

# A Practical Influence Approximation for Privacy-Preserving Data Filtering in Federated Learning

Ljubomir Rokvic<sup>1</sup>, Panayiotis Danassis<sup>2</sup>, Boi Faltings<sup>1</sup>

<sup>1</sup> École Polytechnique Fédérale de Lausanne (EPFL)

<sup>2</sup>Harvard University

{firstname.lastname}@epfl.ch, pdanassis@seas.harvard.edu

## Abstract

Federated Learning by nature is susceptible to low-quality, corrupted, or even malicious data that can severely degrade the quality of the learned model. Traditional techniques for data valuation cannot be applied as the data is never revealed. We present a novel technique for filtering, and scoring data based on a *practical influence approximation* (‘lazy’ influence) that can be implemented in a *privacy-preserving* manner. Each participant uses his *own data* to evaluate the influence of another participant’s batch, and reports to the center an obfuscated score using differential privacy. Our technique allows for highly effective filtering of corrupted data in a variety of applications. Importantly, we show that most of the corrupted data can be filtered out (recall of  $> 90\%$ , and even up to  $100\%$ ), even under really strong privacy guarantees ( $\epsilon \leq 1$ ).

## 1 Introduction

The success of Machine Learning (ML) depends to a large extent on the availability of high-quality data. This is a particularly important issue in Federated Learning (FL) since the model is trained without access to raw training data. Instead, a single *center* uses data held by a set of independent and sometimes self-interested *data holders* to jointly train a model. Having the ability to *score* and *filter* irrelevant, noisy, or malicious data can (i) significantly improve model accuracy, (ii) speed up training, and even (iii) reduce costs for the center when it pays for data.

Federated Learning [McMahan *et al.*, 2017a; Kairouz *et al.*, 2021; Wang *et al.*, 2021] is different from traditional centralised ML approaches. Challenges such as scalability, communication efficiency, and privacy can no longer be treated as an afterthought; rather, they are *inherent constraints* of the setting. For example, data holders often operate resource-constrained edge devices, and include businesses and/or medical institutions that must protect the privacy of their data due to confidentiality or legal constraints.

We are the *first* to introduce a *practical* approach for *scoring*, and *filtering* contributed data in a Federated Learning setting that ensures *strong*, *worst-case* privacy.

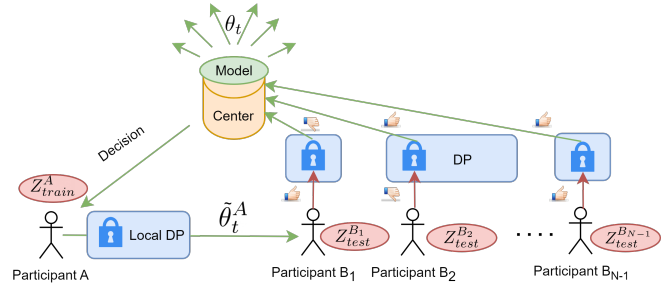


Figure 1: Data filtering procedure. A Center heads a federation of participants  $A, B_1, \dots, B_{N-1}$  that each hold private data relevant to the joint model. Participant  $A$  sends an obfuscated ‘lazy’ (i.e., partial/approximate) parameter update to participants  $B_i$ , who evaluate it using their own testing data, and vote on the quality. Their votes are aggregated using a differentially private mechanism and used by the Center  $C$  to decide on whether to incorporate  $A$ ’s data. See Section 1.2.

A clean way of quantifying the effect of data point(s) on the accuracy of a model is via the notion of *influence* [Koh and Liang, 2017; Cook and Weisberg, 1980]. Intuitively, influence quantifies the marginal contribution of a data point (or batch of points) on a model’s accuracy. One can compute this by comparing the difference in the model’s empirical risk when trained with and without the point in question. While the influence metric can be highly informative, it is impractical to compute: re-training a model is time-consuming, costly, and often impossible, as participants do not have access to the entire dataset. We propose a simple and practical approximation of the *sign* of the exact influence (*‘lazy’ influence approximation*), which is based on an estimate of the direction of the model after a small number of local training epochs with the new data.

Another challenge is to approximate the influence while preserving the privacy of the data. Many approaches to Federated Learning (e.g., [McMahan *et al.*, 2018; Triastcyn and Faltings, 2019]) remedy this by combining FL with Differential Privacy (DP) [Dwork, 2006a; Dwork, 2006b; Dwork *et al.*, 2006a; Dwork *et al.*, 2006b], a data anonymization technique that is viewed by many researchers as the gold standard [Triastcyn, 2020]. We show how the sign of influence can be approximated in an FL setting while maintain-

ing strong differential privacy guarantees. Specifically, there are two sets of participants’ data that we need to protect: the training and the test data (see also Section 1.2). For the training data being evaluated, we clip and add noise to the gradients according to [McMahan *et al.*, 2017b], which achieves a *local* differential privacy guarantee. To ensure the privacy of the test data and the influence approximation itself, we employ a differentially private defense mechanism based on the idea of randomized response [Warner, 1965] (inspired by [Erlingsson *et al.*, 2014]). Together the two mechanisms ensure strong, *worst-case privacy* guarantees, while allowing for accurate filtering of data.

The proposed approach can be used as a ‘right of passage’ every time a participant joins the federation, or periodically during communication rounds (most resource intensive, but would provide the best results), or even as a diagnostic tool. A quality score is useful for various purposes beyond filtering poor data, such as rewarding the data provider, incentivizing users in a crowdsourcing application, assessing a data provider’s reputation, and so on.

