LATENT PRESERVING GENERATIVE ADVERSARIAL NETWORK FOR IMBALANCE CLASSIFICATION

Tanmoy Dam⋆, Md Meftahul Ferdaus†, Mahardhika Pratama†, Sreenatha G. Anavatti⋆, Senthilnath Jayavelu⋆⋆, Hussein Abbass, Fellow, IEEE⋆

⋆ SEIT, University of New South Wales Canberra, Australia
† ATMRI, Nanyang Technological University, Singapore
⋆ ⋆ STEM, University of South Australia, Australia
⋆ ⋆ Institute for InfoComm Research, A*STAR, Singapore

ABSTRACT

Many real-world classification problems have imbalanced frequency of class labels; a well-known issue known as the "class imbalance" problem. Classic classification algorithms tend to be biased towards the majority class, leaving the classifier vulnerable to misclassification of the minority class. While the literature is rich with methods to fix this problem, as the dimensionality of the problem increases, many of these methods do not scale-up and the cost of running them become prohibitive. In this paper, we present an end-to-end deep generative classifier. We propose a domain-constraint autoencoder to preserve the latent-space as prior for a generator, which is then used to play an adversarial game with two other deep networks, a discriminator and a classifier. Extensive experiments are carried out on three different multi-class imbalanced problems and a comparison with state-of-the-art methods. Experimental results confirmed the superiority of our method over popular algorithms in handling high-dimensional imbalanced classification problems. Our code is available on https://github.com/TanmDL/SLPFL-GAN

Index Terms— class imbalance, adversarial learning, oversampling techniques

1. INTRODUCTION

A deep oversampling framework (DOS) was proposed in [1] to fulfill the end-to-end requirement of deep learning algorithms for handling imbalance classification problem. A limitation of the DOS lies in its dependency on the class-wise neighborhood sizes, which are determined by costly parameter tuning. The generative adversarial network (GAN) has gained popularity due to its unique capability in generating synthetic but realistic samples. GAN relies on random noises that may yield a highly entangled process and disruptions of feature orientations. A two-stage framework called BAGAN [2] is proposed by combining autoencoder (AE) with conditional GAN (cGAN) [3]. This approach often leads to boundary distortion as witnessed in [4–7]. In recent time, a deep generative classifier (DGC) is developed in [8] for solving unstable prediction in the imbalance classification problem. A model is perturbed by replacing the fixed values of latent variables with a probability distribution over possible values, whereas the data are perturbed by generating feature/label. To mitigate the majority class influence on the classifier, probabilistic latent codes are over-sampled at different fractions. However, the learned probabilistic latent codes can’t guarantee better separation at the encoded manifold that may adversely influence the classifier’s performance. Another oversampling method known as Generative adversarial minority oversampling” (GAMO) is proposed in [4]. In GAMO, the mixture of generators forcefully generates samples within a specific minority class distribution by utilizing real instances from that minority class. However, their generators are limited by generating class instances instead of generating the real data distribution. To determine class instances from the real data distribution, an adversarial game is played between the mixture of generators, a discriminator and a classifier. Finally, the baseline classifier performance has been improved by considering real distribution and minority class-generator samples. In addition, deep generative classifiers, cross entropy loss [8] or mean-squared loss [4] are commonly used to update the classifier. These loss functions may establish hard partitions between majority and minority classes, causing over-fitting. A classifier is updated twice in [4, 8] based on [9, 10] ideas to reduce the effect of majority class samples on the decision boundary. A similar approach is also followed in this paper. In addition, mixtures of minority class-specific generators are replaced with a single generator only. This is done by deploying a domain-constrained AE to learn the class-specific latent code, preserved and used as a prior for the generative network. Besides, in GAMO, the minority samples are generated only from the feature space while ignoring the possibility of an adversarial data space oversampling approach. Thus, the only three-player strategy of adver-
sarial minority oversampling of GAMO is extended here to two different three-player strategies: 1) adversarial minority oversampling (AMO); 2) adversarial data space oversampling (ADSO). The main contributions of this paper are summarized:

- We define a joint learning framework to preserve the latent space in a low dimensional manifold by utilizing a supervised autoencoder (AE). The learned latent space works as a prior for a single generator, which engages in an adversarial game with a classifier and a discriminator. We leveraged all two possible strategies among three players game to improve the classifier’s performance. First strategy is related to ADSO and the last strategy is based on AMO.

