FEDAUX: Leveraging Unlabeled Auxiliary Data in Federated Learning

Felix Sattler [1] **Tim Korjakow** [1] **Roman Rischke** [1] **Wojciech Samek** [1]

Abstract

Federated Distillation (FD) is a popular novel algorithmic paradigm for Federated Learning, which achieves training performance competitive to prior parameter averaging based methods, while additionally allowing the clients to train different model architectures, by distilling the client predictions on an *unlabeled auxiliary set of data* into a student model. In this work we propose FEDAUX, an extension to FD, which, under the same set of assumptions, drastically improves performance by deriving maximum utility from the unlabeled auxiliary data. FEDAUX modifies the FD training procedure in two ways: First, unsupervised pre-training on the auxiliary data is performed to find a model initialization for the distributed training. Second, (ε, δ)-differentially private certainty scoring is used to weight the ensemble predictions on the auxiliary data according to the certainty of each client model. Experiments on large-scale convolutional neural networks and transformer models demonstrate, that the training performance of FEDAUX exceeds SOTA FL baseline methods by a substantial margin in both the iid and non-iid regime, further closing the gap to centralized training performance. Code is available at `github.com/fedl-repo/fedaux`.

1. Introduction

Federated Learning (FL) allows distributed entities ("clients") to jointly train (deep) machine learning models on their combined data, without having to transfer this data to a centralized location (McMahan et al., 2017). The Federated training process is orchestrated by a central server. The distributed nature of FL improves privacy (Li et al., 2019), ownership rights (Sheller et al., 2020) and security (Mothukuri et al., 2021) for the participants. As the number of mobile and IoT devices and their capacities to collect large amounts of high-quality and privacy-sensitive data steadily grows, Federated training procedures become increasingly relevant.

While the client data in Federated Learning is typically assumed to be private, in most real-world applications the server additionally has access to unlabeled *auxiliary* data, which roughly matches the distribution of the client data. For instance, for many Federated computer vision and natural language processing problems, such auxiliary data can be given in the form of public data bases such as ImageNet (Deng et al., 2009) or WikiText (Merity et al., 2016). These data bases contain millions to billions of data samples but are typically lacking the necessary label information to be useful for training task-specific models.

Recently, Federated Distillation (FD), a novel algorithmic paradigm for Federated Learning problems where such auxiliary data is available, was proposed. In contrast to classic parameter averaging based FL algorithms (McMahan et al., 2017; Mohri et al., 2019; Reddi et al., 2020; Li et al., 2020a; Sattler et al., 2020c), which require all client's models to have the same size and structure, FD allows the clients to train heterogeneous model architectures, by distilling the client predictions on the auxiliary set of data into a student model. This can be particularly beneficial in situations where clients are running on heterogeneous hardware. Studies show that FD based training has favorable communication properties (Itahara et al., 2020; Sattler et al., 2020a), and can outperform parameter averaging based algorithms (Lin et al., 2020).

However, just like for their parameter-averaging-based counterparts, the performance of FD based learning algorithms falls short of centralized training and deteriorates quickly if the training data is distributed in a heterogeneous ("non-iid") way among the clients. In this work we aim to further close this performance gap, by exploring the core assumption of FD based training and deriving maximum utility from the available unlabeled auxiliary data. Our main contributions are as follows:

- We show that a wide range of (out-of-distribution)

[1]Department of Artificial Intelligence, Fraunhofer HHI, Berlin, Germany. Correspondence to: Felix Sattler <felix.sattler@hhi.fraunhofer.de>, Wojciech Samek <wojciech.samek@hhi.fraunhofer.de>.

This work was supported by the German Federal Ministry of Education and Research (BMBF) through the BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037I).

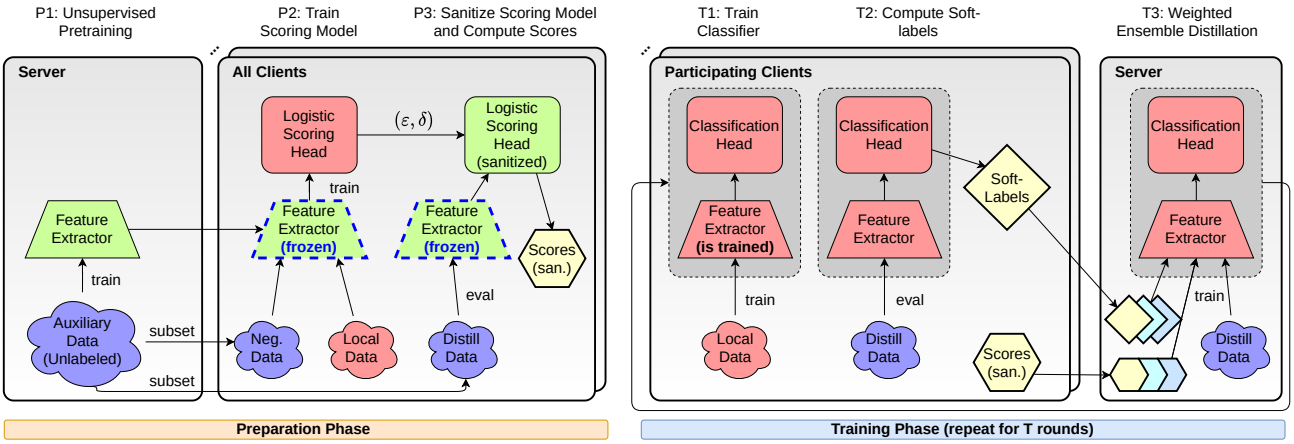


Figure 1. Training procedure of FEDAUX. **Preparation phase:** P1) The unlabeled auxiliary data is used to pre-train a feature extractor (e.g. using contrastive representation learning). P2) The feature-extractor is sent to the clients, where it is used to initialize the client models. Based on extracted features, a logistic scoring head is trained to distinguish local client data from a subset of the auxiliary data. P3) The trained scoring head is sanitized using a (ε, δ)-differentially private mechanism and then used to compute certainty scores on the distillation data. **Training Phase:** T1) In each communication round, a subset of the client population is selected for training. Each selected client downloads a model initialization from the server, and then updates the full model f_i (feature extractor & scoring head) using their private local data. T2) The locally trained classifier and scoring models f_i and s_i are sent to the server, where they are combined into a weighted ensemble. T3) Using the unlabeled auxiliary data and the weighted ensemble as a teacher, the server distills a student model which is used as the initialization point for the next round of Federated training. *Note that in practice we perform computation of soft-labels and scores at the server to save client resources.

auxiliary data sets are suitable for self-supervised pre-training and can drastically improve FL performance across all baselines.

- We propose a novel certainty-weighted FD technique, that improves performance of FD on non-iid data substantially, addressing a long-standing problem in FL research.
- We propose an (ε, δ)-differentially private mechanism to constrain the privacy loss associated with transmitting certainty scores.

These performance improvements are possible a) under the same assumptions made in the FD literature, b) with only negligible additional computational overhead for the resource-constrained clients and c) with small quantifiable excess privacy loss.

2. Related Work

Federated Distillation: Distillation (Bucila et al., 2006; Hinton et al., 2015) is a common technique to transfer the knowledge of one or multiple (You et al., 2017; Anil et al., 2018) machine learning classifiers to a different model, and is typically used in centralized settings before deployment in order to reduce the model complexity, while preserving predictive power. To this end, the predictions of the teacher model(s) on a distillation data set are used to guide the training process of the potentially less complex student model. Federated Distillation (FD) algorithms, which leverage these distillation techniques to aggregate the client knowledge, are recently gaining popularity, because they outperform conventional parameter averaging based FL methods (Lin et al., 2020; Chen & Chao, 2020) like FEDAVG or FedPROX (McMahan et al., 2017; Li et al., 2020a) and allow clients to train heterogeneous model architectures (Li & Wang, 2019; Chang et al., 2019; Li et al., 2021). FD methods can furthermore reduce communication overhead (Jeong et al., 2018; Itahara et al., 2020; Seo et al., 2020; Sattler et al., 2020a), by exploiting the fact that distillation requires only the communication of model predictions instead of full models. In contrast to centralized distillation, where training and distillation data usually coincide, FD makes no restrictions on the auxiliary distillation data[1], making it widely applicable. Our work, is in line with (Lin et al., 2020; Chen & Chao, 2020) in that it aims to improve overall training performance in FL. Both FEDDF (Lin et al., 2020) and FEDBE (Chen & Chao, 2020) combine parameter averaging as done in FedAVG (McMahan et al., 2017) with ensemble distillation to improve FL performance. While FEDDF combines client predictions by means of an (equally weighted) model ensemble, FEDBE forms a Bayesian ensemble from the client models for better robustness to heterogeneous data. Taking FEDDF as a starting point, we additionally leverage the auxiliary distillation data set for unsupervised pre-training

[1]Recent work even suggests that useful distillation data can be generated from the teacher models themselves (Nayak et al., 2019).

and weight the client predictions in the distillation step according to their prediction certainty to better cope with settings where the client's data generating distributions are statistically heterogeneous.

Weighted Ensembles: Weighted ensemble methods were studied already in classical work (Hashem & Schmeiser, 1993; Perrone & Cooper, 1993; Opitz & Maclin, 1999), with certainty weighted ensembles of neural networks in particular being proposed for classification e.g. in (Jiménez, 1998). Mixture of experts and boosting methods (Yuksel et al., 2012; Masoudnia & Ebrahimpour, 2014; Schapire, 1999) where multiple simple classifiers are combined by weighted averaging are frequently used in centralized settings.

A more detailed discussion of related work can be found in Appendix A.

3. Federated Learning with Auxiliary Data

In this section, we describe our method for efficient Federated Learning in the presence of unlabeled auxiliary data (FEDAUX). An illustration of our proposed approach is given in Figure 1. We describe FEDAUX for the homogeneous setting were all clients hold the same model prototype. The detailed algorithm for the more general model-heterogeneous setting can be found in Appendix C. An exhaustive *qualitative* comparison between FEDAUX and baseline methods is given in Appendix D.

3.1. Problem Setting

We assume the conventional FL setting where a population of n clients is holding potentially non-iid subsets of private labeled data $D_1, .., D_n$, from a training data distribution $(\bigcup_{i \leq n} D_i) \sim \varphi(\mathcal{X}, \mathcal{Y})$. We further make the assumption that the server and the clients both have access to a public collection of unlabeled auxiliary data from a deviating distribution $D_{aux} \sim \psi(\mathcal{X})$. The latter assumption is common to all studies on FD.

One round of federated training is then performed as follows: A subset $\mathcal{S}_t$ of the client population is selected by the server and downloads a model initialization. Starting from this model initialization, each client then proceeds to train a model f_i on it's local private data D_i by taking multiple steps of stochastic gradient descent. We assume that these local models can be decomposed into a feature extractor h_i and a classification head g_i according to $f_i = g_i \circ h_i$. Finally, the updated models f_i, $i \in \mathcal{S}_t$ are sent back to the server, where they are aggregated to form a new server model f, which is used as the initialization point for the next round of FL. The goal of FL is to obtain a server model f, which optimally generalizes to new samples from the training data distribution φ, within a minimum number of communication rounds $t \leq T$.

3.2. Federated Ensemble Distillation

Federated Ensemble Distillation is a novel method for aggregating the knowledge of FL clients. Instead of aggregating the parameters of the client models (e.g. via an averaging operation), a student model is trained on the combined predictions of the clients on some public auxiliary data. Let $x \in D_{aux}$ be a batch of data from the auxiliary distillation data set. Then one iteration of student distillation is performed as

$$\theta^{t,j+1} \leftarrow \theta^{t,j} - \eta \frac{\partial D_{KL}(\mathcal{A}(\{f_i(x)|i \in \mathcal{S}_t\}), \sigma(f(x, \theta^{t,j})))}{\partial \theta^{t,j}}. \tag{1}$$

Hereby, D_{KL} denotes the Kullback-Leibler divergence, $\eta > 0$ is the learning rate, σ is the softmax-function and $\mathcal{A}$ is a mechanism to aggregate the soft-labels. Existing work (Lin et al., 2020) aggregates the client predictions by taking the mean according to

$$\mathcal{A}_{mean}(\{f_i(x)|i \in \mathcal{S}_t\}) = \sigma\left(\frac{\sum_{i \in \mathcal{S}_t} f_i(x)}{|\mathcal{S}_t|}\right). \tag{2}$$

Federated Ensemble Distillation is shown to outperform parameter averaging based techniques (Lin et al., 2020).

3.3. Self-supervised Pre-training

Self-supervised representation learning can leverage large records of unlabeled data to create models which extract meaningful features. For the two types of data considered in this study - image and sequence data - strong self-supervised training algorithms are known in the form of contrastive representation learning (Chen et al., 2020; Wang & Isola, 2020) and next-token prediction (Devlin et al., 2019; Radford et al., 2019). As part of the FEDAUX preparation phase (cf. Fig. 1, P1) we propose to perform self-supervised training on the auxiliary data D_{aux} at the server. We emphasize that this step makes no assumptions on the similarity between the local training data and the auxiliary data. This results in a parametrization for the feature extractor h_0. Since the training is performed at the server, using publicly available data, this step inflicts neither computational overhead nor privacy loss on the resource-constrained clients.

