SUPPLEMENTARY MATERIAL FOR: DISCRIMINATIVE AUTO-ENCODING FOR MULTICLASS CLASSIFICATION AND REPRESENTATION LEARNING PROBLEMS

Vipul Bansal, Himanshu Buckchash and Balasubramanian Raman

Machine Vision Lab, Department of Computer Science and Engineering
Indian Institute of Technology, Roorkee, India

ABSTRACT

In this supplementary material we further explain the process of adding noise to DiscAE. We explain the vectorized representation for noise treatment. The relation of DiscAE is discussed with adversarial auto-encoders (AAE). Later we have presented some visual results.

1. EFFECT OF NOISE TREATMENT

It was observed that cluster loss brings a slight improvement in results for discriminative nature but it requires an extensive training to learn better features. To improve the performance, another way is noise treatment.

The output from the encoding distribution $E(\mathcal{Z}|\mathcal{X})$ is impacted by a sampled noise η given as:

$$\eta \sim \mathcal{N}(\mu, \sigma) \tag{1}$$

$$p\left(\mathcal{Z}_i/\mathcal{Y}_j\right) \sim \sum_{k=1}^{\alpha} \mathcal{W}_k \times \mathcal{N}\left(\mu_k, \sigma_k\right) \forall \alpha \in N \tag{2}$$

The noise is sampled from a Gaussian distribution with a mean μ and variance σ. The reason for sampling from the Gaussian distribution is that the $p(\mathcal{Z}_i/\mathcal{Y}_j)$ as mentioned in (2) is a Gaussian mixture model, preferring sampling noise η from $\mathcal{N}(\mu, \sigma)$. The sampled noise is then added to the latent space $\mathcal{Z}$ as given in the equation below:

$$\mathcal{Z} = \mathcal{Z} + \epsilon \times \eta \tag{3}$$

In the above equation ϵ is a random constant coefficient. The impact this has on the distributions $E(\mathcal{Z}|\mathcal{X})$, $D_C(\mathcal{Y}|\mathcal{Z})$, and $D_G(\mathcal{X}|\mathcal{Z})$ can be understood as it makes our assembly more robust towards slight changes in $\mathcal{Z}$ and $p(\mathcal{Z})$. It can be understood from the equations given below:

$$p_r(\hat{\mathcal{X}}) = \int_{\mathcal{Z}} \int_{\eta} D_G(\mathcal{X}|\mathcal{Z}, \eta) p_g(\mathcal{Z}) p_\eta(\eta) d\eta d\mathcal{Z} \tag{4}$$

$$p_c(\hat{\mathcal{Y}}) = \int_{\mathcal{Z}} \int_{\eta} D_C(\mathcal{Y}|\mathcal{Z}, \eta) p_g(\mathcal{Z}) p_\eta(\eta) d\eta d\mathcal{Z} \tag{5}$$

In the above equation $p_\eta(\eta)$ stands for the probability distribution of the added noise. This addition of noise transforms $D_G(\mathcal{X}|\mathcal{Z})$ to $D_G(\mathcal{X}|\mathcal{Z}, \eta)$ and similarly $D_C(\mathcal{Y}|\mathcal{Z})$ to $D_C(\mathcal{Y}|\mathcal{Z}, \eta)$ which implies that the decoding and classifier distributions are adjusted with noise to give better results for both discriminatory and reconstructive nature of encoder.

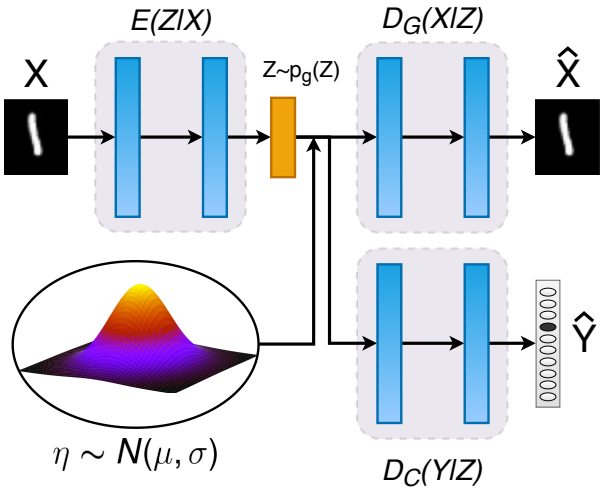


Fig. 1. Addition of noise in the proposed approach (DiscAE).

