Class Balancing GAN with a Classifier in the Loop

Harsh Rangwani[1] **Konda Reddy Mopuri**[2] **R. Venkatesh Babu**[1]

[1]Indian Institute of Science, Bengaluru
[2]Indian Institute of Technology, Tirupati

Abstract

Generative Adversarial Networks (GANs) have swiftly evolved to imitate increasingly complex image distributions. However, majority of the developments focus on performance of GANs on balanced datasets. We find that the existing GANs and their training regimes which work well on balanced datasets fail to be effective in case of imbalanced (i.e. long-tailed) datasets. In this work we introduce a novel theoretically motivated Class Balancing regularizer for training GANs. Our regularizer makes use of the knowledge from a pre-trained classifier to ensure balanced learning of all the classes in the dataset. This is achieved via modelling the effective class frequency based on the exponential forgetting observed in neural networks and encouraging the GAN to focus on underrepresented classes. We demonstrate the utility of our regularizer in learning representations for long-tailed distributions via achieving better performance than existing approaches over multiple datasets. Specifically, when applied to an unconditional GAN, it improves the FID from 13.03 to 9.01 on the long-tailed iNaturalist-2019 dataset.

1 INTRODUCTION

Image Generation witnessed unprecedented success in recent years following the invention of Generative Adversarial Networks (GANs) by Goodfellow et al. [2014]. GANs have improved significantly over time with the introduction of better architectures [Gulrajani et al., 2017, Radford et al., 2015], formulation of superior objective functions [Jolicoeur-Martineau, 2018, Arjovsky et al., 2017], and regularization techniques [Miyato et al., 2018]. An important breakthrough for GANs has been the ability to effectively class conditioning for synthesizing images [Mirza and Osindero, 2014, Miyato and Koyama, 2018]. Conditional GANs have been shown to scale to large datasets such as ImageNet [Deng et al., 2009] with 1000 classes [Miyato and Koyama, 2018].

One of the major issues with unconditional GANs has been their inability to produce balanced distributions over all the classes present in the dataset. This is seen as problem of missing modes in the generated distribution. A version of the missing modes problem, known as the 'covariate shift' problem was studied by Santurkar et al. [2018]. One possible reason is the absence of knowledge about the class distribution $P(Y|X)$[1] of the generated samples during training. Conditional GANs on the other hand, do not suffer from this issue since the class labels Y are supplied to the GAN during training. However, it has been recently found by Ravuri and Vinyals [2019] that despite being able to do well on metrics such as Inception Score (IS) [Salimans et al., 2016] and Frèchet Inception Distance (FID) [Heusel et al., 2017], the samples generated from the state-of-the-art conditional GANs lack the diversity in comparison to the underlying training datasets. Further, we observe that, although conditional GANs work well in balanced case, they suffer performance degradation in the imbalanced case (Table-1).

In order to address these shortcomings, we propose a novel method of inducing the information about the class distribution. We estimate the class distribution $P(Y|X)$ of generated samples in the GAN framework using a pre-trained classifier. The regularizer utilizes the estimated class distribution to penalize excessive generation of samples from the majority classes, thereby enforcing the GAN to also generate samples from minority classes. Our regularizer involves a novel method of modelling the forgetting of samples by GANs, based on the exponential forgetting observed in neural networks [Kirkpatrick et al., 2017]. We show the implications of our regularizer by a theoretical upper bound in Section 3.

[1]Here Y represents labels and X represents data.

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:1618–1627.

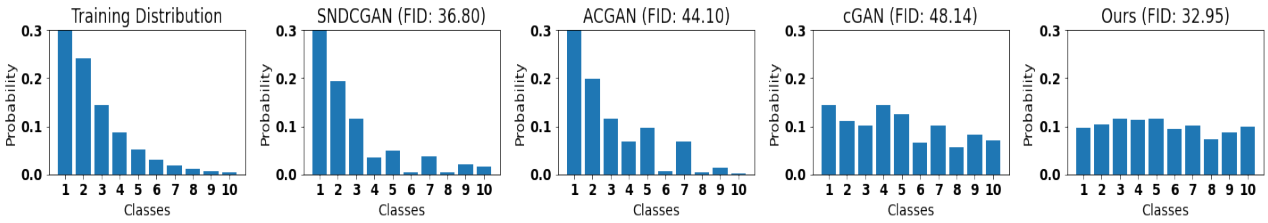


Figure 1: Distribution of classes and corresponding FID scores on long-tailed CIFAR-10. SNDCGAN and ACGAN tend to produce arbitrary distributions which are biased towards majority classes, whereas cGAN samples suffer in quality with high FID. Our method achieves low FID and a balanced distribution at the same time.

We also experimentally demonstrate the effectiveness of the proposed class balancing regularizer in the scenario of training GANs for image generation on long-tailed datasets, including the large scale iNaturalist-2019 [iNaturalist, 2019] dataset. Generally, even in the long-tailed distribution tasks, the test set is balanced despite the imbalance in the training set. This is because it is important to develop Machine Learning systems that generalize well across all the support regions of the data distribution, avoiding undesired over-fitting to the majority (or head) classes.

