electronics

MDPI

Article

SPM-FL: A Federated Learning Privacy-Protection Mechanism Based on Local Differential Privacy

Zhiyan Chen and Hong Zheng *

School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China; acousma527@gmail.com

* Correspondence: zhenghong@ccut.edu.cn

Abstract: Federated learning is a widely applied distributed machine learning method that effectively protects client privacy by sharing and computing model parameters on the server side, thus avoiding the transfer of data to third parties. However, information such as model weights can still be analyzed or attacked, leading to potential privacy breaches. Traditional federated learning methods often disturb models by adding Gaussian or Laplacian noise, but under smaller privacy budgets, the large variance of the noise adversely affects model accuracy. To address this issue, this paper proposes a Symmetric Partition Mechanism (SPM), which probabilistically perturbs the sign of local model weight parameters before model aggregation. This mechanism satisfies strict ϵ-differential privacy, while introducing a variance constraint mechanism that effectively reduces the impact of noise interference on model performance. Compared with traditional methods, SPM generates smaller variance under the same privacy budget, thereby improving model accuracy and being applicable to scenarios with varying numbers of clients. Through theoretical analysis and experimental validation on multiple datasets, this paper demonstrates the effectiveness and privacy-protection capabilities of the proposed mechanism.

Keywords: federated learning; local differential privacy; privacy protection; deep learning

check for updates

Citation: Chen, Z.; Zheng, H. SPM-FL: A Federated Learning Privacy-Protection Mechanism Based on Local Differential Privacy. *Electronics* **2024**, *13*, 4091. https://doi.org/10.3390/electronics13204091

Academic Editor: Andreas Mauthe

Received: 25 September 2024
Revised: 10 October 2024
Accepted: 15 October 2024
Published: 17 October 2024

1. Introduction

Federated learning is a distributed machine learning approach that allows institutions to collaboratively train models without sharing local data. This technology overcomes the limitations of data silos, enabling multiple participants to cooperate while preserving data privacy, thus improving the predictive performance and accuracy of the models [1–3]. Researchers are working to integrate various techniques such as secure multiparty computation, homomorphic encryption, group learning, and differential privacy with federated learning to enhance data privacy protection. For example, Ma et al. [4] proposed a federated learning scheme that combines multi-key homomorphic encryption to optimize computational efficiency while ensuring data privacy. In contrast, Park et al. [5] adopted a method of directly applying homomorphic encryption to model parameters, allowing the central server to perform computations on encrypted data without decryption, thereby enhancing the security and practicality of federated learning. Additionally, blockchain technology has also been widely applied to federated learning to enhance privacy and data security. Specifically, AI-enhanced blockchain technology can improve data security and transparency. For instance, Reference [6] reviews the opportunities for applications combining blockchain and AI, while Reference [7] explores the use of blockchain in enhancing federated learning security and discusses the challenges it faces. Although these techniques provide assurance for data that is "computable but not visible", they also significantly increase the computational and communication burden on the system. Secure multiparty computation relies on complex communication protocols, while homomorphic encryption requires a large number of encryption operations, and clients still face the risk of privacy leakage during parameter uploads and downloads [5,8–13]. While blockchain technology improves data security with its decentralized and tamper-resistant properties, its high computational and storage costs are significant drawbacks. Blockchain networks require validation and consensus for each transaction, which leads to increased system latency and energy consumption, especially in resource-constrained IoT devices. Moreover, as the number of nodes increases, the complexity of the consensus process can affect scalability and system efficiency.

In contrast, differential privacy has emerged as an important research direction in federated learning due to its simplicity and robust privacy-protection capabilities. In particular, Local Differential Privacy (LDP) offers higher security by adding noise locally without relying on the trustworthiness of the central server, effectively preventing server-side privacy breaches. However, LDP also faces several challenges, particularly in the selection of the number of clients. When the number of clients is small, there is often a lack of data diversity, which reduces the model's generalization ability and slows down the convergence rate. Moreover, the individual data of a small number of participants is more easily inferred, increasing the privacy risk. Conversely, while a larger number of clients can provide richer data distribution, it significantly increases communication overhead, affecting system efficiency, especially under bandwidth-constrained conditions. Additionally, the participation of more clients can lead to instability in training, as anomalous data from a single client can substantially impact the overall model. Therefore, balancing scenarios with both few and many clients while maintaining stable and efficient performance presents a major challenge.

In addition, federated learning faces communication cost challenges, particularly in Internet of Things (IoT) environments, where communication costs between devices are high [14–16]. Federated learning relies on frequent parameter exchanges between clients and the server, which is especially noticeable on resource-constrained devices. While differential privacy techniques enhance privacy protection, they also increase communication overhead and training time. Therefore, how to reduce communication costs while effectively protecting privacy remains a significant challenge in federated learning.

In response to the challenges in the aforementioned FL models, the contributions of this paper are as follows:

- We propose a federated learning privacy-protection mechanism based on Local Differential Privacy (LDP)—the Symmetric Piecewise Mechanism (SPM). Unlike traditional noise-adding methods, our mechanism perturbs the sign of the weights probabilistically before uploading the model parameters, thereby ensuring strict differential privacy. We comprehensively consider the expected value and variance of the perturbed model weights and introduce a variance constraint mechanism. By limiting the bounds of the perturbation range, this mechanism minimizes the impact of the perturbation on model accuracy. This mechanism enables the use of a smaller privacy budget while ensuring the same level of privacy protection and preserving the utility of the aggregated model, which closely matches that of the original model. This reduces the impact of noise addition on model accuracy and communication overhead in the federated learning process.
- We have validated the usability and privacy-protection capabilities of the mechanism from both theoretical and empirical perspectives. From a theoretical standpoint, we conducted mathematical derivations for the proposed variance constraint mechanism, as well as the theorems and lemmas. In the experimental section, we used three datasets: MNIST, Fashion-MNIST, and CIFAR-10, with two network models applied to CIFAR-10. We conducted analysis in scenarios with varying numbers of clients, focusing on multiple dimensions such as the mechanism's usability, communication overhead, time consumption, and its ability to defend against DLG attacks. All derivations discussed in this paper can be found in the Appendix A.

The structure of this paper is as follows: Section 1 introduces the challenges in federated learning and privacy protection, as well as the main contributions of this work. Section 2 provides an overview of the advantages and disadvantages of existing methods and highlights the benefits of the proposed approach. Section 3 explains the fundamental concepts of federated learning and differential privacy. Section 4 presents a symmetric partitioning mechanism (SPM) based on localized differential privacy, detailing its implementation process, the cited lemmas and theorems, and providing a theoretical analysis of its privacy and utility. Section 5 discusses the experimental methods and results, including an analysis of three datasets, the usability of the mechanism, and its ability to defend against DLG attacks. Section 6 summarizes the main contributions and discusses future research directions.

2. Related Work

- Applications and Challenges of Differential Privacy in Federated Learning

Differential privacy is widely applied in practice, for example, when medical research centers develop COVID-19 diagnostic models that require collecting CT images of patients [17]. Local differential privacy emphasizes control by data owners, who add noise locally before uploading data to prevent privacy leaks. Tramèr et al. [18] argued that applying differential privacy to model parameters is more appropriate, but the high dimensionality of gradients in federated learning makes direct perturbation increase communication overhead. Zhu et al. [19] found that capturing uploaded gradient data can be used to reconstruct the training data. Yang et al. [20] successfully reconstructed facial images using an auxiliary training set. These studies suggest that protecting model parameters and gradients from attacks is a significant challenge.

The main challenges of federated learning include the selection of the number of clients and the impact of noise addition. The number of clients directly affects model performance and privacy protection: with fewer clients, there may be insufficient data diversity, which could reduce the model's generalization ability and convergence speed; with more clients, although the data is richer, the amount of training data per client decreases, affecting the training results. Furthermore, more clients increase communication overhead, reducing system efficiency. To protect privacy, noise is often added to the uploaded model parameters, which increases computational and communication overhead. This is especially problematic in bandwidth-limited environments, where frequent parameter exchanges can cause system delays and affect training efficiency. To address these issues, blockchain technology has been introduced into federated learning to enhance security and data protection. The blockchain-based federated learning framework proposed in Reference [6] analyzes how to ensure data privacy in a decentralized environment while reducing the risk of single points of failure. However, this approach faces challenges in scalability and high energy consumption when dealing with a large number of nodes, as the verification and consensus processes significantly increase the system's computational and communication burden.

Scholars have introduced adaptive ideas into differential privacy to address gradient adjustment [21–23]. Liu et al. [24] developed a differential privacy federated learning algorithm that adaptively adjusts the gradient clipping threshold, making gradient clipping more flexible in each communication round, thereby reducing the negative impact of unreasonable thresholds on model performance. In contrast, Shen et al. [25] proposed an algorithm called Performance-Enhanced Differential Privacy Federated Learning (PEDPFL), which uses regularization techniques to improve the robustness and generalization ability of the model, thereby enhancing the algorithm's performance under privacy protection. However, after processing the model gradients, these methods may result in slow global convergence and a lack of stability. They do not fundamentally solve the noise issues introduced by differential privacy but rather mitigate them from an external perspective. Therefore, addressing the noise added by differential privacy at its internal source has become a critical challenge that needs to be resolved.

In response, researchers have proposed several improvements. The two localized differential privacy schemes based on mean statistics proposed in the literature [26,27] demonstrated that their mechanisms produce less variance than traditional Laplacian noise under the same privacy budget. However, under smaller privacy budgets, directly perturbing the model parameters still results in larger variance, and the aggregated model may experience gradient explosion during backpropagation. Sun et al. [28], building on the work in [26], improved the LDP mechanism by considering the value range of each layer's weights for the first time, thereby reducing the large variance issues of previous mechanisms. However, in complex models, the adaptive adjustment of key parameters requires manual testing, leading to weaker compatibility. Ren et al. [29] proposed the PNPM mechanism, which selects the perturbation upper limit based on empirical testing and introduces a new variance calculation formula, resulting in smaller variance. However, this mechanism has yet to determine the optimal perturbation boundary, leaving substantial room for improvement and warranting further exploration.

Although these methods have made some progress in enhancing privacy protection and performance in federated learning, many limitations remain. In response to the aforementioned issues and the shortcomings of mechanisms [27–29], this paper proposes a novel Symmetric Piecewise Mechanism (SPM), which adapts to different client scenarios. The mechanism aims to better address communication overhead, time consumption, and attack defense challenges arising from the number of clients. Both theoretical and practical validations have been conducted. The following section analyzes the superiority of the SPM mechanism compared to other mechanisms, as detailed in Table 1.

Table 1. Comparison Analysis of the SPM mechanism and other federated learning privacy-protection mechanisms.

Comparison Dimension	**Comparison Mechanism**	**Other Mechanism Algorithms**	**SPM Mechanism**	**Advantages of the Proposed Mechanism**
Model Weight Perturbation Output Method and Perturbation Domain Selection	PM [27]	Weights are directly output as the perturbation domain $t^* \in [-C, C]$.	The absolute value of model weights is desensitized and multiplied by the perturbation coefficient $t^* \in [-C, -1] \cup [1, C]$, and then output.	The SPM mechanism significantly reduces the risk of model weights outputting zero by multiplying with perturbation parameters greater than one in absolute value, combined with symmetric inversion to avoid more complex scenarios. It represents an improvement over the PM mechanism.
Adaptive Weight Range Selection	Sun et al. [28]	Adaptive hierarchical perturbation based on the range of model weights.	Perturbation is applied to all positive and negative weights (excluding 0), with normalization tracking to achieve adaptive effects, making it suitable for both deep and shallow networks.	Sun et al. [28] pointed out that their mechanism has limited adaptability to complex models, requiring manual tuning and lacking adaptive capabilities. In contrast, the SPM mechanism can adapt to deep networks, ensuring error-free weight normalization, which is described in detail in the experimental section. This represents an improvement in adaptability over the mechanism proposed by Sun et al. [28]
Boundary Value Selection for the Perturbation Domain and the Variance Calculation Formula	PNPM [29]	Perturbation domain boundaries are set based on empirical data, and variance is calculated.	A variance constraint mechanism is proposed to theoretically limit the boundaries of the perturbation domain and minimize the calculated variance.	By using a variance constraint mechanism to determine the perturbation boundaries and variance calculation formula, the variance is minimized. Theoretical analysis shows that the variance of the SPM mechanism is smaller than that of the PNPM and PM mechanisms, allowing for the use of a smaller privacy budget.

3. Preliminary

3.1. Federated Learning

With the rapid development of artificial intelligence, high-quality data has become key to improving the performance of machine learning models. However, the sensitivity and high value of data make organizations and companies reluctant to share it, exacerbating the issue of data silos. Federated learning emerged to provide a method for collaborative training by sharing model updates instead of raw data, effectively protecting data privacy and promoting cooperation between different entities. Federated learning is particularly important in scenarios where data privacy regulations (such as GDPR) restrict cross-border data transfers. By avoiding data centralization, federated learning reduces risks such as data breaches and compliance issues. Federated learning can be divided into three main types based on data distribution characteristics: horizontal federated learning, vertical federated learning, and federated transfer learning [30]. In addition to protecting privacy, federated learning also facilitates the integration of data value across different organizations. For instance, hospitals can jointly train diagnostic models without directly sharing patient records, thereby enhancing diagnostic capabilities while preserving data privacy. This privacy-preserving distributed approach ensures robust model performance and mitigates risks like single points of failure and large-scale data breaches faced by centralized machine learning. In the federated learning framework, the system typically consists of a central server and N clients, where each client C_i possesses its local dataset M_i, where $i \in \{1, 2, \ldots, N\}$. The total dataset across all clients is M, that is, $\sum_{i=1}^{N} M_i = |M|$. The server collects and performs weighted aggregation of the local model parameters sent by the clients to generate the global model parameters, which are used to optimize the overall model performance.

$$\omega = \frac{1}{N} \sum_{i=1}^{N} \omega_i \tag{1}$$

3.2. Differential Privacy

Differential Privacy [31] is a technique that protects individual privacy by adding noise to the statistical results of raw data, ensuring that the inclusion or removal of a single data point does not significantly alter the output, thereby effectively safeguarding individual information. The basic principle involves adding random noise so that the output of a statistical query on a dataset remains almost the same whether a specific individual's data is included or not, making it difficult for an attacker to accurately infer the presence of any specific data point. The applications of differential privacy are not limited to privacy protection and statistical analysis; it also plays a role in several other areas. For instance, in machine learning, differential privacy is used to protect data privacy during model training, preventing the model from "memorizing" sensitive data, thereby avoiding the exposure of personal information during the inference stage. Additionally, differential privacy is used for generating synthetic data, allowing data scientists to perform modeling and analysis without accessing raw data. It is also applied in recommendation systems, where noise is added to user behavior data to protect individual privacy while improving recommendation quality.

Definition 1. *ϵ-differential privacy.*

An algorithm R is said to satisfy ε-differential privacy if, for any two adjacent datasets M and M_1, and any output set S of the algorithm R, the output probabilities satisfy the following inequality:

$$\Pr[R(M) \in S] \leq \Pr[R(M_1) \in S] \cdot e^{\epsilon} \tag{2}$$

In the definition of differential privacy, the probability $\Pr[R(M) \in S]$ represents the probability that the output of algorithm R, when run on dataset M, falls within the set

S. The privacy parameter ε measures the output difference of the algorithm on adjacent datasets, controlling the ratio of output probabilities. A smaller ε value indicates that the algorithm produces nearly identical outputs on adjacent datasets, thus providing stronger privacy protection, making it more difficult for an attacker to infer whether a particular data point is included in the dataset.

Traditional differential privacy achieves privacy protection by adding noise to the output of an algorithm. Its core idea is to obscure the influence of data through randomization, ensuring that the presence or absence of any single data point does not significantly change the output. Common noise mechanisms include the Laplace mechanism and the Gaussian mechanism. The Laplace mechanism is used for strict ε-differential privacy, where noise following a Laplace distribution is added to ensure that the output satisfies the differential privacy probability inequality. For any two adjacent datasets, the algorithm's probability density is strictly controlled within the range of ε, providing a theoretically strong privacy guarantee and making it more difficult for attackers to infer changes in the original data by observing the output.

In contrast, the Gaussian mechanism is used for (ε, δ)-differential privacy, where noise following a Gaussian distribution is added, allowing the privacy condition to be relaxed with a probability tolerance of δ. This means that the algorithm has a small probability of not satisfying the strict ε-differential privacy constraint, making it suitable for scenarios where both privacy and utility are highly demanded. Although (ε, δ)-differential privacy introduces more flexibility for privacy protection, strict ε-differential privacy is still considered a more ideal choice in many applications because it provides stricter and more explicit control over privacy leakage, offering a higher level of security assurance.

Our mechanism also uses strict ε-differential privacy to achieve privacy protection, but it differs significantly from the methods mentioned above. This approach ensures that in all cases, the algorithm's output meets the strict constraints of differential privacy, providing more reliable privacy protection for users' data while avoiding potential information leakage risks associated with looser privacy constraints.

