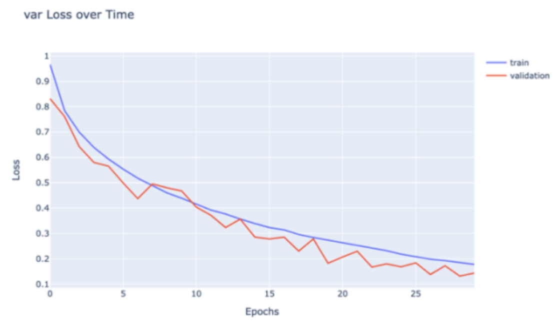
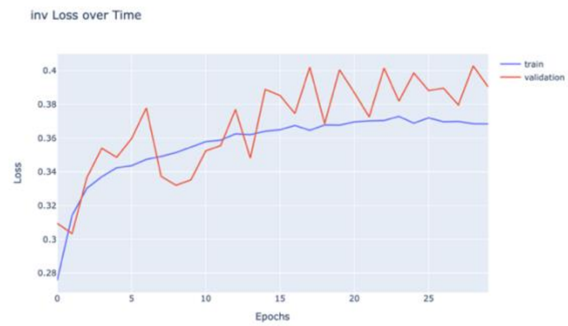
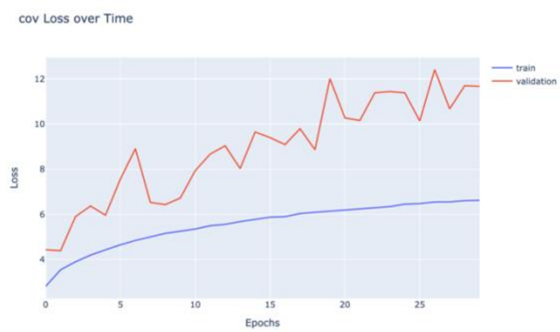
AML -ex2- Matan Cohen – 208346320-

Q1

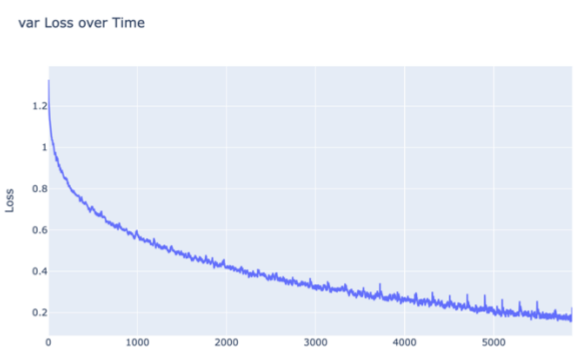
Loss over Epochs:

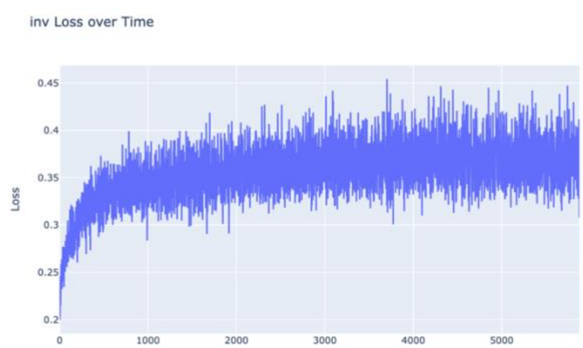


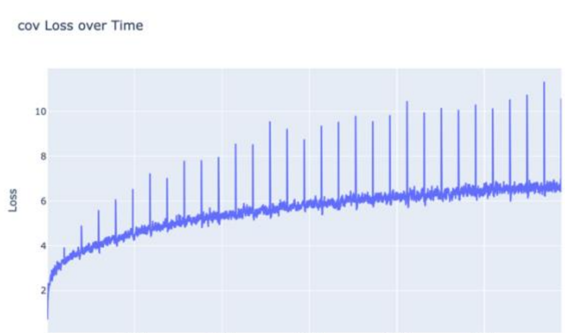




Loss over training samples:



