

ELEC 475 Lab 2

William Dormer, 20176544

Description of Hardware Used:

I trained on my desktop PC using my graphics card (GPU) for both the 1k and 10k stages. I have an NVIDIA GTX 1070. My processor is a 12th Gen Intel Core i7-12700.

Hyperparameters:

Common Hyperparameters

- resize to 512x512
- random cropping images to 246
- Optimizer: ADAM
- Learning rate: 1e-4
- batch size: 8
- epochs: 40

I was having RAM issues, and I don't have a very powerful graphics card, hence the small batch sizes.

1K training:

- Learning rate decay: 1e-4
- gamma: 0.3

I landed on these hyperparameters after several iterations of trying different things out with the model.

I Noticed that the learning rate seemed not to be decaying fast enough, so I thought I'd increase the lr decay for the 10k. I'll also give a bit more weight to the style with higher gamma.

10k training:

- Learning rate decay: 3e-4
- gamma: 0.6

Source Implementations

I followed primarily the guidance of the provided github for the training loop implementation and supporting functions: <https://github.com/naoto0804/pytorch-AdaIN/blob/master/train.py>

Other than that, I simply used what was already provided in the lab code, and the materials that we had available to us in the lecture notes.

Training time approximations

1K

68 seconds per epoch

40 epochs

45 min total train time

10K

647 seconds per epoch

40 epochs

7 hours 11 min total train time

Gamma value explanation

Originally I was using a gamma of 5 (the default suggested in the AdaIN implementation), because I thought that the loss for the style would be more important, however it was losing content features. So I adjusted the gamma down to 1, where the loss from the style and content is balanced for training. That gave pretty good results, but I was still losing out on content features, so I further reduced it. For the trial with the 1k image set, I used 0.3, which worked well. However, it did lose some of the style details, so for the 10k run, I decided to boost it back up to 0.6 to capture more of the style details. However, I believe I overshot this. If I ran it again, I would set gamma to 0.4 or 0.5.

