FedAUXfdp: Differentially Private One-Shot Federated Distillation

Haley Hoech[1], Roman Rischke[1], Karsten Müller[1], Wojciech Samek[1]

[1]Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute
{haley.hoech, roman.rischke, karsten.mueller, wojciech.samek}@hhi.fraunhofer.de

Abstract

Federated learning suffers in the case of "non-iid" local datasets, i.e., when the distributions of the clients' data are heterogeneous. One promising approach to this challenge is the recently proposed method FedAUX, an augmentation of federated distillation with robust results on even highly heterogeneous client data. FedAUX is a partially (ϵ, δ)-differentially private method, insofar as the clients' private data is protected in only part of the training it takes part in. This work contributes a *fully differentially private* extension, termed FedAUX*fdp*. In experiments with deep networks on large-scale image datasets, FedAUXfdp with strong differential privacy guarantees performs significantly better than other equally privatized SOTA baselines on non-iid client data in just a single communication round. Full privatization results in a negligible reduction in accuracy at all levels of data heterogeneity.

1 Introduction

Federated learning (FL) is a form of decentralized machine learning, in which a global model is formed by an orchestration server aggregating the outcome of training on a number of local client models without any sharing of their private training data [McMahan et al., 2017]. Interest in federated learning has increased recently for its privacy and communication-efficiency advantages over centralized learning on mobile and edge devices [Li et al., 2019, Sattler et al., 2021c]. A classical mechanism for model aggregation in FL is federated averaging (FedAVG), where the locally trained models are weighted proportionally to the size of the local dataset. In each communication round of federated averaging, weight updates of the clients' local models are sent to the orchestration server, averaged by the server, and the average is sent back to the federation of clients to initialize the next round of training [McMahan et al., 2017].

Federated ensemble distillation (FedD), an often even more communication-efficient and accurate alternative to FedAVG, uses knowledge distillation to transfer knowledge from clients to server [Itahara et al., 2020, Lin et al., 2020, Chen and Chao, 2020, Sattler et al., 2021b]. In FedD, clients and server share a public dataset auxiliary to the clients' private data. The clients communicate the output of their privately trained models on the public distillation dataset to the server, which uses the average of these outputs as supervision for the distillation data in training the global model. In comparison to federated averaging, federated ensemble distillation offers additional privacy, as direct white box attacks are not possible for example, and allows combining different model architectures, making it appealing in an Internet-of-Things ecosystem [Li et al., 2020, Chang et al., 2019, Li et al., 2021].

FedAUX is an augmentation of federated distillation, which derives its success from taking full advantage of the AUXiliary data. FedAUX uses this auxiliary data for model pretraining and relevance weighting. To perform the weighting, the clients' output on each data point of the distillation

dataset is individually weighted by a measure of similarity between that distillation datapoint and the client's local data, called a 'certainty score'. Weighting the outputs by the scores prioritizes votes from clients whose local data is more similar to the auxiliary/distillation data.

A major challenge of federated learning is performance when the distributions of the clients' data are heterogeneous, i.e. performance on "non-iid" data, as is often the circumstance in real-world applications of FL [Kairouz et al., 2021]. FedAUX overcomes that challenge, performing remarkably more efficiently on non-iid data than other state-of-the-art federated learning methods, federated averaging, federated proximal learning, Bayesian federated learning, and federated ensemble distillation. For example on MobilenetV2, FedAUX achieves 64.8% server accuracy, while even the second-best method only achieves 46.7% [Sattler et al., 2021a].

Despite its privacy benefits, federated distillation still presents a privacy risk to clients participating [Papernot et al., 2017]. Data-level differential privacy protects the clients' data by limiting the impact of any individual datapoint on the model and quantifies the privacy loss associated with the method. Both governments and private institutions are increasingly interested in securing their data using differential privacy. [Sattler et al., 2021a] train two models on private data, but only privatize the scoring model, leaving the data participating in the classification model exposed. In this work, we add a local, data-level (ϵ, δ)-differentially private mechanism for this second model and give an upper bound on the L_2-sensitivity of regularized multinomial logistic regression. By appropriately modifying the FedAUX method, we contribute a fully privatized version of FedAUX.

In results with deep neural networks on large scale image datasets at an $(\epsilon = 0.6, \delta = 2 * 10^{-5})$ level of differential privacy we compare fully differentially private FedAUXfdp with two privatized baselines, federated ensemble distillation and federated averaging in a single communication round. FedAUXfdp outperforms these baselines dramatically on the heterogeneous client data. We also see a negligible reduction in accuracy of applying this strong amount of differential privacy to the modified FedAUX method.

