PPGAN: Privacy-preserving Generative Adversarial Network

1st Yi Liu (S'19)
School of Data Science and Technology
Heilongjiang University
Harbin, China
97liuyi@ieee.org

2nd Jialiang Peng*
School of Data Science and Technology
Heilongjiang University
Harbin, China
Pengjialiang@hlju.edu.cn
*Corresponding Author

3rd James J.Q. Yu (S'11-M'15)
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
yujq3@sustech.edu.cn

4th Yi Wu
School of Data Science and Technology
Heilongjiang University
Harbin, China
1995050@hlju.edu.cn

***Abstract*—Generative Adversarial Network (GAN) and its variants serve as a perfect representation of the data generation model, providing researchers with a large amount of high-quality generated data. They illustrate a promising direction for research with limited data availability. When GAN learns the semantic-rich data distribution from a dataset, the density of the generated distribution tends to concentrate on the training data. Due to the gradient parameters of the deep neural network contain the data distribution of the training samples, they can easily *remember* the training samples. When GAN is applied to private or sensitive data, for instance, patient medical records, as private information may be leakage. To address this issue, we propose a *Privacy-preserving Generative Adversarial Network (PPGAN)* model, in which we achieve differential privacy in GANs by adding well-designed noise to the gradient during the model learning procedure. Besides, we introduced the *Moments Accountant* strategy in the PPGAN training process to improve the stability and compatibility of the model by controlling privacy loss. We also give a mathematical proof of the differential privacy discriminator. Through extensive case studies of the benchmark datasets, we demonstrate that PPGAN can generate high-quality synthetic data while retaining the required data available under a reasonable privacy budget.**

***Index Terms*—Privacy leakage, GAN, deep learning, differential privacy, moments accountant.**

I. INTRODUCTION

In recent years, researchers have used a large number of training data to perform data mining tasks, in the field of medical and health informatics, such as disease prediction and auxiliary diagnosis. Deep learning models are employed to remember the characteristics of a large number of training samples for classification or prediction purposes. However, organizations such as hospitals and research institutes are paying more and more attention to the protection of data [1]. Additionally, the *General Data Protection Regulation (GDPR)*[2] issued by the European Union prohibits organizations from sharing private data. It is increasingly difficult for researchers to obtain training data unlimited legally.

Fortunately, the generative model provides us with a solution to the issue of data scarcity [3], yet data privacy leakage issues may arise. StyleGAN [4] shown impressive performance in generating fake face images. In principle, it can memorize data distribution from the small amount of training data, rendering indistinguishable high-quality "fake" samples. However, for most people, they expect their face data not to be used as a training sample.

GAN can implicitly disclose the privacy information of training samples. GAN model produces high-quality "fake" samples through continuous training and resampling. This training method grants hackers the opportunity to restore the original samples. In the literature, Hitaj et al. [5] proposed an attack model based on the distribution of training data to reconstruct a real sample from synthesized data. This hinders these models from learning trained sufficiently, leading to inferior performance. Therefore, we not only need high-quality sample generation approaches but also need to achieve a reasonable level of data privacy.

Based on the above findings, we propose a **Privacy-preserving GAN (PPGAN)**. PPGAN combine with differential privacy [6] to ensure that the exact training samples can not be revealed by adversaries from the trained model, resulting in well-protected data privacy. In particular, we added well-designed noise to the gradients in the training process in PPGAN and used the framework of the WGAN [7] model as the main skeleton of PPGAN. The proposed model does not suffer from privacy leakage issue whose proportional to the volume of data thanks to the introduced average aggregator that offsets the privacy overhead of large datasets. We evaluate PPGAN in MNIST and Electronic Health Records (EHRs)

This work is supported in part by the Nature Science Foundation of Heilongjiang Province of China (Grant NO.F2016035 and NO.QC2016091), Ministry of Education of China and the School of Entrepreneurship Education of Heilongjiang University (Grant NO.201910212133).

datasets and demonstrate that PPGAN can generate high-quality synthesized data while providing adequate protection via differential privacy with a reasonable budget of privacy.

We would like to point out our main contributions as follows:

- We propose a new Privacy-preserving GAN (PPGAN) model that can generate high-quality data points while protecting data privacy. PPGAN combines noise well-designed in the differential privacy with training gradients to disturb the distribution of the original data. Finally, we give a rigorous proof of the differential privacy discriminator in mathematics.
- We introduced the *Moments Accountant* strategy that maintains the boundedness of the function, controls the privacy level and significantly improves the stability of the model training.
- We evaluated PPGAN with benchmark datasets. The results show that PPGAN can generate high-quality data with adequately protected privacy under a reasonable privacy budget.

The overall structure of this paper is as follows. First, we briefly summarize the relevant literature in Section II and then introduce the proposed PPGAN framework and its theoretical proof in Section III. We assess the performance of our framework in Section IV. Finally, this paper is concluded in Section V.

II. RELATED WORK

In this section, we focus on the literature on privacy-preserving deep learning. Existing literature can be roughly classified along several axes: generative adversarial networks in the medical field, differential privacy, and deep learning with differential privacy.

