FedBC: Federated Learning Beyond Consensus

Anonymous authors
Paper under double-blind review

Abstract

Federated learning (FL) algorithms, such as FedAvg/FedProx, commonly rely on the consensus constraint, enforcing local models to be equal to the global model obtained through the averaging of local updates. However, in practical FL settings with heterogeneous agents, we question the necessity of enforcing consensus. We empirically observe that relaxing consensus constraint an improve both local and global performance to a certain extent. To mathematically formulate it, we replace the consensus constraint in standard FL objective with the proximity between the local and the global model controlled by a tolerance parameter γ, and propose a novel Federated Learning Beyond Consensus (FedBC) algorithm to solve it. Theoretically, we establish that FedBC converges to a first-order stationary point at rates that matches the state of the art, up to an additional error term that depends on a tolerance parameter γ. Finally, we demonstrate that FedBC balances the global and local model test accuracy metrics across a suite of datasets (Synthetic, MNIST, CIFAR-10, Shakespeare), achieving competitive performance with state-of-the-art.

1 Introduction

Federated Learning (FL) has gained popularity as a powerful framework to train machine learning models on edge devices without transmitting the local private data to a central server (McMahan et al., 2017). Mathematically, we can write the FL problem as

$$\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x}) := \frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x}), \tag{1}$$

where $\mathcal{X} \subset \mathbb{R}^d$ is a compact convex set and $F(\mathbf{x})$ is the sum of N possibly non-convex local objectives $f_i(\mathbf{x})$ which could be stochastic as well $f_i(\mathbf{x}) := \mathbb{E}_{\zeta_i}[f(x, \zeta_i)]$ with $\zeta_i \sim \mathbb{P}(\zeta_i)$. Following standard FL literature (McMahan et al., 2017; Karimireddy et al., 2020)), we consider that all the devices are connected in a star topology to a central server. The FL problem is challenging because of the heterogeneity across devices which might be due to different sources, such as the local training data sets can have different sample sizes and might not even necessarily be drawn from a common distribution, meaning that $\mathbb{P}(\zeta_i)$ is allowed to be heterogeneous for each device i. The goal of standard FL is to train a global model $\mathbf{x}^*$ by solving (1), which performs well or at least uniformly across all the clients (McMahan et al., 2017; Li et al., 2020).

In the presence of data heterogeneity across devices, it is highly unlikely that one global model would work well for all devices. This has been highlighted in (Li et al., 2019), where a large spread in terms of performance of the global model was noted across devices. The requirement of uniform performance of the global model across devices is also connected to *fairness* in FL (Li et al., 2019). In FL, the global model is generally constructed from an aggregation of *local models* learned at each device. The simplest is the average of local models in FedAVG (McMahan et al., 2017). When devices' local objectives are distinct, solving (1) can potentially lead to global model which is far away from the local model obtained by solving:

$$\min_{\mathbf{x}\in\mathcal{X}} f_i(\mathbf{x}), \tag{2}$$

for device i. For instance, consider the problem of learning "language models" for a cellphone keyboard, where the goal is to predict the next word. FL can be used in such a case to learn a common global model,

but a global model might fail to capture distinctive writing styles, as well as the cultural nuances of different users. In such a case, a specific local model [cf. (2)] for each device is required; however, due to sub-sampling error, data at device i might not be sufficient to obtain a reasonable model via only local data. Therefore, there are two competing criteria: *global performance* in terms of (1) evaluated at the global model and a *local performance* evaluated at the local model [cf. (2)]. The notion of global and local models naturally arises in FL and exists in FedAVG (McMahan et al., 2017), FedProx (Li et al., 2020), SCAFFOLD (Karimireddy et al., 2020), etc. Predominately, the focus in the existing literature is either on training only the global model or the local model. Hence we pose the following question:

"How can one automate the balance between global and local model performance simultaneously in FL?"

We answer this question affirmatively in this work by developing a novel framework of federated learning beyond consensus (`FedBC`). We propose to consider a problem in which the global objective (1) is primal, which owing to node-separability, allows each device to only prioritize its local objective (2). Then, we introduce a constraint to control the deviation of the local model from the global model with a local hyper-parameter γ_i for each device i.

Contributions. We summarize our *main contributions* as follows:

(1) We provide a novel connection between the global and local model improvement and consensus tolerance parameter which is missing from the literature. To characterize it mathematically, we propose a framework of federated learning beyond consensus, which allows us to calibrate the performance of global and local models across devices in FL (cf. 7). This formulation itself is novel for the FL settings.

(2) We derive the Lagrangian relaxation of this problem and an instantiation of the primal-dual method, which, owing to node-separability of the Lagrangian, admits a federated algorithm we call `FedBC` (cf. Algo. 1).

(3) We establish the convergence of the proposed `FedBC` theoretically and show that the rates are at par with the state of the art. We also illustrate the efficacy of `FedBC` via showing the performance of global and local models on a range of datasets (Synthetic, MNIST, CIFAR-10, Shakespeare).

Related Works. Current approaches in literature tend to focus either only on the performance of the global model (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020), or the local model (Fallah et al., 2020; Hanzely et al., 2020), but do not quantitatively calibrate the trade-off between them. Prioritizing global model performance only amongst the individual devices admits a reformulation as a consensus optimization problem (Nedic & Ozdaglar, 2009; Nedic et al., 2010), which gives rise to FedAvg (McMahan et al., 2017). In this context, it is well-known that averaging steps approximately enforce consensus (Shi et al., 2015), whereas one can enforce the constraint exactly by employing Lagrangian relaxations, namely, ADMM (Boyd et al., 2011), saddle point methods (Nedić & Ozdaglar, 2009), and dual decomposition (Terelius et al., 2011). This fact has given rise to efforts to improve the constraint violation of FL algorithms, as in FedPD (Zhang et al., 2021) and FedADMM (Wang et al., 2022). Other approaches involve using model-agnostic meta-learning (Fallah et al., 2020), in which one executes one gradient step as an approximation for (2) as input for solving (1) with objective $\frac{1}{N}\sum_{i=1}^{N} f_i(\mathbf{x} - \alpha\nabla f_i(\mathbf{x}))$. However, it does not explicitly allow one to trade off local and global performance. Several works have sought to balance these competing local and global criteria based upon regularization (Li et al., 2020; Hanzely et al., 2020; T Dinh et al., 2020; Li et al., 2021b). Alternatives prioritize the performance of the global model amidst heterogeneity via control variate corrections (Karimireddy et al., 2020; Acar et al., 2021). Please refer to Appendix A for additional related work context.

