Supplementary

1. Few-Shot Image Generation

Few-shot image generation (StyleGAN2) We follow the experimental setting of [15] and show performance on 100-shot Obama, Panda and Grumpy Cat datasets (having 256 $\times$ 256 resolution) using FFHQ [6] pre-trained StyleGAN2 model. Table 1 shows DISP training leads to consistent improvement in FID scores over several baseline techniques except on Grumpy Cat dataset. We hypothesize that this is because the prior features of this dataset has low diversity and hence the priors used are not informative enough to lead to improved performance with DISP.

Method	Style-GAN 2 (256 x 256) Panda FID ↓	Grumpy Cat FID ↓	Obama FID ↓
FreezeD	16.69	**29.67**	62.26
+ DISP-Vgg16	**14.66**	29.93	**54.87**
DiffAugment	12.06	**27.08**	46.87
+ DISP-Vgg16	**11.14**	28.45	**43.79**
BSA*	21.38	34.20	50.72
GLANN + DISP-Vgg16	**11.51**	**29.85**	**38.57**

Table 1: 100-shot image generation results using StyleGAN2 [7] model pre-trained on FFHQ dataset for Panda, Grumpy-cat and Obama datasets. FID is computed between 5k generated and the complete training dataset. * denotes directly reported from the paper [15].

Impact of loss function To analyze the role of GAN loss function, we show the performance of DISP with different variants. Specifically, we choose these three loss functions: hinge loss (originally in our experiments), non-saturating loss [5] and the wasserstein loss [1]. Table 2 shows the corresponding results when DISP is used with FreezeD and DiffAugment. We observe that in case of FreezeD+DISP wasserstein loss significantly outperforms non-saturating loss and hinge loss. In case of DiffAugment hinge loss performs best followed by non-saturating loss and wasserstein loss.

Samples by varying number of training images Figure 1 shows samples generated by our approach when we vary the number of training examples in Anime dataset from 25-500. For quantitative results please refer Figure 3c in main submission.

Implementation Details We summarize the training procedure of DISP in Algorithm 1.

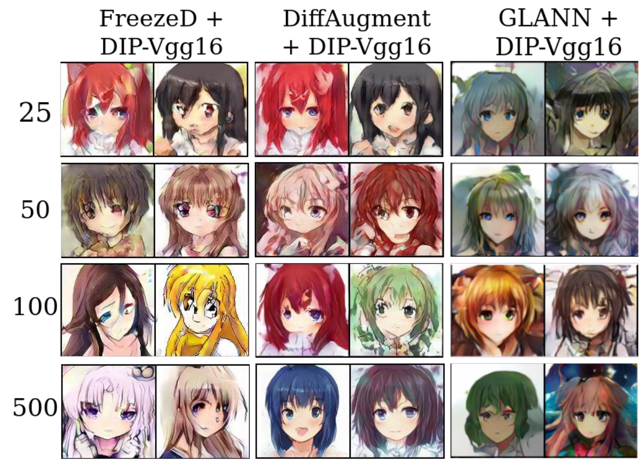


Figure 1: Samples of few-shot image generation on varying the number of training data from 25 to 500 images of Anime dataset using DISP along with different approaches with SNGAN backbone.

Method	Pre-training	SNGAN (128 x 128) Anime H	Anime NS	Anime W	Faces H	Faces NS	Faces W
FreezeD	✓	109.40	102.43	148.99	107.83	105.34	209.23
+ DISP-Vgg16		93.36	82.49	74.91	77.09	77.38	71.05
DiffAugment	$\times$	85.16	106.96	252.11	109.25	107.18	325.85
+ DISP-Vgg16		48.67	48.61	56.43	62.44	68.66	81.03

Table 2: Comparison between different loss functions in few-shot image generation using 100 training images (FID: lower is better). H is hinge loss, NS is non saturating loss and W is wasserstein loss.

In SNGAN architecture, while training with DISP, G_{emb} and D_{emb} are matrices which linearly transform the pre-trained features into generator conditional space of dimension 128 and discriminator feature space of dimension 1024. For baseline training, we use an embedding for each of the 100 training images to ensure minimal difference between baseline and our approach without increasing number of parameters. We also experimented with self-modulated [3] and unconditional training which resulted in either training collapse or worse results in all approaches. In DiffAugment, we use three augmentations: translation, cutout, and color with consistency regularization hyperparameter as 10 and training is done from scratch following the implementation in their paper [15]. In FreezeD, we freeze the first five blocks of the discriminator and finetune the rest. We use spectral normalization for both generator and discriminator during training with batch size of 25, number of discriminator steps as 4, G and D learning rate as $2e-4$, $\mathbf{z}$ dimension as 120 and maximum number of training steps as

