On-Line Adaptative Curriculum Learning for GANs

Thang Doan,[1,8] **João Monteiro,**[2] **Isabela Albuquerque,**[2] **Bogdan Mazoure,**[3]
Audrey Durand,[4,5] **Joelle Pineau,**[4,5,6] **R Devon Hjelm**[5,7]
[1]Desautels Faculty of Management, McGill University
[2]INRS-EMT, Université du Québec
[3]Department of Mathematics & Statistics, McGill University
[4]School of Computer Science, McGill University
[5]Mila – Quebec Artificial Intelligence Institute
[6]Facebook AI Research,[7]Microsoft Research Montreal

Abstract

Generative Adversarial Networks (GANs) can successfully approximate a probability distribution and produce realistic samples. However, open questions such as sufficient convergence conditions and mode collapse still persist. In this paper, we build on existing work in the area by proposing a novel framework for training the generator against an ensemble of discriminator networks, which can be seen as a one-student/multiple-teachers setting. We formalize this problem within the full-information adversarial bandit framework, where we evaluate the capability of an algorithm to select mixtures of discriminators for providing the generator with feedback during learning. To this end, we propose a reward function which reflects the progress made by the generator and dynamically update the mixture weights allocated to each discriminator. We also draw connections between our algorithm and stochastic optimization methods and then show that existing approaches using multiple discriminators in literature can be recovered from our framework. We argue that less expressive discriminators are smoother and have a general coarse grained view of the modes map, which enforces the generator to cover a wide portion of the data distribution support. On the other hand, highly expressive discriminators ensure samples quality. Finally, experimental results show that our approach improves samples quality and diversity over existing baselines by effectively learning a curriculum. These results also support the claim that weaker discriminators have higher entropy improving modes coverage.

1 Introduction

Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) have reshaped the state of machine learning in tasks that involve generating data. A GAN is an unsupervised method that consists of two neural networks, a generator and a discriminator, with opposing (or *adversarial*) objectives. The typical goal of the generator is to transform noise (e.g., drawn from a normal distribution) into samples whose statistical and structural characteristics match well those of an empirical target dataset (such as a collection of images). The discriminator, which acts as an *adversary* to the generator, needs to discriminate between (or *classify*) samples as coming from the real data or the generator.

While GANs can achieve impressive qualitative performance (most notably with image data, e.g., see (Roth et al. 2017; Miyato et al. 2018; Karras et al. 2017)), the most successful methods depart from the original formulation to address various instabilities and other optimization difficulties (Arjovsky and Bottou 2017; Arjovsky, Chintala, and Bottou 2017). One such difficulty in training GANs occurs when the generator produces samples only from a small subset of the target distribution, a phenomenon known as *missing modes* (a.k.a., *mode-dropping*, e.g. see (Che et al. 2016)). Numerous works try to address the problem by modifying the original objective, such as unrolling (Metz et al. 2016), aggregating samples (Lin et al. 2017), stacked architectures (Huang et al. 2016; Karras et al. 2017), mutual information / entopy maximization (Belghazi et al. 2018), multiple discriminators (Neyshabur, Bhojanapalli, and Chakrabarti 2017; Juefei-Xu, Boddeti, and Savvides 2017), or multiple generators (Tolstikhin et al. 2017; Hoang et al. 2017; Kwak and Zhang 2016).

In our work, we follow the intuition that missing modes in GANs are due in part to mode-specific vanishing gradients. As a simple illustrative example which we explore in detail in our experiments below (Fig. 1), consider a discriminator that is well representing the target distribution and a generator that is only generating a subset of the modes in the data. If any of the missing modes are *disjoint* from those represented in the generator (i.e., are composed of sets of features with low intersection), there is no way for the generator to receive gradient signal on missing modes from the discriminator. However, if the discriminator only represents the data approximately (in the sense that it also cannot fully distinguish between these modes), it may be possible to recover the missing mode gradient signal. If this can be achieved by using

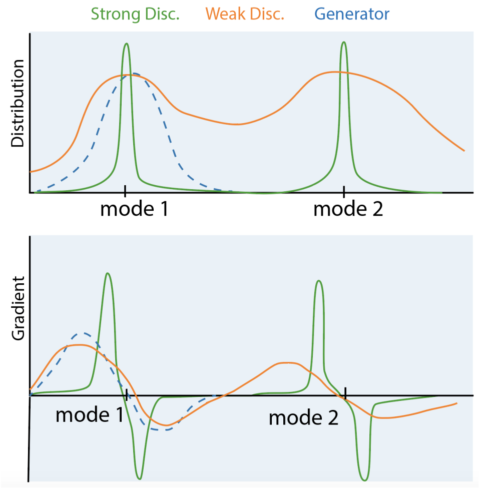

Figure 1: Recovering dropped modes via multiple discriminators. The weak discriminator provides feedback, allowing the generator to recover forgotten modes. The strong discriminator experiences vanishing gradient and cannot help the generator to recover modes.

