Statistical Distance Latent Regulation Loss for Latent Vector Recovery

Jeongik Cho[1]

jeongik.jo.01@gmail.com[1]

Abstract

Finding a latent vector that can generate specific data by inverting a generative model is called latent vector recovery (or latent vector projection). When performing gradient descent based latent recovery, the latent vector being recovered may deviate from the train latent distribution. To prevent this, latent regulation loss or element resampling has been used in some papers.

In this paper, we propose a statistical distance latent regulation loss, which is a latent regulation loss that can be used when the generative model is trained with IID (Independent and Identically Distributed) random variables. The statistical distance latent regulation loss is the distance between the distribution followed by train latent random variables and the discrete uniform distribution, assuming that each element of the latent vector has the same probability. Since the statistical distance latent regulation loss considers the correlation between each element of the latent vector, better latent vector recovery is possible.

In addition, in this paper, when evaluating the performance of latent vector recovery, we propose latent distribution goodness of fit test, an additional test that checks whether the distribution of all elements of all recovered latent vectors follows the distribution of the train latent random variable. Passing the latent distribution goodness of fit test does not mean that the latent vector recovery is properly performed, but when the latent recovery is properly performed, the latent distribution goodness of fit test must be passed.

In this paper, the performance of the statistical distance latent regulation loss was compared with other latent regulation losses and element resampling methods.

In conclusion, the performance of the statistical distance latent regulation loss using Wasserstein distance or Energy distance was the best.

1. Introduction

The generative model (generator) G is trained to transform a multivariate random variable $Z \in R^{d_z}$ of dimension d_z with a certain distribution into a data multivariate random variable $X \in R^{d_x}$ of dimension d_x. At this time, finding the ideal latent vector z^* that can generate some data x sampled from the data random variable X by inverting the pre-trained generator G is called latent vector recovery (or latent vector projection). There are gradient descent-based methods [1, 2, 3, 4, 5] and encoder-based methods [6, 7, 8, 9, 10] for latent vector recovery. The encoder-based methods require additional encoder training. In this paper, only gradient descent-based methods are covered.

Gradient descent-based latent vector recovery receives an error between the data $G(z_p)$ generated through the latent vector z_p and the input data x as a reconstruction loss and repeatedly performs gradient descent on the latent vector z_p. The following function shows the gradient descent-based latent vector recovery process.

$function\ latent\_recovery(x, G, t, opt)$:

$\quad z_p \leftarrow initialize(\,)$

$\quad repeat\ t\ times$:

$\quad\quad L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

$\quad\quad L \leftarrow L_{rec}$

$\quad\quad z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

$\quad return\ z_p$

Fig.1 Latent recovery function

$initialize$ is a function that initializes the values of z_p. t is the number of times to perform gradient descent. opt is an optimizer. $diff$ is a function that measures the difference between two data. In [5], the performance when using different $diff$ functions was compared. L_{rec} is the reconstruction loss. L is the total loss.

Through the function above, it can be found that the latent vector z_p that minimizes the reconstruction loss L_{rec}. However, even except for the local optimum problem, the latent vector z_p found is not always the ideal latent vector z^*. The reason is that the probability that the latent vector z_p was sampled from the train latent random variable Z may be very low. To prevent this, an additional term that maximizes $P(Z = z_p)$ is needed.

To maximize $P(Z = z_p)$, latent regulation loss was added to loss L in [1, 2], and some elements of latent vector z_p were resampled after gradient descent in [3, 4]. The following function shows latent vector recovery using latent regulation loss.

$function\ latent\_recovery(x, G, t, opt)$:

$\quad z_p \leftarrow initialize(\,)$

$\quad repeat\ t\ times$:

$\quad\quad L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

$\quad\quad L \leftarrow L_{rec} + \lambda_{lr} L_{lr}$

$\quad\quad z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

$\quad return\ z_p$

Fig.2 Latent recovery function with latent regulation loss

L_{lr} is the latent regulation loss, and λ_{lr} is the latent regulation loss weight, respectively. The following function is a function that performs latent vector recovery using element resampling.

$function\ latent\_recovery(x, G, t, opt)$:

$\quad z_p \leftarrow initialize(\,)$

$\quad repeat\ t\ times$:

$L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

$L \leftarrow L_{rec}$

$z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

$z_p \leftarrow resampling(z_p)$

$return\ z_p$

Fig.3 Latent recovery function with element resampling

$resampling$ is a function that resampling specific elements of latent vector z_p from train latent random variable Z.

In this paper, to maximize $P(Z = z_p)$, we propose a statistical distance latent regulation loss, which is a latent regulation loss that can be used assuming that the train latent random variable Z is an IID random variable that follows some distribution A^{d_z}. The statistical distance latent regulation loss is the distance between the distribution A followed by train latent random variables, and the discrete uniform distribution S, assuming that each element of the latent vector has the same probability (probability mass function $P_S(x) = \begin{cases} \frac{1}{d_z} & if\ x \in z_p \\ 0 & otherwise \end{cases}$).

