Catastrophic forgetting and mode collapse in GANs

1st Hoang Thanh-Tung
Applied Artificial Intelligence Institute
Deakin University
hoangtha@deakin.edu.au

2nd Truyen Tran
Applied Artificial Intelligence Institute
Deakin University
truyen.tran@deakin.edu.au

***Abstract*—In this paper, we show that Generative Adversarial Networks (GANs) suffer from catastrophic forgetting even when they are trained to approximate a single target distribution. We show that GAN training is a continual learning problem in which the sequence of changing model distributions is the sequence of tasks to the discriminator. The level of mismatch between tasks in the sequence determines the level of forgetting. Catastrophic forgetting is interrelated to mode collapse and can make the training of GANs non-convergent. We investigate the landscape of the discriminator's output in different variants of GANs and find that when a GAN converges to a good equilibrium, real training datapoints are wide local maxima of the discriminator. We empirically show the relationship between the sharpness of local maxima and mode collapse and generalization in GANs. We show how catastrophic forgetting prevents the discriminator from making real datapoints local maxima, and thus causes non-convergence. Finally, we study methods for preventing catastrophic forgetting in GANs.**

***Index Terms*—GANs, generative, catastrophic forgetting, mode collapse**

I. Introduction

GANs [1, 2] are a powerful tool for modeling complex distributions. Training a GAN to approximate a single target distribution is often considered as a single task. In this paper, we introduce a novel view of GAN training as a continual learning problem in which the sequence of changing model distributions are considered as the sequence of tasks. We discover a surprising result that GANs suffer from catastrophic forgetting, a problem often observed in continual learning settings [3]. Catastrophic forgetting (CF) in artificial neural networks [4, 5, 6] is the problem where the knowledge of previously learned tasks is abruptly destroyed by the learning of the current task. When a GAN suffers from CF, it exhibits undesired behaviors such as mode collapse and non-convergence.

In section III, we show that GAN training is actually a continual learning problem and demonstrate the CF problem on a number of datasets. We show that catastrophic forgetting and mode collapse [1] are two different but interrelated problems and together, they can make the training of GANs non-convergent (section III-B, IV-B). To avoid mode collapse and improve convergence, it is important to address the CF problem. We identify 2 factors that causes CF in GANs: 1) Information from previous tasks is not used in the current task, 2) Knowledge from previous tasks is not usable for the current task and vice versa. Our findings shed light on how to avoid catastrophic forgetting to learn the target distribution properly (Section V).

In section IV, we investigate the effect of CF and mode collapse on the landscape of the discriminator's output. We find that when a GAN converge to a good local equilibrium without mode collapse, real datapoints are wide local maxima of the discriminator. We show that the sharper the local maxima are, the more severe mode collapse is. Section IV-B shows that when CF happen, the discriminator is directionally monotonic. A GAN with a directionally monotonic discriminator does not converge to an equilibrium. The fact confirms that CF is a cause of non-convergence.

Section V explains how state-of-the-art methods for stabilizing GANs such as Wasserstein GAN [7, 8], zero-centered gradient penalty on training examples (GAN-R1) [9], zero-centered gradient penalty on interpolated samples (GAN-0GP) [10], and optimizers with momentum, can prevent CF and mode collapse. Finally, we introduce a new loss function that helps preventing CF while adding zero computational overhead.

Contributions:

1) We detect the CF problem in GANs.
2) We show the relationship between CF, mode collapse, and non-convergence.
3) We study the relationship between the sharpness of local maxima and mode collapse.
4) We show that CF tends to make the discriminator directionally monotonic around real datapoints.
5) We identify the causes of CF and explain the effectiveness of methods for preventing CF in GANs.

II. Related works

Convergence. Prior works on the convergence of GANs usually consider the convergence in parameter space [9, 11, 12, 13]. However, convergence in parameter space tells little about the quality of the equilibrium that a GAN converge to. For example, Thanh-Tung et al. demonstrated that TTUR [12] can make GAN converge to collapsed equilibrium. Consensus Optimization [13] can introduce spurious local equilibria with unknown properties to the game.

We directly study the behaviors of GANs in the data space. By analyzing the discriminator's output landscape, we find that when a GAN converges, real datapoints are local maxima of the discriminator. We discover the relationship between the sharpness of local maxima and mode collapse, generalization.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Catastrophic forgetting. Seff et al. [14] studied the standard continual learning setting in which a GAN is trained to generate samples from a set of distributions introduced sequentially. The problem is solved by the direct application of continual learning algorithms such as Elastic Weight Consolidation (EWC) [3] to GANs. Liang et al. [15] independently came up with a similar intuition that GAN training is a continual learning problem.[1] The paper, however, did not study the causes and effects of the problem and focused on applying continual learning algorithms to address catastrophic forgetting in GANs. We focus on explaining the causes and effect of the problem and its relationship to mode collapse and non-convergence.

III. Catastrophic forgetting problem in GANs

A. GANs training as continual learning problems

Let us consider a GAN with generator $G(\cdot;\boldsymbol{\theta}) : \mathbb{R}^{d_z} \to \mathbb{R}^d$, a continuous function with parameter $\boldsymbol{\theta} \in \mathbb{R}^m$; and discriminator $D(\cdot;\boldsymbol{\psi}) : \mathbb{R}^d \to \mathbb{R}$, a continuous function with parameter $\boldsymbol{\psi} \in \mathbb{R}^n$. G transforms a d_z-dimensional noise distribution p_z to a d-dimensional model distribution p_g that approximates a d-dimensional target distribution p_r. D maps d-dimensional inputs to 1-dimensional outputs. Let $\mathcal{L}_D$ be the loss function for D, $\mathcal{L}_G$ be the loss function for G (Table I). In practice, G and D are neural networks trained by alternating SGD [1].

At each iteration of the training process, G is updated to better fool D. p_g^t, the model distribution at iteration t, is different from the model distribution at the previous iteration p_g^{t-1} and the next iteration p_g^{t+1}. The knowledge required to separate p_g^t from p_r is different from that for the pair $\{p_g^{t-1}, p_r\}$. $\{p_g^{t-1}, p_r\}$ and $\{p_g^t, p_r\}$ are two different classification tasks to the discriminator.[2] The sequence of changing model distributions $\left\{p_g^i\right\}_{i=1}^T$ and the target distribution p_r form a sequence of tasks $\left\{\mathcal{T}^i = \{p_g^i, p_r\}\right\}_{i=1}^T$ to the discriminator. Because the generator at iteration t, G^t, can only generate samples from p_g^t, D^t, the discriminator at iteration t, cannot access samples from previous model distributions $p_g^{<t}$. That makes the learning process of D a continual learning problem. Similarly, the generator has to fool a sequence of changing discriminators $\{D^i\}_{i=1}^T$. The training process of a GAN poses a different continual learning problem to each of the players. In this paper, we focus on the continual learning problem in the discriminator as many prior works have showed that the quality of a GAN mainly depends on its discriminator [16, 17, 18].

