Neurocomputing 468 (2022) 1–10

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Federated synthetic data generation with differential privacy

Bangzhou Xin [a,b], Yangyang Geng [a], Teng Hu [b,c], Sheng Chen [a], Wei Yang [a,*], Shaowei Wang [d], Liusheng Huang [a]

[a] University of Science and Technology of China, Hefei, China
[b] Institute of Computer Application, China Academy of Engineering Physics, Mianyang, China
[c] Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China
[d] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, China

ARTICLE INFO

Article history:
Received 21 December 2020
Revised 31 August 2021
Accepted 2 October 2021
Available online 7 October 2021
Communicated by Zidong Wang

Keywords:
Synthetic data
Generative adversarial network
Federated learning
Differential privacy

ABSTRACT

Distributed machine learning has attracted much attention in the last decade with the widespread use of the Internet of Things. As a generative model, Generative Adversarial Network (GAN) has excellent empirical performance. However, the distributed storage of data and the fact that data cannot be shared for privacy reasons in a federated learning setting bring new challenges to training GAN. To address this issue, we propose private FL-GAN, a differentially private GAN based on federated learning. By strategically combining the Lipschitz condition with differential privacy sensitivity, our model can generate high-quality synthetic data without sacrificing the training data's privacy. When communication between clients becomes the main bottleneck for federated learning, we propose to use a serialized model-training paradigm, which significantly reduces communication costs. Considering the distributed data is often non-IID in reality, which poses challenges to modeling, we further propose universal private FL-GAN to approach this problem. We not only theoretically prove that our algorithms can provide strict privacy guarantees with differential privacy, but also experimentally demonstrate that our models can generate satisfactory data while protecting the privacy of the training data, even if the data is non-IID.

1. Introduction

The Internet of Things has led to a proliferation of data and has also triggered the rapid development of distributed machine learning models. To meet the needs of different scenarios, many machine learning models [1,2] have been developed for different purposes. Most of these models perform well based on a large amount of training data. However, for some specific reasons, the data is uneasy to access. For example, for privacy reasons, data holders prefer to store data that contains personally identifiable information locally rather than share it.

Fortunately, the advent of generative models has provided an effective way to mitigate data scarcity. As a superior generative model, Generative Adversarial Network (GAN) [3], and its variants [4] are capable of generating data that can be spuriously realistic. By learning the distribution of the training data, GAN is capable of generating an unlimited amount of high-quality data based on the learned distribution. However, although GAN only learns about the distribution of training data through adversarial training, repeated sampling of it may still expose data privacy [5]. For instance, Hitaj et al. [6] introduced an active inference attack model that can reconstruct training samples from the generated samples. Therefore, there is an urgent need for a generation model that can generate high-quality samples while protecting the privacy of training data.

Differential privacy (DP) [7] is a de facto concept to preserve the data privacy without sacrificing data utility (see Definition 1). Many utility-optimized algorithms have been proposed focusing on applying DP in deep learning networks [8,9]. In order to keep track of the privacy budget in deep learning, Abadi et al. [10] proposed the moments accountant technique, which can calculate the privacy loss in the training process more accurately. To protect the privacy of training data, two earlier differentially private GAN: DPGAN [11], and GANobfuscator [12] have been proposed. These two works are dedicated to training differential privacy generative models on a single dataset. However, the distributed case is not taken into account.

To maximize the use of distributed stored data without violating user privacy, the term federated learning (FL) was introduced in 2016 by McMahan et al. [13]. It is a distributed machine learning setting where multiple clients collaborate in solving a machine

[*] Corresponding author.
E-mail address: qubit@ustc.edu.cn (W. Yang).

https://doi.org/10.1016/j.neucom.2021.10.027
0925-2312/© 2021 Elsevier B.V. All rights reserved.

learning problem under the orchestration of a central server or service provider. To ensure privacy, the raw data for each client is stored locally and not exchanged; instead, the learning goals are achieved by periodically aggregating updates of the model parameters [14]. This area has received significant attention in recent years, both from a research and application perspective. As the first work to train a differentially private generative model with federated learning, DP-FedAvg-GAN [15] focuses on a large number of small mobile device scenarios. In fact, what is often encountered in the industry is a more centralized scenario of a small number of data centers, called a "cross-silo", where several organizations work together, for example, which is also the main application scenario of this paper.

In this paper, we present private federated learning of GAN (private FL-GAN), a novel method to train a GAN in a distributed way, which can use DP to preserve the privacy of training data. Our private FL-GAN addresses the generation of privacy-preserving data under the setting of "cross-silo" in federated learning. By strategically combining the Lipschitz condition with the differential privacy sensitivity, we minimize the impact of noise addition on GAN performance. Serialized training of models between clients reduces communication overhead and minimizes the risk of training data being leaked. Our contributions are summarized as follows:

- We present private FL-GAN, the first privacy-preserving data generation model that can be applied to the "cross-silo" setting. By organically adopting Elastic Weight Consolidation (EWC) [16], and Memory Replay [17] technologies, we extend it to non-IID datasets.
- Using RDP [18] and moments accountant to calculate privacy loss, we achieve accurate calculation of it and theoretically prove that our algorithms can provide privacy guarantee with differential privacy.
- We conduct comprehensive experimental performance studies of our schemes using different real datasets and demonstrate that with a reasonable privacy budget, private FL-GAN can generate high-quality data while protecting privacy, even if the datasets are non-IID.
- By implementing an attack model trained with a shadow model against private FL-GAN, we demonstrate that our algorithm is immune to membership inference attack.

Compared with the preliminary conference version [19] of our paper, we have expanded this work to be able to generate data on non-IID datasets. Moreover, we have introduced a new technique for tracking the privacy budget to provide a tighter bound on the privacy budget, allowing for less noise without compromising the privacy guarantees. In addition to these, we also verify the security of our algorithm using a membership inference attack.

The remainder of this paper is organized as follows. The preliminaries and problem descriptions are introduced in Section 2 and Section 3, respectively. We detailedly present the private FL-GAN in Section 4. Then, Section 5 extends the private FL-GAN to non-IID datasets. We perform comprehensive experiments in Section 6. Finally, we conclude this paper in Section 7.

2. Preliminaries

First, we review two concepts used in our work, differential privacy [7] and Rényi differential privacy (RDP) [18], respectively. The mathematical notations frequently used in this paper are summarized in Table 1.

Table 1
Frequently used symbols.