4. SPM-FL: A Federated Learning Privacy-Protection Mechanism Based on Local Differential Privacy

This section proposes a new federated learning protection mechanism based on local differential privacy to enhance data security. As shown in Figure 1, the framework consists of two main steps: client training and server aggregation. In each round of global iteration, the server randomly selects a subset of clients to participate in training. The selected clients update the model on their local data and apply probabilistic positive or negative perturbations to the model weights to protect data privacy, then send the updated model parameters to the server. The server collects these parameters, performs aggregation, and updates the global model, gradually improving model performance while ensuring data privacy. The specific process can be found in Algorithm 1.

(1) Clients download the initial model or the aggregated updated model from the server and perform parallel training on local data to update the model parameters.

(2) Apply the SPM mechanism to the updated model parameters to add noise, thereby enhancing privacy protection.

(3) Upload the perturbed model parameters to the server, which aggregates the parameters from all clients until the preset number of iterations is reached.

This framework effectively protects client parameters through the perturbation mechanism while maintaining the training effectiveness of federated learning, all under the premise of ensuring data privacy.

Algorithm 1 Federated learning algorithm with localized differential privacy protection.

Input: Initial model parameters ω, learning rate l, number of clients C, client sampling rate q, communication rounds between clients and server T, number of local iterations on the client E, batch size B, client data M.

Output: Processed model parameters ω_{t+1}.

```
for $t = 1$ to $T$ do
    // Server-side aggregation phase
    $\tilde{\omega}_t \leftarrow \frac{1}{|C|}\sum_{c=1}^{C}\tilde{\omega}_t^c$    // Aggregate and update the global model
    Init $\omega_{t+1} \leftarrow \tilde{\omega}_t$
    // Local training phase
    for $c = 1$ to $C \cdot q$ do
        // Perform $E$ local iterations
        for $e = 1$ to $E$ do
            for each batch $B_i$ in $M$ do    // $i$ is an integer from 1 to $\lceil M/B \rceil$
                $g \leftarrow \nabla L(\omega, B_i)$
                $\omega_{t+1} \leftarrow \omega_{t+1} - lg$
            end for
        end for
        // Apply perturbation to the model parameters
        for each $\omega_t$ do
            $\overline{\omega_t} \leftarrow \mathrm{SPM}(\omega_t)$
        end for
    end for
end for
```

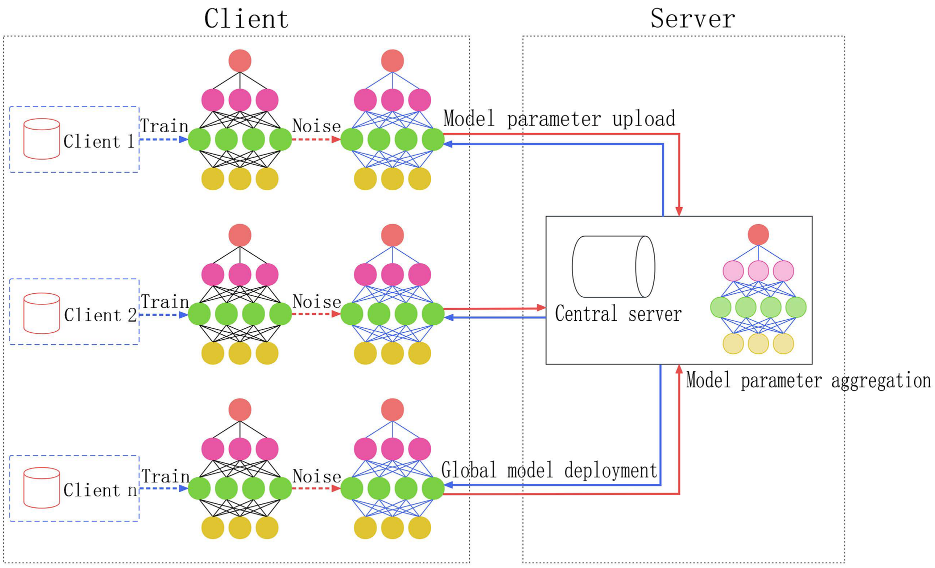


Figure 1. Federated learning framework with local differential privacy protection.

4.1. Symmetric Piecewise Mechanism

Unlike traditional noise addition methods in federated learning (such as Gaussian noise or Laplacian noise), this paper adopts a mechanism based on probabilistic inversion of the sign of model weights, which satisfies strict differential privacy requirements and protects the privacy of model parameters. This method makes it difficult to distinguish the sign of model weights, ensuring that even if an attacker gains access to the model parameters, they cannot accurately determine the sign of the weights, thereby effectively protecting privacy. Additionally, zero bias is introduced during the estimation of weight means to ensure that the aggregated perturbed model is as close as possible to the original model without noise. The specific operational steps are as follows (Algorithm 2):

1. The sign of the model weights is labeled as $t \in \{-1, 1\}$, with positive numbers marked as 1 and negative numbers as -1. Weights equal to 0 are not processed since they do not provide meaningful information to the model.

2. Set the range of the perturbation coefficient $C = \frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}$ as the boundary of the perturbation domain.

3. Based on the directionality of the model parameters and the algorithm mechanism, calculate the perturbation value $t^* \in [1, C]$. Then apply symmetric inversion to obtain the reverse perturbation domain $t^* \in [-C, -1]$. The probability density functions p and $\frac{e^{\varepsilon}}{p}$ are set for the perturbation values to meet strict differential privacy requirements.

4. The absolute value of the model weights $|\omega|$ is desensitized, then multiplied by the perturbation coefficient in probabilistic form to obtain the final uploaded model parameter $\bar{\omega} = |\omega| \cdot t^*$. This mechanism, through probabilistic inversion of the model weights, allows the server to cancel out noise during the joint aggregation of perturbed model parameters uploaded by the clients, thereby updating the global model parameters. This not only satisfies strict differential privacy protection but also maintains the overall performance of the model.

Algorithm 2 Symmetric piecewise mechanism.

```
Input: Privacy budget $\varepsilon$ and model parameters $\omega$.
Output: Perturbed model parameters $\bar{\omega}$.
Create mask with the same shape as $\omega$
Create direction_matrix with the same shape as $\omega$
Create $t^*$ with the same shape as $\omega$ and initialized to 0
threshold $\leftarrow \frac{e^{\varepsilon}}{e^{\varepsilon}+1}$
for each parameter $i$ in $\omega$ do
    if parameter $> 0$ then
        direction_matrix[i] $\leftarrow 1$
    else
        direction_matrix[i] $\leftarrow -1$
    end if
    Generate random number $x$ between 0 and 1
    if x < threshold then
        mask[i] $\leftarrow 1$
    else
        mask[i] $\leftarrow 0$
    end if
end for
$C \leftarrow \frac{e^{\varepsilon}+1}{e^{\varepsilon}-1}$
// Define vectorized computation for $lt$ and $rt$
$lt \leftarrow \frac{(C+1)}{2} \cdot$ direction_matrix $- \frac{(C-1)}{2}$
$rt \leftarrow \frac{(C+1)}{2} \cdot$ direction_matrix $+ \frac{(C-1)}{2}$
perturbation_lt_rt $\leftarrow [lt, rt]$
perturbation_rt_lt $\leftarrow [-rt, -lt]$
// Define the range for $t^*$
$t^*$[mask] $\leftarrow$ perturbation_lt_rt[mask]
$t^*$[$\neg$mask] $\leftarrow$ perturbation_neg_rt_lt[$\neg$mask]
$\bar{\omega} \leftarrow t^* \cdot |\omega|$
Return $\bar{\omega}$
```

As shown in Algorithm 2, in the SPM mechanism, the boundary value C of the perturbation domain is calculated through the left and right boundaries lt and rt. Regardless of the sign of the model weights, the perturbation value always falls between the two fixed value sets $\{1, C\}$ and $\{-C, -1\}$. Therefore, it can be derived that the perturbation value $t^* \in [-C, -1] \cup [1, C]$. Due to the design of the symmetric piecewise mechanism, the perturbation value cannot take values within the interval $[-1, 1]$, meaning the model weights will only be scaled proportionally in either the positive or negative direction.

As the boundary value of the perturbation domain, the appropriate selection of C determines the range of the scaling factor for model weights. The probability density p, which determines the perturbation value responsible for weight inversion (i.e., the probability of falling in the positive or negative range), dictates whether the model weights will change their sign. We found that the selection of the p value for the symmetric interval must meet the strict differential privacy theorem (as proven in Theorem A2), meaning it must satisfy a specific ratio and therefore cannot be arbitrarily modified. However, there is some flexibility in the choice of C. During the model update process, if the weight of a certain sample does not undergo probabilistic inversion, it only experiences coefficient scaling. In such cases, appropriately constraining the size of C can control the range of the perturbation domain and thereby reduce the absolute value of the perturbation. As the perturbation domain narrows, the variance introduced by the perturbation mechanism (i.e., the degree of dispersion of the perturbed model weights) will also decrease accordingly.

Since C serves as the boundary of the perturbation domain, we introduced zero bias during the mean estimation process. Therefore, when calculating the expected value, it is necessary to account for the probability density p of the perturbation value. There is a complex relationship between the C value, p value, and the privacy budget ε. Specifically, by setting C and p to satisfy a certain relationship, we can ensure the zero bias condition holds during the mean estimation of the weights. Thus, we ensure that the SPM mechanism satisfies both strict differential privacy and zero bias during the mean estimation of the weights. We conducted a mathematical analysis of the relationship between these three factors and successfully established the maximum constraint on the C value within the mechanism, ensuring C is the minimum possible value used in the SPM mechanism without affecting other aspects. This approach results in significant improvement in the variance, which is why this theorem is referred to as the "Variance Constraint Mechanism". It is worth noting that the detailed proofs of the lemmas and theorems discussed in this paper are included in the Appendix A.

Theorem 1. ***SPM Mechanism and Differential Privacy.*** *When the boundary C of the perturbation coefficient and the probability density p of the perturbation value satisfy Equation (3), the SPM mechanism can ensure that the model parameters uploaded by clients participating in federated learning training meet ε-differential privacy. Moreover, this mechanism ensures zero bias during mean estimation of the weights and minimizes the dispersion (variance) of the model weights after being processed by the perturbation mechanism.*

$$\begin{cases} C = \frac{e^{\epsilon}+1}{e^{\epsilon}-1} \\ p = \frac{e^{\epsilon}-1}{2} \end{cases} \tag{3}$$

For the direction of model weights, we propose two options, and for the perturbation domain, there are also two probability choices. Therefore, the probability density function should satisfy:

$$\begin{cases} \operatorname{pdf}(t^* = x \mid t_i) = \begin{cases} p & \text{if } x \in [l(t), r(t)],\ t_i = 1 \\ \frac{p}{e^{\epsilon}} & \text{if } x \in [-r(t), -l(t)],\ t_i = 1 \end{cases} \\ \operatorname{pdf}(t^* = x \mid t_i) = \begin{cases} \frac{p}{e^{\epsilon}} & \text{if } x \in [l(t), r(t)],\ t_i = -1 \\ p & \text{if } x \in [-r(t), -l(t)],\ t_i = -1 \end{cases} \end{cases} \tag{4}$$

where $l(t) = \frac{C+1}{2} \cdot t_i - \frac{C-1}{2}$ and $r(t) = \frac{C+1}{2} \cdot t_i + \frac{C-1}{2}$, with t_i being the direction of the model weight.

Notably, "zero bias" in statistics and machine learning refers to an estimation method whose expected estimate equals the true value. In federated learning, the server generates a global model update by aggregating local model updates (such as gradients or weights) from different clients. When the federated learning process satisfies "zero bias", the expected value of the global model parameters equals the true mean of the client model parameters. This ensures that no systematic error is introduced during aggregation, making the global model more representative and accurate.

As previously mentioned, introducing "zero bias" is based on combining differential privacy to maintain model efficiency while protecting privacy. In federated learning, if only the general form of the SPM mechanism is used, it is possible to resist attacks through probabilistic flipping of weights; however, it does not ensure model accuracy (i.e., the relationships between parameters are not preserved, and a smaller value of C strengthens privacy protection but may reduce model accuracy). Introducing zero bias effectively addresses this issue, with both mechanisms complementing and constraining each other. The variance constraint mechanism was derived through parameter constraints, thus forming the SPM mechanism (Algorithm 2). The variance constraint mechanism constrains relationships between parameters (such as the settings of C and p), minimizing the impact of the SPM mechanism while ensuring privacy protection. This allows the federated learning process to achieve not only zero bias but also strict differential privacy requirements.

Of course, theoretical proof alone is insufficient to verify its effectiveness. Therefore, in Section 5, we further validate the mechanism's effectiveness through experiments across multiple dimensions. More details of the theoretical derivations can be found in the Appendix A.

4.2. Usability and Privacy Analysis

This paper evaluates the usability and privacy of the SPM mechanism through both theoretical and experimental approaches. For usability analysis, Lemma A1 is proposed to support the theoretical part, and experimental analyses are conducted in Sections 5.2.1– 5.2.3 and 5.2.5. For privacy analysis, Theorems A2 and A3 provide the theoretical proof, and the experimental Section 5.2.4 uses the traditional DLG attack to verify the privacy-protection capability of the SPM mechanism, aiming to fully assess the robustness of the mechanism.

In this section, we use theoretical proof to verify the mechanism's usability, with variance as the evaluation metric. Similar to traditional noise perturbation in federated learning, Gaussian noise and Laplacian noise produce irregular noise. We selected three advanced differential privacy mechanisms in federated learning and compared their variances under the same privacy budget, analyzing them alongside traditional noise-adding methods. The following theorems and lemmas establish the advantages of the SPM mechanism in terms of variance and the model's usability. The proof of this lemma is provided in the Appendix A.

Lemma 1. *The variance of the SPM mechanism is strictly smaller than the variance of the Laplace mechanism and the variances in the literature [27,29], and it is independent of the value of the privacy budget ε.*

Lemma A1 proves the usability of this mechanism, showing that the model weight parameters, after being perturbed by the mechanism, still maintain high accuracy after aggregation across clients. Compared to other algorithms, it has a smaller variance, making the SPM mechanism more favorable than the Laplace mechanism and the solutions presented in the literature [27,29]. See Figure 2.

Since the PM mechanism has already been proven to have a variance strictly smaller than that of the Laplace mechanism, we only need to verify that the variance of the SPM mechanism is strictly smaller than that of the other mechanisms. Figure 2 shows that when the privacy budget is 1.5, the variance of SPM remains relatively stable, with a maximum value of approximately 1, which is lower than that of DuChi et al. [26], the PM mechanism [27], and PNPM mechanism [29]. When the privacy budget is 0.6, although the variance of SPM increases with the model weights, it remains significantly lower than that of the other mechanisms. This reduces the impact of perturbation on model weights, thereby enhancing the level of privacy protection. This indicates that the SPM mechanism is more suitable for smaller privacy budgets, which will be further demonstrated in the subsequent experimental design section.

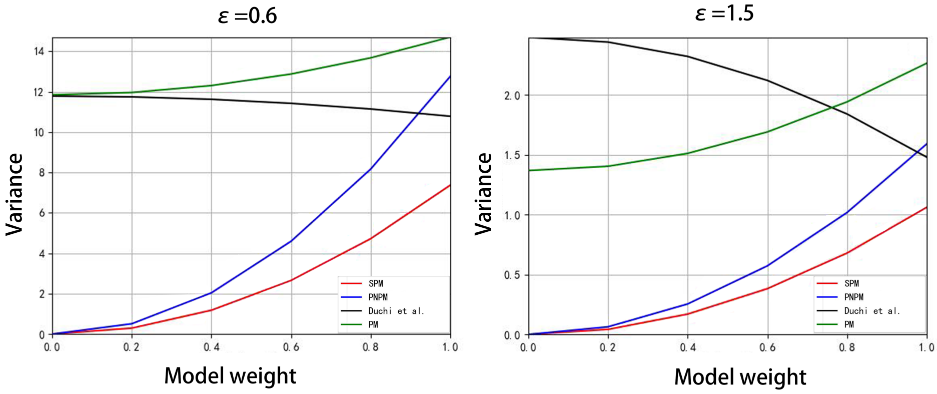


Figure 2. The impact of model weights on variance under different privacy budgets [26].

Theorem 2. *For any model weight input values $t, t' \in \{1, -1\}$ and perturbation coefficient output value $t^* \in \left[-\frac{e^\epsilon+1}{e^\epsilon-1}, -1\right] \cup \left[1, \frac{e^\epsilon+1}{e^\epsilon-1}\right]$, the SPM mechanism satisfies $\frac{pdf(t^*|t)}{pdf(t^*|t')} \leq \frac{p}{\frac{p}{e^\epsilon}} = e^\epsilon$, thereby ensuring ϵ-local differential privacy. Additionally, a zero bias is introduced in the mean estimation of the weights to ensure that $E[\bar{S}(\omega)] = \bar{\omega}$, which means that the expected value of the mean parameters of the aggregated perturbation model equals the mean parameters of the original aggregated model.*

Theorem A2 proves that this mechanism satisfies strict ε-localized differential privacy and that the joint model after perturbation aggregation can effectively approximate the utility of the original aggregated model, denoted as $E[\overline{S(\omega)}] = \bar{\omega}$. Based on this, we can derive Theorem A3, which establishes the asymptotic error bound of $\overline{S(\omega)}$. The proof of this theorem is provided in the Appendix A.