Since the statistical distance latent regulation loss considers the correlation between each element of the latent vector, better latent vector recovery is possible. Also, since statistical distance latent regulation loss can be used if the train latent random variable Z is an IID random variable, it can be used when $Z \sim U(a,b)^{d_z}$, $Z \sim N(\mu, \sigma^2)^{d_z}$, or any distribution $Z \sim A^{d_z}$, so it is more versatile.

In most previous works (e.g. [1,2,4]), the performance of latent vector recovery was evaluated only with the reconstruction loss L_{rec} (or $diff\left(x, G(z_p)\right)$ using different $diff$). However, as explained previously, when the latent vector z_p that minimizes the reconstruction loss L_{rec} is found, the z_p cannot always be the ideal latent vector z^*.

In [3], after generating data $G(z_k)$ through some latent vector z_k, latent vector z_p is obtained by performing latent vector recovery on data $G(z_k)$. After that, the latent vector recovery is evaluated with the error between z_k and z_p. However, this evaluation method cannot be a proper evaluation method because it evaluates latent vector recovery only with data that generator G can generate.

In this paper, we propose latent distribution goodness of fit test that tests whether the distribution of all elements of all recovered latent vector z_p follows the train latent random variable Z to evaluate whether correct latent vector recovery has been performed. Passing the latent distribution goodness of fit test does not mean that the latent vector recovery is successful, but if the latent vector recovery is successful, it should pass the latent distribution goodness of fit test.

2. Statistical distance latent regulation loss

When there is a latent vector z_p that minimizes the reconstruction loss L_{rec}, the obtained latent vector z_p cannot always be an ideal latent vector z^*. The reason is that the probability that the latent vector z_p was sampled from the train latent random variable Z may be very low. For example, in MNIST handwritten data, assume that x is the handwritten data of the number one, currently $G(z_p)$ generates number zero, and $z_p[1]$ (the first element of z_p) represents the width of the letter. If the other elements of the latent vector z_p remain unchanged and $z_p[1]$ becomes extremely low, the width of the letter becomes very narrow and may look like the number one. At this time, when the reconstruction loss L_{rec} of the latent vector z_p is sufficiently low, the latent vector z_p may be a local optimum or a global optimum for the reconstruction loss L_{rec}. However, $P(Z = z_p)$ at this time will be very low or zero. Also, since generator G is not trained to generate out-of-distribution data, there is always a tendency to generate data distribution X. For example, the following figure is an image generated by the GAN generator that trained MNIST handwriting data with the train latent vector $Z \sim U(-1,1)^{d_z}$.

Fig.4 Generated data by GAN trained with $Z \sim U(-1,1)^{d_z}$

The FID [11] of this GAN is 5.64053. When $Z' \sim U(-10,10)^{d_z}$ is input to this pre-trained GAN, the following data is generated.

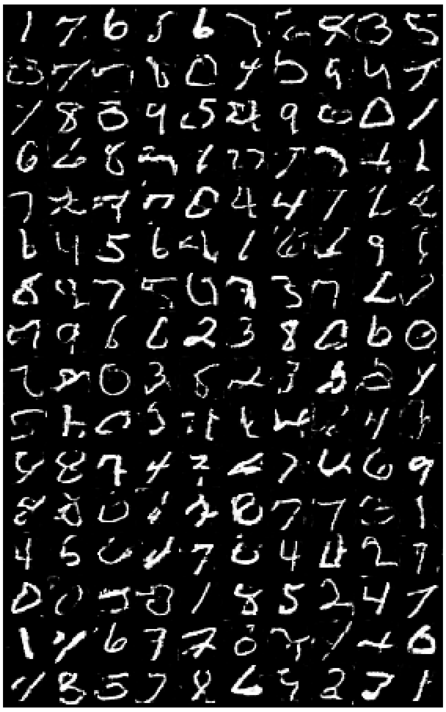

Fig.5 Generated data by GAN with input

$Z' \sim U(-10,10)^{d_z}$

You can see that a lot of data looks like in-distribution data. This model's FID, measured when the input is $Z' \sim U(-10,10)^{d_z}$, is 40.521896.

On the other hand, if $Z \sim U(-1,1)^{d_z}$ is input to the untrained GAN, the following images are generated.