If the sequence p_g^t converges to a distribution p_g^*, then the sequence of tasks $\left\{\mathcal{T}^i\right\}_{i=1}^T$ converges to a single task of separating 2 distributions p_g^* and p_r. In practice, however, the sequence of model distributions does not always converge. Nagarajan and Kolter [11] formally proved that the players in Wasserstein GAN [7] do not converge to an equilibrium but oscillate in a small cycle around the equilibrium. Although non-saturating GAN (GAN-NS) [1] was proven to be convergent under strong assumptions [11, 12], Fedus et al. [19] observed that on many real world datasets, the distance between p_g^t and p_r (measured in KL-divergence and Jensen-Shannon divergence) does not decrease as t increases. The authors suggested that p_g can approach p_r in many different and unpredictable ways. These results imply that in the most common variants of GANs, p_g^t can be arbitrarily different from p_g^{t-n} for large n. If the knowledge used for separating p_g^t and p_r cannot be used for separating p_g^{t-n} and p_r, a discriminator trained on $\mathcal{T}^t$ could forget $\mathcal{T}^{t-n}$, i.e. it classifies samples in $\mathcal{T}^{t-n}$ wrongly (Fig. 11b). When this happens, we say that the discriminator exhibits catastrophic forgetting behaviors.

B. Catastrophic forgetting in GANs

1) Catastrophic forgetting on synthetic dataset: We begin by analyzing the problem on the 8 Gaussian dataset, a dataset generated by a mixture of 8 Gaussians placed on a circle. In Fig. 1, red datapoints are generated samples, blue datapoints are real samples. The discriminator and generator are 2 hidden layer MLP with 64 hidden neurons. ReLU activation function was used. p_z is a 2-dimensional standard normal distribution. SGD with constant learning rate of $\alpha = 3 \times 10^{-3}$ was used for both networks. The vector at a datapoint $\boldsymbol{x}$ shows the negative gradient $-\partial\mathcal{L}_G/\partial\boldsymbol{x}$. The vector shows the direction in which $\mathcal{L}_G$ decreases the fastest. The length of the vector corresponds to the speed of change in $\mathcal{L}_G$. Because the gradient field is conservative, the the difference between the loss of two datapoints $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ is:

$$\mathcal{L}_G(\boldsymbol{x}_0) - \mathcal{L}_G(\boldsymbol{x}_1) = \int_{\mathcal{C}} \boldsymbol{v} \cdot d\boldsymbol{s} \tag{1}$$

where $\boldsymbol{v} = -\partial\mathcal{L}_G/\partial\boldsymbol{x}$ and $\mathcal{C}$ is a path from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$. For the variants in Table I, $\partial\mathcal{L}_G/\partial\boldsymbol{x}$ only depends on $\boldsymbol{x}$ and D. Because decreasing $\mathcal{L}_G$ in these GANs corresponds to increasing $D(\boldsymbol{x})$, going in the direction of $-\partial\mathcal{L}_G/\partial\boldsymbol{x}$ increases the score $D(\boldsymbol{x})$. Let $\boldsymbol{y}_0 = G(\boldsymbol{z}_0),\ \boldsymbol{z}_0 \sim p_z$ be a fake datapoint. Updating $\boldsymbol{y}_0$ with SGD with a small enough learning rate will move $\boldsymbol{y}_0$ in the direction of $-\partial\mathcal{L}_G/\partial\boldsymbol{y}_0$ by a distance proportional to $\|-\partial\mathcal{L}_G/\partial\boldsymbol{y}_0\|$. If the discriminator is fixed, then SGD updates will move $\boldsymbol{y}_0$ along its integral curve, in the direction of increasing $D(\boldsymbol{y}_0)$.[3]

Fig. 1a - 1d show the evolution of a GAN-NS on 8 Gaussian dataset. In Fig. 1a - 1c, the discriminator assigns higher score to datapoints that are further away from the fake datapoints, regardless of the true labels of these points. This is shown by the gradient vectors pointing away from the fake datapoints. The integral curves do not converge to any real datapoints. If D is fixed, updating G with gradient descent makes p_g diverges. Because gradients w.r.t. different fake datapoints have

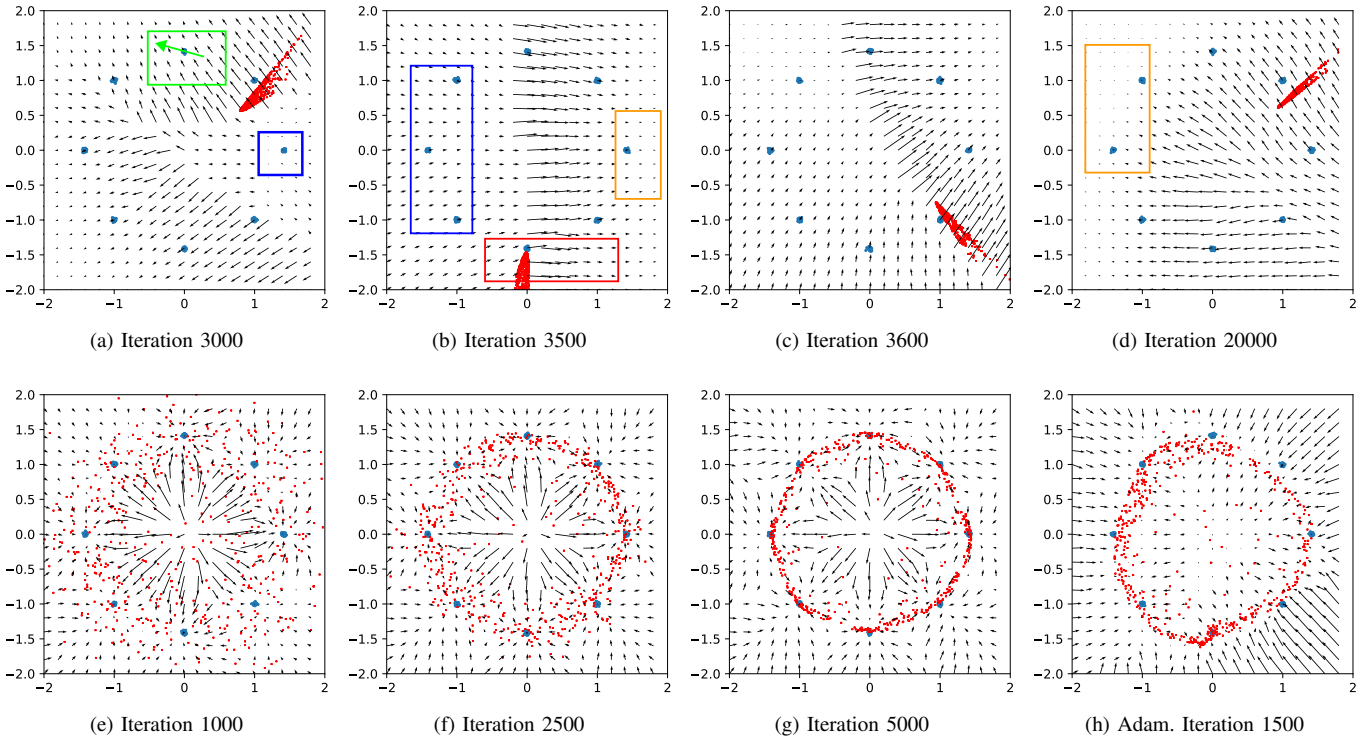
[1] Liang et al. came up with the idea a few months after us. They agreed that we are the first to consider the catastrophic forgetting problem in a single GAN. Their preprint has not been published at any conferences or journals.

[2] In the original theoretical formulation of GAN, at every GAN iteration, the discriminator and the generator are trained until convergence [1]. That means p_g^t can be arbitrarily different from p_g^{t-1}. In practice, at each iteration, only a limited number of gradient updates are applied to the players. We can consider a chunk of consecutive model distributions as a task to the discriminator.

