

Digital Finance

VAEFL: Integrating variational autoencoders for privacy preservation and performance retention in federated learning

Zhixin Li^{1,2}, Yicun Liu¹, Jiale Li¹, Guangnan Ye^{1,2,*}, Hongfeng Chai^{1,2},
Zhihui Lu^{1,2}, and Jie Wu^{1,2}

¹ School of Computer Science, Fudan University, Shanghai 200438, China

² Institute of FinTech, Fudan University, Shanghai 200438, China

Received: 30 December 2023 / Revised: 14 March 2024 / Accepted: 24 April 2024 / Published online: 21 June 2024

Abstract Federated Learning (FL) heralds a paradigm shift in the training of artificial intelligence (AI) models by fostering collaborative model training while safeguarding client data privacy. In sectors where data sensitivity and AI model security are of paramount importance, such as fintech and biomedicine, maintaining the utility of models without compromising privacy is crucial with the growing application of AI technologies. Therefore, the adoption of FL is attracting significant attention. However, traditional FL methods are susceptible to Deep Leakage from Gradients (DLG) attacks, and typical defensive strategies in current research, such as secure multi-party computation and differential privacy, often lead to excessive computational costs or significant decreases in model accuracy. To address DLG attacks in FL, this study introduces VAEFL, an innovative FL framework that incorporates Variational Autoencoders (VAEs) to enhance privacy protection without undermining the predictive prowess of the models. VAEFL strategically partitions the model into a private encoder and a public decoder. The private encoder, remaining local, transmutes sensitive data into a latent space fortified for privacy, while the public decoder and classifier, through collaborative training across clients, learn to derive precise predictions from the encoded data. This bifurcation ensures that sensitive data attributes are not disclosed, circumventing gradient leakage attacks and simultaneously allowing the global model to benefit from the diverse knowledge of client datasets. Comprehensive experiments demonstrate that VAEFL not only surpasses standard FL benchmarks in privacy preservation but also maintains competitive performance in predictive tasks. VAEFL thus establishes a novel equilibrium between data privacy and model utility, offering a secure and efficient FL approach for the sensitive application of FL in the financial domain.

Keywords Federated learning, variational autoencoders, deep leakage from gradients, AI model security, privacy preservation

Citation Li Z, Liu Y, Li J, et al. VAEFL: Integrating variational autoencoders for privacy preservation and performance retention in federated learning. Security and Safety 2024; **3**: 2024005. <https://doi.org/10.1051/sands/2024005>

1 Introduction

Artificial intelligence (AI) technologies have been increasingly applied in areas such as digital finance and biomedicine [1–4]. With the implementation of the General Data Protection Regulation (GDPR),

* Corresponding author (email: yegn@fudan.edu.cn)

the importance of data and model security has become more pronounced. Presently, AI techniques, especially Deep Neural Networks (DNNs), are popular and widely used in fields like computer vision and natural language processing due to their ability to learn complex patterns from large datasets. However, the high dependency of DNNs on data poses significant challenges when the data is distributed across multiple entities, given privacy and regulatory constraints.

Federated Learning [5] (FL) emerges as a distributed machine learning paradigm offering a solution that enables collaborative training of machine learning and deep learning models without compromising individual data privacy. Despite its promise, FL is not entirely immune to privacy threats, particularly the deep leakage from gradients [6, 7], which have been proven capable of reconstructing client data from shared gradients and model parameters [8]. Traditional privacy-preserving techniques in FL, such as Homomorphic Encryption [9, 10] (HE) and Differential Privacy [11, 12] (DP), present a trade-off between security and efficiency. HE, while secure, incurs excessive computational and communication overhead, making it impractical for large-scale DNN models. On the other hand, DP offers a more lightweight solution but at the cost of model accuracy and vulnerability to complex data reconstruction attacks.

To address these challenges, this work introduces VAEFL, an innovative FL framework that leverages Variational Autoencoders [13] (VAEs) to protect data privacy without compromising model prediction accuracy. VAEFL advances the concept of dividing client networks into private and public segments, a strategy that maintains data confidentiality by keeping sensitive data transformations localized while allowing shared learning benefits across the network. The incorporation of VAEs offers dual advantages: firstly, it significantly reduces the potential for data leakage as no model that directly handles raw data is exposed to the server; secondly, it allows the server to aggregate knowledge extracted from client models without accessing any public data. This novel approach not only enhances privacy protection in FL but also maintains the integrity and effectiveness of the learning process, demonstrating a substantial step forward in the realm of secure and efficient distributed machine learning.

This paper, titled “VAEFL”, makes the following three key contributions:

- (1) The incorporation of VAEs into the FL training architecture is proposed to mitigate deep gradient leakage attacks. This integration aims to secure client-side data and prevent the transmission and leakage of data feature gradients while maintaining the collective knowledge within the global model for competitive performance.
- (2) The utilization of knowledge distillation on the server side to reduce the aggregation overhead of encoder models. This ensures the global classifier’s performance while leveraging clients’ knowledge securely and efficiently.
- (3) Through comprehensive experiments, the performance of VAEFL is demonstrated to not only match but sometimes exceed that of baseline FL models. Our privacy analysis confirms VAEFL’s robustness against gradient inversion attacks, establishing a new standard in privacy-preserving FL, especially in sensitive sectors such as fintech.

2 Preliminaries and related works

2.1 Federated learning and privacy protection

FL is a novel paradigm in distributed machine learning computation, initially proposed by McMahan *et al.* [5]. The essence of FL lies in enabling numerous clients, such as mobile devices or distributed servers, to collaboratively train machine learning models with shared gradient parameters, while keeping the training data localized [14]. This approach transforms the traditional centralized training process into a decentralized one, where data remains within the local domain, thereby reducing risks associated with data transfer and storage on central servers [15]. It addresses significant concerns related to data privacy and security. Additionally, FL offers the advantage of leveraging diverse data sources, leading to potentially more robust and generalizable models [16].

However, despite its benefits, FL faces challenges in terms of efficiency and vulnerability to privacy attacks, such as model inversion and data reconstruction [17]. In a federated environment, data holders (clients) participating in the joint training of a model share only their model’s gradients (parameter updates) with the server side, which aggregates these shared gradients to train a global model. If the server is curious or malicious, once it obtains the gradients, attackers can use the method of gradient

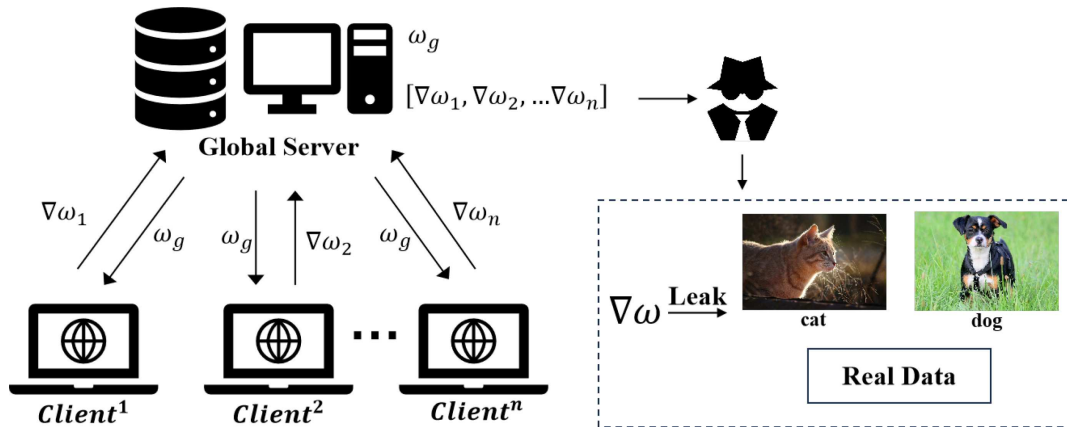


Figure 1. Illustration of the federated learning architecture showcasing the interaction between clients and a global server. Each client $client_i$ computes a gradient $\nabla \omega_i$ with respect to its local data and sends it to the global server. The server then aggregates these gradients (ω_g) to update the global model. The diagram also highlights a potential data leakage scenario, where the aggregated gradients can lead to the exposure of features from the real data, as depicted with the example images of a “cat” and a “dog”

reverse propagation to attempt to reconstruct the original data or key feature information that contributed to those gradients. This reverse process essentially involves solving an optimization problem, which is to find a set of input data that, for given model parameters, makes the gradients calculated through forward and backward propagation as consistent as possible with the gradients actually observed.

