Overcoming Mode Collapse with Adaptive Multi Adversarial Training

Karttikeya Mangalam* University of California at Berkeley
mangalam@cs.berkeley.edu
Rohin Garg*† University of California at San Diego
rgarg@ucsd.edu

Abstract

Generative Adversarial Networks (GANs) are a class of generative models used for various applications, but they have been known to suffer from the *mode collapse* problem, in which some modes of the target distribution are ignored by the generator. Investigative study using a new data generation procedure indicates that the mode collapse of the generator is driven by the discriminator's inability to maintain classification accuracy on previously seen samples, a phenomenon called Catastrophic Forgetting in continual learning. Motivated by this observation, we introduce a novel training procedure that adaptively spawns additional discriminators to remember previous modes of generation. On several datasets, we show that our training scheme can be plugged-in to existing GAN frameworks to mitigate mode collapse and improve standard metrics for GAN evaluation. Code and pre-trained models are available at https://github.com/gargrohin/AMAT

1 Introduction

Generative Adversarial Networks (GANs) [8] are an extremely popular class of generative models used for text and image generation in various fields of science and engineering, including biomedical imaging [31, 44, 45], autonomous driving [13, 47], and robotics [3, 34]. However, GANs are widely known to be prone to *mode collapse*, which refers to a situation where the generator only samples a few modes of the real data, failing to faithfully capture other more complex or less frequent categories. While the mode collapse problem is often overlooked in text and image generation tasks, and even traded off for higher realism of individual samples [4, 15], dropping infrequent classes may cause serious problems in real-world problems, in which the infrequent classes represent important anomalies. For example, a collapsed GAN can produce racial/gender biased images [25].

Moreover, mode collapse causes instability in optimization, which can damage both diversity and the realism of individual samples of the final results. As an example, we visualized the training progression of the vanilla GAN [8] for a simple bimodal distribution in the top row of Figure 1. At collapse, the discriminator conveniently assigns high realism to the region unoccupied by the generator, regardless of the true density of real data. This produces a strong gradient for the generator to move its samples toward the dropped mode, swaying mode collapse to the other side. So, the discriminator loses its ability to detect fake samples it was previously able to, such as point $\mathbf{X}_\bullet$. The oscillation continues without convergence.

* Equal technical contribution. † Work done during internship.

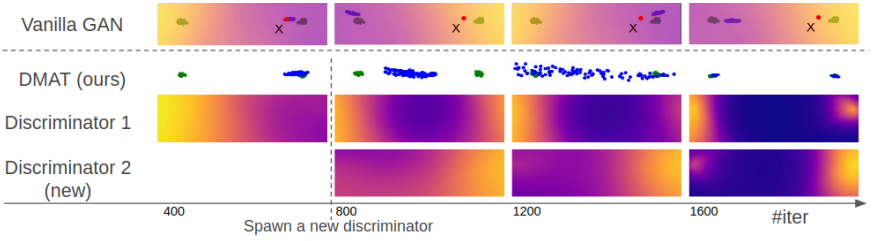

Figure 1: **Visualizing training trajectories**: Distribution of real (green dots) and fake (blue dots) over the course of vanilla GAN (top row) and our method (the second row and below). The background color indicates the prediction heatmap of the discriminator with blue being fake and warm yellow being real. Once the vanilla GAN falls into mode collapse (top row), it ends up oscillating between the two modes without convergence. Also, the discriminator's prediction at point X oscillates, indicating catastrophic forgetting in the discriminator. AMAT algorithm adapts to the need, and a new discriminator is spawned during training which effectively learns the forgotten mode, guiding the GAN optimization toward convergence.

We observe that the mode collapse problem is closely related to Catastrophic Forgetting [23, 24, 35] in continual learning. A promising line of works [7, 33, 37, 46] tackle the problem in the supervised learning setting by instantiating multiple predictors, each of which takes charge in a particular subset of the whole distribution. We also tackle the problem of mode collapse in GAN by tracking the severity of Catastrophic Forgetting by storing a few exemplar data during training, spawning an additional discriminator if forgetting is detected, Figure 1. The key idea is that the added discriminator is left intact unless the generator recovers from mode dropping of that sample, essentially sidestepping catastrophic forgetting.