$$p_c(\hat{\mathcal{Y}}) = \int_{\mathcal{Z}} D_C(\mathcal{Y}|\mathcal{Z}) p_g(\mathcal{Z}) d\mathcal{Z} \tag{6}$$

$$p_r(\hat{\mathcal{X}}) = \int_{\mathcal{Z}} D_G(\mathcal{X}|\mathcal{Z}) p_g(\mathcal{Z}) d\mathcal{Z} \tag{7}$$

$$D_G(\mathcal{X}|\mathcal{Z}) = \int_{\eta} D_G(\mathcal{X}|\mathcal{Z}, \eta) p_\eta(\eta) d\eta \tag{8}$$

$$D_C(\mathcal{Y}|\mathcal{Z}) = \int_{\eta} D_C(\mathcal{Y}|\mathcal{Z}, \eta) p_\eta(\eta) d\eta \tag{9}$$

The above equations show how the noise is adjusted in $D_G(\mathcal{X}|\mathcal{Z}, \eta)$ and $D_C(\mathcal{Y}|\mathcal{Z}, \eta)$ to give back the original distribution $D_G(\mathcal{X}|\mathcal{Z})$ and $D_C(\mathcal{Y}|\mathcal{Z})$ and use them for reconstruction and classification as shown in (6), (7).

The addition of noise to the DiscAE model is shown in Fig. 1. The network is trained by passing the dataset from the encoding distribution $E(\mathcal{Z}|\mathcal{X})$ followed by addition of noise η and passing it simultaneously through $D_G(\mathcal{X}|\mathcal{Z})$ and $D_C(\mathcal{Y}|\mathcal{Z})$ followed by calculation of MSE and BCE/NLL

losses. The calculated loss is back propagated simultaneously over the graph $\mathcal{G}(\theta)$; for this, auto-grad feature of PyTorch was used for our experiments. The need for both reconstruction and classification losses is because both $p_r(\hat{\mathcal{X}})$ and $p_c(\hat{\mathcal{Y}})$ are required for understanding the reconstructive and discriminatory nature of the latent space $\mathcal{Z}$ upon addition of noise η.

1.1. Vectorized Hypothesis for Noise Treatment

Another possible way to explain why noise treatment is effective is by a vectorized Hypothesis $\mathcal{H}$. Let $\vec{\mathcal{Z}}$ be the latent space vector generated from $E(\vec{\mathcal{Z}}|\vec{\mathcal{X}})$ for a input $\vec{\mathcal{X}}$.

$$p_g(\mathcal{Z}) = \int_{\mathcal{X}} E(\mathcal{Z}|\mathcal{X}) p_i(\mathcal{X}) d\mathcal{X} \tag{10}$$

$$p_g(\vec{\mathcal{Z}}) = \int_{\vec{\mathcal{X}}} E(\vec{\mathcal{Z}}|\vec{\mathcal{X}}) p_i(\vec{\mathcal{X}}) d\vec{\mathcal{X}} \tag{11}$$

The above equation shows the vectorized format of (10). Let us consider an input data point represented by vector $\vec{\mathcal{X}}_i$ giving a latent space vector $\vec{\mathcal{Z}}_i^j$ on a given forward pass.

The vector $\vec{\mathcal{Z}}_i^j$ has to change its movement in an N-dimensional space $\mathcal{S}$ using a Gaussian probability $g_\theta \sim \mathcal{N}(\mu, \sigma)$, changing the position to $\vec{\mathcal{Z}}_i^{j'}$ as given in the equation below:

$$\vec{\mathcal{Z}}_i^{j'} = \vec{\mathcal{Z}}_i^j + \epsilon \times \vec{g}_\theta \tag{12}$$

Let $\hat{d}$ be the directional change between the initial position $\vec{\mathcal{Z}}_i^j$ and final position $\vec{\mathcal{Z}}_i^{j'}$. This positional change $\hat{d}$ has an impact on both $p_r(\hat{\mathcal{X}})$ and $p_c(\hat{\mathcal{Y}})$. $\hat{d}$ can be given by the equation:

$$\hat{d} = \frac{\vec{\mathcal{Z}}_i^{j'} - \vec{\mathcal{Z}}_i^j}{|\vec{\mathcal{Z}}_i^{j'} - \vec{\mathcal{Z}}_i^j|} \tag{13}$$