To address these issues, advanced cryptographic techniques like Homomorphic Encryption (HE) and privacy-preserving algorithms such as Differential Privacy (DP) have been integrated into the FL framework [18, 19]. Recent works have shown that this type of combination between DP(HE) and FL achieves great results [20, 21]. However, the introduction of these methods significantly impacts computational efficiency and model performance. Recognizing the limitations of HE and DP, a new approach in FL focuses on splitting neural networks into private and public models, sharing only the latter to enhance the privacy-efficiency balance. This strategy, while innovative, still grapples with the trade-offs between data privacy and model performance.

2.2 DLG and its impact on FL

Deep Leakage from Gradients [7] (DLG) is a challenge in deep learning, involving the potential leakage of sensitive information from a model through gradient signals. The root cause of this issue lies in the fact that, when training DNNs, the model may learn private information about the input data, which can be maliciously inferred by observing the model’s gradients. Attackers can gain insights into sensitive information about the training data by monitoring the gradients of model parameters, potentially leading to privacy breaches and security risks [22–25].

DLG has recently been recognized as a significant challenge in FL, critically undermining its foundational premise of privacy protection [7]. Research by Zhu and others first exposed the startling capability of DLG to reconstruct private training data from shared gradients, reshaping our understanding of privacy risks in FL. Subsequent studies by Zhao *et al.* [26, 27], further demonstrated DLG’s proficiency in achieving pixel-level data recovery and even retrieving label information from gradients, exacerbating the vulnerability of FL to complex privacy attacks. The development of DLG advanced into a more sophisticated stage with Ren *et al.* [28], introducing Generative Reversible Neural Networks (GRNN). GRNN marks a leap in privacy invasion techniques, utilizing generative models to create plausible yet fake data and labels, as opposed to the traditional method of direct regression from random initialization. This innovation signifies a more complex and subtle threat to FL privacy, underscoring the urgency of robust defence mechanisms against such advanced adversarial strategies. Yang *et al.* [29] made a comprehensive conclusion in gradient leakage attacks, encapsulating all DLG-related attacks and directing future ways to improve these attacks.

Table 1. Notations for all the used variables.

Variables	Interpretation
$\mathcal{E}, \mathcal{C}, \mathcal{E}_n, \mathcal{D}$	Respectively extractor function, classifier function, encoder and decoder
$\theta_E, \theta_C, \theta_{E_n}, \theta_D$	Parameters of extractor, classifier, encoder and decoder
F	Features
Z	Latent space
Z^*	Gaussian noise
α, β	Regularization parameters
$q(z x), p(x z)$	Distributions of encoder and decoder respectively
θ_{global}	Output of global model
K	Number of clients participating in aggregation process
T	Temperature parameter
$z_{\text{global}}, z_{\text{local}}$	Logits from global model and local model

These developments have spurred significant discussions within the FL community, driving researchers and practitioners to reassess and strengthen privacy protection mechanisms within the FL framework. The continuous evolution of DLG and its variants necessitates persistent efforts to develop more sophisticated and resilient privacy protection strategies, ensuring the safe and reliable application of FL, especially in scenarios involving sensitive data. Figure 1 depicts the FL architecture and the potential for DLG, illustrating how multiple clients share gradient information with a central server, which could lead to inadvertent exposure of private data features. This highlights the critical need for effective defence strategies in FL to address privacy concerns and safeguard against the reconstruction of sensitive data by adversaries.

2.3 VAE in FL

Generative models may offer a new paradigm for addressing DLG in the FL framework. FedCG [30] is a novel FL approach that leverages conditional generative adversarial networks to achieve a high level of privacy protection while maintaining competitive model performance. However, generative adversarial nets (GAN) [31] model training is often unstable and may lead to mode collapse, resulting in slow training. Variational Autoencoders (VAEs) have recently garnered attention in the FL landscape as a potent tool for enhancing data privacy [32–35]. VAEs, with their capability to transform data into a compressed, latent representation, present an innovative approach to tackling privacy concerns in FL. Unlike traditional methods that often rely on direct data exposure or simplistic anonymization techniques, VAEs offer a more sophisticated mechanism for data protection. They encode sensitive information into a latent space, effectively obfuscating the original data characteristics while retaining the essential features necessary for model training. This encoding process not only adds a layer of security against privacy threats like DLG but also aligns with the decentralized nature of FL, as it allows each client to locally process their data before participating in the collaborative learning process.

Incorporating VAEs into FL, however, is not without its challenges. One of the primary concerns is the maintenance of data utility after transformation. The encoded representations must preserve enough information to allow effective model learning, which can be a delicate balance to achieve. Furthermore, the integration of VAEs into FL systems needs careful consideration regarding model architecture and training dynamics to ensure that the benefits of VAEs are fully leveraged without compromising the collaborative learning process. Recent works have begun exploring these challenges, seeking to optimize VAE architectures for federated settings and experimenting with various training strategies to maximize both data privacy and model performance. These studies primarily employ VAEs to approximate the client data distribution, subsequently considering the generated data distribution for fairness decisions during global aggregation on the server side. Directly learning the local data distribution may lead to the leakage of distribution information [36–38]. The use of VAEs in FL, particularly in domains with stringent privacy requirements such as healthcare and finance, opens new avenues for research and development, promising a future where data privacy and model utility coexist harmoniously in a distributed learning environment. However, no existing works have employed VAEs in the FL framework to tackle the problem of DLG.