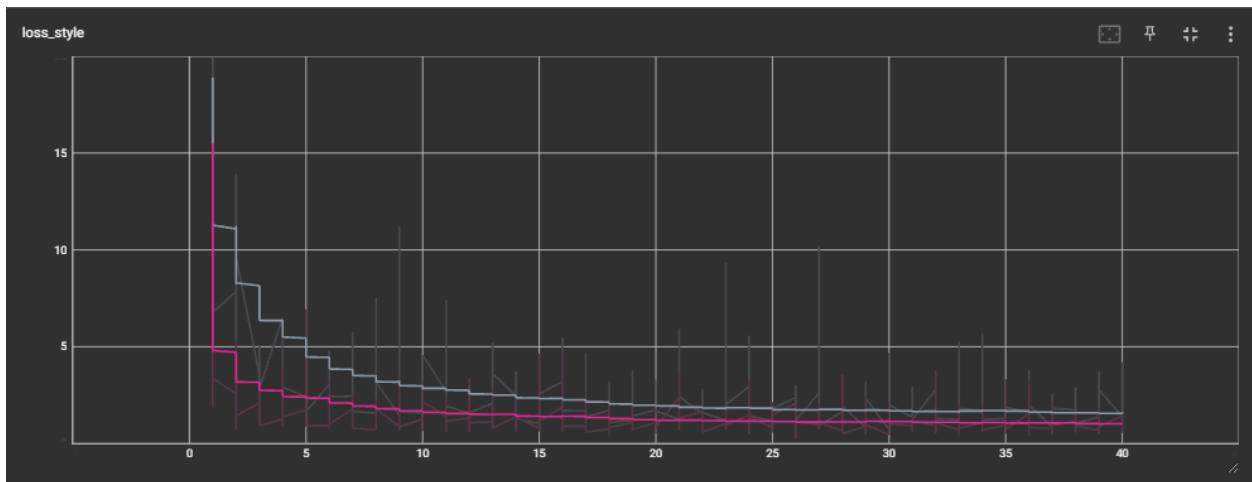
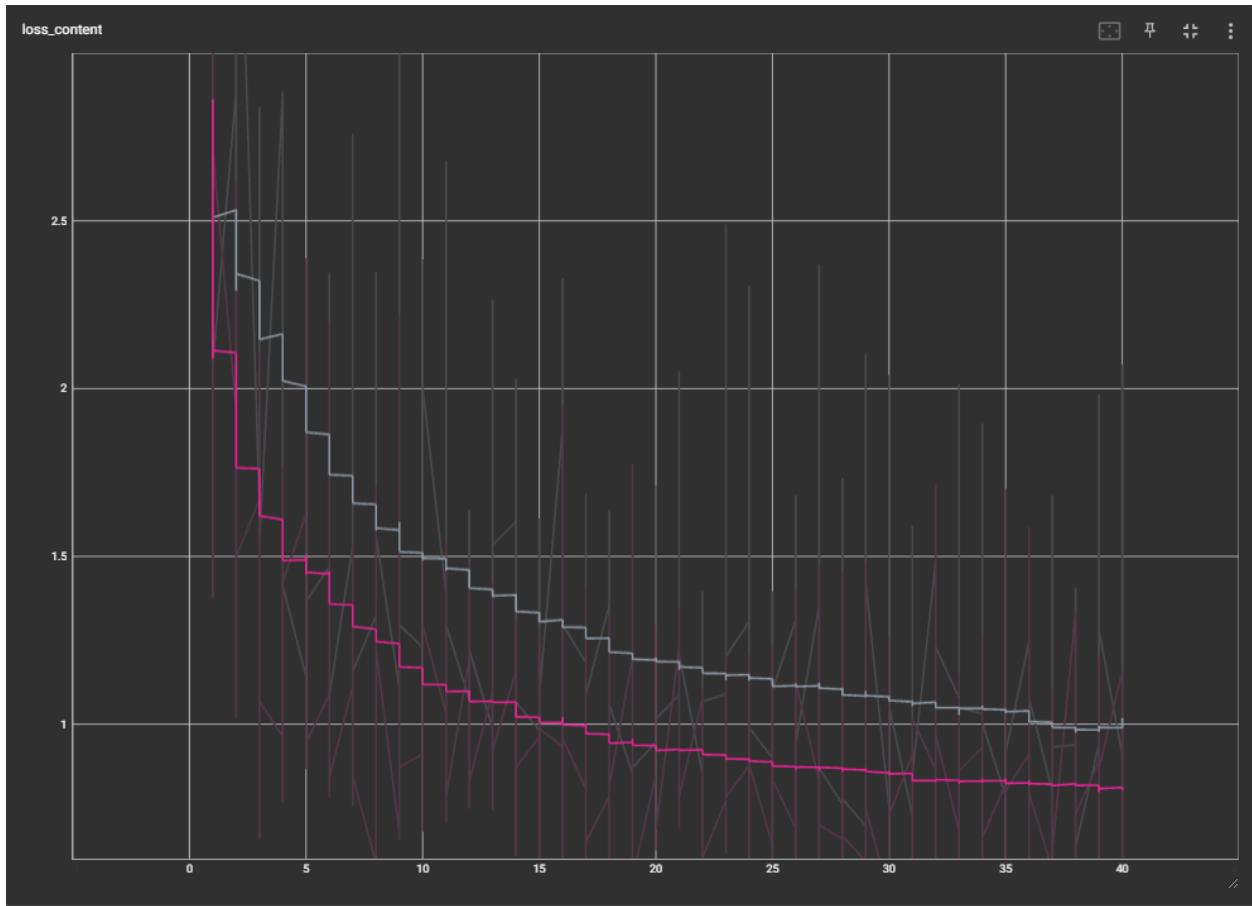
Differences in Loss Curves For 1k vs 10k test

My loss plots were generated using tensorboard, using the values collected by the Writer during training. You can use the tensorboard viewer if you want to look at the graphs yourself.

type tensorboard --logdir="src/logs/" into the terminal.

You will not see the loss plots in the zip file, since the data is in a different format controlled by tensorboard within the logs folder. The two that are relevant are "onek_gamma0point3" for the 1k logs, and "tenk_40epoch_06gamma" for the final 10k logs.

