`FedDANE`: A Federated Newton-Type Method

Tian Li[†]   Anit Kumar Sahu[‡]   Manzil Zaheer[*]   Maziar Sanjabi[¶]   Ameet Talwalkar[†§]   Virginia Smith[†]

[†]Carnegie Mellon University [‡]Bosch Center for AI [*]Google Research [¶]University of Southern California [§]Determined AI

[†]{tianli, talwalkar, smithv}@cmu.edu, [‡]anit.sahu@gmail.com, [*]manzilz@google.com, [¶]maziar.sanjabi@gmail.com

Abstract—**Federated learning aims to jointly learn statistical models over massively distributed remote devices. In this work, we propose `FedDANE`, an optimization method that we adapt from `DANE` [9, 10], a method for classical distributed optimization, to handle the practical constraints of federated learning. We provide convergence guarantees for this method when learning over both convex and non-convex functions. Despite encouraging theoretical results, we find that the method has underwhelming performance empirically. In particular, through empirical simulations on both synthetic and real-world datasets, `FedDANE` consistently underperforms baselines of `FedAvg` [8] and `FedProx` [6] in realistic federated settings. We identify low device participation and statistical device heterogeneity as two underlying causes of this underwhelming performance, and conclude by suggesting several directions of future work.**

I. INTRODUCTION

Federated learning is a distributed learning paradigm that considers training statistical models in heterogeneous networks of remote devices [5, 8]. Learning a model while keeping data localized can provide both computational and privacy benefits compared to transmitting raw data across the network.

To handle heterogeneity and high communication costs in federated networks, a popular approach for federated optimization methods involves allowing for local updating and low participation [5]. One method along these lines is `FedAvg` [8], which has demonstrated robust empirical performance in non-convex settings. `FedAvg` assumes only a small subset of devices (e.g., 1% out of thousands to millions) participate in training at each communication round. Each selected device then performs variable amounts of local work before sending model updates back to the server, which can enable a flexible trade-off between communication and computation.

Although `FedAvg` performs well empirically, it can diverge when the data is statistically heterogeneous (i.e., generated in a non-identically distributed manner across the network) [6, 8]. A recent approach, `FedProx` [6], has attempted to mitigate this issue by adding a proximal term to the subproblem on each device, which helps to improve the stability of the method.

In this work, we take a similar approach to `FedProx`, and draw inspiration from `DANE` and variants [9, 10], which are popular methods developed for the distributed data center setting. In particular, Reddi et al. [9] propose inexact-`DANE`, a variant of `DANE` that allows for local updating, which is beneficial when communication is a bottleneck. Compared with `FedAvg`, `DANE` and inexact-`DANE` use a different local subproblem which includes two additional terms—a gradient correction term and a proximal term. As data is statistically heterogeneous in federated networks, these terms can potentially

improve convergence by forcing model updates to be closer to the current global model, making the method more stable and amenable to theoretical analysis. Including the gradient correction term also allows the update to take on the form of an approximate Newton-type method, which can lead to provably improved convergence for certain well-behaved objectives [10].

Despite the merits of (inexact) `DANE`, the method has not been analyzed in settings with statistically heterogeneous data or low participation amongst the devices, which are critical challenges in realistic federated networks. Indeed, at each communication round, `DANE` requires every device to collectively evaluate the gradient of the global function. This is prohibitive in federated networks as it requires the server to communicate with each device in a potentially massive network, and does not allow for the case of devices dropping out. A natural way to address this issue is to approximate the gradient via a subsample of the devices. Based on this idea, we propose `FedDANE`, a variant of inexact `DANE` for federated learning.[1] Similar to inexact `DANE`, `FedDANE` inexactly solves an approximate Newton-type subproblem, but only collects gradient updates from a subset of devices at each round.

We provide convergence guarantees for `FedDANE` for both convex and non-convex functions in low participation settings, and allow for the scenario that each device generates data from a possibly differing distribution. Despite encouraging theoretical results, our empirical evaluation indicates that while `FedDANE` is more expensive as it needs two rounds of communication for one update, it consistently underperforms `FedAvg` and `FedProx` due to the inexact estimation of the full gradient and the statistical heterogeneity in the network. Our study highlights the drawbacks of the gradient correction term in `FedDANE`, and suggests the superiority of `FedProx` which leverages just the proximal term to achieve improved performance for federated optimization. Our work also suggests several directions of future work in federated optimization.

II. RELATED WORK

`DANE` and Other Communication-efficient Distributed Methods. Methods that employ local updating (i.e., computing and applying a variable number of updates locally, rather than just evaluating the gradients once and sending them back for aggregation) are a popular approach for improving communication-efficiency in distributed optimization. By

---

[1]We note that the gradient correction term in `FedDANE` was explored briefly in prior work of `FedProx` [6] (Appendix B), though this work is the first to theoretically analyze `FedDANE` and provide a systematic evaluation of the method in federated settings.

solving the local subproblems inexactly at each round, such schemes enable a flexible trade-off between communication and computation. For example, COCOA [12] is a communication-efficient primal-dual framework that leverages duality to decompose the global objective into subproblems that can be solved inexactly. Several primal methods [e.g., 9, 10, 13, 14, 17, 18], including DANE [10] and inexact DANE [9], have also been proposed, and have the added benefit of being applicable to non-convex objectives. While these methods make a seemingly small change over standard mini-batch methods, they enable drastically improved performance in practice, and have been shown to achieve orders-of-magnitude speedups over mini-batch methods in real-world data center environments. This is especially critical in communication-constrained environments such as federated settings.

