FedLAP-DP: Federated Learning by Sharing Differentially Private Loss Approximations

Hui-Po Wang
CISPA Helmholtz Center for Information Security
Germany
hui.wang@cispa.de

Dingfan Chen
CISPA Helmholtz Center for Information Security
Germany
dingfan.chen@cispa.de

Raouf Kerkouche
CISPA Helmholtz Center for Information Security
Germany
raouf.kerkouche@cispa.de

Mario Frtiz
CISPA Helmholtz Center for Information Security
Germany
fritz@cispa.de

ABSTRACT

Conventional gradient-sharing approaches for federated learning (FL), such as FedAvg, rely on aggregation of local models and often face performance degradation under differential privacy (DP) mechanisms or data heterogeneity, which can be attributed to the inconsistency between the local and global objectives. To address this issue, we propose FedLAP-DP, a novel privacy-preserving approach for FL. Our formulation involves clients synthesizing a small set of samples that approximate local loss landscapes by simulating the gradients of real images within a local region. Acting as loss surrogates, these synthetic samples are aggregated on the server side to uncover the global loss landscape and enable global optimization. Building upon these insights, we offer a new perspective to enforce record-level differential privacy in FL. A formal privacy analysis demonstrates that FedLAP-DP incurs the same privacy costs as typical gradient-sharing schemes while achieving an improved trade-off between privacy and utility. Extensive experiments validate the superiority of our approach across various datasets with highly skewed distributions in both DP and non-DP settings. Beyond the promising performance, our approach presents a faster convergence speed compared to typical gradient-sharing methods and opens up the possibility of trading communication costs for better performance by sending a larger set of synthetic images. The source is available at https://github.com/a514514772/FedLAP-DP.

KEYWORDS

neural networks, federated learning, differential privacy

1 INTRODUCTION

Federated Learning (FL) [36] is a distributed learning framework that allows participants to train a model collaboratively without sharing their data. Predominantly, existing works [19, 31, 36] achieve this by training local models on clients' private datasets for several local epochs and sharing only the averaged gradients with the central server. Despite extensive research over the past few years, these prevalent gradient-based methods still suffer from several challenges [18] when training with heterogeneous clients. These issues could result in significant degradation in performance and convergence speed, thus making FL difficult to scale up.

Aside from the data heterogeneity that has been notoriously known to interfere with federated optimization [16, 19, 32], potential privacy breaches can still occur in the FL gradient-sharing scheme. For instance, data reconstruction [4, 11, 13, 15, 29, 48, 61, 64] and membership inference attacks [37, 40] have been widely studied in FL applications. In response to these emerging privacy risks, differential privacy (DP) has been established as the golden standard for providing theoretical privacy guarantees against potential privacy leakage. The most prevalent algorithm to ensure DP properties, namely DP-SGD [1], operates by clipping per-sample gradients and adding Gaussian noise, thus obfuscating the influence of each data record. Despite its guarantee, the noisy gradients introduced by DP-SGD could induce additional heterogeneity, an aspect not thoroughly explored in existing FL literature. This aligns with the observation that DP noise leads to a more significant performance drop in FL problems than in centralized settings and was partially explored by Yang et al. [56] in a personalized FL setting.

The fundamental cause of such degradation is that *the local updates, driven by the distinct objectives of heterogeneous clients, optimize the models towards their local minima instead of the global objective*. This inconsistency drives the (weighted) average models learned by existing approaches to sub-optimal performance or even fails the convergence. Existing efforts have proposed various ideas to address the issue of heterogeneity, including variance-reduction [19], reducing the dissimilarity among clients [30, 31, 51], or personalized models [10, 33, 35]. However, these methods mainly focus on non-DP cases and still heavily rely on biased local updates, which are the root cause of the degradation. Overall, the gap between DP, data heterogeneity, and objective inconsistency has yet to be fully bridged and deserves further investigation.

As a pioneering step, this work proposes FedLAP-DP, a novel differentially private framework designed to approximate local loss landscapes and counteract biased federated optimization through the utilization of synthetic samples. As illustrated in Fig. 1, unlike traditional gradient-sharing schemes [36] that are prone to inherently biased global update directions, our framework transmits synthetic samples encoding the local optimization landscapes. This enables the server to faithfully uncover the global loss landscape,

Proceedings on Privacy Enhancing Technologies 2024(3), 372–390

https://doi.org/10.56553/popets-2024-0083

overcoming the biases incurred by conventional gradient-sharing schemes and resulting in substantial improvements in convergence speed (refer to Sec. 6). Additionally, we introduce the usage of a trusted region to faithfully reflect the approximation quality, further mitigating bias stemming from potential imperfections in the local approximation within our scheme.

Based on the insights, our approach offers a novel perspective to incorporate record-level differential privacy into federated learning. In particular, we begin with applying DP protection mechanisms, such as DP-SGD, to the gradients produced from real client data. These DP-protected gradients then serve as the learning objective for synthetic dataset optimization. Thanks to the post-process theorem, we are allowed to conduct multiple optimization steps based on the protected real gradients to further improve the approximation quality of the synthetic data without incurring additional privacy costs. Our formal privacy analysis demonstrates that FedLAP-DP consumes the same privacy costs as traditional gradient-sharing baselines. Together with the proposed trusted regions, FedLAP-DP provides reliable utility under privacy-preserving settings, especially when considering low privacy budgets and highly skewed data distributions.

Lastly, we note that our approach has several additional benefits. (1) Our approach is more communication efficient as it enables the execution of multiple optimization rounds on the server side. This is particularly advantageous for large models, where transferring synthetic data is notably less costly than gradients in each round. (2) While FedLAP-DP exhibits superior performance under the same communication costs as the baselines, our method can increase the convergence speed and further improve performance by synthesizing a larger set of synthetic images. This opens up the possibility of trading additional costs for better performance, which is little explored in existing literature. Overall, we summarize our contributions as follows.

- We propose FedLAP-DP that offers a novel perspective of federated optimization by transferring local loss landscapes to uncover the global loss landscape on the server side, avoiding potentially biased local updates and enabling unbiased federated optimization on the server.
- We demonstrate how to synthesize samples that approximate local loss landscapes and effective regions on clients, minimizing the effect of imperfect approximation.
- Along with a formal privacy analysis, we demonstrate an innovative way to integrate record-level DP into federated learning without incurring additional privacy costs.
- Extensive experiments confirm the superiority of our formulation over existing gradient-sharing baselines in terms of performance and convergence speed, particularly when considering tight privacy budgets and highly skewed data distributions.

2 RELATED WORK

Non-IID Data in Federated Learning. The existing gradient-sharing schemes typically begin with training local models on local private data for several local epochs, and the server aggregates those models to update the global model. However, the applications of FL may naturally introduce heterogeneity among clients due to the contrasting behaviors of users. Such heterogeneity inevitably results in the inconsistency between local and global learning objectives, making the (weighted) averages of local models sub-optimal to the global learning task. This issue [18] often causes degradation in performance and convergence speed and has attracted a significant amount of attention from the community.

To address the issue, existing efforts mainly fall into the following categories: variance-reduction techniques [19], constraining the dissimilarity of updates among clients [30, 31, 51], and adjusting the global model to a personalized version at the inference stage [10, 33, 35]. Variance-reduction techniques [19] consider an additional variable to explicitly correct the error induced by the heterogeneity. However, the variable consumes additional communication costs and could be costly. For instance, SCAFFOLD [19] consumes twice the cost compared to plain FedAvg. On the other hand, several prior works propose various methods to limit the dissimilarity among clients, such as additional regularization terms [31, 51] and contrastive learning [30]. Despite the efficacy, these methods still rely on local models that are biased toward the local optimum. Recently, a growing body of literature has investigated personalizing federated models to compensate for the heterogeneity [10, 33, 35]. These personalized methods target different applications and may be constrained when the models are deployed for tasks that extend beyond the original scope of the clients' applications, such as in transfer learning scenarios, even though some existing efforts [12] have been made to make them DP.

In contrast, our approach takes advantage of loss landscape approximation by synthetic samples, offering a new global perspective to federated optimization. It does not rely on potentially biased locally updated models and outperforms typical gradient-sharing schemes without introducing additional communication costs or personalized models.

Differential Privacy in Machine Learning. To address privacy concerns, Differential Privacy (DP) has been established as a standard framework for formalizing privacy guarantees, as outlined in Dwork's foundational work [9]. However, implementing DP involves a compromise between model utility and privacy. This is due to the necessity of modifying model training processes with techniques like clipping updates and injecting noise, which can lead to negative effects of the model performance. Current state-of-the-art methods for training models while maintaining high utility under DP primarily use transfer learning. This involves leveraging models pre-trained on extensive public datasets, subsequently fine-tuned with private data [57]. These methods have proven effective across various fields where substantial public datasets exist, showing promising results even with limited private data [46]. Moreover, parameter-efficient fine-tuning techniques like adaptors, including LoRA [17], have demonstrated competitiveness in transfer learning within DP contexts[46, 58]. Similarly, compression strategies have been investigated to minimize the model's size or its updates. This size reduction consequently lowers the sensitivity and the associated noise introduced by DP [20, 22]. However, only a limited number of research studies have explored the application of DP in federated settings characterized by significant heterogeneity

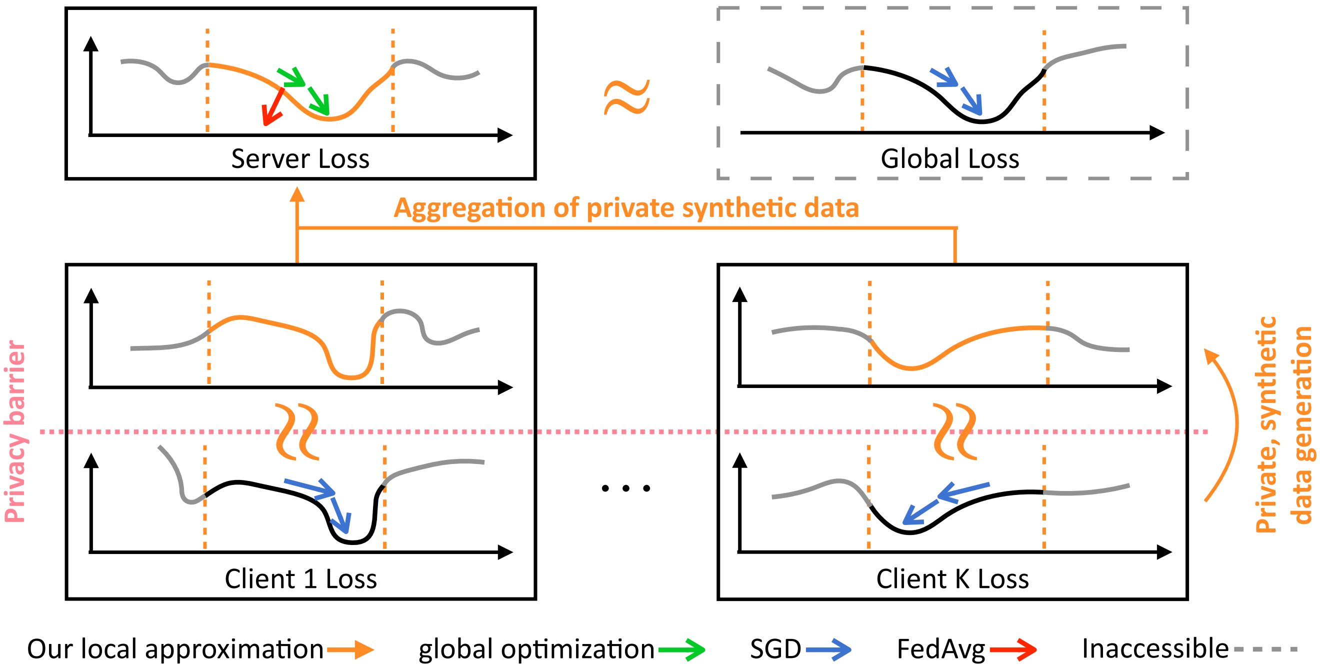


Figure 1: An overview of FedLAP-DP. FedLAP-DP addresses the limitations of methods like FedAvg, which aggregate locally optimized gradients (blue) to meet a global objective. Such methods often lead to sub-optimal results (red) due to goal misalignment. This problem further intensifies with increasing client data heterogeneity. FedLAP-DP approximates local neighborhoods with synthetic images on the clients (local approximation, Sec. 4.2) and optimizes the model according to the reconstructed loss landscape on the server (global optimization, Sec. 4.3). Differential privacy is integrated to introduce privacy barriers (Sec. 4.4).

among client datasets [2, 34, 41]. These approaches, similar to FedAvg, heavily rely on biased local models, which is the root cause of performance degradation.