In Section 3 we outline the original FedAUX, in Section 4 we explain our extension, including our privacy mechanism as well as background on differential privacy, and in Section 5 we detail the experimental set-up and highlight important results.

2 Related Work

Our method extends [Sattler et al., 2021a], who contributed a semi-differentially private FedAUX method. For a discussion of works related to the non-privacy aspects of FedAUX, we refer to their paper.

Cynthia Dwork introduced differential privacy [Dwork and Roth, 2014] and [Kasiviswanathan et al., 2008] local differential privacy. Differential privacy bounds were greatly improved with the introduction of the moments accountant in [Abadi et al., 2016].

In addition to quantifying privacy loss, differential privacy protects provably against membership inference attacks [Shokri et al., 2017, Choquette-Choo et al., 2021], in which an adversary can determine if a data point participated in the training of a model. This can pose a privacy threat, for example, if participation in model training could imply a client has a particular disease or other risk factor. Alternatives for privatization in general include secure multi-party computation or homomorphic encryption, though neither protect against membership inference attacks [Shokri et al., 2017]. Others have combined local differential privacy and federated learning, notably [Geyer et al., 2018, McMahan et al., 2018]. While [Sun and Lyu, 2021] combined federated model distillation with differential privacy, they only attain robust results on non-iid data when the client and distillation data contains the same classes.

3 FedAUX

3.1 Method

In FedAUX, there are two actors, the clients and the orchestration server. Each client, $i = 1, \ldots, n$, has its own private, local, labeled dataset D_i. Auxiliary to the client data, is a public, unlabeled dataset D_{aux}. The auxiliary data is further split into the negative data D^-, used in training the certainty score models, and the distillation data $D_{distill}$, used for knowledge distillation.

There are three types of models, the clients' scoring models, the clients' classification models, and the server's global model, which can all be decomposed into a feature extractor h and linear or logistic regression classification head. Whether the full model or just the classification head is trained varies by model and we outline this next. In FedAUX four kinds of training are conducted (See Figure 1):

1. **Feature extractor.** Unsupervised pretraining with the public auxiliary data D_{aux} on the server to obtain the feature extractor, h_0, which is sent to the clients and initialized in all their models as well as the server's.

2. **Scoring model heads.** Supervised training of the scoring model classification heads s_i of all clients, in combination with the frozen feature extractor h_0 to generate scoring models $f_i = s_i \circ h_0$. Each training is a binary logistic regression on the extracted features of their private local data and the public negative data $h_0(D_i \cup D^-)$.

3. **Classification models.** Supervised training of the clients' full classification models $g_i = c_i \circ h_i$, consisting of a feature extractor h_i (initialized with h_0 from the pretraining) and linear classification head c_i, on their local datasets D_i.

4. **Server model.** Supervised training of the server's full model S, consisting of a feature extractor h (initialized with h_0 from the pretraining) and linear classification head. The server calculates an initial weight update of the clients' average class model weight updates from their training round. For the server's training, the input data X is the unlabeled $D_{distill}$ and the supervision Y a $(|D_{distill}| \times n_{classes})$-dimensional matrix of the softlabel output of the class model $g_i(D_{distill})$, weighted by a certainty score for each distillation datapoint. The certainty scores are the output of the (ϵ, δ)-differentially privatized scoring model on the distillation data $f_i(D_{distill})$, measures of similarity between each distillation data point and the client's local data. Each entry in Y is:

$$\frac{\sum_i f_i(x) \cdot g_i(x)}{\sum_i f_i(x)}, \text{for } x \in D_{distill}. \tag{1}$$

3.2 Privacy

Participating in the training of the scoring classification heads and classification models presents a privacy risk to the private data of the clients. In FedAUX, the scoring heads are sanitized using an (ϵ, δ)-differentially private sanitization mechanism. FedAUX's mechanism for privatizing the scoring model is based on freezing the feature extractor and using a logistic classification head. As the feature extractor was trained on public data, only sanitizing this head is required to yield a differentially private model. Further, using the L-BFGS optimizer in sci-kit learn's logistic regression guarantees finding optimal weights for the logistic regression heads. In FedAUXfdp we privatize the classification models in a similar fashion. This thereby makes the server models learned in FedAUXfdp fully differentially private, as discussed in Section 4, with the specific privacy mechanism outlined in Section 4.2.