In summary, our contributions can be listed as follows:

- We propose a 'class-balancing' regularizer that makes use of the statistic $P(Y|X)$ of generated samples to promote uniformity while sampling from an unconditional GAN. The effect of our regularizer is depicted both theoretically (Section 3) and empirically (Section 4).
- We show that our regularizer enables GANs to learn uniformly across classes even when the training distribution is long-tailed. We observe consistent gains in FID and accuracy of a classifier trained on the generated samples.
- Our method is able to scale to large and naturally occurring datasets such as iNaturalist-2019 and, achieves state-of-the-art FID score of 9.01.

2 BACKGROUND

2.1 GENERATIVE ADVERSARIAL NETWORKS (GANS)

Generative Adversarial Network (GAN) is a two player game in which the discriminator network D tries to classify images into two classes: real and fake. The generator network G tries to generate images (transforming a noise vector $z \sim P_z$) which fool the discriminator D into classifying them as real. The game can be formulated as the following mathematical objective:

$$\min_G \max_D E_{x\sim P_r}[\log(D(x))] + E_{z\sim P_z}[\log(1 - D(G(z))] \quad (1)$$

The exact inner optimization for training D is computationally prohibitive in large networks; hence the GAN is trained through alternate minimization of loss functions. Multiple loss functions have been proposed for stabilizing GAN training. In our work we use the relativistic loss function [Jolicoeur-Martineau, 2018] which is formulated as:

$$L_D^{rel} = -E_{(x,z)\sim(P_r,P_z)}[\log(\sigma(D(x) - D(G(z)))] \quad (2)$$

$$L_G^{rel} = -E_{(x,z)\sim(P_r,P_z)}[\log(\sigma(D(G(z)) - D(x))] \quad (3)$$

Issue in the long-tailed scenario: This unconditional GAN formulation does not use any class information $P(Y|X)$ about the images and tends to produce different number of samples from different classes [Santurkar et al., 2018]. In other words, the generated distribution is not balanced (uniform) across different classes. This issue is more severe when the training data is long-tailed, where the GAN might completely ignore learning some (minority) classes (as shown in the SNDCGAN distribution of Figure 1).

2.2 CONDITIONAL GAN

The conditional GAN [Mirza and Osindero, 2014] generates images associated with a given input label y using the following objective:

$$\min_G \max_D E_{x\sim P_r}[\log(D(x|y))] + E_{z\sim P_z}[\log(1 - D(G(z|y))] \quad (4)$$

The Auxillary Classifier GAN (ACGAN) [Odena et al., 2017] uses an auxiliary classifier for classification along with a discriminator to enforce high confidence samples from the conditioned class y. Whereas cGAN with projection [Miyato and Koyama, 2018] uses Conditional Batch Norm [De Vries et al., 2017] in the generator and a projection step in the discriminator to provide class information to the GAN. We refer to this method as cGAN in the subsequent sections.

Issue with Conditional GAN in Long-tailed Setting: The objective in eq.(4) can be seen as learning a different $G(z|y)$

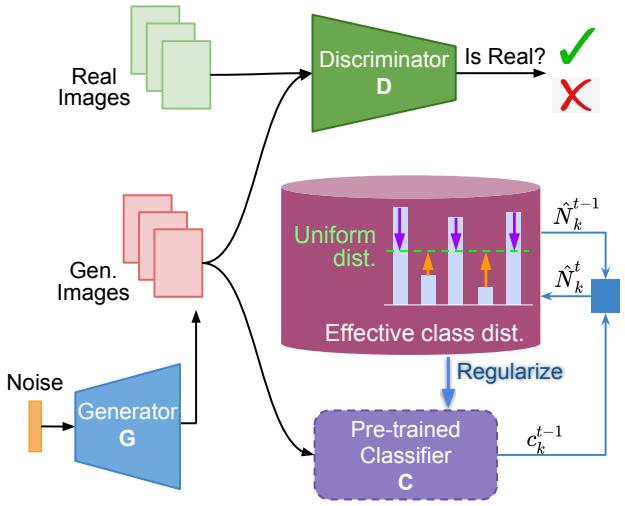


Figure 2: **Class Balancing Regularizer** aims to help GANs generate balanced distribution of samples across classes even when they are trained on imbalanced datasets. The method achieves this by keeping an estimate of effective class distribution of generated images using a pre-trained classifier. The GAN is then incentivized to generate images from underrepresented classes, which moves the GAN towards generating uniform distribution.

and $D(x|y)$ for each of the K classes. In this case the tail classes with fewer samples can suffer from poor learning of $G(z|y)$ as there are very few samples for learning. However, in practice there is parameter sharing among different class generators, yet there are class specific parameters present in the form of Conditional BatchNorm. We find that performance of conditional GANs degrades more in comparison to unconditional GANs in the long-tailed scenario (observed in Table 1).

3 METHOD

In our method we use a pretrained classifier (C) to provide feedback to the generator about the label distribution $P(Y|X)$ through the proposed regularizer. One can train the classifier (C) before the GAN, on the underlying long-tailed dataset. However, if the full set of labels are not available, we show (Section 4.3) that a limited set of labeled data is sufficient for training the classifier. The proposed regularizer term is then added to the generator loss and trained using backpropogation. We first describe the method of modelling the class distribution in Section 3.1. Exact formulation of the regularizer and its theoretical properties are described in Section 3.2. The overview of our method is presented in Figure 2.

