CoLa: Decentralized Linear Learning

Lie He[*]
EPFL
lie.he@epfl.ch

An Bian[*]
ETH Zurich
ybian@inf.ethz.ch

Martin Jaggi
EPFL
martin.jaggi@epfl.ch

Abstract

Decentralized machine learning is a promising emerging paradigm in view of global challenges of data ownership and privacy. We consider learning of linear classification and regression models, in the setting where the training data is decentralized over many user devices, and the learning algorithm must run on-device, on an arbitrary communication network, without a central coordinator. We propose CoLa, a new decentralized training algorithm with strong theoretical guarantees and superior practical performance. Our framework overcomes many limitations of existing methods, and achieves communication efficiency, scalability, elasticity as well as resilience to changes in data and allows for unreliable and heterogeneous participating devices.

1 Introduction

With the immense growth of data, decentralized machine learning has become not only attractive but a necessity. Personal data from, for example, smart phones, wearables and many other mobile devices is sensitive and exposed to a great risk of data breaches and abuse when collected by a centralized authority or enterprise. Nevertheless, many users have gotten accustomed to giving up control over their data in return for useful machine learning predictions (e.g. recommendations), which benefits from joint training on the data of all users combined in a centralized fashion.

In contrast, decentralized learning aims at learning this same global machine learning model, without any central server. Instead, we only rely on distributed computations of the devices themselves, with each user's data never leaving its device of origin. While increasing research progress has been made towards this goal, major challenges in terms of the privacy aspects as well as algorithmic efficiency, robustness and scalability remain to be addressed. Motivated by aforementioned challenges, we make progress in this work addressing the important problem of training generalized linear models in a fully decentralized environment.

Existing research on decentralized optimization, $\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x})$, can be categorized into two main directions. The seminal line of work started by Bertsekas and Tsitsiklis in the 1980s, cf. [Tsitsiklis et al., 1986], tackles this problem by splitting the parameter vector $\mathbf{x}$ by coordinates/components among the devices. A second more recent line of work including e.g. [Nedic and Ozdaglar, 2009, Duchi et al., 2012, Shi et al., 2015, Mokhtari and Ribeiro, 2016, Nedic et al., 2017] addresses sum-structured $F(\mathbf{x}) = \sum_k F_k(\mathbf{x})$ where F_k is the local cost function of node k. This structure is closely related to empirical risk minimization in a learning setting. See e.g. [Cevher et al., 2014] for an overview of both directions. While the first line of work typically only provides convergence guarantees for smooth objectives F, the second approach often suffers from a "lack of consensus", that is, the minimizers of $\{F_k\}_k$ are typically different since the data is not distributed i.i.d. between devices in general.

[*]These two authors contributed equally

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Contributions. In this paper, our main contribution is to propose CoLA, a new decentralized framework for training generalized linear models with convergence guarantees. Our scheme resolves both described issues in existing approaches, using techniques from primal-dual optimization, and can be seen as a generalization of CoCoA [Smith et al., 2018] to the decentralized setting. More specifically, the proposed algorithm offers

- *Convergence Guarantees:* Linear and sublinear convergence rates are guaranteed for strongly convex and general convex objectives respectively. Our results are free of the restrictive assumptions made by stochastic methods [Zhang et al., 2015, Wang et al., 2017], which requires i.i.d. data distribution over all devices.
- *Communication Efficiency and Usability:* Employing a data-local subproblem between each communication round, CoLA not only achieves communication efficiency but also allows the re-use of existing efficient single-machine solvers for on-device learning. We provide practical decentralized primal-dual certificates to diagnose the learning progress.
- *Elasticity and Fault Tolerance:* Unlike sum-structured approaches such as SGD, CoLA is provably resilient to changes in the data, in the network topology, and participating devices disappearing, straggling or re-appearing in the network.

Our implementation is publicly available under github.com/epfml/cola .

1.1 Problem statement

Setup. Many machine learning and signal processing models are formulated as a composite convex optimization problem of the form

$$\min_{\mathbf{u}} \; l(\mathbf{u}) + r(\mathbf{u}),$$

where l is a convex loss function of a linear predictor over data and r is a convex regularizer. Some cornerstone applications include e.g. logistic regression, SVMs, Lasso, generalized linear models, each combined with or without L1, L2 or elastic-net regularization. Following the setup of [Dünner et al., 2016, Smith et al., 2018], these training problems can be mapped to either of the two following formulations, which are dual to each other

$$\min_{\mathbf{x}\in\mathbb{R}^n} \left[F_A(\mathbf{x}) := f(\mathbf{A}\mathbf{x}) + \textstyle\sum_i g_i(x_i) \right] \tag{A}$$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \left[F_B(\mathbf{w}) := f^*(\mathbf{w}) + \textstyle\sum_i g_i^*(-\mathbf{A}_i^\top \mathbf{w}) \right], \tag{B}$$

where f^*, g_i^* are the convex conjugates of f and g_i, respectively. Here $\mathbf{x} \in \mathbb{R}^n$ is a parameter vector and $\mathbf{A} := [\mathbf{A}_1; \dots; \mathbf{A}_n] \in \mathbb{R}^{d\times n}$ is a data matrix with column vectors $\mathbf{A}_i \in \mathbb{R}^d, i \in [n]$. We assume that f is smooth (Lipschitz gradient) and $g(\mathbf{x}) := \sum_{i=1}^n g_i(x_i)$ is *separable*.

Data partitioning. As in [Jaggi et al., 2014, Dünner et al., 2016, Smith et al., 2018], we assume the dataset $\mathbf{A}$ is distributed over K machines according to a partition $\{\mathcal{P}_k\}_{k=1}^K$ of the *columns* of $\mathbf{A}$. Note that this convention maintains the flexibility of partitioning the training dataset either by samples (through mapping applications to (B), e.g. for SVMs) or by features (through mapping applications to (A), e.g. for Lasso or L1-regularized logistic regression). For $\mathbf{x} \in \mathbb{R}^n$, we write $\mathbf{x}_{[k]} \in \mathbb{R}^n$ for the n-vector with elements $(\mathbf{x}_{[k]})_i := x_i$ if $i \in \mathcal{P}_k$ and $(\mathbf{x}_{[k]})_i := 0$ otherwise, and analogously $\mathbf{A}_{[k]} \in \mathbb{R}^{d\times n_k}$ for the corresponding set of local data columns on node k, which is of size $n_k = |\mathcal{P}_k|$.

Network topology. We consider the task of joint training of a global machine learning model in a decentralized network of K nodes. Its connectivity is modelled by a mixing matrix $\mathcal{W} \in \mathbb{R}_+^{K\times K}$. More precisely, $\mathcal{W}_{ij} \in [0, 1]$ denotes the connection strength between nodes i and j, with a non-zero weight indicating the existence of a pairwise communication link. We assume $\mathcal{W}$ to be symmetric and doubly stochastic, which means each row and column of $\mathcal{W}$ sums to one.

The spectral properties of $\mathcal{W}$ used in this paper are that the eigenvalues of $\mathcal{W}$ are real, and $1 = \lambda_1(\mathcal{W}) \geq \cdots \geq \lambda_n(\mathcal{W}) \geq -1$. Let the second largest magnitude of the eigenvalues of $\mathcal{W}$ be $\beta := \max\{|\lambda_2(\mathcal{W})|, |\lambda_n(\mathcal{W})|\}$. $1 - \beta$ is called the *spectral gap*, a quantity well-studied in graph theory and network analysis. The spectral gap measures the level of connectivity among nodes. In the extreme case when $\mathcal{W}$ is diagonal, and thus an identity matrix, the spectral gap is 0 and there is no communication among nodes. To ensure convergence of decentralized algorithms, we impose the standard assumption of positive spectral gap of the network which includes all connected graphs, such as e.g. a ring or 2-D grid topology, see also Appendix B for details.

1.2 Related work

Research in decentralized optimization dates back to the 1980s with the seminal work of Bertsekas and Tsitsiklis, cf. [Tsitsiklis et al., 1986]. Their framework focuses on the minimization of a (smooth) function by distributing the components of the parameter vector $\mathbf{x}$ among agents. In contrast, a second more recent line of work [Nedic and Ozdaglar, 2009, Duchi et al., 2012, Shi et al., 2015, Mokhtari and Ribeiro, 2016, Nedic et al., 2017, Scaman et al., 2017, 2018] considers minimization of a sum of individual local cost-functions $F(\mathbf{x}) = \sum_i F_i(\mathbf{x})$, which are potentially non-smooth. Our work here can be seen as bridging the two scenarios to the primal-dual setting (A) and (B).

While decentralized optimization is a relatively mature area in the operations research and automatic control communities, it has recently received a surge of attention for machine learning applications, see e.g. [Cevher et al., 2014]. Decentralized gradient descent (DGD) with diminishing stepsizes was proposed by [Nedic and Ozdaglar, 2009, Jakovetic et al., 2012], showing convergence to the optimal solution at a sublinear rate. [Yuan et al., 2016] further prove that DGD will converge to the neighborhood of a global optimum at a linear rate when used with a constant stepsize for strongly convex objectives. [Shi et al., 2015] present EXTRA, which offers a significant performance boost compared to DGD by using a gradient tracking technique. [Nedic et al., 2017] propose the DIGing algorithm to handle a time-varying network topology. For a static and symmetric $\mathcal{W}$, DIGing recovers EXTRA by redefining the two mixing matrices in EXTRA. The dual averaging method [Duchi et al., 2012] converges at a sublinear rate with a dynamic stepsize. Under a strong convexity assumption, decomposition techniques such as decentralized ADMM (DADMM, also known as consensus ADMM) have linear convergence for time-invariant undirected graphs, if subproblems are solved exactly [Shi et al., 2014, Wei and Ozdaglar, 2013]. DADMM+ [Bianchi et al., 2016] is a different primal-dual approach with more efficient closed-form updates in each step (as compared to ADMM), and is proven to converge but without a rate. Compared to CoLa, neither of DADMM and DADMM+ can be flexibly adapted to the communication-computation tradeoff due to their fixed update definition, and both require additional hyperparameters to tune in each use-case (including the ρ from ADMM). Notably CoLa shows superior performance compared to DIGing and decentralized ADMM in our experiments. [Scaman et al., 2017, 2018] present lower complexity bounds and optimal algorithms for objectives in the form $F(\mathbf{x}) = \sum_i F_i(\mathbf{x})$. Specifically, [Scaman et al., 2017] assumes each $F_i(\mathbf{x})$ is smooth and strongly convex, and [Scaman et al., 2018] assumes each $F_i(\mathbf{x})$ is Lipschitz continuous and convex. Additionally [Scaman et al., 2018] needs a boundedness constraint for the input problem. In contrast, CoLa can handle non-smooth and non-strongly convex objectives (A) and (B), suited to the mentioned applications in machine learning and signal processing. For smooth nonconvex models, [Lian et al., 2017] demonstrate that a variant of decentralized parallel SGD can outperform the centralized variant when the network latency is high. They further extend it to the asynchronous setting [Lian et al., 2018] and to deal with large data variance among nodes [Tang et al., 2018a] or with unreliable network links [Tang et al., 2018b]. For the decentralized, asynchronous consensus optimization, [Wu et al., 2018] extends the existing PG-EXTRA and proves convergence of the algorithm. [Sirb and Ye, 2018] proves a $O(K/\epsilon^2)$ rate for stale and stochastic gradients. [Lian et al., 2018] achieves $O(1/\epsilon)$ rate and has linear speedup with respect to number of workers.

In the distributed setting with a central server, algorithms of the CoCoA family [Yang, 2013, Jaggi et al., 2014, Ma et al., 2015, Dünner et al., 2018]—see [Smith et al., 2018] for a recent overview—are targeted for problems of the forms (A) and (B). For convex models, CoCoA has shown to significantly outperform competing methods including e.g., ADMM, distributed SGD etc. Other centralized algorithm representatives are parallel SGD variants such as [Agarwal and Duchi, 2011, Zinkevich et al., 2010] and more recent distributed second-order methods [Zhang and Lin, 2015, Reddi et al., 2016, Gargiani, 2017, Lee and Chang, 2017, Dünner et al., 2018, Lee et al., 2018].

In this paper we extend CoCoA to the challenging *decentralized* environment—with no central coordinator—while maintaining all of its nice properties. We are not aware of any existing primal-dual methods in the decentralized setting, except the recent work of [Smith et al., 2017] on federated learning for the special case of multi-task learning problems. Federated learning was first described by [Konecnỳ et al., 2015, 2016, McMahan et al., 2017] as decentralized learning for on-device learning applications, combining a global shared model with local personalized models. Current

Algorithm 1: CoLa: **Co**mmunication-Efficient Decentralized **L**inear Le**a**rning

Input: Data matrix $\mathbf{A}$ distributed column-wise according to partition $\{\mathcal{P}_k\}_{k=1}^K$. Mixing matrix $\mathcal{W}$. Aggregation parameter $\gamma \in [0,1]$, and local subproblem parameter σ' as in (1). Starting point $\mathbf{x}^{(0)} := \mathbf{0} \in \mathbb{R}^n$, $\mathbf{v}^{(0)} := \mathbf{0} \in \mathbb{R}^d$, $\mathbf{v}_k^{(0)} := \mathbf{0} \in \mathbb{R}^d \; \forall \, k = 1, \dots K$;

for $t = 0, 1, 2, \dots, T$ **do**
  for $k \in \{1, 2, \dots, K\}$ **in parallel over all nodes do**
    compute locally averaged shared vector $\mathbf{v}_k^{(t+\frac{1}{2})} := \sum_{l=1}^K \mathcal{W}_{kl} \mathbf{v}_l^{(t)}$
    $\Delta\mathbf{x}_{[k]} \leftarrow \Theta$-approximate solution to subproblem (1) at $\mathbf{v}_k^{(t+\frac{1}{2})}$
    update local variable $\mathbf{x}_{[k]}^{(t+1)} := \mathbf{x}_{[k]}^{(t)} + \gamma \, \Delta\mathbf{x}_{[k]}$
    compute update of local estimate $\Delta\mathbf{v}_k := \mathbf{A}_{[k]} \Delta\mathbf{x}_{[k]}$
    $\mathbf{v}_k^{(t+1)} := \mathbf{v}_k^{(t+\frac{1}{2})} + \gamma K \Delta\mathbf{v}_k$
  end
end

federated optimization algorithms (like FedAvg in [McMahan et al., 2017]) are still close to the centralized setting. In contrast, our work provides a fully decentralized alternative algorithm for federated learning with generalized linear models.

2 The decentralized algorithm: CoLa

The CoLa framework is summarized in Algorithm 1. For a given input problem we map it to either of the (A) or (B) formulation, and define the locally stored dataset $\mathbf{A}_{[k]}$ and local part of the weight vector $\mathbf{x}_{[k]}$ in node k accordingly. While $\mathbf{v} = \mathbf{A}\mathbf{x}$ is the shared state being communicated in CoCoA, this is generally unknown to a node in the fully decentralized setting. Instead, we maintain $\mathbf{v}_k$, a local estimate of $\mathbf{v}$ in node k, and use it as a surrogate in the algorithm.

New data-local quadratic subproblems. During a computation step, node k locally solves the following minimization problem

$$\min_{\Delta\mathbf{x}_{[k]} \in \mathbb{R}^n} \; \mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]}; \mathbf{v}_k, \mathbf{x}_{[k]}), \tag{1}$$

where

$$\begin{aligned} \mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]}; \mathbf{v}_k, \mathbf{x}_{[k]}) := \; & \tfrac{1}{K} f(\mathbf{v}_k) + \nabla f(\mathbf{v}_k)^\top \mathbf{A}_{[k]} \Delta\mathbf{x}_{[k]} \\ & + \tfrac{\sigma'}{2\tau} \left\| \mathbf{A}_{[k]} \Delta\mathbf{x}_{[k]} \right\|^2 + \textstyle\sum_{i \in \mathcal{P}_k} g_i(x_i + (\Delta\mathbf{x}_{[k]})_i). \end{aligned} \tag{2}$$

Crucially, this subproblem only depends on the local data $\mathbf{A}_{[k]}$, and local vectors $\mathbf{v}_l$ from the neighborhood of the current node k. In contrast, in CoCoA [Smith et al., 2018] the subproblem is defined in terms of a global aggregated shared vector $\mathbf{v}_c := \mathbf{A}\mathbf{x} \in \mathbb{R}^d$, which is not available in the decentralized setting.[2] The aggregation parameter $\gamma \in [0,1]$ does not need to be tuned; in fact, we use the default $\gamma := 1$ throughout the paper, see [Ma et al., 2015] for a discussion. Once γ is settled, a safe choice of the subproblem relaxation parameter σ' is given as $\sigma' := \gamma K$. σ' can be additionally tightened using an improved Hessian subproblem (Appendix E.3).

