- I started with exploring grokking on the modular arithmetic tasks. Found that generalization happens when all numbers align their embeddings on a circle in 2D PCA- which makes sense in the context of modular addition.

- When making the same transformer architecture on a classification task for the binary operation between two points predicting the gradient, I found that the embeddings of the points align themselves on the grid when model generalizes. This is an encouraging sign since it can figure out the (x,y) coordinates from the points just by training the model- and makes me believe that it ends up using the classic algorithm to evaluate gradient with (x,y)- and it does generalize well.

- I tried aimlessly to do normal division, but that did not end up working well- transformers with one layer seem incapable of doing this. A deep mind paper discusses this limitation that is then solved by changing the architecture to Arithmetic Logic Unit.

- Then I started working with images and lines- using a ViT. I implemented three different architectures: Every pixel individual token, patch linear embedding then transformer, and transformer patch embedding and then another global transformer. All the transformers learned with high test accuracy for clean data. However, when training with noisy data, only the double transformer architecture could reach high test accuracy. However, when testing its resilience to noise- with test of changing the output with flipping one random pixel in the image- only 36% of images were resilient to noise. On the other hand, only the pixel-token transformer showed translational invariance when loking at 2D PCA- I think the ViT is just not capable of grasping this with the preset patch splitting in the image and, by extension, incapable of discovering RANSAC. Maybe with more layers it could. Also, when trying dimensionality reduction, the MLP layer that combines the attention values of patches was heavily relied on- any reduction to its dimension was detrimental to the overall accuracy. However, reducing any other layer by up to 90% does not damage the high accuracy of around 85.5%.