Acceleration in Distributed Optimization under Similarity

Ye Tian[*] **Gesualdo Scutari**[*] **Tianyu Cao**[*] **Alexander Gasnikov**[†,⋄]
[*]Purdue University [†] Moscow Institute of Physics and Technology
[⋄] ISP RAS Research Center for Trusted Artificial Intelligence

Abstract

We study distributed (strongly convex) optimization problems over a network of agents, with no centralized nodes. The loss functions of the agents are assumed to be *similar*, due to statistical data similarity or otherwise. In order to reduce the number of communications to reach a solution accuracy, we proposed a *preconditioned, accelerated* distributed method. An ε-solution is achieved in $\tilde{\mathcal{O}}\big(\sqrt{\frac{\beta/\mu}{1-\rho}}\log 1/\varepsilon\big)$ number of communications steps, where β/μ is the relative condition number between the global and local loss functions, and ρ characterizes the connectivity of the network. This rate matches (up to poly-log factors) lower complexity communication bounds of distributed gossip-algorithms applied to the class of problems of interest. Numerical results show significant communication savings with respect to existing accelerated distributed schemes, especially when solving ill-conditioned problems.

1 INTRODUCTION

We study distributed optimization over a network of m agents, in the form

$$\min_{x\in\mathbb{R}^d} \; u(x) \triangleq f(x) + r(x), \quad f(x) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_i(x), \tag{P}$$

where $f_i : \mathbb{R}^d \to \mathbb{R}$ is the loss function of agent i, known only to that agent; and $r : \mathbb{R}^d \to [-\infty, \infty]$ is an extended-value function (known to all agents), which is instrumental to enforce further conditions on the solution, such as sparsity or constraints. The network of agents is modelled as undirected, fixed graph, with no centralized node; we refer to such architectures as *mesh* networks.

An instance of (P) of particular interest to this work is the distributed Empirical Risk Minimization (ERM) whereby the goal is to minimize the average loss over some dataset, distributed across the nodes of the network. Specifically, denoting by $\mathcal{D}_i = \{z_i^{(1)}, \ldots, z_i^{(n)}\}$ the set of n samples owned by agent i, the empirical risk f_i in (P) reads

$$f_i(x) = \frac{1}{n}\sum_{j=1}^{n} \ell\big(x; z_i^j\big), \tag{1}$$

where $\ell(x; z_i^j)$ measures the mismatch between the parameter x and the sample z_i^j.

The lack of global knowledge of f from the agents and of a centralized node in the network calls for the design of distributed algorithms, whereby agents alternate a computation procedure based on local information and communication round(s) with neighboring nodes. Since the cost of communications is often the bottleneck in distributed computing, if compared with local (parallel) computations (e.g., Bekkerman et al. (2011); Lian et al. (2017)), a lot of research has been devoted to designing distributed algorithms that are *communication efficient.*

Communication-saving via acceleration: Acceleration (in the sense of Nesterov) has been extensively investigated as a procedure to reduce the communication burden of distributed algorithms– Table 1 summarizes the communication complexity of existing first-order methods over mesh networks (see Sec. 1.2 for a discussion of these works). For L-smooth and μ-strongly convex functions f in (P), linear convergence rate is certified, with a constant factor scaling with $\sqrt{\kappa}$ ($\kappa \triangleq L/\mu$ is the condition number of f). This dependence is not improvable, meaning that it matches lower communication complexity bounds for the class of L-smooth and μ-strongly convex function f_i's (Scaman et al., 2017). However, for ill-conditioned problems–e.g.,

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

the typical setting of many ERM problems wherein the optimal regularization parameter for test predictive performance is very small–κ can be extremely large; hence the aforementioned scaling of the number of communications with $\sqrt{\kappa}$ is unsatisfactory.

Exploiting function similarity: Further improvements can be obtained if extra structure is postulated for the f_i's. This is, e.g., the case of ERM problems wherein each f_i [see (1)] has an additional finite-sum structure. This is an instance of the property known as *function similarity* (see, e.g., Shamir et al. (2014); Arjevani and Shamir (2015); Zhang and Lin (2015); Hendrikx et al. (2020b)):

$$\left\|\nabla^2 f_i(x) - \nabla^2 f(x)\right\| \le \beta, \tag{2}$$

for all x in a proper domain of interest and all $i = 1, \ldots, m$, where $\beta > 0$ measures the degree of similarity between the Hessian matrices of local and global losses. For instance, in the aforementioned ERM setting, when data are i.i.d. among machines, the f_i's in (1) reflect statistical similarities in the data residing at different nodes, resulting in $\beta = \tilde{O}(1/\sqrt{n})$ w.h.p., where n is the local sample size ($\tilde{O}$ hides log-factors and dependence on d).

The situation $1 + \beta/\mu \ll \kappa$, happens in several scenarios. For instance, consider some ill-conditioned functions. Another example are ERMs with optimal regularization $\mu = \mathcal{O}(1/\sqrt{mn})$ and $L = \mathcal{O}(1)$ (e.g., see (Zhang and Lin, 2015, Table 1) for ridge regression), we have: $\kappa = \mathcal{O}(\sqrt{m \cdot n})$ while $\beta/\mu = \mathcal{O}(\sqrt{m})$–the former grows with the local sample size n, while the latter is independent. This motivated a surge of studies aiming at exploiting function similarity coupled with acceleration to boost communication efficiency (see Sec. 1.2): linear convergence is certified with a number of communication steps scaling (asymptotically (Hendrikx et al., 2020b)) with $1 + \sqrt{\beta/\mu}$ (Zhang and Lin, 2015), which can be significantly smaller than $\sqrt{\kappa}$, and matches lower complexity bounds (Arjevani and Shamir, 2015) up to log-factors. These algorithms however are *centralized* and *cannot* be implemented over *mesh* networks. This suggests the following open question:

Is linear convergence with a number of communications scaling with $1 + \sqrt{\beta/\mu}$ achievable by any distributed algorithm over *mesh* networks?

1.1 Contributions

We provide a positive answer to the above question.

(i) Algorithm design: We proposed Accelerated-SONATA (`ACC-SONATA`), an inexact accelerated proximal method (outer loop) for (P), embedded with the distributed algorithm known as `SONATA` (Sun et al., 2022) (inner loop), which approximatively solves the proximal subproblem over mesh networks, according to a properly designed notion of inexactness. `SONATA` couples local preconditioning via surrogation of f_i with a gradient tracking mechanism (Di Lorenzo and Scutari, 2016; Xu et al., 2017), estimating locally the gradient of the global loss f. At high level, the outer loop ensures acceleration while `SONATA` exploits function similarity to boost the convergence rate of the inner loop. A direct acceleration of the mirror method, achieving $\widetilde{\mathcal{O}}(\beta/\mu)$ over star-networks (Lu et al., 2020), does not seem possible in general (Dragomir et al., 2019).

(ii) New analysis: While `ACC-SONATA` is inspired by proximal acceleration for centralized optimization (d'Aspremont et al., 2021), such as Catalyst (Hongzhou et al., 2015), formally it is not an instance of any of existing methods. It is also different from distributed algorithms accelerated á la Catalyst (Li and Lin, 2020; Hendrikx et al., 2020a), which neither exploit function similarity nor use gradient-tracking, and deal with smooth optimization ($r \equiv 0$). Hence, a new convergence analysis is needed for `ACC-SONATA`, which represents the technical contribution of this work. We hinge on new potential functions (for the inner and outer loop) that incorporate consensus errors, extrapolation variables, and gradient-tracking variables. Such potentials also shed light on the appropriate warm-restart strategy and termination criterion of the inner-loop `SONATA`, which are implementable in a distributed setting (this is not the case if using criteria in Hongzhou et al. (2018), in particular when $r \neq 0$). We remark that the proposed analysis, although developed for `ACC-SONATA`, is fairly general and potentially applicable to a variety of other distributed algorithms replacing `SONATA` in the inner loop. This will be the subject of future investigation.

(iii) Guarantees: By a proper choice of local surrogations (mirror-prox-like), `ACC-SONATA` provably achieves an ε-solution (on the objective value) of (P) in $\tilde{\mathcal{O}}\big(\sqrt{\frac{\beta/\mu}{1-\rho}} \log 1/\varepsilon\big)$ number of communications steps, where ρ characterizes the connectivity of the network. This matches for the first time lower communication complexity bounds, up to poly-log factors. On the other hand, when $1 + \sqrt{\beta/\mu} > \sqrt{\kappa}$ a different choice of surrogate (linearization of each f_i) is possible, which guarantees linear convergence with communication scaling of $\sqrt{\kappa}$–this compares favorable with existing first-order accelerated methods (see Table 1), with `ACC-SONATA` being applicable on

Table 1: **Distributed Accelerated Algorithms over Mesh Networks**: `SSDA/MSDA` (Scaman et al., 2017), `OPAPC` (Kovalev et al., 2020), `Accelerated Dual Ascent` (Uribe et al., 2020, Alg. 3), `APM-C` (Li et al., 2018), `Mudag` (Ye et al., 2020a), `Accelerated EXTRA` (Li and Lin, 2020), `DAccGD` (Rogozin et al., 2020), and `DPAG` (Ye et al., 2020b). L (resp. μ) denotes the smoothness (resp. strong convexity) constant of F while $L_{\mathrm{m}x}$ (resp. $\mu_{\mathrm{m}n}$) is the largest smoothness (smallest strong convexity) constant of the f_i's; ρ is the network connectivity; $\widetilde{\mathcal{O}}$ hides poly-log factors.

Algorithm	Problem	Similarity	Gossip Matrix	Rate (# comm.)
`SSDA/MSDA`, `OPAPC`, `Accelerated Dual Ascent`	f_i scv, $r \equiv 0$	✗		$\mathcal{O}\left(\sqrt{\frac{L_{\mathrm{m}x}}{\mu_{\mathrm{m}n}}}\sqrt{\frac{1}{1-\rho}}\log(\frac{1}{\varepsilon})\right)$
`APM-C`	f_i scv, $r \equiv 0$	✗	PSD	$\mathcal{O}\left(\sqrt{\frac{L_{\mathrm{m}x}}{\mu_{\mathrm{m}n}}}\sqrt{\frac{1}{1-\rho}}\log^2(\frac{1}{\varepsilon})\right)$
`Mudag`	f scv, $r \equiv 0$	✗	PSD	$\widetilde{\mathcal{O}}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{1}{1-\rho}}\log(\frac{1}{\varepsilon})\right)$
`Accelerated EXTRA`	f_i scv, $r \equiv 0$	✗		$\widetilde{\mathcal{O}}\left(\sqrt{\frac{L_{\mathrm{mx}}}{\mu_{\mathrm{mn}}}}\sqrt{\frac{1}{1-\rho}}\log(\frac{1}{\varepsilon})\right)$
`DAccGD`	f_i scv, $r \equiv 0$	✗		$\widetilde{\mathcal{O}}\left(\sqrt{\frac{L}{\mu}}\frac{1}{1-\rho}\log^2(\frac{1}{\varepsilon})\right)$
`DPAG`	f scv, $r \not\equiv 0$	✗	PSD	$\widetilde{\mathcal{O}}\left(\sqrt{\frac{L}{\mu}}\sqrt{\frac{1}{1-\rho}}\log(\frac{1}{\varepsilon})\right)$
ACC-SONATA (this work)	f scv, $r \not\equiv 0$	✓		$\widetilde{\mathcal{O}}\left(\sqrt{\frac{\beta}{\mu}}\sqrt{\frac{1}{1-\rho}}\log(\frac{1}{\varepsilon})\right)$

general (P) with $r \not\equiv 0$ and with no restrictions on gossip matrices. Our numerical results on synthetic and real data supports the competitive performance of our method over the state of the art.

1.2 Related Works

Acceleration under function similarity (2) has been extensively investigated over master/workers architectures while it remains unexplored over *mesh networks*, as inferred by the following discussion.

• **Master/workers systems:** Several papers employed acceleration (in the sense of Nesterov) to solve (P) over such architectures; see, e.g., Gorbunov et al. (2020); d'Aspremont et al. (2021); Hongzhou et al. (2015); Wang et al. (2020); Li et al. (2020b) and references therein for a comprehensive description of state-of-the-art methods, including their application to federated learning systems and stochastic optimization. Given the focus on the paper, next we comment in details works exploring the idea of statistical preconditioning to further decrease the communication complexity of solving (P).

`DANE` Shamir et al. (2014) is a mirror-descent type algorithm for (P) with $r = 0$ whereby workers perform a local data preconditioning via a suitably chosen Bregman divergence, and the master averages the solutions of the workers. For quadratic losses, `DANE` achieves communication complexity $\widetilde{\mathcal{O}}((\beta/\mu)^2 \log 1/\varepsilon)$. More recently, Fan et al. (2019) proposed `CEASE`, which achieves `DANE`'s complexity but for nonquadratic losses and $r \neq 0$. Applying the convergence analysis of mirror descent in Lu et al. (2020) to `CEASE` enhances its rate to $\widetilde{\mathcal{O}}((\beta/\mu)\log 1/\varepsilon)$.

Further improvements are achievable employing acceleration. Efforts include: `DiSCO` (Zhang and Lin, 2015), an inexact damped Newton method coupled with a preconditioned conjugate gradient (to compute the Newton direction), which achieves communication complexity $\widetilde{\mathcal{O}}((1 + \sqrt{\beta/\mu})\log 1/\varepsilon)$ for self-concordant losses (and $r \neq 0$); `AIDE` (Reddi et al., 2016), which uses the Catalyst framework (Hongzhou et al., 2015), matching the rate of `DiSCO` for quadratic losses (and $r = 0$); `DANE-HB` (Yuan and Li, 2020), a variant of `DANE` equipped with Heavy Ball momentum and matching for quadratic functions the communication complexity of `DiSCO` and `AIDE`; and `SPAG`(Hendrikx et al., 2020b), a preconditioned direct accelerated method, achieving for nonquadradic losses *asymptotically* the convergence rate $\mathcal{O}((1 - 1/\sqrt{\beta/\mu})^k)$ (k is the iteration index).

None of above methods are implementable over mesh networks, because they all rely on the presence of a master node. Notice also that, although designed for mesh networks, our proposed method, `ACC-SONATA` compares favorably also with the aforementioned schemes (specifically designed for star networks) by achieving communication complexity $\mathcal{O}(\sqrt{\beta/\mu}\log(\beta/\mu)\log 1/\varepsilon)$ for nonquadratic losses.

• **Acceleration over mesh networks:** Given the focus of this work, we discuss next only (provably convergent) distributed algorithms over mesh networks employing some form of acceleration–they are summarized in Table 1. Although substantially different–some are primal (Ye et al., 2020a,b; Li and Lin, 2020; Kovalev et al., 2020; Rogozin et al., 2020) others are dual or penalty-based (Scaman et al., 2017; Uribe et al., 2020; Li et al., 2018) methods, and applicable to special instances of (P) (mainly with $r = 0$) and subject to special design constraints (e.g., positive semidefinite gossip matrix)–they all achieve linear convergence rate, with communication complexity scaling some with $\sqrt{\kappa_\ell}$ ($\kappa_\ell = L_{mx}/\mu_{mn}$ is the "local" condition number) and others with $\sqrt{\kappa}$ ($\kappa = L/\mu$ is the condition number of f). Note that in general $\kappa \ll \kappa_\ell$; hence the latter group is preferable to the former. By using only gradient information of the local f_i's, none of these methods can take advantage of function similarity. This means that their rates still scale as $\sqrt{\kappa}$ no matter how small β is (even $\beta = 0$, i.e., all f_i's identical), which is highly sub-optimal when considering, e.g., ill-conditioned losses, and contrasts with the lower complexity bound $\mathcal{O}(\sqrt{\beta/\mu})$ (cf. Sec. 2.1).

To our knowledge, `Network-DANE` (Li et al., 2020a) and `SONATA` (Sun et al., 2022) are the only two methods that leverage statistical similarity to enhance convergence over mesh networks, with the latter achieving communication complexity scaling with $\widetilde{\mathcal{O}}(\beta/\mu)$ for nonquadratic losses and $r \neq 0$. These methods however are not accelerated, missing thus the more favorable scaling $\sqrt{\beta/\mu}$. The proposed accelerated method, `ACC-SONATA`, fills exactly this gap.

2 SETUP AND BACKGROUND

We study (P) under the following assumptions.

Assumption 1. *Given problem* (P),

(i) $r : \mathbb{R}^d \to \mathbb{R}$ *is a proper, closed, convex function; let* $\operatorname{dom} r$ *denote its domain;*

(ii) Each f_i *is convex and twice differentiable over (an open set containing)* $\operatorname{dom} g$*;*

(iii) f *is* μ*-strongly convex and* L*-smooth on* $\operatorname{dom} g$*, with* $0 < \mu \leq L < \infty$*. Define* $\kappa = L/\mu$*.*

Note that (i)-(ii) implies that each f_i is μ_i-strongly convex and L_i-smooth, with $0 \leq \mu_i \leq L_i < \infty$ (not each f_i need to be strongly convex); let $L_{\max} = \max_{i\in[m]} L_i$,, were we denoted $[m] \triangleq \{1, \ldots, m\}$.

Function similarity is captured by the following.

Definition 2. *Under Assumption 1, let* $\beta \geq 0$ *the smallest number such that*

$$\max_{i\in[m]} \sup_{x\in\operatorname{dom} r} \left\|\nabla^2 f_i(x) - \nabla^2 f(x)\right\| \leq \beta.$$

The smaller β, the more similar f_i's are. Note that the following bound holds for β:

$$\beta \leq \max_{i\in[m]} \max\left\{|L_i - \mu|,\ |\mu - L_i|\right\}.$$

The case of interest is of course when $1+\beta/\mu < L/\mu$, which is the typical situation of ill-conditioned f's.

Network model: The network of agents is an undirected, graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where $\mathcal{V} = [m]$ denotes the vertex set (the set of agents) while $\mathcal{E}$ is the set of edges; $\{i, j\} \in \mathcal{E}$ if there is a communication link between agent i and agent j. For the sake of notation, we assume $\{i, i\} \in \mathcal{E}$ for any $i \in [m]$. We make the blanket assumption that the graph $\mathcal{G}$ is connected.

The distributed algorithms of interest employ gossip communications–each node averages the values of its neighbors' variables. The weights of this averaging process (collected into a matrix $W \in \mathbb{R}^{m\times m}$) satisfy the following standard assumptions.

Assumption 3. *The matrix* $W \in \mathbb{R}^{m\times m}$ *belongs to the class* $W = P_M(\overline{W})$*, for some* $M \in \mathbb{N}_{++}$ *and* $\overline{W} \in \mathcal{R}_+^{m\times m}$*, where* P_M *is a polynomial with degree at most* M *with* $P_M(1) = 1$*, and* $\overline{W}$ *satisfies the following conditions:*

(i) $\overline{W}$ *is compliant with* $\mathcal{G}$*, that is its* (i, j)*-entries* $\bar{w}_{ij}$ *satisfies:* $\bar{w}_{ii} > 0$*, for all* $i \in [m]$*;* $\bar{w}_{ij} > 0$*, if* $(i, j) \in \mathcal{E}$*; and* $\bar{w}_{ij} = 0$ *otherwise;*

(ii) $1^\top \overline{W} = 1^\top$ *and* $\overline{W}1 = 1$ *(doubly stochasticity).*

Define $\rho \triangleq \left\|W - 11^\top/m\right\| < 1$*.*

The above class of weight matrices captures single ($M = 1$) and multiple ($M > 1$) rounds of communications per optimization step (notice that $P_M(1) = 1$ is to ensure the doubly stochasticity of W when $\overline{W}$ is so). Several rules have been proposed in the literature fulfilling Assumption 3, including the Laplacian, the Metropolis-Hasting, and the maximum-degree weights rules as well as Chebyshev (Wien, 2011; Scaman et al., 2017) or Jacobi (Berthier et al., 2020) polynomials-based accelerations.

2.1 Lower Complexity Bounds over Mesh Networks under Similarity

We informally state here lower communication complexity bounds over mesh networks for the class of problems (P) satisfying Assumptions 1 and 3, and certain distributed gossip algorithms of interest (see Definition 6 in Appendix A):

$$\Omega\left(\sqrt{\frac{\beta/\mu}{1-\rho}}\log\left(\frac{\mu\left\|x^\star\right\|^2}{\varepsilon}\right)\right). \tag{3}$$

The formal statement of this result can be found in the supplementary material (cf. Theorem 7, Appendix A). The next section is devoted to the design of the first distributed algorithm matching such a lower complexity bound (up to poly-log factors). As anticipated, our scheme hinges on the SONATA algorithm (Sun et al., 2022), which we recall next.

