OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations

Pramuditha Perera (Johns Hopkins University)
Ramesh Nallapati, Bing Xiang (AWS AI)

CVPR2019

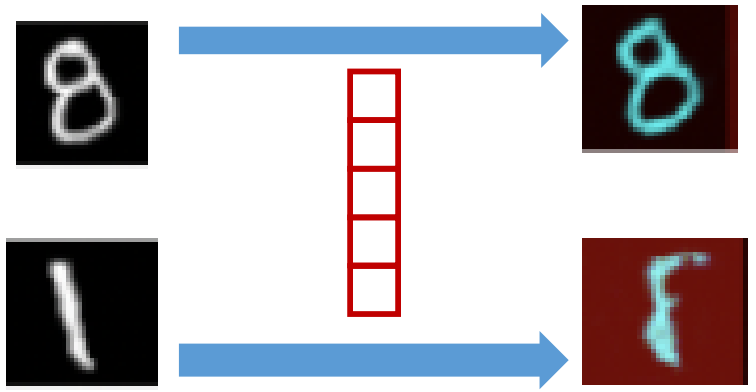
1 Introduction

One-class novelty detection: quantifying the probability that a test example belongs to the distribution defined by training examples.

Learn a representative **latent space** for the given class.
Infer based on the projection of a query image onto the learned latent space.

Compare query image and its inverse image (reconstruction)

Assumption:
when an out-of-class object is presented to the network, it will do a poor job of describing the object, thereby reporting a relatively higher reconstruction error.



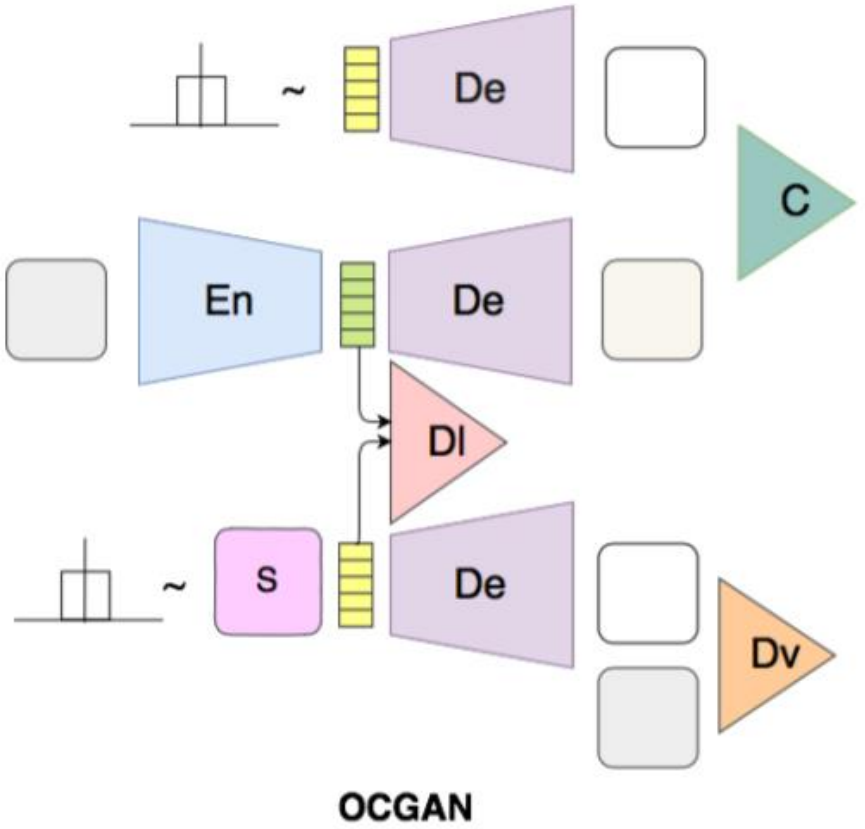
1 Introduction

- Auto-encoders trained on digits with a simple shape such as 0 and 1 have high novelty detection accuracy.
- In contrast, digits with complex shapes, such as digit 8, have relatively weaker novelty detection accuracy.
- A latent space learned for a class with complex shapes inherently learns to represent some of out-of-class objects as well.

