Defending Against Free-Riders Attacks in Distributed Generative Adversarial Networks

Zilong Zhao*, Jiyue Huang*, Lydia Y. Chen, and Stefanie Roos

TU Delft, the Netherlands
{z.zhao-8,j.huang-4,y.chen-10,s.roos}@tudelft.nl

Abstract. Generative Adversarial Networks (GANs) are increasingly adopted by the industry to synthesize realistic images using competing generator and discriminator neural networks. Due to data not being centrally available, Multi-Discriminator (MD)-GANs training frameworks employ multiple discriminators that have direct access to the real data. Distributedly training a joint GAN model entails the risk of free-riders, i.e., participants that aim to benefit from the common model while only pretending to participate in the training process. In this paper, we first define a free-rider as a participant without training data and then identify three possible actions: not training, training on synthetic data, or using pre-trained models for similar but not identical tasks that are publicly available. We conduct experiments to explore the impact of these three types of free-riders on the ability of MD-GANs to produce images that are indistinguishable from real data. We consequently design a defense against free-riders, termed DFG, which compares the performance of client discriminators to reference discriminators at the server. The defense allows the server to evict clients whose behavior does not match that of a benign client. The result shows that even when 67% of the clients are free-riders, the proposed DFG can improve synthetic image quality by up to 70.96%, compared to the case of no defense.

Keywords: Multi-Discriminator GANs · Free-rider attack · Anomaly detection · Defense

1 Introduction

Generative Adversarial Networks (GANs) are an emerging methodology to generate synthetic data [3, 11, 38, 39], especially for visual data. GANs are capable of generating real-world-like images and are increasingly adopted by industry for data augmentation and refinement [28]. Their success is attributed to their unique architecture of training two competing neural networks, called discriminator and generator. The well-trained generator can then be used to generate synthetic data. If GANs are trained centrally, a single generator and discriminator are trained iteratively, where the former generates realistic images to fool latter, and the latter then gives feedback to the former by comparing the generated and real images. As a consequence of privacy regulations imposed on

* equal contribution

data sources, e.g., GDPR [33] and HIPAA [4], GANs often have to employ distributed architectures such that they can learn from multiple sources without illegally sharing the raw data.

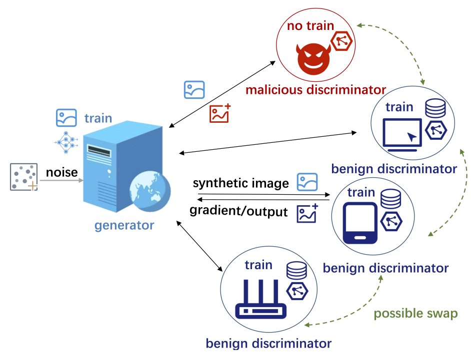


Fig. 1: Architecture of Multi-Discriminator GAN: one generator, and four discriminators, one of which being free-rider.

Multi-Discriminator GAN (MD-GAN), Distributed GAN architectures have been adopted in medical (e.g., medical images) and financial (e.g., financial tabular data) domains [5,7,15,30,37], two areas that have stringent privacy constraints. Typically, as shown in Fig. 1, there are one generator and multiple discriminators, one discriminator for each data source. To learn such an MD-GAN, an iterative training procedure between generator and discriminators takes place. The generator synthesizes images that imitate the real data, whereas the discriminators provide feedback to the generator based on their local image set. A variant of MD-GAN further allows discriminators to exchange their local networks with peers to avoid overfitting [15]. Though such a distributed architecture guarantees that raw data is not shared, it comes with the risk of misbehaving discriminators and the need to defend against them.

Free-riders are a common threat to distributed systems in which the same task is executed by multiple parties, meaning that individuals can hide that they did not execute their task properly as the task is still completed by the other parties in the system. Examples are peer-to-peer file sharing [8,9,25] or Federated Learning systems [27,36]. Free-riders in Federated Learning systems [10,22] try to gain access to the so-called global model from the server, which is aggregated from local models of all contributors without sharing local data. Here, free-riders can simply return the previous global model (possibly with perturbation added) as their contribution. In the context of MD-GAN systems, free-riders aim to gain access to the valuable well-trained generator model without using any real data to train a discriminator. In contrast to Federated Learning systems, where the server model has the same structure as the client model, free-riders and benign discriminators in MD-GAN do not have any information about the concrete generator network. Moreover, it is no mean feat to detect free-riders in MD-GAN as the generator only receives the distributed feedback on how well the synthetic images compared to the real ones, i.e., gradients back-propagated from the discriminator.

In this paper, we aim to answer two research questions: what is the impact of free-riders on MD-GAN frameworks and how can benign participants defend against such free-riders? We conduct the first empirical characterization study on how different numbers and types of free-riders affect the quality of synthetic images of MD-GAN when training image benchmarks. We introduce three attack strategies for free-riders: They obtain a discriminator by i) using a randomly initialized discriminator model without training, ii) training a discriminator model on synthetic data, and iii) using a publicly available pre-trained discriminator model without any additional training. Note that the pre-trained discriminator is not for exactly the same task but for a related task with similar data. Our results show that having 30% or more free-riders considerably degrades MD-GAN's performance, as measured by the Fréchet Inception Distance (FID) score [17]. Free-riders who take advantage of the pre-trained model are less harmful than others but still, free-riders are shown to be a serious issue.

Consequently, we propose a novel **D**efense strategy against **F**ree-riders in MD-**G**AN, termed DFG, where the generator can filter out the contributions of free-riders. The two key steps of DFG are (i) the generator periodically sends out a probing dataset to all discriminators, and (ii) clusters their responses in combination with the reference responses of the "detector", a free-rider and a benign client trained on the generator side. If MD-GAN allows the discriminators to periodically swap models, DFG optionally contains a third defense step at the discriminators, enabling peers to reject swapping with potential free-riders. We evaluate DFG for different attacks, numbers of free-riders, and variants of MD-GAN on CIFAR10 and CIFAR100. Our results indicate that DFG can improve synthetic data quality for all considered scenarios. If the free-riders do not train its discriminator, which is the simplest scenario, DFG reduces FID by 45.05% (CIFAR10) and 33.64% (CIFAR100) with 1 free-rider and 5 benign clients in the system. When varying number of free-riders from 2 to 5, DFG averagely reduces FID by 73.71% (CIFAR10) and 68.39% (CIFAR100). If the free-riders use a pre-trained discriminator, which is the most stealthy type, DFG reduces FID by 60.86% on CIFAR100 dataset when half of the clients are free-riders, and by 70.96% on CIFAR10 dataset even when 67% of the clients are free-riders.

In summary, we make two novel main contributions: (1) A first characterization of three types of free-riders of MD-GAN. (2) Proposing a novel and effective defense strategy DFG and evaluating it on two image benchmarks (i.e., CIFAR10 and CIFAR100).

2 Background on MD-GAN and Free-riders

In this section, we introduce the concept of MD-GANs and our adversarial model.

