# PRIVATEFL: Accurate, Differentially Private Federated Learning via Personalized Data Transformation

*Yuchen Yang*, Bo Hui*, Haolin Yuan*, Neil Gong†, and Yinzhi Cao*
*The Johns Hopkins University, †Duke University*

## Abstract

Federated learning (FL) enables multiple clients to collaboratively train a model with the coordination of a central server. Although FL improves data privacy via keeping each client's training data locally, an attacker—e.g., an untrusted server—can still compromise the privacy of clients' local training data via various inference attacks. A de facto approach to preserving FL privacy is Differential Privacy (DP), which adds random noise during training. However, when applied to FL, DP suffers from a key limitation: it sacrifices the model accuracy substantially—which is even more severely than being applied to traditional centralized learning—to achieve a meaningful level of privacy.

In this paper, we study the accuracy degradation cause of FL+DP and then design an approach to improve the accuracy. First, we propose that such accuracy degradation is partially because DP introduces additional heterogeneity among FL clients when adding different random noise with clipping bias during local training. To the best of our knowledge, we are the *first* to associate DP in FL with client heterogeneity. Second, we design PRIVATEFL to learn accurate, differentially private models in FL with reduced heterogeneity. The key idea is to jointly learn a differentially private, personalized data transformation for each client during local training. The personalized data transformation shifts client's local data distribution to compensate the heterogeneity introduced by DP, thus improving FL model's accuracy.

In the evaluation, we combine and compare PRIVATEFL with eight state-of-the-art differentially private FL methods on seven benchmark datasets, including six image and one non-image datasets. Our results show that PRIVATEFL learns accurate FL models with a small $\varepsilon$, e.g., 93.3% on CIFAR-10 with 100 clients under ($\varepsilon = 2$, $\delta = 1e-3$)-DP. Moreover, PRIVATEFL can be combined with prior works to reduce DP-induced heterogeneity and further improve their accuracy.

*Equal contribution

## 1 Introduction

Federated Learning (FL) is a distributed learning framework, which allows multiple clients to collaboratively train a model with the coordination of a central server. Although FL improves data privacy without uploading local training data to the server, an adversary can still infer local data via various attacks. For example, Nasr et al. [32] shows that a malicious server can rely on membership inference attacks to infer whether a local client has a given data sample. Therefore, FL is often used together with Differential Privacy (DP) [4, 12, 13, 42], a de facto approach in preserving data privacy with formal guarantees.

Historically, there are two versions of differential privacy [31]: Local Differential Privacy (LDP) and Central Differential Privacy (CDP). The former adds noise at client and ensures each client with a privacy guarantee against malicious server and client, and the latter adds noise at server and ensures the global model with a privacy guarantee against malicious clients. Recently, researchers also propose Distributed Differential Privacy (DDP) [5, 18, 42] in between LDP and CDP to prevent an honest-but-curious server [35].

However, regardless of the DP variation, one key challenge in applying DP to the FL setting is the degradation of the model's accuracy. While it is natural that DP compromises model's accuracy due to the inherent utility-privacy trade-off, such an accuracy degradation is even more severe under the FL setting. For example, our experiment shows that the accuracy degradation comparing learning models with and without ($\varepsilon = 2$, $\delta = 1e-3$)-LDP under an FL setting with 100 clients is 12.5% as opposed to 1.2% under a centralized learning with all other parameters being the same (i.e., MNIST as the training set with i.i.d. distribution and three-fully-connected-layer neural network as the model).

In the first part of the paper, we study the cause of such a big accuracy degradation under FL and different variations of DP (called FL+DP) as the motivation of our research. Specifically, an FL model's accuracy is sensitive to the client data distributions [26], e.g., an FL model trained with heterogeneous data distribution among clients usually performs worse compared with one trained with homogeneous data distribution. Our observation and study show that all variations of DP introduce additional heterogeneity to FL clients during training, thus hampering the overall FL model's accuracy. In other words, when noises are either added independently at each local client in LDP and DDP or diversified by each local client in CDP, such noises enlarge heterogeneity across clients. To the best of our knowledge, we are the *first* to associate FL accuracy degradation caused by DP with heterogeneity.

There is no prior work that studies the specific problem of client heterogeneity introduced by DP under the FL setting. On one hand, prior works, e.g., Papernot et al. [34] and Tramer et al. [44], proposed to improve DP's utility with more data samples (e.g., those public data used to train an encoder), better features, and different activation functions. However, such approaches do not reduce client heterogeneity introduced by DP. On the other hand, prior works proposed to tame client data heterogeneity via personalized FL [19, 26]. Particularly, DP-SCAFFOLD [33] combines a popular, personalized FL called SCAFFOLD [19] with DP to improve privacy with heterogeneous data. However, such approaches, including DP-SCAFFOLD, are designed for general training data heterogeneity at each client but *not* those introduced by DP: General training data heterogeneity is stable between each round, but DP-induced heterogeneity changes due to random noise added in each round.

Therefore, in the second and main part of the paper, we design and implement an accurate, differentially private federated learning, called PRIVATEFL, with novel personalized data transformation. The key insight of PRIVATEFL is to shift client data distribution with personalized data transformation at each FL client to tame heterogeneity introduced by DP. While intuitively simple, the challenge is how to find the optimal transformation to reduce heterogeneity with improved utility. Therefore, PRIVATEFL optimizes the transformation in each FL round together with learning models and such an optimized transformation has the following properties:

- Optimized for utility. PRIVATEFL *learns* the personalized data transformation based on the local data distribution and optimizes the transformation to minimize learning loss and maximize local client's model utility.
- Differentially private. PRIVATEFL ensures the original privacy guarantee of different DP variations (L/D/CDP) under their threat model. For example, PRIVATEFL applies DP-SGD upon the data transformation during backpropagation for LDP to *guarantee* that the combination of the DNN and the data transformation is differentially private.
- Compatible. PRIVATEFL, serving as a *pluggable* component to DNN, is complementary to and compatible with personalized FL and/or existing DP utility improvement.

We evaluate the generality of PRIVATEFL using seven benchmark datasets (e.g., CIFAR-100) on multiple model architectures including the latest CLIP [36]. We also combine PRIVATEFL with existing DP improvement methods [34, 44] and personalized FLs (e.g., FedBN [26] and DP-SCAFFOLD [33]). Our evaluation shows that PRIVATEFL further improves FL models' accuracy when combined with those works. Take DP-SCAFFOLD for example. PRIVATEFL further improves the FL model's accuracy by 5.7% upon DP-SCAFFOLD with $\varepsilon = 8$ and by 22.5% with $\varepsilon = 2$.

To summarize, our key contributions are as follows:

- We find that the utility degradation of DP+FL is partially due to additional heterogeneity introduced by DP.
- We design and implement PRIVATEFL, the first approach to tame heterogeneity introduced by DP and improve model utility via a personalized, optimized data transformation. Our implementation is open-source at this repository (`https://github.com/BHui97/PrivateFL`).
- We demonstrate that PRIVATEFL can be combined with personalized FL and other DP utility improvement methods to further improve FL's utility with DP.

## 2 Problem Formulation

### 2.1 Preliminary

**Differential Privacy (DP) [4, 40].** Let $D$ denote all possible datasets and $R$ denote the domain of all possible trained models. A randomized mechanism $A : D \rightarrow R$ satisfies $(\varepsilon, \delta)$-differential privacy if—for any two neighboring datasets $D_1, D_2 \in D$ that differ in only a single data sample and for any subset of output $S \subseteq R$—the following Equation 1 holds:

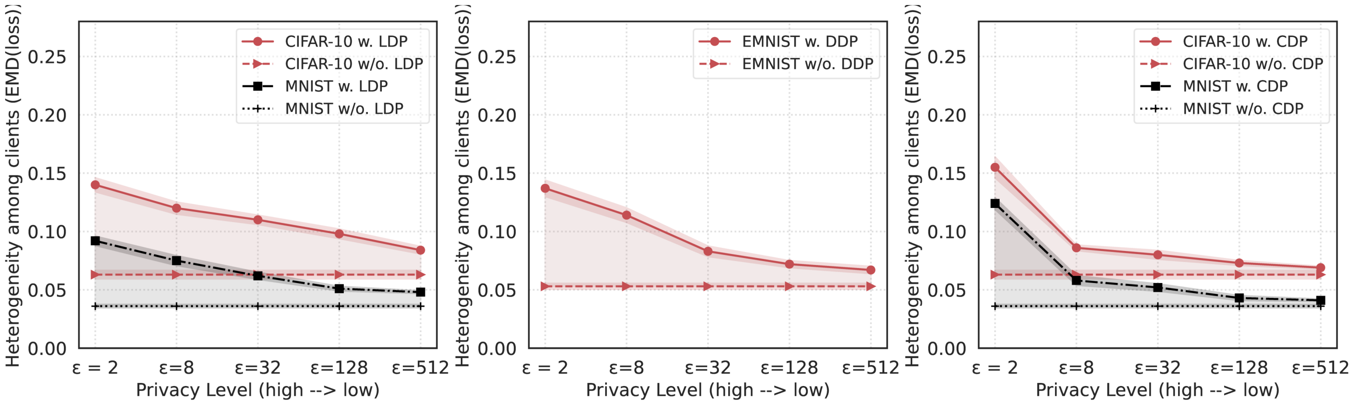
$$\Pr[A(D_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[A(D_2) \in S] + \delta. \quad (1)$$

The parameters $\varepsilon$ and $\delta$ are related to the unconditional upper-bound of the potential privacy leakage, and are known as *privacy budgets*. Specifically, $\varepsilon$ represents the privacy guarantee: a lower $\varepsilon$ corresponds to a higher level of privacy; and $\delta$ indicates the probability that the upper-bound does not hold, and is usually set to be the inverse of the training dataset size.

**Federated Learning (FL) [26, 28].** FL allows multiple clients to train a global model collaboratively while keeping their training data locally. Specifically, in each training round, the server sends the current global model to the clients or a sampled subset of them; each client trains a local model using its training data, and uploads it to the server; and the server aggregates the clients' local models as a new global model. For instance, FedAVG [28] is a standard way to aggregate local models, which takes the average of the local models as a global model: $\theta^{r+1} = \frac{1}{K}\sum_{k=0}^{K}\theta_k^r$, where $\theta^{r+1}$ is the global model for the $(r+1)$-th round, $K$ is the number of clients, and $\theta_k^r$ is the local model of client $k$ in round $r$.

### 2.2 Threat Model

An adversary aims to infer sensitive information about clients' local training data, e.g., data distribution [16] or memberships [32]. We first define different types of adversaries, and




(a) [LDP] Heterogeneity vs. Privacy Level (b) [DDP] Heterogeneity vs. Privacy Level (c) [CDP] Heterogeneity vs. Privacy Level

Figure 1: A motivating example to illustrate that DP introduces additional client heterogeneity during training. Each horizontal line shows the baseline heterogeneity introduced by clients' non-i.i.d local training data.

then introduce different variants of FL+DP that can prevent different adversaries.

