1 Testing Classifiers

We are using a binary classifier to distinguish between Events generated by `Geant4` and (a poorly trained) `Caloclouds`. We are using the following distributions:

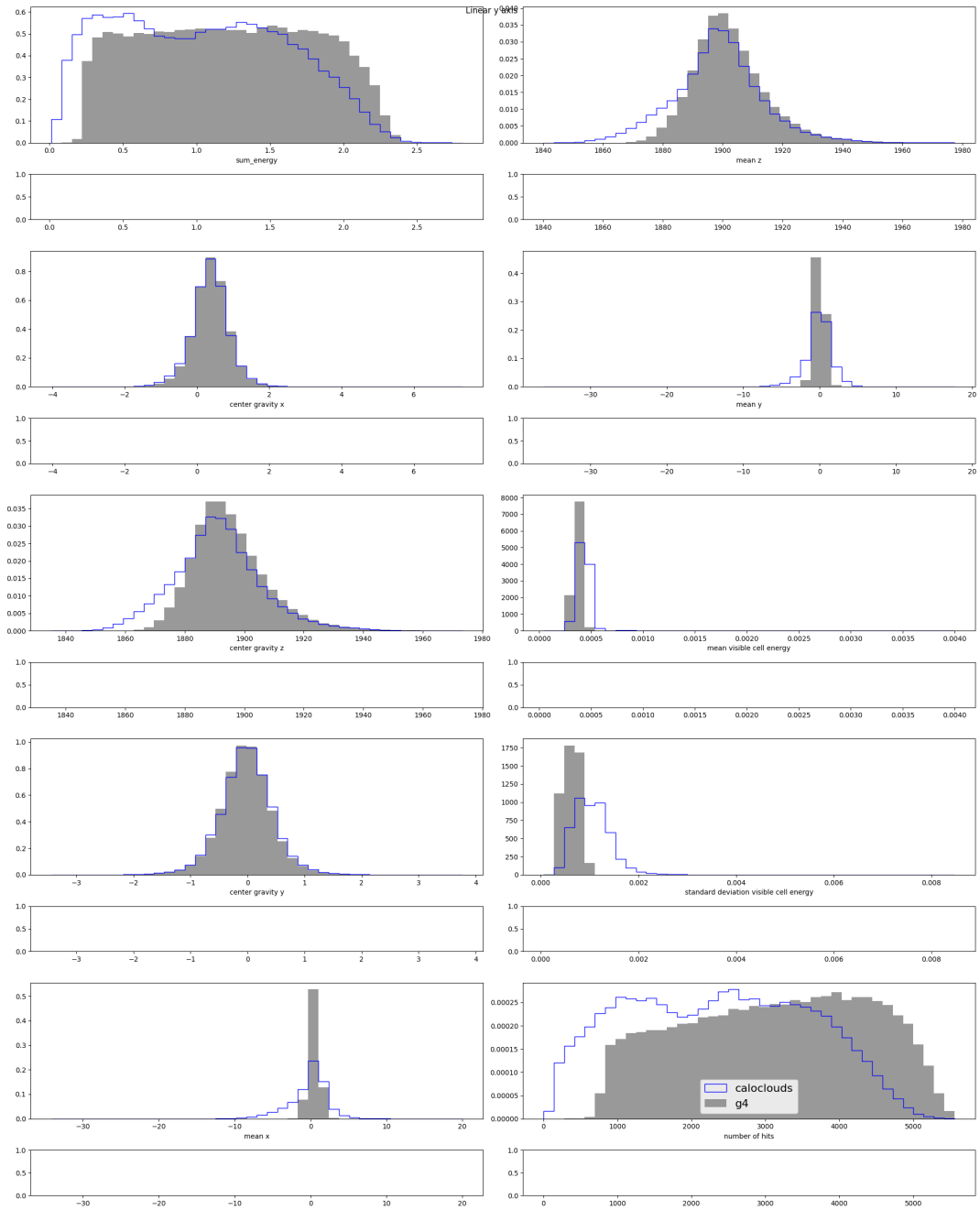


Figure 1: Linear Scale

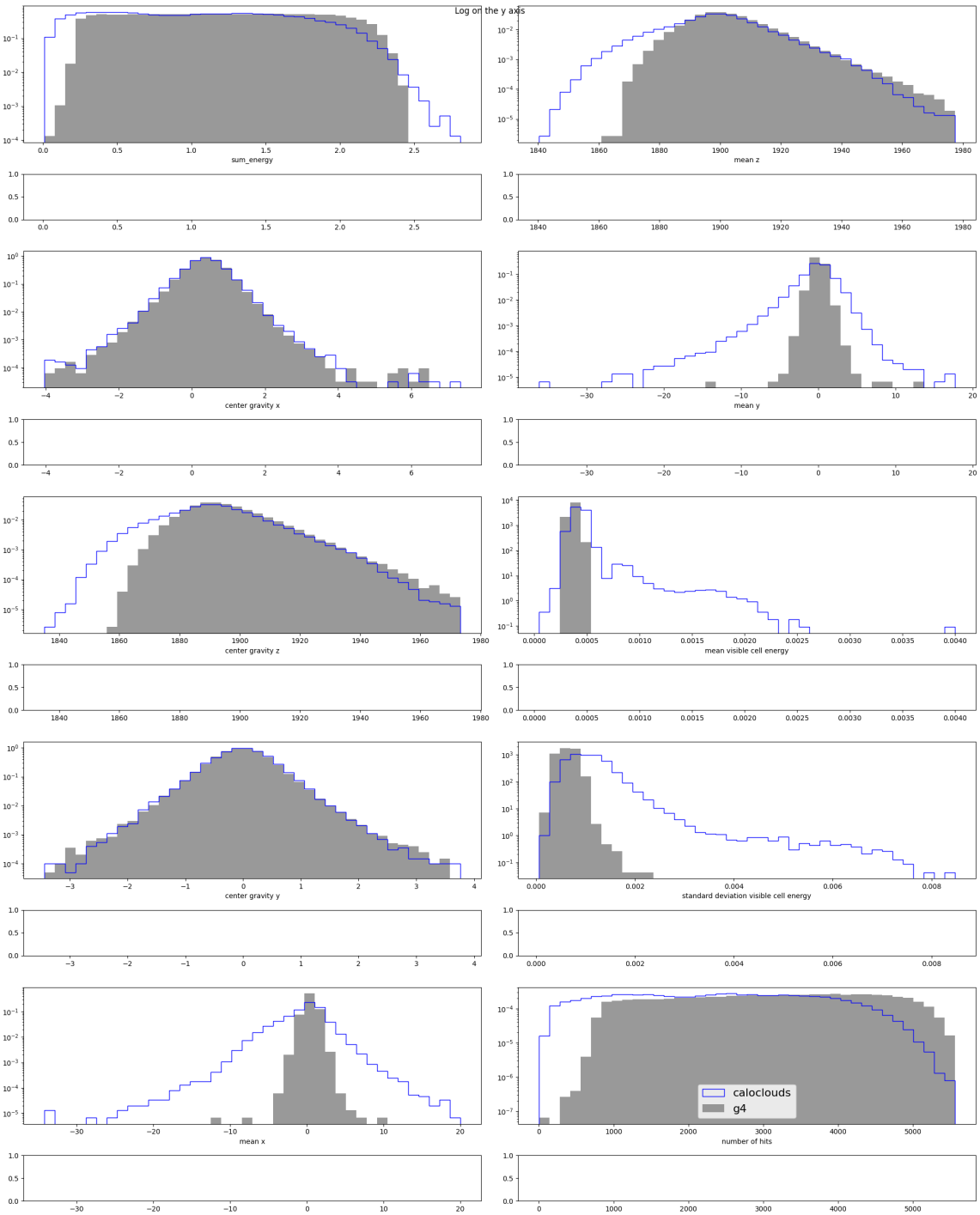


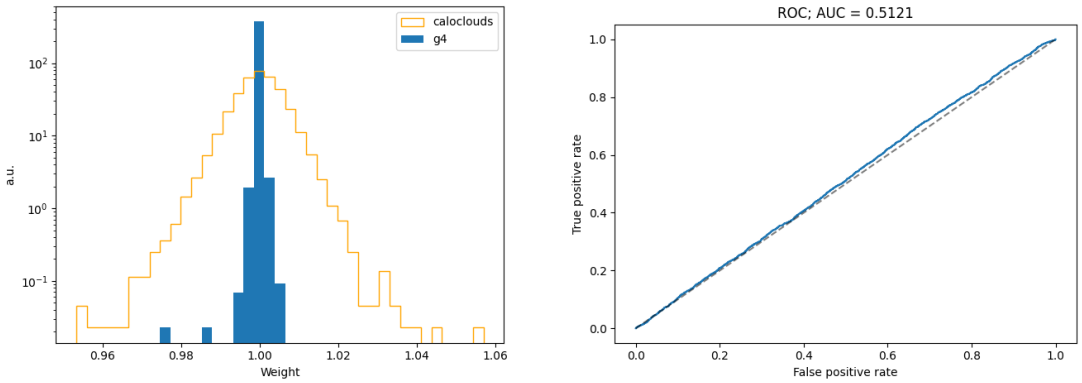
Figure 2: Log Scale

Parameter	Value
Optimizer	Adam
Learningrate	0.001
LR Schedule	Step
Stepsize	10
Decayfactor	0.95
Batchsize	1024
epochs	50
Number of Layers	2
Hidden nodes	5
Dropout	0
Activation Function	leaky ReLU
Training samples	42075
Testing samples	4675
Validation samples	16500

Table 1: Hyperparameters of the Classifier

We notice, that there are some distributions, where `Caloclouds` reproduces `Geant4` very well (i.e. the center of gravity distributions); others look off but not terribly wrong (mean xyz-locations, sum_energy, Number of hits), some are completely off(mean and standard deviation of cell energy). For the case of evaluating the performance of classifiers, this is desireable, as one can exclude distributions from training and thus examine each case individually.
We are using a classifier that is well able to distinguish between the generated and truth datasets when training on all distributions. We can see that in the ROC curve and its AUC score as well as a weight distribution shifted to Values $w < 1$. With $C(x)$ being the Classifier output of Event x, $w = \frac{C(x)}{1-C(x)}$. For reference, the classifier is also trained on a heldout `Geant4` Dataset to show, how Data drawn from the same distribution as the Truth dataset performs.

When training only on good distributions (center of gravity in x- and y direction), the Classifier performs not significantly better than a random Classifier, as expected; AUC$\approx$ 0.5. While the Weight distribution is centered around 1 as well, it is significantly broader than for the Classifier trained on the heldout `Geant4` dataset, indicating phasespacevolumes, where `Caloclouds` over- or underproduces events; however in average the generator performs good.