3.4. Weighted Ensemble Distillation

Different studies have shown that both the training speed, stability and maximum achievable accuracy in existing FL algorithms deteriorate if the training data is distributed in a heterogeneous "non-iid" way among the clients (Zhao et al., 2018; Sattler et al., 2020c; Li et al., 2020b). Federated Ensemble Distillation makes no exception to this rule (Lin et al., 2020).

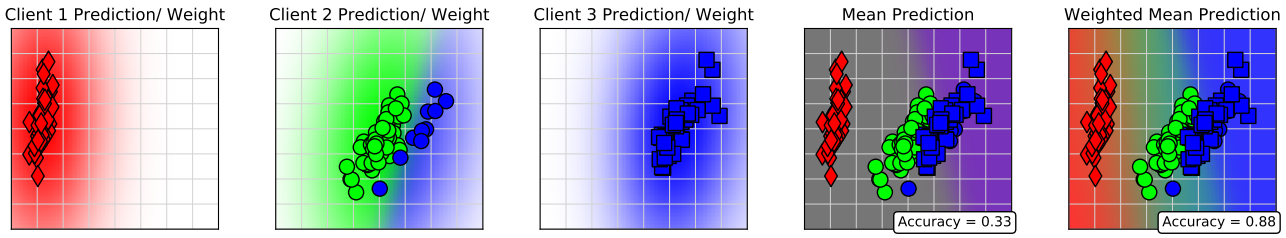


Figure 2. **Weighted Ensemble Distillation** illustrated in a toy example on the Iris data set (data points are projected to their two principal components). Three Federated Learning clients hold disjoint non-iid subsets of the training data. Panels 1-3: Predictions made by linear classifiers trained on the data of each client. Labels and predictions are color-coded, client certainty (measured via Gaussian KDE) is visualized via the alpha-channel. The mean of client predictions (panel 4) only poorly captures the distribution of training data. In contrast, the certainty-weighted mean of client predictions (panel 5) achieves much higher accuracy.

The underlying problem of combining hypotheses derived from different source domains has been explored in multiple-source domain adaptation theory (Mansour et al., 2008; Hoffman et al., 2018), which shows that standard convex combinations of the hypotheses of the clients as done in (Lin et al., 2020) may perform poorly on the target domain. Instead, a distribution-weighted combination of the local hypotheses is shown to be robust (Mansour et al., 2008; Hoffman et al., 2018). A simple toy example, displayed in Figure 2, further illustrates this point.

Inspired by these results, we propose to modify the aggregation rule of FD (2) to a certainty-weighted average:

$$\mathcal{A}_s(\{(f_i(x), s_i(x))|i \in \mathcal{S}_t\}) = \sigma\left(\frac{\sum_{i\in\mathcal{S}_t} s_i(x) f_i(x)}{\sum_{i\in\mathcal{S}_t} s_i(x)}\right) \tag{3}$$

The question remains, how to calculate the certainty scores $s_i(x)$ in a privacy preserving way and for arbitrary high-dimensional data, where simple methods, such as Gaussian KDE used in our toy example, fall victim to the curse of dimensionality. To this end, we propose the following methodology:

We split the available auxiliary data randomly into two disjoint subsets, $D^- \cup D_{distill} = D_{aux}$, the "negative" data and the "distillation" data. Using the pre-trained model h_0 ($\rightarrow$ sec. 3.3) as a feature extractor, on each client, we then train a logistic regression classifier to separate the local data D_i from the negatives D^-, by optimizing the following regularized empirical risk minimization problem

$$w_i^* = \arg\min_w J(w, h_0, D_i, D^-) \tag{4}$$

with

$$J(w, h_0, D_i, D^-) = a \sum_{x\in D_i\cup D^-} l(t_x \langle w, \tilde{h}_0(x)\rangle) + \lambda R(w). \tag{5}$$

Hereby $t_x = 2(\mathbb{1}_{x\in D_i}) - 1 \in [-1, 1]$ defines the binary labels of the separation task, $a = (|D_i| + |D^-|)^{-1}$ is a normalizing factor and $\tilde{h}_0(x) = h_0(x)(\max_{x\in D_i\cup D^-} \|h_0(x)\|)^{-1}$ are the normalized features. We choose $l(z) = \log(1 + \exp(z))$ to be the logistic loss and $R(w) = \frac{1}{2}\|w\|_2^2$ to be the ℓ_2-regularizer. Since J is λ-strongly convex in w, problem (4) is uniquely solvable. This step is performed only once on every client, during the preparation phase (cf. Fig. 1, P2) and the computational overhead for the clients of solving (4) is negligible in comparison to the cost of multiple rounds of training the (deep) model f_i.

Given the solution of the regularized ERM w_i^*, the certainty scores on the distillation data $D_{distill}$ can be obtained via

$$s_i(x) = (1 + \exp(-\langle w_i^*, \tilde{h}_0(x)\rangle))^{-1} + \xi. \tag{6}$$

A small additive $\xi > 0$ ensures numerical stability when taking the weighted mean in (3) (we set $\xi = 1e-8$). In Appendix I, we provide further empirical results, suggesting that our certainty-weighted averaging method (3) approximates a robust aggregation rule proposed in (Mansour et al., 2008).

3.5. Privacy Analysis

Sharing the certainty scores $\{s_i(x)|x \in D_{distill}\}$ with the central server intuitively causes privacy loss for the clients. After all, a high score $s_i(x)$ indicates, that the public data point $x \in D_{distill}$ is similar to the private data D_i of client i (in the sense of (4)). To protect the privacy of the clients, quantify and limit the privacy loss, we propose to use data-level differential privacy (cf. Fig. 1, P3). Following the classic definition of (Dwork & Roth, 2014), a randomized mechanism is called differentially private, if it's output on any input data base d is indistinguishable from output on any neighboring database d' which differs from d in one element.

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

$$P[\mathcal{M}(d) \in S] \leq \exp(\varepsilon) P[\mathcal{M}(d') \in S] + \delta. \tag{7}$$

Differential privacy of a mechanism $\mathcal{M}$ can be achieved, by limiting it's sensitivity

$$\Delta(\mathcal{M}) = \max_{d_1, d_2 \in \mathcal{D}} \|\mathcal{M}(d_1) - \mathcal{M}(d_2)\| \tag{8}$$

and then applying a randomized noise mechanism. We adapt a Theorem from (Chaudhuri et al., 2011) to establish the sensitivity of (4):

Theorem 1. *If $R(\cdot)$ is differentiable and 1-strongly convex and l is differentiable with $|l'(z)| \leq 1\ \forall z$, then the ℓ^2-sensitivity $\Delta_2(\mathcal{M})$ of the mechanism*

$$\mathcal{M} : D_i \mapsto \arg\min_w J(f, h_0, D_i, D^-) \tag{9}$$

is at most $2(\lambda(|D_i| + |D^-|))^{-1}$.

The proof can be found in Appendix J. As we can see the sensitivity scales inversely with the size of the total data $|D_i| + |D^-|$. From Theorem 1 and application of the Gaussian mechanism (Dwork & Roth, 2014) it follows that the randomized mechanism

$$\mathcal{M}_{san} : D_i \mapsto \arg\min_f J(f, h_0, D_i, D^-) + N \tag{10}$$

with $N \sim \mathcal{N}(\mathbf{0}, I\sigma^2)$ and $\sigma^2 = \frac{8\ln(1.25\delta^{-1})}{\varepsilon^2\lambda^2(|D_i|+|D_{aux}|)^2}$ is (ε, δ)-differentially private.

The post-processing property of DP ensures that the release of any number of scores computed using the output of mechanism $\mathcal{M}_{san}$ is still (ε, δ)-private. Note, that in this work we restrict ourselves to the privacy analysis of the scoring mechanism. The differentially private training of deep classifiers f_i is a challenge in it's own right and has been addressed e.g. in (Abadi et al., 2016). Following the basic composition theorem (Dwork & Roth, 2014), the total privacy cost of running FEDAUX is the sum of the privacy loss of the scoring mechanism $\mathcal{M}_{san}$ and the privacy loss of communicating the updated models f_i (the latter is the same for all FL algorithms).

4. Experiments

4.1. Setup

Datasets and Models: We evaluate FEDAUX and SOTA FL methods on both Federated image and text classification problems with large scale convolutional and transformer models respectively. For our image classification problems we train ResNet- (He et al., 2016), MobileNet- (Sandler et al., 2018) and ShuffleNet- (Zhang et al., 2018) type models on CIFAR-10 and CIFAR-100 and use STL-10, CIFAR-100 and SVHN as well as different subsets of ImageNet (Mammals, Birds, Dogs, Devices, Invertebrates, Structures)[2] as auxiliary data. In our experiments, we always use 80% of the auxiliary data as distillation data $D_{distill}$ and 20% as negative data D^-. For our text classification problems we train Tiny-Bert (Jiao et al., 2020) on the AG-NEWS (Zhang et al., 2015) and Multilingual Amazon Reviews Corpus (Keung et al., 2020) and use BookCorpus (Zhu et al., 2015) as auxiliary data.

Federated Learning environment and Data Partitioning: We consider Federated Learning problems with up to $n = 100$ participating clients. In all experiments, we split the training data evenly among the clients according to a dirichlet distribution following the procedure outlined in (Hsu et al., 2019) and illustrated in Fig. 6. This allows us to smoothly adapt the level of non-iid-ness in the client data using the dirichlet parameter α. We experiment with values for α varying between 100.0 and 0.01. A value of $\alpha = 100.0$ results in an almost identical label distribution, while setting $\alpha = 0.01$ results in a split, where the vast majority of data on every client stems from one single class. See Appendix B for a more detailed description of our data splitting procedure. We vary the client participation rate C in every round between 20% and 100%.

Pre-training strategy: For our image classification problems, we use contrastive representation learning as described in (Chen et al., 2020) for pre-training. We use the default set of data augmentations proposed in the paper and train with the Adam optimizer, learning rate set to 10^{-3} and a batch-size of 512. For our text classification problems, we pre-train using self-supervised next-word prediction.

Training the Scoring model and Privacy Setting: We set the default privacy parameters to $\lambda = 0.1$, $\varepsilon = 0.1$ and $\delta = 1e-5$ respectively and solve (4) by running L-BFGS (Liu & Nocedal, 1989) until convergence (≤ 1000 steps).

Baselines: We compare the performance of FEDAUX to state-of-the-art FL methods: FEDAVG (McMahan et al., 2017), FEDPROX (Li et al., 2020a), Federated Ensemble Distillation (FEDDF) (Lin et al., 2020) and FEDBE (Chen & Chao, 2020). To clearly discern the performance benefits of the two components of FEDAUX (unsupervised pre-training and weighted ensemble distillation), we also report performance metrics on versions of these methods where the auxiliary data was used to pre-train the feature extractor h ("FEDAVG+P", "FEDPROX+P", "FEDDF+P" resp. "FEDBE+P"). For FEDBE we set the sample size to 10 as suggested in the paper. For FEDPROX we always tune the proximal parameter μ.

Optimization: On all image classification task, we use the very popular Adam optimizer (Kingma & Ba, 2014), with a fixed learning rate of $\eta = 10^{-3}$ and a batch-size of 32 for local training. Distillation is performed for one epoch for all methods using Adam at a batch-size of 128 and fixed learning rate of $5e-5$. More detailed hyperparameter anal-

[2]The methodology for generating these subsets is described in Appendix F

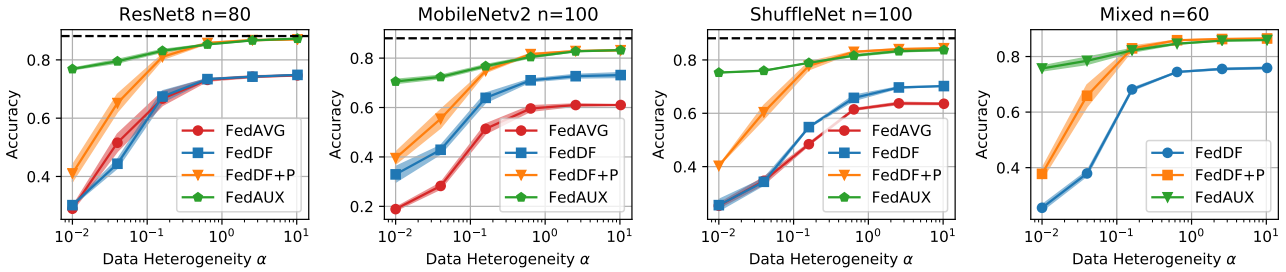


Figure 3. Evaluation on **different neural networks** and client population sizes n. Accuracy achieved after $T = 100$ communication rounds by different Federated Distillation methods at different levels of data heterogeneity α. STL-10 is used as auxiliary data set. In the "Mixed" setting one third of the client population each trains on ResNet8, MobileNetv2 and Shufflenet respectively. Black dashed line indicates centralized training performance.