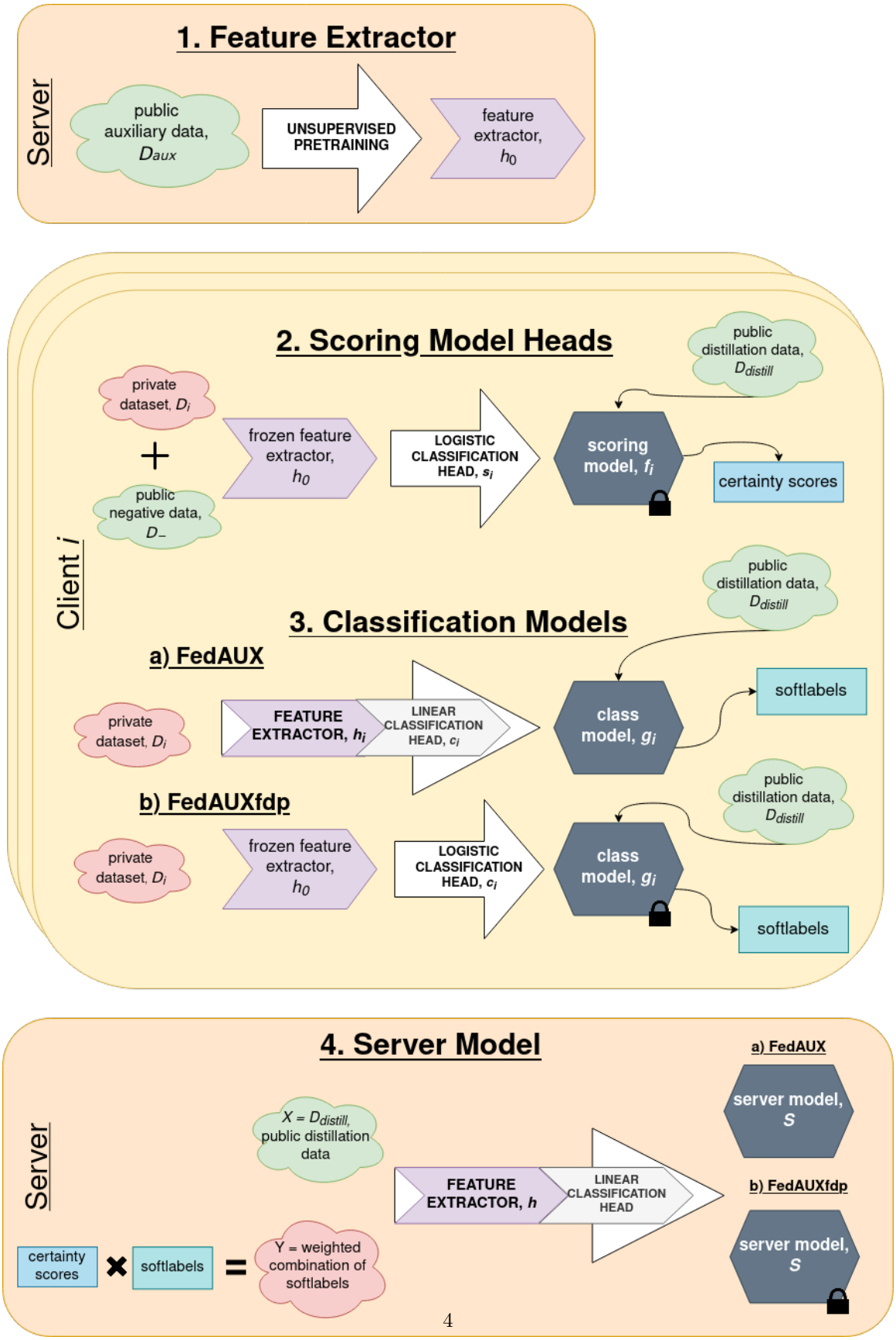

Figure 1: Overview of FedAUX and FedAUXfdp training

4 FedAUXfdp

In the fully differentially private version of FedAUX, we adapt the training of the classification and server models as follows. Rather than training the full client models, we freeze the feature extractors and train only the classification heads using a multinomial logistic regression on extracted features of the client's local dataset D_i. As communicating model updates to the server poses a privacy threat, we no longer initialize the server with the averaged weight update of the clients. Accordingly, step three in the process is changed as follows:

3. **Classification model.** Supervised training of the classification model heads c_i of the clients, combined with the frozen feature extractor h_0, to generate class models $g_i = c_i \circ h_0$. Each is a multinomial logistic regression on the extracted features of their private local data $h_0(D_i)$. See Figure 1.

As with the scoring models in the original FedAUX, freezing the feature extractors, which have been trained on public data, allows us to make the models differentially private by simply sanitizing the classification heads. Again, we opt for logistic classification heads because the L-BFGS optimizer in sci-kit learn's logistic regression guarantees convergence to globally optimal weights of the logistic regression.

We formulate the training of these classifiers as regularized empirical risk minimization problems.

