

---

# Optimal Client Sampling for Federated Learning

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 It is well understood that client-master communication can be a primary bottleneck  
2 in Federated Learning. In this work, we address this issue with a novel client sub-  
3 sampling scheme, where we restrict the number of clients allowed to communicate  
4 their updates back to the master node. In each communication round, all partici-  
5 pated clients compute their updates, but only the ones with “important” updates  
6 communicate back to the master. We show that importance can be measured using  
7 only the norm of the update and give a formula for optimal client participation.  
8 This formula minimizes the distance between the full update, where all clients  
9 participate, and our limited update, where the number of participating clients is  
10 restricted. In addition, we provide a simple algorithm that approximates the optimal  
11 formula for client participation which only requires secure aggregation and thus  
12 does not compromise client privacy. We show both theoretically and empirically  
13 that our approach leads to superior performance for Distributed SGD (DSGD) and  
14 Federated Averaging (FedAvg) compared to the baseline where participating clients  
15 are sampled uniformly. Our approach is orthogonal to and compatible with exist-  
16 ing methods for reducing communication overhead, such as local methods and  
17 communication compression methods.

## 18 1 Introduction

19 We consider the standard cross-device Federated Learning (FL) setting [13], where the objective is of  
20 the form

$$\min_{x \in \mathbb{R}^d} \left[ f(x) := \sum_{i=1}^n w_i f_i(x) \right], \quad (1)$$

21 where  $x \in \mathbb{R}^d$  represents the parameters of a statistical model we aim to find,  $n$  is the total number of  
22 clients,  $f_i: \mathbb{R}^d \rightarrow \mathbb{R}$  is a continuously differentiable local loss function which depends on the data  $\mathcal{D}_i$   
23 owned by client  $i$  via  $f_i = \mathbb{E}_{\xi \sim \mathcal{D}_i} [f(x, \xi)]$ , and  $w_i \geq 0$  are client weights such that  $\sum_{i=1}^n w_i = 1$ .  
24 We assume the classical FL setup in which a central master (server) orchestrates the training by  
25 securely aggregating updates from clients without seeing the raw data.

### 26 1.1 Communication as the Bottleneck

27 It is well understood that cost of communication can be the primary bottleneck in Federated Learning.  
28 Indeed, wireless links and other end-user internet connections typically operate at lower rates than  
29 intra-datacenter or inter-datacenter links and can be potentially expensive and unreliable. Moreover,  
30 the capacity of the aggregating master and other FL system considerations impose direct or indirect  
31 constraints on the number of clients that are allowed to participate in each communication round.  
32 These considerations have led to significant interest in reducing the communication bandwidth of FL  
33 systems.

### 34 1.1.1 Local Methods

35 One of the most popular strategies is to reduce the frequency of communication and put more  
36 emphasis on computation. This is usually achieved by asking the devices to perform multiple local  
37 steps before communicating their updates. A prototype method in this category is the Federated  
38 Averaging (FedAvg) algorithm [23]. The original work was a heuristic, offering no theoretical  
39 guarantees, which motivated the community to try to understand the method and various existing and  
40 new variants theoretically [35, 21, 15, 37, 17, 9].

### 41 1.1.2 Communication Compression

42 Another popular approach is to reduce the size of the object (typically gradients) communicated from  
43 clients to the master. These techniques are usually referred to as gradient/communication *compression*.  
44 In this approach, instead of transmitting the full-dimensional gradient/update vector  $g \in \mathbb{R}^d$ , one  
45 transmits a compressed vector  $\mathcal{C}(g)$ , where  $\mathcal{C} : \mathbb{R}^d \rightarrow \mathbb{R}^d$  is a (possibly random) operator chosen  
46 such that  $\mathcal{C}(g)$  can be represented using fewer bits, for instance by using limited bit representation  
47 (quantization) or by enforcing sparsity (sparsification). A particularly popular class of quantization  
48 operators is based on random dithering [7, 30]; see [1, 41, 42, 28]. A new variant of random dithering  
49 developed in [10] offers an exponential improvement on standard dithering. Sparse vectors can be  
50 obtained by random sparsification techniques that randomly mask the input vectors and preserve a  
51 constant number of coordinates only [40, 18, 36, 24, 39]. There is also a line of work [10, 3] where a  
52 combination of sparsification and quantization was proposed to obtain a more aggressive combined  
53 effect.

### 54 1.2 Related Work

55 Importance sampling methods for optimization have been studied extensively in the last few years in  
56 several contexts, including convex optimization and deep learning. LASVM developed in [5], which is  
57 an online algorithm that uses importance sampling to train kernelized support vector machines. The  
58 first importance sampling for randomized coordinate descent methods was proposed in a seminal  
59 paper in [26]. It was showed in [29] that the proposed sampling is optimal. Later, several extensions  
60 and improvements followed [33, 20, 6, 27, 2, 38]. Another branch of work studies sample complexity.  
61 In [25, 43], the authors make a connection with the variance of the gradient estimates of SGD and  
62 show that the optimal sampling distribution is proportional to the per-sample gradient norm. In terms  
63 of computation, obtaining this distribution is as hard as the computation of the full gradient, thus  
64 it is not practical. For simpler problems, one can sample proportionally to the norms of the inputs,  
65 which can be linked to the Lipschitz constants of the per-sample loss function for linear and logistic  
66 regression. For instance, it was shown in [11] that static optimal sampling can be constructed even  
67 for mini-batches and the probability is proportional to these Lipschitz constants under the assumption  
68 that these constants of the per-sample loss function are known. Unfortunately, importance measures  
69 such as smoothness of the gradient are often hard to compute/estimate for more complicated models  
70 such as those arising in deep learning, where most of the importance sampling schemes are based  
71 on heuristics. A manually designed sampling scheme was proposed in [4]. It was inspired by the  
72 perceived way that human children learn; in practice, they provide the network with examples of  
73 increasing difficulty in an arbitrary manner. In a diametrically opposite approach, it is common for  
74 deep embedding learning to sample hard examples because of the plethora of easy non-informative  
75 ones [32, 34]. Other approaches use a history of losses for previously seen samples to create the  
76 sampling distribution and sample either proportionally to the loss or based on the loss ranking [31, 22].  
77 In [16], the authors propose to sample based on the gradient norm of a small uniformly sampled  
78 subset of samples.

