

# Annealing Genetic GAN for Minority Oversampling

Jingyu Hao<sup>1</sup>

haojy5@mail2.sysu.edu.cn

Chengjia Wang<sup>2</sup>

chengjia.wang@ed.ac.uk

Heye Zhang<sup>1</sup>

zhangheye@mail.sysu.edu.cn

Guang Yang<sup>3</sup>

g.yang@imperial.ac.uk

<sup>1</sup> School of Biomedical Engineering

Sun Yat-sen University

Shenzhen, CHN

<sup>2</sup> University of Edinburgh

London, UK

<sup>3</sup> Imperial College London

London, UK

---

## Abstract

The key to overcome class imbalance problems is to capture the distribution of minority class accurately. Generative Adversarial Networks (GANs) have shown some potentials to tackle class imbalance problems due to their capability of reproducing data distributions given ample training data samples. However, the scarce samples of one or more classes still pose a great challenge for GANs to learn accurate distributions for the minority classes. In this work, we propose an Annealing Genetic GAN (AGGAN) method, which aims to reproduce the distributions closest to the ones of the minority classes using only limited data samples. Our AGGAN renovates the training of GANs as an evolutionary process that incorporates the mechanism of simulated annealing. In particular, the generator uses different training strategies to generate multiple offspring and retain the best. Then, we use the Metropolis criterion in the simulated annealing to decide whether we should update the best offspring for the generator. As the Metropolis criterion allows a certain chance to accept the worse solutions, it enables our AGGAN steering away from the local optimum. According to both theoretical analysis and experimental studies on multiple imbalanced image datasets, we prove that the proposed training strategy can enable our AGGAN to reproduce the distributions of minority classes from scarce samples and provide an effective and robust solution for the class imbalance problem.

## 1 Introduction

In machine learning applications, class imbalance problem, i.e., differences in prior class probabilities, hinder the performance of many standard classifiers [8]. The key to improve the classification performance with imbalanced data is to capture the distributions of minority classes precisely [10]. However, it is difficult to learn accurate distributions from scarce samples of the minority classes. A common strategy to tackle class imbalance problem is to increase the minority samples in order to have a better representation for the distributions

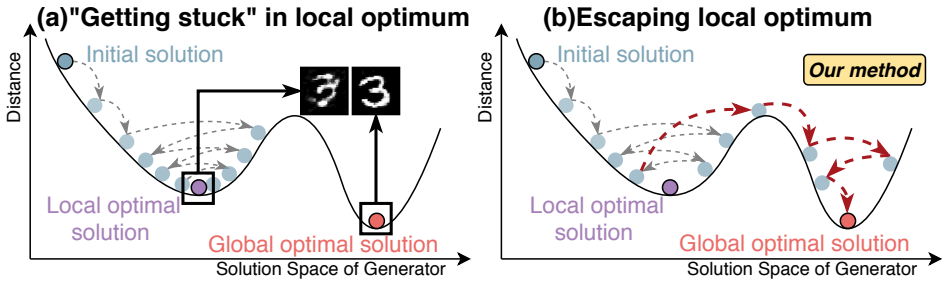


Figure 1: (a) Schematic illustration of local optimum trapping for GANs. In this situation, GANs will keep fluctuating around the local optimum until the end of training and cannot learn a distribution that is close enough to the minority class. Consequently, the samples GANs generated may have low quality. (b) Our method integrates the simulated annealing genetic algorithm into the training of GANs. In doing so, GANs may update to a worse solution with a decreasing probability, which enables GANs to escape from the local optimum.

of the minority classes. In particular, previous methods usually increased the size of minority classes by replicating or interpolating the samples from the minority classes. However, repetition may cause the problem of over-fitting because the samples from minority classes are overemphasised. On the other hand, because data are normally sitting in a high dimensional space, interpolation is nontrivial and may generate low-quality samples due to the complexity of the data manifold [4].

Recently, Generative Adversarial Networks (GANs) have shown some potentials to tackle class imbalance problems because theoretically they are able to reproduce the distributions of minority classes through adversarial learning. In the training process of GANs, the generator learns the mapping from a latent encoding space to the minority class distribution, and the discriminator needs to determine whether an input sample is actually drawn from the minority class or created by the generator [9]. As these two networks confront each other constantly, the performances of the generator and discriminator are improved alternately. Finally, the generator reproduces the distributions of the minority classes that the discriminator would not be able to distinguish from the actual minority classes.