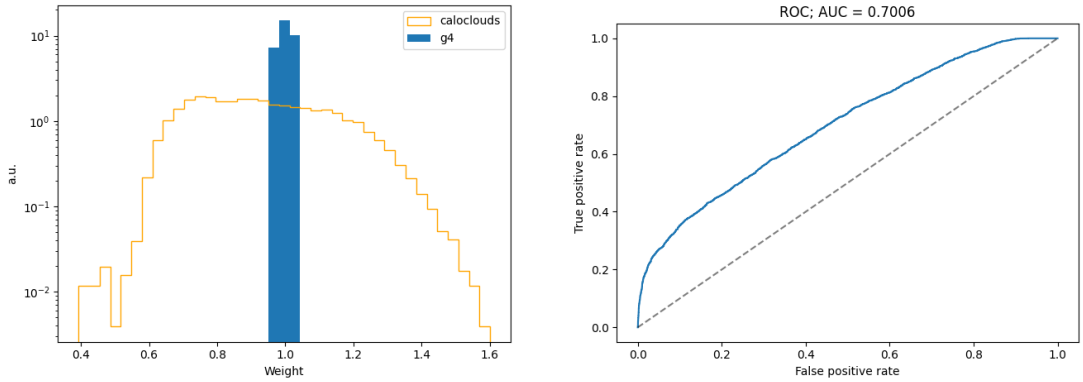


(a) Weight distributions

(b) ROC Curve

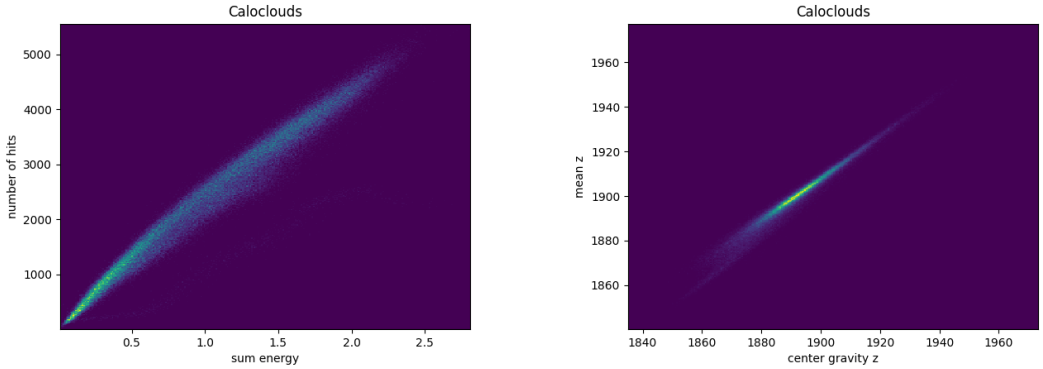
Figure 3: Training only on good distributions

When the Classifier is trained on all distributions, the AUC is significantly larger than 0.5, idicating, that it can distinguish the Datasets qutie good. Also, the Weight distribution is clearly shifted to weights $w(x) < 1$, as expected; however, there is also a non neglectible amount of events with weight $w(x) > 1$, again indicating phasespace volumes where `Caloclouds` underestimates the density.
Having a closer look at the distributions, we notice that some of them are strongly correlated, in `Geant4` and `Caloclouds`, for example Sum Energy/Number of hit or mean z/center gravity z.



(a) Weight distributions (b) ROC Curve

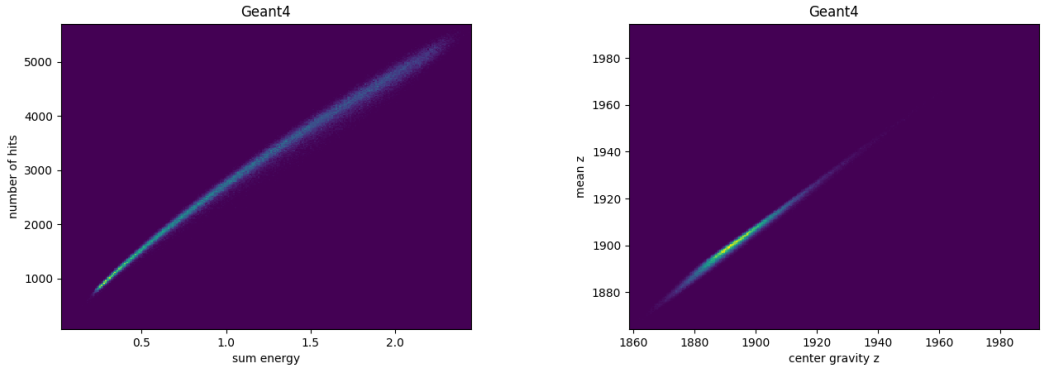
Figure 4: Training on all distributions



(a) Correlation between Sum Energy and Number of hits (b) Correlation between mean z and center gravity z