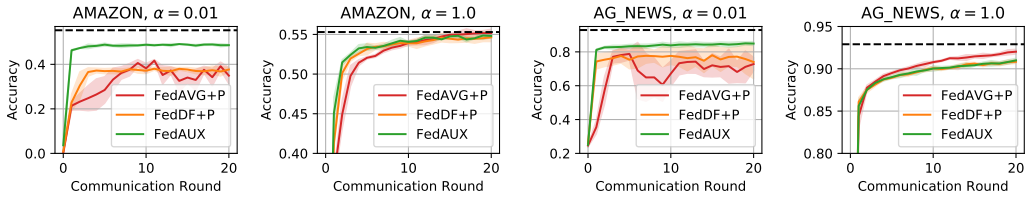


Figure 4. Evaluating FEDAUX on **NLP Benchmarks**. Performance of FEDAUX for different combinations of local datasets and heterogenity levels α. 10 clients training TinyBERT at $\alpha = 0.01$ and $C = 100\%$. Bookcorpus is used as auxiliary data set. Black dashed line indicates centralized training performance.

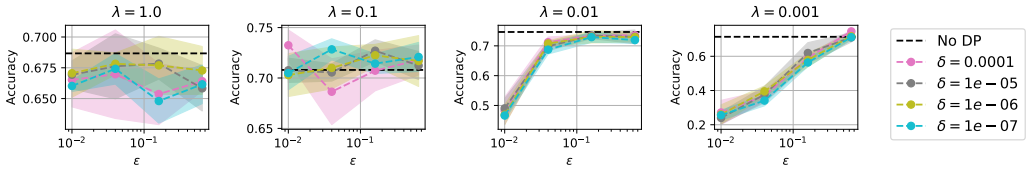


Figure 5. **Privacy Analysis**. Performance of FEDAUX for different combinations of the privacy parameters ε, δ and λ. 40 clients training Resnet-8 for $T = 10$ rounds on CIFAR-10 at $\alpha = 0.01$ and $C = 40\%$. STL-10 is used as auxiliary data set.

ysis in Appendix H shows that this choice of optimization parameters is approximately optimal for all of the methods. If not stated otherwise, the number of local epochs E is set to 1.

4.2. Evaluating FEDAUX on common Federated Learning Benchmarks

We start out by evaluating the performance of FEDAUX on classic benchmarks for Federated image classification. Figure 3 shows the maximum accuracy achieved by different Federated Distillation methods after $T = 100$ communication rounds at different levels of data heterogeneity. As we can see, FEDAUX distinctively outperforms FEDDF on the entire range of data heterogeneity levels α on all benchmarks. For instance, when training ResNet8 with $n = 80$ clients at $\alpha = 0.01$, FEDAUX *raises the maximum achieved accuracy from 18.2% to 78.1%* (under the same set of assumptions). The two components of FEDAUX, unsupervised pre-training and weighted ensemble distillation, both contribute independently to the performance improvement, as can be seen when comparing with FEDDF+P, which only uses unsupervised pre-training. Weighted ensemble distillation as done in FEDAUX leads to greater or equal performance than equally weighted distillation (FEDDF+P) across all levels of data heterogeneity. The same overall picture can be observed in the "Mixed" setting where clients train different model architectures. Detailed training curves are given in the Appendix E.

Table 1 compares the performance of FEDAUX and baseline methods at different client participation rates C. We can see that FEDAUX benefits from higher participation rates. In all scenarios, methods which are initialized using the pre-trained feature-extractor h_0 distinctively outperform their randomly initialized counterparts. In the iid setting at $\alpha = 100.0$ FEDAUX is mostly en par with the (improved) parameter averaging based methods FEDAVG+P and FEDPROX+P, with a maximum performance gap of 0.8%. At $\alpha = 0.01$ on the other hand FEDAUX outperforms all other methods with a margin of up to 29%.

4.3. Evaluating FEDAUX on NLP Benchmarks

Figure 4 shows learning curves for Federated training of TinyBERT on the Amazon and AG-News datasets at two different levels of data heterogeneity α. We observe, that

Table 1. Maximum accuracy achieved by FEDAUX and other baseline FL methods after $T = 100$ communication rounds, at **different participation rates** C and levels of data heterogeneity α. 20 Clients training ResNet-8 on CIFAR-10. Auxiliary data used is STL10. *Methods assume availability of auxiliary data. †Improved Baselines.

	$\alpha = 0.01$			$\alpha = 100.0$		
Method	$C = 0.2$	$C = 0.4$	$C = 0.8$	$C = 0.2$	$C = 0.4$	$C = 0.8$
FEDAVG (McMahan et al., 2017)	19.9±0.7	23.6±2.0	28.9±2.0	81.3±0.1	82.2±0.0	82.3±0.1
FEDPROX (Li et al., 2020a)	28.4±2.5	34.0±1.9	42.0±1.0	81.4±0.1	82.3±0.2	82.0±0.3
FEDDF* (Lin et al., 2020)	25.0±0.8	27.8±0.8	30.6±0.3	80.8±0.1	81.4±0.3	81.5±0.3
FEDBE* (Chen & Chao, 2020)	20.9±0.6	25.7±1.4	29.1±0.1	81.4±0.7	82.0±0.1	82.2±0.2
FEDAVG+P*†	30.4±7.9	32.1±2.0	38.4±0.5	89.0±0.1	**89.5±0.1**	**89.6±0.1**
FEDPROX+P*†	42.8±2.7	43.1±0.2	49.0±0.7	88.9±0.0	89.1±0.1	89.4±0.0
FEDDF+P*†	28.8±3.0	39.3±3.6	48.1±1.1	88.8±0.0	88.9±0.1	88.9±0.1
FEDBE+P*†	30.2±2.2	29.8±0.8	37.7±0.0	**89.1±0.1**	89.5±0.2	89.5±0.0
FEDAUX*	**54.2±0.3**	**71.2±2.1**	**78.5±0.0**	88.9±0.0	89.0±0.0	89.0±0.1

Table 2. Maximum accuracy achieved by FEDAUX and other baseline FL methods after 100 communication rounds, when **different sets of unlabeled auxiliary data** are used for pre-training and/ or distillation. 40 Clients training ResNet-8 on CIFAR-10 at $C = 40\%$.

		Auxiliary Data							
α	Method	STL-10	CIFAR-100	SVHN	Invertebr.	Birds	Devices	Dogs	Structures
0.01	FEDDF	27.9±3.2	29.5±6.2	28.1±3.9	28.5±3.6	30.1±2.0	26.3±0.2	28.9±5.1	30.2±7.0
	FEDDF+P	43.0±5.2	41.6±1.1	29.6±3.4	38.8±6.5	41.4±5.9	35.9±4.9	41.1±7.3	36.7±7.1
	FEDAUX	**76.8±0.9**	**71.5±2.5**	**43.7±1.5**	**68.2±0.7**	**65.7±3.1**	**71.5±0.1**	**71.8±3.8**	**64.1±3.3**
100.00	FEDDF	79.3±0.7	79.9±0.1	80.9±0.1	80.2±0.1	80.2±0.4	79.4±0.3	79.7±0.4	80.1±0.2
	FEDDF+P	88.3±0.0	86.7±0.0	**81.7±0.2**	87.4±0.1	87.6±0.0	87.7±0.1	88.4±0.0	**87.4±0.1**
	FEDAUX	**88.5±0.0**	**86.7±0.1**	81.6±0.0	**87.8±0.1**	**87.8±0.1**	**87.8±0.0**	**88.6±0.0**	87.3±0.1

Table 3. **One-shot performance** of different FL methods. Maximum accuracy achieved after $T = 1$ communication rounds at participation-rate $C = 100\%$. Each client trains for $E = 40$ local epochs.

	MobileNetv2, $n = 100$				Shufflenet, $n = 100$			
Method	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$	$\alpha = 0.01$	$\alpha = 0.04$	$\alpha = 0.16$	$\alpha = 10.24$
FEDAVG	10.3±0.0	13.6±2.3	23.6±0.0	30.5±0.9	12.1±0.8	17.4±0.4	28.2±0.8	37.8±0.7
FEDPROX	11.6±0.8	14.3±1.4	23.7±0.3	30.5±0.5	12.9±1.7	18.9±0.2	29.4±0.3	38.9±0.5
FEDDF	16.8±4.2	29.5±3.8	37.7±1.1	40.4±0.5	16.0±5.1	27.3±0.1	38.7±0.2	45.5±0.5
FEDAVG+P	24.3±1.1	44.0±4.4	57.6±3.7	69.9±0.0	25.5±1.4	44.2±0.1	62.9±1.6	71.9±0.1
FEDPROX+P	27.2±2.2	43.4±3.6	56.9±3.9	70.0±0.1	28.4±0.2	47.1±1.5	63.3±1.2	71.9±0.1
FEDDF+P	46.7±5.6	61.1±1.3	67.6±0.5	71.2±0.1	40.4±2.7	59.4±0.8	68.8±0.2	72.7±0.0
FEDAUX	**64.8±0.0**	**65.5±1.0**	**68.2±0.2**	**71.3±0.1**	**66.9±0.6**	**68.6±0.4**	**70.8±0.3**	**72.9±0.1**

FEDAUX significantly outperforms FEDDF+P as well as FEDAVG+P in the heterogeneous setting ($\alpha = 0.01$) and reaches 95% of its final accuracy after one communication round on both datasets, indicating suitability for one-shot learning. On more homogeneous data ($\alpha = 1.0$) FEDAUX performs mostly en par with pre-trained versions of FEDAVG and FEDDF, with a maximal performance gap of 1.1 % accuracy on the test set. We note, that effects of data heterogeneity are less severe as in this setting as both the AG News and the Amazon data set only have four and five labels respectively and an α of 1.0 already leads to a distribution where each clients owns a subset of the private data set containing all possible labels. Further details on our implementation can be found the Appendix G.

4.4. Privacy Analysis of FEDAUX

Figure 5 examines the dependence of FEDAUX' training performance of the privacy parameters ε, δ and the regularization parameter λ. As we can see, performance comparable to non-private scoring is achievable at conservative privacy parameters ε, δ. For instance, at $\lambda = 0.01$ setting $\varepsilon = 0.04$ and $\delta = 10^{-6}$ reduces the accuracy from 74.6% to 70.8%. At higher values of λ, better privacy guarantees have an even less harmful effect, at the cost however of an

overall degradation in performance. Throughout this empirical study, we have set the default privacy parameters to $\lambda = 0.1$, $\varepsilon = 0.1$ and $\delta = 1e-5$. We also perform an empirical privacy analysis in the Appendix K, which provides additional intuitive understanding and confidence in the privacy properties of our method.

4.5. Evaluating the dependence on Auxiliary Data

Next, we investigate the influence of the auxiliary data set D_{aux} on unsupervised pretraining, distillation and weighted distillation respectively. We use CIFAR-10 as training data set and consider 8 different auxiliary data sets, which differ w.r.t their similarity to this client training data - from more similar (STL-10, CIFAR-100) to less similar (Devices, SVHN)[3]. Table 2 shows the maximum achieved accuracy after $T = 100$ rounds when each of these data sets is used as auxiliary data. As we can see, performance *always* improves when auxiliary data is used for unsupervised pre-training. Even for the highly dissimilar SVHN data set (which contains images of house numbers) performance of FEDDF+P improves by 1% over FEDDF in both the iid and non-iid regime. For other data sets like Dogs, Birds or Invertebrates performance improves by up to 14%, although they overlap with only one single class of the CIFAR-10 data set. The outperformance of FEDAUX on such a wide variety of highly dissimilar data sets suggest that beneficial auxiliary data should be available in the majority of practical FL problems and also has positive implications from the perspective of privacy. Interestingly, performance of FEDDF seems to only weakly correlate with the performance of FEDDF+P and FEDAUX as a function of the auxiliary data set. This suggests, that the properties, which make a data set useful for distillation are not the same ones that make it useful for pre-training and weighted distillation. Investigating this relationship further is an interesting direction of future research.

4.6. FEDAUX in hardware-constrained settings

Linear Evaluation: In settings where the FL clients are hardware-constrained mobile or IoT devices, local training of entire deep neural networks like ResNet8 might be infeasible. We therefore also consider the evaluation of different FL methods, when only the linear classification head g is updated during the training phase. Figure 7 shows training curves in this setting when clients hold data from the CIFAR-10 data set. We see that in this setting performance of FEDAUX is high, independent of the data heterogeneity levels α, suggesting that in the absence of non-convex training dynamics our proposed scoring method actually yields robust weighted ensembles in the sense of (Mansour et al., 2008). We note, that FEDAUX also trains much more smoothly, than all other baseline methods.

[3]The CIFAR-10 data set contains images from the classes airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truc.

