I. Experimental Design and Methods

A. Proposed Changes and Rationale

We propose to make three architectural changes to the current design of the network. From a high-level, this consists of a multi-head attention layer, a number of convolutional layers, and a combined MSE and Gaussian Error Loss Function (ERLF). ERLF was chosen because it is used for edge-preserving image smoothing in the field of computer vision [1], and we want to see if this is applicable to this domain.

Each of these changes has the goal of creating smoother predicted images with sharper edges than model 1. The idea is that both our multi-head attention layer and convolution layers will produce spatial information for the already existing MLP to learn with. Along with this, ERLF has been used for edge-preserving image smoothing, and we are interested to know whether using this in combination with MSE will produce more smooth and edge-preserving results.

B. Implementation Details

In our call function for the new model, we will independently produce our encoder embeddings, our multi-head attention features with self-attention on our encoder embeddings, and our convolution features using a number of convolution layers which have an increasing number of filters and a decreasing kernel size. The convolution features are then flattened and tiled over the batch dimension. The attention features and convolution features are then both normalised. This is important as we are concatenating our encoder embeddings, attention features, and convolution features, and we want to ensure that the attention and convolution features do not overshadow our encoder embeddings.

Hyper-parameter changes consist of decreasing n_rays from 2048 to 1024, and increasing the hidden_dim from 32 to 124. We decreased n_rays because 2048 rays was too memory intensive for our introduced layers, and having a perfect-square number of rays allowed us to more easily reshape and convolute our projection image. The idea behind increasing the hidden_dim was to carry more information through our MLP layers. This was important because the structure of our new layers are not feed-forward, but rather each are independently run and their outputs are concatenated to produce the input to our existing MLP. This means that our last dimension for the MLP input is much longer.

Since we are feeding the projection image into the call function, we also needed to update our get_sample_slices to call the call function and pass a projection image. The chosen solution was to take the z-axis value of our current slice, and then find the nearest projection from this. Alternatively, we could have just passed in a flag that fills the convolution features with zeros. This was not further explored due to time constraints, and our chosen method did not seem to have a drastic impact on the results.

ERLF will be used in combination with MSE, with three different lambda values of [0.1, 0.5, 1], where lambda controls the proportion of the loss that each loss function has influence over. In specific, lambda=1 will mean that the loss function is entirely ERLF. Using guidance from [1], we will use a p of 0.25, m of 0.5, and epsilon of 0.001 for ERLF. All models, other than our last three which use ERLF, will have a lambda of 0.

C. Method of Comparison

Our method of comparing architectures will consist of running many different experiments, each with different parts of our architecture, and observing how this affects the MSE, SSIM, PSNR, and our predicted projection of the chest images. We expect that the SSIM and PSNR will increase if we produce images that are smoother with shaper edges, as smoother images with sharper edges should be more similar to our ground-truth.

Specifically, we will test the original model, a model with only convolution layers, a model with only multi-head attention, a model with both convolution layers and multi-head attention, and three models with ERLF and different lambda values. Due to time constraints, we will run experiments for 250 epochs for model 1 and our convolution and attention model, and 100 epochs for every other model.

II. Results

A. Predicted Images

We have plotted our chest_50.pickle data after 10, 50, and 250 epochs, on the second image of the .tiff output files, which can be seen in Fig. 4, placed in appendix A.

B. Graphed Metrics

Each of the models loss, SSIM, and PSNR have been individually graphed, and placed in appendix B. This has been done because we have seven different changes, and graphing them all together makes it difficult to observe individual model results.

Each model has also had its MSE loss, SSIM, and PSNR graphed on a single plot for ease of comparison. We cannot compare MSE loss to our combined MSE and ERLF losses. For this reason, we have removed ERLF models from the MSE loss plots.

Each eligible model's MSE loss can be seen in Fig. 1. Each model's SSIM can be seen in Fig. 2. Each model's PSNR can be seen in Fig. 3.