79 In our work, we avoid all the aforementioned problems as our motivation is not to reduce computation,  
80 which is not the main bottleneck of Federated Learning, but to *use importance sampling to decrease*  
81 *the number of bits communicated*. This, as we show in Section 2, allows us to construct *optimal*  
82 *adaptive sampling*; that is, we do not need to rely on any heuristics, historical losses, or partial  
83 information.

84 **1.3 Contributions**

85 In this work, we propose a new approach to addressing the communication bandwidth issues appearing  
 86 in FL. Our approach is based on the observation that in the situation where partial participation  
 87 is desired and a budget on the number of participating clients is applied, *careful selection of the*  
 88 *participating clients can lead to better communication complexity, and hence faster training.* In other  
 89 words, we claim that in any given communication round, some clients will have “more informative”  
 90 updates than others and that the training procedure will benefit from capitalizing on this fact by  
 91 ignoring some of the worthless updates.

92 In particular, we propose a principled *optimal client sampling scheme* capable of identifying the most  
 93 informative clients in any given communication round. Our scheme works by minimizing the variance  
 94 of the stochastic gradient produced by the partial participation procedure, which then translates to  
 95 a probable reduction in the number of communication rounds. To the best of our knowledge, this  
 96 approach was not considered before. Moreover, our proposal is orthogonal to and hence combinable  
 97 with existing approaches to communication reduction such as communication compression or local  
 98 updates (Section 3.2).

99 Our contributions can be summarized as follows:

- 100 • we propose a *novel adaptive partial participation strategy for reducing communication in*  
 101 *FL* that works by a careful selection of the clients that are allowed to communicate their  
 102 updates to the master node in any given communication round;
- 103 • our *adaptive client sampling procedure is optimal* in the sense that it minimizes the variance  
 104 of the master update;
- 105 • we propose an approximation to our optimal adaptive sampling strategy which only requires  
 106 aggregation, thus allows for *secure aggregation* and *stateless clients*;
- 107 • we show theoretically that our approach allows for *larger learning rates* for Distributed SGD  
 108 and FedAvg algorithms than the baseline which performs uniform client sampling, and as a  
 109 result leads to *better communication complexity*.
- 110 • we show empirically that the performance of our approach is superior to uniform sampling  
 111 and is close to full participation.

112 **2 Smart Client Sampling for Reducing Communication**

113 We now describe our client sampling strategy for reducing the communication bottleneck in Federated  
 114 Learning. Each client  $i$  participating in round  $k$  computes an update vector  $\mathbf{U}_i^k \in \mathbb{R}^d$ . For simplicity  
 115 and ease of exposition, we assume that all clients  $i \in [n] := \{1, 2, \dots, n\}$  are available in each round.  
 116 However, we would like to point out that this is not a limiting factor, and all presented theory can be  
 117 easily extended to the case of partial participation with an arbitrary distribution. In our framework,  
 118 only a subset of clients communicates their updates to the master node in each communication round  
 119 in order to reduce the number of transmitted bits.

120 In order to provide analysis in this framework, we consider a general partial participation frame-  
 121 work [12], where we assume that the subset of participating clients is determined by an arbitrary  
 122 random set-valued mapping  $\mathbb{S}$  (a “sampling”) with values in  $2^{[n]}$ . A sampling  $\mathbb{S}$  is uniquely defined  
 123 by assigning probabilities to all  $2^n$  subsets of  $[n]$ . With each sampling  $\mathbb{S}$  we associate a *probability*  
 124 *matrix*  $\mathbf{P} \in \mathbb{R}^{n \times n}$  defined by  $\mathbf{P}_{ij} := \text{Prob}(\{i, j\} \subseteq \mathbb{S})$ . The *probability vector* associated with  $\mathbb{S}$  is  
 125 the vector composed of the diagonal entries of  $\mathbf{P}$ :  $p = (p_1, \dots, p_n) \in \mathbb{R}^n$ , where  $p_i := \text{Prob}(i \in \mathbb{S})$ .  
 126 We say that  $\mathbb{S}$  is *proper* if  $p_i > 0$  for all  $i$ . It is easy to show that  $b := \mathbb{E}[|\mathbb{S}|] = \text{Trace}(\mathbf{P}) = \sum_{i=1}^n p_i$ ,  
 127 and hence  $b$  can be seen as the expected number of clients participating in each communication round.  
 128 Given parameters  $p_1, \dots, p_n \in [0, 1]$ , consider a random set  $\mathbb{S} \subseteq [n]$  generated as follows: for each  
 129  $i \in [n]$ , we include  $i$  in  $\mathbb{S}$  with probability  $p_i$ . This is called *independent sampling*, since the event  
 130  $i \in \mathbb{S}$  is independent of  $j \in \mathbb{S}$  for any  $i \neq j$ .

131 While our client sampling strategy can be adapted to essentially any underlying learning method, we  
 132 give details here for DSGD:

$$x^{k+1} = x^k - \eta^k \mathbf{G}^k, \quad \mathbf{G}^k := \sum_{i \in \mathbb{S}^k} \frac{w_i}{p_i^k} \mathbf{U}_i^k, \quad (2)$$

133 where  $S^k \sim \mathbb{S}^k$  and  $\mathbf{U}_i^k = g_i^k$  is an unbiased estimator of  $\nabla f_i(x^k)$ . The scaling factor  $\frac{1}{p_i^k}$  is necessary  
 134 in order to obtain an unbiased estimator of the true update, i.e.,  $\mathbb{E}_{S^k} [\mathbf{G}^k] = \sum_{i=1}^n w_i \mathbf{U}_i^k$ .