Symbol	Description
$\mathscr{D}$	A dataset
$\mathscr{M}$	A randomised algorithm
ϵ, δ, α	Differential privacy parameters
$\mathscr{D}_1, \mathscr{D}_2$	Any two adjacent datasets
$\mathscr{S}$	A set of possible outputs
aux	Auxiliary input
$p_n(\boldsymbol{x})$	The training data distribution
$p(\boldsymbol{z})$	The latent variable distribution
$\mathscr{F}$	Fisher information matrix
q	Batch sampling rate
σ	Noise multiplier
$Adam$	A method for Stochastic Optimization

Definition 1. ((ϵ, δ)-Differential Privacy [7]). If a randomized mechanism $\mathscr{M}$ satisfies (ϵ, δ)-differential privacy, for any two inputs $\mathscr{D}_1$ and $\mathscr{D}_2$ differing in at most a single point (i.e., adjacent datasets), and for any subset of outputs $\mathscr{S}$:

$$Pr[\mathscr{M}(\mathscr{D}_1) \in \mathscr{S}] \leqslant exp(\epsilon) \cdot Pr[\mathscr{M}(\mathscr{D}_2) \in \mathscr{S}] + \delta \tag{1}$$

where parameters ϵ and δ are non-negative real numbers, $\mathscr{M}(\mathscr{D}_1)$ and $\mathscr{M}(\mathscr{D}_2)$ are the outputs of the mechanism for input datasets $\mathscr{D}_1$ and $\mathscr{D}_2$, respectively, and Pr denotes the probability of an event. This definition specifies that the outputs of the random mechanism $\mathscr{M}$ should satisfy certain restrictions on the adjacent datasets $\mathscr{D}_1$ and $\mathscr{D}_2$. In simple terms, it means that no individual's data has a large impact on the output of the algorithm. Small ϵ indicates strong privacy protection, and vice versa. δ indicates the probability of privacy protection failure, and its recommended value should be smaller than the inverse of the database size.

Rényi Differential Privacy (RDP) [18] is a variant of differential privacy that has recently been proposed. It offers a unified view of the ϵ-differential privacy (pure DP), (ϵ, δ)-differential privacy (approximate DP), and the related notion of Concentrated Differential Privacy (CDP) [20]. The RDP point of view on differential privacy is particularly useful when the dataset is accessed by a sequence of randomized mechanisms. The following is the definition of RDP.

Definition 2. (Rényi Differential Privacy [18]). We say that a mechanism $\mathscr{M}$ is (α, ϵ)-RDP with order $\alpha \in (1, \infty)$ if for all adjacent datasets $\mathscr{D}_1, \mathscr{D}_2$

$$RDP(\alpha) := D_\alpha(\mathscr{M}(\mathscr{D}_1) \| \mathscr{M}(\mathscr{D}_2)) \leqslant \epsilon \tag{2}$$

where $D_\alpha(P\|Q) \triangleq \frac{1}{\alpha-1} log E_{x\sim Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha$ is the Rényi divergence of order α between two distributions P and Q. While RDP shares many important properties with the standard definition of differential privacy, some excellent properties come with it, allowing it to perform a more rigorous analysis of heterogeneous compound mechanisms. Here, we highlight two key properties that are relevant to this paper.

Lemma 1. *(Composition Theorem of RDP [18]). If $\mathscr{M}_1, \mathscr{M}_2$ respectively satisfy $(\alpha, \epsilon_1), (\alpha, \epsilon_2)$-RDP for $\alpha \geqslant 1$, then the composition of two mechanisms $(\mathscr{M}_1(\mathscr{D}), \mathscr{M}_2(\mathscr{D}))$ satisfies $(\alpha, \epsilon_1 + \epsilon_2)$-RDP.*

Lemma 2. *(RDP to DP conversion, Proposition 3 of [18]). If $\mathscr{M}$ obeys (α, ϵ)-RDP, then $\mathscr{M}$ obeys $(\epsilon + log(1/\delta)/(\alpha - 1), \delta)$-DP for all $0 < \delta < 1$.*

3. Problem description

The application background of our work is the “cross-silo” setting of federated learning, which corresponds to multiple data centers in reality with large amounts of data. Although a single data center has many data, the data is often homogeneous and cannot cover all possible samples. The generalization performance of the model trained in this way is low, so multiple data centers need to cooperate. The scenario we explore includes two parts: The server and the clients that own datasets. Both client-client and client–server can communicate normally, but there is a communication bottleneck. In the classic FedAvg algorithm, the client implements collaborative training by transmitting model parameters. Although the algorithm only exchanges model parameters, direct leakage of user data is prevented. However, existing study [5] has shown that this approach does not provide differential privacy protection. Attackers can invert the model to see if specific data exists in the training set of the model. Even worse, if the attacker is a malicious server, the attacker can reconstruct the training data of each participant based on the model gradient uploaded by the client [21]. In this case, how to make the best use of these data resources stored on the client-side while protecting the privacy of the data is the problem addressed in this paper. This optimization problem is considered under an honest but curious condition (also known as a semi-honest model or a passive model), where participants may try to infer the data owned by other participants from a trained model. Nevertheless, privacy is not mined in a malicious way (for example, by attacking the client’s device or using other information). To solve the above problem, our goal is to use GAN and differential privacy to generate “fake” data as valuable as the original training data. These generated data can be published by the clients or server without leaking any private information of the original training data, thereby using data resources and protecting data privacy. For the non-IID problem, we learn from the approach to catastrophic forgetting in continuous learning [22,17]. In order to generate the “fake” data which is difficult to distinguish between true and fake, we use the stable performance WGAN with gradient penalty model [23]. In the training process of GAN, carefully designed noise is added to the gradients to reduce information leakage caused by model parameters and to retain the ideal data utility in the published model. We theoretically prove that the parameters of the generator guarantee differential privacy with respect to the sampled training data, and ensure that any data generated from the generator will not disclose the privacy of the training data, which shows that private FL-GAN can indeed achieve privacy protection. Since the privacy bound produced by the strong composition theorem is often too loose, Abadi et al. [10] developed the moments accountant technique for analyzing differentially private stochastic gradient descent algorithm. In order to obtain tighter privacy bound, we exploit RDP accountant technique [18] in universal private FL-GAN, which uses RDP to measure privacy loss. For a given sampling rate q and noise multiplier σ, through the method of RDP accountant, we can obtain RDP privacy parameters as a function of $\alpha > 1$ for one run of DP-SGD. When DP-SGD is run iteratively, we can compose the Rényi privacy parameter across all runs using Lemma 1. After obtaining an expression for the overall RDP privacy parameter values, any (α, ϵ)-RDP guarantee can be converted into (ϵ, δ)-DP by Lemma 2.

4. Design of private FL-GAN

In this section, we will introduce the design of private FL-GAN in detail. Private FL-GAN is a differentially private generative adversarial network for federated learning, which can mitigate information leakage and retain ideal utility in published models.

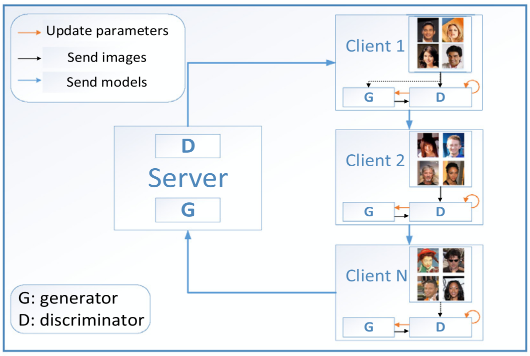


Fig. 1. Algorithm Framework.