Heterogeneity-aware Federated Optimization. An important distinction between federated optimization and classical distributed optimization is the presence of *heterogeneity*, i.e., non-identically distributed data and heterogeneous systems across the network. Smith et al. [11] propose a primal-dual optimization method that learns separate but related models for each device through a multi-task learning framework. This setup naturally captures statistical heterogeneity, and also considers systems issues such as stragglers in the method and theory. However, such an approach is not generalizable to non-convex problems. There are several recent works that provide theoretical analysis specifically for federated optimization. FedProx [6] characterizes the convergence behavior under a dissimilarity assumption of local functions, while accounting for the low participation of devices. Other works analyze different methods with non-identically distributed data, but under different (possibly) limiting assumptions, such as using SGD as a specific local solver [7], full device participation [15, 16], convexity [4, 7, 15], or uniformly-bounded gradients [7, 16]. For instance, SCAFFOLD [4] is a recent method for federated optimization related to DANE where it maintains a similar gradient correction term in the local subproblem. However, its convergence results are limited to strongly convex functions, and the method has yet to be explored empirically. Our convergence analysis of FedDANE also accounts for low device participation and data heterogeneity, and covers both convex and non-convex functions (Section IV).

III. METHODS

In this section, we propose FedDANE, a heterogeneity-aware federated optimization method. Before introducing FedDANE (Section III-C), we first formally define the optimization objective we consider in this paper (Section III-A), and provide some background on FedAvg and DANE (Section III-B).

A. Problem Setup

Federated learning typically aims to minimize the empirical risk over heterogeneous data distributed across multiple devices:

$$\min_w \ f(\mathbf{w}) = \sum_{k=1}^{N} p_k F_k(\mathbf{w}) = \mathbb{E}_k[F_k(\mathbf{w})], \qquad (1)$$

where N is the number of devices, $p_k \geq 0$, and $\sum_k p_k = 1$. In general, the local objectives measure the local empirical risk over possibly differing data distributions $\mathcal{D}_k$, i.e., $F_k(w) := \mathbb{E}_{x_k \sim \mathcal{D}_k} f_k(w; x_k)$, with n_k samples available at each device k. Hence, we can set $p_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ is the total number of data points on all devices. In this work, we consider the typical centralized setup where N devices are connected to one central server.

B. Preliminaries: FedAvg and DANE

In FedAvg [8], a subset of devices are sampled, and perform variable iterations of SGD to solve their local subproblems inexactly at each communication round. In particular, each selected device k runs E epochs of SGD on the local function F_k to obtain local updates, then sends the updates back for aggregation in a synchronous manner. The details are summarized in Algorithm 1.

Algorithm 1 Federated Averaging (FedAvg)

1: **Input:** K, T, η, E, $\mathbf{w}^0$, N, p_k, $k = 1, \cdots, N$
2: **for** $t = 1, \cdots, T$ **do**
3:     Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)
4:     Server sends $\mathbf{w}^{t-1}$ to all chosen devices
5:     Each device $k \in S_t$ updates $\mathbf{w}^{t-1}$ for E epochs of SGD on F_k with step-size η to obtain $\mathbf{w}_k^t$
6:     Each device $k \in S_t$ sends $\mathbf{w}_k^t$ back to the server
7:     Server aggregates the $\mathbf{w}$'s as $\mathbf{w}^t = \frac{1}{K} \sum_{k \in S_t} \mathbf{w}_k^t$
8: **end for**

In data center settings, DANE [10] and its inexact variants [9] are another set of approaches which have been analyzed in depth. In its simplest form, DANE has each worker k solve the following subproblem:

$$\mathbf{w}_k^t = \operatorname*{argmin}_{\mathbf{w}} F_k(\mathbf{w}) + \left\langle \nabla f(\mathbf{w}^{t-1}) - \nabla F_k(\mathbf{w}^{t-1}), \mathbf{w} - \mathbf{w}^{t-1} \right\rangle$$
$$+ \frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}^{t-1} \right\|^2. \qquad (2)$$

Similarly, after each worker solves its subproblem, the central server collects those updates and aggregates them to obtain $\mathbf{w}^t$. The update is in fact a two-step process, as (2) requires the workers to first collectively compute the overall gradient of the function, $\nabla f(\mathbf{w}^{t-1})$, and can be interpreted as a distributed variant of SVRG [9]. Inexact DANE allows the flexibility of solving (2) inexactly [9]. Based on inexact DANE, we next introduce FedDANE.

C. Proposed Method: FedDANE

The inexact DANE method mentioned above cannot be directly applied to federated settings. One critical challenge is that computing the full gradient $\nabla f(\mathbf{w}^{t-1})$ requires the server to communicate with all the devices and then average the local gradients, which is infeasible in massive federated networks.

In FedDANE, we propose to approximate the full gradients using a subset of gradients from randomly sampled devices.