[3] In practice, gradient updates are not applied to $\boldsymbol{y}_0$ but to the generator's parameters. Because the generator also minimizes $\mathcal{L}_G$, gradient updates to the generator move $\boldsymbol{y}_0$ in a direction that approximates $-\partial\mathcal{L}_G/\partial\boldsymbol{y}_0$. $-\partial\mathcal{L}_G/\partial\boldsymbol{y}_0$ is a good approximation of the direction that $\boldsymbol{y}_0$ will move in the next iteration.

	$\mathcal{L}_D$	$\mathcal{L}_G$
WGANGP	$-\mathbb{E}_{\boldsymbol{x}\sim p_r}[D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[D(G(\boldsymbol{z}))] + \lambda\mathbb{E}_{\boldsymbol{u}}[(\|(\nabla D)_{\boldsymbol{u}}\| - 1)^2]$ where $\boldsymbol{u} = \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}; \boldsymbol{x} \sim p_x, \boldsymbol{y} \sim p_g, \alpha \sim \mathcal{U}(0,1)$	$-\mathbb{E}_{\boldsymbol{z}\sim p_z}[D(G(\boldsymbol{z}))]$
GAN-NS	$\mathbb{E}_{\boldsymbol{x}\sim p_r}[-\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(1 - D(G(\boldsymbol{z})))]$	$\mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(D(G(\boldsymbol{z})))]$
GAN-R1	$\mathbb{E}_{\boldsymbol{x}\sim p_r}[-\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(1 - D(G(\boldsymbol{z})))] + \lambda\mathbb{E}_{\boldsymbol{x}\sim p_r}[\|(\nabla D)_{\boldsymbol{x}}\|^2]$	$\mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(D(G(\boldsymbol{z})))]$
GAN-0GP	$\mathbb{E}_{\boldsymbol{x}\sim p_r}[-\log(D(\boldsymbol{x}))] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(1 - D(G(\boldsymbol{z})))] + \lambda\mathbb{E}_{\boldsymbol{u}}[\|(\nabla D)_{\boldsymbol{u}}\|^2]$ where $\boldsymbol{u} = \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}; \boldsymbol{x} \sim p_x, \boldsymbol{y} \sim p_g, \alpha \sim \mathcal{U}(0,1)$	$\mathbb{E}_{\boldsymbol{z}\sim p_z}[-\log(D(G(\boldsymbol{z})))]$

TABLE I: Loss functions of GAN variants considered in this paper.



(a) Iteration 3000 (b) Iteration 3500 (c) Iteration 3600 (d) Iteration 20000

(e) Iteration 1000 (f) Iteration 2500 (g) Iteration 5000 (h) Adam. Iteration 1500

Fig. 1: (a) - (d) catastrophic forgetting in GAN-NS trained on the 8 Gaussian dataset. (e) - (g) GAN-R1 with $\lambda = 10$. GAN-0GP and WGAN-GP exhibit similar behaviors on this dataset. (h) GAN-NS trained with Adam. Viewing on computer is recommended.

the same direction, almost all of fake datapoints move in the same direction and do not spread out over the space. *Because of CF, the generator is unable to break out of mode collapse.*

Inside the green box (Fig. 1a), gradients at all datapoints have approximately the same direction. The loss $\mathcal{L}_G$ decreases (the score $D(\cdot)$ increases) monotonically along the direction of the green vector $\boldsymbol{u}$, a random vector that points away from the fake datapoints.[4] We have the following observation:

Observation 1. *In a large neighborhood around a real datapoint, $\mathcal{L}_G$ (and therefore, $D(\cdot)$) is directionally monotonic.*

A theoretical explanation to this phenomenon is given in Sec. IV-B. Because fake samples in Fig. 1a-1d are concentrated in a small region (i.e. mode collapse), D can easily separate them from distant real samples and does not learn useful features of the real data. We say that D *catastrophically forgets* real samples that are far away from the current fake samples. *Mode collapse and CF are interrelated, one problem makes the other more severe.*

In Fig. 1b, fake datapoints on the right of the red box have higher scores than real datapoints on the left, although in Fig. 1a, these real datapoints have higher scores than these fake datapoints. Going from Fig. 1a to 1d, we observe that the vectors' directions change as soon as fake datapoints move. The phenomenon suggests that *information about previous model distributions is not preserved in the discriminator*. As D^t tries to separate p_g^t from p_r, it assigns low scores to regions with fake samples and higher scores to other regions. Because D^t does not 'remember' $p_g^{<t}$, it could assign high scores to regions previously occupied by p_g^t, i.e. D^t could classify old fake samples as real. Fake samples at iteration 3000 (Fig. 1a)

[4]Graphically, we see that the angles between the green vector $\boldsymbol{u}$ and $\boldsymbol{v} = -\partial\mathcal{L}_D/\partial\boldsymbol{x}$ are less than 90° for all $\boldsymbol{x}$ in the box. Thus, the dot product $\boldsymbol{v} \cdot d\boldsymbol{u}$ is positive. The line integral in Eqn. 1 is positive for $\boldsymbol{x}_0, \boldsymbol{x}_1$ in the box that satisfy $\boldsymbol{x}_1 = \boldsymbol{x}_0 + k\boldsymbol{u}$, $k > 0$. $\mathcal{L}_G$ monotonically decreases along the direction of $\boldsymbol{u}$. We say that $\mathcal{L}_G$ is monotonic in direction $\boldsymbol{u}$.

are classified as real by D^{3500} (Fig. 1b). Similar behaviors are observed on MNIST (Fig. 11b). Because of forgetting, D could direct G to move to a region which G has visited before. That could cause G and D to fall in a learning loop and do not converge to an equilibrium. In Fig. 1a - 1d, the model distribution rotates around the circle indefinitely. *CF is a cause of non-convergence.*

2) Catastrophic forgetting on image datasets: We performed experiments on real world datasets to confirm the existence of CF in GANs. We visualize the landscape around a real datapoint $\boldsymbol{x}$ by plotting the output of the discriminator along a random line through $\boldsymbol{x}$. We choose a random unit vector $\hat{\boldsymbol{u}} \in \mathbb{R}^d, \|\hat{\boldsymbol{u}}\| = 1$ and plot the value of the function

$$f(k) = D(\boldsymbol{x} + k\hat{\boldsymbol{u}}) \tag{2}$$

for $k \in [-100, 100]$. We use the same $\hat{\boldsymbol{u}}$ for all images in Fig. 2. We choose to visualize $D(\cdot)$ instead of $\mathcal{L}_G$ because $\mathcal{L}_G$ explodes if $D(\cdot) \ll 1$. The quality of the image $\boldsymbol{x} + k\hat{\boldsymbol{u}}$ decreases as $|k|$ increases. A good discriminator D^* should assign lower scores to samples with lower quality. $D^*(\boldsymbol{x})$ should be higher than $D^*(\boldsymbol{x} + k\hat{\boldsymbol{u}}),\ k > 0$, i.e. $\boldsymbol{x}$ is a local maximum of D^*. If $\boldsymbol{x}$ is a local maximum of D^*, $f^*(k)$ must have a local maximum at $k = 0$ (the center of each subplot). The result reported below was observed in all 10 different runs of the experiment.

