A cost-effective method for improving and re-purposing large, pre-trained GANs by fine-tuning their class-embeddings

Qi Li[1], Long Mai[2], Michael A. Alcorn[1], and Anh Nguyen[1]

[1] Department of Computer Science and Software Engineering, Auburn University, Auburn AL 36849, USA
{qzl0019, alcorma}@auburn.edu, anh.ng8@gmail.com
[2] Adobe Inc., San Jose CA 95110, USA malong@adobe.com

Abstract. Large, pre-trained generative models have been increasingly popular and useful to both the research and wider communities. Specifically, BigGANs—a class-conditional Generative Adversarial Networks trained on ImageNet—achieved excellent, state-of-the-art capability in generating realistic photos. However, fine-tuning or training BigGANs from scratch is *practically impossible* for most researchers and engineers because (1) GAN training is often unstable and suffering from mode-collapse; and (2) the training requires a significant amount of computation, 256 Google TPUs for 2 days or 8 × V100 GPUs for 15 days. Importantly, many pre-trained generative models both in NLP and image domains were found to contain biases that are harmful to the society. Thus, we need computationally-feasible methods for modifying and re-purposing these huge, pre-trained models for downstream tasks. In this paper, we propose a cost-effective optimization method for improving and re-purposing BigGANs by fine-tuning only the class-embedding layer. We show the effectiveness of our model-editing approach in three tasks: (1) significantly improving the realism and diversity of samples of complete mode-collapse classes; (2) re-purposing ImageNet BigGANs for generating images for Places365; and (3) de-biasing or improving the sample diversity for selected ImageNet classes.

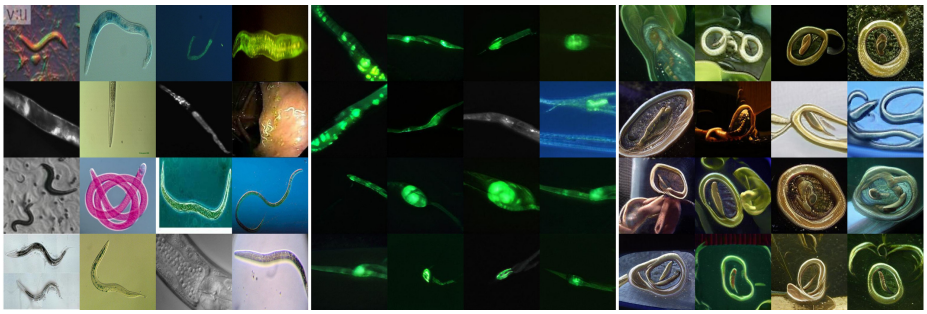
1 Introduction

From GPT-2 [1] to BigGAN [2], large, pre-trained generative models have been increasingly popular and useful to both the research and wider communities. Interestingly, these pre-trained models have remarkably high utility but near-zero re-trainability. That is, GPT-2 or BigGANs were all trained on extremely large-scale computational infrastructure, which is not available to the rest of the community. In practice, training or fine-tuning such models is impossible to most researchers and engineers. Importantly, pre-trained generative models in both text and image domains were found to capture undesired, hidden biases that may be harmful to the society [3,4]. Therefore, the community needs techniques for fine-tuning and re-purposing pre-trained generative models.

The class-conditional BigGAN [2] has reached an unprecedented state-of-the-art image quality and diversity on ImageNet by using large networks and batch sizes. However, fine-tuning or training BigGANs from scratch is impractical for most researchers and engineers due to two main reasons. First, Generative Adversarial Networks (GANs) training is notoriously unstable and subject to mode-collapse [5,2] i.e. the generated distribution does not capture all modes of the true distribution [5]. Consistent with [6], we observed that BigGAN samples from a set of ∼50 classes exhibit substantially lower diversity than samples from other classes do. For example, BigGAN samples from the window screen class are rubbish examples i.e. noisy patterns that are not recognizable to humans (Fig. 1a). Similarly, nematode samples are heavily biased towards green worms on black, but the training data includes worms of a variety of colors and backgrounds (Fig. 1b).

(A) ImageNet (B) BigGAN [2] (C) AM (ours)

(a) Samples from the window screen class (904).



(b) Samples from the nematode class (111).

Fig. 1: For some classes, 256 × 256 BigGAN samples (B) have poor realism and diversity (i.e. samples are biased towards one type of data) while the real ImageNet images (A) are diverse. AM samples (C) are of higher diversity than the original BigGAN samples (B).

Second, re-training BigGANs requires significantly expensive computation—the original 256 × 256 model took 48 hours of training on 256 Google Cloud

TPUs. On more modest hardware of 8 × V100 GPUs, the training is estimated to take more than 2 weeks[7] but has not been found to match the published results in [2]. Importantly, re-training or finetuning BigGANs were found to still cause a set of classes to collapse as observed in a BigGAN-deep model [6] (in addition to BigGAN models) released by [2].

