PRIV-HFL: Privacy-Preserving and Robust Federated Learning for Heterogeneous Clients Against Data Reconstruction Attacks

Mohammadreza Najafi[1], Hooman Alavizadeh[2], Ahmad Salehi Shahraki[3],
A.S.M Kayes[4], Wenny Rahayu[5]
[1–5]Department of Computer Science & Information Technology, La Trobe University, Melbourne, Australia
[1]m.najafi@latrobe.edu.au, [2]H.Alavizadeh@latrobe.edu.au, [3]A.SalehiShahraki@latrobe.edu.au,
[4]A.Kayes@latrobe.edu.au, [5]W.Rahayu@latrobe.edu.au

Abstract*—Federated Learning (FL) is a machine learning paradigm that allows multiple local clients to collaboratively train a global model by sharing their model parameters instead of private data, thereby mitigating privacy leakage. However, recent studies have shown that gradient-based Data Reconstruction Attack (DRA) can still expose private information by exploiting model parameters from local clients. Existing privacy-preserving FL strategies provide some defense against these attacks, but at the cost of significantly reduced model accuracy. Moreover, the issue of client heterogeneity, particularly in Non-Identical and Independent Distributions (Non-IID) clients, further exacerbates these FL methods, resulting in drifted global models, slower convergence, and decreased performance. This study aims to address the two main challenges of FL: Non-IID data and client privacy through DRA. To this end, it leverages the lagrangian duality approach and incorporates a generator model to enable Knowledge Distillation (KD) among clients. By facilitating improved local model performance through inter-client knowledge transfer, the proposed method aims to simultaneously address the practical challenges commonly encountered by FL systems. Our study demonstrates a remarkable improvement in model accuracy, with KD boosting it by up to 15% on CIFAR-10 and MNIST classification tasks in Non-IID client settings. Furthermore, we propose an aggregation algorithm that inherently preserves client data privacy during the training phase, offering resilience against DRA.

***Index Terms*—Data Reconstruction Attack, Federated Learning, Data Heterogeneity, Dual-Decomposition, Generative Adversarial Networks**

I. Introduction

With the advancement of wireless communication networks that enable the Internet of Things (IoT), edge clients are now equipped with increasingly sophisticated sensing, communication, computing, and analysis capabilities [1]. To address data-driven learning tasks arising from various real-world problems, Artificial Intelligence (AI), particularly deep learning and deep neural networks [2], has been widely adopted for feature extraction and model building to enhance decision-making. This approach is expected to benefit numerous application fields, including smart healthcare [3], smart manufacturing [4], smart cities [5], and smart agriculture [6]. In the traditional deep learning framework, large volumes of raw data collected from edge clients are directly transmitted to a central server to develop comprehensive deep learning models. However, this data often contains sensitive information (e.g., healthcare records, private photos), posing significant privacy risks. Consequently, ensuring data security and privacy is one of the most critical concerns in this context.

Federated Learning (FL) [7] is an approach to enhance privacy by transmitting gradients or model parameters rather than raw data. This framework exchanges parameters between the server and each client, enabling the server to learn from the client's private data without directly accessing it. According to the data processing inequality [8], using parameters that deterministically map local data reduces the risk of compromising data privacy compared to sharing raw data. However, recent research has demonstrated that even parameter-based communication can still lead to privacy breaches. For instance, multiple studies have shown that adversaries can perform Membership Inference Attacks (MIA) using the transmitted model parameters [9], [10], [11], thereby compromising data privacy and anonymity. Additionally, attackers can fully reconstruct training data through Data Reconstruction Attacks (DRA) [12], [13], [14], [15], [16], posing significant privacy threats to FL systems. Several innovative privacy-preserving techniques have been proposed to mitigate these risks. For example, [17] introduces an FL scheme that combines quality-based aggregation with an extended Dynamic Contribution Broadcast Encryption (DConBE) and local Differential Privacy (DP). [18] proposes an adaptive FL framework that integrates adaptive gradient descent with DP mechanisms. To address the dual challenges of ensuring provable privacy guarantees while minimising communication and computational overheads in FL, especially for linear regression models, [19] suggests a differential private FL scheme. [20] presents the noising before model aggregation FL framework, which enhances DP by adding artificial noise to parameters at the client's side before aggregation.

In addition to privacy concerns, a significant challenge in FL is data heterogeneity, where clients' data exhibit Non-Identical and Independent Distributions (Non-IID). Non-IID data refer to substantial variations across clients in terms of statistical properties, such as data distributions, feature representations, and class imbalances. This heterogeneity complicates model training, as models trained on Non-IID data may struggle to generalize effectively across different clients or accurately represent the overall patterns within the entire dataset. Conventional FL algorithms, such as FedAvg [21], have been observed to result in divergent local models and a considerable loss of global knowledge under Non-IID conditions, leading to decreased performance and slower convergence [22], [23]. To address this issue, the concept of Knowledge Distillation (KD), as introduced by [24], offers a potential solution by transferring knowledge from teacher (client) models to a student (global) model. For instance, FedDF [25] employs ensemble distillation for model fusion, where the global model is trained using the averaged logits from local models. FedAUX [26] enhances this approach by determining model initialization for local models and assigning weights to local model logits using (ϵ, δ)-differential private certainty scoring. FedBE [27] takes a Bayesian approach, generating multiple global models from local models and then merging them into a single global model through ensemble KD. FedDiff [28], utilizes an attention-based diffusion model as a generative component. The pseudo-data generated by this model are then used to fine-tune the global model, thereby facilitating effective knowledge transfer across statistically heterogeneous clients and demonstrating significant performance improvements. Similarly, FedFTG [29] employs a Generative Adversarial Network (GAN) to produce synthetic data, which enhances the representativeness of local training data and supports better convergence in Non-IID settings. While these generative and ensemble-based methods notably improve convergence and generalization, they often assume access to client-generated logits or rely on privacy guarantees that are not robust enough, leaving them vulnerable to sophisticated data reconstruction attacks.

Beyond heterogeneity, a fundamental challenge in FL lies in the inherent trade-off between privacy preservation and model utility. Specifically, applying DP mechanisms to local models often introduces noise that can significantly degrade the quality of shared information, thereby hindering effective KD [30], [31], [32]. Consequently, improving accuracy typically necessitates relaxed privacy constraints [33], creating a difficult balance to strike. Combining privacy preservation with effective KD remains a significant challenge in FL. In particular, applying DP to local models introduces noise that can degrade the quality of shared information, making effective KD more difficult [30], [31]. As a result, there is often a trade-off between privacy and performance, while improving accuracy generally requires relaxed privacy constraints [33], [34], [35]. To address this critical trade-off and overcome the limitations of existing approaches, we propose PRIV-HFL, a privacy-preserving FL framework designed for heterogeneous clients. Rather than introducing entirely new algorithms, PRIV-HFL strategically integrates two established techniques: dual-decomposition optimization and GAN-based KD, into a unified framework. This integration allows PRIV-HFL to simultaneously mitigate privacy risks and improve model performance under Non-IID conditions. Unlike methods such as FedFTG and other generative KD approaches, PRIV-HFL explicitly enforces privacy constraints during local training through Lagrangian dual variables, ensuring stronger privacy protection. This is further bolstered by its use of GAN generated pseudo-data, which facilitates effective knowledge transfer at the server side without requiring direct access to potentially noisy client outputs or the transmission of model logits. By effectively decoupling privacy-preserving local training from global model distillation, PRIV-HFL mitigates the utility loss typically seen in standard DP-based FL methods, thereby achieving stronger privacy guarantees alongside improved performance in heterogeneous environments and minimizing the risk of private data leakage. The primary contributions of this work are outlined as follows:

- We present PRIV-HFL, a privacy-preserving FL framework that thoughtfully combines dual-decomposition and KD to enhance model utility while safeguarding client privacy under Non-IID settings.
- We present an innovative method for enhancing client model performance through data-free KD.
- We propose a novel KD mechanism using a modified GAN that generates pseudo-data for effective cross client knowledge transfer, mitigating the adverse effects of privacy induced noise in Non-IID settings.
- We incorporate a dual-decomposition optimization technique to enforce privacy constraints at the client level, while enabling utility-preserving knowledge distillation at the server, balancing the trade-off between privacy and performance.

The rest of the paper is organized as follows. In Section II, we provide a detailed overview of the preliminary concepts and methodologies that form the foundation of our proposed approach. Section III describes the methodology of the proposed method, outlining its design and key components. In Section IV, we present the experimental results, implementation, comparing the performance of our approach with state-of-the-art methods to demonstrate its effectiveness. Finally, Section V discusses the existing challenges and outlines potential directions for future research.