Generative Adversarial Network. In recent years, GAN and its variants have made meaningful progress in the academic and medical fields. Choi et al. [8] proposed medGAN, which is a generative adversarial network for generating multi-label discrete patient records. Arnab Kumar Mondal et al.[9] solved the problem of segmenting 3D multimodal medical images with a few examples of maker are available for training. A new method based on the generative adversarial network was proposed to train a segmentation model with both labeled and unlabeled images [10]. The presented method prevented overfitting by learning to distinguish between true and false patches obtained by the generator network. Qi et al. [11] presented the Lipschitz regularization theory and algorithms for a novel Loss-Sensitive Generative Adversarial Network (LS-GAN). This model trains the loss function by identifying the real and "false" samples of the marginal region, using the idea of game theory to cause the generator to generate the most realistic samples. LS-GAN performs very well in medical image classification tasks. Brett K. Beaulieu-Jones et al. [12] proposed AC-GAN (under differential privacy and labeled private) to simulate participants in the *SPRINT* clinical trial. However, the previously described GANs do not meet the data management requirements of *GDPR* for privacy data protection.

Differential Privacy. Differential privacy (DP), local differential privacy (LDP), and other related algorithms combined with deep neural networks have become one of the most popular algorithmic models in the field of privacy protection. Dwork et al. [5], the author of the concept of differential privacy, laid a lot of theoretical foundations for the field of differential privacy. Song et al. [13] added perturbations to random descent gradients, which can improve network performance after batch training. Many machine learning algorithms can achieve differential private by introducing randomization in the calculation, usually by noise [13]. Kamalika Chaudhuri et al. [14] proposed a basic framework for differential privacy, a key mechanism for ensuring privacy, and how to find a private differential approximation of several contemporary machine learning tools.

Privacy-Preserving Deep Learning. Recently, the application of differential privacy in deep learning has been studied in several papers: Abadi et al. [15] designed a deep learning model based on the differential privacy framework to perform a detailed analysis of privacy costs in the model. Reza Shokri et al. [16] designed a privacy-preserving deep learning system that does not require the sharing of input datasets and can perform accurate in-depth neural network predictions under secure multi-party computing. Nicolas Papernot et al. [17] proposed that Private Aggregation of Teacher Ensembles (PATE) provide strong privacy protection for training data. The method combines multiple models trained using disjoint datasets in a black-box manner, such as records from different subsets of users.

We propose *PPGAN* to address the challenges that appeared in the previous works. In [16], although the privacy-preserving deep learning system does not need to share datasets, it still reveals the user's privacy when uploading local parameters to the server. What is different from [16] is that we add well-designed noise during the process of stochastic gradient descent. In [15], the privacy overhead of the deep learning model based on the differential privacy framework is directly proportional to the number of datasets, which will significantly reduce the accuracy of the model. We solve this problem by training the differential privacy generator through the differentially private discriminator model *(DPDM)* and can generate differential privacy high-quality data points. We introduced a moments accountant strategy, which not only successfully incorporated the privacy enhancement mechanism into the training depth generation model but also significantly improved the stability and scalability of the generation model training itself.

III. METHODOLOGY

In this section, we elaborate on the proposed privacy protection framework PPGAN. We first introduce the concept of differential privacy. Subsequently, a brief introduction to GAN and WGAN. After that, we show the proposed PPGAN with theoretical analyses and the way noise is added to the gradients. Finally, we introduce *moments accountant* [15],

which is the fundamental idea in our framework to ensure the privacy of the iterative gradient descent process. We strictly prove in mathematics that the use of the moments accountant allows the discriminator to guarantee differential privacy.

A. Differential Privacy

Differential privacy (DP) [5], [6], [15] constitutes a solid standard for privacy guarantee for algorithms on the database. For all two datasets x and y, which differ by at most one record, we refer to these two datasets as an neighboring datasets. In the above description, natural measure of the distance between two databases x and y will be their distance:

Definition 1: **(Distance Between Databases)**
The ℓ_1 norm of a database x is denoted $||x||_1$ and is defined to be:

$$||x||_1 = \sum_{i=1}^{|\aleph|} |x_i| \tag{1}$$

The ℓ_1 distance between two databases x and y is $||x-y||_1$. In particular, when $||x-y||_1 = 1$, x and y are mutually referred to as neighboring datasets.

Definition 2: **((ε, δ)-DP)**
A randomized algorithm $\phi(\cdot)$ with domain $\Phi^{|x|}$ is (ε, δ)-DP if for all $O \subseteq Range(\phi)$ and for all $d, d' \in \Phi^{|\aleph|}$ (for any neighbouring datasets) such that $||d - d'|| \leq 1$:

$$Pr[\phi(d) \in O] \leq e^{\varepsilon} Pr[\phi(d') \in O] + \delta \tag{2}$$

Noted that ϵ stands for privacy budget, which controls the level of privacy guarantee achieved by mechanism ϕ. And when $\varepsilon = \infty$, this case is non-private. Actually, the canonical definition of ε-DP does not include the additive term δ in definition 2, which was definited as follows:

$$Pr[\phi(d) \in O] \leq e^{\varepsilon} Pr[\phi(d') \in O] \tag{3}$$

Since the privacy control in equation 3 is too strict and can not apply to deep neural networks, δ is added to optimize the original definition. (ε, δ)-DP provides freedom to violate strict ϵ-differential privacy for some low probability events. (δ preferably smaller than $1/|d|$.)