Figure 5: Correlations for `Caloclouds`

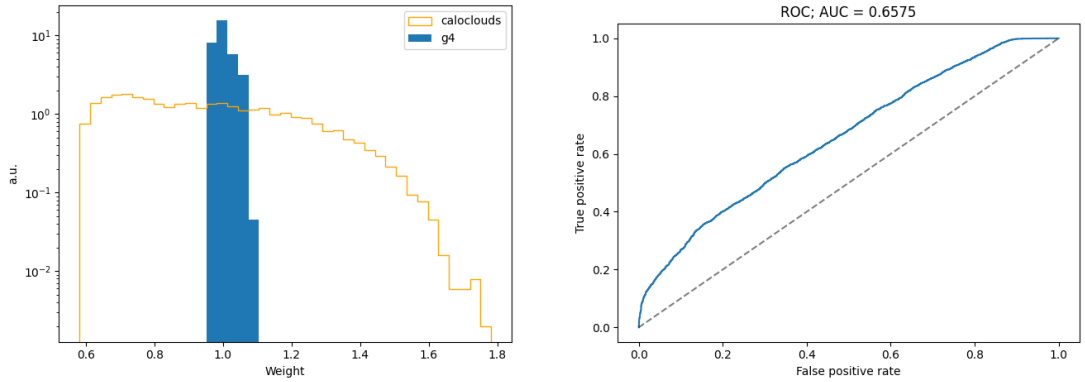
This sparks the question of how the Classifier responds to this input. We train the Classifier on the good dimensions and both correlated distributions as well as only on one correlated distribution.
It is clear, that showing both distributions does not improve the AUC; the difference between here is perhaps just down to noise and increased network size for a larger input. We can also see, that the weight distribution gets more sensitive, especially to high event weights if the Classifier sees both distributions. The same effect can also be seen, if we train on all distributions exept one of the correlated ones:
Note, that this can unfortunately not be done for the x and y mean/center of gravity distributions, as their 2d correlations for `Caloclouds` resemble more a 2d-gaussian distribution than a linear relationship.