2.1 Preliminaries on MD-GAN

Key components of MD-GAN are one server and N clients maintaining one generator and N discriminators, respectively. In general, generator and discriminators are all deep neural networks[1] characterized by their model weights. The generator network, $\mathcal{G}$, aims to synthesize images that are indistinguishable from real ones. Each of the N discriminator networks, $\mathcal{D}_i, i \in \{1, 2, ..., N\}$, has direct access to its own set of real images, X_i. They aim to correctly differentiate fake images generated by the generator from real images. Fig. 1 illustrates an example of one generator and four discriminators. For the MD-GAN setting in this paper, all of the clients must join for the full duration of the training process. After training, they obtain the model of the generator to synthesize data.

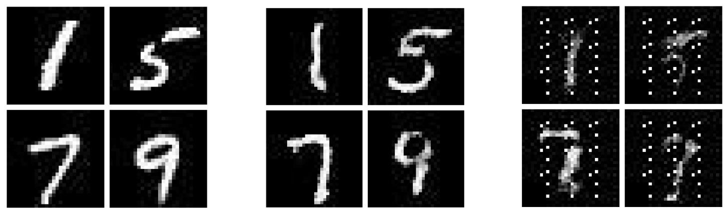
To train an MD-GAN, the generator and discriminators take turns to train and update their network weights over multiple rounds until reaching convergence. One training round consists of multiple mini batches of data. For batch j, discriminator i, and round t, $\mathcal{G}$ produces a synthetic dataset $S^j_{t,i}$ from a vector of Gaussian noise $z^j_{t,i}$. The discriminator trains on $S^j_{t,i}$ together with its real data.

Discriminator training: The discriminator uses its local real images X^j_i (i.e., real image mini batch j at i^{th} discriminator) and the synthetic images $S^j_{t,i}$ from the generator to train itself. Specifically, the generator remains fixed during the discriminator training, we only optimize the discriminator loss and update the weights of discriminator networks through stochastic gradient descent algorithms [32].

Generator training: When calculating generator loss, one can imagine that generator and discriminator are connected as one neural network. The i^{th} discriminator calculates the loss for synthetic images $S^j_{t,i}$ from the generator and back-propagates gradients. After $\mathcal{G}$ receives all of the back-propagated gradients of synthetic images $S^j_{t,i}$ from every i^{th} discriminator, the generator accumulates all the gradients and updates its network weights. During generator training, the weights of the discriminators remain fixed.

2.2 Free-rider adversarial model

We consider free-riders on the discriminator side, i.e., clients want to obtain the final generator model without contributing to the training of MD-GAN. Their goal is not to degrade the image quality of the generator. In this sense, they are rational parties rather than malicious. They deviate from the expected learning procedure to gain utility, namely access to the generator model, without having the necessary data. Free-riders aim to be *stealthy* to overcome any defenses employed by the generator. Such free-riders are local, internal, and active adversaries. In other words, they can only observe and participate in the communication and computation of their own training process. Moreover, free-riders **do not own any data** for training MD-GAN, nor do they have access to the data of others and they cannot observe the communication of others. They do not collude. The assumption of non-collusion is sensible as additional free-riders might decrease the quality of the final model they obtain, so parties are unlikely to reveal their free-riding to others.



(a) Real MNIST images (b) Synthetic images without free-riders (c) Synthetic images with 5 free-riders

Fig. 2: Real v.s. synthetic MNIST images from generators of MD-GAN encountering 0 and 5 free-riders with 5 benign discriminators.

3 Free-rider Attacks in MD-GAN

This section explores different strategies for free riding discriminators. We describe the attack strategy and then evaluate their effectiveness.

3.1 Attacks

Free-riders aim to obtain the generator in the end of the training, such that they can synthesis data of high quality without contributing real data to the training process. To do so, they might need to bypass defenses aimed at detecting free-riding and hence want to be stealthy. A first method to achieve a certain degree of stealthiness is not to follow the random initialization method expected by the generator. The generator can easily compare the gradients provided by a discriminator to those produced by a random model with the same initialization method. If the provided gradients resemble those from a random model, the generator can identify the discriminator as a free-rider, a defense we explore more closely in the next section. To overcome such an straight-forward defense, free-riders can use a different initialization method. In our evaluation, we consider four initialization methods: (i) Kaiming initialization [16], (2) Xavier initialization [12], (3) uniform and (4) normal. Note that all benign clients follow Kaiming initialization (default method by Pytorch).

In order to consider more stealthy free-riders, we note that they have two potential sources of information that they can use to obtain a better model despite not having data to train: i) the synthetic data provided by the generator to generate the gradient feedback and ii) any publicly available pre-trained discriminator models for similar tasks, i.e., GANs for synthesizing images. In summary, we have the following adversarial behaviors for discriminators:

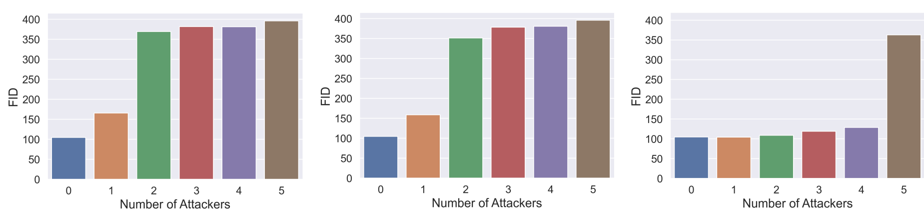
FR$-L$: Also termed lazy free-riders, they choose a random initialization method to initialize the model. Afterwards, they compute the gradients expected by the generator based on the random initial model without any training.

FR$-D$: As detailed in Section 2.1, the generator provides mini batches of synthetic images to the discriminators. So, while a free-rider does not have real data to train on, they can still utilize the synthetic data, which is what FR$-D$ leverages. Concretely, the free-rider uses generated images provided by the generator as "real" data and randomly generates an equal number of images deemed

[1] We interchangeably use terms of networks and models

as fake data by sampling every pixel from a uniform distribution. It then trains its discriminator using these two datasets in the same way as a benign client. In the later phase of the training, i.e., when the synthetic images from generator are very close to real images, FR$-D$'s model is likely relatively good, making it hard to detect them as a free-rider.

FR$-M$: A discriminator outputs whether the data is real and synthetic. Since the output is not class-related, a pre-trained discriminator, which has been used in another GAN framework, can potentially be re-purposed. Note that the generator and benign discriminators do not start training from a pre-trained model themselves because it can affect convergence negatively [1]. But for a free-rider, a well-trained discriminator could be less harmful than a random initial model. Therefore, we assume FR$-M$ is a free-rider that uses a pre-trained discriminator, e.g., one downloaded from the internet. We typically assume that datasets used to train the pre-trained discriminator are different from the ones used to train the current ones. However, to assess the impact of this assumption, we also consider a pre-trained discriminator for the same data in our evaluation.



(a) FR$-L$ (b) FR$-D$ (c) FR$-M$

Fig. 3: Final FID of Multi-Discriminator GAN for different types of free-rider. Number of free-riders varies from 0 to 5, number of benign clients is fixed to 5.