4.1 Regularized Empirical Risk Minimization

Let $\boldsymbol{\beta} := (\boldsymbol{\beta}_1^T, \dots, \boldsymbol{\beta}_C^T)^T \in \mathbb{R}^{C(p+1)}$ with $\boldsymbol{\beta}_k := (\beta_{k,0}, \dots, \beta_{k,p})^T \in \mathbb{R}^{p+1}$ be the vector of trainable parameters of the regularized multinomial logistic regression problem with C classes

$$\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}, h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} -\log(p_{y_i}(h(\mathbf{x}_i))) + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2 \tag{2}$$

with softmax function

$$p_{y_i}(\boldsymbol{\beta}, h(\mathbf{x}_i)) = \frac{\exp(\boldsymbol{\beta}_{y_i}^T h(\mathbf{x}_i))}{\sum_{k=1}^{C} \exp(\boldsymbol{\beta}_k^T h(\mathbf{x}_i))}$$

for a labeled data point $(\mathbf{x}_i, y_i)$ from a dataset D.

Thereby, $h(\mathbf{x}_i) \in \mathbb{R}^{p+1}$ is an extracted feature vector with the first coordinate being a constant for the bias term $\beta_{k,0}$, and $y_i \in \{1, \dots, C\}$ the corresponding class label. We assume w.l.o.g. that

$$\|h(\mathbf{x})\|_2 \leq 1. \tag{3}$$

To fulfill this assumption, we normalize the input features for the logistic regression problem as follows

$$\tilde{h}(\mathbf{x}) := h(\mathbf{x}) \Big(\max_{\mathbf{x} \in D} \|h(\mathbf{x})\|_2 \Big)^{-1}. \tag{4}$$

4.2 Privacy

We privatize the classification models using (ϵ, δ)-differential privacy. Informally, differential privacy anonymizes the client data in this context, insofar as with very high likelihood the results of the model would be very similar regardless whether or not a particular data point participates in training [Dwork and Roth, 2014].

4.2.1 Definitions

Definition 1. *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy, if for any two adjacent inputs D_1 and D_2 that only differ in one element and for any subset of outputs $S \subseteq \mathcal{R}$,*

$$P[D_1 \in S] \leq \exp(\epsilon) P[\mathcal{M}(D_2) \in S] + \delta.$$

We use the Gaussian mechanism, in which a specific amount of Gaussian noise is added relative to the l^2-*sensitivity* [Dwork and Roth, 2014] and according to pre-selected ϵ and δ values.

Definition 2. *For $\epsilon \in (0, 1)$, $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c\Delta_2(\mathcal{M})/\epsilon$ is (ϵ, δ)-differentially private.*

Definition 3. *For two datasets, D_1, D_2 differing in one datapoint, the L_2-sensitivity is*

$$\Delta(\mathcal{M}) = \max_{D_1, D_2 \in \mathcal{D}} \|\mathcal{M}(D_1) - \mathcal{M}(D_2)\|_2$$

4.2.2 Sensitivity of the Classification Models

We contribute the following theorem for the L_2-sensitivity of regularized multinomial logistic regression (2), which generalizes a corollary from [Chaudhuri et al., 2011] .

Theorem 1. *The L_2-sensitivity of regularized multinomial logistic regression, as defined in* (2)*, is at most $\frac{2\sqrt{C}}{\lambda |D|}$.*

Proof. W.l.o.g. we set $h(\mathbf{x}) = \mathbf{x}$ in this proof. Let $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$ and $D' = (D \setminus \{(\mathbf{x}_N, y_N)\}) \cup \{(\mathbf{x}'_N, y'_N)\}$. That is, D and D' differ in exactly one data point. Furthermore, let

$$\boldsymbol{\beta}_1^* = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}, D) \tag{5}$$

$$\boldsymbol{\beta}_2^* = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta}, D'). \tag{6}$$

The goal is to show that $\|\boldsymbol{\beta}_1^* - \boldsymbol{\beta}_2^*\|_2 \leq \frac{2\sqrt{C}}{\lambda N}$. We define

$$\begin{aligned} d(\boldsymbol{\beta}) &:= J(\boldsymbol{\beta}, D') - J(\boldsymbol{\beta}, D) \\ &= \frac{1}{N} \left(l(\boldsymbol{\beta}, \mathbf{x}'_N) - l(\boldsymbol{\beta}, \mathbf{x}_N) \right), \end{aligned} \tag{7}$$

with the log-softmax loss function

$$l(\boldsymbol{\beta}, \mathbf{x}) := -\log(p_y(\boldsymbol{\beta}, \mathbf{x})) \tag{8}$$

for an arbitrary data point $(\mathbf{x}, y)$.