3.1 CLASS STATISTICS FOR GAN

GAN is a dynamic system in which the generator G has to continuously adapt itself in a way that it is able to fool the discriminator D. During the training, discriminator D updates itself, which changes the objective for the generator G. This change in objective can be seen as learning of different tasks for the generator G. In this context, we draw motivation from the seminal work on catastrophic forgetting in neural networks [Kirkpatrick et al., 2017] which shows that a neural network trained using SGD suffers from exponential forgetting of earlier tasks when trained on a new task. Based on this, we define *effective class frequency* $\hat{N}_k^t$ of class k at a training cycle t as:

$$\hat{N}_k^t = (1-\alpha)\hat{N_k}^{t-1} + \beta c_k^{t-1} \tag{5}$$

Here c_k^{t-1} is the number of samples of class k produced by the GAN in cycle $(t-1)$ and $\hat{N_k}^0$ is initialized to a constant for all k classes. The class to which the generated sample belongs to is determined by a pretrained classifier C. We find that using exponential decay, using either $(\beta = 1)$ or a convex combination $(\beta = \alpha)$ is sufficient for all the experiments. Although D gets updated continuously, the update is slow and requires some iterations to change the form of D. Hence we update the statistics after certain number of iterations which compose a cycle. Here α is the exponential forgetting factor which is set to 0.5 as default in our experiments. We normalize the $\hat{N}_k^t$ to obtain discrete *effective class distribution* N_k^t:

$$N_k^t = \frac{\hat{N}_k^t}{\sum_k \hat{N}_k^t} \tag{6}$$

3.2 REGULARIZER FORMULATION

The regularizer objective is defined as the minimization of the term (L_{reg}) below:

$$\min_{\hat{p}} \sum_k \frac{\hat{p}_k \log(\hat{p}_k)}{N_k^t} \tag{7}$$

where $\hat{p} = \sum_{i=1}^{n} \frac{C(G(z_i))}{n}$ and $z_i \sim P_z$. In other words, $\hat{p}$ is the average softmax vector (obtained from the classifier C) over the batch of n samples and $\hat{p_k}$ is its k^{th} component corresponding to class k. If the classifier C recognizes the samples confidently with probability ≈ 1, $\hat{p}_k$ can be seen as the approximation to the ratio of the number of samples that belong to class k to the total number of samples in the batch n. N_k^t in the regularizer term is obtained through the update rule in Section 3.1 and is a constant during backpropagation. The regularizer objective in Eq. (7) when multiplied with a negative unity, can also be interpreted as maximization of the weighted entropy computed over the batch.

Proposition 1: The minimization of the proposed objective in (7) leads to the following bound on $\hat{p_k}$:

$$\hat{p_k} \leq e^{-K(log(K)-1)\frac{N_k^t}{\sum_k N_k^t}-1} \tag{8}$$

where K is the number of distinct class labels produced by the classifier C.

Proof:

$$\min_{\hat{p}} \sum_k \frac{\hat{p_k}\log(\hat{p_k})}{N_k^t} \tag{9}$$

Introducing the probability constraint and the Lagrange multiplier λ:

$$L(\hat{p},\lambda) = \sum_k \frac{\hat{p_k}\log(\hat{p_k})}{N_k^t} - \lambda(\sum \hat{p_k} - 1) \tag{10}$$

On solving the equations obtained by setting $\frac{\partial L}{\partial \hat{p_k}} = 0$:

$$\frac{1}{N_k^t} + \frac{\log(\hat{p_k})}{N_k^t} - \lambda = 0 \implies \hat{p_k} = e^{\lambda N_k^t - 1} \tag{11}$$

Using the constraint $\frac{\partial L}{\partial \lambda} = 0$ we get:

$$\sum_k \hat{p_k} = 1 \implies \sum_k e^{\lambda N_k^t - 1} = 1 \implies \sum_k e^{\lambda N_k^t} = e \tag{12}$$

Now we normalize both sides by K, the number of distinct labels produced by classifier and apply Jensen's inequality for concave function $\psi(\frac{\sum a_i x_i}{\sum a_i}) \geq \frac{\sum a_i \psi(x_i)}{\sum a_i}$ and take ψ as log function:

$$\frac{e}{K} = \sum_k \frac{e^{\lambda N_k^t}}{K} \implies \log(\frac{e}{K}) = \log(\sum_k \frac{e^{\lambda N_k^t}}{K}) \geq \sum_k \frac{\lambda N_k^t}{K} \tag{13}$$

On substituting the value of λ in inequality from Eq. 11:

$$K(1-\log(K)) \geq \lambda \sum_k N_k^t \implies \tag{14}$$

$$K(1-\log(K)) \geq (\sum_k N_k^t)\frac{1+\log(\hat{p_k})}{N_k^t} \tag{15}$$

On simplifying and exponentiation we get the following result:

$$\hat{p_k} \leq e^{-K(\log(K)-1)\frac{N_k^t}{\sum_k N_k^t}-1} \tag{16}$$

The penalizing factor $K(\log(K)-1)$ is increasing in terms of number of classes K in the dataset. This helps the overall objective since we need a large penalizing factor to compensate for as $N_k^t/\sum_k N_k^t$ will be smaller when number of classes is large in the dataset. Also, in case of generating a balanced distribution, $\hat{p_k} = 1/K$ which leads to the exponential average $N_k^t = 1/K$ given sufficient iterations. In this case the upper bound value will be $1/K$ which equals the value of $\hat{p_k}$, proving that the given bound is tight. We would like to highlight that the proof is valid for any $N_k^t > 0$ but in our case $\sum_k N_k^t = 1$.