## 135 2.1 Optimal Client Sampling

136 We start with a simple observation that the variance of our gradient estimator  $\mathbf{G}^k$  can be decomposed  
 137 as

$$\mathbb{E} \left[ \left\| \mathbf{G}^k - \nabla f(x^k) \right\|^2 \right] = \mathbb{E} \left[ \left\| \mathbf{G}^k - \sum_{i=1}^n w_i \mathbf{U}_i^k \right\|^2 \right] + \mathbb{E} \left[ \left\| \sum_{i=1}^n w_i \mathbf{U}_i^k - \nabla f(x^k) \right\|^2 \right].$$

138 Note that the second term on the right-hand side is independent of the sampling procedure and  
 139 the first term is zero if every client sends its update (i.e., if  $p_i^k = 1$  for all  $i$ ). In order to provide  
 140 meaningful results, we restrict the expected number of clients to communicate in each round by  
 141 bounding  $b^k := \sum_{i=1}^n p_i^k$  by some positive integer  $m \leq n$ . This raises the following question: *What*  
 142 *is the sampling procedure that minimizes (3) for any given  $m$ ?* We answer this question using the  
 143 following technical lemma:

144 **Lemma 1.** *Let  $\zeta_1, \zeta_2, \dots, \zeta_n$  be vectors in  $\mathbb{R}^d$  and  $w_1, w_2, \dots, w_n$  be non-negative real numbers*  
 145 *such that  $\sum_{i=1}^n w_i = 1$ . Define  $\tilde{\zeta} := \sum_{i=1}^n w_i \zeta_i$ . Let  $S$  be a proper sampling. If  $v \in \mathbb{R}^n$  is such that*

$$\mathbf{P} - pp^\top \preceq \mathbf{Diag}(p_1 v_1, p_2 v_2, \dots, p_n v_n), \quad (3)$$

146 then

$$\mathbb{E} \left[ \left\| \sum_{i \in S} \frac{w_i \zeta_i}{p_i} - \tilde{\zeta} \right\|^2 \right] \leq \sum_{i=1}^n w_i^2 \frac{v_i}{p_i} \|\zeta_i\|^2, \quad (4)$$

147 where the expectation is taken over  $S$ . Whenever (3) holds, it must be the case that  $v_i \geq 1 - p_i$ .

148 It turns out that given probabilities  $\{p_i\}$ , among all samplings  $S$  satisfying  $p_i = \text{Prob}(i \in S)$ , the  
 149 independent sampling minimizes the left-hand side of (4). This is due to two nice properties: a) any  
 150 independent sampling admits optimal choice of  $v$ , i.e.,  $v_i = 1 - p_i$  for all  $i$ , and b) for independent  
 151 sampling (4) holds as equality. In the context of our method, these properties can be written as

$$\mathbb{E} \left[ \left\| \mathbf{G}^k - \sum_{i=1}^n w_i \mathbf{U}_i^k \right\|^2 \right] = \mathbb{E} \left[ \sum_{i=1}^n w_i^2 \frac{1 - p_i^k}{p_i^k} \|\mathbf{U}_i^k\|^2 \right]. \quad (5)$$

152 It now only remains to find the parameters  $\{p_i^k\}$  defining the optimal independent sampling, i.e., one  
 153 that minimizes (5) subject to the constraints  $0 \leq p_i^k \leq 1$  and  $b^k := \sum_{i=1}^n p_i^k \leq m$ . It turns out that  
 154 this problem has the following closed-form solution:

$$p_i^k = \begin{cases} (m + l - n) \frac{\|\tilde{U}_i^k\|}{\sum_{j=1}^l \|\tilde{U}_{(j)}^k\|}, & \text{if } i \notin A^k, \\ 1, & \text{if } i \in A^k, \end{cases} \quad (6)$$

155 where  $\tilde{U}_i^k := w_i \mathbf{U}_i^k$ , and  $\|\tilde{U}_{(j)}^k\|$  is the  $j$ -th largest value in  $\left\{ \|\tilde{U}_i^k\| \right\}_{i=1}^n$ ,  $l$  is the largest integer for  
 156 which  $0 < m + l - n \leq \frac{\sum_{i=1}^l \|\tilde{U}_{(i)}^k\|}{\|\tilde{U}_{(l)}^k\|}$  (note that this inequality at least holds for  $l = n - m + 1$ ), and  
 157  $A^k$  contains indices  $i$  such that  $\|\tilde{U}_i^k\| \geq \|\tilde{U}_{(l+1)}^k\|$ . We summarize this procedure in Algorithm 1.

## 158 2.2 Secure Aggregation

159 Note that in the case  $l = n$ , the optimal probabilities  $p_i^k = m \frac{\|\tilde{U}_i^k\|}{\sum_{j=1}^n \|\tilde{U}_j^k\|}$  can be computed easily: the  
 160 master aggregates the norm of each update and then sends the sum back to the clients. However, if  
 161  $l < n$ , in order to compute optimal probabilities, the master would need to identify the norm of every

---

**Algorithm 1** Optimal Client Sampling (OCS).

---

- 1: **Input:** expected batch size  $m$
  - 2: each client  $i$  computes a local update  $\mathbf{U}_i^k$  (in parallel)
  - 3: each client  $i$  sends the norm of its update  $u_i^k = w_i \|\mathbf{U}_i^k\|$  to the master (in parallel)
  - 4: master computes optimal probabilities  $p_i^k$  using equation (6)
  - 5: master broadcasts  $p_i^k$  to all clients
  - 6: each client  $i$  sends its update  $\frac{w_i}{p_i^k} \mathbf{U}_i^k$  to the master with probability  $p_i^k$  (in parallel)
- 

162 update and perform partial sorting, which can be computationally expensive and also slightly violates  
163 the privacy requirements of clients in FL.

