Little is Enough: Boosting Privacy by Sharing Only Hard Labels in Federated Semi-Supervised Learning

Amr Abourayya [1,2], Jens Kleesiek [1], Kanishka Rao [5], Erman Ayday [4], Bharat Rao [5], Geoffrey I. Webb [3], Michael Kamp [1,2,3,5]

[1] Institute for AI in medicine (IKIM),University Hospital Essen, Germany
[2] Institute for Neuroinformatics, Ruhr University Bochum, Germany
[3] Department of Data Science & AI, Monash University, Australia
[4] Department of Computer and Data Sciences, Case Western Reserve University, USA
[5] Carenostics, USA
Amr.Abourayya@uk-essen.de , Michael.Kamp@uk-essen.de

Abstract

In many critical applications, sensitive data is inherently distributed and cannot be centralized due to privacy concerns. A wide range of federated learning approaches have been proposed to train models locally at each client without sharing their sensitive data, typically by exchanging model parameters, or probabilistic predictions (soft labels) on a public dataset or a combination of both. However, these methods still disclose private information and restrict local models to those that can be trained using gradient-based methods. We propose a federated co-training (FEDCT) approach that improves privacy by sharing only definitive (hard) labels on a public unlabeled dataset. Clients use a consensus of these shared labels as pseudo-labels for local training. This federated co-training approach empirically enhances privacy without compromising model quality. In addition, it allows the use of local models that are not suitable for parameter aggregation in traditional federated learning, such as gradient-boosted decision trees, rule ensembles, and random forests. Furthermore, we observe that FEDCT performs effectively in federated fine-tuning of large language models, where its pseudo-labeling mechanism is particularly beneficial. Empirical evaluations and theoretical analyses suggest its applicability across a range of federated learning scenarios.

Extended version — https://arxiv.org/abs/2310.05696
Code — github.com/kampmichael/federatedcotraining

1 Introduction

Can we train models using distributed sensitive datasets while maintaining data privacy? Federated learning (FEDAVG) (McMahan et al. 2017) addresses this challenge by collaboratively training a joint model without directly disclosing distributed sensitive data, but instead sharing information from locally trained models. Most approaches share model parameters that are aggregated at a server (Kamp 2019; Mcdonald et al. 2009; Kairouz et al. 2021), most prominently federated averaging (FEDAVG) (McMahan et al. 2017). This results in high model performance, but allows an attacker or curious observer to make non-trivial inferences about local data from those model parameters (Ma et al. 2020) or model updates (Zhu and Han 2020). Adding tailored noise to the parameters before sharing improves privacy and provides differential privacy guarantees, as in differentially private FEDAVG (Wei et al. 2020) (DP-FEDAVG), but reduces model performance (Xiao, Wan, and Devadas 2022). An alternative is to use a form of federated semi-supervised learning that shares information via a public unlabeled dataset, which can improve privacy—most commonly soft labels are shared (Gong et al. 2022; Lin et al. 2020; Jiang, Shan, and Zhang 2020; Li and Wang 2019), e.g., as in distributed distillation (Bistritz, Mann, and Bambos 2020) (DD). (Struppek, Hintersdorf, and Kersting 2024) showed, however, that data can be reconstructed also from soft labels. In Fig. 1, we compare the empirical privacy vulnerability of these approaches: Although sharing soft labels (DD) offers a better privacy-utility trade-off than DP-FEDAVG and FEDAVG, the gap to ideal privacy (VUL$=0.5$, where membership inference attacks are akin to random guessing) remains significant.

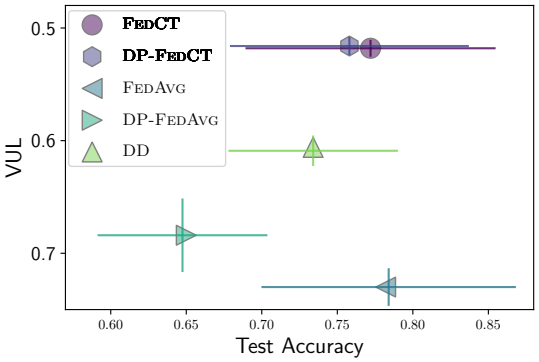

Figure 1: Vulnerability (VUL) to membership inference attacks on the communication of 5 clients and their test accuracy (avg and std over 5 datasets). VUL is measured empirically as the success probability of infering membership correctly, VUL $= 0.5$ implies optimal privacy.

We propose instead to share definitive class labels (hard labels) to improve privacy: We develop a novel federated co-training approach (FEDCT) where clients iteratively share

hard labels on an unlabeled public dataset. A server forms a consensus of these predictions, e.g., via majority vote (Blum and Mitchell 1998), and clients use this consensus as pseudo-labels for the unlabeled dataset in their local training.

We prove convergence of this approach and empirically show that it provides a favorable privacy-utility trade-off (see Fig. 1), achieving near-optimal privacy levels. Using a majority vote as consensus mechanism results in a model performance that is at least en par with model parameter and soft-label sharing. In tasks where the initial models already provide good predictions, such as fine-tuning of LLMs, FEDCT can even outperform FEDAVG. We show how differential privacy guarantees (Ziller et al. 2021; Chaudhuri, Imola, and Machanavajjhala 2019) can be provided for FEDCT via a suitable noise mechanism for binary outputs, which not only protect privacy but can improve robustness against poisoning and backdoor attacks (Bagdasaryan et al. 2020; Sun et al. 2019). For that, we provide a novel bound on the sensitivity of hard label sharing for local learning algorithms that are on-average-leave-one-out-stable. Due to the inherent robustness of majority voting (Papernot et al. 2017), the added noise in differentially private federated co-training (DP-FEDCT) does not reduce model performance as much as in FEDAVG. As Bistritz, Mann, and Bambos (2020) noted, sharing labels can also reduce communication substantially, improving scalability. We confirm this for FEDCT, reducing communication over FEDAVG by up to two orders of magnitude.

While federated semi-supervised learning performs well on heterogeneous feature distributions, both soft and hard label often underperform on data with heterogeneous label distributions, since local models are unable to provide meaningful predictions for labels they have not observed during training. With mild-to-medium data heterogeneity, label sharing performs well, but its performance drops for pathological distributions. We propose using a qualified majority as a consensus for pathological distributions which improves the performance substantially, making FEDCT competitive with FEDAVG also for pathological non-iid data.

A positive side-effect of sharing hard labels is that one is no longer limited to models that lend themselves to parameter aggregation as in FEDAVG, or gradient-based training as used in soft-label sharing: With FEDCT we can train, e.g., interpretable models, such as decision trees (Quinlan 1986), XGBoost (Friedman 2001; Chen and Guestrin 2016), rule ensembles (Friedman and Popescu 2008), and Random Forests (Breiman 2001), as well as mixtures of model types.