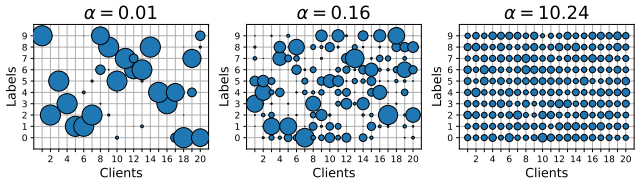


Figure 6. Illustration of the **Dirichlet data splitting strategy** we use throughout the paper, exemplary for a Federated Learning setting with 20 Clients and 10 different classes. Marker size indicates the number of samples held by one client for each particular class. Lower values of α lead to more heterogeneous distributions of client data. Figure adapted from (Lin et al., 2020).

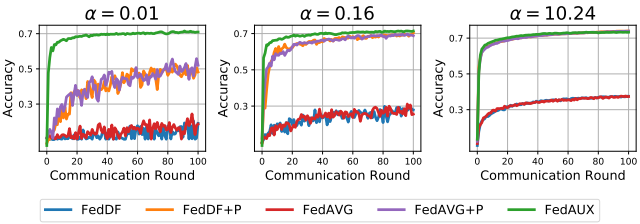


Figure 7. **Linear evaluation**. Training curves for different Federated Learning methods at different levels of data heterogeneity α when only the classification head g is updated in the training phase. A total of $n = 80$ clients training ResNet8 on CIFAR-10 at $C = 40\%$, using STL-10 as auxiliary data set.

One-Shot Evaluation: In many FL applications, the number of times a client can participate in the Federated training is restricted by communication, energy and/ or privacy constraints (Guha et al., 2019; Papernot et al., 2018). To study these types of settings, we investigate the performance of FEDAUX and other FL methods in Federated one-shot learning where we set $T = 1$ and $C = 100\%$. Table 3 compares performance in this setting for $n = 100$ clients training MobileNetv2 resp. ShuffleNet. FEDAUX outperforms the baseline methods in this setting at all levels of data heterogeneity α.

5. Conclusion

In this work, we explored Federated Learning in the presence of unlabeled auxiliary data, an assumption made in the quickly growing area of Federated Distillation. By leveraging auxiliary data for unsupervised pre-training and weighted ensemble distillation we were able to demonstrate that this assumption is rather strong and can lead to drastically improved performance of FL algorithms. These results reveal the limited merit in comparing FD based methods with parameter averaging based methods (which do not make this assumption) and thus have implications for the future evaluation of FD methods in general.

References

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

Ahn, J.-H., Simeone, O., and Kang, J. Wireless federated distillation for distributed edge learning with heterogeneous data. In *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6. IEEE, 2019.

Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G. E., and Hinton, G. E. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, 2010.

Bucila, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 535–541, 2006.

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

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

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

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 1597–1607, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL-HLT)*, volume 1, pp. 4171–4186, 2019.

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

Ghosh, A., Hong, J., Yin, D., and Ramchandran, K. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

Guha, N., Talwalkar, A., and Smith, V. One-shot federated learning. *arXiv preprint arXiv:1902.11175*, 2019.

Hashem, S. and Schmeiser, B. Approximating a function and its derivatives using mse-optimal linear combinations of trained feedforward neural networks. In *Proceedings of the World Congress on Neural Networks*, volume 1, pp. 617–620, 1993.

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

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hoffman, J., Mohri, M., and Zhang, N. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 8256–8266, 2018.

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

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

Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., and Kim, S. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

Jeong, W., Yoon, J., Yang, E., and Hwang, S. J. Federated semi-supervised learning with inter-client consistency. *arXiv preprint arXiv:2006.12097*, 2020.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. TinyBERT: Distilling BERT for natural language understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings (EMNLP)*, pp. 4163–4174, 2020.

Jiménez, D. Dynamically weighted ensemble neural networks for classification. In *IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence*, volume 1, pp. 753–756, 1998.

Keung, P., Lu, Y., Szarvas, G., and Smith, N. A. The multilingual amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4563–4568, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Li, D. and Wang, J. FedMD: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*, 2019.

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

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020a.

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

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

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

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

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 21, pp. 1041–1048, 2008.

Mansour, Y., Mohri, M., Ro, J., and Suresh, A. T. Three approaches for personalization with applications to federated learning. *arXiv preprint arXiv:2002.10619*, 2020.

Masoudnia, S. and Ebrahimpour, R. Mixture of experts: A literature survey. *Artif. Intell. Rev.*, 42(2):275–293, 2014.

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

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 4615–4625, 2019.

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A., and Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.*, 115:619–640, 2021.

Nayak, G. K., Mopuri, K. R., Shaj, V., Radhakrishnan, V. B., and Chakraborty, A. Zero-shot knowledge distillation in deep networks. In *Proceedings of the 36th International Conference on Machine Learning, (ICML)*, pp. 4743–4751, 2019.

Opitz, D. W. and Maclin, R. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.*, 11:169–198, 1999.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with PATE. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018.

Perrone, M. P. and Cooper, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In Mammone, R. J. (ed.), *Neural Networks for Speech and Image Processing*. Chapman and Hall, 1993.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

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

Sattler, F., Marban, A., Rischke, R., and Samek, W. Communication-efficient federated distillation. *arXiv preprint arXiv:2012.00632*, 2020a.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–13, 2020b.

Sattler, F., Wiedemann, S., Müller, K.-R., and Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.*, 31(9):3400–3413, 2020c.

Schapire, R. E. A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1401–1406, 1999.

Seo, H., Park, J., Oh, S., Bennis, M., and Kim, S. Federated knowledge distillation. *arXiv preprint arXiv:2011.02367*, 2020.

Sharkey, A. J. C. On combining artificial neural nets. *Connect. Sci.*, 8(3):299–314, 1996.

Sheller, M. J., Edwards, B., Reina, G. A., Martin, J., Pati, S., Kotrotsou, A., Milchenko, M., Xu, W., Marcus, D., Colen, R. R., et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):1–12, 2020.

Smith, V., Chiang, C., Sanjabi, M., and Talwalkar, A. S. Federated multi-task learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 4424–4434, 2017.

Sollich, P. and Krogh, A. Learning with ensembles: How overfitting can be useful. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 8, pp. 190–196, 1995.

Sun, L. and Lyu, L. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537*, 2020.

Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019.

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Wu, H., Chen, C., and Wang, L. A theoretical perspective on differentially private federated multi-task learning. *arXiv preprint arXiv:2011.07179*, 2020.

You, S., Xu, C., Xu, C., and Tao, D. Learning from multiple teacher networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1285–1294, 2017.

Yuksel, S. E., Wilson, J. N., and Gader, P. D. Twenty years of mixture of experts. *IEEE Trans. Neural Networks Learn. Syst.*, 23(8):1177–1193, 2012.

Zhang, F., Kuang, K., You, Z., Shen, T., Xiao, J., Zhang, Y., Wu, C., Zhuang, Y., and Li, X. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*, 2020a.

Zhang, X., Zhao, J. J., and LeCun, Y. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 649–657, 2015.

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

Zhang, Z., Yao, Z., Yang, Y., Yan, Y., Gonzalez, J. E., and Mahoney, M. W. Benchmarking semi-supervised federated learning. *arXiv preprint arXiv:2008.11364*, 2020b.

Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., and Chandra, V. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Zhou, Y., Pu, G., Ma, X., Li, X., and Wu, D. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020.

Zhu, Y., Kiros, R., Zemel, R. S., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 19–27, 2015.

FEDAUX: Leveraging Unlabeled Auxiliary Data in Federated Learning
- SUPPLEMENTARY MATERIALS -

A. Extended Related Work Discussion

Ensemble Distillation in Federated Learning:

A new family of Federated Learning methods leverages model distillation (Hinton et al., 2015) to aggregate the client knowledge (Jeong et al., 2018; Lin et al., 2020; Itahara et al., 2020; Chen & Chao, 2020). These Federated Distillation (FD) techniques have at least three distinct advantages over prior, parameter averaging based methods and related work can be organized according to which of these aspects it primarily focuses on.

First, Federated Distillation enables aggregation of client knowledge independent of the model architecture and thus allows clients to train models of different structure, which gives additional flexibility, especially in hardware-constrained settings. FEDMD (Li & Wang, 2019), Cronus (Chang et al., 2019) and FEDH2L (Li et al., 2021) address this aspect. FedMD additionally requires to locally pre-train on the *labeled* public data which makes it difficult to perform a fair numerical comparison. FedH2L requires communication of soft-label information after every gradient descent step and is thus not suitable for most practical FL applications where communication channels are intermittent. Cronus addresses aspects of robustness to adversaries but is shown to perform consistently worse than FEDAVG in conventional FL. While we do not focus on this aspect, our proposed approach is flexible enough to handle heterogeneous client models (c.f. Appendix C).

Second, Federated Distillation has advantageous communication properties. As models are aggregated by means of distillation instead of parameter averaging it is no longer necessary to communicate the raw parameters. Instead it is sufficient for the clients to only send their soft-label predictions on the distillation data. Consequently, the communication in FD scales with the size of the distillation data set and not with the size of the jointly trained model as in the classical parameter averaging based FL. This leads to communication savings, especially if the local models are large and the distillation data set is small. Jeong et. al and subsequent work (Jeong et al., 2018; Itahara et al., 2020; Seo et al., 2020; Sattler et al., 2020a) focus on this aspect. These methods however are computationally more expensive for the resource constrained clients, as distillation needs to be performed locally and perform worse than parameter averaging based training after the same number of communication rounds. Our proposed approach relies on communication of full models and thus requires communication at the order of conventional parameter averaging based methods.

Third, when combined with parameter averaging, Federated Distillation methods achieve better performance than purely parameter averaging based techniques. Both the authors in (Lin et al., 2020) and (Chen & Chao, 2020) propose FL protocols, which are based on classical FEDAVG and perform ensemble distillation after averaging the received client updates at the server to improve performance. FEDBE, proposed by (Chen & Chao, 2020), additionally combines client predictions by means of a Bayesian model ensemble to further improve robustness of the aggregation. Our work primarily focuses on this latter aspect. Building upon the work of (Lin et al., 2020), we additionally leverage the auxiliary distillation data for unsupervised pre-training and weigh the client predictions in the distillation step according to their certainty scores to better cope with settings where the client's data generating distributions are statistically heterogeneous.

We also mention the related work by Guha et al. (Guha et al., 2019), which proposes a one-shot distillation method for convex models, where the server distills the locally optimized client models in a single round as well as the work of (Sun & Lyu, 2020) which addresses privacy issues in Federated Distillation. Federated one-shot distillation is also addressed in (Zhou et al., 2020). Federated Distillation for edge-learning was proposed in (Ahn et al., 2019).

Weighted Ensembles: The study of weighted ensembles started around the '90s with the work by (Hashem & Schmeiser, 1993; Perrone & Cooper, 1993; Sollich & Krogh, 1995). A weighted ensemble of models combines the output of the individual models by means of a weighted average in order to improve the overall generalization performance. The weights allow to indicate the percentage of trust or expected performance for each individual model. See (Sharkey, 1996; Opitz & Maclin, 1999) for an overview of ensemble methods. Instead of giving each client a static weight in the aggregation step of distillation, we weight the clients on an instance base as in (Jiménez, 1998), i.e., each clients prediction is weighted using a data-dependent

certainty score. Weighted combinations of weak classifiers are also commonly leveraged in centralized settings in the context of of mixture of experts and boosting methods (Yuksel et al., 2012; Masoudnia & Ebrahimpour, 2014; Schapire, 1999).

Data Heterogeneity in Federated Learning: As the training data is generated independently on the participation devices, Federated Learning problems are typically characterised by statistically heterogeneous client data (McMahan et al., 2017). It is well known, that conventional FL algorithms like FEDAVG (McMahan et al., 2017) perform best on statistically homogeneous data and suffer severely in this ("non-iid") setting (Zhao et al., 2018; Li et al., 2020b). A number of different studies (Li et al., 2020a; Zhao et al., 2018; Sattler et al., 2020c; Chen & Chao, 2020) have tried to address this issue, but relevant performance improvements so far have only been possible under strong assumptions. For instance (Zhao et al., 2018) assume that the server has access to *labeled* public data from the *same* distribution as the clients. In contrast, we only assume that the server has access to *unlabeled* public data from a potentially *deviating* distribution. Other approaches (Sattler et al., 2020c) require high-frequent communication, with up to thousands of communication rounds, between server and clients, which might be prohibitive in a majority of FL applications where communication channels are intermittent and slow. In contrast, our proposed approach can drastically improve FL performance on non-iid data even after just one single communication round. For completeness, we note that there exists also a different line of research, which aims to address data heterogeneity in FL via meta- and multi-task learning. Here, separate models are trained for each client (Smith et al., 2017; Wu et al., 2020) or clients are grouped into different clusters with similar distributions (Ghosh et al., 2019; Sattler et al., 2020b).