We show that our proposed approach based on adaptive addition of discriminators can be added to any of the existing GAN frameworks, and is most effective in preventing mode collapse. Furthermore, the improved stability of training boosts the standard metrics on popular GAN frameworks. To summarize, our contributions are: *First*, we propose a novel GAN framework, named Adaptive Multi Adversarial Training (AMAT), that effectively prevents Catastrophic Forgetting in GANs by spawning additional discriminators during training. *Second*, we also propose a computationally efficient synthetic data generation procedure for studying mode collapse in GANs that allows visualizing high dimensional data using normalizing flows. We show that mode collapse occurs even in the recent robust GAN formulations. *Third*, our method can be plugged into any state-of-the-art GAN frameworks and still improve the quality and coverage of the generated samples.

2 Related Works

Previous works have focused on independently solving either catastrophic forgetting in supervised learning or mode collapse during GAN training. In this section we review these works in detail and discuss our commonalities and differences.

2.1 Mitigating Mode Collapse in GANs

Along with advancement in the perceptual quality of images generated by GAN [4, 15, 16, 28], a large number of papers [1, 6, 15, 22, 26, 27, 30, 39] identify the problem of mode collapse in GANs and aim to mitigate it. However mode collapse was seen as a secondary symptom that would be naturally solved as the stability of GAN optimization progresses [1, 2, 26]. To explicitly address mode collapse, Unrolled GAN [27] proposes an unrolled optimization of

$g(\mathbf{z}) =$	1		$\mathbf{A}_{392\times2}$		$\mathbf{z}$		MLP		MLP, $\mathbf{A}_{392\times2}$		MNIST
Label	Level I		Level II		Level III		Level IV		Level V		-
GAN-NS [8]	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
WGAN [1]	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✓
Unrolled GAN [27]	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
D2GAN [30]	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
GAN-NS + AMAT	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓

Table 1: ✓ indicates that the generator could effectively learn all the data modes, while ✗ means *despite best efforts with tuning* the training suffers from mode collapse (more than a quarter of data modes dropped). We show results with the SGD (left) & ADAM (right) optimizers. MNIST results with ADAM optimizer are provided for reference. We observe that MNIST is a relatively easy dataset, falling between `Level I` and `II` in terms of complexity.

the discriminator to optimally match the generator objective, thus preventing mode collapse. VEEGAN [39] utilizes the reconstruction loss on the latent space. PacGAN [22] feeds multiple samples of the same class to the discriminator when making the decisions about real/fake. In contrast, our approach can be plugged into existing state-of-the-art GAN frameworks to yield additional performance boost.

2.2 Multi-adversarial Approaches

The idea of employing more than one adversarial network in GANs to improve results has been explored by several previous works independent of the connection to continual learning and catastrophic forgetting. MGAN [12] uses multiple generators, while D2GAN [30] uses two discriminators, and GMAN [6] and MicrobatchGAN [29] proposed a method with more than two discriminators that can be specified as a training hyperparameter beforehand. However, all previous works require the number of discriminators to be fixed beforehand, which is a major drawback since it depends on several intricate factors such as training dynamics, data distribution complexity, model architecture, initialization hyper-parameters etc. and is expensive and difficult to approximate even with several runs of the algorithm. In contrast, noting by the connection of multi-adversarial training to parameter expansion approaches to catastrophic forgetting, we propose an *adaptive* method that can add discriminators incrementally during training thus achieving superior performance than existing works both on data quality metrics as well as overall computational effort.

2.3 Overcoming Catastrophic Forgetting in GAN

Methods to mitigate catastrophic forgetting can be categorized into three groups: a) regularization based methods [18] b) memory replay based methods [36] c) network expansion based methods [33, 47]. Our work is closely related to the third category of methods, which dynamically adds more capacity to the network, when faced with novel tasks. This type of methods, adds *plasticity* to the network from new weights (fast-weights) while keeping the *stability* of the network by freezing the past-weights (slow-weights). Additionally, we enforce stability by letting a discriminator to focus on a few set of classes, not by freezing its weights.

The issue of catastrophic forgetting in GANs has been sparsely explored before. Chen et al. [5] and Tran et al. [42] propose a self-supervised learning objective to prevent catastrophic forgetting by adding new loss terms. Liang et al. [21] proposes an online EWC based solution to tackle catastrophic forgetting in the discriminator. We propose a prominently different approach based on parameter expansion rather than regularization. While the regularization based approaches such as Liang et al. [21] attempt to retain the previously learnt knowledge by constrained weight updates, the parameter expansion approaches effectively sidestep

catastrophic forgetting by freezing previously encoded knowledge. Thanh-Tung and Tran [40] also discuss the possibility of catastrophic forgetting in GAN training but their solution is limited to theoretical analyses with simplistic proposals such as assigning larger weights to real samples and optimizing the GAN objective with momentum. Practically, we observed that their method performs worse than a plain vanilla DCGAN on simple real world datasets like CIFAR10. In contrast, our method leverages insights from continual learning and has a direct connections to prevalent parameter expansion approaches in supervised learning. We benchmark extensively on several datasets and state-of-the-art GAN approaches where our method consistently achieves superior results to the existing methods.