II. Preliminaries

In this paper, our methodology tackles Non-IID challenges by distilling knowledge between clients and employing dual-decomposition to preserve client parameter privacy against DRA. This section elucidates the characteristics of the methods utilized in the development of the PRIV-HFL approach.

A. Dual-decomposition Optimization

Dual-decomposition optimization, as proposed in [36], is a privacy-preserving optimization technique designed to address multi-constraint problems. In FL, the set of active clients is denoted as A where $|A| = N$, and each client, $i \in A$, has access to its local private dataset, D_i. The goal for each client is to determine the optimal weight vector $\theta_i \in \mathcal{R}^m$, solving the following optimization problem:

$$\underset{\theta_i}{min} \sum_{i=1}^{N} \mathcal{L}_{X \sim D_i}(\sigma(C(\theta_i, X_j))). \tag{1}$$

In this context, C denotes the classifier, σ represents the softmax activation function, and $\mathcal{L}$ refers to the loss function employed for computing the error of the model. Here, θ_i denotes the model parameters for the client i that is used to generate the prediction score for (X_j, Y_j), where X_j is the input data and Y_j is its ground-truth label, with $(X_j, Y_j) \in D_i$. For utilizing the dual-decomposition, the Eq. (1) is reformulated into an equivalent constrained optimization problem that mathematically shares a common optimal value which each client aims to solve the following problem:

$$\begin{gathered} \underset{\theta_i}{min} \sum_{i=1}^{N} \mathcal{L}_{X \sim D_i}(\sigma(C(\theta_i, X_j)))., \\ \text{subject to } \theta_i = \theta_q, \forall i, q \in A \end{gathered} \tag{2}$$

Eq. (2) represents that each client weight vector, $\theta_i \quad i \in A$, is required to be equal to other clients weight vector, $\theta_q \quad q \in A$, at the end of the training processes. However, it can be noticed this constraint can be replaced with pairs of double-sided inequality constraints, and Eq. (2) can be rewritten as a standard constrained distributed optimization problem within equality constraints as follows:

$$\begin{gathered} \underset{\theta_i}{min} \sum_{i=1}^{N} \mathcal{L}_{X \sim D_i}(\sigma(C(\theta_i, X_j)))., \\ \text{subject to } \sum_{i=1}^{N} e_i(\theta_i) \leq 0, \end{gathered} \tag{3}$$

where $e_i(\cdot) \in \mathcal{R}^m \to \mathcal{R}^{2m(N-1)}$ is a mapping function that encodes the equality constraint in Eq. (2). It can be designed in infinitely many ways to represent this constraint. For instance, Eq. (4) provides a specific formulation to calculate the output vector $e_i(\theta_i)$, where$e_i(\theta_i)$ corresponds to the i^{th} column of the matrix E, as explicitly defined in the equation.

$$E = [e_1(\theta_1), e_2(\theta_2), \cdots, e_N(\theta_N)]_{2m(N-1)\times N} = \begin{bmatrix} \theta_1 & -\theta_2 & [0] & \ldots & [0] & [0] \\ -\theta_1 & \theta_2 & [0] & \ldots & [0] & [0] \\ [0] & \theta_2 & -\theta_3 & \cdot & [0] & [0] \\ [0] & -\theta_2 & \theta_3 & \cdot & [0] & [0] \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ [0] & [0] & [0] & \ldots & \theta_{N-1} & -\theta_N \\ [0] & [0] & [0] & \ldots & -\theta_{N-1} & \theta_N \end{bmatrix} \tag{4}$$

The E matrix represents one of the infinitely many possible realizations that can satisfy the constraint in Eq. (3). In Eq. (4), $[\mathbf{0}]$ denotes the zero vector of size m. Considering Eq. (3) and the lagrangian multipliers vector with zero initial value, denoted as λ_i for client i, along with the ν_i vector calculated as $\nu_i = m \cdot e_i(\theta_i)$, the training process can be executed as follows:

1) Clients randomly initialize $\theta_i \in \mathcal{R}^m$.
2) Clients initialize $\lambda_i \in \mathcal{R}^{2m(N-1)}$ with zero value and $\nu_i \in \mathcal{R}^{2m(N-1)}$ with $m \cdot e_i(\theta_i)$ value and send them to the server.
3) The $\overline{\lambda}$ and $\overline{\nu}$ are calculated in the server from Eq. (5) and (6). These values are transferred to the clients in each communication round.

$$\overline{\lambda} \leftarrow \frac{1}{N}. \left(\sum_{i=1}^{N} \lambda_i \right). \tag{5}$$

$$\overline{\nu} \leftarrow \frac{1}{N}. \left(\sum_{i=1}^{N} \nu_i \right). \tag{6}$$

4) Each client calculates its loss value:

$$f_{ci} \leftarrow \mathcal{L}_{X \sim D_i}(\sigma(C(\theta_i, X)))). \tag{7}$$

5) Each client fine-tunes its gradient of loss value as below:

$$u_i \leftarrow \triangledown f_{ci} + \triangledown e_i(\theta_i)^T \overline{\lambda}. \tag{8}$$

6) Each client updates its weight with learning rate α_c as below:

$$\theta_{update} \leftarrow \theta_i - \alpha_c \cdot u_i. \tag{9}$$

7) Each client updates its λ_i and ν_i with hyperparameter, σ, form Eq. (10)–(12), and transfer to the server.

$$q_i \leftarrow \overline{\nu} - \sigma.\alpha_c.\overline{\lambda}. \tag{10}$$

$$\lambda_i \leftarrow [\overline{\lambda} + \alpha_c.q_i]. \tag{11}$$

$$\nu_i \leftarrow \overline{\nu} + m(e_i(\theta_{update}) - e_i(\theta_i). \tag{12}$$

8) Repeat steps 3 to 7 till the end of the training phase.

Assumption 1. For all $i \in A$, we assume that the $e_i(\cdot)$ are not disclosed to the server. In other words, the server is unaware of the exact realization of $e_i(\cdot)$ for all $i \in A$.

B. Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by [37], are a class of deep learning models designed to generate new data samples that resemble a given training dataset. GANs have gained significant attention in privacy-preserving data generation due to their ability to generate pseudo-data that preserve the statistical properties of the original data without revealing sensitive information. This pseudo-data can then be used for the training phase without compromising privacy. A GAN consists of two neural networks, namely a Generator $\mathcal{G}$ with a set of model parameters denoted as ω and a Discriminator $\mathcal{D}$, which are trained simultaneously. Through an adversarial process, the generator takes a random noise vector z sampled from a prior distribution, which typically is a uniform or Gaussian distribution, and the desire label Y as input. The generator maps this noise vector to the data space, aiming to produce pseudo-data indistinguishable from real data. As illustrated in Fig. 1, the discriminator receives either a real data sample X with label Y or a generated sample x' as input and outputs a probability representing the likelihood that the input is real rather than generated.

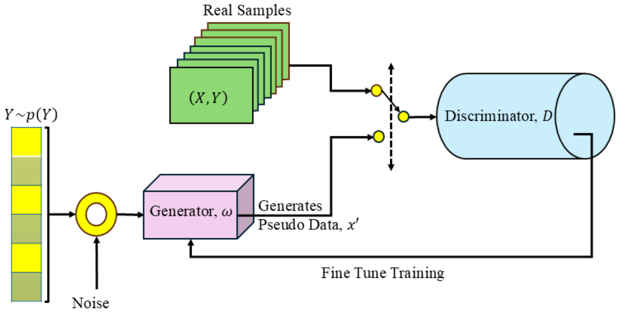


Fig. 1: The GAN model training procedure architecture.

The GAN framework is mathematically formulated as a minimax optimization problem, where the objective is to find an equilibrium between the generator and the discriminator [37]. The objective function for GANs can be expressed as Eq. (13) where $z \sim \mathcal{N}(0,1)$ and $Y \sim p(Y)$:

$$\underset{\omega}{min}\,\underset{\mathcal{D}}{max}\,\mathbb{E}_x[log\,\mathcal{D}(X)]+\mathbb{E}_{z,Y}[1-log\,\mathcal{D}(\mathcal{G}(\omega,z,Y))]. \quad (13)$$

The generator aims to minimize the loss function by generating samples that maximize the discriminator's error, while the discriminator tries to maximize the loss function by correctly classifying real and generated samples.