Dataset Distillation. Our work is largely motivated by recent progress in the field of *dataset distillation* [5, 52, 60, 62], a process aimed at distilling the necessary knowledge of model training into a small set of synthetic samples. These samples are optimized to serve as surrogates for real data in subsequent model training. The concept of dataset distillation dates back to Wang et al. [52], who formulated the dataset distillation as a bi-level optimization problem and introduced a meta-learning approach. In this approach, the inner optimization simulates training a neural network on a distilled synthetic dataset, while the outer optimization updates the synthetic dataset through gradient descent. While this method has shown effectiveness for certain smaller networks, it requires substantial computational resources due to the gradient unrolling operation required in bi-level optimization, making it challenging to be used with larger, more common models.

Recent developments have enhanced this original approach by eliminating the need for complex bi-level optimization. In this context, various directions have been explored, such as gradient matching [62], distribution matching [60], and trajectory matching [5]. Our approach builds upon gradient matching, which offers more computational efficiency and better compatibility with DP training than trajectory matching. Furthermore, it more effectively exploits discriminative information from a specific global model, making it generally more suitable for FL settings. Unlike distribution matching, which tends to prioritize generalization across various downstream models, gradient matching focuses on specialization for a certain global model, a subtler but more advantageous trade-off for FL training. Specifically, our approach draws inspiration from DSC [62] but incorporates several key distinctions to enhance the efficacy and applicability in FL training.

The proposed FedLAP-DP (1) focuses on finding local approximation and assembling the global loss landscape to facilitate federated optimization, (2) is class-agnostic and complements record-level differential privacy while prior works often consider class-wise alignment and could cause privacy risks, and (3) is designed for multi-round training with several critical design choices. Notably, the most closely related research to ours is PSG [7], which also explored class-agnostic distillation with DP guarantees. However, the focus of PSG is on single-round distillation rather than the standard multi-round federated learning.

3 BACKGROUND

3.1 Federated Learning

In federated learning, we consider training a model $\mathbf{w}$ that maps the input $\boldsymbol{x}$ to the output prediction y. We assume K clients participate in the training, and each owns a private dataset $\mathcal{D}_k$ associated with distribution p_k. We use the subscript k to represent the indices of clients and the superscript m and t to denote the m-th communication round and t-th local step, respectively, unless stated otherwise.

Overall, the learning objective is to find the optimal model $\mathbf{w}$ that minimizes the empirical global loss over the population distribution:

$$\mathcal{L}(\mathbf{w}) = \mathbb{E}_{(\boldsymbol{x},y)\sim p}[\ell(\mathbf{w},\boldsymbol{x},y)] = \frac{1}{N}\sum_{j=1}^{N}\ell(\mathbf{w},\boldsymbol{x}_j,y_j) \tag{1}$$

where ℓ could be arbitrary loss criteria such as cross-entropy and N is the total dataset size. However, in federated settings, direct access to the global objective is prohibited as all client data is stored locally. Instead, the optimization is conducted on local surrogate objectives $\mathcal{L}_k(\mathbf{w})$:

$$\mathcal{L}(\mathbf{w}) = \sum_{k=1}^{K}\frac{N_k}{N}\mathcal{L}_k(\mathbf{w})\,,\ \mathcal{L}_k(\mathbf{w}) = \sum_{j=1}^{N_k}\frac{1}{N_k}\ell(\mathbf{w},\boldsymbol{x}_j,y_j),$$

where $(\boldsymbol{x}_j, y_j)$ are data samples from the client dataset $\mathcal{D}_k$.

Existing methods, such as FedAvg [36], simulate stochastic gradient descent on the global objective by performing local gradient updates and periodically aggregating and synchronizing them on the server side. Specifically, at the m-th communication round, the server broadcasts the current global model weights $\mathbf{w}_g^{m,1}$ to each client, who then performs T local iterations with learning rate η.

$$\begin{gathered}\mathbf{w}_k^{m,1} \leftarrow \mathbf{w}_g^{m,1}, \forall k \in [K]\\ \mathbf{w}_k^{m,t+1} = \mathbf{w}_k^{m,t} - \eta\nabla\mathcal{L}_k(\mathbf{w}_k^{m,t}), \forall t \in [T]\end{gathered} \tag{2}$$

The local updates $\Delta\mathbf{w}_k^m$ are then sent back to the server and combined to construct $\widehat{\boldsymbol{g}^m}$, which is essentially a linear approximation of the true global update $\boldsymbol{g}^m$:

$$\begin{gathered}\widehat{\boldsymbol{g}^m} = \sum_{k=1}^{K}\frac{N_k}{N}\Delta\mathbf{w}_k^m = \sum_{k=1}^{K}\frac{N_k}{N}(\mathbf{w}_k^{m,T} - \mathbf{w}_k^{m,1})\\ \mathbf{w}_g^{m+1,1} = \mathbf{w}_g^{m,1} - \eta\widehat{\boldsymbol{g}^m}\end{gathered} \tag{3}$$

In this work, we focus on the conventional FL setting in which each client retains their private data locally, and the local data cannot be directly accessed by the server. Every data sample is deemed private, with neither the server nor the clients using any additional (public) data, which stands in contrast to previous works that require extra data on the server side [28, 63]. Furthermore, all clients aim towards a singular global objective (Eq. 1), which is distinct from personalized approaches wherein evaluations are conducted based on each client's unique objective and their own data distribution [10, 33].

3.2 Non-IID Challenges

The heterogeneity of client data distributions presents several major challenges to FL, such as a significant decrease in the convergence speed (and even divergence) and the final performance when compared to the standard IID training setting [19, 23, 31, 32]. This can be easily seen from the mismatch between the local objectives that are being solved and the global objective that FL ultimately aims to achieve, i.e., $\mathcal{L}_k(\mathbf{w}) \neq \mathbb{E}_{(\boldsymbol{x},y)\sim p}[\ell(\mathbf{w},\boldsymbol{x},y)]$ if $p_k \neq p$ for some k. Executing multiple local steps on the local objective (Eq. 2) makes the local update $\Delta\mathbf{w}_k^m$ deviate heavily from the true global gradient $\nabla\mathcal{L}(\mathbf{w})$, inevitably resulting in a biased approximation of the global gradient via Eq. 3, i.e., $\widehat{\boldsymbol{g}^m} \neq \boldsymbol{g}^m$, where $\boldsymbol{g}^m$ is derived from the true loss $\mathcal{L}(\boldsymbol{w})$ (See Fig. 1 for a demonstration.).

Despite significant advances achieved by existing works in alleviating divergence issues, these methods still exhibit a systematic bias generally deviating from optimizing the global objective as they rely on the submitted client updates $\Delta\mathbf{w}_k^m$, which only reflect a single direction towards the client's local optimum.

In contrast, our method communicates the synthetic samples $\mathcal{S}_k$ that encode the local optimization landscapes, i.e., gradient directions within a trust region around the starting point that summarize on possible trajectories $(\mathbf{w}_k^{m,1}, \mathbf{w}_k^{m,2}, ..., \mathbf{w}_k^{m,T+1})$. This differs significantly from traditional methods that convey only a single direction $\Delta\mathbf{w}_k^m = \mathbf{w}_k^{m,T+1} - \mathbf{w}_k^{m,1}$. This fundamental change provides the central server with a global perspective that faithfully approximates the ground-truth global optimization (See Fig. 1 top row) than existing approaches.

3.3 Differential Privacy

Differential Privacy provides theoretical guarantees of privacy protection while allowing for quantitative measurement of utility. We review several definitions used in this work in this section.

Definition 3.1 (Differential Privacy [9]). A randomized mechanism $\mathcal{M}$ with range $\mathcal{R}$ satisfies (ε, δ)-DP, if for any two adjacent datasets E and E', i.e., $E' = E \cup \{\boldsymbol{x}\}$ for some $\boldsymbol{x}$ in the data domain (or vice versa), and for any subset of outputs $O \subseteq \mathcal{R}$, it holds that

$$\Pr[\mathcal{M}(E) \in O] \leq e^{\varepsilon}\Pr[\mathcal{M}(E') \in O] + \delta \tag{4}$$

Intuitively, DP guarantees that an adversary, provided with the output of $\mathcal{M}$, can only make nearly identical conclusions (within an ε margin with probability greater than $1 - \delta$) about any specific record, regardless of whether it was included in the input of $\mathcal{M}$ or not [9]. This suggests that, for any record owner, a privacy breach due to its participation in the dataset is unlikely.

In FL, the notion of *adjacent (neighboring) datasets* used in DP generally refers to pairs of datasets differing by either one user (*user-level* DP) or a single data point of one user (*record-level* DP). Our work focuses on the latter. The definition is as follows.

Definition 3.2 (Record-level DP). A randomized mechanism $\mathcal{M}$ with range $\mathcal{R}$ is said to satisfy (ε, δ) record-level DP in the context of federated learning if, for any pair of adjacent datasets E and E' differing by a single data point on a client (i.e., $E' = E \cup \{\boldsymbol{x}\}$), the mechanism satisfies (ε, δ)-DP. Here, $E = \bigcup E_i$ denotes the union of client datasets.

While there are established methods providing record-level DP for training federated models [21, 44, 47], these primarily operate on the transmitted single client gradients. In contrast, our novel formulation allows efficient communication of comprehensive information, thereby circumventing biased optimization and displaying improved training stability and utility.

We use the Gaussian mechanism to upper bound privacy leakage when transmitting information from clients to the server.

Definition 3.3 (Gaussian Mechanism [9]). Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^d$ be an arbitrary function with sensitivity being the maximum Euclidean

Algorithm 1 FedLAP: Local Approximation

```
function ClientExecute($k$, $r$, $\mathbf{w}_g^{m,1}$) :
  Initialize $\mathcal{S}_k$: $\{\hat{\boldsymbol{x}}_k^m\}$ from Gaussian noise or $\{\hat{\boldsymbol{x}}_k^{m-1}\}$, $\{\hat{y}_k\}$ to
  be a balanced set
  for $i = 1, \dots, R_i$ do
    /* Resample training trajectories */
    Reset $t \leftarrow 1$, model $\mathbf{w}_k^{m,1} \leftarrow \mathbf{w}_g^{m,1}$, and $\mathcal{S}_k^{i,0} \leftarrow \mathcal{S}_k^{i-1}$
    while $||\mathbf{w}_k^{m,t} - \mathbf{w}_k^{m,1}|| < r$ do
      Sample real data batches $\{(\mathbf{x}_k, y_k)\}$ from $\mathcal{D}_k$
      Compute $g^{\mathcal{D}} = \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \{(\mathbf{x}_k, y_k)\})$
      for $j = 1, \dots, R_b$ do
        /* Update synthetic set $\mathcal{S}_k$ given the real gradient */
        $\mathcal{S}_k^{i,j+1} = \mathcal{S}_k^{i,j} - \tau \nabla_{\mathcal{S}_k} \mathcal{L}_{dis}\left(g^{\mathcal{D}}, \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k^{i,j})\right)$
      end for
      for $l = 1, \dots, R_l$ do
        /* Update local models from $\mathbf{w}_k^{m,t}$ to $\mathbf{w}_k^{m,t+l}$ */
        $\mathbf{w}_k^{m,t+1} = \mathbf{w}_k^{m,t} - \eta \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k)$
        $t \leftarrow t + 1$
      end for
    end while
  end for
  Measure $r_k$ on $\mathcal{D}_k$ (See Fig. 2)
Return: Synthetic set $\mathcal{S}_k^{R_i}$, calibrated radius $r_k$
```

distance between the outputs over all adjacent datasets E and E':

$$\Delta_2 f = \max_{E,E'} \|f(E) - f(E')\|_2 \tag{5}$$

The Gaussian Mechanism $\mathcal{M}_\sigma$, parameterized by σ, adds noise into the output, i.e.,

$$\mathcal{M}_\sigma(\boldsymbol{x}) = f(\boldsymbol{x}) + \mathcal{N}(0, \sigma^2 \mathbb{I}). \tag{6}$$

$\mathcal{M}_\sigma$ is (ε, δ)-DP for $\sigma \geq \sqrt{2 \ln(1.25/\delta)} \Delta_2 f / \varepsilon$.

Theorem 3.4 (Post-processing [9]). *If $\mathcal{M}$ satisfies (ε, δ)-DP, $G \circ \mathcal{M}$ will satisfy (ε, δ)-DP for any data-independent function G.*

Moreover, we use Theorem 3.4 to guarantee that the privacy leakage is bounded upon obtaining gradients from real private data in our framework. This forms the basis for the overall privacy guarantee of our framework and enables us to enhance the approximation quality without introducing additional privacy costs.

4 FEDLAP-DP

4.1 Overview

In this section, we start by introducing FedLAP, the non-DP variant of FedLAP-DP, to demonstrate the concept of loss approximation and global optimization. Next, we discuss the effective integration of record-level DP into Federated Learning (FL). This will be followed by a detailed privacy analysis presented in Sec. 5.

