IDEAL: Inexact DEcentralized Accelerated Augmented Lagrangian Method

Yossi Arjevani
NYU
yossia@nyu.edu

Joan Bruna
NYU
bruna@cims.nyu.edu

Bugra Can
Rutgers University
bc600@scarletmail.rutgers.edu

Mert Gürbüzbalaban
Rutgers University
mg1366@rutgers.edu

Stefanie Jegelka
MIT
stefje@csail.mit.edu

Hongzhou Lin
MIT
hongzhou@mit.edu

Abstract

We introduce a framework for designing primal methods under the decentralized optimization setting where local functions are smooth and strongly convex. Our approach consists of approximately solving a sequence of sub-problems induced by the accelerated augmented Lagrangian method, thereby providing a systematic way for deriving several well-known decentralized algorithms including EXTRA [41] and SSDA [37]. When coupled with accelerated gradient descent, our framework yields a novel primal algorithm whose convergence rate is optimal and matched by recently derived lower bounds. We provide experimental results that demonstrate the effectiveness of the proposed algorithm on highly ill-conditioned problems.

1 Introduction

Due to their rapidly increasing size, modern datasets are typically collected, stored and manipulated in a distributed manner. This, together with strict privacy requirements, has created a large demand for efficient solvers for the decentralized setting in which models are trained locally at each agent, and only local parameter vectors are shared. This approach has become particularly appealing for applications such as edge computing [25, 42], cooperative multi-agent learning [6, 33] and federated learning [26, 43]. Clearly, the nature of the decentralized setting prevents a global synchronization, as only communication within the neighboring machines is allowed. The goal is then to arrive at a consensus on all local agents with a model that performs as well as in the centralized setting.

Arguably, the simplest approach for addressing decentralized settings is to adapt the vanilla gradient descent method to the underlying network architecture [9, 16, 29, 47]. To this end, the connections between the agents are modeled through a mixing matrix, which dictates how agents average over their neighbors' parameter vectors. Thus, the mixing matrix serves as a communication oracle which determines how information propagates throughout the network. Perhaps surprisingly, when the stepsizes are constant, simply averaging over the local iterates via the mixing matrix only converges to a neighborhood of the optimum [41, 50]. A recent line of works [15, 30, 31, 34, 40, 41] proposed a number of alternative methods that linearly converge to the global minimum.

The overall complexity of solving decentralized optimization problems is typically determined by two factors: (i) the condition number of the objective function κ_f, which measures the 'hardness' of solving the underlying optimization problem, and (ii) the condition number of the mixing matrix κ_W, which quantifies the severity of information 'bottlenecks' present in the network. Lower complexity bounds recently derived for distributed settings [1, 3, 37, 46] show that one cannot expect to have a better dependence on the condition numbers than $\sqrt{\kappa_f}$ and $\sqrt{\kappa_W}$. Notably, despite the considerable

Preprint. Under review.

recent progress, none of the methods mentioned above is able to achieve accelerated rates, that is, a square root dependence for both κ_f and κ_W—simultaneously.

An extensive effort has been devoted to obtaining acceleration for decentralized algorithms under various settings [10, 11, 14, 22, 37, 38, 45, 48, 51]. When a dual oracle is available, that is, access to the gradients of the dual functions is provided, optimal rates can be attained for smooth and strongly convex objectives [37]. However, having access to a dual oracle is a very restrictive assumption, and resorting to a direct 'primalization' through inexact approximation of the dual gradients leads to sub-optimal worst-case theoretical rates [45]. In this work, we propose a novel primal approach that leads to optimal rates in terms of dependency on κ_f and κ_W.

Our contributions can be summarized as follows.

- We introduce a novel framework based on the accelerated augmented Lagrangian method for designing primal decentralized methods. The framework provides a simple and systematic way for deriving several well-known decentralized algorithms [15, 40, 41], including EXTRA [41] and SSDA [37], and unifies their convergence analyses.
- Using accelerated gradient descent as a sub-routine, we derive a novel method for smooth and strongly convex local functions which achieves optimal *accelerated* rates on both the condition numbers of the problem, κ_f and κ_W, using *primal* updates, see Table 2.
- We perform a large number of experiments, which confirm our theoretical findings, and demonstrate a significant improvement when the objective function is ill-conditioned and $\kappa_f \gg \kappa_W$.

2 Decentralized Optimization Setting

We consider n computational agents and a network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ which defines how the agents are linked. The set of vertices $\mathcal{V} = \{1, \cdots, n\}$ represents the agents and the set of edges $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ specifies the connectivity in the network, i.e., a communication link between agents i and j exists if and only if $(i, j) \in \mathcal{E}$. Each agent has access to local information encoded by a loss function $f_i : \mathbb{R}^d \to \mathbb{R}$. The goal is to minimize the global objective over the entire network,

$$\min_{x \in \mathbb{R}^d} f(x) := \sum_{i=1}^{n} f_i(x). \tag{1}$$

In this paper, we assume that the local loss functions f_i are differentiable, L-smooth and μ-strongly convex.[1] Strong convexity of the component functions f_i implies that the problem admits a unique solution, which we denote by x^*.

We consider the following computation and communication models [37]:

- **Local computation:** Each agent is able to compute the gradients of f_i and the cost of this computation is one unit of time.
- **Communication:** Communication is done synchronously, and each agent can only exchange information with its neighbors, where i is a neighbor of j if $(i, j) \in \mathcal{E}$. The ratio between the communication cost and computation cost per round is denoted by τ.

We further assume that propagation of information is governed by a mixing matrix $W \in \mathbb{R}^{n \times n}$ [29, 37, 50]. Specifically, given a local copy of the decision variable $x_i \in \mathbb{R}^d$ at node $i \in [1, n]$, one round of communication provides the following update $x_i \leftarrow \sum_{i=1}^{n} W_{ij} x_j$. The following standard assumptions regarding the mixing matrix [37] are made throughout the paper.

Assumption 1. *The mixing matrix W satisfies the following:*

1. **Symmetry:** $W = W^T$.
2. **Positiveness:** *W is positive semi-definite.*
3. **Decentralized property***: If $(i, j) \notin \mathcal{E}$ and $i \neq j$, then $W_{ij} = W_{ji} = 0$.*
4. **Spectrum property:** *The kernel of W is given by the vector of all ones Ker$(W) = \mathbb{R}\mathbf{1_n}$.*

[1] f is L-smooth if ∇f is L-Lipschitz; f is μ-strongly convex if $f - \frac{\mu}{2}\|x\|^2$ is convex.

Algorithm 1 Decentralized Augmented Lagrangian framework

Input: mixing matrix W, regularization parameter ρ, stepsize η.

1: **for** $k = 1, 2, ..., K$ **do**
2: $\quad \mathbf{X}_k = \arg\min \left\{ P_k(\mathbf{X}) := F(\mathbf{X}) + \mathbf{\Lambda}_k^T \mathbf{X} + \frac{\rho}{2} \|\mathbf{X}\|_{\mathbf{W}}^2 \right\}$.
3: $\quad \mathbf{\Lambda}_{k+1} = \mathbf{\Lambda}_k + \eta\, \mathbf{W}\mathbf{X}_k$.
4: **end for**

A typical choice of the mixing matrix is the (weighted) Laplacian matrix of the graph. Another common choice is to set W as $I - \tilde{W}$ where $\tilde{W}$ is a doubly stochastic matrix [5, 8, 41]. By Assumption 1.4, all the eigenvalues of W are strictly positive, except for the smallest one. We let $\lambda_{\max}(W)$ denote the maximum eigenvalue, and let $\lambda_{\min}^{+}(W)$ denote the smallest positive eigenvalue. The ratio between these two quantities plays an important role in quantifying the overall complexity of this problem.

Theorem 1 (Decentralized lower bound [37])**.** *For any first-order black-box decentralized method, the number of time units required to reach an ϵ-optimal solution for (1) is lower bounded by*

$$\Omega\left(\sqrt{\kappa_f}(1+\tau\sqrt{\kappa_W})\log\left(\frac{1}{\epsilon}\right)\right), \tag{2}$$

where $\kappa_f = L/\mu$ is the condition number of the loss function and $\kappa_W = \lambda_{\max}(W)/\lambda_{\min}^{+}(W)$ is the condition number of the mixing matrix.

The lower bound decomposes as follows: a) **computation cost**, given by $\sqrt{\kappa_f}\log(1/\epsilon)$, and b) **communication cost**, given by $\tau\sqrt{\kappa_f\kappa_W}\log(1/\epsilon)$. The computation cost matches lower bounds for centralized settings [2, 32], while the communication cost introduces an additional term which depends on κ_W and accounts for the 'price' of communication in decentralized models. It follows that the effective condition number of a given decentralized problem is $\kappa_W\kappa_f$.

Clearly, the choice of the matrix W can strongly affect the optimal attainable performance. For example, κ_W can get as large as n^2 in the line/cycle graph, or be constant in the complete graph. In this paper, we do not focus on optimizing over the choice of W for a given graph G; instead, following the approach taken by existing decentralized algorithms, we assume that the graph G and the mixing matrix W are given and aim to achieve the optimal complexity (2) for this particular choice of W.