Theorem 3. *For all $\omega \in W$, there exists $\lambda = O\left(\frac{|\omega|}{\varepsilon} \cdot \sqrt{\frac{\ln(1/\beta)}{n}}\right)$ such that the absolute difference $|\overline{S(\omega)} - \bar{\omega}| < \lambda$ with a probability of at least $1 - \beta$.*

Theorem A3 establishes the accuracy guarantee of the SPM mechanism, where W represents the set of model parameters. The proof of this theorem is provided in the Appendix A. From the above theorems, it is clear that the model weight parameters, after being perturbed by the SPM mechanism, undergo probabilistic sign inversion and become indistinguishable. As shown by the variance $\mathrm{Var}[S(\omega)] = |\omega|^2 \cdot \frac{3e^\varepsilon + \frac{1}{3e^\varepsilon} - \frac{2}{3}}{(e^\varepsilon - 1)^2}$, as ω increases, the variance of the mechanism also increases, leading to greater perturbation and stronger privacy protection, effectively resisting membership inference attacks [32].

4.3. Security Model

This section introduces the security model in federated learning. It begins with an analysis of the threats faced by the system, followed by a clarification of the security objectives, and concludes with corresponding defense strategies to ensure the system's privacy and integrity.

4.3.1. Threat Model

Honest-but-curious server: The server operates according to the federated learning protocol but attempts to extract additional information from user-uploaded gradients or weights to infer training data features, leading to privacy leakage.

DLG attack (Deep Leakage from Gradients): An attacker intercepts the gradients uploaded by clients and uses reverse optimization to gradually reconstruct the original training data. Even without direct access to the data, privacy can be compromised through gradient leakage.

Inference attacks: An attacker infers whether specific data was used in training by analyzing the model's output, particularly accumulating knowledge about the training set over multiple interactions, threatening user privacy.

Reconstruction attacks: An attacker uses shared model updates (e.g., weights or gradients) to reconstruct the original training data. This is particularly feasible when the dataset is small, allowing the attacker to infer training data details from changes in model parameters.

4.3.2. Security Objectives

Protect user privacy: Ensure that the server or other attackers cannot reconstruct the original training data from the gradients or weights uploaded by clients, preventing the leakage of sensitive information.

Resist gradient attacks: Ensure that even if an attacker obtains the uploaded gradient information, it remains difficult to reconstruct the original training data through reverse optimization.

Prevent membership inference: Reduce the attacker's ability to infer whether a particular data point is part of the training set from the model's output, thereby protecting the privacy of the training data.

Defend against reconstruction attacks: Ensure that attackers cannot reconstruct the original training data from shared model updates, especially by adding perturbations to increase reconstruction difficulty.

Model integrity: Ensure that the final global model is not compromised by malicious clients, maintaining the accuracy and robustness of the global model.

4.3.3. Defense Strategies

Symmetric Partition Mechanism (SPM) and Differential Privacy Application: In federated learning, clients use the SPM to probabilistically perturb the sign of weights before uploading the model, which satisfies ϵ-differential privacy. By introducing noise through this perturbation method, the likelihood of attackers inferring the original training data from gradients or weights is significantly reduced, effectively defending against common privacy threats such as inference attacks, DLG attacks, and reconstruction attacks.

Variance Constraint Mechanism: The perturbation process adheres to a variance constraint mechanism to control the variance of the added noise without altering the strength of privacy protection. This mechanism ensures that the model's performance does not degrade significantly due to excessive noise while maintaining data privacy. By constraining the variance of the noise, it effectively reduces model fluctuation, allowing it to remain both resistant to attacks and accurate.

Zero Bias Design: Zero bias is introduced in the federated learning process by ensuring that the perturbed weights from each client remain statistically unbiased, meaning the expected value of the weights does not change. This approach not only satisfies differential privacy requirements but also minimizes the negative impact of perturbation on model accuracy, ensuring that the global model achieves a good balance between privacy protection and utility after aggregation.

5. Experiments

5.1. Experiment Settings

- Datasets:

  - MNIST Dataset: The dataset used for handwritten digit recognition contains images of digits from 0 to 9. It consists of grayscale images with a resolution of 28 × 28 pixels, with 60,000 images used for the training set and 10,000 images for the test set.
  - Fashion-MNIST: The fashion classification dataset contains 70,000 grayscale images in 10 categories. Each image has a resolution of 28 × 28 pixels. The entire dataset is divided into two parts: 60,000 images form the training set, and 10,000 images are used for the test set.
  - CIFAR-10 Dataset: It consists of colored images in 10 categories, including ships, airplanes, cars, trucks, birds, cats, deer, dogs, frogs, and horses, all with a resolution of 32 × 32 pixels. A total of 50,000 images are used for the training set, and 10,000 images are used for the test set.

The reasons for selecting the MNIST, Fashion-MNIST, and CIFAR-10 datasets for the study of federated learning and differential privacy are as follows:

(1) Diversity and Representativeness: These datasets cover image classification tasks ranging from simple (MNIST, Fashion-MNIST) to complex (CIFAR-10), allowing the proposed method to be validated in terms of adaptability and performance across tasks of varying difficulty.

(2) Wide Usage and Recognition: These datasets are widely used in the field of machine learning, and selecting them facilitates comparison with existing research, demonstrating the effectiveness and improvement of the proposed method.

(3) Federated Learning and Differential Privacy Verification: MNIST and Fashion-MNIST are suitable for testing basic privacy-protection effects, while CIFAR-10 is used to evaluate the performance and robustness of the method in handling more complex data.

(4) Standard Benchmark: As classic benchmark datasets, using these three datasets enhances the reproducibility of experimental results and adds reference value to the research.

- Models:

For simple datasets such as MNIST and Fashion-MNIST, the network model first flattens the 28 × 28 pixel images into a 784-dimensional vector. Then, a fully connected layer maps the 784 input features to 256 dimensions, with a ReLU activation function applied to the output of this layer to introduce nonlinearity. For the more complex CIFAR-10 dataset, we used two models: (1) The first model contains two convolutional layers, which map the input features to 64 and 128 channels, respectively, and are processed by ReLU activation functions and max-pooling layers. The feature maps are then flattened into a vector and passed through three fully connected layers, transforming the features from a shape of "128 channels, 8 height, 8 width" to 384 and 192 dimensions, and finally outputting the classification result for 10 categories. (2) The second model uses the ResNet18 network.

- Experimental setup:

The experiments were conducted using the PyTorch (Torch1.10.0 Py3.8(ub20.04) Cu113) framework running on an NVIDIA GeForce RTX 4090 server, with a CPU consisting of 16 vCPUs Intel(R) Xeon(R) Gold 6430.

5.2. Experimental Results and Analysis

Model accuracy reflects the model's classification ability on the test dataset, i.e., the proportion of correctly classified samples. Higher accuracy indicates good classification performance on the test set, directly representing the overall performance of the model. Analyzing model accuracy under different client numbers can effectively assess the robustness of the algorithm. Since the total dataset size remains constant, changes in the number of clients lead to different allocations of the training dataset. Therefore, the ability of the model to maintain high accuracy under varying training samples is a key challenge in federated learning. To evaluate the impact of different client numbers in federated learning on model accuracy, communication overhead, and strategies for handling attacks, the experimental design in this paper is divided into two parts, analyzed from different perspectives, with each section interconnected.

Part 1: We compare the SPM mechanism with the basic federated learning algorithm without differential privacy (NoDP-FL) [30] and three advanced differential privacy mechanisms: (1) the PM mechanism [27], (2) the mechanism by Sun et al. [28], and (3) the PNPM mechanism [29]. In Sections 5.2.1 and 5.2.2, we verify the model accuracy of different mechanisms in scenarios with varying client numbers to comprehensively analyze the robustness of each mechanism. In the first part of Section 5.2.3, we conducted a usability analysis, comparing the privacy budget consumption of each mechanism in achieving the same model accuracy. In the second part of Section 5.2.3, we selected representative client numbers of 30 and 300 to analyze the usability of the SPM mechanism under different privacy budgets and explore its maximum model accuracy under the same configuration.

Part 2: We will further explore the SPM mechanism. In Section 5.2.4, we examine the robustness of the mechanism and its defense strategies when facing gradient attacks. In Section 5.2.5, we investigate the impact of local iteration count on communication overhead in federated learning. In our analysis, in addition to evaluating the robustness of the algorithm, we also conducted a comprehensive assessment of the mechanism's usability, privacy-protection capability, and defense performance in attack scenarios.

These analyses help us gain a comprehensive understanding of the mechanism's performance and applicability in real-world applications.

5.2.1. Model Accuracy Analysis in Scenarios with Few Clients

In this section of the experiment, we ensure that the parameter configurations and privacy budgets for each mechanism remain consistent, selecting a client count of 5 to 50 for the limited client scenario, in which two network models are used with the CIFAR-10 dataset. The specific settings are as follows: a sampling rate of 0.6 (with a sampling rate of 1 for the two models in the CIFAR-10 dataset), a local iteration count of 3 (with the two models in the CIFAR-10 dataset set to 5 and 7 iterations, respectively), a batch size of 64 (with a batch size of 32 for CIFAR-10 Model 2), and a global communication round count of 50. The selection of these parameters is based on a comprehensive consideration of communication and time overhead. In Section 5.2.4, we will discuss in detail how to slightly enhance model performance by increasing local computation time while maintaining the same privacy budget, and simultaneously reducing global communication costs appropriately. Next, we will analyze the model accuracy performance of different mechanisms under each dataset.

The MNIST dataset uses a privacy budget of 0.3. As shown in Table 2, there are significant differences in the performance of various algorithms under different client counts. First, in terms of overall accuracy, the NoDP algorithm exhibits the best model accuracy across all client counts, consistently maintaining over 90% accuracy with minimal fluctuation as the number of clients increases. This is because it does not introduce noise perturbations, resulting in weaker privacy protection; it will serve as our baseline comparison mechanism for this section. In contrast, the SPM mechanism closely follows, demonstrating model accuracies near that of NoDP, especially with client counts of 10 and 20, where SPM achieves accuracies of 89.48% and 89.66%, nearly matching NoDP. This indicates that SPM can still provide privacy protection without significantly reducing model accuracy, which is its advantage over other privacy-protection mechanisms. The accuracies of mechanisms like PNPM and those by Sun et al. [26] are relatively low, particularly when the number of clients is small. While the mechanism by Sun et al. [26] performs poorly with a small number of clients, it shows improvement when the client count reaches 30. The PM mechanism clearly performs the worst, especially with a client count of 5, achieving an accuracy of only 52.73%, which is significantly lower than other algorithms, indicating that this mechanism is not suitable for scenarios with a privacy budget.

Table 2. Model accuracy data for different mechanisms under the limited client scenario of the MNIST dataset.

Algorithm	C = 5	C = 10	C = 20	C = 30	C = 40	C = 50
NoDP	90.35	90.39	90.55	90.42	90.48	90.28
SPM	89.4	89.48	89.66	89.25	89.21	88.92
PNPM	85.37	86.42	85.53	84.45	84.04	82.88
Sun et al. [26]	79.64	81.27	79.8	80.89	79.39	78.74
PM	52.73	66.58	75.73	63.26	57.89	67.78

Figure 3 illustrates the trend of model accuracy under different mechanisms on the MNIST dataset. From the figure, it can be observed that the NoDP mechanism maintains a relatively stable model accuracy as the number of clients increases, while other mechanisms exhibit varying degrees of decline, with SPM showing the slowest decrease, indicating that this mechanism is more suitable for limited client scenarios. In contrast, other mechanisms perform poorly when the number of clients reaches 50. This is due to the inverse relationship between the number of clients and the amount of training data allocated to each client; the reduction in the training dataset leads to a decline in the model accuracy of these mechanisms.

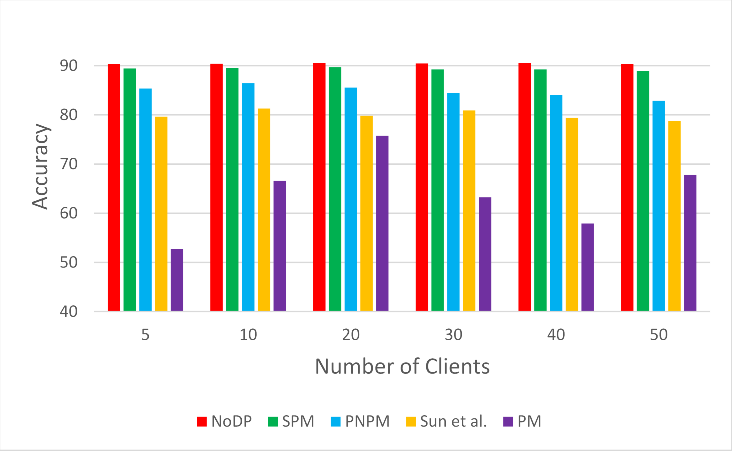


Figure 3. Comparative analysis of model accuracy for different mechanisms under the limited client scenario of the MNIST dataset [26].

The Fashion-MNIST dataset uses a privacy budget of 0.6. As shown in Table 3, the NoDP mechanism maintains stable model accuracy across different client counts, while SPM, although slightly lower, is close to NoDP, particularly with 5 to 10 clients, where SPM's accuracies are 83.78% and 83.99%, with a difference of less than 1%. In contrast, the mechanisms by PNPM and Sun et al. [26] show a significant decline in accuracy as the number of clients increases, especially with PNPM dropping to 80.71% when the client count reaches 50. The PM mechanism performs poorly with a limited number of clients, achieving an accuracy of only 56.05%. Although it improves subsequently, the performance remains unstable.

Table 3. Model accuracy data for different mechanisms under the limited client scenario of the Fashion-MNIST dataset.

Algorithm	5	10	20	30	40	50
NoDP	84.47	84.56	84.54	84.55	84.54	84.58
SPM	83.78	83.99	83.64	83.28	82.93	82.62
PNPM	81.89	81.56	81.05	81.14	80.56	80.71
Sun et al. [26]	80.71	80.67	81.08	80.95	80.62	80.97
PM	56.05	64.20	70.46	69.85	64.25	71.88

Figure 4 illustrates the trend of model accuracy under different mechanisms on the Fashion-MNIST dataset. It can be observed that the model accuracy of all mechanisms shows a slight decline as the number of clients increases, but remains relatively stable at client counts of 5 and 10. Notably, the SPM mechanism, when the client count reaches 50, has an accuracy that is 1.96% lower than NoDP, yet still surpasses the accuracy of other mechanisms, demonstrating the advantages of the SPM mechanism.

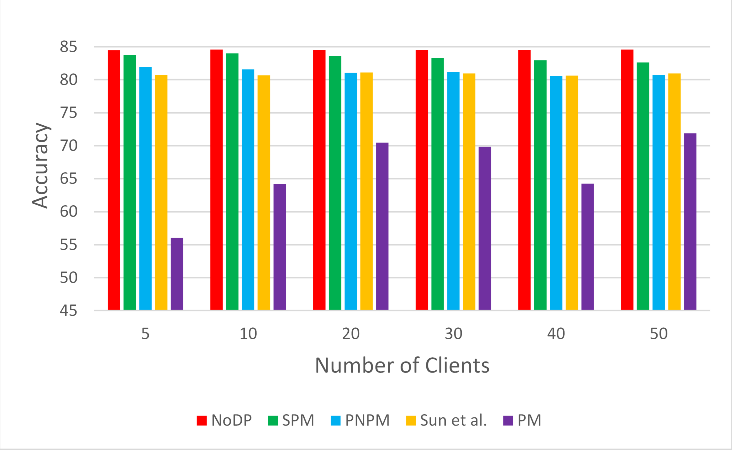


Figure 4. Comparative analysis of model accuracy for different mechanisms under the limited client scenario of the Fashion-MNIST dataset [26].

In the CIFAR-10 dataset (Model 1), a privacy budget of 1.9 is used (NaN indicates gradient explosion under this privacy budget, preventing convergence). As shown in Table 4, the NoDP mechanism performs steadily with client counts of 5 to 10, consistently maintaining an accuracy of over 81%. The SPM mechanism experiences a slight decline in accuracy as the number of clients increases, yet still maintains an accuracy of 75.14% with 50 clients, which is higher than that of other mechanisms. In contrast, the PNPM and Sun et al. [26] mechanisms exhibit lower accuracies and demonstrate a noticeable decline in performance as the number of clients increases. The PM mechanism performs poorly with a limited number of clients. This indicates that under a lower privacy budget, the SPM mechanism can maintain high performance.

Table 4. Model accuracy data for different mechanisms under the limited client scenario of the CIFAR-10 dataset (Model 1).

Algorithm	5	10	20	30	40	50
NoDP	81.19	81.44	79.50	76.42	79.25	77.66
SPM	80.19	80.41	79.36	76.08	75.01	75.14
PNPM	68.49	67.73	64.18	65.09	66.04	63.29
Sun et al. [26]	58.39	58.76	54.79	54.96	55.37	55.46
PM	NaN	NaN	NaN	NaN	NaN	NaN

Figure 5 illustrates the trend of model accuracy under different mechanisms on the CIFAR-10 dataset (Model 1). We observe that with 5 to 30 clients, SPM maintains a model accuracy similar to that of NoDP. However, with 40 to 50 clients, all mechanisms show a significant decline in model accuracy. This may be related to the CIFAR-10 dataset consisting of color three-channel images, which increases training complexity; however, this trend differs from the performance observed in other datasets. To further investigate this situation, we employed a second network architecture for this dataset, which will be analyzed comprehensively in Model 2.

