Privacy-preserving Collaborative Learning with Scalable Image Transformation and Autoencoder

Yuting Ma[1], Yuanzhi Yao[1,2,*], Xiaowei Liu[1], and Nenghai Yu[1]
[1]School of Information Science and Technology, University of Science and Technology of China, Hefei 230027, China
[2]School of Computer and Information, Hefei University of Technology, Hefei 230601, China
e-mail: ytma@mail.ustc.edu.cn, yaoyz@ustc.edu.cn, lxw1212@mail.ustc.edu.cn, ynh@ustc.edu.cn

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001061

***Abstract*—Collaborative learning in which local clients jointly train a deep learning model by sharing parameters to the centralized server has gained great popularity. However, recent works have shown that local private data can be leaked to the server by gradient sharing. In this paper, a privacy-preserving collaborative learning scheme is proposed to defend against gradient-based reconstruction attacks. The sensitive training images are firstly permutated by transformation with scalable block sizes. Then, features of permutated images are extracted by a classification-compliant autoencoder for meaningful representation of high-dimensional images and facilitating classification. The model accuracy constraint is incorporated in the training process to maintain decent classification accuracy. Experimental results demonstrate that the proposed scheme can achieve high privacy preservation with minimal impact on model accuracy.**

***Index Terms*—Collaborative learning, privacy preservation, image transformation, autoencoder**

I. Introduction

Privacy concerns over sharing raw data have led to a growing interest in collaborative learning which enables clients' sensitive data to remain on local devices for deep learning [1]. Although clients do not disclose their training data in collaborative learning, they have to send shared parameters (e.g., gradients) to the centralized server. These shared parameters can leak the privacy of sensitive data indirectly. A few recent works [2], [3] have demonstrated the possibility of pixel-wise accurate recovery for training images exploring shared gradients. Yin *et al.* [2] proposed to recover hidden training images via label restoration given batch-averaged gradients. Reconstruction attacks [3] posed serious challenges to privacy preservation in collaborative learning.

In the open literature, typical techniques for defending against reconstruction attacks on training data (e.g., sensitive images) in deep learning include homomorphic encryption [4], differential privacy [5], [6], and image transformation [7], [8], etc. Homomorphic encryption is applied to encrypt shared gradients [4] and shared weights for preventing local data leakage but it requires extensive computation resources. Differential privacy is a natural approach to prevent privacy leakage by adding random noise. In [5], a differentially private stochastic gradient descent (SGD) algorithm is designed. Yuan *et al.* [6] utilized analytic Gaussian mechanism-based differential privacy to avoid the privacy leakage of medical images in collaborative learning. However, differentially private schemes suffer from model accuracy degradation because the intensity of added noise scales proportionally with the model size. Sharma *et al.* [7] presented an efficient way for privacy preservation where the training images are firstly transformed before being sent to the collaborative learning system. Cheung *et al.* [8] proposed to use the random neural network for image transformation. The intuition of image transformation-based privacy-preserving deep learning schemes is that deep learning is powerful enough to extract features even if training data are transformed [9].

The fundamental issue of image transformation-based privacy-preserving deep learning is to design a proper transformation algorithm which perturbs the original image content and maintains efficient feature representation. Perceptual image encryption [10] and transformation policy selection [11] are considered respectively. However, transformed images in [10], [11] still contain sensitive information which leaves traces for reconstruction attacks. To the best of our knowledge, this is the first paper which combines a block-wise scalable image transformation algorithm and a classification-compliant autoencoder to reduce the feature dimension of permutated images and maintain classification accuracy. The autoencoder has twofold functions which are efficient feature representation [12] and reduction in the feature dimension of permutated images respectively.

In summary, we have made the following technical contributions which will be detailed in this paper:

- We design a novel image encryption strategy based on scalable image transformation to defend against privacy leakage from gradients.
- We investigate how to integrate the scalable image transformation and the classification-compliant autoencoder into the privacy-preserving collaborative learning scheme while considering model accuracy constraint.
- We perform extensive experiments to validate the effectiveness of the proposed privacy-preserving collaborative learning scheme in terms of model accuracy and privacy preservation performance.