3 Related Work and the Dual Formulation

A standard approach to adress problem (1) is to express it as a constrained optimization problem

$$\min_{\mathbf{X}\in\mathbb{R}^{nd}} F(\mathbf{X}) := \frac{1}{n}\sum_{i=1}^{n} f_i(x_i) \quad \text{such that} \quad x_1 = x_2 = \cdots = x_n \in \mathbb{R}^d, \tag{P}$$

where $\mathbf{X} = [x_1; x_2; \cdots x_n] \in \mathbb{R}^{nd}$ is a concatenation of the vectors. To lighten the notation, we introduce the global mixing matrix $\mathbf{W} = W \otimes I_d \in \mathbb{R}^{nd\times nd}$, where $\otimes$ denotes the Kronecker product, and let $\|\cdot\|_{\mathbf{W}}$ denote the semi-norm induced by $\mathbf{W}$, i.e. $\|\mathbf{X}\|_{\mathbf{W}}^2 = \mathbf{X}^T\mathbf{W}\mathbf{X}$. With this notation in hand, we briefly review existing literature on decentralized algorithms.

Decentralized Gradient Descent The decentralized gradient method [29, 50] has the update rule

$$\mathbf{X}_{k+1} = \mathbf{W}\mathbf{X}_k - \eta\nabla F(\mathbf{X}_k). \tag{DGD}$$

However, with constant stepsize, the algorithm does not converge to a global minimum of (P), but rather to a neighborhood of the solution [50]. A decreasing stepsize schedule may be used to ensure convergence, but this yields a sublinear convergence rate, even in the strongly convex case.

Linearly convergent primal algorithms By and large, recent methods that achieve linear convergence in the strongly convex case [15, 30, 31, 34, 40, 41, 44] can be shown to follow a general framework based on the augmented Lagrangian method, see Algorithm 1; The main difference lies

Algorithm 2 Accelerated Decentralized Augmented Lagrangian framework

Input: mixing matrix W, regularization parameter ρ, stepsize η, extrapolation parameters $\{\beta_k\}_{k\in\mathbb{N}}$

1: Initialize dual variables $\mathbf{\Lambda}_1 = \mathbf{\Omega}_1 = \mathbf{0} \in \mathbb{R}^{nd}$.
2: **for** $k = 1, 2, ..., K$ **do**
3: $\quad \mathbf{X}_k = \arg\min \left\{ P_k(\mathbf{X}) := F(\mathbf{X}) + \mathbf{\Omega}_k^T \mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|_{\mathbf{W}}^2 \right\}$.
4: $\quad \mathbf{\Lambda}_{k+1} = \mathbf{\Omega}_k + \eta \mathbf{W} \mathbf{X}_k$
5: $\quad \mathbf{\Omega}_{k+1} = \mathbf{\Lambda}_{k+1} + \beta_{k+1}(\mathbf{\Lambda}_{k+1} - \mathbf{\Lambda}_k)$
6: **end for**

Output: $\mathbf{X}_K$.

Algorithm 3 IDEAL: Inexact Acc-Decentralized Augmented Lagrangian framework

Additional Input: A first-order optimization algorithm $\mathcal{A}$

Apply $\mathcal{A}$ to solve the subproblem P_k warm starting at $\mathbf{X}_{k-1}$ to find an approximate solution

$$\mathbf{X}_k \approx \arg\min \left\{ P_k(\mathbf{X}) := F(\mathbf{X}) + \mathbf{\Omega}_k^T \mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|_{\mathbf{W}}^2 \right\},$$

Option I: stop the algorithm when $\|\mathbf{X}_k - \mathbf{X}_k^*\|^2 \leq \epsilon_k$, where $\mathbf{X}_k^*$ is the unique minimizer of P_k.

Option II: stop the algorithm after a prefixed number of iterations T_k.

in how subproblems P_k are solved. Shi et al. [40] apply an alternating directions method; in [41], the EXTRA algorithm takes a single gradient descent step to solve P_k, see Appendix B for details. Jakovetić et al. [15] use multi-step algorithms such as Jacobi/Gauss-Seidel methods. To the best of our knowledge, the complexity of these algorithms is not better than $O\left((1+\tau)\kappa_f \kappa_W \log(\frac{1}{\epsilon})\right)$, in other words, they are non-accelerated. The recently proposed algorithm APM-C [22] enjoys a square root dependence on κ_f and κ_W, but incurs an additional $\log(1/\epsilon)$ factor compared to the optimal attainable rate.

Optimal method based on the dual formulation By Assumption 1.4, the constraint $x_1 = x_2 = \cdots = x_n$ is equivalent to the identity $\mathbf{W} \cdot \mathbf{X} = 0$, which is again equivalent to $\sqrt{\mathbf{W}} \cdot \mathbf{X} = 0$. Hence, the dual formulation of (P) is given by

$$\max_{\mathbf{\Lambda} \in \mathbb{R}^{dn}} -F^*(-\sqrt{\mathbf{W}}\mathbf{\Lambda}). \tag{D}$$

Since the primal function is convex and the constraints are linear, we can use strong duality and address the dual problem instead of the primal one. Using this approach, [37] proposed a dual method with optimal accelerated rates, using Nesterov's accelerated gradient method for the dual problem (D). As mentioned earlier, the main drawback of this method is that it requires access to the gradient of the dual function which, unless the primal function has a relatively simple structure, is not available. One may apply a first-order method to approximate the dual gradients inexactly at the expense of an additional $\sqrt{\kappa_f}$ factor in the computation cost [45], but this woul make the algorithm no longer optimal. This indicates that achieving optimal rates when using primal updates is a rather challenging task in the decentralized setting. In the following sections, we provide a generic framework which allows us to derive a primal decentralized method with optimal complexity guarantees.

4 An Inexact Accelerated Augmented Lagrangian framework

In this section, we introduce our inexact accelerated Augmented Lagrangian framework, and show how to combine it with Nesterov's acceleration. To ease the presentation, we first describe a conceptual algorithm, Algorithm 2, where subproblems are solved exactly, and only then introduce inexact inner-solvers.

Similarly to Nesterov's accelerated gradient method, we use an extrapolation step for the dual variable $\mathbf{\Lambda}_k$. The component $\mathbf{W}\mathbf{X}_k$ in line 4 of Algorithm 2 is the negative gradient of the Moreau-envelope[2] of the dual function. Hence our algorithm is equivalent to applying Nesterov's method

[2] A proper definition of the Moreau-envelope is given in [36], readers that are not familiar with this concept could take it as an implicit function which shares the same optimum as the original function.

on the Moreau-envelope of the dual function, or equivalently, an accelerated dual proximal point algorithm. This renders the optimal dual method proposed in [37] as a special case of our algorithmic framework (with ρ set to 0).

While Algorithm 2 is conceptually plausible, it requires an exact solution of the Augmented Lagrangian problems, which can be too expensive in practice. To address this issue, we introduce an inexact version, shown in Algorithm 3, where the k-th subproblem P_k is solved up to a predefined accuracy ϵ_k. The choice of ϵ_k is rather subtle. On the one hand, choosing a large ϵ_k may result in a non-converging algorithm. On the other hand, choosing a small ϵ_k can be exceedingly expensive as the optimal solution of the subproblem $\mathbf{X}_k^*$ is not the global optimum $\mathbf{X}^*$. Intuitively, ϵ_k should be chosen to be of the same order of magnitude as $\|\mathbf{X}_k^* - \mathbf{X}^*\|$, leading to the following result.

Theorem 2. *Consider the sequence of primal variables $(\mathbf{X}_k)_{k\in\mathbb{N}}$ generated by Algorithm 3 with the subproblem P_k solved up to ϵ_k accuracy in Option I. With parameters set to*

$$\beta_k = \frac{\sqrt{L_\rho} - \sqrt{\mu_\rho}}{\sqrt{L_\rho} + \sqrt{\mu_\rho}}, \quad \eta = \frac{1}{L_\rho}, \quad \epsilon_k = \frac{\mu_\rho}{2\lambda_{\max}(W)}\left(1 - \frac{1}{2}\sqrt{\frac{\mu_\rho}{L_\rho}}\right)^k \Delta_{dual}, \tag{3}$$

where $L_\rho = \frac{\lambda_{\max}(W)}{\mu+\rho\lambda_{\max}(W)}$, $\mu_\rho = \frac{\lambda^+_{\min}(W)}{L+\rho\lambda^+_{\min}(W)}$ and Δ_{dual} is the initial dual function gap, we obtain

$$\|\mathbf{X}_k - \mathbf{X}^*\|^2 \le C_\rho \left(1 - \frac{1}{2}\sqrt{\frac{\mu_\rho}{L_\rho}}\right)^k \Delta_{dual}, \tag{4}$$

where $\mathbf{X}^* = \mathbf{1}_n \otimes x^*$ and $C_\rho = 258\frac{L_\rho\lambda_{\max}(W)}{\mu^2\mu_\rho^2}$.