2 Problem Formulation

In this section, to solve (1) in a federated manner, we consider a consensus reformulation of (1), where each device i is now only responsible for its local copy $\mathbf{x}_i$ of the global model $\mathbf{z}$:

$$\min_{\{(\mathbf{z},\mathbf{x}_i)\in\mathcal{X}\}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) \quad \text{s.t. } \mathbf{x}_i = \mathbf{z}, \quad \forall i. \tag{3}$$

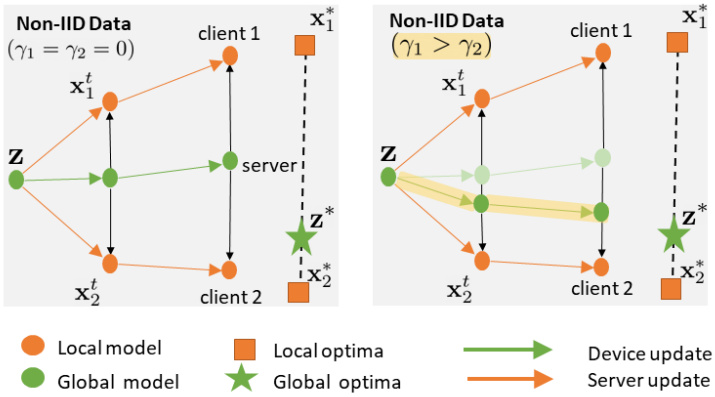


Figure 1: In the left side figure, note that the consensus in standard FL results in averaging at the server, which doesn't allow it to converge to optimal. For the right side figure, the parameter γ_i introduces beyond consensus feature and allows the server model to converge to optimal.

The linear equality constraints $\mathbf{x}_i = \mathbf{z}$ for all i in (3) enforce consensus among all the devices. To solve (3), one may employ techniques from multi-agent optimization (Nedic & Ozdaglar, 2009; Nedic et al., 2010) and consider localized gradient updates followed by averaging steps, as in FedAvg (McMahan et al., 2017). Setting aside the issue of how sharply one enforces the constraints for the moment, observe that in (3), each device must balance between the two competing global and local objectives. These quantities only coincide when the set of minimizers of the sum is contained inside the set of minimizers of each cost function in the sum. This holds only when the sampling distributions $\mathbb{P}(\zeta_i)$ coincide which is not true for FL in general. Efforts to deal with the gap between the global (1) and local (2) objectives have relied upon augmentations of the local objective, e.g.,

$$f_i(\mathbf{x}_i) + (\mu/2)\|\mathbf{x}_i - \mathbf{z}\|^2 \qquad \text{in FedProx,} \tag{4}$$

$$\arg\min_{\boldsymbol{\theta}} f_i(\boldsymbol{\theta}) + (\mu/2)\|\boldsymbol{\theta} - \mathbf{z}\|^2 \qquad \text{in pFedMe,} \tag{5}$$

$$f_i(\mathbf{x} - \nabla f_i(\mathbf{x})) \qquad \text{in Per-FedAvg.} \tag{6}$$

In the above objectives, observe that a penalty coefficient is introduced to obtain a suitable tradeoff between global and local performance. This relationship is even more opaque in meta-learning, as the tradeoff then depends upon mixed first-order partial derivatives of the local objective with respect to the global model – see (Fallah et al., 2020). Therefore, it makes sense to discern whether it is possible to obtain a methodology to solve for the suitable trade-off between local and global performance while solving for the model parameters themselves. To do so, we reinterpret the penalization in (4) as a constraint, which gives rise to the following problem:

$$\min_{\{(\mathbf{z},\mathbf{x}_i)\in\mathcal{X}\}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) \quad \text{s.t. } \|\mathbf{x}_i - \mathbf{z}\|^2 \leq \gamma_i, \quad \forall i\,, \tag{7}$$

for some $\gamma_i \geq 0$. We call this formulation FL beyond consensus because $\gamma_i > 0$ would allow local models to be different from each other and no longer enforces consensus as in (3).

Interpretation of γ_i: The introduction of γ_i provides another degree of freedom to the selection of local $\mathbf{x}_i$ and global model $\mathbf{z}$. Instead of forcing $\mathbf{x}_i = \mathbf{z}$ for all i in (3), they both can differ from each other while still solving the FL problem. For instance, consider the example in Fig. 1 (left), where we generalize the example from (Tan et al., 2022) and show (Fig. 1 (right)) that a strictly positive γ_i can result in a better global model. Further, as a teaser in Fig. 2, we also note experimentally that γ_i calibrates the trade-off between the performance of the local and global model. For simplicity in Fig. 2, we kept γ the same for all i, and we note that local test accuracy and global test accuracy both increase as we start increasing γ from zero, and then eventually global performance starts deteriorating after $\gamma > 0.05$ and local performance is still improving. This makes sense because by making γ larger, we are just focusing on minimizing the individual

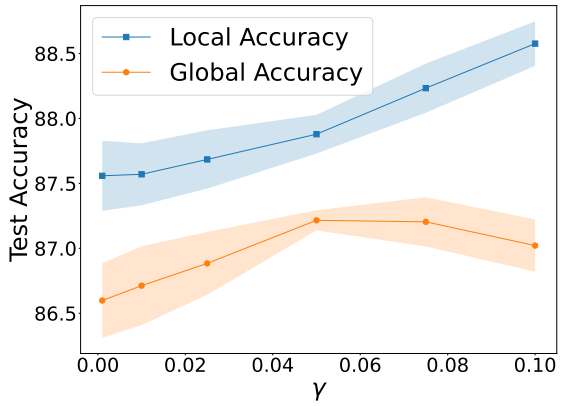


Figure 2: In this figure, $\gamma > 0$ establishes that there is a region $0 < \gamma < 0.05$ to further improve the performance of the global model, as compared to existing FL approaches such as FedAvg, FedProx, SCAFFOLD, etc (where $\gamma = 0$).