a low capacity[1] discriminator, it is ultimately undesirable given that the end goal is to generate samples that resemble well the target dataset. From now on, we will refer to such low capacity discriminators as *weak* and to high capacity discriminators as *strong*. In order to ensure both high quality and mode coverage, we consider multiple discriminators (as in (Durugkar, Gemp, and Mahadevan 2016)) with different strengths to train the generator. We propose to train the generator using a curriculum based on an on-line multi-armed bandit algorithm (Matiisen et al. 2017; Graves et al. 2017), dynamically changing the weight/resources allocated to each discriminator, which we show is crucial for achieving good results. Our primary contributions are:

1. We provide important insights into the missing mode problem as demonstrated by the gradient signal available to the generator from the discriminator.
2. As a potential solution to the missing modes problem, we introduce a new framework based on adversarial bandits (Littlestone and Warmuth 1994; Auer et al. 1995; Freund and Schapire 1997) resource allocation, where the generator gets its training signal from a set of teacher networks with increasing capacity.
3. We show that the proposed approach leads to a curriculum learning characterized by successive phases of the generator prioritizing different discriminators.

The remainder of this paper is organized as follows. Previous literature relevant to this work is briefly reviewed on Section 2. The proposed approach is formally introduced in Section 3, and an empirical analysis is reported in Section 4. Conclusions and future directions are finally presented in Section 5.

2 Related Work

Mode coverage and data / model augmentation The intuition that missing modes are due to vanishing gradients resonates with some successful approaches on stabilizing and improving GAN training through data and model augmentation. Instance noise (Arjovsky and Bottou 2017) has been shown to improve stability (see also (Roth et al. 2017)), which can be understood as smoothing the data modes in the pixel space. Progressively reducing the downsampling through training (either by copying parameters or feeding low resolution samples into a larger generator) have also been considered previously (Huang et al. 2016; Karras et al. 2017) as solutions to increase mode overlap.

This is akin to a hand-crafted curriculum, progressively increasing the difficulty of the problem at a-priori chosen points in the complete training procedure.

Multiple discriminators and generators Several works have also incorporated multiple generators or discriminators in order to improve learning. Multiple-generator methods (Tolstikhin et al. 2017; Hoang et al. 2017; Kwak and Zhang 2016) typically work by encouraging the generators to divide the task of generating by modes in the target dataset (without additional supervision). Using multiple discriminators (Neyshabur, Bhojanapalli, and Chakrabarti 2017; Juefei-Xu, Boddeti, and Savvides 2017), on the other hand, is known to provide a better learning signal for the generator if said discriminators compositionally represent well the target datasets. Closest to our work, (Durugkar, Gemp, and Mahadevan 2016) consider discriminators of different complexity to provide varied signal. We will show that wisely designing the reward allows to track the progress made by the generator and encourages a curriculum learning.

Multi-armed bandit as a curriculum learning method for GANs Curriculum learning (Bengio et al. 2009) phrases a given machine learning problem as a set of tasks of increasing difficulty. GANs can also be said to share aspects with curriculum learning: the discriminator defines an objective of progressive difficulty,

thus allowing the generator to gradually learn to more faithfully mimic the target distribution. However, there is no explicit mechanism to encourage a sensible curriculum for either model. For example, if the discriminator learns to represent disjoint modes faster than the generator learns to cover them, this can lead to the generator missing modes with no gradient signal to recover.

In this paper, we propose an algorithm which gives rise to a curriculum in a direct manner. Our approach borrows from curriculum learning in multi-armed bandit setting (Matiisen et al. 2017; Graves et al. 2017), where learning is typically done by measuring the change in a performance criterion of a given agent (i.e. a loss function, score or gradient norm can be used) that appears to affect the form of the optimal policy. In our method, given a set of discriminators, the goal is to weight the feedback received by the generator proportionally to the information contained in the gradients from each discriminator.

[1] Throughout the paper, we refer to *capacity* as the architecture size of a given neural network in terms of number of parameters.

3 Adaptative Curriculum GAN

Here we formulate the problem and approach for training a single generator on a target dataset using a curriculum over multiple discriminators, which we call *Adaptative Curriculum GAN* (acGAN). First, define a generator function, $G : \mathcal{Z} \mapsto \mathcal{X}$, which maps noise from a domain $\mathcal{Z}$ to the domain of a target dataset, $\mathcal{X}$ (such as the space of images). Let $p(x)$ denote the target density [2], and let $p(z)$ denote the prior density defined on Z used to draw noise samples for input into the generator. We wish to train this generator function using N discriminators, $\mathcal{D} = \{D_i : \mathcal{X} \mapsto \mathbb{R}\}_{i=1}^N$, such that on each episode t, we select the mixture of discriminators that provides the best learning signal.