3 Proposed Method

In this section, we first describe our proposed data generation procedure that we use as a petri dish for studying mode collapse in GANs. The procedure uses random normalizing flows for simultaneously allowing training on complex high dimensional distributions yet being perfectly amenable to 2D visualizations. Next, we describe our proposed Adaptive Multi Adversarial Training (AMAT) algorithm that effectively detects catastrophic forgetting and spawns a new discriminator to prevent mode collapse.

3.1 Synthetic Data Generation with Normalizing flows

Mode dropping in GANs in the context of catastrophic forgetting of the discriminator is a difficult problem to investigate using real datasets. This is because the number of classes in the dataset cannot be easily increased, the classes of fake samples are often ambiguous, and the predictions of the discriminator cannot be easily visualized across the whole input space. In this regard, we present a simple yet powerful data synthesis procedure that can generate complex high dimensional multi-modal distributions, yet maintaining perfect 2-D visualization capabilities. Samples from a 2-D Gaussian distribution are augmented with biases and subjected to an invertible normalizing flow [14] parameterized by well conditioned functions $g_i : \mathbb{R}^{d_i^0} \rightarrow \mathbb{R}^{d_i^1}$. This function can be followed by a linear upsampling transformation parameterized by a $d_i^1 \times d_{i+1}^0$ dimensional matrix A^i (Algorithm 1). The entire transform is deliberately constructed to be a bijective function so that every generated sample in $\hat{y} \in \mathbb{R}^D$ can be analytically mapped to $\mathbb{R}^2$, allowing perfect visualization on 2D space. Furthermore, by evaluating a dense grid of points in $\mathbb{R}^2$, we can understand discriminator's learned probability distribution on $\mathbf{z}$ manifold as a heatmap on the 2D plane. This synthetic data generation procedure enables studying mode collapse in a controlled setting. This also gives practitioners the capability to train models on a chosen data complexity with clean two-dimensional visualizations of both the generated data and the discriminator's learnt distribution. This tool can be used for debugging new algorithms using insights from the visualizations. In the case of mode collapse, a quick visual inspection would give the details of which modes face mode collapse or get dropped from discriminator's learnt distribution.

3.2 Adaptive Multi Adversarial Training

Building upon the insight on relating catastrophic forgetting in discriminator to mode collapse in generator, we propose a multi adversarial generative adversarial network training procedure. The key intuition is that the interplay of catastrophic forgetting in the discriminator with the GAN minimax game, leads to an oscillation generator. Thus, as the generator shifts to a new set of modes the discriminator forgets the learnt features on the previous modes.

Algorithm 1 Synthetic Data Generation

Input: Mean $\{\mu_i\}_{i=1}^{K}$ and standard deviation $\{\sigma_i\}_{i=1}^{K}$ for initialization, $\{g_i\}_{i=1}^{L}$ well conditioned $\mathbb{R}^2 \to \mathbb{R}^2$ functions
Sample weights $\boldsymbol{w} \sim \text{Dirichlet}(K)$
/* Sample from 2D gaussian mixture */
$\mathbf{x}_{2D} \sim \Sigma_{i=1}^{N} w_i \mathcal{N}(\mu_i, \sigma_i)$
$\mathbf{x}_{2D}^0 = \left[[x_{2D}^0; 1], [x_{2D}^1; 1]\right]$
/* Randomly Init Normalizing Flow */
for $k = 1$ to $k = K$ **do**
  if k is even **then**
    $\boldsymbol{x}^k = [\boldsymbol{x}_0^k, \boldsymbol{x}_1^k \cdot g_k(\boldsymbol{x}_0^k)]$
  else
    $\boldsymbol{x}^k = [\boldsymbol{x}_0^k \cdot g_k(\boldsymbol{x}_1^k), \boldsymbol{x}_1^k]$
  end if
end for