In summary, our contributions are:

(i) An investigation of the privacy-utility trade-off between sharing model parameters, soft labels, and hard labels via a novel federated co-training approach (FEDCT);

(ii) A novel sensitivity bound on sharing hard labels for local learning algorithms that are on-average-replae-one stable;

(iii) A theoretical analysis of the convergence and sensitivity, and an extensive empirical evaluation of FEDCT.

2 Related Work

We briefly discuss closely related work here, focussing on privacy in federated learning and distributed semi-supervised learning and provide a more comprehensive discussion of related work in App. C.

Privacy in Federated Learning: Collaboratively training a model without sharing sensitive data is a key advantage of (horizontal) federated learning (McMahan et al. 2017) which trains local models and aggregates their parameters periodically. Communicating only model parameters, however, does not entirely protect local data: An attacker can make inferences about local data from model parameters (Shokri et al. 2017; Ma et al. 2020) and model updates (Zhu and Han 2020). A common defense is perturbing shared information, e.g., applying appropriate clipping and noise before sending model parameters. This can guarantee ϵ, δ-differential privacy for local data (Wei et al. 2020) at the cost of model quality. This technique also defends against backdoor and poisoning attacks (Sun et al. 2019). Truex et al. (2019) proposes enhancing the privacy of data exchange in traditional distributed algorithms through the use of secure multi-party communication (SMPC) and differential privacy (DP) in applications where the scalability and efficiency limitations of SMPC are irrelevant. In these cases, SMPC (as well as homomorphic encryption (Roth et al. 2019) and trusted execution environments (Subramanyan et al. 2017)), offers an additional level of privacy that can be combined with other approaches (cf. Sec. 4 in Kairouz et al. 2021).

Distributed semi-supervised learning: Semi-supervised learning utilizes both labeled and unlabeled data (Zhou and Li 2005; Rasmus et al. 2015) for training. In centralized co-training, classifiers are independently trained on distinct feature sets, or views, of labeled data and their consensus on unlabeled data is used as pseudo-labels (Blum and Mitchell 1998; Ullrich et al. 2017). Papernot et al. (2017) propose a distributed—but not collaborative—knowledge distillation approach called PATE where teachers are trained distributedly and a consensus of their predictions on the unlabeled data is used to train a student model. We show empirically that PATE and its differentially private variant DP-PATE is outperformed by collaborative approaches. Bistritz, Mann, and Bambos (2020) propose to share soft predictions on unlabeled data to reduce communication in federated (and decentralized) deep learning and term their approach distributed distillation (DD). We compare to DD, showing that we achieve similar model quality and communication with improved privacy. Chen and Chao (2020) employ knowledge distillation to train a student model based on predictions from a Bayesian model ensemble (FedBE). Similarly, (Lin et al. 2020)'s FedDF also uses knowledge distillation in a federated context to create a global model by fusing client models. Both FedBE and FedDF require sharing local model parameters and thus have the same privacy issues that FEDAVG has.

3 Federated Semi-Supervised Learning

Preliminaries: We assume learning algorithms $\mathcal{A} : \mathcal{X} \times \mathcal{Y} \to \mathcal{H}$ that produce models $h \in \mathcal{H}$ using a dataset $D \subset \mathcal{X} \times \mathcal{Y}$ from an input space $\mathcal{X}$ and output space $\mathcal{Y}$, i.e., $h_{t+1} = \mathcal{A}(D)$, or iterative learning algorithms (cf. Chp. 2.1.4 Kamp 2019) $\mathcal{A} : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathcal{H}$ that update a model $h_{t+1} = \mathcal{A}(D, h_t)$. Given a set of $m \in \mathbb{N}$ clients with local datasets $D^1, \ldots, D^m \subset \mathcal{X} \times \mathcal{Y}$ drawn iid from a data

distribution $\mathcal{D}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, the goal is to find a set of local models $h^{1*}, \ldots, h^{m*} \in \mathcal{H}$ that each minimize the risk $\varepsilon(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]$.

In *centralized learning*, datasets are pooled and $\mathcal{A}$ is applied to $D = \bigcup_{i\in[m]} D^i$ until convergence, e.g., as full or mini-batch training. Convergence is measured in terms of low training loss, small gradient, or small deviation from previous iterations. In standard *federated learning* (McMahan et al. 2017), $\mathcal{A}$ is applied in parallel for $b \in \mathbb{N}$ rounds on each client to produce local models $h^1, \ldots, h^m$ which are centralized and aggregated using an aggregation operator $\text{agg} : \mathcal{H}^m \to \mathcal{H}$, i.e., $\overline{h} = \text{agg}(h^1, \ldots, h^m)$. The aggregated model $\overline{h}$ is redistributed to local clients which perform another b rounds of training using $\overline{h}$ as a starting point. This is iterated until convergence of $\overline{h}$. When aggregating by averaging, this method is known as federated averaging (FEDAVG).

In federated semi-supervised learning, a public unlabeled dataset U is available to all clients. Clients can share predictions on U (both hard and soft labels), as well as model parameters. Since sharing model parameters threatens privacy, we consider semi-supervised approaches that share predictions (other approaches are discussed in App. C).

Algorithm 1: Federated Co-Training (FEDCT)

Input: communication period b, m clients with local datasets $D^1, \ldots, D^m$ and local learning algorithms algorithms $\mathcal{A}^1, \ldots \mathcal{A}^m$, unlabeled shared dataset U, total number of rounds T
Output: final models $h_T^1, \ldots, h_T^m$

1 initialize local models $h_0^1, \ldots, h_0^m$, $P \leftarrow \emptyset$
2 **Locally** *at client i at time t* **do**
3 $h_t^i \leftarrow \mathcal{A}^i(D^i \cup P, h_{t-1}^i)$
4 **if** $t \,\%\, b = b - 1$ **then**
5 $L_t^i \leftarrow h_t^i(U)$
6 send L_t^i to server and receive L_t
7 $P \leftarrow (U, L_t)$
8 **end**
9 **At server** *at time t* **do**
10 receive local pseudo-labels $L_t^1, \ldots, L_t^m$
11 $L_t \leftarrow \text{consensus}(L_t^1, \ldots, L_t^m)$
12 send L_t to all clients

A Federated Co-Training Approach: We propose a federated variant of co-training—originally developed for multi-view semi-supervised learning (Blum and Mitchell 1998)—that iteratively updates pseudo-labels of U via the consensus of shared predictions. That is, in a communication round $t \in \mathbb{N}$ each client $i \in [m]$ shares local labels $L_t^i = h_t^i(U)$ (not soft predictions) on U with the server, which produces a consensus labeling $L_t \subset \mathcal{Y}$ via an appropriate consensus mechanism. The resulting pseudo-labeled dataset $P = (U, L_t)$ augments local training sets. We call this approach federated co-training (FEDCT). Sharing hard labels not only improves privacy, but also allows us to use any supervised learning method for local training.

