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 x 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Style-GAN 2 (256 x 256)</th>
<th>Panda</th>
<th>Grumpy Cat</th>
<th>Obama</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreezeD</td>
<td>16.69</td>
<td>29.67</td>
<td>62.26</td>
<td></td>
</tr>
<tr>
<td>+ DISP-Vgg16</td>
<td>14.66</td>
<td>29.93</td>
<td>54.87</td>
<td></td>
</tr>
<tr>
<td>DiffAugment</td>
<td>12.06</td>
<td>27.08</td>
<td>46.87</td>
<td></td>
</tr>
<tr>
<td>+ DISP-Vgg16</td>
<td>11.14</td>
<td>28.45</td>
<td>43.79</td>
<td></td>
</tr>
<tr>
<td>BSA*</td>
<td>21.38</td>
<td>34.20</td>
<td>50.72</td>
<td></td>
</tr>
<tr>
<td>GLANN + DISP-Vgg16</td>
<td>11.51</td>
<td>29.85</td>
<td>38.57</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Pre-training</th>
<th>Anime</th>
<th>Faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreezeD</td>
<td>✓</td>
<td>109.40</td>
<td>102.43</td>
</tr>
<tr>
<td>+ DISP-Vgg16</td>
<td></td>
<td>93.36</td>
<td>82.49</td>
</tr>
<tr>
<td>DiffAugment</td>
<td>×</td>
<td>85.16</td>
<td>106.96</td>
</tr>
<tr>
<td>+ DISP-Vgg16</td>
<td></td>
<td>48.67</td>
<td>48.61</td>
</tr>
</tbody>
</table>

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 $2e^{-4}$, z dimension as 120 and maximum number of training steps as

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.
which is a two step training procedure, as follows: (1) Opti-

z

mize for image embeddings \( \{ e_i \} \) of all training images \( \{ x_i \} \) jointly with a generator network \( G \) using perceptual loss; and (2) Learn a sampling function \( T : z \rightarrow e \) through IMLE for generating random images during inference. For using data instance prior in the training procedure of GLANN, instead of directly optimizing for \( \{ e_i \} \), we optimize for the following modified objective:

\[
\arg \min_{G,G_{emb}} \sum_i L_{\text{perceptual}}(G \circ G_{emb} \circ C(x_i), x_i)
\]

where \( \{ e_i \} = \{ G_{emb} \circ C(x_i) \} \)

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 \( z \) 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 2 show the benefit of directly using features as data instance prior instead of only assigning labels based on feature clustering.

<table>
<thead>
<tr>
<th>Method</th>
<th>Anime (SNGAN)</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreezeD + DISP</td>
<td>93.36</td>
<td></td>
</tr>
<tr>
<td>FreezeD + Logo-GAN (K=5)</td>
<td>226.60</td>
<td></td>
</tr>
<tr>
<td>FreezeD + Logo-GAN (K=10)</td>
<td>183.38</td>
<td></td>
</tr>
<tr>
<td>DiffAugment + DISP</td>
<td>48.67</td>
<td></td>
</tr>
<tr>
<td>DiffAugment + Logo-GAN (K=5)</td>
<td>130.54</td>
<td></td>
</tr>
<tr>
<td>DiffAugment + Logo-GAN (K=10)</td>
<td>190.59</td>
<td></td>
</tr>
</tbody>
</table>

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].

---

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

<table>
<thead>
<tr>
<th>Pearson Correlation</th>
<th>Anime</th>
<th>FFHQ</th>
<th>CIFAR-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_f$ cosine vs VGG Perceptual</td>
<td>0.65</td>
<td>0.81</td>
<td>0.80</td>
</tr>
<tr>
<td>$D_f$ cosine vs Image $L_2$</td>
<td>-0.46</td>
<td>-0.61</td>
<td>-0.54</td>
</tr>
</tbody>
</table>

FreezeD, and 5k steps for the rest. The moving average weights of the generator are used for evaluation. We use pre-trained network from [2] for finetuning.

3. Large-Scale Image Generation

Image inversion To invert a query image, $x_q$ 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

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

**Objective:**

\[
\mathbf{z}^*, \mathbf{C}_1^*, \ldots \mathbf{C}_k^* = \arg \min_{\mathbf{z}, \mathbf{C}_1, \ldots \mathbf{C}_2} \| G(\mathbf{z}|\mathbf{C}_1, \ldots \mathbf{C}_k) - \mathbf{x}_q \|_2^2.
\]

\[
\mathbf{x}_q^{inv} = G(\mathbf{z}^*|\mathbf{C}_1^*, \ldots \mathbf{C}_k^*)
\]

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.
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 re-cluster 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 $32 \times 32$; i.e., 10k, 10k, and 50k, respectively. For FFHQ and LSUN-bedroom datasets, we
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, \( G(z|C(x)) \), relied on the conditional prior, \( C(x) \). Complementarily, variations in the latent code \( z \sim \mathcal{N}(0, I) \) induced fine-grained changes such as skin texture, face shape, etc. This suggests that we can exploit conditional prior, \( C(x) \), to get some control over the high-level semantics of generated image. We show that by altering an image \( x \) (through CutMix, CutOut, etc) and using \( C(x) \) 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 5). The generation of samples in this case is simply done by using \( C(x) \) as prior in \( G \).

References


