- Move the whole setup into work directory (3TB space)
- 50M sample production

Default CNN: Image

Extended 2, Epoch = 20



Default CNN, Epoch = 20



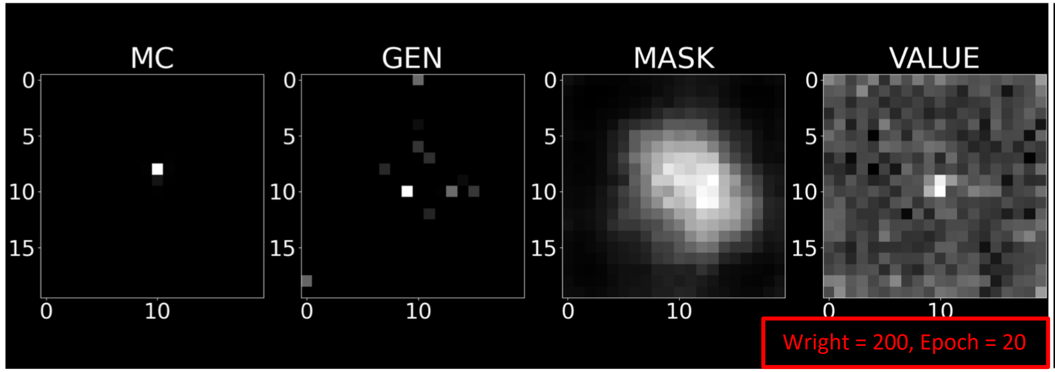
- The modified CNN gives a better result.

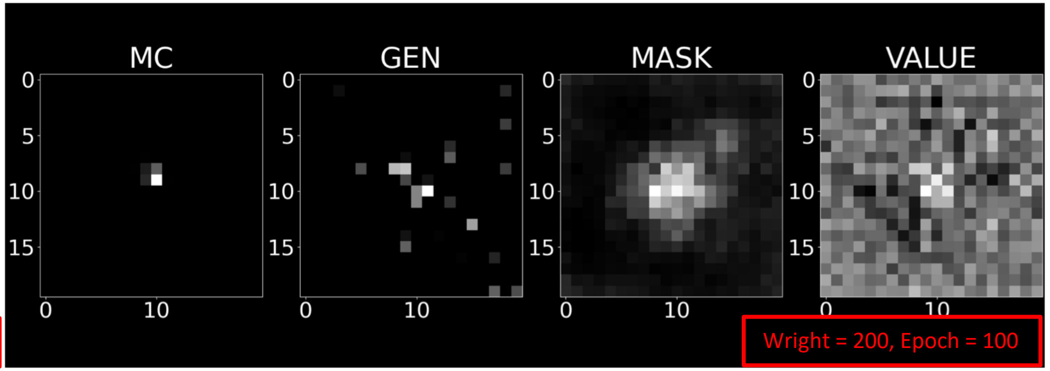
Loss weighting: defintaion

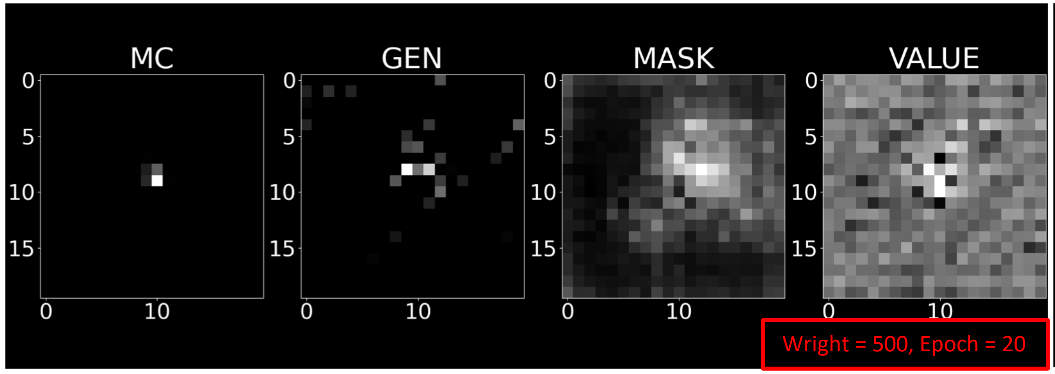
中央研究院物理研究所
INSTITUTE OF PHYSICS, ACADEMIA SINICA