4.1. Private FL-GAN framework

Federated learning is a learning model of decentralized data storage. In this mode, the data storage institution is called the client. In order to ensure the privacy of the data, federated learning generally does not directly transfer data and instead transmits model or parameter updates. However, researchers have found that transmission of models and even parameter updates could lead to the disclosure of data privacy [24,5]. Given this, we propose a differentially private generative adversarial network for the “cross-silo” setting of federated learning, namely private FL-GAN. Unlike the classic federated learning algorithm FedAvg [13], where multiple clients train models simultaneously, the model in our algorithm serializes flows among clients. The core idea of FedAvg is that the server averages the parameter updates of multiple clients to get updates for each round. However, when the number of clients is small, access to individual clients is more frequent to train an adequate model, increasing the risk of privacy leakage. Furthermore, in the “cross-silo” setting, communication is likely to be a primary bottleneck, which will cause the model parameters not to be updated in a timely manner. Inspired by [25], we propose that each client sequentially updates the parameters of the same model, training better models with less data access. The framework of our algorithm is shown in Fig. 1. In our setup, each client has a unique dataset. By adding noise to the training process, each client makes its output model satisfy differential privacy, so that the final model obtained by the server can also meet the privacy protection requirements. Therefore, even if the model is made public, the data privacy of the client will not be violated. The following is the algorithm flow:

- The model is initialized by the server, including discriminator and generator. The server sends the model to any client *i*.
- For the current client, in each round of generator training for a total of T_g rounds, the generator is updated after the discriminator has been trained T_d times.
- Randomly select a remaining client, pass the trained model to it, and then return to the second step until all the clients have participated in the training before returning the final model to the server.

4.2. Algorithm detail

```
Algorithm 1: Private FL-GAN
Input: $\alpha_d$, learning rate of discriminator. $\alpha_g$, learning rate of
    generator. $c_g$, bound on the gradients of Wasserstein
    distance with respect to weights. $m$, batch size. $T_d$, number
    of discriminator iterations per generator iteration. $T_g$,
    generator iteration. $\sigma_n$, noise scale. $v$, penalty coefficient.
    $\mathscr{N}$, the set of data-holders (clients).
Output: Differentially private data generation model
 1: Initialize discriminator parameters $\boldsymbol{\omega}$, generator
    parameters $\boldsymbol{\theta}$;
 2: while $\mathscr{N}$ is not empty do
 3:    Take out a client from $\mathscr{N}$
 4:    for $t_1$ = 1, 2, ..., $T_g$ do
 5:       for $t_2$ = 1, 2, ..., $T_d$ do
 6:          for $i$ = 1, 2, ..., m do
 7:             Sample training data $\boldsymbol{x} \sim p_n(\boldsymbol{x})$, latent variable
  $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\beta \sim U[0,1]$;
 8:             $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z}), \hat{\boldsymbol{x}} \leftarrow \beta\boldsymbol{x} + (1-\beta)\tilde{\boldsymbol{x}}$
 9:             $\boldsymbol{L}^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + v(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
 10:             $\boldsymbol{g}^{(i)} \leftarrow clip(\nabla_w \boldsymbol{L}^{(i)}, -c_g, c_g)$
// computing and clipping discriminator's gradients
 11:          $\boldsymbol{\eta} \sim N(0, \sigma_n^2 c_g^2 \boldsymbol{I})$
// generating Gaussion noise
 12:          $\boldsymbol{w} \leftarrow Adam((\frac{1}{m}\Sigma_{i=1}^m \boldsymbol{g}^{(i)} + \boldsymbol{\eta}), \boldsymbol{w}, \alpha_d)$
// updating discriminator
 13:       Sample a batch of latent samples$\{\boldsymbol{z}_i\}_{i=1}^m \sim p(\boldsymbol{z})$
 14:       $\boldsymbol{\theta} \leftarrow Adam(\nabla_\theta \frac{1}{m}\Sigma_{i=1}^m - D_w(G_\theta(\boldsymbol{z})), \boldsymbol{\theta}, \alpha_g)$
// updating generator
 15:       Update privacy accountant with $(T_g, \sigma_n, c_g)$
 16: return $\theta$
```

Next, we describe in detail how private FL-GAN achieves a privacy-preserving goal. We present the detailed algorithm of private FL-GAN in Alg. 1. In general, private FL-GAN is built on the framework of WGAN with gradient penalty. A naïve solution to achieve differential privacy is to inject noise in training both G and D. However, the minimax game formulation makes it difficult to tightly estimate the privacy loss, resulting in excessive degradation in the produced models. Instead, our method is to inject Gaussian noise when updating discriminators. Discriminator parameters can provide differential privacy protection for training data. According to Definition 1, the privacy of data that is not sampled to participate in training is naturally guaranteed, because changes in this part of the data will not have any impact on the output. According to the post-processing property [26] of differential privacy: If any mapping is made to the output satisfying differential privacy, the result still retains the differential privacy property. In our algorithm, the parameters of the discriminator satisfy differential privacy, and the mapping corresponds to the calculation of the generator parameters. Therefore, the parameters of the generator can guarantee differential privacy for the training data. Since the generator parameters guarantee the differential privacy of the data, this also means that even if the observer obtains the generator itself, there is no way to violate the privacy of the training data. Specifically, in the process of training the model on each client, when calculating the discriminator gradients of the training data, we first clip the gradients by c_g (which is bound on the gradients of Wasserstein distance with respect to weights), then Gaussian noise is added to protect the privacy of training data (lines 10, 11 and 12 in Alg. 1). The loss function in line 9 ensures that the gradients update of the discriminator is K-Lipschitz. We set the clipping boundary value to be the same as the Lipschitz condition, so that the clipping operation in line 10 has little impact on the gradients update, and the network convergence will not be greatly affected. The Adam in line 12 and line 14 is an effective stochastic optimization method with high computational efficiency and minimal memory requirements [27]. In dealing with privacy budgets, we use the moments accountant [10] to track cumulative privacy loss in training.

4.3. Theoretical analysis

In private FL-GAN, the key to calculating the accumulated privacy loss is the moments accountant. It calculates the privacy loss of a single iteration of the algorithm, and then iteratively accumulates to calculate the total privacy loss. The moments accountant is based on the assumption that the composition of Gaussian mechanisms is being used. Assessing that a mechanism $\mathscr{M}$ is $(O(\epsilon,\delta))$-differentially private is equivalent to a certain tail bound on $\mathscr{M}$'s privacy loss random variable. The moments accountant keeps track of a bound on the moments of the privacy loss random variable defined as:

$$c(o;\mathscr{M},aux,\mathscr{D}_1,\mathscr{D}_2) = log\frac{Pr[\mathscr{M}(aux,\mathscr{D}_1)=o]}{Pr[\mathscr{M}(aux,\mathscr{D}_2)=o]} \tag{3}$$

Since the approach is the sequential application of the same privacy mechanism, in order to bound this variable, we can define the λ^{th} moment $\alpha_{\mathscr{M}}(\lambda,aux,\mathscr{D}_1,\mathscr{D}_2)$ as the log of the moment generating function evaluated at the value λ:

$$\begin{aligned}\alpha_{\mathscr{M}}(\lambda;\quad & aux,\mathscr{D}_1,\mathscr{D}_2) \triangleq \\ & log\mathbb{E}_{o\sim\mathscr{M}(aux,\mathscr{D}_1)}[exp(\lambda c(o;\mathscr{M},aux,\mathscr{D}_1,\mathscr{D}_2))]\end{aligned} \tag{4}$$