With

$$\nabla_{\boldsymbol{\beta}}\, l(\boldsymbol{\beta}, \mathbf{x}) = -\frac{\nabla_{\boldsymbol{\beta}}\, p_y(\boldsymbol{\beta}, \mathbf{x})}{p_y(\boldsymbol{\beta}, \mathbf{x})} \tag{9}$$

we obtain

$$\frac{\partial\, p_k(\boldsymbol{\beta}, \mathbf{x})}{\partial\, \boldsymbol{\beta}_k} = \frac{\exp(\boldsymbol{\beta}_k^T \mathbf{x}) \cdot \sum_{j \neq k} \exp(\boldsymbol{\beta}_j^T \mathbf{x})}{(\sum_j \exp(\boldsymbol{\beta}_j^T \mathbf{x}))^2} \mathbf{x} \tag{10}$$

$$\frac{\partial\, p_k(\boldsymbol{\beta}, \mathbf{x})}{\partial\, \boldsymbol{\beta}_{\ell \neq k}} = -\frac{\exp(\boldsymbol{\beta}_k^T \mathbf{x}) \cdot \exp(\boldsymbol{\beta}_\ell^T \mathbf{x})}{(\sum_j \exp(\boldsymbol{\beta}_j^T \mathbf{x}))^2} \mathbf{x} \tag{11}$$

$$\frac{\partial\, l(\boldsymbol{\beta}, \mathbf{x})}{\partial\, \boldsymbol{\beta}_{k=y}} = \frac{\sum_{j \neq k} \exp(\boldsymbol{\beta}_j^T \mathbf{x})}{\sum_j \exp(\boldsymbol{\beta}_j^T \mathbf{x})} \mathbf{x} \tag{12}$$

$$\frac{\partial\, l(\boldsymbol{\beta}, \mathbf{x})}{\partial\, \boldsymbol{\beta}_{\ell \neq y}} = -\frac{\exp(\boldsymbol{\beta}_\ell^T \mathbf{x})}{\sum_j \exp(\boldsymbol{\beta}_j^T \mathbf{x})} \mathbf{x}. \tag{13}$$

Note, that the factors on the rhs of (12) and (13) have absolute values of at most 1. Hence, we can bound

$$\begin{aligned} \|\nabla_{\boldsymbol{\beta}}\ d(\boldsymbol{\beta})\|_2 &= \frac{1}{N}\|\nabla_{\boldsymbol{\beta}}\ l(\boldsymbol{\beta}, \mathbf{x}'_N) - \nabla_{\boldsymbol{\beta}}\ l(\boldsymbol{\beta}, \mathbf{x}_N)\|_2 \\ &\leq \frac{1}{N}\left(\|\nabla_{\boldsymbol{\beta}}\ l(\boldsymbol{\beta}, \mathbf{x}'_N)\|_2 + \|\nabla_{\boldsymbol{\beta}}\ l(\boldsymbol{\beta}, \mathbf{x}_N)\|_2\right) \\ &\leq \frac{1}{N}\left(\sqrt{C}\|\mathbf{x}'_N\|_2 + \sqrt{C}\|\mathbf{x}_N\|_2\right) \\ &\leq \frac{2\sqrt{C}}{N}, \end{aligned} \tag{14}$$

where the last inequality follows from assumption (3) that $\|\mathbf{x}\|_2 \leq 1$.

We observe that due to the convexity of $l(\boldsymbol{\beta}, \mathbf{x})$ in $\boldsymbol{\beta}$ and the 1-strong convexity of the L_2-regularization term in (2), $J(\boldsymbol{\beta}, D)$ is λ-strongly convex. Hence, we obtain by Shalev-Shwartz inequality [Shalev-Shwartz, 2007]

$$\left(\nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{1}}, D) - \nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{2}}, D)\right)^T \left(\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\right) \geq \lambda\|\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\|_2^2. \tag{15}$$

Moreover, by construction of $d(\boldsymbol{\beta})$,

$$J(\boldsymbol{\beta}^*_{\mathbf{2}}, D) + d(\boldsymbol{\beta}^*_{\mathbf{2}}) = J(\boldsymbol{\beta}^*_{\mathbf{2}}, D'). \tag{16}$$

By optimality of $\boldsymbol{\beta}^*_{\mathbf{1}}$ and $\boldsymbol{\beta}^*_{\mathbf{2}}$, it holds

$$\mathbf{0} = \nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{1}}, D) = \nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{2}}, D') = \nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{2}}, D) + \nabla_{\boldsymbol{\beta}}\ d(\boldsymbol{\beta}^*_{\mathbf{2}}).$$