We describe federated co-training in Algorithm 1: at each client i, the local model is updated using local dataset D^i combined with the current pseudo-labeled public dataset P (line 4). In a communication round (line 5), the updated model is used to produce improved pseudo-labels L^i for the unlabeled data U (line 6), which are sent to a server (line 7). At the server, when all local prediction $L^1, \ldots, L^m$ are received (line 12), a consensus L is formed (line 13) and broadcasted to the clients (14). At the client, upon receiving the consensus labels (line 8), the pseudo-labeled dataset is updated (line 9), and another iteration of local training is performed. For classification problems the majority vote is a reliable consensus mechanism (Papernot et al. 2017).

Convergence Analysis: The convergence of federated co-training depends on the convergence of the local learning algorithms $\left(\mathcal{A}^i\right)_{i\in[m]}$. Under the assumption that these algorithms converge on a fixed training set, it remains to show that the training set eventually stabilizes. That is, there exists a round $t_0 \in \mathbb{N}$ such that for all $t > t_0$ it holds that $L_t = L_{t-1}$. For classification problems, this depends on the local training accuracy. If local training accuracy reaches $a_t = 1.0$, then the approach trivially converges, since local models will reproduce L_t in every subsequent round. This assumption can usually be fulfilled for over-parameterized models. In the following, we show that the approach also converges with high probability, if the training accuracy is ≤ 1, but increasing with t. This is typically the case in federated learning (Zhao et al. 2018) and federated semi-supervised learning (Papernot et al. 2017). To simplify the analysis, we approximate this via a linearly increasing training accuracy.

Proposition 1. *For $m \geq 3$ clients with local datasets $D^1, \ldots, D^m$ and unlabeled dataset U, let $\mathcal{A}^i$ for $i \in [m]$ be a set of learning algorithms that all achieve a linearly increasing training accuracy a_t for all labelings of U, i.e., there exists $c \in \mathbb{R}_+$ such that $a_t \geq 1 - c/t$, then there exists $t_0 \in \mathbb{N}$ such that $a_t \geq 1/2$ and* FEDCT *with majority vote converges with probability $1 - \delta$, where $\delta \leq |U|(4c)^{\frac{m}{2}}\zeta\left(\frac{m}{2}, t_0 + 1\right)$ and $\zeta(x, q)$ is the Hurwitz zeta function.*

Proof. (sketch:) If local models are of sufficient quality, then in round $t \geq t_0$, the probability that the consensus labels change, δ_t, is bounded. The probability can be determined via the CDF of the binomial distribution, which can be bounded via the Chernoff bound: $\delta_t \leq |U|4^{\frac{m}{2}} a_t^{\frac{m}{2}} (1 - a_t)^{\frac{m}{2}}$. Then the probability that the consensus labels remain constant for the remainder, i.e., the sum of δ_t from t_0 to ∞, is bounded as well. Using the assumption that a_t grows linearly, we can express this infinite series as $\sum_{t=t_0}^{\infty} \delta_t \lesssim \sum_{t=0}^{\infty} \frac{1}{t}^{\frac{m}{2}} - \sum_{t=0}^{t_0} \frac{1}{t}^{\frac{m}{2}}$, that is, the difference of the Riemann zeta function and the t_0-th generalized harmonic number, $\sum_{t=t_0}^{\infty} \delta_t \lesssim \zeta(m/2) - H_{t_0}^{m/2}$. This difference can be expressed via the Hurwitz zeta function $\zeta(m/2, t_0 + 1)$. □

The full proof is provided in App. A in the extended version

Communication Complexity: The communication complexity of FEDCT is in the same order as standard federated learning, i.e., treating the message size as a constant, the

communication complexity is in $\mathcal{O}(T/b)$, where b is the communication period. The actual number of bits transferred in each round, however, depends on the size of U: Encoding predictions as binary vectors, for a classification problem with $C \in \mathbb{N}$ classes the communication complexity is in $\mathcal{O}(TC|U|/b)$. As Bistritz, Mann, and Bambos (2020) observed, transferring predictions on U can reduce communication substantially over transferring the weights of large neural networks. For example on FashionMINST with $|U| = 10^4$ and a a neural network with 669 706 parameters, FEDCT and FEDAVG both achieve a test accuracy of 0.82, resp. 0.83 (cf. Tab. 1), but FEDCT transmits only $\approx 12.2KB$ bits in each round, whereas FEDAVG transmits $\approx 2.6MB$. Thus, FEDCT reduces communication by a factor of ≈ 214.

4 Differential Privacy for FEDCT

We assume the following attack model: clients are honest and the server is honest-but-curious (or semi-honest, i.e., it follows the protocol execution correctly, but may try to infer sensitive information about clients). The attack goal is to infer sensitive information about local training data from shared information. This assumption is stronger than an attacker intercepting individual communication, or an honest-but-curious client, since the server receives shared information from all clients. We also assume that parties do not collude. Details are deferred to App. E.1. Sharing hard labels on an unlabeled dataset empirically improves privacy substantially[1], as we show in Sec. 5. An empirical improvement in privacy is, however, no guarantee. Differential privacy instead provides a fundamental guarantee of privacy which is achieved through randomization of shared information: A randomized mechansim $\mathcal{M}$ with domain $\mathcal{X}$ and range $\mathcal{Y}$ is ϵ-differential private if for any two neighboring inputs $D, D^{'} \subset \mathcal{X}$ and for a substet of outputs $S \in \mathcal{Y}$ it holds that $P\left(\mathcal{M}(D) \in S\right) \leq \exp(\epsilon) P\left(\mathcal{M}(D^{'}) \in S\right)$ (Dwork, Roth et al. 2014). To obtain differential privacy (DP), the randomization has to be suitable to the information that is shared. In FEDCT local clients share hard labels, i.e., categorical values in case of classification. Standard DP mechanisms, like the Gaussian (Dwork, Roth et al. 2014) or Laplacian mechanissm (Dwork et al. 2006) are not suitable for categorical data. Therefore, we interpret labels as binary vectors via one-hot encoding. That is, for an unlabeled dataset U and a classification problem with $C \in \mathbb{N}$ classes, the predictions sent by a client with local dataset $D \subset \mathcal{X}$ to the server can be interpreted as the binary matrix output of a deterministic mechanism $f(D) \in \{0,1\}^{|U| \times C}$. Then, any DP-mechanism for binary data can be used, such as the XOR mechanism (Ji et al. 2021), or randomized graph-coloring (D'Oliveira, Médard, and Sadeghi 2021)[2].