The remainder of this paper is organized as follows. The problem statement is introduced in Section II. The proposed privacy-preserving collaborative learning scheme is elaborated

*Corresponding author.

in Section III. The experimental results are presented in Section IV. Finally, Section V concludes the paper.

II. Problem Statement

A. Collaborative Learning

In this paper, we consider the generic system model of collaborative learning [13] where there are K clients. Each client $\mathcal{C}_k$ has a local dataset $\mathcal{D}_k = \{z_{k,1}, z_{k,2}, \ldots, z_{k,n_k}\}$ where each sample $z_{k,i} = (x_{k,i}, y_{k,i}), i \in \{1, 2, \ldots, n_k\}$ has a data $x_{k,i}$ and a label $y_{k,i}$. The total number of all clients' samples equals $n = \sum_{k=1}^{K} n_k$. At the beginning of each global communication round, a random fraction of clients is selected, and the server distributes the current global model parameters to each client. Each client conducts the local update based on the global model and the local dataset, and then shares its local model parameters to the server. The server aggregates these shared parameters to update the global model, and the process repeats. The objective of a collaborative optimization problem is to find the optimal model parameter vector $\mathbf{w}^*$ by

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \sum_{k=1}^{K} \frac{n_k}{n} F_k(\mathbf{w}), \quad \text{where } F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{z_{k,i}\in\mathcal{D}_k} f(z_{k,i}; \mathbf{w}). \tag{1}$$

In (1), $F_k(\mathbf{w})$ and $f(z_{k,i}; \mathbf{w})$ represent the prediction losses of all samples on client $\mathcal{C}_k$'s local model and an individual sample on client $\mathcal{C}_k$'s local model respectively.

Let the learning rate be η and the current downloaded global model be $\mathbf{w}^{(t)}$. Each client $\mathcal{C}_k$ computes $\mathbf{g}_k^{(t)} = \nabla F_k(\mathbf{w}^{(t)})$ and uploads the shared gradient $\mathbf{g}_k^{(t)}$ to the server. Thus, the server aggregates these gradients and updates the global model as follows:

$$\mathbf{w}^{(t+1)} \leftarrow \mathbf{w}^{(t)} - \eta \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{g}_k^{(t)}. \tag{2}$$

B. Privacy Leakage from Gradients

We assume that the honest-but-curious adversary server has the goal of uncovering clients' local data after receiving shared gradients transmitted by each client and the ability to separately store and process shared gradients [3]. To recover the data from gradients of client $\mathcal{C}_k$, the adversary server firstly randomly initializes a dummy data $x'_{k,i}$ and a dummy label $y'_{k,i}$. The adversary server then feeds the dummy sample $z'_{k,i} = (x'_{k,i}, y'_{k,i})$ into models to get dummy gradients $\mathbf{g}'_{k,i}$ by

$$\mathbf{g}'_{k,i} = \frac{\partial f(z'_{k,i}; \mathbf{w})}{\partial \mathbf{w}}. \tag{3}$$

If dummy gradients can approach real shared gradients, the dummy data gets close to the real local data. Thus, the reconstructed data $x^*_{k,i}$ and the reconstructed label $y^*_{k,i}$ can be obtained by minimizing the following objective

$$x^*_{k,i}, y^*_{k,i} = \arg\min_{x'_{k,i}, y'_{k,i}} \|\mathbf{g}'_{k,i} - \mathbf{g}_{k,i}\|^2, \tag{4}$$

where the distance $\|\mathbf{g}'_{k,i} - \mathbf{g}_{k,i}\|^2$ is differentiable with regard to the dummy data $x'_{k,i}$ and the dummy label $y'_{k,i}$. Therefore, the distance can be optimized using gradient descent algorithms. Under the attack of privacy leakage from gradients, the privacy performance can be measured by the peak signal-to-noise ratio (PSNR) between the reconstructed data $x^*_{k,i}$ and the original data $x_{k,i}$.