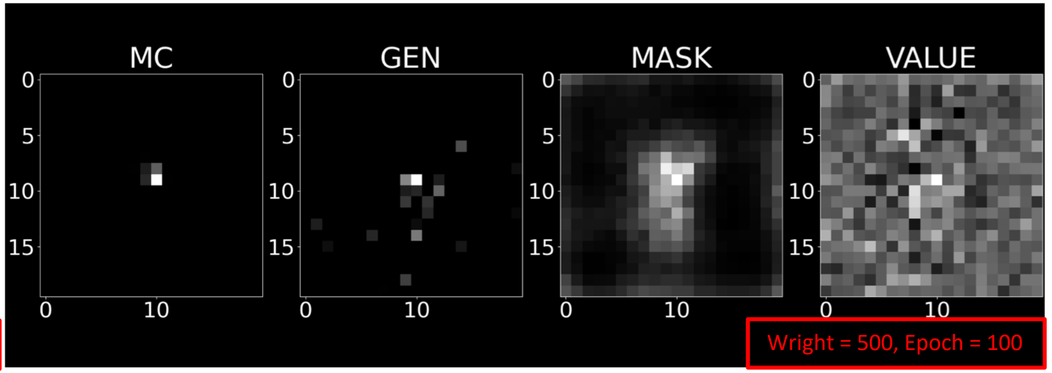
- We removed GAN part of the algorithm so the training loss, $L_{train} = L_{MASK} + L_{VALUE}$.
- Typical value of L_{MASK} and L_{VALUE} are: `Loss(MASK, REG) = (  0.297824,    0.002209)`
- Therefore $L_{MASK} \cong 145\ L_{VALUE}$. In order to balance the loss, we add a weighting to L_{VALUE}.
- We have tried weight = 200 , 500 and 1000 to check the effect.

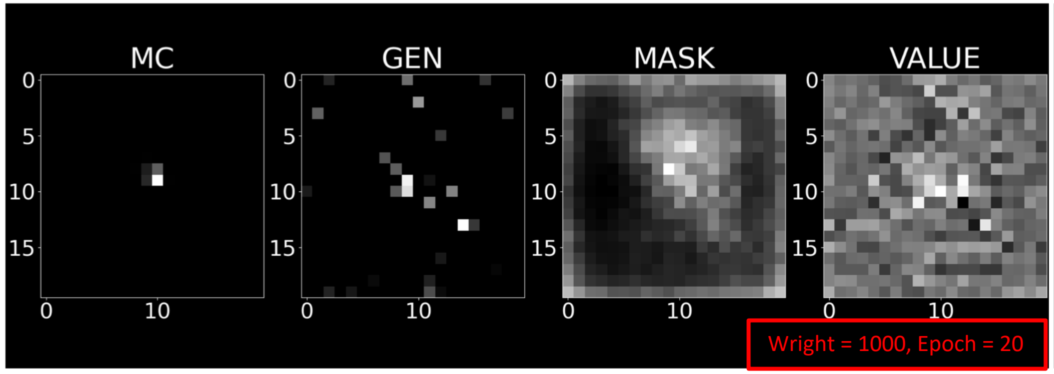
Test result (image)