2.2 A Building Block: SONATA Algorithm

The instance of SONATA used in this work is summarized in Algorithm 1 (assumed to be applied to (P)). Each agent i owns local copies x_i of the shared optimization variable x along with the auxiliary variable y_i that is a local proxy of ∇f, which is not available at the agents' sides. In parallel and iteratively, agents update their x-variables, first solving in (S.1) a local approximation of (P) wherein $\widetilde{f}_i(x; x_i^k)$ is a surrogate of f_i at x_i^k and the linear term $y_i^k - \nabla f_i(x_i^k)$ is an estimate of $\sum_{j\neq i} \nabla f_j(x_i^k)$. This is followed by a communication step, (S.2), instrumental to enforce asymptotic consensus among the x-variables and track ∇f via the y-ones.

Several surrogate functions are feasible, see Sun et al. (2022). Here, we focus on the following two:

$$\widetilde{f}_i(x; x_i^k) = f_i(x) + \frac{\beta}{2}\|x - x_i^k\|^2; \tag{4}$$

$$\widetilde{f}_i(x; x_i^k) = f_i(x_i^k) + \langle \nabla f_i(x_i^k), x - x_i^k\rangle + \frac{L}{2}\|x - x_i^k\|^2. \tag{5}$$

Note that the use of the linearization (5) corresponds to perform at each agent's side a proximal gradient step; this is the typical update of the majority of existing distributed algorithms (as those in Table 1). Such a choice does not permit to take advantage of function similarity, if any. In fact, when (5) is employed and a weight matrix W satisfying Assumption 3 is used for the consensus and tracking steps, SONATA applied to (P) achieves an ε-solution (in terms of objective value) in $\widetilde{\mathcal{O}}\big(\kappa \frac{1}{1-\rho} \log 1/\varepsilon\big)$ number of communications. Communication saving under function similarity is provably achievable instead using surrogate (4), resulting in a communication complexity of $\widetilde{\mathcal{O}}\big(\frac{\beta}{\mu} \cdot \frac{1}{1-\rho} \log 1/\varepsilon\big)$. This motivated us to use SONATA as building block of our accelerated method aiming at exploiting function similarity.

3 ACCELERATED SONATA

We are ready to introduce our main algorithm, Algorithm 2. At high level, the scheme can be interpreted as a successive application of SONATA for the inexact minimization of the function

$$u_k(x) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_i^k(x) + r(x), \tag{6}$$

Algorithm 1 SONATA($\{f_i\}_{i\in[m]}$, x^0, y^0, T)

Input: $\{f_i(x)\}_{i\in[m]}$, $r(x)$ [cf. (P)];
$x^0 = (x_i^0)_{i\in[m]}$ [init. points],
$y^0 = (y_i^0)_{i\in[m]}$ [grad.-tracking init.],
$T > 0$ [# iterations];
Output: $x^T = (x_i^T)_{i\in[m]}$, $y^T = (y_i^T)_{i\in[m]}$;
for $k = 0, 1, 2, \ldots, T-1$ **do**
(S.1) Local computations: for all $i \in [m]$,

$$x_i^{k+1/2} = \underset{x\in\mathbb{R}^d}{\operatorname{argmin}}\ \tilde{f}_i(x; x_i^k) + \langle y_i^k - \nabla f_i(x_i^k), x - x_i^k\rangle + r(x);$$

(S.2) Communications: for all $i \in [m]$,

$$x_i^{k+1} = \sum_{j=1}^{m} w_{ij} x_j^{k+1/2},$$

$$y_j^{k+1} = \sum_{j=1}^{m} w_{ij}\big(y_j^k + \nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)\big).$$

end for

with $f_i^k(x) = f_i(x) + (\delta/2)\|x - z_i^k\|^2$, wherein the z-variable in the quadratic term plays the role of the extrapolation à la Nesterov, to gain acceleration. The use of SONATA in the inner loop allows us to take advantage of function similarity, if any, by choosing surrogates as in (4) and a suitable value for $\delta > 0$ (see Theorem 4). Notice the warm restart of SONATA every T iterations, with in particular the gradient tracking initialization unconventionally chosen, as recommended by our convergence analysis.

Algorithm 2 Accelerated SONATA

Input: β, μ, $\delta > 0$, $\alpha = \sqrt{\mu/(\mu+\delta)}$;
$x_i^0 = z_i^0 = z_i^{-1} = 0$, $y_i^0 = \nabla f_i(x_i^0)$
Output: $x^K = (x_i^K)_{i\in[m]}$
for $k = 0, 1, 2, \ldots, K-1$ **do**
Set: $f_i^k(x) = f_i(x) + \frac{\delta}{2}\left\|x - z_i^k\right\|^2$;
(S.1) Inner loop via SONATA:

$$\big(x^{k+1}, y^{k+1}\big) = \texttt{SONATA}\Big(\big\{f_i^k\big\}_{i\in[m]}, x^k, y^k + \delta\left(z^{k-1} - z^k\right), T\Big);$$

(S.2) Extrapolation step:

$$z_i^{k+1} = x_i^{k+1} + \frac{1-\alpha}{1+\alpha}(x_i^{k+1} - x_i^k), \quad \forall i \in [m].$$

end for

Algorithm rationale: The genesis of the algorithmic design can be traced back to the idea of acceleration of a centralized inexact proximal method (outer loop) (see, e.g., d'Aspremont et al. (2021)), whose proximal subproblems are approximately solved in a distributed fashion via the SONATA algorithm (inner loop), satisfying a suitable notion of inexactness (defined in this paper) for proximal operations. In fact, assuming one can absorb consensus errors on the agents' variables x_i's and momentum vectors z_i's into such a criterion of solution approximation, we can approximate (S.1) and (S.2) as

$$x_i^k \approx \bar{x}^k \triangleq \frac{1}{m}\sum_{i=1}^m x_i^k \quad \text{and} \quad z_i^k \approx \bar{z}^k \triangleq \frac{1}{m}\sum_{i=1}^m z_i^k,$$

$$\texttt{(S.1)}': \quad \bar{x}^{k+1} \approx \operatorname*{argmin}_{x\in\mathbb{R}^d} u(x) + \frac{\delta}{2}\left\|x - \bar{z}^k\right\|^2,$$

$$\texttt{(S.2)}': \quad \bar{z}^{k+1} = \bar{x}^{k+1} + \frac{1-\alpha}{1+\alpha}\left(\bar{x}^{k+1} - \bar{x}^k\right),$$

where we used the fact that the minimization of u_k in (6) and that of the function in (S.1)′ have the same solution. The dynamics above are a resemble of an inexact proximal acceleration (d'Aspremont et al., 2021; Hongzhou et al., 2015, 2018).

Challenges: Despite the above connection, existing convergence analyses of centralized accelerated methods break down when applied to ACC-SONATA. Specifically, (i) the notions of approximate solutions for proximal problems as in d'Aspremont et al. (2021); Hongzhou et al. (2018) cannot be satisfied here, because of the aforementioned consensus errors, let alone their practical verification in a distributed setting and in the presence of the nonsmooth function r; and (ii) the potential functions used therein are not adequate to certify linear convergence of the outer loop of ACC-SONATA at the desired accelerated rate, they do not capture unavoidable consensus and gradient tracking errors coming out of the inexact, distributed minimization of u_k in (6) via SONATA. Furthermore, the convergence proof of SONATA as in Sun et al. (2022) is not directly applicable to study the inner loop, due to the unconventional restart of the gradient tracking variables. Also, R-linear convergence of the objective-value gap and consensus/tracking errors therein seems no longer sufficient to provably obtain acceleration of the outer loop. Our convergence analysis addresses these challenges–we refer to Appendix C for the complete proof (and Appendix C.1 for a sketch).

3.1 Convergence Analysis

Communication complexity of Acc-SONATA is established in Theorem 4 and Theorem 5 below, pertaining to the use in the inner algorithm SONATA of the surrogates (4) and (5), respectively. The explicit expression of the constants hidden in the big-O notation can be found in the supplementary material.

Theorem 4. *Consider problem* (P) *under Assumption 1, with optimal value function* $u^\star$ *and* $\beta > \mu$ *w.l.o.g.. Let* $\{x^k \triangleq (x_i^k)_{i\in[m]}\}$ *be the sequence generated by* ACC-SONATA *under Assumption 3, with*

$$\rho \le \mathcal{O}\left(\left(1 + \frac{\kappa - 1}{\beta/\mu}\right)^{-2}\right), \tag{7}$$

and the following tuning:

$$\delta = \beta - \mu, \quad T = \mathcal{O}\left(\log \beta/\mu\right),$$

and agents' surrogate functions (4) *in* SONATA. *Define* $\bar{x}^k \triangleq \frac{1}{m}\sum_{i=1}^m x_i^k$ *and the optimality gap*

$$\Delta(x^k) \triangleq \max\left(\frac{1}{m}\sum_{i=1}^m u(x_i^k) - u^\star, \frac{1}{m}\sum_{i=1}^m \left\|x_i^k - \bar{x}^k\right\|^2\right). \tag{8}$$

Then, there holds

$$\Delta(x^k) = \mathcal{O}\left(\left(1 - c\sqrt{\mu/\beta}\right)^k\right),$$

where $c \in (0,1)$ *is some universal constant. Therefore,* $\Delta(x^K) \le \varepsilon$*,* $\varepsilon > 0$*, in*

$$\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}} \cdot T \cdot \log\frac{1}{\varepsilon}\right) \tag{9}$$

total (inner plus outer loop) communication steps.

Note that (9) states linear convergence with optimal dependence on β/μ, up to the log-factor $T = \log(\beta/\mu)$. This is achieved under (7), which requires the network to be sufficiently connected. If the network is not part of the design, (7) might not be satisfied by the topology under consideration. Still, (7) can be enforced by running multiple communication rounds per iteration (computation steps) in the inner loop SONATA. Specifically, let $\bar{\rho} = \|\overline{W} - 11^\top/m\|$ be the connectivity of the graph associated with a given weight matrix $\overline{W}$ (satisfying Assumption 3); suppose we run M steps of communications per iteration (computation) in Step (S.2) of the SONATA algorithm, each time using the weight matrix $\overline{W}$. This yields an effective network with matrix $W = \overline{W}^M$ ($P_M(x) = x^M$) and improved connectivity $\rho = \bar{\rho}^M$. One can then choose M so that $\bar{\rho}^M$ satisfies (7), resulting in $M = \mathcal{O}\big(\log(1 + (\kappa - 1)/(\beta/\mu))/\log(1/\bar{\rho})\big) = \mathcal{O}\big(\log(1 + (\kappa - 1)/(\beta/\mu))/(1 - \bar{\rho})\big)$ rounds of communications. The dependence on $\bar{\rho}$ can be improved leveraging Chebyshev polynomials of the gossip matrix $\overline{W}$ (further assumed to be symmetric) of order at most M, that is, $W = P_M(\overline{W})$, where P_M are Chebyshev polynomials. At the agents' side, the updates $P_M(\overline{W})$ are implemented via a shift-register gossip protocol, resulting in time-varying weights w_{ij}'s in (S.2) of the SONATA algorithm–see Berthier et al. (2020) for details. It is not difficult to show that $M = \mathcal{O}\big(\log(1 + (\kappa - 1)/(\beta/\mu))/(\sqrt{1-\bar{\rho}})\big)$ suffices to satisfy (7), yielding an overall communication complexity

$$\mathcal{O}\left(\sqrt{\frac{\beta/\mu}{1-\bar{\rho}}}\,\log\left(1+\frac{\kappa-1}{\beta/\mu}\right)\,\log\left(\frac{\beta}{\mu}\right)\,\log\frac{1}{\varepsilon}\right). \tag{10}$$

This matches the lower complexity bound in (3) (cf. Theorem 7 in Appendix A) up to poly-log factors.

Consistently with the behaviour of SONATA, if surrogates in the form (5) are used by the agents, acceleration is still achieved, but with a linear rate scaling with $\sqrt{\kappa}$ rather than $\sqrt{\beta/\mu}$. This is formalized next.

Theorem 5. *Consider problem* (P) *under Assumption 1, with optimal value function $u^\star$ and $\kappa > 1$ w.l.o.g.. Let $\{x^k \triangleq (x_i^k)_{i\in[m]}\}$ be the sequence generated by* ACC-SONATA *under Assumption 3, with*

$$\rho \leq \mathcal{O}\left(\left(2+\frac{\beta/\mu-1}{\kappa}\right)^{-2}\right), \tag{11}$$

and the following tuning:

$$\delta = L - \mu, \quad T = \mathcal{O}\left(\log\kappa\right),$$

and agents' surrogate functions (5) *in* SONATA*. Then, there holds*

$$\Delta(x^k) = \mathcal{O}\left(\left(1 - c\,\frac{1}{\sqrt{\kappa}}\right)^k\right), \tag{12}$$

where $c \in (0,1)$ is some universal constant. Therefore, $\Delta(x^K) \leq \varepsilon$, $\varepsilon > 0$, in

$$\mathcal{O}\left(\sqrt{\kappa}\,T\log\frac{1}{\varepsilon}\right) \tag{13}$$

total (inner plus outer loop) communication steps.

Enforcing (11) via multiple rounds of communications based on Chebyshev polynomials (applied to a symmetric matrix $\overline{W}$, with $\bar{\rho} = \|\overline{W} - 11^\top/m\|$), the communication complexity in Theorem 5 becomes

$$\mathcal{O}\left(\sqrt{\frac{\kappa}{1-\bar{\rho}}}\,\log\left(2+\frac{\beta/\mu-1}{\kappa}\right)\,\log\left(\kappa\right)\,\log\frac{1}{\varepsilon}\right). \tag{14}$$

Comparing (10) with (14) shows that, if properly exploited, function similarity provably leads to communication saving. This calls for the use of the surrogate (4) over (5); and thus it comes generally at the cost of solving computationally more demanding subproblems at agents' sides. This is common to all the methods (including centralized) exploiting similarity and discussed in Sec. 1.2, and seems an unavoidable tradeoff. These methods are in fact designed with the goal of saving communications, at the cost of more computations.

Inexact ACC-SONATA: To alleviate the computation cost of solving agent's subproblems with surrogate (4) when a closed form solution is not available, in Appendix E, we discuss how to modify ACC-SONATA to accommodate inexact solutions of agents' subproblems in Step (S.1) of SONATA. We defer to the appendix for details; here we only point out that, by carefully choosing the inexact criterion for solving approximately the local optimization subproblems, the communication complexity of the resulting inexact ACC-SONATA, termed Inexact ACC-SONATA (Algorithm 5), matches that of ACC-SONATA as in (9) (see Theorem 12). We also study the computational complexity of Inexact-ACC-SONATA (see Theorem 13). For instance, if each agent's subproblem with surrogate (4) is solved (up to a suitably chosen accuracy) via accelerated proximal gradient, Inexact ACC-SONATA reaches an ε-solution of (P) after

$$\tilde{\mathcal{O}}\left(\sqrt{1+\frac{\kappa+\beta/\mu}{2\beta/\mu-1}}\cdot\frac{\beta}{\mu}\cdot\left(\log\frac{1}{\varepsilon}\right)^2\right) \tag{15}$$

total gradient evaluations/agent, where $\tilde{\mathcal{O}}$ hides log-factors. On the other hand, if surrogates (5) are used in the agents' subproblem, the total computation complexity of Inexact ACC-SONATA is still given by Theorem 5, and thus reads $\tilde{\mathcal{O}}(\sqrt{\kappa}\log(1/\varepsilon))$, which might be more favorable than (15).

Quite interestingly, the proposed accelerated framework offers the flexibility, within the same algorithm, to privilege computation or communication savings, based upon the choice of the right surrogate function, achieving (up to poly-log factors) either optimal computation or communication complexity (under similarity).

ACC-SONATA on star-networks: Although ACC-SONATA has been designed specifically for mesh networks, it readily applies to master/workers architectures; details can be found in the supplementary material. Here we only remark that a direct application of Theorem 4 and Theorem 5 leads to the following communication complexity to solve (P) over master/workers architectures

$$\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}}\log\left(\frac{\beta}{\mu}\right)\log\frac{1}{\varepsilon}\right) \text{ and } \mathcal{O}\left(\sqrt{\kappa}\log(\kappa)\log\frac{1}{\varepsilon}\right),$$

respectively. Quite interestingly, these rates compare favorably with those of the centralized methods discussed in Sec. 1.2.

4 NUMERICAL RESULTS

We present numerical results on synthetic and real data, corroborating our complexity analysis (Theorems 4 and 5). Additional experiments on different problem classes and data sets as well as including more algorithms are reported in the supplementary material.

1) Ridge regression: Our first experiment concerns a ridge regression problem over a network of agents, modeled as a Erdos-Renyi graph with $m = 30$ nodes and edge probability $p = 0.5$. The problem is an instance of (P) with $f_i(x) = 1/(2n)\|A_i x - b_i\|^2 + \lambda\|x\|^2$ [agent i owns data $A_i \in \mathbb{R}^{n\times d}$, $b_i \in \mathbb{R}^n$] and $r = 0$. Data are generated as follows (Sun et al., 2022): Each row of A_i is i.i.d, drawn from $\mathcal{N}(0, \Sigma)$, where $\Sigma = \sum_{j=1}^{d} \lambda_j u_j u_j^\top$. The λ_j's are uniformly distributed in $[\mu_0,\, L_0]$, with $\mu_0 = 1$ and $L_0 = 1000$, and $\mathbf{u}_1, \ldots, \mathbf{u}_d$ are obtained via the QR decomposition of a $d \times d$ random matrix with standard Gaussian i.i.d elements. We set $b_i = A_i x^\star + w_i$, where $x^\star \sim \mathcal{N}(5 \cdot 1_d, I)$ is the ground truth and $w_i \sim \mathcal{N}(0, 0.1 \cdot I)$ is the additive noise (1_d is the d-dimensional vector of all ones).

Algorithms: We simulated two instances of `ACC-SONATA`, corresponding to the choices of the surrogates (4) and (5) in `SONATA` (inner-loop); we termed them as `ACC-SONATA-F` and `ACC-SONATA-L`, respectively (`F` stands for "full local function" while `L` for "linearization"). The solution of the agents' subproblems solved in `ACC-SONATA-F` and `ACC-SONATA-L` is computed in closed form. According to Theorems 4 and 5, `ACC-SONATA-F` is expected to outperform first-order methods, including `ACC-SONATA-L`, whenever $1 + \beta/\mu < \kappa$, while `ACC-SONATA-L` should be competitive otherwise. The free parameters of these two instances are tuned as suggested by the theory, with $T = \lceil\log(\beta/\mu)\rceil$ for `ACC-SONATA-F` and $T = \lceil\log(\kappa)\rceil$ for `ACC-SONATA-L`, where L and μ are estimated by the data (quadratic function) and so β using Definition 2. The weight matrix W according to the Metropolis-Hasting rule. We compare our algorithms with the following, widely tested in the literature (Table 1): `APM-C`, `Mudag`, and `ACC-EXTRA`. The tuning of these schemes follows the recommendations as in their respective papers. In particular, `Mudag` and `APM-C` require the gossip matrix to be positive definite, so we set $(W + I)/2$, with W being the matrix used in the other algorithms.

In Fig. 2(**left-panel**), we fix $\beta/\mu < \kappa$, and plot the optimality gap $\frac{1}{m}\sum_{i=1}^{m}\|x_i^k - x_{\text{rg}}\|^2$ versus the total number of communications, for each of the algorithms, where x_{rg} is the solution of the ridge regression problem (computed in closed form). All the schemes achieve linear convergence. As predicted, `ACC-SONATA-F` outperforms the other accelerate methods that do not take advantage of function similarity. Quite interestingly, `ACC-SONATA-L` compares quite favorably also with directed acceleration methods, such as `Mudag` and `AMP-C`, while sharing similar computational costs.

To investigate the impact of κ and β/μ on the convergence rate of the algorithms, in the next experiment we consider the following two scenarios:

(i) Changing β/μ with (almost) fixed κ: We generate instances of ridge regression with decreasing β and (almost) fixed κ, setting $\lambda = 0$ and increasing the local sample size n (20 values) within $[100, 40000]$; the empirical κ remains approximately constant close to the nominal value $L_0/\mu_0 = 1000$. Fig. 2(**mid-panel**) captures this scenario, we plot the number of communications to drive the optimality gap below 10^{-4} versus β/μ, and $\kappa \approx 930$.