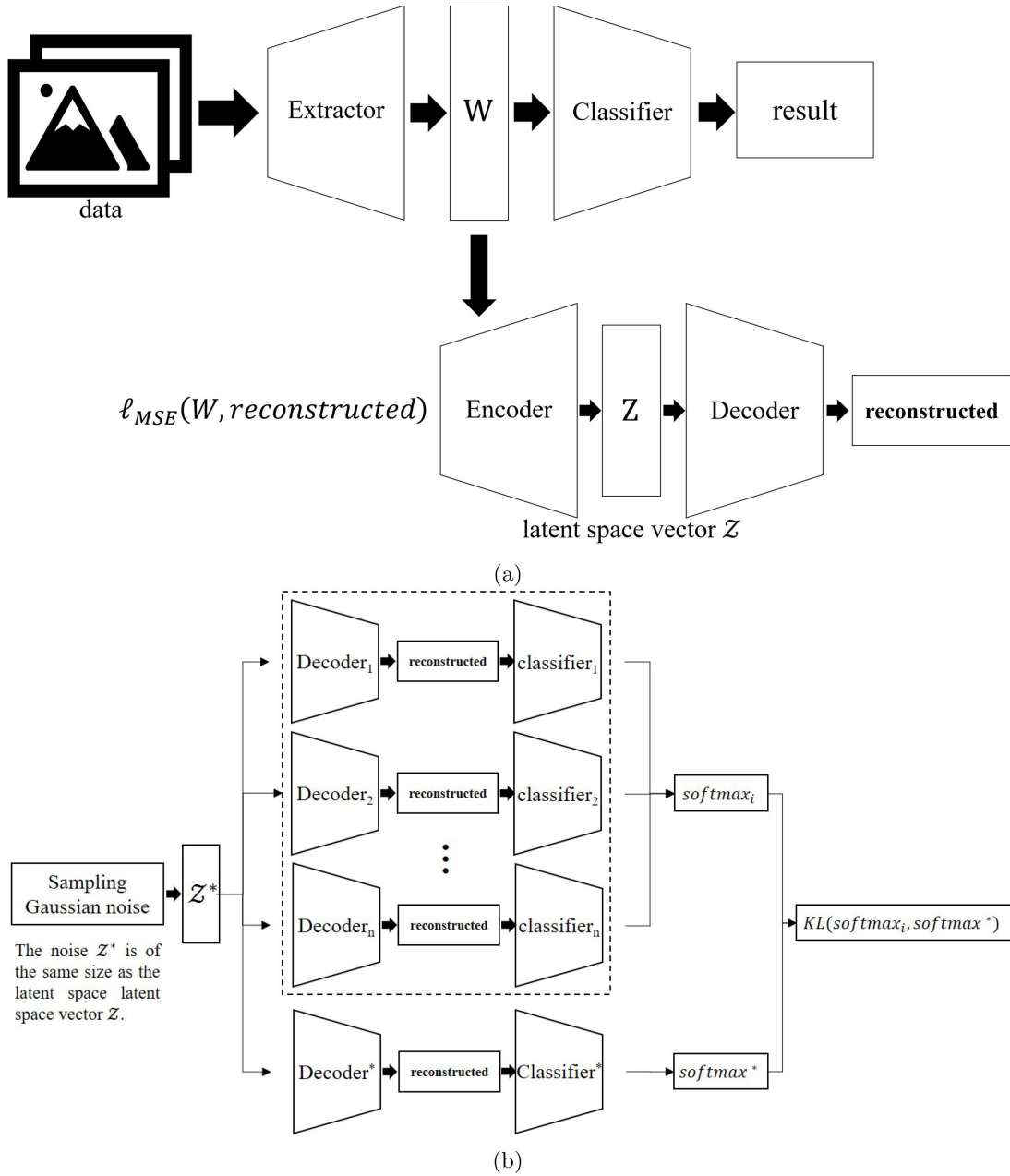


Figure 2. Illustration of the FL process with emphasis on privacy protection and data reconstruction. (a) The process of training the client model with data extraction and classification. (b) The server model illustrates the use of Gaussian noise for generating synthetic data

3 VAEFL: Methodology

3.1 Mathematical formulation

The VAEFL framework employs a federated approach to train a privacy-preserving model across multiple clients, each with its unique data distribution. Table 1 lists the notations for all the variables used in this paper. This subsection articulates the foundational knowledge of VAEFL, focusing on the privacy preservation and data reconstruction processes facilitated by VAEs.

As shown in Figure 2a, each client model consists of two primary components: an extractor and a classifier. The extractor function $\mathcal{E}(x; \theta_E)$ extracts features F from the input data x , and the classifier $\mathcal{C}(F; \theta_C)$ predicts the outcome based on these features. The VAE architecture further incorporates an

encoder-decoder pair, $\mathcal{E}_n(x; \theta_{E_n})$ and $\mathcal{D}(Z; \theta_D)$, which encodes the data into a latent space and reconstructs it, respectively. This mechanism is crucial for protecting data privacy, as the reconstruction from the latent space Z to the data space approximates the original data distribution while mitigating the risk of revealing sensitive information.

Upon the completion of local training, clients synchronize only the structure of their decoder and classifier models to the server, without uploading the feature representations F . This ensures that the original data and the learned feature representations do not leave the local environment, thus enhancing data privacy.

As illustrated in Figure 2b, the server model enhances data privacy by generating synthetic data through sampling Gaussian noise. The noise Z^* is sampled with the same dimensionality as the latent space vector Z , produced by the VAE encoder. This ensures that the synthetic data preserves the statistical properties of the original data without access to the clients' model weights or real labels.

Moreover, the server computes the Kullback–Leibler (KL) divergence between the output of the global classifier model and the outputs from each client's classifier model, as shown by the red box in Figure 2b. This step is crucial for optimizing the global model:

$$\text{KL}(\text{softmax}_i, \text{softmax}^*), \quad (1)$$

where softmax_i represents the softmax output of the i th client's classifier and softmax^* is the softmax output of the global classifier. This divergence measure helps to ensure that the global model is well-aligned with the clients' models, maintaining the collective intelligence of the federated network while preserving data privacy.

The server model is trained by mimicking the output distribution of the client model (soft labels), which is a form of distillation. The training process involves sampling from a Gaussian distribution in the latent space to generate synthetic data, followed by reconstruction of the data using the server's decoder and subsequent classification. The loss function of the server model includes a term for calculating the KL divergence between the server predictions and the weighted sum of client model predictions.

$$\min_{\theta_D} \sum_{i=1}^N \alpha \mathcal{L}_{\text{MSE}}(\text{reconstructed}_i) + \beta \mathcal{L}_{\text{KL}}(q, p). \quad (2)$$

Here, α and β are regularization parameters that balance the fidelity of the reconstructed data and the consistency with the client classifier, q is the logarithm of the soft maximum output of the server classifier, p is the weighted sum of client model predictions, processed through the soft maximum function and is gradient-free (i.e., it is a constant or has gradient stopped). Reconstructed_i is the data reconstructed by the server decoder, and $\mathcal{L}_{\text{KL}}$ is the KL divergence loss, which quantifies the difference between the predicted distributions, promoting alignment between the global model and the client model while preserving data privacy. Specifically: α is a hyperparameter that controls the weight of the Reconstruction Loss in the total loss function. The Reconstruction Loss measures the difference between the data reconstructed by the VAEs and the original input data. The larger the value of α , the more the model tends to reduce the reconstruction error during training, thereby improving the quality and fidelity of the reconstructed data. β is a hyperparameter that controls the weight of the KL divergence in the total loss function. The KL divergence is used to measure the difference between the distribution of the encoded latent space and the prior distribution (usually assumed to be Gaussian). By adjusting the value of β , the balance between the compactness of data encoding (i.e., regularization of the latent space) and the consistency with the client classifier can be managed. An increase in the value of β makes the model pay more attention to the regularization of the latent space, which helps to enhance data privacy protection.

3.2 Client model training and update

Each client aims to minimize the local loss function, which comprises two terms: the classification loss and the VAE reconstruction loss. The classification loss is defined as the cross-entropy between the predicted labels and the true labels:

$$\mathcal{L}_{\text{CE}}(C(E(x)), y) = - \sum_i y_i \log C(E(x_i)), \quad (3)$$

where E is the Extractor, C is the Classifier, x is the input data, and y is the corresponding label.

The VAE loss is a combination of reconstruction loss and the KL divergence, encouraging the encoded data to approximate a prior distribution, typically a Gaussian distribution:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{z \sim q(z|x)} [\log p(x|z)] - D_{\text{KL}}(q(z|x) || p(z)), \quad (4)$$

where $q(z|x)$ is the encoder's distribution, $p(x|z)$ is the decoder's distribution, and D_{KL} represents the KL divergence.

The reconstruction loss ensures that the decoded samples are similar to the original inputs:

$$\mathbb{E}_{z \sim q(z|x)} [\log p(x|z)] = -\frac{1}{N} \sum_i \|x_i - \text{VAE}_{\text{dec}}(\text{VAE}_{\text{enc}}(x_i))\|^2. \quad (5)$$

The KL divergence acts as a regularizer:

$$D_{\text{KL}}(q(z|x) || p(z)) = -\frac{1}{2} \sum_i (1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2), \quad (6)$$

where μ and σ are the mean and standard deviation of the encoded latent variables.

Algorithm 1: VAEFL Client Algorithm

Data: Client ID, training ($\mathcal{D}_{\text{train}}$), validation ($\mathcal{D}_{\text{val}}$), and test datasets ($\mathcal{D}_{\text{test}}$)

Result: Trained models and validation/test performance metrics

Initialize data loaders for $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{test}}$ with batch size *batch_size*.

Initialize models: Extractor E , Classifier C , VAE Encoder VAE_{enc} , VAE Decoder VAE_{dec} .

Define optimizers for E and C with learning rate lr and weight decay *weight_decay*.

Define loss functions: BCE, CE, MSE, and Cosine Similarity.

for each local training epoch **do**

 Train E and C with $\mathcal{D}_{\text{train}}$.

 Optionally, add Gaussian noise to inputs.

 Perform forward and backward propagation.

 Update parameters of E and C .

 Train VAE_{enc} and VAE_{dec} with $\mathcal{D}_{\text{train}}$.

 Perform VAE forward and backward propagation.

 Update parameters of VAE_{enc} and VAE_{dec} .

end

Evaluate the model on $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{test}}$.

Return performance metrics.