3.1 Mixing discriminators

This mixture-of-experts problem, where each discriminator plays the role of a teacher, can be tackled under the full-information adversarial bandit setting (Littlestone and Warmuth 1994; Freund and Schapire 1997; Auer et al. 1995).

On each episode t, a bandit player associates normalized weights $\Pi(t) = \{\pi_i(t)\}_{i=1}^N$ with discriminators $\{D_i\}_{i=1}^N$. The generator is then trained based on the mixture described by $\Pi(t)$, and a reward $\mathcal{R}_i(t)$ is observed for each discriminator D_i, characterizing the generator's improvement with respect to D_i. Let $\mathcal{R}(t) = \sum_{i=1}^N \pi_i \mathcal{R}_i(t)$ denote the total observed reward at time t. The goal of the player is to learn the optimal policy $\Pi^\star(t) := \operatorname{argmax}_{\Pi \in \Delta(N-1)} \mathbb{E}_{\Pi(t),p(z)}[\mathcal{R}(t)]$ that maximizes the expected total reward[3].

The Hedge algorithm (Freund and Schapire 1997), also known as Boltzmann or Gibbs distribution, addresses this full-information game by maintaining probabilities

$$\pi_i(t) = \frac{\exp \lambda Q_i(t)}{\sum_{j=1}^{N} \exp \lambda Q_j(t)}, \quad \lambda \geq 0, \tag{1}$$

for each discriminator D_i, where $Q_i(t)$ estimates the gain of D_i at episode t. In this case, λ is a parameter of the distribution: $\lambda = 0$ corresponds to a uniform distribution over all models. We found experimentally that using a moving average on previous rewards (which also featured in (Matiisen et al. 2017)) stabilizes the training:

$$Q_i(t) = \alpha \mathcal{R}_i(t) + (1-\alpha) Q_i(t-1), \tag{2}$$

where $\alpha \in (0, 1)$ is the smoothing parameter.

To demonstrate how this can be used to train GANs, consider the usual value function (Goodfellow et al. 2014):

$$V(D,G) = \mathbb{E}_{p(x)}[\log(D(x))] + \mathbb{E}_{p(z)}[\log(1 - D(G(z)))]. \tag{3}$$

On each episode t, given the mixture of discriminators $\Pi(t)$, each discriminator is trained by taking a gradient step to increase the expected value function

$$\mathbb{E}_{\Pi(t)}[V(D_i, G)] = \sum_j \pi_j(t) V(D_j, G), \tag{4}$$

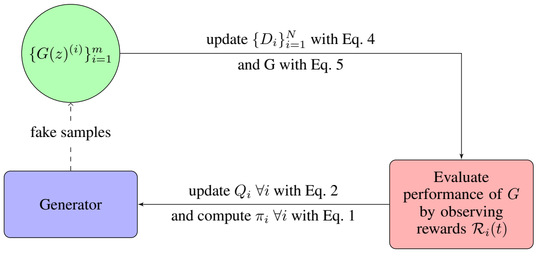
$\{G(z)^{(i)}\}_{i=1}^m$
update $\{D_i\}_{i=1}^N$ with Eq. 4
and G with Eq. 5
fake samples
Generator
update Q_i $\forall i$ with Eq. 2
and compute π_i $\forall i$ with Eq. 1
Evaluate performance of G by observing rewards $\mathcal{R}_i(t)$

Figure 2: Proposed procedure for training the generator

and the generator is trained by taking a gradient step to increase

$$\mathbb{E}_{\Pi(t)}[\mathbb{E}_{p(z)}[\log(D(G(z)))]]. \tag{5}$$

The latter corresponds to the non-saturated version of Eq. 4 for the generator. The intuition is that training the generator with all the discriminators *simultaneously* (as a mixture) should force the generator to fool all discriminators at the same time (Durugkar, Gemp, and Mahadevan 2016). Since each discriminator has an increasing level view of the modes distribution, they should have a complementary role. While the weaker discriminator focuses on modes coverage, the stronger discriminator ensures samples quality (showed in Section 4.1). This should result into a better overall coverage of the modes in the input distribution.

Algorithm 1 describes our proposed acGAN procedure. We denote and parameterize this algorithm as acGAN($\lambda, \alpha, \mathcal{R}_r$) where $\lambda \geq 0, \alpha \in (0, 1)$.