To compute an actual DP-guarantee for a given DP-mechanism requires a bound on the sensitivity of the underlying deterministic function f. That is, given two neighboring datasets D, D' (i.e., they differ only in a single element), the sensitivity of f is defined as $s_f = \sup_{f(D), f(D')} \|f(D) \oplus f(D')\|_F^2$, where $\oplus$ denotes binary XOR. For FEDCT this means providing a bound on how much the predictions of a client on the unlabeled dataset can change if one element of its local training set is removed. In the following, we bound the sensitivity of any learning algorithm that is on-average-replace-one-stable with monotonically increasing stability rate $\epsilon : \mathbb{N} \to \mathbb{R}$ for a learning algorithm $\mathcal{A}$ and loss function ℓ (cf. Shalev-Shwartz and Ben-David (2014)).

Proposition 2. *For classification models $h : \mathcal{X} \to \mathcal{Y}$, let ℓ be a loss function that upper bounds the $0-1$-loss and $\mathcal{A}$ a learning algorithm that is on-average-replace-one stable with stability rate $\epsilon(n)$ for ℓ. Let $D \cup U$ be a local training set with $|U| = n$, and $\delta \in (0,1)$. Then with probability $1-\delta$, the sensitivity s_* of $\mathcal{A}$ on U is bounded by*

$$s_* \leq \left\lceil n\epsilon(n) + P\sqrt{n\epsilon(n)(1-\epsilon(n))} + \frac{P^2}{3} \right\rceil \ ,$$

where $P = \Phi^{-1}(1-\delta)$ and Φ^{-1} is the probit function.

The proof is provided in App. B. We now provide a DP analysis using the XOR-mechanism (Ji et al. 2021). For that, let $\mathcal{B} \in \{0,1\}^{N \times P}$ denote a matrix-valued Bernoulli random variable, i.e., $\mathcal{B} \sim \mathrm{Ber}_{N,P}(\mathbf{\Theta}, \mathbf{\Lambda}_{1,2}, \cdots, \mathbf{\Lambda}_{N-1,N})$ with a matrix-valued Bernoulli distribution with quadratic exponential dependence structure. Here, $\mathbf{\Theta}$ is the $P \times P$ association parametric matrix including log-linear parameters describing the association structure of the columns, and $\mathbf{\Lambda}_{i,j}$ is the $P \times P$ association parametric matrix of rows i and j. The XOR-mechanism applies this random matrix to the output of the deterministic mechanism via the XOR operator $\oplus$ and yields a randomized mechanism $\mathcal{M}(D) = f(D) \oplus \mathcal{B}$. We represent local predictions of clients L_t^i as binary matrices and apply the XOR-mechanism, i.e., $\widehat{L}_t^i = L_t^i \oplus \mathcal{B}$. These randomized predictions are then send to the server, resulting in differentially private federated co-training (DP-FEDCT): Defining the sensitivity of DP-FEDCT as $s_* = \max\{s_{f^1}, \ldots, s_{f^m}\}$, it follows directly from Theorem 1 in Ji et al. (2021) that DP-FEDCT achieves ϵ-differential privacy.

Corollary 1. *Applying the XOR mechanism to* FEDCT *with sensitivity s_* achieves ϵ-DP if $\mathbf{\Theta}$ and $\mathbf{\Lambda}_{i,j}$ satisfy*

$$s_*(||\boldsymbol{\lambda}(\mathbf{\Theta})||_2 + \textstyle\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} ||\boldsymbol{\lambda}(\mathbf{\Lambda}_{i,j})||_2\Big) \leq \epsilon, \quad (1)$$

where $||\boldsymbol{\lambda}(\mathbf{\Theta})||_2$ and $||\boldsymbol{\lambda}(\mathbf{\Lambda}_{i,j})||_2$ are the l_2 norms of the eigenvalues of $\mathbf{\Theta}$ and $\mathbf{\Lambda}_{i,j}$.

On-average-replace-one-stability holds for many supervised learning methods. For example, every regularized risk minimizer for a convex, Lipschitz loss using a strongly convex regularizer, like Thikonov-regularization, is on-average-replace-one-stable (cf. Chp. 13.3 in Shalev-Shwartz and Ben-David 2014), as well as deep learning with SGD (Hardt, Recht, and Singer 2016), including cases involving non-smooth loss functions (Bassily et al. 2020). We empirically evaluate the privacy-utility trade-off of FEDCT with differential privacy in Sec. 5.

[1] This differs from label leakage (Li and Zhang 2021), where predictions on the private data are shared.

[2] Label differential privacy (Ghazi et al. 2021) is applicable, but require a different sensitivity analysis.

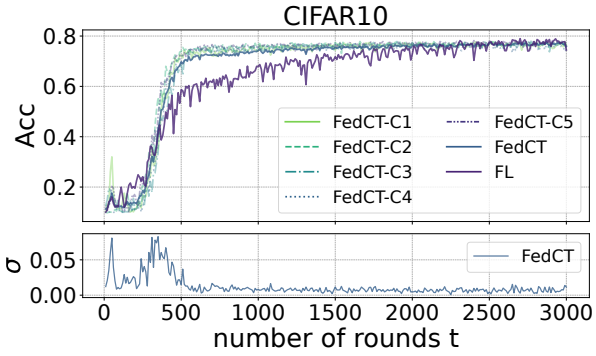

Figure 2: **Top**: Test ac. (ACC) over time on CIFAR10 of FL, and FEDCT's local models and their mean. **Bottom**: Standard deviation of test accuracy of local models in FEDCT.

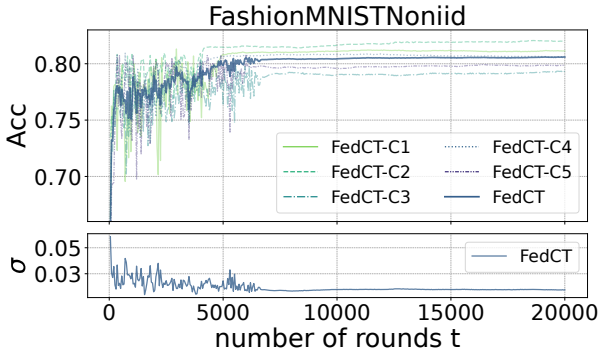

Figure 3: **Top**: Test acc- (ACC) over time of FEDCT on non-iid distribution on FashionMNIST. **Bottom**: Standard deviation of test accuracy of local models in FEDCT.