(ii) Changing κ with fixed β/μ: We generate a sequence of instances of ridge regression with varying κ (acting on λ) while keeping β/μ constant by changing the local sample size n to compensate for the variations of μ (due to λ). Fig. 2(**right-panel**) plots the number of communications to drive the optimality gap below 10^{-4} versus κ, for $\beta/\mu \approx 232$.

The following comments are in order. First, the **mid-panel** confirms what predicted by Theorem 4: the number of communications of `ACC-SONATA-F` scales roughly with $\sqrt{\beta/\mu}$ while that of first-order distributed schemes is fairly invariant with β/μ. This is because methods using only gradient information (including `ACC-SONATA-L`) cannot benefit from statistical similarity. On the other hand, the **right-panel** shows that communication complexity of accelerated first-order methods (including `ACC-SONATA-L`, as stated in Theorem 5) deteriorates when κ grows whereas that of `ACC-SONATA-F` is (almost) invariant. It is interesting to remark that `ACC-SONATA-F` remains competitive, outperforming

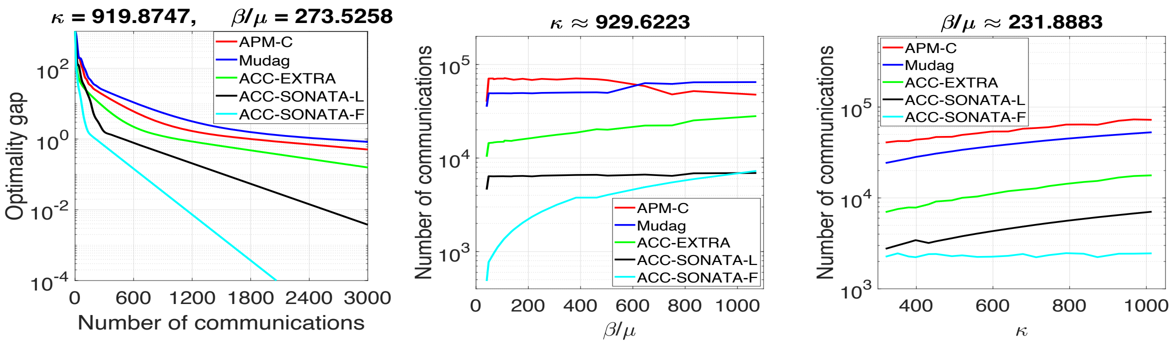

Figure 1: Comparison of distributed accelerated algorithms on ridge regression (synthetic data). **(left panel)**: optimality gap versus total number of communications, for given β/μ and κ; **(mid panel)**: number of communications to reach a precision of 10^{-4} versus β/μ, for fixed κ; **(right panel)**: same quantity versus κ, for fixed β/μ.

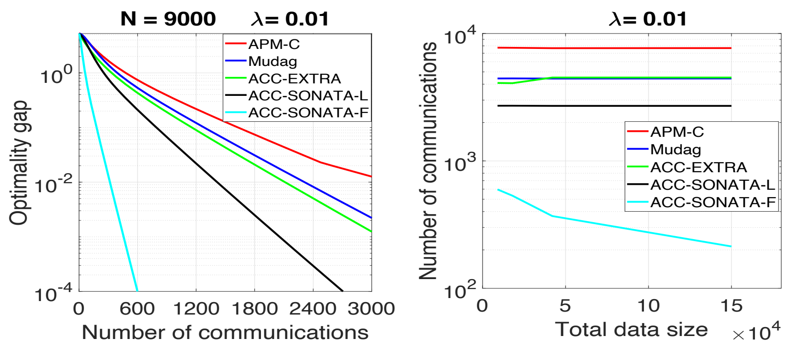

Figure 2: Comparison of distributed accelerated algorithms on hinge loss minimization (`COV1` dataset). **(left panel)**: optimality gap versus total number of communications; **(right panel)**: number of communications to reach a precision of 10^{-4} versus (total) sample.

the other schemes, even when $\beta/\mu \approx \kappa$.

2) Hinge loss minimization on real data: We consider the minimization problem $\min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} l(y_i\langle x, z_i\rangle) + \frac{\lambda}{2}\|x\|^2$ over the same network of agents considered in the previous example, where ℓ is the smooth hinge loss as in Shamir et al. (2014). We experimented on the data set `COV1` (see, e.g., Shalev-Shwartz and Zhang (2013)). The tuning of the algorithm is the same a described in the previous experiment, based upon estimation of the quantities μ, L, L_{mx}, μ_{mn}, and β from the data–see supplementary material for details. We notice that $\mu \approx \mu_{\text{mn}} \approx \lambda$. The solution of the agents' subproblems in `ACC-SONATA-F` is estimated up to the accuracy 10^{-10} by running the gradient algorithm with step-size equal to 0.03.

Fig. 2 (**left-panel**) plots the optimality gap $\frac{1}{m}\sum_{i=1}^{m}\|x_i^k - x_{\text{op}}\|^2$ versus the total number of communications, for each of the algorithms, where x_{op} is an estimate of the solution of the problem (obtained running the centralized gradient algorithm). The results show that the proposed methods are competitive also on real data, with `ACC-SONATA-F` outperforming the others, when enough samples are present at the agents' sides. The **right-panel** plots the number of communications to reach an accuracy of 10^{-4} versus the total sample size (since m is fixed, the local samples size varies). This corresponds to decrease β while keeping μ (roughly) constant. As predicted, we observe that the communication saving experienced by `ACC-SONATA-F` improves with the local sample size, as the method takes advantage of the local function structure, a feature that first-order methods are lacking.

Acknowledgments

We are grateful to the five anonymous Referees for their thoughtful and constructive comments, which helped improve the quality of the paper.

The work of Tian, Scutari, and Cao has been supported by the Army Research Office (ARO) under the grant No. W911NF1810238, and the Office of Naval Research (ONR) under the grant No. N00014-21-1-2673.

The work of Gasnikov was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Ivannikov Institute for System Programming of the Russian Academy of Sciences dated November 2, 2021, No. 70-2021-00142. A. Gasnikov is also grateful to D. Kovalev (KAUST) for useful discussions.

References

Arjevani, Y. and Shamir, O. (2015). Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764.

Bekkerman, R., Bilenko, M., and Langford, J. (2011). *Scaling up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press.

Berthier, R., Bach, F., and Gaillard, P. (2020). Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM J. on Mathematics of Data Science*, 1:24–47.

Chang, C.-C. and Lin, C.-J. (2011). Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27.

d'Aspremont, A., Scieur, D., and Taylor, A. B. (2021). Acceleration methods. *ArXiv preprint, arXiv:2101.09545*.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Di Lorenzo, P. and Scutari, G. (2016). Next: In-network nonconvex optimization. *IEEE Trans. Signal Inf. Process. Netw.*, 2(2):120–136.

Dragomir, R.-A., Taylor, A., d'Aspremont, A., and Bolt, J. (2019). Optimal complexity and certification of bregman first-order methods. *arXiv preprint, arXiv:1911.08510*.

Fan, J., Guo, Y., and Wang, K. (2019). Communication-efficient accurate statistical estimation. *arXiv:1906.04870*.

Gorbunov, E., Rogozin, A., Beznosikov, A., Dvinskikh, D., and Gasnikov, A. (2020). Recent theoretical advances in decentralized distributed convex optimization. *arXiv:2011.13259*.

Hendrikx, H., Bach, F., and Massoulié, L. (2020a). Dual-free stochastic decentralized optimization with variance reduction. In *Advances in Neural Information Processing Systems*.

Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., and Massoulie, L. (2020b). Statistically preconditioned accelerated gradient method for distributed optimization. In *International Conference on Machine Learning*, pages 4203–4227. PMLR.

Hongzhou, L., Mairal, J., and Harchaoui, Z. (2015). A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392.

Hongzhou, L., Mairal, J., and Harchaoui, Z. (2018). Catalyst acceleration for first-order convex optimization: from theory to practice. *J. of Machine Learning Research*, pages 1–54.

Kovalev, D., Salim, A., and Richtárik, P. (2020). Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33.

Li, B., Cen, S., Chen, Y., and Chi, Y. (2020a). Communication-efficient distributed optimization in networks with gradient tracking and variance reduction. *J. of Machine Learning Research*, (180):1–51.

Li, H., Fang, C., Yin, W., and Lin, Z. (2018). A sharp convergence rate analysis for distributed accelerated gradient methods. *arXiv preprint arXiv:1810.01053*.

Li, H. and Lin, Z. (2020). Revisiting extra for smooth distributed optimization. *SIAM Journal on Optimization*, 30(3):1795–1821.

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

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu., J. (2017). Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340.

Lu, H., Freund, R. M., and Nesterov, Y. (2020). Relatively smooth convex optimization by first-order methods, and applications. *SIAM J. on Optimization*, 28(1):333–354.

Reddi, S. J., Konecny, J., Richtarik, P., Poczos, B., and Smola, A. (2016). Aide: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*.

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press.

Rogozin, A., Lukoshkin, V., Gasnikov, A., Kovalev, D., and Shulgin, E. (2020). Towards accelerated rates for distributed optimization over time-varying networks. *arXiv preprint arXiv:2009.11069*.

Scaman, K., Bach, F., Bubeck, S., Lee, Y. T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International conference on machine learning*, pages 3027–3036. PMLR.

Shalev-Shwartz, S. and Zhang, T. (2013). Stochastic dual coordinate ascent methods for regularized loss. *J. of Machine Learning Research*, (14):567–599.

Shamir, O., Srebro, N., and Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *International conference on machine learning*, pages 1000–1008. PMLR.

Sun, Y., Daneshmand, A., and Scutari, G. (2022). Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation. *SIAM J. on Optimization*, pages 1–31.

Uribe, C. A., Lee, S., Gasnikov, A., and Nedić, A. (2020). A dual approach for optimal algorithms in distributed optimization over networks. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–37. IEEE.

Wang, J., Tantia, V., Ballas, N., and Rabbat, M. (2020). Improving communication-efficient distributed sgd with slow momentum. In *The International Conference on Learning Representations (ICLR)*.

Wien, A. (2011). *Iterative solution of large linear systems*. Lecture Notes, TU Wien.

Xu, J., Zhu, S., Soh, Y. C., and Xie, L. (2017). Convergence of asynchronous distributed gradient methods over stochastic networks. *IEEE Trans. Automat. Contr.*, 63(2):434–448.

Ye, H., Luo, L., Zhou, Z., and Zhang, T. (2020a). Multi-consensus decentralized accelerated gradient descent. *arXiv preprint arXiv:2005.00797*.

Ye, H., Zhou, Z., Luo, L., and Zhang, T. (2020b). Decentralized accelerated proximal gradient descent. *Advances in Neural Information Processing Systems*, 2020.

Yuan, X.-T. and Li, P. (2020). On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond. *J. Mach. Learn. Res.*, 21:206–1.

Zhang, Y. and Lin, X. (2015). Disco: Distributed optimization for self-concordant empirical loss. In *International conference on machine learning*, pages 362–370. PMLR.

Supplementary Material: Acceleration in Distributed Optimization under Similarity

This document serves as supplementary material of the paper entitled "Acceleration in Distributed Optimization under Similarity". It contains additional theoretical and numerical results, along with all the proofs of the theorems presented in the main paper. Specifically

Sec. A contains the lower complexity bounds for the class of problems and oracle algorithms of interest;

Sec. B presents additional numerical results for different classes of problems on synthetic and real data;

Sec. C provides the proof of Theorem 4 and Theorem 5;

Sec. D customizes `ACC-SONATA` to star-topologies (master/workers architectures).

A LOWER COMPLEXITY BOUNDS OVER MESH NETWORKS UNDER SIMILARITY

Given (P) over a mesh network $\mathcal{G}$, we consider the following general class of distributed algorithms, which generalize the oracle model (Arjevani and Shamir, 2015) for centralized schemes (star architectures) and smooth ($r = 0$) instances of (P).

Definition 6 (Distributed oracle)**.** *Each agent i has its own local memory $\mathcal{M}_i = \{0\}$, updated as follows:*

• `Local computation:` *Between communication rounds, each agent i iteratively computes and adds to $\mathcal{M}_i$ some finite number of points x, each satisfying*

$$\tau_1 \, x + \tau_2 \nabla f_i(x) + \tau_3 \, g \, \in \mathrm{span}\Big\{ x', \, \nabla f_i(x'), (\nabla^2 f_i(x') + D)x'', (\nabla^2 f_i(x') + D)^{-1} x'' \Big\},$$

for given $x', x'' \in \mathcal{M}_i$, $g \in \partial r(x)$, $\tau_1, \tau_2, \tau_3 \geq 0$ such that $\tau_1 + \tau_2 + \tau_3 > 0$; and D is a diagonal matrix such that the inverse matrices above exist;

• `Communications:` *After every communication round, each agent i updates $\mathcal{M}_i$ according to*

$$\mathcal{M}_i = \mathrm{span} \left\{ \underset{(i,j)\in\mathcal{E}}{\cup} M_j \right\};$$

• `Output:` *$x_i \in \mathcal{M}_i$, for all $i \in [m]$.*

The above procedure models a fairly general class of distributed algorithms over graphs. Computations at each node are based on linear operations involving current or past iterates, local gradients, and vector products with local Hessians and their inverses; exact optimization of local subproblem involving such quantities or proximal solutions are also incorporated. During communications, the agents can share with their neighbors any of the vectors they have computed up until that time.

The following result provides lower complexity bounds for solving Problem (P) by any distributed algorithm in $\mathcal{A}$–this is an extension of (Arjevani and Shamir, 2015, Th. 1) to the decentralized setting.

Theorem 7. *For any $\mu \in [0, 1)$, $\beta \in (0, 1)$, and $\rho \in [0, 1)$, there exist (i) an instance of* (P) *(with sufficiently large d and solution $x^\star$) satisfying Assumption 1, with $r = 0$, f_i's being β-similar and $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x)$ being 1-smooth and μ-strongly convex; and (ii) a gossip matrix W satisfying Assumption 3 over the graph $\mathcal{G}$ with parameter ρ such that for any distributed algorithm in $\mathcal{A}$ using the matrix W in the communications,* *the number of communication rounds N required for obtaining the solution x_i's such that $u(x_i) - u(x^\star) \le \epsilon$, for all $i \in [m]$, is*

$$N = \Omega\left(\sqrt{\frac{\beta/\mu}{1-\rho}}\log\left(\frac{\mu\left\|x^\star\right\|^2}{\varepsilon}\right)\right).$$

Proof. The case of fully connected networks ($\rho = 0$) has been already studied in (Arjevani and Shamir, 2015, Th. 1). Therefore, here we assumed $\rho > 0$. Our proof is inspired by (Scaman et al., 2017, Th. 2), with some key differences due to the different setting of our problem.

We first introduce the local cost functions of agents numbered as $1, 2, \ldots, m$. With $\zeta = \frac{1}{32}$, define the following two subsets of agents:

$$\mathcal{A}_l = \left\{i \,\middle|\, 1 \le i \le \lceil \zeta m \rceil\right\} \quad \text{and} \quad \mathcal{A}_r = \left\{i \,\middle|\, \lfloor (1-\zeta)m \rfloor + 1 \le i \le m\right\}.$$

We then define for each agent i the cost function $f_i : \ell^2 \to \mathbb{R}$ as

$$f_i(x) = \begin{cases} \frac{\beta(1-\mu)}{8}\frac{m}{\lceil \zeta m\rceil} x^\top A_1 x - \frac{\beta(1-\mu)}{4}\frac{m}{\lceil \zeta m\rceil} e_1^\top x + \frac{\mu}{2}\left\|x\right\|^2, & 1 \le i \le \lceil \zeta m \rceil \\ \frac{\mu}{2}\left\|x\right\|^2, & \lceil \zeta m \rceil + 1 \le i \le \lfloor (1-\zeta)m \rfloor \\ \frac{\beta(1-\mu)}{8}\frac{m}{\lceil \zeta m\rceil} x^\top A_2 x + \frac{\mu}{2}\left\|x\right\|^2, & \lfloor (1-\zeta)m \rfloor + 1 \le i \le m \end{cases}$$

with

$$A_1 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 1 & -1 & \cdots \\ 0 & 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \qquad A_2 \triangleq \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & \cdots \\ -1 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}.$$

We define the distance between the two subsets $\mathcal{A}_l$ and $\mathcal{A}_r$ as d_c. It follows that, to have at least one non-zero element in the kth entry of the local copies of agents in both of the above two subsets, one must perform at least k local computation steps and $(k-1)d_c$ communication steps. The number of communication rounds required to obtain $f(\hat{x}) - f(x^\star) \le \epsilon$ is thus

$$\Omega\left(d_c\sqrt{\frac{\beta}{\mu}}\log\left(\frac{\mu\left\|x^\star\right\|^2}{\epsilon}\right)\right).$$

We then describe the communication graph for the given ρ. For $m \ge 2$, we define

$$\rho_m = \frac{\rho}{2+\rho} + \frac{2}{2+\rho}\cos\frac{\pi}{m}.$$

Since $\rho_2 = \frac{\rho}{2+\rho} < \rho$ and $\lim_{k\to\infty}\rho_k = 1$, we know that there exist m such that $\rho_m < \rho \le \rho_{m+1}$. We discuss the cases of $m \ge 3$ and $m = 2$ separately:

i) $m \ge 3$. We begin defining a Laplacian matrix $L_{m,a}$ for a line graph composed of m nodes; specifically:

$$L_{m,a} = L_{m,a}^\top, \quad L_{m,a}\,1 = 0, \quad \text{and} \quad L_{m,a}(i, i+1) = a\,\mathbb{1}\{i=1\} - 1,$$

with $a \in [0, 1)$, where $\mathbb{1}\{\bullet\}$ is the indicator function. We then define

$$W_{m,a} = I - \frac{1}{2+\rho}L_{m,a}.$$

It is not difficult to check that $W_{m,a}$ satisfies Assumption 3, with $\left\|W_{m,0} - 1\,1^\top/m\right\| = \rho_m$. Furthermore, since $\left\|W_{m,1} - 1\,1^\top/m\right\| = 1$, by continuity, we know that there exists an $a \in (0,1)$ such that $\left\|W_{m,a} - 1\,1^\top/m\right\| = \rho$. In addition, we have

$$1 - \rho \geq 1 - \rho_{m+1} = \frac{2}{2+\rho}\left(1 - \cos\frac{\pi}{m+1}\right) \geq \frac{2}{3}\left(1 - \cos\frac{\pi}{m+1}\right) \overset{(*)}{\geq} \frac{8}{3}\frac{1}{(m+1)^2}.$$

Note that $(*)$ is due to that $\cos\frac{\pi}{n} \leq 1 - \frac{4}{n^2}$ for $n \geq 4$. Equivalently,

$$m \geq \sqrt{\frac{8}{3(1-\rho)}} - 1.$$

The distance between the two subsets is thus

$$d_c = \lfloor (1-\zeta)m \rfloor + 1 - \lceil \zeta m \rceil \geq \frac{15}{16}m - 1 \geq \frac{15}{16}\left(\sqrt{\frac{8}{3(1-\rho)}} - 1\right) - 1 \overset{(*)}{\geq} \frac{4}{25}\sqrt{\frac{1}{1-\rho}}.$$

Note that $(*)$ is due to that $\rho > \rho_3 > \frac{1}{2}$.

ii) $m = 2$. In this case, we have $\rho \leq \rho_3 = \frac{1+\rho}{2+\rho}$ and equivalently, $\rho \in (0, \frac{\sqrt{5}-1}{2}]$. We define the Laplacian matrix for a complete graph of 3 nodes as: $L_a = L_a^\top$, $L_a 1 = 0$, and for $i < j$, $L_a(i,j) = a\,\mathbb{1}\{i=1,\, j=2\} - 1$. We then set $W_a = I - \frac{1}{3}L_a$. Due to $\left\|W_1 - 1\,1^\top/3\right\| = \frac{2}{3} > \frac{\sqrt{5}-1}{2}$ and $\left\|W_0 - 1\,1^\top/3\right\| = 0$, there exists an $a \in (0,1)$ such that $\left\|W_a - 1\,1^\top/3\right\| = \rho$. In this case, we have

$$d_c = 1 \geq \sqrt{\frac{3-\sqrt{5}}{2}}\frac{1}{\sqrt{1-\rho}}.$$

Therefore, for any $\rho \in (0,1)$, there exists a communication matrix W satisfying Assumption 3, with $\left\|W - 1\,1^\top/m\right\| = \rho$, such that the number of communication rounds required to obtain $f(\hat{x}) - f(x^\star) \leq \epsilon$ is

$$\Omega\left(\sqrt{\frac{\beta}{\mu\,(1-\rho)}}\log\left(\frac{\mu\left\|x^\star\right\|^2}{\epsilon}\right)\right).$$

$\square$

B ADDITIONAL NUMERICAL RESULTS

This section complements Sec. 4 of the main paper, providing additional numerical results and details on the experiments presented therein. Specifically, we consider the following problems and data sets:

- **Sec. B.2** studies hinge-loss minimization on `MNIST` (Deng, 2012) and `HIGGS` (Chang and Lin, 2011, from `LIBSVM`) datasets;
- **Sec. B.3** considers logistic regression problems on the `SUSY` dataset (Chang and Lin, 2011, from `LIBSVM`).