(a) Correlation between Sum Energy and Number of hits

(b) Correlation between mean z and center gravity z

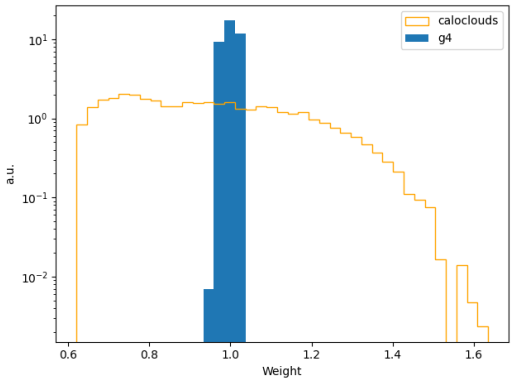
Figure 6: Correlations for `Geant4`



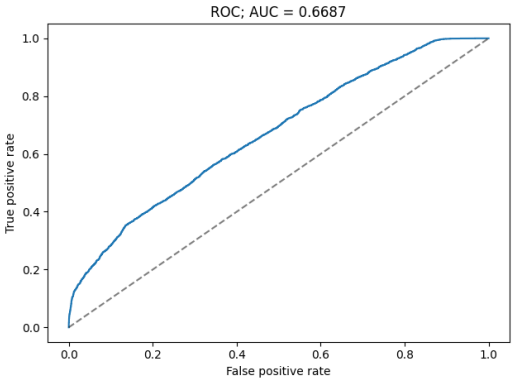
(a) Weight distributions

(b) ROC Curve

Figure 7: Training good and both correlated Sum Energy and Number of hits distribution