Algorithm description. At time t on node k, $\mathbf{v}_k^{(t+\frac{1}{2})}$ is a local estimate of the shared variable after a communication step (i.e. gossip mixing). The local subproblem (1) based on this estimate is solved

[2] *Subproblem interpretation:* Note that for the special case of $\gamma := 1$, $\sigma' := K$, by smoothness of f, our subproblem in (2) is an upper bound on

$$\min_{\Delta\mathbf{x}_{[k]} \in \mathbb{R}^n} \tfrac{1}{K} f(\mathbf{A}(\mathbf{x} + K\Delta\mathbf{x}_{[k]})) + \textstyle\sum_{i \in \mathcal{P}_k} g_i(x_i + (\Delta\mathbf{x}_{[k]})_i), \tag{3}$$

which is a scaled block-coordinate update of block k of the original objective (A). This assumes that we have consensus $\mathbf{v}_k \equiv \mathbf{A}\mathbf{x} \; \forall \, k$. For *quadratic* objectives (i.e. when $f \equiv \|.\|_2^2$ and $\mathbf{A}$ describes the quadratic), the equality of the formulations (2) and (3) holds. Furthermore, by convexity of f, the sum of (3) is an upper bound on the centralized updates $f(\mathbf{x} + \Delta\mathbf{x}) + g(\mathbf{x} + \Delta\mathbf{x})$. Both inequalities quantify the overhead of the distributed algorithm over the centralized version, see also [Yang, 2013, Ma et al., 2015, Smith et al., 2018] for the non-decentralized case.

and yields $\Delta\mathbf{x}_{[k]}$. Then we calculate $\Delta\mathbf{v}_k := \mathbf{A}_{[k]}\Delta\mathbf{x}_{[k]}$, and update the local shared vector $\mathbf{v}_k^{(t+1)}$. We allow the local subproblem to be solved approximately:

Assumption 1 (Θ-approximation solution)**.** *Let $\Theta \in [0,1]$ be the* relative accuracy *of the local solver (potentially randomized), in the sense of returning an approximate solution $\Delta\mathbf{x}_{[k]}$ at each step t, s.t.*

$$\frac{\mathbb{E}[\mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]};\mathbf{v}_k,\mathbf{x}_{[k]}) - \mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]}^\star;\mathbf{v}_k,\mathbf{x}_{[k]})]}{\mathscr{G}_k^{\sigma'}(\ \mathbf{0}\ ;\mathbf{v}_k,\mathbf{x}_{[k]})\ - \mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]}^\star;\mathbf{v}_k,\mathbf{x}_{[k]})} \le \Theta,$$

where $\Delta\mathbf{x}_{[k]}^\star \in \arg\min_{\Delta\mathbf{x}\in\mathbb{R}^n} \mathscr{G}_k^{\sigma'}(\Delta\mathbf{x}_{[k]};\mathbf{v}_k,\mathbf{x}_{[k]})$, for each $k \in [K]$.

Elasticity to network size, compute resources and changing data—and fault tolerance. Real-world communication networks are not homogeneous and static, but greatly vary in availability, computation, communication and storage capacity. Also, the training data is subject to changes. While these issues impose significant challenges for most existing distributed training algorithms, we hereby show that CoLa offers adaptivity to such dynamic and heterogenous scenarios.

Scalability and elasticity in terms of availability and computational capacity can be modelled by a node-specific local accuracy parameter Θ_k in Assumption 1, as proposed by [Smith et al., 2017]. The more resources node k has, the more accurate (smaller) Θ_k we can use. The same mechanism also allows dealing with fault tolerance and stragglers, which is crucial e.g. on a network of personal devices. More specifically, when a new node k joins the network, its $\mathbf{x}_{[k]}$ variables are initialized to $\mathbf{0}$; when node k leaves, its $\mathbf{x}_{[k]}$ is frozen, and its subproblem is not touched anymore (i.e. $\Theta_k = 1$). Using the same approach, we can adapt to dynamic changes in the dataset—such as additions and removal of local data columns—by adjusting the size of the local weight vector accordingly. Unlike gradient-based methods and ADMM, CoLa does not require parameter tuning to converge, increasing resilience to drastic changes.

Extension to improved second-order subproblems. In the centralized setting, it has recently been shown that the Hessian information of f can be properly utilized to define improved local subproblems [Lee and Chang, 2017, Dünner et al., 2018]. Similar techniques can be applied to CoLa as well, details on which are left in Appendix E.

Extension to time-varying graphs. Similar to scalability and elasticity, it is also straightforward to extend CoLa to a time varying graph under proper assumptions. If we use the time-varying model in [Nedic et al., 2017, Assumption 1], where an undirected graph is connected with B gossip steps, then changing CoLa to perform B communication steps and one computation step per round still guarantees convergence. Details of this setup are provided in Appendix E.

3 On the convergence of CoLa

In this section we present a convergence analysis of the proposed decentralized algorithm CoLa for both general convex and strongly convex objectives. In order to capture the evolution of CoLa, we reformulate the original problem (A) by incorporating both $\mathbf{x}$ and local estimates $\{\mathbf{v}_k\}_{k=1}^K$

$$\begin{aligned} &\min_{\mathbf{x},\{\mathbf{v}_k\}_{k=1}^K} \mathcal{H}_A(\mathbf{x},\{\mathbf{v}_k\}_{k=1}^K) := \tfrac{1}{K}\textstyle\sum_{k=1}^K f(\mathbf{v}_k) + g(\mathbf{x}), \\ &\text{such that} \qquad \mathbf{v}_k = \mathbf{A}\mathbf{x},\ k = 1,...,K. \end{aligned} \tag{DA}$$

While the consensus is not always satisfied during Algorithm 1, the following relations between the decentralized objective and the original one (A) always hold. All proofs are deferred to Appendix C.

Lemma 1. *Let $\{\mathbf{v}_k\}$ and $\mathbf{x}$ be the iterates generated during the execution of Algorithm 1. At any timestep, it holds that*

$$\tfrac{1}{K}\textstyle\sum_{k=1}^K \mathbf{v}_k = \mathbf{A}\mathbf{x}, \tag{4}$$

$$F_A(\mathbf{x}) \le \mathcal{H}_A(\mathbf{x},\{\mathbf{v}_k\}_{k=1}^K) \le F_A(\mathbf{x}) + \tfrac{1}{2\tau K}\textstyle\sum_{k=1}^K \|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|^2. \tag{5}$$

The dual problem and duality gap of the decentralized objective (DA) are given in Lemma 2.

Lemma 2 (Decentralized Dual Function and Duality Gap)**.** *The Lagrangian dual of the decentralized formation (DA) is*

$$\min_{\{\mathbf{w}_k\}_{k=1}^K} \mathcal{H}_B(\{\mathbf{w}_k\}_{k=1}^K) := \tfrac{1}{K}\textstyle\sum_{k=1}^K f^*(\mathbf{w}_k) + \sum_{i=1}^n g_i^*\Big(-\mathbf{A}_i^\top(\tfrac{1}{K}\sum_{k=1}^K \mathbf{w}_k)\Big). \tag{DB}$$

Given primal variables $\{\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K\}$ *and dual variables* $\{\mathbf{w}_k\}_{k=1}^K$*, the duality gap is:*

$$G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\mathbf{w}_k\}_{k=1}^K) := \tfrac{1}{K}\sum_k (f(\mathbf{v}_k)+f^*(\mathbf{w}_k)) + g(\mathbf{x}) + \sum_{i=1}^n g_i^*\left(-\tfrac{1}{K}\sum_k \mathbf{A}_i^\top \mathbf{w}_k\right). \quad (6)$$

If the dual variables are fixed to the optimality condition $\mathbf{w}_k = \nabla f(\mathbf{v}_k)$, then the dual variables can be omitted in the argument list of duality gap, namely $G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K)$. Note that the decentralized duality gap generalizes the duality gap of CoCoA: when consensus is ensured, i.e., $\mathbf{v}_k \equiv \mathbf{A}\mathbf{x}$ and $\mathbf{w}_k \equiv \nabla f(\mathbf{A}\mathbf{x})$, the decentralized duality gap recovers that of CoCoA.

3.1 Linear rate for strongly convex objectives

We use the following data-dependent quantities in our main theorems

$$\sigma_k := \max_{\mathbf{x}_{[k]}\in\mathbb{R}^n} \left\|\mathbf{A}_{[k]}\mathbf{x}_{[k]}\right\|^2 / \|\mathbf{x}_{[k]}\|^2, \ \sigma_{\max} = \max_{k=1,\dots,K}\sigma_k, \ \sigma := \sum_{k=1}^K \sigma_k n_k. \quad (7)$$

If $\{g_i\}$ are strongly convex, CoLA achieves the following linear rate of convergence.

Theorem 1 (Strongly Convex g_i)**.** *Consider Algorithm 1 with* $\gamma := 1$ *and let* Θ *be the quality of the local solver in Assumption 1. Let* g_i *be* μ_g*-strongly convex for all* $i \in [n]$ *and let* f *be* $1/\tau$*-smooth. Let* $\bar{\sigma}' := (1+\beta)\sigma'$, $\alpha := (1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta})^{-1}$ *and* $\eta := \gamma(1-\Theta)(1-\alpha)$

$$s_0 = \tfrac{\tau\mu_g}{\tau\mu_g + \sigma_{max}\bar{\sigma}'} \in [0,1]. \quad (8)$$

Then after T *iterations of Algorithm 1 with*[3]

$$T \geq \tfrac{1+\eta s_0}{\eta s_0}\log\tfrac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{\mathcal{H}}},$$

it holds that $\mathbb{E}\left[\mathcal{H}_A(\mathbf{x}^{(T)}, \{\mathbf{v}_k^{(T)}\}_{k=1}^K) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}_k^\star\}_{k=1}^K)\right] \leq \varepsilon_{\mathcal{H}}$*. Furthermore, after* T *iterations with*

$$T \geq \tfrac{1+\eta s_0}{\eta s_0}\log\left(\tfrac{1}{\eta s_0}\tfrac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{G_{\mathcal{H}}}},\right)$$

we have the expected duality gap $\mathbb{E}[G_{\mathcal{H}}(\mathbf{x}^{(T)}, \{\sum_{k=1}^K \mathcal{W}_{kl}\mathbf{v}_l^{(T)}\}_{k=1}^K)] \leq \varepsilon_{G_{\mathcal{H}}}$.

3.2 Sublinear rate for general convex objectives

Models such as sparse logistic regression, Lasso, group Lasso are non-strongly convex. For such models, we show that CoLA enjoys a $\mathcal{O}(1/T)$ sublinear rate of convergence for all network topologies with a positive spectral gap.

Theorem 2 (Non-strongly Convex Case)**.** *Consider Algorithm 1, using a local solver of quality* Θ*. Let* $g_i(\cdot)$ *have L-bounded support, and let* f *be* $(1/\tau)$*-smooth. Let* $\varepsilon_{G_{\mathcal{H}}} > 0$ *be the desired duality gap. Then after* T *iterations where*

$$T \geq T_0 + \max\left\{\left\lceil\tfrac{1}{\eta}\right\rceil, \tfrac{4L^2\sigma\bar{\sigma}'}{\tau\varepsilon_{G_{\mathcal{H}}}\eta}\right\}, \qquad T_0 \geq t_0 + \left[\tfrac{2}{\eta}\left(\tfrac{8L^2\sigma\bar{\sigma}'}{\tau\varepsilon_{G_{\mathcal{H}}}} - 1\right)\right]_+$$

$$t_0 \geq \max\left\{0, \left\lceil\tfrac{1+\eta}{\eta}\log\tfrac{2\tau(\mathcal{H}_A(\mathbf{x}^{(0)}, \{\mathbf{v}_l^{(0)}\}) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}^\star\}))}{4L^2\sigma\bar{\sigma}'}\right\rceil\right\}$$

and $\bar{\sigma}' := (1+\beta)\sigma'$, $\alpha := (1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta})^{-1}$ *and* $\eta := \gamma(1-\Theta)(1-\alpha)$*. We have that the expected duality gap satisfies*

$$\mathbb{E}\left[G_{\mathcal{H}}(\bar{\mathbf{x}}, \{\bar{\mathbf{v}}_k\}_{k=1}^K, \{\bar{\mathbf{w}}_k\}_{k=1}^K)\right] \leq \varepsilon_{G_{\mathcal{H}}}$$

at the averaged iterate $\bar{\mathbf{x}} := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1}\mathbf{x}^{(t)}$*, and* $\mathbf{v}_k' := \sum_{l=1}^K \mathcal{W}_{kl}\mathbf{v}_l$ *and* $\bar{\mathbf{v}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1}(\mathbf{v}_k')^{(t)}$ *and* $\bar{\mathbf{w}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1}\nabla f((\mathbf{v}_k')^{(t)})$.

Note that the assumption of bounded support for the g_i functions is not restrictive in the general convex case, as discussed e.g. in [Dünner et al., 2016].

[3] $\varepsilon_{\mathcal{H}}^{(0)} := \mathcal{H}_A(\mathbf{x}^{(0)}, \{\mathbf{v}_k^{(0)}\}_{k=1}^K) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}_k^\star\}_{k=1}^K)$ is the initial suboptimality.

3.3 Local certificates for global accuracy

Accuracy certificates for the training error are very useful for practitioners to diagnose the learning progress. In the centralized setting, the duality gap serves as such a certificate, and is available as a stopping criterion on the master node. In the decentralized setting of our interest, this is more challenging as consensus is not guaranteed. Nevertheless, we show in the following Proposition 1 that certificates for the decentralized objective (DA) can be computed from local quantities:

Proposition 1 (Local Certificates)**.** *Assume g_i has L-bounded support, and let $\mathcal{N}_k := \{j : \mathcal{W}_{jk} > 0\}$ be the set of nodes accessible to node k. Then for any given $\varepsilon > 0$, we have*

$$G_{\mathcal{H}}(\mathbf{x}; \{\mathbf{v}_k\}_{k=1}^K) \leq \varepsilon,$$

if for all $k = 1, \dots, K$ the following two local conditions are satisfied:

$$\mathbf{v}_k^\top \nabla f(\mathbf{v}_k) + \sum_{i \in \mathcal{P}_k} \left(g_i(\mathbf{x}_i) + g_i^*(-\mathbf{A}_i^\top \nabla f(\mathbf{v}_k)) \right) \leq \frac{\varepsilon}{2K} \tag{9}$$

$$\left\| \nabla f(\mathbf{v}_k) - \tfrac{1}{|\mathcal{N}_k|} \textstyle\sum_{j \in \mathcal{N}_k} \nabla f(\mathbf{v}_j) \right\|_2 \leq \left(\textstyle\sum_{k=1}^K n_k^2 \sigma_k \right)^{-1/2} \tfrac{1-\beta}{2L\sqrt{K}} \varepsilon, \tag{10}$$

The local conditions (9) and (10) have a clear interpretation. The first one ensures the duality gap of the local subproblem given by $\mathbf{v}_k$ as on the left hand side of (9) is small. The second condition (10) guarantees that consensus violation is bounded, by ensuring that the gradient of each node is similar to its neighborhood nodes.