Algorithm 1: Data InStance Prior (DISP) training algorithm

Input:G, D network with parameters θ_G and θ_D, pre-trained model C for extracting prior condition, samples from real data distribution $q(x)$ and latent distribution $p(z)$, batch size b, number of training iterations, discriminator update steps d_{step} for each generator update, Adam optimizer hyperparameters α, β_1, β_2.

for *number of training iterations* **do**

  for $t : 1...d_{step}$ **do**

    Sample batch $x \sim q(x)$, $z \sim p(z)$

    $x_{fake} = G(z|C(x))$

    $D(x, C(x)) = D_f(x) \cdot D_{emb}(C(x)) + D_l \circ D_f(x)$

    $D(x_{fake}, C(x)) = D_f(x_{fake}) \cdot D_{emb}(C(x)) + D_l \circ D_f(x_{fake})$

    $L_D = \max(0, 1 - D(x, C(x))) + \max(0, 1 + D(x_{fake}, C(x)))$

    Update $\theta_D \leftarrow Adam(L_D, \alpha, \beta_1, \beta_2)$

  end

  Sample $z \sim p(z)$

  Generate images $x_{fake} = G(z|C(x)$

  $D(x_{fake}, C(x)) = D_f(x_{fake}) \cdot D_{emb}(C(x)) + D_l \circ D_f(x_{fake})$

  $L_G = -D(x_{fake}, C(x))$

  Update $\theta_G \leftarrow Adam(L_G, \alpha, \beta_1, \beta_2)$

end

return θ_G, θ_D.

$30K$. During evaluation, moving average weights [12] of the generator is used in all experiments unless stated otherwise. For FID calculation, we select the snapshot with best FID similar to [4, 15]. For calculating precision and recall based on the k-nearest neighbor graph of inception features, as in [8], we use k as 10 for Precision and 40 for Recall.

For StyleGAN2, G_{emb} is a 2-layer MLP with ReLU nonlinearity which maps $C(\mathbf{x})$ to a 512-dimensional generator conditional space. It is then concatenated with random noise $\mathbf{z}$ of dimension 512 which is used as input in the mapping network. D_{emb} is a linear transformation matrix and discriminator loss is projection loss combined with real/fake loss. Training is done with batch-size of 16 for DiffAugment[1] and 8 for FreezeD[2] till $20k$ steps.

In case of BSA, we show that DISP can be used to improve the results on similar non-adversarial generative models. Specifically, we perform experiments with GLANN [3] which is a two step training procedure, as follows: (1) Optimize for image embeddings $\{\mathbf{e}_i\}$ of all training images $\{\mathbf{x}_i\}$ jointly with a generator network G using perceptual loss; and (2) Learn a sampling function $T : \mathbf{z} \rightarrow \mathbf{e}$ through IMLE for generating random images during inference. For using data instance prior in the training procedure of GLANN, instead of directly optimizing for $\{\mathbf{e}_i\}$, we optimize for the following modified objective:

$$\begin{aligned} &\arg\min_{G, G_{emb}} \sum_i L_{perceptual}(G \circ G_{emb} \circ C(\mathbf{x}_i), \mathbf{x}_i) \\ &\text{where } \{e_i\} = \{G_{emb} \circ C(\mathbf{x}_i)\} \end{aligned} \tag{1}$$

We finetune the pre-trained generator on batch-size of 50 with a learning rate of 0.01 for 4000 epochs. During second step of IMLE optimization, we use a 3-layer MLP with $\mathbf{z}$ dimension as 64 and train for 500 epochs with a learning rate of 0.05.

Comparison with Logo-GAN Logo-GAN [11] has shown advantage of using features from pre-trained ImageNet network in unconditional training by assigning class label to each instance based on clustering in the feature space. We compare our approach with this method in the few-shot data setting. For implementing logo-GAN, we perform class-conditional training [10] using labels obtained by K-means clustering on Vgg16 features of 100-shot Anime dataset. The results reported in Table 3 show the benefit of directly using features as data instance prior instead of only assigning labels based on feature clustering.

Method	Anime (SNGAN) FID ↓
FreezeD + DISP	**93.36**
FreezeD + Logo-GAN (K=5)	226.60
FreezeD + Logo-GAN (K=10)	183.38
DiffAugment + DISP	**48.67**
DiffAugment + Logo-GAN (K=5)	130.54
DiffAugment + Logo-GAN (K=10)	190.59

Table 3: 100-shot image generation comparison of DISP with Logo-GAN [11] on Anime dataset where priors are derived from Vgg16 network trained on ImageNet. FID is computed between 10k generated and real samples (disjoint from training set).