Algorithm 1 outlines the procedure for each client in the VAEFL framework. Each client initializes their local data loaders and the required neural network models including an Extractor, a Classifier, and the components of a VAE, namely the Encoder and Decoder. The algorithm proceeds by defining optimizers and loss functions for the training process. During each local epoch, the client trains the Extractor and Classifier on their local dataset, potentially enhancing data privacy through the addition of Gaussian noise. Simultaneously, the VAE components are trained to encode the local data into a latent space, thereby providing a privacy-preserving representation. The client then evaluates the performance of the trained models on local validation and test datasets, ensuring that the local model achieves a reliable performance before participating in the global model aggregation.

3.3 Server aggregation

The server's objective is to aggregate the client models in a way that improves the global model's performance while preserving data privacy. In the VAEFL framework, the server updates the global model by aggregating knowledge from the client models. This process integrates the outputs from the decoders and

classifiers provided by each client. Specifically, each client's output is a function of the Gaussian noise sample input. The server-side aggregation process can be described by the following formula:

$$\theta_{\text{global}} = \frac{1}{K} \sum_{k=1}^K f_k(z_k), \quad (7)$$

where:

- (1) θ_{global} represents the output of the global model.
- (2) K is the total number of clients participating in the aggregation process.
- (3) f_k represents the model of the k th client. This model consists of a decoder and a classifier, where the decoder maps the Gaussian noise sample z_k to an output, and the classifier maps the output to the final classification result.

Here, z_k is the Gaussian noise sample generated by the server for the k th client, and $f_k(z_k)$ is the client model's response to these samples. The global model output θ_{global} is the average of these responses, thus integrating the knowledge learned by the clients, while avoiding direct sharing of client weights, thereby enhancing data privacy protection.

The server then performs knowledge distillation to align the global model with the aggregated knowledge:

$$\mathcal{L}_{\text{distill}} = T^2 \times D_{\text{KL}} \left(\text{softmax} \left(\frac{z_{\text{global}}}{T} \right), \text{softmax} \left(\frac{z_{\text{local}}}{T} \right) \right), \quad (8)$$

where T is the temperature parameter that softens the softmax outputs, z_{global} is the logits from the global model, and z_{local} is the logits from the clients' models.

Algorithm 2: VAEFL Server Algorithm

Data: List of clients $\mathcal{C}$ and models

Result: Aggregated global model and performance metrics

Initialize global models: VAE Encoder $\text{VAE}_{\text{enc}}^{\text{global}}$, VAE Decoder $\text{VAE}_{\text{dec}}^{\text{global}}$, Classifier C^{global} .

Define optimizer for global models with learning rate lr .

Define loss functions: KL Divergence and Cross-Entropy.

for each global training epoch **do**

 Aggregate local client models into the global model.

for each batch in global iteration **do**

 Perform global training using aggregated model.

 Apply distillation to align global and local models.

 Update global model parameters.

end

end

Evaluate aggregated model performance across all clients $\mathcal{C}$.

Return aggregated performance metrics.

Algorithm 2 details the server-side operations within the VAEFL architecture. The server's role begins by initializing the global models that mirror the client's VAEs components and Classifier. It then determines the clients' contributions to the global model by considering their respective dataset sizes. In the training phase, the server aggregates the parameters of the client's local models, adjusting the global model by training it with the aggregated data. A key feature of this process is the use of distillation techniques to harmonize the local and global models, ensuring that the global model benefits from the diversity of all clients' data while preserving privacy. Post-training, the server assesses the aggregated model's performance across all clients, providing an overall metric of effectiveness and ensuring that the VAEFL system operates optimally. In the VAEFL framework, by integrating VAEs with a weighted aggregation mechanism, we effectively balance the complexity and convergence of the model. The local client loss function utilizes stochastic gradient descent. On the server side, a weighted aggregation strategy is employed, where the contribution of each client is weighted according to its proportion of the data volume. This weighted aggregation ensures that the global model can converge stably and effectively after multiple rounds of iteration. Additionally, the convergence of the algorithm is validated in Section 4.2 Experiment Results.

3.4 Workflow of VAEFL

The VAEFL framework operates through a collaborative yet privacy-preserving mechanism, involving a series of interactions between client-side and server-side entities. The workflow is orchestrated to ensure the confidentiality of the data while leveraging the distributed nature of the data for model training. Here, the core workflow steps of VAEFL are outlined as follows:

- (1) Initialization: The server initializes a global model, which includes a VAE Encoder, VAE Decoder, and a Classifier. Each client also initializes their local versions of these models.
- (2) Local training: On the client side, local training involves the use of VAEs to encode sensitive data into a latent space, followed by training the classifier on these encoded representations. This step ensures that each client's data remains private and is never shared in its original form.
- (3) Model aggregation: After local training, clients send their model parameters to the server. The server then performs a weighted aggregation of the received parameters to update the global model, considering the size of each client's dataset as the weight.
- (4) Knowledge distillation: To further align the global model with the clients' models, the server applies knowledge distillation techniques. This step involves training the global model to imitate the output distributions of the clients' models, effectively transferring their learned knowledge without exposing any private data.
- (5) Global model broadcast: Once the global model has been updated and distilled, it is sent back to the clients. This global model serves as a starting point for the next round of local training.
- (6) Model fine-tuning: Clients then fine-tune the global model using their local data, which allows the model to adapt further to the specific characteristics of each client's dataset.
- (7) Convergence check: This iterative process of local training, model aggregation, distillation, and broadcasting continues until the global model converges or a predefined number of iterations is reached.
- (8) Evaluation: Finally, the performance of the global model is evaluated on each client's local test set to ensure that the model has not only preserved privacy but also maintained high predictive accuracy.

4 Evaluation results

In this section, the performance of the proposed VAEFL framework is compared against traditional FL baselines and the state-of-the-art method FedCG. Furthermore, the privacy-preserving capabilities inherent to the VAEFL approach are assessed.

4.1 Experimental setup

4.1.1 Model architecture

Our chosen architecture is the LeNet5 model, as introduced by LeCun *et al.* [39], which serves as the backbone network for classification tasks within FL systems. In our design, LeNet5 is partitioned into two segments: the initial two convolutional layers, designated as the *private* feature extractor, and the subsequent linear layers, termed the *public* classifier. This bifurcation enables us to investigate the efficacy of VAEFL in segregating private and public model components.

LeNet-5 has been empirically proven to be effective in handling image classification tasks, boasting a concise structure with minimal training overhead compared to other networks such as ResNet and VGGNet. Moreover, the proposed approach in this paper is tailored to the FL framework. Regardless of the backbone network employed, corresponding gradient leakage issues arise, which can be adeptly addressed by this proposed method. Furthermore, across various dataset configurations, our method consistently demonstrates improvement trends in different network architectures like ResNet and VGGNet.

Hence, after careful consideration, LeNet-5 is chosen as the backbone network for this study.

4.1.2 Datasets

To rigorously evaluate our VAEFL model, Five diverse image datasets are employed, including FMNIST [40], CIFAR10 [41], Digit5 [42], Office-Caltech10 [43], and DomainNet [42]. FMNIST and CIFAR10 are

Table 2. Main hyperparameters of VAEFL

Hyperparameters	Recommendation
α	0.1
β	0.01
Learning rate	Cat{0.001, 0.0003}
Weight decay	Float{0.0001}
Batch size	Cat{4, 8, 16}
Global epochs	Int{20, 50}
Local epochs	Int{20, 50}
Training iterations	Int{200, 500}
Global iterations per epoch	Int{100, 200}
Dropout	0.0
Scheduler	Cat{MSE, cosine}
Gaussian noise Z^*	Float{0.01, 0.1, 1.0}

utilized to simulate an IID (independently and identically distributed) setting, while the remaining datasets, characterized by their collection from heterogeneous domains, naturally constitute a Non-IID environment. Notably, Digit5 amalgamates five distinct digit recognition benchmarks, namely MNIST, Synthetic Digits, MNIST-M, SVHN, and USPS. Office-Caltech10 encompasses 10 categories sourced from four distinct domains: Amazon, DSLR, Webcam, and Caltech. DomainNet, on the other hand, includes six domains: Clipart, Infograph, Painting, Quickdraw, Real, and Sketch.