Algorithm 2 DSPAWN: Discriminator Spawning Routine

Require: Exemplar Data $\{\boldsymbol{e}\}_{i=1}^{m}$
Input: Discriminator set $\mathbb{D} = \{f_w^i\}_{i=1}^{K}$
/* Check forgetting on exemplars */
for $i = 1$ to $i = m$ **do**
  $\boldsymbol{s}[k] \leftarrow f_w^k(\boldsymbol{e}_i)\ \forall\ k \in \{1 \dots K\}$
  if $K * \max(\boldsymbol{s}) > \alpha_t * \sum_k \boldsymbol{s}[k]$ **then**
    Initialize f_w^{K+1} with random weights w
    /* Spawn a new discriminator */
    Initialize random weight w^{K+1}
    $\mathbb{D} \leftarrow \{f_w^i\}_{i=1}^{K} \bigcup f_w^{K+1}$
    break
  end if
end for
return Discriminator Set $\mathbb{D}$

Algorithm 3 A-MAT: Adaptive Multi-Adversarial Training

Require: $\boldsymbol{w}_0^i$, $\boldsymbol{\theta}_0$ initial discriminator & generator params, greediness param ε, $\{T_k\}$ spawn warmup iteration schedule
$\mathbb{D} \leftarrow \{f_w^0\}$
while $\boldsymbol{\theta}$ has not converged **do**
  Sample $\{\boldsymbol{z}^{(i)}\}_{i=1}^{B} \sim p(z)$
  Sample $\{\boldsymbol{x}^{(i)}\}_{i=1}^{B} \sim \mathbb{P}_r$
  Sample $\{\sigma_1(i)\}_{i=1}^{B} \sim \text{Uniform}(1, K)$
  Sample $\{\alpha(i)\}_{i=1}^{B} \sim \text{Bernoulli}(\varepsilon)$
  /* Loss weights over discriminators */
  Sample weights $\boldsymbol{m} \sim \text{Dirichlet}(K)$
  $\hat{\boldsymbol{x}}^{(i)} \leftarrow g_\theta(\boldsymbol{z}^{(i)})$
  $\sigma_2(i) \leftarrow \arg\min_k f_w^k(\hat{\boldsymbol{x}}^{(i)})$
  /* Discriminator responsible for $\hat{x}^{(i)}$ */
  $\sigma_z(i) \leftarrow \alpha(i)\sigma_1(i) + (1 - \alpha(i))\sigma_2(i)$
  /* Discriminator responsible for $x^{(i)}$ */
  $\sigma_x(i) \leftarrow \sigma_1(i)$
  /* Training Discriminators */
  $L_w \leftarrow \sum_{i=1}^{B} [f_w^{\sigma_x(i)}(\boldsymbol{x}_i) - 1]^- - [f_w^{\sigma_z(i)}(\hat{\boldsymbol{x}}_i) + 1]^+$
  for $k = 1$ to $k = |\mathbb{D}|$ **do**
    $w^k \leftarrow \text{ADAM}(L_w)$
  end for
  /* Training Generator */
  $s[k] \leftarrow \sum_{i=1}^{B} f_w^k(\hat{\boldsymbol{x}}^{(i)})\ \forall\ k \in \{1 \dots |\mathbb{D}|\}$
  /* Weighed mean over discriminators */
  $L_\theta \leftarrow \text{sort}(\boldsymbol{m}) \cdot \text{sort}(s)$
  $\theta \leftarrow \text{ADAM}(L_\theta)$
  if more than T_t warm-up iterations since the last spawn **then**
    $\mathbb{D} \leftarrow \text{DSPAWN}(\{f_w^i\})$
  end if
end while

However if there are multiple discriminators available, each discriminator can implicitly *specialize* on a subset of modes. Thus even if the generator oscillates, each discriminator can remember their own set of modes, and they will not need to move to different set of modes. This way we can effectively *sidestep* forgetting and ensure the networks do not face significant distribution shift. A detailed version of our proposed method is presented in Algorithm 3.

Spawning new discriminators: We initialize the AMAT training Algorithm 3 with a regular GAN using just one discriminator. We also sample a few randomly chosen exemplar data points with a maximum of one real sample per mode, depending on dataset complexity.

The exemplar data points are used to detect the presence of catastrophic forgetting in the currently active set of discriminators $\mathbb{D}$ and spawn a new discriminator if needed. Specifically (Algorithm 2), we propose that if *any* discriminator among $\mathbb{D}$ has an unusually high score over an exemplar data point $\boldsymbol{e}_i$, this is because the mode corresponding to $\boldsymbol{e}_i$ has either very poor generated samples or has been entirely dropped.