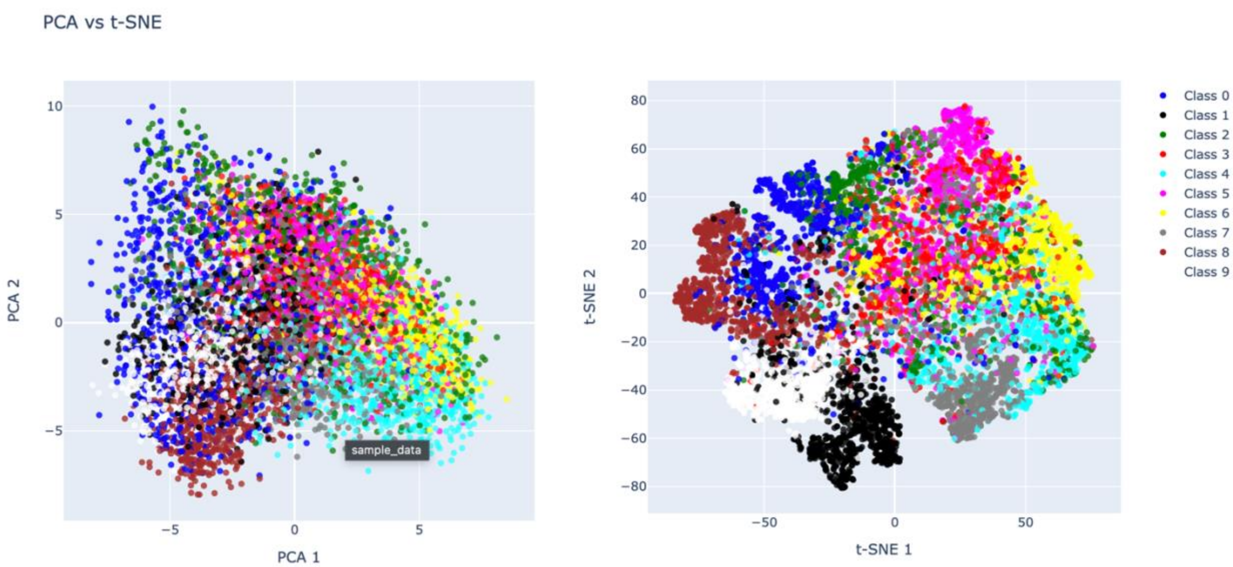




Q2 – Results & Answers

Effectiveness of PCA vs t-SNE:
I got more discernable clusters using SNE. This is certainly due to the non-linear approach of SNE (pairwise distances for local similarities), which might make clusters of similar data points more discernible.
VICReg performance:
If the t-SNE visualization shows that data points (images) from the same class tend to group together in clusters, this suggests that the VICReg has indeed captured class information effectively.
Entangled classes:
Looking at our SNE representation (more informative than PCA) I notice that different classes are mixed in the plot (the clusters are not entirely distinct). The classes are "entangled", which means that the learned representations for these classes are not clearly separable.

I surmise different causes for this. First, it is not because the data classes are separate (Dog! = Ship), that there aren't any similarities between different datapoints from different classes. For example, the sky might be blue in both a dog image and a ship image. These are attributes we hopefully want representation to be invariant to. Therefore, our model and final representation is not strong enough. Second, a limited number of outliers may be more indicative of the dataset then model's performance. (How clustered is the representation a better indicator of our representation strength. I think)



Q3.
When training for 10 epochs on the train data, the accuracy for the linear classifier is : 72.42 %

Q4

The variance objective penalizing cases where the variance of the representations in each dimension falls below a certain threshold. By removing it there is no way to control the variance of different dimensions across the representations in the batch. This allows the possibility of having zero variance essentially causing representations in batch to have the same value for a given dimension. I expect training to converge rapidly as a result since the number of dimensions we optimize upon could potentially decrease. (Optimizing upon the covariance and invariance objectives). I therefore expect a faster convergence and a more compact visualization. I predict the linear classifier to perform poorly: The representations will be harder to classify due to less meaningful dimensions.

- Training for 10 epochs on the train data, the Accuracy for the linear classifier is : 17.58 %

Q5.

Problem using LO:

In VICReg method, the encoder is amortized/shared across all samples in the dataset. By performing Latent Optimization (LO) instead of having a single shared encoder, we optimize a separate latent code for each individual sample.

Other than the increase in complexity, the problem is that VICReg relies on pairs of different augmentations of the same sample to compute its loss. And when using LO, each of these augmentations has its own optimized latent code which aren't in any way reconcilable nor directly related in the optimization process.

Solution using LO:

When optimizing the latent code for each sample, we can compute and optimize upon a latent vector for each augmented point. That is, for each sample, we optimize a latent code such that the decoded versions of this code under multiple augmentations are as close as possible to the corresponding augmented versions of the sample.

The added value of this correction is that it preserves the connection between different views of the same sample, which is essential for VICReg. Moreover, it could potentially lead to more sample-specific and diverse representation since each sample has its own optimized latent code instead of sharing a common encoder with all other samples. This could lead to better performance. Note that we perform a separate optimization process for each sample which increases complexity drastically.

Note: The variance loss in VICReg is computed across batch augmentations. It is not entirely evident that the solution proposed can deal with this especially if batch size if high. What do you think?

Q6:

Test Accuracy of the model on the test images: 34.91 %

I surmise that the accuracy is lower because the L2 metric used to get the nearest neighbor of a representation, is limited. There is nothing in the original learning objectives that forces semantically closed datapoints to have a small l2 distance. Each datapoint is augmented and upon this the l2 loss is minimized. No datapoint is compared to any other throughout the training. Although the classifier can use the resulting representations for effective classification there is nothing in the original laten space that makes similar datapoints representations have a low l2 distance.