- **Untrusted Server**. Such an adversary controls the FL server and may either passively exploit each client's local models (i.e., honest-but-curious server) or actively tamper with the aggregation procedure (i.e., malicious server) to perform inference attacks.
- **Untrusted Client**. Such an adversary controls one or multiple clients, and tries to perform inference attacks to other clients' local training data via passively exploiting the global models from the server (i.e., honest-but-curious clients) or actively tampering with the local models on its controlled clients (i.e., malicious clients).

Different FL+DP variants have different threat models against the aforementioned adversaries. We present them from the strongest to the weakest.

- **Local Differential Privacy (LDP) [43, 47]**. LDP is the strongest variant, which defends against both untrusted server and untrusted clients. Each LDP client adds noise in each training step using DP-SGD [4] when training its local model in each FL round. Note that the clients may use different privacy budgets $\varepsilon$, and the overall privacy guarantee depends on the largest $\varepsilon$ among all the clients.
- **Distributed Differential Privacy (DDP) [5, 18]**. DDP is a weaker variant, which defends against a honest-but-curious server and untrusted clients. Each DDP client adds noise to its trained local model (but not the training procedure) and the server adopts secure aggregation. Note that since the added noise is insufficient for an LDP-like guarantee, DDP is still vulnerable to a malicious server [35].
- **Central Differential Privacy (CDP) [15, 29]**. CDP is the weakest version of FL+DP. The CDP server adds calibrated noise during aggregation to achieve DP for the global model. CDP can defend against untrusted clients, but not an untrusted server. The reason is that the server can access the clients' noise-free local models in CDP.

### 2.3 Motivation

We motivate the design of PRIVATEFL by first understanding the cause of the accuracy degradation of DP+FL. One cause, according to our observation, is that DP introduces additional heterogeneity to FL clients. In this subsection, we perform experiments to validate our observation and our experimental setup is as follows. We assume 10 clients with non-i.i.d local training data and each client has data from six random classes. Then, we adopt the CIFAR-10 and MNIST datasets for LDP and CDP, and the EMNIST dataset—following the original paper [5]—for DDP. The model architecture, sample rate, $(\varepsilon, \delta)$, learning rate, and batch size are the same as our experiment setup in Section 5 (particularly Table 3).

In each FL training round, we calculate a client's local training loss of its local model on its training data. Thus, for each client, we have a distribution of its local training loss across the FL training rounds. Such distribution is homogeneous in an ideal case, i.e., the clients have very similar distributions when their local training data are i.i.d. and DP is not used during training. We then calculate the *earth mover's distance (EMD)* [37] between such distributions for each pair of clients. A larger EMD between the local training loss distributions of two clients indicates more heterogeneity. Figure 1 shows the mean and standard deviation of the EMDs between the clients as $\varepsilon$ varies for each of the three DP variants. Each horizontal line in Figure 1 is the baseline heterogeneity introduced by clients' non-i.i.d local training data. The margin between a line and the horizontal line indicates the additional heterogeneity introduced by DP.

**LDP-introduced Heterogeneity.** Intuitively, LDP introduces heterogeneity from two aspects. First, DP-SGD clips gradients of random samples of training data, which introduces clipping bias [9] and thus heterogeneity. Second, DP-SGD adds random noise to the clipped gradients, which further introduces heterogeneity among clients. Figure 1a shows

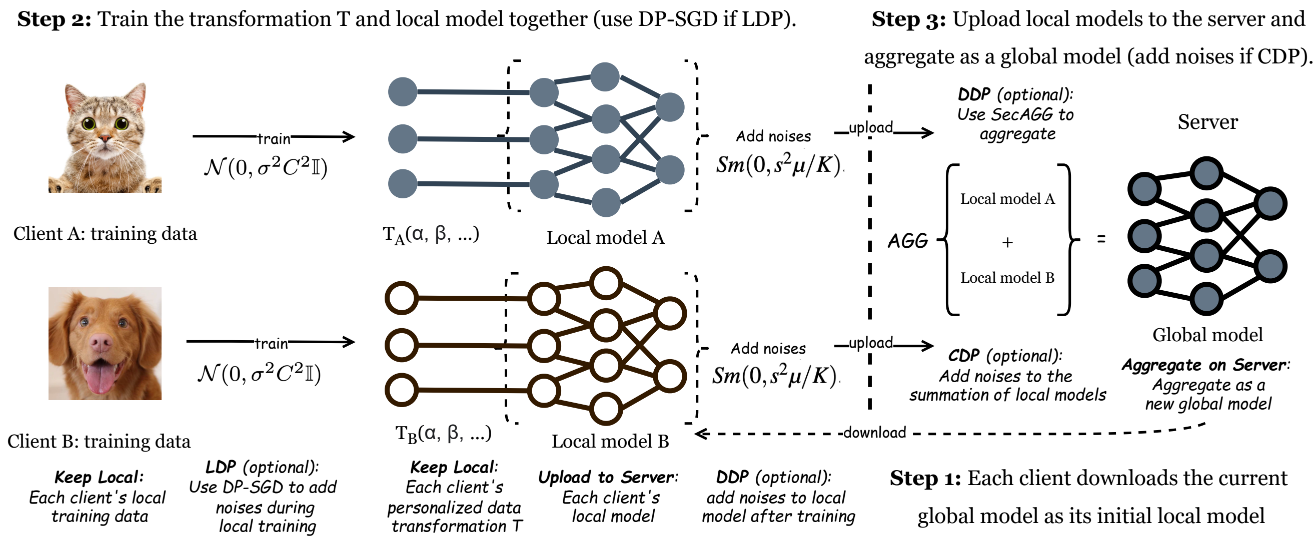



Figure 2: In each round, PRIVATEFL performs three steps. Step 1: each client downloads the current global model. Step 2: each client learns a personalized data transformation and a local model simultaneously (DP-SGD is adopted as the optimizer for LDP and noise is added to the learnt local model for DDP). Step 3: each client uploads its local model to the server and keeps its personalized data transformation locally. The server aggregates the local models using secure aggregation (DDP) or FedAVG with noise added to the aggregation (CDP).

the heterogeneity when ε varies. The heterogeneity increases as ε decreases, since a smaller ε adds more noises, thus introducing more heterogeneity.

**DDP-introduced Heterogeneity.** Similar to LDP, DDP introduces heterogeneity because DDP adds random (different) noise to the local models. Figure 1b shows that the heterogeneity decreases as ε increases (i.e., privacy level decreases). The reason is that the noise level decreases as ε increases and thus the noise-introduced heterogeneity also decreases.

**CDP-introduced Heterogeneity.** CDP introduces additional heterogeneity when the clients' local training data are non-i.i.d: while the noise is added by the server, the same noise in a global model is amplified differently during local training by clients. Specifically, the clients change the same noise added by the server to the global model differently because their local training data are non-i.i.d and introduce additional heterogeneity to the local models returned to the server. Figure 1c shows the heterogeneity as ε varies. Such heterogeneity is relatively small when ε is large but increases drastically as ε decreases. This is because a small ε introduces more noise in a global model, which is further diversified and amplified by the clients' heterogeneous training data during local training. We note that, when the clients' local training data are i.i.d, CDP does not introduce much additional heterogeneity because the clients would amplify the noise added to the global model similarly, as shown in Figure 8b in Appendix A.

*Motivation Takeaway*: Differential Privacy (DP) introduces *additional* heterogeneity among FL clients, thus hurting FL model's utility; and DP-introduced heterogeneity increases as the privacy level.

## 3 Methodology

In this section, we describe personalized data transformation under FL and then its combination with DP and other DP utility improvement methods. Our high-level intuition is that personalized data transformation is optimized during each FL training round to better fit a local model with new DP-introduced noise in the training round, thus improving the accuracy of the global model.

### 3.1 Personalized Data Transformation

**Notations.** We assume $K$ clients and denote by $D_k$ the local training dataset for client $k$, where $k = 1, 2, \cdots, K$. The local training dataset size is $|D_k|$. We consider $(x, y)$ as a training sample, where $x \in \mathbb{R}^n$ denotes the training input and $y$ the label of the input. We denote θ as the global model and $r$ as the training round index.

**Definition.** We define a personalized data transformation as a local function $x_t = T_k(x)$ at client $k$, where $T_k$ transforms a given training sample $x$ to $x_t$. Different clients have different $T_k$, which are kept locally without being aggregated by the central server. The choice of $T_k$ follows two principles: (i) *Taming heterogeneity* and (ii) *Preserving original features*. For the former one, $T_k$ needs to shift the local data distribu-

**Algorithm 1** PRIVATEFL-LDP/DDP/CDP

**Input:** Learning rate $\eta$ and the number of sampled clients $K_s$ in each round. For LDP: noise scale $\sigma$, group size $L$, and gradient norm bound $C$. For DDP: bit-width $b$, target central noise variance $\mu > 0$, signal bound multiplier $m > 0$, bias $v \in [0,1)$, gradient norm bound $C$, $j \times j$ diagonal matrix $J$ with uniformly random $\{$-1, 1$\}$ values, $j \times j$ Hadamard matrix $H$, and $s$ that satisfies $2^b = 2m\sqrt{C^2K_s^2/j + K_s/(4s^2)} + \mu$. For CDP: gradient norm bound $C$.

**Output:** Global model $\theta$ and the overall privacy budget $(\varepsilon, \delta)$ computed using a privacy accounting method.

1: Server initializes $\theta^0$ randomly
2: **for** each round $r = 1, 2, \cdots$ **do**
3: $\quad \mathcal{K}^r \leftarrow$ server samples a set of $K_s$ clients randomly
4: $\quad$ **for** each client $k \in \mathcal{K}^r$ in parallel **do**
5: $\quad\quad \theta^r_{k_L} = \texttt{LocalUpdate}(k, \theta^r)$ ▷ get local model from client $k$
6: $\quad$ **if** LDP **then** ▷ update global model
7: $\quad\quad \theta^{r+1} \leftarrow \frac{1}{K_s}\sum_{k \in \mathcal{K}^r} \theta^r_{k_L}$
8: $\quad$ **if** DDP **then**
9: $\quad\quad \theta^r_L \leftarrow$ sum of $\{\theta^r_{k_L}\}_{k \in \mathcal{K}^r}$ modular $2^b$
10: $\quad\quad \theta^{r+1} \leftarrow \frac{1}{s} J H^\top \theta^r_L$
11: $\quad$ **if** CDP **then**
12: $\quad\quad \zeta_k \leftarrow ||\theta^r_{k_L} - \theta^r||_2, \forall k \in \mathcal{K}^r$
13: $\quad\quad \zeta \leftarrow median\{\zeta_k\}_{k \in \mathcal{K}^r}$ ▷ median norm bound for CDP
14: $\quad\quad \theta^{r+1} \leftarrow \theta^r + \frac{1}{K_s}(\sum_{k \in \mathcal{K}^r} \frac{\theta^r_{k_L} - \theta^r}{\max(1, \frac{\zeta_k}{\zeta})} + \mathcal{N}(0, \sigma^2\zeta^2\mathbb{I}))$
15: return $\theta^r$ and $(\varepsilon, \delta)$