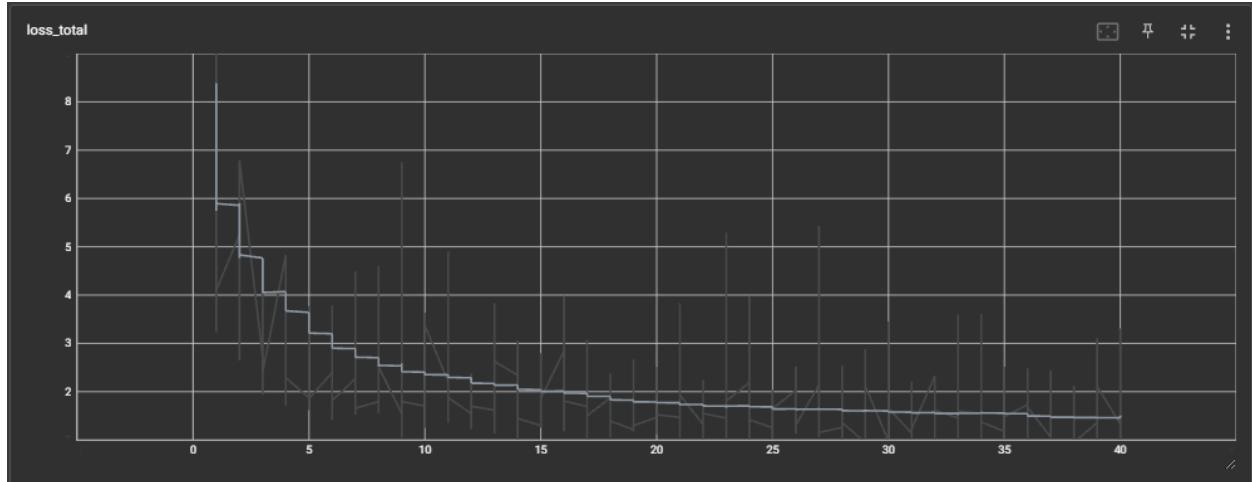
You have to be careful looking at the absolute value of the total_loss to compare the results because the gamma changes the magnitude of the results, and thus they can't be compared one to one. However, for the individual style and content loss they can be compared for the 10k and 1k models.



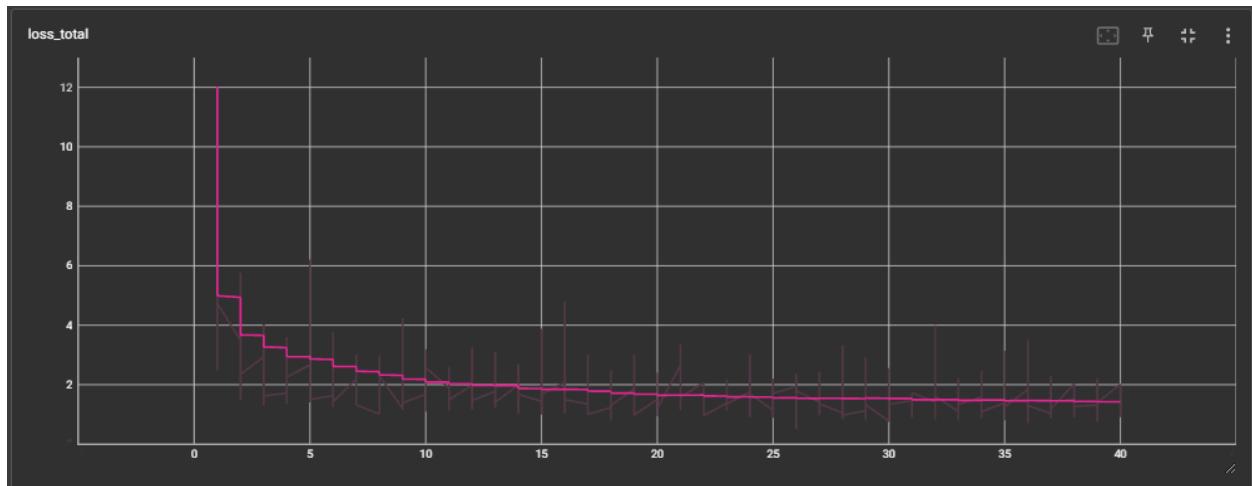
Where the purple is the 10k, and gray is the 1k. As you can see, the loss for both content and style is better for the tenk version, which is unsurprising given the larger dataset, and thus more

batches per epoch. The loss in the content seems to have not quite plateaued yet for either model, indicating that training for more epochs could improve the content loss. The style loss, however, does seem to have plateaued. This also supports the idea that I described in the previous section that reducing the gamma for the 10k result could improve results.

