Supplementary Material
Improving GANs for Long-Tailed Data through Group Spectral Regularization

Harsh Rangwani[1], Naman Jaswani[1], Tejan Karmali[1,2],
Varun Jampani[2], and R. Venkatesh Babu[1]

[1] Indian Institute of Science, Bengaluru, India
[2] Google Research

Table of Contents

1 Notations

We summarize the notations used in the paper in Table 2.

2 Addtional Metrics

In addition to FID and IS reported for experiments in main paper, we also evaluate additional metrics of Precision [7], Recall [7], Density [16] and Cover-

Table 1: Additional metrics on CIFAR-10 dataset.

Imb. Ratio (ρ)	100					1000				
	Intra-class FID	Precision	Recall	Density	Coverage	Intra-class FID	Precision	Recall	Density	Coverage
SNGAN	78.36	0.69	0.53	0.67	0.51	121.57	0.60	**0.40**	0.43	0.32
+ gSR (Ours)	**55.71**	**0.71**	**0.56**	**0.76**	**0.67**	**108.12**	**0.63**	0.39	**0.53**	**0.34**
BigGAN	57.82	0.65	**0.58**	0.63	0.67	109.29	0.56	0.50	0.40	0.40
+ gSR (Ours)	**43.41**	**0.74**	0.56	**0.93**	**0.80**	**98.59**	**0.59**	**0.51**	**0.49**	**0.51**

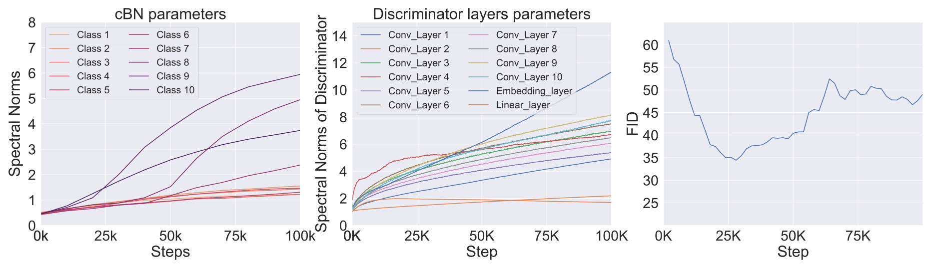
Table 2: Notation Table

Symbol	Space	Meaning
K	$\mathbb{N}$	Number of Classes
y	$\{1, 2, ..., K\}$	Class label
$\mathbf{z}$	$\mathbb{R}^{256}$	Noise vector
D		Discriminator
G		Generator
$\mathbf{x}$	$\mathbb{R}^{3\times H\times W}$	Image
$\mathbf{x_y^l}$	$\mathbb{R}^d$	Feature vector from the Generator's l^{th} cBN's input feature map
$\mu_\mathbf{B}^\mathbf{l}$	$\mathbb{R}^d$	Mean of incoming features to Generator's l^{th} cBN from minibatch B
$\sigma_\mathbf{B}^\mathbf{l}$	$\mathbb{R}^d$	Std. dev. of incoming features to Generator's l^{th} cBN from minibatch B
$\gamma_\mathbf{y}^\mathbf{l}$	$\mathbb{R}^d$	Gain parameter for y^{th} class of l^{th} cBN layer of Generator
$\beta_\mathbf{y}^\mathbf{l}$	$\mathbb{R}^d$	Bias parameter for y^{th} class of l^{th} cBN layer of Generator
n_g	$\mathbb{R}$	Number of groups
n_c	$\mathbb{R}$	Number of columns
$\mathbf{\Gamma_y^l}$	$\mathbb{R}^{n_g\times n_c}$	$\gamma_\mathbf{y}^\mathbf{l}$ after grouping
$\mathbf{B_y^l}$	$\mathbb{R}^{n_g\times n_c}$	$\beta_\mathbf{y}^\mathbf{l}$ after grouping
σ_{max}	$\mathbb{R}^+$	Spectral norm
n_y	$\mathbb{N}$	Number of samples in class y
ρ	$\mathbb{R}$	Imbalance ratio: Ratio between the most and the least frequent classes of the dataset