loss functions for each device i than focusing on minimizing the sum. But remarkably, the region between $0 \leq \gamma \leq 0.05$ is interesting because both local and global performance increases, which tells us that $\gamma = 0.05$ is superior to choosing than $\gamma = 0$ as used in the standard FL (McMahan et al., 2017; Li et al., 2020). Hence, this basic experiment in Fig. 2 establishes that there is some room to improve the existing FL models (even if we just focus on the performance of the global model) with a non-zero γ_i, which has not yet been utilized anywhere to the best of our knowledge. Therefore, this work is the first attempt to show the benefits of using $\gamma > 0$. We further solidify our claims in Sec. 5. Next, we derive an algorithmic tool to solve (7).

3 FedBC: Federated Learning Beyond Consensus

To solve (7), one could consider the primal-dual method (Nedić & Ozdaglar, 2009) or ADMM (Boyd et al., 2011). However, as the constraints [cf (7) are nonlinear, ADMM requires a nonlinear optimization in the inner loop. Thus, we consider the primal-dual method, which may be derived by Lagrangian relaxation of (7) as:

$$\mathcal{L}(\mathbf{z}, \{\mathbf{x}_i, \lambda_i\}_{i=1}^N) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i(\mathbf{z}, \mathbf{x}_i, \lambda_i), \tag{8}$$

where $\mathcal{L}_i(\mathbf{z}, \mathbf{x}_i, \lambda_i) := \left[f_i(\mathbf{x}_i) + \lambda_i \left(\|\mathbf{x}_i - \mathbf{z}\|^2 - \gamma_i \right) \right]$. Then we alternate between primal minimization and dual maximization. To do so, ideally one would minimize the Lagrangian (8) with respect to $\mathbf{x}_i$ while keeping $\mathbf{z}$ and $\{\lambda_i\}_{i=1}^N$ constant, i.e., at give instant t, we solve for $\mathbf{x}_i$ as

$$\mathbf{x}_i^{t+1} = \operatorname*{argmin}_{\mathbf{x}} \mathcal{L}_i(\mathbf{x}, \mathbf{z}^t, \lambda_i^t). \tag{9}$$

As local objectives may be non-convex, solving (9) is not simpler than solving (7) for given $\mathbf{z}^t$ and λ_i^t. To deal with this, we consider an oracle that provides an ϵ_i-approximated solution of the form

$$\mathbf{x}_i^{t+1} = \text{Oracle}_i(\mathcal{L}_i(\mathbf{x}_i^t, \mathbf{z}^t, \lambda_i^t), K_i); \ [K_i\text{-local updates}] \tag{10}$$

where the ϵ_i-approximate solution $\mathbf{x}_i^{t+1}$ is a stationary point of the Lagrangian in the sense of $\|\nabla_{\mathbf{x}_i} \mathcal{L}_i(\mathbf{x}_i^{t+1}, \mathbf{z}^t, \lambda_i^t)\|^2 \leq \epsilon_i$. In case of a stochastic gradient oracle, this condition instead may be stated as $\mathbb{E}\left[\|\nabla_{\mathbf{x}_i} \mathcal{L}_i(\mathbf{x}_i^{t+1}, \mathbf{z}^t, \lambda_i^t)\|^2\right] \leq \epsilon_i$. We note that any iterative optimization algorithm can be used to perform the K_i local updates. The number of local updates K_i depends upon the accuracy parameter ϵ_i. For instance, in the case of non-convex local objective, a gradient descent-based oracle would need $K_i = \mathcal{O}(1/\epsilon_i)$ and

Algorithm 1 Federated Learning **B**eyond **C**onsensus (`FedBC`)

1: **Input**: T, K_i for each device i, γ_i step size parameters α and β.
2: **Initialize**: $\mathbf{x}_i^0$, $\mathbf{z}^0$, and λ_i^0 for all i.
3: **for** $t = 0$ to $T-1$ **do**
4: **Select** a subset of S devices uniformly from N devices , we get $\mathcal{S}_t \in \{1, 2, \cdots, N\}$
5: Send $\mathbf{z}^t$ to each $j \in \mathcal{S}_t$
6: **Parallel** loop for each device $j \in \mathcal{S}_t$
7: Primal update: $\mathbf{x}_j^{t+1} = \text{Oracle}_j(\mathcal{L}_j(\mathbf{x}_j^t, \mathbf{z}^t, \lambda_j^t), K_j)$ according to Algorithm 2
8: Dual update: $\lambda_j^{t+1} = \mathcal{P}_\Lambda\left[\lambda_i^t + \alpha(\|\mathbf{x}_i^{t+1} - \mathbf{z}^t\|^2 - \gamma_i)\right]$
9: Each device j sends $\mathbf{x}_j^{t+1}, \lambda_j^{t+1}$ back to server
10: **Server** updates

$$\mathbf{z}^{t+1} = \frac{1}{\sum_{j\in\mathcal{S}_t} \lambda_j^{t+1}} \sum_{j\in\mathcal{S}_t} (\lambda_j^{t+1}\mathbf{x}_j^{t+1}) \tag{13}$$