Proposed OCGAN:
- in-class samples are well represented
- out-of-class samples are poorly represented

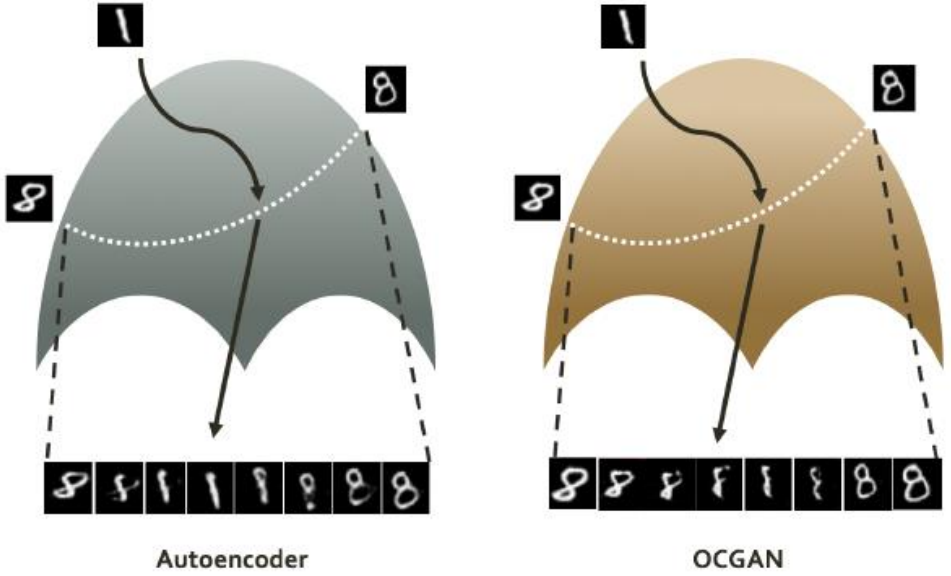
Top: Input Image
Middle: Output of an AE trained on digit 8
Bottom: Output of proposed method

2 Proposed Method



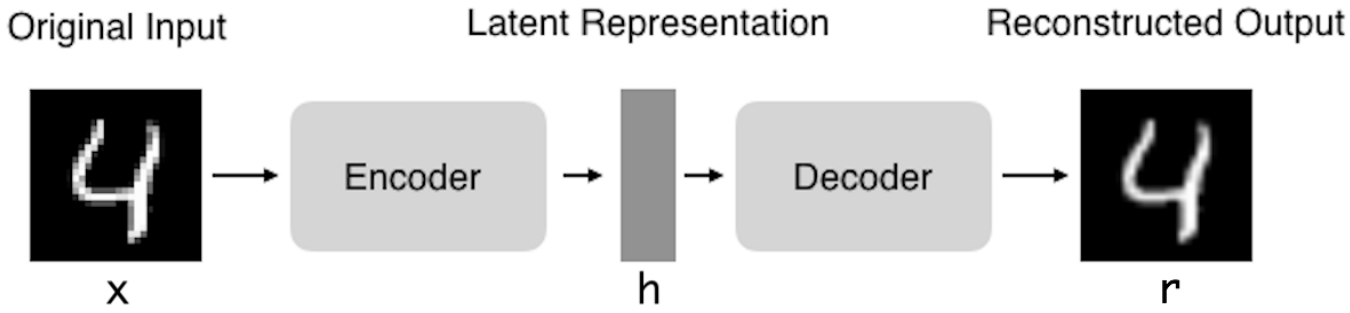
OCGAN

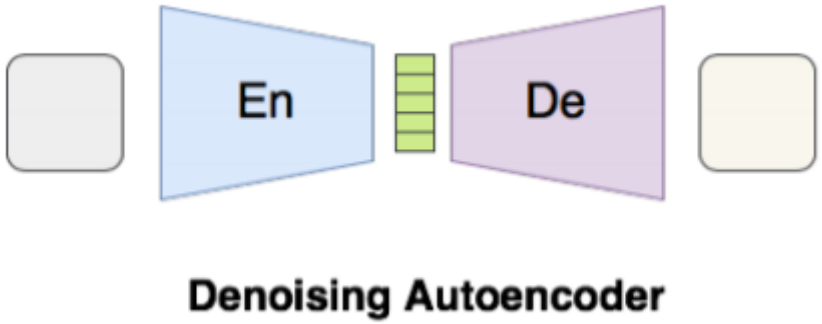
2-1 Motivation



Force the entirety of the latent space to represent **only** the **given class!**