Algorithm 1 Generic acGAN algorithm

```
Given: $N$: number of discriminators, $T_{max}$: time steps, $T_{warmup}$: warmup time, $\alpha$: moving average coefficient, $\lambda$: Boltzmann constant
$Q_i(0) \leftarrow 0, \forall i = 1, \ldots, N$
for $t = 1, \ldots, T_{max}$ do
    Update all discriminators $\{D_i\}_{i=1}^N$ using Eq. 4
    Update the generator $G$ using Eq. 5
    if $t \geq T_{warmup}$ then
        Evaluate the performance of $G$ and observe a reward $\mathcal{R}_i(t)$ for each discriminator $i$
        Update all values $\{Q_i(t)\}_{i=1}^N$ according to Eq. 2
        $\pi_i(t) \leftarrow \exp^{\lambda Q_i(t)} / \sum_{j=1}^{N} \exp^{\lambda Q_j(t)} \quad \forall i = 1 \ldots N$
    end if
end for
```

Remark 1. *At the beginning of the training, we define a warm-up period T_{warmup}, prior to which we train D_i and G with a uniform probability, i.e $\pi_i = \frac{1}{N}, \forall i = 1, \ldots, N$. In other words, we consider $\lambda = 0, \forall t \leq T_{warmup}$. This guarantees that each discriminator is updated a minimum number of times (or provides feedback a minimum number of times to the generator) and prevents one D_j from dominating the others (i.e, $\pi_j \gg \pi_i, \forall i \neq j$) at the beginning of the training. Without this safeguard, the remaining weights $\pi_i, i \neq j$ would*

[2] Here, we assume for the sake of notation that the target data admits a density.

[3] $\Delta(N-1)$ denotes the standard simplex on $\mathbb{R}^N$.

hardly recover a significant probability and the generator may never get informative gradient from the corresponding discriminator. Note that warm-ups are not uncommon either in bandits algorithm, e.g. for adding robustness to the tails of reward distributions (Baransi, Maillard, and Mannor 2014).

3.2 Reward shaping

In order to provide meaningful feedback for learning efficient mixtures of discriminators, we consider different reward functions to generate $\mathcal{R}_i(t)$. We argue that progress (i.e., the learning slope (Matiisen et al. 2017; Graves et al. 2017)) of the generator is a more sensible way to evaluate our policy. Let $\theta(t)$ be the generator parameters at episode t. We define the two following quantities for measuring generator progress:

$$\mathcal{R}_i^{\mathcal{S}}(t) = \mathbb{E}_{p(z)}[D_i(G(z;\theta(t))) - D_i(G(z;\theta(t-1)))], \tag{6}$$

$$\mathcal{R}_i^{\mathcal{V}}(t) = \mathbb{E}_{p(z)}[V(D_i, G(z;\theta(t))) - V(D_i, G(z;\theta(t-1)))]. \tag{7}$$

The former measures the progress of the generator with respect to the discriminator i score $D_i(\cdot)$, while the latter assess the change in the loss function (Eq. 3). Since the change in the quality sample (Eq .6) led to better performance than the change in the loss function (Eq .7), all our experiments (see Section. 4) use Eq .6.

3.3 Connection to existing methods

Interestingly, some existing methods in the GAN literature can be seen as a specific case of acGAN:

GMAN: The original GMAN (Durugkar, Gemp, and Mahadevan 2016) algorithm can be recovered by setting $\alpha = 1$ and taking the loss function to be the reward $\mathcal{R}_i(t) = V(D_i, G)$. Note how the authors of GMAN call their algorithm GMAN-λ, where λ is also the Boltzmann coefficient.

Uniform: The uniform case is defined by assigning a fixed uniform probability for each discriminator D_i:

$$\pi_i(t) = \frac{1}{N}, \quad \forall t \in \mathbb{N}.$$

This corresponds to Eq. 1 with $\lambda = 0$.

To support the results of our theoretical work, we conducted a set of experiments which we describe below.

4 Experiments

In this section, we first give an understanding of how each discriminator provides informative feedback to the generator. We then compare our proposed approach (acGAN) against existing methods from the literature.

4.1 Retaining mode information through weaker capacity discriminators and smoothness

We begin by analyzing the gradient norm of the discriminator networks and we show that weak capacity discriminators are *smoother* than strong discriminators. This property corresponds to a "coarse-grained" representation of the distribution, which allows the generator to recover missing modes. We further show we can increase the smoothness of a weak discriminator by corrupting its inputs with white noise. This results in an increase of the discriminator's entropy (see Supplementary Material for more details) and hence smoother gradient signal.

Weak Discriminators: a way to retain modes We now highlight the role of weaker capacity discriminators. To this extent, we performed the following experiments on the 8 Gaussian synthetic dataset:

- We pretrained the generator (with 3 dense layers of 400 units with ReLU activation layers except for the last layer) with one discriminator on only 2 of the original 8 modes.
- We trained a (vanilla) GAN on all 8 Gaussian components, initializing with the 2-mode generator above. The discriminator had 3 dense layers of 400 units (ReLU hidden activation layers).
- We trained acGAN with the generator initialized with the 2-mode generator (as with vanilla GAN). We considered 3 discriminators, with 1, 2 and 3 dense layers respectively (same activation scheme as previously applies here).