C. Discussion

Our results clearly show that the experimental ERLF stopped our model from training properly. This can be seen in the SSIM plateauing at around 10% to 20%, the PSNR plateauing at around 12dB. The images produced by the ERLF, no matter the lambda value, are visually far from the ground-truth too. It is unclear whether this is due to a poor implementation of the results found in **??**, or if their results are simply not applicable to this domain.

The rest of our results are much more minor. Over 250 epochs, the convolution and attention model has performed slightly better than model 1 for our MSE loss, SSIM, and

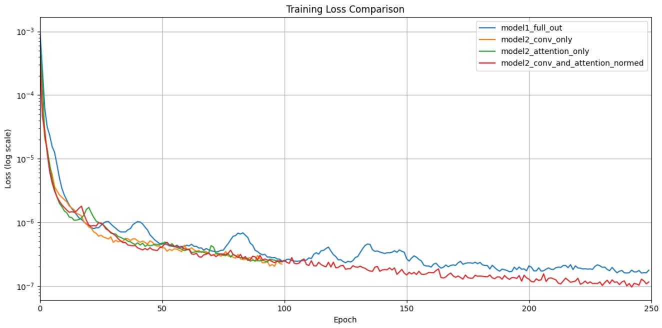

Fig. 1. MSE loss in log scale for models which use the MSE loss function only.

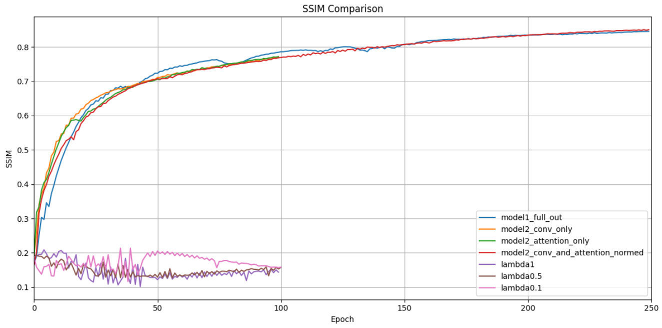

Fig. 2. SSIM for all models.

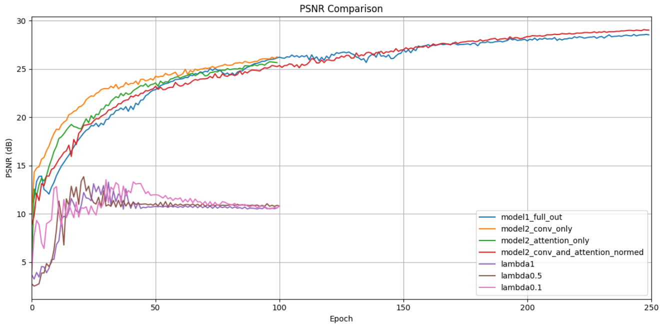

Fig. 3. PSNR for all models.

PSNR. This is a very small difference however, with roughly a 0.5% increase for SSIM and a 0.5dB increase for PSNR. This is practically unnoticeable when observing the predicted images after 250 epochs.

More interestingly, for roughly the first 50 epochs, our convolution only, attention only, and convolution and attention models have performed slightly better than model 1 for SSIM and PSNR. This seem to have quite a noticeable affect when observing the predicted images after 10 epochs. In particular, our convolution only model has produced observationally sharper images after 10 epochs than model 1. This is likely due to the convolution only model having a higher PSNR than our other models for around the first 50 epochs.

The final interesting observation from our results is that our convolution only model, our attention only model, and our convolution and attention model each had a much more stable MSE loss over training than model 1.

III. Conclusion

Our hypothesis was that using convolution layers, a multi-head attention layer, and an ERLF loss function would produce smoother predicted images with sharper edges. Had this been effective, we would have seen a noticeable increase in the SSIM and PSNR. We discussed that our ERLF loss function clearly did not work, however it is still worth discussing the affects of our convolution layers and our multi-head attention layer. Over 250 epochs, we did see a small increase in the SSIM and PSNR of our convolution and attention model, but this increase is too small to conclude that our convolution layers and multi-head attention lead to smoother images with sharper edges.