2-2 Denoising auto-encoder



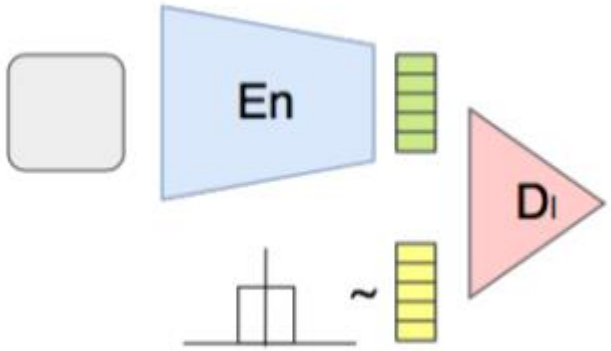


$$l_{\text{MSE}} = ||x - \text{De}(\text{En}(x+n))||_2^2$$

Add noise to input image

Output denoised image

2-2 Latent Discriminator



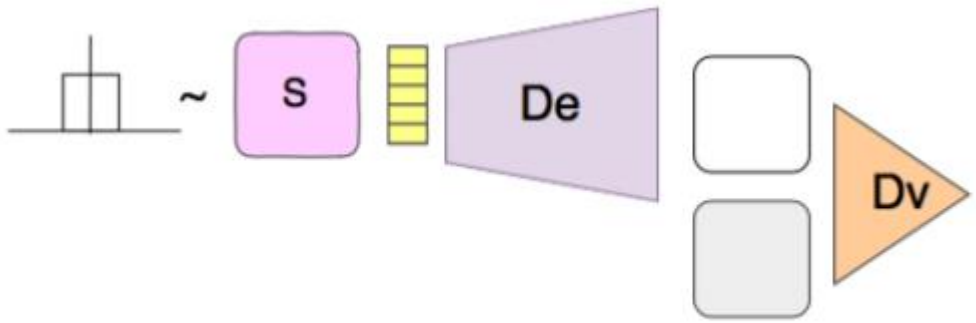
Force latent representations of in-class examples to be distributed uniformly across the latent space.

The latent discriminator is trained to differentiate between latent representations of real images of the given class and samples drawn from a $U(-1,1)^d$ distribution.

$$l_{\text{latent}} = -(\mathbb{E}_{s\sim\mathbb{U}(-1,1)}[\log D_l(s)] + \mathbb{E}_{x\sim p_x}[\log(1 - D_l(\text{En}(x+n)))])$$

$$\max_{\text{En}} \min_{D_l} l_{\text{latent}}$$

2-2 Visual Discriminator



Visual Discriminator

sample exhaustively from the latent space and ensure corresponding images are not from out-of-class

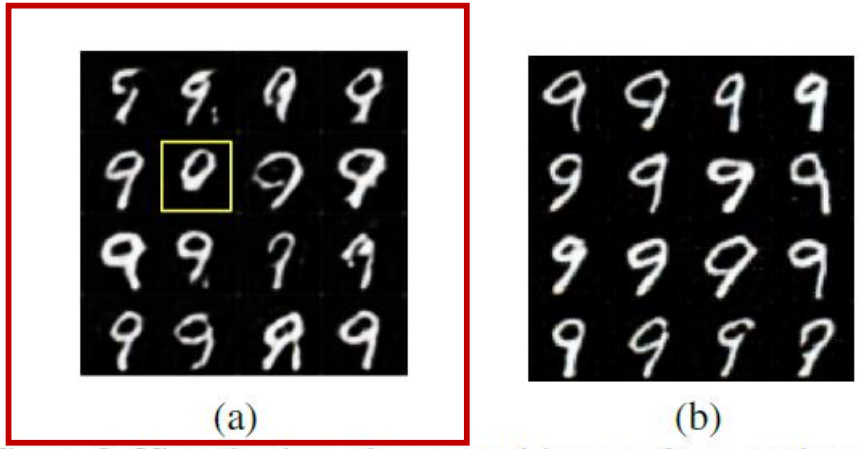
all images generated from latent samples are from the same image space distribution as the given class

$$l_{\text{visual}} = -(\mathbb{E}_{s\sim\mathbb{U}(-1,1)}[\log D_v(\text{De}(s))] + \mathbb{E}_{x\sim p_l}[\log(1 - D_v(x))])$$

Visual discriminator is trained to differentiate between images of the given class and images generated from random latent samples.