**Algorithm 2** $\texttt{LocalUpdate}(k, \theta)$

**Input:** Client $k$ and global model $\theta$
**Output:** Local model $\theta_{k_L}$

$\theta_k \leftarrow \theta_{k_T} \odot \theta$ ▷ obtain an extended local model
**for** each local iteration **do**
$\quad$ **if** LDP **then**
$\quad\quad$ Take a random mini-batch $B$ with sampling probability $L/n_k$
$\quad\quad$ **for** each $(x_i, y_i) \in B$ **do**
$\quad\quad\quad g_i \leftarrow \nabla l(\theta_k, x_i, y_i)$ ▷ compute gradient
$\quad\quad\quad g_i \leftarrow g_i / \max(1, \frac{||g_i||_2}{C})$ ▷ clip gradient if needed
$\quad\quad g \leftarrow \frac{1}{|B|}\sum_i (g_i + \mathcal{N}(0, \sigma^2C^2\mathbb{I}))$ ▷ add noise
$\quad$ **if** CDP or DDP **then**
$\quad\quad$ Take a random mini-batch $B$ with size $L$
$\quad\quad g \leftarrow \frac{1}{|B|}\sum_{(x_i,y_i) \in B} \nabla l(\theta_k, x_i, y_i)$ ▷ non-private gradient
$\quad \theta_k \leftarrow \theta_k - \eta g$
$\theta_{k_L} \leftarrow$ get local model from $\theta_k$
**if** DDP **then**
$\quad \theta_{k_L} \leftarrow \texttt{DDPNoise}(\theta_{k_L})$ ▷ add noise for DDP
return $\theta_{k_L}$ ▷ upload local model

**Algorithm 3** $\texttt{DDPNoise}(\theta_{k_L})$

**Input:** Local model $\theta_{k_L}$
**Output:** Noisy local model $\theta_{k_L}$

$\theta_{k_L} \leftarrow s \cdot min(1, C/||\theta_{k_L}||_2) \cdot \theta_{k_L}$ ▷ clip local model
$\theta_{k_L} \leftarrow \frac{1}{\sqrt{j}} H J \theta_{k_L}$ ▷ random rotation
$M \leftarrow min\{(sC + \sqrt{j})^2, s^2C^2 + j/4 + \sqrt{2log(1/v)} \cdot (sC + \sqrt{j}/2)\}$
**while** $||\theta_{k_L}||_2^2 \leq M$ **do**
$\quad \theta_{k_L} \leftarrow$ stochastically round the coordinates of $\theta_{k_L}$
$\theta_{k_L} \leftarrow \theta_{k_L} + Sk(0, s^2\mu/K^r)$ ▷ add noise drawn from Skellam distribution
return $\theta_{k_L}$

tion of each client to tame heterogeneity caused either by the inherent data heterogeneity or DP. For the latter one, $T_k$ needs to keep the original features of the training data so that FL can learn such features to train an accurate global model. Theoretically, $T_k$ can be any function. However, our empirical study in Section 4 finds that linear transformation satisfies these two principles well, thus outperforming many other alternatives.

### 3.2 Learning the Transformation

Intuitively, our idea is to learn a $T_k$ together with the local model and tailor $T_k$ for different clients to tame heterogeneity. In particular, we view $T_k$ as the beginning layer (called layer zero) of a local model and denote $T_k$ as $\theta_{k_T}$ for the purpose of learning. Formally, we denote by $\theta_{k_L}$ a local model and $\theta_k$ an *extended local model* of client $k$, where $\theta_k$ is the composition of the transformation layer $\theta_{k_T}$ and the local model $\theta_{k_L}$. Figure 2 shows the detailed procedure of learning $\theta_k$ including $\theta_{k_T}$. Specifically, a client $k$ learns an extended local model via solving the following optimization problem in round $r$:

$$\min_{\theta^r_k} \frac{1}{|D_k|} \sum_{(x_i,y_i) \in D_k} l(\theta^r_k, x_i, y_i), \quad (2)$$

where $l(\theta^r_k, x_i, y_i)$ is the loss of the extended local model $\theta^r_k$ for a training sample $(x_i, y_i)$. When solving the optimization problem, the local model $\theta^r_{k_L}$ is initialized as the global model $\theta^r$ and the transformation layer $\theta^r_{k_T}$ is initialized as the learnt transformation layer $\theta^{r-1}_{k_T}$ in the previous round. After learning an extended local model, client $k$ uploads the local model $\theta^r_{k_L}$ to the server. The server aggregates the local models as a new global model for the next round.

Note that our transformation $\theta^r_{k_T}$ is different from a classic neural network layer due to the following reasons. First, $\theta^r_{k_T}$ is personalized and not aggregated by the server. The purpose of personalization is to tame client heterogeneity. Second, $\theta^r_{k_T}$ is initialized to be *identity transformation* in round zero, i.e., $\theta^0_{k_T}(x) = x$. Intuitively, we adopt this initialization setting because an identity transformation preserves the original data's features. We will discuss different initializations in Section 4.

### 3.3 Differentially Private Transformation

The transformation $\theta_{k_T}$ that we introduced so far is personalized to tame heterogeneity, but not yet to be combined with DP. In this section, we describe how to combine our personalized transformation with each of the three DP variants in PRIVATEFL. Algorithm 1 and Algorithm 2 show the pseudocode of PRIVATEFL on the server and client, respectively. Next, we describe them separately.

**Server (Algorithm 1).** The server first randomly initializes the global model $\theta^0$. In each round, the server randomly selects a subset of $K_s$ clients and asks each of them to compute a local model. In LDP, the server takes the average of the clients' local models as a new global model (Line 7), following Fe-

Table 1: Impact of transformation function on the accuracy of the global model and average Earth Mover's Distance (EMD) between the clients' local training loss distributions in PRIVATEFL.

| | linear transformation | | | non-linear transformation | | |
|---|---|---|---|---|---|---|
| | function | accuracy | EMD | function | accuracy | EMD |
| one-to-one | $\alpha x+\beta$ | **0.892** | 0.220 | $Sigmoid(\alpha x+\beta)$ | 0.457 | 0.411 |
| | | | | $Tanh(\alpha x+\beta)$ | 0.613 | 0.393 |
| many-to-one | $Conv2d(x)$ | 0.829 | 0.272 | $\alpha x^2+\beta$ | 0.137 | 0.142 |
| | | | | $ReLU(\alpha x+\beta)$ | 0.421 | 0.419 |

dAvg. In DDP, the server aggregates the clients' local models using a particular aggregation rule (Line 10), which the authors of DDP called *secure aggregation*. In CDP, the server calculates the norm of each local model update (Line 12), takes the median of the norms (Line 13), uses the median norm to clip the local model updates if needed, and adds noise to the average clipped local model update before using it to update the global model (Line 14). The norm bound is the median value which follows the original CDP paper [15]. Note that privacy protection is ensured because the server in CDP is trustworthy.

**Client (Algorithm 2).** Given the current global model, a client aims to learn a local model and a personalized data transformation simultaneously. Specifically, a client first initializes an extended local model via concatenating its transformation layer learnt in the last round and the current global model (Line 1). Then, the client iteratively updates its extended local model. In each iteration, for LDP, the client samples a mini-batch of its training data, computes the gradient of the loss function for each sampled training sample, clips the gradient if needed, adds noise to the clipped gradient, and uses the average noisy, clipped gradient of the mini-batch to update the extended local model. For DDP or CDP, the client uses the average gradient of a mini-batch to update the extended local model in each iteration, following the standard stochastic gradient descent. After learning an extended local model, the client obtains the local model part (Line 13). In LDP and CDP, the client uploads the local model to the server. In DDP, the client adds noise to the local model using Algorithm 3 and uploads the noisy local model to the server.

**DP Guarantee.** PRIVATEFL-LDP ensures $(\varepsilon,\delta)$-DP for each extended local model, which includes both transformation layer and local model. Specifically, each client learns an extended local model using DP-SGD and thus an extended local model achieves $(\varepsilon,\delta)$-DP. Therefore, the final global model also achieves $(\varepsilon,\delta)$-DP. PRIVATEFL-DDP ensures each local model achieves DP using DDP and thus the global model $\theta$ is differentially private. PRIVATEFL-CDP ensures the global model $\theta$ is differentially private via adding noise to the aggregation using CDP, though the local models are not differentially private. Therefore, PRIVATEFL-LDP (or DDP or CDP) has the same DP guarantee as FedAvg+LDP (or DDP or CDP).

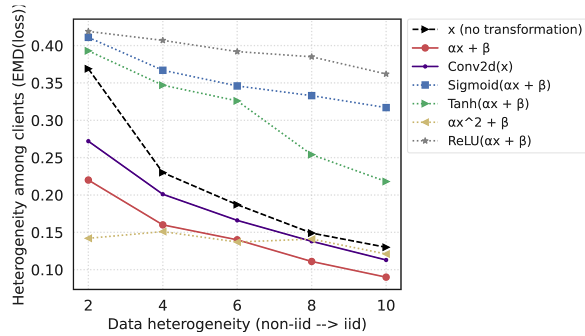



Figure 3: Impact of different transformation functions on the clients' average EMD in PRIVATEFL as the clients' local training data heterogeneity varies.

### 3.4 Compatibility with Existing Approaches

One advantage of PRIVATEFL is that it is compatible—and can be combined—with existing utility improvement approaches for both DP and FL. Essentially, PRIVATEFL provides a data transformation upon training data, which can be used together with other approaches. We use two examples for illustration. First, consider state-of-the-art work [44] that uses a feature extractor (called *encoder*) pre-trained on public dataset to improve DP utility. PRIVATEFL can use a pre-trained encoder to extract features of the training inputs and treat them as new training inputs. Then, we can apply PRIVATEFL on the new training inputs. Second, consider existing personalized FL [19, 26], especially personalized FL with DP, namely DP-SCAFFOLD [33]. Because PRIVATEFL performs upon the training data and DP-SCAFFOLD performs on the client updates, they are naturally compatible with each other.

## 4 Empirical Analysis of Transformation

In this section, we perform empirical analysis of possible personalized data transformations and their impacts on PRIVATEFL's performance. Specifically, we study two important factors: the transformation function space and the parameter space. Note that our analysis is by no means exhaustive and the purpose is to show that the empirical results confirm our intuition on the transformation choice.

Our empirical analysis adopts the following setting: LDP with $\varepsilon=8$ and $\delta=1e-3$, MNIST dataset, and 100 clients (each client has data from two classes). Without any transformation, the accuracy of the global model is 0.811 and the average EMD (discussed in Section 2.3) between clients' local training loss distributions is 0.369.