2. Limited data Image Generation

Experiments on CIFAR-10 and CIFAR-100 For results shown in Table 3 of main submission, BigGAN model used for training CIFAR-10 and CIFAR-100 is same as the one used for large scale experiments in Section 5.3 of main submission. In DiffAugment with BigGAN architecture, we use all three augmentations: translation, cutout, and color along with consistency regularization hyperparameter as 10. In DiffAugment + DISP consistency regularization hyperparameter is changed to 1. For experiments on StyleGAN2 architecture we use the code-base of DiffAugment [4].

[1] https://github.com/mit-han-lab/data-efficient-gans

[2] https://github.com/sangwoomo/FreezeD

[3] https://github.com/yedidh/glann

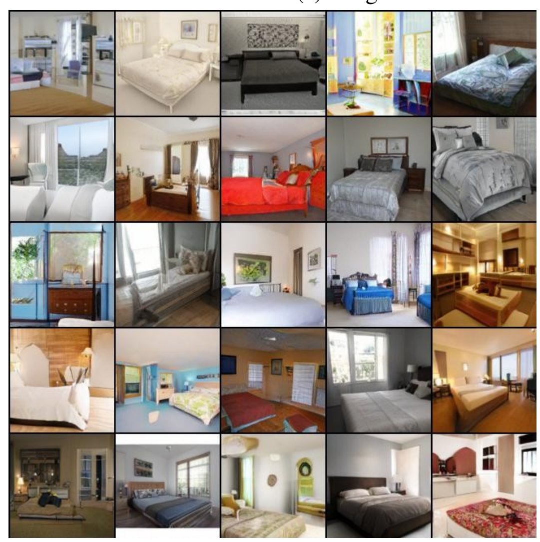
[4] https://github.com/mit-han-lab/data-efficient-gans/tree/master/DiffAugment-stylegan2

(a) CIFAR-10

(b) CIFAR-100

(c) ImageNet-32x32

(d) FFHQ



(e) LSUN-Bedroom

Figure 2: Samples generated by our DISP-Vgg16 approach on large-scale image generation

Implementation details of experiment on 128 Resolution datasets with BigGAN architecture in Section 5.2 of main submission We use our approach in conjunction with existing methodologies in a similar way as the few-shot setting with G_{emb} and D_{emb} as linear transformation matrices which transform the data priors into the generator's conditional input space of dimension 128 and discriminator feature space of dimension 1536. During baseline training, we use self-modulation [3] in the batch-norm layers similar to [4, 13]. In DiffAugment, we use three augmentations: translation, cutout, and color with consistency regularization hyperparameter as 10. During FreezeD training, we freeze the first 4 layers of discriminator. For TransferGAN, FreezeD, MineGAN and its augmentation with DISP, we use the following hyperparameter setting: batch size 256, G and D lr $2e-4$ and $\mathbf{z}$ dimension 120. For DiffAugment, batch size is 32, D-steps is 4 and rest of the hyperparameters are same. Training is done till 30k steps for DiffAugment, FreezeD, and 5k steps for the rest. The moving average weights of the generator are used for evaluation. We use pre-trained network from [5] [2] for finetuning.

Pearson Correlation	Anime	FFHQ	CIFAR-10
D_f cosine vs VGG Perceptual	0.65	0.81	0.80
D_f cosine vs Image L_2	-0.46	-0.61	-0.54

Table 4: Pearson Correlation between cosine similarity in Discriminator feature space (D_f) vs Vgg-16 perceptual similarity/ L_2 distance in Image space on Anime, FFHQ and CIFAR-10 dataset

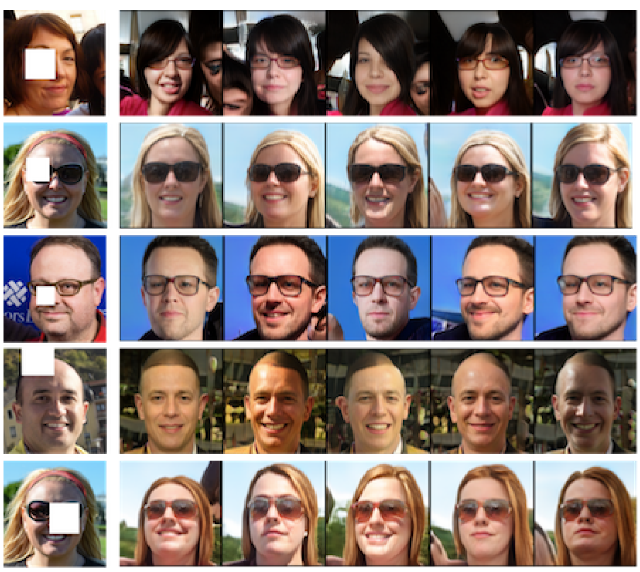
3. Large-Scale Image Generation

Image inversion To invert a query image, $\mathbf{x}_q$ using our trained model, we optimize the prior (after passing it to G_{emb}) that is used to condition each resolution block, independently. Mathematically, we optimize the following

