Distributed Fixed Point Methods with Compressed Iterates*

Sélim Chraibi[♦♣] Ahmed Khaled[♦♠] Dmitry Kovalev[♦]
Peter Richtárik[♦] Adil Salim[♦] Martin Takáč[♥]

[♦] King Abdullah University of Science and Technology
[♣] Université Grenoble Alpes
[♠] Cairo University
[♥] Lehigh University

December 20, 2019
Revised on February 17, 2021†

Abstract

We develop distributed iterative fixed point methods which address the communication bottleneck by compressing their iterates before communication. Our work is motivated by the practice of federated learning, where model compression is a common practice, albeit without theoretical guarantees, and where various gradient-type local update methods are applied, which we model via the application of local operators on each device. We develop standard and variance reduced methods, and establish communication complexity bounds. Our algorithms are the first theoretically justified distributed methods with compressed iterates, and the first fixed point methods with compressed iterates.

Contents

*This paper was prepared during Summer 2019 when the first two authors were research interns at KAUST.

†We have added a relaxation/stepsize parameter to Algorithms 1 and 3, and have added a large number of new experiments.

1 Introduction

Communication efficiency and memory issues often arise in machine learning algorithms dealing with large models. This is particularly true in federated learning [6, 26, 27, 33], where a network of a very large number (n) of computing devices, such as mobile phones or hospitals, is required to jointly train a machine learning model from the private data stored on these devices, and more so if these models are large. To train such a model, various distributed training methods have been devised, and all necessarily need to communicate model updates if a globally optimal model is to be found. However, communication in federated learning is very expensive [26], and forms a major bottleneck of existing systems. To overcome this issue, standard techniques proceed by compressing the communicated messages, typically gradients or stochastic gradients [1, 5, 28, 46].

In this paper we consider the case where the iterates themselves need to be compressed. This case is relevant even in the $n = 1$ regime, where there is only one computing agent, provided that the model is too large to keep in memory and needs to be compressed. Training with compressed iterates was only considered in one recent work [24], which introduced the gradient descent algorithm with compressed iterates (GDCI). In this paper we improve the results of [24] and generalize them in two ways. First, in the case $n = 1$, we consider iterate compression in any algorithm that can be formulated as a (stochastic) fixed point iteration. This covers gradient descent and stochastic gradient descent, among others. Second, we consider the distributed case $n \geq 2$, where the network has to jointly find a fixed point of some map, in a distributed manner over the nodes, and using iterate compression. This distributed fixed point problem covers many applications of federated learning [26], including distributed minimization or distributed saddle point problems. We remark that while both gradient compression and iterate compression are used by practicioners of federated learning, iterate compression is much less well-understood [23] and our work aims to fill this gap.

1.1 Summary of contributions

To address these problems we first study a naive approach that relies on compressing the iterates after each iteration. This iterates compression introduces an extra source of variance in the algorithms. We then propose a variance reduced approach that allows to remove the variance induced by the compression. In summary, we make the following contributions:

- We propose new distributed algorithms (non variance reduced and variance reduced) to learn with compressed iterates in a stochastic fixed point framework, which we show captures gradient descent as well as a variety of other methods such as (stochastic) gradient descent-ascent, Davis-Yin splitting, and others. These are the first federated fixed point methods with compression.

- We derive non-asymptotic convergence rates for these methods, and our theory allows improved rates when specialized to GDCI compared to prior work.

- We show our variance reduced algorithm is able to retain the *linear convergence* of gradient descent on strongly convex objectives despite iterate compression: this is new, and it shows we can guarantee the fast convergence of gradient descent despite communicating only compressed iterates.

- We conduct extensive numerical experiments (55 figures, each with multiple plots) with the developed algorithms on synthetic and real datasets and report our findings, highlighting many of their properties in tandem with our theory.

1.2 Related work

Communication-Efficiency. In distributed optimization, the communication cost is the bottleneck. In order to reduce it, many methods have been suggested, including the use of intermittent communication and decentralization [44], or exchanging only compressed or quantized information between the computing units [42]. Usually, the exchanged information is some compressed gradient [1, 5, 40] or compressed model update [2, 37] in a distributed master-worker setting. We note that in the setting of gradient compression, various methods have also been developed to reduce the noise from gradient compression and the method we develop is similar in spirit to some of them, like [22, 34]. As noted before, iterate compression is also used in Federated Learning, see e.g., [6, 23, 26], where concerns of communication efficiency and memory usage are particularly important.

Decentralized Methods. In decentralized settings, one can distinguish between exact and approximate methods. Among inexact methods, some compute (sub)gradients at compressed iterates: [35,36]. Exact methods usually exchange compressed gradients or iterates [4,16,25,29,38,48]. Our focus in this work is on centralized methods, and we leave an extension to decentralized settings to future work.

Beyond Compression Operators. In the $n = 1$ case, our method can also be seen as an analysis of fixed point methods with perturbed iterates in a similar spirit to [15], who analyze gradient descent methods given access to an inexact oracle.

The remainder is organized as follows. In the next section we provide some background on distributed fixed point problems and compression operators, and make our assumptions. In Section 3, we consider the case $n = 1$, where there is only one computing unit. We describe our algorithms, state the main results and instantiate the algorithms to practical (stochastic) fixed point iterations. The case $n = 1$ is generalized in Section 4 where a network of computing units is considered. We describe our distributed algorithms, state the main results and instantiate the distributed algorithms to practical distributed (stochastic) fixed point iterations. Finally, we experiment on several federated learning tasks in Section 5. The proofs and additional numerical experiments are postponed to the appendix.

2 The Setting

In this section we introduce our key problem, a distributed fixed point problem, and establish notation and assumptions that we will use in the rest of the paper. We also define the notion of a randomized compression operator that will be used to compress the iterates.

2.1 Distributed fixed point framework

Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$ be operators on $\mathbb{R}^d$, i.e., $\mathcal{T}_i : \mathbb{R}^d \to \mathbb{R}^d$. We consider the so-called finite-sum optimization setting and set

$$\mathcal{T}(x) \coloneqq \frac{1}{n} \sum_{i=1}^{n} \mathcal{T}_i(x); \tag{1}$$

our goal is to find a fixed point $x^\star$ of $\mathcal{T}$; that is,

$$\mathcal{T}(x^\star) = x^\star. \tag{2}$$

Consider a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, a family $s := (s_i)_{i\in\{1,\dots,n\}}$ of random variables defined on $(\Omega, \mathscr{F}, \mathbb{P})$ with values in some measurable space $(\Xi, \mathscr{G})$. Denote by $\mathcal{S}_i$ the distribution of s_i and by $\mathcal{S}$ the distribution (over Ξ^n) of s. We allow each $\mathcal{T}_i$ to have the following stochastic representation:

$$\mathcal{T}_i(x) \coloneqq \mathbb{E}_{s_i}\left[\mathcal{T}_i(x, s_i)\right], \tag{3}$$

where, with a small abuse of notation, $\mathcal{T}_i(x,\cdot)$ denotes an $\mathcal{S}_i$-integrable function for every $x \in \mathbb{R}^d$. We also set

$$\mathcal{T}(x, s) \coloneqq \frac{1}{n}\sum_{i=1}^n \mathcal{T}_i(x, s_i), \tag{4}$$

for every $x \in \mathbb{R}^d$. Note that $\mathcal{T}(x,\cdot)$ is $\mathcal{S}$-integrable and that $\mathbb{E}_s\left[\mathcal{T}(x, s)\right] = \mathcal{T}(x)$.

We assume the following contraction property for the stochastic map $\mathcal{T}(\cdot, s)$.

Assumption 1. *There exist $x^\star \in \mathbb{R}^d$, $B \geq 0$ and $\rho \in (0, 1)$ such that, for every $x \in \mathbb{R}^d$,*

$$\mathbb{E}_s\left[\left\|\mathcal{T}(x, s) - x^\star\right\|^2\right] \leq (1-\rho)\left\|x - x^\star\right\|^2 + B. \tag{5}$$

This assumption is satisfied by many maps $\mathcal{T}(\cdot, s)$ describing (stochastic) optimization algorithms under some strong convexity and smoothness assumptions; see Sections 3 and 4. We shall also use the expected Lipschitz continuity of $\mathcal{T}_i(x, s)$ defined as follows.

Assumption 2. *For every $i \in \{1, \dots, n\}$, there exists $c_i \geq 0$ such that for every $x, y \in \mathbb{R}^d$:*

$$\mathbb{E}_s\left[\left\|\mathcal{T}_i(x, s) - \mathcal{T}_i(y, s)\right\|^2\right] \leq c_i\|x - y\|^2, \tag{6}$$

and we denote $c^2 \coloneqq \frac{1}{n}\sum_{i=1}^n c_i^2$.

2.2 Compression operators

To apply randomized compression to the iterates, we require access to a *stochastic compression operator* and we formalize our assumptions on this operator next. Consider a family $\xi := (\xi_i)_{i\in\{1,\dots,n\}}$ of random variables defined on $(\Omega, \mathscr{F}, \mathbb{P})$ with values $(\Xi, \mathscr{G})$. If $n = 1$, we shall prefer the notation ξ for ξ_1. We consider a measurable map $\mathcal{C} : \mathbb{R}^d \times \Xi \to \mathbb{R}^d$ such that, for every $i \in \{1, \dots, n\}$,

$$x = \mathbb{E}_{\xi_i}\left[\mathcal{C}(x, \xi_i)\right]. \tag{7}$$

The map $\mathcal{C}$ is called a compression operator. We make the following assumption on $\mathcal{C}$.