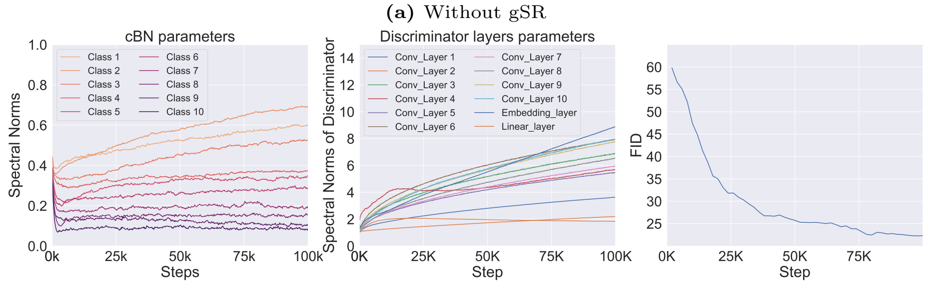
age [16] and Intra-FID for CIFAR-10 dataset. We observe that across all the 4 different imbalance configurations (as in main paper Table 2) there is significant improvement in all metrics but Recall (which is comparable to baseline in all cases.

3 Correlations between Spectral Norms and Class-Specific Mode Collapse

In this section, we provide additional details and comparisons to emphasize the differences between class-specific mode collapse and the usual mode collapse (as described in main paper Sec. 3.2). In SNGAN [9] and BigGAN [1], the discriminator's (D) weights' spectral norms tend to explode as the mode collapse occurs for balanced data. To determine if this also occurs in long-tailed case we train a SNGAN on CIFAR-10 ($\rho = 100$) (with and without gSR) and plot the spectral norm of weights of discriminator layers. We find that spectral explosion for discriminator weights is not observed in the class-specific mode collapse (without gSR case), as we report in Fig. 1. Discriminator's layers' spectral norms do not show significant change before and after applying gSR . On the other hand,



(a) Without gSR



(b) With gSR

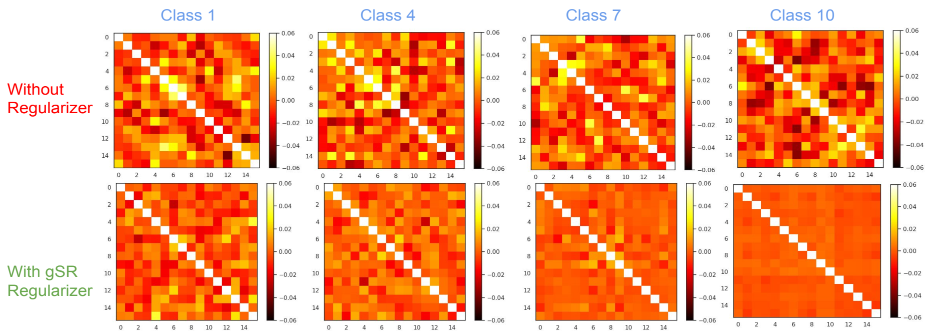
Fig. 1: Class-specific mode collapse exhibits unique behaviour with respect to cBN parameters. Class-specific mode collapse leads to spectral explosion in Generator's cBN parameters' spectral norms (left), which correlates with explosion of FID (right), while having little effect on discriminator's parameters' spectral norms (middle). Class-specific mode collapse is remedied by gSR which keeps the cBN parameters' spectral norms under control.

before applying gSR the spectral norms of class-specific parameters of cBN explode (at step 25k and 50k). At the same stage FID suddenly increases, whereas there is no anomaly in Discriminator's spectral norms'. Thus, the class-specific mode collapse behaviour is different as compared to that of the mode collapse previously reported in the literature [1,10], and cannot be detected through discriminator spectral norms. Hence, it's detection requires the analysis of spectral norms of grouped parameters in cBN which we propose in this paper.

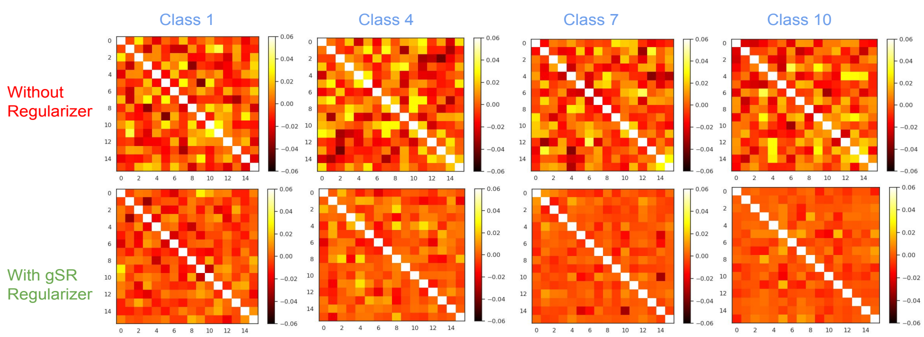
The above spectral explosion of the generator's cBN motivates us to formulate gSR (Sec.3.3). We find (Fig. 1) that after applying gSR there is no spectral collapse and training is stabilized (decreasing FID).