Corollary 3. *The number of subproblems P_k to achieve $\|\mathbf{X}_k - \mathbf{X}^*\|^2 \le \epsilon$ in IDEAL is bounded by*

$$K = O\left(\sqrt{\frac{L_\rho}{\mu_\rho}}\log\left(\frac{C_\rho\Delta_{dual}}{\epsilon}\right)\right). \tag{5}$$

We remark that inexact accelerated Augmented Lagrangian methods have been previously analyzed under different assumptions [19, 28, 49]. The main difference is that here, we are able to establish a linear convergence rate, whereas existing analyses only yield sublinear rates. One of the reasons for this discrepancy is that, although F^* is strongly convex, the dual problem (D) is not, as the mixing matrix $\mathbf{W}$ is singular. The key to obtaining a linear convergence rate is a fine-grained analysis of the dual problem, showing that the dual variables always lie in the subspace where strong convexity holds. The proof of the theorem relies on the equivalence between Augmented Lagrangian methods and the dual proximal point algorithm [7, 35], which can be interpreted as applying an inexact accelerated proximal point algorithm [13, 23] to the dual problem. A complete convergence analysis is deferred to Section C in the appendix.

Theorem 2 provides an accelerated convergence rate with respect to the 'augmented' condition number $\kappa_\rho := L_\rho/\mu_\rho$, as determined by the Augmented Lagrangian parameter ρ in Algorithm 3. We have the following bounds:

$$\underbrace{1}_{\rho=\infty} \le \kappa_\rho = \frac{L+\rho\lambda^+_{\min}(W)}{\mu+\rho\lambda_{\max}(W)}\frac{\lambda_{\max}(W)}{\lambda^+_{\min}(W)} \le \underbrace{\frac{L}{\mu}\frac{\lambda_{\max}(W)}{\lambda^+_{\min}(W)}}_{\rho=0} = \kappa_f\kappa_W, \tag{6}$$

where we observe that the condition number κ_ρ is a decreasing function of the regularization parameter ρ. When $\rho = 0$, the maximum value is attained at $\kappa_\rho = \kappa_f\kappa_W$, the effective condition number of the decentralized problem. As ρ goes to infinity, the augmented condition number κ_ρ goes to 1. Naively, one may want to take ρ as large as possible to get a fast convergence. However, one must also take into account the complexity of solving the subproblems. Indeed, since W is singular, the additional regularization term in P_k does not improve the strong convexity of the subproblems, yielding an increase in inner loops complexity as ρ grows. Hence, the optimal choice of ρ requires balancing the inner and outer complexity in a careful manner.

To study the inner loop complexity, we introduce a warm-start strategy. Intuitively, the distance between $\mathbf{X}_{k-1}$ and the k-th solution $\mathbf{X}_k^*$ to the subproblem P_k is roughly on the order of ϵ_{k-1}. More precisely, we have the following result.

	GD	AGD	SGD
T_k	$\tilde{O}\left(\frac{L+\rho\lambda_{\max}(W)}{\mu}\right)$	$\tilde{O}\left(\sqrt{\frac{L+\rho\lambda_{\max}(W)}{\mu}}\right)$	$\tilde{O}\left(\frac{\sigma^2}{\mu^2\epsilon_k}\right)$
ρ	$\frac{L}{\lambda_{\max}(W)}$	$\frac{L}{\lambda_{\max}(W)}$	$\frac{L}{\lambda^+_{\min}(W)}$
$\sum_{k=1}^{K} T_k$	$\tilde{O}\left(\kappa_f\sqrt{\kappa_W}\log(\frac{1}{\epsilon})\right)$	$\tilde{O}\left(\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$	$\tilde{O}\left(\frac{\sigma^2\kappa_f\kappa_W}{\mu^2\epsilon}\right)$

Table 1: The first row indicates the number of iterations required for different inner solvers to achieve ϵ_k accuracy for the k-th subproblem P_k; the $\tilde{O}$ notation hides logarithmic factors in the parameters ρ, κ_f and κ_W. The second row shows the theoretical choice of the regularization parameter ρ. The last row shows the total number of iterations according to the choice of ρ.

Lemma 4. *Given the parameter choice in Theorem 2, initializing the subproblem P_k at $\mathbf{X}_{k-1}$ yields,*

$$\|\mathbf{X}_{k-1} - \mathbf{X}_k^*\|^2 \leq \frac{8C_\rho}{\mu_\rho}\epsilon_{k-1}.$$

Consequently, the ratio between the initial gap at the k-th subproblem and the desired gap ϵ_k is bounded by

$$\frac{\|\mathbf{X}_{k-1} - \mathbf{X}_k^*\|^2}{\epsilon_k} \leq \frac{8C_\rho}{\mu_\rho}\frac{\epsilon_{k-1}}{\epsilon_k} \leq \frac{16C_\rho}{\mu_\rho} = O(\kappa_f\kappa_W\rho^2),$$

which is independent of k. In other words, the inner loop solver only needs to decrease the iterate gap by a constant factor for each P_k. If the algorithm enjoys a linear convergence rate, a constant number of iteration is sufficient for that. If the algorithm enjoys a sublinear convergence, then the inner loop complexity grows with k. To illustrate the behaviour of different algorithms, we present the inner loop complexity T_k for gradient descent (GD), accelerated gradient descent (AGD) and stochastic gradient descent (SGD) in Table 1. Note that while the inner complexity of GD and AGD are independent of k, the inner complexity for SGD increases geometrically with k. Other possible choices for inner solvers are the alternating directions or Jacobi/Gauss-Seidel method, both of which yield accelerated variants for [40] and [15].

In fact, the theoretical upper bounds on the inner complexity also provide a more practical way to halt the inner optimization processes (see Option II in Algorithm 3). Indeed, one can predefine the computational budget for each subproblem, for instance, 100 iterations of AGD. If this budget exceeds the theoretical inner complexity T_k in Table 1, then the desired accuracy ϵ_k is guaranteed to be reached. In particular, we do not need to evaluate the sub-optimality condition, it is automatically satisfied as long as the budget is chosen appropriately.

Finally, the global complexity is obtained by summing $\sum_{k=1}^{K} T_k$, where K is the number of subproblems given in (5). Note that, so far, our analysis applies to any regularization parameter ρ. Since $\sum_{k=1}^{K} T_k$ is a function of ρ, this implies that one can select the parameter ρ such that the overall complexity is minimized, leading to the choices of ρ described in Table 1.

Two-fold acceleration In our setting, acceleration seems to occur in two stages (when compared to the non-accelerated $O\left(\kappa_f\kappa_W\log(\frac{1}{\epsilon})\right)$ rates in [15, 30, 34, 40, 41]). First, combining IDEAL with GD improves the dependence on the condition of the mixing matrix κ_W. Secondly, when used as an inner solver, AGD improves the dependence on the condition number of the local functions κ_f. This suggests that the two phenomena are independent; while one is related to the consensus between the agents, as governed by the mixing matrix, the other one is related to the respective centralized hardness of the optimization problem.

Stochastic oracle Our framework also subsumes the stochastic setting, where only noisy gradients are available. In this case, since SGD is sublinear, the required iteration counters T_k for the subproblem must increase inversely proportional to ϵ_k. Also the stepsize at the k-th iteration needs to be decreased accordingly. The overall complexity is now given by $\tilde{O}\left(\frac{\sigma^2\kappa_f\kappa_W}{\mu^2\epsilon}\right)$. However, in this case, the resulting dependence on the graph condition number can be improved [11].

	ρ	Computation cost	Communication cost
SSDA+AGD	0	$\tilde{O}\left(\kappa_f\sqrt{\kappa_W}\log(\frac{1}{\epsilon})\right)$	$O\left(\tau\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$
IDEAL+AGD	$\frac{L}{\lambda_{\max}(W)}$	$\tilde{O}\left(\sqrt{\kappa_f\kappa_W}\right)\log(\frac{1}{\epsilon})$	$\tilde{O}\left(\tau\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$
MSDA+AGD	0	$\tilde{O}\left(\kappa_f\log(\frac{1}{\epsilon})\right)$	$O\left(\tau\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$
MIDEAL+AGD	$\frac{L}{\lambda_{\max}(Q(W))}$	$\tilde{O}\left(\sqrt{\kappa_f}\log(\frac{1}{\epsilon})\right)$	$\tilde{O}\left(\tau\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$

Table 2: The communication cost of the presented algorithms are all optimal, but the computation cost differs. An additional factor of $\sqrt{\kappa_f}$ is introduced in SSDA/MSDA compared to their original rate in [37], due to the gradient approximation. The optimal computation cost is achieved by combining our multi-stage algorithm MIDEAL with AGD as an inner solver.

Multi-stage variant (MIDEAL) We remark that the complexity presented in Table 2 is abbreviated, in the sense that it does not distinguish between communication cost and computation cost. To provide a more fine-grained analysis, it suffices to note that performing a gradient step of the subproblem $\nabla P_k(\mathbf{X}) = \nabla F(\mathbf{X}) + \mathbf{\Omega}_k + \rho\mathbf{W}\mathbf{X}$ requires one local computation to evaluate ∇F, and one round of communication to obtain $\mathbf{W}\mathbf{X}$. This implies that when GD/AGD/SGD is combined with IDEAL, the number of local computation rounds is roughly the number of communication rounds, leading to a sub-optimal computation cost, as shown in Table 2.

To achieve optimal accelerated rates, we enforce multiple communication rounds after one evaluation of ∇F. This is achieved by substituting the regularization metric $\|\cdot\|^2_{\mathbf{W}}$ with $\|\cdot\|^2_{Q(\mathbf{W})}$, where Q is a well-chosen polynomial. In this case, the gradient of the subproblem becomes $\nabla P_k(\mathbf{X}) = \nabla F(\mathbf{X}) + \mathbf{\Omega}_k + \rho\, Q(\mathbf{W})\mathbf{X}$, which requires $\deg(Q)$ rounds of communication.