In order to prove privacy guarantees of a mechanism, it is useful to bound all possible $\alpha_{\mathscr{M}}(\lambda;aux,\mathscr{D}_1,\mathscr{D}_2)$. We define

$$\alpha_{\mathscr{M}}(\lambda) \triangleq \max_{aux,\mathscr{D}_1,\mathscr{D}_2} \alpha_{\mathscr{M}}(\lambda;aux,\mathscr{D}_1,\mathscr{D}_2) \tag{5}$$

Lemma 3. *(Composability and Tail Bound [10]). Let $\alpha_{\mathscr{M}}(\lambda)$ be defined as above, then it has the following characteristics:*

1) *given a set of k consecutive mechanisms, for each λ:*

$$\alpha_{\mathscr{M}}(\lambda) \leqslant \sum_i^k \alpha_{\mathscr{M}_i}(\alpha) \tag{6}$$

2) *for any $\epsilon > 0$, the mechanism $\mathscr{M}$ is (ϵ,δ)-differentially private for:*

$$\delta = \min_\lambda exp(\alpha_{\mathscr{M}}(\lambda) - \lambda\epsilon) \tag{7}$$

Lemma 4. *(Parallel Composition [28]). For disjoint subsets $\mathscr{D}_i \subseteq \mathscr{D}$, let mechanism $\mathscr{M}(\mathscr{D}_i)$ satisfy ϵ-differential privacy. The sequence of $\mathscr{M}(\mathscr{D}_i)$ still satisfies ϵ-differential privacy.*

Theorem 1. *The generator learned in Alg. 1 guarantees (ϵ,δ)-differential privacy for the appropriately chosen setting of the noise scale σ and the clipping threshold c_g.*

Proof. When the chosen parameters σ and λ satisfy the conditions in Lemma 3 of [10], the log moment of discriminator in Alg. 1 can be bounded as $\alpha(\lambda) \leqslant Tq^2\lambda^2/\sigma^2$, where T is the total number of discriminator iterations and q is the sampling ratio per iteration. When $Tq^2\lambda^2/\sigma^2 \leqslant \lambda\epsilon/2$ and $exp(-\lambda\epsilon/2) \leqslant \delta$, according to Lemma 3, the discriminator trained in each client satisfies (ϵ, δ)-differential privacy. In addition, the generator does not have direct access to the training data. With the effect of the post-processing property of differential privacy [26], the generator also satisfies (ϵ, δ)-differential privacy. Due to Lemma 4, when the model is passed between the clients, the datasets of the clients are disjoint, which does not increase the risk of data privacy exposure in other clients. Therefore, the final model derived from all clients' training is (ϵ, δ)-differential privacy. □

5. Universal private FL-GAN

Our private FL-GAN implements differential privacy synthetic data generation on the "cross-silo" setting of federated learning. When communication between the clients is the main bottleneck of federated learning, we adopt a serialized model training method to complete the training of the entire generated model locally on the client. In order to protect the privacy of the training data, we add carefully designed Gaussian noise to the gradients update process of the discriminator. Thanks to the excellent property of differential privacy, the resulting model can not only generate high-quality synthetic data, but also provide adequate protection for privacy under a reasonable privacy budget. However, the situation we considered is idealized, and we supposed that all clients' data is independent and identically distributed (IID). However, a more realistic scenario is that the data owned by the clients are often not independent and identically distributed (non-IID). Therefore, in order to make our work more practical, we extend the algorithm to non-IID datasets. There are deep parallels between the serialized federated learning problem and another fundamental machine learning problem known as lifelong learning. The challenge of lifelong learning [29] is to continue learning task B using the same model after learning task A, but there is no "forgetting". This problem (called "catastrophic forgetting") [30] is very similar to the non-IID problem in the serialized federated learning "cross-silo" setting that we would like to solve. Therefore, we consider the method of lifelong learning to approach non-IID problems.

5.1. Framework of universal private FL-GAN

The universal private FL-GAN keeps the framework of the previous algorithm unchanged, and our object is still the "cross-silo" setting of federated learning. However, the new algorithm has some changes in the process. The following is the algorithm flow of the universal private FL-GAN:

- The model is initialized by the server, including discriminator and generator. The server sends the model to any client *i*.
- Use the received generative model to generate synthetic data and add it to the local training set, and then start training. If the received model is the initial model from the server, there is no need to generate synthetic data but start training directly.
- In each round of generator training for a total of T_g rounds, the generator is updated after the discriminator has been trained T_d times.
- Randomly select a remaining client, pass the trained model to it, and then return to the second step until all the clients have participated in the training before returning the final model to the server.

Compared to the previous version, the universal private FL-GAN uses a replay mechanism to solve the non-IID problem. Because GAN can generate high-quality pictures to supplement the training set, the non-IID problem of the data is addressed by enriching the diversity of the data. In addition, we apply the elastic weight consolidation (EWC) regularization [16] in the discriminator training process to prevent forgetting in GAN, which allows us to use a small amount of synthetic data (step two) to overcome the non-IID problem.

5.2. Algorithm detail of the universal private FL-GAN

Algorithm 2: Universal Private FL-GAN

```
Input: $\alpha_d$, learning rate of discriminator. $\alpha_g$, learning rate of
  generator. $c_g$, bound on the gradients of Wasserstein
  distance with respect to weights. $m$, batch size. $T_d$, number
  of discriminator iterations per generator iteration. $T_g$,
  generator iteration. $\sigma_n$, noise scale. $v$, penalty coefficient.
  $\mathcal{N}$, the set of data-holders (clients). $D_t$, the training dataset.
  $D_g$, the generating dataset. $\mathcal{F}$, Fisher information matrix. $\mu$,
  importance of the old model.
Output: Differentially private data generation model
 1: Initialize discriminator $\boldsymbol{\omega}_0$, generator $\boldsymbol{\theta}_0$, $\mathcal{F}_0 = 0$, and
    $k = 0$;
 2: while $\mathcal{N}$ is not empty do
 3:   $k = k + 1$
 4:   Take out a client $C_k$ from $\mathcal{N}$, then pass $\boldsymbol{\omega}_{k-1}$, $\mathcal{F}_{k-1}$ and
    $\boldsymbol{\theta}_{k-1}$ to it.
 5:   if The received model $\boldsymbol{\theta}_{k-1}$ is not $\boldsymbol{\theta}_0$ then
 6:     $D_g$ ←Generating data with $\boldsymbol{\theta}_{k-1}$
 7:     $D_t \leftarrow D_t \cup D_g$
      //using existing generators to enrich training data
 8:   for $t_1$ = 1, 2, ..., $T_g$ do
 9:     for $t_2$ = 1, 2, ..., $T_d$ do
10:       for $i$ = 1, 2, ..., m do
11:         Sample training data $\boldsymbol{x} \sim p_n(\boldsymbol{x})$, latent variable
    $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\beta \sim U[0, 1]$;
12:         $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z}), \hat{\boldsymbol{x}} \leftarrow \beta\boldsymbol{x} + (1-\beta)\tilde{\boldsymbol{x}}$
13:         $\boldsymbol{L}^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + v(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
14:         $\boldsymbol{L}^{(i)} \leftarrow \boldsymbol{L}^{(i)} + \mu(\boldsymbol{\omega}_k - \boldsymbol{\omega}_{k-1})^2 \mathcal{F}_{k-1}$
15:         $\boldsymbol{g}^{(i)} \leftarrow clip(\nabla_w \boldsymbol{L}^{(i)}, -c_g, c_g)$
          // computing and clipping discriminator's gradients
16:       $\boldsymbol{\eta} \sim N(0, \sigma_n^2 c_g^2 \boldsymbol{I})$
        // generating Gaussion noise
17:       $\boldsymbol{w} \leftarrow Adam((\frac{1}{m}\Sigma_{i=1}^m \boldsymbol{g}^{(i)} + \boldsymbol{\eta}), \boldsymbol{w}, \alpha_d)$
        // updating discriminator
18:     Sample a batch of latent samples$\{\boldsymbol{z}_i\}_{i=1}^m \sim p(\boldsymbol{z})$
19:     $\boldsymbol{\theta} \leftarrow Adam(\nabla_\theta \frac{1}{m}\Sigma_{i=1}^m - D_w(G_\theta(\boldsymbol{z})), \boldsymbol{\theta}, \alpha_g)$
      // updating generator
20:     Update privacy accountant with $(T_g, \sigma_n, c_g)$
21:   $\mathcal{F}_k$ ← Calculate the Fisher information matrix of $\boldsymbol{\omega}_k$.
22: return $\boldsymbol{\theta}$
```