164 Therefore, we develop an algorithm for approximately solving the problem, which only requires to  
165 perform aggregation at the master node without compromising privacy of any client. The construction  
166 of this algorithm is similar to [40]. We first set  $\tilde{p}_i^k = \frac{m \|\tilde{U}_i^k\|}{\sum_{j=1}^n \|\tilde{U}_j^k\|}$  and  $p_i^k = \min\{\tilde{p}_i^k, 1\}$ . In an ideal  
167 situation, this would be sufficient. However, due to the truncation operation, the expected minibatch  
168 size  $b^k = \sum_{i=1}^n p_i^k \leq \sum_{i=1}^n \frac{m \|g_i^k\|}{\sum_{j=1}^n \|g_j^k\|} = m$  can be strictly less than  $m$  if  $\tilde{p}_i^k > 1$  holds true for at  
169 least one  $i$ . Hence, we employ an iterative procedure to fix this gap by rescaling the probabilities  
170 which are smaller than 1, as summarized in Algorithm 2. This algorithm is much easier to implement  
171 and computationally more efficient on parallel computing architectures. In addition, it only requires a  
172 secure aggregation procedure on the master, which is essential in privacy preserving FL, and thus it is  
173 compatible with existing FL software and hardware. We realize that Algorithm 2 brings some extra  
174 communication costs, but this is not an issue as it only requires to communicate  $\mathcal{O}(j_{\max})$  extra floats  
175 for each client. We pick  $j_{\max} = \mathcal{O}(1)$ , and thus it is negligible for large models of size  $d$ .

176 *Remark 1.* We realize that our algorithm requires two communication rounds per optimization round,  
177 but the first round is negligible due to the minimal number of communicated bits as argued above.

### 178 3 Convergence Guarantees

179 In this section, we provide convergence analysis of DSGD and FedAvg with our optimal client sampling  
180 technique and compare it with full participation and independent uniform sampling of  $m$  clients.  
181 We use standard assumptions [14] and assume throughout that  $f$  has a unique minimizer  $x^*$  with  
182  $f^* = f(x^*) > -\infty$ . We further assume that  $f$  is  $\mu$ -strongly convex and  $f_i$ 's are  $L$ -smooth and  
183 convex. Detailed definitions of convexity and smoothness can be found in the Appendix. Note that  
184 nothing prevents us from extending the results in this section to convex and non-convex cases with a  
185 similar standard analysis, since our proposed method only affects the aggregation step as described in  
186 Section 2, which is independent of the strong convexity assumption.

187 **Assumption 1** (Gradient oracle for DSGD). The stochastic gradient estimator  $g_i^k = \nabla f_i(x^k) + \xi_i^k$  of  
188 the local gradient  $\nabla f_i(x^k)$ , for each round  $k$  and all  $i = 1, \dots, n$ , satisfies

$$189 \quad \mathbb{E} [\xi_i^k] = 0 \quad (7)$$

189 and

$$\mathbb{E} [\|\xi_i^k\|^2 | x_i^k] \leq M \|\nabla f_i(x^k)\|^2 + \sigma^2, \quad \text{for some } M \geq 0. \quad (8)$$

190 This further implies that  $\mathbb{E} [\frac{1}{n} \sum_{i=1}^n g_i^k | x^k] = \nabla f(x^k)$ .

191 **Assumption 2** (Gradient oracle for FedAvg). The stochastic gradient estimator  $g_i(y_{i,r}^k) =$   
192  $\nabla f_i(y_{i,r}^k) + \xi_{i,r}^k$  of the local gradient  $\nabla f_i(y_{i,r}^k)$ , for each round  $k$ , each local step  $r = 0, \dots, R$  and  
193 all  $i = 1, \dots, n$ , satisfies

$$194 \quad \mathbb{E} [\xi_{i,r}^k] = 0 \quad (9)$$

194 and

$$\mathbb{E} [\|\xi_{i,r}^k\|^2 | y_{i,r}^k] \leq M \|\nabla f_i(y_{i,r}^k)\|^2 + \sigma^2, \quad \text{for some } M \geq 0, \quad (10)$$

195 where  $y_{i,0}^k = x^k$  and  $y_{i,r}^k = y_{i,r-1}^k - \eta g_i(y_{i,r}^k)$ ,  $r = 1, \dots, R$ .

---

**Algorithm 2** Approximate Optimal Client Sampling (AOCS).

---

- 1: **Input:** expected batch size  $m$ , maximum number of iteration  $j_{\max}$
  - 2: each client  $i$  computes an update  $\mathbf{U}_i^k$  (in parallel)
  - 3: each client  $i$  sends the norm of its update  $u_i^k = w_i \|\mathbf{U}_i^k\|$  to the master (in parallel)
  - 4: master aggregates  $u^k = \sum_{i=1}^n u_i^k$
  - 5: master broadcasts  $u^k$  to all clients
  - 6: each client  $i$  computes  $p_i^k = \min\{\frac{mu_i^k}{u^k}, 1\}$  (in parallel)
  - 7: **for**  $j = 1, \dots, j_{\max}$  **do**
  - 8:   each client  $i$  sends  $t_i^k = (1, p_i^k)$  to the master if  $p_i^k < 1$ ; else sends  $t_i^k = (0, 0)$  (in parallel)
  - 9:   master aggregates  $(I^k, P^k) = \sum_{i=1}^n t_i^k$
  - 10:   master computes  $C^k = \frac{(m-n+I^k)}{P^k}$
  - 11:   master broadcasts  $C^k$  to all clients
  - 12:   each client  $i$  recalibrates  $p_i^k = \min\{C^k p_i^k, 1\}$  if  $p_i^k < 1$  (in parallel)
  - 13:   **if**  $C^k \leq 1$  **then**
  - 14:     break
  - 15:   **end if**
  - 16: **end for**
  - 17: each clients  $i$  sends its update  $\frac{w_i}{p_i^k} \mathbf{U}_i^k$  to master with probability  $p_i^k$  (in parallel)
- 

196 We also define two quantities, which appear in our convergence guarantees:

$$R_i := f_i(x^*) - f_i^*, \quad r^k := x^k - x^*, \quad (11)$$

197 where  $f_i^*$  is the functional value of  $f_i$  at its optimum.  $R_i$  represents the mismatch between the local  
198 and global minimizer, and  $r^k$  captures the distance of the current point to the minimizer of  $f$ .

