LoCoDL: Communication-Efficient Distributed Learning with Local Training and Compression

Laurent Condat HTTPS://LCONDAT.GITHUB.IO/
Artavazd Maranjyan
Peter Richtárik
Center of Excellence for Generative AI, King Abdullah University of Science and Technology (KAUST)
& SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence (SDAIA-KAUST AI)
Thuwal, Kingdom of Saudi Arabia

Abstract

In Distributed optimization and Learning, and even more in the modern framework of federated learning, communication, which is slow and costly, is critical. We introduce LoCoDL, a communication-efficient algorithm that leverages the two popular and effective techniques of Local training, which reduces the communication frequency, and Compression, in which short bitstreams are sent instead of full-dimensional vectors of floats. LoCoDL works with a large class of unbiased compressors that includes widely-used sparsification and quantization methods. LoCoDL provably benefits from local training and compression and enjoys a doubly-accelerated communication complexity, with respect to the condition number of the functions and the model dimension, in the general heterogenous regime with strongly convex functions. This is confirmed in practice, with LoCoDL outperforming existing algorithms.

1. Introduction

Distributed computations are now pervasive in all areas of science. Notably, Federated Learning (FL) involves training machine learning models in a distributed, collaborative way [5, 26, 27, 34]. The key idea is to exploit information stored on distant devices, such as mobile phones or hospital workstations. Challenges in FL include data privacy, robustness to adversarial attacks, and, critically, communication efficiency [24, 28, 48]. Unlike centralized settings, FL clients perform computations locally and communicate with a distant server, often over slow, costly, and unreliable networks, creating a bottleneck for large-scale deployment.

To reduce communication burden, two strategies have emerged: 1) **Local Training (LT)**, which reduces communication frequency by performing multiple computation steps between rounds; and 2) **Communication Compression (CC)**, where compressed information is sent instead of full-dimensional vectors. We review LT and CC literature in Section A.

We propose a new randomized algorithm, LoCoDL, integrating LT and unbiased CC for communication efficient FL and distributed optimization. This variance-reduced algorithm [13, 16, 20] converges to an exact solution and benefits from both strategies, achieving a doubly accelerated communication complexity with improved dependence on the condition number and model dimension.

1.1. Problem and Motivation

We study distributed optimization problems of the form

$$\min_{x\in\mathbb{R}^d}\ \tfrac{1}{n}\textstyle\sum_{i=1}^n f_i(x) + g(x), \tag{1}$$

where $d \geq 1$ is the model dimension, and $f_i : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$ are *smooth*. We consider a server-client model with n clients computing in parallel and communicating with a server. The function f_i is owned by client i. Problem (1) models empirical risk minimization, important in machine learning [44, 45], and minimizing a sum of functions is common in various fields. Our goal is to solve Problem (1) communication-efficiently in a **heterogeneous** setting where f_i and g can be *arbitrarily different*.

We focus on the strongly convex setting, assuming:

Assumption 1 (strongly convex functions) *The functions f_i and g are L-smooth and μ-strongly convex, for some $0 < \mu \leq L$. The unique solution of Problem (1) is denoted by $x^\star$, with condition number $\kappa := L/\mu$.*

Problem (1) minimizes the average of n functions (f_i+g), computed via $\nabla(f_i+g) = \nabla f_i+\nabla g$. We illustrate the significance of the **additional function** g through four viewpoints, noting that we can handle $g = 0$ as discussed in Section 3.1.

• *Regularization.* The function g can act as a regularizer, such as $g = \frac{\mu}{2}\|\cdot\|^2$ for small $\mu > 0$, making the problem μ-strongly convex.

• *Shared Dataset.* The function g can represent the cost of a common dataset known to all clients.

• *Server-Aided Training.* The function g can model the cost of a core dataset known only to the server, as explored in several works. However, we focus on the heterogeneous setting where g and f_i are not necessarily similar.

• *New Mathematical and Algorithmic Principle.* This concept led to the construction of LoCoDL, detailed in Section 2.1.

In LoCoDL, clients make all gradient calls, computing ∇f_i and ∇g. A variation where the server makes calls to ∇g is provided in Appendix.

1.2. A General Class of Unbiased Random Compressors

We define the set $\mathbb{U}(\omega)$ of random compression operators $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ that are unbiased, satisfying $\mathbb{E}[\mathcal{C}(x)] = x$, and for every $x \in \mathbb{R}^d$,

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

Given a collection $(\mathcal{C}i)i = 1^n$ of compression operators in $\mathbb{U}(\omega)$, we introduce the constant $\omega_{\mathrm{av}} \geq 0$ to characterize their joint variance, satisfying

$$\mathbb{E}\left[\left\|\tfrac{1}{n}\textstyle\sum_{i=1}^n \left(\mathcal{C}i(x_i) - x_i\right)\right\|^2\right] \leq \tfrac{\omega_{\mathrm{av}}}{n} \textstyle\sum i = 1^n \left\|x_i\right\|^2. \tag{3}$$

This inequality holds with $\omega_{\mathrm{av}} = \omega$ due to the convexity of the squared norm. The convergence rate depends on ω_{av}, typically smaller than ω. If the compressors $\mathcal{C}_i$ are mutually independent, then (3) holds with $\omega_{\mathrm{av}} = \frac{\omega}{n}$.

1.3. Challenge and Contributions

This work addresses the question: *Can we combine LT and CC with compressors from the generic class $\mathbb{U}(\omega)$ to achieve a doubly-accelerated communication complexity?*

We answer affirmatively. LoCoDL utilizes the same probabilistic LT mechanism as Scaffnew and incorporates CC with compressors in $\mathbb{U}(\omega)$, ensuring linear convergence under Assumption 1 without additional requirements. By optimizing communication probability and variance ω, we achieve double acceleration. Consequently, LoCoDL matches the theoretical complexity of Compressed-Scaffnew while supporting a broader range of compressors than its cumbersome permutation-based counterpart. Notably, with compressors that enable sparsification and quantization, LoCoDL outperforms existing algorithms, as demonstrated in Section B. While ADIANA offers better theoretical complexity for $n > d$ (see Table 1), it underperforms compared to LoCoDL in practice. Thus, LoCoDL establishes new standards for communication efficiency.

2. Proposed Algorithm LoCoDL

2.1. Principle: Double Lifting of the Problem to a Consensus Problem

In LoCoDL, every client stores and updates *two* local model estimates. They will all converge to the same solution $x^\star$ of (1). This construction comes from two ideas.

Local steps with local models. In algorithms making use of LT, such as FedAvg, Scaffold and Scaffnew, the clients store and update local model estimates x_i. When communication occurs, an estimate of their average is formed by the server and broadcast to all clients. They all resume their computations with this new model estimate.

Compressing the difference between two estimates. To implement CC, a powerful idea is to compress not the vectors themselves, but *difference vectors* that converge to zero. This way, the algorithm is variance-reduced; that is, the compression error vanishes at convergence. The technique of compressing the difference between a gradient vector and a control variate is at the core of algorithms such as DIANA and EF21. Here, we want to compress differences between model estimates, not gradient estimates. That is, we want Client i to compress the difference between x_i and another model estimate that converges to the solution $x^\star$ as well. We see the need of an additional model estimate that plays the role of an anchor for compression. This is the variable y common to all clients in LoCoDL, which compress $x_i - y$ and send these compressed differences to the server.

Combining the two ideas. Accordingly, an equivalent reformulation of (1) is the consensus problem with $n+1$ variables

$$\min_{x_1,\ldots,x_n,y} \ \frac{1}{n}\sum_{i=1}^n f_i(x_i) + g(y) \ \text{ s.t. } \ x_1 = \cdots = x_n = y.$$

The primal–dual optimality conditions are $x_1 = \cdots = x_n = y$, $0 = \nabla f_i(x_i) - u_i \ \forall i \in [n]$, $0 = \nabla g(y) - v$, and $0 = u_1 + \cdots + u_n + nv$ (dual feasibility), for some dual variables $u_1, \ldots, u_n, v$ introduced in LoCoDL, that always satisfy the dual feasibility condition.