The universal private FL-GAN differs slightly from the previous version in the execution steps of the algorithm. See Alg. 2 for details of the algorithm. Before the client starts training the model, it needs to check whether the model it received is the initial model given by the server (line 5 in Alg. 2). If not, use the received generator to generate a small amount of fake data containing the data characteristics of the previous client, and add it to the local training set to alleviate the problem of imbalanced data distribution caused by non-IID (lines 6 and 7). Otherwise, directly enter the training phase of the model. In order to further approach the non-IID challenge of the data, we make some modifications to the objective function of WGAN-GP. After training the model in each client, we calculate the Fisher information matrix $\mathscr{F}$ corresponding to the discriminator model parameters at this moment (line 21). $\mathscr{F}$ reflects the influence of the previous training data distribution on the model parameters, and to ensure privacy, we treat the information matrix with the same privacy parameters (bound on weights and noise scale) for privacy. In the process of model training, when the client calculates the loss function of the discriminator, it adds an additional regularization term related to $\mathscr{F}$ to prevent the current model parameter from "forgetting" the distribution of the client data before, and μ is a parameter that controls the degree of influence of the previous model on the current model (line 14).

5.3. Theoretical analysis

When tracking accumulated privacy loss, we replace the moments accountant with RDP accountant [18]. RDP, as a variant of the standard DP, uses Rényi divergence to measure the difference between two distributions. Rényi divergence is more suitable for strictly capturing privacy losses under the Gaussian mechanism [31], which provides stricter constraints for the calculation of privacy loss.

Theorem 2. *The generator learned in Alg. 2 guarantees* (ϵ, δ)*-differential privacy for the appropriately chosen setting of the noise scale* σ *and the clipping threshold* c_g.

Proof. When the client trains the model, Gaussian noise with scale σ is added to the gradient of the discriminator. According to Definition 2, the discriminator satisfies (α, ϵ_i)-RDP, where ϵ_i is calculated by RDP accountant with respect to Corollary 3 of [18]. With the post-processing property of differential privacy [26], the generator also satisfies (α, ϵ_i)-RDP. Then, Lemma 1 is used to linearly accumulate the RDP privacy parameters with respect to each round of iterations to obtain a cumulative privacy loss. Finally, by Lemma 2, the (α, ϵ)-RDP guarantee is converted to the standard (ϵ, δ)-DP, that is, our algorithm satisfies (ϵ, δ)-differential privacy. In addition, the models received from the previous client meet the differential privacy protection. Therefore, using them to generate training data (lines 6 and 7 in Alg. 2) does not increase the risk of privacy leakage. When the Fisher information matrix of discriminator parameters is computed with a small amount of data required, we compute the privacy loss by adding noise to the computation and accumulate it to the loss from the previous model training. Therefore, each step of the algorithm satisfies differential privacy. □

6. Experimental evaluation

This section presents extensive experiments to evaluate our proposed private FL-GAN and its extended version, i.e., universal private FL-GAN. The experiments are conducted on three benchmark datasets (MNIST,[1] CelebA[2] and Fashion-MNIST[3]). The experiments revolve around the quality of the data generated, the ability to generate data when the data distribution is non-IID, and the ability to protect privacy.[3]

6.1. Experimental setting

Our experiments rely on the following three datasets:

1) MNIST, which consists of 70 K handwritten digital images of size 28 × 28, split into 60 K training and 10 K test samples.
2) CelebA, which is a large-scale face attributes dataset with more than 200 K celebrity images of size 64 × 64, each with 40 attribute annotations.
3) Fashion-MNIST, which contains a total of 70 K images of the product in ten categories, each of which is 28 × 28 in size.

In the experiments, we set the learning rate of the first client discriminator and generator to 1e−4, and the other clients to 5e−5. The clip norm bound c_g is set to 1, the same as the Lipschitz condition. The batch size is 64. Parameter δ is set to the inverse of the number of samples. The number of iterations on discriminator T_d is 5, and the iteration parameter of generator T_g depends on the privacy budget ϵ. The dimension of z is 100, and every coordinate is within [-1, 1]. The penalty coefficient v is 10. The noise scale σ is generally around 1. Similar to private FL-GAN, universal private FL-GAN takes the same value for the common parameters. Furthermore, the importance of the old model μ is 50. Aiming to obtain good statistical properties, we choose Gaussian noise with zero mean and multiple values of standard deviation. In order for each client to have a certain amount of training data, we split datasets into several parts, so we can simulate multiple clients.

6.2. Generation performance of private FL-GAN

In order to explore the impact of the specific value of the privacy level on the quality of the images, we conducted several experiments on MNIST and CelebA datasets. According to the selection of privacy budgets in GANobfuscator [12], we selected some relatively large range (from 0.8 to 9) in our experiments to evaluate the performance of private FL-GAN. In these experiments, we trained by setting different privacy parameters ϵ and obtained several models of privacy protection levels. The generated images are shown in Fig. 2 and Fig. 3, with respect to different levels of privacy parameters.