3.2 Empirical analysis on CIFAR-100

Here, we evaluate the effectiveness of our attacks on MD-GAN. We vary the number of attackers between 0 and 5 and always have 5 benign discriminators. CIFAR-100 [18] and MNIST [19] are used as the dataset. We evaluate the quality of generated images by measuring the Fréchet inception distance (FID) [17], which calculates the difference between real and generated images. It is defined as follows:

$$\mathrm{FID} = ||\mu_1 - \mu_2||^2 + \mathrm{tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_1\Sigma_2)^{1/2})$$

where μ_1 and μ_2 denote the feature-wise mean of the real and generated images; Σ_1 and Σ_2 refer to the covariance matrix for the real and generated feature vectors; $||\mu_1 - \mu_2||^2$ refers to the sum-squared difference between the two mean vectors; and tr is the trace linear algebra operation. Intuitively, the lower the FID, the closer the generated and real images. We measure the FID of generated images with an increasing number of attackers. Neural networks and training parameters are provided in Section 5. We start by evaluating lazy free-riders and then turn to the more sophisticated behaviors. For FR$-M$ the pre-trained discriminator is trained on CIFAR100. In general, as stated above, we assume that the pre-trained model is trained on a dataset different from that used by benign clients. For simplicity, we use the same dataset here but provide more experiments on the role of the dataset in Sec. 5.

Baseline of FR$-L$ We first visually motivate why free-riders are important to consider. Fig. 2c shows that MD-GANs can create synthetic images that are very close to the original real MNIST images. Yet, if half the discriminators are free-riders, the images are barely readable and exhibit little similarity with the original images. We now quantify these difference using the FID for CIFAR-100. In Fig. 3a, we can observe that without free-riders, the FID is barely above 100 at the end of the training. With one free-rider, the FID only slightly increases. If two or more free-riders are present, the FID is close to 400, which is the FID without training. Thus, the random initialized discriminator cannot distinguish real and synthetic images and the gradients obtained from the lazy free-riders corrupt the utility of the final generator.

Free data v.s. free model We expect the more sophisticated free-riders to have less negative impact on the quality of the generated images. In Fig. 3, our three types of free-riders are compared. For all types, the impact increases with the number of free-riders, as a large amount of discriminators without useful data is bound to increase the impact. FR$-D$ (Fig. 3b) is only slightly better than FR$-L$ for one or two free-riders. For a higher number of attackers, the model is again almost of the same quality as a random initial model. We conclude that training on synthetic data without any real-world examples is not promising, at least not in the sense that it can result in a useful generator in MD-GAN, which is the goal of both the benign participants and the free-riders.

In contrast, pre-trained discriminators (Fig. 3c) are very effective. For one or two free-riders, the FID is largely unaffected by the free-riders. Even for 3 or 4 free-riders, the increase in FID is small, as it remains below 130, up from 104. If half of the discriminators are free-riders, only having a pre-trained model is insufficient for maintaining high quality, as indicated by Fig. 3c.

4 Defending MD-GAN against Free-riders

Reacting to the severe impact free-riders can have, in this section, we propose DFG, a defense strategy against free-riders in MD-GAN. The **objectives of DFG** are three-fold: **(1)** accurately detecting free-riders in each round and excluding their gradients from accumulation, **(2)** improving the FID for the case when free-riders are present but not considerably decreasing the FID in the absence of free-riders, and **(3)** entailing low additional overhead. Note that the first goal also implies that benign clients should not be classified as free-riders. Indeed, as a low number of free-riders can be tolerated, we consider accidentally classifying free-riders as benign less severe than vice versa. Classifying benign users as free-riders means that they cannot receive earned benefits in the form of the final model. Having a high risk of accidentally being declared a free-rider hence may disincentivize participation. The second part of the second goal is important as a defense that decreases the performance, e.g., by excluding benign clients, in the absence of an attack is unlikely to be adopted, especially if the impact of a low number of free-riders is less than the decrease in image quality caused by the defense. The last goal is necessary because the generator and discriminators might be unwilling to deploy a defense that considerably increases delays, computation, or communication overhead.

4.1 Protocol of DFG for MD-GAN

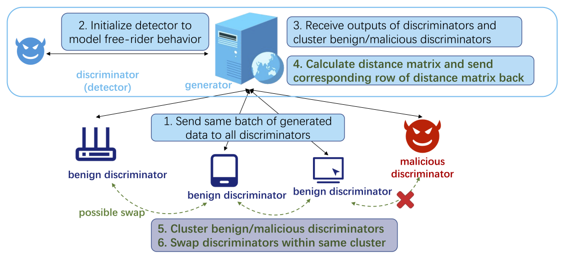


Fig. 4: Key steps of DFG.

The core idea of DFG is to leverage a probing set and detect free-riders based on their responses to the probing set, using either clustering or outlier detection to distinguish responses of free-riders from benign ones. In the following, we detail the 6 steps of DFG, defending free-riders in MD-GAN.

Step 1: In our defense, $\mathcal{G}$ periodically, i.e., every L rounds, generates a probing set $\hat{S}$ to the clients. The set can act as a replacement for $S^j_{t,i}$ (i.e., synthetic images at round t and batch j of the i^{th} discriminator). In contrast to the standard algorithm, DFG sends the same set $\hat{S}$ to all clients. The clients evaluate their discriminators on the set $\hat{S}$ and return the results in the form of a vector. Concretely, for each image s_k, with $1 \leq k \leq |\hat{S}|$, discriminator $\mathcal{D}_i$ computes $\mathcal{D}_i(s_k)$ and the returned vector is:

$$Pr_i(\hat{S}) = \Big(\mathcal{D}_i(s_1), \mathcal{D}_i(s_2), \ldots, \mathcal{D}_i(s_{|\hat{S}|})\Big).$$

Step 2: Additionally, to detect free-riders, $\mathcal{G}$ makes use of two detectors. Concretely, the generator $\mathcal{G}$ randomly initializes two discriminators $\mathcal{D}_{N+1}$ and $\mathcal{D}_{N+2}$. $\mathcal{D}_{N+1}$ is used as a reference model of a free-rider and $\mathcal{D}_{N+2}$ is used as a reference model of a benign client. To train $\mathcal{D}_{N+2}$ in a same way as other benign clients, we assume that there is real data on the server side. $\mathcal{D}_{N+1}$ does not train during the whole training process. Every time when $\mathcal{D}_{N+1}$ and $\mathcal{D}_{N+2}$ receive $\hat{S}$, they compute $Pr_{N+1}(\hat{S})$ and $Pr_{N+2}(\hat{S})$ based on their local discriminators.

Step 3: After the generator collects all the vectors $Pr_i(\hat{S})$, $1 \leq i \leq N+2$, it applies binary clustering, e.g., k-means with k equal to 2., or anomaly detection (e.g. isolation forest) on all vectors $Pr_i(\hat{S})$. Clustering is a promising solution because it divides clients into two groups, which should be benign clients and free-riders. However, this might not work if two free-riders behave differently from each other. Then outlier detection, which identifies unusual behavior such as free-riding when training on local data is considered normal, can be more promising. We only apply clustering or outlier detection and not both. A combined defense, e.g., one that classifies a client as a free-rider if they are classified by any of the two, is bound to have a higher false positive rate, i.e., it accidentally classifies benign clients as free-riders, which we want to avoid. Intuitively, the $Pr_i(\hat{S})$ of a benign client is expected to have a low distance to the $Pr_i(\hat{S})$ of other benign clients, whereas they have a high distance to the $Pr_i(\hat{S})$ of the free-riders, including $\mathcal{D}_{N+1}$. Consequently, when a clustering algorithm is used, we classify all clients in the cluster that contains the $\mathcal{D}_{N+2}$ as benign clients, and the rest are free-riders. When an anomaly detection algorithm is used, all the clients are clustered into two groups: normal and abnormal. The clients in the normal group are considered benign. One exception is that when $\mathcal{D}_{N+2}$ is in the abnormal group, then we treat all the clients in abnormal group as benign clients and normal group members as free-riders. Note that there is a unique scenario where one group of the cluster or the abnormal group contains only $\mathcal{D}_{N+1}$ and another group contains the remaining clients. Accordingly, we believe this case to be no free-rider in the system.