Unlabeled Data in Federated Learning: To the best of our knowledge, there do not exist any prior studies on the use of unlabeled auxiliary data in FL outside of Federated Distillation methods. Federated semi-supervised learning techniques (Zhang et al., 2020b; Jeong et al., 2020) assume that clients hold both labeled and unlabeled private data from the local training distribution. In contrast, we assume that the server has access to public unlabeled data that may differ in distribution from the local client data. Federated self-supervised representation learning (Zhang et al., 2020a) aims to train a feature extractor on private unlabeled client data. In contrast, we leverage self-supervised representation learning at the server to find a suitable model initialization.

Personalization and Federated Transfer Learning: The aim of Transfer Learning is to transfer learned knowledge from a specific domain or task to related domains or tasks. Transfer learning methods are of particular interest in FL settings where the client's local data generating distributions are statistically heterogeneous. To address the statistical heterogeneity, methods for personalizing the server model to the client's local distributions, e.g. by using distillation (Li & Wang, 2019), parameter fine-tuning (Wang et al., 2019; Mansour et al., 2020) or regularization (Li et al., 2020a), have been proposed. Transferring knowledge from one domain to another domain raises the question of the generalization capabilities and domain adaptation theory gives answers in the form of generalization bounds. Particularly, multiple-source domain adaptation theory (Mansour et al., 2008; Ben-David et al., 2010; Hoffman et al., 2018), which considers the capabilities of transferring knowledge from multiple source domains to some target domain, is relevant for FL. One interesting question when having knowledge in multiple source domains is how to weight each individual source domain in the process of transferring knowledge to the target domain. In the FEDDF algorithm (Lin et al., 2020), the client's local hypotheses are uniformly averaged to obtain a global hypothesis and it is remarked that domain adaptation theory (Mansour et al., 2008; Hoffman et al., 2018) has shown such standard convex combinations of source hypotheses not to be robust for the target domain. A distribution-weighted combination of the local hypotheses, as suggested by domain adaptation theory (Mansour et al., 2008) (Hoffman et al., 2018), based on a privacy-preserving local distribution estimation is posed as an open problem for FL in (Lin et al., 2020). We address exactly this open question.

Algorithm 1 FEDAUX Preparation Phase (with different model prototypes $\mathcal{P}$)

init: Split $D^- \cup D_{distill} \leftarrow D_{aux}$
init: HashMap $\mathcal{R}$ that maps client i to model prototype P
Server does:
for each model prototype $P \in \mathcal{P}$ **do**
  $h_0^P \leftarrow$ train_self_supervised(h^P, D_{aux})
end for
for each client $i \in \{1, .., n\}$ **in parallel do**
  Client i does:
  $P \leftarrow \mathcal{R}[i]$
  $\sigma^2 \leftarrow \frac{8 \ln(1.25\delta^{-1})}{\varepsilon^2 \lambda^2 (|D_i|+|D^-|)^2}$
  $w_i^* \leftarrow \arg\min_w J(w, h_0^P, D_i, D^-) + \mathcal{N}(\mathbf{0}, I\sigma^2)$
  $\gamma_i \leftarrow \max_{x \in D_i \cup D^-} \|h_0^P(x)\|$
end for
Server does:
for $i = 1, .., n$ **do**
  create HashMap
  $s_i \leftarrow \{x \mapsto (1 + \exp(-\langle w_i^*, \gamma_i^{-1} h_0^P(x) \rangle))^{-1} + \xi$ for $x \in D_{distill}\}$
end for

B. Data Splitting Methodology

We split the training data among the clients using the common Dirichlet splitting strategy proposed in (Hsu et al., 2019) and later used in (Lin et al., 2020) and (Chen & Chao, 2020). This approach allows us to smoothly adapt the level of heterogeneity in the client data via the concentration parameter α. To generate the data split, we sample c vectors

$$p_1, .., p_c \sim \text{Dir}(\alpha), \tag{11}$$

where c is the number of classes, from the symmetric n-categorical Dirichlet distribution. For all $p_i \in \mathbb{R}^n_{\geq 0}$ it then holds $\|p_i\|_1 = 1$. The vectors are then stacked To address the statistical heterogeneity, methods for personalizing the server model to the client's local distributions, e.g. by using distillation (Li & Wang, 2019), parameter fine-tuning (Wang et al., 2019; Mansour et al., 2020) or regularization (Li et al., 2020a), have been proposed. Transferring knowledge from one domain to another domain raises the question of the general

into a matrix

$$P = [p_1, .., p_c] \in \mathbb{R}^{n,c} \tag{12}$$

which is standardized, by repeatedly normalizing the columns and rows. This process converges quickly and is stopped after 1000 iterations. Let M_j be the amount of data points belonging to class j in the training data set. Each client i is then assigned $P_{i,j}M_j$ (non-overlapping) data points from all classes $j = 1, .., c$. Figure 8 illustrates the splitting procedure and displays random splits of data for $n = 20$ and $c = 10$. In all our experiments, the data splitting process is controlled by a random seed, to ensure that the different baseline methods are all trained on the same split of data.

C. Detailed Algorithm

The training procedure of FEDAUX can be divided into a preparation phase, which is given in Alg. 1 and a training phase, which is given in Alg. 2. We describe the general setting where clients may hold different model prototypes P from a set of prototypes $\mathcal{P}$. This general setting simplifies to the setting described in Sec. 3 if $|\mathcal{P}| = 1$.

Preparation Phase: In the preparation phase, the server uses the unlabeled auxiliary data D_{aux}, to pre-train the feature extractor h^P for each model prototype P using self-supervised training. Suitable methods for self-supervised pre-training are contrastive representation learning (Chen et al., 2020), or self-supervised language modeling/ next-token prediction (Devlin et al., 2019). The pre-trained feature extractors h^P_0 are then communicated to the clients and used to initialize part of the local classifier $f = g \circ h$. The server also communicates the negative data D^- to the clients (in practice we can instead communicate the extracted features $\{|h^P_0(x)|x \in D^-\}$ of the raw data D^- to save communication). Each client then optimizes the logistic similarity objective J (4) and sanitizes the output by adding properly scaled Gaussian noise. Finally, the sanitized scoring model w^*_i is communicated to the server, where it is used to compute certainty scores s_i on the distillation data (the certainty scores can also be computed on the clients, however this results in additional communication of distillation data and scores).

Algorithm 2 FEDAUX Training Phase (with different model prototypes $\mathcal{P}$). Training requires feature extractors h^P_0 and scores s_i from Alg. 1. The same $D^- \cup D_{distill} \leftarrow D_{aux}$ as in Alg. 1 is used. Choose learning rate η and set $\xi = 10^{-8}$.

init: HashMap $\mathcal{R}$ that maps client i to model prototype P
init: Inverse HashMap $\tilde{\mathcal{R}}$ that maps model prototype P to set of clients (s.t. $i \in \tilde{\mathcal{R}}[\mathcal{R}[i]]\ \forall i$)
init: Initialize model prototype weights θ^P with feature extractor weights h^P from Alg. 1
for communication round $t = 1, .., T$ **do**
  select subset of clients $\mathcal{S}_t \subseteq \{1, .., n\}$
  for selected clients $i \in \mathcal{S}_t$ **in parallel do**
    Client i does:
    $\theta_i \leftarrow \text{train}(\theta_0 \leftarrow \theta^{\mathcal{R}[i]}, D_i)$ # Local Training
  end for
  Server does:
  for each model prototype $P \in \mathcal{P}$ **do**
    $\theta^P \leftarrow \sum_{i \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} \frac{|D_i|}{\sum_{l \in \mathcal{S}_t \cap \tilde{\mathcal{R}}[P]} |D_l|} \theta_i$ # Parameter Averaging
    for mini-batch $x \in D_{distill}$ **do**
      $\tilde{y} \leftarrow \sigma\left(\frac{\sum_{i \in \mathcal{S}_t} s_i[x] f_i(x, \theta_i)}{\sum_{i \in \mathcal{S}_t} s_i[x]}\right)$ # Can be arbitrary
      $\theta^P \leftarrow \theta^P - \eta \frac{\partial D_{KL}(\tilde{y}, \sigma(f(x, \theta^P)))}{\partial \theta^P}$ # Optimizer
    end for
  end for
end for

Training Phase: The training phase is carried out in T communication rounds. In every round $t \leq T$, the server randomly selects a subset $\mathcal{S}_t$ of the overall client population and transmits to them the latest server models $\theta^{\mathcal{R}}[i]$, which match their model prototype P (in round $t = 1$ only the pre-trained feature extractor h^P_0 is transmitted). Each selected client updates it's local model by performing multiple steps of stochastic gradient descent (or it's variants) on it's local training data. This results in an updated parameterization θ_i on every client, which is communicated to the server. After all clients have finished their local training, the server gathers the updated parameters θ_i. For each model prototype P the corresponding parameters are then aggregated by

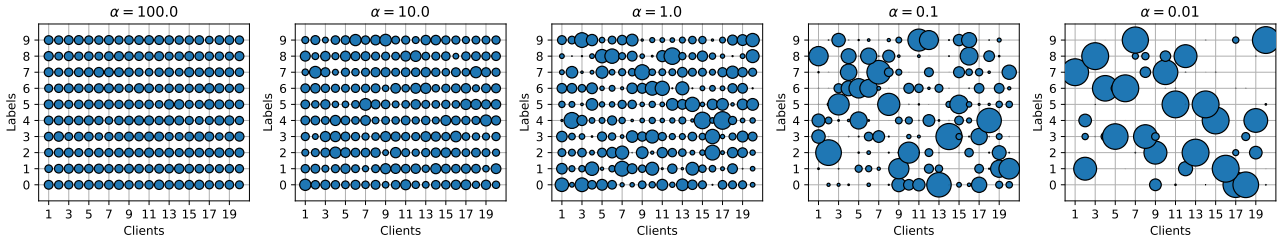


Figure 8. Illustration of the Dirichlet data splitting strategy used throughout the paper. Dot size represents number of data points each client holds from any particular class. Lower values of α lead to more heterogeneous splits of data.

weighted averaging. Using the model averages as a starting point, for each prototype the server then distills a new model, based on the client's certainty-weighted predictions.

D. Qualitative Comparison with Baseline Methods

Table 4 gives a qualitative comparison between FEDAUX and the baseline methods FEDAVG and FEDDF.

- Compared with FEDAVG and FEDDF, FEDAUX additionally requires the clients to once solve the λ-strongly convex ERM (4). For this problem linearly convergent algorithms are known (Liu & Nocedal, 1989) and thus the computational overhead is negligible compared with the complexity of multiple rounds of locally training deep neural networks.

- FEDAUX also adds computational load to the server for self-supervised pre-training and computation of the certainty scores s_i. As the server is typically assumed to have massively stronger computational resources than the clients, this can be neglected.

- Once, in the preparation phase of FEDAUX, the scoring models w_i^* need to be communicated from the clients to the server. The overhead of communicating these H-dimensional vectors, where H is the feature dimension, is negligible compared to the communication of the full models f_i.

- FEDAUX also requires the communication of the negative data D^- and the feature extractor h_0 from the server to the clients. The overhead of sending h_0 is lower than sending the full model f, and thus the total downstream communication is increased by less than a factor of $(T+1)/T$. The overhead of sending D^- is small (in our experiments $|D^-| = 0.2|D_{aux}|$) and can be further reduced by sending extracted features $\{|h_0^P(x)|x \in D^-\}$ instead of the full data. For instance, in our experiments with ResNet-8 and CIFAR-100 we have $|D^-| = 12000$ and $h_0^P(x) \in \mathbb{R}^{512}$, resulting in a total communication overhead of $12000 \times 512 \times 4B = 24.58$MB for D^-. For comparison the total communication overhead of once sending the parameters of ResNet-8 (needs to be done T times) is 19.79MB.

- Communicating the scoring models w_i^* incurs additional privacy loss for the clients. Using our proposed sanitation mechanism this process is made (ε, δ)-differentially private. Our experiments in section 4.4 demonstrate that FEDAUX can achieve drastic performance improvements, even under conservative privacy constraints. All empirical results reported are obtained with (ε, δ) differential privacy at $\varepsilon = 0.1$ and $\delta = 10^{-5}$.

- Finally, FEDAUX makes the additional assumption that unlabeled auxiliary data is available to the server. This assumption is made by all Federated Distillation methods including FEDDF.

E. Additional Results and Detailed Training Curves

In this sections we give detailed training curves for the results shown in Figure 3. As can be seen, in the highly non-iid setting at $\alpha \in \{0.01, 0.04\}$, all methods exhibit convergence issues. This behavior is well known in FL and is described for instance in (Zhao et al., 2018; Sattler et al., 2020c). Notably, the performance of FEDAUX after one single communication round exceeds the maximum achieved performance of all other methods over the entire course of training. At higher values of $\alpha \geq 0.16$ all methods train smoothly and validation performance asymptotically increases over the curse of training. FEDAUX dominates all baseline methods at all communication rounds in the heterogeneous settings. In the mostly iid-setting at $\alpha = 10.24$ FEDAUX is en par with the pre-trained version of FEDDF.