4.1.3 Baselines

For comparative analysis, baseline methodologies include FedAvg [5], FedProx [14], FedDF [44], FedSplit [45], and FedGen [46]. These can be broadly categorized into two classes: the first, comprising FedAvg, FedProx, and FedDF, involves sharing the entire network architecture, including both private extractor and public classifier, with the server. The second class, encompassing FedSplit and FedGen, entails clients sharing only the public classifier component with the server. Our VAEFL model is also compared against the state-of-the-art method FedCG [30], providing a comprehensive evaluation of its performance.

4.1.4 Configurations

Hyperparameters are essential to VAEFL in the experiments, thus a list of recommended settings is given in Table 2. In the experiments, the specific values of α and β often need to be adjusted and optimized through cross-validation or predefined experimental settings. Generally, the range of these parameters can vary from 0.01 to 1.0, and the specific values depend on the characteristics of the dataset and the specific needs of model training. In our experimental setup, through multiple sets of experiments, we found that setting α to 0.1 and β to 0.01 provides a good balance in most cases, ensuring the accuracy of data reconstruction while maintaining good consistency with the client classifier. In the implemented experiments of VAEFL, a total of 200 global communication rounds were conducted, with each client undergoing 20 local epochs throughout the experiment. The batch size was fixed at 8, and the Adam optimizer was chosen with a learning rate of $3e-4$ and a weight decay of $1e-4$. For the FMNIST, CIFAR10, and Digit5 datasets, 2000 images were randomly selected to constitute each client's training set. In the case of Office-Caltech10 and DomainNet, 50% of the data from each domain was utilized for training. In the case of Digit5, Office-Caltech10, and DomainNet datasets, each domain, except for the MNIST domain in Digit5 and the Painting domain in DomainNet, was treated as an individual client. Conversely, in the FedDF setup, the MNIST and Painting domains were employed as distillation data.

The model's performance was evaluated using validation and testing on client datasets, with accuracy measurements taken across five random seeds. Additionally, the running time per epoch for each method on all datasets was recorded to analyze model efficiency. All experiments were executed on NVIDIA A100 GPUs.

Table 3. Experimental results of various FL methods on different datasets

Method	FMNIST(4)	CIFAR10(4)	CIFAR10(8)	Digit5(4)	Office(4)	DomainNet(5)
Local	79.40 $\pm$ 0.90	40.00 $\pm$ 1.84	40.91 $\pm$ 0.65	83.23 $\pm$ 0.38	63.68 $\pm$ 5.74	50.22 $\pm$ 0.92
FedAvg	82.69 $\pm$ 0.59	47.28 $\pm$ 1.62	48.30 $\pm$ 0.74	84.37 $\pm$ 0.52	63.96 $\pm$ 2.84	49.72 $\pm$ 0.96
FedProx	83.42 $\pm$ 0.86	49.27 $\pm$ 1.03	51.88 $\pm$ 1.08	84.20 $\pm$ 0.61	63.69 $\pm$ 2.99	50.68 $\pm$ 1.06
FedDF	83.60 $\pm$ 0.66	46.90 $\pm$ 2.08	50.64 $\pm$ 0.60	83.05 $\pm$ 0.84	(*)	50.86 $\pm$ 0.55
FedSplit	82.07 $\pm$ 0.63	44.63 $\pm$ 1.07	45.87 $\pm$ 1.17	83.89 $\pm$ 0.50	62.73 $\pm$ 1.42	50.79 $\pm$ 0.95
FedGen	81.60 $\pm$ 0.72	44.19 $\pm$ 1.53	45.41 $\pm$ 1.79	82.43 $\pm$ 0.94	62.20 $\pm$ 1.70	50.14 $\pm$ 1.10
FedCG	83.63 $\pm$ 0.51	46.49 $\pm$ 0.47	48.13 $\pm$ 0.79	84.38 $\pm$ 0.73	66.70 $\pm$ 1.28	51.90 $\pm$ 1.14
VAEFL	83.62 $\pm$ 0.60	47.18 $\pm$ 0.50	48.21 $\pm$ 1.07	84.43 $\pm$ 0.60	66.18 $\pm$ 1.93	51.37 $\pm$ 0.68

Notes. VAEFL demonstrates competitive performance across multiple datasets, highlighting the efficacy of the approach in FL scenarios. (*) indicates that no result is measured.

4.2 Experiment results

4.2.1 Performance evaluation

As indicated in Table 3, the VAEFL method demonstrates competitive results across multiple datasets. In particular, the method outperforms others in the “Digit5(4)” and “DomainNet(5)” scenarios, highlighting its robustness in varied FL environments. The bold entries in the table denote the top-performing methods for each dataset. It is noteworthy that while the VAEFL method does not always lead to the highest accuracy, its performance is consistently close to the best, with less variability in results as indicated by the smaller standard deviations. In comparison to the state-of-the-art method FedCG, VAEFL surpasses FedCG in accuracy across CIFAR10(4), CIFAR10(8) and Digit(4) experimental setups. In the remaining experimental settings, including Office(4), FMNIST(4), and DomainNet(5), VAEFL’s accuracy differs by less than 1% compared to FedCG, yet consistently outperforms the baseline in all three settings. Furthermore, VAEFL incurs smaller computational costs and time expenditures than FedCG, emphasizing its superiority in terms of accuracy with reduced computational burdens. This underlines the effectiveness of VAEFL in ensuring stable performance across different types of data distributions.

The “Office(4)” and “DomainNet(5)” datasets, which have higher complexity, show that methods with advanced generalization capabilities, such as FedCG and VAEFL, achieve better performance. This suggests that these methods can better handle the heterogeneous data distributions that are characteristic of real-world FL scenarios. The results from the “CIFAR10(4)” and “CIFAR10(8)” datasets indicate that increasing the number of classes does not significantly impact the performance of VAEFL, implying that it can scale well with the complexity of the task.

Overall, the experimental results validate the efficacy of the VAEFL approach, especially in settings where data is distributed and non-IID. Future work may explore further enhancements to the method, such as incorporating domain adaptation techniques to address the challenges presented by datasets with significant distribution shifts.

The ultimate objective of FL is to attain superior performance on local test data using the received network. Consequently, a further comparison is made between methods and the Local method on each dataset. On IID scenarios, such as “CIFAR(8)” shown in Figure 3a, all methods perform better than Local, among which all the “fully-shared” networks apparently outperform other methods. In Non-IID scenarios like Figure 3b, it’s easy to observe that FedCG and VAEFL outperform other methods.

4.2.2 Privacy evaluation

Peak Signal-to-Noise Ratio (PSNR) is selected as the metric to quantify the similarity between each picture recovered from DLG and the original image. A higher PSNR score means less difference between two pictures. Different standard deviations are applied to generate various Gaussian noises for shared gradients.

As shown in Table 4, VAEFL has an apparent advantage over FedAvg on the PSNR metric. Though FedAvg can achieve better accuracy sometimes (on Digit5 and CIFAR10 datasets), it suffers from a high risk of privacy leakage. Adding different Gaussian noise to the gradients can reduce this question, but