Fig. 2 demonstrates the problem on MNIST. The generator and discriminator are 3 hidden layer MLPs with 512 hidden neurons. SGD with constant learning rate $\alpha = 3 \times 10^{-4}$ was ued in training.

As shown in Fig. 2, the generated images keep changing from one shape to another, implying that the game does not converge to an equilibrium. In a large neighborhood around every real image, the discriminator's output is monotonic in the sampled direction. At iteration 100000, for every image, f is a decreasing function (Fig. 2f), while at iteration 200000, f is an increasing function (Fig. 2g). More conretely, let $\nabla_{\hat{\boldsymbol{u}}} D^t(\boldsymbol{x}_0)$ be the discriminator's directional derivative along direction $\hat{\boldsymbol{u}}$ at $\boldsymbol{x}_0$ at iteration t. Then Fig. 2f and 2g shows that $\nabla_{\hat{\boldsymbol{u}}} D^{100000}(\boldsymbol{x}_0)$ and $\nabla_{\hat{\boldsymbol{u}}} D^{200000}(\boldsymbol{x}_0)$ for some $\boldsymbol{x}_0$ near the real datapoint $\boldsymbol{x}$, have opposite directions. The knowledge of D^{200000} (what D^{200000} learned on $\{p_g^{200000}, p_r\}$) is not usable for $\{p_g^{100000}, p_r\}$.

We trained DCGAN [20] on CelebA [21] and CIFAR-10 [22] to study the effect of network architecture and dataset complexity on the level of forgetting. Network architecture and hyper parameters are given in Table II.

On CelebA, Fig. 9a - 9g show that CNN suffers less from CF than MLP. The discriminator in DCGAN-NS is not directional monotonic and it successfully makes many real datapoints its local maxima (see Sec. IV for more). The discriminator can effectively discriminate real images from neighboring noisy images. The generator moves fake datapoints toward these local maxima and produces recognizable faces.

On CIFAR-10 (Fig. 10a - 10g), the discriminator cannot discriminate real images from noisy images. The function $f(k)$ in Fig. 10b is almost an increasing function while in Fig. 10d it is almost a decreasing function. The training does not converge as fake images change significantly as the learning progresses.

Conclusion: GAN-NS trained on high dimensional datasets exhibits the same catastrophic forgetting behaviors as on toy datasets: (1) real datapoints are not local maxima of the discriminator or in more extreme cases, the discriminator is directionally monotonic in the neighborhoods of real datapoints; (2) the gradients w.r.t. datapoints in the neighborhood of a real datapoint change their directions significantly as fake datapoints move.

3) The causes of Catastrophic Forgetting: Based on the above experiments, we identified two reasons for CF:

1) *Information from previous tasks is not carried to/used for the current task.* SGD does not use information from previous model distributions, $p_g^{<t}$. At iteration t, SGD update for the discriminator is computed from samples from p_g^t and p_r only. Because information from $p_g^{<t}$ is not used in training, the discriminator forgets $p_g^{<t}$, i.e. it does not assign low score to samples from $p_g^{<t}$.
2) *The current task is significantly different from previous tasks so the knowledge of the current task cannot be used for previous tasks and vice versa.* As old knowledge is overwritten by new knowledge, optimizing the discriminator on the current task will degrade its performance on older tasks.

Methods for preventing CF is studied in Section V.

IV. The output landscape

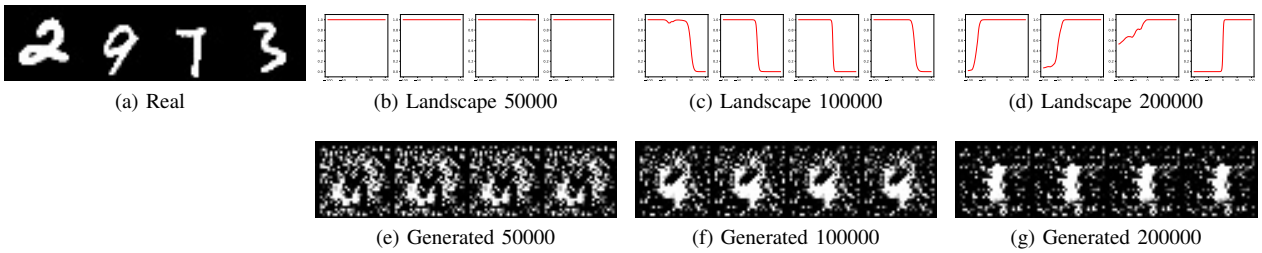
A. The evolution of the landscape

We apply the visualization technique in Section III-B2 to other variants of GAN. We reuse the network architecture and learning rate from the experiment in Fig. 2. We replace SGD with Adam with $\beta_1 = 0.5, \beta_2 = 0.99$. We run each experiment 10 times with different random seeds and report results that are consistent between different runs. The evolution of the landscape and generated samples of GAN-NS, GAN-0GP with $\lambda = 100$, GAN-R1 with $\lambda = 100$, and WGAN-GP with $\lambda = 10$ are shown in Fig. 3, 4, 5, and 6 respectively.

GAN-0GP, GAN-R1, and WGAN-GP have significantly better sample quality and diversity than GAN-NS. GAN-NS does not exhibit good convergence behavior: the digit in a image changes from one digit to another as the training progresses (Fig. 3).[5] GAN-0GP, GAN-R1, and WGAN-GP exhibit better convergence behaviors: for many images, the digits stay the same during training.

We observe that throughout the training process of GAN-0GP, GAN-R1, and WGAN-GP, for every real datapoint, the function $f(k)$ always has a local maximum at $k = 0$, implying that real datapoints are local maxima of the discriminator. This can also be seen in GAN-R1 trained on the 8 Gaussian dataset (Fig. 1e - 1g): the gradients w.r.t. datapoints in the neighborhood of a real datapoint point toward that real datapoint (GAN-0GP and

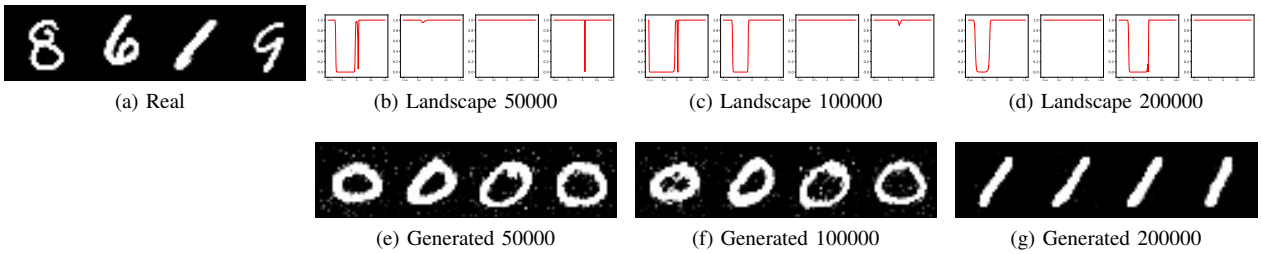
[5]Note that this does not contradict the statement in [11] that GAN-NS converge to an equilibrium. Many of the assumptions in that paper is not satisfied in practice, e.g. the learning rate is not decayed toward 0.