Among the mechanisms for achieving differential privacy, the two most widely used are the *Laplace mechanism and the Gaussian noise mechanism (GNM)* [18]. Due to the combined properties of the GNM, it is prevalent in many DP protection models. In PPGAN, we use the GNM because the moments accountant (detailed in Section III-D) provides an improved privacy boundary analysis and is well-matched to the combined properties of the GNM. The GNM is defined as follows:

$$\phi(x) \triangleq f(x) + N(0, \sigma^2 {s_f}^2) \tag{4}$$

The s_f is defined as *sensitivity*, which is only related to query type f. The sensitivity is defined as follows:

Definition 3: **(Sensitivity)**
We given the neighboring datasets x and x' and given a query $f : x \to \Omega$, the *sensitivity* of f as follows:

$$\Delta f = \max_{x,x'} ||f(x) - f(x')||_1 \tag{5}$$

Noted that it records the largest difference between query results on datasets x and x'.

According to the algotithm $\phi(\cdot)$ in definition 2 is stochastic and is not related to the distribution of the output data. Moreover, the Gaussian noise mechanism adds a well-design noise to a single gradient without affecting the entire gradient aggregation. Therefore, we can use this attribute with GAN so that GAN can generate high-quality data while satisfying differential privacy.

B. GAN and WGAN

Generative adversarial network (GAN)[4], [11], [19], [20] is a class of deep neural network architectures comprised of two networks, pitting one against the other (thus the *"adversarial"*). Suppose our generative model is $G(z)$, where z is

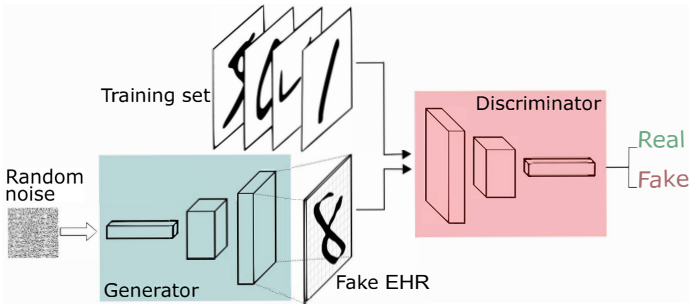


Fig. 1. Structure of the GAN model.

random noise and G converts this random noise into x. Take with contradicting training adjective *Electronic Health Record (EHR)* as an example. Let G be a generator synthesizing EHR, and D is the discriminator in the generator model. For an arbitrary input x, the output of $D(x)$ is a real number in the range [0,1] that determines how likely this EHR is authentic. Let P_r and P_g represent the distribution of real ones and the distribution of generated EHRs, respectively. The objective function of the discriminative model is as follows:

$$\max_D E_{x \sim \mathrm{Pr}}[\log(D(x)] + E_{x \sim Pg}[\log(1 - D(x)] \tag{6}$$

The goal of a similar from distinguishing is to prevent them from real records and the generated ones. The entire optimization objective function is as follows:

$$\begin{aligned} \min_G \max_D V(G, D) = {} & E_{x \sim P_{data}(x)}[\log(D(x)] \\ & + E_{z \sim P_z(z)}[\log(1 - D(G(z)))] \end{aligned} \tag{7}$$

WGAN[7] uses the *Wasserstein* distance instead of the *Jensen-Shannon* distance. Compared with the original GAN, WGAN's parameters are less sensitive and the training process is smoother. It solves a minimax two-player game that finds the balance point of each other:

$$\min_G \max_{w \in W} E_{x \sim \mathrm{P}data(x)}[f_w(x)] - E_{z \sim P_{z(z)}}[f_w(G(z))] \tag{8}$$

Finally, we give the WGAN training algorithm as follows:

Algorithm 1 WGAN, Arjovsky et al. [7] proposed the traning algorithm.

Require:

$\alpha = 0.00005$, the learning rate of WGAN. $c = 0.01$, the clipping parameter. m, the mini-batch size. $n_{critic} = 5$, the number of iterations per generator.

Require:

Initial critic parameters and generator's parameters ω_0, θ_0, respectively.

1: **while** θ has not converged **do**
2: **for** $t = 0, ..., n_{critic}$ **do**
3: $\{x^{(i)}\}_{i=1}^{m} \sim \mathrm{P}_\Upsilon$ a mini-batch from the real sample.
4: $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a mini-batch of prior samples.
5: $g_\omega \leftarrow \nabla_\omega 1/m \sum_{i=1}^{m} [f_\omega(x^{(i)}) - f_\omega(g_\theta(z^{(i)}))]$
6: $\omega \leftarrow \omega + \alpha \cdot RMS\,\mathrm{Pr}\,op(\omega, g_\omega)$
7: $\omega \leftarrow clip(\omega, -c, c)$
8: **end for**
9: Repeat the line 4.
10: $g_\theta \leftarrow -\nabla \frac{1}{m} \sum_{i=1}^{m} f\omega(g_\theta(z^{(i)})$
11: $\theta \leftarrow \theta - \alpha \cdot RMS\,\mathrm{Pr}\,op(\theta, g_\theta)$
12: **end while**

C. PPGAN framework

In this section, we present the proposed *Privacy-preserving Generative Adversarial Network (PPGAN) model*, which is detailed in Algorithm 2 and illustrated in Fig. 3. Noted that the $Discriminator$ has access to the real data, while the $Generator$ only receives feedback on the real data through the $Discriminator$'s output. This will be useful in PPGAN since only the $Discriminator$ is required to differential privacy. The $Generator$'s utilizes the result from the $Discriminator$, thus differential privacy [21]. (So we add noise proportional to the training data on the gradient of the Wasserstein distance, rather than adding noise to the final parameters.[22], [23])