Algorithm 2 Proposed method: `FedDANE`

1: **Input:** K, T, η, E, $\mathbf{w}^0$, N, p_k, $k = 1, \cdots, N$
2: **for** $t = 1, \cdots, T$ **do**
3:     Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)
4:     Server sends $\mathbf{w}^{t-1}$ to all chosen devices
5:     Each selected device computes $\nabla F_k(\mathbf{w}^{t-1})$ and sends it to the central server
6:     The server aggregates the gradients into
$$g_t = \frac{1}{K} \sum_{k \in S_t} \nabla F_k(\mathbf{w}^{t-1})$$
7:     Server selects another subset S_t' of K devices at random; each device $k \in S_t'$ solves the following subproblem inexactly to obtain $\mathbf{w}_k^t$:
$$\mathbf{w}_k^t = \underset{\mathbf{w}}{\arg\min} \, F_k(\mathbf{w}) + \left\langle g_t - \nabla F_k(\mathbf{w}^{t-1}), \mathbf{w} - \mathbf{w}^{t-1} \right\rangle$$
$$+ \frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}^{t-1} \right\|^2$$
8:     Each device $k \in S_t'$ sends $\mathbf{w}_k^t$ back to the server
9:     Server aggregates the $\mathbf{w}$'s as $\mathbf{w}^t = \frac{1}{K} \sum_{k \in S_t} \mathbf{w}_k^t$
10: **end for**

Collecting the gradients from a subset S_t ($|S_t| = K$) of devices at each iteration t yields:

$$\nabla f(\mathbf{w}^{t-1}) \approx g_t = \frac{1}{K} \sum_{k \in S_t} \nabla F_k(\mathbf{w}^{t-1}).$$

After computing g_t, `FedDANE` selects another subset of devices where each device $k \in S_t$ solves the following subproblem inexactly:

$$\mathbf{w}_k^t = \underset{\mathbf{w}}{\arg\min} \, F_k(\mathbf{w}) + \left\langle g_t - \nabla F_k(\mathbf{w}^{t-1}), \mathbf{w} - \mathbf{w}^{t-1} \right\rangle$$
$$+ \frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}^{t-1} \right\|^2. \quad (3)$$

The server then aggregates the updates from the selected devices. See Algorithm 2 for details. We note that one limitation of `FedDANE` is that each outer iteration incurs two rounds of communication, making it less efficient than `FedAvg` and `FedProx`. This leads us to suggest a variant of `FedDANE` leveraging a pipelined approach to perform one update in a single round of communication (see Section V-C for more discussions). However, as we will see in our empirical valuation (Section V), even the less efficient (and more accurate) `FedDANE` proposed here results in inferior practical performance compared to `FedAvg` and `FedProx`.

IV. ANALYSIS

We now provide our convergence analysis of `FedDANE` for both convex and non-convex problems. Recall that `FedDANE` allows each selected device to solve a subproblem inexactly at each updating round to reduce communication. We first formally define a parameter γ to quantify the inexactness, which will be used throughout our analysis.

Definition 1 (γ-inexact Solution)**.** We say that $\mathbf{w}^t$ is a γ-inexact minimizer of (3) if $\|\mathbf{w}^t - \underline{\mathbf{w}}^t\| \leq \gamma \|\underline{\mathbf{w}}^t - \mathbf{w}^{t-1}\|$, where $\gamma \in [0, 1)$, and $\underline{\mathbf{w}}^t$ is the exact minimizer of (3). Note that a smaller γ corresponds to higher accuracy.

In order to quantify the dissimilarity between devices in a federated network, following Li et al. [6], we define B-local dissimilarity as follows.

Definition 2 (B-local Dissimilarity)**.** The local functions F_k are B-locally dissimilar at $\mathbf{w}$ if $\mathbb{E}_k \|\nabla F_k(\mathbf{w})\|^2 \leq \|\nabla f(\mathbf{w})\|^2 B^2$. We further define $B(\mathbf{w}) = \sqrt{\frac{\mathbb{E}_k \|\nabla F_k(\mathbf{w})\|^2}{\|\nabla f(\mathbf{w})\|^2}}$ for $\|\nabla f(\mathbf{w})\| \neq 0$.

When the devices are homogeneous with I.I.D. data, $B(\mathbf{w}) = 1$ for every $\mathbf{w}$. The more heterogeneous the data are in the network, the larger $B(\mathbf{w})$ is. As discussed later, our convergence results are a function of the device dissimilarity bound B.

A. Convex Case

We first investigate the convergence results for convex F_k's.

Theorem 3 (Sufficient Decrease)**.** *Assume F_k's are convex, and have L-Lipschitz continuous gradients. Moreover, assume B-dissimilarity is bounded by B at point $\mathbf{w}^{t-1}$. Given the inexact criterion in Definition 1, if μ, γ, L, and B satisfy*

$$\rho = \left(\frac{2 - 3\gamma}{2\mu} - \frac{2L(1+\gamma)^2 + 3L}{2\mu^2} \right.$$
$$\left. - (B^2 - 1) \left(\frac{L(1+\gamma)^2 + L}{\mu^2} + \frac{\gamma}{\mu} \right) \right) > 0,$$

then at iteration t of Algorithm 2, we have the following expected decrease in the global objective:

$$\mathbb{E}_{S_t} \left[f(\mathbf{w}^t) \right] \leq f(\mathbf{w}^{t-1}) - \rho \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2,$$

where S_t represents the distribution of a set of random devices selected at time t.

We defer the readers to Appendix V-D for a complete proof. At a high-level, we first use the γ-inexactness and other assumptions to attain a decrease in the objective, then take an expectation over randomly selected devices and apply the bounded B-dissimilarity to obtain the above results.

Corollary 4 (Convergence: Convex Case)**.** *Let the assertions of Theorem 3 hold. In addition, let $\gamma = 0$, i.e., all the local problems are solved exactly, if $1 \ll B$ then we choose $\mu \approx 5LB^2$ from which it follows that $\rho \approx \frac{3}{25LB^2}$.*

B. Non-convex Case

We have the following convergence characterization for non-convex functions.