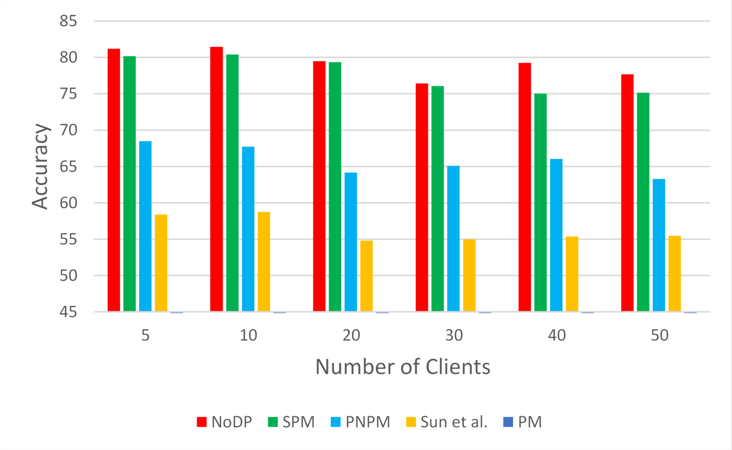


Figure 5. Comparative analysis of model accuracy for different mechanisms under the limited client scenario of the CIFAR-10 dataset (Model 1) [26].

As shown in Table 5, in the CIFAR-10 dataset (Model 2), we set the privacy budget to 3.9. We found that in a more complex network (ResNet18), the lower privacy budget of 1.9 could not yield optimal performance for the mechanisms, so we made a moderate increase in the privacy budget. With a client count of 5, all mechanisms achieved high model accuracy, with the lowest being 78.33%. This phenomenon can be attributed to the limited number of clients, allowing each client to utilize a more sufficient training set, along with the ability of deep networks to extract more useful information. However, as the client count increases to 40 to 50, the reduction in the local training set leads to a decline in the model accuracy of all mechanisms, while the SPM mechanism still maintains a high accuracy of 84.53%, with a difference of 2.93% compared to NoDP.

We further analyzed the reasons for the significant decline in Model 1 when the client count was between 40 and 50. Compared to Model 1, Model 2 has a greater depth, enabling it to extract more information across a larger number of clients (e.g., 50 clients). Additionally, the lower privacy budget used in Model 1 is also a contributing factor to the decline. Since deep networks learn more information when learning sample features, a very low privacy budget (such as 1.9 in Model 1) may lead to overfitting in the early stages of training, thus preventing convergence. Meanwhile, a smaller privacy budget can also alter the features learned by the model in the early stages during aggregation. In Section 5.2.4, we will explore how to enhance model accuracy through other means while maintaining the privacy budget unchanged.

Table 5. Model accuracy data for different mechanisms under the limited client scenario of the CIFAR-10 dataset (Model 2).

Algorithm	5	10	20	30	40	50
NoDP	87.78	86.95	87.79	86.77	86.87	87.46
SPM	86.99	86.94	85.61	85.08	84.68	84.53
PNPM	82.87	83.43	83.17	79.97	80.50	83.07
Sun et al. [26]	78.98	72.99	71.91	71.39	68.75	67.60
PM	78.33	79.06	75.31	53.52	54.44	58.93

Figure 6 illustrates the trend of model accuracy under different mechanisms on the CIFAR-10 dataset (Model 2).

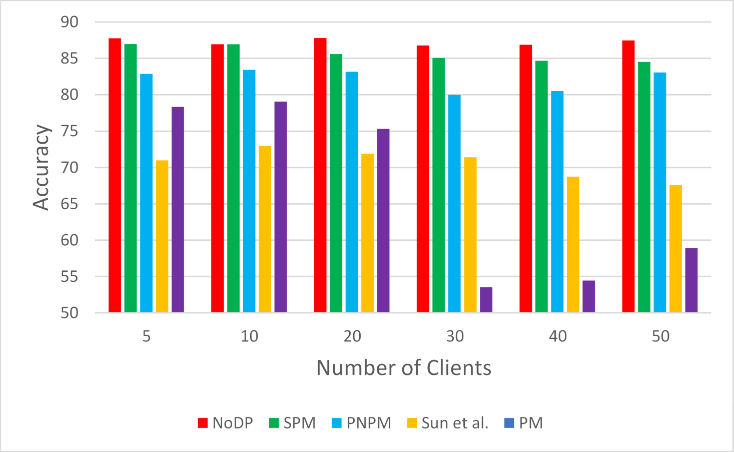


Figure 6. Comparative analysis of model accuracy for different mechanisms under the limited client scenario of the CIFAR-10 dataset (Model 2) [26].

5.2.2. Model Accuracy Analysis in Scenarios with Many Clients

In the previous experiment, we explored the model accuracy of various mechanisms in a limited client scenario; however, in real-world applications, the number of clients often varies dynamically due to communication costs and resource constraints. Therefore, the robustness and adaptability of mechanisms in multi-client scenarios are particularly important to ensure that the model maintains good performance across different environments. Hence, our mechanisms need to be adapted to multi-client scenarios.

In this section, we selected a client count ranging from 100 to 500 as the multi-client scenario. The CIFAR-10 dataset also employs two network models. The specific settings are as follows: a sampling rate of 0.6 (the sampling rate cannot be 1 in multi-client scenarios to prevent excessive local noise from interfering with global model aggregation), local iterations set to 3 (with 15 and 10 iterations for the two models in the CIFAR-10 dataset, and 10 iterations for the Fashion-MNIST dataset), a batch size of 64 (with a batch size of 32 for CIFAR-10 Model 2), and 50 global communication rounds (with 20 for the Fashion-MNIST dataset). Next, we will analyze the performance of different mechanisms in terms of model accuracy for each dataset.

The privacy budget for the MNIST dataset is set to 0.3. As shown in Table 6, in the multi-client scenario, we find that the model accuracy of NoDP on the MNIST dataset does not decline. This is because the dataset is relatively simple and easy to train. With a client count of 100, the accuracy of SPM is 88.37%, which is 1.63% lower than NoDP's 90; with a client count of 500, SPM's accuracy is 84.31%, differing by 5.85% from NoDP. Nevertheless, SPM's accuracy remains higher than that of other mechanisms, further validating the effectiveness of our algorithm.

Table 6. Model accuracy data for different mechanisms under the multi-client scenario of the MNIST dataset.

Algorithm	100	200	300	400	500
NoDP	90.00	90.44	90.63	90.18	90.16
SPM	88.37	87.11	85.88	85.34	84.31
PNPM	80.76	79.92	78.45	76.00	76.42
Sun et al. [26]	73.75	69.03	70.04	67.05	57.82
PM	55.46	36.86	33.90	25.74	19.41

Figure 7 illustrates the trend of model accuracy under different mechanisms on the MNIST dataset.

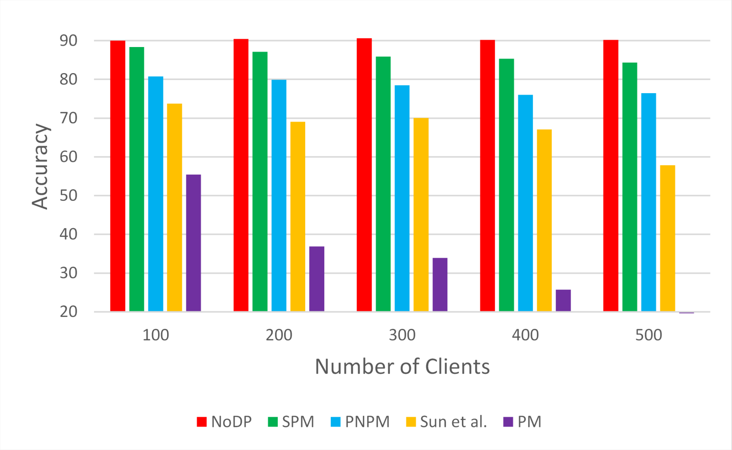


Figure 7. Comparative analysis of model accuracy for different mechanisms under the multi-client scenario of the MNIST dataset [26].

The privacy budget for the Fashion-MNIST dataset is set to 0.9. Compared to the limited client scenario, we slightly increased the privacy budget in the multi-client scenario because the increase in the number of clients leads to a reduction in the training set allocated to each client, consequently reducing the available feature information. A moderate increase in the privacy budget can effectively reduce noise, preventing the model from experiencing gradient explosion due to noise interference. As shown in Table 7, among all clients, the model accuracy of the SPM mechanism differs from that of the NoDP mechanism by at least 0.06% and at most 1.87%. Except for the PM mechanism, the model accuracies of the other mechanisms are relatively high, indicating that in the multi-client scenario, the SPM, PNPM, and Sun et al. [26] mechanisms can all demonstrate good performance.

Table 7. Model accuracy data for different mechanisms under the multi-client scenario of the Fashion-MNIST dataset.

Algorithm	100	200	300	400	500
NoDP	82.64	81.66	81.87	80.97	80.46
SPM	82.60	81.60	80.40	79.74	78.59
PNPM	79.77	78.94	77.66	78.89	75.06
Sun et al. [26]	77.51	78.28	77.16	76.77	74.99
PM	67.45	54.83	68.17	55.81	39.76

Figure 8 illustrates the trend of model accuracy under different mechanisms on the Fashion-MNIST dataset.

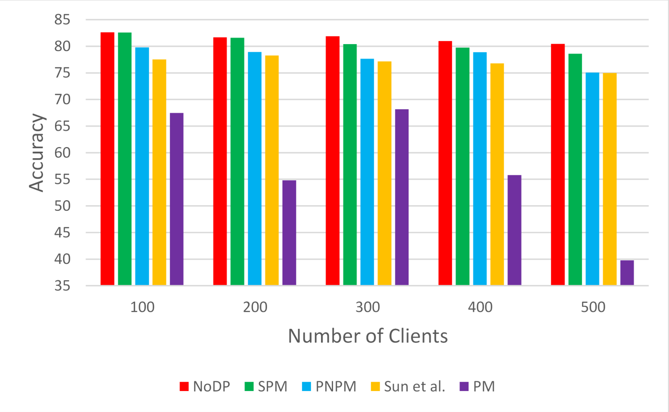


Figure 8. Comparative analysis of model accuracy for different mechanisms under the multi-client scenario of the Fashion-MNIST dataset [26].

The privacy budget for the CIFAR-10 dataset (Model 1) is set to 2.9. For this relatively complex dataset, our privacy budget is slightly increased compared to the limited client scenario. As shown in Table 8, with a client count of 100, the SPM mechanism differs from NoDP by 1.74%, outperforming other mechanisms. However, as the client count increases, the reduction in the training set leads to an expanding gap between SPM and NoDP, with similar performance trends observed for other mechanisms. With a client count of 500, the accuracy of NoDP is 67.27%, while the accuracies of SPM, PNPM, Sun et al. [26], and PM are 57.83%, 54.01%, 50.71%, and 45.16%, respectively. Nevertheless, SPM still maintains the highest accuracy among all mechanisms.

This indicates that in multi-client scenarios, shallow network models struggle to handle more useful information for complex datasets. If we aim to improve model accuracy, a moderate increase in the privacy budget may be necessary, though this could also reduce the privacy-protection capability of the mechanism. In this regard, we analyze the usability of our algorithm under different privacy budgets in Section 5.2.2.

Table 8. Model accuracy data for different mechanisms under the multi-client scenario of the CIFAR-10 dataset (Model 1).

Algorithm	100	200	300	400	500
NoDP	76.46	74.69	71.46	67.67	67.27
SPM	74.72	70.39	65.55	59.85	57.83
PNPM	65.70	56.51	56.45	55.21	54.01
Sun et al. [26]	61.19	55.30	50.62	51.23	50.71
PM	53.36	49.82	49.37	50.49	45.16

Figure 9 illustrates the trend of model accuracy under different mechanisms on the CIFAR-10 dataset (Model 1).

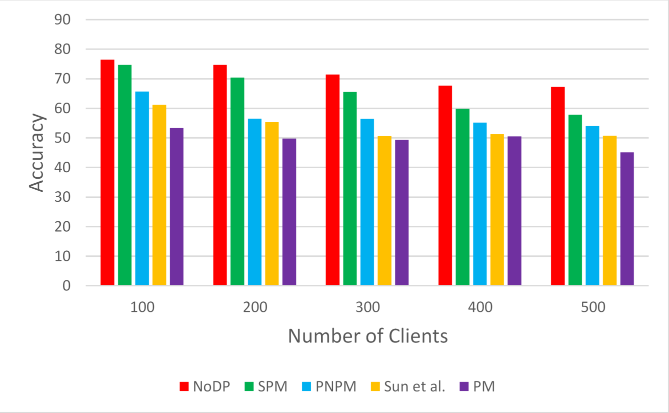


Figure 9. Comparative analysis of model accuracy for different mechanisms under the multi-client scenario of the CIFAR-10 dataset (Model 1) [26].

The privacy budget for the CIFAR-10 dataset (Model II) is set to 3.9, similar to the low client scenario. As shown in Table 9, we find that the performance of the SPM and PNPM mechanisms is nearly consistent across clients, with only minor differences. For example, with 100 clients, the accuracy of SPM is 83.58%, while that of PNPM is 82.47%, indicating that SPM slightly outperforms PNPM. However, with 500 clients, the accuracy of PNPM (77.56%) surpasses that of SPM (76.64%). The accuracy of other mechanisms is relatively low. Based on the above experimental analysis, we conclude that in deeper network models, the accuracy of all mechanisms generally improves, demonstrating superior performance compared to shallow models. Even with a larger number of clients (500), the accuracy of SPM and PNPM is similar to that of NoDP. This indicates that deep models can learn more useful information from relatively smaller training sets, thereby enhancing model accuracy, although their training time cost is significantly higher than that of shallow models.

Table 9. Model accuracy data for different mechanisms in the multi-client scenario of the CIFAR-10 dataset (Model 2).

Algorithm	100	200	300	400	500
NoDP	84.28	82.33	81.56	80.74	79.29
SPM	83.58	81.45	79.24	78.74	76.64
PNPM	82.47	81.22	78.58	78.64	77.56
Sun et al. [26]	74.56	68.58	64.05	63.14	59.25
PM	67.91	60.91	60.39	69.32	59.87

Figure 10 illustrates the variation trend of model accuracy for different mechanisms under the CIFAR-10 dataset (Model 2).

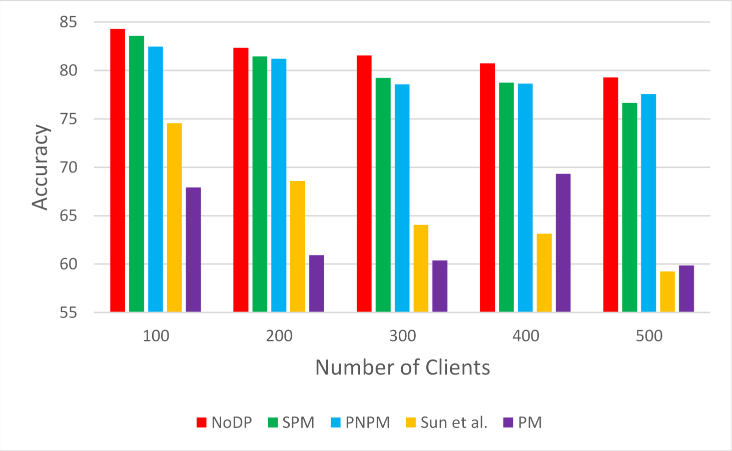


Figure 10. Comparative analysis of model accuracy for different mechanisms in the multi-client scenario of the CIFAR-10 dataset (Model 2) [26].

5.2.3. Usability Analysis

In this section, we will analyze the usability of the mechanisms, with the experimental scenario specifically designed in two parts, which we will elaborate on below.

Part One: Usability Analysis of Mechanisms.

We will compare the performance of the SPM mechanism with other federated learning privacy-protection mechanisms across three datasets, primarily examining the privacy budget used to achieve maximum model accuracy. This analysis aims to compare privacy-protection capabilities while ensuring model performance. A smaller privacy budget indicates stronger privacy-protection capabilities. For the low and high client scenarios, we selected median quantities as representatives, specifically 30 and 300, to validate performance under different scenarios. The parameters used for each mechanism remain consistent with those in the previous two sections, with specific values detailed in Tables 10 and 11.

In the low client scenario, there are significant differences in the privacy budgets required by each mechanism to achieve similar model accuracies. The SPM mechanism requires the lowest privacy budget, followed closely by PNPM, indicating that both mechanisms are suitable for scenarios with smaller privacy budgets. The Sun et al. [26] mechanism also performs well on the MNIST and Fashion-MNIST datasets, while the PM mechanism fails to achieve the model accuracy of other mechanisms on the CIFAR-10 dataset, and is therefore replaced with NaN. Furthermore, on other datasets, the privacy budget of the PM mechanism is excessively high, indicating that it is not suitable for low privacy budget scenarios. Under small privacy budgets, although the performance differences among the mechanisms are minor, they still reveal significant gaps, further demonstrating the advantages of the SPM mechanism in terms of privacy protection.

Table 10. Comparison of effectiveness of different mechanisms in the low client scenario (model accuracy).