Assumption 3. *There exists $\omega \geq 0$ such that, for every $i \in \{1, \dots, n\}$ and $x \in \mathbb{R}^d$,*

$$\mathbb{E}_{\xi_i}\left[\left\|\mathcal{C}(x; \xi_i) - x\right\|^2\right] \leq \omega\|x\|^2. \tag{8}$$

Assumption 3 has been used before, either in this general form or in special cases, in the analysis of gradient methods with *compressed gradients* [22, 25] and *compressed iterates* [24]. Many practical compression operators satisfy this assumption; e.g., natural compression and natural dithering [21], standard dithering, sparsification, and quantization [1, 22, 24, 41].

3 Results in the Single Node Case ($n = 1$)

In this section, we present two algorithms to solve (2) in the case when $n = 1$ and state two theorems related to these algorithms. While this simplified setting may be of interest on its own, we consider it first as it will allow for a more gradual exposition into our methods. However, keeping in mind that we are primarily interested in the distributed setting, our methods should still be understood as performing communication with a server performing (in this case dummy) aggregation over all (in this case 1) devices.

3.1 Fixed Point Method with Compressed Iterates

Consider stochastic fixed point iterations of the form

$$x^{k+1} = \mathcal{T}(x^k, s^k), \tag{9}$$

where s^k is a sequence of i.i.d. copies of s. Our first algorithm (FPMCI) applies the stochastic fixed point operator to x^k (this step leads to $\mathcal{T}(x^k, s^k)$), applies compression $\mathcal{C}$ (this step leads to the compressed vector $\mathcal{C}(\mathcal{T}(x^k, s^k), \xi^k)$), and subsequently performs a relaxation step with relaxation parameter η. If $\mathcal{C}$ is the identity map, FPMCI is a stochastic fixed point method with relaxation.

Algorithm 1 FPMCI: Fixed Point Method with Compressed Iterates

Initialization: $x^0 \in \mathbb{R}^d$, (ξ^k) i.i.d. copies of ξ, (s^k) i.i.d. copies of s, stepsize $\eta \in (0, 1]$.
for $k = 0, 1, 2, \ldots$ **do**

$$x^{k+1} = (1 - \eta)x^k + \eta\mathcal{C}(\mathcal{T}(x^k, s^k), \xi^k).$$

end for

Our first result characterizes the convergence of Algorithm 1.

Theorem 1. *Suppose that Assumptions 1, 2 and 3 hold. Set* $r^k := \left\|x^k - x^\star\right\|^2$. *Suppose that the stepsize* $\eta > 0$ *is chosen such that* $\eta \leq \min\left(1, \frac{\rho}{4\omega c^2}\right)$. *Then the iterates of Algorithm 1 satisfy*

$$\mathbb{E}\left[r^k\right] \leq \left(1 - \frac{\eta\rho}{2}\right)^k r^0 + \frac{2B}{\rho} + \frac{4\eta\omega\sigma^2}{\rho},$$

where $\sigma^2 := \mathbb{E}_s\left[\left\|\mathcal{T}(x^\star, s)\right\|^2\right]$.

The convergence rate is linear up to a ball of squared radius $\frac{2B}{\rho} + \frac{4\eta\omega\sigma^2}{\rho}$. The first term $\frac{2B}{\rho}$ comes from Assumption 1 and is inherent to the mapping $\mathcal{T}$. This first term is also proportional to B, and the value of B is usually zero for deterministic fixed point maps $\mathcal{T}$ and nonzero for stochastic mappings; see the next subsection. The presence of the second term $\frac{4\eta\omega\sigma^2}{\rho}$ is a consequence of the variance of the compression operator. If $\omega = 0$ (no compression), then the second term is equal to zero.[1] Also notice that the second term is proportional to η and can be made small if η is chosen small.

[1] Having $\sigma^2 = 0$ is hopeless, except in particular cases like $\mathcal{T}$ deterministic and $x^\star = 0$.

3.2 Variance Reduced Fixed Point Method with Compressed Iterates

We now address the presence of the second term by reducing the variance introduced by the compression operator. This leads to a new method called Variance Reduced (VR) FPMCI, described in Algorithm 1.

Algorithm 2 VR-FPMCI: Variance Reduced Fixed Point Method with Compressed Iterates

Initialization: $x^0, h^0 \in \mathbb{R}^d$, (ξ^k) i.i.d. copies of ξ, (s^k) i.i.d. copies of s, stepsizes $\eta \in (0,1]$ and $\alpha > 0$.
for $k = 0, 1, 2, \ldots$ **do**

$$\begin{aligned} \delta^{k+1} &= \mathcal{C}(\mathcal{T}(x^k, s^k) - h^k, \xi^k) \\ h^{k+1} &= h^k + \alpha\delta^{k+1} \\ x^{k+1} &= (1-\eta)\, x^k + \eta\left(h^k + \delta^{k+1}\right). \end{aligned}$$

end for

The improved convergence rate of Algorithm 2 is stated in the next theorem.

Theorem 2. *Let Ψ^k be the following Lyapunov function:*

$$\Psi^k := \left\|x^k - x^\star\right\|^2 + \frac{4\eta^2\omega}{\alpha}\mathbb{E}_s\left[\left\|h^k - \mathcal{T}(x^\star, s^k)\right\|^2\right].$$

Suppose that Assumptions 1, 2 and 3 hold. Choose the stepsizes α, η such that $\alpha \leq \frac{1}{\omega+1}$ and $\eta = \min\left\{1, \frac{\rho}{12\omega c^2}\right\}$. Then the iterates defined by Algorithm 2 satisfy

$$\mathbb{E}\left[\Psi^k\right] \leq \left(1 - \frac{\min\{\alpha, \eta\rho\}}{2}\right)^k \mathbb{E}\left[\Psi^0\right] + \frac{2\eta B}{\min\{\alpha, \eta\rho\}}. \tag{10}$$

Note that, as promised, there is no additive term depending on ω in the r.h.s. of (10), thanks to the variance reduction property of our method. In particular, Algorithm 2 converges linearly if $B = 0$, and allows for arbitrarily large compression variance factor ω.

3.3 Examples

We give several examples of operators $\mathcal{T}$ to which our analysis of Algorithms 1 and 2 applies.

Gradient Descent. Consider an L-smooth μ-strongly convex objective function $F : \mathbb{R}^d \to \mathbb{R}$ and a stepsize $\gamma \in \left(0, \frac{1}{L}\right]$. Then

$$\mathcal{T}_{\mathrm{GD}} : x \mapsto x - \gamma\nabla F(x) \tag{11}$$

satisfies Assumption 1 with $\rho = \gamma\mu$ and $B = 0$, and Assumption 2 with $c = 1$, see [3]. As a result, for any compression operator $\mathcal{C}$ satisfying Assumption 3, Theorem 1 states that

$$\mathbb{E}\left[r^k\right] \leq \left(1 - \frac{\gamma\mu\eta}{2}\right)^k r^0 + \frac{4\eta\omega\sigma^2}{\gamma\mu},$$

where $r^k \coloneqq \left\|x^k - x^\star\right\|^2$. Without relaxation (i.e. with $\eta = 1$), *this result improves upon the result obtained in [24]* by requiring $\omega < \frac{1}{2\kappa}$ rather than $\omega < \frac{1}{76\kappa}$, while still guaranteeing convergence. By properly choosing the relaxation parameter η, we can guarantee convergence to a neighborhood for *any* $\omega > 0$. Moreover, Theorem 2 shows that $\mathbb{E}\left[r^k\right]$ converges linearly to zero, rather than to a neighborhood of the solution, if Algorithm 2 is applied.

Stochastic Gradient Descent (SGD). Consider a μ-strongly convex function $F : \mathbb{R}^d \to \mathbb{R}$ written as $F(x) = \mathbb{E}_s\left[f(x,s)\right]$, where $f(\cdot, s)$ is convex and L-smooth for every s. Then, $g(x,s) \coloneqq \nabla f(x,s)$ is an unbiased estimate of ∇F; that is, for every $x \in \mathbb{R}^d, \mathbb{E}_s\left[g(x,s)\right] = \nabla F(x)$. Moreover, Assumption 2 is satisfied by the map

$$\mathcal{T}_{\text{SGD}} : (x,s) \mapsto x - \gamma g(x,s),$$

with $c = 1$, for the same reason as above. Finally, Assumption 1 is satisfied with $\rho = \gamma\mu$ and $B > 0$ in general, see e.g. [20].

Proximal SGD. One can generalize the previous example to the map

$$\mathcal{T}_{\text{prox-SGD}} : (x,s) \mapsto \text{prox}_{\gamma H}(x - \gamma g(x,s)),$$

where H is a convex, lower semicontinuous and proper function $\mathbb{R}^d \to (-\infty, +\infty]$, and $\text{prox}_{\gamma H}$ is the proximity operator of γH defined as

$$\text{prox}_{\gamma H}(x) \coloneqq \underset{y \in \mathbb{R}^d}{\arg\min} \left\{ \frac{1}{2}\|x - y\|^2 + \gamma H(y) \right\}.$$

The map $\mathcal{T}_{\text{prox-SGD}}$ also satisfies the Assumption 2, see [3]. Besides, Assumption 1 is satisfied with $\rho = \gamma\mu$ and $B > 0$ in general, see e.g. [19].

Note that a fixed point of $\mathcal{T}_{\text{prox-SGD}}$ is a minimizer of $F + H$.

