electronics

MDPI

Article

OFPP-GAN: One-Shot Federated Personalized Protection–Generative Adversarial Network

Zhenyu Jiang [1], Changli Zhou [1,2,*], Hui Tian [1,2] and Zikang Chen [1]

[1] College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China; jiangzy@stu.hqu.edu.cn (Z.J.); htian@hqu.edu.cn (H.T.); 2125102008@stu.hqu.edu.cn (Z.C.)
[2] Xiamen Key Laboratory of Data Security and Blockchain Technology, Huaqiao University, Xiamen 361021, China
* Correspondence: zcl@hqu.edu.cn

Abstract: Differential privacy techniques have shown excellent performance in protecting sensitive information during GAN model training. However, with the increasing attention to data privacy issues, ensuring high-quality output of generative models and the efficiency of federated learning while protecting privacy has become a pressing challenge. To address these issues, this paper proposes a One-shot Federated Personalized Protection–Generative Adversarial Network (OFPP-GAN). Firstly, this scheme employs dual personalized differential privacy to achieve privacy protection. It adjusts the noise scale and clipping threshold based on the gradient changes during model training in a personalized manner, thereby enhancing the performance of the generative model while protecting privacy. Additionally, the scheme adopts the one-shot federated learning paradigm, where each client uploads their local model containing private information only once throughout the training process. This approach not only reduces the risk of privacy leakage but also decreases the communication overhead of the entire system. Finally, we validate the effectiveness of the proposed method through theoretical analysis and experiments. Compared with existing methods, the generative model trained with OFPP-GAN demonstrates superior security, efficiency, and robustness.

Keywords: information security; one-shot federated learning; personalized differential privacy; data generation

check for updates

Citation: Jiang, Z.; Zhou, C.; Tian, H.; Chen, Z. OFPP-GAN: One-Shot Federated Personalized Protection–Generative Adversarial Network. *Electronics* **2024**, *13*, 3423. https://doi.org/10.3390/electronics13173423

Academic Editor: Aryya Gangopadhyay

Received: 3 August 2024
Revised: 20 August 2024
Accepted: 28 August 2024
Published: 29 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The Generative Adversarial Network (GAN) is an efficient technology that can abstract and approximate the probability distribution of data, achieving great success in various generative tasks that replicate real-world content. In 2014, Goodfellow et al. proposed GAN, a generative model that bridges the gap between supervised and unsupervised learning [1]. In 2018, Turing Award winner Yann LeCun called it "the most interesting idea in machine learning in the last ten years". In recent years, GAN has been widely applied in fields such as medical diagnosis [2], image processing [3], and image translation [4], due to their outstanding ability to generate realistic synthetic samples. However, GAN also has some issues. The model may overly rely on training data and, due to the high complexity of deep neural networks, easily memorize training samples. When GAN models are applied to scenarios involving sensitive or private data, such as patient medical records and bank statements, there is a risk of privacy leakage [5]. Therefore, it is urgent to find methods to construct protected generative models to ensure data privacy and security.

In 2006, Dwork et al. proposed a new privacy protection technique called differential privacy to address the problem of statistical data privacy leakage [6]. Under this technique, the released data are appropriately perturbed so that whether a single data record is in the database has a negligible impact on the computation results. Due to its independence from background knowledge and strict mathematical definition, differential privacy has been widely applied in various fields for privacy protection. To meet third-party demands for

data usability and privacy while protecting private data, many researchers have combined GAN with differential privacy to construct differentially private generative models. The Differential Privacy Generative Adversarial Network (DPGAN) algorithm is suitable for fields with data scarcity due to privacy issues [7]. For example, hospitals cannot directly release patient information due to privacy and sensitivity issues, but they wish to collaborate with third parties on disease diagnosis and other research work. Although hospitals cannot directly release patient information, they can use patient data locally. In this case, hospitals can use the DPGAN framework to train a DPGAN model with patient data locally, generate a protected generative model locally, and then publish the generative model. Third parties can use the generative model to generate synthetic data that meet the statistical characteristics of the original data for various analytical tasks, thus having high research value. Although DPGAN models perform well in centralized learning, their application is limited by the "data silo" nature of hospital data. This challenge has driven researchers to explore a privacy-preserving distributed machine learning approach known as federated learning [8]. Federated learning not only enhances resource utilization in shared environments but also protects users' sensitive information by keeping data local, thus preventing the formation of data silos. Consequently, researchers have sought to combine federated learning with GAN to further improve model utility while ensuring effective protection of user privacy.

However, mainstream differential privacy methods based on GAN have a significant drawback: they do not consider personalized differential privacy schemes but use uniform noise scales and clipping thresholds across all clients. This approach easily overlooks the differences in data distribution among different clients, affecting model accuracy and privacy protection capability [9]. On the one hand, an overly large noise scale during the training process will make the generated model excessively randomized, leading to a decline in model quality. On the other hand, an overly small noise scale cannot effectively protect privacy, increasing the risk of model attacks. Additionally, setting the clipping threshold is equally important. A too-high clipping threshold will result in severe loss of gradient information, affecting model training performance; a too-low threshold will introduce too much noise, damaging the quality of generated data [10]. Therefore, a uniform noise scale and clipping threshold cannot balance privacy protection and model performance. To address these issues, this paper proposes a method to calculate noise scales and clipping thresholds based on gradients in a personalized manner during GAN model training. By analyzing the data characteristics of each client and dynamically adjusting the noise scale and clipping threshold, we can improve the quality and accuracy of generated models while ensuring privacy protection. This personalized differential privacy scheme can better adapt to the personalized data distribution of each client, significantly enhancing the performance of GAN models. By introducing a personalized differential privacy scheme, we achieve a dual improvement in privacy protection and model performance in GAN model training, providing a more secure and effective solution.

Additionally, existing federated learning frameworks rely on traditional paradigms, where frequent communication exposes sensitive data to potential attacks during transmission, increasing the risk of privacy leakage [11]. For example, Hitaj et al. proposed an active inference attack model that can reconstruct training samples from the released generative model, leading to the leakage of original training data [12]. Fredrikson et al. proposed a more robust inversion attack method that uses predicted confidence values to recover sensitive information from original training data [13]. To solve these problems, this paper proposes using the one-shot federated learning paradigm. In this new paradigm, clients need to upload their locally trained generative models containing private information only once during the entire training process. By reducing communication frequency, we not only significantly reduce the communication overhead in GAN model training but also effectively decrease the risk of privacy leakage. This advantage is particularly evident in the training of generative models involving highly sensitive data, as the one-shot federated learning paradigm not only protects data privacy but also improves the performance and stability of generative models.

In summary, based on the above issues, this paper proposes the One-shot Federated Personalized Protection–Generative Adversarial Network (OFPP-GAN) scheme. The main contributions of this paper include the following three parts:

- To better balance the privacy and usability of GAN models, we introduce dual personalized differential privacy in federated GAN model training, adjusting noise scales and clipping thresholds based on gradient variations in model training. This marks the first application of personalized differential privacy in federated learning with GAN, ensuring the usability of GAN models without compromising privacy protection capabilities.
- To reduce the substantial communication overhead during federated learning training and prevent malicious attackers from targeting the model, we introduce the one-shot federated learning paradigm. This method significantly reduces communication overhead by minimizing frequent data transmissions during training, further enhancing the model's privacy protection capabilities.
- Through experiments, we demonstrate that our method outperforms state-of-the-art techniques in training GAN models. Our method not only generates high-quality synthetic images but also reduces communication overhead during training, all without compromising the intensity of privacy protection or affecting the overall computational cost of the system.

2. Preliminaries

2.1. Generative Adversarial Network

As shown in Figure 1, GAN consists of two components: the generator and the discriminator. The generator's task is to generate samples that are as realistic as possible, while the discriminator's task is to distinguish between the generated samples and real samples. During the training process, the generator and discriminator compete with each other, continuously optimizing their respective parameters until they reach a balanced state. The basic idea is to train the generator and discriminator through a competitive process, so the generator generates samples similar to real data. The loss function of the GAN is shown in Equation (1):

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[1-\log D(G(z))] \quad (1)$$

where G represents the generator, and D represents the discriminator. $V(G,D)$ is the objective function of the Generative Adversarial Network (GAN), where $p_{data}(x)$ denotes the real data distribution, representing the probability distribution of sampling x from the real dataset. $p_z(z)$ denotes the noise distribution, representing the probability distribution of sampling z from the noise. The generator $G(z)$ transforms the noise z into synthetic data samples, while the discriminator $D(x)$ estimates the probability that the sample x comes from the real data.