Algorithm	MNIST	Fashion-MNIST	CIFAR-10 (Model 1)	CIFAR-10 (Model 2)
PM	88.57 ($\epsilon = 9$)	79.42 ($\epsilon = 13$)	Nan	80.12 ($\epsilon = 10$)
PNPM	88.97 ($\epsilon = 1$)	82.38 ($\epsilon = 1.5$)	77.64 ($\epsilon = 2.9$)	86.22 ($\epsilon = 6$)
Sun et al. [26]	88.09 ($\epsilon = 2$)	83.99 ($\epsilon = 2$)	74.87 ($\epsilon = 3.9$)	85.09 ($\epsilon = 7.5$)
SPM	89.25 ($\epsilon = 0.3$)	83.28 ($\epsilon = 0.6$)	76.08 ($\epsilon = 1.9$)	85.08 ($\epsilon = 3.9$)

In the multi-client scenario, we find that the privacy budgets required by each mechanism are generally larger, especially on the CIFAR-10 dataset. This is primarily due to the complexity of this dataset compared to others, necessitating smaller noise perturbations to maintain model performance. For the CIFAR-10 dataset, in Model I, the privacy budget required by the SPM mechanism is 2.9, while the minimum privacy budget for other mechanisms must reach 5.5. In Model II, the privacy budgets of the mechanisms are close, with SPM at 3.9, PNPM at 4.5, and Sun et al. at 8.5. This indicates that, compared to privacy budgets below 3, the differences among budgets exceeding 3 are not significant, and the privacy-protection capabilities provided are relatively weak. This further validates the effectiveness of our algorithm.

Table 11. Comparison of effectiveness of different mechanisms in the multi-client scenario (model accuracy).

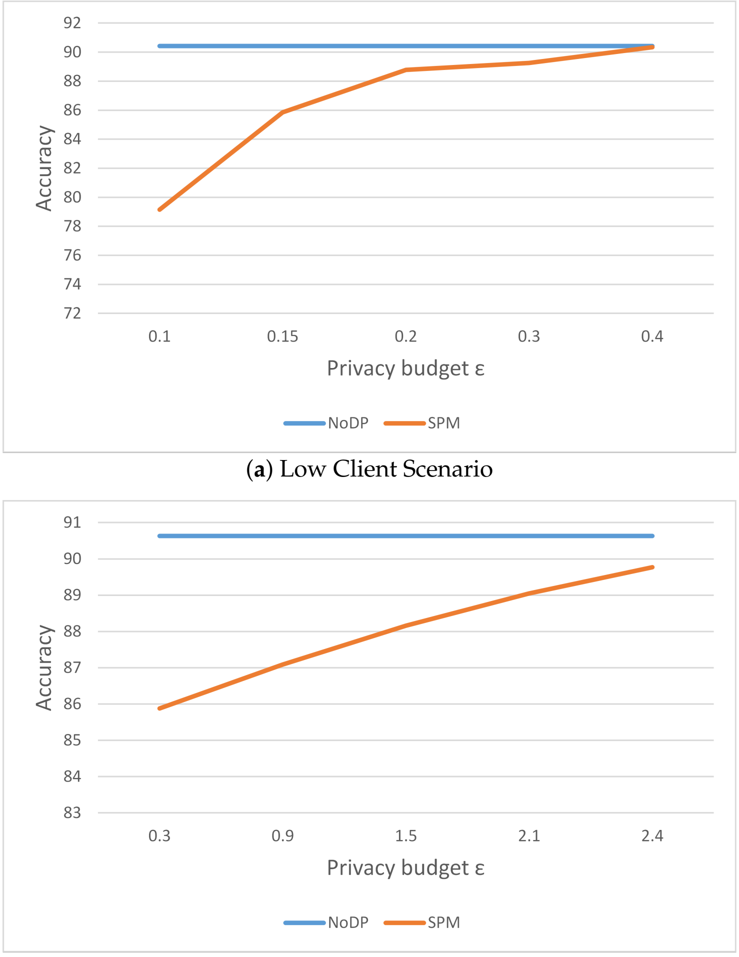
Algorithm	MNIST	Fashion-MNIST	CIFAR-10 (Model 1)	CIFAR-10 (Model 2)
PM	82.41 ($\epsilon = 15$)	77.59 ($\epsilon = 15$)	Nan	79.29 ($\epsilon = 10$)
PNPM	86.63 ($\epsilon = 0.9$)	79.62 ($\epsilon = 1.2$)	60.19 ($\epsilon = 5.5$)	78.69 ($\epsilon = 4.5$)
Sun et al. [26]	84.20 ($\epsilon = 1.5$)	81.20 ($\epsilon = 1.5$)	62.34 ($\epsilon = 7.5$)	78.76 ($\epsilon = 8.5$)
SPM	85.88 ($\epsilon = 0.3$)	80.4 ($\epsilon = 0.9$)	65.55 ($\epsilon = 2.9$)	79.24 ($\epsilon = 3.9$)

Part Two: Usability Analysis of the SPM Mechanism.

In this section, we will explore the usability analysis of the SPM mechanism. As described in Sections 5.2.1 and 5.2.2, the parameters used in these sections are based on a comprehensive selection of privacy-protection capabilities. This section will examine the differences between the SPM mechanism and the NoDP mechanism without noise processing when the privacy budget is appropriately increased, to demonstrate the usability of the SPM mechanism. We will use the NoDP values from the previous section and maintain the client numbers consistent with those in Part One of this section, selecting 30 and 300 as representatives. By comparing the performances of SPM and NoDP under different privacy budgets, we aim to reveal the balance between privacy protection and model accuracy in the SPM mechanism. We will analyze each dataset separately below.

For the MNIST dataset, we selected a privacy budget of 0.3 in Sections 5.2.1 and 5.2.2. In this section, our privacy budgets are set as follows: 0.1 to 0.4 for the low client scenario and 0.3 to 2.4 for the multi-client scenario. This is because in the multi-client scenario, small increases in the privacy budget do not lead to significant performance improvements, so the maximum privacy budget we selected is the minimum value that allows the model accuracy to approach that of NoDP.

As shown in Figure 11, in the low client scenario, selecting a privacy budget of 0.4 allows the performance of the SPM mechanism to approach that of NoDP. However, in the multi-client scenario, the privacy budget needs to be set to 2.4. We infer that in the multi-client case, selecting a privacy budget of 2.4 is a reasonable choice, but it also implies a decrease in privacy-protection capability. We will further evaluate the privacy-protection capability of the privacy budget values used in this section in Section 5.2.4.

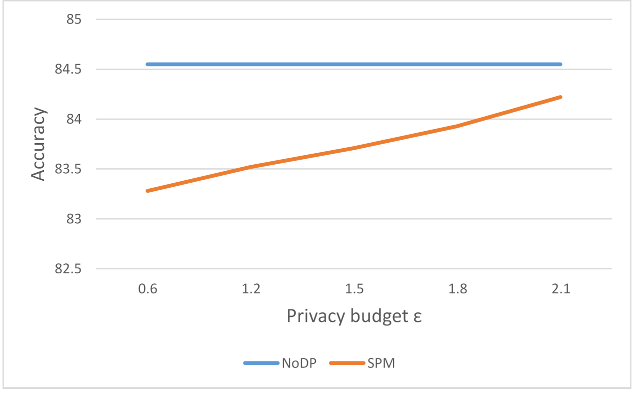


(**a**) Low Client Scenario

(**b**) Multi Client Scenario

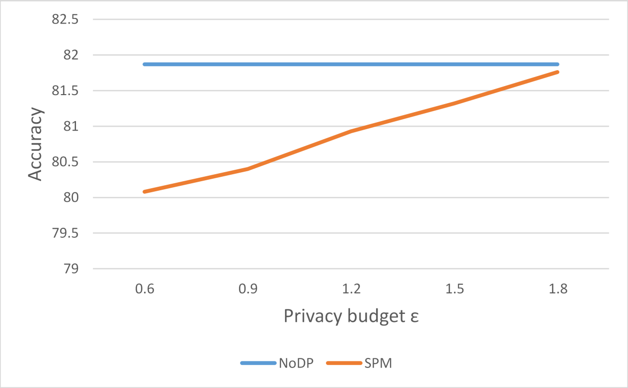
Figure 11. Variation trend of model accuracy under different privacy budgets in different client scenarios for the MNIST dataset.

In the Fashion-MNIST dataset, as shown in Figure 12, we selected a privacy budget range of 0.6 to 2.1 in the low client scenario. However, with a privacy budget of 2.1, the model accuracy of the SPM mechanism still shows a 1% gap compared to NoDP, but is significantly higher than the accuracy at 0.6, which is acceptable. Although we did not continue to increase the privacy budget, a moderate increase does improve model accuracy. Therefore, for this scenario, selecting a privacy budget of 2.1 can provide better performance, but it also reduces privacy-protection capability.



(**a**) Low Client Scenario

Figure 12. *Cont.*



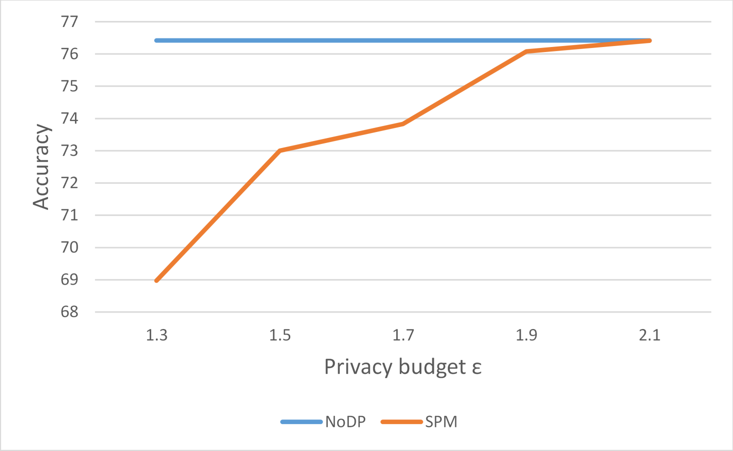
(**b**) Multi Client Scenario

Figure 12. Variation trend of model accuracy under different privacy budgets in different client scenarios for the Fashion-MNIST dataset.

In the multi-client scenario, we selected a privacy budget range of 0.6 to 1.8. We found that when the privacy budget is 1.8, the performance of the SPM mechanism approaches that of NoDP, and the overall accuracy shows an upward trend with increasing privacy budgets. This indicates that selecting a privacy budget of 1.8 is a good choice in the multi-client scenario. This is because, compared to the low client scenario, there are fewer locally allocated datasets in this scenario, so the chosen number of local iterations (set to 10) is adjusted to accommodate this change.

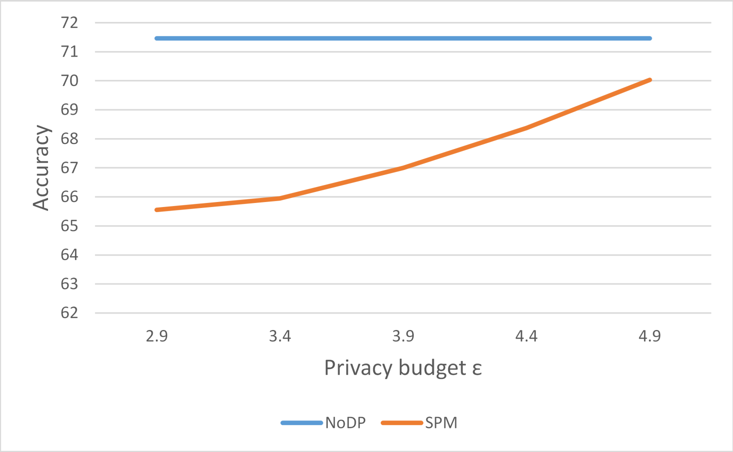
In CIFAR-10 (Model 1), as shown in Figure 13, due to the model being a shallow network, we set the privacy budget range as follows: 1.3 to 2.1 for the low client scenario and 2.9 to 4.9 for the multi-client scenario. In the low client scenario, when the privacy budget is 1.9 and 2.1, the model accuracy approaches that of NoDP, indicating that a privacy budget of 2.0 is a good choice, ensuring high performance while providing good privacy protection.

In the multi-client scenario, even with a higher privacy budget of 4.9, the accuracy of the SPM mechanism still shows an approximate 3% gap compared to NoDP. We believe this is primarily due to the model being shallow, along with an increase in the number of clients leading to a reduction in the local training set, which consequently decreases the content that can be learned. Even with an increased privacy budget, the model struggles to learn more useful information without being disrupted. Therefore, we decided not to continue increasing the privacy budget for further investigation.



(**a**) Low Client Scenario

Figure 13. *Cont.*



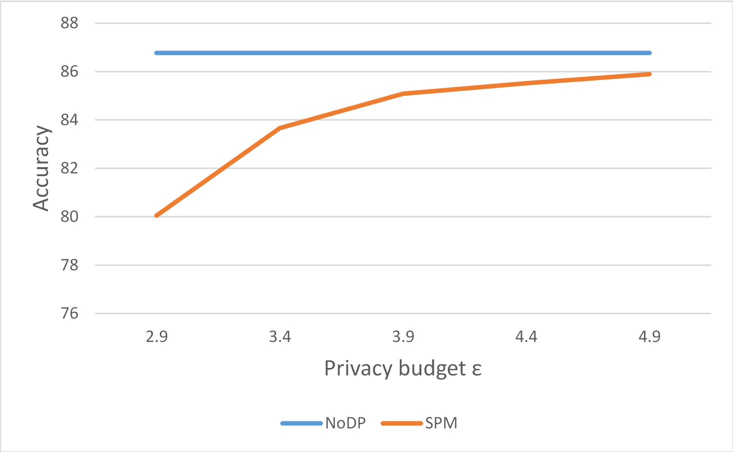
(**b**) Multi Client Scenario

Figure 13. Variation trend of model accuracy under different privacy budgets in different client scenarios for the CIFAR-10 dataset (Model 1).

In CIFAR-10 (Model II), as shown in Figure 14, we conducted additional experiments addressing the issue of being unable to learn more information in shallow networks in the multi-client scenario. By employing the deeper ResNet18 architecture, we found that under the same privacy budget (4.4), this model can provide better performance, with a gap of less than 0.2% compared to NoDP. Therefore, we set the privacy budgets as follows: 2.9 to 4.9 for the low client scenario and 2.4 to 4.4 for the multi-client scenario.

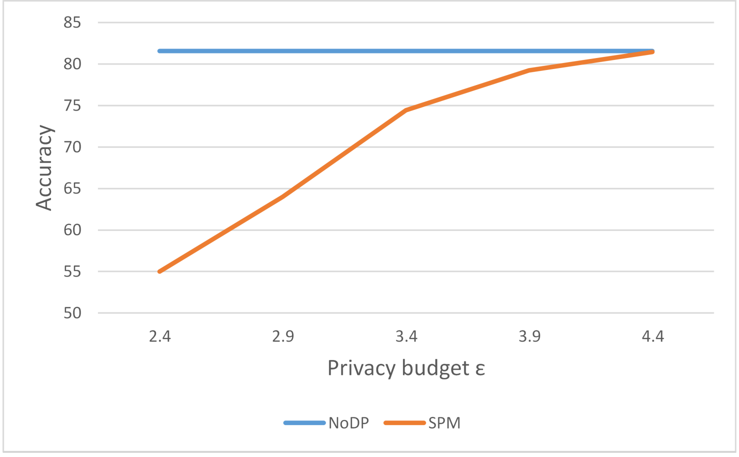
In the low client scenario, when the privacy budget is 4.4 and 4.9, the gap between SPM and NoDP can also be maintained within 1%. This further indicates that adopting a deeper network architecture is necessary for more complex datasets. However, this also implies that the time and privacy overhead incurred may increase.

Finally, based on the above observations, we conclude that appropriately increasing the privacy budget helps improve model accuracy to meet practical needs, but it may also lead to a decrease in privacy-protection capability. To this end, we will introduce DLG attacks in Section 5.2.4 to test the privacy-protection capability of the privacy budgets used in this section and conduct a comprehensive analysis of the optimal privacy budget for the SPM mechanism.



(**a**) Low Client Scenario

Figure 14. *Cont.*



(**b**) Multi Client Scenario

Figure 14. Variation Trend of model accuracy under different privacy budgets in different client scenarios for the CIFAR-10 dataset (Model 2).

5.2.4. Privacy-Protection Capability Evaluation

In this section, we selected three privacy budget values for the SPM mechanism in different client scenarios to assess its usability and privacy-protection capability. The first two privacy budget values are selected from the privacy budgets in Sections 5.2.1 and 5.2.2 to evaluate the level of privacy protection when balancing privacy-protection capability and performance, while the third privacy budget is chosen from the maximum privacy budgets for each dataset in Section 5.2.3 to measure the privacy-protection capability at maximum accuracy. We employed classic DLG attacks on the datasets corresponding to each privacy budget, recording the convergence count of the attacks as Attack T (AT), and selecting three phases: (1/3 AT, 2/3 AT, 3/3 AT) to observe the completion of the disguised images during the attacks. We use the following two metrics to evaluate the experimental results: (1) Structural Similarity Index (SSIM), which is used to assess the difference between the attack-reconstructed image and the original image; (2) DLG attack loss value, which measures the difference between the gradients of the original sample and the disguised sample.