I am not entirely convinced that the implication of our results is that convolution layers and attention layers do not make a difference in NeRF networks, and it is entirely possible that implementation changes could lead to better results. While wall-clock speed was not measured during training, and it is not practical to compare models by their wall-clock speed, it was clear that the attention and convolution layers added a significant amount time to the training process. This makes me question whether adding convolution layers and multi-head attention layers to our base MLP is a worthwhile trade-off in terms of time and resources to run, compared to our running model 1 for many more epochs.

Interestingly, our convolution and attention layers lead to more stable training, and slightly earlier convergence. It is possible that rather than using additional convolution and attention layers to produce a higher quality result, they could instead be used to produce a result of the same quality in fewer epochs.

There are three main questions which I have from my experiments. What would happen if we further tuned the ERLF parameters lambda, m, p, and epsilon? What would happen if we further tuned both our convolution layers and attention layers? What would happen if we normalised the encoder embeddings such that each of the three concatenated embeddings and features have equal influence over the MLP's training? This last question is particularly interesting, and I hypothesise that not normalising our encoder embeddings, while normalising the convolution features and attention features, meant that our encoder embeddings had much more influence over training the MLP. This might explain why all of our results were so similar to model 1.

References

[1] W. Dong, L. Zeng, S. Ji, and Y. Yang, "Gaussian error loss function for image smoothing," *Image and Vision Computing*, vol. 152, p. 105300, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885624004050