There are many successful applications of using GANs. However, GANs can easily get stuck at local optimum when they try to learn the distributions from scarce samples of the minority classes that is also known as mode collapse [6] as shown in Figure 1 (a). A more effective training strategy is highly in demand for GANs to avoid the trapping at the local optimum. Previous attempts tried to avoid local optimum by improving adversarial learning objectives [18] [27] [9]. However, these previous strategies still used a single adversarial learning target in the training process that might still fail to fully overcome the local optimum problem and might also have other limitations, e.g., Wasserstein distance has non-convergent limit cycles near the equilibrium [20].

Considering the problems and limitations of those previously proposed methods, we propose a new model, namely Annealing Genetic GAN (AGGAN), to incorporate simulated annealing genetic algorithm into the training process of GANs to avoid the local optimum trapping problem (Figure 1 (b)). The primary contributions of the proposed method are summarised as follows:

- We develop a strategy to incorporate simulated annealing genetic algorithm into the

training process of GANs to avoid the local optimum trapping.

- Through theoretical analysis, we prove that the simulated annealing genetic algorithm enables our proposed AGGAN to reproduce the distributions closest to the minority classes.
- We also conduct comprehensive experimental studies and show that our proposed AGGAN method can solve the class imbalance problem efficiently and effectively.

## 2 Related Work

**Class Imbalance Problem** Class imbalance is a common problem in practical classification tasks. The scarce samples of a minority class make it difficult for the classifier to find the boundaries between the distributions of different classes correctly. Therefore, the key to solve class imbalance is to learn an accurate distribution of the minority classes with limited samples. Oversampling can improve the accuracy of the learning on the distributions of minority classes by increasing minority samples, which is a common method to tackle the class imbalance problem [1]. Random oversampling, Synthetic Minority Over-Sampling Technique (SMOTE) [2] and Border-line SMOTE[3] are commonly used oversampling methods in classic imbalance problems. However, when dealing with data in high dimensional space, the quality of the synthesised new data points could still be compromised due to noise and poor distance measurement in the high dimensional space [4].

**Generative Adversarial Networks** Recently, GANs have achieved great success in image generation [5][6], image-to-image synthesis [7], image super-resolution [8] and other applications due to its excellent capability of learning the data distributions by providing abundant training samples. In addition, GANs have also shown some potentials to solve class imbalance problems by learning the distributions of minority classes. Although GANs have been successfully applied in many tasks, due to the limited number of samples in the minority class, GANs may only be able to learn part of the minority class distribution at the end of training, and therefore could be trapped by the local optimum. Some studies have been done to enable GANs to learn a more accurate distribution during training process by utilising improved adversarial learning objectives (e.g. LSGAN [9], energy-based GAN [10] and WGAN [11]). Nevertheless, there still exist limitations when using fixed adversarial training objectives in the training of GANs. More recently, Evolutionary-GAN (E-GAN) [12] was proposed and multiple generators were created by different adversarial objectives to overcome the limitations of the fix adversarial objectives, and always kept the well-performed generator in the training process. However, local optimum trapping problem has yet been addressed.

## 3 Annealing Genetic GAN (AGGAN)

In this paper, we propose AGGAN that aims to learn the accurate distribution from the minority class. First, our AGGAN uses different adversarial learning objectives to improve performance of the generator. Second, our AGGAN incorporates the mechanism of simulated annealing into the training so that the model can converge to the distribution that is closest to the minority class. In particular, in lieu of normal training of GANs, our AGGAN

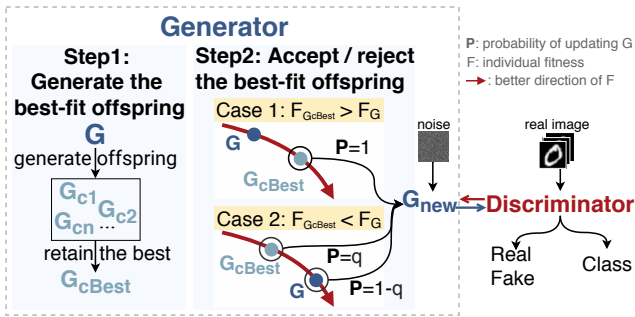


Figure 2: We illustrate the training process of the generator here. This training process can be divided into two steps, which are described in Section 3 with more details.

is trained as an evolutionary process, with the discriminator  $D$  as the *environment* and the generator  $G$  as the *individual*. Each iteration of the individual is divided into two steps including (1) generating the best-fit offspring and (2) updating the generator as illustrated in Figure 2.