199 Equipped with these assumptions, we are ready to proceed with our convergence guarantees. We start  
200 with the definition of the improvement factor

$$\alpha^k := \frac{\mathbb{E} \left[ \left\| \sum_{i \in S^k} \frac{w_i}{p_i^k} \mathbf{U}_i^k - \sum_{i=1}^n w_i \mathbf{U}_i^k \right\|^2 \right]}{\mathbb{E} \left[ \left\| \sum_{i \in U^k} \frac{w_i}{p_i^k} \mathbf{U}_i^k - \sum_{i=1}^n w_i \mathbf{U}_i^k \right\|^2 \right]}, \quad (12)$$

201 where  $S^k \sim \mathbb{S}^k$  with  $p_i^k$  defined in (6) and  $U^k \sim \mathbf{U}$  is an independent uniform sampling with  
202  $p_i^U = m/n$ . By construction,  $\alpha^k$  is less than or equal to one, as  $\mathbb{S}^k$  minimizes the variance term. In  
203 addition,  $\alpha^k$  can reach zero in the case where there are at most  $m$  non-zero updates. If  $\alpha^k = 0$ ,  
204 our method performs as if all updates were communicated. In the worst-case  $\alpha^k = 1$ , our method  
205 performs as if we picked  $m$  updates uniformly at random, and one cannot do better due to the  
206 structure of the updates  $\mathbf{U}_i^k$ . In the following subsections, we analyze specific methods for solving the  
207 optimization problem (1) under the aforementioned assumptions. The proofs and detailed description  
208 are deferred to the Appendix.

209 **Fairness.** Based on our sampling strategy, it might be tempting to assume that the obtained solution  
210 could exhibit fairness issues. In our convergence analysis, we show that this is not the case, as our  
211 proposed methods converge to the optimal solution. Hence, as long as the original objective has no  
212 inherent issue with fairness, our methods do not exhibit any fairness issues. Besides, our algorithm  
213 can be used in conjunction with other “more fair” objectives, e.g., tilted ERM [19].

### 214 3.1 Distributed SGD with Optimal Client Sampling

215 We begin with the convergence analysis for DSGD (see (2)) with optimal client sampling.

---

**Algorithm 3** FedAvg with Optimal Client Sampling.

---

```
1: Input: initial global model  $x^1$ , global and local step-sizes  $\eta_g^k, \eta_l^k$ 
2: for each round  $k = 1, \dots, K$  do
3:   master broadcasts  $x^k$  to all clients  $i \in [n]$ 
4:   for each client  $i \in [n]$  (in parallel) do
5:     initialize local model  $y_{i,0}^k \leftarrow x^k$ 
6:     for  $r = 1, \dots, R$  do
7:       compute mini-batch gradient  $g_i(y_{i,r-1}^k)$ 
8:       update  $y_{i,r}^k \leftarrow y_{i,r-1}^k - \eta_l^k g_i(y_{i,r-1}^k)$ 
9:     end for
10:    compute  $\mathbf{U}_i^k := \Delta y_i^k = x^k - y_{i,R}^k$ 
11:    compute  $p_i^k$  using Algorithm 1 or 2
12:    send  $\frac{w_i}{p_i^k} \Delta y_i^k$  to master with probability  $p_i^k$ 
13:  end for
14:  master computes  $\Delta x^k = \sum_{i \in S^k} \frac{w_i}{p_i^k} \Delta y_i^k$ 
15:  master updates global model  $x^{k+1} \leftarrow x^k - \eta_g^k \Delta x^k$ 
16: end for
```

---

216 **Theorem 2.** Let  $f_i$  be  $L$ -smooth and convex for all  $i = 1, \dots, n$ . Let  $f$  be  $\mu$ -strongly convex. Suppose  
217 that Assumption 1 holds. Choose  $\eta^k \in \left(0, \frac{\gamma^k}{(1 + \max_{i \in [n]} \{w_i\} M)L}\right)$ , where

$$\gamma^k := \frac{m}{\alpha^k(n-m) + m} \in \left[\frac{m}{n}, 1\right], \quad k = 0, \dots, K-1.$$

218 Define

$$\beta_1 := \sum_{i=1}^n w_i^2 (2L(1+M)R_i + \sigma^2) \quad \text{and} \quad \beta_2 := 2L \sum_{i=1}^n w_i^2 R_i.$$

219 Then, the iterates of DSGD with optimal client sampling (6) satisfy

$$\mathbb{E} \left[ \|r^{k+1}\|^2 \right] \leq (1 - \mu\eta^k) \mathbb{E} \left[ \|r^k\|^2 \right] + (\eta^k)^2 \left( \frac{\beta_1}{\gamma^k} - \beta_2 \right). \quad (13)$$

220 **Interpretation.** In order to understand the results of Theorem 2, we first look at the best and worst  
221 case scenarios. In the best case scenario, we have  $\gamma^k = 1$  for all  $k$ . This implies that there is no  
222 loss of speed comparing to the method with full participation. It is indeed confirmed by our theory  
223 as our obtained recursion recovers the best-known rate of DSGD in the full participation regime [8].  
224 Similarly, in the worst case, we have  $\gamma^k = m/n$  for all  $k$ 's, which corresponds to uniform sampling  
225 with sample size  $m$  and our recursion recovers the best-know rate for DSGD in this regime. This is  
226 expected as (12) implies that each update  $\mathbf{U}_i^k$  is equivalent, thus we cannot hope for better rate than  
227 the uniform sampling. In the general scenario, our obtain recursion sits somewhere between full  
228 and uniform partial participation, where the actual position is determined by  $\gamma^k$  which capture the  
229 distribution of updates (here gradients) on clients. For instance, with a larger number of  $\gamma^k$ 's tending  
230 to 1, we are closer to full participation regime. Similarly, with more  $\gamma^k$ 's tending to  $m/n$ , we are  
231 closer to the rate of partial participation.

### 232 3.2 FedAvg with Optimal Client Sampling

233 One of the most common approaches to optimization for Federated Learning is Federated Averaging  
234 (FedAvg) [23], an adaption of local-update to parallel SGD. In FedAvg, each client runs some  
235 number of SGD steps locally, and then local updates are averaged to form the global update which is  
236 then used for the global model on the master. Pseudo-code that adapts the standard FedAvg algorithm  
237 to our framework is given in Algorithm 3.