Contrary to conventional methods that generally transmit local update directions to estimate the global objective (Eq. 3), FedLAP uniquely simulates global optimization by sending a compact set of synthetic samples. These samples effectively represent the local loss landscapes, as illustrated in Fig. 1.

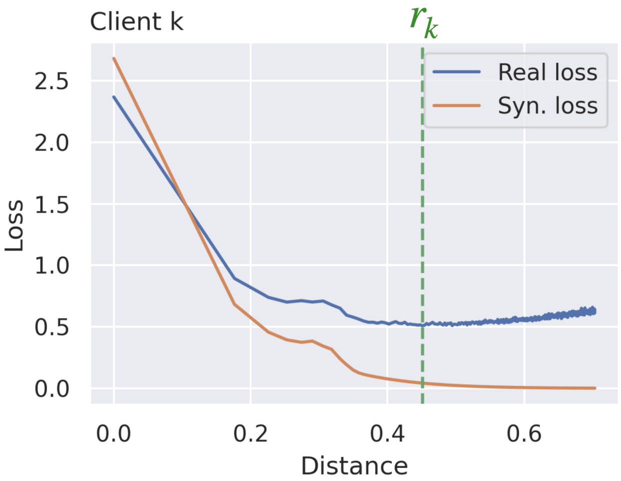


Figure 2: r_k selection. The loss on private real and synthetic data decreases initially but deviates later. r_k is defined as the turning points with the smallest real loss.

Let p_k and the $p_{\mathcal{S}_k}$ be the distribution of the real client dataset $\mathcal{D}_k$ and the corresponding synthetic dataset $\mathcal{S}_k$, respectively. We formalize our objective and recover the global objective as follows:

$$\begin{gathered} \mathbb{E}_{(\boldsymbol{x},y)\sim p_k}[\ell(\mathbf{w}, \boldsymbol{x}, y)] \simeq \mathbb{E}_{(\hat{\boldsymbol{x}},\hat{y})\sim p_{\mathcal{S}_k}}[\ell(\mathbf{w}, \hat{\boldsymbol{x}}, \hat{y})] \\ \mathcal{L}(\mathbf{w}) = \sum_{k=1}^{K} \frac{N_k}{N} \mathcal{L}_k(\mathbf{w}) \simeq \sum_{k=1}^{K} \frac{N_k}{N} \widehat{\mathcal{L}}_k(\mathbf{w}) \end{gathered} \tag{7}$$

Thus, performing global updates is then equivalent to conducting vanilla gradient descent on the recovered global objective, i.e., by training on the synthetic set of samples.

We demonstrate our framework in Fig. 1. In every communication round, synthetic samples are optimized to approximate the client's local loss landscapes (Sec 4.2) and then transmitted to the server. The server then performs global updates on the synthetic samples to simulate global optimization (Sec. 4.3). Finally, Sec. 4.4 explains the integration of *record-level* differential privacy into Federated Learning, resulting in the creation of FedLAP-DP.

The overall algorithm is depicted in Algorithm 1 and 2 for non-DP settings and Algorithm 3 for DP settings, where the indices i being the number of training trajectories observed by the synthetic images and j being the number of updates on a sampled real batch. For conciseness, we omit the indices in the following where their absence does not affect clarity or understanding.

4.2 Local Approximation

The goal of this step is to construct a set of synthetic samples $\mathcal{S}_k$ that accurately captures necessary local information for subsequent global updates. A natural approach would be to enforce similarity between the gradients obtained from the real client data and those obtained from the synthetic set:

$$\nabla_{\mathbf{w}} \mathbb{E}_{(\boldsymbol{x},y)\sim p_k}[\ell(\mathbf{w}, \boldsymbol{x}, y)] \simeq \nabla_{\mathbf{w}} \mathbb{E}_{(\hat{\boldsymbol{x}},\hat{y})\sim p_{\mathcal{S}_k}}[\ell(\mathbf{w}, \hat{\boldsymbol{x}}, \hat{y})]$$

We achieve this by minimizing the distance between the gradients:

$$\arg\min_{\mathcal{S}_k} \quad \mathcal{L}_{\text{dis}}\Big(\nabla\mathcal{L}(\mathbf{w}, \mathcal{D}_k), \nabla\mathcal{L}(\mathbf{w}, \mathcal{S}_k)\Big) \tag{8}$$

where $\nabla\mathcal{L}(\mathbf{w}, \mathcal{D}_k))$ denotes the stochastic gradient of network parameters on the client dataset $\mathcal{D}_k$, and $\nabla\mathcal{L}(\mathbf{w}, \mathcal{S}_k)$ the gradient on the synthetic set for brevity. $\mathcal{L}_{\text{dis}}$ can be arbitrary metric that measures the similarity.

Distance Metric. In this study, we utilize a layer-wise cosine distance as described by Zhao et al. [62]. Furthermore, we have empirically determined that incorporating a mean square error term, which captures directional information and controls the differences in magnitudes, enhances both stability and approximation quality. We provide a more comprehensive explanation and analysis in Sec. A.1. The combination of the distance metric and the mean square error term is formalized as follows.

$$\begin{aligned} &\mathcal{L}_{\text{dis}}\Big(\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, \mathcal{D}_k), \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, \mathcal{S}_k)\Big) \\ &= \sum_{l=1}^{L} d\Big(\nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)}, \mathcal{D}_k), \nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)}, \mathcal{S}_k)\Big), \\ &+\lambda\|\nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)}, \mathcal{D}_k) - \nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)}, \mathcal{S}_k)\|_2^2 \end{aligned} \tag{9}$$

where λ is a hyper-parameter controlling the strength of regularization, and d denotes the cosine distance between the gradients at each layer:

$$d(\boldsymbol{A}, \boldsymbol{B}) = \sum_{i=1}^{out}\left(1 - \frac{\boldsymbol{A}_{i\cdot} \cdot \boldsymbol{B}_{i\cdot}}{\|\boldsymbol{A}_{i\cdot}\|\|\boldsymbol{B}_{i\cdot}\|}\right) \tag{10}$$

$\boldsymbol{A}_{i\cdot}$ and $\boldsymbol{B}_{i\cdot}$ represent the flattened gradient vectors corresponding to each output node i. In fully-connected layers, the parameters $\boldsymbol{w}^{(l)}$ inherently form a 2D tensor, whereas the parameters of convolutional layers are represented as 4D tensors. To compute the loss, we flatten the last three dimensions, converting them into 2D tensors as well. denote the number of output and input channels, kernel height, and width, respectively.

Effective Approximation Regions. While solving Eq. 8 for every possible $\mathbf{w}$ would lead to perfect recovery of the ground-truth global optimization in principle. However, it is practically infeasible due to the large space of (infinitely many) possible values of $\mathbf{w}$. Additionally, as $|\mathcal{S}_k|$ is set to be much smaller than N_k (for the sake of communication efficiency), an exact solution may not exist, resulting in approximation error for some $\mathbf{w}$. To address this, we explicitly constrain the problem space to be the most achievable region for further global updates. Specifically, we consider $\mathbf{w}_k$ that is sufficiently close to the initial point of the local update and is located on the update trajectories (Eq. 12). Formally,

$$\arg\min_{\mathcal{S}_k} \sum_{t=1}^{T} \mathcal{L}_{\text{dis}}\Big(\nabla\mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{D}_k), \nabla\mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k)\Big) \tag{11}$$

$$\text{s.t.} \quad \|\mathbf{w}_k^{m,t} - \mathbf{w}_k^{m,1}\| < r, \tag{12}$$

$$\mathbf{w}_k^{m,t+1} = \mathbf{w}_k^{m,t} - \eta\nabla\mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k) \tag{13}$$

where r represents a radius suggested by the server, defining the coverage of update trajectories, and η denotes the model update learning rate shared among the server and clients.

In the m-th communication round, the clients first synchronize the local model $\mathbf{w}_k^{m,1}$ with the global model $\mathbf{w}_g^{m,1}$ and initialize the synthetic features $\{\hat{\mathbf{x}}_k^m\}$ either from Gaussian noise or to be the ones obtained from the previous round $\{\hat{\mathbf{x}}_k^{m-1}\}$. Synthetic labels $\{\hat{y}_k\}$ are initialized to be a fixed, balanced set and are not optimized during the training process. The number of synthetic samples $|\mathcal{S}_k|$ is kept equal for all clients in our experiments, though it can be adjusted for each client depending on factors such as local dataset size and bandwidth in practice.

To simulate the local training trajectories, the clients alternate between updating synthetic features using Eq. 11 and updating the local model using Eq. 12. This process continues until the current local model weight $\mathbf{w}_k^{m,t}$ exceeds a predefined region r determined by the Euclidean distance on the flattened weight vectors, meaning it is no longer close to the initial point. On the other hand, the server optimization should take into consideration the approximation quality of $\mathcal{S}_k$. Thus, as illustrated in Fig. 2, each client will suggest a radius r_k indicating the distance that $\mathcal{S}_k$ can approximate best within the radius r. For the DP training setting, we make the choice of r_k data-independent by setting it to be a constant (the same as r in our experiments).

Details of Algorithm. The process is outlined in Algorithm 1. In detail, the local model $w^{m,t}$ is reset to the initial global model $w^{m,1}$ for R_i iterations. This step explores various training paths, enhancing generalizability. Synthesis occurs when the current model $w^{m,t}$ is within a certain distance, defined as the server-suggested radius r, from the initial point $w^{m,1}$. For each real batch, we update synthetic images R_b times and the local model R_l times, moving from local step t to $t + R_l$. Finally, the client assesses the effective approximation regions and communicates the radius r_k. Building on previous studies [62], we note that the constraints are not enforced through classical means of constrained optimization, like KKT conditions. Rather, they are applied via the termination of loops, a method necessitated by the infeasibility of exact solutions.

4.3 Global Optimization

Once the server received the synthetic set $\mathcal{S}_k$ and the calibrated radius r_k, global updates can be performed by conducting gradient descent directly on the synthetic set of samples. The global objective can be recovered by $\widehat{\mathcal{L}}_k(\mathbf{w})$ according to Eq. 7 (i.e., training on the synthetic samples), while the scaling factor $\frac{N_k}{N}$ can be treated as the scaling factor of the learning rate when computing the gradients on samples from each synthetic set $\mathcal{S}_k$, namely:

$$\begin{aligned} \mathbf{w}_g^{m,t+1} &= \mathbf{w}_g^{m,t} - \sum_{k=1}^{K} \eta \cdot \frac{N_k}{N}\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}_g^{m,t}, \mathcal{S}_k) \\ \text{s.t.} \quad &\|\mathbf{w}_g^{m,t} - \mathbf{w}_g^{m,1}\| \le \min\{r_k\}_{k=1}^{K} \end{aligned} \tag{14}$$

The constraint in Eq. 14 enforces that the global update respects the vicinity suggested by the clients, meaning updates are only made within regions where the approximation is sufficiently accurate.

Algorithm 2 FedLAP: Global Optimization

```
function ServerExecute:
    Initialize global weight $\mathbf{w}_g^{1,1}$, radius $r$
    /* Local approximation */
    for $m = 1, \ldots, M$ do
        for $k = 1, \ldots, K$ do
            $\mathcal{S}_k, r_k \leftarrow$ ClientExecute($k$, $r$, $\mathbf{w}_g^{m,1}$)
        end for
        /* Global optimization */
        $r_g \leftarrow \min\{r_k\}_{k=1}^{K}$
        $t \leftarrow 1$
        while $\|\mathbf{w}_g^{m,1} - \mathbf{w}_g^{m,t}\| < r_g$ do
            $\mathbf{w}_g^{m,t+1} = \mathbf{w}_g^{m,t} - \sum_{k=1}^{K} \eta \frac{N_k}{N} \nabla \mathcal{L}(\mathbf{w}_g^{m,t}, \mathcal{S}_k)$
            $t \leftarrow t + 1$
        end while
        $\mathbf{w}_g^{m+1,1} \leftarrow \mathbf{w}_g^{m,t}$
    end for
    Return: global model weight $\mathbf{w}_g^{M+1,1}$
```