## 1.1 Our Contributions

There are two major challenges we address in this work: (i) efficiently estimating the quality of a batch of training data, and (ii) keeping both the training and test data used for this estimate private. For the former, we develop a novel metric called ‘*lazy*’ influence, while for the latter we add noise to the gradients, and propose a differentially private voting scheme. More specifically:

(1) We present a novel technique (*‘lazy’ influence approximation*) for scoring and filtering data in Federated Learning.

(2) Our proposed distributed influence aggregation scheme allows for a robust scoring, even under really strict, worst-case differential privacy guarantees (privacy cost  $\epsilon < 1$ ). This is the recommended value in DP literature, and much smaller than many other AI or ML applications.<sup>1</sup>

(3) We evaluate our approach on two well-established datasets (CIFAR10, and CIFAR100), and demonstrate that filtering using our scheme can eliminate the negative effects of inaccurate data.

## 1.2 High Level Description of Our Setting

A center  $C$  coordinates a set of participants to train a single model (Figure 1).  $C$  has a small set of ‘warm-up’ data which are used to train an initial model  $M_0$  that captures the desired input/output relation. We assume that each data holder has a set of training points that will be used to improve the model, and a set of test points that will be used to evaluate the contributions of other participants. To prohibit participants from tailoring their contributions to the test data, it must be kept private. For each federated learning round  $t$  (model  $M_t$ ), each data holder participant will assume two roles: the role of the

<sup>1</sup>AI or ML applications often assume  $\epsilon$  as large as 10 [Triastcyn and Faltings, 2019] (see e.g., [Tang *et al.*, 2017]). For certain attacks,  $\epsilon = 10$  means that an adversary can theoretically reach accuracy of 99.99% [Triastcyn and Faltings, 2019]

contributor ( $A$ ), and the role of the tester ( $B$ ). As a contributor, a participant performs a small number of local epochs to  $M_t$  – enough to get an estimate of the gradient<sup>2</sup> – using a batch of his training data  $z_{A,t}$ . Subsequently,  $A$  sends the updated partial model  $M_{t,A}$ , with specifically crafted noise to ensure local DP, to every other participant (which assumes the role of a tester). The applied noise protects the update gradient, while still retaining information on the usefulness of data. Each tester  $B$  uses its test dataset to approximate the empirical risk of  $A$ ’s training batch (i.e., the approximate influence). This is done by evaluating each test point and comparing the loss. In a FL setting, we can not re-train the model to compute the exact influence; instead,  $B$  performs only a small number of training epochs, enough to estimate the direction of the model (‘*lazy*’ influence approximation). As such, we opt to look at the sign of the approximate influence (and not the magnitude). Each tester aggregates the signs of the influence for each test point, applies controlled noise to ensure DP, and sends this information to the center. Finally, the center decides to accept  $A$ ’s training batch if the majority of  $B$ s report positive influence, and reject otherwise.

## 2 Related Work and Discussion

**Federated Learning** Federated Learning (FL) [McMahan *et al.*, 2017a; Kairouz *et al.*, 2021; Wang *et al.*, 2021; Li *et al.*, 2020] has emerged as an alternative method to train ML models on data obtained by many different agents. In FL a center coordinates agents who acquire data and provide model updates. FL has been receiving increasing attention in both academia [Lim *et al.*, 2020; Yang *et al.*, 2019; He *et al.*, 2020; Caldas *et al.*, 2018] and industry [Hard *et al.*, 2018; Chen *et al.*, 2019], with a plethora of real-world applications (e.g., training models from smartphone data, IoT devices, sensors, etc.). Moreover, clustering techniques have seen recent usage in Federated Learning [Shu *et al.*, 2022].

**Influence functions** Influence functions are a standard method from robust statistics [Cook and Weisberg, 1980] (see also Section 3), which were recently used as a method of explaining the predictions of black-box models [Koh and Liang, 2017]. They have also been used in the context of fast cross-validation in kernel methods and model robustness [Liu *et al.*, 2014; Christmann and Steinwart, 2004]. While a powerful tool, computing the influence involves too much computation and communication, and it requires access to the training and testing data (see [Koh and Liang, 2017] and Section 3).

**Data Filtering** A common but computationally expensive approach for filtering in ML is to use the Shapley Value of the Influence to evaluate the quality of data [Jia *et al.*, 2019b; Ghorbani and Zou, 2019a; Jia *et al.*, 2019a; Yan *et al.*, 2020; Ghorbani and Zou, 2019b]. Other work includes for example rule based filtering of least influential points [Ogawa *et al.*, 2013], or constructing weighted data subsets (corsets) [Dasgupta *et al.*, 2009]. While data filtering might not always pose a significant problem in traditional ML, in a FL setting

<sup>2</sup>The number of local epochs is a hyperparameter. We do not need to fully train the model. See Section 3.4.

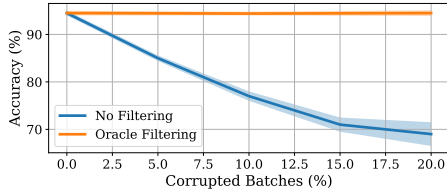


Figure 2: Model accuracy (relative to the fully trained model on the entire MNIST dataset) for increasing mislabeling rates. We compare a centralized model with no filtering of mislabeled data (blue), to a FL model under perfect (oracle) filtering (orange).

it is more important because even a small percentage of mislabeled data can result in a significant drop in the combined model’s accuracy. As a motivating example, consider Figure 2. In this scenario, we have participants with corrupted data (replaced the original label with a random one). Even a 5 – 10% of corrupted participants results in a practically unusable model. Filtering those corrupted participants (orange line), restores the model’s performance.