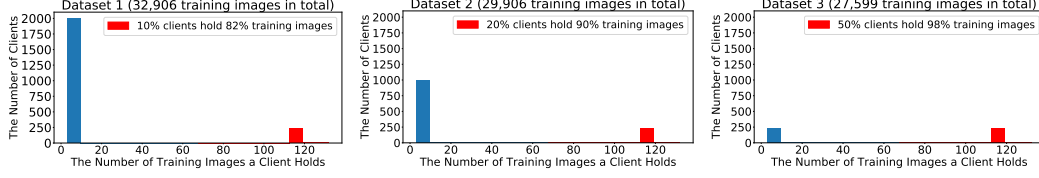


Figure 1: Distributions of the three datasets considered.

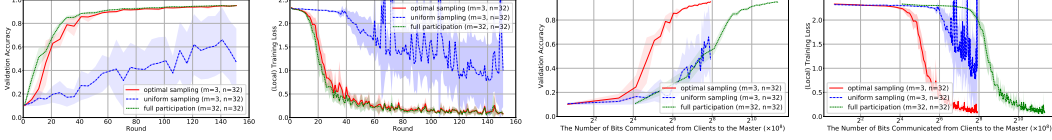


Figure 2: (Dataset 1) validation accuracy and (local) training loss as a function of the number of communication rounds and the number of bits communicated from clients to the master.

238 **Theorem 3.** Assume that  $f_i$  is  $L$ -smooth and  $\mu$ -strongly convex for all  $i = 1, \dots, n$  and Assumption 2  
 239 holds. Let  $\eta^k := R\eta_i^k\eta_g^k$  be the effective step-size and  $\eta_g^k \geq \sqrt{\frac{\gamma^k}{\sum_i w_i^2}}$ , where

$$\gamma^k := \frac{m}{\alpha^k(n-m) + m} \in \left[\frac{m}{n}, 1\right].$$

240 If  $\eta^k \leq \frac{1}{8} \min \left\{ \frac{1}{L(2+M/R)}, \frac{\gamma^k}{(1+\max_{i \in [n]} \{w_i\})(1+M/R)L} \right\}$ , then the iterates of FedAvg ( $R \geq 2$ ) with  
 241 optimal client sampling (6) satisfy

$$\frac{3}{8} \mathbb{E} [(f(x^k) - f^*)] \leq \frac{1}{\eta^k} \left(1 - \frac{\mu\eta^k}{2}\right) \mathbb{E} [\|r^k\|^2] - \frac{1}{\eta^k} \mathbb{E} [\|r^{k+1}\|^2] + \eta^k \beta_1^k + (\eta^k)^2 \beta_2,$$

242 where

$$\beta_1^k := \frac{2\sigma^2}{\gamma^k R} \sum_{i=1}^n w_i^2 + 4L \left(\frac{M}{R} + 1 - \gamma^k\right) \sum_{i=1}^n w_i^2 R_i \quad \text{and} \quad \beta_2 := 72L^2 \left(1 + \frac{M}{R}\right) \sum_{i=1}^n w_i R_i.$$

243 **Interpretation.** Similar to DSGD, the convergence guarantees of FedAvg with optimal client sam-  
 244 pling (Algorithm 3) sits somewhere between the performances of those with full and uniform partial  
 245 participations, where the actual position is again determined by the distribution of updates which  
 246 directly impact  $\alpha^k$ 's that are linked to  $\gamma^k$ 's. In the edge cases, i.e.  $\gamma^k = 1$  (best case) or  $\gamma^k = m/n$   
 247 (worst case), we recover the state-of-the-art complexity guarantees provided in [15] in both regimes.  
 248 Note that our results are slightly more general, as [15] assumes  $M = 0$  and  $w_i = 1/n$ .

## 249 4 Experiments

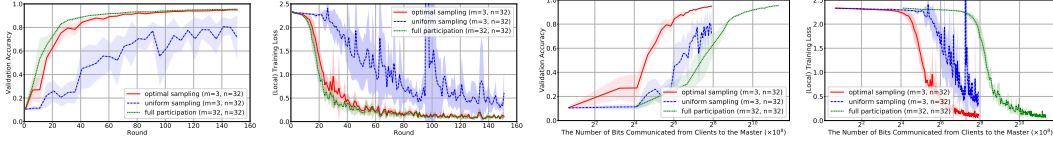
250 In this section, we empirically evaluate our optimal client sampling method, comparing it with 1) the  
 251 baseline where participating clients are sampled uniformly from available clients in each round and  
 252 2) full participation where all available clients participate. We simulate the cross-device FL setting  
 253 and train our models using TensorFlow Federated (TFF)<sup>1</sup>. For all three methods, we report validation  
 254 accuracy and (local) training loss (vertical axis) as a function of the number of communication  
 255 rounds and the number of bits communicated from clients to the master (horizontal axis). Each figure  
 256 displays the mean performance with standard error over 5 independent runs. For a fair comparison,  
 257 we use the same random seed for the three compared methods in a single run and vary random seeds  
 258 across different runs.

259 **Setup.** We conclude an evaluation on FedAvg where we extend the TFF implementation of FedAvg<sup>2</sup>  
 260 to fit our framework. For the model, we use the two-layer Convolutional Neural Network (CNN)

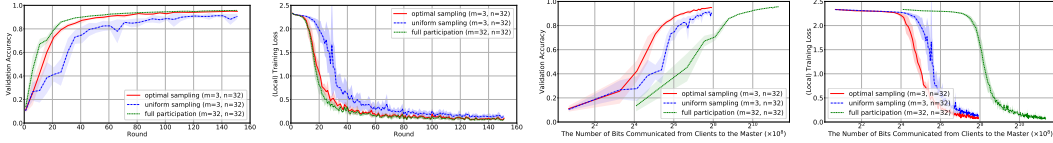
<sup>1</sup><https://github.com/tensorflow/federated>

<sup>2</sup>[https://github.com/tensorflow/federated/tree/master/tensorflow\\_federated/python/examples/simple\\_fedavg](https://github.com/tensorflow/federated/tree/master/tensorflow_federated/python/examples/simple_fedavg)





**Figure 3:** (Dataset 2) validation accuracy and (local) training loss as a function of the number of communication rounds and the number of bits communicated from clients to the master.