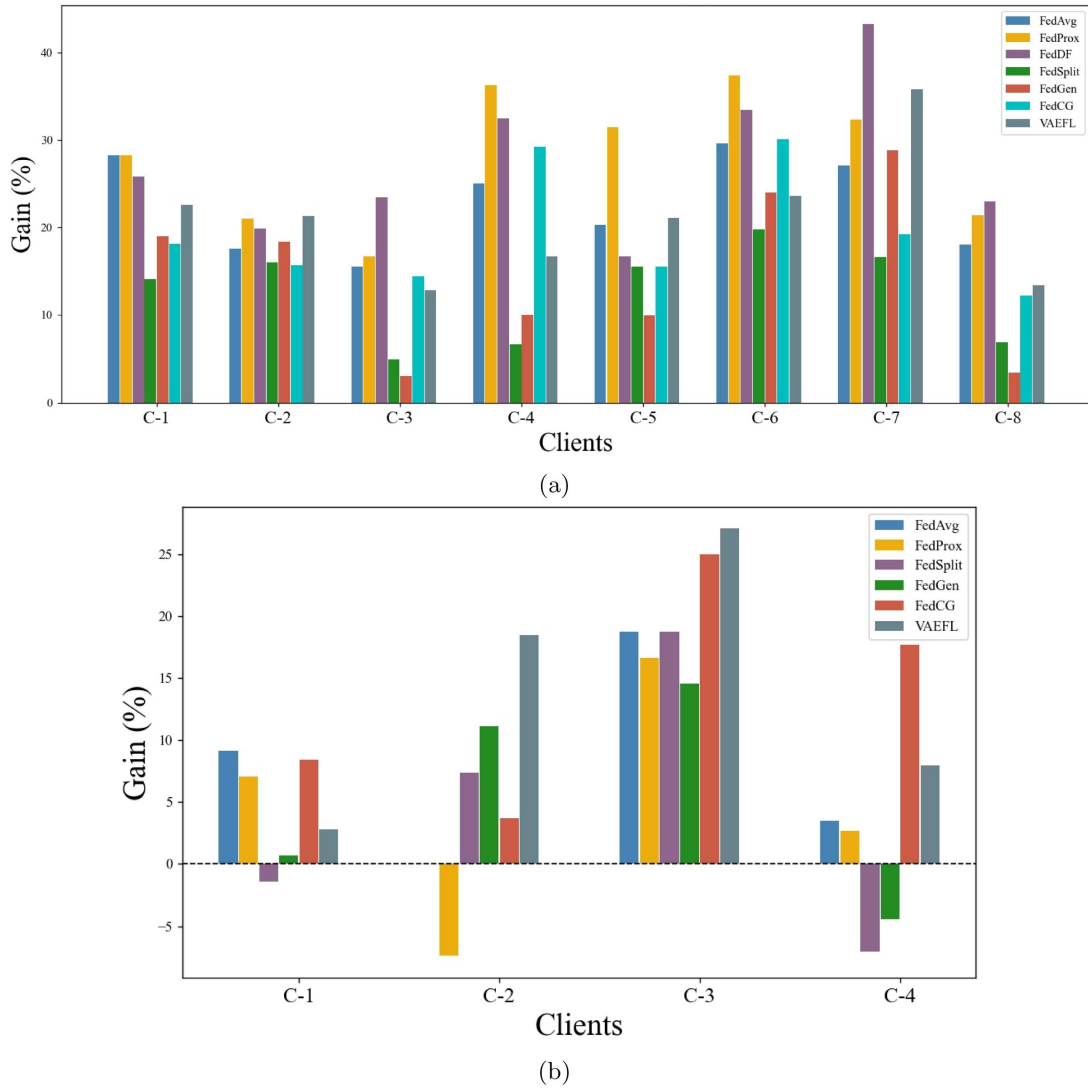


Figure 3. Accuracy gains achieved by previous methods and VAEFL over Local of each client on two datasets. The vertical axis denotes the difference between each method and Local. A positive(negative) gain denotes the client performs better(worse) than the client of Local method. (a) 8 Clients on CIFAR dataset; (b) 4 Clients on Office dataset

Table 4. Comparison of different FL methods on CIFAR10, Digit5, and Office datasets

Dataset	Metric	FedAvg	FedAvg (DP: 0.001)	FedAvg (DP: 0.1)	VAEFL
CIFAR10	Accuracy (%)	47.28	40.04	37.92	47.18
	PSNR (dB)	20.98	18.86	6.85	8.27
Digit5	Accuracy (%)	84.43	68.57	66.98	84.37
	PSNR (dB)	21.28	19.32	6.42	6.17
Office	Accuracy (%)	63.96	51.95	50.9	66.18
	PSNR (dB)	20.50	19.47	7.12	5.94

lowering the PSNR is at the cost of the drop in accuracy (up to 18%). On the other hand, VAEFL can better protect client privacy from being leaked without losing its vantage on accuracy. According to Figure 4, compared to other scenarios, VAEFL can achieve the highest accuracy over different privacy-protected methods (7% or higher).

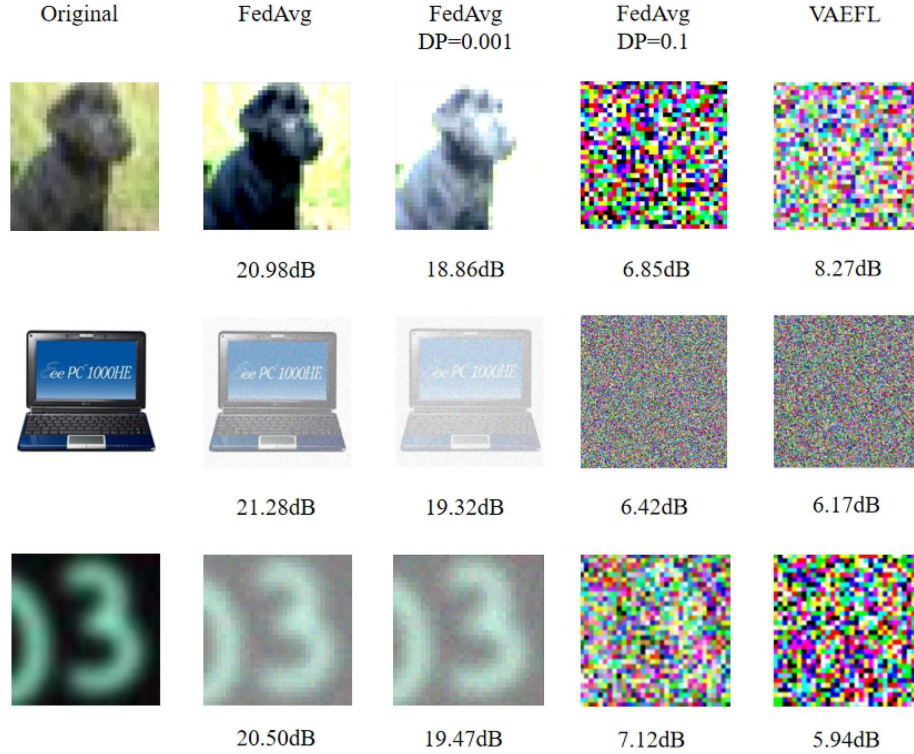


Figure 4. Restored images of CIFAR10, Office and Digit5 datasets using FedAvg with different DP and VAEFL methods. PSNR score is recorded under each picture

4.2.3 Efficiency evaluation

The computational complexity analysis of the VAEFL method encompasses three main aspects: client model training, server-side aggregation process, and communication costs. On the client side, VAEFL involves the training of VAEs, with its computational complexity primarily depending on the number of layers and neurons per layer in the encoder and decoder. The computational complexity of the server-side aggregation process is related to the number of clients and model parameters. Communication costs are proportional to the number of clients and the total number of model parameters. Experiments demonstrate that the additional computational burden introduced by incorporating VAEs on the client side is minimal, with the algorithm's convergence time being similar to methods that do not employ VAEs. Figure 5 shows that VAEFL's workflow does not consume enormous extra time, especially in "DomainNet(5)" and "Digit5(4)" experimental settings, where its time consumption is close to baselines which share the full networks with the server. Unlike FedCG, which takes a large scale of time to locally train and communicate, VAEFL is over twice as quick as FedCG in the training process on average, substantially saving lots of time and computational resources. For example, in the "Digit5(4)" experimental scenario, FedCG takes 580s to implement a full "local train and communication" circle, while VAEFL only takes 230s, which reduces the training time to only 40% of FedCG's without noticeable accuracy loss.

For other methods that run faster than VAEFL, Local, FedGen and FedSplit cannot gain similar accuracy to VAEFL, while FedAvg, FedProx and FedDF suffer from the risk of gradient leakage. Overall, VAEFL makes a trade-off among accuracy, privacy and efficiency, retaining outstanding performance on all 3 dimensions.