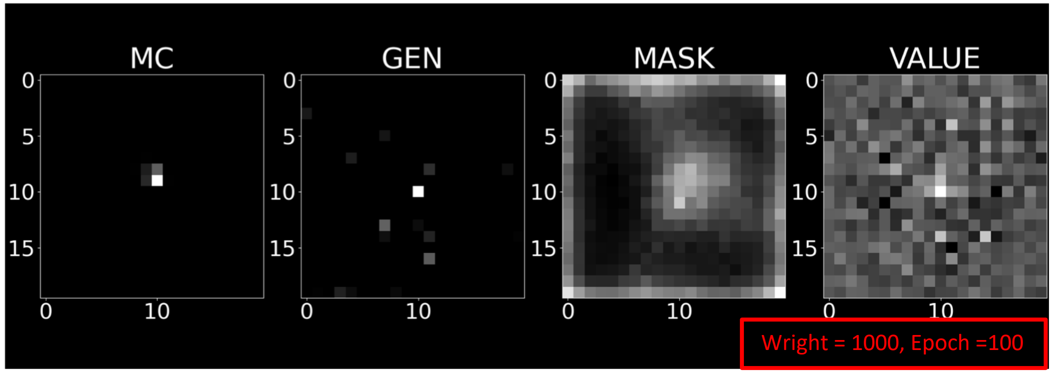






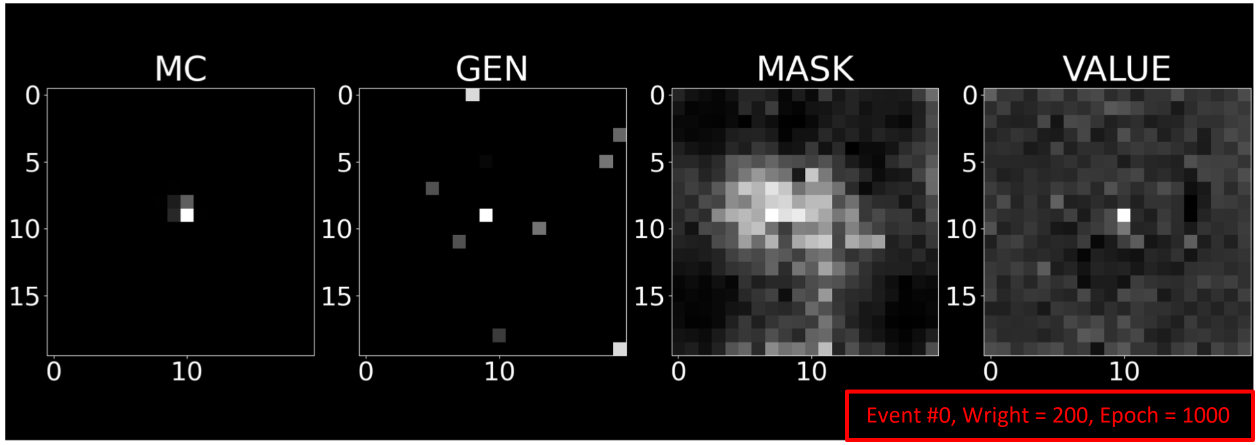


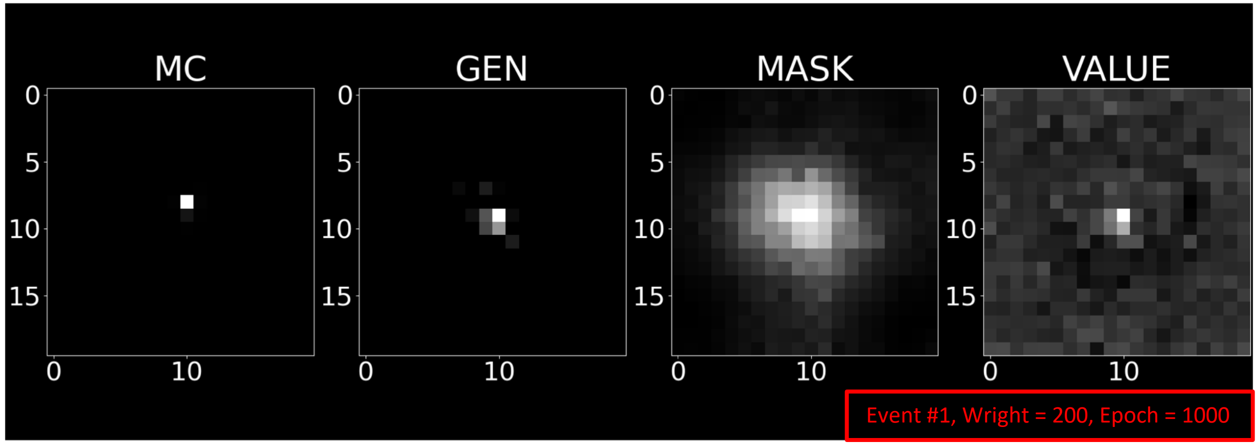




Test result (image)

- We can see that weight = 200 gives a better result so we stick with it, and moved the whole setting to /work
- Here we have much more space (30 times) so we did a 1000 epoch run.