C. Data Reconstruction Attack

DRA is a significant privacy threat in FL and distributed machine learning settings, where an adversary attempts to reconstruct private training data from shared gradients or model updates. Since gradient updates contain implicit information about the training data, attackers can exploit them to infer sensitive information. DRA has gained attention due to its implications for data confidentiality, particularly in scenarios where privacy is a primary concern, such as healthcare and finance. DRA can be formally characterized as an optimization problem [38], which the adversary seeks to reconstruct a private input sample $X_j \in D_i$ from client i by minimizing the discrepancy between the model outputs for the original data and a reconstructed candidate. In typical FL settings, the local model parameters θ_i^T, transmitted by the client to the central server at communication round T, are observable. Given the sequential transmission of updated model parameters over communication rounds, the parameter update can be derived from Eq. (14).

$$\Delta\theta_i = \theta_i^T - \theta_i^{T-1}. \quad (14)$$

Assuming that the model is trained using first-order optimization methods such as stochastic gradient descent (SGD), the adversary can approximate the gradient through Eq. (15).

$$\Delta\theta_i \approx -\eta\nabla_{\theta_i}\mathcal{L}(X_j;\theta_i). \quad (15)$$

where η is the learning rate. This approximation allows the adversary to estimate the gradient information indirectly. Consequently, the reconstruction of a synthetic input sample x_{rec} is formulated as the Eq. (16) optimization problem.

$$x_{rec} = \arg\min_{x_{rec}} \left\|\Delta\theta_i + \eta\nabla_{\theta_i}\mathcal{L}(x_{rec};\theta_i)\right\|^2. \quad (16)$$

Alternatively, in model inversion attacks [39] or Data Leakage from Gradient (DLG)-style approaches [40], the adversary performs forward simulations and optimizes the synthetic input by minimizing the divergence between the model's predicted output, as obtained in Eq. (17).

$$x_{rec} = \arg\min_{x_{rec}} \|f_{\theta_i}(x_{rec}) - \hat{y}\|^2. \quad (17)$$

Where $f_{\theta_i}(x_{rec})$ represents the prediction of the model parameterized by θ_i when provided with an input x_{rec}.

Attackers employ this proposed optimization-based approach to iteratively refine x_{rec} until they closely resemble the original data, X_j. One of the most well-known types of DRA is DLG attack [30]. DLG exploits the fact that gradients encode information about the input data and iteratively reconstructs training samples from shared gradients by solving an optimization problem. The approach introduced in [41] further illustrates that despite the incorporation of DP mechanisms, certain attacks still remain able to reconstruct high-fidelity approximations of sensitive input data.

III. Methodology

In this section, we define the PRIV-HFL training phase with Non-IID clients. Fig. 2 represents our proposed model architecture. Based on the proposed method, each client initializes two models: a local model for the FL objective task, denoted as $\theta \in \mathcal{R}^m$, and a generator model, denoted as $\mathcal{G}$ with model parameters set ω, aimed at enhancing local model performance under Non-IID conditions.

During each communication round, clients employ a local optimizer loss function, $\mathcal{L}$, to train their local model on their private dataset, D_i. Furthermore, the local model at each client is refined using pseudo-data generated by $\mathcal{G}$. Unlike conventional FL methods, where clients transfer their local model gradients or parameters, θ_i, for aggregation, PRIV-HFL clients employ the modified dual-decomposition optimization method, as explained in the section II-A. They transfer only λ_i and ν_i vectors. As a result, PRIV-HFL protects clients from sharing their local model parameters, thereby defending them against DRA. The PRIV-HFL client-side and server-side processes are further explained in the subsequent subsections.

A. Client-Side Procedure in the PRIV-HFL

In PRIV-HFL, clients utilize a decomposition-based optimization approach, as described in Subsection II-A, to enable secure aggregation while preserving data privacy against DRA.

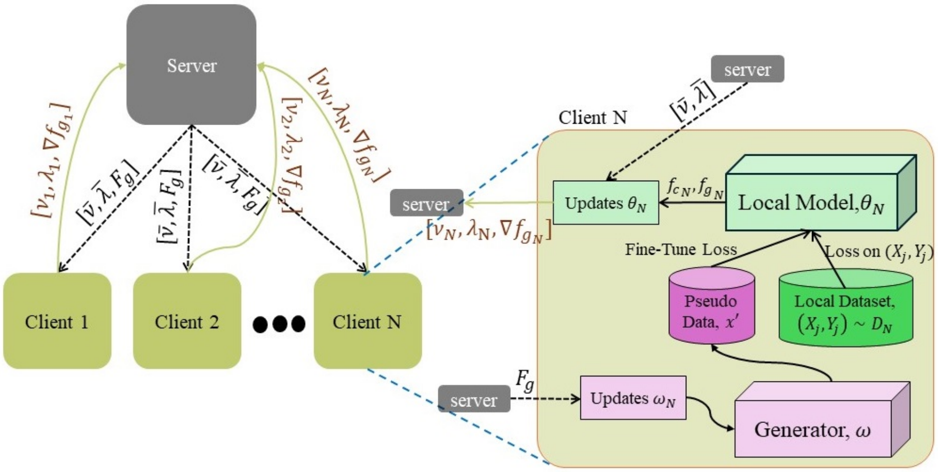

Server
$[\nu_1, \lambda_1, \nabla f_{g_1}]$
$[\bar{\nu}, \bar{\lambda}, F_g]$
$[\bar{\nu}, \bar{\lambda}, F_g]$
$[\bar{\nu}, \bar{\lambda}, F_g]$
$[\nu_2, \lambda_2, \nabla f_{g_2}]$
$[\nu_N, \lambda_N, \nabla f_{g_N}]$
Client 1
Client 2
Client N
server
$[\bar{\nu}, \bar{\lambda}]$
Client N
Updates θ_N
f_{c_N}, f_{g_N}
Local Model, θ_N
server
$[\nu_N, \lambda_N, \nabla f_{g_N}]$
Fine-Tune Loss
Loss on (X_j, Y_j)
Pseudo Data, x'
Local Dataset, $(X_j, Y_j) \sim D_N$
server
F_g
Updates ω_N
Generator, ω

Fig. 2: The proposed PRIV-HFL model architecture.

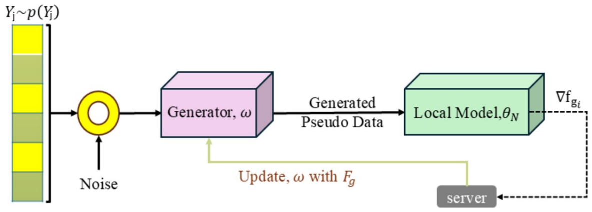


Fig. 3: The modified GAN training procedure.

Furthermore, to mitigate the challenges associated with non-IID client distributions, PRIV-HFL integrates a generative model $\mathcal{G}$ based on a GAN architecture. This model synthesizes pseudo-data, facilitating knowledge transfer across clients in a data-free KD, thereby improving overall performance.

During the training phase, $\mathcal{G}$ gets noise z, and Y_j the ground-truth label for generating pseudo-data, x', as described in Eq. (18).

$$x' = \mathcal{G}(z, Y_j; \omega_i). \tag{18}$$

The ω_i is the $\mathcal{G}$ model parameter for the client i, z is Gaussian noise $z \sim \mathcal{N}(0,1)$ and $Y_j \sim p(Y_j)$. Let ${n^i}_{Y_j}$ denote the number of real data samples $(X_j, Y_j) \in D_i$ within client i's local dataset that are associated with label Y_j. The probability of generating pseudo-data with label Y_j, denoted as $p(Y_j)$, is computed based on Eq. (19). Although this probability distribution is derived from local class statistics, all computations remain on the client-side and are not shared with the server to preserve data privacy.

$$p(Y_j) \propto \sum_{j=1}^{{n^i}_{Y_j}} \mathbb{E}_{(X_j,Y_j)\sim\mathcal{D}_i}\left[1_{Y_j=Y}\right] = R{n^i}_{Y_j}. \tag{19}$$

Where the indicator function $1_{\text{condition}}$ evaluates to 1 when the specified condition is satisfied and 0 otherwise. Furthermore, as defined in Eq. (19), the term $Rn^i_{Y_j}$ denotes a randomly sampled variable associated with each label Y_j, deliberately chosen to be substantially larger than the corresponding data count $n^i_{Y_j}$. This scaling strategy increases the relative sampling probability of minority classes, thereby compensating for local class imbalance and encouraging a more balanced label distribution in the generated pseudo-data. As a result, this mechanism facilitates more effective KD and mitigates the negative impact of data heterogeneity across clients in FL.