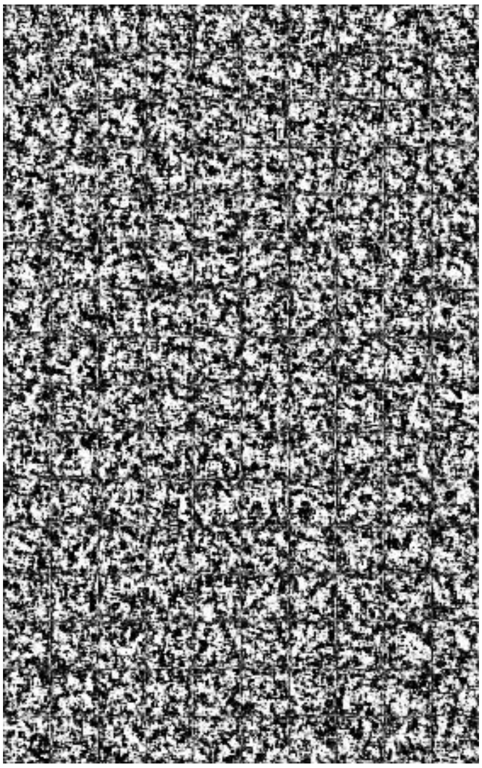

Fig.6 Generated data by untrained GAN with input $Z \sim U(-1,1)^{d_z}$

The FID of GAN at this time is 459.56543. That is, generator G of trained GAN tends to generate in-distribution data even for latent vector k with low $P(Z=k)$. Therefore, $P(Z=z_p)$ can be very low for a latent vector z_p that is a local optimum or a global optimum that sufficiently minimizes the reconstruction loss L_{rec}. These latent vectors z_p cannot be considered as ideal latent vectors z^*. This means that an additional term is needed to maximize $P(Z=z_p)$.

In papers [1, 2], latent regulation loss L_{lr} was added to loss L, and in papers [3, 4], some elements of z_p were resampled after gradient descent. However, these latent regulation loss or resampling methods have problems because they do not maximize $P(Z=z_p)$.

2.1 Z score square

In [1], the following latent regulation loss L_{lr} was used when the train latent random variable $Z \sim N(\mu, \sigma^2)^{d_z}$.

$$L_{lr} = \left(\frac{z_p - \mu}{\sigma}\right)^2$$

However, in the case of the z score square latent regulation loss, $P(Z=z_p)$ is not maximized, but $\sum_{i=1}^{d_z} P(Z[i]=z_p[i])$ is maximized. Therefore, correct latent vector recovery cannot be achieved with Z score square latent regulation loss. Also, the Z score square latent regulation loss cannot be used when the train latent random variable $Z \nsim N(\mu, \sigma^2)^{d_z}$.

2.2 Fool discriminator

In [2], the following latent regulation loss L_{lr} was used.

$$L_{lr} = L_{adv}^{g}$$

L_{adv}^{g} is the adversarial loss of generator G. For example, if GAN was trained with the adversarial loss of LSGAN, then $L_{lr} = \left(D\left(G(z_p)\right) - 1\right)^2$. D is the discriminator.

However, the fool discriminator latent regulation loss does not maximize $P(Z = z_p)$, and there is no guarantee that that the adversarial loss L_{adv}^{g} of generator G is minimized when $P(Z = z_p)$. Moreover, since discriminator D is used to calculate the latent regulation loss L_{lr}, the operation is very slow.

2.3 Boundary resampling

In [3], when the train latent random variable $Z \sim U(a,b)^{d_z}$, all elements of the latent vector z_p out of the range $[a,b]$ were resampled from $U(a,b)$. Boundary resampling resample all elements of latent vector z_p out of range $[a,b]$ from $U(a,b)$, so $\sum_{i=1}^{d_z} P(Z[i] = z_p[i])$ is maximized. However, it does not maximize $P(Z = z_p)$.

2.4 Stochastic resampling

In [4], when the train latent random variable $Z \sim N(\mu, \sigma^2)^{d_z}$, each element of the latent vector z_p is stochastically resampled from $N(\mu, \sigma^2)$. The probability of each element is resampled according to the value of the element $z_p[i]$ of the latent vector z_p according to the probability function proposed in the paper. The closer $z_p[i]$ to μ, the lower the probability of resampling. The paper argued that the following two resampling probability functions have good performance.

$$f_{lc}(z_p) = \frac{1}{1 + e^{-a(|z_p| - b)}}$$

$$f_{tc}(z_p) = \begin{cases} if\ |z_p| < a: \dfrac{e^{-\frac{a^2}{2}}}{e^{-\frac{{z_p}^2}{2}}} \\ else: 1 \end{cases}$$

In the above equation, for convenience, $Z \sim N(0, 1^2)^{d_z}$ is assumed. If it is not $Z \sim N(0, 1^2)^{d_z}$, then scale and shift are needed to $Z \sim N(0, 1^2)^{d_z}$. f_{lc} is a logistic cutoff function, and f_{tc} is a truncated normal cutoff function. The output of each function is the probability of resampling.

It is unlikely that stochastic resampling maximizes $\sum_{i=1}^{d_z} P(Z[i] = z_p[i])$, and does not maximize $P(Z = z_p)$.