The choice of the polynomial Q relies on Chebyshev acceleration, which is introduced in [4, 37]. More concretely, the Chebyshev polynomials are defined by the recursion relation $T_0(x) = 1$, $T_1(x) = x$, $T_{j+1}(x) = 2xT_j(x) - T_{j-1}(x)$, and Q is defined by

$$Q(x) = 1 - \frac{T_{j_W}(c(1-x))}{T_{j_W}(c)} \quad \text{with} \quad j_W = \lfloor\sqrt{\kappa_W}\rfloor, \quad c = \frac{\kappa_W+1}{\kappa_W-1}. \tag{7}$$

Applying this specific choice of Q to the mixing matrix W reduces its condition number by the maximum amount [4, 37], yielding a graph independent bound $\kappa_{Q(W)} = \lambda_{\max}(Q(W))/\lambda^+_{\min}(Q(W)) \leq 4$. Moreover, the symmetry, positiveness and spectrum property in Assumption 1 are maintained by $Q(W)$. Even though $Q(W)$ no longer satisfies the decentralized property, it can be implemented using $\lfloor\sqrt{\kappa_W}\rfloor$ rounds of communications with respect to W. The implementation details of the resulting algorithm are similar to Algorithm 2, and follow by substituting the mixing matrix W by $Q(W)$ (Algorithm 5 in Appendix E).

Comparison with inexact SSDA/MSDA [37] Recall that SSDA/MSDA are special cases of our algorithmic framework with the degenerate regularization parameter $\rho = 0$. Therefore, our complexity analysis naturally extends to an inexact anlysis of SSDA/MSDA, as shown in Table 2. although the resulting communication costs are optimal, the computation cost is not, due to the additional $\sqrt{\kappa_f}$ factor introduced by solving the subproblems inexactly. In contrast, our multi-stage framework achieves the optimal computation cost.

- **Low communication cost regime:** $\tau\sqrt{\kappa_W} < 1$, the computation cost dominates the communication cost, a $\sqrt{\kappa_f}$ improvement is obtained by MIDEAL comparing to MSDA.
- **Ill conditioned regime:** $1 < \tau\sqrt{\kappa_W} < \sqrt{\kappa_f}$, the complexity of MSDA is dominated by the computation cost $\tilde{O}\left(\kappa_f\log(\frac{1}{\epsilon})\right)$ while the complexity MIDEAL is dominated by the communication cost $\tilde{O}\left(\tau\sqrt{\kappa_f\kappa_W}\log(\frac{1}{\epsilon})\right)$. The improvement is proportional to the ratio $\sqrt{\kappa_f}/\tau\sqrt{\kappa_W}$.
- **High communication cost regime:** $\sqrt{\kappa_f} < \tau\sqrt{\kappa_W}$, the communication cost dominates, and MIDEAL and MSDA are comparable.

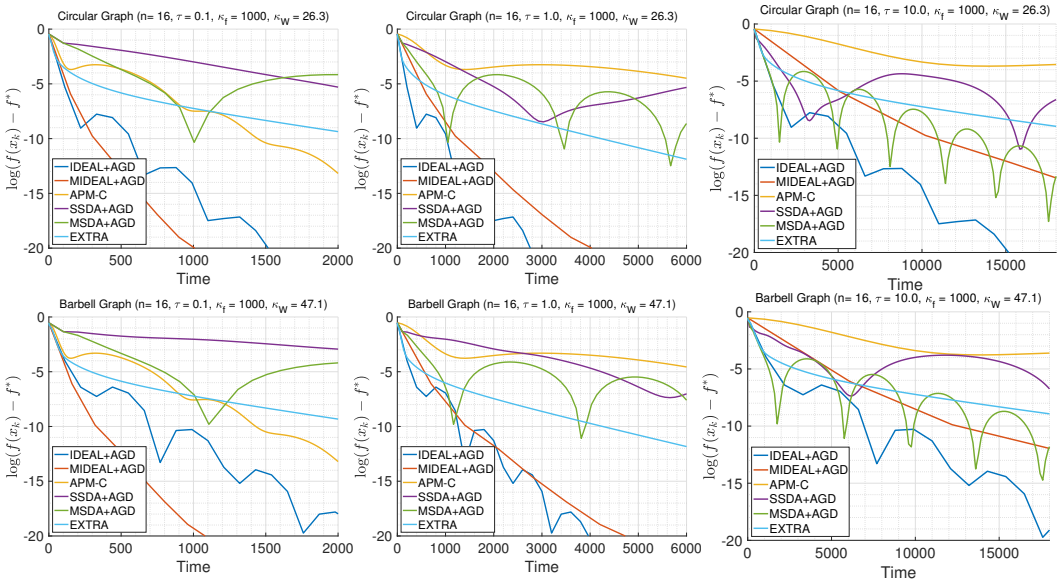

Figure 1: We evaluate the empirical performance of existing state-of-the-art algorithms, where the underlying network is a circular graph (top) and a barbell graph (bottom). We consider the following regimes: low communication cost (left), Ill-condition problems (middle) and High communication cost (right). The x-axis is the time counter, i.e. the sum of the communication cost and the computation cost; the y-axis is the log scale suboptimality. We observe that our algorithms IDEAL/MIDEAL are optimal under various regimes, validating our theoretical findings.

5 Experiments

Having described the IDEAL/MIDEAL algorithms for decentralized optimization problem (1), we now turn to presenting various empirical results which corroborate our theoretical analysis. To facilitate a simple comparison between existing state-of-the-art algorithms, we consider an ℓ_2-regularized logistic regression task over two classes of the MNIST [21] benchmark dataset. The smoohtness parameter (assuming normalized feature vectors) can be shown to be bounded by $1/4$, which together with a regularization parameter $\mu \approx 1\mathrm{e}{-3}$, yields a relatively high $1\mathrm{e}3$-bound on the condition number of the loss function. Further empirical results which demonstrate the robustness of IDEAL/MIDEAL under wide range of parameter choices are provided in Appendix G.

We compare the performance of IDEAL/MIDEAL with the state-of-the-art algorithms EXTRA [41], APM-C [22] and the inexact dual method SSDA/MSDA [37]. We set the inner iteration counter to be $T_k = 100$ for all algorithms, and use the theoretical stepsize schedule. The decentralized environment is modelled in a synthetic setting, where the communication time is steady and no latency is encountered. To demonstrate the effect of the underlying network architecture, we consider: a) a circular graph, where the agents form a cycle; b) a Barbell graph, where the agents are split into two complete subgraphs, connected by a single bridge (shown in Figure 2 in the appendix).

As shown in Figure 1, our multi-stage algorithm MIDEAL is optimal in the regime where the communication cost τ is small, and the single-stage variant IDEAL is optimal when τ is large. As expected, the inexactness mechanism significantly slows down the dual method SSDA/MSDA in the low communication cost regime. In contrast, the APM-C algorithm performs reasonably well in the low communication regime, but performs relatively poorly when the communication cost is high.

6 Conclusions

We propose a novel framework of decentralized algorithms for smooth and strongly convex objectives. The framework provides a unified viewpoint of several well-known decentralized algorithms and, when instantiated with AGD, achieves optimal convergence rates in theory and state-of-the-art performance in practice. We leave further generalization to (non-strongly) convex and non-smooth objectives to future work.

Acknowledgements

YA and JB acknowledge support from the Sloan Foundation and Samsung Research. BC and MG acknowledge support from the grants NSF DMS-1723085 and NSF CCF-1814888. HL and SJ acknowledge support by The Defense Advanced Research Projects Agency (grant number YFA17 N66001-17-1-4039). The views, opinions, and/or findings contained in this article are those of the author and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

Broader impact

Centralization of data is not always possible because of security and legacy concerns [12]. Our work proposes a new optimization algorithm in the decentralized setting, which can learn a model without revealing the privacy sensitive data. Potential applications include data coming from healthcare, environment, safety, etc, such as personal medical information [17, 18], keyboard input history [20, 27] and beyond.

References

[1] Y. Arjevani and O. Shamir. Communication complexity of distributed convex learning and optimization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

[2] Y. Arjevani and O. Shamir. On the iteration complexity of oblivious first-order optimization algorithms. In *International Conferences on Machine Learning (ICML)*, 2016.

[3] Y. Arjevani, O. Shamir, and N. Srebro. A tight convergence analysis for stochastic gradient descent with delayed updates. In *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117, pages 111–132, 2020.

[4] W. Auzinger and J. Melenk. Iterative solution of large linear systems. *Lecture Note*, 2011.

[5] N. S. Aybat and M. Gürbüzbalaban. Decentralized computation of effective resistances and acceleration of consensus algorithms. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 538–542. IEEE, 2017.

[6] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[7] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[8] B. Can, S. Soori, N. S. Aybat, M. M. Dehvani, and M. Gürbüzbalaban. Decentralized computation of effective resistances and acceleration of distributed optimization algorithms. *arXiv preprint arXiv:1907.13110*, 2019.

[9] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3): 592–606, 2011.

