FedBAR: Block Level Data Augmentation Privacy Protection method of Federated Learning

Abstract. Federal Learning (FL) is a privacy-centric distributed machine learning framework, which mitigates privacy risks by sharing model updates rather than data. However, recent research indicates that sharing model updates cannot exempt FL from the threat of inference attacks. This study delves into the reasons for data leakage in FL under heterogeneous environments, revealing that attack algorithms rely on prior knowledge and auxiliary information drawn from gradients for attacks. Based on these findings, we propose a block-level data augmentation privacy protection method for FL, addressing the reliance of attack algorithms on prior knowledge and auxiliary information. By undermining the validity of prior knowledge and preventing attack algorithms from utilizing auxiliary information to reconstruct private samples, the privacy risk can be reduced while maintaining the performance of FL. This is achieved by applying data augmentation techniques that incorporate privacy protection capabilities to the data representation. This paper conducts reconstruction attack experiments and, without compromising accuracy, significantly decreases the correlation between the data representations restored by the attack algorithms and the true data representations. This enhances the privacy protection capability of FL.

Keywords: Federated Learning, Privacy Protection, Data Augmentation.

1 Introduction

Federal Learning (FL), a distributed machine learning framework initiated by Google [1], substantially distinguishes itself from traditional distributed machine learning through retains local data during training tasks. This approach enables a cooperative sharing of local model parameters or grading without having to reveal underlying local data across various devices.

Recent studies have indicated that the gradient of client models uploaded in FL can also be subject to malicious attacks, such as attribute inference attacks [2] and model inversion attacks, including DLG [3], IDLG [4], and GGL [5]. These threats arise due to the inclusion of information from training data within these gradients. Specifically, attribute inference attacks can deduce sensitive data properties such as labels and sample volumes from uploaded model updates; concurrently, model inversion attacks reconstruct the training samples by using the uploaded model gradients. Both types of attack represent a significant risk to the privacy of data in FL.

In recent years, a multitude of privacy-preserving techniques have emerged to mitigate the privacy leakage caused by these attacks, with the main strategies encompassing three types: differential privacy [6], secure multi-party computation [7], and homomorphic encryption [8]. Each of these methods frequently come with drawbacks. These include reduced model accuracy or substantial computational overhead. Further complicating the matter, the data from client devices in FL is often non-IID (independent and identically distributed) [9]; such a heterogeneous environment exacerbates these deficiencies.

Furthermore, there exist several privacy-preserving studies that do not align directly within the above three categories, amongst which Instahide [10] is a noteworthy exemplar. This approach introduces data augmentation techniques in FL, thereby generating new samples for model training to curtail the leakage of personal sample information. Nevertheless, certain research [11] proposes that data representation leakage is actually the fundamental cause of attacks. Compounded by the impacts of heterogeneous data environments within FL, the entanglement amongst data representations reduces, hence making gradients more susceptible to sample information leakage. As a result, this paper advocates for using data augmentation techniques on data representations rather than the samples themselves to safeguard privacy, followed by implementing clipping to counteract reconstruction attacks.

To our knowledge, this paper presents the first privacy-preserving scheme that enhances data representations using data augmentation, thereby protecting privacy during the training process without compromising model accuracy. Specifically, the algorithmic contributions of this paper are as follows: 1) The use of heterogeneous labeled representations for data augmentation, which enables the reduction of attackers' exploitation of supplemental information in gradients, while simultaneously enhancing model accuracy. 2) Implementing gradient clipping post-data augmentation, which alleviates the exploding gradients caused by data heterogeneity and data augmentation, effectively defending against reconstruction attacks. In order to appraise the privacy protection efficiency of our algorithm, this study conducts comparative experiments against several well-known attacks, including DLG, IDLG, IG [12], and GGL. The results demonstrate that our method can significantly reduce data representation leaks while effectively enhancing the accuracy of the FL model, thus protecting the privacy of client data within FL.

The subsequent structure of this manuscript is delineated as follows. Chapter 2 presents an overview of the current research landscape in the realm of privacy protection approaches and attack methods within the framework of FL. Chapter 3 delves into the details of data augmentation and the causative factors leading to privacy leakage. In Chapter 4, the proposed algorithm, along with the advantages it holds in the area of privacy protection, is articulated. In Chapter 5, a security analysis of the implemented method in this paper is undertaken. Chapter 6 encompasses the experimental results and comparative analysis of the devised method regarding model accuracy and privacy protection competency. Finally, Chapter 7 offers a comprehensive summary of the manuscript.

2 Related Work

2.1 Privacy Preservation in FL

In the current landscape of FL, there are three principal methods for preserving privacy, namely: DP, HE, and MPC. Both HE and MPC possess specific requirements. For instance, HE is incapable of supporting complex computations, while MPC necessitates extensive communication to ensure trustworthiness. Additionally, implementing these methodologies incurs substantial costs, requiring the client-side to support a significant volume of local encryption computations and consequently amplifying the communication overhead markedly. Therefore, this paper primarily elaborates on the progression of non-encryption technologies.

DP seeks to protect data privacy by adding noise to the data or model parameters. Miao et al. [13] put forth a FL solution for compressing and protecting privacy within a Deep Neural Network (DNN) architecture based on Compressed Sensing and Adaptive Local DP (termed as CAFL). This method by means of a reconstruction algorithm, the compressed and privacy-protected local model is reconstructed to obtain a global model. In the shuffle model setting, Liu et al. [14] initially presented a FL framework as well as an extended simple protocol (SS-Simple) and a supplementary protocol (SS-Double), utilizing sub-sampling to amplify privacy. Their method significantly enhanced model performance in an equivalent privacy protection environment. Within a contrasting modality, Andrew et al. [15] proposed a DP approach with an adaptive clipping threshold. This strategy relentlessly adjusts the clipping threshold by finding suitable percentiles.