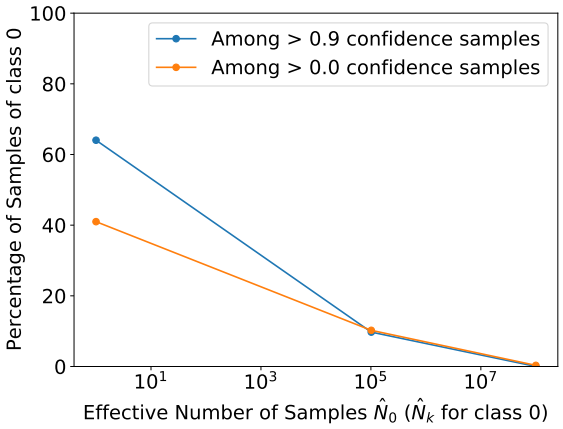


Figure 3: Shows the percentage of generated samples for class 0 by SNDCGAN on CIFAR-10 for varying values of effective class frequency $\hat{N}_0$. When $\hat{N}_0$ is large, the network tries to decrease fraction of class 0 samples whereas when $\hat{N}_0$ is small it tries to increase fraction of class 0 samples among the generated samples. The blue and orange lines respectively correspond to the percentage of class 0 samples, in > 0.9 confidence samples and all the samples.

Implications of the proposition 1: The bound on $\hat{p_k}$ is inversely proportional to the exponent of the fraction of effective class distribution N_k^t for a given class k. To demonstrate the effect of our regularizer empirically, we construct two extreme case examples:

- If $N_k^t \gg N_i^t, \forall i \neq k, N_k^t \approx 1$, then the bound on $\hat{p_k}$ would approach $e^{-K(\log(K)-1)-1}$. Hence the network is expected to decrease the proportion of class k samples.
- If $N_k^t \ll N_i^t, \forall i \neq k, N_k^t \approx 0$, then the bound on $\hat{p_k}$ will be e^{-1}. Hence the network might increase the proportion of class k samples.

We verified these two extreme cases by training a SNDCGAN [Miyato et al., 2018] (DCGAN with spectral normalization, hyperparameters defined in Appendix Table 1.4.3) on CIFAR-10 and fixing $\hat{N}_k^t$ (unnormalized version of N_k^t) across time steps and denote it as $\hat{N}_k$. We run two experiments by initialising $\hat{N}_k$ to a very large value and a very small value. Results presented in Figure 3 show that the GAN increases the proportion of samples of class k in case of low $\hat{N}_k$ and decreases the proportion of samples in case of large $\hat{N}_k$. This shows the balancing behaviour of proposed regularizer.

Proposition 2 [Guiaşu, 1971]: If each $\hat{p_k} = e^{\lambda N_k^t - 1}$ where λ is obtained from solution of $\sum_k e^{\lambda N_k^t - 1} = 1$, then the regularizer objective in Eq. 7 attains the optimal minimum value of $\lambda - \sum_k \frac{e^{\lambda N_k^t - 1}}{N_k^t}$. For proof please refer to Appendix 1.1.

Implications of Proposition 2: As we only use necessary conditions to prove the bound in proposition 1, the optimal solution can be a maxima, minima or a saddle point. Prop. 2 result shows that the optimal solution found in Prop. 1 (i.e. $\hat{p_k} = e^{\lambda N_k^t - 1}$) is inded an optimal minima.

3.3 COMBINING THE REGULARIZER AND GAN OBJECTIVE

The regularizer can be combined with the generator loss in the following way:

$$L_g = -E_{(x,z)\sim(P_r,P_z)}[\log(\sigma(D(G(z)) - D(x)))] + \lambda L_{reg} \tag{17}$$

It has been recently shown [Jolicoeur-Martineau, 2019] that the first term of the loss leads to minimization of $D_f(P_g, P_r)$ that is f-divergence between real (P_r) and generated data distribution (P_g). The regularizer term ensures that the distribution of classes across generated samples is uniform. The combined objective provides insight into the working of framework, as the first term ensures that the generated images fall in the image distribution and the second term ensures that the distribution of classes is uniform. So the first term leads to *divergence minimisation* to real data while satisfying the *constraint* of class balance (i.e. second term), hence overall it can be seen as *constained optimization*.

As P_r comprises of diverse samples from majority class the first objective term ensures that P_g is similarly diverse. The second term in the objective ensures that the discriminative properties of all classes are present uniformly in the generated distribution, which ensures that minority classes get benefit of diversity present in the majority classes. This is analogous to approaches that transfer knowledge from majority to minority classes for long-tailed classifier learning [Liu et al., 2019, Wang et al., 2017].