Next, we introduce the differentially private discriminator model (DPDM). As shown in Fig. 3. DPDM reduces the sensitivity of these gradient-decreasing updates (and thus the overall accuracy) by clipping the stochastic gradients, virtually ensuring that the gradient will be within a bounded range. Hence, we only need to add noise proportional to the sensitivity to ensure differential privacy. (In the Algorithm 2.) We present the steps of PPGAN in Fig. 2:

D. Privacy Guarantees of PPGAN

To show that PPGAN in Algorithm 2 does satisfy the differential privacy, we prove that the parameters of the generator guarantee the differential privacy relative to the sample training point under the condition that the discriminator parameters satisfy the differential privacy. Therefore, the generated data from G satisfies the differential privacy, which means that G does not leakage the privacy of the datasets. Through moment accountant strategy, we can control the boundary of $g_w(x^{(i)}, z^{(i)})$ and calculate the final privacy loss. Along with Definition 2, intuitively, we have the definition of privacy loss at τ:

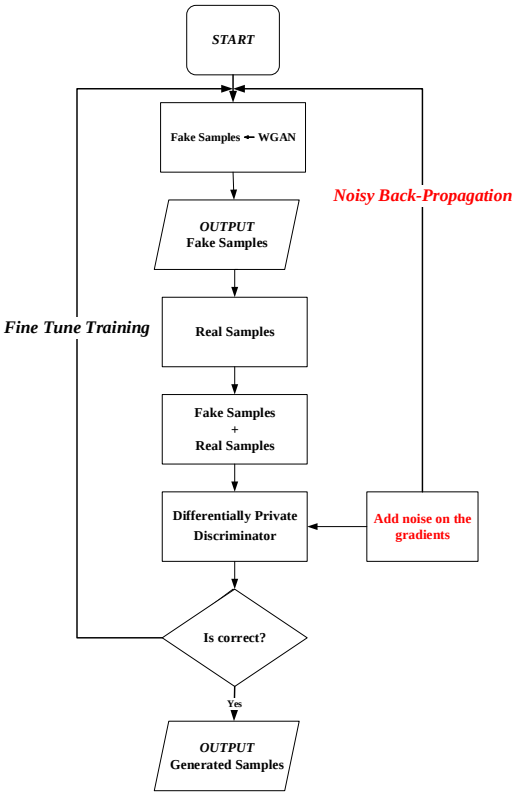


Fig. 2. The algorithm flow framework.

Algorithm 2 Privacy-preserving Generative Adversarial Network (PPGAN)

Require:

The learning rate: α. The clipping parameter: c. The batch size: m. The number of discriminator iterations per generator iteration: n_d. Generator iteration: n_g. Noise scale: σ_n. Gradient bound: C.

Ensure:

DP generator θ;

1: Initialize generator parameters and discriminator parameters ω_0, θ_0, respectively.
2: **for** $t_1 = 1, ..., n_g$ **do**
3: **for** $t_2 = 1, ..., n_d$ **do**
4: $\{x^{(i)}\}_{i=1}^{m} \sim \mathrm{P}_\Upsilon$ a mini-batch from the real data.
5: $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a mini-batch of prior samples.
6: $g_w \leftarrow g_w \min(1, C/||gw||) + N(0, {\sigma_n}^2 c_g^2 I)$ **(adding noise)**
7: Algorithm 1's line 6.
8: Algorithm 1's line 7.
9: **end for**
10: Repeat the line 5.
11: Algorithm 1's line 12.
12: $\theta \leftarrow \theta - \alpha_g \cdot RMS\,\mathrm{Pr}\,op(\theta, g_\theta)$
13: **end for**
14: **return** θ;

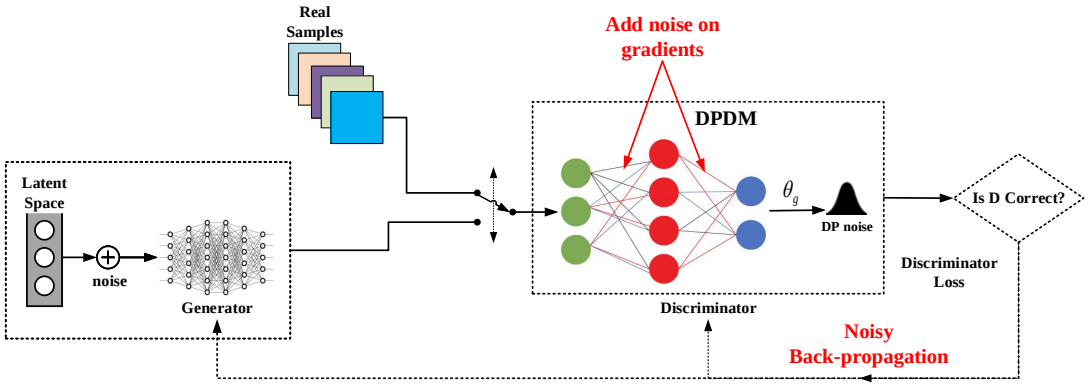


Fig. 3. Overview of our Privacy-preserving Generative Adversarial Network (PPGAN) model.