- A set of experiments have been carried out under different imbalance ratios, where the experimental results support that an ADSO-based classifier performs better than the AMO-based approach and state-of-the-art baselines, and it does so with significant margins.

2. PROBLEM FORMULATIONS

Let us assume a set of $N$ samples $\{x_i, y_i\}_{i=1}^N \in (X_{org}, Y_{org})$, a multi-class imbalance dataset distributed over $C$ classes. Here, $x_i \in \mathbb{R}^d$ is an $i$-th input image with its corresponding target class $y_i$. Without loss of generalisation, the class distribution follows $p_{maj} \geq p_2 \geq \ldots p_l \geq \ldots \geq p_{min}$, $N = \sum_{l=1}^C p_l$, in which $p_{maj}$ and $p_{min}$ denote the majority class and the minority class, respectively. The relationship between the majority class and minority class(es) is set as $p_{maj} \geq 50 \times p_{min}$ following the problem setting of [4, 6, 7]. The main objective is to design a deep neural network estimating the underlying data distribution of $C$ classes, thereby producing robust decision boundaries.

The network structure of our approach consists of three parts: supervised latent preserving prior learning (SLPPL), ADSO and AMO. The adversarial game is played among preserved latent prior generator($G_\omega$), a discriminator($Dis_\xi$) and a classifier($Q_\varrho$) to improve the classification performance. In SLPPL, the $Enc_\varrho$ encodes the original class instances $(x_i \in X_{org})$ into a lower-dimensional latent space $(z_i \in Z)$ and $Dec_\varrho$ takes the encoded latent space $(z_i)$ to produce the reconstructed original data $(\hat{x}_i \in X_{org})$. The $Enc_\varrho$ learns class distribution $(Enc_\varrho(y_i|x_i))$ for $i$-th class instance. After learning the latent space $(z_i)$ from data directly, a Gaussian multivariate normal distribution is constructed as a prior for $G_\omega$. In the adversarial game, we introduce mainly two strategies, ADSO and AMO, among the pretrained prior $G_\omega$, a $Dis_\xi$ and a $Q_\varrho$.

3. OUR APPROACH

3.1. SLPPL

High dimensional data always retain their characteristics in a low dimensional encoded manifold that inspired us to design a latent prior for each class distribution [11]. A low dimensional manifold learning approach attains optimum performance by freely moving the latent space when the distribution of data is uniform in nature [11]. If data are not distributed uniformly, the majority class-driven latent space always dominates minority classes. To alleviate a biased prediction issue favouring the majority class, we use $(X_{org}, Y_{org})$ class distribution learning and reconstruction learning to bind the latent space as much as feasible. Label information in SLPPL under the deterministic autoencoder (AE) paradigm yields the bounded latent space. This technique reduces latent space stochasticity, unlike VAE-based priors [12]. Finally, the resulting latent space is a smooth manifold. The AE objective is to minimize the reconstruction loss ($L_{rec}$) between input data $(X_{org})$ and decoded output $(Dec_\varrho(Enc_\varrho(X_{org})))$:

$$L_{rec}(X_{org}, Dec_\varrho(Enc_\varrho(X_{org}))) = \mathbb{E}_{x_i \in X_{org}} ||x_i - Dec_\varrho(Enc_\varrho(x_i))||_2 \quad (1)$$

Similarly, for the bounded latent space, the encoder network $(Enc_\varrho)$ estimates the class distribution $(Y_{org})$.

$$L_{bce}(Y_{org}, (Enc_\varrho(X_{org}))) = \mathbb{E}_{y_i \in Y_{org}} y_i log(Enc_\varrho(x_i)) \quad (2)$$

Finally, merging two losses, the final loss function for SLPPL can be minimized with respect to two networks parameter.