The pseudo-data x' generated by $\mathcal{G}$ is employed to fine-tune the client loss function, f_{ci}. By evaluating the loss of the local model, θ_i, on these pseudo-data, F_{g_i}, as detailed in Eq. (20), the performance loss induced by decomposition optimization is mitigated. Moreover, the $p(Y_j)$ distribution compels ω_i to generate pseudo-data that guide the client towards a more uniform data distribution for training. This approach mitigates the drift during aggregation caused by the Non-IID challenge, as demonstrated by the experimental results in the following section.

$$f_{ci} \leftarrow \mathcal{L}_{X_j\sim D_i}(\sigma(C(\theta_i, X_j)))) + \underbrace{\mathcal{L}(\sigma(C(\theta_i, x')}_{f_{g\,i}}. \tag{20}$$

After calculating the f_{ci}, the u_i is computed from Eq. (8) and then clients updates their model parameters, θ_i, with α_c learning rate from Eq. (9). In the end, based on the Eq. (10), (11), and (12), clients update the λ_i and ν_i for transferring them to the server for aggregation. PRIV-HFL addresses the privacy performance loss and Non-IID challenge by using ω_i during the training phase, but training the ω_i still remains a critical task. The $\mathcal{G}$ cannot use client's local private dataset, D_i, for training ω_i because in the Non-IID situation training ω_i on the clients private dataset leads to the generation of pseudo-data in a similar distribution as client has and it cannot help us to address the Non-IID. Therefore, for training the ω_i, PRIV-HFL modified the formal GAN architecture as illustrated in Fig. 3. In training phase of ω_i, PRIV-HFL remove Discriminator $\mathcal{D}$, Fig. 3, and utilize f_{g_i} loss value as described in Eq. (20). F_{g_i} calculates the loss of the local model on the pseudo-data generated by ω_i. Clients transfer the gradient of f_{g_i} to the server for aggregation and receive the F_g from the server to update ω_i with α_g learning rates as described in Eq. (21). The method for calculating F_g from $\triangledown f_{g_i}$ is detailed in Eq. (22) in the server-side subsection.

$$\omega_i \leftarrow \omega_i - \alpha_g . F_g. \tag{21}$$

Algorithm 1 PRIV-HFL Client-Side Procedure

Inputs: Average Lagrangian multiplier $\overline{\nu}$; Average dual variable $\overline{\lambda}$; Average generator gradient F_g; Client model learning rate α_c; Generator learning rate α_g; Noise scale σ.

```
repeat
    for all clients $i \in \{1, \dots, N\}$ in parallel do
        Receive $\overline{\nu}$, $\overline{\lambda}$, and $F_g$ from the server
        Update generator model parameters $\omega_i$ using Eq. (21)
        Sample latent vector and label:
            $z \sim \mathcal{N}(0, 1), \quad Y_j \sim p(Y_j)$
        Generate pseudo-data $x'$ using Eq. (18)
        Update generator loss $f_{g_i}$ using Eq. (20)
        Compute client loss $f_{c_i}$ using Eq. (20)
        Compute local update vector:
            $u_i \leftarrow \nabla f_{c_i} + \nabla e_i(\theta_i)^\top \overline{\lambda}$
        Update auxiliary variable $q_i$ using Eq. (10)
        Update model parameters:
            $\theta_{\text{update}} \leftarrow \theta_i - \alpha_c \cdot u_i$
        Update Lagrangian multipliers:
            $\lambda_i \leftarrow \text{update via Eq. (11)}, \quad \nu_i \leftarrow \text{update via Eq. (12)}$
        Set $\theta_i \leftarrow \theta_{\text{update}}$
        Send $\nu_i$, $\lambda_i$, and $\nabla f_{g_i}$ to the server
    end for
until convergence or stopping criteria is met
```

As depicted in Fig.2, during training, each client not only transmits the Lagrangian multipliers λ_i and ν_i to the server but also shares the gradient ∇f_g^i to support the update of its generator parameters ω_i. The client-side workflow of PRIV-HFL is detailed in Algorithm1. Each client starts by receiving the globally averaged variables $\overline{\nu}$, $\overline{\lambda}$, and the average generator gradient F_g from the server. With these inputs, the client updates its generator model ω_i and synthesizes pseudo-data x' using labels sampled from the distribution $p(Y_j)$, following Eq. 19. The client then updates its model parameters θ_i by minimizing a local loss computed over both real and pseudo-data, while incorporating the Lagrangian regularization via $\overline{\lambda}$. Lastly, each client updates its local multipliers λ_i and ν_i, and returns them along with ∇f_g^i to the server for use in the subsequent global aggregation step.

Remark.1 The *Assumption* 1 ensures that when executing Algorithm 1, the server is unable to reconstruct θ_i, thereby preventing access to information about the clients' private local dataset D_i. It is important to note that the $\triangledown f_{g_i}$ transferred between clients and the server represents the gradient of the loss function of the local model θ_i on the pseudo-data x' generated from Eq. (18). Consequently, state-of-the-art privacy attacks can only reconstruct information about x' (the pseudo-data) and not the clients' private data. Additionally, since $R_{nY} \gg {n^i}_Y$ in Eq. (19), the distribution of the private dataset D_i cannot be inferred from ∇f_{g_i}. This aspect will be elaborated upon in the subsequent section, where the PRIV-HFL is evaluated under scenarios involving gradient-based DRA. Therefore, Algorithm 1 strengthens the privacy of PRIV-HFL by safeguarding the clients' private data.

TABLE I: The state-of-the-art methods main features.

Model Name	**Local Model**	**KD**	**Generator**	**Privacy against DRA**
FedAvg	CNN	×	-	×
FedProx	CNN	×	-	×
SCAFFOLD	CNN	×	-	×
FedGen	ResNet18	✓	Autoencoder	×
FedFTG	ResNet18	✓	GAN	×
FedRand	CNN	×	-	✓
FedDiff	CNN	✓	Diffusion	×
FedKD	MLP	✓	No generator	✓
f-differential	CNN	×	-	✓
PRIV-HFL	ResNet18	✓	GAN	✓

Algorithm 2 PRIV-HFL Server-Side Procedure

Inputs: Number of communication rounds T; Clients' Lagrangian multipliers $\{\nu_i\}_{i=1}^N$; Clients' dual variables $\{\lambda_i\}_{i=1}^N$; Clients' generator gradients $\{\nabla f_{g,i}\}_{i=1}^N$.

```
for $t = 1, \dots, T$ do
    Receive $\nu_i$, $\lambda_i$, and $\nabla f_{g,i}$ from all clients $i \in \{1, \dots, N\}$.
    Compute the average dual variable:
        $\overline{\lambda} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \lambda_i \quad \text{(see Eq. (5))}$
    Compute the average Lagrangian multiplier:
        $\overline{\nu} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \nu_i \quad \text{(see Eq. (6))}$
    Compute the average generator loss gradient:
        $F_g \leftarrow \frac{1}{N} \sum_{i=1}^{N} \nabla f_{g,i} \quad \text{(see Eq. (22))}$
    Send $\overline{\lambda}$, $\overline{\nu}$, and $F_g$ to all clients.
end for
```

B. Server-Side Procedure in the PRIV-HFL

As mentioned above, unlike conventional FL methods, PRIV-HFL protects client DRA by avoiding the transfer of the client's local model parameters, θ_i, to the server. Consequently, there is no global model on the server-side for training or aggregation. Additionally, the server lacks any information about the clients' data distributions and the $e_i(\cdot)$ function used for decomposition optimization, as explained in *Remark.1* in subsection III-A, thereby ensuring the privacy of PRIV-HFL's clients. As it represented in Algorithm 2, Server received the

λ_i, ν_i and $\triangledown f_{g_i}$ from clients and processes to calculate the average of these parameters as formulated in Eq. (5), Eq. (6), and Eq. (22) and sends back the $\overline{\lambda}$, $\overline{\nu}$ and F_g to the clients for updates their models.

$$F_g \leftarrow \frac{1}{N} \cdot \left(\sum_{i=1}^{N} \triangledown f_{g_i} \right). \tag{22}$$

IV. Experiments

This section elucidates the implementation aspects of PRIV-HFL and provides a comparative analysis of its performance against state-of-the-art methods to address Non-IID challenges. Additionally, a comparison of its performance in situations involving DRA will be conducted in section IV-B.

A. Implementation Details

In this section, we conduct a comparative analysis to evaluate the performance of PRIV-HFL against other notable related works. Detailed implementation specifics and comprehensive experimental results are discussed in the following subsections. In addition, the Python source code for PRIV-HFL is accessible via the following link: https://github.com/mrezn/PRIV-HFL.