Remark 1. *The resulting certificate from Proposition 1 is local, in the sense that no global vector aggregations are needed to compute it. For a certificate on the global objective, the boolean flag of each local condition (9) and (10) being satisfied or not needs to be shared with all nodes, but this requires extremely little communication. Exact values of the parameters β and $\sum_{k=1}^K n_k^2 \sigma_k$ are not required to be known, and any valid upper bound can be used instead. We can use the local certificates to avoid unnecessary work on local problems which are already optimized, as well as to continuously quantify how newly arriving local data has to be re-optimized in the case of online training. The local certificates can also be used to quantify the contribution of newly joining or departing nodes, which is particularly useful in the elastic scenario described above.*

4 Experimental results

Here we illustrate the advantages of CoLa in three respects: firstly we investigate the application in different network topologies and with varying subproblem quality Θ; secondly, we compare CoLa with state-of-the-art decentralized baselines: ①, DIGing [Nedic et al., 2017], which generalizes the gradient-tracking technique of the EXTRA algorithm [Shi et al., 2015], and ②, Decentralized ADMM (aka. consensus ADMM), which extends the classical ADMM (Alternating Direction Method of

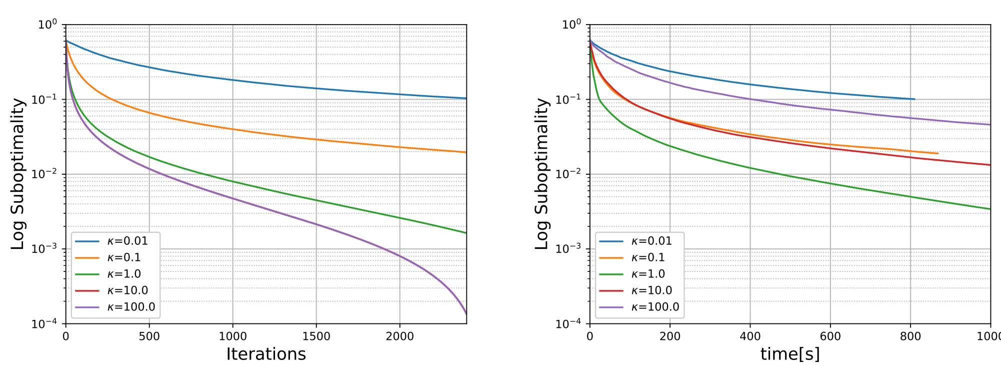

Figure 1: Suboptimality for solving Lasso (λ=10^{-6}) for the RCV1 dataset on a ring of 16 nodes. We illustrate the performance of CoLa: a) number of iterations; b) time. κ here denotes the number of local data passes per communication round.

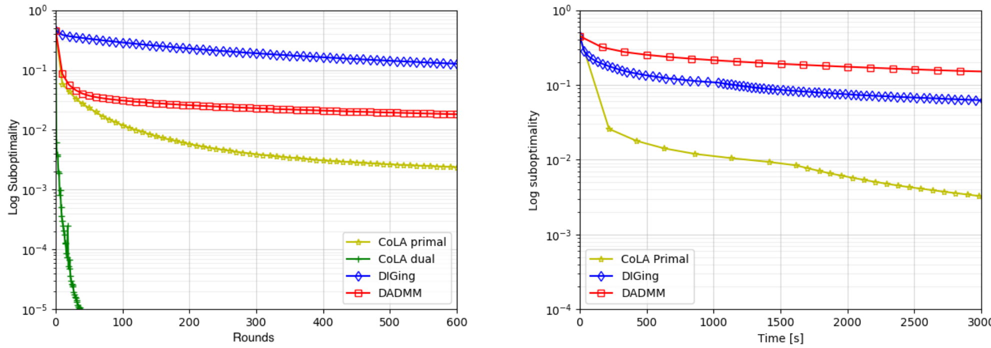

Figure 2: Convergence of CoLA for solving problems on a ring of K=16 nodes. Left) Ridge regression on URL reputation dataset (λ=10^{-4}); Right) Lasso on webspam dataset (λ=10^{-5}).

Multipliers) method [Boyd et al., 2011] to the decentralized setting [Shi et al., 2014, Wei and Ozdaglar, 2013]; Finally, we show that CoLA works in the challenging unreliable network environment where each node has a certain chance to drop out of the network.

We implement all algorithms in PyTorch with MPI backend. The decentralized network topology is simulated by running one thread per graph node, on a 2×12 core Intel Xeon CPU E5-2680 v3 server with 256 GB RAM. Table 1 describes the datasets[4] used in the experiments. For Lasso, the columns of $\mathbf{A}$ are features. For ridge regression, the columns are features and samples for CoLA primal and CoLA dual, respectively. The order of columns is shuffled once before being distributed across the nodes. Due to space limit, details on the experimental configurations are included in Appendix D.

Table 1: Datasets Used for Empirical Study

Dataset	#Training	#Features	Sparsity
URL	2M	3M	3.5e-5
Webspam	350K	16M	2.0e-4
Epsilon	400K	2K	1.0
RCV1 Binary	677K	47K	1.6e-3

Effect of approximation quality Θ. We study the convergence behavior in terms of the approximation quality Θ. Here, Θ is controlled by the number of data passes κ on subproblem (1) per node. Figure 1 shows that increasing κ always results in less number of iterations (less communication rounds) for CoLA. However, given a fixed network bandwidth, it leads to a clear trade-off for the overall wall-clock time, showing the cost of both communication and computation. Larger κ leads to less communication rounds, however, it also takes more time to solve subproblems. The observations suggest that one can adjust Θ for each node to handle system heterogeneity, as what we have discussed at the end of Section 2.

Effect of graph topology. Fixing K=16, we test the performance of CoLA on 5 different topologies: ring, 2-connected cycle, 3-connected cycle, 2D grid and complete graph. The mixing matrix $\mathcal{W}$ is given by Metropolis weights for all test cases (details in Appendix B). Convergence curves are plotted in Figure 3. One can observe that for all topologies, CoLA converges monotonically and especailly when all nodes in the network are equal, smaller β leads to a faster convergence rate. This is consistent with the intuition that $1-\beta$ measures the connectivity level of the topology.

Superior performance compared to baselines. We compare CoLA with DIGing and D-ADMM for strongly and general convex problems. For general convex objectives, we use Lasso regression with $\lambda = 10^{-4}$ on the webspam dataset; for the strongly convex objective, we use Ridge regression with $\lambda = 10^{-5}$ on the URL reputation dataset. For Ridge regression, we can map CoLA to both primal and dual problems. Figure 2 traces the results on log-suboptimality. One can observe that for both generally and strongly convex objectives, CoLA significantly outperforms DIGing and decentralized ADMM in terms of number of communication rounds and computation time. While DIGing and D-ADMM need parameter tuning to ensure convergence and efficiency, CoLA is much easier to deploy as it is parameter free. Additionally, convergence guarantees of ADMM relies on exact subproblem solvers, whereas inexact solver is allowed for CoLA.

[4] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

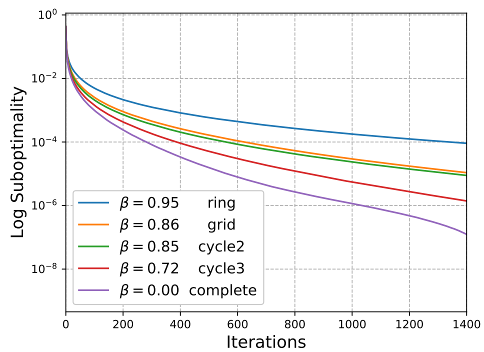

Figure 3: Performance comparison of CoLa on different topologies. Solving Lasso regression (λ=10^{-6}) for RCV1 dataset with 16 nodes.

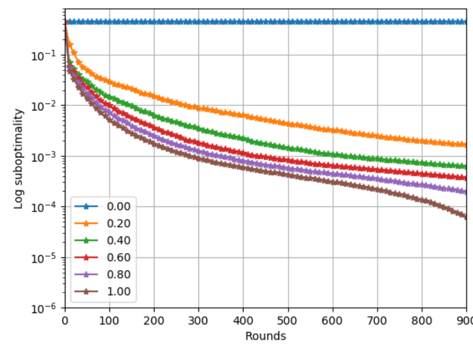

Figure 4: Performance of CoLa when nodes have p chance of staying in the network on the URL dataset (λ=10^{-4}). Freezing $\mathbf{x}_{[k]}$ when node k leaves the network.

Fault tolerance to unreliable nodes. Assume each node of a network only has a chance of p to participate in each round. If a new node k joins the network, then local variables are initialized as $\mathbf{x}_{[k]} = 0$; if node k leaves the network, then $\mathbf{x}_{[k]}$ will be frozen with $\Theta_k = 1$. All remaining nodes dynamically adjust their weights to maintain the doubly stochastic property of $\mathcal{W}$. We run CoLa on such unreliable networks of different ps and show the results in Figure 4. First, one can observe that for all $p > 0$ the suboptimality decreases monotonically as CoLa progresses. It is also clear from the result that a smaller dropout rate (a larger p) leads to a faster convergence of CoLa.

5 Discussion and conclusions

In this work we have studied training generalized linear models in the fully decentralized setting. We proposed a communication-efficient decentralized framework, termed CoLa, which is free of parameter tuning. We proved that it has a sublinear rate of convergence for general convex problems, allowing e.g. L1 regularizers, and has a linear rate of convergence for strongly convex objectives. Our scheme offers primal-dual certificates which are useful in the decentralized setting. We demonstrated that CoLa offers full adaptivity to heterogenous distributed systems on arbitrary network topologies, and is adaptive to changes in network size and data, and offers fault tolerance and elasticity. Future research directions include improving subproblems, as well as extension to the network topology with directed graphs, as well as recent communication compression schemes [Stich et al., 2018].

Acknowledgments. We thank Prof. Bharat K. Bhargava for fruitful discussions. We acknowledge funding from SNSF grant 200021_175796, Microsoft Research JRC project 'Coltrain', as well as a Google Focused Research Award.

References

John N Tsitsiklis, Dimitri P Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

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

J C Duchi, A Agarwal, and M J Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, March 2012.

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

Aryan Mokhtari and Alejandro Ribeiro. DSA: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(61):1–35, 2016.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

Volkan Cevher, Stephen Becker, and Mark Schmidt. Convex Optimization for Big Data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, 2014.

Virginia Smith, Simone Forte, Chenxin Ma, Martin Takác, Michael I Jordan, and Martin Jaggi. CoCoA: A General Framework for Communication-Efficient Distributed Optimization. *Journal of Machine Learning Research*, 18(230):1–49, 2018.

Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with Elastic Averaging SGD. In *NIPS 2015 - Advances in Neural Information Processing Systems 28*, pages 685–693, 2015.

Jialei Wang, Weiran Wang, and Nathan Srebro. Memory and Communication Efficient Distributed Stochastic Optimization with Minibatch Prox. In *ICML 2017 - Proceedings of the 34th International Conference on Machine Learning*, pages 1882–1919, June 2017.

Celestine Dünner, Simone Forte, Martin Takác, and Martin Jaggi. Primal-Dual Rates and Certificates. In *ICML 2016 - Proceedings of the 33th International Conference on Machine Learning*, pages 783–792, 2016.

Martin Jaggi, Virginia Smith, Martin Takác, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 3068–3076, 2014.

Kevin Scaman, Francis R. Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 3027–3036, 2017.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for non-smooth distributed optimization in networks. *arXiv preprint arXiv:1806.00291*, 2018.

Dusan Jakovetic, Joao Xavier, and Jose MF Moura. Convergence rate analysis of distributed gradient methods for smooth optimization. In *Telecommunications Forum (TELFOR), 2012 20th*, pages 867–870. IEEE, 2012.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

Ermin Wei and Asuman Ozdaglar. On the O(1/k) Convergence of Asynchronous Distributed Alternating Direction Method of Multipliers. *arXiv*, July 2013.

Pascal Bianchi, Walid Hachem, and Franck Iutzeler. A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization. *IEEE Transactions on Automatic Control*, 61(10):2947–2957, 2016.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5336–5346, 2017.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *ICML 2018 - Proceedings of the 35th International Conference on Machine Learning*, 2018.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. *arXiv preprint arXiv:1803.07068*, 2018a.

Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *NIPS 2018 - Advances in Neural Information Processing Systems*, 2018b.

Tianyu Wu, Kun Yuan, Qing Ling, Wotao Yin, and Ali H Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2): 293–307, 2018.

Benjamin Sirb and Xiaojing Ye. Decentralized consensus algorithm with delayed and stochastic gradients. *SIAM Journal on Optimization*, 28(2):1232–1254, 2018.

Tianbao Yang. Trading Computation for Communication: Distributed Stochastic Dual Coordinate Ascent. In *NIPS 2014 - Advances in Neural Information Processing Systems 27*, 2013.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I Jordan, Peter Richtárik, and Martin Takác. Adding vs. Averaging in Distributed Primal-Dual Optimization. In *ICML 2015 - Proceedings of the 32th International Conference on Machine Learning*, pages 1973–1982, 2015.

Celestine Dünner, Aurelien Lucchi, Matilde Gargiani, An Bian, Thomas Hofmann, and Martin Jaggi. A Distributed Second-Order Algorithm You Can Trust. In *ICML 2018 - Proceedings of the 35th International Conference on Machine Learning*, pages 1357–1365, July 2018.

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.

Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370, 2015.

Sashank J Reddi, Jakub Konecnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

Matilde Gargiani. Hessian-CoCoA: a general parallel and distributed framework for non-strongly convex regularizers. Master's thesis, ETH Zurich, June 2017.

Ching-pei Lee and Kai-Wei Chang. Distributed block-diagonal approximation methods for regularized empirical risk minimization. *arXiv preprint arXiv:1709.03043*, 2017.

Ching-pei Lee, Cong Han Lim, and Stephen J Wright. A distributed quasi-newton algorithm for empirical risk minimization with nonsmooth regularization. In *ACM International Conference on Knowledge Discovery and Data Mining*, 2018.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated Multi-Task Learning. In *NIPS 2017 - Advances in Neural Information Processing Systems 30*, 2017.

Jakub Konecnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015.

Jakub Konecnỳ, H Brendan McMahan, Felix X Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

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

Sebastian U. Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *NIPS 2018 - Advances in Neural Information Processing Systems*, 2018.

Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57 (1):97–109, 1970.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS Workshop on Autodiff*, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Appendix

A Definitions

Definition 1 (L-Lipschitz continuity)**.** *A function $h : \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz continuous if $\forall\, \mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, it holds that*

$$|h(\mathbf{u}) - h(\mathbf{v})| \leq L\|\mathbf{u} - \mathbf{v}\|.$$

Definition 2 ($1/\tau$-Smoothness)**.** *A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is $1/\tau$-smooth if its gradient is $1/\tau$-Lipschitz continuous, or equivalently, $\forall\, \mathbf{u}, \mathbf{v}$ it holds*

$$f(\mathbf{u}) \leq f(\mathbf{v}) + \langle \nabla f(\mathbf{v}), \mathbf{u} - \mathbf{v}\rangle + \tfrac{1}{2\tau}\|\mathbf{u} - \mathbf{v}\|^2. \tag{11}$$

Definition 3 (L-Bounded support)**.** *The function $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ has L-bounded support if it holds $g(\mathbf{u}) < +\infty \Rightarrow \|\mathbf{u}\| \leq L$.*

Definition 4 (μ-Strong convexity)**.** *A function $h : \mathbb{R}^n \to \mathbb{R}$ is μ-strongly convex for $\mu \geq 0$ if $\forall\, \mathbf{u}, \mathbf{v}$, it holds $h(\mathbf{u}) \geq h(\mathbf{v}) + \langle \mathbf{s}, \mathbf{u} - \mathbf{v}\rangle + \frac{\mu}{2}\|\mathbf{u} - \mathbf{v}\|^2$, for any $\mathbf{s} \in \partial h(\mathbf{v})$, where $\partial h(\mathbf{v})$ is the subdifferential of h at $\mathbf{v}$.*

Lemma 3 (Duality between Lipschitzness and L-Bounded Support)**.** *A generalization of [Rockafellar, 2015, Corollary 13.3.3]. Given a proper convex function g it holds that g has L-bounded support w.r.t. the norm $\|.\|$ if and only if g^* is L-Lipschitz w.r.t. the dual norm $\|.\|_*$.*

B Graph topology

Let $\mathcal{E}$ be the set of edges of a graph. For time-invariant undirected graph the mixing matrix should satisfy the following properties:

1. (*Double stochasticity*) $\mathcal{W}\mathbf{1} = \mathbf{1}, \mathbf{1}^\top\mathcal{W} = \mathbf{1}^\top$;
2. (*Symmetrization*) For all i, j, $\mathcal{W}_{ij} = \mathcal{W}_{ji}$;
3. (*Edge utilization*) If $(i, j) \in \mathcal{E}$, then $\mathcal{W}_{ij} > 0$; otherwise $\mathcal{W}_{ij} = 0$.