4 EXPERIMENTS

For evaluating the effectiveness of our balancing regularizer, we conduct image generation experiments over long-tailed distributions. In these experiments we aim to train a GAN using data from a long-tailed dataset, which is common in the real world setting. For achieving good performance on all classes in the dataset, the method requires to transfer knowledge from majority to minority classes. Several works have focused on learning classifiers on long-tailed distributions [Cao et al., 2019, Cui et al., 2019, Liu et al., 2019]. Yet, works focusing on Image Generation using long-tailed dataset are limited. Generative Minority Oversampling (GAMO) [Mullick et al., 2019] attempts to solve the problem by introducing a three player framework, which is an encoder-decoder network and not a GAN. We do not compare our results with GAMO as it is not trivial to extend GAMO to use schemes such as Spectral Normalization [Miyato et al., 2018], and ResGAN like architecture [Gulrajani et al., 2017] which impede fair comparison.

Datasets: We perform extensive experimentation on CIFAR-10 and a subset of LSUN, as these are widely used for evaluating GANs. The LSUN subset consists of 250K training images and 1.5K validation images. The LSUN subset is composed of 5 balanced classes. Santurkar et al. [2018] identified this subset to be a challenging case for GANs to generate uniform distribution of classes. The original CIFAR-10 dataset is composed of 50K training images and 10K validation images. We construct the long-tailed version of the datasets by following the same procedure as Cao et al. [2019]. Here, images are removed from training set to convert it to a long-tailed distribution while the validation set is kept unchanged. The imbalance ratio (ρ) determines the ratio of number of samples in most populated class to the least populated one: $\rho = max_k\{n_k\}/min_k\{n_k\}$.

Pre-Trained Classifier: An important component of our framework is the pre-trained classifier. All the pre-trained classifiers in our experiments use a ResNet32 [He et al., 2016] architecture. The classifier is trained using Deferred Re-Weighting (DRW) scheme [Cao et al., 2019, Cui et al., 2019] on the long-tailed data. We use the available open source code[2]. We use the following learning rate schedule: initial learning rate of 0.01 and multiplying by 0.01 at epoch 160 and 180. We train the models for 200 epochs and start reweighting (DRW) at epoch 160. We give a summary of the validation accuracy of the models for various imbalance ratios (ρ) in Table 2.

GAN Architecture: We use the SNDCGAN architecture for experiments on CIFAR-10 with images of size of 32×32 and SNResGAN (ResNet architecture with spectral normalization) structure for experiments on LSUN dataset with images of size 64×64. For the conditional GAN baselines we conditioned the generator using Conditional BatchNorm. We compare our method to two widely used conditional GANs: ACGAN and cGAN. The other baseline we use is the unconditional GAN (SNDCGAN & SNResGAN) without our regularizer. All the GANs were trained with spectral normalization in the discriminator for stabilization [Miyato et al., 2018].

Training Setup: We use the learning rate of 0.0002 for both generator and discriminator. We use Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for SNDCGAN and $\beta_1 = 0$ and $\beta_2 = 0.999$ for SNResGAN. We use a batch size of

[2]https://github.com/kaidic/LDAM-DRW

Imbalance Ratio	100			10			1
	FID (↓)	KLDiv(↓)	Acc.(↑)	FID(↓)	KLDiv(↓)	Acc.(↑)	FID (↓)
CIFAR-10							
SNDCGAN	36.97 ± 0.20	0.31 ± 0.0	68.60	32.53 ± 0.06	0.14 ± 0.0	80.60	27.03 ± 0.12
ACGAN	44.10 ± 0.02	0.33 ± 0.0	43.08	38.33 ± 0.10	0.12 ± 0.0	60.01	24.21 ± 0.08
cGAN	48.13 ± 0.01	0.02 ± 0.0	47.92	26.09 ± 0.04	0.01 ± 0.0	68.34	18.99 ± 0.03
Ours	32.93 ± 0.11	0.06 ± 0.0	72.96	30.48 ± 0.07	0.01 ± 0.0	82.21	25.68 ± 0.07
LSUN							
SNResGAN	37.70 ± 0.10	0.68 ± 0.0	75.27	33.28 ± 0.02	0.29 ± 0.0	79.20	28.99 ± 0.03
ACGAN	43.76 ± 0.06	0.39 ± 0.0	62.33	31.98 ± 0.02	0.05 ± 0.0	75.47	26.43 ± 0.04
cGAN	75.39 ± 0.12	0.01 ± 0.0	44.40	30.68 ± 0.04	0.00 ± 0.0	72.93	27.59 ± 0.03
Ours	35.04 ± 0.19	0.06 ± 0.0	77.93	28.78 ± 0.01	0.01 ± 0.0	82.13	28.15 ± 0.05

Table 1: Results on CIFAR-10 (top panel) and 5 class subset of LSUN (bottom panel) datasets with varying imbalance. In the last column FID values in balanced scenarios are present for ease of reference. FID, KL Div. and Acc. are calculated on 50K sampled images from each GAN.

Imbalance Ratio (ρ)	100	10	1
CIFAR-10	76.67	87.70	92.29
LSUN	82.40	88.07	90.53

Table 2: Validation Accuracy of the PreTrained Classifiers used with GANs. The balanced classifier also serves as an annotator.

256 and perform 1 discriminator update for every generator update. As a sanity check, we use the FID values and visual inspection of images on the balanced dataset and verify the range of values from Kurach et al. [2019]. We update the statistics N_k^t using Eq. 5 after every 2000 iterations, for all experiments in Table 1. The implementation of the GANs is done with PyTorchStudioGAN Kang and Park [2020]. Further details and ablations are present in the Appendix.