Douglas–Rachford splitting. The Douglas–Rachford splitting (or ADMM) [31] is an algorithm allowing to minimize $G + H$, where G and H are convex nonsmooth functions. More precisely, assume that $G, H : \mathbb{R}^d \to (-\infty, +\infty]$ are lower semicontinuous and proper, and denote $C_G(x) = 2\,\text{prox}_G(x) - x$. Given $\gamma > 0$, Douglas–Rachford corresponds to iterations of the map

$$\mathcal{T}_{\text{DR}} : x \mapsto \frac{1}{2} C_{\gamma H}(C_{\gamma G}(x)) + \frac{1}{2} x.$$

A fixed point of the Douglas–Rachford algorithm is an element $x \in \mathbb{R}^d$ such that $\text{prox}_{\gamma G}(x) \in \arg\min G + H$. Moreover, $\mathcal{T}_{\text{DR}}$ satisfies Assumption 2 with $c = 1$, see [3]. The map $\mathcal{T}_{\text{DR}}$ also satisfies Assumption 1 with $B = 0$ under various assumptions, for instance if G is strongly convex and H is smooth.

Proximal Alternating Predictor–Corrector (PAPC). The PAPC algorithm [10, 17, 32] is an algorithm for minimizing $F + H \circ L$, where F is smooth and convex, H is nonsmooth and convex and L is a linear operator. The PAPC algorithm corresponds to iterations of a map $\mathcal{T}_{\text{PAPC}}$, whose fixed points are minimizers of $F + H \circ L$. The map $\mathcal{T}_{\text{PAPC}}$ satisfies Assumption 2 with $c = 1$, see [17]. Moreover, this map satisfies Assumption 1 with $B = 0$ if F is strongly convex and H the function equal to $+\infty$ everywhere except at one point b, see [39]. In this case, minimizing $F + H \circ L$ is equivalent to minimizing F under the affine constraints $Lx = b$.

Primal–Dual Hybrid Gradient (PDHG). The PDHG algorithm [7, 8], a.k.a Chambolle–Pock, allows to minimize $G + H \circ L$, where G, H are nonsmooth convex functions and L is a linear operator. If L is the identity, then the PDHG algorithm boils down to the Douglas–Rachford algorithm. The PDHG algorithm corresponds to iterations of a map $\mathcal{T}_{\mathrm{PDHG}}$ satisfying Assumption 2. Moreover, the map $\mathcal{T}_{\mathrm{PDHG}}$ satisfies Assumption 1 with $B = 0$ if G is strongly convex and H is smooth.

Davis–Yin splitting. Davis-Yin splitting [14] is an optimization algorithm to minimize a sum of three convex functions $F + G + H$. It is a generalization of Gradient Descent, Proximal Gradient Descent, and Douglas Rachford algorithms, and it corresponds to iterations of a map $\mathcal{T}_{\mathrm{DY}}$. The map $\mathcal{T}_{\mathrm{DY}}$ satisfies Assumptions 1. Moreover, Assumption 2 is satisfied with $B = 0$ if at least one of F, G or H is strongly convex and at least one of G or H is smooth, see [14].

Condat–Vũ splitting. Condat–Vũ splitting [11, 12, 43] is an optimization algorithm to minimize a sum of three convex functions $F + G + H \circ L$ where L is a linear operator. It is a generalization of Proximal Gradient Descent and PDHG algorithms, and it corresponds to iterations of a map $\mathcal{T}_{\mathrm{CV}}$. The map $\mathcal{T}_{\mathrm{CV}}$ satisfies Assumptions 1 and 2 with $B = 0$ if G is strongly convex and H is smooth.

Primal–Dual 3 Operators (PD3O) and Primal–Dual Davis–Yin (PDDY) splitting The PD3O [47] and the PDDY [39] algorithms are similar to the Condat–Vũ algorithm to tackle the minimization of $F + G + H \circ L$ [13]. They satisfy our assumptions under various hypotheses on the functions F, G and H. Interestingly, PD3O and PDDY admit stochastic versions, also satisfying our assumptions. In these stochastic versions, the gradient of F is replaced by a stochastic gradient $\nabla f(x, s)$ at each iteration [39].

(Stochastic) Gradient Descent Ascent. Consider a μ-strongly convex-concave function $F : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ (strongly convex in x and strongly concave in y) with L-Lipschitz continuous gradient. Then, the map

$$\mathcal{T}_{\mathrm{GDA}}(x, y) = \begin{pmatrix} x \\ y \end{pmatrix} - \gamma \begin{pmatrix} \nabla_x F(x, y) \\ -\nabla_y F(x, y) \end{pmatrix} \tag{12}$$

satisfies Assumption 2 and Assumption 1 with $B = 0$ if γ is small enough. Moreover, a fixed point of $\mathcal{T}_{\mathrm{GDA}}$ is a saddle point $(x^\star, y^\star)$ of F. This example can be generalized to the case where $(\nabla_x F(x, y), -\nabla_y F(x, y))^T$ is replaced by an unbiased estimate (similarly to SGD), in which case Assumption 1 holds with $B \geq 0$ in general.

4 Results in the Distributed Case ($n > 1$)

We now consider the case where n computing agents are required to compute a fixed point of $\mathcal{T}$, under the restriction that each node i only has access to the *local* random map $\mathcal{T}_i(\cdot, \xi_i)$. This is the standard setup in distributed finite-sum optimization problems where a dataset is divided among several nodes, and is crucial in the setting of federated learning [23].

We solve this problem in a distributed master/slave setting, where each iteration is divided into a computation step and a communication step. During the computation step, every node i uses $\mathcal{T}_i(\cdot, \xi_i)$ to update some variables only locally. Then, during the communication step, each node

sends its local variable to the master node of the network that aggregates the variables and sends back the result to the other nodes.

We extend Algorithm 1 (resp. Algorithm 2) to this setting, as well as Theorem 1 (resp. Theorem 2).

4.1 Distributed Fixed Point Method with Compressed Iterates

Extension of Algorithm 1 to the distributed (i.e., $n > 1$) setting is formulated as Algorithm 3.

Algorithm 3 Distributed Fixed Point Method with Compressed Iterates

Initialization: $x^0 \in \mathbb{R}^d$, (ξ^k) i.i.d. copies of ξ, (s^k) i.i.d. copies of s, stepsize $\eta > 0$.
for $k = 0, 1, 2, \ldots$ **do**
  Broadcast x^k to all nodes
  for $i = 1, \ldots, n$ in parallel **do**
    Compute the compressed iterate

$$\delta_i^{k+1} = \mathcal{C}\left(\mathcal{T}_i(x^k, s_i^k); \xi_i^k\right)$$

    Communicate the compressed vector δ_i^{k+1} to the master node
  end for
  Compute the average of the communicate messages:

$$x^{k+1} = (1-\eta)x^k + \frac{\eta}{n}\sum_{i=1}^n \delta_i^{k+1} \tag{13}$$

  Broadcast the new iterate x^{k+1} to the workers
end for

The convergence rate of this method is a direct generalization of Theorem 1.

Theorem 3. *Let Assumptions 1, 2 and 3 hold. Assume moreover that $s_1, \ldots, s_n$ are independent. Let $r^k := \left\|x^k - x^\star\right\|^2$. Let stepsize $\eta > 0$ satisfy $\eta \leq \min\left(1, \rho n/(4\omega c^2)\right)$. Then the iterates of Algorithm 3 satisfy*

$$\mathbb{E}\left[r^k\right] \leq \left(1 - \frac{\eta\rho}{2}\right)^k r^0 + \frac{2B}{\rho} + \frac{4\eta\omega\sigma^2}{\rho n},$$

where $\sigma^2 := \frac{1}{n}\sum_{i=1}^n \mathbb{E}_{s_i}\left[\left\|\mathcal{T}_i(x^\star, s_i)\right\|^2\right]$.

4.2 Distributed Variance-Reduced Fixed Point Method with Compressed Iterates

Once again, the rate suffers from the variance term $\frac{4\eta\omega\sigma^2}{\rho n}$, which is removed by our variance reduced approach summarized in Algorithm 4. Finally, the next result is the analogue of Theorem 2 in the distributed setting.

By Assumption 1, when compression is not used and the iterate $x_+ = T(x, s)$ for some $s \sim \mathcal{S}$ is used, we have

$$\mathbb{E}\left[\|x_+ - x^\star\|^2\right] \leq (1-\rho)\,\mathbb{E}\left[\|x - x^\star\|^2\right] + B. \tag{14}$$

Here we can gain insight as to what compression does by comparing this with the result of Theorem 3: we see that rate at which the initial distance is forgotten is reduced by a factor $2\omega c^2/n$ and that convergence is only guaranteed to a neighborhood $\mathcal{O}(\omega\sigma^2/n)$. Thus, to guarantee convergence and that the neighborhood is small enough we must take $\omega = \mathcal{O}(\varepsilon n/\sigma^2)$. Depending on the accuracy,this may allow only very small levels of compression that makes using Algorithm 3 undesirable from a theoretical point of view. To allow larger values ω and still guarantee similar convergence properties as (14), we introduce a *variance-reduced method* that generalizes Algorithm (2). It is given by Algorithm 4.