Theorem 5 (Sufficient Decrease)**.** *Assume F_k's are non-convex, and have L-Lipschitz continuous gradients. Moreover, assume there exists a λ such that $\lambda \mathbf{I} + \nabla^2 F_k(w) \succ 0$, with $\mu - \lambda > 0$.*

Assume B-dissimilarity is bounded by B at point $\mathbf{w}^{t-1}$. Given the inexact criterion in Definition 1, if μ, γ, L, and B satisfy

$$\rho = \left(\frac{1}{\mu} - \frac{3\gamma}{2(\mu - \lambda)} - \frac{L(1+\gamma)^2}{(\mu - \lambda)^2} - \frac{3L}{2\mu(\mu - \lambda)} \right.$$
$$\left. - (B^2 - 1) \left(\frac{L(1+\gamma)^2}{(\mu - \lambda)^2} + \frac{L}{\mu(\mu - \lambda)} + \frac{\gamma}{\mu - \lambda} \right) \right) > 0$$

then at iteration t of Algorithm 2, we have the following expected decrease in the global objective:

$$\mathbb{E}_{S_t}[f(\mathbf{w}^t)] \le f(\mathbf{w}^{t-1}) - \rho \|\nabla f(\mathbf{w}^{t-1})\|^2,$$

where S_t represents the devices randomly selected at time t.

The proof (Appendix V-E) is similar to the proof for Theorem 3. Now we can use the above sufficient decrease to the characterize the rate of convergence to the set of approximate stationary solutions $\{w \mid \mathbb{E}\left[\|\nabla f(\mathbf{w}^t)\|^2\right] \le \epsilon\}$.

Theorem 6 (Convergence: Non-convex Case). *Let the assumption Theorem 5 hold at each iteration of* `FedDANE`*. Moreover, $f(\mathbf{w}^0) - f^* = \Delta$. Then, after $T = O(\frac{\Delta}{\rho\epsilon})$ iterations, we have $\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla f(\mathbf{w}^t)\|^2\right] \le \epsilon$.*

Note that the convergence rates of `FedDANE` derived here recover the results in `FedProx` [6], which are also asymptotically the same as SGD [3].

C. Device-specific Constants

While the previous results assume the same constants L (the Lipschitz constant of gradients), μ (the penalty constant of the proximal term), and γ (the degree of inexactness) across all devices, we can easily extend the analysis to allow for variable constants across devices.

Theorem 7 (Convergence with Device-specific Constants). *Assume F_k's are convex, and have L_k-Lipschitz continuous gradients. Moreover, assume B-dissimilarity is bounded by B at point $\mathbf{w}^{t-1}$. Given the inexact criterion in Definition 1, if constants μ_k, γ_k, L_k, and B are chosen such that*

$$\rho = \left(\frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{1}{\mu_k} - \frac{3\gamma_k}{2\mu_k} - \frac{L_k(1+\gamma_k)^2}{\mu_k^2} - \frac{3L_k}{2\mu_k^2} \right) \right.$$
$$\left. - \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{L(1+\gamma_k)^2}{\mu_k^2} + \frac{L_k}{\mu_k^2} + \frac{\gamma_k}{\mu_k} \right) (B^2 - 1) \right) > 0,$$

then at iteration t of Algorithm 2, we have the following expected decrease in the global objective:

$$\mathbb{E}_{S_t}[f(\mathbf{w}^t)] \le f(\mathbf{w}^{t-1}) - \rho \|\nabla f(\mathbf{w}^{t-1})\|^2,$$

where S_t represents the distribution of a set of random devices selected at time t.

See Appendix V-F for a full proof. Note that our analysis is general in that it is agnostic of any specific local solver, and covers both cases of sampling devices with and without replacement.

V. EXPERIMENTS

A. Experimental Setup

Datasets. We evaluate the performance of `FedDANE` using both synthetic and real-world federated datasets. The datasets are curated from the LEAF benchmark [2] as well as previous work on federated learning [6]. In particular, we use a set of synthetic datasets with varying degrees of data heterogeneity following the setup in Li et al. [6]. We also study three real datasets in LEAF: FEMNIST for image classification with a convex model, Shakespeare for next-character prediction, and Sent140 for sentiment analysis, both with non-convex deep neural network models. These datasets are naturally partitioned into different devices in the network [2]. Data statistics are summarized in Table I below.

TABLE I: Statistics of three real federated datasets.

Datasets	# Devices	# Samples	# Samples/device	
			mean	stdev
FEMNIST	200	18,345	92	159
Sent140	772	40,783	53	32
Shakespeare	143	517,106	3,616	6,808

Implementation & Hyper-parameters. We implement all code in Tensorflow [1], simulating a federated setup where N devices (N is the total number of devices shown in Table I) are connected with a central server. For `FedAvg` and `FedProx`, we directly take the tuned hyper-parameters reported in [6]. For `FedDANE`, we use the same learning rates and batch sizes as in `FedAvg` on the same dataset. We tune μ (the penalty constant in the proximal term) for `FedDANE` from a candidate set $\{0, 0.001, 0.01, 0.1, 1\}$ and pick a best μ based on the training loss. All code, data, and experiments are publicly available at `github.com/litian96/FedDANE`.