Algorithm 3 FedLAP-DP

```
function ServerExecute:
    Initialize global weight $\mathbf{w}_g^{1,1}$, Fix the radius $r$
    /* Local approximation */
    for $m = 1, \ldots, M$ do
        for $k = 1, \ldots, K$ do
            $\mathcal{S}_k \leftarrow$ ClientsExecute($k$, $r$, $\mathbf{w}_g^{m,1}$)
        end for
        /* Global optimization */
        $t \leftarrow 1$
        while $\|\mathbf{w}_g^{m,1} - \mathbf{w}_g^{m,t}\| < r$ do
            $\mathbf{w}_g^{m,t+1} = \mathbf{w}_g^{m,t} - \sum_{k=1}^{K} \eta \frac{1}{K} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_g^{m,t}, \mathcal{S}_k)$
            $t \leftarrow t + 1$
        end while
        $\mathbf{w}_g^{m+1,1} \leftarrow \mathbf{w}_g^{m,t}$
    end for
    Return: global model weight $\mathbf{w}_g^{M+1,1}$

function ClientExecute($k$, $r$, $\mathbf{w}_g^{m,1}$) :
    Initialize $\mathcal{S}_k$: $\{\hat{\boldsymbol{x}}_k^m\}$ from Gaussian noise or $\{\hat{\boldsymbol{x}}_k^{m-1}\}$, $\{\hat{y}_k\}$ to
    be a balanced set
    for $i = 1, \ldots, R_i$ do
        /* Resample training trajectories */
        Reset $t \leftarrow 1$, model $\mathbf{w}_k^{m,1} \leftarrow \mathbf{w}_g^{m,1}$, and $\mathcal{S}_k^{i,0} \leftarrow \mathcal{S}_k^{i-1}$
        while $||\mathbf{w}_k^{m,t} - \mathbf{w}_k^{m,1}|| < r$ do
            Sample random batch $\{(\boldsymbol{x}_k^i, y_k^i)\}_{i=1}^{\mathbb{B}}$ from $\mathcal{D}_k$
            for each $(\boldsymbol{x}_k^i, y_k^i)$ do
                /* Compute per-example gradients on client data */
                $g^{\mathcal{D}}(\boldsymbol{x}_k^i) = \nabla_{\mathbf{w}} \ell(\mathbf{w}_k^{m,t}, \boldsymbol{x}_k^i, y_k^i)$
                /* Clip gradients with bound $\mathbb{C}$ */
                $\widetilde{g^{\mathcal{D}}}(\boldsymbol{x}_k^i) = g^{\mathcal{D}}(\boldsymbol{x}_k^i) \cdot \min(1, \mathbb{C}/\|g^{\mathcal{D}}(\boldsymbol{x}_k^i)\|_2)$
            end for
            /* Add noise to average gradient by Gaussian mechanism
            */
            $\nabla \widetilde{\mathcal{L}}(\mathbf{w}_k^{m,t}, \mathcal{D}_k) = \frac{1}{\mathbb{B}} \sum_{i=1}^{\mathbb{B}} (\widetilde{g^{\mathcal{D}}}(\boldsymbol{x}_k^i) + \mathcal{N}(0, \sigma^2 \mathbb{C}^2 I))$
            for $j = 1, \ldots, R_b$ do
                /* Update synthetic set $\mathcal{S}_k$ */
                $\mathcal{S}_k^{i,j+1} = \mathcal{S}_k^{i,j} - \tau \nabla_{\mathcal{S}_k} \mathcal{L}_{dis}\left(\nabla \widetilde{\mathcal{L}}(\mathbf{w}_k^{m,t}, \mathcal{D}_k), \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k)\right)$
            end for
            for $l = 1, \ldots, R_l$ do
                /* Update local model parameter $\mathbf{w}_k$ */
                $\mathbf{w}_k^{m,t+1} = \mathbf{w}_k^{m,t} - \eta \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k)$
                $t \leftarrow t + 1$
            end for
        end while
    end for
    Return: Synthetic set $\mathcal{S}_k^{R_i}$
```

4.4 Record-level DP

While federated systems offer a basic level of privacy protection, recent works identify various vulnerabilities under the existing framework, such as membership inference [37, 40]. Though Dong et al. [8] uncovers that distilled datasets may naturally introduce privacy protection, we further address possible privacy concerns that might arise during the transfer of synthetic data in our proposed method. Specifically, we rigorously limit privacy leakage by integrating record-level DP, a privacy notion widely used in FL applications. This is especially important in cross-silo scenarios, such as collaborations between hospitals, where each institution acts as a client, aiming to train a predictive model and leveraging patient data with varying distributions across different hospitals while ensuring strict privacy protection for patients.

Threat Model. In a federated system, there can be one or multiple colluding adversaries who have access to update vectors from any party during each communication round. These adversaries may have unlimited computation power but remain passive or "honest-but-curious," meaning they follow the learning protocol faithfully without modifying any update vectors [21, 44, 47]. These adversaries can represent any party involved, such as a malicious client or server, aiming to extract information from other parties. The central server possesses knowledge of label classes for each client's data, while clients may or may not know the label classes of other clients' data. While we typically do not intentionally hide label class information among clients, our approach is flexible and can handle scenarios where clients want to keep their label class information confidential from others.

We integrate record-level DP into FedLAP to provide theoretical privacy guarantees, which yields FedLAP-DP. Given a desired privacy budget (ε, δ), we clip the gradients derived from real data with the Gaussian mechanism, denoted by $\nabla \widetilde{\mathcal{L}}(\mathbf{w}_k^{m,t}, \mathcal{D}_k)$. The DP-guaranteed local approximation can be realized by replacing the learning target of Eq. 11 with the gradients processed by DP while leaving other constraints the same. Formally, we have

$$\begin{aligned} \arg\min_{\mathcal{S}_k} \; & \sum_{t=1}^{T} \mathcal{L}_{\text{dis}}\left(\nabla \widetilde{\mathcal{L}}(\mathbf{w}_k^{m,t}, \mathcal{D}_k), \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k)\right) \\ \text{s.t.} \quad & \|\mathbf{w}_k^{m,t} - \mathbf{w}_k^{m,1}\| < r, \\ & \mathbf{w}_k^{m,t+1} = \mathbf{w}_k^{m,t} - \eta \nabla \mathcal{L}(\mathbf{w}_k^{m,t}, \mathcal{S}_k) \end{aligned} \tag{15}$$

During the server optimization, we do not use a data-dependent strategy to decide r. Instead, we use the same radius as adopted for local approximation (c.f. Eq. 14).

$$\begin{aligned} \mathbf{w}_g^{m,t+1} &= \mathbf{w}_g^{m,t} - \sum_{k=1}^{K} \eta \cdot \frac{N_k}{N} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}_g^{m,t}, \mathcal{S}_k) \\ \text{s.t.} \quad & \|\mathbf{w}_g^{m,t} - \mathbf{w}_g^{m,1}\| \leq r \end{aligned} \tag{16}$$

We describe the full algorithm of FedLAP-DP in Algorithm 3 and present the privacy analysis of FedLAP-DP in the Sec. 5. Our analysis suggests that with equivalent access to private data, FedLAP incurs the same privacy costs as gradient-sharing approaches. Our method further demonstrates a better privacy-utility trade-off in Sec. 6.3, confirming its robustness under DP noise.

5 PRIVACY ANALYSIS

5.1 Definitions

Definition 5.1 (Rényi divergence). Let P and Q be two distributions defined over the same probability space $\mathcal{X}$. Let their respective densities be denoted as p and q. The Rényi divergence, of a finite order $\alpha \neq 1$, between the distributions P and Q is defined as follows:

$$D_\alpha (P \parallel Q) \overset{\Delta}{=} \frac{1}{\alpha - 1} \ln \int_{\mathcal{X}} q(x) \left(\frac{p(x)}{q(x)} \right)^{\alpha} dx \, .$$

Rényi divergence at orders $\alpha = 1, \infty$ are defined by continuity.

Definition 5.2 (Rényi differential privacy (RDP) [38]). A randomized mechanism $\mathcal{M} : \mathcal{E} \rightarrow \mathcal{R}$ satisfies (α, ρ)-Rényi differential privacy (RDP) if for any two adjacent inputs $E, E' \in \mathcal{E}$ it holds that

$$D_\alpha \left(\mathcal{M}(E) \parallel \mathcal{M}(E') \right) \leq \rho$$

In this work, we call two datasets E, E' to be adjacent if $E' = E \cup \{\boldsymbol{x}\}$ (or vice versa).

Definition 5.3 (Sampled Gaussian Mechanism (SGM) [1, 39]). Let f be an arbitrary function mapping subsets of $\mathcal{E}$ to $\mathbb{R}^d$. We define the Sampled Gaussian mechanism (SGM) parametrized with the sampling rate $0 < q \leq 1$ and the noise $\sigma > 0$ as

$$\mathrm{SG}_{q,\sigma} \overset{\Delta}{=} f\left(\{\boldsymbol{x} : \boldsymbol{x} \in E \text{ is sampled with probability } q\}\right) + \mathcal{N}(0, \sigma^2 \mathbb{I}_d),$$

where each element of E is independently and randomly sampled with probability q without replacement.

The sampled Gaussian mechanism consists of adding i.i.d Gaussian noise with zero mean and variance σ^2 to each coordinate of the true output of f. In fact, the sampled Gaussian mechanism draws random vector values from a multivariate isotropic Gaussian distribution denoted by $\mathcal{N}(0, \sigma^2 \mathbb{I}_d)$, where d is omitted if it is unambiguous in the given context.

5.2 Analysis

The privacy analysis of FedLAP-DP and other DP baselines follows the analysis framework used for gradient-based record level-DP methods in FL [21, 44, 47]. In this framework, each individual local update is performed as a single SGM (Definition 5.3) that involves clipping the per-example gradients on a local batch and subsequently adding Gaussian noise to the averaged batch gradient (Algorithm 3). The privacy cost accumulated over multiple local updates and global rounds is then quantified utilizing the revisited moment accountant [39], which presents an adapted version of the moments accountant introduced in Abadi et al. [1] by adapting to the notion of RDP (Definition 5.2). Finally, to obtain interpretable results and enable transparent comparisons to established approaches, we convert the privacy cost from (α, ρ)-RDP to (ε, δ)-DP by employing Theorem 5.7 provided by [3].

Theorem 5.4. *(Mironov et al. [39]). Let* $\mathrm{SG}_{q,\sigma}$ *be the Sampled Gaussian mechanism for some function* f *with* $\Delta_2 f \leq 1$ *for any adjacent* $E, E' \in \mathcal{E}$. *Then* $\mathrm{SG}_{q,\sigma}$ *satisfies* (α, ρ)*-RDP if*

$$\begin{aligned} &\rho \leq D_\alpha \Big(\mathcal{N}(0, \sigma^2) \, \Big\| \, (1-q)\mathcal{N}(0, \sigma^2) + q\mathcal{N}(1, \sigma^2) \Big) \\ \textit{and} \quad &\rho \leq D_\alpha \Big((1-q)\mathcal{N}(0, \sigma^2) + q\mathcal{N}(1, \sigma^2) \, \Big\| \, \mathcal{N}(0, \sigma^2) \Big) \end{aligned}$$

Theorem 5.4 reduce the problem of proving the RDP bound for $\mathrm{SG}_{q,\sigma}$ to a simple special case of a mixture of one-dimensional Gaussians.

Theorem 5.5. *[39]. Let* μ_0 *denote the* pdf *of* $\mathcal{N}(0, \sigma^2)$, μ_1 *denote the* pdf *of* $\mathcal{N}(1, \sigma^2)$, *and let* μ *be the mixture of two Gaussians* $\mu = (1-q)\mu_0 + q\mu_1$. *Let* $\mathrm{SG}_{q,\sigma}$ *be the Sampled Gaussian mechanism for some function* f *and under the assumption* $\Delta_2 f \leq 1$ *for any adjacent* $E, E' \in \mathcal{E}$. *Then* $\mathrm{SG}_{q,\sigma}$ *satisfies* (α, ρ)*-RDP if*

$$\rho \leq \frac{1}{\alpha - 1} \log\left(\max\{A_\alpha, B_\alpha\}\right) \tag{17}$$

where $A_\alpha \overset{\Delta}{=} \mathbb{E}_{z \sim \mu_0}[(\mu(z)/\mu_0(z))^\alpha]$ *and* $B_\alpha \overset{\Delta}{=} \mathbb{E}_{z \sim \mu}[(\mu_0(z)/\mu(z))^\alpha]$

Theorem 5.5 states that applying SGM to a function of sensitivity (Eq. 5) at most 1 (which also holds for larger values without loss of generality) satisfies (α, ρ)-RDP if $\rho \leq \frac{1}{\alpha-1} \log(\max\{A_\alpha, B_\alpha\})$. Thus, analyzing RDP properties of SGM is equivalent to upper bounding A_α and B_α.

From Corollary 7 in Mironov et al. [39], $A_\alpha \geq B_\alpha$ for any $\alpha \geq 1$. Therefore, we can reformulate 17 as

$$\rho \leq \xi_{\mathcal{N}}(\alpha | q) := \frac{1}{\alpha - 1} \log A_\alpha \tag{18}$$

To compute A_α, we use the numerically stable computation approach proposed in Section 3.3 of Mironov et al. [39]. The specific approach used depends on whether α is expressed as an integer or a real value.