On back propagation, changes can be seen in the $E(\vec{\mathcal{Z}}|\vec{\mathcal{X}})$ distribution such that on next forward pass the position of point $\vec{\mathcal{Z}}_i^j$ changes to a new position say, $\vec{\mathcal{Z}}_i^{j+1}$. Let $\hat{d}_c$ be the new change vector given as:

$$\hat{d}_c = \frac{\vec{\mathcal{Z}}_i^{j+1} - \vec{\mathcal{Z}}_i^j}{|\vec{\mathcal{Z}}_i^{j+1} - \vec{\mathcal{Z}}_i^j|} \tag{14}$$

According to our hypothesis $\mathcal{H}$, the added noise $\vec{g}_\theta$ proves to be beneficial in improving the discriminatory and generative property of latent space if:

$$\hat{d}_c.\hat{d} \to 1 \tag{15}$$

which shows that model tends to move the position of data point $\mathcal{X}_i$ in latent space when changed to $\vec{\mathcal{Z}}_i^{j'}$, and has a positive impact on reconstruction and classification. On the contrary if:

$$\hat{d}_c.\hat{d} \to 0 \tag{16}$$

then noise has a negative impact on reconstruction and classification. The above hypothesis stands for a predominant case and is effected by various other factors which include the learning of discriminatory and reconstructive properties learned by $E(\vec{\mathcal{Z}}|\vec{\mathcal{X}})$.

2. RELATIONSHIP WITH ADVERSARIAL AUTO ENCODER

Adversarial Auto Encoder (AAE) is another prominent way of auto-encoding which uses concept of adversarial training to generate adversarial examples [1]. The most basic concept of adversarial auto-encoder is to use a Gaussian posterior distribution, $E(\mathcal{Z}|\mathcal{X})$, and enforce it over the feature space. This can be clearly explained with the following equations:

$$E(\mathcal{Z}|\mathcal{X}) = \int_\eta E(\mathcal{Z}|\mathcal{X}, \eta) p_\eta(\eta) d\eta \tag{17}$$

$$p_d(\mathcal{Z}) = \int_{\mathcal{X}} \int_\eta E(\mathcal{Z}|\mathcal{X}, \eta) p_i(\mathcal{X}) p_\eta(\eta) d\eta d\mathcal{Z} \tag{18}$$

In the above case, $p_d(\mathcal{Z})$ comes from the data-distribution of the input $p_i(\mathcal{X})$ and the added noise η at the end of the encoding distribution $E(\mathcal{Z}|\mathcal{X}, \eta)$. Also, the posterior distribution $E(\mathcal{Z}|\mathcal{X})$ is not constrained to be learned as a Gaussian, and is also dependent on our input $p_i(\mathcal{X})$.

In our model, the distribution is learned over $p\left(\mathcal{Z}_i/\mathcal{Y}_j\right)$ with the help of a discriminator assembly $D_C(\mathcal{Y}|\mathcal{Z})$ similar to learning of the distribution in adversarial auto-encoder by enforcement through a discriminator. Also, our distribution is affected by the the input data-distribution, $p_i(\mathcal{X})$, and learning of combined discriminatory as well as reconstructive features by our model.

3. RESULTS AND VISUALIZATIONS

3.1. Dimensionality Reduction Using DiscAE

DiscAE has the potential for decent dimensionality reduction. The latent space has a possibility of variation in dimension, impacting the learning and representation of the features. Sometimes lower dimensions can be better than higher dimensions or vice-versa.

Figure 2 shows the representation plots for different latent spaces over multiple dimensions across the three datasets. Fig. 2 (a),(b),(c) show the dimensions for a 2D space which shows a clear representation of class clusters and gives a good indication about latent learning of the network. 10D and 100D feature spaces have been represented by a two dimensional space using t-SNE algorithm. It can be seen that the MNIST dataset gives the best representation plots in comparison to others.

3.2. Cluster Loss Analysis

Seeing the formation of clusters in the latent space for a given label, the concept of cluster loss was introduced. Cluster loss is sensitive to hyper-parameters like batch size. Also, cluster amplification factor was used to increase the impact of cluster loss.

Table 1. Variation of accuracy for different batch sizes on using cluster loss for MNIST dataset.