4 Analysis of Covariance of grouped cBN Parameters

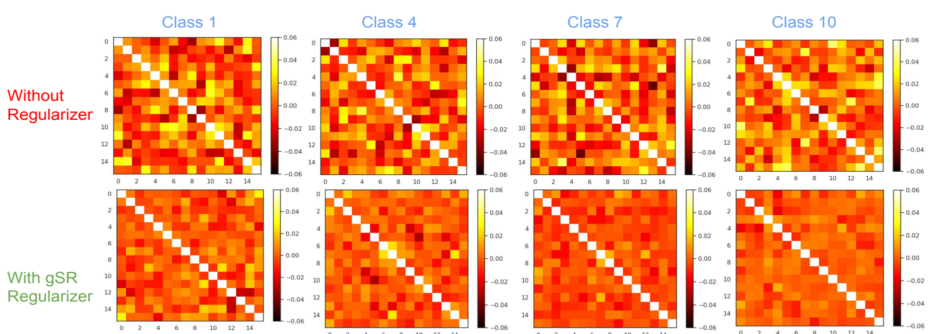
For analyzing the decorrelation effect of gSR (explained in Sec. 3.3), we train a SNGAN on CIFAR-10 (ρ=100) with gSR. We then visualize the covariance matrices of $\mathbf{\Gamma^l_y}$ (grouped $\gamma^{\mathbf{l}}_{\mathbf{y}}$) across cBN at different layers $\mathbf{l}$ in the generator. gSR leads to suppression of covariance between off-diagonal features of $\mathbf{\Gamma^l_y}$ belonging to the tail classes, implying decorrelation of parameters (Sec. 3.3). As we go from initial to final cBN layers of the Generator, we see that this suppression is reduced in the case when gSR is applied. This leads to increased similarity



(a) Covariance matrices of $\mathbf{\Gamma_y^l}$ for $(\mathbf{l = 1})$ for SNGAN baseline.



(b) Covariance matrices of $\mathbf{\Gamma_y^l}$ for $(\mathbf{l = 3})$ for SNGAN baseline.



(c) Covariance matrices of $\mathbf{\Gamma_y^l}$ for $(\mathbf{l = 5})$ for SNGAN baseline.

Fig. 2: Covariance matrices of $\mathbf{\Gamma_y^l}$ for SNGAN baseline on CIFAR-10 ($\rho = 100$).

between the covariance matrices of the head class and tail class. This effect can be attributed to the features learnt at the respective layers. The initial layers (in G) are responsible for more abstract and class-specific features, whereas the final layers produce features while are more fine-grained and generic across different

Fig. 3: Qualitative comparison of BigGAN variants on Tail classes from iNaturalist 2019 dataset (ρ=100) (64 $\times$ 64). Each row represents images from a distinct class.

classes. This is in contrast to what is observed for a classifier, as the generator is an inverted architecture in comparison to a classifier.

5 Qualitative Results

We show generated images on iNaturalist-2019 and AnimalFace in Fig. 4 and Fig. 3. These are naturally occurring challenging data distributions for training a GAN. Sample diversity as well as quality is improved after applying our gSR regularizer. We also provide a video showing class specific collapse for BigGAN for CIFAR-10 in `gSR.mp4`.

6 Experimental Details

In this section, we elaborate on the technical and implementation details provided in Sec. 4 of the main paper.

6.1 Datasets

We describe the datasets used in our work below:

CIFAR-10: We use CIFAR-10 [6] dataset which comprises of 32 $\times$ 32 images. The dataset is split into 50k training images and 10k test images. We use the training images for GAN training and the 10k test set for calculation of FID.