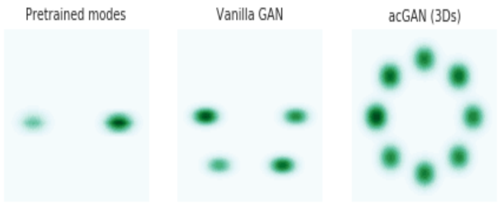

Figure 3: Modes used for pretraining the generator (left) and modes recovered by Vanilla GAN (middle) and acGAN (right). The more modes the better.

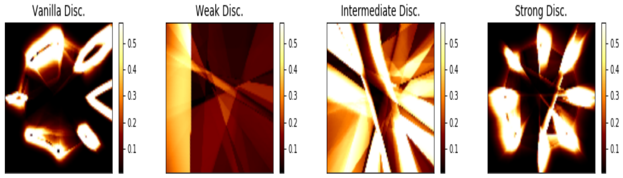

Figure 4: Gradient norm of each discriminator with respect to the input. We clipped the magnitude with respect to the weaker discriminator range. Since weaker discriminators are smoother by construction, they help the generator to recover missing modes. On the other hand, vanilla GAN can hardly recover modes due to its vanishing gradient.

Results (Fig. 3) show the Vanilla GAN could only retrieve 2 additional modes, while acGAN recovered all (8) modes. We examined the gradients provided by the discriminators using a density plot (Fig. 4) of the gradient norm for each

	FD	Modes	Quality samples
Vanilla GAN	7.28	17	88%
Uniform (3D)	6.64	20	93.4%
acGAN (3D)	6.65	25	92.9%

Table 1: Results on the Gaussian mixture synthetic data. Our method acGAN could cover allc 25 modes.

discriminator with respect to the input, i.e., $||\nabla_X D(X)||_2$ for $X \in [-2, 2]^2$. Observe that there is a clear progression from a stronger discriminator with more distinct, higher gradients to the weaker discriminator smoother gradients. Additionally, note that the discriminator from the vanilla GAN, which has very high gradient norm values, has gradients for modes not present in the generator: the discriminator has information useful for learning about these missing modes, but the generator does not learn these modes due to vanishing gradients. Our results support both our original hypothesis that missing modes are due to vanishing gradients and that using a coarse-grain discriminator can be used to recover missing modes. To provide further insight, we show the evolution of the gradient norm of each discriminator at training time in the Supplementary Material. We also note that the discontinuities in the gradients is due to the ReLU activation partitioning the subspace through overlapping half-planes, which contrasts the smooth decay of hyperbolic tangent and sigmoid[4] nonlinearities, and we further explore the effect of different nonlinear activation layers on the gradient norm of the weak discriminator in the Supplementary Material.

4.2 Performance of acGAN against existing baselines

In this section, we evaluate the performance of our proposed method (acGAN), on various datasets. All experiments consider the reward shown in Eq. 6. We first conducted a sanity check on 2 mode-dropping datasets: synthetic data consisting of a mixture of 25 Gaussians and Stacked-MNIST with 1000 modes. We then tested it on CIFAR10 and finally show generated samples on celebA dataset (see Supplementary Material). We aim to analyze specific properties such as diversity of generated samples and quality in terms of FID ((Heusel et al. 2017)) score when available, along with convergence of the method (how fast it reaches its minimum FID score). Additionally, our results hint at the emergence of a curriculum during the training process.

All parameters used to obtain the results can be found in the Supplementary Material. We split the batch of inputs between discriminators. We abuse of language with the term *epoch*, which in the context of the current paper means that the generator has been trained on a number of iterations equivalent to an epoch. For example, CIFAR-10 has 50,000 training images and, assuming a batch size of 64, one epoch represents roughly 781 iterations for the generator.

Synthetic Gaussian mixture dataset The synthetic dataset is composed of 25 bivariate Gaussian mixtures arranged in a two-dimensional grid. We launched a single run of 15 epochs for all methods with 3 discriminators. We report 3 measures in Table 1: the Fréchet Distance (FD), the number of recovered modes and the proportion of high quality samples (which is the proportion of samples covering a mode). More details on those metrics can be found in the Supplementary Material.

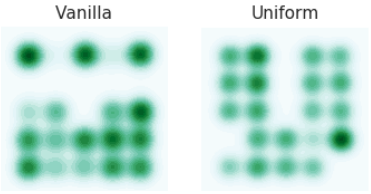

Figure 5: KDE plots of the modes recovered by each examined approach with 3 discriminators.