III. Proposed Scheme

A. System Model

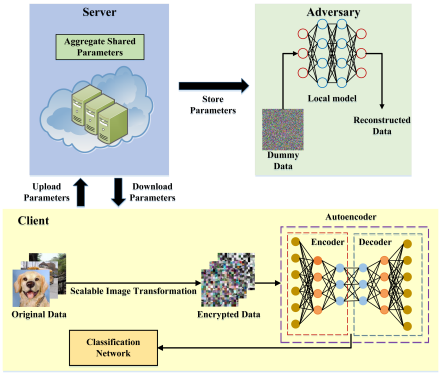


Fig. 1. Framework of the proposed privacy-preserving collaborative learning scheme.

Fig. 1 depicts the framework of the proposed privacy-preserving collaborative learning scheme where all clients' local data are images.

At the client side, each client firstly encrypts the original data $x_{k,i}$ using the scalable image transformation to generate the encrypted data $\hat{x}_{k,i}$. Secondly, the client updates the local model which is the classification-compliant autoencoder using the encrypted data $\hat{x}_{k,i}$ and the original label $y_{k,i}$.

The server aggregates shared gradients from all clients and updates the global model. Meanwhile, the adversary server is allowed to separately store and process shared gradients transmitted by each client. Because the server does not have the data encryption keys, it cannot obtain the original data $x_{k,i}$. Moreover, even if the attack of privacy leakage from gradients is conducted, the server can only obtain the data which approaches the encrypted data $\hat{x}_{k,i}$.

B. Scalable Image Transformation

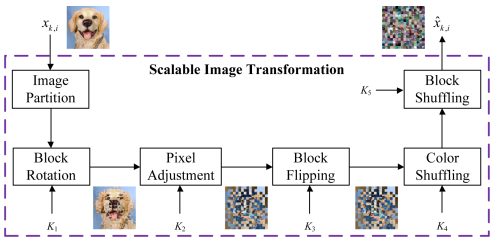


Fig. 2. Architecture of the scalable image transformation.

Before scalable image transformation, data augmentation (i.e., random cropping and horizontally/vertically flipping)

strategies [14] are conducted to restrain overfitting. As shown in Fig. 2, the scalable image transformation consists of a series of operators which are image partition, block rotation, pixel adjustment, block flipping, color shuffling, and block shuffling respectively.

- ImagePartition: The image partition operator partitions the original image $x_{k,i}$ with size of $w \times h$ into blocks $\mathcal{B}_l, l \in \{1, 2, \ldots, n_b\}$ with size of $b \times b$. The block size of each block can be adjusted so this operator is block size scalable.
- BlockRotation: The block rotation operator can rotate the input block by 90 degrees, 180 degrees, and 270 degrees respectively. The output block has four possible states. This operator is controlled by the encryption key K_1.
- PixelAdjustment: The pixel adjustment operator adjusts the L-bit pixel $x_{k,i}(j)$ with a pseudo-random bit r_j which is controlled by the encryption key K_2. If the input block has three color channels, pixels in each channel should be adjusted. The adjusted pixel is calculated by

$$\hat{x}_{k,i}(j) = \begin{cases} x_{k,i}(j), & \text{if } r_j = 0 \\ x_{k,i}(j) \oplus (2^L - 1), & \text{if } r_j = 1. \end{cases} \quad (5)$$

- BlockFlipping: The block flipping operator can achieve the horizontal flipping and the vertical flipping of the input block. The output block has three possible states. This operator is controlled by the encryption key K_3.
- ColorShuffling: The color shuffling operator shuffles three color channels of each pixel. The output block has six possible states. This operator is controlled by the encryption key K_4.
- BlockShuffling: The block shuffling operator shuffles all image blocks of the input image. The output image has $n_b!$ possible states. This operator is controlled by the encryption key K_5.

Algorithm 1 Scalable Image Transformation

Input Original image $x_{k,i}$ of client $\mathcal{C}_k$, encryption keys $K_i, i \in \{1, 2, \ldots, 5\}$, and scalable block size b.

Output Encrypted image $\hat{x}_{k,i}$.