A desired mixing matrix can be constructed using Metropolis-Hastings weights [Hastings, 1970]:

$$\mathcal{W}_{ij} = \begin{cases} 1/(1 + \max\{d_i, d_j\}), & \text{if } (i, j) \in \mathcal{E} \\ 0, & \text{if } (i, j) \notin \mathcal{E} \text{ and } j \neq i \\ 1 - \sum_{l\in\mathcal{N}_i} \mathcal{W}_{il}, & \text{if } j = i, \end{cases}$$

where $d_i = |\mathcal{N}_i|$ is the degree of node i.

C Proofs

This section consists of three parts. Tools and observations are provided in Appendix C.1; The main lemmas for the convergence analysis are proved in Appendix C.2 ; The main theorems and implications are proved in Appendix C.3.

In some circumstances, it is convenient to use notations of array of stack column vectors. For example, one can stack local estimates $\mathbf{v}_k$ to matrix $\mathbf{V} := [\mathbf{v}_1; \cdots; \mathbf{v}_K]$, $\Delta\mathbf{V} = [\Delta\mathbf{v}_1; \cdots; \Delta\mathbf{v}_K]$. The consensus vector $\mathbf{v}_c$ is repeated K times which will be stacked similarly: $\mathbf{V}_c := \mathbf{A}\mathbf{x}\mathbf{1}_K^\top = \mathbf{V}\mathbf{E}$ where $\mathbf{E} = \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top$. The consensus violation under the two notations is written as

$$\|\mathbf{V} - \mathbf{V}_c\|_F^2 = \textstyle\sum_{k=1}^K \|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|_2^2 .$$

Then Step 8 in CoLa is equivalent to

$$\mathbf{V}^{(t+1)} = \mathbf{V}^{(t)}\mathcal{W} + \gamma K \Delta\mathbf{V}^{(t)} \tag{12}$$

Besides, we also adopt following notations in the proof when there is no ambiguity: $\mathbf{v}'_k := \sum_{l=1}^K \mathcal{W}_{kl}\mathbf{v}_l$, $\mathbf{g}_k := \nabla f(\mathbf{v}_k)$, $\mathbf{g}'_k := \nabla f(\mathbf{v}'_k)$ and $\bar{\mathbf{g}} := \frac{1}{K}\sum_{k=1}^K \mathbf{g}_k$. For the decentralized duality gap $G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\mathbf{w}_k\}_{k=1}^K)$, when $\mathbf{w}_k = \nabla f(\mathbf{v}_k)$, we simplify $G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\mathbf{w}_k\}_{k=1}^K)$ to be $G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K)$ in the sequel.

On a high level, we prove the convergence rates by bounding per-iteration reduction $\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)}]$ using decentralized duality gap and other related terms, then try to obtain the final rates by properly using specific properties of the objectives.

However, the specific analysis of the new fully decentralized algorithm CoLa poses many new challenges, and we propose significantly new proof techniques in the analysis. Specifically, i) we introduce the decentralized duality gap, which is suited for the decentralized algorithm CoLa; ii) consensus violation is the usually challenging part in analyzing decentralized algorithms. Unlike using uniform bounds for consensus violations, e.g., [Yuan et al., 2016], we properly combine the consensus violation term and the objective decrease term (c.f. Lemmas 6 and 8), thus reaching arguably tight convergence bounds for both the consensus violation term and the objective.

C.1 Observations and properties

In this subsection we introduce basic lemmas. Lemma 1 establishes the relation between $\{\mathbf{v}_k\}_{k=1}^K$ and $\mathbf{v}_c$ and bounds $F_A(\mathbf{x})$ using $\mathcal{H}_A(\mathbf{x})$ and the consensus violation.

Lemma 1. *Let $\{\mathbf{v}_k\}$ and $\mathbf{x}$ be the iterates generated during the execution of Algorithm 1. At any timestep, it holds that*

$$\tfrac{1}{K}\textstyle\sum_{k=1}^K \mathbf{v}_k = \mathbf{A}\mathbf{x}, \tag{4}$$

$$F_A(\mathbf{x}) \le \mathcal{H}_A(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K) \le F_A(\mathbf{x}) + \tfrac{1}{2\tau K}\textstyle\sum_{k=1}^K \|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|^2 . \tag{5}$$

Proof of Lemma 1. Let $\widetilde{\mathbf{v}} := \frac{1}{K}\sum_{k=1}^K \mathbf{v}_k$. Using the doubly stochastic property of the matrix $\mathcal{W}$

$$\begin{aligned}\widetilde{\mathbf{v}}^{(t+1)} &= \frac{1}{K}\sum_{k=1}^K \mathbf{v}_k^{(t+1)} = \frac{1}{K}\sum_{k=1}^K\left(\sum_{l=1}^K \mathcal{W}_{kl}\mathbf{v}_l^{(t)} + \gamma K \Delta\mathbf{v}_k^{(t)}\right)\\ &= \frac{1}{K}\sum_{l=1}^K \mathbf{v}_l^{(t)} + \gamma\sum_{k=1}^K \Delta\mathbf{v}_k^{(t)} = \widetilde{\mathbf{v}}^{(t)} + \gamma\sum_{k=1}^K\Delta\mathbf{v}_k^{(t)}\end{aligned}$$

On the other hand, $\mathbf{v}_c^{(t)} := \mathbf{A}\mathbf{x}^{(t)}$ is updated based on all changes of local variables $\{\mathbf{x}_{[k]}\}_{k=1}^K$

$$\mathbf{v}_c^{(t+1)} = \mathbf{v}_c^{(t)} + \gamma\sum_{k=1}^K \Delta\mathbf{v}_k^{(t)}.$$

Since $\widetilde{\mathbf{v}}^{(0)} = \mathbf{v}_c^{(0)}$, we can conclude that $\widetilde{\mathbf{v}}^{(t)} = \mathbf{v}_c^{(t)}\ \forall\, t$. From convexity of f we know

$$F_A(\mathbf{x}) = f(\mathbf{v}_c) + g(\mathbf{x}) = f\left(\frac{1}{K}\sum_{k=1}^K \mathbf{v}_k\right) + g(\mathbf{x}) \le \frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) + g(\mathbf{x}) = \mathcal{H}(\mathbf{x})$$

Using $1/\tau$-smoothness of f gives

$$\begin{aligned}\mathcal{H}_A(\mathbf{x}) =& \frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) + g(\mathbf{x})\\ \le& \frac{1}{K}\sum_{k=1}^K\left(f(\mathbf{v}_c) + \nabla f(\mathbf{v}_c)^\top(\mathbf{v}_k - \mathbf{v}_c) + \frac{1}{2\tau}\|\mathbf{v}_k - \mathbf{v}_c\|^2\right) + g(\mathbf{x})\\ =& F_A(\mathbf{x}) + \frac{1}{2\tau K}\sum_{k=1}^K \|\mathbf{v}_k - \mathbf{v}_c\|^2 .\end{aligned}$$

□

The following lemma introduces the dual problem and the duality gap of (DA).

Lemma 2 (Decentralized Dual Function and Duality Gap)**.** *The Lagrangian dual of the decentralized formation* (DA) *is*

$$\min_{\{\mathbf{w}_k\}_{k=1}^K} \mathcal{H}_B(\{\mathbf{w}_k\}_{k=1}^K) := \tfrac{1}{K}\textstyle\sum_{k=1}^K f^*(\mathbf{w}_k) + \sum_{i=1}^n g_i^*\Big(-\mathbf{A}_i^\top(\tfrac{1}{K}\sum_{k=1}^K \mathbf{w}_k)\Big). \tag{DB}$$

Given primal variables $\{\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K\}$ *and dual variables* $\{\mathbf{w}_k\}_{k=1}^K$*, the duality gap is:*

$$G_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\mathbf{w}_k\}_{k=1}^K) := \tfrac{1}{K}\textstyle\sum_k (f(\mathbf{v}_k) + f^*(\mathbf{w}_k)) + g(\mathbf{x}) + \sum_{i=1}^n g_i^*\big(-\tfrac{1}{K}\sum_k \mathbf{A}_i^\top \mathbf{w}_k\big). \tag{6}$$

Proof. Let $\boldsymbol{\lambda}_k$ be the Lagrangian multiplier for the constraint $\mathbf{v}_k = \mathbf{A}\mathbf{x}$, the Lagrangian function is

$$L(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\boldsymbol{\lambda}_k\}_{k=1}^K) = \frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) + \sum_{i=1}^n g_i(\mathbf{x}_i) + \sum_{k=1}^K \langle \boldsymbol{\lambda}_k, \mathbf{A}\mathbf{x} - \mathbf{v}_k\rangle$$

The dual problem of (DA) follows by taking the infimum with respect to both $\mathbf{x}$ and $\{\mathbf{v}_k\}_{k=1}^K$:

$$\begin{aligned}
&\inf_{\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K} L(\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K, \{\boldsymbol{\lambda}_k\}_{k=1}^K) \\
&= \inf_{\mathbf{x}, \{\mathbf{v}_k\}_{k=1}^K} \frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) + \sum_{i=1}^n g_i(\mathbf{x}_i) + \sum_{k=1}^K \langle \boldsymbol{\lambda}_k, \mathbf{A}\mathbf{x} - \mathbf{v}_k\rangle \\
&= \sum_{k=1}^K \inf_{\{\mathbf{v}_k\}_{k=1}^K} (\frac{1}{K} f(\mathbf{v}_k) - \langle \boldsymbol{\lambda}_k, \mathbf{v}_k\rangle) + \inf_{\mathbf{x}}(\sum_{i=1}^n g_i(\mathbf{x}_i) + \sum_{k=1}^K \langle \boldsymbol{\lambda}_k, \mathbf{A}\mathbf{x}\rangle) \\
&= -\sum_{k=1}^K \sup_{\{\mathbf{v}_k\}_{k=1}^K} (\langle \boldsymbol{\lambda}_k, \mathbf{v}_k\rangle - \frac{1}{K} f(\mathbf{v}_k)) - \sup_{\mathbf{x}}(-\sum_{k=1}^K \langle \mathbf{a}_k, \mathbf{A}\mathbf{x}\rangle - \sum_{i=1}^n g_i(\mathbf{x}_i)) \\
&= -\sum_{k=1}^K \frac{1}{K} f^*(K\boldsymbol{\lambda}_k) - \sum_{i=1}^n g_i^*(-\sum_{k=1}^K \mathbf{A}_i^\top \boldsymbol{\lambda}_k)
\end{aligned}$$

Let us change variables from $\boldsymbol{\lambda}_k$ to $\mathbf{w}_k$ by setting $\mathbf{w}_k := K\boldsymbol{\lambda}_k$. If written in terms of minimization, the Lagrangian dual of $\mathcal{H}_A$ is

$$\min_{\{\mathbf{w}_k\}_{k=1}^K} \mathcal{H}_B\{\mathbf{w}_k\}_{k=1}^K = \tfrac{1}{K}\textstyle\sum_{k=1}^K f^*(\mathbf{w}_k) + \sum_{i=1}^n g_i^*\left(-\tfrac{1}{K}\sum_{k=1}^K \mathbf{A}_i^\top \mathbf{w}_k\right) \tag{13}$$

The optimality condition is that $\mathbf{w}_k = \nabla f(\mathbf{v}_k)$. Now we can see the duality gap is

$$\begin{aligned}
G_{\mathcal{H}} =& \mathcal{H}_A + \mathcal{H}_B \\
=& \frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) + \sum_{i=1}^n g_i(x_i) + \frac{1}{K}\sum_{k=1}^K f^*(\mathbf{w}_k) + \sum_{i=1}^n g_i^*\left(-\frac{1}{K}\sum_{k=1}^K \mathbf{A}_i^\top \mathbf{w}_k\right)
\end{aligned}$$

□

The following lemma correlates the consensus violation with the magnitude of the $\mathbf{v}$ parameter updates $\|\Delta\mathbf{v}_k\|_2^2$.

Lemma 4. *The consensus violation during the execution of Algorithm 1 can be bound by*

$$\sum_{k=1}^K \left\|\mathbf{v}_k^{(t+1)} - \mathbf{A}\mathbf{x}^{(t+1)}\right\|_2^2 \le \beta \sum_{k=1}^K \left\|\mathbf{v}_k^{(t)} - \mathbf{A}\mathbf{x}^{(t)}\right\|_2^2 + (1-\beta)c_1(\beta,\gamma,K)\sum_{k=1}^K \left\|\Delta\mathbf{v}_k^{(t)}\right\|_2^2 \tag{14}$$

where $c_1(\beta,\gamma,K) := \gamma^2 K^2/(1-\beta)^2$.

Proof. Consider the norm of consensus violation at time $t+1$ and apply Algo. Step 8

$$\left\|\mathbf{V}^{(t+1)} - \mathbf{V}_c^{(t+1)}\right\|_F^2 = \left\|\mathbf{V}^{(t+1)}(\mathbf{I}-\mathbf{E})\right\|_F^2 = \left\|(\mathbf{V}^{(t)}\mathcal{W} + \gamma K\Delta\mathbf{V}^{(t)})(\mathbf{I}-\mathbf{E})\right\|_F^2.$$

Further, use $\mathcal{W}(\mathbf{I}-\mathbf{E}) = (\mathbf{I}-\mathbf{E})(\mathcal{W}-\mathbf{E})$, $\|\mathbf{I}-\mathbf{E}\|_\infty = 1$, and Young's inequality with $\varepsilon_{\mathbf{v}}$

$$\left\|\mathbf{V}^{(t+1)} - \mathbf{V}_c^{(t+1)}\right\|_F^2 \le (1+\varepsilon_{\mathbf{v}}) \left\|\mathbf{V}^{(t)}(\mathbf{I}-\mathbf{E})(\mathcal{W}-\mathbf{E})\right\|_F^2 + (1+\frac{1}{\varepsilon_{\mathbf{v}}})\gamma^2K^2 \left\|\Delta\mathbf{V}^{(t)}\right\|_F^2.$$

Use the spectral property of $\mathcal{W}$ we therefore have:

$$\left\|\mathbf{V}^{(t+1)} - \mathbf{V}_c^{(t+1)}\right\|_F^2 \le (1+\varepsilon_{\mathbf{v}})\beta^2 \left\|\mathbf{V}^{(t)} - \mathbf{V}_c^{(t)}\right\|_F^2 + (1+\frac{1}{\varepsilon_{\mathbf{v}}})\gamma^2K^2 \left\|\Delta\mathbf{V}^{(t)}\right\|_F^2. \tag{15}$$

Recursively apply (15) for $i = 0, \dots, t-1$ gives

$$\left\|\mathbf{V}^{(t)} - \mathbf{V}_c^{(t)}\right\|_F^2 \le (1+\frac{1}{\varepsilon_{\mathbf{v}}})\gamma^2K^2 \sum_{i=0}^{t-1}((1+\varepsilon_{\mathbf{v}})\beta^2)^{t-1-i} \left\|\Delta\mathbf{V}^{(i)}\right\|_F^2. \tag{16}$$