1K loss plots:



10K loss plots:



Notice the sharper drop in loss for the 10k result. This is likely due to the increased number of batches, and thus weight updates, per epoch.

The faded values on the graph are the range of the values that each of the batches gave for the loss, and the bold line is the average loss for that epoch.

10k Model Output Examples:

Content images: (all from coco1k)



000000000436 - guy eating donut

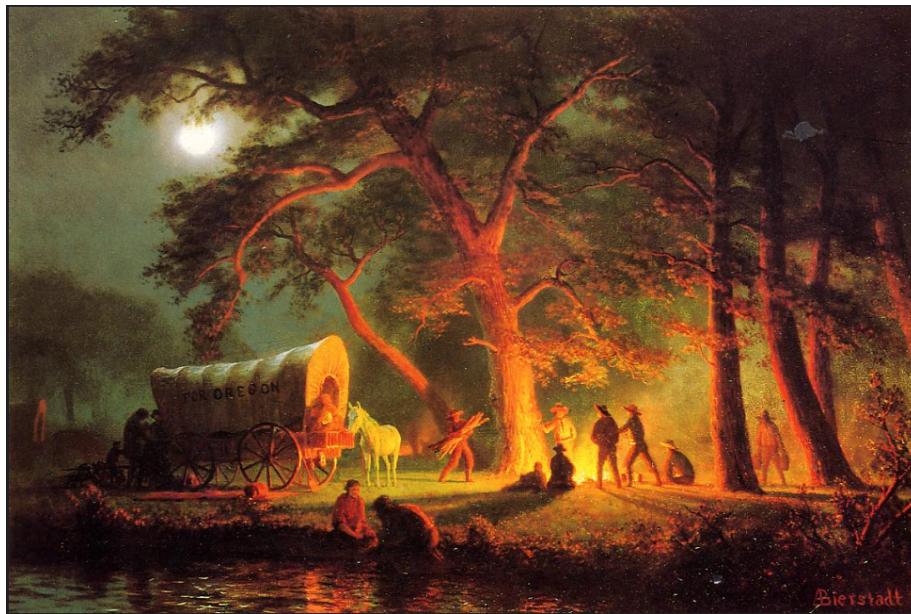


000000064086 - rusty stop sign

Style images: (all from wikiart100)



2b1c40a7550e4279bef2ec1e91167616c - abstract pastels



649044acbd7fd1a759bd0649766cae5c - Oregon trail painting



1d9f48c515e7437a2fa6eeb1c42fc44ec - medieval tapestry

All the following tests for the 10k and 1k models used alpha = 0.1, 0.5 and 0.9

10K tests:

abstract pastels - donut guy



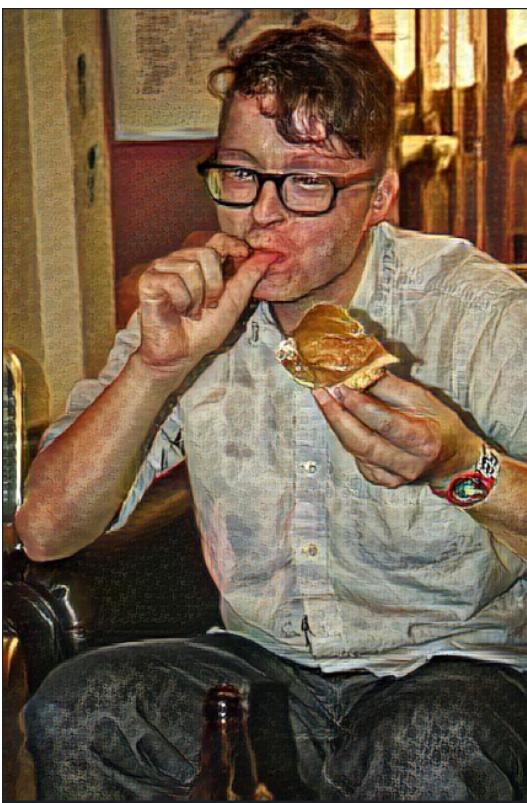
it definitely is picking up on the colour palette, and the blotchy patches. But no distinct oil paint elements.

Oregon trail painting - donut guy



It does a pretty good job of picking up the texture of the canvas, and giving that glowing hue dark warm colour scheme.

medieval tapestry - donut guy



I was particularly impressed with this result, because you can see especially in the hands how it's using the very fine detailed but bright style of the tapestry. It also copies the tapestry texture pretty well, especially if you look at the shirt or the pants.

abstract pastels - stop sign





The colors are definitely showing up, but again no definitive oil patches.