1: Partition the original image $x_{k,i}$ with size of $w \times h$ into blocks $\mathcal{B}_l, l \in \{1, 2, \ldots, n_b\}$ with size of $b \times b$
2: **for** each block $\mathcal{B}_l, l \in \{1, 2, \ldots, n_b\}$ **do**
3: $\quad \mathcal{B}_l^{(1)} \leftarrow \text{BlockRotation}(\mathcal{B}_l, K_1)$
4: $\quad \mathcal{B}_l^{(2)} \leftarrow \text{PixelAdjustment}(\mathcal{B}_l^{(1)}, K_2)$
5: $\quad \mathcal{B}_l^{(3)} \leftarrow \text{BlockFlipping}(\mathcal{B}_l^{(2)}, K_3)$
6: $\quad \mathcal{B}_l^{(4)} \leftarrow \text{ColorShuffling}(\mathcal{B}_l^{(3)}, K_4)$
7: **end for**
8: Assemble the intermediary image $\bar{x}_{k,i}$ consisting of $\mathcal{B}_l^{(4)}, l \in \{1, 2, \ldots, n_b\}$
9: Shuffle all $b \times b$ blocks in the intermediary image $\bar{x}_{k,i}$ to generate the encrypted image $\hat{x}_{k,i}$ with the encryption key K_5
10: **return** $\hat{x}_{k,i}$

Algorithm 1 outlines the scalable image transformation process. As the image encryption strategy, the scalable image transformation has the following advantages. Firstly, the encrypted image state space is large enough so that it can effectively resist the exhaustive attack under the premise of the existing computing resources. Secondly, the encryption keys for each time can be different. The encryption security analysis is discussed in Section IV.

C. Classification-compliant Autoencoder

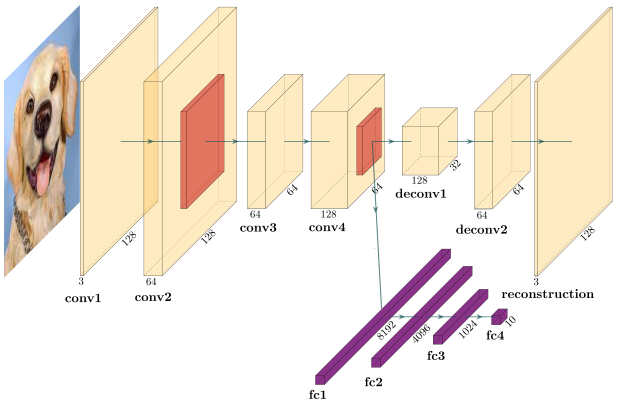


Fig. 3. Architecture of the classification-compliant autoencoder.

The difficulty of many classification tasks depends strongly on the choice of features used to represent the data. In the classification task-based collaborative learning, it is necessary to extract efficient features from high-dimensional data. The autoencoder can reduce the feature dimension of input data to achieve efficient feature representation [12]. Even if the input data are encrypted, the autoencoder can still obtain features to represent class information. As shown in Fig. 3, we design a classification-compliant autoencoder based on the VGG network [15]. This classification-compliant autoencoder includes an encoder, a decoder, and a classification network. The encoder and the decoder are illustrated as follows:

- The encoder converts encrypted images into latent features. The architecture of the encoder contains four convolutional layers with stride 1 and kernel size 3, four batch normalization layers, and two max-pooling layers.
- The decoder obtains decoded images from latent features. The architecture of the decoder contains two deconvolutional layers with stride 2 and kernel size 4, and two batch normalization layers.

The first objective of the classification-compliant autoencoder is to extract efficient features from encrypted images, which can be achieved by minimizing the decoding loss $\mathcal{L}_{\text{dec}}$:

$$\mathcal{L}_{\text{dec}} = \|\tilde{x}_{k,i} - \hat{x}_{k,i}\|^2, \quad (6)$$

where $\hat{x}_{k,i}$ is the encrypted image generated from the scalable image transformation and $\tilde{x}_{k,i}$ is the decoded image generated from the decoder. Smaller decoding losses mean that the classification-compliant autoencoder can obtain more meaningful features from encrypted images.