LSUN: We use a 250k subset of LSUN [15] dataset as followed by [11,12], which is split across the classes of bedroom, conference room, dining room, kitchen

Fig. 4: Qualitative Results. The baseline is composed of BigGAN [1]+LeCam [14]+DiffAug [17]. gSR improves the quality and diversity of the images generated by baseline over challenging iNaturalist-19 and AnimalFace datasets.

and living room classes. We use a balanced subset of 10k images balanced across classes for FID calculation.

iNaturalist-2019: The iNaturalist-2019 [4] is a long-tailed dataset composed of 268,243 images present across 1010 classes in the training set. The validation set contains 3030 images balanced across classes, used for FID calculation.

AnimalFace [13]: The AnimalFace dataset contains 2,200 RGB images across 20 different categories with images containing animal faces. We use the training set for calculation of FID as there is no seperate validation set provided for baselines. Our results on this dataset show that our regularizer can also help in preventing collapse in extremely low data (i.e. few shot) scenario's as well.

6.2 LeCam Regularizer

We use LeCam regularizer [14] for all our experiments.

$$R_{LC} = \mathop{\mathbb{E}}_{\mathbf{x}\sim\mathcal{T}}[\|D(\mathbf{x}) - \alpha_F\|^2] + \mathop{\mathbb{E}}_{\mathbf{z}\sim p_{\mathbf{z}}}[\|D(G(\mathbf{x})) - \alpha_R\|^2] \tag{1}$$

LeCam regularizer computes exponential moving average of discriminator outputs for real and generated images. The difference between discriminator outputs for real and generated images is taken against the moving averages of discriminator outputs of generated images (α_F) and real images (α_R) respectively. This does not allow the discriminator to output predictions with very high confidence, thereby preventing overfitting by keeping the predictions in a particular range. We use the λ_{LC} value of 0.1, 0.3 and 0.01 as suggested by the authors [14] ,which is specified in Table 4. The term $\lambda_{LC}R_{LC}$ is then added to discriminator loss for regularization.

6.3 Spectral Norm Computation Time

Since our regularizer involves estimating largest singular value for $\mathbf{\Gamma_y^l}$, this can be done through either power iteration or SVD. We use power iterations method to calculate the singular values of $\mathbf{\Gamma_y^l}$ and $\mathbf{B_y^l}$. We use 4 power iterations for estimating the largest singular value. For perfect decorrelation, other techniques like Group Whitening [3] can also be used, but they involve full SVD computation. We provide a comparison of time for 100 generator steps of training for baseline, baseline (w/ power iteration (piter)) and baseline (with full SVD) computation for iNaturalist 2019 dataset in table below. All the runs were done on NVIDIA RTX 3090 GPU on the same machine.

	Time (in secs)
BigGAN	68
BigGAN (w/ piter)	77
BigGAN (w/ SVD)	1126

Table 3: Comparison of time taken for 100 updates of generator(G) on iNaturalist-2019 dataset.

As for each class seperate SVD computation is performed we find that the SVD computation becomes very expensive (Table 3) for large datasets like iNaturalist-2019. Whereas as the power iteration can be done in parallel there is not much computation overhead with addition of each class. Hence, techniques

Table 4: Hyperparameter setups for all the reported experiments. α_D, and α_G denote the learning rates for Discriminator and Generator respectively.

Setting	Adam (α_D, α_G, β_1, β_2)	n_{dis}	λ_{LC}	G_{EMA}	EMA Start	Total Iterations
A	2e-4, 2e-4, 0.5, 0.9	5	0.3	False		120k
B	2e-4, 2e-4, 0.5, 0.999	5	0.1	True	1k	120k
C	2e-4, 2e-4, 0.5, 0.9	5	0.3	True	1k	200k
D	2e-4, 2e-4, 0.0, 0.999	2	0.01	True	20k	120k
E	2e-4, 2e-4, 0.5, 0.999	5	0.01	True	1k	120k
F	4e-4, 1e-4, 0.5, 0.9	5	0.5	True	1k	120k

	CIFAR-10	LSUN	iNaturalist-19	AnimalFace
LSGAN [8] SNGAN [9] + gSR (Ours)	A		—	
BigGAN [1] + gSR (Ours)	B	C—F	D	E

like Group Whitening [3] which use SVD are not a viable baseline for our case. It can be observed that despite having large number of classes in iNaturalist there is only addition of 9 sec, which shows the scalability and viability of proposed gSR. We provide a PyTorch implementation of cBN, detailing the process of spectral norm calculation as part of the supplemental material.