Appendix

Appendix A
Model Predicted Image Matrix

Appendix B
Individual Model Metric Graphs

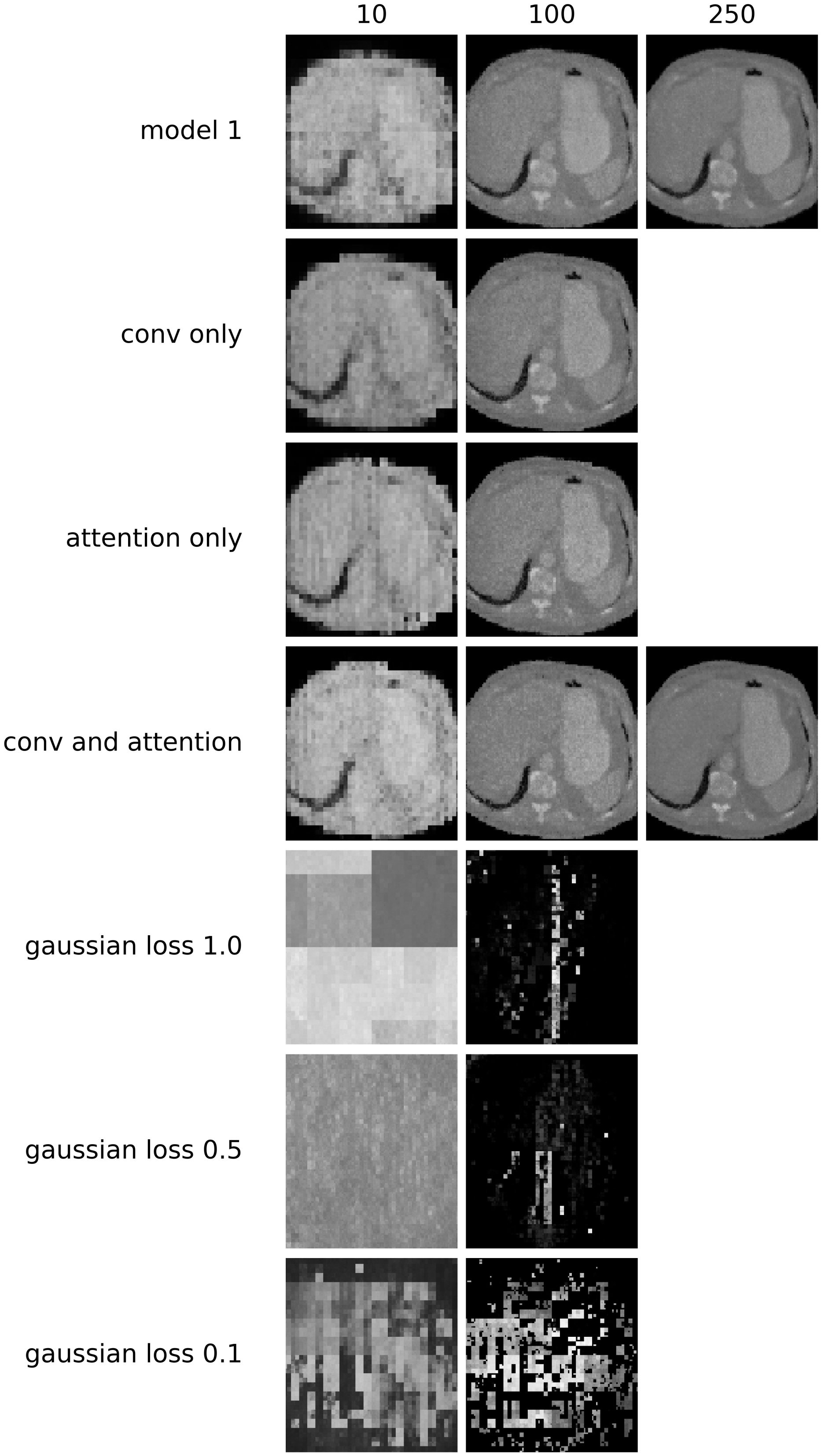

Fig. 4. Grid of second predicted image in the produced .tiff when using the chest_50.pickle data for each model and for 10, 100, and 250 epochs.