5 Empirical Evaluation

We empirically show that FEDCT presents a favorable privacy-utility trade-off compared to other federated learning approaches by showing that it achieves similar test accuracy with substantially improved privacy. We compare FEDCT to federated averaging (McMahan et al. 2017) (FEDAVG), differentially private federated averaging (DP-FEDAVG) achieved via the Gaussian mechanism (Geyer, Klein, and Nabi 2017), and distributed distillation (Bistritz, Mann, and Bambos 2020) (DD) on 3 benchmark datasets and 2 medical image classification datasets, as well as on a fine-tuning task for large language models. We also compare with PATE (Papernot et al. 2017), although it is not collaborative, because it shares hard labels (see App. D.3 for details).

Experimental Setup We use three benchmark image classification datasets, CIFAR10 (Krizhevsky, Nair, and Hinton 2010), FashionMNIST (Xiao, Rasul, and Vollgraf 2017), and SVHN (Netzer et al. 2011), as well as two real medical image classification datasets, MRI scans for brain tumor detection (Chakrabarty 2018), and chest X-rays for pneumonia detection (Kermany et al. 2018). We evaluate FEDCT with interpretable models on five benchmark datasets, WineQuality (Cortez et al. 2009), Breastcancer (Street, Wolberg, and Mangasarian 1993), AdultsIncome (Becker and Kohavi 1996), Mushroom (Bache and Lichman 1987), and Covertype (Blackard 1998). We fine-tune an LLM on the IMDB dataset (Maas et al. 2011) and the Twitter dataset (kaggle). We first divide each dataset into a test and training set and further divide the training set into an unlabeled dataset U and a set of m local training sets (sampling iid. for all experiments, except for the experiments on heterogeneous data distributions). We also investigated how the distribution and size of unlabeled datasets affect performance, as demonstrated in experiments App.5 and App.D.7. The architectures of the neural networks are provided in App. E. The hyperparameters are optimized individually for all methods on a subset of the training set via cross-validation. We select the number of rounds to be the maximum rounds required so that all methods converge, i.e., $T = 2 * 10^4$. We measure empirical privacy vulnerability by performing a large number of membership inference attacks and compute the probability of inferring upon sensitive data, using the ML Privacy Meter tool (Murakonda and Shokri 2020). The **vulnerability (VUL)** of a method is the ROC AUC of membership attacks over K runs over the entire training set. A vulnerability of 1.0 means that membership can be inferred with certainty, whereas 0.5 means that deciding on membership is a random guess. While our privacy evaluation reflects the relative performance of the different methods, it can be inaccurate in assessing the actual privacy risk, particularly for the most vulnerable data points (Aerni, Zhang, and Tramèr 2024). More details on the follwoing experiments, additional experiments on using mixed model types, a comparison with FEDMD (Li and Wang 2019), and an abblation study can be found in App. D

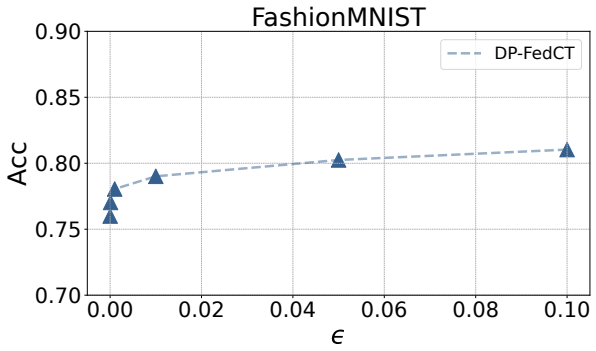

Figure 4: Accuracy (ACC) of DP-FEDCT on the FashionMNIST dataset under different levels of privacy ϵ.

Privacy-Utility-Trade-Off: We first evaluate the performance of FEDCT and baselines for deep learning on homogeneous data distributions. We use an unlabeled dataset of size $|U| = 10^4$ for CIFAR10, $|U| = 5 \cdot 10^4$ for FashionMNIST, $|U| = 170$ for MRI, $|U| = 900$ for Pneumonia, and $|U| = 35 \cdot 10^4$ for SVHM. Note that only FEDCT, DP-FEDCT, PATE, and DD use the unlabeled dataset. The remaining training data is distributed over $m = 5$ clients. We repeat all experiments 3 times and report average test

Method	CIFAR10		FashionMNIST		Pneumonia		MRI		SVHN	
	ACC	VUL	ACC	VUL	ACC	VUL	ACC	VUL	ACC	VUL
FEDCT	$\mathbf{0.77} \pm 0.003$	0.52	$\mathbf{0.84} \pm 0.004$	**0.51**	$\mathbf{0.78} \pm 0.008$	**0.51**	0.64 ± 0.004	0.52	$\mathbf{0.91} \pm 0.002$	**0.53**
DP-FEDCT ($\epsilon = 0.1$)	0.76 ± 0.002	**0.51**	0.80 ± 0.001	0.52	0.75 ± 0.004	**0.51**	0.62 ± 0.002	**0.51**	0.86 ± 0.001	**0.53**
FEDAVG	$\mathbf{0.77} \pm 0.020$	0.73	0.83 ± 0.024	0.72	0.74 ± 0.013	0.76	0.66 ± 0.015	0.73	$\mathbf{0.91} \pm 0.026$	0.71
DP-FEDAVG ($\epsilon = 0.1$)	0.68 ± 0.002	0.70	0.69 ± 0.002	0.71	0.61 ± 0.004	0.69	0.56 ± 0.003	0.62	0.71 ± 0.005	0.70
DD	0.70 ± 0.012	0.61	0.82 ± 0.016	0.60	$\mathbf{0.78} \pm 0.003$	0.63	$\mathbf{0.68} \pm 0.008$	0.60	0.73 ± 0.014	0.59
PATE	0.69 ± 0.002	0.60	0.73 ± 0.001	0.59	0.75 ± 0.003	0.59	0.61 ± 0.001	0.60	0.87 ± 0.002	0.58
DP-PATE	0.67 ± 0.003	0.58	0.73 ± 0.002	0.57	0.71 ± 0.001	0.58	0.60 ± 0.001	0.57	0.86 ± 0.002	0.57

Table 1: Test acc. (ACC) and privacy vulnerability (VUL, smaller is better) for $m = 5$ clients on iid data.

	$\alpha_1 = 100$ $\alpha_2 = 2$	$\alpha_1 = 100$ $\alpha_2 = 0.01$	$\alpha_1 = 0.01$ $\alpha_2 = 0.01$
FEDCT	0.7975	0.7905	0.6252
FEDCT (QM)	0.7968	0.7808	0.7358
FEDAVG	0.8150	0.7950	0.7346
DD	0.7684	0.7219	0.5580

Table 2: Non-iid data distributions with $m = 10$ clients.

accuracy and maximum deviation (see App. E for details).