Because of the privacy requirements in FL, contributed data is not directly accessible for assessing its quality. [Tuor *et al.*, 2021] propose a decentralized filtering process specific to federated learning, yet they do not provide any formal privacy guarantees.

**Differential Privacy** Differential Privacy (DP) [Dwork, 2006a; Dwork, 2006b; Dwork *et al.*, 2006a; Dwork *et al.*, 2006b] has emerged as the de facto standard for protecting the privacy of individuals. Informally, DP captures the increased risk to an individual’s privacy incurred by his participation in the learning process. As a simplified intuitive example, consider a participant being surveyed on a sensitive topic. In order to achieve differential privacy, one needs a source of randomness, thus the participant decides to flip a coin. Depending on the result (heads or tails), the participant can reply truthfully, or at random. Now an attacker can not know if the decision was taken based on the participant’s actual preference, or due to the coin toss. Of course, to get meaningful results, we need to bias the coin towards the true data. In this simple example, the logarithm of the ratio  $Pr[\text{heads}]/Pr[\text{tails}]$  represent the privacy cost (also referred to as the privacy budget), denoted traditionally by  $\epsilon$ . Yet, one has to be careful in designing a DP mechanism, as it is often hard to achieve a meaningful privacy guarantee in a practical way (i.e., avoid adding a lot of noise and maintain high accuracy) [Triastcyn and Faltings, 2019; Danassis *et al.*, 2022]. A variation of DP, especially useful in our context, given the decentralized nature of federated learning, is Local Differential Privacy (LDP) [Dwork *et al.*, 2014]. LDP is a generalization of DP that provides a bound on the outcome probabilities for any pair of individual participants rather than populations differing on a single participant. Intuitively, it means that one cannot hide in the crowd. Another strength of LDP is that it does not use a centralized model to add noise—participants sanitize their data themselves—providing privacy protection against a malicious data curator. For a more comprehensive overview of DP, we refer the reader to [Triastcyn, 2020;

Dwork *et al.*, 2014]. We assume that the participants and the Center are *honest but curious*, i.e., they don’t actively attempt to corrupt the protocol but will try to learn about each other’s data.

### 3 Methodology

We aim to address two challenges: approximating the influence of a (batch of) datapoint(s) without having to re-train the entire model from scratch, and doing so while protecting privacy of training and testing data. The latter is important not only to protect the sensitive information of users, but also to ensure that malicious participants can not tailor their contributions to the test data.

In what follows, we first introduce the notion of *influence* [Cook and Weisberg, 1980], and our proposed ‘lazy’ approximation. Second, we describe a differentially private reporting scheme for crowdsourcing the approximate influence values.

#### 3.1 Setting

We consider a classification problem from some input space  $\mathcal{X}$  (e.g., features, images, etc.) to an output space  $\mathcal{Y}$  (e.g., labels). In a Federated Learning setting, there is a center  $C$  that wants to learn a model  $M(\theta)$  parameterized by  $\theta \in \Theta$ , with a non-negative loss function  $L(z, \theta)$  on a sample  $z = (\bar{x}, y) \in \mathcal{X} \times \mathcal{Y}$ . Let  $R(Z, \theta) = \frac{1}{n} \sum_{i=1}^n L(z_i, \theta)$  denote the empirical risk, given a set of data  $Z = \{z_i\}_{i=1}^n$ . We assume that the empirical risk is differentiable in  $\theta$ . The training data are supplied by a set of data holders.

#### 3.2 Exact Influence

In simple terms, influence measures the marginal contribution of a data point on a model’s accuracy. A positive influence value indicates that a data point improves model accuracy, and vice-versa. More specifically, let  $Z = \{z_i\}_{i=1}^n$ ,  $Z_{+j} = Z \cup z_j$  where  $z_j \notin Z$ , and let

$$\hat{R} = \min_{\theta} R(Z, \theta) \quad \text{and} \quad \hat{R}_{+j} = \min_{\theta} R(Z_{+j}, \theta)$$

where,  $\hat{R}$  and  $\hat{R}_{+j}$  denote the minimum empirical risk their respective set of data. The *influence* of datapoint  $z_j$  on  $Z$  is defined as:

$$\mathcal{I}(z_j, Z) \triangleq \hat{R} - \hat{R}_{+j} \quad (1)$$

Despite being highly informative, influence functions have not achieved widespread use in Federated Learning (or Machine Learning in general). This is mainly due to the computational cost. Equation 1 requires a complete retrain of the model, which is time-consuming, and very costly; especially for state-of-the-art, large ML models. Moreover, specifically in our setting, we do not have direct access to the training data. In the following section, we will introduce a practical approximation of the influence, applicable in Federated Learning scenarios.

#### 3.3 Influence Approximation

Koh and Liang [2017] adopted the first order Taylor approximation of influence (based on [Cook and Weisberg, 1982]) to understand the effects of training points on the predictions of

a *centralised* ML model. To the best of our knowledge, this is the current state-of-the-art approach to utilizing the influence function in ML, thus it is worth taking the time to understand the challenges that arise if we try to adopt this approximation in the Federated Learning setting.

Let  $\hat{\theta} = \arg \min_{\theta} R(Z, \theta)$  denote the empirical risk minimizer. The approximate influence of a training point  $z_j$  on the test point  $z_{test}$  can be computed without having to re-train the model, according to the following equation:

$$\mathcal{I}_{appr}(z_j, z_{test}) \triangleq -\nabla_{\theta} L(z_{test}, \hat{\theta}) H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z_j, \hat{\theta}) \quad (2)$$

where  $H_{\hat{\theta}}^{-1}$  is the inverse Hessian computed on all the model’s training data. The advantage of Equation 2 is that we can answer counterfactuals on the effects of up/down-scaling a training point, without having to re-train the model. One can potentially average over the test points of a tester participant, and/or across the training points in a batch of a contributor participant, to get the total influence.