Applying the Cauchy-Schwartz inequality finally leads to

$$\begin{aligned} & \|\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\|_2 \cdot \|\nabla_{\boldsymbol{\beta}}\ d(\boldsymbol{\beta}^*_{\mathbf{2}})\|_2 \geq \left(\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\right)^T \nabla_{\boldsymbol{\beta}}\ d(\boldsymbol{\beta}^*_{\mathbf{2}}) \\ = & \left(\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\right)^T \left(\nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{1}}, D) - \nabla_{\boldsymbol{\beta}}\ J(\boldsymbol{\beta}^*_{\mathbf{2}}, D)\right) \\ \geq & \lambda\|\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\|_2^2, \end{aligned} \tag{17}$$

which concludes the proof, since

$$\|\boldsymbol{\beta}^*_{\mathbf{1}} - \boldsymbol{\beta}^*_{\mathbf{2}}\|_2 \leq \frac{\|\nabla_{\boldsymbol{\beta}}\ d(\boldsymbol{\beta}^*_{\mathbf{2}})\|_2}{\lambda} \leq \frac{2\sqrt{C}}{\lambda N}. \tag{18}$$

□

We remark that in the binary case ($C = 2$) one regression head parameterized by $\boldsymbol{\beta} \in \mathbb{R}^{(p+1)}$ suffices, resulting in an L_2-sensitivity of at most $\frac{2}{\lambda|D|}$.

4.2.3 Private Mechanism

Using Theorem 1 and the Gaussian mechanism, we get our (ϵ, δ)-differentially private mechanism for sanitizing the multinomial classification models as follows:

$$\mathcal{M}_{priv}(D) = \mathcal{M}(D) + \mathcal{N}(0, I\sigma^2), \text{ where}$$
$$\sigma^2 = \frac{8Cln(1.25\delta^{-1})}{\epsilon^2\lambda^2|D|^2}$$

This leads to the overall training procedure for the classification models described in Algorithm 1.

Algorithm 1 Classification model training and privatization

for each client **do**
$\boldsymbol{\beta}^* \rightarrow \text{argmin}_{\boldsymbol{\beta}} J(\boldsymbol{\beta}, h, D)$
$\sigma^2 \rightarrow \frac{8C \ln(1.25\delta^{-1})}{\epsilon^2 \lambda^2 (|D|)^2}$
$\boldsymbol{\beta}^* \rightarrow \boldsymbol{\beta}^* + \mathcal{N}(0, I\sigma^2)$
end for

4.2.4 Algorithm

4.3 Total Differential Privacy

By the composability and post-processing properties of differentially private mechanisms [Dwork and Roth, 2014], the total privacy loss for an individual client's dataset in training of the server's model is equal to the sum of the loss of the scoring and classification models. The server model is (ϵ, δ)-differentially private, where

$$\epsilon = \epsilon_{scores} + \epsilon_{classes}$$
$$\delta = \delta_{scores} + \delta_{classes}$$

5 Experiments

We ran experiments on large-scale convolutional, ShuffleNet- [Zhang et al., 2018], MobileNet- [Sandler et al., 2018], and ResNet-style [He et al., 2016] networks, using CIFAR-10 as local client data and both STL-10 and CIFAR-100 as auxiliary data. Of the auxiliary data, 80% is used for distillation and 20% for unsupervised pretraining. The pretraining is done by contrastive representation learning using the Adam optimizer with a learning rate of 10^{-3}.

The number of clients is $n = 20$ and there is full participation in one round of communication. The training data is split among the clients using a Dirichlet distribution as in [Hsu et al., 2019] using the Dirichlet parameter α. With the lowest $\alpha = 0.01$, clients see almost entirely one class of images. With the highest $\alpha = 10.24$, each client sees a substantial number of images from every class.

Class	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
First	94.5%	75.3%	56.8%	15.1%
Second	5.2%	16.6%	22.3%	13.6%
Third	0.3%	5.6%	10.1%	12.0%

Table 1: Ranked percentage of data coming from the three largest classes for each level of data heterogeneity

We find the optimal weights of the class model logistic regressions using sci-kit learn's LogisticRegression with the L-BFGS [Liu and Nocedal, 1989] optimizer. For baselines, we chose Federated Ensemble Distillation (FedD) and Federated Averaging (FedAVG), which we pretrain (+P) in the same fashion as FedAUXfdp. For FedAUXfdp and FedD+P, the full server model is trained for 10 distillation epochs using the Adam optimizer with a learning rate of $5 \cdot 10^{-5}$ and a batch size of 128. For FedAVG+P, the average of the weights of the clients' logistic regressions is used as a classification head on top of the frozen feature extractor on the server.