Consider $\left\|\Delta\mathbf{V}^{(t)}\right\|_F^2$ generated at time t, it will be used in (16) from time $t+1, t+2, \dots$, with coefficients $1, (1+\varepsilon_{\mathbf{v}})\beta^2, ((1+\varepsilon_{\mathbf{v}})\beta^2)^2, \dots$. Sum of such coefficients are finite

$$(1+\frac{1}{\varepsilon_{\mathbf{v}}})\gamma^2K^2 \sum_{t=T}^{\infty}((1+\varepsilon_{\mathbf{v}})\beta^2)^{t-T} \le \gamma^2K^2 \frac{1+1/\varepsilon_{\mathbf{v}}}{1-(1+\varepsilon_{\mathbf{v}})\beta^2} =: c_1(\beta,\gamma,K) \tag{17}$$

where we need $(1+\varepsilon_{\mathbf{v}})\beta^2 < 1$. To minimize $c_1(\beta,\gamma,K)$ we can choose $\varepsilon_{\mathbf{v}} = 1/\beta - 1$

$$c_1(\beta,\gamma,K) = \gamma^2K^2/(1-\beta)^2 \tag{18}$$

Then (15) becomes

$$\sum_{k=1}^{K} \left\|\mathbf{v}_k^{(t+1)} - \mathbf{A}\mathbf{x}^{(t+1)}\right\|_2^2 \le \beta \sum_{k=1}^{K} \left\|\mathbf{v}_k^{(t)} - \mathbf{A}\mathbf{x}^{(t)}\right\|_2^2 + (1-\beta)c_1(\beta,\gamma,K) \sum_{k=1}^{K} \left\|\Delta\mathbf{v}_k^{(t)}\right\|_2^2$$

□

Lemma 5. *Let $\Delta\mathbf{x}_{[k]}^\star$ and $\Delta\mathbf{x}_{[k]}$ be the exact and Θ-inexact solution of subproblem $\mathscr{G}_k^{\sigma'}(\,\cdot\,;\mathbf{v}_k,\mathbf{x}_{[k]})$. The change of iterates satisfies the following inequality*

$$\frac{\sigma'}{4\tau}\sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|_2^2 \le (1+\Theta)(\mathcal{H}_A(\mathbf{0};\{\mathbf{v}_k\}) - \mathcal{H}_A(\Delta\mathbf{x}^\star;\{\mathbf{v}_k\})) \tag{19}$$

Proof. First use the Taylor expansion of $\mathscr{G}_k^{\sigma'}(\,\cdot\,;\mathbf{v}_k,\mathbf{x}_{[k]})$ and the defnition of $\Delta\mathbf{x}_{[k]}^\star$ we have

$$\frac{\sigma'}{2\tau}\left\|\mathbf{A}(\Delta\mathbf{z}-\Delta\mathbf{x}^\star)_{[k]}\right\|_2^2 \le \mathscr{G}_k^{\sigma'}\left(\Delta\mathbf{z}_{[k]};\mathbf{v}_k,\mathbf{x}_{[k]}\right) - \mathscr{G}_k^{\sigma'}\left(\Delta\mathbf{x}_{[k]}^\star;\mathbf{v}_k,\mathbf{x}_{[k]}\right) \tag{20}$$

for all $\Delta\mathbf{z}_{[k]} \in \mathbb{R}^n$ and $k = 1, \dots, K$. Apply (20) with $\Delta\mathbf{z}_{[k]} = \mathbf{0}$ for all k and sum them up yields

$$\frac{\sigma'}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}^\star\right\|_2^2 \le \mathcal{H}_A(\mathbf{0};\{\mathbf{v}_k\}) - \mathcal{H}_A(\Delta\mathbf{x}^\star;\{\mathbf{v}_k\}) \tag{21}$$

Similarly, apply (20) for $\Delta\mathbf{z}_{[k]} = \Delta\mathbf{x}_{[k]}$ for all k and sum them up gives

$$\frac{\sigma'}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}(\Delta\mathbf{x}-\Delta\mathbf{x}^\star)_{[k]}\right\|_2^2 \le \mathcal{H}_A(\Delta\mathbf{x};\{\mathbf{v}_k\}) - \mathcal{H}_A(\Delta\mathbf{x}^\star;\{\mathbf{v}_k\}) \tag{22}$$

By Assumption 1 the previous inequality becomes

$$\frac{\sigma'}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}(\Delta\mathbf{x}-\Delta\mathbf{x}^\star)_{[k]}\right\|_2^2 \le \Theta(\mathcal{H}_A(\mathbf{0};\{\mathbf{v}_k\}) - \mathcal{H}_A(\Delta\mathbf{x}^\star;\{\mathbf{v}_k\})) \tag{23}$$

The following inequality is straightforward

$$\frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|_2^2 \le \sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}^\star\right\|_2^2 + \sum_{k=1}^{K}\left\|\mathbf{A}(\Delta\mathbf{x}-\Delta\mathbf{x}^\star)_{[k]}\right\|_2^2 \tag{24}$$

Multiply (24) with $\sigma'/(2\tau)$ and use (21) and (23)

$$\frac{\sigma'}{4\tau}\sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|_2^2 \le (1+\Theta)(\mathcal{H}_A(\mathbf{0};\{\mathbf{v}_k\}) - \mathcal{H}_A(\Delta\mathbf{x}^\star;\{\mathbf{v}_k\})) \tag{25}$$

□

C.2 Main lemmas

We first present two main lemmas for the per-iteration improvement.

Lemma 6. *Let g_i be strongly convex with convexity parameter $\mu_g \geq 0$ with respect to the norm $\|\cdot\|, \forall\, i \in [n]$. Then for all iterations t of outer loop, and any $s \in [0,1]$, it holds that*

$$\begin{aligned}&\mathbb{E}\left[\mathcal{H}_A(\mathbf{x}^{(t)};\mathbf{v}_k^{(t)}) - \mathcal{H}_A(\mathbf{x}^{(t+1)};\mathbf{v}_k^{(t+1)}) - \alpha\frac{\gamma\sigma_1'}{2\tau}\sum_{k=1}^K \left\|\mathbf{A}\Delta\mathbf{x}_{[k]}^{(t)}\right\|_2^2\right] \\ \geq&\eta\left(sG_{\mathcal{H}}(\mathbf{x}^{(t)};\{\mathbf{v}_k'^{(t)}\}_{k=1}^K) - \frac{s^2\bar{\sigma}'}{2\tau}R^{(t)}\right) - \frac{9\beta\eta}{2\tau\sigma'}\sum_{k=1}^K\left\|\mathbf{v}_k^{(t)} - \mathbf{A}\mathbf{x}^{(t)}\right\|_2^2\end{aligned} \tag{26}$$

where $\alpha \in [0,1]$ is a constant and $\eta := \gamma(1-\Theta)(1-\alpha)$ and $\sigma_1' := \frac{(1-\Theta)}{2(1+\Theta)}\sigma'$ and $\bar{\sigma}' := (1+\beta)\sigma'$ and $\mathbf{v}_k' := \sum_{l=1}^K \mathcal{W}_{kl}\mathbf{v}_l$.

$$R^{(t)} := -\tfrac{\tau\mu_g(1-s)}{\bar{\sigma}'s}\left\|\mathbf{u}^{(t)} - \mathbf{x}^{(t)}\right\|^2 + \textstyle\sum_{k=1}^K \left\|\mathbf{A}(\mathbf{u}^{(t)} - \mathbf{x}^{(t)})_{[k]}\right\|^2 \tag{27}$$

for $\mathbf{u}^{(t)} \in \mathbb{R}^n$ with $\bar{\mathbf{g}}'^{(t)} := \frac{1}{K}\sum_{k=1}^K \nabla f(\mathbf{v}_k'^{(t)})$

$$u_i^{(t)} \in \partial g_i^*(-\mathbf{A}_i^\top \bar{\mathbf{g}}'^{(t)}) \qquad k \text{ s.t. } i \in \mathcal{P}_k \tag{28}$$

Proof of Lemma 6. For simplicity, we write $\mathcal{H}_A^{(t)}$ instead of $\mathcal{H}_A(\mathbf{x}^{(t)};\{\mathbf{v}_k^{(t)}\}_{k=1}^K)$ and $\mathbf{v}_k' := \sum_{i=1}^K \mathcal{W}_{ik}\mathbf{v}_i$.

$$\begin{aligned}&\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)}] \\ =&\frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) - \frac{1}{K}\sum_{k=1}^K f\left(\mathbf{v}_k' + \gamma K\Delta\mathbf{v}_{[k]}\right) + g(\mathbf{x}) - g(\mathbf{x}+\gamma\Delta\mathbf{x}) \\ =&\underbrace{\frac{1}{K}\sum_{k=1}^K f(\mathbf{v}_k) - \frac{1}{K}\sum_{k=1}^K f\left(\mathbf{v}_k'\right)}_{D_1} \\ &+\underbrace{\sum_{k=1}^K\left\{\frac{1}{K}f\left(\mathbf{v}_k'\right) + g(\mathbf{x}_{[k]})\right\} - \sum_{k=1}^K\left\{\frac{1}{K}f\left(\mathbf{v}_k' + \gamma K\Delta\mathbf{v}_{[k]}\right) + g(\mathbf{x}_{[k]} + \gamma\Delta\mathbf{x}_{[k]})\right\}}_{D_2}\end{aligned}$$

By the convexity of f, $D_1 \geq 0$. Using the convexity of f and g in D_2 we have

$$\begin{aligned}\frac{1}{\gamma}D_2 \geq&\sum_{k=1}^K\left\{\frac{1}{K}f\left(\mathbf{v}_k'\right) + g(\mathbf{x}_{[k]})\right\} - \sum_{k=1}^K\left\{\frac{1}{K}f\left(\mathbf{v}_k' + K\Delta\mathbf{v}_{[k]}\right) + g(\mathbf{x}_{[k]} + \Delta\mathbf{x}_{[k]})\right\} \\ \geq&\mathbb{E}\left[\sum_{k=1}^K \mathscr{G}_k^{\sigma'}\left(\mathbf{0};\mathbf{v}_k',\mathbf{x}_{[k]}\right) - \sum_{k=1}^K \mathscr{G}_k^{\sigma'}\left(\Delta\mathbf{x}_{[k]};\mathbf{v}_k',\mathbf{x}_{[k]}\right)\right]\end{aligned}$$

Use the Assumption 1 we have

$$\begin{aligned}D_2 \geq&\gamma\mathbb{E}\left[\sum_{k=1}^K \mathscr{G}_k^{\sigma'}\left(\mathbf{0};\mathbf{v}_k',\mathbf{x}_{[k]}\right) - \sum_{k=1}^K \mathscr{G}_k^{\sigma'}\left(\Delta\mathbf{x}_{[k]};\mathbf{v}_k',\mathbf{x}_{[k]}\right)\right] \\ \geq&\gamma(1-\Theta)\underbrace{\sum_{k=1}^K\left\{\mathscr{G}_k^{\sigma_2'}\left(\mathbf{0};\mathbf{v}_k',\mathbf{x}_{[k]}\right) - \mathscr{G}_k^{\sigma_2'}\left(\Delta\mathbf{x}_{[k]}^\star;\mathbf{v}_k',\mathbf{x}_{[k]}\right)\right\}}_{C}\end{aligned}$$

Let $\alpha \in [0,1]$ and apply Lemma 5, the previous inequality becomes

$$D_2 \geq \gamma(1-\Theta)(1-\alpha)C + \alpha\frac{\gamma\sigma_1'}{2\tau}\sum_{k=1}^K\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|_2^2 \tag{29}$$

where $\sigma_1' := \frac{(1-\Theta)}{2(1+\Theta)}\sigma'$. From the definition of u_i we know

$$g_i(u_i) = u_i(-\mathbf{A}_i^\top \bar{\mathbf{g}}') - g_i^*(-\mathbf{A}_i^\top \bar{\mathbf{g}}') \tag{30}$$

Replacing $\Delta\mathbf{x}_i = s(u_i - x_i)$ in C gives

$$\begin{aligned}
C \geq& \sum_{k=1}^K \left\{ \sum_{i\in\mathcal{P}_k} (g_i(x_i) - g_i(x_i+\Delta x_i)) - \left\langle \mathbf{g}_k', \mathbf{A}\Delta\mathbf{x}_{[k]} \right\rangle - \frac{\sigma'}{2\tau} \left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|^2 \right\} \\
\geq& \sum_{k=1}^K \left\{ \sum_{i\in\mathcal{P}_k} (sg_i(x_i) - sg_i(u_i) + \frac{\mu_g}{2\tau}(1-s)s(u_i-x_i)^2) - \left\langle \mathbf{g}_k', \mathbf{A}\Delta\mathbf{x}_{[k]} \right\rangle - \frac{\sigma'}{2\tau} \left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|^2 \right\} \\
\overset{(30)}{=}& \sum_{k=1}^K (\sum_{i\in\mathcal{P}_k} \left(sg_i(x_i) + sg_i^*(-\mathbf{A}_i^\top \bar{\mathbf{g}}')\right) + s\langle \mathbf{v}_k/K, \mathbf{g}_k'\rangle) - s\sum_{k=1}^K (\langle \mathbf{v}_k/K, \mathbf{g}_k'\rangle - \langle \mathbf{A}\mathbf{u}_{[k]}, \bar{\mathbf{g}}'\rangle) \\
&+ \sum_{k=1}^K \sum_{i\in\mathcal{P}_k} \left\{ \frac{\mu_g}{2\tau}(1-s)s(u_i-x_i)^2 \right\} - \sum_{k=1}^K \left\langle \mathbf{g}_k', \mathbf{A}\Delta\mathbf{x}_{[k]} \right\rangle - \sum_{k=1}^K \frac{\sigma'}{2\tau} \left\|\mathbf{A}\Delta\mathbf{x}_{[k]}\right\|^2 \\
=& \sum_{k=1}^K \{ \sum_{i\in\mathcal{P}_k} \left(sg_i(x_i) + sg_i^*(-\mathbf{A}_i^\top \bar{\mathbf{g}}')\right) + s\langle \mathbf{v}_k/K, \mathbf{g}_k'\rangle \} + \frac{\mu_g}{2}(1-s)s\left\|\mathbf{u}-\mathbf{x}\right\|^2 \\
&- \sum_{k=1}^K \frac{s^2\sigma'}{2\tau} \left\|\mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\right\|^2 - s\sum_{k=1}^K (\langle \mathbf{v}_k/K, \mathbf{g}_k'\rangle - \langle \mathbf{A}\mathbf{u}_{[k]}, \bar{\mathbf{g}}'\rangle + \langle \mathbf{g}_k', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle) \\
=& sG_{\mathcal{H}}(\mathbf{x}; \{\mathbf{v}_k'\}_{k=1}^K) + \frac{\mu_g}{2}(1-s)s\left\|\mathbf{u}-\mathbf{x}\right\|^2 \\
&- \frac{s^2\sigma'}{2\tau} \sum_{k=1}^K \left\|\mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\right\|^2 - s\sum_{k=1}^K (\langle \mathbf{v}_k'/K, \mathbf{g}_k'\rangle - \langle \mathbf{A}\mathbf{u}_{[k]}, \bar{\mathbf{g}}'\rangle + \langle \mathbf{g}_k', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle)
\end{aligned}$$

We can bound the last term of the previous equation as D_3

$$\begin{aligned}
\frac{1}{s}D_3 =& \sum_{k=1}^K (\langle \mathbf{v}_k'/K, \mathbf{g}_k'\rangle - \langle \mathbf{A}\mathbf{u}_{[k]}, \bar{\mathbf{g}}'\rangle + \langle \mathbf{g}_k', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle) \\
=& \sum_{k=1}^K (\langle \mathbf{g}_k', \mathbf{v}_k'/K\rangle - \langle \mathbf{g}_k', \mathbf{A}\mathbf{u}/K\rangle + \langle \mathbf{g}_k', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle) \\
=& \frac{1}{K}\sum_{k=1}^K \langle \mathbf{g}_k', \mathbf{v}_k' - \mathbf{A}\mathbf{x}\rangle - \sum_{k=1}^K \langle \bar{\mathbf{g}}', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle + \sum_{k=1}^K \langle \mathbf{g}_k', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle \\
=& \frac{1}{K}\sum_{k=1}^K \langle \mathbf{g}_k' - \bar{\mathbf{g}}', \mathbf{v}_k' - \mathbf{A}\mathbf{x}\rangle + \sum_{k=1}^K \langle \mathbf{g}_k' - \bar{\mathbf{g}}', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle
\end{aligned}$$