In a classification task with C classes, the classification-compliant autoencoder also aims to maintain decent classification accuracy by minimizing the cross-entropy-based classification loss $\mathcal{L}_{\text{cla}}$:

$$\mathcal{L}_{\text{cla}} = -\sum_{j=1}^{C} y_{k,i}(j) \log \hat{y}_{k,i}(j), \tag{7}$$

where $y_{k,i}(j)$ is the ground truth and $\hat{y}_{k,i}(j)$ is the local model output corresponding to the j-th class. Because the encoder converts encrypted images into efficient features, the architecture of the classification network can be a convolutional neural network or a fully connected neural network. The local model output equals the classification network output which varies according to the number of classes.

Finally, by combining the decoding loss and the classification loss, the proposed classification-compliant autoencoder loss $\mathcal{L}_{\text{cca}}$ is defined as:

$$\mathcal{L}_{\text{cca}} = \lambda_{\text{dec}} \mathcal{L}_{\text{dec}} + \lambda_{\text{cla}} \mathcal{L}_{\text{cla}}, \tag{8}$$

where $\lambda_{\text{dec}} \in [0, 1]$ and $\lambda_{\text{cla}} \in [0, 1]$ are hyper parameters which control the relative importance of decoding losses and classification losses respectively. We ensure that $\lambda_{\text{dec}} + \lambda_{\text{cla}} = 1$ and our test experiments show that the highest model accuracy can be obtained when $\lambda_{\text{dec}} = 0.5$.

The proposed classification-compliant autoencoder loss $\mathcal{L}_{\text{cca}}$ is a linear combination of two losses which are fully differentiable. Therefore, its gradients can be easily computed and the standard stochastic gradient descent algorithm [5] can be used to optimize it.

IV. Experimental Results and Analysis

A. Experimental Settings

1) Comparison Schemes: In our experiments, all comparison schemes are implemented in the PyTorch [16] platform which can efficiently calculate high-order gradients.

- **NPT [10]:** This scheme uses negative-positive transformation (NPT) to encrypt images for privacy-preserving deep learning. Moreover, data augmentation is performed before encryption to improve model accuracy.
- **ATS [11]:** To achieve privacy-preserving deep learning, this scheme applies automatic transformation search (ATS) to find optimal image transformation (e.g., horizontal/vertical shifting, brightness adjustment, and contrast enhancement) strategies.
- **EtC [17]:** This scheme utilizes block scrambling-based image encryption to encrypt images for privacy-preserving deep learning.
- **Our proposed scheme:** In our proposed scheme, the scalable image transformation and the classification-compliant autoencoder are designed for achieving privacy-preserving deep learning.

TABLE I
Training Parameter Settings for Different Datasets.

Dataset	Block size	Learning rate	Weight decay	Epoch
MNIST	4×4	0.05	0.0005	100
CIFAR-10	4×4	0.05	0.0005	200
Food-101	16×16	0.001	0.001	150
miniImageNet	16×16	0.01	0.001	150

2) Datasets: We choose two types of datasets in our experiments. Low-resolution datasets include MNIST [18] and CIFAR-10 [19]. MNIST comprises a training set of 60,000 images and a test set of 10,000 images and each image is a 28×28 grayscale image with a label from 10 classes. CIFAR-10 consists of 60,000 32×32 color images in 10 classes and there are 50,000 training images and 10,000 test images.

High-resolution datasets include Food-101 [20] and miniImageNet [21]. We randomly select 10 classes from them as clients' local datasets. For Food-101, we randomly select 900 images as the test set and remaining images as the training set. The original images are cropped to the size of 256×256. For miniImageNet, we crop original images to the size of 128×128 and 600 test images are randomly selected.

3) Model and Parameters: We implement a collaborative learning system with ten clients for low-resolution datasets and two clients for high-resolution datasets. Each client has the same number of local data. The collaborative learning system is trained using an NVIDIA GeForce RTX 3090Ti GPU with the stochastic gradient descent optimizer algorithm. Training parameter settings for different datasets are listed in Table I, where the block size is used in scalable image tansformation.