2.5 Statistical distance latent regulation loss

In this paper, to maximize $P(Z = z_p)$, we propose a statistical distance latent regulation loss, which is a latent regulation loss that can be used assuming that the train latent random variable Z is an IID random variable that follows some distribution A^{d_z}. The statistical distance latent regulation loss is the distance between the distribution A followed by train latent random variables, and the discrete uniform distribution S, assuming that each element of the latent vector has the same probability (probability mass function $P_S(x) = \begin{cases} \frac{1}{d_z}\ if\ x \in z_p \\ 0\ otherwise \end{cases}$). The statistical distance latent regulation loss is as follows.

$$L_{lr} = Dist(P_A, P_S)$$

$Dist$ is a function that calculates the statistical distance between two distributions. P_A is the probability density function of distribution A.

P_S is the probability mass function of the discrete uniform distribution created by the latent vector z_p. Since the statistical distance latent regulation loss considers the correlation between each element of the latent vector, better latent vector recovery is possible. Also, since statistical distance latent regulation loss can be used if the train latent random variable Z is an IID random variable, it can be used when $Z \sim U(a,b)^{d_z}$, $Z \sim N(\mu,\sigma^2)^{d_z}$, or any distribution $Z \sim A^{d_z}$, so it is more versatile.

Among several statistical distances, in this paper, two statistical distances were used: Wasserstein distance and Energy distance. The following table summarizes the conditions required for each latent regulation loss or resampling method.

Name	Z~ALL	Z~IID	Z~N	Z~U
Wasserstein distance		O	O	O
Energy distance		O	O	O
Fool discriminator	O	O	O	O
Z score square			O	
Logistic cutoff			O	
Truncated normal cutoff			O	
Boundary resampling				O

Table.7 Requirements for each method

Z~ALL in the table above means that the train latent vector Z can be used regardless of the distribution, and Z~IID means that it can be used when Z is an IID random variable.

3. Latent distribution goodness of fit test

As explained previously, the latent vector z_p with low reconstruction loss L_{rec} is not always the ideal latent vector z^*. To check whether the latent vector z_p is sampled from the train latent random variable Z, this paper proposes a latent distribution goodness of fit test. In [12], the goodness of fit test was used to evaluate the performance of GAN. However, in this paper, it is used to verify that the correct latent vector has been recovered.

Suppose that for some distribution A, the train latent random variable $Z \sim A^{d_z}$, and latent vector recovery for the test data was properly performed. At this time, the distribution of all elements of all recovered latent vector z_p, that is, $test\ data\ size \times d_z$ samples, will follow distribution A. Latent distribution goodness of fit test tests whether the $test\ data\ size \times d_z$ samples follow the distribution A. If the distribution consisting of elements of latent vectors does not pass the latent distribution goodness of fit test, it cannot be considered that proper latent vector recovery has been achieved.

However, just passing the latent distribution goodness of fit test does not mean that proper latent vector recovery has been achieved. For example, if the latent vector z_p is initialized to a value sampled from the train latent random variable Z and then latent vector recovery is performed with a very low learning rate, the latent distribution goodness of fit test can be passed. Therefore, the reconstruction loss L_{rec} (or $diff\left(x, G(z_p)\right)$) is still important for evaluation.

In other words, the Latent distribution goodness of fit test is an additional test whether latent vector z_p minimizing reconstruction loss L_{rec} (or $diff\left(x, G(z_p)\right)$) has been properly recovered.

And even if the train latent random variable Z is not an IID random variable, the latent distribution goodness of fit test can be performed. Elements with the same index of the recovered latent vector z_p should have been sampled from the same distribution. In other words, it is possible to evaluate whether latent recovery is appropriate by performing a latent distribution goodness of fit test for each distribution of elements with the same index.

4. Material and methods

4.1 Model train

For the experiment, we trained the GAN that generates the MNIST handwriting dataset [14] using the adversarial loss of LSGAN [13]. The latent vector dimension $d_z = 256$. Each GAN used in the experiment referred to the structure of DCGAN[15], and used $Adam(learning\ rate = 10^{-5}), epoch = 200, batch\ size = 32$. For evaluation, classification tests, latent distribution goodness of fit test, L1 loss, and L2 loss were used. The classifier used in the evaluation was trained using $optimizer = Adam(learning\ rate = 10^{-5}), epoch = 50, batch\ size = 32$.

4.2 Latent vector recovery

z_p initialize function $initialize()$ is $sampling(Z)$. Sixteen latent vectors per data were initialized and optimized in parallel, and among them, the latent vector z_p with the lowest loss L was selected. For $diff$, mean absolute error, which obtained the best result in [5], was used. The number of gradient descent $t = 200$ and optimizer $opt = Adam$. In the evaluation, only 1000 randomly selected out of 10000 test data were used for each experiment run. KS-test (Kolmogorov–Smirnov test) was used as the latent distribution goodness of fit test. The test is a two-sided test. The null hypothesis H_0 is "All elements of all recovered latent vector z_p were sampled from the train latent distribution Z". Each table in the resulting section has a p-value. If the significance probability is 5%, the null hypothesis H_0 is rejected when the p-value is less than 5%, and the alternative hypothesis H_1 is rejected when it is above 95%. Wasserstein distance and energy distance were measured by sampling sufficiently many samples (10000 samples) from the train latent random variable Z.