1) Reference Methods: In this study, we compare PRIV-HFL with several significant related works, including FedAvg [7], FedProx [42], SCAFFOLD [43], FedFTG [29], FedDiff [28], FedGen [44], FedRand [45], FedKD [46], and f-differential [47]. Table I outlines the features of these state-of-the-art methods and demonstrates that FedGen and FedFTG utilize KD to address the Non-IID challenge. FedFTG employs a classification model for the client's local model $\sigma(C(\theta_i))$, utilizing ResNet18 [48], and uses a GAN architecture [37] for its $\mathcal{G}$ model. Similarly, FedGen also employs ResNet18 as the local model and utilizes a simple Autoencoder model for its $\mathcal{G}$ architecture. Notably, FedAvg, FedProx, SCAFFOLD, FedGen, and FedFTG do not mention any techniques in their methodologies to ensure client privacy against DRA. Meanwhile, the FedKD local model employs a Multi-Layer Perceptron (MLP) architecture and utilizes the KD for performance improvement but does not employ any generator model.

2) PRIV-HFL Networks Architecture: PRIV-HFL, similar to FedFTG and FedGen, has a ResNet18 architecture for clients local model $\sigma(C(\theta_i))$. Additionally, the GAN proposed in [49] is employed as the $\mathcal{G}$ model.

3) Datasets: The effectiveness of PRIV-HFL is evaluated using the CIFAR-10 [50] and MNIST [51] datasets, which feature heterogeneous dataset partitions. To emulate Non-IID data distribution among clients, a practice adopted by previous studies [52], [53], [54], [29], we utilize the Dirichlet distribution Dir(ψ) on label ratios. Here, a smaller value of ψ signifies increased data heterogeneity. For PRIV-HFL implementation, and $\psi = 0.3, 0.6$.

4) Differential Privacy (DP): We further compare the performance of PRIV-HFL with several baseline methods under DP conditions, following the framework proposed by [55]. DP is widely recognized as an effective approach to mitigate DRA, making it a suitable benchmark for privacy-preserving FL. Given that our proposed method aims to jointly address both the Non-IID client distribution and privacy preservation challenges, we apply the DP mechanism to state-of-the-art methods specifically designed to handle Non-IID scenarios. In this comparative analysis, we incorporate Gaussian noise with varying privacy budgets (ϵ) into the clients' local model parameters to evaluate privacy protection against DRA under different levels of noise injection.

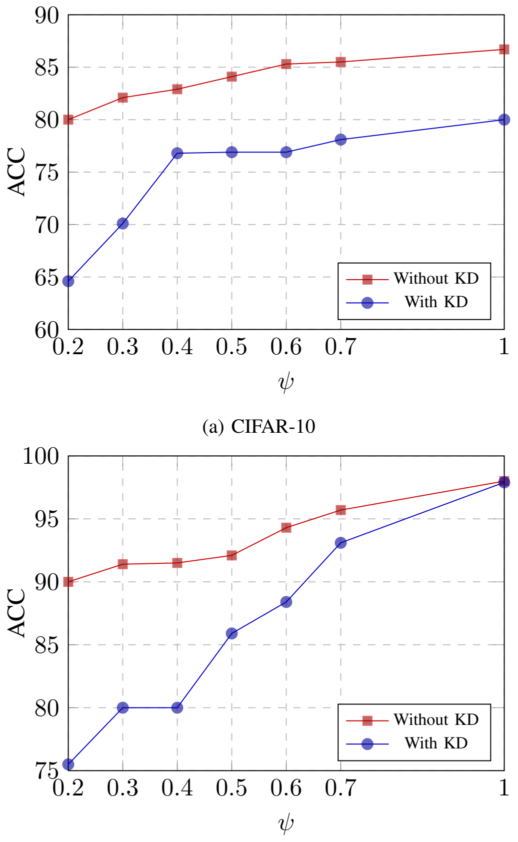
5) Hyperparameter: The number of clients $N = 100$, number of local training epochs 50, communication rounds $T = 1000$, for local training, the batch size is 50. Moreover the α_c and α_g are both initialized to 0.1 and σ is 0.3 . The dimension of X and x' are $32 \times 32 \times 3$ for CIFAR-10 and $28 \times 28 \times 1$ for MNIST.

B. Performance Comparison

In this section, we begin by evaluating the performance of the proposed PRIV-HFL framework in two configurations: one excluding the generative model $\mathcal{G}$, relying solely on the decomposition optimization strategy, and the other incorporating the full implementation with $\mathcal{G}$. This comparison underscores the critical role of the generative model and the effectiveness of KD, particularly in scenarios involving Non-IID client distributions. Subsequently, we present a comprehensive comparative analysis of PRIV-HFL's classification accuracy against several state-of-the-art baselines across four distinct experimental setups. In subsection IV-B2, we examine the case of Non-IID clients without any privacy-preserving mechanisms. Subsection IV-B3 addresses the scenario in which clients are IID and DP is employed to protect training data. Subsection IV-B4 extends this analysis to Non-IID clients under the same privacy constraints. Finally, in subsection IV-B5, we assess the privacy leakage of the proposed method under a targeted DRA, thereby evaluating the model's robustness against adversarial inference.

1) Comparison of PRIV-HFL Model Without Utilizing KD Method: We evaluate the impact of incorporating KD into PRIV-HFL by comparing its performance with and without the KD mechanism. The results, presented in Fig. 4, illustrate the accuracy trends across varying values of the heterogeneity parameter ψ for both CIFAR-10 and MNIST datasets. As observed, the KD-enhanced PRIV-HFL consistently outperforms the baseline (without KD), especially under higher heterogeneity conditions (lower ψ values). Notably, at $\psi = 0.2$, KD improves the accuracy by approximately 17% on CIFAR-10 and 12% on MNIST, highlighting its effectiveness in mitigating the performance degradation caused by non-IID data distributions in FL.

2) Performance Without Privacy Consideration: The overall accuracy comparison between PRIV-HFL and other state-of-the-art methods is presented in Table II for the CIFAR-10 and MNIST datasets across two ψ values ($\psi = 0.3$ and $\psi = 0.6$), with all experiments conducted using three random seeds. As shown in Table II, PRIV-HFL generally surpasses most comparison methods, with the notable exception of FedDiff.



(a) CIFAR-10

(b) MNIST

Fig. 4: Compare the accuracy, (ACC), of PRIV-HFL with and without KD on MNIST and CIFAR-10

TABLE II: The average test accuracy (%) of different FL methods on CIFAR-10 and MNIST.

Model Name	CIFAR-10		MNIST	
	$\psi = 0.6$	$\psi = 0.3$	$\psi = 0.6$	$\psi = 0.3$
FedAvg	82.04	79.59	93.84	90.16
FedProx	82.36	80.12	93.83	90.10
SCAFFOLD	84.55	82.14	97.14	95.94
FedGen	82.23	79.72	95.52	93.03
PRIV-HFL	85.78	82.254	94.07	91.81
FedFTG	86.06	84.38	98.91	97.01
FedDiff	**87.18**	**85.97**	**99.09**	**97.51**

FedDiff exhibits robust performance in FL scenarios with Non-IID clients, primarily due to its adoption of a diffusion model architecture as the generator module. However, as illustrated in Fig. 5 which presents a comparison of test accuracy across various ψ values, the accuracy of PRIV-HFL is comparable to FedDiff when the ψ value is close to 1, indicating clients are IID. While PRIV-HFL achieves competitive accuracy across a range of heterogeneity levels, it does not fully match the performance of FedDiff under non-private conditions, particularly for lower values of ψ where client data distributions are highly Non-IID. This gap can be primarily attributed to the privacy-centric design of PRIV-HFL, which avoids direct sharing of model parameters. Unlike FedDiff, which relies on a diffusion-based generator, PRIV-HFL decouples generator training from local updates to ensure robustness against DRA. By integrating a dual-decomposition optimization strategy, we enforce local privacy constraints without compromising the overall federated training protocol. However, this inherently limits the richness of information available for the KD, especially in highly heterogeneous settings. As a result, performance may slightly trail state-of-the-art non-private methods like FedDiff when privacy is not a consideration.

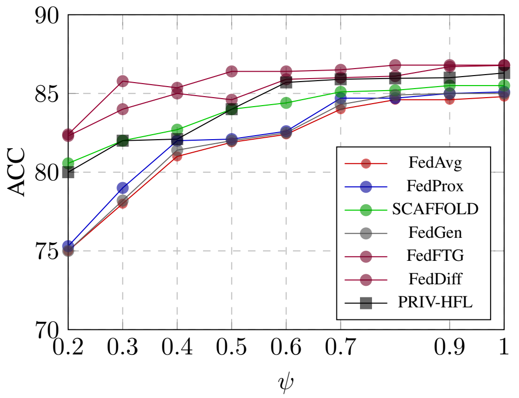


Fig. 5: Compare the accuracy, (ACC), of state-of-the-art methods on different ψ (data heterogeneity) on CIFAR-10.