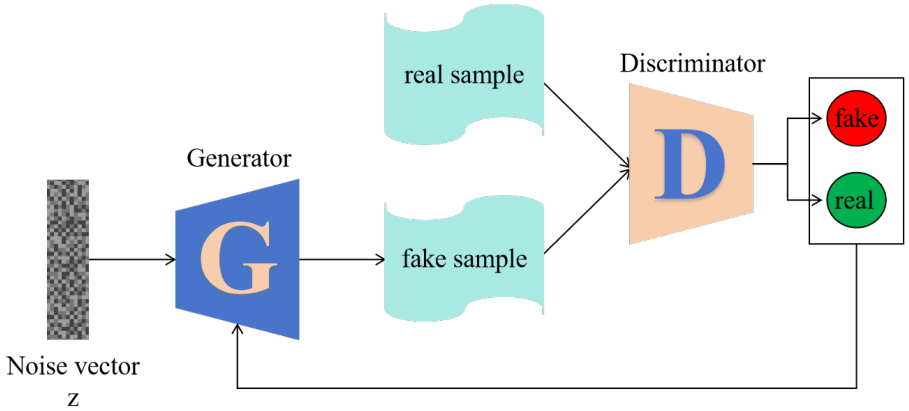


Figure 1. The architecture of GAN.

2.2. Differential Privacy

In our method, the privacy model used is differential privacy, denoted by $A_p(\cdot)$, which is a randomized algorithm such that it is difficult for an observer to re-identify the input data. Here, the observer is anyone using the data to obtain the algorithm's output.

Definition 1 (Differential Privacy, DP)**.** *A randomized algorithm A_p satisfies differential privacy if and only if for two adjacent datasets D and D', and a generated dataset S obtained through the randomized algorithm A_p, the following holds:*

$$\Pr\big(A_p(D) \in S\big) \leq e^{\varepsilon} \cdot \Pr\big(A_p\big(D'\big) \in S\big) + \delta \tag{2}$$

where $\Pr(\cdot)$ denotes the probability that the result of the randomized algorithm A_p is the given target dataset, ε denotes the privacy budget, and δ denotes the probability of not satisfying strict differential privacy.

2.3. Federated Learning

Federated learning algorithms are defined by their aggregation strategies and whether they impact the local training process. The aggregation strategy determines how model weights from different nodes are considered when computing the federated model weights. Initially proposed by McMahan et al., the federated strategy known as federated averaging (FedAvg) uses a weighted average of the model weights returned by the nodes to calculate the federated model weights [8]. The weighting coefficients are the number of samples each node used for model training divided by the total number of samples from all sites. Equation (3) shows the averaging process:

$$w_{federated} = \frac{1}{\sum_{i=1}^{K} N^{(i)}} \sum_{i=1}^{K} N^{(i)} w_{site}^{(i)} \tag{3}$$

where $w_{federated}$ is the global model parameter, $N^{(i)}$ is the data size of the i-th client, and $w_{site}^{(i)}$ is the local model parameter of the i-th client.

For scenarios where clients have Independent Identically Distributed (IID) datasets, this federated strategy has shown promising results. However, this strategy biases the final federated model towards the models trained on clients with the most samples, ultimately causing "client drift" [14].

3. Related Works

3.1. Privacy Protection Scheme Based on GAN

In recent years, GANs have demonstrated remarkable capabilities in generating high-quality synthetic data, but they have simultaneously raised significant concerns regarding data privacy leakage. To mitigate these risks, various privacy-preserving schemes leveraging GAN have been proposed. Among these, DPGAN stands out as a pioneering framework, employing GAN architecture under differential privacy constraints by introducing noise into the discriminator's gradients during training. This method effectively ensures the entire GAN framework adheres to differential privacy standards [7]. However, the utility of data in DPGAN is often compromised due to the noise addition, posing a challenge to maintaining data quality. Huang et al. attempted to address this issue with the DPWGAN scheme, integrating moment accounting and Wasserstein loss to strike a better balance between privacy and data utility [15]. Although DPWGAN shows improvements in data utility, its performance is heavily dependent on the complexity of the dataset and the choice of hyperparameters, which may limit its generalizability across different applications. Other notable contributions include the PE-GAN framework, which employs truncated centralized differential privacy techniques to enforce stricter privacy boundaries [16]. While PE-GAN enhances privacy protection, its approach may lead to increased computational complexity, making it less suitable for real-time applications. LDP-GAN, on

the other hand, applies differential privacy noise directly to the data, tailoring the noise to the data's specific characteristics [17]. This method offers a more granular approach to privacy but requires careful calibration to avoid excessive utility loss. PATE-GAN uses a multi-discriminator strategy to improve both the quality and privacy of synthetic data [18]. However, the increased complexity of the model raises concerns about scalability and the potential for overfitting, especially in scenarios with limited data. PAR-GAN enhances GAN's generalization capabilities to defend against membership inference attacks, yet it remains unclear how well this defense holds under different attack vectors [19]. In addition, privacy-preserving methods using Deep Convolutional GAN (DCGAN) combined with encrypted transmission and embedding detection techniques have been proposed to enhance model stability and information protection in high-risk applications, but these methods often require significant computational resources [20].

Furthermore, researchers have proposed various innovative methods to defend against potential attacks on GAN model training, such as membership inference attacks and poisoning attacks. Chen et al. proposed the first classification method for membership inference attacks on generative models, highlighting the privacy risks inherent in these models [21]. GANobfuscator demonstrated that introducing differential privacy mechanisms during training could effectively reduce information leakage risks posed by membership inference attacks [22]. PrivacyNet introduced privacy-preserving attributes to facial images through image perturbation techniques, but its effectiveness may be limited to specific types of visual data [23]. The PID method addressed GAN poisoning attacks in federated learning by utilizing multi-trust domains and differential privacy techniques, yet the robustness of this approach against sophisticated adversaries remains an open question [24]. Additionally, Auto-Driving GAN and TPE-GAN have been developed to protect data in specialized domains like autonomous vehicles and encrypted cloud storage, respectively, though their applicability beyond these niches is yet to be fully explored [25,26].

3.2. Federated Learning Scheme Based on GAN

Research on federated learning based on GAN primarily focuses on two aspects.

On the one hand, federated learning frameworks cannot guarantee the security of both clients and servers, creating vulnerabilities that attackers may exploit using GAN. Hitaj et al. hypothesized a scenario where a malicious client could still reveal sensitive information from the target training set by generating adversarial samples, even when privacy protection measures such as differential privacy are employed [12]. This work underscores the inherent risks in federated learning systems, suggesting that privacy-preserving mechanisms must be robust against such sophisticated attacks. Similarly, Wang [27] considered the possibility of a malicious server, demonstrating that a server equipped with a multi-task discriminator could potentially generate private data from specific clients, further highlighting the need for more secure federated learning protocols.

On the other hand, in scenarios requiring stringent privacy measures, such as in the medical field, data custodians are often reluctant to share their data, making centralized model training impractical. To address this, several studies have explored the feasibility of training GAN models within federated learning frameworks. Zhang et al. proposed the FedDPGAN framework, which effectively generates synthetic medical data while preserving privacy, offering a viable solution to the challenge of fragmented medical data [28]. However, the framework's reliance on the quality of local data raises concerns about its scalability across diverse healthcare settings. Xin et al. introduced a novel data generation method that combines differential privacy with a serial federated learning paradigm, significantly reducing communication overhead compared to traditional federated learning models [29]. Despite its efficiency, the method's applicability to non-IID data remains an area for further investigation. Meanwhile, Cao et al. proposed a personalized GAN training scheme, allowing clients to design and train their models independently, which could potentially enhance model personalization and data utility [30]. However, the decentralized nature of this approach may introduce challenges in coordinating and integrating models

across different clients. Hardy et al. proposed MD-GAN, a federated learning framework where the server hosts the generator while each client hosts a discriminator [31]. This peer-to-peer communication strategy aims to improve computational efficiency, but its reliance on network stability could be a limiting factor in real-world deployments. FedGAN, designed to train GAN on non-IID data distributions, reduces communication complexity, yet its effectiveness in maintaining model accuracy across heterogeneous datasets requires further validation [32]. Additionally, the literature has explored methods for fine-tuning the federated GAN using synthetic source data to generate unbiased synthetic datasets [33] and the Fed-TGAN framework, which facilitates learning complex tabular data across different clients [34], both of which show promise but also highlight the ongoing challenges of balancing data utility and privacy in federated learning environments.