6.4 Sanity Checks

We build our experiments over the PyTorch-StudioGAN framework, which provides a simple framework over standard GAN architectures and setups. Since we are not using the official code for the LeCam Regularizer baseline [14], we first reproduce the BigGAN (+ LeCam + DiffAug) results on CIFAR-10 to ensure that our codebase is on par with the official codebase of the LeCam GAN. Our code obtains an FID of $7.59_{\pm 0.04}$ vs. $8.31_{\pm 0.03}$ reported in same setting by *Tseng et al.* [14], which verifies the authenticity of our experiments. Hence, we compare our results to a stronger baseline which is due to improved implementation of BigGAN in the framework.

6.5 Hyperparameters

We provide the details of the hyperparameters used in the experiments in Table 1 and 2 of the main paper in Table 4. For CBGAN [11] based experiments we follow the same setup as reported in the paper (except using a ResNet [2] architecture for fairness in experiments). For BigGAN on LSUN dataset we use configuration C for the imbalance factor ($\rho = 100$) and F for imbalance factor ($\rho = 1000$). In our tuning experiments we explored the configurations in Table 4 and use the configuration which produces best FID for baseline. Then we add gSR regularizer to obtain our results.

High-Resolution Experiments: For the high resolution (128×128) image synthesis on LSUN we find that we only require very small change in hyperparameters for obtaining results. For SNGAN, we use configuration A in Table 4 with EMA starting at 1k along with $\lambda_{LC} = 0.5$. For the BigGAN we use the same configuration as in the Table 4. We find that for higher resolutions a larger λ_{LC} helps the purpose.

6.6 Intuition about n_c and n_g

As we group the parameters $\gamma_{\mathbf{y}}^{\mathbf{l}}$ (Eq. 3 in main paper) to a matrix $\mathbf{\Gamma}_{\mathbf{y}}^{\mathbf{l}}$ of $n_c \times n_g$. The matrix can be decomposed into $\min(n_c, n_g)$ (matrix rank) number of independent and diverse components through SVD. As the scope of attaining maximal orthogonal and diverse components (matrix rank) is when $n_c \approx n_g$, it helps gSR to ensure maximal diversity and performance (as seen in main paper Table 5). In case of gSR we find that almost all eigen values of $\mathbf{\Gamma}_{\mathbf{y}}^{\mathbf{l}}$ have a similar value, which demonstrates orthogonality and diversity.

7 Analysis of gSR

How much should be gSR's strength (λ_{gSR})? We experiment with different values of λ_{gSR} for gSR in SNGAN as shown in Fig. 5. λ_{gSR} value of 0.5 attains best FID scores, hence we use it for all our experiments. The value of FID changes marginally when λ_{gSR} goes from 0.25 to 1 which highlights its robustness (*i.e.* less sensitivity).

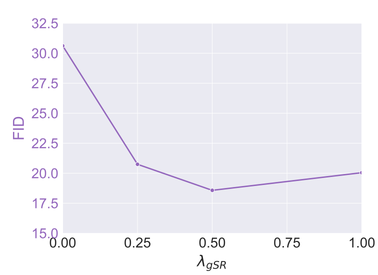


Fig. 5: Sensitivity to λ_{gSR}. On CIFAR-10, the FID marginally changes with λ_{gSR} (0.25 to 1).

How does performance of gSR change with degree of imbalance? Fig. 6.B shows the comparison of mean FID of SOTA BigGAN (with DiffAug+LeCAM) and BigGAN (with gSR) in which we find that addition of *gSR significantly improves performance across degrees of imbalance ratio*. Also, in the balanced case the performance with gSR is only slightly worse (by 0.95 FID) to the baseline.

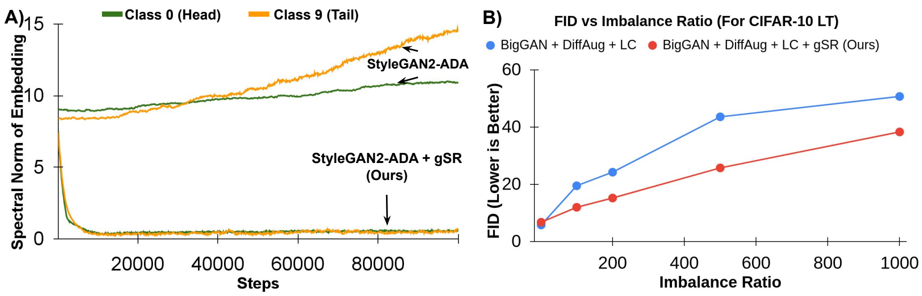


Fig. 6: A) Spectral Norm comparison of StyleGAN2-ADA with and without gSR B) mean FID comparison of baseline with gSR across degrees of imbalance

Table 5: Quantitative comparison of gSR over StyleGAN2-ADA baseline.