[10] D. Dvinskikh and A. Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.

[11] A. Fallah, M. Gürbüzbalaban, A. Ozdaglar, U. Simsekli, and L. Zhu. Robust distributed accelerated stochastic gradient methods for multi-agent networks. *arXiv preprint arXiv:1910.08701*, 2019.

[12] GDPR. The eu general data protection regulation (gdpr). 2016.

[13] O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

[14] H. Hendrikx, F. Bach, and L. Massoulie. An optimal algorithm for decentralized finite sum optimization, 2020.

[15] D. Jakovetić, J. M. Moura, and J. Xavier. Linear convergence rate of a class of distributed augmented lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2014.

[16] D. Jakovetić, J. Xavier, and J. M. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[17] A. Jochems, T. M. Deist, J. Van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin, and A. Dekker. Distributed learning: developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept. *Radiotherapy and Oncology*, 121(3):459–467, 2016.

[18] A. Jochems, T. M. Deist, I. El Naqa, M. Kessler, C. Mayo, J. Reeves, S. Jolly, M. Matuszak, R. Ten Haken, J. van Soest, et al. Developing and validating a survival prediction model for nsclc patients through distributed learning across 3 countries. *International Journal of Radiation Oncology* Biology* Physics*, 99(2):344–352, 2017.

[19] M. Kang, M. Kang, and M. Jung. Inexact accelerated augmented lagrangian methods. *Computational Optimization and Applications*, 62(2):373–404, 2015.

[20] J. Konevcný, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL https://arxiv.org/abs/1610.05492.

[21] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]*, 2, 2010. URL http://yann.lecun.com/exdb/mnist.

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

[23] H. Lin, J. Mairal, and Z. Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *The Journal of Machine Learning Research*, 18(1):7854–7907, 2017.

[24] J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2016.

[25] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4):2322–2358, 2017.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[27] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

[28] V. Nedelcu, I. Necoara, and Q. Tran-Dinh. Computational complexity of inexact gradient augmented lagrangian methods: application to constrained mpc. *SIAM Journal on Control and Optimization*, 52(5):3109–3134, 2014.

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

[30] A. Nedic, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

[31] A. Nedić, A. Olshevsky, W. Shi, and C. A. Uribe. Geometrically convergent distributed optimization with uncoordinated step-sizes. In *2017 American Control Conference (ACC)*, pages 3950–3955. IEEE, 2017.

[32] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

[33] L. Panait and S. Luke. Cooperative multi-agent learning: The state of the art. *Autonomous agents and multi-agent systems*, 11(3):387–434, 2005.

[34] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2017.

[35] R. T. Rockafellar. Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

[36] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.

[37] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conferences on Machine Learning (ICML)*, 2017.

[38] K. Scaman, F. Bach, S. Bubeck, L. Massoulié, and Y. T. Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.

[39] M. Schmidt, N. L. Roux, and F. R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2011.

[40] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7): 1750–1761, 2014.

[41] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[42] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[43] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[44] Y. Sun, A. Daneshmand, and G. Scutari. Convergence rate of distributed optimization algorithms based on gradient tracking. *arXiv preprint arXiv:1905.02637*, 2019.

[45] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić. A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pages 1–40, 2020.

[46] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2018.

[47] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1):33–46, 2007.

[48] J. Xu, Y. Tian, Y. Sun, and G. Scutari. Accelerated primal-dual algorithms for distributed smooth convex optimization over networks. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[49] S. Yan and N. He. Bregman augmented lagrangian and its acceleration, 2020.

[50] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[51] J. Zhang, C. A. Uribe, A. Mokhtari, and A. Jadbabaie. Achieving acceleration in distributed optimization via direct discretization of the heavy-ball ode. In *2019 American Control Conference (ACC)*, pages 3408–3413. IEEE, 2019.

A Remark on the choice of the mixing matrix

In the main paper, the mixing matrix W is defined following the convention used in [37], where the kernel of W is the vector of all ones. It is worth noting that the term mixing matrix is also used in the literature to denote a doubly stochastic matrix W_{DS} (see e.g. [8, 15, 30, 31, 34, 40, 41]). These two approaches are equivalent as given a doubly stochastic matrix W_{DS}, the matrix

$$I - W_{DS} \text{ is a mixing matrix under Definition 1.}$$

In the following discussion, we will use W_{DS} to draw the connection when necessary.

B Recovering EXTRA under the augmented Lagrangian framework

The goal of this section is to show that EXTRA algorithm [41] is a special case of the non-accelerated Augmented Lagrangian framework in Algorithm 1.

Proposition 5. *The EXTRA algorithm is equivalent to applying one step of gradient descent to solve the subproblem in Algorithm 1.*

Proof. Taking a single step of gradient descent in the subproblem P_k in Algorithm 1 warm starting at X_{k-1} yields the update

$$\begin{aligned} X_k &= X_{k-1} - \alpha(\nabla F(X_{k-1}) + \Lambda_k + \rho W X_{k-1}). \\ \Lambda_{k+1} &= \Lambda_k + \eta W X_k. \end{aligned} \tag{8}$$

Using the $(k+1)$-th update,

$$X_{k+1} = X_k - \alpha(\nabla F(X_k) + \Lambda_{k+1} + \rho W X_k). \tag{9}$$

and subtracting (8) from (9) gives

$$X_{k+1} = (2 - \alpha(\rho+\eta)W)X_k - (1 - \alpha\rho W)X_{k-1} - \alpha(\nabla F(X_k) - \nabla F(X_{k-1})).$$

When incorporating with the mixing matrix $W = I - W_{DS}$ and taking $\rho = \eta = \frac{1}{2\alpha}$ gives,

$$X_{k+1} = (I + W_{DS})X_k - \left(I + \frac{W_{DS}}{2}\right) X_{k-1} - \alpha(\nabla F(X_k) - \nabla F(X_{k-1})),$$

which is the update rule of EXTRA [41]. □

Remark 6. *When expressing the parameters in terms of ρ, the inner loop stepsize reads as $\alpha = \frac{1}{2\rho}$, and the outer-loop stepsize reads as $\eta = \rho$.*

C Proof of Theorem 3

Algorithm 4 (Unscaled) Accelerated Decentralized Augmented Lagrangian framework

Input: mixing matrix W, regularization parameter ρ, stepsize η, extrapolation parameters $\{\beta_k\}_{k\in\mathbb{N}}$

1: Initialize dual variables $\mathbf{\Lambda}_1 = \mathbf{\Omega}_1 = \mathbf{0} \in \mathbb{R}^{nd}$.
2: **for** $k = 1, 2, ..., K$ **do**
3: $\quad \mathbf{X}_k \approx \arg\min \left\{ P_k(\mathbf{X}) := F(\mathbf{X}) + (\sqrt{\mathbf{W}}\mathbf{\Omega}_k)^T\mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|^2_{\mathbf{W}} \right\}$.
4: $\quad \mathbf{\Lambda}_{k+1} = \mathbf{\Omega}_k + \eta\sqrt{\mathbf{W}}\mathbf{X}_k$
5: $\quad \mathbf{\Omega}_{k+1} = \mathbf{\Lambda}_{k+1} + \beta_{k+1}(\mathbf{\Lambda}_{k+1} - \mathbf{\Lambda}_k)$
6: **end for**

Output: $\mathbf{X}_K$.

We start by noting that Algorithm 2 is equivalent to the "unscaled" version of Algorithm 4. More specifically, we recover Algorithm 2 by substituting the variables

$$\Lambda \leftarrow \sqrt{\mathbf{W}}\Lambda, \quad \Omega \leftarrow \sqrt{\mathbf{W}}\Omega.$$

The unscaled version is computationally inefficient since it requires the computation of the square root of W. This is the reason why we choose to present the scaled version Algorithm 2 in the main paper. However, the unscaled version is easier to work with for the analysis. **In the following proof, the variables $\mathbf{\Lambda}$ and $\mathbf{\Omega}$ are referred to as in the unscaled version Algorithm 4.**

The key concept underlying our analysis on is the Moreau-envelope of the dual problem:

$$\Phi_\rho(\Lambda) = \min_{\Gamma \in \mathbb{R}^{nd}} \left\{ F^*(-\sqrt{W}\Gamma) + \frac{1}{2\rho}\|\Gamma - \Lambda\|^2 \right\}. \tag{10}$$

Similarly, we define the associated proximal operator

$$\text{prox}_{\Phi_\rho}(\Lambda) = \arg\min_{\Gamma \in \mathbb{R}^{nd}} \left\{ F^*(-\sqrt{W}\Gamma) + \frac{1}{2\rho}\|\Gamma - \Lambda\|^2 \right\}. \tag{11}$$

Note that when the inner problem is strongly convex, the proximal operator is unique (that is, a single-valued operator). The following is a list well known properties of the Moreau-envelope:

Proposition 7. *The Moreau envelope Φ_ρ enjoys the following properties*

1. *Φ_ρ is convex and it shares the same optimum as the dual problem (D).*
2. *Φ_ρ is differentiable and the gradient of Φ_ρ is given by*

$$\nabla \Phi_\rho(\mathbf{\Lambda}) = \frac{1}{\rho}(\mathbf{\Lambda} - \text{prox}_{\Phi_\rho}(\mathbf{\Lambda}))$$