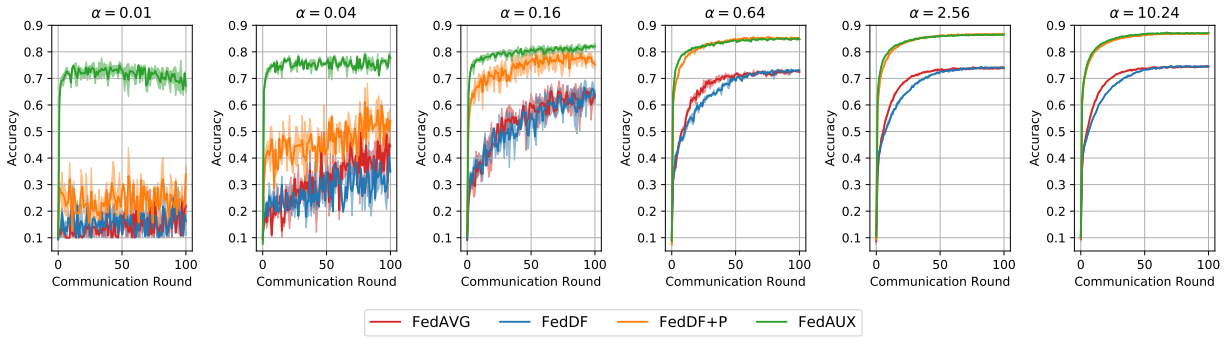


Figure 9. Detailed training curves for ResNet-8 trained on CIFAR-10, $n = 80$ Clients, $C = 40\%$.

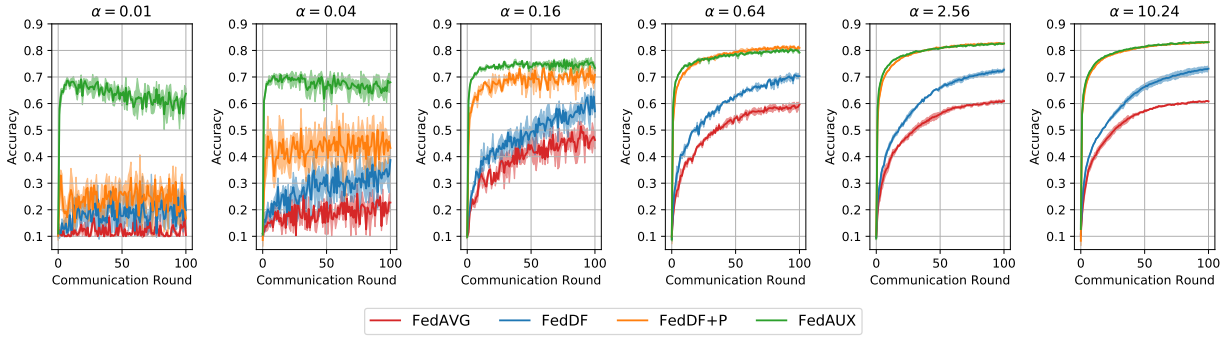


Figure 10. Detailed training curves for MobileNetv2 trained on CIFAR-10, $n = 100$ Clients, $C = 40\%$.

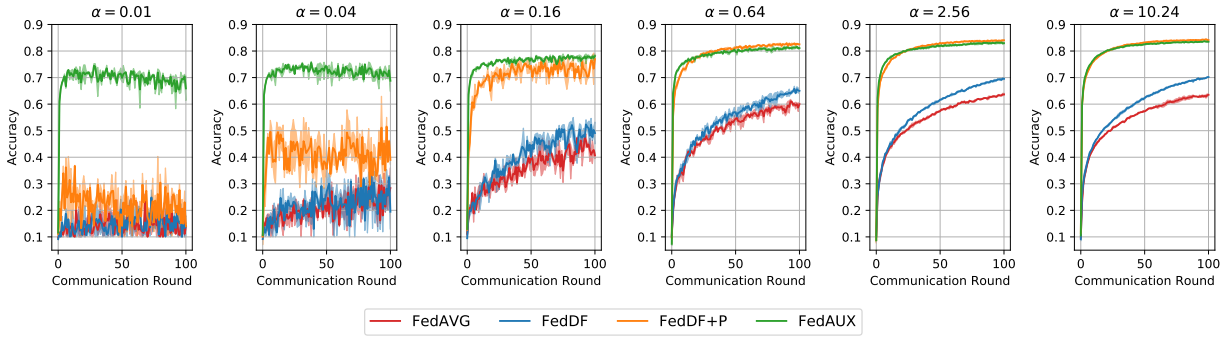


Figure 11. Shufflenet trained on CIFAR-10, $n = 100$ Clients, $C = 40\%$.

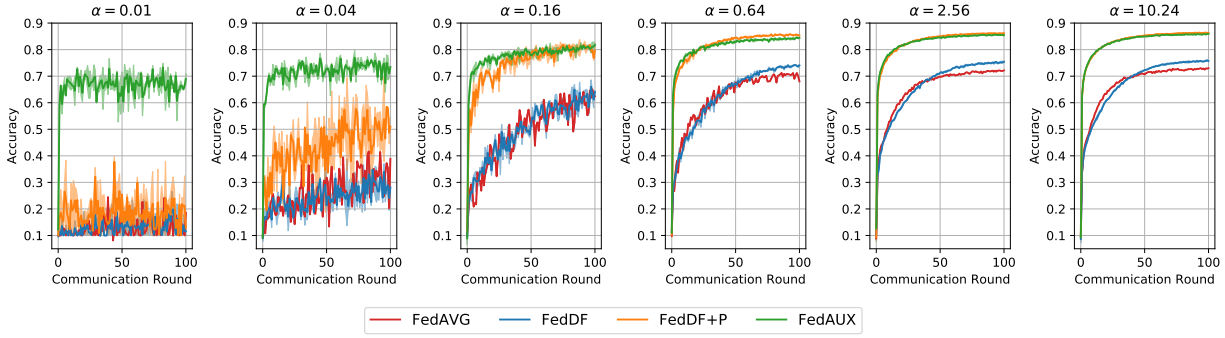


Figure 12. Detailed training curves for mixed models trained on CIFAR-10. 20 each train ResNet8, MobileNetv2 and Shufflenet respectively.

Table 4. **Qualitative Comparison:** Complexity, communication overhead, privacy loss after T communication rounds as well as implicit assumptions made by different Federated Learning methods.

	FEDAVG	FEDDF	FEDAUX (preparation phase)	FEDAUX (training phase)
Operations (Clients)	Local Training ($\times T$)	Local Training ($\times T$)	Solve λ-strongly convex ERM (4)	Local Training ($\times T$)
Operations (Server)	Model Averaging ($\times T$)	Model Averaging, Distillation ($\times T$)	Self-Supervised Pre-training of h_0, Computation of certainty scores s_i	Model Averaging, Distillation ($\times T$)
Communication Clients $\rightarrow$ Server	Model Parameters f_i ($\times T$)	Model Parameters f_i ($\times T$)	Scoring Models w_i^*	Model Parameters f_i ($\times T$)
Communication Server $\rightarrow$ Clients	Model Parameters f ($\times T$)	Model Parameters f ($\times T$)	Negative Data D^-, Feature Extractor h_0	Model Parameters f ($\times T$)
Privacy Loss	Privacy loss of communicating f_i ($\times T$)	Privacy loss of communicating f_i ($\times T$)	(ε, δ)-DP	Privacy loss of communicating f_i ($\times T$)
Assumptions	No Assumptions	Auxiliary Data	Auxiliary Data	Auxiliary Data

Table 5. Results on data sets with **higher number of classes.** Training ResNet-8 on **CIFAR-100**. Accuracy achieved after $T = 100$ communication rounds by different Federated Distillation methods at different levels of data heterogeneity α. STL-10 is used as auxiliary data set.

	α					
	0.01	0.04	0.16	0.64	2.56	10.24
FedAVG	24.1	36.3	47.2	50.7	52.2	52.2
FedDF	11.4	24.4	45.0	49.5	52.5	51.2
FedDF+P	18.2	42.0	**58.0**	**60.8**	61.6	62.0
FedAUX	**34.1**	**47.4**	56.4	60.7	**62.5**	**62.5**

Table 6. **Auxiliary data sets** used in this study and their defining Wordnet IDs and data sets sizes.

Data set	Wordnet ID	Dataset Size
Imagenet Devices	n03183080	165747
Imagenet Birds	n01503061	76541
Imagenet Animals	n00015388	510530
Imagenet Dogs	n02084071	147873
Imagenet Invertebrates	n01905661	79300
Imagenet Structures	n04341686	74400

Table 5 compares performance of FEDAUX to baseline methods on the CIFAR-100 data set. Again FEDAUX outperforms FEDAVG and FEDDF across all level of data heterogeneity α and shows superior performance to the improved FEDDF+P when data is highly heterogeneous at $\alpha = \{0.01, 0.04\}$. Interestingly in this setting FEDDF+P manages to slightly outperform FEDAUX at medium data heterogeneity levels $\alpha = \{0.16, 0.64\}$. This indicates that our proposed differentially private certainty scoring method may insufficiently approximate the true client certainty in this setting. We leave potential improvements of this mechanism for future work.

F. Details on generating Imagenet subsets

To simulate the effects of a wide variety of auxiliary data sets on the training performance of FEDAUX, we generate different structured subsets of the ImageNet data base (resized to $32 \times 32 \times 3$). Each subset is defined via a top-level Wordnet ID which is shown in Table 6. To obtain the images from the subset, we select all leaf-node IDs of the respective top-level IDs via the Imagenet API

`http://www.image-net.org/api/text/wordnet.structure.hyponym?wnid=<top-levelID>&full=1`

and then take only those classes from the full Imagenet data set, which match these leaf-node IDs. Table 6 also shows the number of samples contained in every subset that was generated this way.

G. Details on the Implementation and Results of the NLP Benchmarks

As mentioned in section 4.3 *Evaluating* FEDAUX *on NLP Benchmarks* we used TinyBERT as a model for our NLP experiments. TinyBERT was pre-trained on Bookcorpus[4] which led us to select the same dataset as a public dataset in order to follow the methodology outlined in section 3.3. As private datasets we chose the AG News dataset[5] (Zhang et al., 2015), a topic classification dataset, and the english texts from the Multilingual Amazon Reviews Corpus[6] (Keung et al., 2020), which we use for predicting how many stars a review gets. The pre-trained weights and the tokenizer for TinyBERT are available at the corresponding repository[7]. All experiments were conducted using $\epsilon = 0.1$ and $\delta = 10^{-5}$ as differential privacy parameters, 1 epoch for local training and distillation, ten clients and 100% participation rate as well as 160000 disjoint data points, which were sampled from BookCorpus, for the public and distillation datasets respectively. Furthermore the ADAM optimizer with a learning rate of 10^{-5} was used for both local training and distillation. The regularization strength of the logistic regression classifier was set to 0.01. The batch size for D_i, D^- and $D_{distill}$ was 32. Detailed results for figure 4 are depicted in table 7.

H. Hyperparameter Evaluation

In this section we provide a detailed hyperparameter analysis for our proposed method and the baseline methods used in this study. For all methods we use the very popular Adam optimizer for both local training and distillation. We vary the learning rate in $\{1e-2, 1e-3, 1e-4, 1e-5\}$ for local training an distillation. For FedPROX, we vary the parameter λ_{prox}, controlling the proximal term in the training objective in $\{1e-2, 1e-3, 1e-4, 1e-5\}$. Figure 13 compares the maximum achieved accuracy after 50 communication rounds for the different methods and hyperparameter settings, for a FL setting with 20 clients training ResNet-8 on CIFAR-10 at a participation-rate of 40%. The auxiliary data set we use is STL-10.

For each method and each level of data heterogeneity, table 8 shows the accuracy of the best performing combination of hyperparameters. As we can see FEDAUX matches the performance of the best performing methods in the iid setting with $\alpha = 100.0$ and outperforms all other methods distinctively in the non-iid setting with $\alpha = 0.01$.

[4] https://huggingface.co/datasets/bookcorpus

[5] https://huggingface.co/datasets/ag_news

[6] https://huggingface.co/datasets/amazon_reviews_multi

[7] https://huggingface.co/huawei-noah/TinyBERT_General_4L_312D

I. Domain-Adaptation-Theoretic Motivation for weighted ensemble distillation

Domain adaptation theory (Mansour et al., 2008; Ben-David et al., 2010; Hoffman et al., 2018), and in particular with multiple sources, can be used in order to obtain generalization bounds for non-iid FL settings as it has been done in (Lin et al., 2020) for uniformly averaging of the client hypotheses to obtain a global hypothesis. From multiple-source adaptation theory we know that a distribution-weighted combination of the client hypotheses is robust w.r.t. generalization for any target domain that is a convex combination of the source domains. However, exact information about the local distributions is rarely present in practical applications of FL and if it is, then directly sharing this information with the server in order to get a better global hypothesis is often not feasible in FL settings due to privacy restrictions. Nonetheless, settings with exact or approximate information about the local distributions (e.g. obtained by KDE) show us, what is possible if the server had access to this information and thus leads to benchmarks with a solid theoretic foundation to which we can compare our approach. Consequently, we aim at a weighting of the client's local hypotheses based on a privacy-preserving local distribution estimation that respects both the theoretical generalization capabilities and the privacy restrictions in FL.