2.2. Description of LoCoDL

LoCoDL is a randomized primal–dual algorithm, shown as Algorithm 1. At every iteration, for every $i \in [n]$ in parallel, Client i first constructs a prediction $\hat{x}_i^t$ of its updated local model estimate, using

Algorithm 1 LoCoDL

```
1:  input: stepsizes $\gamma > 0$, $\chi > 0$, $\rho > 0$; probability $p \in (0,1]$; variance factor $\omega \geq 0$; local initial
    estimates $x_1^0, \dots, x_n^0 \in \mathbb{R}^d$, initial estimate $y^0 \in \mathbb{R}^d$, initial control variates $u_1^0, \dots, u_n^0 \in \mathbb{R}^d$
    and $v^0 \in \mathbb{R}^d$ such that $\frac{1}{n}\sum_{i=1}^n u_i^0 + v^0 = 0$.
2:  for $t = 0, 1, \dots$ do
3:    for $i = 1, \dots, n$, at clients in parallel, do
4:      $\hat{x}_i^t := x_i^t - \gamma \nabla f_i(x_i^t) + \gamma u_i^t$
5:      $\hat{y}^t := y^t - \gamma \nabla g(y^t) + \gamma v^t$   // the clients store and update identical copies of $y^t$, $v^t$, $\hat{y}^t$
6:      flip a coin $\theta^t \in \{0,1\}$ with $\mathrm{Prob}(\theta^t = 1) = p$
7:      if $\theta^t = 1$ then
8:        $d_i^t := \mathcal{C}_i^t\big(\hat{x}_i^t - \hat{y}^t\big)$
9:        send $d_i^t$ to the server
10:       at server: aggregate $\bar{d}^t := \frac{1}{2n}\sum_{j=1}^n d_j^t$ and broadcast $\bar{d}^t$ to all clients
11:       $x_i^{t+1} := (1-\rho)\hat{x}_i^t + \rho(\hat{y}^t + \bar{d}^t)$
12:       $u_i^{t+1} := u_i^t + \frac{p\chi}{\gamma(1+2\omega)}\big(\bar{d}^t - d_i^t\big)$
13:       $y^{t+1} := \hat{y}^t + \rho \bar{d}^t$
14:       $v^{t+1} := v^t + \frac{p\chi}{\gamma(1+2\omega)}\bar{d}^t$
15:     else
16:       $x_i^{t+1} := \hat{x}_i^t$, $y^{t+1} = \hat{y}^t$, $u_i^{t+1} := u_i^t$, $v^{t+1} := v^t$
17:     end if
18:   end for
19: end for
```

a GD step with respect to f_i corrected by the dual variable u_i^t. It also constructs a prediction $\hat{y}^t$ of the updated model estimate, using a GD step with respect to g corrected by the dual variable v^t. Since g is known by all clients, they all maintain and update identical copies of the variables y and v. If there is no communication, which is the case with probability $1-p$, x_i and y are updated with these predicted estimates, and the dual variables u_i and v are unchanged. If communication occurs, which is the case with probability p, the clients compress the differences $\hat{x}_i^t - \hat{y}^t$ and send these compressed vectors to the server, which forms $\bar{d}^t$ equal to one half of their average. Then the variables x_i are updated using a convex combination of the local predicted estimates $\hat{x}_i^t$ and the global but noisy estimate $\hat{y}^t + \bar{d}^t$. y is updated similarly. Finally, the dual variables are updated using the compressed differences minus their weighted average, so that the dual feasibility condition remains satisfied. The model estimates x_i^t, $\hat{x}_i^t$, y^t, $\hat{y}^t$ all converge to $x^\star$, so that their differences, as well as the compressed differences as a consequence of (2), converge to zero. This is the key property that makes the algorithm variance-reduced. We consider the following assumption.

Assumption 2 (class of compressors) *In LoCoDL the compressors $\mathcal{C}_i^t$ are all in $\mathbb{U}(\omega)$ for some $\omega \geq 0$. Moreover, for every $i \in [n]$, $i' \in [n]$, $t \geq 0$, $t' \geq 0$, $\mathcal{C}_i^t$ and $\mathcal{C}_{i'}^{t'}$ are independent if $t \neq t'$ ($\mathcal{C}_i^t$ and $\mathcal{C}_{i'}^t$ at the same iteration t need not be independent). We define $\omega_{\mathrm{av}} \geq 0$ such that for every $t \geq 0$, the collection $(\mathcal{C}_i^t)_{i=1}^n$ satisfies (3).*

Remark 3 (partial participation) *LoCoDL allows for a form of partial participation if we set $\rho = 1$. Indeed, in that case, at steps 11 and 13 of the algorithm, all local variables x_i as well as the common variable y are overwritten by the same up-to-date model $\hat{y}^t + \bar{d}^t$. So, it does not matter that* *for a non-participating client i with $d_i^t = 0$, the $\hat{x}_i^{t'}$ were not computed for the $t' \leq t$ since its last participation, as they are not used in the process. However, a non-participating client should still update its local copy of y at every iteration. This can be done when ∇g is much cheaper to compute that ∇f_i, as is the case with $g = \frac{\mu}{2}\|\cdot\|^2$. A non-participating client can be completely idle for a certain period of time, but when it resumes participating, it should receive the last estimates of x, y and v from the server as it lost synchronization.*

3. Convergence and Complexity of LoCoDL

Theorem 4 (linear convergence of LoCoDL) *Suppose that Assumptions 1 and 2 hold. In LoCoDL, suppose that $0 < \gamma < \frac{2}{L}$, $2\rho - \rho^2(1+\omega_{\text{av}}) - \chi \geq 0$. For every $t \geq 0$, define the Lyapunov function*

$$\Psi^t := \frac{1}{\gamma}\left(\sum_{i=1}^n \left\|x_i^t - x^\star\right\|^2 + n\left\|y^t - x^\star\right\|^2\right) + \frac{\gamma(1+2\omega)}{p^2\chi}\left(\sum_{i=1}^n \left\|u_i^t - u_i^\star\right\|^2 + n\left\|v^t - v^\star\right\|^2\right), \tag{4}$$

where $v^\star := \nabla g(x^\star)$ and $u_i^\star := \nabla f_i(x^\star)$. Then LoCoDL converges linearly: for every $t \geq 0$,

$$\mathbb{E}\left[\Psi^t\right] \leq \tau^t \Psi^0, \quad \textit{where} \quad \tau := \max\left((1-\gamma\mu)^2, (1-\gamma L)^2, 1 - \frac{p^2\chi}{1+2\omega}\right) < 1. \tag{5}$$

In addition, for every $i \in [n]$, $(x_i^t)_{t\in\mathbb{N}}$ and $(y^t)_{t\in\mathbb{N}}$ converge to $x^\star$, $(u_i^t)_{t\in\mathbb{N}}$ converges to $u_i^\star$, and $(v^t)_{t\in\mathbb{N}}$ converges to $v^\star$, almost surely.

We place ourselves in the conditions of Theorem 4. We observe that in (5), the larger χ, the better, so given ρ we should set $\chi = 2\rho - \rho^2(1+\omega_{\text{av}})$. Then, choosing ρ to maximize χ yields

$$\chi = \rho = \frac{1}{1+\omega_{\text{av}}}. \tag{6}$$

We now study the complexity of LoCoDL with χ and ρ chosen as in (6) and $\gamma = \Theta(\frac{1}{L})$. We remark that LoCoDL has the same rate $\tau^\sharp := \max(1-\gamma\mu, \gamma L - 1)^2$ as mere distributed gradient descent, as long as p^{-1}, ω and ω_{av} are small enough to have $1 - \frac{p^2\chi}{1+2\omega} \leq \tau^\sharp$. This is remarkable: communicating with a low frequency and compressed vectors does not harm convergence at all, until some threshold.

The iteration complexity of LoCoDL to reach ϵ-accuracy, i.e. $\mathbb{E}\left[\Psi^t\right] \leq \epsilon\Psi^0$, is

$$\mathcal{O}\left(\left(\kappa + \frac{(1+\omega_{\text{av}})(1+\omega)}{p^2}\right)\log \epsilon^{-1}\right). \tag{7}$$

By choosing

$$p = \min\left(\sqrt{\frac{(1+\omega_{\text{av}})(1+\omega)}{\kappa}}, 1\right), \tag{8}$$

the iteration complexity becomes $\mathcal{O}\Big(\big(\kappa + \omega(1+\omega_{\text{av}})\big)\log\epsilon^{-1}\Big)$ and the communication complexity in number of communication rounds is p times the iteration complexity, that is

$$\mathcal{O}\left(\left(\sqrt{\kappa(1+\omega_{\text{av}})(1+\omega)} + \omega(1+\omega_{\text{av}})\right)\log\epsilon^{-1}\right).$$

If the compressors are mutually independent, $\omega_{\text{av}} = \frac{\omega}{n}$ and the communication complexity can be equivalently written as

$$\mathcal{O}\left(\left(\left(1 + \sqrt{\omega} + \frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega\left(1 + \frac{\omega}{n}\right)\right)\log\epsilon^{-1}\right),$$

as shown in Table 2.

Let us consider the example of independent rand-k compressors, for some $k \in [d]$. We have $\omega = \frac{d}{k} - 1$. Therefore, the communication complexity in numbers of reals is k times the complexity in number of rounds; that is, $\mathcal{O}\left(\left(\left(\sqrt{kd} + \frac{d}{\sqrt{n}}\right)\sqrt{\kappa} + d\left(1 + \frac{d}{kn}\right)\right)\log \epsilon^{-1}\right)$. We can now choose k to minimize this complexity: with $k = \lceil \frac{d}{n} \rceil$, it becomes $\mathcal{O}\left(\left(\left(\sqrt{d} + \frac{d}{\sqrt{n}}\right)\sqrt{\kappa} + d\right)\log \epsilon^{-1}\right)$, as shown in Table 1. Let us state this result:

Corollary 5 *In the conditions of Theorem 4, suppose in addition that the compressors $\mathcal{C}_i^t$ are independent* rand-k *compressors with $k = \lceil \frac{d}{n} \rceil$. Suppose that $\gamma = \Theta(\frac{1}{L})$, $\chi = \rho = \frac{n}{n-1+d/k}$, and*

$$p = \min\left(\sqrt{\frac{dk(n-1)+d^2}{nk^2\kappa}}, 1\right). \tag{9}$$

Then the uplink communication complexity in number of reals of LoCoDL *is*

$$\mathcal{O}\left(\left(\sqrt{d}\sqrt{\kappa} + \frac{d\sqrt{\kappa}}{\sqrt{n}} + d\right)\log \epsilon^{-1}\right). \tag{10}$$

This is the same complexity as CompressedScaffnew [9]. However, it is obtained with simple independent compressors, which is much more practical than the permutation-based compressors with shared randomness of CompressedScaffnew. Moreover, this complexity can be obtained with other types of compressors, and further reduced, when reasoning in number of bits and not only reals, by making use of quantization [1], as we illustrate by experiments in the next section.