	CIFAR10-LT ($\rho = 100$)		LSUN-LT ($\rho = 100$)	
	FID-10k [17] (↓)	IS(↑)	FID-10k [17] (↓)	IS(↑)
StyleGAN2-ADA	$71.09_{\pm 0.12}$	$5.66_{\pm 0.03}$	$55.04_{\pm 0.07}$	$3.92_{\pm 0.02}$
+gSR(Ours)	$\mathbf{22.76}_{\pm 0.17}$	$\mathbf{7.55}_{\pm 0.01}$	$\mathbf{27.85}_{\pm 0.06}$	$\mathbf{4.32}_{\pm 0.01}$

8 gSR for StyleGAN2

We train and analyze the spectral norm of class-conditional embeddings in StyleGAN2-ADA implementation available [5] on long-tailed datasets (CIFAR10 and LSUN), to find that it also suffers from spectral collapse of tail class embedding parameters (Fig. 6.A) as BigGAN and SNGAN. We then implement gSR for StyleGAN2 generator by grouping 512 dimensional class conditional embeddings to 16x32 and calculating their spectral norm which is added to loss (Eq. 5) as R_{gSR}. We find that gSR is able to effectively prevent the mode collapse (Fig. 7) and also results in significant improvement in FID (Tab. 5) in comparison to StyleGAN2-ADA baseline.

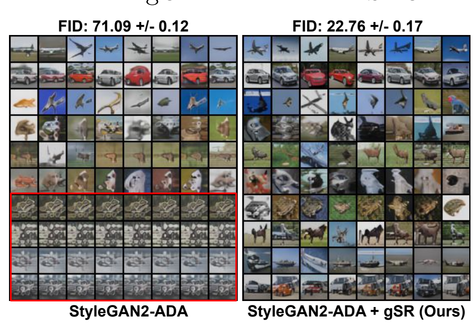


Fig. 7: StyleGAN2-ADA On CIFAR-10 ($\rho = 100$), comparison of gSR with the baseline.

References

1. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018) 2, 3, 6, 8
2. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. In: Advances in neural information processing systems (2017) 9
3. Huang, L., Zhou, Y., Liu, L., Zhu, F., Shao, L.: Group whitening: Balancing learning efficiency and representational capacity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021) 7, 8
4. iNaturalist: The inaturalist 2019 competition dataset. https://github.com/visipedia/inat_comp/tree/2019 (2019) 6
5. Kang, M., Park, J.: Contrastive generative adversarial networks. arXiv preprint arXiv:2006.12681 (2020) 10
6. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009) 5
7. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in Neural Information Processing Systems **32** (2019) 1
8. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision (2017) 8
9. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018) 2, 8
10. Miyato, T., Koyama, M.: cGANs with projection discriminator. In: International Conference on Learning Representations (2018) 3
11. Rangwani, H., Mopuri, K.R., Babu, R.V.: Class balancing gan with a classifier in the loop. arXiv preprint arXiv:2106.09402 (2021) 5, 9
12. Santurkar, S., Schmidt, L., Madry, A.: A classification-based study of covariate shift in gan distributions. In: International Conference on Machine Learning. pp. 4480–4489 (2018) 5
13. Si, Z., Zhu, S.C.: Learning hybrid image templates (hit) by information projection. IEEE Transactions on pattern analysis and machine intelligence **34**(7) (2011) 6
14. Tseng, H.Y., Jiang, L., Liu, C., Yang, M.H., Yang, W.: Regularizing generative adversarial networks under limited data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021) 6, 7, 8
15. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. CoRR **abs/1506.03365** (2015) 5
16. Yu, N., Li, K., Zhou, P., Malik, J., Davis, L., Fritz, M.: Inclusive gan: Improving data and minority coverage in generative models. arXiv preprint arXiv:2004.03355 (2020) 1, 2
17. Zhao, S., Liu, Z., Lin, J., Zhu, J.Y., Han, S.: Differentiable augmentation for data-efficient gan training. Advances in Neural Information Processing Systems **33** (2020) 6, 10