In this paper, we propose a cost-effective method for improving sample diversity of BigGANs and re-purposing it for generating images of unseen classes. Leveraging the intuition that the BigGAN generator is already able to synthesize photo-realistic images for many ImageNet classes [2], we propose to modify *only the class embeddings* while keeping the generator unchanged (Fig. 2). We demonstrate our simple yet effective approach on three different use cases:[3]

1. Changing only the embeddings is surprisingly sufficient to "recover" diverse and plausible samples for complete mode-collapse classes e.g. window screen (Fig. 1a).
2. We can re-purpose a BigGAN, pre-trained on ImageNet, for generating images matching unseen Places365 classes (Sec. 3.2).
3. On ImageNet, our method improves the sample diversity by $\sim 50\%$ for the pre-trained BigGANs released by the authors—at 256×256 and 128×128 resolutions by finding multiple class embeddings for each class (Sec. 3.7). A human study confirmed that our method produced more diverse and similarly realistic images compared to BigGAN samples (Sec. 3.6).

2 Methods

2.1 Problem formulation

Let G be a class-conditional generator, here a BigGAN pre-trained by [2], that takes a class embedding $\boldsymbol{c} \in \mathbb{R}^{128}$ and a latent vector $\boldsymbol{z} \in \mathbb{R}^{140}$ as inputs and outputs an image $G(\boldsymbol{c}, \boldsymbol{z}) \in \mathbb{R}^{256\times256\times3}$. We test improving BigGAN's sample diversity by only updating the embeddings (pre-trained during GAN training).

Increasing Diversity Intuitively, we search for an input class embedding $\boldsymbol{c}$ of the generator G such that the set of output images $\{G(\boldsymbol{c}, \boldsymbol{z}^i)\}$ is diverse with random latent vectors $\boldsymbol{z}^i \sim \mathcal{N}(0, I)$. Specifically, we encourage a small change in the latent variable to yield a large change in the output image [8] by maximizing:

$$\max_{\boldsymbol{c}} \; L_{\mathrm{D}}(\boldsymbol{c}) = \mathbb{E}_{\boldsymbol{z}^i, \boldsymbol{z}^j \sim \mathcal{N}(0,I)} \frac{\left\| \phi(G(\boldsymbol{c}, \boldsymbol{z}^i)) - \phi(G(\boldsymbol{c}, \boldsymbol{z}^j)) \right\|}{\|\boldsymbol{z}^i - \boldsymbol{z}^j\|} \tag{1}$$

where $\phi(.)$ is a feature extractor. In [8], $\phi(.)$ is an identity function to encourage pixel-wise diversity. We also tested with $\phi(.)$ being outputs of the conv5 layer and the output softmax layer of AlexNet.

Via hyperparameter tuning, we found that maximizing the above objective via 10 unique pairs of $(\boldsymbol{z}^i, \boldsymbol{z}^j)$ selected from $\mathcal{Z}$ to be effective (full hyperparameter details are in Sec. 2.4).

[3] Code for reproducibility is available at `https://github.com/qilimk/biggan-am`.

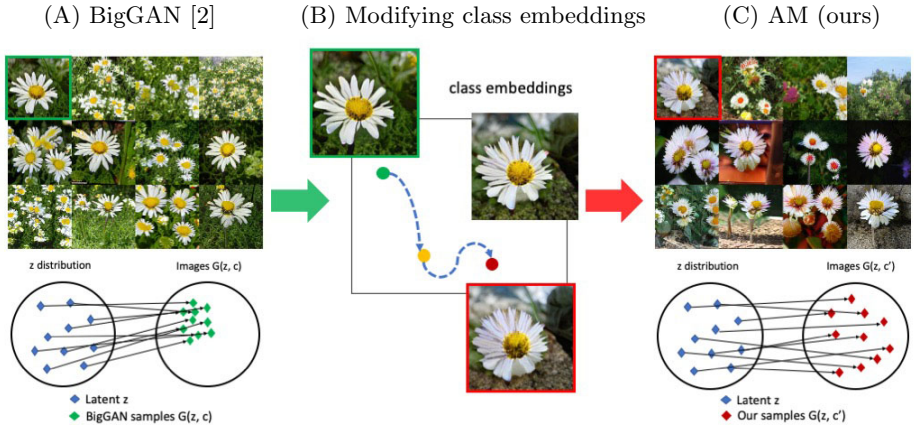


Fig. 2: With BigGAN embeddings (A), the latent $\boldsymbol{z}$ vectors are mapped to nearby points (green ♦) i.e. similarly-looking images. Our embedding optimization moves the original embedding to a new vector where the generated samples (red ♦) are more diverse. Here, the updated class embedding $\boldsymbol{c}$ changes the background of a daisy from green grass (□) to brown soil (□). Note that the pose of the flower (controlled by $\boldsymbol{z}$) remain the same. Effectively, with only a change in the embedding, the latent vectors are re-mapped to more spread-out points or more diverse set of samples (C).