Algorithm 4 Distributed Variance-Reduced Fixed Point Method with Compressed Iterates

Initialization: $x^0, h_1^0, h_2^0, \ldots, h_n^0 \in \mathbb{R}^d$, stepsize $\eta \in (0,1]$, stepsize $\alpha > 0$, $h^0 = \frac{1}{n}\sum_{i=1}^n h_i^0$

for $k = 0, 1, 2, \ldots$ **do**

Broadcast x^k to all nodes.

for $i = 1, \ldots, n$ in parallel **do**

Compute

$$\begin{aligned} \delta_i^{k+1} &= \mathcal{C}(\mathcal{T}_i(x^k, s_i^k) - h_i^k; \xi_i^k) \\ h_i^{k+1} &= h_i^k + \alpha\delta_i^{k+1} \end{aligned}$$

Communicate the compressed vector δ_i^{k+1} to the master node

end for

At the master node, compute:

$$\begin{aligned} \delta^{k+1} &= \frac{1}{n}\sum_{i=1}^n \delta_i^{k+1} \\ h^{k+1} &= h^k + \alpha\delta^{k+1} \\ \Delta^{k+1} &= \delta^{k+1} + h^k \\ x^{k+1} &= (1-\eta)\,x^k + \eta\Delta^{k+1} \end{aligned}$$

Broadcast the new iterate x^{k+1} to the workers

end for

Theorem 4. *Define the Lyapunov function*

$$\Psi^k \coloneqq \left\|x^k - x^\star\right\|^2 + \frac{4\eta^2\omega}{\alpha n^2}\sum_{i=1}^n \mathbb{E}_{s_i}\left[\left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right].$$

Suppose that Assumptions 1, 2 and 3 hold. Assume moreover that $s_1, \ldots, s_n$ are independent. Then the iterates defined by Algorithm 4 satisfy

$$\mathbb{E}\left[\Psi^k\right] \leq \left(1 - \frac{\min\{\alpha, \eta\rho\}}{2}\right)^k \mathbb{E}\left[\Psi^0\right] + \frac{2\eta B}{\min\{\alpha, \eta\rho\}}, \tag{15}$$

if the stepsizes α, η *satisfy* $\alpha \leq \frac{1}{\omega+1}$ *and* $\eta = \min\left\{\frac{\rho n}{12\omega c^2}, 1\right\}$.

Algorithm 4 converges linearly if $B = 0$. Hence, it trades off some local storage cost (by keeping the variables h_i^k) for faster convergence while still minimizing the cost of communication. We further note that in the special case $\alpha \simeq 1$ the algorithm reduces to compressing the *model update* in expectation, a practice that is already common in practice. This further mirrors the results of [22, 34] where compressing gradient differences rather than gradients enables several benefits over compressing gradients. The message of Theorem 4, therefore, is that compressing iterate differences rather than iterates also leads to better convergence properties.

4.3 Examples

Distributed (Stochastic) Gradient Descent. Consider a μ-strongly convex objective function $F : \mathbb{R}^d \to \mathbb{R}$ expressed as a finite-sum problem,

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

where each f_i is L_i-smooth and convex. Then one can check that the map $\mathcal{T}_{GD}$ defined in (11) takes the form (1) and that Assumptions 1 and 2 are satisfied by this map if γ is small enough, see e.g. [20]. Algorithm 4 is then a distributed gradient descent algorithm with iterates compression that converges linearly. More generally, if the f_i are themselves written as expectations and have the expected Lipschitz continuity property and convexity, one can check that Assumptions 1 and 2 are still satisfied.

Distributed (Stochastic) Gradient Descent Ascent. Example (12) can be extended to the case where F is expressed as an empirical mean over the nodes of the network, if each term has a Lipschitz continuous gradient. In this case, the distributed Algorithm 4 converges linearly to a saddle point $x^\star$ of F.

5 Experimental Results

In this section we present numerical results that demonstrate the conclusions of the theoretical convergence results for Algorithms FPMCI (Algorithm 3) and VR-FPMCI (Algorithm 4) on LibSVM datasets [9]. We summarize the essential characteristics of these datasets in the appendix. Let us remark that we selected both smaller and larger datasets to demonstrate various aspects of the algorithms.

Datasets of the main paper. We experiment with four datasets from the LibSVM library; see Table 1.

Three Problems. We solve three different problems:

- Regression (R),

$$F(x) := \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}(a_i^T x - b_i)^2,$$

Table 1: The basic characteristics of datasets used in our experiments. Datasets can be downloaded from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Name	**# samples**	d	**Source**
breast-cancer	683	10	[18]
webspam uni	350,000	254	[45]
rcv1	20,242	47,236	[30]
real-sim	72,309	20,958	[18]

- Ridge-Regression (RR),

$$F(x) := \frac{\lambda}{2}\|x\|^2 + \frac{1}{n}\sum_{i=1}^{n} \tfrac{1}{2}(a_i^T x - b_i)^2,$$

- and Logistic Regression (LR):

$$F(x) := \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i a_i^T x),$$

where $a_1, a_2, \ldots, a_n \in \mathbb{R}^d$ are data points, $b_1, \ldots, b_n \in \mathbb{R}$ and $y_1, \ldots, y_n \in \{-1, 1\}$. In order to compute $\|x^k - x^*\|^2$, we generated b such that we know the true value of x^*. This process is more challenging for larger datasets and hence we run RR exclusively for the *breast-cancer* dataset. On each plot we clearly note the Problem (R/RR/LR), the dataset used, and also some setting of algorithms. Unless explicitly stated otherwise, we solve all of the problems with the gradient descent fixed point operator.

Compression operators. In the experiments, we use two compression operators: Standard Dithering and Natural Dithering, as described in [21]. When we refer to b-bit precision, the compression operator is using 1 bit to encode the sign and $b-1$ bits to encode the compressed value. Hence we have 2^{b-1} possible positive and 2^{b-1} negative values the compression operator can produce.

Hardware. The experiments were conduced on a CentOS Linux (release 7.6.1810) Linux machine with Intel(R) Xeon(R) CPU E5-2670, 2.60GHz CPUs, with 128GB of RAM.

Software. We used Python 3.7.0 with numpy (version 1.17.3) and scipy (version 1.3.1).

More experiments. We have performed many additional experiments which can be found in the appendix.

Details for Ridge Regression experiments. The choice of γ for a gradient fixed point operator was $1/L$, where L is the maximum eigenvalue of the hessian. The code which generate the Ridge Regression problem will compute the constant L. The choince of other parameters η, λ is described in figures. We have run each algorithm 10x and we are reporting the mean and confidence intervals using seaborn lineplot function with default settings.

Details for Regression experiments. For rcv1 we have chosen $\gamma = 22.286873$ and run the algorithm $10\times$ and report mean and confidence intervals using seaborn lineplot function with default settings.

Details for Logistic Regression experiments. For the experiments run the algorithm 5x for each setting of n and report mean and confidence intervals using seaborn lineplot function with default settings. For webspam and real-sim we choose $n \in \{2, 4, 8, 16, 32, 64, 128\}$ and for rcv1 we choose $n \in \{2, 4, 8, 16, 32, 64\}$. For all datasets we have chosen $\gamma = 1$. Other parameters are described in the main paper.

5.1 Effect of the relaxation parameter η on convergence

For a gradient descent fixed-point operator, we have $B = 0$. From Theorem 1 we can see that for a given fixed-point operator, we have a fixed decrease factor guarantee of $1 - \rho$. The smaller the value η is, the slower the convergence. On the other hand, with the same compression (with variance ω), we get a smaller error at convergence with smaller η.

In Figure 1 we compare various compression levels of Natural Dithering with different values of ω for two values of $\eta \in \{1.0, 0.1\}$. Note that for some compression precisions, the algorithm is not convergent because η is chosen larger than required by the theory.

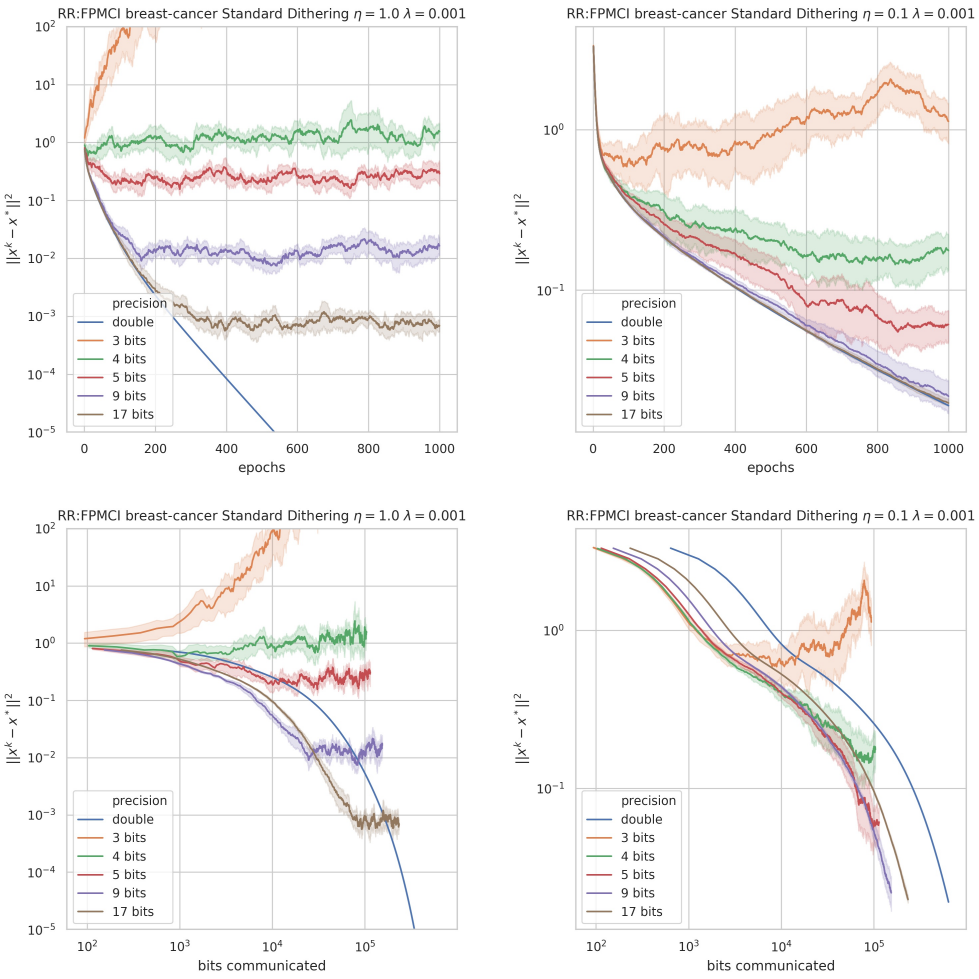

Figure 1: Comparison of convergence of FPMCI for two choices of $\eta \in \{1.0, 0.1\}$. The smaller the value of η, the slower the convergence gets, but smaller error can be achieved. E.g. the 3-bit precision for $\eta = 1.0$ is divergent; for 4-bit precision, $\eta = 0.1$ achieves a much better error.