B.1 Setting and Algorithm Tuning

The setting of the experiments is the same as the one described in Sec. 4, expect of course for the instance of (P).

The tuning of the simulated algorithms follows the instructions as in the associated papers. To set the free parameters, an estimate of the smoothness constants L_i's and strong-convexity constants μ_i's is needed. We use the following bounds for the hinge and logistic losses. In both cases, the local losses have the following structure:

$$f_i(x) = \frac{1}{n}\sum_{j=1}^{n} \ell\left(b_i^j \cdot \left\langle x, a_i^j \right\rangle\right) + \frac{\lambda}{2}\|x\|^2,$$

where $a_i^j \in \mathbb{R}^d$ are the feature vectors and $b_i^j \in \{-1, 1\}$ are the associated labels. The Hessian matrix of f_i is

$$\nabla^2 f_i(x) = \frac{1}{n}\sum_{j=1}^{n} \ell''\left(b_i^j \left\langle x, a_i^j \right\rangle\right)(b_i^j)^2 a_i^j (a_i^j)^\top + \lambda I,$$

where ℓ'' denotes the second derivative of the loss $\ell : \mathbb{R} \to \mathbb{R}$. Under the assumption that $\ell'' \leq C_\ell$,

$$\lambda I \preceq \nabla^2 f_i(x) \preceq \frac{1}{n}\sum_{j=1}^{n} C_\ell (b_i^j)^2 a_i^j (a_i^j)^T + \lambda I \triangleq H_i.$$

In particular, $C_\ell = 1$ for the smooth hinge loss and $C_\ell = 1/4$ for the logistic loss.

Based on H_i above, we use λ as estimate of μ_i and the largest eigenvalue of H_i as that of L_i. Furthermore, for the smooth constant L of the average loss f we use the overestimate $\hat{L} \triangleq \frac{1}{m}\sum_{i=1}^{m} L_i \geq L$; and for β we use $\hat{\beta} \triangleq \max_{i\in[m]} \left\|H_i - \frac{1}{m}\sum_{i=1}^{m} H_i\right\|$.

The other tuning of the algorithms is as follows:

- `ACC-SONATA-F`: The inner iterations T of `SONATA-F` and `SONATA-L` are set to $\left\lceil \frac{7}{5} \cdot \log \frac{\hat{L}}{\mu} \right\rceil$ and $\left\lceil \log \frac{\hat{\beta}}{\mu} \right\rceil$, respectively;
- `Mudag`: the number of inner loops is set to $\left\lceil \frac{1}{5\sqrt{1-\rho}} \log \frac{L_{\text{mx}}}{\mu} \right\rceil$;
- `ACC-EXTRA`: the number inner loops is set to $\left\lceil \frac{1}{5(1-\rho)} \log \frac{L_{\text{mx}}}{\mu(1-\rho)} \right\rceil$;
- `APM-C`: the number of inner loops is set to $T_k = \left\lceil \frac{k\sqrt{\mu/L_{\text{mx}}}}{100(\sqrt{1-\rho})} \right\rceil$;
- `OPAPC`: This is a single-loop distributed algorithm and all parameters are set according to the instructions in Kovalev et al. (2020).

Whenever the subproblems of the agents do not have a closed form solution, the gradient algorithm is employed, and terminated when an accuracy of 10^{-10} is reached on the Euclidean distance between variables of two consecutive iterations.

We are now ready to describe our experiments.

B.2 Hinge Loss Minimization

Consider the following instance of (P):

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} \frac{1}{n}\sum_{j=1}^{n} \ell_s(b_i^j \cdot \left\langle x, a_i^j \right\rangle) + \frac{\lambda}{2}\|x\|^2,$$

where ℓ_s is the smooth hinge loss, defined as:

$$\ell_s(t) = \begin{cases} 0 & t > 1, \\ \frac{1}{2}(t-1)^2 & t \in [0,1], \\ \frac{1}{2} - t & t < 0. \end{cases}$$

We consider two datasets for the above problem, namely the MNIST and the HIGGS. We use the label 1 for images of digit 4 and −1 for the others. Results are summarized in Fig. 3 (MNIST) and Fig. 4 (HIGGS).

Specifically, Fig. 3 (resp. Fig. 4)-**left-panel** plots the optimality gap $\frac{1}{m}\sum_{i=1}^{m}\|x_i^k - x_{\text{op}}\|^2$ versus the total number of communications, achieved by the different algorithms, where x_{op} is the optimal solution of the problem (estimated running the gradient algorithm up to a precision of 10^{-8} on the norm gradient). In the **mid-panel**, we plot the number of communications to drive the optimality gap below 10^{-4} versus the total sample size; we consider four sizes, namely: 1.8×10^4, 3×10^4, 4.8×10^4 and 6×10^4 for the MNIST dataset and 1.2×10^5, 2.4×10^5, 4.8×10^5 and 9×10^5 for the HIGGS dataset. The **righ-panel** shows the same results for ACC-SONATA and OPAPC on a re-scaled y-axes, to highlight the decreasing number of communications with the local sample size. These results confirm that ACC-SONATA-F and OPAPC consistently outperform the other methods. Notice that, differently ACC-SONATA, OPAPC is applicable only to smooth, unconstrained instances of (P) (i.e., with $r \equiv 0$).

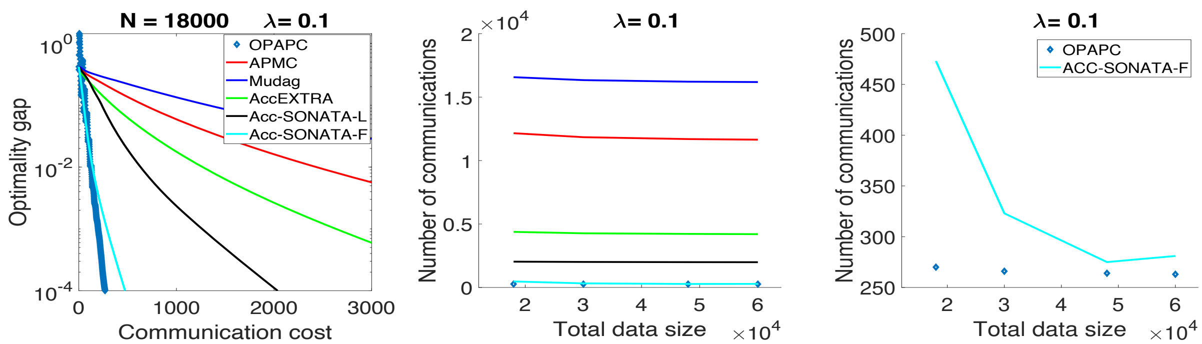

Figure 3: Hinge loss minimization, MNIST dataset. **(left panel)**: optimality gap versus total number of communications; **(mid panel)**: number of communications to reach a precision of 10^{-4} versus (total) sample; **(right panel)**: the mid panel on a different scale of the y-axes.

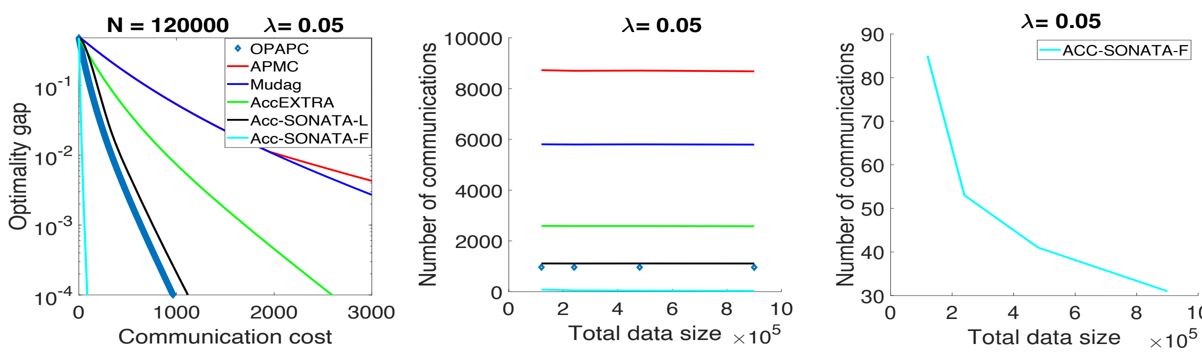

Figure 4: Hinge loss minimization, HIGGS dataset. **(left panel)**: optimality gap versus total number of communications; **(mid panel)**: number of communications to reach a precision of 10^{-4} versus (total) sample; **(right panel)**: the mid panel on a different scale of the y-axes.

B.3 Logistic Regression

We consider here a logistic regression problem on SUSY dataset:

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}\ell_r\left(b_i^j \cdot \left\langle x, a_i^j\right\rangle\right) + \frac{\lambda}{2}\|x\|^2, \tag{16}$$

where $\ell_r = \log(1 + e^{-t})$.

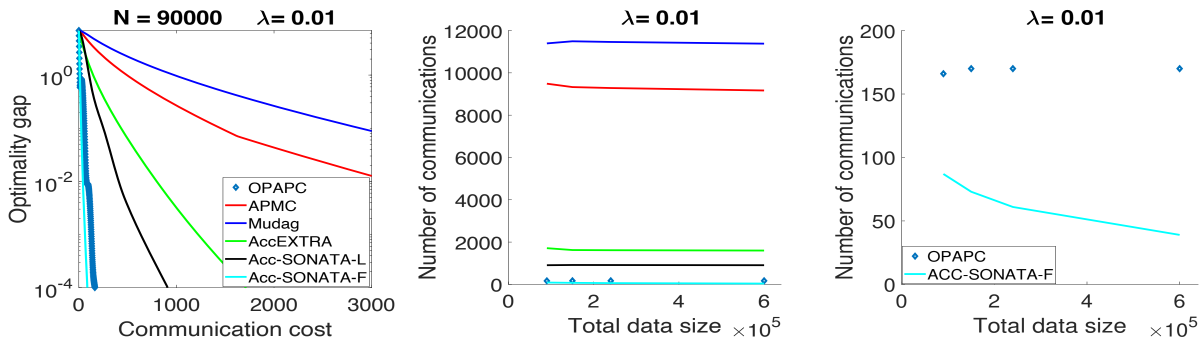

Figure 5: Logistic regression, `SUSY` dataset. **(left panel)**: optimality gap versus total number of communications; **(mid panel)**: number of communications to reach a precision of 10^{-4} versus (total) sample; **(right panel)**: the mid panel on a different scale of the y-axes.

Fig. 5 (**left-panel**) is a plot of optimality gap $\frac{1}{m}\sum_{i=1}^{m}\|x_i^k - x_{\text{op}}\|^2$ versus total number of communications. In the **mid-panel**, we plot the number of communications to obtain optimality gap 10^{-4} versus the size of total data samples, namely: 9×10^4, 1.5×10^5, 2.4×10^5 and 6×10^5. The **righ-panel** shows the same results for `ACC-SONATA` and `OPAPC` on a re-scaled y-axes. Consistently with the other results, also for this class of (nonquadratic) problems, `ACC-SONATA` and `OPAPC` exhibit favorable performance.

C PROOF OF THEOREM 4 AND THEOREM 5

C.1 Sketch of the Proof

As discussed in Sec. 3 [see $(\texttt{S.1})'$ and $(\texttt{S.2})'$], `ACC-SONATA` can be interpreted as an accelerated version of the inexact proximal point algorithm. The challenge here is finding a suitable notion of inexactness for proximal operations that captures all consensus errors (on x, y, z-variables) while being implementable in the distributed setting and, at the same time, retains (for the outer loop) the convergence rate, α, of the exact accelerated proximal method. Our path towards this goal consists in the following two steps:

- **Step 1 (inexactness and outer-loop convergence):** We introduce an inner termination condition for the `SONATA` algorithm [see (21), Sec. C.2], serving as inexact notion of approximate proximal solution. This criterion hinges on a proper potential function controlling the decrease of the optimality gap up to consensus/tracking errors. Roughly speaking, this quantifies the amount of errors in the minimization of u_k [see (6)] that can be tolerated to preserve the convergence of the outer loop at the desired rate $\alpha = \sqrt{\frac{\mu}{\mu+\delta}}$. Convergence of the outer loop at such a rate is established by introducing a proper potential function [see (22), Sec. C.2]. Such a potential function certifies linear convergence of the optimality gap

$$\Delta(x^k) = \max\left(\frac{1}{m}\sum_{i=1}^{m} u(x_i^k) - u^\star, \frac{1}{m}\sum_{i=1}^{m}\left\|x_i^k - \bar{x}^k\right\|^2\right)$$

  at rate α. In the setting of Theorem 4 (i.e., $\delta = \beta - \mu$) and Theorem 5 (i.e., $\delta = L - \mu$), α reads

$$\alpha = \sqrt{\frac{\mu}{\beta}} \quad \text{and} \quad \alpha = \sqrt{\frac{1}{\kappa}}, \tag{17}$$

  respectively.

- **Step 2 (inner-loop convergence):** We introduce a refined analysis of the `SONATA` algorithm based on a new potential function certifying that the inner termination criterion defined in **Step 1** is met in $T = \tilde{\mathcal{O}}(1)$ number of iterations.

Combining **Step 1** and **Step 2**, we can then conclude that ACC-SONATA achieves an ε-solution of Problem (P) in

$$\mathcal{O}\left(T\frac{1}{\alpha}\,\log\frac{1}{\varepsilon}\right)=\widetilde{\mathcal{O}}\left(\frac{1}{\alpha}\,\log\frac{1}{\varepsilon}\right)=\begin{cases}\widetilde{\mathcal{O}}\left(\sqrt{\frac{\beta}{\mu}}\,\log\frac{1}{\varepsilon}\right), & \text{if } \delta=\beta-\mu \text{ (Theorem 4)},\\ \widetilde{\mathcal{O}}\left(\sqrt{\kappa}\,\log\frac{1}{\varepsilon}\right), & \text{if } \delta=L-\mu \text{ (Theorem 5)},\end{cases}$$

total number of communications. The formal proof of **Step 1** and **Step 2** is given in Sec. C.2 and Sec. C.3, respectively.

Notation: Before detailing the two steps, we introduce some notation used throughout the proofs. For each $i\in[m]$, we denote by $x_i^{k,t}$ and $y_i^{k,t}$ the decision variable and tracking variable of agent i after $t=0,\ldots,T-1$ inner iterations in the k-th outer iteration, respectively. Clearly, it must be

$$x_i^{k,0}=x_i^k,\quad y^{k,0}=y_i^k+\delta\left(z_i^{k-1}-z_i^k\right),\quad x_i^{k,T}=x_i^{k+1},\quad\text{and}\quad y^{k,T}=y_i^{k+1}.$$

Furthermore, we define

$$x^\star\triangleq\operatorname*{argmin}_{x\in\mathbb{R}^d}u(x),\quad x^{k+1,\star}=\operatorname*{argmin}_{x\in\mathbb{R}^d}u_k(x),\quad u_k^\star=\min_{x\in\mathbb{R}^d}u_k(x).$$

For any vector $x=[x_1^\top,\ldots,x_m^\top]^\top$, we use $\bar{x}$ to denote the average of its d-dimensional blocks x_i's, that is, $\bar{x}=\frac{1}{m}\sum_{i=1}^m x_i$.

Consensus and tracking errors associated with the iterates $x_i^{k,t}$ and $y_i^{k,t}$ are defined as

$$\left\|x_\perp^{k,t}\right\|^2=\frac{1}{m}\sum_{i\in[m]}\left\|x_i^{k,t}-\bar{x}^{k,t}\right\|^2\quad\text{and}\quad\left\|y_\perp^{k,t}\right\|^2=\frac{1}{m}\sum_{i\in[m]}\left\|y_i^{k,t}-\bar{x}^{y,t}\right\|^2,\tag{18}$$

respectively.

C.2 Step 1: Inexactness and Outer Loop Convergence

Our inexact notion of approximate proximal solution of the minimization of u_k [see (6)] is defined in terms of the (average) optimality gap,

$$g^{k,t}\triangleq\frac{1}{m}\sum_{i=1}^m\left(u_k(x_i^{k,t})-u_k^\star\right),\quad k=0,1,\ldots,\ t=0,\ldots,T,\tag{19}$$

and the consensus and tracking errors (18), captured by

$$e^{k,t}\triangleq c_x\left\|x_\perp^{k,t}\right\|^2+c_y\left\|y_\perp^{k,t}\right\|^2,\quad k=0,1,\ldots,\ t=0,\ldots,T,\tag{20}$$

where $c_x,c_y>0$ are suitably defined universal constants. Specifically, under the following event

$$g^{k,T}+e^{k,T}\le\epsilon^{k+1},\quad k=0,1,\ldots,\tag{21}$$

with $\{\epsilon^k\}$ being a suitably defined geometrically-vanishing positive sequence, we establish linear decay of the following potential function along the outer-loop iterates of ACC-SONATA:

$$P^k\triangleq\frac{1}{m}\sum_{j=1}^m\left(u(x_j^k)-u^\star\right)+\frac{1}{m}\sum_{j=1}^m\frac{\mu}{2}\left\|x_j^{k-1}+\frac{1}{\alpha}(x_j^k-x_j^{k-1})-x^\star\right\|^2+e^{k-1,T},\quad k=0,1,\ldots,\tag{22}$$

where we set $e^{-1,T}=c_y\left\|y_\perp^0\right\|^2$ and $x_i^{-1}=0$, $\forall i$.

In **Step 2** we will show that (21) can be met by running SONATA for $T=\tilde{\mathcal{O}}(1)$ iterations.

Proposition 8. *Consider problem* (P) *under Assumption 1, with optimal objective value* $u^\star$. *Let* $\{(x_i^k, z_i^k)_{i\in[m]}\}$ *be the sequence generated by* `ACC-SONATA` *under (21), with error sequence* $\epsilon^k = P^0\,(1-c\alpha)^k$, $k = 1, \ldots,$ *where* c *is any given constant in* $(0,1)$. *Then, the potential function* P^k *in (22) satisfies:*

$$P^k \le c_2\, P^0\,(1 - c\cdot\alpha)^k, \tag{23}$$

with

$$c_2 = \frac{(2+\sqrt{c_1})^2}{\left(\sqrt{\frac{1-c\cdot\alpha}{1-\alpha}}-1\right)^2(1-\alpha)} \quad \text{and} \quad c_1 = 1 + \frac{\delta}{c_x}\frac{\frac{3}{2}(1-c\alpha)^2 + 5 - 4c\,\alpha}{(1-c\alpha)^2}. \tag{24}$$

Therefore,

$$\max\left(\frac{1}{m}\sum_{i=1}^m u(x_i^k) - u^\star, \frac{1}{m}\sum_{i=1}^m \left\|x_i^k - \bar{x}^k\right\|^2\right) = \mathcal{O}\left((1-c\cdot\alpha)^k\right).$$

Proof. We provide a constructive proof for the choice of the error sequence ϵ^k in (21) and potential function (22), yielding (23).