In such a situation, if training were to continue we risk catastrophic forgetting in the active set $\mathbb{D}$, if the generator oscillate to near $\boldsymbol{e}_i$. This is implemented by comparing the max score over at $\boldsymbol{e}_i$ to the average score over and spawning a new discriminator when the ratio exceeds $\alpha_t(>1)$. Further, we propose to have $\alpha_t(>1)$ a monotonically increasing function of $|\mathbb{D}|$, thus successively making it harder to spawn each new discriminator. Additionally, we use a warmup period T_t after spawning each new discriminator from scratch to let the spawned discriminator train before starting the check over exemplar data-points.

Multi-Discriminator Training: We evaluate all discriminators in $\mathbb{D}$ on the fake samples but do not update all of them for all the samples. Instead, we use the discriminator scores to assign responsibility of each data point to only one discriminator.

Training over fake samples: We use an ε-greedy approach for fake samples where the discriminator with the lowest output score is assigned responsibility with a probability $1-\varepsilon$ and a random discriminator is chosen with probability ε.

Training over real samples: The discriminator is always chosen uniformly randomly thus we slightly prefer to assign the same discriminator to the fake datapoints from around the same mode to ensure that they do not forget the already learnt modes and switch to another mode. The random assignment of real points ensure that the same preferentially treated discriminator also gets updated on real samples.

Further for optimization stability, we ensure that the real and fake sample loss incurred by each discriminator is roughly equal in each back-propagation step by dynamically reweighing them by the number of data points the discriminator is responsible for. We only update the discriminator on the losses of the samples they are responsible for. While it may seem that adding multiple discriminators makes the procedure expensive, in practice the total number of added discriminator networks never surpass three for the best results. It is possible to change the hyperparameters to allow a large number of discriminators but that results in sub-optimal results and incomplete training. The optimal hyperparameter selection is explained in the Appendix for each dataset. Further, the additional discriminators get added during later training stages and are not trained from the start, saving compute in comparison to prior multi-adversarial works which train all the networks from the beginning. Also, unlike AdaGAN [41] and similar Boosted GAN models that need to store multiple Generators post training, the final number of parameters required during inference remains unchanged under AMAT . Thus the inference time remains the same, but with enhanced mode coverage and sample diversity. Unlike [41], our discriminator addition is adaptive, i.e. discriminators are added during the training thus being more efficient.

Generator Training: We take a weighted mean over the discriminators scores on the fake datapoints for calculating the generator loss. At each step, the weights each discriminator in $\mathbb{D}$ gets is in decreasing order of its score on the fake sample. Hence, the discriminator with the lowest score is given the most weight since it is the one that is currently specializing on the mode the fake sample is related to. In practice, we sample weights randomly from a

	GAN	UnrolledGAN	D2GAN	RegGAN	DCGAN	with AMAT
# Modes covered	628.0 ± 140.9	817.4 ± 37.9	1000 ± 0.0	955.5 ± 18.7	849.6 ± 62.7	$\mathbf{1000 \pm 0.0}$
KL (samples $\|\|$ data)	2.58 ± 0.75	1.43 ± 0.12	0.080 ± 0.01	0.64 ± 0.05	0.73 ± 0.09	0.078 ± 0.01

Table 2: **Quantitative Results on the Stacked MNIST dataset**: Applying our proposed adaptive multi adversarial training (AMAT) procedure to a simple DCGAN achieves perfect mode coverage, better than many existing methods for mode collapse.

Model	D2GAN	MicrobatchGAN	GAN-NS w/ ResNet	AMAT + GAN-NS	DCGAN	AMAT + DCGAN
IS	7.15 ± 0.07	6.77	6.7 ± 0.06	$\mathbf{8.1 \pm 0.04}$	6.03 ± 0.05	$\mathbf{6.32 \pm 0.06}$
FID	-	-	28.91	**16.35**	33.42	**30.14**
Model	WGAN-GP w/ ResNet	AMAT + WGAN-GP	SN-GAN	AMAT + SN-GAN	BigGAN	AMAT + BigGAN
IS	7.59 ± 0.10	$\mathbf{7.80 \pm 0.07}$	8.22 ± 0.05	$\mathbf{8.34 \pm 0.04}$	9.22	$\mathbf{9.51 \pm 0.06}$
FID	19.2	**17.2**	14.21	**13.8**	8.94	**6.11**

Table 3: **Quantitative Results on CIFAR10**: We benchmark AMAT against several other multi-adversarial baselines as well as on several GAN architectures across all of which we observe a consistent performance increase.