Until now, step 1, 2 and 3 are all defense procedures for standard MD-GAN. But an advanced setting of MD-GAN allows all discriminators to periodically swap their weights between them, we denote this variant as MD-GAN^w. While helping to prevent the over-fitting of discriminator to local data, it also creates challenges for defenses. For this variant, a discriminator is not trained by one single client and hence it is hard to determine whether one party has (not) trained properly. Free-riders can obtain a properly trained discriminator by swapping. This exacerbate the difficulty of differentiating the gradients obtained from free-riders and benign discriminators. To introduce a discriminator-side defense, we take advantage of one information: the benign discriminators know that they are not free-riders. So once a benign client is asked to swap with another that is suspected to be a free-rider, it can refuse. The following steps are added:

Step 4: After the generator has all the vectors $Pr_i(\hat{S})$, $1 \leq i \leq N+1$, they compute a $(N+1) \times (N+1)$ matrix V of pair-wise L2 distances between the Pr vectors of the discriminators, including the detector, i.e., the element V_{ij} is $||Pr_i(\hat{S}) - Pr_j(\hat{S})||_2$. The generator shares the computed distances $V_{i1}, \dots V_{i(N+1)}$ with the i^{th} client.

Step 5: A benign client i then performs binary clustering or anomaly detection on these distances, excluding V_{ii}. The cluster with lower mean distances or the normal group judged by anomaly detection algorithm is taken to be the group of benign clients. The underlying assumption here is that the distance between two properly trained discriminators is less than the distance between benign discriminator and free-rider.

Step 6: A benign client only swaps with parties that are in the same cluster or group as it according to its local clustering or outlier detection, respectively.

All steps are shown in Fig. 4. Due to page limit, the theoretical analysis of computation and communication overhead of DFG is provided in Appendix A.

The overhead of DFG is small in comparison to the cost of normal training operations of MD-GAN.

5 Experimental Evaluation

In this section, we first introduce the experimental setups including datasets, baselines and the testbed. Then we clarify the evaluation metrics to demonstrate the effectiveness of DFG. Last, we summarize and analyze our experimental results for the different free-rider attack strategies with and without defense.

5.1 Experimental setup

Testbed. Experiments are mainly run on two machines, both running Ubuntu 20.04. Each machine is equipped with 32 GB memory, GeForce RTX 2080 Ti GPU and 10-core Intel i9 CPU. Each CPU core has two threads, hence each machine contains 20 logical CPU cores in total. One machine hosts the generator, the other hosts all the discriminators. A third machine with same hardware is used to host 5 discriminators for the experiment with 10 free-riders. The machines are interconnected via 1G Ethernet links. The MD-GAN system is implemented using the Pytorch RPC framework, details are provided in Appendix B. Our code is publicly hosted on github[2].

Datasets. We test our algorithms on two commonly used image datasets: CIFAR10 [18] and CIFAR100 [18]. CIFAR10 and CIFAR100 have 50 000 (10/100 classes) training images in color. Each benign client and the server individually possess 5 000 images, which are evenly distributed over all of the classes.

Baselines. To show the effectiveness of DFG, we simulate MD-GAN and MD-GAN^w with different types of free-riders (i.e., FR$-L$, FR$-D$ and FR$-M$) compared with scenarios without any defense. The pre-trained discriminator for FR$-M$ is trained in the traditional centralized setting with one generator and one discriminator. The pre-trained discriminator is trained on CIFAR100 with the whole dataset for 200 epochs. For both experiments on CIFAR10 and CIFAR100, we use the same pre-trained discriminator to determine the impact of using a similar dataset in contrast to the same dataset. Therefore, we can observe the transfer learning effect on the CIFAR10 experiment with the CIFAR100 pre-trained discriminator.

Notation. We use No_Def_Simple and No_Def_Swap to refer to MD-GAN and MD-GAN^w, respectively, for the scenario without defense. For the scenario with DFG, Def_Simple and Def_Swap are used. In step 3 and 5 of DFG, there are two choices to identify free-riders: (1) binary clustering and (2) anomaly detection. We refer to these two options as Def_X_C and Def_X_{AD} (X is either *Simple* or *Swap*).

Networks. For all experiments, we use the widely adopted and effective Wasserstein GAN with Gradient Penalty (WGAN-GP) [14] structure to train generator and discriminator models. The network of each discriminator consists of three repeated blocks. Each block concatenates 2D Convolution, Instance

[2] `https://github.com/zhao-zilong/DFG`

Normalization, and Leaky Relu layers. $\mathcal{G}$ is also composed of three concatenating blocks. Each block contains 2D Transposed Convolution, Batch Normalization, and Relu layers. The batch size B is set to 500. Since each client owns 5 000 images, there are 10 mini batches per training round. Due to the characteristics of WGAN-GP, the generator is trained once per 5 times the discriminators are trained. Therefore, for each round, the discriminator is trained by all 10 mini batches, but the generator is only trained twice. For DFG, when it evaluates the quality of the discriminators every 10 rounds, it only does that during the first training batch out of two within the round. We repeat each experiment 3 times and report the average.

We fix the number of **benign clients** to 5 for all experiments and vary the number of free-riders from 0 to 5, similar to [5, 15] with the typical setting of 10 clients (in our paper, 5 free-riders + 5 benign clients) in the system. In order to show if and how the system deals with an extreme number of free-riders, we furthermore extend the number of free-riders to 10 for CIFAR10. For CIFAR100, we exclude this experiment due to the high computational overhead. The server broadcasts the initialization method (i.e., Kaiming initialization, default setting by Pytorch) for all discriminators and all benign clients apply this initialization. In contrast, free-riders randomly choose one of the four initialization methods introduced in Sec. 3.1. The "detector" on the server made up of $\mathcal{D}_{N+1}$ and $\mathcal{D}_{N+2}$ uses the same initialization method as benign clients. The total number of training rounds is 100. $\mathcal{G}$ generates 10 000 images every 5 rounds, which are used to evaluate $\mathcal{G}$'s performance in terms of FID. Every 10 rounds, we execute DFG: the generator sends the same probing set $\hat{S}$ of 500 images to all clients and the detectors, and $\hat{S}$ varies over rounds.