$$\max_{\text{De}} \min_{D_v} l_{\text{visual}}$$

2-2 Informative-negative Mining



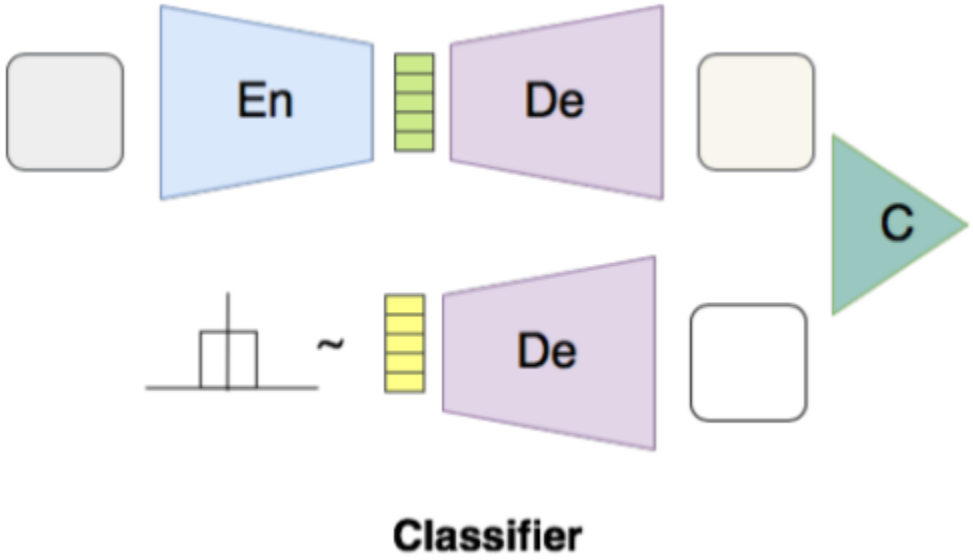
(a) (b)

Figure 3. Visualization of generated images from random latent samples when the network is trained (a) without informative-negative mining (b) with informative-negative mining, for digit 9. In the former case, obtained digits are of a different shape in certain instances. For example, the highlighted generated-image looks like a 0. In the latter case, all generated digits consistently look like a 9.

There are latent space regions that do not produce images of the given class. This is because sampling from all regions in the latent space is impossible during training.

Actively seek regions in the latent space that produce images of poor quality.

2-2 Informative-negative Mining



Classifier determines how well the given image resembles content of the given class

Positive: in-class samples
Negative: fake images

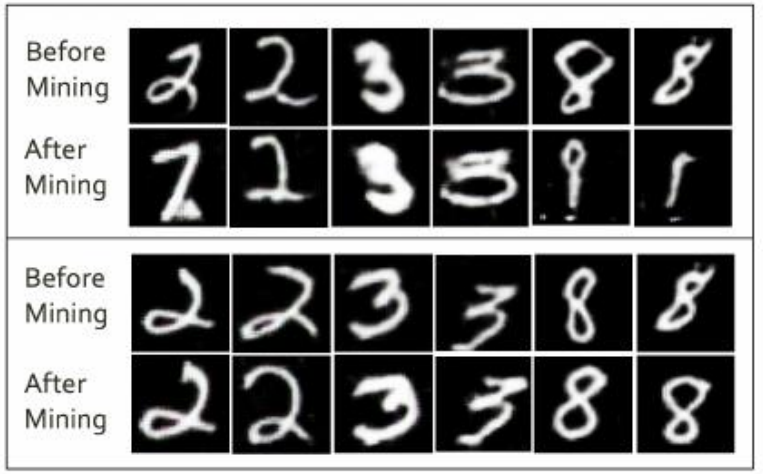
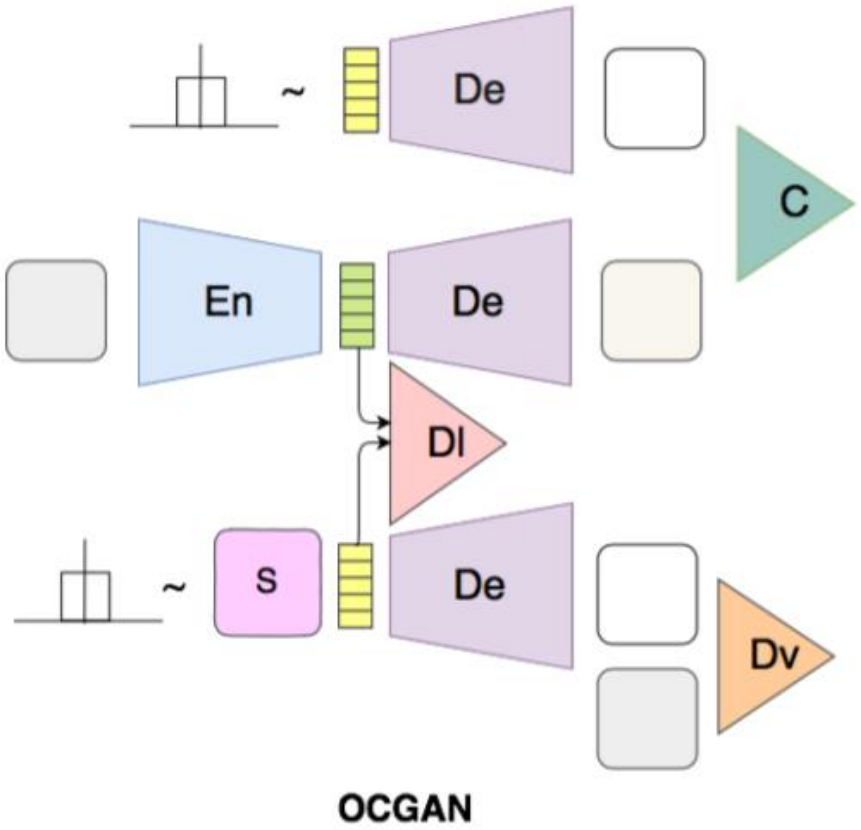