In the case of logistic cutoff, two hyperparameters are used, of which b is fixed to 2, which has the best performance in the paper. All experiments were conducted 3 times including GAN and classifier training. All figures in the Results section are the average of the results of the three trials.

5. Experimental results and discussion

The classifier shows 99.18% accuracy for test data.

The following tables show the performance according to the latent regulation loss when the train latent random variable $Z \sim N(0, 1^2)^{d_z}$.

The FID of GAN is 6.576.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		0.003	0.001	0.003	0.006	0.034
Latent variance		0.998	0.994	0.999	1.331	19.630
Goodness of fit p-value (%)		41.21%	37.24%	26.34%	0.00%	0.00%
L1 loss		167.491	122.348	38.603	18.930	20.860
L2 loss		15.355	12.366	5.088	2.678	2.947
Classifier accuracy (%)		41.63%	63.90%	96.27%	99.00%	98.20%

Table 8. Without regulation loss performance

When the learning rate is low, the latent vector z_p hardly changes from the initial latent vector, so the p-value is high, but the L1 loss and L2 loss are high and the classifier accuracy is low. That is, L_{rec} is too large. When the $learning\ rate$ is high, L_{rec} is sufficiently low because L1 loss, L2 loss, and classifier accuracy are low, but since the p-value is too low, it is difficult to say that latent vector recovery has been properly performed. When learning rate = 0.001, sufficient latent space search was not performed due to low learning rate. When the $learning\ rate = 0.01$, due to the high learning rate, sufficient latent space search was performed, but the p-value was too low. Therefore, $learning\ rate = 0.01$ was used in later experiments using latent regulation loss or resampling.

Wasserstein distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.004	0.002	0.000	0.000	0.000
Latent variance		1.305	1.180	1.000	0.995	0.995
Goodness of fit p-value (%)		0.00%	0.00%	99.99%	59.08%	56.63%
L1 loss		18.876	18.798	19.976	27.861	69.839
L2 loss		2.668	2.650	2.789	3.804	8.193
Classifier accuracy (%)		98.80%	99.10%	98.90%	98.73%	85.97%

Table 9. Wasserstein latent regulation loss results

Energy distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.006	0.003	0.000	0.000	0.000
Latent variance		1.319	1.231	1.022	0.996	0.996
Goodness of fit p-value (%)		0.00%	0.00%	61.86%	99.18%	89.56%
L1 loss		18.840	18.889	19.589	29.403	82.368
L2 loss		2.663	2.673	2.745	3.979	9.273
Classifier accuracy (%)		99.03%	98.33%	98.93%	98.30%	81.83%

Table 10. Energy latent regulation loss results

Wasserstein latent regulation loss and energy latent regulation loss showed sufficiently high p-value when using the appropriate latent regulation loss weight λ_{lr}, and showed better performance in all aspects than when the latent regulation loss was not used. Also, because the p-value is very high, the alternative hypothesis H_1 (All elements of all recovered latent vector z_p were not sampled from the train latent distribution Z) can also be rejected.

Z score square		Latent regulation loss weight			
		0.001	0.0032	0.0057	0.01
Latent mean		0.006	0.004	0.003	0.004
Latent variance		1.259	1.124	0.995	0.816
Goodness of fit p-value (%)		0.00%	0.00%	5.58%	0.00%
L1 loss		18.919	18.829	18.628	18.218
L2 loss		2.665	2.659	2.624	2.559
Classifier accuracy (%)		98.70%	98.70%	98.97%	99.07%

Table 11. Z score square latent regulation loss results

In the case of the Z score, a more meticulous hyperparameter search was performed so that the latent variance was 1 to find the latent regulation loss weight λ_{lr} with a barely significant p-value. However, it is still much lower than the statistical distance regulation loss.