Definition 4: **(Privacy Loss)**

$$c(\tau;\phi,aux,d,d') \triangleq \log \frac{\mathrm{P}[\phi(aux,d)=\tau]}{\mathrm{P}[\phi(aux,d')=\tau]} \tag{9}$$

We introduce privacy loss to measure the distribution difference between two changing data. The privacy loss random variable is derived from the definition 2, which is used to describe the privacy budget of $\phi(d)$. For a given mechanism ϕ, we define the υ^{th} moment $\beta_\phi(\upsilon;aux,d,d')$ as the log of the moment generating function evaluated at the value:

Definition 5: **(Log moment generating function)**

$$\beta_\phi(\upsilon;aux,d,d') \triangleq \log E_{o\sim\phi}[e^{\upsilon C(\phi,aux,d,d')}] \tag{10}$$

Definition 6: **(Moments Accountant)**

$$\beta_\phi(\upsilon) \triangleq \max_{aux,d,d'} \beta_\phi(\upsilon;aux,d,d') \tag{11}$$

The basic idea behind the moments accountant is to accumulate the privacy expenditure by framing the privacy loss as a random variable and using its moment-generating functions to understand that variables distribution better. According to the definition of moments accountant and Post-Processing property (First presented in [15]), we know that the moments accountant has an excellent property, that is, the sum in each training iteration will limit the moments accountant. This property makes the PPGAN model training more stable. The tail bound can also be applied to privacy guarantee (In [15]). Since the moments accountant saves a factor of $\sqrt{\log(n_g/\delta)}$, according to Definition 2, this is a significant improvement for the large iteration n_g.

The following theorem, a proof of which can be found in [3], [15], [24], allows us to move the burden of differential privacy to the discriminator; the differential privacy of the generator will follow by the theorem.

Theorem 1: **(Post-processing)**
Let ϕ be an (ε,δ)-differentially private algorithm and let $f: \xi \to \xi'$ where ξ' is any arbitrary space. Then $f\circ\phi$ meets (ε,δ)-differentially private.

Proof 1: Related proofs can be seen [3], [15], [24].

Next, we present the mathematical reasoning proof that the discriminator satisfies the differential privacy. First, the moments accountant needs to add noise to the $g_w(x^{(i)},z^{(i)})$ which is limited to the boundary. Liyang Xie et al. add noise by clipping the norm of $g_w(x^{(i)},z^{(i)})$ in [3]. But we do not use the method in PPGAN. We clipping the parameter ω to add noise to the $g_w(x^{(i)},z^{(i)})$ in Algorithm 2.

Lemma 1: According to the conditions of Algorithm 2, we assume that the activate-function of D is bounded and bounded anywhere: $\varsigma(\cdot)\le\sigma_\varsigma$ and $\varsigma'(\cdot)\le\sigma_{\varsigma'}$, and each data distribution point x satisfies $||x||\in[-\sigma_x,\sigma_x]$. Then we have $||g_w(x^{(i)},z^{(i)})||\le c_g$ for some constant c_g.

Proof 2: We assume that p: input layer number; $\omega^{(m)}(l=1,...,p)$: weight matrix; $A^{(m)}$: the diagonal Jacobian of non-linearities of m-th layer. Then we have:

$$A_{ij}{}^{(m)} = \begin{cases} \varsigma'(\omega_{i,:}{}^{(m)}\varsigma(z^{(m-1)})), i=j \\ 0, i\neq j \end{cases} \tag{12}$$

where $\omega_{i,:}{}^{(m)}$ represents the ith row of $\omega^{(m)}$ and $\varsigma(z^{(m-1)})$ represents the output of the $(m-1)$-th layer. Then the back-propagation algorithm on the fully connected network is as follows:

$$\begin{aligned} \varphi^{(p)} &= \nabla_\lambda \odot \varsigma'(z^{(p)}), \\ \varphi^{(m)} &= ((\omega^{(m+1)})^T\varphi^{(m+1)})\odot\varsigma'(z^{(m)}), \\ \tfrac{\partial L}{\partial \omega_{jk}{}^{(m)}} &= \lambda_k{}^{(m-1)}\varphi_j{}^{(m)}, \end{aligned} \tag{13}$$

where L is the loss function, $z^{(m)}$ is the input layer m, $\lambda^{(m)}$ is the output layer m and $\varphi^{(m)}$ is the error vector of layer m. From Equation13 we have:

$$\begin{aligned} \frac{\partial L}{\partial \omega^{(m_0)}} &= (A^{(m)}(\omega^{(m+1)})^T ... A^{(p-1)}(\omega^{(p)})^T) \\ &\quad * (\lambda^{(m-1)})^T\varsigma'(z^{(p)}). \end{aligned} \tag{14}$$

For $\frac{\partial L}{\partial \omega^{(m_0)}}$, we have:

$$\begin{aligned} [A^{(m)}(\omega^{(m+1)})^T]_{ij} &\le c_p\sigma_{\varsigma'} \\ [A^{(m)}\omega^{(m+1)})^T A^{(m+1)}(\omega^{(m+2)})^T]_{ij} &\le (c_p\sigma_{\varsigma'})^2 n_{m+1} \end{aligned} \tag{15}$$

where we assume that $c_p \leq \frac{1}{n_{m+1}\sigma_{\varsigma'}}$. And n_{m+1} represents the number of nodes in the $m+1$-th layer. So we have:

$$[\prod A^{(m)}(\omega^{(m+1)})^T] \leq (c_p\sigma_{\varsigma'})^{p-m_0} \prod n_{m+1}. \quad (16)$$

Combining Lemma1 and Theorem2, we have $[\frac{\partial L}{\partial \omega^{(l)}}]_{ij} \leq c_p\sigma_\varsigma{\sigma_{\varsigma'}}^2$. Therefore, we have:

$$||g_w(x^{(i)}, z^{(i)}|| \leq 2c_p\sigma_\varsigma{\sigma_{\varsigma'}}^2 \sum_{k=1}^{m-1} n_k n_{k+1} = c_g \quad (17)$$

According to [3], the conditions for the discriminator to guarantee differential privacy are given as follows:

$$\sigma_n = 2q\sqrt{n_d \log(\frac{1}{\delta})}/\varepsilon \quad (18)$$

where q is the sampling probability and n_d is the number of iterations of the discriminator in each loop.