3.3. One-Shot Federated Learning

One-shot federated learning minimizes communication costs and enhances privacy protection by restricting interactions between clients and the server to a single round. This makes it extremely effective in scenarios where multiple rounds of communication are impractical or pose privacy risks.

For instance, Guha et al. proposed an algorithm for training support vector machines in a one-shot manner and introduced a framework that not only uploads the model but also includes information about local dataset distributions [35]. This approach is inappropriate for handling privacy-sensitive datasets. Meanwhile, Kasturi et al. employed knowledge transfer to refine models on each client, which could alter the original model structure and result in significant communication costs when sharing the student models [36]. Song et al. attempted to change the original optimization objective by sending the refined images to a central server but did not address privacy concerns [37]. Additionally, the FedKT achieves federated learning in a one-shot communication round, supports various classification models, and provides differential privacy protection, effectively addressing the issues of multiple communication rounds and high privacy risks [38].

4. Scheme Design

The process of a personalized privacy-preserving method based on GAN is shown in Figure 2. First, the server initializes the parameters for the generator, discriminator, and other components and sends them to each federated learning participant. Next, the federated learning participants locally train the GAN model by training the generator and discriminator separately. During the local training process, the participants personalize the noise scale and clipping threshold in differential privacy, ensuring that the trained generator model has differential privacy protection. Finally, the federated learning participants upload the trained local generator models to the server, which aggregates all the local models to obtain a final aggregated global generator model.

4.1. Personalized Noise Addition

In machine learning, when gradient changes increase, the magnitude of parameter adjustments also increases. Such large adjustments may cause the model to more prominently memorize certain data points' features, thereby increasing the potential risk of privacy leakage [39,40]. For example, during the backpropagation process, attackers might infer the presence and characteristics of specific data points by analyzing gradient changes. This is particularly concerning in distributed training like federated learning, where shared gradient information could be exploited by malicious attackers to reverse-engineer the original sensitive data.

To address this, our study employs a gradient-based personalized noise addition mechanism. By using hierarchical noise injection, clients introduce noise proportional to the parameter changes by applying personalized noise locally to the parameters. Specifically, the more significant the parameter fluctuations, the more noise is added, and vice versa.

Conversely, if the parameter fluctuations are minor, less noise is added. The formula for defining the personalized noise scale is as follows:

$$\sigma_{l_i} = \frac{g_\omega}{\sqrt{E[g^2]_t + \tau}}\sigma \tag{4}$$

where $E[g^2]_t$ represents the current average of the squared gradients at round t, g_w denotes the gradient in the current round, and τ is introduced to prevent the denominator from being zero, usually set to 1×10^{-8}. σ is the initially set noise scale. σ_{l_i} represents the personalized noise added to the i-th dimension of the local client's parameters to meet the noise requirements of different gradient changes.

Our study found that this personalized noise scheme can add differential privacy noise when there are significant gradient changes, providing corresponding protection. However, during the training process of GAN, sometimes there are minor gradient changes, and the amount of noise added is correspondingly small, making it challenging to achieve effective data protection. Therefore, to accommodate different situations in training, we set a global minimum noise scale $\sigma_{\min}$ to enhance the privacy protection effect of the entire model. Finally, the noise scale for each client is as follows:

$$\sigma_{l_i} = \max(\sigma_{\min}, \sigma_{l_i}) \tag{5}$$

where σ_{l_i} represents the personalized added noise, and $\sigma_{\min}$ represents the predefined noise lower bound. Before adding noise, the larger value between σ_{l_i} and $\sigma_{\min}$ is chosen as the amount of noise to be added.

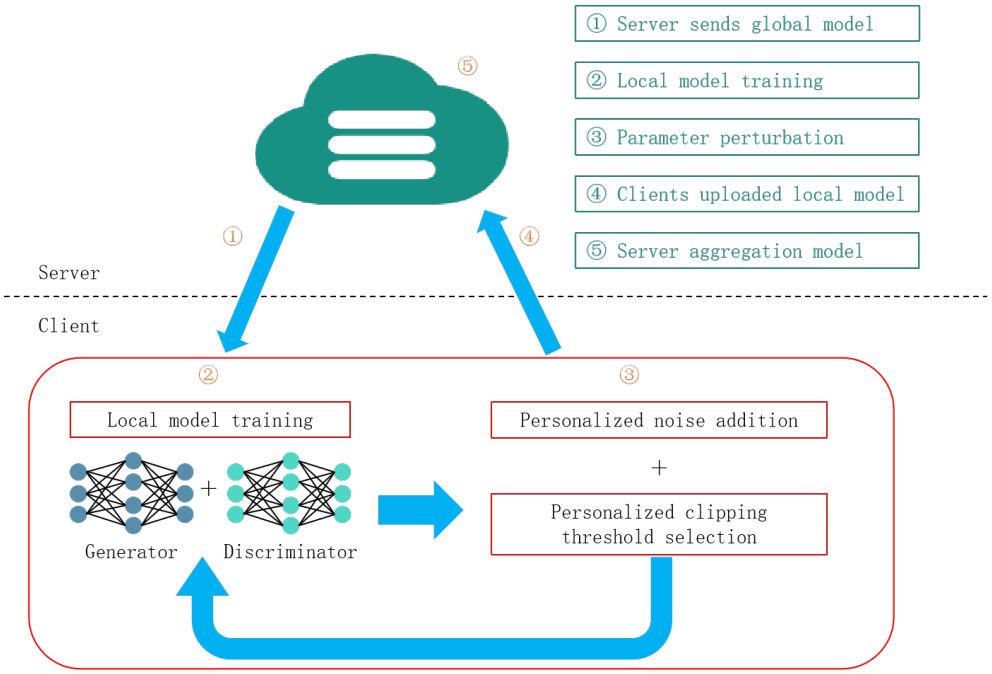


Figure 2. Overview of OFPP-GAN framework.

4.2. Personalized Clipping Threshold Selection

In addition, this study considered the impact of the clipping threshold on the model. If the clipping threshold is set too low, it will not only lead to excessive noise addition but also cause excessive gradient clipping, resulting in slow aggregation. Conversely, if the clipping threshold is set too high, it means that the added noise is insufficient to protect privacy, thus increasing the leakage of sensitive information. Due to the significant differences in gradient changes across different layers and their continuous variations during training, determining a fixed clipping threshold close to the optimal value becomes challenging.

Therefore, instead of inputting a fixed clipping threshold for global training, it is more effective to calculate the clipping threshold for each batch individually. Currently, there are two main forms of clipping in differential privacy: value-based clipping and norm-based clipping. Compared to value-based clipping, norm-based clipping allows more information to be retained within the gradient by scaling the gradient vector. Thus, this study adopted norm-based personalized clipping thresholds, defined by the following formula:

$$C = median(\|\omega_1\|_2, ..., \|\omega_i\|_2) \tag{6}$$

where C represents the median of the $l2$ norms of the gradients, and ω_i represents the parameters of each dimension in the model.

By employing this dual differential privacy method, each client is designed with personalized differential privacy parameters during local training. This approach ensures that the differential privacy parameters vary not only across different clients but also within the same client during each training session, thereby maximizing personalized differential privacy protection.