With the help of a toy example in Fig. 14 we illustrates that the certainty scores $s_i(\cdot), i \in \{1, \dots, n\}$, obtained via privacy-preserving logistic regression give a good approximation to the distribution-weights suggested by domain adaptation theory (Mansour et al., 2008), i.e. we show that $s_i(x)/\sum_j s_j(x) \approx D_i(x)/\sum_j D_j(x)$ for $x \in \mathcal{X}$.

J. Proof of Theorem 1

Theorem 2. *If $R(\cdot)$ is differentiable and 1-strongly convex and l is differentiable with $|l'(z)| \leq 1\ \forall z$, then the ℓ^2-sensitivity $\Delta_2(\mathcal{M})$ of the mechanism*

$$\mathcal{M} : D_i \mapsto \arg\min_w J(w, h_0, D_i, D^-) \tag{13}$$

is at most $2(\lambda(|D_i| + |D^-|))^{-1}$.

Proof. The proof is an adaptation of the result shown in (Chaudhuri et al., 2011). We have

$$J(w, h_0, D_i, D^-) = a \sum_{x \in D_i \cup D^-} l(t_x \langle w, \tilde{h}_0(x) \rangle) + \lambda R(w) \tag{14}$$

with $t_x = 2(\mathbb{1}_{x \in D_i}) - 1 \in [-1, 1]$, $a = (|D_i| + |D^-|)^{-1}$ and $\tilde{h}_0(x) = h_0(x)(\max_{x \in D^- \cup D_i} \|h_0(x)\|)^{-1}$.

Let $D_i = \{x_1, .., x_N\}$ and $D_i' = \{x_1, .., x_N'\}$ be two local data sets that differ in only one element. For arbitrary D^-

Table 7. **NLP Benchmarks** of different FL methods. Maximum accuracy achieved after $T = 20$ communication rounds at participation-rate $C = 100\%$.

Method	AG News		Amazon	
	$\alpha = 0.01$	$\alpha = 1.0$	$\alpha = 0.01$	$\alpha = 1.0$
FEDAVG+P	78.80±4.40	**92.17±1.98**	41.70±0.58	**55.17±0.40**
FEDDF+P	78.05±7.64	90.83±0.25	38.04±0.84	54.63±0.66
FEDAUX	**85.04±1.21**	91.00±0.30	**49.11±0.22**	54.86±0.61

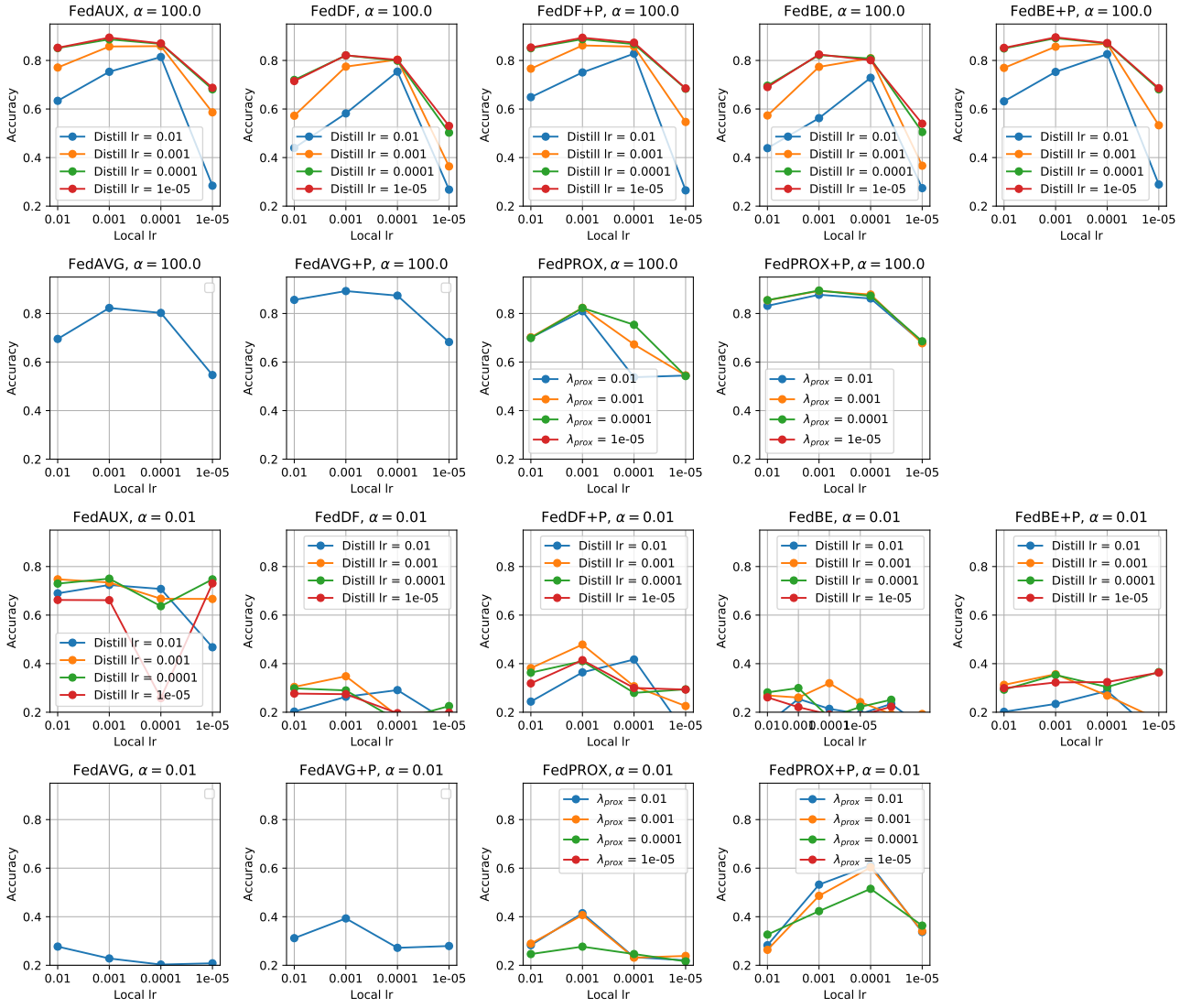


Figure 13. Results of our **hyperparameter optimization** for ResNet8. 20 Clients are trained for 50 communication rounds, at a participation rate of $C = 40\%$. Both local training and distillation is performed for 1 epoch.

and h_0 define

$$w^* = \arg\min_w J(w, h_0, D_i, D^-), \tag{15}$$

$$v^* = \arg\min_w J(w, h_0, D_i', D^-), \tag{16}$$

$$n(w) = J(w, h_0, D_i, D^-) \tag{17}$$

Table 8. **Best performing hyperparameter combinations** for each method when training ResNet8 with $n = 20$ clients for 50 communication rounds at a participation rate of $C = 40\%$. Both local training and distillation is performed for 1 epoch. Methods sorted by top accuracy.

Method	Alpha	Local LR	Distill LR	λ FedProx	Accuracy
FedPROX+P	100	0.001	-	0.0001	0.8946
FedAUX		0.001	1e-05	-	0.8941
FedDF+P		0.001	1e-05	-	0.8936
FedAVG+P		0.001	-	-	0.8924
FedBE		0.001	1e-05	-	0.8246
FedPROX		0.001	-	0.001	0.8232
FedAVG		0.001	-	-	0.8228
FedDF		0.001	1e-05	-	0.8210
FedAUX	0.01	0.001	0.0001	-	0.7501
FedPROX+P		0.01	-	0.01	0.6122
FedDF+P		0.001	0.001	-	0.4786
FedPROX		0.001	-	0.01	0.4145
FedAVG+P		0.001	-	-	0.3929
FedDF		0.001	0.001	-	0.3481
FedBE		0.001	0.001	-	0.3196
FedAVG		0.0001	-	-	0.2770

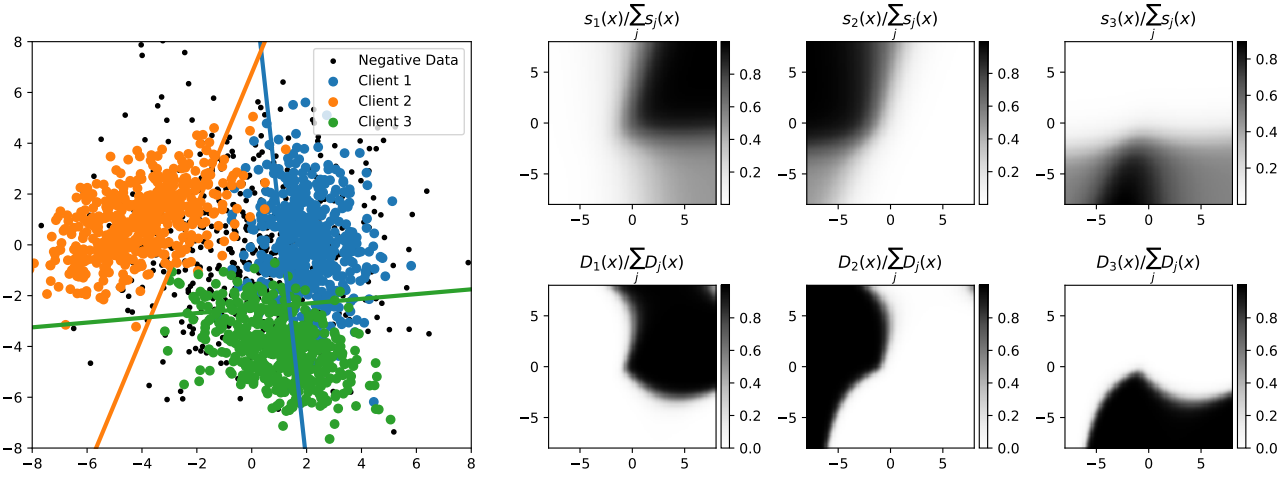


Figure 14. Left: Toy example with 3 clients holding data sampled from multivariate Gaussian distributions D_1, D_2 and D_3. All clients solve optimization problem J by contrasting their local data with the public negative data, to obtain scoring models s_1, s_2, s_3 respectively. As can be seen in the plots to the right, our proposed scoring method approximates the robust weights proposed in (Mansour et al., 2008) as it holds $s_i(x)/\sum_j s_j(x) \approx D_i(x)/\sum_j D_j(x)$ on the support of the data distributions.

and

$$m(w) = J(w, h_0, D_i, D^-) - J(w, h_0, D_i', D^-) \quad (18)$$

we have

$$m(w) = a(l(t_x\langle w, h_0(x_N)\rangle) - l(t_x\langle w, h_0(x_N')\rangle)) \quad (19)$$

Since

$$\nabla m(w) = a(t_x l'(t_x\langle w, h_0(x_N)\rangle)h_0(x_N)^T - \quad (20)$$

$$t_x l'(t_x\langle w, h_0(x_N')\rangle)h_0(x_N')^T) \quad (21)$$

which can be bounded in norm

$$\|\nabla m(w)\| = a(\|h_0(x_N) - h_0(x'_N)\|) \tag{22}$$
$$\leq a(\|h_0(x_N)\| + \|h_0(x'_N)\|) \tag{23}$$
$$\leq 2a \tag{24}$$

as $t_x \in [-1, 1]$, $|l'(x)| \leq 1$ and

$$\|\tilde{h}_0(x)\| = \|h_0(x)(\max_{x \in D_i \cup D^-} h_0(x))^{-1}\| \leq 1. \tag{25}$$

Furthermore, since $n(w)$ is λ-strongly convex it follows by Shalev-Schwartz inequality

$$(\nabla n(w^*) - \nabla n(v^*))^T (w^* - v^*) \geq \lambda \|w^* - v^*\|^2. \tag{26}$$

Combining this result with Cauchy-Schwartz inequality and $\nabla m(v^*) = \nabla n(v^*) - \nabla n(w^*)$ yields

$$\|w^* - v^*\| \|\nabla m(v^*)\| \geq (w^* - v^*)^T \nabla m(v^*) \tag{27}$$
$$= (w^* - v^*)^T (\nabla n(v^*) - \nabla n(w^*)) \tag{28}$$
$$\geq \lambda \|w^* - v^*\|^2 \tag{29}$$