### Challenges

While Equation 2 can be an effective tool in understanding centralised machine learning systems, it is *ill-matched* for Federated Learning models, for several key reasons.

To begin with, evaluating Equation 2 requires *forming and inverting* the Hessian of the empirical risk. With  $n$  training points and  $\theta \in \mathbb{R}^m$ , this requires  $O(nm^2 + m^3)$  operations [Koh and Liang, 2017], which is *impractical* for modern day deep neural networks with millions of parameters. To overcome these challenges, Koh and Liang [2017] used implicit Hessian-vector products (HVPs) to more efficiently approximate  $\nabla_{\theta} L(z_{test}, \hat{\theta}) H_{\hat{\theta}}^{-1}$ , which typically requires  $O(p)$  [Koh and Liang, 2017]. While this is a somewhat more efficient computation, it is *communication-intensive*, as it requires *transferring all of the (either training or test) data* at each FL round. Most importantly, it *can not provide any privacy* to the users’ data; an important, inherent requirement/constraint in FL.

Finally, to be able to compute Equation 2, the loss function has to be strictly convex and twice differentiable (which is not always the case in modern ML applications). Koh and Liang [2017] propose to swap out non-differentiable components for smoothed approximations, but there is no quality guarantee of the influence calculated in this way.

### 3.4 ‘Lazy’ Influence: A Practical Influence Metric for Filtering Data in FL Applications

The key idea is that *we do not need to approximate the influence value* to filter data; we only need an accurate estimate of its *sign* (in expectation). Recall that a positive influence value indicates that a data point improves model accuracy, and vice-versa, thus we only need to approximate the sign of Equation 1, and use that information to filter out data whose influence falls below a certain threshold.

Our proposed approach works as follows (recall that each data holder participant assumes two roles: the role of the contributor ( $A$ ), and the role of the tester ( $B$ )):

(i) For each federated learning round  $t$  (model  $M_t(\theta_t)$ ), the contributor participant  $A$  performs a small number  $k$  of

---

#### Algorithm 1: Filtering Poor Data Using Influence Approximation in Federated Learning

---

**Data:**  $\theta_0, Z_i, Z_{test}, Z_{init}$

**Result:**  $\theta_T$

```

1  $C$ : The center ( $C$ ) initializes the model  $M_0(\theta_0)$ 
2 for  $t \in T$  rounds of Federated Learning do
3    $C$ : Broadcasts  $\theta_t$ 
4   for  $P_i$  in Participants do
5      $P_i$ : Acts as a contributor ( $A$ ). Performs  $k$  local
      epochs with  $Z_{A,t}$  on the partially-frozen
      model  $\tilde{\theta}_t^A$ .
6      $P_i$ : Applies DP noise to  $\tilde{\theta}_t^A$ .
7      $P_i$ : sends last layer of  $\tilde{\theta}_t^A$  to  $Participants_{-i}$ .
8     for  $P_j$  in  $Participants_{-i}$  do
9        $P_j$ : Acts as a tester ( $B$ ). Evaluates the loss
      of  $Z_{test}^B$  on  $\theta_t$ 
10       $P_j$ : Evaluates the loss of  $Z_{test}^B$  on  $\tilde{\theta}_t^A$ 
11       $P_j$ : Calculates vote  $v$  (sign of influence),
      according to Equation 3
12       $P_j$ : Applies noise to  $v$  according to his
      privacy parameter  $p$  to get  $v'$ 
13       $P_j$ : Sends  $v'$  to  $C$ 
14      $C$ : Filters out  $P_i$ ’s data based on the votes
      from  $Participants_{-i}$  (i.e., if
       $\sum_{\forall B} I_{proposed}(Z_{test}^B) < T$ ).
15    $C$ : Updates  $\theta_t$  using data from unfiltered
      Participants;
```

---

local epochs to  $M_t$  using a batch of his training data  $Z_{A,t}$ , resulting in  $\tilde{\theta}_t^A$ .  $k$  is a hyperparameter.  $\tilde{\theta}_t^A$  is the partially trained model of participant  $A$ , where most of the layers, except the last one have been frozen. The model should not be fully trained for two key reasons: efficiency, and avoiding over-fitting (e.g., in our simulations we only performed 1-9 epochs). Furthermore,  $A$  adds noise to  $\tilde{\theta}_t^A$  (see Section 3.5) to ensure strong, worst-case local differential privacy. Finally,  $A$  sends only the last layer (to reduce communication cost) of  $\tilde{\theta}_t^A$  to every other participant.

(ii) Each tester  $B$  uses his test dataset  $Z_{test}^B$  to estimate the sign of the influence using Equation 3. Next, the tester applies noise to  $I_{proposed}(Z_{test}^B)$ , as will be explained in Section 3.6, to ensure strong, worst-case differential privacy guarantees (i.e., keep his test dataset private).

$$I_{proposed}(Z_{test}^B) \triangleq \text{sign} \left( \sum_{z_{test} \in Z_{test}^B} L(z_{test}, \theta_t) - L(z_{test}, \theta_t^A) \right) \quad (3)$$

(iii) Finally, the center  $C$  aggregates the obfuscated votes  $I_{proposed}(Z_{test}^B)$  from all testers, and filters out data with cumulative score *below a threshold* ( $\sum_{\forall B} I_{proposed}(Z_{test}^B) < T$ ). Specifically, we cluster the votes into two clusters (using k-means), and use the arithmetic mean of the cluster centers as the filtration threshold.

The pseudo-code of the proposed approach is presented in Algorithm 1.