Activation maximization When a class embedding changes, it is critical to keep the generated samples to be still realistic and in the target class. To achieve that, we also move the class embedding $\boldsymbol{c}$ of the generator G such that the output image $G(\boldsymbol{c}, \boldsymbol{z})$ for any random $\boldsymbol{z} \sim \mathcal{N}(0, I)$ would cause some classifier P to output a high probability for a target class $\boldsymbol{y}$ (Fig. 3). Here, we let P be a pre-trained ImageNet classifier [9] that maps an image $\boldsymbol{x} \in \mathbb{R}^{256\times256\times3}$ onto a softmax probability distribution over 1,000 output classes. Formally, we maximize the following objective given a pre-defined class y_c:

$$\max_{\boldsymbol{c}} \; L_{\text{AM}}(\boldsymbol{c}) = \mathbb{E}_{\boldsymbol{z}\sim\mathcal{N}(0,I)} \log P(\boldsymbol{y} = y_c \mid G(\boldsymbol{c}, \boldsymbol{z})) \tag{2}$$

The above objective is basically a common term in the classification objectives for class-conditional GAN discriminators [10,2,11] and also called the Activation Maximization (AM) in image synthesis using pre-trained classifiers [12,13,14,15,16]. We try to solve the above AM objective via mini-batch gradient descent. That is, we iteratively backpropagate through both the classifier P and the generator G and change the embedding $\boldsymbol{c}$ to maximize the expectation of the log probabilities over a set $\mathcal{Z}$ of random latent vectors.

In sum, we encouraged the samples to be diverse but still remain in a target class $\boldsymbol{y}$ via the full objective function below (where λ is a hyperparameter):

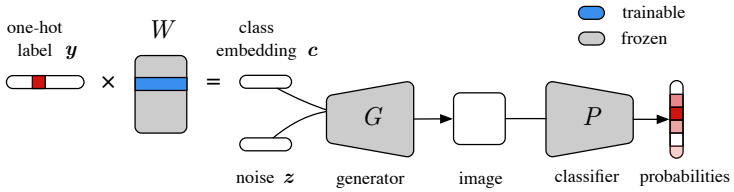


Fig. 3: To improve the samples for a target class represented by a one-hot vector $\boldsymbol{y}$, we iteratively take steps to find an embedding $\boldsymbol{c}$ (i.e. a row in the embedding matrix W) such that all the generated images $\{G(\boldsymbol{c}, \boldsymbol{z}^i)\}$, for different random noise vectors $\boldsymbol{z}^i \sim \mathcal{N}(0, I)$, would be (1) classified as the target class $\boldsymbol{y}$; and (2) diverse i.e. yielding different softmax probability distributions. We backpropagate through both the frozen, pre-trained generator G and classifier P and perform gradient descent to maximize the target-class probability of the generated samples over a batch of random latent vectors $\{\boldsymbol{z}^i\}$.

$$\max_{\boldsymbol{c}} \; L_{\text{AM-D}}(\boldsymbol{c}) = L_{\text{AM}} + \lambda L_{\text{D}} \tag{3}$$

2.2 Datasets and Networks

Datasets While the generators and classifiers were pre-trained on the full 1000-class ImageNet 2012 dataset, we evaluated our methods on a subset of 50 classes (hereafter, ImageNet-50) where we qualitatively found BigGAN samples exhibit the lowest diversity. The selection of 50 classes were informed by two diversity metrics (see below) but decided by humans before the study.

Generators We used two pre-trained ImageNet BigGAN generators [2], a 256×256 and a 128×128 model, released by the authors in PyTorch [7]. For the purpose of studying diversity, all generated images in this paper were sampled from the full, non-truncated prior distribution [2].

2.3 Evaluation metrics

Because there is currently no single metric that is able to capture the multi-dimensional characteristics of an image set [17], we chose a broad range of common metrics to measure sample diversity and sample realism separately.

Diversity We measured intra-class diversity by randomly sampling 200 image pairs from an image set and computing the MS-SSIM [10] and LPIPS [18] scores for each pair. For each method, we computed a mean score across the 50 classes $\times$ 200 image pairs.

Realism To measure sample realism, we used three standard metrics: Inception Score (IS) with 10 splits [19], Fréchet Inception Distance (FID) [20], and Inception Accuracy (IA) [10]. These three metrics were computed for every set of 50,000 images = 50 classes $\times$ 1000 images. To evaluate the set of mixed samples from both BigGAN and AM embeddings, we randomly select 500 images from each and create a new set contains 1000 images per ImageNet class.

2.4 Implementation details

We found two effective strategies for implementing the AM method (described in Sec. 2.1) to improve BigGAN samples: (1) searching within a small region around the original embeddings (AM-S); (2) searching within a large region around the mean embedding (AM-L).

Hyperparameters For AM-S, we randomly initialized the embedding within a Gaussian ball of radius 0.1 around the original embedding. We used a learning rate of 0.01. For AM-L, we randomly initialized the embedding around the mean of all 1000 embeddings and used a larger learning rate of 0.1. For both settings, we maximized Eq. 2 using the Adam optimizer and its default hyperparameters for 200 steps. We re-sampled a set $\mathcal{Z} = \{\boldsymbol{z}^i\}_{20}$ every 20 steps. Every step, we kept the embeddings within $[-0.59, 0.61]$ by clipping. To evaluate each trial, we used the embedding from the last step and sampled 1000 images per class. We ran 5 trials per class with different random initializations. We used 2 to 4 $\times$ V100 GPUs for each optimization trial.