Using the definition of u_k, we have: for all $x_j \in \operatorname{dom} r$,

$$\begin{aligned}
&\frac{1}{m}\sum_{j=1}^m u(x_j^{k+1}) = \frac{1}{m}\sum_{j=1}^m u_k(x_j^{k+1}) - \frac{\delta}{2m^2}\sum_{i,j}\left\|x_j^{k+1} - z_i^k\right\|^2 \overset{(21)}{\le} u_k^\star + \epsilon^{k+1} - e^{k,T} - \frac{\delta}{2m^2}\sum_{i,j}\left\|x_j^{k+1} - z_i^k\right\|^2\\
&\le \frac{1}{m}\sum_{j=1}^m\left(u_k(x_j) - \frac{\mu+\delta}{2}\left\|x_j - x^{k+1,\star}\right\|^2\right) + \epsilon^{k+1} - e^{k,T} - \frac{\delta}{2m^2}\sum_{i,j}\left\|x_j^{k+1} - z_i^k\right\|^2\\
&= \frac{1}{m}\sum_{j=1}^m u_k(x_j) - \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x_j^{k+1}\right\|^2 + \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x_j^{k+1}\right\|^2 - \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x^{k+1,\star}\right\|^2\\
&\quad + \epsilon^{k+1} - e^{k,T} - \frac{\delta}{2m^2}\sum_{i,j}\left\|x_j^{k+1} - z_i^k\right\|^2\\
&= \frac{1}{m}\sum_{j=1}^m u(x_j) + \frac{1}{m^2}\sum_{i,j}\frac{\delta}{2}\left\|x_j - z_i^k\right\|^2 - \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x_j^{k+1}\right\|^2 + \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x_j^{k+1}\right\|^2\\
&\quad - \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x^{k+1,\star}\right\|^2 + \epsilon^{k+1} - e^{k,T} - \frac{\delta}{2m^2}\sum_{i,j}\left\|x_j^{k+1} - z_i^k\right\|^2\\
&= \frac{1}{m}\sum_{j=1}^m u(x_j) - \frac{\mu}{2m}\sum_{j=1}^m\left\|x_j - x_j^{k+1}\right\|^2 + \frac{\delta}{2m}\sum_{j=1}^m\left(\frac{1}{m}\sum_{i=1}^m\left\|x_j - z_i^k\right\|^2 - \left\|x_j - x_j^{k+1}\right\|^2 - \frac{1}{m}\sum_{i=1}^m\left\|x_j^{k+1} - z_i^k\right\|^2\right)\\
&\quad + \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x_j^{k+1}\right\|^2 - \frac{1}{m}\sum_{j=1}^m\frac{\mu+\delta}{2}\left\|x_j - x^{k+1,\star}\right\|^2 + \epsilon^{k+1} - e^{k,T}\\
&\le \frac{1}{m}\sum_{j=1}^m u(x_j) - \frac{\mu}{2m}\sum_{j=1}^m\left\|x_j - x_j^{k+1}\right\|^2 + \frac{\delta}{m}\sum_{j=1}^m\left\langle x_j - x_j^{k+1}, x_j^{k+1} - \bar{z}^k\right\rangle\\
&\quad - \frac{1}{m}\sum_{j=1}^m(\mu+\delta)\left\langle x_j - x_j^{k+1}, x_j^{k+1} - x^{k+1,\star}\right\rangle + \epsilon^{k+1} - e^{k,T}.
\end{aligned}$$

Setting $x_j = x^\star$, $j \in [m]$, leads to

$$\begin{aligned}
\frac{1}{m}\sum_{j=1}^m u(x_j^{k+1}) &\le u^\star - \frac{\mu}{2m}\sum_{j=1}^m\left\|x^\star - x_j^{k+1}\right\|^2 + \frac{\delta}{m}\sum_{j=1}^m\left\langle x^\star - x_j^{k+1}, x_j^{k+1} - \bar{z}^k\right\rangle\\
&\quad - \frac{1}{m}\sum_{j=1}^m(\mu+\delta)\left\langle x^\star - x_j^{k+1}, x_j^{k+1} - x^{k+1,\star}\right\rangle + \epsilon^{k+1} - e^{k,T}.
\end{aligned} \tag{25}$$

Similarly, setting $x_j = x_j^k$, we have

$$\begin{aligned}\frac{1}{m}\sum_{j=1}^{m} u(x_j^{k+1}) \leq \frac{1}{m}\sum_{j=1}^{m} u(x_j^{k}) - \frac{\mu}{2m}\sum_{j=1}^{m}\left\|x_j^k - x_j^{k+1}\right\|^2 + \frac{\delta}{m}\sum_{j=1}^{m}\left\langle x_j^k - x_j^{k+1}, x_j^{k+1} - \bar{z}^k\right\rangle \\ - \frac{1}{m}\sum_{j=1}^{m}(\mu+\delta)\left\langle x_j^k - x_j^{k+1}, x_j^{k+1} - x^{k+1,\star}\right\rangle + \epsilon^{k+1} - e^{k,T}.\end{aligned} \tag{26}$$

Define the optimality gap at the beginning of the k-th outer iteration, pertaining to the minimization of the original objective function $u(x)$ as:

$$p^k \triangleq \frac{1}{m}\sum_{j=1}^{m} u(x_j^k) - u^\star.$$

Then, multiplying (25) by α and (26) by $(1-\alpha)$, and suming the obtained equations, yields

$$\begin{aligned} p^{k+1} &\leq (1-\alpha)p^k - \frac{\alpha\mu}{2m}\sum_{j=1}^{m}\left\|x^\star - x_j^{k+1}\right\|^2 - \frac{(1-\alpha)\mu}{2m}\sum_{j=1}^{m}\left\|x_j^k - x_j^{k+1}\right\|^2 \\ &\quad + \frac{\delta}{m}\sum_{j=1}^{m}\left\langle \alpha x^\star + (1-\alpha)x_j^k - x_j^{k+1}, x_j^{k+1} - \bar{z}^k\right\rangle \\ &\quad - \frac{\mu+\delta}{m}\sum_{j=1}^{m}\left\langle \alpha x^\star + (1-\alpha)x_j^k - x_j^{k+1}, x_j^{k+1} - x^{k+1,\star}\right\rangle + \epsilon^{k+1} - e^{k,T} \\ &= (1-\alpha)p^k - \frac{\alpha\mu}{2m}\sum_{j=1}^{m}\left\|x^\star - x_j^{k+1}\right\|^2 - \frac{(1-\alpha)\mu}{2m}\sum_{j=1}^{m}\left\|x_j^k - x_j^{k+1}\right\|^2 - \frac{\delta}{m}\sum_{j=1}^{m}\left\|x_j^{k+1} - \bar{z}^k\right\|^2 \\ &\quad + \frac{\delta}{m}\sum_{j=1}^{m}\left\langle \alpha x^\star + (1-\alpha)x_j^k - \bar{z}^k, x_j^{k+1} - \bar{z}^k\right\rangle \\ &\quad - \frac{\mu+\delta}{m}\sum_{j=1}^{m}\left\langle \alpha x^\star + (1-\alpha)x_j^k - x_j^{k+1}, x_j^{k+1} - x^{k+1,\star}\right\rangle + \epsilon^{k+1} - e^{k,T}. \end{aligned} \tag{27}$$

The above is an approximate descent on p^k, up to the error term $\epsilon^{k+1} - e^{k,T}$ and the two inner-product terms. We proceed working first on the term $\left\langle \alpha x^\star + (1-\alpha)x_j^k - \bar{z}^k, x_j^{k+1} - \bar{z}^k\right\rangle$. Define

$$v_j^k \triangleq x_j^{k-1} + \frac{1}{\alpha}(x_j^k - x_j^{k-1}).$$

We have

$$\begin{aligned} v_j^{k+1} &= x_j^k + \frac{1}{\alpha}(x_j^{k+1} - x_j^k) = \left(1 - \frac{1}{\alpha}\right)x_j^k + \frac{1}{\alpha}z_j^k + \frac{1}{\alpha}(x_j^{k+1} - z_j^k) \\ &= \alpha z_j^k + \frac{1+\alpha}{\alpha}(z_j^k - x_j^k) - (1+\alpha)z_j^k + 2x_j^k + \frac{1}{\alpha}(x_j^{k+1} - z_j^k) \\ &= \alpha z_j^k + \frac{1+\alpha}{\alpha}(z_j^k - x_j^k) + (1-\alpha)x_j^{k-1} + \frac{1}{\alpha}(x_j^{k+1} - z_j^k) \\ &= \alpha z_j^k + (1-\alpha)v_j^k + \frac{1}{\alpha}(x_j^{k+1} - z_j^k) = \alpha x_j^{k+1} + (1-\alpha)v_j^k + \left(\frac{1}{\alpha} - \alpha\right)\left(x_j^{k+1} - z_j^k\right). \end{aligned}$$

As byproduct of the above derivation, we also have

$$\alpha z_j^k + (1-\alpha)v_j^k = \frac{1}{\alpha}\left(z_j^k - (1-\alpha)x_j^k\right).$$

Therefore,

$$
\begin{aligned}
\frac{\mu}{2}\left\|v_j^{k+1}-x^\star\right\|^2 &= \frac{\mu}{2}\left\|\alpha x_j^{k+1}+(1-\alpha)v_j^k-x^\star\right\|^2+\frac{\mu}{2}\left(\frac{1}{\alpha}-\alpha\right)^2\left\|x_j^{k+1}-z_j^k\right\|^2\\
&\quad+\mu\left(\frac{1}{\alpha}-\alpha\right)\left\langle \alpha x_j^{k+1}+(1-\alpha)v_j^k-x^\star, x_j^{k+1}-z_j^k\right\rangle\\
&=\frac{\mu}{2}\left\|\alpha x_j^{k+1}+(1-\alpha)v_j^k-x^\star\right\|^2+\frac{\mu}{2}\left(\frac{1}{\alpha}-\alpha\right)^2\left\|x_j^{k+1}-z_j^k\right\|^2+\mu\left(\frac{1}{\alpha}-\alpha\right)\alpha\left\|x_j^{k+1}-z_j^k\right\|^2\\
&\quad+\mu\left(\frac{1}{\alpha}-\alpha\right)\left\langle \alpha z_j^{k}+(1-\alpha)v_j^k-x^\star, x_j^{k+1}-z_j^k\right\rangle\\
&\le\frac{(1-\alpha)\mu}{2}\left\|v_j^k-x^\star\right\|^2+\frac{\alpha\mu}{2}\left\|x_j^{k+1}-x^\star\right\|^2+\frac{\delta^2+2\mu\delta}{2(\mu+\delta)}\left\|x_j^{k+1}-z_j^k\right\|^2\\
&\quad+\frac{\mu}{\alpha}\left(\frac{1}{\alpha}-\alpha\right)\left\langle z_j^k-(1-\alpha)x_j^k-\alpha x^\star, x_j^{k+1}-z_j^k\right\rangle.
\end{aligned}
\tag{28}
$$

Combining (27) and (28), we obtain the decay of the potential function P^k defined in (22):

$$
\begin{aligned}
P^{k+1} &= p^{k+1}+\frac{1}{m}\sum_{j=1}^m\frac{\mu}{2}\left\|v_j^{k+1}-x^\star\right\|^2+e^{k,T}\\
&\le(1-\alpha)\left(p^k+\frac{1}{m}\sum_{j=1}^m\frac{\mu}{2}\left\|v_j^k-x^\star\right\|^2\right)+\frac{\delta}{m}\sum_{j=1}^m\left(\left\|z_j^k-\bar z^k\right\|^2+2\left\langle x_j^{k+1}-\bar x^{k+1},\bar z^k-z_j^k\right\rangle\right)\\
&\quad+\underbrace{\left(\mu-\frac{\mu+\delta}{2}-\frac{\mu^2}{2(\mu+\delta)}\right)}_{\le 0}\frac{1}{m}\sum_{j=1}^m\left\|x_j^{k+1}-z_j^k\right\|^2\\
&\quad+\frac{\delta}{m}\sum_{j=1}^m\left(\left\langle z_j^k-\bar z^k,x_j^{k+1}-\bar x^{k+1}\right\rangle+(1-\alpha)\left\langle z_j^k-\bar z^k,x_j^k-\bar x^k\right\rangle-\left\|z_j^k-\bar z^k\right\|^2\right)\\
&\quad-\frac{\mu+\delta}{m}\sum_{j=1}^m\left\langle\alpha x^\star+(1-\alpha)x_j^k-x_j^{k+1},x_j^{k+1}-x^{k+1,\star}\right\rangle+\epsilon^{k+1}\\
&\le(1-\alpha)\left(p^k+\frac{1}{m}\sum_{j=1}^m\frac{\mu}{2}\left\|v_j^k-x^\star\right\|^2\right)+\frac{\delta}{m}\sum_{j=1}^m\left((1-\alpha)\left\langle z_j^k-\bar z^k,x_j^k-\bar x^k\right\rangle-\left\langle z_j^k-\bar z^k,x_j^{k+1}-\bar x^{k+1}\right\rangle\right)\\
&\quad-\frac{\mu+\delta}{m}\sum_{j=1}^m\left\langle\alpha x^\star-\alpha v_j^{k+1},x_j^{k+1}-x^{k+1,\star}\right\rangle+\epsilon^{k+1}\\
&\le(1-\alpha)\underbrace{\left(p^k+\frac{1}{m}\sum_{j=1}^m\frac{\mu}{2}\left\|v_j^k-x^\star\right\|^2+e^{k-1,T}\right)}_{=P^k}+\frac{\delta\alpha}{m}\left\|v_\perp^k\right\|\left\|z_\perp^k\right\|\\
&\quad+\alpha\frac{\mu+\delta}{m}\sum_{j=1}^m\left\|x^\star-v_j^{k+1}\right\|\left\|x_j^{k+1}-x^{k+1,\star}\right\|+\epsilon^{k+1}\\
&\le(1-\alpha)P^k+\frac{\delta}{1+\alpha}\frac{1}{m}\left(\left\|x_\perp^{k+1}\right\|+(1-\alpha)\left\|x_\perp^k\right\|\right)\left(2\left\|x_\perp^k\right\|+(1-\alpha)\left\|x_\perp^{k-1}\right\|\right)\\
&\quad+\alpha(\mu+\delta)\sqrt{\frac{1}{m}\sum_{j=1}^m\left\|x^\star-v_j^{k+1}\right\|^2}\sqrt{\frac{1}{m}\sum_{j=1}^m\left\|x_j^{k+1}-x^{k+1,\star}\right\|^2}+\epsilon^{k+1}
\end{aligned}
$$

$$\leq (1-\alpha)\, P^k + \frac{\delta}{1+\alpha}\,\frac{1}{m}\left(\frac{3}{2}\left\|x_\perp^{k+1}\right\|^2 + \left(\frac{7}{2} - 3\alpha + \frac{\alpha^2}{2}\right)\left\|x_\perp^{k}\right\|^2 + (1-\alpha)^2\left\|x_\perp^{k-1}\right\|^2\right)$$
$$+ \sqrt{2}\alpha\sqrt{\mu+\delta}\sqrt{\frac{1}{m}\sum_{j=1}^m \left\|x^\star - v_j^{k+1}\right\|^2}\cdot\sqrt{g^{k,T}} + \epsilon^{k+1}.$$

We proceed now to bound $g^{k,T}$ and the consensus error terms by ϵ^{k+1}. For the former we readily have $g^{k,T} \leq \epsilon^{k+1}$, due to (21). For the latter, we choose $\{\epsilon^k\}$ as

$$\epsilon^{k+1} \triangleq \epsilon^k\,(1-c\alpha)\,, \quad k = 0,1,\ldots, \tag{29}$$

where c is any constant in $(0,1)$ and ϵ^0 is to be determined; and use $e^{k,T} \leq \epsilon^{k+1}$, for any $k = 0,1\ldots$ (still due to (21)). We can write

$$P^{k+1} \leq (1-\alpha)P^k + \sqrt{2\mu}\sqrt{\frac{1}{m}\sum_{j=1}^m \left\|x^\star - v_j^{k+1}\right\|^2}\cdot\sqrt{\epsilon^{k+1}} + c_1\epsilon^{k+1},$$

where

$$c_1 = 1 + \frac{\delta}{c_x}\,\frac{\frac{3}{2}(1-c\alpha)^2 + 5 - 4c\,\alpha}{(1-c\alpha)^2}.$$

Define $\lambda^k \triangleq (1-\alpha)^k$; we have

$$\frac{P^{k+1}}{\lambda^{k+1}} \leq \frac{P^k}{\lambda^k} + \sqrt{2\mu}\sqrt{\frac{1}{m}\sum_{j=1}^m \left\|x^\star - v_j^{k+1}\right\|^2}\cdot\frac{\sqrt{\epsilon^{k+1}}}{\lambda^{k+1}} + c_1\frac{\epsilon^{k+1}}{\lambda^{k+1}}$$
$$\leq P^0 + c_1\sum_{t=1}^{k+1}\frac{\epsilon^t}{\lambda^t} + \sum_{t=1}^{k+1}\frac{\sqrt{2\mu\epsilon^t}}{\lambda^t}\sqrt{\frac{1}{m}\sum_{j=1}^m \left\|x^\star - v_j^{k+1}\right\|^2}. \tag{30}$$

Introducing the following quantities:

$$\hat{v}^k \triangleq \sqrt{\frac{\mu}{2\lambda^k}}\cdot\sqrt{\frac{1}{m}\sum_{j=1}^m \left\|v_j^k - x^\star\right\|^2}, \quad a^k \triangleq 2\sqrt{\frac{\epsilon^k}{\lambda^k}}, \quad \text{and} \quad S^k \triangleq P^0 + c_1\sum_{t=1}^k \frac{\epsilon^t}{\lambda^t}, \tag{31}$$

(30) can be rewritten as

$$(\hat{v}^k)^2 \leq S^k + \sum_{t=1}^k a^t\hat{v}^t. \tag{32}$$

Using (32) we can invoke (Hongzhou et al., 2015, Lemma A.10) and conclude

$$S^k + \sum_{t=1}^k a^t\hat{v}^t \leq \left(\sqrt{S^k} + \sum_{t=1}^k a^t\right)^2. \tag{33}$$

This, together with (30), yields

$$P^k \leq \lambda^k\left(\sqrt{S^k} + \sum_{t=1}^k a^t\right)^2 \overset{(31)}{\leq} \lambda^k\left(\sqrt{L^0} + (2+\sqrt{c_1})\sum_{t=1}^k\sqrt{\frac{\epsilon^t}{\lambda^t}}\right)^2.$$

Choosing $\epsilon^0 = P^0$, resulting in $\epsilon^k = P^0\cdot(1-c\cdot\alpha)^k$ [cf. (29)], leads to the desired result:

$$P^k \leq c_2\,P^0\,(1-c\cdot\alpha)^{k+1},$$

with

$$c_2 = \frac{(2+\sqrt{c_1})^2}{\left(\sqrt{\frac{1-c\cdot\alpha}{1-\alpha}} - 1\right)^2(1-\alpha)}.$$

□

C.3 Step 2: SONATA and Inner-Loop Convergence

Given Proposition 8, to complete the proof of Theorem 4 and Theorem 5, we need to show that the error condition (21) is satisfied if SONATA runs for $T = \tilde{\mathcal{O}}(1)$ iterations; the explicit expression of T depends on the specific setting considered for the algorithm, and it is different for the one specified in Theorem 4 and in Theorem 5–the two cases are studied separately in Sec. C.3.1 and Sec. C.3.2, respectively.

To control the number of calls of SONATA we need to bound $g^{k,0} + e^{k,0}$, which is done in the following lemma.