We can distinguish 2 regimes:

1. In the "large d small n" regime, i.e. $n = \mathcal{O}(d)$, the communication complexity of LoCoDL in (10) becomes $\mathcal{O}\left(\left(\frac{d\sqrt{\kappa}}{\sqrt{n}} + d\right)\log \epsilon^{-1}\right)$. This is the state of the art, as reported in Table 1.

2. In the "large n small d" regime, i.e. $n = \Omega(d)$, the communication complexity of LoCoDL in (10) becomes $\mathcal{O}\left(\left(\sqrt{d}\sqrt{\kappa} + d\right)\log \epsilon^{-1}\right)$. If n is even larger with $n = \Omega(d^2)$, ADIANA achieves the even better complexity $\mathcal{O}\left((\sqrt{\kappa} + d)\log \epsilon^{-1}\right)$.

Yet, in the experiments we ran with different datasets and values of d, n, κ, LoCoDL outperforms the other algorithms, including ADIANA, in all cases.

3.1. The Case $g = 0$

We have assumed the presence of a function g in Problem (1), whose gradient is called by all clients. In this section, we show that we can handle the case where such a function is not available. So, let us assume that we want to minimize $\frac{1}{n}\sum_{i=1}^n f_i$, with the functions f_i satisfying Assumption 1. We now define the functions $\tilde{f}_i := f_i - \frac{\mu}{4}\left\|\cdot\right\|^2$ and $\tilde{g} := \frac{\mu}{4}\left\|\cdot\right\|^2$. They are all $\tilde{L}$-smooth and $\tilde{\mu}$-strongly convex, with $\tilde{L} := L - \frac{\mu}{2}$ and $\tilde{\mu} := \frac{\mu}{2}$. Moreover, it is equivalent to minimize $\frac{1}{n}\sum_{i=1}^n f_i$ or $\frac{1}{n}\sum_{i=1}^n \tilde{f}_i + \tilde{g}$. We can then apply LoCoDL to the latter problem. At Step 5, we simply have $y^t - \gamma\nabla\tilde{g}(y^t) = (1 - \frac{\gamma\mu}{2})y^t$. The rate in (5) applies with L and μ replaced by $\tilde{L}$ and $\tilde{\mu}$, respectively. Since $\kappa \leq \tilde{\kappa} := \frac{\tilde{L}}{\tilde{\mu}} \leq 2\kappa$, the asymptotic complexities derived above also apply to this setting. Thus, the presence of g in Problem (1) is not restrictive at all, as the only property of g that matters is that it has the same amount of strong convexity as the f_is.

Acknowledgement. The research reported in this publication was supported by funding from King Abdullah University of Science and Technology (KAUST): i) KAUST Baseline Research Scheme, ii) Center of Excellence for Generative AI, under award number 5940, iii) SDAIA-KAUST Center of Excellence in Artificial Intelligence and Data Science.

References

[1] A. Albasyoni, M. Safaryan, L. Condat, and P. Richtárik. Optimal gradient compression for distributed and federated learning. preprint arXiv:2010.03246, 2020.

[2] D. Basu, D. Data, C. Karakus, and S. N. Diggavi. Qsparse-Local-SGD: Distributed SGD With Quantization, Sparsification, and Local Computations. *IEEE Journal on Selected Areas in Information Theory*, 1(1):217–226, 2020.

[3] D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific, Belmont, MA, USA, 2015.

[4] A. Beznosikov, S. Horváth, P. Richtárik, and M. Safaryan. On biased compression for distributed learning. preprint arXiv:2002.12410, 2020.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proc. of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/%7Ecjlin/libsvm.

[7] L. Condat and P. Richtárik. MURANA: A generic framework for stochastic variance-reduced optimization. In *Proc. of the conference Mathematical and Scientific Machine Learning (MSML), PMLR 190*, 2022.

[8] L. Condat and P. Richtárik. RandProx: Primal-dual optimization algorithms with randomized proximal updates. In *Proc. of International Conference on Learning Representations (ICLR)*, 2023.

[9] L. Condat, I. Agarský, and P. Richtárik. Provably doubly accelerated federated learning: The first theoretically successful combination of local training and compressed communication. preprint arXiv:2210.13277, 2022.

[10] L. Condat, K. Li, and P. Richtárik. EF-BV: A unified theory of error feedback and variance reduction mechanisms for biased and unbiased compression in distributed optimization. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2022.

[11] L. Condat, I. Agarský, G. Malinovsky, and P. Richtárik. TAMUNA: Doubly accelerated federated learning with local training, compression, and partial participation. preprint arXiv:2302.09832 presented at the *Int. Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[12] I. Fatkhullin, I. Sokolov, E. Gorbunov, Z. Li, and P. Richtárik. EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback. preprint arXiv:2110.03294, 2021.

[13] E. Gorbunov, F. Hanzely, and P. Richtárik. A unified theory of SGD: Variance reduction, sampling, quantization and coordinate descent. In *Proc. of 23rd Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR 108*, 2020.

[14] E. Gorbunov, D. Kovalev, D. Makarenko, and P. Richtárik. Linearly converging error compensated SGD. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2020.

[15] E. Gorbunov, K. Burlachenko, Z. Li, and P. Richtárik. MARINA: Faster non-convex distributed learning with compression. In *Proc. of 38th Int. Conf. Machine Learning (ICML)*, pages 3788–3798, 2021.

[16] R. M. Gower, M. Schmidt, F. Bach, and P. Richtárik. Variance-reduced methods for machine learning. *Proc. of the IEEE*, 108(11):1968–1983, November 2020.

[17] M. Grudzień, G. Malinovsky, and P. Richtárik. Can 5th Generation Local Training Methods Support Client Sampling? Yes! In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2023.

[18] K. Gruntkowska, A. Tyurin, and P. Richtárik. EF21-P and friends: Improved theoretical communication complexity for distributed optimization with bidirectional compression. In *Proc. of 40th Int. Conf. Machine Learning (ICML)*, 2023.

[19] Farzin Haddadpour, Mohammad Mahdi Kamani, Aryan Mokhtari, and Mehrdad Mahdavi. Federated learning with compression: Unified analysis and sharp guarantees. In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR 130*, pages 2350–2358, 2021.

[20] F. Hanzely and P. Richtárik. One method to rule them all: Variance reduction for data, parameters and many new methods. preprint arXiv:1905.11266, 2019.

[21] Yutong He, Xinmeng Huang, and Kun Yuan. Unbiased compression saves communication in distributed optimization: When and how much? preprint arXiv:2305.16297, 2023.

[22] S. Horváth, C.-Y. Ho, L. Horváth, A. N. Sahu, M. Canini, and P. Richtárik. Natural compression for distributed deep learning. In *Proc. of the conference Mathematical and Scientific Machine Learning (MSML), PMLR 190*, 2022.

[23] S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *Optimization Methods and Software*, 2022.

[24] P. Kairouz et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2), 2021.

[25] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proc. of 37th Int. Conf. Machine Learning (ICML)*, pages 5132–5143, 2020.

[26] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: distributed machine learning for on-device intelligence. arXiv:1610.02527, 2016.

[27] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016. arXiv:1610.05492.

[28] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 3(37):50–60, 2020.

[29] Zhize Li, Dmitry Kovalev, Xun Qian, and Peter Richtárik. Acceleration for compressed gradient descent in distributed and federated optimization. In *Proc. of 37th Int. Conf. Machine Learning (ICML)*, volume PMLR 119, 2020.

[30] Xiaorui Liu, Yao Li, Jiliang Tang, and Ming Yan. A double residual compression algorithm for efficient distributed learning. In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR 108*, pages 133–143, 2020.

[31] G. Malinovsky, K. Yi, and P. Richtárik. Variance reduced ProxSkip: Algorithm, theory and application to federated learning. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2022.

[32] Grigory Malinovsky and Peter Richtárik. Federated random reshuffling with compression and variance reduction. preprint arXiv:arXiv:2205.03914, 2022.

[33] Artavazd Maranjyan, Mher Safaryan, and Peter Richtárik. Gradskip: Communication-accelerated local gradient methods with better computational complexity, 2023.

[34] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS), PMLR 54*, 2017.

[35] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik. Distributed learning with compressed gradient differences. arXiv:1901.09269, 2019.

[36] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik. ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally! In *Proc. of the 39th International Conference on Machine Learning (ICML)*, July 2022.

[37] Constantin Philippenko and Aymeric Dieuleveut. Artemis: tight convergence guarantees for bidirectional compression in federated learning. preprint arXiv:2006.14591, 2020.

[38] Constantin Philippenko and Aymeric Dieuleveut. Preserved central model for faster bidirectional compression in distributed settings. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2021.

[39] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani. FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization. In *Proc. of Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, pages 2021–2031, 2020.

[40] P. Richtárik, I. Sokolov, and I. Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. In *Proc. of 35th Conf. Neural Information Processing Systems (NeurIPS)*, 2021.