Nonetheless, PRIV-HFL remains competitive, especially in moderate to near-IID scenarios ($\psi \geq 0.5$), while offering stronger protection against reconstruction and inference attacks. For ψ values less than 0.3, SCAFFOLD's accuracy exceeds that of PRIV-HFL. This is likely due to PRIV-HFL's approach of not transferring clients' local model parameters θ_i and its utilization of decomposition optimization for privacy considerations, which inherently constrains information flow more than some other methods. Despite this, PRIV-HFL's accuracy remains competitive with state-of-the-art methods, particularly for Non-IID clients in CIFAR-10 classification applications, and it shows minimal differences from FedFTG. In future work, we plan to explore hybrid techniques that could better adapt to high heterogeneity without sacrificing privacy, such as selective feature sharing or privacy-aware personalization modules.

3) Performance With Privacy Consideration: The strength of PRIV-HFL lies in its ability to ensure client privacy against DRA while maintaining performance in heterogeneous systems, as it mentioned in *Remark.1*III-A. DP is a well-known method utilized to address the privacy challenges of DRA in FL applications. Table III compares various state-of-the-art FL methods that employ DP techniques with PRIV-HFL, using different ϵ values on the MNIST dataset. As shown, for lower ϵ values, the accuracy of f-differential slightly surpasses PRIV-HFL by less than 1%. However, for higher ϵ values, PRIV-HFL outperforms all methods with significant differences and remains independent of ϵ values because it

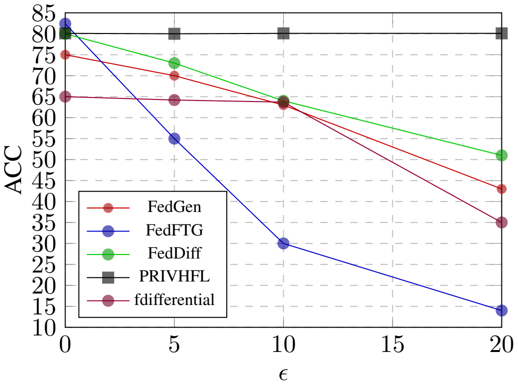


Fig. 6: Compare the accuracy, ACC, of state-of-the-art methods on different ϵ while $\psi = 0.5$ (Non-IID clients) on CIFAR-10.

maintains privacy without relying on DP techniques. Like PRIV-HFL, FedAKD employs the KD method to preserve accuracy, but its performance does not compare favorably to PRIV-HFL.

TABLE III: Comparison of various FL methods on MNIST with IID clients ($\psi = 0$) considering privacy with different ϵ.

ϵ	FedRand	f-differential	FedAvg	FedAKD	PRIV-HFL
0	97.1	**98.74**	78.12	60.15	98.71
5	96.2	**98.72**	78.16	60.14	98.71
10	94.3	98.55	70.15	60.16	**98.71**
20	34.5	90.11	46.15	20.48	**98.71**

Table IV compares state-of-the-art methods with accuracy levels similar to PRIV-HFL in MNIST classification, now applied to CIFAR-10 classification across varying ϵ values. This comparison underscores how PRIV-HFL performs relative to these leading methods in maintaining accuracy while addressing privacy concerns. Unlike in the MNIST classification, where PRIV-HFL's accuracy is comparable, in CIFAR-10 classification, PRIV-HFL's accuracy is notably higher than that of other methods, demonstrating significant performance improvements with greater ϵ.

TABLE IV: Comparison of various FL methods on CIFAR-10 with IID clients ($\psi = 0$) considering privacy for different ϵ).

ϵ	FedRand	f-differential	PRIV-HFL
0	64.20	64.72	**86.06**
5	60.00	62.55	**86.06**
10	42.20	59.61	**86.06**
20	22.40	34.12	**86.06**

4) Performance With Privacy Consideration and Non-IID Clients: PRIV-HFL effectively preserves client privacy against DRA while simultaneously addressing the challenges posed by Non-IID clients. As discussed in previous subsections, its performance in independent scenarios, Non-IID environments, and privacy protection against DRA is comparable with state-of-the-art methods. We now compare PRIV-HFL's performance with other leading methods that either outperform or match PRIV-HFL in the previously discussed scenarios. Figure 6 illustrates its accuracy in scenarios where both privacy must be maintained and clients are Non-IID, with a ψ value of 0.5 for CIFAR-10 classification. Based on *Remark 1*III-A, PRIV-HFL ensures privacy against DRA, whereas other methods utilize DP with varying ϵ values for comparison. As shown in Figure 6, PRIV-HFL demonstrates a significant advantage over other methods at ϵ values greater than zero, specially in compare to FedFTG that has better performance than PRIV-HFL when $\psi = 0.5$. However, its performance under DP is significantly reduced and less effective than PRIV-HFL, underscoring PRIV-HFL's strength for real-world F applications.

5) Privacy Analysis of PRIV-HFL under Deep Leakage Attack: We evaluate the privacy-preserving capability of PRIV-HFL when subjected to DLG attack [30], a well-known DRA attack. Specifically, we analyze the reconstructed data obtained from the DLG attack and compare it against the clients' private data using the Structural Similarity Index Measure (SSIM) as the performance metric. The formula for SSIM is derived from Eq. (23).

$$\text{SSIM}(int1, int2) = \frac{(2\mu_{int1}\mu_{int2} + C_1)(2\sigma_{(int1,int2)} + C_2)}{(\mu_{int1}^2 + \mu_{int2}^2 + C_1)(\sigma_{int1}^2 + \sigma_{int2}^2 + C_2)} \tag{23}$$

Where in this equation, μ_{int1} and μ_{int2} are the means of first input data, $int1$, and second input data, $int2$, σ_{int1}^2 and σ_{int2}^2 are the variances of $int1$ and $int2$, respectively. $\sigma_{(int1,int2)}$ is the covariance of $int1$ and $int2$. C_1 and C_2 are constants to stabilize the division when the denominator is small.

Prior to applying the deep leakage attack, we compute the divergence between the distribution of the pseudo-data, x', generated by the model, and the distribution of the clients' private data, D_i. This divergence is quantified using the Kullback-Leibler (KL) divergence, which is defined in Eq. (24).

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \tag{24}$$

Here, P represents the distribution of the pseudo-data x'. Q represents the distribution of the clients' private data D_i.

As illustrated in Fig. 7, the KL divergence which measures the discrepancy between two probability distributions, pseudo-data and the clients' private data, demonstrates a consistent downward trend across successive communication rounds for each CIFAR-10 label ID. This decline signifies the progressive refinement of the generator throughout the training process. Moreover, the KL divergence for all class labels undergoes a rapid decrease within the first 100 communication rounds, where values typically fall from above 100 to below 30, highlighting a fast convergence phase. An inflection point becomes evident around rounds 150 to 200, after which the rate of decrease diminishes, indicating a transition to a slower convergence regime. By approximately round 400, the divergence values begin to stabilize, ranging between 1 and 10 depending on the label ID. This stabilization suggests

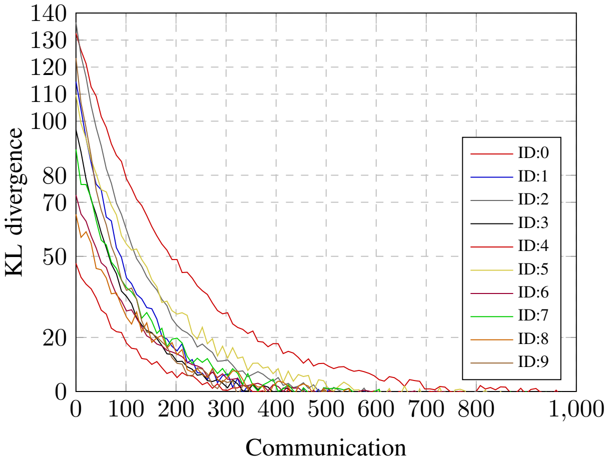


Fig. 7: The KL divergence between pseudo-data and clients private data in the each communication round for each CIFAR-10 class ID.