$$\min_{\varrho, \phi} L_{SLPPL} = L_{rec} + L_{bce} \quad (3)$$

The proposed SLPPL is able to preserve stable network parameters $(\varphi, \theta)$, and to maintain stable class distributions in the low-dimensional manifold where (3) is solved using the ADAM optimiser [7]. The $i$-th sample is easily encoded in the latent space afterward as $z_i \in Z = Enc_\varrho(x_i)$ and preserves a multivariate normal distribution (MND) [11] for the $i$-th class. The MND definition for the $i$-th class is $z_i \in \mathcal{N} (\mu_i, \sigma_i^2)$ where $\mu_i \in \mathbb{R}^d$ and $\sigma_i \in \mathbb{R}^{d \times q}$ are the mean and variance of the latent space respectively. The prior $(\mu_i)$ preserving latent space is applied to improve image generation through adversarial game between $Q_\varrho$ and $Dis_\xi$ through two adversarial strategies: ADSO and AMO. In both the oversampling cases, the $G_\omega$ network structure is same as $Dec_\varrho$. Hence, the initialisation of $G_\omega$ network weights is taken from pretrained $Dec_\varrho$, whereas the structure of $Dis_\xi$ and $Q_\varrho$ are almost similar to $Enc_\varrho$ network but the last layer of $Dis_\xi$ gives a single output. Besides, the learned feature layers of $Enc_\varrho$ are used to initialise the weights of both the $Dis_\xi$ and $Q_\varrho$. However, to reduce the over-fitting effect at $Q_\varrho$, we apply dropout after activation function [13].
3.2. ADSO & AMO

In ADSO, the minority class(es) is repeated to form a balanced data distribution \((X_{bal}, Y_{bal})\) in the data space. An adversarial game is played among \(G_{\omega}, \text{Dis}_\xi\) and \(Q_\varrho\) afterward, where the \(G_{\omega}\) network is updated by fooling only the discriminator \(\text{Dis}_\xi\) but by favouring the classifier \(Q_\varrho\). The generator network \(G_{\omega}\) aims to generate samples to be classified by \(Q_\varrho\) as the same class, and thus, the generated samples assign real scores while updating the generator \(G_{\omega}\) through the classifier \(Q_\varrho\). The discriminator \(\text{Dis}_\xi\) enforces the generator \(G_{\omega}\) to follow the real data distribution. While updating the classifier \(Q_\varrho\) through the generator \(G_{\omega}\), generated samples assign a fake score to the classifier \(Q_\varrho\).

We can formulate the following optimisation problem for three players adversarial game among \(G_{\omega}, \text{Dis}_\xi\) and \(Q_\varrho\):

\[
\min_{\omega, \varrho} \max_\xi L_{\text{ADSO}}(G_{\omega}, \text{Dis}_\xi, Q_\varrho) \tag{4}
\]

The ADSO loss can be expressed as:

\[
L_{\text{ADSO}} = L_{G_{\text{ADSO}}} + L_{\text{Dis}_\xi} + L_{Q_\varrho_{\text{ADSO}}}, \tag{5}
\]

\[
L_{G_{\text{ADSO}}} = \mathbb{E}_{G(z_i) \in Z}(f(1 - \text{Dis}_\xi(G(z_i)))+ \mathbb{E}_{G(z_i) \in Z}(y_i \log(Q_\varrho(G_{\omega}(z_i)))) \tag{6}
\]

\[
L_{Q_\varrho_{\text{ADSO}}} = \mathbb{E}_{y_i \in Y_{bal}} y_i \log(Q_\varrho(x_i)) + \mathbb{E}_{G(z_i) \in Z}(y_i \log(1 - Q_\varrho(G_{\omega}(z_i)))) \tag{7}
\]

\[
L_{\text{Dis}_\xi} = \mathbb{E}_{x_i \in X_{bal}} (f(\text{Dis}_\xi(x_i)))+ \mathbb{E}_{G(z_i) \in Z} f(1 - \text{Dis}_\xi(G_{\omega}(z_i))) \tag{8}
\]

In AMO, generator network \(G_{\omega}\) aims to generate samples to be classified by \(Q_\varrho\) as the same class, and thus, the generated samples assign real scores while updating the generator \(G_{\omega}\) through the classifier \(Q_\varrho\). To mitigate, majority class biases at \(Q_\varrho\), is updated through the real samples as well as minority-class generated samples. \(\text{Dis}_\xi\) network forces \(G_{\omega}\) to learn the real-data distributions. The following optimisation can be formulated by playing three players game among \(G_{\omega}, \text{Dis}_\xi\) and \(Q_\varrho\):

\[
\min_{\omega, \varrho} \max_\xi L_{\text{AMO}}(G_{\omega}, \text{Dis}_\xi, Q_\varrho) \tag{9}
\]