[41] A. Sadiev, D. Kovalev, and P. Richtárik. Communication acceleration of local gradient methods via an accelerated primal-dual algorithm with an inexact prox. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2022.

[42] A. Sadiev, G. Malinovsky, E. Gorbunov, I. Sokolov, A. Khaled, K. Burlachenko, and P. Richtárik. Federated optimization algorithms with random reshuffling and gradient compression. preprint arXiv:2206.07021, 2022.

[43] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal convergence rates for convex distributed optimization in networks. *Journal of Machine Learning Research*, 20:1–31, 2019.

[44] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[45] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. The MIT Press, 2011.

[46] A. Tyurin and P. Richtárik. 2Direction: Theoretically faster distributed training with bidirectional communication compression. In *Proc. of Conf. Neural Information Processing Systems (NeurIPS)*, 2023.

[47] Alexander Tyurin and Peter Richtárik. DASHA: Distributed nonconvex optimization with communication compression, optimal oracle complexity, and no client synchronization. In *Proc. of International Conference on Learning Representations (ICLR)*, 2023.

[48] J. Wang et al. A field guide to federated optimization. preprint arXiv:2107.06917, 2021.

Contents

Appendix A. State of the Art

We review the latest developments on communication-efficient algorithms for distributed learning, making use of LT, CC, or both. Before that, we note that we should distinguish uplink, or clients-to-server, from downlink, or server-to-clients, communication. Uplink is usually slower than downlink communication, since uploading different messages in parallel to the server is slower than broadcasting the same message to an arbitrary number of clients. This can be due to cache memory and aggregation speed constraints of the server, as well as asymmetry of the service provider's systems or protocols used on the internet or cell phone network. In this work, we focus on the **uplink communication complexity**, which is the bottleneck in practice. Indeed, the goal is to exploit parallelism to obtain better performance when n increases. Precisely, with LoCoDL, the uplink communication complexity decreases from $\mathcal{O}\left(d\sqrt{\kappa}\log\epsilon^{-1}\right)$ when n is small to $\mathcal{O}\left(\sqrt{d}\sqrt{\kappa}\log\epsilon^{-1}\right)$ when n is large, where the condition number κ is defined in Assumption 1, see Corollary 5. Many works have considered bidirectional compression, which consists in compressing the messages sent both ways [7, 14, 18, 30, 37, 38, 46] but to the best of our knowledge, this has no impact on the downlink complexity, which cannot be reduced further than $\mathcal{O}\left(d\sqrt{\kappa}\log\epsilon^{-1}\right)$, just because there is no parallelism to exploit in this direction. Thus, we focus our analysis on theoretical and algorithmic techniques to reduce the uplink communication complexity, which we call communication complexity in short, and we ignore downlink communication.

Communication Compression (CC) consists in applying some lossy scheme that compresses vectors into messages of small bit size, which are communicated. For instance, the well-known $\texttt{rand-}k$ compressor selects k coordinates of the vector uniformly at random, for some $k \in [d] := \{1,\ldots,d\}$. k can be as small as 1, in which case the compression factor is d, which can be huge. Some compressors, such as $\texttt{rand-}k$, are unbiased, whereas others are biased; we refer to Albasyoni et al. [1], Beznosikov et al. [4], Condat et al. [10], Horváth et al. [22] for several examples and a discussion of their properties. The introduction of DIANA by Mishchenko et al. [35] was a major milestone, as this algorithm converges linearly with the large class of unbiased compressors defined in Section 1.2 and also considered in LoCoDL. The communication complexity $\mathcal{O}\left(d\kappa\log\epsilon^{-1}\right)$ of the basic Gradient Descent (GD) algorithm is reduced with DIANA to $\mathcal{O}\left((\kappa+d)\log\epsilon^{-1}\right)$ when n is large, see Table 1. DIANA was later extended in several ways [7, 13, 23]. An accelerated version of DIANA called ADIANA based on Nesterov Accelerated GD has been proposed [29] and further analyzed in He et al. [21]; it has the state-of-the-art theoretical complexity.

Algorithms converging linearly with biased compressors have also been proposed, such as EF21 [10, 12, 40], but the acceleration potential is less understood than with unbiased compressors. Algorithms with CC such as MARINA [15] and DASHA [47] have been proposed for nonconvex optimization, but their analysis requires a different approach and there is a gap in the achievable performance: their complexity depends on $\frac{\omega\kappa}{\sqrt{n}}$ instead of $\frac{\omega\kappa}{n}$ with DIANA, where ω characterizes the compression error variance, see (2). Therefore, we focus on the convex setting and leave the nonconvex study for future work.

Local Training (LT) is a simple but remarkably efficient idea: the clients perform multiple Gradient Descent (GD) steps, instead of only one, between successive communication rounds. The intuition behind is that this leads to the communication of richer information, so that the number of communication rounds to reach a given accuracy is reduced. We refer to Mishchenko et al. [36] for a comprehensive review of LT-based algorithms, which include the popular FedAvg and Scaffold algorithms of McMahan et al. [34] and Karimireddy et al. [25], respectively. Mishchenko et al.

Table 1: (Uplink) communication complexity in number of reals to reach ϵ-accuracy for linearly-converging algorithms allowing for CC, with an optimal choice of unbiased compressors. We provide the leading asymptotic factor and ignore log factors such as $\log \epsilon^{-1}$. The state of the art is highlighted in green.

Algorithm	complexity in # reals	case $n = \mathcal{O}(d)$
DIANA	$(1+\frac{d}{n})\kappa + d$	$\frac{d}{n}\kappa + d$
EF21	$d\kappa$	$d\kappa$
5GCS-CC	$\left(\sqrt{d}+\frac{d}{\sqrt{n}}\right)\sqrt{\kappa}+d$	$\frac{d}{\sqrt{n}}\sqrt{\kappa}+d$
ADIANA	$\left(1+\frac{d}{\sqrt{n}}\right)\sqrt{\kappa}+d$	$\frac{d}{\sqrt{n}}\sqrt{\kappa}+d$
CompressedScaffnew	$\left(\sqrt{d}+\frac{d}{\sqrt{n}}\right)\sqrt{\kappa}+d$	$\frac{d}{\sqrt{n}}\sqrt{\kappa}+d$
FedCOMGATE	$d\kappa$	$d\kappa$
LoCoDL	$\left(\sqrt{d}+\frac{d}{\sqrt{n}}\right)\sqrt{\kappa}+d$	$\frac{d}{\sqrt{n}}\sqrt{\kappa}+d$

[36] made a breakthrough by proposing Scaffnew, the first LT-based variance-reduced algorithm that not only converges linearly to the exact solution in the strongly convex setting, but does so with accelerated communication complexity $\mathcal{O}(d\sqrt{\kappa}\log \epsilon^{-1})$. In Scaffnew, communication can occur randomly after every iteration, but occurs only with a small probability p. Thus, there are in average p^{-1} local steps between successive communication rounds. The optimal dependency on $\sqrt{\kappa}$ [43] is obtained with $p = 1/\sqrt{\kappa}$. LoCoDL has the same probabilistic LT mechanism as Scaffnew but does not revert to it when compression is disabled, because of the additional function g and tracking variables y and v. A different approach to LT was developed by Sadiev et al. [41] with the APDA-Inexact algorithm, and generalized to handle partial participation by Grudzień et al. [17] with the 5GCS algorithm: in both algorithms, the local GD steps form an inner loop in order to compute a proximity operator inexactly.

Combining LT and CC while retaining their benefits is very challenging. In our strongly convex and heterogeneous setting, the methods Qsparse-local-SGD [2] and FedPAQ [39] do not converge linearly. FedCOMGATE features LT + CC and converges linearly [19], but its complexity $\mathcal{O}(d\kappa \log \epsilon^{-1})$ does not show any acceleration. We can mention that random reshuffling, a technique that can be seen as a type of LT, has been combined with CC in Malinovsky and Richtárik [32], Sadiev et al. [42]. Recently, Condat et al. [9] managed to design a specific compression technique compatible with the LT mechanism of Scaffnew, leading to CompressedScaffnew, the first LT + CC algorithm exhibiting a doubly-accelerated complexity, namely $\mathcal{O}\big(\big(\sqrt{d}\sqrt{\kappa}+\frac{d\sqrt{\kappa}}{\sqrt{n}}+d\big)\log \epsilon^{-1}\big)$, as reported in Table 1. However, CompressedScaffnew uses a specific linear compression scheme that requires shared randomness; that is, all clients have to agree on a random permutation of the columns of the global compression pattern. No other compressor can be used, which notably rules out any type of quantization.

Table 2: Communication complexity in number of communication rounds to reach ϵ-accuracy for linearly-converging algorithms allowing for CC with independent compressors in $\mathbb{U}(\omega)$ for any $\omega \geq 0$. Since the compressors are independent, $\omega_{\mathrm{av}} = \frac{\omega}{n}$. We provide the leading asymptotic factor and ignore log factors such as $\log \epsilon^{-1}$. The state of the art is highlighted in green.