5.2 Effect of ρ on convergence

Theorem 1 states that for a fixed η, the convergence speed is faster for a fixed point operator with a larger value of ρ. For the Ridge-Regression problem with a gradient descent operator, this translates to a smaller condition number (i.e. better conditioning). Smaller condition number (larger λ) implies larger ρ and hence faster convergence. Moreover, Theorem 1 also implies that the larger ρ gets, the smaller the radius of $\|x^k - x^*\|^2$ gets at convergence.

In Figure 2 we compare FPMCI algorithm for Ridge-Regression problem with two values of $\lambda \in \{0.1, 0.001\}$. The convergence speed for $\lambda = 0.1$ is faster than in case of $\lambda = 0.001$. Moreover, for the same precision level, the FPMCI achieves better accuracy for larger λ.

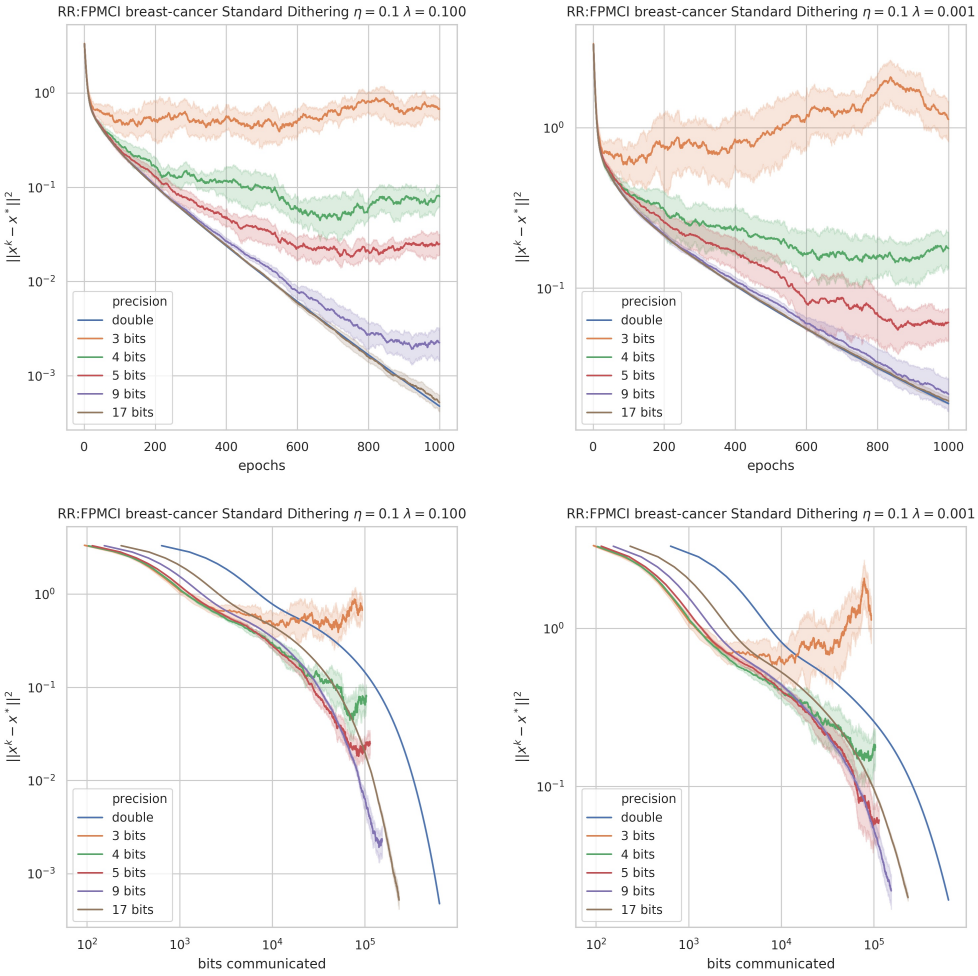

Figure 2: Convergence of FPMCI for Ridge-Regression with varying regularization parameter λ. The larger λ, the smaller the condition number of the problem and the larger the value of ρ. Theorem 1 implies that larger ρ would lead to faster convergence and lower the threshold achievable by $\|x^k - x^*\|^2$.

5.3 Effect of the compression operator $\mathcal{C}$ on convergence

Theorem 1 implies that FPMCI converges linearly up to a neighborhood whose radius is proportional to ω (in case of $B = 0$). The smaller the ω, the smaller the size of the neighborhood to which the algorithm converges. In Figure 3, we compare the evolution of $F(x^k) - F(x^*)$ on the *rcv1* dataset for both the Standard and Natural Dithering compression operators with various levels of precision (3-bit, 4-bit, 5-bit, 9-bit, and 17-bit). Natural Dithering achieves much smaller ω than Standard Dithering, and we can see that FPMCI is already convergent with 4-bit precision. In contrast, FPMCI with Standard Dithering is not convergent even with 5-bit precision. Note that for some precision levels, FPMCI diverges because η is larger than required by the theory. This shows that the relaxation parameter η is necessary for aggressive compression.

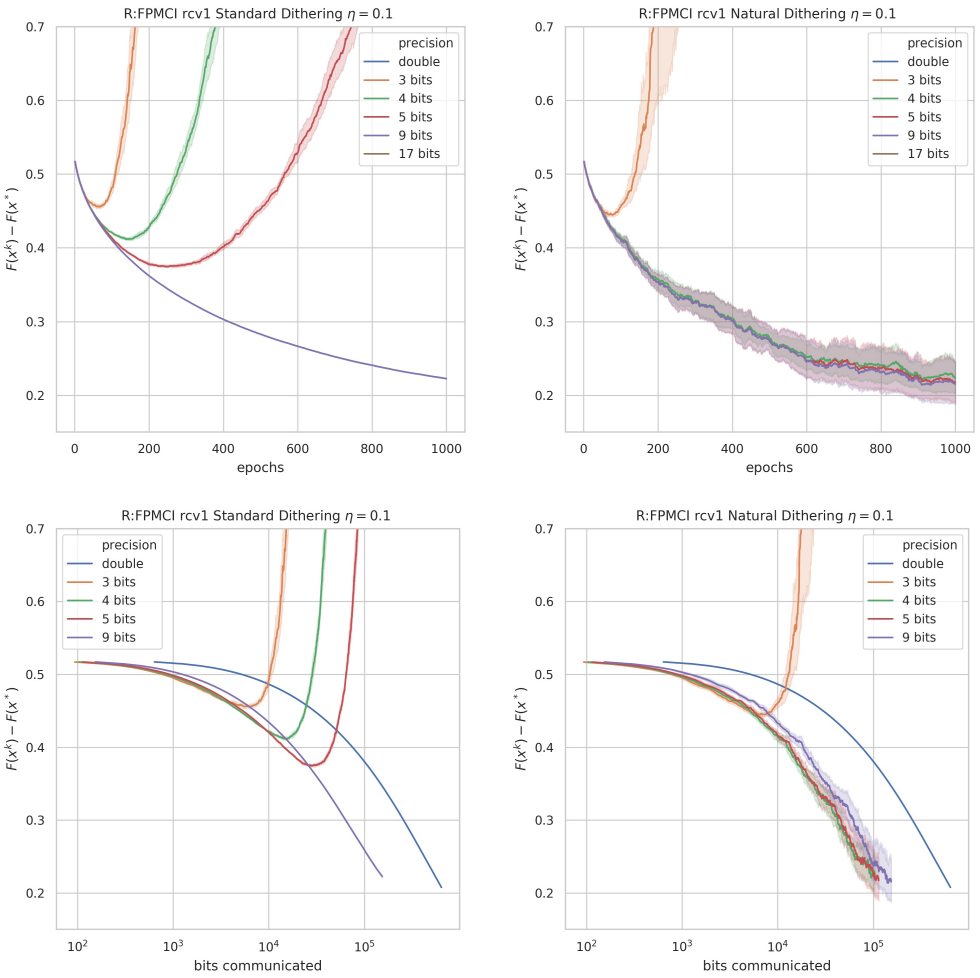

Figure 3: Comparison of convergence speed of FPMCI with Standard and Natural Dithering compressions. The smaller the variance of the compression operator, the smaller the radius of the neighborhood FPMCI converges to. Natural Dithering (right) has a much smaller variance (ω) for the same number of bits.