Evaluation We used the following evaluation metrics:

1. KL Divergence w.r.t. Uniform Distribution of labels: Labels for the generated samples are obtained by using the pre-trained classifier (trained on balanced data) as an annotator. The annotator is just used for evaluation on long-tailed data. Low values of this metric signify that the generated samples are uniformly distributed across classes.
2. Classification Accuracy (CA): We use the $\{(X, Y)\}$ pairs from the GAN generated samples to train a ResNet32 classifier and test it on real data. For unconditional GANs, the label Y is obtained from the classifier trained on long-tailed data. Note that this is similar to Classifier Accuracy Score of [Ravuri and Vinyals, 2019].
3. Frèchet Inception Distance (FID): It measures the 2-Wasserstein Distance on distributions obtained from Inception Network [Heusel et al., 2017]. We use 10K samples from CIFAR-10 validation set and 10K (2K from each class) fixed random images from LSUN dataset for measuring FID.

Discussion of Results: We present our results below:
1) **Stability**: In terms of stability, we find that cGAN suffers from early collapse in case of high imbalance ($\rho = 100$) and stops improving within 10K iterations. Therefore, although cGAN is stable in balanced scenario, it is unstable in case of long-tailed version of the given dataset.
2) **Biased Distribution**: Contrary to cGAN, we find that the distribution of classes generated by ACGAN, SNDCGAN and SNResGAN is imbalanced. The images obtained by sampling uniformly and labelling by annotator, suffers from a high KL divergence to the uniform distribution. Some classes are almost absent from the distribution of generated samples as shown in Figure 1. In this case, in Table 1 we observe FID score just differs by a small margin even if there is a large imbalance in class distribution. This happens for ACGAN as its loss is composed of GAN loss and classification loss terms, therefore in the long-tailed setting the ACGAN gets biased towards optimising GAN loss ignoring classification loss, hence tends to produce arbitrary distribution. Contrary to this, our GAN produces class samples uniformly as it is evident from the low KL Divergence.
3) **Comparison with State-of-the-Art Methods**: In this work we observe that classification accuracy is weakly correlated with FID score which is in agreement with Ravuri and Vinyals [2019]. We achieve better classifier accuracy when compared to cGAN in all cases, which is the current state-of-the-art for Classifier Accuracy Score (CAS). Our method shows minimal degradation in FID in comparison to the corresponding balanced case. It is also able to achieve the best FID in 3 out of 4 long-tailed cases. Hence we expect that methods such as Consistency Regularization [Zhang et al., 2019] and Latent Optimization [Wu et al., 2019] can be applied in conjunction with our method to further improve the quality of images. However in this work we specifically focused on techniques used to provide class information (Y) of the images (X) to the GAN. Several state-of-the-art GANs use the approach of cGAN [Wu et al., 2019, Brock et al., 2018] for conditioning the discriminator

	FID (↓)	KLDiv(↓)	FID (↓)	KLDiv(↓)
Imbalance Ratio	100		10	
	CIFAR-10			
SNDCGAN	36.97 ± 0.20	0.31 ± 0.0	32.53 ± 0.06	0.14 ± 0.0
Ours (Supervised)	32.93 ± 0.11	0.06 ± 0.0	30.48 ± 0.07	0.01 ± 0.0
Ours (Semi Supervised)	33.32 ± 0.03	0.14 ± 0.0	30.37 ± 0.14	0.04 ± 0.0
	LSUN			
SNResGAN	37.70 ± 0.10	0.68 ± 0.0	33.28 ± 0.02	0.29 ± 0.0
Ours (Supervised)	35.04 ± 0.19	0.06 ± 0.0	28.78 ± 0.01	0.01 ± 0.0
Ours (Semi Supervised)	35.95 ± 0.05	0.15 ± 0.0	30.96 ± 0.07	0.06 ± 0.0

Table 3: Comparison of results in Semi Supervised Setting. The pretrained classifier used in our framework is fine-tuned with 0.1% of labelled data. The same classifier trained on balanced dataset is used as annotator for calculating KL Divergence for all baselines.

and the generator.

4.1 EXPERIMENTS ON DATASETS WITH LARGE NUMBER OF CLASSES

For showing the effectiveness of our method on datasets with large number of classes, we show results on long-tailed CIFAR-100 ($\rho = 10$) and iNaturalist-2019 [iNaturalist, 2019] (1010 classes). The iNaturalist dataset consists of image of species which naturally have a long-tailed distribution. For the iNaturalist dataset we use a batch size of 256 which is significantly less than number of classes present (1010), and use $\beta = \alpha$ for updating the *effective class distribution* in Eq. 5. We use a SNResGAN based architecture for both datasets. Additional hyperparameters and training details are present in Appendix Section 1.5. We use CIFAR-100 validation set and a balanced subset of (16160) iNaturalist images for calculation of FID with 50K generated images in each case.