In the field of privacy protection, apart from the mainstream techniques, other researches have been geared towards protection against specific characteristics of reconstruction attacks. The scheme Soteria implements defensive measures against inference attacks by identifying and pruning gradients that are more prone to leak data privacy [11]. It achieves this through an analysis of the sources of gradient leaks. However, the attacker can infer the pruning masks from the gradient information and use these masks as supplementary information to facilitate attacks. On the other hand, InstaHide [10] proposed an encryption technique for training image data based on data augmentation. Random sign changes are also applied to pixels during the encryption process. Nonetheless, attackers can still infer label information using gradients and utilize this information as assistance to conduct attacks. These studies suggest that while current privacy protection measures can mitigate the hazard of reconstruction attacks, these defenses are not foolproof. Attackers can still leverage auxiliary information to bypass these defenses. Hence, continued focus on and research in this area are crucial to fortify defenses against diverse classes of attacks.

In summary, DP often necessitate a trade-off in model performance and pay insufficient attention to the employment of prior knowledge and auxiliary information in privacy leakage attacks. Although the impact of both the Soteria and InstaHide methods on model performance is negligible, they fail to counter attackers who employ gradient inferencing to leverage auxiliary information in heterogeneous environments.

2.2 Privacy Leakage via Gradient in FL

Research into privacy leakage attacks in FL is extensive. This article primarily discusses inferring private data through gradient information. Yang et al. [16] divide existing gradient leakage attacks into two categories: optimization-based methods and analytic-based methods. Optimization-based methods typically resort to Generative Adversarial Networks (GAN) [17] for synthesizing false data, followed by gradient computations. The process continually refines the discrepancy between authentic and artificial gradients, which ultimately contributes to the reconstruction of the genuine data. L Zhu et al. [3] proposed an algorithm called DLG, which seeks the global optimum by solving a non-convex optimization problem, minimizing the distance between the fictitious gradient and the real gradient. Building on DLG, some investigators introduced iDLG [4], aiming to refine data labels via gradients, consequently enhancing the precision and efficiency of DLG. Aside from utilizing auxiliary information, Geiping et al. [12] also designed a gradient inversion attack, demonstrating that the reconstruction of images can be significantly improved based on prior knowledge of the model gradient. It was proven that this inference attack might still succeed, even with trained deep networks.

Analytic methods typically derive the original private data by solving a set of linear equations, consequently facilitating quicker data reconstruction. Phong et al [18] pioneered an analytic method, demonstrating how to compute original inputs from gradients. To address complex deep networks in real-world scenarios, Zhu et al [19] proposed R-gap, an approach that utilizes rank analysis to estimate the feasibility of performing gradient leakage attacks under a given network architecture. Fawl et al [20] introduced an environment of active attacks where the server maliciously modulates a fully connected layer to conveniently infer information from client samples.

Optimization-based methods necessitate only gradient information to conduct privacy leakage attacks, drastically impeding the detection of such attacks in a federated learning setting. Further, with the essentially fewer environmental constraints, such as bias terms, in comparison to analytic-based methods, optimization-based methods are applicable in the majority of federated learning scenarios. Consequently, this study zeroes in on defensive measures against optimization-based approaches. Frequently, these methods resort to finding the optimum solutions of non-convex problems to recover the private data, and tend to infer auxiliary details such as labels or related data distributions from the gradients. Exploiting prior knowledge and these auxiliary insights, privacy is threatened. Therefore, the crux of our method lies in sabotaging the efficacy of prior knowledge, therein thwarting the reconstruction of private samples by the attack algorithms using auxiliary data.

3 Basic Technology

This chapter addresses the issue of data representation leakage in FL, which is primarily attributed to insufficient entanglement of data representations. Concurrently, we will introduce the Block-level Data Augmentation method employed in this study, along with discussing its role in privacy protection.

3.1 Representation Leakage in FL

Sun et al. [11] pinpointed data representation leakage as the fundamental cause of privacy breaches in FL. For sake of simplicity, the concept can be illustrated using a fully connected layer as an instance to analyze how data representation leakage occurs in FL. We denote the fully connected layer as $f(x) = W\,x$, where x is the input to the fully connected layer (i.e., the output data representation from the preceding layer), W

denotes the model weight matrix, and f symbolizes the output of this layer. Consequently, given a batch of training samples B, the gradient of the model W can be explicated as:

$$\frac{\frac{1}{|B|}\sum_{i=1}^{|B|}\partial l^i}{\partial W} = \frac{1}{|B|}\sum_{i=1}^{|B|}\frac{\partial l^i}{\partial f^i}\frac{\partial f}{\partial W} = \frac{1}{|B|}\sum_{i=1}^{|B|}\frac{\partial l^i}{\partial f^i}(x^i)^T \tag{1}$$

where li, xi, and fi represent, respectively, the loss, input and output of the i^{th} sample within the given batch. As it can be observed from the mathematical structure, the gradient of a specified sample i in this layer, $\frac{\partial l^i}{\partial W}$, is indeed the product of the column vector $\frac{\partial l^i}{\partial f^i}$ and the transpose of the row vector $(x^i)^T$. If we denote the training data label set as Y, the batch sample set B can be consequently disassembled into $B = \{B_0, B_1, \ldots\ldots, B_C\}$, where B_k indicates the data samples labeled with k. Put simply, the original equation, termed as Equ. (1) in our analysis, can be rewritten as:

$$\frac{\frac{1}{|B|}\sum_{k=1}^{|B|}\partial l^k}{\partial W} = \sum_{i=1}^{C}\left(\frac{1}{|B_i|}\sum_{j\in B_i}\frac{\partial l^i}{\partial f^i}(x^i)^T\right) \triangleq \sum_{i=1}^{C}\boldsymbol{G}(B_i) \tag{2}$$

where $G(B_i)$ denotes the gradient of sample B_i in a given layer. When batch B manifests a rich label set Y, the data representation of the batch becomes intermingled, which diminishes the sample data leakage derived from gradients. In extreme cases, when the label set in a batch is significantly deficient, malicious actors can almost accurately infer the label set utilized in that batch. In the context of FL, which is frequently characterized by heterogenous environments, the limited client sample sets are unavoidable. Consequently, this situation results in clients in FL displaying very low degrees of data intertwining, making the privacy of client samples in these FL environments more susceptible to breaches.