Classifiers In the preliminary experiments, we tested four 1000-class-ImageNet classifiers: AlexNet [9], Inception-v3 [21], ResNet-50 [22], and a ResNet-50 [23] that is robust to pixel-wise noise. By default, we resized the BigGAN output images to the appropriate input resolution of each classifier.

With Inception-v3, we achieved an FID score that is (a) substantially better than those for the other three classifiers (Table S2; 30.24 vs. 48.74), and (b) similar to that of the original BigGAN (30.24 vs. 31.36). The same trends were observed with the Inception Accuracy metrics (Table S2). However, we did not find any substantial qualitative differences among the samples of the four treatments. Therefore, we chose AlexNet because of its fastest run time.

3 Experiments and Results

3.1 Repairing complete mode-collapse classes of BigGANs

Consistent with [6], we found that BigGAN samples for some classes, e.g. window screen, contain similar, human-unrecognizable patterns (see Fig. 1a). However, re-training BigGANs is impractical to most researchers given its significance computation requirement.

Here, we apply AM-L (see Sec. 2.4) to "repair" the mode-collapse window screen embedding to generate more realistic and diverse images. Intuitively, AM-L enables us to make a larger jump out of the local optimum than AM-S.

Results Interesting, by simply changing the embedding, AM-L was able to turn the original rubbish images into a diverse set of recognizable images of window screens (see Fig. 1a). Quantitatively, the AM embedding improved BigGAN window screen samples in all metrics: LPIPS ($0.62 \rightarrow 0.77$), IS ($2.76 \rightarrow 2.91$), and IA ($0.56 \rightarrow 0.7$).

While the embeddings found by our AM methods changed the generated samples entirely, we observed that interpolating in the latent or embedding spaces still yields realistic intermediate samples (Fig. 4).

Fig. 4: Interpolation between a $\boldsymbol{z}$ pair in the window screen class using the original BigGAN embedding (top) yields similar and unrealistic samples. The same interpolation with the embedding found by AM (bottom) produced realistic intermediate samples between two window screen images.

Significantly faster computation According to a PyTorch BigGAN re-implementation by authors [2], BigGAN training can take at least 15 days on 8 V100 GPUs. This is significantly more time-consuming and costly than our AM approach which takes at most 1 hour for generating 5 embeddings (from which users could choose to use one or more) on a single V100 GPU (see Table 1). The original DeepMind's training [2] requires even more expensive and unique hardware of 256 Google Cloud TPU, which is not available to most of the community and so is not compared here.

Method	Time (hours)	Number of GPUs (Tesla V100)	AWS price (USD)
1. BigGAN training [7]	24×15 days=360	8	8812.8
2. AM optimization	1	1	3.1

Table 1: BigGAN training is not only 360× more time-consuming but also almost 3,000× more costly. The AWS on-demand price-per-hour is $ 24.48 for 8×V100 and $ 3.06 for 1×V100 [24].

Note that our method is essentially finding a new sampler for the same BigGAN model. After a new embedding is found via optimization, the samples are generated fast via standard GAN sampling procedure [25].

3.2 Synthesizing Places365 images using pre-trained ImageNet BigGAN

While original BigGAN is not able to synthesize realistic images for all 1000 ImageNet classes (see Fig. 1), it does so for a few hundred of classes.

Therefore, here, we test whether it is possible to re-use the same ImageNet BigGAN generator for synthesizing images for unseen categories in the target Places365 dataset [26], which contains 365 classes of scene images. For evaluation, we randomly chose 50 out of 365 classes in Places365 (hereafter, Places-50).

Mean initialization As we want to generate images for unseen classes, the Places365-optimal embeddings are intuitively far from the original ImageNet embeddings. Therefore, we chose AM-L (instead of AM-S) for making larges jumps. We ran the AM-L algorithm for 5 trials per class with the same hyperparameters as in Sec. 3.1 but with a ResNet-18 classifier [22] pre-trained on Places365.

Top-5 initialization Besides initializing from mean embeddings, we also tested initializing from the top-5 embeddings whose 10 random generated samples were given the highest average accuracy scores by the Places365 classifier. For example, to synthesize the hotel room images for Places365, the top-1 embedding in the ImageNet dataset is for class quilt (Fig. 6). We reproduced 5 AM-L trials but each was initialized with a unique embedding among the top-5.

Baseline We used the original BigGAN samples for the top-1 ImageNet classes found from the top-5 initialization procedure above as a baseline.

Qualitative Results AM-L found many class embeddings that produced plausible images for Places365 scene classes using the same ImageNet BigGAN generator. For example, to match the hotel room class, which does not exist in ImageNet, AM-L synthesized bedroom scenes with lights and windows whereas the top-1 class (quilt) samples mostly shows beds with blankets (Fig. 5). See Fig. 6 for some qualitative differences between the generated images with original vs. AM embeddings for the same set of random latent vectors.