Fool discriminator		Latent regulation loss weight				
		0.000001	0.0001	0.01	1	100
Latent mean		0.003	0.004	0.006	0.005	0.004
Latent variance		1.328	1.330	1.298	1.119	1.119
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	0.00%	0.00%
L1 loss		19.054	18.840	21.413	152.095	183.491
L2 loss		2.690	2.665	3.020	14.423	16.311
Classifier accuracy (%)		98.80%	98.77%	98.50%	45.67%	38.83%

Table 12. Fool discriminator latent regulation loss results

Logistic cutoff		Hyperparameter		
		2	2.5	3
Latent mean		0.001	0.001	0.000
Latent variance		0.329	0.280	0.261
Goodness of fit p-value (%)		0.00%	0.00%	0.00%
L1 loss		99.827	73.777	57.304
L2 loss		10.558	8.373	6.798
Classifier accuracy (%)		83.70%	93.80%	95.70%

Table 13. Logistic cutoff latent resampling results

Truncated normal cutoff		Hyperparameter					
		2	2.5	3	3.5	4	4.5
Latent mean		0.000	0.002	0.002	0.003	0.003	0.005
Latent variance		0.529	0.520	0.573	0.730	1.022	1.223
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
L1 loss		155.650	108.708	61.151	35.834	25.424	20.694
L2 loss		14.563	11.241	7.193	4.494	3.308	2.837
Classifier accuracy (%)		49.43%	80.47%	95.93%	98.63%	98.93%	98.83%

Table 14. Truncated normal cutoff latent resampling results

Fool discriminator, logistic cutoff latent resampling, and truncated normal cutoff latent resampling failed to find latent regulation loss weight with a meaningful p-value. This means that proper latent vector recovery has not been performed. The following tables show the performance according to the latent regulation loss when the train latent random variable $Z \sim U(-1,1)^{d_z}$. The FID of GAN is 6.9265.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		-0.002	-0.001	-0.001	-0.007	-0.077
Latent variance		0.332	0.332	0.341	0.568	19.296
Goodness of fit p-value (%)		20.89%	0.00%	0.00%	0.00%	0.00%
L1 loss		164.220	101.742	31.117	18.299	24.599
L2 loss		15.158	10.809	4.208	2.580	3.433
Classifier accuracy (%)		42.70%	73.30%	98.13%	98.90%	96.33%

Table 15. Without latent regulation loss

It shows poor performance when there is no latent regulation loss. $learning\ rate = 0.01$ is the value used in the previous experiment, and since the latent variance exceeds $\frac{1}{3}$ when $learning\ rate = 0.01$, $learning\ rate = 0.01$ was used in the next experiments. However, in the case of boundary resampling, since there is no hyperparameter, we instead tested the performance according to the learning rate.

Wasserstein distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		-0.007	-0.003	0.000	0.000	0.000
Latent variance		0.563	0.499	0.346	0.333	0.333
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	99.97%	100.00%
L1 loss		18.492	18.292	18.420	22.434	40.983
L2 loss		2.594	2.570	2.546	3.071	5.305
Classifier accuracy (%)		98.90%	98.63%	98.60%	98.90%	96.43%

Table 16. Wasserstein latent regulation loss results

Energy distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		-0.007	-0.004	0.000	0.000	0.000
Latent variance		0.562	0.510	0.362	0.334	0.333
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	66.76%	99.99%
L1 loss		18.260	18.197	18.409	25.097	57.141
L2 loss		2.573	2.550	2.564	3.426	7.003
Classifier accuracy (%)		98.93%	99.00%	98.93%	98.20%	90.93%

Table 17. Energy latent regulation loss results

When using the statistical distance latent regulation loss, like the previous experiment, latent vectors with very high p-value and low L_{rec} were found.

Fool discriminator		Latent regulation loss weight				
		0.000001	0.0001	0.01	1	100
Latent mean		-0.009	-0.007	-0.007	-0.005	-0.006
Latent variance		0.568	0.566	0.551	0.449	0.451
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	0.00%	0.00%
L1 loss		18.232	18.431	20.738	158.037	186.580
L2 loss		2.566	2.596	2.931	14.780	16.488
Classifier accuracy (%)		98.80%	98.73%	98.47%	44.67%	36.53%

Table 18. Fool discriminator latent regulation loss results

Boundary resampling		Learning rate				
		0.001	0.01	0.1	1	10
Latent mean		0.000	-0.003	-0.006	-0.005	0.000
Latent variance		0.268	0.246	0.297	0.321	0.333
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	0.00%	39.54%
L1 loss		39.809	22.854	48.838	95.118	174.681
L2 loss		5.117	3.041	5.865	10.108	15.798
Classifier accuracy (%)		96.50%	98.77%	96.40%	82.77%	37.23%

Table 19. Boundary resampling results

On the other hand, when fool discriminator latent regulation loss was used, as in previous experiments, latent vector with meaningful p-value could not be found. When the learning rate is high, boundary resampling has a meaningful p-value because it is almost always resampling, but L_{rec} is too high.

The following experiment shows the result of latent recovery by applying statistical distance latent regulation loss when the train latent random variable Z is a unique IID random variable ($Z \sim A^{d_z}$). A is half uniform and half normal distribution.

Probability density function of A is $P_A(x) = \begin{cases} 0 \ if \ x < -1 \\ 0.5 \ if \ x \in [-1,0] \\ \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \ if \ 0 < x \end{cases}$.

The following graph shows the graph of the probability density function of A.