Theorem 5.6 (Composability [38]). *Suppose that a mechanism* $\mathcal{M}$ *consists of a sequence of adaptive mechanisms* $\mathcal{M}_1, \ldots, \mathcal{M}_k$ *where* $\mathcal{M}_i : \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{E} \rightarrow \mathcal{R}_i$. *If all mechanisms in the sequence are* (α, ρ)*-RDP, then the composition of the sequence is* $(\alpha, k\rho)$*-RDP.*

In particular, Theorem 5.6 holds when the mechanisms themselves are chosen based on the (public) output of the previous mechanisms. By Theorem 5.6, it suffices to compute $\xi_{\mathcal{N}}(\alpha|q)$ at each step and sum them up to bound the overall RDP privacy budget of an iterative mechanism composed of single DP mechanisms at each step.

Theorem 5.7 (Conversion from RDP to DP [3]). *If a mechanism* $\mathcal{M}$ *is* (α, ρ)*-RDP then it is* $((\rho + \log((\alpha-1)/\alpha) - (\log\delta + \log\alpha)/(\alpha-1), \delta)$*-DP for any* $0 < \delta < 1$.

	DSC†	FedSGD (1×)	FedAvg (1×)	FedProx (1×)	SCAFFOLD (2×)	FedDM (0.96×)	Ours (0.96×)
MNIST	98.90±0.20	87.07±0.65	96.55±0.21	96.26±0.04	97.56±0.06	96.66±0.18	**98.08±0.02**
Fa.MNIST	83.60±0.40	75.10±0.16	79.67±0.56	79.37±0.29	82.17±0.37	83.10±0.16	**87.37±0.09**
CIFAR-10	53.90±0.50	60.91±0.19	**75.20±0.12**	63.84±0.45	56.27±1.19	70.51±0.45	71.91±0.20

Table 1: Performance comparison on benchmark datasets. The relative communication cost of each method (w.r.t. the model size) is shown in brackets. DSC† is ported from the original paper and conducted in a one-shot centralized setting.

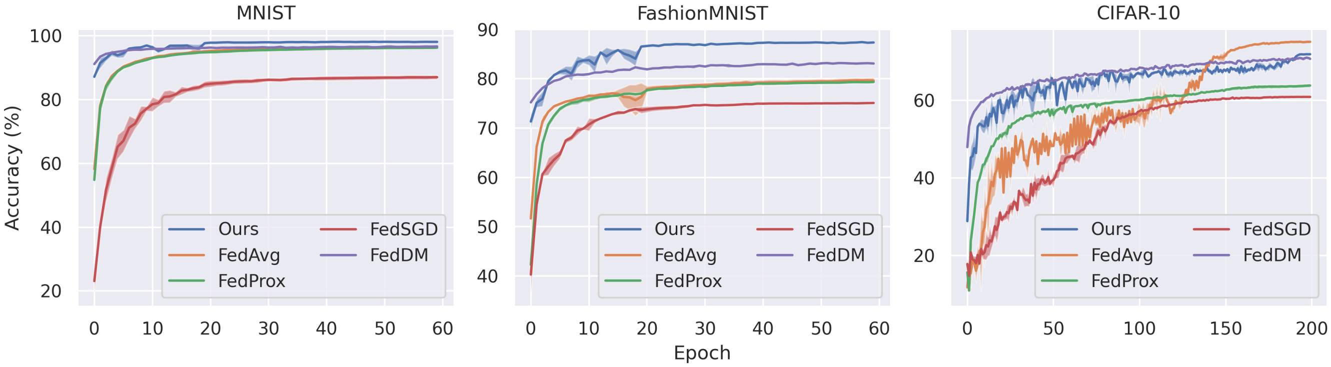


Figure 3: Accuracy over communication rounds with extremely non-IID data.

Theorem 5.8 (Privacy of FedLAP-DP). *For any $0 < \delta < 1$ and $\alpha \geq 1$, FedLAP-DP is (ε, δ)-DP, with*

$$\varepsilon = \min_{\alpha}\Big(M \cdot \xi_{\mathcal{N}}(\alpha|q_1) + M(T-1) \cdot \xi_{\mathcal{N}}(\alpha|q_2) + \log((\alpha-1)/\alpha) - (\log\delta + \log\alpha)/(\alpha-1)\Big) \quad (19)$$

Here, $\xi_{\mathcal{N}}(\alpha|q)$ is defined in Eq. 18, $q_1 = \frac{C\cdot\mathbb{B}}{\min_k |\mathcal{D}_k|}$, $q_2 = \frac{\mathbb{B}}{\min_k |\mathcal{D}_k|}$, M is the number of federated rounds, T is the total number of local updates (i.e., total accesses to local private data) per federated round, C is the probability of selecting any client per federated round, $\mathbb{B}$ is the local batch size, and $|\mathcal{D}_k|$ denotes the local dataset size.

The proof follows from Theorems 5.5, 5.6,5.7 and the fact that a record is sampled in the very first SGD iteration of every round if two conditions are met. First, the corresponding client must be selected, which occurs with a probability of C. Second, the locally sampled batch at that client must contain the record, which has a probability of at most $\frac{\mathbb{B}}{\min_k |\mathcal{D}_k|}$. However, the adaptive composition of consecutive SGD iterations are considered where the output of a single iteration depends on the output of the previous iterations. Therefore, the sampling probability for the first batch is $q_1 = \frac{C\cdot\mathbb{B}}{\min_k |\mathcal{D}_k|}$, while the sampling probability for every subsequent SGD iteration within the same round is at most $q_2 = \frac{\mathbb{B}}{\min_k |\mathcal{D}_k|}$ *conditioned* on the result of the first iteration [21].

6 EXPERIMENTS

6.1 Setup

Overview. We consider a standard classification task by training federated ConvNets [27] on three benchmark datasets: MNIST [26], FashionMNIST [54], and CIFAR-10 [24]. Our study focuses on a non-IID setting where five clients possess disjoint class sets, meaning each client holds two unique classes. This scenario is typically considered challenging [16] and mirrors the cross-silo setting [18] where all clients participate in every training round while maintaining a relatively large amount of data, yet exhibiting statistical divergence (e.g., envision the practical scenario for collaborations among hospitals). Our method employs a learning rate of 100 for updating synthetic images and 0.1 with cosine decay for model updates. We set by default $(R_i, R_l, R_b, r) = (4, 2, 10, 1.5)$ and $(1, 0, 5, 10)$ for DP and non-DP training, respectively. Additionally, we include a weight of 0.1 for Mean Squared Error (MSE) regularization in our method (Eq. 11). To prevent infinite loops caused by the neighborhood search, we upper bound the while loops in Algorithm 1 by 5 iterations. We follow FL benchmarks [36, 45] and the official codes for training the baselines. All experiments are repeated over three random seeds. In this work, we only consider a balanced setting where every client owns the same amount of training samples. We acknowledge that minor sample imbalances can be managed by existing federated algorithms via modifying the aggregation weights, for example, N_k/N as indicated in Eq. 7. However, severe imbalances present a significant challenge that might affect the efficacy of both our method and gradient-sharing techniques, and addressing this issue falls beyond the purview of the current study.

Architecture. We provide the details of the federated ConvNet used in our paper. The network consists of three convolutional layers, followed by two fully-connected layers. ReLU activation functions are applied between each layer. Each convolution layer, except for the input layer, is composed of 128 (input channels) and 128 (output channels) with 3×3 filters. Following prior work [5, 45, 50, 60], we attach Group Normalization [53] before the activation functions to stabilize federated training. For classification, the network utilizes a global average pooling layer to extract features, which are then fed into the final classification layer for prediction. The entire

	PSG† [7]	DP-FedAvg	DP-FedProx	Ours	DP-FedAvg	DP-FedProx	Ours
ε	32	2.79			10.18		
MNIST	88.34±0.8	45.25±6.9	54.58±4.9	**60.72±1.3**	86.99±0.5	**88.75±0.5**	87.77±0.8
FMNIST	67.91±0.3	50.11±4.2	54.57±2.9	**59.85±1.5**	72.78±1.3	71.67±2.2	**73.00±0.7**
CIFAR-10	34.58±0.4	17.11±0.7	19.40±0.7	**21.42±1.4**	31.15±0.4	35.04±1.1	**36.09±0.5**

Table 2: Utility and Privacy budgets at varying privacy regimes. The high privacy regime with $\varepsilon = 2.79$ corresponds to the first communication round, while a privacy level of $\varepsilon = 10.18$ represents the commonly considered point (ε=10) in private learning literature. PSG† corresponds to a one-shot centralized setting and is reproduced from the official code with the default configuration that yields $\varepsilon = 32$ in federated settings.

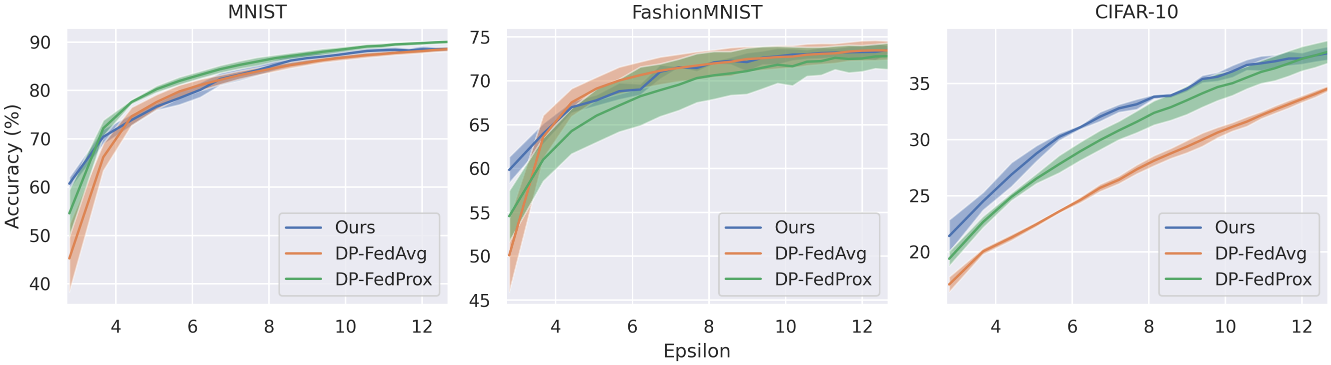


Figure 4: Privacy-utility trade-off with $\delta = 10^{-5}$. A *smaller* value of ε (x-axis) indicates a *stronger* privacy guarantee. Evaluation is conducted at each communication round.

network contains a total of 317,706 floating-point parameters. The model details are listed in the appendix.

6.2 Data Heterogeneity

We first demonstrate the effectiveness of FedLAP over various baselines on benchmark datasets in a non-IID setting. Our method assigns 50 images to each class, resulting in comparable communication costs to the baselines. The baselines include: **DSC** [62], the dataset distillation method considering centralized one-shot distillation; **FedSGD** [36], that transmits every single batch gradient to prevent potential model drifting; **FedAvg** [36], the most representative FL method; **FedProx** [31], **SCAFFOLD** [19], state-of-the-art federated optimization for non-IID distributions, and **FedDM** [55], a concurrent work that shares a similar idea but without considering approximation quality. Note that DSC operates in a (one-shot) centralized setting, SCAFFOLD incurs double the communication costs compared to the others by design, and FedDM requires class-wise optimization. As depicted in Table 1, our method surpasses DSC and FedSGD, highlighting the benefits of multi-round training on the server and client sides, respectively. Moreover, our method presents superior performance over state-of-the-art optimization methods, validating the strength of optimizing from a global view. We also plot model utility over training rounds in Fig. 3, where our method consistently exhibits the fastest convergence across three datasets. In other words, our methods consume fewer costs to achieve the same or better performance level and more communication efficiency.

6.3 Privacy Protection

We evaluate the trade-off between utility and privacy costs ε on benchmark datasets against two state-of-the-art methods, DP-FedAvg (the local DP version in Truex et al. [47]) and DP-FedProx. Note that FedDM [55] is incomparable since it considers class-wise optimization, introducing additional privacy risks and a distinct privacy notion. Our method assigns 10 images per class and is evaluated under the worst-case scenario, i.e., we assume the maximum of 5 while loops is always reached (Sec. 6.1) for the ε computation, despite the potential early termination (and thus smaller ε) caused by the radius r (Eq. 12). To ensure a fair and transparent comparison, we require our method to access the same amount of private data as the baselines in every communication round and consider a noise scale $\sigma = 1$ for all approaches. Fig. 4 demonstrates that our framework generally exhibits superior performance, notably with smaller ε and more complex dataset such as CIFAR-10. This superiority is further quantified in Table 1 under two typical privacy budgets of 2.79 and 10.18. Moreover, when compared to the private one-shot dataset condensation method (**PSG** [7]), our approach presents a better privacy-utility trade-off, effectively leveraging the benefits of multi-round training in the challenging federated setting.

6.4 Ablation Study

Radius Selection. As in Sec. 4, we assess various server optimization radius selection strategies: *Fixed*, *Max*, *Median*, and *Min*. The *Fixed* strategy employs a static length of 100 iterations regardless of quality, *Max* pursues swift optimization with the largest radius, *Median* moderates by adhering to the majority, and *Min*—used in

Min	Max	Median	Fixed
71.90	72.39	72.33	71.26

Table 3: Performance comparison between radius selection strategies.