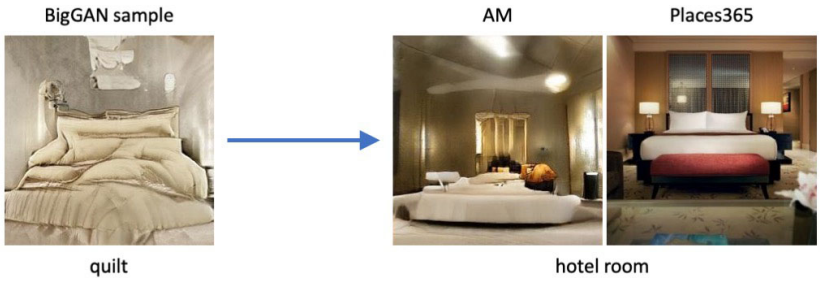


Fig. 5: The closest ImageNet class that the BigGAN was pre-trained to generate is quilt, which contains mostly blankets and pillows. Surprisingly, with AM embeddings, the same BigGAN can generate remarkable images for unseen category of hotel room. The rightmost is an example Places365 image for reference.

Quantitative Results Compared to the baseline, AM-L samples have substantially higher realism in FID (41.25 vs. 53.15) and in ResNet-18 Accuracy scores (0.49 vs. 0.17). In terms of diversity, AM-L and the baseline performed similarly and both were slightly worse than the real images in MS-SSIM (0.42 vs. 0.43) and LPIPS (0.65 vs. 0.70). See Table S3 for detailed quantitative results.

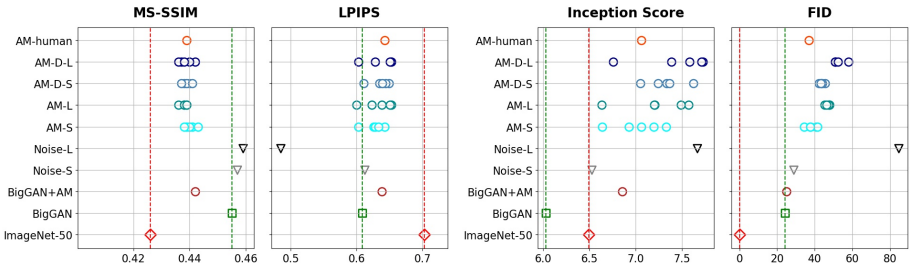
Fig. 6: AM-L generated plausible images for two Places365 classes, plaza (top) and hotel room (bottom), which do *not* exist in the ImageNet training set of the BigGAN generator. For example, AM-L synthesizes images of squares with buildings and people in the background for the plaza class (C) while the samples from the top-1 ImageNet class, here, parking meter, shows parking meters on the street (B). Similarly, AM-L samples for the hotel room class has the unique touches of lighting, lamps, and windows (C) that do not exist in the BigGAN samples for the quilt class (B). The latent vectors are held constant for corresponding images in (B) and (C). See Figs. S21, S22, S23, and S24 for more side-by-side image comparisons.

3.3 Improving sample diversity of 256×256 BigGAN

To evaluate the effectiveness of our method in improving sample diversity for many classes, here, we ran both AM-S and AM-L on 50 classes in ImageNet-50. The goal is to compare the original BigGAN samples vs. a mixed set of samples generated from both the original BigGAN embeddings and AM embeddings found via our AM method. That is, AM optimization is so inexpensive that users can generate many embeddings and use multiple of them to sample images.

BigGAN vs. AM Across 50 classes × 5 AM trials, we found that both AM-S and AM-L produced samples of higher diversity than the original BigGAN samples. For both MS-SSIM and LPIPS, on average, our AM methods reduced the gap between the original BigGAN and the real data by ∼50% (Fig. 7a; AM-S and AM-L vs. BigGAN).

For all 50 classes, we always found at least 1 out of 10 trials (i.e. both AM-S and AM-L combined) that yielded samples that match the real data in MS-



(a) Diversity comparison. (b) Realism comparison.

Fig. 7: Each point in the four plots is a mean score across 50 classes from one AM optimization trial or one BigGAN model. The ultimate goal here is to close the gap between the BigGAN samples (- - - -) and the ImageNet-50 distribution (- - - -) in all four metrics. Naively adding noise degraded the embeddings in both diversity (MS-SSIM and LPIPS) and quality (IS and FID) scores i.e. the black and gray ∇ actually moved away from the red lines. Our optimization trials, on average, closed the *diversity* gap by ∼50% i.e. the AM circles are half way in between the green and red dash lines (a). By mixing AM samples with the original BigGAN samples, the BigGAN+AM image-set (○) has substantially higher diversity (MS-SSIM and LPIPS) and similar quality (IS and FID) to BigGAN (□). That is, that multi-embeddings improved the sample diversity of BigGAN without compromising the quality.

SSIM or LPIPS scores. The statistics also align with our qualitative observations that AM samples often contain a more diverse set of object poses, shapes and backgrounds than the BigGAN samples (see Figs. S9–S11).