Table 4 summarizes the results on the above two datasets. Our method clearly outperforms all the other baselines in terms of FID. For the CIFAR-100 dataset our method achieves balanced distribution similar to the cGAN. In the case of iNaturalist our method achieves KL DIV of 0.6 which is significantly better than other baselines. Other baselines cause significant imbalance in the generated distribution and hence are unsuitable for real world long-tailed datasets such as iNaturalist (examples are shown in Figure 4. The superior results on iNaturalist demonstrate that our method is also effective in the case when batch size is less than the number of classes present in the dataset.

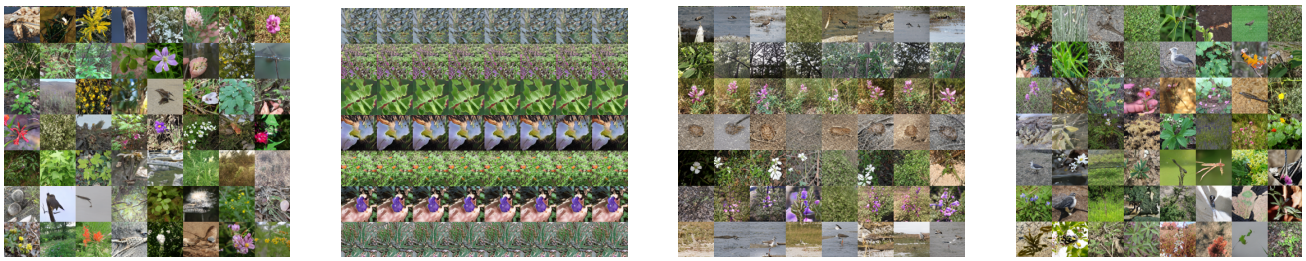
4.2 OTHER BASELINES USING PRE-TRAINED CLASSIFIER

Since the proposed generative framework utilizes a pre-trained classifier, we believe the comparison (against other generative model) should be performed via providing the same classifier. Therefore, in this subsection, we choose ACGAN framework and build a baseline by adding the pre-trained classifier. Note that in ACGAN discriminator not only performs the real-fake distinction but also serves as an auxiliary classifier and labels the sample into the underlying classes in the dataset. In this baseline, we replace the latter part with the pre-trained classifier used in the proposed framework. Hence both the frameworks are even with respect to the availability of the $P(Y/X)$ information.

The resulting generator can avail the label information of the generated samples from this classifier. In other words, if the generator intends to produce a sample of class y via conditioning, the pretrained classifier can provide the required feedback in the form of cross entropy loss. However, we find that this baseline of ACGAN that employs a pre-trained classifier suffers from mode collapse (42.28 FID) and only generates extremely limited within class diversity in images (e.g. in Fig. 4). On the contrary, our method (9.01 FID) using the same pre-trained classifier doesn't suffer from mode collapse and also works in case of iNaturalist dataset. This shows that the proposed framework and regularizer are non-trivial and prevent the GAN from mode collapse. We believe there is scope for understanding the nature of the involved optimization in this future. For the exact details of implementation please refer to the Appendix Section 1.6. An overview of the baseline is depicted in Figure 3.

4.3 SEMI-SUPERVISED CLASS-BALANCING GAN

In case of conditional GANs, class labels are required for GAN training. However, in our case the stage of classifier learning which requires labels is decoupled from GAN learning. In this section we show how this can be advantageous in practice. Since our framework only requires knowledge of $P(Y/X)$, we find that a classifier trained through any of a variety of sources could be used for providing feedback to the Generator. This feedback allows the GAN to generate class balanced distributions even in cases when the labels for underlying long-tailed distributions are not known. This reduces the need for labelled data in our framework and shows the effectiveness over conditional GAN. Note that the performance of conditional GANs deteriorates [Lucic et al., 2019] when used with limited labelled data. We use a ResNet-50 pretrained model on ImageNet from BiT (Big Image Transfer) [Kolesnikov et al., 2019] and finetune it using 0.1 % of labelled data of balanced training set (i.e. 5 images per class for CIFAR-10 and 50 images per class for LSUN dataset).



(a) SNResGAN (13.03 FID) (b) ACGAN (47.15 FID) (c) cGAN (21.53 FID) (d) Ours (9.01 FID)

Figure 4: Shows the 64×64 generated images for the iNaturalist-2019 dataset for different baselines.

	iNatuarlist 2019		CIFAR-100	
	FID (↓)	KL Div(↓)	FID (↓)	KL Div(↓)
SNResGAN	13.03 ± 0.07	1.33 ± 0.0	30.05 ± 0.05	0.18 ± 0.0
ACGAN	47.15 ± 0.11	1.80 ± 0.0	69.90 ± 0.13	0.40 ± 0.0
cGAN	21.53 ± 0.14	1.47 ± 0.0	30.87 ± 0.06	0.09 ± 0.0
Ours	9.01 ± 0.08	0.60 ± 0.0	28.17 ± 0.06	0.11 ± 0.0

Table 4: Results on iNaturalist (2019) and CIFAR-100 ($\rho = 10$) dataset. Significant performance increase is achieved by our method in comparison to baselines.

In Table 3 we use the classifier trained with 0.1% of labelled data to train GAN on long-tailed version of CIFAR-10 and LSUN datasets ($\rho = 10, 100$) We observe that even with semi-supervised classifier, our method is able to produce an avg reduction of **0.26** in KL Divergence when compared to unsupervised GAN (SNResGAN) and also achieves better FID score in all cases. Note that this application is unique to our framework as conditional GANs explicitly require labels for whole dataset for training.