5 Discussion

The experimental results presented in Table 3 and Figure 3 substantiate the efficacy of the VAEFL framework when juxtaposed with conventional FL methods and the state-of-the-art FedCG. VAEFL

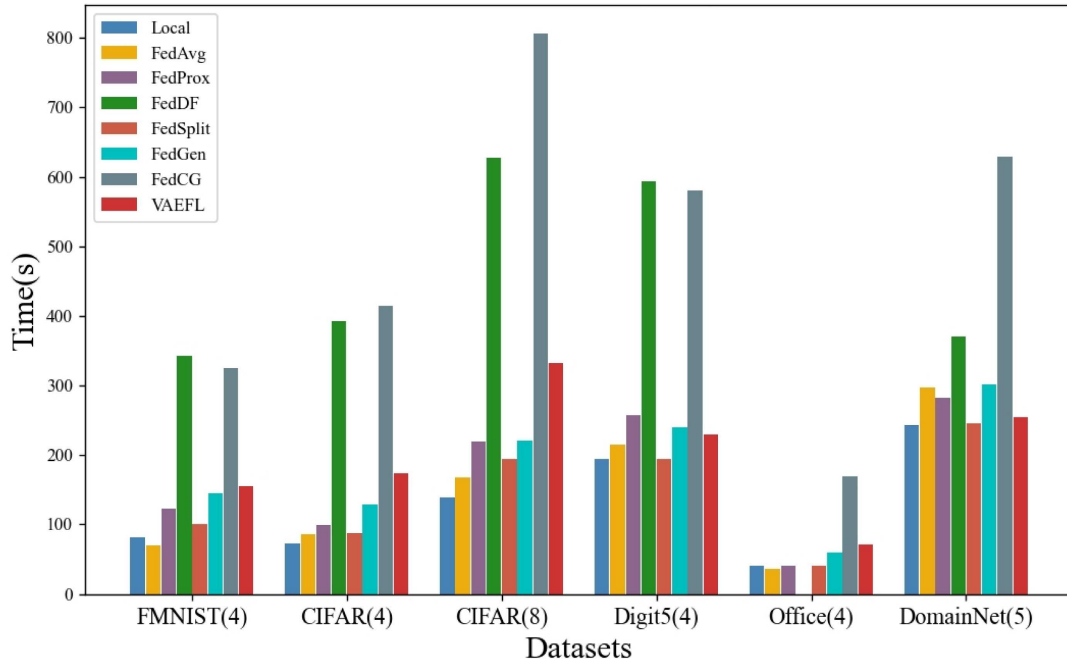


Figure 5. Time consumption per epoch of each method running on different datasets. In comparison with other methods, VAEFL takes an “acceptable” extra running time over other baselines, while it reduces the time consumption over 50% compared with sota-method FedCG. As shown in Table 3, VAEFL achieves similar accuracy to FedCG, while being over 2 times more efficient

consistently outperforms the baseline methods on the FMNIST and Digit5 datasets while maintaining competitive performance on CIFAR10, Office, and DomainNet datasets.

VAEFL’s superior performance on FMNIST and Digit5 can be attributed to its robust feature extraction mechanism facilitated by the VAE, which is particularly adept at handling the complexity and variability inherent to these datasets. Moreover, the marginal performance difference observed on the Office and DomainNet datasets suggests that VAEFL can adapt to diverse data distributions, a crucial attribute for FL systems operating in real-world scenarios.

The accuracy gains depicted in Figures 3a and b highlight VAEFL’s advantage over the Local method, especially on the “Office(4)” dataset, where the gains are pronounced across all clients. This indicates that VAEFL’s approach to learning shared representations is highly beneficial when dealing with heterogeneous data sources.

As to privacy, which is shown in Figure 4 and Table 4, VAEFL perfectly protects clients’ privacy from being leaked. Pictures reconstructed from DLG are hardly illegible, retaining a relatively low PSNR score without losing any of the accuracy. In the CIFAR10, Office, and Digit5 settings, images restored by VAEFL through DLG exhibit a significant advantage in PSNR, surpassing FedAvg methods without added DP by over 13 points. On the Office dataset, VAEFL not only achieves an accuracy improvement of more than 2% but also maintains an accuracy difference of less than 0.1% compared to FedAvg on the CIFAR10 and Digit5 datasets.

It is noteworthy that in the case of adding Differential Privacy to FedAvg, although VAEFL’s PSNR values are comparable to $DP = 0.1$ FedAvg, the fact that data with PSNR values less than 10 are no longer discernible to the naked eye suggests that VAEFL and $DP = 0.1$ FedAvg are similar in terms of privacy protection. However, the crucial point to emphasize is that VAEFL’s accuracy averages approximately 13% higher, indicating a significant superiority in balancing accuracy and privacy protection. Combined with accuracy, VAEFL is superior over the baseline FedAvg both with and without adding Gaussian noise, which demonstrates the advantage of our proposed method.

In terms of time efficiency, as shown in Figure 5, VAEFL demonstrates an “acceptable” additional computational cost compared to other baseline methods. Notably, it reduces time consumption by over

50% when compared to the state-of-the-art method FedCG. This is a significant finding as it balances the trade-off between performance and efficiency, two pivotal factors in the deployment of FL systems.

However, the increased time consumption per epoch for VAEFL, as compared to FedAvg and FedProx, necessitates a discussion on the practicality of its deployment in scenarios with stringent time constraints. Despite this, the substantial gains in performance and the reduction in time consumption relative to FedCG make VAEFL a compelling choice for scenarios where accuracy is of paramount importance and moderate time delays are acceptable.

In conclusion, VAEFL presents a promising direction for FL research, particularly in privacy-preserving and complex data environments. Future work may focus on optimizing the VAEFL framework to enhance computational efficiency further while preserving, if not improving, its accuracy across diverse federated settings.

6 Conclusion

The VAEFL framework introduces a transformative approach to FL, expertly balancing privacy sanctity with practical utility. It ingeniously allows clients to partake in collaborative learning without exposing sensitive data, thus redefining collaborative intelligence. VAEFL's novel method enhances learning confidentiality, fostering essential trust for its adoption in privacy-conscious sectors. Future enhancements aim to boost computational efficiency and accuracy within various federated contexts. VAEFL's advancements signal a move towards more secure, efficient, and cooperative data handling, promoting broader acceptance in sectors where privacy is paramount.

Conflict of interest

The authors declare that they have no conflict of interest.

Data Availability

The dataset of CIFAR10 is available at: [CIFAR10](#), the dataset of FMNIST is available at: [FMNIST](#), the dataset of Office-Caltech10 is available at: [Office-Caltech10](#), the dataset of Digit5 is available at: [Digit5](#), and the dataset of DomainNet is available at: [DomainNet](#). Code is available at: [Code](#)

Authors' Contributions

Zhixin Li contributed to the framework design, as well as most of the experimental studies and manuscript writing, serving as the first author. Yicun Liu contributed to some experimental studies, extra experiments and manuscript writing of the Evaluation Results and Discussion sections, as well as the proofreading and editing of the final paper. Jiale Li contributed to the manuscript writing of the Introduction section. Prof. Hongfeng Chai, Prof. Zhihui Lu and Prof. Jie Wu helped revise the manuscript. Prof. Guangnan Ye led this research project.

Acknowledgements

We thank all anonymous reviewers for their helpful comments and suggestions.

Funding

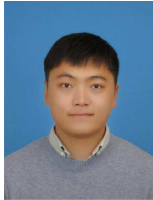
The work of this paper is supported by the Yangtze River Delta Science and Technology Innovation Community Joint Research Project (2022CSJGG0800) and the Shanghai Science and Technology Project (22510761000).

References

- [1] Cao L. AI in finance: Challenges, techniques, and opportunities. *ACM Comput Surv (CSUR)* 2022; **55**: 1–38.
- [2] Holzinger A, Keiblinger K, Holub P, et al. AI for life: Trends in artificial intelligence for biotechnology. *New Biotechnol* 2023; **74**: 16–24.
- [3] Rajpurkar P, Chen E, Banerjee O, et al. AI in health and medicine. *Nat Med* 2022; **28**: 31–38.
- [4] Weber P, Carl KV and Hinz O. Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Manag Rev Q* 2023; 1–41.
- [5] McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: *Artificial Intelligence and Statistics*, PMLR, 2017, 1273–1282.
- [6] Goodfellow IJ, Shlens J and Szegedy C. Explaining and harnessing adversarial examples. *Statistics* 2014; **1050**: 20.
- [7] Zhu L, Liu Z and Han S. Deep leakage from gradients. *Adv Neur Inf Process Syst* 2019; 32.
- [8] Li Z, Zhang J, Liu L, et al. Auditing privacy defenses in federated learning via generative gradient leakage. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 10132–10142.