Lemma 2: Equation18 represents the relationship between the noise level σ_n and the privacy level ϵ. When we give a fixed perturbation σ_n on the gradient, according to equation18, we know that the larger the q, the D gets the fewer privacy guarantee. Because the D calculates more data, the privacy that can be allocated on each data point is limited. In addition, due to the data provides more information, more iterations (n_d) will result in fewer privacy guarantees. The facts described above require us to be cautious when choosing parameters to achieve a reasonable level of privacy. Finally, we use the following theorem as a privacy guarantee for generator parameters:

Theorem 2: The output of generator learned in Algorithm 2 guarantees (ε, δ)-DP.

PPGAN modifies the GAN framework to keep differentially private while relying on Theorem 1 and Lemma 2 to change the differential private G to train the differentially private D. The key idea is to add noise to the gradients of the discriminator during the training process to create a differential privacy guarantee.

IV. EXPERIMENTS

In this section, we will conduct a series of experiments to investigate how the privacy budget affects the effectiveness of PPGAN on the two benchmark datasets MNIST [25] and MIMIC-III [26], [27]. MIMIC-III is a well-known public EHR database that includes medical records of 46,520 intensive care units (ICUs) over the age of 11[3]. We employ PPGAN to generate EHRs and protected privacy information at the same time. In the experiment, we focus on three issues: 1) Relationship between Privacy budget and Generation Performance; 2) Relationship between Privacy budget and High-quality Datasets; 3) Utility Evaluation of PPGAN.

A. Data preprocessing

First, we only use the extracted ICD9 code (The ICD9 code represents the type of disease, and the range of coding is $C \in [1, 1071]$.[23], [28]) and use the first three digits for encoding. We then record the patient's admission to the disease and turn it into a vector x. For example, patient P was diagnosed with three diseases at admission, and the disease codes are indicated by 9, 42, 146, respectively. (So the ICD9 code consists of 9, 42 and 146.) We use the vector x to indicate the patient's access record, where the vector is at position 9, the 42nd and 146th bits are set to 1, and the rest are set to 0. Then we aggregate the patient's longitudinal record into a single fixed-size vector $x \in \mathrm{Z}^+$, where $|C| = 1071$ for dataset. Noted that the MIMIC-III is transformed into 0-1 codes for experiments with binary variables.

B. Relationship between Privacy budget and Generation Performance

In this section, we mainly explore the relationship between privacy budget and generation performance. Considering the combined properties data of $Gaussian\ noise$, we add Gaussian noise in the process of stochastic gradient descent. Different Gaussian noises can produce different levels of privacy. We input the same set of MNIST image datasets and observe the output generated samples. In the experiments, $\alpha_d = 5.0\times10^{-5}$ learning rate of discriminator; $\alpha_g = 5.0 \times 10^{-5}$, learning rate of generator; moments accountant parameter $C = 1.0 \times 10^{-2}$; noise scale $\delta = 1.0 \times 10^{-5}$, and the number of iterations on discriminator t_d and generator t_g are 5 and 5.0×10^5, respectively. The experimental results are shown in Fig. 4. The code is available.[1]

As shown in Fig. 4, as the privacy budget increases, the quality of the generated images is getting worse. We add well-designed noise that disturbs the data point distribution of the image. Since the noise is randomly added, the distribution of disturbing data points is not fixed, thus ensuring differential privacy. Please note that we cannot ignore the quality of the generated image because we protect private information. We should choose a reasonable privacy budget to generate an applicable image. Generating quality and privacy budget is a compromise issue.

Next, we will focus on the impact of noise on PPGAN's loss function during training. The results are shown in Fig. 5 In the non-private case, we observe the training loss of the first 100 epoch in training. The result indicates that the loss of GAN is smooth and stable, and no large fluctuations exist in this round of training. When the loss of the PPGAN with noise starts to fluctuate at the tail of the curve, PPGAN can still converge. As can be inspected from Fig. 5, the convergence rate of PPGAN is acceptable as the compromise of the introduced privacy preservation capability.

C. Relationship between Privacy budget and High-quality Datasets

In this section, we quantitatively evaluate the performance of PPGAN. Specifically, we first compare generated data with real data based on statistical characteristics. We propose a *Generate score* to measure the quality of data generated by GAN.