Lemma 9. *Instate the setting of Proposition 8. For $k \geq 1$, if $g^{k-1,T} + e^{k-1,T} \leq \epsilon^k$, then the following holds:*

$$g^{k,0} + e^{k,0} \leq P^0 \left(2(1 - c\cdot\alpha)^k + \max\left(\frac{1}{\mu+\delta}, 2\,c_y\right) \frac{72\,\delta^2}{\mu}\, c_2\, (1 - c\cdot\alpha)^{k-1} \right). \tag{34}$$

Proof. Since

$$g^{k,0} + e^{k,0} = \frac{1}{m}\sum_{i=1}^{m} \left(u_k(x_i^k) - u_k^\star\right) + c_x \left\|x_\perp^{k,0}\right\|^2 + c_y \left\|y_\perp^{k,0}\right\|^2,$$

we begin bounding $\frac{1}{m}\sum_{i=1}^{m}\left(u_k(x_i^k) - u_k^\star\right)$. Postponing the proof to the end of this section, we show that the following holds

$$\frac{1}{m}\sum\nolimits_{i=1}^{m} u_k(x_i^k) - u_k^\star \leq 2\epsilon^k - 2e^{k-1,T} + \frac{\delta^2}{\mu+\delta}\left\|\bar{z}^k - \bar{z}^{k-1}\right\|^2. \tag{35}$$

Therefore, we can write

$$\begin{aligned}
g^{k,0} + e^{k,0} &\overset{(35)}{\leq} 2\epsilon^k - 2e^{k-1,T} + \frac{\delta^2}{\mu+\delta}\left\|\bar{z}^k - \bar{z}^{k-1}\right\|^2 + c_x\left\|x_\perp^k\right\|^2 + 2\,c_y\left\|y_\perp^k\right\|^2 + 2\,c_y\,\delta^2\left\|(z^{k-1} - z^k)_\perp\right\|^2 \\
&\leq 2\epsilon^k + \max\left(\frac{1}{\mu+\delta}, 2\,c_y\right)\delta^2 \frac{1}{m}\sum_{i\in[m]}\left\|z_i^k - z_i^{k-1}\right\|^2 \\
&\leq 2\epsilon^k + \max\left(\frac{1}{\mu+\delta}, 2\,c_y\right)\frac{72\,\delta^2}{\mu}\frac{P^k}{(1-c\cdot\alpha)^2} \\
&= P^0\left(2(1-c\cdot\alpha)^k + \max\left(\frac{1}{\mu+\delta}, 2\,c_y\right)\frac{72\,\delta^2}{\mu}\,c_2\,(1-c\cdot\alpha)^{k-1}\right).
\end{aligned}$$

It remains to show that (35) holds. We prove it following similar ideas as in (Hongzhou et al., 2015, Lemma B.1), with differences due to the distributed setting. Writing

$$\begin{aligned}
u_k(x) - u_{k-1}(x) - (u_k(z) - u_{k-1}(z)) &= \frac{\delta}{2m}\sum_{i=1}^{m}\left(\left\|x - z_i^k\right\|^2 - \left\|x - z_i^{k-1}\right\|^2 - \left\|z - z_i^k\right\|^2 + \left\|z - z_i^{k-1}\right\|^2\right) \\
&= \delta\left\langle \bar{z}^{k-1} - \bar{z}^k, x - z\right\rangle,
\end{aligned}$$

we have

$$\begin{aligned}
&u_k(x_i^k) - u_k(x^{k+1,\star}) \\
&= u_{k-1}(x_i^k) - u_{k-1}(x^{k+1,\star}) + \delta\left\langle \bar{z}^{k-1} - \bar{z}^k, x_i^k - x^{k+1,\star}\right\rangle \\
&= u_{k-1}(x_i^k) - u_{k-1}(x^{k,\star}) + u_{k-1}(x^{k,\star}) - u_{k-1}(x^{k+1,\star}) + \delta\left\langle \bar{z}^{k-1} - \bar{z}^k, x_i^k - x^{k+1,\star}\right\rangle \\
&\leq u_{k-1}(x_i^k) - u_{k-1}(x^{k,\star}) - \frac{\mu+\delta}{2}\left\|x^{k,\star} - x^{k+1,\star}\right\|^2 + \delta\left\langle \bar{z}^{k-1} - \bar{z}^k, x_i^k - x^{k+1,\star}\right\rangle.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\frac{1}{m}\sum_{i=1}^{m}\left(u_k(x_i^k) - u_k(x^{k+1,\star})\right) \\
&\leq \epsilon^k - e^{k-1,T} - \frac{\mu+\delta}{2}\left\|x^{k,\star} - x^{k+1,\star}\right\|^2 + \delta\left\langle \bar{z}^{k-1} - \bar{z}^k, \bar{x}^k - x^{k+1,\star}\right\rangle.
\end{aligned} \tag{36}$$

At the same time, it holds

$$\begin{aligned}\delta\left\langle \bar{z}^{k-1}-\bar{z}^{k}, x^{k,\star}-x^{k+1,\star}\right\rangle &\leq \frac{\mu+\delta}{2}\left\|x^{k,\star}-x^{k+1,\star}\right\|^2+\frac{\delta^2}{2(\mu+\delta)}\left\|\bar{z}^{k-1}-\bar{z}^{k}\right\|^2,\\ \delta\left\langle \bar{z}^{k-1}-\bar{z}^{k}, \bar{x}^{k}-x^{k,\star}\right\rangle &\leq \frac{\mu+\delta}{2}\left\|\bar{x}^{k}-x^{k,\star}\right\|^2+\frac{\delta^2}{2(\mu+\delta)}\left\|\bar{z}^{k-1}-\bar{z}^{k}\right\|^2.\end{aligned} \tag{37}$$

Combining (36) and (37), leads to

$$\begin{aligned}&\frac{1}{m}\sum_{i=1}^{m}\left(u_k(x_i^k)-u_k(x^{k+1,\star})\right)\leq \epsilon^k-e^{k-1,T}+\frac{\delta^2}{\mu+\delta}\left\|\bar{z}^{k-1}-\bar{z}^{k}\right\|^2+\frac{\mu+\delta}{2}\left\|\bar{x}^{k}-x^{k,\star}\right\|^2\\ &\leq \epsilon^k-e^{k-1,T}+\frac{\delta^2}{\mu+\delta}\left\|\bar{z}^{k-1}-\bar{z}^{k}\right\|^2+\frac{1}{m}\sum_{i=1}^{m}\left(u_{k-1}(x_i^k)-u_{k-1}(x^{k,\star})\right)\\ &\leq 2\epsilon^k-2e^{k-1,T}+\frac{\delta^2}{\mu+\delta}\left\|\bar{z}^{k-1}-\bar{z}^{k}\right\|^2, \qquad \text{for } k\geq 1.\end{aligned}$$

□

C.3.1 Proof of Theorem 4

We begin proving convergence of `SONATA` in the setting of Theorem 4–we refer to such an instance of `SONATA` as `SONATA-F`. Note that convergence established in Sun et al. (2022) is not directly applicable here. First, there is a mismatch between the gradient tracking initialization therein and the one used in our setting. Second, R-linear convergence of the optimality gap as in Sun et al. (2022) is not enough to certify that the termination criterion (21) is satisfied after a finite number of iterations $T=\tilde{\mathcal{O}}(1)$. Our refined convergence analysis of `SONATA-F` is stated in Lemma 10 below.

Notice that, with $\delta=\beta-\mu$, we have that: (i) the objective function $u_k(x)$ is $\mu_{\mathrm{u}_k}=\beta$-similar and β-strongly convex; and (ii) every $f_i^k(x)$ is $L_{\mathrm{mx}}=L+2\beta-\mu$ smooth. By using the surrogates (4), `SONATA-F` takes advantage of similarity and achieves linear convergence rate, scaling with $\mathcal{O}(\beta/\mu_{\mathrm{u}_k})=\mathcal{O}(1)$.

Lemma 10. *Consider the minimization of $u_k(x)$ wherein $\delta=\beta-\mu$, running `SONATA-F` (initialized as in `ACC-SONATA`). With*

$$c_x=\frac{8(L+2\beta-\mu)^2}{\beta}, \quad c_y=\frac{4}{\beta},$$

and the network connectivity ρ satisfying

$$\rho\leq\frac{1}{4\sqrt{1785}}\frac{\beta(2\beta-\mu)}{(L+2\beta-\mu)(L+4\beta-\mu)}, \tag{38}$$

`SONATA-F` converges Q-linearly, that is,

$$g^{k,t+1}+e^{k,t+1}\leq\frac{33}{34}\left(g^{k,t}+e^{k,t}\right). \tag{39}$$

Proof. The proof builds on some intermediate results in Sun et al. (2022); when recalled here, we use the same notation as defined therein.

Consider (Sun et al., 2022, Proposition 3.4), and set therein $\epsilon_{opt}=\frac{1}{2}(2\beta-\mu)$; we get

$$\sigma(1)\leq\frac{16}{17} \quad \text{and} \quad \eta(1)\leq\frac{18}{17\beta}.$$

Therefore,

$$g^{k,t+1}\leq\frac{16}{17}g^{k,t}+\frac{9}{17}e^{k,t}. \tag{40}$$

According to (Sun et al., 2022, Proposition 3.5), we have

$$\left\|x_\perp^{k,t+1}\right\|^2 \le 2\rho^2\left\|x_\perp^{k,t}\right\|^2 + 2\rho^2\frac{1}{m}\left\|d^{k,t}\right\|^2,$$
$$\left\|y_\perp^{k,t+1}\right\|^2 \le 3\rho^2\left\|y_\perp^{k,t}\right\|^2 + 12L_{\rm mx}^2\rho^2\left\|x_\perp^{k,t}\right\|^2 + 3L_{\rm mx}^2\rho^2\frac{1}{m}\left\|d^{k,t}\right\|^2;$$

which leads to

$$4L_{\rm mx}^2\left\|x_\perp^{k,t+1}\right\|^2 + 2\left\|y_\perp^{k,t+1}\right\|^2 \le \rho^2\left(32L_{\rm mx}^2\left\|x_\perp^{k,t}\right\|^2 + 6\left\|y_\perp^{k,t}\right\|^2 + 14L_{\rm mx}^2\frac{1}{m}\left\|d^{k,t}\right\|^2\right). \tag{41}$$

(Sun et al., 2022, Proposition 3.6) becomes

$$\frac{1}{m}\left\|d^{k,t}\right\|^2 \le \frac{6}{\beta}\frac{(4\beta-\mu)^2+4L_{\rm mx}^2}{(2\beta-\mu)^2}g^{k,t} + \frac{3}{(2\beta-\mu)^2}\left\|y_\perp^{k,t}\right\|^2. \tag{42}$$

Combining (40), (41) and (42) leads to

$$\begin{aligned} g^{k,t+1}+e^{k,t+1} \le &\left(\frac{16}{17}+\rho^2\frac{168L_{\rm mx}^2}{\beta^2}\frac{(4\beta-\mu)^2+4L_{\rm mx}^2}{(2\beta-\mu)^2}\right)g^{k,t} \\ &+\left(\frac{9}{17}+\rho^2\max\left(8,\ 3+21\frac{L_{\rm mx}^2}{(2\beta-\mu)^2}\right)\right)e^{k,t}. \end{aligned}$$

It is not difficult to check that contraction of $g^{k,t}+e^{k,t}$ as in (39) holds when ρ satisfies

$$\rho \le \min\left(\sqrt{\frac{15}{272}},\ \sqrt{\frac{15}{34\left(3+\frac{21L_{\rm mx}^2}{(2\beta-\mu)^2}\right)}},\ \sqrt{\frac{\beta^2}{5712L_{\rm mx}^2}\frac{(2\beta-\mu)^2}{(4\beta-\mu)^2+4L_{\rm mx}^2}}\right).$$

A sufficient condition for that is (38). □

Invoking Lemma 9, the number of inner iterations needed for (21) to hold, for any $k=1,2,\ldots$, can be bounded as

$$\begin{aligned} T &\le \left\lceil 34\log\frac{g^{k,0}+e^{k,0}}{\epsilon^{k+1}}\right\rceil \overset{(34)}{\le} \left\lceil 34\log\frac{P^0\left(2(1-c\cdot\alpha)^k+\frac{576(\beta-\mu)^2}{\mu\beta}c_2\,(1-c\cdot\alpha)^{k-1}\right)}{P^0\,(1-c\cdot\alpha)^{k+1}}\right\rceil \\ &= \left\lceil 34\log\frac{2(1-c\cdot\alpha)+\frac{576(\beta-\mu)^2}{\mu\beta}c_2}{(1-c\cdot\alpha)^2}\right\rceil \\ &= \left\lceil 34\log\left(\frac{2}{1-c\cdot\sqrt{\mu/\beta}}+\frac{576(\beta-\mu)^2}{\mu\beta(1-c\cdot\sqrt{\mu/\beta})^2}\frac{\left(2+\sqrt{1+\frac{(\beta-\mu)\beta}{8(L+2\beta-\mu)^2}\frac{\frac{3}{2}(1-c\sqrt{\mu/\beta})^2+5-4c\sqrt{\mu/\beta}}{(1-c\sqrt{\mu/\beta})^2}}\right)^2}{\left(\sqrt{\frac{1-c\cdot\sqrt{\mu/\beta}}{1-\sqrt{\mu/\beta}}}-1\right)^2(1-\sqrt{\mu/\beta})}\right)\right\rceil \\ &= \mathcal{O}\left(\log\frac{\beta}{\mu}\right). \end{aligned} \tag{43}$$

For $k=0$, due to $g^{0,0}+e^{0,0}\le P^0$, we have $T=\left\lceil 34\log\frac{1}{1-c\cdot\alpha}\right\rceil$, which is smaller than the RHS in (43).

This, together with Proposition 8 completes the proof of Theorem 4. □

C.3.2 Proof of Theorem 5

We study now convergence of SONATA in the setting of Theorem 5–we refer to such an instance as SONATA-L.

We begin noticing that, with $\delta = L - \mu$, we have the following properties for $u_k(x)$ and $f_i^k(x)$'s: (i) $u_k(x)$ is $(2L-\mu)$-smooth, β-similar and L-strongly convex; and (ii) every $f_i^k(x)$ is $L_{\text{mx}} = 2L + \beta - \mu$ smooth. Hence, when using the linearization surrogates (5), SONATA-L achieves linear rate, scaling as $\mathcal{O}((2L-\mu)/L) = \mathcal{O}(1)$. We establish such a result below, following the same path as in the proof of Lemma 10.

Lemma 11. *Consider the minimization of $u_k(x)$ wherein $\delta = L - \mu$, running SONATA-L (initialized as in ACC-SONATA). With*

$$c_x = \frac{56(2L+\beta-\mu)^2}{L}, \quad c_y = \frac{28}{L},$$

and the network connectivity ρ satisfying

$$\rho \leq \frac{1}{70\sqrt{15}} \frac{L^2}{(2L-\mu+\beta)^2}, \tag{44}$$

SONATA-L converges Q-linearly, that is,

$$g^{k,t+1} + e^{k,t+1} \leq \frac{9}{10}\left(g^{k,t} + e^{k,t}\right).$$

Proof. We set ϵ_{opt} in (Sun et al., 2022, Proposition 3.4) as $\epsilon_{opt} = \frac{L}{2}$; one gets therein $\sigma(1) \leq \frac{4}{5}$ and $\eta(1) \leq \frac{7}{L}$. Therefore,

$$g^{k,t+1} \leq \frac{4}{5} g^{k,t} + \frac{1}{2} e^{k,t}. \tag{45}$$

In addition, (Sun et al., 2022, Proposition 3.6) becomes

$$\frac{1}{m}\left\|d^{k,t}\right\|^2 \leq \frac{6}{L}\left(\frac{9}{4} + \frac{4L_{\text{mx}}^2}{(2L-\mu)^2}\right) g^{k,t} + \frac{3}{(2L-\mu)^2}\left\|y_{\perp}^{k,t}\right\|^2. \tag{46}$$

Combining (41), (45) and (46), leads to

$$g^{k,t+1} + e^{k,t+1} \leq \left(\frac{4}{5} + 1176\,\rho^2 \frac{L_{\text{mx}}^2}{L^2}\left(\frac{9}{4} + \frac{4L_{\text{mx}}^2}{(2L-\mu)^2}\right)\right) g^{k,t} + \left(\frac{1}{2} + \rho^2 \max\left(8,\ 3 + 21\frac{L_{\text{mx}}^2}{(2L-\mu)^2}\right)\right) e^{k,t}$$

A contraction on $g^{k,t} + e^{k,t}$ is ensured choosing

$$\rho \leq \min\left(\frac{1}{2\sqrt{5}},\ \sqrt{\frac{2}{5\left(3 + 21\frac{L_{\text{mx}}^2}{(2L-\mu)^2}\right)}},\ \frac{1}{28}\sqrt{\frac{1}{15\frac{L_{\text{mx}}^2}{L^2}\left(\frac{9}{4} + \frac{4L_{\text{mx}}^2}{(2L-\mu)^2}\right)}}\right).$$

The above is satisfied if ρ satisfies (44). □

We can conclude the proof of Theorem 4, using Lemma 9 to determine the number of inner iterations needed for (21) to hold: for any $k = 1, 2, \ldots,$ we have

$$T \leq \left\lceil 10 \log \frac{g^{k,0} + e^{k,0}}{\epsilon^{k+1}} \right\rceil \leq \left\lceil 10 \log \frac{2(1 - c\cdot\alpha) + \frac{4032(L-\mu)^2}{L\mu} c_2}{(1 - c\cdot\alpha)^2} \right\rceil = \mathcal{O}(\log \kappa). \tag{47}$$

For $k = 0$, due to $g^{0,0} + e^{0,0} \leq P^0$, we have $T = \left\lceil 10 \log \frac{1}{1 - c\cdot\alpha} \right\rceil$, which is smaller than the value of T in (47) for $k \geq 1$.

D ACC-SONATA OVER STAR-NETWORKS

In this section we customize ACC-SSONATA to the special setting of master/workers architectures. Specifically, consider Problem (P) over a star (unidirected) graph with m nodes, where one of them (the master node) connects with all the others (workers). The workers still own only one function f_i of the sum-cost f in (P). The application of ACC-SSONATA to such a setting boils down to customize the inner algorithm SONATA–we use the instance in (Sun et al., 2022, Algorithm 3), which is reported in Algorithm 3 below (applied to (P)) for convenience. Note that the consensus step is now replaced by the exact average performed by the master node (see (S.4)), based upon reception of the local optimization variables from the workers. Also, there is no need of the gradient-tracking mechanism, as the master node can directly broadcast to the workers the aggregate gradient $\nabla f(x^k)$. Notice that SONATA-star can be interpreted as a special instance of SONATA described in Algorithm 1 (up to a proper initialization) if the weight matrix W therein is set to $W = [1, 0_{m,m-1}][1/m, 0_{m,m-1}]^\top$, whose associated $\rho = \|W - 11^\top/m\| = 0$.

Equipped with SONATA-star, ACC-SONATA-star reduces to Algorithm 4 below. Convergence as discussed in the main text is a consequence of Theorem 4 and Theorem 5. Note that the condition of ρ is trivially satisfied as $\rho = 0$.

Algorithm 3 SONATA-star($\{f_i\}_{i\in[m]}$, $(x_i^0)_{i\in[m]}, T$)

Input: $\{f_i(x)\}_{i\in[m]}$, $r(x)$ [cf. (P)];
$(x_i^0)_{i\in[m]}$ [initialization points],
$T > 0$ [# iterations];
Output: x^T;
for $k = 0, 1, 2, \ldots, T-1$ **do**

(S.1): Each worker i evaluates $\nabla f_i(x^k)$ and sends it to the master node;

(S.2): The master broadcasts $\nabla f(x^k) = 1/m \sum_{i=1}^m \nabla f_i(x^k)$ to the workers;

(S.3): Each worker i computes

$$x_i^{k+1/2} \triangleq \underset{x_i\in\mathbb{R}^d}{\operatorname{argmin}}\ \tilde{f}_i(x_i; x^k) + \left(\nabla f(x^k) - \nabla f_i(x^k)\right)^\top (x_i - x^k) + r(x_i),$$

and sends $x_i^{k+1/2}$ to the master;

(S.4): The master computes the average

$$x^{k+1} = \frac{1}{m}\sum_{i=1}^m x_i^{k+1/2},$$

and sends it back to the workers.

end for

E SOLVING AGENTS' SUBPROBLEMS INEXACTLY

There are applications wherein the local optimization problems in the Algorithm 1

$$x_i^{k+1/2} = \underset{x\in\mathbb{R}^d}{\operatorname{argmin}}\ \tilde{f}_i(x; x_i^k) + \left\langle y_i^k - \nabla f_i(x_i^k), x - x_i^k\right\rangle + r(x),$$

do not have a closed-form solution or cannot be solved efficiently to arbitrary precision, especially when the surrogate function (4) is adopted. In this section, we discuss how to modify ACC-SONATA-F (and by-product ACC-SONATA-L) to accommodate computations of inexact solutions of agents' subproblems. We prove that, by carefully choosing the *inexact criterion* for solving approximately the local optimization subproblems, the communication complexity of the resulting inexact ACC-SONATA-F, termed Inexact ACC-SONATA-F (Algorithm 5), matches that of ACC-SONATA-F as in (9) (see Theorem 12). We also study the computational complexity of Inexact-ACC-SONATA-F (see Theorem 13).

Algorithm 4 `Accelerated SONATA-star`

Input: β, μ, $\delta > 0$, $\alpha = \sqrt{\mu/(\mu+\delta)}$;
$x^0 = z^0 = 0$;
Output: x^K
for $k = 0, 1, 2, \ldots, K-1$ **do**

`Set:` $f_i^k(x) = f_i(x) + \frac{\delta}{2}\left\|x - z^k\right\|^2,$

`(S.1) Inner loop via SONATA-star:`

$$x^{k+1} = \texttt{SONATA-star}\Big(\left\{f_i^k\right\}_{i\in[m]}, x^k, T\Big);$$

`(S.2) Extrapolation step:`

$$z^{k+1} = x^{k+1} + \frac{1-\alpha}{1+\alpha}\,(x^{k+1} - x^k).$$

end for

We begin introducing the inexact instance of the SONATA algorithm, termed `Inexact-SONATA`, and described in Algorithm 5. Notice the presence therein of an additional input sequence $\left\{\xi^k\right\}_{k=0}^{T-1} = \left\{(\xi_i^k)_{i\in[m]}\right\}_{k=0}^{T-1}$ (to be properly chosen), determining the accuracy the solution of the agents' subproblems (48) in `(S.1)` is estimated.