The larger the privacy budget of the model (i.e., less noise is added), the higher the quality of the corresponding generated picture, indicating that the distortion of the image is caused by noise, not the non-convergence of the models or bad training images. In Fig. 2, we can see that private FL-GAN can generate all numbers from 0 to 9, and the same numbers will have many different styles. Similarly, in Fig. 3, the same gender has different hair colors, expressions, etc. It shows that our model does not simply remember training samples, but can indeed generate samples with unique details. In order to guarantee the good quality of the generated images, it is recommended to set the noise level to a slightly large value. Later, we will prove that the noise of this level will not significantly increase the risk of privacy leakage for existing attack methods through membership inference attacks. Inheriting the advantages of the WGAN with gradient penalty network structure, our model does not suffer from mode collapse or gradient vanishing. Next, we used two specific metrics to quantitatively measure the quality of the data generated by private FL-GAN. For comparison, we conducted the same experiment with no privacy protection (FL-GAN), and DP-FedAvg-GAN [15]. Since DP-FedAvg-GAN is targeted at a massive number of edge devices, while our algorithm targets a smaller number of data centers with more concentrated data, we adapt DP-FedAvg-GAN so that each iteration will involve all data centers, rather than just a subset as in the original scenario. In addition, DP-FedAvg-GAN is an algorithmic framework based on DP-FedAvg [32], which requires the central server to be trusted. In the framework, each client computes updates of the model and uploads them directly to the server, which then aggregates the updates and adds differential privacy protection. DP-

[1] MNIST: http://yann.lecun.com/exdb/mnist/
[2] CelebA: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.
[3] Fashion-MNIST: https://github.com/zalandoresearch/fashion-mnist.

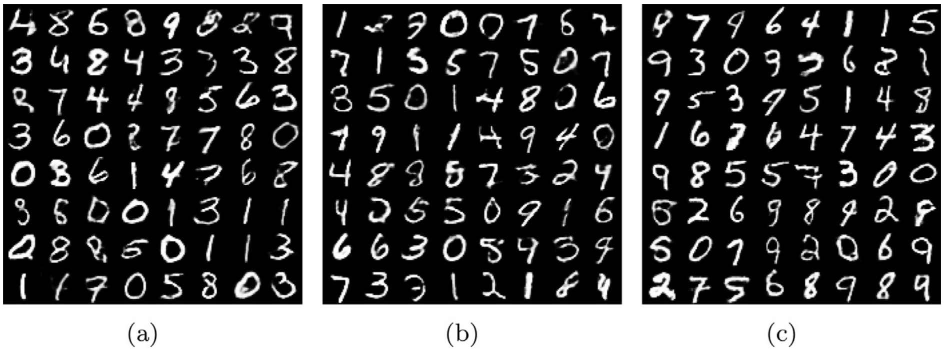

(a) (b) (c)

Fig. 2. Synthetic data with three different ϵ on MNIST dataset. (a) ϵ = 4. (b) ϵ = 6. (c) ϵ = 8.

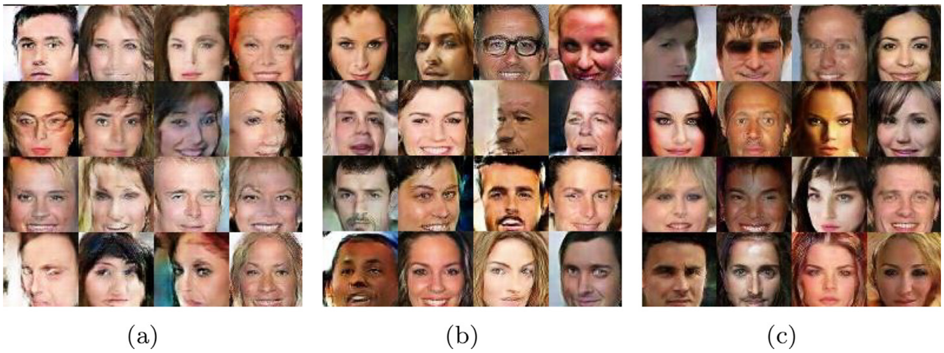

(a) (b) (c)

Fig. 3. Synthetic data with three different ϵ on CelebA dataset. (a) ϵ = 4. (b) ϵ = 6. (c) ϵ = 8.

FedAvg-GAN belongs to centralized differential privacy, while our algorithm belongs to local differential privacy. All privacy operations in our algorithm are implemented locally without any security assumptions on the server, which is more secure than DP-FedAvg-GAN. Furthermore, DP-FedAvg-GAN requires multiple rounds of client–server interaction to achieve model convergence, which introduces more communication overhead. In order to simulate a different number of client scenarios, we split the datasets into different portions. After the three models have been trained for the same number of rounds, we calculate the Inception Score (IS) [33] of synthetic data. Proposed by Salimans et al., IS is perhaps the most widely adopted score for GAN evaluation. It uses a pre-trained neural network to capture the desirable properties of generated samples. The IS shows a reasonable correlation with the quality and diversity of generated images. For this evaluation index, the higher the score, the higher the quality of the generated image, and the greater the diversity. It can be seen from Fig. 4 that the score of the images generated by private FL-GAN is always better than that of DP-FedAvg-GAN, and it is pretty close to the score of the images generated by the algorithm without privacy protection. To make the results more convincing, we also adopted another method, Fréchet Inception Distance (FID) [34], to evaluate the generated images. Introduced by Heusel et al., FID embeds a set of generated samples into a feature space given by a specific layer of Inception Net. Viewing the embedding layer as a continuous multivariate Gaussian, the mean and covariance are estimated for both the generated data and the real data. Lower FID means smaller distances between synthetic and real data distributions. In other words, contrary to the IS, under the Fréchet Inception Distance standard, a lower FID represents a higher quality of synthetic data. The FID of the real data and synthetic data on the CelebA are shown in Fig. 5. It can be seen that the FIDs between real images and the images generated by private FL-GAN are always low. And they are close to the FIDs between real images and the images generated by FL-GAN. However, the FIDs of images generated by DP-FedAvg-GAN are relatively higher. Combining Figs. 4 and 5, we can see that the quality of the data generated by private FL-GAN is superior to DP-FedAvg-GAN and is fairly close to FL-GAN without privacy protection.

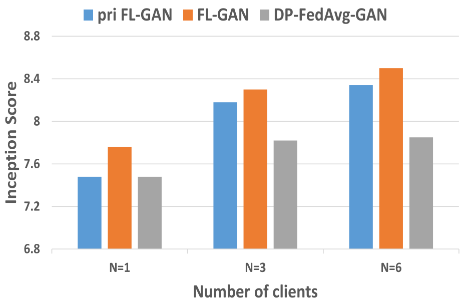


Fig. 4. Comparison of IS on MNIST dataset.

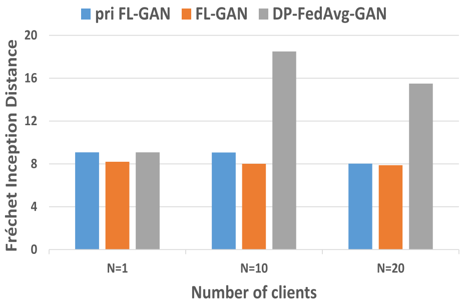


Fig. 5. Comparison of FID on CelebA dataset.