5.4 Variance reduced FPMCI

FPMCI can suffer from a large variance induced by the randomness of the compression operator ω. VR-FPMCI is designed to improve the convergence speed by reducing the effect of the variance of the compression operator. In Figure 4, we compare FPMCI with its variance reduced variant VR-FPMCI on a Ridge Regression problem. Note that Standard Dithering Compression has a large variance, and for 3-, 4-, and 5-bit precision runs, FPMCI converges slower, and the radius of the ball where it gets stuck is large. On the other hand, the trajectories of variance reduced VR-FPMCI are very close to each other for all levels of precision. When counting the number of bits communicated, the low precision compressions are superior.

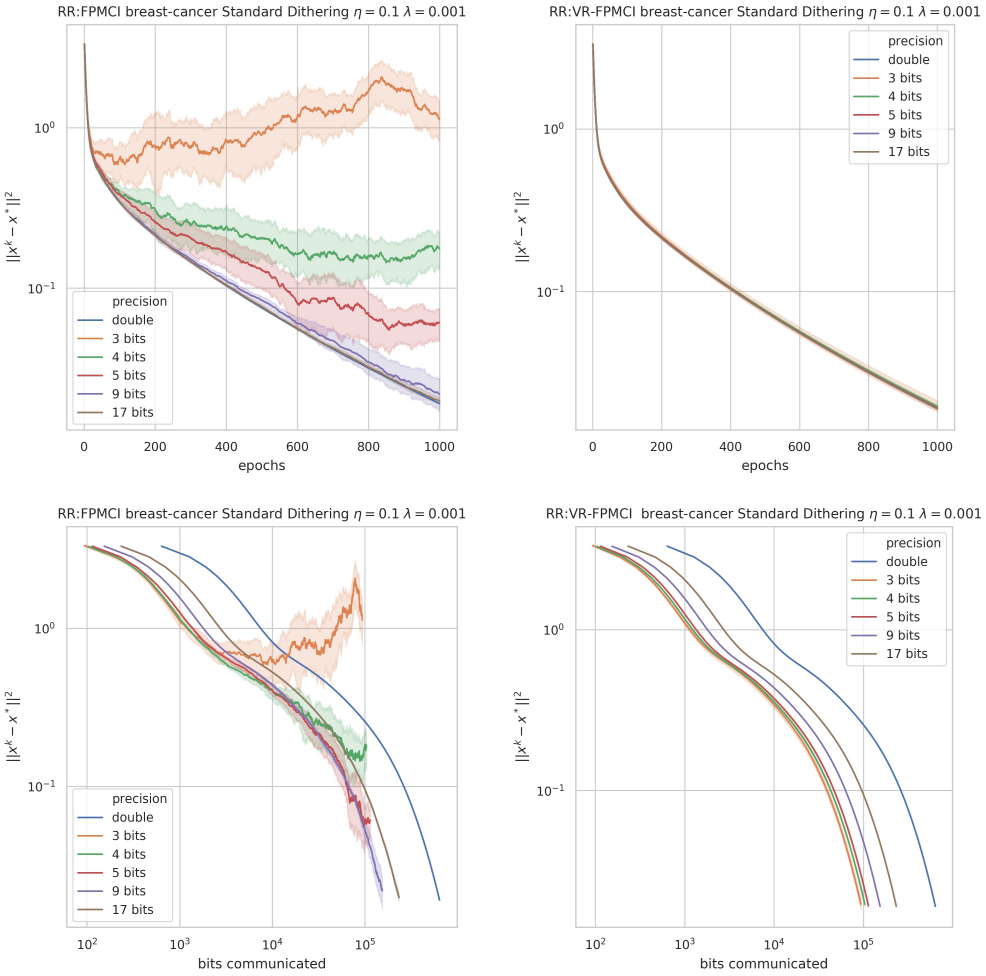

Figure 4: Comparison of FPMCI (left) with VR-FPMCI (right) for Ridge Regression with $\lambda = 0.001$. For both algorithms, we choose $\eta = 0.1$ and the Standard Dithering compression operator. The low precision runs converge slowly, and the radius they approach is significantly visible. On the other hand, for the variance-reduced methods all runs, with various compression levels, behave very similarly.

5.5 Distributed FPMCI

For Distributed FPMCI and VR-FPMCI, Theorems 3 and 4 indicates that both convergence speed and the radius of the neighborhood of convergence depends on the number of nodes n, with larger n corresponding to a faster convergence, a smaller radius, larger values of η to make convergence faster. In Figure 5, we show Distributed FPMCI for solving logistic regression problem with stochastic gradient descent. The plots clearly show that the size of the radius gets smaller as we increase the number of nodes n.

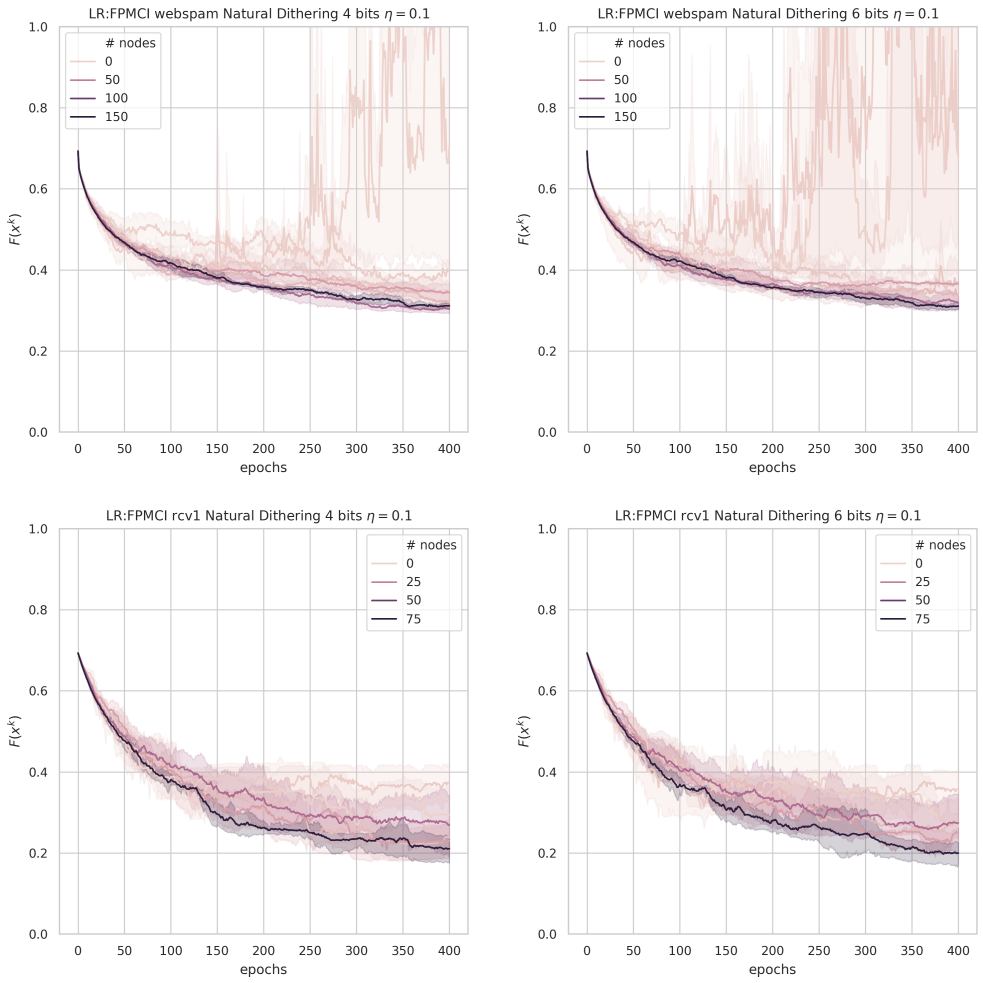

Figure 5: Distributed FPMCI for Logistic Regression with stochastic gradient as the number of nodes n varies. Having more nodes increases the accuracy achievable by FPMCI.

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

[2] Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-SGD: Distributed SGD with Quantization, Sparsification and Local Computations. In *Advances in Neural Information Processing Systems 32*, pages 14668–14679. Curran Associates, Inc., 2019.

[3] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

[4] Albert S Berahas, Charikleia Iakovidou, and Ermin Wei. Nested Distributed Gradient Methods with Adaptive Quantized Communication. *arXiv preprint arXiv:1903.08149*, 2019.

[5] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed Optimisation for Non-Convex Problems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 560–569, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[6] Sebastian Caldas, Jakub Konečny, H. Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

[7] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[8] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

[9] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[10] P. Chen, J. Huang, and X. Zhang. A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems*, 29(2), 2013.

[11] L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

[12] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms: A tour of recent advances, with new twists. *preprint arXiv:1912.00137*, 2019.

[13] L. Condat, G. Malinovsky, and P. Richtárik. Distributed proximal splitting algorithms with rates and acceleration. *preprint arXiv:2010.00952*, 2020.

[14] D. Davis and W. Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, 25(4):829–858, 2017.

[15] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, Aug 2014.

[16] Thinh T. Doan, Siva Theja Maguluri, and Justin Romberg. Fast Convergence Rates of Distributed Subgradient Methods with Adaptive Quantization. *arXiv preprint arXiv:1810.13245*, 2018.

[17] Y. Drori, S. Sabach, and M. Teboulle. A simple algorithm for a class of nonsmooth convex concave saddle-point problems. *Oper. Res. Lett.*, 43(2):209–214, 2015.

[18] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[19] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690, 2020.

[20] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General Analysis and Improved Rates. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5200–5209, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[21] Samuel Horvath, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. *arXiv preprint arXiv:1905.10988*, 2019.

[22] Samuel Horváth, Dmitry Kovalev, Konstantin Mishchenko, Sebastian Stich, and Peter Richtárik. Stochastic Distributed Learning with Gradient Quantization and Variance Reduction. *arxiv preprint arXiv:1904.05115*, 2019.