### 3.1 Generating the Best-Fit Offspring

In each iteration,  $G$  gives birth to different offspring  $G_c$  by various adversarial learning objectives. Each  $G_c$  represents a solution in the parameter space of the generator network. The individual fitness of offspring  $G_c$  is evaluated based on the diversity and quality of the generated samples. Then the best-fit offspring  $G_{cBest}$  is retained while other offspring are eliminated. This process of generating the best-fit offspring  $G_{cBest}$  reflects the concept of ‘survival of the fittest’ in the GA. The strategy of using different adversarial objectives overcomes the limitations of using a fixed objective and helps the final learned generator to achieve a better performance.

### 3.2 Updating the Generator

In our study, we propose to use the mechanism of Simulated Annealing algorithm to update the generator. If the individual fitness of  $G_{cBest}$  was higher than  $G$ ,  $G_{cBest}$  will be updated to  $G_{new}$  with a probability of 1. If the individual fitness of  $G_{cBest}$  was lower than the previous generation  $G$ ,  $G_{cBest}$  will be updated to  $G_{new}$  with a probability of  $P$ . The probability  $P$  is determined based on the current temperature  $T_c$  and the difference between the two individual fitness. The temperature  $T_c$  gradually decreases from the initial temperature  $T$  conditioning on the annealing coefficient  $\alpha$ . In doing so, updating  $G$  with a decreasing probability in a worse direction enables AGGAN to asymptotically converge to the global optimum.

Finally, after updating the individual, the environment (i.e., the discriminator)  $D$  is updated and the training loop of our AGGAN starts the next evolutionary iteration. As the training progresses, the data generated by  $G$  gradually close to the true distribution, which helps  $D$  to continuously improve classification accuracy. Algorithm 1 summarises the whole training procedure of the AGGAN.

**Algorithm 1** Annealing Genetic GAN (AGGAN)**Input:** The minority class  $X$  in the imbalanced dataset.**Parameter:** the number of mutations  $n_m$ , different training strategies  $\mathcal{M}_c(c = 0, \dots, n_m)$ , initial temperature  $T$ , annealing coefficient  $\alpha$ .**Output:** The generated samples for the minority class.

---

```

1:  $\theta_D, \theta_G \leftarrow$  initialise network parameters
2: repeat
3:   Calculate individual fitness of  $G$  and expressed in  $\mathcal{F}_G$ 
4:   for  $c = 0, \dots, n_m$  do
5:      $Z \leftarrow$  samples from  $\mathcal{N}(0, 1)$ 
6:     Individual  $G$  generate offspring  $G_c$  with different adversarial objectives:
7:      $\theta_{G_c} \leftarrow \nabla_{\theta_{G_c}} \mathcal{M}_c(Z)$ 
8:     Calculate individual fitness of  $G_c$  and expressed in  $\mathcal{F}_c$ 
9:   end for
10:   $\{\mathcal{F}_{cbest}, \mathcal{F}_{c2}, \dots\} \leftarrow \text{sort}(\{\mathcal{F}_c\})$ 
11:  if  $\mathcal{F}_{cbest} > \mathcal{F}_G$  then
12:    Update  $G_{cbest}$  to  $G_{new}$ .
13:  else
14:     $\Delta = \mathcal{F}_{cbest} - \mathcal{F}_G$ 
15:     $P = e^{-\Delta/T}$ 
16:    Update  $G_{cbest}$  to  $G_{new}$  with probability  $P$ 
17:  end if
18:   $T = \alpha * T$ 
19:   $Z \leftarrow$  samples from  $\mathcal{N}(0, 1)$ 
20:   $\theta_D \leftarrow \nabla_{\theta_D} [\log D(X) + \log(1 - D(G(Z)))]$ 
21: until Convergence

```

---

## 4 Theoretical Analysis

In this section, we perform a theoretical analysis of the proposed AGGAN. We will prove that incorporating simulated annealing genetic algorithm into the training process of GANs can reinforce our AGGAN to learn the distribution closest to the minority class, that is, the AGGAN can converge to the global optimum solution with a probability of 1.