The results presented in Tab. 1 show that FEDCT achieves a test accuracy comparable to both FEDAVG, and DD, while preserving privacy to the highest level. That is, FEDCT performs best on CIFAR10, has a similar performance to both on FashionMNIST, Pneumonia, and SVHN, and is slightly worse on MRI. All collaborative approaches outperform PATE. The vulnerability is around 0.5, so that membership inference attacks are akin to random guessing. FEDAVG instead consistently has a vulnerability over 0.7. DP-FEDAVG improves privacy, but reduces the test accuracy substantially. Our experiments show that DD substantially improves privacy over both FEDAVG and DP-FEDAVG, yet still is vulnerable ($VUL \approx 0.6$). Since FEDCT does not produce a global model, we investigate the convergence behavior of individual client models in terms of test accuracy on CIFAR10 in Fig. 2. From the standard deviation between clients σ we see that they converge to a consensus after around 700 rounds with only slight deviations afterwards. Overall, FEDCT converges slightly faster than FEDAVG, though the latter increases its test accuracy slightly further, eventually.

Privacy-Utility Trade-Off With Differential Privacy: Differential privacy guarantees typically come at a cost in terms of utility, which in our case means a loss in model quality. Analyzing this privacy-utility trade-off requires estimating the sensitivity. Since stability-bounds for neural networks tend to underestimate the on-average-replace-one stability, leading to vacuous results for generalization (Nagarajan and Kolter 2019; Petzka et al. 2021), using them to bound sensitivity would underestimate utility. Using an empirical approximation provides a more accurate estimate for the privacy-utility trade-off (Rubinstein and Aldà 2017). To get this approximation, we apply FEDCT with $m = 5$ clients on the FashionMNIST dataset (Xiao, Rasul, and Vollgraf 2017) for various privacy levels ϵ. We estimate the sensitivity of DP-FEDCT by sampling $n = 100$ datasets $D'_1, \ldots, D'_n$ neighboring a local training set D to approximate $s_* \approx \max_{i \in [n]} \|f(D) \oplus f(D'_i)\|_F^2$, which yields $s_* \approx 3000$. Using this estimate, Fig. 4 shows that DP-FEDCT achieves a high utility in terms of test accuracy even for moderate-to-high privacy levels ϵ with an accuracy of 0.8 for $\epsilon = 0.1$[3] (without any noise, FEDCT achieves an accuracy of 0.82 in this setup). As Papernot et al. (2017) observed, the reason for the good trade-off probably lies in the consensus mechanism: for a single unlabeled example $\mu > m/C$ clients predict the majority class, so the XOR-mechanism has to change the predictions of at least $\mu - m/C$ many clients to change its consensus label.

Heterogeneous Data Distributions: In many realistic applications, local datasets are not heterogeneous. We show that FEDCT performs similar to FEDAVG for non-pathological non-iid data distributions, but FEDCT with majority voting, as well as soft label sharing, are outperformed on pathological non-iid distributions. This is remedied by using a qualified majority vote as consensus. For a non-pathological non-iid data distribution, we sample half of the training data from a Dirichlet distribution over labels with $\alpha_1 = 100$ (mild heterogeneity) and the other half with $\alpha_2 = 2$ (medium heterogeneity) and $\alpha_2 = 0.01$ (strong heterogeneity). For both cases, we see that FEDCT, DD, and FEDAVG perform similarly (see Tab. 2). In Fig. 3 we show the convergence behavior of individual clients for $\alpha_1 = 100, \alpha_2 = 2$ which is similar to the iid case, but with higher variance between individual client models. For the pathological case ($\alpha_1 = \alpha_2 = 0.01$) FEDAVG still performs well, outperforming both FEDCT and DD. We conjecture that a meaningful consensus requires clients to achieve a minimum performance for all labels. In the pathological case, clients observe only a small subset of labels and thus perform poorly on a majority of data. Using a qualified majority (FEDCT (QM)) with a quorum of 0.9 improves the performance of FEDCT to the level of FEDAVG, showing that FEDCT using QM also performs well in the pathological case. Further details are deferred to App. D.6.

Effect of Unlabeled Dataset Distribution In our main experiments in Tab.1, the unlabeled set U is drawn iid from the respective datasets. In practice, however, publicly available datasets will be similar, but not equal to a private dataset's distribution. We therefore explore what the impact of the distribution of the unlabeled dataset is on the performance of FEDCT. We investigate this using iid samples from the CIFAR10 dataset as private data. We then test FEDCT on various unlabeled distribution, ranging from similar to private

[3]Note that using the trivial upper bound of $s_*^W = |U| = 5 \cdot 10^4$ instead of our estimate results in a slightly higher epsilon: for a noise level that achieves $\epsilon = 0.1$ with the empirical estimate of s_*, the worst-case bound results in $\epsilon = 0.1 \cdot s_*^W / s_* = 5/3$, instead.

Dataset	DT		RuleFit		XGBoost		Random Forest	
	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED	FEDCT	CENTRALIZED
WineQuality	0.95 ± 0.01	0.92	0.93 ± 0.01	0.95	0.94 ± 0.01	0.94	0.96 ± 0.01	0.98
BreastCancer	0.89 ± 0.01	0.89	0.92 ± 0.01	0.93	0.93 ± 0.01	0.94	0.90 ± 0.02	0.93
AdultsIncome	0.81 ± 0.01	0.82	0.84 ± 0.02	0.85	0.85 ± 0.02	0.87	0.85 ± 0.01	0.86
Mushroom	0.98 ± 0.01	1	0.98 ± 0.02	1	0.98 ± 0.01	1	0.99 ± 0.01	1
Covertype	0.88 ± 0.02	0.94	0.73 ± 0.02	0.76	0.84 ± 0.02	0.87	0.90 ± 0.01	0.95

Table 3: Test accuracy (ACC) of FEDCT with interpretable models.

data to very dissimilar. As very similar unlabeled dataset, we use a non-iid sample of CIFAR10 sampling using a Dirichlet distribution over labels with $\alpha = 0.5$. As dissimilar unlabeled dataset, we use an iid sample of CIFAR100—the distribution is highly dissimilar, since not a single class of CIFAR10 is present in CIFAR100. As a middle ground, we have chosen classes from CIFAR100 that are semantically more similar to the CIFAR10 classes (see Tab. 9). In all cases we have chosen $|U| = 8500$. The results shown in Tab. 4 show that while the test accuracy of FEDCT decreases the more dissimilar the unlabaled data distribution is from private data, it remains high even for very dissimilar distributions.