6.3. Regarding non-IID

As a fundamental challenge in federated learning, the problem of non-IID data has been attracting many researchers. While the meaning of IID is generally straightforward, data can be non-IID in many ways. In this paper, we focus on the case where the data type is the same, but the label distribution is different [14]. We conducted experiments on MNIST and Fashion-MNIST datasets. Assuming there are 6 clients, each with a privacy budget $\epsilon = 8$. To get closer to reality, we determine the data that each client has in random selection. Specifically, MNIST has a total of ten category labels with numbers 0 to 9. We first determine how many categories of numbers each client has by generating random numbers. Then, for each client, generate a round of random numbers to determine which categories of numbers the client has. Similarly, Fashion-MNIST owns a total of ten product categories, including T-shirts, trousers, shoes, and so on. We also generated random numbers to determine the number of categories each client has and which categories of data it has.

For the MNIST dataset, the numbers of labels owned by the 6 clients are [7,3,8,4,8,6], and the categories owned by client 1 are [0,1,3,5,6,7,8]. From Fig. 6(a) we can see, the generator well-trained by client 1 can generate every category of number in the training data. Fig. 6(b) shows the generation results of client 2. The training data of client 2 only has numbers 2, 5, and 9, total three categories of numbers, but the generator trained by client 2 can generate the numbers owned by clients 1 and 2. With the powerful ability of GAN to generate samples, supplemented by EWC [16] to constrain the model parameters, the clients can overcome the non-IID. The images generated by the final model are shown in Fig. 6(c), through which we can see the model can generate numbers for all categories and overcome the non-IID. In order to provide an intuitive comparison display of the generated results, we put the results generated by the model trained on the IID datasets in Fig. 6(d). At the same time, we consider using only one of Memory Replay and EWC technology for the experiment. The results are: The model using only Memory Replay technology generates images with poor detail and diversity; the model using only EWC constraint can only generate the types of numbers contained in the last client training data. Due to space limitations, they are not shown in the paper.

For the Fashion-MNIST dataset, the numbers of labels owned by the six clients are [5,3,7,4,6,5]. For the sake of brevity, we represent the ten categories of the product by the numbers of 0 to 9, respectively. Because pullovers, coats and shirts are not easy to distinguish visually, we removed the coats and shirts represented by the numbers 4 and 6 in the training set. Client 1 has 6 categories of product pictures labeled [2,3,5,7,9]. Fig. 7(a) shows the generated results of the model trained on client 1, and it can be seen that the model can generate all 5 kinds of products in the training set. Fig. 7(b) shows the generated results of training on client 2 with only 3 kinds of products as the training set. It can be seen that the model can generate more than 3 kinds of products and can generate the product that exists in the training set of client 1. From the final model generation results (Fig. 7(c)), we can see that the final model can generate a total of 8 kinds of product pictures contained in all clients, demonstrating that universal private FL-GAN can generate high-quality pictures while providing privacy protection, and can overcome non-IID difficulties. Similarly, we show the generated results on IID data in Fig. 7(d).

6.4. Privacy protection performance

To explore the privacy protection performance of universal private FL-GAN, we use membership inference attack [5] to measure the privacy risk that data faces when used to train the model. Given a model and a record, when the adversary knows whether the record is used for model training, it is regarded as information leakage. We conducted experiments on FL-GAN, DP-FedAvg-GAN and universal private FL-GAN to carry out a membership inference attack on the MNIST dataset. When implementing a membership inference attack, we use the same model as the discriminator as the shadow model, and then we construct a fully connected network with five hidden layers as our attack model, using "relu"

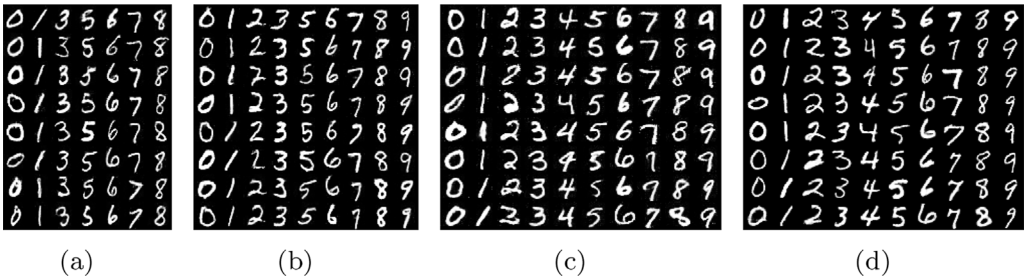


Fig. 6. Synthetic results on MNIST dataset. (a) Client 1 has 7 types of numbers. (b) Client 2 has 3 types of numbers. (c) The last client has 6 types of numbers. (d) Synthetic results on IID MNIST dataset.

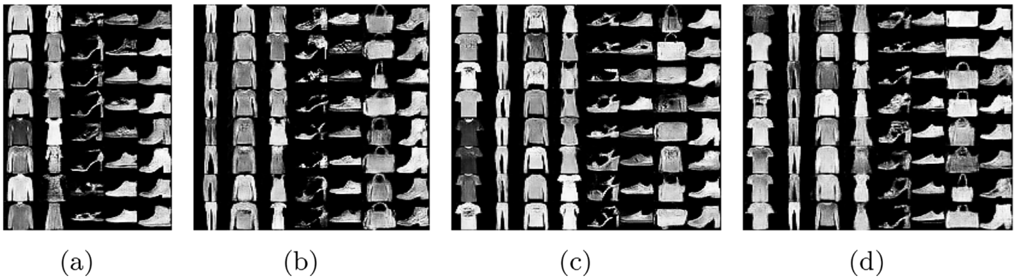


Fig. 7. Synthetic results on Fashion-MNIST dataset. (a) Client 1 has 5 types of products. (b) Client 2 has 3 types of products. (c) The last client has 6 types of products. (d) Synthetic results on IID Fashion-MNIST dataset.

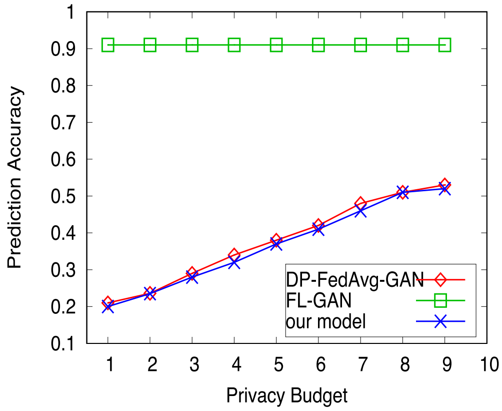


Fig. 8. The precision of membership inference attack.

and "sigmoid" as activation functions, respectively. We evaluated the privacy protection effects of the three models under different privacy budgets. As can be seen in Fig. 8, the accuracy of the attack model is high (91%) on the model with no privacy protection applied, and universal private FL-GAN and DP-FedAvg-GAN with differential privacy technology, are well resistant to membership inference attack. As the privacy budget increases, the accuracy of the attack gradually improves, but even if the privacy budget reaches a large value, the attack can only have a 50% accuracy rate, which is equivalent to guessing without any background knowledge. Thus, our universal private FL-GAN is well resistant to membership inference attacks, which is in line with the expectation of protecting the privacy of training data.