The AMO loss can be expressed as:

\[
L_{\text{AMO}} = L_{G_{\text{AMO}}} + L_{\text{Dis}_\xi} + L_{Q_\varrho_{\text{AMO}}}, \tag{10}
\]

\[
L_{G_{\text{AMO}}} = \mathbb{E}_{G(z_i) \in Z}(f(1 - \text{Dis}_\xi(G(z_i)))+ \mathbb{E}_{G(z_i) \in Z}(y_i \log(1 - Q_\varrho(G_{\omega}(z_i)))) \tag{11}
\]

\[
L_{\text{Dis}_\xi} = \mathbb{E}_{x_i \in X_{org}} (f(\text{Dis}_\xi(x_i)))+ \mathbb{E}_{G(z_i) \in Z} f(1 - \text{Dis}_\xi(G_{\omega}(z_i))) \tag{12}
\]

Cross-entropy (CE) loss and complementary CE (CCE) loss are represented by the expressions \(\log Q(.)\) and \(\log(1 - Q(.))\), respectively. For both the cases (AMO, ADSO), the functional operator \(f(.)\) selects different GANs types. For vanilla [14] and Wasserstein GANs (WGANs) [15], \(f\) is represented as \(f(x) = \log x\) and \(f(x) = x\) respectively. We follow WGAN’s zero center (0)-gradient penalty (0-gp) for all the GAN strategies [15].

4. EXPERIMENTS

4.1. Datasets

Two single-channel (MNIST [16], and FMNIST [17]) and a three-channel (CelebA [18]) image sets are used here. Properties of these datasets are tabulated in Table 1 where IR indicates the imbalance ratio. Five metrics are used here to measure the imbalance classification performance: 1) average class specific accuracy (ACSA); 2) macro-averaged F-measure \((F_{\text{macro}})\); 3) macro-averaged geometric mean \((G_{\text{mean}})\); 4) precision of majority class \((P_{\text{maj}})\); and 5) recall of minority class \((R_{\text{min}})\).

Our proposed method is compared against seven different baselines namely BAGAN [2], DFBS [19], GAMO [4], max-margin deep (mmd) generative models [20], BayesCNN (BCNN) [21], DGC [8], and random oversampling (RO) plus baseline \(Q_\varrho\). Results of all above-mentioned baselines are adopted from [8]. For our two proposed models, the size of latent space \((z_i \in R^n)\) is set to \(q = 64\) for the MNIST and FMNIST datasets [8], assigned as \(q = 128\) for the CelebA dataset [8].

4.2. Numerical Results

The overall classification performances in terms of average of ACSA, \(F_{\text{macro}}\), and \(G_{\text{macro}}\) are reported in Table 2. The ADSO-based strategy we suggest is the one that performs the best overall for handling imbalance classification. In contrast with the ADSO, some limitations of baselines for getting comparative poorer performance are as follows: DFBS can not create sufficient margins among classes. GAMO utilizes computationally expensive MSE loss that requires real samples in each generator. The mode collapse problem may appear in BAGAN due to the initialization mechanism of subsequent GAN. BCNN uses model perturbation, however few
### Table 1. The detailed description of Experimental datasets

| Datasets | Data Dimensions | IR | Classes | Training Set                                                                 | Testing Set                                                                 |
|----------|-----------------|----|---------|-------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| MNIST    | $28 \times 28 \times 1$ | 100 | 10      | [4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40]                           | [980, 1135, 1032, 1010, 982, 892, 985, 1028, 974, 1009] |
| FMNIST   | $28 \times 28 \times 1$ | 100 | 10      | [4000, 2000, 1000, 750, 500, 350, 200, 100, 60, 40]                           | [1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000]                               |
| CelebA   | $64 \times 64 \times 3$ | 100 | 5       | [15000, 1500, 750, 300, 15]                                                  | [2660, 5422, 412, 3428, 535]                                                  |