5.2 Evaluation Metrics

We compute the final performance of the generated data from $\mathcal{G}$ using the Fréchet inception distance (FID) [17], as introduced in Section 3. To further show the effectiveness of DFG, we use two different metrics. For MD-GAN without swapping, the **precision** and **recall** of the identified "free-riders" are reported. The precision quantifies the fraction of actual free-riders in the group of clients that are detected to be free-riders by our algorithm. The recall is to measure the fraction of free-riders identified by our defense. Here, a free-rider is labelled as Positive and a benign client as Negative for the calculation [29]. Note that recall is not defined in the absence of free-riders. For MD-GAN^w, our focus lies in preventing discriminator swapping between benign and malicious clients. If the DFG prevents a swapping request between two benign clients, we define this as a **wrong prevention**. And if DFG does not stop a swapping between a benign and a malicious client, we call this a **wrong permission**. Intuitively, for the client-side defense, misclassifying a free-rider as a benign client does not increase wrong prevention but increases wrong permission. And misclassifying a benign client as a free-rider increases both wrong prevention and wrong permission. We count the numbers of the prevention and permission and report the percentages of **wrong prevention** and **wrong permission**.

Table 1: Final FID for MD-GAN and MD-GAN^w on FR$-L$ (**A.** is short for the number of free-riders). Best result in **bold**.

Setup	**CIFAR100**						**CIFAR10**						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
No_Def_Simple	104.6	165.6	369.3	381.4	381.7	396.5	79.5	146.5	390.9	439.0	443.8	454.1	470.8
Def_Simple$_C$	102.8	117.6	120.6	124.7	150.4	163.4	78.6	85.4	97.6	121.3	124.3	137.9	152.9
Def_Simple$_{AD}$	102.5	**109.9**	**115.4**	**119.9**	**120.3**	**128.8**	80.1	**80.5**	**92.6**	**116.0**	**118.5**	**128.7**	**140.2**
No_Def_Swap	110.7	193.4	397.9	418.8	418.9	420.8	80.1	193.7	420.8	465.9	470.3	472.1	477.5
Def_Swap$_C$	108.3	120.9	132.5	156.9	177.2	198.1	80.0	110.8	132.8	136.6	155.2	172.3	436.5
Def_Swap$_{AD}$	109.2	**119.8**	**120.1**	**123.0**	**124.2**	**124.6**	80.0	**89.8**	**100.0**	**118.6**	**120.5**	**128.9**	**427.7**

Table 2: Precision(%)/Recall(%) for MD-GAN and MD-GAN^w on FR$-L$.

Setup	**CIFAR100**						**CIFAR10**						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
Def_Simple$_C$	100/-	100/97	100/92	96/83	95/79	95/65	100/-	100/100	100/100	100/89	95/79	95/53	98/37
Def_Simple$_{AD}$	100/-	100/100	100/100	100/100	95/83	98/73	100/-	100/100	100/100	100/100	90/83	85/62	86/45
Def_Swap$_C$	100/-	100/94	100/87	96/80	95/77	95/60	100/-	100/99	100/97	96/84	95/77	94/63	84/10
Def_Swap$_{AD}$	100/-	100/100	100/100	100/100	98/83	98/76	100/-	100/100	100/100	100/100	100/83	100/73	70/15

5.3 Evaluation Results

Defense against FR$-L$ Tab. 1 shows the final FID of MD-GAN and MD-GAN^w with and without DFG. As the number of free-riders increases, so does the severity of the attack and the final FID. The random initialization used by the free-riders lead to wrong predictions and hence useless feedback for the generated data. Note that MD-GAN^w has a higher FID for all datasets and scenarios, including the one without free-riders. So swapping does not necessarily help convergence, e.g., when the data among discriminators has low heterogeneity.

DFG greatly improves the performance for both MD-GAN and MD-GAN^w. Even with 50% of the clients being free-riders, the achieved FID remains below 130 while it is around or even above 400 without a defense. In comparison, without an attack, the final FID is 104.6 and 79.5 for CIFAR100 and CIFAR10, respectively. Hence, the defense almost nullifies the attack in that it results in a FID only slightly higher than the FID in the absence of attacks. Even if there are 10 free-riders, i.e., the free-riders outnumber the benign clients 2:1, DFG still provides protection for MD-GAN. However, in line with our expectation that swapping hinders detection of free-riders, DFG provides little protection for MD-GAN^w if there are 10 free-riders.

Using isolation forest for anomaly detection always makes for a stronger defense than using clustering with 2-means. Clustering tends to fail as two free-riders that use different initialization methods end up with very different models and hence are not clustered together. In contrary, they are both seen as outliers in comparison to benign clients under isolation forest, so anomaly detection is more effective.

Let us zoom in to consider the precision and recall of DFG, shown in Tab. 6 for both CIFAR10 and CIFAR100. Almost all clients identified as free-riders by our defense are indeed free-riders, so the precision is close to 100 for nearly all settings. Indeed, if the number of free-riders is less than 3, the precision is 100. Recall is lower than precision. As we argue in Section 4, precision is more important than recall as a low number of free-riders can be tolerated and we do not want to disincentivize participation from benign clients. As long as less than

Table 3: Wrong Prevention(%)/Wrong Permission(%) for MD-GAN^w on FR$-L$.

Setup	CIFAR100						CIFAR10						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
Def_Swap$_C$	0/-	0/8	10/12	31/35	37/42	37/45	0/-	0/8	5/12	30/33	35/35	39/40	55/68
Def_Swap$_{AD}$	0/-	0/0	0/0	10/0	20/10	33/14	0/-	0/0	0/0	15/0	18/10	24/20	52/50

Table 4: Final FID with FR$-D$ on CIFAR100

Setup	CIFAR100						CIFAR10						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
No_Def_Simple	104.6	158.5	351.7	378.5	381.1	396.1	78.0	144.1	299.8	391.92	434.1	449.8	470.1
Def_Simple$_C$	102.8	106.9	127.9	128.4	156.4	163.6	77.8	102.7	110.6	122.3	125.3	131.7	180.8
Def_Simple$_{AD}$	103.5	**105.5**	**110.4**	**116.5**	**125.0**	**132.1**	77.1	**98.2**	**106.8**	**117.2**	**120.2**	**129.2**	**135.6**

50% of the clients are free-riders, the recall is still above 75%. Once the number of free-riders is at least equal to the number of benign clients, it becomes hard to identify them, especially if swapping and 10 free-riders are present.

For MD-GAN^w, we evaluate the impact of step 4–6 of our defense. Tab. 3 shows the percentage of wrong prevention and wrong permission. In line with the results on FID, precision, and recall, Def_Swap$_{AD}$ performs better than Def_Swap$_C$ in all the experiments. Concretely, there are no wrong permission for Def_Swap$_{AD}$ for up to three free-riders whereas Def_Swap$_C$ can have up to 35% of wrong permission. The fraction of wrong prevention is slightly higher for Def_Swap$_{AD}$ than the fraction of wrong permission. Note that for Def_Swap$_C$, the fraction of wrong prevention is lower than the fraction of wrong permission. For 10 free-riders, more than 50% of prevention and permission are incorrect. The result is in line with what we observe for the final FID in Tab. 1: DFG fails when there are a lot of free-riders and swapping is applied. With the free-riders making up the majority of the clients, it becomes almost impossible to distinguish them initially and once discriminators have been swapped, free-riders can utilize the already-trained discriminators to appear like they participate in the training.