Bound the gradient terms with consensus violation. First bound $\sum_{k=1}^K \|\mathbf{g}_k' - \bar{\mathbf{g}}'\|_2^2$, define $\mathbf{g}_c := \nabla f(\mathbf{A}\mathbf{x})$

$$\sum_{k=1}^K \|\mathbf{g}_k' - \bar{\mathbf{g}}'\|_2^2 \leq 2\sum_{k=1}^K \left(\|\mathbf{g}_k' - \mathbf{g}_c\|_2^2 + \|\mathbf{g}_c - \bar{\mathbf{g}}'\|_2^2 \right) \leq 2\sum_{k=1}^K \|\mathbf{g}_k' - \mathbf{g}_c\|_2^2 + 2\frac{1}{K}\sum_{k=1}^K \|\mathbf{g}_c - \mathbf{g}_k'\|_2^2$$

Apply the $1/\tau$-smoothness of f we have

$$\sum_{k=1}^K \|\mathbf{g}_k' - \bar{\mathbf{g}}'\|_2^2 \leq \frac{4}{\tau^2}\sum_{k=1}^K \|\mathbf{v}_k' - \mathbf{A}\mathbf{x}\|_2^2 \leq \frac{4\beta^2}{\tau^2}\sum_{k=1}^K \|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|_2^2 \tag{31}$$

Bound the first term in D_3

$$s\frac{1}{K}\sum_{k=1}^K \langle \mathbf{g}_k' - \bar{\mathbf{g}}', \mathbf{v}_k' - \mathbf{A}\mathbf{x}\rangle \leq \frac{s}{2K}\sum_{k=1}^K \left(\tau\|\mathbf{g}_k' - \bar{\mathbf{g}}'\|_2^2 + \frac{1}{\tau}\|\mathbf{v}_k' - \mathbf{A}\mathbf{x}\|_2^2 \right)$$

$$\overset{(31)}{\leq} \frac{5\beta^2 s}{2\tau K} \sum_{k=1}^{K} \|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|_2^2$$

Bound the second term in D_3

$$s\sum_{k=1}^{K}\langle \mathbf{g}'_k - \bar{\mathbf{g}}', \mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\rangle \leq \frac{\tau}{2\sigma'\beta}\sum_{k=1}^{K}\|\mathbf{g}'_k - \bar{\mathbf{g}}'\|_2^2 + \frac{s^2\sigma'\beta}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\right\|_2^2$$

$$\overset{(31)}{\leq} \frac{4\beta}{2\tau\sigma'}\sum_{k=1}^{K}\|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|_2^2 + \frac{s^2\sigma'\beta}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\right\|_2^2$$

Then

$$C \geq sG_{\mathcal{H}}(\mathbf{x}, \{\mathbf{v}'_k\}_{k=1}^K) + \frac{\mu_g}{2}(1-s)s\|\mathbf{u}-\mathbf{x}\|^2 - \frac{s^2(\sigma'+\beta\sigma')}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}(\mathbf{u}-\mathbf{x})_{[k]}\right\|^2$$

$$- \frac{9\beta}{2\tau\sigma'}\sum_{k=1}^{K}\|\mathbf{v}_k - \mathbf{A}\mathbf{x}\|_2^2$$

Then let $\bar{\sigma}' := (1+\beta)\sigma'$ and $\eta := \gamma(1-\Theta)(1-\alpha)$ we have

$$\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} - \alpha\frac{\gamma\sigma'_1}{2\tau}\sum_{k=1}^{K}\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}^{(t)}\right\|_2^2]$$

$$\geq \eta\left(sG_{\mathcal{H}}(\mathbf{x}^{(t)}; \{\mathbf{v}_k'^{(t)}\}_{k=1}^K) - \frac{s^2\bar{\sigma}'}{2\tau}R^{(t)}\right) - \frac{9\eta\beta}{2\tau\sigma'}\sum_{k=1}^{K}\left\|\mathbf{v}_k^{(t)} - \mathbf{A}\mathbf{x}^{(t)}\right\|_2^2$$

□

The following lemma correlates the consensus violation with the size of updates

Lemma 7. *Let $c > 0$ be any constant value. Define $\delta^{(0)} := \mathbf{0}$ and*

$$\delta^{(t+1)} := \beta\delta^{(t)} + cc_1\sum_{k=1}^{K}\left\|\Delta\mathbf{v}_k^{(t)}\right\|_2^2 \tag{32}$$

Then the consensus violation has an upper bound.

$$\sum_{k=1}^{K}\left\|\mathbf{v}_k^{(t)} - \mathbf{v}^{(t)}\right\|_2^2 \leq e_1\delta^{(t)} \tag{33}$$

where $e_1 := (1-\beta)/c$.

Proof. Let

$$a_t := \sum_{k=1}^{K}\left\|\Delta\mathbf{v}_k^{(t)}\right\|_2^2, b_t := \sum_{k=1}^{K}\left\|\mathbf{v}_k^{(t)} - \mathbf{v}^{(t)}\right\|_2^2 \tag{34}$$

We want to prove that

$$b_t \leq e_1\delta^{(t)} \tag{35}$$

First $t = 0$, $b_0 = \delta^{(0)} = 0$. If the claim holds for time $t-1$, then $b_{t-1} \leq e_1\delta^{(t-1)}$. At time t, we have

$$b_t \overset{(14)}{\leq} \beta b_{t-1} + (1-\beta)c_1 a_{t-1} \tag{36}$$

$$\leq \beta\frac{1-\beta}{c}\delta^{(t-1)} + (1-\beta)c_1 a_{t-1} \tag{37}$$

$$\leq \frac{1-\beta}{c}(\beta\delta^{(t-1)} + cc_1 a_{t-1}) \tag{38}$$

$$\overset{(32)}{\leq} e_1\delta^{(t)} \tag{39}$$

Thus we proved the lemma. □

Lemma 8. *Let g_i be strongly convex with convexity parameter $\mu_g \geq 0$ with respect to the norm $\|\cdot\|, \forall\, i \in [n]$. Then for all iterations t of outer loop, and $s \in [0,1]$, it holds that*

$$\begin{aligned}&\mathbb{E}[\mathcal{H}_A(\mathbf{x}^{(t)};\{\mathbf{v}_k^{(t)}\}_{k=1}^K) - \mathcal{H}_A(\mathbf{x}^{(t+1)};\{\mathbf{v}_k^{(t+1)}\}_{k=1}^K) + \frac{1+\beta}{2}\delta^{(t)} - \delta^{(t+1)}] \\ \geq&\eta\left(sG_{\mathcal{H}}(\mathbf{x}^{(t)};\{\textstyle\sum_{i=1}^K \mathcal{W}_{ki}\mathbf{v}_i^{(t)}\}_{k=1}^K) - \frac{s^2\bar{\sigma}'}{2\tau}R^{(t)}\right)\end{aligned} \tag{40}$$

where $\alpha := (1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta})^{-1} \in [0,1]$, $\eta := \gamma(1-\Theta)(1-\alpha)$, $\bar{\sigma}' := (1+\beta)\sigma'$ and

$$R^{(t)} := -\frac{\tau\mu_g(1-s)}{\bar{\sigma}' s}\left\|\mathbf{u}^{(t)} - \mathbf{x}^{(t)}\right\|^2 + \textstyle\sum_{k=1}^K \left\|\mathbf{A}(\mathbf{u}^{(t)} - \mathbf{x}^{(t)})_{[k]}\right\|^2 \tag{41}$$

for $\mathbf{u}^{(t)} \in \mathbb{R}^n$ with $\bar{\mathbf{g}}'^{(t)} := \sum_{k=1}^K \nabla f(\sum_{i=1}^K \mathcal{W}_{ik}\mathbf{v}_i^{(t)})$

$$u_i^{(t)} \in \partial g_i^*(-\mathbf{A}_i^\top \bar{\mathbf{g}}'^{(t)}) \qquad k \text{ s.t. } i \in \mathcal{P}_k. \tag{42}$$

where $\delta^{(t)}$ is defined in Lemma 7.

Proof. In this proof we use $\mathbf{v}'_k := \sum_i \mathcal{W}_{ki}\mathbf{v}_i$. From Lemma 6 we know that

$$\begin{aligned}&\mathbb{E}\left[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} - \alpha\frac{\gamma\sigma_1'}{2\tau}\sum_{k=1}^K\left\|\mathbf{A}\Delta\mathbf{x}_{[k]}^{(t)}\right\|_2^2 + \frac{9\eta\beta}{2\tau\sigma'}\sum_{k=1}^K\left\|\mathbf{v}_k^{(t)} - \mathbf{A}\mathbf{x}^{(t)}\right\|_2^2\right] \\ \geq&\eta\left(sG_{\mathcal{H}}(\mathbf{x}^{(t)};\{\mathbf{v}_k'^{(t)}\}_{k=1}^K) - \frac{s^2\bar{\sigma}'}{2\tau}R^{(t)}\right)\end{aligned}$$

Use the following notations to simplify the calculation

$$a_t := \sum_{k=1}^K\left\|\Delta\mathbf{v}_k^{(t)}\right\|_2^2, b_t := \sum_{k=1}^K\left\|\mathbf{v}_k^{(t)} - \mathbf{v}^{(t)}\right\|_2^2, f_1 := \alpha\frac{\gamma\sigma_1'}{2\tau}, f_2 := \frac{9\eta\beta}{2\tau\sigma'} \tag{43}$$

From Lemma 7 we know that

$$f_2 b_t - f_1 a_t \leq f_2 e_1\delta^{(t)} - f_1(\delta^{(t+1)} - \beta\delta^{(t)})/(cc_1) = (f_2e_1 + \frac{f_1\beta}{cc_1})\delta^{(t)} - \frac{f_1}{cc_1}\delta^{(t+1)} \tag{44}$$

Fix constant c such that $\frac{f_1}{cc_1} = 1$ in (44)

$$c = \frac{f_1}{c_1} = \alpha\frac{(1-\beta)^2\sigma_1'}{2\tau\gamma K^2} \tag{45}$$

Fix $(f_2e_1 + \frac{f_1\beta}{cc_1}) = \frac{1+\beta}{2} < 1$ in (44), to determine $\alpha \in [0,1]$. First consider f_2e_1

$$f_2e_1 = \frac{9\gamma(1-\Theta)(1-\alpha)\beta}{2\tau\sigma'}\frac{1-\beta}{c} \overset{(45)}{=} \frac{1-\alpha}{\alpha}\frac{9(1-\Theta)\beta}{1-\beta}\frac{\gamma K}{\sigma_1'} \tag{46}$$

Then we have

$$f_2e_1 + \frac{f_1\beta}{cc_1} = \frac{1-\alpha}{\alpha}\frac{9(1-\Theta)\beta}{1-\beta}\frac{\gamma K}{\sigma_1'} + \beta = \frac{1+\beta}{2} < 1. \tag{47}$$

Thus we can fix $\alpha \in [0,1]$ to be

$$\alpha := \left(1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta}\right)^{-1} \tag{48}$$

So when have these information.

$$f_2b_t - f_1a_t \leq \frac{1+\beta}{2}\delta^{(t)} - \delta^{(t+1)} \tag{49}$$

Finally, using all of the previous equations we know

$$\mathbb{E}\left[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} + \frac{1+\beta}{2}\delta^{(t)} - \delta^{(t+1)}\right] \geq \eta\left(sG_{\mathcal{H}}(\mathbf{x}^{(t)};\{\mathbf{v}_k'^{(t)}\}_{k=1}^K) - \frac{s^2\bar{\sigma}'}{2\tau}R^{(t)}\right) \tag{50}$$

□

C.3 Main theorems

Here we present the proofs of Theorem 1 and Theorem 2.

Lemma 9. *If g_i^* are L-Lipschitz continuous for all $i \in [n]$, then*

$$\forall\, t : R^{(t)} \leq 4L^2 \sum_{k=1}^{K} \sigma_k n_k = 4L^2 \sigma, \tag{51}$$

where $\sigma_k := \max_{\mathbf{x}_{[k]} \in \mathbb{R}^n} \left\| \mathbf{A}_{[k]} \mathbf{x}_{[k]} \right\|^2 / \left\| \mathbf{x}_{[k]} \right\|^2$.

Proof. For general convex functions, the strong convexity parameter is $\mu_g = 0$, and hence the definition (41) of the complexity constant $R^{(t)}$ becomes

$$R^{(t)} = \sum_{k=1}^{K} \left\| \mathbf{A}(\mathbf{u}^{(t)} - \mathbf{x}^{(t)})_{[k]} \right\|^2 = \sum_{k=1}^{K} \sigma_k \left\| (\mathbf{u}^{(t)} - \mathbf{x}^{(t)})_{[k]} \right\|^2 \leq \sum_{k=1}^{K} \sigma_k |\mathcal{P}_k| 4L^2 = 4L^2 \sigma$$

Here the last inequality follows from L-Lipschitz property of g^*. ☐

Theorem 1 (Strongly Convex g_i)**.** *Consider Algorithm 1 with $\gamma := 1$ and let Θ be the quality of the local solver in Assumption 1. Let g_i be μ_g-strongly convex for all $i \in [n]$ and let f be $1/\tau$-smooth. Let $\bar{\sigma}' := (1+\beta)\sigma'$, $\alpha := (1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta})^{-1}$ and $\eta := \gamma(1-\Theta)(1-\alpha)$*

$$s_0 = \tfrac{\tau \mu_g}{\tau \mu_g + \sigma_{max} \bar{\sigma}'} \in [0, 1]. \tag{8}$$

Then after T iterations of Algorithm 1 with[5]

$$T \geq \tfrac{1 + \eta s_0}{\eta s_0} \log \tfrac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{\mathcal{H}}},$$

it holds that $\mathbb{E}\big[\mathcal{H}_A(\mathbf{x}^{(T)}, \{\mathbf{v}_k^{(T)}\}_{k=1}^K) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}_k^\star\}_{k=1}^K)\big] \leq \varepsilon_{\mathcal{H}}$. Furthermore, after T iterations with

$$T \geq \tfrac{1 + \eta s_0}{\eta s_0} \log \left(\tfrac{1}{\eta s_0} \tfrac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{G_{\mathcal{H}}}}, \right)$$

we have the expected duality gap $\mathbb{E}[G_{\mathcal{H}}(\mathbf{x}^{(T)}, \{\sum_{k=1}^K \mathcal{W}_{kl} \mathbf{v}_l^{(T)}\}_{k=1}^K)] \leq \varepsilon_{G_{\mathcal{H}}}$.

Proof. If $g_i(\cdot)$ are μ_g-strongly convex, one can use the definition of σ_k and $\sigma_{\max}$ to find

$$\begin{aligned} R^{(t)} &\leq - \frac{\tau \mu_g (1-s)}{\bar{\sigma}' s} \left\| \mathbf{u}^{(t)} - \mathbf{x}^{(t)} \right\|^2 + \sum_{k=1}^{K} \left\| \mathbf{A}(\mathbf{u}^{(t)} - \mathbf{x}^{(t)})_{[k]} \right\|^2 \\ &\leq \left(- \frac{\tau \mu_g (1-s)}{\bar{\sigma}' s} + \sigma_{\max} \right) \left\| \mathbf{u}^{(t)} - \mathbf{x}^{(t)} \right\|^2 . \end{aligned} \tag{52}$$

If we set

$$s_0 = \frac{\tau \mu_g}{\tau \mu_g + \sigma_{\max} \bar{\sigma}'} \tag{53}$$

then $R^{(t)} \leq 0$. The duality gap has a lower bound duality gap

$$G_{\mathcal{H}}(\mathbf{x}^{(t)}, \{\mathbf{v}_k'^{(t)}\}_{k=1}^K) \geq \mathcal{H}_A(\mathbf{x}^{(t)}, \{\mathbf{v}_k'^{(t)}\}_{k=1}^K) - \mathcal{H}_A^* \geq \mathcal{H}_A(\mathbf{x}^{(t+1)}, \{\mathbf{v}_k^{(t+1)}\}_{k=1}^K) - \mathcal{H}_A^* \tag{54}$$

and use Lemma 8, we have

$$\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} + \frac{1+\beta}{2} \delta^{(t)} - \delta^{(t+1)}] \geq \eta s_0 G_{\mathcal{H}} \geq \eta s_0 (\mathcal{H}_A^{(t+1)} - \mathcal{H}_A{}^\star) \tag{55}$$

Then

$$\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A{}^\star + \frac{1+\beta}{2} \delta^{(t)}] \geq (1 + \eta s_0) \mathbb{E}[\mathcal{H}_A^{(t+1)} - \mathcal{H}_A{}^\star + \delta^{(t+1)}] \tag{56}$$

[5] $\varepsilon_{\mathcal{H}}^{(0)} := \mathcal{H}_A(\mathbf{x}^{(0)}, \{\mathbf{v}_k^{(0)}\}_{k=1}^K) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}_k^\star\}_{k=1}^K)$ is the initial suboptimality.