[5] https://github.com/ajbrock/BigGAN-PyTorch

(a) Custom Editing - First column shows human-edited version where certain portion of image is substituted with another to achieve desired semantics. Rest columns correspond to images generated when Vgg16 features of human-edited version is provided as prior to DISP module.

(b) Sketch-to-Image - First column shows sketch describing desired high-level semantics. Rest columns correspond to images generated when Vgg16 features of the sketch version is provided as prior in DISP module.



(c) Inpainting - First column shows a cutout in an Image. Rest columns correspond to images generated when Vgg16 features of the cutout version is provided as prior in DISP module.

(d) Colourization - First column shows gray-scale image describing desired high-level semantics. Rest columns correspond to images generated when Vgg16 features of the gray-scale version is provided as prior in DISP module.

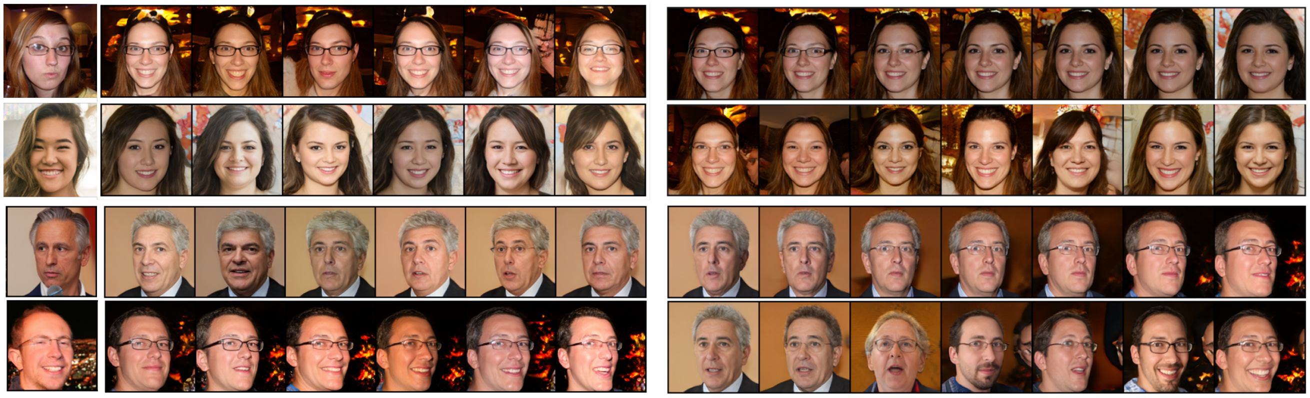
Figure 3: Examples of semantic diffusion used in image manipulation on FFHQ dataset using our DISP-Vgg16 approach. *Top-Left:* Custom Editing; *Top-Right:* Sketch-to-Image; *Bottom-Left:* Inpainting; *Bottom-Right:* Colorization

objective:

$$\mathbf{z}^*, C_1^*, ..C_k^* = arg \min_{\mathbf{z}, C_1, ..C_2} \|G(\mathbf{z}|C_1, ..C_k) - \mathbf{x}_q\|_2^2,$$

$$\mathbf{x}_q^{inv} = G(\mathbf{z}^*|C_1^*, ..C_k^*)$$

Here, C_i (after passing it through G_{emb}) is the prior that is used to condition the $i^{th} \in \{1...k\}$ resolution block.



(a) Semantic Variations - First column corresponds to image whose Vgg16 features are given as prior to DISP module. Rest columns correspond to images generated using random noise. As can be seen the generated images are consistent with the prior image in terms of high-level semantics.

(b) Interpolation - First and last column corresponds to images used for interpolation. Rest columns correspond to images generated when Vgg16 features of images in first and last column are interpolated and given as prior to DISP module.

Figure 4: Semantic variations and interpolation of generated samples using pre-trained Vgg16 conditional DISP module on FFHQ dataset. *(Left (top and bottom):)* Random samples generated with prior as feature of the first column of images in each row; *(Right (top and bottom):)* all 4 rows show interpolation of images between the generated image in the first and last column.