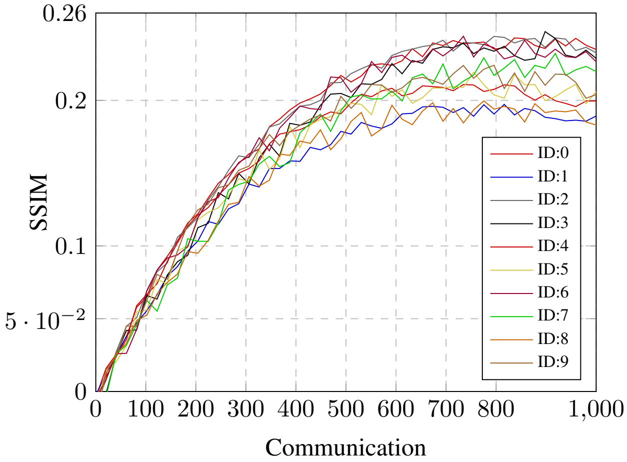


Fig. 8: The comparing the reconstructed data, x_{rec} and clients private data in the each communication round for each CIFAR-10 class ID.

that the generator has effectively learned the underlying data distribution, with the pseudo-data increasingly resembling the distribution of the clients' private data.

However, it is important to note that, while two images may exhibit similar KL divergence values, their pixel-wise comparison might reveal significant differences. This observation underscores that the KL divergence measures the disparity between distributions rather than the quality of the distributions. For instance, in the context of this study, where the inputs consist of RGB images labeled according to the CIFAR-10 dataset, two images with the same label may yield a high SSIM value while exhibiting minimal KL divergence.

From a privacy perspective, the SSIM value is critical, as reconstructed data (x_{rec}) derived from Eq. (23) should not exhibit a high SSIM value compared to the private data. This ensures that the reconstructed data remain dissimilar to the clients' original private data, thus preserving privacy.

As shown in Fig. 8, the SSIM between the clients' private data and the reconstructed data (x_{rec}) is presented for each class label ID in the CIFAR-10 dataset. The SSIM values for each label ID increase as the communication rounds progress, primarily due to the generator being trained during these rounds. As discussed in the previous paragraph, the pseudo-data distribution becomes increasingly similar to the clients' private data distribution, and ∇f_{g_i} gradually captures more information about the clients' private data as the communication rounds progress. Consequently, x_{rec} is able to reconstruct additional information about x' as described in Eq. (23).

However, when comparing x_{rec} with the clients' private data using SSIM as the performance metric, the SSIM values converge to less than 0.25 for the label ID 9. According to the findings in [56], which employed DP mechanisms to counter gradient-based DRAs in FL systems, maintaining the SSIM value between reconstructed data (x_{rec}) and clients' private data below 0.3 indicates that the proposed method effectively preserves client privacy against DRA.

V. Discussion

In this section, we discuss the strengths and limitations of PRIV-HFL across several key dimensions relevant to FL. Each subsection explores a distinct aspect, including privacy-utility, communication cost and applicability.

1) Privacy vs. Utility Trade-off: In FL, achieving a balance between privacy preservation and model utility is a persistent challenge. Traditional DP based FL methods introduce noise into model updates to ensure privacy, which often leads to substantial accuracy degradation, especially under strict privacy budgets. In contrast, PRIV-HFL avoids the direct transmission of gradients or model parameters, relying instead on pseudo-data generation and KD. As shown in our results, PRIV-HFL maintains SSIM values below 0.25, indicating strong privacy protection, while achieving less than 3% accuracy drop compared to the non-private baseline. This demonstrates PRIV-HFL's ability to better mitigate the privacy-utility trade-off than DP based approaches.

2) Effectiveness of Generative Models in Non-IID Settings: Statistical heterogeneity among clients, referred to as Non-IID data, is a well-known obstacle to convergence and generalization in FL systems. PRIV-HFL leverages a GAN based generative module that enables clients to synthesize data representations reflecting broader class distributions without requiring data sharing. This mechanism allows clients with limited or missing class data to still contribute effectively to the global model. Experimental results indicate up to 10% accuracy improvement over baseline FL approaches in highly Non-IID settings, underscoring the effectiveness of this approach.

3) Client Resource Analysis: One of the primary goals in the design of PRIV-HFL was to ensure its applicability to resource-constrained clients, such as IoT or mobile devices. In

PRIV-HFL, all fine-tuning of the global model and computation related to the GAN generator are performed exclusively on the central server. The client-side responsibilities are limited to a single phase of standard local model training using ResNet-18, followed by computation of local gradients and updates of Lagrangian parameters λ_i and ν_i. The Lagrangian multipliers λ_i and ν_i are vectors in $\mathbb{R}^{2m(N-1)}$, where m is related to the size of the local model (the number of parameters), and N is the number of participating clients. As the number of clients N increases, the dimensionality of λ_i and ν_i scales linearly with $(N-1)$, since each client must maintain dual variables corresponding to every other peer in the system. This linear growth introduces additional memory and computational overhead per client. In practical FL scenarios with $N = 100$ clients and a local model size of approximately $m \approx 11.7 \times 10^6$ (number of parameters in ResNet-18), the memory usage for each λ_i and ν_i vector has a size of $2m(N-1) = 2 \times 11.7\text{M} \times 99 \approx 2.1$ billion scalar values. When combined with the client's local model parameters, this leads to significantly higher storage requirements compared to conventional FL approaches, which typically store only the model weights. To address this, our future work will focus on reducing this storage burden through vector compression or approximation techniques.

Moreover, in addition to standard local training, each client in PRIV-HFL performs extra computation to update the Lagrangian multipliers λ_i and ν_i, which introduces additional Multiply-Accumulate (MAC) operations. Specifically, computing λ_i followed by vector operations involving $\overline{\nu}$, as outlined in Eq.(11). Similarly, updating ν_i requires computing the constraint function $e_i(\theta_i)$ and its interaction with $\overline{\lambda}$, as in Eq.(12). However, since these updates are based on fixed-size vectors proportional to the model dimensions and client count, their MAC cost remains predictable and can be optimized further through lightweight approximations or partial updates in future work.

4) Communication Cost: A notable limitation of PRIV-HFL is its communication overhead. Although it avoids transmitting full model parameters, it requires exchanging auxiliary optimization vectors (λ_i, ν_i, and ∇f_g^i), with the size of λ_i and ν_i growing proportionally to the number of clients N. For example, when $N = 100$ and using 32-bit floating-point representation, each client transmits approximately 7.82 GB per round that it is more than standard FL methods, which typically exchange only model weights (for ResNet-18 is around 0.04 GB). While this design strengthens privacy, it introduces scalability challenges in large deployments. However, these Lagrangian vectors do not need to be transmitted frequently and can be compressed or approximated. As part of future work, we aim to explore techniques like client sampling or clustered updates to limit the communication cost per round. This will help reduce the dimensionality of λ_i and ν_i, enabling more efficient communication without compromising convergence or privacy guarantees.

Moreover, as shown in Table V, PRIV-HFL requires more communication rounds to reach target accuracies compared to several existing methods. This increased round count stems from the use of a dual optimization framework based on Lagrangian multipliers, which decouples the direct exchange of model parameters in favor of optimizing auxiliary variables. While this design enhances privacy and decouples client updates, it introduces additional communication overhead, especially in earlier rounds. Nevertheless, PRIV-HFL achieves competitive accuracy while ensuring stronger privacy guarantees, highlighting a trade-off between communication efficiency and privacy preservation.

TABLE V: Comparison of different FL methods on CIFAR-10 ($\psi = 0.3$), in terms of the number of communication rounds to reach target test accuracy (*acc*), without privacy consideration

Model Name	**CIFAR-10**	
	acc = 75%	***acc*** = 80%
FedAvg	153.67±20.33	425.33±61.67
FedProx	143.67±0.33	391.67±13.33
FedDyn	90.67±2.33	183.67±23.33
MOON	128.00±10.00	347.00±24.00
PRIV-HFL	365.78±10.37	482.25±5.10
SCAFFOLD	100.33±14.67	212.00±24.00
FedGen	140.00±4.00	406.67±29.33
FedDF	132.67±11.33	329.00±42.00
FedFTG	92.68±31.33	188.67±31.33
FedDiff	**50.12±10.69**	**96.00±5.00**

5) Applicability: A notable limitation of PRIV-HFL lies in its assumption of homogeneous client model architectures, requiring all participants to use the same neural network structure with consistent parameter dimensions. While this assumption simplifies global aggregation and model fusion. It significantly restricts the framework's practical applicability, particularly in real-world federated learning deployments, where clients often vary in computational capabilities, hardware resources, and preferred model architectures. This homogeneity constraint limits PRIV-HFL's deployment in cross device FL scenarios, where supporting diverse models is essential. In future work, we plan to relax this assumption by incorporating heterogeneous model aggregation techniques, such as KD across dissimilar architectures, representation alignment, or intermediate feature matching. Such extensions would allow PRIV-HFL to generalize to more realistic and scalable federated systems while preserving its privacy-preserving benefits.