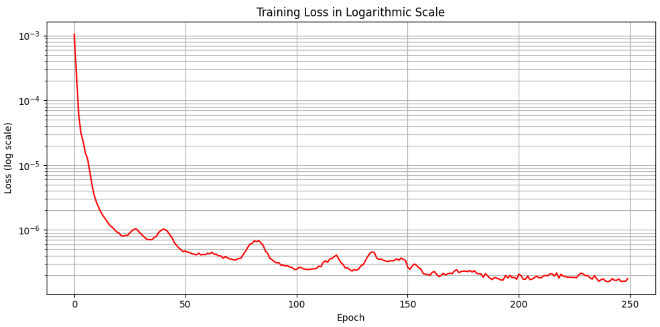

Fig. 5. Model 1's MSE loss on logarithmic scale over 250 epochs

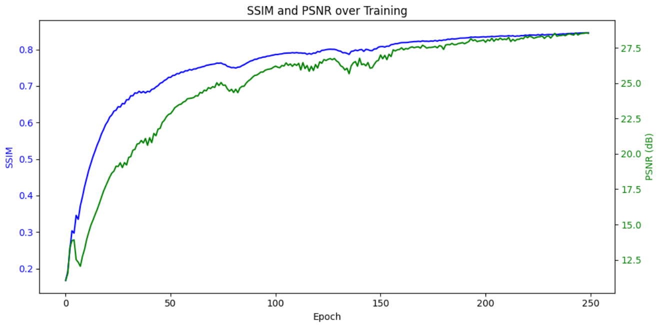

Fig. 6. Model 1's SSIM and PSNR over 250 epochs

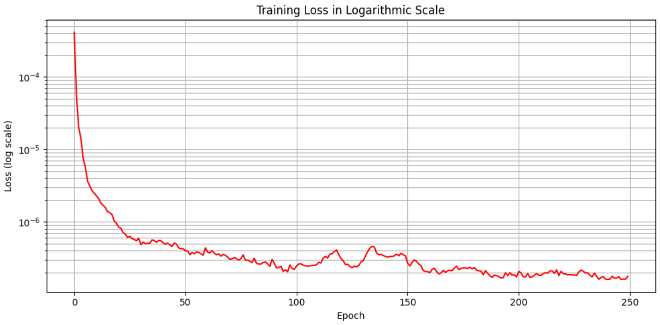

Fig. 7. Convolution only model's MSE loss on logarithmic scale over 100 epochs

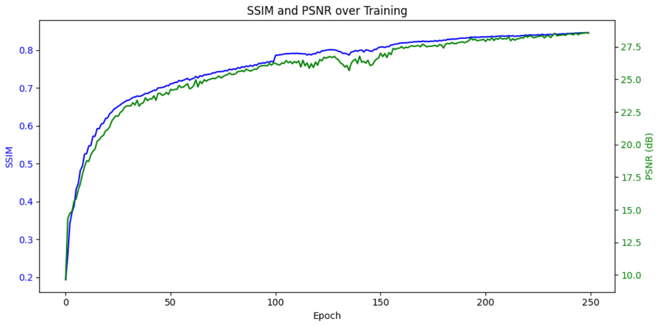

Fig. 8. Convolution only model's SSIM and PSNR over 100 epochs

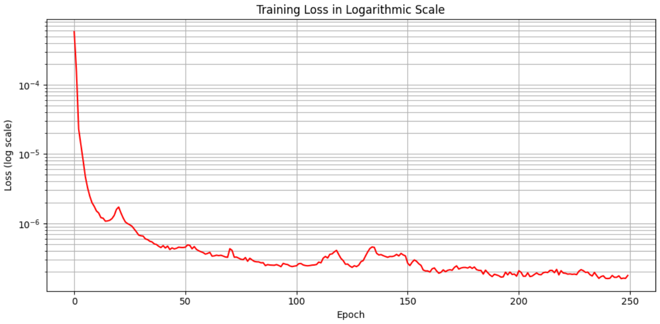

Fig. 9. Attention only model's MSE loss on logarithmic scale over 100 epochs

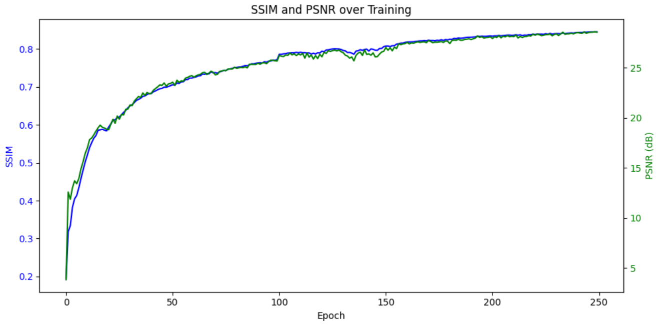

Fig. 10. Attention only model's SSIM and PSNR over 100 epochs

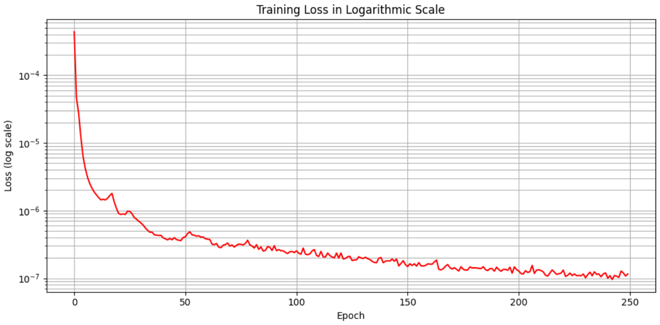

Fig. 11. Convolution and attention model's MSE loss on logarithmic scale over 250 epochs

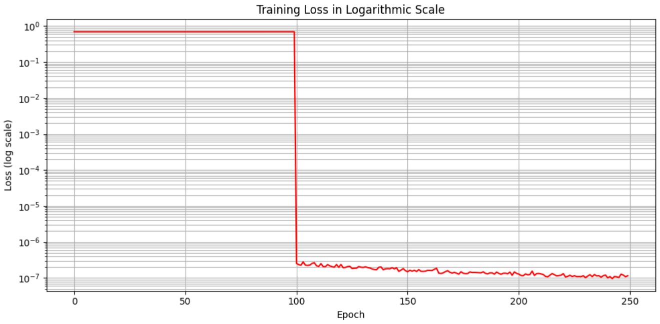

Fig. 12. Convolution and attention and ERLF lambda=1.0 model's loss over 100 epochs