	5%	10%	50%	100%
FedAvg	55.16 (-20.04)	61.43 (-13.77)	71.34 (-3.86)	75.20
Ours	56.34 (-15.57)	62.76 (-9.15)	68.97 (-2.94)	71.91

Table 4: Impact of different training set sizes (5%, 10%, 50%, and 100% of the original dataset): Parenthesized values indicate the reduction in performance when compared to utilizing the full training set (100%)

all experiments—targets the safest region agreed by all synthetic image sets. Table 3 shows that strategies mindful of approximation quality surpass the fixed approach. Detailed analysis in Sec. A.2 reveals that aggressive strategies yield inferior intermediate performance, unsuitable for federated applications needing satisfactory intermediate results. Among the strategies, *Min* proves optimal.

Size of Synthetic Datasets. We investigate the impact of synthetic dataset size on approximation. In general, higher numbers of synthetic samples submitted by clients lead to greater information communication. To further explore this concept, we conducted experiments on CIFAR-10, building upon the previous experiment (shown in Fig. 3) by adding five additional settings in which we assigned 10, 20, 100, 150, and 200 images to each class (referred to as "image per class" or #ipc). Our results, presented in Fig. 5, demonstrate that our method performs best when assigning 200 images, supporting the hypothesis that more synthetic samples convey more information. Additionally, our method produces superior outcomes regardless of #ipc when communication costs are restricted, making it advantageous for resource-constrained devices.

Size of Training Sets. In this experiment, we show that given the same amount of optimization steps, our method can approximate the information more faithfully as the number of training examples that each client can access decreases. Table 4 presents the impact of varying training set sizes. We sub-sample 5%, 10%, 50%, and 100% of samples from the original CIFAR-10 dataset and repeat the same experiments as in Table 1. In other words, each client has 500, 1000, and 5000 images, respectively. It is observed that the performance of both methods degrades as the number of training samples decreases. Our method demonstrates greater resilience to reductions in training sample size than FedAvg by exhibiting less performance degradation when compared to using the full training set. A potential explanation is that a smaller sample size may be simpler to approximate given the same number of optimization steps.

6.5 Qualitative Results

We visualize the synthetic images generated at different training stages in both DP and non-DP settings. Specifically, we consider the same setting as in Sec. 6.2 and Sec. 6.3 on CIFAR-10. We randomly sample synthetic images associated with classes *airplane*, *automobile*, *bird*, and *cat* at epochs 5, 50, and 196. Fig. 6 shows that the images synthesized by our approach look drastically different from real images even in non-DP settings. Most of the details have been obfuscated to carry essential gradient information. Notably, a recent work [8] arguably suggests that dataset distillation may naturally introduce privacy protection, making models more robust than the ones trained by plain gradients.

On the other hand, Fig. 7 visualizes the synthetic images generated under the DP protection of $\varepsilon = \{6.72, 10.93, 14.30\}$. It is observed that even with the loosest privacy budgets ($\varepsilon = 14.30$), the images tend to look like random noise and significantly obfuscate most of the visual contents compared to Fig. 6. The protection gets further stronger as the privacy budget decreases. These visualizations verify the effectiveness of our approach in introducing record-level DP and might offer a new tool to visually examine the privacy protection of the communicated information.

6.6 Privacy Auditing

Despite the theoretical privacy protection introduced by DP, we examine the empirical privacy protection by conducting membership inference attacks [25] on both our method and FedAvg in DP and non-DP scenarios, respectively. In particular, we consider two attack settings. We first divide the testing set (i.e., non-members) into two disjoint partitions and use one of them for training attack models while using the other as the audit dataset. In the first setting, we initially collect the loss values of the examples in a black-box manner, which are later used to train a neural network to distinguish whether a given example belongs to the training set. On the other hand, the second setting adopts the gradient norms of the examples, providing more detailed information for membership inference attacks. The access of gradient norms demands a white-box setting, i.e., assuming the model parameters are accessible.

We present the Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC) scores for both scenarios in Figures 8 and 9. The diagonal lines present the baseline performance of random guesses, which are characterized by AUC scores of 0.5. Methods that perform closer to random guesses provide better privacy protection since the adversaries hardly infer information from the victim models. It is observed that methods without DP protection are vulnerable to membership inference attacks. Notably, our methods are more robust than gradient-sharing schemes, even without DP protection. The resilience may stem from the approximation process that further limits information leakage. In contrast, the methods with DP protection consistently produce AUC scores around 0.5 regardless of attack settings, verifying the effectiveness of DP protection.

7 DISCUSSION

Future Directions. While the primary contribution of FedLAP-DP lies in utilizing local approximation for global optimization, we demonstrate in the appendix that its performance can be further enhanced by improving the quality of the approximation. Moreover, ongoing research in synthetic data generation [5, 60, 62] represents a potential avenue for future work, which could potentially benefit our formulation. The potential directions include explicitly

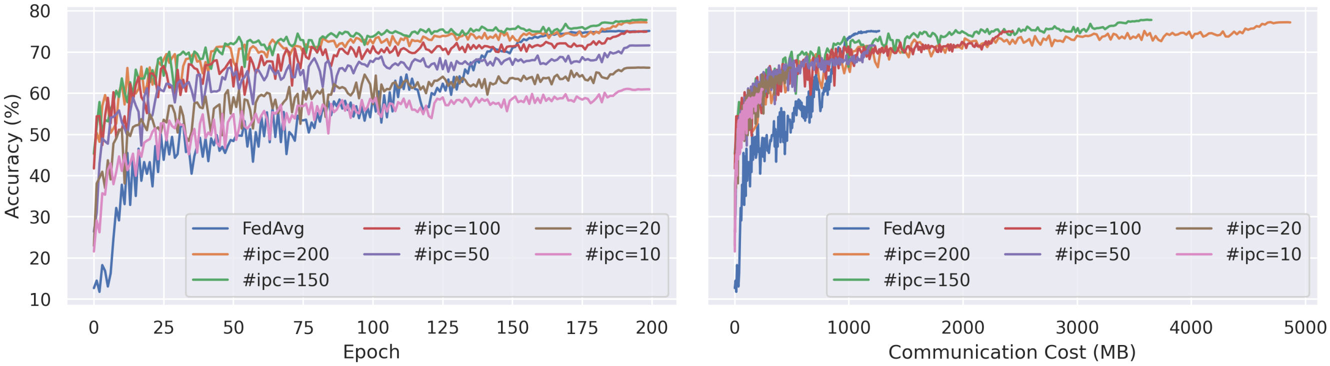


Figure 5: Ablation study on the number of images per class (#ipc).

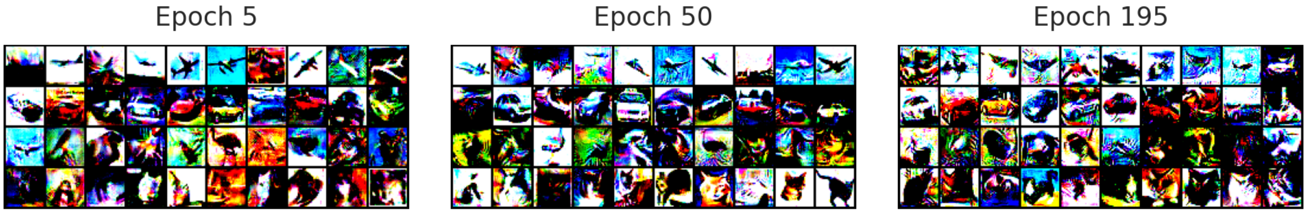


Figure 6: Visualization of synthetic images at epochs 5, 50, and 196 in the non-private CIFAR-10 experiment. The pixel values are clipped to the range $[0, 1]$. Each row corresponds to airplane, automobile, bird, and cat, respectively.

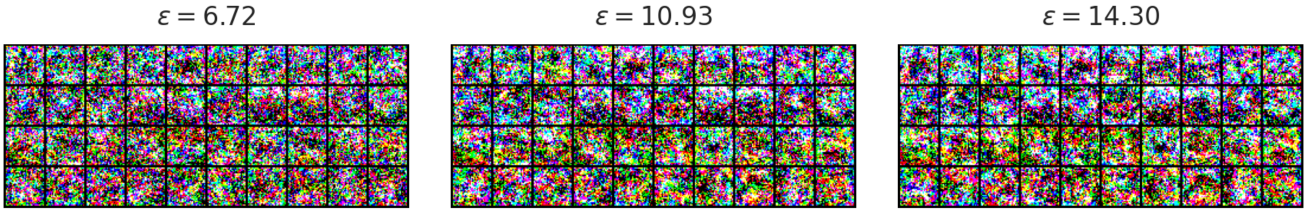
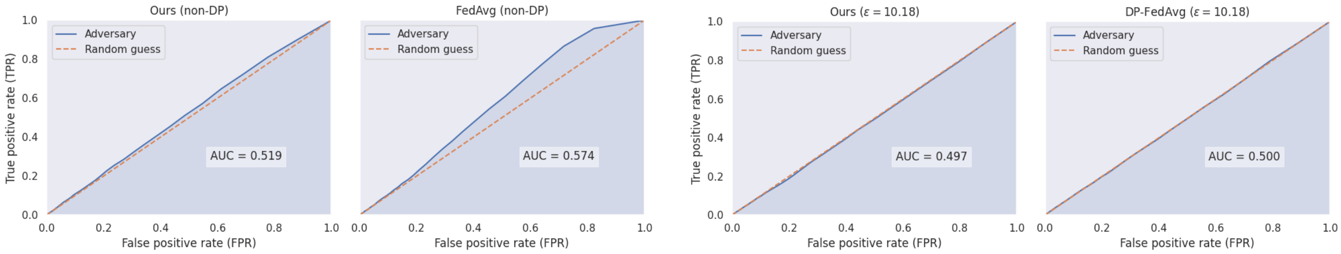


Figure 7: Visualization of synthetic images for $\varepsilon = 6.72$, 10.93, and 14.30 in the privacy-preserving CIFAR-10 experiment. The pixel values are clipped to the range $[0, 1]$. Each row corresponds to airplane, automobile, bird, and cat, respectively.

matching training trajectories [5] or leveraging off-shelf foundation models [6]. Moreover, we also observe that given the same amount of optimization steps, the performance of our method may decrease when the number of classes increases on clients. For instance, the performance on CIFAR-10 in an IID scenario drops from 71.91 to 62.52 while FedAvg remains at a similar level of 73.5. This suggests that if there is no performance degradation caused by data heterogeneity, our method loses information during image synthesis. Future works that improve the efficiency of approximation could further bridge such gap and enable more efficient federated learning under extremely non-IID scenarios.

Computation Overhead. Our method suggests an alternative to current research, trading computation for improved performance and communication costs incurred by slow convergence and biased optimization. We empirically measured the computation time needed for a communication round by a client. We observed an increase from 0.5 minutes (FedAvg) to 2.5 minutes (FedLAP) using one NVIDIA Titan X. Despite the increase, the computation time is still manageable in cross-silo environments, where participants are deemed to have more computation power. A thorough analysis can be found in Sec. B. We anticipate this work will motivate the community to further explore the trade-off between computation and communication resource demands beyond local epochs, as we have demonstrated in Fig. 5.

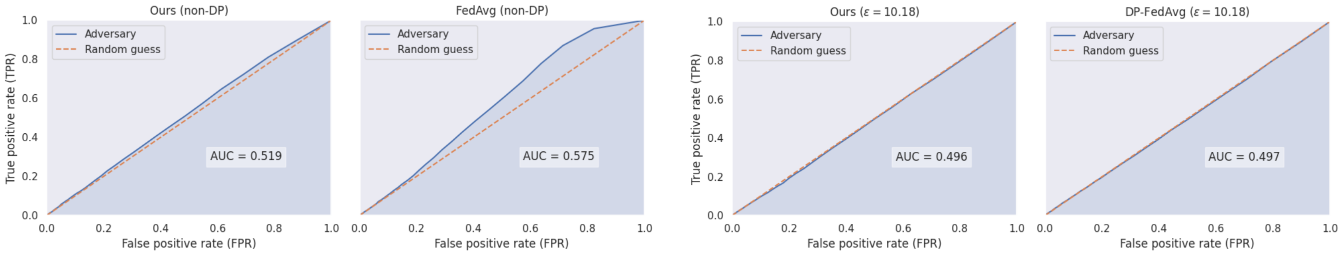
(Visual) Privacy. Dong et al. [8] were among the first to show that data distillation may naturally offer superior privacy protections compared to traditional gradient information in centralized settings. In Sec. 6.6 of our work, we extend this concept into federated environments, although our synthetic images are not intended to resemble realistic or class-specific content. Despite these advances, a comprehensive analysis of existing dataset distillation approaches in terms of privacy remains pending. This necessitates further



(a) non-DP

(b) DP ($\varepsilon = 10.18$)

Figure 8: Black-box membership inference attacks based on loss values targeting FedLAP (ours) and FedAvg under non-DP and DP ($\varepsilon = 10.18$) settings.