**Figure 4:** (Dataset 3) validation accuracy and (local) training loss as a function of the number of communication rounds and the number of bits communicated from clients to the master.

261 provided in the implementation. The default dataset is Federated EMNIST with only digits, but as this  
 262 is a well-balanced dataset with mostly the same quality data on each client, we modify it by removing  
 263 some clients or some of their training images, in order to better simulate conditions in which our  
 264 proposed methods bring significant theoretical improvements. As a result, we produce 3 unbalanced  
 265 datasets as summarized in Figure 1, on which we train the CNN model. For validation, we use the  
 266 unchanged validation set in the Federated EMNIST dataset, which consists of 40, 832 validation  
 267 images. In each communication round of FedAvg,  $n = 32$  clients are sampled uniformly from the  
 268 client pool, each of which then performs several SGD steps on its local training images for 1 epoch  
 269 with batch size 20. For partial participation, the expected number of clients allowed to communicate  
 270 their updates back to the master is set to  $m = 3$  for all the experiments. We use constant step sizes,  
 271 where we set  $\eta_g = 1$  and tune  $\eta_l$  from the set of values  $\{2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}, 2^{-5}\}$  using a holdout  
 272 set. We implement our sampling procedure using Algorithm 2, as this supports stateless clients and  
 273 secure aggregation. We include extra communication costs in our results, where we set  $j_{\max} = 4$ .  
 274 More details of the hyper-parameters that we use can be found in the Appendix.

275 **Results and Discussions.** As predicted by our theory, the performance of FedAvg with our proposed  
 276 optimal client sampling strategy is in between the performances of that with full and uniform partial  
 277 participation. Figures 2, 3 and 4 (red curves: optimal sampling; blue curves: uniform sampling; green  
 278 curves: full participation) show that, for all three datasets, the optimal sampling strategy performs  
 279 slightly worse than but is still competitive with the full participation strategy in terms of the number  
 280 of communication rounds – it almost reached the performance of full participation while only less  
 281 than 10% of the available clients communicate their updates back to the master. Note that the uniform  
 282 sampling strategy performs significantly worse, which indicates that a careful choice of sampling  
 283 probabilities can go a long way towards closing the gap between the performance of naive uniform  
 284 sampling and full participation.

285 More importantly, and this was the main motivation of our work, our optimal sampling strategy is  
 286 significantly better than both the uniform sampling and full participation strategies when we compare  
 287 validation accuracy as a function of the number of bits communicated from clients to the master.  
 288 For instance, in case of Dataset 1 (Figure 2), while our optimal sampling approach reached around  
 289 85% validation accuracy after  $2^6 \times 10^8$  communicated bits, neither the full nor the uniform sampling  
 290 strategies are able to exceed 40% validation accuracy within the same communication budget. Indeed,  
 291 to reach the same 85% validation accuracy, full participation approach needs to communicate more  
 292 than  $2^9 \times 10^8$  bits, i.e.,  $8 \times$  more, and uniform sampling approach needs to communicate about the  
 293 same number of bits as full participation or even more. The results for Datasets 2 and 3 are of a  
 294 similar qualitative nature, showing that these conclusions are robust across the datasets considered.

295 In the Appendix, we include additional figures which show the current best validation accuracy as a  
 296 function of the number of communication rounds and the number of bits communicated from clients  
 297 to the master.

298 **References**

- 299 [1] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Randomized quantization  
300 for communication-optimal stochastic gradient descent. *arXiv preprint arXiv:1610.02132*, 2016.
- 301 [2] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated  
302 coordinate descent using non-uniform sampling. In *International Conference on Machine*  
303 *Learning*, pages 1110–1119, 2016.
- 304 [3] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-SGD: Dis-  
305 tributed SGD with quantization, sparsification and local computations. In *Advances in Neural*  
306 *Information Processing Systems*, pages 14668–14679, 2019.
- 307 [4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning.  
308 In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48,  
309 2009.
- 310 [5] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with  
311 online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619, 2005.
- 312 [6] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent.  
313 *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.
- 314 [7] WM Goodall. Television by pulse code modulation. *Bell System Technical Journal*, 30(1):33–49,  
315 1951.
- 316 [8] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter  
317 Richtárik. SGD: General analysis and improved rates. *Proceedings of the 36th International*  
318 *Conference on Machine Learning, Long Beach, California*, 2019.
- 319 [9] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models.  
320 *arXiv:2002.05516*, 2020.
- 321 [10] Samuel Horváth, Chen-Yu Ho, L’udovít Horváth, Atal Narayan Sahu, Marco Canini, and Peter  
322 Richtárik. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*,  
323 2019.
- 324 [11] Samuel Horváth and Peter Richtárik. Nonconvex variance reduced optimization with arbitrary  
325 sampling. *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- 326 [12] Samuel Horváth and Peter Richtárik. A better alternative to error feedback for communication-  
327 efficient distributed learning. *arXiv preprint arXiv:2006.11077*, 2020.
- 328 [13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Ar-  
329 jun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings,  
330 et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*,  
331 2019.
- 332 [14] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-  
333 gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on*  
334 *Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.
- 335 [15] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich,  
336 and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for on-device federated  
337 learning. *arXiv preprint arXiv:1910.06378*, 2019.
- 338 [16] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning  
339 with importance sampling. *arXiv preprint arXiv:1803.00942*, 2018.
- 340 [17] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on  
341 identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence*  
342 *and Statistics (AISTATS 2020)*, 2020.
- 343 [18] Jakub Konečný and Peter Richtárik. Randomized distributed mean estimation: Accuracy vs.  
344 communication. *Frontiers in Applied Mathematics and Statistics*, 4:62, 2018.