Distribution of $\lvert U \rvert$	**FEDCT**
iid CIFAR10	0.77
non-iid CIFAR10 ($\alpha = 1.0$)	0.74
non-iid CIFAR10 ($\alpha = 0.5$)	0.72
similar CIFAR100 classes	0.71
iid sample of CIFAR100	0.65

Table 4: Test accuracy (ACC) of FEDCT on CIFAR10 for three different scenarios of using a public dataset with different data distribution.

Interpretable Models: FEDCT allows training models that cannot be aggregated in FEDAVG and cannot be trained via soft label sharing (e.g., as in DD). Many interpretable models, such as decision trees (Quinlan 1986), XGBoost (Chen and Guestrin 2016), Random Forest (Breiman 2001), and RuleFit (Friedman and Popescu 2008) fall under this class. We evaluate FEDCT on such models on 5 benchmark datasets with $m = 5$ clients and compare its performance to pooling all data and training a model centrally (Centralized). The results in Tab. 3 show that FEDCT can train interpretable models in a federated learning setup, achieving a model quality comparable to centralized training. In App. D.1 we show that FEDCT can also train mixed models, i.e., each client training a different model type, to high accuracy.

Fine-Tuning Large Language Models: In fine-tuning, model quality is already high from the start so that pseudo-labels are likely of high quality from the start. If true, semi-supervised approaches should improve performance over FEDAVG. To test this hypothesis, we fine-tune the GPT2 model transformer with a sequence classification head (linear layer) comprising of 124.44 million parameters on the IMBD sentimental dataset (Maas et al. 2011) and the Twitter sentiment dataset (kaggle), using $m = 10$ clients with $|U| = 150$ for IMBD and $|U| = 35,000$ for Twitter. Indeed, we observe that on IMDB, FEDCT achieves a test accuracy of 0.73, whereas FEDAVG achieves an ACC of 0.59 and on Twitter FEDCT achieves a test accuracy of 0.65, whereas FEDAVG achieves an ACC of 0.61 (see App. D.2 for details).

6 Discussion and Conclusion

We propose a federated semi-supervised co-training approach that collaboratively trains models via sharing predictions on an unlabeled dataset U. In many applications, such unlabeled datasets are available, e.g. in healthcare[4], or can be synthetically generated (El Emam, Mosquera, and Hoptroff 2020). A limitation of FEDCT is that it does not produce a global model. Instead, it promotes agreement between local models. Our experiments, however, show that local models quickly converge to similar test accuracy so that each local model could act as global model. At the same time, FEDCT allows us to use different models that can be tailored to each client (see App. D.1). A second limitation, revealed by our experiments, is that on pathological non-iid data, where clients only observe a small subset of labels, soft label sharing and hard label sharing with majority voting are outperformed by FEDAVG. Using a qualified majority vote as consensus remedies this issue. Similar to federated learning variants tailored to heterogeneous data, such as FedProx (Li et al. 2020) and SCAFFOLD (Karimireddy et al. 2020), it would make for excellent future work to further improve FEDCT's performance in this case, e.g., by using more elaborate consensus mechanisms (Warfield, Zou, and Wells 2004). Furthermore, investigating client subsampling in FEDCT and its impact on the consensus mechanism, other communication-efficient strategies (e.g., Kamp et al. 2016, 2019), and learning from small datasets (Kamp, Fischer, and Vreeken 2023) is interesting. The results on fine-tuning LLMs are promising and suggest that semi-supervised learning can be particularly beneficial in federated fine-tuning of foundation models, which will be interesting to further investigate in the future.

We show that FEDCT matches the model quality of FEDAVG and DD while significantly improving privacy over both FEDAVG and DD, as well as DP-FEDAVG. From this we conclude that sharing little is enough: sharing hard labels improves privacy substantially while maintaining a favorable privacy-utility trade-off, in particular for fine-tuning LLMs. Moreover, FEDCT allows us to train interpretable models, such as rule ensembles and XGBoost, in a federated learning setup. The proposed approach facilitates the deployment of machine learning in critical domains such as healthcare, where highly sensitive private datasets are distributed across sites, and public unlabeled datasets are often available.

[4]Examples for large public health databases are the US NCHS DB, UK NHS DB, UK Biobank (Sudlow et al. 2015), MIMIC-III database (Johnson, Pollard, and Mark 2016), TCGA public dataset (NIH 2011), and EU EHDS.

Acknowledgements

Amr Abourayya, Jens Kleesiek, and Michael Kamp received support from the Cancer Research Center Cologne Essen (CCCE). Erman Ayday was partly supported by the National Science Foundation (NSF) under grant numbers 2141622, 2427505, and OAC-2112606.

References

Aerni, M.; Zhang, J.; and Tramèr, F. 2024. Evaluations of Machine Learning Privacy Defenses are Misleading. *ACM CCS*, 1271–1284.

Bache, K.; and Lichman, M. 1987. Mushroom. UCI Machine Learning Repository.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *AISTATS, PMLR*, 2938–2948.

Bassily, R.; Feldman, V.; Guzmán, C.; and Talwar, K. 2020. Stability of stochastic gradient descent on nonsmooth convex losses. *NeurIPS*, 33: 4381–4391.

Becker, B.; and Kohavi, R. 1996. Adult. UCI Machine Learning Repository.

Bistritz, I.; Mann, A.; and Bambos, N. 2020. Distributed distillation for on-device learning. *NeurIPS*, 33: 22593–22604.

Blackard, J. 1998. Covertype. UCI Machine Learning Repository.

Blum, A.; and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *COLT*, 92–100.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Chakrabarty, N. 2018. Brain Tumor Detection, Kaggle.

Chaudhuri, K.; Imola, J.; and Machanavajjhala, A. 2019. Capacity bounded differential privacy. In *NeurIPS*, volume 32.

Chen, H.-Y.; and Chao, W.-L. 2020. FedBE: Making Bayesian Model Ensemble Applicable to Federated Learning. In *ICLR*.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *KDD*, 785–794.

Cortez, P.; Cerdeira, A.; Almeida, F.; Matos, T.; and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4): 547–553.

D'Oliveira, R. G.; Médard, M.; and Sadeghi, P. 2021. Differential privacy for binary functions via randomized graph colorings. In *IEEE ISIT*, 473–478. IEEE.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *TCC*, 265–284. Springer.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4): 211–407.

El Emam, K.; Mosquera, L.; and Hoptroff, R. 2020. *Practical synthetic data generation: balancing privacy and the broad availability of data*. O'Reilly Media.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Friedman, J. H.; and Popescu, B. E. 2008. Predictive learning via rule ensembles. *The annals of applied statistics*, 916–954.