3.2 Block level data augmentation privacy protection in data representation space

This section introduces a block-level data augmentation method in the data representation space [20]. Initially, let $g(x)$be the method obtains data representation of a sample x. Then, a binary mask, M, is generated by randomly selecting data representation blocks. The masked parts of the data representation are mixed using a vector data augmentation. The Soft PatchUp operation is defined as follows:

$$Mix_\lambda(a,b) = \lambda \cdot a + (1-\lambda)\cdot b \tag{3}$$

where a and b denote the two vectors to be mixed, and λ falls within the range [0, 1] representing the mixing coefficient, the operation of block-level data augmentation at the k-th layer can be described as follows:

$$\emptyset\left(g_k(x_i), g_k(x_j)\right) = M * g_k(x_i) + Mix_\lambda \left[\left((1-M)*g_k(x_i)\right),\left((1-M)*g_k(x_j)\right)\right] \tag{4}$$

where x_i and x_j respectively denote two different samples, and λ is a random value drawn from the $\lambda \sim Beta(\alpha, \alpha)$ distribution, where λ lies in the interval [0, 1]. The hyperparameter α modulates the shape of the Beta distribution, also concurrently determining the relative proportion represented by the two samples in the masked data segments.

Correspondingly, the loss function following data augmentation can be defined as follows:

$$\begin{aligned} L(f) = {} & \mathbb{E}_{(x_i,y_i)\sim P}\mathbb{E}_{(x_j,y_j)\sim P}\mathbb{E}_{\lambda\sim Beta(\alpha,\alpha)}\mathbb{E}_{k\sim S} \\ & Mix_{P_u}[l(f_k(\emptyset_k), y_i), l(f_k(\emptyset_k), Y)] \\ & +l\big(f_k(\emptyset_k), Mix_{P_u}(y_i, Mix_\lambda(y_i, y_j))\big) \end{aligned} \tag{5}$$

where P_u represents the proportion of x_i's data representation that does not participate in the data augmentation process, whilst S denotes the set of layers undergoing data enhancement.

4 Method

This chapter initially delineates the overall process and algorithms presented in this study, followed by a detailed elucidation of the block-level data augmentation techniques and cropping strategies employed.

4.1 Block level data augmentation privacy protection method of FL

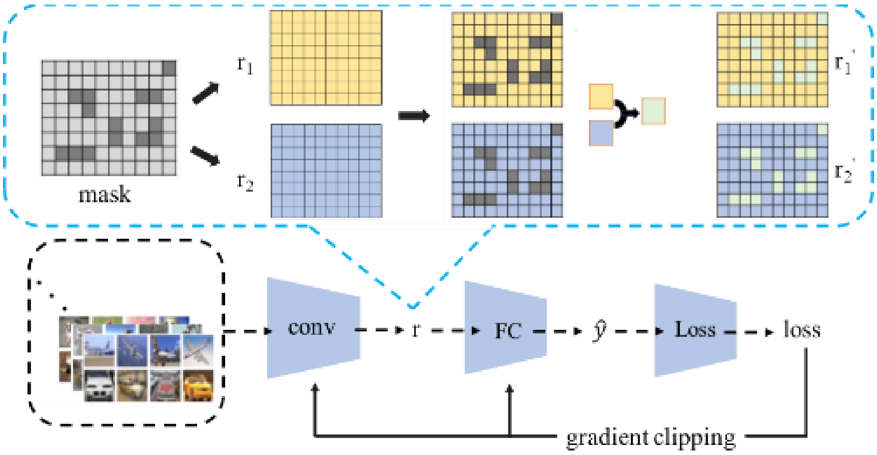


Fig. 1. FedBAR framework, where r₁ and r₂ represent data representations of two different samples.

The local training process of this article is shown in Figure 1. As elucidated in the preceding sections, the primary cause of privacy leakage in FL primarily stems from minimal entanglement between data representations, invoked by data heterogeneity. Hence, this study proposes the enhancement of data entanglement through data augmentation as a viable strategy to fortify against privacy leakage.

The specific objectives are outlined below:

- Through the application of data augmentation techniques, the entanglement of data representation is enhanced, which mitigates the threat auxiliary information poses to privacy, thereby promoting privacy protection.
- The use of gradient clipping techniques maintains the efficacy of FL, serving as a defense against the potential impact of data augmentation on model aggregation arising from disparate label data representations.

Algorithm 1. Federated learning with **B**lock level Data **A**ugmentation privacy protection algorithm for Data **R**epresentation.(FedBAR)

```
Input: learning rate η, total round t, clipping gamma γ, clipping learning rate $\eta^g$.
1: function Server(η, t, γ, $\eta^g$)
2:     Initialize model $W^0$
3:     for each round t = 1, 2, …, do
4:     // Server Update
5:         $C_t$ sample all users uniformly
6:         for each client c∈$C_t$ do
7:             $W_t^c$← ClientUpdate(c, $W^t$, η, $\eta^g$,γ)
8:         end for
9:         $W_{t+1} \leftarrow \sum_{c\in C} \frac{n_k}{n} W_{t+1}^c$
10:    end for
11: end function
12: function ClientUpdate(c, $W^t$, η , $\eta^g$, γ)
13:    $B$ ← user c's local data split into batches
14:    // Client Update
15:    for batch b∈B do
16:        train W with data augmentation using Algorithm 2
17:    end for
18:    $\Delta \leftarrow W - W_t$
19:    $g^t \leftarrow$ γ-th percentile of $\Delta$
20:    p $\leftarrow I_{\|\Delta\|<g^t}$ //
21:    $\Delta' \leftarrow \Delta \cdot \min(1, \frac{g^t}{\|\Delta\|})$
22:    $g^{t+1} \leftarrow g^t \cdot exp(-\eta^g(p-\gamma))$
23:    return $\Delta'$
24: end function
```