For elaboration purpose, we consider the training of GANs as a combinatorial optimisation problem. We consider this combinatorial optimisation problem as a pair of  $(G, f)$ , where  $G$  is a finite set of the solution space of the generator  $g$  and  $f$  is the object function. The aim is to find a global optimum to minimise  $f$ . It is of note that the finiteness of  $G$  implies that  $f$  has at least one minimum over  $G$ .

Below we provide the definitions of generating the best-fit offspring and updating the generator in AGGAN from a mathematical perspective and prove that the simulated annealing genetic algorithm will make AGGAN to converge to the optimum solution as the training progressing.

Symbol	Definition	Symbol	Definition
$g$	Individual	$g_c$	Offspring of $g$
$g_{cbest}$	The best offspring of $g$	$g_b$	The best $g$ so far
$G$	Solution space of $g$	$G_n$	Solution space for the $n_{th}$ iteration
$\{G_n   n \in N\}_g$	The solution space sequence with initial state $g$		

Table 1: Definition of symbols.

## 4.1 Definition

For readability, we first define the symbols we used in Table 1. Then we give the mathematical definition of ‘generating the best-fit offspring’ and ‘updating the generator’ in AGGAN. In addition, in order to obtain the monotonicity without changing the training mechanism, we add a second element  $g_b$  to each individual, which represents the best  $g$  so far.

**Generating the best-fit offspring  $F_{gen}$**  First, we use a choice function to find the parent  $g$  from  $[g, g_b]$ . Then  $g$  generates offspring  $g_c$  under the production function. Finally, we retain the best-fit offspring  $g_{cbest}$  from an individual fitness function. All the steps above describe the whole process of ‘generating the best-fit offspring’. Since the choice function, production function and individual fitness function will not be mentioned in the analysis below, we collectively denote the above process using  $F_{gen}$

$$F_{gen}(g, g_b) = g_{cbest}. \quad (1)$$

As the parameters of the choice function, production function and individual fitness function do not depend on the number of iterations,  $F_{gen}$  will follow the same distribution with different number of iterations.

**Updating the generator  $F_{upd}$**  The process of ‘updating the generator’ uses the mechanism of SA. We define it as follows

$$F_{upd}(g, g_b, g_{cbest}) = \begin{cases} [g, g'_b], & \text{if } e^{-\frac{\Delta}{T}} < \gamma \\ [g_{cbest}, g'_b], & \text{otherwise} \end{cases} \quad (2)$$

where

$$g'_b = \begin{cases} g_b, & \text{if } f(g_{cbest}) > f(g_b) \\ g_{cbest}, & \text{if } f(g_b) > f(g_{cbest}) \end{cases} \quad (3)$$

and  $\Delta$  is the difference between the individual fitness of  $g$  and  $g_{cbest}$ ,  $\gamma$  is a random variable between 0 and 1, and  $T$  denotes the temperature parameter for our AGGAN.

We combine  $F_{upd}$  and  $F_{gen}$  to obtain  $F$  for the representation of the iterative process of the generator

$$F(g, g_b) = F_{upd}(g, g_b, F_{gen}(g, g_b)), \quad (4)$$

and we use  $F_n$  to denote the  $F$  in the  $n_{th}$  iteration.

## 4.2 Proof of Convergence

In this section, according to Corollary 1, we will prove that  $\{G_n|n \in N\}_g$  satisfies the following properties, which can ensure that our AGGAN will converge to the global optimum with a probability of 1.

1 **Monotonicity** The minimum value of  $G_n$  on  $f$  decreases as  $n$  becomes larger.

2 **Homogeneity** If  $\{G_n|n \in N\}_g$  is a Markov chain, and the  $F_n$  have the same distribution then the chain is homogeneous.

First, we discuss the monotonicity of  $\{G_n|n \in N\}_g$ . Since  $F_{upd}$  always keeps the best  $f$  value, for any  $g$  we have

$$\begin{aligned} \min\{f(s)|s \in g\} &\geq \min\{f(s)|s \in [g, g_b, g_{cbest}]\} \\ &\geq \min\{f(s)|s \in f_{upd}(g, g_b, g_{cbest})\} \end{aligned} \quad (5)$$

holds.