B. Ablation Studies

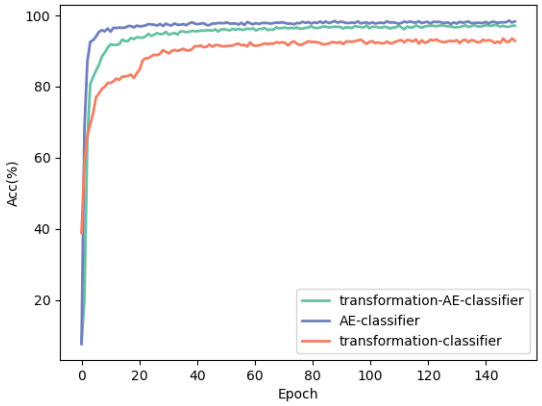


Fig. 4. Model accuracy curves of different model architectures on the test set of dataset MNIST during training.

We conduct ablation studies to isolate the effects of the decoding loss and the classification loss to verify the superiority of the classification-compliant autoencoder (AE). We consider three policies in ablation studies. The first policy is the proposed scheme without scalable image transformation

which denotes *AE-classifier*. The second policy is the proposed scheme which denotes *transformation-AE-classifier*. The third policy is the proposed scheme without classification-compliant autoencoder which denotes *transformation-classifier*. The classification network is ResNet-18 [22]. The model accuracy is calculated by the ratio of the number of correctly predicted samples to the number of test samples at the server side.

The model accuracy curves of different model architectures on the test set of dataset MNIST during training are shown in Fig. 4. It can be observed that *AE-classifier* can achieve the highest model accuracy. Due to the scalable image transformation, the model accuracy of *transformation-AE-classifier* is a little lower than that of *AE-classifier*. The model accuracy of *transformation-classifier* is obviously lower than that of *transformation-AE-classifier*. This demonstrates the effect of the autoencoder on efficient representation of clients' local data.

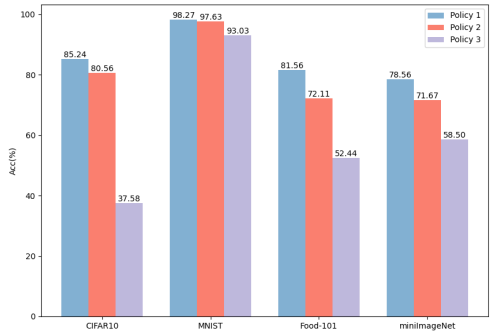


Fig. 5. Model accuracy comparison of three datasets under different policies. Policy 1: proposed scheme without scalable image transformation. Policy 2: proposed scheme. Policy 3: proposed scheme without classification-compliant autoencoder.

The model accuracy comparison of three datasets under different policies is shown in Fig. 5, which implies that the classification-compliant autoendocer performs well on low-resolution datasets and high-resolution datasets. Due to the privacy-preserving requirement, the scalable image transformation is essential even if the model accuracy slightly degrades.

C. Encryption Security Analysis

The size of the encrypted image state space determines the difficulty of an adversary encounters using an exhaustive attack. In the image encryption strategy based on scalable image transformation, the number of blocks can be determined by

$$n_b = \frac{w \cdot h}{b \cdot b}, \tag{9}$$

where w is the width of the original image, h is the height of the original image, and b is the block size. We formulate $N_{\mathrm{r}}(n_b)$, $N_{\mathrm{a}}(n_b)$, $N_{\mathrm{f}}(n_b)$, $N_{\mathrm{c}}(n_b)$, and $N_{\mathrm{s}}(n_b)$ as the possible state numbers of block rotation, pixel adjustment, block flipping, color shuffling, and block shuffling respectively in Algorithm 1 and the encrypted color image state space size $N_{\mathrm{trans}}(n_b)$ is represented by

$$\begin{aligned} N_{\mathrm{trans}}(n_b) &= N_{\mathrm{r}}(n_b) \cdot N_{\mathrm{a}}(n_b) \cdot N_{\mathrm{f}}(n_b) \cdot N_{\mathrm{c}}(n_b) \cdot N_{\mathrm{s}}(n_b) \\ &= 4^{n_b} \cdot 2^{n_b} \cdot 3^{n_b} \cdot 6^{n_b} \cdot n_b!. \end{aligned} \tag{10}$$