(a) non-DP

(b) DP ($\varepsilon = 10.18$)

Figure 9: White-box membership inference attacks based on gradient norms targeting FedLAP (ours) and FedAvg under non-DP and DP ($\varepsilon = 10.18$) settings.

exploration in diverse settings, including but not limited to, defense against reconstruction attacks and membership inference, as well as an in-depth privacy comparison with non-DP gradient-sharing techniques. Beyond the realm of membership privacy, safeguarding visual privacy—the discernible content within images—remains a complex issue [42]. While several strategies like adversarial entropy maximization [49] and image masking [43] have been explored, finding the right balance between utility and privacy varies with the use case and presents opportunities for enhancement.

8 CONCLUSION

In conclusion, this work introduces FedLAP-DP, a novel approach for privacy-preserving federated learning. FedLAP-DP utilizes synthetic data to approximate local loss landscapes within calibrated trust regions, effectively debiasing the optimization on the server. Moreover, our method seamlessly integrates record-level differential privacy, ensuring strict privacy protection for individual data records. Extensive experimental results demonstrate that FedLAP-DP outperforms gradient-sharing approaches in terms of faster convergence on highly-skewed data splits and reliable utility under differential privacy settings. We further explore the critical role of radius selection, the influence of synthetic dataset size, open directions, and potential enhancements to our work. Overall, FedLAP-DP presents a promising approach for privacy-preserving federated learning, addressing the challenges of convergence stability and privacy protection in non-IID scenarios.

ACKNOWLEDGMENTS

This work is partially funded by the Helmholtz Association within the project "Trustworthy Federated Data Analytics (TFDA)" (ZT-I-OO1 4), Medizininformatik-Plattform "Privatsphären-schutzende Analytik in der Medizin" (PrivateAIM), grant No. 01ZZ2316G, and Bundesministeriums fur Bildung und Forschung (PriSyn), grant No. 16KISAO29K. The work is also supported by ELSA – European Lighthouse on Secure and Safe AI funded by the European Union under grant agreement No. 101070617. Moreover, The computation resources used in this work are supported by the Helmholtz Association's Initiative and Networking Fund on the HAICORE@FZJ partition. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or European Commission. Neither the European Union nor the European Commission can be held responsible for them. Dingfan Chen was partially supported by Qualcomm Innovation Fellowship Europe.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[2] Nasser Aldaghri, Hessam Mahdavifar, and Ahmad Beirami. 2021. FeO2: Federated Learning with Opt-Out Differential Privacy. *CoRR* (2021).

[3] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. 2020. Hypothesis testing interpretations and renyi differential privacy. In *International Conference on Artificial Intelligence and Statistics.*

[4] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. 2018. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984* (2018).

[5] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. 2023. Generalizing Dataset Distillation via Deep Generative Prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. 2022. Private Set Generation with Discriminative Information. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[8] Tian Dong, Bo Zhao, and Lingjuan Lyu. 2022. Privacy for free: How does dataset condensation help privacy?. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[9] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* (2014).

[10] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[11] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. 2022. Label Inference Attacks Against Vertical Federated Learning. In *31st USENIX Security Symposium (USENIX Security 22)*.

[12] Filippo Galli, Kangsoo Jung, Sayan Biswas, Catuscia Palamidessi, and Tommaso Cucinotta. 2023. Advancing Personalized Federated Learning: Group Privacy, Fairness, and Beyond. *SN Computer Science* (2023).

[13] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[14] Yang He, Hui-Po Wang, and M Fritz. 2021. Cossgd: Communicationefficient federated learning with a simple cosine-based quantization. In *1st NeurIPS Workshop on New Frontiers in Federated Learning (NFFL)*.

[15] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*.

[16] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335* (2019).

[17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022.* OpenReview.net.

[18] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* (2021).

[19] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[20] Raouf Kerkouche, Gergely Ács, Claude Castelluccia, and Pierre Genevès. 2021. Constrained differentially private federated learning for low-bandwidth devices. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (Proceedings of Machine Learning Research)*, Cassio de Campos and Marloes H. Maathuis (Eds.). PMLR, 1756–1765.

[21] Raouf Kerkouche, Gergely Acs, Claude Castelluccia, and Pierre Genevès. 2021. Privacy-preserving and bandwidth-efficient federated learning: An application to in-hospital mortality prediction. In *Proceedings of the Conference on Health, Inference, and Learning.*

[22] Raouf Kerkouche, Gergely Ács, Claude Castelluccia, and Pierre Genevès. 2021. Compression Boosts Differentially Private Federated Learning. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*.

[23] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. 2019. First analysis of local gd on heterogeneous data. *arXiv preprint arXiv:1909.04715* (2019).

[24] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[25] Sasi Kumar and Reza Shokri. 2020. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998).

[27] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. 2010. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems.*

[28] Daliang Li and Junpu Wang. 2019. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581* (2019).

[29] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. 2020. Label leakage and protection in two-party split learning. *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning (SpicyFL)* (2020).

[30] Qinbin Li, Bingsheng He, and Dawn Song. 2021. Model-contrastive federated learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[31] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* (2020).

[32] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. 2019. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189* (2019).

[33] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. 2021. FedBN: Federated Learning on Non-IID Features via Local Batch Normalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[34] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. 2021. Projected federated averaging with heterogeneous differential privacy. *Proceedings of the VLDB Endowment* (2021).

[35] Mi Luo, Fei Chen, Dapeng Hu, Yifan Zhang, Jian Liang, and Jiashi Feng. 2021. No fear of heterogeneity: Classifier calibration for federated learning with non-iid data. *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[36] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[37] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*.

[38] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*.

[39] Ilya Mironov, Kunal Talwar, and Li Zhang. 2019. R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530* (2019).

[40] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*.

[41] Maxence Noble, Aurélien Bellet, and Aymeric Dieuleveut. 2022. Differentially private federated learning on heterogeneous data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 10110–10145.

[42] José Ramón Padilla-López, Alexandros Andre Chaaraoui, and Francisco Flórez-Revuelta. 2015. Visual privacy protection methods: A survey. *Expert Systems with Applications* (2015).

[43] JithendraK Paruchuri, Sen-chingS Cheung, and MichaelW Hail. 2009. Video data hiding for managing privacy information in surveillance systems. *EURASIP Journal on Information Security* (2009).

[44] Daniel Peterson, Pallika Kanani, and Virendra J Marathe. 2019. Private federated learning with domain adaptation. *arXiv preprint arXiv:1912.06733* (2019).

[45] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. 2021. Adaptive Federated Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[46] Marlon Tobaben, Aliaksandra Shysheya, John Bronskill, Andrew Paverd, Shruti Tople, Santiago Zanella Béguelin, Richard E. Turner, and Antti Honkela. 2023. On the Efficacy of Differentially Private Few-shot Image Classification. *CoRR* (2023).

[47] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security.*

[48] Aidmar Wainakh, Fabrizio Ventola, Till Müßig, Jens Keim, Carlos Garcia Cordero, Ephraim Zimmer, Tim Grube, Kristian Kersting, and Max Mühlhäuser. 2022. User-Level Label Leakage from Gradients in Federated Learning. *Proceedings on Privacy Enhancing Technologies* (2022).

[49] Hui-Po Wang, Tribhuvanesh Orekondy, and Mario Fritz. 2021. Infoscrub: Towards attribute privacy by targeted obfuscation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

[50] Hui-Po Wang, Sebastian Stich, Yang He, and Mario Fritz. 2022. ProgFed: effective, communication, and computation efficient federated learning by progressive training. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[51] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems (NeurIPS)* (2020).

[52] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018. Dataset distillation. *arXiv preprint arXiv:1811.10959* (2018).

[53] Yuxin Wu and Kaiming He. 2018. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[54] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).

[55] Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. 2023. Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[56] Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. 2023. {PrivateFL}: Accurate, Differentially Private Federated Learning via Personalized Data Transformation. In *32nd USENIX Security Symposium (USENIX Security 23)*.

[57] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[58] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. 2022. Differentially Private Fine-tuning of Language Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[59] Bo Zhao and Hakan Bilen. 2021. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*.

[60] Bo Zhao and Hakan Bilen. 2023. Dataset condensation with distribution matching. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[61] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. idlg: Improved deep leakage from gradients. *arXiv preprint arXiv:2001.02610* (2020).

[62] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset Condensation with Gradient Matching. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[63] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).

[64] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in Neural Information Processing Systems (NeurIPS)* (2019).

APPENDICES

A ADDITIONAL ANALYSIS

A.1 Matching Criteria

We analyze our design choices by using our method to fit gradients computed on a single batch. We simplify the learning task by employing only one client that contains all training data, resembling centralized training or FedSGD with a single client. In this setup, the client immediately communicates the gradients to the central server after computing them on a single batch. This scenario can be seen as a trivial federated learning task, as it does not involve any model drifting (non-IID) or communication budget constraints. It is worth noting that the baseline performance in this setting is an *ideal* case that does not apply in any practical FL use cases (or out of scope of federated learning).

Gradient Magnitudes. While previous research [14, 59, 62] suggests that gradient directions are more crucial than magnitudes (Eq. 9), our study demonstrates that as training progresses, the magnitudes of synthetic gradients (i.e., gradients obtained from synthetic images) can differ significantly from real gradients. In Fig. 11, we display the gradient magnitudes of each layer in a ConvNet. Our findings indicate that even with only 500 iterations, the magnitudes of synthetic gradients (orange) noticeably deviate from the real ones (blue), causing unnecessary instability during training.

Post-hoc Magnitude Calibration. To further validate the issue, we implement a post-hoc magnitude calibration, called *Calibration* in Fig. 12. It calibrates the gradients obtained from the synthetic images on the server. Specifically, the clients send the layer-wise magnitudes of real gradients $\|\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w},\mathcal{D}_k)\|$ to the server, followed by a transformation on the server:

$$\frac{\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w},\mathcal{S}_k)}{\|\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w},\mathcal{S}_k)\|}\|\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w},\mathcal{D}_k)\|. \tag{20}$$

In Fig. 12, We observe that the synthetic images with the magnitude calibration successfully and continuously improve over the one without the calibration (Vanilla). It implies that even with the same cosine similarity, inaccurate gradient magnitudes could dramatically fail the training. It is worth noting that the performance gap between the baseline and our method in this experiment does not apply to federated learning since FL models suffer from non-IID problems induced by multiple steps and clients, the gradient-sharing schemes, such as FedAvg, overly approximate the update signal, further enhancing the problem. Meanwhile, it also suggests an opportunity to improve our method if future work can further bridge the gap.

Mean Square Error (MSE) Regularization. Although calibration can improve performance, it is not suitable for federated learning due to two reasons. Firstly, when the synthetic images $\mathcal{S}_k$ from different clients are merged into a dataset for server optimization, it remains unclear how to apply Eq. 9 to the averaged gradients of the merged dataset $\mathcal{S}$, especially when multiple-step optimization is involved. Secondly, transmitting magnitudes can pose privacy risks. Instead of explicit calibration, we propose using MSE regularization (Eq. 9, termed *Regularization* in Fig. 12) to limit potential magnitude deviations while focusing on the directions. As depicted in Fig. 12, our proposed method remains close to the calibration method, suggesting that regularization prevents mismatch.

Min	Max	Median	Fixed
71.90	72.39	72.33	71.26

Table 5: Performance comparison between radius selection strategies.

Moreover, we present an additional implementation with an MSE matching criterion (termed *MSE* in Fig. 12)). It solely matches the mean square error distance between $\mathcal{L}(\mathbf{w},\mathcal{D}_k)$ and $\mathcal{L}(\mathbf{w},\mathcal{S}_k)$ regardless of gradient directional information. That is,

$$\|\nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)},\mathcal{D}_k) - \nabla_{\mathbf{w}^{(l)}}\mathcal{L}(\mathbf{w}^{(l)},\mathcal{S}_k)\|_2^2 \tag{21}$$

Despite the improvement over the vanilla method, it still falls behind *Calibration* and *Regularization*, highlighting the importance of directional information and justifying our design choice.

A.2 Radius Selection

As in Sec. 4, we evaluate different radius selection strategies for server optimization. We consider four strategies: *Fixed*, *Max*, *Median*, and *Min*. *Fixed* uses a fixed length of 100 iterations, ignoring the quality. *Max* aims for the fastest optimization by using the largest radius. *Median* optimizes in a moderate way by considering the majority. *Min*, adopted in all experiments, focuses on the safest region agreed upon by all synthetic image sets. Table 3 reveals that the proposed strategies consistently outperform the fixed strategy. Meanwhile, Fig. 14 presents that a more aggressive strategy leads to worse intermediate performance, which may not be suitable for federated applications requiring satisfactory intermediate performance. Among them, *Min* delivers the best results. Finally, Fig. 15 demonstrates that the radii proposed by different strategies change across epochs, indicating that a naively set training iteration may not be optimal. Additionally, this finding suggests the possibility of designing heuristic scheduling functions for adjusting the radius in a privacy-preserving way. The corresponding server training iterations can be found in the appendix.