Thus, when extracting the nearest neighbor based on l2, the neighbor may be not semantically related to the original datapoint effectively misleading the model. It is the same as introducing unrelated pictures instead of the augmentations in the original training.

Note that in the original training the "generated neighbors" are augmentations that are semantically close to the original datapoint effectively forcing the resulting representation to encode these semantic attributes. If similar pictures have similar attributes, we expect similar pictures to have similar representations. (Seen in t-SNE)

Q7

From the figure it is obvious that the VICReg method is more effective. It manages to capture the class information accurately since we have clusters of classes. The Laplacian eigenmaps representations on the other hand, have no perceivable structure with no separation between different classes.

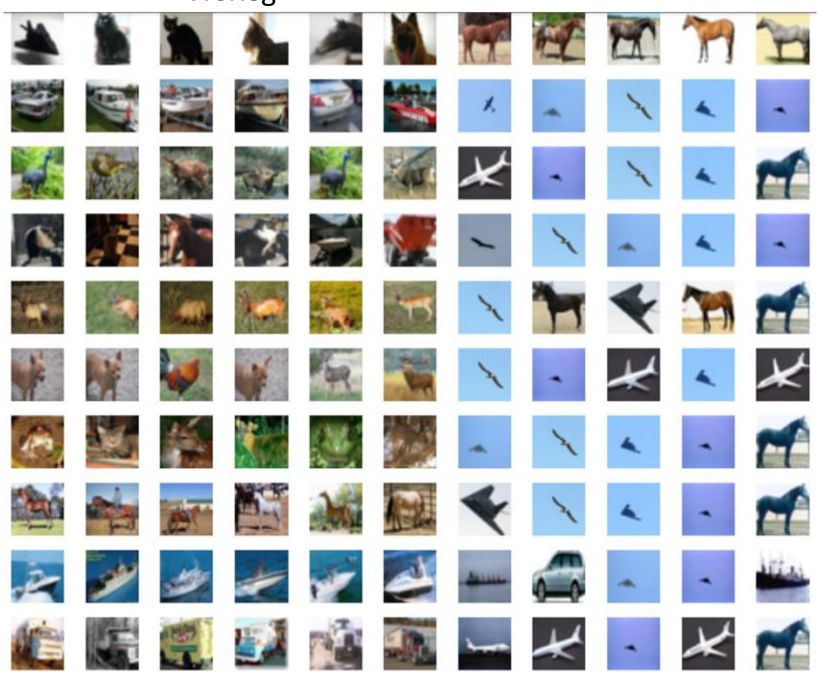
1-The Amortization is crucial for generalization as we have seen.

2-"generated neighbors" in the form of augmentation in our case is also important by replacing the need for NNs as seen before.

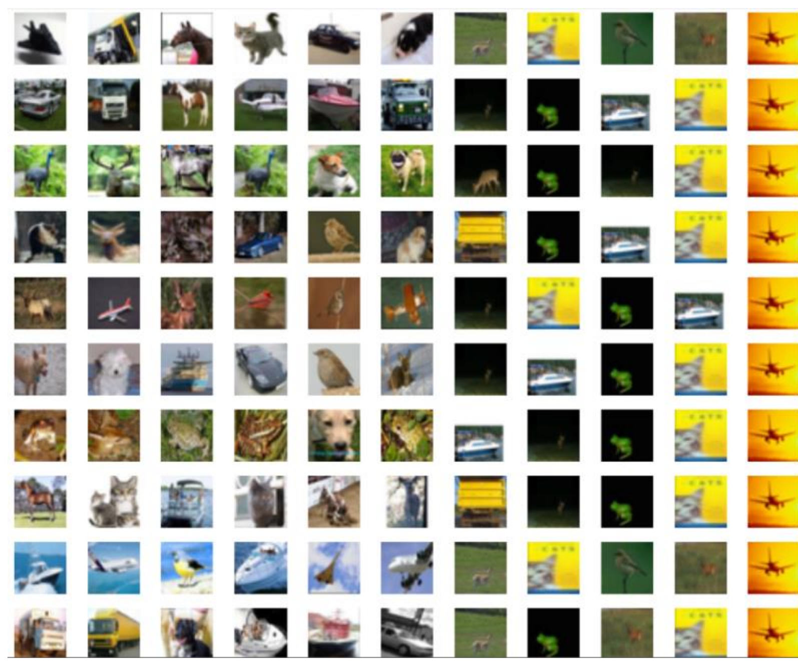
Q8

The figure on the right is of the VICReg method. The one on the left is the No Generated Neighbors (NGN). In each figure, the order is as follows: each row is a different samples image from a different class. The first column is the sample itself. The next 5 pictures are the NN's. The last 5 are the furthest neighbors to the sample.