Algorithm 1 illustrates the process of the proposed FL algorithm. Upon commencement of FL, the server initializes a model (Line 2), then samples a client and sends the model and hyperparameters to the client. Following the receipt of the model, the client proceeds with training and applies a data augmentation algorithm on data representation during this process (Line 16). After training all samples, the client calculates the model update (Line 18). Subsequently, it uses these updates to compute the adaptive clipping threshold (Lines 19-22) and performs the clipping (Line 21). Finally, the client sends the clipped updates back to the server, which, after receiving the updates, aggregates the model using the number of samples as the weights (Line 9), and disseminates the new aggregated model for the next round of training.

4.2 Advantages of block level data representation enhancement in privacy protection

In comparison to InstaHide, this study employs data augmentation techniques to protect the privacy of client training data analogous to InstaHide. However, these two approaches differ in terms of the subject of data augmentation. InstaHide enhances the image samples themselves. Because the contributions from the same regions across different image samples towards prediction or classification often vary, the augmented samples might lose crucial information from the original samples, leading to a reduction in accuracy. In contrast, this study opts to augment the data representations of the samples. As each dimension value in data representations contributes similarly towards prediction or classification, the augmented data representations still remain applicable for prediction or classification tasks.

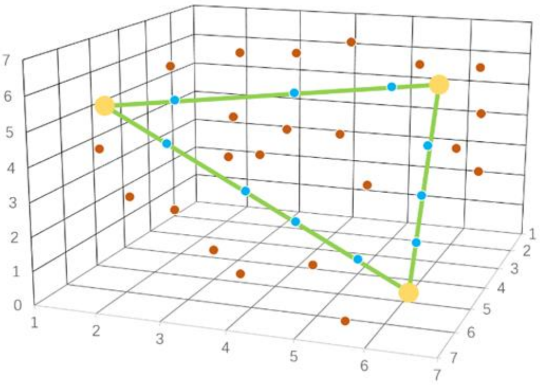


Fig. 2. An example to demonstrate the effectiveness of data augmentation in data representation space. In this figure, the yellow dots serve as the representation of real data, while the results obtained from the Manifold algorithm are denoted by blue dots. Brown dots illustrate the result generated from the algorithm presented within this study.

Compared to Manifold [21], block-level data representation augmentation and Manifold both bolster model accuracy by enhancing data representations. Yet, it's worth noting that the former does not augment all data representations but selectively manipulates certain positioned data representations. From a privacy protection perspective, Manifold essentially implements a weighted average on two vectors, preserving all the data representation information from the image and maintaining a linear relationship.

As illustrated in Figure 2, given two distinct data representation distributions (yellow dots), when we perform data augmentation on them in entirety, the resulting new data representation is a linear combination (blue dots). Frequently, models employ one-hot encoding to train the data representation distribution of different labels, making it challenging to accurately judge to which category an augmented data representation belongs. This in turn might lead to inference attack failures when attackers use ineffective auxiliary information inferred from the gradients.

However, when attackers already know the augmented data representation, it is conceivably possible for them to infer the data augmentation parameters by enumeration (green line). Moreover, as previously mentioned, not all augmented data representations can provide an effective privacy protection. Hence, block-level data augmentation ensures that the mixed data representations (brown dots) break away from the linear relationship with the two that underwent augmentation; this feature significantly complicates the deduction of actual data representations even if attackers know the generated data representations.

4.3 Adaptive gradient clipping algorithm

Owing to the enhancement of block-level data representation applied exclusively to the fully-connected layers in this work, the entanglement among the data representations increases, leading to smaller gradient variance in these layers compared to the remaining layers. Therefore, the efficacy of a fixed clipping threshold tends to be unsatisfactory. Consequently, we employed the adaptive gradient clipping algorithm proposed by Andrew et al. [15]. However, we diverged from the original approach by eschewing the aggregation of clipping threshold information by a server. Instead, we implemented the adaptive adjustment of the clipping threshold utilizing the client-side local information. This deviation is attributable to the fact typically the servers in the FL schema might be curious or untrustworthy.

5 Security Analysis

Attack scenario. In each epoch, all clients utilize block-level data representation enhancement algorithms to augment the data representation of each sample partaking in the training process. The attacker has knowledge of the client model parameters and the fact that data augmentation has been implemented. The attacker, through an enumerative approach, infers the accurate label and augmentation coefficient. In cases where the clients use a public dataset for data augmentation, the attacker can access the public samples utilized by the client.

5.1 Analytic-based methods

Assuming that the attacker randomly generates and acquires two samples, with the intention to generate gradients through data augmentation algorithms to compare the discrepancy with real gradients, this problem transforms to whether it's possible to reproduce the same result 'r' known by the two data representations r_1 and r_2 used for data augmentation. Considering that the attacker knows the enhancement coefficient, the crux of the problem lies in whether consistent data augmentation blocks can be obtained. Assuming the data representation generated by the FL model is of D dimension, and B blocks of data representation are selected for data augmentation, this problem can be translated into the probability problem of selecting a unique result from C_D^B combinations. Given that the number of B in this algorithm is at least half of D, the task can be simplified into the C_D^{D-B} problem. When $B \neq D$, that is, when there are dimensions that have not participated in data augmentation, there will be C_D^{D-B} possibilities. Taking into account that in typical image networks, the data representation dimension D often exceeds 100, the probability of the attacker restoring data augmentation blocks can be simplified to $1/100^{D-B}$, which makes the recreation of original samples a near impossible event.