[23] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and Open Problems in Federated Learning. *arXiv preprint arXiv:1912.04977*, 2019.

[24] Ahmed Khaled and Peter Richtárik. Gradient Descent with Compressed Iterates. *arXiv preprint arXiv:1909.04716*, 2019.

[25] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized Stochastic Optimization and Gossip Algorithms with Compressed Communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine*

Learning, volume 97 of *Proceedings of Machine Learning Research*, pages 3478–3487, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[26] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

[27] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: distributed machine learning for on-device intelligence. *arXiv:1610.02527*, 2016.

[28] Jakub Konečný and Peter Richtárik. Randomized distributed mean estimation: accuracy vs communication. *Frontiers in Applied Mathematics and Statistics*, 4(62):1–11, 2018.

[29] Chang-Shen Lee, Nicolò Michelusi, and Gesualdo Scutari. Finite rate quantized distributed optimization with geometric convergence. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1876–1880. IEEE, 2018.

[30] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[31] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[32] I. Loris and C. Verhoeven. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27(12), 2011.

[33] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[34] Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[35] Angelia Nedic, Alex Olshevsky, Asuman Ozdaglar, and John N Tsitsiklis. Distributed subgradient methods and quantization effects. In *2008 47th IEEE Conference on Decision and Control*, pages 4177–4184. IEEE, 2008.

[36] Michael G Rabbat and Robert D Nowak. Quantized incremental algorithms for distributed optimization. *IEEE Journal on Selected Areas in Communications*, 23(4):798–808, 2005.

[37] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. FedPAQ: A Communication-Efficient Federated Learning Method with Periodic Averaging and Quantization. *arXiv preprint arXiv:1909.13014*, 2019.

[38] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. Quantized decentralized consensus optimization. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5838–5843. IEEE, 2018.

[39] Adil Salim, Laurent Condat, Konstantin Mishchenko, and Peter Richtárik. Dualize, split, randomize: Fast nonsmooth optimization algorithms. *arXiv preprint arXiv:2004.02635*, 2020.

[40] Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[41] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with Memory. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 4452–4463. Curran Associates, Inc., 2018.

[42] John N Tsitsiklis and Zhi-Quan Luo. Communication complexity of convex optimization. *Journal of Complexity*, 3(3):231–243, 1987.

[43] B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Applied Mathematics*, 38(3):667–681, 2013.

[44] Jianyu Wang and Gauri Joshi. Cooperative SGD: A Unified Framework for the Design and Analysis of Communication-Efficient SGD Algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[45] Steve Webb, James Caverlee, and Calton Pu. Introducing the webb spam corpus: Using email spam to identify web spam automatically. In *CEAS*, 2006.

[46] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 1509–1519. Curran Associates, Inc., 2017.

[47] Ming Yan. A new primal–dual algorithm for minimizing the sum of three functions with a linear operator. *Journal of Scientific Computing*, 76(3):1698–1717, 2018.

[48] Xin Zhang, Jia Liu, Zhengyuan Zhu, and Elizabeth S Bentley. Compressed Distributed Gradient Descent: Communication-Efficient Consensus over Networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2431–2439. IEEE, 2019.

Appendix

A Basic Facts

We recall the following fact about the variance of a random variable: Given a fixed $Y \in \mathbb{R}^d$ and a random variable $X \in \mathbb{R}^d$, we have

$$\mathbb{E}\left[\|X - Y\|^2\right] = \mathbb{E}\left[\|X - \mathbb{E}[X]\|^2\right] + \|\mathbb{E}[X] - Y\|^2. \tag{16}$$

If $X_1, X_2, \ldots, X_n$ are independent random variables then

$$\mathbb{E}\left[\left\|\sum_{i=1}^{n} X_i - \mathbb{E}[X_i]\right\|^2\right] = \sum_{i=1}^{n} \mathbb{E}\left[\|X_i - \mathbb{E}[X_i]\|^2\right]. \tag{17}$$

We also recall the following inequality from linear algebra: for any $a, b \in \mathbb{R}^d$ we have,

$$\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2. \tag{18}$$

We will also use the following fact: which follows from the convexity of the squared Euclidean norm: for $\eta \in [0, 1]$ we have,

$$\|\eta a + (1 - \eta) b\|^2 \leq \eta\|a\|^2 + (1 - \eta)\|b\|^2. \tag{19}$$

Moreover, we shall use the following lemma without mention.

Lemma 1. *Let $0 < A < 1$ and $B > 0$ and let $\{r_k\}_{k \geq 0}$ be a sequence of real numbers with $r_0 > 0$ satisfying the recursion*

$$r_{k+1} \leq A r_k + B.$$

Then

$$r_k \leq A^k r_0 + \frac{B}{1 - A}.$$

B Proof of Theorems 1 and 3

Since Theorem 1 is a particular case of Theorem 3, we only prove Theorem 3.

Proof of Theorem 3. From (13), conditionally on (x^k, s^k) we get

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] &= \mathbb{E}\left[\left\|(1-\eta)x^k+\eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}\left(\mathcal{T}_i(x^k,s_i^k);\xi_i^k\right)-x^\star\right\|^2\right] \\
&\overset{(16)}{=} \mathbb{E}\left[\left\|\eta\frac{1}{n}\sum_{i=1}^n \mathcal{C}\left(\mathcal{T}_i(x^k,s_i^k);\xi_i^k\right)-\eta\frac{1}{n}\sum_{i=1}^n \mathcal{T}_i(x^k,s_i^k)\right\|^2\right]+\left\|(1-\eta)x^k+\eta\frac{1}{n}\sum_{i=1}^n \mathcal{T}_i(x^k,s_i^k)-x^\star\right\|^2 \\
&\overset{(1)}{=} \frac{\eta^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n\left(\mathcal{C}\left(\mathcal{T}_i(x^k,s_i^k);\xi_i^k\right)-\mathcal{T}_i(x^k,s_i^k)\right)\right\|^2\right]+\left\|(1-\eta)x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2 \\
&\overset{(17)}{=} \underbrace{\frac{\eta^2}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathcal{C}\left(\mathcal{T}_i(x^k,s_i^k);\xi_i^k\right)-\mathcal{T}_i(x^k,s_i^k)\right\|^2\right]}_{A_1}+\underbrace{\left\|(1-\eta)x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2}_{A_2}. \qquad (20)
\end{aligned}$$

The first term in (20) can be bounded using Assumption 3 as follows:

$$\begin{aligned}
A_1 &\leq \frac{\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^k,s_i^k)\right\|^2 \\
&\overset{(18)}{\leq} \frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^k,s_i^k)-\mathcal{T}_i(x^\star,s_i^k)\right\|^2+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2 \\
&\overset{(6)}{\leq} \frac{2\eta^2\omega}{n^2}\sum_{i=1}^n c_i^2\left\|x^k-x^\star\right\|^2+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2 \\
&= \frac{2\eta^2\omega c^2}{n}\left\|x^k-x^\star\right\|^2+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2. \qquad (21)
\end{aligned}$$

The second term in (20) can be bounded using the convexity of the squared norm:

$$A_2=\left\|(1-\eta)x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2 \leq (1-\eta)\left\|x^k-x^*\right\|^2+\eta\left\|\mathcal{T}(x^k,s^k)-x^*\right\|^2. \qquad (22)$$

Plugging (21) and (22) in (20), we get

$$\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] \leq \frac{2\eta^2\omega c^2}{n}\left\|x^k-x^\star\right\|^2+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2+(1-\eta)\left\|x^k-x^*\right\|^2+\eta\left\|\mathcal{T}(x^k,s^k)-x^*\right\|^2.$$

Therefore, conditionally on x^k,

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] &\leq \left(1-\eta+\frac{2\eta^2\omega c^2}{n}\right)\left\|x^k-x^\star\right\|^2+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right]+\eta\cdot\mathbb{E}\left[\left\|\mathcal{T}(x^k,s^k)-x^\star\right\|^2\right] \\
&\overset{(5)}{\leq} \left(1-\eta+\eta(1-\rho)+\frac{2\eta^2\omega c^2}{n}\right)\left\|x^k-x^\star\right\|^2+\eta B+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] \\
&= \left(1-\eta\rho+\frac{2\eta^2\omega c^2}{n}\right)\left\|x^k-x^\star\right\|^2+\eta B+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right].
\end{aligned}$$

Recall that we assume the stepsize $\eta>0$ satisfies $\eta\leq\min\left(1,\frac{\rho n}{4\omega c^2}\right)$. Using this we get

$$\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] \leq \left(1-\frac{\eta\rho}{2}\right)\left\|x^k-x^\star\right\|^2+\eta B+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right].$$

Finally, taking unconditional expectations yields the theorem's claim. □

C Proof of Theorems 2 and 4

Since Theorem 2 is a particular case of Theorem 4, we only prove Theorem 4.