Cosine Similarity	x **and** $G(z\|C(x))$	**Random pair**
VGG perceptual space	0.512 ± 0.067	0.382 ± 0.050
Discriminator's feature space	0.59 ± 0.096	0.50 ± 0.070

Table 5: Similarity between x and $G(z|C(x))$ vs Similarity between a random pair of images from FFHQ dataset.

To get a faster and better convergence, we initialize all C_i as $G_{emb}(C(\mathbf{x}_q))$. The optimization is achieved via back-propagation using Adam optimizer with learning rate of 0.1. Figure 5 (main submission) shows sample inverted images on FFHQ and LSUN-Bedroom datasets. From the figure, we can see that models trained via DISP invert a given query image better than the corresponding baselines.

Equivalence of closeness in latent and image space In our algorithm, we use projection loss in discriminator latent space D_f to enforce that a generated image $G(z|C(x)$ is semantically similar/close to a given image x. And to verify if discriminator latent space is indeed good space to measure similarities, we measure the correlation between cosine similarity in Discriminator feature D_f and Vgg-16 feature (perceptual similarity) space. Vgg-perceptual similarity is an accepted measure of image similarity and has been used in generative models like IMLE, GLANN, BSA as a proxy for constraints in image space. Additionally, we also report the correlation between cosine similarity in Discriminator feature space and L_2 closeness measure in the image space. Table 4 reports our findings where we observe a high positive correlation between cosine similarity in D_f and VGG perceptual similarity; and a moderate negative correlation between cosine similarity D_f in and L_2 distance in Image space.

To quantitatively verify that $G(z|C(x))$ is close to x in the trained model, we also show in Table 5, the perceptual similarity between the two as compared to a random pair of images from FFHQ dataset. We can observe that x and $G(z|C(x)$ are more similar than any random pair of images.

Implementation Details We use a single linear layer to transform the pre-trained image features to the generator's conditional input space of 128 dimensions, and discriminator feature space of 1024 dimensions respectively. A hierarchical latent structure similar to [2] is used during DISP training. During evaluation with K-means and GMM on ImageNet and LSUN-Bedroom we first randomly sample 200K training images and then fit the distribution since clustering on complete training set which is in the order of millions is infeasible. In the training of the unconditional baseline, we use self-modulation [3]. In SSGAN, for rotation loss we use the default parameter of 0.2 for generator and 1.0 for discriminator as mentioned in [4]. For training Self-Conditional GAN [9], we set the number of clusters to 100 for all datasets. For CIFAR-10 and CIFAR-100, we recluster at every 25k iterations with 25k samples, and for ImageNet, at every 75k iterations with 50k samples following default implementation as in [9]. Following standard practice [14], we calculate FID, Precision and Recall between test split and an equal number of generated images for-10, CIFAR-100, and ImageNet 32×32, i.e., 10k, 10k, and 50k, respectively. For FFHQ and LSUN-bedroom datasets, we use 7k and 30k generated and real (disjoint from training) samples, respectively. For all datasets and methods, training is done with batch size of 64, G and D learning rate is set to 0.0002, $\mathbf{z}$ dimension equals 120 and spectral normalization is used in both generator and discriminator networks. Training is done till 100k steps for all datasets except ImageNet which is trained for 200k steps and moving average weights of generator are used during evaluation.

Semantic diffusion for image manipulation We observed that high-level semantics (e.g. hair, gender, glasses, etc in case of faces) of a generated image, $G(\mathbf{z}|C(\mathbf{x}))$, relied on the conditional prior, $C(\mathbf{x})$. Complementarily, variations in the latent code $\mathbf{z} \sim \mathbf{N}(0, I)$ induced fine-grained changes such as skin texture, face shape, etc. This suggests that we can exploit conditional prior, $C(\mathbf{x})$, to get some control over the high-level semantics of generated image. We show that by altering an image $\mathbf{x}$ (through CutMix, CutOut, etc) and using $C(\mathbf{x})$ of the altered image as our new input prior helps in generating samples with the desired attributes, as shown in Fig 3. In a similar manner, DISP also allows generation of images with certain cues (like sketch to image generation, as shown in Fig 3). The generation of samples in this case is simply done by using $C(\mathbf{x})$ as prior in G.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[3] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[4] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12154–12163, 2019.

[5] 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.

[6] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[7] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[8] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, pages 3927–3936, 2019.

[9] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

[11] Alexander Sage, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Logo synthesis and manipulation with clustered generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5879–5888, 2018.

[12] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016.

[13] Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A u-net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8207–8216, 2020.

[14] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[15] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020.