11: **end for**
12: **Output**: $\mathbf{z}^T$

an SGD-based oracle would require $K_i = \mathcal{O}\left(1/\epsilon_i^2\right)$ number of local steps – see (Wright et al., 1999). A gradient descent-based iteration as an instance of (10) is given in Algorithm 2. Next, we present the Lagrange multiplier updates initially under the hypothesis that all devices communicate, which we will subsequently relax. In particular, after collecting the locally updated variables at the server, $\mathbf{x}_i^{t+1}$, the dual variable is updated via a gradient ascent step given by:

$$\lambda_i^{t+1} = \mathcal{P}_\Lambda\left[\lambda_i^t + \alpha(\|\mathbf{x}_i^{t+1} - \mathbf{z}^t\|^2 - \gamma_i)\right], \tag{11}$$

where the dual variable λ_i^{t+1} is projected ($\mathcal{P}_\Lambda$ denotes projection operation) onto a compact domain given by $\Lambda := [\lambda_{\min}, \lambda_{\max}]$, where the values of $\lambda_{\min}$ and $\lambda_{\max}$ will be derived later from the analysis. Then, we shift to minimization with respect to the global model variable $\mathbf{z}$, which by the strong convexity of the Lagrangian [cf. (8)] in this variable is obtained by equating $\nabla_{\mathbf{z}}\mathcal{L}(\mathbf{z}, \{\mathbf{x}_i^{t+1}, \lambda_i^{t+1}\}_{i=1}^N) = 0$ and given by

$$\mathbf{z}^{t+1} = \frac{1}{\sum_{i=1}^N (\lambda_i^{t+1})} \sum_{i=1}^N (\lambda_i^{t+1})\mathbf{x}_i^{t+1}, \tag{12}$$

The server update in (12) requires access to all local models $\mathbf{x}_i$ and Lagrange multipliers λ_i. To perform the update (12), we use device selection as is common in FL, we uniformly sample a set of $|\mathcal{S}_t|$ devices from N total devices. All the steps are summarized in Algorithm 1.

Connection to Existing Approaches: FL algorithms alternate between localized updates and server-level information aggregation. The most common is FedAvg (McMahan et al., 2017), which is an instance of FedBC with $\lambda_i^t = 0$ for all i. Furthermore, FedProx is an augmentation of FedAvg with an additional proximal term in the device loss function. Observe that `FedBC` algorithm with $\lambda_i^t = \mu$ for all i and t reduces to FedProx (Li et al., 2020) for (1). Furthermore, for $\gamma_i = 0$ and without device sampling, the algorithm would become a version of FedPD (Zhang et al., 2021). For constant $\lambda_i = c$ and with $K_i = 1$ local GD step, our algorithm reduces to L2GD (Hanzely & Richtárik, 2020), which is limited to convex settings. (Li et al., 2021b) similarly mandates constant Lagrange multipliers and $K_i = 1$.

4 Convergence Analysis

In this section, we establish performance guarantees of Algorithm 1 in terms of solving the global [cf. (1)] and the local problem (2). We first state the assumptions:

Assumption 4.1. *The domain $\mathcal{X}$ of functions f_i in (2) is compact with diameter R, and at least one stationary point of $\nabla_{\mathbf{x}_i}\mathcal{L}_i(\mathbf{x}_i, \mathbf{z}, \lambda_i) = 0$ belongs to $\mathcal{X}$.*

Algorithm 2 Oracle$_i$ in Equation (10) [K_i-local updates]

Input: K_i, γ_i, β, $\mathbf{x}_i^t, \mathbf{z}^t, \lambda_i^t$
Initialize: $\mathbf{w}_i^0 = \mathbf{x}_i^t$
for $k = 0$ to $K_i - 1$ for each device i **do**
  Update the local model via any optimizer
    GD optimizer:
  $\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \beta \left(\nabla_{\mathbf{w}} f_i(\mathbf{w}_i^k) + (2\lambda_i^t) \left(\mathbf{w}_i^k - \mathbf{z}^t \right) \right)$
    SGD optimizer:
  $\mathbf{w}_i^{k+1} = \mathbf{w}_i^k - \beta \left(\mathbf{g}_i^k + (2\lambda_i^t) \left(\mathbf{w}_i^k - \mathbf{z}^t \right) \right)$
end for
Output: $\mathbf{w}_i^{K_i}$

Assumption 4.2 (Lipschitz gradients)**.** *The gradient of the local objective $\nabla f_i(\mathbf{x})$ of each device is Lipschitz continuous, i.e., $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L_i \|\mathbf{x} - \mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}$.*

Assumption 4.3 (Bounded Heterogeneity)**.** *For any device pair (i, j), it holds that $\max_{(a,b)\in\Lambda} \|a\nabla f_i(\mathbf{x}) - b\nabla f_j(\mathbf{x})\| \leq \delta$, for all $\mathbf{x} \in \mathcal{X}$.*

Next, we describe the assumption required when we use stochastic gradients instead of the actual gradients. If we denote the stochastic gradient for agent i as $\mathbf{g}_i$, it satisfies the following assumption.

Assumption 4.4 (Stochastic Gradient Oracle)**.** *If a stochastic gradient oracle is used at device i, then $\mathbf{g}_i$ satisfies $\mathbb{E}[\mathbf{g}_i \mid \mathcal{H}_k] = \nabla f(\mathbf{x}_i)$, and $\mathbb{E}[\|\mathbf{g}_i - \nabla f(\mathbf{x}_i)\|^2 \mid \mathcal{H}_k] \leq \sigma^2$, $\forall i$, where $\mathcal{H}_k$ is defined as filtration or σ-algebra generated by past realizations $\{\zeta_i^u\}_{u<k}$.*

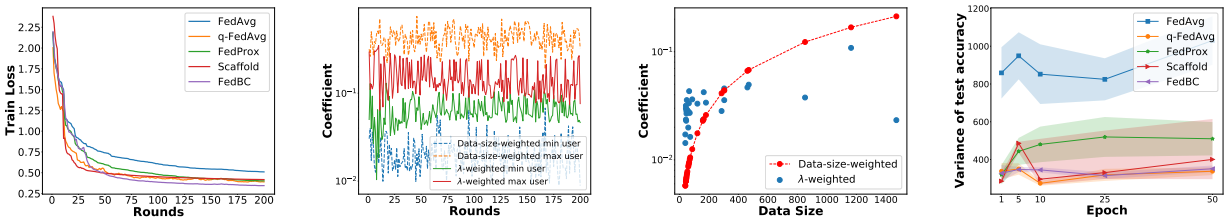
We note that the Assumptions 4.1-4.4 are standard (Nemirovski et al., 2009). Assumption 4.2 makes sure that the local non-convex objective is smooth with parameter L_i. Assumption 4.3 is a version of the heterogeneity assumption considered in the literature (Assumption 3 in (T Dinh et al., 2020)). Assumption 4.4 imposes conditions on the stochastic gradient oracle, particularly unbiasedness and finite variance, which are standard. We are now ready to present the main results of this work in the form of Theorem 4.5. For the convergence analysis, we consider the performance metric $\frac{1}{T}\sum_{t=1}^{T} \|\nabla f(\mathbf{z}^t)\|^2$ which is widely used in the literature (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020; T Dinh et al., 2020). Under these conditions, we have the following convergence result.