For privacy, we chose ($\epsilon = 0.1, \delta = 10^{-5}$) for the scores and unless otherwise mentioned ($\epsilon = 0.5, \delta = 10^{-5}$) for the classes. We choose regularization parameter $\lambda = 0.01$ for both the certainty score and class models unless otherwise mentioned.

As shown in Table 2, FedAUXfdp significantly outperforms baselines in the most heterogeneous settings ($\alpha = 0.01, 0.04$). While the baselines undergo a steady reduction in accuracy as client data heterogeneity increases, FedAUXfdp is even improving. The reason being, as data heterogeneity increases fewer classes per client result in the addition of less noise, see Theorem 1.

Model	Method	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
ShuffleNet	FedAVG+P	46.0± 0.4	56.7± 6.6	67.5± 3.5	74.1 ± 1.4
	FedD+P	41.8 ± 4.4	54.7 ± 5.0	68.8 ± 2.1	72.3 ± 1.6
	FedAUXfdp	**75.2 ± 1.1**	**74.6 ± 1.1**	72.3 ± 0.6	71.7 ± 1.3
MobileNetV2	FedAVG+P	47.2± 2.6	54.2± 5.5	65.6± 0.9	72.0 ± 0.6
	FedD+P	43.7 ± 1.8	52.2 ± 4.6	67.0 ± 1.7	70.8 ± 0.2
	FedAUXfdp	**72.8 ± 0.4**	**72.0 ± 1.2**	70.8 ± 0.2	69.4 ± 0.8

Table 2: Server model inference accuracy of **FedAUXfdp as compared to FL baselines** with pretraining. All methods total differential privacy ($\epsilon = 0.6, \delta = 2e-05$).

Table 3 shows the impact on accuracy of different levels of privacy in FedAUXfdp. We use FedAUXfdp with no class differential privacy as baseline (FedAUX with logistic classification heads on the client models and no server weight update) to isolate the impact of the class model differential privacy. Privatizing FedAUXfdp at additional epsilon-delta values of $(0.5, 10^{-5})$ results in nearly no reduction in accuracy over FedAUXfdp with no class model privacy. Only at $\epsilon = 0.1$ we see a drop in accuracy. With equal regularization, the additional differential privacy impacts the models trained on the non-iid data distributions less than those trained on homogeneous data, again due to the C-term in the L_2-sensitivity from Theorem 1.

Model	Method	Class DP	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
ShuffleNet	FedAUXfdp	None	76.1 ± 0.3	75.6 ± 0.4	75.2 ± 0.5	75.4 ± 0.1
	FedAUXfdp	(1.0, 1e-05)	75.7 ± 0.7	75.1 ± 0.7	74.6 ± 0.5	74.9 ± 0.2
	FedAUXfdp	(0.5, 1e-05)	75.2 ± 1.1	74.6 ± 1.1	72.3 ± 0.6	71.7 ± 1.3
	FedAUXfdp	(0.1, 1e-05)	60.8 ± 2.4	59.4 ± 5.8	33.9 ± 5.4	34.6 ± 3.0
	FedAUXfdp	(0.01, 1e-05)	36.3 ± 5.1	39.8 ± 7.5	12.6 ± 5.1	11.7 ± 3.5
MobileNetV2	FedAUXfdp	None	73.0 ± 0.5	73.3 ± 0.6	73.2 ± 0.2	73.0 ± 0.1
	FedAUXfdp	(1.0, 1e-05)	73.0 ± 0.4	72.7 ± 1.0	72.7 ± 0.3	72.4 ± 0.0
	FedAUXfdp	(0.5, 1e-05)	72.8 ± 0.4	72.0 ± 1.2	70.8 ± 0.2	69.4 ± 0.8
	FedAUXfdp	(0.1, 1e-05)	66.4 ± 3.3	53.1 ± 12.9	38.9 ± 4.4	34.9 ± 3.3
	FedAUXfdp	(0.01, 1e-05)	44.4 ± 6.8	28.7 ± 5.1	16.6 ± 5.8	11.5 ± 0.8

Table 3: FedAUXfdp server model inference accuracy at **various levels of class differential privacy** (ϵ, δ). Scoring model privacy for all methods ($\epsilon = 0.1, \delta = 1e-05$).

The drop in accuracy of differential privacy can be partially compensated for by increasing the regularization parameter λ of the client models' logistic regressions, as shown in Figure 2. On ShuffleNet, increasing the regularization from $\lambda = 0.01$ to $\lambda = 10$ nearly eliminates the gap between the accuracy with and without ($\epsilon = 0.01, \delta = 10^{-5}$) class model differential privacy at all levels of data heterogeneity α. The additional regularization does, however, reduce the accuracy of the model without the class differential privacy, moreso the more homogeneous the client data.