4.3. One-Shot Federated Learning

The traditional federated learning paradigm is illustrated in Figure 3. In this paradigm, clients' local models are frequently transmitted over insecure communication channels, significantly increasing the risk of privacy breaches. Consequently, some studies have improved the traditional federated learning paradigm. Xin et al. modified the traditional federated learning paradigm by adopting a serial federated learning approach to train GAN models. In this approach, each client trains the global model sequentially, reducing the number of times that training data is accessed [29]. Zhang et al. found that the traditional federated learning paradigm requires extensive communication interactions, leading to significant communication overhead and increasing the model's vulnerability to malicious attacks [41].

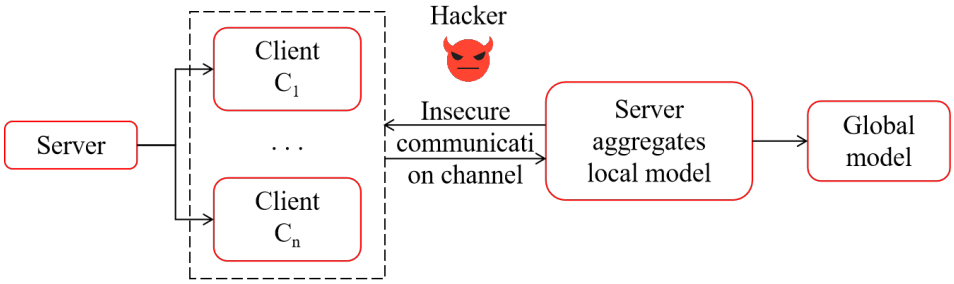


Figure 3. Traditional federated learning paradigm.

Inspired by these insights, this study designed the one-shot federated learning training paradigm for GAN models, as shown in Figure 4. In this one-shot federated learning paradigm, participating clients locally train their respective GAN models and implement differential privacy for privacy protection. Subsequently, the privacy-protected models are uploaded to the server for aggregation to obtain the final global model $\theta = \sum_{i=1}^{k} \frac{N_i}{N} \theta^i$, where N_i represents the number of data samples held by the i-th client, N is the total number of data samples across all clients, and k denotes the number of participating clients. Throughout the training process, the server and clients interact only once. Although clients' local models are still transmitted through insecure communication channels, the need for frequent interactions with the server is eliminated, significantly reducing the risk of privacy breaches. This effectively balances communication and computation costs while achieving privacy protection.

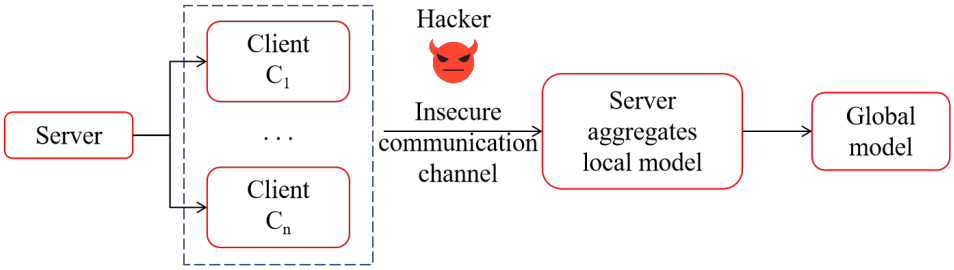


Figure 4. One-shot federated learning paradigm.

4.4. Global Algorithm Design

Algorithm 1 provides a detailed introduction to the training process of OFPP-GAN, incorporating steps to implement a dual personalized differential privacy algorithm. Table 1 provides a legend for notations in Algorithm 1. As stated in line 16, the clipping threshold is personalized and set to the median of the $l2$ norms of the gradients in the current training round. The personalized noise method described in lines 17–21 is a key technique in OFPP-GAN, allowing the noise scale to be adjusted during training based on gradient changes, thereby effectively adding noise. Due to the post-processing property of differential privacy, any samples generated by the generator model enjoy the same privacy guarantee. Through one-shot federated learning, the final global model can be obtained after the server aggregates the model once.

Table 1. Legend for notations in Algorithm 1.

Notation	Definition
x_i	Real data sample
u_i	Synthetic sample
σ	Noise scale
C	Clipping threshold
lr	Learning rate
T	Number of iterations per client
P	Set of data-holders
$\sigma_{\min}$	Minimum threshold for noise scale
θ	Generator parameters
w	Discriminator parameters
g_w	Gradient of the discriminator
g_θ	Gradient of the generator
n	Number of samples
RMSProp	Root Mean Square Propagation optimizer

4.5. Differential Privacy Constraint Design

In this section, we theoretically demonstrate that our proposed OFPP-GAN scheme satisfies the definition of differential privacy under certain constraints.

Consider two adjacent datasets D_i and D_i', where the only difference between the two datasets is a single data point. The sensitivity for the i-th client can be calculated as follows:

$$\Delta s^{D_i} = \frac{2C}{|D_i|} \tag{7}$$

where Δs^{D_i} represents the sensitivity of the dataset for the ith client, C denotes the clipping threshold, and $|D_i|$ is the size of the ith client's dataset.

Algorithm 1 One-shot Federated Personalized Protection–Generative Adversarial Network (OFPP-GAN).

Input: Examples $\{x_1, x_2, \ldots, x_n\}$, noise scale σ, clip norm bound C, learning rate lr, number of iterations per client T, the set of data-holders P, minimum threshold for sigma $\sigma_{\min}$
Output: Differentially Private generator G with parameters θ
Server executes:
for each client $k \in P$ in parallel **do**
  $\theta^k \leftarrow$ **ClientUpdate**(θ_0, ω_0, k) // Update parameters on the client
  $\theta \leftarrow \sum_{i=1}^{k} \frac{N_i}{N} \theta^i$ // Aggregate client updates
end for
ClientUpdate(θ_0, ω_0, k)**:**
for $t = 1, \ldots, T$ **do**
  Discriminator Step:
  Generate synthetic samples $\{u_1, u_2, \ldots, u_n\}$ using Generator $\theta^{(t-1)}$
  Sample $\{x_1, x_2, \ldots, x_n\} \overset{i.i.d}{\sim} D$ from the datasets
  For each i, $g_w(x_i) \leftarrow \nabla_{w^{(t)}}(f_{w^{(t-1)}}(x_i))$ // Calculate gradient on real data
  For each i, $g_w(u_i) \leftarrow \nabla_{w^{(t)}}(f_{w^{(t-1)}}(u_i))$ // Calculate gradient on fake data
  $g_w \leftarrow \frac{1}{n}(\sum_{i=1}^{n}(g_w(x_i) - g_w(u_i))$ // Compute the difference between gradients
  $C \leftarrow \text{median}(\|w_1\|_2, \ldots, \|w_i\|_2)$ // Personalized clipping threshold selection
  for i in all dimensions **do** // Personalized noise addition
    $\sigma_{l_i} \leftarrow \frac{g_w}{\sqrt{E[g^2]_t + \tau}}\sigma$
    $\sigma_{l_i} \leftarrow \max(\sigma_{min}, \sigma_{l_i})$
    $\overline{g_w^i} \leftarrow g_w^i / \max(1, \frac{\|g_w^i\|_2}{C}) + N(0, C^2\sigma_{l_i}^2 I)$
  end for
  $w^{(t)} \leftarrow w^{(t-1)} + lr \cdot \text{RMSProp}(w^{(t)}, g_w)$ // Update discriminator's weights
  $w^{(t)} \leftarrow \text{clip}(w^{(t)}, -c_p, c_p)$ // c_p is the clipping parameter for weight updates
  Generator Step:
  Generate synthetic samples $\{u_1, u_2, \ldots, u_n\}$ using Generator $\theta^{(t-1)}$
  $g_\theta \leftarrow \nabla_{\theta^{(t-1)}} \frac{1}{n} \sum_{i=1}^{n} f_{w^{(t)}}(u_i)$ // Calculate gradient for generator
  $\theta^{(t)} \leftarrow \theta^{(t-1)} + lr \cdot \text{RMSProp}(\theta^{(t)}, g_\theta)$ // Update generator's weights
end for
return $\theta^{(t)}$ // Return the updated generator parameters

Based on the above analysis, in order to achieve minimal global sensitivity, this study calculates the standard deviation of the added Gaussian noise according to the sensitivity calculation formula. As can be seen from the definition of differential privacy:

$$\left| \ln \frac{\Pr[A_p(D') = S]}{\Pr[A_p(D) = S]} \right| \leq \varepsilon \tag{8}$$

where ε denotes the privacy budget in differential privacy.