Oregon trail painting - stop sign

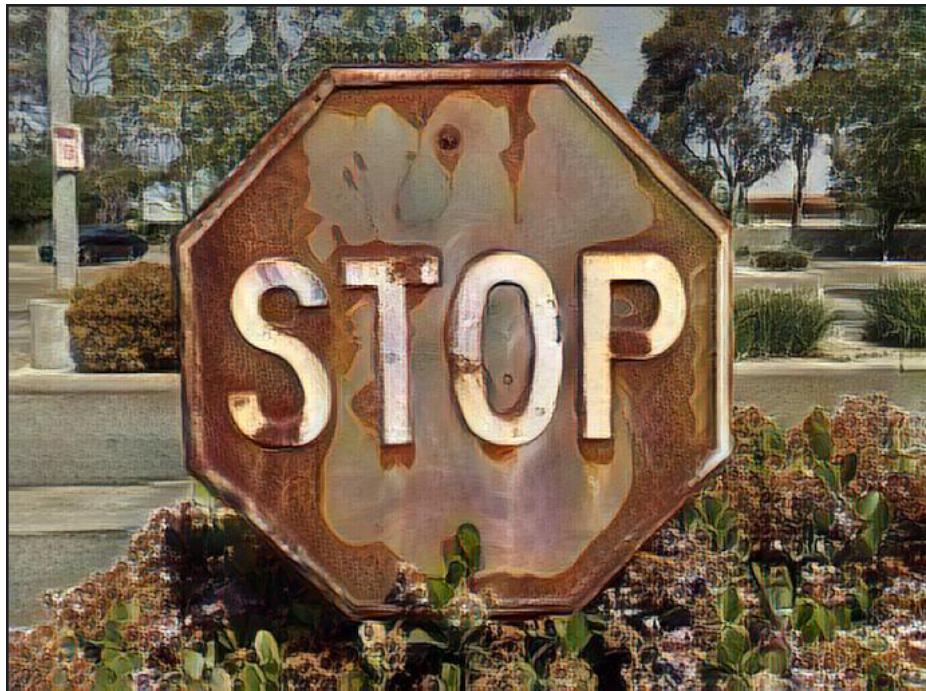




I think the rusty sign already blends fairly well into the Oregon trail painting theme, so we see a dimming of the colors, and a slight change in texture.

Medieval tapestry - stop sign





Similar to the man with the donut, you can see the texture of the tapestry.

Looking over these results, I'm fairly satisfied. The results from the man eating the donut were pretty good for all three, particularly when looking at the texture of the tapestry. Throughout my tests of different style images, the selection of a style image that isn't too busy and has a clear pattern tends to lead to better results (e.g. the brushstrokes one), but I wanted to try something new for the TAs to look at, so I picked some new ones out of the style training sets. Perhaps the rusty stop sign is not the best example image to show the style on, but it does show the level of

content detail that the model is picking up on well, which is why I chose it. Notice that it keeps the detail in the car in the back and the many plants, which shows that the content reconstruction is working well. I expand on this in the next section.

Qualitative Differences between the 1k and 10k tests

Content Loss Comparison:

here's the original for reference again



This is from the 1k model with 0.0 alpha



this is from the 10k model with 0.0 alpha:



As you can see, the reconstruction of the content image is slightly better with the 10k version of the model than with the 1k version, which is what you'd expect looking at the loss plots. For example, look in the back right corner, between the two larger trees, there is a small tree far in the background in the original image. While it isn't constructed completely in either image, the tree's trunk is visible in the 10k version, but not in the 1k version. Another example is the front left hubcap of the car, which is present in the 10k version but not the 1k version.