Defense against FR$-D$ FR$-D$ utilizes its synthetic data and data from the generator to train the generator. Thus, for FR$-D$, the expectation is that it can leverage the knowledge obtained from generator to train a better discriminator than FR$-L$. The results are displayed in Tab. 4. Comparing to Tab. 1, we find that without a defense, FR$-D$ exhibits a slightly lower FID for a low number of attackers than FR$-L$. So the negative impact of the attack is slightly less since FR$-D$ performs actual training. Given that randomly generated data instead of real data is used, the positive impact is minimal in terms of improved data quality. Yet, free-riders applying FR$-D$ are still quite different from benign clients and can hence be detected. In the presence of DFG, FR$-D$ leads to a similar performance as FR$-L$. Hence, DFG works for multiple attack strategies.

Defense against FR$-M$ In FR$-M$, free-riders use a pre-trained discriminator model. Recall that for both datasets, the pre-trained discriminator is based on CIFAR100. Based on [1], training a GAN from a pre-trained discriminator means that the loss function of the GAN is saturated and the learning process is slow or unstable. Overall, using a pre-trained discriminator results in the least negative impact of all considered attack strategies. The result is expected as

Table 5: Final FID with FR$-M$.

Setup	**CIFAR100**						**CIFAR10**						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
No_Def_Simple	104.6	**101.4**	**108.7**	**118.9**	**128.9**	362.8	77.6	87.6	107.1	120.1	125.4	174.3	475.9
Def_Simple$_C$	102.4	110.4	118.9	131.0	136.3	**142.0**	77.5	**82.0**	**94.2**	**106.9**	**110.6**	**116.5**	**138.2**
Def_Simple$_{AD}$	104.2	104.3	110.1	125.3	134.4	172.3	77.9	85.9	98.3	108.7	116.1	139.6	143.8

these free-riders provide discriminators of actual relevance rather than ones that are random or trained on random data.

The exact results differ slightly depending on the combination of training dataset and choice of pre-trained discriminator. If CIFAR100 is used both for the pre-trained discriminator and the training dataset for MD-GAN, using DFG actually decreases the performance slightly if the number of attackers is less than 50%. DFG struggles to distinguish benign clients and free-riders (precision and recall result for DFG on FR$-M$ is provided in Appendix C). Indeed, the free-riders appear very similar to each other as they all start from the same pre-trained generator. In contrast, the benign clients are initially more diverse, which can make them accidentally be considered as outliers. Thus, DFG removing clients just degrades the performance and does not remove any negative influence from the training. If 5 clients are free-riders, FID does not converge without a defense. DFG here improves the situation, though the results are worse than for FR$-L$ and FR$-D$ as detection is harder. For CIFAR10, the pre-trained discriminator is for a different dataset than the training dataset. Thus, the discriminator is less suitable and degrades the FID more than for CIFAR100 if no defense is applied. However, the FID is still better than for other types of free-riders. DFG again largely nullifies the impact of the attack.

A key difference when defending against FR$-M$ in comparison to previous attacks lies in the choice of defense. For FR$-M$, clustering is more effective than anomaly detection while the opposite is observed for FR$-L$ and FR$-D$. While the pre-trained discriminators may be different from the actual discriminators trained by benign clients, they are not different enough to be considered an anomaly.

In addition to the attacks considered above, we also evaluated the combined behavior of FR$-D$ and FR$-M$: FR$-DM$ obtains a publicly available pre-trained discriminator and then during the training uses the images provided by the generator to fine-tune the discriminator. Due to space constraints, we present the detailed results in Appendix D. In a nutshell, FR$-DM$ has the least effect on the final FID. Even with 5 attackers, the FID increases by less than 50%. DFG can largely overcome the attack.

All results indicate that DFG is an effective defense that only fails if the number of free-rider considerably exceeds the number of benign clients. It hardly ever excludes benign clients and only has minimal impact in the absence of attacks. Notably, DFG is effective against different types of free-riders.

6 Related work

In this section, we summarize the related studies on multi-discriminator GAN frameworks and free-rider attacks in distributed learning systems.

MD-GAN: Overcoming the data privacy issues of centralized GANs [20, 21, 23], distributed GANs [5, 7, 13, 15, 30, 31, 40] enable multiple data owners to collaboratively train GAN systems. Existing distributed GAN frameworks can be summarized as Federated Learning GANs (FLGANs) [13, 31, 40] and MD-GANs [5, 7, 15, 30]. In FLGANs, a client trains both a generator and a discriminator network and a server aggregates both networks from all clients. Consequently, FLGANs require all participants to have high computational capacity. In contrast, MD-GAN architectures offload the intensive training of the generator to the server and keep the lighter training of the discriminator on the client side. In this manner, MD-GANs are also able to involve a massive number of edge nodes [6, 35]. The various architectures of MD-GAN differ with regard to model exchange between discriminators. AsynDGAN [5] and GMAN [7] are elementary MD-GAN architectures where discriminators only directly communicate with the generator. In order to improve the drawbacks of MD-GAN when discriminators only own small datasets, Hardy et al. [15] propose that discriminators are swapped between clients, opening an opportunity for free-riders to act stealthily.

Free-riders: The concept of free-riders first emerged in economics [2] but has been essential in various distributed systems. In peer-to-peer file-sharing systems, free-riders join to download files without uploading any files [8, 24]. In Federated Learning systems [27, 36], Lin et. al. [22] first propose stealthy free-rider attacks for image classification: instead of sending a random model, free-riders send the global model of the previous round back with small perturbation noises added or provide a fake gradient using the previous difference of weights. Defenses are designed accordingly based on the DAGMM [41] network, which is a recent anomaly detection method so as to catch the differences on deep feature by gradients for free-riders. Fraboni et. al. [10] further explore the attack of adding perturbation noises [22] and provide a convergence guarantee of the global model in the presence of a single free-rider. However, as both studies are concerned with Federated Learning systems, where the clients and the server are curating models of the same structure, they are not directly applicable to MD-GAN systems where the server and client train different types of models. Additionally, none of them has provided a systematic study on the influence of (multiple) free-riders. To the best of our knowledge, this paper is the first to study free-riders in MD-GANs.

7 Conclusion

In this first study of free-riders on MD-GAN, we explore multiple types of free-rider attacks. They all can severely degrade the quality of the trained generator, emphasizing the need for a defense. Our defense, DFG, distinguishes free-riders from benign clients through clustering or anomaly detection. It is highly effective and efficient. With the FID being about 100 without attacks and 400 with attacks and no defense, DFG enables the system to maintain an FID of less than 130 in the presence of attacks, even if the attackers make up 50% of the clients. Future work should target more malicious adversaries that actively aim to degrade performance.