The SSIM value ranges from 0 to 1, with values closer to 1 indicating higher image similarity, and 1 representing identical images. SSIM is calculated based on three aspects: luminance, contrast, and structure, which respectively measure the differences in mean luminance, contrast, and local structural similarity. These metrics help comprehensively evaluate the effectiveness of the attack and the performance of privacy protection. Next, we will independently analyze the experimental results of each dataset.

In the MNIST dataset, the privacy budgets we adopted are 0.3, 1.5, and 2.4. From Table 12 and Figure 15, it can be seen that as the number of attack rounds increases, the loss value decreases while the SSIM value continues to increase. With a privacy budget of 0.3, the provided privacy protection is the strongest, with a very high loss value of 119.6575 and a relatively low SSIM value of 0.4582, resulting in the final disguised image showing almost no features of the original image. This indicates that using a privacy budget of 0.3 can effectively protect privacy in the low client scenario. With a privacy budget of 1.5, the disguised image displays only a few features, making it difficult to distinguish obvious characteristics, indicating that the multi-client scenario also possesses strong privacy-protection capability. However, when the privacy budget is increased to 2.4, we find that the attacked disguised images exhibit numerous features, which can be used to distinguish the original training images after further processing, especially considering that this dataset is small, containing only ten classes. As mentioned in the conclusion of the previous section, increasing the privacy budget reduces privacy-protection capability, which also indirectly reflects the privacy-protection capability of the SPM mechanism.

Table 12. Effectiveness of DLG attacks on the model under different privacy budgets for the MNIST dataset.

Indicators	1/3 AT	2/3 AT	3/3 AT
Loss Value (ϵ = 0.3)	119.6662	119.6584	119.6575
SSIM Value (ϵ = 0.3)	0.1803	0.3468	0.4582
Loss Value (ϵ = 1.5)	3.1977	3.1946	3.2009
SSIM Value (ϵ = 1.5)	0.1910	0.3855	0.6319
Loss Value (ϵ = 2.4)	83.5240	83.5232	6.9459
SSIM Value (ϵ = 2.4)	0.1438	0.1169	0.6989

Figure 15 shows the variation trend of disguised images under different privacy budgets in the MNIST dataset.

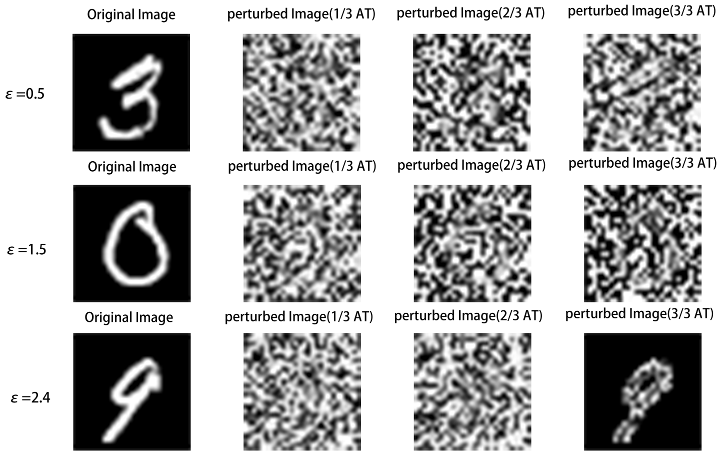


Figure 15. DLG attack effect on the model under different privacy budgets for the MNIST dataset.

In the Fashion-MNIST dataset, as shown in Figure 16, the protection capability of the SPM mechanism is relatively unstable. In the final iteration, the disguised images under all privacy budgets displayed some features, and the prominence of these features increased with the privacy budget. As shown in Table 13, it can be seen that, except for the case where the privacy budget is 0.6, where the loss value does not converge, the loss values for other privacy budgets converge well, and the SSIM values can reach around 80%. This indicates that in this dataset, if we wish to enhance the privacy-protection capability of the SPM mechanism, it is necessary to consider reducing the privacy budget or employing deeper neural networks.

Table 13. Effectiveness of DLG attacks on the model under different privacy budgets for the Fashion-MNIST dataset.

Indicators	1/3	2/3	3/3
Loss Value (ϵ = 0.6)	717.1729	717.1757	717.2051
SSIM Value (ϵ = 0.6)	0.3830	0.2078	0.1613
Loss Value (ϵ = 1.2)	0.2467	0.2147	0.2102
SSIM Value (ϵ = 1.2)	0.2336	0.5325	0.7217
Loss Value (ϵ = 1.8)	1.3854	0.9854	0.7643
SSIM Value (ϵ = 1.8)	0.2985	0.1425	0.8168

Figure 16 shows the variation trend of disguised images under different privacy budgets in the Fashion-MNIST dataset.

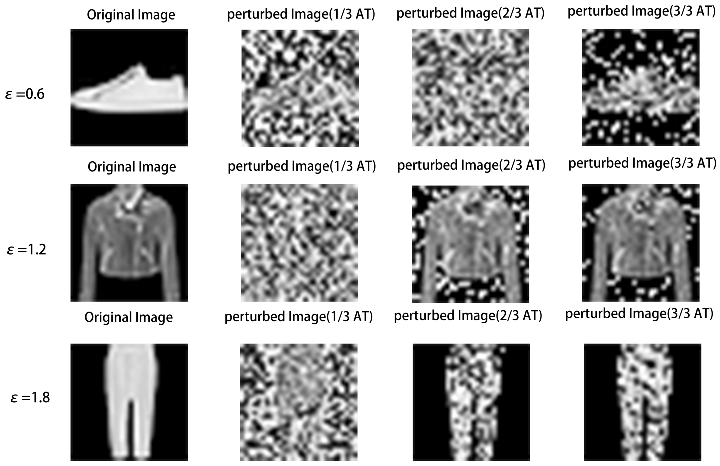


Figure 16. DLG attack effect on the model under different privacy budgets for the Fashion-MNIST dataset.

In the CIFAR-10 dataset (Model 1), the privacy budgets were increased to 1.9, 2.9, and 4.9 due to the dataset consisting of color images. As shown in Table 14, the loss values and SSIM values under each privacy budget did not exhibit non-convergence. However, with a privacy budget of 1.9, the disguised images displayed almost no features. As the privacy budget increased, the images gradually revealed some features, but there was still considerable noise interference, resulting in blurry images. This indicates that training the SPM mechanism on this dataset using Model 1 is advisable, demonstrating good privacy-protection capability. Furthermore, the optimal choice of privacy budget aligns with those selected in Sections 5.2.1 and 5.2.2.

Table 14. Effectiveness of DLG attacks on the model under different privacy budgets for the CIFAR-10 dataset (Model 1).

Indicators	1/3	2/3	3/3
Loss Value (ϵ = 1.9)	10.1564	7.9269	7.5234
SSIM Value (ϵ = 1.9)	0.1641	0.5540	0.6650
Loss Value (ϵ = 2.9)	2.6184	1.6189	1.4455
SSIM Value (ϵ = 2.9)	0.3035	0.7160	0.8255
Loss Value (ϵ = 4.9)	0.5326	0.3208	0.2914
SSIM Value (ϵ = 4.9)	0.8073	0.8888	0.9293

Figure 17 shows the variation trend of disguised images under different privacy budgets in the CIFAR-10 dataset (Model 1).

In the CIFAR-10 dataset (Model 1), as shown in Table 15, we used a deeper network structure and relatively low privacy budgets of 3.9, 4.4, and 4.9. With a privacy budget of 3.9, we observed a trend consistent with the SPM mechanism, where the colors of the disguised images were completely opposite to those of the original images, indicating a reversal of the color channels in the attacked images. This phenomenon reflects the disturbance method of our mechanism, where applying directional perturbations after desensitizing the gradient weights can effectively disrupt the direction of DLG attacks, thereby protecting the original data while maintaining high accuracy. For the settings of other privacy budgets, we found it difficult to obtain useful information. As seen in Table 1, the SSIM values for each privacy budget are relatively low, with the highest being only around 70%. This indicates that, despite the higher privacy budgets, the metrics have limitations across multiple dimensions, with actual pixel similarity being only about one-third of the current value.

We believe that even with higher privacy budgets, the combination of the SPM mechanism and deep neural networks maintains privacy-protection capability. Therefore, for complex datasets, although shallower networks can provide smaller privacy budget settings, deep neural networks can still maintain similar privacy-protection capabilities even with larger privacy budgets, albeit with increased time costs.

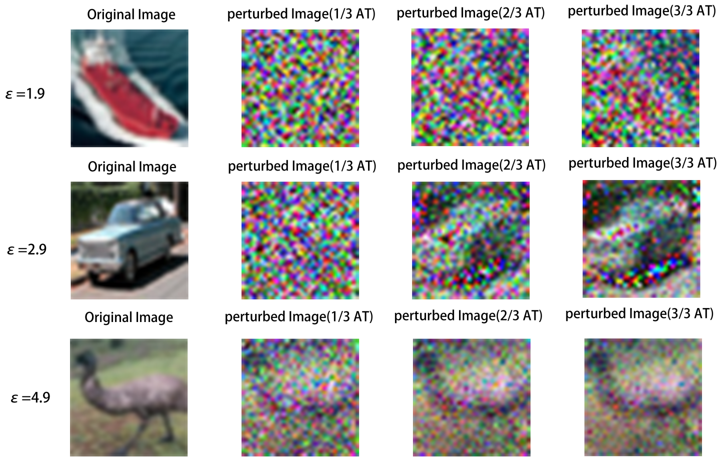


Figure 17. DLG attack effect on the model under different privacy budgets for the CIFAR-10 dataset (Model 1).

Table 15. Effectiveness of DLG attacks on the model under different privacy budgets for the CIFAR-10 dataset (Model 2).

Indicators	1/3	2/3	3/3
Loss Value (ϵ = 3.9)	9.6695	8.9784	11.0653
SSIM Value (ϵ = 3.9)	0.4942	0.6270	0.6768
Loss Value (ϵ = 4.4)	10.5954	10.3584	10.5718
SSIM Value (ϵ = 4.4)	0.5516	0.6997	0.7175
Loss Value (ϵ = 4.9)	5.6114	3.9893	4.5776
SSIM Value (ϵ = 4.9)	0.6179	0.7271	0.7234

Figure 18 shows the variation trend of disguised images under different privacy budgets in the CIFAR-10 dataset (Model 2).

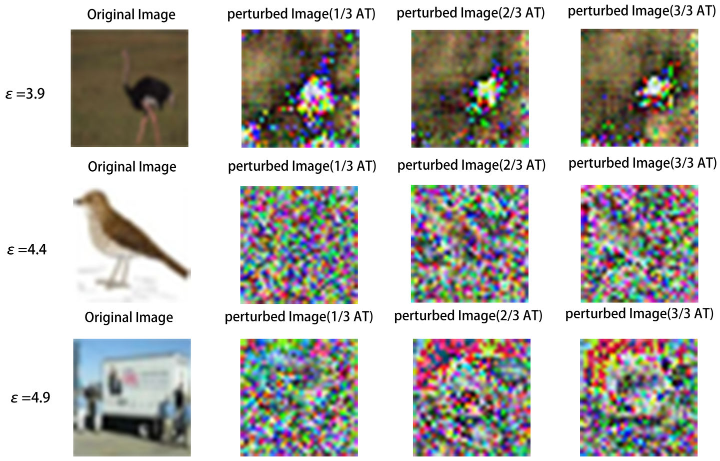


Figure 18. DLG attack effect on the model under different privacy budgets for the CIFAR-10 dataset (Model 2).

Figure 18 shows the variation trend of disguised images under different privacy budgets in the CIFAR-10 dataset (Model 2). In conclusion, we summarize as follows: The SPM mechanism implements positive and negative disturbances by taking the absolute value of the model gradient parameters and multiplying by a disturbance coefficient, achieving a desensitization effect that meets strict differential privacy requirements. Combined with the characteristics of DLG attacks, this mechanism can effectively reduce the likelihood of successful attacks. First, the positive and negative disturbances increase the randomness of the model gradients, making it difficult for attackers to extract useful information. Second, the dynamic adjustment of the disturbance coefficient can enhance resistance to specific attack patterns, adapting to different attack environments. Furthermore, the design of the SPM mechanism ensures that the leakage of sensitive information during gradient updates is significantly reduced, thereby suppressing the effectiveness of DLG attacks. Overall, the SPM mechanism not only protects data privacy but also effectively mitigates the risk of DLG attacks by enhancing the fuzziness and randomness of the gradients.

Furthermore, we analyzed how the SPM mechanism effectively reduces the risk of membership inference attacks. Membership inference attacks attempt to infer whether specific data points were used in training by analyzing the model outputs, while the SPM mechanism applies noise and processes gradients in absolute value, making the model outputs more random and making it difficult for attackers to extract clear membership information. Notably, the SPM mechanism is particularly suitable for multi-client scenarios. In multi-client environments, the datasets used by each client are usually smaller, and the diversity of data is relatively low, which makes the features learned by the attack models trained on shadow datasets used in membership inference attacks more difficult to discern against the features of our model, thus reducing the effectiveness of membership inference attacks.

The design of the SPM mechanism not only protects user data privacy but also significantly reduces the risk of the model facing various types of attacks by enhancing the fuzziness of the model outputs.

5.2.5. Impact of Local Iteration Count on Communication Overhead

In this section, we conducted an internal exploration of the SPM mechanism to investigate whether "increasing local iterations can reduce global communication costs at the expense of local time overhead". We selected the parameter settings from Sections 5.2.1 and 5.2.2 as a control, appropriately reducing the global communication rounds T while increasing the local iterations E, ensuring that the privacy budget remained unchanged. This was done to compare the model accuracy against the original parameter settings and to analyze the changes in other parameters while maintaining a similar model accuracy and ensuring the effectiveness of privacy protection. The client scenarios were set to 30 and 300 clients, respectively. Next, we will analyze the experimental results one by one.

In the MNIST dataset, as shown in Table 16, we increased the local iterations E from 3 to 5, 7, 9, and 11 for the small client scenario, and to 11, 13, 15, and 17 for the multi-client scenario, while reducing the global communication rounds T to 20. As seen in Table 1, appropriately increasing the local iterations not only maintains the original model accuracy but also leads to some improvements, while significantly reducing global communication costs. Therefore, for the SPM mechanism, this paper recommends adopting the parameter settings shown in the table for this dataset to achieve higher communication efficiency and model performance.

Table 16. Analysis of Model accuracy under different parameter settings for the MNIST dataset.

C = 30, ϵ = 0.3	**E = 3**	**E = 5**	**E = 7**	**E = 9**	**E = 11**
SPM (T = 50)	89.25	-	-	-	-
SPM (T = 20)	-	90.02	89.77	89.41	89.66
C = 300, ϵ = 0.3	**E = 3**	**E = 11**	**E = 13**	**E = 15**	**E = 17**
SPM (T = 50)	85.88	-	-	-	-
SPM (T = 20)	-	87.83	88.05	88.47	87.74

In the Fashion-MNIST dataset, as shown in Table 17, we found that reducing the local iterations, for example setting E = 2 and simultaneously decreasing the global communication rounds T, led to a decline in model accuracy from the original 83.28 to 82.80. However, when E = 7 or higher, the model accuracy remains stable, while communication costs are significantly reduced. This conclusion applies to both client scenarios in this dataset. However, as the number of local iterations increases, the computational time overhead also rises rapidly, which is another important issue that requires careful consideration.

Table 17. Analysis of model accuracy under different parameter settingsfor the Fashion-MNIST dataset.

C = 30, ϵ = 0.6	**E = 3**	**E = 2**	**E = 5**	**E = 7**	**E = 9**
SPM (T = 50)	83.28	-	-	-	-
SPM (T = 30)	-	82.80	82.95	83.35	83.08
C = 300, ϵ = 0.9	**E = 10**	**E = 12**	**E = 14**	**E = 16**	**E = 18**
SPM (T = 20)	80.40	-	-	-	-
SPM (T = 15)	-	80.94	81.00	81.30	81.24

In the CIFAR-10 dataset (Model 2), as shown in Table 18, we found that reducing the number of iterations did not yield significant results for complex network models. For example, increasing the number of iterations from E = 5 to E = 8 resulted in negligible improvement in model accuracy, while reducing the global communication rounds T led to a decrease in model accuracy, which contradicts the purpose of this experiment. We believe this is due to the deeper internal connections of the network, which are already capable of fully learning feature information under the current iteration settings. Therefore, this approach cannot effectively reduce communication overhead for this dataset. We will explore more strategies to optimize communication overhead in future research.

Table 18. Analysis of model accuracy under different parameter settings for the CIFAR-10 dataset (Model 2).

C = 30, ϵ = 3.9	**E = 5**	**E = 6**	**E = 7**	**E = 8**	**E = 9**
SPM (T = 50)	85.08	-	-	-	-
SPM (T = 30)	-	84.82	84.78	85.16	84.37
C = 300, ϵ = 3.9	**E = 10**	**E = 12**	**E = 14**	**E = 16**	**E = 18**
SPM (T = 50)	79.24	-	-	-	-
SPM (T = 30)	-	79.51	79.34	79.06	79.13

6. Discussion

In this chapter, we discuss the applications of our research findings, current limitations, and future directions for improvement, providing guidance for subsequent research.