### 4.1 Transformation Function Space Analysis

We first analyze different transformation functions and compare their effectiveness in decreasing client heterogeneity and improving model accuracy in PRIVATEFL. Since the trans-

Table 2: Testing accuracy of the global model in PRIVATEFL when $\alpha$ and $\beta$ have different free dimensions.

| | $\beta = 0$ | $\beta \in \mathbb{R}^d$ | $\beta \in \mathbb{R}^{n\times n\times d}$ |
|---|---|---|---|
| $\alpha = 1$ | 0.811 | 0.865 | 0.841 |
| $\alpha \in \mathbb{R}^d$ | 0.862 | 0.876 | **0.892** |
| $\alpha \in \mathbb{R}^{n\times n\times d}$ | 0.849 | 0.863 | 0.802 |

formation function space is infinite, we have to choose a limited set of popular functions for experiments. Specifically, we choose *Polynomial*, *Sigmoid*, *Conv2d* (we use kernel size of $7\times 7$), and *Tanh*, and compare their performance with a *linear* transformation ($\alpha x+\beta$). Note that we represent a training input $x$ as a $n\times n\times d$ tensor, where $n$ is the image size and $d$ is the number of channels (3 for color image and 1 for gray image). $\alpha$ and $\beta$ are also $n\times n\times d$ tensors. $\alpha x$ represents element-wise product between $\alpha$ and $x$. Note that the parameters of a transformation function are automatically learnt in PRIVATEFL.

**Results.** Table 1 shows the impact of different transformation functions on the accuracy of the global model and clients' average EMD, while Figure 3 shows the impact of different transformation functions on the clients' average EMD when the clients' local training data heterogeneity varies, i.e., each client has 2, 4, 6, 8, or 10 classes of data. We observe that linear transformation $\alpha x+\beta$ achieves the best results, i.e., the accuracy is the largest and the average EMD is small. We note that the transformation $\alpha x^2+\beta$ achieves stable average EMD as the clients' local training data heterogeneity varies, and it achieves smaller EMD than the linear transformation when the clients' local training data are highly non-i.i.d (see Figure 3). However, $\alpha x^2+\beta$ substantially sacrifices the model accuracy, e.g., accuracy reduces to 0.137 in Table 1. This is because such transformation dramatically ruins the features, which reduces clients' heterogeneity but also makes the features useless.

**Intuitive Explanation.** Linear transformation better preserves features in the training samples and at the same time reduces heterogeneity introduced by DP. Consider that there exists a manifold that separates different classes of the original training data before transformation. Linear transformation shifts the manifold and keeps the original separability of different classes. As a high-level illustration, we show some transformed samples in Appendix B. By contrast, non-linear transformation distorts the manifold and distances between transformed samples. Specifically, activation functions like *Tanh* and *Sigmoid* regularize the range of feature values and reduce the original data's feature values. Similarly, one-to-one transformation better preserves features of original training data. Consider the aforementioned manifold separating classes. Many-to-one transformation distorts the manifold and may map samples of different classes to the same data point. More specifically, $Conv2d(x)$ maintains the correlation of features but loses some of them via convolution operation.

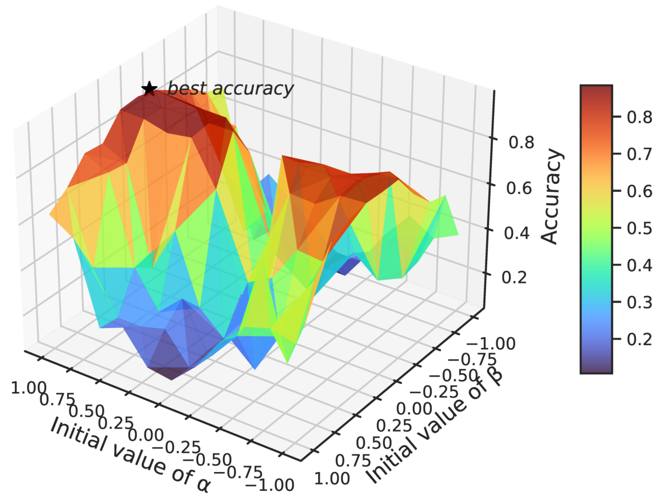



Figure 4: Testing accuracy of the global model in PRIVATEFL when $\alpha$ and $\beta$ have different initializations.

### 4.2 Transformation Parameter Space Analysis

After we determine that linear transformation works better than other alternatives, we then explore the parameter space of linear transformation, which includes parameter dimensions and initial values.

**4.2.1. Parameter Dimension.** Table 2 shows the global model's average testing accuracy with different free dimensions of parameters $\alpha$ and $\beta$. Specifically, $\alpha = 1$ (or $\beta = 0$) means that each element of $\alpha$ is 1 (of $\beta$ is 0). $\alpha \in \mathbb{R}^d$ (or $\beta \in \mathbb{R}^d$) means that each $n\times n$ matrix of $\alpha$ (or $\beta$), which corresponds to an image channel, shares the same value. In other words, $\alpha$ or $\beta$ has $d$ free parameters. $\alpha \in \mathbb{R}^{n\times n\times d}$ (or $\beta \in \mathbb{R}^{n\times n\times d}$) means that each element of $\alpha$ (or $\beta$) is a free parameter, leading to $dn^2$ free parameters for $\alpha$ (or $\beta$). We observe that $\alpha \in \mathbb{R}^d$ and $\beta \in \mathbb{R}^{n\times n\times d}$ achieve the best accuracy.

**Intuitive Explanation.** There is a tradeoff between parameter dimension size and the model's performance under DP + FL. On one hand, prior work [34] shows that a DP model's performance decreases as the parameter size increases because the DP noise is related to $\sqrt{\#\text{parameters}}$. On the other hand, a larger parameter dimension can better shift the training data distribution and tame client heterogeneity. Therefore, our finding shows that a channel-wise $\alpha$ and a pixel-channel-wise $\beta$, i.e., a parameter dimension in between, works the best among different alternatives.

**4.2.2. Parameter Initialization.** Figure 4 shows the global model's testing accuracy with different initial values of $\alpha$ and $\beta$. Specifically, we assign both parameters' initial values in the range [-1,1] with a 0.2 step size. PRIVATEFL achieves the highest accuracy when $\alpha = 1$ as the initial value, and the accuracy starts to decrease as $\alpha$ approaches 0 and then in-

Table 3: Default settings for architectures and hyperparameters of PRIVATEFL on different datasets.

| Dataset | DP | architecture | # clients | # rounds | sample rate | $(\epsilon, \delta)$ | $\eta$ | batch size |
|---|---|---|---|---|---|---|---|---|
| MNIST | LDP/CDP | 3-layer DNN | 100 | 150 | 1 | (8, 1e-3) | 1e-1/5e-3 | 64 |
| Fashion-MNIST | LDP/CDP | 3-layer DNN | 100 | 150 | 0.3 | (8, 1e-3) | 1e-1/5e-3 | 64 |
| EMNIST | LDP/CDP | 3-layer DNN | 100 | 100 | 0.3 | (8, 1e-3) | 1e-1 | 64 |
| CH-MNIST | LDP/CDP | AlexNet | 40 | 30/150 | 0.8 | (8, 1e-3) | 1e-1/1e-4 | 64 |
| CIFAR-10 | LDP/CDP | ResNet | 100 | 150/60 | 1 | (8, 1e-3) | 1e-1/5e-4 | 64/16 |
| CIFAR-100 | LDP/CDP | CLIP + 1-layer DNN | 100 | 20 | 1 | (8, 1e-3) | 1e-1 | 4/250 |
| Purchase-100 | LDP/CDP | 4-layer DNN | 50 | 580/150 | 0.1 | (8, 1e-3) | 1e-1/5e-3 | 64 |

creases as it approaches -1. PRIVATEFL achieves the highest accuracy when $\beta = 0$ as the initial value, and the accuracy starts to decrease as it moves either to 1 or -1.

**Intuitive Explanation.** Identity transformation preserves original features of the training samples and that is why it works the best as the initial parameters. Instead, assigning random values to initialize both $\alpha$ and $\beta$ makes a model less likely to learn true features because the randomness arbitrarily distorts features of training samples.

## 5 Implementation and Experimental Setup

**Implementation.** We implement PRIVATEFL using Python 3.8. Our implementation is open-source at this repository (https://github.com/BHui97/PrivateFL). We follow PyTorch Opacus 1.1.2 [3] to implement LDP, previous work [1] to implement CDP, and TensorFlow Federated 0.24.0 to implement DDP [2]. We use the moment accountant [40] to compute privacy budget for LDP, CDP, and DDP. All experiments are performed using two GeForce RTX 3090 graphics cards (NVIDIA). Our evaluation mainly adopts two metrics: testing accuracy of the global model and EMD between clients' local training loss distributions.

**Experimental Setup.** We use multiple benchmark datasets following previous DP work [34, 44], i.e., MNIST, Fashion-MNIST, and CIFAR-10. In addition, we also use EMNIST (a popular FL dataset), CH-MNIST (a medical dataset), CIFAR-100 (a challenging dataset with 100 classes), and Purchase-100 (a non-image dataset). Appendix C shows more details about the datasets. We follow the state-of-the-art personalized FL work [39] to assign training data to clients. Specifically, we first choose $N$ classes for each client uniformly at random. Then, we assign $\frac{p_{k,c}}{\sum p_{k,c}}$ of the training samples in class $c$ to each client $k$ whose chosen classes contain $c$, where $p_{k,c}$ is a random number in the range $(0.4, 0.6)$. Our default data distribution adopts $N = 2$ following previous work [31, 39]. Table 3 shows other default parameter settings, which we adopt in experiments unless otherwise mentioned.

We use FedAVG as our default aggregation rule. We follow previous works [15, 31] to set the default privacy budget $(\epsilon, \delta)$ as $(8, 1e-3)$. By default, we use one local training epoch, i.e., the number of local training iterations is $\frac{|D_k|}{|B|}$, where $|D_k|$ is the local training dataset size of client $k$ and $|B|$ is the batch size. The noise multiplier is calculated by the Opacus RDP accountant given the number of training rounds, sample rate, target $\epsilon$, and target $\delta$. For DDP, we evaluate on EMNIST following the setting in the original paper [5]. We also evaluate different $\epsilon$ values, i.e., [2, 4, 6]. Since $\epsilon$ depends on the number of training rounds, we use different number of training rounds for different $\epsilon$ values. In particular, for the considered $\epsilon$ values, we use [60, 80, 100] training rounds for MNIST, [60, 80, 100] for Fashion-MNIST, [30, 50, 60] for EMNIST, [10, 15, 25] for CH-MNIST, [280, 400, 500] for Purchase-100, [30, 40, 50] for CIFAR-10, and [20, 20, 20] for CIFAR-100.

## 6 Evaluation

We answer the following Research Questions (RQs).