References

1. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=Hk4_qw5xe
2. Baumol, W.J.: Welfare economics and the theory of the state. In: The encyclopedia of public choice, pp. 937–940. Springer (2004)
3. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=B1xsqj09Fm
4. Centers for Medicare & Medicaid Services: The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/ (1996)
5. Chang, Q., Qu, H., Zhang, Y., Sabuncu, M.R., Chen, C., Zhang, T., Metaxas, D.N.: Synthetic learning: Learn from distributed asynchronized discriminator GAN without sharing medical image data. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. pp. 13853–13863. Computer Vision Foundation / IEEE (2020). https://doi.org/10.1109/CVPR42600.2020.01387, https://openaccess.thecvf.com/content_CVPR_2020/html/Chang_Synthetic_Learning_Learn_From_Distributed_Asynchronized_Discriminator_GAN_Without_Sharing_CVPR_2020_paper.html
6. Correia, C., Correia, M., Rodrigues, L.: Omega: a secure event ordering service for the edge. In: 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2020, Valencia, Spain, June 29 - July 2, 2020. pp. 489–501. IEEE (2020). https://doi.org/10.1109/DSN48063.2020.00062, https://doi.org/10.1109/DSN48063.2020.00062
7. Durugkar, I.P., Gemp, I., Mahadevan, S.: Generative multi-adversarial networks. CoRR **abs/1611.01673** (2016), http://arxiv.org/abs/1611.01673
8. Feldman, M., Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems. SIGecom Exch. **5**(4), 41–50 (2005). https://doi.org/10.1145/1120717.1120723, https://doi.org/10.1145/1120717.1120723
9. Feldman, M., Papadimitriou, C.H., Chuang, J., Stoica, I.: Free-riding and whitewashing in peer-to-peer systems. IEEE J. Sel. Areas Commun. **24**(5), 1010–1019 (2006). https://doi.org/10.1109/JSAC.2006.872882, https://doi.org/10.1109/JSAC.2006.872882
10. Fraboni, Y., Vidal, R., Lorenzi, M.: Free-rider attacks on model aggregation in federated learning. In: Banerjee, A., Fukumizu, K. (eds.) The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 130, pp. 1846–1854. PMLR (2021), http://proceedings.mlr.press/v130/fraboni21a.html
11. Gao, Y., Singh, R., Raj, B.: Voice impersonation using generative adversarial networks. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018. pp. 2506–2510. IEEE (2018). https://doi.org/10.1109/ICASSP.2018.8462018, https://doi.org/10.1109/ICASSP.2018.8462018
12. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, D.M. (eds.) Proceedings of the

Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010. JMLR Proceedings, vol. 9, pp. 249–256. JMLR.org (2010), http://proceedings.mlr.press/v9/glorot10a.html
13. Guerraoui, R., Guirguis, A., Kermarrec, A., Merrer, E.L.: Fegan: Scaling distributed gans. In: Silva, D.D., Kapitza, R. (eds.) Middleware '20: 21st International Middleware Conference, Delft, The Netherlands, December 7-11, 2020. pp. 193–206. ACM (2020). https://doi.org/10.1145/3423211.3425688, https://doi.org/10.1145/3423211.3425688
14. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. arXiv preprint arXiv:1704.00028 (2017)
15. Hardy, C., Merrer, E.L., Sericola, B.: MD-GAN: multi-discriminator generative adversarial networks for distributed datasets. In: 2019 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2019, Rio de Janeiro, Brazil, May 20-24, 2019. pp. 866–877. IEEE (2019). https://doi.org/10.1109/IPDPS.2019.00095, https://doi.org/10.1109/IPDPS.2019.00095
16. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
17. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf
18. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
20. Li, C., Alvarez-Melis, D., Xu, K., Jegelka, S., Sra, S.: Distributional adversarial networks. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=HJtk5HkDG
21. Liang, X., Hu, Z., Zhang, H., Gan, C., Xing, E.P.: Recurrent topic-transition GAN for visual paragraph generation. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 3382–3391. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.364, https://doi.org/10.1109/ICCV.2017.364
22. Lin, J., Du, M., Liu, J.: Free-riders in federated learning: Attacks and defenses. CoRR **abs/1911.12560** (2019), http://arxiv.org/abs/1911.12560
23. Liu, G., Khalil, I., Khreishah, A.: Zk-gandef: A GAN based zero knowledge adversarial training defense for neural networks. In: 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019. pp. 64–75. IEEE (2019). https://doi.org/10.1109/DSN.2019.00021, https://doi.org/10.1109/DSN.2019.00021
24. Locher, T., Moor, P., Schmid, S., Wattenhofer, R.: Free riding in bittorrent is cheap. In: Kohler, E., Minshall, G. (eds.) 5th ACM Workshop on Hot Topics in Networks - HotNets-V, Irvine, California, USA, November 29-30, 2006. ACM SIGCOMM (2006), http://conferences.sigcomm.org/hotnets/2006/locher06free.pdf

25. Locher, T., Moore, P., Schmid, S., Wattenhofer, R.: Free riding in bittorrent is cheap. In: 5th Workshop on Hot Topics in Networks (HotNets) (2006), http://eprints.cs.univie.ac.at/5696/
26. Maji, P., Mullins, R.: On the reduction of computational complexity of deep convolutional neural networks. Entropy **20**(4) (2018). https://doi.org/10.3390/e20040305, https://www.mdpi.com/1099-4300/20/4/305
27. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Singh, A., Zhu, X.J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA. Proceedings of Machine Learning Research, vol. 54, pp. 1273–1282. PMLR (2017), http://proceedings.mlr.press/v54/mcmahan17a.html
28. Peres, R.S., Azevedo, M., Araújo, S.O., Guedes, M., Miranda, F., Barata, J.: Generative adversarial networks for data augmentation in structural adhesive inspection. Applied Sciences **11**(7), 3086 (2021)
29. Powers, D.M.W.: Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. CoRR **abs/2010.16061** (2020), https://arxiv.org/abs/2010.16061
30. Qu, H., Zhang, Y., Chang, Q., Yan, Z., Chen, C., Metaxas, D.N.: Learn distributed GAN with temporary discriminators. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (eds.) Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVII. Lecture Notes in Computer Science, vol. 12372, pp. 175–192. Springer (2020). https://doi.org/10.1007/978-3-030-58583-9_11, https://doi.org/10.1007/978-3-030-58583-9_11
31. Rasouli, M., Sun, T., Rajagopal, R.: Fedgan: Federated generative adversarial networks for distributed data. CoRR **abs/2006.07228** (2020), https://arxiv.org/abs/2006.07228
32. Robbins, H., Monro, S.: A stochastic approximation method. The annals of mathematical statistics pp. 400–407 (1951)
33. Union, E.: Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Official Journal L110 **59**, 1–88 (2016-05-04)
34. Wang, Y., Wu, C., Herranz, L., Weijer, J., Gonzalez-Garcia, A., Raducanu, B.: Transferring GANs: Generating Images from Limited Data: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VI, pp. 220–236 (09 2018)
35. Wang, Z., Xu, H., Liu, J., Huang, H., Qiao, C., Zhao, Y.: Resource-efficient federated learning with hierarchical aggregation in edge computing. In: 40th IEEE Conference on Computer Communications, INFOCOM 2021, Vancouver, BC, Canada, May 10-13, 2021. pp. 1–10. IEEE (2021). https://doi.org/10.1109/INFOCOM42981.2021.9488756, https://doi.org/10.1109/INFOCOM42981.2021.9488756
36. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2), 12:1–12:19 (2019). https://doi.org/10.1145/3298981, https://doi.org/10.1145/3298981
37. Zhao, Z., Birke, R., Kunar, A., Chen, L.Y.: Fed-tgan: Federated learning framework for synthesizing tabular data. arXiv preprint arXiv:2108.07927 (2021)