We set μ_0 to represent the Gaussian distribution $N(0, \sigma^2)$, and μ_1 to represent the mixture probability density function $qN(\Delta s, \sigma^2) + (1-q)N(0, \sigma^2)$, μ_0 and μ_1 are defined by Equations (9) and (10):

$$\mu_0(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2\sigma^2}\right) \tag{9}$$

$$\mu_1(z) = \frac{q}{\sqrt{2\pi}} \exp\left(-\frac{(z+\Delta s)^2}{2\sigma^2}\right) + \frac{1-q}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2\sigma^2}\right) \tag{10}$$

where σ represents the standard deviation of the added Gaussian noise, Δs denotes the sensitivity of the data, and q is the client sampling rate.

In the context of differential privacy, z is typically related to the noise added to the data to ensure privacy. The noise is sampled from a Gaussian distribution, and z represents a possible realization of this noise. Substituting Equations (9) and (10) into Equation (8) and summing over T rounds, we obtain the following formula:

$$\sum_{i=1}^{T} \ln\left(1 - q + q\exp\left(-\frac{2z\Delta s + \Delta s^2}{2\sigma^2}\right)\right) = \ln\left(\prod_{i=1}^{T}\left(1 - q + q\exp\left(-\frac{2z\Delta s + \Delta s^2}{2\sigma^2}\right)\right)\right) \quad (11)$$

where T represents the number of training rounds.

Combining the results of Equations (8) and (11), considering the independence of added noise, we can obtain:

$$T\ln(1 - q + q\exp\left(-\frac{2z\Delta s + \Delta s^2}{2\sigma^2}\right)) \geq -\varepsilon \quad (12)$$

Solving for z in terms of ε yields:

$$z \leq -\frac{\sigma^2}{\Delta s^D}\ln\left(\frac{\exp\left(-\frac{\varepsilon}{T}\right) - 1}{q} + 1\right) - \frac{\Delta s^D}{2} \quad (13)$$

where Δs^D represents the global sensitivity.

For ease of calculation, let $b = -\frac{T}{\epsilon}\ln\left(\frac{\exp\left(-\frac{\epsilon}{T}\right)-1}{q} + 1\right)$. Thus,

$$-\frac{\varepsilon b}{T} = \ln\left(\frac{\exp\left(-\frac{\epsilon}{T}\right) - 1}{q} + 1\right) \quad (14)$$

The values of ε and T must satisfy the following conditions:

$$\varepsilon < -T\ln(1 - q) \quad (15)$$

Then, modifying Equation (13), we get:

$$z \leq \frac{\sigma^2 b\varepsilon}{T\Delta s^D} - \frac{\Delta s^D}{2} \quad (16)$$

From the tail constraint [42], we obtain the formula regarding δ:

$$\Pr[z > \beta] \leq \delta \quad (17)$$

$$\Pr[z > \beta] \leq \frac{\sigma}{\beta\sqrt{2\pi}}\exp\left(-\frac{\beta^2}{2\sigma^2}\right) \quad (18)$$

where β represents the tail bound of the Gaussian distribution, and δ is the relaxation parameter in differential privacy.

Transforming Equations (17) and (18), we obtain:

$$\frac{\sigma}{\beta\sqrt{2\pi}}\exp\left(-\frac{\beta^2}{2\sigma^2}\right) < \delta \quad (19)$$

Transforming the form, we obtain Equation (20):

$$\ln\frac{\beta}{\sigma} + \frac{\beta^2}{2\sigma^2} > \ln\left(\frac{1}{\delta}\sqrt{\frac{2}{\pi}}\right) \quad (20)$$

If $\frac{b\varepsilon}{T} \in (0,1)$, substituting $\sigma = \frac{c\Delta s_D T}{b\varepsilon}$ solves the formula (20) to obtain (21):

$$\begin{cases} c^2 \geq 2\ln\left(\frac{1.25}{\delta}\right) \\ \varepsilon < -T\ln\left(1-q+\frac{q}{e}\right) \end{cases} \tag{21}$$

where c denotes a parameter for adjusting the noise scale.

If $\frac{b\varepsilon}{T} > 1$, we can adjust the value of c to adjust the noise size to satisfy the definition of differential privacy. The overall noise formula is as follows:

$$\sigma \geq \frac{2cT}{b\varepsilon m} \tag{22}$$

where m represents the number of clients or the total dataset size.

5. Experiments

5.1. Experimental Settings

5.1.1. Training Environment

The training environment for OFPP-GAN utilized the PyTorch framework, with hardware parameters, including an NVIDIA GTX Force RTX 3080 GPU, Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, 16 GB RAM, and Windows 10 64-bit operating system based on an x64 processor system.

5.1.2. Evaluation Indicators

Inception Score (IS): Salimans et al. proposed using the IS to evaluate the quality of data generated by GAN [43]. The IS is defined as:

$$s(G) = \exp\left(E_{x\sim G(z)} KL(\Pr(y|x) \| \Pr(y))\right) \tag{23}$$

where x represents samples generated by the generator G, $\Pr(y|x)$ represents the class distribution predicted by a pre-trained classifier that assigns label y to sample x, and $\Pr(y) = \int \Pr(y|x = G(z))dz$ represents the average class distribution of the input images. By measuring the KL divergence between these two distributions, $s(G)$ can assess the quality and diversity of the generated data. Therefore, a higher IS indicates better quality and diversity of the generated images, proving the usability of the GAN model.

Frechet Inception Distance (FID): FID is a metric used to evaluate the difference between the distribution of synthetic images generated by a model and the distribution of real data. It was proposed by Martin Heusel et al. in 2017 and is widely used as an evaluation metric [44]. FID is defined as:

$$FID(r,g) = \|\mu_r - \mu_g\|^2 + Tr(\sigma_r + \sigma_g - 2(\sigma_r\sigma_g)^{1/2}) \tag{24}$$

where μ_r and μ_g are the feature means of the real and synthetic data, respectively, and σ_r and σ_g are the feature covariance matrices of the real and synthetic data, respectively. A lower FID indicates that the synthetic data distribution is closer to the real data distribution, implying higher quality and better diversity of the synthetic images.

Classification Accuracy (ACC): In this study, a CNN classification model was used to test the ACC. The CNN classification model is a deep learning model that uses convolutional neural networks to extract image features and perform classification, widely applied in image recognition and computer vision fields.

5.1.3. Datasets

MNIST: The MNIST dataset consists of 70,000 images of 10 different handwritten digit classes, including 60,000 training images with corresponding labels and 10,000 test images with labels. Each image is 28 × 28 pixels in size. Figure 5 shows some real data from the MNIST dataset.

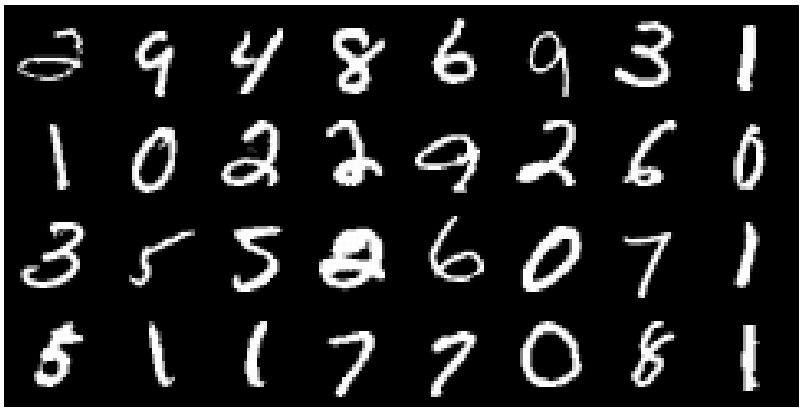


Figure 5. MNIST real images.

FashionMNIST: The FashionMNIST dataset consists of 70,000 images representing 10 different fashion items. This dataset includes 60,000 training images with labels and 10,000 test images with labels. Each image in the dataset is 28 × 28 pixels, showcasing various clothing items. Figure 6 shows some real data from the FashionMNIST dataset.