### Advantages of the proposed ‘lazy’ influence

The designer may select any optimizer to perform the model updates, depending on the application at hand. We do not require the loss function to be twice differentiable and convex; only once differentiable. It is significantly more *computation and communication efficient*; an important prerequisite for any FL application. This is because participant  $A$  only needs to send (a *small part* of) the model parameters  $\theta$ , and not his training data. Moreover, computing a few model updates (using e.g., SGD, or any other optimizer) is significantly faster than computing either the exact influence or an approximation, due to the challenges mentioned in Sections 3.2 and 3.3. Finally, and importantly, we ensure the *privacy* of both the train and test dataset of every participant.

### 3.5 Sharing the Partially Updated Joint Model: Privacy and Communication Cost

Each contributor participant  $A$  shares a partially trained model  $\hat{\theta}_t^A$  (see step (i) of Section 3.4). It is important to stress that  $A$  only sends the last layer of the model. This has two significant benefits: it *reduces the communication overhead* (in our simulations, *we only send 0.009% of the model’s weights*),<sup>3</sup> and minimize the impact of the differential privacy noise. We follow [McMahan *et al.*, 2017b] to ensure strong local differential privacy guarantees by (i) imposing a bound on the gradient (using a clipping threshold  $\Delta$ ), and (ii) adding carefully crafted Gaussian noise (parameterised by  $\sigma$ ). For more details, please see [McMahan *et al.*, 2017b].

### 3.6 Differentially Private Reporting of the Influence

Along with the training data, we need to also ensure the privacy of the test data used to calculate the influence. Protecting the test data in a FL setting is critical since (i) it is an important constraint of the FL setting, (ii) participants want to keep their sensitive information (and potential means of income, e.g., in a crowdsourcing application) private, and (iii) the center wants to ensure that malicious participants can not tailor their contributions to the test set.

We proposed to obfuscate the influence reports using RAPPOR [Erlingsson *et al.*, 2014], which results in an  $\epsilon$ -differential privacy guarantee [Dwork *et al.*, 2006b]. The obfuscation process (permanent randomized response [Warner, 1965]) takes as input the participant’s true influence value  $v$  (binary) and privacy parameter  $p$ , and creates an obfuscated (noisy) reporting value  $v'$ , according to Equation 4. Subsequently,  $v'$  is memorized and reused for all future reports on this distinct value  $v$ .

$$v' = \begin{cases} +1, & \text{with probability } \frac{1}{2}p \\ -1, & \text{with probability } \frac{1}{2}p \\ v, & \text{with probability } 1 - p \end{cases} \quad (4)$$

$p$  is a *user-tunable* parameter that allows the participants themselves to *choose their desired level of privacy*, while

<sup>3</sup>Moreover, as explained in the Introduction, this communication cost will be incurred as little as *one* time, when we use our approach as a ‘right of passage’ every time a participant joins the federation.

maintaining reliable filtering. The worst-case privacy guarantee can be computed by each participant *a priori*, using the following formula [Erlingsson *et al.*, 2014]:

$$\epsilon = 2 \ln \left( \frac{1 - \frac{1}{2}p}{\frac{1}{2}p} \right) \quad (5)$$

It is important to note that in a Federated Learning application, the center  $C$  aggregates the influence sign from a *large number of participants*. This means that even under *really strict* privacy guarantees, *the aggregated influence signs (which is exactly what we use for filtering), will match the true value* in expectation. This results in *high quality filtering*, as we will demonstrate in Section 4.

To demonstrate the effect of Equation 4, we visualize the obfuscation process in Figure 3. Figure 3a shows us the sum of true votes ( $y$ -axis) for the test data of each contributor participant ( $x$ -axis). Here we can see a clear distinction in votes between corrupted and correct batches. Most of the corrupted batches (corrupted contributor participant) take negative values, meaning that the majority of the testers voted against them, while the correct batches are close to the upper bound. Figure 3b demonstrates the effect of applying DP noise ( $\epsilon = 1$ ) to the votes. Due to the noise, differentiating between the two groups becomes more challenging. To find an effective decision threshold, we use k-means to cluster the votes into two clusters and use the arithmetic mean of the cluster centers as the filtration threshold (Figure 3c).

## 4 Evaluation Results

We evaluated the proposed approach on two well-established datasets:

1. **CIFAR10** 32x32 images, 10 classes [Krizhevsky, 2009].
2. **CIFAR100** A dataset similar to the previous one, where the number of classes has been expanded to 100 [Krizhevsky, 2009].

### Setup

Our evaluation involves a single round of Federated Learning. A small portion of every dataset (around 1%) is selected as the ‘warm-up’ data used by the center  $C$  to train the initial model  $M_0$ . Each participant has two datasets: a training batch ( $Z_A$ , see Section 3.4, step (i)) which the participant uses to update the model when acting as the contributor participant, and a test dataset ( $Z_{test}^B$ , see Section 3.4, step (ii)), which the participant uses to estimate the sign of the influence when acting as a tester participant. The ratio of these datasets is 2 : 1. The training batch size is 100 (i.e., the train dataset includes 100 points, and the test dataset 50 points). This means that e.g., for a simulation with 100 participants, each training batch is evaluated on  $50 \times (100 - 1)$  test points, and that for each training batch (contributor participant  $A$ ), the center collected  $(100 - 1)$  estimates of the influence sign (Equation 3).

The corruption used for the evaluation is generated by replacing the original label with a random one. We corrupted 30% of the total batches (i.e., participants). For each corrupted batch, we corrupted 90% of the data points.