(a) Real (b) Landscape 50000 (c) Landscape 100000 (d) Landscape 200000

(e) Generated 50000 (f) Generated 100000 (g) Generated 200000

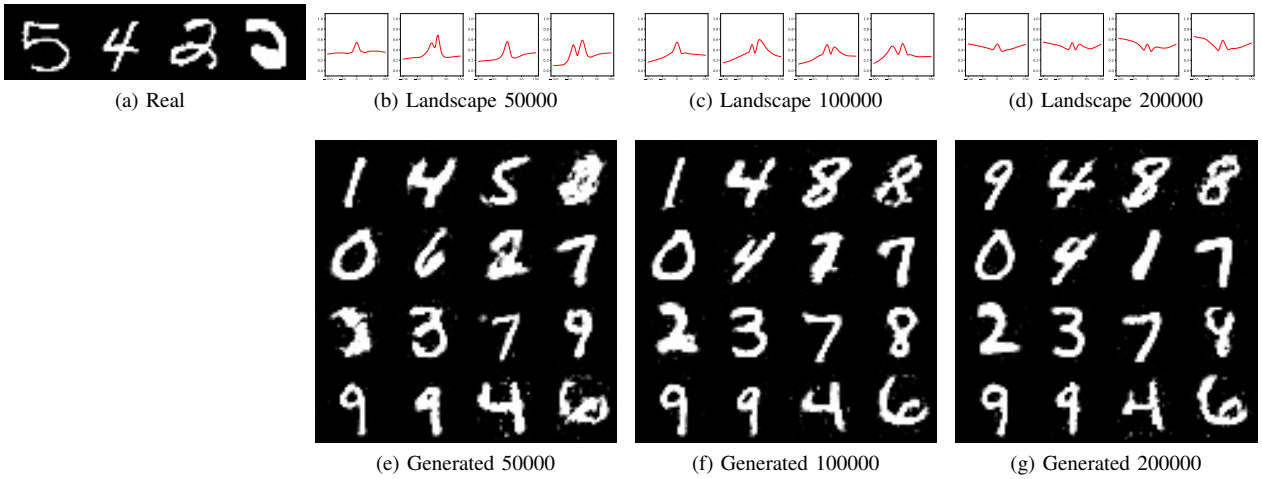
Fig. 2: Catastrophic forgetting problem in GAN-NS trained with SGD. (a) real datapoints from MNIST dataset. (b) - (d) the landscape around these real datapoints at different training iterations. In each subplot, the X-axis represent k, the Y-axis represent $D(\cdot)$. (e) - (g) generated data at different iterations. The same noise inputs were used for all iterations.



(a) Real (b) Landscape 50000 (c) Landscape 100000 (d) Landscape 200000

(e) Generated 50000 (f) Generated 100000 (g) Generated 200000

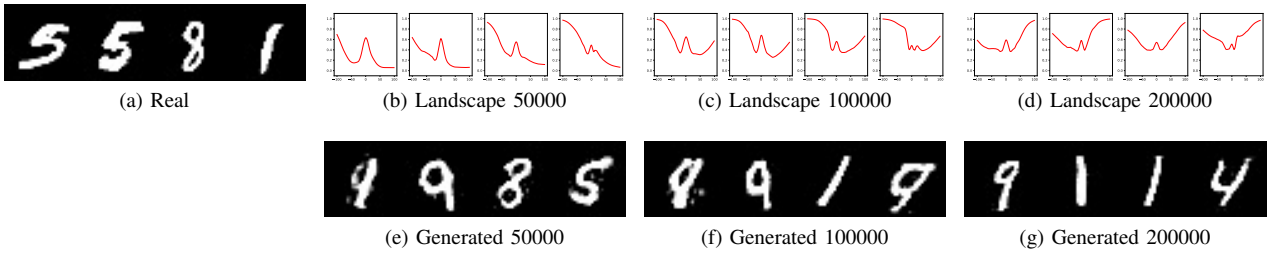
Fig. 3: Output landscape and generated samples from GAN-NS + Adam.



(a) Real (b) Landscape 50000 (c) Landscape 100000 (d) Landscape 200000

(e) Generated 50000 (f) Generated 100000 (g) Generated 200000

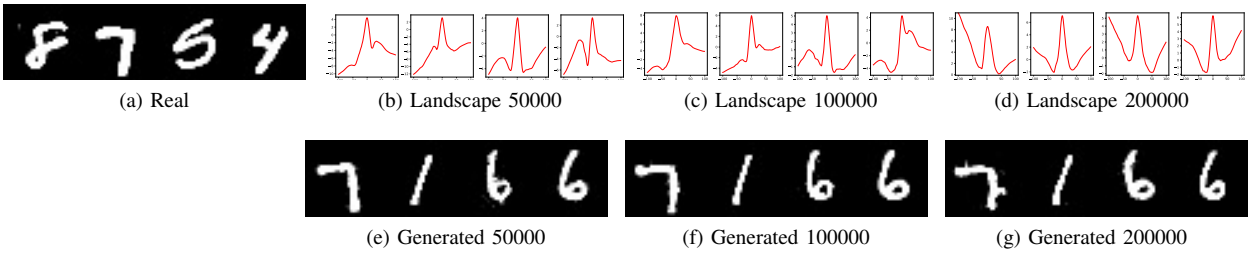
Fig. 4: Output landscape and generated samples from GAN-0GP with $\lambda = 100$.



(a) Real (b) Landscape 50000 (c) Landscape 100000 (d) Landscape 200000

(e) Generated 50000 (f) Generated 100000 (g) Generated 200000

Fig. 5: Output landscape and generated samples from GAN-R1, $\lambda = 100$.



(a) Real (b) Landscape 50000 (c) Landscape 100000 (d) Landscape 200000

(e) Generated 50000 (f) Generated 100000 (g) Generated 200000

Fig. 6: Output landscape and generated samples from WGAN-GP, $\lambda = 10$, 5 discriminator updates per 1 generator update.

WGANGP exhibit the same behaviors). If a fake datapoint is in the basin of attraction of a real datapoint and gradient updates are applied directly on the fake datapoint, it will be attracted toward the real datapoint. Different attractors (local maxima) at different regions of the data space attract different fake datapoints toward different directions, spreading fake datapoints over the space, effectively reducing mode collapse.

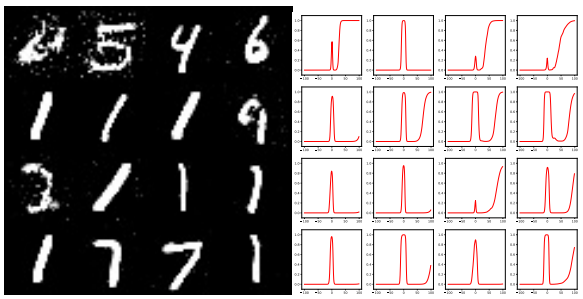
Fig. 7 shows that GAN-0GP with $\lambda = 10$ suffers from mild mode collapse.[6] The maxima in Fig. 7 are much sharper than those in Fig. 6. The discriminator overfits to the real training datapoints and forces the scores of near by datapoints to be close to 0. That creates many flat regions where the gradients of the discriminator w.r.t. datapoints in these regions are vanishingly small. A fake datapoint located in a flat region cannot move toward the real datapoint because the gradient is vanishingly small. Real datapoints in Fig. 7 have small basin of attraction and cannot effectively spread fake samples over the space. The diversity of generated samples is thus reduced, making mode collapse visible. In order to attract fake datapoints toward different directions, *local maxima should be wide*, i.e. they should have large basin of attraction.