5.2 Optimization-based methods [10]

The adversary successfully captures the genuine gradient information ∇W uploaded by the client, along with obtaining the parameter W of the client's trained model. The relationship between parameter W and gradient information ∇W is then utilized to analyze the data representation produced during the user's training process.

Preliminarily, there is no direct correlation between the gradient obtained after data augmentation and the gradients generated by the two data representations that partake in the data augmentation. Supposing there are two data representations r_1 and r_2, undergoing data augmentation. For simplicity, we select the dimensions of all data representations as augmentation sections and set the augmentation coefficient to 0.5, and the augmented data representation $r = (r_1 + r_2)/2$. Given $L(\sigma(f(r)), y)$ as the loss function, where $f(r)$ represents the output of the fully connected layer and $\sigma(\cdot)$ represents the activation function, the gradient of the fully connected layer equals $\frac{\partial L(\sigma(f(r)),y)}{\partial f}$. The unique gradients of r_1 and r_2 in the fully connected layer are then $\frac{\partial L(\sigma(f(r_1)), y_1)}{\partial f}$ and $\frac{\partial L(\sigma(f(r_2)),y_2)}{\partial f}$ respectively. Being that the activation function is a nonlinear function, $\sigma\big(f(r)\big) \neq (\sigma(f(r_1)) + \sigma(f(r_2)))/2$, it is therefore unfeasible to divide the different sample gradients from the augmented data gradient.

Subsequently, acknowledging the presence of data augmentation obfuscates the fact that the derived data representation from gradients does not signify genuine data. To facilitate comprehension, suppose the client undertakes a binary classification predictive task, where the loss function $L(f_d(x_d), y)$ and the fully connected layers $f_d(x_d)$ respectively:

$$L(f_d(x_d), y) = \log\left(1 + e^{-f_d(x_d)}\right) \tag{6}$$

$$f_d(x_d) = yW_d\sigma_{d-1}\left(f_{d-1}(x_{d-1})\right) \tag{7}$$

where d denotes the number of layers in the model, f_d represents the function of the d-th layer, W_d signifies the model parameters of the d-th layer, x_d stands for the input of the d-th layer, $\sigma_d(\cdot)$ corresponds to the activation function of the d-th layer, and y is the real label. Hence, the associated gradient can be defined as follows:

$$\frac{\partial L}{\partial W_d} = y\frac{\partial L}{\partial f_d} f_{d-1}^T \tag{8}$$

$$\frac{\partial L}{\partial W_{d-1}} = \left(\left(W_d^T\left(y\frac{\partial L}{\partial f_d}\right)\right) \odot \sigma'_{d-1}\right) f_{d-2}^T \tag{9}$$

where we would be denoting scalar product as $\odot$, whereas σ'_{d-1} refers to the derivative of σ_{d-1}. Our analysis targets the adversarial reconstruction of full-connectivity layer input x_d, i.e., $\sigma_{d-1}\big(f_{d-1}(x_{d-1})\big)$. Given that the attacker is well-aware of y and W_d, reconstructing $f_d(x_d)$ alone would suffice. It can be obtained by conducting relevant calculations, as outlined below:

$$\frac{\partial L}{\partial f_d} f_d = \frac{-f_d}{1 + e^{f_d}} \tag{11}$$

Finally, exploiting the correlation between $\frac{\partial L}{\partial f_d} f_d$ and f_d under various loss functions, attackers obtain an approximation $\widetilde{f_d}$, and subsequently approximate the full connection layer input $\widetilde{x_d}$. However, data augmentation can weaken the correlation between the data representation x_d and the full connection layer gradient $\frac{\partial L}{\partial f_d} f_d$, thereby causing a reduction in the correlation between $\frac{\partial L}{\partial f_d} f_d$ and f_d.

6 Experiments

6.1 Experimental Setup

Datasets and model. The experiment applies the ResNet-18 model, utilizing the Cifar10 image dataset [23] and the Celeba face dataset, the latter being used for reconstruction attack experiments.

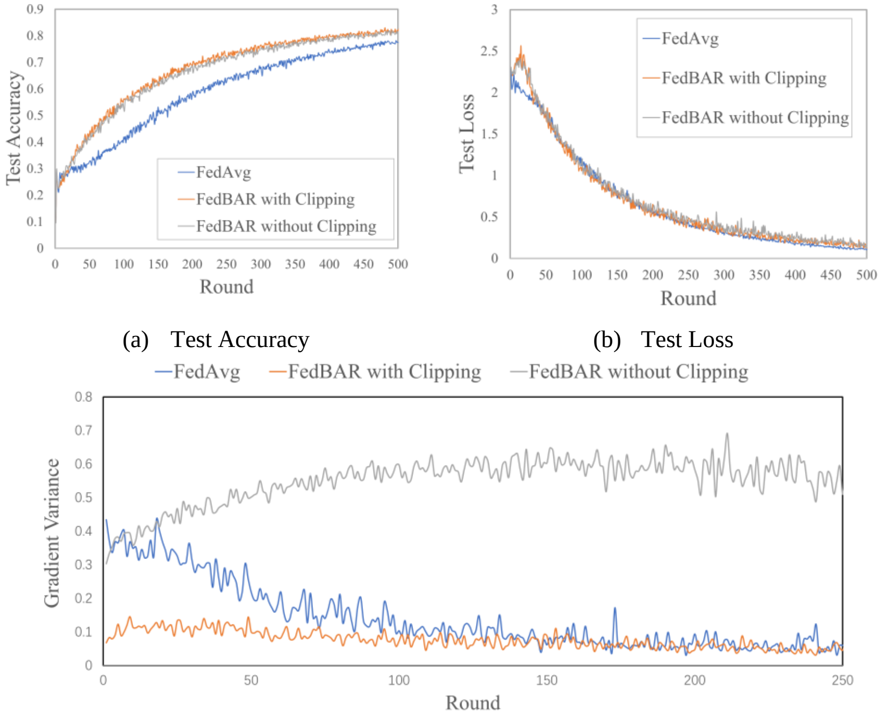
Hyperparameter configurations. In training, included 500 rounds of training with 10 clients, a learning rate of 0.005, a batch size of 32, a single client epoch. The initial percentile for adaptive pruning threshold is set at 0.5, with the learning rate for adaptive pruning is set at 0.2. The number of sample classes and the number of samples held by each client were set randomly, following a Dirichlet distribution, to mimic the heterogeneous data environment found in real-world situations.