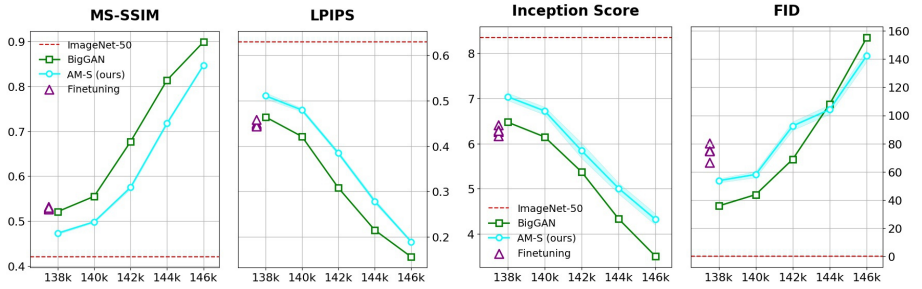
BigGAN vs. BigGAN+AM Most importantly, the set of images generated by both BigGAN and two AM embeddings obtained higher diversity in MS-SSIM and LPIPS while obtaining similar realism FID scores (Fig. 7; BigGAN vs. BigGAN+AM). We constructed each BigGAN+AM set per class using one BigGAN and one AM embedding (selected by humans out of 5 embeddings).

3.4 Adding noise to or finetuning the class embeddings did not improve diversity

Adding noise A naive attempt to improve sample diversity is adding small random noise to the embedding vector of a low-diversity class. Across 50 classes, we found that adding small noise $\sim \mathcal{N}(0, 0.1)$ almost did not quantitatively change the image quality and diversity (Fig. 7; Noise-S) while adding larger noise $\sim \mathcal{N}(0, 0.3)$ degraded the samples on both criteria (Fig. 7; Noise-L).

For example, daisy samples gradually turned into human-unrecognizable rubbish images as we increased the noise (Fig. S4).

Finetuning Another strategy to improve sample diversity is to finetune BigGANs. However, how to finetune a BigGAN to improve its sample diversity is



(a) Diversity comparison (b) Realism comparison

Fig. 8: Each point in the four plots is a mean score across 50 classes and five AM-S trials or one 128×128 BigGAN model. Finetuning the 138k snapshot neither improved the sample diversity nor realism (purple Δ vs. green $\square$). Optimizing the embeddings via AM-S consistently improved the diversity in both MS-SSIM and LPIPS (a). IS and FID metrics disagree on whether AM-S (cyan $\circ$) sample quality is better or worse than that of the BigGAN samples. See Fig. 9 for a side-by-side comparison of the samples from these five snapshots.

an open question. The BigGAN pre-trained model would start to degrade if we kept training it using the original hyperparameters as reported in [2].

To minimize the GAN training instability and compare with other approaches in this paper, we only finetuned one embedding at a time, keeping the other embeddings and all parameters in the generator and discriminator frozen. Because [2] only released the discriminator for their 128×128 generator but not for the 256×256 model, we only finetuned the 128×128 model. For each class, we added a small amount of noise $\sim \mathcal{N}(0, 0.1)$ to the associated embedding vector and finetuned it using the original BigGAN training objective for 10 iterations until the training collapsed. Across 50 classes $\times$ 5 trials, quantitatively, finetuning did not improve the sample diversity but lowered the realism (Fig. 8; purple Δ vs. green $\square$).

3.5 Explicitly encouraging diversity yielded worse sample realism

Inspired by [8], here, we used the sample diversity further by incorporating a diversity term into the previous two AM-S and AM-L methods (Sec. 2.1) to produce two new variants AM-D-S and AM-D-L. We tested encouraging diversity in the (1) image space; (2) `conv5` feature space; and (3) softmax outputs of AlexNet and found they can qualitatively bias the optimization towards different interesting spaces of diversity.

However, the addition of the diversity term quantitatively improved the diversity but at a large cost of lower sample quality (Fig. 7b AM-S vs. AM-D-S and AM-L vs. AM-D-L). Similarly, the IA scores of the AM-D methods were consistently lower than those of the original AM methods (Table S1). See Sec. S1 for more details.

We hypothesize that the intrinsic noise from mini-batch SGD [27] also contributes to the increased sample diversity caused by AM embeddings.

3.6 Humans rated AM samples more diverse and similarly realistic

Because quantitative image evaluation metrics are imperfect [17], we ran a human study to compare the AM vs. original BigGAN samples. For each class, across all 20 embeddings from 5 trials $\times$ 4 methods (AM-S, AM-L, AM-D-S, and AM-D-L), we manually chose one embedding that qualitatively is a balance between diversity and realism to sample images to represent our AM method in the study. As a reference, this set of AM images were more diverse and less realistic than BigGAN samples according to the quantitative metrics (Fig. 7; AM-human vs. BigGAN).

Experiments We created two separate online surveys for diversity and realism, respectively. For each class, the diversity survey showed a panel of 8×8 AM images side-by-side a panel of 8×8 BigGAN samples and asked participants to rate which panel is more diverse on the scale of 1–5. That is, 1 or 5 denotes the left or right panel is clearly more diverse, while 3 indicates both sets are similarly diverse. For each class, the AM and BigGAN panels were randomly positioned left or right. The realism survey was a duplicate of the diversity except that each panel only showed 3×3 images so that participants could focus more on the details.