Theorem 4.5. *Under Assumption 4.1-4.3, for the iterates of proposed Algorithm 1, we establish that the global performance satisfies:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(\mathbf{z}^t)\|^2] = \mathcal{O}\left(\frac{B_0}{T}\right) + \mathcal{O}(\epsilon) + \mathcal{O}(\delta^2) + \mathcal{O}(\alpha) + \mathcal{O}\left(\frac{1}{NT}\sum_{i=1}^{N}\gamma_i\right), \tag{14}$$

where B_0 is the initialization dependent constant, $\epsilon = \max_i \epsilon_i$ is the accuracy with each agent solves the local optimization problem in the algorithm, δ is the heterogeneity parameter (cf. Assumption 4.3), $\alpha > 0$ is the step size, and γ_i's are the local parameters.

The proof of Theorem 4.5 is provided in Appendix C. The expectation in (14) is with respect to the randomness in the stochastic gradients and device sampling. The first term is the initialization dependent term, and as long as the initialization B_0 is bounded, the first term reduces linearly with respect to T and goes to zero in the limit as $T \to \infty$. This term is present in any state-of-the-art FL algorithm (McMahan et al., 2017; Li et al., 2020; Karimireddy et al., 2020; T Dinh et al., 2020). The second term is $\mathcal{O}(\epsilon)$, which depends upon the worst local approximated solution across all the devices. Note that the individual approximation errors ϵ_i depend on the number of local iterations K_i. This term is also present in most of the analyses of FL algorithms for non-convex objectives. The third term is due to the heterogeneity across the devices and is a specific feature of the FL problem. The fourth term is the step size-dependent term. The last term is important here because that appears due to the introduction of γ_i in the problem formulation in (7), and it is completely novel to the analysis in this work. This term decays linearly even if $\gamma_i > 0$ for all i. The γ_i's



(a) Train loss. (b) Max/min coefficient. (c) Coefficient vs n_i. (d) Test accuracy variance.

Figure 3: We use 30 as the total number of users and $E = 5$. (a) We plot global train loss vs the number of rounds and observe that `FedBC` achieves the lowest train loss. (b) We track the user/device with the smallest (min user) or largest (max user) number of data at each round and plot the min or max user's coefficient in computing the global model based on either its local dataset size (n_i) or λ_i. We observe that the magnitude difference between min and max user's coefficient based on λ_i is consistently smaller than that based on n_i. (c) We plot λ_i against n_i for each user at the end of training and observe that users of small dataset sizes (e.g. < 200) are able to contribute significantly to the global model in `FedBC`. (d) We show the variance of test accuracy of the global model on each user's local data for different Es (shaded area shows standard deviation), and observe that the model of `FedBC` achieves high uniformity.

are directly affecting the global performance because they are allowing device models to move away from each other, hence affecting the global performance. We remark that for the special case of $\gamma_i = 0$ for all i, our result in (4.5) is equivalent to FedPD (Zhang et al., 2021), pFedMe (T Dinh et al., 2020) except for the fact that there is no device sampling in FedPD.

The technical points of departure in the analysis of `FedBC` (cf. Algorithm 1) from prior work are associated with the fact that we build out from an ADMM-style analysis. (Zhang et al., 2021) However, due to non-linear constraint (7), one cannot solve the argmin exactly. This introduces an additional $\mathcal{O}(\epsilon)$ error term that we relate to K_i in (10). This issue also demands we constrain the dual variables to a compact set in (11). Moreover, device sampling for the server update (cf. (13)) is introduced here for the first time in a primal-dual framework, which does not appear in (Zhang et al., 2021). Furthermore, our nonlinear proximity constraints [cf. (7)] additionally permits us to relate the performance in terms of the local objective [cf. (2)] to the proximity to the global model defined in (7) as a function of tolerance parameter γ_i. We formalize their interconnection in the following corollary.

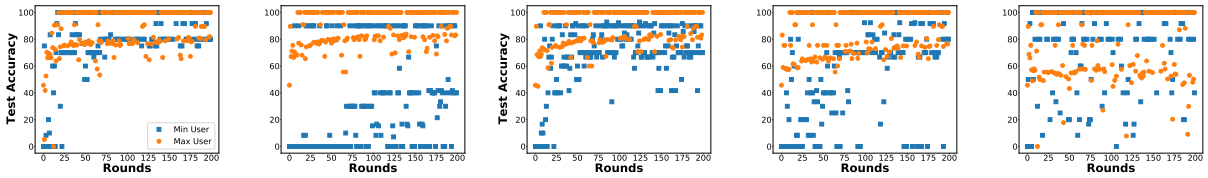
Corollary 4.6 (Local Performance)**.** *Under Assumption 4.1-4.3, for the iterates of Algorithm 1, we establish that*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla f_i(\mathbf{x}_i^t)\|^2 \leq \mathcal{O}\left(\epsilon_i\right) + \alpha^2\mathcal{O}\left(\left[\sum_{k=0}^{T}\mathbb{I}_{\{\|\mathbf{x}_i^k-\mathbf{z}^{k-1}\|^2\leq\gamma_i)\}}\right]_+\right)^2. \tag{15}$$

The proof is provided in Appendix D. We note that the local stationarity of each client i actually depends on the local ϵ_i approx error and γ_i via a complicated term present in the second term in (15), where $\mathbb{I}$ is an indicator function that is 1 if the condition is not satisfied, and -1 otherwise. We note that the term inside the big bracket is larger (worse local performance) for lower γ_i, and vice versa. Hence we have a relationship between global and local model performance in terms of γ_i.