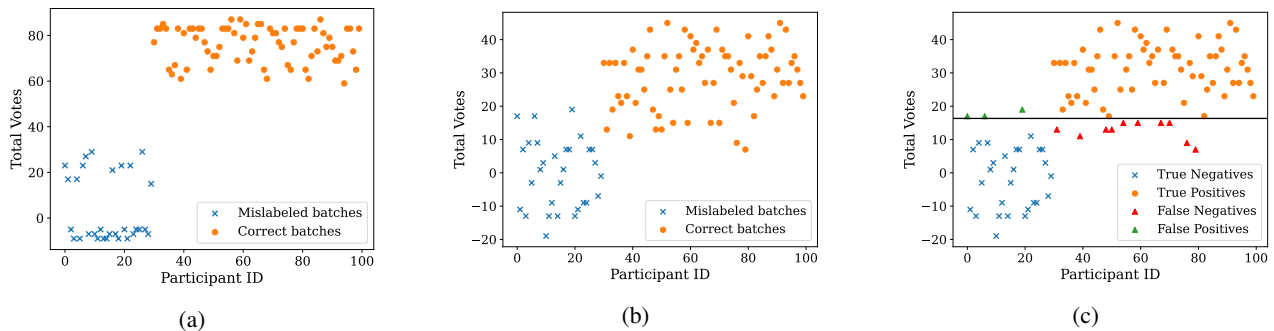


Figure 3: Visualization of the voting scheme. The  $x$ -axis represents a contributor participant  $A$ . The  $y$ -axis shows the sum of all votes from all the testers, i.e.,  $\sum_{\forall B} I_{proposed}(Z_{test}^B)$ . Figure 3a corresponds to the sum of true votes (no privacy) for the test data of each contributor on the  $x$ -axis, while Figure 3b depicts the sum of differentially private votes ( $\epsilon = 1$ ), according to Equation 4. Finally, Figure 3c shows the filtration threshold, which corresponds to the arithmetic mean of the two cluster centers (computed using k-means).

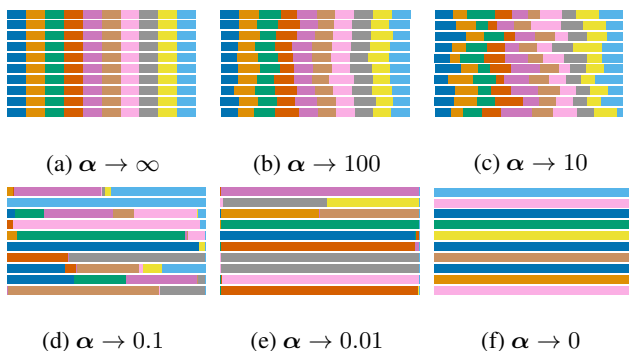


Figure 4: Dirichlet distribution visualisation for 10 classes, parametrized by  $\alpha$ .  $\alpha$  controls the concentration of different classes. Each row represents a participant, each color a different class, and each colored segment the amount of data the participant has from each class. For  $\alpha \rightarrow \infty$ , each participant has the same amount of data from each class (IID distribution). For  $\alpha \rightarrow 0$ , each participant only holds data from one class. In this work, we use  $\alpha \rightarrow 0.1$  for a non-IID distribution.

Each simulation was run 8 times. We report average values, and standard deviations. For a comprehensive overview of the results, please see the supplement.

### Non-IID Setting

The main hurdle for Federated Learning is the fact that not all data is IID. Heterogeneous data distributions are all but uncommon in the real world. To simulate a Non-IID distribution we used Dirichlet distribution to split the training dataset as in related literature [Hsu *et al.*, 2019; Lin *et al.*, 2020; Hoech *et al.*, 2022; Yu *et al.*, 2022]. This distribution is parameterized by  $\alpha$  which controls the concentration of different classes, as visualised in Figure 4. In this work, we use  $\alpha \rightarrow 0.1$  for a non-IID distribution, as in related literature (e.g., [Yu *et al.*, 2022]).

### Implementation

The proposed approach is model-agnostic, and can be used with *any* gradient-descent based machine learning method. For our simulations we used a Vision Transformer (ViT), as

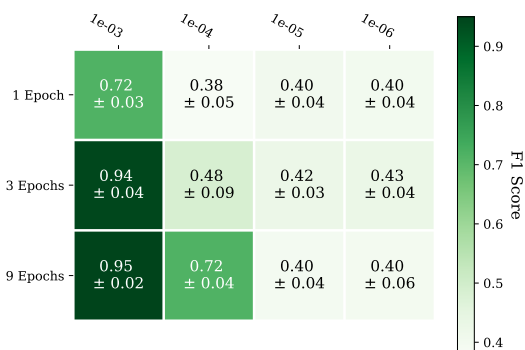


Figure 5: F1 score on Cifar10, IID,  $\epsilon = 1$ , with 100 participants. We vary the training parameters (training epochs in the vertical axis, learning rate in the horizontal) used for partially training a model by the contributor participant  $A$  (see step (i) of Section 3.4).

		Filtration Metrics			
		Distribution	Recall	Precision	Accuracy
CIFAR 10	IID		97.08 $\pm$ 3.51 %	91.91 $\pm$ 7.15 %	96.38 $\pm$ 2.83 %
	Non-IID		93.75 $\pm$ 5.12 %	69.02 $\pm$ 6.28 %	85.00 $\pm$ 3.28 %
CIFAR 100	IID		99.17 $\pm$ 2.20 %	97.96 $\pm$ 2.30 %	99.12 $\pm$ 1.27 %
	Non-IID		92.50 $\pm$ 5.71 %	55.41 $\pm$ 3.94 %	75.12 $\pm$ 3.76 %

Table 1: Quality of filtration metrics for a setting with 100 participants, under strict worst-case privacy guarantees ( $\epsilon = 1$ ). Please see the supplement for the complete results.

it exhibits state-of-the-art performance [Dosovitskiy *et al.*, 2020] (specifically, HuggingFace’s implementation [Wolf *et al.*, 2020]).