Attack models. Four attack models were incorporated in the reconstruction attacks during the experiment: (1) Deep Leakage Generation (DLG), which calculates the L2 loss of the generated sample gradient and the real gradient, optimized by the optimizer; (2) Improved Deep Leakage Generation (iDLG), building on DLG and incorporating the use of prior knowledge about labels; (3) Inverse Gradient (IG), a gradient leakage attack that uses cosine distance for loss, also incorporating the use of total variance prior knowledge of gradients; and (4) Generative Gradient Leakage (GGL), utilizing a GAN trained with a public dataset to generate samples. All attack models used the Adam optimizer [24] to optimize generated samples and the model and algorithm were consistent with the ones used in this paper.

Privacy protection methods. Four privacy protection methods were applied and compared with the method proposed in this paper: (1) Differential Privacy (DP), which preserves privacy by injecting noise into the gradient, with the noise being Gaussian noise $\varepsilon \sim N(\sigma^2 I)$, where $\sigma = 1$; (2) Gradient Sparsity, pruning the gradient to achieve a sparsity of 90%; and (3) Soteria, applying an algorithm based on information extraction to prune gradients with high extraction rates of 80%.

Evaluation metrics. For evaluation standards, model performance was assessed in terms of accuracy and loss. For assessing the effectiveness of privacy protection, quantitative evaluation employing the following metrics was done, aside from visual comparison: (1) Mean Squared Error for images (MSE-I): representing pixel-level MSE between the original and reconstructed samples; (2) Peak Signal-to-Noise Ratio (PSNR): the ratio of the maximum pixel value to the MSE; (3) Learned Perceptual Image Patch Similarity (LPIPS) [25]: a perceptual similarity score between the original and the reconstructed samples, measured by a deep neural network trained on a large-scale dataset, a VGG model [26] was utilized during the experiments; (4) Mean Squared Error of data representations (MSE-R): the MSE between the original and reconstructed data representations. In the forthcoming experiments, the symbol "↓" denotes a phenomenon where lower values of the given metric correspond to weaker privacy protection, while the symbol "↑" indicates that higher values of the said metric result in a diminished privacy protection.

6.2 Comparison and Analysis with FL



(a) Test Accuracy

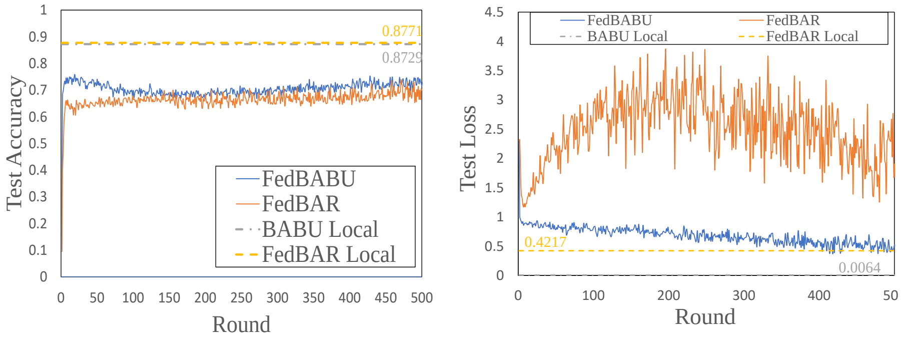
(b) Test Loss

(c) Gradient Variance

Fig. 3. Comparison of model performance between the FedBAR algorithm with and without gradient clipping, in comparison to an algorithm without privacy protection.

As depicted in Figure 3, the FedBAR algorithm (represented by the orange and grey lines) outperforms a non-privacy-preserving algorithm (symbolized by the blue line) in terms of both test accuracy and loss. This advantage arises from the mitigation of detrimental effects isolated to heterogeneous environments through the implementation of data augmentation. Of interest, unclipped FedBAR (shown in grey) shows a marked increase in gradient variance—a stark contrast to the tapering trend exposed by its non-privacy-preserving counterpart (blue line). This discrepancy suggests that data augmentation techniques do indeed influence model convergence in heterogeneous environments. An intriguing distinction is found in the trend of the clipped FedBAR version (orange line), which maintains low variance levels, aligning with the non-privacy-preserving algorithms later on. This indicates that the clipped FedBAR algorithm experiences slower early-stage convergence during training—an assertion empirically validated by fluctuations in test accuracy and loss.

In order to investigate the interaction between heterogeneous environments and data augmentation further, we conducted a comparative experiment based on the FedBABU algorithm [27]. FedBABU is a personalized FL method, in which the central server updates only the body of the model (i.e., non-fully connected layers) during federated training, while the clients individually update the head of the model (i.e., the fully connected layers) locally.



(a) Test Accuracy (b) Test Loss

Fig. 4. Comparison of Model Performance between FedBAR Algorithm and Non-Privacy Protection Algorithm in FedBABU environment. The solid lines denote the global models, and the dashed lines represent the client-customized models after training completion.

The results show that in the FedBABU context, test accuracy over the global model (solid line) of FedBAR is reduced by approximately 0.07 in comparison to algorithms without privacy preservation, accompanied by a significant fluctuation in test loss. However, the local models at the client side (dashed line) maintain virtually identical test accuracies. This suggests that the negative impacts exerted by heterogenous environmental conditions on the model's performance can be mitigated through implementing data augmented on local models at the client's side, thereby maintaining the performance of the model. The pronounced fluctuations in test loss imply a significant influence of the training progress of the fully-connective layer on our method's performance. This phenomenon can be attributed to the fact that data augmentation alters the labels of samples, rendering the untrained fully-connected layer incapable of accurately predicting the labels of augmented samples, consequently leading to a highly volatile loss value.