Hence by Eq. 4, Eq. 5 and

$$G_{n+1} = F_n(G_n, g_b) \quad (6)$$

we have that

$$\begin{aligned} \min\{f(s)|s \in G_{n+1}\} &= \min\{f(s)|s \in F_n(G_n, g_b)\} \\ &= \min\{f(s)|s \in F_{upd}(G_n, g_b, F_{gen}(G_n, g_b))\} \\ &\leq \min\{f(s)|s \in G_n\}, \end{aligned} \quad (7)$$

so that  $\{G_n|n \in N\}_g$  is monotone.

Then we discuss the homogeneity. Due to the memory-less property of GANs, for any  $n$ , the conditional probability distribution of  $G_{n+1}$  (conditional on both past and present states) depends only upon the  $G_n$ , not on the sequence of  $G$  that preceded it. Then we have

$$\begin{aligned} &P(G_{n+1} = g_{n+1} | (G_n = g_n) \wedge \dots \wedge (G_0 = g)) \\ &= P(G_{n+1} = g_{n+1} | G_n = g_n), \end{aligned} \quad (8)$$

so that  $\{G_n|n \in N\}_g$  has Markov property. Since  $F_n$  have the same distribution then

$$P(G_m = y | G_{m-1} = z) = P(G_n = y | G_{n-1} = z) \quad (9)$$

for any  $y, z \in G$  and  $m, n \in N$ . So that  $\{G_n|n \in N\}_g$  is homogeneous.

The following Corollary 1 is derived from the Theorem 2 in the paper of Aartset *al.* [10].

**Corollary 1.** *Let  $g \in G$  and the following conditions be satisfied:*

- (a)  $\{G_n|n \in N\}_g$  is monotone
  - (b)  $\{G_n|n \in N\}_g$  is homogeneous
  - (c) for every  $h \in \text{succ}(g)$  there exists at least one accessible optimum.
- Then  $\{G_n|n \in N\}_g$  surely reaches an optimum.

As shown above, we can prove that our AGGAN can converge to the global optimum, which is the closest solution to the minority class distribution with a probability of 1.

Dataset	MNIST			Fashion-MNIST			SVHN		
IR	10	50	100	10	50	100	10	50	100
CN	97.93	94.80	90.27	93.53	86.73	80.00	74.65	58.15	52.00
Os+CN	98.60	96.33	95.53	94.67	92.07	88.13	87.60	66.50	58.90
ACGAN	99.53	98.13	96.60	<b>97.80</b>	96.20	94.13	90.50	74.65	70.55
E-GAN	99.53	98.33	97.00	97.67	96.13	94.40	90.85	74.90	75.50
<b>AGGAN</b>	<b>99.60</b>	<b>98.47</b>	<b>97.53</b>	<b>97.80</b>	<b>96.53</b>	<b>95.07</b>	<b>90.95</b>	<b>81.60</b>	<b>77.70</b>
Dataset	CIFAR			CELEBA			LSUN		
IR	10	50	100	10	50	100	10	50	100
CN	72.00	58.85	57.65	71.10	64.40	60.80	81.20	74.60	52.00
Os+CN	90.60	82.75	80.55	86.70	78.85	77.65	92.05	89.95	64.00
ACGAN	92.00	84.70	83.65	91.65	81.40	78.60	95.05	91.10	89.50
E-GAN	92.10	86.55	85.75	91.65	81.75	80.80	95.20	92.40	91.00
<b>AGGAN</b>	<b>94.00</b>	<b>89.25</b>	<b>86.25</b>	<b>92.05</b>	<b>84.20</b>	<b>81.40</b>	<b>95.35</b>	<b>93.10</b>	<b>91.70</b>

Table 2: Accuracy(%) on imbalanced binary classification with various imbalance ratio.