Batch size	Cluster amplification factor	Accuracy
128	1000	0.9902
256	1000	0.9906
512	1000	0.9912
1024	1000	0.9916
2048	1000	0.9883
4096	1000	0.9891

Table 2. Variation of accuracy for different batch sizes on using cluster loss for fashion-MNIST dataset.

Batch size	Cluster amplification factor	Accuracy
128	1000	0.9185
256	1000	0.9234
512	1000	0.9200
1024	1000	0.9200
2048	1000	0.9153
4096	1000	0.9160

3.3. Impact of Batch Size

Table 1 and 2 show the impact of cluster loss on the classification accuracy upon using various batch sizes for training with a cluster amplification factor of 1000.

For MNIST dataset, it can be observed that the maximum classification accuracy occurs at a batch size of 1024 i.e., 0.9916. Similarly, for Fashion MNIST dataset, the maximum accuracy 0.9234 occurs for a batch size of 256.

It is observed that clustering loss is effective at larger batch sizes, since, a larger batch size allows the better estimation of centroidal position. However, the larger batch size also affects the training accuracy. Hence, an optimal batch size is required for the best impact of cluster loss.

3.4. Impact of Cluster Amplification Factor

Table 3 shows the impact of cluster amplification factor on accuracy. It can be seen that on increasing the value of cluster amplification factor the accuracy increases up to a certain limit. It is observed that for a cluster amplification factor of 1000, maximum accuracy (0.9916) is achieved.

3.5. Comparison to Auto Encoder

Simple auto-encoder learns the features directly through reconstruction, whereas DiscAE can learn from both reconstructions as well as classification task. The 2D t-SNE plots for 100D latent space are shown in the Fig. 3.

Table 3. Variation of accuracy for different values of cluster amplification factor upon using cluster loss for MNIST dataset.

Batch size	Cluster amplification factor	Accuracy
1024	1	0.9892
1024	10	0.9897
1024	100	0.9906
1024	1000	0.9916
1024	10000	0.9806

3.6. Effect of Label Information

Figure 4 shows the effect of availability of label information on the accuracy during classification. Fig. 4 (a),(c),(e) show the t-SNE plot for using 2 label, 4 label, and 6 label respectively. Fig. 4 (b),(d),(f) show the results over 10 labels for MNIST dataset.