B. Evaluation Results

We compare the convergence of `FedDANE` with `FedAvg` and `FedProx`. For each method, we select 10 devices at each updating round, and let each device perform E epochs of local updates ($E = 20$). We plot the training loss versus the updating rounds (treating two communication rounds in `FedDANE` as one). The results are shown in Figure 1. We see that `FedDANE` consistently performs worse than both `FedAvg` and `FedProx`. This indicates that statistical heterogeneity and low device participation (the inaccurate approximation of the full gradients) may hurt the convergence of `FedDANE`. We further investigate the effects of varying participating devices and show that whether selecting more devices to get a better approximation of the full gradients can lead to improved performance depends on the degree of data heterogeneity. We then create an extreme 'unrealistic' setting that favors `FedDANE`, where we select a large subset of devices (78% of the total devices on average) and let each device perform only one epoch of local updates, trying to prevent local models from deviating too much from the global model. Even in this unrealistic setting, the performance of `FedDANE` is still
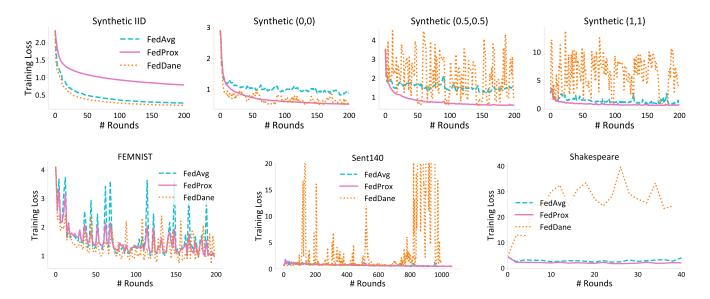
Fig. 1: Convergence of `FedDANE` compared with `FedAvg` and `FedProx`. For synthetic datasets in the first row, from left to right, data are becoming more heterogeneous. Except for the perfect I.I.D. dataset (Synthetic IID), `FedDANE` underperforms both `FedAvg` and `FedProx` on all datasets—either converging more slowly or diverging.

disappointing. The results of all the additional experiments are provided in Appendix V-G.

C. Discussions

Despite encouraging theoretical results, `FedDANE` demonstrates underwhelming empirical performance. This indicates that several of our theoretical assumptions may not hold in practical scenarios. These violations may include (1) the lowest eigenvalue of the Hessian $\nabla^2 F_k(w)$ is too small, (2) the choice of μ does not make the local subproblem strongly convex, and (3) the choices of the constants μ, γ, L and B may not guarantee sufficient decrease. More generally, the discrepancy between theory and practice suggests that the practical issues of low device participation and statistical heterogeneity in distributed optimization require careful theoretical consideration—for `FedDANE` as well as for methods such as `FedAvg` and `FedProx`. Developing a better understanding of this setting may help to enable improved empirical performance for the increasingly prevalent problem of federated learning.

We note that there are other possible variants of `DANE` that may address the drawbacks of `FedDANE`. For instance, in order to mitigate the negative effects of the gradient correction term, we can consider decaying this term over the optimization process. The 'decayed' `FedDANE` will eventually reduce to `FedProx` as the gradient correction term becomes closer to zero. Another limitation with the proposed `FedDANE` method is that it requires two rounds of communication for one update. One could imagine a 'pipelined' variant of `FedDANE` where the overall gradient and the local model updates are transmitted together to the server. In this way, however, the selected devices have to use the stale gradients for the gradient correction term in the local subproblem. Exploring such variants is an interesting direction of future research.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. K. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.

[2] S. Caldas, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. Leaf: A benchmark for federated settings. *arXiv:1812.01097*, 2018.

[3] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIOPT*, 2013.

[4] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. Scaffold: Stochastic

controlled averaging for on-device federated learning. *arXiv:1910.06378*, 2019.

[5] T. Li, A. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

[7] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of fedavg on non-iid data. In *ICLR*, 2020.

[8] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[9] S. J. Reddi, J. Konečnỳ, P. Richtárik, B. Póczós, and A. Smola. Aide: Fast and communication efficient distributed optimization. *arXiv:1608.06879*, 2016.

[10] O. Shamir, N. Srebro, and T. Zhang. Communication-efficient distributed optimization using an approximate newton-type method. In *ICML*, 2014.

[11] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar. Federated multi-task learning. In *NeurIPS*, 2017.

[12] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi. Cocoa: a general framework for communication-efficient distributed optimization. *JMLR*, 2018.

[13] S. U. Stich. Local sgd converges fast and communicates little. In *ICLR*, 2019.

[14] J. Wang and G. Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv:1808.07576*, 2018.

[15] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive federated learning in resource constrained edge computing systems. *J-SAC*, 2019.

[16] H. Yu, S. Yang, and S. Zhu. Parallel restarted sgd for non-convex optimization with faster convergence and less communication. In *AAAI*, 2018.

[17] S. Zhang, A. E. Choromanska, and Y. LeCun. Deep learning with elastic averaging sgd. In *NeurIPS*, 2015.

[18] F. Zhou and G. Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *IJCAI*, 2018.