C.1 Two lemmas for Algorithm 4

Lemma 2. *Under Assumption 3, if* $0 < \alpha \leq \frac{1}{\omega+1}$, *then for every* $i = \{1, \dots, n\}$ *the iterates of Algorithm 4 satisfy conditionally on* x^k *and* h_i^k*:*

$$\mathbb{E}\left[\left\|h_i^{k+1} - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right] \leq (1-\alpha)\,\mathbb{E}\left[\left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right] + \alpha\mathbb{E}\left[\left\|\mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right]. \tag{23}$$

Proof. Conditionally on $x^k, h_1^k, \dots, h_n^k, s_1^k, \dots, s_n^k$ we have

$$\begin{aligned}
\mathbb{E}\left[\left\|h_i^{k+1} - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right] &= \mathbb{E}\left[\left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k) + \alpha\delta_i^k\right\|^2\right] \\
&= \left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2 + 2\alpha\left\langle h_i^k - \mathcal{T}_i(x^\star, s_i^k), \mathbb{E}\left[\delta_i^k\right]\right\rangle + \alpha^2\mathbb{E}\left[\left\|\delta_i^k\right\|^2\right] \\
&\leq \left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2 + 2\alpha\left\langle h_i^k - \mathcal{T}_i(x^\star, s_i^k), \mathcal{T}_i(x^k, s_i^k) - h_i^k\right\rangle + \alpha^2(\omega+1)\left\|\mathcal{T}_i(x^k, s_i^k) - h_i^k\right\|^2 \\
&\leq \left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2 + 2\alpha\left\langle h_i^k - \mathcal{T}_i(x^\star, s_i^k), \mathcal{T}_i(x^k, s_i^k) - h_i^k\right\rangle + \alpha\left\|\mathcal{T}_i(x^k, s_i^k) - h_i^k\right\|^2 \\
&= \underbrace{\alpha\left\langle 2h_i^k - 2\mathcal{T}_i(x^\star, s_i^k) + \mathcal{T}_i(x^k, s_i^k) - h_i^k, \mathcal{T}_i(x^k, s_i^k) - h_i^k\right\rangle}_{\mathcal{I}} + \left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2.
\end{aligned}$$

For the inner product $\mathcal{I}$ in the last inequality, we have

$$\begin{aligned}
\mathcal{I} &= \left\langle h_i^k - \mathcal{T}_i(x^\star, s_i^k) + \mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k), \mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k) - \left(h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right)\right\rangle \\
&= -\left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2 + \left\|\mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k)\right\|^2.
\end{aligned}$$

Using this in the previous inequality, we get

$$\mathbb{E}\left[\left\|h_i^{k+1} - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right] = (1-\alpha)\left\|h_i^k - \mathcal{T}_i(x^\star, s_i^k)\right\|^2 + \alpha\left\|\mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k)\right\|^2.$$

It remains to take expectation with respect to the randomness in s_i^k. □

Lemma 3. *Under Assumptions 1 and 3, the iterates of Algorithm 4 satisfy,*

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{k+1} - x^\star\right\|^2\right] \leq\ & (1-\eta\rho)\left\|x^k - x^\star\right\|^2 + \eta B + \mathbb{E}\left[\left\|\mathcal{T}_i(x^\star, s_i^k) - h_i^k\right\|^2\right] \\
& + \frac{2\eta^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathcal{T}_i(x^k, s_i^k) - \mathcal{T}_i(x^\star, s_i^k)\right\|^2\right].
\end{aligned} \tag{24}$$

Proof. Conditionally on $x^k, h_1^k, \ldots, h_n^k, s_1^k, \ldots, s_n^k$ we have,

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] &= \mathbb{E}\left[\left\|(1-\eta)\,x^k+\frac{\eta}{n}\sum_{i=1}^n\left(\delta_i^k+h_i^k\right)-x^\star\right\|^2\right]\\
&\overset{(16)}{=} \left\|(1-\eta)\,x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2+\frac{\eta^2}{n^2}\mathbb{E}\left[\left\|\sum_{i=1}^n\delta_i^k-\mathbb{E}\left[\delta_i^k\right]\right\|^2\right]\\
&\overset{(17)}{=} \left\|(1-\eta)\,x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2+\frac{\eta^2}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\delta_i^k-\mathbb{E}\left[\delta_i^k\right]\right\|^2\right]\\
&\leq \left\|(1-\eta)\,x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2+\frac{\eta^2\omega}{n^2}\sum_{i=1}^n\left\|\mathcal{T}_i(x^k,s_i^k)-h_i^k\right\|^2.
\end{aligned}$$

We now take expectation with respect to the randomness in $s_1^k,\ldots,s_n^k$ and conditionally on $x^k, h_1^k, \ldots, h_n^k$:

$$\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right]\leq \mathbb{E}\left[\left\|(1-\eta)\,x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2\right]+\frac{\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^k,s_i^k)-h_i^k\right\|^2\right]. \tag{25}$$

To bound the first term in (25) we use the convexity of the squared norm as follows,

$$\begin{aligned}
\mathbb{E}\left[\left\|(1-\eta)\,x^k+\eta\mathcal{T}(x^k,s^k)-x^\star\right\|^2\right] &= \mathbb{E}\left[\left\|(1-\eta)\left(x^k-x^\star\right)+\eta\left(\mathcal{T}(x^k,s^k)-x^\star\right)\right\|^2\right]\\
&\overset{(19)}{\leq} (1-\eta)\left\|x^k-x^\star\right\|^2+\eta\mathbb{E}\left[\left\|\mathcal{T}(x^k,s^k)-x^\star\right\|^2\right]\\
&\overset{(5)}{\leq} (1-\eta+\eta\,(1-\rho))\left\|x^k-x^\star\right\|^2+\eta B\\
&= (1-\eta\rho)\left\|x^k-x^\star\right\|^2+\eta B.
\end{aligned} \tag{26}$$

For the second term in (25) we have,

$$\mathbb{E}\left[\left\|\mathcal{T}_i(x^k,s_i^k)-h_i^k\right\|^2\right]\overset{(18)}{\leq} 2\mathbb{E}\left[\left\|\mathcal{T}_i(x^k,s_i^k)-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right]+2\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)-h_i^k\right\|^2\right]. \tag{27}$$

It remains to substitute with (26) and (27) in (25):

$$\begin{aligned}
\mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] \leq\ & (1-\eta\rho)\left\|x^k-x^\star\right\|^2+\eta B+\mathbb{E}\left[\left\|\mathcal{T}_i(x^\star,s_i^k)-h_i^k\right\|^2\right]\\
&+\frac{2\eta^2\omega}{n^2}\sum_{i=1}^n\mathbb{E}\left[\left\|\mathcal{T}_i(x^k,s_i^k)-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right].
\end{aligned}$$

□

C.2 Proof of Theorem 4

Proof of Theorem 4. By Lemmas 3 and 2 taking conditional expectation w.r.t. $x^k, h_1^k, \ldots, h_n^k$,

$$\begin{aligned}
\mathbb{E}\left[\Psi^{k+1}\right] &= \mathbb{E}\left[\left\|x^{k+1}-x^\star\right\|^2\right] + \frac{4\eta^2\omega}{\alpha n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|h_i^{k+1}-x^\star\right\|^2\right] \\
&\overset{(24)+(23)}{\leq} (1-\eta\rho)\left\|x^k-x^\star\right\|^2 + \frac{6\eta^2\omega}{n^2}\sum_{i=1}^n \mathbb{E}\left[\left\|\mathcal{T}_i(x^k,s_i^k)-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] \\
&\quad + \frac{4\eta^2\omega}{\alpha n^2}\left(1-\frac{\alpha}{2}\right)\sum_{i=1}^n \mathbb{E}\left[\left\|h_i^k-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] + \eta B \\
&\overset{(6)}{\leq} (1-\eta\rho)\left\|x^k-x^\star\right\|^2 + \frac{6\eta^2\omega}{n^2}\sum_{i=1}^n c_i^2\cdot\left\|x^k-x^\star\right\|^2 \\
&\quad + \frac{4\eta^2\omega}{\alpha n^2}\left(1-\frac{\alpha}{2}\right)\sum_{i=1}^n \mathbb{E}\left[\left\|h_i^k-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] + \eta B \\
&= \left(1-\eta\rho+\frac{6\eta^2\omega c^2}{n}\right)\left\|x^k-x^\star\right\|^2 + \eta B + \frac{4\eta^2\omega}{\alpha n^2}\left(1-\frac{\alpha}{2}\right)\sum_{i=1}^n \mathbb{E}\left[\left\|h_i^k-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right]. \quad (28)
\end{aligned}$$

To get the optimal stepsize $\eta \in (0,1]$ we solve

$$\min_{\eta\in(0,1]}\left\{1-\eta\rho+\frac{6\eta^2\omega c^2}{n}\right\},$$

One can observe that the solution of this problem is the value of η in Theorem 4. Using this choice of η we get

$$1-\eta\rho+\frac{6\eta^2\omega c^2}{n} = 1-\frac{\eta\rho}{2}-\frac{\eta\rho}{2}\left(1-\frac{12\eta\omega c^2}{n\rho}\right) \leq 1-\frac{\eta\rho}{2}. \quad (29)$$

Hence using (29) in (28),

$$\begin{aligned}
\mathbb{E}\left[\Psi^{k+1}\right] &\leq \left(1-\frac{\eta\rho}{2}\right)\left\|x^k-x^\star\right\|^2 + \eta B + \frac{4\eta^2\omega}{\alpha n^2}\left(1-\frac{\alpha}{2}\right)\sum_{i=1}^n \mathbb{E}\left[\left\|h_i^k-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] \\
&\leq \max\left\{1-\frac{\eta\rho}{2}, 1-\frac{\alpha}{2}\right\}\mathbb{E}\left[\left\|x^k-x^\star\right\|^2 + \frac{4\eta^2\omega}{\alpha n^2}\sum_{i=1}^n\left\|h_i^k-\mathcal{T}_i(x^\star,s_i^k)\right\|^2\right] + \eta B \\
&= \left(1-\frac{\min\{\alpha,\eta\rho\}}{2}\right)\Psi^k + \eta B. \quad (30)
\end{aligned}$$

It remains to take unconditional expectations in (30), yielding (15). □