[1]https://github.com/hdliuyi/PPGANs-Privacy-preserving-GANs

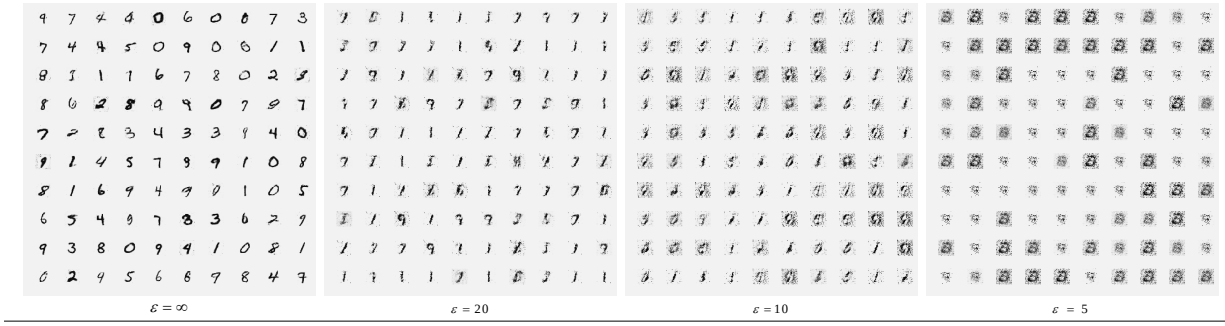
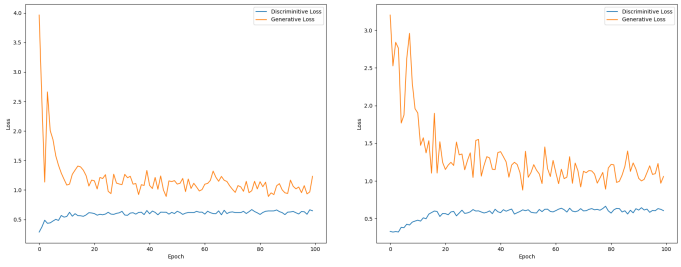


Fig. 4. Four differentaϵ values are generated for four different quality pictures on MNIST dataset.($\varepsilon = \infty, \varepsilon = 20, \varepsilon = 10, \varepsilon = 5$; $\delta = 1.0 \times 10^{-5}$)



Non-private Case **Private Case**

Fig. 5. Loss of Non-private Case ($\varepsilon = \infty$) and Private Case ($\varepsilon \neq \infty$).

We proposed *Generate score* ($GS(P_g)$) to measure the quality of data generated by PPGAN, which can be formally defined as follows for P_g:

Definition 7: (Generate scores):

$$\begin{aligned} IS(P_g) &= e^{E_{x\sim P_g}[KL(PM(y|x)||PM(y))]} \\ GS(P_g) &= |\frac{IS(P_g) - mean(IS(P_g))}{\max(IS(Pg)) - \min(IS(P_g))}| \end{aligned} \tag{19}$$

where $IS(P_g)$ is *Inception score* which is measure of the performance of the GAN.

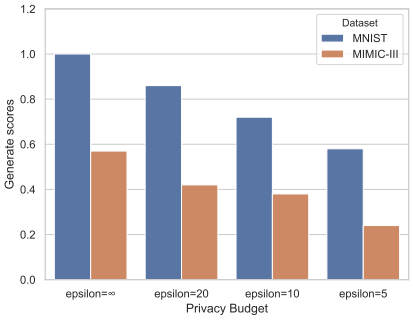


Fig. 6. Generate scores of generative data on MNIST.

The experimental result is shown in Fig. 6. The generated data's (generated by PPGAN) generate score is compared to the real data of the MNIST dataset with different privacy budgets. The larger the score value, the better the quality of the data generated by the generator. The figure shows the distribution of the generate scores of PPGAN in the case of $\epsilon = 20, 10, 5$. It can be seen from the figure that the score is very close to the real data generated by the WGAN (non-private case, $\varepsilon = \infty$.). When $\epsilon = 20$, the PPGAN generate score is only 0.14 different from the WGAN generate score, which indicates that the PPGAN generation quality is close to the WGAN.

To evaluate the performance of PPGAN, we compare three solutions, namely dp-GAN [29], DPGAN [3] and WGAN [7] (Non-private Case) in terms of the quality of the generated data.

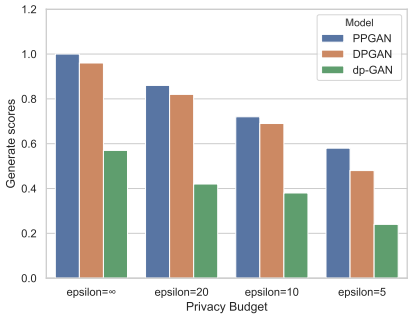


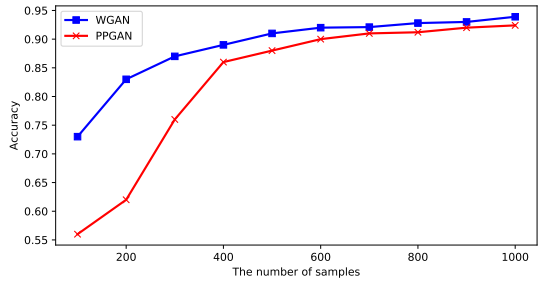
Fig. 7. Generate scores of generative data on model PPGAN, DPGAN and dp-GAN. ($\delta = 1.0 \times 10^{-5}$)

As can be seen from Fig. 7, the data quality generated by PPGAN is better than dp-GAN and DPGAN.