VICReg



NGN

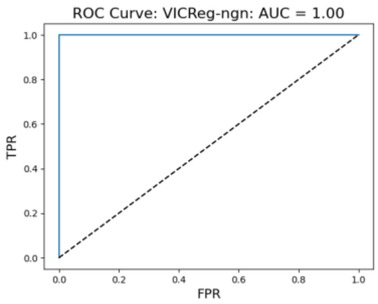


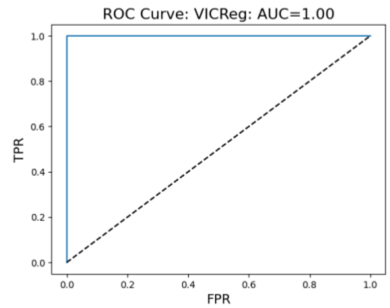
Note on dataset: We can see from above that some of the distant neighbors of the selected samples are very much alike the sample. For example, the dog in (6-vic). Either there are duplicate datapoints or there are datapoints that are very close (such as the result of a photographed scene)

Near Neighbors: Note the NGN algorithm returned neighboring images that share similar attributes such as background (3,9) and size of main object in picture (5). It did generalize well on attributes that cross classes. This is a strength. The best results seem to be from the VICReg method. On the other hand, the VICReg method returned neighboring samples that almost entirely belong to the same class. This doesn't mean boat is a bird lying on a stick in the middle of the sea.

Distant Neighbors:Both methods have set upon "average" samples that are the furthers from all others. The blue birds for VICReg and the yellow plane for NGN. For both are a selected (4-5) samples that are furthest from all. For VICReg (1) seems interesting. All the most distant neighbors come from the same class. However, let us note that there is no discernible pattern (horses appear also in nearest neighbors)

Anomaly detection:





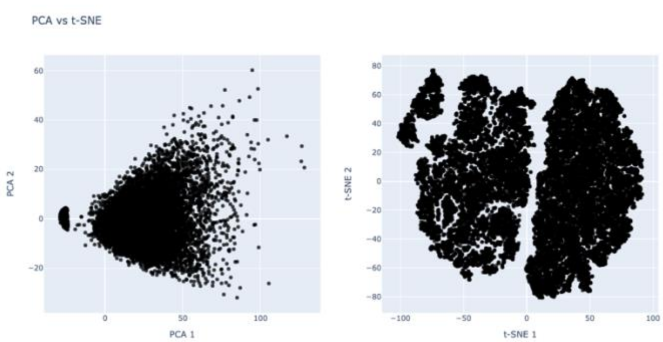
Q1-

Q2- **ROC AUC Evaluation**

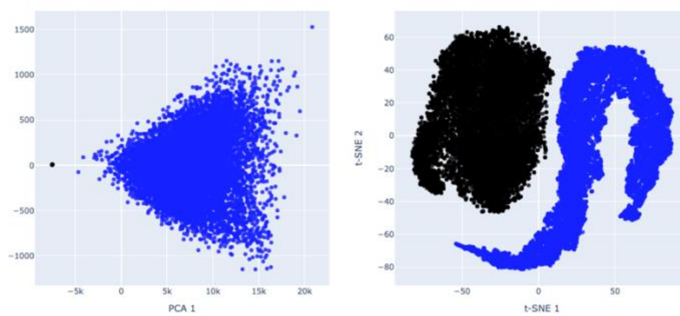
The VICReg method has perfect ROC. The NGN method is also perfect with AUC of 1. It seems that both models where able to distinguish between CIFAR and MNIST datapoints perfectly. That is, detect the MNIS datapoints as anomalies. I would have expected the method without neighbor generation to perform worse with an impartial AUC since the linear probing results (34% acc) and the SNE (no visible class clusters) weren't very encouraging.

Let us plot the SNE visualization of the representations of each method on the test set (both MNIST and CIFAR):

VICReg



VICReg-ngn



-Ignore colors-

These are nice results. For VICReg we can clearly see the separation between the two datasets as well as separation whiting different classes in the CIFAR dataset (I surmise).

For the VICReg-ngn datasets are also completely distinguishable yet we do not see any internal clusters consistent with the low accuracy we got in Q6.

Lastly, an interesting point is the PCA visualization for the VICReg_ngn which mapped the entire MNIST dataset to one single point. (The black background may be a good feature)

Q3 - **Qualitative Evaluation Ambiguity of Anomaly Detection**
Since both models were able to detect the anomalies perfectly, we get that the most anomalous samples for each are very similar. All from the MNIST dataset. One model seems to have relied on thinner “number images” with a recurring class (8). The other relied on thicker numbers with a recurrig class (1). Note all the datapoints for the MNIST dataset have a black background.