D. Proof for Theorem 3

Proof. We have by the Lipschitz continuity of the gradients:

$$f(\mathbf{w}^t) \leq f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \mathbf{w}^{t-1} \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2$$

$$\leq f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \underline{\mathbf{w}}^t \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2 \tag{4}$$

By optimality conditions, we have that $\underline{\mathbf{w}}^t$ satisfies

$$\nabla F_k \left(\underline{\mathbf{w}}^t \right) + \mathbf{g}_t - \nabla F_k \left(\mathbf{w}^{t-1} \right) + \mu \left(\underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right) = 0. \tag{5}$$

We denote the local subproblem (3) as $P_t(w)$. We also note that, $P_t(\mathbf{w})$ is μ-strongly convex,

$$\mu \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| \leq \left\| \nabla P_t(\mathbf{w}^{t-1}) \right\| = \left\| \mathbf{g}_t \right\|. \tag{6}$$

We derive a bound for $\left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|$ next.

$$\left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\| \leq \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| + \left\| \underline{\mathbf{w}}^t - \mathbf{w}^t \right\| \leq (1 + \gamma) \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\|. \tag{7}$$

Using (5)-(6) in (4), we have,

$$f(\mathbf{w}^t) \leq f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \underline{\mathbf{w}}^t \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2$$

$$\leq f(\mathbf{w}^{t-1}) - \frac{1}{\mu} \langle \nabla f(\mathbf{w}^{t-1}), \nabla F_k \left(\underline{\mathbf{w}}^t \right) + \mathbf{g}_t - \nabla F_k \left(\mathbf{w}^{t-1} \right) \rangle + \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \mathbf{w}^t - \underline{\mathbf{w}}^t \right\| + \frac{L(1+\gamma)^2}{2\mu^2} \left\| \mathbf{g}_t \right\|^2$$

$$\leq f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1}) \mathbf{g}_t}{\mu} + \frac{L}{\mu} \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| + \gamma \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| + \frac{L(1+\gamma)^2}{2\mu^2} \left\| \mathbf{g}_t \right\|^2$$

$$\leq f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1}) \mathbf{g}_t}{\mu} + \frac{L}{2\mu^2} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right) + \frac{\gamma}{2\mu} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right) + \frac{L(1+\gamma)^2}{2\mu^2} \left\| \mathbf{g}_t \right\|^2. \tag{8}$$

Taking expectation with respect to the randomly chosen devices S_t yields

$$\mathbb{E}_{S_t} \left[f(\mathbf{w}^t) \right] \leq f(\mathbf{w}^{t-1}) - \left(1 - \frac{3\gamma}{2} \right) \frac{\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2}{\mu} + \left(\frac{2L(1+\gamma)^2}{2\mu^2} + \frac{3L}{2\mu^2} \right) \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2$$

$$+ \left(\frac{L(1+\gamma)^2}{\mu^2} + \frac{L}{\mu^2} + \frac{\gamma}{\mu} \right) \mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right], \tag{9}$$

where in the last step, we used the inequality that

$$\left\| \mathbf{g}_t \right\|^2 = \left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) + \nabla f(\mathbf{w}^{t-1}) \right\|^2 \leq 2 \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + 2 \left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2. \tag{10}$$

Note that

$$\mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right] = \mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \leq \mathbb{E}_k \left[\left\| \nabla F_k(\mathbf{w}^{t-1}) \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \leq (B^2 - 1) \left\| f(\mathbf{w}^{t-1}) \right\|^2. \tag{11}$$

Plugging into (9), we get

$$\mathbb{E}_{S_t} \left[f(\mathbf{w}^t) \right] \leq f(\mathbf{w}^{t-1}) - \rho \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2, \tag{12}$$

where

$$\rho = \frac{2 - 3\gamma}{2\mu} - \frac{2L(1+\gamma)^2 + 3L}{2\mu^2} - (B^2 - 1) \left(\frac{L(1+\gamma)^2 + L}{\mu^2} + \frac{\gamma}{\mu} \right). \tag{13}$$

$\square$

E. Proof for Theorem 5

Proof. We have by the Lipschitz continuity of the gradients:

$$f(\mathbf{w}^t) \le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \mathbf{w}^{t-1} \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \underline{\mathbf{w}}^t \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2 \tag{14}$$

By optimality conditions, we have that $\underline{\mathbf{w}}^t$ satisfies

$$\nabla F_k \left(\underline{\mathbf{w}}^t \right) + \mathbf{g}_t - \nabla F_k \left(\mathbf{w}^{t-1} \right) + \mu \left(\underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right) = 0. \tag{15}$$

We denote the local subproblem (3) as $P_t(w)$. We also note that, $P_t(\mathbf{w})$ is $(\mu - \lambda)$-strongly convex,

$$(\mu - \lambda) \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| \le \left\| \nabla P_t(\mathbf{w}^{t-1}) \right\| = \left\| \mathbf{g}_t \right\|. \tag{16}$$

Using (15)-(16) in (14), we have,

$$f(\mathbf{w}^t) \le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}^t - \underline{\mathbf{w}}^t \rangle + \frac{L}{2} \left\| \mathbf{w}^t - \mathbf{w}^{t-1} \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{1}{\mu} \langle \nabla f(\mathbf{w}^{t-1}), \nabla F_k \left(\underline{\mathbf{w}}^t \right) + \mathbf{g}_t - \nabla F_k \left(\mathbf{w}^{t-1} \right) \rangle + \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \mathbf{w}^t - \underline{\mathbf{w}}^t \right\| + \frac{L(1+\gamma)^2}{2(\mu - \lambda)^2} \left\| \mathbf{g}_t \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1}) \mathbf{g}_t}{\mu} + \frac{L}{\mu} \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| + \gamma \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}^t - \mathbf{w}^{t-1} \right\| + \frac{L(1+\gamma)^2}{2(\mu - \lambda)^2} \left\| \mathbf{g}_t \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1}) \mathbf{g}_t}{\mu} + \frac{L}{2\mu(\mu - \lambda)} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right)$$

$$+ \frac{\gamma}{2(\mu - \lambda)} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right) + \frac{L(1+\gamma)^2}{2(\mu - \lambda)^2} \left\| \mathbf{g}_t \right\|^2$$