Algorithm	Com. complexity in # rounds	case $\omega = \mathcal{O}(n)$	case $\omega = \Theta(n)$
DIANA	$(1 + \frac{\omega}{n})\kappa + \omega$	$\kappa + \omega$	$\kappa + \omega$
EF21	$(1 + \omega)\kappa$	$(1 + \omega)\kappa$	$(1 + \omega)\kappa$
5GCS-CC	$\left(1+\sqrt{\omega}+\frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$
ADIANA[1]	$\left(1+\frac{\omega^{3/4}}{n^{1/4}}+\frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$\left(1+\frac{\omega^{3/4}}{n^{1/4}}\right)\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$
ADIANA[2]	$\left(1 + \frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$\left(1 + \frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$
lower bound[2]	$\left(1 + \frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$\left(1 + \frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$
LoCoDL	$\left(1+\sqrt{\omega}+\frac{\omega}{\sqrt{n}}\right)\sqrt{\kappa} + \omega(1+\frac{\omega}{n})$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$	$(1+\sqrt{\omega})\sqrt{\kappa} + \omega$

[1] This is the complexity derived in the original paper Li et al. [29].

[2] This is the complexity derived by a refined analysis in the preprint He et al. [21], where a matching lower bound is also derived.

Appendix B. Experiments

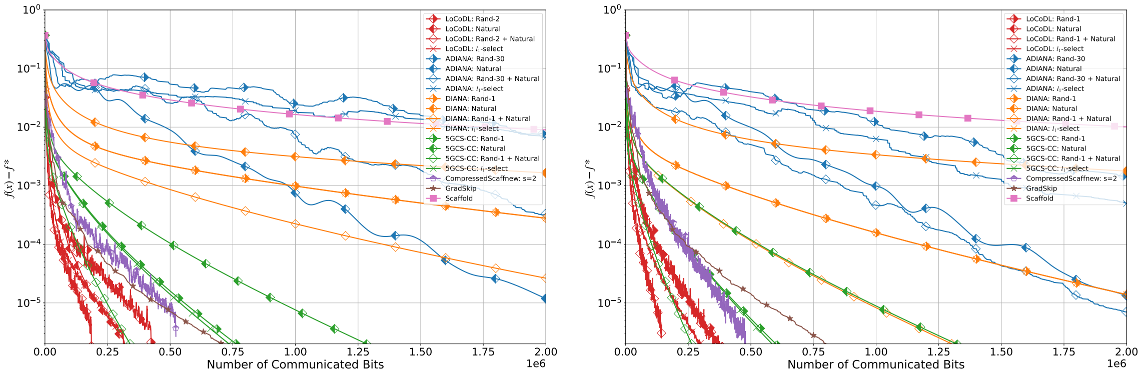
We evaluate the performance of our proposed method LoCoDL and compare it with several other methods that also allow for CC and converge linearly to $x^\star$. We also include GradSkip [33] and Scaffold [34] in our comparisons. We focus on a regularized logistic regression problem, which has the form (1) with

$$f_i(x) = \frac{1}{m}\sum_{s=1}^{m} \log\Big(1+\exp\left(-b_{i,s}a_{i,s}^\top x\right)\Big) + \frac{\mu}{2}\|x\|^2 \tag{11}$$

and $g = \frac{\mu}{2}\|x\|^2$, where n is the number of clients, m is the number of data points per client, $a_{i,s} \in \mathbb{R}^d$ and $b_{i,s} \in \{-1, +1\}$ are the data samples, and μ is the regularization parameter, set so that $\kappa = 10^4$. For all algorithms other than LoCoDL, for which there is no function g, the functions f_i in (11) have a twice higher μ, so that the problem remains the same.

We considered several datasets from the LibSVM library [6] (3-clause BSD license). We show the results with the 'a5a' dataset in Figure 1 and with other datasets in the Appendix. We prepared each dataset by first shuffling it, then distributing it equally among the n clients (since m in (11) is an integer, the remaining datapoints were discarded). We used four different compression operators in the class $\mathbb{U}(\omega)$, for some $\omega \geq 0$:

- `rand`-k for some $k \in [d]$, which communicates $32k+k\lceil\log_2(d)\rceil$ bits. Indeed, the k randomly chosen values are sent in the standard 32-bits IEEE floating-point format, and their locations are encoded with $k\lceil\log_2(d)\rceil$ additional bits. We have $\omega = \frac{d}{k} - 1$.
- Natural Compression [22], a form of quantization in which floats are encoded into 9 bits instead of 32 bits. We have $\omega = \frac{1}{8}$.



(a) $n = 87$

(b) $n = 288$

Figure 1: Comparison of several algorithms with several compressors on logistic regression with the 'a5a' dataset from the LibSVM, which has $d = 122$ and 6,414 data points. We chose different values of n to illustrate the two regimes $n < d$ and $n > d$, as discussed at the end of Section 3.

- A combination of rand-k and Natural Compression, in which the k chosen values are encoded into 9 bits, which yields a total of $9k + k\lceil \log_2(d) \rceil$ bits. We have $\omega = \frac{9d}{8k} - 1$.
- The l_1-selection compressor, defined as $C(x) = \text{sign}(x_j)\|x\|_1 e_j$, where j is chosen randomly in $[d]$, with the probability of choosing $j' \in [d]$ equal to $|x_{j'}|/\|x\|_1$, and e_j is the j-th standard unit basis vector in $\mathbb{R}^d$. $\text{sign}(x_j)\|x\|_1$ is sent as a 32-bits float and the location of j is indicated with $\lceil \log_2(d) \rceil$, so that this compressor communicates $32 + \lceil \log_2(d) \rceil$ bits. Like with rand-1, we have $\omega = d - 1$.

The compressors at different clients are independent, so that $\omega_{\text{av}} = \frac{\omega}{n}$ in (3).

We can see that LoCoDL, when combined with rand-k and Natural Compression, converges faster than all other algorithms, with respect to the total number of communicated bits per client. We chose two different numbers n of clients, one with $n < d$ and another one with $n > 2d$, since the compressor of CompressedScaffnew is different in the two cases $n < 2d$ and $n > 2d$ [9]. LoCoDL outperforms CompressedScaffnew in both cases. As expected, all methods exhibit faster convergence with larger n. Remarkably, ADIANA, which has the best theoretical complexity for large n, improves upon DIANA but is not competitive with the LT-based methods CompressedScaffnew, 5GCS-CC, and LoCoDL. This illustrates the power of doubly-accelerated methods based on a successful combination of LT and CC. In this class, our new proposed LoCoDL algorithm shines. For all algorithms, we used the theoretical parameter values given in their available convergence results (Corollary 5 for LoCoDL). We tried to tune the parameter values, such as k in rand-k and the (average) number of local steps per round, but this only gave minor improvements. For instance, ADIANA in Figure 1 was a bit faster with the best value of $k = 20$ than with $k = 30$. Increasing the learning rate γ led to inconsistent results, with sometimes divergence.

Appendix C. Conclusion

We have proposed LoCoDL, which combines a probabilistic Local Training mechanism similar to the one of Scaffnew and Communication Compression with a large class of unbiased compressors. This successful combination makes LoCoDL highly communication-efficient, with a doubly accelerated complexity with respect to the model dimension d and the condition number of the functions. In practice, LoCoDL outperforms other algorithms, including ADIANA, which has an even better complexity in theory obtained from Nesterov acceleration and not Local Training. This again shows the relevance of the popular mechanism of Local Training, which has been widely adopted in Federated Learning. A venue for future work is to implement bidirectional compression [30, 38]. We will also investigate extensions of our method with calls to stochastic gradient estimates, with or without variance reduction, as well as partial participation. These two features have been proposed for Scaffnew in Malinovsky et al. [31] and Condat et al. [11], but they are challenging to combine with generic compression.

Appendix D. Proof of Theorem 4

We define the Euclidean space $\mathcal{X} := \mathbb{R}^d$ and the product space $\boldsymbol{\mathcal{X}} := \mathcal{X}^{n+1}$ endowed with the weighted inner product

$$\langle \mathbf{x}, \mathbf{x}' \rangle_{\boldsymbol{\mathcal{X}}} := \sum_{i=1}^{n} \langle x_i, x_i' \rangle + n \langle y, y' \rangle, \quad \forall \mathbf{x} = (x_1, \ldots, x_n, y), \mathbf{x}' = (x_1', \ldots, x_n', y'). \tag{12}$$

We define the copy operator $\mathbf{1} : x \in \mathcal{X} \mapsto (x, \ldots, x, x) \in \boldsymbol{\mathcal{X}}$ and the linear operator

$$S : \mathbf{x} \in \boldsymbol{\mathcal{X}} \mapsto \mathbf{1}\bar{x}, \ \text{ with } \bar{x} = \frac{1}{2n}\left(\sum_{i=1}^{n} x_i + ny\right). \tag{13}$$

S is the orthogonal projector in $\boldsymbol{\mathcal{X}}$ onto the consensus line $\{\mathbf{x} \in \boldsymbol{\mathcal{X}} \ : \ x_1 = \cdots = x_n = y\}$. We also define the linear operator

$$W := \mathrm{Id} - S : \mathbf{x} = (x_1, \ldots, x_n, y) \in \boldsymbol{\mathcal{X}} \mapsto (x_1 - \bar{x}, \ldots, x_n - \bar{x}, y - \bar{x}), \text{ with } \bar{x} = \frac{1}{2n}\left(\sum_{i=1}^{n} x_i + ny\right), \tag{14}$$

where Id denotes the identity. W is the orthogonal projector in $\boldsymbol{\mathcal{X}}$ onto the hyperplane $\{\mathbf{x} \in \boldsymbol{\mathcal{X}} \ : \ x_1 + \cdots + x_n + ny = 0\}$, which is orthogonal to the consensus line. As such, it is self-adjoint, positive semidefinite, its eigenvalues are $(1, \ldots, 1, 0)$, its kernel is the consensus line, and its spectral norm is 1. Also, $W^2 = W$. Note that we can write W in terms of the differences $d_i = x_i - y$ and $\bar{d} = \frac{1}{2n}\sum_{i=1}^{n} d_i$:

$$W : \mathbf{x} = (x_1, \ldots, x_n, y) \mapsto \big(d_1 - \bar{d}, \ldots, d_n - \bar{d}, -\bar{d}\big). \tag{15}$$