- 345 [19] Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. Tilted empirical risk minimization.  
346 In *International Conference on Learning Representations*, 2021.
- 347 [20] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method.  
348 In *Advances in Neural Information Processing Systems*, pages 3059–3067, 2014.
- 349 [21] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don’t use large mini-batches,  
350 use local SGD. *arXiv preprint arXiv:1808.07217*, 2018.
- 351 [22] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks.  
352 *arXiv preprint arXiv:1511.06343*, 2015.
- 353 [23] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas.  
354 Communication-efficient learning of deep networks from decentralized data. In *Artificial*  
355 *Intelligence and Statistics*, pages 1273–1282, 2017.
- 356 [24] Konstantin Mishchenko, Filip Hanzely, and Peter Richtárik. 99% of parallel optimization is  
357 inevitably a waste of time. *arXiv preprint arXiv:1901.09437*, 2019.
- 358 [25] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling,  
359 and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*,  
360 pages 1017–1025, 2014.
- 361 [26] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems.  
362 *SIAM Journal on Optimization*, 22(2):341–362, 2012.
- 363 [27] Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with  
364 arbitrary sampling. In *Advances in Neural Information Processing Systems 28*, pages 865–873,  
365 2015.
- 366 [28] Ali Ramezani-Kebrya, Fartash Faghri, and Daniel M Roy. NUQSGD: Improved communication  
367 efficiency for data-parallel SGD via nonuniform quantization. *arXiv preprint arXiv:1908.06077*,  
368 2019.
- 369 [29] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent  
370 methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38,  
371 2014.
- 372 [30] Lawrence Roberts. Picture coding using pseudo-random noise. *IRE Transactions on Information*  
373 *Theory*, 8(2):145–154, 1962.
- 374 [31] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay.  
375 *arXiv preprint arXiv:1511.05952*, 2015.
- 376 [32] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for  
377 face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and*  
378 *pattern recognition*, pages 815–823, 2015.
- 379 [33] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent  
380 for regularized loss minimization. In *International conference on machine learning*, pages  
381 64–72, 2014.
- 382 [34] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc  
383 Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In  
384 *Proceedings of the IEEE International Conference on Computer Vision*, pages 118–126, 2015.
- 385 [35] Sebastian U Stich. Local SGD converges fast and communicates little. *ICLR 2019 - International*  
386 *Conference on Learning Representations*, 2019.
- 387 [36] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory.  
388 In *Advances in Neural Information Processing Systems*, pages 4447–4458, 2018.
- 389 [37] Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates  
390 for SGD with delayed gradients and compressed communication. *ICLR 2020 - International*  
391 *Conference on Learning Representations*, 2020.

- 392 [38] Sebastian U Stich, Anant Raj, and Martin Jaggi. Safe adaptive importance sampling. In  
393 *Advances in Neural Information Processing Systems*, pages 4381–4391, 2017.
- 394 [39] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gra-  
395 dient compression for distributed optimization. In *Advances in Neural Information Processing*  
396 *Systems*, pages 14236–14245, 2019.
- 397 [40] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for  
398 communication-efficient distributed optimization. In *Advances in Neural Information Pro-*  
399 *cessing Systems*, pages 1299–1309, 2018.
- 400 [41] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad:  
401 Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural*  
402 *Information Processing Systems*, pages 1509–1519, 2017.
- 403 [42] Hantian Zhang, Jerry Li, Kaan Kara, Dan Alistarh, Ji Liu, and Ce Zhang. Zipml: Training linear  
404 models with end-to-end low precision, and a little bit of deep learning. In *Proceedings of the*  
405 *34th International Conference on Machine Learning-Volume 70*, pages 4035–4043. JMLR. org,  
406 2017.
- 407 [43] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized  
408 loss minimization. In *international conference on machine learning*, pages 1–9, 2015.

409 **Checklist**

- 410 1. For all authors...
- 411 (a) Do the main claims made in the abstract and introduction accurately reflect the paper's  
412 contributions and scope? [Yes]
- 413 (b) Did you describe the limitations of your work? [Yes] See Remark 1, where we  
414 acknowledge that our algorithm requires two (although the first one is negligible)  
415 communication rounds per iteration.
- 416 (c) Did you discuss any potential negative societal impacts of your work? [Yes] We  
417 discussed potential fairness issues in Section 3.
- 418 (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
419 them? [Yes]
- 420 2. If you are including theoretical results...
- 421 (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 3  
422 and Appendix A.
- 423 (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix C-F for  
424 complete proofs. We also provided interpretations of our theorems in the main paper  
425 and discussed the relationship between our results and related results in the literature.
- 426 3. If you ran experiments...
- 427 (a) Did you include the code, data, and instructions needed to reproduce the main experi-  
428 mental results (either in the supplemental material or as a URL)? [Yes] See supplemen-  
429 tal material.
- 430 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
431 were chosen)? [Yes] See Section 4 and Appendix B
- 432 (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
433 ments multiple times)? [Yes] We ran every experiment 5 times with different random  
434 seeds and reported results with error bars.
- 435 (d) Did you include the total amount of compute and the type of resources used (e.g., type  
436 of GPUs, internal cluster, or cloud provider)? [N/A] Since we only run simulations,  
437 this is not applicable.
- 438 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- 439 (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4
- 440 (b) Did you mention the license of the assets? [Yes] All the data and assets we used in this  
441 manuscript are open-source.
- 442 (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]  
443 We included an anonymized URL in the supplemental material for the datasets used.
- 444 (d) Did you discuss whether and how consent was obtained from people whose data you're  
445 using/curating? [N/A]
- 446 (e) Did you discuss whether the data you are using/curating contains personally identifiable  
447 information or offensive content? [N/A]
- 448 5. If you used crowdsourcing or conducted research with human subjects...
- 449 (a) Did you include the full text of instructions given to participants and screenshots, if  
450 applicable? [N/A]
- 451 (b) Did you describe any potential participant risks, with links to Institutional Review  
452 Board (IRB) approvals, if applicable? [N/A]
- 453 (c) Did you include the estimated hourly wage paid to participants and the total amount  
454 spent on participant compensation? [N/A]