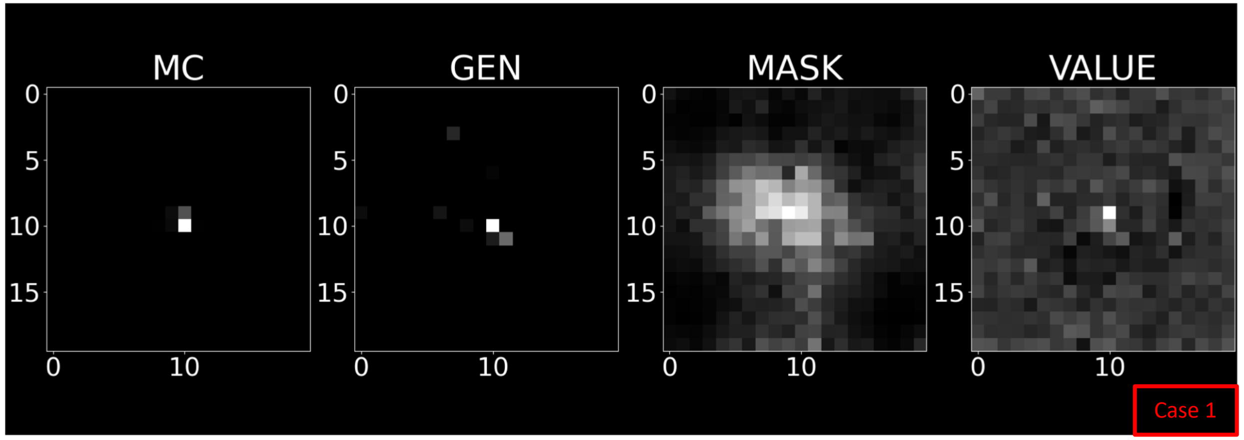


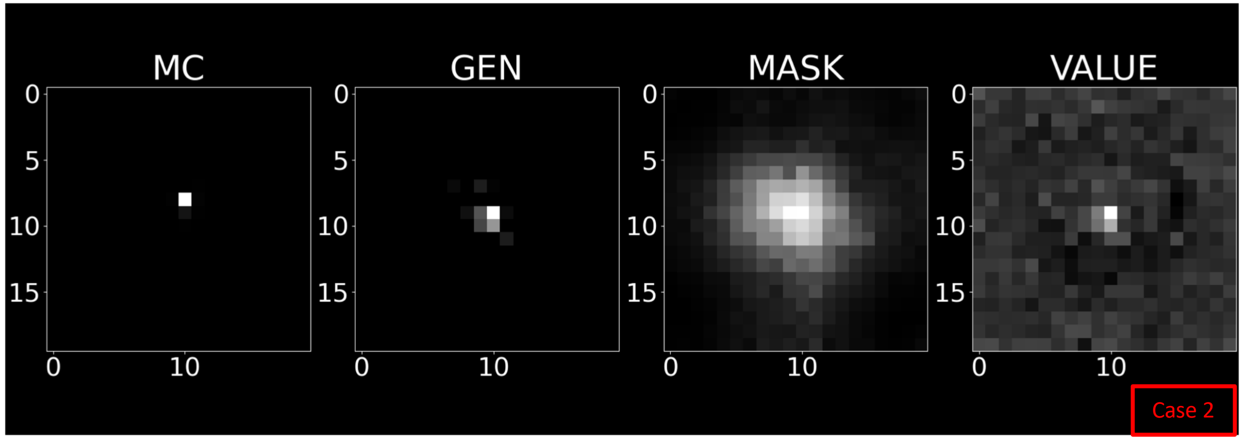


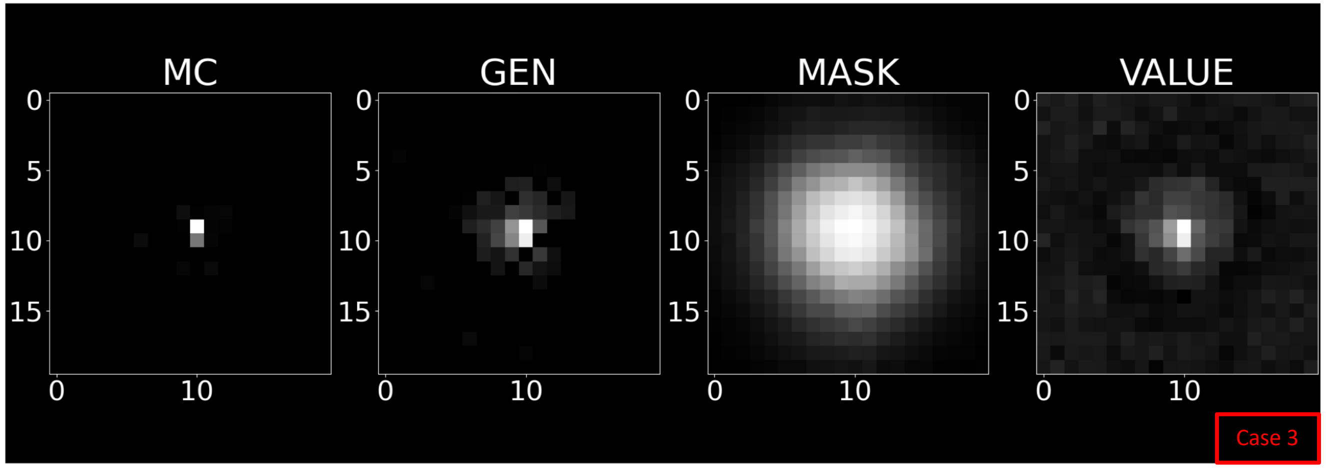
- Although Event#0 (the one we used to compare results so far) looks worse, event#1 gives a much better results.
- Now we wonder how many pattern of MASK image do we have?