4.4 ANALYSIS ON THE EFFECT OF CLASSIFIER PERFORMANCE

For analyzing the effect of classifier performance on the GAN training in our method, we train the GAN on long-tailed CIFAR-10 ($\rho = 10$) with different classifiers. We learn multiple classifiers with different imbalance ratios (ρ) of $1, 10, 100, 500$ and 5000. As the imbalance ratio increases, the accuracy of the resulting classifier decreases, hence, we can have classifiers of varied accuracy for deploying in our framework. Note that in case of high imbalance ratio, it becomes harder for the classifier to learn the tail part of the distribution. Figure 5 shows the performance of the resulting GAN (in terms of the FID and KLDIV measures) with respect to the classifier performance on the tail class (i.e. least populated class).

From Figure 5 it can be observed that for a large range of classifier accuracies on the tail class, GANs learned in our framework are able to achieve similar FID and KL Divergence performance. Our framework only requires reasonable classifier performance which can be easily achieved via normal cross entropy training on the long-tailed dataset. For further enforcing this claim, we only use a classifier that is trained using cross entropy loss for iNaturalist-2019 experiments, in which our method achieves state-of-the-art FID score. Hence one does not need to explicitly resort to any complex training regimes for the classifier. In the extreme case of $\rho = 5000$, the classifier performance is 0 on the tail class. In this case, our algorithm diverges and is not able to produce meaningful results. However, note that with a classifier accuracy of as small as 20% our framework achieves decent FID and KL Divergence.

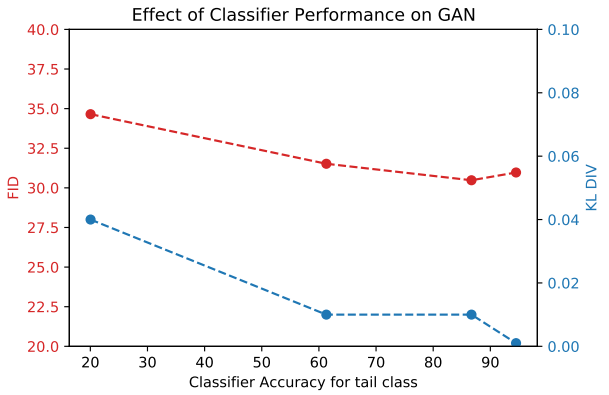


Figure 5: Analysis of Classifier Performance on long-tailed CIFAR-10 ($\rho = 10$). The data points from left to right correspond to imbalance ratio's (ρ) 500, 100, 10 and 1 respectively. The performance is significantly robust after tail class accuracy reaches 60%.

5 CONCLUSION

Long-tailed distributions are a common occurrence in real-world datasets in the context of learning problems such as object recognition. However, a vast majority of the existing contributions consider simplified laboratory scenarios in which the data distributions are assumed to be uniform over classes. In this paper, we consider learning a genera-

tive model (GAN) on long-tailed data distributions with an objective to faithfully represent the tail classes. We propose a class-balancing regularizer to balance class distribution of generated samples while training the GAN. We justify our claims on the proposed regularizer by presenting a theoretical bound and comprehensive experimental analysis. The key idea of our framework is having a classifier in the loop for keeping an uninterrupted check on the GAN's learning which enables it to retain the minority nodes of the underlying data distribution. We demonstrated that the dependency on such a classifier is not arduous. Our experimental analysis clearly brings out the effectiveness of our regularizer in the GAN framework for generating the images from complex long-tailed datasets such as iNaturalist, on which it achieves the state-of-the-art performance.

Acknowledgements

Harsh Rangwani acknowledges the support from Prime Minister's Research Fellowship (PMRF). This work was supported by SERB, DST, Govt. of India (Project: STR/2020/000128). We thank Sravanti Addepalli, Gaurang Sriramanan and other Video Analytics Lab members for their valuable feedback.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pages 6594–6604, 2017.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

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.

Silviu Guiaşu. Weighted entropy. *Reports on Mathematical Physics*, 2(3):165–179, 1971.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

iNaturalist. The inaturalist 2019 competition dataset. `https://github.com/visipedia/inat_comp/tree/2019`, 2019.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018.

Alexia Jolicoeur-Martineau. On relativistic f-divergences. *arXiv preprint arXiv:1901.02474*, 2019.

Minguk Kang and Jaesik Park. Contrastive generative adversarial networks. *arXiv preprint arXiv:2006.12681*, 2020.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *arXiv preprint arXiv:1912.11370*, 2019.

Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in gans. In *International Conference on Machine Learning*, pages 3581–3590. PMLR, 2019.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Mario Lucic, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. *arXiv preprint arXiv:1903.02271*, 2019.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=ByS1VpgRZ`.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative adversarial minority oversampling. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

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.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems*, pages 12268–12279, 2019.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

Shibani Santurkar, Ludwig Schmidt, and Aleksander Madry. A classification-based study of covariate shift in gan distributions. In *International Conference on Machine Learning*, pages 4480–4489. PMLR, 2018.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *Advances in Neural Information Processing Systems*, pages 7029–7039, 2017.

Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy Lillicrap. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. *arXiv preprint arXiv:1910.12027*, 2019.