Algorithm 5 `Inexact-SONATA`$(\{f_i\}_{i\in[m]}, x^0, y^0, T, \left\{\xi^k\right\}_{k=0}^{T-1})$

Input: $\{f_i(x)\}_{i\in[m]}$, $r(x)$ [cf. (P)];
$x^0 = (x_i^0)_{i\in[m]}$ [initialization points],
$y^0 = (y_i^0)_{i\in[m]}$ [gradient-tracking initialization],
$T > 0$ [# iterations],
$\left\{\xi^k\right\}_{k=0}^{T-1} = \left\{(\xi_i^k)_{i\in[m]}\right\}_{k=0}^{T-1}$ [inexactness parameters];
Output: $x^T = \left(x_i^T\right)_{i\in[m]}$, $y^T = \left(y_i^T\right)_{i\in[m]}$;
for $k = 0, 1, 2, \ldots, T-1$ **do**
`(S.1) Local computations:` for all $i \in [m]$,

$$\begin{aligned} &\tilde{x}_i^{k+1/2} \approx \underset{x\in\mathbb{R}^d}{\operatorname{argmin}}\ \ u_i^k(x) \triangleq \tilde{f}_i(x; x_i^k) + \left\langle y_i^k - \nabla f_i(x_i^k), x - x_i^k\right\rangle + r(x), \\ &\textbf{s.t.}\quad u_i^k(\tilde{x}_i^{k+1/2}) - \min_{x\in\mathbb{R}^d} u_i^k(x) \le \xi_i^k; \end{aligned} \tag{48}$$

`(S.2) Communications:` for all $i \in [m]$,

$$x_i^{k+1} = \sum_{j=1}^m w_{ij}\tilde{x}_j^{k+1/2},$$

$$y_j^{k+1} = \sum_{j=1}^m w_{ij}\big(y_j^k + \nabla f_j(x_j^{k+1}) - \nabla f_j(x_j^k)\big).$$

end for

Equipped with `Inexact-SONATA` Algorithm, the inexact version of `Acc-SONATA` is presented in Algorithm 6, where in the inner loop is now invoked `Inexact-SONATA`.

The next theorem studies convergence of `Inexact ACC-SONATA`; we focus on `Inexact ACC-SONATA-F`, i.e., the instance of `Inexact ACC-SONATA` using (4) as surrogate functions in the step `(S.1)` of `Inexact SONATA`. With a properly chosen accuracy sequence, we show that `Inexact ACC-SONATA-F` inherits the same communication

Algorithm 6 `Inexact Accelerated SONATA`

Input: β, μ, $\delta > 0$, $\alpha = \sqrt{\mu/(\mu+\delta)}$, $\{\xi^{k,t}\}$;
$x_i^0 = z_i^0 = z_i^{-1} = 0$, $y_i^0 = \nabla f_i(x_i^0)$
Output: $x^K = (x_i^K)_{i\in[m]}$
for $k = 0, 1, 2, \ldots, K-1$ **do**
`Set:` $f_i^k(x) = f_i(x) + \frac{\delta}{2}\left\|x - z_i^k\right\|^2$;
`(S.1) Inner loop via SONATA:`

$$\left(x^{k+1}, y^{k+1}\right) = \texttt{Inexact-SONATA}\Big(\left\{f_i^k\right\}_{i\in[m]}, x^k, y^k + \delta\left(z^{k-1} - z^k\right), T, \left\{\xi^{k,t}\right\}_{t=0}^{T-1}\Big);$$

`(S.2) Extrapolation step:`

$$z_i^{k+1} = x_i^{k+1} + \frac{1-\alpha}{1+\alpha}\left(x_i^{k+1} - x_i^k\right), \quad \forall i \in [m].$$

end for

complexity of its exact counterpart, `ACC-SONATA-F`.

Theorem 12. *Consider problem* (P) *under Assumption 1, with optimal value function* $u^\star$ *and* $\beta > \mu$ *w.l.o.g.. Let* $\{x^k \triangleq (x_i^k)_{i\in[m]}\}$ *be the sequence generated by the Algorithm 6 under Assumption 3, with*

$$\rho \le \mathcal{O}\left(\left(1 + \frac{\kappa - 1}{\beta/\mu}\right)^{-2}\right), \tag{49}$$

and the following tuning:

$$\delta = \beta - \mu, \quad T = \mathcal{O}\left(\log \beta/\mu\right), \quad \xi_i^{k,t} = \mathcal{O}\left(\left(1 - c\sqrt{\mu/\beta}\right)^k (16/17)^t\right), \ \forall\, i \in [m], \tag{50}$$

and agents' surrogate functions (4) *in* `Inexact-SONATA`*. Recall the optimality gap* $\Delta(x^k)$ *defined in* (8)*; then, there holds*

$$\Delta(x^k) = \mathcal{O}\left(\left(1 - c\sqrt{\frac{\mu}{\beta}}\right)^k\right),$$

where $c \in (0,1)$ *is some universal constant. Therefore,* $\Delta(x^K) \le \varepsilon$*,* $\varepsilon > 0$*, in*

$$\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}} \cdot T \cdot \log\frac{1}{\varepsilon}\right) \tag{51}$$

total (inner plus outer) communication steps.

Proof. See Appendix E.1. □

As anticipated, the above result shows that if the subproblems (48) are solved with increasing accuracy, as specified by the decay of $\{\xi^{k,t}\}$ in (50), the total number of communication steps (as in (51)) for `Inexact ACC-SONATA-F` to reach an ε-solution of (P) matches that of `ACC-SONATA-F`, despite the presence of computation errors. We discuss next the computation complexity of `Inexact ACC-SONATA-F`.

Suppose we use a solution method $\mathcal{M}$ to solve the local optimization (48); and let $T^{k,t}$ be the number of iterations required by $\mathcal{M}$ to solve (48) within the precision $\xi^{k,t}$ (at the inner iteration t of the outer step k of `Inexact ACC-SONATA-F`). Then, in the setting of Theorem 12, the total number of steps taken by $\mathcal{M}$ for $\Delta(x^K) \le \varepsilon$ reads $\sum_{k=0}^{K-1}\sum_{t=0}^{T-1} T^{k,t}$. We provide next an explicit expression of such a number when $\mathcal{M}$ is a linearly convergent method.

Theorem 13. *Let $\{x^k \triangleq (x_i^k)_{i\in[m]}\}$ be the sequence generated by the Algorithm 6 under Assumption 3, in the same setting of Theorem 12. Suppose that $\mathcal{M}$ is such that*

$$T^{k,t} = \mathcal{O}\left(\kappa_{\mathcal{M}} \log 1/\xi_i^{k,t}\right),$$

for some $\kappa_{\mathcal{M}} \geq 1$. Then, to reach $\Delta(x^K) \leq \varepsilon$, $\varepsilon > 0$, the total number of iterations taken by $\mathcal{M}$ reads

$$\mathcal{O}\left(\kappa_{\mathcal{M}} \frac{\beta}{\mu} \log \frac{\beta}{\mu} \left(\log \frac{1}{\varepsilon}\right)^2\right). \tag{52}$$

Proof. See Appendix E.1. □

We can make (52) a bit more explicit depending on the choice of $\mathcal{M}$. Note that the condition number of $u_i^k(x)$ in (48) (based on the surrogate (4)) is

$$\frac{\tilde{L}_{\text{mx}}}{\tilde{\mu}_{\text{mn}}} = 1 + \frac{L+\beta}{2\beta - \mu}.$$

Therefore, if $\mathcal{M}$ is the proximal gradient algorithm with stepsize $2/(\tilde{L}_{\text{mx}} + \tilde{\mu}_{\text{mn}})$, we have

$$\kappa_{\mathcal{M}} = 1 + \frac{L+\beta}{2\beta-\mu},$$

and thus (52) becomes

$$\tilde{\mathcal{O}}\left(\left(1 + \frac{\kappa + \beta/\mu}{2\beta/\mu - 1}\right) \cdot \frac{\beta}{\mu} \cdot \left(\log \frac{1}{\varepsilon}\right)^2\right) \tag{53}$$

number of total gradient evaluations/agent, while for $\mathcal{M}$ being the accelerated proximal gradient (with nominal tuning), (52) reads

$$\tilde{\mathcal{O}}\left(\sqrt{1 + \frac{\kappa + \beta/\mu}{2\beta/\mu - 1}} \cdot \frac{\beta}{\mu} \cdot \left(\log \frac{1}{\varepsilon}\right)^2\right) \tag{54}$$

number of total gradient evaluations, where $\tilde{\mathcal{O}}$ hides log-factors.

E.1 Proof of Theorem 12 and Theorem 13

E.1.1 Sketch of the Proof

The proof for Theorem 12 and Theorem 13 is organized in the following three steps:

Step 1: We first establish convergence for `Inexact-SONATA`. Specifically, we prove the connections among the optimality measures and the consensus/tracking error in (68), (70) and (71), in the presence of the inexactness parameter ξ_i^k's;

Step 2: we show that by carefully choosing the inexactness parameters for every inner iteration of every outer loop, the number of inner iterations can still be a constant as $T = \mathcal{O}\left(\log \frac{\beta}{\mu}\right)$; and finally

Step 3: combines the results in **Step 1** and **Step 2** to determine the overall communication and computational complexity of `Inexact ACC-SONATA-F`.

Basic definitions and notation. We begin introducing some definitions and basic facts that will be used throughout the proof.

We denote by $\tilde{\mu}_i$ and $\tilde{L}_i$ the strong convexity and smoothness constants of the surrogate function $\tilde{f}_i(\bullet\,; x_i^k)$, respectively. We define:

$$g^k \triangleq \frac{1}{m}\sum_i u(x_i^k) - u^\star, \quad d_i^k \triangleq \tilde{x}_i^{k+1/2} - x_i^k, \quad \tau_i^k \triangleq \nabla f(x_i^k) - y_i^k, \quad \tilde{\mu}_{\text{mn}} \triangleq \min_{i\in[m]} \tilde{\mu}_i, \quad \tilde{L}_{\text{mx}} \triangleq \max_{i\in[m]} \tilde{L}_i, \tag{55}$$

and the concatenation of local variables $d^k \triangleq \left[{d_1^k}^\top, \cdots, {d_m^k}^\top\right]^\top$, $\tau^k \triangleq \left[{\tau_1^k}^\top, \cdots, {\tau_m^k}^\top\right]^\top$. By Sun et al. (2022), we know that there exist $\{D_i^\ell, D_i^u\}_{i\in[m]}$ such that

$$D_i^\ell \mathbf{I} \preceq \nabla^2 \tilde{f}_i(x;y) - \nabla^2 f(x) \preceq D_i^u \mathbf{I}, \quad \forall x, y \in \mathbb{R}^d.$$

We denote $D_i \triangleq \max\left\{\left|D_i^\ell\right|, \left|D_i^u\right|\right\}$, $D_{\text{mx}} = \max_{i\in[m]} D_i$, and $D_{\text{mn}}^\ell \triangleq \min_{i\in[m]} D_i^\ell$.

We will leverage the following basic facts on subdifferential calculus; see, e.g., Rockafellar (1970). The ϵ-subdifferential of a convex function f, whit $\epsilon > 0$, is defined as

$$\partial_\epsilon f(x) \triangleq \left\{ g \,\middle|\, f(y) \geq (f(x) - \epsilon) + g^\top (y - x),\ \forall\, y \in \mathbb{R}^d \right\}.$$

A direct consequence of this definition is the following fact about ϵ-minimizers, x_ε, of f (assumed to be convex and proper), i.e.,

$$x_\epsilon:\ f(x_\epsilon) - \min_{x\in\mathbb{R}^d} f(x) \leq \epsilon \quad \Leftrightarrow \quad 0 \in \partial_\epsilon f(x_\epsilon). \tag{56}$$

If, in addition, f is also L-smooth, we have the following.

Lemma 14. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and L-smooth. Then,*

$$\partial_\epsilon f(x) \subseteq \left\{ \nabla f(x) + \chi \,\middle|\, \|\chi\|^2 \leq 2L\,\epsilon \right\}. \tag{57}$$

E.1.2 Step 1: Convergence Results of `Inexact-SONATA`

we present the convergence results of `Inexact-SONATA` to solve a general convex optimization problem

$$\min_{x\in\mathbb{R}^d} \ u(x) \triangleq f(x) + r(x), \quad f(x) \triangleq \frac{1}{m}\sum_{i=1}^m f_i(x), \tag{58}$$

with $f(x)$ being Υ-strongly convex and $\mathcal{L}$-smooth. We use single time index $(\bullet)^k$ to denote any variable $(\bullet)$ at the iteration k of the Algorithm 5. To facilitate the discussion, we denote the smooth part of $u_i^k(x)$ in (48) as

$$s_i^k(x) \triangleq \tilde{f}_i(x; x_i^k) + \left\langle y_i^k - \nabla f_i(x_i^k), x - x_i^k \right\rangle,$$

and thus write

$$u_i^k(x) = s_i^k(x) + r(x).$$

Our first step is to establish an inexact descent property of u at each $\tilde{x}_i^{k+1/2}$, as stated in the following lemma, which can be deemed as a counterpart of (Sun et al., 2022, Lemma 3.1), in the presence of inexact computations of the solutions of the subproblems in `(S.1)`.

Lemma 15. *Let $\{x_i^k\}$ be the sequence generated by the Algorithm 5; there holds*

$$u(\tilde{x}_i^{k+1/2}) \leq u(x_i^k) - \frac{1}{2}\left(\tilde{\mu}_{mn} + D_{mn}^\ell - \epsilon_1 - \epsilon_2\right)\left\|d_i^k\right\|^2 + \frac{1}{2\epsilon_1}\left\|\tau_i^k\right\|^2 + \left(\frac{\tilde{L}_{mx}}{\epsilon_2} + 1\right)\xi_i^k. \tag{59}$$

ϵ_1, ϵ_2 with $\epsilon_1 + \epsilon_2 < \tilde{\mu}_{mn} + D_{mn}^l$ are parameters to be determined.

Proof. At step `(S.1)`, $\tilde{x}_i^{k+1/2}$ is a ξ_i^k-optimal solution of $\min_{x\in\mathbb{R}^d} u_i^k(x)$. Using (56), this implies that $\tilde{x}_i^{k+1/2}$ satisfies

$$0 \in \partial_{\xi_i^k} u_i^k(\tilde{x}_i^{k+1/2}) \subset \partial_{\xi_i^k} s_i^k(\tilde{x}_i^{k+1/2}) + \partial_{\xi_i^k} r(\tilde{x}_i^{k+1/2}). \tag{60}$$

Using (60), (57) and the fact that $s_i^k(x)$ is $\tilde{L}_i$-smooth, we infer that there exists a $\chi_i^k \in \mathbb{R}^d$ with

$$\left\|\chi_i^k\right\|^2 \leq 2\tilde{L}_i \xi_i^k, \tag{61}$$

such that,

$$h_i^k \triangleq \nabla s_i^k(\tilde{x}_i^{k+1/2}) + \chi_i^k \in \partial_{\xi_i^k} s_i^k(\tilde{x}_i^{k+1/2}) \quad \text{and} - h_i^k \in \partial_{\xi_i^k} r(\tilde{x}_i^{k+1/2}).$$

Therefore, for any $x \in \mathbb{R}^d$, we have

$$r(x) - r(\tilde{x}_i^{k+1/2}) + \xi_i^k \geq \left\langle x - \tilde{x}_i^{k+1/2}, -h_i^k \right\rangle = \left\langle \tilde{x}_i^{k+1/2} - x, \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}, x_i^k) + y_i^k - \nabla f_i(x_i^k) + \chi_i^k \right\rangle. \quad (62)$$

Using (62) with $x = x_i^k$ leads to

$$r(x_i^k) - r(\tilde{x}_i^{k+1/2}) + \xi_i^k \geq \left\langle d_i^k,\ \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) + y_i^k - \nabla f_i(x_i^k) + \chi_i^k \right\rangle = \left\langle d_i^k,\ y_i^k + \widetilde{H}_i^k d_i^k + \chi_i^k \right\rangle \quad (63)$$

with $\widetilde{H}_i^k \triangleq \int_0^1 \nabla^2 \tilde{f}_i(\theta \tilde{x}_i^{k+1/2} + (1-\theta) x_i^k; x_i^k) \mathrm{d}\theta$. Then, we get

$$\begin{aligned}
f(\tilde{x}_i^{k+1/2}) &\overset{(a)}{=} f(x_i^k) + \left\langle \nabla f(x_i^k), d_i^k \right\rangle + \left\langle H_i^k d_i^k, d_i^k \right\rangle = f(x_i^k) + \left\langle \tau_i^k, d_i^k \right\rangle + \left\langle y_i^k, d_i^k \right\rangle + \left\langle H_i^k d_i^k, d_i^k \right\rangle \\
&\overset{(b)}{\leq} f(x_i^k) + \left\langle \tau_i^k, d_i^k \right\rangle + \left\langle H_i^k d_i^k, d_i^k \right\rangle + r(x_i^k) - r(\tilde{x}_i^{k+1/2}) - \left\langle \widetilde{H}_i^k d_i^k, d_i^k \right\rangle - \left\langle d_i^k, \chi_i^k \right\rangle + \xi_i^k \\
&\overset{(c)}{\leq} f(x_i^k) + \left\langle \tau_i^k, d_i^k \right\rangle - \frac{1}{2}\left(D_{\mathrm{mn}}^\ell + \tilde{\mu}_{\mathrm{mn}}\right) \left\| d_i^k \right\|^2 + r(x_i^k) - r(\tilde{x}_i^{k+1/2}) - \left\langle d_i^k, \chi_i^k \right\rangle + \xi_i^k \\
&\leq f(x_i^k) - \frac{1}{2}(D_{\mathrm{mn}}^\ell + \tilde{\mu}_{\mathrm{mn}}) \left\| d_i^k \right\|^2 + r(x_i^k) - r(\tilde{x}_i^{k+1/2}) + \frac{1}{2}\left(\epsilon_1 \left\| d_i^k \right\|^2 + \epsilon_1^{-1} \left\| \tau_i^k \right\|^2\right) \\
&\quad + \frac{1}{2}\left(\epsilon_2 \left\| d_i^k \right\|^2 + \epsilon_2^{-1} \left\| \chi_i^k \right\|^2\right) + \xi_i^k,
\end{aligned}$$

where in (a) we used the Taylor's formula with $H_i^k \triangleq \int_0^1 (1-\theta) \nabla^2 f(\theta \tilde{x}_i^{k+1/2} + (1-\theta) x_i^k) \mathrm{d}\theta$; in (b), we upper bound $\left\langle y_i^k, d_i^k \right\rangle$ through (63); and in (c), $H_i^k - \widetilde{H}_i^k \preceq -\frac{1}{2}\left(D_{\mathrm{mn}}^\ell + \tilde{\mu}_{\mathrm{mn}}\right) \mathbf{I}$, obtained by setting $\alpha = 1$ in (Sun et al., 2022, (32)). We obtain (59) using (61). □

Recalling the definition of the optimality gap g^k as in (55) and using (59) and the convexity of u, we get

$$g^{k+1} \leq g^k - \frac{1}{2m}\left(\tilde{\mu}_{\mathrm{mn}} + D_{\mathrm{mn}}^\ell - \epsilon_1 - \epsilon_2\right) \left\| d^k \right\|^2 + \frac{1}{2m\epsilon_1} \left\| \tau^k \right\|^2 + \left(\frac{\tilde{L}_{\mathrm{mx}}}{\epsilon_2} + 1\right) \frac{1}{m} \sum_{i=1}^m \xi_i^k. \quad (64)$$

As second step, we derive below the lower bound of $\left\| d^k \right\|$, which is the counterpart of (Sun et al., 2022, Lemma 3.2) when inexact solutions are allowed in agents' subproblems.