- [9] Jin W, Yao Y, Han S, et al. FedML-HE: An efficient homomorphic-encryption-based privacy-preserving federated learning system. In: International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023, 2023.
- [10] Zhang Q, Jing S, Zhao C, et al. Efficient federated learning framework based on multi-key homomorphic encryption. In: Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021), 2022, Springer, 88–105.
- [11] Wei K, Li J, Ding M, et al. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Trans Inf Forens Secur* 2020; **15**: 3454–3469.
- [12] Padala M, Damle S, Gujar S. Federated learning meets fairness and differential privacy. In: Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part VI 28, Springer, 2021, 692–699.
- [13] Kingma DP and Welling M. Auto-encoding variational bayes. *Statistics* 2014; **1050**: 1.
- [14] Li T, Sahu AK, Zaheer M, et al. Federated optimization in heterogeneous networks. *Proc Mach Learn Syst* 2020; **2**: 429–450.
- [15] Kairouz P, McMahan HB, Avent B, et al. Advances and open problems in federated learning. *Found Trends Mach Learn* 2021; **14**: 1–210.
- [16] Smith V, Chiang CK, Sanjabi M, et al. Federated multi-task learning. *Adv Neur Inf Process Syst* 2017; 30.
- [17] Bonawitz K, Ivanov V, Kreuter B, et al. Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, 1175–1191.
- [18] Geyer RC, Klein T and Nabi M. Differentially private federated learning: A client level perspective, arXiv preprint [arXiv:1712.07557](https://arxiv.org/abs/1712.07557), 2017.
- [19] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, 308–318.
- [20] Ma Z, Liu Y, Miao Y, et al. Flgan: Gan-based unbiased federated learning under non-IID settings. *IEEE Trans Knowl Data Eng* 2023.
- [21] Chunyong YIN and Rui QU. Federated learning algorithm based on personalized differential privacy. *J Comput Appl* 2023; **43**: 1160.
- [22] Wei W and Liu L. Gradient leakage attack resilient deep learning. *IEEE Trans Inf Forens Secur* 2021; **17**: 303–316.
- [23] Chakraborty A, Alam M, Dey V, et al. Adversarial attacks and defences: A survey, arXiv preprint [arXiv:1810.00069](https://arxiv.org/abs/1810.00069), 2018.
- [24] Zhang R, Guo S, Wang J, et al. A survey on gradient inversion: Attacks, defenses and future directions. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, 2023, 5678–685.
- [25] Liu X, Xie L, Wang Y, et al. Privacy and security issues in deep learning: A survey. *IEEE Access* 2020; **9**: 4566–4593.
- [26] Zhao B, Mopuri KR and Bilen H. iDLG: Improved deep leakage from gradients, arXiv preprint [arXiv:2001.02610](https://arxiv.org/abs/2001.02610), 2020.
- [27] Geiping J, Bauermeister H, Dröge H, et al. Inverting gradients-how easy is it to break privacy in federated learning? *Adv Neural Inf Process Syst* 2020; **33**: 16937–16947.
- [28] Ren H, Deng J and Xie X. GRNN: Generative regression neural network—a data leakage attack for federated learning. *ACM Trans Intell Syst Technol (TIST)* 2022; **13**: 1–24.
- [29] Yang H, Ge M, Xue D, et al. Gradient leakage attacks in federated learning: Research frontiers, taxonomy and future directions. *IEEE Netw* 2023; 1–8.
- [30] Wu Y, Kang Y, Luo J, et al. Fedcg: Leverage conditional gan for protecting privacy and maintaining competitive performance in federated learning. In: International Joint Conference on Artificial Intelligence, 2022, 2334–2340.
- [31] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. *Adv Neur Inf Process Syst* 2014; 27.
- [32] Yang H, Ge M, Xiang K, et al. Fedvae: Communication-efficient federated learning with non-IID private data. *IEEE Syst J* 2023.
- [33] Polato M. Federated variational autoencoder for collaborative filtering. In: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, 1–8.
- [34] Jiang Y, Wu Y, Zhang S, et al. Fedvae: Trajectory privacy preserving based on federated variational autoencoder. In: 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall), IEEE, 2023, 1–7.
- [35] Yu Z, Lu Y and Suri N. Raff: A robust and adaptive federated meta-learning framework against adversaries. In: 2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS), IEEE, 2023, 496–504.
- [36] Wang Z, Fan X, Wang Z, et al. Fedave: Adaptive data value evaluation framework for collaborative fairness in federated learning. *Neurocomputing* 2024; **574**: 127227.
- [37] Huong TT, Bac TP, Ha KN, et al. Federated learning-based explainable anomaly detection for industrial control systems. *IEEE Access* 2022; **10**: 53854–53872.
- [38] Cui S, Pan W, Liang J, et al. Addressing algorithmic disparity and performance inconsistency in federated learning. *Adv Neural Inf Process Syst* 2021; **34**: 26091–26102.
- [39] LeCun Y, Bottou L and Bengio Y. Gradient-based learning applied to document recognition. *Proc IEEE* 1998; **86**: 2278–2324.
- [40] Xiao H, Rasul K and Vollgraf R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint [arXiv:1708.07747](https://arxiv.org/abs/1708.07747), 2017.
- [41] Krizhevsky A and Hinton G. Learning Multiple Layers of Features from Tiny Images, University of Toronto: Toronto, 2009.
- [42] Peng X, Bai Q, Xia X, et al. Moment matching for multi-source domain adaptation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 1406–1415.
- [43] Gong B, Shi Y, Sha F, et al. Geodesic flow kernel for unsupervised domain adaptation. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, 2066–2073.

- [44] Lin T, Kong L, Stich SU, et al. Ensemble distillation for robust model fusion in federated learning. *Adv Neural Inf Process Syst* 2020; **33**: 2351–2363.
- [45] Gu H, Fan L, Li B, et al. Federated deep learning with bayesian privacy, arXiv preprint [arXiv:2109.13012](https://arxiv.org/abs/2109.13012), 2021.
- [46] Zhu Z, Hong J and Zhou J. Data-free knowledge distillation for heterogeneous federated learning. In: *International Conference on Machine Learning*, PMLR, 2021, 12878–12889.



Zhixin Li graduated from the School of Computer Science at Fudan University, China, with a master's degree in engineering. Currently, he is pursuing a doctoral degree at the School of Computer Science at Fudan University. His current research interests include fintech and security, federated learning, and AI security.



Yicun Liu is currently an undergraduate student in the School of Computer Science at Fudan University, China. His research interests include fintech and security, machine learning and AI security.



Jiale Li graduated from the School of Software, Dalian University of Technology, China. He is currently a master's student at Fudan University, China. His research interests include fintech, natural language processing, and federated learning.



Guangan Ye is currently a Researcher and doctoral supervisor at the School of Computer Science at Fudan University, Institute of FinTech at Fudan University, China. He received a Ph.D. degree from Columbia University in the United States. His main research areas include financial technology, graph model algorithms, multimodal feature fusion, and computer vision.



Hongfeng Chai is an Academician of the Chinese Academy of Engineering and an expert in Financial Information Engineering Management. He currently serves as the Dean of the Institute of Financial Technology at Fudan University, China, and as a Professor and doctoral supervisor in the School of Computer Science and Technology at Fudan University, China. His primary research interests include financial information engineering and security in financial technology.



Zhihui Lu received his Ph.D. degree in computer science from Fudan University, China, in 2004, where he is now a Professor in the School of Computer Science. His research interests include cloud computing and service computing technologies, blockchain, big data architecture, edge computing, and IoT distributed systems.



Jie Wu received a Ph.D. degree in computer science from Fudan University, China, in 2008. He is currently a Professor at the School of Computer Science, Fudan University, China. His research interests include internet technology, big data architecture, edge computing, cloud computing, and blockchain distributed systems.