5 Experiments

In this section, we aim to address the following questions with our experiments: ① *Does the introduction of γ_i help `FedBC` to improve global performance compared to other FL algorithms in heterogeneous environments?* ② *Does `FedBC` allow users to have their own localized models and to what extent?* Interestingly, we observed that `FedBC`, with the help of $\gamma_i > 0$, tends to weight the importance of each device equally and hence achieves *fairness* as defined by Li et al. (2019). We specifically test the fairness of the global model in terms of its



(a) FedBC (b) FedAvg (c) q-FedAvg (d) FedProx (e) Scaffold

Figure 4: Test accuracy of the global model on min and max (defined in Figure 3 caption) users' local data at each round of communication for $E = 5$. The global model of FedBC (a) initially has a performance gap on min and max users' data, but the gap is largely eliminated in the end. For FedAvg (b), q-FedAvg (e), FedProx (f), and Scaffold (g), this gap is more apparent and persistent throughout training.

performance on user/derive with minimum data (called min user) and device with maximum data (called max user) in the experiments. Please refer to Appendix E for additional detailed experiments.

Table 1: Synthetic dataset classification global test accuracy for the different numbers of local training epochs, i.e., $K_i = E$ for each device i in Algorithm 1. The $\pm$ shows the standard deviation.

Algorithm	Epochs				
	1	5	10	25	50
FedAvg	83.61 $\pm$ 0.43	83.42 $\pm$ 0.58	83.49 $\pm$ 0.70	83.73 $\pm$ 0.57	82.94 $\pm$ 0.66
q-FedAvg	87.12 $\pm$ 0.25	86.76 $\pm$ 0.15	86.46 $\pm$ 0.28	86.76 $\pm$ 0.12	86.71 $\pm$ 0.12
FedProx	86.23 $\pm$ 0.42	85.59 $\pm$ 0.37	85.34 $\pm$ 0.61	85.00 $\pm$ 0.63	85.34 $\pm$ 0.48
Scaffold	83.84 $\pm$ 0.09	82.95 $\pm$ 0.30	83.48 $\pm$ 0.21	83.60 $\pm$ 0.21	82.80 $\pm$ 0.62
FedBC	**87.83 $\pm$ 0.35**	**87.48 $\pm$ 0.18**	**87.43 $\pm$ 0.20**	**86.99 $\pm$ 0.11**	**87.26 $\pm$ 0.12**

Experiment Setup. The synthetic dataset is associated with a 10-class classification task, and is adapted from (Li et al., 2020) with parameters α and β controlling model and data variations across users (see Appendix E.1 for details). For real datasets, we use MNIST and CIFAR-10 for image classification. MNIST and CIFAR-10 datasets consist of handwritten digits and color images from 10 different classes respectively (Krizhevsky et al., 2009) (LeCun et al., 1998). We denote C to represent the most common number of classes in users' local data (see Appendix E.3 for details). To evaluate the global performance of FedBC, we compare it with 4 other FL algorithms, i.e., FedAvg (McMahan et al., 2017), q-FedAvg (Li et al., 2019), FedProx (Li et al., 2020), and Scaffold (Karimireddy et al., 2020). We use the term *global accuracy* while reporting the performance of the global model ($\mathbf{z}^t$) on the entire test dataset and use the term *local accuracy* while reporting the performance of each device's local model ($\mathbf{x}_i^t$) using its own test data and take the average across all devices.

Selection of γ_i: Since we do not know the optimal γ_i for each user i apriori, for experiments, we initialize them to be 0, and propose a heuristic to let the device decide its own γ_i. To achieve that, we observe γ_i participating in the Lagrangian defined in (8) and which defines a loss with respect to primal variables, and we want to minimize it. Hence, we take the derivative of the Lagrangian in (8) with respect to γ_i, and perform a gradient-descent update for γ_i. Interestingly, we note that the derivative of γ_i is $-\lambda_i$, which means that γ_i tends to always increase when gradient descent is performed. This implies that initially, each device's local model remains closer to the global model (similar to standard FL), but gradually incentivizes moving away from the global model to improve the overall performance. Experimentally, we show that this heuristic works very well in practice. (see Appendix E.1 for additional details).

Synthetic Dataset Experiments: We start by presenting the global accuracy results of the synthetic dataset classification task in Table 1. Note that FedBC outperforms all other algorithms for different numbers of local training epochs E. Figure 3 provides empirical justifications for such remarkable performance. Figure 3a shows that FedBC achieves a lower training loss than others, whereas algorithms such as FedAvg plateaus at an early stage. Next, to understand the calibrating behavior of FedBC, we compare the contributions from min and max users' local models (defined in Figure 3 caption) in updating the global model $\mathbf{z}^t$ via (13). To this end, we plot their λ_is at the end of each communication round in Figure 3b. It is evident that FedBC is

Table 2: Global model performance of FedBC and other baselines on CIFAR-10 classification ($E = 5$). All colored cells denote the proposed algorithms.