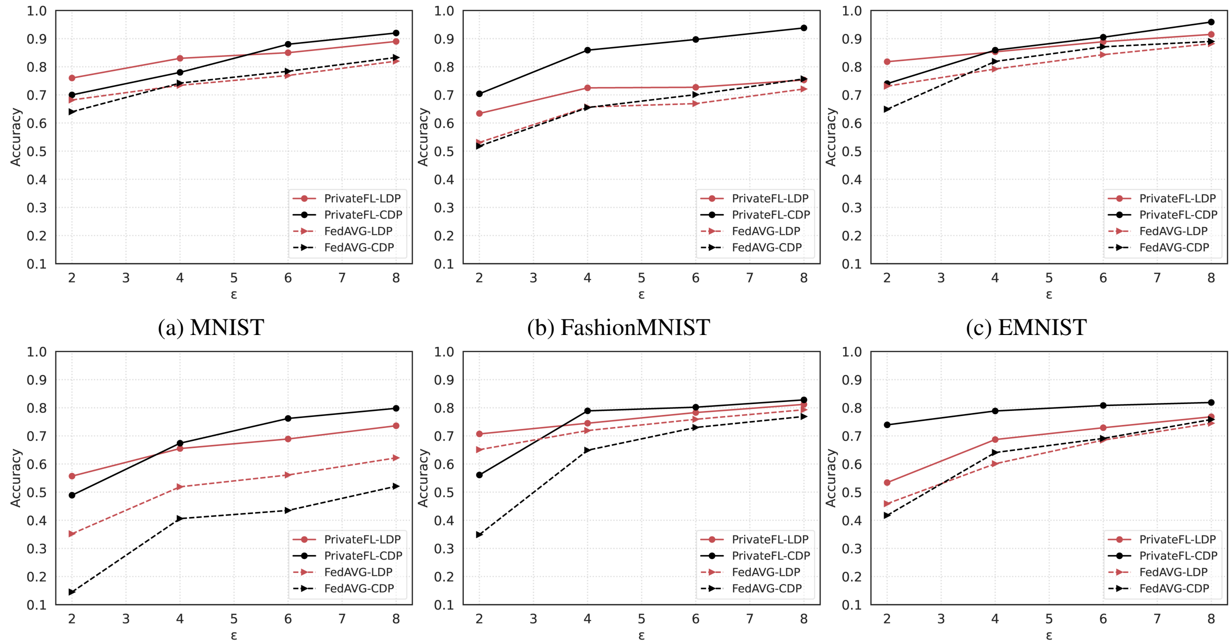
- [RQ1] How does PRIVATEFL improve LDP, CDP, and DDP on FedAVG and personalized FLs?
- [RQ2] What is the performance of existing DP-improving methods on FL, and how does PRIVATEFL further improve them as an add-on method?
- [RQ3] Why can PRIVATEFL improve FL's accuracy under DP?
- [RQ4] How does different client data distribution affect the performance of PRIVATEFL?
- [RQ5] How does different number of clients affect the performance of PRIVATEFL?
- [RQ6] How does PRIVATEFL perform in cross-device FL?

### 6.1 RQ1: FL+DP Accuracy Improvement

In this Research Question, we show that PRIVATEFL can improve FL model's accuracy with LDP, CDP, and DDP. Our evaluation starts from FedAVG [28] and then comes to two personalized FL, namely FedBN [26] and SCAFFOLD [33].

**6.1.1. Differentially Private FedAVG.** We first compare FedAVG and PRIVATEFL on both image and non-image datasets for LDP, CDP, and DDP. For LDP and CDP, we adopt the settings in Table 3. For DDP, we follow exactly the same setting as the original paper [5]. That is, we only evaluate DDP on EMNIST. We follow their setting to assign the training data to 3,400 clients and use a sample rate of 0.03 during training.

**LDP.** Figure 5 shows the testing accuracy of different methods when the privacy budget $\epsilon$ varies. First, PRIVATEFL-LDP improves FedAVG-LDP's accuracy by 2.3% to 10.2% on image dataset (6.3% on average), and by 11.4%–20.5% on a




(a) MNIST (b) FashionMNIST (c) EMNIST

(d) Purchase-100 (e) CH-MNIST (f) CIFAR-10

Figure 5: [RQ1-1] Testing accuracy comparison between PrivateFL and FedAVG + DP with different privacy budgets.

Table 4: [RQ1-1] Testing accuracy comparison of DDP [5] and PrivateFL-DDP with different privacy budgets.

| EMNIST | $\varepsilon = 2$ | $\varepsilon = 4$ | $\varepsilon = 6$ | $\varepsilon = 8$ |
|---|---|---|---|---|
| DDP [5] | 0.691 | 0.725 | 0.776 | 0.785 |
| PrivateFL-DDP | **0.776** | **0.806** | **0.839** | **0.845** |

non-image dataset, i.e., particularly Purchase-100 (14.5% on average). Second, we observe a big improvement on Purchase-100. The reasons are two-fold. On one hand, the feature values of Purchase-100 are boolean, which are easily perturbed by random noise, leading to more performance degradation (and hence the followup improvement). On the other hand, the Purchase-100 dataset has 100 classes, leading to more training data heterogeneity compared with other datasets. Third, the accuracy gap between PrivateFL-LDP and FedAVG-LDP becomes larger when $\varepsilon$ is smaller on majority of the datasets. This result aligns with our motivation example, i.e., Figure 1a: DP introduces more heterogeneity among the clients when $\varepsilon$ is smaller because the noise is larger.

**CDP.** Figure 5 also shows that PrivateFL-CDP improves FedAVG-CDP's accuracy by 3.4%–32.2% on image dataset (12.1% on average) and by 26.8%–34.4% on Purchase-100 (30.4% on average). First, a big improvement on Purchase-100 still holds. Second, compared with LDP, PrivateFL-CDP brings a bigger improvement over FedAVG-CDP on average. The reason is that CDP does not require adding noise on the personalized data transformation. Thus, it is more effective to mitigate the heterogeneity compared with LDP that adds noise on the transformation.

Table 5: [RQ1-2] Testing accuracy of (differentially private) personalized FL methods and their combinations with PrivateFL. Note that DP-SCAFFOLD is only compatible with LDP but not CDP. PP-SGD is only compatible with CDP but not LDP.

| DP | FL | $\varepsilon = 2$ | $\varepsilon = 4$ | $\varepsilon = 6$ | $\varepsilon = 8$ |
|---|---|---|---|---|---|
| CDP | FedBN [26] | 0.712 | 0.827 | 0.837 | 0.865 |
| | FedBN+PrivateFL | **0.829** | **0.916** | **0.934** | **0.938** |
| | PP-SGD [6] | 0.222 | 0.365 | 0.607 | 0.898 |
| | PP-SGD+PrivateFL | **0.228** | **0.394** | **0.651** | **0.924** |
| LDP | DP-SCAFFOLD [33] | 0.306 | 0.448 | 0.560 | 0.634 |
| | DP-SCAFFOLD+PrivateFL | **0.531** | **0.557** | **0.652** | **0.691** |
| | FedGN (a variant of FedBN [26]) | 0.672 | 0.723 | 0.751 | 0.802 |
| | FedGN+PrivateFL | **0.784** | **0.843** | **0.874** | **0.893** |

**DDP.** Table 4 shows that PrivateFL-DDP improves DDP's accuracy by 6.0%–8.5% on EMNIST dataset (7.2% on average) with different privacy budgets. The reason is that DDP adds local noise to each client, thus introducing client heterogeneity. We also observe a bigger improvement when $\varepsilon$ is smaller. The reason is the same as we discussed previously, i.e., a smaller $\varepsilon$ requires more noise thus introduces more heterogeneity. We note that the accuracy is generally lower than LDP/CDP because the DDP setting has 3,400 clients as opposed to 100 clients for the experiments on LDP/CDP.

We also found that in the same FL settings, DDP and CDP achieve similar accuracy. Thus, we mainly focus on LDP and CDP in the rest of our evaluation unless otherwise mentioned.

**6.1.2. Differentially Private Personalized FL.** Table 5 shows the accuracy of (differentially private) personalized

Table 6: [RQ2-1] Testing accuracy of different methods on CIFAR-10 and CIFAR-100 when different pre-trained encoders are available. $\varepsilon = \infty$ indicates non-private training.

| | Pretrained | DP | Method | $\varepsilon=2$ | $\varepsilon=4$ | $\varepsilon=6$ | $\varepsilon=8$ | $\varepsilon=\infty$ |
|---|---|---|---|---|---|---|---|---|
| CIFAR10 | ResNeXt on CIFAR-100 | CDP | FedAVG | 0.416 | 0.497 | 0.526 | 0.529 | 0.865 |
| | | | DN [44] | 0.480 | 0.549 | 0.562 | 0.579 | 0.911 |
| | | | DN+PrivateFL | **0.812** | **0.876** | **0.889** | **0.892** | **0.939** |
| | | LDP | FedAVG | 0.369 | 0.429 | 0.484 | 0.504 | 0.865 |
| | | | DN [44] | 0.402 | 0.526 | 0.564 | 0.598 | 0.911 |
| | | | DN+PrivateFL | **0.816** | **0.860** | **0.874** | **0.880** | **0.939** |
| | SimCLR on ImageNet | CDP | FedAVG | 0.438 | 0.496 | 0.556 | 0.562 | 0.854 |
| | | | DN [44] | 0.443 | 0.495 | 0.556 | 0.562 | 0.869 |
| | | | DN+PrivateFL | **0.836** | **0.854** | **0.866** | **0.883** | **0.934** |
| | | LDP | FedAVG | 0.322 | 0.417 | 0.493 | 0.516 | 0.854 |
| | | | DN [44] | 0.349 | 0.429 | 0.491 | 0.507 | 0.869 |
| | | | DN+PrivateFL | **0.724** | **0.805** | **0.816** | **0.842** | **0.934** |
| | CLIP on public data (400 million) | CDP | FedAVG | 0.861 | 0.870 | 0.896 | 0.902 | 0.925 |
| | | | DN [44] | 0.861 | 0.882 | 0.895 | 0.905 | 0.927 |
| | | | DN+PrivateFL | **0.933** | **0.945** | **0.953** | **0.957** | **0.961** |
| | | LDP | FedAVG | 0.761 | 0.780 | 0.797 | 0.827 | 0.925 |
| | | | DN [44] | 0.763 | 0.772 | 0.795 | 0.821 | 0.927 |
| | | | DN+PrivateFL | **0.867** | **0.882** | **0.899** | **0.919** | **0.961** |
| CIFAR100 | SimCLR on ImageNet | CDP | FedAVG | 0.021 | 0.023 | 0.031 | 0.036 | 0.499 |
| | | | DN [44] | 0.014 | 0.025 | 0.038 | 0.041 | 0.512 |
| | | | DN+PrivateFL | **0.560** | **0.612** | **0.628** | **0.644** | **0.802** |
| | | LDP | FedAVG | 0.064 | 0.072 | 0.078 | 0.103 | 0.499 |
| | | | DN [44] | 0.082 | 0.095 | 0.103 | 0.105 | 0.512 |
| | | | DN+PrivateFL | **0.682** | **0.711** | **0.716** | **0.732** | **0.802** |
| | CLIP on public data (400 million) | CDP | FedAVG | 0.023 | 0.026 | 0.048 | 0.062 | 0.635 |
| | | | DN [44] | 0.015 | 0.025 | 0.042 | 0.068 | 0.649 |
| | | | DN+PrivateFL | **0.594** | **0.663** | **0.674** | **0.774** | **0.839** |
| | | LDP | FedAVG | 0.345 | 0.413 | 0.423 | 0.435 | 0.635 |
| | | | DN [44] | 0.337 | 0.416 | 0.438 | 0.442 | 0.649 |
| | | | DN+PrivateFL | **0.711** | **0.744** | **0.758** | **0.768** | **0.839** |

FLs and their combinations with PrivateFL. We choose the same datasets, number of clients, and data distributions as those used in the original papers. For example, we use MNIST with 100 clients and two classes per client for DP-SCAFFOLD [33] and FedBN [26]. Similarly, we use EMNIST for PP-SGD [6] following the original paper's setting. Note that DP-SCAFFOLD is only applicable to LDP but not CDP, while PP-SDG is only applicable to CDP but not LDP because the personalization is not protected by DP. In addition, we replace the batch normalization (BN) in FedBN with group normalization (GN) under LDP following the implementation of Pytorch Opacus because BN does not work for DP. The variant is thus called FedGN.