Figure 4. Informative-negative mining. Shown in the image are image pairs before and after mining process for different digits. In the top row, original images are subjected to substantial changes where they have been converted into a different digits altogether. These are the informative-negatives we are looking for. In the bottom row, the change is not substantial, which means the samples we mined are not informative. However, it still does not hurt our training process.

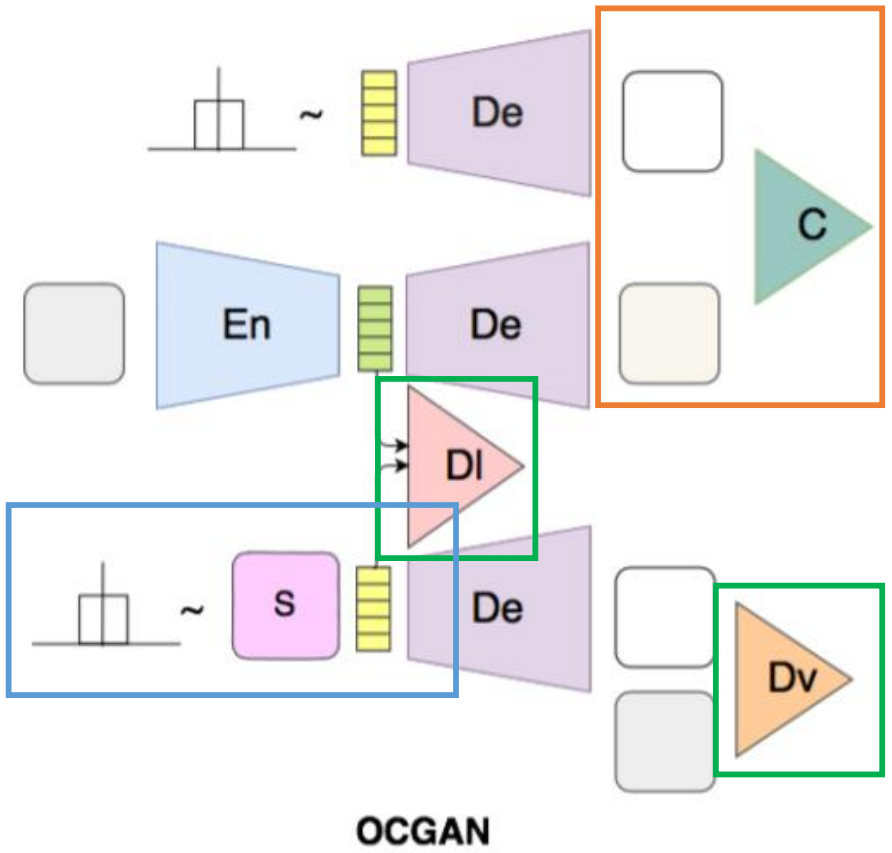
2-2 Network



OCGAN

- Images of known class
- Reconstructed image using known class image
- Fake Image
- Latent vector of known class image
- Sample of latent distribution
- Visual Discriminator
- Latent Discriminator
- Decoder
- Encoder
- Classifier
- Sampler