The landscapes of GAN-NS in Fig. 2 and 3 contain many flat regions where the scores $D(\cdot)$ are very close to 1 or 0. The same problem is seen on the 8 Gaussian dataset (datapoints in the orange and blue boxes in Fig. 1a-1d have scores close to 1 and 0, respectively). However, unlike Fig. 7, the real datapoints in Fig. 1a - 1d, 2, and 3 are not local maxima. The discriminator in GAN-NS underfits the data.

CNN based discriminators do not create flat regions in the output landscape (Fig. 9b-9d and 10b-10d). However, when the dataset is more complicated, DCGAN-NS discriminator fails to make real datapoints local maxima and the training does not converge (Fig. 10a-10g). The discriminator underfits the data because it is not powerful enough to learn features that separate real and fake/noisy samples. More powerful discriminators based on ResNet [23] significantly improve the quality of GANs (e.g. [24]). We make the following observation:

Observation 2. *For a GAN to converge to a good local equilibrium, real datapoints should be wide local maxima of the discriminator.*

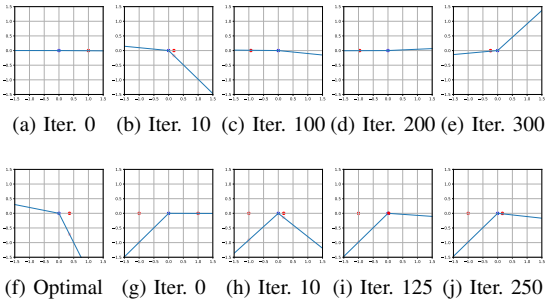
[6]This is consistent with the analysis by the authors of GAN-0GP. Thanh-Tung et al. claimed that larger λ leads to better generalization but may slow down the training.



(a) Generated 100000 (b) Landscape 100000

Fig. 7: Mode collapse without CF in GAN-0GP, $\lambda = 10$.

B. The effect of catastrophic forgetting on the landscape



(a) Iter. 0 (b) Iter. 10 (c) Iter. 100 (d) Iter. 200 (e) Iter. 300

(f) Optimal (g) Iter. 0 (h) Iter. 10 (i) Iter. 125 (j) Iter. 250

Fig. 8: High capacity Dirac GAN with $n = 2$. The blue line represents the discriminator's function. The real and fake datapoints are shown by the blue and red dots, respectively. (a) - (e): Dirac GAN trained on the current fake example only. (f): empirically optimal Dirac discriminator trained on the current fake example only. (g) - (j): Dirac GAN trained on two fake examples: old fake example on the left and current fake example on the right.

We investigate the effect of CF on Dirac GAN [9], a GAN that learns a 1 dimensional Dirac distribution located at the origin, $p_r = \delta_0$. In the original Dirac GAN, the discriminator is a linear function with 1 parameter, $D(x) = \psi x,\ \psi \in [-1, 1]$ and the model distribution is a Dirac distribution located at θ, $p_g = \delta_\theta$. θ is the generator's parameter. Initially, $\theta \neq 0$. At each iteration, the training dataset of Dirac GAN contains two training examples: a real training example $x_0 = 0$, and a fake

training example $y_0 = \theta$. Gradient updates are applied directly on the fake training example.

$$-\mathcal{L}_G^{dirac} = \mathcal{L}_D^{dirac} = -D(0) + D(x) \tag{3}$$

The unique equilibrium is $\psi = \theta = 0$. Mescheder et al. showed that the players in Dirac GAN do not converge to an equilibrium (see Fig. 1 in [9]). To make the game converge to the above equilibrium, the authors proposed R1 gradient penalty which pushes the gradient w.r.t. the real datapoint to $\mathbf{0}$ (Table I). A high dimensional GAN can be narrowed to a Dirac GAN by considering a pair of real and fake sample and the discriminator's output along the line connecting these samples (similar to the landscape in Fig. 2-6).

Because the discriminator in the original Dirac GAN is a linear function with a single parameter, the output of Dirac discriminator is always a monotonic function. We consider a generic discriminator which is a 1 hidden layer neural network: $\hat{D}(x) = \mathbf{\Psi}_1^\top \sigma(\mathbf{\Psi}_0 x)$ where $\mathbf{\Psi}_0, \mathbf{\Psi}_1 \in [-1, 1]^{n \times 1}$, and σ is a monotonically increasing activation function such as Leaky ReLU (Fig. 8). At equilibrium, $\theta = 0$ and $\hat{D}(x)$ is any function with a global maximum at $x = 0$. Although $\hat{D}$ can have global maxima (see Fig. 8h), optimizing $\hat{D}$ only on the current task makes $\hat{D}$ a monotonic function (Fig. 8f).

Proposition 1. *The optimal Dirac discriminator $\hat{D}^*(x)$ that minimizes $\mathcal{L}_D^{dirac}$ in Eqn. 3 is a monotonic function.*

Proof. Let $\hat{D}(x) = \mathbf{\Psi}_1^\top \sigma(\mathbf{\Psi}_0 x)$ where $\mathbf{\Psi}_0, \mathbf{\Psi}_1 \in [-1, 1]^{n \times 1}$ be the discriminator and σ be a non-decreasing activation function such as ReLU, Leaky ReLU, Sigmoid, or Tanh. Let $x_0 = 0$ be the real datapoint, $y_0 = \theta \neq 0$ be the fake datapoint. The empirically optimal discriminator D^* must maximize the difference $D^*(x_0) - D^*(y_0)$.

$$\begin{aligned}
\hat{D}(x_0) &= \mathbf{\Psi}_1^\top \sigma(\mathbf{\Psi}_0 \times 0) \\
&= \mathbf{\Psi}_1^\top \sigma(\mathbf{0}) \\
&= \sum_{i=1}^{n} \Psi_{1,i} \sigma(0) \\
\hat{D}(y_0) &= \mathbf{\Psi}_1^\top \sigma(\mathbf{\Psi}_0 \times y_0) \\
&= \sum_{i=1}^{n} \Psi_{1,i} \sigma(\Psi_{0,i} y_0) \\
\hat{D}(x_0) - \hat{D}(y_0) &= \sum_{i=1}^{n} \Psi_{1,i} \times (\sigma(0) - \sigma(\Psi_{0,i} y_0))
\end{aligned}$$

Because

$$\Psi_{0,i} y_0 \leq |y_0|$$

and σ is non-decreasing

$$\sigma(0) - \sigma(-|y_0|) \geq \sigma(0) - \sigma(\Psi_{0,i} y_0) \geq \sigma(0) - \sigma(|y_0|)$$

If σ is ReLU or Leaky ReLU or Tanh, then $\sigma(0) = 0$, $|\sigma(|y_0|)| \geq |\sigma(-|y_0|)|$, thus

$$|\sigma(0) - \sigma(|y_0|)| > |\sigma(0) - \sigma(-|y_0|)|$$

Architecture	DCGAN Pytorch example
Learning rate	2e-4
Batch size	64
Optimizer	Adam, $\beta_1 = 0.5, \beta_2 = 0.99$
No. filters at 1st layer	64

TABLE II: DCGAN model architecture & hyper parameters.