Classes	Algorithm	Power Law Exponent				
		1.1	1.2	1.3	1.4	1.5
C = 1	FedAvg	50.15 ± 0.58	50.66 ± 1.88	57.23 ± 0.74	53.69 ± 2.24	60.82 ± 0.85
	q-FedAvg	49.17 ± 1.38	49.32 ± 1.64	57.35 ± 1.05	54.40 ± 1.57	60.56 ± 1.19
	FedProx	49.92 ± 1.16	50.21 ± 2.00	57.14 ± 0.60	56.30 ± 1.95	60.57 ± 0.65
	Scaffold	46.86 ± 2.03	36.55 ± 2.63	37.81 ± 2.50	30.99 ± 1.23	36.08 ± 3.25
	Per-FedAvg	45.93 ± 0.86	37.03 ± 6.22	56.43 ± 2.35	52.82 ± 1.98	56.31 ± 5.52
	pFedMe	47.18 ± 1.28	43.69 ± 1.38	50.76 ± 1.44	45.12 ± 1.92	50.19 ± 2.24
	FedBC	**50.35 ± 0.91**	**55.25 ± 1.27**	**58.93 ± 1.52**	**58.10 ± 1.56**	**61.12 ± 1.24**
C = 2	FedAvg	57.10 ± 0.85	56.94 ± 1.66	57.67 ± 2.76	32.87 ± 2.82	58.59 ± 3.14
	q-FedAvg	**57.20 ± 0.68**	**58.29 ± 1.67**	57.71 ± 2.09	57.10 ± 2.44	57.90 ± 3.27
	FedProx	57.18 ± 1.21	57.64 ± 1.64	57.98 ± 1.73	39.99 ± 4.18	58.63 ± 2.91
	Scaffold	55.51 ± 1.79	24.48 ± 3.34	36.69 ± 2.78	41.79 ± 0.42	32.18 ± 9.18
	Per-FedAvg	55.29 ± 0.82	54.67 ± 1.88	54.64 ± 1.96	39.66 ± 6.94	60.16 ± 1.46
	pFedMe	51.45 ± 0.44	46.80 ± 2.04	47.98 ± 1.46	30.81 ± 3.22	49.25 ± 1.68
	FedBC	56.45 ± 1.05	55.64 ± 1.45	**58.02 ± 2.10**	**60.92 ± 2.43**	**64.20 ± 2.13**
C = 3	FedAvg	64.19 ± 2.06	53.83 ± 3.38	57.54 ± 1.17	60.96 ± 0.95	58.91 ± 2.76
	q-FedAvg	62.76 ± 1.53	61.37 ± 2.06	61.80 ± 1.73	63.40 ± 2.16	64.01 ± 1.96
	FedProx	**64.40 ± 1.80**	54.81 ± 2.16	57.28 ± 1.90	61.66 ± 0.42	59.85 ± 2.81
	Scaffold	61.46 ± 1.86	54.04 ± 6.99	38.38 ± 2.80	30.75 ± 5.68	33.04 ± 9.85
	Per-FedAvg	61.83 ± 1.74	52.24 ± 1.84	58.16 ± 0.61	60.36 ± 1.96	59.27 ± 2.46
	pFedMe	53.52 ± 1.59	42.92 ± 1.64	52.50 ± 1.79	54.39 ± 1.91	53.26 ± 1.24
	FedBC	63.43 ± 2.55	**62.90 ± 2.26**	**64.87 ± 1.44**	**66.23 ± 0.65**	**66.80 ± 1.07**

Table 3: Local model performance of FedBC and other baselines on CIFAR-10 classification ($E = 5$). All colored cells denote the proposed algorithms (see Algorithm 3 in the appendix for Per-FedBC).

Classes	Algorithm	Power Law Exponent				
		1.1	1.2	1.3	1.4	1.5
C = 1	Per-FedAvg	**99.92 ± 0.15**	**85.58 ± 0.66**	**94.99 ± 0.37**	91.10 ± 1.90	85.09 ± 1.42
	pFedMe	86.28 ± 0.59	72.05 ± 0.35	81.21 ± 0.27	82.46 ± 0.65	76.94 ± 0.82
	FedBC	91.96 ± 0.70	77.52 ± 0.60	84.87 ± 1.01	86.15 ± 0.43	81.31 ± 0.22
	FedBC -FineTune	97.36 ± 0.22	84.54 ± 0.17	91.89 ± 0.44	87.18 ± 0.21	82.01 ± 0.17
	Per-FedBC	99.39 ± 1.09	85.42 ± 0.75	93.79 ± 1.57	**91.15 ± 2.10**	**85.18 ± 1.08**
C = 2	Per-FedAvg	**93.45 ± 0.28**	86.29 ± 0.93	87.38 ± 1.19	62.01 ± 5.60	86.28 ± 1.32
	pFedMe	73.16 ± 0.45	68.01 ± 0.55	69.76 ± 0.68	49.97 ± 0.58	60.16 ± 1.50
	FedBC	78.04 ± 0.57	73.24 ± 0.47	74.95 ± 0.36	53.83 ± 0.56	71.22 ± 1.21
	FedBC -FineTune	89.27 ± 0.29	77.64 ± 0.28	79.69 ± 0.11	56.83 ± 0.34	80.63 ± 0.51
	Per-FedBC	93.09 ± 0.41	**86.55 ± 1.76**	**88.47 ± 2.11**	**70.20 ± 4.60**	**87.47 ± 1.12**
C = 3	Per-FedAvg	**85.79 ± 0.87**	75.69 ± 2.30	89.14 ± 0.71	90.98 ± 3.37	81.63 ± 3.32
	pFedMe	57.93 ± 1.05	47.84 ± 1.17	70.16 ± 0.60	68.76 ± 0.91	59.71 ± 0.50
	FedBC	60.31 ± 0.74	52.77 ± 1.34	73.09 ± 0.85	74.56 ± 0.90	65.05 ± 1.00
	FedBC -FineTune	74.42 ± 0.29	67.46 ± 0.59	90.08 ± 0.42	88.84 ± 0.82	72.60 ± 0.34
	Per-FedBC	85.61 ± 1.18	**80.96 ± 2.50**	**91.22 ± 0.95**	**91.85 ± 1.54**	**83.40 ± 1.41**

significantly less biased towards the min user. The min user's coefficient eventually catches that of the max user for FedBC, and the difference between them is one order of magnitude less than that of the data-size-based coefficient. This enables FedBC to be *better in terms of fairness* as compared to other algorithms. Figure 3c shows the distribution of λ- coefficients at the end of training for devices of different data sizes. Interestingly, the max user's coefficient is almost the same as those of users of small data sizes. In fact, the coefficients of users of data size less than 300 for FedBC are consistently larger than their data-size-based counterparts, and vice versa for data sizes greater than 300. Lastly, Figure 3d shows the model of FedBC achieves a high uniformity in test accuracy over users' local data at different values of E.