Proposed Algorithm



OCGAN

Input : Set of training data x, iteration size N, parameter λ
Output: Models: En, De, C, D_l , D_v

for *iteration 1* **to** $\rightarrow N$ **do**

Classifier update: keep D_l, D_v, En, De fixed.
$n \longleftarrow \mathcal{N}(0, I)$
$l_1 \longleftarrow En(x + n)$
$l_2 \longleftarrow \mathbb{U}(-1, 1)$
$l_{classifier} \longleftarrow C(De(l_2), 0) + C(De(l_1), 1)$
Back-propagate $l_{classifier}$ to change C

Discriminator update:
$l_{latent} \longleftarrow D_l(l_1, 0) + D_l(l_2, 1)$
$l_{visual} \longleftarrow D_v(De(l_2), 0) + D_v(x, 1)$
Back-propagate $l_{latent} + l_{visual}$ and change D_l, D_v

Informative-negative mining : Keep all networks fixed.
for *sub-iteration 1* **to** $\rightarrow 5$ **do**
  $l_{classifier} \longleftarrow C(De(l_2), 1)$
  Back-propagate $l_{classifier}$ to change l_2
end

Generator update: keep D_l, D_v,C fixed.
$l_{latent} \longleftarrow D_l(l_1, 1) + D_l(l_2, 0)$
$l_{visual} \longleftarrow D_v(De(l_2), 1) + D_v(x, 0)$
$l_{mse} \longleftarrow ||x - De(l_1)||^2$
Back-propagate $l_{latent} + l_{visual} + \lambda l_{mse}$ to change En, De

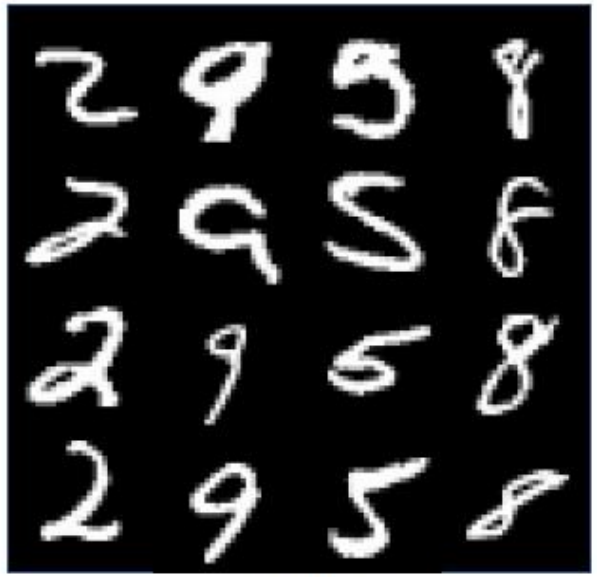
end

3 Experiments

Datasets

COIL

FMNIST



MNIST

3 Experiments

Table 1. Mean One-class novelty detection using Protocol 1.

	MNIST	COIL	fMNIST
ALOCC DR [20]	0.88	0.809	0.753
ALOCC D [20]	0.82	0.686	0.601
DCAE [21]	0.899	0.949	0.908
GPND [16]	0.932	0.968	0.901
OCGAN	**0.977**	**0.995**	**0.924**

3 Experiments

Table 2. One-class novelty detection results for MNIST dataset using Protocol 2.

	0	1	2	3	4	5	6	7	8	9	MEAN
OCSVM [24]	0.988	**0.999**	0.902	0.950	0.955	0.968	0.978	0.965	0.853	0.955	0.9513
KDE [2]	0.885	0.996	0.710	0.693	0.844	0.776	0.861	0.884	0.669	0.825	0.8143
DAE [4]	0.894	**0.999**	0.792	0.851	0.888	0.819	0.944	0.922	0.740	0.917	0.8766
VAE [6]	0.997	**0.999**	0.936	0.959	0.973	0.964	0.993	0.976	0.923	0.976	0.9696
Pix CNN [26]	0.531	0.995	0.476	0.517	0.739	0.542	0.592	0.789	0.340	0.662	0.6183
GAN [23]	0.926	0.995	0.805	0.818	0.823	0.803	0.890	0.898	0.817	0.887	0.8662
AND [1]	0.984	0.995	**0.947**	0.952	0.960	0.971	**0.991**	0.970	0.922	0.979	0.9671
AnoGAN [23]	0.966	0.992	0.850	0.887	0.894	0.883	0.947	0.935	0.849	0.924	0.9127
DSVDD [19]	0.980	0.997	0.917	0.919	0.949	0.885	0.983	0.946	**0.939**	0.965	0.9480
OCGAN	**0.998**	**0.999**	0.942	**0.963**	**0.975**	**0.980**	**0.991**	**0.981**	**0.939**	**0.981**	**0.9750**