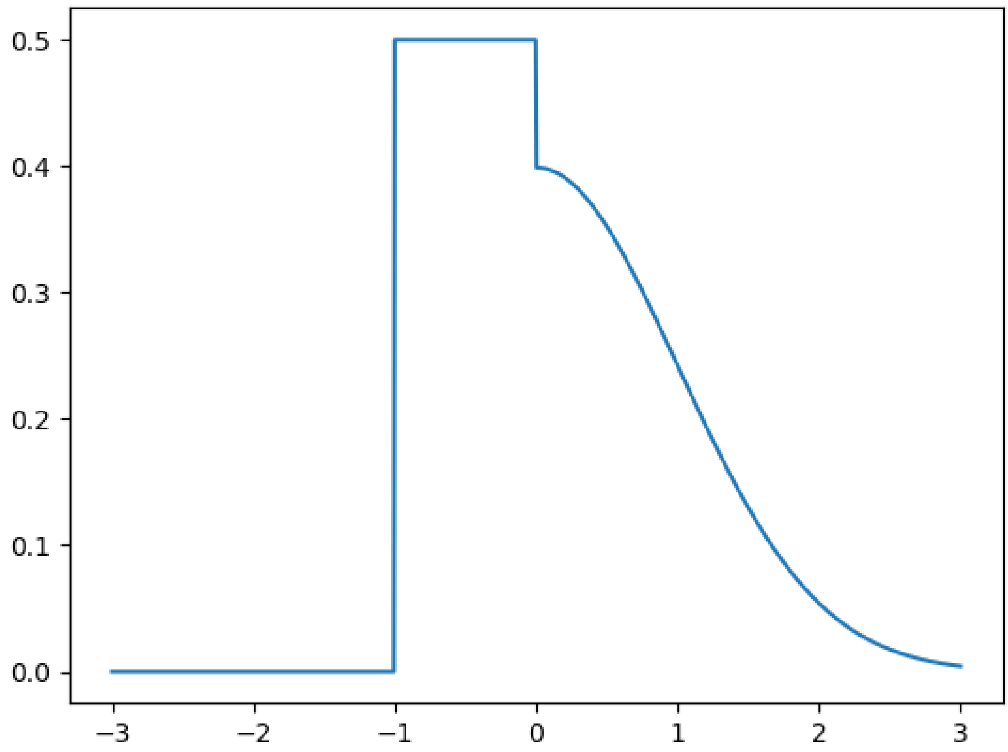

Figure 20. Half uniform and half normal probability density function

The FID of the GAN trained with $Z{\sim}A^{d_z}$ is 7.424.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		0.150	0.148	0.153	0.258	1.653
Latent variance		0.646	0.641	0.650	0.912	17.549
Goodness of fit p-value (%)		39.04%	0.00%	0.00%	0.00%	0.00%
L1 loss		166.662	115.787	35.291	19.588	22.805
L2 loss		15.300	11.859	4.677	2.725	3.159
Classifier accuracy (%)		39.77%	67.37%	96.97%	99.10%	97.50%

Table 21. Without latent regulation loss results

Without latent regulation loss, as in previous experiments, it was impossible to recover the latent vector to have a low L_{rec} while having a meaningful p-value. The following tables show the performance when using the statistical distance latent regulation loss when the $learning\ rate = 0.01$, like the previous experiments.

Wasserstein distance		Latent regulation loss weight			
		0.001	0.01	0.1	1	10
Latent mean		0.251	0.213	0.146	0.148	0.149
Latent variance		0.904	0.823	0.650	0.642	0.642
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	56.67%	57.19%
L1 loss		19.263	19.195	19.200	25.190	55.796
L2 loss		2.691	2.665	2.644	3.395	6.824
Classifier accuracy (%)		98.70%	98.47%	99.23%	98.63%	91.47%

Table 22. Wasserstein latent regulation loss results

Energy distance		Latent regulation loss weight			
		0.001	0.01	0.1	1	10
Latent mean		0.255	0.222	0.147	0.148	0.149
Latent variance		0.905	0.860	0.674	0.642	0.642
Goodness of fit p-value (%)		0.00%	0.00%	0.00%	29.56%	84.71%
L1 loss		19.545	19.367	18.904	26.919	70.570
L2 loss		2.722	2.696	2.603	3.617	8.187
Classifier accuracy (%)		98.77%	98.90%	98.90%	98.67%	86.53%

Table 23. Energy distance latent regulation loss results

The above experimental results show that the statistical distance latent regulation loss can also be used for latent vector recovery using a generative model trained with a unique IID random variable Z.

6. Conclusion

In this paper, we evaluated the performance of latent vector recovery according to the types of latent regulation loss and resampling methods. In addition, an additional test, the latent distribution goodness of fit test, was proposed to evaluate whether latent vector recovery was properly performed.