Dirichlet distribution (and hence implicitly they sum to 1) and sort according to discriminator scores to achieve this. We choose soft weighing over hard binary weights because since the discriminators are updated in an ε greedy fashion, the discriminators other than the one with the best judgment on the fake sample might also hold useful information. Further, we choose the weights randomly instead of fixing a chosen set to ensure AMAT is more deadset agnostic since the number of discriminator used changes with the dataset complexity so does the number of weights needed. While a suitable function for generating weights can work well on a particular dataset, we found random weights to work as well across different settings.

4 Results

We test our proposed method on several synthetic and real datasets & report a consistent increase in performance on GAN evaluation metrics such as Inception Score [38] and Frechét Inception Distance [11] with our proposed AMAT. We also showcase our performance in the GAN fine-tuning regime with samples on the CUB200 dataset [43] which qualitatively are more colorful and diverse than an identical BigGAN without AMAT (Figure 3).

4.1 Synthetic Data

We utilize the proposed synthetic data generation procedure with randomly initialized normalizing flows to visualize the training process of a simple DCGAN [32]. Figure 1 visualizes such a training process for a simple bimodal distribution. Observing the pattern of generated samples over the training iteration and the shifting discriminator landscape, we note a clear mode oscillation issue present in the generated samples driven by the shifting discriminator output distribution. Focusing on a single fixed real point in space at any of the modes, we see a clear oscillation in the discriminator output probabilities strongly indicating the presence of catastrophic forgetting in the discriminator network.

Effect of Data Complexity on Mode Collapse: We use the flexibility in choosing transfor-

Effect Ablation	Large $\|\mathbb{D}\|$ Small α, Short T_t	Spawn too late Long T_t schedule	Greedy ∇D $\epsilon = 0$	Random for fake ϵ-greedy for real	**1**-hot weight vector $\boldsymbol{m}$	Proposed Method
IS	8.83 ± 0.04	9.28 ± 0.08	9.31 ± 0.06	8.95 ± 0.04	9.25 ± 0.05	9.51 ± 0.06
FID	14.23	9.37	8.6	12.5	9.25	6.11

Table 4: **BigGAN + AMAT Ablations on CIFAR10** (A) A spawning condition with small α and short warmup schedule that leads to large number of discriminators (>7) (B) Long warm-up schedules that spawn new discriminators late into training (C) A greedy strategy for assigning responsibility of fake samples ($\epsilon = 0$) (D) Flipping the data splitting logic with responsibilities of fake samples being random and of real being ϵ-greedy (E) Choosing the discriminator with lowest score for updating Generator instead of soft random weighting.

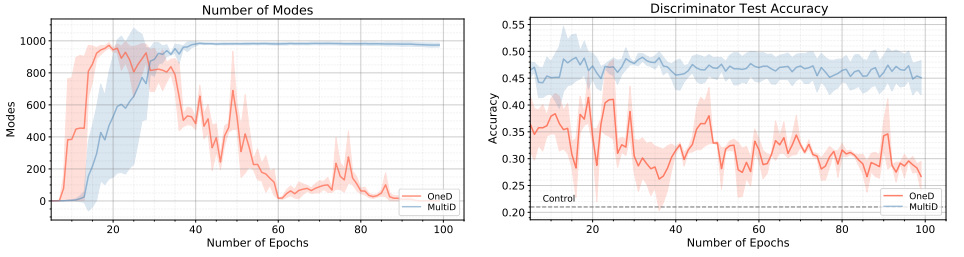

Figure 2: **Investigating the forgetting-collapse interplay:** We investigate our hypothesis that catastrophic forgetting is associated with mode collapse. On the left pane, we plot the magnitude of mode collapse by counting the number of modes produced by the generator. On the right pane, we assess the quality of the discriminator features by plotting the accuracy of linear classifier on top of the discriminator features at each epoch. In the original model, the coverage of modes and the quality of discriminator features are both low and decreasing. In particular, the test accuracy from the discriminator's features drops almost to randomly initialized weights (shown as *control*). On the other hand, adding AMAT (*MultiD*) dramatically improves both mode coverage and the discriminator test accuracy.

mations g_i to generate datasets of various data distribution complexities as presented in Table 1. Choosing $g(z)$ with successively more complicated transformations can produce synthetic datasets of increasing complexity, the first five of which we roughly classify as `Levels`. The `Levels` are generated by using simple transforms such as identitym constant mapping, small Multi layer perceptrons and well conditioned linear transforms (**A**).