To further emphasize the fairness aspect of FedBC, we plot the test accuracy of the global model on min and max users' local data throughout the entire communication in Figure 4. We observe that the global model performs better on max user's data than on min user's data in general for all algorithms. Most importantly, FedBC demonstrates a superior advantage in reducing this performance gap. After 100 rounds of communication, test accuracy for min and max users are nearly the same, as shown in Figure 4a. Whereas for FedAvg (Figure 4b), this gap can be as large as 100% even after nearly 200 rounds of communication. As compared to q-Fedavg (Figure 4c), FedProx (Figure 4d), or Scaffold (Figure 4e), FedBC has a much higher

fraction of points at which test accuracy for min and max user overlap, which indicates that they are being treated equally well (*enforcing fairness*). We also present additional results in Appendix F (Figure 10-12) for $E = 25, 50$, because the local model differs more from the global model as E increases. The trend is similar, and FedBC can still make good predictions on the min user's local data despite an increase in the performance gap compared to $E = 5$. This is significantly different from the case of FedAvg, in which its model fails to make any correct predictions on the min user's local data for the majority of times, as shown in Figure 10-12 in Appendix F. In essence, `FedBC` has the best performance because it allows users to participate fairly in updating the global model. This performance benefit is credited to using non-zero γ_i, which is the main contribution of this work.

Real Dataset Experiments: The experiments on real datasets are in line with our previous observations in Figure 3. We first report the results for global model performance on the CIFAR-10 dataset in Table 2 for $E = 5$ (see Appendix 6 for $E = 1$). We note that FedBC outperforms FedAvg by 7.89% for $C = 3$. For the most challenging situation of $C = 1$, `FedBC` outperforms all other baselines. The superior performance of `FedBC` is attributed to the fact that we observe unbiasedness in computing the coefficient for the global model when $C = 1$, 2, or 3 (see Figure 13 in Appendix G.1). We also observe the high uniformity in test accuracy over users' local data for all classes with different power law exponents (see Figure 15 in Appendix G.1).

We show the classification results on MNIST in Table 7 and Table 8 in Appendix G.2 and observe that `FedBC` outperforms all the other baselines. We also present the results on the Shakespeare dataset in Table 9 in Appendix G.3. From Table 2, we also notice the global performance of pFedMe and Per-FedAvg is worse than that of FedAvg, FedProx or `FedBC` when $C = 1$. This is mainly because personalized algorithms are designed to optimize the local objective of (2). However, this may create conflicts with the global objective in (1) and lead to poor global performance.

Local Performance. We have established that a non-zero γ_i in `FedBC` leads to obtaining a better global model. This is because it provides additional freedom to automatically decide the contributions of local models rather than sticking to a uniform averaging, as done in existing FL methods. But a remark regarding the individual performance of local models $\mathbf{x}_i^t$ is due. We can evaluate the test accuracy of local model $\mathbf{x}_i$ at each device i to see how it performs with respect to local test data. Table 3 presents the local performance of `FedBC` and other personalization algorithms. We observe that `FedBC` performs better and worse than pFedMe and Per-FedAvg, respectively. We can expect this performance because our algorithm is not designed to just focus on improving the local performance compared to pFedMe and Per-FedAvg. But an interesting point is that we can utilize the local models $\mathbf{x}_i^t$ obtained by `FeBC` to act as a good initializer, and after doing some fine tuning at device i, can improve the local performance as well. For instance, by doing 1-step fine-tuning on the test data (we call it `FedBC`-FineTune), we can improve the local performance of `FedBC`. For example, `FedBC`-FineTune achieves a 7.55% performance increase over `FedBC` when $C = 3$. To further improve the local model performance, as an additional experimental study, we incorporated MAML-type training into `FedBC` (cf. Algorithm 3 in Appendix E.4), which we call Per-`FedBC` algorithm, we can significantly improve the local performance for both $C = 1$ and $C = 3$.

Takeaways: In summary, we experimentally show that `FedBC` has the best global performance when compared against all baselines (addresses ①). Moreover, `FedBC` has reasonably good local performance but can be improved by fine-tuning or performing MAML-type training (addresses ②). We leave the question of how to fully exploit the advantage in the freedom of choosing γ_i to achieve personalization for future work.

6 Conclusions

In this work, we delved into the intricate relationship between local and global model performance in federated learning (FL). We introduced a new proximity constraint to the FL framework, enabling the automatic determination of local model contributions to the global model. Our research demonstrates that by recognizing the flexibility to not force consensus among local models, we can simultaneously improve both the global and local performance of FL algorithms. Building on this insight, we developed the novel `FedBC` algorithm, which has been shown to perform well across a broad range of synthetic and real data sets. It outperforms state-of-the-art methods by automatically calibrating local and global models efficiently across devices.

References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. In *International Conference on Learning Representations*, 2021. 2, 14

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. 2, 4

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in Neural Information Processing Systems*, 33:3557–3568, 2020. 2, 3, 14, 24, 25, 26

Chen Fan, Parikshit Ram, and Sijia Liu. Sign-maml: Efficient model-agnostic meta-learning by signsgd. *arXiv preprint arXiv:2109.07497*, 2021. 26

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020. 5

Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtárik. Lower bounds and optimal algorithms for personalized federated learning. *Advances in Neural Information Processing Systems*, 33:2304–2315, 2020. 2

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020. 1, 2, 6, 8, 14, 24

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 8, 24

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8, 24

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*, 2021a. 24

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. In *International Conference on Learning Representations*, 2019. 1, 7, 8, 24, 25

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. 1, 2, 4, 5, 6, 8, 14, 24

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International Conference on Machine Learning*, pp. 6357–6368. PMLR, 2021b. 2, 5

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*, pp. 1273–1282. PMLR, 2017. 1, 2, 3, 4, 5, 6, 8, 14, 24

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009. 2, 3

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009. 2, 4

Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010. 2, 3

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009. 6

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015. 2

Dimitris Stripelis and José Luis Ambite. Accelerating federated learning in heterogeneous data and computational environments. *arXiv preprint arXiv:2008.11281*, 2020. 24, 25

Canh T Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405, 2020. 2, 6, 7, 14, 24

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. 3, 14

Håkan Terelius, Ufuk Topcu, and Richard M Murray. Decentralized multi-agent optimization via dual decomposition. *IFAC proceedings volumes*, 44(1):11245–11251, 2011. 2

Han Wang, Siddartha Marella, and James Anderson. Fedadmm: A federated primal-dual algorithm allowing partial participation. *arXiv preprint arXiv:2203.15104*, 2022. 2

Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999. 5

Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–6070, 2021. 2, 5, 7, 14