38. Zhao, Z., Kunar, A., Birke, R., Chen, L.Y.: Ctab-gan: Effective table data synthesizing. In: Balasubramanian, V.N., Tsang, I. (eds.) Proceedings of The 13th Asian Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 157, pp. 97–112. PMLR (17–19 Nov 2021), https://proceedings.mlr.press/v157/zhao21a.html
39. Zhao, Z., Kunar, A., Birke, R., Chen, L.Y.: Ctab-gan+: Enhancing tabular data synthesis. arXiv preprint arXiv:2204.00401 (2022)
40. Zhao, Z., Wu, H., van Moorsel, A., Chen, L.Y.: Gtv: Generating tabular data via vertical federated learning. arXiv preprint arXiv:2302.01706 (2023)
41. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=BJJLHbb0-

Appendix A Overhead Analysis

We discuss the computation and communication complexity of DFG in relation to the complexity of MD-GANs without any defense. We start by analyzing the additional computation overhead, which is the main cost factor as by our experiments.

Computation of DFG for MD-GAN: In DFG, the clients need to compute $D_i(s_k)$ for $s_k \in \hat{S}$. However, when training without the defense, they also need to compute $D_i(s'_k)$ for $s'_k \in S^j_{t,i}$ with $|S^j_{t,i}| = |\hat{S}|$ to evaluate the quality of their current model. Hence, there is no additional overhead on the client-side with regard to computation. The generator has to compute the pair-wise distances between $N+2$ vectors of length $|\hat{S}|$ and perform a binary clustering or anomaly detection of $N+2$ values. The distance computation has complexity $\mathcal{O}\left(|\hat{S}|N^2\right)$. And for instance, if K-means is used for clustering, then the complexity is $\mathcal{O}\left(N^2\right)$. If isolation forest is used for anomaly detection, the complexity is $\mathcal{O}\left(tNlog_2(N)\right)$ where t is the number of trees in the Isolation Forest. As the number of images is expected to be much larger than the number of clients (and trees), we have a complexity of $\mathcal{O}\left(|\hat{S}|N^2\right)$. In comparison, the cost of a training round without DFG is $\mathcal{O}\left(|\hat{S}|MQ\right)$ where M is the image size and Q a factor related to the structure and size of the neural network [26]. When it comes to the complexity of the detectors, $\mathcal{D}_{N+1}$ does not train so it only computes $D_i(s_k)$ for $s_k \in \hat{S}$ with the complexity of $\mathcal{O}\left(|\hat{S}|MQ\right)$. $\mathcal{D}_{N+2}$ trains its discriminator with real data, the training also has the complexity of $\mathcal{O}\left(|\hat{S}|MQ\right)$. According to common real-world applications of MD-GAN [5], M should be much larger than N^2, meaning that the computation complexity of the normal MD-GAN execution by far exceeds the complexity of DFG. If MD-GAN allows swapping discriminators, the clustering or anomaly detection is also performed on the client-side, incurring an overhead of $\mathcal{O}\left(N^2\right)$ or $\mathcal{O}\left(tNlog_2(N)\right)$ for each client.

Communication of DFG for MD-GAN and MD-GAN^w: For MD-GAN, the only additional communication overhead is on the detector side and client side, namely sending the values $D_i(s_k)$ to the generator. Hence the communication complexity is $\mathcal{O}(N|\hat{S}|)$. During a normal MD-GAN round, the generator sends $N|\hat{S}|$ images of size M, so the complexity is $\mathcal{O}(MN|\hat{S}|)$. If swapping discriminators (i.e., MD-GAN^w) is used, the generator sends N vectors of length $N+2$ that allow the clients to perform clustering or anomaly detection, so the communication complexity of DFG is $\mathcal{O}(N|\hat{S}| + N^2)$. In practice, we expect a low number of clients in comparison to images, so $\mathcal{O}(N|\hat{S}|)$ is the dominating factor, which, as explained for the computation overhead, is much lower than the complexity of an MD-GAN execution without defense.

Appendix B Implementation of MD-GAN

MD-GAN is implemented using the Pytorch v1.8.1 RPC framework. This choice enables the generator to control the flow of of training steps with ease. Clients just join the system, then wait to be initialized and assigned tasks. To parallelize the training across all clients, RPC provides a function *rpc_async()* that allows the generator to make non-blocking RPC calls to run functions at a client. To implement synchronization points, RPC provides a blocking function *wait()* for the return from a previous call to the function *rpc_async()*. The return of *rpc_async()* is a *future* type object. Once *wait()* is called on this object, the process is blocked until the return values are received from the client. We use *rpc_async()* for both training the generator and discriminators networks. The training of the generator network does not continue until it receives input from all discriminators and vice versa.

Appendix C Precision and Recall of FR$-M$

Table 6: Precision(%)/Recall(%) for MD-GAN and MD-GAN^w on FR$-M$.

Setup	**CIFAR100**						**CIFAR10**						
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**10 A.**
Def_Simple$_C$	100/-	100/78	76/67	70/67	65/67	65/62	100/-	100/100	100/89	100/86	100/78	80/67	88/56
Def_Simple$_{AD}$	100/-	70/78	68/40	67/22	62/22	58/22	100/-	100/89	100/78	95/67	90/62	82/65	76/52

Appendix D Free-rider: FR$-DM$

FR$-DM$: This type of free-rider is a combination of FR$-D$ and FR$-M$. It obtains a publicly available discriminator and also uses the generated image from generator as real data to fine-tune itself. The intuition behind this setting is that the pre-trained discriminator can be previously trained on a different dataset. If we use generated images to train the pre-trained discriminator, it indirectly learns the features from current synthetic data that incorporates the knowledge of the benign discriminators. his is similar to the principle of transfer learning [34].

Table 7: Final FID with FR$-DM$.

Setup	**CIFAR100**						**CIFAR10**					
	0 A.	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**	**0 A.**	**1 A.**	**2 A.**	**3 A.**	**4 A.**	**5 A.**
No_Def_Simple	100.32	103.24	107.33	115.67	130.48	145.27	77.22	83.55	88.40	90.50	90.56	103.42
Def_Simple$_C$	101.35	**102.70**	**104.32**	**104.89**	**105.95**	**106.55**	77.38	**80.65**	**81.04**	**81.33**	**82.10**	**85.63**
Def_Simple$_{AD}$	103.10	102.93	105.31	111.29	112.50	114.10	77.31	82.87	83.33	85.30	90.55	96.13

Defense against FR$-DM$ The results for FR$-DM$ in terms of FID are displayed in Tab. 7. Comparing the no defense case (i.e., No_Def_Simple) in Tab. 5 and Tab. 7, FR$-DM$ does not significantly increases the FID and less harmful than FR$-M$, FR$-D$ and FR$-L$. But Def_Simple$_C$ still averagely decreases FID by 11.62% and 9.69% on CIFAR100 and CIFAR10. Training with generated data let the well-trained discriminator adopt features from current data. The No_Def_Simple result on CIFAR10 confirms that transfer learning with a pre-trained discriminator works in MD-GAN framework.