6.3 Comparison and analysis with other privacy protection methods

From the experimental results in Table 1, it can be observed that the utilization of auxiliary information significantly exacerbates the degree of threat posed by the attack (IDLG compared to DLG). This severity is further intensified when resorting to algorithms exploiting prior knowledge (IG and GGL), thereby leading to an enhanced extent of privacy leakage. These findings highlight the substantial influence that both prior knowledge and auxiliary information have on the intensity of reconstruction attacks.

Table 1. Quantitative comparison between FedBAR and various privacy protection methods under the attack of advanced methods.

Defense	Metrics	Attack			
		DLG	IDLG	IG	GGL
FedBAR	MSE-E ↓	0.6320	0.5069	0.4500	**0.3919**
	PSNR ↑	1.9926	2.9509	3.4677	**4.3426**
	LPIPS ↓	0.7515	0.7744	**0.3699**	0.7495
	MSE-R ↓	0.1180	0.2015	**0.0058**	0.1418
Soteria	MSE-E ↓	0.5930	0.5283	0.3562	**0.2259**
	PSNR ↑	2.2692	2.7716	4.4826	**6.4615**
	LPIPS ↓	0.6506	0.5444	0.5604	**0.3377**
	MSE-R ↓	0.0138	0.1330	**0.0178**	0.0561
Pruning	MSE-E ↓	0.3375	0.3557	0.2195	**0.1217**
	PSNR ↑	4.7177	4.4887	6.7317	**9.1488**
	LPIPS ↓	0.7452	0.6996	0.5344	**0.2828**
	MSE-R ↓	0.4636	0.4214	0.0760	**0.0144**
Noise	MSE-E ↓	0.4042	0.3572	0.2238	**0.1447**
	PSNR ↑	3.8263	4.4546	6.5020	**8.6013**
	LPIPS ↓	0.7410	0.3696	0.4393	**0.3260**
	MSE-R ↓	0.3117	0.0790	0.0291	**0.0010**

The visualization of the experimental results, utilizing the sophisticated reconstruction attack method GGL, is presented in Figure 5. Distinctly, the absence of any privacy preserving mechanism, as in FedAvg, leaves no room for privacy against sophisticated attack algorithms. In contrast, the results from pruning and Soteria defenses against reconstruction attacks reveal a degree of degradation of facial information due to gradient destruction, rendering the faces blurry, yet discernible details such as skin tone and gender persist. Noise performs commendably with a reconstructed result that exhibits minimal visual correlation to the original image, yet aspects like gender and facial outline can still be observed. Notably, the privacy preservation method proposed in this study demonstrates superior performance. The result of the reconstruction attack appears to be a noise map, with nearly indiscernible facial information, rendering it challenging to even confirm as a human face image. This suggests that our FedBAR algorithm protects the information of the original samples without disclosing the information utilized for data augmentation (Mixed).

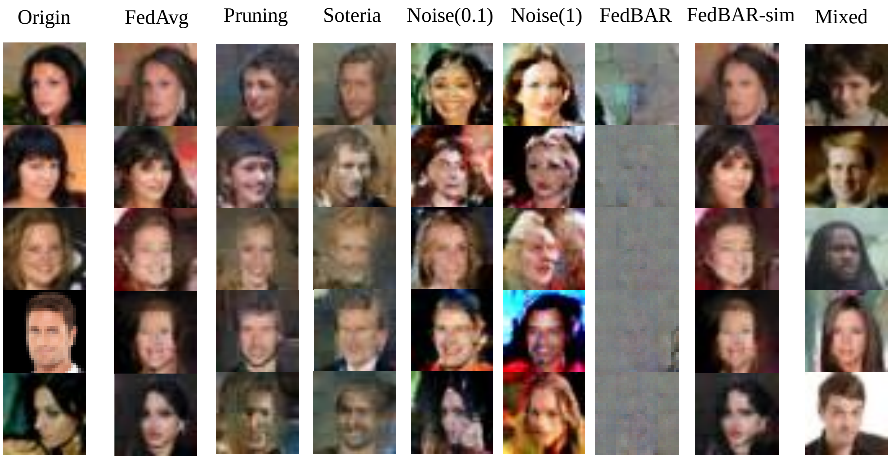


Fig. 5. The results of defense methods under GGL attacks. The "FedBAR" column represents the proposed algorithm implemented with heterogeneously labelled data for data augmentation, while the "FedBAR-sim" indicates the same algorithm applied using homogeneously labelled data for the enhancement process.

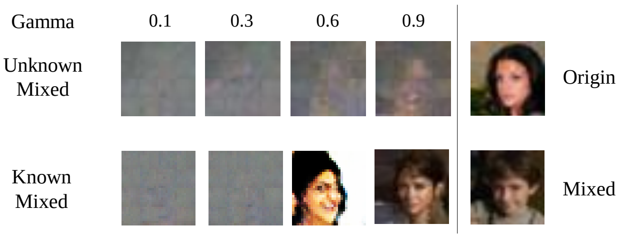


Fig. 6. Privacy protection performance using public datasets at varying ratios of data augmentation blocks. In the figure, 'Gamma' represents the size of the data augmentation block relative to the total data representation dimension. The 'Known Mixed' row denotes the scenario where attackers possess the original mixed samples.

Further to this, in consideration of cases where the client can only access single-category tagged samples, we posit that the client holds single-class private data and enhances data representation through samples from public datasets with differing labels.