	Modes (max 1000)	KL
DCGAN (Radford, Metz, and Chintala 2015)	99.0	3.40
ALI (Dumoulin et al. 2016)	16.0	5.40
Unrolled GAN (Metz et al. 2016)	48.7	4.32
VEEGAN (Srivastava et al. 2017a)	150.0	2.95
PacGAN (Lin et al. 2017)	1000.0 ± 0.00	$0.06 \pm 1.0e^{-2}$
GAN+MINE (Belghazi et al. 2018)	1000.0 ± 0.00	$0.05 \pm 6.3e^{-3}$
acGAN (3D)	1000.0 ± 0.00	$7.4e^{-2} \pm 0.0$
acGAN (5D)	1000.0 ± 0.00	$9.65e^{-2} \pm 0.0$

Table 2: Number of modes covered and Kullback-Leiber divergence between the real and generated distributions on Stacked-MNIST. acGAN could recover the 1000 modes.

We compared the performance of our proposed methods to that of the Uniform algorithm and of the vanilla GAN (Goodfellow et al. 2014). Our proposed methods could cover the 25 modes. KDE plots for the 3 discriminators case are shown in Fig. 5.

Stacked-MNIST We use the Stacked-MNIST dataset (Srivastava et al. 2017b) to measure the mode coverage of our proposed approach. The dataset is generated by stacking 3 randomly selected digits from the MNIST dataset: one on each RGB channel to produce a final $28 \times 28 \times 3$ RGB tensor. The dataset has 128,000 training images and is assumed to have 10^3 modes. Results of our experiments are shown in Table 2.

We report our results (averaged over 10 runs) in Table 2 and compare them with other existing baselines in the literature. Our method could recover all 1000 modes like PaCGAN (Lin

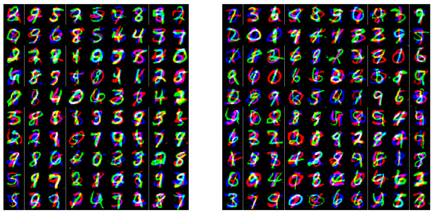

Figure 6: Stacked-MNIST generated samples for acGAN with 3 (left) and 5 (right) discriminators.

[4] $\sigma(y) = 1/1 + e^{-y}$

et al. 2017) and MINE (Belghazi et al. 2018); these two approaches either increase the dimensionality of the generator inputs either by packing multiple samples or by adding a latent code vector which helps overcoming mode collapse. Generated samples are shown in Fig. 6, our results further verify our hypothesis that acGAN is a sensible approach to ensuring good mode coverage and sample quality.

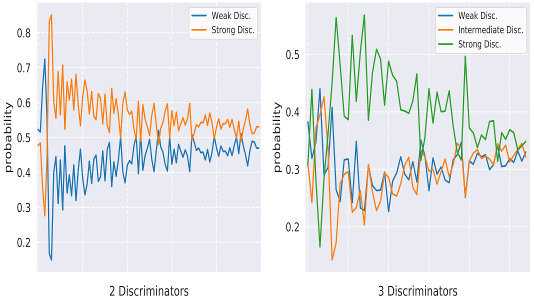

Figure 7: Weight π_i of each discriminator over the training epochs. We can see phase switching at the beginning where each discriminator's weight is dominating before eventually converging to a uniform distribution.

CIFAR-10 We conducted an in-depth study of acGAN's performance on CIFAR-10 by running experiments on 5 independent seeds for 50 epochs each.

We found a particular pattern in the acGAN's learning process: it consists of distinct regimes where one discriminator's weight π_i dominates over the others. To illustrate this, we averaged the sampling probability of each discriminator over every 200 iterations and plotted results in Fig. 7 for 2 and 3 discriminators, respectively. The reported curves suggest that, for $N = 2$ discriminators, the weakest discriminator network is often sampled at the beginning until the generator G learns enough from it, at which point it begins to use the stronger discriminator more often. Note how the strong discriminator is sampled more frequently than the weak one. In fact, because the generator needs to produce samples of higher quality to fool the strong discriminator, training with the latter might take longer as opposed to using weaker discriminators (which are more lenient). By the end of training, all discriminators are being used in equal proportions, meaning that every discriminator plays a complementary role from mode coverage to quality samples. A similar pattern is observed for the 3-discriminators case.

To assess the quality of produced results, we report the minimum Fréchet Inception Distance FID ((Heusel et al. 2017)) (and corresponding epoch) reached in Table 3. The squared FID was computed every epoch with 1,000 held-out samples at training time. As in (Fedus et al. 2017), a ResNet pre-trained on CIFAR-10 was employed to obtain representations for FID computation rather than Inception V3. Proceeding this way yields a more informative score, given that our classifier was trained on the same data as the generative models. Details on the FID score can be found in the Supplementary Material.