## 5 Experimental Studies and Discussion

We have used a collection of 6 image datasets for our experiments, namely MNIST [14], Fashion-MNIST [24], SVHN [21], CIFAR-10 [13], CelebA[16], and LSUN[25]. We evaluate our method in two imbalanced environments: two-class and multi-class. In binary classification, because all the selected datasets are multi-class datasets, we randomly select two classes from each dataset. Choose Digit 5 (positive) and Digit 6 (negative) from MNIST, Sandal (positive) and Sneaker (negative) from Fashion-MNIST, and Airplane (positive) and Automobile (negative) from CIFAR-10, Digit 8 (positive) and Digit 9 (negative) from SVHN, Eyeglasses (positive) and No-eyeglasses (negative) from CelebA, and choose Church(positive) and Classroom (negative) from LSUN. In multi-classification, we transform the original balanced training in the same way as the paper of Mullick *et al.* [19]. And we define the Imbalance Ratio (IR) as the number of training samples in the largest class divided by the smallest one.

### 5.1 Implementation Details

In the binary experiment, we have compared AGGAN, against baseline classifier network (CN), Os+CN (training set is random oversampled), ACGAN[27] and Evolutionary-GAN[23] (the version of the discriminator with classifier) to prove the effectiveness of our method. The same network structures are used for these different methods to achieve a fair comparison. In particular, our AGGAN and E-GAN use the same adversarial learning objectives to generate multiple offspring generators in the experiment, including the minimax loss, the modified minimax loss and the least-squares loss. The same evaluation function has been used to measure the individual fitness of generators. In the multi-classification, we compare the proposed method with three the state-of-the-art algorithms which are Class-Balanced[2], DOS[2] and GAMO[19] respectively. All our experiments have been repeated 5 times to mitigate any bias generated due to randomization and the means of the index values are reported.



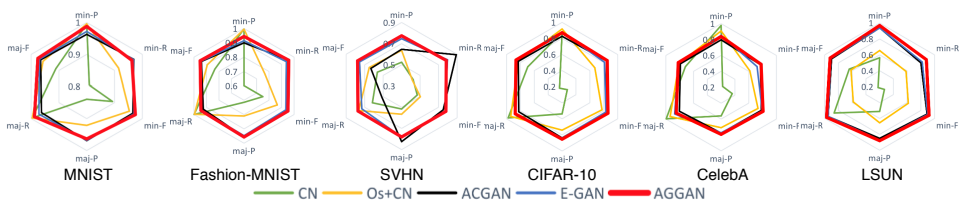


Figure 3: Precision, recall, F1-score of majority and minority classes.

Dataset	CN	ACGAN	EGAN	<b>AGGAN</b>	CB-loss	DOS	GAMO
MNIST	89.87	90.60	91.41	<b>92.41</b>	92.24	90.60	91.01
Fashion-MNIST	79.64	80.51	82.50	<b>83.20</b>	82.84	82.74	83.00

Table 3: Accuracy (%) of imbalanced multi-classification in MNIST and Fashion-MNIST.

## 5.2 Classification Performance

Table 2 shows the accuracy in binary classification under different imbalance ratios. The experimental results of the original imbalance data and the random oversampled data indicate that data imbalance can significantly affect the performance of the classifier, and simply repeating the minority data could not lead to better performance. The experimental results using GANs for oversampling indicate that GANs can mitigate class imbalance problem. Meanwhile, we can observe that when the IR is low, *e.g.* 10, different GANs perform equally well. However, as the degree of imbalance increasing, the advantages of AGGAN become obvious. When the IR reaches 100, the proposed AGGAN outperforms all other GANs significantly. Figure 3 shows the three indicators (precision, recall, and F1-score) of the majority and minority classes in the testing dataset under different methods. We can see that for imbalanced data, the recall of the minority class and the precision of the majority class are low. This is because the classifier will judge the data as the majority class as much as possible to improve the accuracy. The proposed method outperforms the other three methods, which is particularly evident in the recalls of minority class. This provides solid evidence that the ability of AGGAN to reconstruct the distribution of the minority class can make more minority data be classified correctly and improve the performance of the classifier significantly. Table 3 shows the experimental results on multi-classification. It indicates that AGGAN can still perform well on more complex imbalanced datasets and achieve the state-of-the-art level. Compared with ACGAN, E-GAN uses the idea of evolution, and our method further combines the methods of genetic and simulated annealing. The experimental results of these three methods fully indicate the effectiveness of the modules of genetic and simulated annealing in AGGAN.