Test result (image)

- We have checked 120 events and we can summarise them into 3 types (by eyes)
- We found what we are expecting in Case 3. However Case 1 and 2 are dominating in this 120 events.

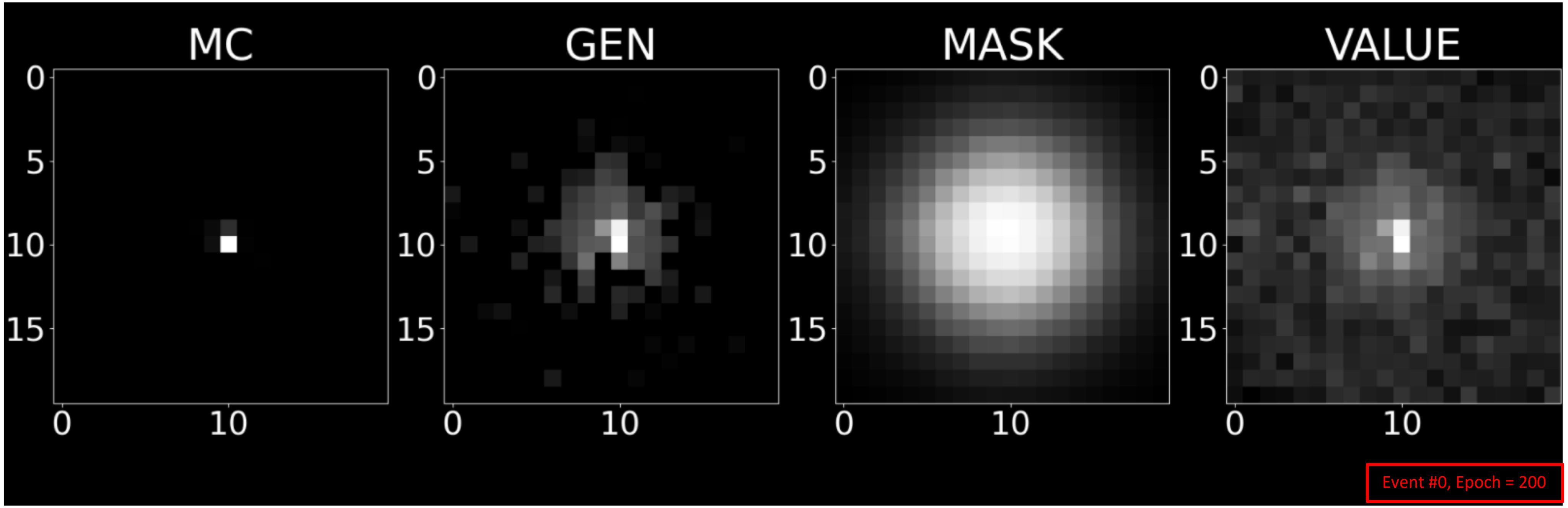






50M samples

- We now have a much larger space to work with, we can therefore train a larger samples.
- Therefore we generate a 50M e^+ sample (5 times than pervious)
- First we check the resulting image by using the default CNN
- Running 200 epochs with $L_{train} = L_{MASK} + L_{VALUE}$ (No weighting)



Summary

- We found that by weighting the L_{VALUE} by 200, we can improve the preference.
- We found that there are 3 types of image.
- More studies needed with 50M sample.