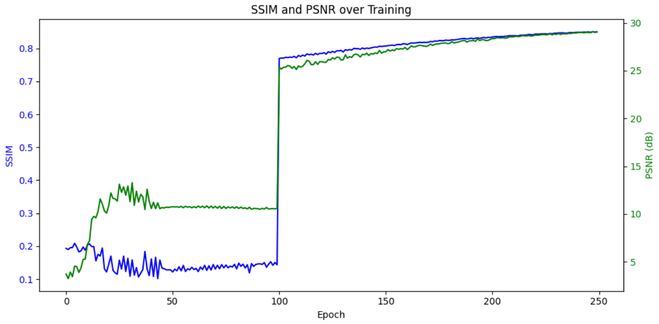

Fig. 13. Convolution and attention and ERLF lambda=1.0 model's SSIM and PSNR over 100 epochs

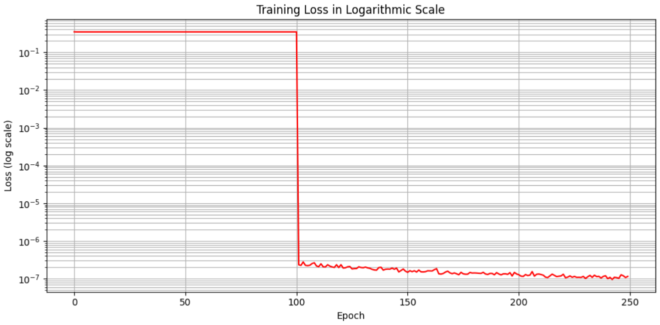

Fig. 14. Convolution and attention and ERLF lambda=0.5 model's loss over 100 epochs

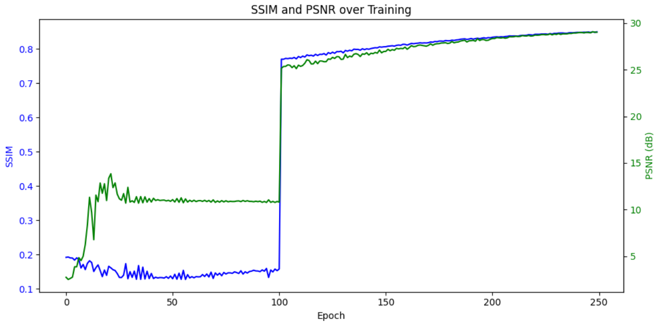

Fig. 15. Convolution and attention and ERLF lambda=0.5 model's SSIM and PSNR over 100 epochs

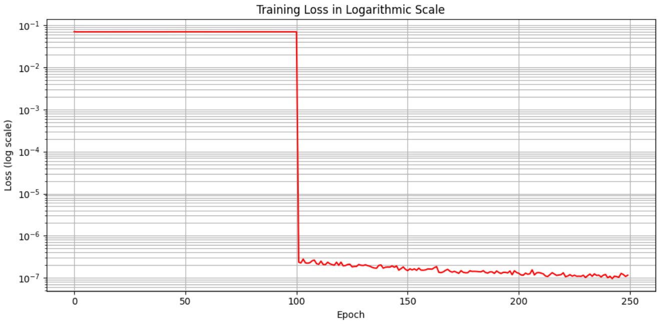

Fig. 16. Convolution and attention and ERLF lambda=1.0 model's loss over 100 epochs

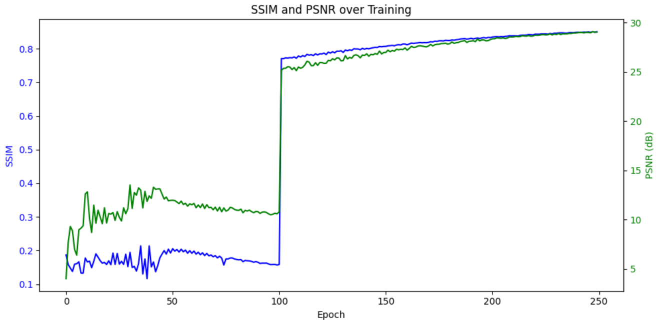

Fig. 17. Convolution and attention and ERLF lambda=1.0 model's SSIM and PSNR over 100 epochs