If σ is Sigmoid, then $\sigma(0) = 0.5$ and $|\sigma(0) - \sigma(|y_0|)| = |\sigma(0) - \sigma(-|y_0|)|$. For both cases, we have

$$|\sigma(0) - \sigma(\Psi_{0,i} y_0)| \leq |\sigma(0) - \sigma(|y_0|)| \tag{4}$$

Thus

$$\Psi_{1,i}(\sigma(0) - \sigma(\Psi_{0,i} y_0)) \leq 1 \times |\sigma(0) - \sigma(|y_0|)| \tag{5}$$

The equality for both Eqn. 1 and 2 is achieved for all cases when $\Psi_{1,i} = -1$ and $\sigma(\Psi_{0,i} y_0) = \sigma(|y_0|) \Rightarrow \Psi_{0,i} y_0 = |y_0| \Rightarrow \Psi_{0,i} = \text{sign}(y_0)$. The optimal discriminator's parameters are $\mathbf{\Psi}_0^* = \text{sign}(y_0) \times \mathbf{1}, \mathbf{\Psi}_1^* = -\mathbf{1}$.

$$D(x) = -\mathbf{1}^\top \sigma(x \times \text{sign}(y_0) \times \mathbf{1})$$

Without loss of generality, assume $\text{sign}(y_0) = 1$.

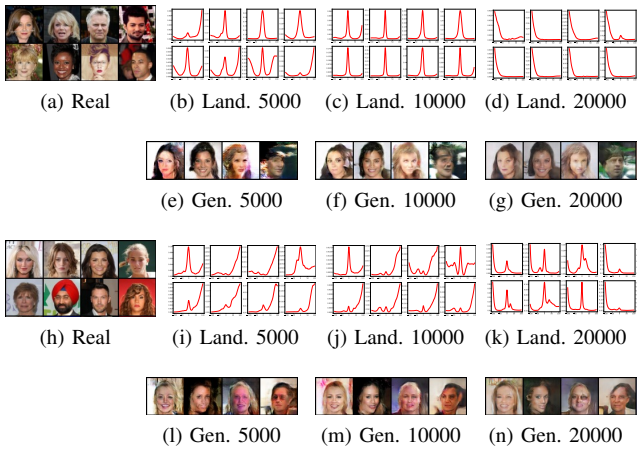
$$D(x) = -\mathbf{1}^\top \sigma(x \times \mathbf{1}) = -n\sigma(x)$$

Because σ is monotonic, $D(x)$ is monotonic. □

Optimizing the performance of $\hat{D}$ pushes it toward $\hat{D}^*$, making $\hat{D}$ monotonic (Fig. 8a - 8e). This explains the directional monotonicity of discriminators in Fig. 1a-1d, 2.

Although the discriminator in Fig. 8f minimizes the score of the current fake datapoint, it assigns high scores to (old) fake datapoints on the left of the real datapoint, i.e. it forgets these datapoints. If the discriminator is fixed, then minimizing $\mathcal{L}_G^{dirac}$ corresponds to moving θ to $-\infty$. *Dirac GAN with a monotonic discriminator does not converge.* When the generator and discriminator are trained with alternating SGD, the two players oscillate around the equilibrium (Fig. 8a - 8e).

The problem can be alleviated if one old fake datapoint is added to the training dataset. Fig. 8g - 8j shows that when old fake example is added, Dirac GAN has better convergence behavior (the small fluctuation is due to the large constant learning rate of 0.1). The discriminator at iteration 10 has a global maximum at the origin. If the discriminator is fixed, then θ will converge to 0. The experiment suggests that information about previous model distributions helps GANs converge. [25] used a buffer of recent old fake samples to refine reasonably good fake samples. Recent old fake samples reduce the oscillation around the equilibrium, helping GANs to converge faster and produce sharper images. However, because the number of samples needed to capture the statistics of a distribution grows exponentially with it dimensionality, storing old fake datapoints is not efficient for high dimensional data. In the next section, we study more efficient methods for preserving information about old distributions.



(a) Real (b) Land. 5000 (c) Land. 10000 (d) Land. 20000

(e) Gen. 5000 (f) Gen. 10000 (g) Gen. 20000

(h) Real (i) Land. 5000 (j) Land. 10000 (k) Land. 20000

(l) Gen. 5000 (m) Gen. 10000 (n) Gen. 20000

Fig. 9: Result on CelebA. (a) - (g) DCGAN-NS. (h) - (n) DCGAN-0GP

(a) Real (b) Land. 5000 (c) Land. 10000 (d) Land. 20000

(e) Gen. 5000 (f) Gen. 10000 (g) Gen. 20000

(h) Real (i) Land. 5000 (j) Land. 10000 (k) Land. 20000

(l) Gen. 5000 (m) Gen. 10000 (n) Gen. 20000

Fig. 10: Result on CIFAR-10. (a) - (g) DCGAN-NS. (h) - (n) DCGAN-imba, $\gamma = 10$.

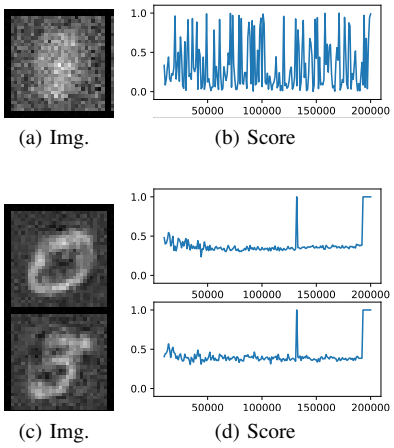
V. Preventing Catastrophic Forgetting

Based on the reasons identified in Section III-B, we propose the following ways to address CF problem:

1) *Preserve and use information from previous tasks in the current task.*
2) *Introduce prior knowledge to the game in a way such that old knowledge is useful for the new task and is not erased by the new task.*

	mean/std
DCGAN	2.054/0.913
DCGAN-imba, $\gamma = 10$	3.381/0.078
DCGAN-0GP, $\lambda = 100$	2.705/0.901
DCGAN-0GP-imba, $\lambda = 100, \gamma = 10$	3.038/0.342

TABLE III: Inception scores of models at iteration 50k. The result is averaged over 10 different runs.



(a) Img. (b) Score

(c) Img. (d) Score

Fig. 11: Score of fixed fake images during training from iteration 10000 to 200000. The same MLP in Fig. 2 was trained with SGD with learning rate $3e-4$. (a) - (b) GAN-NS. (c) - (d) GAN-0GP with $\lambda = 100$. GAN-NS assigns random scores to the same fake image, implying that it does not remember information about this fake sample. GAN-0GP is much more stable and consistently assigns scores lower than 0.5 to old fake samples.

A. Preserving and using old information

Optimizers with momentum. The update rule of SGD with momentum

$$\begin{aligned} \boldsymbol{g}^t &= \gamma \boldsymbol{g}^{t-1} + \eta \nabla_\theta^t \\ \boldsymbol{\theta}^{t+1} &= \boldsymbol{\theta}^t - \boldsymbol{g}^t \end{aligned}$$

The momentum term $\gamma \boldsymbol{g}^{t-1}$ is a simple form of memory that carries gradient information from previous training iterations to the current iteration. When the discriminator/generator is updated with $\boldsymbol{g}^t$, the performance of the network on previous tasks is also improved. The effectiveness of momentum in preventing CF is demonstrated in Fig. 1h: the discriminator's gradient pattern is more stable and similar to those of GAN-0GP and GAN-R1.