Geyer, R. C.; Klein, T.; and Nabi, M. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Ghazi, B.; Golowich, N.; Kumar, R.; Manurangsi, P.; and Zhang, C. 2021. Deep learning with label differential privacy. *NeurIPS*, 34: 27131–27145.

Gong, X.; Sharma, A.; Karanam, S.; Wu, Z.; Chen, T.; Doermann, D.; and Innanje, A. 2022. Preserving privacy in federated learning with ensemble cross-domain knowledge distillation. In *AAAI*, volume 36, 11891–11899.

Hardt, M.; Recht, B.; and Singer, Y. 2016. Train faster, generalize better: Stability of stochastic gradient descent. In *ICML*, 1225–1234. PMLR.

Ji, T.; Li, P.; Yilmaz, E.; Ayday, E.; Ye, Y.; and Sun, J. 2021. Differentially private binary-and matrix-valued data query: an XOR mechanism. *VLDB*, 14(5): 849–862.

Jiang, D.; Shan, C.; and Zhang, Z. 2020. Federated learning algorithm based on knowledge distillation. In *ICAICE*, 163–167. IEEE.

Johnson, A.; Pollard, T.; and Mark, R. 2016. MIMIC-III clinical database (version 1.4). *PhysioNet*, 10: C2XW26.

kaggle. 2021. Twitter Sentiment Analysis.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Kamp, M. 2019. *Black-Box Parallelization for Machine Learning*. Ph.D. thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Universitäts-und Landesbibliothek Bonn.

Kamp, M.; Adilova, L.; Sicking, J.; Hüger, F.; Schlicht, P.; Wirtz, T.; and Wrobel, S. 2019. Efficient decentralized deep learning by dynamic model averaging. In *ECMLPKDD*, 393–409. Springer.

Kamp, M.; Bothe, S.; Boley, M.; and Mock, M. 2016. Communication-efficient distributed online learning with kernels. In *ECMLPKDD*, 805–819. Springer.

Kamp, M.; Fischer, J.; and Vreeken, J. 2023. Federated Learning from Small Datasets. In *ICLR*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *ICML*, 5132–5143. PMLR.

Kermany, D. S.; Goldbaum, M.; Cai, W.; Valentim, C. C.; Liang, H.; Baxter, S. L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. 2018. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5): 1122–1131.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2010. Cifar-10 (canadian institute for advanced research).

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

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. In *Machine learning and systems*, volume 2, 429–450.

Li, Z.; and Zhang, Y. 2021. Membership leakage in label-only exposures. In *ACM CCS*, 880–895.

Lin, T.; Kong, L.; Stich, S. U.; and Jaggi, M. 2020. Ensemble distillation for robust model fusion in federated learning. *NeurIPS*, 33: 2351–2363.

Ma, C.; Li, J.; Ding, M.; Yang, H. H.; Shu, F.; Quek, T. Q.; and Poor, H. V. 2020. On safeguarding privacy and security in the framework of federated learning. *IEEE network*, 34(4): 242–248.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *ACL HLT*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Mcdonald, R.; Mohri, M.; Silberman, N.; Walker, D.; and Mann, G. 2009. Efficient large-scale distributed training of conditional maximum entropy models. *NeurIPS*, 22.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Murakonda, S. K.; and Shokri, R. 2020. Ml privacy meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. *arXiv preprint arXiv:2007.09339*.

Nagarajan, V.; and Kolter, J. Z. 2019. Uniform convergence may be unable to explain generalization in deep learning. *NeurIPS*, 32.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

NIH, N. C. I. 2011. The Cancer Genome Atlas Program (TCGA).

Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised knowledge transfer for deep learning from private training data. *ICLR*.

Petzka, H.; Kamp, M.; Adilova, L.; Sminchisescu, C.; and Boley, M. 2021. Relative flatness and generalization. *NeurIPS*, 34: 18420–18432.

Quinlan, J. R. 1986. Induction of decision trees. *Machine learning*, 1: 81–106.

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T. 2015. Semi-supervised learning with ladder networks. *NeurIPS*, 28.

Roth, E.; Noble, D.; Falk, B. H.; and Haeberlen, A. 2019. Honeycrisp: large-scale differentially private aggregation without a trusted core. In *ACM SOSP*, 196–210.

Rubinstein, B. I.; and Aldà, F. 2017. Pain-free random differential privacy with sensitivity sampling. In *ICML*, 2950–2959. PMLR.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Street, W. N.; Wolberg, W. H.; and Mangasarian, O. L. 1993. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, 861–870. SPIE.

Struppek, L.; Hintersdorf, D.; and Kersting, K. 2024. Be Careful What You Smooth For: Label Smoothing Can Be a Privacy Shield but Also a Catalyst for Model Inversion Attacks. *ICLR*.

Subramanyan, P.; Sinha, R.; Lebedev, I.; Devadas, S.; and Seshia, S. A. 2017. A formal foundation for secure remote execution of enclaves. In *ACM CCS*, 2435–2450.

Sudlow, C.; Gallacher, J.; Allen, N.; Beral, V.; Burton, P.; Danesh, J.; Downey, P.; Elliott, P.; Green, J.; Landray, M.; et al. 2015. UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3): e1001779.

Sun, Z.; Kairouz, P.; Suresh, A. T.; and McMahan, H. B. 2019. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*.

Truex, S.; Baracaldo, N.; Anwar, A.; Steinke, T.; Ludwig, H.; Zhang, R.; and Zhou, Y. 2019. A hybrid approach to privacy-preserving federated learning. In *ACM workshop on artificial intelligence and security*, 1–11.

Ullrich, K.; Kamp, M.; Gärtner, T.; Vogt, M.; and Wrobel, S. 2017. Co-regularised support vector regression. In *ECMLP-KDD*, 338–354. Springer.

Warfield, S. K.; Zou, K. H.; and Wells, W. M. 2004. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, 23(7): 903–921.

Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H. H.; Farokhi, F.; Jin, S.; Quek, T. Q.; and Poor, H. V. 2020. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15: 3454–3469.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:cs.LG/1708.07747.

Xiao, H.; Wan, J.; and Devadas, S. 2022. Differentially Private Deep Learning with ModelMix. *arXiv preprint arXiv:2210.03843*.

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

Zhou, Z.-H.; and Li, M. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11): 1529–1541.

Zhu, L.; and Han, S. 2020. Deep leakage from gradients. In *Federated learning*, 17–31. Springer.

Ziller, A.; Usynin, D.; Braren, R.; Makowski, M.; Rueckert, D.; and Kaissis, G. 2021. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1): 13524.