### Table 2. Overall classification performance on various datasets

| Methods   | MNIST | FMNIST | CelebA |
|-----------|-------|--------|--------|
|           | AC$\alpha$ | $F_{macro}$ | $G_{macro}$ | AC$\alpha$ | $F_{macro}$ | $G_{macro}$ | AC$\alpha$ | $F_{macro}$ | $G_{macro}$ |
| BAGAN     | 0.8848 | 0.8785 | 0.9295 | 0.7814 | 0.7610 | 0.8546 | 0.5972 | 0.5152 | 0.6554 |
| DFBS      | 0.7812 | 0.7838 | 0.8083 | 0.5135 | 0.4620 | 0.6382 | 0.2109 | 0.1335 | 0.2664 |
| GAMO      | 0.8826 | 0.8794 | 0.9308 | 0.7929 | 0.7880 | 0.8740 | 0.6409 | 0.5903 | 0.7472 |
| BCNN      | 0.9158 | 0.9141 | 0.9512 | 0.7934 | 0.7835 | 0.8701 | 0.5517 | 0.4936 | 0.6534 |
| mmD       | 0.9066 | 0.9039 | 0.9449 | 0.8091 | 0.7984 | 0.8796 | 0.3760 | 0.0618 | 0.3754 |
| DGC       | 0.9480 | 0.9474 | 0.9704 | 0.8364 | 0.8314 | 0.9010 | 0.6755 | 0.6454 | 0.7779 |
| RO+$Q_\omega$ | 0.9339 | 0.9325 | 0.9619 | 0.8450 | 0.8436 | 0.9089 | 0.7078 | 0.6663 | 0.7997 |
| AMO       | 0.9403 | 0.9386 | 0.9656 | 0.8428 | 0.8378 | 0.9046 | 0.6702 | 0.6210 | 0.7628 |
| ADSO      | 0.9613 | 0.9609 | 0.9781 | 0.8675 | 0.8648 | 0.9221 | 0.7595 | 0.7217 | 0.8359 |

### Table 3. Classification performance on the largest ($P_{maj}$) and smallest class ($R_{min}$)

| Methods   | MNIST | FMNIST | CelebA |
|-----------|-------|--------|--------|
|           | $R_{min}$ | $P_{maj}$ | $R_{min}$ | $P_{maj}$ | $R_{min}$ | $P_{maj}$ | $R_{min}$ | $P_{maj}$ | $R_{min}$ | $P_{maj}$ |
| BAGAN     | 0.5354 | 0.8541 | 0.7306 | 0.5709 | 0.0192 | 0.5064 |
| DFBS      | 0.5946 | 0.5118 | 0.4412 | 0.3395 | 0.0522 | 0.2174 |
| GAMO      | 0.6394 | 0.8812 | 0.7928 | 0.6165 | 0.2302 | 0.6687 |
| BCNN      | 0.7578 | 0.8896 | 0.8474 | 0.6022 | 0.1063 | 0.5225 |
| mmD       | 0.6525 | 0.8459 | 0.8160 | 0.5942 | 0.0006 | 0.4110 |
| DGC       | 0.8276 | 0.9270 | 0.8864 | 0.6900 | 0.2987 | 0.7603 |
| RO+$Q_\omega$ | 0.7393 | 0.8761 | 0.9130 | 0.7598 | 0.2783 | 0.7483 |
| AMO       | 0.7365 | 0.9055 | 0.9170 | 0.6752 | 0.2405 | 0.5785 |
| ADSO      | 0.8840 | 0.9348 | 0.9510 | 0.7885 | 0.3223 | 0.8132 |

Minority class occurrences may not be enough to train the model. mmD employs the discriminative classifier, restricting their performance in imbalanced datasets. Both data and model perturbation help DGC outperform baselines. Random oversampling of original data may improve the performance of the $Q_\omega$, but, cannot guarantee its consistency in all cases as observed in Table 2. The full-optimal performance of $Q_\omega$ is observed in ADSO where $Q_\omega$ forces the parameters of $G_{\omega}$ to generate relevant samples to the $Q_\omega$.