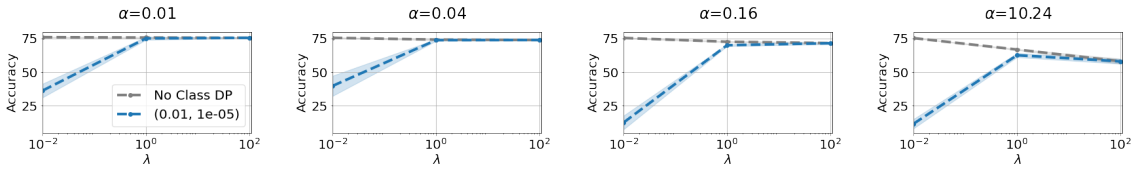

Figure 2: **Accuracy vs. regularization**. Server model inference accuracy of FedAUXfdp on ShuffleNet with and without ($\epsilon = 0.01, \delta = 1e-05$) class model differential privacy at various levels of class model regularization λ.

Table 4 shows results on ResNet with both STL-10 and CIFAR-100 as distillation data. STL-10 and CIFAR10 share 9/10 of the same classes, while CIFAR-100 has completely different classes. Even with distillation classes unmatching client classes, we still see robust results.

Distill Data	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
STL-10	77.2 ± 0.5	75.4 ± 1.0	74.7 ± 0.9	74.4 ± 0.8
CIFAR-100	70.4 ± 0.7	68.9 ± 1.8	67.6 ± 1.6	68.5 ± 1.9

Table 4: FedAUXfdp server model inference accuracy on ResNet8 with distillation data sharing 9/10 classes (STL-10) versus **entirely different distillation data classes** (CIFAR-100).

6 Conclusion

In this work, we have extended the FedAUX method, an augmentation of federated distillation, to be fully differentially private. We have contributed a mechanism that privatizes respectably with little loss in model accuracy, particularly on non-iid client data. We additionally contributed a theorem for the sensitivity of L_2 regularized multinomial logistic regression. On large scale image datasets we have examined the impact of different amounts of differential privacy and regularization. Measuring the impact of federated averaging, distillation, and differential privacy on the attackability of the global server model would be an interesting investigation direction.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.

H. Chang, V. Shejwalkar, R. Shokri, and A. Houmansadr. Cronus: Robust and heterogeneous collaborative learning with black-box knowledge transfer. *arXiv preprint arXiv:1912.11279*, 2019.

K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *J. Mach. Learn. Res.*, 12:1069–1109, 2011.

H.-Y. Chen and W.-L. Chao. FedDistill: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*, 2020.

C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot. Label-only membership inference attacks. In *Proceedings of the 38th International Conference on Machine Learning, PMLR*, volume 139, pages 1964–1974, 2021.

C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

R. C. Geyer, T. Klein, and M. Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557v2*, 2018.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

T.-M. H. Hsu, H. Qi, and M. Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *arXiv preprint arXiv:2008.06180*, 2020.

P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, and M. Bennis. Advances and open problems in federated learning. In *Foundations and Trends in Machine Learning*, volume 14, pages 1–210, 2021.

S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008.

Q. Li, Z. Wen, and B. He. Federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*, 2019.

X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of FedAvg on non-iid data. In *Proceedings of 8th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2020.

Y. Li, W. Zhou, H. Wang, H. Mi, and T. M. Hospedales. Fedh2l: Federated learning with model and statistical heterogeneity. *arXiv preprint arXiv:2101.11296*, 2021.

T. Lin, L. Kong, S. U. Stich, and M. Jaggi. Ensemble distillation for robust model fusion in federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1-3):503–528, 1989.

B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282, 2017.

B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2018.

N. Papernot, M. Abadi, Úlfar Erlingsson, I. Goodfellow, and K. Talwar. Semi-supervized knowledge transfer for deep learning from private training data. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2017.

M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

F. Sattler, T. Korjakow, R. Rischke, and W. Samek. Fedaux: Leveraging unlabeled auxiliary data in federated learning. In *IEEE Transactions on Neural Networks and Learning Systems*, 2021a.

F. Sattler, A. Marban, R. Rischke, and W. Samek. Cfd: Communication-efficient federated distillation via soft-label quantization and delta coding. *IEEE Trans. Netw. Sci. Eng.*, 2021b.

F. Sattler, K.-R. Müller, and W. Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2021c. doi: 10.1109/TNNLS.2020.3015958.

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, Hebrew University, 2007.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18, 2017.

L. Sun and L. Lyu. Federated model distillation with noise-free differential privacy. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, 2021.

X. Zhang, X. Zhou, M. Lin, and J. Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6848–6856, 2018.