In Sec. 6.4, we show that the effective approximation regions change across rounds. A fixed pre-defined training iterations may cause sub-optimal performance. To complement the experiments, we additionally plot the corresponding training iterations on the server side in Fig. 13. We observed that *Max* and *Median* tend to be more aggressive by updating for more epochs, granting faster improvement, while *Min* optimizes more conservatively. Interestingly, we found that all three proposed strategies exhibit similar behavior in the later stages of training, which is in stark contrast to the fixed strategy.

A.3 Qualitative Results

In this section, we examine the synthetic images. In non-private settings, Fig. 16 displays the pixel value distributions of all synthetic images at epochs 5, 50, 100, 150, and 195, both on the server and the clients. We made several observations from these visualizations. Firstly, within the same round, the distributions of pixel values

```
ConvNet(
  (features): Sequential(
    (0): Conv2d(3, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): GroupNorm(128, 128, eps=1e-05, affine=True)
    (2): ReLU(inplace=True)
    (3): AvgPool2d(kernel_size=2, stride=2, padding=0)
    (4): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (5): GroupNorm(128, 128, eps=1e-05, affine=True)
    (6): ReLU(inplace=True)
    (7): AvgPool2d(kernel_size=2, stride=2, padding=0)
    (8): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (9): GroupNorm(128, 128, eps=1e-05, affine=True)
    (10): ReLU(inplace=True)
    (11): AvgPool2d(kernel_size=2, stride=2, padding=0)
  )
  (classifier): Linear(in_features=2048, out_features=10, bias=True)
)
```

Figure 10: Architecture of ConvNets used in the federated experiments.

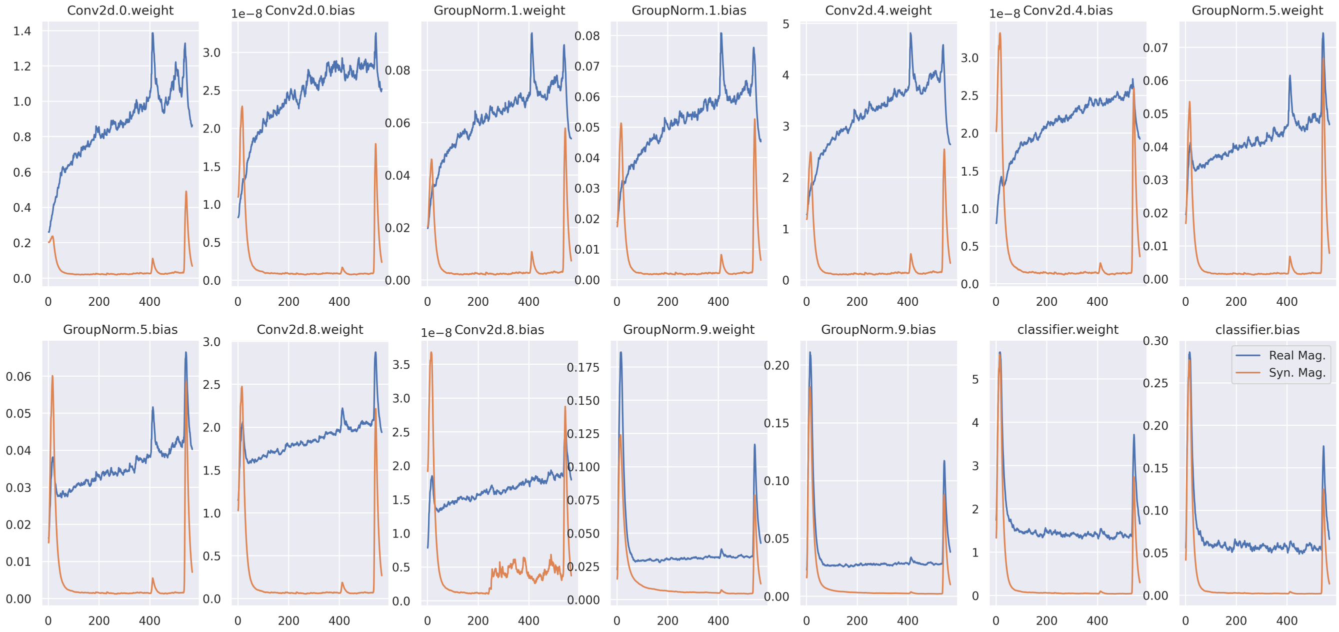


Figure 11: Discrepancy in gradient magnitudes between real and synthetic data. The noticeable difference in magnitudes during training highlights the limitation of solely regulating gradient directions, as optimizing without magnitude information introduces training instability.

on each client exhibit distinct behavior, indicating the diversity of private data statistics across clients. Secondly, the distributions also vary across different epochs. At earlier stages, the synthetic images present a wider range of values, gradually concentrating around zero as training progresses. This phenomenon introduces more detailed information for training and demonstrates that our method faithfully reflects the training status of each client. Additionally, Fig. 6 displays visual examples of synthetic images corresponding to the labels "airplane," "automobile," "bird," and "cat." These visuals indicate that earlier epoch images exhibit larger pixel values and progressively integrate more noise during training, reflecting gradient details. It's crucial to underscore that our method is *not intended* to produce realistic data but to approximate loss landscapes.

B COMPUTATION COMPLEXITY

Our method suggests an alternative to current research, trading computation for improved performance and communication costs incurred by slow convergence and biased optimization. We present the computation complexity analysis for one communication round

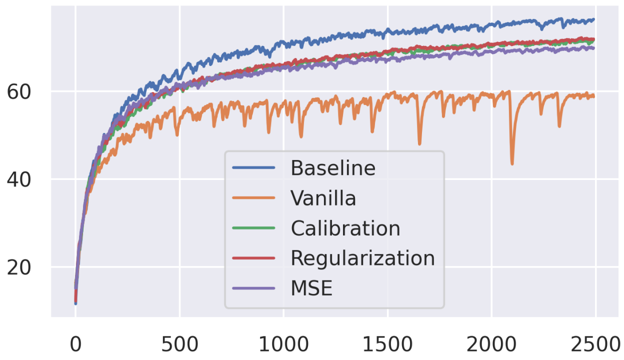


Figure 12: Performance comparison of fitting one batch. Baseline: FedSGD with one client. Vanilla: our method with cosine similarity. Calibration: our method with cosine similarity and magnitude calibration. Regularization: our method with cosine similarity and MSE regularization (i.e., Eq. 9). MSE: gradient matching by solely measuring mean square errors (Eq. 21).

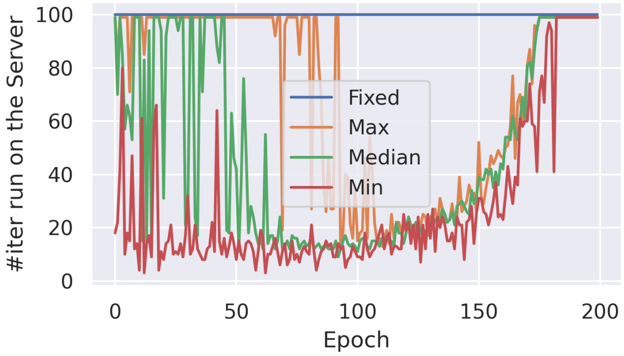


Figure 13: Training iterations for different radius selection strategies.

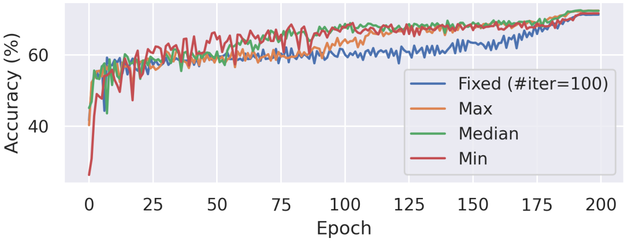


Figure 14: Ablation study on radius selection.

below and conclude with empirical evidence that the additional overhead is manageable, especially for cross-silo scenarios.

We begin with the SGD complexity $O(d)$, where d denotes the number of network parameters. Suppose synthetic samples contain p trainable parameters; then the complexity can be formulated as follows.

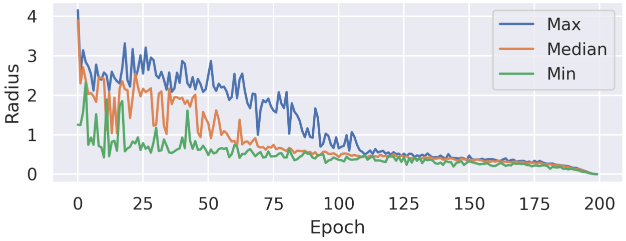


Figure 15: Radius suggested by different strategies.

$$
\begin{aligned}
& O\Big(R_i \cdot 5 \cdot \Big(N(d + 2R_b(d+p)) + R_l d\Big)\Big) \\
= \; & O\Big(5R_i \cdot (2NR_b + R_l + N)d + 10R_iR_bp\Big) \\
= \; & O\Big(5R_iN \cdot (2R_b + \frac{R_l}{N} + 1)d + 10R_iR_bp\Big)
\end{aligned}
$$

Note that $5R_iN$ determines how much real data we will see during synthesis. For comparison, we make $5R_iN$ equal in both our method and gradient-sharing baselines (i.e., five local epochs in FedAvg with complexity $O(5R_iNd)$). Overall, our method introduces $2R_b + \frac{R_l}{N} + 1$ times more computation on network parameters and an additional $10R_iR_b$ term for updating synthetic samples.

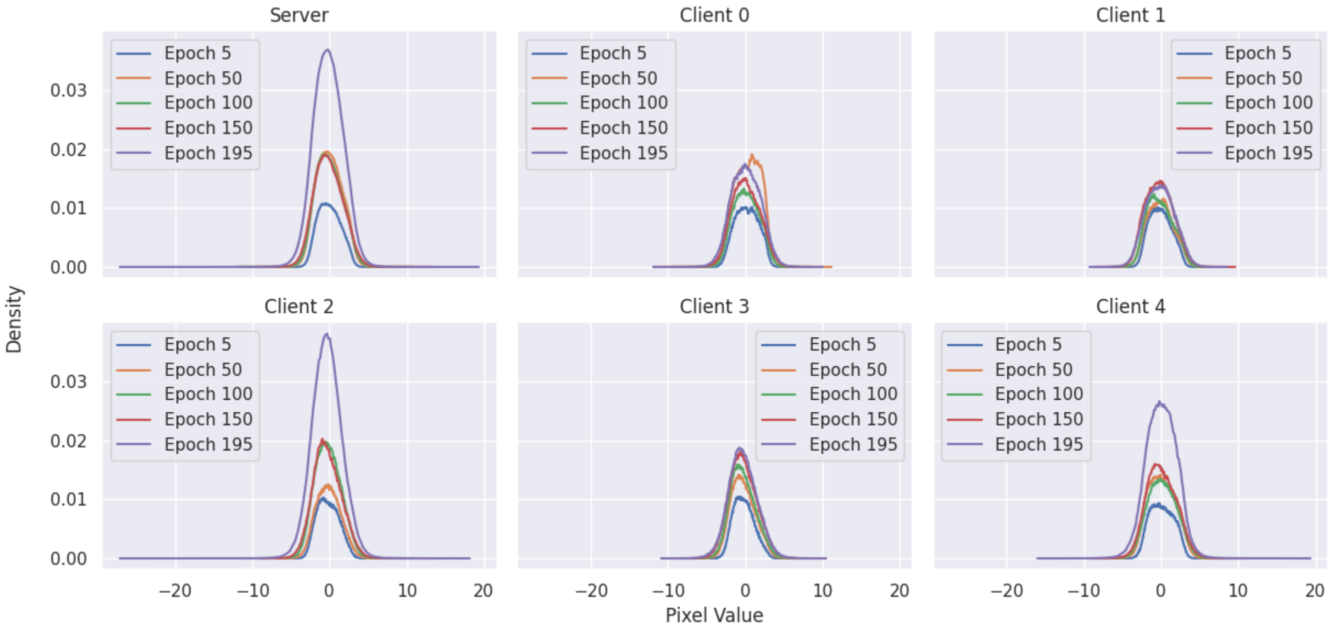


Figure 16: Pixel distributions in non-private settings. The plot illustrates the evolution of pixel values in synthetic images during training. At the early training stage, the pixel range is wider and gradually concentrates around zero as the model approaches convergence, implying that the synthetic images reflect the training status.