VI. Conclusion

In this work, we proposed PRIV-HFL, a novel data-free KD framework for FL that effectively balances privacy preservation and model utility, particularly under Non-IID client settings. By replacing traditional parameter sharing with alternative statistical representations, namely λ_i, ν_i, and ∇f_g^i PRIV-HFL safeguards clients against DRAs while ensuring collaborative learning remains effective. Leveraging a GAN based generative model, PRIV-HFL facilitates reliable pseudo-data generation, significantly improving model performance across heterogeneous clients. Compared to existing data-free methods such as FedDiff and FedFTG, PRIV-HFL achieves up to 50% higher accuracy on CIFAR-10 under strict privacy budgets ($\epsilon = 10$, $\psi = 0.5$), while maintaining SSIM values below 0.25, indicating strong privacy guarantees. Furthermore, PRIV-HFL introduces only a modest communication overhead and demonstrates scalability to large client populations without suffering from training instability or mode collapse. These results confirm PRIV-HFL as a robust and practical solution for privacy-preserving FL in real-world, data-sensitive environments.

Acknowledgment

The research underpinning this publication was undertaken while completing a PhD at La Trobe University, Melbourne, Victoria. We would also like to express our sincere gratitude to Professor Masoud Daneshtalab for his valuable guidance and insightful advice throughout the course of this research.

References

[1] D. C. Nguyen, M. Ding, and Pathirana, "6g internet of things: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 359–383, 2021.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[3] R. Gupta, A. Shukla, and S. Tanwar, "Bats: A blockchain and ai-empowered drone-assisted telesurgery system towards 6g," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 2958–2967, 2020.

[4] C.-C. Lin, D.-J. Deng, Y.-L. Chih, and H.-T. Chiu, "Smart manufacturing scheduling with edge computing using multiclass deep q network," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4276–4284, 2019.

[5] S. K. Singh, Y.-S. Jeong, and J. H. Park, "A deep learning-based iot-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, p. 102252, 2020.

[6] D. Vimalajeewa and Kulatunga, "A service-based joint model used for distributed learning: Application for smart agriculture," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 2, pp. 838–854, 2021.

[7] B. McMahan and Moore, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[8] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[9] N. Carlini, S. Chien, and Nasr, "Membership inference attacks from first principles," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 1897–1914.

[10] C. A. Choquette-Choo, "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.

[11] Y. Long and Wang, "A pragmatic approach to membership inferences on machine learning models," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 521–534.

[12] N. Carlini and Tramer, "Extracting training data from large language models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2633–2650.

[13] J. Chen, Y. Zhao, Q. Li, X. Feng, and K. Xu, "Feddef: defense against gradient leakage in federated learning-based network intrusion detection systems," *IEEE Transactions on Information Forensics and Security*, 2023.

[14] N. Haim, G. Vardi, G. Yehudai, O. Shamir, and M. Irani, "Reconstructing training data from trained neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 911–22 924, 2022.

[15] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 603–618.

[16] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Y. Gao, L. Zhang, L. Wang, K.-K. R. Choo, and R. Zhang, "Privacy-preserving and reliable decentralized federated learning," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 2879–2891, 2023.

[18] X. Wu, Y. Zhang, M. Shi, P. Li, R. Li, and N. N. Xiong, "An adaptive federated learning scheme with differential privacy preserving," *Future Generation Computer Systems*, vol. 127, pp. 362–372, 2022.

[19] A. Anand, S. Dhakal, and Akdeniz, "Differentially private coded federated linear regression," in *2021 IEEE Data Science and Learning Workshop (DSLW)*. IEEE, 2021, pp. 1–6.

[20] K. Wei, J. Li, and Ding, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE transactions on information forensics and security*, vol. 15, pp. 3454–3469, 2020.

[21] B. McMahan and Moore, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[22] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," 2019.

[23] A. Khaled, K. Mishchenko, and P. Richtarik, "Tighter theory for local sgd on identical and heterogeneous data," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, 26–28 Aug 2020, pp. 4519–4529.

[24] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[25] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2351–2363. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/18df51b97ccd68128e994804f3eccc87-Paper.pdf

[26] F. Sattler, T. Korjakow, R. Rischke, and W. Samek, "Fedaux: Leveraging unlabeled auxiliary data in federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 9, pp. 5531–5543, 2023.

[27] H.-Y. Chen and W.-L. Chao, "Fedbe: Making bayesian model ensemble applicable to federated learning," 2021.

[28] M. Najafi, M. Daneshtalab, J.-A. Lee, G. Saadloonia, and S. Shin, "Enhancing global model performance in federated learning with non-iid data using a data-free generative diffusion model," *IEEE Access*, vol. 12, pp. 148 230–148 239, 2024.

[29] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-iid federated learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 174–10 183.

[30] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[31] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[32] S. Z. El Mestari, G. Lenzini, and H. Demirci, "Preserving data privacy in machine learning systems," *Computers Security*, vol. 137, p. 103605, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823005151

[33] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[34] A. Yazdinejad, A. Dehghantanha, H. Karimipour, G. Srivastava, and R. M. Parizi, "A robust privacy-preserving federated learning model against model poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 6693–6708, 2024.

[35] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," *ACM Comput. Surv.*, vol. 57, no. 6, Feb. 2025. [Online]. Available: https://doi.org/10.1145/3712001

[36] K. Tjell and R. Wisniewski, "Privacy preservation in distributed optimization via dual decomposition and admm," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 7203–7208.

[37] I. Goodfellow and J. Pouget-Abadie, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[38] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients – how easy is it to break privacy in federated learning?" 2020. [Online]. Available: https://arxiv.org/abs/2003.14053

[39] Y. Sun, Z. Liu, J. Cui, J. Liu, K. Ma, and J. Liu, "Client-side gradient inversion attack in federated learning using secure aggregation," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 28 774–28 786, 2024.

[40] X. Xu, P. Liu, W. Wang, H.-L. Ma, B. Wang, Z. Han, and Y. Han, "Cgir: Conditional generative instance reconstruction attacks against federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4551–4563, 2023.

[41] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients-how easy is it to break privacy in federated learning?" *Advances in neural information processing systems*, vol. 33, pp. 16 937–16 947, 2020.

[42] T. Li, A. K. Sahu, M. Zaheer, and Sanjabia, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429–450, 2020.

[43] S. P. Karimireddy, S. Kale, and Mohri, "Scaffold: Stochastic controlled averaging for federated learning," in *International conference on machine learning*. PMLR, 2020, pp. 5132–5143.

[44] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *International conference on machine learning*. PMLR, 2021, pp. 12 878–12 889.

[45] M. Varun, S. Feng, H. Wang, S. Sural, and Y. Hong, "Towards accurate and stronger local differential privacy for federated learning with staircase randomized response," in *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy*, 2024, pp. 307–318.

[46] G. Gad, E. Gad, Z. M. Fadlullah, M. M. Fouda, and N. Kato, "Communication-efficient and privacy-preserving federated learning via joint knowledge distillation and differential privacy in bandwidth-constrained networks," *IEEE Transactions on Vehicular Technology*, 2024.

[47] Q. Zheng, S. Chen, Q. Long, and W. Su, "Federated f-differential privacy," in *International conference on artificial intelligence and statistics*. PMLR, 2021, pp. 2251–2259.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] G. Fang, J. Song, C. Shen, X. Wang, D. Chen, and M. Song, "Data-free adversarial distillation," *arXiv preprint arXiv:1912.11006*, 2019.

[50] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[51] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges — yann.lecun.com," https://yann.lecun.com/exdb/mnist/, 2010, [Accessed 14-08-2024].

[52] D. A. E. Acar and Zhao, "Federated learning based on dynamic regularization," *arXiv preprint arXiv:2111.04263*, 2021.

[53] C. He, S. Li, and So, "Fedml: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[54] M. Yurochkin and Agarwal, "Bayesian nonparametric federated learning of neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 7252–7261.

[55] M. Talaei and I. Izadi, "Adaptive differential privacy in federated learning: A priority-based approach," *arXiv preprint arXiv:2401.02453*, 2024.

[56] H. Liu, B. Li, C. Gao, P. Xie, and C. Zhao, "Privacy-encoded federated learning against gradient-based data reconstruction attacks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5860–5875, 2023.