We compared our results to (Durugkar, Gemp, and Mahadevan 2016). Since the authors reported that GMAN-1 ($\lambda = 1$) had an overall better performance, we used this version in our experiments and refer to it as GMAN. Previously, we observed that the feedback provided to the generator is shared between all the discriminators. Especially, not all gradient comes from the strong discriminator (unlike for the Vanilla GAN). One might be concerned by a degradation of the quality samples. We show that having more discriminators leads to better mode coverage and samples quality (see the FID curves for an increasing number of discriminators in the Supplementary Material).

Overall, we noticed that acGAN achieved the best FID score when compared to the baseline as presented in Fig. 8 and 9 (plots are shown in a larger format in the Supplementary Material). GMAN performed worse than expected and increasing the number of discriminators did not significantly improve its FID score. We suspect that the original loss function of the GAN (which is equivalent to the Jensen-Shannon divergence minimization) is not a good signal to assess the progress of G. Indeed, (Arjovsky, Chintala, and Bottou 2017) argued and introduced a toy example showing that this version of adversarial nets is not informative when there is little overlap between the supports of the true and approximate distributions, as commonly seen at the beginning of the training process. Finally, not keeping a moving average via a Q-value can lead to high variance.

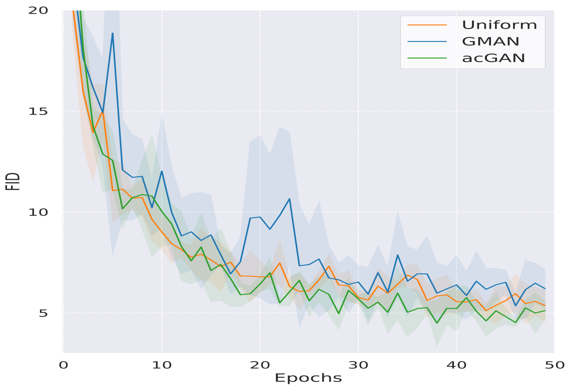

Figure 8: FID scores computed with 1,000 samples at the end of each epoch for different methods with 3 discriminators. acGAN outperforms the baselines Uniform and GMAN.

		Best FID *(epoch)*	Mean Best FID
	Vanilla GAN	*5.02 (20) - 5.28 (27) -* ***4.27 (30)*** *- 4.80 (34) - 4.63 (41)*	*4.80*
	WGAN-GP[5]	*4.29 (43) - 4.24 (28) - 3.98 (47) - 3.99 (37) -* ***3.93 (50)***	*4.08*
3 Disc	Uniform	*4.18 (20) -* ***4.07 (39)*** *- 4.35 (45) - 5.07 (30) - 4.39 (47)*	*4.41*
	GMAN	***3.87 (43)*** *- 4.05 (46) - 5.24 (42) - 5.71 (42) - 4.10 (22)*	*4.59*
	acGAN	*3.93 (39) - 3.57 (38) - 4.25 (42) - 3.43 (40) -* ***3.11 (43)***	**3.66**
5 Disc	Uniform	***3.42 (47)*** *- 3.69 (49) - 4.37 (37) - 3.64 (37) - 3.47 (40)*	*3.72*
	GMAN	*4.58 (44) - 4.40 (20) -* ***3.91 (47)*** *- 4.81 (25) - 4.42 (38)*	*4.42*
	acGAN	*3.62 (35) -* ***2.62 (49)*** *- 4.14 (35) - 2.66 (42) - 3.67 (34)*	**3.34**

Table 3: Best FID scores on CIFAR-10 computed on 1,000 samples during training time (lower is better).

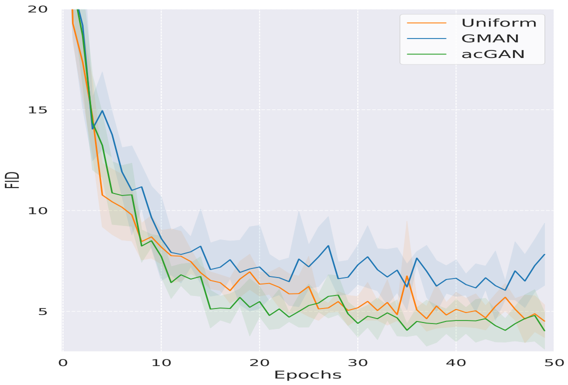

Figure 9: FID curves with 5 discriminators. acGAN presented earlier convergence and reached lower FID values.

5 Conclusion

In this work, we model the training of the generator against discriminators of increasing complexity within a one-student/multiple-teachers paradigm. We address this mixture-of-experts problem under the adversarial bandit setting with full-information, where we rely on the Hedge algorithm to learn the weights assigned to each discriminator in the mixture. Since designing a suitable reward function is a key ingredient to control the shape of the learned policy, we examined two sensible reward functions which relied on sample quality and the GAN loss function. We empirically found the high quality sample reward (Eq. 6) to yield the best results. Keeping a moving average on the rewards helped smoothing the weights put on discriminators and resulted in a more stable mixture.