Thus

$$\|w^* - v^*\| \leq \frac{\|\nabla m(v^*)\|}{\lambda} \leq \frac{2a}{\lambda} \tag{30}$$

which concludes the proof. □

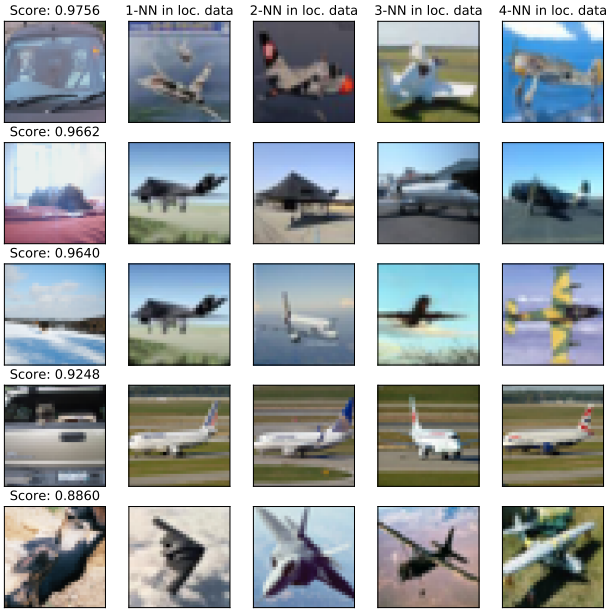
K. Empirical Privacy Evaluation

Our proposed method is provably differentially private and achieves state-of-the-art performance, even at very conservative privacy levels. If not explicitly stated otherwise, all results presented in this study were achieved with (ε, δ)-differentially private certainty scores at conservative privacy parameters $\delta = 10^{-5}$ and $\varepsilon = 0.1$. In this section, we additionally evaluate the privacy properties of the certainty scores empirically. Figure 15 shows, for four different clients, the 5 images x from the distillation data set $D_{distill}$, which were assigned the highest certainty score $s_i(x)$ by the client's scoring model w_i^* (left column). Displayed next to the images are their 4 nearest neighbors x' in feature space which maximize the cosine-similarity

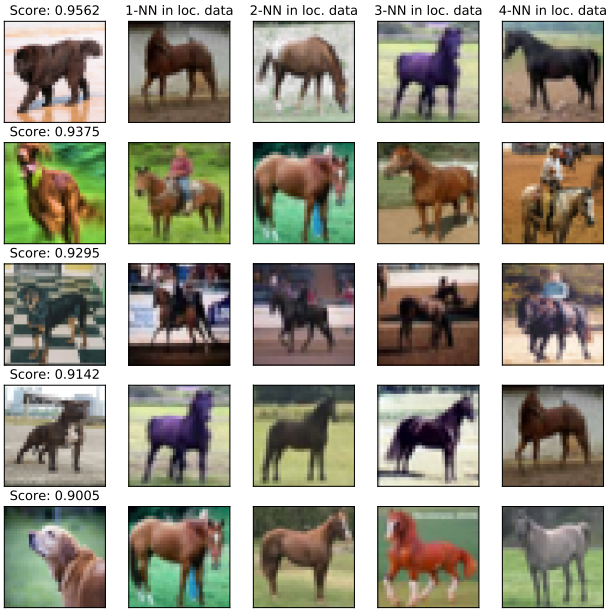
$$\text{sim}(x, x') = \frac{\langle h_0(x), h_0(x') \rangle}{\|h_0(x)\| \|h_0(x')\|}. \tag{31}$$

In this example the clients hold non-iid subsets of CIFAR-10 ($\alpha = 0.01$) and the "Imagenet Dogs" (c.f. Appendix F) data set is used as auxiliary data. Using weighted ensemble distillation in this setting improves training performance from 48.46% to 75.59%. As we can see, while certainty scores are able to inform the distillation process and allow FEDAUX to outperform baseline methods on heterogeneous data, they reveal only fuzzy, indirect information about the local training data. For instance, client 1, which in this example is mainly holding data from the airplane class, assigns the highest scores to pictures in the auxiliary data set that show dogs in cars or in front of blue skies. From this it could be concluded that a majority of the clients training data contains man-made objects in front of blue backgrounds, but direct exposure of single data points is improbable.

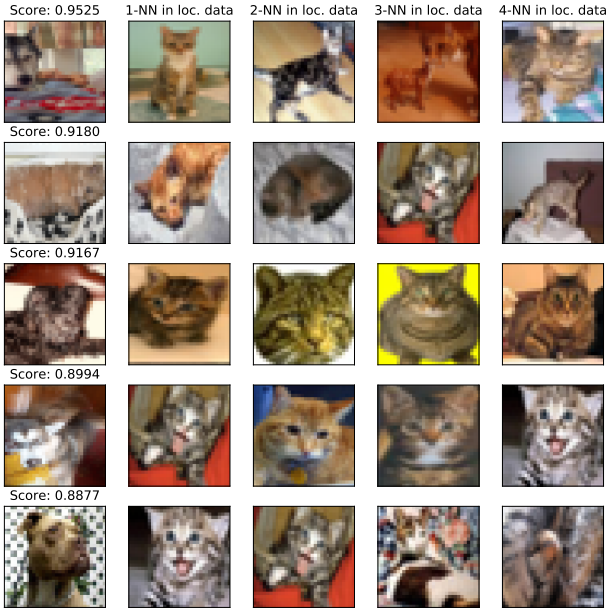
Note that there exist also many FL scenarios in which the server is assumed to be trustworthy, and only the final trained model which is released to the public needs to be privately sanitized. In these settings, direct inspection of certainty scores by outside adversaries is not possible and thus privacy loss through certainty scores is even less critical. Future work could also explore the use encryption-based techniques for secure weighted aggregation of client predictions.



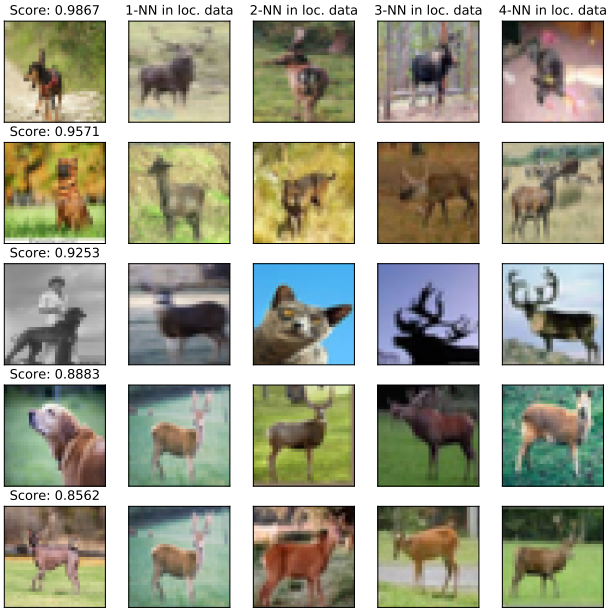
(a) Client 1: Images x from the distill data set with the highest scores $s_i(x)$ and their nearest neighbors in feature space in the local data set D_i.



(b) Client 2: Images x from the distill data set with the highest scores $s_i(x)$ and their nearest neighbors in feature space in the local data set D_i.

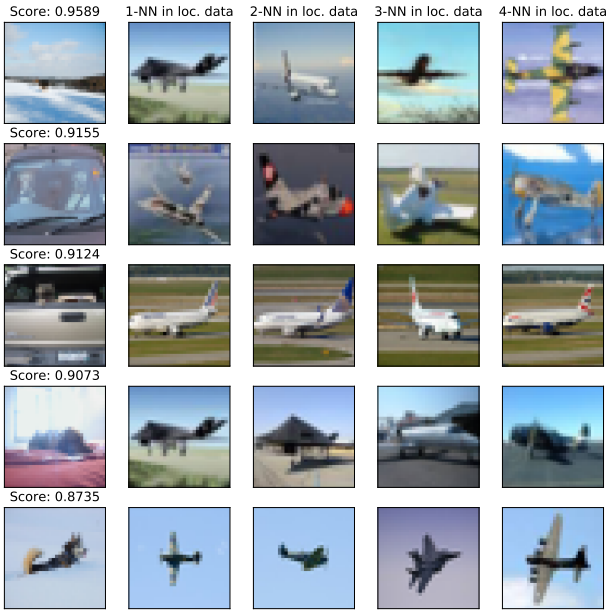


(c) Client 3: Images x from the distill data set with the highest scores $s_i(x)$ and their nearest neighbors in feature space in the local data set D_i.

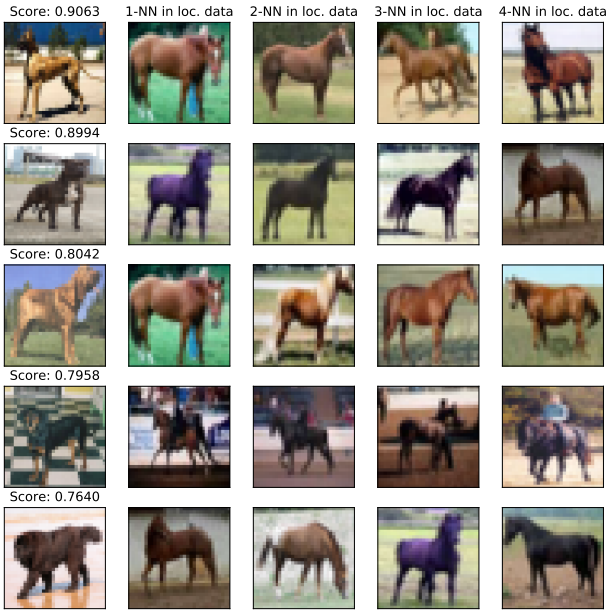


(d) Client 4: Images x from the distill data set with the highest scores $s_i(x)$ and their nearest neighbors in feature space in the local data set D_i.

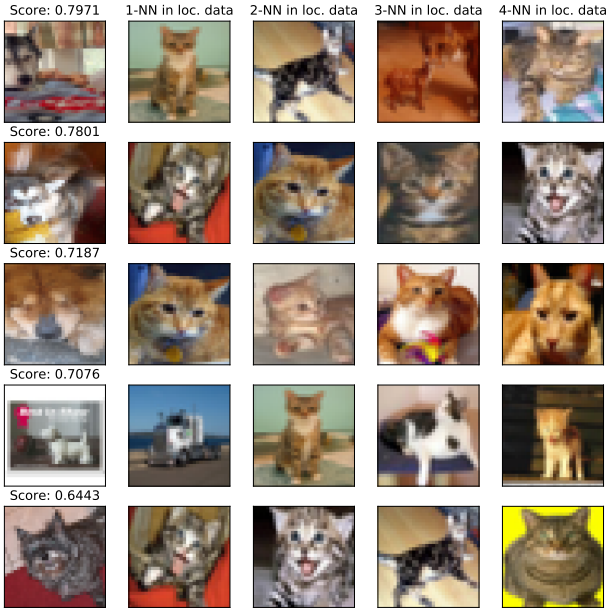
Figure 15. Data points x from the auxiliary data set which were assigned the highest scores $s_i(x)$ and their nearest neighbors in the data of 4 randomly selected clients D_i. Clients hold non-iid subsets from the CIFAR-10 data set ($\alpha = 0.01$). Auxiliary data used is ImageNet Dogs (cf. Appendix F). No differential privacy is used.



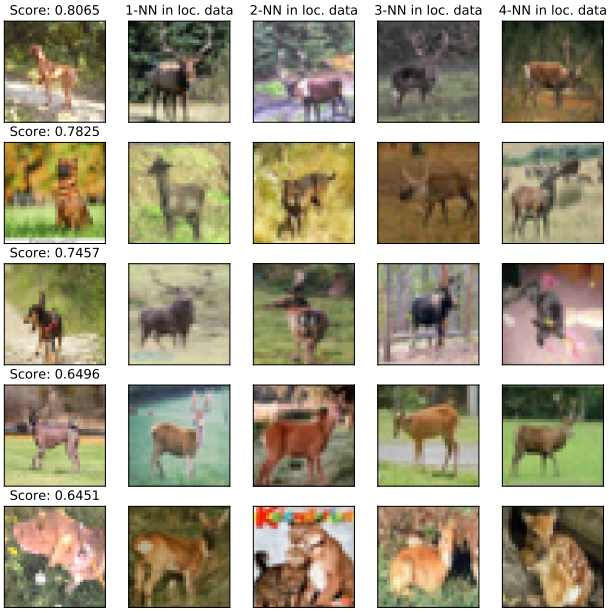
(a) Images from the distill data set with the higher scores and their nearest neighbors in feature space in the local data set of client 1.



(b) Images from the distill data set with the higher scores and their nearest neighbors in feature space in the local data set of client 2.



(c) Images from the distill data set with the higher scores and their nearest neighbors in feature space in the local data set of client 3.



(d) Images from the distill data set with the higher scores and their nearest neighbors in feature space in the local data set of client 4.

Figure 16. Data points x from the auxiliary data set which were assigned the highest scores $s_i(x)$ and their nearest neighbors in the data of 4 randomly selected clients D_i. Clients hold non-iid subsets from the CIFAR-10 data set ($\alpha = 0.01$). Auxiliary data used is ImageNet Dogs (cf. Appendix F). Scores obtained with differential privacy at $\varepsilon = 0.1$, $\delta = 10^{-5}$.