4. REFERENCES

[1] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

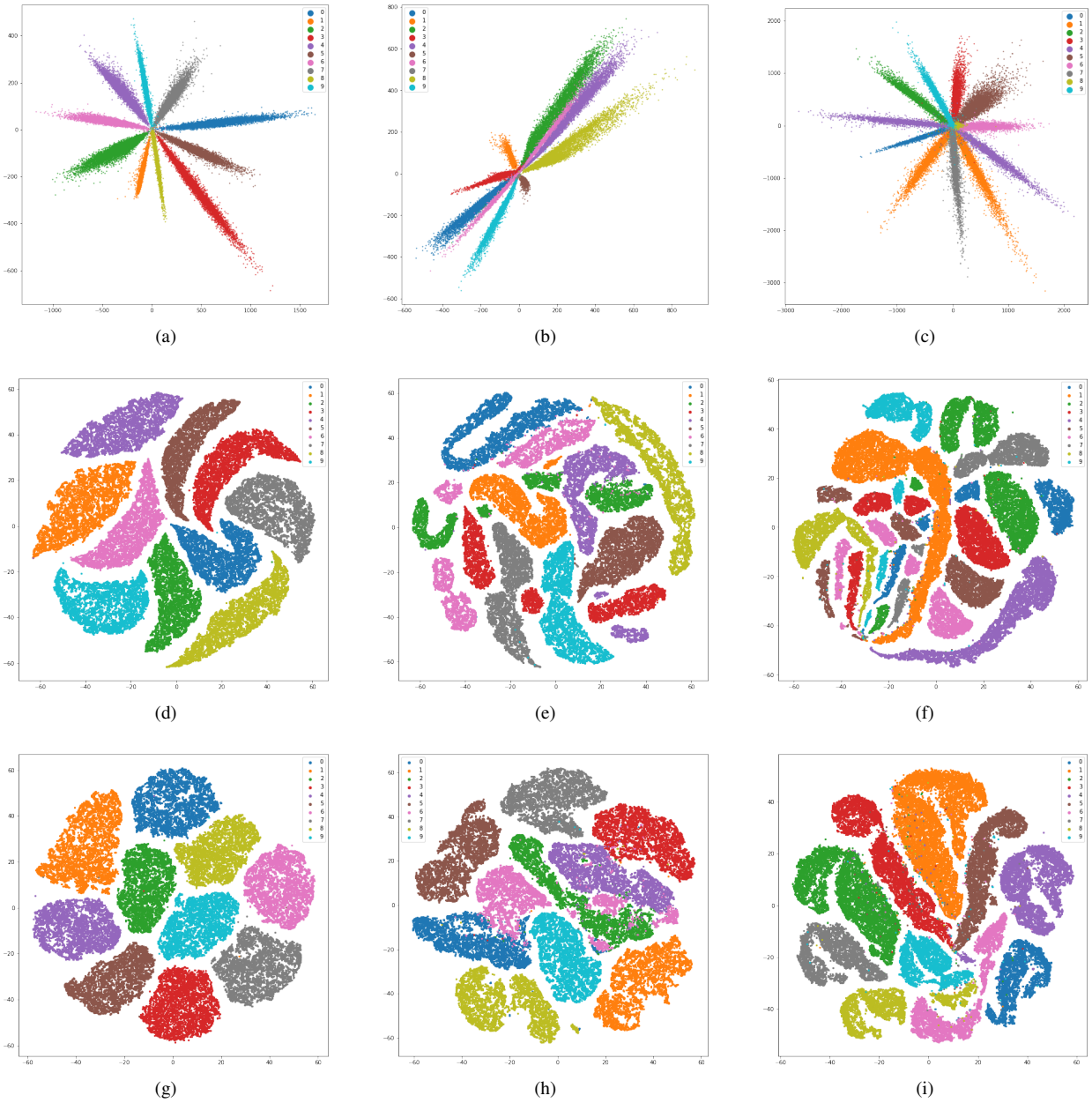
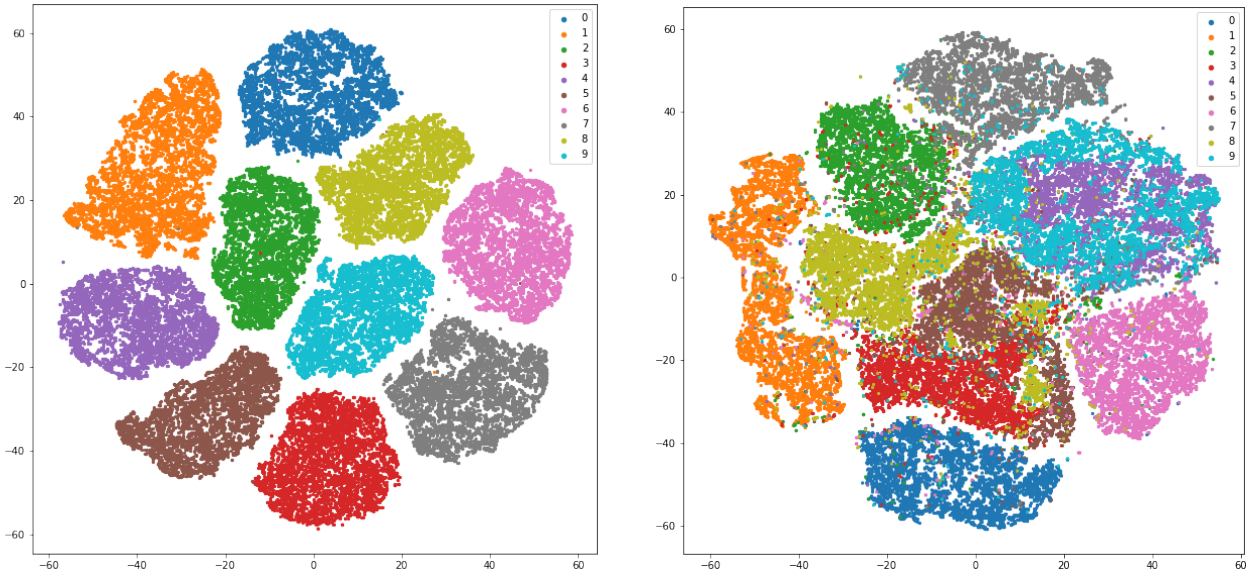
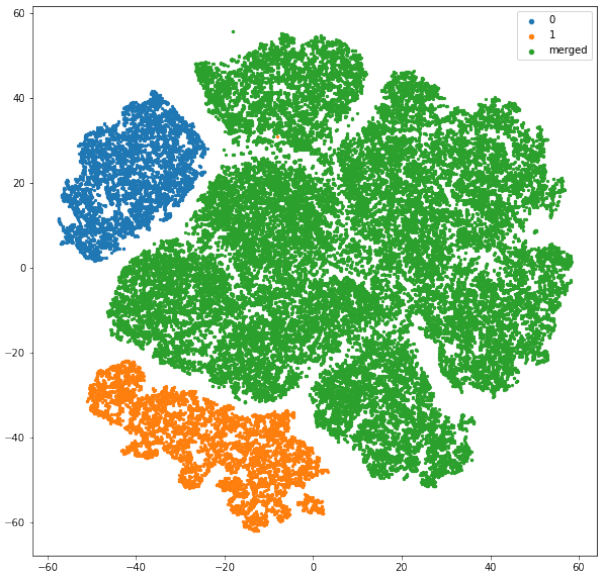