Therefore if we denote $\varepsilon_{\mathcal{H}}^{(t)} := \mathcal{H}_A^{(t)} - \mathcal{H}_A^\star + \delta^{(t)}$ we have recursively that

$$\mathbb{E}[\varepsilon_{\mathcal{H}}^{(t)}] \le \left(1 - \frac{\eta s_0}{1+\eta s_0}\right)^t \varepsilon_{\mathcal{H}}^{(0)} \le \exp\left(-\frac{\eta s_0}{1+\eta s_0} t\right) \varepsilon_{\mathcal{H}}^{(0)}$$

The right hand side will be smaller than some $\varepsilon_{\mathcal{H}}$ if

$$T \ge \frac{1+\eta s_0}{\eta s_0} \log \frac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{\mathcal{H}}}$$

Moreover, to bound the duality gap $G_{\mathcal{H}}^{(t)}$, we have

$$\eta s_0 G_{\mathcal{H}}^{(t)} \overset{(55)}{\le} \mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} + \frac{1+\beta}{2}\delta^{(t)} - \delta^{(t+1)}] \le \mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A{}^* + \delta^{(t)}]$$

Hence if $\varepsilon_{\mathcal{H}} \le \eta s_0 \varepsilon_{G_{\mathcal{H}}}$ then $G_{\mathcal{H}}^{(t)} \le \varepsilon_{G_{\mathcal{H}}}$. Therefore after

$$T \ge \frac{1+\eta s_0}{\eta s_0} \log\left(\frac{1}{\eta s_0}\frac{\varepsilon_{\mathcal{H}}^{(0)}}{\varepsilon_{G_{\mathcal{H}}}}\right)$$

iterations we have obtained a duality gap less then $\varepsilon_{G_{\mathcal{H}}}$.

□

Theorem 2 (Non-strongly Convex Case)**.** *Consider Algorithm 1, using a local solver of quality* Θ*. Let* $g_i(\cdot)$ *have L-bounded support, and let* f *be* $(1/\tau)$*-smooth. Let* $\varepsilon_{G_{\mathcal{H}}} > 0$ *be the desired duality gap. Then after* T *iterations where*

$$T \ge T_0 + \max\left\{ \left\lceil \tfrac{1}{\eta} \right\rceil, \tfrac{4L^2\sigma\bar{\sigma}'}{\tau \varepsilon_{G_{\mathcal{H}}} \eta} \right\}, \qquad T_0 \ge t_0 + \left[\tfrac{2}{\eta}\left(\tfrac{8L^2\sigma\bar{\sigma}'}{\tau\varepsilon_{G_{\mathcal{H}}}} - 1\right)\right]_+$$
$$t_0 \ge \max\left\{0, \left\lceil \tfrac{1+\eta}{\eta} \log \tfrac{2\tau(\mathcal{H}_A(\mathbf{x}^{(0)}, \{\mathbf{v}_l^{(0)}\}) - \mathcal{H}_A(\mathbf{x}^\star, \{\mathbf{v}^\star\}))}{4L^2\sigma\bar{\sigma}'} \right\rceil \right\}$$

and $\bar{\sigma}' := (1+\beta)\sigma'$, $\alpha := (1 + \frac{(1-\beta)^2}{36(1+\Theta)\beta})^{-1}$ *and* $\eta := \gamma(1-\Theta)(1-\alpha)$*. We have that the expected duality gap satisfies*

$$\mathbb{E}\big[G_{\mathcal{H}}(\bar{\mathbf{x}}, \{\bar{\mathbf{v}}_k\}_{k=1}^K, \{\bar{\mathbf{w}}_k\}_{k=1}^K)\big] \le \varepsilon_{G_{\mathcal{H}}}$$

at the averaged iterate $\bar{\mathbf{x}} := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} \mathbf{x}^{(t)}$*, and* $\mathbf{v}'_k := \sum_{l=1}^K \mathcal{W}_{kl}\mathbf{v}_l$ *and* $\bar{\mathbf{v}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} (\mathbf{v}'_k)^{(t)}$ *and* $\bar{\mathbf{w}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} \nabla f((\mathbf{v}'_k)^{(t)})$*.*

Proof. We write $\mathcal{H}_A^{(t)}$ instead of $\mathcal{H}_A(\mathbf{x}^{(t)}; \{\mathbf{v}_k^{(t)}\}_{k=1}^K)$ and $\mathcal{H}_A^\star$ instead of $\mathcal{H}_A(\mathbf{x}^\star; \{\mathbf{v}_k{}^\star\}_{k=1}^K)$. We begin by estimating the expected change of feasibility for $\mathcal{H}_A$. We can bound this above by using Lemma 8 and the fact that $F_B(\cdot)$ is always a lower bound for $-F_A(\cdot)$ and then applying (51) to find

$$(1+\eta s)\mathbb{E}[\mathcal{H}_A^{(t+1)} - \mathcal{H}_A^\star + \delta^{(t+1)}] \le (\mathcal{H}_A^{(t)} - \mathcal{H}_A^\star + \delta^{(t)}) + \eta\frac{\bar{\sigma}' s^2}{2\tau}4L^2\sigma \tag{57}$$

Use (57) recursively we have

$$\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^\star + \delta^{(t)}] \le (1+\eta s)^{-t}(\mathcal{H}_A^{(0)} - \mathcal{H}_A^\star + \delta^{(0)}) + s\frac{4L^2\bar{\sigma}'\sigma}{2\tau} \tag{58}$$

We know that $\delta^{(0)} = 0$. Choose $s = 1$ and

$$t = t_0 := \max\left\{0, \left\lceil \frac{1+\eta}{\eta} \log \frac{2\tau(\mathcal{H}_A^{(0)} - \mathcal{H}_A^\star)}{4L^2\sigma\bar{\sigma}'} \right\rceil \right\} \tag{59}$$

leads to

$$\mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^\star + \delta^{(t)}] \le \frac{4L^2\bar{\sigma}'\sigma}{\tau} \tag{60}$$

Next, we show inductively that

$$\forall\, t \geq t_0 : \mathbb{E}[\mathcal{H}_A^{(t)} - \mathcal{H}_A^\star + \delta^{(t)}] \leq \frac{4L^2\bar{\sigma}'\sigma}{\tau(1+\frac{1}{2}\eta(t-t_0))}. \tag{61}$$

Clearly, (60) implies that (61) holds for $t = t_0$. Assuming that it holds for any $t \geq t_0$, we show that it must also hold for $t+1$. Indeed, using

$$s = \frac{1}{1+\frac{1}{2}\eta(t-t_0)} \in [0,1], \tag{62}$$

we obtain

$$\mathbb{E}[\mathcal{H}_A^{(t+1)} - \mathcal{H}_A^\star + \delta^{(t+1)}] \leq \frac{4L^2\sigma\bar{\sigma}'}{\tau} \underbrace{\left(\frac{1+\frac{1}{2}\eta(t-t_0) - \frac{1}{2}\gamma(1-\Theta)}{(1+\frac{1}{2}\eta(t-t_0))^2}\right)}_{D}$$

by applying the bounds (57) and (61), plugging in the definition of s (62), and simplifying. We upper bound the term D using the fact that geometric mean is less or equal to arithmetic mean:

$$\begin{aligned} D =& \frac{1}{1+\frac{1}{2}\eta(t+1-t_0)} \underbrace{\frac{(1+\frac{1}{2}\eta(t+1-t_0))(1+\frac{1}{2}\eta(t-1-t_0))}{(1+\frac{1}{2}\eta(t-t_0))^2}}_{\leq 1} \\ \leq& \frac{1}{1+\frac{1}{2}\eta(t+1-t_0)}. \end{aligned}$$

We can apply the results of Lemma 8 to get

$$\eta s G_{\mathcal{H}}(\mathbf{x}^{(t)}, \{\mathbf{v}_k^{(t)}\}_{k=1}^K) \leq \mathcal{H}_A^{(t)} - \mathcal{H}_A^{(t+1)} + \delta^{(t)} - \delta^{(t+1)}$$

Define the following iterate

$$\bar{\mathbf{x}} := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} \mathbf{x}^{(t)}, \bar{\mathbf{v}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} \mathbf{v}_k^{(t)}, \bar{\mathbf{w}}_k := \frac{1}{T-T_0}\sum_{t=T_0+1}^{T-1} \nabla f(\mathbf{v}_k^{(t)})$$

use Lemma 9 to obtain

$$\begin{aligned} \mathbb{E}[G_{\mathcal{H}}(\bar{\mathbf{x}}, \{\bar{\mathbf{v}}_k\}_{k=1}^K, \{\bar{\mathbf{w}}_k\}_{k=1}^K)] \leq& \frac{1}{T-T_0}\sum_{t=T_0}^{T-1} \mathbb{E}[G_{\mathcal{H}}\left(\mathbf{x}^{(t)}, \{\mathbf{v}_k^{(t)}\}_{k=1}^K\right)] \\ \leq& \frac{1}{\eta s}\frac{1}{T-T_0}\mathbb{E}[\mathcal{H}_A^{(T_0)} - \mathcal{H}_A^\star + \delta^{(T_0)}] + \frac{4L^2\sigma\bar{\sigma}' s}{2\tau} \end{aligned}$$

If $T \geq \lceil \frac{1}{\eta} \rceil + T_0$ such that $T_0 \geq t_0$ we have

$$\begin{aligned} \mathbb{E}[G_{\mathcal{H}}(\bar{\mathbf{x}}, \{\bar{\mathbf{v}}_k\}_{k=1}^K, \{\bar{\mathbf{w}}_k\}_{k=1}^K)] \leq& \frac{1}{\eta s}\frac{1}{T-T_0}\left(\frac{4L^2\bar{\sigma}'\sigma}{\tau(1+\frac{1}{2}\eta(T_0-t_0))}\right) + \frac{4L^2\sigma\bar{\sigma}' s}{2\tau} \\ =& \frac{4L^2\sigma\bar{\sigma}'}{\tau}\left(\frac{1}{\eta s}\frac{1}{T-T_0}\frac{1}{(1+\frac{1}{2}\eta(T_0-t_0))} + \frac{s}{2}\right). \end{aligned}$$

Choosing

$$s = \frac{1}{(T-T_0)\eta} \in [0,1] \tag{63}$$

gives us

$$\mathbb{E}[G_{\mathcal{H}}(\bar{\mathbf{x}}, \{\bar{\mathbf{v}}_k\}_{k=1}^K, \{\bar{\mathbf{w}}_k\}_{k=1}^K)] \leq \frac{4L^2\sigma\bar{\sigma}'}{\tau}\left(\frac{1}{1+\frac{1}{2}\eta(T_0-t_0)} + \frac{1}{2}\frac{1}{(T-T_0)\eta}\right). \tag{64}$$

To have right hand side of (64) smaller then $\varepsilon_{G_{\mathcal{H}}}$ it is sufficient to choose T_0 and T such that

$$\frac{4L^2\sigma\bar{\sigma}'}{\tau}\left(\frac{1}{1+\frac{1}{2}\eta(T_0-t_0)}\right) \leq \frac{1}{2}\varepsilon_{G_{\mathcal{H}}} \tag{65}$$

$$\frac{4L^2\sigma\bar{\sigma}'}{\tau}\left(\frac{1}{2}\frac{1}{(T-T_0)\eta}\right)\leq \frac{1}{2}\varepsilon_{G_{\mathcal{H}}} \tag{66}$$

Hence if $T_0 \geq t_0 + \frac{2}{\eta}\left(\frac{8L^2\sigma\bar{\sigma}'}{\tau\varepsilon_{G_{\mathcal{H}}}} - 1\right)$ and $T \geq T_0 + \frac{4L^2\sigma\bar{\sigma}'}{\tau\varepsilon_{G_{\mathcal{H}}}\eta}$ then (65) and (66) are satisfied.

□

Proposition 1 (Local Certificates)**.** *Assume g_i has L-bounded support, and let $\mathcal{N}_k := \{j : \mathcal{W}_{jk} > 0\}$ be the set of nodes accessible to node k. Then for any given $\varepsilon > 0$, we have*

$$G_{\mathcal{H}}(\mathbf{x}; \{\mathbf{v}_k\}_{k=1}^K) \leq \varepsilon,$$

if for all $k = 1, \ldots, K$ the following two local conditions are satisfied:

$$\mathbf{v}_k^\top \nabla f(\mathbf{v}_k) + \sum_{i\in\mathcal{P}_k}\left(g_i(\mathbf{x}_i) + g_i^*(-\mathbf{A}_i^\top \nabla f(\mathbf{v}_k))\right) \leq \frac{\varepsilon}{2K} \tag{9}$$

$$\left\|\nabla f(\mathbf{v}_k) - \tfrac{1}{|\mathcal{N}_k|}\textstyle\sum_{j\in\mathcal{N}_k}\nabla f(\mathbf{v}_j)\right\|_2 \leq \left(\textstyle\sum_{k=1}^K n_k^2\sigma_k\right)^{-1/2}\tfrac{1-\beta}{2L\sqrt{K}}\varepsilon, \tag{10}$$

Proof. If the $\mathbf{w}_k$ variable in the duality gap (6) is fixed to $\mathbf{w}_k = \mathbf{g}_k := \nabla f(\mathbf{v}_k)$, then using the equality condition of the Fenchel-Young inequality on f, the duality gap can be written as follows

$$G_{\mathcal{H}}(\mathbf{x}; \{\mathbf{v}_k\}_{k=1}^K) := \sum_{k=1}^K\left(\langle\mathbf{v}_k, \mathbf{g}_k\rangle + \sum_{i\in\mathcal{P}_k} g_i(\mathbf{x}_i) + g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}})\right) \tag{67}$$

where $\bar{\mathbf{g}} = \frac{1}{K}\sum_{k=1}^K \mathbf{g}_k$ is the only term locally unavailable.

$$G_{\mathcal{H}} \leq \sum_{k=1}^K\left(\langle\mathbf{v}_k, \mathbf{g}_k\rangle + \sum_{i\in\mathcal{P}_k} g_i(\mathbf{x}_i) + g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)\right) + \left|\sum_{k=1}^K\sum_{i\in\mathcal{P}_k}\left(g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}}) - g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)\right)\right| \tag{68}$$

If both terms in (68) are less than $\varepsilon/2$, then $G_{\mathcal{H}} \leq \varepsilon$. Since the first term can be calculated locally, we only need for all $k = 1, \ldots K$

$$\langle\mathbf{v}_k, \mathbf{g}_k\rangle + \sum_{i\in\mathcal{P}_k} g_i(\mathbf{x}_i) + g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k) \leq \frac{\varepsilon}{2K}. \tag{69}$$

Consider the second term in (68). Compute the difference between $g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}})$ and $g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)$

$$|g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}}) - g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)| \leq L| - \mathbf{A}_i^\top(\bar{\mathbf{g}} - \mathbf{g}_k)| \leq L\left\|\mathbf{A}_i\right\|_2\left\|\bar{\mathbf{g}} - \mathbf{g}_k\right\|_2 \tag{70}$$

where we use Lemma 3 and L-Lipschitz continuity. Then sum up coordinates $i \in \mathcal{P}_k$ on node k

$$\left|\sum_{i\in\mathcal{P}_k}\left(g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}}) - g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)\right)\right| \leq L\left\|\bar{\mathbf{g}} - \mathbf{g}_k\right\|_2\sum_{i\in\mathcal{P}_k}\left\|\mathbf{A}_i\right\|_2. \tag{71}$$