Style Loss Comparison:

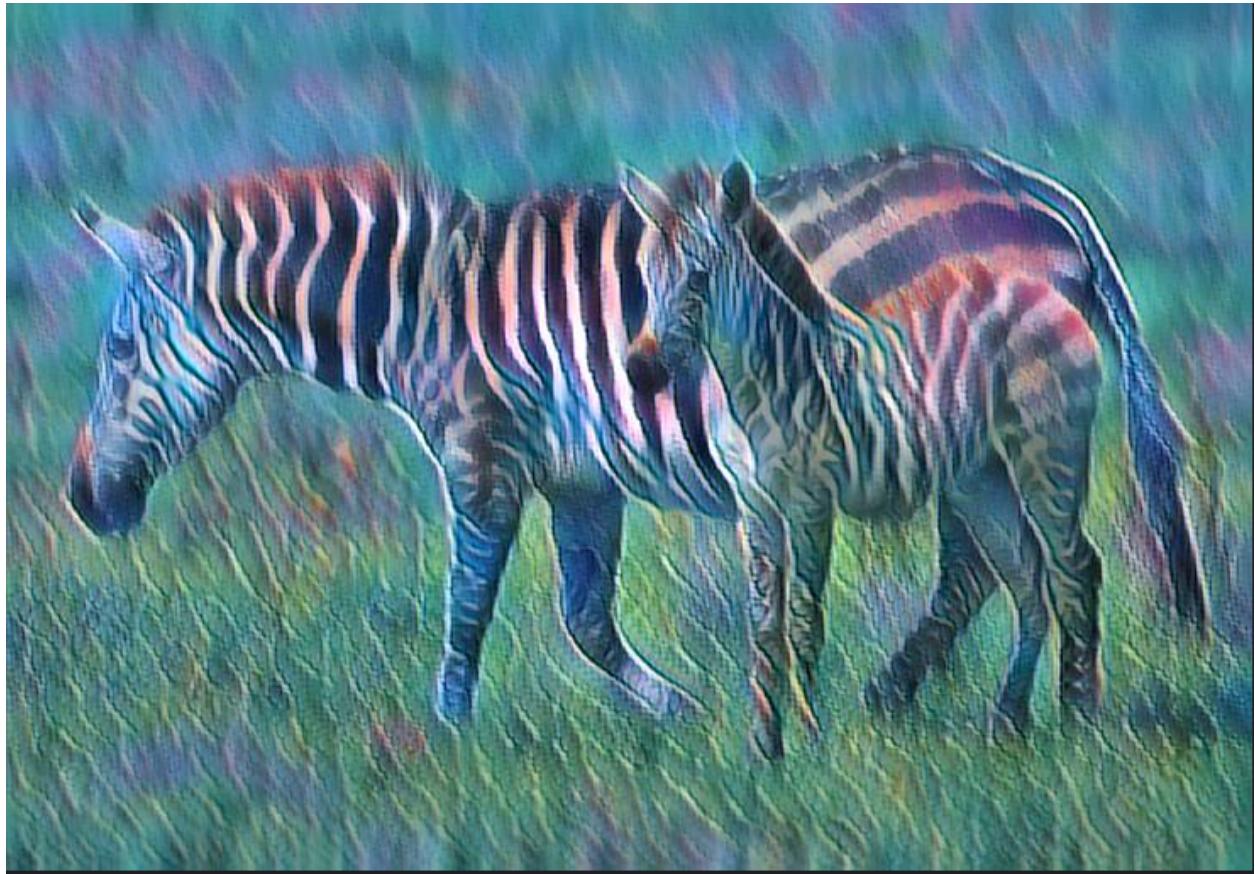
For a side by side comparison of the style performance, I'll use this zebra image with brushstrokes. I think brushstrokes.jpg is a good style image to use for comparison because it is simple and only the style is present in the image with almost no content.



for the 10k model with brushstrokes and zebra image



There is a clear brush texture, but it's more dull than in the 1k version. However, the content features are more intact. The stripes are closer to accurate and the boundaries between the background and foreground are more clear.



Interestingly, the 1k version here has more defined brush stroke texture, but you can see that it lost some of the detail in the zebras. If you look at the shoulders and thighs of the zebras, the stripes begin to go in the wrong direction, and don't follow the correct pattern as well as the 10k result does.

It is difficult to objectively determine which output is better, both look decent but not perfect. I personally think that the 1k style results look better than the 10k results. The defined brush strokes add more to the painting. This is especially strange given that the gamma was lower for the 1k version, meaning there was less emphasis in the style loss vs the content loss when compared to the 10k version. The 10k version also had a lower overall style loss according to the loss plots. It's hard to explain this phenomenon. One possible explanation could be that since the style training loss for the 10k plateaued early, and then kept training, perhaps it has overfit? This could be determined by looking at the validation loss, however this was not recorded for my runs.