Fig. 2. (a) 2D latent space plot for MNIST (b) 2D latent space plot for Fashion MNIST (c) 2D latent space plot for SVHN (d) 10D latent space plot for MNIST using t-SNE (e) 10D latent space plot for Fashion MNIST using t-SNE (f) 10D latent space plot for SVHN using t-SNE (g) 100D latent space plot for MNIST using t-SNE (h) 100D latent space plot for Fashion MNIST using t-SNE (i) 100D latent space plot for SVHN using t-SNE.

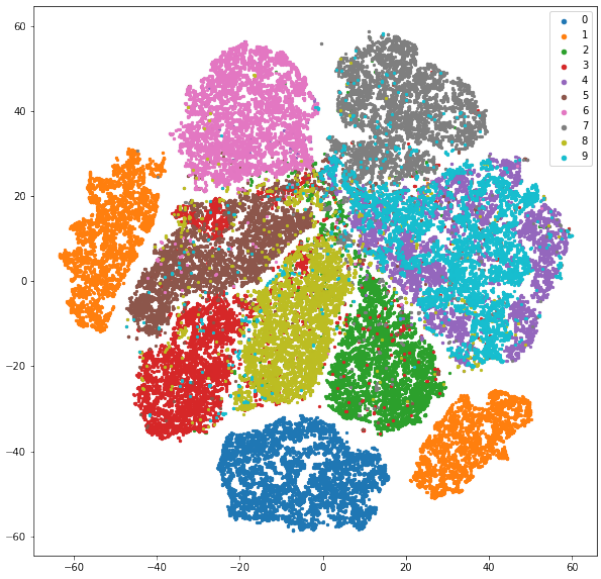


(a) (b)

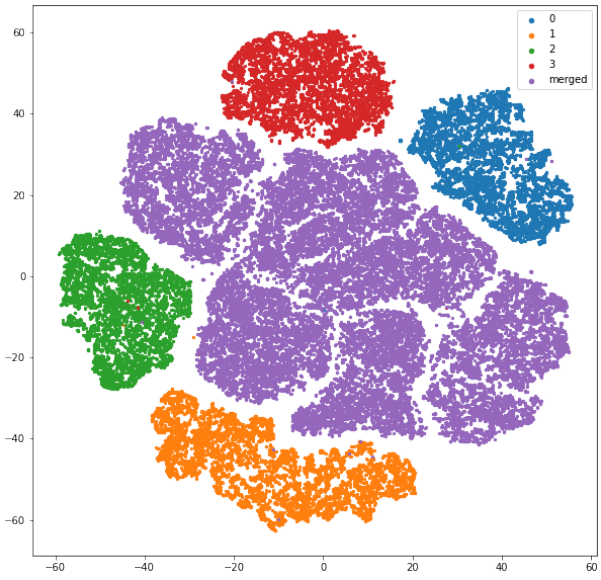
Fig. 3. 100D latent space t-SNE visualization on MNIST dataset for (a) DiscAE (without cluster-loss and noise) (b) Simple auto-encoder.