Figure 6. FashionMNIST real images.

5.2. Compare Algorithms

This study compares the OFPP-GAN algorithm with the following algorithms:

RealData: The RealData comparison algorithm uses real data to test various evaluation metrics. The results of this algorithm can analyze and derive the optimal values for each evaluation metric.

GAN: The GAN comparison algorithm uses a standard Generative Adversarial Network for training without any privacy protection measures. The results of this algorithm are crucial for comparing the impact of differential privacy on the quality of generated data. The results of this algorithm are crucial for comparing the impact of differential privacy on the quality of generated data.

DPGAN: The DPGAN comparison algorithm is a classic algorithm in the problem of differentially private generative adversarial networks. This algorithm ensures differential privacy protection for sensitive data during the training process through gradient clipping and noise perturbation. The results of this algorithm can demonstrate the advantages of OFPP-GAN compared to the traditional DPGAN.

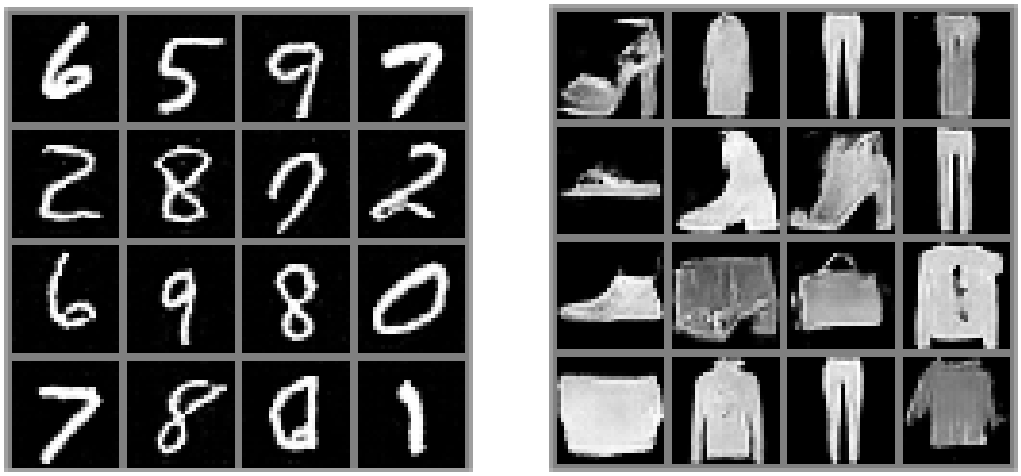
GANobfuscator: The GANobfuscator comparison algorithm is an excellent solution in the field of privacy-preserving data publishing. This approach proposes using a small portion of publicly available datasets to calculate the average gradient clipping threshold, and then applying the gradient clipping thresholds obtained from the public data to the training process of the real data. Additionally, differential privacy noise is introduced to

achieve privacy protection. The results of this algorithm can demonstrate the advantages of OFPP-GAN over the improved DPGAN.

FedDPGAN: The FedDPGAN comparison algorithm applies the DPGAN scheme in a federated learning scenario. Federated learning effectively mitigates the "data island" problem in traditional training and significantly enhances the performance of the trained model. The results of this algorithm can demonstrate the advantages of OFPP-GAN in federated learning scenarios.

5.3. Experimental Results and Analysis

Figure 7a,b show the synthetic images generated using the OFPP-GAN scheme for the MNIST and FashionMNIST datasets. It can be observed that the quality of the synthetic images is not significantly affected visually. To further validate the usability of the synthetic images, this study tested relevant evaluation metrics such as IS, FID, and ACC.



(**a**) (**b**)

Figure 7. Fake images of MNIST and FashionMNIST. (**a**) MNIST fake images; (**b**) FashionMNIST fake images.

Table 2 presents the trade-offs between privacy protection and model performance for different algorithms on the MNIST and FashionMNIST datasets, particularly under varying privacy budgets. By comparing the IS, FID, and ACC of each algorithm, the impact of privacy-preserving measures on model performance becomes evident. In the MNIST dataset, RealData, serving as a baseline, shows the best IS and ACC. Although the standard GAN model generates high-quality images, its ACC significantly decreases compared to RealData, indicating a performance gap despite generating images close to real data. After introducing differential privacy mechanisms, DPGAN shows a marked decline in IS and ACC, while FID increases, highlighting the trade-off between privacy and performance. However, as the privacy budget increases, DPGAN's performance improves, suggesting that a more lenient privacy budget can mitigate performance loss. FedDPGAN and GANobfuscator, as enhanced privacy-preserving variants, outperform DPGAN in IS and FID but still lag behind the standard GAN in ACC. OFPP-GAN demonstrates a good balance between privacy protection and model performance. Under a lower privacy budget, OFPP-GAN outperforms other privacy-preserving algorithms in IS and FID, with ACC close to that of the standard GAN. With a higher privacy budget, its performance becomes even more outstanding, maintaining high-quality data generation while ensuring strong privacy protection. The trend on the FashionMNIST dataset is similar to that of the MNIST dataset. OFPP-GAN, by personalizing the computation of differential privacy parameters, consistently performs well across different datasets and privacy budgets, finding a better balance between privacy and performance, and highlighting its advantages.

Table 2. Comparison of different algorithms on MNIST and FashionMNIST datasets.

Dataset	Algorithm	IS	FID	ACC	IS	FID	ACC
		$(4, 10^{-5}) - DP$			$(10, 10^{-5}) - DP$		
MNIST	RealData	9.88	-	99.6%	9.88	-	99.6%
	GAN	8.83	46.07	84%	8.83	46.07	84%
	DPGAN	7.37	140.32	66%	8.12	87.69	73%
	FedDPGAN	7.59	63.82	60%	7.84	55.88	67%
	GANobfuscator	8.06	67.26	71%	8.49	53.27	75%
	OFPP-GAN	8.14	60.66	78%	8.62	50.72	81%
FashionMNIST	RealData	9.68	-	92.27%	9.68	-	92.27%
	GAN	8.48	57.81	81%	8.48	57.81	82%
	DPGAN	7.24	189.38	55%	7.67	124.51	68%
	FedDPGAN	7.63	135.23	71%	8.00	72.29	75%
	GANobfuscator	7.01	136.54	59%	7.72	132.13	62%
	OFPP-GAN	7.79	94.97	73%	7.89	76.42	77%

Note: IS: Inception Score, FID: Fréchet Inception Distance, ACC: Accuracy, *DP*: Differential Privacy.

Subsequently, this study tested the impact of different numbers of clients on OFPP-GAN, with experimental data shown in Figure 8. From this figure, it can be observed that as the number of clients decreases, the IS value of the synthetic images generated by the generator increases, indicating that the synthetic images are closer to the original dataset, and the performance of the trained generator model is higher. Therefore, it can be inferred that this scheme is more suitable for scenarios with fewer federated learning participants; the fewer federated learning participants, the higher the quality of the generator model trained using OFPP-GAN.

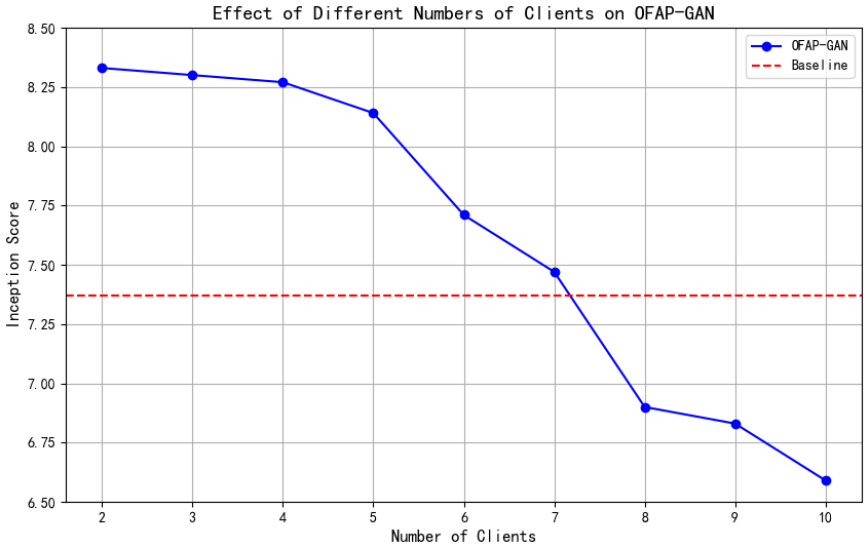


Figure 8. Influence of different number of clients on OFPP-GAN.