For dataset CIFAR-10 and $n_b = 4$, $N_{\mathrm{trans}}(n_b)$ equals $4^{64} \cdot 2^{64} \cdot 3^{64} \cdot 6^{64} \cdot 64!$ which is a huge number. It becomes difficult for the adversary to break the encryption.

D. Model Accuracy

TABLE II
MODEL ACCURACY (%) COMPARISON OF DIFFERENT PRIVACY-PRESERVING COLLABORATIVE LEARNING SCHEMES UNDER DATASETS MNIST AND CIFAR-10.

Scheme	MNIST	CIFAR-10
NPT [10]	95.92	59.31
ATS [11]	96.66	71.52
EtC [17]	97.17	64.09
Proposed	**97.63**	**80.36**

Table II reports the model accuracy (%) comparison of different privacy-preserving collaborative learning schemes under datasets MNIST and CIFAR-10. It can be seen that our proposed scheme performs best in terms of model accuracy.

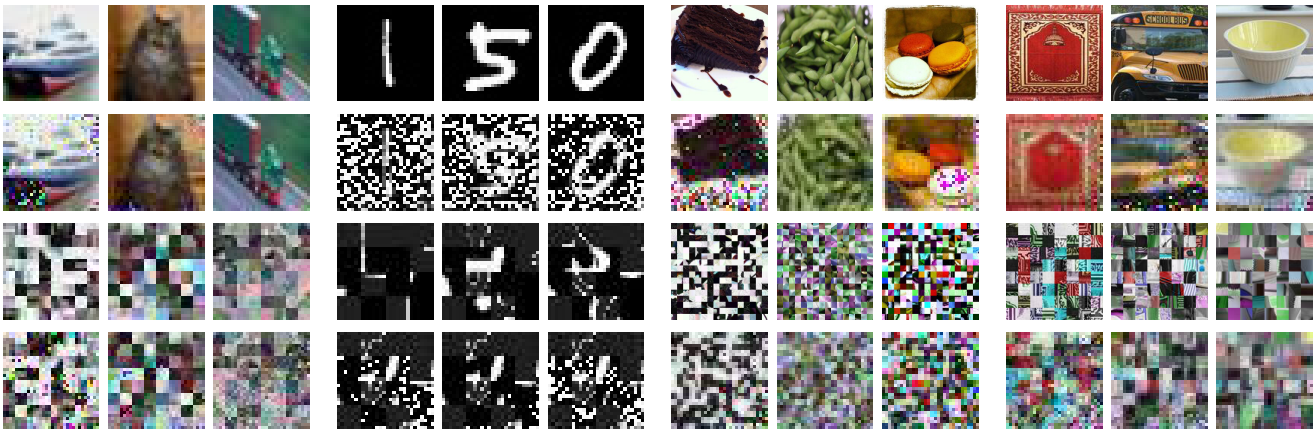
E. Privacy Preservation Performance

We choose deep leakage from gradients (DLG) [3] as the reconstruction attack method for privacy-preserving collaborative learning. In our experiments, the server performs DLG on received shared gradients after certain global communication rounds. The visual results of the DLG attack with and without our proposed scheme are shown in Fig. 6. As shown in row 2 of Fig. 6, it can be observed that an adversary can reconstruct clients' original images with high quality when our proposed scheme is not used. However, when our proposed scheme is utilized, the reconstruction quality of DLG becomes worse. It means that the privacy preservation performance has been enhanced due to our proposed scheme.

TABLE III
AVERAGE PSNR VALUES (DB) OF RECONSTRUCTED IMAGES UNDER THE DLG [3] ATTACK.