0
1
merged

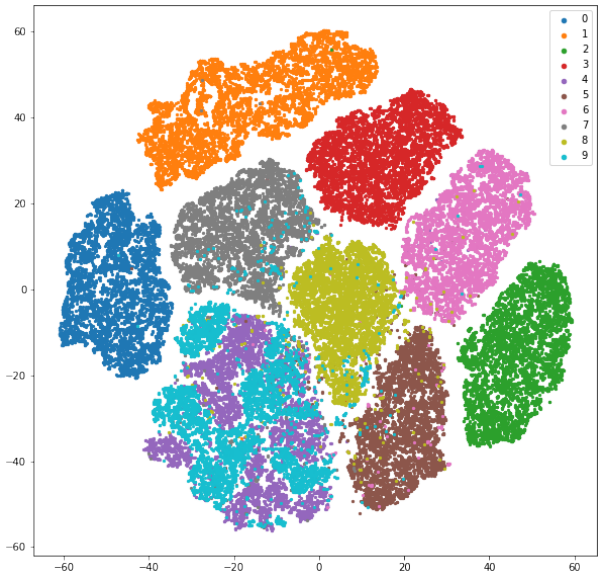
(a)


0
1
2
3
4
5
6
7
8
9

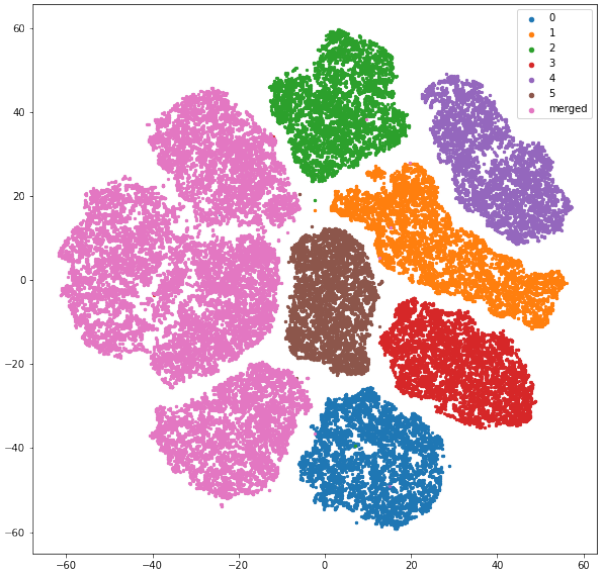
(b)


0
1
2
3
merged

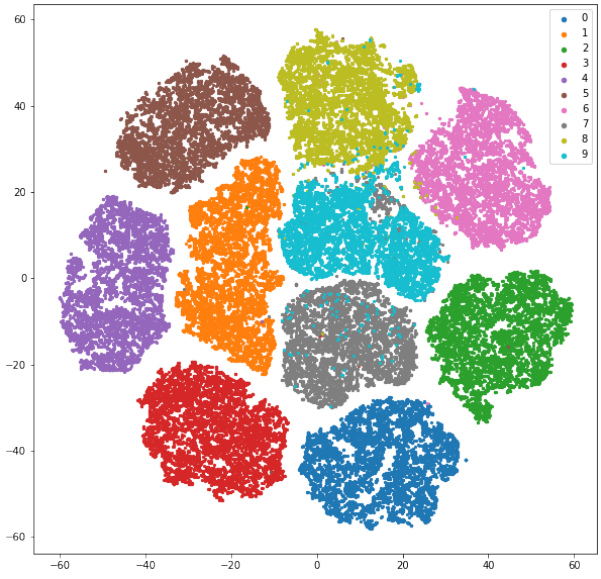
(c)


0
1
2
3
4
5
6
7
8
9

(d)

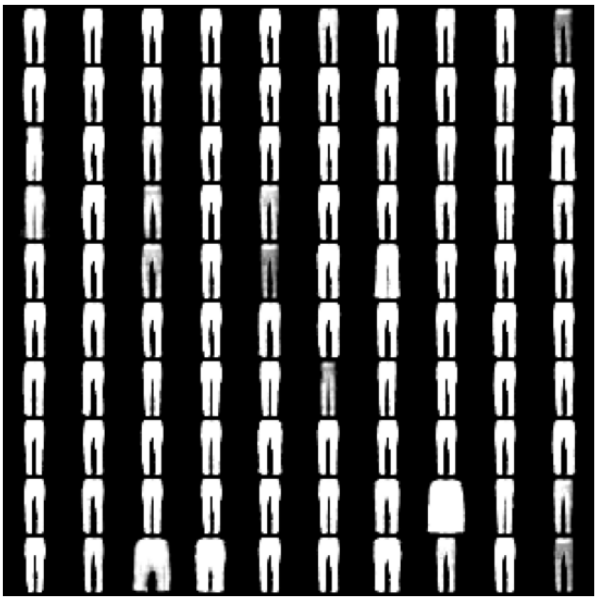

0
1
2
3
4
5
merged

(e)