On this benchmark, we investigate mode collapse across different optimizers such as SGD & ADAM [17] on several popular GAN variants such as the non-saturating GAN Loss (GAN-NS) [8], WGAN [1] and also methods targeting mitigating mode collapse specifically such as Unrolled GAN [27] and D2GAN [30]. We show results of our proposed AMAT training procedure with a simple GAN-NS, which matches performance with other more complicated mode collapse specific GAN architectures, all of which are robust to mode collapse up to `Level IV`. In practice we find all benchmarked methods to collapse at `Level V`. Thus, in contrast to other simple datasets like MNIST [19], Gaussian ring, or Stacked MNIST [27], the complexity of our synthetic dataset can be arbitrarily tuned up or down to gain insight into the training and debugging of GAN via visualizations.

4.2 Stacked MNIST

We also benchmark several models on the Stacked MNIST dataset following [27, 39]. Stacked MNIST is an extension of the popular MNIST dataset [20] where each image is expanded in the channel dimension to $28 \times 28 \times 3$ by concatenating 3 single channel images. The resulting

Classes	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Avg
BigGAN	24.23	12.32	24.85	21.21	12.81	22.74	17.95	13.16	12.11	18.39	8.94
+ AMAT	20.50	10.30	23.48	18.48	11.51	19.41	11.50	12.24	10.69	12.94	6.11
Δ%	18.2	19.6	5.8	14.8	11.3	17.2	**56.1**	7.5	11.7	**42.1**	**46.3**

Table 5: **Per-class FID on CIFAR10**: FID improves consistently across all classes.

dataset has a 1000 overall modes. We measure the number of modes covered by the generator as the number of classes that are generated at least once within a pool of $25,600$ sampled images. The class of the generated sample is identified with a pretrained MNIST classifier operating channel wise on the original stacked MNIST image.

Understanding the forgetting-collapse interplay: In Section 1, we discuss our motivation for studying catastrophic forgetting for mitigating mode collapse. We also design an investigative experiment to explicitly observe this interplay by comparing the number of modes the generator learns against the quality of features the discriminator learns throughout GAN training on the stacked MNIST dataset. We measure the number of modes captured by the generator through a pre-trained classification network trained in a supervised learning fashion and frozen throughout GAN training. To measure the amount of *'forgetting'* in discriminator, we extract features of real samples from the penultimate layer of the discriminator and train a small classifier on the real features for detecting real data mode. This implicitly indicates the quality and information contained in the the discriminator extracted features. However, the performance of classification network on top of discriminator features is confounded by the capacity of the classification network itself. Hence we do a control experiment, where we train the same classifier on features extracted from a randomly initialized discriminator, hence fixing a lower-bound to the classifier accuracy. Referring to Figure 2, we observe a clear relation between the number of modes the generator covers at an iteration and the accuracy of the classification network trained on the discriminator features at the same iteration. In the vanilla single discriminator scenario, the classification accuracy drops significantly, indicating a direct degradation of the discriminative features which is followed by a complete collapse of G. In the collapse phase, the discriminator's learnt features are close to random with the classification accuracy being close to that of the control experiment. This indicates the presence of significant catastrophic forgetting in the the discriminator network.

In contrast, training the same generator with the proposed AMAT procedure leads to stable training with almost all the modes being covered. The classification accuracy increasing before saturation. Catastrophic forgetting is *effectively sidestepped* by adaptive multi adversarial training which produces stable discriminative features during training that provide a consistent training signal to the generator thereby covering all the modes with little degradation.

4.3 CIFAR10

We extensively benchmark AMAT on several GAN variants including unconditional methods such as DCGAN [32], ResNet-WGAN-GP [9, 10] & SNGAN [28] and also conditional models such as BigGAN [4]. Table 3 shows the performance gain on standard GAN evaluation metrics such as Inception Score and Fréchet distance of several architectures when trained with AMAT procedure. The performance gains indicate effective curbing of catastrophic forgetting in the discriminator with multi adversarial training. We use the public evaluation code from SNGAN [28] for evaluation. Despite having components such as spectral normalization, diversity promoting loss functions, additional R1 losses & other stable training tricks that

Figure 3: **Sample Diversity on CUB200:** We showcase samples from a BigGAN pretrained on imagenet & finetuned on CUB200 with the AMAT procedure (left) and from an identical BigGAN finetuned without AMAT (right). Observe that while the sample quality is good for both, the samples generated with AMAT are more colorful & diverse, with bright reds and yellow against several backgrounds. While the samples from vanilla fine-tuning are restricted to whites/grays & only a hint of color.

might affect catastrophic forgetting to different extents, we observe a consistent increase in performance across all models. Notably the ResNet GAN benefits greatly with AMAT despite a powerful backbone – with IS improving from 6.7 to 8.1, indicating that the mode oscillation problem is not mitigated by simply using a better model.