7. Conclusion

In this paper, we studied the problem of training differentially private GAN in the "cross-silo" setting of federated learning, where the data is stored in a distributed manner. The private FL-GAN we proposed organically combines differential privacy sensitivity and Lipschitz condition to more accurately measure privacy loss and improve model convergence speed. We have theoretically proved that private FL-GAN can rigorously guarantee (ϵ, δ)-differential privacy. Furthermore, we also extended private FL-GAN to non-IID data and proposed universal private FL-GAN. Through experiments, we demonstrated that both private FL-GAN and universal private FL-GAN could generate high-quality data under a reasonable privacy budget.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Funding: This work was supported by the National Natural Science Foundation of China (No. 62172385), the Anhui Initiative in Quantum Information Technologies (No. AHY150300), the Director of Computer Application Research Institute Foundation (No. SJ2020A08) and China Academy of Engineering Physics Innovation and Development Fund Cultivation Project (No. PY20210160).

References

[1] C. Ruffino, R. Hérault, E. Laloy, G. Gasso, Pixel-wise conditioned generative adversarial networks for image synthesis and completion, Neurocomputing..
[2] Y. Ding, Y. Zhu, J. Feng, P. Zhang, Z. Cheng, Interpretable spatio-temporal attention lstm model for flood forecasting, Neurocomputing 403 (2020) 348–359.
[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680..
[4] Y. Chen, S. Xia, J. Zhao, M. Jian, Y. Zhou, Q. Niu, R. Yao, D. Zhu, Person image synthesis through siamese generative adversarial network, Neurocomputing 417 (2020) 490–500.
[5] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 3–18..
[6] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the gan: information leakage from collaborative deep learning, in, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 603–618.
[7] C. Dwork, Differential privacy: A survey of results, in: International conference on theory and applications of models of computation, Springer, 2008, pp. 1–19..
[8] R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1310–1321..
[9] Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, H. Zhu, Differentially private data generative models (2018). arXiv:1812.02274..
[10] M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, L. Zhang, Deep learning with differential privacy, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 308–318.
[11] L. Xie, K. Lin, S. Wang, F. Wang, J. Zhou, Differentially private generative adversarial network, arXiv preprint arXiv:1802.06739..
[12] C. Xu, J. Ren, D. Zhang, Y. Zhang, Z. Qin, K. Ren, Ganobfuscator: Mitigating information leakage under gan via differential privacy, IEEE Trans. Inf. Forensics Secur. 14 (9) (2019) 2358–2371.
[13] H.B. McMahan, E. Moore, D. Ramage, S. Hampson, et al., Communication-efficient learning of deep networks from decentralized data, arXiv preprint arXiv:1602.05629..
[14] P. Kairouz, H.B. McMahan, B. Avent, A. Bellet, M. Bennis, A.N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, et al., Advances and open problems in federated learning, arXiv preprint arXiv:1912.04977..
[15] S. Augenstein, H.B. McMahan, D. Ramage, S. Ramaswamy, P. Kairouz, M. Chen, R. Mathews, B.A. y Arcas, Generative models for effective ml on private, decentralized datasets (2019). arXiv:1911.06679..
[16] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.
[17] C. Wu, L. Herranz, X. Liu, J. van de Weijer, B. Raducanu, et al., Memory replay gans: Learning to generate new categories without forgetting, Advances in Neural Information Processing Systems (2018) 5962–5972.
[18] I. Mironov, Rényi differential privacy, in: 2017 IEEE 30th Computer Security Foundations Symposium (CSF), IEEE, 2017, pp. 263–275..
[19] B. Xin, W. Yang, Y. Geng, S. Chen, S. Wang, L. Huang, Private fl-gan: Differential privacy synthetic data generation based on federated learning, in: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 2927–2931.
[20] C. Dwork, G.N. Rothblum, Concentrated differential privacy (2016). arXiv:1603.01887..
[21] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, Adv. Neural Inf. Process. Syst. 32 (2019) 14774–14784.
[22] A. Seff, A. Beatson, D. Suo, H. Liu, Continual learning in generative adversarial nets, arXiv preprint arXiv:1705.08395..
[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of wasserstein gans, in: Advances in neural information processing systems, 2017, pp. 5767–5777..
[24] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, in: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015, pp. 1322–1333.
[25] C. Hardy, E. Le Merrer, B. Sericola, Md-gan: Multi-discriminator generative adversarial networks for distributed datasets, in: 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2019, pp. 866–877..
[26] C. Dwork, A. Roth, et al., The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.
[27] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980..
[28] F.D. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of data, 2009, pp. 19–30.
[29] J. Ramapuram, M. Gregorova, A. Kalousis, Lifelong generative modeling, Neurocomputing..
[30] R.M. French, Catastrophic forgetting in connectionist networks, Trends Cogn. Sci. 3 (4) (1999) 128–135.
[31] U. Tantipongpipat, C. Waites, D. Boob, A.A. Siva, R. Cummings, Differentially private mixed-type data generation for unsupervised learning (2019). arXiv:1912.03250..

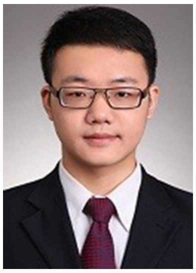
[32] R.C. Geyer, T. Klein, M. Nabi, Differentially private federated learning: A client level perspective, arXiv preprint arXiv:1712.07557..

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in neural information processing systems, 2016, pp. 2234–2242..

[34] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Advances in Neural Information Processing Systems (2017) 6626–6637.

Bangzhou Xin is a Ph.D. candidate in School of CyberScience, University of Science and Technology of China. His research interests include data privacy and machine learning.

Yangyang Geng is a Ph.D. candidate in School of Computer Science and Technology, University of Science and Technology of China. His research interests include data privacy and cloud computing.



Teng Hu received his B.S. degree from Sichuan University in 2011, M.S. degree from Beijing University of Posts and Telecommunications in 2014, and Ph.D. from the University of Electronic Science and Technology of China in 2021. He has been working at the Institute of Computer Application, China Academy of Engineering Physics since 2014. His research interests are cybersecurity-related issues in artificial intelligence, blockchain, and big data technologies.

Sheng Chen received the BS degree from Sichuan University, China, in 2013. He is currently working toward the Ph.D. degree in the Department of Computer Science and Technology, University of Science and Technology of China. His research interests include ubiquitous computing, security of IoT and Deep Learning.

Wei Yang is an associate professor in School of Computer Science and Technology at the University of Science and Technology of China. He is also a visiting professor at the National University of Singapore. In 2007, he received his Ph.D. degree in computer science from University of Science and Technology of China and was awarded the Dean's Prize of Chinese Academy of Sciences. His research interests include information security, quantum information and computer vision. He has authored or co-authored over 150 technical papers in major international journals and conferences. In 2014, he got the Natural Science Award of Ministry of Education of People's Republic of China. In 2016, he won the Best Paper Award at ACM UbiComp. Recently, he won the championship on the CVPR 2020 KITTI-MOTS Challenge.

Shaowei Wang is an associate professor in Institute of Artificial Intelligence and Blockchain at Guangzhou University. He received PhD degree in the School of Computer Science and Technology, University of Science and Technology of China (USTC). His research interests are data privacy, federated learning and recommendation systems.

Liusheng Huang is a professor in School of Computer Science and Technology at the University of Science and Technology of China. His research interests include Internet of Things, cloud computing, and data privacy.