We observe that PrivateFL can further improve accuracy of personalized FL methods for both CDP and LDP. For CDP and personalized FL, PrivateFL improves model's accuracy by 0.6%–11.7% with an average of 6.01%. For LDP and personalized FL, PrivateFL improves model's accuracy by 5.7%–22.5%. This is because personalized FL and PrivateFL capture different types of heterogeneity (e.g., local training data heterogeneity and DP-induced heterogeneity), and PrivateFL can be combined with personalized FL to further improve their accuracy. Take PP-SGD+PrivateFL as an example. PP-SGD separates local and global models, but DP-induced heterogeneity still exists in the global model.

Table 7: [RQ2-2] Testing accuracy when different activation functions are used.

| | DP | Method | $\varepsilon=2$ | $\varepsilon=4$ | $\varepsilon=6$ | $\varepsilon=8$ |
|---|---|---|---|---|---|---|
| MNIST | CDP | ReLU | 0.612 | 0.753 | 0.788 | 0.797 |
| | | Tanh [34] | 0.620 | 0.741 | 0.784 | 0.793 |
| | | Tanh+PrivateFL | **0.701** | **0.783** | **0.884** | **0.923** |
| | LDP | ReLU | 0.379 | 0.496 | 0.511 | 0.548 |
| | | Tanh [34] | 0.702 | 0.734 | 0.769 | 0.812 |
| | | Tanh+PrivateFL | **0.743** | **0.831** | **0.852** | **0.895** |
| EMNIST | DDP [5] | ReLU | 0.691 | 0.725 | 0.747 | 0.761 |
| | | Tanh [34] | 0.679 | 0.732 | 0.736 | 0.754 |
| | | Tanh+PrivateFL | **0.747** | **0.793** | **0.825** | **0.847** |

Table 8: [RQ2-3] Testing accuracy of per-round clip and per-step clip for CDP on MNIST.

| | $\varepsilon=2$ | $\varepsilon=4$ | $\varepsilon=6$ | $\varepsilon=8$ |
|---|---|---|---|---|
| Per-round-clip | 0.641 | 0.745 | 0.796 | 0.827 |
| Per-step-clip [29] | 0.652 | 0.763 | 0.774 | 0.829 |
| Per-step-clip+PrivateFL | **0.726** | **0.804** | **0.892** | **0.911** |

Therefore, PrivateFL can be used to help PP-SGD reduce such heterogeneity in the global model.

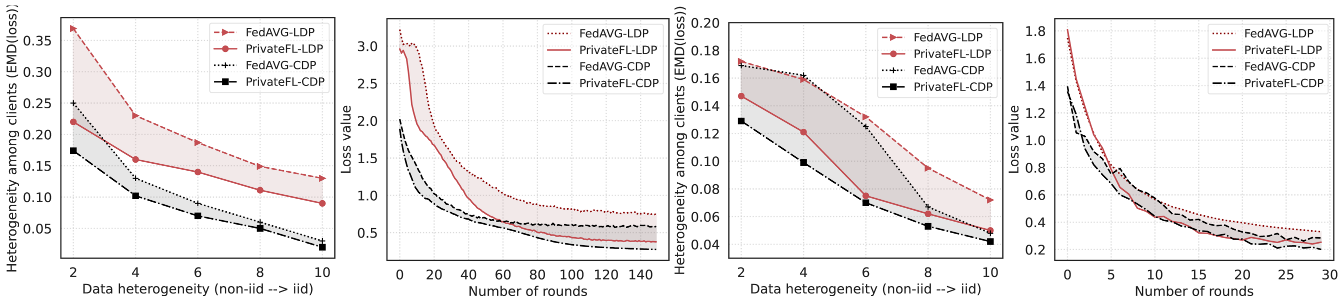
[RQ1] *Take-away*: PrivateFL improves model's accuracy under different DP variants and can be combined with personalized FLs to further improve accuracy.

### 6.2 RQ2: Combination with DP-improving Methods

In this Research Question, we show that PrivateFL complements existing methods to improve utility of DP and can be combined with them to further improve DP's utility.

**6.2.1. Pre-trained Encoder.** Previous works [27,44] showed that an encoder (i.e., a feature extractor) pre-trained on public data can be used to build more accurate differentially private classifiers. Specifically, we can use the encoder to extract features and train a differentially private linear classifier based on the features. We consider multiple pre-trained encoders including a ResNeXt classifier [50] pre-trained on CIFAR-100, a SimCLR encoder [8] pre-trained on unlabeled ImageNet, and CLIP [36], which is the state-of-the-art encoder pre-trained on 400 million image-text pairs. Given a pre-trained encoder, each client uses it to extract features for its training/testing inputs. Then, FL (e.g., FedAVG) trains a linear classifier based on the extracted features. *Data normalization (DN)* [44] further normalizes the features before using them for training. The mean and variance in DN for ResNeXt are collected from CIFAR-100, while for SimCLR and CLIP, both values are set as 0.5 following the DN paper.

Table 6 shows the accuracy of FedAVG, DN, and DN + PrivateFL in different settings. First, DN + PrivateFL consistently achieves the best accuracy, indicating that PrivateFL can be combined with existing pre-trained encoder based methods to further improve DP's utility. The reason is that FedAVG and DN do not consider DP-induced heterogeneity when training linear classifiers based on the extracted




(a) [MNIST] Heterogeneity (b) [MNIST] Convergence (c) [CIFAR-10] Heterogeneity (d) [CIFAR-10] Convergence

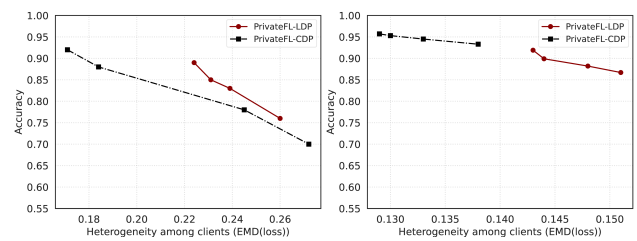
Figure 6: [RQ3] Illustration on why PRIVATEFL can improve the accuracy of FL+DP. (a)(c): Average EMD between the clients' local training loss distributions when each client has 2, 4, 6, 8, or 10 classes of data. (b)(d): testing loss of the global model as a function of training round. We use 3-layer DNN for MNIST and pre-trained CLIP encoder plus 1-layer DNN for CIFAR-10.

features. Interestingly, except for ResNext where DN can utilize the statistics collected from CIFAR-100, DN often achieves lower accuracy than FedAVG.

Second, PRIVATEFL can significantly improve the accuracy of private training when a pre-trained encoder is available, and reduce the accuracy gap between private and non-private training. Take CIFAR-10 as an example, the accuracy gap between non-private and private DN+PRIVATEFL with CDP ($\varepsilon = 8$) ranges from 0.4% (CLIP) to 5.1% (SimCLR). In comparison, the accuracy gap between non-private and private DN with CDP ($\varepsilon = 8$) ranges from 2.2% (CLIP) to 33.2% (ResNeXt). Third, CLIP consistently achieves the highest accuracy improvement. The reason is that CLIP is pre-trained on the largest dataset, which enables it to extract better features, compared to the other two pre-trained encoders.

**6.2.2. Activation Function Substitution.** Papernot et al. [34] shows that Tanh activation function can improve the accuracy of a model trained by DP-SGD. We follow the paper to replace ReLU activation function as Tanh in FedAVG. Table 7 shows the testing accuracy when different activation functions are used. First, we observe that Tanh introduces substantial accuracy improvement for LDP while minimal improvement for CDP/DDP. In some cases, Tanh even reduces the accuracy by around 1% for CDP/DDP. The reason is that Tanh can avoid gradient explosion for DP-SGD, which is used by LDP, and thus improves accuracy for LDP. However, CDP and DDP do not add noise during local training, and thus their performance does not explicitly depend on activation functions. Second, Tanh+PRIVATEFL consistently achieves the best accuracy, indicating that PRIVATEFL can be combined with existing activation function based methods to further improve DP's utility.

**6.2.3. Per-round Clip vs. Per-step Clip for CDP.** McMahan et al. [29] shows that we can improve the accuracy of CDP when a client clips its local model in each local training step. We apply this method to FedAVG+CDP. Table 8 shows the testing accuracy of per-round clip, per-step clip, and the combination of PRIVATEFL and per-step clip on MNIST. Our results align with McMahan et al., i.e., per-step clip improves accuracy for CDP. Moreover, PRIVATEFL can be combined with per-step clip to further improve accuracy.




(a) MNIST (b) CIFAR-10

Figure 7: [RQ3] DP-induced heterogeneity vs. accuracy

[RQ2] *Take-away:* PRIVATEFL can be combined with existing DP utility improvement methods to further boost utility by reducing DP-induced heterogeneity.

### 6.3 RQ3: Heterogeneity Reduction

In this Research Question, we explore why PRIVATEFL can improve the accuracy of DP in FL. In particular, we show that PRIVATEFL reduces additional heterogeneity introduced by DP, thus improving FL accuracy. We use two metrics to show the heterogeneity reduction. First, we use the average EMD between the clients' local training loss distributions as discussed in Section 2.3. Second, we use the testing loss of the global model. Our evaluation includes scenarios without and with a pre-trained encoder, i.e., a 3-layer DNN for MNIST trained with DP from scratch, and 1-layer DNN with a pre-trained CLIP encoder for CIFAR-10.

Figures 6a and 6c show the EMD of different methods when the training data heterogeneity changes from two classes to 10 classes per client. Our results align with our earlier motivations, i.e., PRIVATEFL reduces DP-induced heterogeneity and thus improves accuracy. Figure 7 explicitly shows the accuracy as a function of average EMD. In these experiments, each client has two classes of data and we vary the privacy budget $\varepsilon$ as 8, 6, 4, and 2. For each privacy budget, we obtain an average EMD and accuracy for a method. Thus, we obtain multiple accuracy-EMD pairs for each method, which

Table 9: [RQ4] Testing accuracy of FedAVG and PRIVATEFL with LDP and CDP on MNIST and CIFAR-10 datasets when each client has 2, 4, 6, 8, or 10 (i.i.d) classes of data.

| | DP | Method | 2 | 4 | 6 | 8 | 10 (i.i.d) |
|---|---|---|---|---|---|---|---|
| MNIST | CDP | FedAVG | 0.811 | 0.831 | 0.852 | 0.871 | 0.892 |
| | | PRIVATEFL | **0.922** | **0.911** | **0.909** | **0.904** | **0.906** |
| | LDP | FedAVG | 0.793 | 0.815 | 0.842 | 0.851 | 0.855 |
| | | PRIVATEFL | **0.886** | **0.882** | **0.876** | **0.868** | **0.872** |
| CIFAR-10 | CDP | FedAVG | 0.892 | 0.903 | 0.907 | 0.916 | 0.924 |
| | | PRIVATEFL | **0.949** | **0.942** | **0.936** | **0.933** | **0.932** |
| | LDP | FedAVG | 0.815 | 0.823 | 0.844 | 0.852 | 0.858 |
| | | PRIVATEFL | **0.911** | **0.889** | **0.892** | **0.874** | **0.883** |

are shown in Figure 7. Our results clearly show that accuracy is higher when heterogeneity (measured by EMD) is less severe. Second, Figure 6b and Figure 6d show that PRIVATEFL makes the global model converge faster, i.e., the testing loss decreases faster, which further explains why reducing heterogeneity improves accuracy.