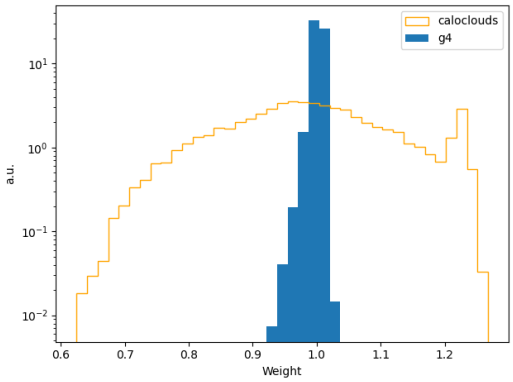


(a) Weight distributions

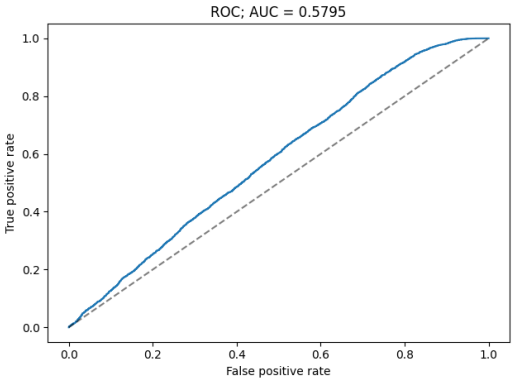


(b) ROC Curve

Figure 8: Training good and only one correlated (Number of hits) distribution

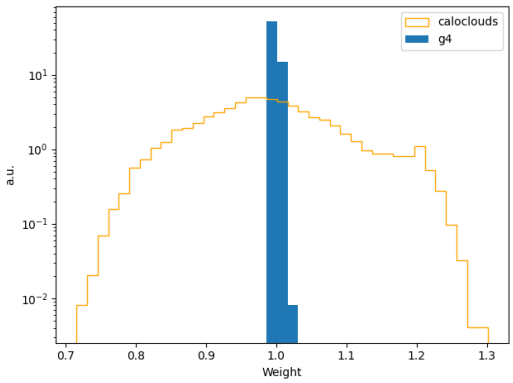


(a) Weight distributions

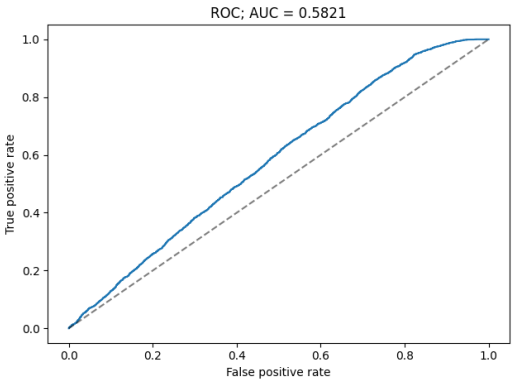


(b) ROC Curve

Figure 9: Training good and both correlated mean z and center gravity z distributions

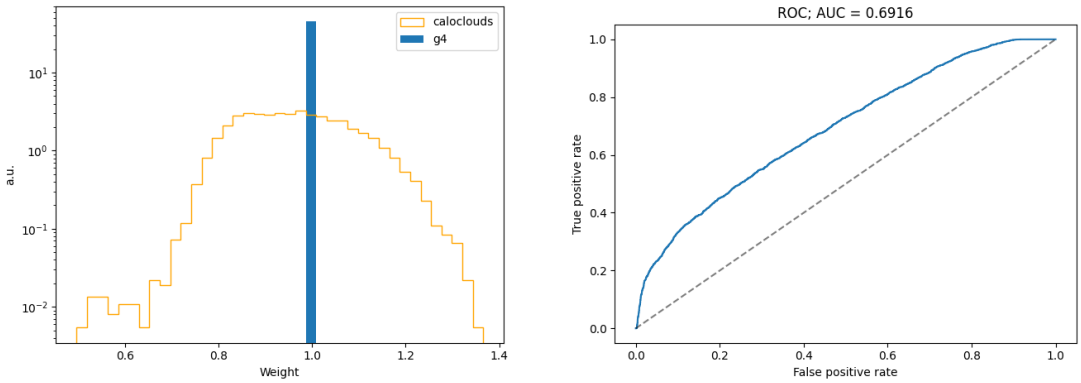


(a) Weight distributions



(b) ROC Curve

Figure 10: Training good and only one (center gravity z) distributions



(a) Weight distributions (b) ROC Curve

Figure 11: Training all but one of the corrolated distributions

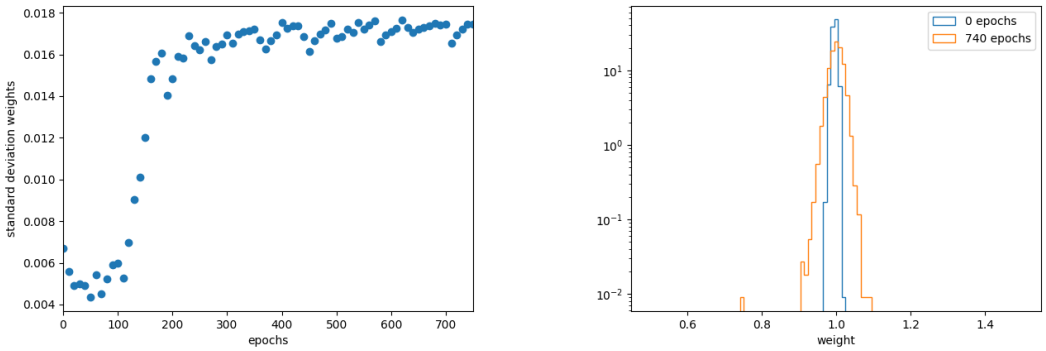


Figure 12: Evolution of the Standard deviation of the weight distribution of `Geant4` trained against itself on a heldout dataset