3. *If F is twice differentiable, then its convex conjugate F^* is also twice differentiable. In this case, Φ_ρ is also twice differentiable and the Hessian is given by*

$$\nabla^2 \Phi_\rho(\Lambda) = \frac{1}{\rho} I - \frac{1}{\rho^2}\left[\frac{1}{\rho} I + \sqrt{\mathbf{W}}\nabla^2 F^*(-\sqrt{\mathbf{W}}\,\text{prox}_{\Phi_\rho}(\Lambda))\sqrt{\mathbf{W}}\right]^{-1}.$$

Corollary 8. *The Moreau envelope Φ_ρ satisfies*

1. *Φ_ρ is L_ρ-smooth, where $L_\rho = \frac{\lambda_{\max}(W)}{\mu + \rho\lambda_{\max}(W)} \leq \frac{1}{\rho}$.*
2. *Φ_ρ is μ_ρ-strongly convex in the image space of $\sqrt{W}$, where $\mu_\rho = \frac{\lambda^+_{\min}(W)}{L + \rho\lambda^+_{\min}(W)}$.*

Proof. These properties follow from the expressions for the Hessian of Φ_ρ and by the fact that F^* is $\frac{1}{\mu}$-smooth and $\frac{1}{L}$ strongly convex. □

In particular, Φ_ρ is only strongly convex on the image space of $\sqrt{W}$, one of the keys to prove the linear convergence rate is the following lemma.

Lemma 9. *The variables $\mathbf{\Lambda}_k$ and $\mathbf{\Omega}_k$ in the un-scaled version Algorithm 4 all lie in the image space of $\sqrt{W}$ for any k.*

Proof. This can be easily derived by induction according to the update rule in line 4, 5 of Algorithm 4. □

Similar to the dual Moreau-envelope, we also define the weighted Moreau-envelope on the primal function

$$\Psi_\rho(\Omega) = \min_{\mathbf{X}} \left\{ F(\mathbf{X}) + \Omega^T\mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|^2_{\mathbf{W}} \right\} \tag{12}$$

and its associated proximal operator

$$\text{prox}_{\Psi_\rho}(\Omega) = \arg\min_{\mathbf{X}} \left\{ F(\mathbf{X}) + \Omega^T\mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|^2_{\mathbf{W}} \right\}. \tag{13}$$

Indeed, this function corresponds exactly to the subproblem solved in the augmented Lagrangian framework (line 3 of Algorithm 2). Similar property holds for Ψ_ρ:

Proposition 10. *The Moreau envelope Ψ_ρ enjoys the following properties:*

1. *Ψ_ρ is concave.*
2. *Ψ_ρ is differentiable and the gradient of Ψ_ρ is given by*
$$\nabla \Psi_\rho(\Omega) = \mathrm{prox}_{\Psi}(\Omega).$$
3. *If F is twice differentiable, then Ψ_ρ is also twice differentiable and the Hessian is given by*
$$\nabla^2 \Psi_\rho(\Omega) = -\left[\nabla^2 F(\mathrm{prox}_{\Psi}(\Omega)) + \rho W\right]^{-1}.$$
In particular, Ψ_ρ is $\frac{1}{\mu}$-smooth and $\frac{1}{L+\rho\lambda_{\max}(W)}$ strongly concave.

The dual Moreau-envelope Φ_ρ and primal Moreau-envelope Ψ_ρ are connected through the following relationship.

Proposition 11. *The gradient of the Moreau envelope Φ_ρ is given by*
$$\nabla \Phi_\rho(\mathbf{\Lambda}) = -\sqrt{\mathbf{W}} \nabla \Psi_\rho(\sqrt{\mathbf{W}}\mathbf{\Lambda}). \tag{14}$$

Proof. To simplify the presentation, let us denote
$$\mathbf{X}(\mathbf{\Lambda}) = \arg\min_{\mathbf{X}} \left\{ F(\mathbf{X}) + (\sqrt{\mathbf{W}}\mathbf{\Lambda})^T \mathbf{X} + \frac{\rho}{2}\|\mathbf{X}\|_{\mathbf{W}}^2 \right\} = \nabla \Psi_\rho(\sqrt{\mathbf{W}}\mathbf{\Lambda}).$$
From the optimality of $\mathbf{X}(\mathbf{\Lambda})$, we have
$$\nabla F(\mathbf{X}(\mathbf{\Lambda})) + \sqrt{\mathbf{W}}\mathbf{\Lambda} + \rho \mathbf{W}\mathbf{X}(\mathbf{\Lambda}) = 0$$
From the fact that $\nabla F(x) = y \Leftrightarrow \nabla F^*(y) = x$, we have
$$\mathbf{X}(\mathbf{\Lambda}) = \nabla F^* \left(-\sqrt{\mathbf{W}} \left[\mathbf{\Lambda} + \rho \sqrt{\mathbf{W}} \mathbf{X}(\mathbf{\Lambda}) \right] \right).$$
Let $\mathbf{\Gamma} = \mathbf{\Lambda} + \rho\sqrt{\mathbf{W}}\mathbf{X}(\mathbf{\Lambda})$, then
$$-\sqrt{\mathbf{W}} \nabla F^*(-\sqrt{\mathbf{W}}\mathbf{\Gamma}) + \frac{1}{\rho}(\mathbf{\Gamma} - \mathbf{\Lambda}) = 0.$$
Therefore $\mathbf{\Gamma}$ is the minimizer of the function $F^*(-\sqrt{W}\Gamma) + \frac{1}{2\rho}\|\Gamma - \Lambda\|^2$, namely
$$\mathrm{prox}_{\Phi_\rho}(\mathbf{\Lambda}) = \mathbf{\Lambda} + \rho\sqrt{\mathbf{W}}\mathbf{X}(\mathbf{\Lambda}).$$
Then based on the expression for the gradient in Prop 7, we obtain the desired equality (14). □

Proposition 14 demonstrates that solving the augmented Lagrangian subproblem could be viewed as evaluating the gradient of the Moreau-envelope. Hence applying gradient descent on the Moreau-envelope gives the non-accelerated augmented Lagrangian framework Algorithm 1. Even more, applying Nesterov's accelerated gradient on the Moreau-envelope Φ_ρ yields accelerated Augmented Lagrangian Algorithm 4. In addition, when the subproblems are solved inexactly, this corresponds to an inexact evaluation on the gradient. This interpretation allows us to derive guarantees for the convergence rate of the dual variables. Before present the the convergence analysis in detail, we formally establish the connection between the primal solution and the dual solution.

Lemma 12. *Let x^* be the optimum of f and define $\mathbf{X}^* = \mathbf{1}_n \otimes x^* \in \mathbb{R}^{nd}$. Then there exists a unique $\mathbf{\Lambda}^* \in Im(\mathbf{W})$ such that $\mathbf{\Lambda}^*$ is the optimum of the dual problem (D). Moreover, it satisfies*
$$\nabla F(\mathbf{X}^*) = -\sqrt{\mathbf{W}}\mathbf{\Lambda}^*.$$

Proof. Since $Ker(W) = \mathbb{R}\mathbf{1}_n$, we have
$$Ker(\mathbf{W}) = Ker(W \otimes I_d) = Vect(\mathbf{1}_n \otimes \mathbf{e}_i, i = 1, \cdots, d),$$

where $\mathbf{e}_i$ is the canonical basis with all entries 0 except the i-th equals to 1. By optimality, $\nabla f(x^*) = \sum_{i=1}^n \nabla f_i(x^*) = 0$. This implies that $\nabla F(x^*)^T(\mathbf{1}_n \otimes \mathbf{e}_i) = 0$, for all $i = 1, \cdots d$. In other words, $\nabla F(X^*)$ is orthogonal to the null space of $\mathbf{W}$, namely $\nabla F(X^*) \in Im(\mathbf{W})$. Therefore, there exists $\mathbf{\Lambda}$ such that $\nabla F(\mathbf{X}^*) = -\mathbf{W}\mathbf{\Lambda}$. By setting $\mathbf{\Lambda}^* = \sqrt{\mathbf{W}}\mathbf{\Lambda}$, we have $\mathbf{\Lambda}^* \in Im(\mathbf{W})$ and $\nabla F(\mathbf{X}^*) = -\sqrt{\mathbf{W}}\mathbf{\Lambda}^*$. In particular, since $\nabla F(x) = y \Leftrightarrow \nabla F^*(y) = x$, we have,

$$\sqrt{\mathbf{W}}\nabla F^*(-\sqrt{\mathbf{W}}\mathbf{\Lambda}^*) = \sqrt{\mathbf{W}}\mathbf{X}^* = 0. \tag{15}$$

Hence $\mathbf{\Lambda}^*$ is the solution of the dual problem (D) and it is the unique one lies in the $Im(\mathbf{W})$. □

Throughout the rest of the paper, we use Λ^* to denote the unique solution as shown in the lemma above. We would like to emphasize that even though F^* is strongly convex, the dual problem (D) is not strongly convex, because W is singular. Hence, the solution of the dual problem is not unique unless we restrict to the image space of $\mathbf{W}$. To derive the linear convergence rate, we need to show that the dual variable always lies in this subspace where the Moreau-envelope Φ_ρ is strongly convex.