4 Conclusion

- First we restricted the latent space to be bounded and forced latent projections of in-class population to be distributed evenly in the latent space using a latent discriminator.
- Then, we sampled from the latent space and ensured using a visual discriminator that any random latent sample generates an image from the same class.
- Finally, in an attempt to reduce false positives we introduced an informative-negative mining procedure.

Dataset

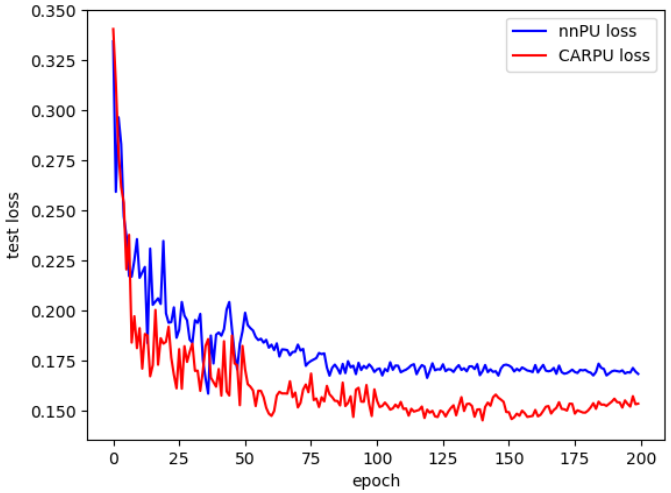
- Cifar-10
  - P: ‘airplane’, ‘automobile’, ‘ship’ and ‘truck’
  - N: ‘bird’, ‘cat’, ‘deer’, ‘dog’, ‘frog’ and ‘horse’
  - Network: (32*32*3)-[C(3*3,96)]*2-C(3*3,96,2)-[C(3*3,192)]*2-C(3*3,192,2)-C(3*3,192)-C(1*1,192)-C(1*1,10)-1000-1000-1

- MNIST
  - P: 0, 2, 4, 6, 8
  - N: 1, 3, 5, 7, 9
  - Network: 784-300-300-300-300-1

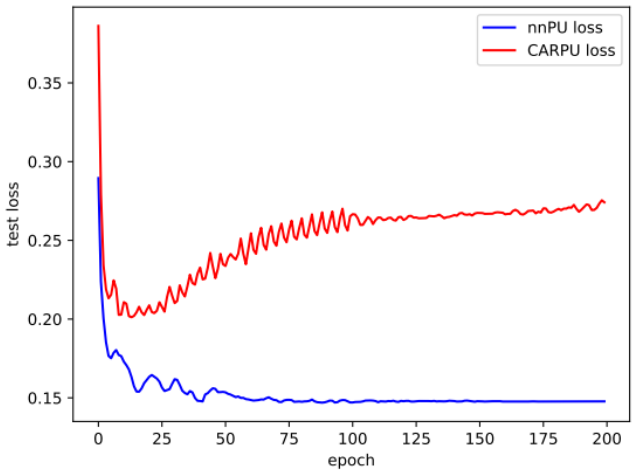
- 20NewsGroup
  - P: ‘alt.’, ‘comp.’, ‘misc.’ and ‘rec.’
  - N: ‘sci.’, ‘soc.’ and ‘talk.’
  - Network: d-avg_pool(word_emb(d,300))-300-300-1