AMAT improves performance by over 35% even on a well performing baseline such as BigGAN (Table 3). We investigate classwise FID scores of a vanilla BigGAN and an identical BigGAN + AMAT on CIFAR10 and report the results in Table 5. Performance improves across all classes with previously poor performing classes such as 'Frog' & 'Truck' experiencing the most gains. Further, we ablate several key components of AMAT procedure on the BigGAN architecture with results reported in Table 4. We observe all elements to be critical to overall performance. Specifically, having a moderate α schedule to avoid adding too many discriminators is critical. Another viable design choice is to effectively flip the algorithm's logic and instead choose the fake points randomly while being ϵ greedy on the real points. This strategy performs well on simple datasets but loses performance with BigGAN on CIFAR10 (Table 4). In all experiments, the computational time during inference is the same as the base model, *irrespective of the number of discriminators* added during the training, since only a single generator is trained with AMAT and all the discriminators are discarded.

5 Conclusion

In summary, motivated from the observation of catastrophic forgetting in the discriminator, we propose a new adaptive GAN training framework that adds additional discriminators to prevent mode collapse. We show that our method can be added to existing GAN frameworks to prevent mode collapse, generate more diverse samples and improve FID & IS. In future, we plan to apply AMAT to fight mode collapse in high resolution image generation settings.

Acknowledgements

We thank Jathushan Rajasegaran for his helping with the forgetting-collapse interplay experiments and Taesung Park for feedback and comments on the early drafts of this paper.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the International Conference Computer Vision (ICCV)*, 2019.

[3] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250. IEEE, 2018.

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

[5] Ting Chen, Xiaohua Zhai, and Neil Houlsby. Self-supervised gan to counter forgetting. *arXiv preprint arXiv:1810.11598*, 2018.

[6] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

[7] Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[8] 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.

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

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

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

[12] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. Mgan: Training generative adversarial nets with multiple generators. In *International Conference on Learning Representations*, 2018.

[13] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018.

[14] Mahdi Karami, Dale Schuurmans, Jascha Sohl-Dickstein, Laurent Dinh, and Daniel Duckworth. Invertible convolutional flow. In *Advances in Neural Information Processing Systems*, pages 5635–5645, 2019.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. 2019.

[16] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. 2020.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[19] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Kevin J Liang, Chunyuan Li, Guoyin Wang, and Lawrence Carin. Generative adversarial network training is a continual learning problem. *arXiv preprint arXiv:1811.11083*, 2018.

[22] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in neural information processing systems*, pages 1498–1507, 2018.

[23] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

[24] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

[25] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine learning (ICML)*, 2018.

[27] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

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

[29] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. microbatchgan: Stimulating diversity with multi-adversarial discrimination. *arXiv preprint arXiv:2001.03376*, 2020.

[30] Tu Nguyen, Trung Le, Hung Vu, and Dinh Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2670–2680, 2017.

[31] Dong Nie, Roger Trullo, Jun Lian, Li Wang, Caroline Petitjean, Su Ruan, Qian Wang, and Dinggang Shen. Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering*, 65(12):2720–2730, 2018.

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

[33] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. In *Advances in Neural Information Processing Systems 32*, pages 12669–12679. Curran Associates, Inc., 2019.

[34] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. Rl-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.

[35] Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[36] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[37] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

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

[39] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pages 3308–3318, 2017.

[40] Hoang Thanh-Tung and Truyen Tran. On catastrophic forgetting and mode collapse in gans. *arXiv preprint arXiv:1807.04015*, 2020.

[41] Ilya O Tolstikhin, Sylvain Gelly, Olivier Bousquet, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. Adagan: Boosting generative models. In *Advances in neural information processing systems*, pages 5424–5433, 2017.

[42] Ngoc-Trung Tran, Viet-Hung Tran, Bao-Ngoc Nguyen, Linxiao Yang, et al. Self-supervised gan: Analysis and improvement with multi-class minimax game. In *Advances in Neural Information Processing Systems*, pages 13253–13264, 2019.

[43] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[44] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Išgum. Generative adversarial networks for noise reduction in low-dose ct. *IEEE transactions on medical imaging*, 36(12):2536–2545, 2017.

[45] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical image analysis*, 58:101552, 2019.

[46] Jeffrey O. Zhang, Alexander Sax, Amir Zamir, Leonidas J. Guibas, and Jitendra Malik. Side-tuning: Network adaptation via additive side networks. 2019.

[47] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic testing and input validation framework for autonomous driving systems. In *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 132–142. IEEE, 2018.