Lemma 16. *The following lower bound holds for* $\left\| d^k \right\|^2$*:*

$$\frac{1}{m} \left\| d^k \right\|^2 \geq \frac{\Upsilon}{D_{mx}^2} \left(g^{k+1} - \frac{2}{m\mu} \left\| \tau^k \right\|^2 - \frac{1}{m}\left(1 + \frac{4\tilde{L}_{mx}}{\Upsilon}\right) \sum_{j=1}^m \xi_j^k \right). \quad (65)$$

Proof. Applying (62) with $x = x^\star$ leads to

$$r(x^\star) - r(\tilde{x}_i^{k+1/2}) + \xi_i^k \geq \left\langle \tilde{x}_i^{k+1/2} - x^\star, \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) + y_i^k - \nabla f_i(x_i^k) + \chi_i^k \right\rangle \quad (66)$$

By the Υ-strongly convexity of f, we have

$$\begin{aligned}
&u(x^\star) \geq u(\tilde{x}_i^{k+1/2}) + r(x^\star) - r(\tilde{x}_i^{k+1/2}) + \left\langle \nabla f(\tilde{x}_i^{k+1/2}), x^\star - \tilde{x}_i^{k+1/2} \right\rangle + \frac{\Upsilon}{2} \left\| x^\star - \tilde{x}_i^{k+1/2} \right\|^2 \\
&\overset{(66)}{\geq} u(\tilde{x}_i^{k+1/2}) + \left\langle x^\star - \tilde{x}_i^{k+1/2}, \nabla f(\tilde{x}_i^{k+1/2}) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) - y_i^k + \nabla f_i(x_i^k) - \chi_i^k \right\rangle + \frac{\Upsilon}{2} \left\| x^\star - \tilde{x}_i^{k+1/2} \right\|^2 - \xi_i^k \\
&\geq u(\tilde{x}_i^{k+1/2}) - \frac{1}{2\Upsilon} \left\| \nabla f(\tilde{x}_i^{k+1/2}) - \nabla f(x_i^k) + \nabla f_i(x_i^k) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) + \tau_i^k - \chi_i^k \right\|^2 - \xi_i^k \\
&\geq u(\tilde{x}_i^{k+1/2}) - \frac{D_i^2}{\Upsilon} \left\| d_i^k \right\|^2 - \frac{2}{\Upsilon} \left\| \tau_i^k \right\|^2 - \frac{2}{\Upsilon} \left\| \chi_i^k \right\|^2 - \xi_i^k \overset{(61)}{\geq} u(\tilde{x}_i^{k+1/2}) - \frac{D_i^2}{\Upsilon} \left\| d_i^k \right\|^2 - \frac{2}{\Upsilon} \left\| \tau_i^k \right\|^2 - \left(1 + \frac{4\tilde{L}_i}{\Upsilon}\right) \xi_i^k
\end{aligned} \quad (67)$$

Summing the inequalities over i, and using the convexity of $u(x)$ lead to the conclusion. □

Combining (64) and (65) to cancel out $\left\|d^k\right\|^2$, and noticing that

$$\frac{1}{m}\left\|\tau^k\right\|^2 \leq 4L_{\text{mx}}^2\left\|x_\perp^k\right\|^2 + 2\left\|y_\perp^k\right\|^2,$$

we obtain

$$g^{k+1} \leq \sigma_0 g^k + \eta_0\left(4L_{\text{mx}}^2\left\|x_\perp^k\right\|^2 + 2\left\|y_\perp^k\right\|^2\right) + \eta_1\frac{1}{m}\sum_{j=1}^m \xi_j^k, \tag{68}$$

with

$$\begin{aligned}
&\sigma_0 \triangleq \frac{2D_{\text{mx}}^2}{2D_{\text{mx}}^2 + \Upsilon\left(\tilde{\mu}_{\text{mn}} + D_{\text{mn}}^\ell - \epsilon_1 - \epsilon_2\right)}, \qquad \eta_0 \triangleq \sigma_0\left(\frac{1}{2\epsilon_1} + \frac{\tilde{\mu}_{\text{mn}} + D_{\text{mn}}^\ell - \epsilon_1 - \epsilon_2}{D_{\text{mx}}^2}\right),\\
&\eta_1 \triangleq \sigma_0\left(\frac{\tilde{L}_{\text{mx}}}{\epsilon_2} + 1 + \frac{\Upsilon\left(\tilde{\mu}_{\text{mn}} + D_{\text{mn}}^\ell - \epsilon_1 - \epsilon_2\right)}{2D_{\text{mx}}^2}\left(1 + \frac{4L_{\text{mx}}^2}{\Upsilon}\right)\right).
\end{aligned} \tag{69}$$

On the other hand, we recall the following result on the consensus and tracking error from (41):

$$4L_{\text{mx}}^2\left\|x_\perp^{k+1}\right\|^2 + 2\left\|y_\perp^{k+1}\right\|^2 \leq \rho^2\left(32L_{\text{mx}}^2\left\|x_\perp^k\right\|^2 + 6\left\|y_\perp^k\right\|^2 + 14L_{\text{mx}}^2\frac{1}{m}\left\|d^k\right\|^2\right). \tag{70}$$

Equipped with (68) and (70), we proceed to bound $\left\|d^k\right\|^2$ as a function of the optimality gap g^k and the consensus/tracking error to close the loop. To this end, we give the following result, which is the counterpart of (Sun et al., 2022, Proposition 3.6) in the presence of inexact solutions.

Proposition 17. *The following upper bound hold for* $\left\|d^k\right\|^2$*:*

$$\frac{1}{m}\left\|d^k\right\|^2 \leq \sigma_1 g^k + \eta_2\left\|y_\perp^k\right\|^2 + \frac{\eta_3}{m}\sum_{j=1}^m \xi_j^k, \tag{71}$$

with

$$\sigma_1 \triangleq \frac{2}{\Upsilon}\left(2 + \frac{8D_{mx}^2}{\tilde{\mu}_{mn}^2} + \frac{32L_{mx}^2}{\tilde{\mu}_{mn}^2}\right), \quad \eta_2 \triangleq \frac{8}{\tilde{\mu}_{mn}^2}, \quad \eta_3 \triangleq \left(\frac{16L_{mx}}{\tilde{\mu}_{mn}^2} + \frac{4}{\tilde{\mu}_{mn}}\right).$$

Proof. Applying (62) with $x = x^\star$ and invoking the optimality condition of $x^\star$, we get

$$\begin{aligned}
&r(x^\star) - r(\tilde{x}_i^{k+1/2}) - \left\langle \tilde{x}_i^{k+1/2} - x^\star, \nabla\tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) + y_i^k - \nabla f_i(x_i^k) + \chi_i^k\right\rangle + \xi_i^k \geq 0,\\
&\left\langle \nabla f(x^\star), \tilde{x}_i^{k+1/2} - x^\star\right\rangle + r(\tilde{x}_i^{k+1/2}) - r(x^\star) \geq 0.
\end{aligned}$$

Combining the above two, we get

$$\begin{aligned}
0 &\le \left\langle \nabla f(x^\star) - y_i^k + \nabla f_i(x_i^k) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k) - \chi_i^k \pm \bar{y}^k,\ \tilde{x}_i^{k+1/2} - x^\star \right\rangle + \xi_i^k \\
&\le \left\langle \nabla f(x^\star) - \frac{1}{m}\sum_{j=1}^m \nabla f_j(x_j^k) + \nabla f_i(x_i^k) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k),\ \tilde{x}_i^{k+1/2} - x^\star \right\rangle \\
&\quad + \|\bar{y}^k - y_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \|\chi_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \xi_i^k \\
&\le \left\langle \nabla f(x^\star) - \nabla f(x_i^k) + \nabla f_i(x_i^k) \pm \nabla \tilde{f}_i(x^\star; x_i^k) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k),\ \tilde{x}_i^{k+1/2} - x^\star \right\rangle \\
&\quad + \|\bar{y}^k - y_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \|\chi_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \left\|\nabla f(x_i^k) - \frac{1}{m}\sum_{j=1}^m \nabla f_j(x_j^k)\right\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \xi_i^k \\
&= \left\langle \int_0^1 \left(\nabla^2 f(\theta x^\star + (1-\theta)x_i^k) - \nabla^2 \tilde{f}_i(\theta x^\star + (1-\theta)x_i^k; x_i^k)\right)\left(x^\star - x_i^k\right) \mathrm{d}\theta,\ \tilde{x}_i^{k+1/2} - x^\star \right\rangle \\
&\quad + \left\langle \nabla \tilde{f}_i(x^\star; x_i^k) - \nabla \tilde{f}_i(\tilde{x}_i^{k+1/2}; x_i^k),\ \tilde{x}_i^{k+1/2} - x^\star \right\rangle + \|\bar{y}^k - y_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| \\
&\quad + \left(\frac{1}{m}\sum_{j=1}^m L_j \|x_j^k - x_i^k\|\right) \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \|\chi_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \xi_i^k \\
&\le D_i \|x^\star - x_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| - \tilde{\mu}_i \left\|\tilde{x}_i^{k+1/2} - x^\star\right\|^2 + \|\bar{y}^k - y_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| \\
&\quad + \left(\frac{1}{m}\sum_{j=1}^m L_j \|x_j^k - x_i^k\|\right) \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \|\chi_i^k\| \left\|\tilde{x}_i^{k+1/2} - x^\star\right\| + \xi_i^k \\
&\le \frac{D_i}{2q_1} \|x_i^k - x^\star\|^2 + \frac{1}{2q_2} \|\bar{y}^k - y_i^k\|^2 + \frac{1}{2q_3} \left(\frac{1}{m}\sum_{j=1}^m L_j \|x_j^k - x_i^k\|\right)^2 + \frac{1}{2q_4} \|\chi_i^k\|^2 \\
&\quad + \left(\frac{1}{2}\left(D_i q_1 + q_2 + q_3 + q_4\right) - \tilde{\mu}_i\right) \left\|\tilde{x}_i^{k+1/2} - x^\star\right\|^2 + \xi_i^k
\end{aligned}$$

Setting $q_1 = \frac{\tilde{\mu}_i}{4D_i}, q_2 = q_3 = q_4 = \frac{\tilde{\mu}_i}{4}$ in the above and plugging in the following inequalities,

$$\begin{aligned}
\left\|\tilde{x}_i^{k+1/2} - x^\star\right\|^2 &\ge -\|x^\star - x_i^k\|^2 + \frac{1}{2}\|d_i^k\|^2, \\
\|x^\star - x_i^k\|^2 &\le \frac{2}{\Upsilon}\left(u(x_i^k) - u(x^\star)\right), \\
\|x_j^k - x_i^k\|^2 &\le 2\|x_j^k - x^\star\|^2 + 2\|x_i^k - x^\star\|^2,
\end{aligned}$$

we get

$$\|d_i^k\|^2 \le \left(2 + \frac{8D_i^2}{\tilde{\mu}_i^2} + \frac{16L_{\text{mx}}^2}{\tilde{\mu}_i^2}\right)\|x^\star - x_i^k\|^2 + \frac{8}{\tilde{\mu}_i^2}\|\bar{y}^k - y_i^k\|^2 + \frac{16}{\tilde{\mu}_i^2}\frac{L_{\text{mx}}^2}{m}\sum_{j=1}^m \|x_j^k - x^\star\|^2 + \frac{8}{\tilde{\mu}_i^2}\|\chi_i^k\|^2 + \frac{4}{\tilde{\mu}_i}\xi_i^k.$$

Using (61) and summing the above over $i \in [m]$, we get

$$\frac{1}{m}\|d^k\|^2 \le \frac{2}{\Upsilon}\left(2 + \frac{8D_{\text{mx}}^2}{\tilde{\mu}_{\text{mn}}^2} + \frac{32L_{\text{mx}}^2}{\tilde{\mu}_{\text{mn}}^2}\right) g^k + \frac{8}{\tilde{\mu}_{\text{mn}}^2}\|y_\perp^k\|^2 + \left(\frac{16L_{\text{mx}}}{\tilde{\mu}_{\text{mn}}^2} + \frac{4}{\tilde{\mu}_{\text{mn}}}\right)\frac{1}{m}\sum_{j=1}^m \xi_j^k.$$

□

Now we are ready to combine (68), (70) and (71) to obtain the contraction results for the `Inexact-SONATA`. Beforehand, we notice that when `Inexact-SONATA` is adopted inside Algorithm 6 with $\delta = \beta - \mu$, the aforementioned parameters can be set as

$$\Upsilon = \beta, \quad D^{\ell}_{\rm mn} = 0, \quad D_{\rm mx} = 2\beta, \quad \tilde{\mu}_{\rm mn} = 2\beta - \mu, \quad L_{\rm mx} = L + 2\beta - \mu, \quad \tilde{L}_{\rm mx} = L + 3\beta - \mu.$$

In addition, we choose

$$\epsilon_1 = \epsilon_2 = \frac{2\beta - \mu}{4}.$$

One can then easily check in (68), $\sigma_0 \le \frac{16}{17}$, $\eta_0 \le \frac{36}{17\beta}$. Multiplying (70) by $\frac{72}{17\beta}$ and (71) by $\frac{1008 L_{\rm mx}^2 \rho^2}{17\beta}$, and combining the obtained inequalities with (68) yield

$$\begin{aligned} & g^{k+1} + \frac{288 L_{\rm mx}^2}{17\beta}\left\|x_{\perp}^{k+1}\right\|^2 + \frac{144}{17\beta}\left\|y_{\perp}^{k+1}\right\|^2 \\ & \le \left(\frac{16}{17} + \frac{1008\, L_{\rm mx}^2\, \rho^2}{17\beta}\sigma_1\right) g^k + \frac{144 L_{\rm mx}^2}{17\beta}\left(1 + 16\rho^2\right)\left\|x_{\perp}^{k}\right\|^2 + \frac{72}{17\beta}\left(1 + 6\rho^2 + 14 L_{\rm mx}^2 \rho^2 \eta_2\right)\left\|y_{\perp}^{k}\right\|^2 \\ & \quad + \left(\eta_1 + \frac{1008\, L_{\rm mx}^2\, \rho^2}{17\beta}\eta_3\right)\frac{1}{m}\sum_{j=1}^{m}\xi_j^k. \end{aligned} \tag{72}$$

Thus, with

$$\rho^2 \le \min\left(\frac{\beta}{2016\, L_{\rm mx}^2\, \sigma_1},\ \frac{8}{17\left(3 + 7\, L_{\rm mx}^2\, \eta_2\right)}\right), \tag{73}$$

we have

$$g^{k+1} + \frac{288 L_{\rm mx}^2}{17\beta}\left\|x_{\perp}^{k+1}\right\|^2 + \frac{144}{17\beta}\left\|y_{\perp}^{k+1}\right\|^2 \le \frac{33}{34}\left(g^k + \frac{288 L_{\rm mx}^2}{17\beta}\left\|x_{\perp}^{k}\right\|^2 + \frac{144}{17\beta}\left\|y_{\perp}^{k}\right\|^2\right) + \eta_4 \frac{1}{m}\sum_{j=1}^{m}\xi_j^k, \tag{74}$$

where

$$\eta_4 \triangleq \eta_1 + \frac{1008\, L_{\rm mx}^2\, \rho^2}{17\beta}\eta_3.$$

E.1.3 Step 2: `Inexact-SONATA` as an Inner Algorithm in `Inexact ACC-SONATA-F`

In this section, we use double time index $(\bullet)^{k,t}$ to denote any variable $(\bullet)$ of `Inexact SONATA` in the t-th inner iteration of the k-th outer iteration of the Algorithm 6. Recalling the definition of P^k in (22), we define

$$\xi_i^{k,t} \triangleq \frac{P^0}{\eta_4}\left(1 - c\sqrt{\frac{\mu}{\beta}}\right)^k \left(\frac{16}{17}\right)^t, \quad \text{for } \forall\, i \in [m],\ k \ge 0,\ t \ge 0. \tag{75}$$

To comply with the notations in (20), we define

$$e^{k,t} \triangleq c_x \left\|x_{\perp}^{k,t}\right\|^2 + c_y \left\|y_{\perp}^{k,t}\right\|^2, \quad \text{with} \quad c_x \triangleq \frac{288 L_{\rm mx}^2}{17\beta} \quad \text{and} \quad c_y \triangleq \frac{144}{17\beta}. \tag{76}$$

According to the discussion in Section C.2, if one can guarantee

$$g^{k,T} + e^{k,T} \le \epsilon^{k+1}, \quad k = 0, 1, \ldots, \tag{77}$$

with $\epsilon^k \triangleq P^0 \cdot (1 - c\cdot\alpha)^k$, then $P^k = \mathcal{O}\left((1 - c\cdot\alpha)^k\right)$. By (74), we have

$$g^{k,t+1} + e^{k,t+1} \le \frac{33}{34}\left(g^{k,t} + e^{k,t}\right) + P^0\left(1 - c\sqrt{\frac{\mu}{\beta}}\right)^k \left(\frac{16}{17}\right)^t,\ \forall\, k \ge 0,\ t \ge 0.$$

In particular, for a fixed k, applying the above telescopically on t yields

$$g^{k,T}+e^{k,T}\le\left(\frac{33}{34}\right)^T\left(g^{k,0}+e^{k,0}\right)+P^0\left(1-c\sqrt{\frac{\mu}{\beta}}\right)^k\sum_{t=0}^{T}\left(\frac{33}{34}\right)^{T-t}\left(\frac{16}{17}\right)^t$$
$$=\left(\frac{33}{34}\right)^T\left(g^{k,0}+e^{k,0}+P^0\left(1-c\sqrt{\frac{\mu}{\beta}}\right)^k\sum_{t=0}^{T}\left(\frac{32}{33}\right)^t\right)\le\left(\frac{33}{34}\right)^T\left(g^{k,0}+e^{k,0}+33\,P^0\left(1-c\sqrt{\frac{\mu}{\beta}}\right)^k\right).$$

Therefore, according to Lemma 9, for $k\ge 1$, the number of inner loops needed for (77) can be bounded as

$$T\le\left\lceil 34\,\log\frac{g^{k,0}+e^{k,0}+33\,P^0\left(1-c\sqrt{\frac{\mu}{\beta}}\right)^k}{\epsilon^{k+1}}\right\rceil$$
$$\overset{(34)}{\le}\left\lceil 34\,\log\frac{P^0\left(2(1-c\cdot\alpha)^k+\frac{576(\beta-\mu)^2}{\mu\beta}c_2\,(1-c\cdot\alpha)^{k-1}+33(1-c\cdot\alpha)^k\right)}{P^0\,(1-c\cdot\alpha)^{k+1}}\right\rceil$$
$$=\left\lceil 34\,\log\frac{35(1-c\cdot\alpha)+\frac{576(\beta-\mu)^2}{\mu\beta}c_2}{(1-c\cdot\alpha)^2}\right\rceil=\mathcal{O}\left(\log\frac{\beta}{\mu}\right).$$

For $k=0$, due to $g^{0,0}+e^{0,0}\le P^0$, we have $T=\left\lceil 34\,\log\frac{34}{1-c\cdot\alpha}\right\rceil$, which is smaller than the RHS of the above.

E.1.4 Step 3: Complexity of `Inexact ACC-SONATA-F`

The total number of computation steps taken by $\mathcal{M}$ as in Theorem 13 can be readily obtained as follows:

$$\sum_{k=0}^{K-1}\sum_{t=0}^{T-1}T^{k,t}=\kappa_{\mathcal{M}}\sum_{k=0}^{K-1}\sum_{t=0}^{T-1}\log\frac{1}{\xi_1^{k,t}}\overset{(75)}{\le}\kappa_{\mathcal{M}}\sum_{k=0}^{K-1}\sum_{t=0}^{T-1}\left(\log\left(\frac{\eta_4}{P^0}\right)+k\,\log\left(\frac{1}{1-c}\right)+t\,\log\frac{17}{16}\right)$$
$$\le\kappa_{\mathcal{M}}\left(KT\log\left(\frac{\eta_4}{P^0}\right)+T\frac{K(K-1)}{2}\log\left(\frac{1}{1-c}\right)+K\frac{T(T-1)}{2}\log\frac{17}{16}\right).$$

Since $K=\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}}\log\frac{1}{\epsilon}\right)$ and $T=\mathcal{O}\left(\log\frac{\beta}{\mu}\right)$, we have

$$\sum_{k=0}^{K-1}\sum_{t=0}^{T-1}T^{k,t}=\mathcal{O}\left(\kappa_{\mathcal{M}}\,K^2\,T\right)=\mathcal{O}\left(\kappa_{\mathcal{M}}\,\frac{\beta}{\mu}\,\log\frac{\beta}{\mu}\,\left(\log\frac{1}{\varepsilon}\right)^2\right).$$

In addition, the total communication complexity is $K\cdot T=\mathcal{O}\left(\sqrt{\frac{\beta}{\mu}}\cdot T\cdot\log\frac{1}{\varepsilon}\right)$.