Concurrently, the attacker enumerates specific samples from the public data used by the client, meaning the attacker knows the mixed-column samples. As shown in Figure 6, it can observe that as the ratio of the data augmentation block increases, the reconstructed faces in the result become more defined. This can be attributed to the fact that an excessive ratio of data augmentation blocks may pose a risk to the protection of private samples. In scenarios where the adversary is aware of samples from the public dataset, a clear reconstruction of the image commences when the ratio first reaches 0.6. This will inevitably lead to privacy leakage, even though the reconstructed samples exhibit a considerable deviation from private samples. When the attacker possesses sample information used for data augmentation, its correlated measurements are demonstrated in Figure 7. Remarkably, significant fluctuations occur when this ratio reaches 0.6, symbolizing a drastic decrease or increase, corresponding to a rapid decline in the level of privacy protection. This corroborates the view presented in Figure 7, whereby an overly high ratio could result in greater privacy leakage within the samples, regardless of whether the opponent attempts to obtain information about user data augmentation samples.

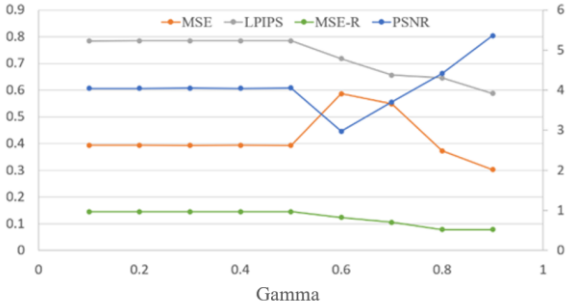


Fig. 7. Proportions of Different Enhancement Blocks and Corresponding Attack Metrics on Public Data Sets, with MSE ↓ , LPIPS ↓ and MSE-R ↓ Mapped to the Left Y-Axis and PSNR ↑ (Blue Line) Mapped to the Right Y-Axis, 'Gamma' represents the size of the data augmentation block relative to the total data representation dimension.

7 Conclusion

In this paper, we introduce a block-level data augmentation privacy protection method suitable for heterogeneous environments of FL. The proposed algorithm utilizes block-level data augmentation for training models within FL and applies gradient clipping. Such a procedure enables enhanced performance of the FL model while ensuring data privacy, divulging an optimal solution for FL under heterogeneous environments. Our experimental results confirm that, for identical image classification tasks, the performance of the proposed algorithm outweighs that of FL models without privacy protection mechanisms. Consequently, while ensuring the accuracy of local models, our algorithm also offers protection for the data privacy of local clients.

References

1. H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In International Conference on Artificial Intelligence and Statistics, 2017.
2. Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In 2019 IEEE Symposium on Security and Privacy (SP). IEEE, 2019.
3. L. Zhu, Z. Liu, and S. Han, Deep leakage from gradients, Advances in Neural Info. Processing Systems, pp. 14747–14756, 2019.
4. B. Zhao, K. R. Mopuri, and H. Bilen, idlg: Improved deep leakage from gradients, accessed on 8 Jan, 2020, available: https://arxiv.org/pdf/2001.02610.pdf.
5. Li Z, Zhang J, Liu L, et al. Auditing privacy defenses in federated learning via generative gradient leakage[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 10132-10142.
6. Dwork C. Differential privacy[C]//International colloquium on automata, languages, and programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006: 1-12.
7. Goldreich O. Secure multi-party computation[J]. Manuscript. Preliminary version, 1998, 78(110).
8. Gentry C. A fully homomorphic encryption scheme[M]. Stanford university, 2009.
9. Zhao Y, Li M, Lai L, et al. Federated learning with non-iid data[J]. arXiv preprint arXiv:1806.00582, 2018.
10. Huang Y, Song Z, Li K, et al. Instahide: Instance-hiding schemes for private distributed learning[C]//International conference on machine learning. PMLR, 2020: 4507-4518.
11. Sun J, Li A, Wang B, et al. Soteria: Provable defense against privacy leakage in federated learning from representation perspective[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 9311-9319.
12. Geiping J, Bauermeister H, Dröge H, et al. Inverting gradients-how easy is it to break privacy in federated learning?[J]. Advances in Neural Information Processing Systems, 2020, 33: 16937-16947.
13. Miao Y, Xie R, Li X, et al. Compressed federated learning based on adaptive local differential privacy[C]//Proceedings of the 38th Annual Computer Security Applications Conference. 2022: 159-170.
14. Liu, R, Cao, Y, Chen, H, Guo, R, Yoshikawa, M(2021). FLAME: Differentially Private Federated Learning in the Shuffle Model. Proceedings of the AAAI Conference on Artificial Intelligence, 35(10), 8688-8696.
15. Andrew G, Thakkar O, McMahan B, et al. Differentially private learning with adaptive clipping[J]. Advances in Neural Information Processing Systems, 2021, 34: 17455-17466.
16. Yang H, Ge M, Xue D, et al. Gradient Leakage Attacks in Federated Learning: Research Frontiers, Taxonomy and Future Directions[J]. IEEE Network, 2023.
17. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.
18. Phong, L. T et al., Privacy-preserving deep learning: Revisited and enhanced, Proc. ATIS, pp. 100-110, 2017.
19. J. Zhu, and M. Blaschko, R-gap: Recursive gradient attack on privacy, accessed on 15 Oct, 2020, available: https://arxiv.org/pdf/2010.07733.pdf.
20. Fowl et al., Robbing the fed: Directly obtaining private data in federated learning with modified models, accessed on 25 Oct, 2021, available: https://arxiv.org/pdf/2110.13057.pdf.

21. Faramarzi M, Amini M, Badrinaaraayanan A, et al. PatchUp: A feature-space block-level regularization technique for convolutional neural networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2022, 36(1): 589-597.
22. Liu Z, Luo P, Wang X, et al. Deep learning face attributes in the wild[C]//Proceedings of the IEEE international conference on computer vision. 2015: 3730-3738.
23. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[J]. 2009.
24. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
25. Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 586-595.
26. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
27. Oh J, Kim S, Yun S Y. Fedbabu: Towards enhanced representation for federated image classification[J]. arXiv preprint arXiv:2106.06042, 2021.