### 4.1 Recall, Precision, and Accuracy of Filtration

Recall is the most informative metric to evaluate the efficiency of our filtering approach. Recall refers to the ratio of detected mislabeled batches over all of the mislabeled batches. *Including a mislabeled batch can harm a model’s performance significantly more, compared to removing an unaltered batch.* Thus, achieving *high recall* is of paramount importance.

Meanwhile, precision represents the ratio of correctly identified mislabeled batches, over all batches identified as mislabeled. An additional benefit of using the proposed ‘lazy’ influence metric for scoring data is that it also allows us to identify correctly labeled data, which nevertheless do not provide a significant contribution to the model.

Table 1 shows the recall, precision, and accuracy of filtration for our two datasets, for both IID and non-IID ( $\alpha = 0.1$ ) distributions, for a setting with 100 participants, and under strict worst-case differential privacy guarantees ( $\epsilon = 1$ ). The proposed ‘lazy’ influence results in *highly effective filtering* of corrupted data (*recall of > 90%*, in both IID and non-IID settings).

Precision and accuracy are also high. Of course, there is some degradation in the precision for the non-IID setting, but this is to be expected given the low concentration of classes per participant (high degree of non-IID). Importantly, the metrics improve (including the precision) as we increase the number of participants (see Figure 6, horizontal axis). In simple terms, more testers mean more samples of the different distributions, thus ‘honest’ participants get over the filtering threshold, even in highly non-IID settings. As seen in Figure 6b, the precision in the non-IID setting increases dramatically to 85% (for the same  $\epsilon = 1$ ) by increasing the number of participants to just 500.

Finally, Figure 5 depicts the effects of different training parameters (for partially training the model by the contributor participant  $A$ , see step (i) of Section 3.4) to the F1 score (harmonic mean of the precision and recall). Our proposed approach requires *only 3-9 epochs* to achieve high quality filtration, instead of a complete re-training of the model for the exact influence.

## 4.2 Privacy

As expected, there is a trade-off between privacy, and quality of filtration (see Figure 6, vertical axis, where  $\epsilon$  refers to the privacy guarantee for both the training, and test data/participant votes). Nevertheless, Figure 6 demonstrates that our approach can provide *reliable filtration*, even under *really strict, worst-case privacy requirements* ( $\epsilon = 1$ , which is the recommended value in the DP literature [Tristcyn, 2020]). Importantly, our decentralized framework allows each participant to *compute* and *tune* his *own* worst-case privacy guarantee *a priori*, using Equation 5.

The *privacy trade-off can be mitigated*, and the quality of the filtration can be significantly improved, by increasing the number of participants (Figure 6, horizontal axis). The higher the number of participants, the better the filtration (given a fixed number of corrupted participants). This is because as the number of participants increases, the aggregated influence signs (which is exactly what we use for filtering), will match the true value in expectation. For 500 participants, we achieve high quality filtration even for  $\epsilon = 0.75$ . This is important given that in most real-world FL applications, we *expect a large number of participants*.

## 5 Conclusion

Privacy protection is a core element of Federated Learning. However, this privacy also means that it is significantly more

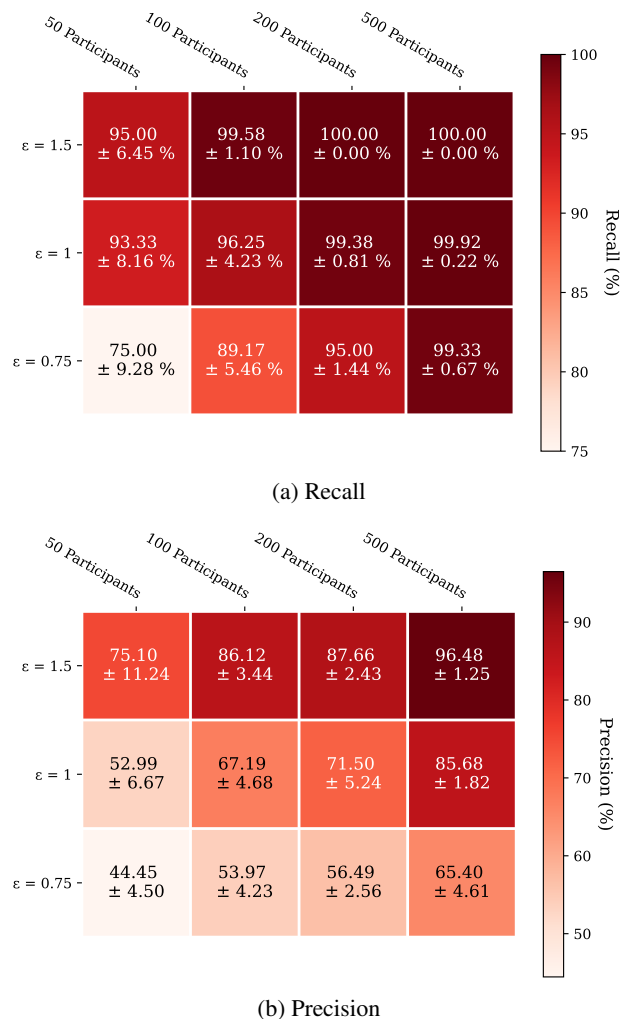


Figure 6: Recall (top), and Precision (bottom) on CIFAR 10, non-IID, for increasing problem size (number of participants), and varying privacy guarantees ( $\epsilon$  – lower  $\epsilon$  provides stronger privacy).

difficult to ensure that the training data actually improve the model. Mislabeled, corrupted, or even malicious data can result in a strong degradation of the performance of model, and privacy protection makes it significantly more challenging to identify the cause.