Results For both tests, we had 52 participants who are mostly university students and do not work with Machine Learning or GANs. On average, AM samples were rated to be more diverse and similarly realistic compared to BigGAN samples. That is, AM images were given better than the neutral score of 3, i.e. 2.24 ± 0.85 in diversity and 2.94 ± 1.15 in realism.

Also, AM samples were rated to be more diverse in 42/50 classes and more realistic in 22/50 classes. See Figs. S9–S11 for your own comparisons.

3.7 Generalization to a 128×128 BigGAN

To test whether our method generalizes to a different GAN at a lower resolution, we applied our AM-S method (see Sec. 3.1) to a pre-trained 128×128 BigGAN released by [7]. As in previous experiments, we ran 50 classes $\times$ 5 trials in total. To evaluate each trial, we used the last-step embedding to sample 1000 images per class.

Consistent with the result on the 256×256 resolution, here, AM-S improved the diversity over the pre-trained model on both MS-SSIM and LPIPS (Fig. 8a; 138k). In terms of quality, FID and IS showed a mixed result of whether AM-S sample realism is lower or higher. See Fig. S17 for side-by-side comparisons.

3.8 Generalization to different training snapshots of 128×128 BigGAN

We have shown that BigGAN sample diversity can be improved substantially by changing only the embeddings (Sec. 3.1) which revealed that the generator was

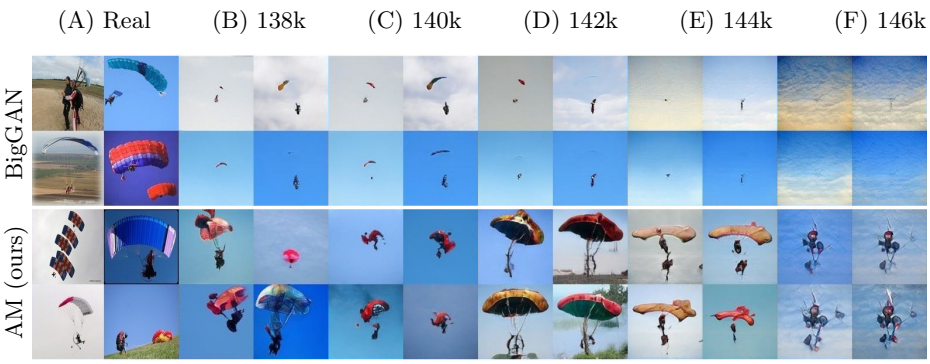


Fig. 9: For the parachute class, the original 128×128 BigGAN samples (top panel) mostly contained tiny parachutes in the sky (B) and gradually degraded into blue sky images only (C–F). AM (bottom panel) instead exhibited a more diverse set of close-up and far-away parachutes (B) and managed to paint the parachutes for nearly-collapsed models (E–F). The samples in this figure correspond to the five snapshots (138k—146k) reported in the quantitative comparison in Fig. 8. See Figs. S6, S7, S8 for more qualitative comparisons.

actually capable of synthesizing those diverse images. Here, we test how much sample diversity and quality can be improved by AM as the BigGAN training gradually collapses, which might impair not only the embeddings but also the generator's parameters.

Experiments We took the pre-trained 128×128 BigGAN model (saved at the 138k-th iteration) and continued training it for 9000 more iterations with the same hyperparameters as in [7]. We applied the AM-S method with the same hyperparameters as in Sec. 3.7 to four BigGAN snapshots captured at the 140k, 142, 144k, and 146k iteration, respectively.

Results AM-S consistently improved the sample diversity of all snapshots. For some classes, AM qualitatively improved both sample diversity and quality (Figs. 9 and S6–S8). However, the diversity and realism of both AM-S and the original BigGAN samples gradually dropped together (Fig. 8; AM-S vs. BigGAN). The result suggests that, as the GAN training gradually collapsed, the synthesis capability is so degraded that changing the class embeddings alone is not sufficient to significantly improve the samples.

4 Related work

Latent space traversal Searching in the latent space of a GAN generator network to synthesize images has been shown effective for many tasks including (1) in-painting [28]; (2) image editing [29]; (3) creating natural adversarial examples [30]; or (4) feature visualization [14]. While all prior work in this line of research

optimized the latent variable $\boldsymbol{z}$, we instead optimize the class embeddings $\boldsymbol{c}$ of a class-conditional generator over a set of random $\boldsymbol{z}$ vectors.

Our method might be the most related to Plug & Play Generative Networks (PPGN) [13] in that both methods sample from the distribution $p_G(\boldsymbol{x}, \boldsymbol{y})$ jointly defined by a generator and a pre-trained classifier. While [13] trained an unconditional generator that inverts the features of an ImageNet classifier, our method is generally applicable to any pre-trained class-conditional generator. Importantly, our goal is novel—to improve the sample diversity of any pre-trained class-conditional generator (here, BigGANs) by changing its class embeddings.