Since for every $\mathbf{x} = (x_1, \ldots, x_n, y)$, $W\mathbf{x} = \mathbf{0} := (0, \ldots, 0, 0)$ if and only if $x_1 = \cdots = x_n = y$, we can reformulate the problem (1) as

$$\min_{\mathbf{x} = (x_1, \ldots, x_n, y) \in \boldsymbol{\mathcal{X}}} \mathbf{f}(\mathbf{x}) \quad \text{s.t.} \quad W\mathbf{x} = \mathbf{0}, \tag{16}$$

where $\mathbf{f}(\mathbf{x}) := \sum_{i=1}^n f_i(x_i) + ng(y)$. Note that in $\boldsymbol{\mathcal{X}}$, $\mathbf{f}$ is L-smooth and μ-strongly convex, and $\nabla\mathbf{f}(\mathbf{x}) = \big(\nabla f_1(x_1), \dots \nabla f_n(x_n), \nabla g(y)\big)$.

Let $t \geq 0$. We also introduce vector notations for the variables of the algorithm: $\mathbf{x}^t := (x_1^t, \dots, x_n^t, y^t)$, $\hat{\mathbf{x}}^t := (\hat{x}_1^t, \dots, \hat{x}_n^t, \hat{y}^t)$, $\mathbf{u}^t := (u_1^t, \dots, u_n^t, v^t)$, $\mathbf{u}^\star := (u_1^\star, \dots, u_n^\star, v^\star)$, $\mathbf{w}^t := \mathbf{x}^t - \gamma\nabla\mathbf{f}(\mathbf{x}^t)$, $\mathbf{w}^\star := \mathbf{x}^\star - \gamma\nabla\mathbf{f}(\mathbf{x}^\star)$, where $\mathbf{x}^\star := \mathbf{1}x^\star$ is the unique solution to (16). We also define $\bar{x}^t := \frac{1}{2n}\big(\sum_{i=1}^n \hat{x}_i^t + n\hat{y}^t\big)$ and $\lambda := \frac{p\chi}{\gamma(1+2\omega)}$.

Then we can write the iteration of LoCoDL as

$$\left|\begin{array}{l}
\hat{\mathbf{x}}^t := \mathbf{x}^t - \gamma\nabla\mathbf{f}(\mathbf{x}^t) + \gamma\mathbf{u}^t = \mathbf{w}^t + \gamma\mathbf{u}^t \\
\text{flip a coin } \theta^t \in \{0,1\} \text{ with } \mathrm{Prob}(\theta^t = 1) = p \\
\textbf{if } \theta^t = 1 \\
\quad \mathbf{d}^t := \big(\mathcal{C}_1^t(\hat{x}_1^t - \hat{y}^t), \dots, \mathcal{C}_n^t(\hat{x}_n^t - \hat{y}^t), 0\big) \\
\quad \bar{d}^t := \frac{1}{2n}\sum_{j=1}^n d_j^t \\
\quad \mathbf{x}^{t+1} := (1-\rho)\hat{\mathbf{x}}^t + \rho\mathbf{1}(\hat{y}^t + \bar{d}^t) \\
\quad \mathbf{u}^{t+1} := \mathbf{u}^t + \lambda\big(\mathbf{1}\bar{d}^t - \mathbf{d}^t\big) = \mathbf{u}^t - \lambda W\mathbf{d}^t \\
\textbf{else} \\
\quad \mathbf{x}^{t+1} := \hat{\mathbf{x}}^t \\
\quad \mathbf{u}^{t+1} := \mathbf{u}^t \\
\textbf{end if}
\end{array}\right. \tag{17}$$

We denote by $\mathcal{F}^t$ the σ-algebra generated by the collection of $\mathcal{X}$-valued random variables $\mathbf{x}^0, \mathbf{u}^0, \dots, \mathbf{x}^t, \mathbf{u}^t$.

Since we suppose that $S\mathbf{u}^0 = \mathbf{0}$ and we have $SW\mathbf{d}^{t'} = \mathbf{0}$ in the update of $\mathbf{u}$, we have $S\mathbf{u}^{t'} = \mathbf{0}$ for every $t' \geq 0$.

If $\theta^t = 1$, we have

$$\begin{aligned}
\left\|\mathbf{u}^{t+1} - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 &= \left\|\mathbf{u}^t - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 + \lambda^2\left\|W\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 - 2\lambda\langle\mathbf{u}^t - \mathbf{u}^\star, W\mathbf{d}^t\rangle_{\boldsymbol{\mathcal{X}}} \\
&= \left\|\mathbf{u}^t - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 + \lambda^2\left\|\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 - \lambda^2\left\|S\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 - 2\lambda\langle\mathbf{u}^t - \mathbf{u}^\star, \mathbf{d}^t\rangle_{\boldsymbol{\mathcal{X}}},
\end{aligned}$$

because $S\mathbf{u}^t = S\mathbf{u}^\star = \mathbf{0}$, so that $\langle\mathbf{u}^t - \mathbf{u}^\star, S\mathbf{d}^t\rangle_{\boldsymbol{\mathcal{X}}} = 0$.

The variance inequality (2) satisfied by the compressors $\mathcal{C}_i^t$ is equivalent to $\mathbb{E}\Big[\left\|\mathcal{C}_i^t(x)\right\|^2\Big] \leq (1+\omega)\left\|x\right\|^2$, so that

$$\mathbb{E}\Big[\left\|\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 \mid \mathcal{F}^t, \theta^t = 1\Big] \leq (1+\omega)\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|_{\boldsymbol{\mathcal{X}}}^2.$$

Also,

$$\mathbb{E}\big[\mathbf{d}^t \mid \mathcal{F}^t, \theta^t = 1\big] = \hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t.$$

Thus,

$$\begin{aligned}
\mathbb{E}\Big[\left\|\mathbf{u}^{t+1} - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 \mid \mathcal{F}^t\Big] &= (1-p)\left\|\mathbf{u}^t - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 + p\mathbb{E}\Big[\left\|\mathbf{u}^{t+1} - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 \mid \mathcal{F}^t, \theta^t = 1\Big] \\
&\leq \left\|\mathbf{u}^t - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 + p\lambda^2(1+\omega)\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 - p\lambda^2\mathbb{E}\Big[\left\|S\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 \mid \mathcal{F}^t, \theta^t = 1\Big] \\
&\quad - 2p\lambda\langle\mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\rangle_{\boldsymbol{\mathcal{X}}} \\
&= \left\|\mathbf{u}^t - \mathbf{u}^\star\right\|_{\boldsymbol{\mathcal{X}}}^2 + p\lambda^2(1+\omega)\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 - p\lambda^2\mathbb{E}\Big[\left\|S\mathbf{d}^t\right\|_{\boldsymbol{\mathcal{X}}}^2 \mid \mathcal{F}^t, \theta^t = 1\Big] \\
&\quad - 2p\lambda\langle\mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t\rangle_{\boldsymbol{\mathcal{X}}}.
\end{aligned}$$

Moreover, $\mathbb{E}\Big[\big\|S\mathbf{d}^t\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta^t = 1\Big] \geq \big\|\mathbb{E}\big[S\mathbf{d}^t \mid \mathcal{F}^t, \theta^t = 1\big]\big\|^2_{\boldsymbol{\mathcal{X}}} = \big\|S\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\big\|^2_{\boldsymbol{\mathcal{X}}}$ and $\big\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\big\|^2_{\boldsymbol{\mathcal{X}}} = \big\|S\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\big\|^2_{\boldsymbol{\mathcal{X}}} + \big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}$, so that

$$\begin{aligned}
\mathbb{E}\Big[\big\|\mathbf{u}^{t+1} - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big] &\leq \big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + p\lambda^2(1+\omega)\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|^2_{\boldsymbol{\mathcal{X}}} - p\lambda^2\left\|S\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|^2 \\
&\quad - 2p\lambda\langle \mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t\rangle_{\boldsymbol{\mathcal{X}}} \\
&= \big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + p\lambda^2\omega\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|^2_{\boldsymbol{\mathcal{X}}} + p\lambda^2\left\|W\hat{\mathbf{x}}^t\right\|^2 - 2p\lambda\langle \mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t\rangle_{\boldsymbol{\mathcal{X}}}.
\end{aligned}$$