0
1
2
3
4
5
6
7
8
9

(f)

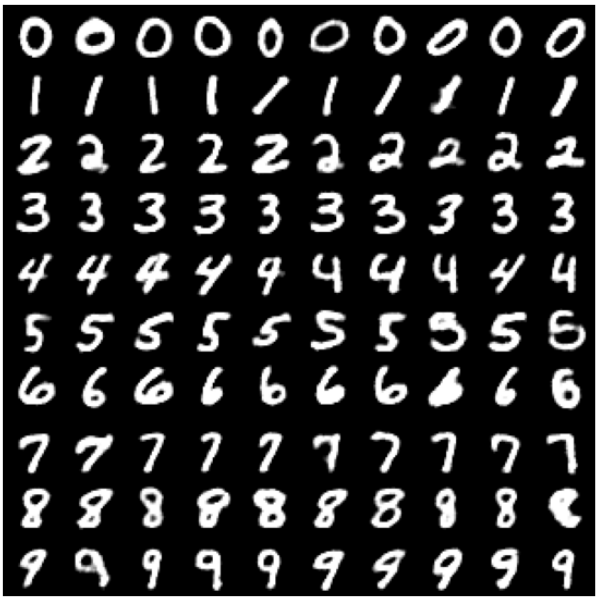
(a)



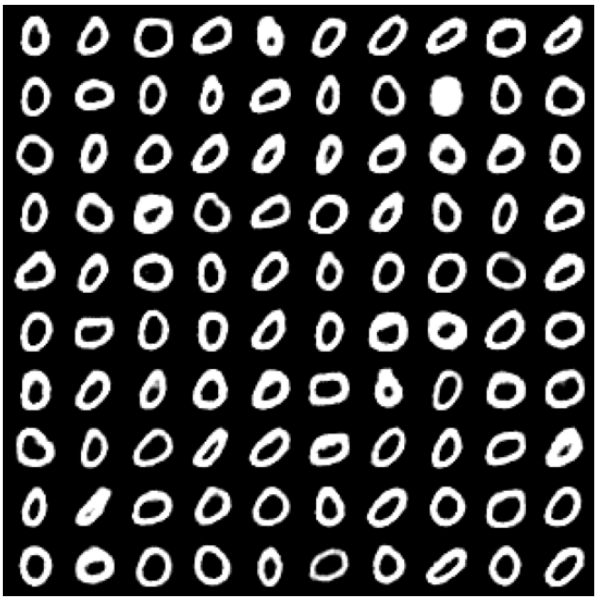
(b)

(c)

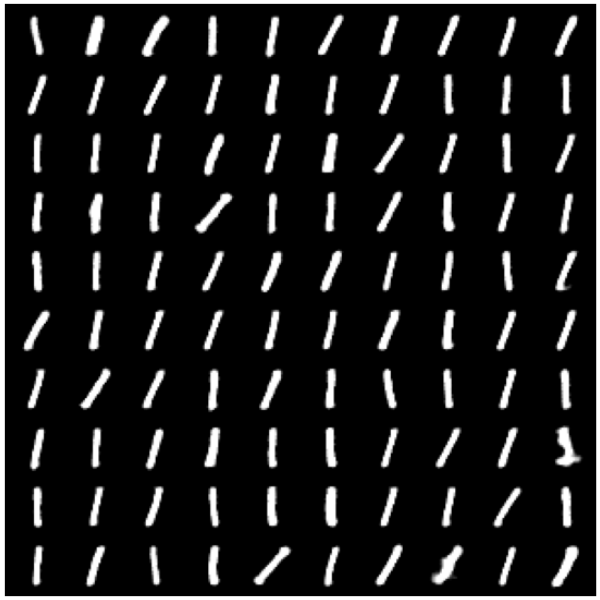
Fig. 5. Reconstructions for Fashion MNIST dataset with 100D latent space.



(a)



(b)



(c)

Fig. 6. Reconstructions for MNIST dataset with 100D latent space.