A recall on minority class is expected from the $Q_\omega$ while maintaining a high precision on majority class. Recall of the smallest class ($R_{min}$) and precision of the largest class ($P_{maj}$) are listed in Table 3, where it is observed that the ADSO outperformed all the baselines including RO+$Q_\omega$ and our AMO strategy. Even if we preserve the prior for the $G_{\omega}$, $Q_\omega$ has influenced it to generate majority classes, i.e., AMO can’t beat ADSO. In contrast, under three-player GAN settings, the proposed ADSO-based $G_{\omega}$ tries to fool itself by generating a subset of each class sample that is relevant to $Q_\omega$.

### 5. CONCLUSIONS

We propose a latent preserving deep generative model for handling imbalanced classification problems. The class constraints AE is used to preserve the latent space, utilized as a prior for $G_{\omega}$. By playing two adversarial games among the latent space preserved across $G_{\omega}$, a Dist$_{\omega}$, and a $Q_\omega$, improvement in the $Q_\omega$’s performance is witnessed. From the experimental results on all three datasets, our ADSO-based strategy performed better than all the baselines. Applications to the real world datasets like semiconductor fault detection, medical diagnosis, hyper-spectral data [22] etc., will be included in our future work.

### Acknowledgements

T. Dam acknowledges UIPA funding from UNSW Canberra. S. Jayavelu and Md M. Ferdaus acknowledges funding from the Accelerated Materials Development for Manufacturing Program at A*STAR via the AME Programmatic Fund by the Agency for Science, Technology and Research under Grant No. A1898b0043.
6. REFERENCES

[1] 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*. Springer, 2017, pp. 770–785.

[2] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, and Cristiano Malossi, “Bagan: Data augmentation with balancing gan,” *arXiv preprint arXiv:1803.09655*, 2018.

[3] Mehdil Mirza and Simon Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.

[4] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das, “Generative adversarial minority oversampling,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1695–1704.

[5] Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry, “A classification-based study of covariate shift in gan distributions,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 4480–4489.

[6] Tanmoy Dam, Sreenatha G Anavatti, and Hussein A Abbass, “Mixture of spectral generative adversarial networks for imbalanced hyperspectral image classification,” *IEEE Geoscience and Remote Sensing Letters*, 2020.

[7] Tanmoy Dam, Md Meftahul Ferdaus, Sreenatha G Anavatti, Senthilnath Jayavelu, and Hussein A Abbass, “Does adversarial oversampling help us?”, in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2970–2973.

[8] Xinyue Wang, Yilin Lyu, and Liping Jing, “Deep generative model for robust imbalance classification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14124–14133.

[9] Robust Statistics, “The approach based on influence functions,” *New York*, 1986.

[10] Pang Wei Koh and Percy Liang, “Understanding black-box predictions via influence functions,” in *International Conference on Machine Learning*. PMLR, 2017, pp. 1885–1894.

[11] Partha Ghosh, Mehdil SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf, “From variational to deterministic autoencoders,” in *International Conference on Learning Representations*, 2019.

[12] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.

[13] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdil Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.

[15] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin, “Which training methods for gans do actually converge?,” in *International conference on machine learning*. PMLR, 2018, pp. 3481–3490.

[16] Yann LeCun, “The mnist database of handwritten digits,” http://yann.lecun.com/exdb/mnist/, 1998.

[17] Han Xiao, Kashif Rasul, and Roland Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[18] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang, “Deep learning face attributes in the wild,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[19] Yi-Hsun Liu, Chien-Liang Liu, and Shin-Mu Tseng, “Deep discriminative features learning and sampling for imbalanced data problem,” in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1146–1151.

[20] Chongxuan Li, Jun Zhu, and Bo Zhang, “Max-margin deep generative models for (semi-) supervised learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 11, pp. 2762–2775, 2017.

[21] Kumar Shridhar, Felix Laumann, and Marcus Luwicki, “A comprehensive guide to bayesian convolutional neural network with variational inference,” *arXiv preprint arXiv:1901.02731*, 2019.

[22] Tanmoy Dam, Nidhi Swami, Sreenatha G Anavatti, and Hussein A Abbass, “Multi-fake evolutionary generative adversarial networks for imbalance hyperspectral image classification,” *arXiv preprint arXiv:2111.04019*, 2021.