From the Peter–Paul inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ for any a and b, we have

$$\begin{aligned}
\left\|\hat{\mathbf{x}}^t - \mathbf{1}\hat{y}^t\right\|^2_{\boldsymbol{\mathcal{X}}} &= \sum_{i=1}^n \left\|\hat{x}_i^t - \hat{y}^t\right\|^2 = \sum_{i=1}^n \left\|(\hat{x}_i^t - \bar{x}^t) - (\hat{y}^t - \bar{x}^t)\right\|^2 \\
&\leq \sum_{i=1}^n \left(2\left\|\hat{x}_i^t - \bar{x}^t)\right\|^2 + 2\left\|\hat{y}^t - \bar{x}^t\right\|^2\right) \\
&= 2\left(\sum_{i=1}^n \left\|\hat{x}_i^t - \bar{x}^t)\right\|^2 + n\left\|\hat{y}^t - \bar{x}^t\right\|^2\right) \\
&= 2\left\|\hat{\mathbf{x}}^t - \mathbf{1}\bar{x}^t\right\|^2_{\boldsymbol{\mathcal{X}}} = 2\left\|W\hat{\mathbf{x}}^t\right\|^2_{\boldsymbol{\mathcal{X}}}. \qquad (18)
\end{aligned}$$

Hence,

$$\mathbb{E}\Big[\big\|\mathbf{u}^{t+1} - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big] \leq \big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + p\lambda^2(1+2\omega)\left\|W\hat{\mathbf{x}}^t\right\|^2_{\boldsymbol{\mathcal{X}}} - 2p\lambda\langle \mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t\rangle_{\boldsymbol{\mathcal{X}}}.$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta = 1\Big] &= (1-\rho)^2\left\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\right\|^2_{\boldsymbol{\mathcal{X}}} + \rho^2\mathbb{E}\Big[\big\|\mathbf{1}(\hat{y}^t + \bar{d}^t) - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta = 1\Big] \\
&\quad + 2\rho(1-\rho)\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \mathbf{1}\big(\hat{y}^t + \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big) - \mathbf{x}^\star\right\rangle_{\boldsymbol{\mathcal{X}}}.
\end{aligned}$$

We have $\mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big] = \frac{1}{2n}\sum_{i=1}^n \hat{x}_i^t - \frac{1}{2}\hat{y}^t = \bar{x}^t - \hat{y}^t$, so that

$$\mathbf{1}\big(\hat{y}^t + \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big) = \mathbf{1}\bar{x}^t = S\hat{\mathbf{x}}^t.$$

In addition,

$$\left\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, S\hat{\mathbf{x}}^t - \mathbf{x}^\star\right\rangle_{\boldsymbol{\mathcal{X}}} = \left\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, S(\hat{\mathbf{x}}^t - \mathbf{x}^\star)\right\rangle_{\boldsymbol{\mathcal{X}}} = \left\|S(\hat{\mathbf{x}}^t - \mathbf{x}^\star)\right\|^2_{\boldsymbol{\mathcal{X}}}.$$

Moreover,

$$\begin{aligned}
\mathbb{E}\Big[\big\|\mathbf{1}(\hat{y}^t + \bar{d}^t) - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta = 1\Big] &= \big\|\mathbf{1}\big(\hat{y}^t + \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big) - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \\
&\quad + \mathbb{E}\Big[\big\|\mathbf{1}\big(\bar{d}^t - \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big)\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta = 1\Big] \\
&= \big\|S\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \\
&\quad + 2n\mathbb{E}\Big[\big\|\bar{d}^t - \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big\|^2 \mid \mathcal{F}^t, \theta = 1\Big]
\end{aligned}$$

and, using (3),

$$\begin{aligned}\mathbb{E}\Big[\big\|\bar{d}^t - \mathbb{E}\big[\bar{d}^t \mid \mathcal{F}^t, \theta = 1\big]\big\|^2 \mid \mathcal{F}^t, \theta = 1\Big] &\le \frac{\omega_{\mathrm{av}}}{4n}\sum_{i=1}^{n}\big\|\hat{x}_i^t - \hat{y}^t\big\|^2\\ &\le \frac{\omega_{\mathrm{av}}}{2n}\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}},\end{aligned}$$

where the second inequality follows from (18). Hence,

$$\begin{aligned}\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta = 1\Big] &\le (1-\rho)^2\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \rho^2\big\|S\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \rho^2\omega_{\mathrm{av}}\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &\quad + 2\rho(1-\rho)\big\|S(\hat{\mathbf{x}}^t - \mathbf{x}^\star)\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &= (1-\rho)^2\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \rho^2\omega_{\mathrm{av}}\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &\quad + (2\rho-\rho^2)\big\|S(\hat{\mathbf{x}}^t - \mathbf{x}^\star)\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &= (1-\rho)^2\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \rho^2\omega_{\mathrm{av}}\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &\quad + (2\rho-\rho^2)\Big(\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\Big)\\ &= \big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \big(2\rho - \rho^2 - \rho^2\omega_{\mathrm{av}}\big)\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big] &= (1-p)\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + p\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t, \theta^t = 1\Big]\\ &\le \big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - p\big(2\rho - \rho^2(1+\omega_{\mathrm{av}})\big)\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}.\end{aligned}$$

Furthermore,

$$\begin{aligned}\big\|\hat{\mathbf{x}}^t - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} &= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \gamma^2\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + 2\gamma\langle \mathbf{w}^t - \mathbf{w}^\star, \mathbf{u}^t - \mathbf{u}^\star\rangle_{\boldsymbol{\mathcal{X}}}\\ &= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \gamma^2\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + 2\gamma\langle \hat{\mathbf{x}}^t - \mathbf{x}^\star, \mathbf{u}^t - \mathbf{u}^\star\rangle_{\boldsymbol{\mathcal{X}}}\\ &= \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \gamma^2\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + 2\gamma\langle \hat{\mathbf{x}}^t, \mathbf{u}^t - \mathbf{u}^\star\rangle_{\boldsymbol{\mathcal{X}}},\end{aligned}$$

which yields

$$\begin{aligned}\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big] &\le \big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \gamma^2\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + 2\gamma\langle \hat{\mathbf{x}}^t, \mathbf{u}^t - \mathbf{u}^\star\rangle_{\boldsymbol{\mathcal{X}}}\\ &\quad - p\big(2\rho - \rho^2(1+\omega_{\mathrm{av}})\big)\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}.\end{aligned}$$

Hence, with $\lambda = \frac{p\chi}{\gamma(1+2\omega)}$,

$$\begin{aligned}&\frac{1}{\gamma}\mathbb{E}\Big[\big\|\mathbf{x}^{t+1} - \mathbf{x}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big] + \frac{\gamma(1+2\omega)}{p^2\chi}\mathbb{E}\Big[\big\|\mathbf{u}^{t+1} - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} \mid \mathcal{F}^t\Big]\\ &\quad\le \frac{1}{\gamma}\big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} - \gamma\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + 2\langle \hat{\mathbf{x}}^t, \mathbf{u}^t - \mathbf{u}^\star\rangle_{\boldsymbol{\mathcal{X}}} - \frac{p}{\gamma}\big(2\rho - \rho^2(1+\omega_{\mathrm{av}})\big)\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &\qquad + \frac{\gamma(1+2\omega)}{p^2\chi}\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \frac{p\chi}{\gamma}\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}} - 2\langle \mathbf{u}^t - \mathbf{u}^\star, \hat{\mathbf{x}}^t\rangle_{\boldsymbol{\mathcal{X}}}\\ &\quad= \frac{1}{\gamma}\big\|\mathbf{w}^t - \mathbf{w}^\star\big\|^2_{\boldsymbol{\mathcal{X}}} + \frac{\gamma(1+2\omega)}{p^2\chi}\left(1 - \frac{p^2\chi}{1+2\omega}\right)\big\|\mathbf{u}^t - \mathbf{u}^\star\big\|^2_{\boldsymbol{\mathcal{X}}}\\ &\qquad - \frac{p}{\gamma}\big(2\rho - \rho^2(1+\omega_{\mathrm{av}}) - \chi\big)\big\|W\hat{\mathbf{x}}^t\big\|^2_{\boldsymbol{\mathcal{X}}}.\end{aligned}$$

Therefore, assuming that $2\rho - \rho^2(1+\omega_{\mathrm{av}}) - \chi \geq 0$,

$$\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}^t\big] \leq \frac{1}{\gamma} \left\|\mathbf{w}^t - \mathbf{w}^\star\right\|^2_{\boldsymbol{\mathcal{X}}} + \left(1 - \frac{p^2\chi}{1+2\omega}\right) \frac{\gamma(1+2\omega)}{p^2\chi} \left\|\mathbf{u}^t - \mathbf{u}^\star\right\|^2_{\boldsymbol{\mathcal{X}}}.$$

According to Condat and Richtárik [8, Lemma 1],

$$\begin{aligned}\left\|\mathbf{w}^t - \mathbf{w}^\star\right\|^2_{\boldsymbol{\mathcal{X}}} &= \left\|(\mathrm{Id} - \gamma\nabla\mathbf{f})\mathbf{x}^t - (\mathrm{Id} - \gamma\nabla\mathbf{f})\mathbf{x}^\star\right\|^2_{\boldsymbol{\mathcal{X}}} \\ &\leq \max(1-\gamma\mu, \gamma L - 1)^2 \left\|\mathbf{x}^t - \mathbf{x}^\star\right\|^2_{\boldsymbol{\mathcal{X}}}.\end{aligned}$$

Hence,

$$\mathbb{E}\big[\Psi^{t+1} \mid \mathcal{F}^t\big] \leq \max\left((1-\gamma\mu)^2, (1-\gamma L)^2, 1 - \frac{p^2\chi}{1+2\omega}\right) \Psi^t. \tag{19}$$