## 5.3 Visualization

Figure 4 (a) we visualize the features of MNIST and Fashion-MNIST datasets before and after using AGGAN to balance the data, respectively. We obtain features by forwarding images to a classifier pre-trained on the original training set, and features with a specific category in each figure are represented in the same color. The test set of each category is represented by translucent dots of corresponding colors. The first column shows the original

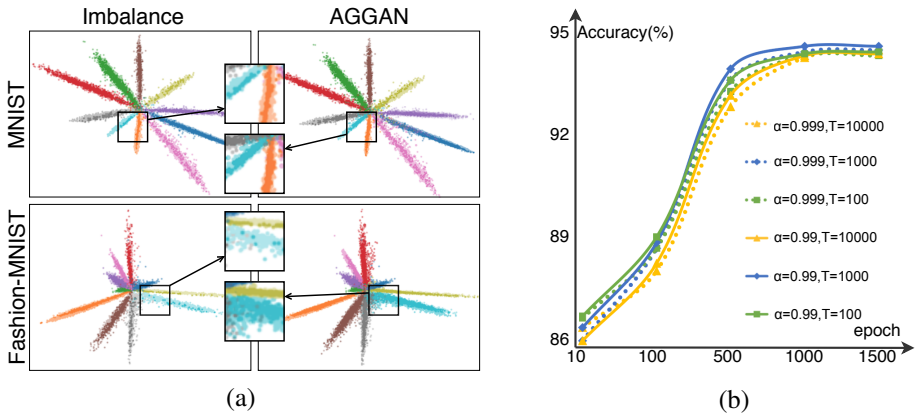


Figure 4: (a) Feature visualization for MNIST and Fashion-MNIST. (b) Simulated annealing hyper-parameter analysis.

imbalanced training datasets, in which the training samples of minority class can only cover part of the minority distribution of the test set (which is balanced). As a result, many samples belong to minority class will be misclassified. However, in the second column of Figure 4 (a), we can see that after using AGGAN for over-sampling, the minority samples in the training set can almost cover the complete distribution of the minority class in the test set, so the performance of the classifier can be significantly improved. These results indicate that AGGAN can learn realistic distribution from scarce minority samples, and in turn prove the superior performance of AGGAN from the perspective of data distribution.

## 5.4 Hyper-parameters analysis

For a better understanding of the role of the initial temperature  $T$  and the annealing coefficient  $\alpha$  proposed in AGGAN, we use the CIFAR-10 dataset with IR 10 to show the accuracy and training epochs of the proposed method in Figure 4 (b). The search for hyper-parameters are  $T \in \{100, 1000, 10000\}$ ,  $\alpha \in \{0.99, 0.999\}$ . We have the following observations that at the beginning of training, smaller value of  $T$  and  $\alpha$  make the model converge faster, while the larger  $T$  and  $\alpha$  (e.g.  $T = 10000, \alpha = 0.999$ ) make the model converge more slowly. However, with an increasing training epoches, the model can achieve higher accuracy finally. These results can indicate that AGGAN is robust to the different values of hyper-parameters.

## 6 Conclusion

In this work, we propose a novel training strategy for GANs, dubbed AGGAN, which aims to reproduce the distributions closest to the ones of the minority classes using limited data samples. Both theoretical analysis and comprehensive experimental studies have shown the robustness and efficacy of our AGGAN.

## References

- [1] Emile Hubertus Leonardus Aarts, Ágoston E Eiben, and KM Van Hee. A general theory of genetic algorithms. 1989.
- [2] Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 770–785. Springer, 2017.
- [3] Martin Arjovsky and L eon Bottou. Towards principled methods for training generative adversarial networks. *Stat*, 1050, 2017.
- [4] Martin Arjovsky, Soumith Chintala, and L eon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.
- [5] Rok Blagus and Lara Lusa. Smote for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1):106, Mar 2013.
- [6] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019.
- [8] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems Man & Cybernetics Part C Applications & Reviews*, 42(4):463–484, 2011.
- [9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [10] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.
- [11] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.
- [12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [14] Yann LeCun, L eon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- [15] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [17] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [18] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.
- [19] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1695–1704, 2019.
- [20] Vaishnavh Nagarajan and J Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems*, pages 5585–5595, 2017.
- [21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [22] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.
- [23] Chaoyue Wang, Chang Xu, Xin Yao, and Dacheng Tao. Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*, 2019.
- [24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [25] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.
- [27] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.