6.1. Advantages and Limitations of the Existing Mechanism

The proposed Symmetric Piecewise Mechanism combines differential privacy and zero bias design, successfully balancing model performance and privacy protection. However, there are some limitations when dealing with open-set scenarios and addressing data imbalance. In open-set classification scenarios, the model needs to make reasonable predictions for inputs not belonging to the training set. The recently proposed FedPD algorithm solves the federated open-set recognition problem through parameter disentanglement, providing a new approach to enhancing model generalization in open-set cases [33]. In the future, we plan to apply the idea of parameter disentanglement to the SPM mechanism to improve the model's ability to recognize new classes of data.

Furthermore, federated learning faces challenges in handling imbalanced data, especially in the medical field where data imbalance is particularly severe. Reference [34] proposes a personalized federated learning approach that uses personalization and anti-degradation strategies to tackle the challenge of imbalanced data. We plan to combine these methods with the SPM mechanism to improve the robustness and generalization capability of the model in handling uneven data across different clients.

6.2. Application of Blockchain Technology in Federated Learning

In terms of the security of federated learning, this study reduces the negative impact of perturbation on global model performance through the SPM mechanism and zero bias design, maintaining statistical unbiasedness of the model. However, we recognize that there is still room for improvement in protecting model integrity. Reference [7] indicates that blockchain technology has significant potential in enhancing the security and transparency of federated learning.

The tamper-resistant and distributed nature of blockchain allows each client's contribution to be recorded and tracked, enabling verifiability and auditability of the model update process. We believe that blockchain technology, such as Vfchain, can enhance model integrity and can be combined with our SPM mechanism to achieve auditability and trustworthiness in federated learning in future research. However, effectively integrating blockchain technology into the federated learning framework is not a trivial task, requiring thorough architectural design and performance optimization. Therefore, we remain cautiously optimistic about the integration of these technologies.

6.3. Future Research Directions

Based on existing research and the proposed SPM mechanism, future research directions mainly include the following:

Enhancement of open-set recognition and generalization ability: Drawing inspiration from the parameter disentanglement method of FedPD, applying it to federated learning to enhance the model's generalization ability in open-set scenarios.

Solutions for personalization and data imbalance: Combining personalized learning mechanisms to address the uneven distribution of client data. In sensitive domains such as healthcare, imbalanced data can severely affect model performance. We refer to Reference [34] to explore personalized anti-degradation mechanisms to improve model robustness in complex data scenarios.

Integration of blockchain technology: Continue exploring how to apply blockchain technology to federated learning to enhance data traceability and model security. Technologies like Vfchain have great potential for improving federated learning security. The decentralized, tamper-resistant, and transparent nature of blockchain can provide additional guarantees for model integrity, making the federated learning process verifiable and auditable.

Through the above discussion, we hope to provide new ideas for future federated learning research, especially in terms of security and adaptability, to further enhance the practicality and reliability of federated learning.

7. Conclusions

This paper proposes a Symmetric Piecewise Mechanism to probabilistically perturb local model weights before aggregation, thereby meeting strict ϵ-differential privacy requirements. Additionally, we propose a variance constraint mechanism to mitigate the impact of noise on model performance and introduce a zero bias design to ensure that the aggregated global model retains its original performance as much as possible. Experimental results show that SPM exhibits better usability and privacy protection compared to other mechanisms under different numbers of clients, particularly in defending against DLG attacks.

Despite significant progress, some issues remain unresolved, especially in terms of privacy loss measurement and data heterogeneity. In the future, we aim to develop more precise privacy budget-management methods and explore ways to improve the robustness and adaptability of the model in heterogeneous data environments.

Author Contributions: Conceptualization, Z.C. and H.Z.; methodology, Z.C. and H.Z.; validation, H.Z.; investigation, Z.C.; data curation, Z.C.; writing—original draft preparation, Z.C.; writing—review and editing, Z.C. and H.Z.; supervision, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof

Theorem A1. ***SPM Mechanism and Differential Privacy.*** *When the boundary C of the perturbation coefficient and the probability density p of the perturbation value satisfy Equation (3), the SPM mechanism can ensure that the model parameters uploaded by clients participating in federated learning training meet ϵ-differential privacy. Moreover, this mechanism ensures zero bias during mean estimation of the weights and minimizes the dispersion (variance) of the model weights after being processed by the perturbation mechanism.*

Proof. In this paper, S represents the perturbation mechanism, ω denotes the model weights, t^* is the perturbation coefficient, and $S(\omega) = S(|\omega| \cdot t^*)$ represents the model weights after perturbation. We calculate the expected value of the weights after perturbation:

$$\mathrm{Var}[S(\omega)] = \mathrm{Var}[|\omega| \cdot t^*] = |\omega|^2 \mathrm{Var}[t^*] \tag{A1}$$

Since $|\omega|^2 \geq 0$, the magnitude of $\mathrm{Var}[S(\omega)]$ changes with $\mathrm{Var}[t^*]$. Therefore, we only need to discuss and compute $\mathrm{Var}[t^*]$, yielding:

$$\mathrm{Var}[t^*] = E[(t^*)^2] - (E[t^*])^2 \tag{A2}$$

(1) Compute $E[t^*]$

$E[t^*]$ is the expectation of t^*, where $l(t) = \frac{(C+1)}{2} \cdot t - \frac{(C-1)}{2}$, $r(t) = \frac{(C+1)}{2} \cdot t + \frac{(C-1)}{2}$

$$E[t^*] = \int_{-r(t)}^{-l(t)} \frac{P}{e^\epsilon} x \, dx + \int_{l(t)}^{r(t)} P x \, dx = \frac{P(r(t)^2 - l(t)^2)(e^\epsilon - 1)}{2e^\epsilon} = \frac{P(C^2-1)(e^\epsilon - 1)}{2e^\epsilon} \cdot t \tag{A3}$$

Let $P = \frac{(e^\epsilon - 1)e^\epsilon}{e^\epsilon + \alpha} \cdot P'$ be the probability density, with bounds $C = \frac{(e^\epsilon + \beta)}{e^\epsilon - 1} \cdot C'$, where P', C', α, β are custom parameters that may be constants or polynomials containing e^ϵ.

$$E[t^*] = P' \left(\frac{(C'^2 - 1)e^{2\epsilon} + (2C'^2\beta + 2)e^\epsilon + C'^2\beta^2 - 1}{2(e^\epsilon + \alpha)} \right) \cdot t \tag{A4}$$

Assume $E[t^*] = t$, implying there are no terms involving $O(\epsilon^2)$ in the numerator, resulting in $C'^2 - 1 = 0$, $|C'| = 1$.

$$E[t^*] = P' \left(\frac{2(\beta + 1)e^\epsilon + \beta^2 - 1}{2(e^\epsilon + \alpha)} \right) \cdot t \tag{A5}$$

The denominator is a linear term in ϵ, and the numerator is adjusted to be linear in ϵ by extracting coefficients.

$$E[t^*] = (\beta + 1)P' \left(\frac{e^\epsilon + \frac{\beta - 1}{2}}{e^\epsilon + \alpha} \right) \cdot t \tag{A6}$$

We conclude that when the following relationship exists, $E[t^*] = t$.

$$\begin{cases} (\beta+1)P' = 1 \\ \beta - 1 = 2\alpha \end{cases} \tag{A7}$$

(2) Compute $E[(t^*)^2]$
$E[(t^*)^2]$ is the expectation of t^* squared:

$$E[(t^*)^2] = \int_{-r(t)}^{-l(t)} \frac{P}{e^{\epsilon}} x^2\,dx + \int_{l(t)}^{r(t)} Px^2\,dx = \frac{P(r(t)^3 - l(t)^3)(e^{\epsilon}+1)}{3e^{\epsilon}} \tag{A8}$$

It can be inferred from the above $l(t)$ and $r(t)$.

$$r(t)^3 - l(t)^3 = (C-1)\left(\frac{3}{4}(C+1)^2 + \frac{1}{4}(C-1)^2\right) = C^3 - 1 \tag{A9}$$

Substituting $P = \frac{(e^{\epsilon}-1)e^{\epsilon}}{e^{\epsilon}+\alpha} \cdot P'$ yields:

$$E[(t^*)^2] = P' \frac{(e^{\epsilon}-1)(e^{\epsilon}+1)}{3(e^{\epsilon}+\alpha)}(C^3-1) \tag{A10}$$

Substituting $C = \frac{(e^{\epsilon}+\beta)}{e^{\epsilon}-1} \cdot C'$ yields:

$$E[(t^*)^2] = P' \frac{(e^{\epsilon}+1)\big((C'^3-1)e^{3\epsilon} + 3(\beta C'^3+1)e^{2\epsilon} + 3(\beta^2 C'^3 - 1)e^{\epsilon} + \beta^3 C'^3 + 1\big)}{3(e^{\epsilon}+\alpha)(e^{\epsilon}-1)^2} \tag{A11}$$

From $|C'| = 1$, since the selected perturbation domain is a symmetric piecewise interval with no overlap between intervals, the interval $[-1, 1]$ is not considered. $C' = -1$ would cause the intervals to intersect, violating the initial setup. Thus, taking $C' = 1$, we have:

$$E[(t^*)^2] = P' \frac{(e^{\epsilon}+1)\big(3(\beta+1)e^{2\epsilon} + 3(\beta^2-1)e^{\epsilon} + \beta^3 + 1\big)}{3(e^{\epsilon}+\alpha)(e^{\epsilon}-1)^2} \tag{A12}$$

From Equation (A1), it is known that when $E[t^*]$ is at $t = \{-1, 1\}$, and Equation (A7) is satisfied, $E[t^*] = t$, $E[t^*]^2 = 1$. In this case, $\mathrm{Var}[t^*]$ is:

$$\mathrm{Var}[t^*] = P' \frac{(e^{\epsilon}+1)\big(3(\beta+1)e^{2\epsilon} + 3(\beta^2-1)e^{\epsilon} + \beta^3 + 1\big)}{3(e^{\epsilon}+\alpha)(e^{\epsilon}-1)^2} - 1 \tag{A13}$$

Considering minimizing $\mathrm{Var}[t^*]$, it suffices to focus on $E[(t^*)^2]$. Substituting Equation (A7) into Equation (A11) gives:

$$\begin{aligned} E[(t^*)^2] &= \frac{(e^{\epsilon}+1)\big(6(\alpha+1)e^{2\epsilon} + 12(\alpha^2+\alpha)e^{\epsilon} + 8\alpha^3 + 12\alpha^2 + 6\alpha + 2\big)}{6(e^{\epsilon}+\alpha)(e^{\epsilon}-1)^2(e^{\epsilon}+1)} \\ &= \frac{(e^{\epsilon}+1)\left((e^{\epsilon}+\alpha)^2 + \frac{(\alpha+1)^2}{3}\right)}{(e^{\epsilon}+\alpha)(e^{\epsilon}-1)^2} = \frac{(e^{\epsilon}+1)\left((e^{\epsilon}+\alpha) + \frac{(\alpha+1)^2}{3(e^{\epsilon}+\alpha)}\right)}{(e^{\epsilon}-1)^2} \\ &\geq \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)^2} \cdot 2\sqrt{\frac{(\alpha+1)^2}{3}} = \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)^2} \cdot \frac{2\sqrt{3}}{3} \cdot |\alpha+1| \end{aligned} \tag{A14}$$

From the inequality $|\alpha+1| \ll |\alpha| + 1$, we obtain:

$$E[(t^*)^2] \geq \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)^2} \cdot \frac{2\sqrt{3}}{3} \cdot (|\alpha| + 1) \tag{A15}$$

Since $\frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)^2} > 0$ and $|\alpha + 1| \gg 0$, minimizing $|\alpha| + 1$ is sufficient to achieve the minimum value of $\mathrm{Var}[t^*]$. Given $|\alpha| \ll 0$, α is set to 0. Substituting these values into Equation (A7) yields a series of parameter values.

$$\begin{cases} \alpha = 0 \\ \beta = 1 \\ P' = \frac{1}{2} \\ C' = 1 \end{cases} \qquad \begin{cases} P = \frac{(e^{\epsilon}-1)}{2} \\ C = \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)} \end{cases} \tag{A16}$$

In summary, when the above parameters satisfy Equation (A16), $E[t^*] = t$, $\mathrm{Var}[t^*]$ is minimized, leading to the minimum variance $\mathrm{Var}[S(\omega)]$ applied to the weights.

$$\mathrm{Var}[t^*] = \frac{(e^{\epsilon}+1)\left(e^{\epsilon} + \frac{1}{3e^{\epsilon}}\right)}{(e^{\epsilon}-1)^2} - 1 = \frac{(3e^{\epsilon} + \frac{1}{3e^{\epsilon}} - \frac{2}{3})}{(e^{\epsilon}-1)^2} \tag{A17}$$

$$\mathrm{Var}[S(\omega)] = \mathrm{Var}[|\omega| \cdot t^*] = |\omega|^2 \mathrm{Var}[t^*] = |\omega|^2 \cdot \frac{(3e^{\epsilon} + \frac{1}{3e^{\epsilon}} - \frac{2}{3})}{(e^{\epsilon}-1)^2} \tag{A18}$$

□

Lemma A1. *The variance of the SPM mechanism is strictly smaller than the variance of the Laplace mechanism and the variances in the literature [27,29], and it is independent of the value of the privacy budget ε.*

Proof. This lemma is compared with the variances in references [27,29]. Since the model weights in reference [28] are adaptively selected, it is impossible to calculate a fixed variance for comparison.

Proof (1): Regardless of the privacy budget value, the variance of SPM is always smaller than the variance of PM.

Given that the variance of the SPM mechanism is $|\omega|^2 \cdot \frac{3e^{\epsilon}+\frac{1}{3e^{\epsilon}}-\frac{2}{3}}{(e^{\epsilon}-1)^2}$ and the variance of the PM mechanism is $\frac{\omega^2}{e^{\epsilon/2}-1} + \frac{e^{\epsilon/2}+3}{3(e^{\epsilon/2}-1)^2}$, let $y = |\omega|^2 \cdot \left(\frac{3e^{\epsilon}+\frac{1}{3e^{\epsilon}}-\frac{2}{3}}{(e^{\epsilon}-1)^2}\right) - \left(\frac{\omega^2}{e^{\epsilon/2}-1} + \frac{e^{\epsilon/2}+3}{3(e^{\epsilon/2}-1)^2}\right) = \omega^2 \cdot \left(\frac{3e^{\epsilon}+\frac{1}{3e^{\epsilon}}-\frac{2}{3}-(e^{\epsilon/2}+1)(e^{\epsilon}-1)}{(e^{\epsilon}-1)^2}\right) - \frac{e^{\epsilon/2}+3}{3(e^{\epsilon/2}-1)^2}$. Let $e^{\epsilon/2} = x$, since $\epsilon \in (0, +\infty)$, it follows that $x \in (1, +\infty)$. $y = |\omega|^2 \cdot \frac{-x^3+2x^2+x+\frac{1}{3x^2}+\frac{1}{3}}{(x^2-1)^2} - \frac{x+3}{3(x-1)^2}$. Let $f(x) = -x^3 + 2x^2 + x + \frac{1}{3x^2} + \frac{1}{3}$. The first derivative is $f'(x) = -3x^2 + 4x - \frac{2}{3x^3} + 1$, the second derivative is $f''(x) = -6x + \frac{2}{x^4} + 4$, and the third derivative is $f'''(x) = -\frac{8}{x^5} - 6$. Because $x \in (1, +\infty)$, and x^5 in $f'''(x)$ does not change its sign, we have $f'''(x) < 0$. Thus, $f''(x)$ is monotonically decreasing for $x \in (1, +\infty)$. Since $f''(1) = 0$, it follows that $f''(x) < 0$. Therefore, for $x \in (1, +\infty)$, $f'(x)$ is a monotonically decreasing function. Given $f'(1) = \frac{4}{3} > 0$, and since x^2 dominates in $f'(x)$ and does not change its sign, we have $\lim_{x \to +\infty} f'(x) = -\infty < 0$. As $f'(x)$ is a polynomial function that is continuous over its domain and is a linear combination of a finite number of continuous functions, the linear combination of continuous functions is also continuous and differentiable everywhere. By the Intermediate Value Theorem, there exists at least one zero point c in the interval $(1, +\infty)$ such that $f'(c) = 0$. Thus, $f(x)$ is monotonically increasing for $x \in (1, c)$ and monotonically decreasing for $x \in (c, +\infty)$, leading to $f(c) > f(1) = \frac{8}{3} > 0$. Since $\lim_{x \to +\infty} f(x) = -\infty < 0$, by the Intermediate Value Theorem, there exists a unique zero point d in the interval $(c, +\infty)$ such that $f(d) = 0$.

By Newton's iteration method, it is known that initializing $x_0 = 2$, $f(x_0) = f(2) = \frac{29}{12}$, $f'(x_0) = f(2)' = -\frac{37}{12}$, updating $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = \frac{103}{37}$, $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$, repeating this process until $|x_{n+1} - x_n| \ll 10^{-6}$, finally obtaining $x \approx 2.4687$, i.e., when $d \approx 2.4687$, $f(d) = 0$. That is, for $x \in (1, d)$, $f(x) > 0$, and for $x \in (d, +\infty)$, $f(x) < 0$.