Theorem 13. *Consider the sequence of primal variables $(\mathbf{X}_k)_{k\in\mathbb{N}}$ generated by Algorithm 3 with the subproblem solved up to ϵ_k accuracy, i.e. Option I. Therefore,*

$$\|\mathbf{X}_{k+1} - \mathbf{X}^*\|^2 \leq 2\epsilon_{k+1} + C\left(1 - \sqrt{\frac{\mu_\rho}{L_\rho}}\right)^k \left(\sqrt{\mu_\rho \Delta_{dual}} + A_k\right)^2 \tag{16}$$

where $\mathbf{X}^* = \mathbf{1}_n \otimes x^*$, $L_\rho = \frac{\lambda_{\max}(W)}{\mu + \rho\lambda_{\max}(W)}$, $\mu_\rho = \frac{\lambda^+_{\min}(W)}{L + \rho\lambda^+_{\min}(W)}$, $C = \frac{2\lambda_{\max}(W)}{\mu^2\mu_\rho^2}$, Δ_{dual} *is the dual function gap defined by* $\Delta_{dual} = F^*(-\sqrt{\mathbf{W}}\mathbf{\Lambda}_1) - F^*(-\sqrt{\mathbf{W}}\mathbf{\Lambda}^*)$ *and* $A_k = \sqrt{\lambda_{\max}(W)}\sum_{i=1}^k \sqrt{\epsilon_i}\left(1 - \sqrt{\frac{\mu_\rho}{L_\rho}}\right)^{-i/2}$.

Proof. The proof builds on the concepts developed so far in this section. We start by showing that the dual variable $\mathbf{\Lambda}_k$ converges to the dual solution $\mathbf{\Lambda}^*$ in a linear rate. From the interpretation given in Prop 7 and Prop 11, the sequence $(\Lambda_k)_{k\in\mathbb{N}}$ given in Algorithm 2 is equivalent to applying Nesterov's accelerated gradient method on the Moreau-envelope Φ_ρ. In the inexact variant, the inexactness on the solution directly translates to an inexact gradient of Φ_ρ, where the inexactness is given by

$$\|e_k\| = \|\sqrt{\mathbf{W}}(X_k - X_k^*)\| \leq \sqrt{\lambda_{\max}(W)}\|X_k - X_k^*\| \leq \sqrt{\lambda_{\max}(W)\epsilon_k}.$$

Hence $(\Lambda_k)_{k\in\mathbb{N}}$ in Algorithm 4 is obtained by applying inexact accelerated gradient method on the Moreau-envelope Φ_ρ. Note that by induction Λ_k and Ω_k belong to the image space of $\sqrt{\mathbf{W}}$, in which the dual Moreau-envelope Φ_ρ is strongly convex. Following the analysis on inexact accelerated gradient method Prop 4 in [39], we have

$$\frac{\mu_\rho}{2}\|\Lambda_{k+1} - \Lambda^*\|^2 \leq \left(1 - \sqrt{\frac{\mu_\rho}{L_\rho}}\right)^{k+1}\left(\sqrt{2\Delta_{\Phi_\rho}} + \sqrt{\frac{2}{\mu_\rho}}A_k\right)^2 \tag{17}$$

where $\Delta_{\Phi_\rho} = \Phi_\rho(\Lambda_1) - \Phi_\rho^*$ and A_k is the accumulation of the errors given by

$$A_k = \sum_{i=1}^k \|e_i\|\left(1 - \sqrt{\frac{\mu_\rho}{L_\rho}}\right)^{-i/2} \leq \sum_{i=1}^k \sqrt{\lambda_{\max}(W)\epsilon_i}\left(1 - \sqrt{\frac{\mu_\rho}{L_\rho}}\right)^{-i/2}.$$

Based on the convergence on the dual variable, we could now derive the convergence on the primal variable. Let X_{k+1}^* be the exact solution of the problem P_{k+1}. Then

$$\begin{aligned}
\|\mathbf{X}_{k+1}^* - \mathbf{X}^*\| &= \|\nabla\Psi_\rho(\sqrt{\mathbf{W}}\mathbf{\Lambda}_{k+1}) - \nabla\Psi_\rho(\sqrt{\mathbf{W}}\mathbf{\Lambda}^*)\| \\
&\leq \frac{1}{\mu}\|\sqrt{\mathbf{W}}(\mathbf{\Lambda}_{k+1} - \mathbf{\Lambda}^*)\| \quad \text{(From Prop 10.3)} \\
&\leq \frac{\sqrt{\lambda_{\max}(W)}}{\mu}\|\mathbf{\Lambda}_{k+1} - \mathbf{\Lambda}^*\|.
\end{aligned} \tag{18}$$

Finally, from triangle inequality

$$\begin{aligned}\|\mathbf{X}_{k+1}-\mathbf{X}^*\|^2 &\leq 2\|\mathbf{X}_{k+1}-\mathbf{X}^*_{k+1}\|^2+2\|\mathbf{X}^*_{k+1}-\mathbf{X}^*\|^2\\ &\leq 2\epsilon_{k+1}+\frac{2\lambda_{\max}(W)}{\mu^2\mu_\rho}(1-\sqrt{\kappa_\rho})^{k+1}\left(\sqrt{2\Delta_{\Phi_\rho}}+\sqrt{\frac{2}{\mu_\rho}}A_k\right)^2.\end{aligned}$$

The desired inequality follows from reorganizing the constant and the fact that $\Delta_{\Phi_\rho}\leq\Delta_{dual}$. □

Proof of Theorem 2. Plugging in the choice of $\epsilon_k=\frac{\mu_\rho}{2\lambda_{\max}(W)}\left(1-\frac{1}{2}\sqrt{\frac{\mu_\rho}{L_\rho}}\right)^k\Delta_{dual}$ in (16) yields the desired convergence rate.

□

D Proof of Lemma 4

Lemma 14. *With the parameter choice as Theorem 2, then warm starting the k-th subproblem P_k at the previous solution $\mathbf{X}_{k-1}$ gives an initial gap*

$$\|\mathbf{X}_{k-1}-\mathbf{X}^*_k\|^2\leq 8\frac{C_\rho}{\mu_\rho}\epsilon_{k-1}.$$

Proof. From triangle inequality, we have

$$\|\mathbf{X}_{k-1}-\mathbf{X}^*_k\|^2\leq 2(\|\mathbf{X}_{k-1}-\mathbf{X}^*\|^2+\|\mathbf{X}^*_k-\mathbf{X}^*\|^2)$$

The desired inequality follows from the convergence on the primal iterates and (18), i.e.

$$\|\mathbf{X}_{k-1}-\mathbf{X}^*_k\|^2\leq\frac{2C_\rho}{\mu_\rho}\epsilon_{k-1},\quad \|\mathbf{X}^*_k-\mathbf{X}^*_k\|^2\leq\frac{2C_\rho}{\mu_\rho}\epsilon_k.$$

□

E Multi-stage algorithm: MIDEAL

Algorithm 5 MIDEAL: Multi-stage Inexact Acc-Decentralized Augmented Lagrangian framework

Input: mixing matrix W, regularization parameter ρ, stepsize η, extrapolation parameters $\{\beta_k\}_{k\in\mathbb{N}}$

1: Initialize dual variables $\mathbf{\Lambda}_1=\mathbf{\Omega}_1=\mathbf{0}\in\mathbb{R}^{nd}$ and the polynomial Q according to (7).
2: **for** $k=1,2,...,K$ **do**
3: $\quad\mathbf{X}_k\approx\arg\min\left\{P_k(\mathbf{X}):=F(\mathbf{X})+\mathbf{\Omega}_k^T\mathbf{X}+\frac{\rho}{2}\|\mathbf{X}\|^2_{Q(\mathbf{W})}\right\}$.
4: $\quad\mathbf{\Lambda}_{k+1}=\mathbf{\Omega}_k+\eta Q(\mathbf{W})\mathbf{X}_k$
5: $\quad\mathbf{\Omega}_{k+1}=\mathbf{\Lambda}_{k+1}+\beta_{k+1}(\mathbf{\Lambda}_{k+1}-\mathbf{\Lambda}_k)$
6: **end for**

Output: $\mathbf{X}_K$.

Algorithm 6 AcceleratedGossip [37]

Input: mixing matrix W, vector or matrix X.

1: Set parameters $\kappa_W=\frac{\lambda_{\max}(W)}{\lambda^+_{\min}(W)}$, $c_2=\frac{\kappa_W+1}{\kappa_W-1}$, $c_3=\frac{2}{(\kappa_W+1)\lambda^+_{\min}(W)}$, # of iterations $J=\lfloor\sqrt{\kappa_W}\rfloor$.
2: Initialize coefficients $a_0=1$, $a_1=c_2$, iterates $X_0=X$, $X_1=c_2(I-c_3W)X$.
3: **for** $j=1,2,...,J-1$ **do**
4: $\quad a_{j+1}=2c_2a_j-a_{j-1}$
5: $\quad X_{j+1}=2c_2(1-c_3W)X_j-X_{j-1}$
6: **end for**

Output: $X_0-\frac{X_J}{a_J}$.