Dataset	Without privacy preservation	Proposed
MNIST	8.31	6.25
CIFAR-10	34.17	16.52
Food-101	9.26	5.61
miniImageNet	19.32	8.30

To quantitatively analyze the privacy preservation performance, Table III reports the average PSNR values (dB) of reconstructed images (i.e., images in row 2 and images in row 4 of Fig. 6) under the DLG [3] attack. It demonstrates that our proposed scheme has the ability to defend against gradient-based reconstruction attacks.



(a) CIFAR-10 (b) MNIST (c) Food-101 (d) miniImageNet

Fig. 6. Visual results of the DLG attack [3] with and without our proposed scheme. Row 1: original images. Row 2: reconstructed images without privacy preservation. Row 3: encrypted images with our proposed scheme. Row 4: reconstructed images with our proposed scheme.

V. Conclusion

We present a novel privacy-preserving collaborative learning scheme with scalable image transformation and autoencoder in this paper. The scalable image transformation algorithm is incorporated into the privacy-preserving collaborative learning scheme while considering model accuracy constraint. The features of permutated images are efficiently extracted by a classification-compliant autoencoder for enhancing classification accuracy. Experimental results indicate that our proposed scheme maintains satisfactory model accuracy while preserving local clients' data.

In the future, privacy-preserving collaborative learning which can defend against other attacks (e.g., member inference attacks) deserves investigation.

VI. Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0804102, in part by the National Natural Science Foundation of China under Grant 61802357, and in part by the Fundamental Research Funds for the Central Universities under Grant WK3480000009.

References

[1] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[2] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through gradients: Image batch recovery via gradinversion," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Jun. 2021, pp. 16332–16341.

[3] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. Ann. Conf. Neural Inf. Process. Syst.*, Dec. 2019, pp. 1–11.

[4] W. Qin, L. Yang, and J. Ma, "FedGR: A lossless-obfuscation approach for secure federated learning," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2021, pp. 1–6.

[5] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM Conf. Computer Commun. Secur.*, Oct. 2016, pp. 308–318.

[6] D. Yuan, X. Zhu, M. Wei, and J. Ma, "Collaborative deep learning for medical image analysis with differential privacy," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2019, pp. 1–6.

[7] S. Sharma and K. Chen, "Image disguising for privacy-preserving deep learning," in *Proc. ACM Conf. Computer Commun. Secur.*, Oct. 2018, pp. 2291–2293.

[8] S.-C. S. Cheung, M. U. Rafique, and W.-T. Tan, "Privacy-preserving distributed deep learning with privacy transformations," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2018, pp. 1–7.

[9] M. Aprilpyone and H. Kiya, "Block-wise image transformation with secret key for adversarially robust defense," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2709–2723, Mar. 2021.

[10] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, "Privacy-preserving deep neural networks with pixel-based image encryption considering data augmentation in the encrypted domain," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2019, pp. 674–678.

[11] W. Gao, S. Guo, T. Zhang, H. Qiu, Y. Wen, and Y. Liu, "Privacy-preserving collaborative learning with automatic transformation search," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Jun. 2021, pp. 114–123.

[12] A. Sellami and S. Tabbone, "Deep neural networks-based relevant latent representation learning for hyperspectral image classification," *Pattern Recognit.*, vol. 121, pp. 108224, Jan. 2022.

[13] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat.*, Apr. 2017, pp. 1–10.

[14] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, Jun. 2019, pp. 113–123.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, May 2015, pp. 1–14.

[16] PyTorch [Online]. Available: https://pytorch.org

[17] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 6, pp. 1515–1525, Jun. 2019.

[18] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[19] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009 [Online].
Available: http://www.cs.toronto.edu/~kriz/cifar.html

[20] L. Bossard, M. Guillaumin, and L. V. Gool, "Food-101–Mining discriminative components with random forests," in *Proc. Euro. Conf. Comput. Vision*, Sep. 2014, pp. 446–461.

[21] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Ann. Conf. Neural Inf. Process. Syst.*, Dec. 2016, pp. 3637–3645.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2016, pp. 770–778.