Because $f(x)$ has positive and negative intervals when $x \in (1, +\infty)$, it is discussed separately. Divide $y = |\omega|^2 \cdot \frac{-x^3+2x^2+x+\frac{1}{3x^2}+\frac{1}{3}}{(x^2-1)^2} - \frac{x+3}{3(x-1)^2}$ into two parts, let $A = |\omega|^2 \cdot \frac{-x^3+2x^2+x+\frac{1}{3x^2}+\frac{1}{3}}{(x^2-1)^2}$ and $B = \frac{x+3}{3(x-1)^2}$, and then analyze the positive and negative values of $A - B$ in different intervals.

- When $x \in (d, +\infty)$, $f(x) = -x^3 + 2x^2 + x + \frac{1}{3x^2} + \frac{1}{3} < 0$, $(e^\epsilon - 1)^2 > 0$, and since $|\omega|^2 > 0$, it follows that $A < 0$. In B, $3(x-1)^2 > 0$ and $x + 3 > 0$, which makes $B > 0$. Subtracting B from A yields a negative result, hence $y < 0$.
- When $x \in (1, d)$, $y = A - B = |\omega|^2 \cdot \frac{(-x^3+2x^2+x+\frac{1}{3x^2}+\frac{1}{3}) - \frac{(x+3)}{3\cdot(x+1)^2}}{(x^2-1)^2}$ where $|\omega|^2 = C$, and $C \in [0, 1]$. Combined, we have $y = \frac{((-C-\frac{1}{3})x^3+(2C-\frac{4}{3})x^2+(C-\frac{7}{3})x+\frac{C}{3x^2}+(\frac{C}{3}-1))}{((x-1)^2(x+1)^2)}$. Let the numerator be denoted as $h(x)$. Differentiating it, the first derivative $h'(x) = (-3C-1)x^2 + (4C - \frac{10}{3})x - \frac{2C}{3x^3} + (C - \frac{7}{3})$, the second derivative $h''(x) = (-6C - 2)x + \frac{2C}{x^4} + 4C - \frac{10}{3}$, and the third derivative $h'''(x) = -\frac{8C}{x^5} - 6C - 2$. For $x \in (1, d)$ and x^5 maintaining its sign, we have $h'''(x) < 0$. $h''(x)$ is a monotonically decreasing function. Since $h''(1) = -\frac{16}{3} < 0$, it follows that $h''(x)$ is always negative, and $h'(x)$ is a monotonically decreasing function. Given $h'(1) = \frac{4}{3} \cdot C - \frac{20}{3}$ and $C \in [0, 1]$, we have $h'(1) < 0$, $h'(x)$ is always negative, and $h(x)$ is a monotonically decreasing function. Thus, $h(1) = \frac{8}{3} \cdot C - \frac{16}{3} < 0$, implying $h(x) < 0$. Since the numerator is negative and the denominator $(x-1)^2(x+1)^2 > 0$, we conclude $x \in (1, d) \subset (1, +\infty)$, hence $y < 0$.

In conclusion, when $x \in (1, +\infty)$, the variance of the SPM mechanism is consistently less than that of the PM mechanism.

Proof (2): Regardless of the privacy budget value, the variance of the SPM mechanism is always less than the variance of the PNPM mechanism.

It is known that the variance of the SPM mechanism is denoted as $|\omega|^2 \cdot \frac{3e^\epsilon+\frac{1}{3e^\epsilon}-\frac{2}{3}}{(e^\epsilon-1)^2}$, and the variance of the PNPM mechanism is denoted as $|\omega|^2 \cdot \frac{4(e^\epsilon+\frac{1}{3})}{(e^\epsilon-1)^2}$. Let $y = |\omega|^2 \cdot \frac{3e^\epsilon+\frac{1}{3e^\epsilon}-\frac{2}{3}}{(e^\epsilon-1)^2} - |\omega|^2 \cdot \frac{4(e^\epsilon+\frac{1}{3})}{(e^\epsilon-1)^2} = |\omega|^2 \cdot \frac{(-1)\cdot(e^\epsilon-\frac{1}{3e^\epsilon}+2)}{(e^\epsilon-1)^2}$ determine the magnitude based on the positive or negative difference. Let $e^\epsilon = T$, $y = |\omega|^2 \cdot \frac{(-1)\cdot(T-\frac{1}{3T}+2)}{(T-1)^2}$, and define $f(T) = T - \frac{1}{3T} + 2$, $f'(T) = 1 + \frac{1}{3T^2}$, where $T \in (1, +\infty)$, it follows that $f'(T)$ is always greater than 0, and $f(T)$ is a monotonically increasing function. Also, $f(1) = 3 - \frac{1}{3} > 0$, indicating that $f(T)$ is always positive in this interval. Since the denominator $(T-1)^2 > 0$ and $|\omega|^2 \geq 0$, it follows that for $\epsilon \in (0, +\infty)$, $y = |\omega|^2 \cdot \frac{(-1)\cdot(e^\epsilon-\frac{1}{3e^\epsilon}+2)}{(e^\epsilon-1)^2} < 0$, proof is complete. □

Theorem A2. *For any model weight input values $t, t' \in \{1, -1\}$ and perturbation coefficient output value $t^* \in \left[-\frac{e^\epsilon+1}{e^\epsilon-1}, -1\right] \cup \left[1, \frac{e^\epsilon+1}{e^\epsilon-1}\right]$, the SPM mechanism satisfies $\frac{pdf(t^*|t)}{pdf(t^*|t')} \leq \frac{p}{\frac{p}{e^\epsilon}} = e^\epsilon$, thereby ensuring ϵ-local differential privacy. Additionally, a zero bias is introduced in the mean estimation of the weights to ensure that $E[\bar{S}(\omega)] = \bar{\omega}$, which means that the expected value of the mean parameters of the aggregated perturbation model equals the mean parameters of the original aggregated model.*

Proof. Given that the mechanism is S, ω represents weights, and $E[S(\omega)]$ denotes the expected weights after perturbation.

$$E[S(\omega)] = E[|\omega| \cdot S(t)] = |\omega| \cdot E[t^*] = |\omega| \cdot t = \omega \tag{A19}$$

Therefore, for the weights of the clients used:

$$E\left[\overline{S(\omega)}\right] = E\left[\frac{1}{n}\sum_{i=1}^{n} S(\omega_i)\right] = \frac{1}{n}\sum_{i=1}^{n}\omega_i = \overline{\omega} \tag{A20}$$

$\mathrm{Var}[t^*]$ represents the variance of perturbed values, hence the variance of weights after perturbation is:

$$\mathrm{Var}[S(\omega)] = \mathrm{Var}[|\omega| \cdot t^*] = |\omega|^2 \cdot \mathrm{Var}[t^*] = |\omega|^2 \cdot \frac{3e^\epsilon + \frac{1}{3e^\epsilon} - \frac{2}{3}}{(e^\epsilon - 1)^2} \tag{A21}$$

□

Theorem A3. *For* $\forall \omega \in W$*, there exists* $\lambda = O\left(\frac{|\omega|}{\epsilon} \cdot \sqrt{\frac{\ln(1/\beta)}{n}}\right)$ *such that* $|\overline{S(\omega)} - \overline{\omega}| < \lambda$ *with at least* $1-\beta$ *probability.*

Proof. For any client i, the weights after perturbation have an upper bound b within a certain range, where the upper bound b of this perturbation range is:

$$b = |S(\omega_i) - \omega_i| \le |S(\omega_i)| + |\omega_i| \le |\omega_i| \cdot |C| + |\omega_i| = |\omega| \cdot \frac{2e^\epsilon}{e^\epsilon - 1} \tag{A22}$$

For the variance of the weight after perturbation, we can conclude:

$$\mathrm{Var}[S(\omega)] = \mathrm{Var}[S(\omega) - \omega] = E[(S(\omega) - \omega)^2] - E[S(\omega) - \omega]^2 \tag{A23}$$

Therefore, $E[S(\omega) - \omega] = E[S(\omega)] - \omega = \omega - \omega = 0$ used:

$$\mathrm{Var}[S(\omega)] = E[(S(\omega) - \omega)^2] \tag{A24}$$

According to the Bernstein inequality, substituting the upper bound b and $\mathrm{Var}[S(\omega)]$, we get:

$$\begin{aligned}
\Pr\left(\left|\sum_{i=1}^{n} X_i\right| \ge \lambda\right) &= \Pr\left(\left|\overline{S(\omega)} - \overline{\omega}\right| \ge \lambda\right) = \Pr\left(\left|\frac{\sum_{i=1}^{n}(S(\omega_i) - \omega_i)}{n}\right| \ge \lambda\right) = \Pr\left(\left|\sum_{i=1}^{n}(S(\omega_i) - \omega_i)\right| \ge n\lambda\right) \\
&\le 2\exp\left(-\frac{(n\lambda)^2}{2(\sum_{i=1}^{n} E[(S(\omega) - \omega)^2] + bn\lambda/3)}\right) \le 2\exp\left(-\frac{(n\lambda)^2}{2(\sum_{i=1}^{n} \mathrm{Var}(X_i) + bn\lambda/3)}\right) \\
&= 2\exp\left(-\frac{n\lambda^2}{|\omega|^2\left(\frac{6e^\epsilon + \frac{2}{3e^\epsilon} - \frac{4}{3}}{n(e^\epsilon - 1)^2} + \frac{4|\omega|\lambda e^\epsilon}{3(e^\epsilon - 1)}\right)}\right) = 2\exp\left(-\frac{n\lambda^2}{|\omega|^2 \cdot O(\epsilon^{-2}) + \lambda|\omega| \cdot O(\epsilon^{-1})}\right)
\end{aligned} \tag{A25}$$

Based on the aforementioned joint agreement, there exists $\lambda = O\left(\frac{|\omega|}{\epsilon} \cdot \sqrt{\frac{\ln(1/\beta)}{n}}\right)$ such that $\left|\overline{S(\omega)} - \overline{\omega}\right| < \lambda$ holds with at least a probability of $1-\beta$. □

References

1. 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 20th International Conference on Artificial Intelligence and Statistics (AISTATS), PMLR, Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282. Available online: https://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com (accessed on 15 September 2024).
2. Yang, Q. AI and data privacy protection: The way to federated learning. *J. Inf. Secur. Res.* **2019**, *5*, 961–965.

3. Warnat-Herresthal, S.; Schultze, H.; Shastry, K.L.; Manamohan, S.; Mukherjee, S.; Garg, V.; Sarveswara, R.; Händler, K.; Pickkers, P.; Aziz, N.A.; et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature* **2021**, *594*, 265–270. [CrossRef]
4. Ma, J.; Naas, S.A.; Sigg, S.; Lyu, X. Privacy-preserving federated learning based on multi-key homomorphic encryption. *Int. J. Intell. Syst.* **2022**, *37*, 5880–5901. [CrossRef]
5. Park, J.; Lim, H. Privacy-preserving federated learning using homomorphic encryption. *Appl. Sci.* **2022**, *12*, 734. [CrossRef]
6. Ressi, D.; Romanello, R.; Piazza, C.; Rossi, S. AI-enhanced blockchain technology: A review of advancements and opportunities. *J. Netw. Comput. Appl.* **2024**, *225*, 103858. [CrossRef]
7. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *Acm Comput. Surv.* **2023**, *55*, 1–31. [CrossRef]
8. Atluri, V.; Pietro, R.D.; Jensen, C.D.; Meng, W. Computer Security–ESORICS 2022. In Proceedings of the 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, 26–30 September 2022; Proceedings, Part III. [CrossRef]
9. Pajooh, H.H.; Demidenko, S.; Aslam, S.; Harris, M. Blockchain and 6G-enabled IoT. *Inventions* **2022**, *7*, 109. [CrossRef]
10. Hosseini, S.M.; Sikaroudi, M.; Babaei, M.; Tizhoosh, H.R. Cluster based secure multi-party computation in federated learning for histopathology images. In *International Workshop on Distributed, Collaborative, and Federated Learning*; Springer Nature: Cham, Switzerland, 2022; pp. 110–118. [CrossRef]
11. Kanagavelu, R.; Wei, Q.; Li, Z.; Zhang, H.; Samsudin, J.; Yang, Y.; Goh, R.S.M.; Wang, S. CE-Fed: Communication efficient multi-party computation enabled federated learning. *Array* **2022**, *15*, 100207. [CrossRef]
12. El Ouadrhiri, A.; Abdelhadi, A. Differential privacy for deep and federated learning: A survey. *IEEE Access* **2022**, *10*, 22359–22380. [CrossRef]
13. Baek, C.; Kim, S.; Nam, D.; Park, J. Enhancing differential privacy for federated learning at scale. *IEEE Access* **2021**, *9*, 148090–148103. [CrossRef]
14. Alasmary, H.; Tanveer, M. ESCI-AKA: Enabling secure communication in an IoT-enabled smart home environment using authenticated key agreement framework. *Mathematics* **2023**, *11*, 3450. [CrossRef]
15. Gupta, S.; Alharbi, F.; Alshahrani, R.; Kumar Arya, P.; Vyas, S.; Elkamchouchi, D.H.; Soufiene, B.O. Secure and lightweight authentication protocol for privacy preserving communications in smart city applications. *Sustainability* **2023**, *15*, 5346. [CrossRef]
16. Kanellopoulos, D.; Sharma, V.K. Dynamic load balancing techniques in the IoT: A review. *Symmetry* **2022**, *14*, 2554. [CrossRef]
17. Wang, S.; Kang, B.; Ma, J.; Zeng, X.; Xiao, M.; Guo, J.; Cai, M.; Yang, J.; Li, Y.; Meng, X.; et al. A deep learning algorithm using CT images to screen for Corona Virus Disease (COVID-19). *Eur. Radiol.* **2021**, 31, 6096–6104. [CrossRef]
18. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M.K.; Ristenpart, T. Stealing machine learning models via prediction APIs. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 601–618. Available online: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer (accessed on 15 September 2024).
19. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019. Available online: https://proceedings.neurips.cc/paper/2019/hash/60a6c4002cc7b29142def8871531281a-Abstract.html (accessed on 15 September 2024).
20. Yang, Z.; Zhang, J.; Chang, E.C.; Liang, Z. Neural network inversion in adversarial setting via background knowledge alignment. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 225–240. [CrossRef]
21. Zhao, J.; Yang, M.; Zhang, R.; Song, W.; Zheng, J.; Feng, J.; Matwin, S. Privacy-enhanced federated learning: A restrictively self-sampled and data-perturbed local differential privacy method. *Electronics* **2022**, *11*, 4007. [CrossRef]
22. Sun, L.; Lyu, L. Federated model distillation with noise-free differential privacy. *arXiv* **2020**, arXiv:2009.05537. [CrossRef]
23. Wu, X.; Zhang, Y.; Shi, M.; Li, P.; Li, R.; Xiong, N.N. An adaptive federated learning scheme with differential privacy preserving. *Future Gener. Comput. Syst.* **2022**, *127*, 362–372. [CrossRef]
24. Liu, W.; Cheng, J.; Wang, X.; Lu, X.; Yin, J. Hybrid differential privacy based federated learning for Internet of Things. *J. Syst. Archit.* **2022**, *124*, 102418. [CrossRef]
25. Shen, X.; Liu, Y.; Zhang, Z. Performance-enhanced federated learning with differential privacy for internet of things. *IEEE Internet Things J.* **2022**, *9*, 24079–24094. [CrossRef]
26. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local privacy, data processing inequalities, and statistical minimax rates. *arXiv* **2013**, arXiv:1302.3203. https://arxiv.org/abs/1302.3203.
27. Wang, N.; Xiao, X.; Yang, Y.; Zhao, J.; Hui, S.C.; Shin, H.; Shin, J.; Yu, G. Collecting and analyzing multidimensional data with local differential privacy. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; IEEE: Piscatway, NJ, USA; pp. 638–649. [CrossRef]
28. Sun, L.; Qian, J.; Chen, X. LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–26 September 2021. [CrossRef]
29. Ren, Y.; Liu, R.; Wang, D.; Yuan, L.; Shen, Y.; Wu, G.; Wang, Q.; Yang, C. Research on a federated learning-based local differential privacy mechanism. *J. Electron. Inf. Technol.* **2023**, *45*, 784–792. [CrossRef]

30. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *Acm Trans. Intell. Syst. Technol. (Tist)* **2019**, *10*, 1–19. [CrossRef]
31. Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* **2014**, *9*, 211–407. [CrossRef]
32. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership inference attacks against machine learning models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–24 May 2017; IEEE: Piscatway, NJ, USA, 2017; pp. 3–18. [CrossRef]
33. Yang, C.; Zhu, M.; Liu, Y.; Yuan, Y. FedPD: Federated Open Set Recognition with Parameter Disentanglement. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 4–6 October 2023; IEEE: Piscatway, NJ, USA, 2023; pp. 4882–4891. Available online: http://openaccess.thecvf.com/content/ICCV2023/html/Yang_FedPD_Federated_Open_Set_Recognition_with_Parameter_Disentanglement_ICCV_2023_paper.html (accessed on 15 September 2024).
34. Chen, Z.; Yang, C.; Zhu, M.; Yuan, Y. Personalized retrogress-resilient federated learning toward imbalanced medical data. *IEEE Trans. Med. Imaging* **2022**, *41*, 3663–3674. [CrossRef]

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.