Third, Figures 6a and 6c show that LDP introduces larger heterogeneity than CDP. The reason is that LDP clips gradients and adds noise during local training of each local model, while CDP does both steps on the trained local models. Figures 6b and 6d also show that the testing loss for LDP is larger than CDP, which again suggests that the heterogeneity introduced by LDP can decrease model's accuracy more.

> [RQ3] *Take-away*: PRIVATEFL improves the accuracy for DP in FL because it can decrease the heterogeneity introduced by DP among the clients.

### 6.4 RQ4: Client Data Heterogeneity

In this Research Question, we explore how the clients' local training data heterogeneity affects the performance of PRIVATEFL. We use MNIST and CIFAR-10 datasets. We set the privacy budget ($\varepsilon = 8, \delta = 1e-5$) to show results on a new privacy budget since our prior experiments are based on the default privacy budget. The other parameters are the same as those in Table 3. We assign 2, 4, 6, 8, or 10 classes of data to each client, where 10 classes of data per client indicate i.i.d. We use a pre-trained CLIP plus 1-layer DNN for CIFAR-10 and a 3-layer DNN for MNIST trained from scratch.

Table 9 shows the testing accuracy of FedAVG and PRIVATEFL in CDP and LDP when each client has different classes of training data. We have three observations. First, PRIVATEFL outperforms FedAVG consistently from non-i.i.d. to i.i.d. for both CDP and LDP with an average accuracy improvement of 4.7%. Second, PRIVATEFL under both CDP and LDP performs the best in the setting with 2 classes per client and the accuracy drops at most 3.7% when the clients' local data distribution becomes i.i.d. As a comparison, FedAVG performs the best in an i.i.d. setting and its accuracy drops at most 9.7% when the data distribution becomes non-i.i.d.

Table 10: [RQ5] Testing accuracy of FedAVG and PRIVATEFL with LDP and CDP on MNIST and CIFAR-10 datasets when the system has 50, 100, 200, or 500 clients.

| | DP | Method | 50 | 100 | 200 | 500 |
|---|---|---|---|---|---|---|
| MNIST | CDP | FedAVG | 0.716 | 0.787 | 0.733 | 0.664 |
| | | PRIVATEFL | **0.891** | **0.912** | **0.866** | **0.843** |
| | LDP | FedAVG | 0.866 | 0.802 | 0.732 | 0.664 |
| | | PRIVATEFL | **0.892** | **0.883** | **0.832** | **0.806** |
| CIFAR-10 | CDP | FedAVG | 0.814 | 0.882 | 0.866 | 0.836 |
| | | PRIVATEFL | **0.942** | **0.949** | **0.941** | **0.937** |
| | LDP | FedAVG | 0.815 | 0.818 | 0.865 | 0.843 |
| | | PRIVATEFL | **0.913** | **0.907** | **0.916** | **0.914** |

The reason is that PRIVATEFL can decrease the heterogeneity introduced by LDP and training data itself, and thus the accuracy gets improved more when the clients' data is more non-i.i.d. Third, PRIVATEFL brings more accuracy improvement on LDP than CDP in the i.i.d. setting, because LDP incurs heterogeneity even under the i.i.d. setting.

> [RQ4] *Take-away*: PRIVATEFL consistently improves accuracy when the clients' local training data has different heterogeneity.

### 6.5 RQ5: Number of Clients

In this Research Question, we explore how different number of clients (ranging from 50 to 500) affects the performance of PRIVATEFL. We use MNIST and CIFAR-10 datasets with the default parameter settings except ($\varepsilon = 8, \delta = 1e-5$). Each client has 2 classes of data. Model architectures are the same as those in Section 6.4.

Table 10 shows the testing accuracy of FedAVG and PRIVATEFL when the number of clients varies. We observe that PRIVATEFL consistently outperforms FedAVG for both CDP and LDP especially when the system has more clients. For instance, the accuracy gap between PRIVATEFL and FedAVG increases from 2.6% with 50 clients to 14.2% with 500 clients on MNIST for LDP. We believe this is partially because more clients indicate more heterogeneity.

> [RQ5] *Take-away*: PRIVATEFL consistently improves the accuracy of FL under DP when the system has different number of clients.

### 6.6 RQ6: Cross-device FL

In this Research Question, we evaluate whether PRIVATEFL can improve the accuracy of cross-device FL, e.g., each client only participates in one or two training rounds. Our experiment setting is as follows. We use the MNIST dataset with the default parameter settings. Each client has two classes of data. We train the global model for a total of 10 or 20 rounds depending on whether each client participates in one or two training rounds. The sample rate is 0.1 in each training round.

Table 11: [RQ6] Testing accuracy of FedAVG and PRIVATEFL on MNIST for cross-device FL, where each client participates in one or two training rounds.

| DP | FL | Round | $\varepsilon=2$ | $\varepsilon=4$ | $\varepsilon=6$ | $\varepsilon=8$ |
|---|---|---|---|---|---|---|
| CDP | FedAVG | 1 | 0.421 | 0.545 | 0.563 | 0.612 |
| | | 2 | 0.483 | 0.619 | 0.641 | 0.653 |
| | PRIVATEFL | 1 | **0.454** | **0.577** | **0.586** | **0.633** |
| | | 2 | **0.527** | **0.677** | **0.695** | **0.714** |
| LDP | FedAVG | 1 | 0.453 | 0.582 | 0.604 | 0.610 |
| | | 2 | 0.501 | 0.608 | 0.631 | 0.639 |
| | PRIVATEFL | 1 | **0.502** | **0.592** | **0.628** | **0.644** |
| | | 2 | **0.529** | **0.640** | **0.683** | **0.728** |

Table 11 shows the accuracy of FedAVG and PRIVATEFL for both LDP and CDP when the privacy budget $\varepsilon$ varies. PRIVATEFL can improve the accuracy of FedAVG from 2.7% to 10.9% for both CDP and LDP under such a cross-device FL setting. Note that the accuracy improvement of PRIVATEFL is larger when each client participates in two training rounds. The reason is that our personalized data transformation can better fit the local training data and reduce DP-induced heterogeneity in such scenarios.

[RQ6] *Take-away*: PRIVATEFL improves the accuracy of cross-device FL + DP.

## 7 Related Work

**Federated Learning (FL).** FL [21, 24, 45, 46, 52] is a collaborate learning setting that allows multiple clients to train a model without sharing local training data. A global server is to collect local model from each client then aggregate them as the global model. Although there is no direct raw data sharing between clients and server, FL is vulnerable to privacy attacks [7]. For instance, Geiping et al. [14] show that a malicious server can recover clients' training data via gradient inversion attack; and Nasr et al. [32] propose that malicious clients or server can launch membership inference attacks. Apart from privacy leakage, data heterogeneity is also a key challenge in FL. Achieving a good accuracy on heterogeneous clients' data in FL has been shown as a challenge [25, 38, 55]. Although there are existing personalized FL methods [19, 26, 53, 54] addressing data heterogeneity problem, they did not consider DP's impacts.

**Differential Privacy in FL.** Differential privacy has been applied in FL [15, 29, 31, 40, 43]. There are three kinds of differential privacy applications in FL, local differential privacy (LDP) [43, 47], central differential privacy (CDP) [15, 29], and distributed differential privacy (DDP) [5, 10, 18]. LDP allows each client to make its local model differentially-private before sending it to the global server, e.g., clients can adopt DP-SGD [4] locally so that they can have their own privacy budgets. CDP requires a trustworthy server which makes the global model differentially-private after receiving all local models. DDP adds local noise after training and relies on secure aggregation to achieve a comparable privacy guarantee with CDP. However, both CDP and LDP cause significant accuracy drop in FL as demonstrated by Naser et al. [31], and DDP [5] has a similar performance with CDP as well as a similar accuracy drop.

Recently, Noble et al. [33] combine LDP with existing personalized FL algorithm SCAFFOLD [19] to show accuracy improvement on heterogeneous data. Stevens et al. [42] utilize learning with error to improve communication efficiency in DDP scenarios. PP-SGD [6] introduces a personalization parameter to balance local learning without DP and global learning with DP, thus improving the accuracy. As a comparison, PP-SGD is only applicable to CDP because the local personalization does not have DP guarantees. Furthermore, PP-SGD's utility improvement under CDP stems from personalization and therefore PRIVATEFL can further improve PP-SGD's performance by reducing DP-induced heterogeneity as shown in our evaluation. In summary, PRIVATEFL addresses the heterogeneity induced by DP and improves the accuracy for LDP, CDP, DDP, and PP-SGD.

**Differential Privacy in Centralized Learning.** Differential privacy has been studied extensively in centralized learning [12, 13, 17, 48, 51]. Abadi et al. propose DP-SGD [4] to make Stochastic Gradient Descent (SGD) differentially-private via clipping and perturbing gradients during training. However, DP-SGD sacrifices accuracy compared with SGD. Researchers have tried to address this issue both theoretically and empirically. Theoretically, Abadi et al. propose moment accountant [4] to derive a tighter privacy budget; and Mironov [30] propose Renyi-DP with a more strict upper-bound of privacy budget. Empirically, Papernot et al. [34] explore different activation functions, e.g., using *Tanh* instead of *ReLU* as the activation function can improve accuracy in DP; and Tramer et al. [44] and Liu et al. [27] show that an encoder pre-trained on public data can be used as a feature extractor to train more accurate differentially private classifiers. As we demonstrated in experiments, those methods cannot address the DP-induced heterogeneity and thus our PRIVATEFL can be combined with them to further improve accuracy of FL+DP.

## 8 Conclusion

In this work, for the first time, we find that DP introduces additional heterogeneity to the clients in FL, thus hampering accuracy. Then, we show that personalized data transformation on the clients can mitigate the DP-induced heterogeneity and thus improve the accuracy of differentially private FL for different variants of DP including LDP, DDP, and CDP. Moreover, linear transformation outperforms other alternatives. Our personalized data transformation can be combined with personalized FL methods and methods on improving accuracy of DP to further improve the accuracy of differentially private FL.

## Acknowledgment

We would like to thank the anonymous shepherd and anonymous reviewers for their helpful comments and feedback. This work was supported in part by Johns Hopkins University Institute for Assured Autonomy (IAA) with grants 80052272 and 80052273, and National Science Foundation (NSF) under grants CNS-21-31859, CNS-21-12562, CNS-19-37786, and CNS18-54000. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or JHU-IAA.