$$\Rightarrow \mathbb{E}_{S_t} \left[f(\mathbf{w}^t) \right] \le f(\mathbf{w}^{t-1}) - \left(1 - \frac{3\gamma\mu}{2(\mu - \lambda)} \right) \frac{\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2}{\mu} + \left(\frac{L(1+\gamma)^2}{(\mu - \lambda)^2} + \frac{3L}{2\mu(\mu - \lambda)} \right) \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2$$

$$+ \left(\frac{L(1+\gamma)^2}{(\mu - \lambda)^2} + \frac{L}{\mu(\mu - \lambda)} + \frac{\gamma}{\mu - \lambda} \right) \mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right], \tag{17}$$

where in the last step, we used the inequality in (10).
Note that

$$\mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right] = \mathbb{E}_{S_t} \left[\left\| \mathbf{g}_t \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \le \mathbb{E}_k \left[\left\| \nabla F_k(\mathbf{w}^{t-1}) \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \le (B^2 - 1) \left\| f(\mathbf{w}^{t-1}) \right\|^2. \tag{18}$$

Plugging into (17), we get

$$\mathbb{E}_{S_t} \left[f(\mathbf{w}^t) \right] \le f(\mathbf{w}^{t-1}) - \rho \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2, \tag{19}$$

where

$$\rho = \frac{1}{\mu} - \frac{3\gamma}{2(\mu - \lambda)} - \frac{L(1+\gamma)^2}{(\mu - \lambda)^2} - \frac{3L}{2\mu(\mu - \lambda)} - (B^2 - 1) \left(\frac{L(1+\gamma)^2}{(\mu - \lambda)^2} + \frac{L}{\mu(\mu - \lambda)} + \frac{\gamma}{\mu - \lambda} \right). \tag{20}$$

$\square$

F. Proof for Theorem 7

Proof. We have by the Lipschitz continuity of the gradients:

$$f(\mathbf{w}_k^t) \le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}_k^t - \mathbf{w}^{t-1} \rangle + \frac{L_k}{2} \left\| \mathbf{w}_k^t - \mathbf{w}^{t-1} \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}_k^t - \underline{\mathbf{w}}_k^t \rangle + \frac{L_k}{2} \left\| \mathbf{w}_k^t - \mathbf{w}^{t-1} \right\|^2 \tag{21}$$

By optimality conditions, we have that $\underline{\mathbf{w}}_k^t$ satisfies

$$\nabla F_k \left(\underline{\mathbf{w}}_k^t \right) + \mathbf{g}_t - \nabla F_k \left(\mathbf{w}^{t-1} \right) + \mu_k \left(\underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right) = 0. \tag{22}$$

Similarly, we denote the local subproblem (3) as $P_t(w)$. We also note that, $P_t(\mathbf{w})$ is $(\mu_k - \lambda)$-strongly convex,

$$\mu_k \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right\| \le \left\| \nabla P_t(\mathbf{w}^{t-1}) \right\| = \left\| \mathbf{g}_t \right\|. \tag{23}$$

We derive a bound for $\left\| \mathbf{w}_k^t - \mathbf{w}^{t-1} \right\|$ next.

$$\left\| \mathbf{w}_k^t - \mathbf{w}^{t-1} \right\| \le \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right\| + \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}_k^t \right\| \le (1 + \gamma_k) \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right\|. \tag{24}$$

Using (22)-(23) in (21), we have,

$$f(\mathbf{w}_k^t) \le f(\mathbf{w}^{t-1}) + \langle \nabla f(\mathbf{w}^{t-1}), \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \rangle + \langle \nabla f(\mathbf{w}^{t-1}), \mathbf{w}_k^t - \underline{\mathbf{w}}_k^t \rangle + \frac{L_k}{2} \left\| \mathbf{w}_k^t - \mathbf{w}^{t-1} \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{1}{\mu_k} \langle \nabla f(\mathbf{w}^{t-1}), \nabla F_k\left(\underline{\mathbf{w}}_k^t\right) + \mathbf{g}_t - \nabla F_k\left(\mathbf{w}^{t-1}\right) \rangle + \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \mathbf{w}_k^t - \underline{\mathbf{w}}_k^t \right\| + \frac{L_k(1+\gamma_k)^2}{2\mu_k^2} \left\| \mathbf{g}_t \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1})\mathbf{g}_t}{\mu_k} + \frac{L_k}{\mu_k} \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right\| + \gamma \left\| \nabla f(\mathbf{w}^{t-1}) \right\| \left\| \underline{\mathbf{w}}_k^t - \mathbf{w}^{t-1} \right\| + \frac{L_k(1+\gamma_k)^2}{2\mu_k^2} \left\| \mathbf{g}_t \right\|^2$$

$$\le f(\mathbf{w}^{t-1}) - \frac{\nabla^\top f(\mathbf{w}^{t-1})\mathbf{g}_t}{\mu_k} + \frac{L_k}{2\mu_k^2} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right)$$

$$+ \frac{\gamma_k}{2\mu_k} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right) + \frac{L_k(1+\gamma_k)^2}{2\mu_k^2} \left\| \mathbf{g}_t \right\|^2$$

$$\Rightarrow f(\mathbf{w}^t) \le \frac{1}{K_t} \sum_{k=1}^{K_t} f(\mathbf{w}_k^t) \le f(\mathbf{w}^{t-1}) - \left(\frac{1}{K_t} \sum_{k=1}^{K_t} \frac{1}{\mu_k} \right) \nabla^\top f(\mathbf{w}^{t-1})\mathbf{g}_t + \frac{1}{K_t} \sum_{k=1}^{K_t} \frac{L_k}{2\mu_k^2} \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right)$$