Among several latent regulation loss and resampling methods, only the statistical distance latent regulation loss proposed in this paper had a very high p-value for the latent distribution goodness of fit test. This shows that statistical distance latent regulation loss maximizes $P(Z = z_p)$, while previous works maximize $\sum_{i=1}^{d_z} P(Z[i] = z_p[i])$.

7. References

[1] A. Creswell and A. A. Bharath, "Inverting the

Generator of a Generative Adversarial Network," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 7, pp. 1967-1974, July 2019, doi: 10.1109/TNNLS.2018.2875194. [Online]. Available: https://ieeexplore.ieee.org/document/8520899

[2] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson and M. N. Do, "Semantic Image Inpainting with Deep Generative Models," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6882-6890, doi: 10.1109/CVPR.2017.728. [Online]. Available: https://ieeexplore.ieee.org/document/8100211

[3] Zachary C. Lipton, Subarna Tripathi, Precise Recovery of Latent Vectors from Generative Adversarial Networks

https://arxiv.org/abs/1702.04782v2

[4] Nicholas Egan, Jeffrey Zhang, Kevin Shen, Generalized Latent Variable Recovery for Generative Adversarial Networks

https://arxiv.org/abs/1810.03764

[5] A. Patro, V. Makkapati and J. Mukhopadhyay, "Evaluation of Loss Functions for Estimation of Latent Vectors from GAN," 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, 2018, pp. 1-6, doi: 10.1109/MLSP.2018.8517097. [Online]. Available: https://ieeexplore.ieee.org/document/8517097

[6] Jiapeng Zhu, Yujun Shen, Deli Zhao, Bolei Zhou, In-Domain GAN Inversion for Real Image Editing

https://arxiv.org/abs/2004.00049

[7] Jeff Donahue, Philipp Krähenbühl, Trevor Darrell, Adversarial Feature Learning

https://arxiv.org/abs/1605.09782

[8] Andrew Brock, Theodore Lim, J.M. Ritchie, Nick Weston, Neural Photo Editing with Introspective Adversarial Networks

https://arxiv.org/abs/1609.07093

[9] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, Aaron Courville, Adversarially Learned Inference

https://arxiv.org/abs/1606.00704

[10] Jeff Donahue, Karen Simonyan, Large Scale Adversarial Representation Learning

https://arxiv.org/abs/1907.02544

[11] Martin Heusel, Hubert Ramsauer, Thomas

Unterthiner, Bernhard Nessler, Sepp Hochreiter, GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium

https://arxiv.org/abs/1706.08500

[12] R. Webster, J. Rabin, L. Simon and F. Jurie, "Detecting Overfitting of Deep Generative Networks via Latent Recovery," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 11265-11274, doi: 10.1109/CVPR.2019.01153. [Online]. Available: https://ieeexplore.ieee.org/document/8953411

[13] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley, Least Squares Generative Adversarial Networks

https://arxiv.org/abs/1611.04076

[14] Yann LeCun, Corinna Cortes, Christopher J.C. Burges

THE MNIST DATABASE of handwritten digits

http://yann.lecun.com/exdb/mnist/

[15] Alec Radford, Luke Metz, Soumith Chintala, Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

https://arxiv.org/abs/1511.06434

8.Appendix

8.1 Model architecture

Generator

Latent vector ([256])
Fully connected layer ([7*7*512], Leaky ReLU)
Reshape ([7, 7, 512])
Instance normalization ()
Up sampling ()
Convolution layer (256, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (256, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (256, [3, 3], Leaky ReLU)
Instance normalization ()
Up sampling ()
Convolution layer (128, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (128, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (128, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (1, [1, 1], tanh)

Discriminator and Classifier architecture

Input image ([28, 28, 1])
Convolution layer (128, [3, 3], Leaky ReLU)
Instance normalization ()
Average Pooling ()
Convolution layer (256, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (256, [3, 3], Leaky ReLU)
Instance normalization ()
Convolution layer (256, [3, 3], Leaky ReLU)

<table>
<tr><td colspan="2">Instance normalization ()</td></tr>
<tr><td colspan="2">Average Pooling ()</td></tr>
<tr><td colspan="2">Convolution layer (512, [3, 3], Leaky ReLU)</td></tr>
<tr><td colspan="2">Instance normalization ()</td></tr>
<tr><td colspan="2">Convolution layer (512, [3, 3], Leaky ReLU)</td></tr>
<tr><td colspan="2">Instance normalization ()</td></tr>
<tr><td colspan="2">Convolution layer (512, [3, 3], Leaky ReLU)</td></tr>
<tr><td colspan="2">Instance normalization ()</td></tr>
<tr><td colspan="2">Flatten ()</td></tr>
<tr><td>Fully connected layer
(1, Linear)
for Discriminator</td><td>Fully connected layer
(10, Softmax)
for Classifier</td></tr>
</table>