Intuitively, we simply replace the mixing matrix W by $Q(W)$, resulting in a better graph condition number. However, each evaluation of the new mixing matrix $Q(W)$ requires $deg(Q)$ rounds of communication, given by the AcceleratedGossip algorithm introduced in [37]. For completeness of the discussion, we recall this procedure in Algorithm 6. In particular, given W and X, AcceleratedGossip(W,X) returns $Q(W)X$, based on the communication oracle W.

F Implementation of Algorithms

Algorithm 7 Implementation: IDEAL+AGD solver

Input: number of iterations $K > 0$, gossip matrix $W \in \mathbb{R}^{n\times n}$

$\omega_i(0) = \vec{0}$, $\gamma_i(0) = \vec{0}$, $x_i(0) = \overline{x_i}(0) = x_0$ for any $i \in [1, n]$

$\kappa_{inner} = \frac{L+\rho\lambda_{\max}(W)}{\mu}$, $\beta_{inner} = \frac{\sqrt{\kappa_{inner}}-1}{\sqrt{\kappa_{inner}}+1}$, $\kappa_\rho = \frac{L+\rho\lambda^+_{\min}(W)}{\mu+\rho\lambda_{\max}(W)}\frac{\lambda_{\max}(W)}{\lambda^+_{\min}(W)}$, $\beta_{outer} = \frac{\sqrt{\kappa_{outer}}-1}{\sqrt{\kappa_{outer}}+1}$

for $k = 1, 2, ..., K$ **do**

  Inner iteration: Approximately solve the augmented Lagrangian multiplier.

    Set $x_{i,k}(0) = y_{i,k}(0) = x_i(k-1)$, $\overline{x_{i,k}}(0) = \overline{y_{i,k}}(0) = \sum_{j\sim i} W_{ij}x_{j,k}(0)$

    for $t = 0, 1, ..., T-1$ **do**

      $x_{i,k}(t+1) = y_{i,k}(t) - \eta(\gamma_i(k) + \nabla f_i(y_{i,k}(t)) + \rho\overline{y_{i,k}}(t))$

      $y_{i,k}(t+1) = x_{i,k}(t+1) + \beta_{inner}(x_{i,k}(t+1) - x_{i,k}(t))$

      $\overline{y_{i,k}}(t+1) = \sum_{j\sim i} W_{ij}y_{j,k}(t+1)$

    end for

    Set $x_i(k) = x_{i,k}(T)$, $\overline{x_i}(k) = \sum_{j\sim i} W_{ij}x_{j,k}(T)$

  Outer iteration: Update the dual variables on each node

    $\lambda_i(k+1) = \omega_i(k) + \rho\overline{x_i}(k)$

    $\omega_i(k+1) = \lambda_i(k+1) + \beta_{outer}(\lambda_i(k+1) - \lambda_i(k))$

end for

Output:

G Further Experimental Results

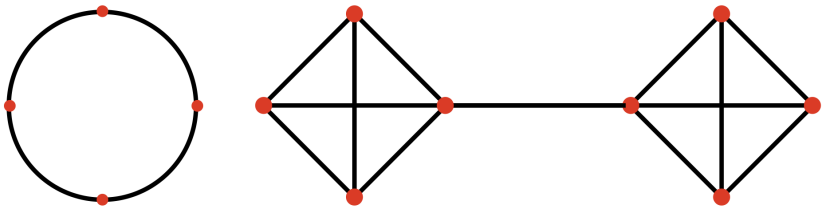

Figure 2: Network Structures: **Left:**Circular graph with 4 nodes. **Right:**Barbell graph with 8 nodes.

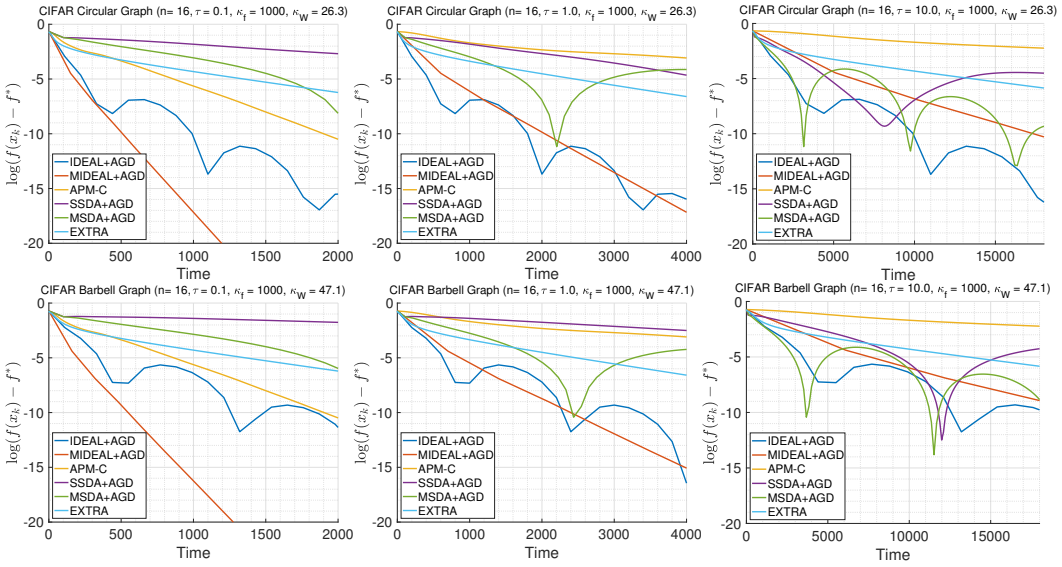

Figure 3: **CIFAR experiments**: we conduct experiments on two classes of CIFAR dataset, where the feature representation of each image was computed using an unsupervised convolutional kernel network Mairal [24]. We observe similar phenomenon as in the MNIST experiment, that the multi-stage algorithm MIDEAL outperforms when the communication cost τ is low and the IDEAL outperforms in the other cases.

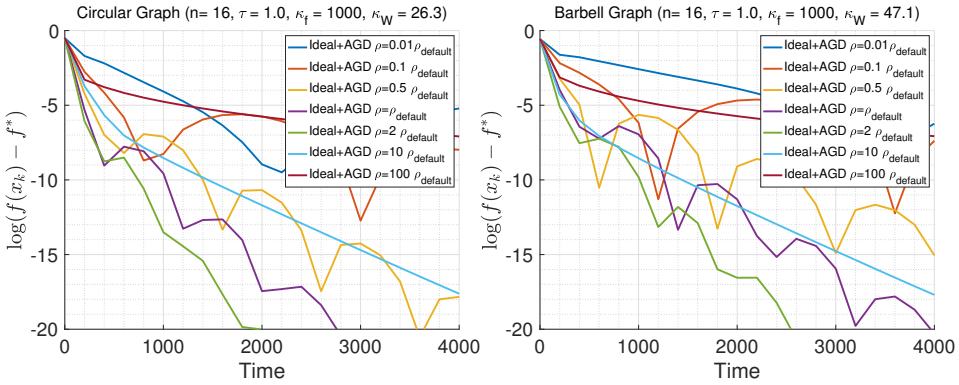

Figure 4: Ablation study on the **regularization parameter** ρ in IDEAL framework. For all the experiments, we use AGD as inner loop solver and set the same parameters as predicted by theory. We observe that when ρ is selected in the range $[0.5\rho_{\text{default}}, 10\rho_{\text{default}}]$, the perfomance of the algorithm is quite similar and robust. We also observe that using a small ρ degrades the performance of the algorithm, this phenomenon is consistent with the observation that the inexact SSD [37] does not perform well since it uses $\rho = 0$. Another observation is that with larger ρ, such as $\rho = 2\rho_{\text{default}}$ or $10\rho_{\text{default}}$, the algorithm is more stable with less zigzag oscillation, which is preferable in practice.

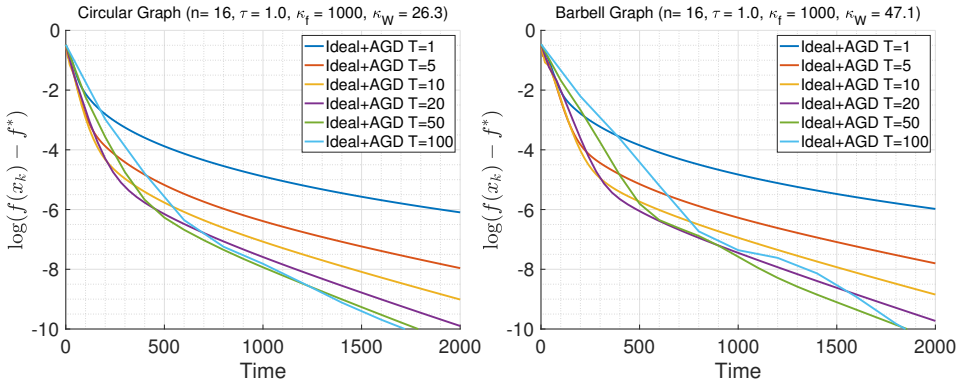

Figure 5: Ablation study on the **inner loop complexity** T_k in IDEAL framework. When the inner loop iteration is small, the algorithm becomes less stable, so we have decreased the momentum parameters to ensure the convergence. For these experiments, we use AGD solver with $\beta_{in} = 0.8$ and $\beta_{out} = 0.4$. As we can see, it is beneficial to perform multiple iterations in the inner loop rather than taking T=1 as in the EXTRA algorithm [41].