**Diagram of the final generative model architecture, and scientific discussions on its design [40]**

**Recognisability of the best output [20]**

**Realism of the best output [20]**

**Unique of the outputs [20]**

So 60 of the marks are simply for good images. I’m going to want a more robust way to produce my images, therefore.

I may as well train on the test dataset. Nah. What I’m struggling to get my head around is identifying good Pegasus. Do we actually want those that the system cannot identify? I think it’s better to, for our training dataset, save every image individually, with an alpha value. Then I’ll have to judge them subjectively.

The reconstruction loss $L(x, x^)

The former term encourages our model to be sensitive to inputs, and the latter discourages overfitting. A scaling parameter is typically included in front of the

\verb|torch.nn.Linear| has by default He/Kaiming uniform initialisation, as confirmed here.

As images are normalised between 0 and 1 by \verb|transforms.ToTensor|, a sigmoid activation has been added to the output layer to get values that match this input value range.

Binary cross-entropy was found to offer marginal improvements over mean-squared, though subjectively.

Spectral Normalisation was also trialled. It was found to improve the convergence rate slightly, from around $250$ epochs to $200$.

# Might want to graph those two. As proof, y’know.

The larger the compressed encoding of the image, the closer a combination of two encodings is to simply naively combining two images.

A successful approach therefore *must* find the latent space.

Using \verb|LeakyReLU| activation functions with a negative slope of 0.2 appears to be current best practice, with use in both ACAI and PIONEER.

Minor data augmentation also found to *decrease* performance of the model. Specifically, a random crop, and a random horizontal flip. The introduction of noise was deemed a complicating factor; it would have necessitated a re-design of the AE into a *denoising* AE.

Even with spectral normalisation and overtraining the discriminator, loss fluctuated wildly, though not in the typically sinusoidal fashion expected (in which, as the discriminator loss decreases, the autoencoder losses increases, and vice versa).

The values for \verb|learning\_rate| and \verb|weight\_decay| given in the paper were found to inhibit performance compared to the defaults.

Over-training the discriminator was found to negatively impact the performance of the autoencoder, due to the decreased number of training images.