Then, we demonstrated a complementary regulation mechanism between weak and strong discriminators. While weaker discriminators enjoy smoother properties and provide more informative feedback to the generator, stronger discriminators focus one finer grain detail to ensure sample quality.

Finally, we conducted a series of experiments to show the emergence of a curriculum during the training process. That is, lower-capacity discriminators have higher weights at the beginning but, as the training progresses, higher weights are allocated to higher-capacity discriminators. We showed how existing algorithms could be recovered from our model via the Q-value. The performed experiments showed that our proposed approach leads to an earlier convergence and a better FID score compared to existing baselines in the field, i.e. Uniform and GMAN.

As a direction for future investigation, approaches not relying on the adversarial framework could be investigated to model the non-stationarity of the reward distributions. For example, finding a meaningful representation for the state of the generator could allow the use of contextual bandits algorithms.

[5] We replaced the batch norm layer with instance norm

References

Arjovsky, M., and Bottou, L. 2017. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *CoRR* abs/1701.07875.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 322.

Baransi, A.; Maillard, O.-A.; and Mannor, S. 2014. Sub-sampling for multi-armed bandits. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 115–131. Springer.

Belghazi, I.; Rajeswar, S.; Baratin, A.; Hjelm, R. D.; and Courville, A. C. 2018. MINE: mutual information neural estimation. *CoRR* abs/1801.04062.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 41–48. New York, NY, USA: ACM.

Che, T.; Li, Y.; Jacob, A. P.; Bengio, Y.; and Li, W. 2016. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*.

Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; and Courville, A. 2016. Adversarially Learned Inference. *ArXiv e-prints*.

Durugkar, I. P.; Gemp, I.; and Mahadevan, S. 2016. Generative multi-adversarial networks. *CoRR* abs/1611.01673.

Fedus, W.; Rosca, M.; Lakshminarayanan, B.; Dai, A. M.; Mohamed, S.; and Goodfellow, I. 2017. Many paths to equilibrium: Gans do not need to decrease adivergence at every step. *arXiv preprint arXiv:1710.08446*.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55(1):119–139.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial networks. *CoRR* abs/1406.2661.

Graves, A.; Bellemare, M. G.; Menick, J.; Munos, R.; and Kavukcuoglu, K. 2017. Automated curriculum learning for neural networks. *CoRR* abs/1704.03003.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Klambauer, G.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR* abs/1706.08500.

Hoang, Q.; Nguyen, T. D.; Le, T.; and Phung, D. Q. 2017. Multi-generator generative adversarial nets. *CoRR* abs/1708.02556.

Huang, X.; Li, Y.; Poursaeed, O.; Hopcroft, J. E.; and Belongie, S. J. 2016. Stacked generative adversarial networks. *CoRR* abs/1612.04357.

Juefei-Xu, F.; Boddeti, V. N.; and Savvides, M. 2017. Gang of gans: Generative adversarial networks with maximum margin ranking. *CoRR* abs/1704.04865.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *CoRR* abs/1710.10196.

Kwak, H., and Zhang, B. 2016. Generating images part by part with composite generative adversarial networks. *CoRR* abs/1607.05387.

Lin, Z.; Khetan, A.; Fanti, G. C.; and Oh, S. 2017. Pacgan: The power of two samples in generative adversarial networks. *CoRR* abs/1712.04086.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Information and computation* 108(2):212–261.

Matiisen, T.; Oliver, A.; Cohen, T.; and Schulman, J. 2017. Teacher-student curriculum learning. *CoRR* abs/1707.00183.

Metz, L.; Poole, B.; Pfau, D.; and Sohl-Dickstein, J. 2016. Unrolled generative adversarial networks. *CoRR* abs/1611.02163.

Miyato, T.; Kataoka, T.; Koyama, M.; and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Neyshabur, B.; Bhojanapalli, S.; and Chakrabarti, A. 2017. Stabilizing GAN training with multiple random projections. *CoRR* abs/1705.07831.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR* abs/1511.06434.

Roth, K.; Lucchi, A.; Nowozin, S.; and Hofmann, T. 2017. Stabilizing training of generative adversarial networks through regularization. *CoRR* abs/1705.09367.

Srivastava, A.; Valkov, L.; Russell, C.; Gutmann, M. U.; and Sutton, C. 2017a. Veegan: Reducing mode collapse in gans using implicit variational learning. *ArXiv e-prints*.

Srivastava, A.; Valkoz, L.; Russell, C.; Gutmann, M. U.; and Sutton, C. 2017b. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, 3310–3320.

Tolstikhin, I. O.; Gelly, S.; Bousquet, O.; Simon-Gabriel, C.-J.; and Schölkopf, B. 2017. Adagan: Boosting generative models. In *Advances in Neural Information Processing Systems*, 5424–5433.