In this work, we propose the ‘lazy’ influence, a practical approximation of the influence to obtain a meaningful score that characterizes the quality of training data and allows for effective filtering (recall of > 90%, and even up to 100% as we increase the number of participants), while fully maintaining the privacy of both the training and test data under *strict, worst-case*  $\epsilon$ -differential privacy guarantees.

The score can be used to filter bad data, recognize good and bad data providers, and pay data holders according to the quality of their contributions. We have documented empirically that poor data have a significant negative impact on the accuracy of the learned model, and that our filtering technique effectively mitigates this, even under strict privacy requirements  $\epsilon < 1$ .

## References

- [Caldas *et al.*, 2018] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.
- [Chen *et al.*, 2019] Mingqing Chen, Rajiv Mathews, Tom Ouyang, and Françoise Beaufays. Federated learning of out-of-vocabulary words. *arXiv preprint arXiv:1903.10635*, 2019.
- [Christmann and Steinwart, 2004] Andreas Christmann and Ingo Steinwart. On robustness properties of convex risk minimization methods for pattern recognition. *JMLR*, 2004.
- [Cook and Weisberg, 1980] R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 1980.
- [Cook and Weisberg, 1982] R Dennis Cook and Sanford Weisberg. *Residuals and influence in regression*. New York: Chapman and Hall, 1982.
- [Danassis *et al.*, 2022] Panayiotis Danassis, Aleksei Triastcyn, and Boi Faltings. A distributed differentially private algorithm for resource allocation in unboundedly large settings. In *Proceedings of the 21th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS-22*. International Foundation for Autonomous Agents and Multiagent Systems, 2022.
- [Dasgupta *et al.*, 2009] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W Mahoney. Sampling algorithms and coresets for  $\ell_p$  regression. *SIAM Journal on Computing*, 2009.
- [Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.
- [Dwork *et al.*, 2006a] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.
- [Dwork *et al.*, 2006b] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 2006.
- [Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- [Dwork, 2006a] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy, July 2006. Springer Verlag.
- [Dwork, 2006b] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, pages 1–12, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [Erlingsson *et al.*, 2014] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.
- [Ghorbani and Zou, 2019a] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.
- [Ghorbani and Zou, 2019b] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2242–2251. PMLR, 09–15 Jun 2019.
- [Hard *et al.*, 2018] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.
- [He *et al.*, 2020] Chaoyang He, Songze Li, Jinhyun So, Xiao Zeng, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.
- [Hoech *et al.*, 2022] Haley Hoech, Roman Rischke, Karsten Müller, and Wojciech Samek. Fedauxfdp: Differentially private one-shot federated distillation, 2022.
- [Hsu *et al.*, 2019] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.
- [Jia *et al.*, 2019a] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *arXiv preprint arXiv:1908.08619*, 2019.
- [Jia *et al.*, 2019b] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, and Costas Spanos. Towards efficient data valuation based on the shapley value. In *AISTATS*, 2019.
- [Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances



- and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.
- [Koh and Liang, 2017] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.
- [Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [Li *et al.*, 2020] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.
- [Lim *et al.*, 2020] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.
- [Lin *et al.*, 2020] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33:2351–2363, 2020.
- [Liu *et al.*, 2014] Yong Liu, Shali Jiang, and Shizhong Liao. Efficient approximation of cross-validation for kernel methods using bouligand influence function. In *ICML*, 2014.
- [McMahan *et al.*, 2017a] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.
- [McMahan *et al.*, 2017b] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.
- [McMahan *et al.*, 2018] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.
- [Ogawa *et al.*, 2013] Kohei Ogawa, Yoshiki Suzuki, and Ichiro Takeuchi. Safe screening of non-support vectors in pathwise svm computation. In *ICML*, 2013.
- [Shu *et al.*, 2022] Jiangang Shu, Tingting Yang, Xinying Liao, Farong Chen, Yao Xiao, Kan Yang, and Xiaohua Jia. Clustered federated multi-task learning on non-iid data with enhanced privacy. *IEEE Internet of Things Journal*, pages 1–1, 2022.
- [Tang *et al.*, 2017] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. Privacy loss in apple’s implementation of differential privacy on macos 10.12. *arXiv preprint arXiv:1709.02753*, 2017.
- [Triastcyn and Faltings, 2019] Aleksei Triastcyn and Boi Faltings. Federated learning with bayesian differential privacy. In *IEEE International Conference on Big Data (Big Data)*. IEEE, 2019.
- [Triastcyn, 2020] Aleksei Triastcyn. *Data-Aware Privacy-Preserving Machine Learning*. PhD thesis, EPFL, Lausanne, 2020.
- [Tuor *et al.*, 2021] Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. Overcoming noisy and irrelevant data in federated learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5020–5027. IEEE, 2021.
- [Wang *et al.*, 2021] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.
- [Warner, 1965] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
- [Wolf *et al.*, 2020] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [Yan *et al.*, 2020] Tom Yan, Christian Kroer, and Alexander Peysakhovich. Evaluating and rewarding teamwork using cooperative game abstractions. *Advances in Neural Information Processing Systems*, 33:6925–6935, 2020.
- [Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.
- [Yu *et al.*, 2022] Yaodong Yu, Alexander Wei, Sai Praneeth Karimireddy, Yi Ma, and Michael Jordan. TCT: Convexifying federated learning using bootstrapped neural tangent kernels. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.