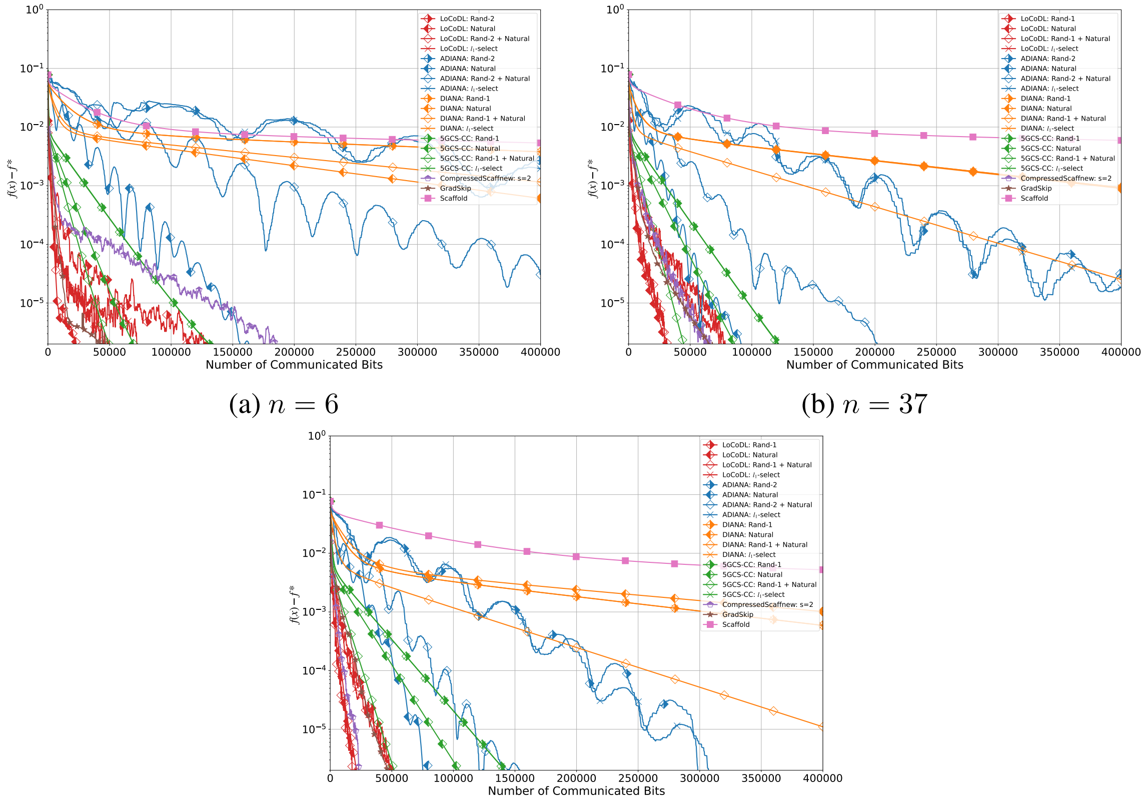
Using the tower rule, we can unroll the recursion in (19) to obtain the unconditional expectation of Ψ^{t+1}.

Using classical results on supermartingale convergence [3, Proposition A.4.5], it follows from (19) that $\Psi^t \to 0$ almost surely. Almost sure convergence of $\mathbf{x}^t$ and $\mathbf{u}^t$ follows.

Appendix E. Additional Experiments

The results for the experiments in Section B with the 'diabetes' dataset from the LibSVM library [6] are shown in Figure 2. The results with the 'w1a' and 'australian' datasets, for the same logistic regression problem with $\kappa = 10^4$, are shown in Figures 3 and 4.

Consistent with our previous findings, LoCoDL outperforms the other algorithms in terms of communication efficiency.

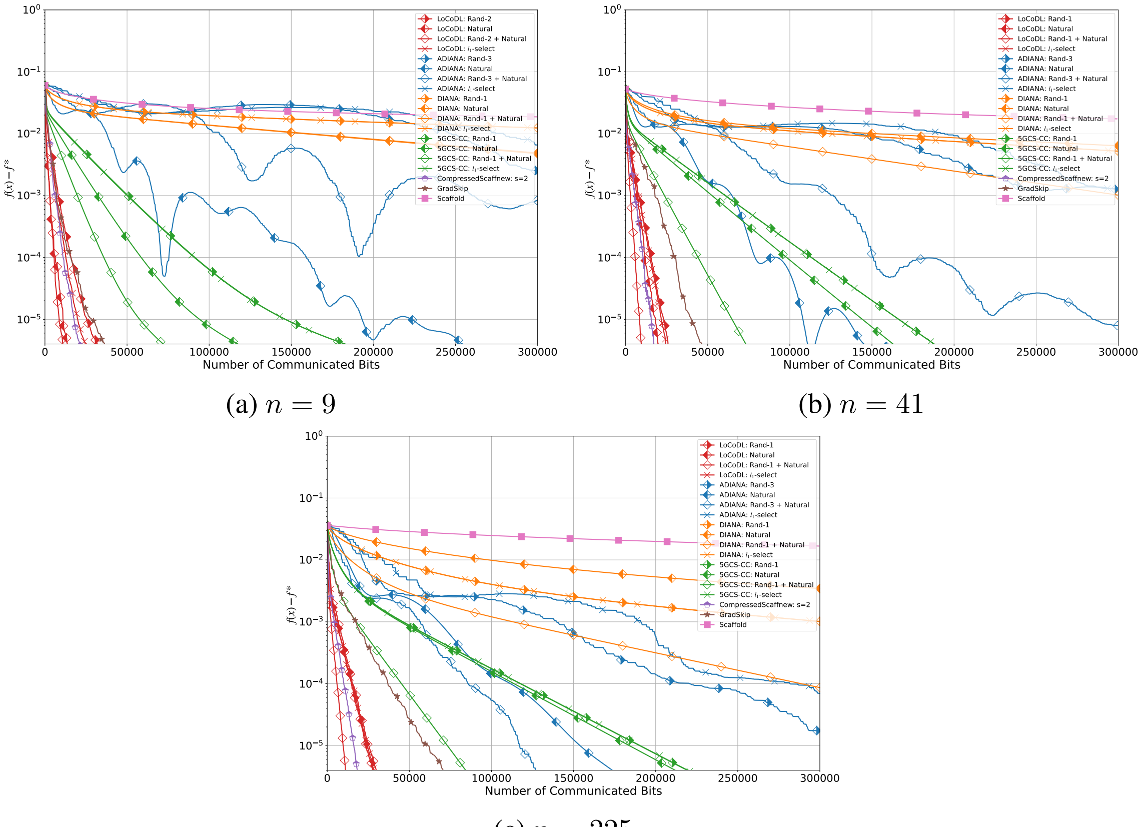


(a) $n = 6$

(b) $n = 37$

(c) $n = 73$

Figure 2: Comparison of several algorithms with several compressors on logistic regression with the ‘diabetes’ dataset from the LibSVM, which has $d = 8$ and 768 data points. We chose different values of n to illustrate the three regimes $n < d$, $n > d$, $n > d^2$, as discussed at the end of Section 3.

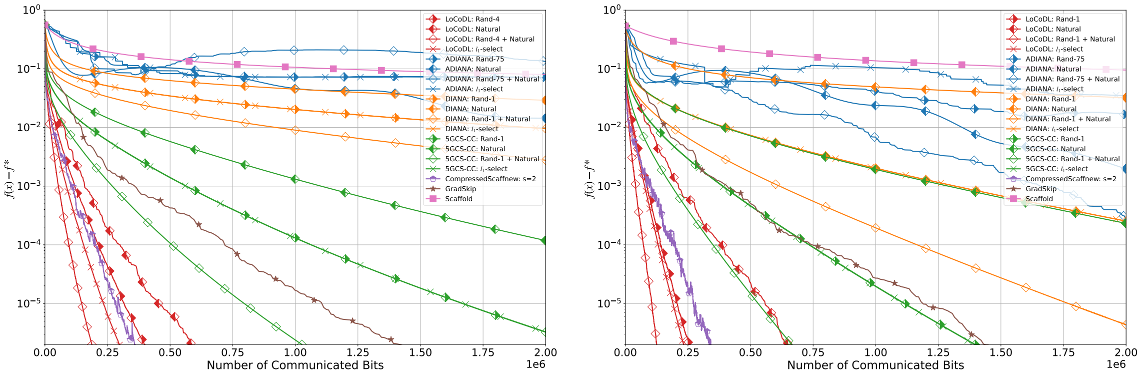


(a) $n = 9$

(b) $n = 41$

(c) $n = 225$

Figure 3: Comparison of several algorithms with various compressors on logistic regression with the 'australian' dataset from the LibSVM, which has $d = 14$ and 690 data points. We chose different values of n to illustrate the three regimes: $n < d$, $n > d$, $n > d^2$, as discussed at the end of Section 3.



(a) $n = 87$

(b) $n = 619$

Figure 4: Comparison of several algorithms with various compressors on logistic regression with the ‘w1a’ dataset from the LibSVM, which has $d = 300$ and 2,477 data points. We chose different values of n to illustrate the two regimes, $n < d$ and $n > d$, as discussed at the end of Section 3.