nnPU + rank

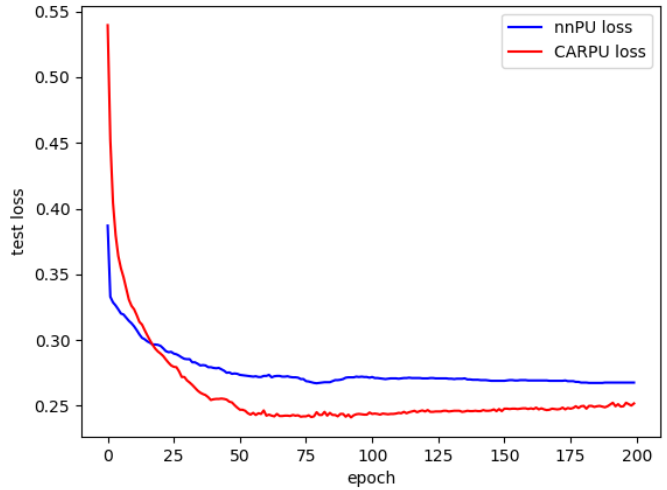
cifar-10



mnist



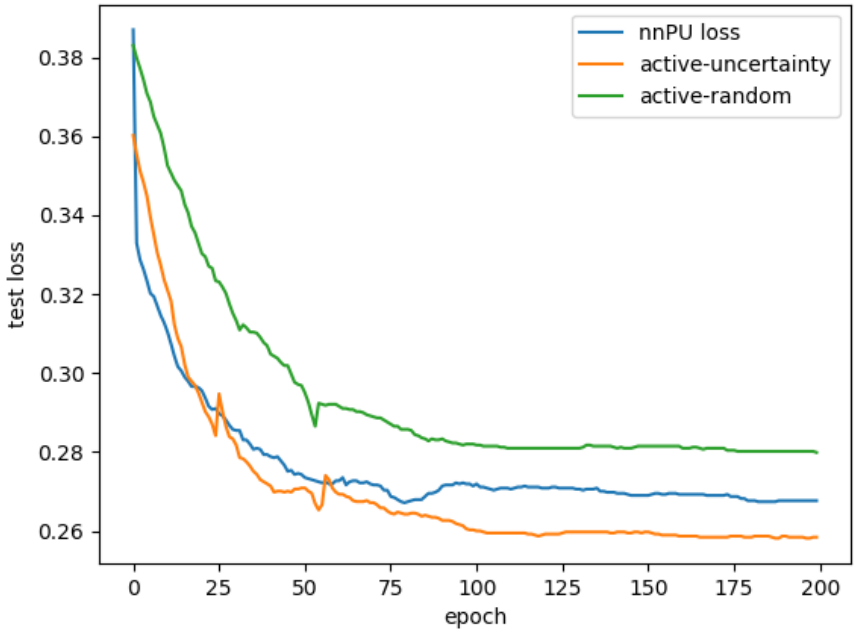
20news



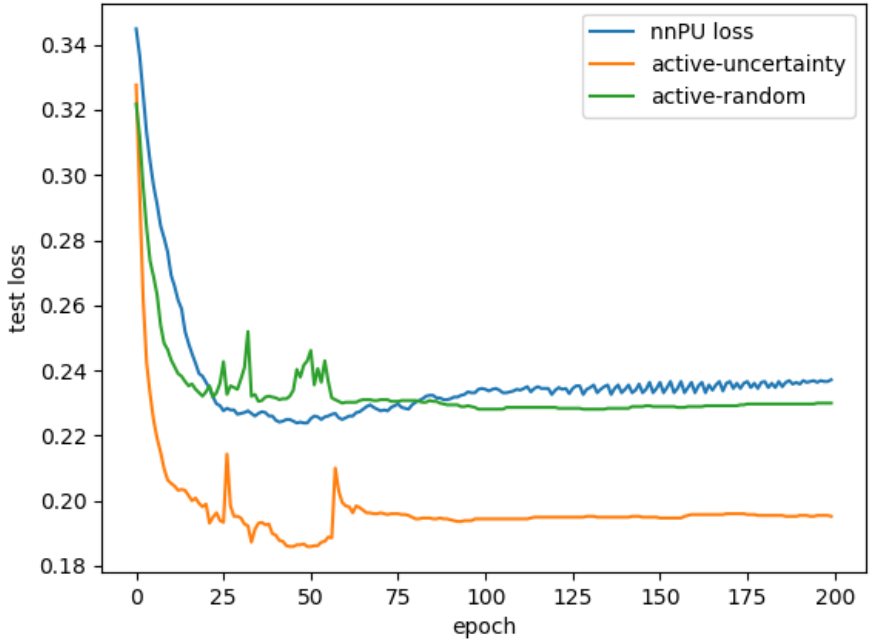
Query

Active learning: add sample and train model till convergence
aaPU: add sample after every epoch

Query strategy: uncertainty & random
Batch size: 20
Dataset: 20NewsGroup



learning_rate=0.0001



learning_rate=0.0002