Sum up (71) for all $k = 1, \ldots, K$ and apply the Cauchy-Schwarz inequality

$$\left|\sum_{k=1}^K\sum_{i\in\mathcal{P}_k}\left(g_i^*(-\mathbf{A}_i^\top\bar{\mathbf{g}}) - g_i^*(-\mathbf{A}_i^\top\mathbf{g}_k)\right)\right| \leq L\sqrt{\sum_{k=1}^K\left\|\bar{\mathbf{g}} - \mathbf{g}_k\right\|_2^2}\sqrt{\sum_{k=1}^K(\sum_{i\in\mathcal{P}_k}\left\|\mathbf{A}_i\right\|_2)^2}. \tag{72}$$

We will upper bound $\sum_{i\in\mathcal{P}_k}\left\|\mathbf{A}_i\right\|_2$ and $\left\|\bar{\mathbf{g}} - \mathbf{g}_k\right\|_2$ separately. First we have

$$\sum_{i\in\mathcal{P}_k}\left\|\mathbf{A}_i\right\|_2 \leq \sqrt{n_k}\left\|\mathbf{A}_{[k]}\right\|_F \leq n_k\left\|\mathbf{A}_{[k]}\right\|_{\infty,2} \leq n_k\sqrt{\sigma_k}, \tag{73}$$

where we write $\|.\|_{\infty,2}$ for the largest Euclidean norm of a column of the argument matrix, and then used the definition of σ_k as in (7). Let us write $\mathbf{G} := [\mathbf{g}_1; \cdots; \mathbf{g}_K]$, $\mathbf{E} := \frac{1}{K}[\mathbf{1}; \cdots; \mathbf{1}]$, then apply Young's inequality with δ

$$\textstyle\sum_{k=1}^K\left\|\mathbf{g}_k - \bar{\mathbf{g}}\right\|_2^2 = \left\|\mathbf{G} - \mathbf{G}\mathbf{E}\right\|_F^2$$

$$
\begin{aligned}
&\leq (1+\frac{1}{\delta}) \left\|\mathbf{G}-\mathbf{G}\mathcal{W}\right\|_F^2 + (1+\delta)\left\|\mathbf{G}\mathcal{W}-\mathbf{G}\mathbf{E}\right\|_F^2 \\
&= (1+\frac{1}{\delta}) \left\|\mathbf{G}-\mathbf{G}\mathcal{W}\right\|_F^2 + (1+\delta)\left\|\mathbf{G}(\mathbf{I}-\mathbf{E})(\mathcal{W}-\mathbf{E})\right\|_F^2 \\
&\leq (1+\frac{1}{\delta}) \left\|\mathbf{G}-\mathbf{G}\mathcal{W}\right\|_F^2 + (1+\delta)\beta^2\left\|\mathbf{G}(\mathbf{I}-\mathbf{E})\right\|_F^2 \\
&= (1+\tfrac{1}{\delta}) \textstyle\sum_{k=1}^K \left\|\mathbf{g}_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k}\mathbf{g}_j\right\|_2^2 + (1+\delta)\beta^2 \sum_{k=1}^K \left\|\mathbf{g}_k-\bar{\mathbf{g}}\right\|_2^2
\end{aligned}
$$

Take $\delta := (1-\beta)/\beta$, then we have

$$
\begin{aligned}
\sum_{k=1}^K \left\|\mathbf{g}_k-\bar{\mathbf{g}}\right\|_2^2 \leq& \frac{1}{1-\beta}\sum_{k=1}^K \left\|\mathbf{g}_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k}\mathbf{g}_j\right\|_2^2 + \beta\sum_{k=1}^K \left\|\mathbf{g}_k-\bar{\mathbf{g}}\right\|_2^2 \\
\leq& \frac{1}{(1-\beta)^2}\sum_{k=1}^K \left\|\mathbf{g}_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k}\mathbf{g}_j\right\|_2^2 \qquad (74)
\end{aligned}
$$

We now use (73) and (74) and impose

$$
\frac{1}{(1-\beta)^2}\sum_{k=1}^K \left\|\mathbf{g}_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k}\mathbf{g}_j\right\|_2^2 \sum_{k=1}^K n_k^2\sigma_k \leq \left(\frac{\varepsilon}{2L}\right)^2 \qquad (75)
$$

then (72) is less than $\varepsilon/2$. Finally, (75) can be guaranteed by imposing the following restrictions for all $k=1,\ldots,K$

$$
\left\|\mathbf{g}_k - \frac{1}{|\mathcal{N}_k|}\sum_{j\in\mathcal{N}_k}\mathbf{g}_j\right\|_2^2 \leq (1-\beta)^2\frac{\varepsilon^2}{4KL^2}\left(\sum_{k=1}^K n_k^2\sigma_k\right)^{-1} \qquad (76)
$$

□

D Experiment details

In this section we provide greater details about the experimental setup and implementations. All the codes are written in PyTorch (0.4.0a0+cc9d3b2) with MPI backend [Paszke et al., 2017]. In each experiment, we run centralized CoCoA for a sufficiently long time until progress stalled; then use their minimal value as the approximate optima.

DIGing. DIGing is a distributed algorithm based on inexact gradient and a gradient tracking technique. [Nedic et al., 2017] proves linear convergence of DIGing when the distributed optimization objective is strongly convex over time-varying graphs with a fixed learning rate. In this experiments, we only consider the time-invariant graph. The stepsize is chosen via a grid search. [Nedic et al., 2017] mentioned that the EXTRA algorithm [Shi et al., 2015] is almost identical to that of the DIGing algorithm when the same stepsize is chosen for both algorithms, so we only present with DIGing here.

CoLa. We implement CoLa framework with local solvers from Scikit-Learn [Pedregosa et al., 2011]. Their ElasticNet solver uses coordinate descent internally. We note that since the framework and theory allow any internal solver to be used, CoLa could benefit even beyond the results shown by using existing fast solvers. We implement CoCoA as a special case of CoLa. The aggregation parameter γ is fixed to 1 for all experiments.

ADMM. Alternating Direction Method of Multipliers (ADMM) [Boyd et al., 2011] is a classical approach in distributed optimization problems. Applying ADMM to decentralized settings [Shi et al., 2014] involves solving

$$
\min_{x_i, z_{ij}} \textstyle\sum_{i=1}^L f_i(x_i) \qquad \text{s.t. } x_i = z_{ij}, x_j = z_{ij}, \qquad \forall\,(i,j)\in\mathcal{E}
$$

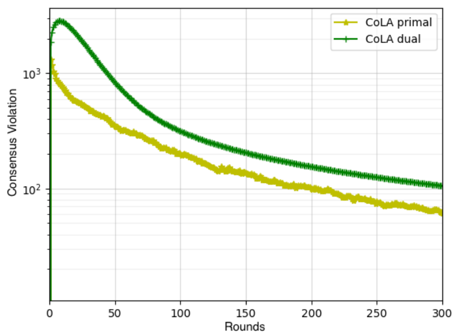

Figure 5: The consensus violation curve of CoLA in Figure 2.

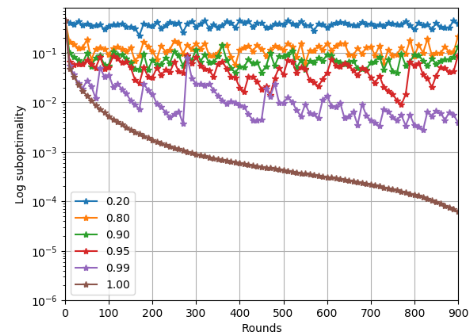

Figure 6: Same settings as Figure 4 except that $\mathbf{x}_{[k]}$ are reset after node k leaving the network.

where z_{ij} is an auxiliary variable imposing the consensus constraint on neighbors i and j. We therefore employ the coordinate descent algorithm to solve the local problem. The number of coordinates chosen in each round is the same as that of CoLA. We choose the penalty parameter from the conclusion of [Shi et al., 2014].

Additional experiments. We provide additional experimental results here. First the *consensus violation* $\sum_{k=1}^{K} \|\mathbf{v}_k - \mathbf{v}_c\|_2^2$ curve for Figure 2 is displayed in Figure 5. As we can see, the consensus violation starts with 0 and soon becomes very large, then gradually drops down. This is because we are minimizing the sum of $\mathcal{H}_A^{(t)}$ and $\delta^{(t)}$, see the proof of Theorem 1. Then another model under failing nodes is tested in Figure 6 where $\mathbf{x}_{[k]}$ are initialized to 0 when node k leave the network. Note that we assume the leaving node k will inform its neighborhood and modify their own local estimates so that the rest nodes still satisfy $\frac{1}{\text{\#nodes}} \sum_k \mathbf{v}_k = \mathbf{v}_c$. This failure model, however, oscillates and does not converge fast.

E Details regarding extensions

E.1 Fault tolerance and time varying graphs

In this section we extend framework CoLA to handle fault tolerance and time varying graphs. Here we assume when a node leave the network, their local variables $\mathbf{x}$ are frozen. We use same assumptions about the fault tolerance model in [Smith et al., 2017].

Definition 5 (Per-Node-Per-Iteration-Approximation Parameter)**.** *At each iteration t, we define the accuracy level of the solution calculated by node k to its subproblem as*

$$\theta_k^t := \frac{\mathcal{G}_k^{\sigma'}(\Delta\mathbf{x}_k^{(t)}; \mathbf{x}_{[k]}^{(t)}, \mathbf{v}_k^{(t)}) - \mathcal{G}_k^{\sigma'}(\Delta\mathbf{x}_k^{\star}; \mathbf{x}_{[k]}^{(t)}, \mathbf{v}_k^{(t)})}{\mathcal{G}_k^{\sigma'}(\mathbf{0}; \mathbf{x}_{[k]}^{(t)}, \mathbf{v}_k^{(t)}) - \mathcal{G}_k^{\sigma'}(\Delta\mathbf{x}_k^{\star}; \mathbf{x}_{[k]}^{(t)}, \mathbf{v}_k^{(t)})} \tag{77}$$

where $\Delta\mathbf{x}_k^{\star}$ is the minimizer of the subproblem $\mathcal{G}_k^{\sigma'}(\cdot; \mathbf{x}_{[k]}^{(t)}, \mathbf{v}_k^{(t)})$. We allow this value to vary between [0, 1] with $\theta_k^t := 1$ meaning that no updates to subproblem $\mathcal{G}_k^{\sigma'}$ are made by node k at iteration t.

The flexible choice of θ_k^t allows the consideration of stragglers and fault tolerance. We also need the following assumption on θ_k^t.

Assumption 2 (Fault Tolerance Model)**.** *Let $\{\mathbf{x}^{(t)}\}_{t=0}^{T}$ be the history of iterates until the beginning of iteration T. For all nodes k and all iterations t, we assume $p_k^t := \mathcal{P}[\theta_k^t = 1] \leq p_{max} < 1$ and $\hat{\Theta}_k^T := \mathbb{E}[\theta_k^T | \{\mathbf{x}^{(t)}\}_{t=0}^{T}, \theta_k^T < 1] \leq \Theta_{max} < 1$.*

In addition we write $\bar{\Theta} := p_{\max} + (1 - p_{\max})\Theta_{\max} < 1$. Another assumption on time varying model is necessary in order to maintain the same linear and sublinear convergence rate. It is from [Nedic et al., 2017, Assumption 1]:

Assumption 3 (Time Varying Model)**.** *Assume the mixing matrix $\mathcal{W}(t)$ is a function of time t. There exist a positive integer B such that the spectral gap satisfies the following condition*

$$\sigma_{max}\left\{\prod_{i=t}^{t+B-1}\mathcal{W}(i)-\tfrac{1}{K}\mathbf{1}\mathbf{1}^\top\right\}\leq\beta_{max}\qquad\forall\, t\geq 0.$$

We change the Algorithm 1 such that it performs gossip step for B times between solving subproblems. In this way, the convergence rate on time varying mixing matrix is similar to a static graph with mixing matrix $\prod_{i=t}^{t+B-1}\mathcal{W}(i)$. The sublinear/linear rate can be proved similarly.

E.2 Data dependent aggregation parameter

Definition 6 (Data-dependent aggregation parameter)**.** *In Algorithm 1, the aggregation parameter γ controls the level of adding γ versus averaging $\gamma:=\frac{1}{K}$ of the partial solution from all machines. For the convergence discussed below to hold, the subproblem parameter σ' must be chosen not smaller than*

$$\sigma'\geq\sigma'_{min}:=\gamma\max_{\mathbf{x}\in\mathbb{R}^n}\frac{\|\mathbf{A}\mathbf{x}\|^2}{\sum_{k=1}^K\left\|\mathbf{A}\mathbf{x}_{[k]}\right\|^2}\tag{78}$$

The simple choice of $\sigma':=\gamma K$ is valid for (78), closer to the actual bound given in σ'_{min}.

E.3 Hessian subproblem

If the Hessian matrix of f is available, it can be used to define better local subproblems, as done in the classical distributed setting by [Gargiani, 2017, Lee and Chang, 2017, Dünner et al., 2018, Lee et al., 2018]. We use same idea in the decentralized setting, defining the improved subproblem

$$\begin{aligned}\mathscr{G}_k^{\sigma'}(\Delta\mathbf{x};\mathbf{x}_{[k]},\mathbf{v}_k):=&\tfrac{1}{K}f(\mathbf{v}_k)+\left\langle\textstyle\sum_{l=1}^K\mathcal{W}_{kl}\nabla f(\mathbf{v}_l),\mathbf{A}\Delta\mathbf{x}_{[k]}\right\rangle\\&+\tfrac{1}{2}(\mathbf{A}\Delta\mathbf{x}_{[k]})^\top\left(\textstyle\sum_{l=1}^K\mathcal{W}_{kl}\nabla^2 f(\mathbf{v}_l)\right)\mathbf{A}\Delta\mathbf{x}_{[k]}+\textstyle\sum_{i\in\mathcal{P}_k}g_i(x_i+\Delta x_i)\end{aligned}\tag{79}$$

The sum of previous subproblems satisfies the following relations

$$\begin{aligned}&\sum_{k=1}^K\mathscr{G}_k^{\sigma'}(\mathbf{0};\mathbf{x}_{[k]}^{(t+1)},\mathbf{v}_k^{(t+1)})\\=&\frac{1}{K}\sum_{k=1}^K f\left(\sum_{l=1}^K\mathcal{W}_{kl}\mathbf{v}_l^{(t)}+\gamma K\mathbf{A}\Delta\mathbf{x}_{[k]}\right)+\sum_{i\in\mathcal{P}_k}g_i(x_i^{(t)}+(\Delta\mathbf{x}_{[k]})_i)\\\leq&\frac{1}{K}\sum_{k=1}^K\sum_{l=1}^K\mathcal{W}_{kl}\left\{f(\mathbf{v}_l^{(t)})+\langle\nabla f(\mathbf{v}_l^{(t)}),\gamma K\mathbf{A}\Delta\mathbf{x}_{[k]}\rangle+\frac{1}{2}(\mathbf{A}\Delta\mathbf{x}_{[k]})^\top\nabla^2 f(\mathbf{v}_l^{(t)})\mathbf{A}\Delta\mathbf{x}_{[k]}\right\}\\&+\sum_{i\in\mathcal{P}_k}g_i(x_i^{(t)}+(\Delta\mathbf{x}_{[k]})_i)\\=&\sum_{k=1}^K\mathscr{G}_k^{\sigma'}(\Delta\mathbf{x};\mathbf{x}_{[k]}^{(t)},\mathbf{v}_k^{(t)})\leq\sum_{k=1}^K\mathscr{G}_k^{\sigma'}(\mathbf{0};\mathbf{x}_{[k]}^{(t)},\mathbf{v}_k^{(t)})\end{aligned}$$

This means that the sequence $\left\{\sum_{k=1}^K\mathscr{G}_k^{\sigma'}(\mathbf{0};\mathbf{x}_{[k]}^{(t)},\mathbf{v}_k^{(t)})\right\}_{t=0}^\infty$ is monotonically non-increasing. Following the reasoning in this paper, we can have similar convergence guarantees for both strongly convex and general convex problems. Formalizing all detailed implications here would be out of the scope of this paper, but the main point is that the second-order techniques developed for the CoCoA framework also have their analogon in the decentralized setting.