## References

[1] [github] differentially private federated learning: A client-level perspective. https://github.com/SAP-samples/machine-learning-diff-private-federated-learning.

[2] [github] distributed differential privacy. https://github.com/google-research/federated/tree/master/distributed_dp.

[3] Opacus. https://github.com/pytorch/opacus.

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

[5] Naman Agarwal, Peter Kairouz, and Ziyu Liu. The skellam mechanism for differentially private federated learning. *Advances in Neural Information Processing Systems*, 2021.

[6] Alberto Bietti, Chen-Yu Wei, Miroslav Dudik, John Langford, and Steven Wu. Personalization improves privacy-accuracy tradeoffs in federated learning. In *International Conference on Machine Learning*, 2022.

[7] Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. When the curious abandon honesty: Federated learning is not private. *arXiv preprint arXiv:2112.02918*, 2021.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

[9] Xiangyi Chen, Steven Z. Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. In *Advances in Neural Information Processing Systems*, 2020.

[10] Albert Cheu, Adam D. Smith, Jonathan R. Ullman, David Zeber, and Maxim Zhilyaev. Distributed differential privacy via shuffling. In *Advances in Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2019.

[11] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, 2017.

[12] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual international conference on the theory and applications of cryptographic techniques*, 2006.

[13] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 2014.

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

[15] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[16] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *ACM SIGSAC conference on computer and communications security*, 2017.

[17] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *IEEE Symposium on Security and Privacy (SP)*, 2019.

[18] Peter Kairouz, Ziyu Liu, and Thomas Steinke. The distributed discrete gaussian mechanism for federated learning with secure aggregation. In *International Conference on Machine Learning*, 2021.

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

[20] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 2016.

[21] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[23] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[24] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 2020.

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

[26] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021.

[27] Hongbin Liu, Wenjie Qu, Jinyuan Jia, and Neil Zhenqiang Gong. Pre-trained encoders in self-supervised learning improve secure and privacy-preserving supervised learning. *arXiv preprint arXiv:2212.03334*, 2022.

[28] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 2017.

[29] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[30] Ilya Mironov. Rényi differential privacy. In *IEEE Computer Security Foundations Symposium (CSF)*, 2017.

[31] Mohammad Naseri, Jamie Hayes, and Emiliano Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. In *Network and Distributed System Security Symposium*, 2022.

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

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

[34] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy, 2020.

[35] Dario Pasquini, Danilo Francati, and Giuseppe Ateniese. Eluding secure aggregation in federated learning via model inconsistency. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2022.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[37] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 2000.

[38] F. Sattler, S. Wiedemann, K. R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[39] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *International Conference on Machine Learning (ICML)*, 2021.

[40] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *ACM SIGSAC conference on computer and communications security*, 2015.

[41] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*, 2017.

[42] Timothy Stevens, Christian Skalka, Christelle Vincent, John Ring, Samuel Clark, and Joseph Near. Efficient differentially private secure aggregation for federated learning via hardness of learning with errors. In *USENIX Security Symposium*, 2022.

[43] Lichao Sun, Jianwei Qian, and Xun Chen. Ldp-fl: Practical private aggregation in federated learning with local differential privacy. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[44] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

[45] Nguyen H. Tran, Wei Bao, Albert Zomaya, Minh N. H. Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE Conference on Computer Communications*, 2019.

[46] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *ACM Workshop on Artificial Intelligence and Security*, 2019.

[47] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *ACM International Workshop on Edge Systems, Analytics and Networking*, 2020.

[48] Di Wang and Jinhui Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. In *AAAI Conference on Artificial Intelligence*, 2019.

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

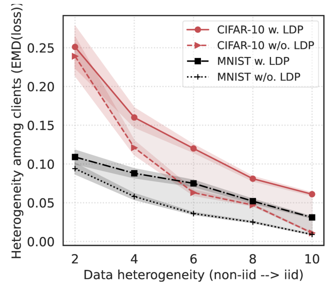
[50] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

[51] Depeng Xu, Wei Du, and Xintao Wu. Removing disparate impact on model accuracy in differentially private stochastic gradient descent. In *ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2021.

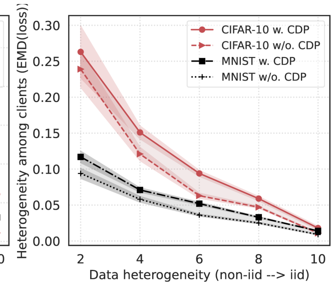
[52] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019.

[53] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*. PMLR, 2020.

[54] Haolin Yuan, Bo Hui, Yuchen Yang, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Addressing heterogeneity in federated learning via distributional transformation. In *European Conference of Computer Vision (ECCV)*, 2022.




(a) [LDP] Data Heterogeneity




(b) [CDP] Data Heterogeneity

Figure 8: DP introduces additional client heterogeneity during training when the clients' local training data heterogeneity varies.

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

# Appendices

## A Additional Motivation Experiments

We show additional motivation experiments when the clients' local training data heterogeneity varies. We assume ten clients with different levels of local training data heterogeneity, i.e., ranging from two to ten classes per client, which represents different data distributions from non-i.i.d. to i.i.d. The privacy budget $\varepsilon$ is set as 8 and we train FL for 100 rounds.

Figure 8 shows the heterogeneity introduced by LDP/CDP among the ten clients as the local training data heterogeneity changes from non-i.i.d. to i.i.d. We observe that LDP/CDP introduces additional heterogeneity no matter how the clients' local training data are distributed. Moreover, LDP introduces larger additional heterogeneity while CDP introduces smaller additional heterogeneity as the clients' local training data are closer to i.i.d.

## B Transformed Samples

Figure 9 shows several transformed samples on the ten clients in training rounds [0 (original), 30, 70, 100] when linear transformation is used, where the dataset is CIFAR-10 and the detailed experimental setup is the same as those in Table 3. Our linear transformation keeps the features of the samples while changing pixel values to reduce clients' heterogeneity.

Figure 10 shows transformed samples of MNIST and Fashion-MNIST datasets when PRIVATEFL uses different transformation functions. $\alpha x + \beta$ keeps the features of an image, *Sigmoid* only remains the edge information, and *Conv2d* blurs the images.

## C Datasets

We use the following datasets in our evaluation:

- **MNIST** [23]. The MNIST dataset contains 60,000 28x28 grayscale hand-written digits images in 10 different classes. We use 50,000 for training and 10,000 for testing.

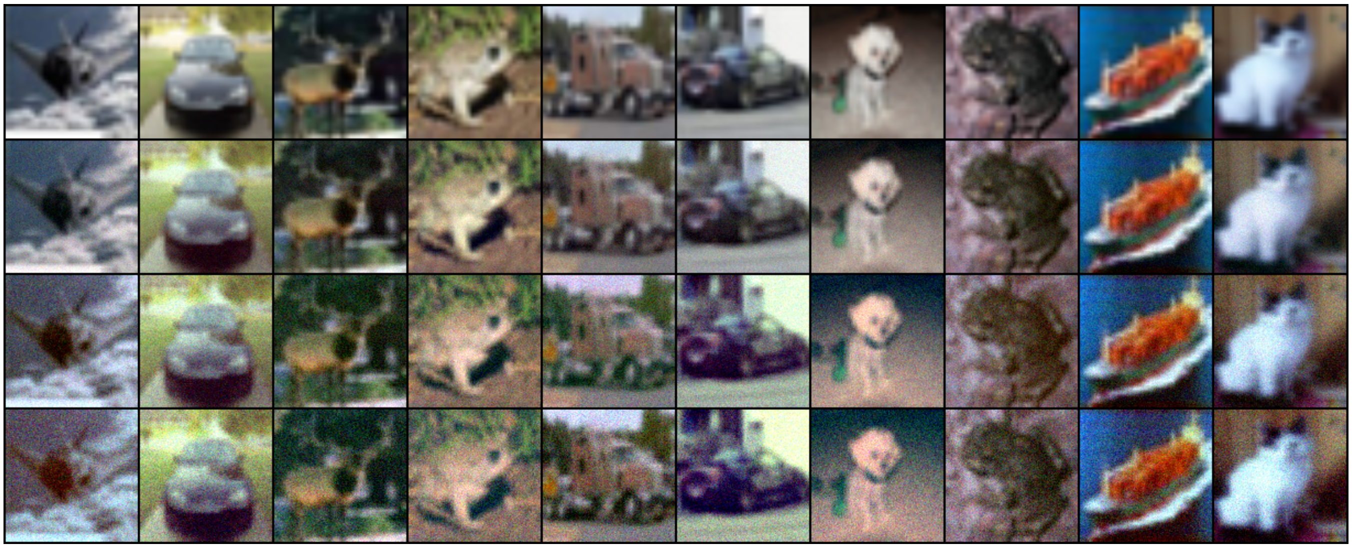
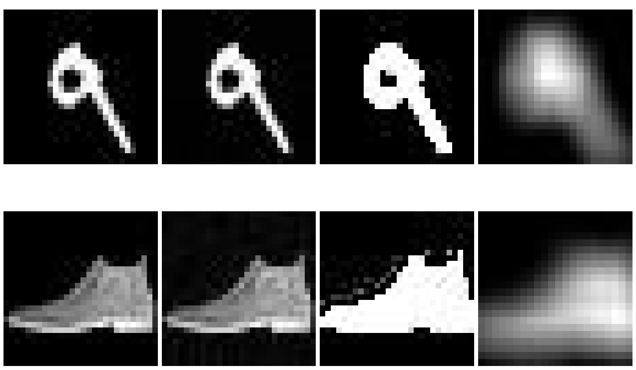


Figure 9: Transformed samples in CIFAR-10 during training in PRIVATEFL with linear transformation. The columns represent the ten clients, and the rows represent training rounds [0 (original), 30, 70, 100] from top to bottom.



(a) $x$ (b) $\alpha x + \beta$ (c) *Sigmoid* (d) *Conv2d*

Figure 10: Transformed samples on MNIST (first row) and Fashion-MNIST (second row) datasets in the last training round when PRIVATEFL uses different transformation functions.

- **Fashion-MNIST** [49]. The Fashion-MNIST dataset contains 70,000 28x28 grayscale images of fashion products in 10 different classes. We use 60,000 for training and 10,000 for testing.
- **EMNIST** [11]. The EMNIST dataset contains 280,000 28x28 grayscale hand-written digits images in 10 different classes. We use 240,000 for training and 40,000 for testing.
- **CH-MNIST** [20]. The CH-MNIST dataset contains eight classes of 5,000 histology tiles images (150x150) from patients with colorectal cancer. We use 4,500 for training and 500 for testing.
- **CIFAR-10** [22]. The CIFAR-10 dataset contains 60,000 32x32 color images in 10 different classes. We use 50,000 for training and 10,000 for testing.
- **CIFAR-100** [22]. The CIFAR-100 dataset contains 60,000 32x32 color images in 100 different classes. We use 50,000 for training and 10,000 for testing.
- **Purchase-100.** The Purchase-100 dataset contains 220,000 samples of 100 shoppers. It is a non-image dataset from Kaggle's "Aquired Valued Shoppers Challenge". We follow the data pre-processing in the previous work [41] and use 176,000 samples for training and 44,000 samples for testing.