Improving sample quality Two methods, MH-GAN [31] and DRS [32], have recently been proposed to improve the samples of a pre-trained GAN by harnessing the discriminator to reject low-probability generated samples. However, these methods are able to only improve sample *quality* but not diversity. In addition, they assume that the discriminator is (a) available, which may not always be the case e.g. in the official BigGAN releases [2]; and (b) optimally trained for their samplers to recover exactly the true distribution. Similar to MH-GAN and PPGN, our method is similar to a Markov chain Monte Carlo (MCMC) sampler that has no rejection steps. A major difference is that we only perform the iterative optimization *once* to update the embedding matrix. After a desired embedding is found, our subsequent samplings of images are fast following standard GANs. In contrast, MH-GAN, DRS, and PPGN samplers often require many rejection or update steps to produce a single image.

Generalization Understanding the image synthesis capability of a trained GAN generator is an active research area. Recent findings showed that GANs trained on a dataset of scene images contain neurons that can paint common objects such as "trees" or "doors" [33]. [34] found that BigGAN is able to perform some general image transforms such as zoom, rotate or brightness adjustment up to a certain limit. However, these methods optimize only the latent variable [34] or both the latent and the generator parameters [33], but not the class embeddings as ours.

5 Conclusion

We showed that the low sample diversity of pre-trained GAN generators can be improved by simply changing the class embeddings, not the generator. Note that one could "recover" the missing modes using our AM methods and improve the sample quality further by sampling from a truncated prior distribution [2]. Our method is also a promising method for de-biasing GAN models. Compared to finetuning or re-training BigGANs from scratch, our method is more tractable even considering that one has to run five 200-step optimization trials to find a desired class embedding.

References

1. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1** (2019) 9
2. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations. (2019)
3. Johnson, K.: Ai weekly: A deep learning pioneer's teachable moment on ai bias — venturebeat. `https://venturebeat.com/2020/06/26/ai-weekly-a-deep-learning-pioneers-teachable-moment-on-ai-bias/` (2020) (Accessed on 07/08/2020).
4. Sheng, E., Chang, K.W., Natarajan, P., Peng, N.: The woman worked as a babysitter: On biases in language generation. arXiv preprint arXiv:1909.01326 (2019)
5. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. (2017)
6. Ravuri, S., Vinyals, O.: Seeing is not necessarily believing: Limitations of biggans for data augmentation. (2019)
7. Brock, A.: ajbrock/biggan-pytorch: The author's officially unofficial pytorch biggan implementation. `https://github.com/ajbrock/BigGAN-PyTorch` (2019) (Accessed on 07/25/2019).
8. Yang, D., Hong, S., Jang, Y., Zhao, T., Lee, H.: Diversity-sensitive conditional generative adversarial networks. In: International Conference on Learning Representations. (2019)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
10. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org (2017) 2642–2651
11. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
12. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., Clune, J.: Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In: Advances in Neural Information Processing Systems. (2016) 3387–3395
13. Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., Yosinski, J.: Plug & play generative networks: Conditional iterative generation of images in latent space. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 4467–4477
14. Nguyen, A., Yosinski, J., Clune, J.: Understanding neural networks via feature visualization: A survey. arXiv preprint arXiv:1904.08939 (2019)
15. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341** (2009) 1
16. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
17. Borji, A.: Pros and cons of gan evaluation measures. Computer Vision and Image Understanding **179** (2019) 41–65

18. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 586–595
19. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in neural information processing systems. (2016) 2234–2242
20. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in Neural Information Processing Systems. (2017) 6626–6637
21. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2818–2826
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
23. Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., Madry, A.: Learning perceptually-aligned representations via adversarial robustness. arXiv preprint arXiv:1906.00945 (2019)
24. Amazon: Amazon ec2 p3 instance product details. `https://aws.amazon.com/ec2/instance-types/p3/` (2020) (Accessed on 07/07/2020).
25. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014)
26. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 1452–1464
27. Wu, J., Hu, W., Xiong, H., Huan, J., Braverman, V., Zhu, Z.: On the noisy gradient descent that generalizes as sgd. arXiv preprint arXiv:1906.07405 (2019)
28. Yeh, R.A., Chen, C., Yian Lim, T., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 5485–5493
29. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: European Conference on Computer Vision, Springer (2016) 597–613
30. Zhao, Z., Dua, D., Singh, S.: Generating natural adversarial examples. In: International Conference on Learning Representations. (2018)
31. Turner, R., Hung, J., Frank, E., Saatchi, Y., Yosinski, J.: Metropolis-Hastings generative adversarial networks. In Chaudhuri, K., Salakhutdinov, R., eds.: Proceedings of the 36th International Conference on Machine Learning. Volume 97 of Proceedings of Machine Learning Research., Long Beach, California, USA, PMLR (2019) 6345–6353
32. Azadi, S., Olsson, C., Darrell, T., Goodfellow, I., Odena, A.: Discriminator rejection sampling. In: International Conference on Learning Representations. (2019)
33. Bau, D., Zhu, J.Y., Strobelt, H., Zhou, B., Tenenbaum, J.B., Freeman, W.T., Torralba, A.: Visualizing and understanding generative adversarial networks. In: International Conference on Learning Representations. (2019)
34. Jahanian, A., Chai, L., Isola, P.: On the"steerability" of generative adversarial networks. arXiv preprint arXiv:1907.07171 (2019)