Continual learning algorithms such as EWC [3] and online EWC [26] prevent important knowledge of previous tasks from being overwritten by the new task. At the end of a task $\mathcal{T}^t$, online EWC computes the importance $\hat{\omega}_i^t$ of each parameter θ_i^t to the task and adds a regularization term to the loss function of task $\mathcal{T}^{t+1}$:

$$\begin{aligned} \omega_i^t &= \alpha \hat{\omega}_i^t + (1-\alpha)\omega_i^{t-1} \\ \mathcal{L}_{EWC}^{t+1} &= \mathcal{L}^{t+1} + \lambda \sum_i \omega_i^t (\theta_i - \theta_i^t)^2 \end{aligned}$$

where θ_i^t is the value of θ_i at the end of task $\mathcal{T}^t$, α balances the importance of the current task and previous tasks, ω_i^t accumulates the importance of θ_i throughout the training process. Because consecutive model distributions are similar, we consider a chunk of τ distributions as a task to the discriminator. The importance ω_i is computed every τ GAN

training iteration. The regularizer prevents important weights from deviating too far from the values that are optimal to previous tasks while allowing less important weights to change more freely. It helps the discriminator preserves important information about old distributions. Liang et al. independently proposed a similar way of adapting continual learning methods to GANs. Experiments in the paper showed that continual learning methods improve the quality of GANs.

B. Introducing prior knowledge to the game

In Dirac GAN, if the discriminator has a local maximum at the real datapoint then it can always classify the real and the fake datapoint correctly, regardless of location of the fake datapoint. Because separating different fake distributions from the target distribution requires the same knowledge, that knowledge will not be erased from the discriminator. We want to introduce to the game the knowledge that real datapoints should be local maxima. R1 and 0GP are two ways to implement that.

R1 regularizer (the third row in Table I) forces the gradients w.r.t. a real datapoint to be $\mathbf{0}$, making it a local extremum of the discriminator. As the discriminator maximizes the score of real datapoints, real datapoints become local maxima of the discriminator. Fig. 1e - 1g shows that real datapoints are always local maxima and the gradient pattern of the discriminator stay unchanged as p_g moves toward p_r. Fig. 5 demonstrates the same effect of R1 on MNIST. Note that noisy images that are far away from the real images (e.g. $\boldsymbol{x} + k\hat{\boldsymbol{u}}$ for $k < -50$) have higher scores than real images. This is because no regularizer is applied to these noisy images.

0GP regularizer (the forth row in Table I) pushes gradients w.r.t. datapoints on the line connecting a real datapoint $\boldsymbol{x}$ and a fake datapoint $\boldsymbol{y}$ toward $\mathbf{0}$. 0GP forces the score to increase gradually as we move from $\boldsymbol{y}$ to $\boldsymbol{x}$. During training, $\boldsymbol{x}$ is paired with different $\boldsymbol{y}_i$. Thus, the score $D(\boldsymbol{x})$ is greater than the scores of fake datapoints in a wider neighborhood. That fixes the problem of R1 and creates wider local maxima (Fig. 4, 9). Thanh-Tung et al. [10] showed that GAN-0GP generalizes better than GAN-R1. Although generalization is beyond the scope of this paper, we believe that the sharpness of the discriminator's landscape is related to its generalization capability. Prior works on generalization of neural networks [27] showed flat (wide) minima of the loss surface generalize better than sharp minima. Creating discriminators with wide local maxima is a good way to improve GANs' generalizability.

WGAN-GP (the first row in Table I) uses 1-centered gradient penalty (1GP) which pushes gradients w.r.t. datapoints on the line connecting a real datapoint $\boldsymbol{x}$ and a fake datapoint $\boldsymbol{y}$ toward $\mathbf{1}$, forcing the score to increase gradually from $\boldsymbol{y}$ to $\boldsymbol{x}$. Fig. 6 shows that real datapoints are local maxima of the discriminator. Wu et al. [28] showed that WGAN-0GP performs slightly better than WGAN-1GP. Our hypothesis is that 0GP creates wider maxima than 1GP as it make the score on the line from $\boldsymbol{y}$ to $\boldsymbol{x}$ to change more slowly.

Imbalanced weights for real and fake samples. To prevent the discriminator from forgetting distant real datapoints, we propose to increase the weight of the loss for real datapoints:

$$\mathcal{L}_D = \gamma \mathcal{L}_{real} + \mathcal{L}_{fake} \tag{6}$$

where $\gamma > 1$ is an empirically chosen hyper parameter, $\mathcal{L}_{real}$, $\mathcal{L}_{fake}$ are the losses for real and fake samples, respectively. When $\gamma > 1$, the discriminator is penalized more if it assigns a low score to a real datapoint. The situation where real datapoints are local minima like in Fig. 10b or have low scores like in the blue boxes in Fig. 1a - 1b will less likely to happen. Fig. 10k shows that the new loss successfully helps the discriminator to make more real datapoints local maxima and thus improve fake samples' quality. Table III shows the effectiveness of imbalanced loss on CIFAR-10 dataset: it significantly improves Inception Score [29] and reduces the score's variance. The imbalanced loss is orthogonal to gradient penalties and can be used to improve gradient penalties (the last two rows in Table III).

VI. Conclusion

Catastrophic forgetting is a important problem in GANs. It is directly related to mode collapse and non-convergence. Addressing catastrophic forgetting leads to better convergence and less mode collapse. Methods such as imbalanced loss, zero centered gradient penalties, optimizers with momentum, and continual learning are effective at preventing catastrophic forgetting in GANs. 0GP helps GANs to converge to good local equilibria where real datapoints are wide local maxima of the discriminator. The gradient penalty is a promising method for improving generalizability of GANs.

References

[1] 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 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[3] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. ISSN 0027-8424.

[4] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press, 1989.

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

[6] Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135, 1999. ISSN 1364-6613.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223. PMLR, 06–11 Aug 2017.

[8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.

[9] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3478–3487, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[10] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. Improving generalization and stability of generative adversarial networks. In *International Conference on Learning Representations*, 2019.

[11] Vaishnavh Nagarajan and J. Zico Kolter. Gradient descent gan optimization is locally stable. In *Advances in Neural Information Processing Systems 30*, pages 5585–5595. Curran Associates, Inc., 2017.

[12] 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 30*, pages 6626–6637. Curran Associates, Inc., 2017.

[13] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. In *Advances in Neural Information Processing Systems 30*, pages 1825–1835. Curran Associates, Inc., 2017.

[14] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *CoRR*, abs/1705.08395, 2017.

[15] Kevin J Liang, Chunyuan Li, Guoyin Wang, and Lawrence Carin. Generative Adversarial Network Training is a Continual Learning Problem. *arXiv e-prints*, art. arXiv:1811.11083, Nov 2018.

[16] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 224–232. PMLR, 06–11 Aug 2017.

[17] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018.

[18] Pengchuan Zhang, Qiang Liu, Dengyong Zhou, Tao Xu, and Xiaodong He. On the discrimination-generalization tradeoff in GANs. In *International Conference on Learning Representations*, 2018.

[19] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *International Conference on Learning Representations*, 2018.

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[22] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

[23] 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.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[25] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[26] Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4528–4537. PMLR, 10–15 Jul 2018.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[28] Jiqing Wu, Zhiwu Huang, Janine Thoma, Dinesh Acharya, and Luc Van Gool. Wasserstein divergence for gans. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 653–668, 2018.

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*, pages 2234–2242. Curran Associates, Inc., 2016.