According to Section 5.1.3, IS can be used to measure the quality and diversity of the generated samples. We evaluate the convergence of the model by observing changes in the IS value, with specific experimental results shown in Figure 9. Figure 9 shows that under the same privacy budget, the convergence speed of OFPP-GAN is significantly better than the traditional federated learning algorithm FedDPGAN. Additionally, the synthetic images generated by OFPP-GAN are significantly superior in quality and diversity compared to those generated by FedDPGAN. This indicates that OFPP-GAN can more efficiently generate higher-quality synthetic data while maintaining privacy protection.

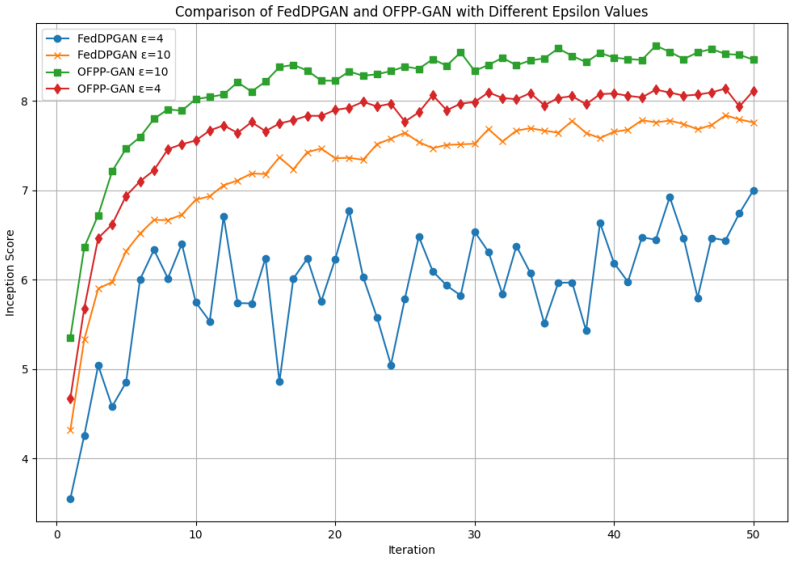


Figure 9. Comparison of training efficiency.

Moreover, this study compared three strategies: 50 rounds of local training, 50 rounds of federated learning with each client performing one round of local training, each client performing 50 rounds of local training followed by one round of federated learning, and each client performing 50 rounds of local training followed by one round of federated learning. The training times obtained from these methods are shown in Figure 10.

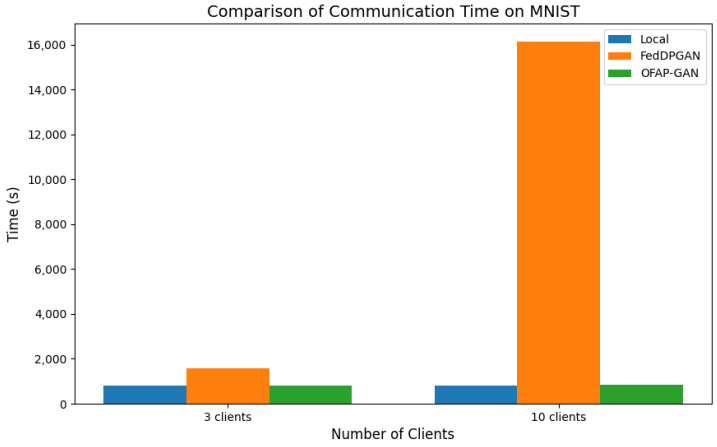


Figure 10. Comparison of communication efficiency.

From the results, it can be observed that although the proposed OFPP-GAN scheme incurs higher communication overhead compared to local training, it significantly reduces communication costs compared to traditional federated learning schemes. This reduction can be attributed to the extensive communication overhead typically involved in server–client interactions in traditional federated learning settings. The one-time federated learning method alleviates this issue by reducing communication losses associated with multiple interactions. Additionally, the proposed scheme shows minimal variation in local computation overhead. This is because traditional federated learning schemes require 50 rounds of interaction with local training, while OFPP-GAN only requires one round of interaction with 50 rounds of local training. Overall, the computational overhead of these two methods is comparable throughout the process. Furthermore, by reducing communication overhead and minimizing client–server interactions, OFPP-GAN reduces the likelihood of system vulnerabilities to malicious attacks during communication, thereby enhancing privacy protection.

Additionally, this study explored the impact of different numbers of clients on communication costs in a federated learning scenario. It can be observed that as the number of clients increases from 3 to 10, the communication overhead generated during training using traditional federated learning also increases. In contrast, the communication costs generated by the proposed OFPP-GAN method remain relatively stable, demonstrating the feasibility of OFPP-GAN.

Finally, we tested the performance of OFPP-GAN under various data heterogeneity scenarios in federated learning, using different Dirichlet coefficients (α). The experimental results are shown in Figure 11. The results indicate that as the Dirichlet coefficient increases, representing decreasing data heterogeneity, the FID (Frechet Inception Distance) values for all models generally improve, with a lower FID indicating better performance. Specifically, FedDPGAN demonstrates a more consistent performance across different heterogeneity levels, with a steadier decrease in FID as α increases. While OFPP-GAN outperforms FedDPGAN and DPGAN in more homogeneous data scenarios (higher α), it struggles with higher FID values in more heterogeneous settings (lower α), indicating a sensitivity to data heterogeneity. OFPP-GAN shows a significant drop in FID as α increases, demonstrating improved generative quality in less heterogeneous data settings.

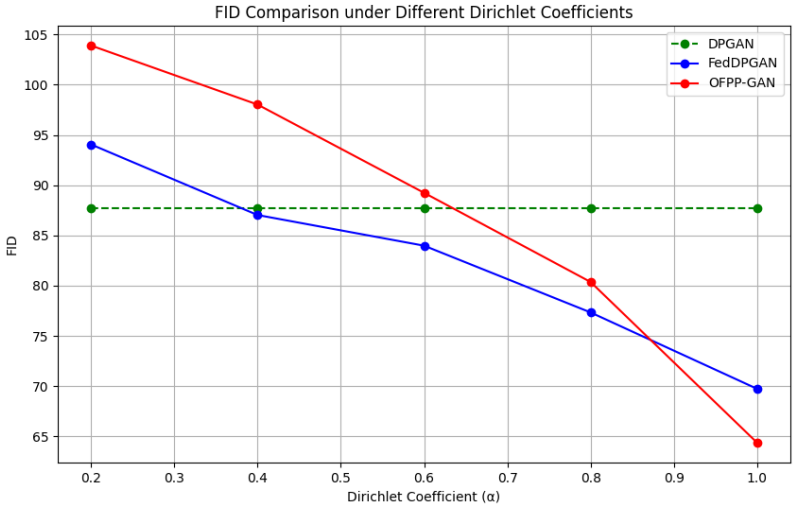


Figure 11. Comparison of different data heterogeneity scenarios in federated learning.

6. Conclusions

In the context of one-shot federated learning, this paper introduces a dual personalized differential privacy scheme called OFPP-GAN, which provides personalized privacy protection for GAN models. By designing personalized noise scales and clipping thresholds, the differential privacy parameters are dynamically adjusted to balance model utility and privacy protection. Additionally, the use of the one-shot federated learning paradigm effectively reduces the frequency of data transmission, further enhancing privacy protection. Through experiments, it can be observed that OFPP-GAN not only demonstrates advantages in model performance but also effectively reduces communication overhead.

However, this one-shot federated learning paradigm is limited in its applicability to federated learning scenarios. In non-IID data distribution scenarios, it is prone to "model drift", making the generation results of the GAN model more biased toward clients with larger data volumes or distinct data characteristics. Therefore, future work will focus on the research of GAN model training in federated learning scenarios with data heterogeneity.