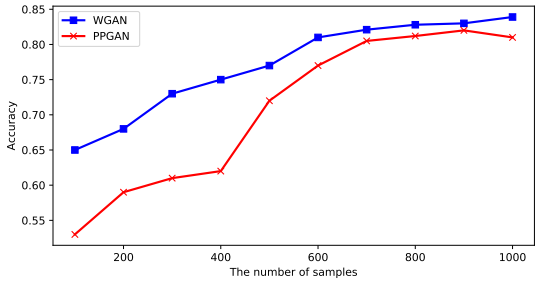
D. Utility Evaluation of PPGAN

In this subsection, we focus on the classification performance of PPGAN on the MNIST and MIMIC-III datasets and compare them with the existing model WGAN. For fixed $\varepsilon = 5$ and $\delta = 1.0 \times 10^{-5}$, we add the generated data from PPGAN and WGAN to the semi-supervised classification task with a small amount of marker data. Then we focus on the impact of the number of training samples on the classification accuracy of the classifier. With 100 training sessions as a unit, randomly select different numbers of samples for repeated training and record the accuracy of the MNIST and MIMIC-III datasets. For the MIMIC-III data set, for fair comparison, we assign values to the initial variables of medGAN in [8] (Learning rate of D, learning rate of G and times per iteration:

$\theta_d = \theta_g = 5.0 \times 10^{-5}, k = 2$) and execute our PPGAN model. (The code is available. [2])



(a)



(b)

Fig. 8. The accuracy of classification on two benchmark datasets. (a) MNIST (b) MIMIC-III

The experimental results are shown in Fig. 8 which show that PPGAN can perfectly perform classification tasks under reasonable privacy budget requirements, because PPGAN can generate high quality data samples with reasonable privacy level. When the privacy budget is within reasonable limits (For example, $\varepsilon \approx 20$), more than 90% of the generated data is of high-quality and availability. This can solve the problem of data sharing in related research.

V. CONCLUSION

In this paper, we propose the PPGAN model that preserves the privacy of training data in a differentially private case. PPGAN mitigates information leakage by adding well-designed noise to the gradient during the learning process. We conducted two experiments to show that the proposed algorithm can converge under the noise and constraints of the training data and generate high-quality data. Also, our experimental results verify that PPGAN does not suffer from mode collapse or gradient disappearance during training, thus maintaining excellent stability and scalability of model training.

ACKNOWLEDGMENTS

This work is supported in part by the Nature Science Foundation of Heilongjiang Province of China (Grant NO.F2016035 and NO.QC2016091), Ministry of Education of China and the School of Entrepreneurship Education of Heilongjiang University (Grant NO.201910212133).

[2]https://github.com/mp2893/medgan

REFERENCES

[1] Y. Wu, Y. Liu, A. Alghamdi, K. Polat, and J. Peng, "Dominant dataset selection algorithms for time-series data based on linear transformation," 2019.
[2] P. Voigt and A. V. D. Bussche, *The EU General Data Protection Regulation (GDPR)*, 2017.
[3] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.
[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," 2018.
[5] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: Information leakage from collaborative deep learning," 2017.
[6] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, & Programming*, 2006.
[7] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017.
[8] E. Choi, S. Biswal, B. Malin, J. Duke, and J. Sun, "Generating multi-label discrete patient records using generative adversarial networks," 2017.
[9] A. K. Mondal, J. Dolz, and C. Desrosiers, "Few-shot 3d multi-modal medical image segmentation using generative adversarial learning," 2018.
[10] X. Yuan, X. Wang, C. Wang, A. Squicciarini, and K. Ren, "Towards privacy-preserving and practical image-centric social discovery," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, vol. 15, no. 5, pp. 251 – 265, 2018.
[11] G. J. Qi, "Loss-sensitive generative adversarial networks on lipschitz densities," 2017.
[12] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, R. Lee, S. P. Bhavnani, J. B. Byrd, and C. S. Greene, "Privacy-preserving generative deep neural networks support clinical data sharing," *Circulation: Cardiovascular Quality and Outcomes*, vol. 12, no. 7, p. e005122, 2019.
[13] S. Shuang, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Global Conference on Signal & Information Processing*, 2014.
[14] K. Chaudhuri, C. Monteleoni, and D. Sarwate, *Differentially private empirical risk minimization*, 2011.
[15] M. Abadi, A. Chu, I. Goodfellow, H. B. Mcmahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," 2016.
[16] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Allerton Conference on Communication*, 2015.
[17] N. Papernot, M. Abadi, l. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2017.
[18] J. Dong, A. Roth, and W. J. Su, "Gaussian differential privacy," 2019.
[19] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, X. Bing, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *International Conference on Neural Information Processing Systems*, 2014.
[20] X. Yuan, X. Wang, C. Wang, J. Weng, and K. Ren, "Enabling secure and fast indexing for privacy-assured healthcare monitoring via compressive sensing," *IEEE Transactions on Multimedia (TMM)*, vol. 18, no. 10, pp. 1–13, 2016.
[21] C. Dwork and A. Roth, *The Algorithmic Foundations of Differential Privacy*, 2014.
[22] A. Triastcyn and B. Faltings, "Generating artificial data for private deep learning," 2018.
[23] A. L. Buczak, S. Babin, and L. Moniz, "Data-driven approach for creating synthetic electronic medical records," *BMC Medical Informatics and Decision Making,10,1(2010-10-14)*, vol. 10, no. 1, p. 59, 2010.
[24] J. Jordon, J. Yoon, and M. van der Schaar, "Pate-gan: generating synthetic data with differential privacy guarantees," 2018.
[25] http://yann.lecun.com/exdb/mnist/.
[26] A. E. W. Johnson, T. J. Pollard, S. Lu, L. W. H. Lehman, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.
[27] https://github.com/MIT-LCP/mimic-code.
[28] S. Mclachlan, K. Dube, and T. Gallagher, "Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record," in *IEEE International Conference on Healthcare Informatics*, 2016.
[29] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," 2017.