$$+ \left(\frac{1}{K_t} \sum_{k=1}^{K_t} \frac{\gamma_k}{2\mu_k} \right) \left(\left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \left\| \mathbf{g}_t \right\|^2 \right) + \left(\frac{1}{K_t} \sum_{k=1}^{K_t} \frac{L_k(1+\gamma_k)^2}{2\mu_k^2} \right) \left\| \mathbf{g}_t \right\|^2$$

$$\Rightarrow \mathbb{E}_{S_t}\left[f(\mathbf{w}^t) \right] \le f(\mathbf{w}^{t-1}) - \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{1}{\mu_k} - \frac{3\gamma_k}{2\mu_k} \right) \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 + \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{L_k(1+\gamma_k)^2}{\mu_k^2} + \frac{3L_k}{2\mu_k^2} \right) \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2$$

$$+ \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{L(1+\gamma_k)^2}{\mu_k^2} + \frac{L_k}{\mu_k^2} + \frac{\gamma_k}{\mu_k} \right) \mathbb{E}_{S_t}\left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right], \tag{25}$$

where $|S_t| = K_t$ and in the last step, we used the inequality in (10).
Note that

$$\mathbb{E}_{S_t}\left[\left\| \mathbf{g}_t - \nabla f(\mathbf{w}^{t-1}) \right\|^2 \right] = \mathbb{E}_{S_t}\left[\left\| \mathbf{g}_t \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \le \mathbb{E}_k\left[\left\| \nabla F_k(\mathbf{w}^{t-1}) \right\|^2 \right] - \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2 \le (B^2 - 1) \left\| f(\mathbf{w}^{t-1}) \right\|^2. \tag{26}$$

Plugging into (25), we get

$$\mathbb{E}_{S_t}\left[f(\mathbf{w}^t) \right] \le f(\mathbf{w}^{t-1}) - \rho \left\| \nabla f(\mathbf{w}^{t-1}) \right\|^2, \tag{27}$$

where

$$\rho = \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{1}{\mu_k} - \frac{3\gamma_k}{2\mu_k} - \frac{L_k(1+\gamma_k)^2}{\mu_k^2} - \frac{3L_k}{2\mu_k^2} \right) - \frac{1}{K_t} \sum_{k=1}^{K_t} \left(\frac{L(1+\gamma_k)^2}{\mu_k^2} + \frac{L_k}{\mu_k^2} + \frac{\gamma_k}{\mu_k} \right) (B^2 - 1). \tag{28}$$
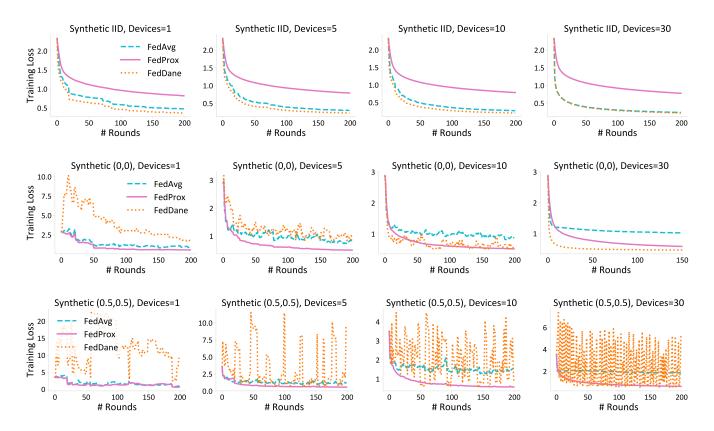
$\square$

G. Additional Experiments

Fig. 2: Effects of low device participation. For the three synthetic datasets with varying statistical heterogeneity, we randomly select 1, 5, 10, or 30 devices (out of 30) at each communication round. We set E to be 20. From the top row to the bottom row, data heterogeneity is increasing. We see that (1) low device participation hurts the performance of `FedDANE` in statistically heterogeneous settings, and (2) in highly heterogeneous environments (e.g., on the Synthetic (0.5,0.5) dataset), even full device participation does not help improve the performance of `FedDANE`.
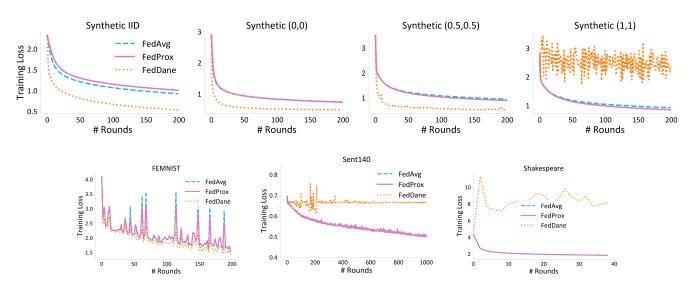

Fig. 3: Convergence of `FedDANE` compared with `FedAvg` and `FedProx` in *unrealistic settings* (nearly full device participation, small local epochs $E = 1$) which favor `FedDANE`. For synthetic datasets, we let all devices participate in learning at each iteration. For FEMNIST, Sent140, and Shakespeare, we select 50%, 26%, and 70% devices respectively at each round in order to better estimate the full gradients. `FedDANE` still performs worse than the other two methods, especially on highly heterogeneous datasets.