Author Contributions: Z.J. was responsible for the coding and algorithm design as well as the overall writing of the manuscript. C.Z. and H.T. were responsible for providing resources and funding assistance. Z.C. was responsible for conducting part of the comparative experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under grants (61802134, 61872154, 61972166, and 61972168), the Fundamental Research Funds for the Central Universities (ZQN-811), and the Natural Science Foundation of Fujian Province of China under grant 2020J05059.

Data Availability Statement: The data that support the findings of this study are available on request from the corresponding author, C.Z., upon reasonable request.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*; Curran Associates, Inc.: Boston, MA, USA, 2014.
2. Ciresan, D.; Giusti, A.; Gambardella, L.; Schmidhuber, J. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems 25*; Curran Associates, Inc.: Boston, MA, USA, 2012.
3. Hinton, G.; Deng, L.; Yu, D.; Dahl, G.E.; Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.N.; et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
4. Zhu, J.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
5. Huang, J.; Wu, C. Privacy leakage in gan enabled load profile synthesis. In Proceedings of the 2022 IEEE Sustainable Power and Energy Conference (iSPEC), Perth, Australia, 4–7 December 2022; pp. 1–5.
6. Dwork, C. Differential privacy. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 1–12.
7. Xie, L.; Lin, K.; Wang, S.; Wang, F.; Zhou, J. Differentially private generative adversarial network. *arXiv* **2018**, arXiv:1802.06739.
8. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Boston, MA, USA, 24–25 October 2017; pp. 1273–1282.
9. Zhu, T.; Li, G.; Zhou, W.; Philip, S.Y. Differentially private data publishing and analysis: A survey. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 1619–1638. [CrossRef]
10. Niu, B.; Chen, Y.; Wang, B.; Wang, Z.; Li, F.; Cao, J. Adapdp: Adaptive personalized differential privacy. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
11. Wei, K.; Deng, C.; Yang, X.; Li, M. Incremental embedding learning via zero-shot translation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 10254–10262.
12. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the gan: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
13. Fredrikson, M.; Lantz, E.; Jha, S.; Lin, S.; Page, D.; Ristenpart, T. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14), San Diego, CA, USA, 20–22 August 2014; pp. 17–32.
14. Horváth, G.; Kerekes, K.; Nyitrai, V.; Balazs, G.; Berisha, H.; Herczeg, G. Exploratory behaviour divergence between surface populations, cave colonists and a cave population in the water louse, asellus aquaticus. *Behav. Ecol. Sociobiol.* **2023**, *77*, 15. [CrossRef]
15. Huang, J.; Huang, Q.; Mou, G.; Wu, C. Dpwgan: High-quality load profiles synthesis with differential privacy guarantees. *IEEE Trans. Smart Grid* **2022**, *14*, 3283–3295. [CrossRef]
16. Pan, K.; Gong, M.; Gao, Y. Privacy-enhanced generative adversarial network with adaptive noise allocation. *Knowl.-Based Syst.* **2023**, *272*, 110576. [CrossRef]
17. Gwon, H.; Ahn, I.; Kim, Y.; Kang, H.J.; Seo, H.; Choi, H.; Cho, H.N.; Kim, M.; Han, J.; Kee, G.; et al. Ldp-gan: Generative adversarial networks with local differential privacy for patient medical records synthesis. *Comput. Biol. Med.* **2024**, *168*, 107738. [CrossRef]
18. Jordon, J.; Yoon, J.; Schaar, M.V.D. Pate-gan: Generating synthetic data with differential privacy guarantees. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
19. Chen, J.; Wang, W.H.; Gao, H.; Shi, X. Par-gan: Improving the generalization of generative adversarial networks against membership inference attacks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, 14–18 August 2021; pp. 127–137.
20. Yan, X.; Cui, B.; Xu, Y.; Shi, P.; Wang, Z. A method of information protection for collaborative deep learning under gan model attack. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *18*, 871–881. [CrossRef]

21. Chen, D.; Yu, N.; Zhang, Y.; Fritz, M. Gan-leaks: A taxonomy of membership inference attacks against generative models. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, 9–13 November 2020; pp. 343–362.
22. Xu, C.; Ren, J.; Zhang, D.; Zhang, Y.; Qin, Z.; Ren, K. Ganobfuscator: Mitigating information leakage under gan via differential privacy. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2358–2371. [CrossRef]
23. Mirjalili, V.; Raschka, S.; Ross, A. Privacynet: Semi-adversarial networks for multi-attribute face privacy. *IEEE Trans. Image Process.* **2020**, *29*, 9400–9412. [CrossRef]
24. Qiao, F.; Li, Z.; Kong, Y. A privacy-aware and incremental defense method against gan-based poisoning attack. *IEEE Trans. Comput. Soc. Syst.* **2023**, *11*, 1708–1721. [CrossRef]
25. Xiong, Z.; Li, W.; Han, Q.; Cai, Z. Privacy-preserving auto-driving: A gan-based approach to protect vehicular camera data. In Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM), Beijing, China, 8–11 November 2019; pp. 668–677.
26. Chai, X.; Wang, Y.; Chen, X.; Gan, Z.; Zhang, Y. Tpe-gan: Thumbnail preserving encryption based on gan with key. *IEEE Signal Process. Lett.* **2022**, *29*, 972–976. [CrossRef]
27. Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; Qi, H. Beyond inferring class representatives: User-level privacy leakage from federated learning. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 2512–2520.
28. Zhang, L.; Shen, B.; Barnawi, A.; Xi, S.; Kumar, N.; Wu, Y. Feddpgan: Federated differentially private generative adversarial networks framework for the detection of covid-19 pneumonia. *Inf. Syst. Front.* **2021**, *23*, 1403–1415. [CrossRef]
29. Xin, B.; Yang, W.; Geng, Y.; Chen, S.; Wang, S.; Huang, L. Private fl-gan: Differential privacy synthetic data generation based on federated learning. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2927–2931.
30. Cao, X.; Sun, G.; Yu, H.; Guizani, M. Perfed-gan: Personalized federated learning via generative adversarial networks. *IEEE Internet Things J.* **2022**, *10*, 3749–3762. [CrossRef]
31. Hardy, C.; Merrer, E.L.; Sericola, B. Md-gan: Multi-discriminator generative adversarial networks for distributed datasets. In Proceedings of the 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), Rio de Janeiro, Brazil, 20–24 May 2019; pp. 866–877.
32. Rasouli, M.; Sun, T.; Rajagopal, R. Fedgan: Federated generative adversarial networks for distributed data. *arXiv* **2020**, arXiv:2006.07228.
33. Mugunthan, V.; Gokul, V.; Kagal, L.; Dubnov, S. Bias-free fedgan: A federated approach to generate bias-free datasets. *arXiv* **2021**, arXiv:2103.09876.
34. Zhao, Z.; Birke, R.; Kunar, A.; Chen, L.Y. Fed-tgan: Federated learning framework for synthesizing tabular data. *arXiv* **2021**, arXiv:2108.07927.
35. Guha, N.; Talwalkar, A.; Smith, V. One-shot federated learning. *arXiv* **2019**, arXiv:1902.11175.
36. Kasturi, A.; Ellore, A.R.; Hota, C. Fusion learning: A one shot federated learning. In Proceedings of the Computational Science–ICCS 2020: 20th International Conference, Amsterdam, The Netherlands, 3–5 June 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 424–436.
37. Song, R.; Liu, D.; Chen, D.Z.; Festag, A.; Trinitis, C.; Schulz, M.; Knoll, A. Federated learning via decentralized dataset distillation in resource-constrained edge environments. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–10.
38. Li, Q.; He, B.; Song, D. Practical one-shot federated learning for cross-silo setting. *arXiv* **2020**, arXiv:2010.01017.
39. Fredrikson, M.; Jha, S.; Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015; pp. 1322–1333.
40. Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P.S.; Zhang, X. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 235. [CrossRef]
41. Zhang, J.; Chen, C.; Li, B.; Lyu, L.; Wu, S.; Ding, S.; Shen, C.; Wu, C. Dense: Data-free one-shot federated learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 21414–21428.
42. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [CrossRef]
43. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the ICML'17: Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; pp. 2642–2651.
44. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; Volume 30.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.